

**PENGEMBANGAN SISTEM *BACKEND* UNTUK MANAJEMEN
ORGANISASI DAN PERANGKAT PADA PLATFORM IOT (STUDI
KASUS: POKDAKAN BINTANG ROSELA JAYA, PRINGSEWU)**

(Skripsi)

Oleh

BILL VALENTINOV

NPM 2115061073



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**PENGEMBANGAN SISTEM *BACKEND* UNTUK MANAJEMEN
ORGANISASI DAN PERANGKAT PADA PLATFORM IOT (STUDI
KASUS: POKDAKAN BINTANG ROSELA JAYA, PRINGSEWU)**

Oleh
BILL VALENTINOV

Skripsi

**Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA TEKNIK**

Pada
Program Studi Teknik Infomatika
Fakultas Teknik Univesitas lampung



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

PENGEMBANGAN SISTEM *BACKEND* UNTUK MANAJEMEN ORGANISASI DAN PERANGKAT PADA PLATFORM IOT (STUDI KASUS: POKDAKAN BINTANG ROSELA JAYA, PRINGSEWU)

Oleh

BILL VALENTINOV

Pokdakan Bintang Rosela Jaya merupakan kelompok pembudidaya ikan yang berlokasi di Desa Patoman, Kecamatan Pagelaran, Kabupaten Pringsewu. Untuk mendukung proses budidaya, diperlukan alat pemantau kualitas air kolam guna menjaga kesehatan ikan. Berdasarkan kebutuhan tersebut, dibutuhkan sebuah sistem yang mampu mengintegrasikan perangkat-perangkat IoT dengan biaya yang terjangkau, khususnya bagi organisasi berskala kecil. Penelitian ini bertujuan untuk mengembangkan sistem *backend* yang dapat mengelola organisasi dan perangkat pada platform IoT. Sistem dikembangkan menggunakan *framework* NestJS, database PostgreSQL, serta protokol komunikasi MQTT. Metode pengembangan yang digunakan adalah *Agile Scrum* dengan pendekatan *iteratif*. Hasil pengujian menunjukkan bahwa sistem mampu mengelola organisasi dan perangkat dengan baik melalui mekanisme otentikasi dan otorisasi bawaan *framework* NestJS, serta dapat mendukung hingga 400 pengguna aktif secara bersamaan. Dengan demikian, sistem ini dapat dijadikan dasar untuk pengembangan platform IoT yang mendukung multi-organisasi dan multi-perangkat secara efisien dan terstruktur.

Kata kunci: IoT, NestJS, PostgreSQL, MQTT, Manajemen Perangkat.

ABSTRACT

***BACKEND SYSTEM DEVELOPMENT FOR ORGANIZATION AND
DEVICE MANAGEMENT ON AN IOT PLATFORM (CASE STUDY:
POKDAKAN BINTANG ROSELA JAYA, PRINGSEWU)***

By

BILL VALENTINOV

Pokdakan Bintang Rosela Jaya is a fish farming group located in Patoman Village, Pagelaran District, Pringsewu Regency. To support the cultivation process, a water quality monitoring device is required to maintain fish health. Based on this need, a system that can integrate IoT devices at an affordable cost is required, especially for small-scale organizations. This research aims to develop a backend system for managing organizations and devices on an IoT platform. The system was developed using the NestJS framework, PostgreSQL database, and MQTT communication protocol. The development process employed the Agile Scrum methodology with an iterative approach. The testing results show that the system can effectively manage organizations and devices through the built-in authentication and authorization mechanism of the NestJS framework and can support up to 400 active users simultaneously. Therefore, this system can serve as a foundation for developing an IoT platform that supports multi-organization and multi-device management in an efficient and structured manner.

Keywords: IoT, NestJS, PostgreSQL, MQTT, Device Management.

Judul : PENGEMBANGAN SISTEM *BACKEND*
UNTUK MANAJEMEN ORGANISASI DAN
PERANGKAT PADA PLATFORM IOT (STUDI
KASUS: POKDAKAN BINTANG ROSELA
JAYA, PRINGSEWU)

Nama Mahasiswa : Bill Valentino

Nomor Pokok Mahasiswa : 2115061073

Program Studi : S1 Teknik Informatika

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

Wahyu Eko Sulistiono, S.T., M.Sc.

NIP. 197412012001121001

Deny Budiyanto, S.Kom., M.T.

NIP. 199112082019031011

2. Mengetahui

Ketua Jurusan

Ketua Program Studi

Teknik Elektro

Teknik Informatika

Herlinawati, S.T., M.T.

NIP. 197103141999032001

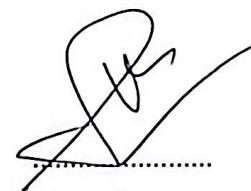
Yessi Mulyani, S.T., M.T.

NIP. 197312262000121001

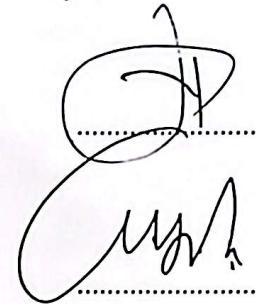
MENGESAHKAN

1. Tim Penguji

Ketua : Wahyu Eko Sulistiono, S.T., M.Sc.



Sekretaris : Deny Budiyanto, S.Kom., M.T.



Penguji : Gigih Forda Nama, S.T.,M.T.I.



2. Dekan Fakultas Teknik

Dr. Eng. Helmy Fitriawan, S.T., M.Sc. J

NIP 197509282001121002

Tanggal Lulus Ujian Skripsi: 20 Oktober 2025

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan dengan sebenarnya bahwa skripsi saya yang berjudul “Pengembangan Sistem *Backend* untuk Manajemen Organisasi dan Perangkat Pada Platfrom IoT (Studi Kasus: Pokdakan Bintang Rosela Jaya, Pringsewu)” merupakan hasil karya saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi saya ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar lampung,

Penulis,



Bill Valentinov

NPM. 2115061073

RIWAYAT HIDUP



Penulis dilahirkan di Bogor pada tanggal 16 November 2003 sebagai anak dari pasangan Bapak Sigit Setijonoegroho dan Ibu Titik Widowati. Penulis memulai pendidikan di SDS Bunda dan lulus pada tahun 2015. Pendidikan menengah pertama diselesaikan di SMPIT Insantama Bogor pada tahun 2018, dan pendidikan menengah atas di SMAIT Insantama Bogor yang diselesaikan pada tahun 2021. Selanjutnya, penulis melanjutkan pendidikan tinggi di Universitas Lampung, Program Studi S1 Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, melalui jalur SBMPTN (Seleksi Bersama Masuk Perguruan Tinggi Negeri). Selama menjalani masa perkuliahan, penulis aktif berpartisipasi dalam berbagai kegiatan, antara lain:

1. Mengikuti program Merdeka Belajar Kampus Merdeka (MBKM) Studi Independen bidang *Cloud Computing* di Bangkit.
2. Melaksanakan kerja praktik di Sekretariat Jenderal Dewan Perwakilan Rakyat Republik Indonesia (Setjen DPR-RI) dengan tugas utama melakukan implementasi ulang aplikasi KSAP.
3. Melaksanakan Kuliah Kerja Nyata (KKN) di Desa Bakti Negara, Kecamatan Pakuan Ratu, Kabupaten Way Kanan, Provinsi Lampung.

MOTTO

طَلَبُ الْعِلْمِ فَرِيضَةٌ عَلَى كُلِّ مُسْلِمٍ وَمُسْلِمَةٍ

(HR. Muslim)

“Bila kaum muda yang telah belajar di sekolah dan menganggap dirinya terlalu tinggi dan pintar untuk melebur dengan masyarakat yang bekerja dengan cangkul dan hanya memiliki cita-cita yang sederhana, maka lebih baik pendidikan itu tidak diberikan sama sekali”

(Tan Malaka)

“Be an engineer, not a framework.”

(Software Engineer)

PERSEMBAHAN

Alhamdulillahirabbil‘alamin

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga skripsi ini telah diselesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, teladan sepanjang masa.

Kupersembahkan Skripsi Ini Kepada

Kepada orang tuaku tercinta, Ibu Titik Widowati, Ibu Vonny Primasari, Bapak Sigit Setijonoegroho dan Bapak Abdollah Anang, yang tak pernah lelah memberikan doa-doa terbaik sepanjang hidup mereka untukku serta memberikan pelajaran yang terpenting dalam hidup ini. Terima kasih sebesar-besarnya atas kasih sayang, dukungan, dan pengorbanan yang tak ternilai, yang tak akan pernah mampu terbalaskan. Semoga Allah SWT senantiasa memberikan perlindungan dunia dan akhirat, Aamiin.

Untuk adik-adiku, Dzafira Nayla Valentinur, Kanaya Kinanti Valentinur, dan Abdillah Pradita Valentinur, sebagai penyemangat dan inspirasi untuk menempuh pendidikan ke jenjang yang lebih tinggi. Semoga Allah SWT senantiasa melimpahkan rahmat, berkah, dan perlindungan-Nya kepada kita semua. Aamiin.

Seluruh Dosen, segenap civitas, serta rekan rekan seperjuangan Teknik Informatika Universitas Lampung, atas motivasi, diskusi, dan kebersamaan yang menjadikan perjalanan akademik ini penuh makna.

SANWACANA

Alhamdulillahirabbil'alamin, penulis mengucapkan puji syukur atas kehadiran Allah SWT, Tuhan Yang Maha Esa, atas segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi berjudul “Pengembangan Sistem *Backend* untuk Manajemen Organisasi dan Perangkat pada Platform IoT (Studi Kasus: Pokdakan Bintang Rosela Jaya, Pringsewu)”.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi S1 Teknik Informatika, Fakultas Teknik, Universitas Lampung. Proses penyusunan skripsi ini berlangsung selama kurang lebih tujuh bulan, berbekal pengalaman dan pembelajaran yang telah penulis tempuh selama empat tahun masa perkuliahan.

Penulis menyadari bahwa penyelesaian skripsi ini tidak terlepas dari dukungan, bimbingan, dan motivasi dari berbagai pihak. Oleh karena itu, dengan tulus hati, penulis mengucapkan terima kasih kepada:

1. Allah SWT, Tuhan Yang Maha Esa, atas segala karunia dan kekuatan yang diberikan.
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. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika.
5. Bapak Wahyu Eko Sulistiono, S.T., M.Sc., selaku Pembimbing Utama, atas bimbingan, bantuan, dan arahan yang sangat berarti selama proses penelitian hingga terselesaikan dengan baik.
6. Bapak Deny Budiyanto, S.Kom., M.T., selaku Pembimbing Pendamping, atas masukan dan arahan yang diberikan dalam penyusunan penelitian ini.
7. Bapak Gigih Forda Nama, S.T., M.T.I., selaku Penguji, atas saran, masukan, dan ilmu yang telah diberikan demi perbaikan penelitian ini.
8. Bapak Mahendra Pratama, S.T., M.Eng., selaku Pembimbing Akademik.

9. Terutama dan paling utama kepada orang tua dan keluarga atas doa, dukungan, semangat, dan bantuan materiel yang tiada henti hingga skripsi ini dapat terselesaikan dengan baik.
10. Seluruh civitas akademika Jurusan Teknik Elektro Universitas Lampung atas segala bantuan administratif dan dukungan selama proses penelitian.
11. Seluruh tim IoT Bridge atas kerja sama dan dukungan yang kuat dalam menyelesaikan pengembangan aplikasi ini.
12. Dan kepada semua pihak yang tidak dapat disebutkan satu per satu, terima kasih atas kontribusi, baik secara langsung maupun tidak langsung, dalam penyelesaian skripsi ini.

Penelitian ini masih jauh dari kata sempurna karena terbatasnya pengalaman dan pengetahuan yang dimiliki. Oleh karena itu, penulis sangat mengharapkan segala bentuk saran serta masukan, bahkan kritik yang membangun dari berbagai pihak. Semoga skripsi ini dapat bermanfaat bagi para pembaca dan semua pihak, khususnya dalam bidang pengembangan perangkat lunak serta *Internet of Things*.

Bandar lampung,
Penulis,

Bill Valentinov

DAFTAR ISI

ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xx
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penelitian.....	4
BAB II TINJAUAN PUSTAKA	5
2.1 Landasan Teori.....	5
2.1.1 <i>Internet of Things (IoT)</i>	5
2.1.2 Platform IoT	5
2.1.3 Node.js.....	6
2.1.4 NestJS	6
2.1.5 Database	7
2.1.6 PostgreSQL.....	8
2.1.7 <i>Message Queuing Telemetry Transport (MQTT)</i>	8
2.1.8 Eclipse Mosquitto	9
2.1.9 Nginx	9
2.1.10 GitHub	10
2.1.11 <i>Virtual Private Server (VPS)</i>	10
2.1.12 Metodologi <i>Agile Scrum</i>	11
2.1.13 Trello	11
2.2 Kondisi Sebelumnya	12
2.3 Penelitian Terkait	14
2.3.1 Sistem Kontrol Suhu dengan PID dan <i>Monitoring Daya Output</i> pada Panel <i>Photovoltaic Portable</i> dengan <i>Virtual Private Server</i>	14

2.3.2	Penerapan Node.js dan PostgreSQL sebagai <i>Backend</i> pada Aplikasi <i>E-commerce</i> Localla	14
2.3.3	Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode <i>Scrum</i>	14
2.3.4	Rancang Bangun Dashboard IoT untuk <i>Monitoring</i> Kolam Akuaponik Berbasis Android	15
2.3.5	REST API Menggunakan Node.js pada Aplikasi Transaksi Jasa Elektronik Berbasis Android	15
2.3.6	Perancangan <i>Back-End</i> Server Menggunakan Arsitektur REST dan Platform Node.js (Studi Kasus: Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang)	15
2.3.7	Aplikasi Android untuk <i>Monitoring</i> Lahan Pertanian secara <i>Realtime</i> Berbasis <i>Internet of Things</i>	16
2.3.8	Implementasi Sistem IoT Pada Akuakultur dan Hydroponik (Akuaponik) Modern Untuk Pertumbuhan Ikan Nila.....	16
2.3.9	iBikeSafe: <i>A Multi-Parameter System for Monitoring, Evaluation, and Visualization of Cycling Paths in Smart Cities Targeted at Cycling Adverse Conditions</i>	17
2.3.10	<i>Vehicle Accident Management and Control System Using MQTT</i> 17	
2.3.11	<i>IoT Platform for COVID-19 Prevention and Control: A Survey</i> 17	
2.3.12	<i>A Real-Time Trust Management Model Using Digital Twin in IoT Networks</i>	18
2.3.13	<i>A Lightweight Authentication Scheme for Power IoT Based on PUF and Chebyshev Chaotic Map</i>	18
2.3.14	<i>Learned Query Optimizers: Evaluation and Improvement</i>	18
2.3.15	<i>A Conceptual Modeling Approach of MQTT for IoT-Based Systems</i> 19	
2.3.16	<i>Integrating Real-Time Wireless Sensor Networks into IoT Using MQTT-SN</i>	19
2.3.17	<i>A Systematic Literature Review of Internet of Things for Higher Education: Architecture and Implementation</i>	19
2.3.18	<i>A Systematic Review of the IoT in Smart University: Model and Contribution</i>	20

2.3.19 <i>Elliptic Curve Cryptography based Security on MQTT System for Smart Home Application</i>	20
BAB III METODOLOGI PENELITIAN	21
3.1 Waktu dan Tempat	21
3.2 Alat dan Bahan.....	21
3.3 Metode Penelitian	23
3.3.1 Tahapan Penelitian.....	24
3.3.2 Tahapan Analisis Kebutuhan	25
3.3.3 Tahapan Pengembangan Sistem	39
3.3.4 Tahapan Penulisan Laporan.....	41
BAB IV HASIL DAN PEMBAHASAN	42
4.1 Proses Pengembangan Sistem dengan Metode <i>Scrum</i>	42
4.2 <i>Product Backlog</i>	42
4.3 Sprint.....	43
4.4 Implementasi <i>Product Backlog</i>	44
4.4.1 Perancangan dan Perencanaan Sistem.....	45
4.4.2 Setup Awal	55
4.4.3 Autentikasi.....	56
4.4.4 Manajemen Organisasi	63
4.4.5 Manajemen Perangkat	70
4.4.6 Setup VPS.....	75
4.4.7 Integrasi MQTT	78
4.4.8 Visualisasi Data <i>Real-Time</i>	82
4.4.9 Deployment Backend	84
4.4.10 Optimalisasi dan Keamanan	89
4.4.11 Notifikasi Untuk Perangkat Mobile.....	91
4.4.12 Penyesuaian Akhir	94
4.4.13 Stress Testing.....	99
4.4.14 Persiapan Lingkungan <i>Production</i>	105
4.5 Evaluasi dan Pengujian Sistem	106
4.5.1 Tujuan Pengujian	106
4.5.2 Metodologi Pengujian	107
4.5.3 Hasil Pengujian.....	107

