

# Implementasi *Scrum* pada Pengembangan UI/UX untuk E-Commerce Persewaan Barang Kebutuhan Event (Studi Kasus: RENT2GO)

Wahib Muhibi Nur\*, Muhammad Shalahuddin Arif Laksono, Baringga Aurico De Erwada, Muhammad Yusuf Purnama, Muhammad Zaidan Hafsh Fawwaz, Sholiq

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

---

## Abstract

Entering the post-pandemic recovery era, the event organizing industry in Indonesia is showing a rapid resurgence with a projected economic turnover exceeding IDR 162 trillion and involving thousands of business actors. However, the rental process for event equipment such as stages, audio-visual systems, and decorations is still conducted conventionally, creating inefficiencies for both event organizers and vendors. This study implements the Scrum method in developing the UI/UX flow for RENT2GO, an e-commerce platform designed to digitalize this process. The project was carried out across 8 sprints over 16 weeks, encompassing product backlog planning, sprint execution, and evaluation through sprint reviews and sprint retrospectives. Initial requirements were gathered through semi-structured interviews with representatives of event organizers and vendors as the basis for backlog development. The results indicate that Scrum enhanced team collaboration, process transparency, and responsiveness to changing requirements. The resulting UI/UX design fulfilled all Definition of Done criteria and received approval from the Product Owner. This study demonstrates that Scrum can serve as an effective method for developing e-commerce platforms with complex business workflows. More broadly, this research contributes an empirically grounded and adaptive model for designing transactional interfaces in rental-based e-commerce, serving as a reference for the development of similar platforms in the digital commerce sector.

*Keywords:* Scrum, User Interface, User Experience, Electronic Commerce, Goods Rental, Event Organizer.

## Abstrak

Memasuki era pemulihan pasca pandemi, industri penyelenggaraan acara di Indonesia menunjukkan kebangkitan pesat dengan proyeksi perputaran nilai ekonomi mencapai lebih dari Rp 162 triliun dan melibatkan ribuan pelaku usaha. Namun, proses sewa-menyewa barang acara seperti panggung, peralatan audio-visual, dan dekorasi masih dilakukan secara konvensional, sehingga menimbulkan inefisiensi bagi penyelenggara acara maupun vendor. Penelitian ini mengimplementasikan metode *Scrum* dalam pengembangan flow UI/UX platform *e-commerce* RENT2GO sebagai solusi digitalisasi proses tersebut. Proyek dilaksanakan dalam 8 *sprint* selama 16 minggu, mencakup tahap perencanaan *product backlog*, pelaksanaan *sprint*, serta evaluasi melalui *sprint review* dan *sprint retrospective*. Kebutuhan awal digali melalui wawancara semi-terstruktur dengan perwakilan penyelenggara acara dan vendor sebagai dasar penyusunan

---

\*Corresponding Author

Email address: wahibmuhibinur@gmail.com (Wahib Muhibi Nur)  
<https://doi.org/10.24089/j.sisfo.2026.09.002>

*backlog*. Hasil implementasi menunjukkan bahwa Scrum meningkatkan kolaborasi tim, transparansi proses, dan responsivitas terhadap perubahan kebutuhan. Desain UI/UX yang dihasilkan telah memenuhi seluruh kriteria *Definition of Done* dan mendapatkan persetujuan dari *Product Owner*. Penelitian ini menunjukkan bahwa *Scrum* dapat menjadi metode yang efektif dalam pengembangan platform *e-commerce* persewaan barang dengan alur bisnis yang kompleks. Secara lebih luas, penelitian ini berkontribusi sebagai model pendekatan empiris yang adaptif dalam merancang antarmuka platform *e-commerce* sewa-menyewa, sehingga dapat menjadi rujukan bagi pengembangan platform serupa di sektor perdagangan elektronik.

**Kata kunci:** *Scrum*, Antarmuka Pengguna, Pengalaman Pengguna, Perdagangan Elektronik, Penyewaan Barang, Penyelenggara Acara.

© 2026 Jurnal SISFO.

*Histori Artikel:* Disubmit 03-07-2024; Direvisi 08-05-2026; Diterima 13-05-2026; Tersedia online 31-05-2026

---

## 1. Pendahuluan

Memasuki era pemulihan pasca pandemi pada hingga 2024, industri penyelenggaraan *event* di Indonesia kembali menunjukkan kebangkitan yang pesat dengan proyeksi perputaran nilai ekonomi mencapai lebih dari Rp 162 triliun. Berdasarkan data Kementerian Pariwisata dan Ekonomi Kreatif, kebangkitan ini didorong oleh ribuan pelaku usaha *event* yang aktif beroperasi dan turut berkontribusi secara signifikan dalam target serapan jutaan tenaga kerja di sektor pariwisata dan ekonomi kreatif (Kemenparekraf, 2023). Dalam era digital yang semakin maju, kebutuhan akan platform *e-commerce* yang efisien dan *user-friendly* menjadi sangat krusial, terutama dalam industri penyelenggaraan acara. *E-commerce* persewaan barang untuk acara, seperti panggung, peralatan *audio-visual*, dekorasi, dan lain sebagainya, telah menjadi solusi yang populer bagi penyelenggara *event* yang membutuhkan akses cepat dan mudah ke berbagai barang sewa dari vendor terpercaya. Namun, untuk menciptakan platform yang efektif dan dapat diandalkan, diperlukan metodologi pengembangan perangkat lunak yang tepat.

Salah satu metodologi yang telah terbukti efektif dalam pengembangan perangkat lunak adalah *Scrum*. *Scrum* adalah *framework* yang berfokus pada pendekatan iteratif dan inkremental, memungkinkan tim pengembang untuk beradaptasi dengan cepat terhadap perubahan kebutuhan dan lingkungan bisnis yang dinamis. Metodologi ini juga menekankan kolaborasi tim, transparansi, dan peningkatan berkelanjutan, yang sangat penting untuk memastikan kualitas produk yang kompetitif di pasar. Proyek akhir ini akan mengimplementasikan metode *Scrum* dalam pengembangan *flow* UI/UX untuk platform *e-commerce* RENT2GO, yang khusus melayani kebutuhan penyelenggara acara dalam menyewa barang dari vendor terpercaya. Dengan menggunakan *Scrum*, diharapkan pengembangan perangkat lunak ini dapat lebih responsif terhadap kebutuhan pengguna dan perubahan di pasar industri penyelenggaraan acara. Selain itu, *Scrum* akan membantu memastikan bahwa *platform* yang dikembangkan memiliki kualitas yang tinggi dan dapat memenuhi ekspektasi pengguna. Makalah ini akan membahas proses implementasi *Scrum* dalam pengembangan *flow* UI/UX untuk RENT2GO, mulai dari tahap perencanaan, pelaksanaan *sprint*, hingga evaluasi dan perbaikan berkelanjutan. Diharapkan hasil dari proyek ini tidak hanya dapat memberikan kontribusi nyata bagi industri *e-commerce* persewaan barang, tetapi juga dapat menjadi referensi bagi pengembang lain yang ingin menerapkan metode *Scrum* dalam proyek mereka.

Implementasi *Scrum* dalam pengembangan *flow* UI/UX untuk platform *e-commerce* RENT2GO akan dimulai dengan tahap perencanaan yang matang. Pada tahap ini, tim pengembang bersama dengan pemangku kepentingan akan mengidentifikasi kebutuhan dan tujuan utama dari platform tersebut. *Product backlog* akan dibuat untuk mencatat semua fitur yang diinginkan beserta prioritasnya. Setiap item dalam *backlog* ini akan dianalisis secara mendalam untuk memastikan bahwa semua aspek fungsional dan non-fungsional telah dipertimbangkan. Perencanaan yang baik pada tahap ini sangat penting untuk memastikan bahwa tim memiliki pandangan yang jelas tentang apa yang harus dicapai selama setiap *sprint*. Setelah tahap

perencanaan selesai, tim akan memulai pelaksanaan *sprint*. Setiap *sprint* biasanya berlangsung selama dua hingga empat minggu dan berfokus pada penyelesaian sejumlah item *backlog* yang telah diprioritaskan. Pada awal setiap *sprint*, dilakukan *sprint planning meeting* untuk menentukan pekerjaan yang akan dilakukan selama *sprint* tersebut. Tim akan bekerja secara kolaboratif untuk mengembangkan, menguji, dan mengintegrasikan fitur-fitur baru. *Daily stand-up meetings* juga dilakukan setiap hari untuk memantau progres dan mengatasi hambatan yang mungkin muncul. Transparansi dan komunikasi yang baik antar anggota tim sangat penting dalam tahap ini untuk memastikan bahwa semua orang berada pada jalur yang sama.

