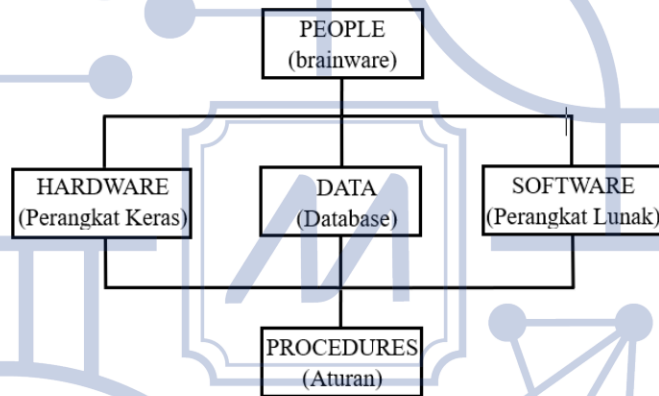


## BAB II

### KAJIAN LITERATUR

#### 2.1 Sistem Informasi

Sistem informasi merupakan suatu sistem yang digunakan untuk mengelola data menjadi informasi yang bermanfaat dalam mendukung kegiatan operasional serta pengambilan keputusan dalam suatu organisasi. Sistem informasi terdiri dari beberapa komponen utama yang saling berinteraksi secara terpadu untuk menghasilkan informasi yang berkualitas. Keterkaitan antarkomponen utama dalam sebuah sistem informasi diilustrasikan pada Gambar 2.1 berikut.



Gambar 2.1 Komponen Sistem Informasi [2]

Berdasarkan Gambar 2.1 di atas, terlihat bahwa basis data (Data) menempati posisi sentral sebagai aset inti yang dikelola dalam sistem. Komponen pendukung di sekelilingnya memiliki peran masing-masing: perangkat keras (*hardware*) dan perangkat lunak (*software*) berfungsi sebagai sarana pemrosesan, prosedur memberikan batasan aturan operasional, sedangkan sumber daya manusia (*people/brainware*) berperan sebagai pengendali yang menjalankan seluruh komponen tersebut. Keberadaan sistem ini sangat penting karena mampu membantu organisasi dalam mengelola data secara sistematis dan terstruktur [1].

Dalam perkembangannya, sistem informasi berbasis komputer telah banyak digunakan untuk menggantikan sistem manual yang dinilai kurang efisien guna meningkatkan kecepatan pengolahan data, meningkatkan akurasi informasi, serta mengurangi kesalahan yang disebabkan oleh faktor manusia (*human error*). Selain itu, sistem informasi memungkinkan penyimpanan data secara terpusat sehingga mempermudah proses pencarian, pengolahan, dan penyusunan laporan dalam jumlah besar [2].

Pada konteks pengelolaan data jemaah umrah, sistem informasi memiliki peranan yang sangat krusial karena data yang dikelola bersifat dinamis dan terus bertambah setiap periode keberangkatan. Tanpa adanya sistem yang terintegrasi, pengelolaan data berpotensi menimbulkan berbagai permasalahan seperti duplikasi data, inkonsistensi informasi, serta keterlambatan dalam penyusunan laporan keberangkatan. Oleh karena itu, diperlukan sistem informasi yang mampu mengelola data secara efektif, efisien, dan terintegrasi pada PT Masar Alsama Global [4].

### 2.1.1 Sistem Informasi Berbasis *Web*

Sistem informasi berbasis *web* merupakan sistem informasi yang dikembangkan dengan memanfaatkan teknologi *web* dan dapat diakses melalui jaringan internet menggunakan peramban (*browser*). Sistem ini memungkinkan pengguna untuk mengakses informasi secara fleksibel tanpa dibatasi oleh waktu dan lokasi, sehingga sangat sesuai digunakan dalam organisasi yang membutuhkan akses data secara *real-time* dan terpusat [1]. Secara teknis, sistem informasi berbasis *web* bekerja menggunakan arsitektur *client-server*, dimana pengguna mengirimkan permintaan melalui *browser*, kemudian peladen (*server*) memproses permintaan tersebut dan memberikan respons yang bersifat dinamis [6].

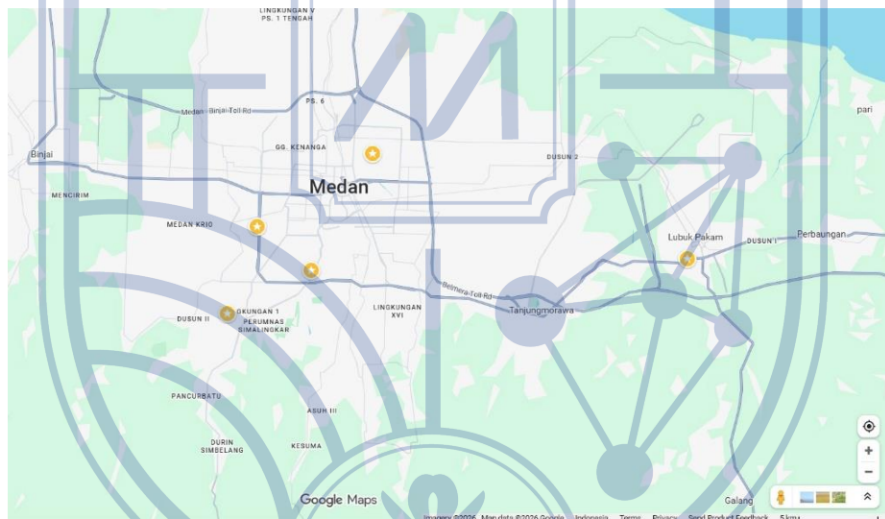
Keunggulan utama sistem berbasis *web* terletak pada kemudahan akses serta fleksibilitas penggunaannya. Selain itu, sistem ini dapat diintegrasikan dengan berbagai teknologi lain seperti basis data (*database*) dan layanan *web* (*web services*), sehingga mampu mendukung pengolahan data secara dinamis dan terstruktur. Hal ini menjadikan sistem berbasis *web* sebagai solusi yang efektif dalam meningkatkan efisiensi operasional serta kualitas pelayanan dalam suatu organisasi [7]. Penggunaan teknologi *open-source* seperti MySQL juga sering menjadi pilihan dalam pengembangan sistem ini karena sifatnya yang reliabel serta mudah dalam proses integrasi data [6].

