PENGEMBANGAN *FRONTEND SISTEM AGRI-INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE* BERBASIS IOT MENGGUNAKAN *LIBRARY REACT.JS*

(SKRIPSI)

Oleh

ALPRIEALIAN RENANDO

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2024

ABSTRAK

PENGEMBANGAN FRONTEND SISTEM AGRI-INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE BERBASIS IOT MENGGUNAKAN LIBRARY REACT.JS

Oleh

ALPRIEALIAN RENANDO

Penelitian ini bertujuan untuk mengembangkan antarmuka frontend dari *Agri-Intelligence Control System* (ICS) pada *Smart Greenhouse* berbasis *Internet of Things (IoT)* menggunakan *React.js*. Sistem ini memungkinkan pemantauan kondisi lingkungan di dalam *greenhouse* secara *real-time*, yang meliputi parameter suhu, kelembapan, kecepatan dan arah angin, serta beberapa parameter yang terkait dengan pasokan nutrisi tanaman. Data yang dikumpulkan melalui sensor IoT diteruskan melalui API, kemudian divisualisasikan dalam bentuk grafik dan tabel pada *dashboard* yang dapat diakses dengan mudah. Selain itu, sistem yang dibangun ini juga mendukung fitur pengunduhan data untuk di analisis lebih lanjut, sehingga mempermudah pengelolaan *agrikultur* dalam *greenhouse*. Dalam pengembangannya, penelitian ini menggunakan metode *Scrum* yang mencakup tahap perencanaan, implementasi, dan pengujian *blackbox* untuk menjamin performa sistem. Hasil pengujian menunjukkan bahwa data ditampilkan secara *real-time* dengan latensi rata-rata 1,5 detik, dan sistem berhasil menjaga stabilitas selama 48 jam pengujian tanpa masalah. *React.js* terbukti mampu menciptakan antarmuka yang responsif dan interaktif, sehingga pengguna dapat dengan mudah memantau kondisi *greenhouse*. *Frontend* berbasis *React.js* ini diharapkan dapat membantu petani dan peneliti dalam mengelola *greenhouse* secara efisien dan berkelanjutan. Sistem *Agri-Intelligence* ini memberikan solusi praktis yang mendukung produksi agrikultur yang lebih optimal dan adaptif terhadap perubahan lingkungan.

Kata kunci: *Greenhouse, IoT, React.js, Scrum, frontend, Agri-Intelligence*

ABSTRACT

DEVELOPMENT OF THE FRONTEND FOR AN AGRI-INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE BASED ON IOT UTILIZING THE REACT.JS LIBRARY

By

ALPRIEALIAN RENANDO

This study aims to develop a frontend interface for the Agri-Intelligence Control System (ICS) Internet of Things (IoT)-based Smart Greenhouse using React.js. This system enables real-time monitoring of environmental conditions within the greenhouse, including parameters such as temperature, humidity, wind speed and direction, as well as some parameters related to plant nutrients supply. Collected data through IoT sensors at the backend is transmitted via an API and then visualized as graphs and tables on an accessible dashboard. Additionally, the system supports data download for further analysis, facilitating the efficient agricultural management of the greenhouse., This study employs the Scrum methodology, encompassing planning, implementation, and blackbox testing to ensure the performance of the system. Testing results indicate that data is displayed in real-time with an average latency of 1.5 seconds, and the system-maintained stability for 48 hours without interruptions. React.js has proven effective in creating a responsive and interactive interface, enabling users to monitor greenhouse conditions with ease. This React.js-based frontend is expected to assist farmers and researchers in efficiently and sustainably managing greenhouse operations. The Agri-ICS provides a practical solution that supports more optimal agricultural production and adapts well to environmental changes.

Keywords*: Greenhouse, IoT, React.js, Scrum, frontend, Agri-Intelligence.*

PENGEMBANGAN FRONTEND SISTEM AGRI-INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE BERBASIS IOT MENGGUNAKAN LIBRARY REACT.JS

Oleh

ALPRIEALIAN RENANDO

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK

PROGRAM STUDI TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2024

Judul Skripsi : **PENGEMBANGAN *FRONTEND* SISTEM *AGRI-INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE* BERBASIS IOT MENGGUNAKAN *LIBRARY* REACT.JS**

Nama Mahasiswa : Alpriealian Renendo

Nomor Pokok Mahasiswa : 2055031011

Program Studi : Teknik Elektro

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama	Pembimbing Pendamping
Dr. Ing. Ardian Ulvan, S.T., M.Sc.	**Aryanto, S.T., M.T.**
NIP 19731128 199903 1 005	NIP 19900621 201903 1 011

2. Mengetahui

Ketua Jurusan Teknik Elektro	Ketua Program Studi Teknik Elektro
Herlinawati, S.T., M.T.	**Sumadi, S.T., M.T.**
NIP 19710314 199903 2 001	NIP 19731104 200003 1 001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. Ing. Ardian Ulvan, S.T., M.Sc.** .....

Sekretaris : **Aryanto, S.T., M.T.** .....

Penguji : **Ir. Meizano Ardhi Muhammad, S.T., M.T.** .....

2. Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.
NIP 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : **25 Oktober 2024**

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepanjang pengetahuan saya tidak terdapat atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam daftar pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi akademik sesuai dengan hukum yang berlaku.

Bandar Lampung, 4 November 2024

Alpriadian Rolando

NPM. 2055031011

RIWAYAT HIDUP

Penulis lahir di Metro, 01 November 2001. Penulis merupakan anak Ketiga dari Ketiga bersaudara dari pasangan Bapak Alm Agustianto dan Ibu Agusriah. Pendidikan penulis SDN 2 Yukum Jaya Lampung Tengah pada tahun 2007 hingga 2013, MTs Negeri 1 Lampung Tengah pada tahun 2013 hingga 2016, dan SMAN 1 Terbanggi Besar pada tahun 2016 hingga 2019.

Penulis menjadi mahasiswa Jurusan Teknik Elektro Universitas Lampung pada tahun 2020 melalui jalur MANDIRI (Perguruan Tinggi Negeri). Selama menjadi mahasiswa, penulis pernah menjadi anggota Himpunan Mahasiswa Teknik Elektro (HIMATRO) sebagai anggota divisi Sosial Wirausaha pada periode 2021 dan periode 2022. Penulis tertarik dalam bidang teknologi khususnya web dan otomasi.

PERSEMBAHAN

Dengan Ridho Allah SWT

Teriring shalawat kepada Nabi Muhammad SAW

kupersembahkan Skripsi ini sebagai tanda terimakasih kepada:

Papa dan Mama Tercinta

Alm. Agustianto dan Agusriah

Dan Kakak Tersayang

Alpriealian Hernandi Dan Alfriealian Tinarina

Serta keluarga besarku yang banyak berjasa Selalu memberikan dukungan, kasih sayang, dan dorongan moral yang tak terhingga selama perjalanan studi saya. Atas doa dan semangat yang selalu mengiringi setiap langkah saya dalam mengejar mimpi terima kasih untuk do'a dan dukungan yang telah diberikan.

MOTTO HIDUP

"Khoirunnas Anfa'uhum Linnas, Sebaik-baik manusia adalah yang bermanfaat bagi manusia lainnya."

(HR. Ahmad)

"Hatiku tenang setelah mengetahui apa yang melewatkanku tidak akan pernah menjadi takdirku, dan apa yang ditakdirkan untukku tidak akan pernah melewatkanku"

(Umar bin Khatab)

"Cukuplah Allah menjadi penolong kami dan sebaik-baik pelindung"

(Q.S Al Imran:173)

"Jangan biarkan rencana yang kita ekspetasikan, merusak rencana indah yang Tuhan siapkan."

(Penulis)

SANWACANA

Alhamdulillahirabbil'alamin. Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayat-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi/tugas akhir ini dengan judul "Implementasi Teknologi Intelligence Control System Dalam Monitoring Dan Pengelolaan Sistem Aktuator Smart *Greenhouse* Untuk Budidaya Tanaman Hortikultura". Dalam pelaksanaan dan pembuatan skripsi ini penulis menerima dukungan baik secara moril maupun materil yang sangat berharga dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih ke semua pihak yang telah membantu, khususnya kepada:

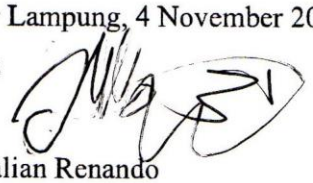
1. Kedua orang tua tercinta dan seluruh keluarga penulis yang tidak hentinya mendo'akan serta memberikan dorongan semangat dan materi;
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Bapak Dr.-ing. Ardian Ulvan, S.T., M.Sc. selaku Dosen Pembimbing Utama yang telah memberikan ilmu, bimbingan, bantuan, masukan, dan pandangan kehidupan kepada penulis disetiap kesempatan dengan baik dan ramah;
5. Bapak Aryanto, S.T., M.T. selaku Dosen Pembimbing Pendamping yang telah memberikan ilmu, bimbingan, masukan, motivasi, dan pandangan kehidupan kepada penulis disetiap kesempatan dengan baik dan ramah;
6. Bapak Ir. Meizano Ardhi Muhammad S.T.,M.T. selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun kepada penulis;
7. Ibu Umi Murdika, S.T, MT selaku Pembimbing Akademik yang telah memberikan arahan, nasehat dan bimbingan yang membangun bagi penulis selama menempuh perkuliahan;
8. Segenap dosen di Jurusan Teknik Elektro yang telah memberikan ilmu yang bermanfaat kepada penulis selama menempuh pendidikan dan perkuliahan;
9. Segenap staff di Jurusan Teknik Elektro yang telah membantu penulis baik dalam hal administrasi dan lain-lain;

10. Keluarga Besar Laboratorium Telekomunikasi yang memberikan banyak ilmu, masukan dan saran kepada penulis yang tidak bisa dibayarkan;
11. Keluarga Besar PPK Ormawa Desa Gisting Permai yang telah membantu banyak hal dalam menyelesaikan skripsi ini;
12. Keluarga Besar *Greenhouse*, Auliya Syahputra Siregar, M. Affan Siddiqie Asmara, M.Herly Pratama, dan Kiagus Muhammad Ade Iqbal Ramadhan yang telah bersama-sama memberikan semangat dan gagasan dalam menyelesaikan skripsi ini;
13. Tim Mahasiswa Fakultas Pertanian yang telah membantu memberikan rekomendasi dan pemeliharaan tanaman;
14. Dan sahabat saya Kurnia Aji Setiawan, Ahmad Nurachman, Yoga, Darul Ikhsan, Shopwan Subhatan, Mahatir Muhammad, Dzikri Afridho, Febrino, Rendi, dan Adi Pratama terima kasih telah memberikan semangat kepada penulis
15. Master Dwi Aji Cahyono yang selalu membantu dalam menyelesaikan semua *trouble* dalam pengerjaan skripsi ini;
16. Keluarga TELTI yang telah membantu, memberi saran, inspirasi, dan canda tawa kepada penulis dalam menyelesaikan skripsi ini;
17. Leting Teknik Elektro'20, terimakasih atas semangat dan kebersamaannya selama menempuh pendidikan di Jurusan Teknik Elektro;
18. Keluarga Besar HIMATRO yang telah memberikan ilmu dan pengalaman kepada penulis selama pendidikan baik secara langsung maupun tak langsung;
19. Seluruh pihak yang telah membantu penulis menyelesaikan skripsi ini.

Penulis mengakui adanya kekurangan dalam skripsi ini dan dengan tulus menerima kritik serta saran yang membangun dari berbagai pihak demi kemajuan bersama. Semoga skripsi ini bermanfaat bagi penulis dan pembaca.