4.5.4	Analisis Hasil.....	111
4.5.5	Kesimpulan.....	112
BAB V KESIMPULAN DAN SARAN		113
5.1	Kesimpulan	113
5.2	Saran	114
DAFTAR PUSTAKA		115
LAMPIRAN		120

DAFTAR GAMBAR

Gambar 1. Metode scrum [18].....	11
Gambar 2. Perangkat pemantauan kualitas air di Pokdakan.	12
Gambar 3. Biaya langganan dan batasan akun pada Blynk.....	13
Gambar 4. Data JSON dari perangkat IoT.....	23
Gambar 5. Alur tahapan penelitian.	24
Gambar 6. Diagram keseluruhan pengembangan sistem.	27
Gambar 7. Use case diagram aplikasi.....	29
Gambar 8. Activity diagram registrasi.....	31
Gambar 9. Activity diagram login.	32
Gambar 10. Activity diagram memperbarui data profil.	33
Gambar 11. Activity diagram mengajukan organisasi.....	33
Gambar 12. Activity diagram memverifikasi organisasi.	34
Gambar 13. Activity diagram menambahkan anggota organisasi.	34
Gambar 14. Activity diagram membuat local member.....	35
Gambar 15. <i>Activity</i> diagram mengatur notifikasi berdasarkan data dari perangkat.	35
Gambar 16. Activity diagram mengatur dashboard.....	36
Gambar 17. Activity diagram mengirimkan data dari sensor.	36
Gambar 18. Activity diagram melihat data history perangkat IoT.	37
Gambar 19. Activity diagram melihat data dari perangkat.....	38
Gambar 20. Entity Relationship Diagram (ERD).....	47
Gambar 21. Header request authorization.	58
Gambar 22. Implementasi kode modul auth.....	59
Gambar 23. Implementasi kode middleware autentikasi.	59
Gambar 24. Dokumentasi auth API.....	60
Gambar 25. Hasil kode middleware autentikasi.	61
Gambar 26. Hasil pengujian auth API.....	62
Gambar 27. Hasil pengujian auth API.	62
Gambar 28. Hasil pengujian auth API.....	62
Gambar 29. Implementasi kode modul organizations.	66
Gambar 30. Implementasi kode modul notifications.	66

Gambar 31. Implementasi kode modul users.	67
Gambar 32. Implementasi kode middleware organization.	67
Gambar 33. Dokumentasi organizations, notifications, dan users API.	68
Gambar 34. Hasil kode middleware organizations.	69
Gambar 35. Hasil pengujian organizations, notifications, dan users API.	70
Gambar 36. Implementasi kode modul devices.	73
Gambar 37. Dokumentasi devices API.	74
Gambar 38. Hasil pengujian devices API.	75
Gambar 39. Implementasi penyewaan VPS.	76
Gambar 40. Implementasi firewall pada VPS.	77
Gambar 41. Konfigurasi ssh pada VPS.	77
Gambar 42. Implementasi groups sudo pada VPS.	78
Gambar 43. Implementasi mosquitto di VPS.	79
Gambar 44. Implementasi MQTT pada aplikasi backend.	80
Gambar 45. Konfigurasi MQTT pada perangkat IoT.	81
Gambar 46. Menghubungkan perangkat IoT dengan MQTT.	81
Gambar 47. Perangkat IoT mengirimkan data ke MQTT.	82
Gambar 48. Implementasi WebSocket pada aplikasi backend.	83
Gambar 49. Hasil pengujian WebSocket.	84
Gambar 50. Implementasi pembelian domain.	86
Gambar 51. Konfigurasi domain pada VPS.	87
Gambar 52. Implementasi continuous deployment.	88
Gambar 53. Implementasi GitHub secret pada repository.	88
Gambar 54. Implementasi penghapusan data otomatis pada VPS.	90
Gambar 55. Implementasi penghapusan data otomatis pada VPS.	90
Gambar 56. Implementasi rate limiting pada VPS.	91
Gambar 57. Implementasi rate limiting pada VPS.	91
Gambar 58. Konfigurasi Firebase Cloud Messaging (FCM).	92
Gambar 59. Implementasi FCM pada kode aplikasi backend.	93
Gambar 60. Pengujian FCM.	94
Gambar 61. Implementasi dapat menambahkan admin.	95
Gambar 62. Implementasi logic minimal 1 admin.	95
Gambar 63. Implementasi endpoint daftar pin.	96

Gambar 64. Implementasi endpoint daftar pin	96
Gambar 65. Pengujian penambahan admin	97
Gambar 66. Pengujian validasi minimal satu admin	97
Gambar 67. Pengujian validasi minimal satu admin	98
Gambar 68. Pengujian endpoint daftar pin	98
Gambar 69. Flowchart skenario stress testing	100
Gambar 70. Script stress testing	103
Gambar 71. MQTT script test	104
Gambar 72. Hasil stress testing pada VPS	105
Gambar 73. Hasil stress testing pada laptop lokal	105
Gambar 74. Implementasi database untuk lingkungan production	106
Gambar 75. Grafik stress testing pada VPS	108
Gambar 76. Error stress testing pada VPS	109
Gambar 77. Grafik stress testing pada laptop pengembang	110

DAFTAR TABEL

Tabel 1. Waktu Penelitian	21
Tabel 2. Alat yang digunakan dalam penelitian.....	21
Tabel 3. Scrum team.	24
Tabel 4. Tim pengembangan proyek.....	25
Tabel 5. Product backlog.	42
Tabel 6. Jadwal sprint.	44
Tabel 7. Teknologi yang digunakan.....	45
Tabel 8. Struktur tabel users.	47
Tabel 9. Struktur tabel user_notification.	48
Tabel 10. Struktur tabel password_reset_tokens.	49
Tabel 11. Struktur tabel verify_email_tokens.....	49
Tabel 12. Struktur tabel organizations.	50
Tabel 13. Struktur tabel organizations_members.	50
Tabel 14. Struktur tabel devices.	51
Tabel 15. Struktur tabel notification_event.	51
Tabel 16. Struktur tabel widget_boxes.	52
Tabel 17. Struktur tabel device_data.	53
Tabel 18. Spesifikasi dan struktur auth API.	57
Tabel 19. Spesifikasi dan struktur organizations, notifications, dan users API....	64
Tabel 20. Spesifikasi dan struktur devices API.	71
Tabel 21. Hasil stress testing pada VPS.	107
Tabel 22. Hasil stress testing pada laptop lokal.....	109

BAB I

PENDAHULUAN

1.1 Latar Belakang

Digitalisasi telah merevolusi kehidupan manusia dalam beberapa tahun terakhir. Proses digitalisasi ini mengarah pada pengembangan komputer, telepon, dan mesin-mesin lainnya yang awalnya beroperasi secara mandiri. Seiring perkembangan teknologi, saat ini memungkinkan mesin-mesin tersebut untuk saling terhubung dalam jaringan yang luas, sehingga membentuk sistem yang dapat dikendalikan dari jarak jauh. Mesin-mesin yang terhubung tersebut kini mendukung berbagai layanan komunikasi, seperti televisi dan internet [1].

The International Telecommunication Union mendefinisikan *Internet of Things* (IoT) sebagai sebuah infrastruktur global untuk masyarakat informasi yang memungkinkan layanan tingkat lanjut melalui interkoneksi berbagai objek, baik fisik maupun virtual, berdasarkan teknologi informasi dan komunikasi yang *interoperabel* serta terus berkembang [2]. IoT merupakan hasil dari perkembangan perangkat seluler dan sistem tertanam (*embedded systems*). Perkembangan ini menciptakan dunia yang lebih cerdas bagi manusia karena objek di sekitarnya dapat memiliki kemampuan untuk memahami keinginan dan kebutuhan manusia [3].

Tujuan utama dari sistem IoT adalah mendukung pengerjaan banyak tugas, meningkatkan keamanan, dan memungkinkan interaksi dengan pengguna. Oleh karena itu, sistem IoT memiliki beberapa persyaratan, antara lain memiliki memori yang dapat diakses oleh program saat berjalan, bekerja dalam waktu nyata (*real-time*), tidak bergantung pada perangkat keras tertentu, memiliki tingkat keamanan yang tinggi, menggunakan protokol jaringan yang andal, memiliki koneksi yang stabil, menyediakan penyimpanan data yang memadai, serta memiliki sistem manajemen perangkat yang efisien [3].

Pokdakan Bintang Rosela Jaya adalah kelompok pembudidaya ikan yang berlokasi di Desa Patoman, Kecamatan Pagelaran, Kabupaten Pringsewu. Untuk mendukung proses budidaya, diperlukan alat pemantau kualitas air kolam guna menjaga kesehatan ikan. Saat ini, Pokdakan Bintang Rosela Jaya telah menggunakan sistem informasi pemantau kualitas air berbasis IoT yang memanfaatkan aplikasi Blynk. Namun, aplikasi tersebut belum dapat sepenuhnya memenuhi kebutuhan spesifik pengguna.

Aplikasi Blynk memiliki keterbatasan dalam biaya berlangganan, yang mengharuskan pengguna membayar lebih untuk mendapatkan skalabilitas yang lebih besar. Hal ini menjadi beban bagi organisasi kecil. Selain itu, Blynk memiliki keterbatasan dalam hal integrasi dengan sistem lain atau sistem yang telah disesuaikan dengan kebutuhan organisasi. Kontrol terhadap data juga menjadi terbatas karena data tersebut disimpan di server mereka.

Melihat hal tersebut, diperlukan sebuah sistem yang dapat mengintegrasikan perangkat-perangkat IoT dengan biaya yang terjangkau bagi organisasi kecil agar dapat memenuhi kebutuhan yang dibutuhkan. Penelitian ini bertujuan untuk mengembangkan sistem manajemen perangkat dan organisasi pada platform IoT. Sistem ini dirancang untuk menerima data dari perangkat-perangkat IoT, menyimpannya dalam database, dan mengirimkan data tersebut secara langsung jika diperlukan. Dengan adanya sistem ini, diharapkan organisasi kecil dapat lebih mudah menerapkan IoT dalam pemantauan yang sesuai dengan kebutuhan mereka.

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah:

1. Bagaimana cara mengembangkan sistem yang dapat menghubungkan perangkat IoT dengan *frontend* web dan aplikasi *mobile*?
2. Bagaimana sistem dapat mengelola banyak organisasi dan perangkat dengan struktur peran yang telah ditentukan?
3. Bagaimana sistem dapat memastikan hanya perangkat dan pengguna yang berwenang dapat mengakses sistem?

4. Bagaimana cara memastikan sistem dapat tetap stabil dan berfungsi dengan baik saat menerima beban tinggi?

1.3 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah:

1. Penelitian ini berfokus pada pengembangan sistem sisi server (*server-side*) dan konfigurasi pada server.
2. Sistem hanya berfokus pada penyimpanan dan pengiriman data dari perangkat IoT, tidak mencakup pengendalian perangkat.
3. Penelitian ini hanya dilakukan dalam lingkup studi kasus Pokdakan Bintang Rosela Jaya, Pringsewu.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian adalah:

1. Membuat sistem yang dapat menerima, menyimpan, dan mengirimkan data dari perangkat IoT ke aplikasi *frontend* web dan *mobile*.
2. Membuat sistem dengan fitur untuk mengelola organisasi, termasuk pengguna dan perangkat yang terdaftar di dalamnya.
3. Membuat sistem yang dapat memastikan bahwa hanya pengguna dan perangkat yang telah memiliki otorisasi yang dapat mengaksesnya.
4. Mengukur ambang batasan beban yang dapat diterima sistem untuk menjaga kestabilan sistem.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan beberapa manfaat sebagai berikut:

1. Membantu organisasi dalam mengimplementasikan perangkat IoT untuk keperluan pemantauan.
2. Mengembangkan sistem yang dapat mengintegrasikan perangkat IoT dengan biaya yang terjangkau.
3. Membantu instansi terkait dalam melakukan pemantauan kualitas air pada masing-masing kolam secara lebih efisien.

1.6 Sistematika Penelitian

Sistematika penelitian ini akan dibuat menjadi beberapa bagian, dengan penguraian sebagai berikut ini:

BAB I: PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penelitian.

BAB II: TINJAUAN PUSTAKA

Bab ini menjelaskan konsep-konsep dasar yang relevan dengan penelitian, seperti *Internet of Things*, NestJS, NodeJS, PostgreSQL, Eclipse Mosquitto, GitHub, *Virtual Private Server*, metodologi *Agile Scrum*, kondisi sebelumnya, serta penelitian terkait.

BAB III: METODOLOGI PENELITIAN

Bab ini menjelaskan waktu, tempat, alat, bahan, serta metode penelitian beserta tahapan pelaksanaannya.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini berisi pembahasan mengenai hasil yang diperoleh selama proses penelitian.

BAB V: KESIMPULAN DAN SARAN

Bab ini memaparkan kesimpulan yang diperoleh dari penelitian serta memberikan saran untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 *Internet of Things* (IoT)

Menurut The International Telecommunication Union, IoT adalah infrastruktur global untuk masyarakat informasi yang memungkinkan layanan tingkat lanjut melalui interkoneksi berbagai hal (fisik dan virtual) berdasarkan teknologi informasi dan komunikasi yang *interoperabel* serta terus berkembang [2]. Saat ini, sistem IoT telah merambah ke berbagai sektor, seperti kesehatan, pertanian, transportasi, dan bidang lainnya.

Dalam penerapannya, sistem IoT menggunakan sensor untuk memperoleh data dari lingkungan sekitar serta aktuator untuk menghasilkan respons berdasarkan kondisi tertentu yang diperoleh dari sensor. Selain itu, sistem IoT melibatkan komunikasi antar perangkat maupun antara perangkat dan pengguna melalui jaringan internet. Sistem IoT juga dapat mendukung sistem informasi, untuk mengumpulkan data, yang akan dikelola menjadi informasi.

Pada penelitian ini, sistem IoT dirancang untuk memperoleh data dari perangkat IoT, kemudian menyimpannya dalam database. Data tersebut akan dikirimkan ke antarmuka pengguna jika diperlukan, sehingga pengguna dapat memantau serta memanfaatkan informasi yang tersedia secara lebih efektif.

2.1.2 Platform IoT

Platform IoT adalah serangkaian entitas yang didukung teknologi termasuk objek fisik, seperti sensor serta layanan dan sistem perangkat lunak yang terhubung dan bekerja bersama [4]. Selain sebagai penghubung, platform IoT juga memiliki peran dalam manajemen perangkat, keamanan, pemrosesan data, penyimpanan data, visualisasi data, serta analisis data. Saat ini telah tersedia berbagai platform IoT dalam bentuk aplikasi maupun layanan berbasis *cloud*, seperti Blynk, AWS IoT Core, Google Cloud IoT Platform, dan Antares.

Pada penelitian ini, pengembangan platform IoT dilakukan menggunakan arsitektur yang terhubung melalui *Message Queuing Telemetry Transport* (MQTT).

2.1.3 Node.js

Pada awalnya, JavaScript merupakan bahasa pemrograman yang digunakan pada sisi klien di dalam peramban (*browser*) yang dikombinasikan dengan HTML. Namun, dengan Node.js, JavaScript dapat dijalankan pada sisi server. Node.js adalah lingkungan eksekusi JavaScript yang berjalan di sisi server, yang dikembangkan berdasarkan implementasi mesin Google V8 .[5]

Node.js memiliki arsitektur I/O *nonblocking* yang bersifat asinkron dan berbasis peristiwa (*event-driven*). Dalam arsitektur ini Node.js menggunakan *single thread* dalam menangani permintaan (*request*) secara asinkron. Salah satu keunggulan utama Node.js adalah kemampuannya dalam menangani lebih dari 40.000 koneksi pengguna dengan memori RAM 8 GB dengan *single thread*. Sebagai perbandingan, bahasa pemrograman lain, seperti PHP, Java, atau .NET, biasanya menggunakan *thread* baru untuk setiap klien [6]. Dengan menggunakan model *single thread* dan I/O asinkron, Node.js dapat mempermudah pengembangan tanpa harus mengelola *thread*.

Dalam penelitian ini, Node.js digunakan untuk menjalankan kode JavaScript yang telah dikompilasi dari TypeScript pada sisi server. Teknologi ini dipilih karena kemampuannya dalam menangani komunikasi data secara efisien dalam sistem perangkat IoT yang dikembangkan.

