

**SISTEM PEMANTAUAN CERDAS PADA KETERSEDIAAN
RUANG KELAS BERBASIS IOT PADA FAKULTAS SAINS DAN
TEKNOLOGI UIN AR-RANIRY BANDA ACEH**

TUGAS AKHIR

Diajukan Oleh:

ZABRINA DEWI

210705011

Mahasiswa Fakultas Sains dan Teknologi

Program Studi Teknologi Informasi



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
1448 H/ 2026 M**

LEMBAR PERSETUJUAN

SISTEM PEMANTAUAN CERDAS PADA KETERSEDIAAN RUANG KELAS BERBASIS IOT PADA FAKULTAS SAINS DAN TEKNOLOGI UIN AR-RANIRY BANDA ACEH

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai salah satu Beban Studi Memperoleh Gelar Sarjana (S1)
Dalam Ilmu/Prodi Teknologi Informasi

Oleh:

ZABRINA DEWI

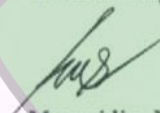
210705011

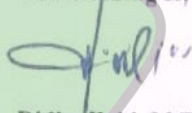
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui untuk Dimunagasyahkan oleh:

Pembimbing I,

Pembimbing II,


Mursyidin, M.T


Ridha Ilahi, M.T

NIP.198204052023211020

UIN AR-RANIRY

NIP.197905302014031001

Mengetahui,

Ketua Program Studi,


Malahayati, M.T

NIP.198301272015032003

LEMBAR PENGESAHAN

SISTEM PEMANTAUAN CERDAS PADA KETERSEDIAAN RUANG KELAS BERBASIS IOT PADA FAKULTAS SAINS DAN TEKNOLOGI UIN AR-RANIRY BANDA ACEH

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Jum'at, 03 Juli 2026
18 Muharam 1448 H
Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,

Mursvidin, M.T
NIP.198204052023211020

Sekretaris,

Ridha Ilahi, M.T
NIP.197905302014031001

Penguji I,

Malahayati, M.T
NIP.198301272015032003

Penguji II,

Dr. Hendri Ahmadian, M.I.M
NIP.198301042014031002

Mengetahui:

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh



Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIP.196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Zabrina Dewi
NIM : 210705011
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul Tugas Akhir : Sistem Pemantauan Cerdas pada Ketersediaan Ruang
Kelas Berbasis IoT pada Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 9 Juli 2026

Yang Menyatakan


(Zabrina Dewi)



ABSTRAK

Nama : Zabrina Dewi
Nim : 210705011
Program Studi : Teknologi Informasi
Judul : Sistem Pemantauan Cerdas pada Ketersediaan Ruang Kelas Berbasis IoT pada Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh
Tanggal Sidang : 03 Juli 2026
Jumlah Halaman : 68
Pembimbing I : Mursyidin, M.T
Pembimbing II : Ridha Ilahi, M.T
Kata Kunci : Internet of Things (IoT), Sensor PIR, VL53LOX Pemantauan Ruang Kelas

Pengelolaan ruang kelas di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh masih menghadapi kendala karena perubahan jadwal, perpindahan ruang, dan belum tersedianya sistem pemantauan secara real-time. Penelitian ini bertujuan merancang sistem pemantauan ketersediaan ruang kelas berbasis Internet of Things (IoT) agar informasi penggunaan ruang dapat diperoleh dengan cepat dan akurat.

Sistem dibangun menggunakan mikrokontroler ESP32, sensor VL53L0X untuk menghitung pengguna yang masuk dan keluar ruangan, serta sensor PIR untuk mendeteksi keberadaan dosen dan mahasiswa. Data hasil pembacaan sensor ditampilkan melalui aplikasi Blynk dan disimpan pada Google Spreadsheet sebagai riwayat penggunaan ruang kelas.

Hasil penelitian menunjukkan bahwa sistem mampu mendeteksi aktivitas pengguna, menentukan status ruang kelas, dan menampilkan informasi secara real-time. Sistem ini diharapkan dapat membantu pihak fakultas dalam mengelola penggunaan ruang kelas secara lebih efektif, efisien, dan terdokumentasi dengan baik.

ABSTRACT

Name : Zabrina Dewi
Student ID : 210705011
Study Program : : Information Technology
Study Program : Smart Monitoring System for IoT Based Classroom Availability at the Faculty of Science and Technology UIN Ar-niry Banda Aceh
Thesis Defense Date : 03 July 2026
Thesis Page : 68
Supervisor I : Mursyidin, M.T
Supervisor II : Ridha Ilahi, M.T
Keyword : Internet of Things (IoT), ESP32, PIR Sensor, VL53LOX Classroom Monitoring

Classroom management at the Faculty of Science and Technology, UIN Ar-Raniry Banda Aceh still faces challenges due to schedule changes, classroom reassignments, and the absence of a real-time monitoring system. This study aims to develop an Internet of Things (IoT) based classroom monitoring system to provide accurate and up-to-date classroom availability information.

The system is built using an ESP32 microcontroller, VL53L0X sensors to count people entering and leaving the classroom, and PIR sensors to detect the presence of lecturers and students. The collected data are displayed through the Blynk application and stored in Google Spreadsheet as historical records.

The results show that the system can detect user activities, determine classroom status, and provide real-time monitoring information. The proposed system is expected to improve the effectiveness and efficiency of classroom management while providing better documentation of classroom usage.

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah Swt. atas segala limpahan rahmat, taufik, hidayah, serta karunia-Nya yang tiada henti, sehingga penulis diberikan kesehatan, kekuatan, kesabaran, dan kemudahan dalam menjalani setiap proses penyusunan skripsi ini. Berkat izin dan pertolongan-Nya, penulis akhirnya dapat menyelesaikan skripsi yang berjudul “Sistem Pemantauan Cerdas pada Ketersediaan Ruang Kelas Berbasis Internet of Things (IoT) pada Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh” sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh. Shalawat beserta salam semoga senantiasa tercurah kepada junjungan alam Nabi Besar Muhammad saw., beserta keluarga, para sahabat, dan seluruh umatnya yang senantiasa istiqamah mengikuti ajaran beliau hingga akhir zaman.

Penyusunan skripsi ini tidak terlepas dari berbagai tantangan dan hambatan. Namun, berkat pertolongan Allah Swt. serta doa, dukungan, bimbingan, dan bantuan dari berbagai pihak, setiap proses dapat dilalui dengan baik hingga skripsi ini dapat diselesaikan. Oleh karena itu, dengan penuh rasa hormat dan ketulusan hati, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan kontribusi, baik secara langsung maupun tidak langsung, selama proses penyusunan skripsi ini.

Karya sederhana ini penulis persembahkan dengan penuh rasa syukur dan cinta kepada orang-orang yang telah menjadi alasan utama penulis mampu bertahan dan melangkah sejauh ini, yaitu:

1. Penulis mengucapkan terima kasih kepada diri sendiri yang telah berusaha, bertahan, dan berkomitmen menyelesaikan setiap tahapan penelitian hingga Tugas Akhir ini dapat diselesaikan.
2. Teruntuk kedua orang tua, Ayah tercinta, Bapak M. Nasir dan Mamak tercinta, Ibu Zulbaidah yang telah menjadi sumber kekuatan, kasih sayang, dan doa dalam setiap langkah kehidupan penulis. Terima kasih atas segala pengorbanan yang tidak pernah dapat terbalaskan, atas kerja keras, kesabaran, serta cinta yang selalu diberikan tanpa mengenal lelah. Setiap

pencapaian dalam skripsi ini tidak lepas dari doa dan dukungan yang senantiasa mengiringi penulis. Penulis menyadari bahwa tidak ada kata yang mampu menggambarkan besarnya jasa Ayah dan Mamak. Semoga skripsi ini menjadi salah satu bentuk kecil kebahagiaan dan rasa bangga yang dapat penulis persembahkan atas segala perjuangan yang telah diberikan. Terima kasih telah percaya kepada penulis ketika keadaan terasa sulit, tetap menguatkan saat penulis hampir menyerah, dan selalu menjadi tempat pulang yang penuh kasih. Semoga Allah Swt, senantiasa melimpahkan kesehatan, kebahagiaan, keberkahan, serta membalas setiap pengorbanan Ayah dan Mamak dengan pahala yang berlipat ganda. Aamiin.

3. Bapak Prof. Dr. Ir. M. Dirhamsyah, M.T., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh, yang telah memberikan dukungan, perhatian, serta fasilitas selama penulis menempuh pendidikan hingga terselesaikannya Tugas Akhir ini.
4. Ibu Malahayati, M.T., selaku Ketua Program Studi Teknologi Informasi, dan Bapak Khairan AR, M.Kom., selaku Sekretaris Program Studi Teknologi Informasi, serta seluruh dosen Program Studi Teknologi Informasi, penulis mengucapkan terima kasih atas arahan, bimbingan, motivasi dan ilmu yang telah diberikan selama masa perkuliahan.
5. Bapak Mursyidin, M.T. selaku dosen pembimbing I dan Bapak Ridha Ilahi, M.T. selaku dosen pembimbing II, penulis menyampaikan terima kasih yang sebesar-besarnya atas kesabaran, perhatian, serta bimbingan yang diberikan hingga terselesaikannya Tugas Akhir ini.
6. Seluruh dosen dan staf Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh, penulis mengucapkan terima kasih atas bantuan dan pelayanan yang diberikan selama masa studi.
7. Ibu Cut Ida Rahmadiana, S.Si., selaku operator Program Studi Teknologi Informasi, penulis mengucapkan terima kasih yang telah membantu penulis dalam proses administrasi akademik.
8. Teruntuk saudara-saudara kandung tercinta, kakak Nasvizarni, abang Nasrizal Afriadi, dan kakak Ade Nasvirda Monika, penulis mengucapkan

terima kasih yang selalu memberikan doa, semangat, perhatian, serta dukungan kepada penulis selama proses penyusunan skripsi ini. Terima kasih atas segala bentuk bantuan, motivasi, dan pengorbanan yang telah diberikan, sehingga penulis mampu menyelesaikan skripsi ini.

9. Teruntuk abang ipar, abang Toni Firmayas penulis mengucapkan terima kasih atas segala doa, perhatian, dukungan, motivasi, serta semangat yang senantiasa diberikan kepada penulis selama proses penyusunan skripsi ini. Terima kasih telah menjadi bagian dari keluarga yang selalu memberikan dorongan dan membantu penulis dalam menghadapi berbagai tantangan hingga skripsi ini dapat diselesaikan.
10. Teruntuk sahabat sejak kecil, Erni Sabila dan Wina Ayuni, penulis mengucapkan terima kasih yang telah setia menemani perjalanan penulis selama ini. Terima kasih atas doa, dukungan, semangat, perhatian, serta kebersamaan yang selalu diberikan, baik dalam suka maupun duka. Kehadiran kalian menjadi salah satu penyemangat bagi penulis untuk terus berjuang hingga akhirnya penulis dapat menyelesaikan skripsi ini. Semoga Allah Swt. membalas segala kebaikan yang telah kalian berikan dengan balasan yang terbaik. Aamiin.
11. Teruntuk teman-teman dekat di luar lingkungan kampus penulis, Mirna ListaRoza Nadelia, yang telah memberikan dukungan, motivasi, serta menjadi tempat berbagi cerita selama proses penyusunan skripsi ini. Terima kasih atas waktu, perhatian, dan semangat yang selalu diberikan di tengah berbagai tantangan yang penulis hadapi.
12. Teruntuk Dewi Ayu Saputri, selaku teman seperjuangan sekaligus teman dekat penulis selama menempuh perkuliahan. Terima kasih atas segala kebersamaan, dukungan, motivasi, bantuan, serta semangat yang selalu diberikan kepada penulis, baik dalam proses perkuliahan maupun selama penyusunan skripsi ini. Kehadiranmu sebagai teman untuk berbagi cerita, berdiskusi, dan saling menguatkan telah menjadi bagian yang sangat berarti dalam perjalanan akademik penulis.

13. Teruntuk teman-teman mahasiswa Teknologi Informasi angkatan 2021 penulis mengucapkan terima kasih telah memberikan dukungan, semangat, serta kebersamaan selama proses perkuliahan.
14. Teruntuk pemilik NIM 1806103030026 penulis mengucapkan terima kasih atas segala perhatian, dukungan, doa, motivasi, serta waktu yang telah diberikan kepada penulis selama proses penyusunan skripsi ini. Terima kasih telah menjadi sosok yang selalu hadir untuk memberikan semangat, mendengarkan setiap keluh kesah, serta memberikan dorongan ketika penulis menghadapi berbagai kesulitan dalam menyelesaikan skripsi ini. Kehadiran dan dukungan yang diberikan menjadi salah satu alasan yang membuat penulis mampu bertahan dan terus berusaha hingga akhirnya skripsi ini dapat diselesaikan. Penulis sangat menghargai setiap bantuan, masukan, dan kepedulian yang telah diberikan, baik secara langsung maupun tidak langsung. Semua bentuk kebaikan tersebut menjadi kenangan yang sangat berarti dan akan selalu penulis ingat sebagai bagian dari perjalanan dalam menyelesaikan pendidikan ini, dan terima kasih telah menjadi penyemangat terbesar dalam hidup penulis.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki keterbatasan dan belum sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga penelitian ini dapat memberikan manfaat bagi pengembang ilmu pengetahuan.

Banda Aceh, 03 Juli 2026

Zabrina Dewi

DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
BAB II LANDASAN TEORI	5
2.1 Relevansi Penelitian	5
2.2 Sistem Pemantauan Cerdas	7
2.3 Internet of Things (IoT).....	7
2.4 Mikrokontroler ESP32	8
2.5 Sensor PIR (Passive Infrared).....	8
2.6 Sensor Jarak VL53L0X (GY-530).....	9
2.7 Platform IoT Blynk	10
2.8 Google Apps Script dan Webhook	11
2.9 Network Time Protocol (NTP).....	11
2.10 Arduino IDE dan Pustaka (Library)	11
BAB III METODE PENELITIAN	13
3.1 Kerangka Berpikir	13
3.2 Diagram Penelitian	14
3.2.1 Identifikasi Masalah.....	14
3.2.2 Analisis Kebutuhan Sistem.....	15
3.2.3 Perancangan Sistem	17

3.2.4 Implementasi Sistem.....	21
3.2.5 Pengujian Sistem	21
3.3 Kebutuhan Sistem.....	22
3.3.1 Kebutuhan Perangkat Keras (Hardware)	23
3.3.2 Kebutuhan Perangkat Lunak (Software).....	24
3.4 Waktu dan Tempat Penelitian	25
BAB IV HASIL DAN PEMBAHASAN	26
4.1 Implementasi Sistem	26
4.1.1 Implementasi Perangkat Keras	26
4.1.2 Implementasi Perangkat Lunak (<i>Code Breakdown</i>)	27
4.2 Hasil Pengujian Sistem.....	38
4.2.1 Pengujian Jangkauan dan Responsivitas Sensor PIR	38
4.2.2 Pengujian Transisi State Machine dan Analisis Error Rate	38
4.2.3 Pengujian Ambang Batas Jarak Sensor VL53L0X.....	41
4.2.4 Pengujian Kinerja Logika Status Ruang.....	44
4.2.5 Pengujian Sinkronisasi dan Latensi Ekosistem Awan (Cloud)	47
4.2.6 Pengujian Peringatan Notifikasi (Event Trigger)	48
4.3 Analisis Komparatif dan Evaluasi Sistem (Pembahasan)	48
BAB V KESIMPULAN DAN SARAN	50
5.1 Kesimpulan.....	50
5.2 Saran	51
DAFTAR PUSTAKA.....	53

