

Contents list available at [www.jurnal.unimed.ac.id](http://www.jurnal.unimed.ac.id)

**CESS**  
**(Journal of Computing Engineering, System and Science)**

journal homepage: <https://jurnal.unimed.ac.id/2012/index.php/cess>



## Scrum Framework Dan Algoritma Greedy Pada Aktivitas Product Backlog Wedding Application Planner (Wepplan)

### Scrum Framework and Greedy Algorithm in Product Backlog Wedding Application Planner (Wepplan) Activities

Muhamad Iqbal Wasta Purnama<sup>1\*</sup>, Fauziah<sup>2</sup>, Ira Diana Sholihati<sup>3</sup>

<sup>1,2,3</sup> Fakultas Teknologi Komunikasi dan Informatika / Universitas Nasional

Jl. Sawo Manila No.61, RT/RW 14/007, Pejaten Barat, Pasar Minggu, Jakarta Selatan 12520, Indonesia.

email: <sup>1</sup>[wastaiqbal.info@gmail.com](mailto:wastaiqbal.info@gmail.com), <sup>2</sup>[fauziah@civitas.unas.ac.id](mailto:fauziah@civitas.unas.ac.id), <sup>3</sup>[iradiana2803@gmail.com](mailto:iradiana2803@gmail.com)

Diterima: 14 Desember 2021 | Diterima setelah perbaikan: 10 Januari 2022 | Disetujui: 17 Januari 2022

#### ABSTRAK

Semakin berkembangnya teknologi maka semakin banyaknya kebutuhan yang diperlukan ini bisa menjadi peluang yang mampu dimanfaatkan untuk mendukung suatu pengembangan teknologi. Pada penelitian ini menerapkan salah satu model dalam perancangan aplikasi dengan mengimplementasikan Scrum framework dan Algoritma Greedy pada aktivitas product backlog Wedding Application Planner (Wepplan). Dimana hal tersebut membentuk 271 jam waktu kerja pada 23 aktivitas product backlog yang terbagi menjadi 4 sprint dengan memberikan solusi optimal pada permasalahan penjadwalan pembagian waktu kerja masing-masing developer. Hasil yang didapat yaitu developer 1 mendapatkan 70 jam waktu kerja, developer 2: 68 jam waktu kerja, developer 3: 69 jam waktu kerja, dan developer 4: 64 jam waktu kerja dalam pengerjaannya. Pada pengujian terhadap nilai response time aplikasi, hasil tercepat pada daftar uji Test Daftar: 27.386 (s), Test Login: 15.378 (s), Test Wedding: 24.433 (s), dan Test Engagement 25.841 (s) didapat menggunakan katalon, pada daftar uji Test Venue: 26.104 (s), Test Catering: 29.244 (s), Test Top Up: 26.617 (s), dan Test Edit Profile: 27.239 (s) menggunakan Testproject. Disimpulkan penelitian ini dapat memberikan manfaat terutama kepada para developer untuk mengefisiensi penjadwalan waktu kerja terhadap rancangan aktivitas dalam membangun sebuah produk aplikasi.

**Kata Kunci:** *Wepplan, Android, Scrum Framework, Algoritma Greedy.*

#### ABSTRACT

Development of a technology, the more needs that are needed this can be an opportunity that can be used to support a technology development. In this study, one of the models in application design is implemented by Scrum framework and Greedy Algorithm in the Wedding Application Planner (Wepplan) product backlog activity. Where this forms 271 hours of work

\*Penulis Korespondensi:

email: [wastaiqbal.info@gmail.com](mailto:wastaiqbal.info@gmail.com)

time on 23 product backlog activities which are divided into 4 sprints by providing optimal solutions to the scheduling problem of each developer's work time division. The results obtained are developer 1 gets 70 hours of work time, developer 2: 68 hours of work time, developer 3: 69 hours of work time, and developer 4: 64 hours of work time in the process. In testing the application response time value, the fastest results on the Test List Signup Test: 27,386 (s), Login Test: 15,378 (s), Wedding Test: 24,433 (s), and Engagement Test 25,842 (s) were obtained using Katalon, while in Venue Test: 26,104 (s), Catering Test: 29,244 (s), Top Up Test: 26,617 (s), and Profile Update Test: 27,239 (s) using Testproject. It is concluded that this research can provide benefits, especially for developers to make work time scheduling efficient for the design of activities in building an application product.

**Keywords:** *Wepplan, Android, Scrum Framework, Greedy Algorithm.*

---

## 1. PENDAHULUAN

Pandemi COVID-19 menimbulkan banyak sekali pengaruh terhadap masyarakat di Indonesia [1]. Pengaruh yang terjadi memiliki banyak dampak negatif maupun positif yang ditimbulkan seperti dalam bidang ekonomi, kesehatan, pendidikan, dan masih lainnya [2]. Pada penelitian ini menjelaskan dampak yang terjadi dalam lingkup suatu pernikahan yang terjadi di masyarakat. Penyebaran COVID-19 menimbulkan terjadinya suatu peningkatan dalam pernikahan, diakibatkan oleh faktor seperti ketergantungan ekonomi di masyarakat [3]. Maka suatu ide muncul yaitu merancang suatu *Wedding Application Planner* (Wepplan). Dengan harapan aplikasi ini dapat mengefisiensi suatu layanan dibidang pernikahan, menciptakan kemudahan bagi masyarakat untuk mewujudkan pernikahan tanpa perlu kesulitan dalam memenuhi kebutuhan seperti penyewaan gedung, dekorasi, *custom*, maupun *catering* yang sudah tersaji dalam satu aplikasi. Sehingga dapat membantu penyedia layanan sekaligus pengguna/*users* dalam mewujudkan pernikahan yang ingin dibuat.