Dalam konteks pengelolaan data jemaah umrah, sistem informasi berbasis *web* memberikan kemudahan dalam proses penginputan, pengolahan, dan pencarian data secara cepat dan akurat. Pengguna dapat mengakses sistem kapan saja dan di mana saja selama terhubung dengan jaringan internet. Hal ini sangat penting untuk memantau status pendaftaran serta kelengkapan dokumen jemaah secara berkala. Penelitian sebelumnya menunjukkan bahwa penerapan sistem informasi berbasis *web* mampu meningkatkan efektivitas kerja, meminimalisir kesalahan manusia (*human error*), serta meningkatkan kualitas pelayanan kepada jemaah secara signifikan [1], [6]

### 2.1.2 Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis (SIG) merupakan sistem berbasis komputer yang digunakan untuk mengelola, menganalisis, dan menampilkan data yang berkaitan dengan lokasi geografis. SIG mengintegrasikan data spasial (data lokasi) dan data atribut (data deskriptif) sehingga dapat divisualisasikan dalam bentuk peta digital yang interaktif [8]. Data spasial merepresentasikan objek di permukaan bumi dalam bentuk koordinat, sementara data atribut memberikan keterangan tambahan mengenai objek tersebut, seperti nama jemaah, usia maupun wilayah tempat tinggal.

SIG juga dapat didefinisikan sebagai sistem yang mampu mengolah data geografis dalam bentuk titik, garis, dan area yang kemudian disajikan dalam bentuk visual seperti peta, tabel, maupun grafik. Kemampuan ini memungkinkan pengguna untuk memahami hubungan antara data dan lokasi secara lebih jelas sehingga mendukung pengambilan keputusan yang lebih tepat [3]. Contoh visualisasi Sistem Informasi Geografis dapat dilihat pada Gambar 2.3.



### Gambar 2.2 Visualisasi Sistem Informasi Geografis (SIG)

Sumber: Google Maps (2026)

Gambar 2.3 menunjukkan contoh visualisasi data spasial dalam bentuk peta digital yang menampilkan persebaran lokasi berdasarkan titik-titik tertentu. Titik-titik tersebut merepresentasikan koordinat keberadaan jemaah yang telah dikonversi dari data alamat fisik menjadi data koordinat geografis (*latitude* dan *longitude*).

Dalam penelitian ini, SIG digunakan untuk memetakan persebaran domisili jemaah umrah berdasarkan wilayah tempat tinggal. Informasi ini sangat penting bagi perusahaan karena dapat digunakan untuk mengetahui konsentrasi jemaah di suatu wilayah tertentu, serta membantu dalam penyusunan strategi pemasaran yang lebih efektif. Selain itu,

penggunaan SIG juga meningkatkan kualitas penyajian informasi karena data dapat ditampilkan secara visual dan mudah dipahami [9].

### 2.1.3 Sistem Informasi Manajemen

Sistem Informasi Manajemen (SIM) adalah sistem yang dirancang untuk menyediakan informasi guna mendukung fungsi manajerial seperti perencanaan, pengendalian, dan pengambilan keputusan dalam sebuah organisasi. SIM berperan strategis dalam menyediakan data yang relevan, terstruktur, dan dapat diakses secara tepat waktu agar proses pengambilan keputusan dapat dilakukan berbasis bukti (*evidence-based decision making*) [10].

Penerapan SIM mencakup beberapa area fungsional yang saling terintegrasi, seperti manajemen keuangan yang digunakan untuk mengelola catatan transaksi dan arus kas guna meningkatkan akurasi data finansial secara *real-time* [11]. Selain itu, manajemen logistik berperan dalam mengelola stok barang serta memantau rotasi persediaan di gudang untuk memastikan ketersediaan perlengkapan operasional perusahaan [12].

Keberhasilan pemanfaatan SIM dalam sebuah organisasi dipengaruhi oleh beberapa faktor pendukung, antara lain dukungan pimpinan, infrastruktur teknologi yang memadai, serta kebijakan internal yang mengatur penggunaan sistem sebagai bagian dari prosedur kerja formal [10]. Dengan integrasi yang baik, SIM mampu meminimalkan penggunaan metode manual yang rentan terhadap kesalahan serta keterlambatan data.

