

## LAPORAN SMART ROOM OCCUPANCY SYSTEM

### KELOMPOK 1



Anggota Kelompok:

1. Muhammad Irsyad RM (20230140227)
2. Hanifatul Nadiva (20230140203)
3. Hafizh Harimurti (20230140216)
4. Muhammad Abdullah Faqih (20230140229)
5. Andhika Prima Hutama (20230140239)
6. Asyura Azzahra Djola (20230140231)
7. Tasnim Fadilah Anwar(20230140240)
8. Firyal Shafa Salsabila (20230140224)
9. Panji Pranantya Imadulhaq (20230140230)

UNIVERSITAS MUHAMMADIYAH YOGYAKARTA

FAKULTAS TEKNIK

TEKNOLOGI INFORMASI

2025/2026

## BAB 1

### Pendahuluan

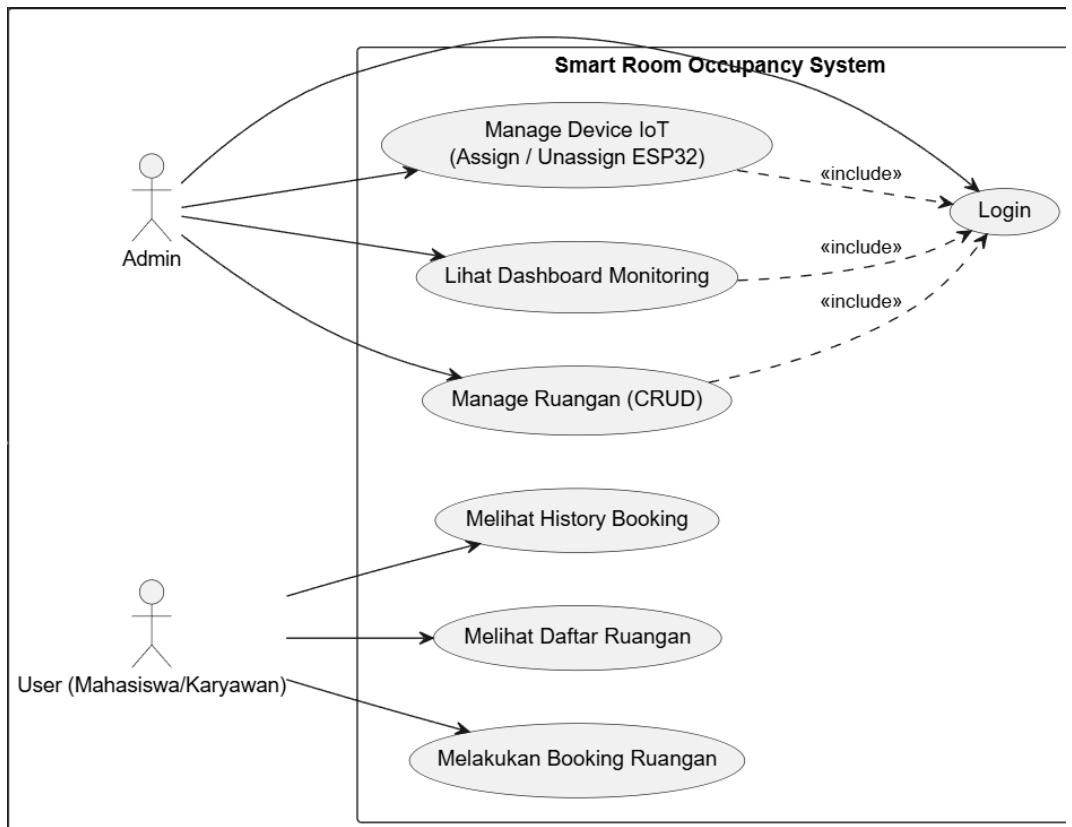
Perkembangan teknologi *Internet of Things* (IoT) telah memungkinkan otomasi pengelolaan fasilitas fisik secara *real-time*. Salah satu implementasi yang krusial adalah pengelolaan ruangan pada gedung perkantoran atau kampus. Sistem Smart Room Occupancy hadir sebagai solusi untuk mengoptimalkan penggunaan ruangan melalui integrasi perangkat keras (sensor) dan perangkat lunak (web/mobile). Dengan sistem ini, status ketersediaan ruangan tidak lagi dikelola secara manual, melainkan didasarkan pada deteksi aktivitas fisik yang akurat[1].

