JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang *P ISSN : 2549-0710*
Tangerang, April 2025, Vol 9, No. 2 pp 135-142 *E ISSN : 2722-2713*

IMPLEMENTASI MICROSERVICE PADA APLIKASI MANAJEMEN PROYEK PERANGKAT LUNAK YANG TERINTEGRASI DENGAN GITEA

Jose Ryu Leonesta[1], Simon Prananta Barus[2]
[1,2] Jurusan Teknik Informatika, Fakultas Sains Teknologi Matematika, Universitas Matana, Serpong
Matana University Tower Jl. CBD Barat Kav, Kelapa Dua, Tangerang Regency, Banten 15810
Co Responden Email: jose.ryu.leonesta@gmail.com

Article history
Received 21 Jan 2025
Revised 20 Mar 2025
Accepted 15 Apr 2025
Available online 30 May 2025

Keywords
Integrated System,
Microservice,
Version Control,
Web Application.

Abstract

Project management has evolved from traditional methods to agile approaches, reducing documentation and using source code as the primary project document to reflect the features and flow of software systems. PT Gaws Inti Solusi, a software company, faces challenges in using Gitea for version control and workload distribution due to an interface that does not meet their needs. This research develops a web-based project management system integrated with Gitea, utilizing microservice architecture and the Spiral methodology to enhance productivity and collaboration in project execution. The findings of this research are beneficial for software companies and project teams seeking to improve task management, streamline collaboration, and optimize the use of version control tools like Gitea. Although the application development is not fully complete, the research successfully addresses questions regarding the design and implementation of the project management system, its integration with Gitea, and the application of microservice architecture, fulfilling most of the essential functional requirements and reaching the beta-release stage.

Riwayat
Diterima 21 Jan 2025
Revisi 20 Mar 2025
Disetujui 15 Apr 2025
Terbit online 30 Mei 2025

Kata Kunci
Aplikasi Web,
Microservice,
Pengendalian Versi,
Sistem Integrasi.

Abstrak

Manajemen proyek telah berevolusi dari metode tradisional ke pendekatan agile, yang mengurangi dokumentasi dan menggunakan kode sumber sebagai dokumen proyek untuk mencerminkan fitur dan alur sistem perangkat lunak. PT Gaws Inti Solusi, sebuah perusahaan perangkat lunak, menghadapi tantangan dalam menggunakan Gitea untuk version control dan distribusi pekerjaan karena antarmuka yang tidak memenuhi kebutuhan mereka. Penelitian ini mengembangkan sistem manajemen proyek berbasis web yang terintegrasi dengan Gitea dan menggunakan arsitektur microservice serta metodologi Spiral, untuk meningkatkan produktivitas dan kolaborasi dalam pelaksanaan proyek. Hasil penelitian ini bermanfaat bagi perusahaan perangkat lunak dan tim proyek yang ingin meningkatkan pengelolaan tugas, memperlancar kolaborasi, dan mengoptimalkan penggunaan alat version control seperti Gitea. Meskipun pengembangan aplikasi belum sepenuhnya selesai, penelitian ini berhasil menjawab pertanyaan mengenai rancangan dan implementasi sistem manajemen proyek, integrasinya dengan Gitea, serta penerapan arsitektur microservice, dan telah menyelesaikan sebagian besar kebutuhan fungsional penting hingga mencapai tahap *beta-release*.

PENDAHULUAN

Manajemen proyek berevolusi dari tradisional ke agile, mengurangi dokumentasi proyek dengan menjadikan kode sumber sebagai dokumen yang mencerminkan fitur dan alur sistem perangkat lunak (Bhavsar* et al., 2020). Seiring perkembangan teknologi, sistem perangkat lunak modern kini lebih banyak dibangun menggunakan teknologi berbasis web. Berbeda dengan situs web tradisional, sistem berbasis web modern sering kali memiliki banyak tingkat izin, menyimpan masukan pengguna dalam basis data, serta memungkinkan interaksi dan berbagi konten antar pengguna, menjadikannya lebih dinamis dan kolaboratif (Hoffman, 2020). Dalam pengembangan perangkat lunak dengan metodologi agile, microservices menjadi

populer karena memungkinkan skalabilitas dan perubahan komponen sistem secara independent (Cerny et al., 2023). Beberapa organisasi menghadapi tantangan dalam mengelola proyek dengan metode agile, seperti yang dialami PT Gaws Inti Solusi, termasuk penggunaan perangkat lunak version control dan distribusi beban kerja. Antarmuka Gitea dinilai tidak memenuhi kebutuhan manajemen proyek mereka, sehingga tidak digunakan untuk tujuan manajemen pekerjaan proyek perangkat lunak.

