



# 7.78%

SIMILARITY OVERALL

SCANNED ON: 8 JUL 2024, 2:16 PM

## Similarity report

Your text is highlighted according to the matched content in the results above.

**IDENTICAL** 0.01%    **CHANGED TEXT** 7.76%    **QUOTES** 0.91%

## Report #21954173

**18 23** BAB I PENDAHULUAN 1.1 Latar Belakang Masalah Perkembangan sistem informasi telah menjadi fondasi utama bagi berbagai industri dan organisasi. Sistem informasi mencakup infrastruktur teknologi yang bertanggung jawab untuk mengumpulkan, menyimpan, mengelola, dan mengolah data guna memberikan informasi yang berharga bagi pengguna. Dalam konteks bisnis modern, sistem informasi memainkan peran sentral dalam memfasilitasi berbagai proses operasional, pengambilan keputusan, dan interaksi antara organisasi dan lingkungan eksternalnya. Dengan sistem informasi yang efektif, perusahaan dapat mengatur informasi mereka dengan lebih efisien, mengoptimalkan proses bisnis, dan meningkatkan kualitas keputusan. Sistem informasi juga memungkinkan perusahaan untuk menanggapi perubahan pasar dengan lebih cepat dan menyesuaikan strategi bisnis mereka secara lebih adaptif. Sebuah dashboard diperlukan untuk berperan sebagai titik fokus yang menggabungkan data dan informasi yang dihasilkan oleh sistem informasi. Dashboard memungkinkan sebuah perusahaan untuk menampilkan informasi yang mereka kumpulkan, simpan, dan kelola melalui sistem informasi mereka. Dengan demikian, dashboard tersebut menjadi alat yang vital dalam memahami dan mengelola informasi untuk kebutuhan operasional dan pengambilan keputusan. Melalui dashboard perusahaan dapat mengoptimalkan proses bisnis mereka dengan mengatur informasi secara lebih efisien. Grafik, tabel, dan metrik kunci yang ditampilkan dalam dashboard



memberikan pandangan yang jelas tentang kinerja perusahaan dan tren yang muncul. Dengan demikian, perusahaan dapat dengan cepat menanggapi perubahan pasar dan menyesuaikan strategi bisnis mereka secara adaptif. Lebih dari sekadar alat pelaporan, dashboard menjadi sarana yang memungkinkan perusahaan untuk meningkatkan kualitas keputusan perusahaan. Dengan visualisasi data yang kuat, perusahaan dapat mengeksplorasi informasi lebih dalam, menganalisis pola-pola yang muncul, dan membuat prediksi berdasarkan tren historis. **54** Dengan adanya dashboard memungkinkan perusahaan untuk mengambil keputusan yang lebih terinformasi dan cerdas. Berbicara tentang dashboard terdapat server yang memainkan peran sentral sebagai pusat kontrol dan penyimpanan data yang mendukung operasional perusahaan seperti dashboard. Sebagai platform utama, server tidak hanya menyediakan layanan, tetapi juga menjadi fondasi bagi aplikasi dan operasi sehari-hari. Oleh karena itu server memainkan peran krusial dalam ekosistem teknologi informasi sebuah perusahaan, terutama dalam konteks pembangunan dan pengoperasian dashboard. Pemilihan dan pengelolaan infrastruktur server yang tepat menjadi kunci dalam memastikan bahwa dashboard dapat beroperasi dengan efisien, handal, dan mampu memenuhi kebutuhan informasi dan analisis perusahaan dengan baik. Paradigma microservice telah muncul sebagai pendekatan inovatif dalam pengembangan perangkat lunak. Microservice mengusung ide memecah aplikasi menjadi serangkaian layanan independen yang beroperasi secara mandiri. Setiap layanan dalam arsitektur microservice dapat dikembangkan, dikelola, dan diimplementasikan secara terpisah, memberikan fleksibilitas dan skalabilitas yang lebih besar dalam lingkungan bisnis yang dinamis. Penerapan microservice memiliki dampak positif pada infrastruktur server dengan karakteristiknya yang memungkinkan peningkatan modularitas dan responsif terhadap perubahan. Beberapa perusahaan, termasuk PT. XYZ, yang merupakan perusahaan berbasis teknologi dengan salah satu produk utamanya berupa aplikasi Enterprise Resource Planning, telah mengadopsi konsep microservice dalam pengembangannya. Fleksibilitas dan skalabilitas

microservice memberikan PT. XYZ kemampuan untuk melakukan pembaruan atau penggantian pada satu layanan tanpa memengaruhi seluruh aplikasi. Dalam konteks ini, adaptasi ukuran infrastruktur sesuai dengan kebutuhan menjadi lebih mudah. Terkait pengelolaan server yang mendukung aplikasi microservice, masalah tersebut menjadi hal yang signifikan dalam konteks pengembangan sistem informasi dan infrastruktur teknologi PT. XYZ. Seiring dengan peningkatan adopsi paradigma microservice, beberapa masalah muncul yang melibatkan kondisi storage, kondisi CPU, dan kondisi RAM dari server yang digunakan oleh aplikasi microservice. Pertama, kondisi storage seringkali kritis karena microservice yang berjalan di server dapat menggunakan storage yang besar akibat aktivitas yang berjalan, terutama ketika berbagai layanan microservice dipasang dan berjalan secara bersamaan didalam satu server yang sama. Karena itu diperlukan pemantauan yang terus-menerus terhadap kapasitas dan kinerja penyimpanan untuk mencegah kehabisan ruang dan menghindari kerusakan data. Kedua, Kondisi CPU dan RAM dari server yang digunakan sangat penting karena setiap layanan microservice memiliki kebutuhan sumber daya yang berbeda-beda. Ketika beban kerja meningkat, server perlu mampu mengalokasikan sumber daya CPU dan RAM dengan efisien untuk menjaga kinerja aplikasi tetap optimal. Seringkali, aplikasi microservice mati karena server tidak mampu menangani traffic yang tinggi, menyebabkan matinya aplikasi secara otomatis. Lonjakan traffic yang tiba-tiba dapat menyebabkan server overload, menyebabkan gangguan dalam layanan dan menurunkan pengalaman pengguna. Dalam kondisi yang semakin kompleks dan berubah-ubah ini, metode monitoring yang dilakukan oleh Staff infrastruktur di PT. XYZ yaitu sekali dalam seminggu dengan cara masuk kedalam server, kemudian memasukan perintah tertentu untuk mendapatkan data resource server dan juga status dari berjalannya aplikasi microservice didalamnya. Dengan metode yang sudah berjalan tersebut Staff infrastruktur tidak dapat melakukan tindakan yang cepat apabila terjadi masalah seperti lonjakan traffic secara tiba-tiba yang menyebabkan server

overload sehingga aplikasi microservice mati. 1 Dalam menghadapi masalah ini, penulis perlu mengembangkan strategi yang efektif untuk memantau kinerja server secara proaktif, mendeteksi masalah potensial, dan mengambil tindakan pencegahan sebelum terjadi masalah yang serius. Selain itu, skala infrastruktur server harus mampu beradaptasi dengan perubahan dalam beban kerja dan permintaan layanan secara dinamis, sehingga memastikan ketersediaan dan keandalan aplikasi microservice. Terkait dengan masalah tersebut, penulis menyadari bahwa PT. XYZ membutuhkan solusi yang canggih dan terintegrasi. Salah satu solusi yang diperlukan adalah pengimplementasian aplikasi monitoring resource server dan aplikasi microservice yang efektif. Aplikasi monitoring ini akan memungkinkan PT. XYZ untuk secara aktif memantau kondisi dan kinerja server, termasuk pemantauan terhadap kondisi storage, CPU dan RAM dari server tersebut, serta diperlukan sebuah notifikasi apabila terdapat resource dari server yang terindikasi penggunaannya melebihi batas yang telah ditentukan. Dengan demikian, Staff infrastruktur dapat mengidentifikasi potensi masalah atau bottleneck yang muncul pada infrastruktur di PT. XYZ dengan cepat dan tepat waktu untuk melakukan pencegahan terhadap matinya aplikasi microservice yang sedang berjalan. 4 8 9 11 12 15 16 18 26 28 45 1.2

Rumusan/Identifikasi Masalah Berdasarkan latar belakang diatas, dapat diidentifikasi bahwa terdapat beberapa rumusan masalah yaitu : 1. PT. XYZ menghadapi kendala dalam memantau resource dari setiap server yang digunakan oleh aplikasi microservice. Kondisi storage, kondisi CPU, kondisi penggunaan RAM dan status aplikasi microservice dipantau dengan cara manual yaitu diharuskan masuk ke setiap server satu persatu agar dapat menarik data resource tersebut sehingga menciptakan proses berulang yang akan memakan waktu. 2. PT. XYZ tidak memiliki informasi mengenai kondisi resource server yang kritis diluar jadwal pemantauan sehingga tidak dapat melakukan tindakan sebelum aplikasi microservice mati secara otomatis. Dari penjelasan di atas, dapat dihasilkan rumusan masalah untuk memenuhi pengembangan aplikasi yang

sesuai dengan kebutuhan, dengan pernyataan permasalahan berikut ini:  
Bagaimana rancang bangun sistem monitoring resource server dan aplikasi microservice berbasis web dengan notifikasi telegram di PT. XYZ menggunakan pendekatan waterfall? 1.3 Batasan Masalah Penelitian tugas akhir ini akan berfokus pada "Rancang Bangun Aplikasi Monitoring Resource Server dan Microservice Berbasis Web dengan Notifikasi Telegram pada PT. XYZ Menggunakan Pendekatan Waterfall. Batasan masalah mencakup dua aplikasi utama yang akan dikembangkan, yaitu Aplikasi Laporan dan Aplikasi Monitoring . 1. Aplikasi Laporan: Aplikasi ini akan diintegrasikan ke dalam server yang memiliki microservice . Fungsinya adalah melaporkan status dari setiap aplikasi microservice yang berjalan dan juga melaporkan status penggunaan resource dari server tersebut. Informasi yang dikumpulkan oleh Aplikasi Laporan akan secara terjadwal dikirimkan ke Aplikasi Monitoring menggunakan scheduler . Dengan demikian, aplikasi ini berperan sebagai sumber data yang menyediakan informasi terkini mengenai kesehatan dan status aplikasi microservice beserta penggunaan resource pada server. 2. Aplikasi Monitoring : Aplikasi ini akan ditempatkan di server khusus yang menerima data dari aplikasi laporan yang terdapat di berbagai server microservice. Tugas utama aplikasi ini adalah menggabungkan dan memvisualisasikan data yang diterima dari aplikasi laporan dalam bentuk dashboard monitoring , Aplikasi Monitoring akan memberikan representasi yang jelas dan komprehensif tentang status dan penggunaan resource dari setiap server microservice . Selain itu, aplikasi ini dilengkapi dengan mekanisme notifikasi yang akan mengirimkan pesan ke telegram jika terdeteksi adanya penggunaan resource server yang melebihi batas yang telah ditentukan sebelumnya. Dengan demikian, tim operasional dapat merespon secara cepat terhadap situasi yang memerlukan tindakan. Dengan adanya kedua aplikasi ini, diharapkan PT. XYZ dapat mengoptimalkan pemantauan terhadap server microservice mereka, mengidentifikasi potensi masalah secara proaktif, dan merespon dengan efektif terhadap perubahan kondisi yang mungkin mempengaruhi kinerja

aplikasi. Sistem ini diharapkan menjadi solusi yang integral dalam mendukung keberhasilan implementasi arsitektur microservice di perusahaan tersebut.

1.4 Tujuan Penelitian

1. Untuk membantu Staff infrastruktur dalam melakukan monitoring terhadap setiap server microservice yang ada di PT. XYZ.
2. Untuk membantu memberikan pemberitahuan kepada Staff infrastruktur untuk melakukan tindakan terhadap kondisi resource server yang mengalami kritis atau overload .
3. Untuk membantu Staff infrastruktur dalam pengambilan keputusan terhadap pemeliharaan server yang ada di PT. XYZ.

2 1.5 Manfaat Penelitian

1. Bagi Penulis Penulis berharap penelitian ini dapat memberikan pengetahuan terhadap penulis tentang aplikasi monitoring terutama pada resource server dan juga bagaimana memantau kondisi dari aplikasi microservice yang ada didalamnya.
2. Bagi PT. XYZ Penulis berharap dengan penelitian ini dapat memenuhi kebutuhan Staff infrastruktur yaitu berupa sebuah dashboard monitoring yang dapat memberikan informasi mendalam mengenai status server yang digunakan dan juga aplikasi yang ada didalamnya. Dashboard ini bukan hanya sekedar alat pemantauan, tetapi juga dilengkapi dengan sistem notifikasi yang akan memberikan notifikasi terhadap situasi yang memerlukan tindakan segera.
3. Bagi Pembaca Penulis berharap penelitian ini dapat menjadi sebuah referensi bagi pembaca didalam membuat aplikasi monitoring dan juga dapat dikembangkan oleh peneliti selanjutnya supaya penelitian ini dapat menjadi jauh lebih baik serta dapat digunakan di tempat peneliti selanjutnya sebagai sistem yang dapat membantu menangani masalah yang ada terutama didalam pemantauan resource server dan aplikasi microservice yang ada didalamnya.

1.6 Sistematika Penulisan Dalam penyusunan skripsi atau tugas akhir ini, terdapat 7 bab utama yang dapat dijabarkan dalam struktur penulisan berikut:

1. BAB I. **11 33** PENDAHULUAN  
Pendahuluan pada Bab I meliputi latar belakang masalah, identifikasi dan perumusan masalah, ruang lingkup dan batasan masalah, maksud dan tujuan penelitian, dan sistematika penulisan. **33**
2. BAB II. TINJAUAN PUSTAKA Pada bab II, terdapat berbagai teori dasar dan tinjauan studi yang mencakup

penjelasan tentang teori-teori dan kajian-kajian sebelumnya yang mendukung tugas akhir ini. 3. BAB III. TAHAP PELAKSANAAN Pada bab III memuat objek penelitian, metode pengumpulan data, Analisis terhadap sistem yang sudah berjalan dan masalah yang terjadi didalam sistem tersebut berikut alternatif pemecahan masalahnya. 4. BAB IV. HASIL DAN ANALISIS PENELITIAN Pada bab IV memuat tentang hasil dan analisis aplikasi monitoring resource server dan microservice yang dilakukan oleh penulis berdasarkan bab III. Hasil dan analisis tersebut berupa analisa perancangan sistem, perancangan diagram sistem usulan, perancangan antar muka pengguna dan perancangan implementasi. 8 23 48 5. BAB V. PENUTUP Pada bab V memuat tentang kesimpulan dan saran dari rancang bangun aplikasi monitoring yang dilakukan oleh penulis. Didalam bab ini proses dan hasil penelitian yang dilakukan oleh penulis akan disimpulkan dan terdapat beberapa saran untuk membuat sistem tersebut menjadi lebih baik. 6. DAFTAR PUSTAKA Pada daftar Pustaka memuat beberapa sumber referensi yang digunakan oleh penulis pada proses perancangan dan pembangunan aplikasi untuk penelitian tugas akhir. 7. LAMPIRAN Pada lampiran memuat dokumen-dokumen tambahan yang diperlukan oleh penulis pada penelitian tugas akhir. 3 4

BAB II TINJAUAN PUSTAKA Dalam bagian ini, peneliti akan menjelaskan aspek-aspek terkait dengan pengkajian kembali terhadap literatur yang terkait dengan topik penelitian. Hal ini dilakukan untuk mendapatkan bahan perbandingan dan referensi terhadap beberapa teori yang relevan dalam kerangka tugas akhir yang berjudul "Rancang Bangun Aplikasi Monitoring Resource Server dan Microservice Berbasis Web dengan Notifikasi Telegram di PT XYZ Menggunakan Pendekatan Waterfall". Aplikasi yang dimaksud mencakup aplikasi dashboard berbasis web dan aplikasi latar belakang yang beroperasi melalui scheduler. Oleh karena itu dibawah ini merupakan teori teori dasar yang mendukung pengembangan aplikasi dan perancangan aplikasi.

2.1 Teori Dasar 2.1.1 Rancang Bangun Woro Isti Rahayu, Ravi Rahmatul Fajri, Parhan Hambali (2019, p 21) mengungkapkan bahwa Rancang bangun adalah produk penelitian yang membantu peneliti

menyelesaikan masalah pada objek penelitian. **74** Istilah ini terdiri dari dua kata yaitu rancang dan bangun. 1) Rancang Didalam rancang, hasil analisis sistem diterjemahkan ke dalam bahasa pemrograman yang tepat. Hal ini bertujuan untuk menjelaskan implementasi komponen-komponen sistem secara detail dan terstruktur. Tahap rancang ibarat merancang cetak biru sebuah bangunan, di mana setiap elemen dan fungsinya dijabarkan dengan jelas. 2) Bangun Setelah rancangan selesai, tahap selanjutnya adalah bangun, yaitu proses mewujudkan sistem tersebut. Tahap ini melibatkan implementasi kode program, pengujian sistem, dan penyempurnaan hingga sistem siap digunakan. Rancang bangun dalam penelitian ini merupakan produk yang dihasilkan dari penerjemahan data penelitian, seperti observasi, wawancara, studi pustaka, dan dokumentasi. Produk tersebut membantu peneliti dalam membuat perancangan dengan lebih mudah dan efisien. Berikut merupakan beberapa manfaat dari rancang bangun didalam proses penelitian. 1) Mempercepat proses penelitian, Dengan merancang dan membangun sistem secara terstruktur, peneliti dapat menghemat waktu dan tenaga dalam menyelesaikan penelitiannya. 2) Meningkatkan ketepatan dan validitas hasil penelitian, Sistem yang dirancang dan dibangun dengan baik akan menghasilkan hasil penelitian yang lebih akurat dan valid. 3) Memudahkan replikasi penelitian, Rancangan dan bangun yang terdokumentasi dengan baik memungkinkan peneliti lain untuk mereplikasi penelitian dengan mudah. 4) Meningkatkan kolaborasi penelitian, Produk rancang bangun dapat dibagikan dengan peneliti lain, sehingga memungkinkan kolaborasi penelitian yang lebih efektif. 2.1

**27** 2 Sistem Informasi Elisabeth Yunaeti Anggraeni dan Rita Irviani (2017, p. 5) mengungkapkan bahwa Sistem informasi adalah suatu sistem yang menyediakan informasi bagi manajemen untuk pengambilan keputusan serta pelaksanaan operasional perusahaan. Sistem ini merupakan kombinasi dari manusia, teknologi informasi, dan prosedur-prosedur yang terorganisir. Sistem informasi sendiri dirancang untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi. **59** Sistem ini dapat membantu perusahaan untuk meningkatkan efisiensi, efektivitas, dan pengambilan keputusan. Berikut



merupakan komponen-komponen dari sistem informasi. 1) Perangkat Keras ( Hardware ) yaitu berupa komputer, server, jaringan, dan perangkat penyimpanan. 2) Perangkat Lunak ( Software ) yaitu berupa sistem operasi, aplikasi, dan database. 3) Orang ( People ) yaitu berupa pengguna, administrator sistem, dan pengembang. 4) Data yaitu berupa fakta, angka, dan teks yang diolah oleh sistem informasi. 5) Prosedur yaitu berupa aturan dan instruksi untuk mengoperasikan sistem informasi. 6) Jaringan Komunikasi ( Communication Networks ) yaitu berupa internet, intranet, dan ekstranet. 7) Keamanan ( Security ) yaitu berupa keamanan fisik, keamanan data, dan keamanan jaringan. 8) Infrastruktur ( Infrastructure ) yaitu berupa pusat data dan peralatan jaringan seperti router, switch, dan firewall. 9) Kebijakan dan Regulasi ( Policies and Regulations ) yaitu berupa kebijakan internal dan regulasi eksternal terkait penggunaan dan perlindungan data. 10) Manajemen Proyek ( Project Management ) yaitu berupa pengelolaan proyek dan metodologi pengembangan seperti Agile, Waterfall, atau DevOps. Salah satu jenis dari sistem informasi merupakan sebagai sistem pendukung keputusan (SPK). Sistem Pendukung Keputusan (SPK) adalah sebuah sistem informasi yang diciptakan khusus untuk membantu dalam proses pengambilan keputusan dalam situasi yang kompleks dan semi-terstruktur. Dengan menyediakan berbagai alat dan teknik analisis data, serta membangun model, SPK bertujuan untuk menghasilkan solusi yang optimal bagi penggunanya. 5 Komponen utama dari SPK meliputi basis data yang menyimpan informasi terkait masalah yang akan dipecahkan, model matematis yang merepresentasikan masalah, alat analisis untuk menganalisis data dan membangun model, serta antarmuka pengguna yang memfasilitasi interaksi dengan sistem. Manfaat yang diperoleh dari penggunaan SPK sangatlah beragam. Pertama-tama, SPK membantu meningkatkan kualitas pengambilan keputusan dengan memungkinkan pengambil keputusan untuk mempertimbangkan semua faktor yang relevan dan memilih solusi yang paling optimal. **41** Selain itu, SPK juga dapat meningkatkan efisiensi pengambilan keputusan dengan mengotomatisasi beberapa

tugas yang terkait, sehingga menghemat waktu dan sumber daya. Yang tak kalah pentingnya, SPK juga dapat meningkatkan komunikasi dan kolaborasi antara para pengambil keputusan, memungkinkan mereka untuk berinteraksi dengan lebih mudah dan efisien dalam mencapai tujuan bersama. **35** Dengan demikian, SPK tidak hanya memberikan dukungan teknis dalam pengambilan keputusan, tetapi juga memfasilitasi proses komunikasi dan kerja sama yang lebih baik di antara para pemangku kepentingan.

