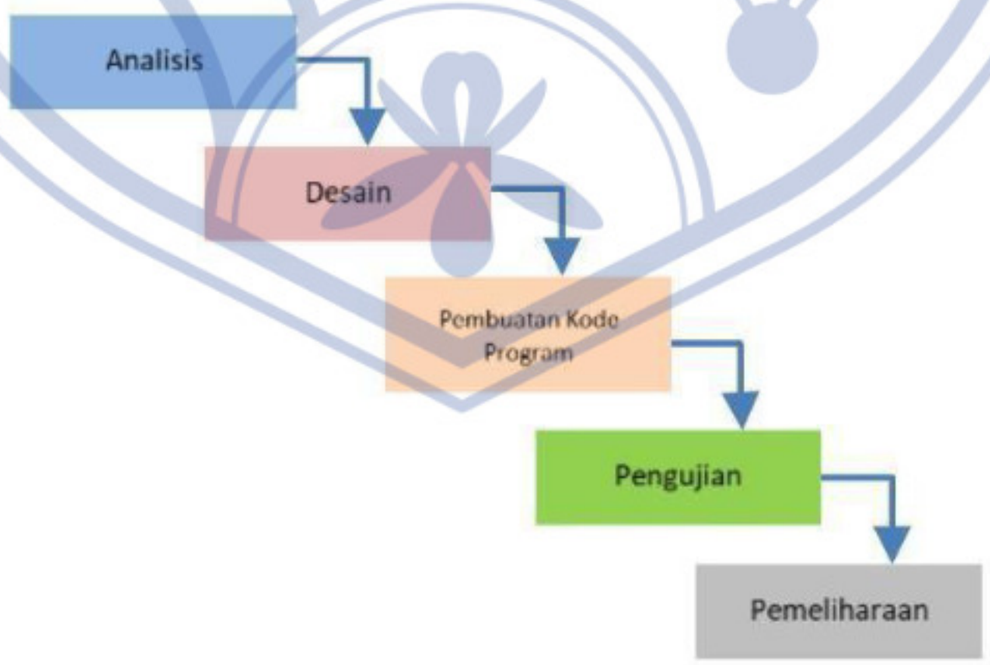
BAB II
KAJIAN LITERATUR

2.1 Metode Pengembangan Sistem

Pada penelitian ini metode yang digunakan dalam pengembangan sistem adalah metode *Waterfall* yang merupakan salah satu model dalam *Software Development Life Cycle (SDLC)* yang menerapkan alur kerja linear dan berurutan, di mana setiap fase pengembangan harus diselesaikan sepenuhnya sebelum lanjut ke tahap berikutnya. Model ini dirancang berdasarkan konsep bahwa proses pengembangan dapat dibagi menjadi fase terpisah seperti analisis kebutuhan, desain sistem, implementasi *(coding)*, pengujian, dan pemeliharaan, yang berjalan dari atas ke bawah seperti aliran air terjun. Karena strukturnya yang sistematis, *Waterfall* memudahkan tim pengembang untuk mendokumentasikan setiap tahap secara formal serta meminimalkan ambiguitas dalam spesifikasi proyek, sehingga proses perencanaan dan pengendalian proyek menjadi lebih terstruktur dibandingkan metode lain yang lebih iteratif. Model ini sering digunakan dalam pengembangan sistem informasi di mana kebutuhan awal sudah cukup jelas dan perubahan selama proses pengembangan relatif sedikit. Studi terbaru juga menunjukkan bahwa pendekatan *Waterfall* tetap relevan dalam konteks sistem informasi berbasis web apabila ruang lingkup proyek dan kebutuhan fungsional telah terdefinisi secara detail sejak awal, sehingga risiko revisi besar di tengah proyek dapat diminimalkan [6]. Berikut ini merupakan gambar ilustrasi alur pengembangan menggunakan metode *Waterfall*.



Gambar 0.1 Ilustrasi alur pengembangan menggunakan metode Waterfall [7]

Selain itu, *Waterfall* memberikan keuntungan berupa kontrol manajemen yang kuat, di mana setiap tahapan memiliki keluaran *(deliverables)* yang terukur dan diuji sebelum tahap

berikutnya dimulai. Hal ini membantu organisasi yang menekankan dokumentasi formal dan kepatuhan terhadap standar teknis, terutama pada proyek berskala menengah sampai besar yang tidak memerlukan perubahan kebutuhan secara intensif setelah fase awal. Namun demikian, literatur juga menggarisbawahi bahwa model *Waterfall* memiliki keterbatasan dalam hal fleksibilitas, karena struktur liniernya membuat adaptasi terhadap perubahan kebutuhan atau revisi desain menjadi relatif sulit dibandingkan dengan metodologi iteratif seperti *Agile*. Perbandingan antara metode *Waterfall* dan metode lain seperti *Rapid Application Development (RAD)* atau *Agile* dalam sejumlah penelitian menunjukkan bahwa *Waterfall* unggul dalam hal perencanaan awal dan kestabilan lingkungan pengembangan, namun kurang responsif terhadap kebutuhan pengguna yang berubah-ubah di tengah jalan, sehingga pemilihan metode harus disesuaikan dengan karakteristik kebutuhan proyek yang akan dikembangkan [7], [8].