Tahap terakhir adalah evaluasi dan perbaikan berkelanjutan yang dilakukan melalui *sprint review* dan *sprint retrospective*. *Sprint review* dilakukan untuk menampilkan hasil kerja *sprint* kepada pemangku kepentingan dan mendapatkan *feedback* yang dapat digunakan untuk perbaikan. Selanjutnya, *sprint retrospective* digunakan oleh tim untuk merefleksikan proses kerja mereka, mengidentifikasi apa yang berjalan baik dan apa yang perlu ditingkatkan. Melalui siklus berkelanjutan ini, tim dapat terus memperbaiki proses kerja mereka dan memastikan bahwa produk yang dihasilkan semakin berkualitas dan sesuai dengan kebutuhan pengguna. Dengan implementasi *Scrum* yang tepat, diharapkan RENT2GO dapat menjadi platform *e-commerce* yang handal dan memenuhi kebutuhan penyelenggara acara dengan baik.

## 2. Tinjauan Pustaka

### 2.1 Agile Development dan Scrum

#### 2.1.1 Agile Development

*Agile Development* merupakan metode pengembangan perangkat lunak yang memberikan fleksibilitas lebih tinggi daripada pendekatan tradisional berbasis rencana (*plan-driven*), sehingga adaptasi terhadap perubahan kebutuhan dapat dilakukan pada berbagai fase pengembangan. Pendekatan ini berpusat pada kolaborasi antara pelanggan dan pengembang, serta mendorong terciptanya tim yang mampu mengatur dirinya sendiri (*self-organizing*) [1]. Secara fundamental, metode ini berlandaskan pada *Agile Manifesto* (2001) yang menetapkan nilai-nilai dan prinsip dasar pengembangan guna memastikan tim dapat bekerja secara adaptif, responsif terhadap dinamika proyek, dan berfokus pada hasil produk yang bernilai [2].

#### 2.1.2 Metodologi Scrum

*Scrum* merupakan sebuah kerangka kerja ringan yang membantu individu, tim, dan organisasi dalam menghasilkan nilai melalui solusi adaptif untuk masalah yang kompleks. Meskipun berakar pada pengembangan perangkat lunak, *Scrum* kini digunakan di berbagai bidang pekerjaan yang kompleks. Di dalam *Scrum*, pekerjaan dilakukan dalam wadah iterasi berdurasi tetap selama satu bulan atau kurang yang disebut sebagai *Sprint*. Berdasarkan panduan resminya [3], setiap *Sprint* melibatkan beberapa acara utama:

1. *Sprint Planning*: Mengawali *Sprint* dengan menjabarkan pekerjaan yang akan dilakukan, di mana tim berkolaborasi untuk menentukan tujuan (*Sprint Goal*) dan merinci item-item dari *Product Backlog*.
2. *Daily Scrum*: Pertemuan harian bagi *Developers* untuk menginspeksi kemajuan menuju *Sprint Goal* dan menyesuaikan rencana kerja untuk hari berikutnya.
3. *Sprint Review*: Tahap peninjauan hasil kerja dari *Sprint* yang telah selesai, di mana tim mempresentasikan hasilnya kepada pemangku kepentingan dan mendiskusikan penyesuaian di masa mendatang.
4. *Sprint Retrospective*: Tahap evaluasi yang bertujuan untuk merencanakan cara-cara meningkatkan kualitas dan efektivitas, dengan mengidentifikasi perubahan yang paling bermanfaat untuk diterapkan pada *Sprint* berikutnya.

Lebih lanjut, kerangka kerja ini secara spesifik mendefinisikan tiga akuntabilitas utama di dalam satu kesatuan Tim *Scrum* [3]:

1. *Product Owner*: Bertanggung jawab untuk memaksimalkan nilai produk yang dihasilkan dari pekerjaan tim, serta mengelola *Product Backlog* secara transparan dan efektif.
2. *Scrum Master*: Bertanggung jawab untuk menetapkan pemahaman *Scrum* sesuai panduan, memastikan efektivitas tim, dan melayani tim untuk menghilangkan hambatan pekerjaan.
3. *Developers*: Para anggota di dalam tim yang memiliki komitmen penuh untuk menciptakan berbagai aspek dari Increment (produk yang berfungsi) yang dapat digunakan pada setiap *Sprint*.

## 2.2 User Interface (UI) dan User Experience (UX)

### 2.2.1 User Interface (UI)

*User Interface* (UI) adalah antarmuka pada perangkat lunak yang berperan penting untuk menarik pengguna, menjaga keterlibatan (*engagement*), dan memfasilitasi penyelesaian tugas-tugas komputasi secara efektif. UI tersusun dari berbagai elemen visual mendasar dengan fungsi spesifik yang dikelompokkan dalam suatu tata letak spasial. Desain UI yang baik harus estetik dan tidak rumit (*cumbersome*) untuk digunakan, karena desain yang buruk akan menurunkan tingkat keberhasilan sebuah aplikasi saat bersaing di pasar [4].

### 2.2.2 User Experience (UX)

*User Experience* (UX) dapat diinterpretasikan sebagai seberapa baik pengguna berinteraksi dengan suatu produk untuk mencapai tujuan mereka, yang mencakup emosi, persepsi, perilaku, dan pencapaian pengguna baik sebelum maupun sesudah menggunakan sistem. Pengalaman pengguna yang positif merupakan bagian krusial dalam proses pengembangan sistem, yang sangat dipengaruhi oleh faktor-faktor seperti kemudahan penggunaan (*usability*), kenyamanan, dan kualitas informasi. Perancangan dan evaluasi UX berfokus pada penentuan perasaan pengguna terhadap sebuah sistem, dengan tujuan akhir agar mereka terus menggunakan aplikasi tersebut dan merekomendasikannya kepada orang lain. [5].

## 2.3 User Flow

*User flow* atau alur pengguna merupakan alur pergerakan, yang umumnya divisualisasikan dalam bentuk diagram, yang dilalui oleh pengguna sejak pertama kali menggunakan sistem hingga langkah terakhir. Pemetaan setiap langkah pengguna dari titik masuk (*entry point*) hingga interaksi akhir ini sangat berguna untuk membantu desainer dalam mengantisipasi pola kognitif pengguna saat berinteraksi dengan produk. Sebagai bagian penting dari perancangan user experience (UX) yang baik, perancangan dan analisis user flow memfasilitasi proses evaluasi agar sistem yang didesain menjadi lebih optimal dan interaktif bagi pengguna [6].

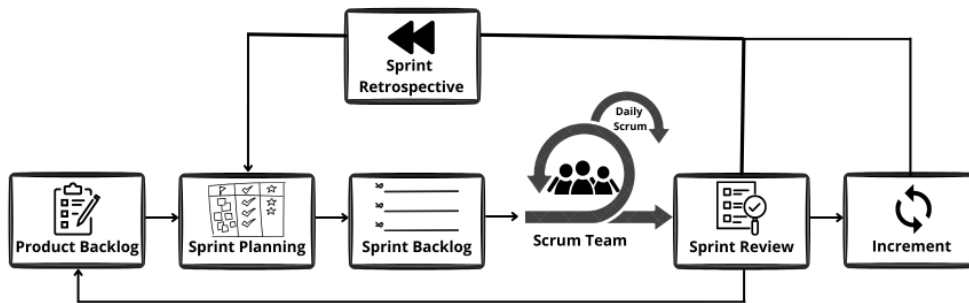
## 2.4 Integrasi Scrum dalam Pengembangan UI/UX

Integrasi *Scrum* dalam pengembangan UI/UX bertujuan untuk menyelaraskan proses rekayasa perangkat lunak yang berfokus pada fungsionalitas dengan desain yang berpusat pada kebutuhan pengguna (*User-Centered Design*) [7]. Melalui pendekatan iteratif, desainer dan pengembang dapat berkolaborasi untuk menguji serta menyempurnakan antarmuka berdasarkan umpan balik pengguna yang diperoleh lebih awal. Berikut adalah beberapa manfaat utama dari penerapan kerangka kerja *Agile* seperti *Scrum* dalam pengembangan UI/UX:

1. Iterasi Cepat: Pengiriman potongan fitur secara berkala memungkinkan tim untuk mendapatkan umpan balik dari pengguna sesegera mungkin sehingga perbaikan usability dapat dilakukan dengan cepat.
2. Kolaborasi Tim: Kerjasama yang erat antara desainer UX dan pengembang membantu dalam membangun pemahaman bersama mengenai visi produk dan meminimalkan konflik prioritas di antara berbagai disiplin ilmu.
3. Sinkronisasi Proses: Penggunaan aktivitas desain yang dilakukan secara paralel atau dalam siklus yang selaras memungkinkan evaluasi usability terintegrasi ke dalam rutinitas pengembangan teknis.

### 3. Metodologi

Metodologi yang diaplikasikan dalam studi ini adalah *Scrum*. Kerangka kerja Scrum digunakan untuk menyelesaikan masalah adaptif yang rumit secara produktif dan kreatif, sehingga mampu menciptakan perangkat lunak atau produk bernilai tinggi [8].



Gambar 1. Kerangka Kerja *Scrum* Model

Berdasarkan kerangka kerja Scrum pada Gambar 1, tim pada penelitian ini melaksanakan proyek selama total 16 minggu. Untuk menjaga ritme kerja yang efisien, periode tersebut dibagi menjadi 8 siklus *sprint*, di mana setiap *sprint* secara konsisten berdurasi 2 minggu. Seluruh pengelolaan alur kerja, pelacakan *sprint*, dan penugasan *Product Backlog* dikelola secara terpusat menggunakan *tools* manajemen proyek Google Sheets, sedangkan desain antarmuka dan interaksi purwarupa (*prototype*) dikembangkan menggunakan Figma. Komunikasi harian dilakukan melalui *daily scrum* untuk membahas hambatan dan progres kepada *stakeholder*.

