Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 64-73
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

IMPLEMENTASI *FRAMEWORK SCRUM* DALAM PENGEMBANGAN *DASHBOARD MONITORING* UNTUK OPTIMASI PENGELOLAAN *DATA INTERFACE*

Dwi Diana Wazaumi[1*], Vian Ardiyansyah Saputro[2], Seftia Khairun Nisa[3] , Sharla Adhita Zahrani[4]

[1,2,3,4]Manajemen Informatika, Jurusan Informatika, Politeknik Astra, Cikarang, Indonesia

Email: [1]dwi.diana@polytechnic.astra.ac.id, [2]vian.saputro@polytechnic.astra.ac.id, [3]0320210057@polman.astra.ac.id, [4]0320210072@polman.astra.ac.id
(* : corresponding author)

Abstrak-Di zaman digital sekarang ini, data yang dikelola secara efisien serta efektif merupakan suatu keberhasilan sebuah perusahaan ketika menghadapi permasalahan dalam bisnis yang selalu berubah. PT XYZ adalah satu dari sekian banyak perusahaan di sektor otomotif, PT XYZ perlu memastikan bahwa data antarmuka mereka dapat dikelola secara maksimal. Namun, pengelolaan data interface sering dihadapkan pada berbagai tantangan, seperti volume data dengan jumlah yang banyak, pemantauan yang tiada henti, dan kebutuhan untuk mengidentifikasi lebih awal permasalahan yang akan muncul. Dalam rangka mengatasi permasahalan tersebut, PT XYZ memiliki rencana untuk merancang sebuah aplikasi dengan nama MyZ sebagai solusi pemantauan melalui *dashboard* yang bertujuan untuk melakukan pemantauan file dan menemukan potensi masalah terkait data antarmuka. Fokus dari penelitian ini adalah pada pengembangan aplikasi MyZ sebagai *dashboard* pemantauan dan mengadopsi pendekatan Scrum untuk mengoptimalkan pengelolaan antarmuka data di PT XYZ. Temuan yang didapatkan dari pengembangan aplikasi MyZ, aplikasi ini dapat menunjukan waktu yang dibutuhkan untuk pengiriman data melalui email rata-rata 3,3 detik hal ini mengindikasikan bahwa aplikasi MyZ memungkinkan pengguna untuk menerima informasi secara cepat, sehingga dapat mempercepat proses pengambilan keputusan. Sedangkan penyajian informasi di dalam dasboard membutuhkan waktu rata-rata 1,5 detik. Hal ini menunjukkan bahwa aplikasi MyZ dapat memberikan respon yang cepat dan efisien dalam mendukung proses pemantauan data interface serta meminimalkan waktu tunggu. Efisiensi ini sangat relevan dalam lingkungan kerja yang dinamis dan membutuhkan respon cepat terhadap perubahan data.

Kata Kunci: myz**,** *dashboard monitoring*, *real-time monitoring*, metodologi *scrum*, *agile*

Abstract-*In today's digital era, data that is managed efficiently and effectively is a success for a company when facing problems in an ever-changing business. PT XYZ is one of the many companies in the automotive sector, PT XYZ needs to ensure that their interface data can be managed optimally. However, interface data management is often faced with various challenges, such as large volumes of data, continuous monitoring, and the need to identify problems that will arise early. In order to overcome these problems, PT XYZ has a plan to design an application called MyZ as a monitoring solution through a dashboard that aims to monitor files and find potential problems related to interface data. The focus of this study is on the development of the MyZ application as a monitoring dashboard and adopting the Scrum approach to optimize data interface management at PT XYZ. The findings obtained from the development of the MyZ application, this application can show the time needed to send data via email an average of 3.3 seconds, this indicates that the MyZ application allows users to receive information quickly, so that it can speed up the decision-making process. While the presentation of information in the dashboard takes an average of 1.5 seconds. This shows that the MyZ application can provide a fast and efficient response in supporting the interface data monitoring process and minimizing waiting time. This efficiency is very relevant in a dynamic work environment and requires a fast response to data changes.*

Keywords*: myz,dashboard monitoring, real-time monitoring, scrum methodology, agile*

1. PENDAHULUAN

Di zaman digital sekarang ini, data yang dikelola secara efisien serta efektif merupakan suatu keberhasilan sebuah perusahaan ketika menghadapi permasalahan dalam bisnis yang selalu berubah. PT XYZ adalah satu dari sekian banyak perusahaan di sektor otomotif, PT XYZ perlu memastikan bahwa data antarmuka mereka dapat dikelola secara maksimal. Kurangnya pengelolaan data antarmuka yang baik dapat menyebabkan berbagai masalah, mulai dari keterlambatan akses, ketidakakuratan informasi, sampai hilangnya kepercayaan dari pengguna [1]. Namun, pengelolaan data *interface* sering dihadapkan pada berbagai tantangan, seperti *volume* data dengan jumlah yang banyak, pemantauan yang tiada henti, dan kebutuhan untuk mengidentifikasi lebih awal permasalahan yang akan muncul. Dengan demikian, diperlukan sebuah sistem pemantauan dalam bentuk *dashboard* yang dapat memberikan data dan informasi data secara langsung. Selain sebagai alat pemantauan [2], juga sebagai alat bantu mengambil kebijakan[3], serta mengurangi kemungkinan terjadinya kesalahan [4] dalam proses manajemen dan optimasi data *interface*.

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 64-73
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

Dalam rangka mengatasi permasahalan tersebut, PT XYZ memiliki rencana untuk merancang sebuah aplikasi dengan nama MyZ sebagai solusi pemantauan melalui *dashboard* yang bertujuan untuk melakukan pemantauan *file* dan menemukan potensi masalah terkait data antarmuka. Aplikasi ini dirancang untuk menyajikan visualisasi data yang mudah dipahami, memperbaiki pemantauan secara efektif serta mempermudah pengambilan kebijakan menggunakan data.