Faizah dkk telah membangun sistem manajemen proyek perangkat lunak dengan menerapkan Kanban Framework pada manajemennya, membuktikan bahwa terdapatnya sebuah sistem manajemen proyek perangkat lunak dapat membantu sebuah software house dalam melakukan pekerjaannya (Faizah et al., 2019). Implementasi sistem manajemen proyek dengan gaya Scrum telah terbukti berguna untuk PT. Debox Indonesia yang mengalami kesulitan dalam manajemen pekerjaan proyek perangkat lunak yang menggunakan Google Spreadsheet (Kharisma & Santoso, 2020). PT. Santai Berkualitas Syberindo mengembangkan sebuah sistem manajemen proyek perangkat lunak berbasis Web dan Mobile yang memenuhi kebutuhan perusahaan tersebut dalam pengarsipan proyek yang jelas (Shidqi & Ricky, 2021).

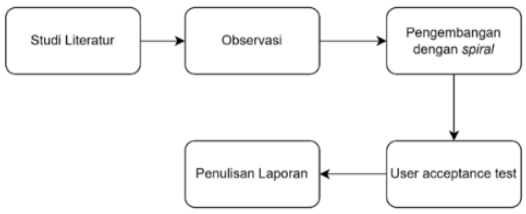
Sebuah studi di Belfast, Inggris, membuktikan bahwa penggunaan perangkat lunak manajemen proyek yang dikolaborasikan dengan penggunaan version control menjadi metode yang efektif dalam membangun sebuah proyek perangkat lunak (Sarhadi et al., 2022). Berdasarkan studi literatur penelitian ini, ketersediaan perangkat lunak yang terintegrasi dengan sebuah version control masih sangat minim yang membuat hal tersebut menjadi research gap dari penelitian ini.

Penelitian ini mengimplementasikan sistem manajemen proyek perangkat lunak berbasis web yang diintegrasikan dengan version control Gitea sesuai kebutuhan PT Gaws Inti Solusi, menggunakan arsitektur microservice dan metodologi Spiral, dengan harapan meningkatkan produktivitas dan kolaborasi dalam proyek, serta dibangun sebagai sistem baru dari awal.

Agile adalah pendekatan manajemen proyek yang mengutamakan fleksibilitas, kolaborasi, respon cepat terhadap perubahan, dan iterasi berkelanjutan, dengan prinsip utama individu dan interaksi, perangkat lunak yang berfungsi, kolaborasi pelanggan, serta respon terhadap perubahan (Layton et al., 2020). Domain-Driven Design (DDD) berfokus pada pemodelan domain bisnis dalam perangkat lunak melalui konsep seperti Domain, Ubiquitous Language, dan Bounded Context untuk mengelola kompleksitas (Khononov, 2021). Penelitian ini menerapkan DDD dan Microservices, di mana Bounded Context diterjemahkan menjadi microservices. Microservice Architecture (MSA) yang diadopsi oleh perusahaan besar seperti Netflix dan Amazon memungkinkan pembangunan layanan kecil yang fokus, otonom, dan sesuai prinsip Single Responsibility, mengurangi kompleksitas sistem monolitik (Newman, 2019; Giamattei et al., 2024).

Pengendalian versi adalah manajemen produk/desain yang melacak iterasi dan perubahan selama pengembangan untuk mendukung kolaborasi, tinjauan, penelusuran, dan pengendalian perubahan (Jones et al., 2021). Gitea, perangkat lunak version control, menyediakan manajemen repositori kode, eksplorasi perubahan, tinjauan, penggabungan kode, dan kolaborasi tim melalui antarmuka RESTful API dengan keamanan OAuth2.0 (Gitea, 2024). Integrasi sistem, seni menghubungkan solusi terpisah untuk memastikan interaksi perangkat keras, perangkat lunak, dan manusia, akan diterapkan pada aplikasi yang dibangun mengikuti protokol HTTP dan otorisasi OAuth2.0 (Grady, 2020). REST, gaya arsitektur layanan web, memiliki karakteristik Stateless, Client-Server, Uniform Interface, dan penggunaan Status Codes untuk komunikasi jaringan (Doglio, 2018).