#### 3.1 *Pra-Scrum*

##### 3.1.1 Prinsip-prinsip *Scrum Team*

Scrum memiliki prinsip-prinsip yang harus diperhatikan oleh *Scrum Team* diantaranya adalah (1) mengutamakan kepuasan pelanggan, (2) keterbukaan terhadap perubahan, (3) menghadirkan dampak kerja yang teratur, (4) pengembang dan pelanggan untuk bekerja sama, (5) memberikan motivasi pribadi kepada anggota tim, (6) menciptakan cara yang efisien serta efektif untuk mengumpulkan informasi, (7) memprioritaskan kemajuan proyek, (8) menjaga kesinambungan hubungan antara sponsor, pengembang, dan pengguna, (9) membayar lebih memperhatikan masalah teknis, (10) membuat segala sesuatunya sesederhana mungkin, (11) merancang arsitektur, persyaratan, dan rencana tim Anda, dan (12) berusaha untuk melakukan pekerjaan secara efisien dan melakukannya secara teratur [8]. Dengan adanya prinsip-prinsip ini membantu memastikan bahwa proyek *Agile* tetap fokus pada memberikan nilai, responsif terhadap perubahan, dan memberdayakan tim untuk bekerja secara efisien dan efektif.

### 3.1.2 Nilai Manifesto *Agile*

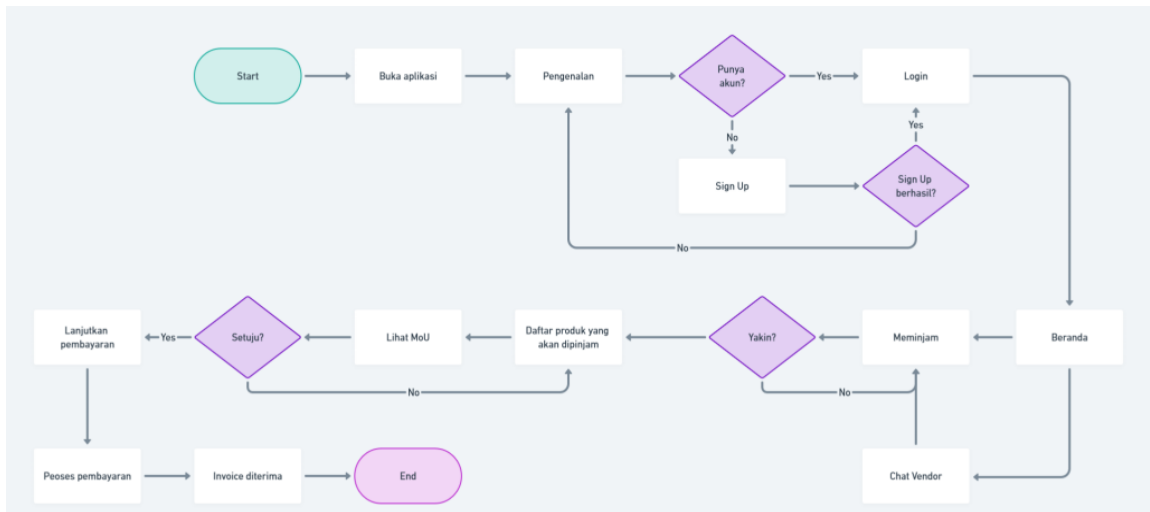
Nilai dalam Manifesto *Agile* berfungsi sebagai panduan fundamental yang memberikan fleksibilitas lebih tinggi dibandingkan pendekatan tradisional berbasis rencana (*plan-driven*). Pengembangan tangkas mengutamakan kolaborasi intensif antara pelanggan dan pengembang serta mendorong pembentukan tim yang mampu mengatur dirinya sendiri (*self-organizing*) [9]. Melalui pendekatan ini, tim dapat fokus pada pengiriman fitur fungsional secara iteratif untuk mendapatkan umpan balik yang cepat, sehingga perubahan kebutuhan dapat diakomodasi pada setiap fase siklus pengembangan demi memastikan kepuasan pelanggan.

### 3.1.3 *Wireframe*

*Wireframe* atau *mockup* merupakan representasi visual awal dari antarmuka sebuah situs web maupun aplikasi, yang umumnya dirancang secara manual dengan memanfaatkan perangkat lunak pengolah gambar [14]. *Wireframe* membantu tim desain, pengembang, dan pemangku kepentingan untuk memahami tata letak dasar dan alur navigasi sebelum masuk ke tahap desain yang lebih rinci. Fungsi dan manfaat *wireframe* meliputi visualisasi struktur, perencanaan konten, pengujian alur navigasi, komunikasi dengan *stakeholders*, serta efisiensi dan penghematan waktu.

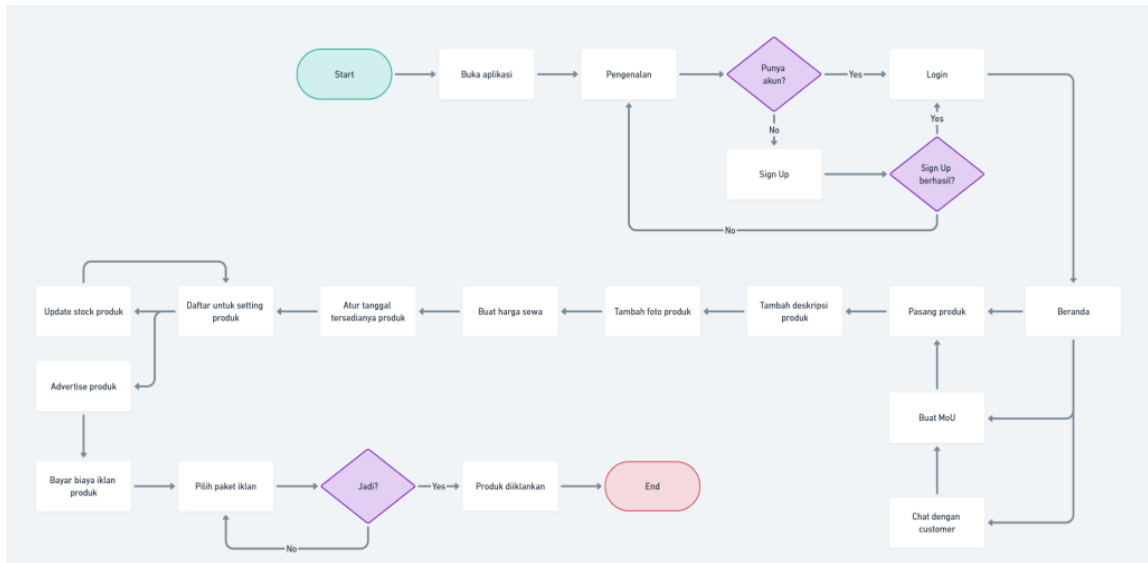
### 3.1.4 *User Flow*

*User flow* dibagi menjadi dua bagian, *user flow* untuk *customer* dan *user flow* untuk *vendor*. *User flow* untuk *customer* digunakan sebagai pedoman atau langkah-langkah bagi *customer* dalam menggunakan aplikasi Rent2Go yang akan ditampilkan pada Gambar 2



Gambar 2. *User Flow Customer*

Setelah *user flow* untuk *customer* telah diberikan tentunya dibutuhkan *user flow* untuk penjual/vendor, *user flow* untuk vendor dimaksudkan sebagai langkah-langkah/pedoman vendor dalam memasang barang yang akan disewakan kepada customer, berikut *user flow* untuk vendor yang akan ditampilkan pada Gambar 3.



Gambar 3. User Flow Vendor

### 3.2 Proses Implementasi Scrum

Implementasi *Scrum* dalam pengembangan *flow* UI/UX untuk platform *e-commerce* RENT2GO melibatkan beberapa tahapan penting. Berikut adalah tahapan dalam proses implementasi *scrum*:

#### 1) Product Backlog

*Product Backlog* adalah daftar fitur, perubahan, perbaikan, dan elemen lainnya yang perlu dikerjakan untuk pengembangan produk. Elemen-elemen dalam *Product Backlog* diurutkan berdasarkan prioritas dan kepentingannya bagi pengembangan produk. Terkait hal tersebut, *Product Owner* memegang tanggung jawab untuk terus berinteraksi dengan tim pengembang mengenai visi dan prioritas agar produk yang dihasilkan memberikan nilai bisnis secara maksimal [10]. Sebelum menyusun daftar fitur ke dalam *Product Backlog*, pengumpulan kebutuhan awal (*user requirements*) dilakukan secara empiris melalui observasi dan wawancara semi-terstruktur dengan pemangku kepentingan utama, yaitu perwakilan penyelenggara acara (*event organizer*) selaku penyewa dan vendor persewaan barang. Tahap ini bertujuan untuk menggali kendala dalam transaksi sewa-menyewa konvensional serta memetakan fungsionalitas yang paling mendesak. Data dan *insight* yang didapatkan dari wawancara tersebut kemudian diekstraksi dan diubah menjadi item-item *Product Backlog* yang konkrit. Isi dari *Product Backlog* mencakup berbagai elemen seperti fitur baru yang ingin ditambahkan ke produk, modifikasi atau peningkatan dari fitur yang sudah ada, perbaikan masalah atau bug yang ditemukan, serta aspek lain seperti kebutuhan teknis, penelitian, atau eksperimen. Setiap elemen dalam *Product Backlog* memiliki tingkat prioritas yang ditentukan oleh *Product Owner*, yang menentukan urutan pengerjaan elemen-elemen tersebut. Elemen dengan prioritas tinggi adalah yang paling penting dan mendesak untuk pengembangan produk. Berikut pada Tabel 1 *product backlog* sistem Rent2Go.