*Mobile Application* dipilih dalam membangun aplikasi Wepplan ini karena aplikasi *mobile* saat ini sedang berkembang pesat karena banyaknya generasi muda yang ingin berlomba-lomba dalam membentuk suatu perusahaan rintisan atau *startup*. [4]. Semakin banyaknya penggunaan aplikasi *mobile* ini akan memberikan dampak baik bagi pemilik bisnis/usaha, pengembang/*developer* dan masyarakat [5].

Pembuatan aplikasi biasanya dilakukan oleh pengembang/*developer* yang memiliki pengetahuan berbasis algoritma, pemrograman dan teknik dalam menggunakan *software* terkait [6]. Dengan pengetahuan tersebut dapat diimplementasikannya *scrum framework* dan algoritma *greedy* pada aplikasi agar menghasilkan suatu *output* dalam bentuk aktivitas yang membentuk waktu kerja optimal. Hal tersebut dapat mendukung terjadinya proses manajemen yang lebih baik dalam menerapkan kebutuhan bisnis, misalnya sebagai potensi usaha dibidang jasa maupun penjualan [7] seperti pada aplikasi Wepplan ini. Dengan dibentuknya aplikasi bisa memberikan banyak keuntungan bagi masyarakat khususnya pada pengelolaan pernikahan. Selain itu diharapkan aplikasi ini mampu memberikan peluang untuk menciptakan proses bisnis yang lebih berkembang lagi.

Penggunaan metode Scrum memungkinkan untuk memberikan perubahan pada *requirements* disaat dilakukannya perancangan perangkat lunak [8]. Scrum memiliki beberapa tahapan yaitu *product backlog*, *sprint*, *daily scrum*, *sprint review*, dan *sprint retrospective* [9]. *Product backlog* pada *scrum* cukup untuk memberikan gambaran adanya kebutuhan bagi

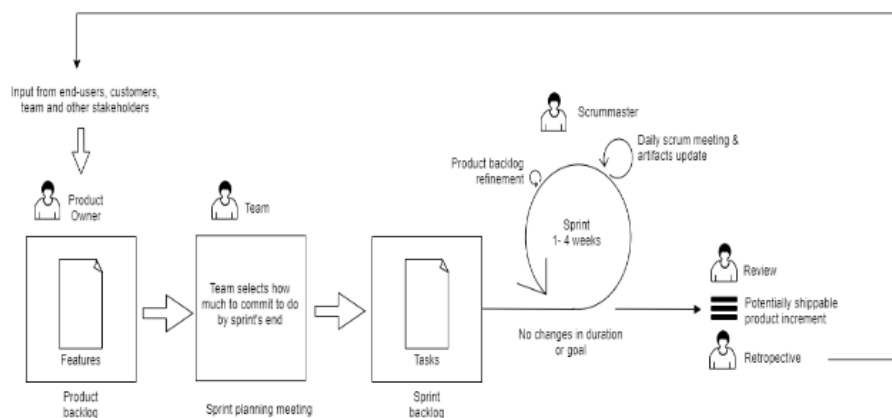
pengguna/users, namun terdapat langkah *review* untuk memberikan evaluasi terhadap produk [10].

Algoritma *Greedy* merupakan algoritma yang dipakai untuk memecahkan permasalahan pada optimasi, sebagai cara untuk mencari nilai maksimum maupun nilai minimum [11]. Sifat algoritma *greedy* yaitu tidak dapat dipulihkan (*irreversible*), apabila suatu tahapan sudah dilakukan maka nilai tersebut tidak diperbolehkan untuk diambil kembali pada *base* sebelumnya [12].

Beberapa penelitian serupa dengan metode scrum pada penerapan *product backlog*. Penelitian serupa lainnya pada penerapan algoritma *greedy* penjadwalan. Penelitian oleh Ruseno membahas mengenai implementasi scrum untuk menentukan pengembangan aplikasi reservasi dengan PHP. Penelitian mengimplementasikan basis data dan antarmuka pada pendekatan *black box* dengan hasil sistem dapat melakukan pencatatan penyewaan dan pemetaan reservasi secara otomatis, lalu meminimalisir terjadinya bentrok data [13]. Penelitian Akbar membahas penggunaan scrum dalam mengefisienkan pekerjaan serta proses bisnis di perusahaan. Penelitian berhasil memberikan analisis pada perangkat penjualan, meningkatkan pengolahan data transaksi penjualan dan pembelian, digunakannya scrum memberikan kemudahan dalam mengontrol perusahaan [14]. Penelitian Ardian penerapan algoritma *greedy* untuk pengalokasian waktu dalam menyelesaikan pekerjaan. Penelitian menghasilkan algoritma *greedy* berhasil memecahkan masalah mengenai pengalokasian waktu pekerjaan, dan memberikan solusi optimal pekerjaan pada *team* [15].

## 2. METODOLOGI PENELITIAN

Pada penelitian ini melakukan perancangan sebuah aplikasi Wepplan, dimana scrum dipakai untuk menciptakan produk yang memiliki kualitas baik secara kreatifitas maupun produktivitasnya [16]. Scrum menjadi model kerangka yang sering digunakan dalam perancangan suatu sistem untuk mencapai kebutuhan *user* [17]. Berikut bentuk dari scrum *model*:



Gambar 1. Scrum Model

### 2.1. Product Backlog

*Product backlog* merupakan urutan daftar kebutuhan yang diperlukan pada produk yang dikerjakan. Kebutuhan aktivitas ditentukan dari hasil diskusi berdasarkan urutan prioritas dan fitur yang diminta.

#### Algoritma Greedy

Aturan dasar Algoritma *greedy* sebagai berikut:

- Himpunan kandidat, mengandung anggota-anggota pembentuk solusi/penyelesaian.

