

Development of Information System using Scrum Model (Case study: Al-Muayyad Windan Islamic Boarding School)

Arafik Nur Fadliansah¹, Nurcahya Pradana Taufik Prakisyah^{1*} , Yusfia Hafid Aristyagama¹ ,
Jimsan²

¹ Universitas Sebelas Maret, Indonesia.

² Universitas Sembilanbelas November Kolaka, Indonesia.

* Corresponding Author. E-mail: nurcahya.ptp@staff.uns.ac.id

Article History

Received:

May 21st, 2025

Revised:

Oct 5th, 2025

Accepted:

Oct 5th, 2025

Keywords

Agile Development;
Islamic Boarding
School; Management
Information System;
Scrum Model; System
Usability.

ABSTRACT

Islamic boarding schools (*pesantren*) are Islamic educational institutions in Indonesia that must adapt to technological developments. Developing a digital-based management information system is one of the solutions to support effective data management and administrative processes. However, data collection at Al-Muayyad Windan Islamic Boarding School is still carried out conventionally, which often causes inefficiency in managing student information. Therefore, this research aims to develop a student management information system using an agile software development methodology, namely the Scrum model, which allows iterative development and continuous feedback. The implementation of Scrum was successfully carried out through stages including Product Backlog, Sprint Planning, Sprint Backlog, Sprint, Sprint Review, and Retrospective. The resulting system has passed feasibility testing and usability evaluation. The SUS test involving 10 respondents produced an average usability score of 73, categorized as Acceptable on the Acceptability Range, Grade C on the Grade Scale, and Good on the Adjective Rating. These results indicate that the developed system is functionally feasible, user-friendly, and capable of supporting digital transformation in Islamic boarding school management.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



PENDAHULUAN

Pondok pesantren sebagai lembaga pendidikan Islam di Indonesia, memiliki hubungan erat dengan karakteristik masyarakat Indonesia yang mengedepankan nilai-nilai strategis [1]. Santri, sebagai bagian integral dari masyarakat dan pondok pesantren, memiliki peran penting dalam mempertahankan sekaligus mengembangkan nilai-nilai tersebut. Seiring dengan pesatnya perkembangan teknologi informasi dan komunikasi [2], [3], [4], berbagai sektor pendidikan kini dituntut untuk beradaptasi dengan sistem digital yang lebih efisien dan terintegrasi [5], [6], [7], [8]. Pondok pesantren, sebagai lembaga pendidikan tradisional, juga perlu mengikuti perkembangan tersebut agar mampu bersaing dan tetap relevan di era transformasi digital. Namun, pengelolaan data akademik di banyak pondok pesantren masih menghadapi kendala, terutama karena proses pencatatan yang dilakukan secara manual melalui buku data santri, sehingga menimbulkan kesulitan dalam pencarian dan pengarsipan data baik bagi santri yang sudah lulus maupun yang masih aktif [9]. Temuan serupa diperoleh dari hasil wawancara dengan santri Pondok Pesantren Al-Muayyad Windan Sukoharjo, di mana sistem pendataan masih

dilakukan secara manual. Setiawan dan Sulaksono mencatat bahwa pengembangan sistem manajemen modern berbasis teknologi dapat membantu pondok pesantren beradaptasi dengan perkembangan zaman [10].

Salah satu wujud pengembangan manajemen di pondok pesantren adalah pengembangan sistem informasi manajemen yang mencakup data santri, berita, materi, jadwal pembelajaran, dan sebagainya. Sebelum memulai perancangan sistem informasi, penting untuk memilih metodologi pengembangan yang sesuai [11]. Secara garis besar, metodologi pengembangan sistem informasi dapat dibagi menjadi tradisional dan agile. Model tradisional, seperti model *waterfall*, cocok untuk sistem generik dengan kebutuhan yang dapat diidentifikasi dari awal [12]. Di sisi lain, metodologi *agile*, seperti model Scrum, berfokus pada fase perencanaan awal dan melibatkan interaksi iteratif dan inkremental sepanjang siklus proyek [13].

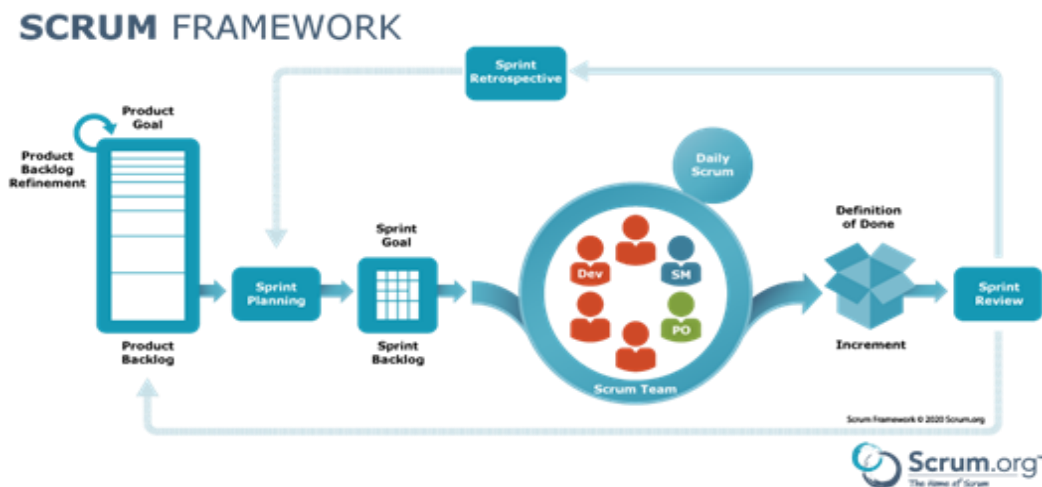
Observasi dan wawancara di Pondok Pesantren Al-Muayyad Windan Sukoharjo mengindikasikan kompleksitas pengembangan sistem informasi manajemen santri dan kebutuhan akan umpan balik berkelanjutan untuk mengatasi perubahan fitur yang tidak sesuai harapan. Model *scrum* dianggap memenuhi kebutuhan ini dengan kelebihan dalam memonitor dan mengontrol aktivitas peningkatan pekerjaan serta mengatasi perubahan dengan mudah [7]. Oleh karena itu, peneliti memilih model Scrum sebagai metodologi yang adaptif dan fleksibel sesuai dengan temuan observasi dan wawancara.

Penelitian ini bertujuan untuk mengembangkan sistem informasi manajemen santri al-muayyad windan menggunakan model scrum yang selanjutnya dapat dilakukan uji kelayakan dan evaluasi kegunaan pada sistem dengan harapan dapat berkontribusi pada pondok pesantren dengan adanya sistem tersebut sehingga dapat terbantu.

METODE PENELITIAN

2. 1 Model Scrum

Model pengembangan SDLC (*software development life cycle*) yang digunakan dalam penelitian ini adalah metode *Agile* dengan model *Scrum* (Gambar 1). Metode Agile dengan model Scrum digunakan dalam penelitian ini karena dalam pengembangan sistemnya metode ini banyak digunakan untuk mengelola manajemen dan pelaksanaan proyek karena memiliki produktivitas yang tinggi. Scrum bisa digunakan untuk pengembangan produk apapun, karena proses perencanaannya yang mudah dipahami dan pengembangannya yang fleksibel.