2.2 Sistem Informasi

Sistem informasi merupakan sekumpulan komponen yang saling terkait yang dirancang untuk mengumpulkan, mengelola, menyimpan, memproses, serta menyebarkan informasi guna mendukung pengambilan keputusan dalam organisasi. Dalam berbagai penelitian terkini, sistem informasi berperan sebagai mekanisme utama dalam mengelola data dan menghasilkan informasi yang efektif, sehingga keberadaannya dapat meningkatkan akurasi dan efisiensi proses organisasi. Kajian studi literatur menunjukkan bahwa sistem informasi tidak hanya bergantung pada teknologi, tetapi juga melibatkan manusia dan prosedur kerja yang sistematis untuk mencapai tujuan organisasi secara optimal [9].

Selain itu, sistem informasi modern juga berperan penting dalam menunjang pengambilan keputusan dan operasional organisasi secara strategis. Implementasi sistem informasi yang baik dapat mempercepat proses analisis data, meningkatkan koordinasi antar unit kerja, dan mendukung perencanaan strategis yang responsif terhadap perubahan lingkungan bisnis atau layanan publik. Penelitian-penelitian yang sama menunjukkan bahwa pemanfaatan sistem informasi secara efektif memberikan kontribusi signifikan terhadap peningkatan kinerja organisasi dalam menghadapi tantangan era digital saat ini [10].

2.3 Sistem Informasi Pemerintahan

Pengembangan sistem informasi pemerintahan merupakan bagian integral dari upaya transformasi digital pelayanan publik. Sistem informasi pemerintah tidak hanya berfungsi sebagai media penyedia layanan administrasi, tetapi juga mendukung transparansi,

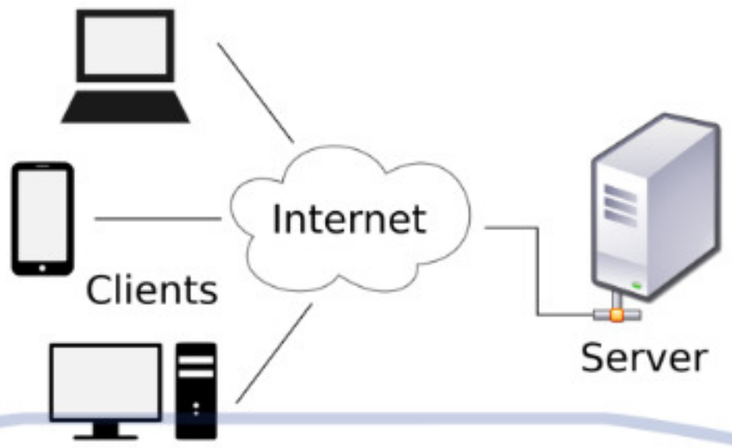
akuntabilitas, dan efektivitas kinerja birokrasi. Transformasi digital dalam pelayanan publik melalui *e-government* menuntut adanya perubahan tata kelola yang lebih responsif, efisien, dan berbasis data. Penelitian sebelumnya mengemukakan bahwa penerapan *e-government* bukan sekadar digitalisasi prosedural, melainkan perubahan sistemik dalam penyampaian layanan publik berbasis prinsip keberlanjutan, inklusi sosial, dan keadilan akses layanan [11]. Temuan penelitian tersebut menegaskan perlunya pengintegrasian teknologi informasi dalam proses pelayanan publik agar pemerintah dapat memberikan pelayanan yang cepat serta menjangkau masyarakat yang selama ini terkendala akses fisik maupun administratif.

Penelitian lainnya pada implementasi aplikasi layanan publik daring di lingkungan pemerintah daerah menunjukkan bahwa sistem informasi berbasis web dapat menjadi alat inovasi layanan publik yang meningkatkan kualitas interaksi pemerintah–masyarakat melalui penyederhanaan proses administrasi, peningkatan kecepatan respon layanan, serta kemudahan pelacakan status layanan secara *real-time*. Integrasi sistem informasi pemerintahan juga memungkinkan pengelolaan basis data penduduk yang lebih akurat dan terstruktur sehingga dapat mengurangi duplikasi pencatatan serta kesalahan penginputan data [12]. Dua studi tersebut memberikan landasan bahwa pengembangan aplikasi web pada konteks pemerintahan tingkat kelurahan/desa berpotensi memberikan dampak signifikan terhadap peningkatan kualitas pelayanan publik, terutama pada aspek kecepatan, transparansi, dan efisiensi administrasi, yang selama ini menjadi tantangan utama dalam penyelenggaraan pelayanan di tingkat pemerintahan terdepan.

2.4 Teknologi Sistem Informasi Berbasis Web

Sistem informasi berbasis web adalah suatu sistem teknologi informasi yang memanfaatkan teknologi internet untuk mengelola data dan layanan informasi yang dapat diakses melalui browser tanpa perlu instalasi perangkat lunak pada sisi klien. Sistem ini dibangun dengan arsitektur *client–server*, di mana pengguna (*client*) mengirimkan permintaan melalui *browser* dan *server* kemudian memproses permintaan tersebut melalui protokol *HTTP* untuk mengirim balikan berupa tampilan dan data yang diperlukan pengguna. Pendekatan ini memungkinkan sistem dapat diakses kapan saja dan dari mana saja selama terhubung dengan jaringan internet serta bersifat *platform-independent*, sehingga tidak terikat pada sistem operasi tertentu di sisi pengguna. Keunggulan ini menjadikan sistem informasi berbasis web lebih efisien dan efektif dibandingkan aplikasi desktop konvensional, karena akses mudah dan pemeliharaan terpusat hanya perlu dilakukan di sisi *server* tanpa memerlukan *update* di

