

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Pertama kali pengembangan dilakukan oleh Sakti dan Hermawan (2020) dengan judul *“Implementasi Arsitektur Microservice pada Back End Sistem Informasi Atlantas berbasis Website.”* Penelitian ini memanfaatkan arsitektur microservice untuk menghubungkan pertukaran data antar unit, sekaligus menyebarkan informasi yang dilaporkan oleh aparat kepolisian secara langsung (real-time), berkaitan dengan insiden pelanggaran lalu lintas maupun kecelakaan yang telah terjadi. Sistem yang dikembangkan juga mampu menampilkan informasi pelanggaran dan kecelakaan secara lebih terstruktur dan spesifik.

Penelitian kedua dilakukan oleh Amin (2020) dengan judul *“Implementasi Restful API Menggunakan Arsitektur Microservice untuk Manajemen Tugas Kuliah (Studi Kasus: Mahasiswa STMIK AKAKOM).”* Penelitian ini menerapkan arsitektur microservice dengan dukungan Restful API untuk memfasilitasi proses pengumpulan dan pengelolaan tugas perkuliahan, baik oleh dosen maupun mahasiswa, sehingga manajemen tugas menjadi lebih efisien dan terintegrasi..

Penelitian ketiga dilakukan oleh Zulkarnain dan Maurieftha (2023) dengan judul *“Pengembangan Sistem Informasi Pelayanan Klinik Gigi Praktik Bersama Sawojajar Malang.”* Penelitian ini membahas pengembangan sistem informasi yang dirancang untuk mendukung dan mempermudah pekerjaan manusia, khususnya dalam hal layanan klinik gigi. Sistem tersebut mencakup fitur konsultasi dan reservasi perawatan rutin, serta berfungsi untuk menyediakan informasi terkait penjadwalan dan manajemen pelayanan secara lebih terstruktur..

Penelitian keempat dilakukan oleh Anwar dan Kautsar (2024) yang berjudul *“Arsitektur Perangkat Lunak Berbasis Layanan Mikro pada Sistem Manajemen Informasi Kantin Author.”* Studi kasus dalam pengembangan ini dilakukan pada aplikasi E-Kantin, mengharuskan proses perputaran data antara mitra, pelanggan, serta transaksi berjalan

secara optimal. Dengan penerapan arsitektur *microservices*, alur data dan kecepatan pemrosesan informasi meningkat, sehingga mendukung kinerja tiap layanan secara maksimal. Hasil pengujian menunjukkan bahwa sistem *microservices* yang dikembangkan mampu menjaga efisiensi dan stabilitas layanan, bahkan saat terjadi lonjakan permintaan atau akses dari banyak pengguna. Temuan ini mengindikasikan bahwa arsitektur *microservices* memberikan dampak positif terhadap efisiensi dan kestabilan pengelolaan data dalam aplikasi E-Kantin..

Tabel 2.1 Literasi Terkait

No	Keterangan	
1	Nama Jurnal	: Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer
	Judul	: Implementasi Restfull API menggunakan arsitektur <i>microservice</i> untuk manajemen tugas kuliah (Studi kasus: Mahasiswa STMIK AKAKOM)
	Kesimpulan	: Pada penelitian ini dibuat sistem yang digunakan untuk dosen dan mahasiswa untuk memanajemen tugas kuliah dengan menggunakan arsitektur <i>microservice</i> . Amin (2020)
	Kelebihan	: Menggunakan RESTFull API dan membantu mahasiswa dan dosen dalam memanajemen tugas kuliah.
	Kekurangan	<i>User interface</i> yang perlu diperbaiki.
2	Nama Jurnal	: Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer
	Judul	: Analisis dan Perancangan Sistem Informasi Rekam Medis Poli Gigi (Studi Kasus : Puskesmas Summersari Kecamatan Saradan Kabupaten Madiun)