Gambar 2. Tahapan Proses Scrum [14]

Adapun tahapan-tahapan yang dilakukan dalam model *scrum* adalah sebagai berikut:

a. *Product Backlog*

Product backlog adalah sebuah daftar dari kebutuhan-kebutuhan yang perlu dipahami dan diketahui. Pada tahapan ini penulis melakukan pendefinisian dari kebutuhan pondok pesantren, pendeskripsian kebutuhan ini bersifat dinamis yang berarti dapat berubah-ubah.

b. *Sprint Planning*

Pada sesi selanjutnya, tim segera melakukan analisis pekerjaan yang akan dilakukan selama *sprint*. Pekerjaan dipilih dari *product backlog*. Saat *sprint planning*, tim merinci *product backlog* menjadi tugas-tugas kecil.

c. *Sprint Backlog*

Sprint backlog berisi semua catatan pekerjaan yang telah disepakati selama *sprint planning*. Jika ada pekerjaan baru, akan ditambahkan ke dalam *sprint backlog*. Pada tahapan ini *sprint backlog* sudah dalam bentuk kumpulan task yang dikerjakan dalam satu kali *sprint* dengan estimasi pengerjaan.

d. *Sprint*

Sprint merupakan iterasi atau siklus dalam satuan waktu terkait rencana task yang akan dikerjakan. *Sprint* dilakukan guna menyelesaikan sisa pekerjaan dari *sprint backlog* yang telah disusun berdasarkan estimasi penyelesaian dalam tiap kali mengerjakan *sprint*.

e. *Sprint Review*

Tahapan ini dilakukan setelah durasi *sprint* berakhir. Ketika sebuah *sprint* mencapai definisi penyelesaian yang telah disepakati sebelumnya, dianggap telah selesai. Dalam pertemuan tinjauan *sprint*, tim memperlihatkan kepada pemangku kepentingan pekerjaan yang telah dilakukan selama *sprint*.

f. *Sprint Retrospective*

Sesi ini diadakan setelah mencapai tinjauan *sprint*. Dalam pertemuan ini, tim *scrum* akan menetapkan pekerjaan yang akan dilakukan dalam rencana *sprint* berikutnya. Pekerjaan untuk *sprint* berikutnya merujuk pada lingkup pekerjaan yang sudah disepakati di awal proyek.

2. 2 Teknik Pengumpulan Data

a. Studi Pustaka

Studi Pustaka di sini kegiatan yang dilakukan untuk memperoleh informasi dan wawasan yang lebih luas tentang sebuah masalah, dengan cara membaca dan mempelajari literatur yang berkaitan dengan sistem informasi manajemen santri.

b. Wawancara

Wawancara ini dilakukan dengan tujuan mendapatkan informasi mengenai kebutuhan apa saja yang diperlukan dalam mengembangkan sebuah sistem Model wawancara yang digunakan pada penelitian ini adalah wawancara tidak terstruktur. Responden dalam wawancara ini adalah santri dan pengurus Pondok Pesantren Al-Muayyad Windan dengan wawancara yang dibagi menjadi 2 tahap, yaitu sebelum penelitian dan selama proses pengembangan sistem

c. Angket

Angket merupakan teknik pengumpulan data yang berisi pertanyaan atau pernyataan kepada responden secara tertulis [8]. Kegunaan angket dalam penelitian ini adalah untuk menghitung tingkat validasi maupun kelayakan suatu media yang telah dilakukan penulis yaitu pada sistem informasi. Angket yang digunakan merupakan angket dengan metode *System Usability Scale (SUS)* dan *Black box testing*.

2. 3 Teknik Evaluasi Penelitian

a. Analisis *burndown chart*

Pada tahapan *sprint retrospective*, menggunakan analisis dengan *burndown chart* untuk evaluasi setiap *iterasi sprint*. Yang didapat dari hasil tabel penghitungan *burndown chart sprint*.

b. Uji Kelayakan *Black Box testing*

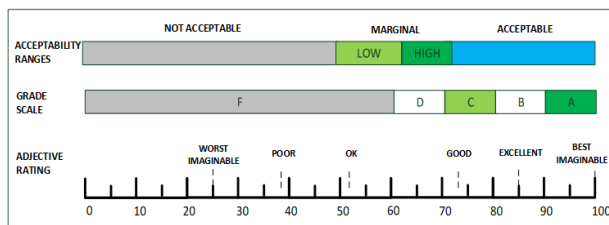
Uji Kelayakan dengan metode *black box testing* berfungsi untuk mendapatkan informasi sistem yang dikembangkan apakah berjalan dengan baik atau belum. Pengujian ini dilakukan berdasar pada fungsional dan hasil yang diharapkan sehingga dapat memvalidasi apakah sistem sesuai dengan harapan pengguna

c. Evaluasi SUS

System Usability Scale (SUS) digunakan sebagai evaluasi kegunaan pada sistem informasi manajemen santri. Evaluasi dilakukan dengan menggunakan angket berbasis *System Usability Scale (SUS)* yang ditujukan kepada pengguna yaitu santri dan pengurus Pondok Pesantren Al-Muayyad Windan. Menurut Elma *System Usability Scale* terdiri dari 10 pernyataan yaitu (angka ganjil pernyataan positif; angka genap pernyataan negatif) dengan pilihan jawaban skala 1-5, mulai dari sangat tidak setuju, tidak setuju, netral, setuju, sangat setuju [9]. Berikut ini adalah [Tabel 1](#) yang menunjukkan daftar pernyataan angket berbasis teknik analisis *System Usability Scale (SUS)* pada [Gambar 2](#). Dalam menentukan hasil evaluasi sistem dengan menggunakan *acceptability, grade scale, dan adjective rating*, dilakukan berdasarkan hasil rekapitulasi nilai kumulatif rerata instrumen yang diperoleh dari responden. Selain itu, untuk menentukan hasil kesimpulan juga dapat dilakukan hanya dengan menggunakan *grade scale* melalui ketentuan untuk rerata sebagai berikut pada [Tabel 1](#)[10].

Tabel 1. *Grade Scale*

Grade	Keterangan
A	Skor $\geq 80,3$
B	Skor ≥ 74 dan $< 80,3$
C	Skor ≥ 68 dan < 74
D	Skor ≥ 51 dan < 68
F	Skor < 51



Gambar 3. Penilaian SUS Score

HASIL DAN PEMBAHASAN

3.1 Tahapan *Scrum*

a. *Product Backlog*

Product Backlog dibuat dengan mengacu pada *functional requirements* yang telah ditulis pada perencanaan awal kebutuhan dan analisis terhadap wawancara pada pihak pondok pesantren.

Daftar *requirements* bersifat dinamis, sehingga akan bertambah apabila mendapat *feedback* dari pihak pondok pesantren pada saat demo hal tersebut dapat dilihat pada [Tabel 2](#).

