



Implementasi *Agile* untuk Pengembangan UI Guidofiks: Aplikasi Diagnosis Kendaraan

Nur Ghulam Mustafa Al Kautsar*, Ahmad Hilmi Dwi Setiawan, Arjuna Ahmad Dewangga Aljabbar, Dafa Fawwaz Alfarisi, Mohammad Rafi Novyansyah, Sholiq

Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember

Abstract

The development of information technology has brought significant changes across various sectors, including the field of User Interface (UI) design. UI development is crucial to ease the use of an application for users. The Guidofiks application, designed for vehicle diagnostics, is created with a user-friendly UI. This application utilizes an On-Board Diagnostics (OBD) system to monitor vehicle performance in real-time. The Agile method and Scrum framework are used for the UI development of the Guidofiks application, involving user stories, product backlog, sprint planning, sprint execution, and sprint review. Nine main items were identified in the product backlog to be developed by the team. Some of the resulting items include login, registration, homepage, diagnostic feature, profile page, breakdown probability feature, community feature, nearest workshop feature, FAQ page, and feedback. The Agile approach in Guidofiks has proven effective in enhancing responsive development to changes. Although still in prototype form, the next project is expected to fully implement Agile, resulting in a complete software product.

Keywords: Agile, User Interface, Scrum

Abstrak

Perkembangan teknologi informasi telah membawa perubahan signifikan di berbagai sektor termasuk di bagian pengembangan desain antarmuka (UI). Pengembangan UI sangat penting untuk memudahkan pengguna dalam menggunakan sebuah aplikasi. Aplikasi Guidofiks, merupakan aplikasi untuk diagnosis kendaraan, dirancang dengan UI agar mudah digunakan. Aplikasi ini memanfaatkan sistem OBD (*On-Board-Diagnostic*) untuk memonitor kinerja kendaraan secara *real time*. Metode *Agile* dan kerangka *Scrum* merupakan metode untuk pengembangan UI dari aplikasi Guidofiks, prosesnya melibatkan *user story*, *product backlog*, perencanaan *sprint*, eksekusi *sprint*, dan *sprint review*. Didapatkan 9 item utama dalam *product backlog*, yang akan dibuat oleh tim. Beberapa hasil item yang didapatkan yaitu, *login*, *register*, *homepage*, fitur diagnosa, halaman profil, fitur probabilitas mogok, fitur komunitas, fitur bengkel terdekat, halaman FAQ, dan *feedback*. Pendekatan *Agile* pada Guidofiks efektif dalam meningkatkan pengembangan yang responsif terhadap perubahan. Meskipun masih dalam bentuk prototipe, proyek selanjutnya diharapkan dapat mengimplementasikan *Agile* secara penuh hingga menghasilkan produk perangkat lunak yang lengkap.

Kata kunci: Agile, Antarmuka Pengguna, Scrum

*Corresponding Author

Email address: ghulammusthafa04@gmail.com (Nur Ghulam Mustafa Al Kautsar)
<https://doi.org/10.24089/j.sisfo.2026.09.001>

1. Pendahuluan

Pada abad ke-21, teknologi informasi berkembang dengan sangat cepat, membawa berbagai perubahan di berbagai sektor [1]. Perkembangan ini tidak hanya mencakup kemajuan perangkat keras dan perangkat lunak, tetapi juga transformasi dalam cara kita bekerja, berkomunikasi, dan menyelesaikan masalah sehari-hari. Salah satu contoh penerapan teknologi informasi (TI) yang menonjol adalah dalam konteks pengembangan Desain Antarmuka Pengguna (*User Interface Design*) suatu aplikasi, yang memungkinkan penggunaannya dapat mengakses serta memahami fitur-fitur dengan mudah dan intuitif.

Pengembangan antarmuka pengguna memiliki peranan penting dalam menentukan keberhasilan sebuah aplikasi. Antarmuka yang ideal dirancang secara sederhana namun tetap menarik, sehingga dapat memudahkan pengguna dalam memahami dan mengoperasikan aplikasi tanpa kesulitan [2]. Dalam konteks ini, aplikasi GuidoFiks, sebuah aplikasi untuk diagnosis kendaraan, dirancang dengan pendekatan yang sederhana dan intuitif untuk menarik dan mempengaruhi pengguna dalam mengoperasikan aplikasi tersebut.

Dalam pengembangan GuidoFiks, metode *Agile* dipilih sebagai kerangka kerja utama demi menekankan fleksibilitas, kolaborasi tim, dan responsivitas terhadap perubahan kebutuhan. Selain itu, *scrum* merupakan *framework* yang tepat dalam pengembangan sistem yang dibuat. *Scrum* membagi pengembangan sistem menjadi beberapa *Sprint*. Dengan cara ini, pengembangan bisa terfokus pada tujuan yang ingin dicapai sambil tetap memberikan kesempatan untuk mengevaluasi sistem jika ada masalah melalui *Sprint Review* [3]. *Sprint* juga memungkinkan tim pengembang untuk membangun atau mengelola proses teknologi informasi menjadi lebih efektif, bekerja dalam iterasi-iterasi pendek dan teratur dengan tujuan untuk menghasilkan produk yang dapat diuji dan dievaluasi secara berkala [4].

Dengan adanya sistem ini, pengguna aplikasi dapat dengan mudah mengakses dan memanfaatkan berbagai fitur diagnostik kendaraan yang disediakan oleh GuidoFiks. Dengan memanfaatkan OBD (*On Board Diagnostics*) sebagai sistem diagnostik bawaan pada kendaraan, aplikasi ini mampu memonitor kinerja berbagai komponen mesin secara *real-time*. Teknologi ini memungkinkan pengguna untuk dengan cepat mengidentifikasi dan mengatasi masalah pada kendaraan mereka, sehingga menghemat waktu dan tenaga. Selain itu, pengalaman pengguna menjadi lebih lancar dan efisien karena mereka dapat mengandalkan informasi yang akurat dan tepat waktu dari sistem OBD untuk perawatan dan perbaikan kendaraan.

Jurnal ini tidak hanya menyajikan gambaran umum tentang implementasi Desain Antarmuka Pengguna, tetapi juga memberikan wawasan mendalam, mulai dari susunan tim pengembang *scrum*, pembuatan *user story*, *product backlog*, hingga kerangka *sprint* yang terperinci untuk mengidentifikasi dan memprioritaskan fitur-fitur yang harus dikembangkan, serta memastikan bahwa setiap iterasi *sprint* menghasilkan nilai yang signifikan bagi pengguna.