No	Keterangan
Kesimpulan	: Penelitian ini berguna untuk membuat sebuah sistem yang difokuskan pada pelayanan poli gigi, dengan fungsi utama untuk mendukung proses pencatatan rekam medis pasien. Sistem yang dikembangkan dirancang agar selaras dengan alur bisnis yang diterapkan di puskesmas terkait. Metode Object Oriented Analysis and Design (OOAD) diterapkan pada penelitian ini, sehingga proses analisa dan perancangannya divisualisasikan menggunakan Unified Modeling Language (UML) (Iflahah et al., 2018).
Kelebihan	: hasil dari penelitian ini mencakup berbagai komponen dalam perancangan sistem, antara lain use case diagram, alur proses bisnis, sequence diagram, desain arsitektur sistem, serta analisis dan elemen lainnya yang mendukung pengembangan sistem secara menyeluruh..
Kekurangan	: Perancangan tampilan kurang lengkap.
3	<p data-bbox="227 954 926 1018">Nama Jurnal : Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer</p> <p data-bbox="227 1029 926 1129">Judul : Pembangunan Aplikasi Pendaftaran dan Rekam Medis Pasien Berbasis Website (Studi Kasus : Klinik Medis Elisa Malang)</p> <p data-bbox="227 1141 926 1394">Kesimpulan : Pengembangan sistem berbasis website yang digunakan untuk pencatatan rekam medis sekaligus pendaftaran pasien. Tujuannya adalah untuk mempermudah pengelolaan data dan meningkatkan mutu pelayanan. Dalam proses pengembangannya diterapkan waterfall model sebagai metode Software</p>

No	Keterangan
	<p>Development Life Cycle (SDLC) (Pusparani et al., 2019).</p> <p>Kelebihan : Dalam proses pengembangannya, penelitian ini menggunakan model Waterfall sebagai pendekatan pengembangan sistem. Pengujian perangkat lunak dilakukan dengan menerapkan white box testing, dengan teknik basic path testing untuk memastikan alur logika program berjalan sesuai dengan yang diharapkan.</p> <p>Kekurangan : Desain UML dan basis data belum disajikan secara visual dalam penelitian ini.</p>
<p>4</p> <p>Nama Jurnal</p> <p>Judul</p> <p>Kesimpulan</p> <p>Kelebihan</p> <p>Kekurangan</p>	<p>: Jurnal AKSARA PUBLIC</p> <p>: Rancang Bangun Sistem Informasi Rekam Medis Berbasis Web (Studi Kasus : Praktik Umum Dr. A. Esti Jakarta)</p> <p>: Pengembangan sistem pencatatan medis berbasis web yang meliputi fitur konsultasi dan manajemen obat, dengan pendekatan model Waterfall dan pemodelan menggunakan UML. Teknologi yang digunakan dalam pengembangannya adalah PHP, JavaScript, dan CSS (Rudianto, 2018).</p> <p>: Menerapkan model pengembangan sistem Rapid Application Development (RAD).</p> <p>: Hak akses sistem tidak tersedia bagi pasien..</p>
<p>5</p> <p>Nama Jurnal</p> <p>Judul</p> <p>Kesimpulan</p>	<p>: Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer</p> <p>: Perawatan Berkala Klinik Gigi (Studi Kasus: Klinik Email Dental Care)</p> <p>: Perancangan sistem konsultasi dan reservasi perawatan di Klinik Gigi Email untuk</p>

