

## PENGEMBANGAN FITUR AUTENTIKASI UNTUK APLIKASI MANAJEMEN DAN PEMESANAN RUANGAN

I.W.Sugiana<sup>1</sup>, I.M Widiartha<sup>2</sup>, N.A.S ER<sup>3</sup>

### ABSTRAK

Dalam melakukan beberapa kegiatan diperlukan fasilitas berupa ruangan untuk melaksanakan kegiatan tersebut. Namun, ketersediaan ruangan dapat menjadi masalah apabila tidak terdapat manajemen ruangan yang baik. Oleh karena itu, untuk mengatasi permasalahan ini maka diperlukan sebuah aplikasi yang dapat mengatur ketersediaan ruangan dan melakukan proses pemesanan ruangan agar permasalahan ketersediaan ruangan dapat diatasi. Dalam mengembangkan aplikasi tersebut penulis menggunakan metode *Agile*. *Agile* adalah sebuah metode dalam *Software Development Life Cycle* yang dapat digunakan dalam mengembangkan suatu aplikasi. Selain itu, dalam pengembangan aplikasi, penulis juga mengimplementasikan metode *Black Box Testing* dalam proses pengujian aplikasi. Hasil pengujian yang didapatkan dengan metode tersebut adalah aplikasi yang dibangun telah memiliki fitur dan berjalan sesuai kebutuhan pengguna. Diharapkan dari pengembangan aplikasi manajemen ruangan ini dapat mengatasi permasalahan ketersediaan ruangan.

**Kata kunci :** Seluler, Flutter, Pemesanan, Ruangan, Manajemen.

### ABSTRACT

In carrying out some activities, room is needed to carry out these activities. But, the availability of rooms can be a problem if there is no good room management. Therefore, to overcome this problem, we need an application that can manage room availability and do the room reservation process so that the room availability problem can be solved. In developing the application, the writer used the Agile method. Agile is a method in Software Development Life Cycle that can be used to develop an application. In addition, in application development, the writer also implements the black box testing method in the application testing process. The test result obtained with this method is that applications that are built already have features and run according to user needs. It is hoped that developing this room management application can solve room availability problems.

**Keywords:** Mobile, Flutter, Booking, Room, Management.

---

<sup>1</sup> Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana, wayansugiana777@gmail.com

<sup>2</sup> Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana, madewidiartha@unud.ac.id

<sup>3</sup> Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana, agus\_sanjaya@unud.ac.id

## **1. PENDAHULUAN**

Pada perkembangan zaman yang serba digital saat ini memberikan banyak perubahan. Perkembangan teknologi mempengaruhi perubahan konsep: ruang kerja, kantor, tempat tinggal, belanja, hotel, sekolah dan ruang publik lainnya (Ray, 2018). Ruangan menjadi fasilitas penunjang yang penting untuk beberapa kegiatan. Oleh karena itu manajemen yang baik diperlukan agar tidak terjadi kesalahan pengaturan jadwal penggunaan ruangan. Untuk mengatasi permasalahan tersebut, diperlukan sebuah sistem yang mampu mengatur ketersediaan ruangan dan melakukan proses pemesanan ruangan.

Industri mobile app telah berkembang pesat dalam dekade terakhir dan diperkirakan akan terus meningkat (Moshen et al., 2022). Oleh karena itu, sistem ini akan dikembangkan pada perangkat mobile dengan framework Flutter dalam pengembangannya. Selain itu dalam proses pengembangan aplikasi, penulis menggunakan metode *Agile* yang merupakan salah satu metode dalam *software development life cycle*. Di akhir pengembangan, penulis melakukan *testing* terhadap aplikasi yang dibangun menggunakan metode *Black Box Testing*.