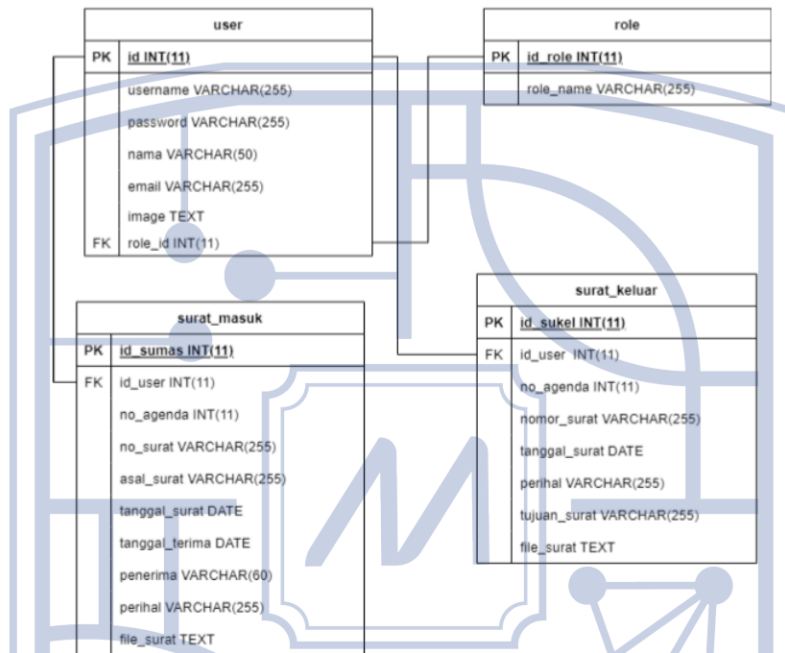
Dalam konteks penelitian ini, konsep Sistem Informasi Manajemen diterapkan pada sistem pengelolaan data jemaah umrah di PT Masar Alsama Global. Sistem yang dikembangkan menyatukan pengelolaan data jemaah, berkas digital, catatan pembayaran, hingga ketersediaan stok perlengkapan dalam satu platform terpusat. Hal ini mempermudah pihak manajemen dalam memantau capaian data serta menyusun laporan keberangkatan secara lebih sistematis dan akurat.

## 2.2 Basis Data

Basis data merupakan kumpulan data yang tersusun secara sistematis dan saling berhubungan sehingga dapat diakses, dikelola, dan diperbarui dengan mudah. Basis data menjadi komponen penting dalam sistem informasi karena berfungsi sebagai media penyimpanan utama yang mendukung seluruh proses pengolahan data secara digital [1].

Penggunaan basis data relasional memungkinkan pengelolaan data terstruktur secara efisien serta membantu meminimalkan terjadinya redundansi atau duplikasi data. Integritas

dan konsistensi data dijaga melalui penerapan aturan pada *Database Management System* (DBMS), seperti penggunaan *Primary Key* (PK) dan *Foreign Key* (FK) untuk menghubungkan antartabel [13]. Dengan penerapan kunci-kunci tersebut, data yang dihasilkan menjadi lebih akurat dan terorganisir. Ilustrasi mengenai contoh penerapan kunci utama dan kunci asing untuk menghubungkan antartabel dalam basis data relasional dapat dilihat pada Gambar 2.4 berikut:



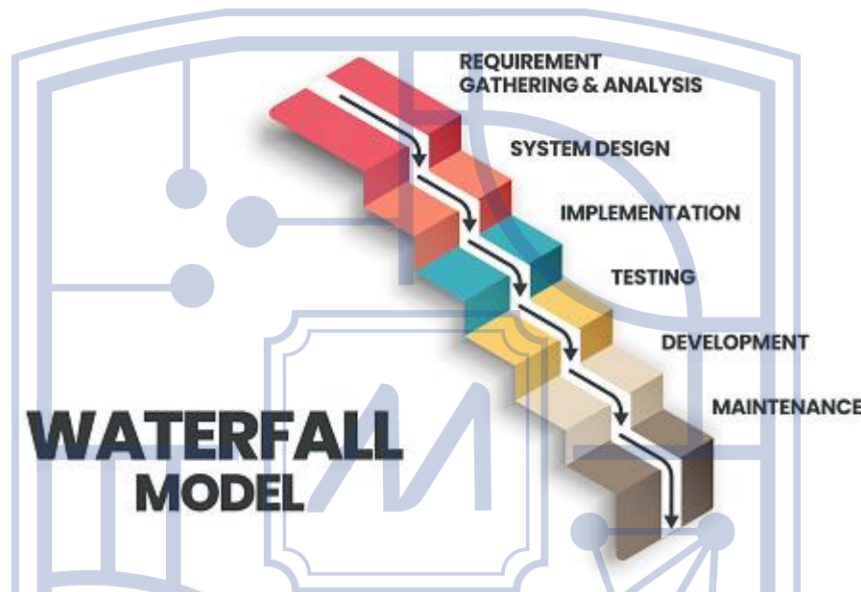
Gambar 2.3 Relasi Antartabel dalam Basis Data [13]

Berdasarkan Gambar 2.4 di atas, terlihat alur teknis di mana kunci utama (*Primary Key*) pada tabel induk dirujuk oleh kunci asing (*Foreign Key*) pada tabel terkait. Sebagai contoh, dalam sistem ini, tabel user terhubung dengan tabel role melalui role\_id. Selain itu, data personal pengguna pada tabel user dihubungkan ke tabel transaksi pengarsipan, yaitu surat\_masuk dan surat\_keluar, melalui atribut id\_user. Penerapan relasi ini memastikan bahwa setiap data surat yang tercatat dalam sistem memiliki keterkaitan yang jelas dengan data pengguna yang mengelolanya [13].

Dalam sistem pengelolaan jemaah umrah pada PT Masar Alsama Global, basis data digunakan untuk menyimpan berbagai informasi krusial seperti data pribadi jemaah, dokumen perjalanan, jadwal keberangkatan, serta data keberangkatan jemaah. Melalui struktur basis data yang terorganisir menggunakan MySQL, proses pencarian dan pembaruan data dapat dilakukan secara *real-time*, yang pada akhirnya mempermudah pihak manajemen dalam menyusun laporan keberangkatan secara otomatis dan akurat [14].

### 2.3 Metode *Waterfall*

Metode *Waterfall* merupakan model pengembangan perangkat lunak klasik yang bersifat sistematis, berurutan, dan bertahap menyerupai aliran air terjun. Dalam model ini, setiap tahapan harus diselesaikan secara penuh sebelum melanjutkan ke tahap berikutnya guna memastikan dokumentasi yang lengkap serta meminimalisir kesalahan dalam proses pengembangan sistem [6]. Alur kerja metode *Waterfall* pada penelitian ini dapat dilihat pada Gambar 2.5.