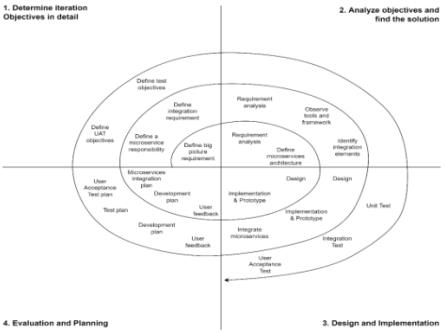
METODE PENELITIAN



Gambar 1. Metode Penelitian

Dalam aktivitasnya, studi literatur perlu dilakukan untuk memastikan keabsahan dari permasalahan yang diteliti pada penelitian ini, dilanjutkan dengan kegiatan pengembangan yang dilakukan secara iteratif, sesuai dengan siklus spiral yang akan disesuaikan dengan keadaan dan kebutuhan, dan terakhir adalah menulis laporan dari hasil perancangan dan pembuatan aplikasi sesuai dengan siklus yang dilakukan.

Model Spiral adalah pendekatan pengembangan perangkat lunak yang menggabungkan desain dan prototyping dalam tahapan iteratif, fokus pada evaluasi risiko dan segmentasi proyek kecil yang fleksibel. Diperkenalkan oleh Boehm, setiap loop spiral mencakup mengidentifikasi tujuan, analisis risiko, pelaksanaan fase pengembangan, dan perencanaan fase berikutnya, menggabungkan aspek model waterfall dan prototype, serta melibatkan ulasan internal atau eksternal untuk evaluasi dan penyesuaian (Akinsola et al., 2020).



Gambar 2. Siklus Spiral yang Dilakukan

Dalam sebuah iterasi, terdapat 4 buah kegiatan utama yang dilakukan, yaitu:

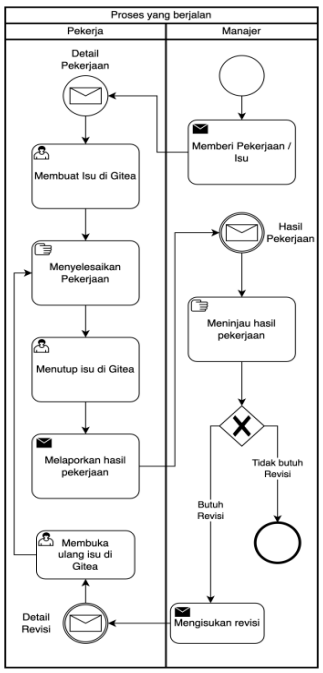
a. Menentukan objektif dari iterasi.
b. Analisa serta mencari solusi untuk meraih sasaran yang ditentukan.
c. Desain program serta implementasi program sesuai dengan analisa sebelumnya yang diakhiri dengan kegiatan uji coba.
d. Evaluasi hasil iterasi dengan mendapatkan umpan balik pengguna serta merencanakan kegiatan di siklus selanjutnya.

Perencanaan siklus spiral ini dibuat untuk menyesuaikan keadaan dan kebutuhan dari aplikasi memastikan bahwa hasil akhir memenuhi ekspektasi dan kebutuhan pengguna dengan optimal.yang akan dibangun, sehingga setiap tahapan dalam proses pengembangan dapat diadaptasi sesuai dengan perubahan atau persyaratan baru yang muncul.

Penelitian diawali dengan wawancara langsung untuk mengidentifikasi kebutuhan spesifik terkait manajemen proyek dan pengendalian versi. Selanjutnya, perancangan aplikasi dilakukan melalui analisis bahasa teknis yang digunakan *(ubiquitous language)*, kebutuhan fungsional, objek domain *(domain objects)*, dan konteks transaksi *(bounded context)*. Pengembangan aplikasi dilakukan secara iteratif dengan pendekatan prototyping untuk memperoleh masukan pengguna dan meminimalkan potensi miskomunikasi. Penelitian diakhiri dengan uji penerimaan aplikasi *(user acceptance test)* yang menurut (Fretheim & Deschene, 2023), adalah tahap pengujian perangkat lunak yang dilakukan oleh organisasi yang akan menggunakan perangkat lunak tersebut dalam produksi.

