

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

Terdapat beberapa landasan teori yang menurut penulis mempunyai korelasi yang kuat dengan penelitian yang akan dilakukan, diantara adalah :

2.1.1 Kebakaran Indoor dan Kebutuhan Evakuasi Dinamis

Kebakaran pada bangunan *indoor* memiliki tingkat risiko tinggi karena ruang tertutup membatasi sirkulasi, mempercepat akumulasi asap, dan menyulitkan proses evakuasi. Pada banyak kejadian, ancaman utama bukan hanya api, tetapi juga asap, berkurangnya visibilitas, serta gas beracun yang memengaruhi kemampuan penghuni untuk mengenali arah keluar. Karena itu, keputusan evakuasi harus dibuat cepat dan harus tetap aman ketika kondisi lingkungan berubah dari menit ke menit (Bai et al., 2025; Huang et al., 2024).

Sistem evakuasi statis masih banyak digunakan pada bangunan. Sistem ini umumnya menetapkan jalur keluar sejak tahap desain gedung. Kelemahannya, jalur tersebut tidak selalu aman saat kebakaran benar-benar terjadi. Jalur terpendek dapat berubah menjadi jalur paling berbahaya ketika asap menyebar, suhu meningkat, atau koridor tertentu tertutup. Karena itu, pendekatan evakuasi modern perlu berpindah dari *static route* ke *dynamic safe route*, yaitu rute yang terus diperbarui sesuai kondisi lapangan (Bai et al., 2025; Mirahadi & McCabe, 2021).

Dengan demikian, masalah utama pada evakuasi kebakaran *indoor* bukan hanya menemukan jalan keluar, tetapi memilih jalan keluar yang aman, dapat diakses, dan tetap relevan ketika bahaya berubah. Konsep ini menjadi dasar bagi penelitian yang menggabungkan sensor IoT, pemetaan bahaya, dan algoritma pembelajaran adaptif.

2.1.2 IoT untuk Deteksi dan Pemantauan Kebakaran Indoor

Internet of Things (IoT) adalah jaringan perangkat fisik yang saling terhubung untuk mengumpulkan, mengirim, dan memproses data. Pada konteks *smart*

building, IoT diposisikan sebagai sumber data *real-time* untuk indikator bahaya kebakaran *indoor*, termasuk sensor-sensor yang merepresentasikan kondisi termal dan asap di ruang tertutup (Shaharuddin et al., 2023). Studi optimasi evakuasi berbasis IoT juga menunjukkan bahwa data sensor dapat dipakai untuk mengestimasi lokasi/penyebaran api serta memberi “sinyal bahaya” bagi modul perencanaan rute (Zhang et al., 2024). Tantangan teoritis yang muncul adalah *reliabilitas* data (ketahanan perangkat, blind spots, dan keterlambatan), karena kualitas *hazard zoning* dan keputusan rute sangat bergantung pada kualitas observasi (Ding et al., 2024; Shaharuddin et al., 2023).

Keunggulan utama IoT pada sistem keselamatan adalah kemampuannya menyediakan observasi terdistribusi. Artinya, sistem tidak hanya menerima satu sinyal alarm, tetapi memperoleh banyak data dari banyak titik ruang. Dengan data ini, sistem dapat mengetahui area mana yang mulai berbahaya, area mana yang masih aman, dan bagaimana perubahan bahaya bergerak dari waktu ke waktu. Tinjauan bibliometrik terbaru juga menunjukkan bahwa IoT untuk keselamatan kebakaran berkembang ke arah deteksi, lokalisasi, dan dukungan evakuasi, bukan sekadar alarm konvensional (AlQahtani et al., 2025).

2.1.3 Kriteria Zona Berbahaya

Zona bahaya pada bangunan indoor dapat dipetakan berdasarkan parameter yang direkomendasikan dalam studi risiko kebakaran. Parameter umum meliputi :

- 1) Konsentrasi Asap (ppm / AQI) (Bai et al., 2025; Xu et al., 2022).
 - a) < 100 ppm → aman
 - b) 100–300 ppm → waspada
 - c) 300 ppm → berbahaya (mengurangi *visibilitas* dan meningkatkan risiko *inhalasi* gas beracun)
- 2) Tingkat Jarak Pandang (meter) (Bai et al., 2025).
 - a) 10m → aman
 - b) 5–10 m → terbatas
 - c) < 5 m → sangat berbahaya

- 3) Temperatur Ruangan ($^{\circ}\text{C}$) (Ding et al., 2024)
- a) $< 50^{\circ}\text{C} \rightarrow$ aman
 - b) $50\text{--}80^{\circ}\text{C} \rightarrow$ risiko sedang
 - c) $80^{\circ}\text{C} \rightarrow$ bahaya tinggi (risiko *heat stress*)

Untuk kebutuhan penelitian ini, hanya menggunakan konsentrasi asap.