### 2.1.3 Microservice Jose Haro Peralta (2023, p. 1)

mengungkapkan bahwa Microservice merupakan pendekatan arsitektural di mana komponen-komponen dari sebuah sistem dirancang sebagai aplikasi yang mandiri dan dapat dideploy secara independen. **55** Setiap aplikasi yang berfungsi secara independen berkomunikasi melalui application programming interface (API). Karakteristik utama dari Microservices adalah kebebasannya, di mana setiap layanan dapat dikembangkan, diuji, dan dideploy secara terpisah. Layanan ini juga haruslah kecil dan fokus pada satu fungsi spesifik. Komunikasi antar layanan dilakukan melalui API, tanpa adanya titik kontrol terpusat, yang memungkinkan sistem untuk tahan terhadap kegagalan. Manfaat utama dari Microservices termasuk kecepatan dalam pengembangan dan peluncuran aplikasi (agility), kemampuan untuk mengatasi peningkatan traffic dengan penskalaan horizontal (scalability), keandalan dalam menghadapi kegagalan (reliability), serta kemudahan pemeliharaan yang lebih baik dibandingkan dengan aplikasi monolitik. Namun, Microservices juga membawa tantangan tersendiri. Kompleksitas dapat meningkat karena banyaknya layanan yang perlu dikelola, sementara keamanan juga menjadi perhatian karena adanya risiko baru yang muncul. Selain itu, monitoring perlu dilakukan dengan cermat untuk memastikan performa sistem tetap optimal.

### 2.1.4 Server Didik Setiawan (2017, p. 9)

mengungkapkan bahwa Server adalah sebuah sistem komputer yang menyediakan berbagai jenis layanan untuk klien dalam sebuah jaringan komputer. Server ini dilengkapi dengan sistem operasi khusus yang mengatur akses dan sumber daya di dalamnya, yang umumnya disebut sebagai sistem operasi jaringan atau Network Operating System. Server sendiri

berfungsi untuk menerima permintaan ( request ) dari klien, memprosesnya, dan memberikan respon atau hasil kembali kepada klien. **67** Beberapa

jenis server meliputi: 1) Server Web Menyediakan halaman web dan konten internet.

Contoh: Apache, Nginx. 2) Server Database Menyimpan dan mengelola data. Contoh: MySQL, PostgreSQL, MongoDB. 3) Server File Menyimpan dan mengelola file yang dapat diakses oleh klien. Contoh: FTP server, file server. 4) Server Aplikasi Menjalankan aplikasi tertentu dan menyediakan

layanan aplikasi. Contoh: Java EE server, Microsoft IIS. 5) Server

Email Menyediakan layanan email. Contoh: Microsoft Exchange Server, Postfix. **42** Server

dapat berupa perangkat keras ( hardware ) yang berupa komputer fisik

atau perangkat lunak ( software ) yang berjalan di atas suatu sistem komputasi. Secara

umum, server dirancang untuk memberikan layanan yang dapat diakses oleh

banyak klien secara bersamaan. Keandalan, kecepatan, dan ketersediaan

tinggi sering menjadi fokus dalam desain dan pengelolaan server. Didalam

Server sendiri terdapat komponen-komponen atau sumberdaya yang dapat

digunakan oleh server untuk menjalankan aplikasi atau menyediakan layanan.

Beberapa sumber daya utama dalam server melibatkan: 1) CPU (Central

Processing Unit) CPU adalah otak dari server yang menangani pemrosesan

instruksi dan eksekusi tugas-tugas komputasi. Kapasitas dan kecepatan CPU

mempengaruhi kemampuan server dalam menangani beban kerja. **64** 2) RAM (Random

Access Memory) RAM menyimpan data sementara yang sedang digunakan oleh server. **63**

Semakin besar kapasitas RAM, semakin banyak data dan aplikasi yang

dapat diakomodasi oleh server. **38** 3) Penyimpanan (Storage) Sumber daya

penyimpanan termasuk hard drive atau solid-state drive (SSD) yang

digunakan untuk menyimpan sistem operasi, aplikasi, dan data. Kapasitas dan jenis

penyimpanan dapat memengaruhi ketersediaan dan kinerja server. 4) Jaringan

( Network ) 6 Sumber daya jaringan mencakup ketersediaan bandwidth dan

konektivitas. Ketersediaan jaringan yang baik penting untuk menyediakan

layanan kepada pengguna dan memastikan konektivitas dengan sumber daya eksternal.

**66** 5) Bandwidth Bandwidth adalah kapasitas maksimum untuk mentransfer

data dalam suatu jaringan. Sumber daya ini penting untuk menangani jumlah

permintaan dan lalu lintas data yang diterima atau dikirim oleh server. 6) Prosesor Grafis (GPU) GPU dapat digunakan untuk akselerasi komputasi tertentu, terutama dalam konteks komputasi paralel, rendering grafis, atau aplikasi kecerdasan buatan. Pemantauan dan manajemen sumber daya sangat penting dalam menjaga ketersediaan, performa, dan keamanan server. Pemilihan sumber daya yang sesuai dengan kebutuhan aplikasi dan beban kerja tertentu juga menjadi pertimbangan penting dalam merancang dan mengelola server.

2.1.5 Monitoring James Turnbull (2014, p. 9) mengungkapkan bahwa monitoring merupakan alat dan proses yang digunakan untuk mengukur dan mengelola suatu sistem. Tujuan utama dari monitoring adalah untuk memastikan kinerja yang optimal, keandalan, keamanan, dan ketersediaan suatu sistem atau proses. Beberapa aspek umum dari monitoring melibatkan:

- 1) Pengumpulan Data Monitoring melibatkan pengumpulan data terkait dengan kinerja, status, atau perilaku suatu sistem. Data ini bisa melibatkan berbagai metrik, log, atau informasi lainnya yang diperlukan untuk evaluasi.
- 2) Analisis Data Data yang dikumpulkan dianalisis untuk mengidentifikasi tren, pola, atau anomali yang dapat menunjukkan masalah atau perubahan kondisi.
- 3) Notifikasi dan Peringatan Berdasarkan hasil analisis, sistem monitoring dapat menghasilkan notifikasi atau peringatan kepada administrator atau pengguna ketika terjadi situasi yang memerlukan perhatian atau tindakan.
- 4) Pelaporan Monitoring juga seringkali melibatkan pembuatan laporan yang merinci kinerja, keandalan, atau kejadian tertentu dalam suatu periode waktu. Laporan ini dapat digunakan untuk evaluasi jangka panjang atau pemantauan tren.
- 5) Pengambilan Keputusan Informasi yang diberikan oleh monitoring dapat digunakan sebagai dasar untuk pengambilan keputusan terkait perbaikan, peningkatan kinerja, atau tindakan lainnya untuk menjaga atau meningkatkan kondisi suatu sistem.

2.1.6 Dashboard Dover (2004, p. 44) mengungkapkan bahwa dashboard adalah alat yang menyajikan ringkasan sekilas dalam format visual dan intuitif yang sangat mudah dipahami. Ungkapan tersebut menekankan bahwa tujuan utama dari dashboard

adalah memberikan informasi secara cepat dan efisien kepada pengguna, sehingga mereka dapat dengan cepat membuat keputusan atau mengambil tindakan yang tepat. Berikut merupakan manfaat dari dashboard antara lain: 1) Mempermudah pemantauan Dashboard memungkinkan pengguna untuk memantau berbagai data penting dalam satu tempat. 60 2) Meningkatkan pemahaman Dashboard menyajikan data secara visual sehingga mudah dipahami dan dianalisis. 40 3) Mempercepat pengambilan keputusan Dashboard membantu pengguna untuk mengambil keputusan yang tepat dengan cepat berdasarkan data yang akurat. 4) Meningkatkan efisiensi Dashboard membantu pengguna untuk menghemat waktu dan sumber daya dengan memvisualisasi data yang kompleks. Disamping itu dashboard memiliki beberapa jenis yang masing-masing memiliki fungsi dan karakteristiknya, yaitu : 1) Dashboard Operasional Digunakan untuk memantau kinerja dan kesehatan sistem secara real-time. 2) Dashboard Analitik Digunakan untuk menganalisis data dan mengidentifikasi tren. 3) Dashboard Strategis Digunakan untuk memantau kemajuan dan kinerja organisasi secara keseluruhan. 2.1.7 SDLC Arief Yanto Rukmana, dkk (2023, p 125) mengungkapkan bahwa dalam upaya pengembangan sistem informasi, diperlukan langkah-langkah formal untuk mencegah kegagalan dan memastikan kesesuaian dengan kebutuhan organisasi atau perusahaan. 53 Proses formal ini dikenal sebagai Siklus Hidup Pengembangan Sistem atau biasa disebut SDLC (System Development Life Cycle). 7 SDLC merupakan suatu proses yang melibatkan pemahaman terhadap bagaimana sistem informasi digunakan dan mendukung kebutuhan bisnis dengan merancang sistem, membangunnya, dan mengirimkannya kepada pengguna. Menurut Alan Dannis dkk (2015) mengungkapkan bahwa tahapan SDLC berupa perencanaan, analisis, desain dan implementasi. 46 1. Perencanaan ( Planning ) Fase perencanaan merupakan awal dari SDLC di mana tujuan, ruang lingkup, sumber daya, jadwal, dan risiko proyek ditentukan. Pada tahap ini, tim proyek dan pemangku kepentingan lainnya merumuskan rencana keseluruhan untuk proyek pengembangan perangkat lunak. Aktivitas yang biasa dilakukan dalam fase perencanaan meliputi : 1) Mendefinisikan tujuan proyek, Apa yang ingin dicapai dengan proyek ini?

2) Menganalisis kebutuhan pengguna, Siapa yang akan menggunakan software ini? Apa kebutuhan dan ekspektasi mereka? 3) Menentukan ruang lingkup proyek, Fitur apa yang akan diimplementasikan? Apa yang tidak termasuk dalam proyek? 4) Memperkirakan waktu dan biaya, Berapa lama waktu yang diperlukan untuk menyelesaikan proyek ini? Berapa besar biaya yang dibutuhkan? 5) Memilih metodologi pengembangan, Metodologi apa yang akan digunakan untuk mengembangkan software ?

2. Analisis ( Analysis ) Fase analisis bertujuan untuk memahami secara menyeluruh kebutuhan dan persyaratan pengguna untuk sistem yang akan dikembangkan. Tim proyek akan menganalisis persyaratan fungsional dan non-fungsional yang diperlukan oleh pengguna. Selama fase ini, sering kali dilakukan wawancara dengan pengguna potensial, observasi langsung, dan pengumpulan dokumen yang relevan. Aktivitas yang biasa dilakukan dalam fase Analisis meliputi: 1) Mengumpulkan data, Data dikumpulkan dari berbagai sumber, seperti wawancara, survei, observasi, dan dokumentasi. 2) Menganalisis data, Data dianalisis untuk memahami kebutuhan pengguna, fungsionalitas yang dibutuhkan, dan batasan-batasan sistem. 3) Membuat model sistem, Model sistem dibuat untuk menggambarkan bagaimana sistem akan bekerja. 4) Menetapkan spesifikasi sistem, Spesifikasi sistem dibuat untuk mendokumentasikan kebutuhan dan fungsionalitas sistem. Hasil dari fase analisis ini biasanya berupa dokumen spesifikasi kebutuhan yang menjadi pedoman bagi tahap selanjutnya.

3. Desain ( Design ) Fase desain melibatkan konversi persyaratan yang telah dianalisis menjadi rancangan teknis yang konkret untuk sistem yang akan dikembangkan. Hal tersebut mencakup perencanaan arsitektur sistem, pemodelan data, desain antarmuka pengguna, dan rancangan algoritma. Tujuan dari fase ini adalah untuk menghasilkan rencana detail tentang bagaimana sistem akan dibangun, termasuk pemilihan teknologi yang tepat. Aktivitas yang biasa dilakukan dalam fase Desain meliputi: 1) Merancang arsitektur sistem, Arsitektur sistem menentukan bagaimana komponen-komponen sistem akan terhubung dan berinteraksi. 2) Merancang antarmuka pengguna ( User Interface ),

Antarmuka pengguna dirancang agar mudah digunakan dan dipahami oleh pengguna ( User ). 3) Merancang database , Database dirancang untuk menyimpan data yang digunakan oleh sistem. 4) Mengembangkan prototipe, Prototipe dibuat untuk menguji desain sistem dan mendapatkan masukan dari pengguna. Hasil dari fase desain ini adalah serangkaian dokumen desain yang akan digunakan oleh tim pengembang selama implementasi. 4. Implementasi ( Implementation ) Fase implementasi merupakan titik krusial dalam proses pengembangan perangkat lunak, di mana tim pengembang secara aktif menerjemahkan rancangan menjadi kumpulan kode yang berfungsi. 65 Dalam proses ini, peran masing-masing tim, baik frontend maupun backend , menjadi sangat penting.

1) Frontend Pada bagian frontend , fokus utamanya adalah pada pembuatan antarmuka pengguna yang responsif dan menarik. 47 Tim frontend bertanggung jawab untuk mengimplementasikan desain yang telah dirancang secara visual ke dalam kode HTML, CSS, dan JavaScript. Mereka harus memastikan bahwa pengguna dapat berinteraksi dengan aplikasi secara intuitif dan lancar. Dalam hal ini, kemampuan untuk menguji antarmuka pengguna secara menyeluruh sangat penting untuk memastikan bahwa setiap elemen bekerja seperti yang diharapkan. 2) Backend Sementara itu, di sisi backend , fokusnya adalah pada pengembangan logika bisnis dan pengelolaan data. Tim backend bertanggung jawab untuk membuat API yang kuat dan efisien untuk mengelola interaksi antara antarmuka 8 pengguna dan basis data. Tim backend harus memastikan bahwa sistem dapat menangani permintaan dari klien dengan cepat dan dapat diandalkan, serta memiliki mekanisme keamanan yang kokoh untuk melindungi data sensitif. Selama fase implementasi, integrasi antara frontend dan backend menjadi kunci. Tim harus bekerja sama untuk memastikan bahwa antarmuka pengguna terhubung dengan lancar dengan logika bisnis di backend. Hal tersebut melibatkan pengujian integrasi yang cermat untuk memastikan bahwa semua komponen berinteraksi dengan benar dan menghasilkan hasil yang diinginkan. Selain itu, pengujian unit juga penting di kedua sisi. Tim harus secara rutin menguji setiap komponen perangkat lunak secara terpisah untuk

memastikan bahwa mereka beroperasi dengan baik dan tidak menyebabkan masalah saat diintegrasikan ke dalam sistem secara keseluruhan. **44** Dalam keseluruhan proses implementasi, komunikasi dan kolaborasi yang efektif antara tim frontend dan backend sangat penting. Hanya dengan kerja sama yang solid, sistem dapat dikembangkan dengan lancar dan efisien, menghasilkan perangkat lunak yang berkualitas tinggi dan memuaskan kebutuhan pengguna dengan baik. Hasil dari fase ini adalah sistem yang siap untuk diuji lebih lanjut dalam fase pengujian. Aktivitas yang biasa dilakukan dalam fase Implementasi meliputi : 1)) Mengembangkan kode program, Kode program ditulis berdasarkan desain yang telah dibuat. 2)) Menguji software, Software diuji untuk memastikan bahwa software tersebut bebas dari bug dan berfungsi sesuai dengan spesifikasi. 3)) Memperbaiki bug, Bug yang ditemukan selama pengujian diperbaiki. 4)) Melakukan deployment , Software diinstal pada lingkungan produksi. Keempat fase dalam SDLC saling terkait dan berurutan. Fase perencanaan menentukan arah proyek, fase analisis memahami kebutuhan pengguna, fase desain merancang solusi, dan fase implementasi mengembangkan dan menguji software. Terdapat dua konsep penting yang harus dipahami tentang SDLC . Pertama, pemahaman menyeluruh tentang tahapan, langkah-langkah, dan teknik yang harus dilewati oleh proyek sistem informasi. Bagian ini akan menjelaskan tentang tahapan, langkah-langkah, dan teknik yang terlibat dalam SDLC . Kedua, penting untuk menyadari bahwa SDLC merupakan proses peningkatan berkelanjutan. Hal tersebut berarti bahwa hasil dari fase analisis memberikan gambaran umum tentang apa yang akan dilakukan oleh sistem baru. Hasil ini kemudian digunakan sebagai masukan dalam fase desain, yang kemudian disempurnakan untuk menghasilkan deskripsi yang lebih rinci tentang bagaimana sistem akan dibangun. Hasil dari fase ini akan digunakan dalam fase implementasi untuk mengarahkan pembuatan sistem yang sebenarnya. **57** Setiap fase bertujuan untuk menyempurnakan dan memperjelas pekerjaan yang telah dilakukan sebelumnya. Konsep ini secara khas terkait dengan model pengembangan perangkat lunak " Waterfall " atau "Air Terjun".