Dalam memastikan pengembangan aplikasi MyZ selesai tepat waktu, adopsi *framework Scrum* menjadi pilihan dalam pengembangan aplikasi ini. *Scrum*, yang merupakan bagian dari metode *Agile*, memiliki kelebihan untuk memberikan perubahan yang dinamis sesuai dengan kebutuhan [5], meningkatkan kolaborasi tim [6], serta melalui iterasi dapat menghadirkan produk yang memiliki kualitas tinggi [7]. Dengan penggunaan metode *Scrum*, tim *developer* diharapkan mampu beradaptasi dengan lebih baik, dan menyelesaikan berbagai fitur utama aplikasi MyZ yang menjadi prioritas tepat sesuai jadwal.

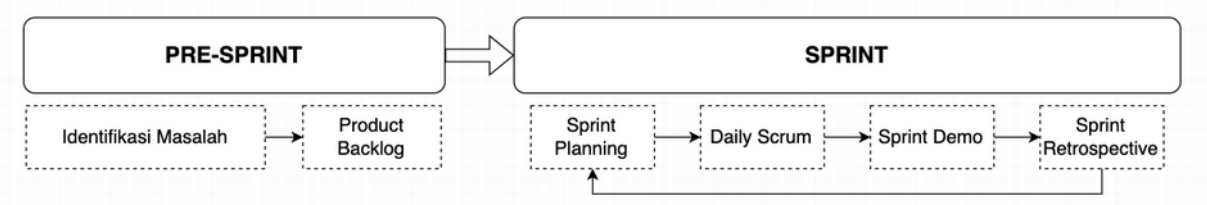
Studi terdahulu mengungkapkan bahwa pendekatan *Scrum* dalam perancangan *dashboard* monitoring untuk pengelolaan proyek, menunjukkan bahwa penerapan metode ini mendukung kerangka kerja yang memungkinkan proses berlangsung secara berulang dan responsif terhadap perubahan kebutuhan. Temuan penelitian itu menunjukkan bahwa pendekatan *Scrum* dapat meningkatkan efisiensi saat melakukan pengembangan fitur-fitur utama secara sistematis dan bertahap [8] serupa dengan penelitian [9] yang menggunakan metode *Scrum* dalam pengembangan sistem informasi layanan kawasan guna meningkatkan efisiensi dan memastikan tim bekerja secara fokus dan produktif. Penelitian lain oleh [10] menunjukkan bahwa adopsi metode *Scrum* untuk perancangan sistem informasi manajemen aset terpadu dapat menjadi jawaban dari berbagai masalah yang timbul selama proses pengembangan, sehingga waktu pengerjaan sistem informasi menjadi lebih efisien. Temuan dari studi yang telah disampaikan memberikan masukan bagi kami dalam pengembangan aplikasi MyZ dengan mengadopsi *framework Scrum*, sebagai harapan dapat menangani tantangan pengelolaan data antarmuka dengan lebih optimal.

Fokus dari penelitian ini adalah pada pengembangan aplikasi MyZ sebagai *dashboard* pemantauan dan mengadopsi pendekatan Scrum untuk mengoptimalkan pengelolaan antarmuka data di PT XYZ. Oleh karena itu, penelitian ini memiliki peran penting dalam mendukung pengelolaan data antarmuka secara optimal di PT XYZ, tetapi juga menyajikan ilmu pengetahuan baru tentang adopsi *framework Scrum* dalam proses pengembangan aplikasi *dashboard* yang kompleks dan fleksibel.

2. METODE PENELITIAN

Penelitian ini menggunakan metodologi *agile* dengan menerapkan framework *Scrum*. Pendekatan ini dipilih karena sesuai dengan kebutuhan pengembangan sistem yang bersifat dinamis dan membutuhkan kolaborasi lintas fungsi [11][12]. PT XYZ merupakan Perusahaan pengembang produk perangkat lunak yang proses pengembangannya diperbaharui secara berkelanjutan. Scrum merupakan kerangka yang bertujuan melakukan pengembangan secara cepat dan masih bisa berubah-rubah seiring berjalannya waktu, disesuaikan dengan hasil evaluasi tiap iterasinya, sehingga scrum dirasa akan cocok dalam penelitian ini. Dalam *framework Scrum*, terdapat tiga peran utama yang sangat penting dalam memastikan proses *sprint* berjalan dengan baik, yaitu:

1. Pemilik Produk
   Pemilik produk adalah penghubung utama antara mitra dan tim pengembang, bertugas untuk memvalidasi visi dan tujuan produk tersampaikan dengan jelas [13].
2. Tim Pengembang
   Tim Pengembang bekerja secara kolaboratif untuk merancang, mengembangkan, dan menguji aplikasi hingga menjadi produk akhir [14].
3. Scrum *Master* (SM)
   Peran strategis *Scrum Master*, seperti mengelola ekspektasi peran, menangani resistensi perubahan, menjaga jadwal, mengelola permintaan perubahan mendesak, dan mendistribusikan tim, menjadi dasar penting dalam proses pemilihan yang harus mempertimbangkan kualifikasi relevan [15].