Untuk memastikan kualitas sistem, setiap item di dalam *Product Backlog* memiliki kriteria penerimaan (*Definition of Done / DoD*) yang mengikat. Sebuah item *backlog* desain UI/UX dinyatakan selesai (*Done*) dan siap diserahkan apabila memenuhi tiga parameter berikut: (1) rancangan visual dan elemen *mockup* telah diselesaikan secara utuh di Figma sesuai spesifikasi *user requirements*, (2) simulasi alur navigasi (*user flow*) pada *prototype* dapat dijalankan tanpa adanya *error* atau *broken link*, dan (3) hasil desain telah

didemonstrasikan, dievaluasi, dan mendapatkan persetujuan akhir (*approval*) secara fungsional maupun visual dari *Product Owner*.

Tabel 1. *Product Backlog*

| No | Item                            | Deskripsi   | Prioritas |
|----|---------------------------------|---|-----------|
| 1. | Design halaman <i>Login</i>     | Membuat halaman <i>login</i> , <i>halaman Sign Up</i> , dan membuat halaman untuk <i>reset password</i>           | Tinggi    |
| 2. | Design halaman utama            | Membuat halaman utama aplikasi untuk vendor dan <i>customer</i>   | Tinggi    |
| 3. | Design halaman transaksi        | Membuat halaman transaksi, halaman <i>chat</i> kepada vendor, serta membuat halaman untuk MoU peminjaman          | Tinggi    |
| 4. | Design keranjang barang         | Membuat halaman untuk membuat keranjang barang  | Sedang    |
| 5. | Design manajemen akun           | Membuat halaman untuk mengatur informasi pribadi ke dalam <i>profile account</i> untuk <i>customer</i> dan vendor | Tinggi    |
| 6. | Design halaman <i>Check Out</i> | Membuat halaman untuk <i>Checkout</i> barang  | Tinggi    |
| 7. | Design Halaman Pembayaran       | Membuat halaman untuk invoice dan pembayaran  | Rendah    |

## 2) *Sprint Planning*

*Sprint Planning* merupakan fase kolaboratif yang menandai dimulainya setiap iterasi, di mana tim menetapkan tujuan *sprint* serta memilih *user stories* dari *backlog* untuk dikerjakan. Aktivitas ini berperan penting dalam menyinkronkan prioritas bisnis dengan kapasitas teknis tim guna memastikan rencana kerja yang detail dan terukur. Melalui perencanaan yang terstruktur pada awal siklus, tim dapat mengidentifikasi hambatan potensial sedini mungkin dan meningkatkan peluang keberhasilan dalam mencapai target fungsionalitas yang telah ditetapkan [11].

## 3) *Sprint Backlog*

*Sprint Backlog* adalah proses pemenuhan daftar kebutuhan sistem yang telah didefinisikan dan disepakati pada tahap *Product Backlog* [12]. Tim *scrum* memilih item-item dari *Product Backlog* sesuai dengan tingkat prioritasnya untuk diselesaikan dalam *sprint* dan memecah menjadi tugas-tugas yang lebih kecil untuk dikerjakan.

## 4) *Sprint Running*

Pelaksanaan *Sprint* adalah tahap di mana tim *scrum* bekerja secara kolaboratif untuk menyelesaikan *backlog* item yang telah disepakati dan dipilih selama *Sprint Planning*. Selama pelaksanaan *sprint*, diadakan pertemuan harian yang disebut *daily scrum*. Pada kegiatan ini, setiap anggota tim melaporkan progres pekerjaan yang telah diselesaikan, rencana pekerjaan selanjutnya, serta mendiskusikan hambatan atau kendala yang dihadapi agar dapat segera diselesaikan bersama [18]. Pada proses ini pula dilakukan perancangan *prototype*, mulai dari pembuatan *wireframe* hingga *mockup* interaktif menggunakan aplikasi Figma, yang berfungsi sebagai gambaran antarmuka (*User Interface*) dan pengalaman pengguna (*User Experience*) sebelum sistem dikembangkan lebih lanjut.

## 5) *Sprint Review*

*Sprint Review* adalah pertemuan yang diadakan pada akhir setiap *sprint* untuk meninjau *increment* dan menilai hasil produk (*product backlog*) yang telah diselesaikan [19]. Pada tahap ini, tim *scrum*

mempresentasikan hasil kerja kepada *stakeholders* untuk diperiksa, sehingga *stakeholders* dapat melihat progres dan memberikan umpan balik secara langsung. *Sprint review* menjadi kesempatan penting bagi tim pengembang dan *stakeholders* untuk berkolaborasi dan menentukan perbaikan apa saja yang perlu dilakukan untuk mengarahkan produk di *sprint* berikutnya [19].

#### 6) *Sprint Retrospective*

*Sprint Retrospective* merupakan tahap evaluasi yang dilaksanakan setelah proses *sprint review* selesai dilakukan. Pada tahap ini, selain meninjau efektivitas pengerjaan, tim *Scrum* juga melakukan pemilihan daftar tugas yang akan dijalankan pada *sprint planning* selanjutnya sesuai dengan kesepakatan awal proyek [13]. Melalui tinjauan pengalaman selama *sprint* dan pemilihan tugas lanjutan ini, tim dapat mengidentifikasi kekuatan dan kelemahan guna merencanakan tindakan perbaikan. Hal ini pada akhirnya akan membantu tim terus berkembang dan meningkatkan kualitas pekerjaan mereka. Semua tahapan *scrum* yang dilakukan terangkum pada Tabel 2.

Tabel 2. Rangkuman Tahapan *Scrum*

| No | Tahap Scrum                  | Deskripsi  | Penanggung Jawab          | Output   |
|----|------------------------------|--|---------------------------|--|
| 1  | Product Backlog              | Daftar fitur, perubahan, perbaikan, dan elemen lainnya yang perlu dikerjakan untuk pengembangan produk. Setiap item diurutkan berdasarkan prioritas oleh Product Owner.          | Product Owner             | Daftar item berprioritas (fitur baru, modifikasi, perbaikan bug, kebutuhan teknis) |
| 2  | Sprint Planning              | Tahap awal setiap sprint di mana tim Scrum berkumpul untuk merencanakan item-item dari Product Backlog yang akan dikerjakan, serta membagi tugas dan estimasi waktu.             | Scrum Team                | Rencana Sprint, pembagian tugas, dan estimasi waktu penyelesaian                   |
| 3  | Sprint Backlog               | Proses pemenuhan daftar kebutuhan sistem yang telah disepakati. Tim memilih item dari Product Backlog sesuai prioritas dan memecahnya menjadi tugas-tugas kecil.                 | Development Team          | Daftar tugas spesifik yang siap dikerjakan dalam satu sprint                       |
| 4  | Sprint Running (Daily Scrum) | Pelaksanaan sprint di mana tim bekerja menyelesaikan backlog item. Diadakan daily scrum setiap hari untuk melaporkan progres, rencana selanjutnya, dan hambatan yang dihadapi.   | Development Team          | Prototype/mockup (wireframe di Figma), pembaruan progres harian, solusi hambatan   |
| 5  | Sprint Review                | Pertemuan di akhir sprint untuk menilai hasil kerja yang dicapai. Tim mempresentasikan increment produk kepada stakeholders untuk mendapatkan umpan balik.                       | Scrum Team & Stakeholders | Increment produk yang telah didemonstrasikan, umpan balik stakeholders             |
| 6  | Sprint Retrospective         | Tahap evaluasi setelah sprint review. Tim meninjau efektivitas sprint, mengidentifikasi kekuatan dan kelemahan, serta menetapkan daftar tugas untuk sprint planning selanjutnya. | Scrum Team                | Catatan evaluasi, rencana perbaikan, daftar tugas sprint berikutnya                |

### 3.3 Metode Pengujian Kebergunaan (SUS)

Tahap akhir evaluasi dilakukan melalui pengujian kebergunaan (*usability testing*) untuk mengukur sejauh mana purwarupa RENT2GO dapat diterima oleh pengguna. Instrumen yang digunakan adalah *System Usability Scale* (SUS), yaitu sebuah kuesioner yang terdiri dari 10 butir pernyataan dengan pengukuran menggunakan skala *Likert* 1 hingga 5. Pengujian ini melibatkan 24 orang responden. Proses pengumpulan data dilakukan secara digital melalui kuesioner yang mencakup 10 butir pernyataan pada Tabel 3.