Penelitian ini diharapkan dapat menjadi referensi berharga bagi mahasiswa lain yang merencanakan atau sedang mengembangkan sistem serupa di lingkungan akademik. Dengan mempelajari pendekatan yang digunakan dalam pengembangan GuidoFiks, mahasiswa dapat memahami bagaimana merancang antarmuka pengguna yang intuitif dan adaptif serta menerapkan metode *Agile* dalam proyek pengembangan perangkat lunak mereka sendiri. Semoga hasil penelitian ini dapat menjadi referensi berharga bagi mahasiswa dan profesional di bidang teknologi informasi.

2. Tinjauan Pustaka

2.1 Agile SCRUM

Agile SCRUM adalah salah satu metodologi pengembangan perangkat lunak yang mengikuti prinsip-prinsip *Agile*. *SCRUM* adalah kerangka kerja yang dirancang untuk membantu tim bekerja sama dalam mengelola pengembangan produk. Ini menggunakan pendekatan iteratif dan inkremental, yang berarti bahwa produk dikembangkan melalui serangkaian iterasi yang disebut "*sprint*". Setiap *sprint* biasanya berdurasi antara satu hingga empat minggu dan berfokus pada penyelesaian sebagian dari produk yang dapat berfungsi penuh [5].

SCRUM memiliki tiga peran utama: *Product Owner*, *Scrum Master*, dan *Development Team*. *Product Owner* bertanggung jawab untuk menentukan fitur dan prioritas produk. *Scrum Master* bertindak sebagai fasilitator yang membantu tim mengatasi hambatan dan memastikan bahwa praktik *SCRUM* diikuti dengan benar. *Development Team* adalah kelompok profesional lintas fungsi yang bertanggung jawab untuk menghasilkan *increment* produk pada setiap *sprint* [5].

Proses *SCRUM* dimulai dengan perencanaan *sprint*, di mana tim dan *Product Owner* mendiskusikan *backlog* produk dan memilih item yang akan dikerjakan selama *sprint*. Selama *sprint*, tim melakukan pertemuan harian (*daily stand-ups*) untuk membahas kemajuan dan hambatan. Setelah *sprint* selesai, tim mengadakan *sprint review* untuk menunjukkan hasil kerja mereka kepada *stakeholder* dan *sprint retrospective* untuk mengevaluasi proses dan mencari cara untuk meningkatkan [6].

2.2 UI (User Interface)

User Interface (UI) adalah bagian dari sistem perangkat lunak yang memungkinkan interaksi antara manusia dan komputer. UI mencakup semua elemen visual dan interaktif yang membantu pengguna berinteraksi dengan perangkat lunak atau perangkat keras, termasuk layar, tombol, ikon, dan menu. Desain UI yang baik bertujuan untuk membuat interaksi pengguna menjadi seefisien dan seintuitif mungkin, mengurangi kurva pembelajaran dan meningkatkan produktivitas pengguna [7].

Desain UI yang efektif mempertimbangkan prinsip-prinsip seperti konsistensi, umpan balik, dan keterjangkauan. Konsistensi berarti elemen UI yang sama harus berperilaku dengan cara yang sama di seluruh aplikasi. Umpan balik adalah pemberian respons langsung terhadap tindakan pengguna, misalnya dengan menampilkan pesan kesalahan atau konfirmasi. Keterjangkauan berarti elemen UI harus mudah ditemukan dan digunakan oleh pengguna [7].

Seiring dengan perkembangan teknologi, UI telah berkembang dari antarmuka berbasis teks menjadi antarmuka grafis yang lebih kompleks dan intuitif. UI modern juga sering mencakup elemen interaktif seperti gerakan (*gestures*) dan suara, yang membuat interaksi menjadi lebih alami dan efisien [8].

2.3 OBD (On-Board Diagnostics)

On-Board Diagnostics (OBD) adalah sistem yang digunakan dalam kendaraan untuk memantau dan melaporkan status berbagai subsistem kendaraan. OBD awalnya dirancang untuk memenuhi peraturan emisi kendaraan, dengan menyediakan cara untuk memantau dan mengontrol emisi kendaraan secara *real-time*. Sistem OBD memberikan akses kepada data diagnostik yang dapat digunakan untuk mendiagnosis dan memperbaiki masalah pada kendaraan [9].

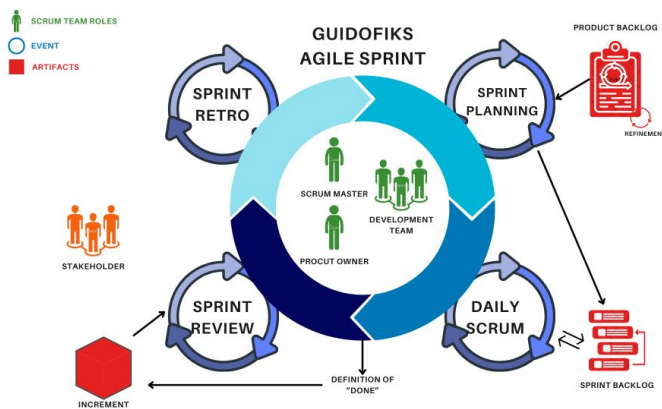
Generasi terbaru dari sistem ini, OBD-II, telah menjadi standar di banyak negara sejak pertengahan 1990-an. OBD-II menyediakan data diagnostik yang lebih rinci dan akses yang lebih luas terhadap informasi kendaraan, termasuk informasi tentang mesin, transmisi, dan komponen lain yang berhubungan dengan

emisi. Sistem ini juga mencakup konektor standar dan serangkaian kode kesalahan diagnostik yang dapat digunakan oleh mekanik dan perangkat lunak untuk mendiagnosis masalah kendaraan [9].

OBD-II dapat memonitor berbagai parameter kendaraan, seperti kecepatan kendaraan, putaran mesin (RPM), tekanan bahan bakar, suhu pendingin, dan lain-lain. Dengan menggunakan *scanner* OBD-II, teknisi dapat membaca kode kesalahan yang disimpan dalam memori komputer kendaraan dan mendapatkan wawasan tentang kondisi dan kinerja kendaraan. Hal ini memungkinkan pemeliharaan dan perbaikan yang lebih efisien, serta membantu dalam pemantauan dan pengurangan emisi kendaraan [10].