HASIL DAN PEMBAHASAN

Hasil Pengumpulan Kebutuhan



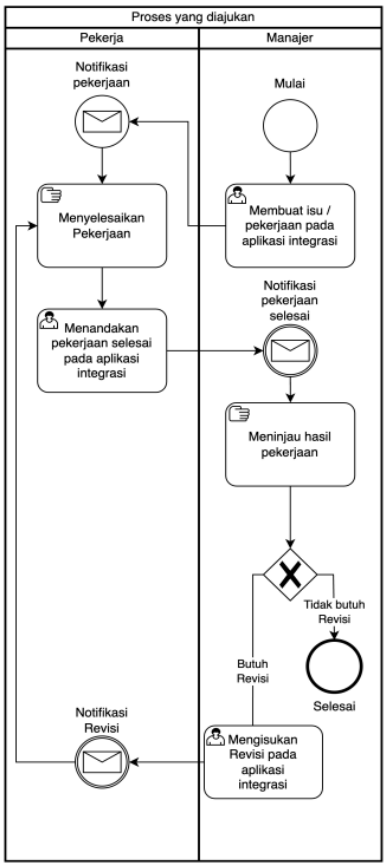
Gambar 3. Proses yang berjalan

Pada proses pengerjaan proyek, pekerja diharuskan membuat isu pada Gitea untuk memastikan bahwa segala perubahan dan pekerjaan pada sumber kode tercatat dengan baik pada Gitea. Pencatatan ini bertujuan untuk menjaga transparansi dan mempermudah pengelolaan pekerjaan, seperti pelacakan progres, diskusi terkait permasalahan, serta dokumentasi historis dari setiap perubahan yang dilakukan.

Tabel 1. *Ubiquitous Language*

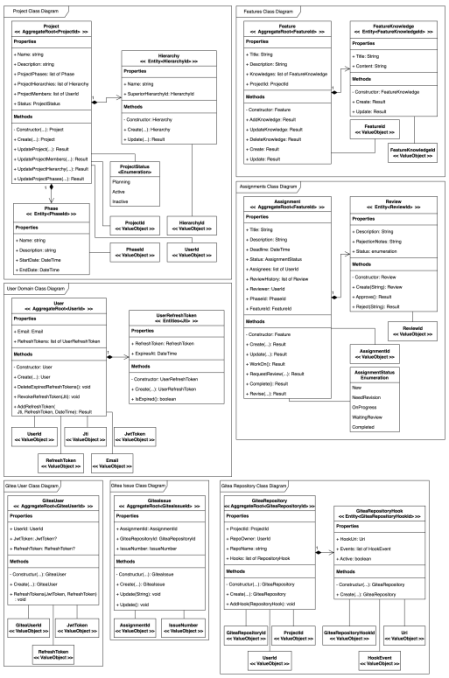
Kata Istilah	Penjelasan
Proyek *(Project)*	Adalah sebuah proyek perangkat lunak yang akan dikerjakan oleh perusahaan ini.
Fitur *(Feature)*	Pengerjaan dari sebuah proyek akan dibagi-bagi menjadi kumpulan fitur-fitur aplikasi proyek.
Pekerjaan *(Assignment)*	Pekerjaan dalam konteks ini adalah suatu hal yang harus dikerjakan untuk membangun sebuah fitur dalam proyek.
Hierarki / Jabatan *(Hierarchy)*	Hierarki proyek berarti posisi dari seseorang yang mengambil peran dalam sebuah proyek. Hierarki ini digunakan untuk hak pemberi pekerjaan untuk memberi pekerjaan kepada orang-orang yang memiliki hierarki dibawahnya.
Pemberi Pekerjaan *(Assignor)*	Adalah seseorang yang mendelegasikan pekerjaan.
Penerima Pekerjaan *(Assignee)*	Adalah seseorang yang diberikan pekerjaan.
Isu *(Issue)* Gitea	Merupakan data *issue* yang berasal dari *Gitea*.
Repository *(Repository)* Gitea	Merupakan sebuah *remote git repository* yang ada pada Gitea.
Kanban	Merujuk pada bahasa Jepang yang berarti sepotong informasi. Dalam konteks sistem ini 1 pekerjaan dapat dikatakan sebagai 1 kanban.
Kanban board	Sebuah kontainer yang berisikan 1 atau lebih *kanban*.

Hasil Rancangan



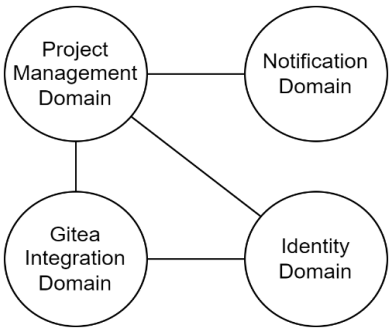
Gambar 4. Proses yang diajukan

Solusi yang diajukan mengeliminasi beberapa proses yang berjalan sehingga proses penugasan lebih efisien.