2.1.4 Pemetaan Zona Berbahaya secara Real-time

Pemetaan zona berbahaya adalah proses mengubah data lingkungan menjadi representasi *spasial* tentang tingkat risiko pada suatu bangunan. Dalam konteks kebakaran, zona ini dapat dibagi menjadi area aman, area waspada, dan area berbahaya. Pembagian dilakukan berdasarkan parameter yang relevan, seperti suhu, jarak pandang, konsentrasi asap, dan kedekatan terhadap sumber bahaya. Penelitian terbaru pada *dynamic fire risk assessment* juga menggunakan parameter seperti temperatur, jarak pandang, dan gas beracun untuk mengukur risiko dalam ruang diskret (Bai et al., 2025).

Pada penelitian ini, pemetaan zona bahaya dijadikan jembatan antara lapisan sensor dan lapisan pengambilan keputusan. Data dari sensor umumnya terlebih dahulu dibersihkan dan dinormalisasi. Setelah itu, setiap node atau grid pada peta bangunan diberi nilai risiko. Nilai inilah yang kemudian menjadi dasar bagi algoritma *Q-learning* untuk menghindari area yang berbahaya. Prinsip serupa juga terlihat pada penelitian *BIM-based fire assessment* yang menggabungkan parameter risiko, pembobotan, dan *visualisasi* untuk mendukung analisis spasial kebakaran (Ding et al., 2024; Terzi et al., 2025).

2.1.5 Representasi Bangunan sebagai Graf atau Grid

Agar bangunan dapat diproses oleh algoritma, tata ruang perlu diubah ke bentuk model komputasi. Dua representasi yang paling umum adalah graf dan grid. Pada model graf, ruang direpresentasikan sebagai node dan koneksi antar-ruang direpresentasikan sebagai edge. Pada model grid, bangunan dibagi ke dalam sel-sel kecil yang masing-masing memiliki status tertentu, misalnya aman, terhalang, atau

berbahaya. Pemodelan seperti ini umum dipakai dalam perencanaan rute dan simulasi evakuasi (Mirahadi & McCabe, 2021; Xu et al., 2022).

Representasi ini sangat penting karena *Q-learning* bekerja pada *state* dan *action* yang jelas. Jika bangunan dimodelkan sebagai grid, maka *state* dapat berupa posisi agen pada satu sel. *Action* dapat berupa gerak ke atas, bawah, kiri, kanan, atau diagonal. Bila bangunan dimodelkan sebagai graf, maka *state* adalah node tertentu dan *action* adalah perpindahan ke node tetangga. Pemilihan model akan memengaruhi kompleksitas pelatihan, ukuran Q-table, dan kecepatan konvergensi.

Untuk penelitian tesis, model grid sering lebih mudah dijelaskan dan diuji pada tahap prototipe. Model ini juga sesuai dengan integrasi zona bahaya *real-time*, karena setiap sel dapat diberi nilai risiko yang langsung diperbarui dari data sensor (Xu et al., 2022). Juga menunjukkan bahwa grid *environment* efektif untuk eksperimen *Q-learning* pada perencanaan rute dalam kondisi berbahaya atau beresiko.

2.1.6 Reinforcement Learning

Reinforcement Learning (RL) adalah pendekatan pembelajaran mesin di mana agen belajar melalui interaksi dengan lingkungan. Agen memilih aksi, menerima *reward*, lalu memperbaiki kebijakan agar *reward* jangka panjang menjadi maksimum. RL sangat cocok untuk masalah pengambilan keputusan berurutan dan lingkungan yang berubah, karena agen tidak hanya menghitung jalur sekali, tetapi belajar memilih tindakan terbaik dari pengalaman berulang (Sutton & Barto, 2018).

Komponen utama RL meliputi *agent*, *environment*, *state*, *action*, dan *reward*. Dalam penelitian ini, *agent* dapat diartikan sebagai pengambil keputusan rute. *Environment* adalah model bangunan dan kondisi bahayanya. *State* menyatakan posisi agen dan situasi lokal. *Action* menyatakan perpindahan ke node lain. *Reward* menyatakan kualitas keputusan, misalnya negatif jika melewati area berbahaya dan positif jika mendekati pintu keluar dengan aman. Konsep ini sesuai untuk evakuasi karena masalahnya memang berupa rangkaian keputusan dari awal hingga keluar gedung.