Gambar 2.4 Tahapan Metode *Waterfall* [15]

Berdasarkan Gambar 2.5, tahapan pengembangan sistem dalam penelitian ini terdiri dari enam tahap utama sebagai berikut:

1. *Requirement Gathering & Analysis*

Tahap ini merupakan proses identifikasi dan pengumpulan seluruh kebutuhan sistem guna memahami spesifikasi fungsional dan nonfungsional yang dibutuhkan oleh pengguna.

2. *System Design*

Pada tahapan ini dilakukan perancangan arsitektur sistem, desain basis data, serta perancangan antarmuka pengguna (*user interface*) guna memetakan alur proses layanan yang akan dibangun.

3. *Implementation*

Tahap implementasi mencakup pembangunan aplikasi berbasis teknologi yang telah dipilih sesuai dengan hasil desain sistem, dimana rancangan diubah menjadi modul-modul program yang saling terintegrasi.

#### 4. *Testing*

Sistem yang telah dibangun akan diuji baik secara fungsional maupun nonfungsional untuk memastikan bahwa seluruh fitur berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan.

#### 5. *Development Maintenance*

Tahap akhir dimana aplikasi diterapkan pada *server* operasional agar dapat diakses oleh pengguna. Selanjutnya, dilakukan pemeliharaan berkala untuk memastikan sistem berjalan stabil dan siap beradaptasi terhadap perubahan di masa mendatang.

Metode *Waterfall* dipilih karena memiliki alur kerja yang terstruktur dan sistematis, sehingga sangat efektif digunakan dalam pengembangan sistem informasi berbasis *web* dengan kebutuhan yang telah didefinisikan secara jelas sejak awal penelitian [3].

### 2.4 Alat Bantu Pengembangan Sistem

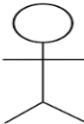
#### 2.4.1 *Use Case Diagram*

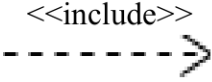
*Use Case Diagram* merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara aktor dengan sistem berdasarkan kebutuhan fungsional. Diagram ini berfungsi sebagai representasi visual untuk menunjukkan fungsi-fungsi yang tersedia dalam sistem yang akan dikembangkan.

Dalam penelitian ini, *Use Case Diagram* digunakan untuk memodelkan fungsionalitas sistem sehingga memudahkan dalam memahami hubungan antara pengguna dengan sistem. Selain itu, diagram ini membantu dalam mengidentifikasi dan menetapkan peran aktor, seperti admin, dalam mengelola data jemaah, paket umroh, keberangkatan, dan laporan secara terintegrasi [4].

Simbol-simbol dalam *Use Case Diagram* digunakan untuk menggambarkan interaksi tersebut dan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-Simbol *Use Case Diagram* [4]

| Simbol                                                                              | Nama            | Keterangan                                                                                    |
|-------------------------------------------------------------------------------------|-----------------|-----------------------------------------------------------------------------------------------|
|  | <i>Actor</i>    | Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> |
|  | <i>Use Case</i> | Abstraksi dan interaksi antara sistem dan aktor                                               |

| Simbol                                                                            | Nama                   | Keterangan                                                                                                                                 |
|-----------------------------------------------------------------------------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Association</i>     | Garis penghubung antara aktor dan <i>use case</i> yang menunjukkan bahwa aktor terlibat atau berpartisipasi dalam <i>use case</i> tersebut |
|  | <i>Include</i>         | Menunjukkan bahwa suatu <i>use case</i> selalu melibatkan <i>use case</i> lain                                                             |
|  | <i>Extend</i>          | Menunjukkan bahwa suatu <i>use case</i> dapat diperluas oleh <i>use case</i> lain                                                          |
|  | <i>Generalization</i>  | Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>                                                           |
|  | <i>System Boundary</i> | Menunjukkan batasan antara sistem dengan lingkungan luar (aktor)                                                                           |

#### 2.4.2 Activity Diagram

*Activity Diagram* merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja (*workflow*) atau aktivitas dalam suatu sistem secara berurutan, mulai dari proses awal hingga proses akhir. Diagram ini juga digunakan untuk memodelkan alur proses serta interaksi antaraktivitas dalam suatu sistem atau proses bisnis [16].

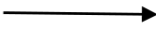
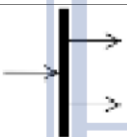
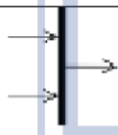
Dalam perancangan sistem informasi, *Activity Diagram* digunakan sebagai alat bantu untuk memvisualisasikan alur proses yang terjadi di dalam sistem. Hal ini membantu dalam memahami urutan aktivitas secara lebih terstruktur dan sistematis, serta mendukung proses analisis dan perancangan sistem [16].

Dalam penelitian ini, *Activity Diagram* digunakan untuk menggambarkan alur proses utama sistem pengelolaan data jemaah berbasis *web*, mulai dari proses *input* data hingga pengelolaan informasi oleh sistem.

Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-Simbol *Activity Diagram* [16]

| No. | Simbol                                                                              | Nama                        | Keterangan                             |
|-----|-------------------------------------------------------------------------------------|-----------------------------|----------------------------------------|
| 1   |  | <i>Initial Node (Start)</i> | Bagaimana objek dibentuk atau diawali. |

| No. | Simbol                                                                              | Nama                       | Keterangan                                                                                |
|-----|-------------------------------------------------------------------------------------|----------------------------|-------------------------------------------------------------------------------------------|
| 2   |    | <i>Activity</i>            | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 3   |    | <i>Decision</i>            | Menunjukkan percabangan berdasarkan kondisi (ya/tidak)                                    |
| 4   |    | <i>Control Flow</i>        | Menunjukkan arah aliran proses dari satu aktivitas ke aktivitas lain                      |
| 5   |    | <i>Action</i>              | <i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi                       |
| 6   |    | <i>Fork</i>                | Memecah satu proses menjadi beberapa aktivitas yang berjalan bersamaan                    |
| 7   |   | Join                       | Menggabungkan kembali beberapa proses paralel menjadi satu alur                           |
| 8   |  | <i>Activity Final Node</i> | Menunjukkan akhir dari proses                                                             |

### 2.4.3 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) merupakan metode perancangan basis data yang digunakan untuk menggambarkan struktur data serta hubungan antarentitas dalam suatu sistem. ERD digunakan untuk membantu memodelkan data secara konseptual sebelum diimplementasikan ke dalam *database*.