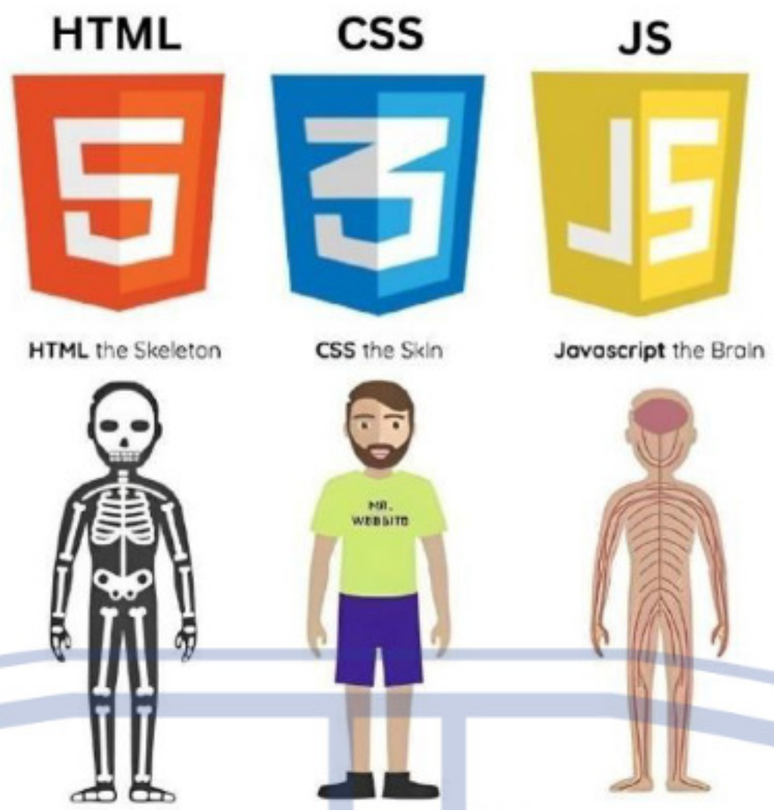
perangkat pengguna satu per satu [13]. Berikut ini merupakan ilustrasi gambar dari *client-server*.



Gambar 0.2 Ilustrasi *Client-Server* [14]

2.4.1 *Client* (*Frontend*)

Frontend web development merupakan disiplin dalam pengembangan aplikasi web yang berfokus pada perancangan dan implementasi antarmuka yang digunakan untuk berinteraksi langsung dengan pengguna melalui *browser*. Pada proses pengembangannya, *frontend* memanfaatkan teknologi dasar seperti *HTML* untuk struktur konten, *CSS* untuk penyajian gaya visual, dan *JavaScript* sebagai bahasa pemrograman utama untuk memberikan interaktivitas pada elemen halaman web. Penggunaan ketiga komponen inti tersebut menjadi fondasi agar situs web dapat dibangun secara konsisten, responsif, serta adaptif pada berbagai perangkat. Studi menunjukkan bahwa pengembangan *frontend* modern harus memperhatikan performa pemuatan halaman dan responsivitas terhadap interaksi pengguna, karena kedua aspek tersebut secara langsung memengaruhi kualitas pengalaman pengguna [14]. Hal ini semakin relevan mengingat meningkatnya kompleksitas aplikasi web dan tuntutan pengguna terhadap kecepatan akses. Berikut ini merupakan gambar ilustrasi dari *HTML*, *CSS* dan *Javascript* dalam pengembangan web.



Gambar 0.3 Ilustrasi dari *HTML, CSS* dan *Javascript* [16]

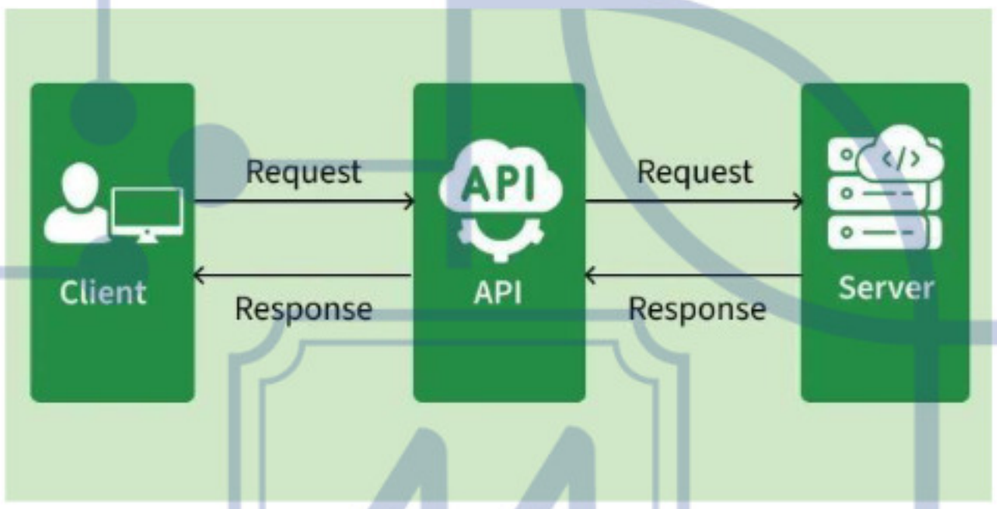
Seiring meningkatnya kebutuhan aplikasi web berskala besar dan kompleks, pengembang cenderung memanfaatkan framework *JavaScript* modern seperti *React, Angular,* dan *Vue* sebagai solusi untuk mempercepat pengembangan sekaligus meningkatkan skalabilitas dan keteraturan kode. *Framework* modern menyediakan konsep berbasis komponen dan fitur manajemen *state* yang mampu membantu pengembang dalam menjaga konsistensi dan struktur aplikasi, terutama pada aplikasi dengan banyak halaman dan fungsi interaktif. Hasil penelitian komparatif menunjukkan perbedaan performa yang signifikan antara *framework*, sehingga pemilihan *framework frontend* perlu mempertimbangkan faktor kebutuhan proyek, arsitektur, dan tujuan performa aplikasi [15]. Perbandingan ini menunjukkan bahwa pemilihan *framework frontend* perlu mempertimbangkan faktor kebutuhan proyek, arsitektur, dan tujuan performa aplikasi.Adapun karakteristik dari masing-masing framework tersebut adalah:

1. React: unggul dalam pembaruan UI dinamis melalui virtual DOM, fleksibel, dan mudah diintegrasikan dengan pustaka eksternal.
2. Vue: memiliki rendering yang cepat, ringan dalam penggunaan memori, dan relatif mudah dipelajari oleh pengembang.
3. Angular: menawarkan arsitektur terpadu dengan fitur bawaan lengkap serta lebih cocok untuk aplikasi berskala besar dengan kompleksitas tinggi.

2.4.2 *Server (Backend)*

Backend atau *server-side development* merujuk pada bagian dari pengembangan perangkat lunak yang berjalan di sisi *server*, bertanggung jawab menangani logika aplikasi, pengelolaan data, dan interaksi dengan basis data yang tidak terlihat oleh pengguna akhir.

Komponen inti *backend* mencakup *server* itu sendiri, *database*, dan *API (Application Programming Interface)* yang menghubungkan *server* dengan klien atau antarmuka pengguna. Dalam arsitektur *backend, server* menerima permintaan *(request)* dari klien, memproses logika bisnis, berkomunikasi dengan basis data, lalu mengirimkan respons kembali ke klien, sering dalam format *JSON (Javascript Object Notation)* pada aplikasi web modern yang berbasis *RESTful*. Pendekatan ini memungkinkan *backend* menjadi tulang punggung aplikasi yang dapat diakses dari berbagai platform sambil memisahkan tanggung jawab antara antarmuka dan *server* aplikasi [16]. Berikut merupakan ilustrasi gambar *API(Application Programming Interface) server*.



Gambar 0.4 Ilustrasi *API (Application Programming Interface)* [17]

Dalam praktik pengembangan *backend*, performa dan efisiensi menjadi aspek penting karena beban kerja server langsung memengaruhi pengalaman pengguna secara keseluruhan. Penelitian komparatif terhadap *framework backend* populer seperti *Spring Boot, Flask, Express.js, Laravel,* dan *Gin* menunjukkan bahwa pilihan *framework* dan bahasa pemrograman memiliki dampak signifikan terhadap kinerja *REST API*. Performa yang baik dalam menangani *request* dan *response* sangat penting untuk aplikasi berskala besar, terutama dalam lingkungan yang memerlukan banyak operasi baca/tulis dan koneksi simultan. Hal ini mencerminkan kebutuhan bagi *backend* untuk mendukung skalabilitas dan stabilitas layanan dalam situasi beban tinggi [18]. Adapun karakteristik dari beberapa framework tersebut antara lain:

1. Spring Boot (Java/Kotlin): memiliki performa tinggi, dukungan ekosistem enterprise, serta kuat dalam pengelolaan proses kompleks dan transaksi besar.
2. Flask (Python): bersifat ringan (microframework), mudah dikembangkan, tetapi performanya cenderung lebih rendah dibandingkan bahasa yang dikompilasi.
3. Express.js (Node.js): non-blocking I/O dan event-driven, unggul dalam penanganan koneksi simultan serta sangat populer untuk aplikasi berbasis REST API.

4. Laravel (PHP): menawarkan struktur pengembangan yang lengkap dan fitur built-in untuk manajemen data, namun performanya sering bergantung pada optimasi server dan konfigurasi aplikasi.
5. Gin (Golang): dikenal memiliki performa sangat baik karena keunggulan bahasa Go yang dikompilasi serta efisien dalam menangani request parallel dengan latensi rendah.

Server backend sering dioptimalkan melalui berbagai strategi infrastruktur seperti *load balancing* dan *clustering* untuk meningkatkan ketersediaan *(high availability)* dan kapasitas penanganan trafik. Implementasi arsitektur server cluster dengan komponen *load balancing*, mekanisme *failover*, dan replikasi basis data telah terbukti meningkatkan keandalan sistem dalam skenario *real-time* data *processing* pada aplikasi kritis. Pendekatan ini menunjukkan bahwa *backend* tidak hanya berperan sebagai eksekutor logika aplikasi, tetapi juga komponen infrastruktur yang memerlukan pengelolaan performa, keandalan, dan keberlanjutan layanan [19].