Bandar Lampung, 4 November 2024

Penulis



Alpriealian Renando

DAFTAR ISI

Halaman

Halaman

DAFTAR GAMBAR

DAFTAR TABEL

Halaman

1. PENDAHULUAN

1.1 Latar Belakang

Greenhouse adalah bangunan yang dirancang untuk dimanipulasi agar kondisi lingkungan sesuai dengan kebutuhan optimal pertumbuhan tanaman. Dalam sektor agribisnis, peran *greenhouse* sangat lengkap karena memungkinkan peningkatan kualitas dan kuantitas hasil panen dengan mengontrol faktor lingkungan[1].. Namun, pengelolaan *greenhouse* menghadapi kendala besar, yaitu ketidakmampuan sistem kontrol konvensional untuk menyesuaikan kondisi iklim mikro *indoor*, yang secara dinamis dapat disesuaikan dengan kebutuhan komoditi tanaman yang sedang dibudidayakan di dalam *greenhouse* tersebut . Oleh karena itu, diperlukan sistem kontrol lingkungan *greenhouse* yang akurat untuk memenuhi semua kebutuhan yaitu kemampuan akuisi data secara *real time*, berkelanjutan, dan handal untuk memonitor dan mengatur kondisi lingkungan bagi pertumbuhan tanaman.

AGRI–*Intelligence Control System* (ICS) adalah sebuah sistem kontrol cerdas yang menerapkan prinsip *Internet-of-Thing* (IoT) dan kecerdasan buatan untuk mengatur keseluruhan fungsi sensor dan instrumen dalam sebuah *greenhouse* agar bekerja secara adaptif dan autonomous (*smart-greenhouse*). Melalui *IoT*, pemantauan dan pengendalian parameter lingkungan secara *real-time*, dapat dilakukan [3]. Selain itu, pengaturan distribusi air dan pupuk, serta pengaturan kondisi iklim *indoor* dalam *greenhouse* secara otomatis menjadi mungkin dengan teknologi ini. Teknologi ini juga memungkinkan integrasi dengan aplikasi *smartphone* dan *web*, sehingga pemantauan dapat dilakukan dari jarak jauh.

AGRI–ICS terdiri dari perangkat keras sub-sistem sensor, aktuator, dan sub-sistem kendali akuisisi data. Sedangkan perangkat lunak AGRI-ICS berfungsi sebagai sub-sistem operasi perangkat, dan pemrosesan data dan informasi. Sub-sistem pemrosesan data dan informasi AGR–ICS terdiri dari *backend office* dan

front-end office. *Backend office* berfungsi dalam penyimpanan (*storage*) data sensor hasil akuisisi kedalam sebuah database management system dan penyediaan sebuah antar muka pemograman aplikasi (*application programming interface* – API) agar data yang tersimpan dalam database dapat proses untuk analisis lebih lanjut atau disajikan dalam melalui platform *frontend office*. Penelitian ini fokus pada pengembangan *platform frontend* AGR–ICS dalam bentuk aplikasi web dengan menggunakan React.js.

Teknologi *React.js* dianggap dapat menyediakan solusi dalam pengembangan *platform* aplikasi *frontend* yang responsif dan mudah digunakan pada platform *smartphone* maupun *web* [2]. Platform aplikasi *frontend* ini memungkinkan pemantauan, pengukuran, dan deteksi dini kebutuhan lengkap di *greenhouse*, seperti debit air, kebutuhan pupuk, dan kondisi lingkungan lainnya, sehingga siapa pun bisa dengan mudah memastikan tanaman mendapatkan perawatan yang optimal. Dengan demikian, pengelolaan *greenhouse*, pemantauan perkembangan tanaman, serta optimalisasi penggunaan sumber daya dapat dilakukan dengan lebih efisien.

Namun demikian, optimalisasi kuantitas, kualitas, dan kontinuitas produksi masih merupakan tantangan yang perlu dicapai. Oleh karena itu, inovasi dalam Desain dan operasional *Greenhouse* diperlukan agar hasil produksi stabil meskipun perubahan iklim terjadi secara dinamis [4]. Dengan latar belakang tersebut, penelitian ini bertujuan untuk mengembangkan sistem AGRI–ICS *Smart Greenhouse* berbasis *IoT* dengan menggunakan *React.js*. Sistem ini diharapkan dapat berkontribusi signifikan dalam meningkatkan efisiensi dan produktivitas agribisnis, serta menjadi solusi praktis bagi petani dalam menghadapi tantangan perubahan iklim yang tidak terduga.

1.2 Tujuan Penelitian

Penelitian ini bertujuan untuk membangun *frontend* sistem *Agri-Intelligence Control System (ICS) Smart Greenhouse* berbasis *IoT* dengan menggunakan *library React.js*. Sistem ini dirancang untuk memantau serta menampilkan data perkembangan tanaman dan kondisi iklim dalam *greenhouse* secara *real-time*.

1.3 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini dapat disimpulkan sebagai berikut:

1. Bagaimana membangun *frontend* dari sistem *Agri-Intelligence Control System (ICS) Smart Greenhouse* berbasis *Internet of Things (IoT)* dengan menggunakan *React.js* untuk memantau serta menampilkan data perkembangan tanaman dan kondisi iklim dalam *Greenhouse* secara *real-time*?

1.4 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut.

1. *Frontend* yang dibangun merupakan bagian dari *capstone project AGRI–ICS Smart Greenhouse* berbasis *IoT*.
2. Data untuk pengembangan dan pengujian disediakan oleh sistem *backend AGRI–ICS Smart Greenhouse* berbasis *IoT*.
3. *Platform monitoring* yang dikembangkan dioptimalkan untuk penggunaan di perangkat desktop.

1.5 Manfaat Penelitian

Manfaat Penelitian ini adalah sebagai berikut:

1. Pengembangan *frontend* ini memungkinkan instansi resmi dan organisasi terkait untuk memantau dan menganalisis data suhu secara *real-time* dalam *Greenhouse* melalui antarmuka yang mudah diakses, yang diharapkan dapat meningkatkan efisiensi dalam pemantauan lingkungan tumbuh tanaman.
2. Dengan adanya *frontend* ini, para petani dapat memantau kondisi *Greenhouse* secara *online*, tanpa harus berada di lokasi secara langsung. Hal ini diharapkan dapat meningkatkan kemudahan akses data dan mendukung pengambilan keputusan yang lebih cepat terkait pengelolaan lingkungan *Greenhouse*.
3. *Frontend* yang dikembangkan juga mendukung penggunaan data secara lebih efisien, sehingga memungkinkan penggunaan sumber daya secara tepat dan berkelanjutan. Visualisasi data yang interaktif diharapkan dapat memudahkan pengguna dalam memahami tren dan kebutuhan tanaman, yang berkontribusi pada pertanian yang lebih ramah lingkungan.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada skripsi ini adalah pembagian menjadi 5 bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini menguraikan secara umum mengenai latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika penulisan mengenai Membangun *Frontend* Sistem *Agri-Intelligence Control System (ICS) Smart Greenhouse* Berbasis *IoT* Menggunakan *Library React.js.*

BAB II TINJAUAN PUSTAKA

Pada Bab ini membahas mengenai penelitian terkait dengan protokol *MQTT* pada sistem iot dan REST API , kondisi saat ini pada alat sensor di salah satu tempat, *Smart Greenhouse (AGRI-ICS), ReactJS, ChartJS,* REST API , *MQTT.JS, Kanban Method, Axios, TailwindCSS, Visual Studio Code, Trello, Webhoster, Blackbox-testing.*

BAB III METODE PENELITIAN

Menjelaskan waktu, tempat, alat dan bahan, dan metode *Scrum* yang terdiri dari *product backlog Sprint Planning, Sprint Execution,Sprint Review Sprint Retrospective* dan penelitian beserta tahapannya mengenai Pengembangan *Dashboard* IoT Untuk Visualisasi Data *Real-time* dari *Smart Greenhouse AGRI-ICS*

BAB IV HASIL DAN PEMBAHASAN

Menjelaskan bagaimana proses pelaksanaan penelitian dan juga hasil yang didapatkan pada penelitian mengenai Implementasi Teknologi Pern Stack Dalam Pengembangan Dashboard IoT Untuk Visualisasi Data *Real-time* dari *smart Greenhouse agri-ics*.

BAB V KESIMPULAN DAN SARAN

Memaparkan kesimpulan apa saja yang didapat dari proses engembangan *Dashboard* IoT Untuk Visualisasi Data *Real-time* dari *AGRI-ICS* dan memberikan saran untuk dilakukan pada penelitian selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Seful Sidik dan Wasis Haryono berjudul "Simulasi Sistem Monitoring dan Pengendalian Suhu dan Kelembaban Ruangan Lab Kalibrasi Berbasis IoT (*Internet of Things*)" (2024)[5] fokus pada pengembangan dan penerapan sistem monitoring serta pengendalian otomatis berbasis IoT untuk mengelola suhu dan kelembaban di ruang laboratorium kalibrasi PT. Quantum Inti Akurasi. Penelitian ini dilatarbelakangi oleh kebutuhan kritis untuk menjaga stabilitas lingkungan di dalam ruang kalibrasi, yang dipengaruhi oleh faktor-faktor internal seperti aktivitas kalibrasi dan jumlah orang di dalam ruangan, serta faktor eksternal seperti kondisi cuaca dan suhu lingkungan sekitar. Sistem ini dirancang menggunakan teknologi IoT yang melibatkan integrasi perangkat keras seperti Arduino, sensor DHT11 dan DHT22 untuk pemantauan suhu dan kelembaban, serta perangkat lunak berbasis *Framework* CodeIgniter yang terhubung dengan basis data MySQL untuk pengelolaan data secara efisien.

Kesimpulan dari penelitian yang dilakukan oleh Seful Sidik dan Wasis Haryono adalah mereka berhasil mengembangkan sebuah sistem monitoring dan pengendalian berbasis IoT yang dapat memantau dan mengendalikan suhu serta kelembaban dengan akurat di ruang laboratorium kalibrasi PT. Quantum Inti Akurasi. Sistem ini memberikan kemudahan bagi pengguna dalam memantau kondisi ruangan secara jarak jauh melalui antarmuka web yang intuitif, serta memungkinkan pengaturan otomatis dan manual terhadap perangkat pengatur suhu dan kelembaban. Implementasi ini tidak hanya memastikan ruang laboratorium tetap dalam kondisi yang ideal, tetapi juga membantu menjaga efisiensi dalam pengoperasian laboratorium, meskipun terjadi perubahan kondisi cuaca yang signifikan. Selain itu, pengembangan sistem ini memperlihatkan potensi besar dari penerapan teknologi IoT dalam mendukung efisiensi dan akurasi dalam

pengelolaan lingkungan laboratorium, yang pada akhirnya meningkatkan kualitas proses kalibrasi yang dilakukan [5] .

Dalam penelitian lainnya, Mahyal Hadi melakukan rancang bangun sistem monitoring *smart home menggunakan energi cadangan berbasis Internet of Things (IoT)* [6]. Fokus penelitiannya pada pengembangan sebuah sistem rumah pintar yang memanfaatkan teknologi *Internet of Things* (IoT) untuk memonitor dan mengendalikan berbagai peralatan elektronik di rumah secara otomatis. Penelitian ini bertujuan untuk memberikan solusi praktis bagi para pemilik rumah yang sering menghadapi masalah efisiensi penggunaan energi serta keamanan, terutama ketika berada di luar rumah. Dengan memanfaatkan Raspberry Pi sebagai pusat pengendali, sistem ini memungkinkan pengendalian perangkat-perangkat seperti lampu, kipas angin, kunci pintu, dan pintu garasi dari jarak jauh, dengan menggunakan aplikasi web yang terhubung ke internet. Pengendalian ini dapat diakses melalui berbagai perangkat, seperti laptop atau smartphone, yang memberikan fleksibilitas dan kemudahan bagi pengguna.

Penelitian ini melakukan berbagai pengujian untuk memastikan efektivitas sistem. Uji coba dilakukan terhadap lima buah lampu, satu motor stepper untuk pintu garasi, satu motor servo untuk kunci pintu, dan satu motor brushless untuk kipas angin. Setiap perangkat berhasil dioperasikan dengan baik melalui akses jarak jauh menggunakan jaringan internet. Hasil pengujian menunjukkan bahwa sistem berfungsi dengan lancar, baik ketika diakses dari jarak dekat maupun dari jarak jauh, seperti di kota yang berbeda. Misalnya, kendali perangkat berhasil dilakukan dari Demak hingga Surabaya, membuktikan bahwa sistem ini tidak saja handal, tetapi juga stabil saat diakses melalui internet dari jarak jauh.