Tabel 2. *Product Backlog*

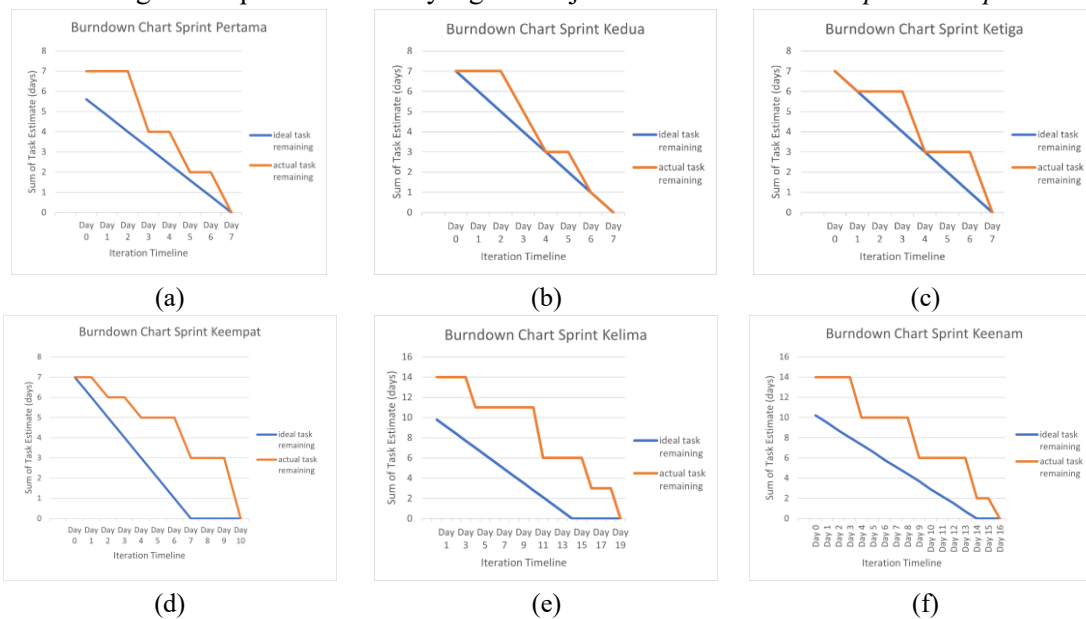
ID	Backlog Item	Prioitas	Estimasi
1.1	Membuat rancangan ERD 69tati, <i>usecase diagram</i> , dan <i>flowchart</i>	Tinggi	1.5
1.2	Membuat <i>wireframes</i> , UI/UX <i>mobile apps</i> , dan <i>web apps</i>	Tinggi	1.5
1.3	Pengkodean halaman pada <i>web apps</i> seperti <i>login</i> , <i>dashboard</i> , dan data santri	Tinggi	2
1.4	Pengkodean halaman pada <i>mobile apps</i> seperti <i>login</i> , <i>dashboard</i> , dan profil	Tinggi	2
2.1	Perbaiki <i>requirements sprint 1</i>	Sedang	1
2.2	Mengimplementasikan fungsi <i>login multi role</i> pada <i>web apps</i>	Tinggi	2
2.3	Mengimplementasikan fungsi CRUD pada data santri <i>web apps</i>	Tinggi	2
2.4	Membuat UI/UX <i>mobile apps</i> halaman berita, detail berita, materi, detail materi, rekap, dan detail rekap	Tinggi	2
3.1	Membuat UI/UX <i>web apps</i> data guru, data admin, dan data materi	Tinggi	1
3.2	Pengkodean halaman <i>mobile apps</i> , seperti berita, detail berita, materi, detail materi, rekap, dan detail rekap	Tinggi	1.5
3.3	Mengimplementasikan fungsi CRUD untuk data guru, dan data admin	Tinggi	3
3.4	Mengimplementasikan fungsi CRUD untuk data materi	Tinggi	1.5
4.1	Membuat UI/UX <i>mobile apps</i> untuk halaman edit profil, dan notifikasi	Tinggi	1
4.2	Pengkodean halaman edit profil dan notifikasi di <i>mobile apps</i>	Tinggi	1
4.3	Implementasi REST API untuk <i>login</i> , detail profil, update profil, dan data materi	Tinggi	2
4.4	Mengimplementasikan fungsi <i>login</i> , data materi, profil, dan update profil pada <i>mobile apps</i>	Tinggi	3
5.1	Implementasi fitur <i>upload</i> gambar profil santri dan <i>logout mobile apps</i>	Tinggi	2.5
5.2	Implementasi fitur menu absensi kelas santri untuk guru pada <i>web apps</i>	Tinggi	2.5
5.3	Implementasi REST API data berita, data rekap, dan <i>upload gambar profil</i>	Tinggi	3
5.4	Mengimplementasikan fitur data berita, dan data rekap pada <i>mobile apps</i>	Tinggi	3
5.5	Kustomisasi <i>firebase</i> dan implementasi <i>login google</i> dengan <i>firebase</i>	Tinggi	3
6.1	Perbaiki <i>requirements sprint 5</i>	Sedang	4
6.2	Implementasi fitur notifikasi, dan <i>dashboard</i> pada <i>mobile apps</i>	Sedang	4
6.3	Mengimplementasikan 69tatic data pada <i>dashboard web apps</i>	Sedang	4
6.4	<i>Deployment web apps</i> ke server hosting	Rendah	2

b. Sprint Planning

Diperlukan tahap *sprint planning* atau perencanaan berapa kali *sprint* yang akan digunakan dalam scrum. Pengembang membagi menjadi enam kali *sprint* dengan tiap kali *sprint* memiliki acuan dalam rentang waktu antara 1 atau 2 minggu lamanya pengerjaan. Kemudian langkah selanjutnya adalah mengelompokkan tiap daftar *product backlog* tersebut menjadi sebuah *sprint backlog*.

c. Sprint Backlog

Sprint backlog merupakan subset dari *product backlog* yang perlu diselesaikan guna mendapatkan hasil produk akhir atau tujuan dari *sprint*, yaitu terciptanya Sistem Informasi Manajemen Santri Al-Muayyad Windan. Setiap *sprint backlog* memiliki estimasi waktu pengerjaan yang dibutuhkan pengembang untuk mengerjakan tiap task. Detail pengerjaan tiap *sprint backlog* dilihat pada Gambar 3 yang menunjukkan *burndown chart* sprint 1 – sprint 6:



Gambar 3. Burndown chart Sprint 1 (a), 2(b), 3(c), 4(d), 5(e), dan 6(f)

d. Sprint

Proses pengembangan dilakukan melalui enam sprint dengan estimasi waktu yang berbeda pada setiap tahap. Pada Sprint 1, peneliti mengerjakan komponen dasar seperti perancangan ERD tabel, *flowchart*, *wireframe*, serta pengkodean beberapa halaman pada web dan *mobile apps* dengan estimasi waktu selama tujuh hari. Selanjutnya, pada Sprint 2, fokus pengerjaan diarahkan pada pengembangan fitur *login* multi-role dan fungsi CRUD (*Create, Read, Update, Delete*) pada *web apps*. Pada tahap ini juga terdapat penambahan *backlog item* berupa perbaikan *requirements* dari *sprint backlog* sebelumnya yang kemudian dimasukkan ke dalam sprint kedua, dengan estimasi waktu pengerjaan tujuh hari. Pada Sprint 3, peneliti mulai mengerjakan aspek UI/UX untuk *web apps*, merancang tampilan dasar pada *mobile apps*, serta mengimplementasikan beberapa fungsi CRUD, yang juga diselesaikan dalam waktu tujuh hari. Berikutnya, pada Sprint 4, fokus pengerjaan beralih pada pengembangan UI/UX untuk *mobile apps*, implementasi REST API, serta penerapan fungsi *login* pada aplikasi mobile dengan estimasi waktu tujuh hari. Pada Sprint 5, peneliti menambahkan fitur *update data* pada *mobile apps*, fitur absensi pada *web apps*, serta melakukan implementasi REST API dan kustomisasi Firebase, dengan estimasi waktu pengerjaan selama empat belas hari. Kemudian, pada Sprint 6, dilakukan pengembangan fitur notifikasi pada *mobile apps*, fitur statistik untuk *web apps*, serta proses

deployment. Tahap ini juga mencakup penambahan *backlog item* berupa perbaikan *requirement* dari sprint sebelumnya, dengan estimasi waktu pengerjaan empat belas hari. Selama keseluruhan proses iterasi, peneliti melaksanakan Sprint Review setelah menyelesaikan setiap tugas pada *sprint backlog*, baik melalui pertemuan daring menggunakan *Google Meet* maupun tatap muka langsung di pondok pesantren, di mana peneliti memperoleh umpan balik (*feedback*) secara langsung untuk perbaikan pada sprint berikutnya.