2.5 Pengelolaan Data Kependudukan Digital

Pengelolaan data kependudukan digital merupakan bagian penting dari transformasi administrasi pemerintahan yang bertujuan meningkatkan efektivitas pelayanan publik. Data kependudukan yang bersifat komprehensif dan terintegrasi memungkinkan pemerintah mengelola informasi penduduk secara lebih akurat, cepat, dan responsif terhadap kebutuhan masyarakat. Sistem pengelolaan data kependudukan digital berperan dalam proses pencatatan, penyimpanan, pembaruan, dan distribusi data yang digunakan oleh berbagai unit layanan publik, seperti Dinas Kependudukan dan Pencatatan Sipil (Disdukcapil), pelayanan administrasi desa/kelurahan, serta unit pemerintahan lainnya.

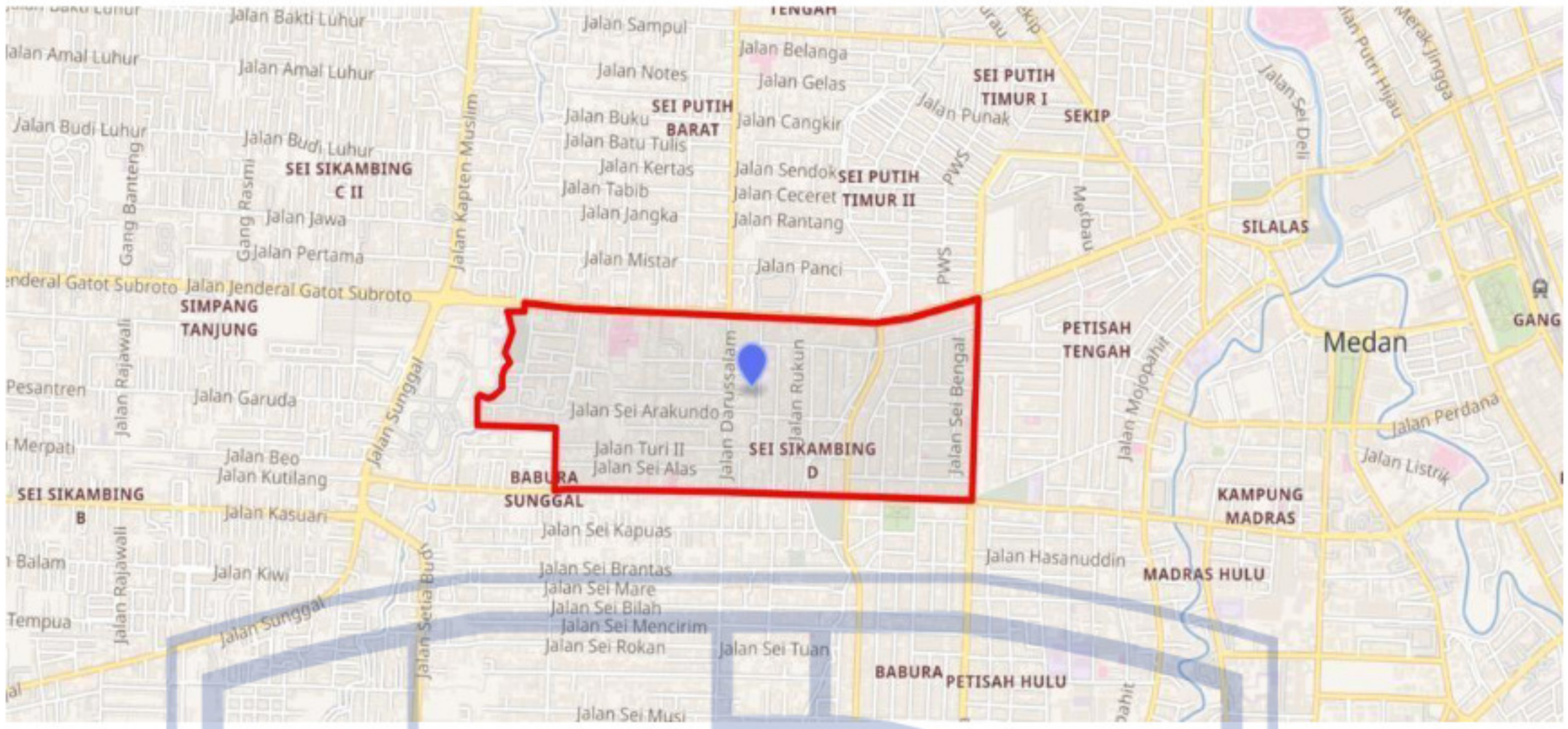
Dalam implementasinya, pengelolaan data kependudukan digital tidak hanya mencakup pemanfaatan basis data terpusat, tetapi juga mencakup integrasi berbagai aplikasi layanan publik agar data dapat diakses secara lintas sistem dalam kerangka pemerintahan elektronik. Studi menunjukkan bahwa penggunaan sistem informasi pengelolaan data kependudukan di Dinas Kependudukan dan Pencatatan Sipil Sleman Kabupaten memungkinkan terciptanya basis data yang tersusun secara terstruktur dan terorganisir, yang dapat digunakan untuk monitoring populasi dan mendukung pengambilan keputusan pelayanan. Hal ini memperlihatkan pentingnya sistem pengelolaan data kependudukan yang akurat, terjamin konsistensinya, dan mudah diakses oleh unit kerja pemerintahan terkait [20].

