

## ADOPSI METODE SCRUM UNTUK ELISITASI KEBUTUHAN SMART TOURISM

Dian Niswati<sup>1</sup>, Asmawati S<sup>2</sup> dan Nuralamsah Zulkarnaim<sup>3</sup>

<sup>1,2,3</sup>Program Studi Informatika, Fakultas Teknik, Universitas Sulawesi Barat, Indonesia

<sup>1</sup>Email: [dianniswati6@gmail.com](mailto:dianniswati6@gmail.com)

<sup>2</sup>Email: [asmawati.s@unsulbar.ac.id](mailto:asmawati.s@unsulbar.ac.id)

<sup>3</sup>Email: [nuralamsah@unsulbar.ac.id](mailto:nuralamsah@unsulbar.ac.id)

### ABSTRAK

Langkah pertama dalam rekayasa perangkat lunak adalah mendapatkan informasi tentang ruang lingkup dan kebutuhan sistem yang akan dibangun. Namun, ada beberapa masalah yang dapat menghambat dalam proses perancangan elisitasi kebutuhan *smart tourism* seperti kesalahpahaman dalam pendefinisian kebutuhan sistem yang telah ditentukan oleh satu pihak sehingga mengakibatkan potensi besar terjadinya perubahan. Tujuan perancangan elisitasi kebutuhan *smart tourism* untuk meminimalisir terjadinya perubahan. Perancangan elisitasi kebutuhan *smart tourism* menggunakan metode *scrum* yang menekankan pada kolaborasi tim dan bersifat adaptif. Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian deskriptif dengan pendekatan kualitatif. Hasil dari perancangan elisitasi kebutuhan *smart tourism* yaitu sebuah fitur untuk perancangan *smart tourism* dengan desain sistem yang fokus pada *use case diagram* menggunakan *Unified Modelling Language* (UML).

Kata kunci: Elisitasi Kebutuhan, *Scrum*, *Smart Tourism*.

### ABSTRACT

The first step in software engineering is to get information about the scope and requirements of the system to be built. However, there are several problems that can hinder the process of designing the elicitation of smart tourism needs, such as a misunderstanding in the definition of system requirements that have been determined by one party, resulting in great potential for change. The purpose of designing the elicitation of smart tourism needs is to minimize the occurrence of change. The design of elicitation of smart tourism needs uses the scrum method which emphasizes team collaboration and is adaptive. The type of research used in this research is descriptive research with a qualitative approach. The results of the elicitation of smart tourism needs are a feature for designing smart tourism with a system design that focuses on use case diagrams using Unified Modeling Language (UML).

Keywords: Requirement Elicitation, Scrum, Smart Tourism

## 1. PENDAHULUAN

Pariwisata menjadi salah satu sebuah sektor ekonomi yang mempunyai daya tarik yang tinggi dan tingkat pertumbuhan yang pesat, baik dari skala global maupun nasional. Khususnya di Provinsi Sulawesi Barat yang merupakan salah satu daerah yang di dalamnya memiliki begitu banyak tempat wisata yang menarik.

Media promosi yang digunakan pada objek wisata di Sulawesi Barat saat ini berupa media kertas seperti buku dan brosur, sehingga masih banyak wisatawan yang belum mengetahui tempat wisata tersebut. Untuk lebih meningkatkan pemahaman serta penjelasan tentang tempat wisata yang ada di Sulawesi Barat, dibutuhkan pemanfaatan kemajuan di bidang teknologi informasi yang sekarang semakin pesat yaitu dengan cara perancangan sistem smart tourism.

Pemanfaatan e-tourism yang bertujuan untuk memudahkan pengguna dalam mengakses informasi tempat wisata dan menghasilkan sebuah sistem informasi berbasis *website* sebagai sarana pengenalan dan promosi wisata bahari yang layak dan dapat diakses secara *online* [1]. Penelitian lainnya yang dilakukan oleh [2] menjelaskan bahwa *smart tourism* merupakan sebuah pemanfaatan teknologi dalam meningkatkan promosi di bidang pariwisata. Smart tourism ini diharapkan dapat menjadi sebuah media utama bagi wisatawan untuk menemukan destinasi wisata yang mereka impikan.

