

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi saat ini telah menjadi kebutuhan pokok dan komoditas baru. Teknologi informasi berperan dalam pengelolaan data menjadi suatu informasi tepat cepat dan akurat, sehingga informasi yang diharapkan dapat mempermudah dan memperlancar pekerjaan atau kegiatan dan tujuannya dapat tercapai secara optimal. Salah satu teknologi yang digunakan untuk mendapatkan informasi atau pengetahuan secara efisien adalah website. Teknologi informasi ini dapat digunakan diberbagai bidang salah satunya di bidang pertanian tanaman kopi, sehingga teknologi informasi ini dapat di terapkan oleh petani.

Kopi merupakan salah satu jenis tanaman perkebunan yang sudah lama di budidayakan dan memiliki nilai ekonomis yang lumayan tinggi. Kopi berasal dari Afrika, yaitu daerah pegunungan di Eropia. Namun, kopi sendiri baru dikenal oleh masyarakat dunia setelah tanaman tersebut di kembangkan diluar daerah asalnya, yaitu Yaman di bagian selatan Arab.

Tanaman kopi merupakan salah satu produk pertanian yang memiliki citarasa yang khas sehingga perlu di perhatikan kualitas agar budidaya kopi memiliki kualitas produksi yang baik. Namun saat ini petani memiliki keluhan seperti kualitas produksi mutu hasil panen kopi rendah, penurunan produktivitas kopi salah satunya hama dan penyakit. Serangan hama penggerek biji kopi ini merupakan pengaruh ekonomi petani kopi. rendahnya kualitas hasil panen

disebabkan karena rusaknya biji kopi yang disebabkan oleh hama dan penyakit. Hama dan penyakit tersebut dapat menyerang biji kopi yang ada di perkebunan maupun biji yang sudah ada di penyimpanan. Salah satu daerah penghasil kopi adalah Desa Basseang, tanaman kopi merupakan sumber penghasilan masyarakat setempat, akan tetapi terdapat masalah pada tanaman kopi tersebut munculnya hama dan penyakit yang merusak tanaman kopi.

Kurangnya informasi yang diketahui oleh pihak perkebunan kopi tentang jenis hama dan penyakit yang ada pada tanaman kopi sehingga tidak tertangani dengan benar. Hal ini mengakibatkan banyak tanaman kopi yang seharusnya bisa terselamatkan menjadi mati dan kualitas kopi tersebut menurun. Jika hal ini dibiarkan terus menerus, maka akan mempengaruhi tingkat produktivitas tanaman kopi tersebut.

Penerapan sistem informasi berbasis web tentunya akan mempermudah petani dalam mendapatkan informasi tentang pengendalian hama penyakit melalui sistem informasi ini petani akan mudah mendapatkan informasi dan mudah diakses oleh petani cepat dan efisien. Berdasarkan masalah yang sudah dijelaskan maka penulis memilih judul “SISTEM INFORMASI HAMA PENYAKIT TANAMAN KOPI BERBASIS WEB DI DESA BASSEANG” yang diharapkan dapat memudahkan dalam mendapatkan sebuah informasi pengendalian hama penyakit pada tanaman kopi.

B. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas maka rumusan masalah yang akan dibahas yaitu: “bagaimana cara pembuatan sistem

informasi mengenai cara pengendalian hama penyakit pada tanaman kopi di Desa Basseang agar masyarakat lebih mudah dalam mendapatkan informasi.

C. Batasan Masalah

1. Aplikasi ini tidak memiliki fitur pendeteksi hama dan penyakit pada tanaman kopi
2. Bahasa pemrograman yang digunakan adalah PHP, HTML, MySQL, XAMPP.

D. Tujuan Penelitian

Tujuan dari penelitian yang dilakukan yaitu:

Untuk mengetahui cara pembuatan sistem informasi mengenai cara pengendalian hama penyakit pada tanaman kopi di Desa Basseang agar masyarakat lebih mudah dalam mendapatkan informasi.

E. Manfaat Penelitian

Adapun manfaat yang dapat diperoleh dari penelitian ini adalah:

1. Bagi Penulis

Agar dapat mengaplikasikan ilmu dan teori yang telah di dapatkan dibangku perkuliahan, serta dapat menambah wawasan tentang Sistem Informasi berbasis *web* khususnya dalam hal informasi pengendalian hama penyakit pada tanaman kopi.

2. Bagi Masyarakat

Masyarakat dapat lebih mudah mendapatkan informasi mengenai cara pengendalian hama penyakit pada tanaman kopi.

BAB II

KAJIAN PUSTAKA

A. Kajian Hasil Penelitian Terdahulu

1. Julia, M. U. (2022). Pengendalian Hama dan Penyakit Pada Tanaman Kopi Robusta (*Coffea canephora Linn*). Di Afd Gentong Perusahaan Daerah Perkebunan (PDP) Kahyangan JemberKebun Gunung. Politeknik Jember. Tujuan dilaksanakannya kegiatan Praktek Kerja Lapangan (PKL) mahasiswa dibor untuk mengerjakan pekerjaan lapangan sekaligus memantapkan keterampilan mengenai teknis budidaya tanaman kopi dengan baik. Pengendalian hama dan penyakit di PDP Kebun Gunung Pasang menggunakan metode pengendalian secara teknis dan mekanis.
2. Riansyah, B. (2019) Sistem Informasi Dan sistem Pakar Diagnosa Penyakit tanaman Kopi. Universitas Lampung. Penelitian ini membahas mengenai sistem informasi dan sistem pakar diagnosa tanaman kopi berisi informasi mengenai pengelolaan tanaman mulai dari penanaman sampai panen. Sistem yang dibangun ini berbasis *web*, dengan adanya sistem ini diharapkan dapat membantu memberikan solusi dalam menangani tanaman kopi yang terserang penyakit, sehingga tanaman kopi banyak yang terselamatkan dari penyakit dan dapat meningkatkan produktifitas tanaman kopi.
3. Wahyudi, J., Elfianty, L., Nur'aini, H., & Andriani, E. (2017). Sistem Informasi Penanggulangan Hama Dan Penyakit Tanaman Bagi Penyuluh Pertanian. Universitas Dehasen Bengkulu. Tujuan dari ini adalah terbentuknya perangkat

lunak (*software*) yang memiliki fungsi sebagai bahan ajar bagi para penyuluh yang akan terjun kelapangan dalam proses bimbingan dan penyuluhan penanggulangan hama dan penyakit tanaman. Berdasarkan pada kenyataan di atas, maka yang perlu mendapat perhatian nantinya adalah tingkat kelengkapan data dan informasi yang berkenaan dengan persoalan berbagai hama, berbagai penyakit yang menyerang pada tanaman dan sekaligus tata cara penanggulangannya.

B. Kajian Teori

1. Sistem Informasi

Sistem informasi adalah sekumpulan subsistem yang saling berhubungan, berkumpul bersama-sama dan membentuk satu kesatuan, saling berintegrasi dan bekerjasama antara bagian satu dengan yang lainnya dengan cara tertentu untuk melakukan fungsi pengolahan data, menerima masukan (*input*) berupa data-data, kemudian mengolahnya (*processing*), dan menghasilkan keluaran (*output*) berupa informasi sebagai dasar pengambilan keputusan yang berguna dan mempunyai nilai nyata yang dapat dirasakan akibatnya baik pada saat itu juga maupun disaat mendatang, mendukung kegiatan operasional, manajerial, dan strategis organisasi, dan memanfaatkan berbagai sumber daya yang ada dan tersedia bagi fungsi tersebut guna mencapai tujuan. Dari uraian beberapa para ahli informasi dapat disimpulkan sistem informasi adalah berupa proses pengolahan data yang menghasilkan berupa informasi yang berfungsi untuk mencapai tujuan. (Nasution et al., 2022)

2. Pengertian Kopi

Kopi merupakan salah satu komoditi dari subsektor perkebunan yang memegang peranan penting bagi perekonomian nasional khususnya sebagai sumber devisa, penyedia lapangan kerja dan sebagai sumber pendapatan bagi petani maupun bagi pelaku ekonomi lainnya yang terlibat dalam budidaya, pengolahan dan pemasaran hasil kopi, terutama di daerah-daerah sentra produksi kopi seperti Sumatera Selatan, Lampung, Sumatera Utara dan Jawa Timur. Pada umumnya perkebunan kopi rakyat belum dikelola secara baik seperti pada perkebunan besar sehingga berbagai masalah muncul salah satunya yaitu masalah produktivitas. Produktivitas yang tinggi akan dicapai apabila semua faktor produksi dialokasikan secara optimal

a. Ciri-Ciri Tanaman Kopi

Ciri-ciri kopi yang dikenal sebagai tanaman kopi dapat ditemukan tumbuh dan dibudidayakan di perkebunan, mulai dari dataran rendah hingga ketinggian 1.400 meter di atas permukaan laut. Habitat kopi adalah tanah yang sedikit asam, tetapi kaya humus, suhu rendah, kelembapan yang tinggi, dan terkena sinar matahari yang cukup. Tanaman kopi memiliki dua tipe pertumbuhan cabang, yaitu tumbuh ke arah cabang (*ortotrop*) dan ke arah *horizontal* (*plagiotrop*). Daun tanaman kopi berwarna hijau mengkilap yang tumbuh berpasangan dengan berlawanan arah. Bentuk daun tanaman kopi lonjong dengan tulang daun yang tegak. Tanaman kopi membutuhkan waktu selama 3 tahun dari saat perkecambahan sampai menjadi tanaman berbunga dan menghasilkan buah kopi.

b. Jenis-Jenis Kopi

Indonesia memiliki empat jenis kopi yang dikenal, yaitu kopi arabika, kopi robusta, kopi liberika, dan kopi ekselsa. Jenis kopi yang dikenal memiliki nilai ekonomis dan diperdagangkan secara komersial yaitu kopi arabika dan kopi robusta. Kopi arabika memiliki kualitas cita rasa tinggi dan kadar kafein lebih rendah dibandingkan dengan robusta sehingga harganya lebih mahal. Kopi liberika dan kopi ekselsa dikenal kurang ekonomis dan komersial karena memiliki banyak variasi bentuk dan ukuran biji serta kualitas cita rasanya.

Penelitian ini menggunakan kopi arabika, kopi robusta, kopi liberika, dan kopi lokal khas kampung Jatiarjo yaitu kopi jawa.

a. Kopi Arabika (*Coffea arabica*)

Kopi arabika adalah kopi pertama yang dikenal dan dikembangkan di dunia. Kopi arabika merupakan tipe kopi tradisional dengan citarasa terbaik. Saat ini telah menguasai sebagian besar pasar kopi dunia dan harganya jauh lebih tinggi daripada jenis kopi lainnya. Anatomi buah kopi arabika terdiri dari kulit luar, kulit ari, daging buah, kulit tanduk. Kopi arabika tumbuh pada ketinggian 600- 2000 meter diatas permukaan laut.

b. Kopi Robusta (*Coffea robusta*)

Kopi robusta dikatakan sebagai kopi kelas dua karena rasanya lebih pahit, sedikit asam, mengandung kadar kafein yang jauh lebih banyak, dan harganya lebih murah. Kualitas buah ini lebih rendah dari kopi arabika dan liberika. Kopi jenis ini lebih resisten terhadap serangan hama dan penyakit. Anatomi buah kopi robusta terdiri dari kulit luar, kulit ari, daging buah, kulit tanduk. Kopi robusta

dapat tumbuh baik di ketinggian 400-700 meter diatas permukaan laut. (Murtadho, 2022).

3. Hama dan Penyakit Tanaman Kopi

Hama tanaman adalah makhluk hidup pengganggu berupa hewan yang umumnya dapat dilihat dengan mata telanjang. Hama merusak tanaman dengan berbagai cara misalnya dengan memakan daun, melubangi dan membuat korok-korok pada batang, melubangi dan membuat korok-korok pada daun, mengisap cairan tanaman, dan memakan bunga dan bagian-bagian bunga dan sebagian. (Wahyudi et al., 2017)

Hama merupakan organisme yang merusak tanaman, mulai dari memakan bagian tanaman (akar, batang, cabang dan buah) sampai menghisap cairan dari jaringan tanaman tersebut. Penyakit tanaman adalah proses gangguan fisiologis pada tanaman yang disebabkan mikroorganisme patogenik seperti bakteri, cendawan, dan virus. Adapun deskripsi hama dan penyakit sebagai berikut:

a. Penggerek Buah Kopi (*Hypothenemus Hampei*)

Hypothenemus Hampei atau PBKo adalah hama kumbang yang menyerang buah kopi dengan cara membuat lubang di sekitar diskus dan masuk ke dalam buah kopi. Serangan pada buah akan berakibat penurunan mutu kopi karena biji berlubang.

b. Penggerek Batang Merah (*Zeuzera coffeae*)

Hama ini menggerek bagian batang atas dari tanaman kopi. Larva mengebor batang tanaman kopi. Apabila luas gerakan melingkar dan bertemu maka bagian tanaman di atas gerakan akan mengering, mati, dan mudah patah.

c. Penggerek Cabang dan Ranting (*Xylosandrus*)

Hama penggerek cabang *Xylosandrus* menggerek cabang kopi yang muda dan masih lunak. Kumbang ini membikin lubang masuk kedalam ranting pohon kopi sehingga ranting atau cabang itu tidak berbuah.

d. Kutu Hijau (*Coccus Viridis*)

Kutu hijau menyerang cabang, ranting dan daun pohon kopi yang berwarna hijau. Kutu ini menjalin hubungan simbiosis mutualisme dengan semut. Kutu hijau lebih suka musim kemarau dan juga lebih senang di dataran rendah daripada di dataran tinggi.

e. Wereng (*Sanurus Indecora*)

Wereng menyerang baik pada daun cabang, dan batang tanaman. Bagian yang terserang akan terhambat pertumbuhannya, tunas mengalami malformasi, rontok, dan mati. Kerusakan dapat bertambah parah jika lapisan lilin tersebut ditumbuhi embun jelaga karena dapat menghambat fotosintesis.

f. Karat Daun (*H. Vastatrix*)

H. Vastatrix yang menyerang saat pembibitan atau saat tanaman dewasa. Gejalanya yaitu daun yang sakit timbul bercak kuning kemudian berubah menjadi coklat. Permukaan bercak pada sisi bawah daun terdapat uredospora seperti tepung berwarna oranye atau jingga.

g. Jamur Upas (*Upasia salmonicolor*)

U. salmonicolor dapat menyerang batang, cabang, ranting dan buah kopi. Serangan dimulai dengan adanya benang-benang jamur tipis seperti sarang laba-

laba. Selanjutnya pada bagian tersebut terjadi nekrosis kemudian membusuk sehingga warnanya menjadi coklat tua atau hitam.

h. Mati Pucuk (*Rhizoctonia sp.*)

Penyakit ini ditemukan pada tanaman kopi muda yang belum dipangkas ujungnya, dan gejala penyakit adalah matinya ujung batang, cabang, atau ranting, yang disertai dengan menguning dan gugurnya daun yang sakit.

i. Bercak Daun (*C. Coffeicola*)

Daun yang terserang akan timbul bercak berwarna kuning yang tepinya dikelilingi halo (lingkaran) berwarna kuning. Buah yang terserang timbul bercak berwarna coklat, biasanya pada sisi yang lebih banyak menerima cahaya matahari.

j. Akar Coklat (*Fomes Noxius Corner*)

Penyakit ini disebabkan oleh Jamur *Fomes Noxius Corner*. Permukaan akar tanaman terserang akan diliputi oleh benang-benang jamur yang berwarna cokelat karat dan mengikat erat butir-butir tanah. Pada butir-butir tanah terdapat hifa jamur berwarna cokelat.

k. Akar Hitam (*Rosellinia Bunodes*)

Gejala serangan jamur ini adalah pohon mati secara mendadak, pada pangkal batang dan akar-akar terdapat banyak benang jamur berwarna hitam, yang sering bersatu dan membentuk lapisan berwarna hitam.

l. Kanker Belah (*Armillaria sp*)

Gejala serangan ditandai dengan daun-daun menguning, layu, dan akhirnya gugur serta cabangcabang mati. Gejala lanjut terdapat cela-celah memanjang pada pangkal batang dan akar tunggang.

m. *Nematoda (Pratylenchus coffeae)*

Serangan nematoda dapat mempengaruhi proses fotosintesis dan transpirasi serta status hara tanaman, sehingga pertumbuhan terhambat, warna daun kuning klorosis dan akhirnya tanaman mati. Selain itu serangan nematoda dapat menyebabkan tanaman lebih mudah terserang patogen atau OPT lainnya.