3. Metodologi

Metodologi pengembangan yang diterapkan pada Guidofiks adalah *Agile Scrum*, yang memungkinkan tim untuk merespons dengan cepat terhadap perubahan kebutuhan pengguna dan mengirimkan fitur dengan efisien. Melalui pendekatan *Scrum*, proses pengembangan dilakukan secara bertahap dalam beberapa *sprint*, sehingga setiap fitur dapat dirancang, diuji, dan dievaluasi secara berkala. Selain itu, *Scrum* juga mendukung kolaborasi tim yang lebih terstruktur melalui pembagian peran, *backlog*, *sprint planning*, dan evaluasi rutin. Dengan demikian, *Agile Scrum* dipilih karena mampu meningkatkan efisiensi pengembangan, mempercepat penyesuaian terhadap perubahan, serta menghasilkan aplikasi yang lebih responsif. Metodologi ini memiliki beberapa fase kunci yang dimulai dengan *Sprint Planning* dan selengkapnya ditunjukkan pada Gambar 1.

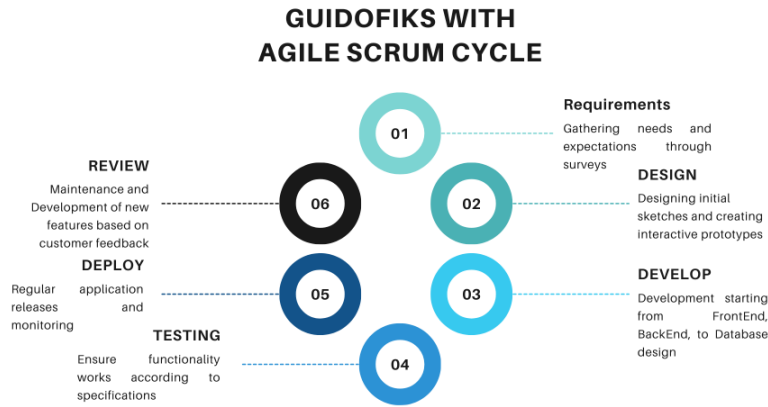


Gambar 1. Agile Sprint Guidofiks

Pada fase *sprint planning*, tujuan *sprint* ditetapkan dengan jelas berdasarkan *backlog* produk yang sudah diprioritaskan. Tim kemudian memecah fitur besar menjadi tugas-tugas kecil yang dapat dikelola, serta mengestimasi waktu dan usaha yang diperlukan untuk setiap tugas. Hal ini bertujuan untuk memastikan bahwa semua anggota tim memahami tujuan dan tugas yang akan dikerjakan selama *sprint* berlangsung. Selama *Daily Standup*, setiap anggota tim memberikan laporan progres harian, mengidentifikasi hambatan yang mungkin mengganggu progres mereka, dan berkoordinasi dengan anggota tim lain untuk memastikan semua tugas berjalan sesuai rencana. Ini membantu dalam menjaga transparansi dan kolaborasi antar anggota tim.

Fase selanjutnya adalah *Sprint Execution*, di mana tugas-tugas yang telah direncanakan diimplementasikan sesuai jadwal. Proses ini melibatkan integrasi perubahan kode secara berkala untuk menghindari konflik dan memastikan semua bagian dari aplikasi bekerja secara harmonis. Progres terus dipantau, dan jika diperlukan, penyesuaian tugas dapat dilakukan untuk memastikan bahwa tujuan *sprint* tercapai. Setelah tugas selesai, dilakukan *Sprint Review*. Dalam fase ini, fitur-fitur yang telah selesai dikembangkan ditunjukkan kepada pemangku kepentingan untuk mendapatkan umpan balik langsung. Umpan balik ini sangat penting karena akan digunakan untuk memperbarui *backlog* produk dan menetapkan

prioritas untuk sprint berikutnya. Setelah *Sprint Review*, tim melaksanakan *Sprint Retrospective* untuk merefleksikan proses yang telah berlangsung selama *sprint*. Tim mengidentifikasi area yang perlu diperbaiki dan merencanakan tindakan perbaikan untuk meningkatkan proses di sprint berikutnya [11]. Pada Gambar 2 menunjukkan beberapa siklus *Agile Scrum* pada aplikasi Guidofiks.



Gambar 2. Siklus *Agile Scrum Process*

Proses pengembangan dimulai dengan *Requirement Analysis*, di mana kebutuhan dan harapan pengguna dikumpulkan melalui survei, wawancara, dan observasi. Hasil dari analisis ini digunakan untuk membuat persona pengguna, yang membantu tim dalam memahami berbagai tipe pengguna dan kebutuhan spesifik mereka [12]. Pada tahap *UI/UX Design*, tim membuat *wireframes* sebagai sketsa awal antarmuka pengguna, diikuti dengan pembuatan prototipe interaktif menggunakan Figma. Prototipe ini kemudian diuji melalui *usability testing* untuk memastikan bahwa desain memenuhi kebutuhan pengguna dan mudah digunakan [13].

Implementasi teknis dimulai dengan pengembangan *frontend*, yang melibatkan pembuatan antarmuka pengguna menggunakan teknologi web modern seperti HTML, CSS, dan JavaScript. *Backend development* mencakup pembuatan logika server dan integrasi dengan OBD-II untuk mengakses data sensor kendaraan secara *real-time*. Selain itu, tim juga mendesain basis data untuk menyimpan informasi pengguna dan data diagnostik kendaraan secara aman dan efisien [14].

Testing merupakan bagian penting dari metodologi ini. Unit testing dilakukan untuk memastikan bahwa setiap fungsi individu bekerja sesuai dengan spesifikasinya. Integration testing kemudian dilakukan untuk memastikan bahwa semua komponen bekerja bersama dengan baik. Terakhir, *User Acceptance Testing (UAT)* dilakukan dengan melibatkan pengguna akhir untuk memastikan bahwa aplikasi memenuhi kebutuhan dan harapan mereka sebelum diluncurkan ke pasar [15].

Setelah semua tahapan testing selesai, *Deployment* dilakukan. Proses ini dimulai dengan pengujian di lingkungan *staging* untuk memastikan bahwa aplikasi bekerja dengan baik sebelum dirilis ke lingkungan produksi. Setelah rilis, aplikasi dipantau secara berkala untuk memastikan performanya optimal dan untuk menangani masalah yang mungkin muncul setelah pengguna mulai menggunakan aplikasi. Tahap *Maintenance* meliputi perbaikan bug yang ditemukan setelah rilis, pengembangan fitur baru berdasarkan umpan balik pengguna, dan pembaruan reguler untuk meningkatkan performa serta keamanan aplikasi. Pendekatan ini memastikan bahwa Guidofiks tidak hanya memenuhi kebutuhan pengguna pada saat peluncuran, tetapi juga terus berkembang dan beradaptasi dengan kebutuhan yang berubah serta teknologi terbaru.