Pembangunan sebuah perangkat lunak akan diawali dengan proses analisis terhadap kebutuhan perangkat lunak dari pengguna [3]. Elisitasi kebutuhan merupakan sebuah kegiatan yang dilakukan untuk menemukan kebutuhan-kebutuhan sistem dengan melalui komunikasi antarpelanggan atau mitra, pengguna sistem dan pihak lain yang memiliki kepentingan dalam pengembangan suatu sistem. Elisitasi kebutuhan menjadi salah satu tahap awal dalam proses sebuah rekayasa perangkat lunak dan memiliki tantangan yang berkaitan dengan identifikasi dan pengelolaan persyaratan untuk pengembangan sistem perangkat lunak yang akan memenuhi standar kualitas fungsionalitas yang diinginkan [4]. Namun dalam proses elisitasi

masih ada beberapa masalah yang dapat menghambat proses elisitasi kebutuhan, yaitu keputusan satu pihak dalam menentukan kebutuhan sistem memiliki potensi besar terjadinya perubahan bahkan keputusan satu pihak dalam menentukan kebutuhan sistem memungkinkan hasil kebutuhan sistem tersebut tidak dipahami terjadinya kesalahpahaman dalam mendefinisikan sebuah kebutuhan. Dengan kondisi yang telah diketahui maka diperlukan metode pengembangan sistem yang dapat mengurangi potensi terjadinya perubahan dalam menentukan kebutuhan sistem serta metode yang tanggap terhadap perubahan.

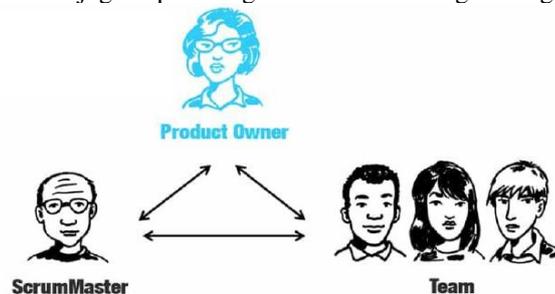
Berdasarkan masalah di atas dapat disimpulkan bahwa dengan melakukan perancangan *smart tourism* untuk melakukan elisitasi kebutuhan dengan cara mengadopsi metode *scrum* menjadi salah satu solusi yang mengutamakan kerja sama tim dan fleksibilitas. *Scrum* adalah sebuah kerangka kerja responsif dalam pengembangan perangkat lunak, pengelolaan produk serta pengembangan aplikasi. *Scrum* memiliki sebuah proses yang kompleks di mana banyak faktor yang akan mempengaruhi hasil akhir [5].

Metodologi *agile* merupakan sebuah pendekatan khusus untuk manajemen produk yang akan digunakan dalam pengembangan perangkat lunak. Metode *agile* akan sangat membantu tim dalam merespon sebuah ketidakpastian dalam pengembangan perangkat lunak dengan menggunakan inkremental berulang yang biasanya disebut dengan *sprint* serta metode pengembangan perangkat lunak yang mendasar pada prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan yang terjadi dalam bentuk apapun [6].

Kerangka kerja *scrum* ini biasanya digunakan untuk mengatasi hambatan yang sering ditemui saat menggunakan metodologi manajemen proyek tradisional yang lebih menekankan pada pendekatan *top-down*. *Scrum* memiliki keunggulan yaitu bersifat adaptif dan dapat beradaptasi dengan keadaan organisasi dari waktu ke waktu. *Scrum* bertujuan untuk menunjukkan cara yang lebih cepat serta efektif untuk pengembangan perangkat lunak pada sebuah industri teknologi. *Scrum* lebih memusatkan untuk menekankan pada kolaborasi tim dibandingkan dengan metode pengembangan perangkat lunak secara tradisional [7].

## 2. MATERI DAN METODE

Penelitian ini menggunakan adopsi metode *scrum* karena mengadopsi *scrum* merupakan sebuah metode dengan cara kerja yang tepat untuk membantu dalam memecahkan masalah dan membuat kerja tim lebih efektif sehingga permasalahan tersebut dapat diselesaikan dan solusi yang optimal dapat diperoleh. Bahkan, mengadopsi metode *scrum* juga dapat menghasilkan dan mengembangkan ide-ide baru.