Kesimpulan dari penelitian ini menunjukkan bahwa prototipe Smart Home berbasis IoT yang dikembangkan berhasil memberikan solusi nyata untuk pemantauan dan pengendalian jarak jauh berbagai peralatan rumah tangga. Sistem ini memungkinkan penghematan energi yang signifikan karena pengguna dapat memantau dan mengendalikan perangkat elektronik yang lupa dimatikan, seperti lampu, dari mana saja. Selain itu, sistem ini meningkatkan aspek keamanan rumah dengan kemampuan untuk mengunci pintu dan mengendalikan garasi dari jarak jauh. Dengan antarmuka web yang dapat diakses dengan mudah, pengguna

diberdayakan untuk memantau kondisi rumah mereka secara *real-time*, menawarkan kenyamanan dan kemudahan dalam kehidupan sehari-hari [6].

Selanjutnya, dalam penelitian yang dilakukan oleh Asti Herliana dan Prima Muhamad Rasyid berjudul "*Sistem Informasi Monitoring Pengembangan Software Pada Tahap Development Berbasis Web*" [7], para penulis melakukan pengembangan sebuah sistem yang memungkinkan pemantauan dan pengelolaan proyek pengembangan perangkat lunak secara *real-time*. Penelitian ini bertujuan untuk memberikan solusi praktis dalam memantau perkembangan proyek perangkat lunak, terutama pada tahap development, yang sering kali menghadapi masalah dalam hal pemberian tugas yang kurang efektif serta penyimpanan data proyek yang tidak terstruktur dengan baik.

Sistem ini dikembangkan menggunakan bahasa pemrograman PHP dan framework CodeIgniter, dengan basis data MySQL. Fungsionalitas utama dari sistem ini adalah untuk mendokumentasikan informasi proyek, seperti pembagian tugas, spesifikasi tugas, kendala yang dihadapi, dan estimasi waktu penyelesaian tugas. Informasi ini kemudian dapat digunakan oleh sistem analis dan programmer untuk memantau kemajuan proyek secara tepat waktu, sehingga meminimalisasi risiko keterlambatan dan meningkatkan efisiensi kinerja.

Kesimpulan dari penelitian ini menunjukkan bahwa sistem informasi monitoring pengembangan perangkat lunak berbasis web yang dikembangkan berhasil mempermudah proses dokumentasi dan pemantauan proyek secara *real-time*. Sistem ini membantu sistem analis dalam pengambilan keputusan terkait kemajuan proyek, serta memudahkan programmer dalam mengelola tugas-tugas yang harus diselesaikan. Selain itu, sistem ini juga memungkinkan penyimpanan informasi terkait perbaikan modul yang dapat diakses oleh programmer. Untuk pengembangan lebih lanjut, penulis menyarankan penambahan fitur *live chat* dan grafik kinerja programmer untuk meningkatkan efektivitas sistem [7].

Berbeda dengan penelitian-penelitian yang dibahas sebelumnya, penelitian skripsi ini lebih fokus pada pengembangan antarmuka *frontend* dari sistem *Agri-Intelligence Control System* (ICS) *Smart Greenhouse* berbasis *Internet of Things* (IoT) menggunakan *library* React.js. Sistem ini dirancang untuk memantau kondisi lingkungan di dalam *greenhouse* secara *real-time*, mencakup parameter seperti

suhu, kelembaban, dan pemantauan pasokan nutrisi tanaman, serta memungkinkan pengelolaan data melalui API. Data yang dikumpulkan divisualisasikan dalam bentuk grafik dan tabel pada *dashboard* berbasis web yang responsif dan interaktif, dengan tambahan fitur pengunduhan data untuk analisis lebih lanjut. Perbedaan signifikan dari tiga penelitian yang menjadi kajian pustaka tersebut, bukan hanya terletak pada ruang lingkup, tujuan penelitian, tapi lebih utama dalam hal mekanisme dan konsep pengembangan dan pemogramannya. Para penulis tersebut lebih melakukan pengembangan sistem dan aplikasi yang fokus pada pemantauan dan pengelolaan untuk meningkatkan efisiensi, dokumentasi dan pembagian tugas. Sebaliknya, penelitian kami ini mencakup pengelolaan lingkungan *greenhouse* yang lebih kompleks, dengan tujuan mendukung efisiensi agrikultur melalui pemantauan suhu, kelembaban, pasokan nutrisi, dan irigasi tanaman. Penelitian ini menggunakan React.js untuk pengembangan *frontend* dengan fokus pada visualisasi data *real-time* dari sistem IoT, sedangkan penelitian-penelitian sebelumnya menggunakan PHP dan CodeIgniter untuk pengembangan sistem berbasis web yang lebih konvensional. Selain itu, penelitian ini menggunakan protokol MQTT untuk data *real-time*, sementara penelitian-penelitian sebelumnya menggunakan sistem berbasis HTTP sederhana. Dengan pendekatan yang berbeda, penelitian ini menghadirkan solusi praktis dalam bidang agrikultur untuk menjawab tantangan perubahan iklim yang dinamis.

2.2 Kondisi Teknologi *Agri-Intelligence* Control System (ICS)

Penelitian ini dilaksanakan di Desa Gisting Permai, Kec. Gisting, Kab. Tanggamus, Prov. Lampung. Penelitian dilakukan sebagai respon dan solusi atas keresahan warga petani di desa tersebut yang terus mengalami penurunan produktifitas pada sebagian besar tanaman hortikultura yang ditanam. Dari kajian awal yang dilakukan, dapat diidentifikasi beberapa faktor yang menyebabkan penurunan produksi tersebut, yaitu:

- Terjadinya penurunan kesuburan dan saya dukung tanah yang disebabkan oleh penggunaan pupuk kimia yang tidak terkendali;

- Berkurangnya luas area pertanian akibat proses pembangunan. Lahan pertanian banyak dikonversi menjadi pemukiman, bangunan dan jalan, serta sentra-sentra usaha non-pertanian lainnya.
- Perubahan iklim, dimana sering terjadinya gagal panen karena **petani** gagal atau tidak berhasil mengantisipasi perubahan iklim yang terjadi.

Atas permasalahan tersebut, tim kami beserta dosen pembimbing berupaya melakukan inovasi di Desa tersebut dengan cara mengembangkan sistem pertanian dalam *greenhouse*. Pengelolaan *greenhouse* ini dilakukan secara cerdas (*smart greenhouse*) dengan sistem pengaturan iklim dan irigasi indoor dilakukan secara autonomous oleh sebuah perangkat cerdas *intelligence control system* (ICS). Sistem ini kami namakan AGRI-ICS.

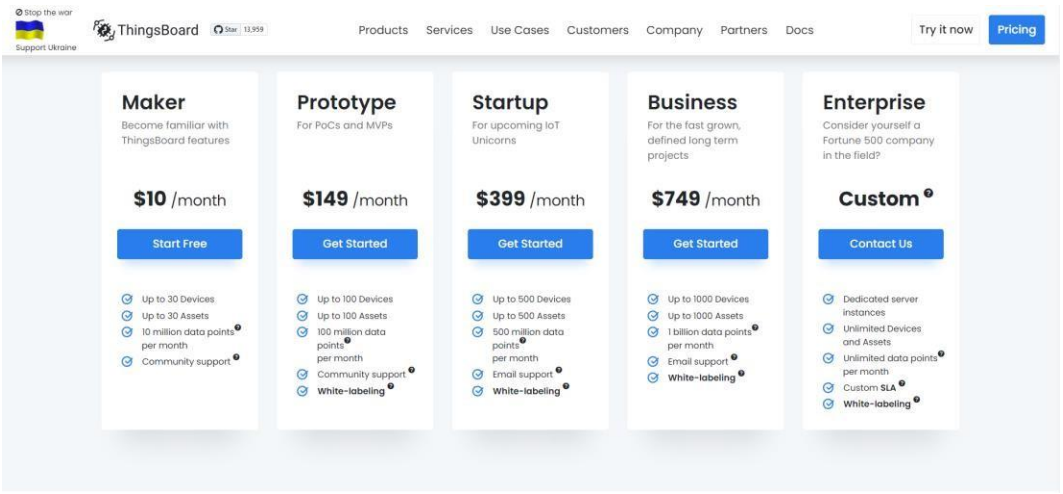
Versi awal sistem AGRI-ICS sudah bekerja secara autonomous dengan mengirimkan data informasi cuaca di dalam *greenhouse* tersebut ke aplikasi dashboard pihak ketiga yaitu *ThingsBoard.* Protokol yang digunakan untuk mengirimkan data tersebut masih menggunakan *Hypertext Transfer Protocol (HTTP).* Data yang diterima oleh sensor akan berlangsung dikirimkan ke server milik *ThingsBoard* menggunakan *HTTP* dan data tersebut akan disimpan ke *database* milik *ThingsBoard.* Gambar 2.1 memperlihatkan *greenhouse* yang dibangun di desa Gisting Permai.

Gambar 2.1 Perangkat Monitoring Sistem AGRI-ICS *Greenhouse* di desa Gisting permai

Pada gambar 2.1 Perangkat Monitoring Sistem AGRI-ICS *greenhouse* di Desa Gisting Permai. Sistem yang berjalan saat ini menggunakan aplikasi pihak ketiga dalam menyimpan dan menampilkan data. Namun ini semua mempunyai

beberapa kelemahan yang menjadi kendala untuk keberlangsungan dari informasi yang diterima oleh umum. Pertama, melakukan penyimpanan dan menampilkan data di dashboard membutuhkan akun, dimaka akun tersebut akan dikenakan biaya langganan dalam penggunaannya. Selain itu, dalam setiap akun dibatasi dengan jumlah data points / *traffic* yang dikirimkan setiap bulannya. Kelemahan selanjutnya adalah keamanan data yang dikirimkan tidak dapat bersifat milik pribadi dikarenakan data tersebut tersimpan di database milik ThingsBoard dan akan dihapus apabila telah melampaui traffic data yang telah ditentukan. Selain itu kelemahan terakhir adalah dalam desain untuk *website* yang akan ditampilkan harus mengikuti desain yang disediakan oleh ThingsBoard yang terbatas.

Penelitian ini dilakukan untuk mengatasi kelemahan yang terjadi ketika menggunakan pihak ketiga sebagai media pemantauan *greenhouse* yang terjadi saat ini berjalan masih menggunakan protokol pengiriman dan penyimpanan data dari aplikasi ketiga. Selain itu, penggunaan protokol HTTP pada lapisan aplikasi, diganti dengan protokol MQTT, yang lebih sesuai dengan prosedur komunikasi *machine-to-machine* pada IoT. Protokol HTTP masih tetap digunakan untuk data konvensional dan yang non-realtime.



Gambar 2.2 Biaya langganan dan batasan akun pada ThingsBoard

Pada gambar 2.2 dapat dilihat biaya langganan dan batasan akun pada ThingsBoard. Pada penelitian ini fokus pengembangan dilakukan pada aplikasi *dashboard frontend* yang menggunakan data yang disimpan pada *backend database*. Data yang divisualisasikan berupa data *real-time* dan data bukan *real-*

time melalui REST -API yang telah disediakan oleh pengembang *backend*, untuk selanjutnya ditampilkan di dashboard Agri-ICS. Website ini akan menggantikan peran dashboard dari paltform IoT *ThingsBoard* yang berbayar namun tidak dapat dikustomisasi. . Pengembangan *frontend dashboard* Agri-ICS menggunakan *library Javascript* ReactJS dapat mewujudkan tujuan dari penelitian ini yaitu untuk menyediakan *platform* IoT yang mudah diakses masyarakat umum dan peneliti untuk mengontrol parameter iklim, irigasi, dan pasokan nutrisi pada *greenhouse.*

2.3 Peran *Website* Dalam Sistem Monitoring *Smart Greenhouse*

Website adalah sebuah media informasi yang ada di internet. *Website* tidak saja dapat digunakan untuk penyebaran informasi, namun juga bisa digunakan untuk membuat toko online (*e-commerce*). [8] *Website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web (WWW)* di Internet. Sebuah halaman *web* adalah dokumen yang ditulis dalam format *HTML (Hyper Text Markup Language)*, yang hampir selalu bisa diakses melalui protokol *HTTP*, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

2.4 Agri-ICS *Smart Greenhouse*

Agri-ICS *Smart Greenhouse* merupakan sistem yang dirancang untuk memantau kondisi lingkungan di dalam rumah kaca (*greenhouse*). Tujuan utama dari Agri-ICS adalah memantau pertumbuhan tanaman secara detail, serta memastikan pemenuhan kebutuhan nutrisi dan air harian tanaman. Sistem ini juga bertujuan untuk mengontrol kondisi ruangan di *greenhouse* agar tanaman dapat memperoleh nutrisi yang sesuai dengan kebutuhan.