- Himpunan solusi, mengandung kandidat terpilih sebagai solusi/penyelesaian permasalahan.
- Fungsi seleksi, menentukan kandidat untuk sampai ke solusi optimal dan kandidat terpilih tidak dapat dipulihkan (*irreversible*).
- Fungsi objektif, berfungsi untuk memaksimalkan maupun meminimumkan nilai solusi (misalnya waktu kerja, kecepatan, dan lainnya).

## 2.2. Sprint

*Sprint* digunakan sebagai daftar isian yang dilalui dari aktivitas suatu periode yang dikerjakan. *Sprint* terfokus pada produk yang dibuat berdasarkan aktivitas terpilih dari *Product Backlog*.

## 2.3. Halaman Antarmuka

Halaman antarmuka menampilkan hasil penyelesaian aktivitas *sprint* yang dapat ditampilkan pada aplikasi.

## 2.4. Pengujian

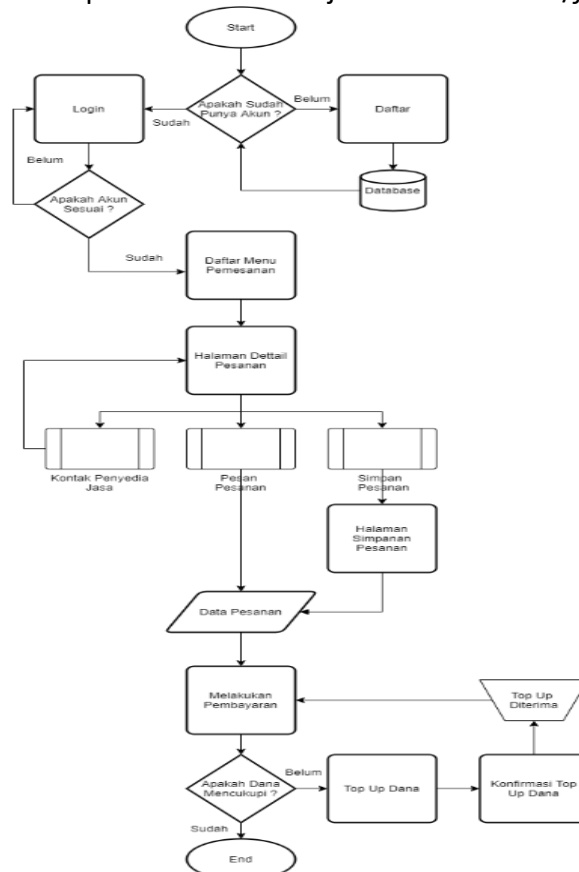
Pengujian dilakukan sebanyak 240 kali pada 8 aktivitas uji dengan 10 kali pengulangan di setiap aktivitasnya menggunakan 3 *software tester* untuk membandingkan pemrosesan database, *script code* dan *graphical user interface* (GUI) berdasarkan *response time* aplikasi.

## 3. HASIL DAN PEMBAHASAN

### 2.1. Product Backlog

Tahapan yang dilakukan antara lain:

- 1) *Flow map System*: terlihat pada Gbr 2 menunjukkan alur sistem/*flowmap* yang diusulkan.



Gambar 2. Flow map System

Gbr 2 Hasil bagian-bagian dari tahap *user* mendaftar, *login*, masuk daftar menu pemesanan dan melakukan konfirmasi pembayaran.

2) *Product Backlog*: melakukan penyusunan kebutuhan sistem berdasarkan daftar aktivitas, dan prioritas untuk mengidentifikasi waktu kerja pada sistem yang dibuat:

**Tabel 1. Initial Product Backlog 1**

No	Aktivitas	Detail Aktivitas	Prioritas	Waktu Kerja (Jam)
1.	Konseptual Aplikasi	Identifikasi Sistem	1	7
		Mockup Sistem	1	16
		Perancangan Database	1	12
2.	Login User	Halaman Daftar	1	14
		Halaman Login	1	10
<b>Sprint 1</b>				<b>59</b>
3.	Homepage	Halaman Homepage	1	14
4.	Top Up	Halaman Top Up	1	14
5.	Profile	Edit Profile	2	7
		Halaman Proses	3	5
		Halaman Selesai	3	5
6.	Save Vendor	Halaman Save Vendor	3	7
<b>Sprint 2</b>				<b>52</b>
7.	Wedding Vendor	Halaman Wedding	1	12
		Halaman Pesanan	1	14
		Halaman Detail Pesanan	1	14
8.	Engagement Vendor	Halaman Engagement	1	12
		Halaman Pesanan	1	14
		Halaman Detail Pesanan	1	14
<b>Sprint 3</b>				<b>80</b>
9.	Venue Vendor	Halaman Venue	1	12
		Halaman Pesanan	1	14
		Halaman Detail Pesanan	1	14
10.	Catering Vendor	Halaman Catering	1	12
		Halaman Pesanan	1	14
		Halaman Detail Pesanan	1	14
<b>Sprint 4</b>				<b>80</b>

Tabel 1 Hasil bentuk dari *product backlog* pada 10 aktivitas utama dan 23 aktivitas kecil yang menjadi 4 bagian *sprint* pada 3 tingkatan prioritas yaitu angka 1 merupakan prioritas tertinggi /*high Priority*, 2 prioritas menengah/*medium priority*, dan 3 prioritas rendah/*low priority*.