Gambar 1. Tahapan-Tahapan Kerangka *Scrum* [16]

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 64-73
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

Ilustrasi tahapan *Scrum* dapat dilihat pada Gambar 1, yang menggambarkan dua tahapan utama, yaitu Pre-*Sprint* dan *Sprint*. Pada tahap *Pre-Sprint*, aktor yang terlibat meliputi *Product Owner* dalam hal ini merupakan perusahaan mitra, *Scrum Master* dalam penelitian ini merupan pengguna atau mentor selama pengembangan, dan tim pengembang untuk merencanakan *backlog* produk dan mendefinisikan tujuan *Sprint*. Sedangkan pada tahap *Sprint*, tim bekerja bersama untuk menyelesaikan item-item *backlog* yang telah ditentukan, dengan fokus pada pengembangan produk yang dapat diuji dan siap untuk diperlihatkan pada akhir *Sprint*. *Scrum Master* memastikan kelancaran proses, sementara *Product Owner* mengelola prioritas dan kebutuhan yang berubah-ubah selama *Sprint* berlangsung.

2.1. Pre-*sprint*

1. Identifikasi Masalah

Tahap ini dimulai dengan diskusi terstruktur bersama tim terkait di PT XYZ, yang melibatkan manajer operasional, tim IT *backend*, tim IT *frontend*, dan tim administrasi. Diskusi ini bertujuan untuk memahami alur pengiriman file yang sedang berjalan, mengidentifikasi kendala yang sering terjadi, serta merumuskan kebutuhan fitur baru yang diperlukan untuk meningkatkan efisiensi proses. Beberapa aspek utama yang didalami meliputi hambatan dalam proses pengiriman *file*, fitur yang dapat mendukung efisiensi, dan preferensi pengguna terkait mekanisme pelaporan.

Selain itu, wawancara dengan pihak-pihak relevan dilakukan untuk menggali tantangan teknis dan operasional secara lebih rinci. Proses ini juga dilengkapi dengan analisis dokumen operasional untuk memahami alur kerja yang ada dan mengidentifikasi kebutuhan spesifik yang harus diakomodasi oleh sistem baru. Seluruh langkah ini dilakukan dengan memperhatikan pendekatan kolaboratif dan iteratif yang sesuai dengan prinsip *Scrum*, yaitu melibatkan pemangku kepentingan utama dalam proses identifikasi kebutuhan.

2. *Product Backlog*

Setelah masalah dan kebutuhan diidentifikasi, *backlog* produk disusun sebagai daftar fitur atau pekerjaan yang harus diselesaikan. Penentuan *backlog* dilakukan dengan mengacu pada hasil analisis kebutuhan pengguna dan diuraikan dalam bentuk item prioritas sesuai tingkat urgensi dan dampak terhadap efisiensi sistem. Setiap item dalam *backlog* dirancang untuk memberikan nilai tambah pada setiap iterasi, sesuai dengan kerangka kerja *Scrum* yang berfokus pada penciptaan nilai secara bertahap.

Selain itu, *backlog* produk diperbarui secara berkala berdasarkan umpan balik dari pengguna atau pemangku kepentingan, sejalan dengan pendekatan *Scrum* yang mengakomodasi perubahan kebutuhan. Pada tahap ini, tim *Scrum* juga menentukan durasi *sprint* yang optimal untuk menyelesaikan item-item dalam *backlog* tersebut, memastikan bahwa *backlog* tetap dapat dieksekusi dalam kerangka iterasi yang singkat dan terukur.

2.2. *Sprint*

1. *Sprint Planning*

   Pada tahap ini, tim merencanakan pekerjaan yang akan diselesaikan dalam satu periode *sprint*. Tim pengembang berdiskusi dan menentukan item *Backlog* berdasarkan prioritas dan nilai tambahnya bagi produk. Selain itu, estimasi waktu untuk setiap item *backlog* ditentukan dengan mempertimbangkan kompleksitas dan kapasitas tim. Hasil dari *Sprint Planning* adalah daftar pekerjaan yang jelas dan dapat dicapai dalam *sprint*, dengan tujuan yang telah disepakati bersama (*Sprint Goal*).
2. *Daily Scrum*

   Daily Scrum adalah pertemuan diskusi mengenai hal-hal yang sedang, telah dilakukan oleh tim pengembang serta dilakukan secara rutin pada pagi atau siang hari sesuai jadwal yang sudah ditentukan, biasanya berlangsung selama 15 menit dan diikuti oleh seluruh anggota pengembang dan *scrum master*. Tujuan utama dari pertemuan ini adalah untuk menjaga tim tetap fokus pada tujuan, mengidentifikasi hambatan yang ada, dan memperbarui rencana kerja harian. Dalam rapat ini, setiap anggota tim umumnya menjawab tiga pertanyaan inti:
   1. Tugas apa yang berhasil saya selesaikan sejak pertemuan harian terakhir?
   2. Rencana apa yang akan saya kerjakan hari ini?
   3. Apakah ada halangan yang menghambat kemajuan pekerjaan saya?
3. *Sprint Demo*

   Tahap *Sprint Demo*, sering disebut juga sebagai *Sprint Demo*, dilakukan di akhir *sprint*. Pada tahap ini, *Scrum Team* dan *stakeholders* berkumpul untuk meninjau hasil kerja selama *sprint*, yaitu *Increment* (fitur atau produk yang selesai dan dapat digunakan). Tim pengembang mendemonstrasikan fitur yang telah selesai untuk mendapatkan umpan balik dari *Product Owner* dan *stakeholder*. Tahap ini bertujuan untuk memastikan bahwa *Increment* memenuhi kebutuhan bisnis dan memungkinkan *backlog* diperbarui sesuai dengan perubahan kebutuhan atau prioritas baru.

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 64-73
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

4. *Sprint Retrospective*
   Tahap *Sprint Retrospective* dilakukan setelah *Sprint Review* dan bertujuan untuk mengevaluasi proses kerja selama *sprint*.