4. Hasil dan Pembahasan

Pada Bagian ini akan dijelaskan mengenai hasil serta pembahasan yang telah didapat dalam penelitian ini.

4.1 User Story

User story merupakan cara mengekspresikan berbagai *requirements software* dengan bentuk deskripsi fungsionalitas yang singkat dari sudut pandang pengguna, terutama pada fitur-fitur yang berharga atau bermanfaat bagi mereka [16]. *User story* sendiri terdiri terbagi menjadi 3 aspek yaitu deskripsi tertulis, percakapan dan hasil testing atau detail dokumen [16]. Pada tahap penyusunan *user story*, proses pengumpulan data dari pengguna dilakukan diskusi informal bersama teman rumah yang akan berperan sebagai calon pengguna. Melalui percakapan ini, peneliti dapat menggali informasi dari pengguna tentang masalah yang sering dialami, kebutuhan utama, serta harapan terhadap aplikasi yang akan dikembangkan. Hasil dari diskusi tersebut digunakan sebagai dasar untuk merumuskan *user story* yang ditunjukkan pada Tabel 1.

Tabel 1. *User story*

Role	I want to	So that I can
User	Mendaftar dan membuat akun di aplikasi guidofiks	Bisa menggunakan aplikasi guidofiks beserta fitur-fiturnya
User	Melihat halaman homepage	Bisa melihat ringkasan dan informasi fitur utama yang tersedia di aplikasi
User	Memiliki halaman profil pribadi untuk akun sendiri	Melihat dan mengedit informasi pribadi pada akun sendiri
User	Mendapatkan informasi mengenai kondisi kerusakan kendaraan	Bisa menentukan langkah-langkah perbaikan yang akan dilakukan
User	Mengetahui probabilitas atau kemungkinan kendaraan saya akan mogok tiba-tiba	Bisa mempersiapkan rencana perjalanan alternatif dan melakukan tindakan pencegahan
User	Memiliki fitur komunitas pada aplikasi	Berinteraksi dan berbagi informasi seputar kendaraan saya dengan pengguna lain
User	Memiliki fitur bengkel terdekat	Mengetahui lokasi bengkel terdekat sehingga bisa mendapatkan bantuan untuk melakukan tindakan pada kendaraan secepat mungkin
User	Memiliki akses ke FAQ	Mendapatkan jawaban atas pertanyaan umum tanpa perlu bantuan dari support
User	Umpan balik tentang aplikasi	Menyampaikan pengalaman dan saran perbaikan

4.2 Product Backlog

Proses *Scrum* dimulai dengan meninjau *Product Backlog*, yang berisi semua persyaratan produk. Fitur dengan prioritas tertinggi diidentifikasi dan diprioritaskan, kemudian tim *Scrum* memperkirakan berapa banyak yang dapat diselesaikan dalam satu *Sprint* [17]. Komponen utama dari *product backlog* adalah *user story* karena *user story* banyak digunakan dalam pengembangan perangkat lunak *agile* untuk mengumpulkan kebutuhan. *Product owner* dalam metode *agile* bertanggung jawab untuk mengelola kebutuhan pelanggan dalam bentuk item *backlog* yang diprioritaskan dan mengkomunikasikannya kepada tim *agile* [18]. *Product Backlog* dapat dilihat di Tabel 2.

Tabel 2. *Product Backlog*

No	Item	Priority
1	UI Login & Register	High
2	UI Homepage	High
3	UI Fitur Diagnosa	High
4	UI Halaman Profil	Medium
5	UI Fitur Probabilitas Mogok	Medium
6	UI Fitur Komunitas	Medium
7	UI Fitur Bengkel Terdekat	Medium
8	UI FAQ	Low
9	UI Feedback	Low

Dalam pengembangan aplikasi, fitur nomor 1 hingga 3 pada tabel 2 dianggap sebagai prioritas utama yang harus diselesaikan selama *Scrum* harian untuk mencapai *Minimum Viable Product* (MVP). Fokusnya adalah pada pembuatan fitur inti yang penting. Setelah MVP tercapai, tim akan mengalihkan perhatian ke fitur nomor 4 hingga 7, yang memiliki prioritas menengah sebagai tambahan. Fitur-fitur ini dikerjakan setelah MVP tercapai dan mencakup penyempurnaan fitur yang ada atau penambahan fitur baru untuk meningkatkan pengalaman pengguna atau memperluas fungsionalitas produk.

Setelah fitur prioritas tinggi dan menengah selesai, tim akan melanjutkan ke fitur nomor 8 dan 9, yang memiliki prioritas rendah. Fitur-fitur ini merupakan tambahan yang tidak mendesak tetapi tetap penting untuk penyempurnaan keseluruhan aplikasi. Dengan membagi backlog menjadi prioritas tinggi, menengah, dan rendah, tim dapat mengelola sumber daya dengan lebih baik dan tetap fleksibel terhadap perubahan kebutuhan atau prioritas.

4.3 *Sprint*

Sprint adalah elemen kunci dalam metode *Scrum*, yang memiliki batasan waktu maksimal satu bulan atau kurang untuk menghasilkan suatu *increment* yang selesai, berfungsi, dan berpotensi untuk dikembangkan lebih lanjut. Durasi *sprint* biasanya tetap dan konsisten. Setelah *sprint* pertama selesai, proses dilanjutkan dengan *sprint* berikutnya [19]. *Sprint*, yang terdiri dari dua sesi yaitu *Sprint Planning* dan *Sprint Backlog*. Pada tahap perencanaan *Sprint*, tim *Scrum* mengadakan pertemuan untuk menilai daftar item dalam *Product Backlog*. Tim ini mendiskusikan setiap fitur yang akan dikerjakan oleh anggota tim, termasuk perkiraan waktu penyelesaian yang tercantum pada Tabel 3.

4.3.1 *Sprint Execution*

Sprint execution adalah sebuah fase dimana tim terlibat dalam implementasi pengembangan fitur yang telah ditentukan dalam *Sprint Planning* [20]. Proses diawali dengan *team meeting*, dilanjutkan dengan membuat *prototype* UI, lalu untuk tahap terakhirnya adalah testing. Proses pada *Sprint Backlog*, daftar fitur yang sebelumnya disusun dalam *Product Backlog* dan *Sprint Planning* dikumpulkan untuk dikerjakan oleh tim, seperti yang ditunjukkan pada Tabel 4, 5, 6, 7 dan 8. Setelah satu fitur dari *Sprint Planning* selesai, fitur berikutnya akan dikerjakan pada *Sprint* selanjutnya.

