ISSN: 2614-6754 (print)
ISSN: 2614-3097(online)

Halaman 8667-8676
Volume 7 Nomor 2 Tahun 2023

Clean Arsitektur

Zaky Muhammad Yusuf*[1], Rolly Maulana Awangga[2]

[1,2] Prodi Teknik Informatika, Universitas Logistikdan Bisnis Internasional.

e-mail: info@ulbi.ac.id[1], zakymuhammadyusuf@gmail.com[2]

Abstrak

Clean Architecture adalah sebuah konsep arsitektur perangkat lunak yang berfokus pada pemisahan antara lapisan bisnis dan teknologi dalam aplikasi. Dalam pengembangan aplikasi restoran menggunakan Golang, penggunaan Clean Architecture sangat dianjurkan karena dapat membantu pengembang dalam membuat aplikasi yang mudah di-maintain dan scalable. Dalam arsitektur ini, aplikasi dibagi menjadi beberapa lapisan, antara lain lapisan presentasi, lapisan bisnis, dan lapisan penyimpanan data, masing-masing dengan tugas dan tanggung jawab yang terpisah. Dengan penggunaan Golang dan Clean Architecture, pengembangan aplikasi restoran dapat dilakukan dengan lebih efektif dan efisien.

Kata kunci; Clean Arsitektur, Golang

Abstract

Clean Architecture is a softwarearchitectureconcept that focuses on separating the business and technology layers in applications. In developingrestaurant applicationsusing Golang, the use of Clean Architecture is highly recommended because it can help developers create applications that are easy to maintain and scalable. In this architecture, applications are divided into multiple layers, including a presentation layer, business layer, and data storage layer, each with separate tasks and responsibilities. By using Golang and Clean Architecture, restaurant application development can be done more effectively and efficiently.

Keywords; Clean Architecture, Golang

PENDAHULUAN

Penggunaan teknologi dalam industri restoran semakin berkembang pesat, khususnya dalam hal pengelolaan data dan pengembangan aplikasi. Golang merupakan bahasa pemrograman yang relatif baru namun memiliki performa yang sangat baik dan mudah di maintain. Sementara itu, MongoDB adalah salah satu database NoSQL yang terkenal karena kecepatan dan kemampuannya dalam menangani data yang kompleks. Kombinasi Golang dan MongoDB dapat memberikan hasil yang optimal dalam pengembangan aplikasi restoran.alam pengembangan aplikasi restoran, terdapat berbagai tantangan yang perlu diatasi, seperti skala besar, kompleksitas bisnis, dan kecepatan respon yang tinggi. Oleh karena itu, dibutuhkan sebuah arsitektur perangkat lunak yang dapat membantu pengembangan aplikasi restoran menjadi lebih mudah di-maintain, scalable, dan modular. Salah satu arsitektur perangkat lunak yang dapat digunakan adalah Clean Architecture [1].

Clean Architecture merupakan sebuah konsep arsitektur perangkat lunak yang fokus pada pemisahan antara lapisan bisnis dan teknologi dalam sebuah aplikasi. Konsep ini telah digunakan secara luas oleh para pengembang perangkat lunak untuk membangun aplikasi yang efektif, scalable, dan mudah di-maintain. Penggunaan Clean Architecture pada pengembangan aplikasi restoran menggunakan Golang dan MongoDB akan membantu pengembang dalam membuat aplikasi yang efektif, efisien, dan mudah di- maintain. Oleh karena itu, penelitian literatur ini bertujuan untuk meninjau penggunaan Golang, MongoDB, dan Clean Architecture dalam pengembangan aplikasi restoran dan menjelaskan manfaat serta keuntungan dari penggunaan ketiganya [1].

Kluster kawanan terdiri dari setidaknya satu manager node, Docker Engine, yang menangani topologi cluster dan mempertahankan status cluster secara keseluruhan. Di dalam kasus beberapa node manajer, mereka mulai dalam status pengikut dan gunakan Algoritma Konsensus Raf untuk menegosiasikan status klaster global dan pilih satu node pemimpin untuk klaster [2].

Perangkat IoT perlu memiliki penyimpanan data yang berbasis cloud untuk menjaga keamanan data dengan sistem backup yang handal dan performa throughput yang baik. Penelitian berjudul "IFCIoT: Integrated Fog Cloud IoT Architectural Paradigm for Future Internet of Things" mencatat bahwa integrasi perangkat IoT dengan cloud dapat meningkatkan performa dan skalabilitas perangkat IoT [3]. Sementara itu, penelitian lain berjudul "A Cloud-IoT Based Sensing Service for Health Monitoring" menekankan bahwa integrasi perangkat IoT dengan cloud dapat meningkatkan aksesibilitas dan penggunaan data dari sensor perangkat IoT [4].