Smart Greenhouse adalah konsep rumah kaca yang dilengkapi dengan teknologi canggih untuk otomatisasi dan optimalisasi lingkungan tumbuh tanaman di dalamnya. Agri-ICS *Smart Greenhouse* merupakan rancangan di mana tanaman yang berada dalam ruangan *greenhouse* dipantau secara optimal agar nutrisi dan

pertumbuhan tanaman lebih terjaga, sehingga ketika tanaman berbuah, hasilnya dapat lebih memuaskan [9].

2.5 *Internet of Things (IoT)*

Internet of Things (IoT) adalah jaringan perangkat fisik kendaraan, bangunan, dan barang lainnya tertanam dengan elektronik, perangkat lunak, sensor, dan konektivitas jaringan yang memungkinkan benda-benda ini mengumpulkan dan bertukar data. Pengertian lain yang menjelaskan bahwa sistem perangkat komputasi yang saling terkait, mesin mekanik dan digital, objek, hewan, atau manusia yang dilengkapi dengan pengidentifikasi unik dan kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer [10].

Tujuan utama dari IoT adalah untuk menciptakan lingkungan yang terhubung secara digital, di mana objek-objek sehari-hari dapat saling berinteraksi dan berbagi informasi secara otomatis tanpa adanya campur tangan manusia. Melalui penggunaan sensor dan perangkat yang terhubung, data yang dikumpulkan dapat digunakan untuk memantau, mengontrol, dan mengelola sistem secara efisien.

Beberapa contoh aplikasi IoT meliputi:

1. *Smart Home*

   Memungkinkan pengendalian perangkat-perangkat di rumah seperti lampu, suhu, keamanan, dan perangkat elektronik lainnya melalui jaringan internet.

2. *Smart City*

   Menggunakan sensor dan koneksi internet untuk mengelola dan memantau infrastruktur kota seperti pencahayaan, parkir, pengelolaan limbah, dan lalu lintas.

3. *Smart Healthcare*

   Memungkinkan pemantauan kesehatan pasien secara *real-time* melalui perangkat medis yang terhubung, serta pengelolaan inventaris obat dan perangkat medis.

2.6 Pemanfaatan *ReactJS* dalam Pengembangan Antarmuka untuk *Smart Greenhouse*

ReactJS adalah salah satu kerangka kerja web populer yang lebih baik daripada kerangka kerja lain seperti Angular, Vue, dll. Ini karena penerapan Virtual DOM, yang tujuan utamanya adalah untuk meningkatkan kinerja aplikasi secara keseluruhan [11].

ReactJS adalah kerangka kerja web yang terutama dirancang untuk mengatasi masalah kinerja dalam aplikasi web. React menggunakan DOM virtual yang memutuskan apakah komponen harus dimuat ulang atau tidak berdasarkan status komponen saat ini dan perubahan yang telah terjadi. Hal ini mencegah aplikasi memuat ulang yang tidak perlu. Selain itu, React juga memperkenalkan aliran data satu arah yang membantu mengontrol aliran data dalam aplikasi yang membuat pelacakan kejadian menjadi lebih mudah dan juga menyederhanakan propagasi dan stabilitas .

ReactJS adalah pustaka (*library*) JavaScript yang populer untuk membangun antarmuka pengguna (*user interface - UI*) interaktif. Dikembangkan oleh *Facebook*, *ReactJS* digunakan untuk membuat komponen *UI* yang dapat digunakan ulang dan dapat dikelola dengan mudah.

Beberapa fitur utama dari ReactJS meliputi:

1. Komponen

   ReactJS memandang UI sebagai kumpulan komponen yang dapat digabungkan bersama untuk membentuk tampilan yang lengkap. Setiap komponen berfungsi sebagai unit terpisah yang dapat memiliki logika, properti (properties), dan keadaan (*state*) mereka sendiri [12].

2. *Virtual DOM*

   ReactJS menggunakan *Virtual DOM (Document Object Model)* untuk mempercepat manipulasi *UI*. *Virtual DOM* adalah representasi virtual dari struktur *DOM* aktual, dan *ReactJS* mengoptimalkan perubahan pada Virtual *DOM* untuk meminimalkan pengubahan langsung pada *DOM* aktual, yang dapat memakan waktu dan mempengaruhi kinerja [13].

3. Reaktivitas

   ReactJS memperhatikan perubahan keadaan (*state*) dan properti (props)

komponen. Ketika ada perubahan dalam keadaan atau properti, *ReactJS* secara otomatis memperbarui tampilan yang terkait dengan efisien melalui rekonsiliasi *Virtual DOM*.

4. *JSX*

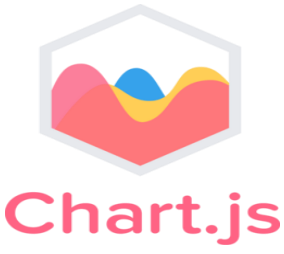
   ReactJS memperkenalkan *JSX (JavaScript XML)* yang memungkinkan penulisan kode *JavaScript* dan *HTML* secara bersamaan. *JSX* mempermudah penulisan dan pemahaman komponen *ReactJS* dengan sintaks yang dekat dengan *HTML* [14].

5. Ekosistem yang Kuat:

   ReactJS memiliki ekosistem yang kaya dengan banyak pustaka dan alat pendukung seperti *React Router* untuk *routing*, *Redux* untuk manajemen keadaan, dan banyak lagi. Ini memudahkan pengembangan aplikasi yang kompleks dengan memanfaatkan komunitas yang aktif.

SPA (Single Page Application) merupakan aplikasi yang bekerja di dalam browser yang tidak membutuhkan reload page saat digunakan [15]. Dengan kata lain, pengguna atau user tidak akan berpindah halaman dengan melakukan request kepada server setiap kali terjadi interaksi pada aplikasi. Yang membedakan SPA dengan non-SPA adalah *single page application* akan melakukan *load* terhadap satu halaman dari server kemudian mekanisme *routing* yang biasanya di*handle* oleh server kini dibebankan pada *client*. Akibatnya, website yang menggunakan SPA memiliki performa yang lebih cepat tanpa harus merequest kembali halaman secara terus menerus.

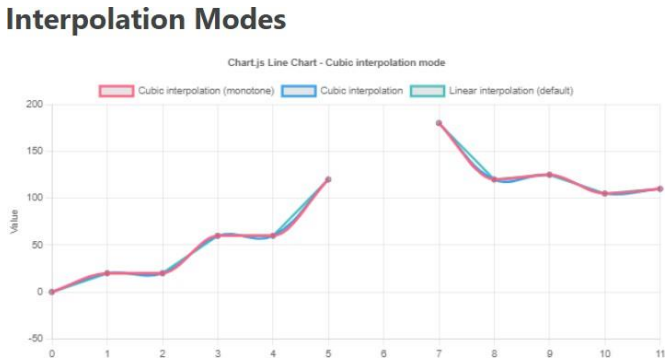
2.7 *Chart.JS*



Gambar 2.3 chart.js [16]

Pada gambar 2.3 *chart.js* adalah salah satu *library* yang memanfaatkan elemen canvas untuk membuat grafik yang akan ditampilkan melalui *web browser*. *Library*

ChartJS dapat dimanfaatkan untuk menampilkan informasi laporan dalam bentuk grafik sehingga lebih mudah dipahami oleh orang lain dari pada menampilkan data dalam bentuk tabel [17].



Gambar 2.4 Tampilan grafik chart.js [18]

Pada gambar 2.4 tampilan grafik *chart js* merupajan *library* grafik berbasis *Javascript* yang gratis dan *open source* untuk visualisasi data yang mendukung delapan jenis tipe grafik, yaitu: *Bar*, *line*, *area*, *pie*, *bubble*, *radar* dan *scatter*. *Chart.js* dibuat oleh pengembang web berbasis London, Nick Downia dan sekarang telah dikembangkan lebih lanjut oleh komunitas Github.

2.8 RESTful-API

Aplikasi dengan arsitektur *client-server* dapat dibangun dengan sistem operasi yang berbeda pada sisi *client* dan sisi s*erver*. REST API digunakan sebagai perantara dua sistem operasi untuk saling berhubungan. REST API adalah kumpulan fungsi, aturan, perintah, dan protokol yang berisi standar umum tentang cara bertukar informasi antara klien dan server.

REST API adalah salah satu jenis dari API yang didasarkan pada arsitektur REST *(Representational State Transfer)*. REST sendiri adalah sebuah arsitektur perangkat lunak yang terdiri dari aturan dan konvensi yang digunakan untuk membuat web service. Dalam REST API, setiap sumber daya (*resource*) diidentifikasi dengan URI *(Uniform Resource Identifier)* dan diakses melalui metode HTTP seperti GET, POST, PUT, DELETE, dan sebagainya.

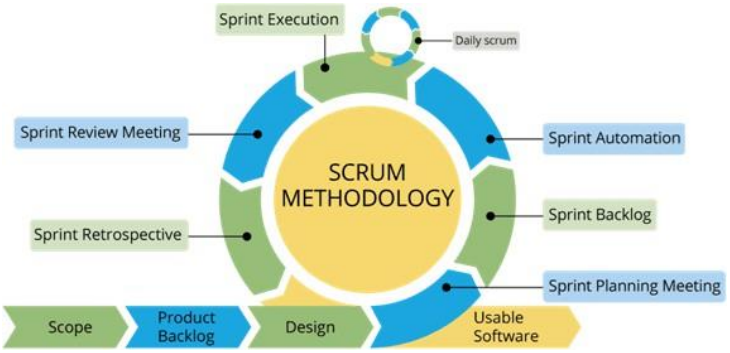
2.9 MQTT.js

MQTT.js adalah pustaka JavaScript yang mengimplementasikan protokol MQTT *(Message Queuing Telemetry Transport)* untuk komunikasi berbasis pesan antar perangkat. MQTT.js memungkinkan aplikasi atau perangkat yang ditulis dalam bahasa JavaScript untuk berkomunikasi dengan *broker* MQTT [19].

MQTT.js memungkinkan aplikasi JavaScript untuk berkomunikasi dengan perangkat lain melalui protokol MQTT yang efisien dan ringan. Hal ini berguna untuk berbagai skenario seperti *Internet of Things* (IoT), sensor monitoring, komunikasi mesin ke mesin (M2M), dan banyak lagi [20].

2.10 Scrum

Scrum adalah kerangka kerja manajemen proyek yang berbasis pada prinsip-prinsip *agile*, dirancang untuk mengelola proyek-proyek kompleks dan terutama dikenal dalam pengembangan perangkat lunak. *Scrum* menawarkan pendekatan iteratif dan inkremental untuk pengembangan produk, yang memungkinkan tim untuk merespons perubahan kebutuhan pelanggan dan kondisi pasar dengan cepat [21].



Gambar 2.5 Metode Scrum [22]

Berikut adala beberapa prinsip dasar dari metode *Scrum*:

Tim Kerja: Tim Scrum terdiri dari individu-individu yang memiliki keterampilan dan peran yang berbeda dalam proyek, termasuk *product owner* (pemilik produk), Scrum *Master*, dan Tim Pengembang. Sprint: Siklus pengembangan dalam Scrum disebut sebagai "Sprint". Setiap Sprint biasanya memiliki durasi yang tetap, misalnya dua minggu, dan dimulai dengan pertemuan

perencanaan Sprint di mana tujuan dan ruang lingkup Sprint ditetapkan. Product *Backlog*: Ini adalah daftar prioritas dari semua fitur, perbaikan, dan pekerjaan lain yang perlu dilakukan pada produk. *Product owner* bertanggung jawab atas Product *Backlog* dan memastikan item-itemnya terus diperbarui sesuai kebutuhan. Sprint *Backlog*: Ini adalah daftar tugas-tugas yang harus diselesaikan oleh tim pengembang selama Sprint tertentu. Sprint *Backlog* dihasilkan dalam pertemuan perencanaan Sprint dan dapat disesuaikan selama Sprint berlangsung. *Sprint Review*: Setelah selesai Sprint, tim melakukan pertemuan ulasan di mana mereka meninjau hasil pekerjaan mereka dengan pemilik produk dan pihak-pihak terkait lainnya. Ini adalah kesempatan untuk mendemonstrasikan fungsionalitas baru kepada pemangku kelengkapan dan mendapatkan umpan balik. Retrospektif Sprint: Setelah *Sprint Review*, tim melakukan pertemuan retrospektif di mana mereka mengevaluasi proses pengembangan mereka sendiri. Mereka mengidentifikasi apa yang berhasil, apa yang tidak berhasil, dan cara untuk meningkatkan kinerja mereka di Sprint berikutnya.