Tabel 3. *Sprint Planning*

No	Sprint Planning	Estimation Time (Time/Day)
1	UI Login & Register	1
2	UI Homepage	3
3	UI Fitur Diagnosa	3
4	UI Halaman Profil	3
5	UI Fitur Probabilitas Mogok	3
6	UI Fitur Komunitas	3
7	UI Fitur Bengkel Terdekat	3
8	UI FAQ	1
9	UI Feedback	1

Tabel 4. *Sprint Execution (Sprint Backlog 1)*

Sprint Backlog 1	Task	Estimation Time (Day/Hours)						
		1	2	3	4	5	6	7
UI Login & Register	Team Meeting	1						
	Buat Prototype UI di Figma	4						
	Testing	2						
Total		7						

Tabel 5. *Sprint Execution (Sprint Backlog 2)*

Sprint Backlog 2	Task	Estimation Time (Day/Hours)						
		1	2	3	4	5	6	7
UI Homepage & Halaman Profil	Team Meeting		2			2		
	Buat Prototype UI di Figma		5	5		5	5	
	Testing				3			3
Total		30						

Tabel 6. *Sprint Execution (Sprint Backlog 3)*

Sprint Backlog 3	Task	Estimation Time (Day/Hours)						
		8	9	10	11	12	13	14
UI Fitur Diagnosa & Probabilitas Mogok	Team Meeting	2			2			
	Buat Prototype UI di Figma	6	8		6	8		
	Testing			5			5	
Total		42						

Tabel 7. *Sprint Execution (Sprint Backlog 4)*

Sprint Backlog 4	Task	Estimation Time (Day/Hours)						
		14	15	16	17	18	19	20
UI Fitur Komunitas & Bengkel Terdekat	Team Meeting	2			2			
	Buat Prototype UI di Figma	5	5		6	8		
	Testing			4			5	
Total		37						

Tabel 8. *Sprint Execution (Sprint Backlog 5)*

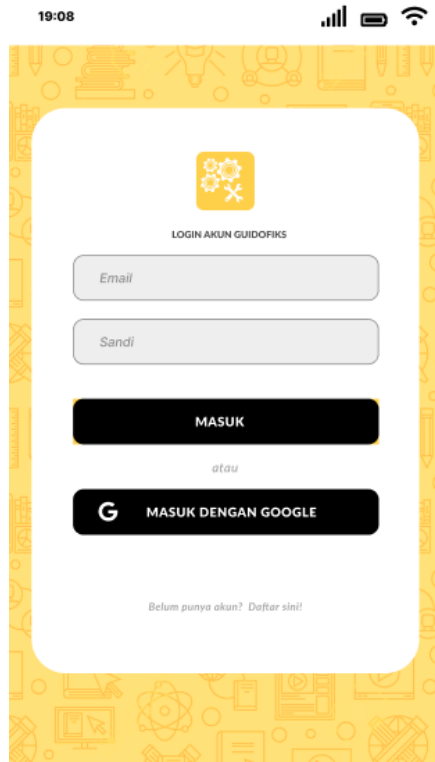
Sprint Backlog 5	Task	Estimation Time (Day/Hours)						
		15	16	17	18	19	20	21
UI FAQ & feedback	Team Meeting						1	1
	Buat Prototype UI di Figma						4	4
	Testing						2	2
Total		14						

4.3.2 *Sprint Review*

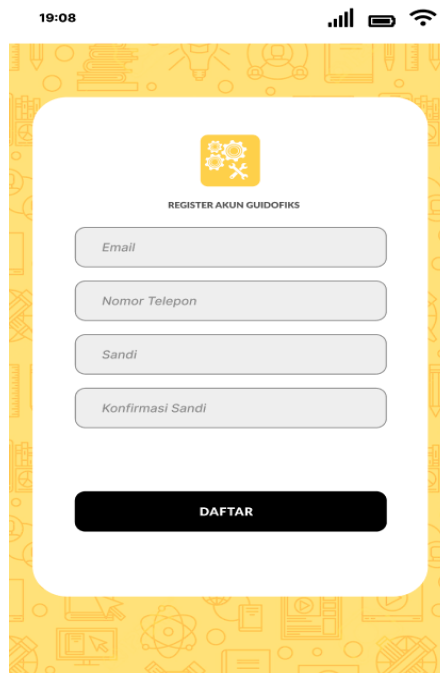
Sprint Review adalah sebuah kegiatan yang dilakukan setelah selesai *sprint* untuk mengevaluasi hasil kerja yang telah dilakukan oleh tim pengembang selama *sprint* tersebut[21]. Setelah serangkaian *Sprint* selesai, Guidofiks diproduksi dan pertama kali didemonstrasikan dalam ulasan *Sprint*. Guidofiks diperiksa secara berkala untuk memantau kemajuan dan menilai apakah produk yang dikembangkan telah memenuhi persyaratan. Kemudian dilakukan pengujian untuk memastikan bahwa fitur-fitur yang dikembangkan pada *Sprint* sebelumnya berfungsi dengan baik. Jika ditemukan fitur yang kurang memadai, fitur tersebut akan diperbaiki dan dikembangkan lebih lanjut dalam *Sprint* berikutnya. Pada Gambar 3 sampai 12 merupakan hasil dari UI Guidofiks di semua fitur yang ditunjukkan.

Fitur yang pertama yang diimplementasikan adalah UI *Login*, yaitu antarmuka awal yang dibangun untuk akses masuk pengguna ke dalam aplikasi Guidofiks. Pengguna dapat ada halaman ini, pengguna dapat melakukan proses autentikasi dengan memasukkan kredensial berupa email dan kata sandi pada kolom yang telah disediakan, kemudian menekan tombol utama “MASUK” untuk melanjutkan proses login. Selain itu terdapat fitur opsi login melalui integrasi pihak ketiga menggunakan akun Google melalui tombol “MASUK DENGAN GOOGLE”, sehingga memberikan kemudahan akses yang lebih praktis bagi pengguna. Bagi pengguna baru yang belum memiliki akun, antarmuka ini turut menyediakan navigasi menuju halaman registrasi melalui tautan “Belum punya akun? Daftar sini!” yang terletak pada bagian bawah halaman.

Pada Gambar 4 menampilkan halaman pendaftaran akun pada aplikasi GUIDOFIKS yang digunakan oleh pengguna baru untuk membuat akun sebelum menggunakan aplikasi. Pada halaman ini, pengguna perlu mengisi 4 *fields*, yaitu email, nomor telepon, kata sandi, dan konfirmasi kata sandi. Setelah pengguna sudah mengisi informasi tersebut, pengguna dapat menekan tombol “DAFTAR” untuk menyelesaikan proses pembuatan akun.