ERD digunakan dalam pengembangan sistem informasi berbasis *web* untuk menggambarkan hubungan antardata seperti entitas pengguna, layanan, dan informasi sistem sehingga dapat membantu dalam penyusunan struktur *database* [8]. ERD juga digunakan dalam perancangan sistem berbasis *web* yang melibatkan pengelolaan data seperti pengguna, pengaduan, dan pengelolaan sistem, sehingga dapat mendukung proses perancangan *database* yang lebih terstruktur [7].

Dalam penelitian ini, ERD digunakan untuk merancang struktur basis data sistem pengelolaan jemaah umrah yang mencakup entitas seperti data jemaah, wilayah, jadwal keberangkatan, dan laporan keberangkatan. ERD membantu menggambarkan hubungan antardata sehingga mempermudah implementasi ke dalam *database* relasional.

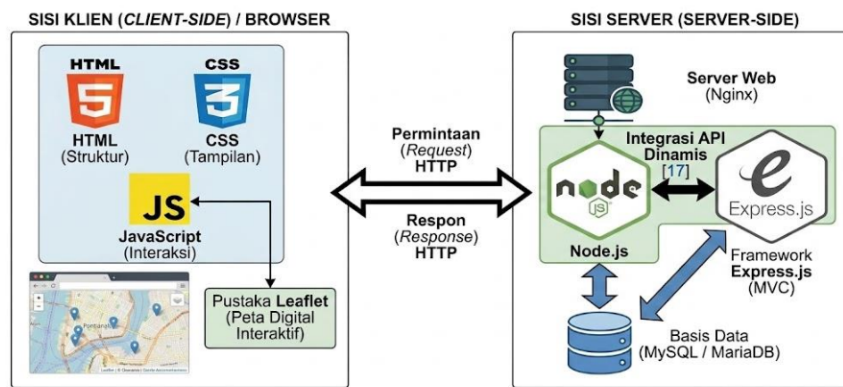
Simbol-simbol yang digunakan dalam *Entity Relationship Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-Simbol *Entity Relationship Diagram* (ERD) [7]

| Simbol                                                                              | Nama       | Keterangan                                                                                                                                   |
|-------------------------------------------------------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|    | Entitas    | Merupakan suatu simbol untuk mewakili suatu objek dengan karakteristik sama yang dilengkapi oleh atribut.                                    |
|    | Relasi     | Simbol yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan. Digambarkan dengan simbol belah ketupat. |
|    | Atribut    | Simbol yang menjelaskan karakteristik suatu entitas dan juga relasinya. Digambarkan dengan elips.                                            |
|  | Connection | Simbol untuk merangkai keterkaitan antara notasi-notasi yang berupa garis penghubung yang digunakan seperti entitas, relasi, dan atribut.    |

## 2.5 Teknologi Pengembangan Web

Pengembangan sistem informasi berbasis *web* pada penelitian ini memanfaatkan tumpukan teknologi (*technology stack*) yang terintegrasi untuk menciptakan aplikasi yang fungsional dan dinamis. Struktur utama antarmuka didefinisikan menggunakan *Hypertext Markup Language* (HTML) yang berfungsi untuk menyusun kerangka elemen seperti formulir pendaftaran, tabel data jemaah, dan kontainer peta. Untuk mengoptimalkan pengalaman pengguna, *Cascading Style Sheets* (CSS) digunakan untuk mengatur tata letak, tipografi, dan estetika visual, sehingga antarmuka sistem tetap konsisten dan responsif saat diakses melalui berbagai perangkat dengan ukuran layar yang berbeda [17]. Alur kerja integrasi teknologi *web* tersebut diilustrasikan secara komprehensif pada Gambar 2.6.



Gambar 2.5 Arsitektur Kerja Teknologi Web [17]

Berdasarkan Gambar 2.6, terlihat pembagian tugas yang jelas antara sisi klien (*client-side*) dan sisi server (*server-side*). Pada sisi klien, JavaScript memegang peranan krusial sebagai bahasa pemrograman yang menangani aspek interaktivitas. Dalam sistem ini, JavaScript bertanggung jawab penuh dalam menjalankan instruksi pustaka Leaflet untuk merender peta digital, mengelola penempatan penanda lokasi (*marker*), serta menampilkan informasi detail jemaah melalui jendela *popup* secara asinkron tanpa harus memuat ulang seluruh halaman web (*reload*) [5], [9].