3.2 Perancangan Sistem

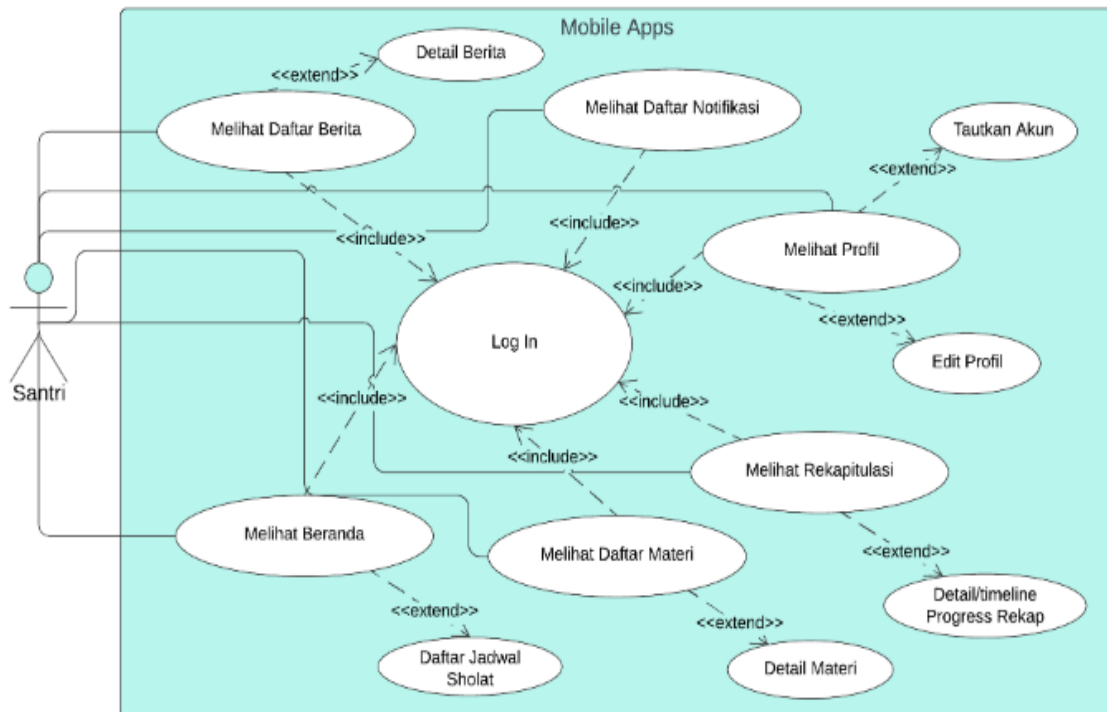
Perancangan sistem ini dilakukan secara bertahap selama proses *sprint* berlangsung agar setiap komponen dapat dikembangkan secara terarah dan terintegrasi.

a. Usecase diagram

Salah satu tahapan penting dalam perancangan adalah penyusunan **use case diagram**, yang berfungsi untuk menggambarkan hubungan antara aktor dengan sistem berdasarkan kebutuhan fungsional. Diagram ini menjelaskan bagaimana pengguna berinteraksi dengan sistem serta tanggung jawab sistem dalam merespons setiap aksi pengguna.

1) Mobile apps

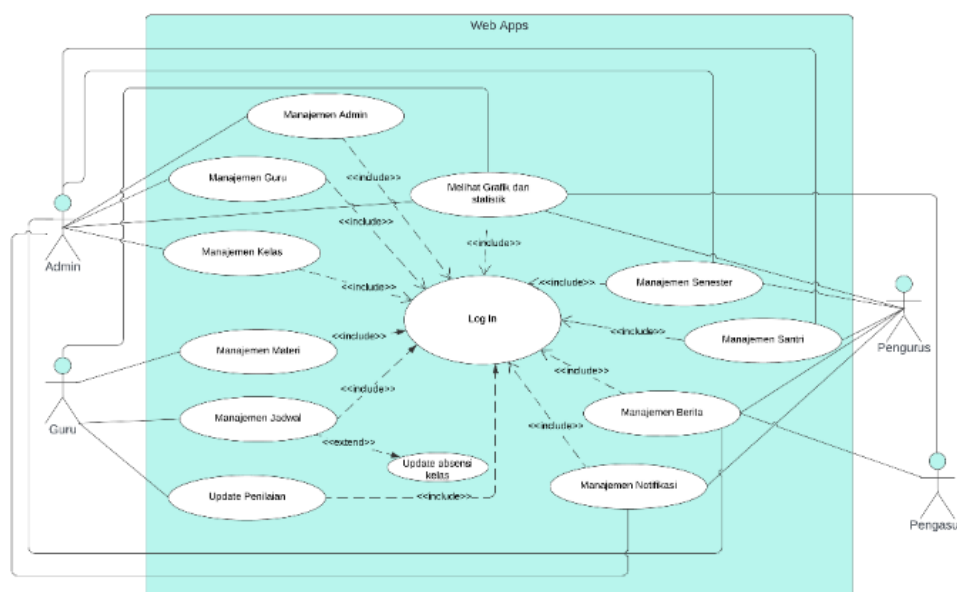
Pada **Gambar 4** menampilkan *Use Case Diagram Mobile Apps* yang menunjukkan interaksi antara pengguna dengan sistem melalui berbagai fitur utama, seperti melakukan *login*, melihat beranda, daftar berita, dan notifikasi, serta mengakses daftar materi dan jadwal salat. Selain itu, pengguna juga dapat melihat dan mengedit profil, menautkan akun, meninjau rekapitulasi, serta memantau progres pembelajaran. Setiap fitur pada diagram saling terhubung melalui relasi *include* dan *extend*, yang menandakan adanya alur proses yang saling melengkapi dalam aplikasi mobile.



Gambar 4. Usecase diagram mobile apps

2) Web apps

Sementara itu, **Gambar 5** menampilkan *Use Case Diagram Web Apps* yang melibatkan beberapa aktor, yaitu admin, guru, pengurus, dan pimpinan. Diagram ini menggambarkan berbagai fungsi utama seperti manajemen data (admin, guru, santri, kelas, materi, jadwal, semester, berita, dan notifikasi), pembaruan absensi kelas, serta pengelolaan penilaian. Terdapat pula fitur *melihat grafik dan statistik* yang memungkinkan pengguna tertentu memantau data hasil belajar dan aktivitas santri secara real time. Struktur diagram memperlihatkan adanya pembagian peran yang jelas antaraktor, di mana setiap aktor memiliki hak akses dan tanggung jawab yang berbeda sesuai dengan fungsinya. Kedua *use case diagram* tersebut menjadi dasar penting dalam pengembangan sistem karena memberikan gambaran menyeluruh mengenai kebutuhan pengguna dan alur interaksi yang terjadi, sehingga proses implementasi fitur dapat dilakukan secara lebih efisien dan selaras dengan tujuan sistem.