Gambar 1. Tim *scrum*

Tim *scrum* memiliki tiga peran yaitu di antaranya *product owner* (pemilik produk), *scrum master* dan *development team* (tim pengembang) yang dapat dilihat pada gambar 1 [8].

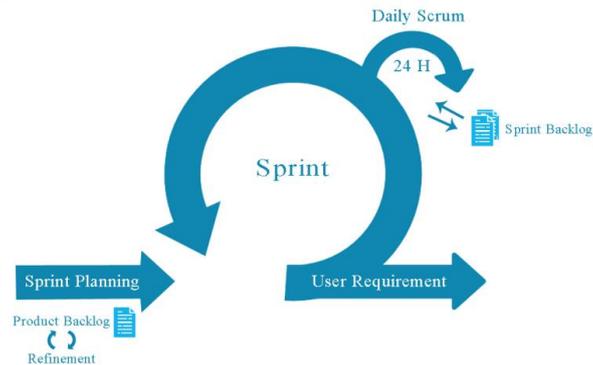
1. *Product owner* memiliki tanggung jawab untuk menentukan dan memaksimalkan nilai bisnis produk. *Product owner* mengumpulkan kebutuhan produk yang merupakan prioritas tinggi dan harus segera dikerjakan oleh tim kebutuhan produk.
2. *Development team* adalah seorang *programmer* yang memiliki peran untuk mengerjakan dan melakukan pengembangan produk.
3. *Scrum master* melakukan manajemen dari setiap proses *scrum* yang sedang berlangsung. *Scrum master* akan menjelaskan proses kerja *scrum* kepada tim yang terlibat dan melihat apakah penerapan *scrum* dapat dipahami oleh setiap individu.

Pada tahap menentukan hasil elisitasi kebutuhan dengan mengadopsi *scrum* menggunakan tahapan *sprint planning*, *daily scrum* (*sprint*) dalam mengadopsi metode *scrum* yang dapat dilihat pada gambar 2.

### 1. *Sprint*

*Sprint* adalah batas waktu sekitar satu bulan atau kurang lebih, di mana potongan produk yang “Selesai”, berpotensi untuk dikembangkan. *sprint* biasanya memiliki durasi yang konsisten selama proses pengembangan produk. *Sprint* baru akan dimulai setelah *sprint* yang sebelumnya berakhir. *Sprint* terdiri dari *sprint planning*, *daily scrum*.

2. *Sprint planning*  
*Sprint planning* adalah tahap perencanaan yang dilakukan di dalam *sprint*. Perencanaan ini dibuat secara kolaboratif oleh seluruh anggota tim. *Sprint planning* dibatasi maksimum delapan jam untuk *sprint* yang lebih pendek, batasan waktu yang digunakan biasanya lebih singkat. Tahap ini dijalankan secara bersama-sama/kolaboratif oleh anggota tim *scrum*. Penerapan *sprint planning* dilakukan dengan menganalisis kebutuhan sistem yang dibuat, seperti penambahan fitur-fitur baru, perbaikan kebutuhan sistem yang akan dikerjakan.
3. *Daily scrum*  
*Daily scrum* adalah tahapan untuk membicarakan pekerjaan dengan batas waktu hingga 15 menit agar tim pengembang dapat menyinkronkan pekerjaan dan rencana mereka selama 24 jam ke depan. Hal ini dilakukan dengan meninjau pekerjaan semenjak acara *daily scrum* terakhir dan memperkirakan pekerjaan yang dapat dilakukan sebelum melakukan *daily scrum* berikutnya.



Gambar 2. Tahapan dalam mengadopsi metode *scrum*

### 3. HASIL DAN PEMBAHASAN

#### Adopsi metode *scrum*

Bagian ini menjelaskan pengelompokan elisitasi kebutuhan dengan mengadopsi metode *scrum*. Dalam proses elisitasi kebutuhan ditentukan prioritas kebutuhan sistem yang akan digunakan untuk tahap perancangan sistem yang akan diberikan kepada tim pengembangan. Tahapan pertama yang dilakukan pada saat mengadopsi metode *scrum* yaitu menentukan *product backlog*. Setelah dilakukan penentuan *product backlog*, maka tahap selanjutnya yaitu keproses penentuan *sprint backlog* dan melakukan sebuah *sprint*.