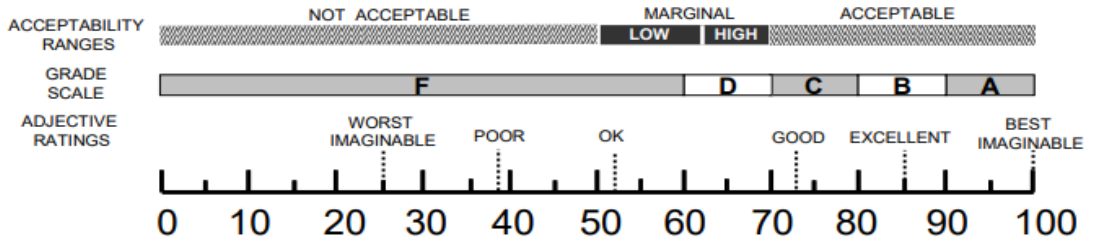
Tabel 3. Butir Instrumen *System Usability Scale* (SUS)

| No | Butir Pernyataan  |
|----|---|
| 1  | Saya berpikir akan menggunakan sistem ini lagi.                                     |
| 2  | Saya merasa sistem ini rumit untuk digunakan.                                       |
| 3  | Saya merasa sistem ini mudah digunakan.   |
| 4  | Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini. |
| 5  | Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.                      |
| 6  | Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.                    |
| 7  | Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.      |
| 8  | Saya merasa sistem ini membingungkan.   |
| 9  | Saya merasa tidak ada hambatan dalam menggunakan sistem ini.                        |
| 10 | Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.         |

Data kuesioner yang terkumpul kemudian diolah menggunakan rumus perhitungan standar SUS. Perhitungan ini membedakan perlakuan antara pernyataan positif (bernomor ganjil) dan pernyataan negatif (bernomor genap) dengan aturan sebagai berikut: skor pada pernyataan ganjil dikurangi 1, sedangkan pada pernyataan genap, nilai 5 dikurangi dengan skor jawaban responden.

Seluruh hasil perhitungan dari tiap butir pernyataan tersebut kemudian dijumlahkan dan dikalikan dengan 2,5 untuk mengkonversi rentang skor mentah ke dalam skala standar 0 hingga 100. Skor rata-rata SUS diperoleh dengan menjumlahkan keseluruhan skor individu, lalu membaginya dengan total responden. Hasil akhir pengukuran ini kemudian dipetakan ke dalam parameter interpretasi yang mencakup tingkat penerimaan sistem (*Acceptability Range*), skala peringkat (*Grade Scale*), dan penilaian adjektiva (*Adjective Rating*). Panduan interpretasi skor SUS secara visual diilustrasikan pada Gambar 4[20].

Merujuk pada Gambar 4, skala interpretasi tersebut digunakan untuk menentukan tingkat kelayakan dan kebergunaan purwarupa yang telah dibangun. Melalui pemetaan ini, peneliti dapat secara objektif menyimpulkan apakah desain antarmuka RENT2GO sudah memenuhi standar minimal penerimaan pengguna atau masih membutuhkan evaluasi lanjutan pada iterasi pengembangan berikutnya.



Gambar 4. Parameter Penilaian dan Interpretasi Skor SUS

#### 4. Hasil dan Pembahasan

##### 4.1 Efektifitas Penggunaan Agile Scrum

Pada tahap uji efektivitas ini, metode yang digunakan adalah wawancara terstruktur yang dipadukan dengan penilaian kuantitatif menggunakan skala *Likert* (1 = Sangat Tidak Setuju hingga 5 = Sangat Setuju) untuk mengukur tingkat kesepakatan secara objektif [15]. Evaluasi ini melibatkan 4 orang responden yang bertindak secara spesifik sebagai evaluator internal proyek (tim pengembang) untuk menilai kualitas proses manajerial selama *sprint* berlangsung. Daftar pertanyaan beserta hasil rata-rata skor dan persentase kesepakatan disajikan pada Tabel 4.

Tabel 4. Hasil Penilaian Wawancara Evaluator Internal

| No.                   | Daftar Pertanyaan Wawancara   | Rata-rata Skor (1-5) | Persentase Kesepakatan |
|-----------------------|---|----------------------|------------------------|
| 1                     | Apakah metode Agile Scrum memudahkan Anda dalam menyelesaikan proyek?                                       | 4.50                 | 90%                    |
| 2                     | Menurut Anda, apakah menggunakan metode Agile Scrum memfasilitasi kolaborasi?                               | 4.75                 | 95%                    |
| 3                     | Menurut Anda, apakah Daily Scrum meningkatkan komunikasi antar tim?   | 4.25                 | 85%                    |
| 4                     | Apakah Sprint Planning dan Sprint Backlog membantu Anda mengerjakan tugas sesuai prioritas yang diperlukan? | 4.50                 | 90%                    |
| 5                     | Apakah Sprint Review membantu Anda meningkatkan kualitas hasil pekerjaan?                                   | 4.75                 | 95%                    |
| 6                     | Apakah Sprint Retrospective membuat komunikasi antar tim menjadi lebih terbuka dan mempererat hubungan tim? | 4.25                 | 85%                    |
| Rata-Rata Keseluruhan |   | 4.50                 | 90%                    |

Hasil wawancara dan kuantifikasi data menunjukkan bahwa secara luas responden setuju (dengan rata-rata persentase kesepakatan 90%) bahwa penggunaan metode *Agile Scrum* sangat efektif dalam menyelesaikan proyek RENT2GO. *Agile Scrum* dinilai membantu mengatur dan menyelesaikan tugas secara terstruktur, memungkinkan tim merespons perubahan dengan lebih fleksibel. Kolaborasi dinilai lebih lancar karena

adanya ritme kerja yang teratur seperti *Daily Scrum*, yang terbukti mampu meningkatkan komunikasi dan koordinasi harian. *Sprint Planning* dan *Sprint Backlog* juga terbukti bermanfaat (skor 4.50) karena secara presisi membantu tim fokus pada tugas dengan prioritas tertinggi, sehingga mendongkrak produktivitas.

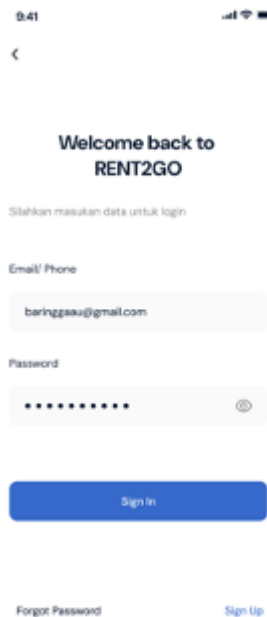
Selanjutnya, *Sprint Review* dinilai sangat krusial (skor tertinggi 4.75) dalam meningkatkan kualitas hasil pekerjaan, karena memberikan ruang untuk memperbaiki produk secara berkala berdasarkan umpan balik langsung. *Sprint Retrospective* juga diakui sebagai waktu yang efektif untuk refleksi timbal balik, memperkuat komunikasi, dan menjaga kekompakan tim untuk peningkatan kinerja di iterasi mendatang.

Selain data persepsi di atas, efektivitas *Scrum* juga dibuktikan melalui pengukuran metrik operasional proyek secara objektif selama 8 *sprint* (16 minggu). Tingkat penyelesaian *sprint* (*Sprint Completion Rate*) menembus angka 93,7%, di mana dari total 32 item *Product Backlog* utama, 30 item berhasil memenuhi kriteria *Definition of Done* (DoD) tepat waktu. Keunggulan adaptasi kerangka kerja ini terlihat dari jumlah *feedback* yang berhasil diimplementasikan, yakni 21 dari 24 *feedback* fungsional (87,5%) berhasil diterapkan pada *sprint* berikutnya. Waktu respons terhadap perubahan (*Time to Pivot*) rata-rata hanya memakan waktu 3 hari sejak *feedback* diberikan hingga masuk ke *Sprint Backlog* siklus selanjutnya. Dengan demikian, penerapan *Agile Scrum* dalam RENT2GO tidak hanya memfasilitasi pengelolaan proyek yang kolaboratif, tetapi juga menghasilkan metrik efisiensi yang sangat terukur.

