**PEMBANGUNAN APLIKASI MONITORING TANAMAN MEMANFAATKAN API DARI ALAT SIRAM TANAMAN OTOMATIS**

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana (S1)

**Prayogi Sukmana**

**10118165**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS KOMPUTER INDONESIA**

**2022**

# BAB 1 PENDAHULUAN

## Latar Belakang Masalah

Setiap rumah memiliki pekarangan sebagai penunjang keindahan rumah tersebut. Biasanya pekarangan rumah dipenuhi oleh tanaman, baik tanaman hias, tanaman obat maupun sayur-sayuran. Jenis tanaman yang ditanam tersebut berskala kecil dalam bentuk pot. Namun bagi rumah yang tidak memiliki pekarangan namun ingin tetap menanam tanaman, bisa menanam tanaman hias dalam skala yang lebih kecil lagi.

Tanaman merupakan salah satu makhluk hidup yang banyak memiliki manfaat untuk lingkungan sekitar. Berbagai macam manfaat dapat didapatkan dari sebuah tanaman, yaitu sebagai obat herbal, bahan makanan, dan yang paling utama adalah tanaman dapat menghasilkan oksigen yang berguna sebagai kelangsungan hidup manusia. Banyak sekali jenis-jenis tanaman, dari mulai bentuk fisiknya yang bermacam-macam, hingga manfaatnya yang berbeda-beda. Dari banyaknya tanaman tersebut tanaman dibagi menjadi beberapa kelompok. Salah satunya adalah kelompok tanaman hias.

Tanaman Hias atau Ornamental Plants diartikan sebagai semua tanaman yang memiliki nilai hias dari mulai bagian bunga, akar, daun, aroma dan batangnya sendiri, dan tanaman tersebut memiliki nilai estetis dan nilai seni (Santoso, 2010, h.5). Jadi dapat dikatakan bahwa segala tanaman yang memiliki fungsi sebagai keindahan dapat digolongkan menjadi kelompok tanaman hias. Tanaman hias sendiri memiliki jenis yang sangat banyak sekali diluar sana, dengan kondisi tersebut manfaat tanaman hias pun semakin meluas. Adapun fungsi lain tanaman hias selain fungsi estetis atau keindahan. Diantaranya adalah tanaman hias dapat berfungsi sebagai stabilisator dan pemeliharaan lingkungan, pendidikan, pemelihara kesehatan, serta ekonomi dan sosial (Rukmana, 2012, h.11).

Beberapa faktor penunjang tanaman agar dapat tumbuh dengan sempurna, yaitu suhu, intensitas cahaya, dan juga kelembapan tanah. Apabila suhu dari lingkungan tanaman melebihi batas optimal atau kurang dari batas optimal, maka akan menyebabkan tumbuhan tidak tumbuh secara optimal. Begitu juga apabila kelembapan tanah dan intensitas cahaya yang tidak mencukupi kebutuhan tanaman tersebut, maka tanaman tidak tumbuh secara optimal pula.

Pada Cactus ThreeD menyediakan tanaman hias berupa kaktus dan sukulen, berdasarkan hasil wawancara dan observasi, tanaman hias yang berada di Cactus ThreeD masih banyak yang tidak tumbuh secara optimal, dikarenakan lokasi dari greenhouse tersebut tidak mendapatkan factor penunjang tanaman yang optimal, yang menyebabkan kebanyakan tanaman hias yang berada disana menjadi busuk

Dari permasalahan diatas, maka dibutuhkan suatu solusi yang dapat membantu pihak pengelola dalam pengoptimalan pertumbuhan tanaman dan memantau kondisi tanaman yang mereka tanam.

## Identifikasi Masalah

Berdasarkan latar belakang di atas, maka didapatkan masalah yang dapat diidentifikasi, yaitu adalah sebagai berikut :

1. Tanaman harus selalu dijaga dan dipantau kondisi nya, salah satu cara nya dengan menyiram tanaman tersebut. Namun masih banyak orang yang tidak memantau kondisi tanaman mereka, dikarenakan kesibukan yang dimiliki atau lupa.
2. Selain dipantau kondisi melalui penyiraman tanaman, tanaman juga dipantau berdasarkan cuaca. Namun masih banyak orang yang hanya memantau kondisi dengan menyiram saja, tidak memperhatikan tanaman berdasarkan kondisi cuaca.

## Maksud dan Tujuan

1. Maksud

Maksud dari penelitian ini adalah untuk membangun sistem atau aplikasi *monitoring* padatanaman dalam proses pertumbuhan berdasarkan cuaca dan jenis tanaman, yang dimana sistem ini dapat memberitahu pengguna apa saja yang harus dilakukan kepada tanaman yang mereka tanam, seperti saat tanaman harus disiram, dipupuk, atau dipindah ke tempat yang lebih cerah atau lebih teduh. Dan sistem ini juga dapat melakukan penyiraman tanaman secara otomatis, dimana nanti sistem ini juga dihubungkan dengan alat untuk menyiram tanaman sehingga pengguna dapat menyiram tanaman dengan mudah tanpa perlu melakukannya sendiri dan sistem juga dapat memberikan notifikasi ketika ada tanaman yang harus disiram pada waktu tertentu

1. Tujuan

Tujuan dari penelitian akan dijabarkan sebagai berikut :

1. Membantu pengelola greenhouse Cactus ThreeD dalam mengontrol pertumbuhan tanaman kaktus
2. Membantu pengelola greenhouse dalam proses penyiraman secara otomatis pada media tanam tanaman kaktus agar mengurangi keterlambatanan dalam penyiraman
3. Membantu pengelola greenhouse memperoleh informasi kondisi jumlah kelembaban, intensitas cahaya, dan suhu sehingga kondisi media tanam tanaman kaktus bisa sesuai dengan kondisi optimal.

## Batasan Masalah

Adapun batasan dalam penelitian ini, yaitu adalah sebagai berikut :

1. Sistem yang dibangun untuk monitoring kelembaban tanah, intensitas cahaya, dan suhu pada media tanam greenhouse
2. Jumlah kelembaban yang dijadikan standar adalah berkisar 70%
3. Jumlah pH yang dijadikan standar adalah berkisar 5,5 hingga 7,0 atau 6,0 hingga 7,0.
4. Jumlah suhu yang dijadikan standar adalah 15 hingga 30 C dengan kelembaban udara rata-rata 50% hingga 60%.
5. Sistem digunakan pada media tanam untuk tanaman kaktus dan sukulen
6. Sistem hanya dapat memonitoring kondisi jumlah kelembaban, suhu dan pH pada media tanam budidaya tanaman mawar
7. Sistem yang dibangun menggunakan koneksi internet agar dapat memonitoring secara *real-time*.
8. API yang digunakan terdapat data – data tanaman, yang meliputi
9. Data Kelembapan Tanah
10. Data Intensitas Cahaya
11. Data Suhu
12. Sistem yang dibangun hanya memberi informasi kondisi kelembaban, pH tanah dan suhu dengan menggunakan framework IoT bebasis website

## Metodologi Penelitian

Metodologi Penelitian yang akan dilaksanakan merupakan sebuah rangkaian tahapan yang disusun secara sistematis, berikut adalah gambaran dari metode penelitian yang akan berjalan.

Dari *flow map* di atas dapat didefinisikan sebagai berikut :

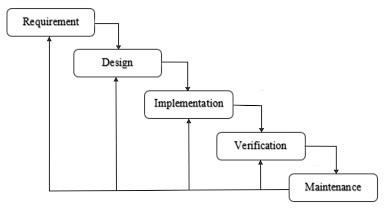
### Pengumpulan Data

Tahapan pertama pada metode penelitian ini adalah pengumpulan informasi. Untuk pengumpulan informasi, akan melalui berbagai cara seperti mencari informasi melalui media buku, atau jurnal. Bahkan peneliti akan melakukan wawancara dan melakukan penelitian di Greenhouse Cactus ThreeD untuk mengumpulkan informasi yang mungkin tidak bisa didapatkan selain dengan mencari informasi melalui buku atau jurnal.

### Analisis Masalah

Selanjutnya adalah tahapan analisis masalah. Sebelum membuat ide, hasil informasi yang didapat dari pengumpulan perlu dianalisis terlebih dahulu.

### Pembangunan Perangkat Lunak

Pada tahapan ini, metode dalam pembangunan perangkat lunak yang digunakan adalah metode Waterfall. Karena metode ini cocok digunakan pada penelitian yang skalanya kecil dan juga memiliki tahapan-tahapan yang sederhana. Ada empat tahapan dalam metode ini yaitu Analisis kebutuhan, Desain, Implementasi dan Pengujian.

Dari gambar metode Waterfall di atas maka prosedur *waterfall* yang diterapkan pada Aplikasi ini sebagai berikut :

1. Analisis Kebutuhan, merupakan tahapan paling awal yang ada dalam metode *waterfall*. Pada tahapan ini diperoleh informasi-informasi yang dibutuhkan untuk keperluan penelitian akan didapatkan melalui media buku, atau jurnal. Bahkan peneliti akan melakukan wawancara dan melakukan penelitian di Cactus ThreeD untuk mengumpulkan informasi yang mungkin tidak bisa didapatkan selain dengan mencari informasi melalui buku atau jurnal.
2. Desain Sistem, tahapan ini mengambil masukan-masukan dari informasi yang telah didapatkan dari tahapan analisis kebutuhaan yang sebelumnya dilakukan. Tahapan ini merupakan perencanaan solusi perangkat lunak yang bisa mencakup desain sistem dan kebutuhan fungsional atau non-fungsionalnya. Penulis menggunakan perangkat keras yaitu laptop. Sedangkat *software*nya penulis menggunakan Bahasa pemrograman Vue.js untuk bagian desain antarmuka, dan Bahasa pemrograman Laravel untuk bagian desain sistem, IDE akan menggunakan Visual Studio Code, sedangkan desain prototype menggunakan aplikasi Figma.
3. Implementasi, pada tahap inilah pengembangan aktual sistem terjadi sesuai dengan spesifikasi desain. Hasil dari langkah ini adalah *prototype* sistem yang dibangun dengan harapan dapat memenuhi semua kebutuhan fungsionalnya.
4. Pengujian, pada tahap ini sistem yang dilakukan oleh penguji terlebih dahulu. Pada tahapan pengujian, peneliti akan menjalanakan sistem yang telah terhubung dengan alat siram otomatis.
5. Maintenance, tahap dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan - perubahan atau penambahan sesuai dengan kritik dan saran yang diberikan. Pada aplikasi ini, nanti jika aplikasi telah dirilis dan ada beberapa fitur yang harus dikembangkan maka sangat diperlukan proses *Maintenance*

## Sistematika Penulisan

Untuk memudahkan penulisan dalam penyusunan skripsi ke arah yang dimaksud, maka digunakan sistematika penulisan yang nantinya akan mempermudah penulisan skripsi, meliputi :

1. BAB 1 - Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, maksud dan tujuan, manfaat, batasan masalah, metodologi penelitian dan sistematika penulisan. Singkatnya bab ini menjelaskan permasalahan yang sedang terjadi dan menjelaskan solusi untuk menyelesaikan masalah tersebut. Terlihat Seperti yang telah ditulis diatas Bab 1 merupakan sebuah uraian tentang latar belakang masalah terbuatnya sistem monitoring tanaman, maksud dan tujuan dibuatnya sistem tersebut, batasan masalah yang digunakan dalam proses pembuatan sistem ini, dan sistematika penulisan laporan ini.

1. BAB 2 – Tinjauan Pustaka

Bab ini membahas teori-teori pendukung yang digunakan dalam proses perencanaan dan proses pembuatan pada penelitian, seperti landasan teori tentang tanaman, tanaman hias, Vue JSdan yang lainnya yang berhubungan dengan penelitian ini

1. BAB 3 – Analisis dan Perancangan Sistem

Bab ini berisi tentang analisis, dan perancangan sistem pada penelitian. Untuk penelitian ini, analisis akan menggunakan tool tambahan berupa UML atau Unified Modelling Language. Bab 3 ini bertujuan untuk memudahkan pembaca atau peneliti selanjutnya dalam memahami fungsionalitas apa saja yang dimiliki oleh sistem.

1. BAB 4 – Implementasi dan Pengujian Sistem

Bab ini berisi tentang implementasi dan juga pengujian sistem. Pengujian yang akan dilakukan adalah pengujian secara black box. Selain itu, bab ini juga akan membahas tentang hasil pengujian yang dilakukan langsung oleh pengguna.

1. BAB 5 – Kesimpulan dan Saran

Bab ini berisi kesimpulan yang diperoleh selama melakukan penelitian dan menghasilkan saran-saran untuk memperbaiki sistem di penelitian berikutnya apabila ada yang melanjutkan

# BAB 2 TINJAUAN PUSTAKA

## Tanaman

Tanaman merupakan salah satu makhluk hidup yang banyak memiliki manfaat untuk lingkungan sekitar. Berbagai macam manfaat dapat didapatkan dari sebuah tanaman, yaitu sebagai obat herbal, bahan makanan, dan yang paling utama adalah tanaman dapat menghasilkan oksigen yang berguna sebagai kelangsungan hidup manusia. Banyak sekali jenis-jenis tanaman, dari mulai bentuk fisiknya yang bermacam-macam, hingga manfaatnya yang berbeda-beda. Dari banyaknya tanaman tersebut tanaman dibagi menjadi beberapa kelompok. Salah satunya adalah kelompok tanaman hias.

### Tanaman Kaktus

Tumbuhan kaktus menjadi salah satu tanaman yang cukup terkenal di masyarakat. Tidak hanya di negara asalnya, melainkan di berbagai macam negara mengetahui tanaman kaktus. Mendengar kata kaktus orang akan mengingat sebuah tumbuhan yang memiliki duri pada batang tubuhnya. Berikut adalah pandangan penulis mengenai tumbuhan kaktus, yakni, Kaktus adalah tumbuhan yang termasuk kedalam keluarga Cactaceae yang memiliki lebih dari 2000 varietas dengan berbagai warna dan bentuk.

Kaktus dikenal sebagai tumbuhan sukulen karena bagian tubuhnya seperti batang, akar, dan daun mampu menyimpan air untuk bertahan hidup. Kaktus adalah tanaman yang memiliki bentuk dan cara bertahan hidup yang unik dan berbeda dari jenis tanaman lain. Kaktus dikenal memiliki duri yang tak lain adalah daunnya. Daun kaktus tumbuh dari struktur khusus yang disebut areole. Ukuran duri bervariasi dan bisa mencapai panjang 15 cm. Daun yang berubah menjadi duri bertujuan untuk mencegah penguapan air berlebih. Sebagian duri kaktus bahkan berwarna terang yang membantu mereka memantulkan sinar matahari sehingga tanaman tetap dingin.

Tumbuhan kaktus memiliki bunga yang sangat indah. Pertumbuhan bunga kaktus terbilang lambat. Butuh waktu sekitar 2 tahun untuk bunga kaktus dapat mekar sempurna. Beberapa bunga kaktus dapat bertahan hidup selama beberapa hari setelah mekar, adapula yang hanya bertahan hidup selama sehari setelah mekar. Keadaan waktu yang dibutuhkan bunga untuk mekarpun berbeda tergantung jenis kakstunya. Sebagian bunga hanya dapat mekar pada saat malam hari dan sebagian lain hanya dapat mekar pada saat siang hari.

Keunikan terjadi pula pada akar kaktus. Tanaman akar yang sangat panjang. Kedalaman akar kaktus tidak lebih dari 10 cm berada di bawah permukaan tanah. Namun, akar kaktus dapat menyebar ke sekeliling hingga meliputi diameter 2 meter. Hal tersebut dilakukan untuk mencari sumber air tanah yang berada di wilayah sekitar kaktus berada. Kaktus merupakan tanaman dengan cara bertahan hidup yang luar biasa. Kaktus mampu hidup di daerah kering seprti gurun pasir, semi gurun dan padang rumput. hidup daerah karena memiliki beberapa kemampuan unik untuk kaktus mampu bertahan hidup dengan menyesuaikan keadaan di lingkungan sekitarnya dengan jumlah air yang sedikit. Kemampuan tersebut menjadikan tumbuhan kaktus istimewa, unik dan amat berbeda dengan jenis tanaman lain.

