

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas hasil dan pembahasan mengenai pengembangan prototipe sistem akuaponik serta rincian untuk setiap tahapnya berdasarkan metode yang telah dijabarkan pada bab metodologi penelitian.

4.1 Hasil Wawancara

Sebelum proses pengembangan sistem dilakukan, peneliti melaksanakan wawancara dengan seorang praktisi akuaponik untuk memperoleh pemahaman mengenai proses dan teknis budidaya pakcoy menggunakan sistem akuaponik. Wawancara ini bertujuan untuk mengidentifikasi parameter penting yang perlu diperhatikan selama budidaya serta menentukan kebutuhan sistem yang dapat mendukung kegiatan monitoring dan pengendalian.

Berdasarkan hasil wawancara, diperoleh informasi bahwa beberapa parameter penting dalam budidaya akuaponik meliputi kelembaban media tanam, suhu, pH air, nutrisi, dan aliran air. Narasumber menjelaskan bahwa kelembaban menjadi salah satu faktor yang perlu diperhatikan karena kondisi yang terlalu lembab dapat menyebabkan busuk akar. Selain itu, rentang pH yang direkomendasikan berada pada nilai 5,5–7,5 dan suhu optimal untuk pertumbuhan tanaman berada pada kisaran 25°C–30°C.

Narasumber juga menyampaikan bahwa sistem monitoring yang ideal sebaiknya mampu menampilkan data secara *real-time*, menyediakan *data logging*, memberikan notifikasi ketika terjadi kondisi yang tidak normal, serta mendukung otomatisasi pada beberapa proses budidaya. Informasi yang diperoleh dari wawancara ini kemudian digunakan sebagai referensi dalam menentukan sensor yang digunakan, fitur yang dikembangkan, serta batasan masalah pada penelitian. Informasi lengkap mengenai hasil wawancara praktisi ditampilkan pada lampiran 1.

4.2 Dokumentasi dan Kebutuhan Sistem

Subbab ini menjelaskan mengenai perencanaan, dokumentasi, implementasi dan pengujian dari pengembangan sistem *website* Smartaquaponic. Semua yang ditampilkan pada subbab ini disusun berdasarkan hasil wawancara dengan praktisi dan planning bersama dengan *product owner* selama keberjalanan Scrum *process* sehingga yang ditampilkan pada subbab ini adalah hasil akhir dari perubahan yang pernah terjadi.

4.2.1 User Story

Berdasarkan hasil wawancara dan *planning* selama Scrum, *Website* Smartaquaponic berbasis *web* memiliki dua aktor utama, yaitu *admin* dan *user*. Kebutuhan yang telah teridentifikasi kemudian diterjemahkan ke dalam bentuk *User Story* untuk menggambarkan tujuan yang ingin dicapai pengguna saat berinteraksi dengan sistem.

User Story admin difokuskan pada fungsi pengelolaan sistem, meliputi pengelolaan data *user*, *site*, *device*, sensor, dan aktuator, serta pemantauan kondisi keseluruhan sistem melalui *dashboard admin*. *Admin* juga memiliki kewenangan untuk melakukan pengelolaan hubungan antara *site* dan *device* yang terdaftar pada sistem. Sementara itu, *User Story user* difokuskan pada monitoring sistem aquaponik milik *user* masing-masing, meliputi pemantauan data sensor dan log aktuator, pengelolaan jadwal feeder dan *grow light*, pengendalian aktuator secara manual, pengelolaan hubungan *device* dengan *site*, serta pengunduhan data *monitoring* dalam format CSV. Selain itu, kedua aktor dapat melakukan autentikasi ke dalam sistem, mengelola akun masing-masing, dan menerima notifikasi yang dikirimkan oleh sistem. User Story yang digunakan dalam penelitian ini disajikan pada Tabel 4.1.

Tabel 4.1 Tabel *User Story*

No	ID US	User Story
1	US-SMAQ-1	Sebagai <i>Admin</i> atau <i>User</i> , saya ingin melakukan <i>login</i> dan <i>logout</i> sehingga saya dapat mengakses sistem dengan aman sesuai hak akses saya.
2	US-SMAQ-2	Sebagai <i>Admin</i> atau <i>User</i> , saya ingin mengelola informasi akun saya sehingga data profil, kredensial, dan pengaturan notifikasi email dan whatsapp saya selalu sesuai dengan keinginan saya.

No	ID US	User Story
3	US-SMAQ-3	Sebagai <i>Admin</i> atau <i>User</i> , saya ingin menerima notifikasi dari sistem sehingga saya dapat mengetahui kondisi abnormal yang terjadi pada sistem.
4	US-SMAQ-4	Sebagai <i>Admin</i> , saya ingin melihat <i>dashboard</i> yang menampilkan informasi keseluruhan sistem sehingga saya dapat melakukan pemantauan dan pengelolaan sistem secara menyeluruh.
5	US-SMAQ-5	Sebagai <i>User</i> , saya ingin melihat <i>dashboard</i> yang menampilkan informasi site milik saya sehingga saya dapat memantau kondisi site yang saya kelola.
6	US-SMAQ-6	Sebagai <i>Admin</i> , saya ingin mengelola data <i>user</i> sehingga saya dapat menambah, melihat, mengubah, dan menghapus akun pengguna.
7	US-SMAQ-7	Sebagai <i>Admin</i> , saya ingin mengelola data device sehingga data perangkat yang tersedia dalam sistem tetap akurat dan terorganisir.
8	US-SMAQ-8	Sebagai <i>Admin</i> , saya ingin menghubungkan atau melepaskan device dengan site sehingga perangkat dapat ditempatkan pada site yang sesuai.
9	US-SMAQ-9	Sebagai <i>User</i> , saya ingin menghubungkan atau melepaskan device pada site milik saya sehingga saya dapat mengatur perangkat apa saja yang digunakan pada site milik saya.
10	US-SMAQ-10	Sebagai <i>Admin</i> , saya ingin mengelola data aktuator sehingga aktuator yang digunakan sistem selalu sesuai dengan kebutuhan yang diperlukan.
11	US-SMAQ-11	Sebagai <i>Admin</i> , saya ingin mengelola data sensor sehingga data sensor yang digunakan sistem selalu sesuai dengan kebutuhan yang diperlukan.
12	US-SMAQ-12	Sebagai <i>Admin</i> , saya ingin mengelola data site sehingga seluruh site yang terdaftar dapat dikelola oleh saya.
13	US-SMAQ-13	Sebagai <i>User</i> , saya ingin melihat riwayat log aktuator dan mengunduhnya dalam format CSV sehingga saya dapat melakukan dokumentasi dan analisis aktivitas aktuator.
14	US-SMAQ-14	Sebagai <i>User</i> , saya ingin melihat data sensor dan mengunduhnya dalam format CSV sehingga saya dapat melakukan analisis terhadap kondisi lingkungan pada site saya.
15	US-SMAQ-15	Sebagai <i>User</i> , saya ingin mengelola jadwal grow light sehingga pencahayaan untuk tanaman dapat berjalan secara otomatis sesuai kebutuhan.
16	US-SMAQ-16	Sebagai <i>User</i> , saya ingin mengelola jadwal feeder sehingga pemberian pakan untuk ikan dapat dilakukan secara otomatis sesuai jadwal yang ditentukan.
17	US-SMAQ-17	Sebagai <i>User</i> , saya ingin mengaktifkan dan menonaktifkan aktuator secara manual sehingga saya dapat melakukan kontrol langsung ketika diperlukan.

4.2.2 Software Requirement Specification

Berdasarkan *User Story* yang disebutkan pada subbab 4.2.1, disusunlah sebuah *system requirement specification* (SRS) yang menjelaskan kebutuhan sistem dengan lebih teknis. Kebutuhan fungsional dari *website smartaquaponic* didokumentasikan dalam dokumen SRS. Isi dari dokumen SRS bisa dilihat pada Tabel 4.2.

Tabel 4.2 SRS Fungsional *Website*

No	Kode SRS	Deskripsi	Role
1	SRS-F-SMAQ-01	<i>Admin dan User dapat melakukan login ke dalam website menggunakan email dan password yang valid dan melakukan logout.</i>	<i>Admin dan User</i>
2	SRS-F-SMAQ-02	<i>Admin dan user dapat melakukan operasi view, edit dan delete pada pengaturan akunnya masing-masing.</i>	<i>Admin dan User</i>
3	SRS-F-SMAQ-03	<i>Admin dan User dapat menerima notifikasi dari sistem</i>	<i>Admin dan User</i>
4	SRS-F-SMAQ-04	<i>Admin dapat membuka halaman dashboard admin yang memuat informasi keseluruhan sistem.</i>	<i>Admin</i>
5	SRS-F-SMAQ-05	<i>User dapat membuka halaman dashboard user yang memuat informasi umum mengenai site-site miliknya saja.</i>	<i>User</i>
6	SRS-F-SMAQ-06	<i>Admin dapat melakukan operasi create, view, edit, dan delete pada data user.</i>	<i>Admin</i>
7	SRS-F-SMAQ-07	<i>Admin dapat melakukan operasi create, view, edit, dan delete pada data devices.</i>	<i>Admin</i>
8	SRS-F-SMAQ-08	<i>Admin dapat melepaskan dan menghubungkan device dan site yang tersedia di database.</i>	<i>Admin</i>
9	SRS-F-SMAQ-09	<i>User dapat melepaskan dan menghubungkan site miliknya dengan device lain yang tersedia.</i>	<i>User</i>
10	SRS-F-SMAQ-10	<i>Admin dapat melakukan operasi create, view, edit, dan delete pada tabel aktuator</i>	<i>Admin</i>
11	SRS-F-SMAQ-11	<i>Admin dapat melakukan operasi create, view, edit, dan delete pada tabel sensor</i>	<i>Admin</i>
12	SRS-F-SMAQ-12	<i>Admin dapat melakukan operasi create, view, edit dan delete pada tabel site.</i>	<i>Admin</i>
13	SRS-F-SMAQ-13	<i>User dapat melakukan operasi view pada tabel log aktuator dan dapat mengunduh file .csv dari tabel log aktuator yang terhubung dengan site miliknya.</i>	<i>User</i>
14	SRS-F-SMAQ-14	<i>User dapat melakukan operasi view pada tabel data sensor dan dapat mengunduh file .csv dari</i>	<i>User</i>

No	Kode SRS	Deskripsi	Role
		tabel data sensor yang terhubung dengan <i>site</i> miliknya.	
15	SRS-F-SMAQ-15	<i>User</i> dapat melakukan operasi <i>create</i> , <i>view</i> , <i>edit</i> , dan <i>delete</i> pada pengaturan jadwal <i>grow light</i> .	<i>User</i>
16	SRS-F-SMAQ-16	<i>User</i> dapat melakukan operasi <i>create</i> , <i>view</i> , <i>edit</i> , dan <i>delete</i> pada pengaturan jadwal <i>feeder</i> .	<i>User</i>
17	SRS-F-SMAQ-17	<i>User</i> dapat mengaktifkan (<i>turn on</i>) dan menonaktifkan (<i>turn off</i>) aktuator bertipe selain <i>grow light</i> dan <i>feeder</i> .	<i>User</i>

Sistem ini juga memiliki kebutuhan-kebutuhan non-fungsional untuk mendukung kebutuhan sistem agar dapat berjalan dengan baik. Kebutuhan non-fungsional yang tercantum mencakup aspek akses, responsifitas antarmuka pada perangkat yang beragam dan penyimpanan data. Kebutuhan non-fungsional sistem dapat dilihat pada Tabel 4.2.

Tabel 4.3 SRS Non-fungsional *Website*

No	Kode SRS	Deskripsi
1	SRS-N-SMAQ-01	<i>Website</i> dapat diakses pada semua <i>Web Browser</i>
2	SRS-N-SMAQ-02	Antarmuka <i>website</i> dibangun dengan desain yang responsif (<i>responsive design</i>) sehingga tata letaknya dapat menyesuaikan secara otomatis dengan berbagai ukuran layar perangkat yang digunakan, mulai dari <i>smartphone</i> , <i>tablet</i> , hingga <i>laptop</i>
3	SRS-N-SMAQ-03	Sistem mampu memproses, menerima, dan merender data dari broker MQTT ke antarmuka dashboard pemantauan secara real-time dengan waktu tunda (<i>latency</i>) jaringan yang minimal.

4.2.3 Use Case

Use case diagram digunakan untuk memodelkan perilaku dan fungsionalitas sistem aplikasi *website* monitoring akuaponik pintar berdasarkan perspektif interaksi pengguna eksternal. Berdasarkan dokumen SRS yang disebutkan pada Tabel 4.2 dan Tabel 4.3 pada bab sebelumnya, sistem ini memisahkan hak akses ke dalam dua aktor utama dengan hak akses dan perilaku yang berbeda, yaitu:

1. Admin

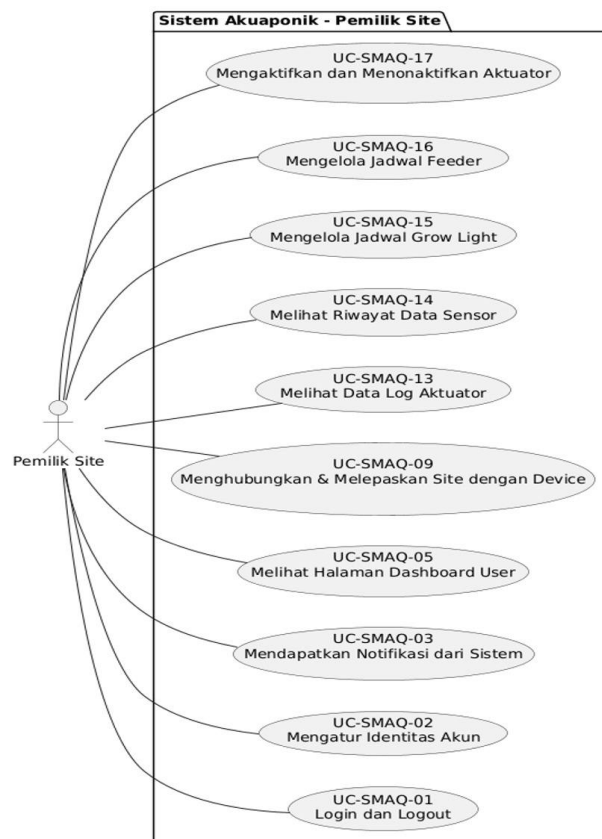
Admin adalah aktor yang memiliki otoritas untuk mengelola sistem, meliputi pengelolaan data pengguna (*User*), pendaftaran perangkat keras baru (*Device*),

manipulasi relasi pemetaan antara lokasi (*Site*) dan perangkat, serta manajemen tipe sensor dan aktuator.

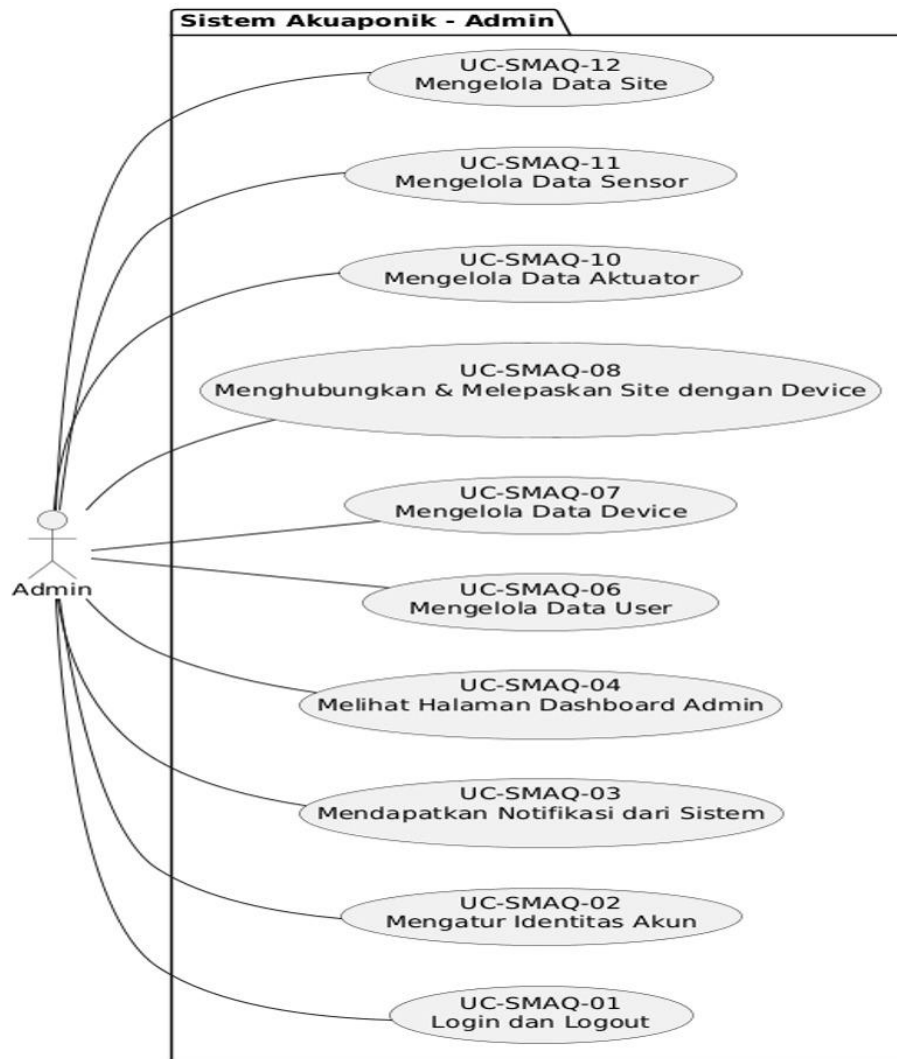
2. *Owner*

Aktor yang bertindak sebagai pemilik atau pengelola langsung dari satu atau beberapa lokasi akuaponik (*Site*). Hak akses aktor ini dibatasi pada pemantauan visualisasi data sensor secara *real-time*, melihat log riwayat, mengatur otomasi jadwal *grow light* dan feeder, serta melakukan kontrol manual terhadap akuaponik miliknya sendiri. Dalam keseluruhan laporan pengembangan, *owner* disebut sebagai aktor *user*.

Interaksi kedua aktor dipindahkan menjadi dua buah diagram *use case* untuk menjaga keterbacaan rancangan sebuah sistem. Diagram untuk aktor *admin* merepresentasikan fungsionalitas *admin* dalam mengelola keseluruhan sistem. Diagram untuk aktor *user* merepresentasikan fungsionalitasnya sebagai pengelola akuaponik (*site*) miliknya sendiri. Dalam skripsi ini, diagram *use case* untuk *user* dan *admin* dapat dilihat pada Gambar 4.1 dan Gambar 4.2 berurutan. Penjelasan untuk masing-masing *use case* dijelaskan pada subbab 4.2.3 mengenai skenario *use case*.



Gambar 4.1 *Use Case Diagram User*



Gambar 4.2 Use Case Diagram Admin

4.2.4 Skenario Use Case

Skenario *use case* disusun untuk memperjelas aliran proses, prakondisi, serta kondisi akhir dari aktivitas yang dapat dilakukan oleh aktor. Skenario ini menjelaskan langkah-langkah interaksi antara sistem dengan *user*. Tabel skenario *use case* terdiri dari kode *use case*, nama *use case*, aktor primer, prakondisi, skenario utama, kondisi akhir, dan alur alternatif. Skenario *use case* dapat dilihat dari Tabel 4.4 sampai dengan Tabel 4.20.

1. Skenario Use Case Login dan Logout

Skenario ini membahas menjelaskan alur *login* dan *logout* dari *website smartaquaponic* yang dapat dilakukan oleh *admin* dan *user*. Skenario *use case* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Skenario *Use Case Login dan Logout*

Kode <i>Use Case</i>	UC-SMAQ-01
Nama <i>Use Case</i>	<i>Login dan Logout</i>
Aktor Primer	<i>Admin dan User</i>
Prakondisi	Aktor telah masuk ke halaman <i>login</i> melalui <i>url</i> . Aktor telah terdaftar ke dalam sistem.
Skenario Utama	a. <i>Login</i> Aktor memasukkan <i>email</i> dan <i>password</i> Aktor menekan tombol <i>log in</i> Aktor masuk ke dalam <i>website</i> b. <i>Logout</i> Aktor masuk ke dalam halaman <i>dashboard</i> Aktor menekan <i>log out button</i> pada bagian <i>sidebar</i>
Kondisi Akhir	Jika berhasil <i>login</i> , aktor akan masuk ke halaman <i>dashboard</i> . Jika berhasil <i>logout</i> , aktor akan kembali ke halaman <i>login</i> .
Alur Alternatif	Jika gagal, pesan <i>error</i> akan ditampilkan.

2. Skenario *Use Case* Mengatur Identitas Akun

Skenario ini menjelaskan mengenai alur *user* dan *admin* dalam mengatur identitas akun, termasuk pengaturan notifikasi email dan whatsapp. Skenario dapat dilihat pada Tabel 4.5.

Tabel 4.5 Skenario *Use Case* Mengatur Identitas Akun

Kode <i>Use Case</i>	UC-SMAQ-02
Nama <i>Use Case</i>	Mengatur Identitas Akun
Aktor Primer	<i>Admin dan User</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan menu “Akun” Aktor diarahkan ke halaman “Akun” b. <i>Edit</i> Aktor telah di halaman pengaturan akun Aktor mengubah data-data yang perlu diubah Aktor menekan tombol “simpan perubahan” Data berubah c. <i>Delete</i> Aktor menekan tombol “hapus akun” Aktor menekan tombol “oke” pada <i>alert</i> yang muncul Akun Aktor menjadi non-aktif
Kondisi Akhir	Jika berhasil, data ditampilkan sesuai dengan operasi yang dilakukan.
Alur Alternatif	-

3. Skenario *Use Case* Mendapatkan Notifikasi dari Sistem

Use Case ini menjelaskan langkah-langkah *user* untuk melihat notifikasi yang ditampilkan pada halaman *website* smartaquaponic. Skenario *use case* dapat dilihat pada Tabel 4.6.

Tabel 4.6. Skenario *Use Case* Mendapatkan Notifikasi dari Sistem

Kode Use Case	UC-SMAQ-03
Nama Use Case	Mendapatkan Notifikasi dari Sistem
Aktor Primer	<i>Admin</i> dan <i>User</i>
Prakondisi	Aktor telah berada di halaman <i>dashboard</i> .
Skenario Utama	Aktor menekan menu “Notifikasi” pada <i>sidebar</i> Aktor diarahkan ke halaman “Notifikasi”
Kondisi Akhir	Jika berhasil, diarahkan ke halaman notifikasi
Alur Alternatif	-

4. Skenario *Use Case* Melihat Halaman *Dashboard Admin*

Use Case ini menjelaskan alur *user admin* dalam membuka halaman *dashboard admin*. Skenario dapat dilihat pada Tabel 4.7.

Tabel 4.7 Skenario *Use Case* Melihat Halaman *Dashboard Admin*

Kode Use Case	UC-SMAQ-04
Nama Use Case	Melihat halaman <i>dashboard admin</i>
Aktor Primer	<i>Admin</i>
Prakondisi	Aktor berada di halaman <i>login</i>
Skenario Utama	Aktor mengisi email Aktor mengisi password yang valid Aktor menekan tombol “Masuk Sekarang” Aktor diarahkan ke halaman <i>dashboard admin</i>
Kondisi Akhir	Jika berhasil, halaman <i>dashboard admin</i> ditampilkan
Alur Alternatif	-

5. Skenario *Use Case* Melihat Halaman *Dashboard User*

Use Case ini menjelaskan alur *user* umum dalam membuka halaman *dashboard user* miliknya sendiri. Skenario dapat dilihat pada Tabel 4.8.

Tabel 4.8 Skenario *Use Case* Melihat Halaman *Dashboard User*

Kode Use Case	UC-SMAQ-05
Nama Use Case	Melihat halaman <i>dashboard User</i>
Aktor Primer	<i>User</i>
Prakondisi	Aktor berada di halaman <i>login</i>
Skenario Utama	Aktor mengisi email Aktor mengisi password yang valid

	Aktor menekan tombol “Masuk Sekarang” Aktor diarahkan ke halaman <i>dashboard user</i>
Kondisi Akhir	Jika berhasil, halaman <i>dashboard user</i> ditampilkan
Alur Alternatif	-

6. Skenario *Use Case* Mengelola Data *User*

Skenario *use case* ini menjelaskan mengenai bagaimana data-data *user* dikelola oleh *admin* dengan operasi *create, read, update, and delete* (CRUD). Skenario dapat dilihat pada Tabel 4.9.

Tabel 4.9 Skenario *Use Case* Mengelola Data *User*

Kode Use Case	UC-SMAQ-06
Nama Use Case	Mengelola data <i>user</i>
Aktor Primer	<i>Admin</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan menu “pengguna” pada <i>sidebar</i> Aktor diarahkan ke halaman pengguna Sistem menampilkan daftar pengguna b. <i>Create</i> Aktor menekan tombol “new user” Aktor mengisi <i>username</i> dan <i>password</i> untuk <i>user</i> Aktor menekan tombol “create” Sistem menambahkan data <i>user</i> ke dalam daftar data c. <i>Edit</i> Aktor menekan tombol “edit” pada data tertentu Aktor mengubah data yang diinginkan Aktor menekan tombol “Update” Data aktor yang diubah telah berubah d. <i>Delete</i> Aktor menekan tombol “hapus” pada data tertentu Aktor menekan button “confirm” Data aktor dihapus
Kondisi Akhir	Jika berhasil, daftar akan ditampilkan sesuai dengan operasi yang dilakukan
Alur Alternatif	Jika gagal, pesan <i>error</i> ditampilkan.