Dalam konteks cloud computing, penyimpanan data dapat diakses melalui internet, baik untuk membaca data (read) maupun menulis data (write). Hal ini membuat cloud computing menjadi solusi penyimpanan yang cocok untuk perangkat IoT karena aksesnya hanya melalui satu jalur yaitu internet. Penelitian tersebut juga mengimplementasikan penggunaan basis data relasional pada cloud computing untuk menyimpan data secara terstruktur, yang memungkinkan pengelolaan dan analisis data yang lebih baik [5].

Basis data relasional merupakan jenis basis data di mana data disimpan dalam bentuk tabel yang terdiri dari baris dan kolom. Basis data relasional memiliki struktur tabel yang tetap (schema) dan menggunakan indeks dan kunci (key) untuk mempercepat operasi manipulasi data melalui query. Penelitian yang berjudul "Perbandingan Performa Relational, Document-Oriented dan Graph Database Pada Struktur Data Directed Acyclic Graph" telah menguji performa dari basis data relasional, document-oriented, dan graph database pada struktur data Directed Acyclic Graph [10].

Pada penelitian ini, Restful API juga diterapkan pada database berbasis cloud untuk menjaga keamanan komunikasi data dan mencegah akses langsung yang berpotensi disusupi oleh pihak tidak berwenang. Seluruh perangkat IoT berkomunikasi melalui Restful API, dan data kemudian disimpan ke dalam database. Keamanan database menjadi prioritas utama dalam penelitian ini karena database berbasis cloud dapat diakses oleh siapa saja melalui internet, seperti yang ditemukan dalam penelitian lainnya “Penerapan Keamanan Penggunaan Data pada Database Kepegawaian Menggunakan Teknik Transparent Data Encryption (Studi Kasus Sekolah Tinggi Teknologi Payakumbuh)” dan penelitian berjudul “Pengamanan Data pada Media Penyimpanan Cloud

Menggunakan Teknik Enkripsi dan Secret Sharing” [11], [12] Peningkatan penggunaan Big Data di Indonesia telah meluas dan mencakup berbagai aspek kehidupan. Dalam konteks ini, kebutuhan akan model data yang fleksibel menjadi penting. Model data relasional dengan struktur tabelnya memiliki keterbatasan dalam menangani volume dan variasi data yang besar. Oleh karena itu, diperlukan pendekatan baru dalam basis data dan model data, yang mengarah pada konsep "NoSQL". Dalam model data NoSQL, struktur basis datanya tidak terdiri dari tabel dengan baris dan kolom seperti pada model relasional. Sebaliknya, basis data NoSQL dapat berbentuk dokumen, grafik, atau berdasarkan nilai kunci [6].

Karena terdapat fitur-fitur yang berbeda dengan basis data SQL, maka diperlukan pemahaman lebih lanjut. Paper ini dibuat dengan tujuan pembaca bisa paham mengenai fitur-fitur yang dimiliki MongoDB, Kelebihan dan kekurangan MongoDB, Aplikasi MongoDB, serta contoh penggunaan query pada MongoDB [6].

Saat ini, beberapa alat ETL (Extract Transform Load) seperti Informatica, Pentaho, dan Talend telah mengembangkan fitur untuk migrasi dari basis data relasional ke MongoDB. Namun, fitur-fitur ini hanya fokus pada proses perpindahan data. Data dari basis data relasional diekstraksi dan diubah sesuai dengan format yang diperlukan agar dapat disimpan dalam bentuk dokumen di MongoDB. Meskipun data telah diekstraksi untuk persiapan migrasi, kelemahan fitur-fitur ini adalah perlunya pembuatan model dokumen pada MongoDB secara manual oleh pengembang atau insinyur basis data sebelum data dapat dipindahkan. Untuk

mengatasi masalah ini, dibutuhkan sistem transformasi skema basis data relasional menjadi model data berbasis dokumen pada MongoDB. Sistem ini diharapkan dapat mengakomodasi proses migrasi basis data dan memberikan solusi untuk pembentukan model dokumen pada MongoDB yang sesuai dengan skema basis data relasional yang digunakan [7].