Model ini menggambarkan proses pengembangan perangkat lunak secara bertahap, dimulai dari tahap awal hingga tahap akhir, dengan setiap tahap menghasilkan keluaran yang menjadi masukan bagi tahap berikutnya. 50 Analoginya, model Waterfall menyerupai aliran air terjun yang mengalir secara berurutan dari satu fase ke fase berikutnya. Ronald E. Giachetti (2016, p. 88) mengungkapkan bahwa Model Waterfall menggambarkan sebuah metodologi di mana fase-fase dilakukan secara berurutan, menggunakan metafora air yang jatuh di tangga-tangga. Model waterfall mendefinisikan fase-fase dalam hal aktivitas mereka dimana setiap fase memiliki satu aktivitas. Berikut adalah fase- fase yang ada dalam model Waterfall .

1. Requirement Analysis (Analisis Kebutuhan)

- 1) Mengumpulkan dan mendokumentasikan semua kebutuhan sistem dari pemangku kepentingan.
- 2) Membuat spesifikasi kebutuhan yang jelas dan lengkap.

2. System Design (Desain Sistem):

- 1) Mengembangkan arsitektur sistem berdasarkan kebutuhan yang telah dianalisis.
- 2) Membuat desain detail yang mencakup struktur data, arsitektur perangkat lunak, antarmuka, dan algoritma.

3. Implementation (Implementasi):

- 1) Mengubah desain menjadi kode sumber.
- 2) Mengembangkan modul sesuai dengan rancangan yang telah dibuat pada tahap sebelumnya.

4. Integration and Testing (Integrasi dan Pengujian):

- 1) Mengintegrasikan semua modul yang telah dikembangkan.
- 2) Melakukan pengujian untuk memastikan bahwa sistem bekerja sesuai dengan kebutuhan yang telah ditentukan, termasuk pengujian unit, integrasi, sistem, dan penerimaan.

5. Deployment (Penempatan):

- 1) Memasang sistem yang telah diuji di lingkungan produksi. 56
- 2) Melakukan konfigurasi yang diperlukan dan memastikan bahwa sistem berjalan dengan baik di lingkungan produksi.

6. Maintenance (Pemeliharaan):

- 1) Memantau sistem untuk menemukan dan memperbaiki masalah yang muncul. 28
- 2) Melakukan perbaikan bug dan peningkatan sistem berdasarkan umpan balik pengguna dan perubahan kebutuhan.

Kelebihan model waterfall mencakup struktur yang jelas, dokumentasi yang rinci, dan kesederhanaan manajemen proyek. Namun, kekurangannya adalah kurangnya fleksibilitas jika perubahan diperlukan setelah tahap dimulai, dan risiko tinggi terkait dengan menunggu hingga

akhir untuk melihat produk yang sebenarnya. Model ini sering digunakan dalam proyek-proyek di mana persyaratan diperkirakan cukup stabil dan perubahan tidak terlalu sering terjadi.

### 2.1.8 Data Flow Diagram

Hanif Al Fatta (2007, p. 105) mengungkapkan bahwa data flow diagram (DFD) merupakan sebuah diagram yang digunakan untuk menggambarkan proses-proses yang terjadi dalam sistem yang akan dibangun. DFD menggambarkan proses-proses yang terjadi dalam sistem, aliran data antara proses-proses tersebut, penyimpanan data, serta sumber dan tujuan data. Diagram ini membantu dalam memahami, menganalisis, dan mendokumentasikan sistem, sehingga mempermudah pengembangan dan perbaikan sistem tersebut. Berikut merupakan komponen dari data flow diagram (DFD) Didalam data flow diagram terdapat tingkatan dimulai dari DFD level 0 hingga DFD level 2 dengan detail sebagai berikut.,

1. Level 0 ( Context Diagram ): Merupakan gambaran umum dari sistem, menunjukkan hubungan antara sistem dengan entitas eksternal tanpa merinci proses internal.
2. Level 1: Menunjukkan rincian lebih lanjut dari proses-proses yang ada di Level 0. Setiap proses di Level 0 dipecah menjadi sub-proses di Level 1.
3. Level 2 dan seterusnya: Menyediakan rincian lebih lanjut dari proses-proses di Level 1, dan seterusnya, hingga mencapai tingkat detail yang diperlukan.

### 2.1.9 Entity Relationship Diagram

Hanif Al Fatta (2007, p. 105) mengungkapkan bahwa Entity Relationship Diagram (ERD) merupakan diagram yang menggambarkan bagaimana informasi dibuat, disimpan, dan digunakan dalam suatu sistem bisnis. Diagram ini sangat berguna dalam perancangan basis data dan sistem informasi karena membantu menggambarkan hubungan antara berbagai entitas dalam sistem tersebut. ERD terdiri dari beberapa komponen utama. Dalam ERD, hubungan (relasi) dapat melibatkan beberapa entitas, yang disebut dengan derajat relasi. Derajat relasi maksimum dikenal sebagai kardinalitas relasi, sedangkan derajat minimum disebut modalitas.

**13** Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat terhubung dengan entitas lain dalam himpunan entitas yang berbeda. Kardinalitas relasi antara dua himpunan entitas (misalnya,

A dan B) dapat berupa: 1. One-to-One (1:1) Satu entitas dalam set A berhubungan dengan tepat satu entitas dalam set B, dan sebaliknya. 2. One-to-Many (1:M) Satu entitas dalam set A berhubungan dengan banyak entitas dalam set B. 3. Many-to-Many (M:M) Banyak entitas dalam set A berhubungan dengan banyak entitas dalam set B.

### 2.1.10 UML Ir. M. Farid Aziz, M.Kom (2005, p. 116) mengungkapkan bahwa UML (Unified Modeling Language) merupakan serangkaian simbol dan diagram yang digunakan untuk memodelkan perangkat lunak. Dengan menggunakan UML, perancangan perangkat lunak dapat direpresentasikan dalam bentuk simbol dan diagram. Representasi ini kemudian dapat diubah menjadi kode program.

10 21 UML menyediakan notasi grafis yang dapat digunakan untuk menggambarkan berbagai aspek dari sistem perangkat lunak, mulai dari struktur statis hingga perilaku dinamis. Terdapat 4 model dari UML yang sering digunakan diantaranya :

- 6 Use Case Diagram Use Case Diagram digunakan untuk menggambarkan interaksi antara sistem dan entitas luar (biasanya pengguna atau sistem lain). Berikut merupakan elemen utama dari use case diagram :
- 6 26 Activity Diagram Activity Diagram menunjukkan alur kerja atau aktivitas dalam suatu sistem atau proses bisnis. Berikut merupakan elemen utama dari activity diagram :
- 6 30 61 Sequence Diagram: Sequence Diagram menunjukkan urutan pesan atau interaksi antar objek dalam suatu skenario. Berikut merupakan elemen utama dari sequence diagram:
- 6 21 30 Class Diagram Class Diagram digunakan untuk menggambarkan struktur kelas dan hubungan antar kelas dalam sistem. Berikut merupakan elemen utama dari class diagram: UML digunakan untuk membantu pemahaman, desain, dan dokumentasi sistem perangkat lunak secara visual. Setiap diagram memiliki tujuan dan konteks penggunaan tertentu dalam proses pengembangan perangkat lunak.

### 2.1.11 Penjadwalan ( Scheduler ) Dana Marsetiya Utama (2023, p. 1) mengungkapkan bahwa Penjadwalan adalah perencanaan urutan kerja dan alokasi sumber daya ( resource ) pada setiap operasi. Scheduler berfungsi untuk merencanakan dan mengatur alokasi sumber daya atau mesin untuk mengeksekusi serangkaian tugas dalam suatu jangka waktu tertentu. Hal ini melibatkan pengaturan urutan kerja dan alokasi sumber

daya, baik dalam hal waktu maupun fasilitas, untuk setiap operasi yang perlu diselesaikan. Dengan kata lain, penjadwalan adalah proses perencanaan yang melibatkan pengorganisasian tugas dan sumber daya guna mencapai tujuan tertentu dalam waktu yang ditentukan. Penjadwalan lebih dari Sekedar Urutan Kerja, Penjadwalan bukan sekedar menyusun daftar tugas. penjadwalan merupakan seni merancang strategi, memetakan alur kerja, dan mengalokasikan sumber daya secara optimal. penjadwalan adalah kunci untuk mencapai tujuan dalam waktu yang ditentukan, meminimalisir pemborosan, dan memaksimalkan hasil. Berikut merupakan beberapa manfaat dari dilakukannya penjadwalan. 1) Efisiensi Penjadwalan membantu menyelesaikan tugas dengan lebih cepat dan tepat waktu, mengantarkan pada gerbang efisiensi. 2) Kualitas Dengan fokus pada tugas-tugas penting, hasil yang berkualitas pun tercipta. 3) Produktivitas Pencapaian optimal dalam waktu singkat, mengantarkan pada puncak performa. Untuk membuat suatu penjadwalan yang efektif diperlukan langkah-langkah berikut : 1) Menentukan Tujuan, Apa yang ingin dicapai? Menjadikannya visi yang jelas. 2) Mendefinisikan Tugas, Merinci langkah-langkah yang mengantarkan pada visi. 3) Memperkirakan Waktu, Memetakan durasi setiap langkah, mengukur waktu yang dibutuhkan. 4) Menentukan Sumber Daya, Mengidentifikasi alat, bahan, dan personil yang diperlukan. 5) Menyusun Jadwal, Merajut urutan langkah dan mengalokasikan sumber daya, bagaikan seniman yang melukis masterpiece alur kerja. 6) Memantau dan Mengevaluasi, Melacak kemajuan dan melakukan penyesuaian, memastikan alur kerja tetap optimal. 2.1.12 Api Waskitho Cito Adiwiguno, Muhammad Wildan Khalilurrahman, Rolly Maulana Awangga (2023, p. 5) mengungkapkan bahwa API (Application Programming Interface) adalah suatu antarmuka yang memungkinkan pengembang perangkat lunak untuk mengakses fungsionalitas yang telah dibuat oleh pengembang lain tanpa perlu mengetahui rincian implementasi internalnya.

22 Dengan kata lain API berperan sebagai jembatan komunikasi antara dua aplikasi atau komponen perangkat lunak yang berbeda, memungkinkan mereka untuk saling berinteraksi dan berbagi data atau fungsi tanpa

harus mengetahui secara detail bagaimana komponen tersebut diimplementasikan. Konsep utama dari API adalah abstraksi. API menyediakan suatu abstraksi atas fungsionalitas atau layanan yang disediakan oleh suatu sistem atau komponen, menyembunyikan detail-detail kompleks tentang bagaimana fungsi-fungsi tersebut diimplementasikan secara internal. Dengan demikian, pengguna API hanya perlu memahami cara menggunakan antarmuka yang telah ditentukan oleh pengembang tanpa harus terlibat dalam kompleksitas dari implementasi di balik layar. API dapat mengambil berbagai bentuk, termasuk API web, hardware API, dan Software API. API web adalah antarmuka yang disediakan oleh aplikasi web atau layanan web yang memungkinkan pengguna untuk berinteraksi dengannya melalui protokol komunikasi web seperti HTTP. Contoh dari API web termasuk RESTful API dan GraphQL API yang digunakan untuk mengakses dan mengelola data dari server web. 2.1.13 Json Ben Smith (2015, p. 37) mengungkapkan bahwa JSON (JavaScript Object Notation) dikenal sebagai standar pertukaran data, yang secara tersirat menyiratkan bahwa ia dapat digunakan sebagai format data di mana pun pertukaran data terjadi. JSON berbasis teks dan terinspirasi oleh sintaks objek dalam bahasa pemrograman JavaScript, namun JSON dapat digunakan dengan berbagai bahasa pemrograman yang mendukung manipulasi string. JSON digunakan sebagai standar pertukaran data menunjukkan fleksibilitasnya didalam beradaptasi dengan lingkungan teknologi yang beragam. Sifatnya yang berbasis teks memungkinkan JSON digunakan dalam konteks apapun di mana pertukaran data diperlukan, baik itu dalam aplikasi web, perangkat lunak desktop, aplikasi mobile, hingga sistem terdistribusi. Namun, yang membuatnya lebih menarik adalah kemampuannya untuk digunakan dengan berbagai bahasa pemrograman lainnya. Oleh karena itu JSON menjadi pilihan yang populer dan dapat diandalkan dalam ekosistem pengembangan perangkat lunak yang berbeda. Pentingnya JSON dalam pertukaran data tidak hanya terbatas pada lingkungan pengembangan perangkat lunak, tetapi juga merambah ke domain lain seperti pertukaran data antar sistem, pengiriman data melalui jaringan, dan penyimpanan

data dalam format yang mudah diakses dan dimanipulasi. Dengan segala keunggulan ini, tidak mengherankan jika JSON terus menjadi salah satu standar de facto dalam pertukaran data di berbagai industri dan aplikasi teknologi saat ini. 2.1.14 Database Ramadhani (2016, p. 105)

Database merupakan tempat penampungan semua data yang ada didalam sistem komputer. Didalam database , data disimpan dalam tabel atau struktur data lainnya dan diorganisir sedemikian rupa sehingga dapat dengan efisien dikelola, diakses, dan dimanipulasi. Database digunakan untuk menyimpan informasi yang berkaitan dengan suatu domain atau aplikasi tertentu. 19

Beberapa konsep dasar dalam database melibatkan: 1) Tabel Tabel

adalah struktur dasar dalam database yang menyimpan data dalam bentuk baris dan kolom. 1

3 5 14 19 Setiap baris dalam tabel mewakili satu record atau entitas,

sedangkan setiap kolom mewakili atribut atau field dari record tersebut. 2) Relasi

Hubungan antar tabel dalam database dapat ditentukan melalui kunci asing ( foreign key ). Hal tersebut memungkinkan penggabungan atau pengambilan data dari beberapa tabel sekaligus. 1 3) Query Query digunakan untuk

mengambil atau memanipulasi data dalam database . 1 3 5 17 24 43 62 Bahasa

pemrograman yang umum digunakan untuk menulis query adalah SQL (Structured Query Language).

58 4) Indeks Indeks digunakan untuk meningkatkan kinerja pencarian data dalam database .

Indeks mempercepat proses pencarian data dengan menciptakan struktur yang memetakan nilai tertentu ke lokasi fisik data. 14 5) Integritas Data

Database memiliki mekanisme untuk menjaga integritas data, seperti aturan

unik, kunci primer, dan kunci asing, untuk memastikan keakuratan dan konsistensi data. 1

3 5 9 12 17 34 6) Sistem Manajemen Basis Data (DBMS) DBMS ( Database

Management System ) adalah perangkat lunak yang digunakan untuk membuat,

mengelola, dan menyediakan antarmuka untuk bekerja dengan database. 1 3 5 24 68

Contoh DBMS meliputi MySQL, PostgreSQL, Microsoft SQL Server, dan Oracle Database .

Database digunakan dalam berbagai konteks, termasuk bisnis, pemerintahan, pendidikan, dan aplikasi perangkat lunak. Mereka membantu dalam penyimpanan dan pengelolaan data dengan efisien, serta menyediakan mekanisme untuk pencarian, pembaruan, dan analisis data. Selain itu

terdapat berbagai macam database antara lain: 1) Relational Database Management System (RDBMS) RDBMS menggunakan struktur tabel yang terdiri dari baris dan kolom untuk menyimpan dan mengorganisir data. Relasi antar tabel ditentukan oleh kunci primer dan kunci asing. 1 3 24 Contoh dari RDMS yaitu MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server. 2) NoSQL Database NoSQL databases menggunakan model data yang berbeda dari RDBMS . Mereka dapat menyimpan data dalam bentuk dokumen, kolom, grafik, atau jenis lainnya. Mereka sering digunakan untuk aplikasi yang membutuhkan skalabilitas horizontal dan fleksibilitas skema. Contoh dari NoSQL database yaitu MongoDB, Cassandra, Couchbase . 3) Graph Database Graph databases dirancang khusus untuk menyimpan dan mengelola data yang memiliki struktur grafik. Mereka menggunakan node , relasi, dan properti untuk merepresentasikan dan menghubungkan entitas dalam basis data. Contoh dari Graph Database yaitu Neo4j, Amazon Neptune, OrientDB. 4) In-Memory Database In-Memory database menyimpan data dalam memori utama komputer, yang memungkinkan akses data yang sangat cepat. Mereka sering digunakan untuk mengatasi kinerja dan latensi dalam aplikasi yang membutuhkan respons waktu yang cepat. Contoh dari In-Memory Database yaitu Redis, Memcached, VoltDB. 5) Time-Series Database Time-series databases dioptimalkan untuk menyimpan dan menganalisis data yang dikumpulkan dari waktu ke waktu, seperti data sensor, log, atau metrik. Mereka menawarkan fitur-fitur khusus untuk mengelola data berbasis waktu dengan efisien. Contoh dari Time-Series Database yaitu InfluxDB, Prometheus, TimescaleDB. 6) Document-Oriented Database Database yang berorientasi pada dokumen menyimpan data dalam bentuk dokumen, sering dalam format seperti JSON atau BSON . Mereka cocok untuk aplikasi yang membutuhkan fleksibilitas skema dan menyimpan data semi- terstruktur. Contoh dari Document-Oriented Database yaitu CouchDB, Firebase Firestore, Amazon DocumentDB. Setiap jenis database memiliki kelebihan dan kekurangan yang berbeda, dan pemilihan yang tepat tergantung pada kebutuhan spesifik aplikasi atau proyek yang sedang dihadapi. 2.1.15 Web Deris Stiawan

(2005, p 149) mengungkapkan bahwa Web atau World Wide Web (WWW) adalah sistem yang memfasilitasi pertukaran informasi melalui jaringan Internet. Web dibangun dengan menggunakan program yang dikenal sebagai server web . Saat ini, pengaruh web telah menjangkau berbagai sektor kehidupan, termasuk dunia bisnis, administrasi pemerintahan, pendidikan, kegiatan keagamaan, interaksi sosial, dan warisan budaya. Untuk memastikan bahwa informasi yang disajikan dalam halaman web dapat diakses oleh pengguna Internet, diperlukan keberadaan mesin server web . Mesin ini bertugas untuk menanggapi permintaan yang datang dari pengguna yang ingin mengakses informasi tersebut. Sebagaimana yang telah diuraikan sebelumnya, beberapa jenis server web yang paling umum digunakan adalah Apache dan IIS. Mesin-mesin ini memiliki peran vital dalam memfasilitasi akses pengguna ke berbagai halaman web di seluruh dunia, memainkan peranan penting dalam memastikan konektivitas yang efisien dan andal di ranah digital saat ini.