7. Skenario *Use Case* Mengelola Data *Device*

Skenario *use case* ini menjelaskan alur pengelolaan *data device* pada *website smartaquaponic* yang dapat dilakukan oleh *Admin*. Skenario *use case* dapat dilihat pada Tabel 4.10.

Tabel 4.10 Skenario *Use Case* Mengelola Data *Device*

Kode Use Case	UC-SMAQ-07
Nama Use Case	Mengelola data <i>device</i>
Aktor Primer	<i>Admin</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan menu “<i>Devices</i>” pada <i>sidebar</i> Aktor diarahkan ke halaman <i>devices</i> Sistem menampilkan daftar <i>devices</i> b. <i>Create</i> Aktor menekan tombol “Tambah <i>Device</i>” Aktor mengisi data-data <i>devices</i> Aktor menekan tombol “Simpan <i>Device</i>” Sistem menambahkan data <i>device</i> baru ke tabel c. <i>Edit</i> Aktor menekan tombol “Edit” pada data tertentu Aktor mengubah data yang diinginkan Aktor menekan tombol “Update” Data aktor yang diubah telah berubah d. <i>Delete</i> Aktor menekan tombol “<i>Delete</i>” pada data tertentu Aktor menekan button “<i>confirm</i>” Data aktor dihapus e. <i>Monitoring</i> (melihat data sensor dari <i>device</i>) Aktor menekan tombol “Monitoring” Aktor diarahkan ke halaman <i>monitoring</i>
Kondisi Akhir	Jika berhasil, data ditampilkan sesuai dengan aksi yang dilakukan.
Alur Alternatif	Jika gagal, pesan <i>error</i> ditampilkan.

8. Skenario *Use Case* Menghubungkan dan Melepaskan *Site* dengan *Device* oleh *Admin*
Skenario *use case* ini membahas alur untuk menghubungkan serta melepaskan hubungan antara *site* dengan *device* pada *website* smartaquaponic yang dapat dilakukan oleh *Admin*. Skenario *use case* dapat dilihat pada Tabel 4.11.

Tabel 4. 11 Skenario *Use Case* Menghubungkan dan Melepaskan *Site* dengan *Device* oleh *Admin*

Kode Use Case	UC-SMAQ-08
Nama Use Case	Menghubungkan dan melepaskan <i>site</i> dengan <i>device</i> oleh <i>admin</i>
Aktor Primer	<i>Admin</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. Menghubungkan Aktor menekan menu “<i>Devices</i>” pada <i>sidebar</i>

	Aktor diarahkan ke halaman <i>devices</i> Aktor memilih <i>site</i> dan <i>device</i> yang dihubungkan Aktor menekan tombol “Hubungkan” <i>Site</i> dan <i>device</i> saling terhubung. b. Melepaskan Aktor menekan tombol “Copot dari <i>Site</i>” Aktor menekan “Oke” pada <i>alert</i> yang muncul <i>Site</i> dan <i>device</i> saling terlepas
Kondisi Akhir	<i>Site</i> dan <i>Device</i> dapat saling terhubung atau terlepas sesuai dengan aksi yang dilakukan.
Alur Alternatif	Jika gagal, pesan <i>error</i> ditampilkan.

9. Skenario *Use Case* Menghubungkan dan Melepaskan *Site* dengan *Device* oleh *User*
Skenario *use case* ini menjelaskan alur untuk menghubungkan serta melepaskan hubungan antara *site* dengan *device* pada *website* smartaquaponic yang dapat dilakukan oleh *User*. Skenario *use case* dapat dilihat pada Tabel 4.12.

Tabel 4.12 Skenario *Use Case* Menghubungkan dan Melepaskan
Site dengan *Device* oleh *User*

Kode Use Case	UC-SMAQ-09
Nama Use Case	Menghubungkan dan melepaskan <i>site</i> dengan <i>device</i> oleh <i>user</i>
Aktor Primer	<i>User</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. Menghubungkan Aktor menekan menu “<i>Devices</i>” pada <i>sidebar</i> Aktor diarahkan ke halaman <i>devices</i> Aktor memilih <i>site</i> dan <i>device</i> yang dihubungkan Aktor menekan tombol “Hubungkan” <i>Site</i> dan <i>device</i> saling terhubung. b. Melepaskan Aktor menekan tombol “Copot <i>Device</i>” Aktor menekan “Oke” pada <i>alert</i> yang muncul <i>Site</i> dan <i>device</i> saling terlepas
Kondisi Akhir	<i>Site</i> dan <i>Device</i> dapat saling terhubung atau terlepas sesuai dengan aksi yang dilakukan.
Alur Alternatif	Jika gagal, pesan <i>error</i> ditampilkan.

10. Skenario *Use Case* Mengelola Data Aktuator
Skenario *use case* ini membahas alur pengelolaan data master aktuator pada *website* smartaquaponic yang dapat dilakukan oleh *Admin*. Skenario *use case* dapat dilihat pada Tabel 4.13.

Tabel 4.13 Skenario *Use Case* Mengelola Data Aktuator

Kode Use Case	UC-SMAQ-10
Nama Use Case	Mengelola data aktuator
Aktor Primer	<i>Admin</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan menu “Daftar Aktuator” pada <i>sidebar</i> Aktor diarahkan ke halaman daftar aktuator Sistem menampilkan daftar aktuator b. <i>Create</i> Aktor menekan tombol “Tambah Aktuator” Aktor mengisi data-data aktuator. Aktor menekan tombol “Simpan” Data Aktuator Tersimpan c. <i>Edit</i> Aktor menekan tombol “edit” pada data tertentu Aktor mengubah data yang diinginkan Aktor menekan tombol “Update” Data aktuator yang diubah telah berubah d. <i>Delete</i> Aktor menekan tombol “hapus” pada data tertentu Aktor menekan tombol “Oke” pada <i>alert</i> Data aktuator dihapus
Kondisi Akhir	Jika berhasil, daftar akan ditampilkan sesuai dengan operasi yang dilakukan
Alur Alternatif	Jika gagal, pesan <i>error</i> ditampilkan.

11. Skenario *Use Case* Mengelola Data Sensor

Skenario *use case* ini menjelaskan alur pengelolaan data master sensor pada *website* smartaquaponic yang dapat dilakukan oleh *Admin*. Skenario *use case* dapat dilihat pada Tabel 4.14.

Tabel 4.14 Skenario *Use Case* Mengelola Data Sensor

Kode Use Case	UC-SMAQ-11
Nama Use Case	Mengelola data sensor
Aktor Primer	<i>Admin</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan menu “Daftar Sensor” pada <i>sidebar</i> Aktor diarahkan ke halaman daftar sensor Sistem menampilkan daftar sensor b. <i>Create</i> Aktor menekan tombol “Tambah Sensor” Aktor mengisi data-data sensor

	Aktor menekan tombol “Simpan” Data sensor tersimpan c. <i>Edit</i> Aktor menekan tombol “edit” pada data tertentu Aktor mengubah data yang diinginkan Aktor menekan tombol “Update” Data sensor yang diubah telah berubah d. <i>Delete</i> Aktor menekan tombol “hapus” pada data tertentu Aktor menekan tombol “Oke” pada <i>alert</i> Data sensor dihapus
Kondisi Akhir	Jika berhasil, daftar akan ditampilkan sesuai dengan operasi yang dilakukan
Alur Alternatif	Jika gagal, pesan <i>error</i> ditampilkan.

12. Skenario *Use Case* Mengelola Data Site

Skenario *use case* ini membahas alur pengelolaan data site atau lokasi budidaya pada *website* smartaquaponic yang dapat dilakukan oleh *Admin*. Skenario *use case* dapat dilihat pada Tabel 4.15.

Tabel 4.15 Skenario *Use Case* Mengelola Data Site

Kode Use Case	UC-SMAQ-12
Nama Use Case	Mengelola data <i>site</i>
Aktor Primer	<i>Admin</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan menu “ <i>Sites</i> ” pada <i>sidebar</i> Aktor diarahkan ke halaman daftar <i>site</i> Sistem menampilkan daftar <i>site</i> b. <i>Create</i> Aktor menekan tombol “Tambah <i>Site</i> ” Aktor mengisi data-data <i>site</i> Aktor menekan tombol “Simpan” Data <i>site</i> tersimpan c. <i>Edit</i> Aktor menekan tombol “ <i>Edit Site</i> ” pada data tertentu Aktor mengubah data yang diinginkan Aktor menekan tombol “Update” Data <i>site</i> yang diubah telah berubah d. <i>Delete</i> Aktor menekan tombol “hapus” pada data tertentu Aktor menekan tombol “Oke” pada <i>alert</i> Data sensor dihapus
Kondisi Akhir	Jika berhasil, daftar akan ditampilkan sesuai dengan operasi yang dilakukan

Alur Alternatif	Jika gagal, pesan <i>error</i> ditampilkan.
-----------------	---

13. Skenario *Use Case* Melihat Data Log Aktuator

Skenario *use case* ini menjelaskan alur untuk melihat data log riwayat aktivitas aktuator pada *website* smartaquaponic yang dapat dilakukan oleh User. Skenario *use case* dapat dilihat pada Tabel 4.16.

Tabel 4.16 Skenario *Use Case* Melihat Data Log Aktuator

Kode Use Case	UC-SMAQ-13
Nama Use Case	Melihat data log aktuator
Aktor Primer	User
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan <i>dropdown</i> “Data Monitoring” pada <i>sidebar</i> Aktor menekan tombol “Log Aktuator” Aktor diarahkan ke halaman log aktuator b. Unduh File .csv Aktor menekan tombol “Download CSV” File CSV diunduh
Kondisi Akhir	Jika berhasil, data log aktuator ditampilkan dan file .csv dapat terunduh dengan benar
Alur Alternatif	-

14. Skenario *Use Case* Melihat Data Riwayat Data Sensor

Skenario *use case* ini membahas alur untuk melihat data riwayat pemantauan sensor secara berkala pada *website* smartaquaponic yang dapat dilakukan oleh User. Skenario *use case* dapat dilihat pada Tabel 4.17.

Tabel 4.17 Skenario *Use Case* Melihat Data Riwayat Data Sensor

Kode Use Case	UC-SMAQ-14
Nama Use Case	Melihat Riwayat Data Sensor
Aktor Primer	User
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan <i>dropdown</i> “Data Monitoring” pada <i>sidebar</i> Aktor menekan tombol “Riwayat Data Sensor” Aktor diarahkan ke halaman riwayat data sensor

	b. Unduh File .csv Aktor menekan tombol “Download CSV” File CSV diunduh
Kondisi Akhir	Jika berhasil, riwayat data sensor ditampilkan dan file data sensor .csv dapat terunduh dengan benar
Alur Alternatif	-

15. Skenario *Use Case* Mengelola Jadwal *Grow Light*

Skenario *use case* ini menjelaskan alur pengelolaan, penambahan, pengubahan, dan penghapusan jadwal otomatisasi grow light pada *website* smartaquaponic yang dapat dilakukan oleh User. Skenario *use case* dapat dilihat pada Tabel 4.18.

Tabel 4.18 Skenario *Use Case* Mengelola Jadwal *Grow Light*

Kode Use Case	UC-SMAQ-15
Nama Use Case	Mengelola Jadwal <i>Grow Light</i>
Aktor Primer	User
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan menu “Jadwal <i>Grow Light</i>” pada <i>sidebar</i> Aktor diarahkan ke halaman jadwal grow light b. <i>Create</i> Aktor menekan tombol “Tambah Jadwal” Aktor mengisi data-data jadwal Aktor menekan tombol “Simpan” Data jadwal <i>grow light</i> tersimpan c. <i>Edit</i> Aktor menekan tombol “edit” pada data tertentu Aktor mengubah data yang diinginkan Aktor menekan tombol “Update” Data jadwal <i>grow light</i> yang diubah telah berubah d. <i>Delete</i> Aktor menekan tombol “hapus” pada data tertentu Aktor menekan tombol “Oke” pada <i>alert</i> Data jadwal <i>grow light</i> dihapus
Kondisi Akhir	Jika berhasil, daftar akan ditampilkan sesuai dengan operasi yang dilakukan
Alur Alternatif	-

16. Skenario *Use Case* Mengelola Jadwal Feeder

Skenario *use case* ini membahas alur pengelolaan, penambahan, pengubahan, dan penghapusan jadwal pemberian pakan otomatis melalui feeder pada *website*

smartaquaponic yang dapat dilakukan oleh *User*. Skenario *use case* dapat dilihat pada Tabel 4.18.

Tabel 4.19 Skenario *Use Case* Mengelola Jadwal Feeder

Kode Use Case	UC-SMAQ-16
Nama Use Case	Mengelola Jadwal Feeder
Aktor Primer	<i>User</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. <i>View</i> Aktor menekan menu “Jadwal Pakan” pada <i>sidebar</i> Aktor diarahkan ke halaman jadwal Feeder b. <i>Create</i> Aktor menekan tombol “Tambah Jadwal” Aktor mengisi data-data jadwal Aktor menekan tombol “Simpan” Data jadwal Feeder tersimpan c. <i>Edit</i> Aktor menekan tombol “edit” pada data tertentu Aktor mengubah data yang diinginkan Aktor menekan tombol “Update” Data jadwal <i>grow light</i> yang diubah telah berubah d. <i>Delete</i> Aktor menekan tombol “hapus” pada data tertentu Aktor menekan tombol “Oke” pada <i>alert</i> Data jadwal <i>grow light</i> dihapus
Kondisi Akhir	Jika berhasil, daftar akan ditampilkan sesuai dengan operasi yang dilakukan
Alur Alternatif	-

17. Skenario *Use Case* Mengaktifkan dan Menonaktifkan Aktuator

Skenario *use case* ini menjelaskan alur kontrol manual untuk mengaktifkan dan menonaktifkan aktuator secara langsung (*real-time*) melalui *website* smartaquaponic yang dapat dilakukan oleh *User*. Skenario *use case* dapat dilihat pada Tabel 4.20.

Tabel 4.20 Skenario *Use Case* Mengaktifkan dan Menonaktifkan Aktuator

Kode Use Case	UC-SMAQ-17
Nama Use Case	Mengaktifkan dan Menonaktifkan Aktuator
Aktor Primer	<i>User</i>
Prakondisi	Aktor telah <i>login</i> ke dalam <i>website</i> dan berada di halaman <i>dashboard</i>
Skenario Utama	a. Mengaktifkan Aktuator Aktor menekan menu “Kontrol Perangkat” Aktor diarahkan ke halaman kontrol perangkat Aktor Menekan tombol “Nyalakan”

	b. Menonaktifkan Aktuator Aktuator menekan tombol “Matikan” pada aktuator
Kondisi Akhir	Jika berhasil, keadaan aktuator berubah sesuai dengan aksi yang dilakukan
Alur Alternatif	-

4.2.5 Sequence Diagram

Sequence diagram disusun berdasarkan *use case diagram*, *class diagram* dan skenario *use case* yang disusun. Alur interaksi antar objek seperti mengenai bagaimana *user* mengakses halaman tertentu atau melakukan aksi tertentu digambarkan dalam *sequence diagram*. *Sequence diagram* dalam penelitian ini ditampilkan sebagai berikut.

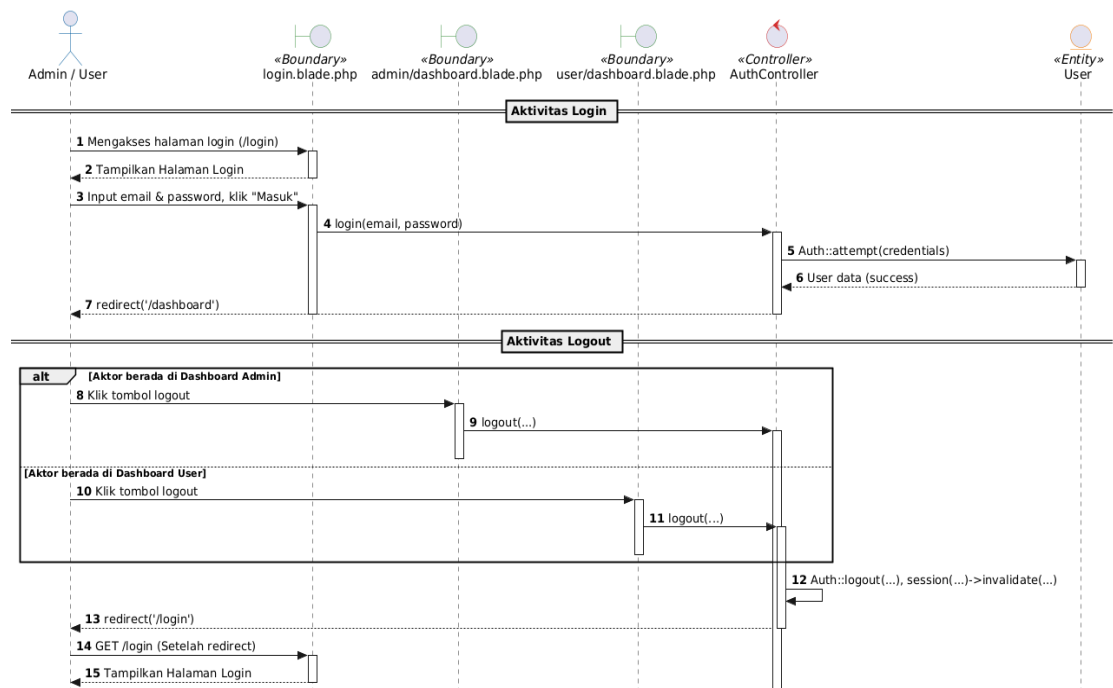
1. Login dan Logout

Kode Sequence : SD-SMAQ-01

Deskripsi : *Sequence diagram* untuk aktivitas *login* dan *logout* oleh *admin* dan *user*. *Diagram* digambarkan pada pada Gambar 4.3.

Objek :

- Boundary* : admin/dashboard.blade.php, user/dashboard.blade.php
- Controller* : AuthController
- Entity* : User



Gambar 4.3 *Sequence Diagram Login dan Logout*

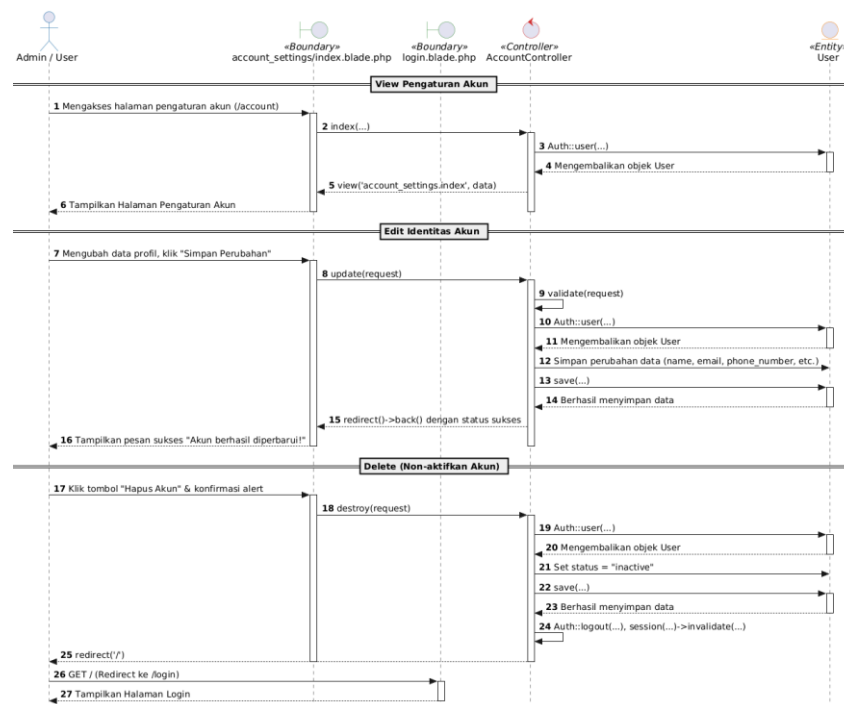
2. Mengatur Identitas Akun

Kode *Sequence* : SD-SMAQ-02

Deskripsi : *Sequence diagram* untuk aktivitas mengatur identitas akun oleh *admin* dan *user*. *Diagram* digambarkan pada pada Gambar 4.4.

Objek :

- a. *Boundary* : login.blade.php, account_settings.blade.php
- b. *Controller* : AccountController
- c. *Entity* : User



Gambar 4.4 *Sequence Diagram Mengatur Identitas Akun*

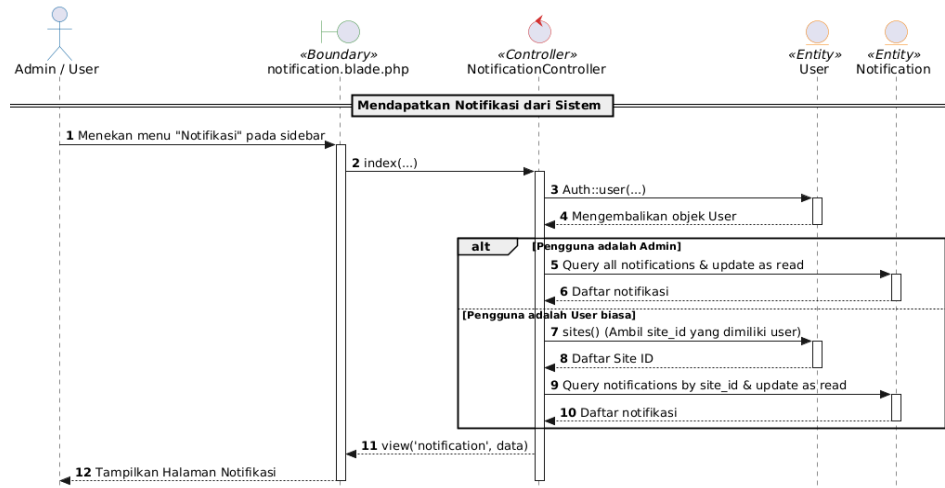
3. Mendapatkan Notifikasi dari Sistem

Kode *Sequence* : SD-SMAQ-03

Deskripsi : *Sequence diagram* untuk aktivitas *mendapatkan notifikasi* dari sistem oleh *admin* dan *user*. *Diagram* digambarkan pada pada Gambar 4.5.

Objek :

- a. *Boundary* : notification.blade.php
- b. *Controller* : NotificationController
- c. *Entity* : User, Notification



Gambar 4.5 *Sequence Diagram* Mendapatkan Notifikasi dari Sistem

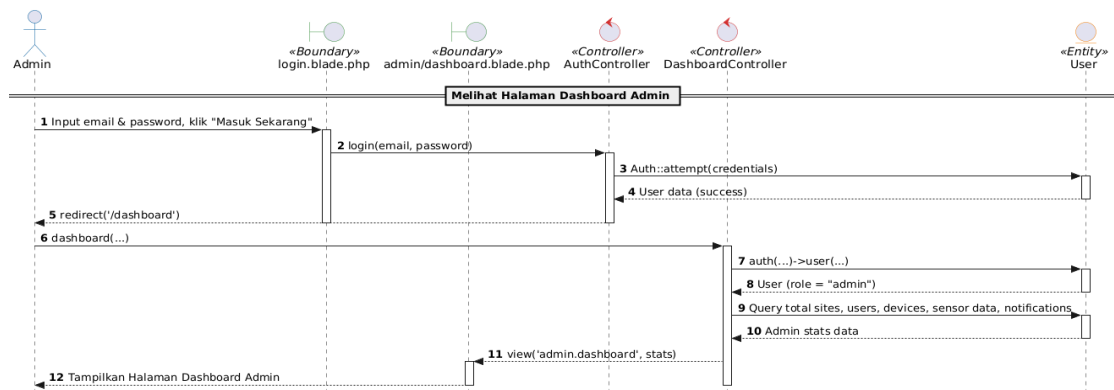
4. Melihat Halaman *Dashboard Admin*

Kode *Sequence* : SD-SMAQ-04

Deskripsi : *Sequence diagram* untuk aktivitas melihat halaman *dashboard admin* oleh *admin*. *Diagram* digambarkan pada pada Gambar 4.6.

Objek :

- Boundary* : login.blade.php, admin/dashboard.blade.php
- Controller* : AuthController, DashboardController
- Entity* : User



Gambar 4.6 *Sequence Diagram* Melihat Halaman Dashboard Admin

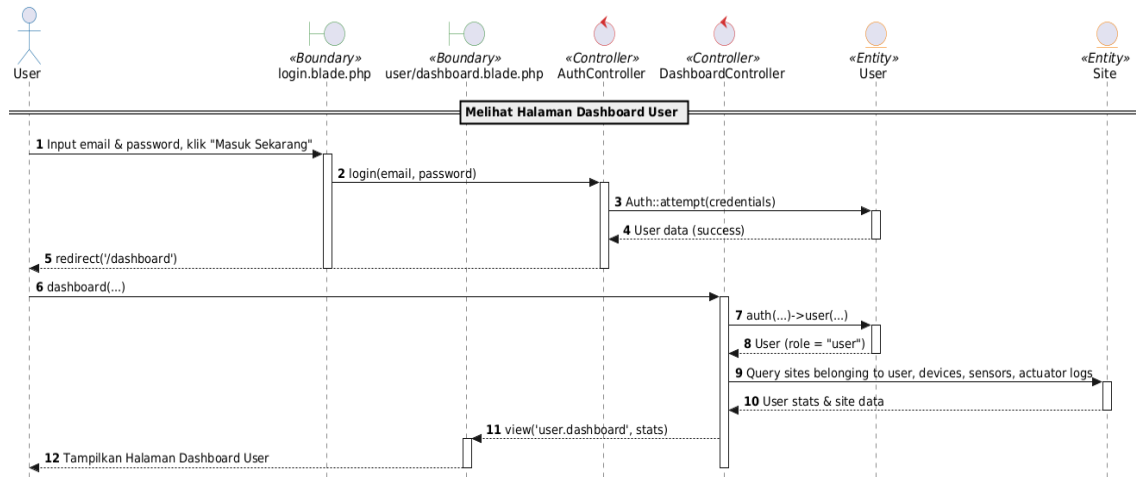
5. Melihat Halaman *Dashboard User*

Kode *Sequence* : SD-SMAQ-05

Deskripsi : *Sequence diagram* untuk aktivitas melihat halaman *dashboard user* oleh *user*. *Diagram* digambarkan pada pada Gambar 4.7.