Gambar 5. *Domain Object Class Diagram*

Sebuah proyek perangkat lunak terdiri atas fase-fase proyek, hierarki/jabatan, dan daftar anggota pekerja yang berpartisipasi. Daftar fitur proyek berfungsi sebagai acuan pemberian tugas serta referensi informasi kebutuhan dan aturan terkait fitur tersebut. Setiap tugas pada fitur dialokasikan untuk dikerjakan oleh satu individu dan ditinjau oleh individu lainnya. Objek pengguna mencakup daftar refresh token untuk keperluan otentikasi. Objek-objek Gitea yang digunakan meliputi Gitea User, Gitea Issue, dan Gitea Repository, yang berperan sebagai data perantara antara aplikasi dan data Gitea..



Gambar 6. *Bounded Context*

Berdasarkan desain objek domain, didapatkan 4 business domain utama. Masing-masing *business domain* menyelesaikan kebutuhan-kebutuhan seperti berikut.

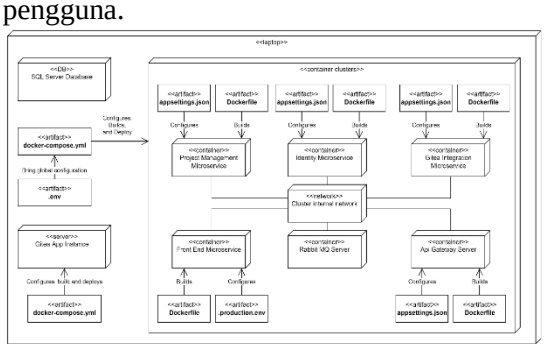
a. *Project Management Domain* menyelesaikan kebutuhan fungsional:
   - Kebutuhan Fungsional: Manajemen Data Proyek
   - Kebutuhan Fungsional: Manajemen Data Fitur dari Proyek
   - Kebutuhan Fungsional: Manajemen Data Pekerjaan Proyek
b. *Notification Domain* menyelesaikan kebutuhan fungsional:
   - Kebutuhan Fungsional: Notifikasi
c. *Gitea Integration Domain* menyelesaikan kebutuhan fungsional:
   - Kebutuhan Fungsional: Integrasi Gitea
d. *Identity Domain* menyelesaikan kebutuhan fungsional:
   - Kebutuhan Fungsional: Autentikasi dan Otorisasi

Project Management dan *Gitea Integration* menjadi *core subdomain*, *Identity Domain* merupakan *generic subdomain*, dan *Notification Domain* menjadi *supporting subdomain* dari solusi ini. Siklus-siklus selanjutnya berfokus terhadap *prototype* dari masing-masing *microservice* yang akan dibuat.

Masing-masing business domain yang telah dianalisa dapat dikembangkan menjadi beberapa microservice. Daftar dari microservice tersebut adalah:

a. *Project Management Microservice*: berisikan fungsi dari *Project Management Domain*.
b. *Gitea Integration Microservice*: Berisikan fungsi dari *Gitea Integration Domain*.
c. *Notification Microservice*: Berisikan fungsi dari *Notification Domain*.
d. *Identity Microservice*: Berisikan fungsi dari *User Management Domain*.
e. *Front End Microservice*: Berperan dalam menyediakan tampilan antarmuka pengguna.

Berdasarkan kepentingan bisnis dari masing-masing *microservice*, pengembangan sistem ini dapat diselesaikan secara bertahap. Pengembangan keseluruhan aplikasi ini dapat dibagi menjadi beberapa siklus dengan objektif dari masing-masing siklus adalah menyelesaikan sebuah *microservice*. *Front end Microservice* akan senantiasa dikembangan pada tiap siklus karena memiliki peran penting yaitu menyediakan antarmuka pengguna.