4. *Website*

Website adalah kumpulan dari halaman web yang sudah dipublikasikan di jaringan internet dan memiliki domain/URL (*Uniform Resource Locator*) yang dapat diakses semua pengguna internet dengan cara mengetikkan alamatnya”. Ada dua jenis aplikasi *website*, *web server* dan *web browser*.

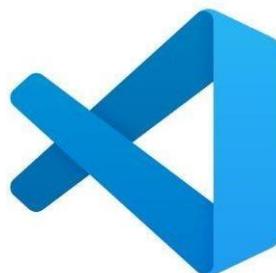
Web browser adalah aplikasi yang mampu menjalankan dokumendokumen web dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat didalam aplikasi *browser* yang biasa disebut *Web Engine*. Semua dokumen web ditampilkan oleh *browser* dengan cara diterjemahkan. Beberapa jenis browser yang populer saat ini diantaranya adalah *Internet Explorer* yang diproduksi oleh *Microsoft*, *Mozilla Firefox*, *Opera*, dan *Safari* yang diproduksi oleh *Apple*. Dari kedua definisi tersebut, dapat disimpulkan *website* adalah kumpulan halaman berisikan informasi-informasi yang dihubungkan oleh jaringan dan disimpan dalam sebuah *web server* (Wahid, 2020)

Web adalah bagian dari layanan yang mampu beroperasi melalui teknologi internet. Akses ke halaman web memanfaatkan teknologi *web server* yang bertindak sebagai penyedia halaman, dengan HTML sebagai bahasa standarnya, dan HTTP sebagai sarana pengiriman dokumen web.

Sebuah situs *web* adalah sebuah halaman informasi yang dapat diakses melalui internet, memungkinkan akses di berbagai lokasi selama terhubung dengan jaringan *internet*. Situs web terbentuk dari berbagai komponen seperti teks, gambar, suara, dan animasi, menciptakan suatu media informasi menarik yang dapat diakses oleh pengguna. Dalam mengategorikan jenisnya, situs web dapat dikelompokkan menjadi tiga kategori yang berbeda. (Hakim & Meilina, 2022)

- a) *Website Statis* merujuk pada jenis situs web yang memiliki halaman - halaman tetap dan tidak berubah. Perubahan pada halaman-halaman tersebut dilakukan secara manual dengan mengedit kode yang menjadi struktur dasar situs, apabila ada kebutuhan perubahan pada suatu halaman.
- b) Sebaliknya, *Website Dinamis* mencakup informasi yang dapat diperbarui melalui aplikasi web yang terhubung. Dengan menggunakan jenis *website* ini, pembaruan informasi dapat dilakukan lebih *fleksibel* tanpa harus menyentuh secara langsung pada kode struktur situs.
- c) Sedangkan, *Website Interaktif* memberikan pengguna kemampuan untuk berinteraksi dan berdiskusi, bahkan adu argumentasi mengenai pemikiran mereka. Pada jenis situs web ini, pengguna dapat lebih aktif terlibat dalam konten dan berpartisipasi dalam diskusi atau aktivitas lainnya.

5. *Visual Studio Code*



Gambar 2. 1 Logo Visual Studio Code

Visual Studio Code merupakan aplikasi editor kode program yang bersifat *open source*, dikembangkan oleh *Microsoft*. Aplikasi ini dapat digunakan pada sistem operasi *Windows*, *Mac OS*, dan *Linux*. *Visual Studio Code* menyederhanakan proses penulisan kode program dan dapat diterapkan untuk berbagai bahasa pemrograman, termasuk *PHP*, *Python*, *Java*, *C#*, *C++*, dan *GO*.

Visual Studio dirancang dengan fokus pada produktivitas. Alat ini juga dikenal sebagai *Rapid Application Development tools (RAD tools)* karena dirancang dan dilengkapi untuk meningkatkan produktivitas. Versi terbaru dari *Visual Studio* didesain untuk kemudahan penggunaan dan pembelajaran. Fitur-fitur dalam *Visual Studio* versi terkini dirancang dengan sederhana agar lebih mudah dipelajari oleh pengguna, dan hal ini memenuhi kebutuhan para *programmer*.

Selain itu, *Visual Studio Code* memiliki kemampuan untuk mengidentifikasi bahasa pemrograman yang sedang digunakan serta fungsi-fungsi yang digunakan dalam struktur kode program. Hal ini ditandai dengan penggunaan warna yang beragam sebagai penanda perbedaan. (Adiyanti et al., 2021)

6. Hypertext Preprocessor (PHP)

Awalnya, PHP berasal dari singkatan *Personal Home Page* dan sering dipakai dalam pembuatan situs web sederhana. Seiring waktu, PHP juga digunakan dalam pembuatan situs populer seperti *Wikipedia*, *WordPress*, *Joomla*, dan

lainnya. Saat ini, PHP diartikan sebagai *Hypertext Preprocessor*. Kode program dalam PHP tidak terlihat oleh pengguna, sehingga keamanan halaman web dapat terjamin. PHP juga bersifat gratis dan *open source*. (Setiawan et al., 2019)

Secara umum, dalam pembangunan halaman web, PHP bukanlah bahasa pemrograman utama. HTML dan CSS sudah cukup untuk menciptakan halaman web statis dengan konten yang tetap. Bahasa PHP lebih sebagai pelengkap, digunakan untuk menghasilkan situs web yang dinamis dan interaktif. Berikut akan dijelaskan mengenai beberapa fungsi dari PHP :

a. Mempersingkat Tatanan HTML dan CSS

Untuk membuat sebuah halaman web yang dinamis, PHP dapat membantu menyederhanakan penggunaan struktur HTML dan CSS. Sebagai ilustrasi, dalam suatu sistem karyawan yang memiliki 100 baris data. Jika menggunakan HTML dan CSS, baris tersebut dapat menjadi sangat panjang. Tetapi, dengan penambahan PHP, Anda dapat mengatur dengan mudah beberapa baris yang perlu ditampilkan.

b. Penginputan Data

Dengan memanfaatkan bahasa pemrograman PHP, Anda dapat menginput data dan menyimpannya dalam database sistem, seperti MySQL."

c. Manajemen *Cookie* dan *Session*

Dalam PHP, *Cookie* dan *Session* berperan dalam menyimpan data pengguna. *Session* dimulai dengan fungsi *session_start()* dan *Cookie* dibuat menggunakan fungsi *setcookie()*. Sebagai contoh penggunaan *cookie*, data seperti *username* dan *password* dapat disimpan di *browser*, memungkinkan pengguna untuk tetap masuk tanpa perlu mengisi formulir saat membuka situs yang sama.

Sebaliknya, *session* dapat digunakan untuk menyimpan informasi *login* yang hanya berlaku selama satu sesi tertentu.

d. Kompres Teks

Di dalam PHP, anda memiliki kemampuan untuk mereduksi panjang teks dengan menggunakan fungsi *gzcompress()* dan mengembalikannya ke bentuk asal dengan fungsi *gzuncompress()*.

7. MySQL (*My Structured Query Language*)

Seiring perkembangan zaman, teknologi, termasuk perangkat lunak, semakin berkembang dengan cepat. Salah satu contoh perangkat lunak yang terus diperbarui oleh produsernya adalah *MySQL*. Perangkat lunak ini merupakan pengembangan lebih lanjut dari proyek UNIREG yang dikerjakan oleh Michael Monty Widenius dan TeX, sebuah perusahaan perangkat lunak asal Swedia.

MySQL adalah *Database Management System (DBMS) open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik dengan penggunaan terbatas). Sebagai *database server*, *MySQL* tersedia secara gratis dengan lisensi GNU *General Public License (GPL)*, sehingga dapat digunakan baik untuk keperluan pribadi maupun komersial tanpa perlu membayar lisensi.

MySQL termasuk dalam jenis RDBMS (*Relational Database Management System*). Oleh karena itu, istilah-istilah seperti baris, kolom, dan tabel digunakan dalam *MySQL*. Sebagai contoh, dalam *MySQL*, sebuah *database* dapat terdiri dari satu atau beberapa tabel.

MySQL berfungsi sebagai *database engine* atau server *database* yang

mendukung bahasa *SQL* sebagai bahasa interaktif dalam pengelolaan data. Ini adalah perangkat lunak sistem manajemen basis data *SQL*, atau *DBMS*, yang bersifat *multithread* dan mendukung banyak pengguna secara bersamaan (Fitri et al., 2020).

MySQL mempunyai beberapa keunggulan yaitu:

- a. *MySQL* memiliki kemampuan untuk mengelola basis data relasional dan dapat digunakan dalam lingkungan *client/server*.
- b. *MySQL* adalah perangkat lunak *open source*, yang berarti bahwa pengguna memiliki kebebasan untuk mengunduh, menggunakan, dan mengadaptasi kode sumbernya tanpa dikenai biaya.
- c. *MySQL* menawarkan performa yang sangat tinggi dengan dukungan *multithreading*, di mana setiap *query* dianggap sebagai *thread-based*, menghasilkan eksekusi yang sangat cepat.
- d. *MySQL* bisa dijangkau oleh klien melalui protokol *TCP/IP* di semua sistem operasi. Di lingkungan *Windows*, klien dapat mengaksesnya dengan menggunakan *named-pipe*, sementara di sistem operasi *UNIX (Linux)*, koneksi dapat dilakukan melalui domain *socket-file*.

8. XAMPP

Salah satu perangkat lunak yang berfungsi sebagai *server* web lokal. *Server* web lokal ini beroperasi di *localhost* komputer dan memiliki peran penting sebagai *server* web serta sistem *database*. *XAMPP* sering digunakan untuk menguji aplikasi web pada *localhost*, baik itu aplikasi web yang dikembangkan secara

native, menggunakan *framework*, atau CMS.

Secara umum, XAMPP berperan sebagai server web yang dapat diakses melalui *localhost* pada komputer. Aplikasi web yang masih dalam tahap pengembangan dapat dikembangkan di dalam server lokal ini. Pengembang dapat menguji seluruh fungsi aplikasi web, melakukan uji coba desain web, menyusun struktur konten, dan hal-hal lain sebelum resmi diluncurkan ke internet.

Dengan XAMPP, aplikasi web dapat dipresentasikan secara nyata. Antarmuka, struktur konten, bahkan fungsi yang berjalan di *back-end* hampir identik dengan aplikasi web yang sebenarnya. Setelah pengujian selesai menggunakan XAMPP, aplikasi web dapat diunggah secara online dengan cara meng-uploadnya ke web hosting.

Fitur-fitur yang tersedia dalam XAMPP sangat mempermudah proses uji coba aplikasi web. Selain digunakan oleh pengembang profesional, XAMPP juga sering digunakan oleh pelajar atau mahasiswa untuk belajar mengembangkan aplikasi web (Wardhani, 2022).

c. *Flowchart*

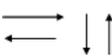
Flowchart atau yang sering disebut sebagai diagram alir, merupakan suatu bentuk diagram yang memvisualisasikan algoritma atau langkah-langkah instruksi secara berurutan dalam suatu sistem. Seorang analis sistem menggunakan *flowchart* sebagai dokumen bukti untuk menjelaskan gambaran logis suatu sistem kepada para *programmer*. Dengan menggunakan *flowchart*, analisis sistem dapat memberikan dokumentasi yang jelas mengenai bagaimana sistem yang akan dibangun bekerja dan memberikan solusi terhadap potensi masalah yang mungkin

muncul selama proses pembangunan sistem. Dasar dari *flowchart* ini terletak pada penggunaan simbol-simbol yang merepresentasikan berbagai proses. Tiap simbol mewakili suatu langkah atau proses tertentu dalam algoritma. Penghubungan antar proses dilakukan dengan menggunakan garis penghubung.

Keberadaan *flowchart* membantu membuat setiap urutan proses menjadi lebih jelas dan memungkinkan penambahan proses baru dengan mudah. Setelah proses pembuatan *flowchart* selesai, tugas selanjutnya jatuh pada *programmer* yang akan menerjemahkan desain logis tersebut ke dalam bentuk program menggunakan berbagai bahasa pemrograman yang telah disepakati.

Simbol-simbol *flowchart*, antara lain :

Tabel 2. 1 Simbol-simbol *flowchart*

No	GAMBAR	NAMA	KETERANGAN
1		<i>Arus / Flow</i>	Penghubung antara prosedur atau proses.
2		<i>Connector</i>	Simbol keluar atau masuk prosedur atau proses dalam lembar atau halaman yang Sama
3		<i>Off-line Connector</i>	Simbol keluar atau masuk prosedur atau proses dalam lembar atau halaman yang lain
4		<i>Process</i>	Simbol yang menunjukkan pengolahan yang dilakukan komputer
5		<i>Decision</i>	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau aksi
6		<i>Predefined Process</i>	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i> .

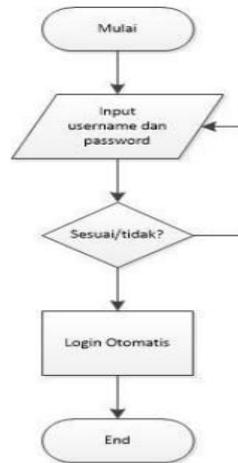
No	GAMBAR	NAMA	KETERANGAN
7		<i>Terminal</i>	Simbol untuk permulaan atau akhir dari suatu program.
8		<i>Manual Input</i>	Simbol untuk pemasukan kata secara manual <i>on-line keyboard</i>
9		<i>Document</i>	Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak di kertas.
10		<i>Input-Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya
11		<i>Disk and On-Line Storage</i>	Simbol untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>
8		<i>Manual Input</i>	Simbol untuk pemasukan kata secara manual <i>on-line keyboard</i>

d. *Whitebox testing*

Proses pengujian organisasi internal perangkat lunak, desain, dan kode program terkait dikenal sebagai pengujian "kotak putih". Teknik ini dapat mendeteksi kelemahan implementasi perangkat lunak. Penguji kotak putih harus menguasai kode sumber perangkat lunak yang mereka uji.

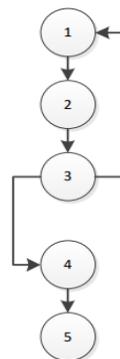
Berikut ini adalah contoh pengujian *white box* pada Aplikasi Pengingat Dan Pendataan Kenaikan Golongan Gaji Berbasis Web (Firman et al., 2021), yaitu sebagai berikut:

1. Proses Login



Gambar 2.12 Proses Login

Hasil dari prosedur login pengujian ini menunjukkan bahwa fungsi aplikasi beroperasi sesuai dengan logika saat ini. Sistem akan langsung menerima dan membawa *administrator ke dashboard* jika pengguna memasukkan login dan password yang benar. Namun, sistem akan mengingatkan pengguna dan meminta mereka mengubah data jika tidak akurat.