### Tanaman Sukulen

Sukulen adalah istilah umum untuk tumbuhan yang dapat menyimpan air di dalamnya. daun. Sukulen atau succulents berasal dari kata latin “Sukkot”. Berarti jus. Dalam bahasa Inggris artinya Berair. Berair Ia memiliki kemampuan untuk menyimpan air di batang dan pulp (Coyne & Knutzen, 2014, hal.2). Di sisi lain, menurut Sueno, sukulen ada di Jati (2016). Tumbuhan dengan batang berdaging tetapi daunnya tebal Itu bukan duri, tapi masih berbentuk seperti daun. Daun sukulen juga berbentuk Lemak dan cair (hal.8). Daun kaya air Tanaman sukulen ini berfungsi sebagai adaptasi terhadap iklim kering Tanaman ini dapat bertahan lama dalam keadaan kering, Itu bisa menahan penguapan. Namun, ada beberapa jenis sukulen Ini dapat menahan kondisi kering dan dingin seperti yucca, houseleek, sedum, dan beberapa agave dan kaktus.

## Vue JS

Vue JS merupakan sebuah kerangka kerja berbasis Javascript yang digunakan untuk membangun tampilan antarmuka pada aplikasi web yang dikembangkan oleh Evan You pada tahun 2013.

Dalam penggunaannya, Vue JS dapat diintregasikan dengan kerangka kerja atau *library* Javascript yang lainnya.

Vue ini dapat menjalankan aplikasi ­*Single Page Application* yang canggih, dimana apabila pengguna melakukan pergantian halaman, maka *browser* yang digunakan tidak akan melakukan *refresh* ketika membuka halaman yang baru, sehingga akan terasa lebih cepat dalam penggunaan web.

Dalam pembangunan aplikasi monitoring tanaman ini menggunakan framework VueJS karena dalam penggunannya menggunakan teknologi *Single Page Application* yang dapat melakukan perubahan tanpa harus melakukan *refresh* pada setiap halaman

## Laravel

Laravel merupakan *framework* dari Bahasa pemrograman PHP yang bersifat *open-souce* menggunakan konsep *Model-View-Controller*(MVC) yang dikembangkan pada tahun 2011 oleh Taylor Otwell.

Laravel diciptakan unutk menyederhanakan proses pengembangan perangkat lunak, dimana dalam pembuatan suatu aplikasi menggunakan Laravel, semua basic nya sudah disediakan dalam dokumentasi Laravel, contoh nya jika ingin membuat *Controller*, kita cukup memasukkan syntax pada command-prompt, maka otomatis laravel akan membuatkannya.

Fungsi dalam aplikasi

Alasan karena

## Vue-Router

Vue-Router merupakan sebuah teknologi yang digunakan dari framework Vue JS,

## Axios

Axios merupakan sebuah *library* yang bersifat terbuka (*open-source)* yang digunakan untuk mengambil atau menampilkan data dari sebuah API. *Library* Axios ini biasa digunakan dalam beberapa Bahasa pemrograman, seperti Vue JS, Axios ini lebih popular dalam pendekatan untuk permintaan atau pengambilan data dari sebuah API.

Axios digunakan dalam pemanggilan data backend dari Laravel, data yang dipanggil pada backend adalah data tanaman yang terdapat pada halaman home, dan juga digunakan dalam pemanggilan data data tanaman dari API Alat Siram Tanaman Otomatis

## MySQL

MySQL dapat dikategorikan perangkat lunak yang membuat database bersifat berbuka (open source). MySQL adalah sistem yang berguna untuk melakukan proses pengaturan koleksi-koleksi struktur data (database) baik meliputi proses pembuatan atau proses pengelolaan database [8]. MySQL banyak digunakan pada kalangan promgrammer karena open source dan mudah dipahami. MySQL bisa disebut perangkat lunak yang membuat database sebagai sumber dari pengolahan data untuk membangun aplikasi ataupun website.

MySQL digunakan dalam menyimpan data tanaman, data kelembapan tanah, intensitas cahaya, dan suhu yang telah didapat dari API yang mengirimkan data ke sistem

## API Alat Siram Tanaman Otomatis

## MQTT

MQTT merupakan sebuah protokol yang diterapkan pada IOT. Protokol ini sangat mendukung untuk jaringan WAN, karena WAN mencakup area yang luas. Protokol MQTT mempunyai kelebihan yaitu dapat bekerja dengan energi dan media penyimpanan yang minimum.

MQTT digunakan dalam penghubung sistem dengan alat, teknologi yang digunakan dalam penghubungan sistem dengan alat ini adalah menggunakan WebSocket, cara penggunaan WebSocket ini dengan memanggil url host, port dan endpoint dari alat, MQTT

# BAB 3 ANALISIS DAN PERANCANGAN

## Analisis

Tahapan analisis bertujuan untuk menganalisis permasalahan-permasalahan yang ada serta menentukan kebutuhan apa saja yang dibutuhkan oleh sistem yang akan dibangun. Analisis yang akan dibahas adalah analisis masalah, analisis teknologi yang digunakan, analisis sistem yang dibangun, analisis perancangan arsitektur sistem, analisis kebutuhan fungsional dan analisis kebutuhan non fungsional. Analisis ini bertujuan untuk membuat sistem yang akan dibangun dapat menyelesaikan semua permasalahan yang ada di Bab 1.

### Analisis Masalah

Analisis masalah merupakan tahapan penjabaran mengenai masalah yang ada sebelum pengembangan sistem dibuat. Dalam penelitian ini, permasalahan pertama yang dihadapi yaitu sulitnya untuk mengontrol tanaman kaktus dan sukulen, dikarenakan belum adanya sistem monitoring tanaman kaktus dan sukulen berdasarkan data suhu, kelembapan tanah dan intensitas cahaya yang dapat mengolah atau memproses data suhu, kelembapan tanah dan intensitas cahaya untuk didapatkan hasil kondisi tanaman.

### Analisis Proses Bisnis

Analisis proses bisnis merupakan tahapan yang bertujuan untuk mengetahui proses bisnis apa saja yang aka nada pada sistem yang akan dibangun

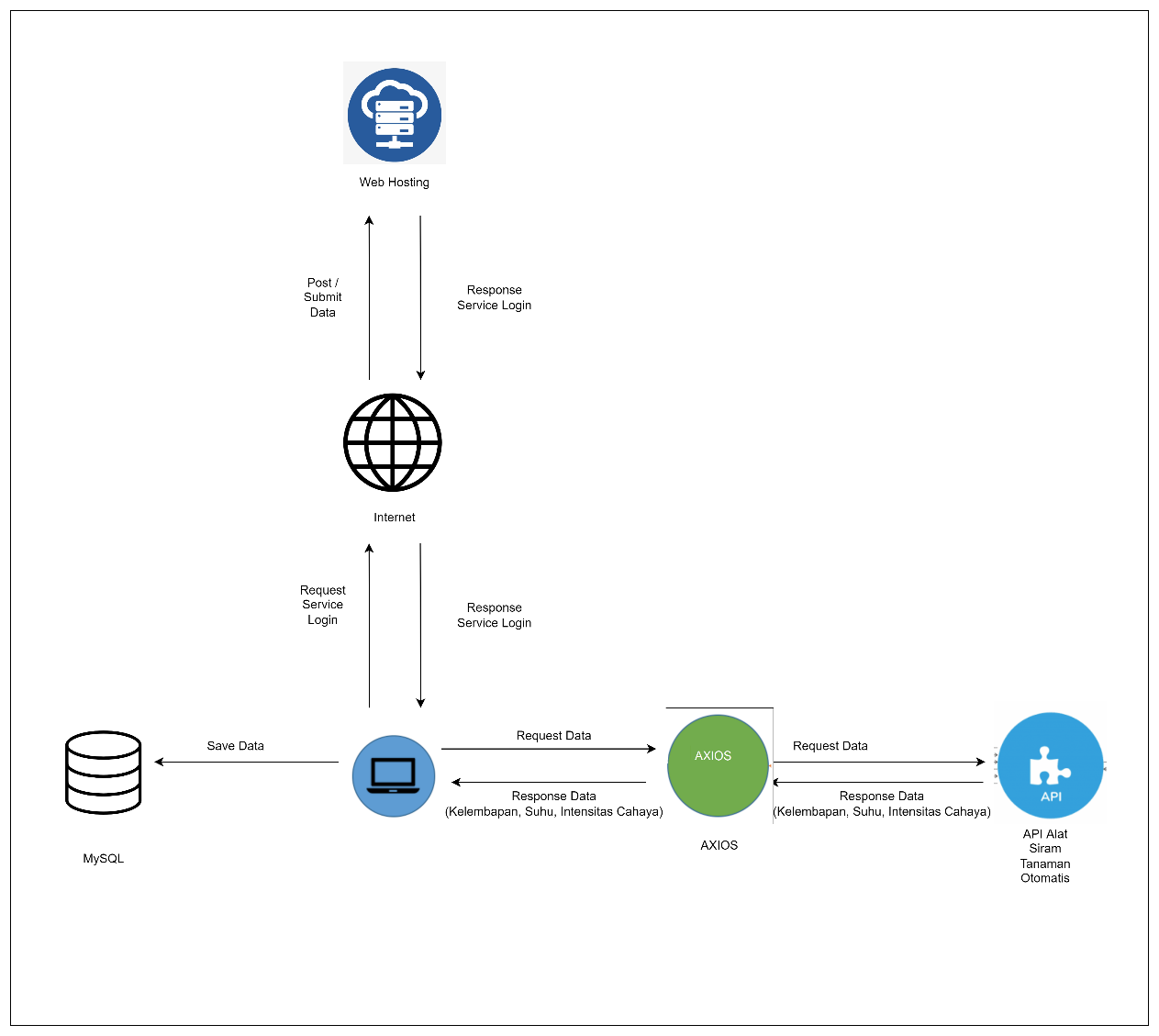
#### Analisis Proses Bisnis Sistem Yang Akan Dibangun

Analisis proses bisnis dari sistem yang akan dibangun merupakan gambaran dari bagaimana cara kerja sistem yang akan dibangun pada penelitian ini. Adapuun alur dari sistem yang akan dibangun adalah sebagai berikut :

1. Proses Bisnis Deteksi Kelembapan Tanah
2. Alat melakukan deteksi pada media tanam
3. Alat mengirimkan data ke API melalui MQTT
4. Sistem melakukan *request* data kelembapan tanah ke API
5. API menerima *request* dari sistem
6. API mengirimkan data kelembapan tanah ke sistem
7. Sistem menerima data kelembapan tanah dari API
8. Sistem menampilkan keterangan berdasarkan data kelembapan tanah yang didapat dari API
9. Sistem menyimpan data dari API ke database
10. Proses Bisnis Penyiraman Tanaman Otomatis
11. Alat melakukan deteksi pada media tanam
12. Alat mengirimkan data ke API melalui MQTT
13. Sistem melakukan *request* data kelembapan tanah ke API
14. API menerima *request* dari sistem
15. API mengirimkan data kelembapan tanah ke sistem
16. Sistem menerima data kelembapan tanah dari API
17. Sistem mengolah informasi kelembapan tanah, apabila kelembapan tanah diatas 700 RH, maka kran akan menyala otomatis, dan apabila kelembapan tanah dibawah 700 RH, maka kran akan otomatis mati
18. Proses Bisnis Deteksi Intensitas Cahaya
19. Alatmelakukan deteksi pada sekitar tanaman
20. Alat mengirimkan data ke API melalui MQTT
21. Sistem melakukan *request* data intensitas cahaya ke API
22. API menerima *request* dari sistem
23. API mengirimkan data intensitas cahaya ke sistem
24. Sistem menerima data intensitas cahaya tanah dari alat
25. Sistem menampilkan keterangan berdasarkan data intensitas cahaya yang didapat dari alat
26. Proses Bisnis Menyalakan Lampu LED Otomatis
27. Sensor *Light Dependant Resistor* melakukan deteksi pada sekitar tanaman
28. Sistem melakukan *request* data intensitas cahaya ke alat
29. Alat Menerima *request* dari sistem
30. Alat mengirimkan data kelembapan intensitas cahaya ke sistem
31. Sistem menerima data intensitas cahaya tanah dari alat
32. Sistem mengolah informasi intensitas cahaya, apabila intensitas cahaya dibawah …., maka lampu LED akan menyala otomatis, dan apabila intensitas cahaya dibawah, maka lampu akan otomatis mati
33. Proses Bisnis Deteksi Suhu
34. Sensor NTC Thermistormelakukan deteksi pada sekitar tanaman
35. Sistem melakukan *request* data suhu ke alat
36. Alat Menerima *request* dari sistem
37. Alat mengirimkan data suhu ke sistem
38. Sistem menerima data suhu dari alat
39. Sistem menampilkan keterangan berdasarkan data suhu yang didapat dari alat
40. Proses Bisnis Menyalakan Blower Otomatis
41. Sensor NTC Thermistormelakukan deteksi pada sekitar tanaman
42. Sistem melakukan *request* data suhu ke alat
43. Alat Menerima *request* dari sistem
44. Alat mengirimkan data suhu ke sistem
45. Sistem menerima data suhu dari alat
46. Sistem mengolah informasi suhu, apabila suhu diatas …., maka blower akan menyala otomatis, dan apabila intensitas cahaya dibawah, maka blower akan otomatis mati

### Analisis Arsitektur Sistem

Arsitektur dibuat untuk mendifinisikan komponen – komponen yang ada dalam sistem secara lebih spesifik. Pada penelitian ini, arsitektur sistem nya adalah sebagai berikut :



Keterangan :

1. Sistem mengirimkan request service untuk login melalui internet
2. Web Hosting menerima request login dari sistem, melakukan proses pengecekan data username yang dikirimkan sistem, dan mengirimkan response datanya kembali kepada sistem
3. Melalui Axios, sistem mengirimkan *request* untuk mengambil data tanaman dari alat siram tanaman otomatis
4. Alat siram tanaman otomatis menerima permintaan dari sistem, dan mengirimkan data kepada sistem
5. Sistem menyimpan data yang diterima dari alat siram tanaman ke dalam basis data untuk nantinya diproses dan ditampilkan kepada pengguna

### Analisis Teknologi Yang Digunakan

Analisis teknologi dilakukan bermaksud untuk memberi gambaran teknologi apa saja yang digunakan dalam penelitian ini

#### API Alat Siram Tanaman Otomatis

API Alat Siram Tanaman Otomatis merupakan API yang dibuat oleh peneliti untuk digunakan dalam sistem ini agar dapat memanfaatkan *Service* pada aplikasi, API Alat Siram Tanaman Otomatis ini digunakan oleh sistem untuk menampilkan data - data dari suatu tanaman, seperti data kelembapan tanah, suhu dan intensitas cahaya, juga untuk menyalakan dan mematikan kran yg terdapat pada tanaman. Untuk mengimplementasikan API Alat Siram Tanaman Otomatis ke sistem, langkah – langkah nya adalah sebagai berikut

1. Menginstall plugin Axios terlebih dahulu
2. Setelah menginstall plugin Axios, panggil fungsi Axios pada halaman yang akan menampilkan data – data dari API, untuk pemanggilan fungsi Axios dapat dilihat pada gambar berikut

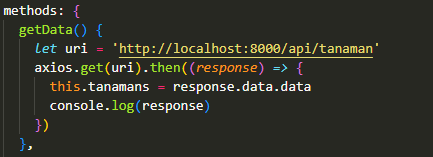


1. Langkah selanjutnya tambahkan *method* ambil data dengan menambahkan skirp url dari API yang digunakan, dan tambahkan *field* apa saja yang akan ditampilkan pada sistem
2. Setelah menambahkan url dari API yang digunakan, maka pada halaman utama, tampilkan *field* yang dipanggil dari skrip url sebelumnya