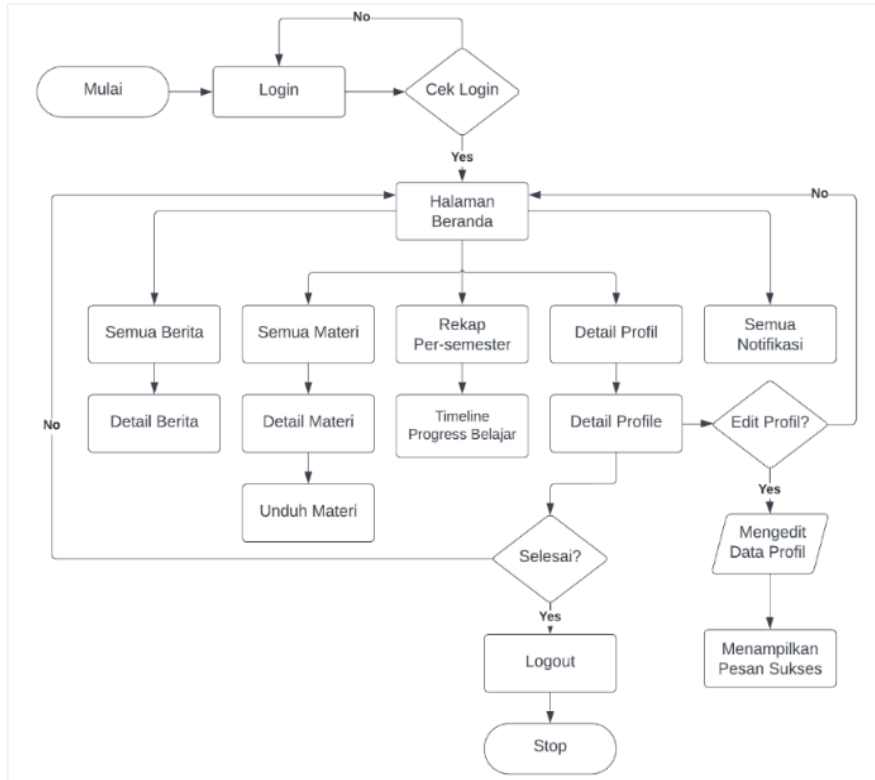
Gambar 5. Usecase diagram web apps

b. Diagram alir (flowchart)

Memiliki fungsi penggambaran terhadap suatu proses dari sebuah sistem yang sedang dikembangkan. Diagram alir dari sistem ini digambarkan dalam 5 jenis berdasarkan aktor yang ada pada sistem.

1) Diagram alir santri

Alur proses pada aplikasi mobile pada **Gambar 6** dimulai ketika pengguna melakukan *login* untuk mengakses sistem. Setelah proses *login* berhasil diverifikasi, pengguna diarahkan ke halaman beranda yang menampilkan beberapa menu utama, yaitu Semua Berita, Semua Materi, Rekap Per Semester, Detail Profil, dan Semua Notifikasi. Melalui menu-menu tersebut, pengguna dapat melihat detail berita, mengakses materi pembelajaran, meninjau progres belajar per semester, serta memperbarui informasi profil. Selain itu, pengguna juga dapat mengunduh materi pembelajaran dan melakukan pengeditan data profil, di mana setelah proses pengeditan selesai sistem akan menampilkan pesan sukses sebagai konfirmasi. Ketika seluruh aktivitas telah dilakukan, pengguna dapat keluar dari sistem melalui proses *logout* sebelum aplikasi dihentikan. Alur ini menggambarkan secara menyeluruh interaksi pengguna dengan sistem dalam menjalankan fungsi-fungsi utama aplikasi mobile.

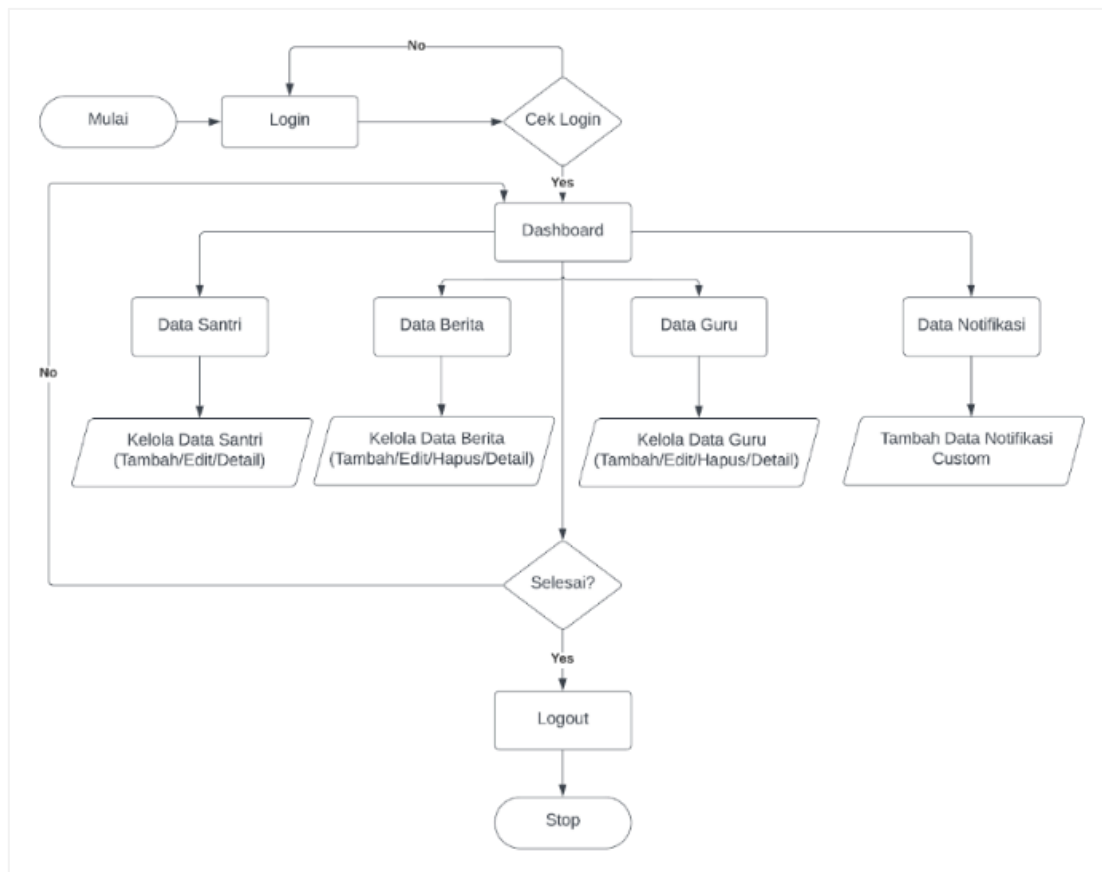


Gambar 6. Diagram alir santri

2) Diagram alir pengurus

Pada Gambar 7 ditunjukkan alur proses utama pada aplikasi web yang diawali ketika pengguna melakukan *login* untuk mendapatkan akses ke dalam sistem. Tahap ini menjadi langkah awal autentikasi guna memastikan bahwa setiap pengguna memiliki hak akses sesuai perannya. Setelah proses verifikasi berhasil, pengguna diarahkan ke halaman *dashboard* yang berfungsi sebagai pusat kendali untuk seluruh aktivitas pengelolaan data dalam sistem. Melalui *dashboard*, pengguna dapat memilih beberapa menu utama, antara lain Data Santri, Data Berita, Data Guru, dan Data Notifikasi. Masing-masing menu tersebut dirancang agar pengguna dapat melakukan berbagai tindakan seperti menambah data baru, mengedit informasi yang sudah ada, menghapus data yang tidak relevan, serta melihat detail informasi dengan mudah. Desain alur ini memprioritaskan kemudahan penggunaan serta efisiensi waktu agar pengguna, khususnya pengurus dan admin, dapat mengelola berbagai data secara efektif dalam satu halaman terpusat tanpa perlu berpindah ke banyak tampilan berbeda.