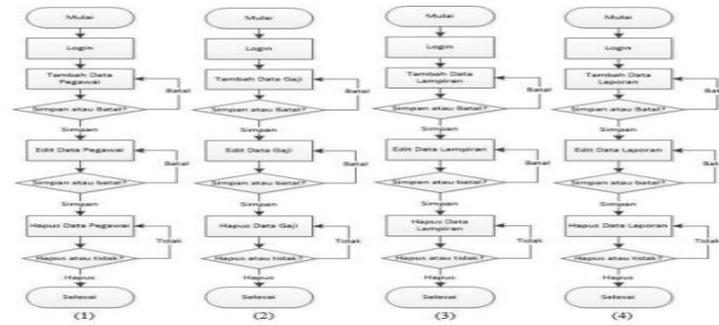
Gambar 2.3 Flowchart Node Login

Lima (N) node dan lima (E) edge membentuk diagram alur siklomatik dari node login. Untuk Path maka

$$V(G) = E - N + 2 = 5 - 5 + 2 = 2$$

Jadi, ada dua path.

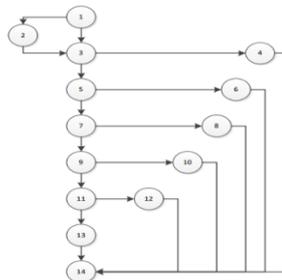
2. Proses Pengolahan Data



Gambar 2.4 Proses Pengelolaan Data Pegawai, Gaji, Lampiran, dan Laporan

Logika pengelolaan data pegawai, pengelolaan data gaji, pengelolaan data lampiran, dan pengelolaan data laporan diuji, dan ternyata semuanya berjalan lancar sesuai dengan alur yang ada.

3. Perhitungan *Kompleksits Cyclomatic*



Gambar 2.5 *Flowchart Note*

Gunakan rumus berikut untuk menentukan *kompleksitas siklomatik*:

$$(G) = E - N + 2 = 19 - 14 + 2 = 7$$

Detail: E adalah jumlah tautan atau busur.

N adalah jumlah node.

Untuk jalur *independent* dari *flowchart* mempunyai 7 path:

Path 1:1-2-3-5-7-9--11-13-14

Path 5:1-3-5-7-9-10-14

Path 2:1-3-4-14

Path 6: 1-3-5-7-9--11-12-14

Path 3:1-3-5-6-14

Path 7: 1-3-5-7-9--11-13-14

Path 4:1-3-5-7-8-14

Tabel 2.3 Rekapitulasi Hasil *Whitebox Testing*

Flowchart	Cyclomatic Complexity	Region (R)	Independent Path
1. Login	2	2	2
2. Data Pegawai	7	7	7
3. Gaji	7	7	7
4. Lampiran	7	7	7
5. Laporan	7	7	7
Jumlah	30	30	30

Hasil pengujian pada tabel di atas menunjukkan bahwa alur logika sistem aplikasi pengingat kenaikan kelas dan gaji berbasis web sudah benar dan dapat diterima karena angka Cyclomatic Complexity = 30, Region = 30, dan Independent Path = 30 semuanya merupakan sama

e. *Blackbox testing*

Tujuan pengujian perangkat lunak berbasis spesifikasi fungsional adalah untuk menentukan apakah *input*, *output*, dan fungsi perangkat lunak memenuhi persyaratan yang diperlukan tanpa memeriksa desain atau kode program. Karena pendekatan pengujian *Blackbox* hanya memerlukan batas data yang diharapkan batas bawah dan atas ini merupakan prosedur yang mudah. Jumlah *entri* data yang diperiksa, persyaratan *entri* yang harus dipenuhi, dan skenario batas atas dan bawah yang relevan semuanya dapat digunakan untuk memperkirakan jumlah data pengujian. Teknik ini memungkinkan kita untuk menentukan apakah *input* yang

tidak terduga masih diterima oleh fungsionalitas perangkat lunak, yang dapat menyebabkan data yang disimpan menjadi tidak *valid* (Cholifah et al., 2018)

Berikut ini adalah contoh pengujian *blackbox* pada Aplikasi Pengingat Dan Pendataan Kenaikan Golongan Gaji Berbasis Web (Firman et al., 2021), yaitu sebagai berikut:

Tabel 2.2 Contoh Pengujian *Blackbox*

No	Fungsionalitas	Perintah dan Masukan	Hasil	Status
1	Menu Login	Admin Memasukkan username dan password dengan benar	Sistem mengarahkan admin ke halaman dashboard	Sukses
		Admin Memasukkan username dan password dengan salah	Sistem menampilkan notifikasi dan tampilan form login	Sukses
2	Menu Pegawai	Admin mengklik menu karyawan serta menambah, mengubah atau menghapus data karyawan	Sistem akan menampilkan data pegawai sesuai perintah dan masukan	Sukses
3	Menu Penggajian	Admin mengklik menu penggajian serta menambah, mengubah atau menghapus data gaji	Sistem akan menampilkan data penggajian sesuai perintah dan masukan	Sukses
4	Menu Laporan	Admin mengklik menu laporan	Sistem akan menampilkan data laporan sesuai ada ada	Sukses

Setiap menu yang ada saat ini dilakukan pengujian aplikasi dengan teknik *black box*. Dalam tes ini, mengklik halaman login, memasukkan nama pengguna dan kata

sandi yang cocok dengan database, lalu mencoba memasukkan yang salah. Hasil pengujian menunjukkan bahwa setiap fitur dan fungsi program berfungsi dengan baik dan sah.

f. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah alat yang digunakan untuk memvisualisasikan serta mendokumentasikan hasil dari analisis dan desain suatu sistem. UML mengandung sintaksis yang memungkinkan pemodelan sistem secara visual. Juga merupakan seperangkat aturan pemodelan yang digunakan untuk menjelaskan atau menggambarkan suatu sistem perangkat lunak yang terkait dengan objek.

Pada masa kini, kebanyakan perancang sistem informasi menggunakan diagram UML untuk menggambarkan informasi. Hal ini bertujuan untuk memfasilitasi komunikasi dalam tim proyek, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau program yang dibuat.

Hingga saat ini, para ahli merasa lebih terbantu dalam menganalisis, merancang, atau memodelkan suatu sistem karena UML menyediakan seperangkat aturan dan notasi grafis yang cukup spesifik.

Unified Modelling Language menyediakan 9 jenis diagram yang dikelompokkan berdasarkan sifat statis atau dinamis. Adapun jenis diagram yang bersifat statis yaitu diagram kelas, diagram objek, *use case diagram*, *component diagram*, dan *deployment diagram*. Sedangkan jenis diagram yang bersifat dinamis antara lain *Sequence diagram*, *collaboration diagram*, *statechart diagram*, dan *activity diagram* (Haviluddin, 2016).

Adapun simbol-simbol dari UML yaitu:

Tabel 2. 4 Symbol Use Case Diagram

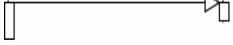
No	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>Use Case</i> .
2		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>Independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		<i>Generalization</i>	Hubungan di mana objek anak (<i>Descendent</i>) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>Ancestor</i>).
4		<i>Include</i>	Menyepifikasikan bahwa <i>Use Case</i> sumber secara Eksplisit.
5		<i>Extend</i>	Menyepifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menyepifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

No	GAMBAR	NAMA	KETERANGAN
11		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

Tabel 2. 5 Symbol Class Diagram

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 6 Symbol Sequence Diagram

No	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

No	GAMBAR	NAMA	KETERANGAN
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
4		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

Tabel 2. 7 Symbol State Chart Diagram

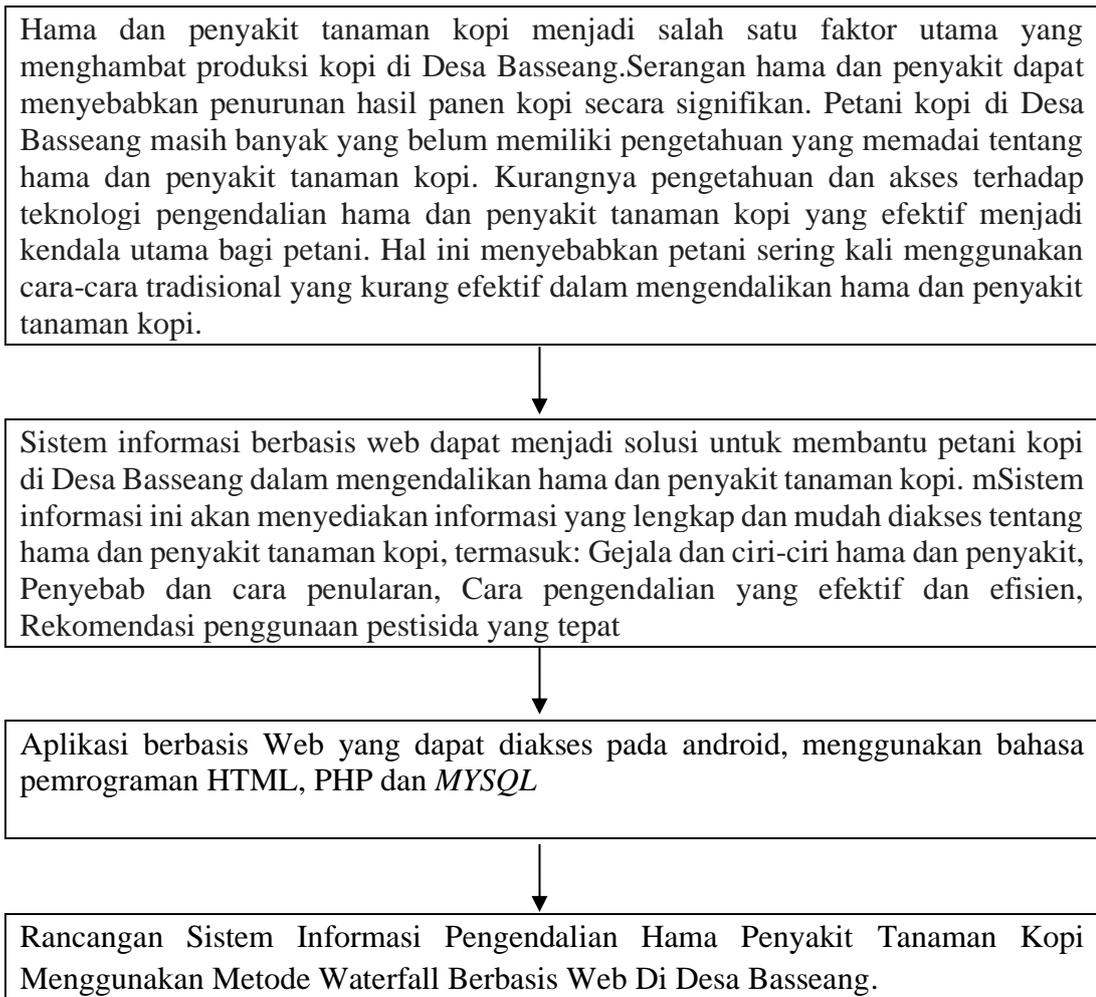
No	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai Link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah <i>state</i> objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 8 Symbol Activity Diagram

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Actifity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

C. Kerangka Pikir

Agar lebih memahami alur penelitian yang telah dipaparkan, diuraikan ke dalam kerangka pikir yang disajikan dalam bentuk diagram berikut ini:



BAB III

METODOLOGI PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang dilakukan merupakan pendekatan kualitatif untuk mendapatkan pemahaman yang mendalam tentang pengendalian hama dan penyakit pada tanaman kopi di Desa Dasseang. Pendekatan kualitatif memungkinkan peneliti untuk memahami konteks, kompleksitas, dan makna dari fenomena yang diteliti.

B. Waktu Penelitian

Tabel 3. 1 Jadwal Pelaksanaan Penelitian

No.	Uraian Kegiatan	Bulan 2024		
		Mei	Juni	Juli
1	Analisis Kebutuhan			
2	Desain			
3	Implementasi			
4	Pengujian			

C. Alat dan Bahan Penelitian

Dalam melakukan penelitian, maka diperlukan alat dan bahan penelitian yang mendukung kegiatan penelitian tersebut. Alat dan bahan yang diperlukan antara lain:

1. Alat yang digunakan selama proses penelitian yaitu: Laptop ASUS dengan spesifikasi:

Tabel 3.2 spesifikasi laptop

No	Nama	Keterangan
1.	Processor	Intel® Celeron®
No	Nama	Keterangan
2.	RAM	2 GB
3.	System Model	Aspire A311-31
4.	Ukuran Layar	11.6 inch
5.	Printer	Epson L350 Series 27

2. Perangkat Lunak yang digunakan dalam pembuatan aplikasi yaitu:

Tabel 3.3 Perangkat Lunak

No	Nama	Keterangan
1.	Sistem Operasi	windows 10 ultimate 64 bit
2.	Aplikasi Program	Php, Msyql, Xampp, Goggle Chrome, <i>Visual Studio Core</i>

E. Teknik Pengumpulan Data

1. Teknik pengumpulan data primer

- a. Observasi, berfungsi untuk pencarian data dengan mengidentifikasi data melalui pengukuran serta pengambilan data secara langsung dilapangan. Kegiatan observasi dilakukan secara sistematis untuk menjajaki masalah dalam penelitian serta bersifat eksplorasi. Observasi dilakukan berupa

pengamatan yang dilakukan secara langsung terhadap kondisi fisik dan ketersediaan fasilitas sosial wilayah penelitian

- b. Wawancara atau *interview* adalah suatu bentuk komunikasi verbal semacam percakapan yang bertujuan memperoleh informasi
- c. Dokumentasi adalah pengumpulan data-data sekunder atau dokumentasi dengan mengambil data-data dalam bentuk visual tentang kondisi lapangan.

2. Teknik Pengumpulan Data Sekunder

Dengan cara mengumpulkan data informasi dengan cara membaca atau mengambil data yang sifatnya dokumen dan literature dinas yang terkait dan buku-buku yang mampu mendukung penelitian.

D. Tahap penelitian

1. Analisis Kebutuhan

Proses pengumpulan data kebutuhan dilakukan secara intensif dan menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Peneliti menggambarkan kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. Implementasi

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari

tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus kepada perangkat lunak secara logis dan fungsional dan memastikan bahwa semua bagian sudah diuji untuk meminimalisir eror dan keluaran harus sesuai. Pemilihan cara pengujian dilakukan dengan menggunakan data-data yang sering digunakan untuk pengolahan data, mulai dari data operasional, data *input* dan *output*.

E. Metode Pengujian

Dalam penelitian ini, digunakan 2 (dua) metode dalam pengujian datanya yaitu

blackbox testing dan *whitebox testing*:

1. *Blackbox testing*

Blackbox testing terfokus pada fungsional dari program yang ada. Pada *Blackbox testing* diuji dengan cara menjalankan program kemudian diamati apakah program tersebut apakah berhasil atau tidak. *Blackbox testing* menggunakan teknik *equivalence partitions* yang merupakan pengujian berdasarkan masukan setiap menu yang terdapat pada program, setiap menu masukan dilakukan pengujian melalui klasifikasi dan pengelompokan berdasar fungsinya.