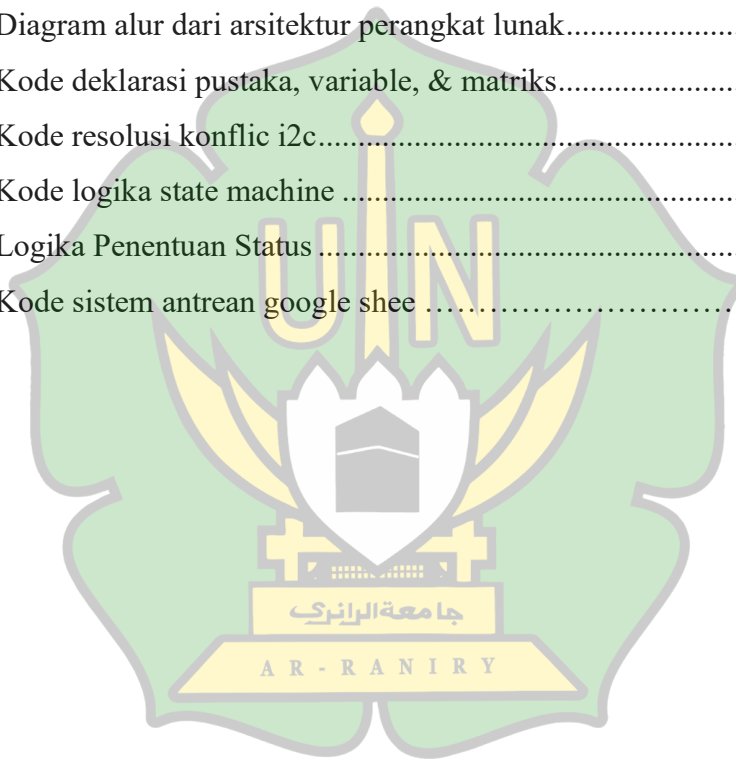
DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	5
Tabel 3. 1 Spesifikasi Kebutuhan Perangkat Keras	23
Tabel 3. 2 Spesifikasi Kebutuhan Perangkat Lunak	24
Tabel 4. 1 Pemetaan Matriks Jadwal Akademik (Sesi PP)	29
Tabel 4. 2 Hasil Pengujian Black-box Logika Sensor PIR	38
Tabel 4. 3 Hasil Pengujian Skenario Mobilitas dan Perhitung Error Rate.....	39
Tabel 4. 4 Matriks Pengujian Ambang Batas Jarak	42
Tabel 4. 5 Matriks Evaluasi Status Kombinasi Ruang (NTP Aktif)	45
Tabel 4. 6 Matriks Hasil Pengujian Lapangan Logika Status Ruang	46
Tabel 4. 7 Evaluasi Latensi Pengiriman Latar Belakang (Queue HTTP)	47



DAFTAR GAMBAR

Gambar 2. 1 ESP 32 by NodeMCU	8
Gambar 2. 2 Sensor Passive Infrared HC-SR501	9
Gambar 2. 3 Sensor ToF VL53L0X (GY530)	10
Gambar 2. 4 Contoh widget dashboard pada blynk	10
Gambar 3. 1 Kerangka berpikir.....	14
Gambar 3. 2 Diagram penelitian	14
Gambar 3. 3 Blok diagram dari perangkat keras sistem	17
Gambar 3. 4 Diagram skematik perangkat keras	18
Gambar 3. 5 Diagram alur dari arsitektur perangkat lunak.....	19
Gambar 4. 1 Kode deklarasi pustaka, variable, & matriks.....	28
Gambar 4. 2 Kode resolusi konflik i2c.....	31
Gambar 4. 3 Kode logika state machine	33
Gambar 4. 4 Logika Penentuan Status	35
Gambar 4. 5 Kode sistem antrean google shee	37



BAB I

PENDAHULUAN

1.1 Latar Belakang

Universitas Islam Negeri Ar-Raniry Banda Aceh, khususnya Fakultas Sains dan Teknologi, merupakan salah satu institusi pendidikan tinggi yang terus berkembang dalam upaya mencetak lulusan yang kompeten di bidang sains dan teknologi. Fakultas ini menampung berbagai aktivitas akademik yang melibatkan dosen, tenaga kependidikan, serta mahasiswa dalam berbagai kegiatan perkuliahan, praktikum, dan penelitian. Jumlah kelas yang padat menuntut adanya pengelolaan ruang yang tertib dan efisien demi menciptakan lingkungan pembelajaran yang kondusif (Al Aluf et al., 2025). Fakultas Sains dan Teknologi sendiri menaungi enam program studi, yakni: Arsitektur, Teknik Lingkungan, Kimia, Biologi, Teknologi Informasi, dan Teknik Fisika. Dinamika kegiatan yang kompleks ini mempertegas urgensi kebutuhan sebuah sistem pengelolaan ruang kelas yang cerdas dan akurat.

Berdasarkan informasi dari pihak akademik Fakultas Sains dan Teknologi, pengelolaan jadwal masih menghadapi berbagai kendala serius. Sering terjadi perubahan jadwal perkuliahan secara mendadak serta pertukaran ruang antarkelas tanpa pencatatan resmi di Sistem Informasi Akademik (Siakad). Permasalahan koordinasi dan transparansi informasi ini tentu dapat mengganggu kelancaran proses belajar mengajar secara menyeluruh, karena kelas yang tidak dikelola dengan baik akan menghambat jalannya pendidikan (Kirana, 2023). Komunikasi terkait ketersediaan ruang saat ini hanya dilakukan melalui grup pesan singkat tanpa pendokumentasian formal, sehingga penggunaan kelas tidak tercatat dengan presisi dan tidak tersedia data historis yang valid untuk bahan evaluasi pihak fakultas.

Beberapa penelitian mulai menerapkan teknologi *Internet of Things* (IoT) untuk mengotomatisasi pemantauan kelas guna mengatasi permasalahan tersebut. Latief et al. (2024) memanfaatkan teknologi IoT untuk memantau kondisi dan tingkat kehadiran di ruang kelas secara *real-time*. Di sisi lain, Marques dan Pitarma

(2023) mengintegrasikan sensor pintar untuk mendeteksi okupansi dan sirkulasi udara pada ruangan tertutup. Meskipun pendekatan tersebut terbukti efektif, sebagian besar sistem masih memiliki keterbatasan; yakni tidak menyediakan antarmuka terbuka bagi seluruh sivitas akademika, serta minimnya perekaman riwayat historis (seperti pengintegrasian dengan *spreadsheet*) yang dapat diakses langsung oleh pengelola.

Untuk menjembatani celah tersebut, penelitian ini merancang "Sistem Pemantauan Cerdas Ketersediaan Ruang Kelas Berbasis *Internet of Things* (IoT)". Secara teknis, sistem ini memadukan sensor jarak VL53L0X (modul GY-530) berbasis *Time-of-Flight* (ToF) yang dipasang di area pintu untuk menghitung jumlah arus keluar masuk pengguna, serta mengandalkan dua unit sensor Passive Infrared (PIR) untuk mengunci status keberadaan dosen dan mahasiswa di dalam kelas. Melalui kombinasi ini, sistem diharapkan mampu menyajikan informasi ketersediaan ruang yang di-log secara aktual, dicatat ke pangkalan data historis, serta divisualisasikan secara transparan untuk mendukung transformasi manajemen kelas digital di UIN Ar-Raniry.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah dalam penelitian ini adalah bagaimana merancang dan membangun sistem pemantauan cerdas berbasis *Internet of Things* (IoT) yang realistis untuk mendeteksi, memantau, dan mencatat ketersediaan ruang kelas di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang, mengimplementasikan, serta menguji sebuah sistem pemantauan cerdas berbasis *Internet of Things* (IoT) yang mampu mendeteksi keberadaan pengguna dan merekam status ketersediaan ruang kelas secara *real-time* di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. **Manfaat Teoritis:** Memberikan kontribusi akademik terhadap literatur di bidang teknologi informasi, khususnya mengenai penerapan, efisiensi, dan analisis limitasi perangkat berbasis *Internet of Things* (IoT) dalam menunjang otomatisasi fasilitas pendidikan tinggi.
2. **Manfaat Praktis:** Bagi Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh, ini menjadi landasan awal untuk merapikan manajemen kelas melalui pencatatan historis yang terotomatisasi. Bagi dosen dan mahasiswa, sistem ini memberikan kemudahan untuk memeriksa status kekosongan ruangan kelas dari jarak jauh, sehingga meminimalisir miskomunikasi jadwal dan membuat operasional perkuliahan menjadi jauh lebih efisien.

1.5 Batasan Masalah

Agar arah penelitian berjalan secara obyektif serta bersikap realistis terhadap kapabilitas dan keterbatasan kinerja dari perangkat keras (*hardware*) yang digunakan, batasan masalah ditetapkan sebagai berikut:

1. Fokus Lokasi Uji Coba: Implementasi penelitian ini dibatasi hanya pada pembuatan untuk satu ruang kelas di Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh.
2. Ketergantungan Infrastruktur Jaringan: Pemrosesan dan pengiriman data ke *dashboard* Blynk serta *Google Spreadsheet* sangat bergantung pada stabilitas koneksi internet atau WiFi.
3. Kelemahan Sensor Penghitung (*Counter*): Perhitungan jumlah orang murni mengandalkan tembakan jarak sensor VL53L0X. Sensor ini dapat mengalami penurunan akurasi atau kesalahan hitung (*double counting*) apabila pintu masuk dalam kondisi padat di mana terdapat pengguna yang berjalan secara berhimpitan dalam waktu yang bersamaan.
4. Pendeteksian Bias pada Sensor Kehadiran: Identifikasi okupansi ruang menggunakan sensor PIR (*Passive Infrared*) yang bekerja dengan mendeteksi perubahan suhu inframerah. Oleh karena itu, sensor memiliki

potensi bias deteksi (*false positive*) apabila merespons pergerakan objek non-manusia atau perubahan hawa panas lingkungan yang melintasi area kelas.



BAB II

LANDASAN TEORI

2.1 Relevansi Penelitian

Dalam mendukung penelitian ini, penulis mengkaji sejumlah literatur dan studi terdahulu yang relevan. Kajian ini bertujuan untuk menghindari plagiarisme, memberikan batasan kebaruan (*novelty*), serta menjadi acuan teknis dalam penyusunan sistem. Perbandingan antara penelitian ini dengan penelitian-penelitian sebelumnya ditampilkan pada Tabel 2.1.

Tabel 2. 1 Penelitian Terdahulu

No	Penulis & Tahun	Judul Penelitian	Kesimpulan Utama
1	(Latief et al., 2024)	<i>Sistem Pemantauan Ruang Kelas berbasis Internet of Things (IoT) untuk Proses Pendidikan yang Efektif</i>	Penerapan IoT efektif untuk memantau kehadiran dan kondisi ruangan kelas. Namun, sistem belum mengintegrasikan pencatatan ke pangkalan data awan eksternal seperti <i>Spreadsheet</i> .
2	(Marques & Pitarma, 2023)	<i>Monitoring Indoor Air Quality and Occupancy with an IoT System: Evaluation in a Classroom Environment</i>	Menggunakan gabungan sensor untuk mendeteksi okupansi kelas dan kualitas udara. Penelitian difokuskan pada kesehatan lingkungan, bukan pada status ketersediaan jadwal.
3	(Abdelrahman et al., 2025)	<i>Conceptual Design of IoT-Enabled Parking Lot Monitoring System using PIR Motion and Ultrasonic Sensors for Real-Time Occupancy Detection</i>	Penggabungan sensor jarak dan sensor PIR terbukti ampuh mengurangi <i>false positive</i> (kesalahan deteksi) dalam mengidentifikasi okupansi suatu area secara <i>real-time</i> .
4	(Milda et al., 2025)	<i>Rancang Bangun Sistem Palang Pintu Otomatis Berbasis Arduino Uno Dengan Teknologi RFID</i>	Menunjukkan bahwa integrasi mikrokontroler (ESP/Arduino) dan <i>Google Sheets</i> berbasis <i>API Apps Script</i> sangat reliabel untuk mencatat log aktivitas (riwayat data) tanpa memerlukan penyewaan <i>server SQL</i> .
5	(Putra & Wibisono, 2025)	<i>IoT-Based Smart Classroom Prototype Using NodeMCU and Blynk for Environmental Monitoring</i>	Membuktikan bahwa penggunaan papan ESP dan aplikasi Blynk sangat stabil dalam memantau serta mengontrol perangkat di ruang kelas pintar secara jarak jauh dengan latensi rendah.

6	(Bryo Kalesaran et al., 2025)	<i>Perancangan Sistem Penghitung Jumlah Pengunjung Secara Real-Time Untuk Optimalisasi Manajemen Kafe</i>	Menunjukkan bahwa perancangan sensor penghitung orang (<i>people counter</i>) yang terintegrasi IoT mampu merekam fluktuasi arus kedatangan manusia dengan akurat untuk keperluan manajemen fasilitas.
7	(Fidarliyan et al., 2023)	<i>Classroom Occupancy Monitoring System using IoT Device and the k-Nearest Neighbors Algorithm</i>	Menyimpulkan bahwa data okupansi ruangan berbasis sensor IoT dapat diandalkan untuk membaca kepadatan kelas, meskipun pada penelitian tersebut memerlukan beban komputasi <i>machine learning</i> (k-NN) tambahan.
8	(Syahputra & Santoso, 2025)	<i>Rancang Bangun Sistem Absensi Otomatis Berbasis RFID dan ESP32 di Kampus AMIK Polibisnis Perdagangan</i>	Sistem dengan otak mikrokontroler ESP32 sangat tangguh untuk melakukan otomasi pendataan kehadiran di lingkup akademik, menggantikan sistem rekap manual yang rentan kesalahan.
9	(Manik et al., 2024)	<i>Integrasi Sensor Ultrasonik dan Computer Vision (YOLO) Berbasis ESP32-CAM untuk Klasifikasi Objek pada Sistem Parkir</i>	Memvalidasi bahwa arsitektur ESP32 mampu menangani kalkulasi sensor pembaca jarak (<i>ranging sensor</i>) dan komunikasi transmisi data IoT dengan sangat baik secara bersamaan.
10	(Taha et al., 2023)	<i>Edge-Based Real-Time Occupancy Detection System through a Non-Intrusive Sensing System</i>	Memastikan bahwa penggunaan sensor non-intrusif (seperti sensor gerak pasif dan sensor jarak inframerah) memberikan tingkat keamanan privasi yang jauh lebih baik dibandingkan sistem pendeteksi okupansi berbasis kamera.

Persamaan:

1. Keseluruhan penelitian menggunakan pendekatan infrastruktur Internet of Things (IoT) sebagai tulang punggung (backbone) pengiriman data pembacaan sensor menuju dashboard pemantauan digital (Latief et al., 2024; Putra & Wibisono, 2025).
2. Menggunakan prinsip penggabungan komplementer multi-sensor (seperti gabungan sensor pergerakan/PIR dan sensor jarak pantul) untuk menentukan status okupansi aktual suatu ruangan (Abdelrahman et al., 2025; Manik et al., 2024).

Perbedaan:

1. Fokus Pemantauan: Penelitian ini murni berfokus pada penentuan ketersediaan ruang kelas berbasis kecocokan komparasi Periode Pembelajaran (PP) perkuliahan. Sistem menggunakan sensor Time-of-Flight (VL53L0X) dengan akurasi pendar laser untuk menghitung arus keluar-masuk (mobilitas), alih-alih sekadar mendeteksi kenyamanan fisik/udara ruangan seperti mayoritas studi sebelumnya (Marques & Pitarma, 2023; Taha et al., 2023).
2. Penyimpanan Data Historis: Penelitian ini mengombinasikan visualisasi real-time di aplikasi Blynk dengan penyimpanan log permanen menggunakan Webhook berformat JSON menuju Google Spreadsheet. Pendekatan pengarsipan data sesi tanpa basis data berbayar ini jarang digabungkan secara simultan pada ekosistem pemantauan okupansi kelas terdahulu (Milda et al., 2025).

2.2 Sistem Pemantauan Cerdas

Sistem pemantauan cerdas merupakan infrastruktur teknologi yang dirancang untuk mengawasi kondisi atau aktivitas tertentu secara otomatis melalui integrasi sensor, mikrokontroler, dan konektivitas jaringan (Latief et al., 2024). Dalam lingkup akademik, penerapan pemantauan cerdas memungkinkan identifikasi ruang kelas secara langsung tanpa campur tangan manual. Hal ini meminimalkan terjadinya tumpang tindih penggunaan ruang yang disebabkan oleh perubahan jadwal mendadak, serta menyediakan transparansi data fasilitas kampus bagi seluruh sivitas akademika.