Sementara itu, pada menu Data Notifikasi, pengguna memiliki kemampuan untuk menambahkan notifikasi khusus (*custom notification*) yang akan ditampilkan kepada penerima tertentu sesuai dengan kebutuhan sistem dan konteks informasi yang ingin disampaikan. Proses ini membantu penyebaran informasi internal agar lebih cepat, tepat sasaran, dan terkoordinasi dengan baik. Setelah seluruh aktivitas pengelolaan data selesai dilakukan, pengguna dapat keluar dari sistem melalui proses *logout*, di mana sistem akan menutup sesi dan memastikan keamanan data dari akses yang tidak sah. Alur ini kemudian berakhir pada tahap *stop* sebagai penanda bahwa seluruh proses telah diselesaikan dengan benar. Diagram ini menggambarkan bahwa sistem web dirancang dengan struktur alur kerja yang sederhana namun efisien, mengutamakan kecepatan, akurasi, serta integrasi antarfungsi untuk mendukung pengelolaan data secara terstruktur dan berkelanjutan di lingkungan lembaga pendidikan.



Gambar 7. Diagram alir pengurus

c. ERD

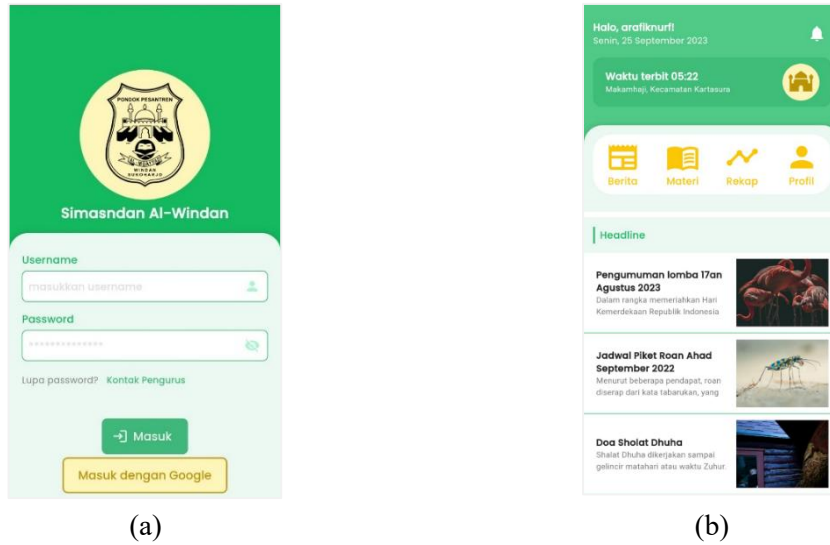
Perancangan *Entity Relationship Diagram* (ERD) pada Simasndan memiliki 13 tabel atau entitas. Seperti Admin, Santri, Guru, Berita, Jadwal, Penilaian, Notifikasi, dan lain sebagainya. Setiap entitas memiliki hubungan atau relasi dengan entitas lain. Terdapat relasi one to many dan many to many.

3.3 Hasil Aplikasi

Hasil akhir dari *sprint* adalah terciptanya produk Sistem Informasi Manajemen Santri Al-Windan (Simasndan) yang siap diserahkan ke klien (pihak Pondok Pesantren Al-Muayyad Windan Sukoharjo)

a. Mobile apps

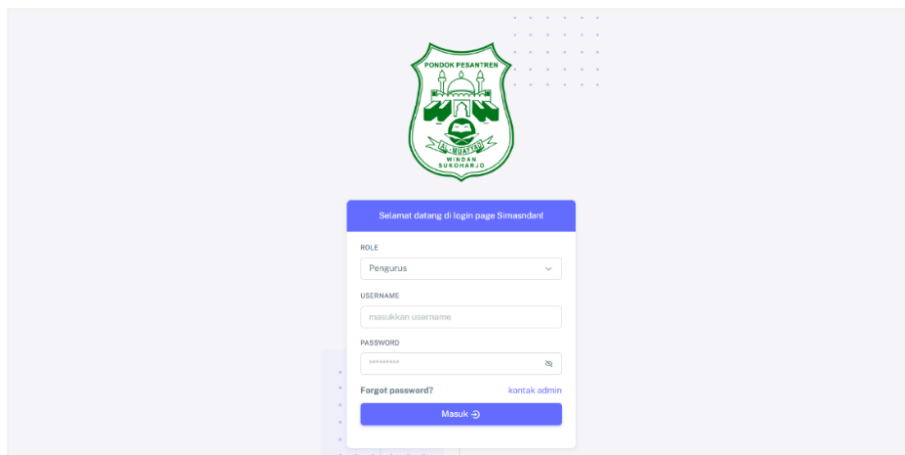
Halaman antarmuka aplikasi mobile terdiri atas dua tampilan utama, yaitu halaman *login* dan halaman beranda. Gambar 8 tampilan *login* (a) menampilkan identitas aplikasi “Simasndan Al-Windan” disertai kolom input untuk *username* dan *password*, tombol masuk, serta opsi masuk menggunakan akun Google. Desain halaman ini sederhana dengan dominasi warna hijau dan kuning lembut, mencerminkan identitas lembaga dan memudahkan pengguna dalam proses autentikasi. Sementara itu, tampilan beranda (b) berfungsi sebagai halaman utama setelah pengguna berhasil masuk ke sistem. Halaman ini menampilkan sapaan pengguna, informasi waktu salat, serta menu navigasi utama yang terdiri atas Berita, Materi, Rekap, dan Profil. Di bagian bawah, tersedia daftar *headline* berita atau pengumuman penting yang ditampilkan secara dinamis. Struktur tampilan ini dirancang agar pengguna dapat mengakses fitur-fitur utama dengan cepat, intuitif, dan nyaman melalui perangkat mobile.