Dalam penelitian ini, kami menciptakan prototipe platform inovasi untuk perusahaan kasus yang dapat digunakan oleh pengembang perusahaan untuk membangun, menerapkan, dan menjalankan aplikasi mereka sendiri. Saat ini, perusahaan kasus menggunakan layanan Amazon Web Services (AWS) dalam konteks mereka untuk menjalankan aplikasi, yang membutuhkan keterampilan dan sumber daya yang signifikan serta biaya lisensi. Tujuan di balik pembuatan prototipe platform inovasi untuk Axis adalah untuk dapat membuat prototipe cepat dan segera mendapatkan umpan balik pelanggan serta mengurangi waktu dan biaya pengembangan[8].

Artikel ini terstruktur sebagai berikut. Bagian II menjelaskan karya terkait dan Bagian III menguraikan metodologi penelitian. Bagian IV menggambarkan implementasi dan Bagian V menyajikan evaluasi prototipe dari kelompok fokus dan wawancara. Terakhir, Bagian VI dan VII menyajikan diskusi diikuti oleh kesimpulan terkait pertanyaan penelitian secara berturut-turut [7].

Ada banyak layanan yang dapat disediakan melalui internet dengan bantuan komputasi awan. Tiga jenis utama dari komputasi awan adalah Infrastructure as a Service (IaaS), Platform as a Service (PaaS), dan Software as a Service (SaaS). Perbedaan antara layanan-layanan ini dijelaskan oleh Goyal [19]. dalam hal apa yang telah dilaksanakan oleh penyedia dan apa yang masih perlu dilaksanakan oleh pelanggan (lihat Tabel 1). IaaS menawarkan kontrol penuh kepada klien dengan menyediakan infrastruktur yang diperlukan. SaaS mengambil konsep ini lebih jauh dengan mendeploy aplikasi yang sudah dikembangkan di dalam infrastruktur awan. Ini memungkinkan klien untuk mengakses aplikasi melalui antarmuka web. PaaS berada di tengah-tengah, menyediakan pengembang dengan tumpukan pengembangan yang siap pakai untuk mengatasi keamanan, pencadangan, dan pemulihan sehingga pengembang dapat fokus pada implementasi aplikasi. PaaS memungkinkan peningkatan kecepatan penyebaran aplikasi [20]. Siklus pengiriman berkelanjutan dari aplikasi perangkat lunak dalam sistem awan dapat ditingkatkan dengan menggunakan fungsionalitas yang ada dalam platform PaaS. Siklus ini tidak perlu ditetapkan, melainkan pengiriman otomatis ke produksi dimungkinkan saat melaksanakan pengiriman berkelanjutan. Pengiriman berkelanjutan merujuk pada memastikan rilis dan penerapan perangkat lunak kapan saja. Ini memungkinkan waktu pemasaran yang lebih cepat, biaya pengembangan yang lebih rendah, lingkungan pengembangan yang lebih fleksibel bagi para pengembang, dan pembaruan perangkat lunak yang lebih cepat bagi klien yang menciptakan nilai bagi perusahaan [9], [10].

Di bagian backend, terdapat berbagai bahasa pemrograman yang dapat digunakan. Pada proyek ini, kami menggunakan bahasa pemrograman Golang. Golang adalah bahasa pemrograman yang dikembangkan oleh Google bersama dengan Ken Thompson, Robert Griesemer, dan Rob Pike pada tahun 2009. Tujuan utama pengembangan bahasa ini adalah untuk menciptakan bahasa yang memiliki keunggulan dalam hal kecepatan, keandalan, skalabilitas, dan kesederhanaan. Golang juga merupakan bahasa pemrograman yang statis-typed dan menghasilkan kode biner yang dapat dikompilasi. Selain itu, Golang juga merupakan hasil pengembangan dari bahasa pemrograman C untuk era abad ke-21. Bahasa Go juga dapat digunakan untuk pembuatan aplikasi, website, dan perangkat lunak lainnya [11].

B Backend merupakan "tulang punggung" sebuah website. Backend engineer bertanggung jawab dalam menyiapkan data yang akan ditampilkan pada website. Backend juga memiliki peran penting dalam menjaga fungsionalitas website tersebut. Meskipun peranannya tidak diketahui oleh banyak orang dan tidak berhubungan langsung dengan pengguna, kelancaran sebuah website sangat bergantung pada bagaimana backend membangun fondasi data di belakangnya [11].

REST (Representational State Transfer) is an architectural communication method that

utilizes the HTTP protocol for data exchange, aiming to create a system that performs well, is fast, and easy to develop and scale, especially in terms of data exchange and communication. REST supports several HTTP methods, including GET, POST, PUT, and DELETE [13].

A web service system that implements the principles of REST is called RESTful. The working principle of RESTful is that the client sends a request through an HTTP Request, and the server responds to the client's request through an HTTP Response. There are two parts to the message for communicating with the server: the header and the body. The HTTP header contains small notes for each transaction in HTTP, while the HTTP body contains the data to be sent [14].