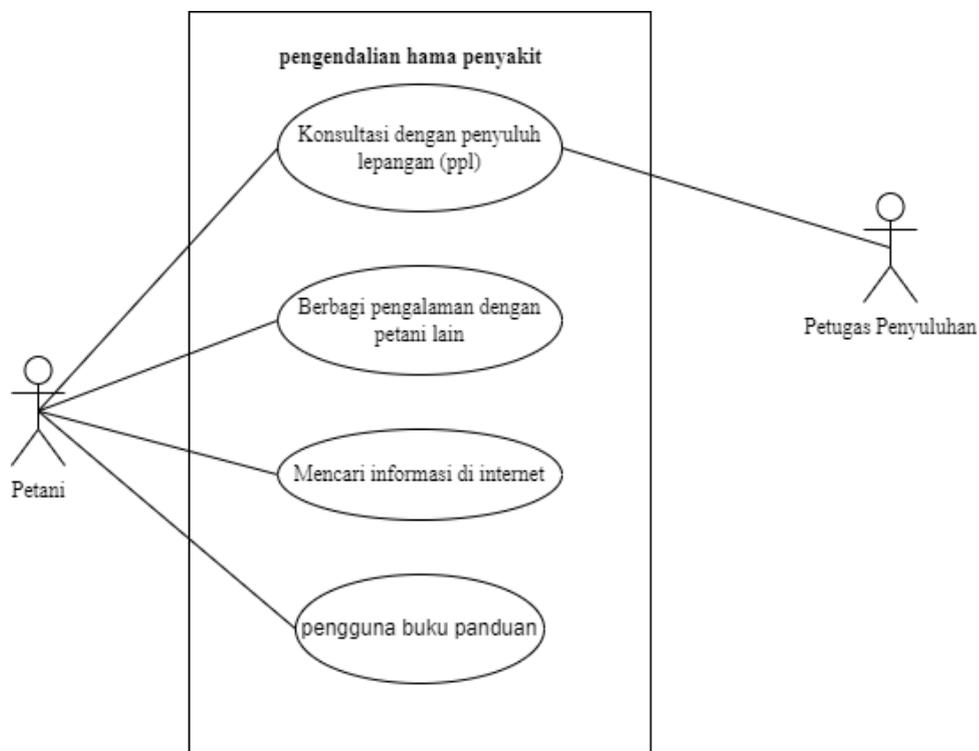
2. *Whitebox testing*

Whitebox testing bertujuan untuk mengetahui apakah struktur pada aplikasi yang dibuat sudah sesuai dengan ketentuan. *Whitebox testing* menitikberatkan pada

pengujian dengan mengecek detail perancangan perangkat lunak. *Whitebo testing* dinilai dengan mendefinisikan semua alur dari perangkat lunak, kemudian membangun kasus yang akan digunakan dalam proses pengujian, kemudian menguji kasus tersebut untuk memperoleh hasilnya.

F. Desain Sistem

1. Desain Sistem yang Berjalan



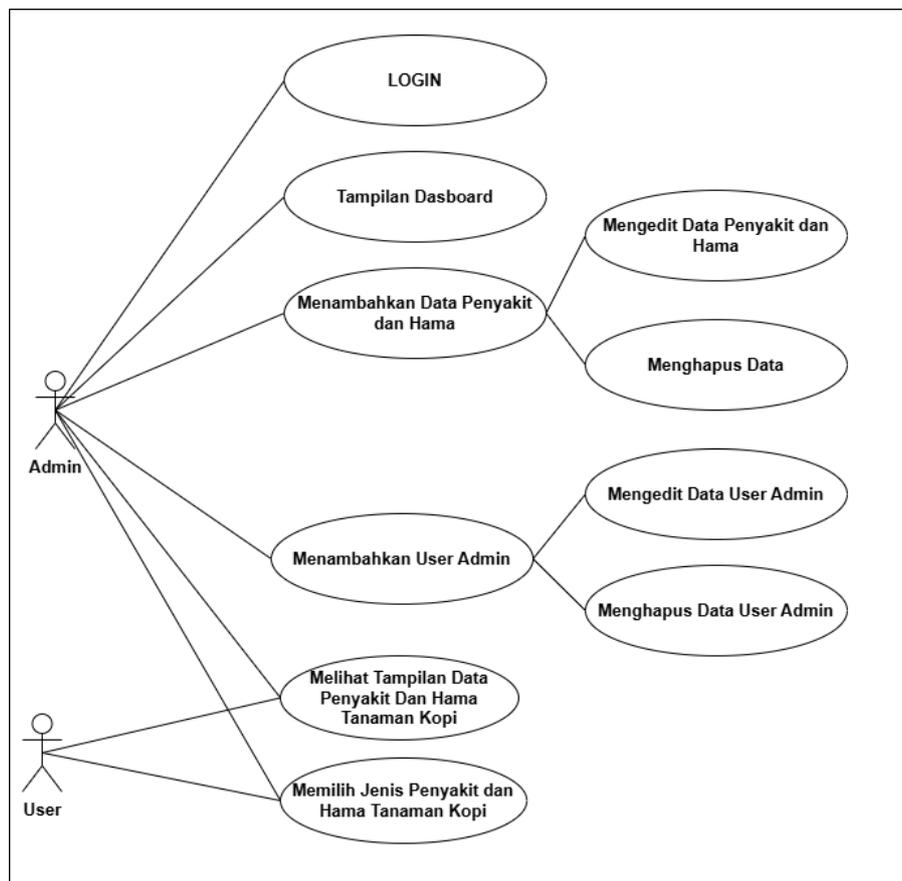
Gambar 3.1 Sistem berjalan

Tabel 3. 4 penjelasan desain sistem yang berjalan

Nama use case	Deskripsi
Konsultasi dengan penyuluh lapangan	Use case ini menjelaskan tentang sistem yang berjalan bahwa user langsung ke penyuluh lapangan

Nama use case	Deskripsi
Berbagi pengalaman dengan petani	<i>Use case</i> ini menjelaskan bahwa user saling berbagi pengalaman sesama petani
Mencari informasi di internet	<i>Use case</i> ini menjelaskan bahwa user mencari informasi di internet
Pengguna buku panduan	<i>Use case</i> ini menjelaskan bahwa user dapat mencari informasi dari buku panduan.

2. Desain Sistem yang Diusulkan



Gambar 3.2 Sistem yang Diusulkan

Tabel 3. 5 penjelasan desain sistem yang akan diusulkan

Nama use case	Deskripsi
Login	<i>Use case</i> ini menjelaskan proses dimana admin dan user login terlebih dahulu
Tampilan Dashboard	<i>Use case</i> ini menjelaskan admin dapat menambahkan informasi.
Menambahkan Data Penyakit dan Hama	<i>Use case</i> ini menjelaskan admin dan user dapat melihat menu.
Mengedit Data Penyakit dan Hama	<i>Use case</i> ini menjelaskan bahwa admin dapat mengedit data penyakit dan hama
Menghapus Data Penyakit Dan Hama	<i>Use case</i> ini menjelaskan bahwa admin dapat menghapus data penyakit dan hama
Menambahkan User Admin	<i>Use case</i> ini menjelaskan bahwa admin dan user dapat melihat halaman jenis-jenis hama dan penyakit
Mengedit Data User Admin	<i>Use case</i> ini menjelaskan bahwa admin dapat mengedit data user admin
Menghapus Data User Admin	<i>Use case</i> ini menjelaskan bahwa admin dapat menghapus data user admin
Melihat Tampilan Data Penyakit Dan Hama Tanaman Kopi	<i>Use case</i> ini menjelaskan bahwa admin dan user dapat melihat halaman tampilan data penyakit dan hama tanaman kopi
Memilih Jenis Penyakit dan Hama Tanaman Kopi	<i>Use case</i> ini menjelaskan bahwa admin dan user dapat melihat melihat halaman jenis penyakit dan hama tanaman kopi

BAB IV

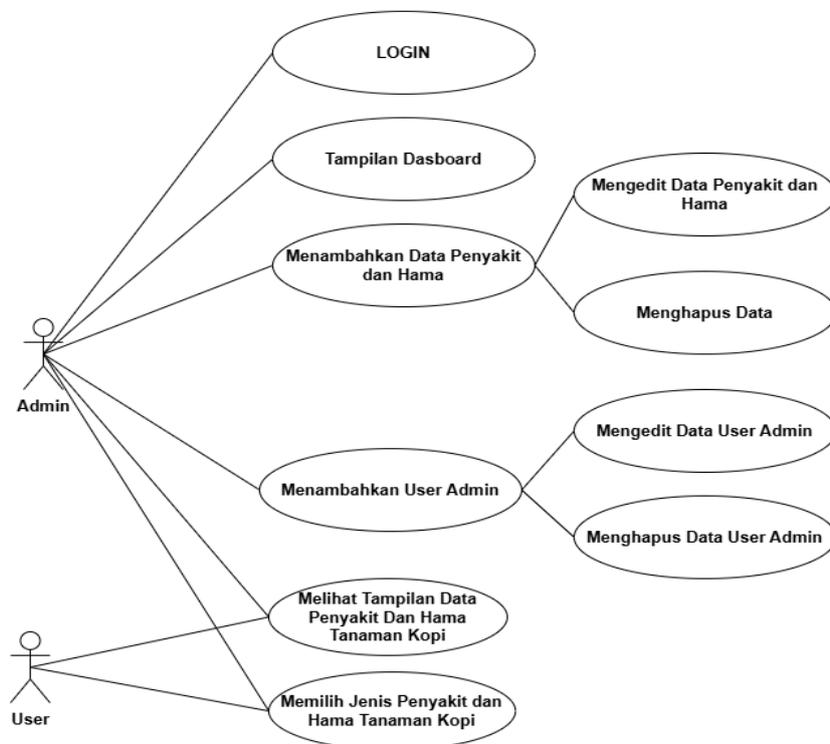
HASIL DAN PEMBAHASAN

A. Analisis Aliran Data UML

Dengan tujuan memberikan informasi tahapan dan alur aplikasi, *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram* digunakan untuk mengorientasikan objek sistem melalui analisis aliran data.

1. Use Case Diagram

Diagram ini membantu menggambarkan fungsionalitas utama dari suatu sistem berdasarkan perspektif pengguna .



Gambar 4.1 Use Case diagram

Tabel 4.1 Penjelasan *Use Case Diagram*

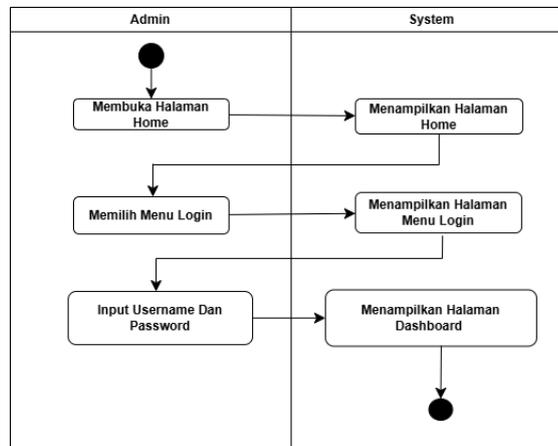
Nama use case	Deskripsi
Login	<i>Use case</i> ini menjelaskan proses dimana admin dan user login terlebih dahulu
Tampilan Dashboard	<i>Use case</i> ini menjelaskan admin dapat menambahkan informasi.
Menambahkan Data Penyakit dan Hama	<i>Use case</i> ini menjelaskan admin dan user dapat melihat menu.
Mengedit Data Penyakit dan Hama	<i>Use case</i> ini menjelaskan bahwa admin dapat mengedit data penyakit dan hama
Menghapus Data Penyakit Dan Hama	<i>Use case</i> ini menjelaskan bahwa admin dapat menghapus data penyakit dan hama
Menambahkan User Admin	<i>Use case</i> ini menjelaskan bahwa admin dan user dapat melihat halaman jenis-jenis hama dan penyakit
Mengedit Data User Admin	<i>Use case</i> ini menjelaskan bahwa admin dapat mengedit data user admin
Menghapus Data User Admin	<i>Use case</i> ini menjelaskan bahwa admin dapat menghapus data user admin
Melihat Tampilan Data Penyakit Dan Hama Tanaman Kopi	<i>Use case</i> ini menjelaskan bahwa admin dan user dapat melihat halaman tampilan data penyakit dan hama tanaman kopi
Memilih Jenis Penyakit dan Hama Tanaman Kopi	<i>Use case</i> ini menjelaskan bahwa admin dan user dapat melihat melihat halaman jenis penyakit dan hama tanaman kopi

2. Activity Diagram

Diagram ini menggambarkan serangkaian aktivitas yang terjadi dalam sistem beserta urutan pelaksanaannya.

a. Activity diagram admin

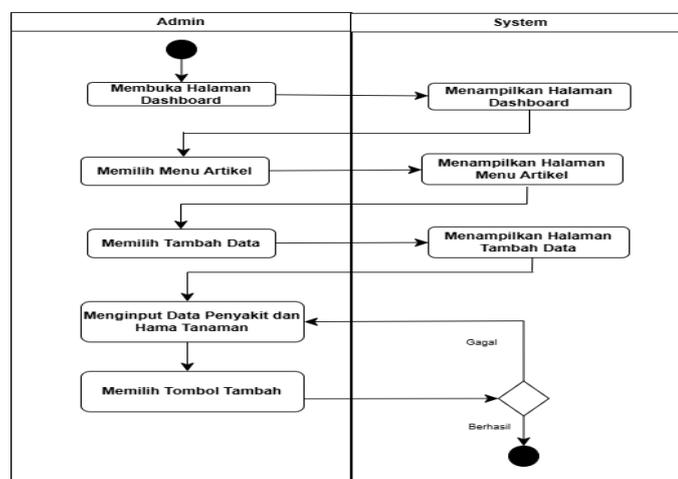
1) Activity diagram login



Gambar 4. 2 Activity diagram login

Gambar 4. 2 menjelaskan cara masuk sebagai *admin*. *Admin* harus terlebih dahulu mengakses situs web, setelah itu *admin* akan melihat tampilan *form login* dan diharuskan memasukkan kembali nama pengguna dan kata sandi yang telah di buat. Kemudian sistem akan mencocokkan; jika cocok, akan berada di halaman *admin*; apabila tidak, tetap di halaman *login*.