Objek :

- a. *Boundary* : user/dashboard.blade.php, login.blade.php
- d. *Controller* : AuthController, DashboardController
- b. *Entity* : User, Site



Gambar 4.7 Sequence Diagram Melihat Halaman Dashboard User

6. Mengelola Data User

Kode Sequence : SD-SMAQ-06

Deskripsi : Sequence diagram untuk aktivitas mengelola data user oleh admin. Diagram digambarkan pada pada Gambar 4.8.

Objek :

- a. *Boundary* : admin/users/index.blade.php
- b. *Controller* : UserController
- c. *Entity* : User

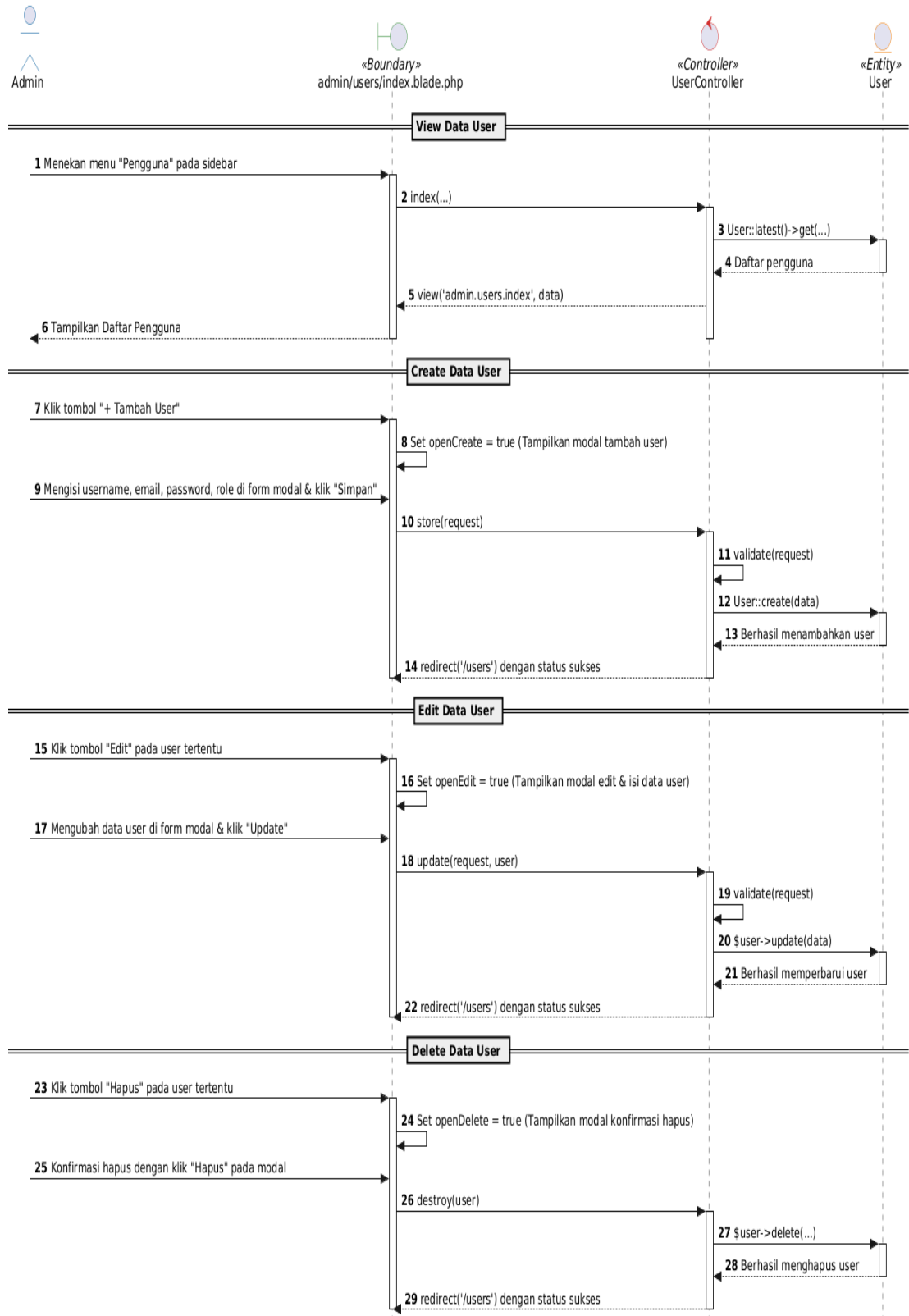
7. Mengelola Data Device

Kode Sequence : SD-SMAQ-07

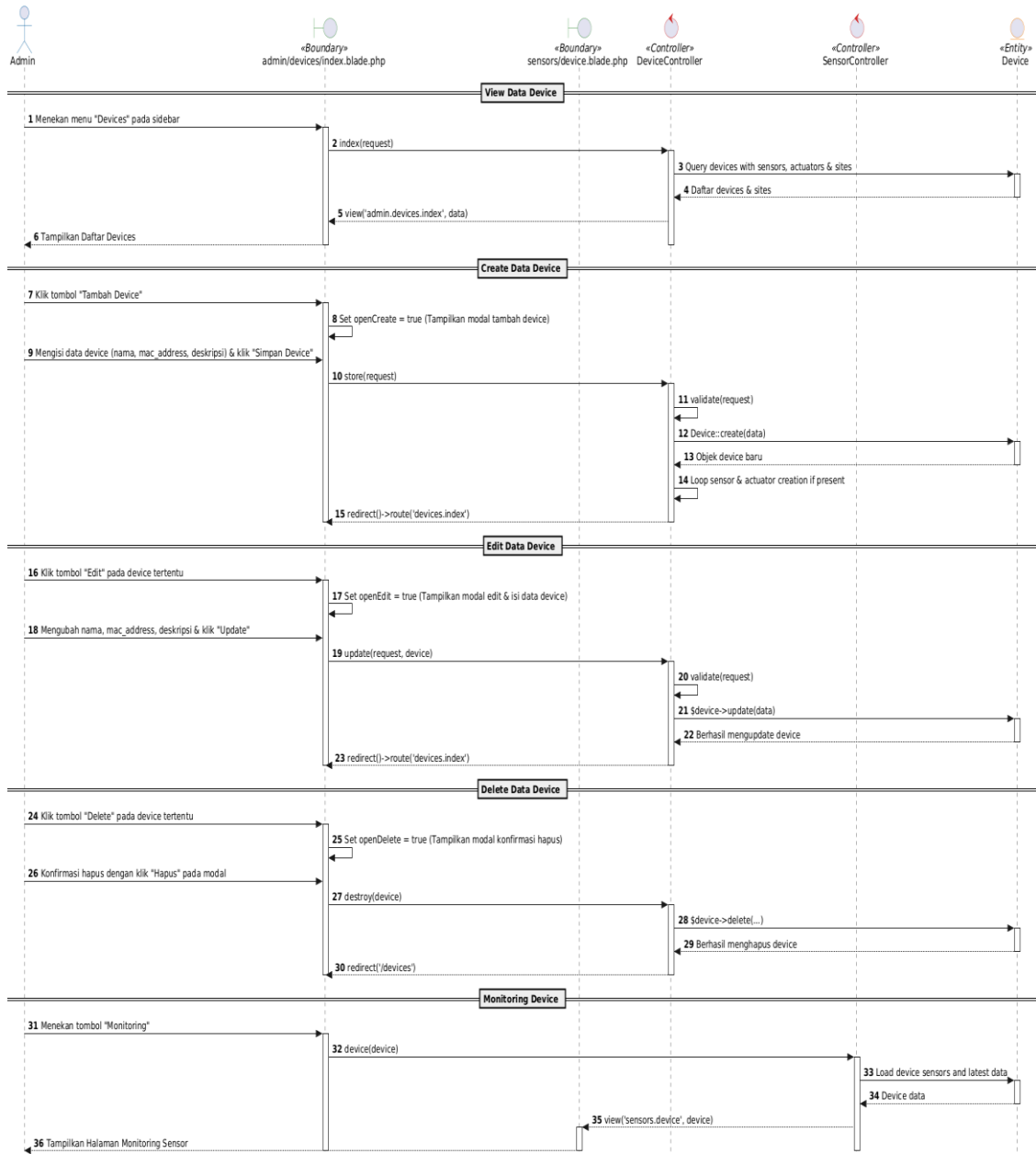
Deskripsi : Sequence diagram untuk aktivitas mengelola data device oleh admin. Diagram digambarkan pada pada Gambar 4.9.

Objek :

- a. *Boundary* : admin/devices/index.blade.php, sensors/device.blade.php
- b. *Controller* : DeviceController, SensorController
- c. *Entity* : Device, Sensor



Gambar 4.8 Sequence Diagram Mengelola Data User



Gambar 4.9 Sequence Diagram Mengelola Data Device

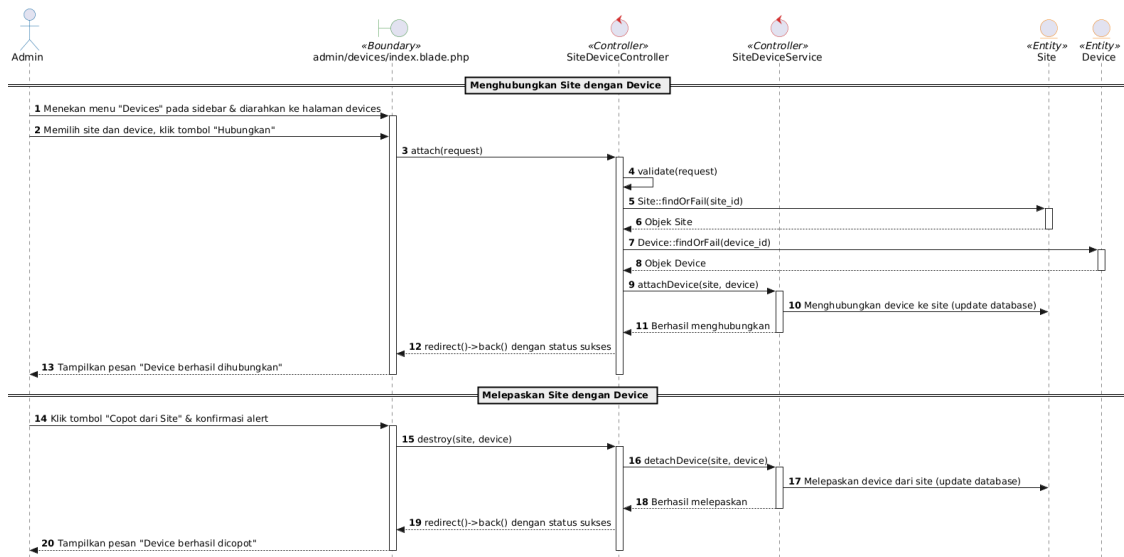
8. Menghubungkan Dan Melepaskan Site dengan Device oleh Admin

Kode Sequence : SD-SMAQ-08

Deskripsi : Sequence diagram untuk aktivitas menghubungkan dan melepaskan site dengan device oleh admin. Diagram digambarkan pada pada Gambar 4.10.

Objek :

- Boundary : admin/devices/index.blade.php
- Controller : SiteDeviceController, SiteDeviceService
- Entity : Site, Device



Gambar 4.10 Sequence Diagram Menghubungkan dan Melepaskan Site dengan oleh Device Admin

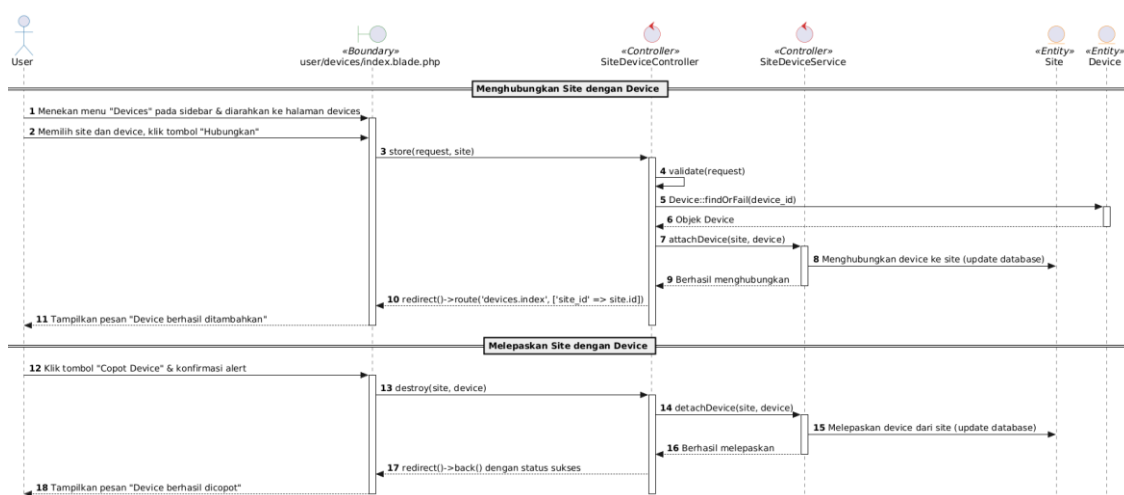
9. Menghubungkan Dan Melepaskan Site dengan Device oleh User

Kode Sequence : SD-SMAQ-09

Deskripsi : Sequence diagram untuk aktivitas menghubungkan dan melepaskan site dengan device oleh user. Diagram digambarkan pada pada Gambar 4.11.

Objek :

- Boundary : user/devices/index.blade.php
- Controller : SiteDeviceController, SiteDeviceService
- Entity : Site, Device



Gambar 4.11 Sequence Diagram Menghubungkan dan Melepaskan Site dengan oleh Device User

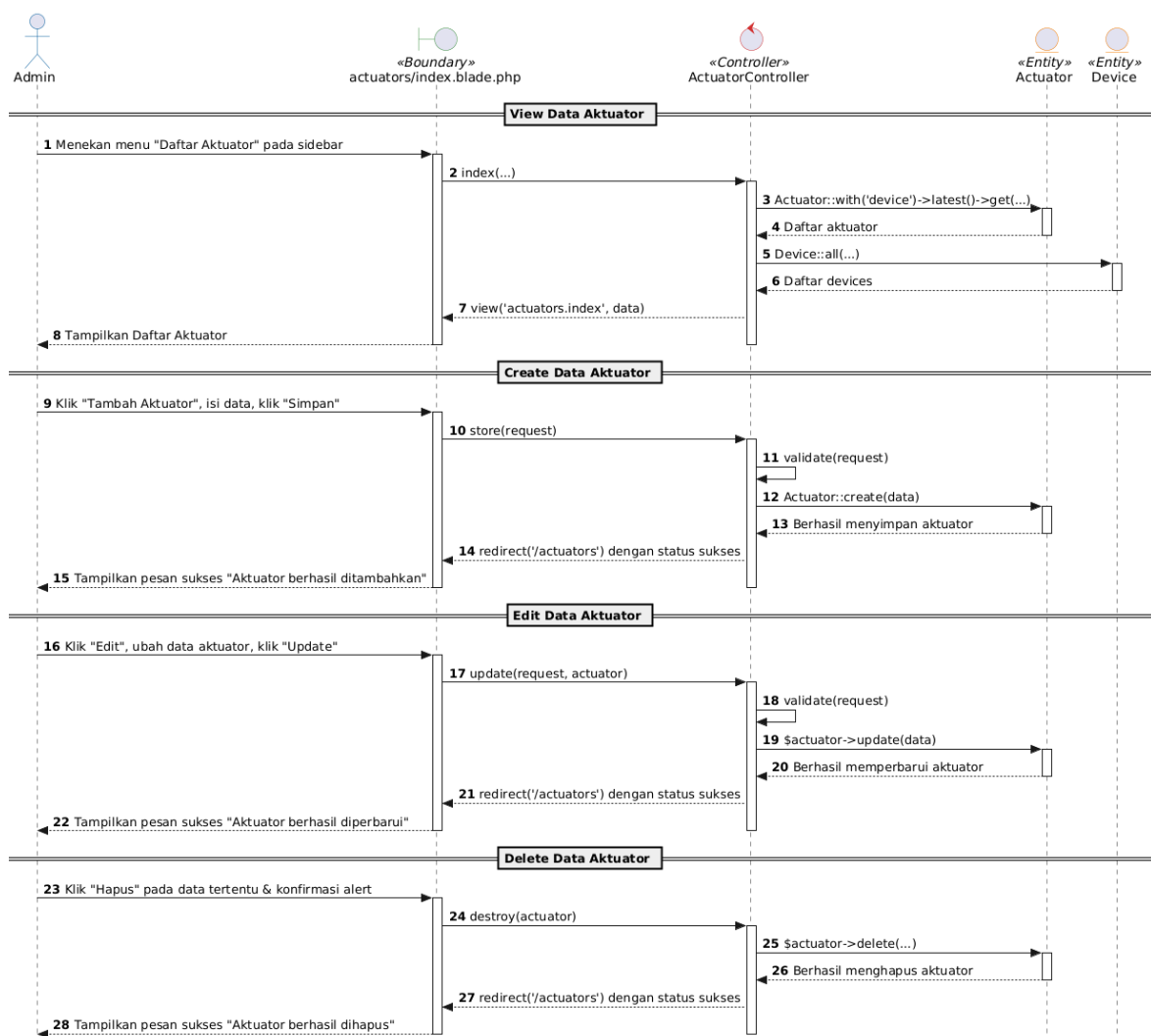
10. Mengelola Data Aktuator

Kode *Sequence* : SD-SMAQ-10

Deskripsi : *Sequence diagram* untuk aktivitas mengelola data aktuator oleh *admin*. Diagram digambarkan pada pada Gambar 4.12.

Objek :

- a. *Boundary* : actuators/index.blade.php
- b. *Controller* : ActuatorController
- c. *Entity* : Actuator, Device



Gambar 4.12 *Sequence Diagram* Mengelola Data Aktuator

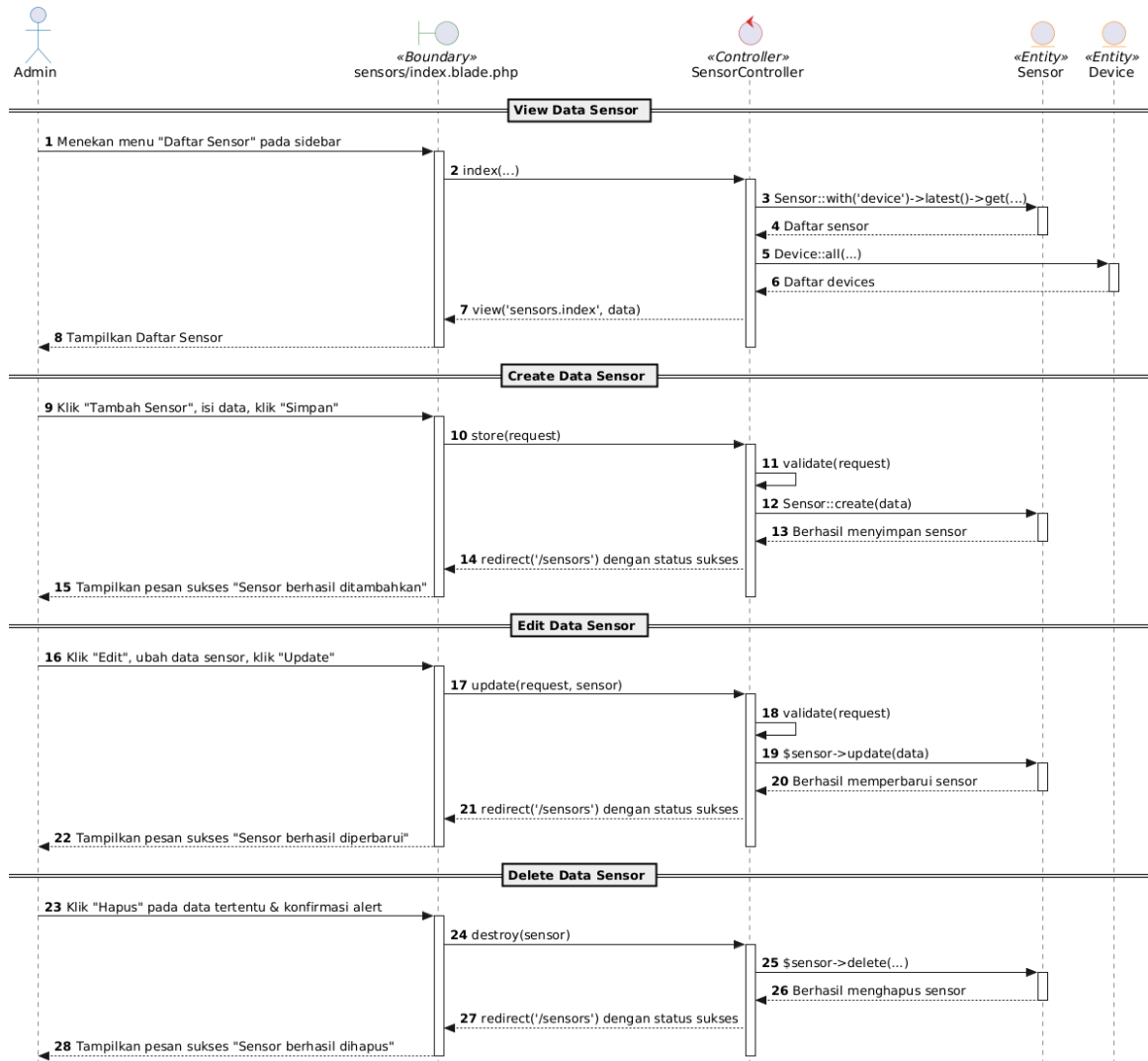
11. Mengelola Data Sensor

Kode *Sequence* : SD-SMAQ-11

Deskripsi : *Sequence diagram* untuk aktivitas mengelola data sensor oleh *admin*. *Diagram* digambarkan pada pada Gambar 4.13.

Objek :

- a. *Boundary* : sensors/index.blade.php
- b. *Controller* : SensorController
- c. *Entity* : Sensor, Device



Gambar 4.13 *Sequence Diagram* Mengelola Data Sensor

12. Mengelola Data Site

Kode *Sequence* : SD-SMAQ-12

Deskripsi : *Sequence diagram* untuk aktivitas mengelola data *site* oleh *admin*. *Diagram* digambarkan pada pada Gambar 4.14.

Objek :

- a. *Boundary* : admin/sites/index.blade.php
- b. *Controller* : SiteController
- c. *Entity* : Site, User



Gambar 4.14 *Sequence Diagram* Mengelola Data Site

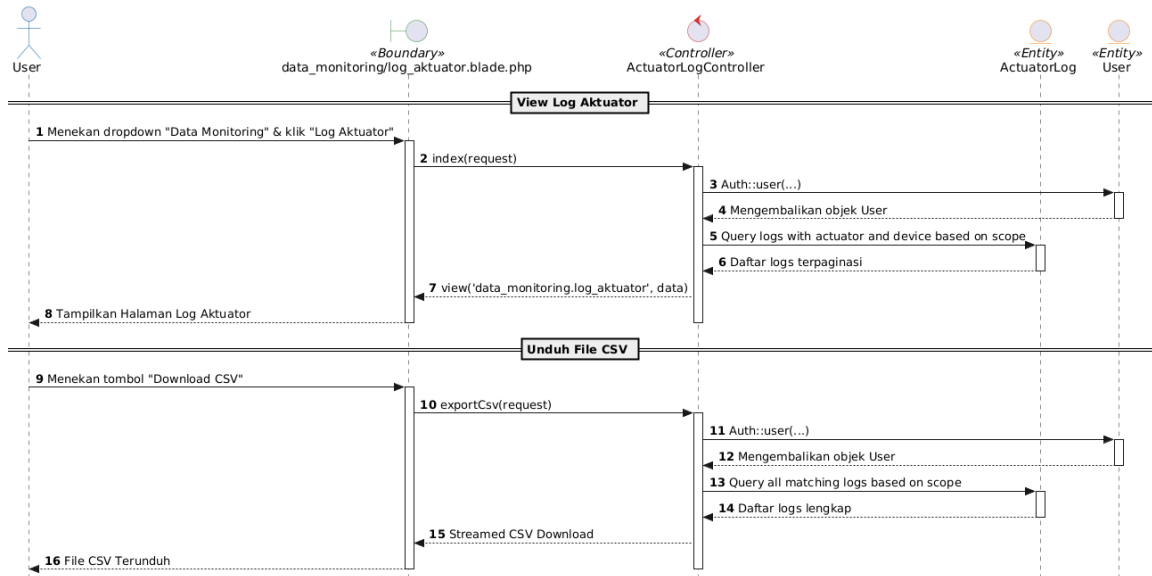
13. Melihat Data Log Aktuator

Kode *Sequence* : SD-SMAQ-13

Deskripsi : *Sequence diagram* untuk aktivitas melihat data log aktuator oleh user. *Diagram* digambarkan pada pada Gambar 4.15.