## 4.2 Implementasi

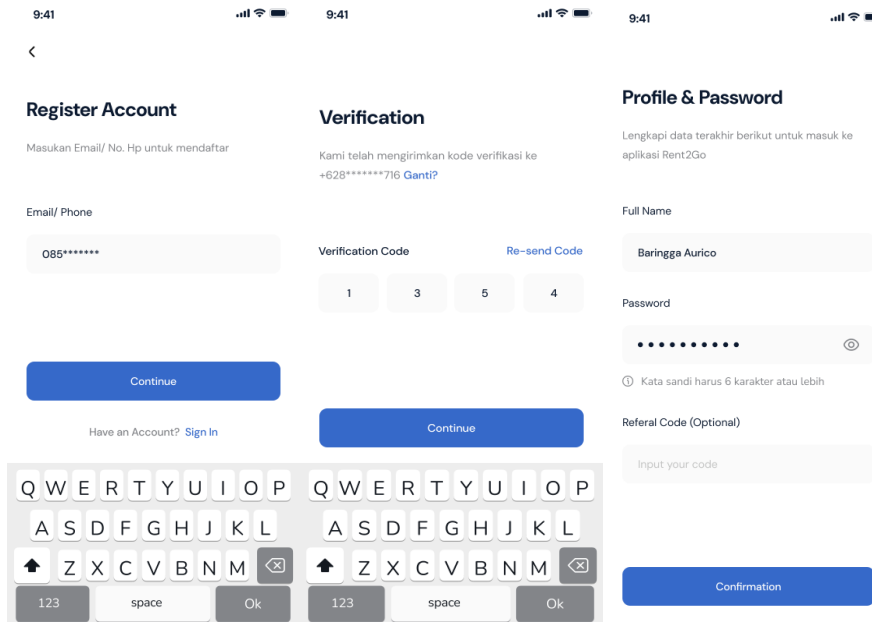
Dengan mengadopsi metode *Scrum*, tim desainer memperoleh umpan balik dengan cepat, sehingga memungkinkan proses desain berlangsung secara efektif dan efisien. Adapun perancangan antarmuka pengguna (*user interface*) yang dihasilkan dalam tahap ini bertujuan untuk memberikan gambaran serta informasi yang jelas mengenai fitur dan tampilan sistem yang akan dikembangkan [16]. Berikut merupakan hasil implementasi desain.

### 4.2.1 Design Halaman *Login*



Gambar 5. Tampilan Halaman *Login*

Sesuai dengan *product backlog*, tim UI/UX membuat halaman *login*, halaman *sign up*, dan halaman reset *password*. Pada halaman *login* di Gambar 5, pengguna dapat menginput *username* dan *password* akun yang sudah dimiliki. Halaman ini juga menyediakan opsi untuk mengingatkan pengguna jika mereka lupa *password*, serta tombol untuk mendaftarkan akun baru. Setelah mengisi informasi yang diperlukan, pengguna dapat mengklik tombol "*Login*" untuk masuk ke halaman utama aplikasi.



Gambar 6. Tampilan Halaman *Sign Up*

Pada halaman *Sign Up* pada Gambar 6, pengguna yang belum memiliki akun dapat mendaftarkan dengan mengisi informasi yang diperlukan seperti email, nomor telepon, kode verifikasi, nama, *password*, dan kode *referral*. Setelah semua informasi diisi dengan benar, pengguna dapat mengklik tombol "*Confirmation*" untuk menyelesaikan proses pendaftaran.

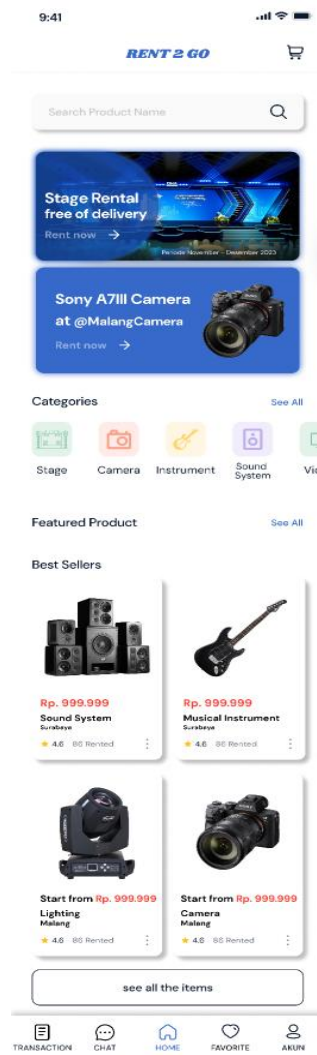


Gambar 7 Tampilan Halaman Reset *Password*

Pada halaman reset *password*, pengguna yang lupa sandi mereka dapat memulihkan akses ke akun mereka. Halaman ini meminta pengguna untuk memasukkan email atau nomor telepon terdaftar untuk menerima tautan reset *password*. Setelah menerima email atau pesan masuk, pengguna dapat mengikuti tautan tersebut untuk membuat sandi baru dan mengakses kembali akun mereka.

#### 4.2.2 Design halaman Utama

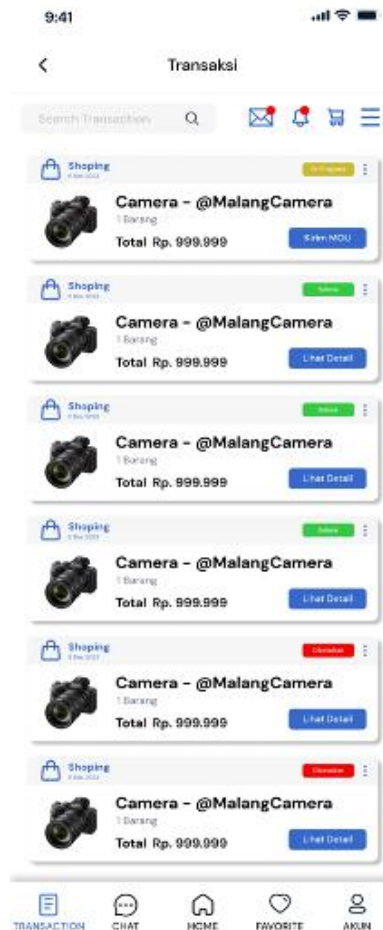
Halaman utama (pada Gambar 8) aplikasi dirancang untuk vendor dan *customer*. Pada halaman ini, pengguna dapat melihat *highlight* produk rental, kategori rental, dan *feature products*. Halaman ini dirancang untuk memudahkan pengguna dalam menavigasi dan menemukan produk yang mereka butuhkan. Terdapat fitur pencarian produk yang memungkinkan pengguna untuk mencari produk spesifik dengan cepat. Selain itu, pengguna juga dapat memeriksa keranjang belanja mereka dan melanjutkan ke proses *checkout* dari halaman ini.



Gambar 8. Tampilan Halaman Utama

### 4.2.3 Desain Halaman Transaksi

Pada halaman transaksi (Gambar 9), pengguna dapat melihat daftar transaksi yang telah mereka lakukan. Halaman ini menampilkan rincian seperti status, barang, dan total pembayaran. Pengguna juga dapat melihat tindakan lebih lanjut tentang setiap transaksi yang dilakukan berdasarkan status.



Gambar 9 Tampilan Halaman Transaksi

Pada halaman chat kepada vendor (Gambar 10), pengguna dapat berkomunikasi langsung dengan vendor terkait produk yang mereka beli atau tanyakan. Halaman ini menyediakan fitur pesan instan di mana pengguna dapat mengajukan pertanyaan, meminta penjelasan lebih lanjut tentang produk, atau menyelesaikan masalah terkait pembelian.

Pada halaman MoU peminjaman (Gambar 11), pengguna dapat mengakses, menyetujui, dan mengajukan modifikasi MoU terkait peminjaman barang atau jasa. Halaman ini menampilkan dokumen MoU yang dapat dibaca oleh pengguna sebelum melanjutkan proses peminjaman.



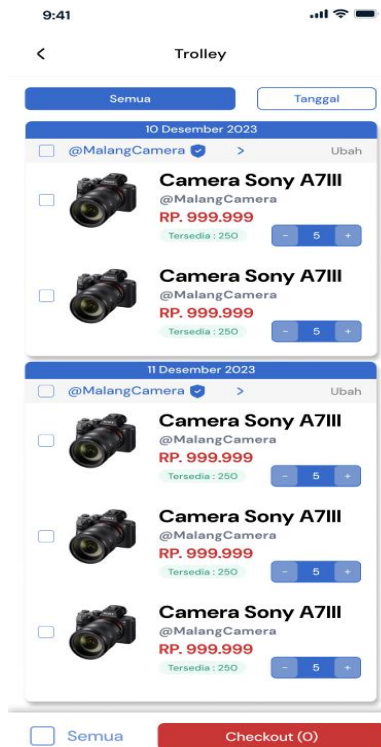
Gambar 10. Tampilan Halaman *Chat* kepada Vendor



Gambar 11. Tampilan MoU Peminjaman

#### 4.2.4 Desain Keranjang Barang

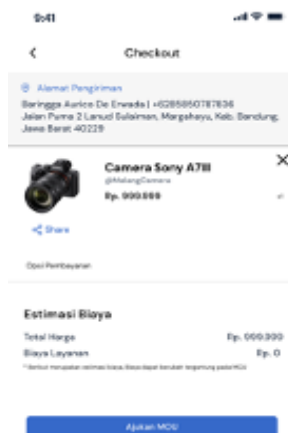
Pada halaman keranjang barang (Gambar 12), pengguna dapat melihat daftar barang yang telah mereka tambahkan dari katalog. Pengguna dapat mengedit jumlah barang, menghapus barang dari keranjang, serta melihat subtotal dari setiap barang dan total keseluruhan. Selain itu, pengguna juga dapat memilih checkout dari tanggal tertentu. Setelah memastikan semua rincian sudah benar, pengguna dapat melanjutkan ke halaman *checkout*.