Penerapan algoritma *greedy* dilakukan dari beberapa tahap sebagai berikut:

- Tahap Pertama: dilakukan *sorting* pada aktivitas utama untuk diringkas menjadi bagian-bagian dari *sprint* yang diurutkan dari tingkat prioritas secara *ascending*.
- Tahap Kedua: menentukan waktu kerja berdasarkan waktu paling lama hingga terbentuk tabel berikut:

**Tabel 2. Product Backlog 1**

Sprint 1				
No	Aktivitas	Detail Aktivitas	Prioritas	Waktu Kerja (Jam)
1.	Konseptual Aplikasi	Mockup Sistem	1	16
		Perancangan Database	1	12
		Identifikasi Sistem	1	7
2.	Login User	Halaman Daftar	1	14
		Halaman Login	1	10

**Tabel 3. Product Backlog 2**

Sprint 2				
No	Aktivitas	Detail Aktivitas	Prioritas	Waktu Kerja (Jam)
1.	Homepage	Halaman Homepage	1	14
2.	Top Up	Halaman Top Up	1	14
3.	Profile	Edit Profile	2	7
		Halaman Proses	3	5
		Halaman Selesai	3	5
4.	Save Vendor	Halaman Save Vendor	3	7

**Tabel 4. Product Backlog 3**

Sprint 3				
No	Aktivitas	Detail Aktivitas	Prioritas	Waktu Kerja (Jam)
1.	Wedding Vendor	Halaman Pesanan	1	14
		Halaman Detail Pesanan	1	14
		Halaman Wedding	1	12
2.	Engagement Vendor	Halaman Pesanan	1	14
		Halaman Detail Pesanan	1	14
		Halaman Engagement	1	12

**Tabel 5. Product Backlog 4**

Sprint 4				
No	Aktivitas	Detail Aktivitas	Prioritas	Waktu Kerja (Jam)
1.	Venue Vendor	Halaman Pesanan	1	14
		Halaman Detail Pesanan	1	14
		Halaman Venue	1	12
2.	Catering Vendor	Halaman Pesanan	1	14
		Halaman Detail Pesanan	1	14
		Halaman Catering	1	12

Tabel 2, 3, 4 dan 5 Hasil ringkasan dari aktivitas *product backlog* pada setiap *sprint* yang sudah diurutkan/*sorting*.

- Tahap Ketiga: penerapan algoritma *greedy* untuk menciptakan himpunan kandidat sebagai berikut:

**Tabel 6.** Himpunan Kandidat 1

Himpunan Solusi Pertama	$C = \{ \text{"Konseptual Aplikasi - Mockup Sistem, Konseptual Aplikasi - Perancangan Database, Konseptual Aplikasi - Identifikasi Sistem, Login User - Halaman Daftar, Login User - Halaman Login"} \}$
Himpunan Solusi Kedua	$C = \{ \text{"Konseptual Aplikasi - Perancangan Database, Konseptual Aplikasi - Identifikasi Sistem, Login User - Halaman Daftar, Login User - Halaman Login"} \}$
Himpunan Solusi Ketiga	$C = \{ \text{"Konseptual Aplikasi - Identifikasi Sistem, Login User - Halaman Daftar, Login User - Halaman Login"} \}$
Himpunan Solusi Keempat	$C = \{ \text{"Login User - Halaman Daftar, Login User - Halaman Login"} \}$
Himpunan Solusi Kelima	$C = \{ \text{"Login User - Halaman Login"} \}$

**Tabel 7.** Himpunan Kandidat 2

Himpunan Solusi Pertama	$C = \{ \text{"Homepage - Halaman Homepage, Top Up - Halaman Top Up, Profile - Edit Profile, Profile - Halaman Proses, Profile - Halaman Selesai, Save Vendor - Halaman Save Vendor"} \}$
Himpunan Solusi Kedua	$C = \{ \text{"Top Up - Halaman Top Up, Profile - Edit Profile, Profile - Halaman Proses, Profile - Halaman Selesai, Save Vendor - Halaman Save Vendor"} \}$
Himpunan Solusi Ketiga	$C = \{ \text{"Profile - Edit Profile, Profile - Halaman Proses, Profile - Halaman Selesai, Save Vendor - Halaman Save Vendor"} \}$
Himpunan Solusi Keempat	$C = \{ \text{"Profile - Halaman Proses, Profile - Halaman Selesai, Save Vendor - Halaman Save Vendor"} \}$
Himpunan Solusi Kelima	$C = \{ \text{"Profile - Halaman Selesai, Save Vendor - Halaman Save Vendor"} \}$
Himpunan Solusi Keenam	$C = \{ \text{"Save Vendor - Halaman Save Vendor"} \}$

**Tabel 8.** Himpunan Kandidat 3

Himpunan Solusi Pertama	$C = \{ \text{"Wedding Vendor - Halaman Pesanan, Wedding Vendor - Halaman Detail Pesanan, Wedding Vendor - Halaman Wedding, Engagement Vendor - Halaman Pesanan, Engagement Vendor - Halaman Detail Pesanan, Engagement Vendor - Halaman Engagement"} \}$
Himpunan Solusi Kedua	$C = \{ \text{"Wedding Vendor - Halaman Detail Pesanan, Wedding Vendor - Halaman Wedding, Engagement Vendor - Halaman Pesanan, Engagement Vendor - Halaman Detail Pesanan, Engagement Vendor - Halaman Engagement"} \}$
Himpunan Solusi Ketiga	$C = \{ \text{"Wedding Vendor - Halaman Wedding, Engagement Vendor - Halaman Pesanan, Engagement Vendor - Halaman Detail Pesanan, Engagement Vendor - Halaman Engagement"} \}$

Himpunan Solusi Keempat	C = { "Engagement Vendor - Halaman Pesanan, Engagement Vendor - Halaman Detail Pesanan, Engagement Vendor - Halaman Engagement" }
Himpunan Solusi Kelima	C = { "Engagement Vendor - Halaman Detail Pesanan, Engagement Vendor - Halaman Engagement" }
Himpunan Solusi Keenam	C = { "Engagement Vendor - Halaman Engagement" }