Objek :

- a. *Boundary* : data_monitoring/log_aktuator.blade
- b. *Controller* : ActuatorLogController
- c. *Entity* : User, ActuatorLog



Gambar 4.15 Sequence Diagram Melihat Data Log Aktuator

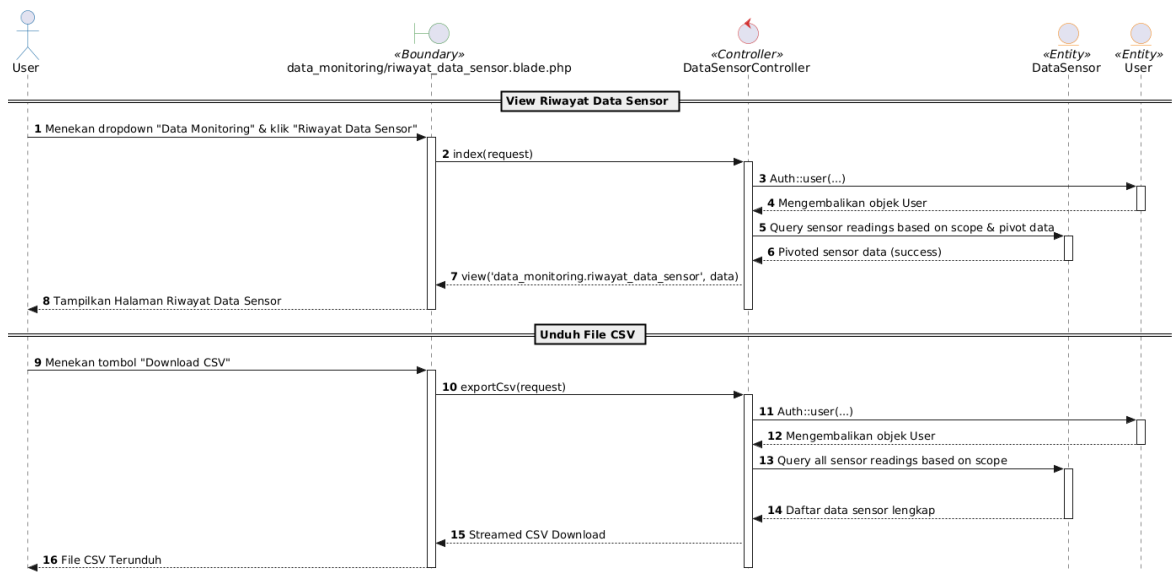
14. Melihat Riwayat Data Sensor

Kode Sequence : SD-SMAQ-14

Deskripsi : Sequence diagram untuk aktivitas melihat riwayat data sensor oleh user. Diagram digambarkan pada pada Gambar 4.16.

Objek :

- Boundary : data_monitoring/riwayat_data_sensor.blade.php
- Controller : DataSensorController
- Entity : User, DataSensor



Gambar 4.16 Sequence Diagram Melihat Riwayat Data Sensor

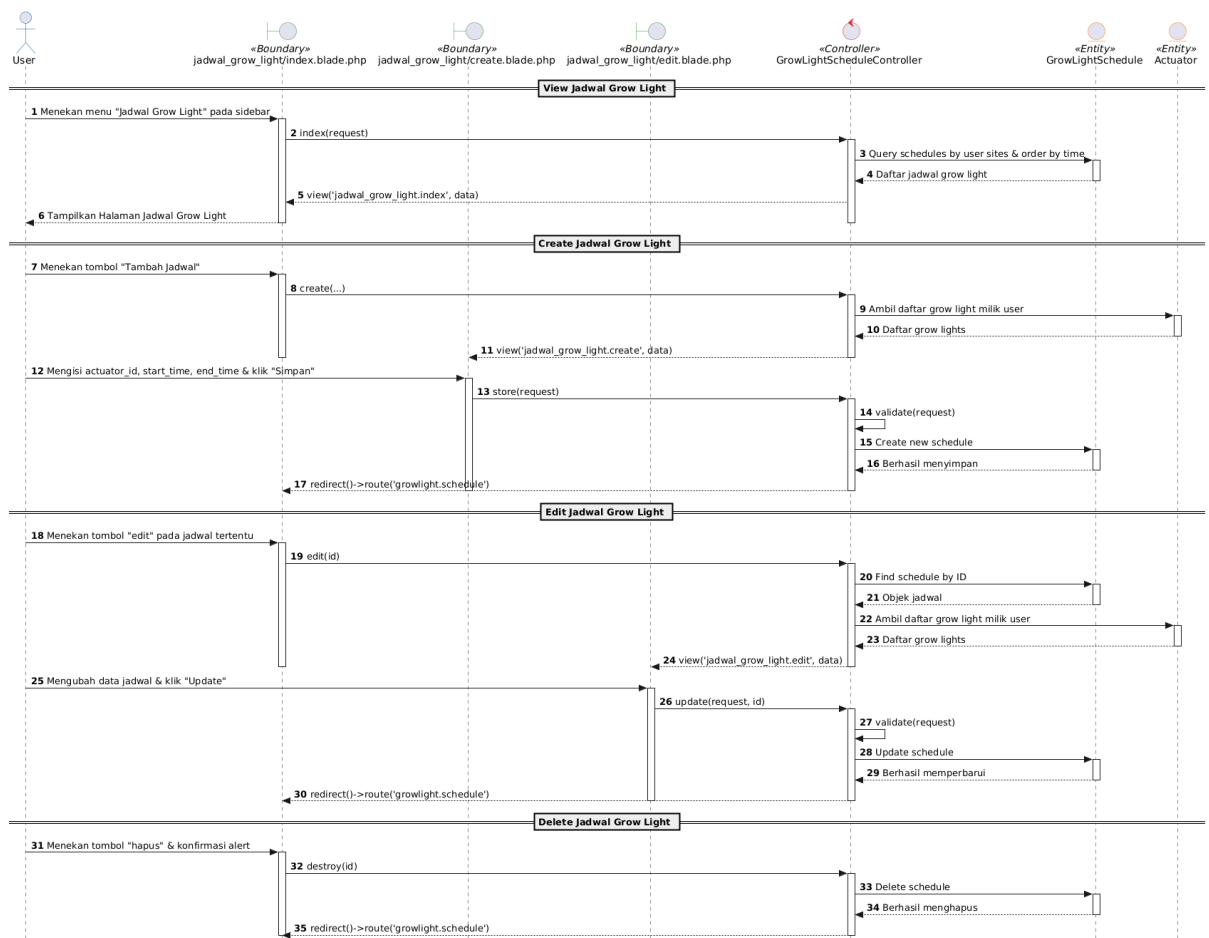
15. Mengelola Jadwal Grow Light

Kode Sequence : SD-SMAQ-15

Deskripsi : *Sequence diagram* untuk aktivitas mengelola jadwal *grow light* oleh *user*. *Diagram* digambarkan pada pada Gambar 4.17.

Objek :

- a. *Boundary* : jadwal_grow_light/index.blade.php
jadwal_grow_light/create.blade.php, jadwal_grow_light/edit.blade.php
- b. *Controller* : GrowLightScheduleController
- c. *Entity* : GrowLightSchedule, Actuator



Gambar 4.17 *Sequence Diagram* Mengelola Jadwal Grow Light

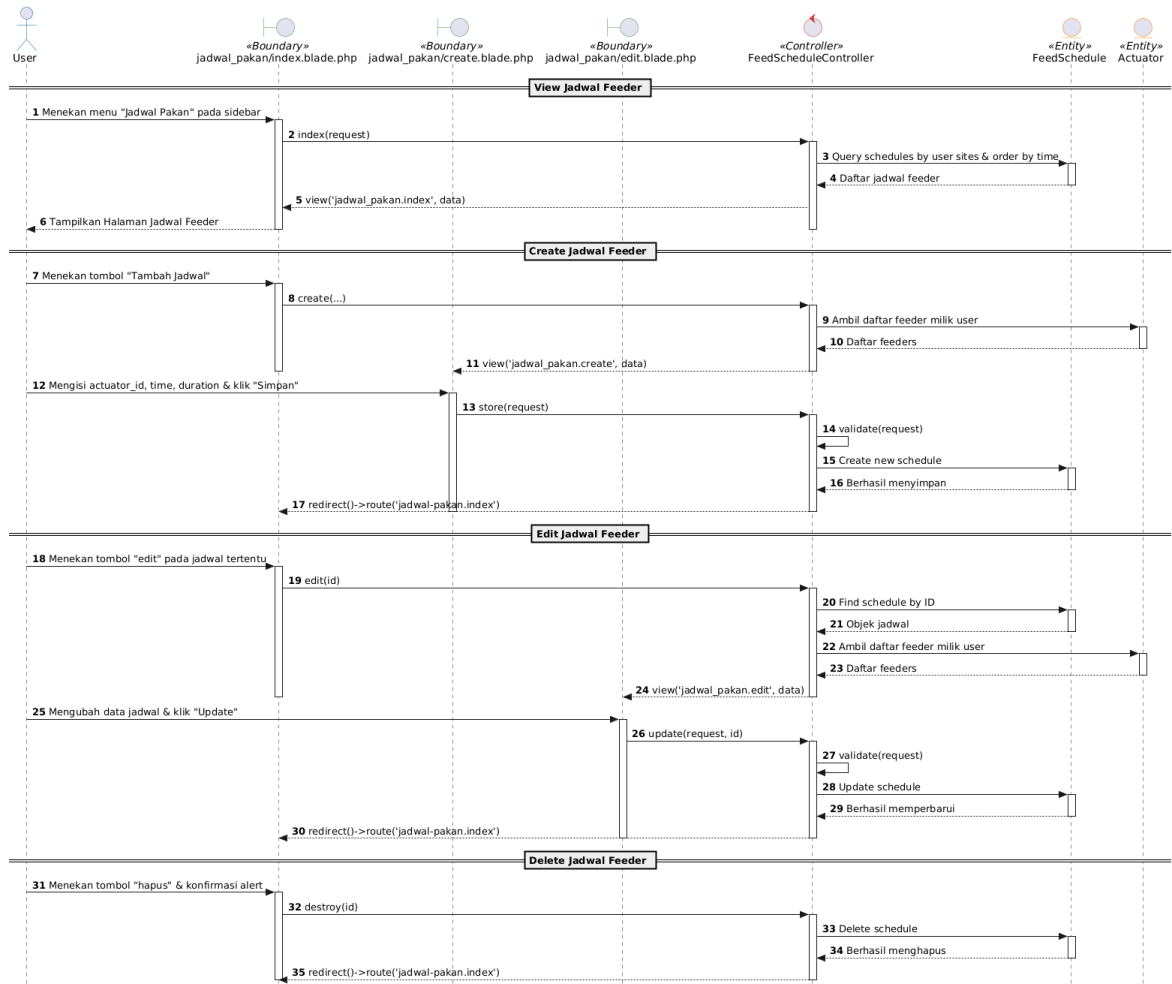
16. Mengelola Jadwal Feeder

Kode Sequence : SD-SMAQ-16

Deskripsi : *Sequence diagram* untuk aktivitas mengelola jadwal *feeder* oleh *user*. *Diagram* digambarkan pada pada Gambar 4.18.

Objek :

- a. *Boundary* : jadwal_pakan/index.blade.php jadwal_pakan/create.blade.php
jadwal_pakan/edit.blade.php
- b. *Controller* : FeedScheduleController
- c. *Entity* : FeedSchedule, Actuator



Gambar 4.18 Sequence Diagram Mengelola Jadwal Feeder

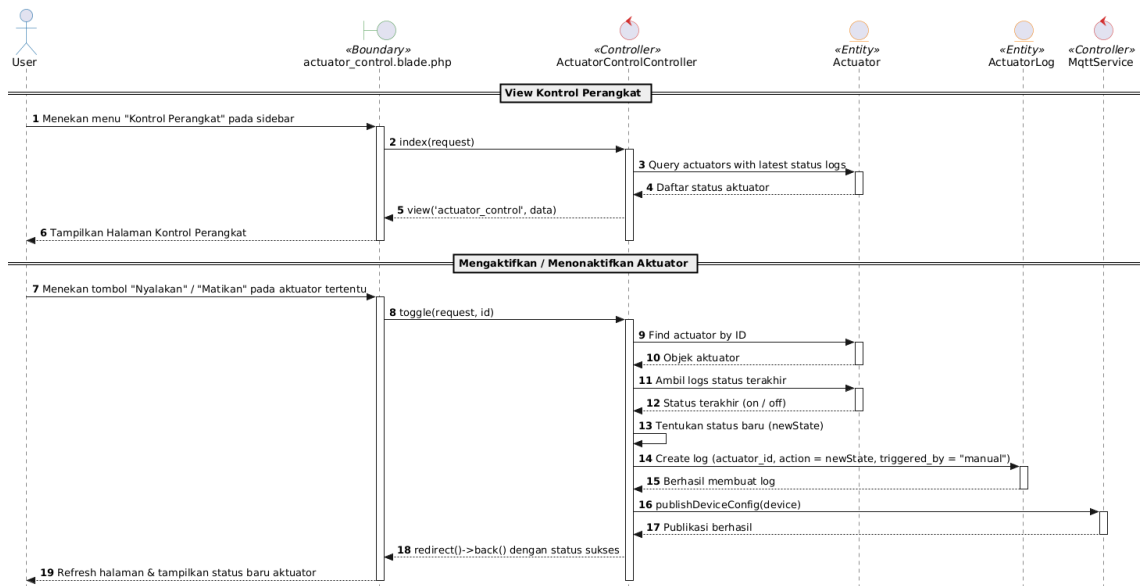
17. Mengaktifkan dan Menonaktifkan Aktuator

Kode Sequence : SD-SMAQ-17

Deskripsi : Sequence diagram untuk aktivitas mengaktifkan atau menonaktifkan aktuator oleh user. Diagram digambarkan pada pada Gambar 4.19.

Objek :

- a. *Boundary* : actuator_control.blade.php
- b. *Controller* : ActuatorControlController, MqttService
- c. *Entity* : Actuator, ActuatorLog

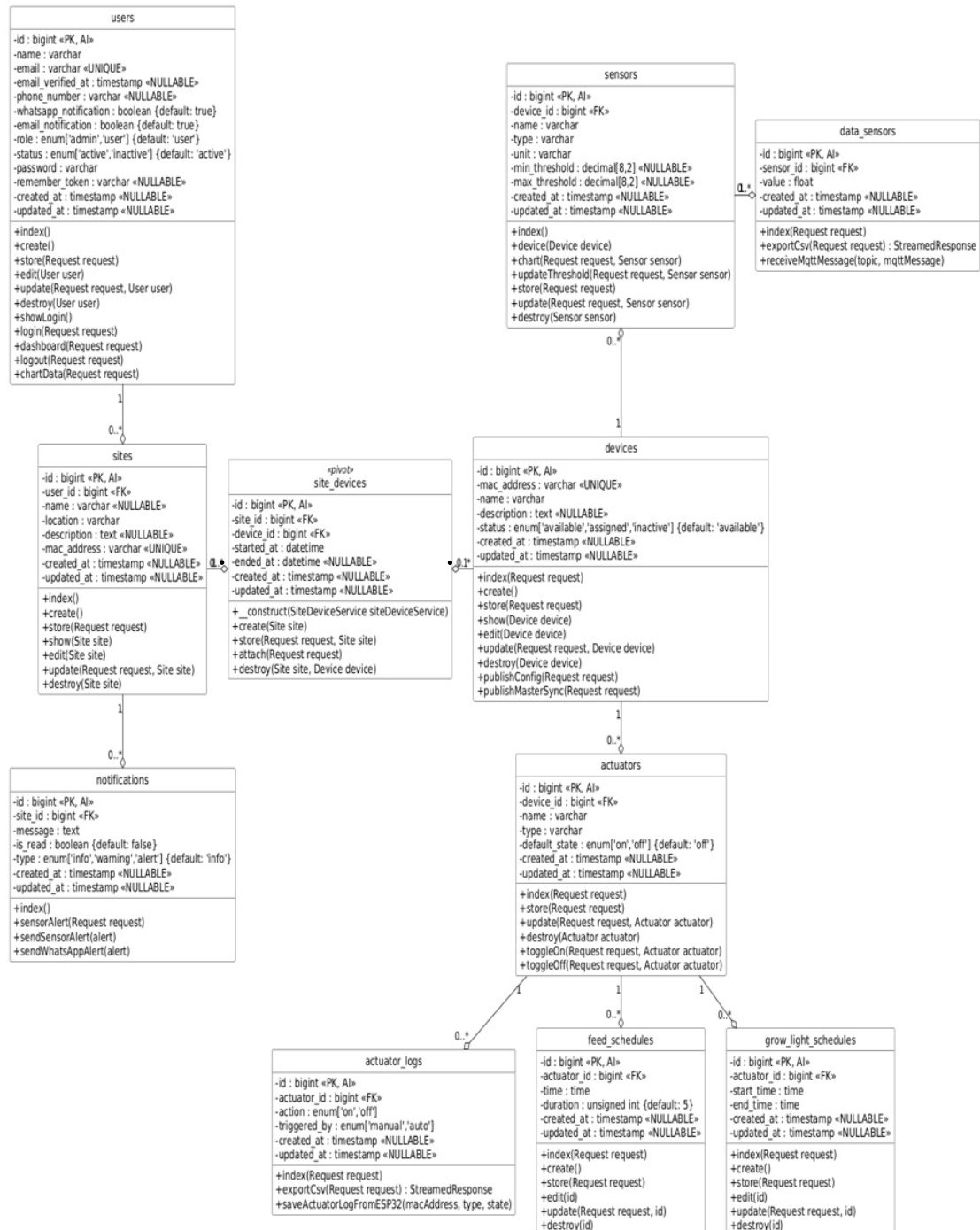


Gambar 4.19 Sequence Diagram Mengaktifkan dan Menon-aktifkan Aktuator

4.2.6 Class Diagram

Class diagram disusun dengan jumlah 11 class dengan relasinya masing-masing antar-class. rancangan class diagram atau database dari sistem ini berpusat pada tiga entitas utama, yaitu pengelolaan data pengguna, sites, dan devices. Untuk bagian hardware, setiap devices bisa memiliki banyak komponen sensor untuk membaca kondisi lingkungan, serta beberapa komponen aktuator yang dikontrol. Semua data hasil pembacaan sensor akan dicatat ke dalam tabel *data_sensors*. Sementara itu, setiap aksi yang dilakukan oleh aktuator akan otomatis terekam di tabel aktuator log. Selain itu, untuk mendukung otomatisasi pada sistem akuaponik ini, saya juga menambahkan tabel *feed_schedules* untuk menjadwalkan pemberian pakan ikan dan *grow_light_schedules* untuk mengatur waktu pencahayaan lampu grow light, di mana keduanya terikat langsung ke masing-masing aktuator yang bersangkutan. Gambar class diagram dapat dilihat pada Gambar 4.20.

Fleksibilitas manajemen operasional di lapangan didukung oleh tabel *intersection site_devices*, di mana sebuah device dapat dipindahkan atau dipetakan ke site yang berbeda dari waktu ke waktu dengan rekam jejak waktu mulai dan waktu selesai. Struktur relasional yang komprehensif ini memastikan seluruh alur pemantauan lingkungan, kontrol otomatisasi, hingga manajemen hak akses pengguna dapat berjalan secara terintegrasi dan terlacak dengan baik. Hasil keseluruhan class diagram dapat dilihat pada Gambar 4.20.

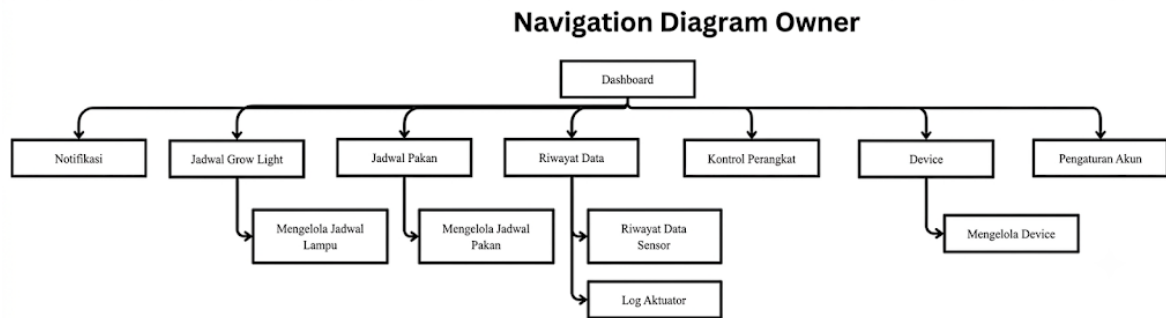


Gambar 4.20 *Class Diagram* Smartaquaponic

4.2.7 Navigation Diagram

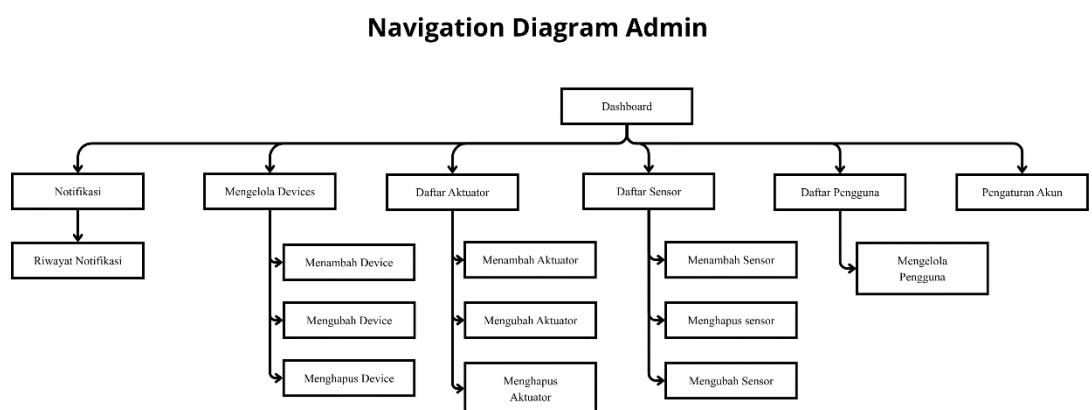
Untuk memberikan gambaran yang jelas mengenai interaksi pengguna dengan sistem, dirancang sebuah diagram navigasi yang memetakan alur perpindahan dari satu halaman ke halaman lainnya. Rancangan ini terbagi menjadi dua diagram navigasi utama berdasarkan

hak akses yaitu *user* dan *admin*, yang digambarkan dan dijelaskan secara berurutan pada Gambar 4.21 dan 4.22.



Gambar 4.21 *Navigation Diagram Owner*

Diagram navigasi bagi aktor *user* dirancang untuk menggambarkan hak akses langsung terhadap monitoring harian sistem akuaponik melalui sebuah *Dashboard* utama. Dari *Dashboard* ini, alur navigasi menyebar ke tujuh menu utama, yaitu Notifikasi, Jadwal *Grow Light*, Jadwal Pakan (*Feeder*), Riwayat Data, Kontrol Perangkat, *Device*, dan Pengaturan Akun. Menu Jadwal *Grow Light* mengarah pada fungsi Mengelola Jadwal Lampu, sedangkan menu Jadwal Pakan memuat fungsi Mengelola Jadwal Pakan. Pada bagian pemantauan, menu Riwayat Data bercabang menjadi Riwayat Data Sensor dan Log Aktuator untuk melihat data yang dihasilkan dari sensor dan status aktuator, sementara menu *Device* menyediakan akses ke sub-menu Mengelola *Device* untuk menyesuaikan integrasi perangkat keras milik pengguna.



Gambar 4.22 *Navigation Diagram Admin*

Navigation diagram bagi aktor *Admin* berfokus pada fungsi untuk mengelola sistem secara keseluruhan melalui *Dashboard* utama *Admin*. Menu utama pada tingkat ini meliputi

Notifikasi, Mengelola *Devices*, Daftar Aktuator, Daftar Sensor, Daftar Pengguna, dan Pengaturan Akun. Alur navigasi pada peran *Admin* cenderung lebih kompleks karena memiliki hak akses hampir setiap elemen sistem. Pada menu Mengelola *Devices*, Daftar Aktuator, dan Daftar Sensor, navigasi diturunkan secara spesifik ke fungsi operasional untuk menambah, mengubah, dan menghapus entitas perangkat keras tersebut dengan menampilkan modal. Selain pengendalian komponen fisik, *Admin* juga dibekali navigasi khusus pada menu Notifikasi untuk melihat Riwayat Notifikasi milik semua *user*, serta menu Daftar Pengguna yang mengarah langsung pada fungsi Mengelola Pengguna untuk mengelola seluruh akun *user* yang terdaftar dalam sistem.

4.2.8 *Product Backlog*

Product Backlog merupakan sebuah daftar terstruktur yang memuat seluruh kebutuhan, fitur, fungsi, dan perbaikan yang harus diimplementasikan ke dalam sistem selama siklus pengembangan perangkat lunak berlangsung. Di dalam metodologi Scrum, *Product Backlog* ini bersifat dinamis dan dapat diubah secara berkala berdasarkan hasil diskusi serta validasi bersama *Product Owner*. Setiap poin di dalam daftar ini dinilai berdasarkan kompleksitasnya menggunakan *Effort Point*. Estimasi *Effort Point* ditentukan secara kolaboratif melalui kesepakatan tim pengembang guna mengukur estimasi waktu pengerjaan yang diperlukan dalam satu *item backlog*.

Keberadaan *item backlog* berfungsi sebagai landasan untuk menjaga transparansi kerja dan menentukan prioritas fitur yang akan ditarik ke dalam *sprint backlog* dalam tahap *Sprint Planning*. Rangkaian *Product Backlog Item* tersebut dikelompokkan secara logis ke dalam beberapa pilar komponen utama rekayasa perangkat lunak, meliputi perancangan antarmuka pengguna (*Front-End/FE*), pembangunan logika fungsional server (*Back-End/BE*), pengintegrasian jalur komunikasi mikrokontroler (*Integrate*), serta persiapan instalasi sistem. Berdasarkan identifikasi tersebut, sistem dikelompokkan menjadi 64 *product backlog item* dengan akumulasi beban kerja sebesar 297 *Effort Point* yang mengindikasikan 297 jam pengerjaan. Total beban poin kerja ini akan didistribusikan secara adaptif ke dalam beberapa tahapan *sprint* guna memastikan fungsionalitas sistem informasi dengan skema perluasan perangkat keras berbasis Laravel dapat terealisasi secara bertahap dan teruji. Daftar kebutuhan fungsional dan teknis yang telah diidentifikasi untuk sistem monitoring akuaponik terintegrasi ini dirangkum ke dalam Tabel 4.21.