2.3. Pengujian Aplikasi

Pada tahap pengujian akhir, dilakukan dua jenis pengujian yang berbeda, yaitu pengujian kecepatan respon dan *blackbox* testing. Pengujian kecepatan respon dilakukan di dua peramban utama, yaitu *Chrome* dan *Microsoft Edge*, untuk memastikan aplikasi dapat memberikan respon yang cepat dan konsisten di berbagai *platform*. Sementara itu, pengujian *blackbox testing* digunakan untuk menilai fungsionalitas aplikasi berdasarkan *input* dan *output* yang diharapkan, tanpa memerlukan pengetahuan tentang struktur internal aplikasi. Setiap siklus pengembangan dalam framework *Scrum* berfokus pada perbaikan berkelanjutan dan pengoptimalan performa aplikasi, termasuk memastikan bahwa kecepatan respon tetap optimal meskipun dalam kondisi beban tinggi dan volume data besar. Hasil pengujian ini bertujuan untuk mengevaluasi apakah aplikasi dapat memenuhi standar performa yang diharapkan oleh pengguna dan memberikan pengalaman yang efisien dan responsif.

3. HASIL DAN PEMBAHASAN

3.1. *Pre-Sprint*

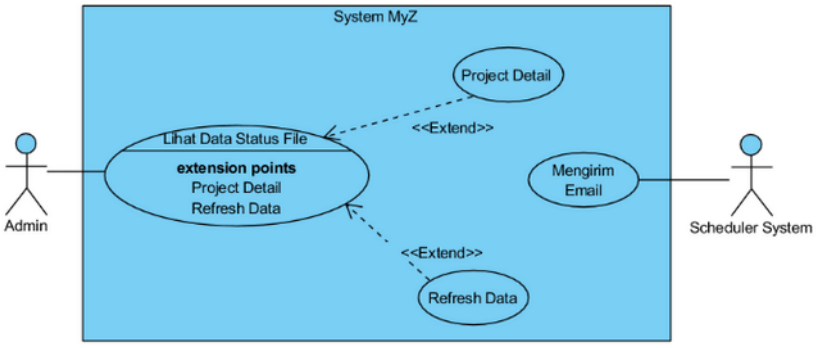
Pada tahapan *Pre-Sprint*, tim PT XYZ mengadakan diskusi intensif dengan berbagai pihak terkait, yaitu tim IT, tim operasional, dan *stakeholder* yang terlibat dalam proses pengiriman file. Diskusi dimulai dengan identifikasi masalah yang ada pada sistem lama, di mana tim menanyakan berbagai aspek terkait pengelolaan status pengiriman file. Beberapa pertanyaan yang diajukan meliputi:

a. Apa saja kendala yang dihadapi dalam memantau status pengiriman file?
b. Bagaimana proses pemantauan dan pengelolaan status pengiriman file saat ini dilakukan?
c. Apa yang menjadi hambatan dalam penggunaan metode manual seperti log dan query database?
d. Apa fitur yang dibutuhkan agar proses tersebut bisa dilakukan secara otomatis dan efisien?

Melalui diskusi ini, tim PT XYZ berhasil mengumpulkan informasi terkait kebutuhan dan harapan pengguna, serta mengidentifikasi bahwa sistem yang ada saat ini tidak memadai untuk memantau dan mengelola status pengiriman file secara otomatis. Sebagai hasilnya, tim memutuskan untuk mengembangkan sistem baru, MyZ, yang dapat mengotomatisasi proses pemantauan dan pengelolaan status pengiriman *file* guna meningkatkan efisiensi.

Tujuan dari pengembangan sistem MyZ adalah untuk menciptakan solusi yang memungkinkan pemantauan status pengiriman *file* secara lebih cepat dan efisien. Visi dari pengembangan sistem ini adalah untuk menyediakan *dashboard monitoring* yang memungkinkan pengguna untuk melihat status pengiriman data secara *real-time*, serta mengirimkan informasi kepada klien secara otomatis mengenai hasil pengiriman data. Identifikasi masalah yang ditemukan adalah kurangnya kemampuan sistem yang ada untuk memberikan laporan dan pemantauan yang cepat, yang mengarah pada kesulitan dalam mendeteksi masalah pengiriman data.

Fitur utama yang diusulkan adalah pembuatan *dashboard monitoring*, yang memungkinkan pengguna untuk melihat status proses pengiriman file yang terbagi menjadi tiga kategori: sukses, gagal, dan tidak ada pengiriman data. Dashboard ini juga menyediakan tombol "*Refresh*" untuk memperbarui data secara otomatis, memastikan bahwa pengguna selalu melihat informasi terbaru. Selain itu, sistem ini akan dilengkapi dengan fitur pengiriman *email*, yang menggunakan *scheduler* untuk mengirimkan informasi terkait isu yang terdeteksi kepada pengguna atau klien yang telah ditentukan secara berkala.



Gambar 2. *Use Case Diagram Dashboard Monitoring*

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 64-73
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

Kemudian, analisis kebutuhan lanjutan dilakukan dengan menggunakan *Use Case Diagram*. Gambar 2 menggambarkan interaksi antara aktor dan sistem yang telah disepakati berdasarkan hasil wawancara, pertemuan, serta ide, visi, dan tujuan yang dikumpulkan dari berbagai pihak. Berdasarkan Gambar 2, *Use Case Diagram* menunjukkan aktor utama, yaitu Admin, yang memiliki akses untuk melihat data status file yang terbagi dalam tiga kategori: sukses, gagal, dan tidak ada pengiriman data. Admin juga dapat mengeklik setiap *file* untuk melihat detail lebih lanjut mengenai proyek terkait melalui *extension point Project Detail*. Selain itu, Admin dapat memperbarui data secara manual dengan menggunakan tombol "*Refresh*" untuk memuat ulang informasi terbaru mengenai pengiriman file. Selanjutnya, terdapat sistem *Scheduler*, yang akan mengirimkan email secara otomatis kepada klien atau pengguna terkait pembaruan status pengiriman file atau masalah yang terdeteksi. Email ini akan dikirimkan secara berkala sesuai dengan jadwal yang telah ditentukan, memastikan bahwa informasi selalu terkirim tepat waktu.