## **2. METODE PELAKSANAAN**

Penelitian yang dilakukan oleh Standish Group pada tahun 2002 - 2010 menunjukkan bahwa *Agile* merupakan metode pengembangan software yang memiliki tingkat kesuksesan yang lebih besar dibandingkan metode lain seperti metode *Waterfall*. Dari penelitian tersebut didapatkan hasil tingkat kesuksesan dengan menggunakan metode *Agile* adalah sebesar 42%, sedangkan tingkat kesuksesan dengan menggunakan metode *Waterfall* hanya sebesar 14% (Murdiani et al., 2020). *Agile* dikembangkan untuk meningkatkan dan mengatasi kekurangan efisiensi pengembangan *software*. Metodologi *Agile* meningkatkan interaksi antara *developer* dan klien untuk membuat produk *software* bebas cacat (Butt et al., 2023). Oleh karena itu, dalam pengembangan aplikasi, penulis akan menggunakan metode *Agile*.

Dalam pengembangan sistem, penulis mengerjakan fungsi autentikasi. Autentikasi merupakan permasalahan yang penting saat ini karena popularitas teknologi komputer (Chen et al., 2022). Aplikasi dan layanan biasanya memiliki satu titik autentikasi di mana pengguna memasukkan kata sandi mereka (Martin et al., 2022). Dalam beberapa dekade terakhir, *user authentication system* (disingkat *authentication system*) telah banyak digunakan dalam kehidupan sehari-hari untuk secara efektif memblokir akses ilegal ke layanan dan data sensitif (Wang et al., 2021). Oleh karena itu, diperlukan sistem autentikasi untuk mengidentifikasi dan memverifikasi pengguna agar aktivitas pengguna dalam penggunaan aplikasi dapat terverifikasi dan aman. Sistem autentikasi yang akan dibangun oleh penulis terdiri dari sistem *login*, sistem pendaftaran akun, dan sistem lupa *password*.

Karena bagian utama dari software dikodekan oleh manusia sejauh ini, cacat akan selalu ada (Cao et al., 2020). Oleh karena itu, selain mengembangkan sistem, penulis juga melakukan *testing* terhadap halaman yang penulis kembangkan. *Testing* sangat penting untuk memastikan *reliability* dari sistem software (Sunman et al., 2022). Aktivitas testing memastikan kesesuaian dari produk software dengan requirements yang didefinisikan oleh customer melalui eksekusi yang sistematis (Barbosa et al., 2022). Adapun metode yang digunakan oleh penulis adalah metode *black box testing*.

## **3. HASIL DAN PEMBAHASAN**

Setelah menganalisis kebutuhan pengguna, penulis kemudian mengimplementasikan fitur-fitur yang didapatkan dari analisis. Setelah mengimplementasikan fitur-fitur tersebut, penulis melakukan proses testing untuk melakukan pengujian terhadap hasil dari implementasi yang telah dilakukan.

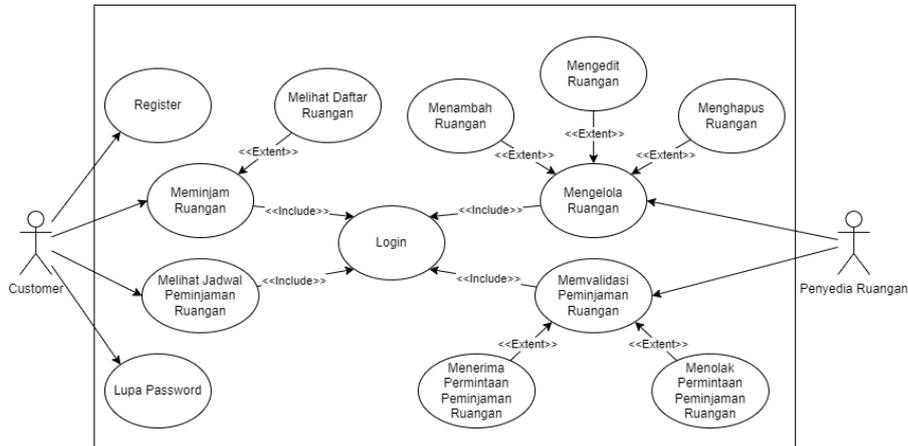
### **3.1. Kebutuhan Pengguna**

Dalam proses implementasi, penulis mengimplementasikan fitur-fitur sesuai kebutuhan dari pengguna. Adapun kebutuhan pengguna yang didapatkan dari analisis adalah fitur login untuk melakukan proses login, fitur

registrasi akun untuk mendaftarkan akun baru dan menyimpan data akun tersebut ke database, serta fitur lupa password untuk mereset password dengan mengirimkan email verifikasi ke akun email pengguna.