Gambar 3. *Sprint Review UI Login*

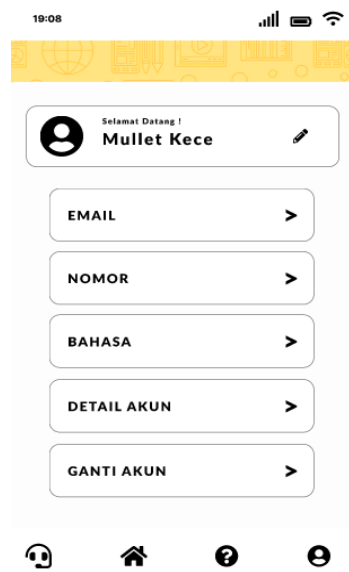


Gambar 4. *Sprint Review UI Register*



Gambar 5. *Sprint Review UI HomePage*

Gambar 5 tersebut menampilkan menu utama yang berisi empat fitur unggulan aplikasi untuk membantu pengguna merawat kendaraan mereka. Fitur pertama adalah Diagnosa untuk mengecek kerusakan, kemudian ada Probabilitas Mogok yang berfungsi memprediksi kemungkinan kendaraan bermasalah di tengah jalan. Selain itu, terdapat fitur Komunitas sebagai wadah berinteraksi dengan sesama pengguna dan fitur Bengkel Terdekat untuk mencari lokasi perbaikan di sekitar posisi pengguna. Di bagian bawah juga tersedia bar navigasi untuk menuju halaman utama, FAQ, halaman profil, dan saluran *feedback*.



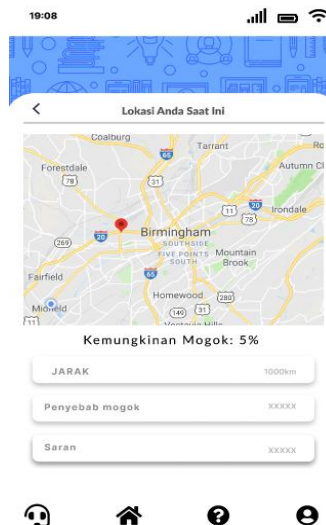
Gambar 6. *Sprint Review UI Halaman Profil*

Pada Gambar 6 menunjukkan halaman profil atau pengaturan akun pengguna di aplikasi GUIDOFIKS. Fitur-fitur yang tersedia di halaman ini berfungsi untuk mengelola informasi pengguna, yang terdiri dari menu Email dan Nomor untuk melihat atau merubah nomor telepon, serta menu Bahasa untuk mengatur pilihan bahasa dalam aplikasi yaitu bahasa indonesia atau bahasa inggris. Selain itu, terdapat fitur Detail Akun untuk melihat keaktifan akun dan juga mengajukan penghapusan akun. Terdapat juga fitur Ganti Akun bagi pengguna yang ingin beralih ke akun lain. Di bagian atas juga tersedia ikon pensil yang berfungsi untuk memperbarui nama profil pengguna.



Gambar 7. *Sprint Review* UI Fitur Diagnosa

Fitur diagnosa tersebut berfungsi untuk memberikan informasi mengenai kesehatan kendaraan pengguna secara mendetail. Fokus utama pada halaman ini adalah indikator persentase kesehatan kendaraan secara keseluruhan yang ditampilkan dalam lingkaran besar di bagian tengah. Selain itu, terdapat fitur rincian kondisi komponen mesin seperti *Engine* yang ditandai dengan tingkat prioritas tinggi untuk menunjukkan urgensi perbaikan, serta komponen *Brake* di bagian bawahnya. Halaman ini dirancang untuk memudahkan pengguna memantau bagian mana dari kendaraan yang membutuhkan perhatian segera agar performanya tetap terjaga.



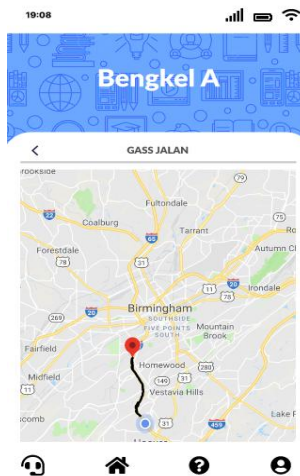
Gambar 8. *Sprint Review* UI Fitur Probabilitas mogok

Fitur Probabilitas Mogok yang berfungsi untuk memantau risiko kendaraan berhenti mendadak berdasarkan lokasi pengguna saat ini. Di bagian atas, terdapat peta yang menunjukkan posisi kendaraan secara *real-time*, diikuti dengan indikator persentase kemungkinan mogok untuk memberi peringatan kepada pengguna. Selain itu, halaman tersebut menyediakan informasi pendukung seperti jarak tempuh yang akan dilalui, kolom penyebab mogok untuk mengetahui potensi masalah teknis, serta kolom saran yang berisi langkah-langkah pencegahan yang harus diambil sebelumnya. Tampilan ini berguna bagi pengguna untuk merasa lebih aman selama perjalanan dengan memprediksi kendala sebelum terjadi.



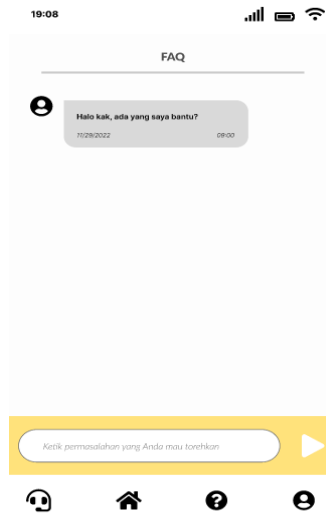
Gambar 9. *Sprint Review* UI Komunitas

Selanjutnya terdapat fitur Komunitas berfungsi sebagai ruang diskusi bagi sesama pengguna aplikasi GUIDOFIKS. Di halaman ini, pengguna bisa saling bertanya dan berbagi informasi seputar masalah kendaraan, seperti cara memperbaiki kampas rem atau mencari tempat ganti oli dan lain sebagainya. Terdapat kolom pencarian untuk menemukan pertanyaan tertentu serta tombol tambah untuk membuat pertanyaan baru. Setiap postingan dilengkapi dengan fitur balasan (*add response*) dan jumlah suka (*like*) untuk melihat seberapa populer atau bermanfaatnya suatu diskusi bagi anggota komunitas lainnya.