**Tabel 9.** Himpunan Kandidat 4

Himpunan Solusi Pertama	C = { "Venue Vendor - Halaman Pesanan, Venue Vendor - Halaman Detail Pesanan, Venue Vendor - Halaman Venue, Catering Vendor - Halaman Pesanan, Catering Vendor - Halaman Detail Pesanan, Catering Vendor - Halaman Catering" }
Himpunan Solusi Kedua	C = { "Venue Vendor - Halaman Detail Pesanan, Venue Vendor - Halaman Venue, Catering Vendor - Halaman Pesanan, Catering Vendor - Halaman Detail Pesanan, Catering Vendor - Halaman Catering" }
Himpunan Solusi Ketiga	C = { "Venue Vendor - Halaman Venue, Catering Vendor - Halaman Pesanan, Catering Vendor - Halaman Detail Pesanan, Catering Vendor - Halaman Catering" }
Himpunan Solusi Keempat	C = { "Catering Vendor - Halaman Pesanan, Catering Vendor - Halaman Detail Pesanan, Catering Vendor - Halaman Catering" }
Himpunan Solusi Kelima	C = { "Catering Vendor - Halaman Detail Pesanan, Catering Vendor - Halaman Catering" }
Himpunan Solusi Keenam	C = { "Catering Vendor - Halaman Catering" }

Tabel 6, 7, 8 dan 9 Menginisialisasi himpunan berdasarkan jumlah aktivitas yang ada pada setiap *sprint*.

Himpunan kandidat didapatkan dari hasil tabel *product backlog* yang dibuat menjadi hasil Himpunan Solusi (S)  $S = \{ \}$ . Daftar himpunan kandidat yang mencapai aktivitas akhir/selesai akan menghasilkan himpunan solusi terlihat pada tabel 10, 11, 12, 13.

**Tabel 10.** Himpunan Solusi 1

Developer 1	S = { Konseptual Aplikasi - Mockup Sistem }
Developer 2	S = { Konseptual Aplikasi - Perancangan Database }
Developer 3	S = { Konseptual Aplikasi - Identifikasi Sistem, Login User - Halaman Login }
Developer 4	S = { Login User - Halaman Daftar }

Tabel 10 Hasil himpunan solusi *sprint* 1 pada aktivitas konseptual aplikasi dan *login user*.



**Tabel 11.** Himpunan Solusi 2

Developer 1	S = {Profile - Edit Profile, Save Vendor - Halaman Save Vendor}
Developer 2	S = { Homepage - Halaman Homepage}
Developer 3	S = {Profile - Halaman Proses, Profile -Halaman Selesai}
Developer 4	S = { Top Up - Halaman Top Up }

Tabel 11 Hasil himpunan solusi *sprint 2* pada aktivitas *homepage, profile, top up, dan save vendor*.

**Tabel 12.** Himpunan Solusi 3

Developer 1	S = {Engagement Vendor - Halaman Pesanan}
Developer 2	S = {Wedding Vendor - Halaman Pesanan, Engagement Vendor - Halaman Detail Pesanan}
Developer 3	S = {Wedding Vendor - Halaman Detail Pesanan}
Developer 4	S = {Wedding Vendor - Halaman Wedding, Engagement Vendor - Halaman Engagement}

Tabel 12 Hasil himpunan solusi *sprint 3* pada aktivitas *wedding vendor, dan engagement vendor*.

**Tabel 13.** Himpunan Solusi 4

Developer 1	S = {Venue Vendor - Halaman Detail Pesanan, Catering Vendor - Halaman Catering}
Developer 2	S = {Catering Vendor-Halaman Pesanan }
Developer 3	S = {Venue Vendor - Halaman Pesanan, Catering Vendor - Halaman Detail Pesanan}
Developer 4	S = {Venue Vendor-Halaman Venue}

Tabel 13 Hasil himpunan solusi *sprint 4* pada aktivitas *venue vendor, dan catering vendor*.

Setelah seleksi dilakukan maka terbentuk daftar aktivitas kerja pada setiap *developer* berdasarkan aktivitas yang dikerjakan beserta waktu kerja yang didapat.

**Tabel 14.** Hasil Penjadwalan

Developer	Aktivitas Kerja	Waktu Kerja (jam)	Total (jam)
Developer 1	Konseptual Aplikasi-Mockup Sistem	16	70
	Profile-Edit Profile	7	
	Save Vendor-Halaman Save Vendor	7	
	Engagement Vendor-Halaman Pesanan	14	
	Venue Vendor-Halaman Detail Pesanan	14	
	Catering Vendor-Halaman Catering	12	
Developer 2	Konseptual Aplikasi-Perancangan Database	12	68
	Homepage-Halaman Homepage	14	
	Wedding Vendor-Halaman Pesanan	14	
	Engagement Vendor-Halaman Detail Pesanan	14	
	Catering Vendor-Halaman Pesanan	14	
Developer 3	Konseptual Aplikasi-Identifikasi Sistem	7	69
	Login User - Halaman Login	10	

	Profile-Halaman Proses	5	
	Profile-Halaman Selesai	5	
	Wedding Vendor-Halaman Detail Pesanan	14	
	Venue Vendor-Halaman Pesanan	14	
	Catering Vendor-Halaman Detail Pesanan	14	
Developer 4	Login User-Halaman Daftar	14	
	Top Up-Halaman Top Up	14	
	Wedding Vendor-Halaman Wedding	12	64
	Engagement Vendor-Halaman Engagement	12	
	Venue Vendor-Halaman Venue	12	

TABEL 14 Hasil dari pembagian aktivitas kerja berdasarkan jam waktu kerja yang diturunkan dari setiap *sprint*.

## 2.2. Sprint

Menampilkan daftar isian dari aktivitas setiap *sprint* yang telah dikerjakan sebagai aktivitas yang telah diselesaikan.