Selain pengorganisasian basis data, proses digitalisasi administrasi kependudukan juga berdampak langsung terhadap kualitas pelayanan publik. Penelitian tentang transformasi pelayanan administrasi kependudukan di Desa Tempurejo mendemonstrasikan bagaimana digitalisasi proses pencatatan dan pengolahan data kependudukan mendekatkan layanan kepada masyarakat desa yang sebelumnya harus menempuh jarak jauh untuk mengurus dokumen. Hal ini menunjukkan bahwa pengelolaan data kependudukan digital tidak hanya berfokus pada integrasi teknologi, tetapi juga memperhatikan aspek aksesibilitas layanan bagi masyarakat [21].

Pengelolaan data kependudukan digital juga berkaitan dengan inovasi layanan publik di pemerintah daerah besar. Studi tentang inovasi manajemen administrasi kependudukan di DKI Jakarta menggunakan aplikasi berbasis informasi elektronik menegaskan bahwa penggunaan teknologi dalam pengelolaan data mampu mempermudah masyarakat dalam mengakses informasi kependudukan dan layanan administratif tanpa harus datang langsung ke kantor. Integrasi data kependudukan dalam sistem informasi publik mempercepat proses pelayanan serta membantu pemerintah dalam menyediakan layanan yang lebih transparan, akuntabel, dan efisien [22].

2.6 Kelurahan Sei Sikambing D

Kelurahan Sei Sikambing D merupakan salah satu kelurahan administratif yang berada di dalam wilayah Kecamatan Medan Petisah, Kota Medan, Provinsi Sumatera Utara. Kelurahan ini termasuk kawasan yang strategis di inti Kota Medan, dengan jarak kurang lebih 1,4 km dari titik nol kota dan memiliki luas wilayah sekitar 0,915 km² yang terbagi atas sejumlah lingkungan administratif. Kelurahan Sei Sikambing D berbatasan langsung dengan beberapa kelurahan lainnya seperti Sei Putih Barat, Sei Putih Tengah, dan Petisah Tengah sehingga posisi geografisnya cukup sentral dalam kegiatan sosial dan pemerintahan di kecamatan ini. Kelurahan ini memiliki jumlah penduduk yang padat, dengan ribuan jiwa yang tersebar di beberapa lingkungan, menunjukkan dinamika sosial yang kompleks dan peran penting kelurahan dalam pelayanan publik.