No	Keterangan
Kelebihan	mendukung pengelolaan jadwal dan layanan pasien. Sistem ini juga menyajikan data yang dapat digunakan staf klinik dalam evaluasi dan peningkatan kualitas pelayanan (Nanda, 2018).oleh staf klinik (Nanda, 2018).
Kekurangan	: Setiap pihak yang terlibat, seperti pasien dan dokter, diberikan hak akses sesuai perannya.. : Tampilan UI terkesan statis dan belum cukup menarik dari segi estetika.
6	<p data-bbox="227 539 674 563">Nama Jurnal : Odonto: Dental Journal</p> <p data-bbox="227 579 926 675">Judul : Design and development of web-based dental electronic medical records according to ministry of health standards</p> <p data-bbox="227 691 926 866">Kesimpulan : Pada Penelitian ini telah berhasil menciptakan aplikasi berbasis web untuk sistem rekam medis elektronik kedokteran gigi yang memiliki spesifikasi standar dari Kementerian Kesehatan. (Wardhana et al., 2023)</p> <p data-bbox="227 882 926 1353">Kelebihan : Ada empat kelebihan yaitu,</p> <ol data-bbox="418 914 926 1353" style="list-style-type: none"> <li data-bbox="418 914 926 1090">1. Efisiensi dan Aksesibilitas: RME memungkinkan akses cepat dan mudah terhadap informasi pasien, yang dapat meningkatkan efisiensi pelayanan kesehatan.</li> <li data-bbox="418 1106 926 1201">2. Keamanan Data: RME mampu mencegah akses ilegal terhadap data pasien melalui penerapan standar keamanan yang baik.</li> <li data-bbox="418 1217 926 1353">3. Integrasi Data: RME memungkinkan integrasi data dari berbagai sumber, yang dapat mendukung pengambilan keputusan klinis yang lebih baik.</li> </ol>

No	Keterangan
Kekurangan	<p>4. Pengurangan Kesalahan: RME dapat mencegah kesalahan medis akibat tulisan tangan dan kehilangan informasi pasien.</p> <p>: Ada empat kekurangan, yaitu</p> <ol style="list-style-type: none"> <li>1. Implementasi sistem RME melibatkan biaya awal yang cukup tinggi, termasuk untuk infrastruktur teknologi dan peningkatan kompetensi sumber daya manusia.</li> <li>2. Kompleksitas Sistem: Sistem RME bisa menjadi kompleks dan memerlukan waktu untuk adaptasi oleh staf medis.</li> <li>3. Masalah Keamanan: Meskipun ada peningkatan keamanan, RME tetap rentan terhadap serangan siber dan pelanggaran data.</li> <li>4. Keterbatasan Infrastruktur: Di beberapa daerah, keterbatasan infrastruktur teknologi informasi dapat menjadi hambatan dalam penerapan RME.</li> </ol>
7	<p>Nama Jurnal : International Journal of Science and Engineering Applications</p> <p>Judul : Web-based Clinic Management System (CMS)</p> <p>Kesimpulan : Penelitian ini menunjukkan bahwa penerapan CMS ini dapat meningkatkan efisiensi, akurasi, dan aksesibilitas catatan pasien serta layanan kesehatan secara keseluruhan. Selain itu, penelitian ini juga menyoroti pentingnya perencanaan teknologi strategis dan penerapan praktik terbaik organisasi dalam pengelolaan informasi kesehatan. Keberhasilan sistem informasi kesehatan dapat memberikan kontribusi besar dalam</p>

No	Keterangan
	mentransformasi organisasi serta meningkatkan kualitas pengambilan keputusan di sektor layanan kesehatan. (Muhammad & Garba, 2019).
Kelebihan	<p>: Ada lima kelebihan, yaitu</p> <ol style="list-style-type: none"> <li>1. Efisiensi dan Akurasi: Sistem ini meningkatkan efisiensi dan akurasi dalam penyimpanan dan pengambilan informasi pasien, yang sebelumnya dilakukan secara manual dan rentan terhadap kesalahan.</li> <li>2. Integrasi Modul: CMS mencakup berbagai modul seperti penjadwalan janji temu, observasi dan diagnosis, rujukan pasien, manajemen apotek, dan konsultasi online, yang semuanya terintegrasi untuk memberikan layanan yang lebih komprehensif.</li> <li>3. Aksesibilitas: Sistem ini memungkinkan akses elektronik ke catatan medis, yang membantu klinisi menghemat waktu dan sumber daya dalam pengelolaan informasi pasien.</li> <li>4. Pengambilan Keputusan: CMS membantu pengambilan keputusan dengan menyediakan informasi yang lengkap dan mudah diakses.</li> <li>5. Peningkatan Layanan: Dengan mengotomatisasi proses manual, sistem ini dapat meningkatkan kualitas layanan klinis dan pengalaman pasien.</li> </ol>
Kekurangan	<p>: Ada lima kekurangan, yaitu</p> <ol style="list-style-type: none"> <li>1. Biaya Implementasi: Untuk membangun dan menjalankan sistem web, dibutuhkan dana awal yang besar untuk alat, aplikasi, dan pelatihan staf.</li> <li>2. Keamanan Data: Meskipun sistem ini meningkatkan aksesibilitas, ada risiko</li> </ol>