Web service adalah teknologi yang memungkinkan komunikasi saling antar internet dengan pola program-to-program. Tujuan penggunaan web service adalah agar klien dari berbagai platform seperti dekstop, website, dan aplikasi mobile dapat mengakses layanan yang disediakan oleh web service[12]. REST adalah suatu arsitektur layanan web yang mengadopsi model client-server. Dalam model ini, klien mengirimkan permintaan kepada server, kemudian server akan memproses permintaan tersebut dan menghasilkan respons yang dikembalikan kepada klien. Restful web service mengacu pada aplikasi web yang menggunakan arsitektur REST. Dalam menggunakan teknologi REST, digunakan sebuah alat bantu bernama API (Application Programming Interface) berbasis website. API secara umum terdiri dari dua bagian, yaitu server sebagai penyedia data dan klien yang dapat melakukan permintaan data. REST API adalah API berbasis website yang menggunakan teknologi REST dan menggunakan format JSON (JavaScript Object Notation), yang merupakan format pertukaran data yang dapat digunakan baik pada front-end maupun back-end dari aplikasi website atau layanan[13].

Dengan teknologi web service REST API ini, integrasi pemesanan makanan dan pegawai dari aplikasi Restoran ini dapat memenuhi kebutuhan dalam menjalankan tugasnya agar menjadi lebih akurat dan efisien[14].

NoSQL juga dirancang untuk menyimpan data yang didistribusikanuntuk kebutuhan data dalam skala besar; misalnya Facebook memiliki 500 juta pengguna dan Twitter terakumulasi terabyte data, database NoSQL telah memiliki popularitas yang tinggi [6].

METODE

Dalam upaya untuk mengoptimalkan sistem restoran, kami mengadopsi teknologi REST API untuk mengintegrasikan aplikasi yang sudah ada, seperti mesin presensi dan aplikasi TOYA DEVASYA. Kami menerapkan tahapan pengembangan perangkat lunak dengan metode waterfall [9], yang cocok untuk produk perangkat lunak/program yang memiliki kebutuhan yang jelas di awal sehingga risiko kesalahan dapat diminimalkan [10]. Metode waterfall ini melibatkan tahapan-tahapan berikut [11].

1. Analisis dan definisi kebutuhan
   Kebutuhan sistem, kendala, dan tujuan ditetapkan melalui konsultasi dengan pengguna, kemudian secara rinci didefinisikan dan berfungsi sebagai spesifikasi sistem.
2. Perancangan sistem dan perangkat lunak
   Tahapan perancangan sistem mengalokasikan kebutuhan sistem, baik perangkat keras maupun perangkat lunak, dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi dasar sistem perangkat lunak dan relasinya.
3. Implementasi
   Pada tahap ini, perancangan perangkat lunak diwujudkan sebagai serangkaian program atau unit program. Pengujian dilakukan untuk memverifikasi bahwa setiap unit memenuhi spesifikasinya.
4. Integrasi dan pengujian sistem
   Program-program digabungkan dan diuji sebagai satu sistem lengkap untuk memastikan kesesuaian dengan kebutuhan perangkat lunak.
5. Operasi dan pemeliharaan
   Pada tahapan ini, dilakukan pemasangan dan penggunaan sistem secara aktif [15]. Tahap

pemeliharaan melibatkan identifikasi dan perbaikan kesalahan yang tidak terdeteksi pada tahapan sebelumnya, peningkatan implementasi unit sistem, serta penyesuaian layanan sistem sesuai dengan kebutuhan yang baru muncul

Reaserch Question

Pada tahap ini ditentukan pertanyaan yang sesuai dengan topik penelitian. Berikut ini merupakan research question pada penelitian ini :

1. RQ1 : bagaimana aplikasi atau sistempemesanan makanan berbasis online ini?
2. RQ2 : Apa metode yang digunakan untukpengambilan data tentang Restoran berbasis online?
3. RQ3 : Apa faktor yang mempengaruhikepuasan pelanggan?

Search Process

Proses pencarian informasi adalah langkah penting dalam mendapatkan sumber yang relevan dengan pertanyaan penelitian. Untuk melakukan pencarian, dapat digunakan platform seperti Google Scholar dengan mengunjungi situs web resminya di https://scholar.google.co.id/.

Documentation

Pada tahap ini, hasil penelitian akan disusun dalam bentuk paper sesuai dengan format yang telah ditentukan.