#### Membuat tabel penentuan kebutuhan prioritas

Tabel 1. Tabel tingkat prioritas kebutuhan

No	Penentuan Prioritas	Keterangan
1	Opsional	Dilakukan setelah menyelesaikan prioritas
2	Prioritas	Pengerjaan didahulukan

Tabel 1 memperlihatkan bahwa akan ditentukan kebutuhan prioritas sistem dengan membagi tingkat prioritas kebutuhan menjadi dua kategori yaitu penentuan prioritas yang didahulukan pengerjaannya dan penentuan prioritas opsional yang dilakukan setelah menyelesaikan prioritas.

#### Menentukan *sprint backlog*

Tabel 2. Tabel *sprint backlog*

No	Tahapan	Estimasi Pengerjaan
1	Menyelesaikan fitur untuk Wisatawan	3 Hari
2	Menyelesaikan fitur untuk Admin	3 Hari
3	Menyelesaikan fitur untuk Usaha Mikro, Kecil, dan Menengah (UMKM)	2 Hari
4	Menyelesaikan fitur untuk Media	2 Hari
5	Menyelesaikan fitur untuk Pemerintah	2 Hari

Tabel 2 memperlihatkan estimasi pengerjaan dari setiap fitur yang akan dikerjakan. Menentukan *sprint backlog* ini dilakukan bersama tim *scrum*, dalam menentukan estimasi pengerjaannya dilihat dari tahapan yang akan dikerjakan seperti untuk fitur Wisatawan dan Admin dilakukan selama 3 hari karena memiliki banyak fitur yang akan di kerjakan sedangkan untuk fitur UMKM, Media dan Pemerintah hanya dilakukan selama 2 hari.

**Menentukan *product backlog* berdasarkan fitur user**

Tabel 3. Tabel *product backlog* berdasarkan fitur user

No	User	Fitur	Penentuan Kebutuhan
1	Wisatawan	Fitur Pendaftaran	Prioritas
		Fitur Login	Prioritas
		Fitur Lupa Password	Prioritas
		Fitur Profile	Prioritas
		Fitur Manual Book	Prioritas
		Fitur Mencari Tempat Wisata	Prioritas
		Fitur Mencari Penginapan	Prioritas
		Fitur Mencari Paket Wisata	Prioritas
		Fitur Booking Tempat	Prioritas
		Fitur Batalan Booking	Prioritas
		Fitur Pembayaran	Prioritas
		Fitur Melaporkan Tempat Wisata	Opsional
		Fitur Lihat Maps	Prioritas
		Fitur Rekomendasi Tempat Wisata	Prioritas
		Fitur Logout	Prioritas
2	Admin	Fitur Pendaftaran	Prioritas
		Fitur Login	Prioritas
		Fitur Lupa Password	Prioritas
		Fitur Profile	Prioritas
		Fitur Membuat Paket Wisata	Prioritas
		Fitur Mengelola Informasi	Prioritas
		Fitur Menerima Booking Paket Wisata	Prioritas
		Fitur Pengumpulan Point	Opsional
		Fitur Mengelola Akun Wisatawan	Prioritas
		Fitur Mengelola Akun Umkm	Prioritas
Fitur Logout	Prioritas		
3	UMKM	Fitur Pendaftaran	Prioritas
		Fitur Login	Prioritas
		Fitur Lupa Password	Prioritas
		Fitur Profile	Prioritas
		Fitur Mengelola Pembayaran	Prioritas
		Fitur Mengelola Informasi	Prioritas
		Fitur Menambahkan Tempat Wisata	Prioritas
		Fitur Rincian Booking	Prioritas
Fitur Logout	Prioritas		
4	Media	Fitur Login	Opsional
		Fitur Lupa Password	Opsional
		Fitur Profile	Opsional
		Fitur Mendapatkan Informasi	Prioritas
		Fitur Logout	Opsional
5	Pemerintah	Fitur Pendaftaran	Prioritas
		Fitur Login	Prioritas
		Fitur Lupa Password	Prioritas
		Fitur Profile	Prioritas
		Fitur Mengelola Informasi Event	Prioritas
		Fitur Melihat Statistik Kunjungan	Prioritas
Fitur Logout	Prioritas		

Tabel 3 memperlihatkan hasil dari *product backlog* yang telah dikerjakan oleh tim *scrum*. *Product backlog* telah dilakukan sesuai dengan estimasi pengerjaan yang telah ditentukan yaitu selama 12 hari.