Berdasarkan hasil observasi wawancara dan analisis, tim telah menyusun *Product Backlog* yang mencakup dua item utama, yaitu: perancangan *mockup* antarmuka, membuat *front end*, membuat *backend*. Setelah menetapkan tujuan proyek dan menyusun *Product Backlog*, tim Scrum melanjutkan dengan menentukan prioritas *backlog* berdasarkan tingkat urgensi dan kompleksitas pengembangan. Penentuan prioritas ini penting untuk memastikan bahwa setiap item dalam *backlog* diselesaikan sesuai dengan skala prioritas yang telah ditetapkan. Melalui perangkat lunak *Hansoft Project Scrum Management* 9, tim *Scrum* menentukan durasi *sprint* berdasarkan tingkat prioritas sebagai berikut:

1. Prioritas Sangat Tinggi: 12-14 hari
2. Prioritas Tinggi: 10-12 hari
3. Prioritas Sedang: 7-9 hari
4. Prioritas Rendah: 4-6 hari.

Tabel 1 menggambarkan daftar *Product Backlog* yang telah dikelompokkan berdasarkan prioritas ini, yang mencakup tugas-tugas seperti perancangan mockup antarmuka, pembuatan *frontend*, serta pengembangan *backend* dengan fitur pengiriman email dan penyimpanan data menggunakan *scheduler*.

Tabel 1. Daftar *Product Backlog Dashboard MyZ*

Backlog	Prioritas	Spesifikasi
Membuat *mockup*	Priotritas Tinggi	Antarmuka *dashboard* yang berisi *card success, fail, delivery,* serta filter berdasarkan grup pengiriman
Membuat *frontend*	Prioritas Sangat Tinggi	Menyesuaikan *sprint* 1
Membuat *backend*	Prioritas Sangat Tinggi	Send email, menyimpan data email menggunakan *scheduler*, *real time data update* pada *card success, fail, delivery*

3.2. *Sprint*

3.2.1. *Sprint* 1

Sebelum melaksanakan pengembangan aplikasi, tim melakukan *Sprint Planning* untuk merencanakan dan mendefinisikan tugas yang akan diselesaikan dalam setiap *sprint*. Pada *Sprint* 1, fokus utama adalah perancangan mockup antarmuka, yang menjadi langkah awal dalam menciptakan desain visual aplikasi. Estimasi waktu yang dibutuhkan untuk menyelesaikan tugas ini adalah 12 hari. Berikut adalah rincian hasil dari *Sprint* Planning untuk *Sprint* 1, yang tercantum dalam Tabel 2.

Tabel 2. *Sprint* 1 dalam Pembuatan *Dashboard Monitoring*

Item	Prioritas	Estimasi Waktu (hari)
Card success	Prioritas Tinggi	4
Card fail	Prioritas Tinggi	4
Card no delivery	Prioritas Tinggi	3
Filter	Prioritas Tinggi	1

Setelah *Sprint Planning*, tim melanjutkan dengan *daily meeting* untuk memantau perkembangan dan mengatasi kendala. Dalam *daily standup*, anggota tim melaporkan kemajuan, rencana hari itu, dan kendala yang dihadapi. Pada *Sprint* 1, kendala utama adalah kesulitan dalam mendesain *mockup* antarmuka yang sesuai dengan kebutuhan pengguna dan pemilihan alat desain yang tepat. Di *daily checkpoint*, tim mengevaluasi hasil *sprint* dan mencari solusi, seperti memperpanjang waktu untuk desain atau menyesuaikan alat desain agar lebih efisien. Desain yang ditampilkan pada Gambar 3 merupakan hasil proses *sprint*, beberapa elemen penting dirancang dengan prioritas tinggi, yaitu *Card success* dan *Card fail*, masing-masing dengan estimasi waktu pengerjaan 4 hari, *Card no delivery* dengan estimasi waktu 3 hari, serta *Filter* yang memerlukan waktu 1 hari.

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 64-73
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

Gambar 3. Tampilan *Sprint* 1: *Mockup Dashboard*

Kemudian, dilakukan *Sprint Demo* berdasarkan *Product Backlog Sprint* 1. Tabel 3 menunjukkan hasil dari demo yang dilakukan bersama *Scrum Master* dan Tim Pengembang. Dalam *Sprint* Demo ini, tim mempresentasikan hasil yang telah dicapai selama *Sprint* 1, menerima umpan balik dari pemangku kepentingan, dan mendiskusikan langkah-langkah perbaikan yang diperlukan.

Tabel 3. Hasil *Sprint Demo* pada *Sprint 1*

Hasil Demo	Umpan Balik	Tindak Lanjut
Presentasi *mockup* antarmuka selesai	Pemangku kepentingan meminta perubahan pada warna dan tata letak untuk mempermudah navigasi	Revisi desain berdasarkan masukan, perbaikan warna dan tata letak

Setelah *Sprint Demo*, tim melanjutkan dengan *Sprint Retrospective* untuk mengevaluasi proses kerja selama *Sprint* 1. Pada retrospektif ini, tim menganalisis apa yang berjalan dengan baik, seperti koordinasi yang efektif antar anggota tim dan penyelesaian mockup antarmuka tepat waktu. Namun, beberapa area yang perlu diperbaiki adalah pemilihan alat desain yang memerlukan waktu lebih lama dari yang diperkirakan dan kurangnya pengujian awal desain pada pengguna potensial. Tim sepakat untuk memperbaiki proses pemilihan alat desain di *sprint* berikutnya dan melakukan pengujian awal lebih awal untuk memastikan kesesuaian desain dengan kebutuhan pengguna. Hasil dari retrospektif ini digunakan untuk membuat perbaikan pada proses tim dan meningkatkan kualitas hasil *sprint* yang akan datang.