#### Axios

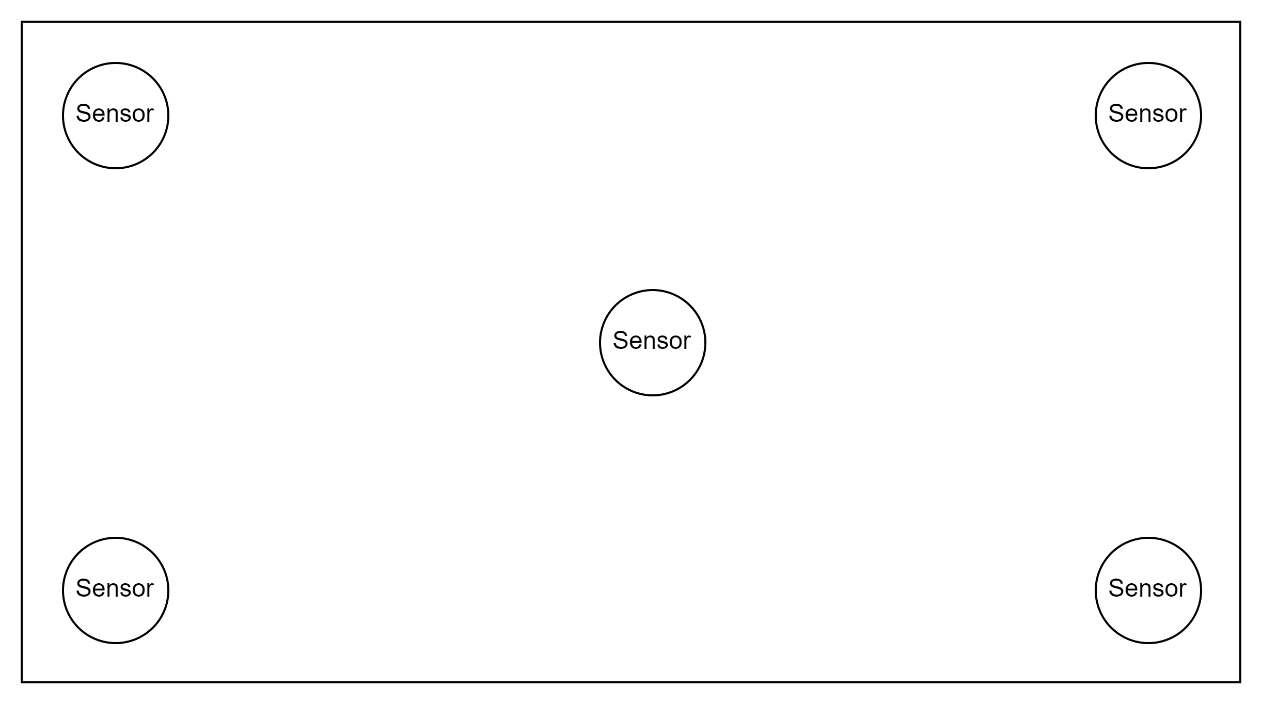
Axios merupakan sebuah *library* yang bersifat terbuka (*open-source)* yang digunakan untuk mengambil atau menampilkan data dari sebuah API. *Library* Axios ini biasa digunakan dalam beberapa Bahasa pemrograman, seperti Vue JS, Axios ini lebih popular dalam pendekatan untuk permintaan atau pengambilan data dari sebuah API.

Singkatnya Axios digunakan pada setiap website yang akan dihubungkan dengan data yg terdapat pada API, maka website harus mengirimkan request ke API melalui Axios, untuk melakukan *request* data kepada API, dapat dilihat dari gambar berikut ini



#### Vue-Router

### Analisis Denah Greenhouse



### Analisis SKPL

Analisis SKPL atau spesifikasi kebutuhan perangkat linak merupakan tahapan yang menjelaskan hasil dari analisis kebutuhan – kebutuhan fungsional apa saja yang harus ada pada sistem supaya dapat memecahkan masalah – masalah yang ada di identifikasi masalah. Selain kebutuhan fungsional, tahapan ini juga membahasa kebutuhan non-fungsional

#### Analisis Spesifikasi Kebutuhan Fungsional

Berdasarkan dari analisis masalah, maka didapatkan spesifikasi kebutuhan fungsional, yaitu dapat dilihat pada tabel di bawah ini.

|  |  |  |
| --- | --- | --- |
| No | SKPL-ID | Spesifikasi Kebutuhan Fungsional |
| 1 | SKPL-F01 | Sistem dapat mengambil data kelembapan tanah pada tanaman |
| 2 | SKPL-F02 | Sistem dapat mengambil data intensitas cahaya di sekitar tanaman |
| 3 | SKPL-F03 | Sistem dapat mengambil data suhu di sekitar tanaman |
| 4 | SKPL-F04 | Sistem dapat menampilkan informasi apabila terdapat salah satu parameter bernilai kurang dari nilai optimal |
| 5 | SKPL-F05 | Sistem dapat melakukan penyiraman tanaman |

#### Analisis Spesifikasi Kebutuhan Non-Fungsional

Sementara untuk kebutuhan non-fungsional, bisa dilihat pada tabel berikut

|  |  |  |
| --- | --- | --- |
| No | SKPL-ID | Spesifikasi Kebutuhan Perangkat Lunak |
| 1 | SKPL-NF01 | Sistem yang dibuat berbasis website |
| 2 | SKPL-NF02 | Bahasa pemrograman yang dibangun untuk membangun sistem yaitu Javascript untuk *Front-end* dengan menggunakan framework Vue.JS dan PHP untuk *Back-end* dengan menggunakan framework Laravel |
| 3 | SKPL-NF03 | Basis data yang digunakan untuk sistem yaitu MySQL |
| 4 | SKPL-NF04 | Sistem dapat berjalan dalam browser Mozilla Firefox, Google Chrome, dan Microsoft Edge |
| 5 | SKPL-NF05 | Untuk perangkat keras, spesifikasi minimal supaya sistem dapat berjalan yaitu prosesor Intel Core i3 5005U , RAM 2GB, Harddisk 100GB, Monitor mendukung 16 bit *colour* |

### Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan perangkat lunak yang dibangun. Kebutuhan non fungsional untuk menjalankan sistem yang dibangun meliputi kebutuhan perangkat lunak, kebutuhan perangkat keras dan kebutuhan perangkat pikir.

#### Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan proses analisis yang lebih menekankan kepada aspek pemanfaatan software. Kebutuhan non fungsional perangkat lunak dapat dilihat pada tabel 3.7 berikut

|  |  |  |
| --- | --- | --- |
| No | SKPL-ID | Spesifikasi Kebutuhan Perangkat Lunak |
| 1 | SKPL-NF01 | Sistem yang dibuat berbasis website |
| 2 | SKPL-NF02 | Bahasa pemrograman yang dibangun untuk membangun sistem yaitu Javascript untuk *Front-end* dengan menggunakan framework Vue.JS dan PHP untuk *Back-end* dengan menggunakan framework Laravel |
| 3 | SKPL-NF03 | Basis data yang digunakan untuk sistem yaitu MySQL |
| 4 | SKPL-NF04 | Sistem dapat berjalan dalam sistem operasi Windows 10 |

##### Fakta Spesifikasi Perangkat Lunak Penliti

Dalam pembuatan sistem, perangkat lunak yang dipakai oleh peneliti dilampirkan dalam tabel 3.8 berikut.

|  |  |
| --- | --- |
| Sistem Operasi | Microsoft Windows 10 |
| DBMS | MySQL |
| Aplikasi Pembuatan | Visual Studio Code |

#### Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras merupakan proses analisis yang lebih menekankan kepada aspek pemanfaatan hardware. Kebutuhan non fungsional perangkat keras dapat dilihat pada tabel 3.10 berikut.

|  |  |  |
| --- | --- | --- |
| No | SKPL-ID | Spesifikasi Kebutuhan Perangkat Lunak |
| 1 | SKPL-NF05 | Untuk spesifikasi kebutuhan perangkat keras, spesifikasi minimal supaya sistem dapat berjalan optimal yaitu, Prosesor Intel® Pentium® 4, RAM 2GB, Harddisk 80GB |
| 2 | SKPL-NF06 | Untuk kebutuhan sensor, perangkat yang dipastikan kompatibel yaitu yang memiliki board yang dapat terhubung dengan Node MCU |

##### Fakta Spesifikasi Perangkat Keras Peneliti

Fakta spesifikasi perangkat keras (komputer) yang peneliti gunakan selama penelitian dapat dilihat pada tabel 3.11 berikut

|  |  |  |
| --- | --- | --- |
| No | Nama Item | Spesifikasi |
| 1 | CPU | Intel Core i3 5005U CPU @ 2.00 GHz |
| 2 | Memory | 6GB |
| 3 | Storage | 500GB |
| 4 | GPU | Intel® HD Graphic 5500 |

#### Analisis Kebutuhan Perangkat Pikir

Analisis kebutuhan perangkat pikir yang dilakukan pada penelitian ini akan ditargetkan dengan empat kategori yaitu tanggung jawab, tingkat pendidikan, tingkat keterampilan yang dimiliki, dan pengalaman menggunakan komputer. Perangkat pikir yang terlibat dalam sistem ini adalah pengguna biasa atau individu yang ingin kondisi kesehatan jantungnya selalu di monitoring, dan olahragawan. Setiap perangkat piker pastinya memiliki hak akses, namun karena ini sistem yang akan dibangun untuk memecahkan semua masalah yang ada di batasan masalah, maka setiap perangkat pikir dapat mengakses semua hak akses yang ada. Untuk kriteria yang ditargetkannya, dapat dilihat pada tabel 3.15 berikut.

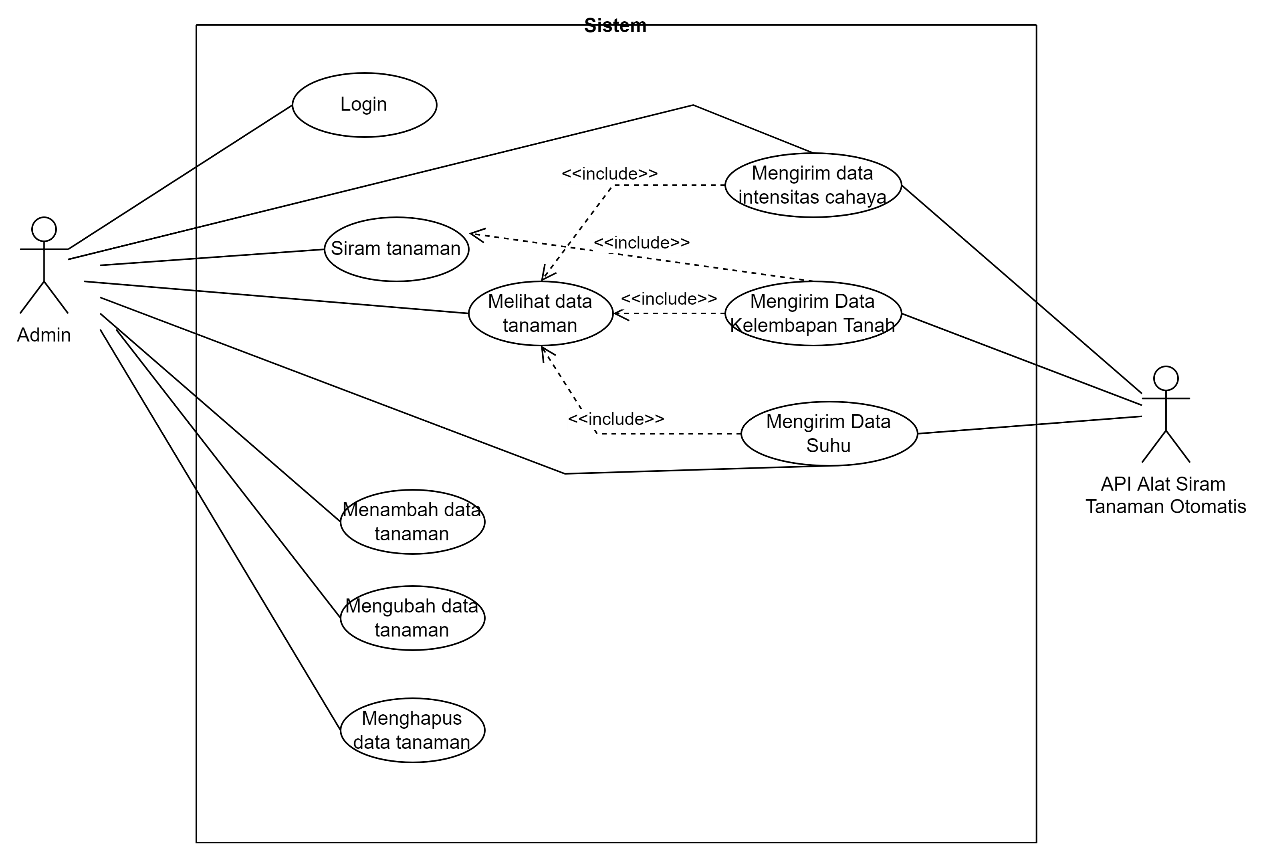
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Pengguna Sistem | Hak Akses | Tingkat Pendidikan | Tingkat Keterampilan yang Harus Dimiliki | Pengalaman Menggunakan Komputer |
| 1 | Pengelola Greenhouse | 1. Dapat mengakses menu tanaman 2. Dapat mendapatkan notifikasi ketika media tanam sebuah tanaman kering 3. Dapat mendapatkan notifikasi ketika intensitas cahaya pada suatu tanaman kurang dari nilai optimal 4. Dapat mendapatkan notifikasi ketika suhu di sekitar tanaman kurang dari nilai optimal 5. Dapat melakukan penyiraman tanaman 6. Dapat melihat riwayat data tanaman | SMP/SMA/SMK/Sederajat/D3/S1/Jenjang Lebih Tinggi | Memiliki kemampuan untuk menggunakan computer atau smartpohne | Baik, |

### Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan menggunakan bantuan tools tambahan, ayitu UML, adapun tahapan analisis sistem menggunakan UML 70 meliputi Use Case Diagram, Use Case Scenario, Activity Diagram, Sequence Diagram, dan Class Diagram.

#### Use Case Diagram

Use case diagram adalah pemodelan yang digunakan untuk menggambarkan kelakuan sistem yang dibuat. Use case diagram menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang dibuat. Secara sederhana, use case diagram digunakan untuk memahami fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut. Berikut use case diagram untuk sistem yang akan dibangun pada gambar berikut ini.