### 3.2. Rancangan Aplikasi

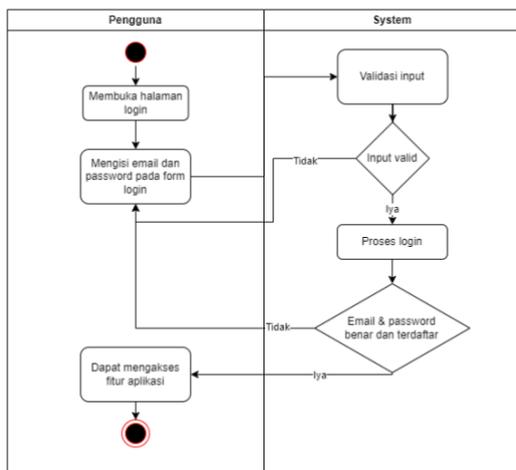
Setelah mendapatkan kebutuhan aplikasi, penulis membuat *use case diagram* untuk memvisualisasikan *flow* dari fitur yang akan dikembangkan. *Diagram use case* digambarkan pada gambar 3.2.1.



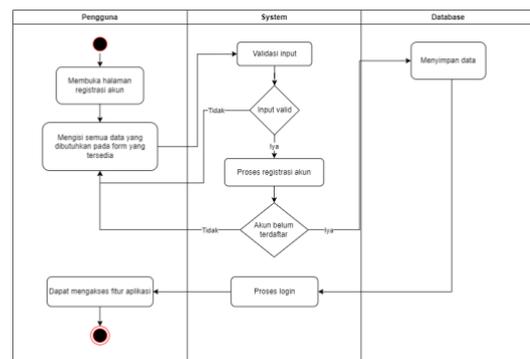
Gambar 3.2.1. Use Case Diagram dari Fitur yang Dikerjakan

Dikarenakan dalam penelitian ini berfokus pada pengembangan fitur autentikasi maka pada pembahasan hasil akan berfokus pada fitur autentikasi yakni *login*, registrasi akun, serta lupa *password*.

Selain *use case diagram*, penulis juga membuat *activity diagram* untuk menggambarkan *flow* secara lebih mendetail untuk setiap fitur. *Activity diagram* dari fitur yang dikembangkan digambarkan pada gambar 3.2.2, gambar 3.2.3, dan gambar 3.2.4.

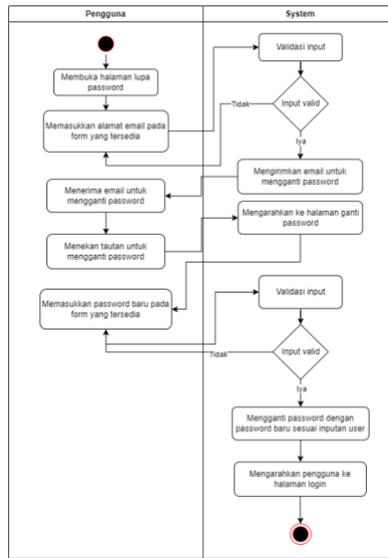


Gambar 3.2.2. Activity Diagram Fitur Login



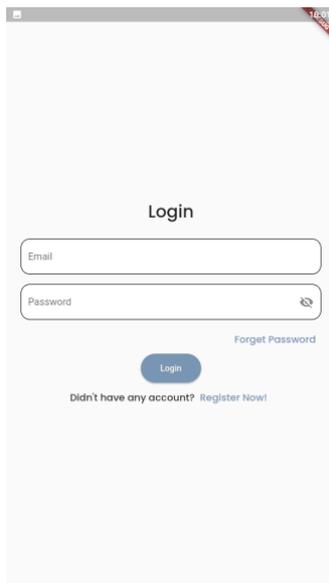
Gambar 3.2.3. Activity Diagram Fitur Registrasi Akun