**Melakukan *sprint***

Tabel 4 memperlihatkan hasil *sprint* dari proses *sprint* yang sedang berlangsung. Hasilnya berupa kebutuhan sistem untuk pengguna Wisatawan yang digunakan pada sistem yang akan dibuat sesuai dengan *Product Backlog Item* yang sudah ada.

Tabel 4. *Sprint 1*

Tahap	Mulai	Hari	Berakhir
Sprint 1	4 Oktober 2021	3 Hari	6 Oktober 2021
Menyelesaikan fitur untuk Wisatawan			

Tabel 5. *Sprint 2*

Tahap	Mulai	Hari	Berakhir
Sprint 2	7 Oktober 2021	3 Hari	9 Oktober 2021
Menyelesaikan fitur untuk Admin			

Tabel 5 memperlihatkan hasil *sprint* dari proses *sprint* yang sedang berlangsung. Hasilnya berupa kebutuhan sistem untuk Admin yang digunakan pada sistem yang akan dibuat sesuai dengan *Product Backlog Item* yang sudah ada. *Sprint 2* terdapat tambahan *product backlog* yaitu tambahkan ke favorit, chat pengelola, fitur memberi komentar dan rating untuk Wisatawan.

Tabel 6. *Sprint 3*

Tahap	Mulai	Hari	Berakhir
Sprint 3	11 Oktober 2021	2 Hari	12 Oktober 2021
Menyelesaikan fitur untuk UMKM			

Tabel 6 memperlihatkan hasil *sprint* dari proses *sprint* yang sedang berlangsung. Hasilnya berupa kebutuhan sistem untuk pengguna UMKM yang digunakan pada sistem yang akan dibuat sesuai dengan *Product Backlog Item* yang sudah ada. *Sprint 3* terdapat tambahan *product backlog* yaitu fitur menambahkan tempat wisata dan penginapan untuk Admin.

Tabel 7. *Sprint 4*

Tahap	Mulai	Hari	Berakhir
Sprint 3	13 Oktober 2021	2 Hari	14 Oktober 2021
Menyelesaikan fitur untuk Media			

Tabel 7 memperlihatkan hasil *sprint* dari proses *sprint* yang sedang berlangsung. Hasilnya berupa kebutuhan sistem untuk pengguna Media yang digunakan pada sistem yang akan dibuat sesuai dengan *Product Backlog Item* yang sudah ada dan pada *sprint* ini tidak mengalami perubahan.

Tabel 8. *Sprint 5*

Tahap	Mulai	Hari	Berakhir
Sprint 3	13 Oktober 2021	2 Hari	14 Oktober 2021
Menyelesaikan fitur untuk Media			

Tabel 8 memperlihatkan hasil *sprint* dari proses *sprint* yang sedang berlangsung. Hasilnya berupa kebutuhan sistem untuk pengguna Pemerintah yang digunakan pada sistem yang akan dibuat sesuai dengan *Product Backlog Item* yang sudah ada. *Sprint 5* terdapat tambahan *product backlog* yaitu fitur mendapatkan informasi *event* untuk Media.

Hasil dari adopsi metode *scrum* ini hanya dilakukan sampai dengan tahap *daily scrum* yang di mana tiap harinya dilakukan *review* di akhir pertemuan dengan penentuan fitur prioritas yang akan digunakan pada perancangan sistem *smart tourism*.