## BAB 2

### PlantUML

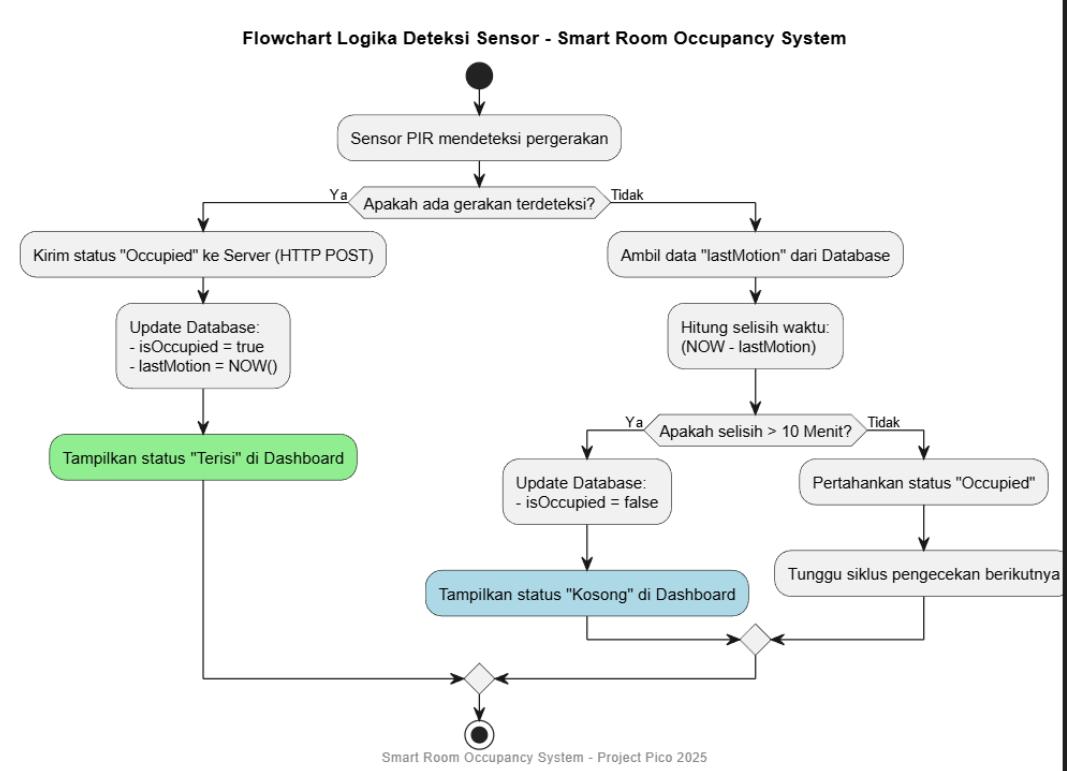
#### 1. USECASE DIAGRAM

Use Case Diagram ini menggambarkan fungsionalitas utama dari sistem *Smart Room* berbasis IoT. Sistem ini memfasilitasi dua jenis pengguna (Admin dan User) untuk berinteraksi dengan layanan peminjaman ruangan dan pemantauan ketersediaan ruangan secara otomatis menggunakan sensor.