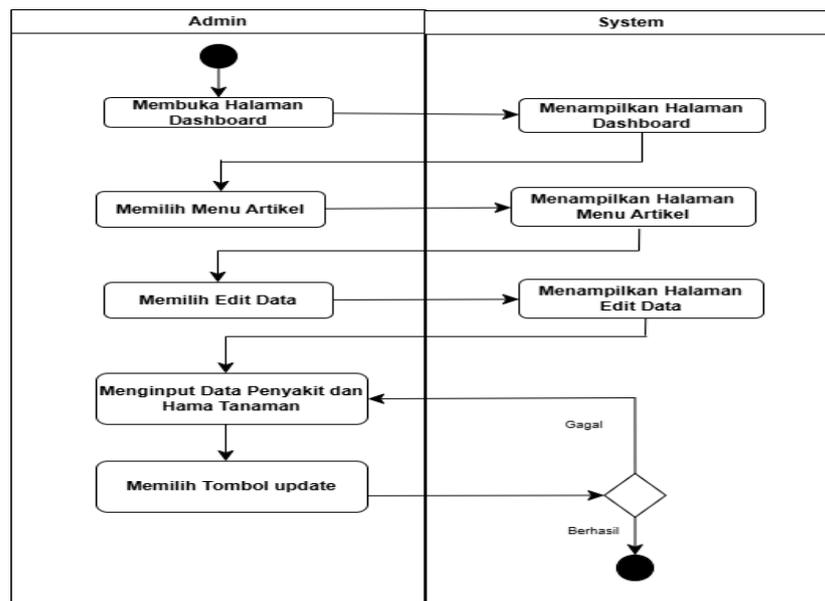
2) Activity diagram tambah data



Gambar 4. 3 Activity diagram tambah data

Gambar 4. 3 menjelaskan langkah-langkah yang dilakukan oleh *admin* untuk menambahkan data penyakit atau hama dan tambah user admin. Halaman *admin* ditampilkan oleh sistem ketika telah dibuka oleh admin. *Admin* kemudian memilih item menu tambah data, sistem kemudian menampilkan menu yang dipilih dan *admin* memilih tambah data. Form tambah data kemudian akan ditampilkan oleh sistem. *Admin* selanjutnya diminta untuk melengkapi *form* tambah data. Setelah selesai, sistem akan mengecek data dan jika berhasil data tersebut tersimpan di halaman home

3). Activity diagram ubah data

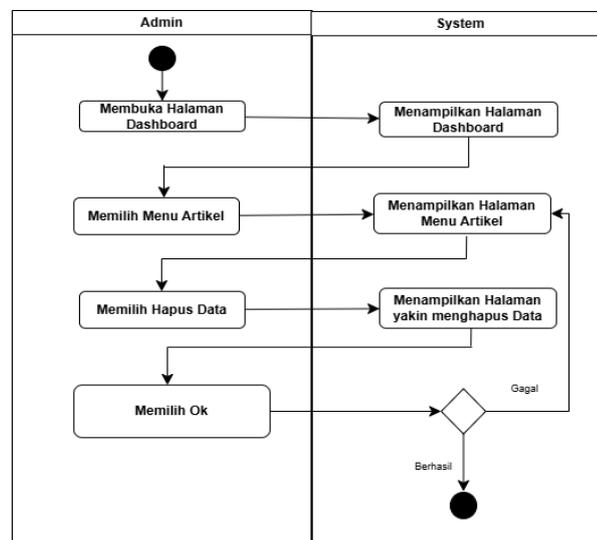


Gambar 4. 4 Activity diagram ubah data

Gambar 4. 4 menjelaskan langkah-langkah yang perlu diikuti oleh *admin* untuk memodifikasi data. Hal pertama yang dilakukan *admin* adalah membuat folder *admin*. Kemudian, sistem akan menampilkan menu *admin*. Selanjutnya, *admin* memilih menu yang akan diubah datanya; sistem kemudian

menampilkan judul menu yang *admin* pilih; terakhir, *admin* mengubah data. Sistem akan menampilkan data dalam bentuk tabel. Setelah *admin* mengisi *form* data, sistem akan memvalidasi informasi tersebut. Jika validasi cocok, sistem menampilkan item menu dengan data yang dimasukkan, jika tidak, administrator disarankan untuk mengisi form data kembali dengan akurat

3) *Activity diagram* hapus data

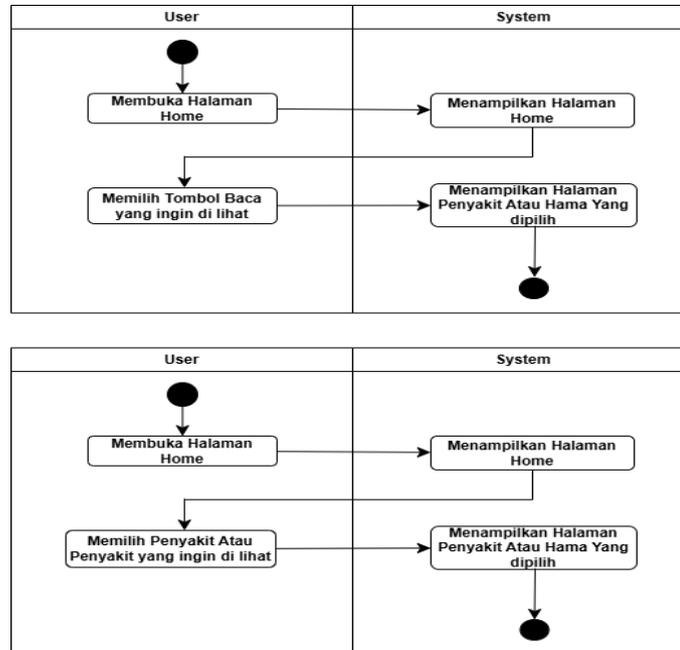


Gambar 4. 5 *Activity diagram* hapus data

Gambar 4. 5 menjelaskan prosedur yang digunakan *admin* untuk menghapus data. Halaman *admin* ditampilkan oleh sistem ketika telah dibuka oleh *admin*. *Admin* kemudian memilih menu yang datanya akan dimusnahkan, sistem menampilkan halaman menu yang telah dipilih oleh *admin*, dan *admin* mengklik pilihan hapus data. Sistem kemudian akan menampilkan konfirmasi penghapusan. *Admin* kemudian melakukan konfirmasi. Jika *admin* konfirmasi, data akan terhapus, jika tidak maka data batal dihapus.

b. Activity diagram User

1) Activity diagram Tampilan data Penyakit Atau Hama



Gambar 4. 6 Activity diagram Tampilan data penyakit atau hama

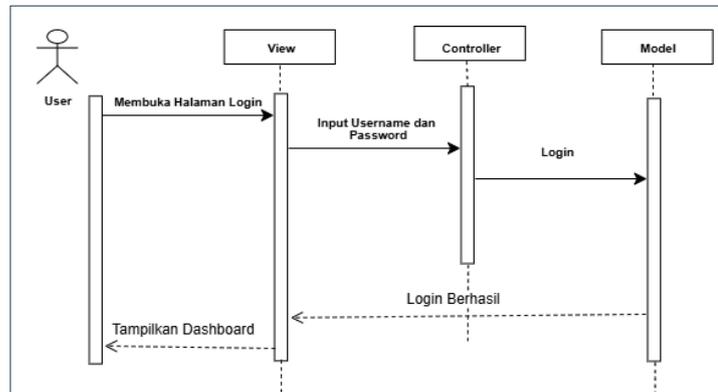
Gambar 4. 6 menjelaskan tentang halaman penyakit atau hama tanaman kopi, dimana user membuka website setelah itu system akan menampilkan halaman home kemudian user akan memilih tombol baca atau memilih penyakit dan hama yang ingin dilihat selanjutnya sitem akan menampilkan salah satu penyakit atau hama yang di pilih oleh user.

3. Sequence Diagram

Diagram yang memvisualisasikan interaksi antara objek-objek secara berurutan dalam sistem seiring berjalannya waktu. Diagram ini memperlihatkan cara objek saling bertukar pesan atau data untuk menyelesaikan suatu tugas atau *use case* tertentu.

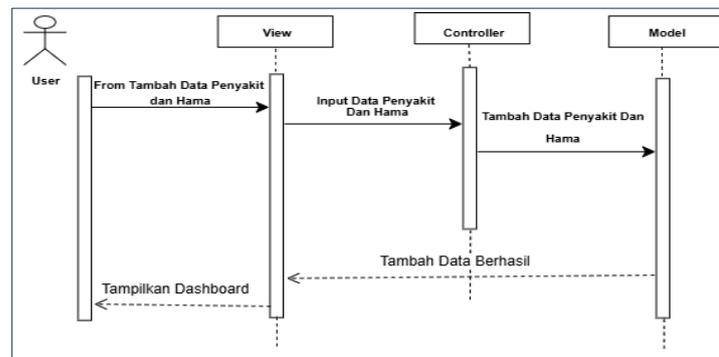
a. Sequence diagram admin

1) Sequence diagram login



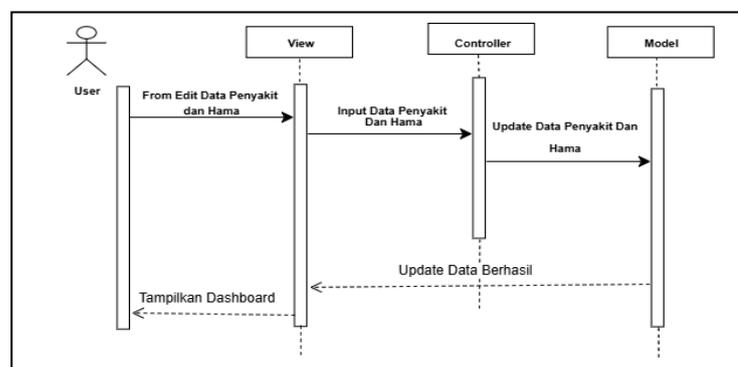
Gambar 4. 7 Sequence diagram Login

2) Sequence diagram tambah data



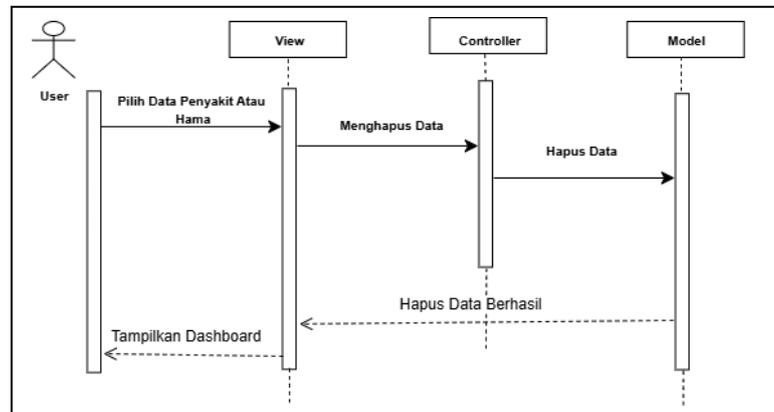
Gambar 4. 8 Sequence diagram tambah data

3) Sequence diagram ubah data



Gambar 4. 9 Sequence diagram ubah data

4) *Sequence diagram* hapus data

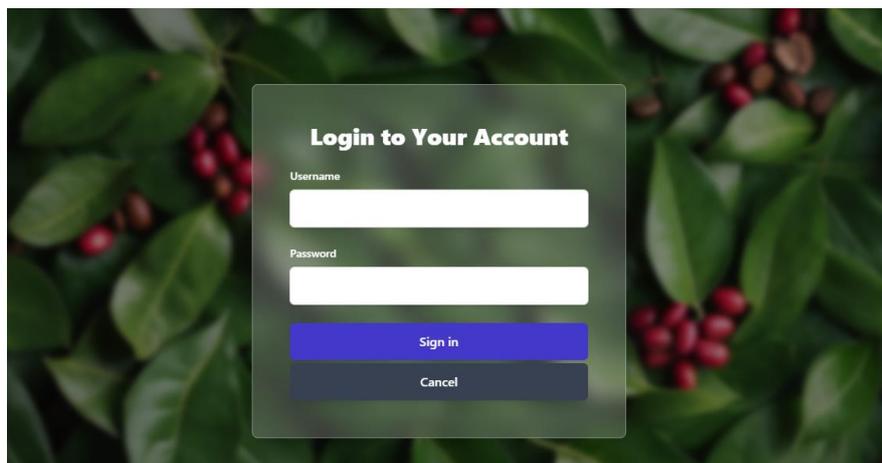


Gambar 4. 10 *Sequence diagram* hapus data

B. Detail Sistem

1. Halaman *Login Admin*

Tampilan halaman *login admin* merupakan fitur pada aplikasi atau sistem manajemen yang bertujuan untuk membatasi akses hanya kepada pengguna yang memiliki hak akses sebagai admin. Halaman *login admin* merupakan halaman utama yang digunakan oleh *admin* untuk mendapatkan akses ke halaman *admin*.



Gambar 4. 11 *Halaman Login admin*

```

<?php
session_start();
require "../config/koneksi.php";

if (isset($_POST["submit"])) {

    $username = $_POST["username"];
    $pass = md5($_POST["pass"]);

    $sql = "SELECT*FROM users WHERE username='$username' AND pass='$pass'";
    $result = $conn->query($sql);
    $row = $result->fetch_assoc();
    if ($result->num_rows > 0) {

        $_SESSION['username'] = $row["username"];
        $_SESSION['level'] = $row["level"];
        $_SESSION['status'] = "y";

        header("Location:index.php");
    } else {
        header("Location:?msg=n");
    }
}
$conn->close();
?>

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>LOGIN</title>

<link rel="stylesheet" href="../src/style.css">
<script src="https://cdn.tailwindcss.com"></script>
<style>
/* Pastikan body memiliki posisi relatif sebagai referensi */
body {
    position: relative;
    /* Jika diinginkan, bisa atur warna latar belakang fallback */
    background-color: #000;
}

/* Pseudo-elemen ::before untuk menampilkan gambar latar dengan efek blur */
body::before {
    content: "";
    position: fixed;
    /* Mengisi seluruh layar */
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-image: url('../uploads/bg-login1.jpg');
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
    filter: blur(4px);
    /* Sesuaikan nilai blur sesuai keinginan */
    z-index: -2;
    /* Letakkan di belakang konten */
}

/* Pseudo-elemen ::after untuk overlay gelap */
body::after {
    content: "";
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(0, 0, 0, 0.4);
    /* Sesuaikan tingkat kegelapan, misal 0.4 */
    z-index: -1;
    /* Di atas gambar tetapi di bawah konten */
}
</style>
</head>

<body class="bg-gray-100">
<div class="flex min-h-screen flex-col justify-center py-12 sm:px-6 lg:px-0">
<div class="sm:mx-auto sm:w-full sm:max-w-md">

<!-- Validasi Login Gagal -->
<?php
if (isset($_GET['msg'])) {
    if ($_GET['msg'] == "n") {
    }
}
?>
<div class="alert alert-danger bg-red-500 text-white p-3 rounded-md mb-4" role="alert">
<strong>Login Gagal</strong>
</div>
<?php
}
?>
</div>

<div class="mt-10 sm:mx-auto sm:w-full sm:max-w-[480px]">
<div class="bg-white/20 px-6 py-12 shadow-lg sm:rounded-lg sm:px-12 border border-white/30 rounded-xl backdrop-blur-md">
<form class="space-y-6" method="POST">
<h2 class="text-center text-3xl font-extrabold text-white mb-6">Login to Your Account</h2>

<div>
<label for="email" class="block text-sm font-medium text-white">Username</label>
<div class="mt-2">
<input name="username" type="text" autocomplete="off" required
class="block w-full px-4 py-3 rounded-md text-gray-900 shadow-md ring-1 ring-white/50 focus:ring-2 focus:ring-indigo-600 focus:outline-none transition duration-200">
</div>
</div>

```



```

<?php
$sql = "SELECT COUNT(*) as jumlah FROM users";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $total_users = $row['jumlah'];
} else {
    $total_users = 0;
}

$sql = "SELECT COUNT(*) as jumlah FROM artikel";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $total_artikel = $row['jumlah'];
} else {
    $total_artikel = 0;
}
$conn->close();
?>
<div class="space-y-16 py-8 xl:space-y-20">
    <div class="mx-auto max-w-7xl px-4 sm:px-6 lg:px-8">
        <div class="mx-auto max-w-2xl lg:mx-0 lg:max-w-none">
            <div class="flex items-center justify-between">
                <h2 class="text-base font-semibold leading-7 text-gray-900">Beranda</h2>
            </div>
        </div>
        <ul role="list" class="mt-6 grid gap-x-6 gap-y-8 justify-center items-center text-center lg:grid-cols-3 xl:gap-x-8">
            <li class="overflow-hidden rounded-xl border border-gray-200">
                <div class="flex items-center gap-x-4 border-b border-gray-900/5 bg-gray-50 p-6">
                    <div class="text-sm font-medium leading-6 text-gray-900">Data</div>
                </div>
                <dl class="-my-3 divide-y divide-gray-100 px-6 py-4 text-sm leading-6">
                    <div class="flex justify-between gap-x-4 py-3">
                        <dt class="text-gray-500">Total</dt>
                        <dd class="flex items-start gap-x-2">
                            <div class="font-medium text-gray-900">?<?php echo $total_artikel; ?></div>
                        </dd>
                    </div>
                </dl>
            </li>
            <li class="overflow-hidden rounded-xl border border-gray-200">
                <div class="flex items-center gap-x-4 border-b border-gray-900/5 bg-gray-50 p-6">
                    <div class="text-sm font-medium leading-6 text-gray-900">Pengguna</div>
                </div>
                <dl class="-my-3 divide-y divide-gray-100 px-6 py-4 text-sm leading-6">
                    <div class="flex justify-between gap-x-4 py-3">
                        <dt class="text-gray-500">Total</dt>
                        <dd class="flex items-start gap-x-2">
                            <div class="font-medium text-gray-900">?<?php echo $total_users; ?></div>
                        </dd>
                    </div>
                </dl>
            </li>
        </ul>
    </div>
</div>
</div>

```

Gambar 4. 13 Halaman Kodingan Branda

3. Halaman tambah data

Halaman ini menyediakan formulir yang dapat digunakan oleh admin untuk menambahkan data hama dan penyakit pada tanaman kopi.