2.1.4 NestJS

NestJS adalah sebuah kerangka kerja (*framework*) yang digunakan untuk membangun aplikasi sisi server berbasis Node.js. NestJS mendukung penggunaan TypeScript dan mengombinasikan berbagai paradigma pemrograman, seperti Pemrograman Berorientasi Objek (*Object-Oriented Programming*), Pemrograman Fungsional (*Functional Programming*), serta Pemrograman Reaktif Fungsional

(*Functional Reactive Programming*). Dengan arsitektur yang terinspirasi dari Angular, NestJS dirancang untuk menyelesaikan permasalahan dalam pengembangan arsitektur aplikasi *server-side* berbasis JavaScript secara lebih terstruktur dan efisien.

NestJS memperluas kerangka kerja Node.js seperti Express atau Fastify, dengan menambahkan organisasi kode yang modular serta berbagai pustaka lain untuk menangani tugas yang berulang [7]. Salah satu keunggulan utama NestJS adalah struktur kode yang terorganisir, sistem injeksi ketergantungan (*dependency injection*), skalabilitas, serta dukungan terhadap berbagai fitur bawaan.

Dalam penelitian ini, NestJS digunakan karena mendukung protokol komunikasi berbasis *Message Queuing Telemetry Transport*. Selain itu, NestJS juga menyediakan *WebSocket*, yang digunakan untuk mengirimkan data dari perangkat ke antarmuka pengguna secara *real-time*.

2.1.5 Database

Database digunakan untuk menyimpan, mengelola, dan menyediakan data yang andal serta terorganisir, sehingga dapat diakses dan dianalisis sesuai kebutuhan. Dua jenis database yang umum digunakan saat ini adalah *relational* database dan NoSQL database, masing-masing memiliki kelebihan dan kekurangan. *Relational* database menawarkan pendekatan penyimpanan data yang terstruktur dan terorganisasi melalui penggunaan tabel-tabel yang saling berelasi. Sementara itu, NoSQL database dirancang untuk menangani data tidak terstruktur, serta menawarkan fleksibilitas dan daya skala tinggi yang berguna dalam pengelolaan data dalam jumlah besar [8].

Pada penelitian ini, dipilih penggunaan *relational* database karena kemampuannya dalam melakukan normalisasi data, yang dapat memastikan integritas data dan manajemen yang efisien melalui pembagian data ke dalam tabel-tabel [8].

2.1.6 PostgreSQL

PostgreSQL adalah sistem manajemen database relasional berorientasi objek atau *Object-Relational Database Management System* (ORDBMS) yang dikembangkan berdasarkan proyek Postgres. PostgreSQL pertama kali dikembangkan di University of California, Berkeley dan kini menjadi database sumber terbuka (*open-source*) yang dapat digunakan pada berbagai sistem operasi, seperti Linux, macOS, dan Windows.

Jika dibandingkan dengan sistem manajemen database lainnya, seperti MySQL, PostgreSQL memiliki waktu eksekusi yang lebih cepat dalam melakukan operasi query SELECT dan INSERT. Selain itu, PostgreSQL lebih cocok digunakan pada lingkungan yang memerlukan latensi data rendah serta kemampuan pemrosesan bersamaan yang tinggi [9]. Salah satu keunggulan utama PostgreSQL adalah dukungan terhadap tipe data abstrak, yang memungkinkan pengguna untuk membuat tipe data baru sesuai dengan kebutuhan [10].

Dalam penelitian ini, PostgreSQL dipilih sebagai database *management system*, karena sistem ini akan menangani penyimpanan data dari perangkat IoT secara intensif melalui operasi INSERT serta melakukan banyak query SELECT untuk menampilkan data yang disimpan dari perangkat IoT.

2.1.7 Message Queuing Telemetry Transport (MQTT)

MQTT merupakan protokol komunikasi tingkat aplikasi yang sesuai untuk konteks IoT karena mampu menjembatani sistem penyedia data sensor dengan sistem penerima data. Protokol ini menggunakan arsitektur berbasis konsep *publish* dan *subscribe*, di mana perangkat IoT sebagai penerbit (*publisher*) akan mengirimkan data ke suatu topik tertentu (misalnya Topik A) yang telah dikonfigurasi pada broker MQTT. Selanjutnya, data tersebut akan diteruskan oleh broker kepada pelanggan (*subscriber*) yang telah berlangganan pada topik yang sama [11].

MQTT memiliki berbagai keunggulan, seperti ringan, ukuran paket data yang kecil, penggunaan daya rendah, serta kemudahan dalam implementasi.

Keunggulan-keunggulan tersebut menjadikan MQTT ideal untuk komunikasi dalam sistem IoT [12].

2.1.8 Eclipse Mosquitto

Eclipse Mosquitto merupakan broker pesan sumber terbuka (*open-source message broker*) yang mengimplementasikan protokol *Message Queuing Telemetry Transport* (MQTT). Mosquitto dikembangkan oleh Eclipse Foundation dengan lisensi Eclipse Public License dan Eclipse Distribution License. Protokol MQTT menyediakan metode pengiriman pesan menggunakan model *publish/subscribe*, yang bersifat ringan dan efisien. Hal ini menjadikannya cocok untuk komunikasi data pada perangkat IoT, seperti sensor, mikrokontroler, dan sistem tertanam.

Dalam penerapannya, MQTT menggunakan arsitektur *client-server*, di mana broker MQTT berfungsi sebagai perantara antara klien yang menerbitkan pesan (*publisher*) dan klien yang berlangganan pesan (*subscriber*). Server MQTT bertanggung jawab untuk memproses permintaan berlangganan dan berhenti berlangganan klien, menerima pesan yang diterbitkan oleh klien, serta mendistribusikan pesan ke klien lain berdasarkan topik yang sesuai [13].

Pada penelitian ini, *Eclipse Mosquitto* dipilih sebagai broker pesan MQTT karena sifatnya yang ringan dan sederhana, tanpa antarmuka pengguna grafis dan hanya menggunakan *Command Line Interface*. Hal ini membuatnya sesuai dengan kebutuhan studi kasus penelitian ini, yang mengutamakan efisiensi dan penggunaan sumber daya yang minimal.

2.1.9 Nginx

Nginx merupakan perangkat lunak *web server open source* yang digunakan untuk melayani permintaan HTTP/HTTPS. Nginx memiliki arsitektur modular, berbasis peristiwa (*event-driven*), asinkron (*asynchronous*), berutas tunggal (*single-threaded*), dan nonblok (*non-blocking*). Jika dibandingkan dengan Apache, Nginx memiliki kinerja yang lebih baik dalam hal responsivitas dan skalabilitas, walaupun efisiensinya dapat menurun saat menghadapi beban berat [14].

Dalam penelitian ini, Nginx digunakan sebagai *reverse proxy* untuk meneruskan permintaan dari pengguna ke aplikasi *backend* NestJS serta mengamankan komunikasi antara klien dan server. Selain itu, Nginx akan diterapkan untuk membatasi jumlah permintaan (*rate limiting*) pada server guna mencegah lonjakan lalu lintas yang dapat menyebabkan kelebihan beban. Nginx berperan dalam mengelola lalu lintas yang masuk ke server, sehingga sistem dapat bekerja lebih optimal dan efisien.

2.1.10 GitHub

Github sebagai tempat penyimpanan perangkat lunak untuk proyek yang menggunakan sistem kontrol versi terdistribusi Git [15]. GitHub menyediakan berbagai fitur untuk mengelola proyek perangkat lunak, termasuk repositori Git, manajemen versi, serta alat kolaborasi bagi pengembang.

Pada penelitian ini, GitHub digunakan untuk menyimpan repositori Git yang berisi kode sumber terbuka. Selain itu, penelitian ini juga memanfaatkan fitur GitHub Actions untuk mengotomatiskan *Continuous Integration* dan *Continuous Deployment* dalam proses penyebaran perangkat lunak. GitHub juga menjadi sarana kolaborasi bagi tim pengembang, termasuk tim *backend*, *frontend* web, dan *mobile*, dalam mengelola kode serta melakukan pengawasan terhadap perubahan versi perangkat lunak.

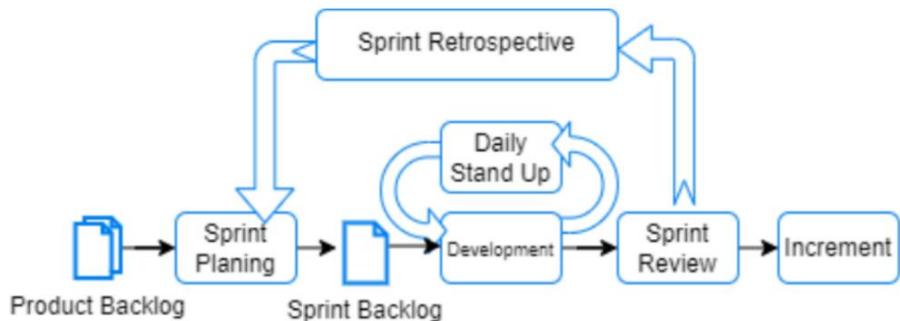
2.1.11 *Virtual Private Server (VPS)*

Virtual Private Server (VPS) adalah server yang sumber dayanya dapat digunakan secara eksklusif oleh satu pengguna tanpa memengaruhi pengguna lainnya. VPS menggunakan teknologi *virtualisasi* yang memungkinkan satu server fisik dibagi menjadi beberapa sumber daya yang dapat dikelola secara independen [16]. Jika menggunakan sistem operasi Linux, VPS dapat diakses melalui terminal atau PuTTY (jika menggunakan Windows) menggunakan protokol *Secure Shell (SSH)*.

Pada penelitian ini, VPS digunakan untuk menempatkan aplikasi *backend*, database PostgreSQL, dan MQTT perantara pesan Eclipse Mosquitto. VPS dipilih karena mampu menyediakan arsitektur perangkat lunak yang berjalan secara kontinu guna mendukung pemantauan kolam secara *real-time*. Selain itu, penggunaan VPS juga lebih ekonomis karena tidak memerlukan pemeliharaan perangkat keras secara langsung.

2.1.12 Metodologi *Agile Scrum*

Metode *Agile Scrum* adalah kerangka kerja yang ringan dengan langkah-langkah terstruktur untuk mengelola pengembangan produk secara iteratif dan bertahap. *Scrum* dirancang untuk meningkatkan kecepatan pengembangan, menyelaraskan pandangan antara individu dan organisasi, membangun budaya kerja yang berorientasi pada kinerja, serta mendukung penilaian pemangku kepentingan [17]. Karena berbasis *Agile*, metode ini memungkinkan untuk pengendalian perubahan yang terjadi selama proses pengembangan.



Gambar 1. Metode *scrum* [18].

Dalam penelitian ini, pengelolaan *Scrum* akan dilakukan menggunakan Trello sebagai alat bantu dalam perencanaan dan pemantauan proyek.

2.1.13 Trello

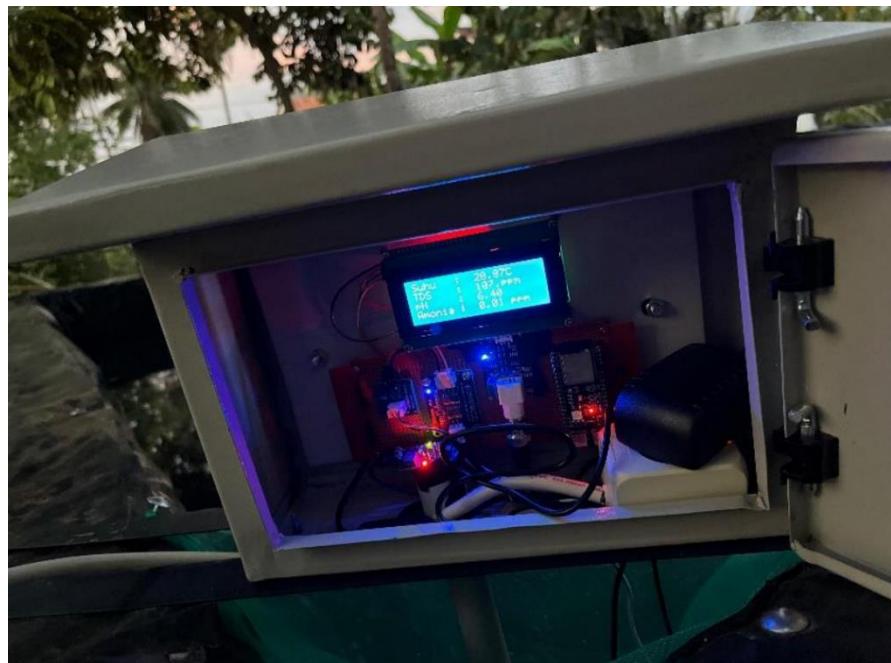
Trello merupakan perangkat lunak berbasis *Software as a Service* yang tersedia dalam versi web, *mobile*, dan *desktop*. Trello berfungsi sebagai alat kolaborasi tim dalam manajemen proyek. Aplikasi ini menyediakan berbagai fitur, seperti

notifikasi email, kalender perencanaan, otomatisasi tugas, serta tema yang berisi cetak biru dari berbagai industri.

Dalam penelitian ini, Trello digunakan sebagai alat bantu dalam manajemen proyek dengan metode *Agile Scrum*. Aplikasi ini akan digunakan untuk mengelola berbagai tahapan dalam pengembangan, seperti *Product Backlog*, *Sprint Backlog*, *Doing*, *Ready to Verify*, *Sprint Review*, dan *Done Sprint*.

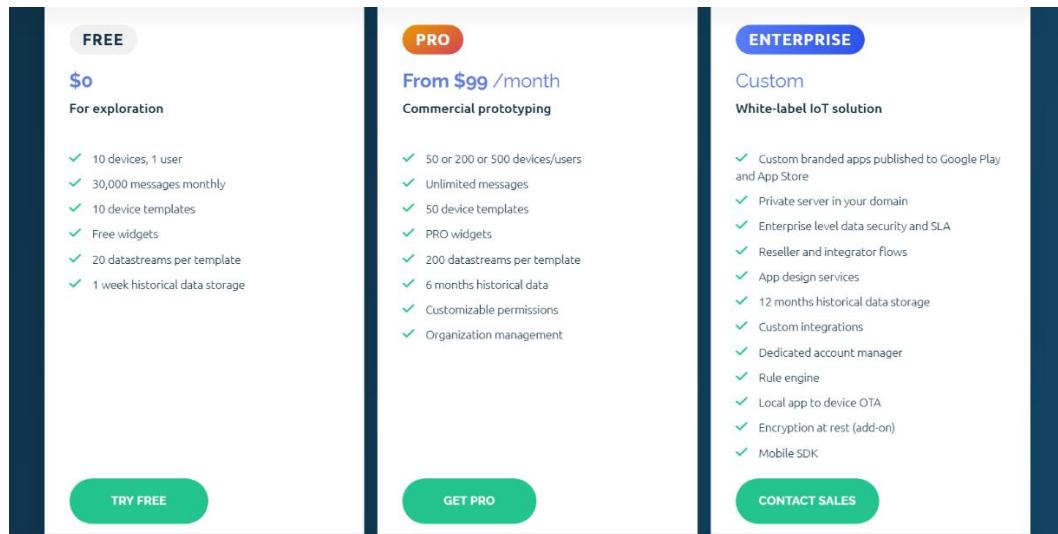
2.2 Kondisi Sebelumnya

Penelitian ini dilakukan berdasarkan sistem IoT yang telah terpasang pada Pokdakan Bintang Rosela Jaya. Sistem IoT yang diterapkan di Pokdakan Bintang Rosela Jaya digunakan untuk memantau kualitas air kolam dengan menggunakan ESP32-S3 sebagai mikrokontroler, sensor pH Robot untuk mengukur pH air, sensor DS18B20 untuk mengukur suhu air, sensor *Total Dissolved Solids* untuk mengukur konsentrasi padatan terlarut dalam air, serta sensor MQ-135 untuk mengukur kadar gas di sekitar kolam. Perangkat IoT tersebut telah diimplementasikan pada keempat kolam.



Gambar 2. Perangkat pemantauan kualitas air di Pokdakan.

Mikrokontroler ESP32-S3 digunakan untuk mengirimkan data sensor ke aplikasi pihak ketiga, yaitu Blynk. Meskipun saat ini sistem IoT tersebut dihentikan sementara dalam penelitian ini, penggunaan Blynk memiliki beberapa keterbatasan. Versi gratisnya membatasi jumlah perangkat yang dapat terhubung, dan karena merupakan layanan pihak ketiga, pengguna tidak memiliki kendali penuh atas data yang dikumpulkan.