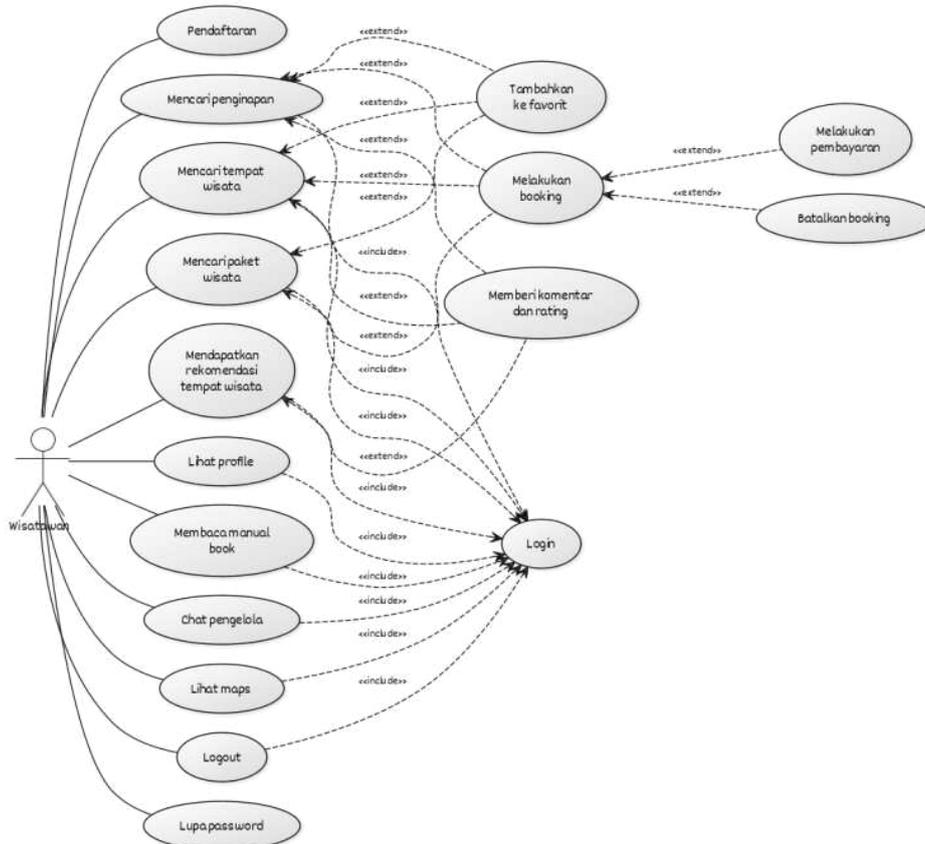
### Desain sistem

*Use case diagram* merupakan sebuah *tools* yang digunakan untuk membuat pemodelan interaksi user dengan sistem [9]. Ada dua elemen penting yang perlu digambarkan pada *use case*, yaitu aktor dan *use case*. Aktor biasanya didefinisikan dengan simbol gambar orang-orangan (*stick-man*) dengan nama kata benda di bagian bawah yang menyatakan peran/sistem. Sedangkan *use case* biasanya dinotasikan dengan simbol *elips* dengan nama kata kerja aktif di bagian dalamnya yang menyatakan aktivitas dari perspektif aktor [10]. Diagram model ini berisikan langkah-langkah hubungan antara pengguna dengan sistem dan terdapat kebutuhan fungsional yang saling terkait.

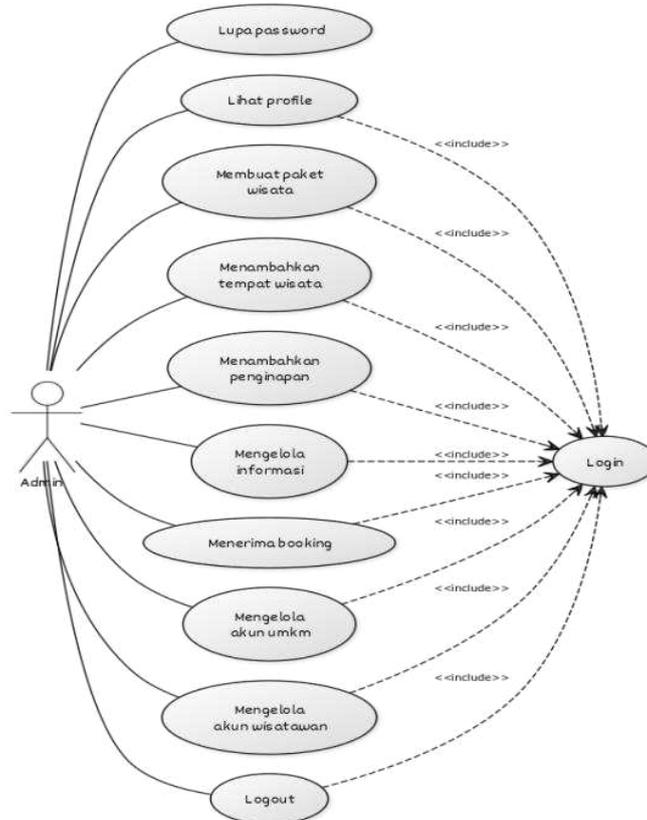
Gambar 3 memperlihatkan *use case diagram* rancangan fitur yang akan digunakan pada *smart tourism* di mana wisatawan memiliki peran untuk melihat dan melakukan pemesanan paket wisata serta mendapatkan rekomendasi objek wisata terdekat dengan lokasi wisatawan.