HASIL DAN PEMBAHASAN

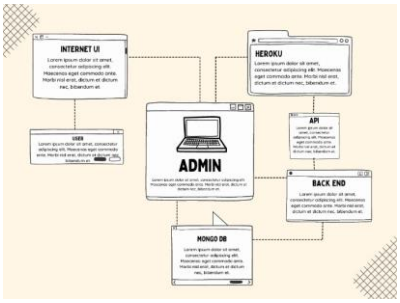
Dalam penelitian ini, kami menggunakan pendekatan pengembangan sistem waterfall untuk mengembangkan aplikasi web service. Berikut ini adalah langkah-langkah yang kami lakukan selama proses pengembangan:

Analisis Kebutuhan

Dalam tahap analisis kebutuhan untuk pengembangan aplikasi restoran, langkah awal yang penting dilakukan adalah mengumpulkan data dan menganalisis permasalahan yang ada. Data dapat dikumpulkan melalui metode observasi proses dan wawancara dengan pihak terkait, termasuk pemilik restoran, karyawan, dan pelanggan. Observasi proses dapat memberikan pemahaman yang lebih baik tentang bagaimana operasional restoran dilakukan saat ini, seperti langkah-langkah pemesanan, pengolahan pesanan, pengiriman makanan, dan pembayaran. Melalui wawancara dengan pihak terkait, dapat diperoleh wawasan yang lebih mendalam mengenai masalah yang dihadapi, kebutuhan yang perlu dipenuhi, serta harapan-harapan terkait dengan aplikasi yang akan dikembangkan.

Desain Sistem

Dalam merancang sistem aplikasi restoranyang akan diterapkan menggunakan platform Heroku, Proses desain akan menghasilkan deskripsi terperinci tentang urutan langkah-langkah dan data yang didasarkan pada hasil analisis. Dengan memahami alur proses yang telah ditetapkan, dapat ditentukan fitur-fitur yang dapat dimanfaatkan oleh pengguna yang terlibat dalam operasional restoran. Sebagai contoh, pada saat ini proses pengiriman pesanan makanan masih menggunakan catatan manual yang dilakukan oleh pelayan restoran. Namun, melalui optimasi sistem dan penerapan Heroku, pelayan restoran dapat dengan mudah memasukkan pesanan melalui aplikasi yang telah dideploy di platform tersebut. Selanjutnya, data pesanan yang diinput akan diverifikasi oleh administrator melalui aplikasi admin yang terhubung dengan sistem yang berjalan di Heroku.



Gambar 1 Rancangan Alur Sistem

Berdasarkan hasil analisis yang telah dilakukan, dirancang sebuah proses bisnis yang diusulkan. Dalam optimasi sistem tunjangan kinerja pegawai, terdapat data dan fitur yang diperlukan, antara lain:

a. Penambahan fitur pada Server Aplikasi Restoran
b. Dalam hal ini, fitur tersebut akan berperan sebagai penerima data masukan dari sistem yang akan dibuat melalui layanan web [15]. Server akan menyediakan data daftar menu dan data presensi yang akan ditampilkan pada aplikasi restoran.
c. Pemberian token kepada operator untuk meningkatkan keamanan data Dalam upaya meningkatkan keamanan data, operator akan diberikan sebuah token. Token ini akan digunakan untuk otentikasi saat mengakses sistem, sehingga hanya pihak yang memiliki token yang dapat mengakses dan mengelola data dengan aman.
d. Sistem akan menggunakan mekanisme login yang sama antara klien (pengguna aplikasi restoran) dan admin restoran untuk mengakses aplikasi website. Hal ini bertujuan untuk memudahkan pengguna dalam mengakses dan mengelola data melalui aplikasi website dengan menggunakan kredensial yang sama seperti saat login ke aplikasi restoran.
e. Aplikasi restoran, dapat diberi aksesuntuk mengubah data pegawai dan user.
f. Data user costumer dimasukan oleh operator Admin melalui hasil pemesanan costumer.