2.2 Tinjauan Studi 12 Penelitian ini dilakukan setelah penulis mengumpulkan informasi dari berbagai jurnal untuk digunakan sebagai rujukan dalam penulisan ini. Berikut beberapa sumber yang dapat menyokong penelitian ini, adalah sebagai berikut:

1. Tinjauan studi pada jurnal yang berjudul “ MONITORING SERVER DENGAN PROMETHEUS DAN GRAFANA SERTA NOTIFIKASI TELEGRAM . Jurnal ini ditulis oleh Dede Rahman, Hydra Amnur, dan Indri Rahmayuni pada tahun 2020 dan diterbitkan oleh Jurnal Ilmiah Teknologi Sistem Informasi. Masalah yang diangkat dalam artikel ini adalah peningkatan kompleksitas jaringan, kelemahan dalam sistem topologi jaringan, dan keterbatasan efisiensi pemantauan jaringan dengan metode tradisional. Administrator jaringan, yang bertanggung jawab untuk memastikan jaringan beroperasi dengan optimal, sering kali menemukan bahwa metode pemantauan konvensional kurang efisien karena memerlukan kehadiran fisik mereka di depan layar. Untuk mengatasi masalah ini, para peneliti memperkenalkan solusi dengan menggunakan aplikasi Prometheus dan Grafana untuk pemantauan jaringan secara real-time. Grafana menyediakan informasi langsung mengenai kondisi komponen



jaringan dan mengirimkan pesan ke administrator melalui Telegram. Artikel ini menyimpulkan bahwa pemantauan server berhasil dilakukan menggunakan Prometheus dan Grafana pada server, dengan Ubuntu Server versi 18.04 sebagai sistem operasi yang digunakan. Sistem ini mampu memberikan notifikasi kepada administrator saat terjadi masalah seperti kegagalan CPU, memori, atau layanan Apache dan MySQL. Selain itu, Grafana secara konsisten mengirimkan pemberitahuan melalui Telegram jika kondisi server melebihi batas yang telah ditetapkan atau jika layanan Apache atau MySQL berhenti beroperasi. Pemilihan konfigurasi didasarkan pada spesifikasi minimal untuk membuat instance yang mencakup VCPU, RAM, dan Disk. 2.

Tinjauan studi pada jurnal yang berjudul “DESAIN DAN IMPLEMENTASI SISTEM MONITORING SUMBER DAYA SERVER MENGGUNAKAN ZABBIX 4.0”. Jurnal ini ditulis oleh Sulasno Sulasno dan Rakhmat Saleh pada tahun 2020. Artikel ini membahas dampak peningkatan penggunaan aplikasi di instansi pemerintah dan perusahaan swasta yang mengakibatkan peningkatan jumlah server yang digunakan. Tantangan yang dihadapi banyak server tanpa sistem pemantauan yang memadai adalah kesulitan dalam memantau penggunaan sumber daya secara real-time. Ketika terjadi gangguan operasional seperti kapasitas hard disk penuh atau kinerja CPU melebihi batas, pemantauan yang cepat untuk penanganan segera menjadi sulit. Penelitian ini bertujuan untuk mengembangkan sistem pemantauan sumber daya server secara real-time. Metode penelitian meliputi pengumpulan data, desain sistem, instalasi, konfigurasi, dan implementasi menggunakan Zabbix 4.0 di Pusat Pendayagunaan Informatika dan Kawasan Strategis Nuklir (PPIKSN) Badan Tenaga Nuklir Nasional (BATAN). Hasil penelitian mencakup sistem pemantauan penggunaan sumber daya server, seperti kapasitas hard disk, kinerja CPU, penggunaan memori, besaran paket data ethernet, dan notifikasi email. Sistem ini telah digunakan oleh Administrator server, memberikan manfaat berupa kemudahan pemantauan sumber daya server secara real-time, peningkatan efisiensi dan produktivitas, serta identifikasi dan penanganan masalah sebelum muncul keluhan, yang pada akhirnya meningkatkan

kualitas layanan. 3. Tinjauan studi pada jurnal yang berjudul “RANCANG BANGUN SISTEM INFORMASI MONITORING SERVER VIRTUAL BERBASIS WEB MENGGUNAKAN SCRIPT MONITORING PADA PROMOX VIRTUAL ENVIRONMENT . Jurnal ini ditulis oleh Evan Prima Prasetyo, Joseph Dedy Irawan, dan F. X. Ariwibisono pada tahun 2022. Artikel ini menjelaskan pentingnya monitoring untuk memastikan server dapat beroperasi dengan optimal. 25 Proses monitoring mencakup pemantauan perangkat keras dan perangkat lunak yang digunakan pada server. Informasi yang diambil dari server diolah menjadi data pemantauan yang ditampilkan untuk menunjukkan status kondisi server. Jika terdapat hasil pemantauan yang abnormal, seperti overheating atau kegagalan aplikasi server, administrator sistem dapat segera melakukan perbaikan yang diperlukan. Administrator sistem juga dapat melakukan perawatan server secara efisien berdasarkan data pemantauan yang terkumpul. Pembuatan aplikasi monitoring server virtual berbasis web menggunakan skrip pemantauan memungkinkan pengumpulan data pemantauan dari server virtual dan penampilannya melalui antarmuka web. Hasil akhir dari aplikasi ini adalah kemampuannya untuk menampilkan data pemantauan perangkat keras dan perangkat lunak server virtual, pemantauan aplikasi server, grafik penggunaan sumber daya, fitur notifikasi peringatan, dan fitur pengendalian server virtual berbasis SSH yang dapat diakses melalui browser web. Selama server web Apache2 beroperasi, aplikasi pemantauan berhasil melakukan pemantauan, menjalankan fitur notifikasi peringatan, dan mengendalikan server dengan tingkat keberhasilan 100%. Namun, akses ke aplikasi pemantauan tidak mungkin jika server web Apache2 dimatikan. 7 31 73

4. Tinjauan studi pada jurnal yang berjudul 5 “APLIKASI MONITORING DAN CONTROLLING SERVER MENGGUNAKAN PROTOCOL ICMP ( INTERNET CONTROL MESSAGE PROTOCOL ) DAN SSH (SECURE SHELL) BERBASIS WEBSITE

7 31 73 . Jurnal ini ditulis oleh Redo Yanuar Pratama, Mira Orisa, dan FX Ariwibisono pada tahun 2020. Artikel ini menjelaskan bahwa sistem pemantauan server sangat penting bagi seorang administrator. Menjamin setiap server dalam kondisi optimal adalah tugas utama administrator, mengingat server harus selalu online atau aktif setiap

saat. Metode pemantauan yang umum digunakan saat ini melibatkan pemeriksaan individual setiap server oleh administrator. Dengan pendekatan ini, administrator dapat mengidentifikasi kondisi fisik dan status server, baik dalam keadaan aktif (up) maupun non-aktif (down). 7 Penggunaan protokol ICMP (Internet Control Message Protocol) memungkinkan pemantauan server secara real-time, sementara protokol SSH (Secure Shell) 13 digunakan untuk mengendalikan server dan mentransfer file ke server. 7 31 Aplikasi pemantauan dan kontrol server ini telah diimplementasikan dan diuji coba di jaringan lokal Laboratorium Jaringan Komputer Teknik Informatika S-1 ITN Malang. Melalui aplikasi ini, administrator dapat memonitor kondisi server secara real-time, baik saat server aktif maupun tidak aktif. Jika server tidak aktif, administrator akan menerima notifikasi. Fitur kontrol server dalam aplikasi ini mencakup kemampuan untuk mematikan atau me-restart server, mengelola layanan yang berjalan di server, serta mengirimkan file ke server.

2 5. Tinjauan studi pada jurnal yang berjudul 6 “ MONITORING APLIKAS I MENGGUNAKAN DASHBOARD UNTUK SISTEM INFORMASI AKUNTANSI PEMBELIAN DAN PENJUALAN (STUDI KASUS : UD APUNG) 2 . Jurnal ini ditulis oleh Sufia

Maulida, Fikri Hamidy, dan Agung Deni Wahyudi pada tahun 2020. Artikel ini membahas masalah operasional yang dihadapi oleh UD Apung, sebuah perusahaan dagang yang fokus pada penjualan alat bangunan. 2 Proses manual dalam penjualan dan pembelian, terutama dalam pembuatan nota penjualan dan faktur pengiriman, menyebabkan keterlambatan pengolahan transaksi. Pencatatan manual pembelian barang dan keterbatasan dalam monitoring serta perekapan data menggunakan Microsoft Excel juga menjadi kendala, memperlambat pembuatan laporan. Meskipun telah menerapkan sistem informasi akuntansi, masih terdapat keterbatasan dalam pemrosesan data penjualan dan pembelian. Meskipun sudah mencoba menggunakan dashboard interaktif, masih ada ketergantungan pada Microsoft Excel. Solusi yang diusulkan adalah mengembangkan sistem informasi akuntansi yang lebih canggih, integratif, dan menyeluruh. Implementasi dashboard interaktif yang sepenuhnya terintegrasi dapat meningkatkan akses real-time, mempercepat pengambilan

keputusan, dan meningkatkan efektivitas dalam mengelola dan mengoptimalkan kinerja perusahaan. Dengan demikian, UD Apung dapat mengatasi kendala operasional dengan lebih efisien dan adaptif terhadap dinamika pasar. Kesimpulan dari artikel ini adalah pemantauan aplikasi melalui dashboard dilakukan dengan mengamati grafik transaksi penjualan dan pembelian barang. **2** Hal ini memungkinkan pimpinan untuk segera mengetahui total transaksi penjualan dan pembelian perusahaan dengan cepat. Selain itu, proses input data penjualan dan pembelian tidak lagi memerlukan pencatatan manual seperti sebelumnya. **2** Hal ini dapat mengurangi risiko kesalahan dalam memasukkan data penjualan dan pembelian, yang dapat menyebabkan kerugian perusahaan. **4 16 14 BAB**

### III METODOLOGI PENELITIAN 3.1 Objek Penelitian

Penelitian ini akan fokus pada analisis server dan aplikasi microservice yang ada di PT. XYZ, sebuah perusahaan teknologi informasi dan komunikasi yang didirikan pada tahun 2007 dan berlokasi di Jakarta Barat. Perusahaan tersebut memiliki beberapa produk yang salah satunya merupakan platform ERP. Platform ini diketahui mengadopsi konsep microservice, sebuah pendekatan arsitektur perangkat lunak yang memecah aplikasi menjadi komponen-komponen independen dan dapat berkembang secara mandiri. Setiap microservice biasanya memiliki servernya masing-masing sehingga terdapat banyak server yang digunakan oleh PT. XYZ untuk menjalankan platform tersebut.

### 3.2 Analisa Sistem Yang Berjalan 3.2 **20 29 51** 1 Metode Pengembangan Sistem

Metode yang digunakan oleh penulis dalam pengembangan sistem ini adalah model Waterfall. Berikut ini adalah penjelasan lebih rinci mengenai tahapan-tahapan dalam model Waterfall yang diterapkan dalam pengembangan aplikasi monitoring resource server dan microservice berbasis web dengan notifikasi Telegram di PT. XYZ:

1. Perencanaan Pada tahap ini, penulis melakukan identifikasi terhadap kebutuhan apa saja yang diperlukan untuk mengatasi masalah yang ada didalam pemantauan resource server dan status aplikasi microservice di PT. XYZ. Identifikasi tersebut membantu menentukan ruang lingkup masalah yang terjadi didalam proses monitoring.
2. Analisis Setelah semua kebutuhan didapat, langkah berikutnya adalah menganalisis

seluruh prosedur pemantauan resource server dan aplikasi microservice yang ada di PT. XYZ. Analisis tersebut meliputi analisis dokumen dan prosedur sistem yang berjalan. Hasil analisis tersebut digunakan untuk menentukan kebutuhan fungsional dan non-fungsional, sehingga dapat memudahkan penulis didalam pembuatan diagram UML, basis data, dan perancangan antarmuka untuk sistem monitoring yang akan dikembangkan.

### 4 3. Desain

Pada tahap desain, penulis membuat berbagai diagram UML seperti use case diagram berikut dengan spesifikasi use case , activity diagram, sequence diagram, dan class diagram berdasarkan analisis yang sudah dilakukan. Selain

itu penulis melakukan perancangan basis data yang mencakup tabel-tabel dan kolom-kolom yang diperlukan. Terakhir, penulis merancang antarmuka sistem monitoring yang dibuat untuk memastikan tampilan yang akan dibuat sesuai dengan kebutuhan.

4. Implementasi Setelah desain sudah di buat oleh penulis, tahap berikutnya adalah melakukan implementasi sistem monitoring , implementasi tersebut meliputi pengkodean untuk membuat sistem tersebut dan apabila sistem telah selesai dibuat dilanjutkan oleh pengujian sistem untuk memastikan kembali bahwa sistem yang dibuat sesuai dengan kebutuhan yang telah dirancang sebelumnya.

3.2.2 Metode Pengumpulan Data Metode yang akan diterapkan dalam penelitian ini adalah metode Kualitatif Deskriptif, yang didasarkan pada pendekatan studi kasus yang memerlukan pengumpulan data untuk menghasilkan informasi yang relevan dengan tujuan penelitian, seperti menggambarkan sistem informasi pemantauan yang tepat. Berikut ini adalah rincian lebih lanjut:

1. Observasi Observasi dilakukan dengan mengamati proses atau sistem yang sudah dijalankan di dalam perusahaan tersebut terkait monitoring . Dari hasil pengamatan tersebut yang akan menjadi dasar dalam pembuatan analisis bisnis proses yang nanti akan di tuangkan kedalam diagram UML dan perancangan sistem yang akan dibangun.

Tabel 3. 1. Hasil Observasi

Tempat yang diamati Lantai 4 – PT. XYZ Petugas yang diamati Staf f infrastruktur Muhammad Lutfi Alfandi Waktu Pengamatan 15 2. Wawancara Metode wawancara dilakukan dengan para pelaku yang terlibat didalam

aktivitas monitoring tersebut. Wawancara bertujuan untuk mengetahui kebutuhan dan permasalahan yang ada didalam sistem monitoring server yang ada di perusahaan tersebut. Tabel 3. 2 Hasil Wawancara Muhammad Lutfi Alfandi Tanggal Wawancara Lokasi Wawancara Narasumber Staff infrastruktur 16 Catatan Hasil Pengamatan Terdapat 2 orang Staff infrastruktur yang ada di PT. XYZ. Staff melakukan monitoring satu kali dalam seminggu. Staff infrastruktur melakukan pemantauan dengan cara masuk ke dalam sistem operasi server melalui Staff memasukan perintah untuk menampilkan beberapa data resource server dan melihat status microservice . Staff melakukan pencatatan terhadap resource dari server tersebut, data resource yang di ambil Memory , Storage dan CPU serta status microservice yang berjalan. Staff melakukan ketiga hal diatas secara berulang ke setiap server yang ada di PT. XYZ. Staff melakukan tindakan lebih lanjut apabila terjadi penggunaan resource yang melebihi batas dan matinya aplikasi microservice dari hasil monitoring. Staff mendapatkan informasi terkait aplikasi microservice mati dari tim helpdesk . No Hal yang Ditanyakan Jawaban Interviewee Seperti apa metode monitoring yang dilakukan saat ini?Metode yang dilakukan saat ini adalah dengan cara masuk ke dalam server melalui SSH. Selanjutnya memasukan perintah untuk menampilkan data resource dibutuhkan dan status aplikasi microservice , setelah data muncul Staff infrastruktur akan melakukan pencatatan terhadap data resource dan status aplikasi tersebut. Proses tersebut dilakukan berulang ke semua server yang ada di Data apa saja dari resource server yang diperlukan atau di resource yang selalu di lakukan pengecekan yaitu, Storage dan CPU. monitoring dilakukan oleh tim? dilakukan sekali dalam seminggu. Tepatnya di penghujung akhir hari kerja yaitu hari jumat. Apa keterbatasan utama yang dihadapi oleh Staff infrastruktur dalam proses monitoring yang berjalan saat ini? Keterbatasan utama dalam metode monitoring dilakukan saat ini yaitu tidak dapat responsif terhadap perubahan mendadak atau keadaan darurat seperti traffic tinggi secara tiba-tiba yang dapat

menyebabkan matinya aplikasi microservice metode yang sekarang membutuhkan waktu yang lama dikarenakan PT. XYZ memiliki banyak server yang harus di monitoring . Jika aplikasi microservice mati diluar jadwal bagaimana caranya Staff infrastruktur mengetahuinya? Staff infrastruktur mengetahuinya melalui tim heldesk yang mendapatkan keluhan dari client mengenai aplikasi tidak berjalan dengan baik, yang kemudian informasi tersebut diteruskan kepada Staff infrastruktur. Apa yang menjadi kebutuhan utama dalam proses yang sesuai bagi PT. XYZ? Kebutuhan utama dalam proses monitoring yang efektif bagi PT. XYZ adalah adopsi sistem monitoring yang otomatis time yang mampu memberikan visibilitas mendalam terhadap penggunaan resource setiap server dan microservice , serta mampu memberikan peringatan ketika terdapat server yang terindikasi mengalami situasi

### 3.2.3 Analisa Dokumen Berdasarkan dari hasil pengumpulan data menggunakan metode observasi dan wawancara tersebut yang dilaksanakan dengan Staff infrastruktur. Hasil dari monitoring server yang dilakukan oleh tim infratraktur berupa catatan yang berisi data dari server tersebut. Berikut merupakan dokumen yang digunakan oleh Staff infrastruktur :

Tabel 3. 3. Analisa Dokumen Laporan Hasil Monitoring Untuk mencatat hasil monitoring ke beberapa server yang dilakukan oleh Staff Laporan di buat setiap minggu. File Elektronik ( File Excel ) Hasil dari catatan tersebut kemudian diberikan kepada kepala departemen IT yang ada di PT. XYZ yang kemudian catatan tersebut akan di jadikan bahan laporan dan diskusi apakah server tersebut perlu dilakukan tindakan seperti melakukan upgrade resource atau mengaktifkan kembali aplikasi microservice yang terindikasi tidak berjalan atau mati.

### 3.2.4 Analisa Proses Bisnis Analisa proses bisnis yang berjalan berdasarkan dari hasil observasi dan wawancara terhadap Staff infrastruktur mengenai prosedur monitoring yang dilakukan selama ini oleh tim dapat digambarkan menggunakan suatu Activity Diagram proses monitoring resource server dan aplikasi microservice , sebagai berikut :

Berdasarkan Activity Diagram diatas pada prosedur monitoring PT. XYZ, dapat dijelaskan sebagai

berikut : 1. Staff infrastruktur akan melakukan request masuk ke dalam UI server yang akan dimonitoring melalui SSH dengan menggunakan username yang terdaftar. 2. Server akan meminta untuk memasukan password berdasarkan username yang dimasukan. 3. Staff infrastruktur memasukan password untuk masuk ke dalam halaman server. 4. Validasi password , jika password yang dimasukan benar maka request masuk berhasil dan Staff infrastruktur dapat mengakses halaman dari server tersebut. 5. Staff infrastruktur memasukan perintah untuk mengakses data resource dari server tersebut. Data resource berupa CPU , Memory dan Storage , serta status aplikasi microservice yang ada didalamnya. 6. Server akan memproses permintaan dari Staff infrastruktur dengan memberikan data sesuai perintah yang dimasukan oleh Staff infrastruktur yaitu berupa data resource dan status aplikasi microservice . 7. Jika resource server dan status aplikasi microservice yang di periksa tidak ditemukan ada masalah maka proses monitoring selesai dan Staff infrastruktur dapat melanjutkan monitoring ke server selanjutnya. 8. Jika server terindikasi ada masalah maka Staff infrastruktur akan melakukan tindakan untuk mengatasi masalah tersebut. Diagram aktivitas diatas merupakan aktivitas yang dilakukan oleh Staff infrastruktur untuk melakukan monitoring terhadap satu server. Aktivitas diatas dilakukan berulang sebanyak jumlah server yang ingin dilakukan monitoring di PT. XYZ.