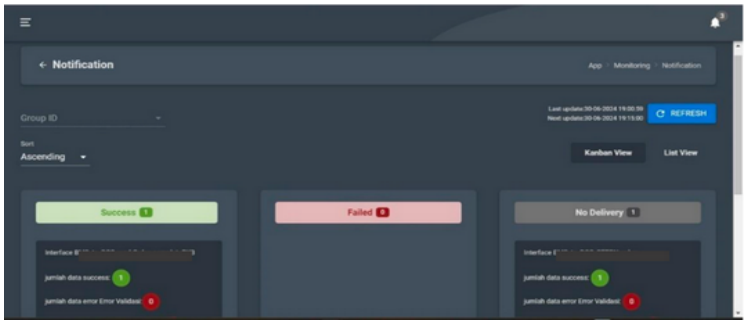
3.2.2. *Sprint* 2

Setelah *Sprint* 1 selesai, tim melanjutkan dengan *Sprint* 2 yang berfokus pada pembuatan *frontend* untuk *dashboard monitoring*. Proses ini mencakup pengembangan antarmuka pengguna yang akan digunakan untuk menampilkan data status dokumen secara visual. Estimasi waktu yang dibutuhkan untuk menyelesaikan tugas ini adalah 14 hari. Berikut adalah rincian hasil dari *Sprint Planning* untuk *Sprint* 2, yang tercantum dalam Tabel 4.

Tabel 4. *Sprint* 2 dalam Pembuatan *Dashboard Monitoring*

Sprint	*ProductBacklog*	Estimasi Waktu (hari)
Sprint 2	Membuat *frontend*	14

Setelah *Sprint Planning*, tim melanjutkan dengan *daily meeting* untuk memantau perkembangan dan mengatasi kendala. Dalam daily standup, anggota tim melaporkan kemajuan, rencana hari itu, dan kendala yang dihadapi. Pada *Sprint* 2, proses berjalan dengan lebih baik, tim pengembang sudah mulai bisa menjalankan ritme *sprint* dengan cepat dan sesuai dengan waktu yang ditetapkan sebagaimana ditunjukkan pada Gambar 4.



Gambar 4. *Hasil Sprint* 2 Implementasi Rancangan Antarmuka

Pada *Sprint Demo*, tim mempresentasikan hasil pengembangan *frontend* yang telah selesai. Hasil *demo* ini diterima dengan umpan balik bahwa front end sudah sesuai dengan *mockup* yang disahkan pada *sprint* 1.

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 64-73
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

Beberapa elemen desainpun sudah disesuaikan agar lebih responsif dan kompatibel dengan berbagai perangkat. Tabel 5 merupakan Hasil *Sprint Demo* pada *Sprint* 2.

Tabel 5. Hasil *Sprint Demo* Pada *Sprint* 2

Hasil Demo	Umpan Balik	Tindak Lanjut
Frontend selesai	Sudah sesuai dengan mockup	Koordinasi dengan tim *backend*

Setelah *Sprint Demo*, tim melanjutkan dengan *Sprint* Retrospective untuk mengevaluasi proses kerja. *Sprint* 2 dinyatakan selesai dengan hasil yang telah dicapai, namun dengan perbaikan yang diperlukan untuk memastikan integrasi yang lebih baik di masa mendatang.

3.2.3. *Sprint* 3

Setelah *Sprint* 2 selesai, tim melanjutkan dengan *sprint* 3 yang berfokus pada pengembangan *backend* untuk fitur yang telah direncanakan, termasuk pengiriman email otomatis dan penyimpanan data menggunakan *scheduler*. Berdasarkan Tabel 6, *sprint* ini membutuhkan waktu 14 hari untuk menyelesaikan seluruh pengembangan *backend* sesuai dengan kebutuhan sistem.

Tabel 6. *Sprint 3* dalam Pembuatan *Dashboard Monitoring*

Sprint	*Product Backlog*	Estimasi Waktu (hari)
Sprint 3	Membuat *backend*	14

Pada tahap ini, tim melakukan *daily meeting* untuk memantau perkembangan dan mengatasi kendala. Dalam daily standup, anggota tim melaporkan kemajuan, rencana hari itu, dan masalah yang menyebabkan beberapa agenda *overdue*. Kendala yang dihadapi pada *Sprint* 3 adalah pengintegrasian *backend* dengan *database* dan penyusunan logika pengiriman email otomatis. Hal ini memerlukan waktu ekstra karena beberapa masalah terkait kompatibilitas server dan query database yang perlu dioptimalkan.

Gambar 5. Hasil *Sprint* 3 Isi *Body Email*

Sprint ketiga menghasilkan *backend* yang ditunjukkan pada Gambar 5, yaitu fitur notifikasi melalui email. *Backend* ini mampu mengirimkan email yang berisi status dari proses file yang berlangsung dalam sebuah antarmuka, serta memungkinkan untuk menentukan penerima yang akan mendapatkan notifikasi tersebut.

Sebagaimana yang terlihat pada Tabel 7, hasil dari *Sprint Demo* pada *Sprint* 3 menunjukkan bahwa pengembangan *backend*, termasuk pengaturan *scheduler* untuk pengiriman email otomatis, telah berhasil diselesaikan tanpa ada revisi atau penyempurnaan lebih lanjut.