Gambar 10. *Sprint Review* UI Fitur Bengkel Terdekat

Fitur Bengkel Terdekat yang berfungsi untuk membantu pengguna menemukan dan menjangkau lokasi perbaikan kendaraan dengan cepat. Pada halaman ini, sistem menampilkan peta navigasi yang menunjukkan rute perjalanan dari posisi pengguna saat ini menuju bengkel tujuan, dalam gambar 10 adalah Bengkel A. Dengan adanya garis rute yang jelas fitur ini memudahkan pengguna untuk mengikuti navigasi secara *real-time* sehingga proses mencari bantuan teknis saat kendaraan bermasalah menjadi lebih efisien.



Gambar 11. *Sprint Review* UI FAQ

Fitur FAQ (*Frequently Asked Questions*) berfungsi sebagai percakapan atau *chat* untuk membantu pengguna mendapatkan solusi atas kendala mereka secara cepat. Di halaman pada gambar 11, terdapat pesan dari asisten virtual yang menawarkan bantuan kepada pengguna. Di bagian bawah layar, tersedia kolom input yang berfungsi sebagai tempat bagi pengguna untuk menuliskan pertanyaan atau masalah spesifik yang mereka alami. Fitur ini bertujuan untuk memberikan layanan dukungan yang interaktif, sehingga pengguna bisa mendapatkan jawaban atau bantuan teknis tanpa harus menunggu lama.



Gambar 12. *Sprint Review* UI Feedback

Tampilan Saluran *Feedback* pada gambar 12 berfungsi sebagai sarana bagi pengguna untuk memberikan masukan, saran, atau keluhan terkait layanan aplikasi GUIDOFIKS. Pada halaman ini, terdapat dua kolom utama, yaitu kolom Perihal untuk menuliskan topik atau judul masukan dan kolom Feedback untuk menjelaskan isi masukan secara detail. Setelah mengisi kedua kolom tersebut, pengguna dapat menekan tombol KIRIM di bagian bawah untuk menyampaikan ulasan mereka kepada tim pengembang. Fitur ini cukup penting untuk membantu pengembang memahami pengalaman pengguna dan melakukan perbaikan aplikasi di masa mendatang.

4.3.3 Testing

Pengembangan UI/UX dalam proyek kami akan kami uji menggunakan metode *scenario testing*. *Scenario testing* adalah metode pengujian pada sebuah proyek perangkat lunak yang berbasis skenario. Skenario sendiri merupakan sebuah cerita hipotetis yang digunakan seseorang untuk membantu mereka berpikir melewati masalah-masalah atau sebuah sistem yang kompleks. Kriteria skenario yang digunakan harus berdasarkan pada sebuah cerita tentang bagaimana program tersebut digunakan dan harus bersifat kredibel atau bisa terjadi di dunia nyata serta bisa dievaluasi [22]. *Scenario testing* pada proyek kami dibagi berdasarkan pengerjaan *sprint* dengan beberapa skenario pengujian yang sudah kami tentukan untuk mengevaluasi keberhasilan UI/UX aplikasi kami yang ditunjukkan pada Tabel 9.

5. Kesimpulan

5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan maka dapat diambil kesimpulan, yaitu:

1. Penerapan metode *Agile Scrum* pada perancangan UI/UX GuidoFiks menjadikan proses desain lebih efektif, terstruktur, dan sesuai dengan kapabilitas tim. Melalui *sprint* yang iteratif dan inkremental, proses perancangan dapat dilakukan secara bertahap sehingga setiap tahap menghasilkan desain yang terarah.
2. Penggunaan *Agile Scrum* memberikan fleksibilitas tinggi dalam proses perancangan, sehingga tim mampu menyesuaikan kebutuhan desain secara lebih cepat dan menjaga proses pengembangan tetap terarah pada tujuan utama aplikasi.
3. Implementasi metode *Agile* juga mendukung terciptanya desain antarmuka pengguna yang intuitif, adaptif, dan *user-friendly*, sehingga rancangan GuidoFiks mampu memberikan pengalaman penggunaan yang lebih optimal.
4. Penelitian ini menunjukkan bahwa metode *Agile Scrum* tidak hanya efektif dalam pengembangan perangkat lunak, tetapi juga relevan digunakan dalam proses perancangan UI/UX, serta dapat menjadi referensi dalam pengembangan aplikasi serupa di masa mendatang.

5.2 Saran

Desain UI/UX GuidoFiks yang telah dirancang masih dapat dikembangkan lebih lanjut hingga tahap implementasi aplikasi secara penuh. Penelitian selanjutnya disarankan untuk melakukan pengembangan sistem secara menyeluruh, termasuk integrasi *backend* dan perangkat OBD, agar rancangan dapat direalisasikan menjadi aplikasi fungsional. Selain itu, pengujian *usability* dengan lebih banyak pengguna juga diperlukan untuk meningkatkan kualitas antarmuka, pengalaman pengguna, serta efektivitas desain yang telah dibuat. Pengembangan fitur tambahan dan perluasan platform juga dapat menjadi langkah lanjutan untuk meningkatkan manfaat aplikasi di masa depan.