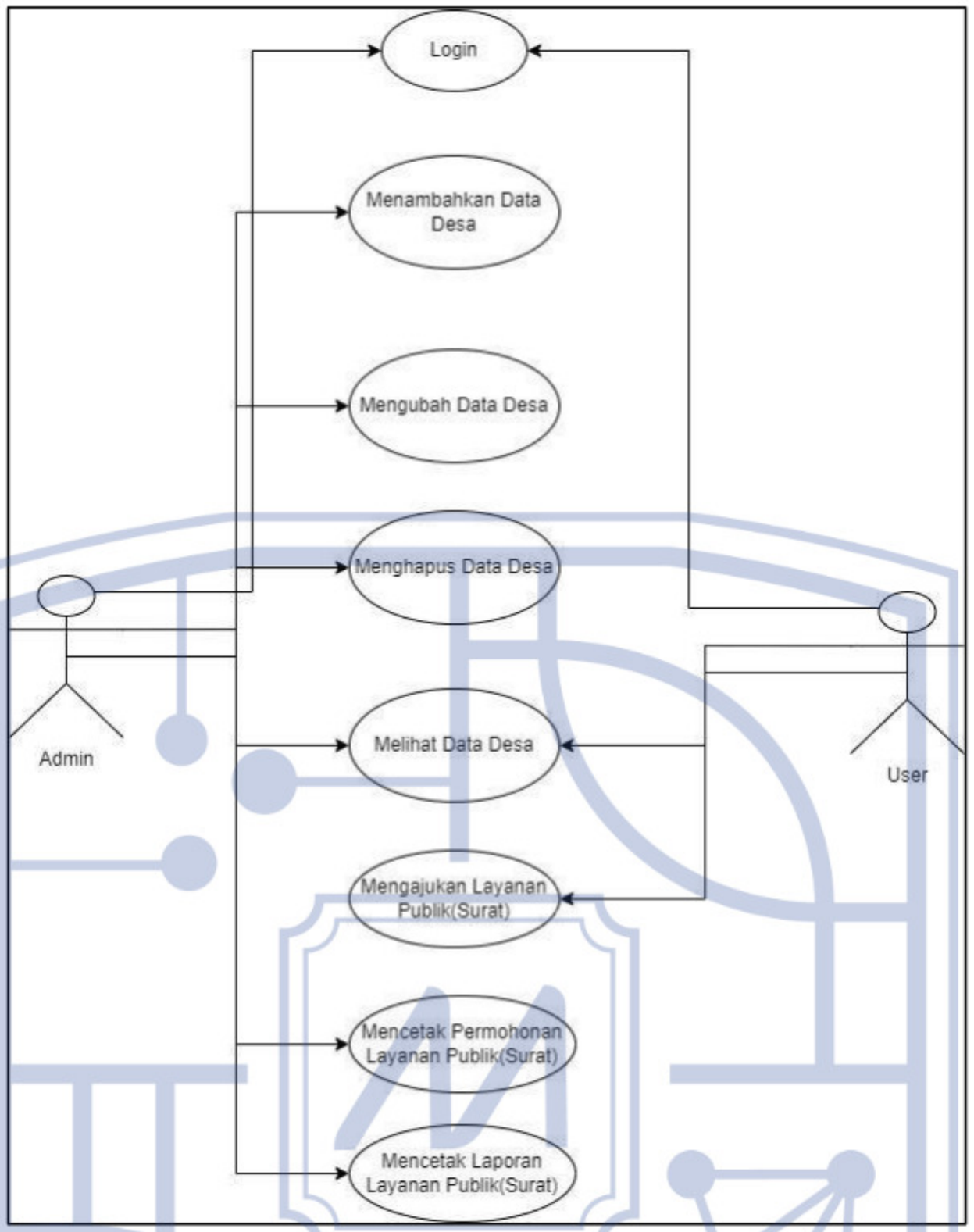


Gambar 0.5 Peta Sei Sikambing D

Sebagai unit pemerintahan terdekat dengan masyarakat, Kantor Kelurahan Sei Sikambing D berfungsi sebagai pusat layanan administrasi publik yang meliputi pengurusan surat-surat resmi seperti surat domisili, surat pindah, surat keterangan usaha dan izin lainnya yang dibutuhkan warga. Posisi kantornya di Jl. Sei Bahkapuran No. 16 juga menunjukkan peran administratifnya yang mudah diakses oleh masyarakat setempat. Kelurahan ini tidak hanya bertanggung jawab atas administrasi kependudukan, tetapi juga menjadi basis pelaksanaan program pemerintahan seperti pemberdayaan UMKM dan kegiatan masyarakat yang mendukung peningkatan kesejahteraan warga lokal. Hal ini sesuai dengan fungsi kelurahan sebagai ujung tombak pelayanan kepada masyarakat di tingkat paling dasar administrasi pemerintahan kota [23].

2.7 *UML (Unified Modeling Language)*

Unified Modeling Language (UML) merupakan bahasa pemodelan visual standar yang digunakan untuk merancang, mendokumentasikan, dan memvisualisasikan sistem perangkat lunak dan sistem informasi, sehingga mempermudah komunikasi antara analis, desainer, dan pengembang dalam setiap tahap pengembangan sistem. Dalam banyak penelitian penerapan sistem informasi berbasis web maupun *mobile*, *UML* sering dimanfaatkan untuk menggambarkan struktur dan perilaku sistem melalui berbagai jenis diagram seperti *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*, yang membantu menetapkan kebutuhan fungsional dan alur interaksi sistem sebelum proses implementasi dimulai [24]. Berikut ini merupakan gambar contoh penggunaan *UML use case Diagram*.



Gambar 0.6 Contoh Penggunaan UML Use Case Diagram [25]

Selain itu, penggunaan *UML* juga terbukti meningkatkan kualitas perancangan dan efisiensi pengembangan sistem informasi dengan memberikan gambaran yang jelas mengenai elemen-elemen penting sistem serta interaksinya dengan aktor luar. Studi literatur menunjukkan bahwa *UML* memfasilitasi identifikasi kebutuhan pengguna dan menjadikan proses analisis serta desain sistem lebih terstruktur dan terstandarisasi, sehingga risiko kesalahan pemahaman spesifikasi dapat diminimalkan dan proses pengembangan dapat dijalankan lebih efektif sesuai kebutuhan organisasi atau layanan yang dikembangkan [26].

2.8 *PIECES*

PIECES merupakan sebuah kerangka kerja yang sering digunakan dalam analisis kebutuhan sistem informasi, yang juga efektif untuk mengidentifikasi kebutuhan non-fungsional dalam pengembangan sistem. *PIECES* sendiri merujuk pada enam aspek penting yaitu *Performance* (kinerja), *Information & Data* (informasi dan data), *Economic* (efisiensi ekonomi), *Control* (kontrol dan keamanan), *Efficiency* (efisiensi operasional), dan *Service*

(pelayanan). Kerangka ini digunakan untuk menilai sejauh mana kebutuhan non-fungsional terpenuhi oleh sistem dan menjadi dasar perbandingan antara sistem lama dan sistem yang direncanakan, sehingga dapat membantu menentukan prioritas perbaikan serta memastikan bahwa sistem mampu beroperasi dengan efektif serta memenuhi ekspektasi pengguna [27].

Dalam praktiknya, *PIECES* membantu tim pengembang dan analis untuk menggali karakteristik non-fungsional dengan cara yang lebih terstruktur dan terukur. Misalnya, aspek *performance* memfokuskan pada waktu respons dan kemampuan sistem dalam menangani beban kerja, sedangkan aspek control memeriksa mekanisme keamanan dan hak akses untuk memastikan data tidak disalahgunakan. Selain itu, aspek *service* melihat kemampuan sistem dalam memberikan layanan yang mudah digunakan dan memuaskan bagi pengguna. Dengan pendekatan ini, analisis kebutuhan non-fungsional tidak hanya menyangkut deskripsi umum, tetapi juga dapat diukur secara konkret sehingga memudahkan dalam perancangan, pengujian, dan evaluasi sistem [28]. Berikut ini merupakan contoh tabel PIECES.