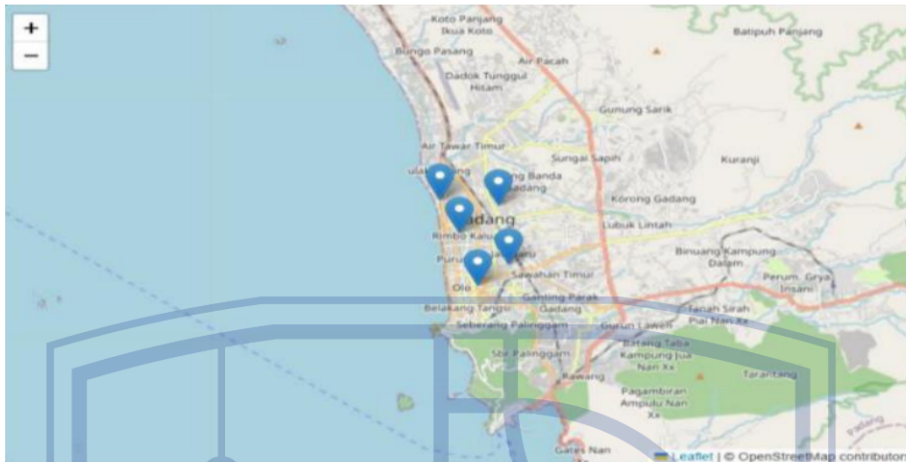
Sementara itu, pada sisi server, Node.js dengan *framework* Express.js bertindak sebagai mesin pengolah utama (*backend engine*). Express.js berfungsi untuk menjalankan logika bisnis, menangani proses autentikasi admin menggunakan *middleware* [18], serta menjembatani komunikasi data dengan sistem manajemen basis data. Melalui Express.js, data koordinat jemaah ditarik dan dikirimkan dalam format JSON agar dapat dipetakan secara dinamis di sisi klien. Integrasi yang solid ini memastikan bahwa manajemen distribusi jemaah umrah pada PT Masar Alsama Global dapat dilakukan secara efisien, akurat, dan berbasis data lokasi [1], [8].

### 2.5.1 Leaflet

*Leaflet* merupakan sebuah pustaka (*library*) JavaScript *open-source* yang digunakan untuk membangun peta digital yang interaktif, ringan, dan responsif. Keunggulan utama *Leaflet* terletak pada ukuran *file*-nya yang kecil namun memiliki fungsionalitas lengkap untuk menangani data spasial, sehingga sangat cocok digunakan dalam pengembangan sistem informasi berbasis web [9].

Dalam perkembangannya, *Leaflet* berfungsi sebagai media visualisasi data geografis yang efektif. Dengan menggunakan *Leaflet*, pengembang dapat menambahkan berbagai elemen visual di atas peta, seperti titik lokasi (*markers*) maupun area (*polygons*). Penjelasan

mengenai struktur komponen visual yang dikelola oleh *Leaflet* dapat dilihat pada Gambar 2.7.



Gambar 2.6 Struktur Komponen Visual *Leaflet* [9]

Gambar 2.7 menunjukkan berbagai komponen yang dapat dikonfigurasi dalam satu tampilan peta, mulai dari lapisan peta (*tile layer*) hingga elemen interaktif seperti *marker* dan *pop-up*. Mekanisme ini memastikan sistem tetap efisien dalam menyajikan data geografis yang kompleks [9].

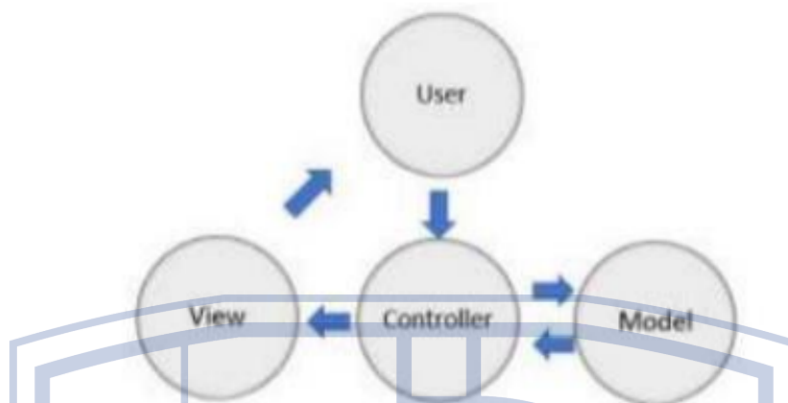
Pada penelitian ini, *Leaflet* digunakan untuk menampilkan lokasi domisili jemaah umrah dalam bentuk *marker*. Melalui integrasi dengan basis data, setiap *marker* dapat menampilkan informasi atribut jemaah melalui fitur *popup*. Visualisasi ini memberikan gambaran jelas mengenai konsentrasi jemaah, sehingga pihak manajemen dapat mengambil keputusan strategis berdasarkan pola distribusi wilayah tersebut.

### 2.5.2 *Framework Express.js dan Arsitektur MVC*

Express.js merupakan salah satu *framework* pengembangan aplikasi *web* minimalis dan fleksibel untuk Node.js yang bersifat *open-source*. *Framework* ini dirancang untuk meningkatkan efisiensi serta kualitas perangkat lunak. *Framework* ini menyediakan berbagai fungsionalitas utama seperti *routing*, penanganan *middleware*, serta integrasi dengan berbagai mesin *database* secara efisien. Dengan pendekatan yang *unopinionated* dan performa tinggi berbasis *non-blocking I/O*, Express.js menjadi standar industri dalam pengembangan sistem informasi berbasis *web* yang skalabel [17],[18]. Dengan sintaks yang ekspresif dan kemudahan penggunaan, Express.js banyak digunakan dalam pengembangan sistem informasi berbasis *web*.

Express.js umumnya menerapkan pola arsitektur *Model-View-Controller (MVC)*, yaitu suatu pendekatan yang memisahkan logika aplikasi menjadi tiga komponen utama agar

pengembangan sistem menjadi lebih terstruktur dan mudah dikelola [19]. Alur kerja arsitektur MVC dapat dilihat pada Gambar 2.8.



Gambar 2.7 Arsitektur *Model-View-Controller* (MVC) [19]

Berdasarkan Gambar 2.8, arsitektur MVC membagi sistem menjadi tiga bagian utama sebagai berikut:

1. *Model (Data Logic)*

*Model* berfungsi untuk mengelola data dan berinteraksi langsung dengan basis data. Komponen ini bertanggung jawab dalam proses penyimpanan, pengolahan, serta pengambilan data. Dalam ekosistem Express.js, *Model* biasanya dikelola menggunakan *library Object-Relational Mapping* (ORM) seperti Sequelize (untuk SQL) atau *Object Data Modeling* (ODM) seperti Mongoose (untuk MongoDB) guna mempermudah interaksi dengan *database*.