#### Definisi Aktor

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Aktor | Aktor dengan role ini memiliki wewenang untuk menambahkan dan memanipulasi data tanaman serta melakukan penyiraman tanaman. |
| 2 | API Alat Siram Tanaman Otomatis | Aktor dengan role ini memiliki wewenang untuk mengirimkan data pada tanaman. |

#### Definisi Use Case

|  |  |  |
| --- | --- | --- |
| **No** | **Use Case** | **Deskripsi** |
| 1 | Login | **Kode Use Case**: UC-01  **Deskripsi Singkat**: Sistem menyediakan halaman login sebelum memasuki laman website  **Aktor**: Admin  **Trigger**: Aktor memasukan username dan password  **Prekondisi**: Halaman utama web ditampilkan  **Pascakondisi**: Sistem menampilkan halaman awal website berupa informasi beberapa tanaman yang sudah ditanam |
| 2 | Siram Tanaman | **Kode Use Case**: UC-02  **Deskripsi Singkat**: Sistem menyediakan halaman untuk melakukan penyiraman tanaman.  **Aktor**: Admin  **Trigger**: Aktor mengetuk tombol “Siram” pada halaman detail tanaman  **Prekondisi**: Halaman untuk detail ditampilkan  **Pascakondisi**: Alat melakukan penyiraman tanaman, dan sistem menerima data kelembapan tanah, suhu yang sudah diupdate dari API. |
| 3 | Melihat Data Tanaman | **Kode Use Case**: UC-03  **Deskripsi Singkat**: Sistem menampilkan data tanaman yang sudah ditanam.  **Aktor**: Admin  **Trigger**: Aktor harus login  **Prekondisi**: Halaman login ditampilkan  **Pascakondisi**: Sistem menampilkan data tanaman. |
| 4 | Menambah Data Tanaman | **Kode Use Case**: UC-04  **Deskripsi Singkat**: Sistem menyediakan tombol untuk menambah data tanaman.  **Aktor**: Admin  **Trigger**: Aktor memilih icon “+” pada halaman utama website dan mengisi form lalu memilih tombol “simpan” pada halaman tambah tanaman.  **Prekondisi**: Halaman home akan ditampilkan  **Pascakondisi**: Sistem menambahkan data kedalam database. |
| 5 | Menghapus Data Tanaman | **Kode Use Case**: UC-05  **Deskripsi Singkat**: Sistem menyediakan pilihan untuk menghapus data tanaman.  **Aktor**: Admin  **Trigger**: Aktor memilih icon “:” pada halaman utama website dan memilih tombol “hapus” pada menu.  **Prekondisi**: Halaman home akan ditampilkan  **Pascakondisi**: Sistem menghapus data kedalam database. |
| 6 | Mengirimkan data intensitas cahaya | **Kode Use Case:** UC-06  **Deskripsi Singkat:** Alat akan mengirimkan data setiap ada perubahan intensitas cahaya  **Aktor:** API Alat Siram Tanaman Otomatis  **Trigger:** Perubahan intensitas cahaya pada tanaman  **Prekondisi:** Alat mengirimkan data pada sistem  **Pascakondisi:** Alat mengedit data yang ada di database ketika terjadi perubahan. |
| 7 | Mengirimkan data kelembapan tanah | **Kode Use Case:** UC-07  **Deskripsi Singkat:** Alat akan mengirimkan data setiap ada perubahan kelembapan tanah  **Aktor:** API Alat Siram Tanaman Otomatis  **Trigger:** Perubahan kelembapan tanah pada media tanam  **Prekondisi:** Alat mengirimkan data pada sistem  **Pascakondisi:** Alat mengedit data yang ada di database ketika terjadi perubahan. |
| 8 | Mengirimkan data suhu | **Kode Use Case:** UC-08  **Deskripsi Singkat:** Alat akan mengirimkan data setiap ada perubahan suhu di sekitar tanaman.  **Aktor:** API Alat Siram Tanaman Otomatis  **Trigger:** Perubahan suhu di sekitar tanaman  **Prekondisi:** Alat mengirimkan data pada sistem  **Pascakondisi:** Alat mengedit data yang ada di database ketika terjadi perubahan. |
| 10 | Menyalakan Alat Siram Tanaman | **Kode Use Case:** UC-10  **Deskripsi Singkat:** Alat akan mengirimkan data setiap ada perubahan suhu di sekitar tanaman.  **Aktor:** Admin  **Trigger:** Perubahan suhu di sekitar tanaman  **Prekondisi:** Alat mengirimkan data pada sistem  **Pascakondisi:** Alat mengedit data yang ada di database ketika terjadi perubahan. |

#### Use Case Scenario

Pada bagian ini akan diisi dengan skenario (flow of event) untuk tiap use case yang menggambarkan urutan interaksi aktor dengan use case tersebut dari awal sampai akhir. Untuk usecase scenario pertama, yaitu usecase scenario untuk usecase dengan kode UC-01 dapat dilihat pada tabel 3.18 berikut.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Login | |
| Goal In Context | Pengguna dapat masuk ke dalam sistem dengan menggunakan akun yang telah terdaftar | |
| Description | Fungsionalitas ini digunakan oleh pengguna untuk masuk atau login ke dalam sistem dengan menggunakan akun yang telah terdaftar di sistem | |
| Related Use Case | UC-02,UC-03,UC-04,UC-06, UC-08, UC-10. UC-12, UC11, UC-14 | |
| Successful End Condition | Pengguna berhasil masuk ke dalam sistem | |
| Failed End Condition | Pengguna tidak berhasil masuk ke dalam sistem | |
| Actors | Admin, Web Service | |
| Trigger | Pengguna menekan tombol masuk dan sistem mengirimkan request login menggunakan API | |
| Main Flow | Step | Action |
|  | 1 | Pengguna membuka aplikasi |
|  | 2 | Sistem menampilkan halaman login |
|  | 3 | Pengguna menekan tombol Login |
|  | 4 | Sistem mengecek apakah akun pengguna sesuai dengan database |
|  | 5 | Pengguna berhasil masuk ke dalam sistem |
|  | 6 | Sistem mengalihkan pengguna ke halaman utama |
| Extension | Step | Branching Action |
|  | 1.1 | Sistem menampilkan halaman untuk mengisi username dan password |

Untuk usecase scenario selanjutnya, yaitu usecase scenario untuk usecase dengan kode UC-02 dapat dilihat pada tabel 3.19 berikut.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Melihat Data Tanaman | |
| Goal In Context | Pengguna dapat melihat data tanaman | |
| Description | Fungsionalitas ini digunakan oleh pengguna untuk melihat kondisi dari tanaman. Sistem akan menampilkan data tanaman terkini apabila di dalam sistem telah ada data – data tanaman | |
| Related Use Case | UC-02,UC-07,UC-08,UC-09 | |
| Successful End Condition | Sistem berhasil menampilkan data – data tanaman | |
| Failed End Condition | Sistem tidak berhasil menampilkan data – data tanaman | |
| Actors | Admin, API | |
| Trigger | Admin telah masuk ke menu utama | |
| Main Flow | Step | Action |
|  | 1 | Admin masuk ke halaman utama |
|  | 2 | Sistem akan mengirimkan *request* ke API untuk meminta data – data tanaman yang berisikan nama tanaman, kelembapan tanah, suhu, intensitas cahaya |
|  | 3 | Sistem menampilkan data – data tanaman |
| Extension | Step | Branching Action |
|  | 2.1 | Apabila di basis data sistem belum terdapat data nama tanaman, jenis tanaman dan catatan, maka sistem akan menampilkan informasi |
|  | 2.2 | Pengguna akan dialihkan ke halaman Register apabila akun Google pengguna belum terdaftar di sistem |

Untuk usecase scenario selanjutnya, yaitu usecase scenario untuk usecase dengan kode UC-03 dapat dilihat pada tabel 3.19 berikut.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Siram Tanaman | |
| Goal In Context | Pengguna dapat melakukan penyiraman tanaman secara manual dengan menekan tombol pada halaman detail | |
| Description | Fungsionalitas ini digunakan oleh pengguna untuk melakukan penyiraman tanaman secara manual | |
| Related Use Case | UC-03 | |
| Successful End Condition | Kran berhasil menyala | |
| Failed End Condition | Kran tidak berhasil menyala | |
| Actors | Admin, API Alat Siram Tanaman Otomatis | |
| Trigger | Pengguna menekan tombol siram tanaman dan sistem mengirim parameter kran menyala ke API Alat Siram Tanaman Otomatis | |
| Main Flow | Step | Action |
|  | 1 | Pengguna masuk kedalam menu detail |
|  | 2 | Sistem menampilkan halaman detail |
|  | 3 | Pengguna menekan tombol Siram |
|  | 4 | Sistem mengirim parameter kran menyala ke API Alat Siram Tanaman Otomatis |
|  | 5 | Kran menyala |
|  | 6 | Sistem menampilkan informasi berhasil disiram |
| Extension | Step | Branching Action |
|  | 3.1 | Sistem menampilkan halaman untuk melakukan penyiraman tanaman. |
|  | 3.2 | Pengguna akan dialihkan ke halaman Register apabila akun Google pengguna belum terdaftar di sistem |

Untuk usecase scenario selanjutnya, yaitu usecase scenario untuk usecase dengan kode UC-04 dapat dilihat pada tabel 3.19 berikut.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Menambah Data Tanaman | |
| Goal In Context | Admin dapat menambahkan data tanaman yang terdiri dari nama tanaman, jenis tanaman dan catatan | |
| Description | Fungsionalitas ini digunakan oleh pengguna untuk menambahkan data tanaman agar masuk kedalam sistem | |
| Related Use Case | UC-03 | |
| Successful End Condition | Admin berhasil menambah data tanaman | |
| Failed End Condition | Admin gagal menambah data tanaman | |
| Actors | Admin | |
| Trigger | Admin mengisi form data tanaman pada halaman tambah tanaman | |
| Main Flow | Step | Action |
|  | 1 | Admin masuk ke menu tambah tanaman |
|  | 2 | Sistem menampilkan form |
|  | 3 | Admin mengisi form dan menekan tombol simpan |
|  | 4 | Sistem menampilkan pop-up data berhasil diinput |
|  | 5 | Data masuk ke dalam database |
| Extension | Step | Branching Action |
|  |  | Sistem menampilkan halaman untuk mengisi alamat email dan password apabila sebelumnya pengguna belum pernah melakukan login menggunakan akun Google di perangkat Androidnya |
|  |  | Pengguna akan dialihkan ke halaman Register apabila akun Google pengguna belum terdaftar di sistem |

Untuk usecase scenario selanjutnya, yaitu usecase scenario untuk usecase dengan kode UC-05 dapat dilihat pada tabel 3.19 berikut.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mengubah Data Tanaman | |
| Goal In Context | Admin dapat menghapus data tanaman | |
| Description | Fungsionalitas ini digunakan oleh Admin untuk mengubah data tanaman yang ada pada sistem | |
| Related Use Case | UC-03 | |
| Successful End Condition | Admin berhasil mengubah data tanaman | |
| Failed End Condition | Admin gagal mengubah data tanaman | |
| Actors | Admin | |
| Trigger | Admin menekan tombol ubah dan mengisikan form, lalu sistem akan mengubah data | |
| Main Flow | Step | Action |
|  | 1 | Admin membuka halaman utama |
|  | 2 | Admin memilih menu ubah tanaman |
|  | 3 | Sistem menampilkan form ubah tanaman |
|  | 4 | Admin memilih menu ubah |
|  | 5 | Sistem akan mengubah data tanaman |
| Extension | Step | Branching Action |
|  |  | Sistem menampilkan halaman untuk mengisi alamat email dan password apabila sebelumnya pengguna belum pernah melakukan login menggunakan akun Google di perangkat Androidnya |

Untuk usecase scenario selanjutnya, yaitu usecase scenario untuk usecase dengan kode UC-06 dapat dilihat pada tabel 3.19 berikut.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Menghapus Data Tanaman | |
| Goal In Context | Admin dapat menghapus data tanaman | |
| Description | Fungsionalitas ini digunakan oleh Admin untuk menghapus data tanaman yang ada pada sistem | |
| Related Use Case | UC-03 | |
| Successful End Condition | Admin berhasil menghapus data tanama | |
| Failed End Condition | Admin gagal menghapus data tanama | |
| Actors | Admin | |
| Trigger | Admin menekan tombol hapus dan sistem akan menghapus data | |
| Main Flow | Step | Action |
|  | 1 | Admin membuka halaman utama |
|  | 2 | Admin memilih menu hapus tanaman |
|  | 3 | Sistem menampilkan pop-up untuk meyakinkan admin akan menghapus data |
|  | 4 | Jika Admin memilih konfirmasi hapus |
|  | 5 | Sistem akan menghapus data tanaman |
| Extension | Step | Branching Action |
|  | 6.1 | Sistem gagal menghapus data tanaman |

Untuk usecase scenario selanjutnya, yaitu usecase scenario untuk usecase dengan kode UC-07 dapat dilihat pada tabel 3.19 berikut.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mengirimkan Data Intensitas Cahaya | |
| Goal In Context | API dapat mengirimkan data intensitas cahaya ke sistem | |
| Description | Fungsionalitas ini digunakan oleh Admin untuk mengetahui data intensitas cahaya pada tanaman | |
| Related Use Case | UC-03 | |
| Successful End Condition | API berhasil mengirimkan data intensitas cahaya | |
| Failed End Condition | API gagal mengirimkan data intensitas cahaya | |
| Actors | API Alat Siram Tanaman Otomatis | |
| Trigger | API akan mengirimkan data setiap 5 menit sekali ke sistem | |
| Main Flow | Step | Action |
|  | 1 | Alat mendeteksi intensitas cahaya |
|  | 2 | Alat mengirimkan data ke API |
|  | 3 | API menerima data dari alat |
|  | 4 | API mengirimkan data ke sistem |
|  | 5 | Sistem menampilkan data dari API ke Admin |
| Extension | Step | Branching Action |
|  | 7.1 | API gagal mengirimkan data ke sistem |
|  | 3.1 | Sistem gagal menampilkan data kepada Admin |

Untuk usecase scenario selanjutnya, yaitu usecase scenario untuk usecase dengan kode UC-08 dapat dilihat pada tabel 3.19 berikut.

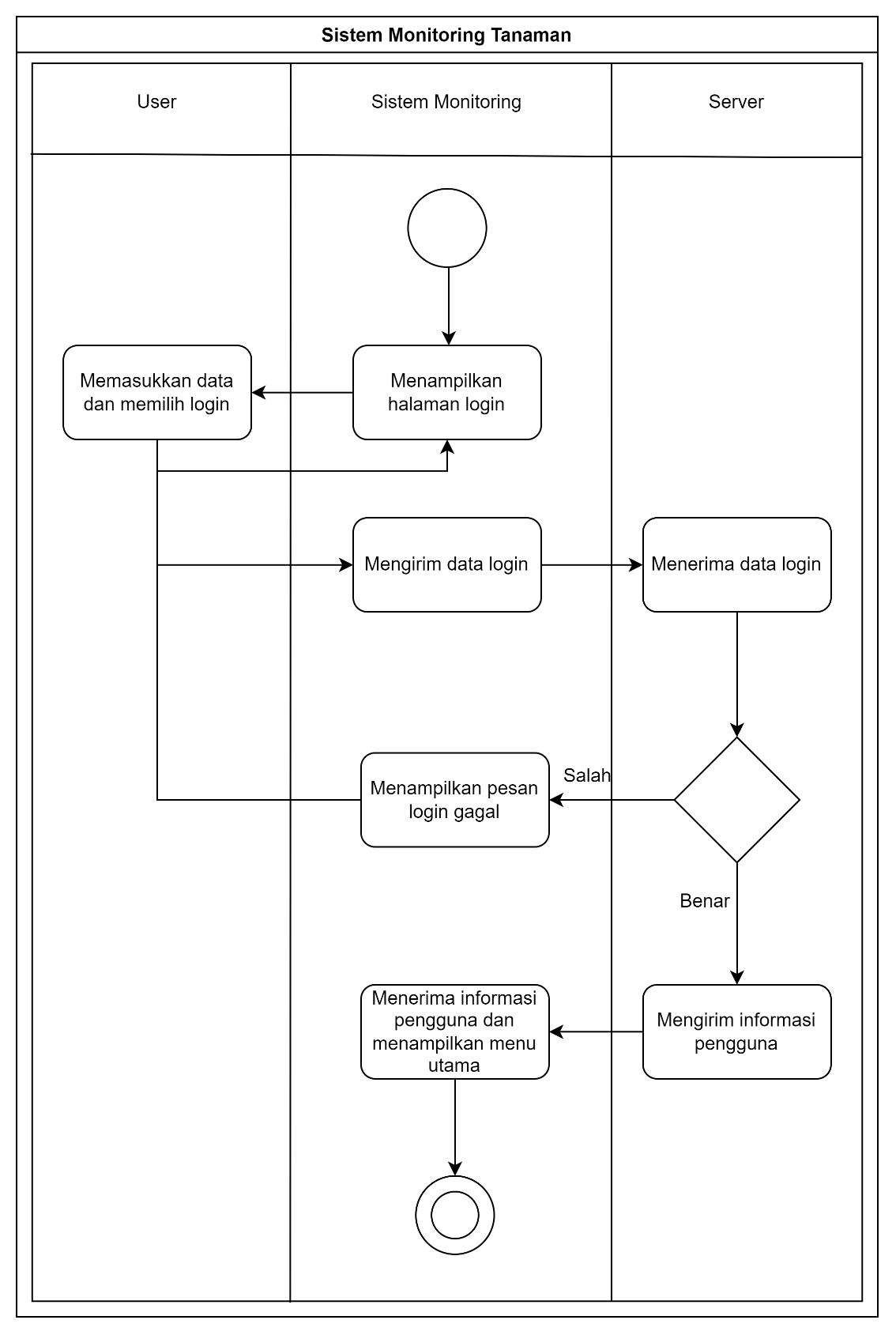
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mengirimkan Data Suhu | |
| Goal In Context | API dapat mengirimkan data suhu ke sistem | |
| Description | Fungsionalitas ini digunakan oleh Admin untuk mengetahui data suhu tanaman | |
| Related Use Case | UC-03 | |
| Successful End Condition | API berhasil mengirimkan data suhu | |
| Failed End Condition | API gagal mengirimkan data suhu | |
| Actors | API Alat Siram Tanaman Otomatis | |
| Trigger | API akan mengirimkan data setiap 5 menit sekali ke sistem | |
| Main Flow | Step | Action |
|  | 1 | Alat mendeteksi suhu |
|  | 2 | Alat mengirimkan data ke API |
|  | 3 | API menerima data dari alat |
|  | 4 | API mengirimkan data ke sistem |
|  | 5 | Sistem menampilkan data dari API |
| Extension | Step | Branching Action |
|  | 8.1 | API gagal mengirimkan data ke sistem |
|  | 3.1 | Sistem gagal menampilkan data kepada Admin |