Gambar 4. 14 Halaman Tambah Data

```

<?php
// Cek apakah form sudah disubmit
if ($_SERVER["REQUEST_METHOD"] == "POST") {
// koneksi database
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "kopi";

// Membuat koneksi
$conn = new mysqli($servername, $username, $password, $dbname);

// Memeriksa koneksi
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

// Mengambil data dari form
$kategori = $_POST['kategori'];
$idjudul = $_POST['idjudul']; // Ambil langsung dari input hidden
$judul = $_POST['judul'];
$deskripsi = $_POST['deskripsi'];
$konten = $_POST['konten'];
$gambar = "";
$video_link = $_POST['video_link'];
$sumber = $_POST['sumber'];

// Handle file upload untuk gambar
if (isset($_FILES['gambar']) && $_FILES['gambar']['error'] == UPLOAD_ERR_OK) {
$gambar = $_FILES['gambar']['name'];
$target_dir = "../uploads/";
$target_file = $target_dir . basename($gambar);

if (move_uploaded_file($_FILES['gambar']['tmp_name'], $target_file)) {
$gambar = $conn->real_escape_string($gambar);
} else {
$gambar = "";
}
}

// Sanitasi input
$judul = $conn->real_escape_string($judul);
$deskripsi = $conn->real_escape_string($deskripsi);
$konten = $conn->real_escape_string($konten);
$video_link = $conn->real_escape_string($video_link);
$sumber = $conn->real_escape_string($sumber);

// **Gunakan idjudul yang sudah didapatkan dari AJAX, jangan buat ulang**
$sql = "INSERT INTO artikel (idjudul, kategori, judul, deskripsi, konten, gambar, video, sumber)
VALUES ('$idjudul', '$kategori', '$judul', '$deskripsi', '$konten', '$gambar', '$video_link', '$sumber')";

if ($conn->query($sql) === TRUE) {
echo "<script>
alert('Artikel berhasil ditambahkan.');"
window.location.href = 'index.php?page=artikel';
</script>";
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
}
}

<div class="max-w-3xl mx-auto px-4 py-8">
<div class="text-xl font-bold text-gray-800 mb-6">Tambah Data</div>
<form action="" method="post" enctype="multipart/form-data" class="bg-white p-8 rounded-lg shadow-lg">
<div class="mb-4">
<div class="mb-4">
<label for="kategori" class="block text-gray-700 font-medium mb-2">Kategori</label>
<select id="kategori" name="kategori" required class="block w-full border border-gray-300 rounded-md p-2 focus:ring-indigo-500 focus:border-indigo-500">
<option value="">Pilih kategori</option>
<option value="nama">nama</option>
<option value="panggilan">Panggilan</option>
</select>
</div>
<div class="mb-4">
<div class="mb-4">
<label for="idjudul" class="block text-gray-700 font-medium mb-2">ID Judul</label>
<input type="text" id="idjudul" name="idjudul" value=""> </div>
<div class="mb-4">
<label for="judul" class="block text-gray-700 font-medium mb-2">Judul</label>
<input type="text" id="judul" name="judul" placeholder="Masukkan judul artikel" required class="block w-full border border-gray-300 rounded-md p-2 focus:ring-indigo-500 focus:border-indigo-500">
</div>
<div class="mb-4">
<div class="mb-4">
<label for="deskripsi" class="block text-gray-700 font-medium mb-2">Deskripsi</label>
<input type="text" id="deskripsi" name="deskripsi" placeholder="Masukkan deskripsi singkat" required class="block w-full border border-gray-300 rounded-md p-2 focus:ring-indigo-500 focus:border-indigo-500">
</div>
<div class="mb-4">
<div class="mb-4">
<label for="konten" class="block text-gray-700 font-medium mb-2">Konten</label>
<input type="text" id="konten" name="konten" rows="8" placeholder="Tulis uraian artikel..." required class="block w-full border border-gray-300 rounded-md p-2 focus:ring-indigo-500 focus:border-indigo-500">
</div>
<div class="mb-4">
<div class="mb-4">
<label for="gambar" class="block text-gray-700 font-medium mb-2">Upload Gambar</label>
<input type="file" id="gambar" name="gambar" required class="block w-full text-gray-700 border border-gray-300 rounded-md p-2 focus:ring-indigo-500 focus:border-indigo-500">
</div>
<div class="mb-4">
<div class="mb-4">
<label for="video_link" class="block text-gray-700 font-medium mb-2">Upload Video</label>
<input type="text" id="video_link" name="video_link" placeholder="Enter YouTube URL" class="block w-full border border-gray-300 rounded-md p-2 focus:ring-indigo-500 focus:border-indigo-500">
</div>
</div>
</div>

```

```

<div class="mb-4">
  <label for="sumber" class="block text-gray-700 font-medium mb-2">Sumber</label>
  <input type="text" id="sumber" name="sumber" placeholder="Masukkan Sumber artikel" required class="block w-full border border-gray-300 rounded-md p-2 focus:ring-indigo-500 focus:border-indigo-500">
</div>

<!-- Tombol Submit -->
<div class="flex items-center justify-end">
  <button type="submit" class="bg-indigo-600 hover:bg-indigo-700 text-white font-medium py-2 px-6 rounded-md focus:outline-none focus:ring-2 focus:ring-indigo-500">
    Tambah
  </button>
</div>
</form>
</div>

<script src="https://cdn.tiny.cloud/1/pqkfk7euyacwbjglo6x38o6ev2c4z04u20lg5j76n/tinymce/7/tinymce.min.js?referrerpolicy=origin"></script>
<script>
  tinymce.init({
    selector: '#konten',
    plugins: 'lists link image media table code fullscreen',
    toolbar: 'undo redo | formatselect | bold italic forecolor backcolor | alignleft aligncenter alignright alignjustify | bullist numlist outdent indent | link image media | table | code | fullscreen',
    height: 400,
    menubar: false,
    branding: false,
    content_style: 'body { font-family:Arial,sans-serif; font-size:14px }',
    toolbar_sticky: true,
    statusbar: false,
    autosave_ask_before_unload: false,
    autosave_interval: '30s',
    autosave_prefix: 'tinyce_',

    setup: function(editor) {
      editor.on('change', function() {
        editor.save(); // Simpan konten setiap kali terjadi perubahan
      });
    }
  });
</script>
<script>
  document.getElementById('kategori').addEventListener('change', function() {
    var kategori = this.value;
    if (kategori) {
      fetch('get_idjudul.php?kategori=' + kategori)
        .then(response => response.text())
        .then(data => {
          document.getElementById('idjudul').value = data;
        })
        .catch(error => console.error('Error:', error));
    } else {
      document.getElementById('idjudul').value = "";
    }
  });
</script>

```

Gambar 4. 15 Halaman kodingan tambah data

4. Halaman ubah data

Halaman ini menyediakan formulir yang dapat memperbarui data pada hama dan penyakit tanaman kopi.

The screenshot shows a web form for updating data. It contains the following elements:

- Kategori:** A dropdown menu with 'Hama' selected.
- ID Judul:** A text input field with '1000' entered.
- Judul:** A text input field with 'Penggerek Buah Kopi' entered.
- Deskripsi:** A text input field with 'Hama penggerek buah kopi (Hypothenemus hampei) menggerek buah muda dan buah yang sudah' entered.
- Konten:** A rich text editor containing a paragraph and a numbered list:
 1. Pengendalian secara kultur teknis, yaitu memuat dan tidak PDBK. Caranya dengan petik buah, seleksi, dan inspeksi. Petik buah adalah panen buah (buah berumur) yang memuat buah kopi yang terinfeksi hama penggerek. Buah kopi hasil dari petik buah, seleksi, dan inspeksi dimasukkan ke dalam karung goni dan disimpan dalam suhu pendinginan 10 menit.
 2. Pengendalian secara hayati. Caranya dengan menggunakan parasitoid *Campoplex chrysoscelus* dan jamur patogen *Solenaspora bombycis*. Penggunaan jamur B. Berasada di aplikasi dengan menggunakan perangkap, sehingga jamur ini aplikasi per musuh panen, penggunaan jamur B. Berasada dalam bentuk tepung yang dicampur dengan dosis 100 g per 10 liter per hektar.
 3. Pengendalian menggunakan senyawa kimia (insektisida). Caranya dengan menggunakan senyawa kimia hama penggerek buah kakao hasil formulasi pua...
- Upload Gambar:** A file upload field with a 'Choose File' button and 'No file chosen' text. Below it is a small image of a coffee fruit.
- YouTube Video Link:** A text input field with 'https://www.youtube.com/embed/GR7W0c-XNDwU' entered.
- Sumber:** A text input field with 'YouTube : nr_24' entered.
- Simpan:** A blue button at the bottom right.

Gambar 4. 16 Halaman ubah data


```

<?php
$id = $_GET['id'];

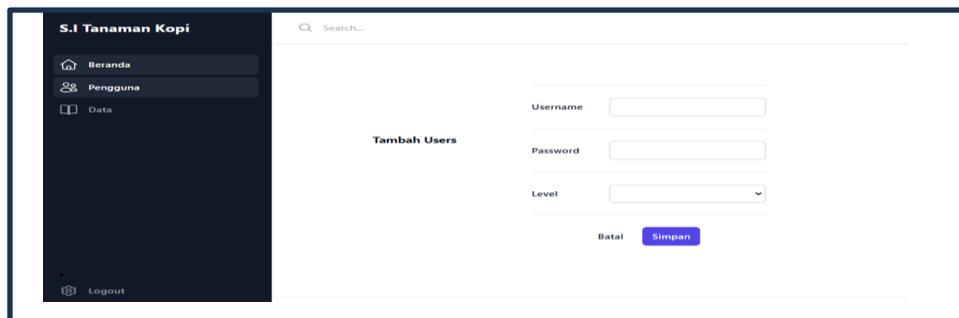
$sql = "DELETE FROM artikel WHERE id='$id'";
if ($conn->query($sql) === TRUE) {
    echo "<script>
        alert('Artikel berhasil dihapus. ');
        window.location.href = 'index.php?page=artikel';
    </script>";
}
$conn->close();

```

Gambar 4. 19 Halaman Kodingan hapus data

6. Halaman tambah data pengguna

Halaman ini menyediakan formulir yang dapat digunakan pengguna untuk menambahkan data admin.



Gambar 4. 20 Halaman tambah user admin

```

<?php
// Proses jika tombol simpan ditekan
if (isset($_POST['simpan'])) {
    $username = $_POST['username'];
    $pass = md5($_POST['pass']);
    $level = $_POST['level'];

    // Validasi Data Users
    $sql = "SELECT * FROM users WHERE username='$username'";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        <div class="alert alert-danger alert-dismissible fade show">
            <button type="button" class="close" data-dismiss="alert">&times;</button>
            <strong>Data Tidak Tersimpan Karena Username Sudah Digunakan</strong>
        </div>
    } else {
        // Proses Simpan Setelah Di Validasi
        $sql = "INSERT INTO users VALUES (Null, '$username', '$pass', '$level')";
        if ($conn->query($sql) === TRUE) {
            echo "Tambah Data Berhasil";
        }
    }
}

$ii = 1;
$sql = "SELECT * FROM users ORDER BY id ASC";
$result = $conn->query($sql);
while ($row = $result->fetch_assoc()) {
    <tr class="hover:bg-gray-50">
        <td class="whitespace-nowrap px-6 py-4 text-sm font-medium text-gray-900">?php echo $ii++; </td>
        <td class="whitespace-nowrap px-6 py-4 text-sm text-gray-900">?php echo $row['username']; </td>
        <td class="whitespace-nowrap px-6 py-4 text-sm text-gray-900">?php echo $row['pass']; </td>
        <td class="whitespace-nowrap px-6 py-4 text-sm text-gray-900">?php echo $row['level']; </td>
        <td class="whitespace-nowrap px-6 py-4 text-right text-sm font-medium">
            <a href="page=users&action=update&id=?php echo $row['id'];" >? " class="text-indigo-600 hover:text-indigo-900">Edit</a>
            <a onclick="return confirm('Yakin Ingin Menghapus Data ini?');" href="page=users&action=hapus&id=?php echo $row['id'];" >? " class="text-red-600 hover:text-red-900 ml-2">Hapus</a>
        </td>
    </tr>
}
}
$conn->close();
}

```

Gambar 4. 21 Halaman Kodingan tambah data pengguna

7. Halaman ubah user admin

Halaman ini menyediakan formulir yang dapat digunakan pengguna untuk mengubah data user admin yang ingin diubah.

Gambar 4. 22 Halaman ubah data user admin

```

<?php
// memanggil Data yang mau di Edit
$id = $_GET['id'];

if (isset($_POST['update'])) {
    // Ambil Data dari Data yang Mau di Update
    $level = $_POST['level'];

    // Proses Update Data Users
    $sql = "UPDATE users SET level=$level WHERE id=$id";
    if ($conn->query($sql) == TRUE) {
        echo "Berhasil";
    }
}

$sql = "SELECT * FROM users WHERE id=$id";
$result = $conn->query($sql);
$row = $result->fetch_assoc();
}

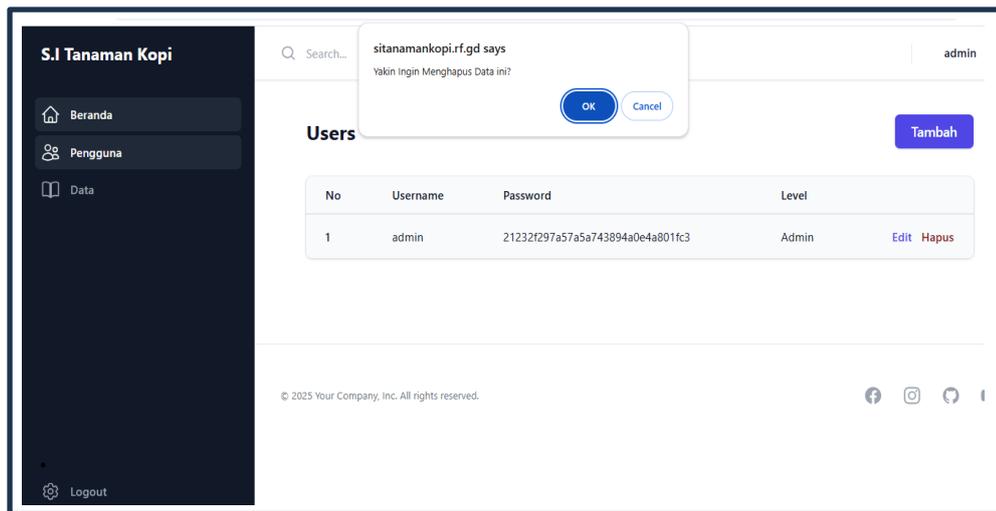
<div action="" method="POST">
<div class="space-y-4">
<div class="flex-center justify-center grid grid-cols-2">
<div class="font-weight-bold text-gray-900 text-center">Edit Users</div>
<div class="flex flex-wrap justify-between text-gray-900 font-weight-normal">
<div class="w-full"></div>
</div>
<div class="border border-gray-200 rounded-md p-4">
<div class="flex justify-between">
<div class="font-weight-normal text-gray-900">Username</div>
<input type="text" value="{row['username']}"/>
</div>
<div class="flex justify-between">
<div class="font-weight-normal text-gray-900">Password</div>
<input type="password"/>
</div>
<div class="flex justify-between">
<div class="font-weight-normal text-gray-900">Level</div>
<select class="form-control" value="{row['level']}"/>
</div>
</div>
<div class="flex justify-between">
<div class="font-weight-normal text-gray-900">Batal</div>
<div class="font-weight-normal text-gray-900">Update</div>
</div>
</div>

```

Gambar 4. 23 Halaman Kodingan hapus data

8. Halaman hapus data user admin

Halaman ini menyediakan formulir yang dapat digunakan pengguna untuk menghapus data user admin.