Tabel 15. *Sprint 1*

Aktivitas	Status	Responsible	Prioritas	Waktu Kerja (Jam)
Mockup Sistem	Selesai	Developer 1	1	16
Perancangan Database	Selesai	Developer 2	1	12
Identifikasi Sistem	Selesai	Developer 3	1	7
Halaman Daftar	Selesai	Developer 4	1	14
Halaman Login	Selesai	Developer 3	1	10

Tabel 15 Hasil aktivitas kerja *sprint 1* dengan pembagian waktu kerja yaitu *developer 1*: 16 jam, *developer 2*: 12 jam, dan *developer 3*: 17 jam, *developer 4*: 14 jam.

Tabel 16. *Sprint 2*

Aktivitas	Status	Responsible	Prioritas	Waktu Kerja (Jam)
Halaman Homepage	Selesai	Developer 2	1	14
Halaman Top Up	Selesai	Developer 4	1	14
Edit Profile	Selesai	Developer 1	2	7
Halaman Proses	Selesai	Developer 3	3	5
Halaman Selesai	Selesai	Developer 3	3	5
Halaman Save Vendor	Selesai	Developer 1	3	7

Tabel 16 Hasil aktivitas kerja *sprint 2* dengan pembagian waktu kerja yaitu *developer 1*: 14 jam, *developer 2*: 14 jam, dan *developer 3*: 10 jam, *developer 4*: 14 jam.

**Tabel 17. Sprint 3**

<b>Aktivitas</b>	<b>Status</b>	<b>Responsible</b>	<b>Prioritas</b>	<b>Waktu Kerja (Jam)</b>
Halaman Pesanan (Wedding)	Selesai	Developer 2	1	14
Halaman Detail Pesanan (Wedding)	Selesai	Developer 3	1	14
Halaman Wedding	Selesai	Developer 4	1	12
Halaman Pesanan (Engagement)	Selesai	Developer 1	2	14
Halaman Detail Pesanan (Engagement)	Selesai	Developer 2	3	14
Halaman Engagement	Selesai	Developer 4	3	12

Tabel 17 Hasil aktivitas kerja *sprint 3* dengan pembagian waktu kerja yaitu *developer 1*: 14 jam, *developer 2*: 28 jam, dan *developer 3*: 14 jam, *developer 4*: 24 jam.

**Tabel 18. Sprint 4**

<b>Aktivitas</b>	<b>Status</b>	<b>Responsible</b>	<b>Prioritas</b>	<b>Waktu Kerja (Jam)</b>
Halaman Pesanan (Venue)	Selesai	Developer 3	1	14
Halaman Detail Pesanan (Venue)	Selesai	Developer 1	1	14
Halaman Venue	Selesai	Developer 4	1	12
Halaman Pesanan (Catering)	Selesai	Developer 2	2	14
Halaman Detail Pesanan (Catering)	Selesai	Developer 3	3	14
Halaman Catering	Selesai	Developer 1	3	12

Tabel 18 Hasil aktivitas kerja *sprint 4* dengan pembagian waktu kerja yaitu *developer 1*: 26 jam, *developer 2*: 14 jam, dan *developer 3*: 28 jam, *developer 4*: 12 jam.

Maka didapatkan hasil total waktu kerja yaitu 271 jam pada 23 aktivitas dengan pembagian waktu kerja setiap *developer* yaitu *developer 1* mendapatkan waktu kerja total 70 jam, *developer 2* waktu kerja total 68 jam, *developer 3* waktu kerja total 69 jam, dan *developer 4* waktu kerja total 64 jam.

### 4.3. Halaman Antarmuka

Aplikasi Wepplan, dibuat dalam bentuk *mobile application*. Berikut antarmuka/*interface* pada aplikasi.

1) Halaman Daftar Bagian 1

Daftar Akun

Username  
username

Password  
\*\*\*\*\*

Email  
anda@wepplan.com

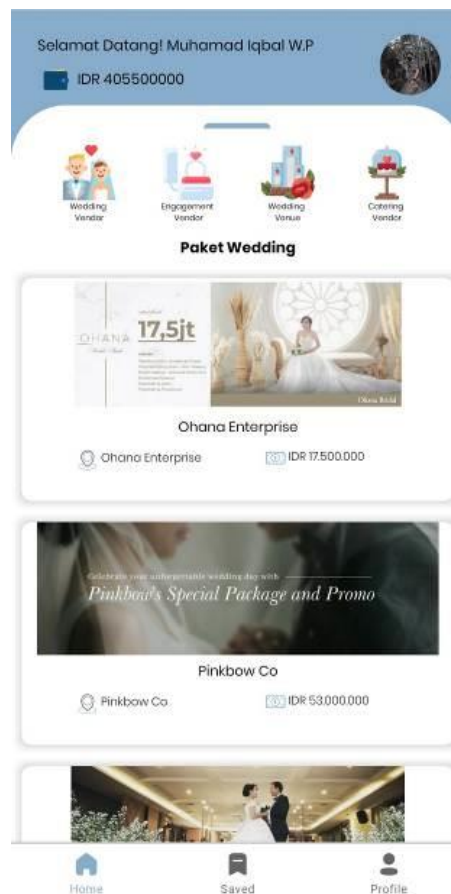
CONTINUE

Sudah Punya Akun? Masuk

**Gambar 3.** Halaman Daftar Bagian 1

Gambar 3 Halaman user untuk mendaftarkan Username, Password dan Email.