Area Penelitian	Tahun	Metode	Hasil Penelitian
ACloud-IoT Based Sensing Service for Health Monitoring (Gabriel Neagu, Stefan Preda, Alexandru Stancium Vladimir Florian)	2017	Cloud-IoT	Dalam penelitian ini, pendekatan Cloud-IoT digunakan untuk menggabungkan keuntungan dari teknologi komputasi awan dan IoT. Penerapan ini memberikan keunggulan dalam aplikasi IoT tradisional seperti Health Monitoring (HM), termasuk penyimpanan dan pemrosesan data yang dapat diskalakan, aksesibilitas dan berbagi data yang mudah, tata kelola data yang canggih, pemulihan bencana, dan efisiensi biaya. Penelitian ini berfokus pada pengembangan layanan sensing berbasis awan yang bertujuan untuk meningkatkan efisiensi penggunaan data sensor. Sebuah skenario implementasi khusus HM telah diusulkan dengan fokus pada peran SSP untuk meningkatkan kualitas layanan, memperluas aksesibilitas bagi fasilitas medis kecil dan menengah, dan mengurangi upaya staf medis dalam pengaturan dan operasi layanan. Identifikasi yang jelas terhadap pemangku kepentingan dan tanggung jawab mereka menciptakan peluang implementasi dan operasi layanan yang berkelanjutan. Saat ini, versi eksperimental dari skenario HM-SS sedang dalam pengembangan pada platform TI yang tersedia, termasuk fasilitas awan yang kuat dan solusi pengumpulan data IoT pilot.
SensIoT: An Extensible and General	2019	SensIoT	Penelitian ini memperkenalkan SensIoT, sebuah kerangka pemantauan sensor umum untuk Internet of Things (IoT). SensIoT

Internet of Things Monitoring Framework (Marcel Grobamnn, Steffen Illig, and Cornelius L. Matejka)			menggunakan teknologi Docker dan Docker Swarm untuk mengatasi masalah perangkat lunak dan memungkinkan pengumpulan data yang fleksibel. Melalui penggunaan perangkat IoT, SensIoT efisien dalam mengumpulkan data lingkungan dan memiliki solusi yang tepat untuk penyimpanan data dan visualisasi yang bermakna. Penelitian ini telah membuktikan fungsionalitas dan keandalan SensIoT melalui pengujian di skenario kehidupan nyata. SensIoT memiliki potensi yang tinggi untuk menjadi kerangka pemantauan sensor umum yang lebih baik untuk IoT.
Web-Based Online Learning System MVC Method (Salmiati, Mardi Turnip)	2021	MVC	Penelitian ini menghasilkan Sistem Pembelajaran Berbasis Online (SPBO) yang menggunakan pendekatan Laravel PHP Framework sebagai Backend dan Blade Templating sebagai pemrosesan tampilan untuk Frontend. SPBO dirancang untuk memfasilitasi kegiatan belajar mengajar secara online dengan fitur seperti konferensi video, kehadiran otomatis, pengumpulan tugas, penyediaan materi, jadwal, dan penilaian tugas. Melalui pelatihan dan pengenalan SPBO kepada peserta, penelitian ini berhasil meningkatkan kesadaran dan penerapan SPBO dalam mendukung pembelajaran. Namun, penelitian ini juga menghadapi kendala dalam pengetahuan peserta mengenai aplikasi komputer yang digunakan.
Implementasi Golang dan New Simple Queue pada Sistem Sandbox Pihak Ketiga Berbasis REST API (Albertus Ari Kristanto, Yulius Harjoseputro, Joseph Eric Samodra)	2021	Clean Architecture	
Penerapan Clean Architecture pada Pengembangan Sistem Payment Point	2022	Clean Architecture	Penelitian ini menghasilkan aplikasi Sistem Payment Point Online Bank (PPOB) yang terdiri dari Web Service, tampilan admin menggunakan VueJS, dan tampilan pengguna menggunakan aplikasi mobile Android dengan Flutter. Web Service menggunakan bahasa Go dengan framework

Bank (Arif Widisan Subagio, Faisal Muttaqin)			Echo untuk REST API. Sistem PPOB juga terintegrasi dengan Xendit sebagai metode pembayaran pihak ketiga. Penelitian ini juga menerapkan Clean Architecture dalam pengembangan sistem, dengan setiap komponen ditempatkan pada folder yang berbeda
Penilaian Kematangan Proses Keamanan Sistem Informasi Pendaftaran Pasien Menggunakan Framework Cobit 4.1 (Setiyowati, Sri Siswanti)	2021	Clean Architecture	Penelitian ini menunjukkan bahwa Rumah Sakit XYZ memiliki alur yang belum standar dalam mengelola proses yang ada. Kurangnya Standar Operasional Prosedur dan komunikasi yang tidak memadai menyebabkan penyimpangan. Untuk meningkatkan kinerja, penelitian merekomendasikan pembuatan standar dan prosedur yang akan memudahkan staf dalam menjalankan tugas mereka dan meminimalkan penyimpangan. Selain itu, dilakukan sosialisasi kepada staf mengenai standar dan prosedur, serta pelatihan untuk meningkatkan keahlian staf dalam penggunaan alat dan otomatisasi. Rekomendasi lainnya termasuk membuat SOP khusus untuk keamanan data, mengadakan pemantauan dan pengelolaan berkala terhadap keamanan sistem, serta melakukan pemasangan perangkat lunak dan alat bantu keamanan. Secara keseluruhan sistem pendaftaran pasien di Rumah Sakit XYZ telah mencapai tingkat kematangan yang cukup baik.