Gambar 3. Biaya langganan dan batasan akun pada Blynk.

Penelitian ini dilakukan untuk mengatasi keterbatasan yang terjadi akibat penggunaan perangkat lunak pihak ketiga, seperti Blynk. Oleh karena itu, dikembangkan sebuah aplikasi yang dapat menerima data dari perangkat IoT, menyimpannya dalam database, serta menampilkannya kepada pengguna secara *real-time*.

Penelitian ini berfokus pada pengembangan sisi server dan *deployment*. Perangkat IoT akan mengirimkan data menggunakan protokol *Message Queuing Telemetry Transport*. Data tersebut akan diterima oleh *Eclipse Mosquitto* sebagai *message broker*, lalu diteruskan ke aplikasi *backend*. Pada aplikasi *backend*, data yang diterima akan divalidasi untuk memastikan bahwa data berasal dari pengguna yang sah. Jika valid, data tersebut akan disimpan dalam database dan dapat dikirimkan ke aplikasi *frontend* web atau *mobile* untuk ditampilkan kepada pengguna. Dalam penelitian ini, komponen seperti aplikasi *backend*, database, dan broker MQTT

akan di *deployment* pada server VPS untuk memastikan sistem dapat berjalan secara mandiri dan lebih fleksibel dalam pengelolaan data.

2.3 Penelitian Terkait

2.3.1 Sistem Kontrol Suhu dengan PID dan *Monitoring Daya Output* pada Panel *Photovoltaic Portable* dengan *Virtual Private Server*

Penelitian yang dilakukan oleh Ardian Nurrahman dan Nazori, AZ. pada tahun 2020 yang berjudul “Sistem Kontrol Suhu dengan PID dan *Monitoring Daya Output* pada Panel *Photovoltaic Portable* dengan *Virtual Private Server*”. Penelitian ini membahas tentang pengembangan sistem untuk pemantauan dan kontrol suhu menggunakan *photovoltaic*. Sistem yang dikembangkan dalam penelitian tersebut menggunakan Raspberry Pi sebagai perangkat utama dan penyimpanan data sementara, di mana data tersebut nantinya akan dikirimkan ke *Virtual Private Server* (VPS) untuk keperluan *monitoring* data secara keseluruhan. Penelitian ini tidak membahas sistem manajemen banyak pengguna dan perangkat pada sistem IoT dan hanya berfokus pada satu perangkat yang digunakan [19].

2.3.2 Penerapan Node.js dan PostgreSQL sebagai *Backend* pada Aplikasi *E-commerce Localla*

Penelitian yang dilakukan oleh M. Barokah dan Siti Barokah pada tahun 2022 dengan judul “Penerapan Node.js dan PostgreSQL sebagai *Backend* pada Aplikasi *E-commerce Localla*” mengembangkan aplikasi sisi server menggunakan Node.js dan PostgreSQL untuk mendukung sistem *E-commerce Localla*. Meski teknologi *backend* dan database yang digunakan serupa, penelitian tersebut tidak menggunakan kerangka kerja NestJS dan tidak mengembangkan sistem IoT [20].

2.3.3 Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode *Scrum*

Penelitian oleh Aryana Andipradana dan Kristoko Dwi Hartono pada tahun 2021 berjudul “Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode *Scrum*” membahas pengembangan aplikasi penjualan berbasis web dengan menggunakan metode *Scrum*. Penelitian ini hanya berfokus

pada metodologi pengembangan perangkat lunak yang menggunakan bahasa pemrograman PHP dan tidak menyinggung sistem IoT maupun pengembangan sistem backend [21].

2.3.4 Rancang Bangun Dashboard IoT untuk *Monitoring* Kolam Akuaponik Berbasis Android

Penelitian oleh Kevin Jonathan pada tahun 2023 berjudul “Rancang Bangun Dashboard IoT untuk *Monitoring* Kolam Akuaponik Berbasis *Android*” mengembangkan sistem pemantauan kolam berbasis IoT dengan parameter *Total Dissolved Solids*, pH, dan kekeruhanan air. Aplikasi yang dikembangkan menggunakan React Native dan Firebase. Meskipun relevan dengan sistem IoT untuk pemantauan kolam ikan, penelitian ini tidak terdapat pengembangan *backend* yang mana menggunakan Firebase dan tidak mendukung manajemen banyak perangkat dan pengguna [22].

2.3.5 REST API Menggunakan Node.js pada Aplikasi Transaksi Jasa Elektronik Berbasis Android

Penelitian oleh Irfan Kurniawan, Humairah, dan F. Rozi pada tahun 2020 yang berjudul “REST API Menggunakan Node.js pada Aplikasi Transaksi Jasa Elektronik Berbasis Android” yang membahas mengenai pengembangan aplikasi sisi server dengan arsitektur REST API untuk pencatatan transaksi elektronik menggunakan Node.js, yang akan digunakan oleh aplikasi mobile Android. Meski penelitian ini membahas pengembangan aplikasi sisi server yang menggunakan arsitektur REST API dan menggunakan Node.js. Namun, tidak membahas mengenai sistem IoT [23].

2.3.6 Perancangan *Back-End* Server Menggunakan Arsitektur REST dan Platform Node.js (Studi Kasus: Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang)

Penelitian yang dilakukan oleh Ahsan Mubariz, Dahlia Nur, Eddy Tungadi, dan Muhammad Nur Yasir Utomo pada tahun 2020 dengan judul “Perancangan *Back-End* Server Menggunakan Arsitektur REST dan Platform Node.js (Studi Kasus:

Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang)”. Penelitian ini membahas pengembangan aplikasi sisi server dengan arsitektur REST dan menggunakan Node.js untuk mendukung sistem pendaftaran ujian masuk di Politeknik Negeri Ujung Pandang. Penelitian ini relevan dari sisi arsitektur *backend* menggunakan Node.js. Namun, tidak membahas sistem IoT [24].

2.3.7 Aplikasi Android untuk *Monitoring* Lahan Pertanian secara *Realtime* Berbasis *Internet of Things*

Penelitian yang dilakukan oleh Jihot Lumban Gaol dan rekan-rekannya pada tahun 2020 berjudul “Aplikasi Android untuk *Monitoring* Lahan Pertanian secara *Realtime* Berbasis *Internet of Things*”. Penelitian ini bertujuan untuk mengembangkan aplikasi Android yang mampu memantau kondisi lahan pertanian secara *real-time* dengan memanfaatkan teknologi IoT. Sistem yang dikembangkan menggunakan berbagai sensor untuk mengukur parameter lingkungan seperti suhu, kelembapan tanah, pH tanah, dan intensitas cahaya. Data yang dikumpulkan oleh sensor kemudian dikirimkan ke server melalui koneksi internet dan ditampilkan dalam bentuk teks dan grafik pada aplikasi Android. Namun, penelitian ini tidak membahas mengenai manajemen perangkat, organisasi, dan otorisasi pengguna dalam sistem IoT dan mengembangkan aplikasi Android bukan aplikasi sisi server [25].

2.3.8 Implementasi Sistem IoT Pada Akuakultur dan Hydroponik (Akuaponik) Modern Untuk Pertumbuhan Ikan Nila

Penelitian yang dilakukan oleh Bagas Setiawan, Styawati, dan Syahirul Alim pada tahun 2024 yang berjudul “Implementasi Sistem IoT Pada Akuakultur dan Hydroponik (Akuaponik) Modern Untuk Pertumbuhan Ikan Nila”. Penelitian ini bertujuan untuk mengembangkan sistem *monitoring* dan kontrol berbasis IoT pada akuaponik modern guna meningkatkan pertumbuhan ikan nila. Sistem yang dikembangkan menggunakan sensor-sensor seperti pH, suhu, dan *Total Dissolved Solids* untuk memantau kualitas air secara *real-time*. Data yang dikumpulkan oleh sensor kemudian dikirimkan ke server dan ditampilkan melalui antarmuka berbasis web, yang memungkinkan pembudidaya untuk memantau kondisi lingkungan

kolam secara efisien. Namun, penelitian ini tidak membahas secara mengenai manajemen banyak perangkat dan pengguna dalam sistem IoT dan tidak membahas mengenai aplikasi sisi server [26].

2.3.9 iBikeSafe: A Multi-Parameter System for Monitoring, Evaluation, and Visualization of Cycling Paths in Smart Cities Targeted at Cycling Adverse Conditions

Penelitian yang dilakukan oleh Ivanovitch Silva, Daniel G. Costa, Franklin Oliveira, dan Luciana Lima pada tahun 2021 berjudul “iBikeSafe: A Multi-Parameter System for Monitoring, Evaluation, and Visualization of Cycling Paths in Smart Cities Targeted at Cycling Adverse Conditions”. Penelitian ini mengembangkan sistem pemantauan kondisi lingkungan jalan menggunakan teknologi IoT. Walaupun relevan dengan topik IoT, fokus utama penelitian ini adalah aplikasi IoT untuk *smart city*, bukan pada pengelolaan perangkat dan organisasi dalam sistem IoT [27].

2.3.10 Vehicle Accident Management and Control System Using MQTT

Penelitian oleh Sudha Senthilkumar, K. Brindha, dan S. Bhandari pada tahun 2020 dengan judul “Vehicle Accident Management and Control System Using MQTT” membangun sistem pelacakan kendaraan (*Vehicle Tracking System*) yang dapat mengirimkan data lokasi secara *real-time* jika terjadi kecelakaan menggunakan protokol MQTT sebagai media komunikasi. Penelitian ini menunjukkan efisiensi penggunaan protokol MQTT, tidak membahas tentang manajemen perangkat IoT maupun sistem *backend* [28].

2.3.11 IoT Platform for COVID-19 Prevention and Control: A Survey

Penelitian yang dilakukan oleh Yudi Dong dan Yu-Dong Yao pada tahun 2021 dengan judul “IoT Platform for COVID-19 Prevention and Control: A Survey” membahas tentang penggunaan IoT dalam mencegah dan mengendalikan *COVID-19*. Menggunakan sensor RFID untuk memantau data fisiologis pola pernapasan dan detak jantung dan menggunakan teknologi jaringan sensor nirkabel dan *fog computing* untuk mengelola data secara *real-time*. Penelitian ini hanya membahas

mengenai penggunaan sistem IoT untuk menangani COVID-19 dan tidak membahas pengembangan sistem *backend* [29].

2.3.12 *A Real-Time Trust Management Model Using Digital Twin in IoT Networks*

Penelitian yang dilakukan oleh Meriem Soula, Bacem Mbarek, dan Aref Meddeb pada tahun 2024 berjudul “*A Real-Time Trust Management Model Using Digital Twin in IoT Networks*”. Penelitian ini menggunakan manajemen kepercayaan *real-time* dengan teknologi *digital twin* untuk keamanan dan keandalan jaringan IoT. Namun, penelitian ini hanya menyoroti aspek keandalan dan keamanan sistem IoT untuk pengiriman data secara *real-time* dan tidak membahas pengelolaan perangkat maupun pengguna [30].

2.3.13 *A Lightweight Authentication Scheme for Power IoT Based on PUF and Chebyshev Chaotic Map*

Penelitian yang dilakukan oleh Xianji Jin, Na Lin, Zhongwei Li, Wenqi Jiang, Yuge Jia, dan Qingyang Li pada tahun 2024 dengan judul “*A Lightweight Authentication Scheme for Power IoT Based on PUF and Chebyshev Chaotic Map*” membahas sistem autentikasi ringan untuk sistem IoT pada sektor tenaga listrik dengan menggunakan *physical unclonable function* dan *chebyshev chaotic Map*. Meski penelitian ini relevan dalam konteks keamanan autentikasi IoT, penelitian ini tidak membahas pengelolaan perangkat dan pengguna maupun sistem *backend* [31].

2.3.14 *Learned Query Optimizers: Evaluation and Improvement*

Penelitian yang dilakukan oleh Artem Mikhaylov, Nina S. Mazyavkina, Mikhail Salnikov, Ilya Trofimov, Fu Qiang, dan Evgeny Burnaev pada tahun 2022 dengan judul “*Learned Query Optimizers: Evaluation and Improvement*”. Penelitian tersebut membahas pengoptimalan query database menggunakan *learned optimizers* untuk mengoptimalkan query tradisional berbasis aturan. Topik pada penelitian ini lebih ke sisi optimasi query pada database secara umum yang biasa digunakan pada sistem *backend* untuk menulis atau membaca data pada database.

Namun, tidak membahas mengenai sistem pengelolaan perangkat maupun organisasi [32].

2.3.15 A Conceptual Modeling Approach of MQTT for IoT-Based Systems

Penelitian yang dilakukan oleh Basma M. Mohammad El-Basioni pada tahun 2024 berjudul “*A Conceptual Modeling Approach of MQTT for IoT-Based Systems*”. Tujuan utama dari penelitian ini adalah untuk menyederhanakan pemahaman spesifikasi protokol MQTT dan memfasilitasi pengembangan aplikasi IoT dengan membuat gambaran diagram untuk memahami protokol MQTT. Penelitian ini hanya membahas protokol MQTT untuk pengembang aplikasi IoT yang menggunakan protokol tersebut dan tidak membahas aspek pengelolaan perangkat IoT [33].

2.3.16 Integrating Real-Time Wireless Sensor Networks into IoT Using MQTT-SN

Penelitian yang dilakukan oleh Valentin Stangaciu dan rekan-rekannya pada tahun 2025 berjudul “*Integrating Real-Time Wireless Sensor Networks into IoT Using MQTT-SN*”. Penelitian ini mengusulkan solusi untuk mengintegrasikan jaringan sensor nirkabel *real-time* ke dalam sistem IoT menggunakan protokol MQTT-SN. Fokus utama penelitian ini adalah pada aspek komunikasi dan integrasi perangkat IoT. Meskipun relevan untuk sistem IoT dan penggunaan sensor menggunakan MQTT secara *real-time*. Namun penelitian tersebut tidak membahas mengenai manajemen perangkat IoT dan pengguna menggunakan sistem *backend* [34].

2.3.17 A Systematic Literature Review of Internet of Things for Higher Education: Architecture and Implementation

Penelitian yang dilakukan oleh Eddy Soeryanto Soegoto dan rekan-rekannya pada tahun 2022 yang berjudul “*A Systematic Literature Review of Internet of Things for Higher Education: Architecture and Implementation*”. Penelitian ini bertujuan untuk menganalisis implementasi dan arsitektur *Internet of Things* (IoT) dalam konteks pendidikan tinggi. Model arsitektur yang diusulkan terdiri dari enam lapisan utama dan satu lapisan vertikal, yaitu: objek IoT, akses, *edge computing*,

infrastruktur, cloud, dan aplikasi. Namun, penelitian ini tidak membahas mengenai manajemen perangkat, organisasi, dan otorisasi pengguna dalam sistem IoT yang mana mungkin dapat berguna dalam penerapannya di lingkungan pendidikan tinggi [35].

2.3.18 A Systematic Review of the IoT in Smart University: Model and Contribution

Penelitian yang dilakukan oleh Senny Luckyardi dengan rekan-rekannya pada tahun 2022 yang berjudul “*A Systematic Review of the IoT in Smart University: Model and Contribution*”. Penelitian ini bertujuan untuk mengidentifikasi berbagai aspek implementasi IoT di lingkungan universitas. Pemanfaatan IoT harus dengan kesiapan adopsi dari sebuah universitas. Namun, fokus penelitian ini adalah pada identifikasi aspek implementasi dan penguatan teori di balik pemanfaatan IoT di lingkungan universitas. Penelitian ini tidak membahas mengenai manajemen perangkat, organisasi, dan otorisasi pengguna dalam sistem IoT yang mana mungkin dapat berguna dalam penerapannya di lingkungan universitas [36].

2.3.19 Elliptic Curve Cryptography based Security on MQTT System for Smart Home Application

Penelitian yang dilakukan oleh Yusoff, Ishak, Rahim, dan Ali pada tahun 2022 yang berjudul “*Elliptic Curve Cryptography based Security on MQTT System for Smart Home Application*”. Penelitian ini bertujuan untuk meningkatkan keamanan sistem IoT pada aplikasi rumah pintar dengan mengintegrasikan kriptografi berbasis kurva *elliptic* ke dalam protokol MQTT. Dengan menggunakan perangkat seperti Raspberry Pi, penelitian ini mengembangkan topologi jaringan sederhana yang menghubungkan berbagai perangkat IoT di laboratorium rumah pintar. Meskipun relevan untuk sistem IoT dan penggunaan sensor menggunakan MQTT. Namun penelitian tersebut tidak membahas mengenai manajemen banyak perangkat IoT dan pengguna menggunakan sistem *backend* [37].