Gambar 7. *Deployment Diagram*

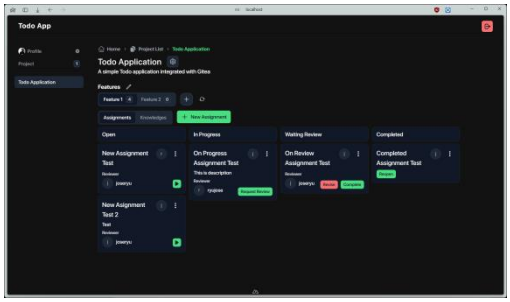
Semua node yang ada disini terinstalasi pada laptop peneliti. Berikut adalah penjelasan dari masing-masing komponen:

a. *Database SQL Server*: tempat penyimpanan data utama yang digunakan oleh berbagai microservice dalam sistem.
b. *Gitea App Instance*: instansi dari aplikasi Gitea yang dikonfigurasi, dibangun, dan di-*deploy* menggunakan file docker-compose.yml.
c. *File* .env dan docker-compose.yml:
   1) *File* .env berisi konfigurasi global untuk seluruh sistem.
   2) *File* docker-compose.yml digunakan

untuk mengkonfigurasi, membangun, dan mendeploy semua microservice dan komponen yang ada.

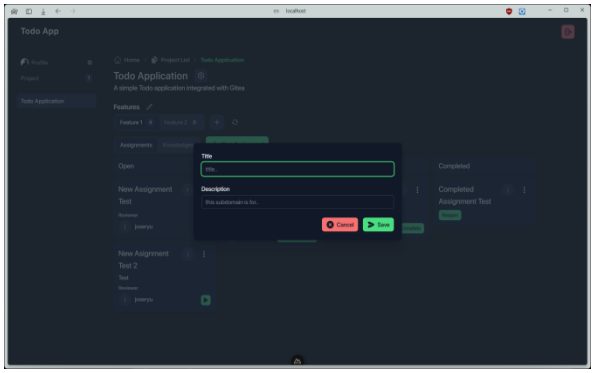
d. *Container Clusters*: kumpulan *container* yang menjalankan berbagai *microservice*. Setiap *microservice* dikonfigurasi dan dibangun menggunakan berkas-berkas yang relevan seperti *Dockerfile* dan appsettings.json.
e. *Project Management Microservice*: di-*build* menggunakan Dockerfile dan dikonfigurasikan oleh appsettings.json.
f. *Identity Microservice*: di-*build* menggunakan *Dockerfile* dan dikonfigurasikan oleh appsettings.json.
g. *Gitea Integration Microservice*: di-*build* menggunakan *Dockerfile* dan dikonfigurasikan oleh appsettings.json.
h. *Front End Microservice*: di-*build* menggunakan *Dockerfile* dan dikonfigurasikan oleh appsettings.json.
i. *Api Gateway Server*: di-*build* menggunakan *Dockerfile* dan dikonfigurasikan oleh appsettings.json.
j. *Rabbit MQ Server*: *container server* yang mengelola antrian pesan untuk komunikasi antar *microservice*.
k. *Cluster internal network*: semua *container* berkomunikasi melalui jaringan internal *cluster* ini.

Antarmuka Sistem



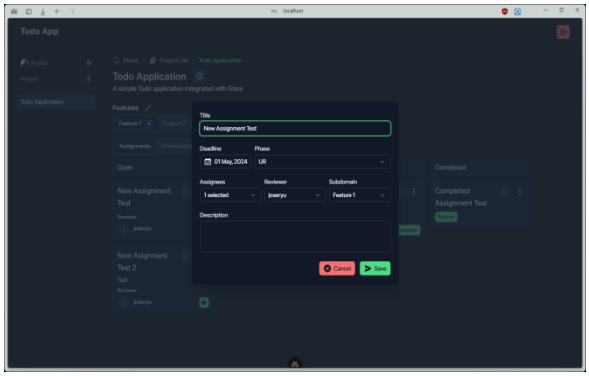
Gambar 8. Layar Halaman Assignments

Pada layar ini, terdapat daftar fitur yang ditampilkan dalam bentuk tabs supaya pengguna dapat memilih untuk melihat daftar pekerjaan atau pengetahuan fitur apa saja. Terdapat tombol ubah untuk menampilkan tombol hapus dan ubah fitur, tombol tambah fitur di sebelah tabs dari fitur, serta tombol untuk memuat ulang data yang ditampilkan.