2) Halaman home



**Gambar 4.** Halaman Home

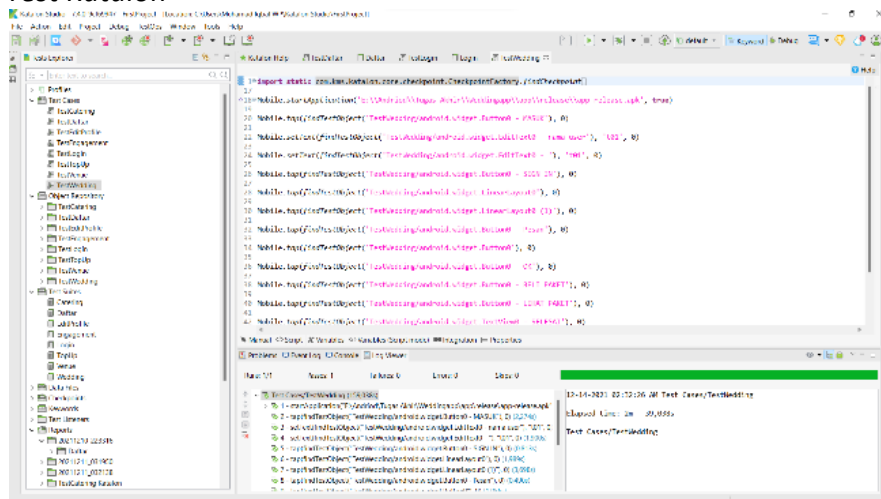
Gambar 4 Halaman *home* berisi informasi *Wedding Vendor, Engagement Vendor, Wedding Venue, Catering Vendor, dan Profile.*

#### 4.4. Pengujian

Hasil *response time* dari setiap aktivitas uji dicatat dan dihitung menjadi *mean/nilai rata-rata* untuk melihat perbedaan yang terjadi terhadap *respon time* aplikasi berdasarkan 3 *software* pembandingan dengan satuan *second/detik* pada pengolahan database, *script code* dan *graphical user interface (GUI).*

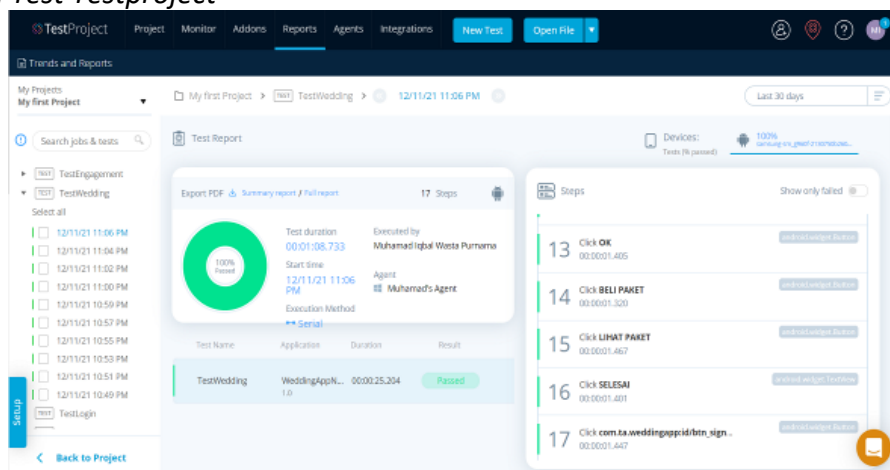
Katalon, Testproject, dan Rapise sering digunakan untuk memvalidasi kualitas perangkat lunak. Pengujian dilakukan secara otomatis menggunakan *software* tersebut guna untuk mempercepat proses testing seperti database, Gui, Coding, dan lain-lain, berikut contoh penggunaan *software*:

##### 1) Halaman Test Katalon



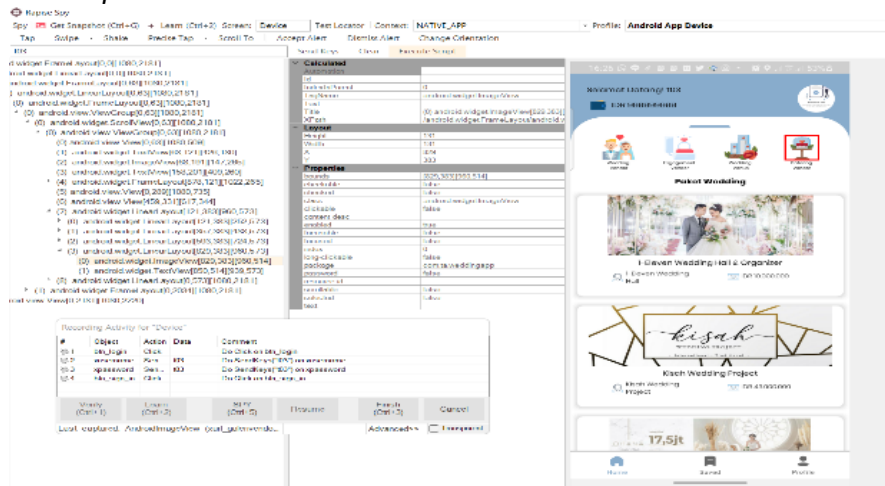
Gambar 5. Halaman Test Katalon

##### 2) Halaman Test Testproject



Gambar 6. Halaman Test Testproject

### 3) Halaman Test Rapise



Gambar 7. Halaman Test Rapise

Tabel 19. Nilai Rata-rata Aktivitas Uji

Aktivitas Uji	Katalon	TestProject	Rapise
Test Daftar	27.386 (s)	27.255 (s)	30.495 (s)
Test Login	15.378 (s)	16.610 (s)	23.152 (s)
Test Wedding	24.433 (s)	27.679 (s)	32.348 (s)
Test Engagement	25.842 (s)	26.284 (s)	33.441 (s)
Test Venue	27.361 (s)	26.104 (s)	31.982 (s)
Test Catering	30.640 (s)	29.244 (s)	36.057 (s)
Test TopUp	33.576 (s)	26.617 (s)	33.567 (s)
Test EditProfile	33.992 (s)	27.239 (s)	34.202 (s)

Tabel 19 Hasil aktivitas uji nilai *response time* tercepat terjadi menggunakan Katalon pada daftar uji Test Daftar: 27.386 (s), Test Login: 15.378 (s), Test Wedding: 24.433 (s), dan Test Engagement: 25.842 (s), sedangkan untuk daftar uji Test Venue: 26.104 (s), Test Catering: 29.244 (s), Test Top Up: 26.617 (s), dan Test Edit Profile: 27.239 (s) lebih cepat terjadi menggunakan Testproject.