Kelebihan RL dibanding metode statis adalah kemampuannya beradaptasi terhadap struktur *reward* yang kompleks. Sistem tidak hanya mengejar rute terpendek, tetapi dapat diarahkan untuk menyeimbangkan beberapa tujuan, seperti keselamatan, waktu, dan penghindaran area berisiko.

2.1.7 Q-learning

Q-learning adalah algoritma RL berbasis nilai. Algoritma ini menyimpan nilai utilitas dari pasangan *state-action* dalam sebuah tabel yang disebut Q-table. Tujuan utamanya adalah memperkirakan seberapa baik suatu aksi dilakukan pada *state* tertentu jika agen ingin mencapai hasil terbaik di masa depan (Sutton & Barto, 2018; Watkins & Dayan, 1992).

Pembaruan Q-learning dinyatakan dengan rumus berikut:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

Keterangan:

$Q(s, a)$ = nilai aksi a pada state s

α = learning rate

r = reward sesaat

γ = discount factor

s' = state berikutnya

$\max_{a'} Q(s', a')$ = nilai aksi terbaik pada state berikutnya

Gambar 2-1 Rumus Q-Learning

Berikut penjelasan pada masing-masing parameter yang ada di rumus :

- 1) *State* (s), merupakan parameter yang representasi kondisi lingkungan yang diamati oleh agen, *state* dapat berupa posisi agen pada grid peta 2D, misalkan posisi (0,0) representasi state 0 (S0), saat pindah ke posisi (0,1) representasi state 1 (S1), pindah ke posisi (0,2) representasi state 2 (S2) dan selanjutnya
- 2) *Action* (a), merupakan tindakan yang dapat dilakukan agen pada setiap *state*, pada penelitian ini *action* yang dapat dilakukan agen hanya ada 4 yaitu : atas, bawah, kiri dan kanan, maksudnya pada setiap state agen dapat memilih *action* selanjutnya yaitu bergerak ke atas, bawah, kiri atau kanan.
- 3) *Reward* (r), merupakan nilai yang diberikan setelah agen melakukan aksi, *reward* ini merujuk pada Gambar 2-2 Rumusan *Reward* pada Rute Evakuasi

- 4) *Learning Rate* (α), menentukan seberapa besar pengalaman baru mempengaruhi nilai Q , nilainya berkisar dari 0 dan 1
- 5) *Discount Factor* (γ), mengatur seberapa penting *reward* masa depan dibanding *reward* saat ini, nilainya berkisar dari 0 dan 1
- 6) *Eksploration Rate* (ϵ), digunakan pada metode ϵ -greedy untuk menentukan apakah agen mengeksplorasi jalur baru atau menggunakan pengalaman terbaik yang sudah dipelajari

Rumus ini menunjukkan bahwa nilai lama diperbarui menggunakan *reward* saat ini dan estimasi *reward* masa depan. Jika *reward* dirancang baik, *Q-learning* akan cenderung memilih rute yang lebih aman dan efisien. Dan sampai sekarang masih menjadi dasar bagi banyak penelitian perencanaan rute (Watkins & Dayan, 1992).

Untuk masalah evakuasi, desain *reward* menjadi sangat penting, salah satu contoh rumus *reward* penelitian adalah :

$$r_t = \begin{cases} +100, & \text{jika mencapai pintu keluar} \\ -100, & \text{jika masuk zona sangat berbahaya} \\ -10 \times R_i, & \text{jika melewati node berisiko} \\ -1, & \text{untuk setiap langkah} \end{cases}$$

Gambar 2-2 Rumus Reward pada Rute Evakuasi

Rumus ini dapat membantu agen mempelajari tiga hal sekaligus: mencapai tujuan, menghindari bahaya, dan tidak berputar terlalu lama. Model seperti ini sesuai dengan pendekatan *multi-objective evacuation* yang menggabungkan keselamatan dan efisiensi waktu (Xu et al., 2022).

2.1.8 Integrasi IoT dan *Q-learning* pada Sistem Evakuasi

Integrasi IoT dan *Q-learning* dapat dijelaskan sebagai alur tiga lapis. Lapis pertama adalah *sensing*, yaitu pembacaan data lingkungan. Lapis kedua adalah *hazard mapping*, yaitu mengubah data sensor menjadi peta zona bahaya. Lapis

ketiga adalah *decision layer*, yaitu agen *Q-learning* yang memilih rute evakuasi berdasarkan peta bahaya terbaru.