SIMPULAN

Clean Architecture adalah sebuah konsep arsitektur perangkat lunak yang bertujuan untuk memisahkan lapisan bisnis dan teknologi dalam aplikasi. Konsep ini sangat dianjurkan dalam pengembangan aplikasi restoran menggunakan Golang karena dapat membantu pengembang dalam menciptakan aplikasi yang mudah di-maintain dan scalable. Dengan menggunakan Golang dan Clean Architecture, pengembangan aplikasi restoran dapat dilakukan dengan lebih efektif dan efisien. Metode penelitian yang digunakan dalam optimasi sistem restoran adalah teknologi REST API Untuk untuk mengintegrasikan aplikasi yang sudah ada, seperti mesin presensi dan aplikasi TOYA DEVASYA, dilakukan pengembangan perangkat lunak menggunakan metode waterfall. Metode ini sangat sesuai digunakan untuk produk perangkat lunak/program yang memiliki kebutuhan yang jelas di awal, sehingga dapat mengurangi kemungkinan terjadinya kesalahan. Tahapan metode waterfall meliputi analisis dan definisi kebutuhan, desain sistem dan perangkat lunak, implementasi, integrasi dan pengujian sistem, serta operasi dan pemeliharaan.

Penelitian ini menggunakan metode waterfall dalam pengembangan aplikasi web service untuk mengoptimalkan sistem restoran. Tahapan yang dilakukan meliputi analisis kebutuhan dan desain sistem. Analisis kebutuhan dilakukan melalui observasi proses dan wawancara dengan pemilik restoran, karyawan, dan pelanggan untuk memahami operasional restoran dan mengidentifikasi masalah serta kebutuhan yang perlu dipenuhi. Desain sistem dilakukan dengan merancang sistem aplikasi restoran dan menentukan fitur-fitur berdasarkan

alur proses yang telah ditetapkan. Melalui optimasi sistem dan penerapan platform Heroku, pengiriman pesanan makanan dapat dilakukan dengan lebih efisien melalui aplikasi yang terhubung dengan sistem yang berjalan di Heroku.

Sebagai hasil dari analisis tersebut, disarankan adanya proses bisnis dan fitur-fitur tertentu yang diperlukan untuk mengoptimalkan sistem tunjangan kinerja pegawai. Hal ini mencakup penambahan fitur pada Server Aplikasi Restoran, pemberian token kepada operator untuk meningkatkan keamanan data, penggunaan login yang sama untuk klien dan admin restoran, akses aplikasi restoran yang memungkinkan pengubahan data pegawai dan pengguna, serta penambahan data pengguna pelanggan melalui hasil pemesanan oleh operator admin.

Saran

Dalam pengembangan aplikasi restoran menggunakan Golang dan MongoDB, disarankan untuk menerapkan Clean Architecture sebagai arsitektur perangkat lunak. Clean Architecture akan membantu dalam memisahkan lapisan bisnis dan teknologi, sehingga aplikasi menjadi lebih mudah di-maintain, scalable, dan modular. Penggunaan Golang dan MongoDB akan memberikan performa yang baik dan kemampuan dalam mengelola data yang kompleks. Dalam implementasi, dapat mempertimbangkan integrasi dengan layanan web service RESTful API untuk agar terjamin keamanan komunikasi data dan mencegah akses ilegal ke database berbasis cloud, sangat penting untuk memperhatikan aspek keamanan. Selain itu, juga diperlukan perhatian terhadap kebutuhan penyimpanan data berbasis cloud dengan sistem backup yang handal dan performa throughput yang optimal guna mendukung perangkat IoT. Dalam hal ini, penggunaan cloud computing dan NoSQL database seperti MongoDB dapat menjadi solusi yang tepat.

DAFTAR PUSTAKA

R. Saputra and A. Ashari, “Integrasi laporan demam berdarah dengue (dbd) menggunakan teknologi web service,” J. Masy. Inform., vol. 2, no. 3, pp. 15–26,2011.

A. Nugroho, “Sistem Informasi Tunjangan Kinerja Prajurit Tni Ad Di Brigif Mekanis 1 Pik/Js,” presented at the Semnas Ristek (Seminar Nasional Riset dan Inovasi Teknologi), 2020, vol. 4, no. 1.

N. D. Amalo and P. Katemba, “Sistem Informasi Perhitungan Tunjangan Pegawai dan Gaji Honorer Berbasis Website pada Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Kabupaten Kupang,” High Educ. Organ. Arch. Qual. J. Teknol. Inf., vol. 12, no. 1, pp. 29–37, 2021.