Tabel 7. *Sprint 3* dalam Pembuatan *Dashboard Monitoring*

Hasil Demo	Umpan Balik	Tindak Lanjut
Pengaturan *scheduler* untuk pengiriman email otomatis	Tidak ada revisi atau penyempurnaan	Tidak ada

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 64-73
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

3.3. Pengujian

Pengujian aplikasi MyZ dilakukan untuk menguji fungsionalitas aplikasi selama proses *sprint* berlangsung. Selain itu, dilakukan pengujian kecepatan pada dua jenis peramban yang berbeda, yaitu *Google Chrome* (PGC) dan *Microsoft Edge* (PME), dengan menguji dua fitur yang berbeda, yaitu *Button Refresh* (BR) dan *Send Email* (SEL). Pengujian dilakukan pada tanggal yang berbeda-beda, namun semuanya dilakukan pada hari kerja dengan waktu yang bervariasi, mulai dari pagi, siang, hingga sore hari. Rincian hasil pengujian untuk setiap fitur dapat dilihat pada Tabel 8 dan Tabel 9.

Berdasarkan Tabel 8 fitur BR lebih cepat memproses pembaruan data sebesar 0,1 detik menggunakan peramban PME dibandingkan peramban PGC, hal tersebut dihitung berdasarkan total waktu pengujian yang tercatat. Pada Chrome, total waktu pengujian untuk *Button Refresh* adalah 6,4 detik, dengan rata-rata 1,6 detik dari empat sesi pengujian. Sementara itu, pada *Microsoft Edge*, total waktu pengujian adalah 6 detik, menghasilkan rata-rata 1,5 detik dari empat sesi pengujian. Hasil pengujian pada Tabel 8 dipengaruhi oleh jumlah pengguna yang mengakses server. Kemudian tidak ditemukan perbedaan signifikan dalam kinerja karena kedua peramban.

Tabel 8. Skenario Uji Kecepatan Berdasarkan Waktu dan Peramban Pada Fitur *BR*

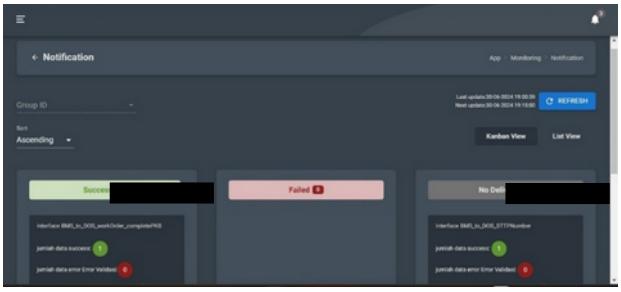
Nomor Skenario	Kode Fitur	Kode Peramban	Tanggal Pengujian	Waktu Pengujian	Durasi (detik)	Rata-Rata
NS1	BR	PGC	13 Juni 2024	10.00	1,1	1,6
NS2				11.00	1,5	
NS3				13.00	2,0	
NS4				15.00	1,8	
NS5		PME	17 Juni 2024	10.00	1,1	1,5
NS6				12.00	1,0	
NS7				13.45	1,9	
NS8				15.00	2,0	

Hasil pengujian kecepatan berdasarkan waktu pada fitur SEL diperoleh rata-rata proses pengiriman data selama 3,3 detik. Dengan rentang waktu tercepat 2 detik pada pukul 20.00 WIB tanggal 14 Juni 2024. Sedangkan untuk rentang waktu pengiriman data yang relatif lama terjadi pada pukul 13.25 WIB yang mencapai 4 detik. Hal ini disebabkan oleh aktivitas pengiriman pada siang hari cenderung lebih banyak dibandingkan malam hari.

Tabel 9. Skenario Uji Kecepatan Berdasarkan Waktu Pada Fitur *SEL*

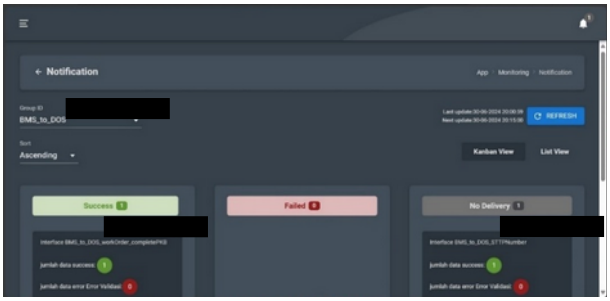
Nomor Skenario	Kode Fitur	Kode Peramban	Tanggal Pengujian	Waktu Pengujian	Durasi (detik)	Rata-Rata
NS9	SEL	-	10 Juni 2024	10.00	3,5	3,3
NS10			14 Juni 2024	13.25	4	
NS11				15.00	3,9	
NS12				17.00	3,1	
NS13				20.00	2	

Selain itu, pengujian kompatibilitas dilakukan untuk mengevaluasi apakah terdapat perbedaan tampilan antarmuka antara *Chrome* dan *Microsoft Edge*. Gambar 6 merupakan gambaran ketika *dashboard* dibuka pada *Chrome*, sedangkan Gambar 7 menunjukkan tampilan *dashboard* pada *Microsoft Edge*.



Gambar 6. Tampilan MyZ pada PGC

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 64-73
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index



Gambar 7. Tampilan MyZ pada PME

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, aplikasi MyZ telah berhasil dikembangkan dan diuji menggunakan kerangka *scrum*. Pengujian dilaksanakan pada setiap *sprint* dengan tiga iterasi menunjukkan bahwa aplikasi ini telah memenuhi kebutuhan pengguna serta mampu mengelola data. Selain itu, pengujian kecepatan pada akhir iterasi menunjukan bahwa rata-rata waktu pengiriman data sekitar 3,3 detik dan waktu proses informasi data *interface* yang berhasil, gagal, dan *no delivery* di *dashboard* tercepat selama 1,5 detik. Kedua peramban yang diuji, Chrome dan *Microsoft Edge*, memberikan kinerja yang serupa tanpa perbedaan signifikan dalam hal kecepatan atau tampilan antarmuka.