Secara konseptual, integrasi ini memberi nilai tambah yang tidak dimiliki sistem alarm biasa. IoT menyediakan informasi bahaya aktual. *Q-learning* menyediakan mekanisme adaptif untuk pengambilan keputusan rute. Ketika data bahaya berubah, *state* lingkungan ikut berubah. Perubahan ini akan memengaruhi pilihan aksi agen. Dengan demikian, sistem yang dihasilkan bukan hanya mendeteksi kebakaran, tetapi juga meresponsnya dengan rekomendasi rute yang lebih aman. Studi terbaru di area *IoT-enabled fire safety* juga menekankan pentingnya integrasi sensor, jaringan, *machine learning*, dan *real-time decision support* (AlQahtani et al., 2025).

Sistem evakuasi cerdas umumnya diuji melalui simulasi. Simulasi diperlukan karena pengujian langsung pada kebakaran nyata memiliki risiko tinggi dan sulit dikendalikan. Di sisi lain, simulasi memungkinkan peneliti membandingkan beberapa skenario sumber api, penyebaran asap, konfigurasi sensor, dan metode pencarian rute secara sistematis. Banyak penelitian evakuasi modern juga bergantung pada simulasi untuk menguji performa model sebelum implementasi lebih lanjut (Ding et al., 2024; Senanayake et al., 2024).

2.1.9 Sensor Asap dan Sensor Suhu

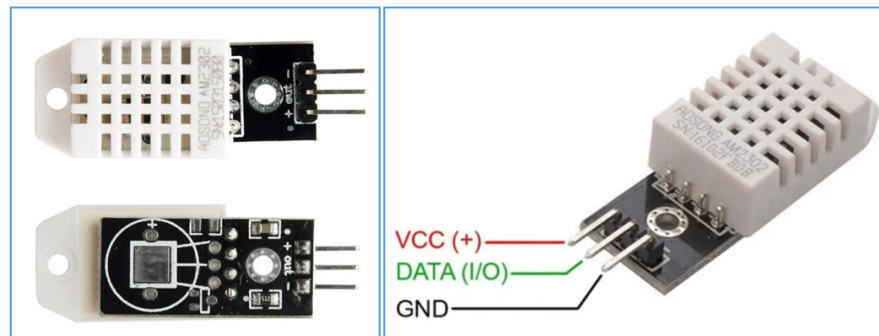
Sensor merupakan perangkat yang digunakan untuk mendeteksi perubahan kondisi fisik maupun kimia pada lingkungan dan mengubahnya menjadi sinyal yang dapat diproses oleh sistem elektronik. Dalam sistem *Internet of Things* (IoT), sensor berfungsi sebagai komponen utama pada lapisan akuisisi data yang menyediakan informasi kondisi lingkungan secara *real-time* untuk mendukung proses pengambilan keputusan.

Pada sistem deteksi kebakaran, sensor asap dan sensor suhu menjadi dua komponen penting yang digunakan untuk mengidentifikasi indikasi awal terjadinya kebakaran. Sensor asap bekerja dengan mendeteksi partikel hasil pembakaran yang tersebar di udara, sedangkan sensor suhu digunakan untuk mendeteksi kenaikan temperatur yang tidak normal pada suatu area. Kombinasi kedua jenis sensor

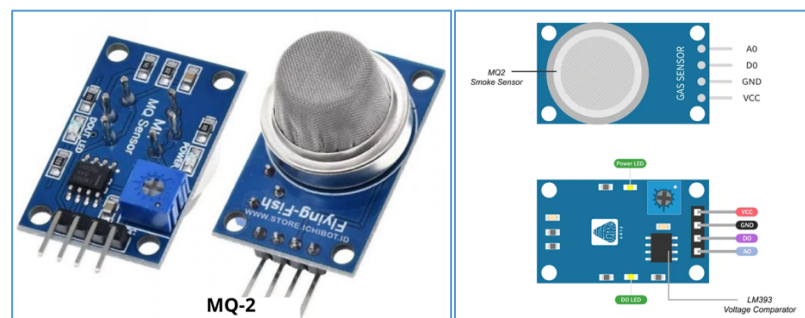
tersebut mampu meningkatkan akurasi deteksi kebakaran karena dapat meminimalkan terjadinya alarm palsu akibat gangguan lingkungan.