2. *View (User Interface)*

*View* merupakan bagian yang menampilkan data kepada pengguna dalam bentuk antarmuka. Komponen ini bertugas menyajikan informasi secara visual. Express.js mendukung berbagai *template engine* seperti EJS, Pug, atau Handlebars untuk membangun tampilan yang dinamis. Dalam pengembangan modern, Express.js juga sering berfungsi sebagai *backend API* yang mengirimkan data dalam format JSON untuk dikonsumsi oleh *framework frontend* seperti React atau Vue.

3. *Controller (Control Logic)*

*Controller* berfungsi sebagai jembatan atau penghubung antara *Model* dan *View*. Komponen ini menerima permintaan (*request*) dari pengguna melalui *route*, memproses logika bisnis, memanggil data melalui *Model*, dan kemudian menentukan respons atau tampilan yang akan dikirimkan kepada pengguna melalui *View*.

Komunikasi antara sisi *client* dan *server* pada Express.js umumnya menggunakan pendekatan berbasis API dengan memanfaatkan protokol HTTP seperti metode GET, POST, PUT, dan DELETE untuk mengelola pertukaran data secara terstruktur. API tersebut biasanya menyediakan *endpoint* yang mengirimkan data dalam format JSON, yang kemudian dapat dimanfaatkan oleh *client-side* untuk menampilkan data secara dinamis [19].

Penerapan arsitektur MVC dalam Express.js memberikan keuntungan signifikan dalam hal pemeliharaan kode (*maintainability*) dan skalabilitas sistem. Dengan pemisahan yang jelas antara logika data dan presentasi, pengembang dapat berkolaborasi lebih efektif dan mengelola kompleksitas aplikasi dengan lebih baik [19].

Selain itu, Express.js didukung oleh ekosistem *Node Package Manager* (NPM) yang menyediakan ribuan *middleware* untuk keamanan (seperti Helmet atau Passport), validasi data, dan manajemen sesi. Hal ini menjadikan Express.js sebagai salah satu *framework* paling populer dan andal dalam pengembangan sistem informasi berbasis *web* saat ini [18].

## 2.6 Black Box Testing

*Black box testing* merupakan metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas berdasarkan spesifikasi kebutuhan tanpa melibatkan pemeriksaan struktur kode internal atau alur logika program. Pengujian ini memungkinkan verifikasi apakah sistem memberikan respons yang tepat terhadap berbagai kondisi *input* yang diberikan [20]. Jika hasil keluaran sistem telah sesuai dengan spesifikasi yang diharapkan, maka sistem dinyatakan berfungsi dengan baik. Sebaliknya, jika sistem gagal menjalankan prosedur yang diminta atau menghasilkan *output* yang tidak valid, maka sistem tersebut dinyatakan membutuhkan perbaikan lebih lanjut sebelum dapat diimplementasikan [21].

Dalam implementasinya, terdapat berbagai teknik yang dapat digunakan dalam menguji *black box* untuk memastikan cakupan pengujian yang luas. Beberapa teknik yang umum diterapkan antara lain [22]:

1. *Equivalence Partitioning*: Teknik ini bekerja dengan cara membagi domain *input* ke dalam beberapa kelompok partisi data yang setara. Jika satu sampel dalam partisi tersebut berhasil diproses oleh sistem, maka seluruh nilai lain dalam kelompok yang sama dianggap valid.
2. *Boundary Value Analysis* (BVA): Teknik ini berfokus pada pengujian titik batas jangkauan *input*, baik batas minimum maupun maksimum suatu rentang data.

3. *State Transision Testing*: Teknik ini menguji perubahan status sistem yang dipicu oleh urutan *input* tertentu. Fokus utamanya adalah memastikan transisi dari satu kondisi ke kondisi berikutnya berjalan sesuai logika.
4. *Decision Table Testing*: Teknik ini menggunakan tabel logika untuk menguji berbagai kombinasi *input* yang kompleks guna memastikan seluruh variasi aturan bisnis dan persyaratan bercabang telah divalidasi.
5. *Graph-Based Testing Methods*: Teknik ini memetakan hubungan antarobjek atau fungsi ke dalam bentuk grafik untuk memastikan bahwa setiap koneksi dan interaksi antarfitur dalam aplikasi berjalan dengan fungsional.

Teknik-teknik tersebut memungkinkan penguji untuk merancang skenario uji yang efektif tanpa harus mengetahui alur kode program di dalamnya. Penggunaan teknik yang tepat, seperti *boundary value analysis*, sangat membantu dalam menemukan kesalahan pada nilai batas yang sering kali menjadi titik lemah dalam sebuah logika fungsional perangkat lunak [20].

Selain kemudahannya, metode ini memiliki keuntungan dan kekurangan tersendiri. Salah satu kelebihanannya adalah kemampuannya dalam membantu menemukan aspek fungsional yang tidak terpenuhi berdasarkan spesifikasi kebutuhan yang telah ditetapkan sejak awal pengembangan [21]. Namun, kekurangan utama dari *black box testing* terletak pada keterbatasan cakupan pengujian yang tidak dapat dilakukan secara menyeluruh terhadap seluruh jalur logika internal karena pengetahuan penguji yang terbatas mengenai struktur kode perangkat lunak yang sedang diuji. Hal ini menyebabkan risiko adanya jalur kode tertentu yang tidak terevaluasi selama proses pengujian berlangsung [22].

## 2.7 Penelitian Terdahulu

Tabel 2.4 berikut ini merangkum beberapa penelitian terdahulu yang berkaitan dengan penelitian pada tugas akhir ini.