Tabel 0.1 Contoh Tabel Analisis PIECES [28]

Analisis	**Sistem Berjalan**	**Sistem Usulan**
Performance	Tampilan dengan iklan-iklan yang muncul di atas barang detail yang membuat teralihkan pandangan untuk melihat detail barang.	Tampilan website yang dibuat fokus pada detail barang yang dijual tersebut, responsif, dan nyaman untuk dilihat.
Information	Informasi yang diberikan tidak sesuai seperti perihal harga yang tertulis tidak sesuai dengan harga sebenarnya.	Memberikan informasi yang jelas dan sesuai.
Economy	Hanya akses internet yang dibutuhkan dan pembayaran iklan barang untuk membuat barang penjualan lebih terekspos.	Hanya akses internet yang dibutuhkan dengan perangkat yang ada untuk mengakses website.
Control	Bebas dan terbuka untuk pembayaran dan mengupload barang yang dilakukan sehingga dapat memicu terjadinya penipuan.	Dapat mencegah aksi penipuan dengan pembayaran yang dilakukan secara langsung dalam sistem dan melakukan validasi barang.

Efficiency	Kurang efisien dalam hal pembayaran dikarenakan tidak tersedia langsung dalam website.	Memberikan kemudahan dalam bertransaksi secara langsung pada website sehingga lebih efisien.
Service	Hanya memfasilitasi untuk mempromosikan atau menampilkan barang.	Memberikan layanan untuk menjual barang, bertransaksi, dan pengelolaan informasi dengan baik dan efektif.

2.9 Database

Database merupakan elemen penting dalam sistem informasi karena berfungsi sebagai tempat penyimpanan, pengelolaan, dan pengambilan data secara efisien. *Model database* yang umum digunakan terbagi menjadi dua kelas besar: relasional *(SQL)* dan non-relasional *(NoSQL)*. *Database* relasional memiliki struktur yang terorganisir dengan baik menggunakan tabel dan relasi antar tabel, serta dijalankan dengan bahasa kueri struktur seperti *SQL*. Di sisi lain, *database NoSQL* menyediakan fleksibilitas model data tanpa skema tetap, sehingga cocok untuk menangani data semi-terstruktur dan skala besar yang sering ditemui pada aplikasi modern seperti media sosial dan sistem berbasis cloud. Studi literatur komparatif menunjukkan bahwa meskipun *SQL* unggul dalam hal integritas data dan konsistensi transaksional, *NoSQL* menawarkan skalabilitas horizontal dan performa tinggi dalam memproses volume data besar dan struktur data yang kompleks [29].

Komparasi kinerja antara *database* relasional dan *NoSQL* telah menjadi fokus penting dalam literatur akademik, terutama dalam konteks kebutuhan aplikasi modern yang berskala besar dan *real-time*. Misalnya, studi sistematis menunjukkan bahwa *NoSQL* mampu memberikan fleksibilitas tinggi dan skalabilitas horizontal dalam pengelolaan data besar, sementara *database* relasional tetap relevan pada pengolahan *transaction-processing* yang membutuhkan konsistensi kuat [30]. Penelitian lain yang membandingkan performa *query* antara *MySQL* (relasional) dan *MongoDB (NoSQL)* juga menegaskan bahwa dalam beberapa skenario skala besar, *NoSQL* dapat memberikan keunggulan performa, sedangkan relasional unggul pada operasi yang memerlukan join kompleks antar tabel [30].

Selain itu, penelitian dalam *performance analysis NoSQL* menunjukkan bahwa berbagai teknologi *NoSQL* seperti *MongoDB, Cassandra, Redis,* dan *Couchbase* memiliki kekuatan yang berbeda tergantung pada pola akses data dan beban kerja yang diuji. Misalnya *MongoDB* menunjukkan fleksibilitas dan integrasi yang baik untuk sistem berbasis cloud,

sedangkan *Cassandra* unggul pada beban tulis yang tinggi dan tingkat fault tolerance tinggi [31]. Hal ini memperkuat pemahaman bahwa pemilihan model database harus didasarkan pada kebutuhan aplikasi, karakteristik data, serta tujuan performa dan skalabilitas yang diinginkan.

2.10 Blackbox Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada evaluasi kesesuaian fungsi sistem berdasarkan spesifikasi kebutuhan tanpa memeriksa struktur internal kode. Pendekatan ini memberikan input kepada sistem dan memeriksa output yang dihasilkan untuk melihat apakah fungsi berjalan sesuai yang diharapkan. Dengan demikian, pengujian berorientasi pada perilaku eksternal sistem dan memastikan bahwa perangkat lunak memenuhi kebutuhan pengguna pada tingkat antarmuka dan proses.

Teknik *Equivalence Partitioning* adalah salah satu pendekatan dalam *Black Box Testing* yang membagi data input ke dalam kelas ekuivalen untuk menghindari pengujian berulang pada data yang serupa. Dalam penelitian yang dilakukan pada sistem informasi akademik, teknik ini digunakan untuk menguji validasi input dan memastikan bahwa fitur akademik seperti *login*, pengelolaan data mahasiswa, dan proses administrasi dapat berjalan sesuai spesifikasi fungsional. Hasil penelitian menunjukkan bahwa metode ini efektif dalam mengidentifikasi kesalahan fungsional secara sistematis dan efisien [32].

Penerapan *Equivalence Partitioning* dalam pengujian website *Wingpos* juga dikaji untuk memastikan fungsionalitas fitur transaksi dan layanan berbasis web. Penelitian tersebut menemukan bahwa metode *Black Box Testing* dapat mengidentifikasi kesalahan pada proses input transaksi dan validasi data pengguna, sehingga membantu memastikan reliabilitas sistem dari perspektif pengguna. Pendekatan pengujian ini dinilai mampu mengungkap error yang tidak terlihat pada logika internal program, karena fokusnya pada hasil keluaran dan interaksi sistem dengan pengguna [33].