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

3.1.1 Waktu

Penelitian ini dilaksanakan selama enam bulan, mulai Februari 2025 hingga Oktober 2025. Adapun waktu pelaksanaan penelitian ini terjadwal pada tabel di bawah ini.

Tabel 1. Waktu Penelitian.

No.	Aktivitas	Feb	Maret	April	Mei	Juni	Juli	Agt	Sep	Okt										
		3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1
1.	Analisis kebutuhan																			
2.	Pengembangan sistem																			
3.	Pengujian sistem																			
4.	Penulisan laporan																			

3.1.2 Tempat

Lokasi penelitian mencakup Laboratorium Teknik Digital, Jurusan Teknik Elektro, Universitas Lampung dan Kelompok Pembudidaya Ikan (Pokdakan) Bintang Rosela Jaya yang berlokasi di Desa Patoman, Kecamatan Pagelaran, Kabupaten Pringsewu.

3.2 Alat dan Bahan

3.2.1 Alat

Adapun alat yang digunakan pada penelitian ini adalah sebagai berikut:

Tabel 2. Alat yang digunakan dalam penelitian.

No	Alat	Spesifikasi	Deskripsi
1.	Laptop	Prosesor AMD Ryzen 5 5500U, RAM 16 GB, SSD 512 GB, Sistem Operasi Windows 11 Home	Perangkat keras yang digunakan untuk pembuatan dan pengujian sistem.

2.	Visual Studio Code	Versi 1.98.2	Perangkat lunak untuk kode editor.
3.	Git dan GitHub	Git Versi 2.40.1.windows.1	Sistem pengelolaan versi yang digunakan untuk kolaborasi dan kontrol versi kode.
4.	Node.js	Versi 22.11.0	Lingkungan <i>runtime</i> untuk bahasa pemrograman JavaScript.
5.	NestJS	Versi 11.0.1	Kerangka kerja untuk pengembangan aplikasi sisi server berbasis Node.js.
6.	PostgreSQL	Versi 16.4	Sistem manajemen database relasional yang digunakan untuk menyimpan data.
7.	Mosquitto MQTT	Versi 3.1.1	Perangkat lunak broker MQTT yang digunakan untuk komunikasi IoT.
8.	Trello	Versi web	Perangkat lunak untuk manajemen proyek berbasis <i>agile scrum</i> .
9.	Postman	Versi 11.37.4	Perangkat lunak untuk pengujian API.
10.	VPS	CPU 1 <i>Core</i> , RAM 1 GB, Sistem Operasi Ubuntu	Server virtual yang digunakan untuk <i>deployment</i> aplikasi.
11.	Nginx	Versi 1.24.0	<i>Web server</i> yang digunakan sebagai <i>reverse proxy</i> untuk mengelola lalu lintas jaringan.

12.	Perangkat IoT	Mikrokontroler dan sensor-sensor	Perangkat untuk pengujian sistem IoT
-----	---------------	----------------------------------	--------------------------------------

3.2.2 Bahan

Bahan yang digunakan dalam penelitian dan pengembangan sistem ini terdiri dari dua jenis data utama, yaitu data dari perangkat IoT dan data dari pengguna. Data dari perangkat IoT dikirim dalam format JavaScript *Object Notation* (JSON) dan berisi parameter seperti nilai V0 dan V1, yang mewakili data sensor. Contoh format data JSON yang digunakan dapat dilihat pada gambar berikut.

```
{
  "V0": 27.5,
  "V1": 65
}
```

Gambar 4. Data JSON dari perangkat IoT.

Selain itu, sistem ini juga mengelola data pengguna, organisasi, dan perangkat yang terhubung.

3.3 Metode Penelitian

Metode yang digunakan dalam penelitian ini menerapkan prinsip-prinsip *Agile* dengan kerangka kerja *Scrum*. Kerangka kerja *Scrum* mengatur bagaimana tim bekerja secara kolaboratif untuk mencapai tujuan proyek dalam siklus iteratif yang disebut sprint.

Dalam *Scrum*, terdapat beberapa peran utama, yaitu:

1. *Product Owner* (Pemilik Produk), yaitu individu yang bertanggung jawab untuk memaksimalkan nilai produk yang dihasilkan oleh tim pengembang dengan mengelola *product backlog* dan memastikan kebutuhan pemangku kepentingan terpenuhi.

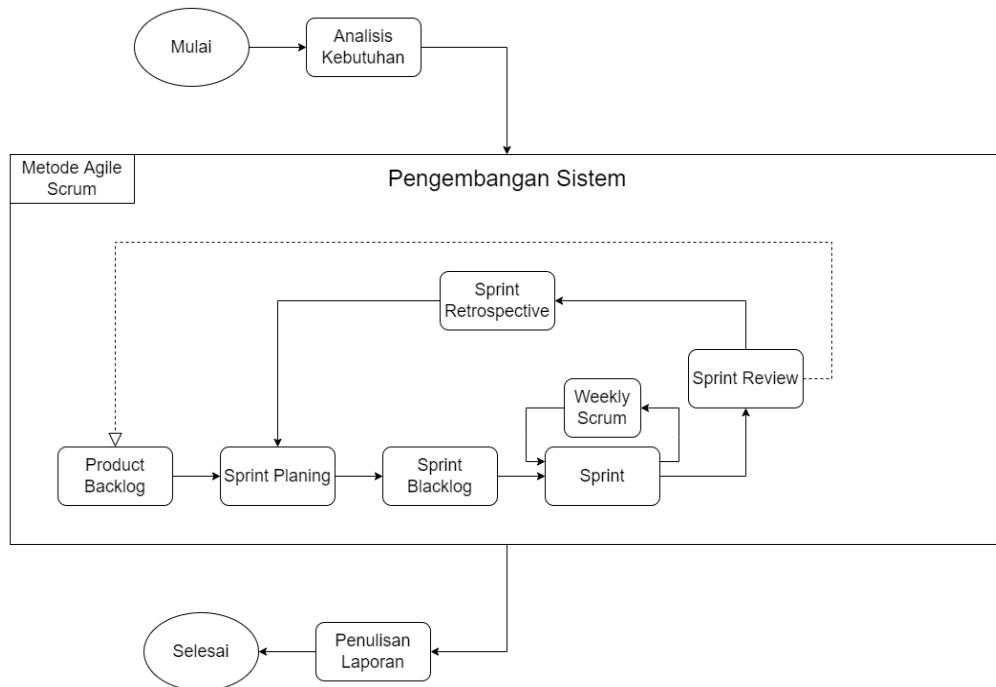
2. *Scrum Master*, yaitu individu yang bertugas membantu tim *scrum* dalam menerapkan praktik dan aturan *scrum* serta menghilangkan hambatan yang dapat menghambat produktivitas tim.
3. Development Team, yaitu kelompok pengembang yang bertanggung jawab untuk merancang, membangun, dan menguji produk yang sedang dikembangkan.

Daftar lengkap anggota tim *scrum* dalam penelitian ini dapat dilihat pada Tabel 3.3 berikut:

Tabel 3. *Scrum team*.

Peran	Nama
<i>Product Owner</i>	Wahyu Eko Sulistiono, S.T.,M.Sc.
<i>Scrum Master</i>	Halimah Mufita
<i>Development Team</i>	Halimah Mufita
	Milano Sheva Wibowo
	Bill Valentinov
	Muhammad Adil Hidayat

3.3.1 Tahapan Penelitian



Gambar 5. Alur tahapan penelitian.

Gambar di atas menunjukkan alur tahapan yang dilakukan selama penelitian ini. Sebelum melakukan pengembangan sistem, terlebih dahulu dilakukan analisis kebutuhan yang didasarkan pada kebutuhan dari *product owner* dan tim pengembang lain, seperti tim pengembang *frontend* web, tim pengembang *mobile*, dan tim pengembang IoT. Kebutuhan tersebut kemudian dimasukkan ke dalam *product backlog* sebagai daftar fitur yang akan dikembangkan.

Adapun daftar dari tim pengembangan adalah sebagai berikut :

Tabel 4. Tim pengembangan proyek.

No.	Nama	Peran
1.	Halimah Mufita	<i>Frontend Web Developer</i>
2.	Milano Sheva Wibowo	<i>Mobile Developer</i>
3.	Bill Valentinov	<i>Backend Developer</i>
4.	Muhammad Adil Hidayat	<i>IoT Developer</i>

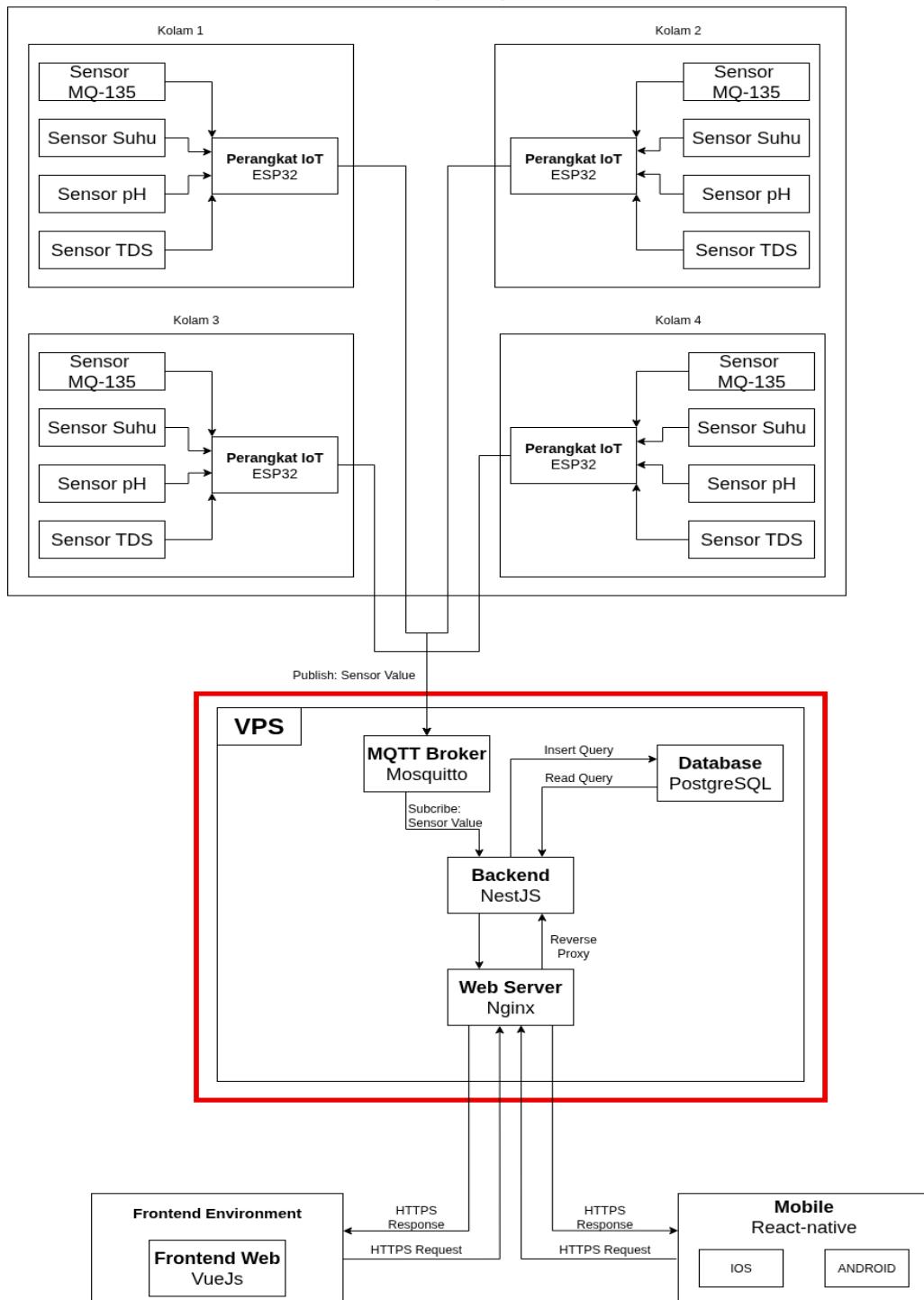
Pengembangan sistem menggunakan metode *agile scrum*, yang terdiri dari beberapa tahapan utama, yaitu *product backlog*, *sprint planning*, *sprint backlog*, *sprint*, *weekly scrum*, *sprint review*, dan *sprint retrospective*. Pada tahap *sprint review*, jika terdapat tambahan atau umpan balik dari *product owner*, maka proses akan kembali ke *sprint planning* untuk dilakukan perbaikan atau penyesuaian. Jika tidak ada perubahan lebih lanjut, maka pengembangan dianggap selesai dan tim dapat melanjutkan pengerjaan item berikutnya dalam *product backlog*.

3.3.2 Tahapan Analisis Kebutuhan

Pada penelitian ini, sebelum melakukan pengembangan sistem, terlebih dahulu dilakukan analisis kebutuhan untuk mengidentifikasi aspek-aspek yang diperlukan dalam proses pengembangan. Analisis kebutuhan dilakukan melalui diskusi dengan *product owner* dan tim pengembang lainnya guna memahami fitur serta fungsionalitas yang dibutuhkan. Hasil dari diskusi tersebut dijadikan acuan dalam

penyusunan *product backlog*, yang berisi daftar fitur dan fungsi yang akan dikembangkan selama proses pengembangan sistem.

Berdasarkan studi kasus dalam penelitian ini, berikut merupakan diagram keseluruhan sistem yang akan dikembangkan:

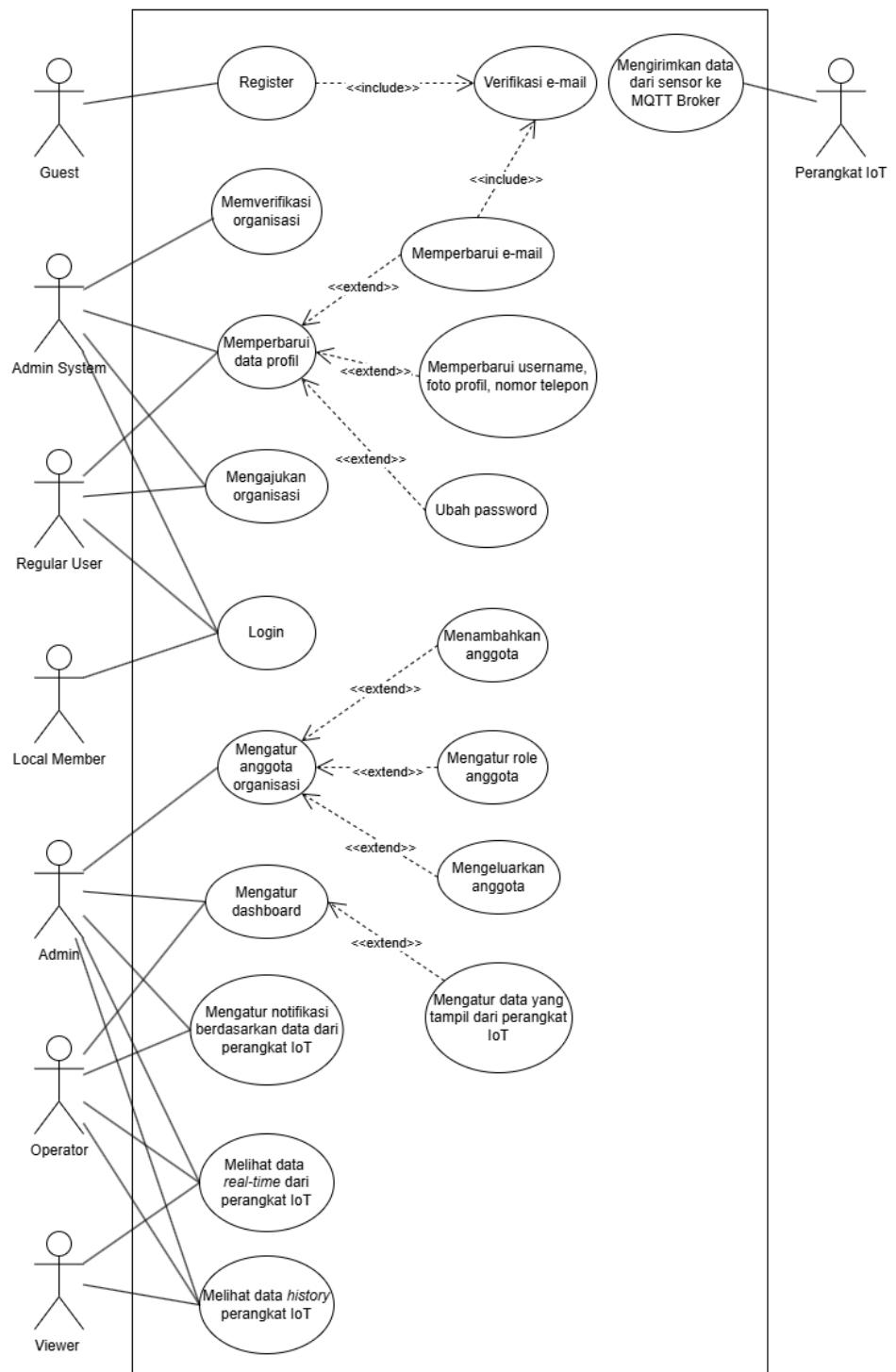


Gambar 6. Diagram keseluruhan pengembangan sistem.