Tabel 4.21 *Product Backlog Smartaquaponic*

No	Kode Product Backlog Item	Deskripsi	Effort Point
1	PB-AKP-1	[FE] Layout - Login Page	2
2	PB-AKP-2	[FE] Layout - Users Management Page (Admin)	6
3	PB-AKP-3	[FE] Layout - Dashboard Monitoring Page (Admin)	10
4	PB-AKP-4	[FE] Layout - Dashboard Monitoring Page (User)	6
5	PB-AKP-5	[FE] Layout - Device Management Page (Admin)	6
6	PB-AKP-6	[FE] Layout - Device Management Page (User)	4
7	PB-AKP-7	[FE] Layout - Sensor Management Page	4
8	PB-AKP-8	[FE] Layout - Site Monitoring (Admin)	8
9	PB-AKP-9	[FE] Layout – Site Management (User)	8
10	PB-AKP-10	[FE] Layout - Actuator Management Page	5
11	PB-AKP-11	[FE] Layout - Feed Schedule Page	8
12	PB-AKP-12	[FE] Layout – Grow Light Schedule Page	8
13	PB-AKP-13	[FE] Layout - Sensor History Page	6
14	PB-AKP-14	[FE] Layout - Actuator Log Page	4
15	PB-AKP-15	[FE] Layout - Actuator Control Page	10
16	PB-AKP-16	[FE] Layout - Profile Management Page	3
17	PB-AKP-17	[FE] Layout - Responsive Mobile Navigation	5
18	PB-AKP-18	[FE] Layout - Realtime Monitoring Chart	12
19	PB-AKP-19	[FE] Layout - Notification Page	5
20	PB-AKP-20	[BE] Authentication System	1
21	PB-AKP-21	[BE] User & Profile Management System	3
22	PB-AKP-22	[BE] Site Management System	8
23	PB-AKP-23	[BE] Device Management System	5
24	PB-AKP-24	[BE] Sensor Management System	2
25	PB-AKP-25	[BE] Actuator Management System	5
26	PB-AKP-26	[BE] Feed Scheduling System	8
27	PB-AKP-27	[BE] Grow Light Scheduling System	8
28	PB-AKP-28	[BE] Sensor & Actuator History System	5
29	PB-AKP-29	[BE] MQTT for IoT Communication	6
30	PB-AKP-30	[BE] Database Design & Configuration	10
31	PB-AKP-31	[BE] WhatsApp Notification System	8
32	PB-AKP-32	[BE] Email Notification System	6
33	PB-AKP-33	[Integrate] MQTT Broker Integration	3
34	PB-AKP-34	[Integrate] ESP32 Master Communication	2
35	PB-AKP-35	[Integrate] ESP32 Slave Device Communication	2
36	PB-AKP-36	[Integrate] Temperature Sensor Monitoring	3
37	PB-AKP-37	[Integrate] pH Sensor Monitoring	4
38	PB-AKP-38	[Integrate] Humidity Sensor Monitoring	3

No	Kode Product Backlog Item	Deskripsi	Effort Point
39	PB-AKP-39	<i>[Integrate] Turbidity Sensor Monitoring</i>	4
40	PB-AKP-40	<i>[Integrate] Relay Actuator Control</i>	4
41	PB-AKP-41	<i>[Integrate] Automatic Feeder Scheduling</i>	8
42	PB-AKP-42	<i>[Integrate] Automatic Lighting System</i>	8
43	PB-AKP-43	<i>[Integrate] Realtime Monitoring System</i>	5
44	PB-AKP-44	<i>[Finalization] Assembly of Casing and IoT Device Protectors</i>	2
45	PB-AKP-45	<i>[Finalization] File Migration to Shared Hosting</i>	6
46	PB-AKP-46	<i>[Finalization] MQTT Configuration in Hosting Environment</i>	2
47	PB-AKP-47	<i>[Finalization] Website Finalization in Hosting Environment</i>	2
48	PB-AKP-48	<i>[Finalization] Finalization of Cable Routing and Tool Electricals</i>	4
49	PB-AKP-49	<i>[Feedback] Sprint 1 Evaluation</i>	1
50	PB-AKP-50	<i>[Feedback] Sprint 2 Evaluation</i>	1
51	PB-AKP-51	<i>[Feedback] Sprint 3 Evaluation</i>	1
52	PB-AKP-52	<i>[FE] Layout - Adding Filter at Sensor History page</i>	2
53	PB-AKP-53	<i>[FE] Layout - Adding Export Feature at Actuator Log page</i>	2
54	PB-AKP-54	<i>[BE] Adding Filter Sensor & Actuator History System</i>	2
55	PB-AKP-55	<i>[FE] Layout - Notification Page Icon</i>	2
56	PB-AKP-56	<i>[BE] Enhance Site Management System</i>	2
57	PB-AKP-57	<i>[FE] Layout - Table Changes at Sensor Management Page</i>	2
58	PB-AKP-58	<i>[FE] Layout - Table Changes at Actuator Management Page</i>	2
59	PB-AKP-59	<i>[BE] Data Retrieval Changes At Sensor Management System</i>	2
60	PB-AKP-60	<i>[BE] Data Retrieval Changes At Actuator Management System</i>	2
61	PB-AKP-61	<i>Instalasi Alat</i>	10
62	PB-AKP-62	<i>[Finalization] Data Migration</i>	1
63	PB-AKP-63	<i>[Finalization] Confirmation for System Installation</i>	2
Total Effort Point			297

4.2.9 Definition of Done

Dalam pelaksanaan penelitian skripsi yang menggunakan metode Scrum, keterbukaan informasi dan penjaminan mutu terhadap setiap luaran fitur memegang peranan yang sangat penting. Oleh karena itu, disusunlah sebuah *Definition of Done* (DoD) sebagai komitmen formal dan standar kualitas yang berlaku bagi seluruh *item* pada *product backlog* di setiap Sprint.

DoD berfungsi sebagai instrumen transparansi guna menyamakan persepsi mengenai batasan atau kriteria suatu fitur telah selesai dikembangkan sepenuhnya dan siap dinyatakan sebagai bagian dari komponen produk (*Increment*) yang berfungsi dengan baik. Apabila suatu *item* yang sedang berjalan belum memenuhi seluruh kriteria di dalam DoD hingga akhir masa *Sprint*, maka item tersebut tidak dapat dianggap sebagai *Increment*, tidak akan didemonstrasikan pada saat *Sprint Review*, serta harus dikembalikan ke dalam Product Backlog untuk dijadwalkan ulang pada *Sprint* berikutnya. DoD sistem Smartaquaponics yang digunakan sepanjang siklus pengembangan disajikan pada Tabel 4.22 berikut:

Tabel 4.22 *Definition of Done* (DoD)

No	<i>Definition of Done</i>
1	Fitur atau modul berhasil diimplementasikan sesuai dengan deskripsi yang tercantum pada <i>Product Backlog Item</i> .
2	Modul IoT (pembacaan sensor dan kontrol aktuator) dapat berjalan dengan stabil tanpa mengalami kegagalan fungsi.
3	Fitur telah terintegrasi penuh ke dalam sistem Smartaquaponics dan tidak mengganggu fungsi lain yang telah berjalan.
4	Fitur siap untuk didemonstrasikan di hadapan dosen pembimbing/pemangku kepentingan pada saat <i>Sprint Review</i>

4.3 Implementasi Scrum

Tahapan implementasi pengembangan Sistem Smartaquaponic Berbasis IoT dan *Real-Time Dashboard* ini dilakukan dengan menerapkan kerangka kerja (*framework*) Scrum. Proses eksekusi pengembangan dibagi ke dalam tiga siklus *Sprint* yang setiap *sprint* dilaksanakan selama 7 hari termasuk hari libur. Terdapat dua anggota Scrum tim yang bertugas untuk menjalankan implementasi program yaitu Daffa Fairuz Annizari (penulis) sebagai *Web developer* dan Bima Aditya Aryono sebagai *IoT Engineer*. Meskipun dua anggota scrum tersebut diidentifikasi sebagai *Web Developer* dan *IoT engineer*, pembagian tugas keduanya tidak terlalu ketat sehingga *web developer* juga mendapatkan *item* yang berhubungan dengan IoT dan begitu juga sebaliknya.

Pengerjaan *item sprint backlog* dibagikan berdasarkan *effort point* sehingga tugas dapat terbagi secara merata dalam Scrum tim karena keduanya juga memiliki kemampuan dalam bidang lainnya sehingga pembagian sistem pembagian tersebut tidak menjadi

masalah. Pembagian ini bertujuan untuk memastikan setiap tahapan pengembangan berjalan secara adaptif dan sesuai *planning* yang dijalankan seluruh tim Scrum.

4.3.1 Sprint 1

Sprint 1 dilaksanakan selama satu minggu (7 hari), terhitung mulai tanggal 13 Mei 2026 sampai dengan 19 Mei 2026. Fokus utama dari *Sprint* 1 ini adalah pengerjaan seluruh rancangan antarmuka pengguna (*layout*) sistem akuaponik berbasis web serta inisialisasi konfigurasi basis data awal sebagai landasan sistem.

4.3.1.1 Sprint Planning 1

Tahapan *Sprint Planning* 1 dilaksanakan pada hari pertama, yaitu tanggal 13 Mei 2026. Berdasarkan hasil diskusi seluruh *Scrum team*, ditentukan tujuan *Sprint* 1 (*Sprint Goal*), yaitu menghasilkan *interface web* fungsional yang responsif terhadap perangkat yang membuka *website* tersebut, beserta skema struktur basis data. *Sprint Backlog* yang ditarik dari *Product Backlog* terdiri dari 20 item ditambah dengan 9 item tambahan yang ditambahkan saat *sprint* 1 berlangsung sehingga total akhir *effort point* adalah 175 *effort point*. Daftar distribusi *Sprint Backlog* 1 ini dirangkum dalam Tabel 4.23.

Tabel 4.23 *Sprint Backlog* 1 Smartaquaponic

No	Kode Product Backlog Item	Deskripsi	Effort Point	Developer
1	PB-SMAQ-1	[FE] Layout - Login Page	2	Bima
2	PB-SMAQ-2	[FE] Layout - Users Management Page (Admin)	6	Bima
3	PB-SMAQ-3	[FE] Layout - Dashboard Monitoring Page (Admin)	10	Bima
4	PB-SMAQ-4	[FE] Layout - Dashboard Monitoring Page (User)	6	Bima
5	PB-SMAQ-5	[FE] Layout - Device Management Page (Admin)	6	Bima
6	PB-SMAQ-6	[FE] Layout - Device Management Page (User)	4	Bima
7	PB-SMAQ-7	[FE] Layout - Sensor Management Page	4	Bima
8	PB-SMAQ-8	[FE] Layout - Site Monitoring (Admin)	8	Bima
9	PB-SMAQ-9	[FE] Layout - Site Management (User)	8	Bima
10	PB-SMAQ-10	[FE] Layout - Actuator Management Page	5	Daffa
11	PB-SMAQ-11	[FE] Layout - Feed Schedule Page	8	Daffa
12	PB-SMAQ-12	[FE] Layout - Grow Light Schedule Page	8	Daffa
13	PB-SMAQ-13	[FE] Layout - Sensor History Page	6	Daffa
14	PB-SMAQ-14	[FE] Layout - Actuator Log Page	4	Daffa
15	PB-SMAQ-15	[FE] Layout - Actuator Control Page	10	Daffa

No	Kode Product Backlog Item	Deskripsi	Effort Point	Developer
16	PB-SMAQ-16	[FE] Layout - Profile Management Page	3	Daffa
17	PB-SMAQ-17	[FE] Layout - Responsive Mobile Navigation	5	Daffa
18	PB-SMAQ-18	[FE] Layout - Realtime Monitoring Chart	12	Bima
19	PB-SMAQ-19	[FE] Layout - Notification Page	5	Daffa
30	PB-SMAQ-30	[BE] Database Design dan Configuration	10	Daffa
20	PB-SMAQ-20	[BE] Authentication System	1	Bima
21	PB-SMAQ-21	[BE] User dan Profile Management System	3	Daffa
22	PB-SMAQ-22	[BE] Site Management System	8	Bima
23	PB-SMAQ-23	[BE] Device Management System	5	Bima
24	PB-SMAQ-24	[BE] Sensor Management System	2	Bima
25	PB-SMAQ-25	[BE] Actuator Management System	5	Daffa
26	PB-SMAQ-26	[BE] Feed Scheduling System	8	Daffa
27	PB-SMAQ-27	[BE] Grow Light Scheduling System	8	Daffa
28	PB-SMAQ-28	[BE] Sensor dan Actuator History System	5	Daffa

4.3.1.2 Implementasi Sprint 1

Implementasi antarmuka pada Sprint 1 ini mencakup pengembangan halaman-halaman utama sistem, mulai dari proses autentikasi hingga manajemen data lingkungan akuaponik. Secara garis besar, antarmuka yang dikembangkan dirancang untuk memberikan kemudahan bagi *admin* maupun *user* dalam memantau dan mengelola sistem.

Pada hari pertama, yaitu tanggal 13 Mei 2026, kegiatan yang dilakukan adalah *Sprint Planning* 1. Pada pertemuan ini, seluruh *product backlog* yang telah diprioritaskan sebelumnya dibahas kembali untuk menentukan ruang lingkup pekerjaan yang akan dikerjakan selama Sprint 1. Selain itu, tim juga melakukan pembagian tugas bagi setiap anggota *developer* sehingga setiap *backlog* memiliki penanggung jawab yang jelas. Karena hari pertama difokuskan pada perencanaan *sprint*, belum terdapat *backlog* yang mulai dikerjakan maupun kendala yang ditemukan.

Pada hari kedua, yaitu tanggal 14 Mei 2026, *Web Developer* berhasil menyelesaikan perancangan dan konfigurasi basis data (PB-SMAQ-30) serta sistem manajemen pengguna dan profil (PB-SMAQ-21). Penyelesaian kedua *backlog* tersebut menjadi fondasi penting bagi pengembangan fitur-fitur lainnya karena berkaitan dengan penyimpanan data dan pengelolaan akun pengguna. Selanjutnya, *Web Developer* merencanakan pengerjaan halaman log aktuator (PB-SMAQ-14) dan halaman kontrol aktuator (PB-SMAQ-15). Berikut ini adalah hasil dari konfigurasi basis data pada sistem Smartaquaponic.

Pada hari yang sama, IoT *Engineer* berhasil menyelesaikan tampilan halaman *login* (PB-SMAQ-1), tampilan halaman manajemen pengguna untuk *admin* (PB-SMAQ-2), serta sistem autentikasi pengguna (PB-SMAQ-20). Ketiga *backlog* tersebut menjadi dasar bagi proses *login* dan pengelolaan akun pada sistem. Setelah *backlog* tersebut selesai, IoT *Engineer* merencanakan pengerjaan tampilan *dashboard monitoring* untuk *admin* (PB-SMAQ-3) dan *user* (PB-SMAQ-04). Tidak terdapat kendala yang dilaporkan oleh kedua anggota tim pada hari kedua. Karena terdapat pengerjaan tambahan tersebut, tim *developer* mengajukan adanya penambahan *sprint backlog* pada *sprint 1* ini kepada *Scrum master* sehingga terdapat 9 *sprint* tambahan pada *sprint 1*.

Pada hari ketiga, yaitu tanggal 15 Mei 2026, Web *Developer* berhasil menyelesaikan halaman *log* aktuator (PB-SMAQ-14) dan halaman kontrol aktuator (PB-SMAQ-15). Kedua fitur ini digunakan untuk memantau aktivitas aktuator serta memberikan kemampuan kepada pengguna untuk melakukan pengendalian perangkat melalui sistem. Setelah penyelesaian *backlog* tersebut, Web *Developer* melanjutkan pengerjaan sistem manajemen aktuator (PB-SMAQ-25), tampilan halaman manajemen aktuator (PB-SMAQ-10), tampilan halaman jadwal pemberian pakan (PB-SMAQ-11), dan sistem penjadwalan pemberian pakan (PB-SMAQ-26).

Sementara itu, IoT *Engineer* berhasil menyelesaikan tampilan *dashboard monitoring* untuk *admin* (PB-SMAQ-3) dan tampilan *dashboard monitoring* untuk pengguna (PB-SMAQ-4). Kedua halaman tersebut berfungsi sebagai pusat informasi untuk menampilkan data *monitoring* sistem akuaponik. Selanjutnya, IoT *Engineer* merencanakan pengerjaan tampilan halaman manajemen perangkat untuk *admin* (PB-SMAQ-5) dan tampilan halaman manajemen perangkat untuk pengguna (PB-SMAQ-6). Pada hari ketiga tidak ditemukan kendala yang menghambat proses pengembangan.

Pada hari keempat, yaitu tanggal 16 Mei 2026, Web *Developer* berhasil menyelesaikan tampilan halaman manajemen aktuator (PB-SMAQ-10), tampilan halaman jadwal pemberian pakan (PB-SMAQ-11), sistem manajemen aktuator (PB-SMAQ-25), dan sistem penjadwalan pemberian pakan (PB-SMAQ-26). Penyelesaian *backlog* tersebut memungkinkan pengguna untuk mengelola aktuator dan mengatur jadwal pemberian pakan secara terstruktur melalui sistem.

Setelah *backlog* tersebut selesai, Web Developer melanjutkan pengerjaan halaman notifikasi (PB-SMAQ-19) dan halaman manajemen profil (PB-SMAQ-16). Di sisi lain, IoT Engineer masih melanjutkan pengerjaan tampilan halaman manajemen perangkat untuk *admin* (PB-SMAQ-5), tampilan halaman manajemen perangkat untuk pengguna (PB-SMAQ-6), serta sistem manajemen perangkat (PB-SMAQ-23). Pada hari keempat tidak terdapat kendala yang dilaporkan.

Pada hari kelima, yaitu tanggal 17 Mei 2026, Web Developer berhasil menyelesaikan halaman notifikasi (PB-SMAQ-19) dan halaman manajemen akun (PB-SMAQ-16). Halaman notifikasi digunakan untuk menampilkan informasi penting yang berkaitan dengan kondisi sistem, sedangkan halaman manajemen profil digunakan untuk mengelola data akun pengguna.

Setelah *backlog* tersebut selesai, Web Developer melanjutkan pengerjaan halaman penjadwalan *grow light* (PB-SMAQ-12), halaman riwayat sensor (PB-SMAQ-13), halaman *log* aktuator (PB-SMAQ-14), dan halaman kontrol aktuator (PB-SMAQ-15). Pada waktu yang sama, IoT Engineer berhasil menyelesaikan tampilan halaman manajemen perangkat untuk *admin* (PB-SMAQ-5), tampilan halaman manajemen perangkat untuk pengguna (PB-SMAQ-6), serta sistem manajemen perangkat (PB-SMAQ-23). Selanjutnya, IoT Engineer merencanakan pengerjaan tampilan halaman manajemen sensor (PB-SMAQ-7) dan sistem manajemen sensor (PB-SMAQ-24). Tidak terdapat kendala yang ditemukan selama pengerjaan pada hari kelima.

Pada hari keenam, yaitu tanggal 18 Mei 2026, Web Developer berhasil menyelesaikan halaman penjadwalan *grow light* (PB-SMAQ-12) dan halaman riwayat sensor (PB-SMAQ-13). Penyelesaian *backlog* tersebut memperluas fungsionalitas sistem dalam melakukan *monitoring* riwayat data sensor, pengelolaan jadwal *grow light*, serta pengendalian aktuator (*ON, OFF*).

Setelah itu, Web Developer merencanakan pengerjaan halaman navigasi responsif untuk perangkat mobile (PB-SMAQ-17), sistem penjadwalan *grow light* (PB-SMAQ-27), dan penyempurnaan sistem penjadwalan pemberian pakan (PB-SMAQ-26). Pada hari yang sama, IoT Engineer berhasil menyelesaikan tampilan halaman manajemen sensor (PB-SMAQ-7) dan sistem manajemen sensor (PB-SMAQ-24). Selanjutnya, IoT Engineer

melanjutkan pengerjaan tampilan monitoring lokasi untuk *admin* (PB-SMAQ-8), tampilan manajemen lokasi pengguna (PB-SMAQ-9), sistem manajemen lokasi (PB-SMAQ-22), serta penyempurnaan sistem manajemen sensor (PB-SMAQ-24).

Pada hari keenam ditemukan kendala pada halaman kontrol aktuator, yaitu mengenai urgensi dari adanya halaman *control actuator* sehingga halaman tersebut dipertimbangkan untuk disatukan dengan halaman lain. Masalah ini terjadi karena halaman tersebut terlihat kosong jika jumlah aktuator masih sedikit. Namun, setelah konsultasi pada *daily scrum*, *Scrum Master* menyarankan untuk tetap mempertahankan halaman tersebut sehingga halaman tersebut tetap menjadi satu halaman tunggal untuk mengontrol aktuator.

Pada hari ketujuh, yaitu tanggal 19 Mei 2026, Web Developer berhasil menyelesaikan sistem penjadwalan *grow light* (PB-SMAQ-27). Dengan selesainya *backlog* tersebut, sistem telah mampu mengelola jadwal otomatis pengoperasian *grow light* sesuai kebutuhan budidaya.

Pada hari yang sama, IoT Engineer berhasil menyelesaikan tampilan monitoring lokasi untuk *admin* (PB-SMAQ-8), tampilan manajemen lokasi pengguna (PB-SMAQ-9), sistem manajemen lokasi (PB-SMAQ-22), serta sistem manajemen sensor. Penyelesaian *backlog* tersebut menandai selesainya seluruh *backlog* yang direncanakan pada Sprint 1.

Menjelang berakhirnya sprint, kedua anggota tim mempersiapkan *Sprint review* 1, *sprint retrospective* 1 dan *sprint planning* 2 sebagai tahapan awal untuk melanjutkan pengerjaan sistem ke *sprint* berikutnya. Meskipun sebagian besar *backlog* berhasil diselesaikan sesuai rencana, Web Developer melaporkan bahwa tata letak beberapa halaman masih perlu dilakukan penyesuaian ulang agar konsistensi antarmuka pengguna dapat ditingkatkan pada sprint selanjutnya.

4.3.1.3 Daily Scrum 1

Kegiatan Daily Scrum dilaksanakan secara rutin setiap hari selama masa Sprint 1 untuk memantau perkembangan pengerjaan fitur serta melakukan sinkronisasi antar anggota tim. *Daily scrum* dilaksanakan dengan menggunakan aplikasi Whatsapp sesuai dengan persetujuan dari *product owner* sekaligus *scrum master*. Laporan hasil dari *daily scrum* disajikan dalam bentuk tabel log kegiatan harian. Tabel 4.24 menyajikan hasil progres harian

yang telah dicapai oleh *Web Developer* dan *IoT Engineer*, mencakup tugas yang diselesaikan, rencana pengerjaan berikutnya, serta identifikasi kendala teknis yang dihadapi selama pengembangan sistem berlangsung.

Tabel 4.24 *Daily Scrum 1 Log*

No	Hari Ke-n	Tanggal	Progress	Daffa Fairuz (Web dev)	Bima Aditya (IoT engineer)
1	1	13 Mei 2026	Progress Selesai	Sprint Planning 1	Sprint Planning 1
			Progress Selanjutnya	-	-
			Kendala (Solusi)	-	-
2	2	14 Mei 2026	Progress Selesai	PB-AKP-30, PB-AKP-21	PB-AKP-1, PB-AKP-2, PB-AKP-20
			Progress Selanjutnya	PB-AKP-14, PB-AKP-15	PB-AKP-3, PB-AKP-4
			Kendala (Solusi)	-	-
3	3	15 Mei 2026	Progress Selesai	PB-AKP-14, PB-AKP-15	PB-AKP-3, PB-AKP-4
			Progress Selanjutnya	PB-AKP-10, PB-AKP-11, PB-AKP-25, PB-AKP-26	PB-AKP-5, PB-AKP-6
			Kendala (Solusi)	-	-
4	4	16 Mei 2026	Progress Selesai	PB-AKP-10, PB-AKP-11, PB-AKP-25, PB-AKP-26	-
			Progress Selanjutnya	PB-AKP-19, PB-AKP-16	PB-AKP-5, PB-AKP-6, PB-AKP-23
			Kendala (Solusi)	-	
5	5	17 Mei 2026	Progress Selesai	PB-AKP-19, PB-AKP-16	PB-AKP-5, PB-AKP-6, PB-AKP-23
			Progress Selanjutnya	PB-AKP-12, PB-AKP-13	PB-AKP-7, PB-AKP-24, PB-AKP-18
			Kendala (Solusi)	-	-
6	6	18 Mei 2026	Progress Selesai	PB-AKP-12, PB-AKP-13, PB-AKP-28	PB-AKP-7, PB-AKP-24
			Progress Selanjutnya	PB-AKP-17, PB-AKP-27, PB-AKP-26	PB-AKP-8, PB-AKP-9, PB-AK-22

			Kendala (Solusi)	halaman control actuator masih memungkinkan jika dihapus	-
7	7	19 Mei 2026	Progress Selesai	PB-AKP-27, PB-AKP-19, PB-AKP-18	PB-AKP-8, PB-AKP-9, PB-AKP-22
			Progress Selanjutnya	Sprint Planning 2	Sprint Planning 2
			Kendala (Solusi)	tata letak beberapa halaman perlu diatur ulang	-

4.3.1.4 Sprint Review 1

Sprint Review 1 diselenggarakan pada tanggal 20 Mei 2026 bersama dosen pembimbing selaku *Product Owner*. Dalam pertemuan ini, tim mendemonstrasikan hasil dari pelaksanaan Scrum selama sprint 1 berjalan. fungsional antarmuka yang telah berhasil dikembangkan dalam lingkungan pengembangan (lokal). Fitur-fitur yang berhasil diverifikasi meliputi Halaman Login, Dashboard Monitoring (*Admin* dan *User*), manajemen perangkat (*Device* dan *Site*), halaman kontrol aktuator, penjadwalan pakan dan lampu (grow light), riwayat sensor, grafik pemantauan data *real-time*, hingga sistem notifikasi. Berdasarkan hasil demonstrasi, *Product Owner* memberikan beberapa poin umpan balik formal (*feedback*) untuk perbaikan fungsionalitas sistem pada tahapan berikutnya, seperti yang ditampilkan pada Tabel 4.25.