No	Keterangan
	<p>terkait keamanan data dan privasi pasien yang harus dikelola dengan baik.</p> <ol style="list-style-type: none"> <li>3. Ketergantungan pada infrastruktur teknologi yang stabil membuat sistem rentan terhadap gangguan teknis yang dapat mengganggu operasional klinik..</li> <li>4. Adaptasi Pengguna: Staf klinik mungkin memerlukan waktu untuk beradaptasi dengan sistem baru, yang dapat mengganggu alur kerja sementara waktu.</li> <li>5. Supaya tetap aman dan berjalan baik, sistem harus sering diperbarui, namun hal ini juga bisa membuat biaya operasional bertambah.</li> </ol>
8	<p>Nama Jurnal : Sistem Microservices</p> <p>Judul : Sistem Pengelolaan Target Perusahaan Dengan Microservices Architecture Untuk Membantu Peningkatan Kinerja Perusahaan</p> <p>kesimpulan : Peneliti telah merancang aplikasi pengelolaan target perusahaan dengan pendekatan microservices, yang meningkatkan kinerja dan memudahkan pengembangan. Aplikasi ini bisa menjadi referensi bagi perusahaan dalam manajemen target yang efektif. Disarankan untuk terus mengembangkan fitur aplikasi agar lebih sesuai dengan kebutuhan pengelolaan kinerja perusahaan.(Mulyawan et al., 2022).</p> <p>kelebihan</p> <ol style="list-style-type: none"> <li>1. Microservices membagi layanan menjadi aplikasi kecil yang berdiri sendiri.</li> <li>2. Pengembangan dan perubahan pada setiap layanan dapat dilakukan secara independen tanpa mengganggu layanan lain.</li> </ol>



No	Keterangan
	3. Memberikan fleksibilitas dan skalabilitas yang tinggi bagi perusahaan. 4. Menyesuaikan dan meningkatkan kinerja aplikasi sesuai dengan kebutuhan bisnis yang berubah-ubah. 5. Mempercepat proses pengembangan aplikasi secara keseluruhan.
kekurangan	1. Kompleksitas dalam manajemen dan monitoring infrastruktur yang terdiri dari banyak layanan independen. 2. Memerlukan upaya tambahan dalam pengelolaan konfigurasi. 3. Koordinasi antar layanan menjadi lebih rumit. 4. Memerlukan pemantauan kinerja dan debugging lintas layanan. 5. Penggunaan multiple bahasa pemrograman dapat menambah kompleksitas dalam pengembangan, debugging, dan maintenance aplikasi.

Beberapa penelitian yang terkait dengan penelitian ini, memiliki kemiripan yang terletak pada konsep penelitian mengenai meningkatkan pelayanan dengan membangun sistem informasi. Pada penelitian saat ini tentang responsive sistem informasi pelayanan pelayanan medis itu menggunakan bahasa pemrograman golang Dan pada sistem menggunakan Vue.JS untuk antarmuka pengguna. Selain itu penggunaan microservice dalam membangun sistem informasi pelayanan klinik gigi pada penelitian ini.

## 2.2 Teori Terkait

Teori memegang peranan penting dalam penelitian ilmiah. Teori yang digunakan harus relevan dan berkaitan langsung dengan

topik penelitian. Oleh karena itu, studi ini dibangun atas dasar teori-teori yang relevan dan mendukung pelaksanaan penelitian.

### **2.2.1 Sistem Informasi**

Sebuah proses yang bertujuan untuk mengolah data menjadi informasi melalui proses input, pengolahan, output, penyimpanan, dan pengendalian merupakan definisi dari Sistem informasi. Sistem ini terdiri dari berbagai elemen seperti hardware, software, manusia, data, serta prosedur kerja, yang bekerja secara terpadu untuk menghasilkan informasi yang bermanfaat (Marimin et al., 2006).