Tabel 9. Testing

Sprint	Testing Scenario	Test Result	Validation
UI Login & Register	Membuat akun baru menggunakan email melalui tombol daftar	Tombol berfungsi dan UI halaman pendaftaran muncul sehingga informasi akun baru bisa diisikan	Sukses
	Login dengan akun yang sudah terdaftar menggunakan tombol masuk	Tombol berfungsi dan user bisa mengetikkan nama dan password akun lalu tampilan berubah ke halaman beranda	Sukses
	Login dengan menggunakan akun google melalui tombol masuk dengan google	Tombol berfungsi dan halaman berubah ke halaman login khusus akun google	Sukses
UI Homepage	Halaman homepage yang sudah muncul bisa dilihat dan berhasil pindah halaman dengan navigation bar dibawah	Halaman homepage muncul dan menampilkan fitur utama aplikasi dan halaman berpindah sesuai dengan navigation bar di bawah UI	Sukses
UI Halaman Profil	Melakukan perubahan informasi akun dengan melakukan penggantian email dan nomor telepon dengan tombol edit	Setelah tombol ditekan muncul halaman perubahan email dan nomor telepon serta tombol ubah	Sukses
	Menampilkan informasi detail pada akun pengguna dengan tombol detail akun	Muncul halaman detail akun yang berisi nama, status keaktifan akun, metode pembayaran yang terhubung, durasi langganan dan opsi penghapusan akun	Sukses
UI Fitur Diagnosa	Menekan tombol fitur diagnosa pada homepage lalu mendiagnosa kendaraan dengan tombol-tombol yang tersedia pada halaman fitur	Halaman berubah menjadi halaman fitur diagnosa dan muncul persentase kondisi kesehatan kendaraan dan kerusakan yang ada	Sukses
UI Fitur Probabilitas Mogok	Menekan tombol fitur probabilitas mogok pada homepage dan melakukan perhitungan kemungkinan mogok dengan rute tertentu dengan menggunakan tombol yang ada	Halaman berubah ke halaman fitur dan bisa memasukkan rute perjalanan dan informasi persentase mogok dengan penyebab tertentu muncul	Sukses
UI Fitur Komunitas	Menekan tombol komunitas pada halaman beranda dan menampilkan halaman yang ada serta membuat posting pada komunitas menggunakan tombol tombol terkait	Halaman komunitas muncul dan posting pada komunitas bisa dilihat dan setelah di klik tombol + akan muncul popup untuk membuat posting baru	Sukses
UI Fitur Bengkel Terdekat	Menekan tombol fitur dari homepage dan menampilkan daftar bengkel terdekat setelah memasukkan lokasi terkini	Halaman berubah menjadi halaman fitur dan daftar bengkel terdekat muncul setelah menekan tombol cari	Sukses
	Menampilkan detail bengkel terdekat dan rute menuju bengkel tersebut	Setelah mengklik bengkel pada daftar yang tersedia halaman berubah menjadi halaman detail bengkel dan setelah mengklik tombol gas kesana rute akan muncul pada halaman baru	Sukses
UI FAQ	Menampilkan FAQ menggunakan navigation bar	Halaman FAQ muncul setelah tombol di navigation bar ditekan	Sukses
UI Feedback	Menampilkan halaman feedback dengan menekan tombol pada navigation bar	Halaman feedback muncul	Sukses

6. Daftar Rujukan

- [1] D. P. Tampubolon, N. Thesalonika, and T. Rustini, “Peran Media Pembelajaran Berbasis Teknologi Dalam Pembelajaran Daring,” *JURNAL ILMIAH SULTAN AGUNG*, pp. 9–20, Jun. 2022.
- [2] R. Fahrudin and R. Ilyasa, “Perancangan Aplikasi ‘Nugas’ Menggunakan Metode Design Thinking dan Agile Development,” *Reza Ilyasa Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 8, no. 1, pp. 35–44, Dec. 2021, doi: <https://doi.org/10.33197/jitter.vol8.iss1.2021.714>.
- [3] F. A. Dzaky and D. Kurniawan, “Implementasi Metode Agile Framework Scrum dalam Pengembangan Sistem Informasi Manajemen Aset Terpadu Universitas Diponegoro Modul Inventarisasi,” *Jurnal Masyarakat Informatika*, vol. 14, no. 1, pp. 53–69, 2023.
- [4] E. D. Ramadhany, A. Wirapraja, K. Afandi, and M. H. Arief, “Proses, Tantangan, dan Faktor Keberhasilan Transformasi Agile: Kajian Literatur Sistematis,” *Journal of Digital Business Innovation (DIGBI)*, vol. 1, no. 1, Nov. 2023, doi: <https://doi.org/10.36858/digbi.v1i1.4>.
- [5] Ken. Schwaber and Mike. Beedle, *Agile software development with Scrum*. Prentice Hall, 2001.
- [6] K. Schwaber and J. Sutherland, “The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game,” 2020.
- [7] Ben. Shneiderman, Catherine. Plaisant, M. Cohen, and S. Jacobs, *Designing the user interface : strategies for effective human-computer interaction*, 5th ed. Pearson, 2009.
- [8] D. A. . Norman, *The design of everyday things*. Basic Books, 2013.
- [9] R. Bosch, *Automotive handbook*, 9th ed. SAE, 2014.
- [10] W. H. . Crouse and D. L. . Anglin, *Automotive mechanics*, 10th ed. Glencoe, 1993.
- [11] D. L. Moody, “The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods,” 2003. [Online]. Available: <http://aisel.aisnet.org/ecis2003/79>
- [12] J. Aranda and G. Venolia, “The Secret Life of Bugs: Going Past the Errors and Omissions in Software Repositories,” in *2009 IEEE 31st International Conference on Software Engineering*, IEEE, 2009, pp. 298–308.
- [13] J. Bosch and P. B. Sijtsema, “Introducing agile customer-centered development in a legacy software product line,” *Softw. Pract. Exp.*, pp. 871–882, 2011, doi: <https://doi.org/10.1002/spe.1063>DigitalObjectIdentifier(DOI).
- [14] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, “Agile Software Development Methods: Review and Analysis,” 2017. doi: <https://doi.org/10.48550/arXiv.1709.08439>.
- [15] T. Memmel, F. Gundelsweiler, and H. Reiterer, “Agile Human-Centered Software Engineering,” in *Proceedings of the 21st British HCI Group Annual Conference on HCI 2007*, 2007.
- [16] M. Cohn, *User Stories Applied for Agile Software Development*. Addison-Wesley, 2009.
- [17] I. Kayes, M. Sarker, and J. Chakareski, “Product backlog rating: a case study on measuring test quality in scrum,” *Innov. Syst. Softw. Eng.*, vol. 12, no. 4, pp. 303–317, Dec. 2016, doi: [10.1007/s11334-016-0271-0](https://doi.org/10.1007/s11334-016-0271-0).
- [18] S. Sharma and D. Kumar, “Product backlog optimization technique in agile software development using clustering algorithm,” *Multimed. Tools Appl.*, vol. 82, no. 30, pp. 46695–46715, Dec. 2023, doi: [10.1007/s11042-023-15406-w](https://doi.org/10.1007/s11042-023-15406-w).
- [19] W. A. Prabowo and C. Wiguna, “Sistem Informasi UMKM Bengkel Berbasis Web Menggunakan Metode SCRUM,” *Jurnal Media Informatika Budidarma*, vol. 5, no. 1, pp. 149–156, Jan. 2021, doi: [10.30865/mib.v5i1.2604](https://doi.org/10.30865/mib.v5i1.2604).
- [20] L. A. Putri, E. D. Wahyuni, and Agussalim, “Rancang Bangun Order Management System Berbasis Web Menggunakan Framework Laravel (Studi Kasus: PT Galena Perkasa),” *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 3, pp. 4209–4216, 2024, doi: <https://doi.org/10.36040/jati.v8i3.9899>.
- [21] M. A. Firdaus, “Implementasi Kerangka Kerja Scrum Pada Manajemen Pengembangan Sistem Informasi,” 2017.
- [22] C. Kaner, “An Introduction to Scenario Testing,” 2013.