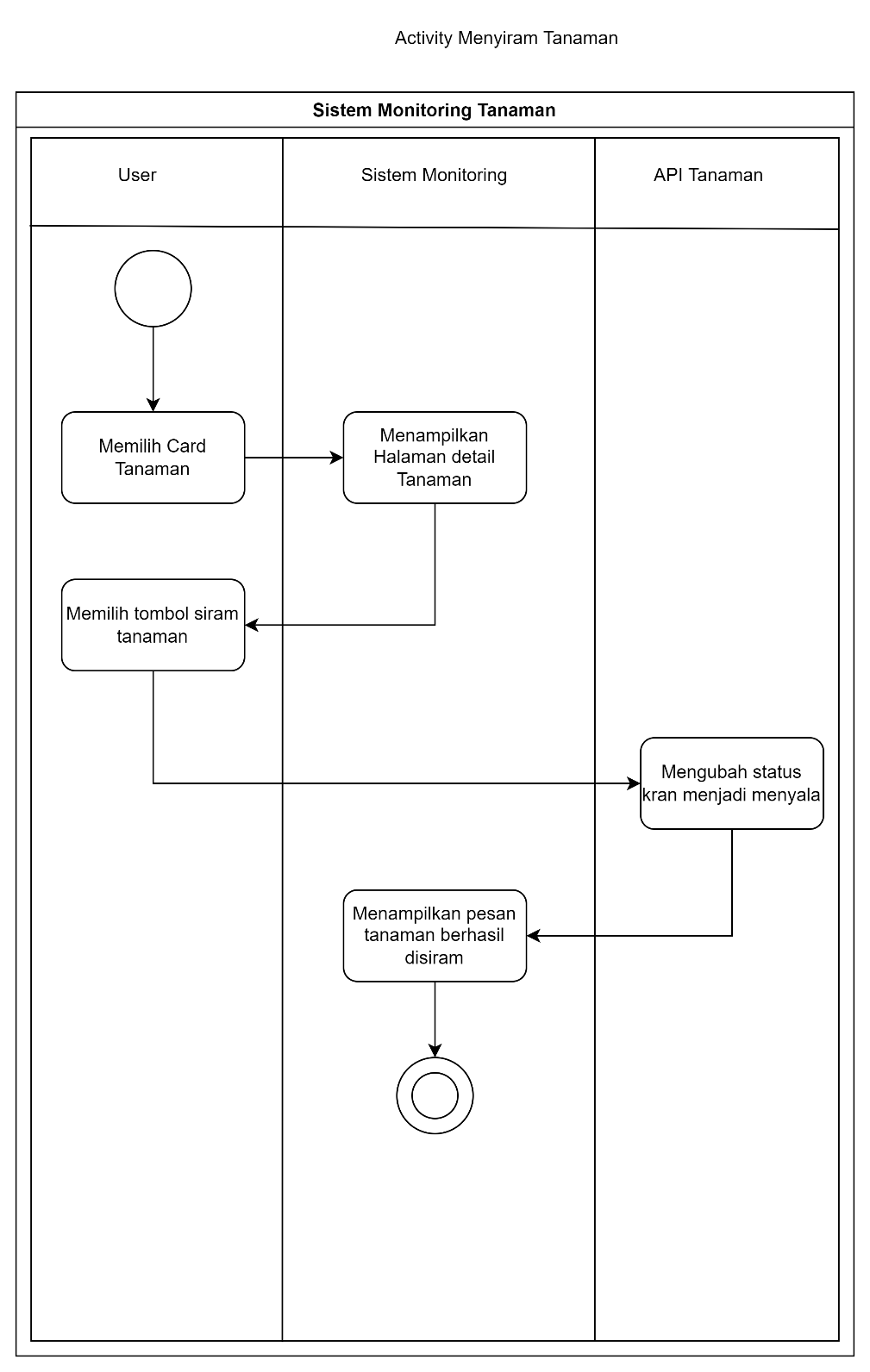
Untuk usecase scenario selanjutnya, yaitu usecase scenario untuk usecase dengan kode UC-09 dapat dilihat pada tabel 3.19 berikut.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Mengirimkan Data Kelembapan Tanah | |
| Goal In Context | API dapat mengirimkan data kelembapan tanah ke sistem | |
| Description | Fungsionalitas ini digunakan oleh Admin untuk mengetahui data kelembapan tanah tanaman | |
| Related Use Case | UC-03, UC-02 | |
| Successful End Condition | API berhasil mengirimkan data kelembapan tanah | |
| Failed End Condition | API gagal mengirimkan data kelembapan tanah | |
| Actors | API Alat Siram Tanaman Otomatis | |
| Trigger | API akan mengirimkan data setiap 5 menit sekali ke sistem | |
| Main Flow | Step | Action |
|  | 1 | Alat mendeteksi kelembapan tanah |
|  | 2 | Alat mengirimkan data ke API |
|  | 3 | API menerima data dari alat |
|  | 4 | API mengirimkan data ke sistem |
|  | 5 | Sistem menampilkan data dari API |
| Extension | Step | Branching Action |
|  | 9.1 | API gagal mengirimkan data ke sistem |
|  | 3.1 | Sistem gagal menampilkan data kepada Admin |

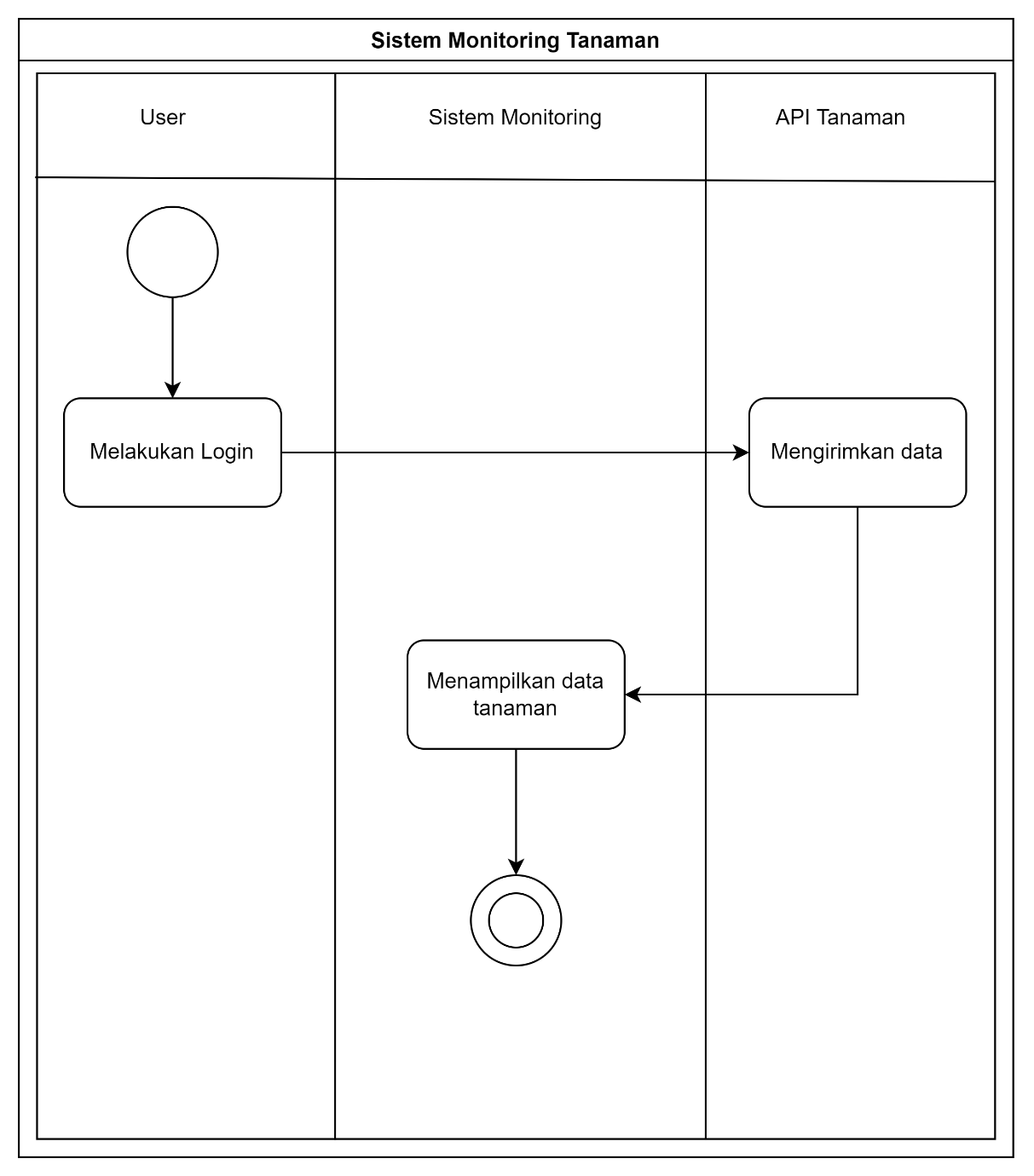
#### Activity Diagram

Activity diagram memodelkan aliran kerja atau workflow dari urutan aktivitas dalam suatu proses yang mengacu pada use case diagram yang ada. Untuk activity diagram untuk usecase dengan kode UC-01 dapat dilihat pada gambar 3.28 berikut.

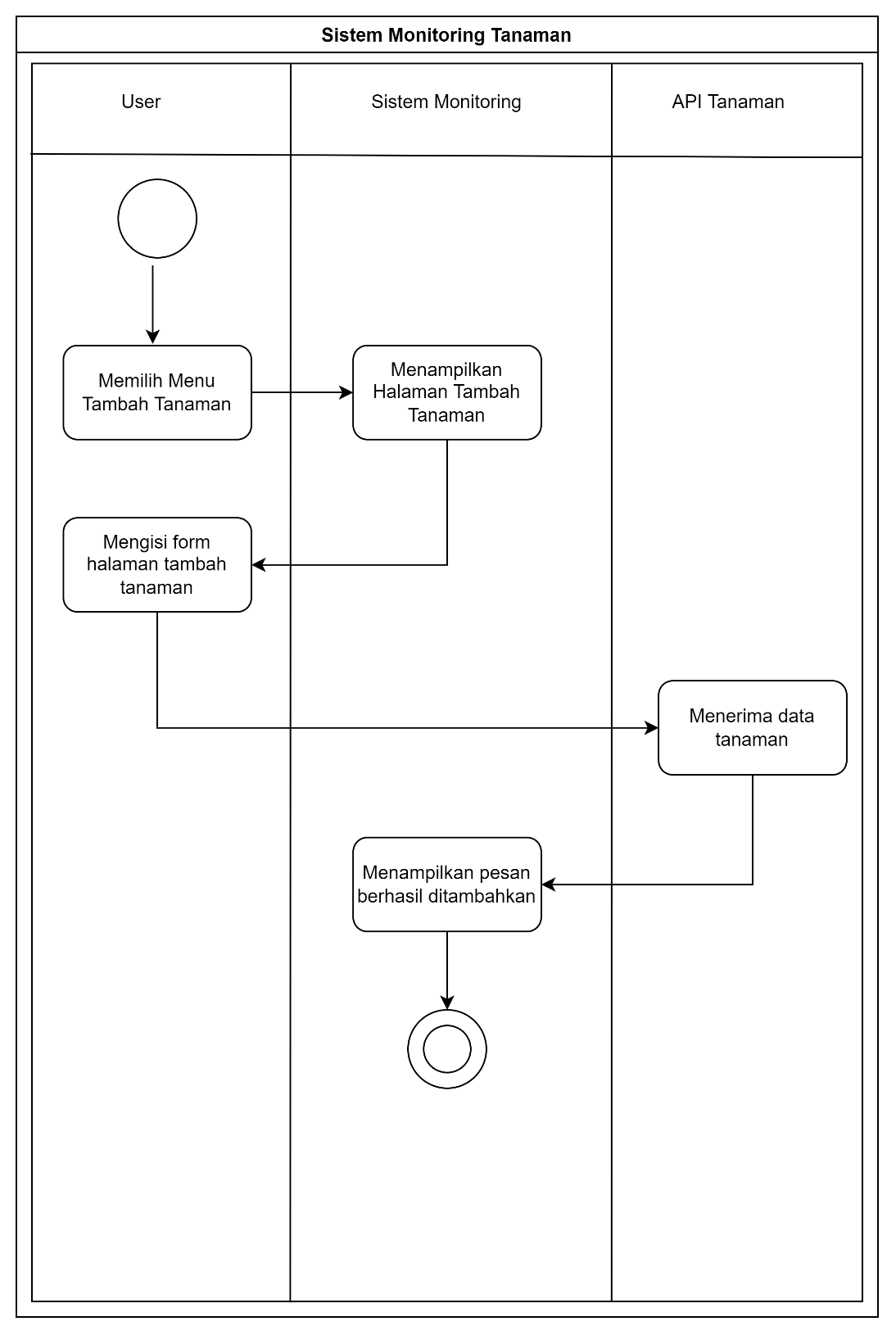
1. Activity Diagram untuk Use Case UC-01 Login
2. Activity Diagram Siram Tanaman