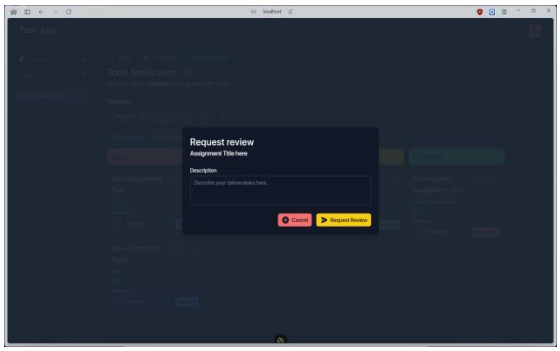
Gambar 9. Formulir Fitur

Ketika tombol tambah atau tombol ubah fitur ditekan, maka sistem akan menampilkan formulir fitur. Pengguna dapat memasukkan nilai sesuai dengan fitur yang akan diubah atau ditambahkan.



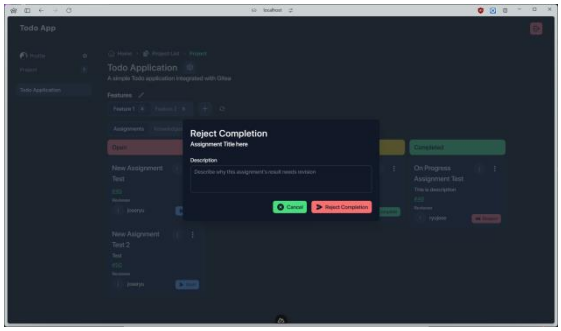
Gambar 10. Formulir Pekerjaan

Ketika tombol tambah atau ubah pekerjaan ditekan, maka sistem menampilkan formulir pekerjaan. Pengguna harus memasukkan nilai yang sesuai pada kolom input masing-masing.



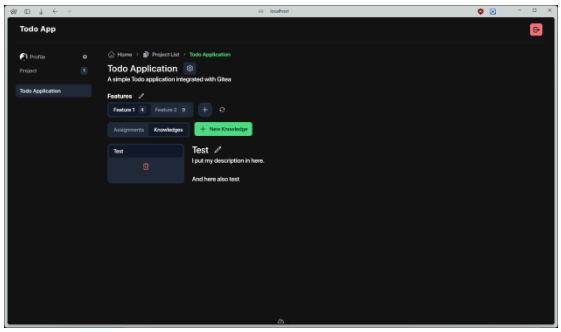
Gambar 11. Formulir Permintaan Peninjauan

Ketika sebuah pekerjaan ingin ditinjau, pekerja harus mengisi deskripsi hasil pekerjaan yang ia kerjakan, disertakan bagaimana reviewer dapat melihat hasilnya.



Gambar 12. Formulir Revisi Pekerjaan

Reviewer harus memasukkan alasan dibalik revisi hasil pekerjaan yang telah dikirimkan oleh pekerja.



Gambar 13. Halaman Feature Knowledge

Fitur ditampilkan dalam bentuk *vertical tabs*. Terdapat tombol tambah fitur di sebelah kanannya tabs pekerjaan dan pengetahuan, serta tombol edit pengetahuan di sebelah kanan judul pengetahuan.

Hasil Uji Coba

Metode UAT yang digunakan dalam penelitian ini adalah Contract Acceptance Testing (CAT). Menurut Mathew (2022), CAT merupakan jenis pengujian yang bertujuan memastikan bahwa perangkat lunak atau sistem yang dikembangkan telah memenuhi seluruh persyaratan dan spesifikasi yang tertuang dalam kontrak antara pengembang dan klien atau pengguna akhir.

Berikut adalah kesimpulan dari hasi uji coba yang dilakukan.

Tabel 2. Jumlah kasus uji

Bounded Context	Jumlah kasus Uji	Jumlah diterima	Jumlah tidak diterima
Manajemen Proyek	29	29	0
Integrasi Gitea	19	19	0

Berdasarkan hasil uji coba, pengguna menyatakan bahwa aplikasi ini sudah bisa digunakan oleh mereka untuk melakukan pekerjaan mereka. Mereka menyatakan bahwa aplikasi ini sudah siap masa *beta release*. UAT dilaksanakan untuk 70% kebutuhan yang telah didefinisikan, sedangkan sisa 30% akan dilaksanakan diluar kegiatan penelitian ini.