Berdasarkan diagram tersebut, setiap kolam akan dipasangi perangkat IoT yang dilengkapi dengan empat sensor, yaitu sensor MQ-135, sensor suhu, sensor pH, dan sensor TDS. Perangkat IoT tersebut akan mengirimkan data sensor dalam

format JSON ke broker MQTT dengan mekanisme *publish* pada topik yang berisi kode autentikasi. Selanjutnya, *backend* melakukan *subscribe* pada topik yang sama dengan topik yang *publish* dari sensor perangkat IoT untuk menerima data dari perangkat IoT. *Backend* juga berfungsi untuk memasukkan dan membaca data pada database, melayani permintaan HTTPS dari *frontend* web maupun *mobile* melalui *reverse proxy* pada Nginx. Kemudian, terdapat database yang menyimpan data dari perangkat IoT maupun pengguna melalui *backend*. Berikutnya, terdapat *web server* yang akan mengelola protokol HTTPS dan meneruskannya pada *backend*. Penelitian ini difokuskan pada pengembangan aplikasi *backend*, serta *deployment* beberapa komponen sistem pada VPS, seperti broker MQTT, database, dan *web server*.

Kemudian, terdapat analisis kebutuhan fungsionalitas yang bertujuan untuk menggambarkan interaksi antara aktor dan sistem, serta mendefinisikan fitur-fitur yang akan dibangun. Berdasarkan hasil analisis tersebut, disusun diagram *use case* yang merepresentasikan fungsi-fungsi pada sistem yang dirancang. Diagram *use case* ini juga menjadi acuan dalam penyusunan diagram aktivitas (*activity diagram*) yang menggambarkan alur interaksi antara keseluruhan komponen sistem. Berikut ini merupakan *use case* diagram aplikasi yang merepresentasikan fungsi-fungsi pada aplikasi:



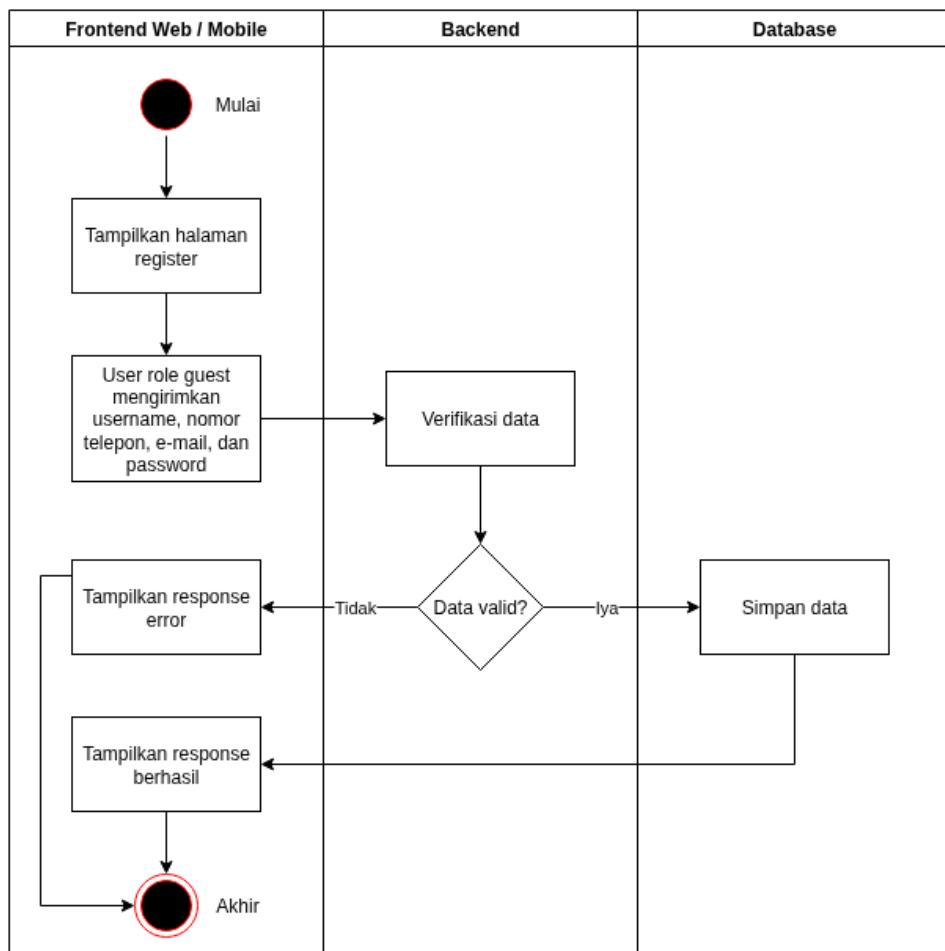
Gambar 7. *Use case* diagram aplikasi.

Berdasarkan diagram tersebut, terdapat beberapa peran (aktor) yang memiliki tanggung jawab dan fungsionalitas berbeda:

- **Guest** merupakan pengguna yang belum terdaftar dalam sistem. Pengguna dengan status ini hanya memiliki akses untuk melakukan proses registrasi agar memperoleh akun dan menjadi *regular user*.
- **Admin System** merupakan pengguna yang bertanggung jawab untuk memverifikasi organisasi yang diajukan oleh *regular user*.
- **Regular User** merupakan pengguna yang telah melakukan registrasi dan memiliki akses untuk *login* ke dalam sistem serta dapat mengajukan pembuatan organisasi.
- **Local Member** adalah pengguna yang dibuat secara langsung oleh Admin Organisasi dan hanya dapat berperan sebagai *viewer* di dalam organisasi tempat ia didaftarkan. Pengguna ini tidak melalui proses registrasi secara mandiri.
- **Admin** adalah pengguna yang memiliki wewenang penuh terhadap pengelolaan organisasi. Admin dapat mengelola anggota organisasi seperti menambahkan anggota, mengatur peran setiap anggota, dan mengeluarkan anggota. Selain itu, admin juga dapat mengatur perangkat, data yang ditampilkan, dan tampilan *dashboard*. Peran sebagai admin ini berasal dari *regular user* yang membuat organisasi atau anggota organisasi yang diberikan peran menjadi admin oleh admin yang lainnya.
- **Operator** merupakan anggota organisasi yang memiliki hak akses untuk mengelola *dashboard* serta menentukan data yang akan ditampilkan kepada pengguna. Namun, peran ini tidak memiliki hak untuk mengelola anggota organisasi. Peran sebagai *operator* ini berasal dari admin organisasi.
- **Viewer** merupakan anggota organisasi yang hanya memiliki hak akses untuk melihat data yang ditampilkan pada *dashboard* dan melihat data *history* perangkat IoT. Peran sebagai *viewer* ini berasal dari admin organisasi.
- **Perangkat IoT** adalah pihak yang mengirimkan data ke broker MQTT.

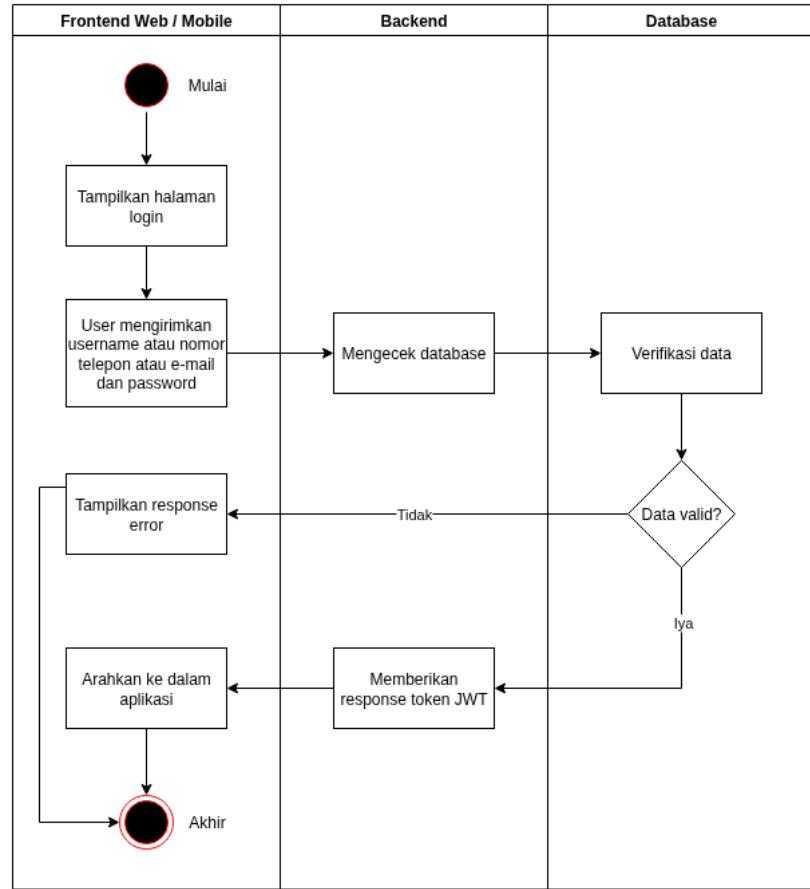
Berdasarkan *use case* diagram tersebut maka dibuat *activity* diagram untuk alur interaksi antara keseluruhan sistem. Berikut ini adalah beberapa *activity* diagram berdasarkan *use case* diagram:

a. Registrasi



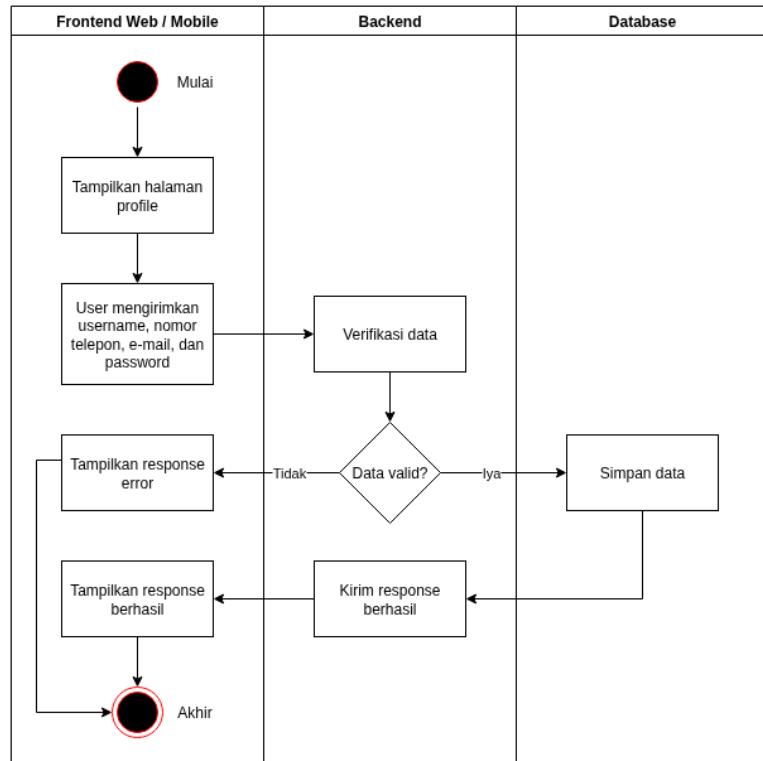
Gambar 8. *Activity* diagram registrasi.

b. *Login*



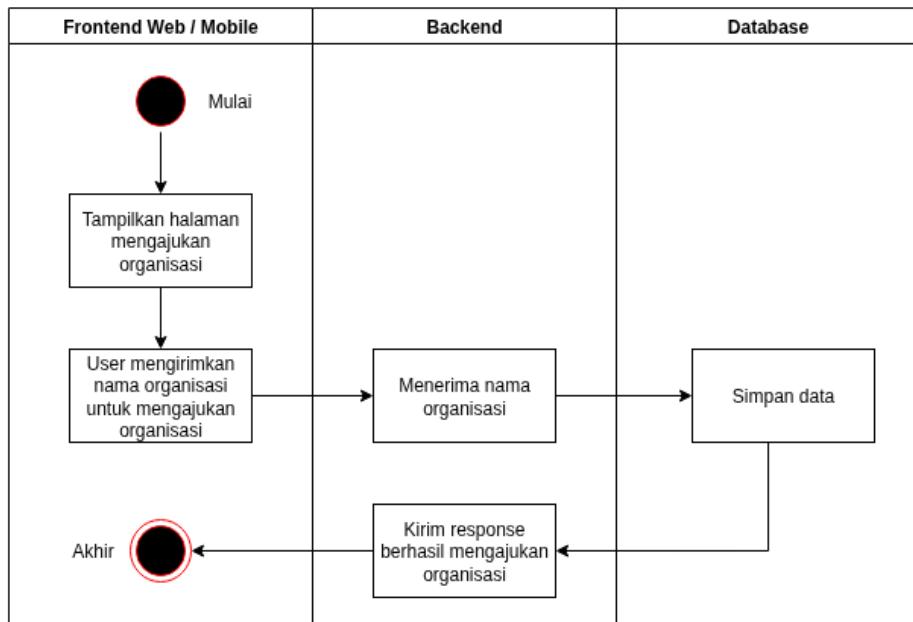
Gambar 9. *Activity diagram login.*

c. Memperbarui data profil



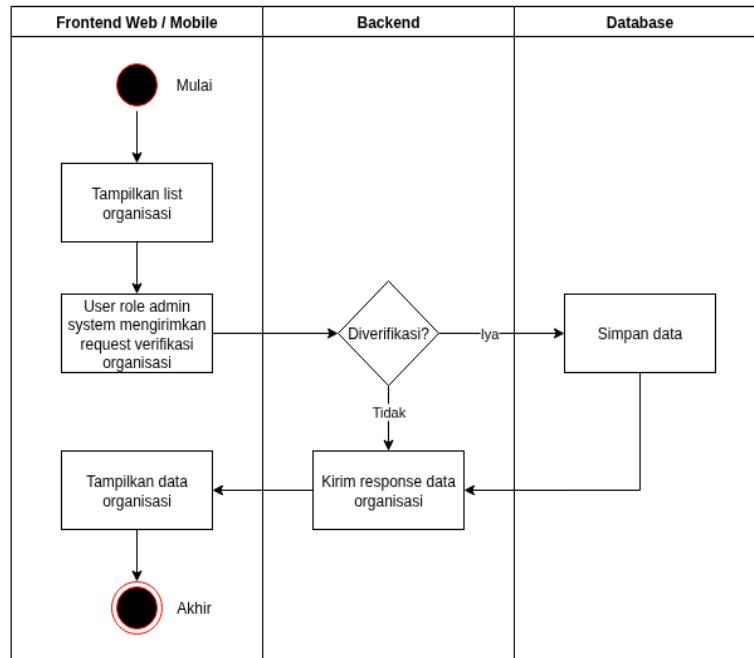
Gambar 10. *Activity diagram memperbarui data profil.*

d. Mengajukan organisasi



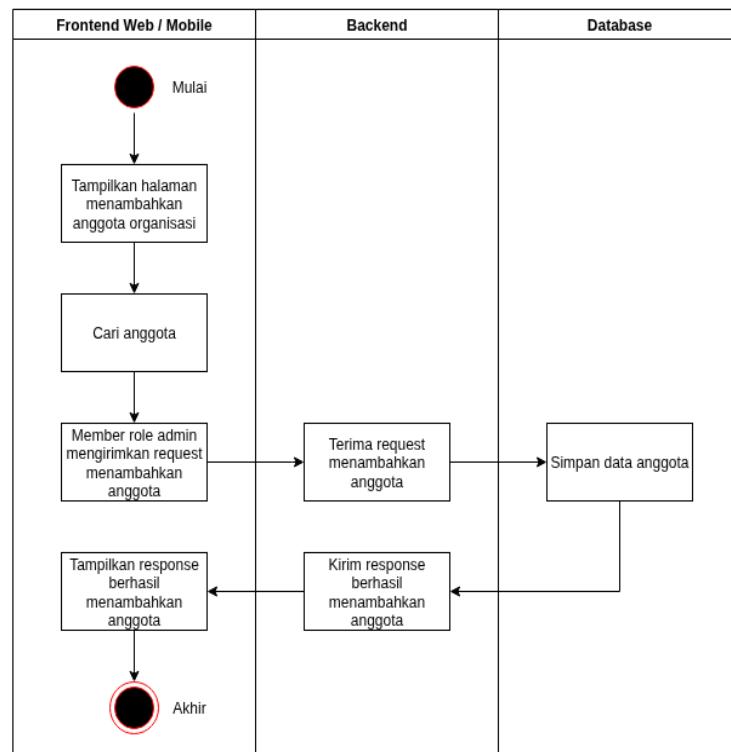
Gambar 11. *Activity diagram mengajukan organisasi.*