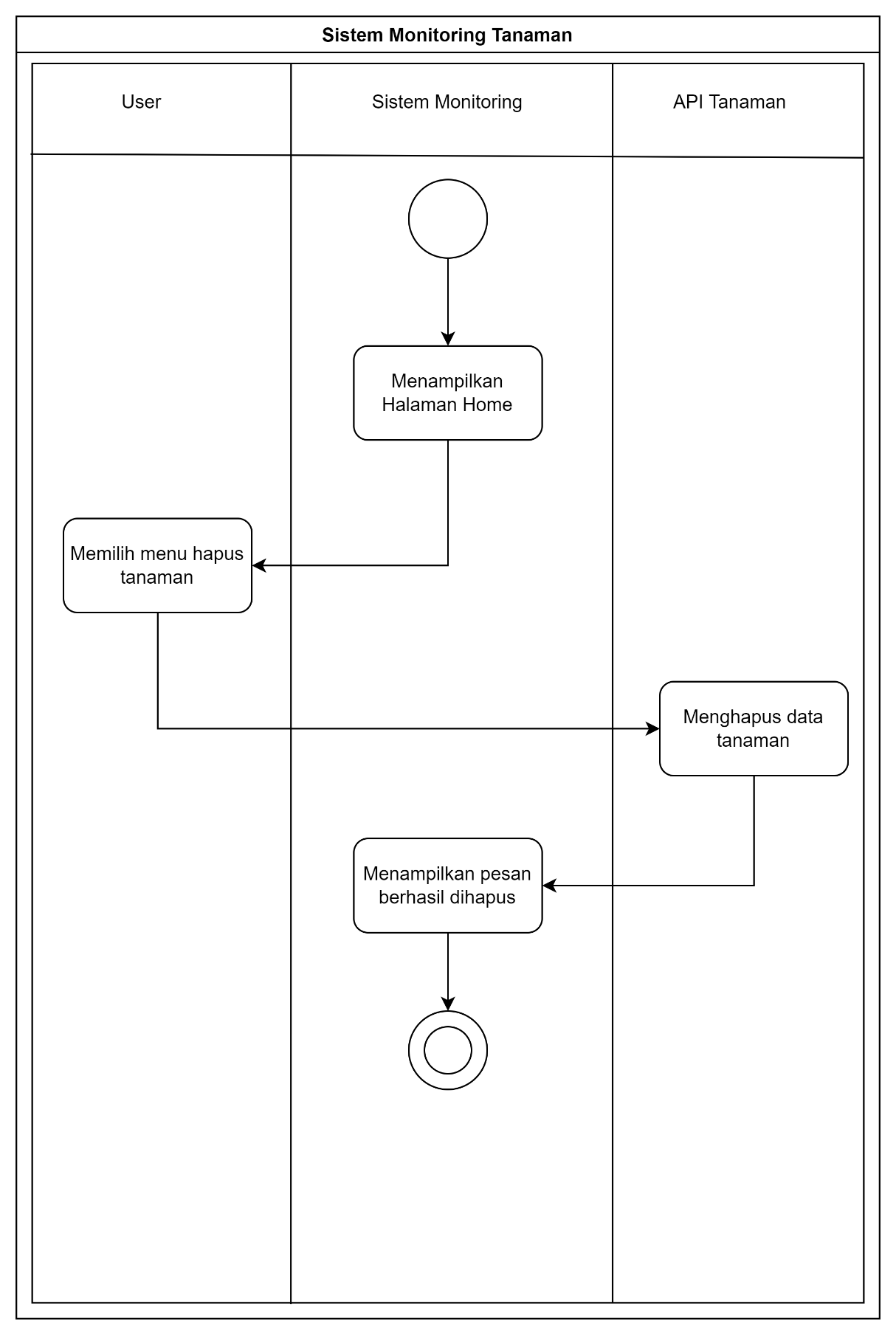
1. Activity Diagram Melihat Data Tanaman



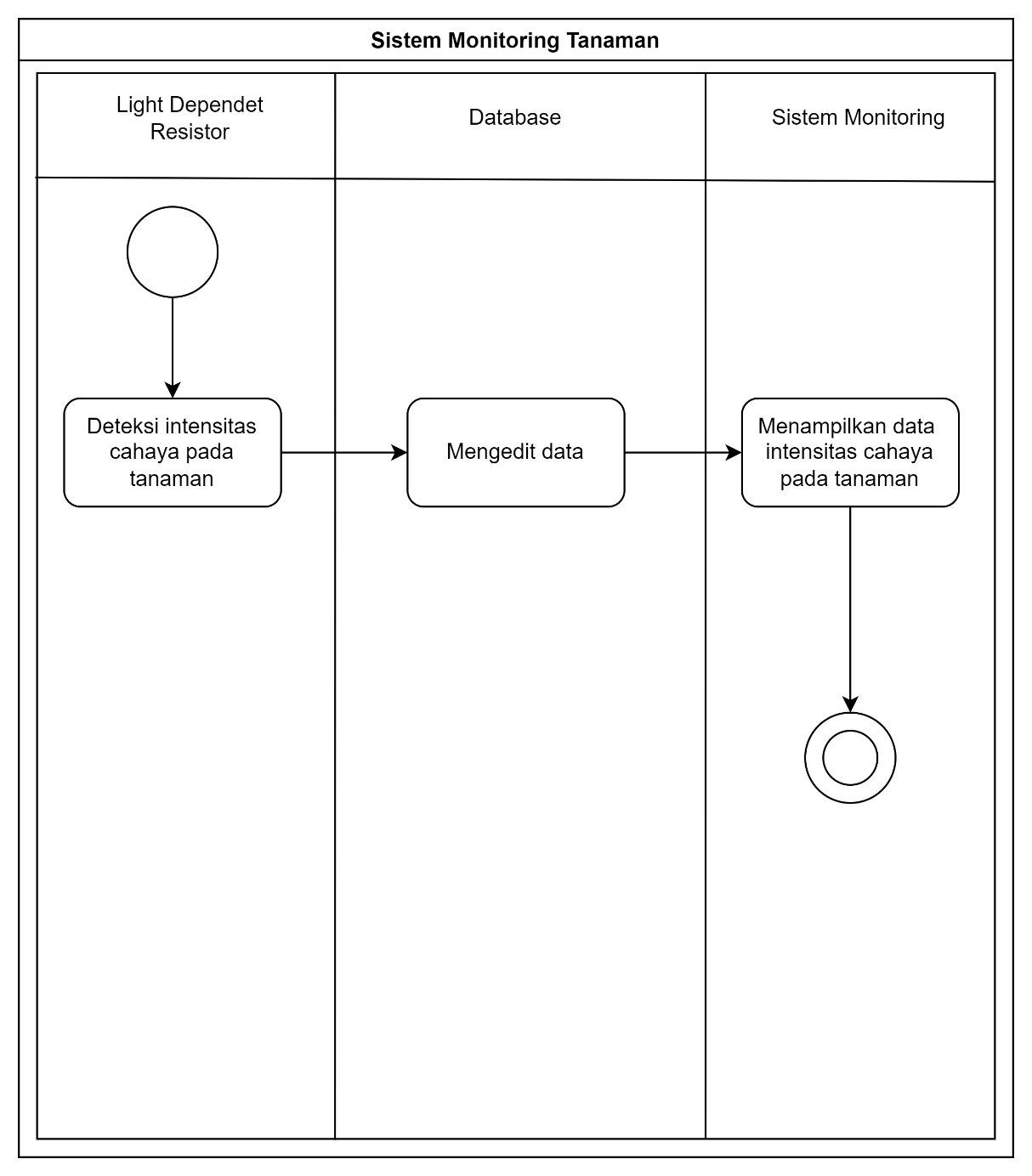
1. Activity Diagram Menambah Data Tanaman



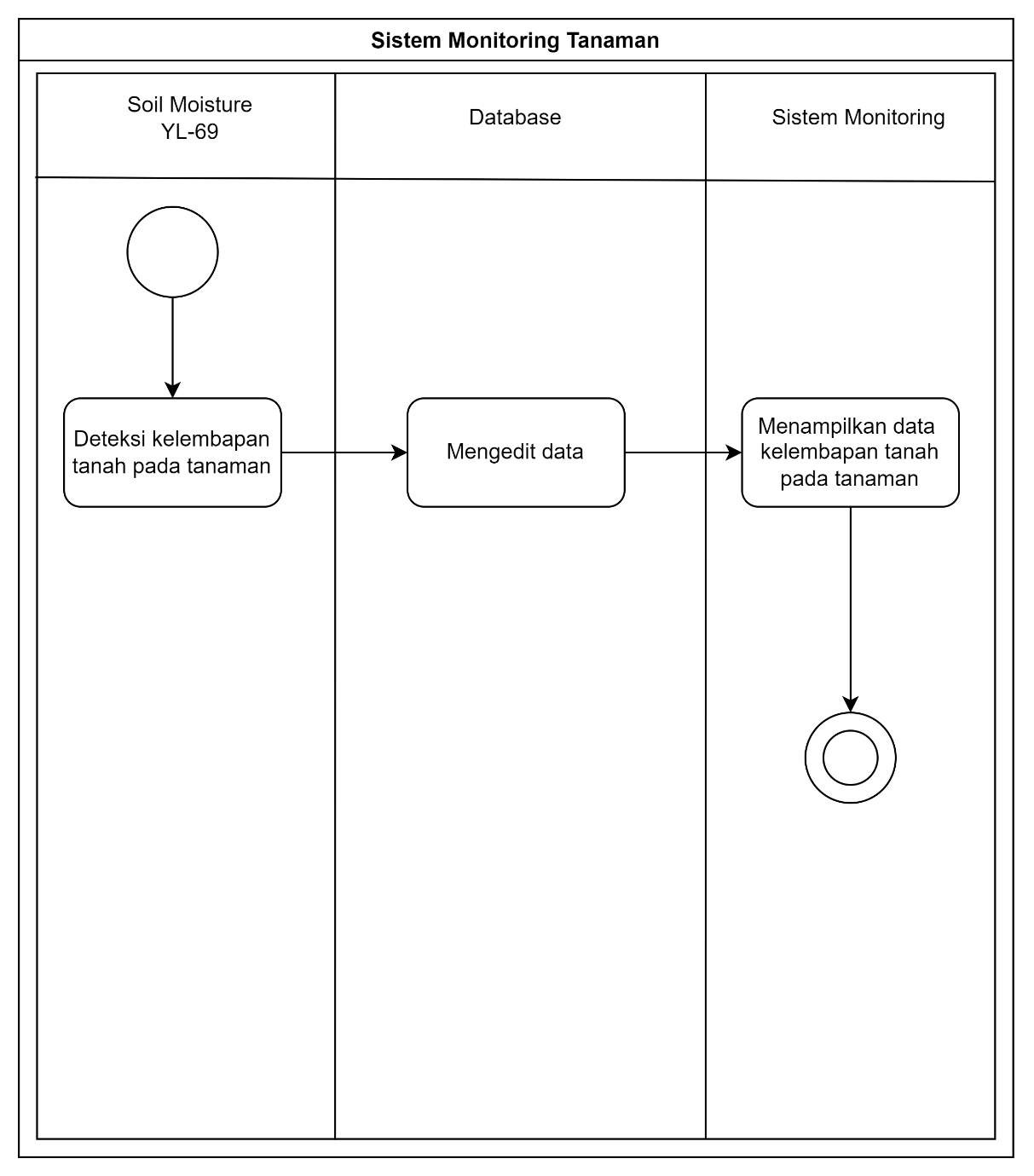
1. Activity Diagram Menghapus Data Tanaman



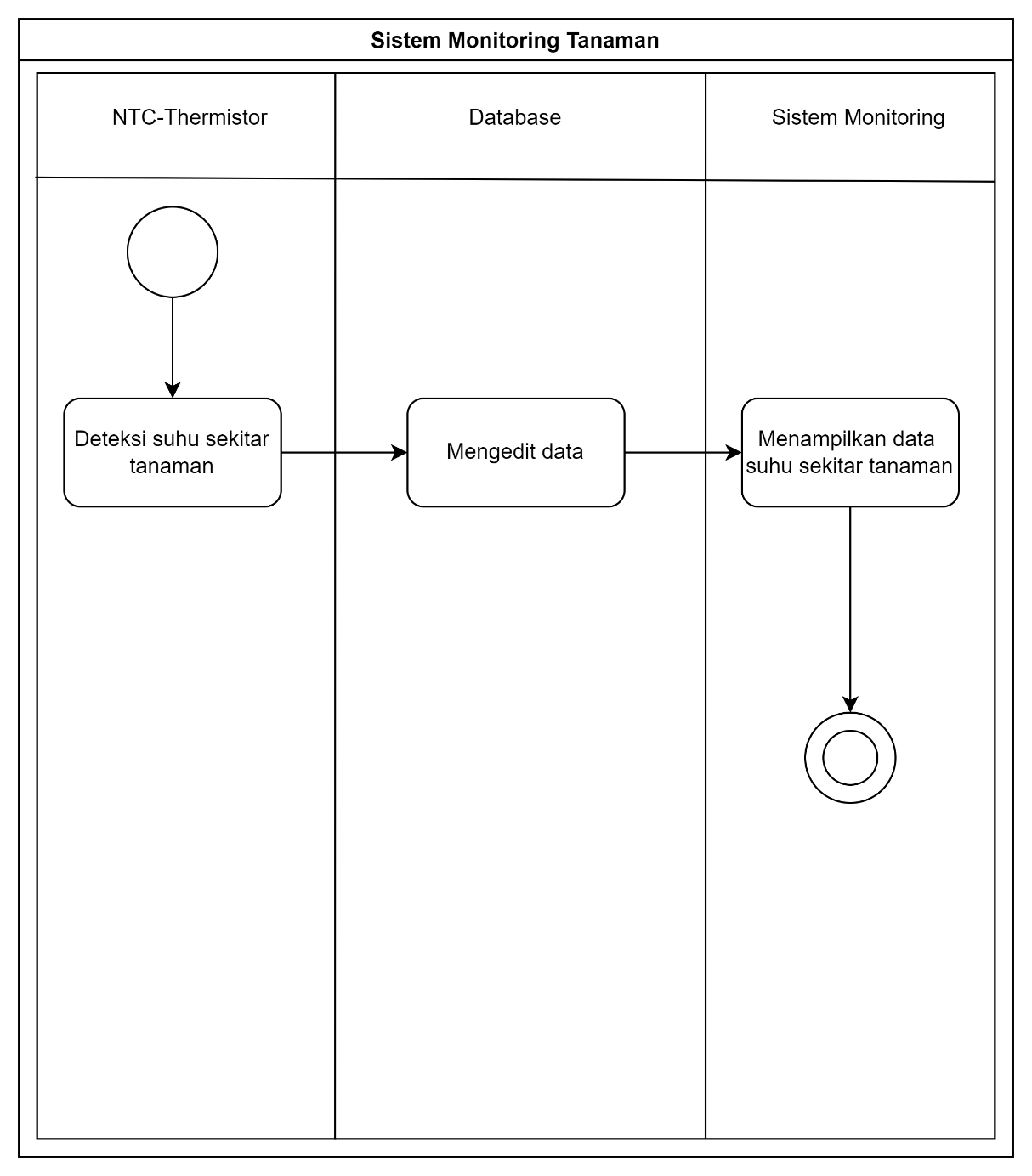
1. Activity Diagram Mengirim Data Intensitas Cahaya



1. Activity Diagram Mengirim Data Kelembapan Tanah



1. Activity Diagram Mengirim Data Suhu



1. Activity Diagram Menyiram Tanaman Otomatis

#### Class Diagram

Class Diagram dibuat untuk menunjukan hubungan antar class yang di dalamnya terdapat atribut dan fungsi atau metode dari suatu objek. Berikut adalah diagram kelas sistem yang akan dibangun dapat dilihat pada gambar 3.45 di bawah ini

#### Sequence Diagram

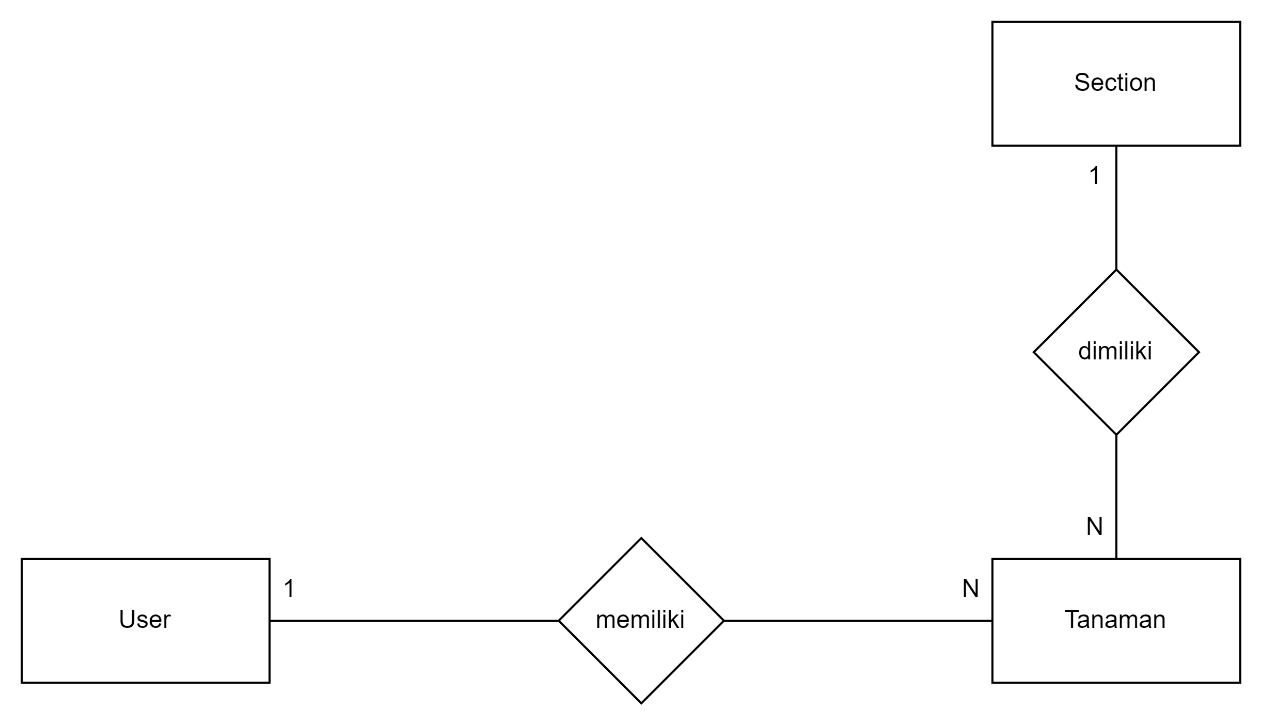
Sequence diagram menggambarkan pengaplikasian dari masing-masing skenario use case yang ada pada sistem. Bagaimana sebuah use case dapat berjalan ketika skenario benar terjadi. Untuk sequence diagram yang pertama, yaitu sequence diagram untuk usecase dengan kode UC-01 dapat dilihat pada gambar 3.46 berikut.