**Pengembangan fitur autentikasi untuk aplikasi manajemen dan pemesanan ruangan**

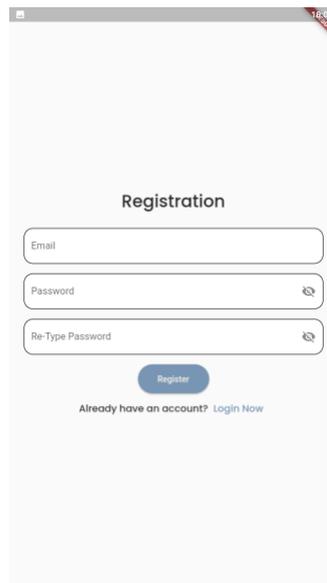


**Gambar 3.2.4.** Activity Diagram Fitur Lupa Password

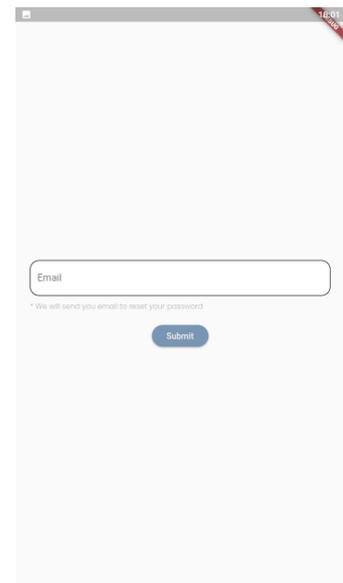
Adapun hasil yang didapatkan dari kegiatan yang telah dilakukan oleh penulis adalah halaman login, halaman pendaftaran akun, dan halaman lupa password pada aplikasi manajemen dan pemesanan ruangan yang ditunjukkan pada gambar 3.2.5, gambar 3.2.6, dan gambar 3.2.7.



**Gambar 3.2.5.** Tampilan Halaman Login



**Gambar 3.2.6.** Tampilan Halaman Registrasi Akun



**Gambar 3.2.7.** Tampilan Halaman Lupa Password

Gambar 3.2.5 menunjukkan halaman login yang dikerjakan penulis. Halaman ini memiliki fungsi validasi inputan untuk mengecek apakah validasi input sudah valid. Apabila semua *input* sudah sesuai maka aplikasi akan melanjutkan proses *login* dan apabila *email* dan *password* sesuai dan telah terdaftar maka *login* akan berhasil dan *user* diarahkan ke *home page*.

Gambar 3.2.6 menunjukkan halaman registrasi yang dikerjakan penulis. Halaman registrasi ini memiliki fungsi untuk validasi input *email*, *password*, dan *retype password*. Apabila *user* memasukkan alamat *email* yang tidak sesuai maupun tidak mengisi *field email* maka akan muncul pesan bahwa *input* tidak *valid*. Apabila *user*

memasukkan *password* kurang dari 7 karakter atau tidak mengisi *field password* maka akan muncul pesan bahwa *input* tidak *valid*. Apabila *user* memasukkan *retype password* yang tidak sama dengan *text* pada *field password* atau tidak mengisi *field retype password* maka akan muncul pesan bahwa *input* tidak *valid*. Apabila semua *input* sudah sesuai maka aplikasi akan melanjutkan proses registrasi akun dan apabila akun belum terdaftar maka akun akan didaftarkan dan apabila berhasil, *user* akan diarahkan ke *home page*.

Gambar 3.2.7 menunjukkan halaman lupa *password* yang dikerjakan penulis. Halaman ini memiliki fungsi untuk validasi *input email*. Apabila *user* memasukkan alamat *email* yang tidak sesuai maupun tidak mengisi *field email* maka akan muncul pesan bahwa *input* tidak *valid*. Bila *input email* sudah *valid*, maka proses lupa *password* akan dijalankan dimana hasil akhirnya berupa *server* akan mengirimkan *email* untuk mengganti *password user*.