e. Memverifikasi organisasi



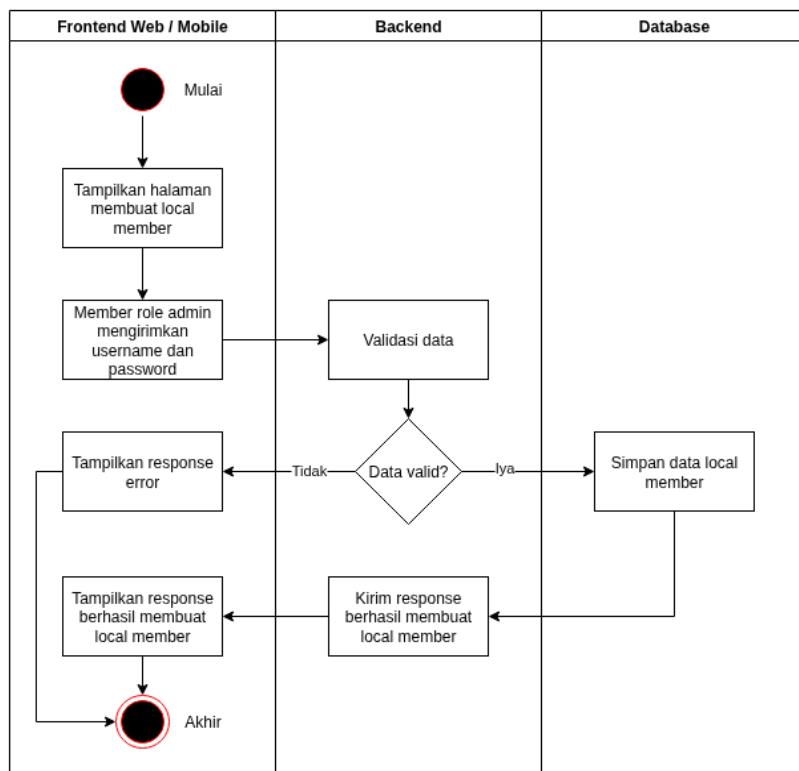
Gambar 12. *Activity* diagram memverifikasi organisasi.

f. Menambahkan anggota organisasi



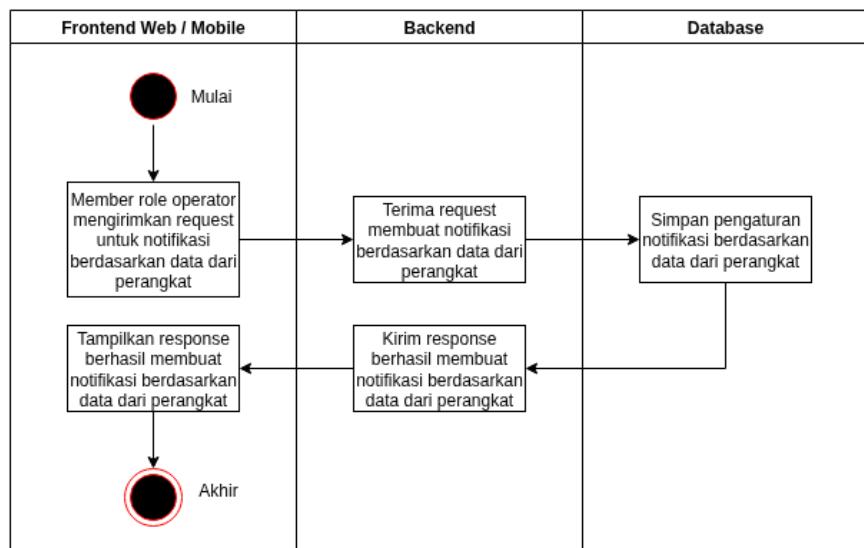
Gambar 13. *Activity* diagram menambahkan anggota organisasi.

g. Membuat *Local Member*



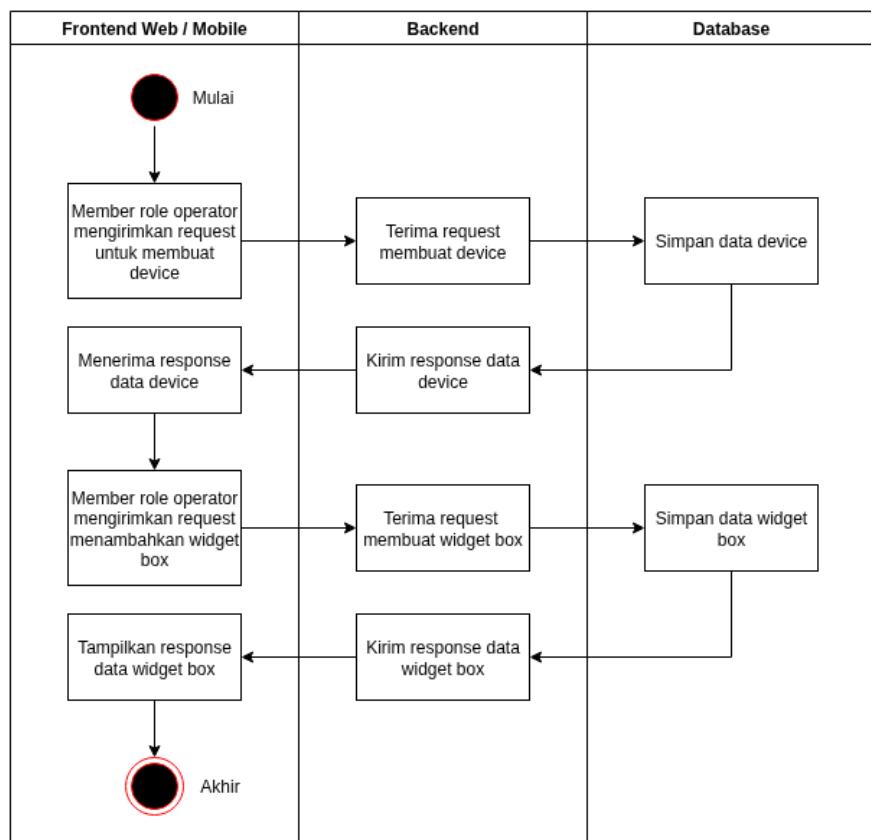
Gambar 14. *Activity diagram membuat local member.*

h. Mengatur notifikasi berdasarkan data dari perangkat



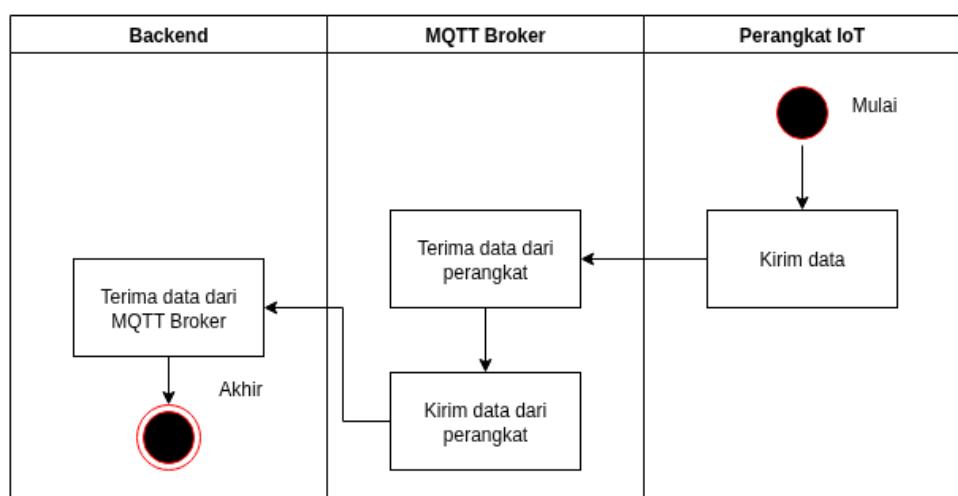
Gambar 15. *Activity diagram mengatur notifikasi berdasarkan data dari perangkat.*

i. Mengatur dashboard



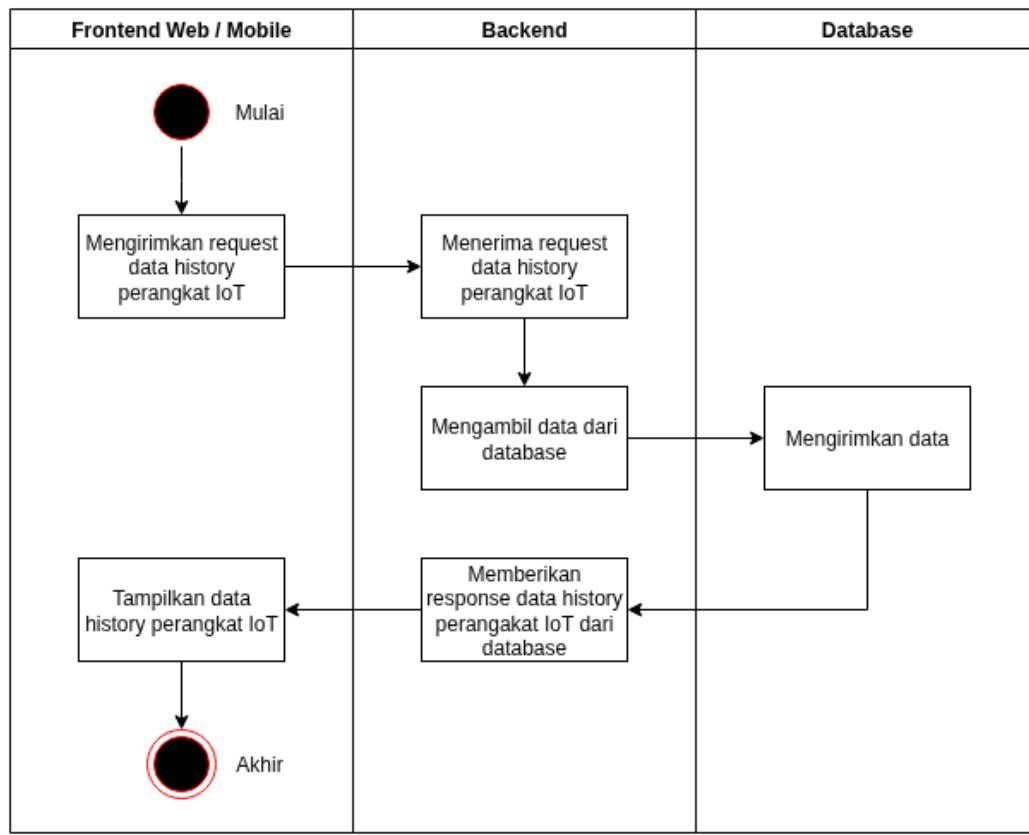
Gambar 16. *Activity* diagram mengatur dashboard.

j. Mengirimkan data dari sensor



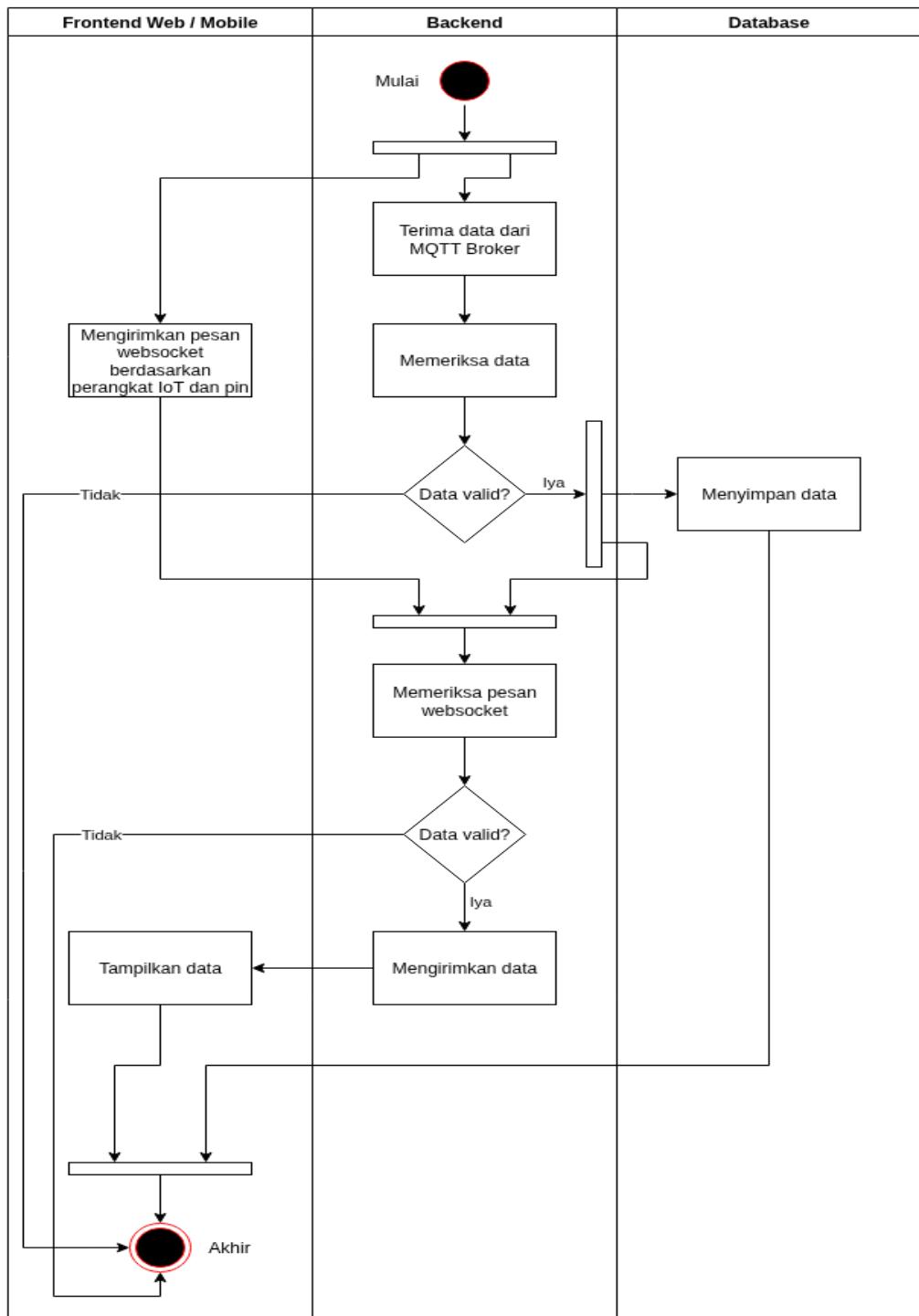
Gambar 17. *Activity* diagram mengirimkan data dari sensor.

k. Melihat data *history* perangkat IoT



Gambar 18. *Activity* diagram melihat data *history* perangkat IoT.

1. Melihat data *real-time* dari perangkat IoT



Gambar 19. *Activity diagram* melihat data dari perangkat.

Analisis yang dilakukan pada tahap ini akan menjadi rujukan dalam tahap pengembangan sistem.

3.3.3 Tahapan Pengembangan Sistem

Pada tahapan ini, sistem dikembangkan berdasarkan hasil analisis kebutuhan yang telah dilakukan sebelumnya. Pengembangan sistem menerapkan metode *agile scrum*, yang memungkinkan proses pengembangan dilakukan secara iteratif dan adaptif. Setiap fitur yang telah dirancang dalam *product backlog* akan diimplementasikan dalam siklus sprint, yang terdiri dari beberapa tahapan, yaitu *product backlog*, *sprint planning*, *sprint backlog*, *sprint*, *weekly scrum*, *sprint review*, dan *sprint retrospective*.

3.4.2.1 Product Backlog

Product backlog merupakan daftar fitur dan fungsi yang akan dikembangkan selama proses pengembangan sistem. Daftar ini bersifat dinamis dan dapat mengalami perubahan apabila terdapat penyesuaian kebutuhan dari *product owner* maupun tim pengembang lainnya. *Product owner* bertanggung jawab atas pengelolaan *product backlog* guna memastikan bahwa fitur yang dikembangkan sesuai dengan tujuan sistem. *Product backlog* ini dicatat dan dikelola menggunakan aplikasi Trello.