Dalam penelitian yang dilakukan sebelumnya, sensor MQ-2 digunakan untuk mendeteksi asap dan gas hasil pembakaran, sedangkan sensor DHT22 digunakan untuk mengukur temperatur lingkungan. Data yang diperoleh dari kedua sensor tersebut dikirimkan secara *real-time* ke sistem monitoring sehingga dapat digunakan untuk mendukung proses mitigasi bahaya kebakaran (Saputro et al., 2025). Hasil penelitian menunjukkan bahwa integrasi sensor dengan sistem IoT mampu meningkatkan kecepatan deteksi kondisi berbahaya serta memudahkan proses pemantauan jarak jauh. Selain digunakan untuk mendeteksi kebakaran, data temperatur dan konsentrasi asap juga dapat digunakan sebagai parameter tingkat bahaya (*hazard level*) yang merepresentasikan kondisi keselamatan pada suatu area bangunan.

Dalam konteks penelitian ini, data yang diperoleh dari sensor asap dan sensor suhu digunakan sebagai representasi kondisi lingkungan yang dinamis. Nilai pembacaan sensor selanjutnya diolah menjadi tingkat risiko pada setiap node lingkungan sehingga dapat dimanfaatkan oleh algoritma *Q-Learning* dalam menentukan rute evakuasi yang paling aman.



Gambar 2-3 Sensor DHT22



Gambar 2-4 Sensor MQ2

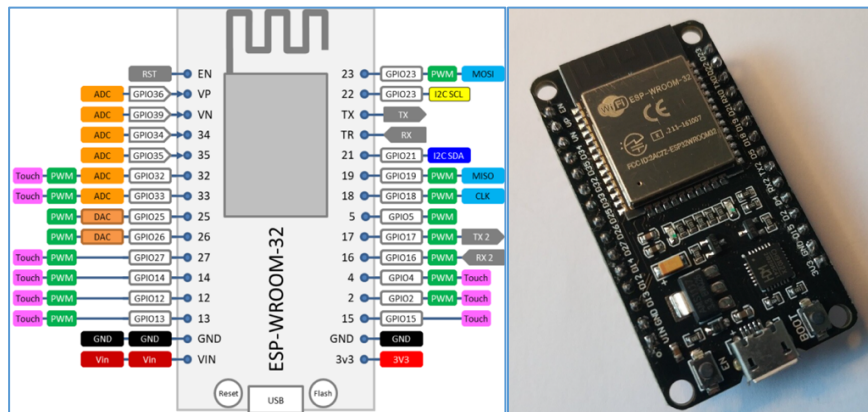
2.1.10 Mikrokontroler ESP32

ESP32 merupakan mikrokontroler berbasis *System-on-Chip* (SoC) yang dikembangkan oleh Espressif Systems dan dirancang khusus untuk aplikasi *Internet of Things*. ESP32 mengintegrasikan prosesor dual-core, memori, antarmuka komunikasi, serta modul Wi-Fi dan Bluetooth dalam satu perangkat sehingga mampu menjalankan fungsi akuisisi data, pemrosesan data, dan komunikasi jaringan secara bersamaan.

Dalam arsitektur IoT modern, ESP32 berperan sebagai *edge device* yang berada dekat dengan sumber data. Konsep *edge computing* memungkinkan sebagian proses komputasi dilakukan langsung pada perangkat sebelum data dikirim ke server pusat. Pendekatan ini dapat mengurangi latensi komunikasi, menghemat penggunaan bandwidth jaringan, dan meningkatkan kecepatan respons sistem terhadap perubahan kondisi lingkungan.

ESP32 memiliki kemampuan yang baik dalam mengelola data dari berbagai sensor secara simultan dan mengirimkan data tersebut melalui protokol MQTT menuju *platform monitoring* berbasis Node-RED (Naa, 2022). Hasil penelitian memperlihatkan bahwa ESP32 mampu mendukung proses *monitoring* secara *real-time* dengan tingkat keandalan yang tinggi.

Dalam penelitian ini, ESP32 digunakan sebagai perangkat akuisisi data yang menerima informasi dari sensor asap dan sensor suhu. Data yang diperoleh kemudian diproses dan dikirimkan secara *real-time* melalui jaringan nirkabel menuju broker MQTT sehingga dapat digunakan oleh sistem pemantauan dan algoritma penentuan rute evakuasi.