2.11 Axios

Axios adalah salah satu sintaks yang sangat sederhana di reactJS untuk berinteraksi dengan API dan secara otomatis mengubah data JSON. Beranjak dari keunggulan dari axios, Axios adalah salah satu sintaks yang sangat sederhana di ReactJS untuk berinteraksi dengan API dan secara otomatis mengubah data JSON. *Library* Axios dapat digunakan untuk melakukan permintaan dengan metode axios get melalui URLs dengan kinerja yang baik dari Fetch API [22].

Library Axios bisa mengatasi berbagai masalah yang akan terjadi pada saat pengiriman data tanpa memerlukan perintah khusus untuk mengatasi pesan error tersebut. Mengingat banyak kelebihan yang ditawarkan ReactJS, maka penelitian ini merasa perlu melakukan pendalaman terhadap cara kerja React.js terutama penggunaan Axios dalam pengambilan dan menampilkan data melalui REST API [23].

2.12 Tailwind CSS

Tailwind adalah sekumpulan kelas *utilitas* dan *preprocessor cascading style sheet* (*CSS*) yang menghasilkan kelas-kelas CSS tersebut dan memungkinkan penggunaan alat CSS tambahan seperti @apply pengarahan. Kita perlu menginstal framework itu sendiri dan juga menambalnya ke dalam rantai alat pemrosesan CSS [24].

Visual Studio Code merupakan sebuah text editor yang dibuat oleh Microsoft. Visual Studio code ini mendukung banyak bahasa pemrograman seperti bahasa javascript, c, c#, c++, python, dan sebagainya. Dengan *plugin* yang di akses di dalam *Marketplace Visual Studio Code*.

2.13 Visual Studio Code

Visual Studio Code (VS Code) versi 1.80.1 yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk *debugging*, *GIT Control* yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, *shortcut keyboard*, dan preferensi. Visual Studio Code gratis dan *open-source*, meskipun unduhan resmi berada di bawah lisensi proprietary. Visual Studio Code memungkinkan pengguna untuk menulis kode pada berbagai bahasa pemrograman, termasuk H*TML, CSS, JavaScript, PHP*, *Python*, dan lain-lain. Visual Studio Code berjalan pada sistem operasi Windows, Mac, dan Linux. Editor teks ini dilengkapi dengan fitur-fitur seperti fitur *debugging, Git integration, autocomplete, snippet, syntax highlighting,* dan lain-lain [25].

Kode Visual Studio didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi *Node.js* untuk desktop yang berjalan pada Blinklayout. Meskipun Atom dan Visual Studio Code menggunakan komponen editor yang sama (diberi kode nama *Monaco*) yang juga digunakan dalam Visual Studio Team Services, sebelumnya disebut *Visual Studio Online*, keduanya memiliki perbedaan signifikan. Atom lebih berfokus pada editor yang dapat disesuaikan dan modular, namun cenderung lebih berat dalam penggunaan sumber daya. Sebaliknya, Visual Studio Code dirancang untuk lebih ringan dan efisien,

dengan integrasi fitur seperti debugging, *Git*, dan *IntelliSense*, serta dukungan untuk berbagai ekstensi yang mempercepat pengembangan perangkat lunak.

2.14 Trello

Trello digunakan untuk manajemen proyek dengan metode Scrum. Alat ini membantu dalam pengelolaan tugas secara visual menggunakan papan Kanban (Kanban *board*). Trello memudahkan tim dalam mengatur pekerjaan dan menetapkan prioritas dengan menggunakan label, kartu tugas, dan pengingat. Versi online Trello digunakan dalam penelitian ini untuk kolaborasi tim secara *real-time* [26].

Trello memiliki beberapa kelebihan yang membedakannya dari aplikasi manajemen proyek dan kolaborasi lainnya yaitu:

1. Antarmuka pengguna yang intuitif:

   Trello memiliki antarmuka pengguna yang sangat mudah digunakan dan intuitif. Pengguna dapat membuat dan mengatur papan tugas, daftar, dan kartu dengan mudah, bahkan bagi mereka yang tidak memiliki pengalaman dengan aplikasi manajemen proyek sebelumnya.
2. Fleksibilitas dan kustomisasi:

   Trello memberikan fleksibilitas dan kustomisasi yang tinggi untuk pengguna. Pengguna dapat menyesuaikan papan tugas dan kartu sesuai dengan kebutuhan mereka, menambahkan label, *member*, daftar, dan catatan pada kartu, serta mengatur *deadline* dan pengingat untuk tugas tertentu.
3. Integrasi dengan aplikasi lain:

   Trello memiliki integrasi dengan banyak aplikasi lain seperti Google Drive, Slack, Dropbox, dan Evernote. Integrasi ini memungkinkan pengguna untuk menghubungkan tugas dan informasi dari berbagai platform dalam satu tempat yang mudah diakses.
4. Mudah diakses dari berbagai perangkat:

   Trello dapat diakses dari berbagai perangkat, termasuk desktop, tablet, dan smartphone. Aplikasi Trello mobile yang tersedia untuk Android dan iOS juga memungkinkan pengguna untuk mengakses papan tugas mereka dari mana saja dan kapan saja.

5. *Trello Business Class*

   Versi berbayar dari *Trello* yang dirancang khusus untuk tim dan organisasi. Versi ini memiliki fitur tambahan seperti pengendalian akses, integrasi dengan aplikasi lain, dan dukungan pelanggan 24/7.

Kelebihan-kelebihan tersebut membuat Trello menjadi pilihan yang tepat bagi tim atau organisasi yang ingin meningkatkan produktivitas dan kolaborasi dalam pekerjaan. Trello sangat mudah digunakan, fleksibel, dan dapat diakses dari berbagai perangkat, serta memiliki berbagai fitur yang dapat membantu pengguna mengelola proyek dan tugas dengan lebih efektif.

2.15 Webhoster

Hosting merupakan sebuah *space* atau tempat penyimpanan data sebuah *blog* atau *website* di internet yang memungkinkan data tersebut diakses secara online dan dilihat oleh semua orang dari seluruh penjuru dunia [27].

Webhoster menyediakan sebuah layanan *hosting* yang *unlimited* dimana pengguna bisa dengan bebas meng*upload* konten website baik itu berupa gambar, video maupun jenis file lain tanpa harus takut kehabisan ruang atau *space* tersebut. Dengan memanfaatkan fitur *Cpanel Hosting* tampilan website bisa di upload dan diakses oleh banyak orang didunia. Dalam *cpanel* bisa membuat folder baru dengan nama website dan domain yang akan digunakan. Dan dengan mengupload build dari project yang sudah dibuat maka *website* akan otomatis akan dipublikasi oleh *webhoster*[28].

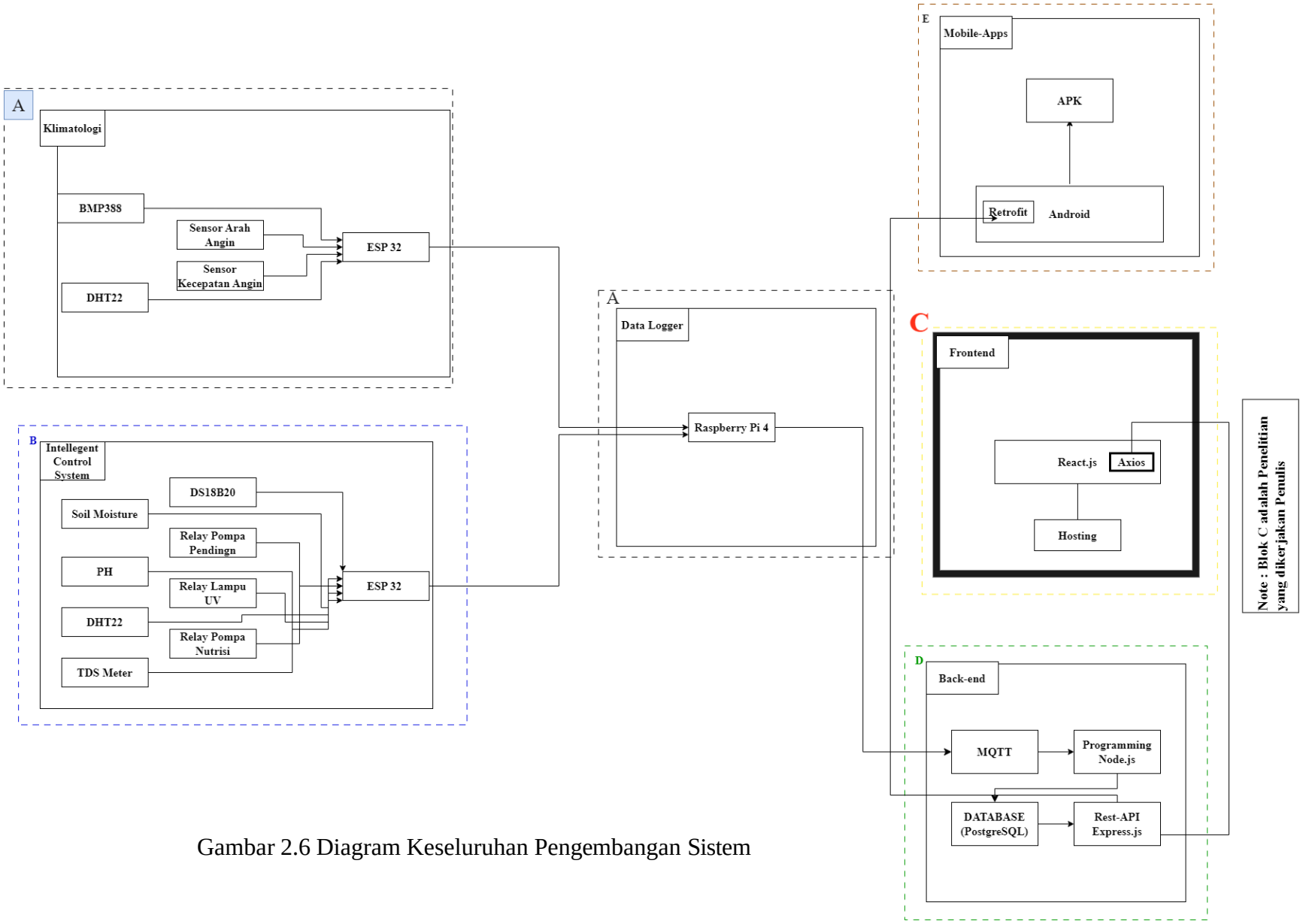
Web hosting adalah layanan jasa atau penyewaan tempat untuk menyimpan file atau bentuk *script* yang berada Internet dan memungkinkan untuk perorangan atau pun organisasi guna menampilkan layanan jasa atau produk di *web* atau bahkan situs Internet seperti *web portal*, *web* pribadi dan banyak lagi.

2.16 Capstone Project

Proyek ini terdiri dari beberapa bagian yang dikerjakan secara terpisah. Pada diagram keseluruhan proyek, terdapat lima bagian utama. Seperti terlihat pada Gambar 2.6 , bagian A mencakup pengerjaan sensor *Intelligence Control System* yang merupakan perangkat keras yang dipasang pada *smart greenhouse* untuk

memantau kondisi tanaman. Bagian B merupakan pengerjaan sensor *microclimate*, yaitu sensor yang mendeteksi kondisi iklim di sekitar *smart greenhouse*. Bagian D adalah pengerjaan *backend* yang mengelola server sebagai penghubung antara perangkat keras dan perangkat lunak yang dikembangkan. Bagian C adalah pengerjaan *frontend* yang bertugas menampilkan data dari sensor-sensor pada *smart greenhouse* melalui situs web. Terakhir, bagian E adalah pengembangan aplikasi *mobile* yang memvisualisasikan data sensor *smart greenhouse* menggunakan aplikasi mobile.