Gambar 12. Tampilan Halaman Keranjang Barang

#### 4.2.5 Desain Halaman *Checkout*

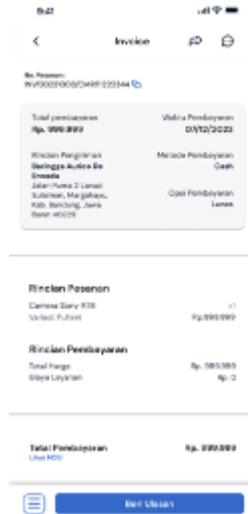
Pada halaman *checkout* (Gambar 13), pengguna dapat melihat rincian alamat pengiriman, rincian pesanan, dan rincian biaya sebelum mengajukan MoU. Setelah memastikan semua rincian sudah benar, pengguna dapat melanjutkan ke halaman MoU.



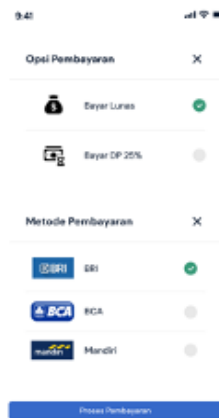
Gambar 13. Tampilan *Checkout* Barang

#### 4.2.6 Desain Halaman Pembayaran

Pada halaman *invoice* (Gambar 14), pengguna dapat melihat rincian pengiriman, rincian pesanan, dan rincian pembayaran. Sementara itu, pada halaman pembayaran (Gambar 15), pengguna dapat memilih opsi dan metode pembayaran yang diinginkan dan melanjutkan proses pembayaran hingga selesai dengan mengklik tombol "Proses Pembayaran".



Gambar 14. Tampilan Halaman *Invoice*



Gambar 15. Tampilan Halaman Pembayaran

#### 4.3 Analisis Pengujian Kebergunaan (SUS)

Setelah tahap implementasi desain selesai, dilakukan pengujian *usability* untuk memvalidasi apakah hasil rancangan pada sub-bab 4.2 telah memenuhi ekspektasi pengguna. Pengujian ini melibatkan 24 responden aktual yang mewakili peran penyewa (*customer*) dan vendor. Berdasarkan hasil pengolahan data kuesioner, rincian skor individu dari ke-24 responden disajikan pada Tabel 5.

Tabel 5. Rekapitulasi Skor SUS

| No | R   | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | $\Sigma$ | Nilai |
|----|-----|----|----|----|----|----|----|----|----|----|-----|----------|-------|
| 1  | R1  | 5  | 1  | 5  | 3  | 5  | 1  | 5  | 1  | 5  | 3   | 36       | 90    |
| 2  | R2  | 4  | 3  | 5  | 4  | 2  | 4  | 5  | 3  | 4  | 2   | 24       | 60    |
| 3  | R3  | 5  | 2  | 4  | 2  | 4  | 2  | 5  | 2  | 4  | 1   | 33       | 82.5  |
| 4  | R4  | 5  | 1  | 4  | 2  | 5  | 1  | 4  | 2  | 5  | 1   | 36       | 90    |
| 5  | R5  | 5  | 3  | 5  | 4  | 5  | 3  | 5  | 3  | 5  | 5   | 27       | 67.5  |
| 6  | R6  | 3  | 2  | 4  | 2  | 4  | 3  | 3  | 4  | 4  | 3   | 24       | 60    |
| 7  | R7  | 5  | 1  | 5  | 2  | 4  | 2  | 5  | 1  | 4  | 2   | 35       | 87.5  |
| 8  | R8  | 4  | 3  | 3  | 2  | 4  | 2  | 4  | 1  | 4  | 2   | 29       | 72.5  |
| 9  | R9  | 4  | 2  | 5  | 1  | 5  | 2  | 4  | 2  | 5  | 1   | 35       | 87.5  |
| 10 | R10 | 4  | 2  | 5  | 2  | 5  | 1  | 4  | 2  | 4  | 2   | 33       | 82.5  |
| 11 | R11 | 4  | 1  | 5  | 5  | 5  | 1  | 5  | 1  | 4  | 1   | 34       | 85    |
| 12 | R12 | 4  | 1  | 5  | 1  | 5  | 2  | 4  | 1  | 5  | 2   | 36       | 90    |
| 13 | R13 | 4  | 2  | 5  | 2  | 5  | 1  | 4  | 1  | 5  | 2   | 35       | 87.5  |
| 14 | R14 | 5  | 1  | 5  | 1  | 5  | 1  | 1  | 1  | 5  | 1   | 36       | 90    |
| 15 | R15 | 5  | 2  | 4  | 1  | 5  | 2  | 4  | 1  | 5  | 2   | 35       | 87.5  |
| 16 | R16 | 5  | 2  | 4  | 1  | 5  | 2  | 4  | 2  | 5  | 2   | 34       | 85    |
| 17 | R17 | 4  | 2  | 5  | 1  | 4  | 1  | 4  | 1  | 4  | 2   | 34       | 85    |
| 18 | R18 | 5  | 2  | 4  | 1  | 4  | 2  | 5  | 2  | 5  | 1   | 35       | 87.5  |
| 19 | R19 | 4  | 2  | 4  | 2  | 4  | 2  | 4  | 2  | 4  | 2   | 30       | 75    |
| 20 | R20 | 4  | 2  | 5  | 2  | 4  | 1  | 4  | 2  | 5  | 1   | 34       | 85    |
| 21 | R21 | 4  | 2  | 4  | 2  | 5  | 1  | 4  | 1  | 5  | 2   | 34       | 85    |
| 22 | R22 | 5  | 3  | 5  | 4  | 4  | 3  | 4  | 3  | 5  | 4   | 26       | 65    |
| 23 | R23 | 4  | 3  | 5  | 3  | 5  | 3  | 4  | 2  | 5  | 4   | 28       | 70    |
| 24 | R24 | 1  | 3  | 2  | 2  | 2  | 2  | 4  | 3  | 1  | 4   | 16       | 40    |

Berdasarkan kalkulasi dari keseluruhan data, diperoleh skor rata-rata SUS sebesar 79,06. Merujuk pada parameter interpretasi skor SUS [20], skor tersebut menempatkan purwarupa RENT2GO pada kategori sebagai berikut:

1. *Acceptability Range: Acceptable* (Dapat diterima dengan baik oleh pengguna).
2. *Grade Scale: Grade B* (Menunjukkan kualitas desain antarmuka yang tinggi).
3. *Adjective Rating: Good* (Baik).

Meskipun terdapat sedikit variasi penilaian yang mengindikasikan perlunya proses adaptasi awal bagi sebagian pengguna awam, secara keseluruhan perolehan skor rata-rata 79,06 berada jauh di atas ambang batas minimal kelayakan (skor 68). Tingginya perolehan nilai ini membuktikan bahwa proses iterasi desain dalam *Scrum* berhasil meminimalisir hambatan navigasi sejak dini. Responden secara kumulatif memberikan nilai tinggi pada aspek kemudahan dipelajari (*learnability*) dan kepercayaan diri saat menggunakan sistem, yang menegaskan bahwa antarmuka yang dihasilkan bersifat intuitif dan berhasil mengakomodasi alur bisnis persewaan acara yang kompleks.

#### 4.4 Analisis Dampak Scrum pada Aspek Kecepatan dan Biaya

Penerapan kerangka kerja *Scrum* terbukti memberikan dampak positif yang signifikan terhadap efisiensi sumber daya selama pengembangan proyek:

1. Kecepatan (Efisiensi Waktu): Metrik *Time to Pivot* yang mencapai rata-rata 3 hari menunjukkan kecepatan luar biasa dari tim dalam merespons umpan balik. Dalam siklus pengembangan tradisional, revisi desain berskala besar seringkali baru terdeteksi di akhir fase. Namun, melalui mekanisme *Sprint Review* setiap 2 minggu pada RENT2GO, hambatan dan anomali desain dapat dideteksi dan diperbaiki secara instan. Ditambah dengan pencapaian *Sprint Completion Rate* sebesar 93,7%, jadwal rilis proyek tetap terjaga tanpa adanya penundaan kumulatif.
2. Biaya (Efisiensi Anggaran): Meskipun metode *Scrum* menuntut intensitas komunikasi dan pertemuan yang tinggi (seperti *Daily Scrum*), biaya produksi secara keseluruhan dapat ditekan secara signifikan. Hal ini dikarenakan minimalnya pengerjaan ulang (*rework*) yang bersifat fatal. Melalui implementasi 87,5% *feedback* fungsional secara bertahap pada setiap *sprint*, pembengkakan biaya akibat perombakan sistem di tahap akhir berhasil dihindari, mengingat setiap desain yang tidak sesuai langsung dikoreksi sebelum diteruskan menjadi kode program.

#### 4.5 Perbandingan Agile Scrum terhadap Metode Waterfall

Untuk mempertegas keunggulan metodologi yang diterapkan dalam menyelesaikan studi kasus ini, dilakukan komparasi antara hasil implementasi RENT2GO (berbasis *Scrum*) dengan karakteristik metode *Waterfall* konvensional, sebagaimana disajikan pada Tabel 6.