4.3.1.5 Sprint Retrospective 1

Sprint Retrospective 1 dilakukan untuk mengevaluasi pelaksanaan Sprint 1 dengan meninjau hasil Sprint Review 1 serta laporan Daily Scrum 1. Evaluasi ini bertujuan untuk mengidentifikasi aspek kolaborasi yang perlu ditingkatkan serta langkah-langkah strategis untuk pengembangan pada Sprint 2. Berdasarkan hasil evaluasi, tim menyepakati beberapa poin penting mengenai tindakan perbaikan dan masukan untuk sprint berikutnya sebagaimana disajikan dalam Tabel 4.25.

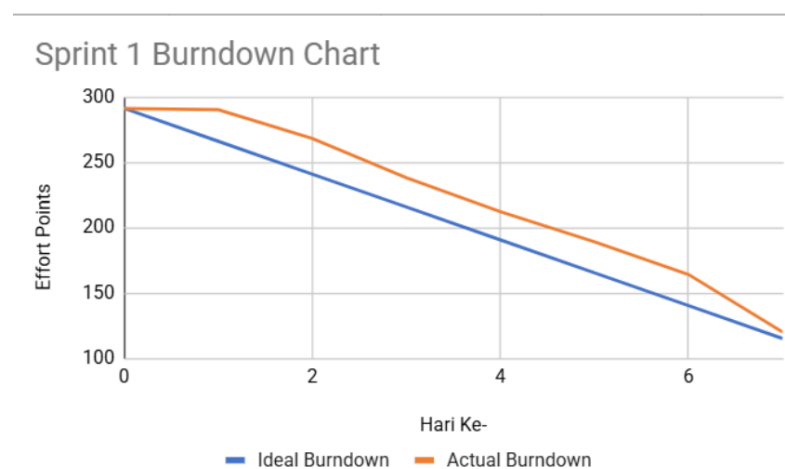
Tabel 4.25 Tabel Hasil *Sprint Review* dan *Sprint Retrospective*

No	Masukan Evaluasi	Tindakan	Masukan ke Sprint Berikutnya
1	Monitoring Data Sensor dan Actuator belum ditampilkan sesuai dengan <i>Device</i> terkait.	Melakukan perbaikan pada <i>query database</i> agar data sensor dan aktuator terfilter berdasarkan <i>Device ID</i> yang aktif.	-
2	Manajemen waktu pengerjaan Sprint Backlog.	Mengevaluasi beban kerja pada Sprint Backlog.	Jika <i>Sprint Backlog</i> dinilai terlalu padat, maka pengerjaan Sprint 2 perlu dilakukan dengan pertimbangan kapasitas yang lebih matang agar target tercapai.
3	Kurangnya notifikasi pada antarmuka <i>website</i> .	-	Menambahkan fitur notifikasi pada ikon notifikasi di bagian kanan atas <i>dashboard</i> .
4	Koordinasi Daily Scrum melalui WhatsApp Group.	-	Base WhatsApp Group harus lebih aktif dalam menjalankan daily scrum, setidaknya dengan adanya pertanyaan pemicu (trigger questions) agar diskusi lebih terarah.
5	User hanya bisa mengakses satu site.	-	Fitur User perlu ditingkatkan agar dapat mengelola lebih dari satu site.
6	Filtering data pada sensor dan aktuator.	-	Daftar Sensor dan Actuator perlu diberi fitur filtering berdasarkan <i>devices</i> / sites agar mempermudah monitoring.

4.3.1.6 Burndown Chart Sprint 1

Sprint 1 telah diselesaikan sesuai dengan rencana dan alur Scrum yang ditetapkan. Setelah itu dilakukanlah analisis dengan menggunakan *burndown chart* untuk menganalisis hasil implementasi *Sprint 1* untuk membandingkan kondisi ideal pengerjaan Scrum berdasarkan *effort point* dengan kondisi aktual Scrum yang dikerjakan. Hasil dari *Burndown chart* ditampilkan pada Gambar 4.23.

Burndown chart Sprint 1 menunjukkan perkembangan penyelesaian pekerjaan selama tujuh hari sprint dengan membandingkan garis *ideal burndown* dan *actual burndown*. Garis aktual berada di atas garis ideal hampir sepanjang sprint, yang mengindikasikan bahwa laju penyelesaian pekerjaan sedikit lebih lambat dari rencana awal. Namun demikian, jumlah effort points terus menurun secara konsisten dari 291 poin pada awal sprint hingga mencapai sekitar 120 poin pada akhir sprint, sehingga menunjukkan bahwa tim tetap mengalami kemajuan yang stabil dalam menyelesaikan backlog meskipun belum sepenuhnya sesuai dengan target ideal karena adanya backlog yang belum selesai dikerjakan pada sprint 1 sehingga menumpuk menjadi di *sprint backlog* pada *sprint 2*.



Gambar 4.23 *Burndown Chart Sprint 1*

4.3.2 Sprint 2

Sprint 2 dilaksanakan pada tanggal 20 Mei 2026 sampai dengan 26 Mei 2026. Sebelum sprint dimulai, tim pengembang melaksanakan Sprint Review 1 untuk mendemonstrasikan hasil pengembangan kepada Product Owner. Berdasarkan hasil evaluasi tersebut, Product Owner memberikan beberapa masukan terhadap fitur yang telah dikembangkan sehingga muncul sejumlah item product *backlog* baru yang ditandai dengan status revisi. Selain itu, masih terdapat beberapa *backlog* yang belum sempat diselesaikan pada Sprint 1.

4.3.2.1 Sprint Planning 2

Pada Sprint Planning 2, *sprint goal* yang ditentukan adalah revisi, dan integrasi *website* dengan sistem IoT dalam lingkungan pengembangan. *Product Owner* mengarahkan tim pengembang untuk memprioritaskan penyelesaian *backlog* revisi dan *backlog* yang belum

selesai pada *sprint* sebelumnya sebelum mengerjakan *backlog* baru. Keputusan tersebut diambil untuk memastikan bahwa seluruh fitur yang telah dikembangkan telah sesuai dengan kebutuhan pengguna dan dapat berfungsi secara optimal sebelum sistem dikembangkan lebih lanjut. Tabel 4.25 menampilkan *sprint backlog* 2 yang mencakup *backlog* tambahan berupa revisi dan *backlog* yang telah direncanakan sebelumnya.

Tabel 4.26 *Sprint Backlog* 2 Smartaquaponic

No	Kode Product Backlog Item	Deskripsi	Effort Point	Developer
1	PB-AKP-17	[FE] Layout - Responsive Mobile Navigation	5	Daffa
2	PB-AKP-52	[FE] Layout - Sensor History Page (Revisi)	2	Daffa
3	PB-AKP-53	[FE] Layout - Actuator Log Page (Revisi)	2	Daffa
4	PB-AKP-54	[BE] Sensor & Actuator History System (Revisi)	2	Daffa
5	PB-AKP-55	[FE] Layout - Notification Page (Revisi)	2	Daffa
6	PB-AKP-56	[BE] Site Management System (Revisi)	2	Daffa
7	PB-AKP-57	[FE] Layout - Sensor Management Page (Revisi)	2	Bima
8	PB-AKP-58	[FE] Layout - Actuator Management Page (Revisi)	2	Daffa
9	PB-AKP-59	[BE] Sensor Management System (Revisi)	2	Daffa
10	PB-AKP-60	[BE] Actuator Management System (Revisi)	2	Daffa
11	PB-AKP-29	[BE] MQTT for IoT Communication	6	Bima
12	PB-AKP-31	[BE] WhatsApp Notification System	8	Bima
13	PB-AKP-32	[BE] Email Notification System	6	Bima
14	PB-AKP-33	[Integrate] MQTT Broker Integration	3	Bima
15	PB-AKP-34	[Integrate] ESP32 Master Communication	2	Daffa
16	PB-AKP-35	[Integrate] ESP32 Slave Device Communication	2	Daffa
17	PB-AKP-36	[Integrate] Temperature Sensor Monitoring	3	Bima
18	PB-AKP-37	[Integrate] pH Sensor Monitoring	4	Bima
19	PB-AKP-38	[Integrate] Humidity Sensor Monitoring	3	Daffa
20	PB-AKP-39	[Integrate] Turbidity Sensor Monitoring	4	Bima
21	PB-AKP-40	[Integrate] Relay Actuator Control	4	Daffa
22	PB-AKP-41	[Integrate] Automatic Feeder Scheduling	8	Bima
23	PB-AKP-42	[Integrate] Automatic Lighting System	8	Daffa
24	PB-AKP-43	[Integrate] Realtime Monitoring System	5	Daffa

4.3.2.2 Sprint Implementation 2

Pada hari pertama, yaitu tanggal 20 Mei 2026, tim melaksanakan *Sprint Planning* 2. Pada kegiatan ini dilakukan pembahasan hasil *Sprint Review* 1, penentuan prioritas *backlog*, serta pembagian tugas kepada masing-masing anggota tim. Web Developer direncanakan

mengerjakan *responsive mobile navigation* (PB-SMAQ-17), perbaikan halaman riwayat sensor (PB-SMAQ-52), dan perbaikan halaman log aktuator (PB-SMAQ-53). Sementara itu, *IoT Engineer* direncanakan mengerjakan komunikasi MQTT untuk perangkat IoT (PB-SMAQ-29) dan sistem notifikasi email (PB-SMAQ-32). Karena hari pertama difokuskan pada perencanaan sprint, belum terdapat *backlog* yang diselesaikan maupun kendala yang ditemukan.

Pada hari kedua, yaitu tanggal 21 Mei 2026, *Web Developer* berhasil menyelesaikan fitur *responsive mobile navigation* (PB-SMAQ-17), revisi halaman riwayat sensor (PB-SMAQ-52), dan revisi halaman *log* aktuator (PB-SMAQ-53). Ketiga *backlog* tersebut merupakan prioritas awal sprint karena merupakan *backlog* yang belum selesai dan *backlog* hasil revisi dari Sprint Review 1. Penyelesaian *backlog* tersebut meningkatkan kenyamanan penggunaan sistem pada perangkat bergerak serta memperbaiki tampilan fitur monitoring yang sebelumnya memerlukan penyempurnaan.

Setelah *backlog* tersebut selesai, *Web Developer* melanjutkan pengerjaan integrasi *monitoring* sensor kelembapan (PB-SMAQ-38), revisi sistem riwayat sensor dan aktuator (PB-SMAQ-54), revisi halaman notifikasi (PB-SMAQ-55), dan revisi sistem manajemen lokasi (PB-SMAQ-56). Pada hari yang sama, *IoT Engineer* berhasil menyelesaikan komunikasi MQTT untuk perangkat IoT (PB-SMAQ-29) dan sistem notifikasi email (PB-SMAQ-32). Kedua *backlog* tersebut berperan penting dalam proses komunikasi data antara perangkat IoT dan sistem serta pengiriman informasi kepada pengguna melalui email. Selanjutnya, *IoT Engineer* merencanakan pengerjaan integrasi monitoring suhu (PB-SMAQ-36), integrasi monitoring pH (PB-SMAQ-37), dan integrasi monitoring kekeruhan air (PB-SMAQ-39). Tidak terdapat kendala yang ditemukan pada hari kedua.

Pada hari ketiga, yaitu tanggal 22 Mei 2026, *Web Developer* berhasil menyelesaikan integrasi monitoring sensor kelembapan (PB-SMAQ-38), revisi sistem riwayat sensor dan aktuator (PB-SMAQ-54), revisi halaman notifikasi (PB-SMAQ-55), dan revisi sistem manajemen lokasi (PB-SMAQ-56). Penyelesaian *backlog* revisi tersebut menunjukkan bahwa sebagian besar masukan yang diberikan Product Owner pada Sprint Review 1 telah berhasil diimplementasikan.

Selanjutnya, *Web Developer* melanjutkan pengerjaan revisi halaman manajemen aktuator (PB-SMAQ-58), revisi sistem manajemen sensor (PB-SMAQ-59), dan revisi sistem manajemen aktuator (PB-SMAQ-60). Pada hari yang sama, *IoT Engineer* berhasil menyelesaikan integrasi monitoring suhu (PB-SMAQ-36), integrasi monitoring pH (PB-SMAQ-37), dan integrasi monitoring kekeruhan air (PB-SMAQ-39). Setelah *backlog* tersebut selesai, *IoT Engineer* melanjutkan pengerjaan revisi halaman manajemen sensor (PB-SMAQ-57) dan melakukan penyempurnaan komunikasi MQTT (PB-SMAQ-29). Tidak ditemukan kendala yang menghambat proses pengembangan.

Pada hari keempat, yaitu tanggal 23 Mei 2026, *Web Developer* berhasil menyelesaikan revisi halaman manajemen aktuator (PB-SMAQ-58), revisi sistem manajemen sensor (PB-SMAQ-59), dan revisi sistem manajemen aktuator (PB-SMAQ-60). Dengan terselesaikannya *backlog* tersebut, seluruh *backlog* revisi utama hasil Sprint Review 1 telah berhasil diimplementasikan ke dalam sistem.

Setelah *backlog* revisi selesai, tim mulai mengerjakan *backlog* integrasi yang berkaitan dengan komunikasi antara sistem web dan perangkat IoT. *Web Developer* melanjutkan pengerjaan integrasi komunikasi ESP32 master (PB-SMAQ-34), integrasi komunikasi ESP32 slave (PB-SMAQ-35), dan sistem monitoring *real-time* (PB-SMAQ-43). Sementara itu, *IoT Engineer* berhasil menyelesaikan revisi halaman manajemen sensor (PB-SMAQ-57) dan komunikasi MQTT untuk perangkat IoT (PB-SMAQ-29). Selanjutnya, *IoT Engineer* melanjutkan pengerjaan sistem notifikasi email (PB-SMAQ-32) dan integrasi MQTT Broker (PB-SMAQ-33). Pada hari keempat tidak ditemukan kendala yang berarti.

Pada hari kelima, yaitu tanggal 24 Mei 2026, *Web Developer* berhasil menyelesaikan integrasi komunikasi ESP32 master (PB-SMAQ-34), integrasi komunikasi ESP32 slave (PB-SMAQ-35), dan sistem monitoring *real-time* (PB-SMAQ-43). Ketiga *backlog* tersebut merupakan bagian penting dari proses integrasi sistem karena memungkinkan pertukaran data antara aplikasi web dan perangkat IoT secara berkelanjutan.

Setelah *backlog* tersebut selesai, *Web Developer* melanjutkan pengerjaan integrasi kontrol aktuator menggunakan relay (PB-SMAQ-40). Pada hari yang sama, *IoT Engineer* berhasil menyelesaikan sistem notifikasi email (PB-SMAQ-32) dan integrasi MQTT Broker (PB-SMAQ-33). Selanjutnya, *IoT Engineer* mulai mengerjakan sistem notifikasi WhatsApp

(PB-SMAQ-31). Tidak terdapat kendala yang ditemukan selama proses pengembangan pada hari kelima.

Pada hari keenam, yaitu tanggal 25 Mei 2026, Web Developer berhasil menyelesaikan integrasi kontrol aktuator menggunakan relay (PB-SMAQ-40). Fitur tersebut memungkinkan sistem untuk mengendalikan aktuator yang terhubung dengan perangkat IoT secara langsung melalui aplikasi web. Setelah *backlog* tersebut selesai, Web Developer melanjutkan pengerjaan sistem pencahayaan otomatis atau automatic lighting system (PB-SMAQ-42).

Pada hari yang sama, *IoT Engineer* berhasil menyelesaikan sistem notifikasi WhatsApp (PB-SMAQ-31) dan melanjutkan pengerjaan sistem penjadwalan pemberian pakan otomatis (PB-SMAQ-41). Dalam proses pengembangan fitur notifikasi WhatsApp ditemukan kendala karena implementasi menggunakan layanan pihak ketiga yang tidak bersifat resmi sehingga terdapat beberapa keterbatasan penggunaan. Untuk mengatasi kendala tersebut, tim melakukan penyesuaian terhadap mekanisme pengiriman pesan dan memberikan batasan penggunaan agar fitur tetap dapat berjalan dengan stabil sesuai kebutuhan sistem.

Pada hari ketujuh, yaitu tanggal 26 Mei 2026, Web Developer berhasil menyelesaikan sistem pencahayaan otomatis atau automatic lighting system (PB-SMAQ-42). Fitur tersebut memungkinkan pengaturan grow light secara otomatis berdasarkan jadwal yang telah ditentukan sebelumnya. Pada hari yang sama, *IoT Engineer* berhasil menyelesaikan sistem penjadwalan pemberian pakan otomatis (PB-SMAQ-41) yang berfungsi untuk mengatur proses pemberian pakan ikan secara otomatis sesuai jadwal yang telah ditetapkan.

Dengan selesainya kedua *backlog* tersebut, seluruh *backlog* yang direncanakan pada Sprint 2 berhasil diselesaikan. Setelah seluruh pekerjaan selesai, tim mempersiapkan Sprint Planning 3 untuk menentukan *backlog* yang akan dikerjakan pada sprint berikutnya. Secara keseluruhan, Sprint 2 berhasil mencapai tujuan yang telah ditetapkan, yaitu menyelesaikan *backlog* revisi hasil Sprint Review 1, menyelesaikan *backlog* yang belum selesai pada Sprint 1, serta mengimplementasikan berbagai fitur integrasi antara sistem web dan perangkat IoT yang menjadi fondasi utama sistem Smart Aquaponic Monitoring and Control.

Meskipun Sprint 2 secara resmi berakhir pada tanggal 26 Mei 2026, kegiatan *Sprint Review 2*, *Sprint Retrospective 2*, dan *Sprint Planning 3* tidak dapat dilaksanakan sesuai jadwal yang telah direncanakan sebelumnya pada tanggal 27 Mei 2026. Kondisi tersebut terjadi karena Product Owner yang juga merangkap sebagai Scrum Master memiliki agenda dan kegiatan yang cukup padat sehingga tidak memungkinkan untuk mengikuti rangkaian kegiatan Scrum pada tanggal yang telah ditentukan.

Setelah dilakukan koordinasi dengan seluruh anggota tim, disepakati bahwa Sprint Review 2, Sprint Retrospective 2, dan Sprint Planning 3 akan dijadwalkan ulang pada tanggal 30 Mei 2026. Penyesuaian jadwal tersebut dilakukan agar seluruh pihak yang terlibat, khususnya Product Owner, dapat berpartisipasi secara penuh dalam proses evaluasi hasil sprint dan perencanaan sprint berikutnya.

Meskipun terjadi pergeseran jadwal kegiatan Scrum, hal tersebut tidak mempengaruhi pekerjaan pengembangan yang telah diselesaikan pada Sprint 2. Seluruh *backlog* yang direncanakan pada sprint tersebut tetap berhasil diselesaikan sesuai target. Oleh karena itu, perubahan jadwal hanya berdampak pada pelaksanaan kegiatan evaluasi dan perencanaan, bukan pada capaian pengembangan sistem yang telah ditetapkan pada Sprint 2.

4.3.2.3 Daily Scrum 2

Sama seperti *daily scrum 1*, Kegiatan *Daily Scrum 2* dilaksanakan secara rutin setiap hari selama masa Sprint 2 untuk memantau perkembangan pengerjaan fitur serta melakukan sinkronisasi antar anggota tim. *Daily scrum* dilaksanakan dengan menggunakan aplikasi whatsapp sesuai dengan persetujuan dari *product owner* sekaligus *scrum master*. Laporan hasil dari *daily scrum* disajikan dalam bentuk tabel log kegiatan harian. Tabel 4.27 menyajikan hasil progres harian yang telah dicapai oleh *Web Developer* dan *IoT Engineer*, mencakup tugas yang diselesaikan, rencana pengerjaan berikutnya, serta identifikasi kendala teknis yang dihadapi beserta solusi yang diterapkan selama pengembangan sistem berlangsung. Tabel 4.27 menyajikan rincian progres harian tersebut. Selama pelaksanaan Sprint 2, tim mengidentifikasi beberapa kendala teknis, terutama terkait dengan integrasi MQTT dan penggunaan layanan WhatsApp pihak ketiga. Pada hari kelima, ditemukan permasalahan stabilitas pada integrasi MQTT yang berdampak pada waktu pengerjaan, namun berhasil diatasi dengan optimasi konfigurasi. Selain itu, pada hari keenam,

penggunaan layanan notifikasi WhatsApp yang tidak resmi memerlukan penyesuaian mekanisme pengiriman pesan agar sistem tetap beroperasi secara stabil dengan batasan penggunaan tertentu.

Tabel 4.27 Tabel *Daily Scrum 2 Log*

No	Hari Ke-n	Tanggal	Progress	Daffa Fairuz (Web dev)	Bima Aditya (IoT engineer)
1	1	20 Mei 2026	Progress Selesai	Sprint Planning 2	Sprint Planning 2
			Progress Selanjutnya	PB-SMAQ-17, PB-SMAQ-52, PB-SMAQ-53	PB-SMAQ-29, PB-SMAQ-32
			Kendala	-	-
2	2	21 Mei 2026	Progress Selesai	PB-SMAQ-17, PB-SMAQ-52, PB-SMAQ-53	PB-SMAQ-29, PB-SMAQ-32
			Progress Selanjutnya	PB-SMAQ-38, PB-SMAQ-54, PB-SMAQ-55, PB-AKP56	PB-SMAQ-36, PB-SMAQ-37, PB-SMAQ-39
			Kendala	-	-
3	3	22 Mei 2026	Progress Selesai	PB-SMAQ-38, PB-SMAQ-54, PB-SMAQ-55, PB-SMAQ-56	PB-SMAQ-36, PB-SMAQ-37, PB-SMAQ-39
			Progress Selanjutnya	PB-SMAQ-58, PB-SMAQ-59, PB-SMAQ-60	PB-SMAQ-57, PB-SMAQ-29
			Kendala	-	-
4	4	23 Mei 2026	Progress Selesai	PB-SMAQ-58, PB-SMAQ-59, PB-SMAQ-60	PB-SMAQ-57, PB-SMAQ-29
			Progress Selanjutnya	PB-SMAQ-34, PB-SMAQ-35, PB-SMAQ-43	PB-SMAQ-32, PB-SMAQ-33
			Kendala	-	-
5	5	24 Mei 2026	Progress Selesai	PB-SMAQ-34, PB-SMAQ-35, PB-SMAQ-43	PB-SMAQ-32, PB-SMAQ-33
			Progress	PB-SMAQ-34, PB-	PB-SMAQ-31

No	Hari Ke-n	Tanggal	Progress	Daffa Fairuz (Web dev)	Bima Aditya (IoT engineer)
			Selanjutnya	SMAQ-35, PB-SMAQ-40	
			Kendala	-	masalah sering terjadi pada integrasi MQTT. namun, telah diselesaikan dengan waktu pengerjaan yang lebih lama
6	6	25 Mei 2026	Progress Selesai	PB-SMAQ-34, PB-SMAQ-35, PB-SMAQ-40	PB-SMAQ-31
			Progress Selanjutnya	PB-SMAQ-42	PB-SMAQ-41
			Kendala	-	Whatsapp menggunakan layanan tidak resmi sehingga perlu ada batasan
7	7	26 Mei 2026	Progress Selesai	PB-SMAQ-42	PB-SMAQ-41
			Progress Selanjutnya	Sprint Planning 3	Sprint Planning 3
			Kendala	-	-

4.3.2.4 Sprint Review 2

Sprint Review 2 dilaksanakan pada tanggal 30 Mei 2026 bersama *Product Owner* untuk mendemonstrasikan hasil pengembangan yang telah diselesaikan pada Sprint 2. Pada tahap ini, tim pengembang telah berhasil menampilkan *website* yang sudah dihosting sehingga dapat diakses secara online. Demonstrasi difokuskan pada fitur-fitur utama yang telah dikembangkan selama Sprint 2, namun belum melibatkan integrasi langsung dengan sistem akuaponik dan perangkat IoT. Pelaksanaan Sprint Review dilakukan secara singkat karena *Product Owner* memiliki keterbatasan waktu akibat agenda kegiatan yang cukup padat.

Selain mendemonstrasikan *website*, tim pengembang juga menyampaikan kendala pada fitur notifikasi WhatsApp yang masih menggunakan layanan pihak ketiga tidak resmi. Penggunaan metode tersebut berpotensi menimbulkan risiko terhadap nomor WhatsApp yang digunakan, seperti pembatasan atau penangguhan akun. Tim pengembang sempat mengusulkan untuk menghapus fitur tersebut, namun Product Owner memutuskan agar fitur notifikasi WhatsApp tetap dipertahankan karena dianggap memiliki manfaat yang tinggi bagi pengguna. Sebagai solusi, disepakati bahwa penggunaan fitur tersebut akan dibatasi untuk mengurangi risiko yang mungkin terjadi pada saat sistem dioperasikan.

4.3.2.5 *Sprint Retrospective 2*

Sprint Retrospective 2 dilaksanakan pada tanggal 30 Mei 2026 setelah Sprint Review 2 selesai dilakukan. Pada kegiatan ini, tim pengembang melakukan evaluasi terhadap proses pengembangan yang berlangsung selama Sprint 2. Secara umum, seluruh *backlog* yang direncanakan pada sprint ini berhasil diselesaikan sesuai target yang telah ditetapkan sehingga tidak terdapat kendala yang menyebabkan keterlambatan penyelesaian sprint.