Penelitian ini fokus pada pengembangan *frontend* pada bagian E , sementara bagian lainnya, seperti perangkat keras dan aplikasi mobile, dikerjakan oleh anggota lain. *Frontend* yang dikembangkan akan mencakup sebuah *dashboard* untuk menampilkan data dari berbagai sensor yang terpasang, yang dirancang untuk diakses melalui perangkat *desktop*. Diagram keseluruhan proyek penelitian dapat dilihat pada Gambar 2.6.



Gambar 2.6 Diagram Keseluruhan Pengembangan Sistem

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian dan pembuatan Skripsi ini dilakukan pada:

Waktu : Januari – Agustus 2024

Tempat` : Laboratorium Telekomunikasi, Teknik Elektro, Universitas Lampung
Desa Gisting Permai, Kec. Gisting, Kab. Tanggamus, Prov. Lampung

3.2 Alat dan Bahan

3.2.1 Alat Penelitian

Tabel 3.1 Alat yang digunakan dalam penelitian

No	Nama Alat	Spesifikasi	Deskripsi
1	Laptop	*Processor Intel Core* i5-10750U, *RAM* 16GB, *SSD* 512GB, Sistem Operasi *windows 10*	Perangkat keras yang digunakan sebagai *compiler* dalam pemrograman
2	*Trello*	*Online Trello*	*Tools* yang digunakan sebagai *kanban board* untuk memvisualkan alur kerja
3	*Visual Studio Code*	Versi 1.80.1	Kode editor untuk membuat aplikasi *website* dan lain-lain
4	*Website Hosting*	*Niaga Hoster*	Tools untuk melakukan hosting *Website*

No	Nama Alat	Spesifikasi	Deskripsi
5	*Figma*	Versi 28.4.12.3	Perangkat lunak untuk melakukan perancangan antarmuka
6	*Git Hub*	Windows Online	Untuk menyinpan file coding secara online.
7	API	Vps	Untuk Menyimpan data sensor

3.2.2 Bahan Penelitian

Bahan yang digunakan dalam pengerjaan penelitian dan pengembangan sistem ini berupa data yang didapatkan dari backend yang mengirimkan data yang berasal dari REST API dan data MQTT:

Tabel 3.2 Bahan REST API yang telah dikirim oleh backend tim penelitian ini

End point last data	https://c-greenproject.org:3333/getOneData
End point 10 data	https://c-greenproject.org:3333/getdatatable
End point chart data	https://c-greenproject.org:3333/getdatachart
End point cahart daily data	https://c-greenproject.org:3333/getDataForOneDay
End point chart weekly data	https://c-greenproject.org:3333/getDataForSevenDays
End point chart monthly data	https://c-greenproject.org:3333/getDataForonemonth
End point 100 data	https://c-greenproject.org:3333/download100latestdata
End point daily data	https://c-greenproject.org:3333/downloaddailydata
End point weekly data	https://c-greenproject.org:3333/downloadweeklydata
End point monthly data	https://c-greenproject.org:3333/downloadmonthlydata

Tabel 3.3 Bahan berupa data sensor dari raspberry pi yang dikirimkan oleh server menggunakan MQTT dan REST API

NO	**Data yang diterima**	**Keterangan**	**Tampilan dalam website**
1	*Detail*	Data yang JSON yang menunjukan status Sensor sensor Tanaman dan sensor microclimate, waktu pengiriman data terakhir, waktu pengiriman data dari sensor ke *raspberry pi*.	"count": 10, "result": [{ "timestamp": "06-09-24 08:11:00", "id": 9236494, "ph": 5.43, "tds": 325, "suhu_air":
2	*Datetime*	Data yang menunjukan tanggal.	Timestamp : 06-09-24 08:11:00
3	*Cooling System*	Data menunjukan air mengalir atau tidak	Pompa Air: Tidak Aktif/Aktif
4	*Nutrient Pump*	Data yang menunjukan air nutrisi mengalir atau tidak	Pompa Nutrisi : Aktif/Tidak Aktif
5	*Uv light*	Data yang menunjukan lampu uv aktif atau tidak	LampuUV: Aktif/Tidak Aktif
7	*TDS*	Data Zat yang masuk dalam tank Nutrisi.	TDS (mg/L)
8	*Water Temperature*	Data Suhu yang berada dalam tank.	Water Temperature (°C)
9	*pH*	Data nilai pH di dalam tank .	pH
10	*Wind Direction*	Data arah angin di dalam ruangan	Wind Direction: ESE,NW,S,W,NNW

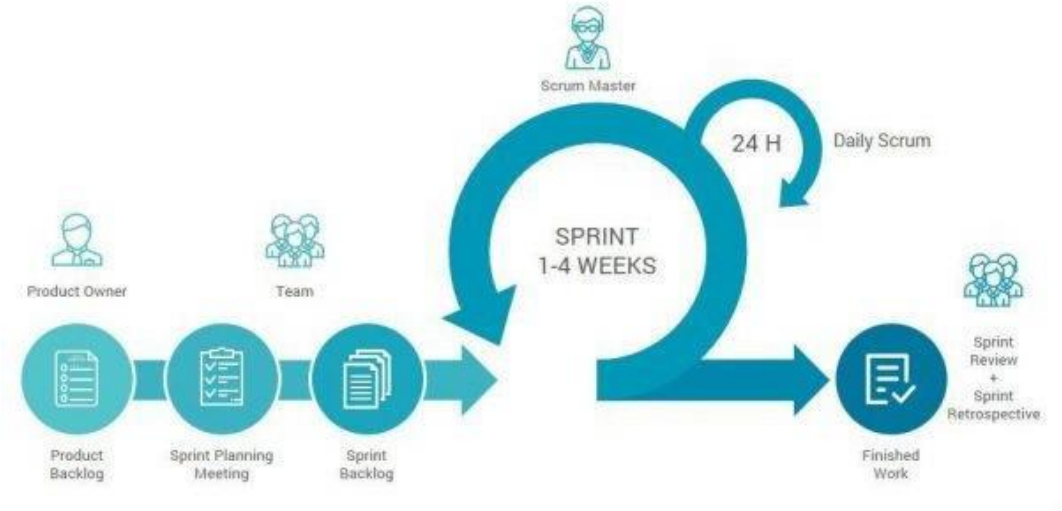
11	*Wind Speed*	Data kecepeatan angina di dalam ruangan.	*Wind Speed* (m/s)
12	*Baromatic Pressure*	Datatekanan udara di dalam ruangan.	*Barometric Pressure* (hPa)
13	*Room Temperature*	Data suhu ruangan di dalam *Greenhouse*.	*Room Temperature* (°C)
14	*Temperature dht*	Data cahaya matahari yang masuk ke dalam *Greenhouse*.	Temperature DHT (°C)
15	*Humidity*	Data kelembapan udara.	Humidity (%RH)
16	*Waterflow*	Data aliran air yang di terima tanaman	Water Flow (L/H)
17	*Soil Moisture*	Data kelembapan tanah di setiap tanaman	Soil Moisture (%)
18	*Weight Table*	Data berat tanaman	Weight (Gram)

Tabel 3.4 Bahan End point dalam bentuk JSON yang telah dikirim oleh back end

https://c-greenproject.org:3333/getOneData	"{\"count\": 1, \"result\": [{\"timestamp\": \"06-09-24 08:11:41\", \"id\": 9236494, \"ph\": 5.43, \"tds\": 325, \"suhu_air\": 31.1, \"winddirection\": 131.53, \"kecepatan_angin\": 0.14, \"berat1\": 0, \"waterflow1\": 0, \"waterflow2\": 0, \"waterflow3\": 0, \"waterflow4\": 0, \"soilmoisture1\": 46.05, \"soilmoisture2\": 47.14, \"soilmoisture3\": 54.18,

3.3 Tahapan Penelitian

Tahapan yang dilakukan dalam penelitian ini mengikuti model pengembangan perangkat lunak Kanban. Berikut merupakan gambar dari tahapan pengembangan menggunakan metode *Scrum*:



Gambar 3.1 Tahap metode *Scrum* [29]

Gambar 3.1 menunjukkan tahapan-tahapan dalam metode Scrum. Dalam metode ini, tim dibentuk dengan pembagian tugas yang jelas. Herly Pratama bertindak sebagai *Scrum* Master, yaitu pemimpin proyek ini, sementara Auliya, Affan, Rey, dan Ade menjadi anggota tim proyek dengan tugas masing-masing. Auliya bertanggung jawab untuk mengintegrasikan semua alat, Affan mengelola penyimpanan data dari sensor, sedangkan Rey dan Ade bertugas memvisualisasikan data tersebut melalui *website* dan aplikasi *mobile*. Pengembangan perangkat lunak dilakukan dalam serangkaian iterasi yang disebut sprint. Durasi setiap *sprint* berkisar antara satu hingga empat minggu, tergantung pada kebutuhan proyek. Setiap sprint memiliki tujuan yang jelas, yaitu menghasilkan inkrementasi produk yang siap digunakan. Berikut adalah tahapan-tahapan dalam metode *Scrum*.

3.3.1 User Story dalam Metode *Scrum*

Pada tahap ini, disusun daftar *User Story* yang digunakan sebagai pedoman untuk menentukan *backlog* dalam pengembangan aplikasi. *Backlog* merupakan hasil turunan atau *breakdown* dari *User Story* tersebut. Format *User Story* yang digunakan dalam penelitian ini mengikuti struktur berikut:

Saya [pengguna] ingin melakukan [kegiatan] agar [hasil yang diharapkan]

Gambar 3. 2 Format *User Story* yang digunakan dalam penelitian.

Pada gambar 3.2, Format User Story yang digunakan dalam penelitian akan mencakup tiga elemen: pengguna, kegiatan yang dilakukan, dan hasil yang diharapkan. Berikut User Story yang berfokus pada pengembangan frontend aplikasi *Agri-Intelligence* Control System (ICS) Smart *Greenhouse* dengan tujuan utama pemantauan:

1. Sebagai seorang petani, saya ingin melihat suhu dalam *Greenhouse* secara *real-time* melalui dashboard agar kondisi lingkungan tetap optimal untuk pertumbuhan tanaman.
2. Sebagai seorang petani, saya ingin memantau status pompa nutrisi dan sistem irigasi melalui antarmuka aplikasi agar saya dapat memastikan tanaman mendapatkan nutrisi yang diperlukan tanpa harus berada di lokasi.
3. Sebagai seorang petani**,** saya ingin mengunduh dan memantau data lingkungan melalui antarmuka frontend agar saya dapat menganalisis perubahan kondisi lingkungan dan mengambil keputusan berdasarkan data yang tersedia.
4. Sebagai seorang petani**,** saya ingin menerima notifikasi jika suhu atau kelembapan tidak optimal agar tindakan pencegahan dapat dilakukan segera untuk menjaga kondisi tanaman.

Dengan format ini, kebutuhan pengguna difokuskan pada pengembangan fitur-fitur *frontend* yang tidak hanya memantau, tetapi juga memungkinkan pengguna untuk 45 kebutuhan pengguna.

Tahap ini dilakukan untuk mendefinisikan mengenai pengguna yang menggunakan aplikasi serta tujuan pengembangan aplikasi. Studi kasus penelitian ini dilakukan untuk pengembangan website informasi data *AGRI-ICS* yang menggunakan protokol *https*. Setelah mengetahui atas kebutuhan dilanjutkan dengan hal yang berhubungan dengan aktivitas yang akan dilakukan dalam pengembangan sistem. Target pengguna website ini adalah masyarakat dapat mengatur proses penanaman dan kebutuhan nutrisi tanaman lebih optimal.

3.3.2 *Sprint Planning*

Sprint Planning akan memiliki beberapa tahapan khusus yang berkaitan dengan karakteristik dan kebutuhan proyek tersebut. Berikut adalah beberapa hal yang mungkin dibahas dalam *Sprint Planning* untuk pengembangan platform IoT Smart *Greenhouse* yaitu ,Tujuan Sprint, Analisa Kebutuhan, penyusunan Sprint *Backlog*, rencana pengujian

3.3.3 *Daily stand up*

Daily Stand-up adalah pertemuan singkat setiap hari di mana tim pengembangan berbagi update tentang kemajuan, rencana kerja, dan hambatan yang dihadapi. Ini membantu menjaga tim tetap terorganisir, sinkron, dan dapat mengatasi masalah dengan cepat.