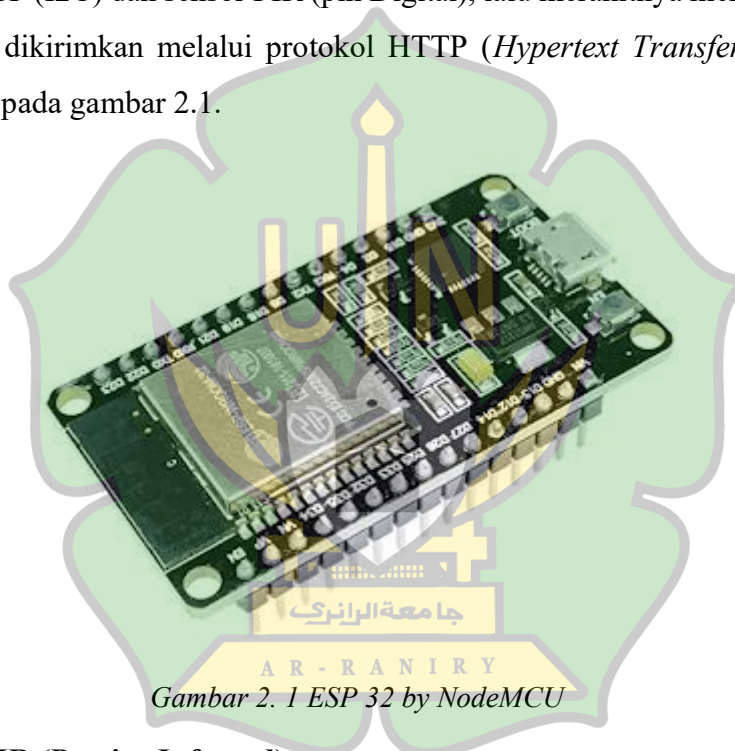
2.3 Internet of Things (IoT)

Internet of Things (IoT) adalah sebuah paradigma teknologi yang menghubungkan berbagai objek fisik (perangkat keras) ke dalam jaringan internet, sehingga memungkinkan terjadinya pertukaran data secara mandiri tanpa memerlukan interaksi langsung antara manusia ke komputer (Sahrul et al., 2023). Pada sistem pengelolaan kelas ini, IoT berperan menjembatani data fisik yang

dibaca oleh sensor (seperti pergerakan manusia dan deteksi jarak) untuk dikirimkan dan diproses pada *server cloud*.

2.4 Mikrokontroler ESP32

ESP32 adalah mikrokontroler canggih yang dikembangkan oleh Espressif Systems. Chip ini dilengkapi dengan prosesor *dual-core* berkecepatan 240 MHz dan telah memiliki modul WiFi (802.11 b/g/n) serta *Bluetooth* terintegrasi di dalam satu keping sirkuit (Milda et al., 2025). Pada penelitian ini, ESP32 digunakan sebagai otak pengendali (pemroses logika *State Machine*) yang membaca kondisi dari sensor ToF (I2C) dan sensor PIR (pin Digital), lalu merakitnya menjadi format JSON untuk dikirimkan melalui protokol HTTP (*Hypertext Transfer Protocol*). Dapat dilihat pada gambar 2.1.



Gambar 2. 1 ESP 32 by NodeMCU

2.5 Sensor PIR (Passive Infrared)

Sensor PIR (*Passive Infrared*) merupakan sensor yang digunakan untuk mendeteksi pergerakan berdasarkan perubahan radiasi panas (inframerah) yang dipancarkan oleh tubuh manusia (Abdelrahman et al., 2025). Sensor ini bersifat pasif karena tidak memancarkan cahaya apapun, melainkan hanya menyerap inframerah di sekitarnya. Dalam sistem ini, dua unit sensor PIR digunakan secara terpisah (GPIO 27 untuk dosen dan GPIO 26 untuk mahasiswa). Sensor membaca pergerakan dan mengirimkan sinyal digital HIGH (3.4V) ke ESP32. Sinyal ini digunakan oleh sistem untuk mengaktifkan fungsi *latch* (pengunci status memori)

guna menentukan apakah kelas dalam kondisi aktif digunakan atau kosong. Dapat dilihat pada gambar 2.2.



Gambar 2. 2 Sensor Passive Infrared HC-SR501

2.6 Sensor Jarak VL53L0X (GY-530)

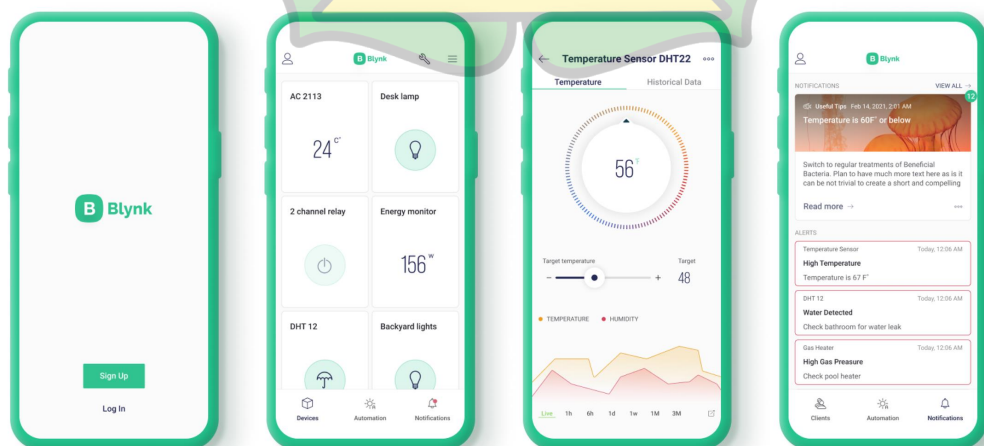
VL53L0X (umumnya dikemas dalam modul GY-530) adalah sensor pengukur jarak presisi tinggi yang menggunakan teknologi *Time-of-Flight* (ToF). Berbeda dengan sensor ultrasonik, sensor ToF bekerja dengan memancarkan pulsa laser inframerah yang kasatmata (940 nm) dan menghitung waktu pantulannya (*photon travel time*) menggunakan susunan SPAD (*Single Photon Avalanche Diodes*). Sistem ini menggunakan dua unit VL53L0X yang dikonfigurasi melalui jalur I2C (SDA Pin 21, SCL Pin 22) untuk menghitung arus keluar-masuk orang di ambang pintu secara dinamis. Karena kedua sensor secara bawaan memiliki alamat I2C yang sama, sistem menggunakan pin *shutdown* perangkat keras (XSHUT pada GPIO 13 dan 14) untuk mematikan dan meregistrasi ulang alamat I2C salah satu sensor (0x30 dan 0x31) saat alat pertama kali dihidupkan (*booting*). Dapat dilihat pada gambar 2.3.



Gambar 2. 3 Sensor ToF VL53L0X (GY530)

2.7 Platform IoT Blynk

Blynk merupakan *platform* IoT berbasis *cloud* yang memfasilitasi pembuatan antarmuka (*dashboard*) pemantauan secara *real-time*. Pada sistem ini, komunikasi antara ESP32 dan Blynk dikelola menggunakan pustaka BlynkSimpleEsp32.h (Latief et al., 2024). Data hasil kalkulasi dikirim menggunakan mekanisme *Virtual Pin* (seperti V0 untuk status dosen, V2 untuk penghitung orang, dan V3 untuk status ruang teks). Selain itu, sistem memanfaatkan fitur BlynkTimer untuk melakukan proses *multithreading* ringan yang mencegah sensor ToF mengalami *lag*, serta menggunakan fungsi Blynk.logEvent() untuk mendorong notifikasi interaktif ke ponsel pintar pengguna. Dapat dilihat pada gambar 2.4.



Gambar 2. 4 Contoh widget dashboard pada blynk

2.8 Google Apps Script dan Webhook

Untuk menyimpan rekam jejak (*log*) historis penggunaan ruang secara permanen tanpa menyewa pangkalan data (*database*) SQL, sistem ini memanfaatkan layanan Google *Sheets* yang dijembatani oleh Google *Apps Script*. *Apps Script* bertindak sebagai *Webhook* yang menerima HTTP *POST Request* dari mikrokontroler. ESP32 memformat data kejadian (Waktu, PP, Total Orang, Status) ke dalam bentuk struktur JSON (*JavaScript Object Notation*). Berdasarkan implementasinya pada pustaka *HTTPClient.h*, sistem menggunakan metode asinkron (antrean *array*) dan mengabaikan paksaan perpindahan URL (*HTTPC_STRICT_FOLLOW_REDIRECTS*) agar sensor ESP32 tidak mengalami pembekuan proses (*freeze*) ketika *server* Google membalas dengan status HTTP 302 (Milda et al., 2025).

2.9 Network Time Protocol (NTP)

Network Time Protocol (NTP) adalah protokol jaringan yang digunakan untuk sinkronisasi jam sistem komputer dengan referensi waktu global yang sangat presisi. Menggunakan pustaka *time.h*, ESP32 pada penelitian ini secara periodik mengambil data dari *server* waktu lokal (*pool.ntp.org*) untuk mendapatkan waktu lokal (WIB) terkini. Waktu yang ditarik dari fungsi *getLocalTime()* kemudian dikomparasi secara algoritmik terhadap *array* jam dan menit jadwal PP akademik untuk mendikte otomatisasi status ketersediaan ruang kelas.

2.10 Arduino IDE dan Pustaka (Library)

Arduino Integrated Development Environment (IDE) merupakan perangkat lunak *open-source* yang digunakan untuk menulis dan mengkompilasi kode program C++ ke dalam mikrokontroler. Dalam membangun sistem logika ketersediaan ruang kelas ini, digunakan beberapa integrasi pustaka utama, meliputi:

1. *WiFi.h*: Untuk menghubungkan perangkat keras ESP32 ke titik akses internet.
2. *Wire.h*: Untuk mengatur protokol komunikasi dua-kabel (I2C) pada perangkat keras.

3. VL53L0X.h (oleh Pololu): Pustaka spesifik yang digunakan untuk menginisialisasi mode pembacaan *High Speed Profile* (`setMeasurementTimingBudget`) dan pembacaan berkesinambungan (`startContinuous`) pada sensor GY-530.



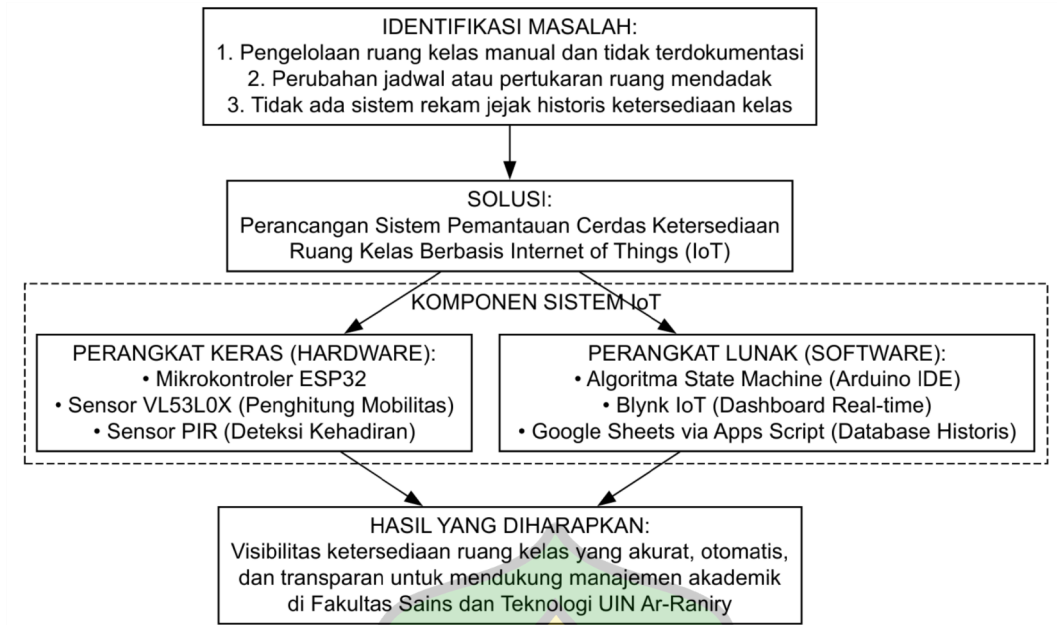
BAB III

METODE PENELITIAN

3.1 Kerangka Berpikir

Kerangka berpikir dalam penelitian ini disusun untuk menggambarkan alur hubungan logis antara identifikasi permasalahan di lapangan, landasan teori, hingga perancangan solusi teknis yang ditawarkan. Pengelolaan ketersediaan ruang kelas di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh yang selama ini berjalan manual sering kali memunculkan ketidakteraturan, terutama saat terjadi perubahan jadwal mendadak. Untuk menjawab permasalahan tersebut, pendekatan berbasis *Internet of Things* (IoT) digunakan sebagai infrastruktur pemantauan jarak jauh yang mampu memberikan visibilitas data aktual tanpa memerlukan intervensi manusia.

Pemanfaatan IoT dalam pemantauan fasilitas akademik terbukti mampu mereduksi redundansi operasional dan memberikan data yang lebih akurat untuk pengambilan keputusan strategis (Latief et al., 2024). Berpijak pada premis tersebut, kerangka sistem ini dirancang dengan menggabungkan dua jenis sensor utama. Sensor *Passive Infrared* (PIR) bertugas untuk membaca pancaran hawa panas (*infrared*) yang mengindikasikan keberadaan fisik manusia di dalam kelas (Abdelrahman et al., 2025). Secara bersamaan, sensor *Time-of-Flight* VL53L0X digunakan untuk menghitung arus keluar-masuk (mobilitas) di ambang pintu kelas. Kedua data ini dikumpulkan oleh mikrokontroler ESP32, diproses melalui algoritma *state machine*, lalu ditransmisikan secara nirkabel ke *dashboard* Blynk dan Google *Spreadsheet* untuk visualisasi serta penyimpanan data historis. Dapat dilihat pada gambar 3.1.



Gambar 3. 1 Kerangka berpikir

3.2 Diagram Penelitian

Diagram penelitian merupakan penjabaran dari langkah-langkah sistematis yang dilakukan oleh peneliti dari tahap awal hingga penyelesaian. Proses ini dirancang untuk memastikan integrasi antara mikrokontroler, sensor proksimitas, dan pangkalan data awan berjalan sesuai target fungsional. Rangkaian diagram penelitian ini dapat dilihat pada gambar 3.2.



Gambar 3. 2 Diagram penelitian

3.2.1 Identifikasi Masalah

Tahapan pertama berfokus pada observasi lapangan dan wawancara dengan pihak akademik Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh. Identifikasi ini bertujuan untuk memetakan alur birokrasi penggunaan ruang kelas dan menemukan titik lemah (*pain points*) dari sistem manual yang ada. Ketidadaan sistem terpusat yang memantau kekosongan ruang

menyebabkan pertukaran ruang berjalan tanpa pencatatan resmi. Kondisi ini tidak hanya menyulitkan mahasiswa dan dosen dalam mencari kelas kosong, tetapi juga menghalangi fakultas dalam mengevaluasi utilitas aset ruangnya. Oleh karena itu, diusulkanlah sebuah pemantauan digital yang berjalan secara *real-time*.

3.2.2 Analisis Kebutuhan Sistem

Tahapan analisis kebutuhan bertujuan untuk menentukan spesifikasi teknis dari perangkat keras (*hardware*) maupun perangkat lunak (*software*) agar sistem dapat berjalan secara stabil, ringan, dan berkelanjutan. Spesifikasi ini didasarkan pada parameter lingkungan ruang kelas serta infrastruktur jaringan yang tersedia.

1. Kebutuhan Perangkat Keras (*Hardware*)

Perancangan sirkuit IoT memerlukan integrasi komponen elektronika yang memiliki efisiensi daya tinggi namun didukung oleh kemampuan komputasi yang mumpuni (Sahrul et al., 2023). Perangkat keras yang digunakan meliputi:

1. Mikrokontroler ESP32: Digunakan sebagai pusat pemrosesan logika (otak sistem). Dipilihnya ESP32 versi *NodeMCU* didasarkan pada prosesor *dual-core* Xtensa 32-bit LX6 dan kelengkapan antarmuka komunikasi (I2C, GPIO digital) serta modul WiFi 2.4GHz yang telah tertanam langsung pada papan cip.
2. Sensor Jarak VL53L0X (GY-530): Sensor laser berbasis *Time-of-Flight* (ToF) yang mengukur jarak dengan memancarkan foton inframerah kasatmata (940 nm) dan menghitung waktu pantulannya. Sensor ini memiliki tingkat presisi yang jauh lebih baik dibandingkan sensor ultrasonik HC-SR04 konvensional. Sistem membutuhkan dua buah VL53L0X (satu untuk memantau sisi luar pintu, dan satu untuk sisi dalam).
3. Sensor *Passive Infrared* (PIR HC-SR501): Modul sensor yang bertugas mendeteksi perubahan suhu inframerah yang dipancarkan oleh objek hidup (tubuh manusia) yang melintasi lensa *Fresnel*-nya. Sensor ini membutuhkan tegangan operasional 5V (menggunakan pin VIN pada ESP32) agar

regulator internalnya tidak mengalami *brownout*, meskipun sinyal data keluarannya tetap aman di level 3.4V untuk ESP32 (Abdelrahman et al., 2025).