Meskipun demikian, selama proses pengembangan ditemukan beberapa kendala teknis, seperti penyesuaian terhadap *backlog* hasil revisi dan implementasi fitur notifikasi WhatsApp yang masih menggunakan layanan pihak ketiga. Kendala tersebut dapat diselesaikan melalui diskusi dan koordinasi antar anggota tim sehingga tidak memberikan dampak signifikan terhadap jadwal pengembangan. Berdasarkan hasil evaluasi, tim sepakat untuk mempertahankan pola komunikasi dan koordinasi yang telah berjalan dengan baik serta meningkatkan perhatian terhadap risiko teknis yang berpotensi muncul pada proses integrasi sistem akuaponik pada sprint berikutnya.

4.3.2.6 *Burndown Chart Sprint 2*

Sprint 2 telah diselesaikan sesuai dengan rencana dan alur Scrum yang ditetapkan. Setelah itu dilakukan analisis menggunakan *burndown chart* untuk mengevaluasi hasil implementasi Sprint 2 dengan membandingkan kondisi ideal pengerjaan Scrum berdasarkan effort point dengan kondisi aktual pelaksanaan sprint. Hasil *burndown chart* ditampilkan pada Gambar 4.24.

Burndown chart Sprint 2 menunjukkan perkembangan penyelesaian pekerjaan selama tujuh hari sprint dengan membandingkan garis *ideal burndown* dan *actual burndown*. Pada awal sprint, garis aktual berada di atas garis ideal yang menunjukkan bahwa penyelesaian pekerjaan masih lebih lambat dari rencana. Namun, mulai pertengahan sprint hingga akhir sprint, laju penyelesaian backlog meningkat sehingga garis aktual mendekati bahkan berada sedikit di bawah garis ideal. Jumlah effort points berhasil berkurang secara konsisten dari 120 poin pada awal sprint hingga mencapai 30 poin pada akhir sprint. Hal ini menunjukkan adanya peningkatan kinerja tim dibandingkan Sprint 1, di mana sebagian *backlog* yang tertunda pada sprint sebelumnya berhasil diselesaikan dan proses pengerjaan mulai berjalan lebih sesuai dengan target yang telah direncanakan.



Gambar 4.24 *Burndown Chart Sprint 2*

4.3.3 Sprint 3

Sprint 3 dilaksanakan pada tanggal 30 Mei 2026 sampai dengan 5 Juni 2026. Meskipun siklus *sprint 3* ini diselesaikan pada tanggal 5 juni, kegiatan instalasi sistem di tempat *product owner* dilaksanakan pada tanggal 6 bersamaan dengan *sprint review 3* dan *sprint retrospective 3*. *Sprint goal* dari *sprint 3* ini adalah persiapan dan instalasi sistem. Pembahasan mengenai implementasi lebih lanjut mengenai bab ini disampaikan pada subbab-subbab berikutnya.

4.3.3.1 Sprint Planning 3

Sprint Planning 3 dilaksanakan pada tanggal 30 Mei 2026 setelah *Sprint Review 2* dan *Sprint Retrospective 2* selesai dilaksanakan. Pada pertemuan ini, *Product Owner* dan tim

pengembang membahas *backlog* yang akan dikerjakan pada Sprint 3 berdasarkan hasil evaluasi sprint sebelumnya. Berbeda dengan Sprint 1 dan Sprint 2 yang berfokus pada pengembangan dan integrasi fitur sistem, Sprint 3 difokuskan pada penyempurnaan sistem, pengujian menyeluruh dan persiapan pemasangan prototipe sistem akuaponik pada lokasi *product owner*.

Berdasarkan hasil diskusi, ditetapkan bahwa tujuan utama Sprint 3 adalah memastikan seluruh komponen sistem, baik *website* maupun perangkat IoT, dapat beroperasi dengan baik sebelum diimplementasikan pada lingkungan operasional. Oleh karena itu, *backlog* yang diprioritaskan pada sprint ini meliputi persiapan instalasi alat akuaponik dan Instalasi dari sistem itu sendiri. Selain itu, tim juga merencanakan demonstrasi sistem secara langsung kepada *Product Owner* setelah proses instalasi selesai dilakukan. Dengan seluruh *backlog* yang tersisa berfokus pada tahap perseiapan dan instalasi, Sprint 3 ditetapkan sebagai sprint terakhir dalam pengembangan sistem. Tabel 4.28 adalah hasil dari *sprint backlog* 3 yang menjadi *sprint* terakhir dalam pengembangan sistem Smartaquaponic.

Tabel 4.28 Tabel *Sprint Backlog* 3

No	Kode Product Backlog Item	Deskripsi	Effort Point	Developer
1	PB-AKP-61	Instalasi Alat	10	Daffa dan Bima
2	PB-AKP-44	[Finalization] Assembly of Casing and IoT Device Protectors	2	Daffa dan Bima
3	PB-AKP-45	[Finalization] File Migration to Shared Hosting	6	bima
4	PB-AKP-46	[Finalization] MQTT Configuration in Hosting Environment	2	bima
5	PB-AKP-47	[Finalization] Website Finalization in Hosting Environment	2	daffa dan bima
6	PB-AKP-48	[Finalization] Finalization of Cable Routing and Tool Electricals	4	daffa
7	PB-AKP-62	[Finalization] Data Migration	1	bima
8	PB-AKP-63	[Finalization] Confirmation for System Installation	2	daffa dan bima

4.3.3.2 Implementasi Sprint Planning 3

Sprint 3 dilaksanakan mulai tanggal 30 Mei 2026 sampai dengan 5 Juni 2026, dengan fokus utama pada migrasi ke shared hosting, finalisasi konfigurasi, serta persiapan dan pelaksanaan instalasi alat akuaponik di lokasi Product Owner. Berbeda dengan *sprint*

sebelumnya yang berfokus pada implementasi fitur-fitur baru, aktivitas pada siklus ini difokuskan pada tahap persiapan (*Finalization*) dan implementasi fisik di lapangan.

Tim melaksanakan kegiatan Sprint Planning 3 pada hari pertama keberjalanan *sprint* untuk menentukan ruang lingkup tugas teknis, menetapkan *Sprint Goal*, dan mendistribusikan beban kerja secara merata. Karena fokus hari pertama adalah koordinasi dan perencanaan, belum ada pekerjaan implementasi yang diselesaikan.

Pada Hari Kedua, *Web Developer* dan *IoT Engineer* berkolaborasi melakukan perakitan fisik kotak pelindung perangkat untuk melindungi komponen-komponen prototipe sistem akuaponik seperti ESP32 dan relay dari lingkungan akuaponik untuk meminimalisir terjadinya masalah yang disebabkan adanya masalah pada lingkungan seperti kebocoran dan lain-lain. Di samping itu, *IoT Engineer* memulai proses pemindahan berkas sistem aplikasi dari lingkungan pengembangan ke lingkungan *server shared hosting* agar dapat diakses secara online secara publik.

Setelah itu, *IoT Engineer* menyelesaikan migrasi berkas dan melanjutkan ke penataan jalur komunikasi MQTT pada server produksi pada hari ketiga. Proses ini krusial untuk menjamin data telemetri dari mikrokontroler dapat diteruskan ke basis data hosting secara stabil.

Pada hari ke empat (2 Juni 2026), *Web Developer* merapikan penataan kabel fisik dan aspek kelistrikan pada modul-modul prototipe sistem akuaponik guna meminimalkan risiko korsleting saat alat diletakkan di area basah. Setelah itu, tidak ada target progress lepasan individual yang diselesaikan pada hari ke lima karena kedua *developer* berfokus melakukan pengujian mandiri dan sinkronisasi awal performa website pasca-migrasi.

Pada tanggal 4 Juni 2026 yang merupakan hari ke enam, kedua anggota tim *developer* melakukan finalisasi *website* bersama di server produksi. Langkah ini mencakup pengujian fungsionalitas menyeluruh berdasarkan metode *Black Box Testing* untuk memastikan tidak ada error atau *bug* setelah kode program dipindahkan dari lingkungan lokal ke *shared hosting*.

Pada hari ketujuh, tim *developer* melakukan konfirmasi kesiapan lokasi bersama *product owner* dan *Scrum master* serta melakukan verifikasi kelengkapan komponen dan kesiapan alat untuk di pasang pada tempat milik *product owner*. Semua butir uji dalam

kriteria *Definition of Done* (DoD) dinyatakan terpenuhi baik pada persiapan ini maupun saat *sprint review* sehingga prototipe siap dipasang.

Sesuai kesepakatan bersama *Product Owner*, tim pengembang mendatangi lokasi untuk melakukan pemasangan fisik seluruh komponen pada hari ke delapan. Satu unit ESP32 *master (site)* dan dua unit ESP32 *slave (device)* diletakkan pada rak besi susun akuaponik vertikal dan diintegrasikan dengan sensor-sensor kualitas air dan udara. Kegiatan ini ditutup dengan pelaksanaan *Sprint Review 3* dan *Sprint Retrospective 3* secara langsung bersama *Product Owner*.

Meskipun kegiatan instalasi telah selesai dilakukan, *developer* masih siap untuk melakukan perbaikan sistem ketika terjadi masalah pada sistem akuaponik. Namun, pembahasan pada laporan ini hanya berfokus pada pelaksanaan dan implementasi Scrum sehingga kegiatan perbaikan tidak dibahas lengkap pada subbab ini.

4.3.3.3 Daily Scrum 3

Sama seperti daily scrum 1 dan 2, kegiatan *daily scrum 3* juga dilakukan dengan menggunakan aplikasi Whatsapp setelah melakukan implementasi harian. Bagian yang membedakan *sprint* ini dengan *sprint 1* dan 2 adalah pada bagian durasinya (*effort point*) yang tidak banyak dan waktu *sprint* yang dimulai lebih telat. Pelaksanaan *Sprint 3* berlangsung tanpa ada kendala yang mengakibatkan telat atau hal lainnya. Hasil dari *log daily scrum 3* ada pada Tabel 4.29.

Tabel 4.29 *Daily Scrum Sprint 3 Log*

No	Hari Ke-n	Tanggal	Progress	Daffa Fairuz (Web Dev)	Bima Aditya (IoT engineer)
1	1	30 Mei 2026	Progress Selesai	Sprint Planning 3	Sprint Planning 3
			Progress Selanjutnya	PB-AKP-44	PB-AKP-45
			Kendala	-	-
2	2	31 Mei 2026	Progress Selesai	PB-AKP-44	PB-AKP-45
			Progress Selanjutnya	-	PB-AKP-46
			Kendala	-	-
3	3	1 Juni 2026	Progress Selesai	-	PB-AKP-46
			Progress Selanjutnya	PB-AKP-48	-
			Kendala	-	-

No	Hari Ke-n	Tanggal	Progress	Daffa Fairuz (Web Dev)	Bima Aditya (IoT engineer)
4	4	2 Juni 2026	Progress Selesai	PB-AKP-48	-
			Progress Selanjutnya	-	-
			Kendala	-	-
5	5	3 Juni 2026	Progress Selesai	-	-
			Progress Selanjutnya	PB-AKP-47	PB-AKP-47
			Kendala	-	-
6	6	4 Juni 2026	Progress Selesai	PB-AKP-47	PB-AKP-47
			Progress Selanjutnya	PB-AKP-63	PB-AKP-63
			Kendala	-	-
7	7	5 Juni 2026	Progress Selesai	PB-AKP-63	PB-AKP-63
			Progress Selanjutnya	PB-AKP-61	PB-AKP-61
			Kendala	-	-
8	8	6 Juni 2026	Progress Selesai	PB-AKP-61	PB-AKP-61
			Progress Selanjutnya	-	-
			Kendala	-	-

4.3.3.4 Sprint Review 3

Sprint Review 3 dilaksanakan pada tanggal 6 Juni 2026 setelah proses instalasi alat akuaponik selesai dilakukan di lokasi *product owner*. Pada kegiatan ini, tim pengembang mendemonstrasikan sistem yang telah terintegrasi secara langsung, mulai dari *website*, MQTT Broker, ESP32 master, ESP32 slave, sensor, hingga aktuator yang digunakan pada sistem akuaponik. Demonstrasi difokuskan pada penggunaan sistem dari sisi pengguna, khususnya bagaimana pengguna dapat memantau data sensor secara *real-time*, mengelola jadwal perangkat, serta mengendalikan aktuator melalui *website*.

Selain demonstrasi *website*, tim juga menunjukkan hasil pemasangan perangkat yang terdiri dari satu ESP32 master dan dua ESP32 *slave* yang terhubung dengan berbagai sensor dan aktuator. *Product Owner* melakukan pengujian secara langsung terhadap fitur monitoring dan pengendalian perangkat yang telah tersedia pada sistem. Berdasarkan hasil demonstrasi tersebut, *Product Owner* menyatakan bahwa sistem telah memenuhi kebutuhan yang ditetapkan pada *product backlog* dan menerima hasil pengembangan yang telah

dilakukan. Meskipun demikian, Product Owner tetap menyampaikan beberapa temuan masalah teknis yang muncul setelah sistem mulai digunakan pada lingkungan operasional sebagai bahan evaluasi untuk penyempurnaan dan perbaikan sistem.

4.3.3.5 *Sprint Retrospective 3*

Sprint Retrospective 3 dilaksanakan pada tanggal 6 Juni 2026 setelah *Sprint Review 3* selesai dilakukan. Pada kegiatan ini, tim pengembang melakukan evaluasi terhadap proses pengembangan yang telah berlangsung selama tiga sprint. Hasil evaluasi menunjukkan bahwa seluruh *backlog* yang direncanakan berhasil diselesaikan sesuai tepat waktu bahkan lebih cepat dan tujuan sprint dapat tercapai dengan baik. Selain itu, komunikasi dan koordinasi antar anggota tim berjalan dengan lancar sehingga tidak terdapat kendala yang berdampak pada penyelesaian pekerjaan selama *Sprint 3*.

Penerapan metode Scrum pada pengembangan sistem dianggap berhasil karena setiap sprint goal dan item *backlog* berhasil tercapai dan diselesaikan pada waktunya. Melalui tiga sprint yang telah dilaksanakan, seluruh product *backlog* berhasil diselesaikan sehingga tidak diperlukan sprint lanjutan. Oleh karena itu, pengembangan sistem dinyatakan selesai pada *Sprint 3* dan dilanjutkan pada tahap implementasi, evaluasi, dan *maintenance* terhadap sistem pada lingkungan operasional yang sebenarnya.

4.3.3.6 *Burndown Chart Sprint 3*

Sprint 3 telah diselesaikan sesuai dengan rencana dan alur Scrum yang ditetapkan. Setelah itu dilakukan analisis menggunakan burndown chart untuk mengevaluasi hasil implementasi *Sprint 3* dengan membandingkan kondisi ideal pengerjaan Scrum berdasarkan effort point dengan kondisi aktual pelaksanaan sprint. Hasil burndown chart ditampilkan pada Gambar 4.25.

Burndown chart Sprint 3 menunjukkan perkembangan penyelesaian pekerjaan selama delapan hari sprint dengan membandingkan garis *ideal burndown* dan *actual burndown*. Pada awal *sprint*, garis aktual berada sedikit di atas garis ideal, kemudian mengalami penurunan yang cukup signifikan hingga mendekati garis ideal pada pertengahan *sprint*. Meskipun terdapat periode stagnasi pada hari keempat hingga hari kelima yang menunjukkan tidak adanya pengurangan *effort points*, tim mampu kembali meningkatkan

laju penyelesaian pekerjaan pada akhir sprint. Jumlah effort points berhasil berkurang secara bertahap dari 30 poin pada awal sprint hingga mencapai 0 poin pada akhir sprint. Kondisi ini menunjukkan bahwa seluruh backlog yang tersisa berhasil diselesaikan sesuai target sprint, sehingga Sprint 3 dapat ditutup dengan penyelesaian backlog secara keseluruhan dan tingkat pencapaian yang sesuai dengan perencanaan Scrum yang telah ditetapkan.



Gambar 4.25 *Burndown Chart Sprint 3*

4.4 Pembahasan dalam Implementasi

Subbab 4.4 ini menyajikan pembahasan mendalam mengenai aspek-aspek teknis dalam implementasi prototipe sistem akuaponik yang dieksekusi sepanjang siklus metodologi Scrum.

4.4.1 Implementasi Basis Data

Basis data yang digunakan dalam implementasi basis data adalah Mysql. Berikut ini adalah tabel hasil implementasi basis data yang dibuat berdasarkan rancangan basis data pada subbab 4.2.6.

1. Implementasi Tabel *User*

Implementasi Tabel *User* dapat dilihat pada Tabel 4.30.

Tabel 4.30 Implementasi Tabel *User*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID unik pengguna
name	VARCHAR(50)	NOT NULL	Nama lengkap pengguna

Kolom	Tipe	Keterangan	Deskripsi
email	VARCHAR(50)	NOT NULL, UNIQUE	Alamat email pengguna
email_verified_at	TIMESTAMP	NULLABLE	Waktu verifikasi email
password	VARCHAR(255)	NOT NULL	Hash <i>password</i> pengguna
phone_number	VARCHAR(14)	NULLABLE	Normor <i>Handphone</i> pengguna
whatsapp_notification	BOOLEAN	Default true	Status notifikasi WhatsApp
email_notification	BOOLEAN	Default true	Status notifikasi email
role	ENUM('admin','user')	Default 'user'	Peran pengguna dalam sistem
status	ENUM('active','inactive')	Default 'active'	Status akun pengguna
remember_token	VARCHAR(255)	NULLABLE	Token autentikasi sesi
created_at	TIMESTAMP	NOT NULL	Waktu pembuatan akun
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir akun diperbarui

2. Implementasi Tabel *Site*

Implementasi Tabel *Site* dapat dilihat pada Tabel 4.31.

Tabel 4.31 Implementasi Tabel *Site*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID unik lokasi SmartAquaponic
user_id	BIGINT	FK → users.id	Pemilik lokasi
name	VARCHAR(255)	NOT NULL	Nama sites
location	VARCHAR(255)	NOT NULL	Nama lokasi
description	TEXT	NULLABLE	Deskripsi lokasi
mac_address	VARCHAR	NOT NULL, UNIQUE	MAC Address ESP32 Master
created_at	TIMESTAMP	NOT NULL	Waktu pembuatan data
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

3. Implementasi Tabel *Site-devices*

Implementasi tabel *site-devices* dapat dilihat pada Tabel 4.32.

Tabel 4.32 Implementasi Tabel *Site-devices*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID relasi perangkat
site_id	BIGINT	FK → sites.id	Relasi ke lokasi

Kolom	Tipe	Keterangan	Deskripsi
device_id	BIGINT	FK → devices.id	Relasi ke perangkat slave
started_at	DATETIME	NULLABLE	Waktu perangkat mulai terhubung
ended_at	DATETIME	NULLABLE	Waktu perangkat dilepas dari lokasi
created_at	TIMESTAMP	NOT NULL	Waktu pembuatan data
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

4. Implementasi Tabel *Devices*

Implementasi tabel *devices* dapat dilihat pada Tabel 4.33.

Tabel 4.33 Implementasi Tabel *Devices*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID unik perangkat
mac_address	VARCHAR(18)	NOT NULL, UNIQUE	Alamat MAC perangkat ESP32
name	VARCHAR(255)	NOT NULL	Nama perangkat
description	VARCHAR(255)	NULLABLE	Deskripsi perangkat
status	ENUM	available, assigned, inactive	Status perangkat
created_at	TIMESTAMP	NOT NULL	Waktu pembuatan data
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

5. Implementasi Tabel Sensor

Implementasi tabel sensor dapat dilihat pada Tabel 4.34.

Tabel 4.34 Implementasi Tabel Sensor

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID unik sensor
device_id	BIGINT	FK → devices.id	Relasi ke perangkat
name	VARCHAR(50)	NOT NULL	Nama sensor
type	VARCHAR(20)	NOT NULL	Jenis sensor
unit	VARCHAR(10)	NOT NULL	Jenis unit atau satuan sensor
min_threshold	DECIMAL(8,2)	NULLABLE	Nilai ambang minimum
max_threshold	DECIMAL(8,2)	NULLABLE	Nilai ambang maksimum
created_at	TIMESTAMP	NOT NULL	Waktu pembuatan data
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

6. Implementasi Tabel *Data_sensor*

Implementasi tabel *data_sensor* dapat dilihat pada Tabel 4.35.

Tabel 4.35 Implementasi Tabel *Data_sensor*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID data sensor
sensor_id	BIGINT	FK → sensors.id	Relasi ke sensor
value	DECIMAL	NOT NULL	Nilai hasil pembacaan sensor
created_at	TIMESTAMP	NOT NULL	Waktu pencatatan data
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

7. Implementasi Tabel *Actuator*

Implementasi tabel *actuator* dapat dilihat pada Tabel 4.36.

Tabel 4.36 Implementasi Tabel *Actuator*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID aktuator
device_id	BIGINT	FK → devices.id	Relasi ke perangkat
name	VARCHAR(50)	NOT NULL	Nama aktuator
type	VARCHAR(20)	NOT NULL	Jenis aktuator
status	ENUM('on','off')	Default 'off'	Status aktuator
created_at	TIMESTAMP	NOT NULL	Waktu pembuatan data
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

8. Implementasi Tabel *Actuator_Logs*

Implementasi tabel *actuator_logs* dapat dilihat pada Tabel 4.37.

Tabel 4.37 Implementasi Tabel *Actuator_logs*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID log aktuator
actuator_id	BIGINT	FK → actuators.id	Relasi ke aktuator
action	enum('on', 'off')	NOT NULL	Aksi yang dilakukan
Triggered_by	enum('manual', 'auto')	NOT NULL	Sumber aksi
created_at	TIMESTAMP	NOT NULL	Waktu pencatatan log
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

9. Implementasi Tabel *Feed_schedules*

Implementasi tabel *feed_schedules* dapat dilihat pada Tabel 4.38.

Tabel 4.38 Implementasi Tabel *Feed_schedules*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID jadwal pakan
actuator_id	BIGINT	FK → actuators.id	Relasi ke aktuator feeder
time	TIME	NOT NULL	Waktu pemberian pakan
duration	INT	NOT NULL	Durasi pemberian pakan
created_at	TIMESTAMP	NOT NULL	Waktu pembuatan data
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

10. Implementasi Tabel *Grow_Light_Schedules*

Implementasi tabel *grow_light_schedules* dapat dilihat pada Tabel 4.39.

Tabel 4.39 Implementasi Tabel *Grow_light_schedules*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID jadwal lampu
actuator_id	BIGINT	FK → actuators.id	Relasi ke grow light
start time	TIME	NOT NULL	Waktu mulai menyala
end time	TIME	NOT NULL	Waktu selesai menyala
created_at	TIMESTAMP	NOT NULL	Waktu pembuatan data
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

11. Implementasi Tabel *Notifications*

Implementasi tabel *notifications* dapat dilihat pada Tabel 4.40.

Tabel 4.40 Implementasi Tabel *Notifications*

Kolom	Tipe	Keterangan	Deskripsi
id	BIGINT	PK, NOT NULL	ID notifikasi
site_id	BIGINT	FK → site.id	Site penerima notifikasi
message	TEXT	NOT NULL	Isi notifikasi
type	enum('info', 'warning', 'alert')	NOT NULL	Jenis notifikasi
is_read	BOOLEAN	Default false	Status telah dibaca
created_at	TIMESTAMP	NOT NULL	Waktu notifikasi dibuat
updated_at	TIMESTAMP	NULLABLE	Waktu terakhir diperbarui

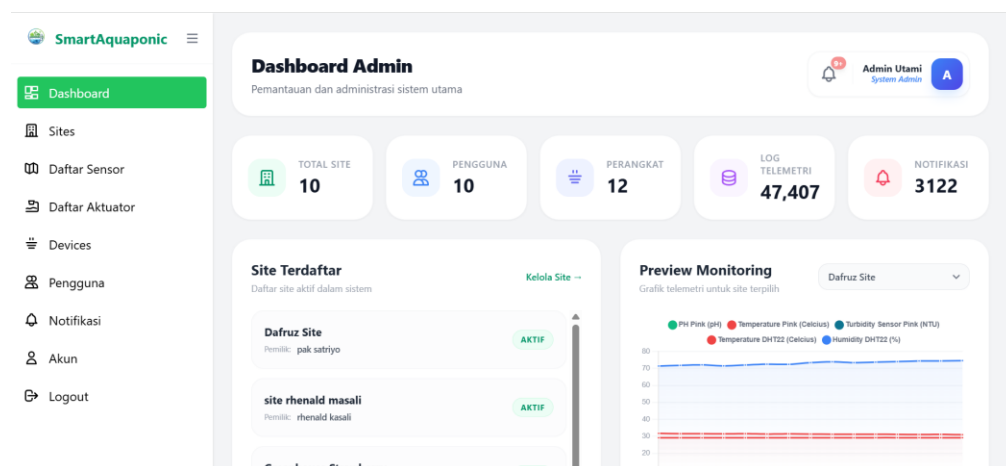
4.4.2 Implementasi Dashboard Admin dan User

Dalam pengembangan prototipe sistem akuaponik, kedua aktor yaitu *user* dan *admin* memiliki hak akses yang berbeda-beda. dalam implementasi *dashboard*, kedua aktor tersebut juga memiliki dua *dashboard* yang berbeda.