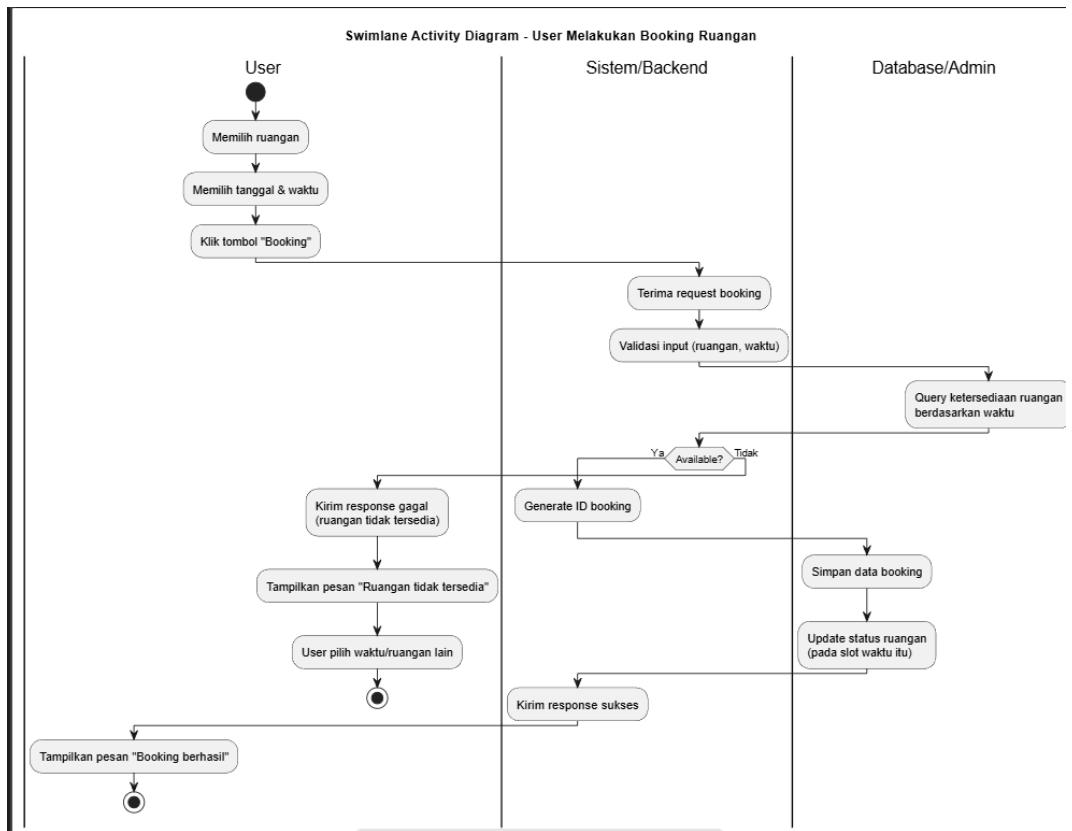
## 2. FLOWCHART DIAGRAM

Flowchart sistem digunakan untuk menjelaskan alur logika proses deteksi okupansi ruangan berdasarkan data dari sensor PIR. Diagram ini menggambarkan bagaimana sistem menentukan status ruangan berdasarkan aktivitas yang terdeteksi.



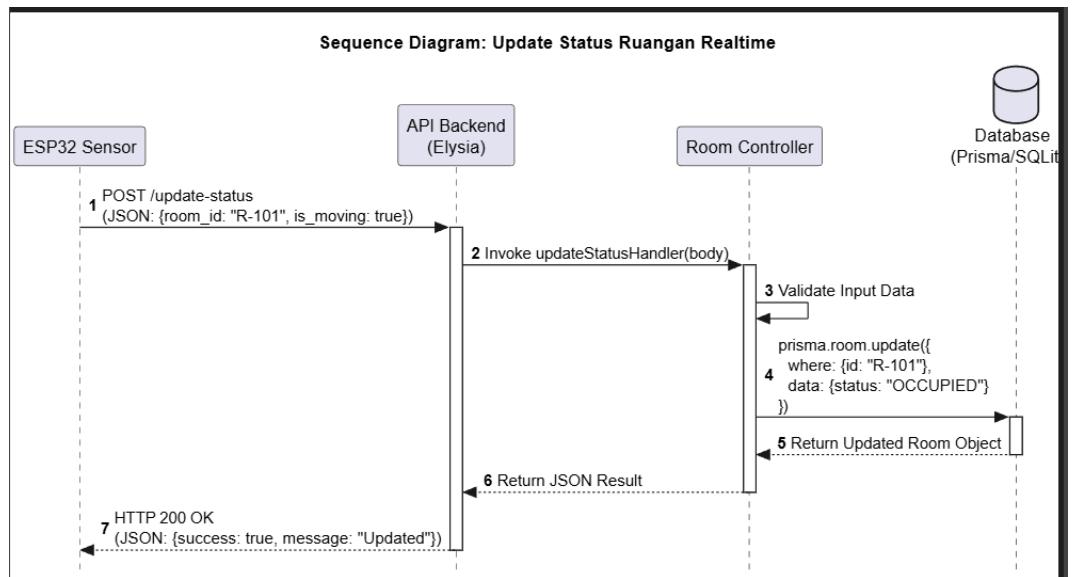
### 3. ACTIVITY DIAGRAM

Activity Diagram sistem terpadu menggambarkan alur proses bisnis secara menyeluruh pada skenario User melakukan booking ruangan. Diagram ini menggunakan pendekatan *swimlane* untuk menunjukkan pembagian tanggung jawab antar entitas.