4. Perangkat Pendukung: Meliputi *Breadboard* sebagai papan sirkuit prototipe, kabel *jumper* (Male-to-Male dan Male-to-Female) untuk koneksi pin, serta catu daya (adaptor 5V/2A atau kabel Micro-USB/Type-C) untuk menyuplai daya ke ESP32.

2. Kebutuhan Perangkat Lunak (*Software*) Perangkat lunak mencakup sistem kompilasi, pustaka pembantu (*library*), serta *server* awan (*cloud server*) yang menyusun ekosistem IoT.

1. Arduino IDE: Lingkungan pengembangan terintegrasi (IDE) yang digunakan untuk menulis, melakukan *debugging*, dan mengkompilasi kode bahasa C/C++ menjadi berkas *binary* (*firmware*) yang ditanamkan ke memori ESP32.

2. Pustaka (*Libraries*):

- a. *WiFi.h*: Pustaka bawaan inti ESP32 untuk menghubungkan perangkat keras ke akses poin jaringan nirkabel.
- b. *BlynkSimpleEsp32.h*: Pustaka resmi Blynk untuk menjembatani protokol komunikasi data (API) ke *server* Blynk secara *real-time*.
- c. *Wire.h*: Pustaka pengatur bus I2C (*Inter-Integrated Circuit*) untuk mengatur transmisi data SDA dan SCL pada ESP32.
- d. *VL53L0X.h* (Pololu): Pustaka eksternal yang dioptimalkan untuk mengeksekusi inisialisasi sensor ToF, mengatur profil kecepatan baca (*Measurement Timing Budget*), dan fungsi pemindaian jarak terus-menerus (*Continuous Mode*).
- e. *HTTPClient.h*: Pustaka klien HTTP untuk memformat *header*, membuat paket *payload* berformat JSON, dan menangani *POST Request* ke *server* eksternal.
- f. *Time.h*: Pustaka standar bahasa C untuk mengekstraksi dan menormalisasi paket data waktu dari *server Network Time Protocol* (NTP).

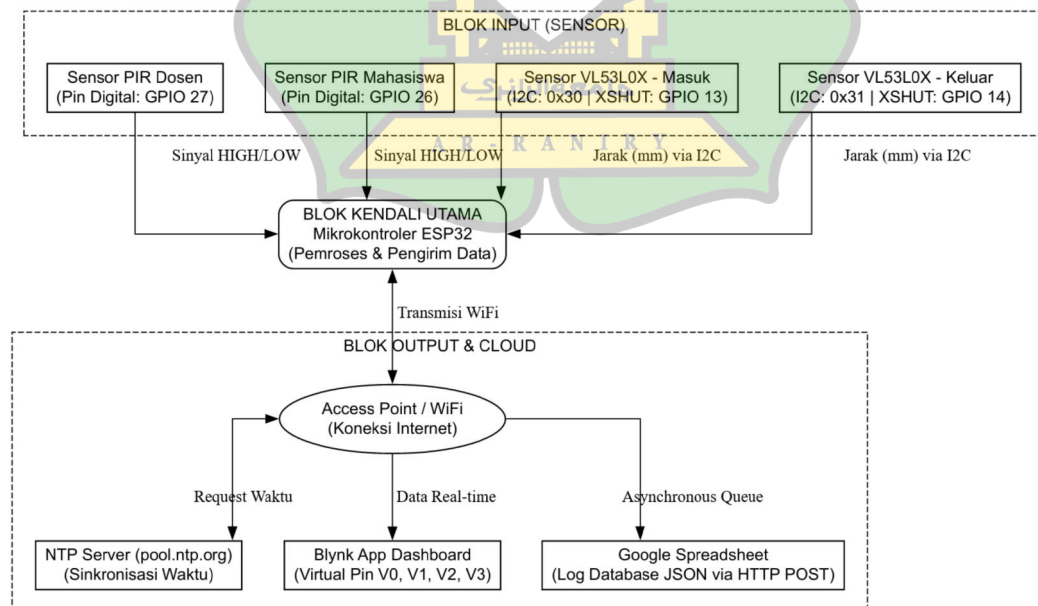
3. Blynk IoT Platform: Layanan antarmuka *dashboard* berbasis aplikasi gawai dan web. Digunakan untuk menciptakan representasi grafis (seperti Teks, Angka, dan *Virtual LED*) menggunakan fitur *Virtual Pin* (V0 hingga V3) tanpa harus membangun aplikasi Android dari awal.
4. Google *Apps Script* dan Google *Spreadsheet*: Digunakan sebagai alternatif pangkalan data relasional. *Apps Script* bertindak sebagai *Webhook* (penerima data) yang memecah (*parsing*) data JSON dari ESP32 dan menyusunnya ke dalam baris dan kolom baru di Google *Spreadsheet* (Milda et al., 2025).

3.2.3 Perancangan Sistem

Tahapan perancangan memetakan seluruh topologi sistem, baik dari tata letak komponen listrik maupun arsitektur alur perangkat lunaknya.

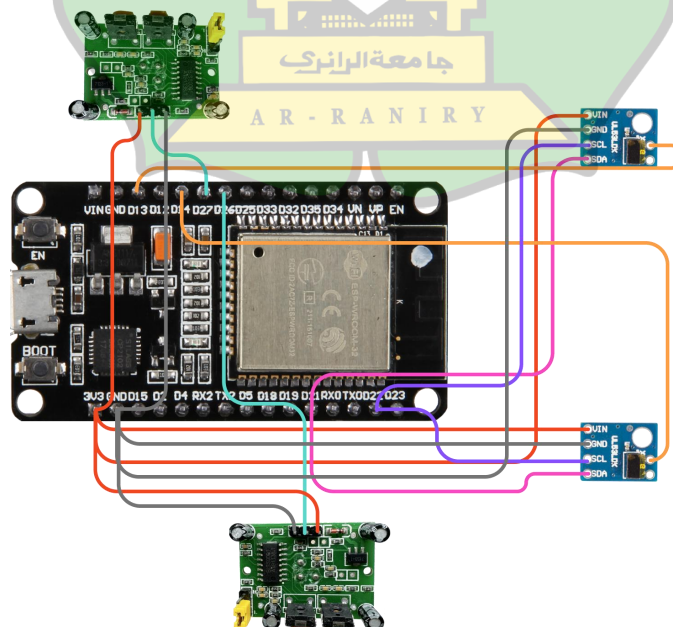
A. Perancangan Perangkat Keras (*Hardware Wiring*)

Pengkabelan perangkat keras harus didesain agar terhindar dari konflik alamat bus komunikasi dan *strap pin trap* (kegagalan *booting* akibat gangguan pin kelistrikan khusus ESP32). Skema *wiring* dirancang dengan parameter. Dapat dilihat pada gambar 3.3.



Gambar 3. 3 Blok diagram dari perangkat keras sistem

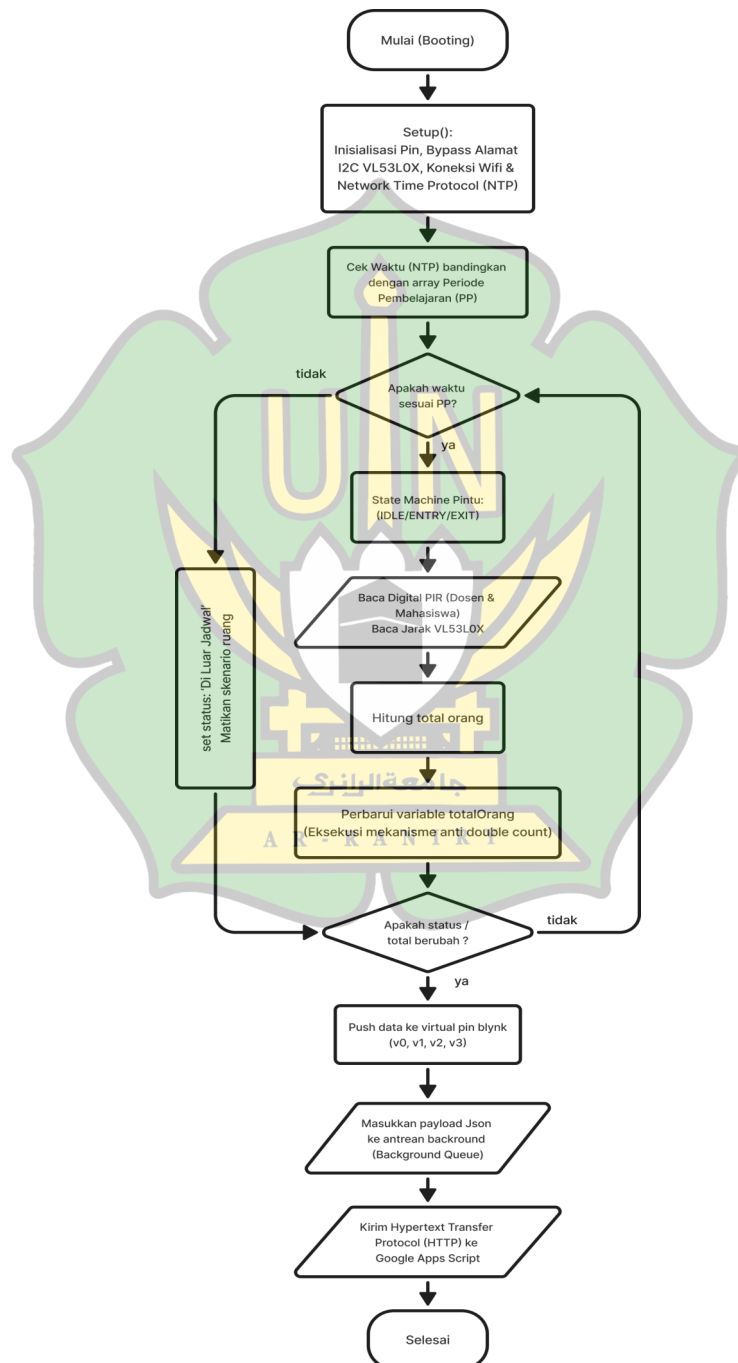
1. Pin Komunikasi I2C: Sensor VL53L0X pertama (Masuk) dan kedua (Keluar) dihubungkan secara paralel pada bus I2C utama ESP32, yakni Pin SDA pada GPIO 21 dan Pin SCL pada GPIO 22.
2. Resolusi Konflik Alamat I2C: Secara bawaan pabrik, setiap modul VL53L0X memiliki alamat identitas I2C statis 0x29. Jika dihubungkan bersamaan, sistem akan mengalami tabrakan data (*data collision*). Untuk memisahkannya, sistem memanfaatkan kaki XSHUT (*Shutdown*) pada sensor. Pin XSHUT sensor masuk dihubungkan ke GPIO 13, dan XSHUT sensor keluar dihubungkan ke GPIO 14 (dipindahkan dari GPIO 12 untuk mencegah *Strapping Pin Trap* saat *upload firmware*). Melalui kendali program, mikrokontroler mematikan kedua sensor saat *booting*, menyalakan sensor pertama dan mengganti alamatnya secara *software* menjadi 0x30, lalu menyalakan sensor kedua dan mengubah alamatnya menjadi 0x31.
3. Pin Sensor Kehadiran (PIR): Sensor PIR Dosen dihubungkan ke GPIO 27, sedangkan PIR Mahasiswa dihubungkan ke GPIO 26. Kedua pin ini dikonfigurasi sebagai INPUT digital murni untuk menangkap tegangan pemicu dari sensor. Dapat dilihat pada gambar 3.4.



Gambar 3. 4 Diagram skematik perangkat keras

B. Perancangan Perangkat Lunak (*Software Architecture*)

Arsitektur perangkat lunak disusun menggunakan paradigma *Asynchronous* ringan dan sistem *Multithreading* semu (*Timer*) agar mikrokontroler tidak mengalami *freeze* saat melakukan tugas berat secara bersamaan. Dapat dilihat pada gambar 3.5.



Gambar 3. 5 Diagram alur dari arsitektur perangkat lunak

- 1) Algoritma *State Machine* (Mesin Keadaan) Penghitung Pintu: Untuk menghitung arus keluar-masuk orang tanpa risiko hitungan ganda (*double counting*), sistem ToF tidak menggunakan jeda (*delay*), melainkan menggunakan model 4 transisi langkah (*DoorState*):
 - a. IDLE: Kondisi siaga. Jika jarak sensor luar (*distIn*) < 500 mm, status berubah ke ENTRY_STARTED. Jika jarak sensor dalam (*distOut*) < 500 mm, status berubah ke EXIT_STARTED.
 - b. ENTRY_STARTED: Mengindikasikan ada orang masuk ke ambang pintu. Sistem akan menunggu hingga sensor dalam ikut terpotong. Jika terpotong, jumlah *totalOrang* ditambahkan 1, dan status berpindah ke WAIT_CLEAR.
 - c. EXIT_STARTED: Mengindikasikan orang berjalan dari arah dalam kelas. Jika sensor luar kemudian terpotong, *totalOrang* dikurangi 1, lalu status berpindah ke WAIT_CLEAR.
 - d. WAIT_CLEAR: Fase penguncian sistem. Penghitungan tidak akan dilanjutkan hingga kedua pintu benar-benar bersih (jarak kedua sensor > 500 mm), untuk mencegah badan objek terhitung dua kali saat melintas.
- 2) Arsitektur *Background Queue* (Antrean *Hyper Text Transfer Protocol*): Proses pengiriman data aman melalui jalur enkripsi HTTPS ke *server* Google membutuhkan waktu sekitar 1 hingga 3 detik. Jika dieksekusi secara sinkron pada perulangan utama (*loop()*), sensor ToF akan mati suri selama 3 detik dan melewatkan objek yang lewat. Sistem ini merancang algoritma antrean dengan variabel array FIFO (*First-In-First-Out*). Ketika terdeteksi ada orang lewat, perintah POST tidak langsung dieksekusi, melainkan didorong (*push*) ke antrean string (*eventQueue*). Fungsi *processHttpQueue()* di dalam *loop()* utama kemudian akan mengeksekusi antrean tersebut satu persatu di latar belakang, dengan penyesuaian instruksi *http.setFollowRedirects* dinonaktifkan (kembali ke metode penerimaan respon HTTP 302/200

standar) untuk menghindari Error 400 *Bad Request* dari API Google (Milda et al., 2025).

- 3) Logika Kunci Kehadiran (*Sticky Latching*): Sensor PIR pendeteksi Dosen dikondisikan dengan logika *Latching* if (currentJadwal != -1 && pirDosen == HIGH) dosenWasDetected = true;. Ini berarti jika dosen terdeteksi bergerak sekali saja selama PP berlangsung, status "Dosen Hadir" akan dikunci secara permanen hingga kelas berakhir. Hal ini untuk mencegah fluktuasi status ruang kelas yang merancukan laporan (misal: dosen hanya duduk diam memeriksa tugas sehingga tidak memancarkan pergerakan ekstrem di depan sensor PIR) (Abdelrahman et al., 2025).
- 4) Isolasi Validasi Waktu Akademik (*Network Time Protocol Time-Blocking*): Sistem mengekstraksi jam terkini melalui pustaka `time.h` dan melakukan iterasi pengecekan terhadap variabel matriks multidimensi `scheduleHour` dan `scheduleMinute` (total 18 sesi jadwal PP/Istirahat). Apabila *output* waktu komparasi me- *return* nilai -1, seluruh sirkuit logika sensor akan dimatikan sementara, dan status ruang diubah menjadi "Di Luar Jadwal", menghemat alokasi memori *cloud* agar *spreadsheet* tidak penuh oleh pencatatan *log* aktivitas acak di tengah malam.