### 3.2.3 Analisa Permasalahan

1. Permasalahan yang Di Hadapi Permasalahan yang dialami oleh PT. XYZ didalam prosedur monitoring server yang telah mereka jalankan yaitu : 1) Dalam proses monitoring saat ini, Staff infrastruktur diharuskan melakukan pengecekan satu per satu pada setiap server yang dimiliki oleh PT. XYZ. Staff infrastruktur diharuskan melakukan aktivitas yang sama secara berulang 17 Berapa interval waktu penarikan data resource server yang diharapkan oleh tim ? Kami berharap dapat real-time namun jika tidak dapat tidak masalah, mungkin bisa 5 menit sekali paling untuk melakukan monitoring sebanyak jumlah server yang ada di PT. XYZ. Metode tersebut sudah dapat memberikan



visibilitas mendalam terhadap masing-masing data server yang dibutuhkan, Namun dapat memakan waktu dan menjadi tidak praktis karena PT. XYZ memiliki sejumlah server yang cukup banyak. 2) Proses monitoring yang dilakukan saat ini di PT. XYZ juga memiliki masalah lain yaitu ketika dihadapkan pada situasi yang memerlukan respon cepat. Metode monitoring saat ini yang dilakukan sesuai jadwal setiap minggunya ini kurang responsif terhadap perubahan mendadak atau keadaan darurat seperti lonjakan traffic tinggi yang menyebabkan matinya aplikasi microservice sehingga memerlukan pemantauan real-time dan notifikasi instan agar tim dapat mengetahui keadaan darurat tersebut dan kemudian dapat dilakukan sebuah tindakan. 2. Alternatif Pemecahan Masalah Peneliti akan membuat sebuah aplikasi monitoring yang nantinya aplikasi tersebut dapat menarik data resource dari setiap server berikut aplikasi microservice didalamnya yang dimiliki oleh PT. XYZ yang kemudian data tersebut akan di satukan didalam satu basis data dan kemudian akan di visualisasikan dalam bentuk dashboard, serta apabila ada resource penggunaan dari server yang melebihi batas wajar atau aplikasi microservice tidak berjalan maka sistem tersebut juga akan mengirimkan peringatan melalui telegram sehingga dapat memberitahu staff infrastruktur untuk dapat segera melakukan tindakan. 3.3 Analisa Kebutuhan Untuk memastikan keberhasilan suatu proyek, diperlukan kepastian bahwa sistem yang dibangun benar-benar memenuhi kebutuhan user. Untuk mencapai hal ini, melakukan analisis kebutuhan yang menyeluruh adalah langkah yang tak terhindarkan. Dalam proses ini, setiap aspek dari kebutuhan user harus dipelajari secara menyeluruh, dari kebutuhan fungsional hingga non-fungsional. Analisis kebutuhan yang menyeluruh memungkinkan identifikasi yang tepat terhadap apa yang diharapkan oleh user dari sistem yang dibangun. Dengan memahami dengan baik kebutuhan user, risiko kegagalan proyek dapat diminimalisir secara signifikan. Sebab, ketika setiap detail telah dipelajari dengan seksama, lebih mudah untuk menghindari kesalahan yang mungkin terjadi karena kekurangan informasi atau pemahaman yang kurang

tepat. Selain mengurangi risiko kegagalan, analisis kebutuhan yang menyeluruh juga memungkinkan penulis untuk mengoptimalkan sistem agar memberikan manfaat maksimal bagi user. Dengan memahami kebutuhan user dengan baik, penulis dapat merancang solusi yang sesuai dan efektif, sehingga meningkatkan kepuasan dan produktivitas user.

### 3.3.1 Analisa Kebutuhan User

Analisis kebutuhan user adalah proses mendalam untuk memahami kebutuhan, keinginan, dan preferensi pengguna yang berkaitan dengan sebuah sistem yang akan dibuat. Tujuan utamanya adalah untuk mengidentifikasi dan mendokumentasikan apa yang diharapkan oleh user dari sistem yang akan dibangun. Berikut ini merupakan daftar kebutuhan user yang diharapkan didalam sistem yang akan dibangun terdapat pada table dibawah ini. Tabel 3. 4. Kebutuhan User Sistem dapat menarik data resource seperti CPU, Memory dan Storage dan status aplikasi dari setiap server microservice. Sistem dapat menarik data secara real time / maksimal 5 menit sekali. Sistem dapat menampilkan data resource server dan status aplikasi microservice dashboard yang mudah dipahami. Sistem dapat menampilkan history dari data yang telah ditarik sebelumnya minimal 1 hari. Sistem dapat mengirimkan notifikasi terhadap data server yang telah melebihi batas wajar dan status aplikasi microservice yang mati.

### 3.3.2 Analisa Kebutuhan Sistem

Setelah memperoleh informasi dari kebutuhan pengguna, langkah berikutnya adalah mengidentifikasi kebutuhan sistem. Tahapan ini penting untuk memahami secara terperinci dan menyeluruh kebutuhan sistem yang akan dibangun, sehingga selama proses pengembangan tidak ada kekurangan informasi yang telah ditetapkan atau kekurangan pada tahap akhir elisitasi. Elisitasi merujuk pada proses mendefinisikan kebutuhan sistem berdasarkan analisis yang telah dilakukan sebelumnya, yaitu hasil dokumentasi dari pengumpulan data. Sistem dapat menampilkan lebih dari satu server. Berikut merupakan tahap elisitasi final yang sudah ditentukan terdapat pada table dibawah ini. Tabel 3. 5. Elisitasi Final

### 18 Elisitasi Tahap Akhir Functional

Terdapat data aktual resource dari setiap server history data resource dari setiap

server Terdapat data aktual status microservice dari setiap server history data status microservice dari setiap server Terdapat master data server beserta data microservice didalamnya Terdapat notifikasi Non Functional Data terbaru di dapatkan maksimal 5 menit sekali Sistem memiliki tampilan yang menarik dan mudah dipahami yang bisa dilihat minimal 1 hari sebelumnya Setelah elisitasi final ditentukan, langkah ini akan menjadi panduan utama dalam pelaksanaan serta analisis hasil penelitian. Elisitasi final berfungsi sebagai fondasi yang mengarahkan penulis didalam proses penelitian menuju tujuan yang telah ditetapkan. Dengan acuan ini, setiap tahap pelaksanaan dapat dijalankan dengan lebih terstruktur dan sistematis, memastikan bahwa analisis yang dilakukan nantinya sesuai dengan kerangka yang telah ditentukan.

72 19 BAB IV HASIL DAN ANALISIS PENELITIAN 4.1 Analisa Perancangan Sistem 4.1 1 Identifikasi Kebutuhan Sistem

Identifikasi kebutuhan sistem didapatkan berdasarkan hasil dari observasi, wawancara dan analisa dokumen yang telah dilakukan oleh penulis. Hasil Identifikasi tersebut berupa permasalahan yang terjadi didalam metode monitoring yang dilakukan oleh tim infrastruktur saat ini dan kendala apa saja yang dialami selama menggunakannya. Berikut merupakan uraian dari masalah yang terjadi didalam metode monitoring saat ini.

1. Dalam proses monitoring saat ini, staf infrastruktur diharuskan memeriksa setiap server milik PT. XYZ satu per satu. Staf tersebut harus melakukan aktivitas yang sama berulang kali untuk melakukan monitoring ke seluruh server yang dimiliki PT. XYZ. Meskipun metode ini dapat memberikan visibilitas mendalam terhadap masing-masing data server yang dibutuhkan, namun memakan waktu dan tidak praktis karena PT. XYZ memiliki banyak server.
2. Proses monitoring saat ini di PT. XYZ juga menghadapi masalah lain ketika diperlukan respons cepat. Metode monitoring yang dilakukan sesuai jadwal mingguan kurang responsif terhadap perubahan mendadak atau keadaan darurat, seperti lonjakan traffic tinggi yang menyebabkan aplikasi microservice mati. Oleh karena itu, dibutuhkan pemantauan real-time dan notifikasi instan agar

tim dapat mengetahui keadaan darurat tersebut dan segera mengambil tindakan. 4.1.2 Analisa Kebutuhan Sistem Setelah melakukan identifikasi kebutuhan sistem, selanjutnya analisa kebutuhan sistem ini dilakukan untuk meninjau ulang kebutuhan yang diperlukan untuk mengatasi masalah tersebut. Dalam konteks penelitian ini, penulis memaparkan solusi dari permasalahan yang teridentifikasi di PT. XYZ, yaitu melalui Rancang Bangun Aplikasi Monitoring Resource Server dan Microservice Berbasis Web dengan Notifikasi Telegram Menggunakan Pendekatan Waterfall. Sistem ini dirancang untuk mempermudah Staff infrastruktur dalam melakukan monitoring terhadap setiap server yang dimiliki oleh PT. XYZ. Berikut merupakan beberapa fitur yang dibutuhkan berdasarkan hasil identifikasi oleh penulis diantaranya sebagai berikut : 1. Terdapat fitur yang dapat menarik data resource dan status aplikasi microservice dari setiap server. Fitur tersebut akan dipenuhi oleh Aplikasi lapor yang merupakan aplikasi yang berjalan secara background menggunakan scheduler, aplikasi tersebut akan dapat mengirimkan data server berikut dengan resource server dan status aplikasi microservice sesuai dengan interval waktu yang telah ditetapkan. 2. Staff infrastruktur dapat melakukan monitoring terhadap beberapa server didalam satu tampilan. Dengan adanya fitur tersebut Staff infrastruktur tidak perlu melakukan aktivitas berulang untuk memeriksa setiap server yang dimiliki oleh PT. XYZ. 3. Staff infrastruktur dapat melakukan monitoring terhadap history penggunaan resource dan status aplikasi microservice dari masing masing server supaya Staff infrastruktur dapat mengetahui di jam berapa aktivitas server sedang tinggi. 4. Staff infrastruktur dapat mengelola aplikasi lapor yang di install pada server yang dilakukan monitoring. 5. Staff infrastruktur mendapatkan notifikasi melalui telegram apabila resource server melebihi batas wajar dan status aplikasi microservice mati supaya tim dapat melakukan tindakan dengan cepat.

37 4.2

Perancangan Diagram Sistem Usulan Dalam merancang dan membangun diagram untuk aplikasi monitoring yang diusulkan, penulis menggunakan pendekatan Unified Modeling Language (UML). Pendekatan ini melibatkan penerapan berbagai

jenis diagram UML yang masing-masing memiliki peran penting dalam memodelkan sistem secara komprehensif. Penulis menggunakan use case diagram, activity diagram, sequence diagram dan class diagram, selain UML didalam perancangan ini penulis menggunakan DFD dan ERD sebagai diagram pendukung didalam merancang aplikasi monitoring. Dengan mengintegrasikan berbagai jenis diagram UML disertai DFD berikut dengan ERD, penulis dapat menciptakan gambaran yang jelas dan terstruktur tentang bagaimana aplikasi monitoring akan dibangun dan berfungsi sesuai dengan kebutuhan.

#### 4.2.1 Use Case Diagram

Dalam proses perancangan use case diagram untuk aplikasi monitoring yang diusulkan, penulis mengidentifikasi dua aktor utama yang berinteraksi dengan sistem. Aktor pertama adalah Staff infrastruktur, yang berperan sebagai pengguna utama aplikasi ini. Aktor kedua adalah aplikasi lapor, yang berfungsi untuk mengirimkan data resource dan status microservice ke aplikasi monitoring. Berikut merupakan rancangan use case diagram untuk aplikasi monitoring di PT. XYZ. Gambar 4.1 merupakan rancangan use case diagram untuk aplikasi monitoring. Selanjutnya penulis akan mendeskripsikan setiap use case dengan spesifikasi use case. Berikut merupakan spesifikasi use

case dari setiap use case tersebut : 20

#### 1. Use Case Description

Login

Use Case Name :
Description: Use case "Login" merupakan proses untuk memastikan bahwa hanya pengguna yang didaftarkan yang dapat mengakses aplikasi monitoring dengan melakukan login.
Pre-conditions: Pengguna harus sudah terdaftar dalam sistem dengan username dan password yang valid.
Post-conditions: Sistem akan menampilkan halaman dashboard monitoring.
Normal Flow: Sistem akan menampilkan halaman login yang terdapat form berisi username dan password. User diminta memasukan username dan password dan klik login.
36 Staff infrastruktur Ketika user ingin mengakses menu yang ada didalam aplikasi monitoring, sebelum bisa masuk akan di arahkan ke halaman login untuk melakukan login terlebih dahulu.
69 Sistem akan melakukan validasi terhadap username dan password yang di masukan.
71 Jika valid maka user

Description: Use case "Login" merupakan proses untuk memastikan bahwa

a hanya pengguna yang didaftarkan yang dapat mengakses aplikasi

monitoring dengan melakukan login. 36 Staff infrastruktur Ketika user ingin

mengakses menu yang ada didalam aplikasi monitoring, sebelum bisa masuk

akan di arahkan ke halaman login untuk melakukan login terlebih dahulu.

Pre-conditions: Pengguna harus sudah terdaftar dalam sistem dengan

username dan password yang valid. Post-conditions: Sistem akan menampilkan

halaman dashboard monitoring. Normal Flow: Sistem akan menampilkan halaman

login yang terdapat form berisi username dan password. User diminta memasukan

username dan password dan klik login. 69 Sistem akan melakukan validasi

terhadap username dan password yang di masukan. 71 Jika valid maka user

berhasil masuk ke halaman dashboard monitoring. Exceptions Flow : Menampilkan notifikasi "Username atau Password Invalid" ketika proses login ke sistem gagal. Tabel diatas merupakan spesifikasi use case Login yang dapat dilakukan oleh Staff infrastruktur. Use case Login diawali dari user ketika ingin mengakses aplikasi monitoring yang kemudian sistem menampilkan halaman login dan diakhiri dengan user yang berhasil masuk ke halaman login jika username dan password yang dimasukan valid dengan exception flow menampilkan notifikasi apabila proses login gagal.

2. Use Case Description Monitoring Server Tabel 4. 2. Use Case Description Monitoring Server Use Case Name : Monitoring Server Description: Use case "Monitoring Server" memungkinkan Staff infrastruktur untuk memantau Resource server dan status aplikasi microservice dari berbagai server. Staff infrastruktur. User menekan button menu Realtime Monitoring pada sidebar dashboard monitoring. **10** Pre-conditions: User telah berhasil login dengan username dan password yang valid. Post-conditions: Sistem akan menampilkan halaman realtime monitoring.

2.1 Normal Flow: User menekan tombol button menu Realtime Monitoring pada sidebar dashboard monitoring. Sistem menampilkan halaman realtime monitoring dengan data resource dan status aplikasi microservice dari berbagai server. Exceptions Flow :

Tabel diatas merupakan spesifikasi use case Monitoring Server yang dapat dilakukan oleh Staff infrastruktur. Use case Monitoring Server diawali dari user ketika menekan tombol menu Realtime Monitoring pada sidebar dashboard monitoring yang kemudian diakhiri dengan sistem menampilkan halaman realtime monitoring .

3. Use Case Description Monitoring History Server Tabel 4. 3. Use Case Description Monitoring History Server Use Case Name : History Monitoring Description: Use case "History Monitoring" memungkinkan Staff infrastruktur untuk melihat data history Resource server dan status aplikasi microservice dari server yang dipilih, data history yang ditampilkan yaitu 24 jam Staff infrastruktur. User menekan button menu History Monitoring pada sidebar dashboard monitoring. **10** Pre-conditions: User telah berhasil login dengan username dan password yang valid. Post-conditions:

Sistem akan menampilkan halaman History monitoring. Normal Flow: User menekan tombol button menu History Monitoring pada sidebar dashboard monitoring. Sistem menampilkan menu history monitoring dengan field dropdown yang terdapat list ip untuk dipilih oleh User memilih ip dari server yang ingin dilihat. Sistem akan menampilkan data resource server dan status aplikasi microservice dari ip yang dipilih. Exceptions Flow : Tabel diatas merupakan spesifikasi use case History Monitoring yang dapat dilakukan oleh Staff infrastruktur. Use case History Monitoring diawali dari user ketika menekan tombol menu History Monitoring pada sidebar dashboard monitoring yang kemudian diakhiri dengan sistem menampilkan data resource server dan status aplikasi microservice dari ip yang dipilih.

#### 4. Use Case Description Mengelola Aplikasi Laporan

Tabel 4. 4 . Use Case Description Mengelola Aplikasi Laporan

Use Case Name : Mengelola Aplikasi Laporan 22 Description: Use case "Mengelola Aplikasi Laporan memungkinkan Staff infrastruktur untuk mengelola data aplikasi laporan dengan mengedit data atau menghapus data. Data yang diubah selanjutnya akan di update juga ke aplikasi Staff infrastruktur. User menekan button menu Manajemen Server pada sidebar dashboard monitoring.

**10** Pre-conditions: User telah berhasil login dengan username dan password yang valid. Post-conditions: Sistem akan menampilkan halaman Manajemen server. Normal Flow: User menekan tombol button menu Manajemen Server pada sidebar dashboard monitoring. Sistem menampilkan menu Manajemen Server. User dapat mencari data server yang terdapat aplikasi laporan untuk dilihat dengan mengetik ip atau nama server yang di cari di kolom pencarian. User dapat mengelola data server seperti : Mengedit data server, selanjutnya ke subflow MS-1 Edit Data Server. Menghapus data server, selanjutnya ke subflow MS-2 Delete Data Server. MS-1 Edit Data Server User berada di halaman manajemen server. User menekan button edit di data server yang diinginkan. Sistem akan menampilkan halaman untuk mengedit data. User melakukan edit data dan melakukan submit.