#### 4. SEQUENCE DIAGRAM

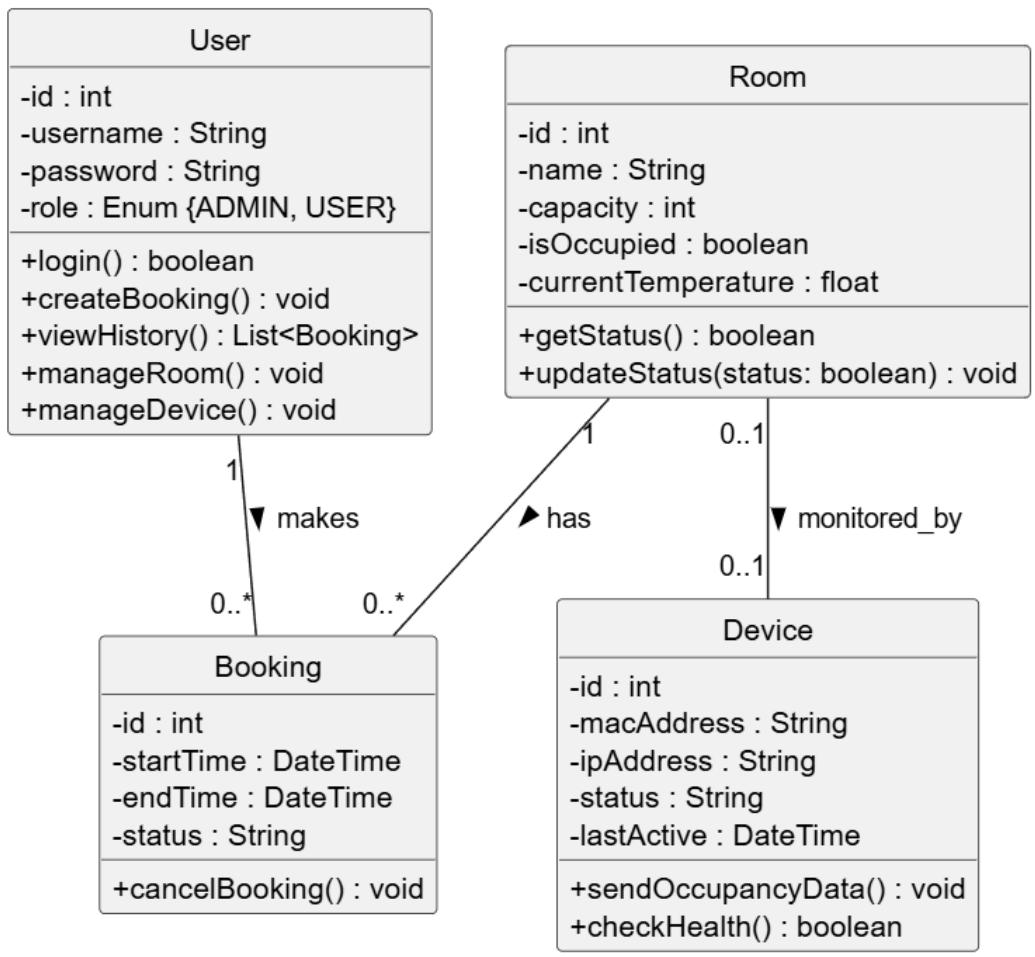
Sequence Diagram ini menggambarkan interaksi objek-objek dalam sistem saat terjadi perubahan kondisi fisik di dalam ruangan. Diagram ini memodelkan skenario "Update Status Realtime", di mana sensor ESP32 mengirimkan data okupansi (terisi/kosong) ke server untuk diperbarui di *database*. Tujuan utama dari alur ini adalah memastikan data yang tampil di *Dashboard* pengguna selalu sinkron dengan kondisi fisik ruangan yang sebenarnya.



## 5. CLASS DIAGRAM

Class Diagram ini menggambarkan struktur data dan logika sistem *Smart Room Occupancy*. Diagram ini memodelkan bagaimana data pengguna, ruangan, jadwal peminjaman, dan perangkat IoT (ESP32) saling terhubung untuk mendukung fitur *monitoring* dan *booking*.