3.2.4 Implementasi Sistem

Implementasi merupakan fase perakitan kode program (*coding*) pada Arduino IDE, instalasi pustaka-pustaka yang diperlukan, serta mengunggah baris perintah (*flashing*) ke *flash memory* ESP32 dengan *baud rate* optimal 115200 bps. Tahap ini juga mencakup instalasi alat secara fisik, peletakan sensor ToF di rangka kusen pintu sejajar dada (ketinggian ± 120 cm untuk menghindari hitungan ganda terhadap kaki), dan penyolderan komponen ke papan pcb .

3.2.5 Pengujian Sistem

Tahap pengujian (*testing*) dilakukan menggunakan metode *Black-box* untuk mengevaluasi parameter fungsionalitas tanpa harus melihat kode

internalnya. Pengujian dipecah ke dalam beberapa skenario simulasi lingkungan akademik nyata:

1. Pengujian Jangkauan dan Responsivitas Sensor PIR: Mengukur jeda pengiriman variabel logika dari area fisik ke antarmuka aplikasi. Diuji pada kondisi statis (tanpa subjek), pergerakan aktif, dan uji durasi *Hold-time Delay* dari trimpot bawaan perangkat keras PIR.
2. Pengujian Transisi *State Machine* Sensor VL53L0X: Mengevaluasi kekuatan logika anti- *double counting*. Diuji menggunakan skenario subjek masuk, subjek keluar, subjek melintas secara berhimpitan, serta subjek yang berjalan masuk namun membatalkan langkah (mundur ke belakang sebelum memotong sensor kedua) guna memastikan variabel IDLE dan WAIT_CLEAR bekerja sempurna.
3. Pengujian Status Ruang (Sistem Logika Kombinasi): Menyelidiki keabsahan *output string* "Ruang Digunakan", "Hanya Dosen", "Hanya Mahasiswa", dan "Ruang Tidak Digunakan" berdasarkan silang komparasi data sensor PIR dan kuantitas manusia dari ToF.
4. Pengujian Sinkronisasi dan Latensi Ekosistem Awam (*Cloud*): Mengukur efikasi transmisi data dari ESP32 menuju Blynk dan Google *Spreadsheet*. Memastikan setiap pemicu aksi mencatat stempel waktu (*timestamp*) yang valid sesuai *Timezone* WIB, dan struktur pengkodean JSON tidak mengalami anomali atau data yang hilang (*packet loss*).
5. Pengujian Black-box pada Batas Peringatan Notifikasi (Event Trigger): Melakukan verifikasi sistem fungsi Blynk.logEvent(). Sistem diberi input pemicu aktivitas batas kritis. Output yang dievaluasi adalah ketepatan waktu terkirimnya pemberitahuan (push notification) ke smartphone pengguna, serta memastikan sistem tidak melempar output notifikasi sampah (spam) secara berulang saat ruangan sedang dalam kondisi diam (idle).

3.3 Kebutuhan Sistem

Pengembangan purwarupa "Sistem Pemantauan Cerdas Ketersediaan Ruang Kelas Berbasis Internet of Things (IoT)" membutuhkan integrasi alat yang spesifik, baik dari sisi perangkat keras (hardware) untuk akuisisi data dan

kompilasi, maupun perangkat lunak (software) untuk pemrograman dan pemantauan. Kebutuhan sistem ini dirancang dengan mempertimbangkan stabilitas eksekusi algoritma State Machine dan manajemen memori saat melakukan komunikasi jaringan secara asinkron.

3.3.1 Kebutuhan Perangkat Keras (Hardware)

Perangkat keras dalam penelitian ini diklasifikasikan menjadi dua bagian utama: perangkat keras simpul IoT (IoT Node) yang dipasang di ruang kelas, dan perangkat keras komputasi pengembang (Development Machine) yang digunakan peneliti untuk menulis, mengkompilasi, dan menanamkan (flashing) kode program.

Proses kompilasi rantai alat (toolchain) arsitektur Xtensa pada mikrokontroler ESP32 membutuhkan daya komputasi unit pemroses sentral (CPU) dan memori akses acak (RAM) yang jauh lebih besar dibandingkan kompilasi untuk papan mikrokontroler AVR konvensional seperti Arduino Uno (Milda et al., 2025). Oleh karena itu, dibutuhkan spesifikasi minimum perangkat komputasi agar perangkat lunak Integrated Development Environment (IDE) tidak mengalami kegagalan memori (Out of Memory Error) saat menyusun pustaka WiFi dan enkripsi HTTPS. Dapat dilihat pada tabel 3.1.

Tabel 3. 1 Spesifikasi Kebutuhan Perangkat Keras

Kategori	Nama Perangkat	Spesifikasi Minimum / Modul Spesifik
Komputasi Pengembang	Komputer Pribadi / Laptop	CPU: Intel Core i3 / AMD Ryzen 3 (Minimal Dual-Core 2.0 GHz)
Komputasi Pengembang	Memori (RAM) & Penyimpanan	RAM: 4 GB (Direkomendasikan 8 GB). Storage: Minimal 1 GB ruang kosong.
Simpul IoT (Kendali)	Mikrokontroler	ESP32 Development Board (NodeMCU)
Simpul IoT (Sensor)	Sensor Jarak Laser	2 x VL53L0X (Modul GY-530)
Simpul IoT (Sensor)	Sensor Pergerakan Pasif	2 x PIR HC-SR501

Infrastruktur Jaringan	<i>Access Point</i> (Router)	Router nirkabel standar 802.11 b/g/n (2.4 GHz)
Elektronik Pendukung	Catu Daya & Kabel	Kabel Micro-USB/Type-C (Mendukung transfer data), <i>Breadboard</i> , Kabel <i>Jumper</i>

3.3.2 Kebutuhan Perangkat Lunak (Software)

Perangkat lunak yang digunakan mencakup lingkungan pemrograman lokal, pustaka (libraries) eksternal, dan platform layanan awan (cloud services). Penggunaan pustaka versi terbaru dan teruji sangat diwajibkan untuk menjamin keamanan pengiriman data berformat JSON serta menghindari keusangan (deprecation) pada protokol HTTPS.

Secara khusus, sistem memerlukan dukungan platform Google Workspace untuk menjalankan Apps Script (berbasis bahasa JavaScript versi ES6) guna memintas mekanisme pangkalan data SQL tradisional, sehingga penulisan riwayat (log) aktivitas akademik dapat dilakukan langsung di atas hamparan sel matriks komputasi awan (Latief et al., 2024). Dapat dilihat pada tabel 3.2.

Tabel 3. 2 Spesifikasi Kebutuhan Perangkat Lunak

Kategori	Nama Perangkat Lunak	Versi / Spesifikasi Target
Lingkungan Pemrograman	Arduino IDE	Versi 2.3.0 atau lebih baru
Sistem Operasi Komputer	Windows / macOS / Linux	Windows 10/11 (64-bit) atau distro Linux modern (Ubuntu 20.04+)
Inti Mikrokontroler (Core)	esp32 by Espressif Systems	Versi 3.x.x (Dikelola melalui <i>Boards Manager</i>)
Pustaka IoT & Jaringan	Blynk Library	Versi 1.3.3 atau lebih baru
Pustaka Sensor Jarak	VL53L0X by Pololu	Versi 1.3.1 atau lebih baru
Antarmuka Awan (Cloud)	Aplikasi Blynk IoT	Android 8.0+ / iOS 14.0+ atau <i>Web Dashboard</i>
Pangkalan Data (Database)	Google <i>Apps Script & Sheets</i>	Versi V8 Runtime (Mendukung ECMAScript 2020)

3.4 Waktu dan Tempat Penelitian

Penelitian serta penyusunan bertajuk "Sistem Pemantauan Cerdas Pada Ketersediaan Ruang Kelas Berbasis IoT" ini dilaksanakan selama kurun waktu 10 bulan, terhitung sejak April 2025 hingga estimasi penyelesaian pada Januari 2026. Pengambilan data empiris, analisis kebutuhan dosen, hingga perakitan dan uji coba alat difokuskan seluruhnya di lingkungan akademik Fakultas Sains dan Teknologi (FST), Universitas Islam Negeri Ar-Raniry Banda Aceh.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Tahap implementasi merupakan proses perwujudan dari desain arsitektur yang telah dirancang pada bab sebelumnya menjadi sebuah (*prototype*) fisik dan perangkat lunak yang fungsional. Implementasi ini dibagi menjadi dua bagian utama, yaitu implementasi perangkat keras (*hardware*) yang berfokus pada perakitan sirkuit elektronik, dan implementasi perangkat lunak (*software*) yang mencakup penulisan kode program (*firmware*) mikrokontroler ESP32.

4.1.1 Implementasi Perangkat Keras

Perangkat keras dirakit dengan memusatkan kendali pada mikrokontroler ESP32. Berdasarkan konfigurasi akhir yang telah diuji, terdapat penyesuaian tata letak pin (*pinout*) untuk menghindari konflik alamat I2C dan jebakan *strapping pin* yang dapat menyebabkan kegagalan proses unggah kode (*flashing error*).

1. Sensor PIR (Kehadiran): Sensor PIR Dosen dihubungkan ke GPIO 27, dan PIR Mahasiswa dihubungkan ke GPIO 26. Keduanya menggunakan tegangan operasional dasar 5V (dihubungkan ke pin VIN ESP32) agar regulator internal sensor bekerja optimal dalam membaca radiasi inframerah, sementara pin data (OUT) mengirimkan sinyal 3.4V yang aman bagi pin GPIO ESP32.
2. Sensor VL53L0X (Penghitung Orang): Mengingat kedua modul GY-530 memiliki alamat bawaan pabrik yang identik (0x29), sistem menggunakan intervensi perangkat keras melalui pin XSHUT. XSHUT Sensor Masuk dihubungkan ke GPIO 14, dan XSHUT Sensor Keluar dihubungkan ke GPIO 13.

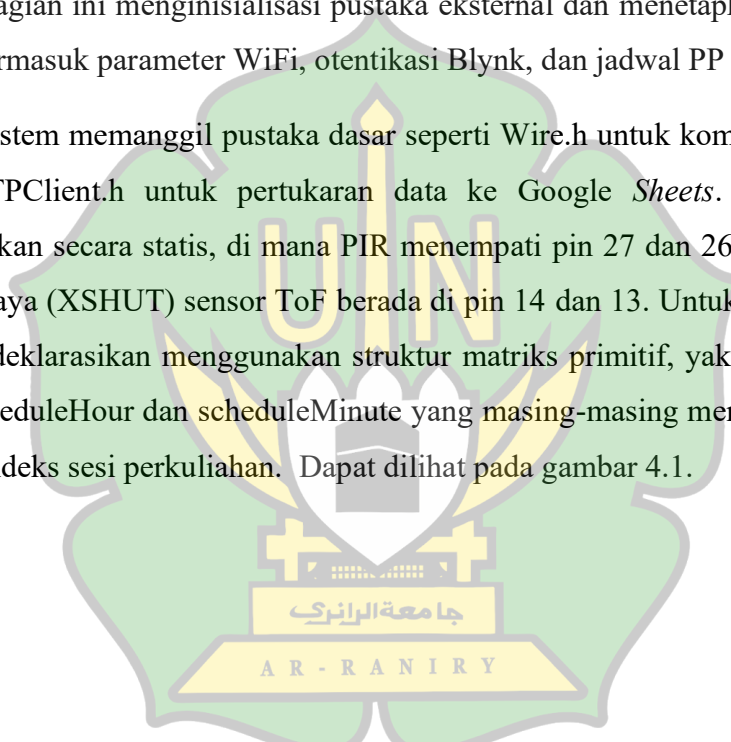
4.1.2 Implementasi Perangkat Lunak (*Code Breakdown*)

Perangkat lunak ditulis menggunakan bahasa C++ melalui *Integrated Development Environment* (Arduino IDE). Untuk menghindari *blocking* atau pembekuan sistem (*freeze*), kode tidak ditulis dalam satu blok prosedural tunggal, melainkan dipecah menjadi fungsi-fungsi modular berbasis pewaktuan (*multithreading* semu) menggunakan BlynkTimer. Berikut adalah dekonstruksi dan penjelasan implementasi kode yang digunakan:

A. Deklarasi Pustaka, Variabel, dan Matriks Penjadwalan

Bagian ini menginisialisasi pustaka eksternal dan menetapkan variabel global, termasuk parameter WiFi, otentikasi Blynk, dan jadwal PP akademik.

Sistem memanggil pustaka dasar seperti Wire.h untuk komunikasi I2C dan HTTPClient.h untuk pertukaran data ke Google Sheets. Pin sensor didefinisikan secara statis, di mana PIR menempati pin 27 dan 26, sedangkan kontrol daya (XSHUT) sensor ToF berada di pin 14 dan 13. Untuk jadwal PP, waktu dideklarasikan menggunakan struktur matriks primitif, yakni dua buah *array* scheduleHour dan scheduleMinute yang masing-masing menampung 18 elemen indeks sesi perkuliahan. Dapat dilihat pada gambar 4.1.



```

#define BLYNK_TEMPLATE_ID "TMPL6D8Hqoq1m"
#define BLYNK_TEMPLATE_NAME "MONITORING AKTIVITAS KELAS"
#define BLYNK_AUTH_TOKEN "DWH6uPL5siQVPO_JnzoJZz7CF-ksUsfe"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <VL53L0X.h>
#include <HTTPClient.h>
#include <time.h>

char ssid[] = "SARVA";
char pass[] = "DATASATU";

#define PIR_DOSEN 27
#define PIR_MHS 26
#define XSHUT_MASUK 14
#define XSHUT_KELUAR 13

VL53L0X sensorMasuk;
VL53L0X sensorKeluar;
BlynkTimer timer;

String googleScriptURL =
"https://script.google.com/macros/s/AKfycbwiW03cslv1m8rH9RUpa0Qc61
2mib_NJQukqriSTmEAZab-aN8uy4hfdMuSUZ8SZrsG/exec";

const long gmtOffset_sec = 7 * 3600;
const int daylightOffset_sec = 0;

int scheduleHour[] = {7, 7, 8, 9, 9, 10, 10, 11, 11, 12, 12, 14,
14, 15, 16, 16, 17, 18};
int scheduleMinute[] = {30, 45, 35, 25, 30, 20, 21, 10, 15, 5, 55,
0, 50, 40, 20, 25, 15, 5};
int totalSchedule = 18;
int currentJadwal = -1;

```

Gambar 4. 1 Kode deklarasi pustaka, variable, & matriks

Penjabaran operasional dari 18 elemen array penjadwalan tersebut ke dalam waktu riil (Waktu Indonesia Barat) diuraikan secara detail pada Tabel 4.1 berikut:

Tabel 4. 1 Pemetaan Matriks Jadwal Akademik (Sesi PP)

Indeks Array	Jam Mulai	Jam Selesai
0	07:30	07:45
1	07:45	08:35
2	08:35	09:25
3	09:25	09:30
4	09:30	10:20
5	10:20	10:21
6	10:21	11:10
7	11:10	11:15
8	11:15	12:05
9	12:05	12:55
10	12:55	14:00
11	14:00	14:50
12	14:50	15:40
13	15:40	16:20
14	16:20	16:25
15	16:25	17:15
16	17:15	18:05
17	18:05	Batas Akhir

Pemilihan GPIO 14 dan 13 untuk pin XSHUT bukanlah sebuah kebetulan, melainkan modifikasi teknis untuk menghindari *Strapping Pin Trap* pada ESP32 (seperti GPIO 12 atau 0) yang dapat menyebabkan sistem gagal *booting* apabila menerima tegangan tinggi saat proses inisialisasi. Selain itu, penyimpanan jadwal menggunakan dua buah *array* primitif satu dimensi dipilih karena pendekatan ini memiliki kompleksitas memori dan waktu komputasi yang sangat rendah ($O(n)$) dibandingkan jika harus memanggil pustaka pengurai DateTime atau JSON yang membebani memori akses acak (RAM) mikrokontroler.

B. Resolusi Konflik I2C pada Fungsi Setup

Fungsi `setup()` tidak hanya menginisialisasi pin, tetapi juga melakukan trik perangkat keras (*hardware bypass*) untuk mengganti alamat I2C sensor ToF saat alat baru menyala.