Gambar 2-5 Mikrokontroler ESP32

2.1.11 Message Queuing Telemetry Transport (MQTT)

Message Queuing Telemetry Transport (MQTT) merupakan protokol komunikasi ringan berbasis arsitektur *publish-subscribe* yang dirancang untuk perangkat dengan sumber daya terbatas dan jaringan dengan *bandwidth* rendah. MQTT menjadi salah satu protokol yang paling banyak digunakan dalam implementasi *Internet of Things* karena mampu menyediakan komunikasi data yang cepat, efisien, dan andal.

Arsitektur MQTT terdiri atas tiga komponen utama yaitu *publisher*, *broker*, dan *subscriber*. *Publisher* bertugas mengirimkan data ke broker melalui topik tertentu, sedangkan *subscriber* menerima data berdasarkan topik yang telah didaftarkan sebelumnya. Mekanisme ini memungkinkan komunikasi antar perangkat berlangsung secara tidak langsung sehingga meningkatkan fleksibilitas dan skalabilitas sistem.

Penggunaan MQTT pada sistem IoT memungkinkan pengiriman data sensor secara *real-time* dengan konsumsi *bandwidth* yang rendah, kombinasi MQTT dan Node-RED mampu menyediakan sistem *monitoring* lingkungan yang responsif dan mudah diintegrasikan dengan berbagai jenis sensor (Utomo & Izzaturrahmani, 2025). Dengan karakteristik tersebut, MQTT sangat sesuai digunakan pada sistem deteksi kebakaran yang membutuhkan kecepatan pengiriman informasi serta kemampuan beradaptasi terhadap perubahan kondisi lingkungan.

Dalam penelitian ini, MQTT digunakan sebagai media komunikasi utama yang menghubungkan perangkat ESP32 dengan Node-RED dan basis data MongoDB. Data sensor yang dikirim melalui MQTT menjadi sumber informasi bagi sistem dalam mengevaluasi tingkat bahaya dan menentukan rute evakuasi yang optimal.

2.1.12 Node-RED

Node-RED merupakan platform pemrograman berbasis alur (*flow-based programming*) yang dikembangkan oleh IBM Emerging Technology Services dan dibangun di atas lingkungan Node.js. Platform ini dirancang untuk mempermudah integrasi antara perangkat keras, layanan web, basis data, serta aplikasi *Internet of Things* (IoT) melalui pendekatan pemrograman visual. Pada Node-RED, setiap

fungsi direpresentasikan dalam bentuk node yang dapat dihubungkan satu sama lain untuk membentuk alur pemrosesan data secara terstruktur.

Dalam arsitektur IoT, Node-RED berperan sebagai *middleware* yang menjembatani komunikasi antara perangkat sensor, protokol komunikasi, sistem penyimpanan data, dan aplikasi pengguna. *Middleware* merupakan lapisan perangkat lunak yang bertugas mengelola pertukaran data antar komponen sistem yang berbeda sehingga proses integrasi dapat dilakukan secara lebih mudah dan efisien. Penggunaan *middleware* menjadi penting pada sistem IoT karena lingkungan IoT umumnya terdiri atas berbagai perangkat dengan protokol komunikasi dan format data yang berbeda.

Secara umum, Node-RED menyediakan tiga kelompok node utama, yaitu *input* node, *processing* node, dan *output* node. *Input* node digunakan untuk menerima data dari berbagai sumber seperti sensor, broker MQTT, maupun layanan web. *Processing* node digunakan untuk melakukan *transformasi*, penyaringan, maupun pengolahan data. Sementara itu, *output* node digunakan untuk mengirimkan data ke *dashboard*, basis data, layanan cloud, maupun perangkat lainnya. Struktur ini memungkinkan proses pengembangan sistem dilakukan secara modular dan mudah dipelihara.

Menurut penelitian yang sudah dilakukan, Node-RED mampu memproses dan memvisualisasikan data lingkungan secara *real-time* dengan tingkat respons yang baik (Utomo & Izzaturrahmani, 2025). Node-RED mampu mengintegrasikan data sensor yang dikirimkan melalui protokol MQTT dengan sistem monitoring berbasis web secara *real-time* (Naa, 2022). Hasil penelitian menunjukkan bahwa penggunaan Node-RED dapat mempercepat proses pengembangan aplikasi IoT karena menyediakan antarmuka visual yang intuitif serta mendukung berbagai protokol komunikasi yang umum digunakan dalam lingkungan IoT.

Selain berfungsi sebagai media integrasi data, Node-RED juga menyediakan *dashboard monitoring* yang memungkinkan pengguna melakukan pengawasan kondisi sistem secara langsung. *Dashboard* tersebut dapat menampilkan berbagai informasi seperti grafik perubahan suhu, tingkat konsentrasi asap, status sensor, maupun tingkat bahaya pada setiap area bangunan. Kemampuan visualisasi data ini

menjadi penting dalam sistem deteksi kebakaran karena dapat membantu operator memahami kondisi lingkungan secara cepat dan akurat.