### Class Diagram: Smart Room Occupancy System

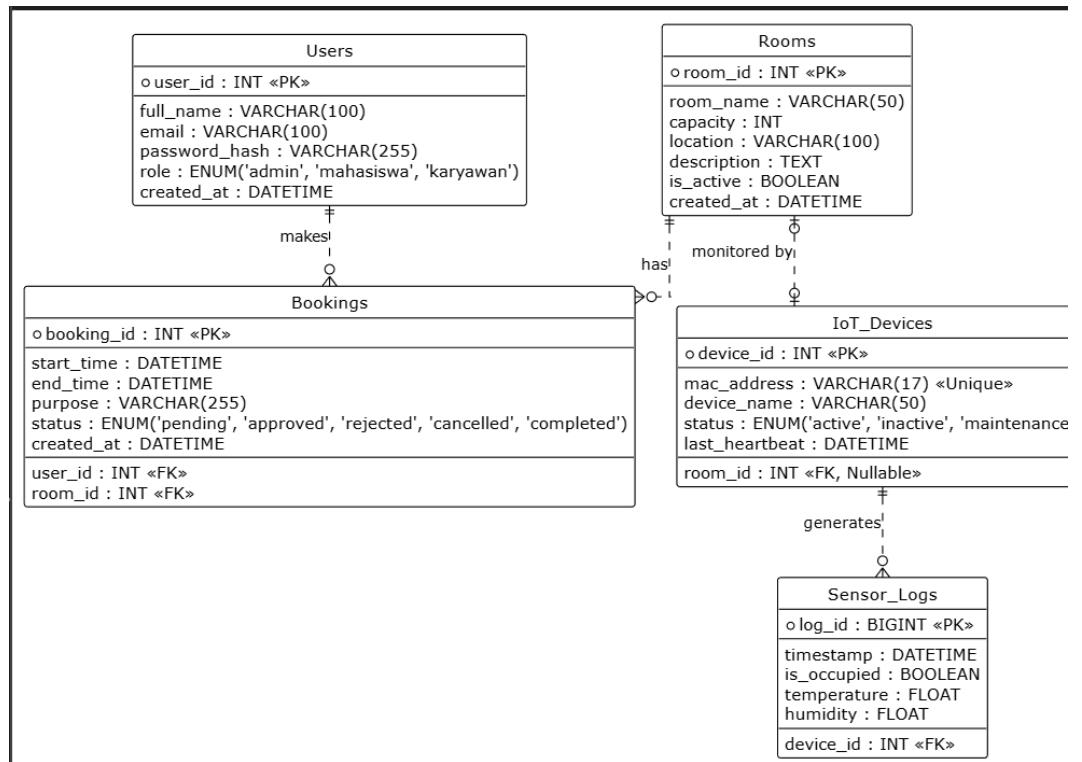


#### 6. ERD

Entity Relationship Diagram (ERD) digunakan untuk menggambarkan desain basis data sistem Smart Room Occupancy System. Diagram ini menunjukkan struktur tabel, primary key, foreign key, serta hubungan antar entitas.

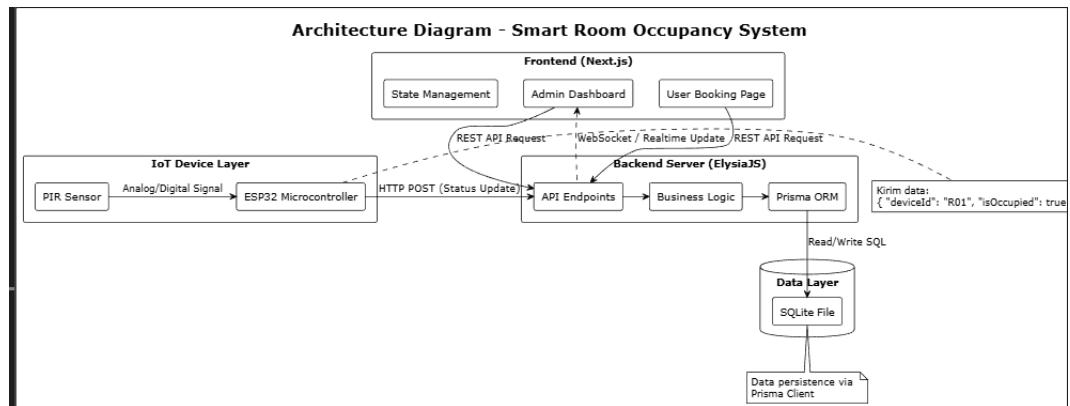
- Tabel **User** menyimpan data pengguna sistem.
- Tabel **Room** menyimpan data ruangan dan informasi status okupansi berdasarkan sensor.
- Tabel **Booking** menyimpan data pemesanan ruangan yang berelasi dengan User dan Room.

Relasi antara User dan Booking bersifat one-to-many, begitu pula relasi antara Room dan Booking. Struktur database ini telah dinormalisasi dan sesuai dengan implementasi Prisma menggunakan SQLite.