Gambar 4 memperlihatkan *use case diagram* rancangan fitur yang akan digunakan pada *smart tourism* di mana Admin memiliki peran untuk mengelola akun Wisatawan dan UMKM serta dapat menambahkan informasi pada sistem *smart tourism*.

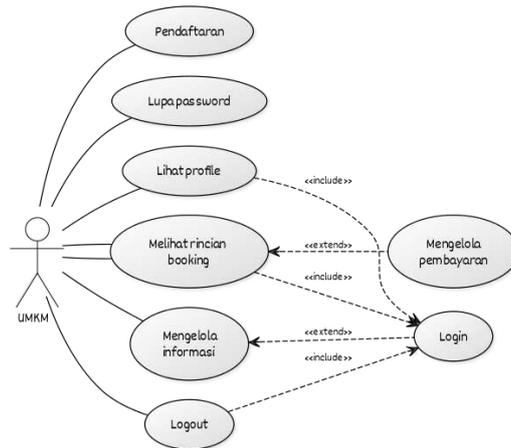
Gambar 5 memperlihatkan *use case diagram* rancangan fitur yang akan digunakan pada *smart tourism* di mana UMKM memiliki peran untuk mengelola informasi dan pembayaran atau pesanan yang dilakukan oleh Wisatawan.



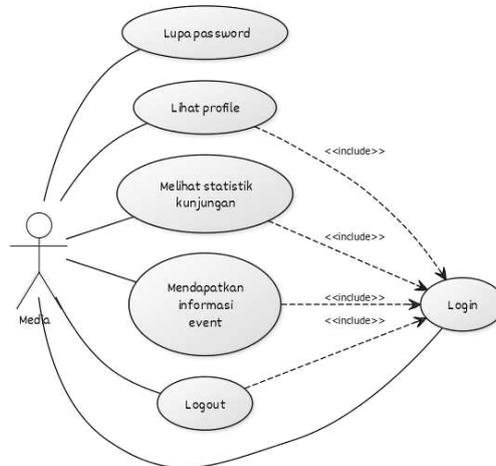
Gambar 3. Use case diagram Wisatawan



Gambar 4. Use case diagram Admin

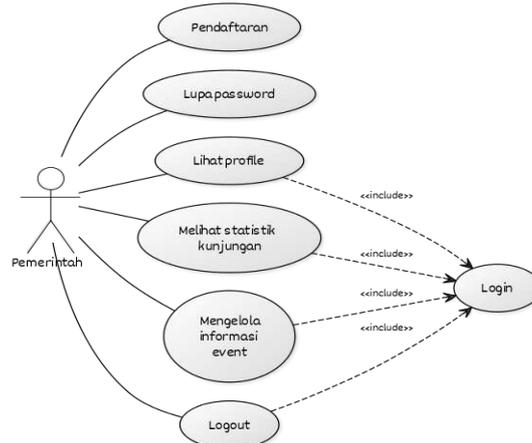


Gambar 5. Use case diagram UMKM



Gambar 6. Use case diagram Media

Gambar 6 memperlihatkan use case diagram rancangan fitur yang akan digunakan pada *smart tourism* di mana Media memiliki peran untuk melihat statistik kunjungan pada objek wisata dan mendapatkan informasi *event* pada sistem *smart tourism*.