### Analisis Basis Data

Setelah berhasil melakukan analisis kebutuhan fungsional, maka tahap analisis basis data bisa dilakukan. Tahapan ini merupakan tahapan untuk menganalisis bagaimana nantinya skema basis data yang akan dibuat.

#### Skema ERD

Skema ERD dibuat dengan bertujuan untuk memudahkan peneliti dalam merancang basis data. Skema ERD pun dibuat berdasarkan dari data – data yang diperlukan untuk memberikan informasi – informasi tanaman kepada pengguna sistem. Untuk lebih jelasnya, bisa dilihat pada gambar 3.63



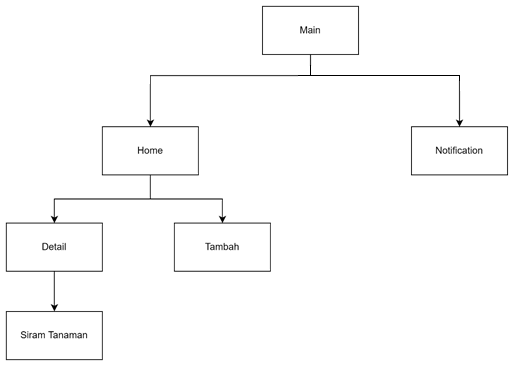
Gambar diatas hanya menunjukan nama tabel – tabelnya saja, tidak dengan atributnya apa saja. Untuk atributnya, penulis melampirkan dalam bentuk tabel pada tabel 3.12. Berikut adalah tabel 3.57 merupakan tabel detail atribut - atribut dari setiap tabel pada gambar ERD 3.63.

|  |  |
| --- | --- |
| Nama Tabel | Atribut |
| Tanaman | id\_tanaman  id\_section  username  nama\_tanaman  jenis\_tanaman  catatan  kelembapan\_tanah  suhu  intensitas\_cahaya |
| Section | id\_section  section |
| User | username  password  nama |

## Perancangan

### Perancangan Menu

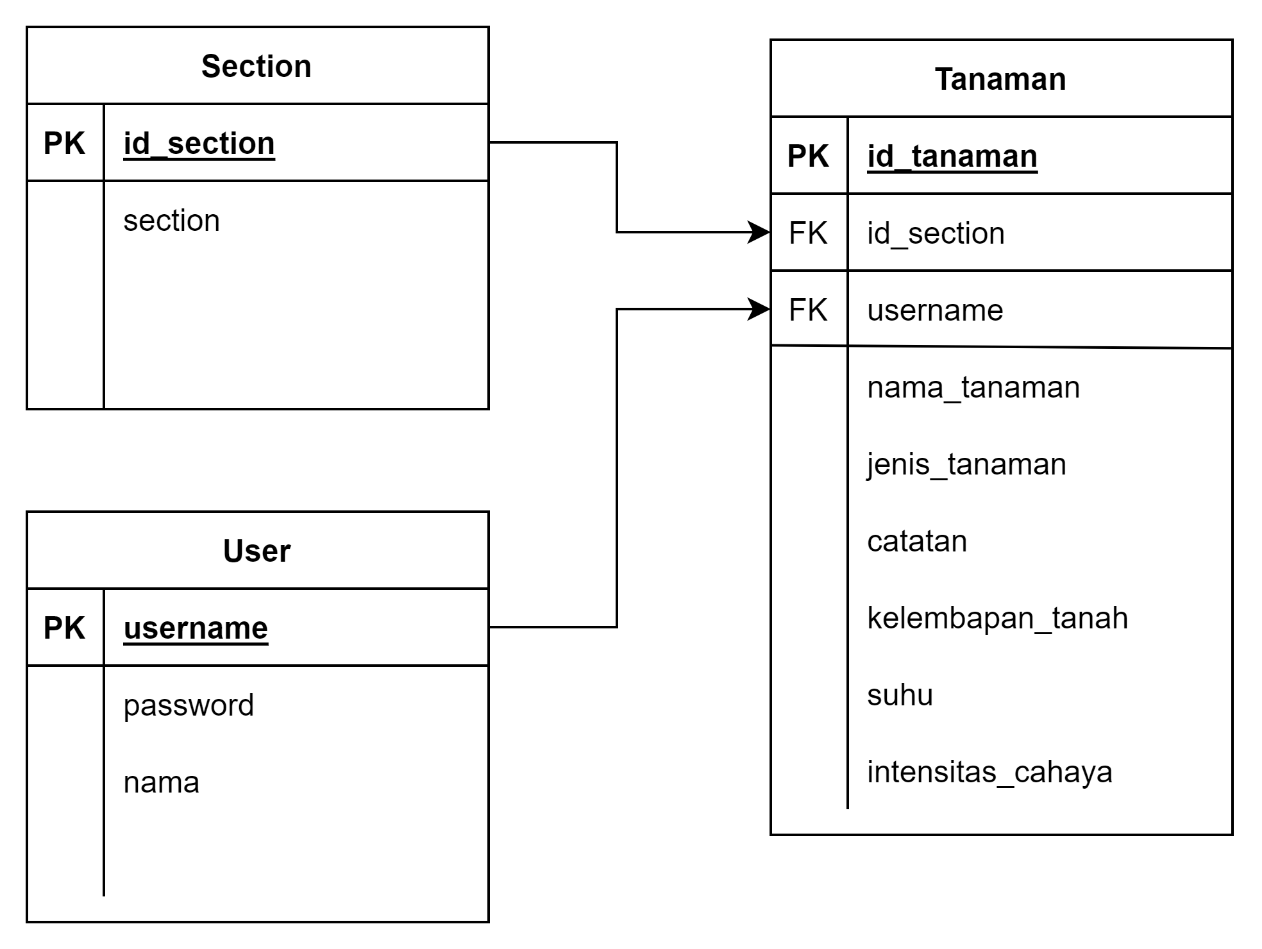
Sistem yang akan dibuat nantinya akan memiliki menu Home sebagai menu utama yang akan ditampilkan setelah login. Di menu Home tersebut, nantinya ada sub-menu lagi seperti menu notifikasi, menu detail tanaman, menu tambah tanaman, dan menu yang lainnya. Untuk lebih jelasnya bisa dilihat pada gambar berikut.



### Perancangan Basis Data

#### Skema Tabel Relasional

Setelah selesai merancang diagram ERD, selanjutnya penelitian merancang diagram tabel relasional. Diagram ini mirip seperti ERD, tetapi kini ada kumpulan nama – nama variabel di setiap tabelnya. Diagram ini nantinya untuk membantu peneliti dalam menentukan struktur di setiap tabel, seperti menentukan foreign key, ukuran variabel dan tipe data variabel. Untuk lebih jelasnya, dapat dilihat pada gambar berikut.



#### Struktur Tabel

Untuk menjelaskan lebih jelas mengenai isi tabel, seperti tipe data variabelnya, ukuran variabel, dan keterangan lainnya, dapat dilihat pada struktur tabel. Untuk struktur tabel tanaman, dapat dilihat pada tabel berikut.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama Data Item | Type | Size | Keterangan |
| 1 | id\_tanaman | Integer | 13 | PRIMARY KEY,  NOT NULL |
| 2 | id\_section | Integer | 13 | Foreign Key from Section,  NOT NULL |
| 3 | username | Varchar | 255 | Foreign Key from User,  NOT NULL |
| 4 | nama\_tanaman | Varchar | 255 | NOT NULL |
| 5 | jenis\_tanaman | Varchar | 255 | NOT NULL |
| 6 | catatan | Varchar | 255 | NOT NULL |
| 7 | kelembapan\_tanah | Integer | 13 | NOT NULL |
| 8 | suhu | Integer | 13 | NOT NULL |
| 9 | intensitas\_cahaya | Integer | 13 | NOT NULL |

Untuk struktur table section, dapat dilihat pada table berikut

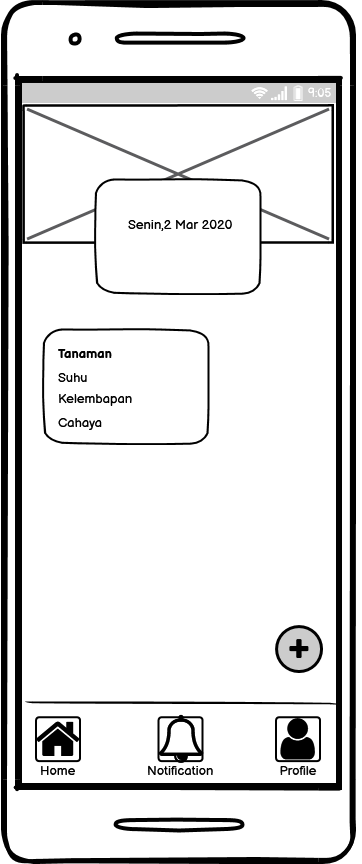
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama Data Item | Type | Size | Keterangan |
| 1 | id\_section | Integer | 13 | PRIMARY KEY,  NOT NULL |
| 2 | section | Varchar | 255 | NOT NULL |

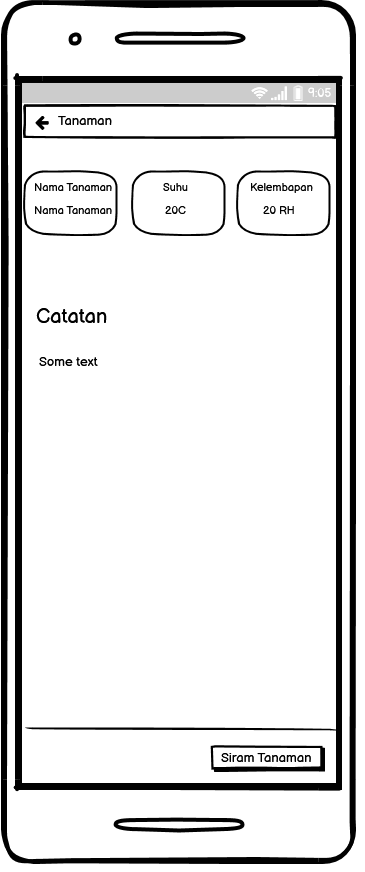
Untuk struktur table user, dapat dilihat pada table berikut

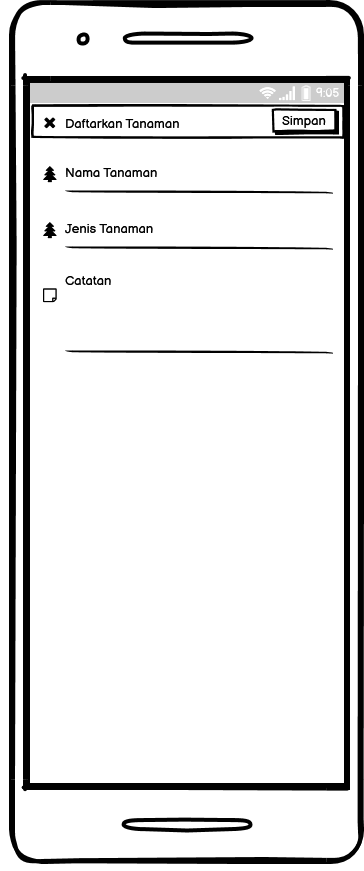
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama Data Item | Type | Size | Keterangan |
| 1 | username | varchar | 255 | PRIMARY KEY,  NOT NULL |
| 2 | password | Varchar | 255 | NOT NULL |
| 3 | nama | Varchar | 255 | NOT NULL |

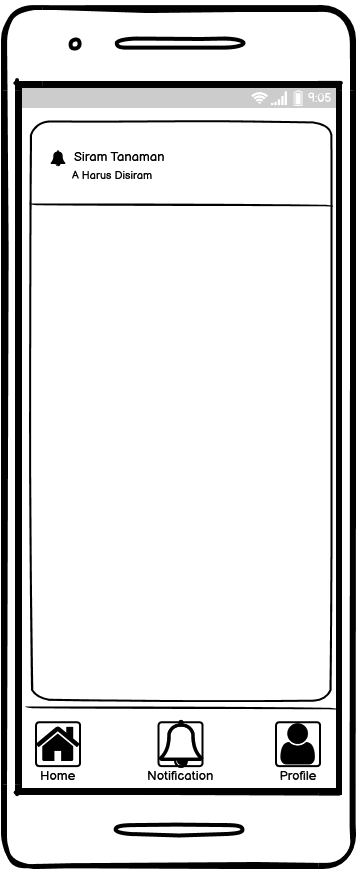
### Perancangan Antarmuka

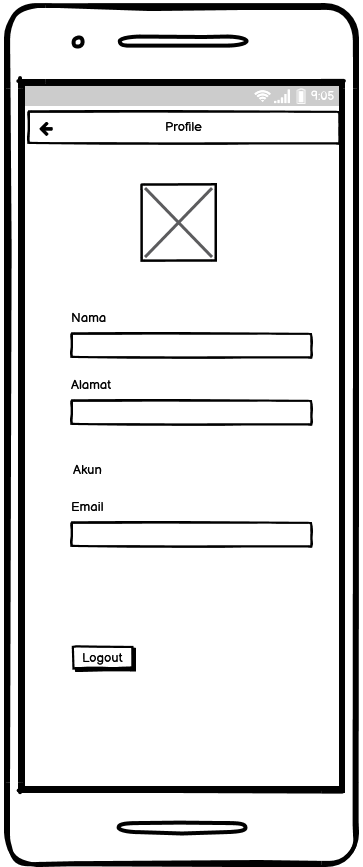
Perancangan antarmuka merupakan tahapan yang sangat penting. Tahapan ini merupakan tahapan mengimplementasikan hasil dari tahapan analisis ke dalam bentuk mockup antarmuka sistem. Berikut adalah rancangan antarmuka dari sistem ini. Gambar pertama dibawah, yaitu gambar 3.66 merupakan tampilan antarmuka dengan kode F01





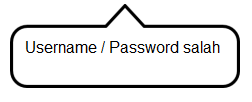




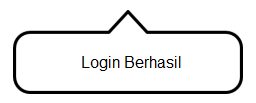


### Perancangan Pesan

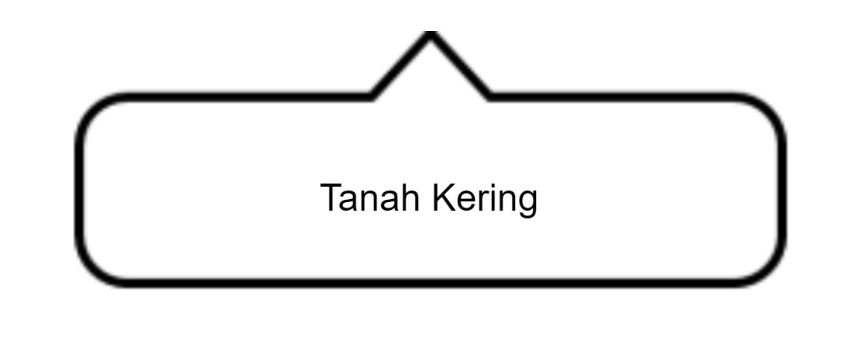
Perancangan pesan merupakan tahapan untuk merancang pesan apa saja yang akan ditampilkan oleh sistem kepada pengguna ketikan sedang menggunakan sistem. Pesan yang keluar dapat berupa pesan konfirmasi, pesan sukses, pesan gagal, atau sekedar pesan informasi saja. Untuk perancangan pesan yang pertama dengan kode P01, dapat dilihat pada gambar 3.93 berikut.