3.4.2.2 Sprint Planning

Sprint planning merupakan tahap perencanaan sprint yang dipimpin oleh *product owner*. Pada tahap ini, ditentukan daftar *product backlog* yang akan dikerjakan dalam satu sprint, sasaran yang harus dicapai dalam setiap *backlog*, serta waktu penyelesaian sprint yang umumnya berlangsung selama dua minggu. Seluruh anggota *scrum team* akan berkumpul dan berdiskusi untuk menyepakati tugas-tugas yang akan dikerjakan selama *sprint* pada waktu dan tempat yang telah ditentukan.

3.4.2.3 Sprint Backlog

Sprint backlog merupakan sekumpulan *product backlog* yang telah dipilih untuk dikerjakan dalam satu sprint. Setiap anggota tim pengembang akan bertanggung

jawab terhadap satu atau beberapa *sprint backlog* yang telah ditentukan selama *sprint planning*.

3.4.2.4 Sprint

Sprint merupakan fase pelaksanaan di mana tim pengembang mengimplementasikan fitur berdasarkan *sprint backlog* yang telah ditentukan. Tim pengembang harus menyelesaikan tugasnya dalam periode sprint yang telah disepakati tanpa melakukan perubahan besar. *sprint backlog* yang sedang dikerjakan akan dipindahkan ke daftar *doing* pada aplikasi Trello sebagai tanda bahwa tugas tersebut sedang dalam proses pengerjaan.

3.4.2.5 Weekly Scrum

Weekly Scrum adalah pertemuan mingguan yang dipimpin oleh *scrum master* dan dilakukan oleh seluruh tim pengembang, baik secara daring maupun tatap muka. Dalam sesi ini, setiap anggota tim melaporkan perkembangan pekerjaannya, mengidentifikasi kendala yang dihadapi, serta berkoordinasi untuk memastikan bahwa semua anggota tim dapat mencapai *sprint backlog* yang telah ditetapkan.

3.4.2.6 Sprint Review

Sprint review merupakan pertemuan yang dihadiri oleh seluruh anggota *scrum team* untuk meninjau hasil kerja selama satu sprint. *Sprint review* dilakukan setelah sprint selesai dan dipimpin oleh *product owner*. Pada tahap ini, *Product owner* akan mengevaluasi *sprint backlog* yang telah diselesaikan, memberikan umpan balik, serta menentukan apakah terdapat perubahan atau tambahan yang perlu dilakukan. Jika ada revisi atau penyesuaian, proses akan dilanjutkan ke *sprint retrospective*, sedangkan jika *backlog* telah disetujui, maka akan dilakukan *sprint planning* untuk menentukan *backlog* untuk *sprint* berikutnya.

3.4.2.7 Sprint Retrospective

Sprint retrospective merupakan tahap evaluasi terhadap proses yang telah dilakukan dalam sprint sebelumnya. Pada tahap ini, tim pengembang dan *product owner* akan mendiskusikan hambatan yang dihadapi, perbaikan yang dapat

dilakukan, serta strategi peningkatan efisiensi dalam sprint berikutnya. Masukan yang diperoleh akan menjadi dasar untuk iterasi selanjutnya dan *sprint planning* berikutnya.

3.3.4 Tahapan Penulisan Laporan

Tahapan penulisan laporan dilakukan setelah seluruh proses pengembangan sistem selesai. Pada tahap ini, dilakukan dokumentasi secara sistematis berdasarkan seluruh proses yang telah dilakukan, mulai dari analisis kebutuhan hingga pengembangan sistem. Laporan ini akan menjadi dokumentasi resmi dari hasil penelitian yang dilakukan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa:

1. Sistem *backend* berhasil dikembangkan dengan menggunakan metode *agile scrum*, yang terdiri dari 11 sprint dan mencakup total 14 *product backlog item*.
2. Berdasarkan hasil pengujian menggunakan Postman pada *product backlog* autentikasi, manajemen organisasi, manajemen perangkat, serta visualisasi data *real-time* menunjukkan bahwa sistem dapat menerima, menyimpan, dan mengirimkan data dari perangkat IoT untuk ditampilkan secara langsung pada aplikasi *frontend web* dan *mobile*.
3. Berdasarkan hasil dari *product backlog* autentikasi, manajemen organisasi, dan manajemen perangkat, sistem memiliki fitur untuk mengelola pengguna, organisasi, dan perangkat yang terhubung.
4. Berdasarkan hasil dari *product backlog* autentikasi, manajemen organisasi, dan manajemen perangkat, sistem mampu memastikan bahwa hanya pengguna dan perangkat yang telah diverifikasi dan memiliki otorisasi yang dapat mengakses sistem, menggunakan *middleware guard* dan penggunaan kode autentikasi sebagai topik pada komunikasi MQTT.
5. Berdasarkan hasil pengujian sistem menggunakan metode *stress testing* pada lingkungan VPS, sistem dapat menerima maksimal 400 pengguna aktif, dengan demikian sistem mampu beroperasi secara stabil hingga batas tersebut namun mulai mengalami penurunan kinerja dan munculnya *error* ketika beban melampaui angka tersebut. Dengan perbandingan pengujian pada laptop, dapat disimpulkan bahwa sistem membutuhkan sumber daya yang lebih tinggi dari spesifikasi VPS agar dapat tetap stabil pada jumlah pengguna yang lebih besar.

5.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem di masa mendatang antara lain:

1. Mengimplementasikan *access token* dan *refresh token* sebagai mekanisme autentikasi guna meningkatkan keamanan proses *login* dan menjaga sesi pengguna tetap valid.
2. Menambahkan fitur *caching* menggunakan Redis untuk meningkatkan performa sistem, khususnya dalam hal waktu respons (*response time*).
3. Mempertimbangkan penggunaan database non-SQL, seperti MongoDB dan InfluxDB, untuk penyimpanan data abstrak dari perangkat IoT yang bersifat tidak terstruktur atau semi-terstruktur.
4. Menggunakan *runtime* Bun sebagai alternatif pengganti Node.js untuk meningkatkan efisiensi eksekusi kode JavaScript dan mempercepat waktu *build* maupun waktu *respons* server.

DAFTAR PUSTAKA

- [1] H. Chaouchi and T. Bourgeau, “Internet of Things: Building the New Digital Society,” *IoT*, vol. 1, no. 1, pp. 1–4, June 2018, doi: 10.3390/iot1010001.
- [2] International Telecommunication Union (ITU), “Overview of the Internet of Things,” ITU Telecommunication Standardization Sector (ITU-T), Recommendation ITU-T Y.2060, 2012.
- [3] F. Javed, M. K. Afzal, M. Sharif, and B.-S. Kim, “Internet of Things (IoT) Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review,” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 3, pp. 2062–2100, 2018, doi: 10.1109/COMST.2018.2817685.
- [4] M. Fahmideh and D. Zowghi, “An Exploration of IoT Platform Development,” *Information Systems*, vol. 87, p. 101409, Jan. 2020, doi: 10.1016/j.is.2019.06.005.
- [5] S. Tilkov and S. Vinoski, “Node.js: Using JavaScript to Build High-Performance Network Programs,” *IEEE Internet Comput.*, vol. 14, no. 6, pp. 80–83, Nov. 2010, doi: 10.1109/MIC.2010.145.
- [6] X. Huang, “Research and Application of Node.js Core Technology,” in *2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, Sanya, China: IEEE, Dec. 2020, pp. 1–4. doi: 10.1109/ICHCI51889.2020.00008.
- [7] A. Muhamad Saepuloh and S. Ginting, “Perancangan Sistem Informasi Manajemen Proyek dengan Menggunakan Software Nest.js Berbasis Web di PT. Mitra Pajakku,” *INFOKOM*, vol. 10, no. 1, pp. 1–9, Nov. 2022, doi: 10.56689/infokom.v10i1.818.
- [8] C. M. Sowandi, I. Leonita, S. Hartanto, P. Arisaputra, and D. David, “Exploring the Effectiveness, Features, and Compatibility of MongoDB and MySQL: A Comprehensive Comparison of NoSQL and Relational Databases,” *MIND (Multimedia Artificial Intelligent Networking Database) Journal*, vol. 8, no. 2, pp. 217–229, Dec. 2023, doi: 10.26760/mindjournal.v1i1.49.

- [9] S. V. Salunke and A. Ouda, “A Performance Benchmark for the PostgreSQL and MySQL Databases,” *Future Internet*, vol. 16, no. 10, p. 382, Oct. 2024, doi: 10.3390/fi16100382.
- [10] M. Stonebraker, L. A. Rowe, and M. Hirohama, “The Implementation of POSTGRES,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 2, no. 1, pp. 125–142, Mar. 1990, doi: 10.1109/69.50912.
- [11] K. Lima, T. D. Oyetoyan, R. Heldal, and W. Hasselbring, “Evaluation of MQTT Bridge Architectures in a Cross-Organizational Context,” 2025, *arXiv*. doi: 10.48550/ARXIV.2501.14890.
- [12] S. Pawar, D. B. Jadhav, M. Lokhande, P. Raskar, and M. Patil, “Evaluation of quality of service parameters for MQTT communication in IoT application by using deep neural network,” *Int. j. inf. tecnol.*, vol. 16, no. 2, pp. 1123–1136, Feb. 2024, doi: 10.1007/s41870-023-01664-2.
- [13] X. Liu, T. Zhang, N. Hu, P. Zhang, and Y. Zhang, “The method of Internet of Things access and network communication based on MQTT,” *Computer Communications*, vol. 153, pp. 169–176, Mar. 2020, doi: 10.1016/j.comcom.2020.01.044.
- [14] Prakash P, Biju R, and M. Kamath, “Performance analysis of process driven and event driven web servers,” in *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, India: IEEE, Jan. 2015, pp. 1–7. doi: 10.1109/ISCO.2015.7282230.
- [15] M. Squire, “Forge++: The Changing Landscape of FLOSS Development,” in *2014 47th Hawaii International Conference on System Sciences*, Waikoloa, HI: IEEE, Jan. 2014, pp. 3266–3275. doi: 10.1109/HICSS.2014.405.
- [16] C. Gea, K. J. D. Lase, and M. Syamsudin, “Implementasi Virtual Private Server untuk Mini Hosting,” *JURNAL INFAC*, vol. 7, no. 01, pp. 5–9, Jan. 2023, doi: 10.61179/jurnalinfact.v7i01.402.
- [17] A. Srivastava, S. Bhardwaj, and S. Saraswat, “SCRUM model for agile methodology,” in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida: IEEE, May 2017, pp. 864–869. doi: 10.1109/ICCAA.2017.8229928.

- [18] I. Tahyudin and Zidni Iman Sholihat, “Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy,” *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 6, no. 1, pp. 169–176, Feb. 2022, doi: 10.29207/resti.v6i1.3793.
- [19] A. Nurrahman and N. AZ, “Sistem Kontrol Suhu dengan PID dan Monitoring Daya Output pada Panel Photovoltaic Portable dengan Virtual Private Server,” *MAESTRO*, vol. 3, no. 2, Oct. 2020.
- [20] S. Sauda and M. Barokah, “Penerapan NodeJS dan PostgreSQL sebagai Backend pada Aplikasi Ecommerce Localla,” *INFOTECH J*, vol. 8, no. 2, pp. 101–105, Nov. 2022, doi: 10.31949/infotech.v8i2.2944.
- [21] A. Andipradana and K. Dwi Hartomo, “Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum,” *Jurnal Algoritma*, vol. 18, no. 1, pp. 161–172, Aug. 2021, doi: 10.33364/algoritma/v.18-1.869.
- [22] K. Jonathan, “Rancang Bangun Dashboard IoT untuk Monitoring Kolam Akuaponik Berbasis Android,” Skripsi, Universitas Dinamika, Surabaya, 2023.
- [23] I. Kurniawan, F. Rozi, and Humaira, “REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android,” *Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 1, no. 4, pp. 127–132, 2020, doi: 10.62527/jitsi.1.4.18.
- [24] A. Mubariz, D. Nur, E. Tungadi, and M. N. Y. Utomo, “Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node.JS (Studi Kasus: Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang),” *Prosiding Seminar Nasional Teknik Elektro dan Informatika (SNTEI)*, Oct. 2020.
- [25] J. Lumban Gaol *et al.*, “Aplikasi Android untuk Monitoring Lahan Pertanian secara Realtime Berbasis Internet of Things,” *JuTISI*, vol. 6, no. 3, Dec. 2020, doi: 10.28932/jutisi.v6i3.3039.
- [26] B. Setiawan, S. Styawati, and S. Alim, “Implementasi Sistem IoT Pada Akuakultur Dan Hydroponik (Akuaponik) Modern Untuk Pertumbuhan

- Ikan Nila,” *JPIT*, vol. 9, no. 1, pp. 47–53, Jan. 2024, doi: 10.30591/jpit.v9i1.5896.
- [27] F. Oliveira, D. G. Costa, L. Lima, and I. Silva, “iBikeSafe: A Multi-Parameter System for Monitoring, Evaluation and Visualization of Cycling Paths in Smart Cities Targeted at Cycling Adverse Conditions,” *Smart Cities*, vol. 4, no. 3, pp. 1058–1086, July 2021, doi: 10.3390/smartcities4030056.
- [28] S. Senthilkumar, K. Brindha, and S. Bhandari, “Vehicle accident management and control system using MQTT,” *IJAAS*, vol. 9, no. 1, p. 1, Mar. 2020, doi: 10.11591/ijaas.v9.i1.pp1-11.
- [29] Y. Dong and Y.-D. Yao, “IoT Platform for COVID-19 Prevention and Control: A Survey,” *IEEE Access*, vol. 9, pp. 49929–49941, 2021, doi: 10.1109/ACCESS.2021.3068276.
- [30] M. Soula, B. Mbarek, and A. Meddeb, “A Real-Time Trust Management Model Using Digital Twin in IoT Networks,” *IEEE Access*, vol. 12, pp. 183326–183343, 2024, doi: 10.1109/ACCESS.2024.3506870.
- [31] X. Jin, N. Lin, Z. Li, W. Jiang, Y. Jia, and Q. Li, “A Lightweight Authentication Scheme for Power IoT Based on PUF and Chebyshev Chaotic Map,” *IEEE Access*, vol. 12, pp. 83692–83706, 2024, doi: 10.1109/ACCESS.2024.3413853.
- [32] A. Mikhaylov, N. S. Mazyavkina, M. Salnikov, I. Trofimov, F. Qiang, and E. Burnaev, “Learned Query Optimizers: Evaluation and Improvement,” *IEEE Access*, vol. 10, pp. 75205–75218, 2022, doi: 10.1109/ACCESS.2022.3190376.
- [33] B. M. M. El-Basioni, “A Conceptual Modeling Approach of MQTT for IoT-Based Systems,” *Journal of Electrical Systems and Inf Technol*, vol. 11, no. 1, p. 62, Dec. 2024, doi: 10.1186/s43067-024-00181-x.
- [34] V. Stangaciu, C. Stangaciu, B. Gusita, and D.-I. Curiac, “Integrating Real-Time Wireless Sensor Networks into IoT Using MQTT-SN,” *J Netw Syst Manage*, vol. 33, no. 2, p. 37, Apr. 2025, doi: 10.1007/s10922-025-09916-1.

- [35] E. S. Soegoto *et al.*, “A systematic Literature Review of Internet of Things for Higher Education: Architecture and Implementation,” *Indonesian J. Sci. Technol.*, vol. 7, no. 3, pp. 511–528, Oct. 2022, doi: 10.17509/ijost.v7i3.51464.
- [36] S. Luckyardi, R. Hurriyati, D. Disman, and P. D. Dirgantari, “A Systematic Review of the IoT in Smart University: Model and Contribution,” *Indonesian J. Sci. Technol.*, vol. 7, no. 3, pp. 529–550, Oct. 2022, doi: 10.17509/ijost.v7i3.51476.
- [37] Z. Y. M. Yusoff, M. K. Ishak, L. A. B. Rahim, and O. Ali, “Elliptic Curve Cryptography based Security on MQTT System for Smart Home Application,” in *2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Prachuap Khiri Khan, Thailand: IEEE, May 2022, pp. 1–4. doi: 10.1109/ECTI-CON54298.2022.9795478.
- [38] M. Bolanowski, M. Ćmil, and A. Starzec, “New Model for Defining and Implementing Performance Tests,” *Future Internet*, vol. 16, no. 10, p. 366, Oct. 2024, doi: 10.3390/fi16100366.