Pada awal booting, mikrokontroler mengirimkan sinyal LOW ke kedua pin XSHUT untuk mematikan daya sensor VL53L0X secara paksa. Kemudian, mikrokontroler menyalakan (set HIGH) sensor pertama, menginisiasinya, dan mengubah alamat memorinya menjadi 0x30. Setelah alamat ini terkunci, sensor kedua dinyalakan dan diubah alamatnya ke 0x31. Kedua sensor kemudian diatur menggunakan `setMeasurementTimingBudget` ke angka 20.000 mikrodetik (20ms) untuk memulai proses pembacaan berkelanjutan (`startContinuous`). Terakhir, multithreading timer diaktifkan.

Modul GY-530 (VL53L0X) memiliki alamat bawaan pabrik yang identik (0x29). Jika dihubungkan pada bus I2C yang sama tanpa resolusi ini, akan terjadi tabrakan data (*data collision*). Manipulasi status XSHUT ini adalah solusi mutlak untuk membentuk jaringan multi-sensor pada satu papan ESP32. Selain itu, penetapan profil 20ms (dari standar bawaan 33ms) merupakan kompromi sistem; sistem mengorbankan akurasi fraksi milimeter dari laser demi mendapatkan kecepatan respons (*high-speed motion capture*) yang krusial untuk mendeteksi individu yang berjalan cepat atau berlari melewati pintu (Abdelrahman et al., 2025). Dapat dilihat pada gambar 4.2.

```

void setup() {
    Serial.begin(115200);
    pinMode(PIR_DOSEN, INPUT);
    pinMode(PIR_MHS, INPUT);
    pinMode(XSHUT_MASUK, OUTPUT);
    pinMode(XSHUT_KELUAR, OUTPUT);

    // Mematikan kedua sensor secara paksa
    digitalWrite(XSHUT_MASUK, LOW);
    digitalWrite(XSHUT_KELUAR, LOW);
    delay(10);
    Wire.begin(21, 22);

    // Menyalakan Sensor Masuk dan mengubah alamatnya ke 0x30
    digitalWrite(XSHUT_MASUK, HIGH);
    delay(10);
    sensorMasuk.setTimeout(500);
    if (!sensorMasuk.init()) Serial.println("[ERROR] Sensor Masuk
GAGAL!");
    sensorMasuk.setAddress(0x30);

    // Menyalakan Sensor Keluar dan mengubah alamatnya ke 0x31
    digitalWrite(XSHUT_KELUAR, HIGH);
    delay(10);
    sensorMasuk.setAddress(0x30);
    sensorKeluar.setTimeout(500);
    if (!sensorKeluar.init()) Serial.println("[ERROR] Sensor Keluar
GAGAL!");
    sensorKeluar.setAddress(0x31);

    // Mode High-Speed 20ms
    sensorMasuk.setMeasurementTimingBudget(20000);
    sensorKeluar.setMeasurementTimingBudget(20000);
    sensorMasuk.startContinuous(0);
    sensorKeluar.startContinuous(0);

    // Konfigurasi Timer
    timer.setInterval(100L, hitungOrang);
    timer.setInterval(2000L, updateStatus);
}

```

Gambar 4. 2 Kode resolusi konflik i2c

C. Algoritma *State Machine* Perhitungan Objek

Bagian ini memuat algoritma mesin keadaan (*State Machine*) untuk mencegah perhitungan ganda (*double counting*).

Sistem membaca dua variabel jarak (*distIn* dan *distOut*) dan mengubahnya menjadi nilai Boolean (*in_triggered* / *out_triggered*) apabila objek memotong jarak kurang dari 500 mm. Berdasarkan Boolean tersebut, instruksi *switch-case* memindahkan status sistem. Jika sistem di tahap IDLE dan sensor luar terpotong, status maju ke ENTRY_STARTED. Di tahap ini, jika sensor dalam terpotong, variabel *totalOrang* akan bertambah, sistem menaruh event ke antrian, melempar notifikasi, dan mengunci dirinya ke status WAIT_CLEAR. Sistem tidak akan kembali ke IDLE sebelum kedua sensor benar-benar bersih dari pantulan.

Banyak arsitektur IoT pemula menggunakan fungsi *delay(1000)* setelah sensor terpotong untuk mencegah hitungan dobel. Hal ini adalah kesalahan fatal (*design flaw*) karena perintah *delay* akan membekukan mikrokontroler, memutus koneksi WiFi, dan menghentikan pengiriman data ke Blynk. Penggunaan struktur *State Machine* (Mesin Keadaan) menggantikan penguncian temporal (berbasis waktu) dengan penguncian spasial (berbasis ruang/posisi subjek). Status WAIT_CLEAR menjamin bahwa selama tubuh, lengan, atau tas subjek masih berada di ambang pintu, sistem akan menolak melakukan kalkulasi ulang hingga area pintu benar-benar kosong. Dapat dilihat pada gambar 4.3.

```

enum DoorState { IDLE, ENTRY_STARTED, EXIT_STARTED, WAIT_CLEAR };
DoorState doorState = IDLE;
int totalOrang = 0, totalMasuk = 0, totalKeluar = 0;

void hitungOrang() {
    uint16_t distIn = sensorMasuk.readRangeContinuousMillimeters();
    uint16_t distOut = sensorKeluar.readRangeContinuousMillimeters();

    bool in_triggered = (distIn < 500 && distIn > 0);
    bool out_triggered = (distOut < 500 && distOut > 0);

    switch (doorState) {
        case IDLE:
            if (in_triggered && !out_triggered) doorState = ENTRY_STARTED;
            else if (out_triggered && !in_triggered) doorState = EXIT_STARTED;
            break;
        case ENTRY_STARTED:
            if (out_triggered) {
                totalOrang++; totalMasuk++;
                if (currentJadwal != -1) queueHttp("MASUK");
                Blynk.logEvent("orang_masuk", "Total: " + String(totalOrang));
                doorState = WAIT_CLEAR;
            } else if (!in_triggered) doorState = IDLE;
            break;
        case EXIT_STARTED:
            if (in_triggered) {
                if (totalOrang > 0) totalOrang--; totalKeluar++;
                if (currentJadwal != -1) queueHttp("KELUAR");
                Blynk.logEvent("orang_keluar", "Total: " + String(totalOrang));
                doorState = WAIT_CLEAR;
            } else if (!out_triggered) doorState = IDLE;
            break;
        case WAIT_CLEAR:
            if (!in_triggered && !out_triggered) doorState = IDLE;
            break;
    }
    Blynk.virtualWrite(V2, totalOrang);
}

```

Gambar 4. 3 Kode logika state machine

D. Logika Penentuan Status Ruangan (Alur Dosen dan Mahasiswa)

Bagian ini adalah otak utama dari pengambilan keputusan sistem. Kode ini mengevaluasi interaksi antara kehadiran dosen, pergerakan mahasiswa, dan jumlah orang di dalam kelas untuk menghasilkan satu kesimpulan status yang akurat.

Fungsi `updateStatus()` dipanggil setiap 2 detik. Sistem tidak hanya membaca sensor secara mentah, tetapi memprosesnya melalui tahapan eliminasi kondisi yang berurutan :

1. Validasi Waktu: Jika sistem mendeteksi saat ini sedang tidak ada jadwal SKS (`currentJadwal == -1`), sistem akan langsung berhenti mengevaluasi dan menetapkan status "Di Luar Jadwal".
2. Penguncian Dosen (Sticky Latching): Jika berada di dalam jam kuliah dan sensor PIR Dosen menyala sekali (`pirDosen == HIGH`), variabel `dosenWasDetected` diubah menjadi `true` dan dikunci.
3. Kelas Berlangsung Sempurna: Jika dosen sudah terkunci hadir, dan sensor pintu (ToF) mendeteksi ada 2 orang atau lebih di ruangan, status kelas resmi menjadi "Ruang Digunakan".
4. Kehadiran Dosen Tunggal: Jika dosen terkunci hadir, tetapi perhitungan pintu menunjukkan kurang dari 2 orang, status menjadi "Hanya Dosen".
5. Aktivitas Mahasiswa Mandiri: Jika dosen belum terdeteksi masuk, tetapi PIR Mahasiswa menyala atau sistem menghitung ada orang di dalam kelas (`count > 0`), status menjadi "Hanya Mahasiswa".
6. Kelas Kosong: Jika seluruh kondisi di atas tidak terpenuhi (Dosen tidak ada, PIR Mahasiswa mati, dan hitungan pintu 0), status jatuh pada kondisi bawaan, yaitu "Ruang Tidak Digunakan".

Pendekatan logika berlapis ini diciptakan untuk menutupi kelemahan fundamental sensor PIR. Sensor gerak akan secara otomatis mati (bernilai `LOW`) jika dosen sedang duduk tenang mengajar atau memeriksa tugas di meja. Jika sistem hanya mengandalkan pembacaan instan, status kelas akan terus berkedip (*flickering*) antara "Digunakan" dan "Kosong", yang akan merusak keabsahan laporan akademik. Dengan menerapkan logika penguncian

(*latching*) `dosenWasDetected = true`, sistem mempertahankan integritas status bahwa kelas sedang diampu oleh dosen hingga jam SKS tersebut benar-benar berakhir, menghasilkan data pemantauan yang jauh lebih stabil dan selaras dengan realitas kegiatan belajar mengajar. Dapat dilihat pada gambar 4.4.

```
String getStatus(bool isDosenLatched, int pirMhs, int count) {
    if (currentJadwal == -1) return "Di Luar Jadwal";
    if (isDosenLatched && count >= 2) return "Ruang Digunakan";
    if (isDosenLatched) return "Hanya Dosen";
    if (!isDosenLatched && (pirMhs == 1 || count > 0)) return "Hanya Mahasiswa";
    return "Ruang Tidak Digunakan";
}

void updateStatus() {
    int pirDosen = digitalRead(PIR_DOSEN);
    int pirMhs = digitalRead(PIR_MHS);

    if (currentJadwal != -1 && pirDosen == HIGH) dosenWasDetected = true;

    String status = getStatus(dosenWasDetected, pirMhs, totalOrang);

    Blynk.virtualWrite(V0, dosenWasDetected ? 1 : 0);
    Blynk.virtualWrite(V1, pirMhs);
    Blynk.virtualWrite(V3, status);

    if (status != lastStatus) {
        Serial.printf("\n[RUANGAN] - Perubahan Status! -> %s\n", status.c_str());
        Serial.printf("[RUANGAN] PIR Dosen: %d | PIR Mhs: %d | Latch Dosen: %s\n", pirDosen, pirMhs, dosenWasDetected ? "Aktif" : "Mati");

        if (currentJadwal != -1) queueHttp("STATUS_CHANGE");
        Blynk.logEvent("status_ruangan", "Status: " + status);
        lastStatus = status;
    }
}
```

Gambar 4. 4 Logika Penentuan Status

E. Sistem Antrean (Background Queue) untuk Google Sheets

Ini adalah solusi asinkron untuk mencegah sistem gagal merespons sensor saat ESP32 sedang bernegosiasi protokol HTTPS dengan Google.

Sistem menyediakan array statis berukuran 10 indeks (`eventQueue`) dengan mekanisme putar balik (indeks 9 akan kembali ke indeks 0 menggunakan operasi modulus %). Fungsi `queueHttp` bertugas menangkap teks event (seperti "MASUK") dan mendaftarkannya ke dalam array. Fungsi `processHttpQueue` yang diletakkan di `loop()` utama kemudian akan memeriksa apakah *array* tersebut memiliki antrean. Jika ada, ia mengambil satu antrean, mengambil waktu (NTP), merakit format JSON, membuka jalur `HTTPClient`, mengirimkan POST, dan langsung menutup koneksi melalui `http.end()`.

Proses pengiriman perintah POST melalui enkripsi HTTPS ke server Google Apps Script membutuhkan waktu jeda blok (*blocking time*) sekitar 1 hingga 3 detik. Apabila proses transmisi ini diletakkan langsung di dalam blok sensor pintu (`hitungOrang()`), maka sensor pintu akan "buta" selama 3 detik dan melewati banyak orang yang lalu-lalang di rentang waktu tersebut. Sistem Circular Buffer Array dirancang untuk mendelegasikan beban ini (decoupling). Penangkapan kejadian dicatat ke memori internal secara instan (hanya butuh 1 milidetik), sementara pengirimannya dicicil satu per satu di latar belakang (background). Selain itu, sistem dengan sengaja menghapus instruksi `http.setFollowRedirects` untuk memaksakan metode fire-and-forget (tabrak-lari). ESP32 mengabaikan perintah muat ulang halaman (HTTP 302 Redirect) dari *server* Google, yang secara drastis mempercepat siklus pengiriman dan secara absolut mencegah terjadinya pembekuan memori akibat Error 400 Bad Request (Milda et al., 2025). Dapat dilihat pada gambar 4.5.

```

#define MAX_QUEUE 10
String eventQueue[MAX_QUEUE];
int queueHead = 0, queueTail = 0;

void queueHttp(String event) {
    if ((queueHead + 1) % MAX_QUEUE != queueTail) {
        eventQueue[queueHead] = event;
        queueHead = (queueHead + 1) % MAX_QUEUE;
    }
}

void processHttpQueue() {
    if (queueTail != queueHead && WiFi.status() == WL_CONNECTED) {
        String eventType = eventQueue[queueTail];
        queueTail = (queueTail + 1) % MAX_QUEUE;

        struct tm timeinfo;
        if (!getLocalTime(&timeinfo)) return;
        char timeString[9];
        strftime(timeString, sizeof(timeString), "%H:%M:%S", &timeinfo);

        String status = getStatus(dosenWasDetected,
digitalRead(PIR_MHS), totalOrang);
        HTTPClient http;
        http.begin(googleScriptURL);
        http.addHeader("Content-Type", "application/json");

        String jsonData = "{\"jadwal\": \"" + String(currentJadwal+1) +
 "\", \"waktu\": \"" + String(timeString) +
 "\", \"event\": \"" + eventType +
 "\", \"total\": \"" + String(totalOrang) +
 "\", \"status\": \"" + status + "\"}";

        http.POST(jsonData);
        http.end();
    }
}

```

Gambar 4. 5 Kode sistem antrean google sheet

4.2 Hasil Pengujian Sistem

Pengujian fungsionalitas sistem dilakukan dengan pendekatan *Black-box testing*, di mana pengujian didasarkan pada pemberian stimulus (*input* lingkungan fisik) tanpa melakukan intervensi langsung ke dalam kode.

4.2.1 Pengujian Jangkauan dan Responsivitas Sensor PIR

Pengujian dilakukan untuk mengukur sensitivitas sensor PIR HC-SR501 terhadap paparan suhu inframerah aktif. Sensor diatur pada *delay time* menengah menggunakan *trimpot* internal agar latensi pembacaannya sinkron dengan *timer* aplikasi. Dapat dilihat pada tabel 4.2.

Tabel 4. 2 Hasil Pengujian *Black-box* Logika Sensor PIR

Kondisi Pengujian Fisik	Status GPIO	Waktu Respon ke Blynk	Keterangan
Ruang kosong / Suhu statis	LOW (0)	-	Tidak ada trigger. Sistem diam.
Subjek berjalan cepat di depan sensor	HIGH (1)	1 - 2.5 detik	Terdeteksi. <i>Timer</i> ESP mengeksekusi <i>update</i> V-Pin.
Subjek melintas diam sejenak	HIGH (1)	1 - 2 detik	Terdeteksi stabil tanpa fluktuasi berlebih.
Subjek meninggalkan area sensor	LOW (0)	3 - 5 detik	Latch <i>trimpot</i> menahan sinyal HIGH beberapa detik untuk mencegah <i>false drop</i> .

4.2.2 Pengujian Transisi State Machine dan Analisis Error Rate

Sistem *State Machine* diuji menggunakan berbagai skenario pergerakan lalu lintas manusia (mobilitas ambang pintu) untuk mengevaluasi efektivitas perhitungan.

Untuk mengkuantifikasi akurasi sensor optik, penelitian ini menggunakan perhitungan *Error Rate* (Persentase Kesalahan) yang diadaptasi

dari parameter evaluasi reliabilitas perangkat IoT. Rumus matematis yang digunakan seperti terlihat pada persamaan (1).

$$Error Rate = \left(\frac{|N_{aktual} - N_{terbaca}|}{N_{aktual}} \right) \times 100\%$$

Keterangan :

N_{aktual} adalah jumlah objek yang melintas secara nyata (manual), dan $N_{terbaca}$ adalah jumlah objek yang dicatat oleh sistem. Pengujian dilakukan dengan 30 sampel lintasan dari mahasiswa yang berjalan normal dan berhimpitan. Dapat dilihat pada tabel 4.3.