**43** Sistem akan menampilkan pesan

**12** “apakah anda yakin ingin melakukan perubahan data . Sistem akan menyimpan

data yang diubah dan melakukan update ke aplikasi lapor yang ada di server yang di Sistem akan menampilkan pesan "Data berhasil diubah . MS-2 Delete Data Server User berada di halaman manajemen server. User menekan button delete di data server yang diinginkan. 43 Sistem akan menampilkan pesan 12 "apakah anda yakin ingin menghapus data 29 49 . Sistem akan menghapus data dan melakukan update ke aplikasi lapor yang ada di server yang dihapus Sistem akan menampilkan pesan 14 "Data berhasil di hapus . Exceptions Flow : Tabel diatas merupakan spesifikasi use case Mengelola Aplikasi Lapor yang dapat dilakukan oleh Staff infrastruktur. Use case Mengelola Aplikasi Lapor diawali dari user ketika menekan tombol button menu Manajemen Server pada sidebar dashboard monitoring yang kemudian diakhiri dengan user yang dapat mengelola data server. 5. Use Case Description Update Data Server Tabel 4. 5 . Use Case Description Update Data Server Use Case Name : Update Data Server Description: Use case "Update Data Server memungkinkan aplikasi lapor yang dipasang di server yang dimonitoring untuk mengirimkan data identitas server tersebut. 23 Aplikasi Lapor. Aplikasi lapor di jalankan di server yang dimonitoring. Pre-conditions: Aplikasi lapor telah diunggah ke server yang dimonitoring. Post-conditions: Sistem akan menyimpan data indentitas server. Normal Flow: Aplikas lapor dijalankan di server yang dimonitoring. Aplikasi lapor mengirimkan data identitas server ke aplikasi monitoring. Exceptions Flow : Tabel diatas merupakan spesifikasi use case Update Data Server yang dapat dilakukan oleh aplikasi lapor. Use case Update Data Server diawali dari aplikasi lapor yang dijalankan di server yang akan dimonitoring yang kemudian diakhiri dengan aplikasi lapor mengirimkan data ke aplikasi monitoring . 6. Use Case Description Mengirimkan Data Resource Server dan Status Microservice Tabel 4. 6. Use Case Description Kirim Data Use Case Name : Mengirimkan Data Resource Server dan Status Microservice Description: Mengirimkan Data Resource Server dan Status Microservic " memungkinkan aplikasi lapor yang dipasang di server yang dimonitoring untuk mengirimkan data resource server dan










status aplikasi microservice ke aplikasi monitoring. Data yang dikirimkan akan dilakukan pengecekan, apabila data yang dikirimkan tidak lolos dalam pengecekan maka sistem akan mengirimkan notifikasi dan selanjutnya data akan di simpan. Aplikasi Lapor dan Sistem Monitoring Scheduler yang berjalan setiap 5 menit. Pre-conditions: Aplikasi lapor berjalan dan aktif di server yang dilakukan monitoring. Post-conditions: Sistem akan melakukan pengecekan terhadap data yang dikirim. Sistem akan mengirimkan notifikasi apabila data yang dikirim. Sistem akan menyimpan data resource dan status aplikasi microservice. Normal Flow: Scheduler di aplikasi lapor aktif. Aplikasi Lapor mengirimkan data resource server dan status aplikasi microservice ke sistem monitoring. Sistem monitoring melakukan pengecekan data yang diterima. Sistem monitoring menyimpan data resource dan status aplikasi microservice yang dikirimkan oleh aplikasi


Exceptions Flow : Apabila data yang diterima tidak lolos dalam pengecekan maka sistem akan mengirimkan notifikasi ke telegram sesuai dengan data yang tidak lolos dalam pengecekan. 4.2.2 Activity Diagram






24 Activity diagram merupakan alat yang penting dalam perancangan dan pengelolaan aplikasi monitoring , karena membantu mengoptimalkan alur kerja dan meningkatkan koordinasi serta efisiensi dalam penyelesaian proses bisnis. Berikut merupakan activity diagram yang dirancang sesuai dengan use case yang telah dibuat sebelumnya. 1. Activity Diagram

Login Berdasarkan Activity Diagram Login diatas User memulai proses

dengan masuk ke web dashboard monitoring .   S istem kemudian menampilkan

halaman login yang meminta user untuk memasukkan username dan password .     

User kemudian memasukkan username dan password tersebut dan menekan tombol login . 

 Sistem melakukan validasi terhadap username dan password yang dimasukkan.    

Jika validasi gagal, sistem akan kembali meminta user untuk

memasukkan username dan password di halaman login . Namun, jika validasi

berhasil, sistem akan menampilkan dashboard kepada user . 2. Activity

Diagram Monitoring Server Berdasarkan Activity Diagram Monitoring Server

didasar Activity Diagram Monitoring Server diatas dapat dijelaskan bahwa User memulai dengan memilih menu realtime

monitoring pada sidebar dashboard monitoring . Setelah itu, sistem akan menampilkan halaman realtime monitoring . Pada halaman tersebut, sistem menyajikan data mengenai resource server dan status aplikasi microservice dari berbagai server, memberikan user informasi terkini dan detail tentang kondisi dan performa server serta aplikasi microservice yang dimonitoring .

3. Activity Diagram Monitoring History Server Berdasarkan Activity Diagram Monitoring History Server diatas dapat dijelaskan bahwa User mengakses dashboard monitoring dan memilih menu History Monitoring " pada sidebar dashboard monitoring . Setelah memilih menu tersebut, sistem langsung mengarahkannya ke halaman History Monitoring " . Di halaman " History Monitoring " , user disajikan dengan field dropdown yang menampilkan daftar IP server yang tersedia. Field dropdown ini memungkinkan user untuk memilih IP server yang ingin di lakukan monitoring oleh user . Sistem meminta user untuk memilih IP server yang diinginkan dari daftar yang tersedia. User kemudian memilih IP server yang ingin dilakukan pengecekan. Setelah user memilih IP server, sistem kemudian menampilkan data yang diminta berdasarkan IP server yang dipilih. Halaman History Monitoring " selanjutnya menampilkan informasi terdahulu tentang resource server dan status aplikasi microservice yang terkait dengan IP server yang dipilih oleh user. Dengan informasi ini, user dapat memantau history dari kinerja server dan status aplikasi yang terkait dengan IP yang dipilihnya.

4. Activity Diagram Update Data Server Berdasarkan Activity Diagram Update Data Server diatas dapat dijelaskan bahwa Ketika Aplikasi lapor aktif beroperasi di server yang sedang dilakukan monitoring , Aplikasi lapor akan menyiapkan data identitas dari server yang sedang dilakukan monitoring . Kemudian aplikasi lapor mengirimkan data terkait server yang dipantau ke dalam sistem dashboard monitoring . Selanjutnya data tersebut diterima dan disimpan ke dalam database .

5. Activity Diagram Mengirimkan Data Resource dan Status Microservice Berdasarkan Activity Diagram Mengirimkan Data Resource dan Status Microservice diatas dapat dijelaskan bahwa Ketika Scheduler dari

aplikasi lapor aktif. Aplikasi lapor akan mengecek status yang ada di aplikasi tersebut. Ketika status aktif "Yes" terverifikasi, aplikasi lapor akan mengekstrak data yang berupa informasi mengenai resource server dan status aplikasi microservice dari server yang dilakukan monitoring. Data yang telah terkumpul tersebut kemudian dikirimkan ke sistem. Setelah menerima data dari aplikasi lapor, sistem melakukan validasi untuk memastikan bahwa data resource server tidak melebihi batas yang telah ditentukan. Jika melebihi, sistem segera mengirimkan notifikasi melalui Telegram berdasarkan data yang melampaui batas tersebut, dan menyimpan informasi terkait ke dalam database. Namun, jika data dalam batas wajar, sistem akan melanjutkan dengan memeriksa status aplikasi microservice. Jika ada yang mati, sistem akan memberikan notifikasi melalui Telegram dan juga menyimpan data ke dalam database. Namun, jika tidak ada masalah, sistem akan langsung menyimpan data yang diterima ke dalam database dan memastikan integritas data yang tersimpan.

#### 6. Activity Diagram Mengelola Aplikasi Lapor Berdasarkan Activity Diagram Mengelola Aplikasi Lapor diatas dapat dijelaskan bahwa

User memulai dengan memilih opsi "Manajemen Serve" dari menu sidebar pada dashboard monitoring. Sistem kemudian menampilkan halaman "History Monitoring" yang berisi daftar data server yang di monitoring. Setelah menelusuri data server, user memilih satu data yang ingin diedit atau dihapus, kemudian menekan tombol action yang diinginkan. Berikut merupakan activity diagram lanjutan yang merupakan detail dari proses edit atau hapus data didalam activity diagram mengelola aplikasi lapor.

#### 1) Edit Data Server

Setelah user memilih data server dan menekan tombol "Edit", sistem akan mengambil data yang dipilih dan membuka sebuah modal edit. Modal tersebut muncul dengan field - field yang sudah terisi oleh data yang dipilih sebelumnya, memungkinkan user untuk mengedit informasi yang diinginkan. Setelah user menyelesaikan pengeditan, user kemudian menekan tombol "Save" untuk menyimpan perubahan tersebut. Sistem akan menampilkan konfirmasi untuk memastikan bahwa user ingin menyimpan perubahan tersebut.

Jika user menyetujui, sistem melanjutkan dengan memperbarui data di database sesuai dengan informasi yang telah dirubah. Selanjutnya, sistem melakukan update data ke aplikasi lapor dengan data yang telah diperbarui. Aplikasi lapor menerima pembaruan data tersebut dan mengintegrasikannya. 2) Delete Data Server Selain itu setelah user memilih data server dan menekan tombol Delete ", sistem memberikan konfirmasi terlebih dahulu untuk memastikan bahwa user benar-benar ingin menghapus data tersebut. Jika user menyetujui penghapusan, sistem melanjutkan dengan memperbarui status data di database bahwa data tersebut dihapus oleh user . Kemudian, seperti sebelumnya, sistem mengirimkan pembaruan data ke aplikasi lapor. Kemudian aplikasi lapor menerima dan memperbaharui data sesuai dengan perubahan yang dilakukan oleh user . Jika user memilih untuk tidak menghapus, sistem kembali menampilkan dashboard

Manajemen Serve " untuk user . 4.2.3 Sequence Diagram Sequence diagram merupakan alat penting untuk mengoptimalkan alur komunikasi dan koordinasi yang terjadi didalam aplikasi monitoring . Berikut adalah sequence diagram yang dirancang berdasarkan use case yang telah dibuat.

1. Sequence Diagram Login Gambar 4.10 menggambarkan diagram urutan ( sequence diagram) untuk proses pengiriman login . Diagram ini melibatkan beberapa komponen utama, yaitu User , User Controller , Table User dan UI Dashboard .

10 Proses login dimulai ketika user memasukkan username dan password ke dalam antarmuka login (UI Login) . Setelah pengguna memasukkan kredensial, UI Login mengirimkan informasi tersebut ke User Controller untuk dilakukan verifikasi. User Controller menerima username dan password dari UI Login dan mengirimkan permintaan ke Table User untuk memeriksa apakah data user yang diberikan ada di dalam database . Table User kemudian mencari data user berdasarkan username dan password yang telah diberikan. Jika Tabel User tidak menemukan data user yang cocok, maka akan dikirimkan pesan ke User Controller bahwa data user tidak ditemukan. User Controller kemudian mengirimkan pesan ke UI Login untuk menampilkan pesan kesalahan kepada user yaitu memberi tahu bahwa

username atau password yang dimasukkan salah. Namun, jika Tabel User menemukan data user yang cocok, maka data pengguna tersebut dikirimkan kembali ke User Controller . Setelah menerima data user , User Controller mengirimkan sinyal ke UI Dashboard untuk mempersiapkan tampilan dashboard bagi user . UI Dashboard kemudian ditampilkan kepada user , sehingga user dapat mengakses berbagai fitur dan data yang tersedia dalam aplikasi monitoring .

2. Sequence Diagram Monitoring Server Gambar 4.11 menggambarkan diagram urutan ( sequence diagram) untuk proses monitoring server. Diagram ini melibatkan beberapa komponen utama, yaitu Dashboard Monitoring , Dashboard Controller , Tabel report server dan Menu Realtime Monitoring . Proses dimulai ketika user memilih menu " Realtime Monitoring " pada UI dashboard monitoring . Setelah user memilih menu realtime monitoring , permintaan tersebut dikirimkan dari ui dashboard monitoring ke Dashboard Controller . Dashboard Controller , yang bertanggung jawab memproses permintaan tersebut, kemudian mengirimkan permintaan user ke tabel report server untuk mengambil data yang diperlukan untuk ditampilkan yaitu berupa data resource server dan status aplikasi microservice dari berbagai server. Tabel report server menerima permintaan dari Dashboard Controller dan mulai mengambil data yang diminta dari database . Setelah proses pengambilan data selesai, tabel report server mengirimkan data yang telah diambil ke Dashboard Controller . Dashboard Controller kemudian menerima data dari tabel report server dan mengirimkannya ke menu " Realtime Monitoring " untuk ditampilkan kepada user .

3. Sequence Diagram Monitoring History Server Gambar 4.12 menggambarkan diagram urutan ( sequence diagram) untuk proses monitoring history server. Diagram ini melibatkan beberapa komponen utama, yaitu Dashboard Monitoring , Dashboard Controller , Tabel Historing dan Menu History Monitoring . Proses dimulai ketika user memilih menu " History Monitoring " pada ui dashboard monitoring . Permintaan tersebut kemudian dikirimkan ke Dashboard Controller untuk diproses lebih lanjut. Dashboard Controller menerima permintaan tersebut dan mengirimkannya

ke Table Report Server untuk mengambil daftar IP yang tersedia. Tabel report server memproses permintaan tersebut dengan mengambil daftar IP dari database dan mengirimkannya kembali ke Dashboard Controller . Setelah menerima daftar IP dari tabel report server, Dashboard Controller mengirimkan daftar tersebut ke ui History Monitoring ". Menu history Monitoring kemudian menampilkan daftar IP kepada user . User kemudian memilih salah satu IP dari yang telah ditampilkan. Setelah user memilih IP, Dashboard Controller mengirimkan permintaan ke tabel report server untuk mengambil data history monitoring yang terkait dengan IP tersebut. 26 Tabel history menerima permintaan tersebut dan mengambil data history monitoring yang sesuai dengan ip tersebut dari database . Data history monitoring yang telah diambil kemudian dikirimkan kembali ke Dashboard Controller . Dashboard Controller menerima data history monitoring tersebut dan mengirimkannya ke menu History Monitoring ". Setelah itu, menu History Monitoring " menampilkan data history monitoring yang terkait dengan IP yang dipilih kepada user . 4.

Sequence Diagram Mengelola Aplikasi Laporan Gambar 4.13 menggambarkan diagram urutan ( sequence diagram) untuk sistem manajemen server yang terdiri dari beberapa komponen, termasuk pengguna ( User ), Dashboard Monitoring , Dashboard Controller , Tabel Laporan Identity, Menu Manajemen Server, dan Aplikasi Laporan. Diagram ini memvisualisasikan alur kerja dari tiga proses utama: melihat daftar server, merubah data server, dan menghapus data server. Pada proses pertama, user memilih menu manajemen server di Dashboard Monitoring . Dashboard Controller kemudian menampilkan daftar server dengan mengambil data dari tabel Laporan Identity dan mengirimkannya kembali ke Controller Dashboard Monitoring . Data tersebut kemudian akan dikirim ke menu Manajemen Server dan ditampilkan kepada user . Setelah data ditampilkan, user akan dapat melihat list dari server yang telah dipasangkan aplikasi laporan. Didalam menu tersebut user dapat memilih salah satu data dari list data server tersebut. Setelah memilih data, user dapat melakukan edit atau

delete terhadap data tersebut dengan menekan button edit atau delete yang ada. Ketika user melakukan salah satu dari kedua action tersebut sistem juga akan melakukan perubahan data ke aplikasi lapor di server yang dipilih, adapun detail proses tersebut dapat dijelaskan melalui sequence diagram dibawah ini adalah sebagai berikut: 1) Edit Untuk proses perubahan data server, user menekan tombol edit pada data server yang dipilih. Menu Manajemen Server kemudian meminta data server yang dipilih dari Tabel Lapor Identity dan menampilkan halaman modal untuk merubah data server tersebut. Setelah user merubah data dan menekan tombol simpan, sistem akan meminta konfirmasi dari user . Setelah konfirmasi diberikan, data server diperbarui di tabel Lapor Identity dan perubahan tersebut dikirim juga ke aplikasi lapor. Aplikasi Lapor menerima perubahan data tersebut dan melakukan perubahan juga ke tabel Lapor, Setelah itu aplikasi lapor mengkonfirmasi ke dashboard controller bahwa data telah di rubah. 2) Delete Pada proses penghapusan data server, pengguna menekan tombol delete pada data server yang dipilih. Menu Manajemen Server memproses permintaan penghapusan dengan meminta konfirmasi dari pengguna. Setelah pengguna memberikan konfirmasi, data server tersebut di perbarui ke tabel Lapor Identity untuk dirubah statusnya bahwa data server tersebut telah dihapus dan perubahan ini dikirim juga ke aplikasi lapor. Aplikasi lapor pun menerima perubahan tersebut dan melakukan perubahan di tabel Lapor, Setelah itu aplikasi lapor mengkonfirmasi ke dashboard controller bahwa data telah di rubah. 5. Sequence Diagram Update Data Server Gambar 4.12 menunjukkan diagram urutan ( sequence diagram) untuk proses pelaporan identitas server melalui Aplikasi Lapor. Diagram ini melibatkan beberapa komponen utama: Aplikasi Lapor, Controller Aplikasi Lapor, Controller Dashboard Monitoring , dan Tabel Lapor Identity . Proses dimulai ketika Aplikasi Lapor dijalankan di server yang akan dilakukan monitoring yang kemudian melakukan trigger ke Controller Aplikasi Lapor agar dapat mengirimkan data identitas server ke Controller Dashboard

Monitoring . Setelah menyiapkan data yang akan dikirim, Controller Aplikasi Laporan meneruskan data tersebut ke Controller Dashboard Monitoring . Kemudian Dashboard Monitoring menerima data tersebut dan menyimpannya ke tabel lapor identity , setelah data berhasil disimpan tabel lapor identity akan memberikan konfirmasi data berhasil disimpan sebagai response untuk Controller Dashboard Monitoring . Terakhir Controller Dashboard Monitoring menyampaikan pesan data berhasil disimpan ke Controller Aplikasi Laporan dan proses selesai.

### 6. Sequence Diagram Mengirimkan Data Resource dan Status Microservice

Gambar 4.13 menggambarkan diagram urutan ( sequence diagram) untuk proses pengiriman data dan pemantauan status dalam Aplikasi Laporan. Diagram ini melibatkan beberapa komponen utama, yaitu Aplikasi Laporan, Controller Aplikasi Laporan, Service Aplikasi Laporan, Controller Dashboard , Service Dashboard , Telegram, dan Table Report Server. Proses dimulai dengan Scheduler Kirim Data Aktif di Aplikasi Laporan, yang memicu pengiriman data. Controller Aplikasi Laporan kemudian memeriksa status sistem dengan meminta data status dari Service Aplikasi Laporan yang berasal dari tabel lapor. Jika status dikonfirmasi aktif oleh controller aplikasi lapor, controller aplikasi lapor meminta data resource dan status aplikasi lapor ke service aplikasi lapor dan kemudian dikirim ke Controller Dashboard . Setelah Controller Dashboard menerima data, dilakukan validasi terhadap dua kondisi utama: pertama, jika penggunaan sumber daya ( resource ) lebih dari 90%, maka sistem akan mengirim notifikasi melalui Telegram; kedua, jika terdapat aplikasi microservice yang mati, notifikasi juga akan dikirim melalui Telegram mengenai aplikasi yang mati tersebut. Setelah validasi dan pengiriman notifikasi, data disimpan ke dalam Tabel Report Server & history , dan konfirmasi penyimpanan data dikirim kembali ke Service Aplikasi Laporan.

27 Jika status dikonfirmasi tidak aktif oleh controller aplikasi lapor, Controller Aplikasi Laporan akan mengecek apabila status tidak aktif selama lebih dari 26 jam maka service aplikasi lapor akan melakukan shutdown pada Aplikasi Laporan.