Tabel 2.4 Penelitian Terdahulu

| No. | Judul                                                                                               | Peneliti                                    | Metode                    | Hasil                                                                                              |
|-----|-----------------------------------------------------------------------------------------------------|---------------------------------------------|---------------------------|----------------------------------------------------------------------------------------------------|
| 1   | HajjConnect: Sistem manajemen <i>travel</i> haji & umroh berbasis <i>web</i> dengan fitur pemesanan | A. A. Kurniawan,<br>M. Niswah,<br>A. Risman | <i>Website</i> ,<br>MySQL | Meningkatkan efisiensi pendaftaran serta pengelolaan data jemaah secara terpusat dan terintegrasi. |

| No. | Judul                                                                                                                                   | Peneliti                                | Metode                                               | Hasil                                                                                                                      |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
|     | dan pengelolaan data jemaah (2025) [15]                                                                                                 |                                         |                                                      |                                                                                                                            |
| 2   | Implementasi otentikasi <i>website</i> Node JS Express menggunakan <i>Passport</i> (2023) [18]                                          | S. A. B. Cahyono, Sucipto, R. Firliana  | <i>Node.js</i> , <i>Express.js</i> , <i>Passport</i> | Memberikan referensi teknis mengenai sistem keamanan dan otentikasi admin pada sisi <i>backend</i> secara optimal.         |
| 3   | Sistem informasi geografis untuk pemetaan lokasi masjid dan meunasah menggunakan algoritma Haversine berbasis <i>website</i> (2025) [3] | A. Prayoga, Fadlisyah, H. A. K. Aidilof | SIG, Algoritma Haversine                             | Memberikan landasan teori mengenai visualisasi titik lokasi secara akurat pada peta digital yang interaktif.               |
| 4   | Perancangan aplikasi sistem manajemen haji dan umroh berbasis <i>website</i> (2025) [4]                                                 | F. Hidayatulloh, R. Komalasari          | <i>Website</i> , UML                                 | Memberikan gambaran visual alur proses bisnis manajemen <i>travel</i> yang terstruktur bagi admin menggunakan diagram UML. |
| 5   | Implementasi peta interaktif berbasis <i>web</i> untuk lokasi SPBU Pertamina di Kota Padang (2025) [9]                                  | W. P. Yuza, Zulfahmi, dkk.              | <i>Website</i> , Peta Interaktif                     | Menyajikan informasi lokasi yang efisien dan interaktif melalui integrasi peta digital yang mudah dipahami.                |

## 2.8 PT Masar Alsama Global

PT Masar Alsama Global, atau yang dikenal dengan nama Almasar Travel Umrah, merupakan perusahaan penyelenggara perjalanan ibadah umrah yang resmi didirikan pada tanggal 20 Mei 2024. Di bawah kepemimpinan Hasbi Anshori Rambe sebagai Direktur Utama, perusahaan ini berpusat di Jl. Karya Kasih, Komplek Palem Indah Johor, No. 2A, Kecamatan Medan Johor, Kota Medan, Sumatera Utara. Almasar Travel Umrah hadir

dengan dedikasi tinggi untuk memberikan pengalaman ibadah yang amanah, nyaman, dan berkesan bagi masyarakat melalui standar pelayanan terbaik (*service excellence*) yang mencakup seluruh tahapan perjalanan, mulai dari pendaftaran hingga kepulangan jemaah ke tanah air.

Salah satu pilar kekuatan utama PT Masar Alsama Global terletak pada kualitas bimbingan ibadahnya yang kredibel. Perusahaan didukung oleh tenaga pembimbing atau *muthawif* profesional yang merupakan alumni dari Universitas Al-Azhar, Kairo, Mesir. Keberadaan para tenaga ahli ini menjamin bahwa setiap rangkaian ibadah jemaah terlaksana sesuai dengan tuntunan syariat secara mendalam. Kualitas pelayanan yang unggul ini telah dibuktikan dengan keberhasilan perusahaan dalam memberangkatkan banyak jemaah dalam waktu yang relatif singkat sejak mulai beroperasi, yang sekaligus mencerminkan tingkat kepercayaan masyarakat yang tinggi terhadap kredibilitas manajemen perusahaan.

Saat ini, PT Masar Alsama Global mengelola data sekitar 300 jemaah umrah. Kebutuhan pengelolaan data ini memerlukan ketelitian dalam pencatatan profil jemaah, verifikasi dokumen persyaratan, pemantauan transaksi keuangan, serta pengelolaan stok perlengkapan umrah (logistik) untuk memantau ketersediaan barang yang akan diberikan kepada jemaah. Pengelolaan data tersebut dipengaruhi oleh sebaran domisili jemaah yang tersebar mulai dari Kota Medan, Perdagangan, Rantauprapat, hingga Kabupaten Labuhanbatu Utara (Labura). Proses bisnis perusahaan mencakup keseluruhan siklus perjalanan, mulai dari pendaftaran, verifikasi dokumen, penyusunan laporan keberangkatan, hingga kepulangan ke tanah air.

Dalam operasionalnya, manajemen menghadapi hambatan administratif akibat ketergantungan pada koordinasi manual melalui aplikasi pesan instan. Pola kerja tersebut menyebabkan data tidak terintegrasi dan berisiko tinggi terhadap kehilangan informasi penting. Selain itu, staf administrasi harus melakukan penyusunan data secara manual ke dalam berbagai format file yang terpisah, sehingga proses kerja menjadi tidak efisien dan rentan terhadap kesalahan manusia (*human error*), kondisi ini juga terjadi karena ketiadaan sistem informasi terpusat. Oleh karena itu, diperlukan pengembangan sistem informasi yang mampu mengotomatisasi proses administrasi dan menyediakan basis data terpusat, sehingga perusahaan dapat meningkatkan efisiensi operasional serta kualitas pelayanan kepada jemaah.