Pada penelitian ini, Node-RED digunakan sebagai *middleware* yang menerima data sensor dari ESP32 melalui broker MQTT, melakukan pengolahan data awal, serta meneruskan data tersebut ke basis data MongoDB. Selain itu, Node-RED juga digunakan untuk menyediakan antarmuka *monitoring* yang menampilkan kondisi lingkungan secara *real-time*. Dengan demikian, Node-RED berperan sebagai pusat integrasi data yang menghubungkan seluruh komponen sistem sehingga informasi yang dibutuhkan oleh algoritma *Q-Learning* dapat diperoleh secara cepat dan konsisten.

2.1.13 MongoDB

MongoDB merupakan sistem manajemen basis data NoSQL yang menggunakan model penyimpanan berbasis dokumen (*document-oriented database*). Berbeda dengan basis data relasional yang menyimpan data dalam bentuk tabel, baris, dan kolom, MongoDB menyimpan data dalam format BSON (Binary JavaScript Object Notation) yang merupakan pengembangan dari format JSON. Pendekatan ini memungkinkan penyimpanan data dilakukan secara lebih *fleksibel* karena setiap dokumen dapat memiliki struktur yang berbeda tanpa memerlukan perubahan skema basis data secara keseluruhan.

Perkembangan teknologi *Internet of Things* menghasilkan volume data yang sangat besar dan bersifat dinamis. Data yang berasal dari sensor umumnya memiliki karakteristik berupa aliran data kontinu (*streaming data*), frekuensi pembaruan yang tinggi, serta struktur data yang dapat berubah sesuai kebutuhan sistem. Kondisi tersebut menyebabkan penggunaan basis data *relasional konvensional* sering kali menghadapi keterbatasan dalam hal *fleksibilitas* dan *skalabilitas*. Oleh karena itu, basis data NoSQL seperti MongoDB menjadi salah satu solusi yang banyak digunakan dalam implementasi sistem IoT.

MongoDB memiliki tiga komponen utama, yaitu *database*, *collection*, dan *document*. *Database* merupakan kumpulan data yang terdiri atas beberapa *collection*. *Collection* berfungsi sebagai wadah penyimpanan dokumen yang memiliki karakteristik serupa. Sementara itu, *document* merupakan unit

penyimpanan data utama yang berisi pasangan atribut dan nilai dalam format JSON atau BSON. Struktur ini memungkinkan penyimpanan data dilakukan secara *fleksibel* tanpa memerlukan definisi skema yang kaku sebagaimana pada basis data relasional.

Salah satu keunggulan utama basis data NoSQL adalah kemampuannya dalam menangani data dalam jumlah besar dengan kebutuhan *skalabilitas* yang tinggi (Moniruzzaman & Hossain, 2013). MongoDB mendukung mekanisme *horizontal scaling* melalui teknik *sharding* yang memungkinkan distribusi data ke beberapa server secara bersamaan. Kemampuan ini menjadikan MongoDB sesuai untuk aplikasi IoT yang menghasilkan data sensor secara terus-menerus.

Fleksibilitas struktur dokumen juga memungkinkan sistem menyimpan berbagai jenis data sensor tanpa memerlukan modifikasi skema yang kompleks. Selain itu, integrasi MongoDB dengan Node-RED dan MQTT dapat dilakukan dengan mudah sehingga mendukung pengembangan sistem *monitoring* yang bersifat *real-time*.

Dalam penelitian ini, MongoDB digunakan sebagai media penyimpanan data yang berasal dari sensor asap dan sensor suhu. Selain menyimpan data lingkungan, MongoDB juga digunakan untuk menyimpan informasi tingkat bahaya dan titik lokasi. Penyimpanan data tersebut diperlukan untuk mendukung proses analisis, evaluasi kinerja sistem, dan visualisasi hasil penelitian.

Penggunaan MongoDB pada penelitian ini didasarkan pada beberapa pertimbangan, yaitu kemampuan menangani data dalam jumlah besar, *fleksibilitas* struktur penyimpanan, kemudahan integrasi dengan Node-RED, serta dukungan terhadap aplikasi *real-time*. Dengan karakteristik tersebut, MongoDB mampu memenuhi kebutuhan penyimpanan data pada sistem evakuasi kebakaran dinamis yang memerlukan pembaruan informasi lingkungan secara berkelanjutan.