3.3.4 *Sprint Execution*

Tim pengembangan bekerja untuk menyelesaikan tugas-tugas yang ada dalam Sprint *Backlog* selama sprint. Mereka fokus pada pemenuhan tujuan sprint dan menciptakan inkrementasi produk yang dapat digunakan. Tim melakukan pekerjaan kolaboratif, memantau kemajuan mereka secara teratur, dan beradaptasi dengan perubahan yang mungkin terjadi selama sprint. Ini adalah waktu untuk menjalankan rencana yang dibuat selama *Sprint Planning* dan mencapai komitmen yang telah dibuat oleh tim.

3.3.5 *Sprint Review*

Setiap sprint di mana tim pengembangan memperlihatkan hasil kerja mereka kepada pemangku kelengkapan, seperti petani atau pemilik rumah kaca. Mereka memperlihatkan fitur atau perubahan yang telah selesai, seperti peningkatan sensor atau fitur pengendalian, dan menerima umpan balik tentang kegunaan dan kinerja platform. Umpan balik ini digunakan untuk memperbaiki dan mengembangkan platform lebih lanjut, memastikan bahwa ia tetap relevan dan efektif dalam mendukung pertanian berkelanjutan.

3.3.6 *Sprint Retrospective*

Sprint Retrospective adalah pertemuan di akhir setiap sprint di mana tim pengembangan melakukan evaluasi diri untuk memeriksa apa yang telah berjalan

baik, apa yang perlu diperbaiki, dan bagaimana mereka bisa meningkatkan proses kerja mereka di sprint berikutnya. Dalam konteks pembuatan platform IoT Smart *Greenhouse*, *Sprint Retrospective* dapat membahas efektivitas sensor, keandalan sistem, atau kemampuan platform dalam memenuhi kebutuhan petani. Ini membantu tim untuk terus meningkatkan produk dan proses pengembangan mereka agar dapat memberikan nilai yang lebih baik kepada pemangku kelengkapan.

3.4 Penentuan environment

Struktur *system project* ini ditentukan berdasarkan *Library* ataupun aplikasi yang digunakan sesuai dengan kebutuhan pengembangan project aplikasi. Beberapa *Library* dipilih sebagai program untuk membuat aplikasi ini, ada pula resource lain yang digunakan dalam pengembangan project. Berikut *environment* dalam mengembangkan project aplikasi ini.

project aplikasi. Struktur tersebut sudah sesuai dibuat berdasarkan sistem yang dipilih untuk mengembangkan aplikasi ini.

Dalam project yang akan dikembangkan tersebut terdapat beberapa bagian yang akan dikerjakan secara terpisah. Beberapa bagian tersebut adalah sebagai berikut :

Tabel 3. *5* Tim pengembangan project

No	*Nama*	*Bagian*
1	*M.Herly Pratama (Project manager)*	*Hardware AGRI-ICS*
2	*Auliya Putra Siregar(Team)*	*Hardware Climate*
3	*M Affan Shidiq(Team)*	*Back-end Developer Web*
4	*Alpriealian Renando(Team)*	*Front-end Developer Web*
5	*Ade Iqbal(Team)*	*Android Developer*

Pada pengembangan ini pengembang fokus pada *frontend website* yang menggunakan *reactJS* yang sebagai *Library* utama pembuatan *website*.

3.5 Penentuan *backlog*

Semua *user story* yang telah didefinisikan sebelumnya lalu dilakukan *breakdown* yang kemudian menjadi sebuah *backlog. Backlog* yang nantinya *backlog. Backlog* yang didapat kemudian diletakan sebagai *task card* pada *kanban board* menggunakan *tools Trello. Backlog* atau *task card* inilah yang kemudian digunakan dalam proses pengembangan.

Gambar 3.3 Task card

Gambar 3.3 merupakan *breakdown* dari *user story* yang telah ditentukan sebelumnya hingga menjadi beberapa *backlog*. Kemudian dari *backlog* yang didapat diletakkan sebagai *task card* pada *Scrum board* menggunakan *tools Trello.*

3.5.1 Task card

Backlog yang telah didapatkan kemudian dituliskan pada sebuah *task card* yang ada pada *trello.* Format pada *task card* yang digunakan adalah sebagai berikut:

a. Label

Label merupakan tanda yang digunakan dalam setiap *task card*. Label yang digunakan dalam pengembangan aplikasi ini adalah label status dari *task card* tersebut yang menunjukkan seberapa lengkap *task card* tersebut harus segera selesai.

Gambar 3.4 Label yang digunakan dalam *Scrum* board

Gambar 3.4. menunjukan label yang digunakan dalam *task card* di *Scrum board* pada proses pengembangan aplikasi dari referensi *Scrum: The Art of Doing Twice the Work in Half the Time* oleh *Jeff Sutherland*. Terdapat 6 buah label yang masing-masing memiliki keterangan sebagai berikut:

1. Label *slow* berwarna hijau gelap diberikan untuk menandakan bahwa *task card* memiliki *deadline* berdasarkan *requirement* yang dibutuhkan dan dengan rentang waktu lebih dari 5 hari.
2. Label *medium* berwarna kuning diberikan pada *task card* dengan berdasarkan *requirement* yang dibutuhkan dan rentang waktu antara 2 sampai dengan 5 hari.
3. Label *urgent* berwarna merah menunjukkan bahwa *task card* tersebut berdasarkan *requirement* yang dibutuhkan dan memiliki *deadline* kurang dari 24 jam.
4. Label *verify* berwarna biru cerah menunjukkan bahwa *task card* tersebut perlu diverifikasi apakah telah memenuhi sesuai kebutuhan atau belum.
5. Label *done* berwarna ungu diberikan pada *task card* yang telah selesai dikerjakan.
6. Label *Late* berwarna merah mudah menunjukan bahwa *taskbar* tersebut mengalami keterlambatan atau kendala pengerjaan.

Dalam pengerjaan task card terdapatan yang harus di lakukan yang pertama membuat halaman *landing page* ketika sudah selesai melanjutkan pembuatan halaman *dashboard monitoring system* setelah selesai langkah selanjut nya pembuatan halamn *marketplace*

b. Deskripsi

Deskripsi *task card* yaitu keterangan dari tugas/*backlog* yang harus diselesaikan serta *role (developer, ui designer, tester)* yang harus mengerjakannya.

c. Deadline

Tanggal yang tampil pada *task card* merupakan batas waktu terakhir atau *deadline* dari *task card* tersebut. Setiap *task card* yang ada diberikan batas waktu atau *deadline.*

3.6 Tahap Pengembangan Sistem

Tahapan ini dilakukan penulisan kode dari sistem yang telah dirancang. Perancangan sistem yang telah dilakukan harus diimplementasikan dalam bentuk kode. Hasil dari proses ini berupa website yang suap untuk dilakukan pengujian atau testing menggunakan metode *blackbox testing*. Sesuai dengan metode *Scrum*, maka dalam proses pengembangan menggunakan prinsip-prinsip diantaranya adalah sebagai berikut:

1. Visualisasi Alur Kerja

   Proses ini adalah alur penggembangan pada sebuah *Scrum Board.* Dalam penelitian ini *Scrum Board* digambarkan menggunakan *tools Trello*. Proses visualisasi alur merupakan penggambaran alur pengembangan pada sebuah *Scrum board*. Dalam penelitian ini sendiri *Scrum board* digambarkan menggunakan *tools* bernama *Trello.*

2. Membatasi *Work in Progress* (pekerjaan yang sedang berjalan)

   Dalam proses pengembangan aplikasi ini jumlah *Work in Progress* pada *Scrum board* maksimal 2 *task card*.

3.6.1 Tahap Testing

Tahapan testing atau pengujian dilakukan menggunakan metode Blackbox testing. Pengujian dilakukan untuk menemukan kesalahan-kesalahan yang mungkin terjadi untuk dilakukan perbaikan. Pengujian dilakukan secara langsung pada *website* Agri-ics-net.org. Teknik pengujian ini menggunakan panduan pada *decision table testing*. *Blackbox testing* atau pengujian secara langsung pada website ini dilakukan dengan cara mengecek aplikasi terhadap use case yang telah dirancang. Hasil akhir dari pengujian adalah aplikasi telah bebas dari kesalahan atau bug sehingga diharapkan saat sampai ke tangan pengguna website sudah siap digunakan.

3.6.2 Tahap Pelaporan

Tahap akhir dari penelitian ini adalah pelaporan hasil dan temuan penelitian mengenai *Website sistem Agri_ICS Smart Greenhouse berbasis IOT menggunakan library React.js*. Dari data yang dihasilkan dan telah dianalisis kemudian dilakukan

pengambilan kesimpulan dan saran. Hasil temuan yang ada kemudian digunakan sebagai skripsi pada Universitas Lampung.

3.6.3 Tahap Analisisa Keberhasilan

Tahap analisa hasil dilakukan untuk menguji keberhasilan dari pengembangan *frontend* aplikasi sebagai *dashboard* monitoring *Agri-Intelligence Control System (ICS)* yang telah selesai dikembangkan. Analisa keberhasilan ini dilakukan dengan beberapa indikator pengujian yang meliputi:

1. Fungsionalitas Aplikasi

   Pengujian fungsionalitas dilakukan untuk memastikan bahwa semua fitur dalam *frontend* web berjalan sesuai dengan spesifikasi yang telah ditentukan. Pengujian ini mencakup verifikasi apakah pengguna dapat melihat data sensor secara *real-time*, memantau dan mengontrol perangkat seperti pompa nutrisi, irigasi, dan lampu UV, serta memastikan bahwa tidak terdapat kesalahan (bug) atau gangguan saat aplikasi digunakan. Pengujian dilakukan dengan menggunakan metode *black-box testing* untuk menguji setiap fitur pada *dashboard*.

2. Kompatibilitas Browser

   Uji kompatibilitas browser dilakukan untuk memastikan bahwa *frontend* web berjalan dengan baik di berbagai browser populer seperti Google Chrome, Mozilla Firefox, dan Microsoft Edge. Pengujian ini bertujuan untuk memastikan bahwa tampilan antarmuka dan fungsionalitas aplikasi tetap konsisten di berbagai platform browser dan resolusi layar. Pengujian ini juga memastikan bahwa aplikasi tetap responsif dan dapat digunakan dengan baik di berbagai perangkat desktop dan tablet. Metode *black-box testing* diterapkan untuk menguji kompatibilitas di berbagai browser.

3. Pengalaman Pengguna

   Pengujian pengalaman pengguna bertujuan untuk menilai sejauh mana antarmuka aplikasi memberikan kenyamanan dan kemudahan penggunaan bagi pengguna. Pengujian ini memastikan bahwa Desain antarmuka aplikasi telah mengikuti pedoman Desain *UI* dan *UX* yang baik, termasuk keterbacaan teks, aksesibilitas tombol, navigasi yang intuitif, dan kemudahan pengguna dalam berinteraksi dengan *dashboard*. Pengujian ini dilakukan dengan menggunakan metode *black-*

box testing dengan skenario yang dirancang untuk pengguna, serta borang penilaian yang diberikan kepada mereka untuk mengevaluasi pengalaman penggunaan aplikasi.