KESIMPULAN

Meskipun pengembangan aplikasi ini belum selesai sepenuhnya sesuai dengan rencana awal, penelitian ini berhasil mengimplementasikan rancangan dan implementasi integrasi aplikasi dengan Gitea yang menerapkan arsitektur microservice. Selain itu, aplikasi ini telah menyelesaikan sebagian besar kebutuhan fungsional yang penting, memungkinkan aplikasi ini untuk mencapai tahap beta-release, meskipun tahap tersebut tidak dilaksanakan dan tidak dilaporkan dalam penelitian ini. Dalam pengembangan integrasi selanjutnya, terdapat permasalahan yang perlu diselesaikan terkait bisnis dari manajemen proyek serta integrasinya kepada Gitea, yaitu meluaskan fitur integrasi antara aplikasi ini dengan aplikasi Gitea, buat integrasi aplikasi ini dengan GitHub, investigasi keperluan-keperluan manajemen proyek dari banyak pihak untuk meng-generalisirkan aplikasi ini dapat digunakan di berbagai tempat.

REFERENSI

Akinsola, J. E. T., Ogunbanwo, A. S., Okesola, O. J., Odun-Ayo, I. J., Ayegbusi, F. D., & Adebiyi, A. A. (2020). Comparative Analysis of Software Development Life Cycle Models (SDLC). In R. Silhavy (Ed.), *Intelligent Algorithms in Software Engineering* (pp. 310–322). Springer International Publishing.

Bhavsar*, K., Shah, D. V., & Gopalan, D. S. (2020). Scrum: An Agile Process Reengineering In Software Engineering. *International Journal of Innovative Technology and Exploring Engineering*, *9*(3), 840–848. https://doi.org/10.35940/ijitee.c8545.019320

Cerny, T., Abdelfattah, A. S., Maruf, A. Al, Janes, A., & Taibi, D. (2023). Catalog and detection techniques of microservice

anti-patterns and bad smells: A tertiary study. *Journal of Systems and Software*, *206*, 111829. https://doi.org/10.1016/j.jss.2023.111829

Faizah, N., Santoso, N., & Soebroto, A. A. (2019). Pengembangan Sistem Aplikasi Manajemen Proyek menggunakan Kanban Framework. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(10), 9747–9754. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6533

Fretheim, E., & Deschene, M. (2023). *Secure Software Systems*. Jones \& Bartlett Learning. https://books.google.co.id/books?id=FmqwEAAAQBAJ

Giamattei, L., Guerriero, A., Pietrantuono, R., & Russo, S. (2024). Automated functional and robustness testing of microservice architectures. *Journal of Systems and Software*, *207*(October 2023), 111857. https://doi.org/10.1016/j.jss.2023.111857

Gitea - Lightweight DevOps Platform. (2024). https://about.gitea.com/

Grady, J. O. (2020). *System Integration*. CRC Press. https://books.google.co.id/books?id=QfzyDwAAQBAJ

Hoffman, A. (2020). *Web Application Security: Exploitation and Countermeasures for Modern Web Applications*. O'Reilly Media. https://books.google.co.id/books?id=3R3UDwAAQBAJ

Jones, D., Nassehi, A., Snider, C., Gopsill, J., Rosso, P., Real, R., Goudswaard, M., & Hicks, B. (2021). Towards integrated version control of virtual and physical artefacts in new product development: Inspirations from software engineering and the digital twin paradigm. *Procedia CIRP*, *100*(March), 283–288. https://doi.org/10.1016/j.procir.2021.05.121

Kharisma, B., & Santoso, N. (2020). Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Kolaboratif Menggunakan Scrum. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, *4*(3), 723–732.

Khononov, V. (2021). *Learning Domain-Driven Design*. O'Reilly Media. https://books.google.co.id/books?id=qAtHEAAAQBAJ

Layton, M. C., Ostermiller, S. J., & Kynaston, D. J. (2020). *Agile Project Management For Dummies*. Wiley. https://books.google.co.id/books?id=gSH7DwAAQBAJ

Mathew, J. (2022). *Learn Manual Software Testing through Interview Questions*. Jimmy mathew. https://books.google.co.id/books?id=TJSdEAAAQBAJ

Newman, S. (2019). *Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith*. O'Reilly Media. https://books.google.co.id/books?id=ota_DwAAQBAJ

Sarhadi, P., Naeem, W., Fraser, K., & Wilson, D. (2022). On the Application of Agile Project Management Techniques, V-Model and Recent Software Tools in Postgraduate Theses Supervision. *IFAC-PapersOnLine*, *55*(17), 109–114. https://doi.org/10.1016/j.ifacol.2022.09.233

Shidqi, M., & Ricky, M. A. (2021). Pengembangan Aplikasi Dan Website Manajemen Proyek Pt Santai Berkualitas Syberindo Menggunakan Metode Agile. *Seminastika*, *3*(1), 8–15. https://doi.org/10.47002/seminastika.v3i1.249