Z. Sitorus, “Kebutuhan Web service untuk Sinkronisasi Data Antar Sistem Informasi dalam Universitas,” J. Tek. Dan Inform., vol. 5, no. 2, pp. 87–90,2018.

S. Wiwit, “Membangun Web service Open Source Menggunakan PHP,” PT Elexmedia Komputindo Jkt., 2004.

G. F. Belete et al., “Exploring low-carbon futures: A web service approach to linking diverse climate-energy- economy models,” Energies,vol. 12, no. 15, p. 2880, 2019.

Z. Zeng, X. Yuan, J. Liang, and Y. Li, “Designing and implementing an SWMM-based web service frameworkto provide decision support for real-time urban stormwater management,” Environ. Model. Softw., vol. 135, p. 104887, 2021.

G. Indrawan, I. Gunadi, and I. Sandhiyasa, “REST API and Real-Time Notification of SIsKA-NGMobile forthe Academic Progress Information System,” in Information and Communication Technology for Competitive Strategies (ICTCS 2020), Springer, 2022, pp. 193–201.

S. P. Roger, “Rekayasa Perangkat Lunak pendekatan praktisi (buku satu),” Yogyak. Andi, 2002.

T. Pricillia, “Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD),” J. Bangkit Indones., vol. 10, no. 1, pp. 6– 12, 2021.

I. Sommerville, “Software Engineering, Addison-Wesley,” Read. MA, 1995.

M. T. Salmiati, “Jurnal Mantik Jurnal Mantik,” Mobile-Based Natl. Univ. Online Libr. Appl. Des.,

vol. 3, no. 2, pp. 10–19, 2019, [Online]. Available:

Setiyowati and Sri Siswanti, “Penilaian Kematangan Proses Keamanan Sistem Informasi Pendaftaran Pasien Menggunakan Framework Cobit 4.1,” SATIN - Sains dan Teknol. Inf., vol. 7,

M. Großmann, S. Illig, and C. L. Matějka, “SensIoT: An extensible and general internet of things monitoring framework,” Wirel. Commun. Mob. Comput., vol. 2019, 2019, doi: 10.1155/2019/4260359.

I. Mulya Pradana and R. Pradana, “Implementasi Advanced Encryption Standard 128 Bit Dan Shamir Secret Sharing Pada Website Data Ulang Pensiun Lembaga Dana Pensiun Pertamina,” Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta- Indonesia, no. September, pp. 121–129, 2022.

A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 6th Ed. New York: McGraw-Hill Companies, Inc, 2011.

A. Manova, “Statistics – Textbook,” 2016. [Online]. Available: http://documents.software.dell.com/Statistics/Textbook/Naive-Bayes-Classifier. [Accessed: 09-Sep-2018].

O. Khalifa, S. Harding, and A. Hashim, “Compression Using Wavelet Transform,” Signal Process. An Int. J., Vol. 2, No. 5, pp. 17–26, 2008.

D. C. S. Kumar, J.Seetha, and S. R. Vinotha, “Security Implications of Distributed Database Management System Models,” Int. J. Soft Comput. Softw. Eng., vol. 2, no. 11, pp. 20–28, 2012.

Date, F. Date, and F. Date, Continued From Part-8, vol. 8, no. 12. 2016.

A. Rahman Fajri and S. Rani, “Penerapan Design Pattern MVVM dan Clean Architecture pada Pengembangan Aplikasi Android (Studi Kasus: Aplikasi Agree Partner),” 2022.

G. Neagu, S. Preda, A. Stanciu, and V. Florian, “A Cloud-IoT based sensing service for health monitoring,” 2017 E-Health Bioeng. Conf. EHB 2017, no. June, pp. 53–56, 2017, doi: 10.1109/EHB.2017.7995359.

A. S. Sari and R. Hidayat, “Designing website vaccine booking system using golang programming language and framework react JS,” J. Inf. Syst. …, vol. 6, no. 1, pp. 22–39, 2022, doi: 10.52362/jisicom.v6i1.760.

A. S. Sari and R. Hidayat, “Designing website vaccine booking system using golang programming language and framework react JS,” J. Inf. Syst. …, vol. 6, no. 1, pp. 22–39, 2022, doi: 10.52362/jisicom.v6i1.760.

M. Kasyful Anwar, “Perancangan Database IoT Berbasis Cloud dengan Restful API Cloud-Based IoT Database Design with Restful API,” vol. 20, no. 2, pp. 268–279, 2021.