## 4. PENUTUP

### 4.1. Kesimpulan

Berdasarkan hasil penelitian perancangan aktivitas *product backlog scrum framework* pada masalah penjadwalan waktu kerja dapat diselesaikan menggunakan penerapan algoritma *greedy*, dengan memberikan hasil waktu kerja yang optimal kepada para *developer* dalam menyelesaikan aktivitas kerjanya yaitu *developer* 1: 70 jam, *developer* 2: 68 jam, *developer* 3: 69 jam, dan *developer* 4: 64 jam pada waktu total 271 jam.

### 4.2. Saran

Saran yang disampaikan pengembangan *Wedding Application Planner* (Wepplan) ini dapat dikembangkan menjadi lebih luas dan diharapkan dapat menjadi salah satu prospek bisnis kedepannya.

## REFERENSI

- [1] M. Jamili, "Dampak Pandemi Covid-19 terhadap Pelayanan Pernikahan dan Permohonan Pernikahan di Kantor Urusan Agama (KUA) Banjarmasin Tengah Kota Banjarmasin," *J. Al 'Idara Balad*, vol. 2, no. 2, p. 36, 2020. Available: <http://ojs-al-iidara-balad.web.id/index.php/ANE2018/article/view/79/76>.
- [2] D. Susanti, "Perancangan Sistem Informasi E-Booking Paket Pernikahan Pada Masa Pandemi Covid-19 Berbasis Web ( Studi Kasus : Galih Wedding Organizer )," no. April, pp. 263–270, 2021.
- [3] M. Z. Alfaruqy, F. K. Putri, and S. I. Soedibyo, "Dinamika Psikologis Menikah pada Masa Pandemi COVID-19," *J. Psikol. Talent.*, vol. 6, no. 2, p. 55, Mar. 2021, doi: 10.26858/talenta.v6i2.19695.
- [4] W. Sukiman, A. Noertjahyana, and A. N. Purbowo, "Aplikasi Pengelola Tamu Wedding berbasis Mobile Android," *J. Infra*, vol. 7, no. 2, pp. 141–147, 2020.
- [5] N. Ariyani, "Aplikasi Reservasi Wedding Organizer Berbasis Android Dengan Generate Random Invitation dan SPK Pemilihan Konsep Pernikahan," pp. 40–43.
- [6] U. Nurhasan, M. Mentari, K. Hartati, and N. Ningtyas, "Penerapan Aplikasi Up-Event pada Tata Kelola Kegiatan Multi Vendor," *Gener. J.*, vol. 4, no. 2, pp. 84–97, 2020, doi: 10.29407/gj.v4i2.14291.
- [7] F. Rizaldi, S. Anggraeni, and I. Sugiyarto, "Perancangan Sistem Informasi Pemesanan Paket Pernikahan Berbasis Web Pada Diamond Wedding Jakarta," vol. 7, no. 1, pp. 1–16, 2021.
- [8] R. Setiawan, A. D. Supriatna, S. Hudawiguna, and F. F. Roji, "Electronic culinary reservations based on Android with the Scrum methodology and Firebase database," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1098, no. 5, p. 052091, 2021, doi: 10.1088/1757-899x/1098/5/052091.
- [9] A. Wibowo, R. Delima, and C. Antonius Rachmat, "Designing and Developing an Agricultural Product Sales Application Catalog with a Hybrid Application Development Framework," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1077, no. 1, p. 012050, 2021, doi: 10.1088/1757-899x/1077/1/012050.
- [10] A. Satapathy, S. S. Sahoo, H. Raj, Y. Gupta, and S. Maity, "Website on Smart Project Management System by Application of Scrum Technology," *Smart Innov. Syst. Technol.*, vol. 153, pp. 675–683, 2021, doi: 10.1007/978-981-15-6202-0\_68.
- [11] S. H. Ardiansyah, "Implementasi algoritma greedy pada aplikasi penjualan berbasis android 1,2," pp. 1063–1069, 2019.
- [12] M. Syahlan, A. Syam, F. M. Faqi, N. Ahmad, and G. Basri, "Sistem Informasi Jasa Wedding Organizer Dengan Fitur Pemilihan Paket Otomatis Menggunakan Algoritma Greedy Pada Koya Wedding," vol. IX, no. 2, pp. 190–202, 2020.
- [13] N. Ruseno, "Implementasi Scrum pada Pengembangan Aplikasi Sistem Reservasi Online Menggunakan PHP," *Gerbang*, vol. 9, no. 1, pp. 8–15, 2019. Available: <http://jurnal.stmik.banisaleh.ac.id/index.php/JIST/article/view/61>.
- [14] R. Risnawati, A. Muhammad, and F. Fatmasari, "Analisis dan perancangan perangkat lunak penjualan menggunakan metode scrum (studi kasus cv. Rizki mandiri tebat jaya belitang)," *J. Mhs. Tek. Inform.*, 2014.
- [15] A. Yunata and Q. Widayati, "Penjadwalan Pembuatan Aplikasi Container Portacamp Menggunakan Php," pp. 183–195.
- [16] T. Wijaya, "Penerapan Metode Scrum Dan Virtual Private Network Dalam Perancangan

- Sistem Ordersales,” *CCIT J.*, vol. 11, no. 1, pp. 115–125, 2018, doi: 10.33050/ccit.v11i1.565.
- [17] W. Suharso, B. I. Wicaksono, and G. I. Marthasari, “SATIN-Sains dan Teknologi Informasi Penerapan Scrum dan Algoritma Cocomo Pada Aplikasi Manajemen Proyek Perangkat Lunak,” vol. 4, no. 1, 2018.