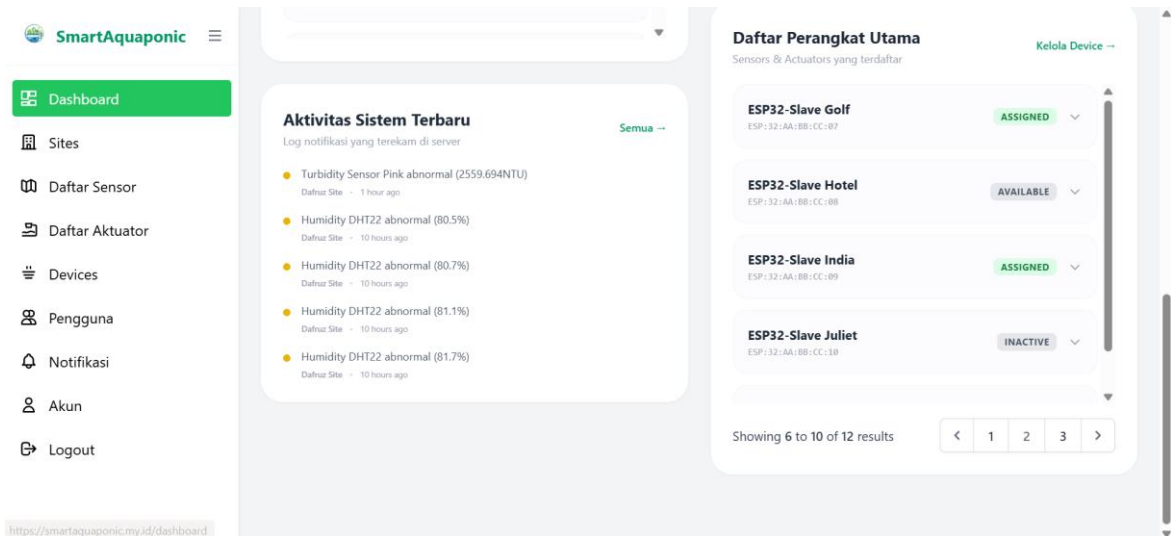
Dashboard yang dibuat untuk aktor *admin* dibuat untuk mempermudah aktor menjaga keseluruhan sistem agar stabil termasuk mengawasi *site-site* milik *user*. *Dashboard* milik *admin* memiliki informasi mengenai jumlah *site*, jumlah pengguna, jumlah perangkat (*device*), jumlah *log* telemetri dan jumlah notifikasi. Dengan ini, *admin* dapat melakukan pemeliharaan sistem secara umum baik untuk penambahan ruang maupun untuk keperluan pemeliharaan server lainnya.

Selain itu, *dashboard admin* juga berisi informasi-informasi yang lebih singkat mengenai status *site*, aktivitas terbaru berupa notifikasi, *preview* monitoring dengan *site* pilihan, dan status *device-device* yang ada. Penambahan informasi ini digunakan agar *admin* lebih cepat dalam melakukan tindakan-tindakan terkait data-data yang tersimpan dalam basis data. Gambar 4.26 dan Gambar 4.27 adalah implementasi dari halaman *dashboard* untuk aktor *admin*.

Berbeda dengan aktor *admin*, *Dashboard* yang dibuat untuk aktor *user* dibuat untuk mempermudah aktor menjaga *site* miliknya sendiri. *User* tidak memiliki tanggung jawab terhadap sistem secara keseluruhan selain hal-hal yang berhubungan dengan *site* miliknya sendiri. Oleh karena itu, *dashboard* milik *user* hanya mencakup data milik pengguna tersebut saja.

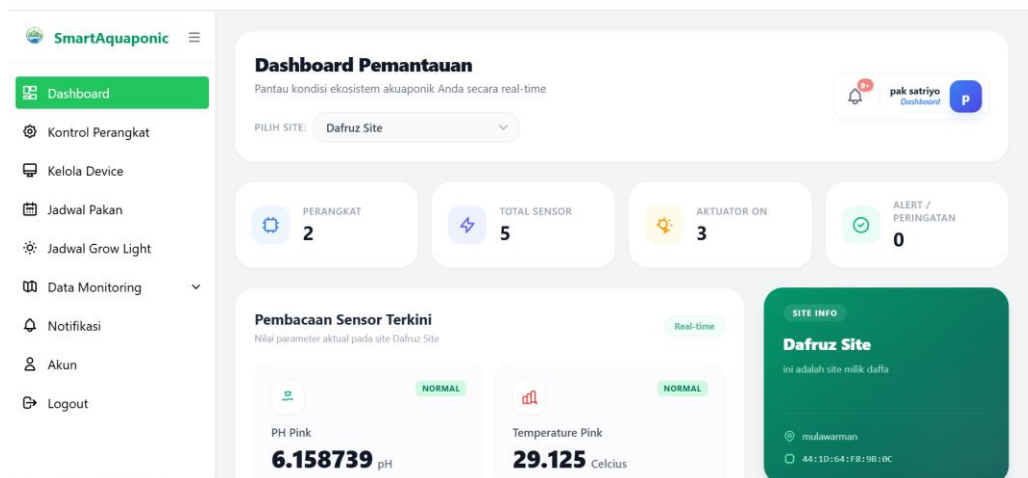


Gambar 4.26 Implementasi Halaman *Dashboard* untuk Admin Bagian Atas

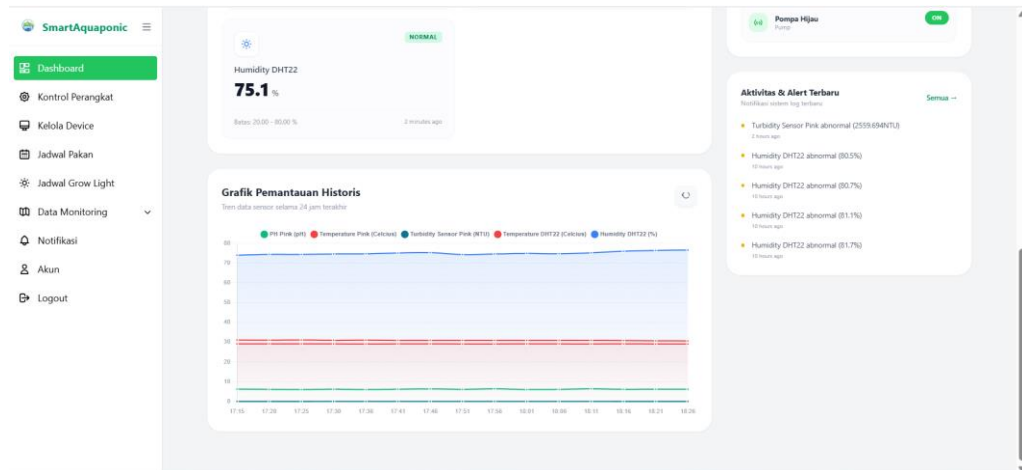


Gambar 4.27 Implementasi Halaman *Dashboard* untuk *Admin* Bagian Bawah

Dashboard untuk aktor *user* memiliki informasi terkait hal-hal yang berhubungan dengan *site* yang terhubung dengan akun *user* seperti jumlah perangkat (*device*) yang terhubung, jumlah total sensor, jumlah aktuator yang sedang dalam kondisi menyala (*on*), peringatan terbaru. Selain itu, halaman ini juga berisi data terbaru dari data setiap sensor yang terpasang pada *site* saat ini dan dilengkapi dengan grafik historis dari data-data yang diperoleh dari setiap sensor dan aktifitas sensor yang terhubung. Bagian terakhir pada halaman ini adalah halaman mengenai *alert* terbaru yang muncul dalam beberapa waktu. Gambar 4.28 dan Gambar 4.29 adalah gambar dari implementasi halaman *dashboard* milik aktor *user*.



Gambar 4.28 Implementasi Halaman *Dashboard* untuk *User* Bagian Atas



Gambar 4.29 Implementasi Halaman *Dashboard* untuk *User* Bagian Bawah

4.4.3 Implementasi Pencegahan *Overlapping* pada Jadwal *Feeder* dan *Grow Light*

Pada pembuatan Jadwal *grow light* dan *feeder*, *Scrum master* memberikan masukan terkait penerapan pembatasan pada pembuatan jadwal untuk *feeder* dan *grow light*. Hal ini bertujuan agar tidak adanya konflik yang terjadi antar *instance* dalam basis data jadwal.

4.4.3.1 Pencegahan *Overlapping* pada Jadwal *Feeder*

Pembuatan jadwal *feeder* menggunakan parameter waktu mulai (*time*) dan durasi aktif (*duration*) dalam menit. Karakteristik ini menimbulkan tantangan teknis ketika durasi aktif aktuator melintasi batas tengah malam (melewati menit ke-1440). Untuk mengatasinya, *FeedScheduleController* menerapkan metode Pemisahan Interval Waktu (*Time Interval Splitting*) melalui fungsi *getFeedIntervals()*. Jika waktu selesai (waktu_mulai + durasi) melebihi 1440 menit, sistem secara otomatis memecah jadwal tersebut menjadi dua bagian *array* multi-dimensi, yaitu segmen hari ini ([start, 1440]) dan segmen hari berikutnya ([0, sisa_minit]). Jika tidak melewati tengah malam, jadwal tetap disimpan dalam satu bagian normal.

Proses deteksi *overlapping* dilakukan oleh fungsi *checkFeedOverlap()* dengan membandingkan irisan seluruh segmen dari jadwal lama dalam basis data dan jadwal baru atau diubah menggunakan *nested loop*. Dua segmen waktu dinyatakan bertabrakan (*overlap*) jika memenuhi kondisi “(start1 < end2) and (end1 > start2)”. Jika salah satu kombinasi segmen terbukti beririsan, fungsi langsung mengembalikan nilai *true*, sehingga sistem akan

membatalkan manipulasi basis data dan mengirimkan pesan *error* validasi kepada pengguna. Implementasi penanganan *overlapping* ini disajikan pada Kode 4.1.

Source Code 4.1:
Kode fungsi untuk pengecekan overlap pada controller feeder schedule

```
private function checkFeedOverlap($time1, $duration1, $time2,
    $duration2)
{
    $t1 = $this->timeToMinutes($time1);
    $t2 = $this->timeToMinutes($time2);

    $intervals1 = $this->getFeedIntervals($t1, $duration1);
    $intervals2 = $this->getFeedIntervals($t2, $duration2);

    foreach ($intervals1 as $int1) {
        foreach ($intervals2 as $int2) {
            if ($int1[0] < $int2[1] && $int1[1] > $int2[0]) {
                return true;
            }
        }
    }
    return false;
}

private function getFeedIntervals($start, $duration)
{
    $end = $start + $duration;
    if ($end > 1440) {
        return [
            [$start, 1440],
            [0, $end - 1440]
        ];
    }
    return [
        [$start, $end]
    ];
}
```

4.4.3.2 Pencegahan *Overlapping* pada Jadwal *Grow Light*

Mekanisme pencegahan overlapping pada komponen lampu pertumbuhan (*grow light*) didasarkan pada perbandingan dua interval waktu diskret. Setiap kali pengguna menambahkan (*store*) atau memperbarui (*update*) jadwal, sistem melakukan konversi string format waktu (H:i) ke dalam satuan menit harian (0 sampai 1440 menit) melalui fungsi *timeToMinutes()*. Logika untuk mendeteksi *overlapping* antara Jadwal A (*s1, e1*) dan Jadwal

B (s_2, e_2) diekspresikan sebagai $\text{Overlap} = (s_1 < e_2) \text{ and } (e_1 > s_2)$ dimana s_1 = waktu mulai jadwal yang tersimpan di basis data, e_1 = waktu selesai jadwal s_1 , s_2 = waktu mulai jadwal yang baru/diubah, dan e_2 = waktu selesai jadwal s_2 .

Source Code 4.2:

Kode fungsi untuk pengecekan overlap pada controller grow light schedule

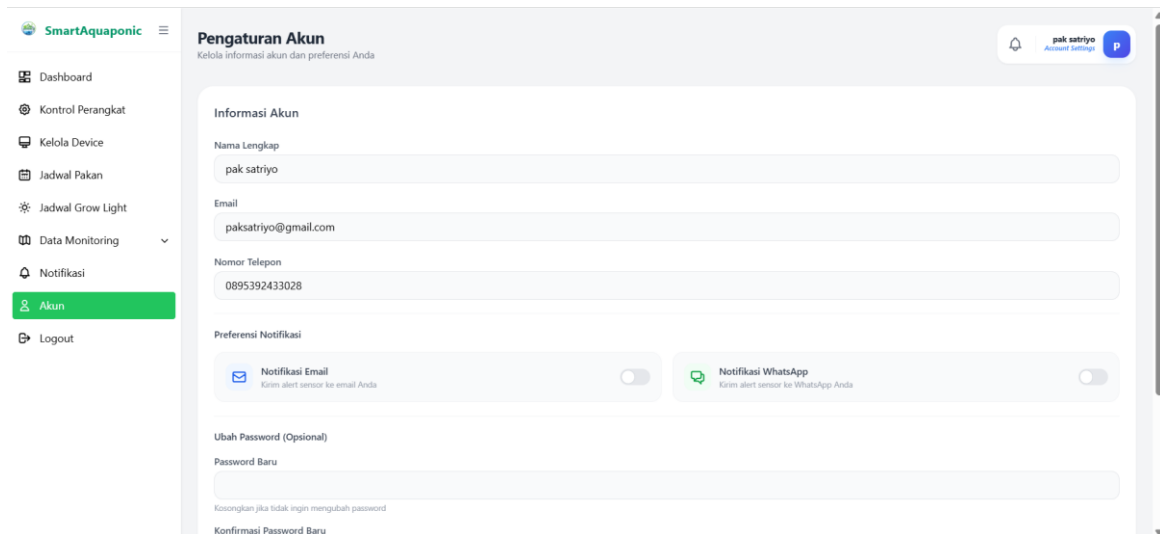
```
private function checkGrowLightOverlap($start1, $end1, $start2, $end2)
{
    $s1 = $this->timeToMinutes($start1);
    $e1 = $this->timeToMinutes($end1);
    $s2 = $this->timeToMinutes($start2);
    $e2 = $this->timeToMinutes($end2);
    return ($s1 < $e2 && $e1 > $s2);
}
```

Fungsi ini dipanggil dalam fungsi *store()* dan *update()* dengan menggunakan *loop* untuk memanggil semua data yang tersimpan dalam basis data sebagai data untuk s_1 dan e_1 . Fungsi ini mencegah sebuah *instance* yang tumpang-tibndih dengan *instance* jadwal lainnya untuk disimpan dalam basis data.

4.4.4 Implementasi Pengaturan Akun

Proses pengelolaan profil dan konfigurasi pengguna dilakukan melalui komponen pengatur akun yang terintegrasi dengan sistem keamanan aplikasi. Pada tahap awal, sistem akan mengambil data profil milik pengguna yang sedang masuk atau terautentikasi saat itu. Informasi profil tersebut kemudian dikirimkan dan ditampilkan pada halaman pengaturan akun. Dengan ini, sistem menjamin bahwa data yang muncul pada antarmuka bersifat pribadi dan hanya dapat diakses oleh pemilik akun yang sah. Gambar 4.30 adalah tampilan dari halaman notifikasi pada *website*.

Halaman ini juga memfasilitasi pengguna untuk memperbarui data diri serta mengatur preferensi notifikasi. Sebelum data baru disimpan, sistem akan memeriksa seluruh masukan untuk memastikan format data sudah benar, seperti keunikan alamat email agar tidak kembar dengan pengguna lain, serta kesesuaian kata sandi baru. Jika semua data telah dinyatakan valid, sistem akan memperbarui informasi nama, email, nomor telepon, serta pilihan aktif-tidaknya notifikasi WhatsApp dan email. Pengaturan notifikasi yang tersedia pada pengaturan akun hanya untuk perubahan basis data saja. Bagian lain yang mengurus agar notifikasi diterukan ke whatsapp atau email dilakukan oleh *controller* notifikasi.



Gambar 4.30 Implementasi Halaman Pengaturan Akun

Dalam pengaturan akun, sistem juga menyediakan fitur untuk menonaktifkan akun. Untuk menjaga keamanan dan riwayat data, proses ini tidak menghapus data pengguna secara permanen dari basis data, melainkan hanya mengubah status akun menjadi tidak aktif. Setelah status akun berubah, sistem secara otomatis akan mengeluarkan pengguna dari aplikasi, menghapus seluruh sesi masuk yang sedang berjalan, dan mengamankan kembali sistem dari akses ilegal. Pengguna kemudian akan dialihkan kembali ke halaman utama disertai dengan pesan pemberitahuan.

4.4.5 Implementasi Notifikasi

Dalam pembuatan halaman notifikasi, pembuatan halaman disesuaikan dengan aktor yang berkaitan. Halaman aktor *admin* dapat melihat seluruh notifikasi yang ada dikirim untuk semua *user* yang ada. Sementara itu, aktor *user* hanya menampilkan seluruh notifikasi yang berkaitan dengan dirinya saja. Implementasi halaman aktor *admin* dan *user* dapat dilihat pada Gambar

Gambar 4.31 dan Gambar 4.32 memperlihatkan bahwa halaman notifikasi untuk aktor *admin* memiliki satu tambahan kolom yaitu kolom pengirim yang menunjukkan bahwa notifikasi tersebut muncul dari *user* mana. Pemanggilan data juga dibedakan dimana pada halaman *user* hanya memanggil data notifikasi sesuai dengan *site* yang terhubung dengannya sementara *admin* memanggil seluruh data notifikasi.

#	WAKTU	PESAN	STATUS
#NOT-3552	2026-06-23 04:35:44 WIB	Humidity DHT22 abnormal (84.4%)	Baru
#NOT-3551	2026-06-23 04:31:50 WIB	PH Pink abnormal (2.866081pH)	Baru
#NOT-3550	2026-06-23 04:30:48 WIB	Humidity DHT22 abnormal (85%)	Baru
#NOT-3549	2026-06-23 04:26:49 WIB	PH Pink abnormal (2.873602pH)	Baru
#NOT-3548	2026-06-23 04:25:44 WIB	Humidity DHT22 abnormal (84.6%)	Baru
#NOT-3547	2026-06-23 04:20:44 WIB	Humidity DHT22 abnormal (84.3%)	Baru
#NOT-3546	2026-06-23 04:15:44 WIB	Humidity DHT22 abnormal (83.6%)	Baru
#NOT-3545	2026-06-23 04:11:46 WIB	PH Pink abnormal (2.983468pH)	Baru

Gambar 4.31 Implementasi Halaman Notifikasi untuk Aktor *User*

#	WAKTU	PENGIRIM	PESAN	STATUS
#NOT-3561	2026-06-23 05:30:44 WIB	pak satriyo	Humidity DHT22 abnormal (86.1%)	✓ Dibaca
#NOT-3560	2026-06-23 05:25:44 WIB	pak satriyo	Humidity DHT22 abnormal (85.5%)	✓ Dibaca
#NOT-3559	2026-06-23 05:21:58 WIB	pak satriyo	PH Pink abnormal (2.873198pH)	✓ Dibaca
#NOT-3558	2026-06-23 05:20:44 WIB	pak satriyo	Humidity DHT22 abnormal (85.9%)	✓ Dibaca
#NOT-3557	2026-06-23 05:05:44 WIB	pak satriyo	Humidity DHT22 abnormal (86.2%)	✓ Dibaca
#NOT-3556	2026-06-23 05:00:44 WIB	pak satriyo	Humidity DHT22 abnormal (86.3%)	✓ Dibaca
#NOT-3555	2026-06-23 04:35:44 WIB	pak satriyo	Humidity DHT22 abnormal (86.1%)	✓ Dibaca
#NOT-3554	2026-06-23 04:51:53 WIB	pak satriyo	PH Pink abnormal (2.979663pH)	✓ Dibaca

Gambar 4.32 Implementasi Halaman Notifikasi untuk Aktor *Admin*

4.5 Pengujian Sistem

Pengujian sistem dalam penelitian ini dilakukan untuk memvalidasi bahwa seluruh fitur yang telah dikembangkan dapat berjalan sesuai dengan kebutuhan fungsional yang telah didefinisikan pada dokumen *Software Requirement Specification* (SRS). Sesuai dengan tahapan metodologi Scrum yang diterapkan, pengujian menyeluruh dilaksanakan pada setiap implementasi *sprint* sebagai langkah verifikasi. Metode yang digunakan adalah *Black Box Testing*, di mana pengujian difokuskan pada pemeriksaan fungsionalitas antarmuka dan alur proses sistem berdasarkan skenario use case yang telah dirancang sebelumnya. Melalui

pengujian ini, setiap butir uji diuji kesesuaiannya antara hasil aktual dengan hasil yang diharapkan berdasarkan skenario pada *use case* untuk memastikan sistem telah layak dan reliabel untuk diimplementasikan pada lingkungan operasional.

4.5.1 Rencana Pengujian Sistem

Perencanaan pengujian sistem ini disusun untuk memverifikasi bahwa seluruh fitur yang telah dikembangkan dapat berfungsi dengan baik sesuai dengan kebutuhan fungsional yang telah didefinisikan dalam dokumen SRS. Pengujian difokuskan pada validasi alur sistem dari sisi pengguna, mulai dari proses autentikasi hingga kontrol aktuator dan pengelolaan jadwal otomatisasi. Dengan menetapkan butir uji yang sistematis, diharapkan setiap fungsionalitas sistem dapat diperiksa kesesuaiannya dengan skenario yang telah dirancang, sehingga potensi kesalahan dapat dideteksi dan diperbaiki sebelum sistem dinyatakan siap untuk dioperasikan. Rencana pengujian ditunjukkan pada tabel 4.30.

Tabel 4.41 Rencana Pengujian

No	Kode Uji	Butir Uji	Kode Use Case
1	TC-SMAQ-1	Login dengan email dan password yang valid	UC-SMAQ-01
2	TC-SMAQ-2	Login dengan email dan/atau password yang tidak valid	
3	TC-SMAQ-3	Logout dari sistem	
4	TC-SMAQ-4	Melihat halaman pengaturan akun	UC-SMAQ-02
5	TC-SMAQ-5	Mengubah data identitas akun	
6	TC-SMAQ-6	Menghapus akun (menonaktifkan akun)	
7	TC-SMAQ-7	Melihat halaman notifikasi	UC-SMAQ-03
8	TC-SMAQ-8	Melihat halaman <i>dashboard admin</i>	UC-SMAQ-04
9	TC-SMAQ-9	Melihat halaman <i>dashboard user</i>	UC-SMAQ-05
10	TC-SMAQ-10	Melihat daftar data <i>user</i>	UC-SMAQ-06
11	TC-SMAQ-11	Menambahkan data <i>user</i> baru	
12	TC-SMAQ-12	Mengubah data <i>user</i>	
13	TC-SMAQ-13	Menghapus data <i>user</i>	UC-SMAQ-07
14	TC-SMAQ-14	Melihat daftar data device	
15	TC-SMAQ-15	Menambahkan data device baru	
16	TC-SMAQ-16	Mengubah data device	
17	TC-SMAQ-17	Menghapus data device	UC-SMAQ-08
18	TC-SMAQ-18	Melihat halaman monitoring device	
19	TC-SMAQ-19	Menghubungkan site dengan device oleh <i>admin</i>	
20	TC-SMAQ-20	Melepaskan device dari site oleh <i>admin</i>	UC-SMAQ-09
21	TC-SMAQ-21	Menghubungkan site dengan device oleh <i>user</i>	

No	Kode Uji	Butir Uji	Kode Use Case
22	TC-SMAQ-22	Melepaskan device dari site oleh <i>user</i>	
23	TC-SMAQ-23	Melihat daftar data aktuator	UC-SMAQ-10
24	TC-SMAQ-24	Menambahkan data aktuator baru	
25	TC-SMAQ-25	Mengubah data aktuator	
26	TC-SMAQ-26	Menghapus data aktuator	
27	TC-SMAQ-27	Melihat daftar data sensor	UC-SMAQ-11
28	TC-SMAQ-28	Menambahkan data sensor baru	
29	TC-SMAQ-29	Mengubah data sensor	
30	TC-SMAQ-30	Menghapus data sensor	
31	TC-SMAQ-31	Melihat daftar data site	UC-SMAQ-12
32	TC-SMAQ-32	Menambahkan data site baru	
33	TC-SMAQ-33	Mengubah data site	
34	TC-SMAQ-34	Menghapus data site	
35	TC-SMAQ-35	Melihat halaman log aktuator	UC-SMAQ-13
36	TC-SMAQ-36	Mengunduh file CSV log aktuator	
37	TC-SMAQ-37	Melihat halaman riwayat data sensor	UC-SMAQ-14
38	TC-SMAQ-38	Mengunduh file CSV riwayat data sensor	
39	TC-SMAQ-39	Melihat halaman jadwal grow light	UC-SMAQ-15
40	TC-SMAQ-40	Menambahkan jadwal grow light baru	
41	TC-SMAQ-41	Mengubah jadwal grow light	
42	TC-SMAQ-42	Menghapus jadwal grow light	
43	TC-SMAQ-43	Melihat halaman jadwal feeder	UC-SMAQ-16
44	TC-SMAQ-44	Menambahkan jadwal feeder baru	
45	TC-SMAQ-45	Mengubah jadwal feeder	
46	TC-SMAQ-46	Menghapus jadwal feeder	
47	TC-SMAQ-47	Mengaktifkan aktuator	UC-SMAQ-17
48	TC-SMAQ-48	Menonaktifkan aktuator	

4.5.2 Hasil Pengujian Sistem

Hasil pengujian dari seluruh butir uji yang telah direncanakan sebelumnya, termasuk pengujian fungsionalitas dan interaksi sistem, telah dirangkum ke dalam tabel hasil pengujian. Hasil pengujian mengacu pada target yang tercantum dalam skenario *use case*. Dengan demikian, butir uji dinyatakan diterima apabila hasil pengujian memenuhi seluruh skenario tersebut serta sesuai dengan *Definition of Done* (DoD). Laporan hasil pengujian selengkapnya disajikan pada Lampiran 3.