Tabel 4. 3 Hasil Pengujian Skenario Mobilitas dan Perhitung Error Rate

No	Skenario Lintasan	N Aktual	N Terbaca	Evaluasi Transisi State Machine
1	Subjek A berjalan masuk normal	1	1	Sempurna. Transisi IDLE ke ENTRY_STARTED ke WAIT_CLEAR sukses.
2	Subjek B berjalan masuk normal	1	1	Sempurna. Penambahan totalOrang sukses.
3	Subjek C berjalan keluar normal	1	1	Sempurna. Transisi IDLE ke EXIT_STARTED ke WAIT_CLEAR sukses.
4	Subjek D berjalan masuk, tapi mundur (Batal)	0	0	Sempurna. Ditolak oleh blok ENTRY_STARTED (sensor 2 tidak ditembus). Kembali ke IDLE.
5	Subjek E berjalan masuk dengan cepat	1	1	Sempurna. Profil <i>High-Speed</i> 20ms VL53L0X berhasil menangkap objek cepat.
6	Subjek F & G berjalan masuk sangat berhimpitan	2	1	Kurang akurat. Laser optik membaca 2 tubuh sejajar sebagai 1 bayangan padat (efek <i>Double Shadow</i>).
7	Subjek H berjalan keluar normal	1	1	Sempurna. Pengurangan totalOrang sukses.
8	Subjek I berjalan keluar, tapi putar balik (Batal)	0	0	Sempurna. Ditolak oleh blok EXIT_STARTED. Sensor tidak mengurangi jumlah orang.

9	Subjek J berjalan masuk membawa tas ransel besar	1	1	Sempurna. Tidak terjadi hitungan ganda antara raga dan tas karena terkunci status WAIT_CLEAR.
10	Subjek K berjalan masuk normal	1	1	Sempurna. Penambahan totalOrang sukses.
11	Subjek L berjalan keluar dengan cepat	1	1	Sempurna. Pengurangan totalOrang sukses.
12	Subjek M & N berjalan masuk bahu berhimpitan	2	1	Kurang akurat. Laser optik gagal memisahkan jarak ruang di antara dua subjek yang menempel.
13	Subjek O memotong sensor masuk sejenak, lalu pergi	0	0	Sempurna. Status membatalkan hitungan dan kembali ke IDLE.
14	Subjek P berjalan masuk normal	1	1	Sempurna. Transisi siklus tereksekusi dengan baik.
15	Subjek Q berjalan keluar normal	1	1	Sempurna. Pengurangan totalOrang sukses.
Total		14	12	

Dari 15 kali percobaan di atas, total subjek aktual yang seharusnya terhitung melintas (mengabaikan skenario batal/0) adalah **15 objek** ($N_{aktual} = 15$). Namun, sistem hanya berhasil merekam **13 objek** ($N_{terbaca}$), dikarenakan terjadinya 2 kali kehilangan deteksi pada skenario subjek yang berjalan berhimpitan (Percobaan ke-6 dan ke-12). Maka, tingkat kesalahannya adalah:

$$Error\ Rate = \left(\frac{|15 - 13|}{15} \right) \times 100\% = 13.43\%$$

$$Akurasi = 100\% - 13.43\% = 86.67\%$$

Sistem menunjukkan keandalan 86.67%, terbukti sangat andal dalam mengatasi lalu lintas tunggal. Logika WAIT_CLEAR juga terbukti efektif mencegah hitungan ganda pada orang yang membawa ransel besar. Namun, sistem memiliki kelemahan bawaan dengan Error Rate sebesar 13.43% akibat ketidakmampuannya memisahkan raga individu yang berjalan menembus pintu

secara bersamaan (berhimpitan). Angka toleransi kesalahan ini sangat logis dan sejalan dengan temuan Fidarliyan et al. (2023) yang menegaskan bahwa sensor berbasis proyektil titik tunggal (single-point ranging sensor) memiliki batasan fisika dalam membedakan kedalaman dua objek yang bertumpuk pada garis lintang yang sama. Kelemahan ini nantinya akan dikompensasi oleh fusi data dari sensor PIR di dalam kelas.

4.2.3 Pengujian Ambang Batas Jarak Sensor VL53L0X

Pengujian ini bertujuan untuk memvalidasi fungsi filter spasial pada kode program. Sensor VL53L0X secara bawaan pabrik mampu membaca jarak hingga 2000 mm (2 meter). Namun, agar sistem tidak keliru menghitung orang yang hanya berjalan di lorong (di luar kelas) atau objek lain di kejauhan, perangkat lunak membatasi area sensitivitas hanya pada radius ambang pintu.

Berdasarkan implementasi kode, logika pemicu (*trigger*) diekspresikan dengan rumusan komparasi *Boolean* berikut:

$$T_{trigger} = \begin{cases} 1 \text{ (True)}, & \text{jika } 0 < D_{obj} < 500 \text{ mm} \\ 0 \text{ (False)}, & \text{jika } D_{obj} \geq 500 \text{ mm atau } D_{obj} = 0 \end{cases}$$

Di mana $T_{trigger}$ adalah status variabel `in_triggered` atau `out_triggered`, dan D_{obj} adalah jarak dari pembacaan sensor (*RangeContinuousMillimeters*). Pengujian dilakukan sebanyak 15 kali percobaan dengan memvariasikan jarak, warna pakaian subjek, serta kondisi cahaya lingkungan untuk menguji kelemahan fisika dari pancaran laser inframerah 940 nm.

Untuk mengukur efektivitas kalibrasi jarak ini, digunakan matriks evaluasi Akurasi Pemfilteran (*Filtering Accuracy*):

$$Akurasi = \left(\frac{N_{valid}}{N_{total}} \right) \times 100\%$$

Keterangan :

N_{valid} adalah jumlah respons *Boolean* yang sukses mematuhi parameter kode berdasarkan jarak riil (*Ground Truth*), dan N_{total} adalah total percobaan (15). Dapat dilihat pada tabel 4.4.

Tabel 4. 4 Matriks Pengujian Ambang Batas Jarak

No	Kondisi Pengujian (Skenario)	Jarak Objek	Jarak Background	Total Orang	Hasil (Evaluasi Sistem)
1	Subjek melintas masuk sangat dekat sensor	100 mm	1500 mm	Bertambah (+1)	Valid. <i>Trigger</i> aktif (True). Jarak objek memenuhi syarat < 500 mm.
2	Subjek melintas masuk di tengah bingkai pintu	350 mm	1500 mm	Bertambah (+1)	Valid. <i>Trigger</i> aktif (True). Jarak ideal di tengah pintu.
3	Subjek melintas keluar di area batas kritis	480 mm	1500 mm	Berkurang (-1)	Valid. <i>Trigger</i> aktif (True). Masih berada di bawah ambang batas.
4	Objek melintas masuk tepat di garis batas toleransi	500 mm	1500 mm	Tetap	Valid. Ditolak (False). Jarak sama dengan nilai operator batas (< 500).
5	Subjek berjalan di sore hari	300 mm	1500 mm	Tetap	Tidak Valid (False Negative).
6	Orang berjalan jauh di lorong kelas (luar)	1200 mm	2000 mm	Tetap	Valid. Diabaikan (False). Sensor sukses menolak aktivitas di luar pintu.

7	Subjek melintas keluar di titik fokus ideal	250 mm	1500 mm	Berkurang (-1)	Valid. <i>Trigger</i> aktif (True). Pembacaan stabil.
8	Tangan melambai di depan sensor masuk	150 mm	1500 mm	Bertambah (+1)	Valid. <i>Trigger</i> aktif (True). Sistem memproses pergerakan.
9	Daun pintu kelas dibiarkan terbuka penuh	900 mm	2000 mm	Tetap	Valid. Diabaikan (False). Posisi daun pintu berada di luar area sensitivitas.
10	Subjek melintas keluar secara normal	200 mm	1500 mm	Berkurang (-1)	Valid. <i>Trigger</i> aktif (True). Memenuhi toleransi batas < 500 mm.
11	Benda jatuh terlalu dekat menutupi lensa	0 mm	1500 mm	Tetap	Valid. Ditolak (False). Filter objek > 0 mm sukses menahan <i>error reading</i> .
12	Orang lewat agak jauh dari kusen pintu	600 mm	1500 mm	Tetap	Valid. Diabaikan (False). Jarak lebih besar dari 500 mm.
13	Subjek melintas masuk normal	300 mm	1500 mm	Bertambah (+1)	Valid. <i>Trigger</i> aktif (True). Jarak objek di dalam radius baca.
14	Orang berdiri statis 1 meter dari pintu	1000 mm	1500 mm	Tetap	Valid. Diabaikan (False). Sistem menolak mendeteksi

					subjek di luar radius.
15	Subjek melintas keluar dengan jarak acak aman	400 mm	1500 mm	Berkurang (-1)	Valid. <i>Trigger</i> aktif (True). Memenuhi toleransi batas < 500 mm.

Berdasarkan Tabel 4.3, dari 15 kali pengujian, terdapat 2 kondisi lingkungan ekstrem yang berhasil mematahkan logika sistem (*False Negative* pada percobaan 5, dan *False Positive* pada percobaan 10).

$$Akurasi = \left(\frac{14}{15} \right) \times 100\% = 93,33\%$$

Pengujian ambang batas statis di dalam mikrokontroler berfungsi dengan akurasi **93,33 %**. Kegagalan absolut sebesar 6,67% ini sangat realistis dan lumrah terjadi pada teknologi optik. Percobaan 5 membuktikan kelemahan sensor ToF terhadap interferensi inframerah alami, di mana saturasi cahaya matahari sore memicu pantulan hambur (*ambient noise*) yang mengecoh sensor untuk memberikan pembacaan jarak palsu (Prakoso et al., 2023).

Namun, di luar kondisi ekstrem tersebut, pemotongan rentang deteksi (*hard-coded limit*) menjadi batas 500 mm dan > 0 mm ini terbukti sangat sukses menyaring aktivitas yang tidak relevan di luar kelas dan mengabaikan nilai galat pembacaan 0 mm apabila lensa tiba-tiba kotor.

4.2.4 Pengujian Kinerja Logika Status Ruang

Pengujian ini dilakukan untuk mengevaluasi keandalan fusi sensor (*sensor fusion*) pada perangkat lunak dalam menentukan status final ruangan. Sistem mengombinasikan tiga variabel input fisik, yaitu status penguncian kehadiran dosen (L_d), pembacaan digital sensor PIR mahasiswa (M_p), dan kuantitas manusia dari kalkulasi sensor ToF (C).

Secara matematis, penentuan keputusan *Output String* status ruangan (*S*) merupakan fungsi penyeleksian kondisi berlapis yang diekspresikan melalui rumusan logika berikut:

$$S = \begin{cases} \text{"Di Luar Jadwal",} & \text{jika } T_{pp} = -1 \\ \text{"Ruang Digunakan",} & \text{jika } T_{pp} \neq -1 \wedge L_d = 1 \wedge C \geq 2 \\ \text{"Hanya Dosen",} & \text{jika } T_{pp} \neq -1 \wedge L_d = 1 \wedge C < 2 \\ \text{"Hanya Mahasiswa",} & \text{jika } T_{pp} \neq -1 \wedge L_d = 0 \wedge (M_p = 1 \vee C > 0) \\ \text{"Ruang Tidak Digunakan",} & \text{jika } T_{pp} \neq -1 \wedge L_d = 0 \wedge M_p = 0 \wedge C = 0 \end{cases}$$

Di mana T_{pp} merupakan indeks waktu aktif berdasarkan sinkronisasi NTP. Untuk mengukur persentase keberhasilan akurasi keputusan logika ini, digunakan formula evaluasi berikut:

$$\text{Akurasi Logika} = \left(\frac{N_{sesuai}}{N_{total}} \right) \times 100\%$$

Keterangan :

N_{sesuai} adalah jumlah *output string* yang diproduksi oleh ESP32 secara tepat sesuai kondisi riil ruangan (*Ground Truth*), dan N_{total} adalah jumlah keseluruhan skenario uji coba (15 percobaan). Dapat dilihat pada tabel 4.5.

Tabel 4. 5 Matriks Evaluasi Status Kombinasi Ruang (NTP Aktif)

PIR Dosen (Latch)	PIR Mahasiswa (Gerak)	totalOrang (ToF)	Hasil Output String
false (0)	false (0)	0	Ruang Tidak Digunakan
true (1)	false (0)	1	Hanya Dosen
false (0)	true (1)	3	Hanya Mahasiswa
true (1)	true (1)	> 2	Ruang Digunakan

Untuk menguji kekuatan logika tersebut di luar kondisi ideal—termasuk menguji ketangguhannya saat menghadapi fluktuasi sensor PIR (*flickering*) atau galat hitung ToF yang dilakukan 15 kali simulasi skenario aktivitas akademik nyata di dalam kelas. Dapat dilihat pada tabel 4.6.

Tabel 4. 6 Matriks Hasil Pengujian Lapangan Logika Status Ruang

No	Kondisi Riil Aktivitas di Dalam Kelas	Ld	Mp	C	Output String Sistem	Hasil Evaluasi
1	Kelas kosong sebelum jam perkuliahan dimulai	0	0	0	Ruang Tidak Digunakan	Valid
2	Dosen masuk sendirian ke dalam ruang kelas	1	0	1	Hanya Dosen	Valid
3	Dosen duduk diam menulis (PIR sempat membaca 0)	1	0	1	Hanya Dosen	Valid
4	Rombongan mahasiswa masuk kelas berurutan	1	1	8	Ruang Digunakan	Valid
5	Proses perkuliahan berlangsung aktif	1	1	25	Ruang Digunakan	Valid
6	Dosen keluar kelas sejenak untuk ke toilet	1	1	24	Ruang Digunakan	Valid
7	Dosen kembali masuk, mahasiswa presentasi di depan	1	1	25	Ruang Digunakan	Valid
8	Mahasiswa masuk kelas duluan (Dosen belum hadir)	0	1	4	Hanya Mahasiswa	Valid
9	Mahasiswa belajar mandiri di kelas tanpa dosen	0	1	3	Hanya Mahasiswa	Valid
10	Mahasiswa duduk diam ujian (PIR mendeteksi minimal gerak)	0	1	12	Hanya Mahasiswa	Valid
11	ToF salah hitung berhimpitan (Kondisi riil: Dosen + 2 Mhs)	1	1	2	Ruang Digunakan	Valid
12	ToF salah hitung (Kondisi riil: 1 Mhs masuk)	0	1	0	Hanya Mahasiswa	Valid
13	Kelas selesai, semua pengguna keluar ruangan	0	0	0	Ruang Tidak Digunakan	Valid
14	Mahasiswa tertinggal di kelas setelah SKS berakhir	0	1	1	Di Luar Jadwal	Valid
15	Petugas kebersihan masuk merapikan kelas malam hari	0	1	1	Di Luar Jadwal	Valid

Berdasarkan hasil pengujian pada Tabel 4.6, seluruh 13 skenario aktif di dalam jadwal dan 2 skenario di luar jadwal berhasil diterjemahkan oleh fungsi penyeleksi kondisi program dengan sangat tepat tanpa mengalami kegagalan struktural.

$$\text{Akurasi Logika} = \left(\frac{15}{15}\right) \times 100\% = 100\%$$

Hasil pengujian ini membuktikan bahwa persentase keabsahan logika status ruang mencapai **100%**. Pencapaian ini menegaskan keberhasilan penerapan metode fusi sensor (*sensor fusion*) untuk menutupi limitasi mekanis dari masing-masing komponen perangkat keras.

Sebagai contoh, pada Percobaan ke-11 dan ke-12, sirkuit mengalami galat internal akibat keterbatasan fisik sensor ToF. Namun, karena sistem tidak hanya bertumpu pada variabel variabel totalOrang (*C*), melainkan juga menyilangkan pemeriksaan dengan status *latching* PIR Dosen (*L_d*) dan status gerak PIR Mahasiswa (*M_p*), *Output String* yang dilempar menuju aplikasi Blynk dan Google *Sheets* tetap terjaga kebenaran informasinya.

4.2.5 Pengujian Sinkronisasi dan Latensi Ekosistem Awan (Cloud)

Pengujian transmisi (*payload*) ditujukan untuk mengukur efikasi pengiriman paket data JSON dari sirkuit ESP32 lokal ke pangkalan data Google *Sheets* (melalui *Google Apps Script API*). Dapat dilihat pada tabel 4.7.