Tabel 6. Perbandingan Implementasi Scrum (RENT2GO) vs Waterfall

| Dimensi Perbandingan       | Agile Scrum (RENT2GO)  | Metode Waterfall  |
|----------------------------|--|---|
| Fleksibilitas Desain       | Sangat Tinggi. Perubahan desain langsung diakomodasi di <i>sprint</i> berikutnya.    | Rendah. Desain dikunci di awal ( <i>fixed</i> ), perubahan membutuhkan revisi dokumen spesifikasi secara total. |
| Keterlibatan Pengguna      | Aktif. Pengguna memberikan <i>feedback</i> setiap 2 minggu ( <i>Sprint Review</i> ). | Pasif. Pengguna umumnya hanya dilibatkan pada tahap awal pengumpulan data dan serah terima akhir.               |
| Respons terhadap Perubahan | Cepat ( <i>Time to Pivot</i> : 3 Hari).  | Lambat. Perubahan seringkali dianggap sebagai hambatan <i>deadline</i> .  |
| Kualitas Akhir             | Teruji secara empiris dan berkelanjutan (dibuktikan dengan Skor SUS: 79,06).         | Berisiko tinggi tidak sesuai dengan ekspektasi pengguna akibat minimnya uji coba di tengah fase pengembangan.   |

## 5. Kesimpulan

### 5.1 Simpulan

Penelitian ini menunjukkan bahwa penerapan metode *Scrum* dalam pengembangan UI/UX platform *e-commerce* RENT2GO mampu mendukung proses pengembangan secara lebih adaptif dan terstruktur melalui pembagian kerja selama 16 minggu. Implementasi *Scrum* membantu tim dalam menjaga koordinasi dan evaluasi menggunakan aktivitas *sprint planning*, *daily Scrum*, *sprint review*, dan *sprint retrospective*. Efektifitas dapat dilihat dari penyelesaian *sprint* yang mencapai 93,7% serta keberhasilan implementasi 87,5% dari *feedback* fungsional untuk *sprint* berikutnya. Hasil pengembangan menghasilkan rancangan UI/UX yang mencakup fitur utama seperti *login*, transaksi, keranjang belanja, *checkout*, pembayaran, Hingga MoU peminjaman. Pengujian *usability* pada metode *Scrum* ini menggunakan metode *System Usability Scale* yang ditujukan untuk 24 responden, yang memperoleh skor rata-rata 79,06 yang digolongkan dalam kategori *acceptable*, *grade B*, dan *good*. Hasil ini menunjukkan bahwa desain antarmuka yang dikembangkan memiliki kemudahan penggunaan dan navigasi aplikasi yang baik bagi pengguna.

Temuan pada penelitian ini menunjukkan bahwa pendekatan iteratif pada *Scrum* mempermudah proses penyesuaian kebutuhan pengguna selama pengembangan berlangsung, khususnya pada alur transaksi penyewaan yang melibatkan interaksi antara *customer* dan vendor. Dengan melakukan evaluasi rutin pada setiap *sprint*, perbaikan desain dapat dilakukan dengan cepat sehingga meminimalkan resiko evaluasi yang menumpuk lebih besar pada akhir pengembangan. Penelitian ini dapat menjadi referensi bagi pengembang untuk aplikasi *e-commerce* serupa dalam menerapkan *Scrum* sebagai metode pengembangan UI/UX.

### 5.2 Saran

Untuk pengembangan selanjutnya, pengujian *usability* dapat dilakukan pada jumlah responden yang lebih besar dan melibatkan pengguna dari berbagai latar belakang agar hasil evaluasi lebih representatif. Selain itu, integrasi *usability* testing pada setiap *sprint* disarankan agar perbaikan antarmuka dapat dilakukan secara lebih cepat dan terukur. Pengembang juga dapat menambahkan metrik evaluasi lain, seperti *user satisfaction* atau *task completion rate*, untuk melengkapi analisis kualitas UI/UX yang dihasilkan.

## 6. Daftar Rujukan

- [1] Ghimire, D., & Charters, S. (2022). The impact of agile development practices on project outcomes. *Software*, 1(3), 265–275. <https://doi.org/10.3390/software1030012>
- [2] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile software development. Diakses dari <https://agilemanifesto.org/>
- [3] Schwaber, K., & Sutherland, J. (2020). The Scrum guide: The definitive guide to Scrum: The rules of the game. Diakses dari <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- [4] K. Moran, C. Bernal-Cárdenas, M. Curcio, R. Bonett, and D. Poshyvanyk, "Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps," *IEEE Transactions on Software Engineering*, vol. 46, no. 2, pp. 196-221, Feb. 2020, doi: 10.1109/TSE.2018.2840000.
- [5] N. F. Syahrozad and A. P. Subriadi, "Evaluation of user experience: a systematic literature review," *E3S Web of Conferences*, vol. 501, p. 02009, 2024, doi: 10.1051/e3sconf/202450102009.
- [6] R. P. Sutanto, "Analisis User Flow pada Website Pendidikan: Studi Kasus Website DKV UK Petra," *Jurnal Desain Komunikasi Visual Nirmana*, vol. 22, no. 1, hal. 41-51, Jan. 2022, doi: 10.9744/nirmana.22.1.41-51.
- [7] G. Jurca, T. D. Hellmann, and F. Maurer, "Integrating Agile and User-Centered Design: A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX," in 2014 Agile Conference, Atlanta, GA, USA, 2014, pp. 24-32, doi: 10.1109/AGILE.2014.17.
- [8] A. P. Pratama dan R. A. Zunaidi, "Implementasi scrum model dalam pengembangan aplikasi e-commerce pada bidang jasa pembangunan rumah," *JENIUS: Jurnal Terapan Teknik Industri*, vol. 4, no. 1, hlm. 39-48, Mei 2023, doi: 10.37373/jenius.v4i1.484.
- [9] D. Ghimire and S. Charters, "The Impact of Agile Development Practices on Project Outcomes," *Software*, vol. 1, no. 3, pp. 265–275, Aug. 2022, doi: 10.3390/software1030012.

- [10] A. Andipradana and K. D. Hartomo, "Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum," *Jurnal Algoritma*, vol. 19, no. 1, pp. 161-172, 2021.
- [11] C. Tam, E. J. D. C. Moura, T. Oliveira, and J. Varajão, "The factors influencing the success of on-going agile software development projects," *International Journal of Project Management*, vol. 38, no. 3, pp. 165–176, Apr. 2020, doi: 10.1016/j.ijproman.2020.02.001.
- [12] Y. T. Widayati, Y. Prihati, S. Widjaja, S. A. Prakoso, and A. R. Notobudojo, "Implementasi Twitter Bootstrap dalam Pengembangan Aplikasi Web E-Commerce (Studi Kasus Toko Putra Reban Kendal)," *Jurnal Transformatika*, vol. 19, no. 1, pp. 26-37, 2021, doi: 10.26623/transformatika.v19i1.3541.
- [13] M. D. Firmansyah and C. Christian, "Pengembangan dan Implementasi Desain UI/UX Website Sistem Laporan Keuangan di SMK Multistudi High School Batam Menggunakan Kerangka Kerja Agile Scrum," *National Conference of Community Service Project (NaCosPro)*, vol. 4, no. 1, Sep. 2022, doi: 10.37253/nacospro.v4i1.7109.
- [14] Ragil Siti Sholehah, Ahmad Fauzi, dan Nuur Wachid Abdul Majid, "Implementasi Metode Agile Scrum Dalam Perancangan UI/UX Design Command Control Map Service Di PT. Len Industri (Persero)," *Jurnal Fasilkom*, vol. 14, no. 1, pp. 149-155, April 2024.
- [15] Harahap, D. E. J., Subana, B., and Raihan, S., "Sistem Informasi Pengelolaan Nilai Berbasis Web Pada SD Negeri Duren Jaya 6," *Jurnal Ilmiah Metadata*, vol. 3, no. 1, pp. 397-404, 2021.
- [16] Yulianti Siti Jamilah dan Ayung Candra Padmasari, "Perancangan User Interface dan User Experience Aplikasi SAY.CO," *Jurnal Tanra*, vol. 9, no. 1, pp. 73, April 2022.
- [17] Kemenparekraf RI. (2023). *Menparekraf: Perputaran Ekonomi Sektor Event di 2023 Capai Rp162 Triliun*. Diakses dari situs resmi Kementerian Pariwisata dan Ekonomi Kreatif Republik Indonesia.
- [18] R. S. Sholehah, A. Fauzi, and N. W. A. Majid, "Implementasi Metode Agile Scrum Dalam Perancangan UI/UX Design Command Control Map Service Di PT. Len Industri (Persero)," *Jurnal FASILKOM*, vol. 14, no. 1, pp. 149-155, 2024.
- [19] R. Gutama and T. Dirgahayu, "Implementasi Scrum Pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP)," *Universitas Islam Indonesia*, Yogyakarta, 2021.
- [20] Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3), 114-123.