Gambar 8. Halaman login (a) dan beranda (b) mobile apps

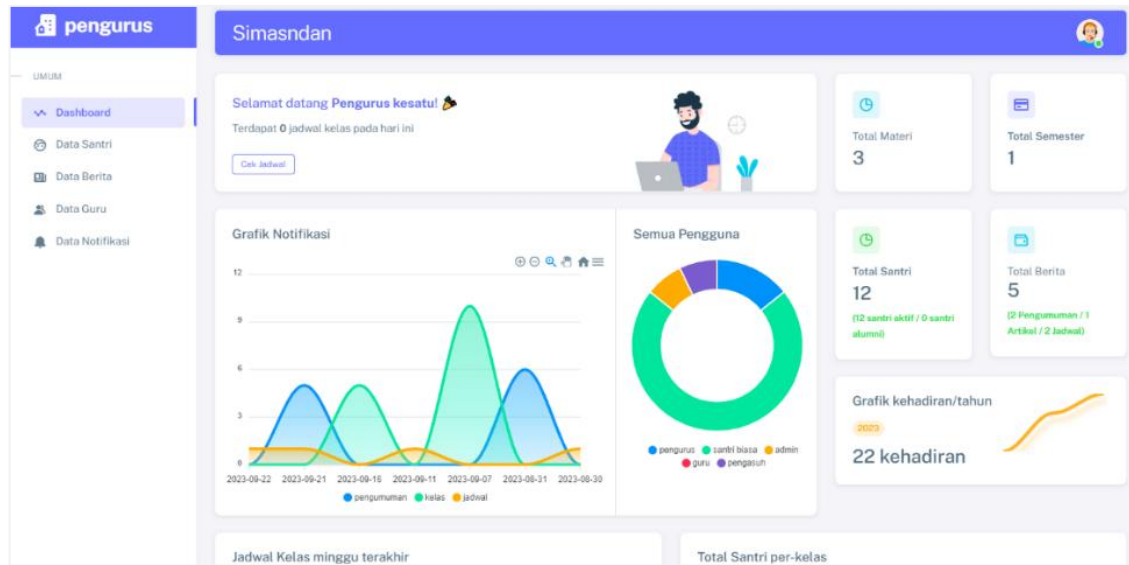
b. Web apps

Tampilan antarmuka web apps terdiri atas dua bagian utama, yaitu halaman *login* dan halaman *dashboard*. Tampilan *login* pada Gambar 9 menampilkan identitas lembaga di bagian atas serta kolom input untuk memilih peran (*role*), mengisi *username*, dan *password*. Tersedia pula opsi pemulihan kata sandi (*forgot password*) dan tautan untuk menghubungi admin apabila pengguna mengalami kendala akses. Desain halaman ini sederhana namun informatif, dengan dominasi warna putih dan biru yang memberikan kesan profesional serta mudah dipahami oleh pengguna dari berbagai peran, seperti admin, guru, maupun pengurus.



Gambar 9. Halaman login web apps

Sementara itu, tampilan *dashboard* pada Gambar 10 berfungsi sebagai pusat kendali bagi pengguna setelah berhasil masuk ke sistem. Halaman ini menampilkan berbagai informasi penting seperti jumlah total materi, semester aktif, total santri, berita, serta grafik notifikasi dan kehadiran. Selain itu, terdapat visualisasi data berupa diagram dan grafik yang memudahkan pengguna dalam memantau aktivitas dan perkembangan santri secara real time. Menu navigasi di sisi kiri menyediakan akses cepat ke fitur-fitur utama, seperti manajemen santri, berita, guru, dan notifikasi. Struktur antarmuka ini dirancang agar pengguna dapat mengelola data secara efisien, dengan tampilan yang responsif dan mudah digunakan pada berbagai perangkat.



Gambar 10. Halaman dashboard web apps

3.4 Hasil Pengujian

Pengujian dilakukan untuk memastikan bahwa setiap fungsi pada aplikasi berjalan sesuai dengan kebutuhan pengguna dan rancangan sistem. Metode yang digunakan adalah **black box testing**, yang berfokus pada pengujian fungsionalitas tanpa melihat kode program secara langsung. Proses pengujian dilakukan pada setiap fitur utama aplikasi untuk memastikan semua alur dapat dijalankan dengan benar oleh pengguna.

a. Black Box Testing

Hasil pengujian pada Tabel 3 menunjukkan bahwa seluruh fungsi utama pada aplikasi mobile berjalan dengan baik sesuai dengan hasil yang diharapkan. Semua kasus uji memperoleh status **PASS**, yang menandakan tidak ditemukan kesalahan dalam proses eksekusi. Dengan demikian, dapat disimpulkan bahwa aplikasi telah memenuhi aspek fungsionalitas dan siap digunakan dalam tahap implementasi atau uji coba pengguna yang lebih luas.

Tabel 3. Pengujian *Black box testing* (Mobile Apps)

Mobile Apps			
Case	Hasil yang diharapkan	Status	Tanggal
User login ke aplikasi	Berhasil masuk ke aplikasi dan redirect ke beranda	PASS	02/10/2023
User melihat headline, jadwal sholat, & menu	Beranda menampilkan headline berita, jadwal sholat, & menu	PASS	02/10/2023
User mengakses menu data berita	Data berita menampilkan daftar semua, filter, & detail berita	PASS	02/10/2023
User mengakses menu data materi	Data materi menampilkan daftar semua, detail, & unduh materi	PASS	02/10/2023
User mengakses menu data rekap	Data rekap menampilkan daftar per-semester, & timeline rekap	PASS	02/10/2023
User mengakses menu profil	Profil menampilkan detail, tautkan akun, & update profil	PASS	02/10/2023
User mengakses fitur Notifikasi	Tampil semua notifikasi, & menampilkan pop up notifikasi	PASS	02/10/2023

Mobile Apps			
Case	Hasil yang diharapkan	Status	Tanggal
User logout dari aplikasi	User dialihkan ke halaman login	PASS	02/10/2023

Selanjutnya, pengujian juga dilakukan pada aplikasi berbasis web untuk memastikan seluruh fitur berfungsi sesuai dengan kebutuhan pengguna. Pengujian ini menggunakan metode black box testing, dengan fokus pada pemeriksaan fungsionalitas tanpa melihat struktur kode program. Setiap kasus uji dilakukan berdasarkan peran pengguna (*role*) seperti admin, guru, dan pengurus, guna memastikan sistem dapat menyesuaikan hak akses dengan benar. Hasil pengujian pada aplikasi web dapat dilihat pada Tabel 4 berikut.

Tabel 4. Pengujian *Black box testing* (Web Apps)

Web Apps			
Case	Hasil yang diharapkan	Status	Tanggal
User login ke web berdasarkan <i>role</i>	Berhasil masuk dan dialihkan ke dashboard berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data santri	Dapat memanajemen data santri berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data admin	Dapat memanajemen data admin berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data berita	Dapat mengelola data berita berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data guru	Dapat mengelola data guru berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data semester	Dapat mengelola data semester berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data notifikasi	Dapat mengelola data notifikasi berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data materi	Dapat mengelola data materi berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data jadwal	Dapat mengelola data jadwal berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data kelas	Dapat mengelola data kelas berdasarkan <i>role</i>	PASS	02/10/2023
User dapat mengelola data penilaian	Dapat mengelola data penilaian berdasarkan <i>role</i>	PASS	02/10/2023
User logout dari web	Berhasil logout dari web dan dialihkan ke halaman login	PASS	02/10/2023

b. *System Usability Scale* (SUS)

Selain pengujian fungsional, dilakukan juga pengujian tingkat kegunaan (*usability testing*) menggunakan metode *System Usability Scale* (SUS) untuk menilai sejauh mana aplikasi mudah digunakan oleh pengguna. Pengujian ini melibatkan sepuluh responden yang diminta mengisi kuesioner berisi sepuluh pertanyaan terkait pengalaman penggunaan aplikasi. Setiap

pertanyaan dinilai menggunakan skala Likert 1–5, kemudian hasilnya dikonversi menjadi skor akhir untuk menentukan tingkat kelayakan aplikasi berdasarkan persepsi pengguna. Nilai hasil pengujian SUS setelah proses konversi disajikan pada [Tabel 5](#) berikut.