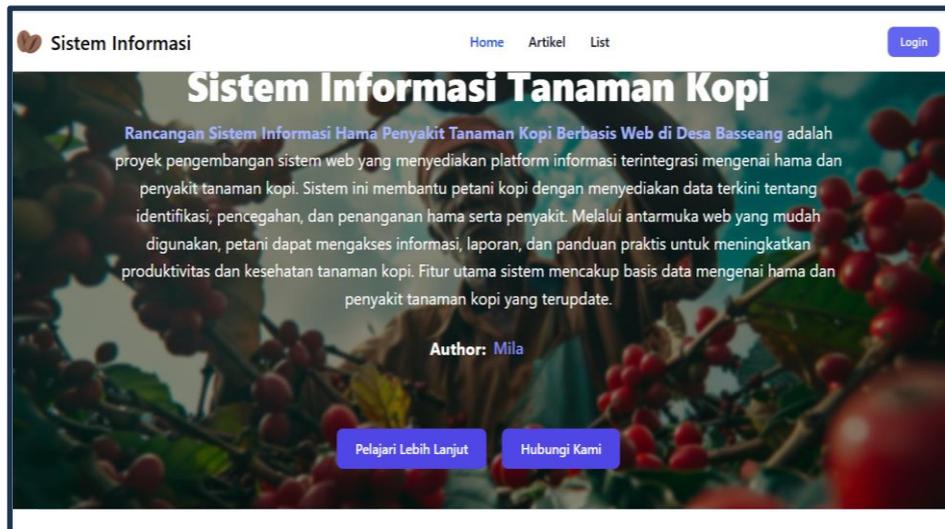
Gambar 4. 24 Halaman hapus data pengguna

```
<?php
$id = $_GET['id'];
$sql = "DELETE FROM users WHERE id='$id'";
if ($conn->query($sql) === TRUE) {
    echo 'Hapus Data Berhasil';
}
$conn->close();
```

Gambar 4. 25 Halaman Kodingan hapus data pengguna

9. Halaman home

Halaman home ini akan menampilkan halaman utama di web. Dimana user membuka halaman website dan untuk memilih penyakit user memilih baca atau jenis penyakit yang akan ditampilkan.



Gambar 4. 26 Halaman home

```

<?php
$sql = "SELECT idjudul, judul, deskripsi FROM artikel";
$result = $conn->query($sql);

$sartikel = array();
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $deskripsi = $row['deskripsi'];
        $kata = explode(' ', $deskripsi);
        if (count($kata) > 10) {
            $kata = array_slice($kata, 0, 20);
            $deskripsi = implode(' ', $kata) . '....';
        }
        $row['deskripsi'] = $deskripsi;
        $artikel[] = $row;
    }
}
?>
<?php foreach ($artikel as $item) : ?>
<div class="my-4 bg-gray-100 p-4 rounded-lg shadow-lg hover:shadow-2xl transition-all duration-300 ease-in-out transform hover:scale-105 fade-in">
<div class="font-bold text-indigo-600 mb-2">

<a href="artikel.php" <php echo urlencode($item['idjudul']); ?>" class="hover:text-indigo-700 transition duration-200 ease-in-out"><php echo htmlspecialchars($item['judul']); </a>
</div>
<div class="text-gray-600 mb-2">
<php echo htmlspecialchars($item['deskripsi']); ?>
</div>
<div class="bg-indigo-700 w-fit rounded-lg hover:bg-indigo-800 transition duration-300 ease-in-out transform hover:scale-105">
<a href="artikel.php" <php echo urlencode($item['idjudul']); ?>" class="p-2 text-white font-semibold block text-center transform hover:scale-105 transition duration-300 ease-in-out">Baca Selengkapnya</a>
</div>
<div class="border-green-700 transition duration-300 ease-in-out transform hover:scale-105">
</div>
</php endforeach; ?>

```

Gambar 4. 27 Halaman kodingan home

10. Halaman *tampilan informasi hama atau penyakit*

Halaman tampilan informasi hama atau penyakit akan menampilkan hama atau penyakit yang telah dipilih, dan halaman tampilan informasi hama atau penyakit ini juga dapat melihat video tentang penjelasan hama atau penyakit pada tanaman kopi.


```

main {
    flex: 1;
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
}

h1,
h2,
h3,
h4,
h5,
h6 {
    margin-bottom: 1rem;
}

.article-image {
    flex: 1 1 30%;
    max-width: 30%;
    overflow: hidden;
    border-radius: 8px;
    transition: transform 0.3s ease;
}

.article-image img {
    width: 100%;
    height: auto;
    object-fit: cover;
    transition: transform 0.3s ease;
}

.article-image:hover img {
    transform: scale(1.05);
}

.article-content {
    flex: 1 1 65%;
    max-width: 65%;
}

.video-container {
    margin-top: 20px;
    position: relative;
    padding-bottom: 56.25%;
    /* 16:9 Aspect Ratio */
    height: 0;
    overflow: hidden;
    max-width: 100%;
    border-radius: 8px;
}

.video-container iframe {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
}

.lexl-description {
    color: #4f4a4a;
    font-size: 1.2rem;
}

/* Header style */
#header {
    width: 100%;
    min-height: 600px;
    background-color: transparent;
    position: relative;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}

#header .overlay {
    position: absolute;
    inset: 0;
    background-image: url('uploads/bgmain.jpg');
    background-size: cover;
    background-position: center;
    filter: blur(2px);
    z-index: 0;
}

#header .overlay .dark overlay {
    position: absolute;
    inset: 0;
    background-color: rgba(0, 0, 0, 0.4);
    /* Overlay Gelap */
    z-index: 0;
}

#header .content {
    z-index: 10;
    text-align: center;
    color: white;
    padding: 1rem;
}

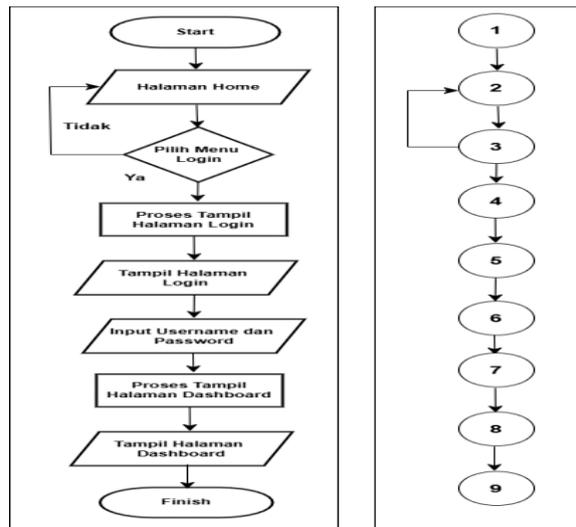
```


white box dan pengujian *Black box*.

1. White box

a. White box testing login

1) Flowchart login dan Flowgraph login



Gambar 4. 30 flowchart login dan Flowgraph login

Perhitungannya dapat diselesaikan sebagai berikut setelah meninjau

Flowchart dan *Flowgraph* di atas:

2) Menghitung kompleksitas *cyclomatic* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 9$$

$$N \text{ (node)} = 9$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 9 - 9 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

3) Berdasarkan perhitungan kompleksitas *cyclomatic* dari *flowgraph* di atas, memiliki $= 2$

4) Jalan independen pada *flowgraph* ini adalah:

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9$$

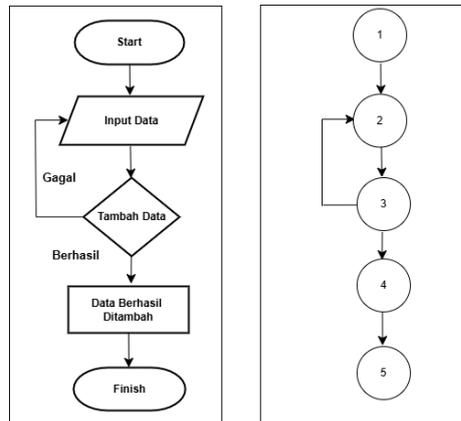
5) Grafik matriks proses deteksi penyakit dan hama tanaman kopi

Tabel 4. 2 Grafik matriks login

	1	2	3	4	5	6	7	8	9	E - 1
1		1								$1 - 1 = 0$
2			1							$1 - 1 = 0$
3		1		1						$2 - 1 = 1$
4					1					$1 - 1 = 0$
5						1				$1 - 1 = 0$
6							1			$1 - 1 = 0$
7								1		$1 - 1 = 0$
8									1	$2 - 1 = 1$
9										$1 - 1 = 0$
	SUM (E + 1)									$1 + 1 = 2$

b. *White box testing* tambah data

1) *Flowchart* tambah data dan *Flowgraph* tambah data



Gambar 4. 31 *Flowchart* tambah data dan *Flowgraph* tambah data

Perhitungannya dapat diselesaikan sebagai berikut setelah meninjau

Flowchart dan *Flowgraph* di atas:

2) Menghitung kompleksitas *cyclomatic* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

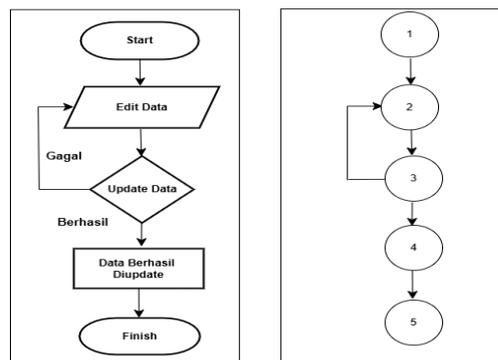
- 3) Berdasarkan perhitungan kompleksitas *cyclomatic* dari *flowgraph* di atas, memiliki = 2
- 4) Jalan independen pada *flowgraph* ini adalah:
- Path 1 = 1 – 2 – 3 – 2
- Path 2 = 1 – 2 – 3 – 4 – 5
- 5) Grafik matriks proses deteksi penyakit dan hama tanaman kopi

Tabel 4. 3 Grafik matriks tambah data

	1	2	3	4	5	E - 1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						$1 - 1 = 0$
	SUM (E + 1)					$1 + 1 = 2$

c. White box testing ubah data

1) **Flowchart ubah data**



Gambar 4. 32 Flowchart ubah data dan Flowgraph ubah data

Perhitungannya dapat diselesaikan sebagai berikut setelah meninjau Flowchart dan Flowgraph di atas:

- 1). Menghitung kompleksitas *cyclomatic* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\text{edge}) = 5$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat } \textit{node}) = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

2). Berdasarkan perhitungan kompleksitas *cyclomatic* dari *flowgraph* di atas, memiliki = 2

3). Jalan independen pada *flowgraph* ini adalah:

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

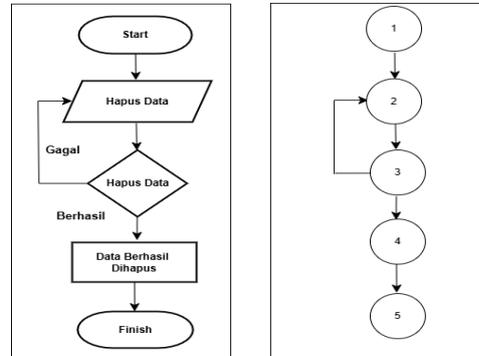
4). Grafik matriks proses deteksi penyakit dan hama tanaman kopi

Tabel 4. 4 Grafik matriks ubah data

	1	2	3	4	5	E - 1
1		1				1 - 1 = 0
2			1			1 - 1 = 0
3		1		1		2 - 1 = 1
4					1	1 - 1 = 0
5						1 - 1 = 0
	SUM (E + 1)					1 + 1 = 2

d. White box testing hapus data

1). *Flowchart* hapus data dan *Flowgraph* hapus data



Gambar 4. 33 *Flowchart* hapus data dan *Flowgraph* hapus data

Perhitungannya dapat diselesaikan sebagai berikut setelah meninjau *Flowchart* dan *Flowgraph* di atas:

1). Menghitung kompleksitas *cyclomatic* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

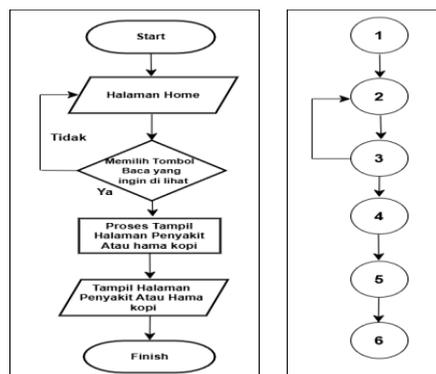
- 2). Berdasarkan perhitungan kompleksitas *cyclomatic* dari *flowgraph* di atas, memiliki = 2
- 3). Jalan independen pada *flowgraph* ini adalah:
- Path 1 = 1 – 2 – 3 – 2
- Path 2 = 1 – 2 – 3 – 4 – 5
- 4). Grafik matriks proses deteksi penyakit dan hama tanaman kopi

Tabel 4. 5 Grafik matriks hapus data

	1	2	3	4	5	E - 1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						$1 - 1 = 0$
	SUM (E + 1)					$1 + 1 = 2$

e. White box testing user

1) *Flowchart home dan Flowgraph home*



Gambar 4. 34 *Flowchart* halaman home dan *Flowgraph* home

Perhitungannya dapat diselesaikan sebagai berikut setelah meninjau *Flowchart* dan *Flowgraph* di atas:

- 1). Menghitung kompleksitas *cyclomatic* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 6 - 6 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

2). Berdasarkan perhitungan kompleksitas *cyclomatic* dari *flowgraph* di atas, memiliki = 2

3). Jalan independen pada *flowgraph* ini adalah:

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6$$

4). Grafik matriks proses deteksi penyakit dan hama tanaman kopi

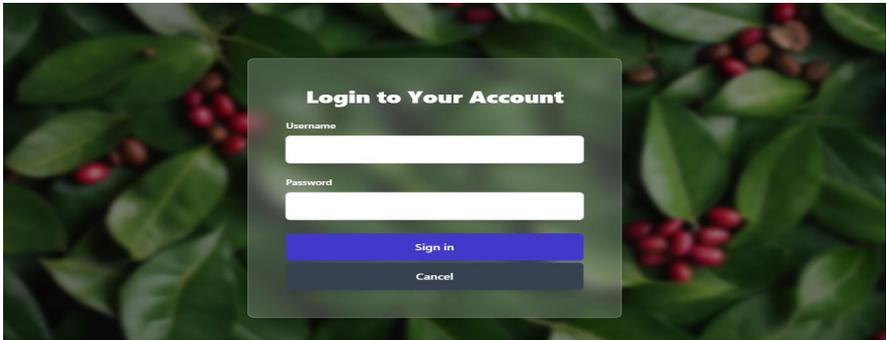
Tabel 4. 6 Grafik matriks home

	1	2	3	4	5	6	E - 1
1		1					1 - 1 = 0
2			1				1 - 1 = 0
3				1			1 - 1 = 0
4			1		1		2 - 1 = 1
5						1	1 - 1 = 0
6							1 - 1 = 0
	SUM (E + 1)						1 + 1 = 2