Untuk pengembangan lebih lanjut, aplikasi ini memiliki potensi untuk ditingkatkan dengan penambahan fitur visualisasi data berbasis grafik yang lebih interaktif. Selain itu, integrasi fitur notifikasi aplikasi pesan instan atau SMS dapat lebih meningkatkan responsivitas pengguna terhadap isu yang muncul tanpa perlu membuka *dashboard* secara langsung.

DAFTAR PUSTAKA

[1] M. Stefanovic, D. Przulj, S. Ristic, D. Stefanovic, and D. Nikolic, "Smart Contract Application for Managing Land Administration System Transactions," *IEEE Access*, vol. 10, no. April, pp. 39154–39176, 2022, doi: 10.1109/ACCESS.2022.3164444.

[2] L. P. Handho and S. D. Purnamasari, "Dasboard Monitoring Mahasiswa Dan Lulusan Untuk Meningkatkan Potensi Penerimaan Mahasiswa Baru Serta Strategi Pemasaran," *J. Comput. Inf. Syst. Ampera*, vol. 1, no. 2, pp. 91–98, May 2020, doi: 10.51519/journalcisa.v1i2.37.

[3] E. Ernawan, "Pemanfaatan Management Dashboard Dalam Pengambilan Keputusan Strategis Pada Perusahaan Bisnis Konstruksi (Studi Kasus PT. XYZ)," *J. Mirai Manag.*, vol. 9, no. 2, p. 124, 2024, doi: 10.37531/mirai.v9i2.7174.

[4] R. Dwi Bima Sakti, S. Lestanti, and S. Nur Budiman, "Perancangan Dashboard Monitoring Penjualan Pada Website Pateron.Id Menggunakan Framework Laravel Dan Vue Js," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 2, pp. 1731–1738, 2024, doi: 10.36040/jati.v8i2.9146.

[5] J. Angara, S. Prasad, and G. Sridevi, "DevOPs project management tools for *sprint* planning, estimation and execution maturity," *Cybern. Inf. Technol.*, 2020, doi: 10.2478/cait-2020-0018.

[6] D. Nugraha, I. L. Nur, M. T. Hidayatuloh, R. H. Laluma, and Gunawan, "Implementasi Sistem Informasi Manajemen Kantor Menggunakan Scrum Framework Di Desa Wangunsari," *J. Ilm. Media Sisfo*, vol. 17, no. 1, pp. 116–124, 2023, doi: 10.33998/mediasisfo.2023.17.1.740.

[7] N. Rafianto and K. Madiun, "979-2466-1-Pb," *J. Sist. Inf. dan Sains Teknol.*, vol. 3, no. 2, pp. 1–14, 2021, [Online]. Available: https://www.neliti.com/publications/492081/penerapan-metode-scrum-pada-pembuatan-user-experience-landing-page-sistem-inform.

[8] A. Bimantara and J. Sutresna, "Perancangan Dashboard Sistem Informasi Dengan Metode Scrum Menggunakan Azure Board," *OKTAL J. Ilmu Komput. dan Sci.*, vol. 3, no. 2, pp. 519–526, 2024.

[9] W. Warkim, M. H. Muslim, F. Harvianto, and S. Utama, "Penerapan Metode SCRUM dalam Pengembangan Sistem Informasi Layanan Kawasan," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 365–378, 2020, doi: 10.28932/jutisi.v6i2.2711.

[10] F. A. Dzaky and D. Kurniawan, "Implementasi Metode Agile Framework Scrum dalam Pengembangan Sistem Informasi Manajemen Aset Terpadu Universitas Diponegoro Modul Inventarisasi," *J. Masy. Inform.*, vol. 14, no. 1, pp. 2777–0648, 2023.

[11] D. Murdiani, A. Yudhana, and S. Sunardi, "Implementasi Agile Method dalam Pengembangan Jurnal Elektronik di Lembaga Penelitian Non Pemerintahan (NGO)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, p. 709, 2020, doi: 10.25126/jtiik.2020741839.

[12] E. Setiadana, "Pengembangan Sistem Penagihan Biaya Kuliah Dengan Fitur WhatsApp Menggunakan Metode Scrum Berbasis Website," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 3, pp. 1252–1264, 2021, doi: 10.35957/jatisi.v8i3.1041.

[13] M. D. Kadenic, D. A. de Jesus Pacheco, K. Koumaditis, G. Tjørnehøj, and T. Tambo, "Investigating the role of Product Owner in Scrum teams: Differentiation between organisational and individual impacts and opportunities," *J. Syst. Softw.*, vol. 206, no. February 2024, p. 111841, 2023, doi: 10.1016/j.jss.2023.111841.
[14] T. Žužek, J. Kušar, L. Rihar, and T. Berlec, "Agile-Concurrent hybrid: A framework for concurrent product development using Scrum," *Concurr. Eng.*, vol. 28, no. 4, pp. 255–264, Sep. 2020, doi: 10.1177/1063293X20958541.
[15] A. Wahyudi, S. Sunardi, and I. Riadi, "Peran Strategis Scrum Master Pada Pengembangan Perangkat Lunak Perpustakaan Sekolah Berbasis Android," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 3, pp. 711–717, 2022, doi: 10.29100/jipi.v7i3.2994.
[16] A. P. Pratama and R. A. Zunaidi, "Implementasi scrum model dalam pengembangan aplikasi e-commerce pada bidang jasa pembangunan rumah," *JENIUS J. Terap. Tek. Ind.*, vol. 4, no. 1, pp. 39–48, 2023, doi: 10.37373/jenius.v4i1.484.