Tabel 5. Nilai Hasil Kuesioner Pengujian SUS Setelah Dikonversi

RESPONDEN	Pertanyaan									
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
R1	3	3	4	3	4	3	3	3	1	1
R2	3	3	2	2	3	3	3	3	3	2
R3	3	3	3	2	3	3	3	2	3	1
R4	4	3	4	1	3	3	3	3	3	1
R5	4	4	4	4	4	4	4	4	4	1
R6	3	3	3	3	3	3	3	3	3	1
R7	4	3	3	3	3	4	3	3	4	1
R8	4	3	3	2	3	3	3	3	3	1
R9	4	3	4	2	4	3	4	4	3	1
R10	3	3	3	2	3	3	3	3	3	1
Total Skor	35	31	33	24	33	32	32	31	30	11
Total Akhir Skor	292									

$$Usability = \frac{\sum Total Skor}{n responden} \times 2,5 \quad (1)$$

$$Usability = \frac{292}{10} \times 2,5 = 73$$

Hasil dari pengolahan data kuesioner pengujian SUS pada 10 responden terhadap 10 pertanyaan menggunakan Persamaan 1, menunjukkan bahwa Simasndan memiliki nilai rata-rata *usability* 73. Berdasarkan penilaian SUS Score pada [Gambar 2](#), nilai *usability* dilihat dari segi *Acceptability Range* masuk ke kategori *Acceptable*, sedangkan pada *Grade Scale* berada di posisi *Grade C* dan pada *Adjective Rating*, berada di posisi *Good*.

KESIMPULAN

Penerapan model *Scrum* dalam pengembangan sistem informasi manajemen santri terbukti efektif untuk pengelolaan proyek secara iteratif dan adaptif terhadap perubahan kebutuhan. Keberhasilan penerapan terlihat dari hasil *sprint retrospective* melalui *burndown chart* yang menunjukkan seluruh *sprint* terselesaikan sesuai rencana. Sistem yang dikembangkan juga telah diuji kelayakan dan berfungsi sesuai kebutuhan pondok pesantren. Hasil pengujian SUS pada 10 responden dengan 10 pertanyaan menghasilkan nilai rata-rata *usability* sebesar 73. Berdasarkan penilaian SUS Score, nilai tersebut termasuk kategori *Acceptable* pada *Acceptability Range*, berada pada *Grade C* pada *Grade Scale*, dan tergolong *Good* pada *Adjective Rating*. Dengan demikian, sistem ini dinyatakan layak, mudah digunakan, dan mampu mendukung digitalisasi pengelolaan data santri di pondok pesantren.

REFERENSI

- [1] H. E. Journal, “Peran Pondok Pesantren Darul Mukhlisin Terhadap Peningkatan Nilai Religius Masyarakat Desa Krincing Secang Magelang,” vol. 3, no. 2, pp. 20–24, 2021.
- [2] F. Adetya, “The ABC Laundry Service Information System Based on Web using SDLC Method,” *J. Media Inf. Teknol.*, vol. 1, no. 2, pp. 53–62, 2024, <https://doi.org/10.69616/mit.v1i2.187>.
- [3] I. Asriani, M. Muchtar, R. R. Ismail, A. Paliling, K. Sya’ban, and R. Karim, “Expert System for Determining Diseases and Pests in Seaweed Using Forward Chaining (Case Study: Watorumbe Village, Mawasangka Tengah),” *Media Comput. Sci.*, vol. 1, no. 1, pp. 47–60, 2024, <https://doi.org/10.69616/mcs.v1i1.175>.
- [4] A. D. Kalifia, L. E. Astrianty, F. I. Sanjaya, and A. Pramudwiatmoko, “Empowering Digital Parenting through Web-Based Admission System and Technology Literacy Training for Families in Early Childhood Education,” *MEKONGGA J. Pengabd. Masy.*, vol. 2, no. 1, pp. 30–38, 2025, <https://doi.org/10.69616/mekongga.v2i1.218>.
- [5] R. M. Irsyad and M. I. P. Nasution, “The Impact of Using Management Information Systems on Customer Satisfaction Levels,” *J. Media Inf. Teknol.*, vol. 2, no. 1, pp. 27–32, 2025, <https://doi.org/10.69616/mit.v2i1.209>.
- [6] A. H. Nasrullah, A. M. Fajar, M. A. Taufiq, N. Rahmat, and F. Adiba, “Evaluation Of Fuzzy C-Means Method For District Clustering,” *Media Comput. Sci.*, vol. 1, no. 2, pp. 117–128, 2024, <https://doi.org/10.69616/mcs.v1i2.203>.
- [7] J. Y. Sari, Y. P. Pasrun, M. Muchtar, R. Karim, and R. A. Saputra, “Development of Multimedia-Based Online Learning Media for Class VI Students of SDN 105 Kendari,” *MEKONGGA J. Pengabd. Masy.*, vol. 1, no. 1, pp. 1–6, 2024, <https://doi.org/10.69616/mekongga.v1i1.170>.
- [8] R. Karim, “Enhancing Students’ Learning Interest Through the Implementation of the Kampus Mengajar Batch 8 Program: Classroom Arrangement,” *MEKONGGA J. Pengabd. Masy.*, vol. 1, no. 2, pp. 61–68, 2024, <https://doi.org/10.69616/mekongga.v1i2.197>.
- [9] K. Nurdianto, M. A. Heses, H. Yulianto, and G. Aditya, “Jurnal Teknik Elektro dan Informatika Sistem Informasi Pengolahan Data Akademik di Pondok Pesantren BUSTANU ADAIL QUR ’ AN Demak,” vol. 3, no. 1, pp. 1–17, 2021, <https://doi.org/10.55542/jurtie.v3i1.322>.
- [10] A. B. Setiawan and J. Sulaksono, “Sistem Informasi Manajemen Santri Di Pondok Pesantren Al Ishlah Kota Kediri,” *Netw. Eng. Res. Oper.*, vol. 4, no. 2, pp. 81–86, 2019, <https://doi.org/10.21107/nero.v4i2.122>.
- [11] J. A. Ramadhan, D. T. Haniva, and A. Suharso, “Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid,” *JIEET J. Inf. Eng. Educ. Technol.*, vol. 07, no. 01, pp. 36–42, 2023, <https://doi.org/10.26740/jieet.v7n1.p36-42>.
- [12] M. Utami, E. Dwika Putra, and Y. Apridiansyah, “Survey Paper: Perbandingan Agile Process Development Method (SCRUM dan RUP),” *JSAI (Journal Sci. Appl. Informatics)*, vol. 5, no. 2, pp. 113–116, 2022, <https://doi.org/10.36085/jsai.v5i2.3499>.

- [13] S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, “Agile Software Development: Methodologies and Trends,” *Artic. Int. J. Interact. Mob. Technol.*, pp. 246–270, 2020, <https://doi.org/10.3991/ijim.v14i11.13269>.
- [14] S. Bantun and J. Y. Sari, “Pengembangan Aplikasi Pencarian Rumah Kos Dengan Metode Scrum Dalam Rangka Digitalisasi UMKM Di Desa Popalia,” *INFORMAL Informatics J.*, vol. 8, no. 1, pp. 12–21, 2023, <https://doi.org/10.19184/isj.v8i1.34507>.
- [15] Z. Elma, “Implementasi Metode Usability Testing Dengan System Usability Scale Dalam Evaluasi Website Layanan Penyedia Subtitle (Studi Kasus: Subscene),” *Ultim. InfoSys J. Ilmu Sist. Inf.*, vol. 10, no. 2, pp. 104–110, 2020, <https://doi.org/10.31937/si.v10i2.1197>.