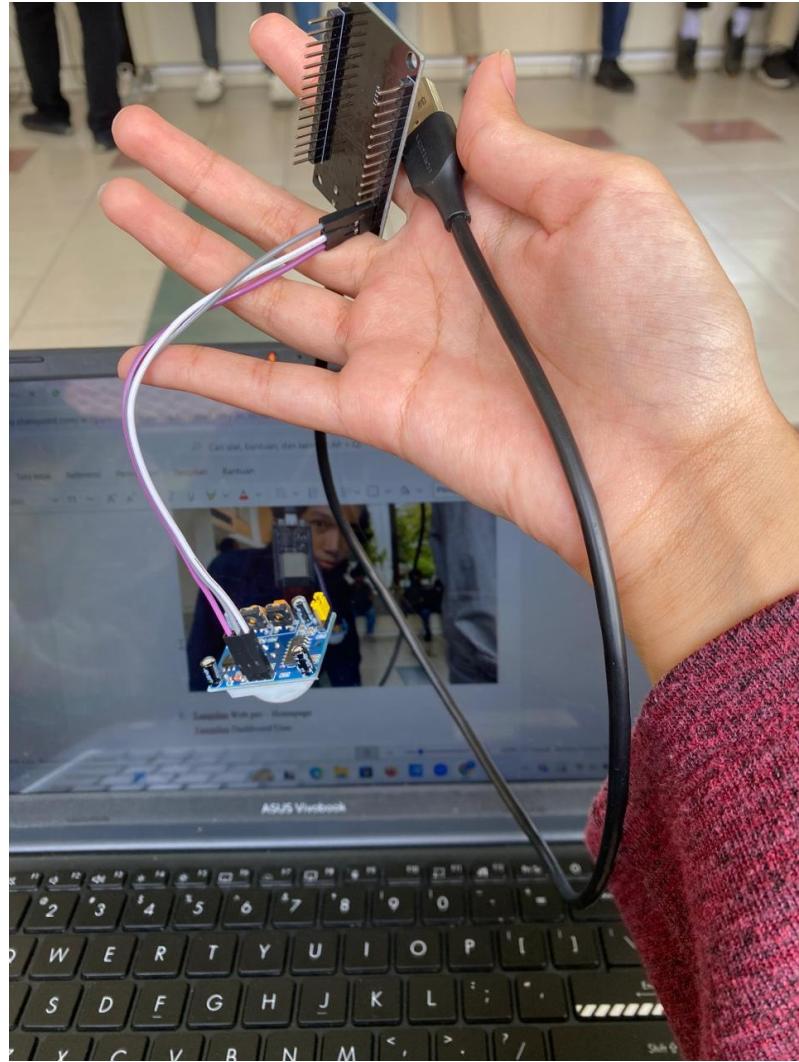
## 7. ARSITEKTURE DIAGRAM

Arsitektur sistem menggambarkan susunan komponen utama dan alur komunikasi antar layer sistem. Sistem terdiri dari empat lapisan utama, yaitu perangkat IoT, backend server, database, dan client application. Arsitektur ini dirancang untuk memastikan pemisahan tanggung jawab (*separation of concern*), kemudahan pemeliharaan, dan kemampuan pengembangan sistem di masa depan.



## BAB 3

### 1. Photo Alat



## 2. Tampilan Web per – Homepage

### Tampilan Dashboard User

The screenshot shows the 'Dashboard User' for the 'Sistem Manajemen Ruangan Berbasis IoT (Port 8000)'. It features a dark header with the title 'SMART ROOM' and a navigation bar with 'Monitoring' and 'Reservasi' buttons. Below the header, there are three colored cards: a blue card for 'Total Ruangan' (2), a green card for 'Tersedia' (2), and a red card for 'Terisi' (0). A section titled 'Status Ruangan Realtime' displays two rooms: 'baca1' (Kapasitas: 1 orang, Tersedia) and 'ruang12' (Kapasitas: 1 orang, Tersedia). Both rooms show their last movement at 2:09:09 AM.

The screenshot shows the 'Dashboard User' interface. It includes a 'Buat Booking Baru' (Create New Booking) form with fields for 'Pilih Ruangan' (Select Room), 'Waktu Mulai' (Start Date mm/dd/yyyy), and 'Waktu Selesai' (End Date mm/dd/yyyy). A 'Konfirmasi Booking' (Confirm Booking) button is at the bottom. To the right, a 'Riwayat Booking Anda' (Your Booking History) section lists past bookings: 'baca1' from 1/14/2026, 1:58:00 AM - 2:59:00 AM, and 'ruang12' from 1/14/2026, 2:07:00 AM - 2:09:00 AM.

### Tampilan Dashboard Admin

The image displays three vertically stacked screenshots of a web-based monitoring application for a 'Smart Room' system. The top two screenshots show the 'Dashboard Monitoring' page, while the bottom one shows the 'Dashboard User' page.

**Dashboard Monitoring (Top Screenshot):**

- Header:** Dashboard Monitoring, Status Real-time & CRUD System.
- Widgets:**
  - Sensor Aktif: 1/1 (Icon: Blue wave)
  - Terdeteksi Gerakan: 1 Ruangan (Icon: Green person)
  - Sensor Offline: 0 Unit (Icon: Red circle with exclamation)
- Data Card:** baca1 (ID: 2 • ESP32-ROOM-001). Status: ADA ORANG (Red background). Last Update: 02.03.31. Progress: 100%.