2.2 Penelitian Terdahulu

Dari telaah literatur terlihat bahwa studi yang ada dapat dikelompokkan ke dalam empat tema besar, yaitu: deteksi kebakaran berbasis IoT, penilaian risiko dan *hazard mapping*, algoritma rute konvensional, dan evakuasi berbasis *reinforcement learning*.

Tabel 2-1 Perbandingan Penelitian Terkait

No	Peneliti	Fokus penelitian	Metode utama	Temuan penting	Kesenjangan terhadap tesis
1	Shaharuddin et al. (2023)	Review sensor IoT pada kebakaran indoor	Systematic review	IoT sangat potensial untuk deteksi bahaya indoor	Belum fokus pada optimasi rute adaptif
2	Mirahadi dan McCabe (2021)	Evakuasi real-time bangunan	Dijkstra-based model	Dapat mencari jalur aman secara real-time	Belum menggunakan pembelajaran adaptif berbasis RL
3	Ding et al. (2024)	Simulasi evakuasi darurat berbasis BIM	BIM-based simulation	BIM membantu pemetaan ruang dan evaluasi evakuasi	Belum mengintegrasikan Q-learning dan sensor IoT secara langsung
4	Wong dan Lee (2023)	Navigasi indoor dan berbagi informasi darurat	BIM + networking	Mendukung navigasi dan informasi kolaboratif saat darurat	Belum fokus pada pemilihan rute belajar adaptif
5	Zhang et al. (2024)	Pelacakan penghuni dan optimasi jalur	IoT-based tracking and path optimization	Data IoT bermanfaat untuk manajemen kebakaran di bangunan besar	Belum menekankan Q-learning sebagai inti keputusan
6	Huang et al. (2024)	Pemilihan rute evakuasi otomatis real-time	Decision-based route model	Rute harus mempertimbangkan proses keputusan pejalan	Belum menekankan integrasi sensor asap terdistribusi dan RL tabular
7	Han et al. (2025)	Evakuasi dengan shared Q-learning	Multi-agent shared Q-learning	Q-learning efektif meningkatkan keputusan evakuasi	Belum difokuskan pada integrasi hazard map dari sensor IoT nyata
8	Xu et al. (2022)	Emergency path planning indoor	Q-learning optimization	Grid environment efektif untuk pelatihan path planning	Belum mengintegrasikan data sensor kebakaran real-time
9	Bai et al. (2025)	Jalur evakuasi berbasis dynamic fire risk	Improved A* + dynamic risk	Risiko dinamis dapat diintegrasikan ke path planning	Masih berbasis A*, belum pada pembelajaran adaptif RL
10	Senanayake et al. (2024)	Review agent-based pedestrian evacuation	Systematic review	Validasi, perilaku, dan kompleksitas simulasi masih menantang	Memberi dasar penting untuk desain pengujian, bukan sistem rute RL-IoT langsung

Berdasarkan studi di atas, penelitian tentang IoT untuk kebakaran sudah cukup kuat pada aspek deteksi dan *monitoring*. Sensor mampu memberi data *real-time* yang bermanfaat untuk sistem keselamatan. Namun, banyak studi masih berhenti pada tahap alarm, visualisasi, atau pelacakan. Hal ini menunjukkan bahwa ruang untuk penelitian pada *decision support* masih terbuka lebar (AlQahtani et al., 2025; Shahrudin et al., 2023).

Pada sisi perencanaan rute, algoritma klasik seperti Dijkstra dan A* masih dominan karena sederhana dan mudah diimplementasikan. Akan tetapi, kedua metode ini memerlukan pembaruan bobot risiko secara eksplisit setiap kali lingkungan berubah. Penelitian sebelumnya memperlihatkan bahwa *dynamic fire risk* memang dapat memperbaiki kualitas perencanaan rute, tetapi pendekatannya masih berbasis algoritma pencarian tradisional. Artinya, kemampuan “belajar” dari pengalaman belum menjadi inti sistem (Bai et al., 2025).

Pada sisi *Reinforcement Learning* (RL), penelitian lainnya juga menunjukkan bahwa *Q-learning* relevan untuk *emergency path planning*, RL unggul karena dapat menyeimbangkan beberapa tujuan sekaligus (Han et al., 2025; Xu et al., 2022). Namun, integrasi langsung antara sensor IoT *real-time*, peta bahaya digital, dan *Q-learning* masih belum banyak dibahas sebagai satu sistem utuh.