4.2.4 Class Diagram Berikut merupakan rancangan class diagram yang dibuat berdasarkan tabel database dari hasil penggambaran sequence diagram sebelumnya yang telah dibuat. **9** Berdasarkan class diagram diatas dapat dirancang spesifikasi database yang sesuai dengan aplikasi monitoring .

1. Tabel Users Tabel 4. 7. Basisdata Tabel Users ID pengguna (primary key) Nama pengguna Kata sandi pengguna Tanggal dan waktu terakhir kali login Tabel 4.7 merupakan tabel yang berfungsi untuk menyimpan data yang digunakan untuk login kedalam aplikasi monitoring .

2. Tabel Identitas Lapor Tabel 4. 8. Basisdata Tabel Identitas Lapor ID (primary key) serverName Endpoint aplikasi lapor yang dijalankan di server yang Status aktif aplikasi lapor microservices url microservice yang ada di aplikasi lapor Tanggal data di update Visibilitas data (aktif / tidak aktif) Tabel 4.8 merupakan tabel yang berfungsi untuk menyimpan data identitas server yang mana server tersebut tempat di pasang nya aplikasi lapor yang digunakan untuk melaporkan data untuk dilakukan monitoring .

3. Tabel Report Server dan History Tabel 4. 9. Basisdata Tabel Report Server dan History ID (primary key) Nama pengguna Berisi data resource server dan status aplikasi microservice Tanggal & waktu data di terima Tabel 4.9 merupakan tabel yang berfungsi untuk menyimpan data aktual (untuk tabel report server) dan juga data history (untuk tabel history ) yang dikirimkan oleh aplikasi lapor yaitu berupa data resource server dan status 28 aplikasi microservice . Data tersebut berada didalam field content yang merupakan tipe data Array Object , Mengenai detail dari field content diatas adalah sebagai berikut.

Tabel 4. 10. Tabel Detail Content Tipe dari data yang dikirim (PERCENTAGE, STATUS) Nama data tersebut (STORAGE, MEMORY, CPU, MICROSERVICE) Angka Penggunaan resource usedPercentage Persentase dari penggunaan resource Angka space kosong dari resource freePercentage Persentase space kosong dari resource Berisi status microservice

Tabel 4.10 merupakan tabel detail dari field content yang ada pada tabel report server dan history . Data dari tabel tersebut

yang akan diolah oleh sistem untuk validasi ketika terdapat data resource yang melebihi batas yang di tentukan, maupun status microservice yang tidak berjalan di server yang mengirimkan data tersebut. Selain itu, data tersebut akan di olah juga oleh sistem agar dapat ditampilkan di halaman dashboard guna kepentingan monitoring

. 4. Tabel Lapor Tabel 4. 11. Tabel Lapor key ) Status yang menjadi acuan aktif atau tidaknya aplikasi lapor Tanggal & waktu data di ubah Tabel 4.11 merupakan tabel yang berfungsi untuk menyimpan data status aplikasi lapor sebagai acuan untuk setiap aksi yang dilakukan oleh aplikasi lapor didalam proses mengirimkan data monitoring . 4.2 **7** 5 Data

Flow Diagram Data Flow Diagram (DFD) memiliki beberapa tingkatan, yaitu diagram konteks (level 0), dan level 1. Berikut adalah DFD untuk aplikasi Monitoring dengan beberapa tingkatan tersebut. 1. DFD Level 0 Gambar 4.19 diatas merupakan data flow diagram level untuk aplikasi monitoring, diagram tersebut memiliki 2 entity yaitu Staff Infrastruktur dan Aplikasi Lapor, kemudian 1 proses yaitu Aplikasi Monitoring. Staff infrastruktur dapat melakukan monitoring dengan menerima data report server dan history report server, selain itu staff infrastruktur dapat mengelola data identitas dengan menerima data identitas dan melakukan perubahan data terhadap data identitas tersebut ke aplikasi lapor. Selain itu terdapat Aplikasi lapor yang melakukan pengiriman data report server dan juga data identitas ke aplikasi monitoring, kemudian aplikasi lapor dapat menerima data identitas apabila terdapat perubahan. 2. DFD Level 1 Gambar 4.20 diatas merupakan data flow diagram level 1 untuk aplikasi monitoring. Didalam diagram tersebut terdapat proses Kirim Data Server yang dilakukan oleh Aplikas Lapor, kemudian terdapat Manajemen Aplikasi Lapor yang dapat dilakukan oleh aplikasi lapor dan juga staff infrastruktur. Setelah itu terdapat History Monitoring dan Realtime Monitoring yang dapat dilakukan oleh Staff Infrastruktur. Mengenai detail dari proses DFD level 1 tersebut dapat dijelaskan pada tabel dibawah ini Tabel 4. 12 Detail Proses DFD Level 1 Aplikasi

Monitoring No Proses Input Output Deskripsi 29 1 Kirim Data Report Server-resourceServer -statusMicroservice Data report server dan data history report server Proses untuk mengirimkan data report server oleh aplikasi lapor yang kemudian data tersebut masuk ke tabel report server untuk dilakukan update dan masuk ke tabel history report server untuk disimpan

2 Manajemen Aplikasi Lapor Data Identitas Proses mengelola data identitas seperti melakukan melihat, update, delete dan bahkan menambahkan data oleh staff infrastruktur dan aplikasi lapor

3 History Monitoring -resourceServer -statusMicroservice Proses menampilkan data resource server dan status microservice dari report server untuk staff infrastruktur

4 Realtime Monitoring -resourceServer -statusMicroservice Proses menampilkan seluruh data resource server dan status microservice dari report history server untuk staff infrastruktur

#### 4.2.6 Entity Relationship Diagram

Gambar 4.20 diatas merupakan entity relationship diagram untuk aplikasi monitoring. Diagram tersebut mencakup entitas, atribut dan proses yang terjadi saat user (Staff infrastruktur) dan aplikasi lapor berinteraksi dengan aplikasi monitoring. Berdasarkan diagram diatas terdapat aplikasi lapor yang mengirimkan data report server, identitas lapor dan report history server ke aplikasi monitoring, Kemudian terdapat user (staff infrastruktur) yang dapat mengelola data identitas lapor didalam aplikasi monitoring.

#### 4.3 Perancangan Antar Muka Pengguna Didalam merancang aplikasi monitoring diperlukan perancangan antar muka ( User Interface ) pengguna yang dirancang sesuai dengan kebutuhan dengan fokus untuk meningkatkan kenyamanan pengguna didalam memakai aplikasi monitoring . Berikut merupakan rancangan antar muka pengguna dari aplikasi monitoring .

1. Antar Muka Login Pada gambar 4.19 menampilkan antarmuka login , Fungsi dari halaman ini adalah untuk memungkinkan pengguna memasukkan nama pengguna ( username ) dan kata sandi ( password ) mereka untuk mengakses aplikasi monitoring setelah menekan tombol " Sign In " .
2. Antar Muka Dashboard / Realtime Monitoring Pada gambar 4.20 menampilkan antar muka menu realtime monitoring , yang mana menu ini

memungkinkan pengguna untuk melihat data aktual mengenai data resource server dan status aplikasi microservice yang ada didalam server tersebut. 3. Antar Muka History Monitoring Pada gambar 4.21 menampilkan antar muka menu history monitoring , yang mana menu ini memungkinkan pengguna untuk melihat data mengenai resource server dan status aplikasi microservice dari waktu yang telah lalu atau lebih tepatnya 24 jam terakhir. Untuk menampilkan data tersebut pengguna perlu untuk memilih IP yang diinginkan untuk menampilkan data tersebut. 4. Antar Muka Manajemen Server Pada gambar 4.22 menampilkan antar muka menu majemen server, menu ini merupakan menu yang berfungsi untuk pengelolaan aplikasi lapor yang terpasang di server. Didalam menu ini pengguna dapat mematikan aplikasi lapor tersebut dan menambahkan aplikasi microservice yang ingin dipantau oleh aplikasi monitoring dengan menekan tombol edit yang kemudian akan memunculkan modal seperti gambar 4.23 dibawah ini.

#### 4.4 Perancangan Implementasi

##### 4.4.1 Perencanaan Implementasi

Untuk mendukung Rancang Bangun Aplikasi Monitoring Resource Server dan Microservice Berbasis Web dengan Notifikasi Telegram Menggunakan Pendekatan Waterfall, persiapan alat-alat dalam kebutuhan perangkat keras ( hardware ) serta perangkat lunak ( software ) sangat penting. Berikut adalah rincian alat-alat yang perlu dipersiapkan:

1. Kebutuhan Alat Perangkat Keras ( Hardware )
  - 1) Laptop dengan Processor M1 dan RAM 8gb
  - 2) Harddisk minimal 50 gb untuk penyimpanan data
  - 3) Wifi dengan kecepatan minimal 10 mb/s
  - 4) Monitor 20 inch
  - 5) Kabel Konektor Monitor
2. Kebutuhan Alat Perangkat Lunak ( Software )
  - 1) MacOs sebagai sistem operasi
  - 2) Amppps dengan Apache yang sudah terinstall
  - 3) Mongodb sebagai basisdata
  - 4) IntelliJ IDE dan Visual Studio sebagai text editor
  - 5) Java dengan versi 13 sebagai bahasa pemrograman
  - 6) Web Browser sebagai alat untuk melakukan uji coba tampilan
  - 7) Postman sebagai alat untuk melakukan uji coba API

Sistem akan dibangun dengan kombinasi dari aplikasi server - side ( server - side application) dan aplikasi berbasis web. Aplikasi ini akan menggunakan

teknologi-teknologi yang handal dan modern, termasuk Java sebagai bahasa pemrograman utamanya, dan Spring Boot sebagai framework yang mendukung pengembangan aplikasi yang kuat dan skalabel. Untuk tampilan antar muka pengguna, aplikasi ini akan menggunakan HTML dan CSS. Selain itu, MongoDB akan digunakan sebagai basis data.

#### 4.4 2 Hasil Implementasi Berikut

merupakan hasil implementasi antarmuka berdasarkan perancangan antarmuka yang telah dibuat sebelumnya. Proses implementasi ini melibatkan beberapa tahapan penting yang bertujuan untuk memastikan antarmuka yang dihasilkan tidak hanya estetik, tetapi juga fungsional dan user - friendly sesuai dengan kebutuhan pengguna.

1. Login Pada gambar 4.24 merupakan hasil implementasi antarmuka login berdasarkan rancangan yang telah dibuat, didalam halaman ini pengguna akan diminta untuk memasukan username dan password yang telah didaftarkan agar bisa masuk ke halaman dashboard monitoring .
2. Realtime Monitoring Pada gambar 4.25 merupakan hasil implementasi antar muka untuk menu Realtime Monitoring berdasarkan rancangan yang telah dibuat, didalam halaman ini sistem akan menampilkan data terkini mengenai data resource dan status aplikasi microservice dari berbagai server yang terdaftar (yang terpasang aplikasi lapor). Apabila terdapat data resource baik itu CPU, Memory dan Storage yang melebihi batas wajar atau aplikasi microservice tidak berjalan, maka sistem akan mengirimkan notifikasi melalui telegram sesuai dengan data tersebut, berikut merupakan contoh dari notifikasi telegram yang dikirimkan oleh sistem.
3. History Monitoring Pada gambar 4.26 merupakan hasil implementasi antar muka untuk menu History Monitoring berdasarkan rancangan yang telah dibuat, didalam halaman tersebut ketika pengguna pertama kali membuka halaman tersebut, hanya akan terdapat field dropdown yang berisi list ip. Pengguna akan diminta untuk memilih salah satu IP yang diinginkan untuk melihat data history resource dan status aplikasi microservice . Setelah pengguna memilih IP, maka sistem akan mengambil data resource dan status aplikasi microservice berdasarkan IP yang telah dipilih selama 24 jam terakhir seperti pada gambar

4.27 dibawah. 4. Manajemen Server Pada gambar 4.28 merupakan hasil implementasi antar muka untuk menu Manajemen Server berdasarkan rancangan yang telah dibuat, didalam menu tersebut pengguna dapat mengelola data server yang terpasang aplikasi lapor didalam dashboard monitoring tersebut. Pengguna dapat mencari data server yang diinginkan dengan fitur pencarian yang terdapat diatas tabel. Selain itu pengguna dapat menghapus data server dengan menekan button delete yang ada di kolom Action pada data server yang diinginkan. Didalam kolom Action juga terdapat button edit yang memungkinkan pengguna dapat mengubah data server, ketika pengguna menekan button tersebut sistem akan mengeluarkan modal edit seperti gambar 4.29, untuk meminta pengguna mengubah data sesuai dengan yang diinginkan. 4.4 **32** 3 Pengujian Sistem Pengujian sistem dilakukan bertujuan untuk memastikan bahwa seluruh komponen sistem yang telah dibuat berfungsi sesuai dengan spesifikasi dan kebutuhan yang telah ditetapkan. **2** **15 20 25 39** Pengujian dilakukan berdasarkan rancangan yang telah dibuat sebelumnya, pengujian sistem dilakukan dengan menggunakan metode black box testing .

#### **2** Tabel 4. 13. Black Box Testing No Test Case Scenario Test

Result 1. dengan username dan password Mengakses halaman aplikasi monitoring username dan password yang telah terdaftar dan menekan tombol submit Sukses 2.Login dengan username dan password Mengakses halaman aplikasi monitoring username dan password yang tidak terdaftar dan menekan tombol submit 31 3.Menampilkan data monitoring terkini Button Realtime Monitoring pada sidebar Menampilkan halaman realtime resource server dan status aplikasi terkini 4. Menampilkan data history berdasarkan IP Button History Monitoring pada sidebar Menampilkan Halaman History dengan data history resource server dan Status microservice berdasarkan IP yang 6. Menambahkan data server Aplikasi lapor diaktifkan di server yang ingin di lakukan Aplikasi lapor mengirimkan data server ke aplikasi Data server terkirim 7.Menampilkan Data server Button Manajemen Server pada sidebar Menampilkan halaman Manajemen Server dengan list data server 8. edit Data Server dan Kirim ke aplikasi lapor

Button Manajemen Server pada sidebar Menekan tombol edit pada data server yang diinginkan edit data Menekan tombol submit Menampilkan konfirmasi perubahan data Klik button konfirmasi Data Server berhasil diubah dan terkirim 9. Menghapus Data Server dan Kirim ke aplikasi Menekan Button Manajemen Server pada sidebar Menekan tombol delete pada data server yang diinginkan Menampilkan konfirmasi penghapusan data Klik button konfirmasi Data Server berhasil di hapus dan terkirim 10. Menekan Button Logout Keluar dari halaman monitoring dan menuju ke login BAB V PENUTUP 5.1 Kesimpulan Penelitian dengan judul “Rancang Bangun Aplikasi Monitoring Resource Server dan Microservice Berbasis Web dengan Notifikasi Telegram di PT.

13 XYZ menggunakan Pendekatan Waterfall telah berhasil diimplementasikan dan memberikan dampak positif yang signifikan bagi Staff infrastruktur di PT. XYZ.

Berikut merupakan proses pengembangan yang telah dilakukan oleh penulis.

1. Perencanaan Penelitian ini dimulai dengan perencanaan yang memfokuskan pada identifikasi kebutuhan untuk mengatasi masalah dalam pemantauan resource server dan aplikasi microservice di PT. XYZ. Identifikasi ini secara efektif menetapkan ruang lingkup masalah yang terjadi dalam proses monitoring.
2. Analisis Selanjutnya, dilakukan analisis menyeluruh terhadap prosedur pemantauan yang ada, mencakup evaluasi dokumen dan sistem yang berjalan. Hasil analisis ini menjadi dasar untuk menentukan kebutuhan fungsional dan non- fungsional yang mendukung pembuatan diagram UML, rancangan basis data, dan desain antarmuka sistem monitoring.
3. Desain Tahap desain berikutnya melibatkan pembuatan berbagai diagram UML seperti use case diagram, activity diagram, sequence diagram, dan class diagram, serta perancangan basis data yang mencakup struktur tabel dan kolom yang dibutuhkan. Desain antarmuka sistem monitoring juga diperancang untuk memastikan kesesuaian tampilan dengan kebutuhan yang telah ditetapkan.
4. Implementasi Implementasi dilanjutkan dengan pengkodean sistem monitoring berdasarkan desain yang telah dibuat. Proses ini diikuti dengan pengujian sistem untuk memverifikasi bahwa sistem yang dikembangkan sesuai dengan spesifikasi dan dapat berfungsi dengan efektif

sesuai kebutuhan yang telah direncanakan sebelumnya. Sistem ini dirancang untuk mengatasi tantangan dalam pemantauan resource server dan status aplikasi microservice yang sebelumnya dilakukan secara manual dan berulang. Metode pemantauan yang sebelumnya juga membuat tim kesulitan dalam mendeteksi penggunaan resource yang tinggi dan kejadian seperti mati mendadak 32 pada aplikasi microservice . Dengan hadirnya aplikasi monitoring berbasis web ini, proses pemantauan menjadi lebih cepat dan mudah bagi staff infrastruktur. **70** Beberapa manfaat utama yang diperoleh dari sistem ini antara lain: 1. Pemantauan resource server dan status aplikasi microservice terhadap setiap server dapat dilakukan dalam satu dashboard monitoring yang terpadu. Hal ini memudahkan Staff infrastruktur untuk mendapatkan gambaran lengkap tentang kondisi server dan aplikasi microservice terkini yang mereka kelola. 2. Sistem ini dilengkapi dengan fitur notifikasi melalui Telegram. Ketika penggunaan resource melebihi batas atau status aplikasi microservice mati, notifikasi segera dikirimkan kepada Staff infrastruktur. **52** Dengan demikian, tim dapat segera merespons dan mengambil tindakan yang diperlukan untuk mengatasi masalah tersebut. 3. Sistem ini juga menyimpan data riwayat resource server dan status aplikasi microservice . Data riwayat ini dapat digunakan untuk analisis tren dan pengambilan keputusan strategis. Misalnya, tim dapat menggunakan data ini untuk merencanakan upgrade pada server, mengatur beban kerja server pada jam tertentu, dan melakukan langkah- langkah preventif lainnya guna mencegah terjadinya downtime pada aplikasi microservice . Dengan demikian, implementasi aplikasi monitoring ini telah berhasil mengatasi masalah dalam aktivitas monitoring di PT. XYZ. Diharapkan, sistem ini dapat terus memberikan manfaat yang signifikan dalam aktivitas monitoring yang dilakukan, serta mendukung operasional perusahaan yang lebih efisien dan andal. Staff infrastruktur kini memiliki alat yang efektif untuk memastikan resource server dan aplikasi microservice selalu dalam kondisi optimal, sehingga mendukung kelancaran operasional perusahaan secara keseluruhan. 5.2 Saran Penulis telah berhasil mengembangkan aplikasi



monitoring yang sesuai dengan kebutuhan PT. XYZ. Mengingat bahwa sistem ini merupakan teknologi yang baru, penulis telah melampirkan panduan instalasi yang lengkap didalam penelitian ini. Panduan ini dirancang untuk memudahkan pengguna dalam menginstal aplikasi lapor tersebut pada setiap server yang ingin dipantau. Dengan adanya panduan ini, diharapkan pengguna dapat dengan mudah menginstal sistem tanpa menemui kendala berarti. Dengan demikian, sistem ini dapat diimplementasikan dengan lancar dan efektif pada berbagai server yang membutuhkan pemantauan. Sistem informasi monitoring yang telah dibuat oleh penulis, tentunya masih memiliki potensi untuk dikembangkan lebih lanjut agar dapat memenuhi kebutuhan yang mungkin akan dibutuhkan kedepannya. 9 Berikut merupakan saran dari penulis untuk pengembangan lebih lanjut pada aplikasi monitoring PT. XYZ: 1. Modul Manajemen Pengguna Dikarenakan saat ini hanya sedikit dari Staff infrastruktur yang ada di PT. XYZ jadi belum diperlukan modul manajemen pengguna, namun kedepannya apabila terdapat banyak anggota Staff infrastruktur diperlukan tambahan modul tersebut yang memungkinkan pengaturan hak akses dan peran. Modul tersebut akan membantu dalam mengelola siapa yang dapat melihat, mengedit, atau menghapus data. 2. Pelacakan Aktivitas Pengelolaan Manajemen Server Setelah dibuatkan modul manajemen pengguna kemudian diperlukan fitur untuk melacak aktivitas pengelolaan manajemen server, seperti siapa yang melakukan edit atau delete pada modul tersebut. Dengan adanya log aktivitas, dapat mempermudah dalam mengidentifikasi perubahan yang dilakukan dan oleh siapa, serta meningkatkan keamanan dan akuntabilitas 3. Perbaikan Tampilan Antarmuka Perbaikan tampilan antarmuka (UI) diperlukan terutama jika jumlah server yang ingin dipantau cukup banyak (lebih dari 20 server). Antarmuka yang lebih intuitif dan terorganisir akan membantu Staff infrastruktur untuk memantau dan mengelola banyak server dengan lebih mudah. Dengan menerapkan saran-saran tersebut, diharapkan aplikasi monitoring PT. XYZ akan menjadi lebih canggih, responsif, dan sesuai dengan kebutuhan operasional perusahaan, sehingga dapat meningkatkan

REPORT #21954173

efisiensi dalam hal waktu dan kinerja Staff infrastruktur secara keseluruhan. 33



REPORT #21954173

## Results

Sources that matched your submitted document.