### 3.3. Hasil Pengujian

Setelah selesai mengimplementasikan fitur sesuai kebutuhan pengguna, penulis melakukan testing dengan metode *Black Box Testing*. Adapun skenario uji yang dilakukan adalah pengujian fitur pendaftaran akun, fitur login, dan fitur lupa *password*. Hasil *Black Box Testing* dijelaskan pada tabel 3.3.1.

Tabel 3.3.1. Tabel Hasil Black Box Testing

No	Skenario Pengujian	Kasus Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Pengguna membuat akun baru	Masukkan semua data pada form yang disediakan kemudian tekan tombol daftar.	Akun berhasil didaftarkan dan data disimpan ke <i>database</i> .	Berhasil
2	Pengguna melakukan proses login	Masukkan semua data pada form yang disediakan kemudian tekan tombol login.	Akun berhasil <i>login</i> dan user diarahkan ke <i>home page</i> .	Berhasil
3	Pengguna menjalankan fungsi lupa <i>password</i>	Masukkan data email pada form yang disediakan kemudian tekan tombol lupa <i>password</i> .	User dikirimkan email untuk melakukan reset <i>password</i> .	Berhasil

## 4. KESIMPULAN DAN SARAN

Hal yang dapat disimpulkan adalah pengembangan fungsi autentikasi berupa fungsi login dan registrasi akun telah dikembangkan dan diuji dengan metode *Black Box Testing*. Hasil pengujian menunjukkan bahwa aplikasi yang dikembangkan telah memiliki fitur dan berjalan sesuai kebutuhan pengguna. Fitur ini diharapkan dapat membantu proses autentikasi *user* ketika menggunakan aplikasi manajemen dan pemesanan ruangan yang dikembangkan oleh CV. Avatar Solution.

## UCAPAN TERIMA KASIH

Terima kasih penulis ucapkan kepada semua pihak yang telah membantu selama penulis melaksanakan kegiatan praktek kerja lapangan. Berkat bantuan dari pihak-pihak tersebut, penulis dapat menyelesaikan kegiatan praktek kerja lapangan dengan baik.

## DAFTAR PUSTAKA

- Ray, P.P. (2018). A Survey on internet of things architectures. *Journal of King Saud University - Computer and Information Sciences*, Vol. 30: No. 3, pp. 291-319.
- Murdiani, D. et. al. (2020). Implementasi Agile Method dalam Pengembangan Jurnal Elektronik di Lembaga Penelitian Non Pemerintahan (NGO). *JTIK: Jurnal Teknologi Informasi dan Ilmu Komputer*. Vol. 7: No. 4, pp. 709-718.
- Chen, J.-A. et. al. (2022). University librarians' perspectives on an importance-performance analysis of authentication system attributes and their attitudes towards authentication log visualization. *The Journal of Academic Librarianship*. Vol. 48: No. 4.
- Butt, S. A. et. al. (2023). Prediction based cost estimation technique in agile development. *Advances in Engineering Software*. Vol. 175.
- Sunman, N., et. al. (2022). Automated web application testing driven by pre-recorded test cases. *Journal of Systems and Software*. Vol. 193, p. 111441.
- Moshen, F. et. al. (2022). Early detection of violating mobile apps: A data-driven predictive model approach. *Systems and Soft Computing*. Vol. 4, p. 200045.
- Wang, X. et. al. (2021). Attacks and defenses in User Authentication Systems: A survey. *Journal of Network and Computer Applications*, Vol. 188, p. 103080.
- Martin, A.G. et. al. (2022). Combining user behavioural information at the feature level to enhance continuous authentication systems. *Knowledge-Based Systems*, Vol. 244, p. 108544.
- Cao, P., et. al. (2020). Optimal selection and release problem in software testing process: A continuous time stochastic control approach, *European Journal of Operational Research*, Vol. 285: No.1, pp. 221-222.
- Barbosa, G. et al. (2022). A systematic literature review on prioritizing software test cases using Markov chains, *Information and Software Technology*, Vol. 147, p. 106902.