### **2.2.2 Microservice**

Microservices merupakan metode pengembangan aplikasi dengan memecah proses pembuatan sistem menjadi bagian-bagian kecil yang lebih terfokus. Setiap bagian atau layanan dalam arsitektur microservices memiliki tugas dan fungsi spesifik, serta dijalankan secara independen dengan CPU, lingkungan kerja (environment), dan tim pengembangnya masing-masing. Karena bersifat terpisah, tiap layanan memiliki karakteristik unik dan beroperasi secara mandiri, namun tetap bisa saling berinteraksi melalui penggunaan API. (Hashina & Raisah, 2023).

### **2.2.3 Golang**

Golang merupakan bahasa pemrograman sumber terbuka yang dikembangkan oleh Google pada tahun 2009, dengan perancang utamanya yaitu Ken Thompson, Rob Pike, dan Robert Griesemer. Golang dibuat agar proses pengembangan aplikasi lebih cepat dan memungkinkan pengembang membuat program aman, mudah dikelola serta efisien. Golang juga dirancang khusus untuk pengembangan aplikasi web dan juga backend. Meskipun bahasa pemrograman ini dipakai untuk perangkat lunak di sejumlah platform. Golang juga mempunyai sintaks sederhana serta mudah dipahami. Bahasa pemrograman ini juga mempunyai beberapa fitur, seperti konkurensi, garbage collection dan penggunaan memori secara efisien. (Team, 2023).

#### **2.2.4 VueJS**

Vue.js adalah framework JavaScript yang kini banyak diminati dan berkembang dengan cepat dalam komunitas pengembang. Framework ini dikembangkan oleh Evan You, seorang programmer asal Eropa. Vue.js dikenal dengan pendekatannya yang sederhana dan mudah dipahami, sehingga memudahkan pengembang dalam membangun aplikasi web yang interaktif dan responsif. (Levi, 2023).

#### **2.2.5 Database**

Database memiliki definisi himpunan data yang tersusun terstruktur dan disimpan di dalam komputer, sehingga memudahkan proses pengelolaan dan pengaksesannya melalui perangkat lunak yang dikenal sebagai Sistem Manajemen Basis Data (DBMS). Informasi yang tersimpan bisa berupa teks, angka, simbol, gambar, atau gabungan dari berbagai bentuk representasi objek dunia nyata, seperti individu, produk, peristiwa, dan konsep lainnya (Fahzirah, 2024). Database dirancang untuk menyimpan informasi dalam jumlah besar secara efisien dan aman, serta memungkinkan pengguna melakukan pengambilan dan pengolahan data secara cepat dan akurat. Model yang umum digunakan dalam pengelolaan database adalah model relasional, di mana data direpresentasikan dalam bentuk tabel yang saling terhubung melalui relasi tertentu. Selain itu, database mendukung pengendalian akses data, keamanan informasi, serta integrasi data yang memungkinkan proses bisnis berjalan lebih efisien dan terstruktur (Fahzirah, 2024).


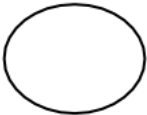

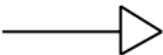
#### **2.2.7 API**



API (Application Programming Interface) merupakan sebuah antarmuka dengan fungsi sebagai jembatan komunikasi antara dua aplikasi, memungkinkan pertukaran data secara efisien dan menjadi komponen krusial dalam pembangunan sistem modern. Salah satu implementasi yang umum digunakan adalah RESTful API, yaitu API yang menggunakan protokol HTTP dengan prinsip Representational State Transfer (REST) yang sederhana, fleksibel, dan mudah diintegrasikan lintas platform. Dalam konteks arsitektur microservices, RESTful API berperan sebagai penghubung antar layanan independen sehingga

masing-masing layanan dapat beroperasi secara terpisah namun tetap saling terintegrasi. Untuk mengatur lalu lintas data dari client ke layanan-layanan tersebut, digunakan API Gateway yang berfungsi sebagai titik masuk utama yang menangani routing, autentikasi, dan monitoring, sehingga interaksi sistem menjadi lebih terstruktur dan efisien (Senduk et al., 2023).