**Dashboard Monitoring (Middle Screenshot):**

- Header:** Dashboard Monitoring, Status Real-time & CRUD System.
- Widgets:**
  - Sensor Aktif: 0/2 (Icon: Blue wave)
  - Terdeteksi Gerakan: 0 Ruangan (Icon: Green person)
  - Sensor Offline: 2 Unit (Icon: Red circle with exclamation)
- Data Cards:**
  - baca1 (ID: 2 • ESP32-ROOM-001) - OFFLINE (Status: OFFLINE), Last Update: 02.04.19, Progress: 100%
  - ruang12 (ID: 3 • esp32) - OFFLINE (Status: OFFLINE), Last Update: 02.04.04, Progress: 100%

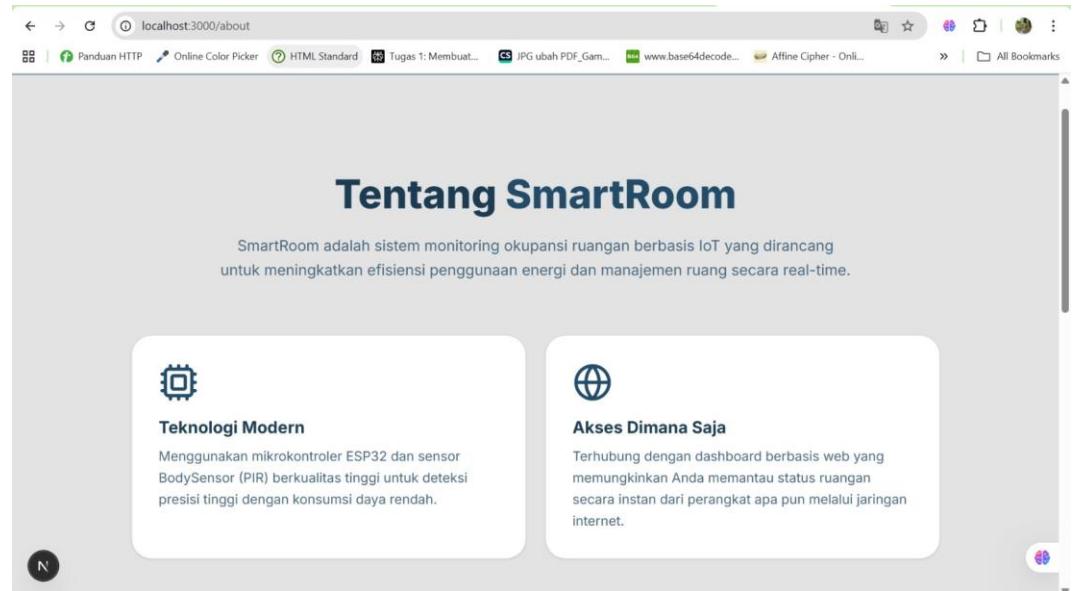
**Dashboard User (Bottom Screenshot):**

- Header:** SMART ROOM, Dashboard User, Sistem Manajemen Ruangan Berbasis IoT (Port 8000).
- Summary Metrics:**
  - Total Ruangan: 2
  - Tersedia: 1
  - Terisi: 1
- Status Ruangan Realtime:**
  - baca1: Kapasitas: 1 orang, Sedang Digunakan (Red icon), Last Movement: 2/07/00 AM
  - ruang12: Kapasitas: 1 orang, Tersedia (Green icon), Last Movement: 2/04/04 AM
- Logout Button:** Logout (Bottom right corner).

Tampilan halaman home



Tampilan halaman about



Tampilan register

**Register Akun**  
Masuk untuk memonitor ruangan Anda

**Nama Lengkap**  
Nama Lengkap

**Email**  
email@contoh.com

**Password**  
Minimal 8 karakter

**Konfirmasi Password**  
Ulangi password

Role akun otomatis sebagai **User**

**Daftar Sekarang →**

Tampilan halaman login

**Selamat Datang**  
Masuk untuk memonitor ruangan Anda

**Email**  
email@contoh.com

**Password**  
\*\*\*\*\*

**Masuk Sekarang →**

Belum punya akun? [Daftar Sekarang](#)

### 3. Penjelasan Susunan Folder dan Source Code

#### 1. Struktur Direktori Sistem

Pada tahap implementasi, sistem yang dikembangkan memiliki struktur direktori yang terbagi menjadi beberapa bagian utama, yaitu frontend, backend, dan program Arduino. Pembagian ini bertujuan untuk memisahkan tanggung jawab setiap komponen agar sistem lebih terstruktur, mudah dikembangkan, dan mudah dipelihara.

## 2. Implementasi Frontend

Frontend sistem dikembangkan menggunakan framework Next.js dengan pendekatan App Router, di mana setiap folder merepresentasikan rute atau halaman aplikasi.