1. Black box

a. Black box testing kesalahan username dan password

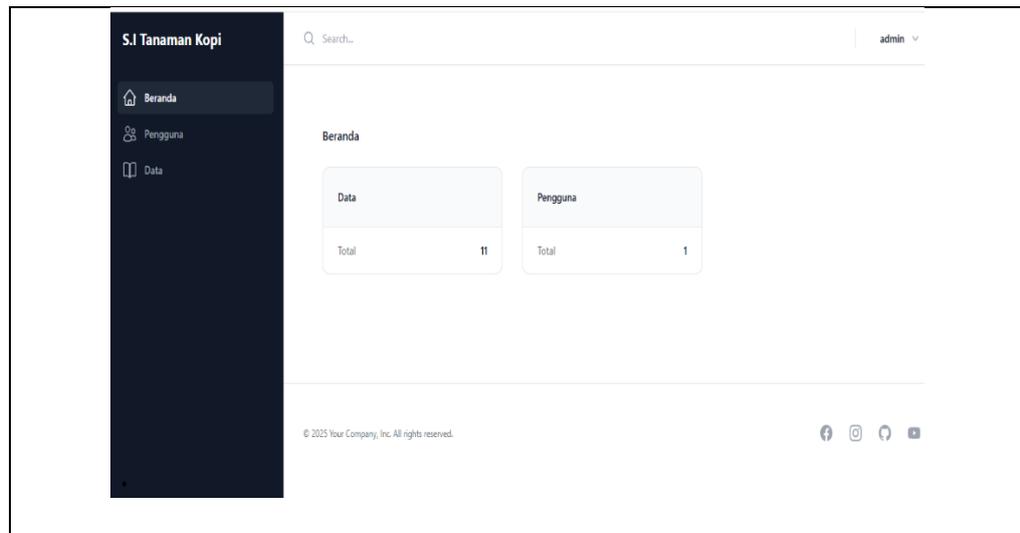
Tabel 4.7 Black box testing kesalahan username dan password

Tes faktor	Hasil	Keterangan
Memasukkan username atau <i>password</i> yang sesuai	√	Berhasil, ketika username atau <i>password</i> yang sesuai, tampil <i>loginfailed</i>
“Screenshot”		
		

b. Blackbox testing login berhasil

Tabel 4.8 Black box testing login berhasil

Tes faktor	Hasil	Keterangan
Memasukkan username atau <i>password</i> berhasil	√	Sistem berhasil menampilkan halaman dashboard.
“Screenshot”		

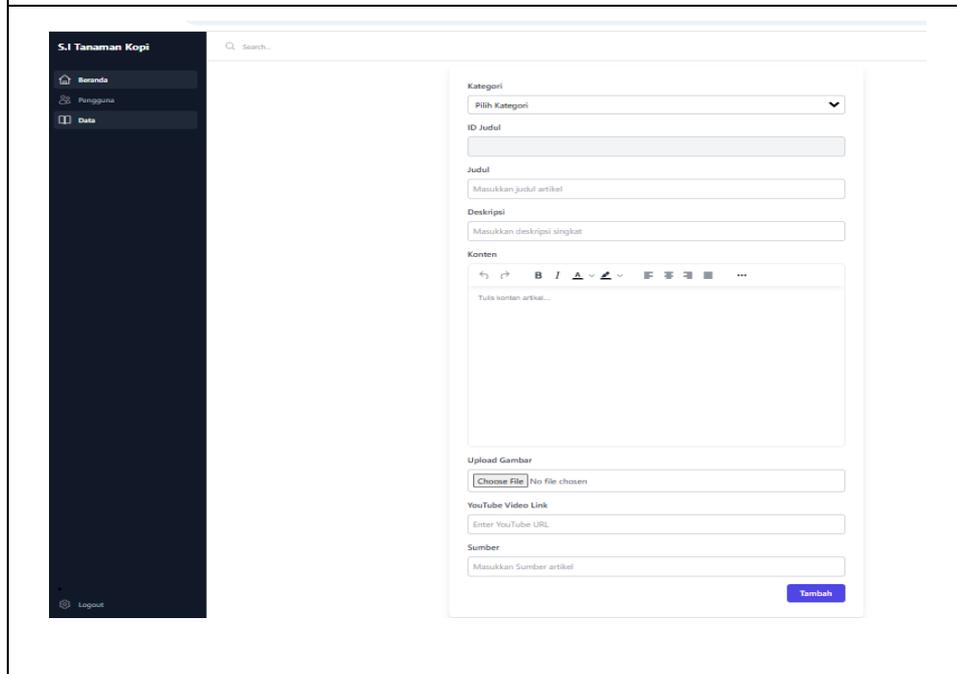


c. Black box testing halaman tambah data

Tabel 4.9 Black box testing tambah data

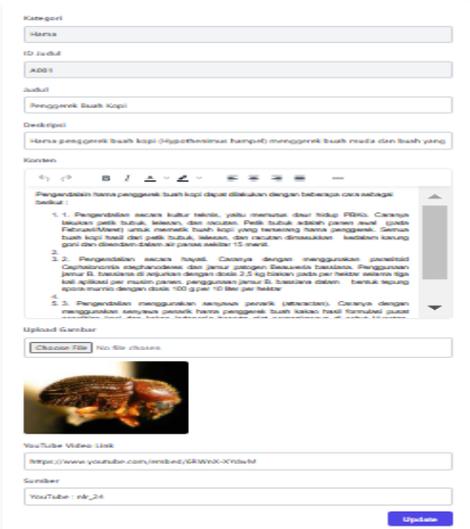
Tes faktor	Hasil	Keterangan
<i>Admin</i> mengisi <i>form</i> tambah dan menekan tombol tambah	√	Sukses, data telah berhasil ditambahkan.

“Screenshot”



d. Black box testing halaman ubah data

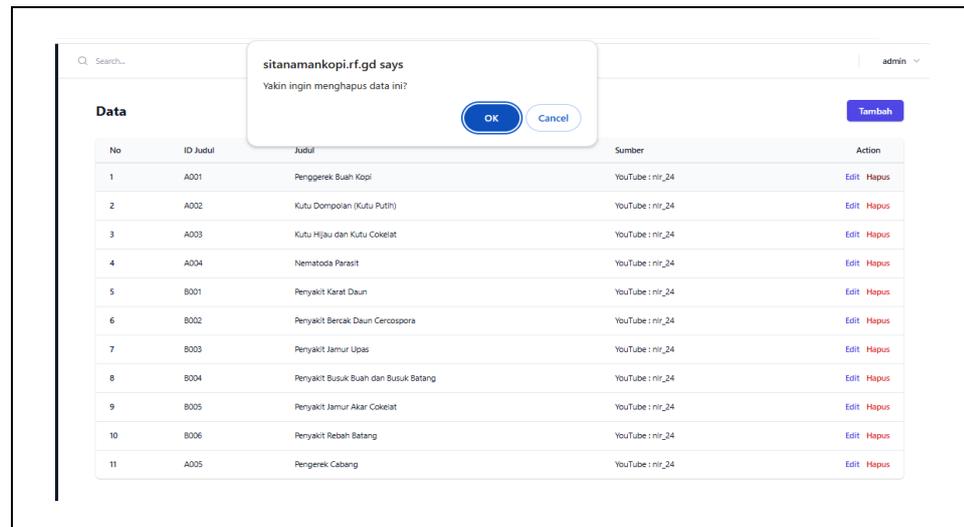
Tabel 4.10 Black box testing ubah data

Tes faktor	Hasil	Keterangan
Admin mengisi form ubah data dan menekan tombol update	√	Sukses, sebuah notifikasi muncul bahwa data telah berhasil diubah.
“Screenshot”		
		

e. Black box testing halaman hapus data

Tabel 4.11 Black box testing hapus data

Tes faktor	Hasil	Keterangan
Admin menghapus salah satu data	√	Sukses, sebuah notifikasi muncul bahwa data telah berhasil dihapus.
“Screenshot”		



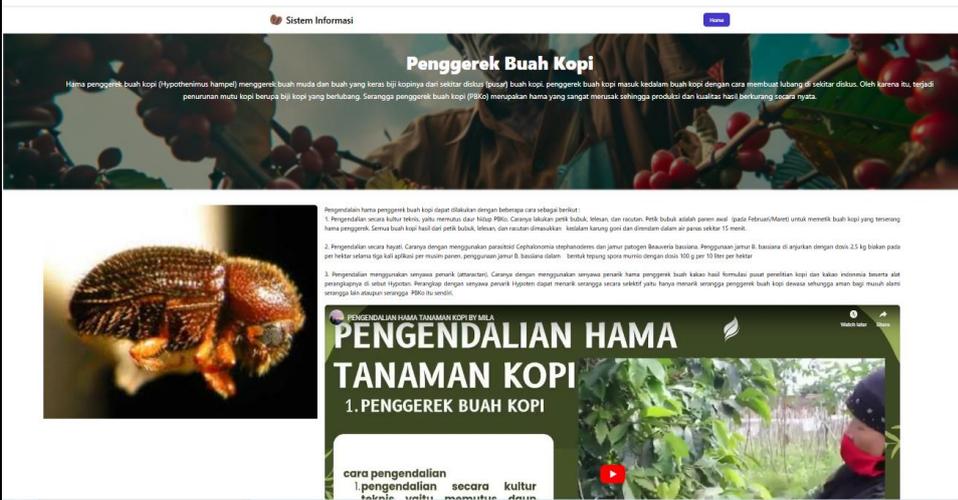
f. Black box testing halaman home

Tabel 4.12 Black box testing home

Tes-faktor	Hasil	Keterangan
User ketika mengakses aplikasi	√	Sukses, tampil-halaman home
“Screenshot”		

g. Black box testing halaman informasi hama atau penyakit

Tabel 4.13 *Black box testing* halaman informasi hama atau penyakit

Tes faktor	Hasil	Keterangan
User menekan tombol baca atau memilih jenis penyakit atau hama kopi yang ingin dilihat	√	Sukses, tampil halaman informasi hama atau penyakit jenis penyakit atau hama kopi
“Screenshot”		
		

D. Rancangan Database

Berikut ini adalah tabel-tabel yang digunakan dalam aplikasi pramuka berbasis web:

1. Tabel Artikel

Tabel 4.14 *database* Artikel

No.	Nama Item Data	Type	Lebar	Keterangan
1.	Id	<i>int</i>	11	<i>AUTO_INCREMENT</i>
2.	Idjudul	<i>varchar</i>	20	-
3.	Kategori	<i>varchar</i>	10	-
4.	Judul	<i>text</i>	-	-
5.	Deskripsi	<i>text</i>	-	-
6.	Konten	<i>text</i>	-	-
7.	Gambar	<i>varchar</i>	255	-
8.	Video	<i>varchar</i>	255	-

No.	Nama Item Data	Type	Lebar	Keterangan
9.	Sumber	<i>varchar</i>	255	-

2. Tabel users

Tabel 4.15 database users

No.	Nama Item Data	Type	Lebar	Keterangan
1.	id	<i>int</i>	11	<i>AUTO_INCREMENT</i>
2.	username	<i>varchar</i>	20	-
3.	pass	<i>Text</i>	-	-
4.	<i>level</i>	<i>varchar</i>	10	-

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Hasil penelitian yang telah dilakukan, berikut adalah kesimpulan yang dapat diperoleh:

1. Aplikasi ini berhasil menampilkan informasi terkait kegiatan penyakit dan hama tanaman kopi dengan jelas dan terstruktur, memudahkan akses bagi pengelola dan petani.
2. Aplikasi ini dibangun menggunakan *Visual Studio Code* sebagai *text editor*, MySQL sebagai sistem manajemen *database*, dan bahasa pemrograman *php* dan *html*.
3. Pada pengujian Blackbox dan Whitebox memastikan struktur internal kode dan logika berjalan sesuai dengan yang dirancang. Seluruh tombol yang ada pada aplikasi panduan lengkap hidroponik dapat berjalan seluruhnya tanpa ada eror

B. Saran

Pada penelitian ini, penulis menyadari adanya beberapa kekurangan yang memerlukan perbaikan dan pengembangan lebih lanjut. Untuk itu, penulis menawarkan beberapa saran untuk penelitian yang akan datang, yaitu:

1. Melakukan pembaruan secara berkala terhadap data penyakit dan hama tanaman kopi.

2. Aplikasi ini dapat ditingkatkan dengan mengembangkan menu dan fungsinya serta menyertakan elemen interaktif agar lebih menarik.

DAFTAR PUSTAKA

- Asmara, J. (2019)Adiyanti, R., Sulaksana, P. T., Syahidin, Y., & Hidayati, M. (2021). Perancangan Sistem Informasi Indeks Penyakit Rawat Inap Menggunakan Microsoft Visual Studio. *Jurnal Teknologi Dan Manajemen Informatika*, 7(1), 10–19.
- Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian black box testing pada aplikasi action & strategy berbasis android dengan teknologi phonegap. *String (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 3(2), 206–210.
- Firman, F., Fauziyah, F., & Komalasari, R. T. (2021). Aplikasi Pengingat Dan Pendataan Kenaikan Golongan Gaji Berbasis Web Menggunakan Metode White Box Testing dan Black Box Testing. *Jurnal Teknologi Informasi*, 7(1), 50–57.
- Fitri, R., Kom, S., & Kom, M. (2020). *Pemrograman Basis Data Menggunakan MySQL*. Deepublish.
- Hakim, Z., & Meilina, P. (2022). Sistem Informasi Akademik Berbasis Webiste. *JUST IT: Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer*, 12(3), 32–37.
- Haviluddin, H. (2016). Memahami Penggunaan UML (Unified Modelling Language). *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 6(1), 1–15.
- Murtadho, A. (2022). Sistem Informasi Perkiraan Produksi Kopi Robusta di Perusahaan Daerah Perkebunan Jember. *Jurnal Aplikasi Bisnis Dan Manajemen (JABM)*, 8(1), 192.
- Nasution, W. R. H., Nasution, M. I. P., & Sundari, S. S. A. (2022). 9 Pendapat Ahli Mengenai Sistem Informasi Manajemen. *Jurnal Inovasi Penelitian*, 3(4), 5893–5896.
- Pudji Pahardjo 2017. Berkebun Kopi. Diedit oleh Atiatul Mu'min. Jakarta:Swadaya
- Setiawan, A. A., Lumenta, A. S. M., & Sompie, S. R. U. A. (2019). *Rancang Bangun Aplikasi Unsrat E-Catalog*.
- Wahid, A. A. (2020). Analisis metode waterfall untuk pengembangan sistem informasi. *J. Ilmu-Ilmu Inform. Dan Manaj. STMIK*, No. November, 1(1), 1–5.
- Wahyudi, J., Elfianty, L., Nur'aini, H., & Andriani, E. (2017). Sistem Informasi Penanggulangan Hama dan Penyakit Tanaman Bagi Penyuluh Pertanian. *Prosiding Semnas Darmajaya*.

Wardhani, Y. K. (2022). aplikasi absensi guru dan karyawan berbasis web pada mts negeri 1 lumajang. *Jurnal Teknik Industri, Sistem Informasi Dan Teknik Informatika*, 1(2), 93–110.