4. Kinerja Aplikasi

   Uji kinerja aplikasi difokuskan pada mengukur seberapa cepat *frontend* merespons input pengguna dan menampilkan data *real-time* dari *Greenhouse*, serta seberapa efisien aplikasi dalam mengelola sumber daya jaringan. Pengujian ini melibatkan pengukuran waktu buka halaman, waktu muat data sensor, serta analisis seberapa baik aplikasi memproses dan menampilkan data dalam grafik dan tabel. Pengujian dilakukan dengan menggunakan alat pemantau kinerja berbasis browser seperti *Google Chrome DevTools* untuk menganalisis waktu respons, konsumsi sumber daya jaringan, dan optimalisasi pemuatan konten.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dan analisa hasil yang diperoleh, didapat kesimpulan sebagai berikut:

1. Sebuah website Agri-ICS telah berhasil dikembangkan menggunakan library *React.js*. Website ini dilengkapi dengan fitur monitoring yang memungkinkan pemantauan terhadap semua data sensor dalam pengawasan tumbuhan di dalam *Greenhouse*. Fitur yang tersedia mencakup tampilan data terakhir melalui card, tabel yang menampilkan 10 data terakhir, serta grafik yang mampu menganalisis data dalam berbagai rentang waktu (1 hari, 7 hari, dan 30 hari). Pengguna dapat dengan mudah memantau kondisi *Greenhouse* secara *real-time* dan melakukan analisis data untuk mendukung optimalisasi pertumbuhan tanaman.
2. Berdasarkan hasil penelitian yang dilakukan, Sistem yang dikembangkan memungkinkan pengguna untuk menampilkan grafik dengan berbagai opsi rentang waktu (1 hari, 7 hari, dan 30 hari). Opsi ini memberikan fleksibilitas dalam memantau dan menganalisis perkembangan data sensor secara lebih terperinci sesuai kebutuhan. Selain itu, fitur zoom in dan zoom out juga disediakan untuk memperbesar atau memperkecil tampilan grafik, sehingga mempermudah analisis data dengan lebih teliti dan akurat.
3. Fitur pengunduhan data telah dikembangkan menggunakan *useState hook* dan *Axios* untuk memastikan interaksi yang lancar dengan API yang terhubung ke *front-end*. Fitur ini menyediakan empat opsi pengunduhan data, yaitu 100 data terakhir, data selama 1 hari, 7 hari, dan 30 hari. Setiap opsi ini dirancang untuk memberikan fleksibilitas kepada pengguna dalam memilih rentang data yang sesuai dengan kebutuhan analisis.

4. Dengan adanya fitur ini, peneliti dapat dengan mudah mengunduh data yang dibutuhkan tanpa perlu melakukan proses manual, sehingga mempercepat proses pengumpulan data untuk analisis lebih lanjut. Implementasi fitur ini juga membantu dalam menjaga keakuratan data yang diambil, karena semua data yang diunduh berasal langsung dari API yang selalu diperbarui. Hal ini sangat lengkap dalam mendukung penelitian yang membutuhkan data yang valid dan terkini untuk menghasilkan analisis yang lebih mendalam dan akurat.
5. Dalam pengembangan proyek ini, terdapat lima jenis *end point* yang disediakan oleh *back-end*, yaitu API last data, data 1 hari, data 1 minggu, dan data 1 bulan. Seluruh *end point* tersebut telah diintegrasikan ke dalam *front-end*, memungkinkan data untuk ditampilkan secara *real-time* dan disesuaikan dengan kebutuhan pengguna. Setiap *end point* dirancang untuk menyediakan data dengan rentang waktu tertentu, sehingga pengguna dapat memilih tampilan data sesuai dengan periode yang diinginkan. Integrasi ini mempermudah proses komunikasi antara *front-end* dan *back-end*, memastikan bahwa data yang disajikan kepada pengguna selalu akurat dan up-to-date
6. Metode *Scrum* telah diterapkan dalam pengembangan *frontend* sistem *Agri-Intelligence* Control System (ICS) Smart *Greenhouse* berbasis IoT untuk memastikan proses pengembangan yang terstruktur dan adaptif. Pekerjaan dibagi menjadi beberapa *sprint*, sehingga penyelesaian fitur-fitur tertentu dapat dilakukan dalam jangka waktu yang telah ditentukan. Setiap *sprint* diakhiri dengan *Sprint Review* untuk mengevaluasi hasil kerja dan *Sprint Retrospective* untuk meningkatkan proses pengembangan pada *sprint* berikutnya. Melalui *daily stand-up*, kemajuan proyek dan hambatan yang dihadapi dapat dipantau dan diselesaikan secara cepat. Dengan diterapkannya metode *Scrum*, penyesuaian sistem terhadap kebutuhan pengguna dapat dicapai, serta proses pengembangan dapat dipercepat, sehingga produk akhir yang dihasilkan sesuai dengan spesifikasi yang telah ditetapkan.

5.2 Saran

Saran yang dapat diberikan untuk melanjutkan penelitian ini berdasarkan hasil penelitian yang telah dilakukan adalah sebagai berikut:

7. Melakukan pengembangan dalam menambahkan fitur PWA *(Progressive Web APP)* untuk memberikan pengalaman yang mirip dengan aplikasi *native*.
8. Pengembangan sistem kendali otomatis berbasis data sensor juga disarankan, di mana sistem bisa mengambil keputusan secara otomatis untuk menyalakan atau mematikan pompa air, lampu UV, atau pengaturan suhu berdasarkan parameter lingkungan yang dipantau secara *real-time*.

DAFTAR PUSTAKA

[1] Y. G. Hutasoit dan Y. B. Kusuma, "Optimalisasi Pemanfaatan Otomasi *Greenhouse* Dan Hydroponic Dalam Meningkatkan Produksi Dan Keberhasilan Terhadap Pertanian Budidaya Pakcoy Di PT Inamas Sintesis Teknologi," Jurnal Kajian Dan Penelitian Umum, vol. 1, no. 2, Apr. 2023.

[2] S. Angelini dan D. H. Bangkalang, "Rekayasa Kebutuhan Sistem Informasi Distribusi Bantuan Sosial Terintegrasi Menggunakan Pendekatan Arsitektur *Microservices Requirements Engineering For Integrated Social Assistance Distribution Information Systems Using A Microservices Architecture Approach*," Sistemasi: Jurnal Sistem Informasi, vol. 13, no. 3, pp. 1267–1280, 2024. [Daring]. Tersedia pada: http://sistemasi.ftik.unisi.ac.id

[3] M. A. Nuh dan R. N. Utami, "Manajemen Irigasi Berbasis IoT Untuk Lahan Kering," Universitas Islam Indonesia, Yogyakarta, 2024.

[4] V. S. Adelia dan J. L. Ginting, "IoT-Based Plant Health Monitoring System as an Innovative Solution for Optimizing Agricultural Production," dalam Prosiding Seminar Nasional Mahasiswa PRO-PLANT, Bogor, 2023.

[5] S. Sidik dan W. Haryono, "Simulasi Sistem Monitoring dan Pengendalian Suhu dan Kelembaban Ruangan Lab Kalibrasi Berbasis IoT (Internet of Things) (Studi Kasus: PT. Quantum Inti Akurasi)," Scientica: Jurnal Ilmiah Sain dan Teknologi, vol. 2, no. 10, pp. 392–403, 2024.

[6] M. Hadi, "Rancang Bangun Sistem Monitoring Smart Home Menggunakan Energi Cadangan Berbasis Internet of Things (IoT)," Jurnal Pendidikan Sains dan Komputer, vol. 2, no. 02, pp. 341–344, Okt. 2022, doi: 10.47709/jpsk.v2i02.1745.

[7] A. Herliana dan P. M. Rasyid, "Sistem Informasi Monitoring Pengembangan Software Pada Tahap Development Berbasis Web," Jurnal Informatika, vol. 3, no. 1, Apr. 2016.

[8] Y. Trimarsiah dan M. Arafat, “Analisis dan Perancangan Website Sebagai Sarana Informasi Pada Lembaga Bahasa Kewirausahaan dan Komputer AKMI Baturaja,” Matrik: Jurnal Ilmiah, vol. 19, no. 1, pp. 1–10, Apr. 2017.

[9] M. Yusniar, “Smart *Greenhouse* Tanaman Seledri Berbasis Raspberry Pi Menggunakan *Internet of Things* (IoT),” Universitas Muhammadiyah, Surabaya, 2021.

[10] Mambang, *Internet of Things*. Banjarmasin: PT. Pena Persada Kerta Utama, 2023.[Daring].Tersediapada:*https://www.researchgate.net/publication/370044088*

[11] Ranto, “Membangun Frontend Website Sistem Informasi Pemantauan dan Peringatan Dini Bencana Tsunami Berbasis IoT dengan Menggunakan Library REACTJS,” Universitas Lampung, Bandar Lampung, 2023.

[12] Nasution dan L. Iswari, “Penerapan React JS pada Pengembangan Frontend Aplikasi Startup Ubaform,” Automata, no. 2, 2021.

[13] Y. Galahartlambang, T. Khotiah, dan Jumain, “Analisa Performa Aplikasi Web Berbasis Manipulasi DOM dan Virtual DOM,” Seminar Nasional Inovasi Teknologi, vol. 5, no. 1, Jul. 2021.

[14] T. Sulistyorini, E. Sova, dan R. Ramadhan, “Pemantauan Kasus Penyebaran Covid-19 Berbasis Website Menggunakan Framework React Js dan API,” vol. 1, no. 4, Jul. 2022, [Daring]. Tersedia pada: www.corona.jakarta.go.id.

[15] M. F. Santoso, “Teknik Single Page Application (SPA) Layout Web dengan Menggunakan React JS dan Bootstrap,” Jurnal Khatulistiwa Informatika, vol. 9, no. 2, Des. 2021.

[16] Ikhwan, “Implementasi Sistem Grafik pada Aplikasi Penghasilan dan Pengeluaran Tiap Mitra di PT Lintas Jaringan Nusantara Menggunakan CHARTJS,” Universitas Islam Sumatera Utara, Medan, 2024.

[17] S. Benbba, “Safwane Benbba Comparison of D3.Js and Chart.Js as Visualisation Tools,” Tampere University, Tampere, 2021.

[18] I. Kurniawan, Humaira, dan F. Rozi, “REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android,” Jurnal Ilmiah Teknologi Sistem Informasi, vol. 1, no. 4, pp. 127–132, 2020, [Daring]. Tersedia pada: http://jurnal-itsi.org

[19] M. Bender, E. Kirdan, M. O. Pahl, dan G. Carle, “Open-source MQTT evaluation,” dalam 2021 IEEE 18th Annual Consumer Communications and Networking Conference, CCNC 2021, Institute of Electrical and Electronics Engineers Inc., Jan. 2021. doi: 10.1109/CCNC49032.2021.9369499.

[20] E. Di Paolo, E. Bassetti, dan A. Spognardi, “Security Assessment of Common Open Source MQTT Brokers and Clients,” *arXiv preprint arXiv*, Sep 2023, [Daring]. Tersedia pada: http://arxiv.org/abs/2309.03547

[21] W. N. Arizky, “Investigasi *Scrum* Product *Backlog* Pada Pengembangan Perangkat Lunak: Implementasi Pembangunan Sistem Informasi Pusat Karir UIN Jakarta,” Universitas Islam Negeri Syarif Hidayatullah, Jakarta, 2022.

[22] S. Rahmadhani, D. W. Wildana, H. W. Arumdanie, dan L. Hakim, “Penerapan React JS dan Axios untuk Pengembangan Front-end Aplikasi iCare,” *Software Development Digital Business Intelligence and Computer Engineering*, vol. 2, no. 02, hlm. 40–46, Mar 2024, doi: 10.57203/session.v2i02.2024.40-46.

[23] H. D. P L Gunasekara, N. R. G I, A. M. N T, A. M. A M, D. I. De Silva, dan T. Dias, “Effectiveness of Cutting-Edge Technology for Library Management System,” International Journal of Engineering and Management Research, vol. 12, no. 5, 2022, doi: 10.31033/ijemr.12.5.57.

[24] N. Rappin, *Modern CSS With Tailwind: Flexible Styling Without The Fuss*. 2021.

[25] N. A. Ramdhan dan D. A. Nufriana, “Rancang Bangun dan Implementasi Sistem Informasi Skripsi Online Berbasis Web,” Jurnal Ilmiah INTECH, vol. 1, no. 2, 2019.

[26] I. Kurniawato, F. Amsury, Heriyanto, dan M. R. Fadhla, “Pelatihan Pemanfaatan Aplikasi Trello Untuk Meningkatkan Efektifitas Manajemen Proyek Pada Karyawan PT. Jaya Persada Indonesia,” *Jurnal Pengambidan Masyarakat Abditeknoyasa*, vol. 3, Des 2022.

[27] M. Megawaty, I. Suana, dan febriyanto, “Sistem Pendukung Keputusan Pemilihan Ayam Broiler Sehat Berbasis WEB,” *Jurnal Akademika*, vol. 15, 2022.

[28]. M. W. Sripunyadach, “Reseller Web Hosting Service,” Bangkok, Nov 2001.