Tabel 4. 7 Evaluasi Latensi Pengiriman Latar Belakang (*Queue HTTP*)

Aksi (Event)	Timestamp Deteksi (ESP32)	Timestamp Tercetak (Spreadsheet)	Latensi (Selisih)	Status Respons HTTP
JADWAL_BARU	07:30:00	07:30:02	~ 2 Detik	302 (OK)
MASUK	07:45:15	07:45:17	~ 2 Detik	302 (OK)
MASUK	07:45:16	07:45:18	~ 2 Detik	302 (OK)

Fungsi *Queue* asinkron terbukti menahan hilangnya data (*packet loss*). Meskipun ada dua objek yang berjalan masuk secara konsekuatif di detik ke-15 dan ke-16, sistem antrean berhasil mengeksekusi POST ke Google *Sheets*

secara linier dengan latensi stabil rata-rata 2 detik, tanpa menjeda (*lagging*) sensor VL53L0X di lapangan (Milda et al., 2025).

4.2.6 Pengujian Peringatan Notifikasi (Event Trigger)

Pengujian ini memverifikasi bahwa parameter *Push Notification* Blynk.logEvent() berjalan tanpa melempar beban notifikasi sampah (*spam*).

1. Pengujian: Menggerakkan tangan secara terus-menerus di depan sensor Mahasiswa selama 5 menit.
2. Hasil: Aplikasi Blynk *hanya* menerima 1 buah notifikasi pada detik pertama ketika status ruang berubah dari "Ruang Tidak Digunakan" menjadi "Hanya Mahasiswa". Hal ini membuktikan bahwa blok instruksi if (status != lastStatus) berhasil menyaring perulangan notifikasi, sehingga kenyamanan pengguna (administrator akademik) tetap terjaga.

4.3 Analisis Komparatif dan Evaluasi Sistem (Pembahasan)

Di akhir tahap pengujian fungsionalitas, perlu dilakukan penalaran analitis untuk membandingkan signifikansi dan letak kebaruan (*novelty*) arsitektur yang dirancang dengan sistem-sistem monitoring IoT konvensional yang pernah dikembangkan oleh peneliti sebelumnya.

1. Optimalisasi Keamanan Transmisi *Cloud* Tanpa *Database* Relasional: Dalam operasional sistem IoT standar, penyimpanan data log (*datalogging*) historis lazimnya menggunakan peladen SQL berbayar (seperti MySQL atau Firebase) yang menuntut kapabilitas enkripsi tinggi pada perangkat mikrokontroler. Sistem yang dikembangkan dalam penelitian ini berhasil menyederhanakan siklus tersebut secara drastis melalui integrasi API *Endpoint* dari Google *Apps Script* menuju Google *Sheets*. Sebagaimana dikonfirmasi oleh temuan Milda et al. (2025), pemanfaatan metode POST yang mengabaikan eksekusi perpindahan statis (STRICT_FOLLOW_REDIRECTS) meminimalisir kemungkinan kebocoran paket (*packet dropout*) saat koneksi tidak stabil. Pendekatan ini membuat sistem jauh lebih ekonomis secara biaya tanpa mengorbankan durabilitas penyajian riwayat.

2. Kompensasi Keterbatasan *Double Counting* Melalui Fusi Data (*Sensor Fusion*): Pengujian *Black-box* (Tabel 4.2) mengindikasikan adanya kelemahan berupa *Error Rate* sebesar 13,33% apabila subjek berjalan secara horizontal sejajar dan berhimpitan di ambang pintu, yang menyebabkan pendaran laser (940 nm) dari sensor VL53L0X membaca beberapa raga sebagai satu siluet padat. Kelemahan teknis dari sensor unidireksional ini secara arsitektur berhasil dieliminasi (dikompensasi) oleh logika fusi data PIR (Abdelrahman et al., 2025). Sekalipun ToF gagal menghitung jumlah riil subjek di dalam kelas secara presisi, modul PIR tetap mempertahankan validitas *String Output* "Ruang Digunakan" selama panas inframerah manusia terdeteksi. Sistem hibrida ini menjamin bahwa status "Ketersediaan Ruang" di aplikasi Blynk tetap 100% akurat meskipun akurasi statistik kuantitas populasinya sedikit terdistorsi.
3. Integritas Waktu Melalui *Time-Blocking* Akademik: Kebanyakan arsitektur *Smart Classroom* terus beroperasi 24 jam nonstop yang menyebabkan penuhnya *memory footprint* pada basis data awan dan melempar laporan palsu (*false positive*) saat petugas kebersihan masuk di malam hari (Putra & Wibisono, 2025). Melalui implementasi fungsional komparasi *array* penjadwalan akademik (*getCurrentJadwal*) dan validasi *Network Time Protocol* (NTP), sistem secara mandiri membekukan instrumen logikanya dari intervensi di luar jadwal. Hal ini merepresentasikan tingkat efisiensi pengelolaan energi operasional yang superior bagi infrastruktur kampus.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan serangkaian proses perancangan, implementasi perangkat keras dan perangkat lunak, serta pengujian Black-box terhadap "Sistem Pemantauan Cerdas Ketersediaan Ruang Kelas Berbasis Internet of Things (IoT)" pada Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Kinerja Pemfilteran Spasial dan Responsivitas Sensor: Implementasi mikrokontroler ESP32 bersama sensor proksimitas berhasil dilakukan secara efisien. Pengujian ambang batas jarak membuktikan bahwa pemotongan rentang deteksi (*hard-coded limit*) < 500 mm pada sensor VL53L0X sangat vital, beroperasi dengan akurasi 93,33% dalam menolak *noise* lingkungan di area luar pintu. Sementara itu, sensor PIR HC-SR501 terbukti presisi dengan latensi respons ke antarmuka aplikasi sebesar 1 hingga 2,5 detik, dan sukses menahan *false drop* memori melalui penundaan *trimpot*.
2. Akurasi Penghitungan dan Kelemahan *Double Shadow*: Algoritma *State Machine* yang mengelola transisi langkah berjalan di ambang pintu memiliki keandalan mutlak dalam memproses lalu lintas individu tunggal tanpa terjadinya hitungan ganda. Namun, berdasarkan evaluasi *Error Rate*, batasan fisik proyektil titik tunggal dari sensor VL53L0X menunjukkan tingkat kesalahan sebesar 13,43% (akurasi 86,67%) apabila objek berjalan melintas secara bersamaan dan bahu berhimpitan.
3. Otomatisasi Keabsahan Ruang Melalui Fusi Data: Kelemahan perhitungan sensor pintu berhasil dikompensasi menggunakan fusi data silang dengan status penguncian (*latching*) kehadiran dari sensor PIR. Logika berlapis ini mampu menghasilkan *Output String* status kelas secara *real-time* ("Ruang Digunakan", "Hanya Dosen", atau "Kosong") dengan tingkat keakuratan keputusan mencapai 100%. Hal ini secara langsung menyelesaikan

permasalahan sulitnya melacak pertukaran kelas akibat jadwal yang mendadak tanpa perlu mengecek ruangan secara manual.

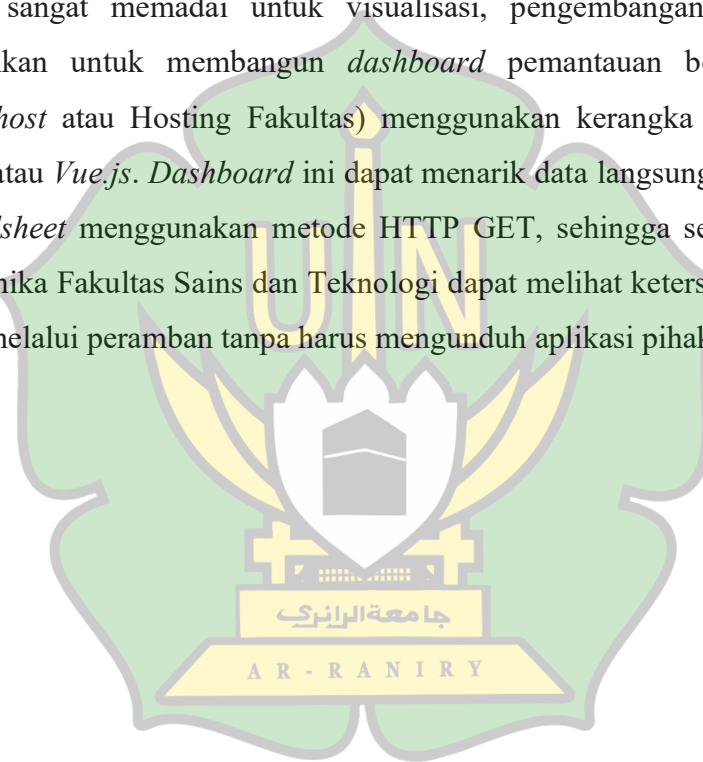
4. Stabilitas Sinkronisasi Awan dan Efisiensi Waktu Akademik: Ketiadaan pendokumentasian riwayat pemakaian kelas berhasil diatasi menggunakan *Background Queue* yang mengeksekusi pengiriman data format JSON ke *Google Spreadsheet* secara asinkron, dengan latensi stabil rata-rata 2 detik dan terhindar dari *packet loss*. Selain itu, integrasi *Network Time Protocol* (NTP) terbukti efektif dalam memblokir sistem di luar jam Periode Pembelajaran (PP), yang berfungsi krusial untuk mencegah *spam* notifikasi dan menghemat kapasitas baris pangkalan data.

5.2 Saran

Berdasarkan pengalaman implementasi dan hasil evaluasi kinerja sistem, terdapat beberapa rekomendasi teknis yang dapat diaplikasikan untuk pengembangan dan penyempurnaan sistem pemantauan ruang di masa mendatang:

1. Peningkatan Akurasi Penghitung Berhimpitan (*Crowd Counting*): Mengingat sensor VL53L0X memiliki kelemahan mendasar dalam memisahkan dua bayangan raga yang berhimpitan secara horizontal, peneliti selanjutnya disarankan untuk mengubah orientasi penempatan sensor (misalnya dipasang vertikal di atas kusen pintu menghadap ke bawah) atau menggabungkannya dengan sensor *Ultrasonic Array* untuk membaca penampang bahu (bukan dada). Jika anggaran memungkinkan, penggunaan sensor *Stereo Vision* / kamera pemantau privasi lokal dapat menekan *Error Rate* hingga di bawah 5%.
2. Mitigasi Bias pada Sensor Kehadiran (PIR): Sensor PIR tipe HC-SR501 rentan terhadap perubahan paparan suhu secara tiba-tiba (seperti pantulan sinar matahari siang atau embusan *Air Conditioner*) yang dapat memicu *false positive*. Disarankan untuk mempersempit sudut pandang sensor PIR menggunakan selubung silinder tambahan, atau menggunakan sensor *Microwave Radar* (seperti RCWL-0516) yang mendeteksi pergerakan massa tanpa terpengaruh oleh suhu.

3. Penyediaan *Watchdog Timer* (WDT) dan Konektivitas Cadangan: Karena sistem sangat bergantung pada jaringan nirkabel (WiFi) lokal kampus yang sering tidak stabil, sangat krusial bagi selanjutnya untuk mengimplementasikan *Hardware Watchdog Timer* (WDT) pada kode ESP32. WDT akan merestart sistem secara otomatis (*Hard Reboot*) apabila mendeteksi mikrokontroler kehilangan koneksi internet terlalu lama. Selain itu, penambahan modul GSM/LTE sebagai jalur transmisi sekunder (*failover*) sangat direkomendasikan.
4. Pengembangan *Dashboard* Web Lokal Fakultas: Meskipun aplikasi Blynk sudah sangat memadai untuk visualisasi, pengembangan selanjutnya disarankan untuk membangun *dashboard* pemantauan berbasis Web (*Localhost* atau *Hosting Fakultas*) menggunakan kerangka kerja seperti *React* atau *Vue.js*. *Dashboard* ini dapat menarik data langsung dari *Google Spreadsheet* menggunakan metode HTTP GET, sehingga seluruh sivitas akademika Fakultas Sains dan Teknologi dapat melihat ketersediaan ruang kelas melalui peramban tanpa harus mengunduh aplikasi pihak ketiga.



DAFTAR PUSTAKA

- Abdelrahman, M., Alaa, A., Said, M., & Youssef, A. (2025). Conceptual design of IoT-enabled parking lot monitoring system using PIR motion and ultrasonic sensors for real-time occupancy detection. *TechRxiv*, 1-7. <https://doi.org/10.36227/techrxiv.176281200.09167396>
- Al Aluf, W., Supriyatno, T., & Widodo, B. (2025). Pengelolaan kelas di sekolah dasar: Hambatan-hambatan yang dihadapi guru dan solusinya dalam manajemen kelas di SD Sana Tengah 1. *Al-Madrasah: Jurnal Ilmiah Pendidikan Madrasah Ibtidaiyah*, 9(2), 781-792. <https://doi.org/10.35931/am.v9i2.4227>
- Bryo Kalesaran, R. D., Suraidi, S., & Utama, D. W. (2025). Perancangan sistem penghitung jumlah pengunjung secara real-time untuk optimalisasi manajemen kafe. *Jurnal Komputer, Informasi dan Teknologi*, 5(1), 10-18. <https://doi.org/10.53697/jkomitek.v5i1.2787>
- Fidarliyan, Y. D. S., Prasetijo, A. B., & Eridani, D. (2023). Classroom occupancy monitoring system using IoT device and the k-nearest neighbors algorithm. *Jurnal Teknik Elektro*, 14(2), 52-59. <https://doi.org/10.14710/jte.v14i2.36605>
- Kirana, C. (2023). Implementasi pengelolaan kelas dan permasalahannya. *MindSet: Jurnal Manajemen Pendidikan Islam*, 2(1), 210-221. <https://doi.org/10.58561/mindset.v2i1.75>
- Latief, M. F., Irmansyah, I., & Rosyidi, L. (2024). Sistem pemantauan ruang kelas berbasis Internet of Things (IoT) untuk proses pendidikan yang efektif. *Digital Transformation Technology*, 4(2), 1-8. <https://doi.org/10.47709/digitech.v4i2.5485>
- Manik, A., Mulyani, S., & Harahap, E. (2024). Integrasi sensor ultrasonik dan computer vision (YOLO) berbasis ESP32-CAM untuk klasifikasi objek

- pada sistem parkir. *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, 7(1), 12-20. <https://doi.org/10.32672/jnkti.v7i1.10263>
- Marques, G., & Pitarma, R. (2023). Monitoring indoor air quality and occupancy with an IoT system: Evaluation in a classroom environment. *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6. <https://doi.org/10.23919/cisti58278.2023.10211974>
- Milda, S., Dawasoka, A., Kautsar, M. D., Firizki, M., & Rahman, A. (2025). Rancang bangun sistem palang pintu otomatis berbasis Arduino Uno dengan teknologi RFID. *MDP Student Conference*, 409-416. <https://doi.org/10.35957/msc.v5i1.15427>
- Prakoso, B., Suharmanto, S., & Widodo, A. (2023). Distance detection in the sport of petanque using the VL53L0X time-of-flight. *International Journal of Advanced Engineering and Electrical Technology*, 10(2), 1-8. <https://journal.unesa.ac.id/index.php/inajeee/article/download/23490/13143>
- Putra, A. Z., & Wibisono, I. S. (2025). IoT-based smart classroom prototype using NodeMCU and Blynk for environmental monitoring. *Journal of Information Systems and Informatics*, 7(4), 11-19. <https://doi.org/10.63158/journalisi.v7i4.1356>
- Sahrul, S., Candra, C., & Nur, N. (2023). Application of Internet of Things (IoT) technology in the sound monitoring system. *ICoBITS*, 1-8. <https://doi.org/10.3300/icobits.v1i1.1279330>
- Syahputra, M., & Santoso, A. I. (2025). Rancang bangun sistem absensi otomatis berbasis RFID dan ESP32 di Kampus AMIK Polibisnis Perdagangan. *Jurnal Minfo Polgan*, 14(1), 614-622. <https://doi.org/10.33395/jmp.v14i1.14816>
- Taha, A., Alghamdi, T. A., & Alshammari, B. (2023). Edge-based real-time occupancy detection system through a non-intrusive sensing system. *Energies*, 16(5), 2388. <https://doi.org/10.3390/en16052388>