### 2.1 Folder app

Folder app berfungsi sebagai pusat pengaturan halaman dan tampilan antarmuka pengguna. Setiap subfolder di dalamnya merepresentasikan halaman tertentu yang dapat diakses oleh pengguna.

Struktur folder app terdiri dari:

- about : halaman informasi sistem
- login : halaman autentikasi pengguna
- dashboard : halaman dashboard pengguna
- dashboardAdmin : halaman dashboard khusus administrator

### 2.2 File layout.tsx

File layout.tsx digunakan sebagai template utama aplikasi. File ini berfungsi untuk menampung komponen yang bersifat global seperti navbar dan footer, sehingga tidak perlu ditulis ulang pada set.

### 2.3 File globals.css

File globals.css digunakan untuk menyimpan gaya tampilan (CSS) global yang diterapkan ke seluruh halaman aplikasi.

## 3. Implementasi Komponen Antarmuka

Untuk meningkatkan efisiensi pengembangan, sistem menggunakan komponen yang dapat digunakan kembali (reusable components) yang disimpan pada folder components.

Folder ini terdiri dari:

- navbar.tsx : komponen navigasi utama
- panelnavbar.tsx : navigasi khusus dashboard atau admin
- footer.tsx : komponen bagian bawah halaman

Komponen-komponen tersebut dipanggil pada file layout.tsx sehingga tampil secara konsisten di seluruh halaman.

#### 4. Implementasi Backend

Backend sistem dikembangkan menggunakan **Node.js** dengan dukungan **TypeScript** serta **Prisma ORM** sebagai penghubung antara aplikasi dan database.

##### 4.1 Folder prisma

Folder prisma digunakan untuk pengelolaan database. File schema.prisma berisi definisi model tabel, relasi antar tabel, serta tipe data yang digunakan. Database yang digunakan pada tahap pengembangan adalah **SQLLite** dengan file dev.db.

##### 4.2 Folder src/lib

Folder src/lib berisi konfigurasi koneksi database. File db.ts digunakan untuk menginisialisasi Prisma Client agar dapat digunakan oleh seluruh endpoint backend.

##### 4.3 Folder src/routes

Folder routes berisi kumpulan endpoint API yang menangani logika bisnis sistem, antara lain:

- users.ts : pengelolaan data pengguna dan autentikasi
- admin.ts : pengelolaan fitur khusus administrator
- rooms.ts : pengelolaan data ruangan
- booking.ts : pengelolaan pemesanan ruangan
- motion.ts : pengelolaan data sensor pergerakan

##### 4.4 File index.ts

File index.ts merupakan entry point backend yang berfungsi untuk menjalankan server dan menghubungkan seluruh endpoint API yang tersedia.

#### 5. Implementasi Sistem Arduino

Sistem ini juga terintegrasi dengan perangkat Arduino yang berfungsi sebagai pengumpul data sensor. Program Arduino disimpan dalam folder arduino dengan nama file smart\_room.ino.

File ini berisi logika pembacaan sensor serta pengiriman data ke server backend untuk selanjutnya diproses dan ditampilkan pada dashboard sistem.

#### 6. Alur Kerja Sistem

- Pengguna mengakses aplikasi melalui antarmuka frontend.
- Frontend mengirimkan permintaan data ke backend melalui API.
- Backend memproses permintaan dan berinteraksi dengan database.
- Data sensor dari Arduino dikirimkan ke backend.
- Data ditampilkan pada dashboard sesuai dengan hak akses pengguna.

## BAB 4

### Kesimpulan

Smart Room Occupancy System berhasil dikembangkan sebagai sistem berbasis Internet of Things (IoT) yang mampu memantau ketersediaan ruangan secara real-time. Sistem ini memanfaatkan sensor PIR dan mikrokontroler ESP32 untuk mendeteksi aktivitas di

dalam ruangan sehingga status okupansi dapat diperbarui secara otomatis tanpa pengelolaan manual.

Implementasi sistem menggunakan framework Next.js sebagai frontend serta Node.js dan Prisma ORM sebagai backend yang terhubung dengan basis data SQLite. Integrasi antar komponen tersebut memungkinkan proses monitoring dan pemesanan ruangan berjalan secara efisien serta meningkatkan efektivitas pengelolaan ruangan, sehingga sistem ini berpotensi diterapkan pada lingkungan kampus maupun perkantoran.

#### Daftar Putsaka

- [1] A. Daissaoui, A. Boulmakoul, L. Karim, and A. Lbath, “IoT and Big Data Analytics for Smart Buildings: A Survey,” *Procedia Computer Science*, vol. 170, pp. 161–168, 2020, doi: 10.1016/j.procs.2020.03.021.

#### Lampiran