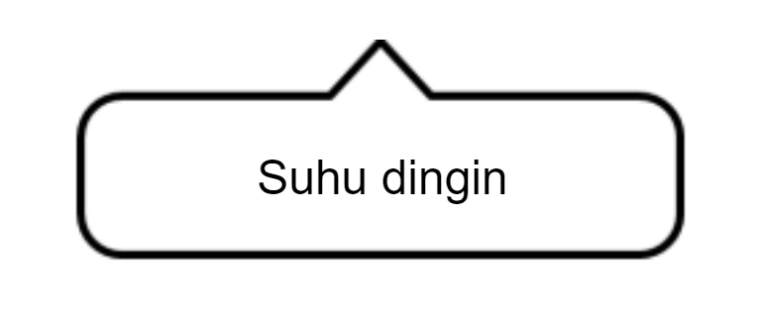
Untuk perancangan pesan berikutnya dengan kode P02, dapat dilihat pada gambar 3.94 berikut



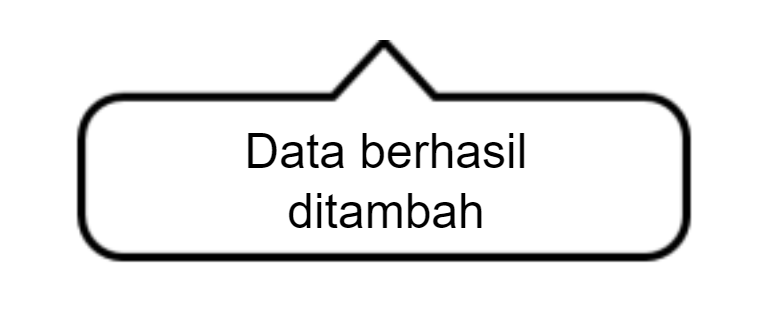
Untuk perancangan pesan berikutnya dengan kode P02, dapat dilihat pada gambar 3.94 berikut



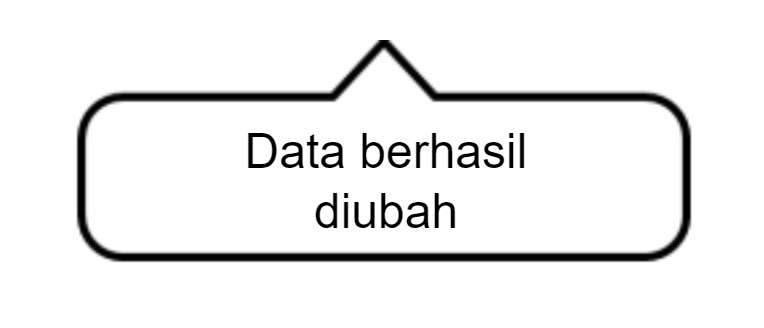
Untuk perancangan pesan berikutnya dengan kode P02, dapat dilihat pada gambar 3.94 berikut



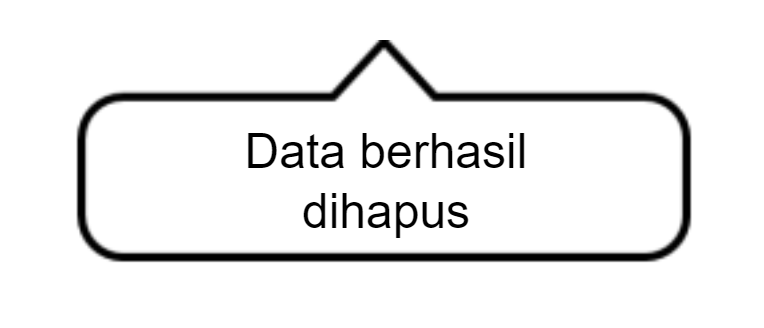
Untuk perancangan pesan berikutnya dengan kode P02, dapat dilihat pada gambar 3.94 berikut



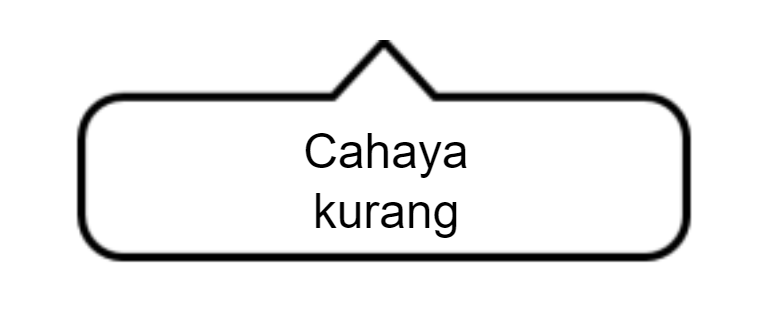
Untuk perancangan pesan berikutnya dengan kode P02, dapat dilihat pada gambar 3.94 berikut



Untuk perancangan pesan berikutnya dengan kode P02, dapat dilihat pada gambar 3.94 berikut



Untuk perancangan pesan berikutnya dengan kode P02, dapat dilihat pada gambar 3.94 berikut



### Perancangan Jaringan Semantik

Perancangan jaringan semantic merupakan tahapan untuk merancang gambar jaringan semantic. Jaringan semantic merupakan gambaran grafis yang menunjukan hubungan antar antarmuka berserta pesan yang akan muncul pada antarmuka tersebut. Adapun jaringan semantic pada sistem yang akan dibangun dapat dilihat pada gambar 3.2.5.1 berikut

# BAB IV IMPLEMENTASI DAN PENGUJIAN

## Implementasi

Implementasi merupakan tahap menerjemahkan perancangan pada tahap analisis dan perancangan. Tujuan implementasi adalah untuk mengkonfirmasi sistem yang telah dirancang sehingga pengguna dapat memberikan masukkan kepada peneliti. Implementasi sistem menjelaskan tentang kebutuhan perangkat lunak peneliti, perangkat keras peneliti, implementasi antarmuka, dan implementasi basis data.

### Perangkat Lunak Peneliti

Untuk perangkat lunak pada laptop yang peneliti pakai selama melakukan pembangunan dan pengujian sistem, dapat dilihat pada tabel 4.1 berikut

|  |  |  |
| --- | --- | --- |
| No | Nama | Spesifikasi |
| 1 | Sistem Operasi | Windows 10 Pro 64 Bit |
| 2 | *Tools* Yang Digunakan | Visual Studio Code |
| Draw.io |
| Balsamiq Mockup |
| MySQL |
| 3 | Browser | Google Chrome |

### Perangkat Keras Peneliti

Untuk perangkat keras pada laptop yang peneliti pakai selama melakukan pembangunan dan pengujian sistem, dapat dilihat pada tabel 4.2 berikut.

|  |  |  |
| --- | --- | --- |
| No | Nama Item | Spesifikasi |
| 1 | CPU | Intel Core i3-5005U |
| 2 | Memory | Single Channel DDR3 2GB |
| 3 | Storage | 500GB |
| 4 | GPU | Intel HD Graphic 5500 |

### Implementasi Antarmuka

Implementasi antarmuka menjelaskan tentang implementasi antarmuka sistem pada aplikasi. Implementasi antarmuka diwakili dengan nama 193 antarmuka beserta nama filenya. Implementasi antarmuka sistem dapat dilihat pada tabel 4.3 berikut.

|  |  |  |
| --- | --- | --- |
| No | Nama Antarmuka | Nama File |
| 1 | Halaman Login | Login.vue |
| 2 | Halaman Utama | Home.vue |
| 3 | Halaman Detail | Detail.vue |
| 4 | Halaman Tambah Tanaman | Create.vue |

### Implementasi Kelas

Implementasi kelas menggambarkan implementasi dari diagram kelas yang terbentuk dari usecase ke dalam bentuk file dengan ekstensi .blade.php. Berikut adalah implementasi kelas dapat dilihat pada tabel 4.4 berikut.

|  |  |  |
| --- | --- | --- |
| No | Nama Kelas | Nama File |
| 1 | Login | login.blade.php |
| 2 | Halaman Utama | index.blade.php |
| 3 | Halaman Detail | show.blade.php |
| 4 | Halaman Tambah Tanaman | create.blade.php |

### Implementasi Basis Data

Implementasi basis data merupakan implementasi rancangan basis data yang sudah dirancang sebelumnya yang merujuk kepada skema relasinya dan diimplementasikan menggunakan MySQL dengan source code untuk setiap tabel dapat dilihat pada tabel 4.5 berikut.

|  |  |  |
| --- | --- | --- |
| No | Nama Tabel | Query SQL |
| 1 | Tanaman | CREATE TABLE Tanaman (  id\_tanaman int NOT NULL PRIMARY KEY,  id\_section int NOT NULL,  username varchar(255) NOT NULL,  nama\_tanaman varchar(255) NOT NULL,  jenis\_tanaman varchar(255) NOT NULL,  catatan varchar(255) NOT NULL,  suhu int,  intensitas\_cahaya int,  kelembapan\_tanah int  ); |
| 2 | Section | CREATE TABLE Section (  id\_section int NOT NULL PRIMARY KEY,  section varchar(255) NOT NULL  ); |
| 3 | User | CREATE TABLE User (  username varchar(50) NOT NULL PRIMARY KEY,  password varchar(50) NOT NULL,  nama varchar(255) NOT NULL  ); |

### Implementasi Teknologi

Implementasi teknologi adalah pembahasan hasil dari beberapa teknologi yang digunakan dalam pembangunan sistem. Adapun teknologi yang digunakan beserta implementasinya adalah sebagai berikut ini.

#### Implementasi Axios

Implementasi Axios pada pembangunan sistem ini digunakan untuk membuat sistem dapat terhubung dengan sistem *backend* dan terhubung dengan API yang digunakan. Dengan menggunakan Axios, maka sistem yang dibangun akan dapat menerima data dari API yang digunakan. Ketika sistem berhasil terhubung dengan API, maka selanjutnya akan dipilih data – data apa saja yang akan diambil dari API. Semua itu bisa dilakukan dengan cara mengimplementasikan Axios kedalam sistem pada penelitian. Langkah pertama dalam mengimplementasikan Axios ke API dalam sistem, adalah dengan menginstall plugin Axios pada sistem yang dibangun. Adapun *syntax* yang harus dimasukkan untuk menginstall plugin Axios pada sistem, dapat dilihat pada gambar berikut



Setelah menginstall plugin Axios pada sistem, langkah berikutnya yaitu mengimplementasikan kode Javascript untuk Axios agar bisa digunakan dalam sistem. Kode Javascript untuk dihubungkan di sistem terdapat pada halaman Index.js yang terdapat pada *Router*. Untuk kode implementasinya, dapat dilihat pada skrip berikut

import axios from 'axios'

Vue.use(axios)

Potongan skrip diatas merupakan skrip Axios agar dapat terhubung dengan sistem. Untuk Penjelasan dari skrip diatas adalah sebagai berikut

1. Baris 1 melakukan *import* dari plugin Axios yang sudah diinstall
2. Baris 2 melakukan pemanggilan Axios dari *import* sebelumnya agar berfungsi didalam sistem

Setelah berhasil melakukan install plugin dan memanggil plugin pada sistem, langkah berikutnya melakukan pemanggilan data dari API menggunakan Axios, agar data dapat ditampilkan dalam sistem. Kode Javascript untuk menghubungkan data dengan sistem terdapat pada halaman utama Home.vue. Untuk kode implementasinya, dapat dilihat pada skrip berikut

<script>

import axios from 'axios'

export default {

methods: {

    getData() {

*let* uri = 'http://localhost:8000/api/tanaman'

      axios.get(uri).then((*response*) *=>* {

      this.tanamans = response.data.data

      })

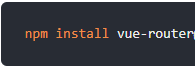
    },

}

Potongan skrip diatas merupakan skrip Axios agar data dari API dapat terhubung dengan sistem. Untuk penjelasan skrip di atas adalah sebagai berikut

#### Implementasi Vue-Router

Vue-Router termasuk kedalam teknologi yang dimiliki dari framework Vue.js , Vue-Router yang diimplementasikan pada sistem yang dibangun pada penelitian ini bertujuan agar sistem yang dibangun dapat berpindah halaman ketika pengguna menekan menu lainnya. Pertama, unutk dapat mengimplementasikan Vue-Router ke dalam sistem, lakukan instalasi Vue-Router pada sistem, adapun *syntax* yang digunakan untuk menginstall Vue-Router kedalam sistem, sebagai berikut



Setelah menginstall Vue-Router pada sistem, langkah berikutnya yaitu mengimplementasikan kode Javascript untuk Vue-Router agar bisa digunakan dalam sistem. Kode Javascript untuk dihubungkan di sistem terdapat pada halaman Index.js yang terdapat pada *Router*. Untuk kode implementasinya, dapat dilihat pada skrip berikut

1 import VueRouter from 'vue-router'

2 Vue.use(VueRouter)

*3 const* routes = [

4  {

5    path: '/',

6    name: 'Home',

7    component: () *=>* import(/\* webpackChunkName: "home" \*/ '@/views/Home.vue'),

8    meta: {

9      layout: 'home'

10    }

  },

  {

    path: '/detail/:id',

    name: 'Detail',

    component: () *=>*

      import(/\* webpackChunkName: "detail" \*/ '@/views/Detail.vue'),

    meta: {

      layout: 'home'

    }

  },

  {

    path: '/create',

    name: 'Create',

    component: () *=>*

      import(/\* webpackChunkName: "create" \*/ '@/views/Create.vue'),

    meta: {

      layout: 'home'

    }

  },

  {

    path: '/login',

    name: 'Login',

    component: () *=>* import(/\* webpackChunkName: "login" \*/ '@/views/Login.vue')

  }

]

Potongan skrip diatas merupakan skrip Vue-Router agar bisa digunakan dalam sistem, dan juga pendaftaran *Routes* dari setiap halaman yang akan dituju. Untuk penjelasan dari skrip di atas adalah sebagai berikut

1. Baris 1 melakukan import dari Vue-Router agar dapat digunakan dalam sistem
2. Baris 2 melakukan pemanggilan Vue-Router yang telah di import pada baris pertama
3. Baris 3 melakukan deklarasi dari *Routes* untuk setiap halaman
4. Baris 5 menentukan *path* dari setiap halaman yang didaftarkan
5. Baris 6 menentukan halaman yang dituju dari *path* yang sebelumnya ditentukan
6. Baris 7 memanggil halaman yang akan ditampilkan ketika memasukkan *path* yang dituju
7. Baris 8 – 10

#### Implementasi API Alat Siram Tanaman Otomatis

## Pengujian

Pada tahapan pengujian sistem dilakukan untuk memastikan apakah semua fungsi di dalam sistem yang telah dibangun dapat berjalan dengan baik sesuai dengan spesifikasi kebutuhan dan mencari kesalahan yang mungkin terjadi pada sistem. Pengujian yang dilakukan pada sistem yang dibangun adalah pengujian alpha secara fungsional. Metode pengujian yang digunakan adalah metode pengujian blackbox yang berfokus pada persyaratan fungsional dari sistem yang dibangun.

### Rencana Pengujian Alpha

Untuk rencana pengujian pada sistem yang dibangun, dapat dilihat pada tabel 4.8 berikut.

# BAB V KESIMPULAN DAN SARAN

## Kesimpulan

Berdasarkan dari hasil implementasi, pengujian, serta wawancara kepada pengguna yang telah dilakukan, maka didapatkan beberapa kesimpulan sebagai berikut :

## Saran