Gambar 7. Use case diagram Pemerintah

Gambar 7 memperlihatkan use case diagram rancangan fitur yang akan digunakan pada *smart tourism* di mana Pemerintah memiliki peran untuk melihat statistik kunjungan pada objek wisata dan dapat mengelola informasi *event* pada sistem *smart tourism*.

#### 4. KESIMPULAN DAN SARAN

Berdasarkan hasil yang dipaparkan diatas maka dapat disimpulkan bahwa perancangan elisitasi kebutuhan *smart tourism* dengan mengadopsi metode *scrum* memudahkan tim pengembang dalam hal mengembangkan *smart tourism* yang dibuat menggunakan UML dengan fokus pada *use case diagram* yang telah dibuat berdasarkan hasil *sprint* yang telah dilakukan serta dengan mengadopsi metode *scrum* dalam menentukan hasil *requirement system* mengurangi terjadinya perubahan dan dapat meminimalisir potensi terjadinya perubahan dalam menentukan *requirement system* serta metode yang tanggap terhadap perubahan.

Adapun saran dari penulis yang dapat diberikan untuk penelitian selanjutnya yang mungkin bisa serupa dengan penelitian ini adalah dalam mengadopsi metode *scrum* untuk elisitasi kebutuhan *smart tourism* dapat dilanjutkan sampai dengan tahap pengembangan *smart tourism* untuk mengetahui proses pengembangan sistem yang adaptif.

#### DAFTAR PUSTAKA

- [1] D. M. Sari and A. S, "Implementasi Sistem Pengenalan Wisata Bahari Berbasis Website Kabupaten Majene," *Jurnal Teknik Informatika*, vol. 13, no. 2, pp. 123–132, 2021, doi: [10.15408/jti.v13i2.16956](https://doi.org/10.15408/jti.v13i2.16956).
- [2] N. K. Sukerti, "Sistem Pendukung Keputusan Menggunakan Simple Additive Weight (SAW) Dalam Merekomendasikan Obyek Wisata di Pulau Nusa Penida," in *Seminar Nasional Royal (SENAR)*, 2018, vol. 1, no. 1, pp. 93–98.
- [3] D. Danuri and J. Jaroji, "ELISITASI KEBUTUHAN PERANGKAT LUNAK REKRUTMEN PEGAWAI DENGAN PENDEKATAN SOFT SYSTEM METHODOLOGY," *Just TI (Jurnal Sains Terapan Teknologi Informasi)*, vol. 10, no. 2, pp. 42–45, 2019.
- [4] H. Saeeda, J. Dong, Y. Wang, and M. A. Abid, "A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project," *Journal of Software: Evolution and Process*, vol. 32, no. 7, pp. 1–24, 2020, doi: [10.1002/smr.2247](https://doi.org/10.1002/smr.2247).
- [5] S. Anwar, L. Andrawina, and A. F. Rizana, "Perancangan Sistem Informasi Untuk Pengelolaan Data Warga Dalam Tingkat Rt Dengan Metode Scrum Information System Design for Managing Data of Citizens in the Neighborhood Level with Scrum Method," vol. 7, no. 2, pp. 6137–6146, 2020.
- [6] R. Wulandari, R. Setiawan, and A. Mulyani, "Perancangan Sistem Informasi Manajemen Wedding Organizer Online Menggunakan Scrum," *Jurnal Algoritma*, vol. 16, no. 2, pp. 139–150, 2020, doi: [10.33364/algoritma/v.16-2.139](https://doi.org/10.33364/algoritma/v.16-2.139).
- [7] N. Abdillah, "Analisis Pengaruh Faktor Adopsi Scrum Pada Startup Digital," *Teknomatika*, vol. x, no. x, p. 1~5, 2019.
- [8] D. B. Kinasih and K. Kunci, "Pengembangan Sistem Informasi Pengelolaan Kinerja Karyawan (Studi Kasus: Modena Strategy System)," *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, vol. 1, 2021.
- [9] L. Setiyani, "Desain Sistem: Use Case Diagram Pendahuluan," *Prosiding Seminar Nasional: Inovasi & Adopsi Teknologi 2021*, no. September, pp. 246–260, 2021.
- [10] T. A. Kurniawan, "Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 1, p. 77, 2018, doi: [10.25126/jtiik.201851610](https://doi.org/10.25126/jtiik.201851610).