#### 2.2.8 Use Case Diagram



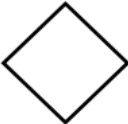
Use Case Diagram adalah diagram yang memvisualisasikan dan merepresentasikan aktor, use case, serta dependensi dalam suatu proyek, dengan tujuan menjelaskan konsep hubungan antara sistem dan lingkungan luar (Mulyani, 2016).



No.	Simbol	Nama	Keterangan
1		<i>Actor</i>	Entitas, proses, atau sistem eksternal yang berhubungan dengan sistem informasi yang dikembangkan, tetapi tidak termasuk di dalam sistem informasi itu sendiri.
2		<i>Use Case</i>	Kemampuan sistem yang diwujudkan dalam bentuk unit-unit yang saling berkomunikasi, baik antarunit maupun dengan aktor.
3		<i>Association</i>	Komunikasi atau aliran informasi antara aktor dan use case yang terlibat, di mana aktor turut berperan serta atau melakukan interaksi dengan use case tersebut.
4		<i>Generalization</i>	Yang mana salah satu fungsi merupakan bentuk

			yang lebih umum dibandingkan fungsi lainnya..
5		<code>&lt;&lt;Include&gt;&gt;</code>	Hubungan antar use case yang menggambarkan bahwa sebuah proses akan berlanjut atau diarahkan ke proses lain sebagai tujuannya.
6		<code>&lt;&lt;Extend&gt;&gt;</code>	Relasi use case yang ditambahkan tetap memiliki kemandirian dan dapat berjalan tanpa harus bergantung pada use case tambahan.

### 2.2.9 Activity Diagram

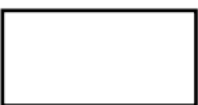
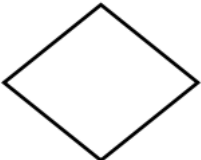
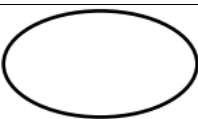

Activity Diagram adalah diagram alur yang menampilkan rangkaian aktivitas pengguna maupun sistem, pihak yang terlibat dalam setiap aktivitas, serta urutan aliran dari aktivitas tersebut. Diagram ini berfungsi untuk memodelkan proses bisnis dan runtutan kegiatan dalam suatu proses (Suendri, 2018).

No.	Simbol	Nama	Keterangan
1		<i>Start</i>	Simbol start digunakan untuk menunjukkan titik awal dari suatu proses.
2		<i>End</i>	Simbol end dipakai untuk menandakan titik akhir dari suatu proses.
3		<i>Decision</i>	Simbol decision digunakan untuk menggambarkan suatu kondisi atau percabangan dalam proses.

4		<i>Activity</i>	Simbol activity merepresentasikan tindakan atau pekerjaan yang dilakukan oleh sistem, dan umumnya diawali dengan kata kerja.
5		<i>State Transition</i>	Simbol state transition berfungsi untuk menghubungkan satu simbol dengan simbol yang lain dalam sebuah diagram.

### 2.2.10 ERD (Entity Relationship Diagram)

ERD merupakan alat bantu berupa diagram dalam model basis data relasional yang digunakan untuk memvisualisasikan hubungan atau keterkaitan antar tabel dalam sebuah database (Raharjo, 2014).

No.	Simbol	Nama	Keterangan
1		<i>Entity</i>	Sekelompok objek yang dapat dikenali atau dibedakan secara unik.
2		<i>Relation</i>	Relasi yang terbentuk antara satu entitas dengan satu atau lebih entitas lainnya, dengan jenis hubungan seperti one to one, one to many, maupun many to many.
3		<i>Attribute</i>	Atribut dari entitas atau relasi yang berfungsi sebagai penjelasan rinci mengenai entitas tersebut.
4		<i>Connector</i>	Keterkaitan antara entitas dengan atribut

			yang dimilikinya, serta antara himpunan entitas dengan himpunan relasi.
--	--	--	---