● IDENTICAL ● CHANGED TEXT

INTERNET SOURCE		
1.	<b>0.53%</b> leravio.com	●
	<a href="https://leravio.com/blog/mengenal-konsep-dasar-database/">https://leravio.com/blog/mengenal-konsep-dasar-database/</a>	
INTERNET SOURCE		
2.	<b>0.49%</b> media.neliti.com	●
	<a href="https://media.neliti.com/media/publications/300977-monitoring-aplikasi-meng...">https://media.neliti.com/media/publications/300977-monitoring-aplikasi-meng...</a>	
INTERNET SOURCE		
3.	<b>0.47%</b> repository.bsi.ac.id	●
	<a href="https://repository.bsi.ac.id/repo/files/368929/download/Sistem-Basis-Data-Full...">https://repository.bsi.ac.id/repo/files/368929/download/Sistem-Basis-Data-Full...</a>	
INTERNET SOURCE		
4.	<b>0.42%</b> repository.uinsu.ac.id	●
	<a href="http://repository.uinsu.ac.id/15067/1/Final%20Wisuda%20Fix.pdf">http://repository.uinsu.ac.id/15067/1/Final%20Wisuda%20Fix.pdf</a>	
INTERNET SOURCE		
5.	<b>0.4%</b> repository.utp.ac.id	●
	<a href="http://repository.utp.ac.id/1291/1/BUKU%20PENGANTAR%20SISTEM%20INFOR...">http://repository.utp.ac.id/1291/1/BUKU%20PENGANTAR%20SISTEM%20INFOR...</a>	
INTERNET SOURCE		
6.	<b>0.37%</b> si.stechoq.com	●
	<a href="https://si.stechoq.com/mod/page/view.php?id=3258">https://si.stechoq.com/mod/page/view.php?id=3258</a>	
INTERNET SOURCE		
7.	<b>0.34%</b> eprints.itn.ac.id	●
	<a href="http://eprints.itn.ac.id/4541/1/1.%20BAGIAN%20AWAL.pdf">http://eprints.itn.ac.id/4541/1/1.%20BAGIAN%20AWAL.pdf</a>	
INTERNET SOURCE		
8.	<b>0.33%</b> repository.dinamika.ac.id	●
	<a href="https://repository.dinamika.ac.id/id/eprint/3341/1/14410100159-2019-STIKOMS...">https://repository.dinamika.ac.id/id/eprint/3341/1/14410100159-2019-STIKOMS...</a>	
INTERNET SOURCE		
9.	<b>0.33%</b> repository.uinjkt.ac.id	●
	<a href="https://repository.uinjkt.ac.id/dspace/bitstream/123456789/48082/1/MUHAMMA...">https://repository.uinjkt.ac.id/dspace/bitstream/123456789/48082/1/MUHAMMA...</a>	



REPORT #21954173

INTERNET SOURCE		
10. 0.32%	repository.uinjkt.ac.id <a href="https://repository.uinjkt.ac.id/dspace/bitstream/123456789/72887/1/AHMAD%2...">https://repository.uinjkt.ac.id/dspace/bitstream/123456789/72887/1/AHMAD%2...</a>	●
INTERNET SOURCE		
11. 0.26%	e-journal.uajy.ac.id <a href="http://e-journal.uajy.ac.id/10902/2/1TF05918.pdf">http://e-journal.uajy.ac.id/10902/2/1TF05918.pdf</a>	●
INTERNET SOURCE		
12. 0.26%	repository.unair.ac.id <a href="https://repository.unair.ac.id/56073/2/KKC%20KK%20ST.SI%2034%20-16%20Vi...">https://repository.unair.ac.id/56073/2/KKC%20KK%20ST.SI%2034%20-16%20Vi...</a>	●
INTERNET SOURCE		
13. 0.25%	mahasiswa.ung.ac.id <a href="https://mahasiswa.ung.ac.id/921411195/home/2012/10">https://mahasiswa.ung.ac.id/921411195/home/2012/10</a>	●
INTERNET SOURCE		
14. 0.24%	appmaster.io <a href="https://appmaster.io/id/blog/apa-itu-basis-data-relasional">https://appmaster.io/id/blog/apa-itu-basis-data-relasional</a>	●
INTERNET SOURCE		
15. 0.24%	repository.uml.ac.id <a href="http://repository.uml.ac.id/3162/1/Skripsi%20Faisal%20Mahfud.pdf">http://repository.uml.ac.id/3162/1/Skripsi%20Faisal%20Mahfud.pdf</a>	●
INTERNET SOURCE		
16. 0.23%	repository.uir.ac.id <a href="https://repository.uir.ac.id/15007/1/133510583.pdf">https://repository.uir.ac.id/15007/1/133510583.pdf</a>	●
INTERNET SOURCE		
17. 0.23%	static.buku.kemdikbud.go.id <a href="https://static.buku.kemdikbud.go.id/content/pdf/bukuteks/kurikulum21/Dasar-...">https://static.buku.kemdikbud.go.id/content/pdf/bukuteks/kurikulum21/Dasar...</a>	●
INTERNET SOURCE		
18. 0.22%	repository.upiyptk.ac.id <a href="http://repository.upiyptk.ac.id/5279/2/SKRIPSI_KURNIA%20SANDI_15101152610..">http://repository.upiyptk.ac.id/5279/2/SKRIPSI_KURNIA%20SANDI_15101152610..</a>	●
INTERNET SOURCE		
19. 0.22%	dte.telkomuniversity.ac.id <a href="https://dte.telkomuniversity.ac.id/mengenal-dbms-konsep-dasar-unsur-elemen...">https://dte.telkomuniversity.ac.id/mengenal-dbms-konsep-dasar-unsur-elemen...</a>	●
INTERNET SOURCE		
20. 0.22%	repository.uinjkt.ac.id <a href="https://repository.uinjkt.ac.id/dspace/bitstream/123456789/71766/1/FITRI%20Y...">https://repository.uinjkt.ac.id/dspace/bitstream/123456789/71766/1/FITRI%20Y...</a>	●



REPORT #21954173

INTERNET SOURCE		
21.	<b>0.21%</b> <a href="https://lmsspada.kemdikbud.go.id">lmsspada.kemdikbud.go.id</a> <i>https://lmsspada.kemdikbud.go.id/course/view.php?id=3759&amp;section=15</i>	●
INTERNET SOURCE		
22.	<b>0.21%</b> <a href="https://www.divasoft.net">www.divasoft.net</a> <i>https://www.divasoft.net/blog/apa-itu-api-cara-kerja-jenis-manfaat-dan-contoh...</i>	●
INTERNET SOURCE		
23.	<b>0.21%</b> <a href="https://eprints.perbanas.ac.id">eprints.perbanas.ac.id</a> <i>https://eprints.perbanas.ac.id/1922/2/BAB%20I.pdf</i>	●
INTERNET SOURCE		
24.	<b>0.21%</b> <a href="https://binus.ac.id">binus.ac.id</a> <i>https://binus.ac.id/malang/2024/05/istilah-serta-perbedaan-basis-data-dan-sist...</i>	●
INTERNET SOURCE		
25.	<b>0.2%</b> <a href="https://repository.its.ac.id">repository.its.ac.id</a> <i>https://repository.its.ac.id/105138/1/5210100024_Undergraduate_Thesis.pdf</i>	●
INTERNET SOURCE		
26.	<b>0.19%</b> <a href="https://lppm.itda.ac.id">lppm.itda.ac.id</a> <i>https://lppm.itda.ac.id/sip/assets/laporan/penelitian/2018/Laporan_Penelitian_...</i>	●
INTERNET SOURCE		
27.	<b>0.19%</b> <a href="https://elibrary.unikom.ac.id">elibrary.unikom.ac.id</a> <i>https://elibrary.unikom.ac.id/id/eprint/5873/7/UNIKOM_YOGI%20R%20ADITAMA..</i>	●
INTERNET SOURCE		
28.	<b>0.19%</b> <a href="https://repository.umj.ac.id">repository.umj.ac.id</a> <i>https://repository.umj.ac.id/15992/1/Skripsi%20Ahmad%20Yodi%20Desyandi%...</i>	●
INTERNET SOURCE		
29.	<b>0.19%</b> <a href="https://repository.uinjkt.ac.id">repository.uinjkt.ac.id</a> <i>https://repository.uinjkt.ac.id/dspace/bitstream/123456789/28147/1/RIDWAN%2..</i>	●
INTERNET SOURCE		
30.	<b>0.17%</b> <a href="https://www.slideshare.net">www.slideshare.net</a> <i>https://www.slideshare.net/slideshow/sequence-diagram-33885923/33885923</i>	●
INTERNET SOURCE		
31.	<b>0.17%</b> <a href="https://ejournal.itn.ac.id">ejournal.itn.ac.id</a> <i>https://ejournal.itn.ac.id/index.php/jati/article/view/2310</i>	●



REPORT #21954173

INTERNET SOURCE		
32.	0.16% <a href="https://ejournal.itn.ac.id">ejournal.itn.ac.id</a> <a href="https://ejournal.itn.ac.id/index.php/jati/article/download/9859/5672/">https://ejournal.itn.ac.id/index.php/jati/article/download/9859/5672/</a>	●
INTERNET SOURCE		
33.	0.16% <a href="https://repository.uinjkt.ac.id">repository.uinjkt.ac.id</a> <a href="https://repository.uinjkt.ac.id/dspace/bitstream/123456789/48115/1/ACHMAD%...">https://repository.uinjkt.ac.id/dspace/bitstream/123456789/48115/1/ACHMAD%...</a>	●
INTERNET SOURCE		
34.	0.15% <a href="https://www.jointer.id">www.jointer.id</a> <a href="https://www.jointer.id/index.php/jointer/article/download/6/1">https://www.jointer.id/index.php/jointer/article/download/6/1</a>	●
INTERNET SOURCE		
35.	0.15% <a href="https://it.proxsisgroup.com">it.proxsisgroup.com</a> <a href="https://it.proxsisgroup.com/kelebihan-dan-kekurangan-togaf-vs-cobit-analisis-l...">https://it.proxsisgroup.com/kelebihan-dan-kekurangan-togaf-vs-cobit-analisis-l...</a>	●
INTERNET SOURCE		
36.	0.14% <a href="https://eproceeding.itenas.ac.id">eproceeding.itenas.ac.id</a> <a href="https://eproceeding.itenas.ac.id/index.php/fti/article/download/1568/1320/1555">https://eproceeding.itenas.ac.id/index.php/fti/article/download/1568/1320/1555</a>	●
INTERNET SOURCE		
37.	0.14% <a href="https://eprints.upj.ac.id">eprints.upj.ac.id</a> <a href="https://eprints.upj.ac.id/id/eprint/6245/11/BAB%20IV.pdf">https://eprints.upj.ac.id/id/eprint/6245/11/BAB%20IV.pdf</a>	●
INTERNET SOURCE		
38.	0.13% <a href="https://aws.amazon.com">aws.amazon.com</a> <a href="https://aws.amazon.com/id/what-is/iaas/">https://aws.amazon.com/id/what-is/iaas/</a>	●
INTERNET SOURCE		
39.	0.13% <a href="https://www.slideshare.net">www.slideshare.net</a> <a href="https://www.slideshare.net/slideshow/arsitektur-pemodelan-data/249912027">https://www.slideshare.net/slideshow/arsitektur-pemodelan-data/249912027</a>	●
INTERNET SOURCE		
40.	0.12% <a href="https://jtlm.akamigas.ac.id">jtlm.akamigas.ac.id</a> <a href="https://jtlm.akamigas.ac.id/index.php/jtlmig/article/download/15/6/18">https://jtlm.akamigas.ac.id/index.php/jtlmig/article/download/15/6/18</a>	●
INTERNET SOURCE		
41.	0.12% <a href="https://www.asdf.id">www.asdf.id</a> <a href="https://www.asdf.id/software-deployment-pengertian-tujuan-manfaat-dan-taha..">https://www.asdf.id/software-deployment-pengertian-tujuan-manfaat-dan-taha..</a>	●
INTERNET SOURCE		
42.	0.12% <a href="https://repository.bsi.ac.id">repository.bsi.ac.id</a> <a href="https://repository.bsi.ac.id/repo/files/361396/download/File_16-BAB-IV-Peranca...">https://repository.bsi.ac.id/repo/files/361396/download/File_16-BAB-IV-Peranca...</a>	●



REPORT #21954173

INTERNET SOURCE		
43.	0.12% widuri.raharjo.info <a href="https://widuri.raharjo.info/index.php?title=SI1122466661">https://widuri.raharjo.info/index.php?title=SI1122466661</a>	●
INTERNET SOURCE		
44.	0.11% appmaster.io <a href="https://appmaster.io/id/blog/analisis-kebutuhan-perangkat-lunak">https://appmaster.io/id/blog/analisis-kebutuhan-perangkat-lunak</a>	●
INTERNET SOURCE		
45.	0.11% rama.unimal.ac.id <a href="https://rama.unimal.ac.id/id/eprint/584/3/Yolanda%20Pardede_190180056_Bab..">https://rama.unimal.ac.id/id/eprint/584/3/Yolanda%20Pardede_190180056_Bab..</a>	●
INTERNET SOURCE		
46.	0.11% www.ruangkerja.id <a href="https://www.ruangkerja.id/blog/manajemen-proyek">https://www.ruangkerja.id/blog/manajemen-proyek</a>	●
INTERNET SOURCE		
47.	0.11% nurosoft.id <a href="https://nurosoft.id/blog/frontend-engineer-adalah/">https://nurosoft.id/blog/frontend-engineer-adalah/</a>	●
INTERNET SOURCE		
48.	0.1% repository.atmaluhur.ac.id <a href="https://repository.atmaluhur.ac.id/bitstream/handle/123456789/3386/BAB%20I...">https://repository.atmaluhur.ac.id/bitstream/handle/123456789/3386/BAB%20I...</a>	●
INTERNET SOURCE		
49.	0.1% repository.bsi.ac.id <a href="https://repository.bsi.ac.id/repo/files/288887/download/File_16-Bab-IV-Peranca...">https://repository.bsi.ac.id/repo/files/288887/download/File_16-Bab-IV-Peranca...</a>	●
INTERNET SOURCE		
50.	0.09% aws.amazon.com <a href="https://aws.amazon.com/id/what-is/sdlc/">https://aws.amazon.com/id/what-is/sdlc/</a>	●
INTERNET SOURCE		
51.	0.09% ejournal.itn.ac.id <a href="https://ejournal.itn.ac.id/index.php/jati/article/download/6064/3587">https://ejournal.itn.ac.id/index.php/jati/article/download/6064/3587</a>	●
INTERNET SOURCE		
52.	0.09% ivosights.com <a href="https://ivosights.com/read/artikel/command-center-kelemahan-pada-sistem-ko..">https://ivosights.com/read/artikel/command-center-kelemahan-pada-sistem-ko..</a>	●
INTERNET SOURCE		
53.	0.09% pustaka.ut.ac.id <a href="https://pustaka.ut.ac.id/lib/wp-content/uploads/pdfmk/MSIM4302-M1.pdf">https://pustaka.ut.ac.id/lib/wp-content/uploads/pdfmk/MSIM4302-M1.pdf</a>	●



REPORT #21954173

INTERNET SOURCE		
54.	0.08% magicvalleyalpacas.com <a href="https://magicvalleyalpacas.com/2023/07/">https://magicvalleyalpacas.com/2023/07/</a>	●
INTERNET SOURCE		
55.	0.08% www.ekrut.com <a href="https://www.ekrut.com/media/microservices-adalah">https://www.ekrut.com/media/microservices-adalah</a>	●
INTERNET SOURCE		
56.	0.08% floothink.com <a href="https://floothink.com/article-detail/39/Memahami-Tahapan-dalam-Software-De...">https://floothink.com/article-detail/39/Memahami-Tahapan-dalam-Software-De...</a>	●
INTERNET SOURCE		
57.	0.08% repository.dinamika.ac.id <a href="https://repository.dinamika.ac.id/id/eprint/1062/5/BAB_II.pdf">https://repository.dinamika.ac.id/id/eprint/1062/5/BAB_II.pdf</a>	●
INTERNET SOURCE		
58.	0.08% alldataint.com <a href="https://alldataint.com/struktur-data-dalam-database-operasional-memahami-o...">https://alldataint.com/struktur-data-dalam-database-operasional-memahami-o...</a>	●
INTERNET SOURCE		
59.	0.08% jurnal.penerbitdaarulhuda.my.id <a href="https://jurnal.penerbitdaarulhuda.my.id/index.php/MAJIM/article/viewFile/411/...">https://jurnal.penerbitdaarulhuda.my.id/index.php/MAJIM/article/viewFile/411/...</a>	●
INTERNET SOURCE		
60.	0.08% www.slideshare.net <a href="https://www.slideshare.net/slideshow/penyajian-data-dalam-diagram/2427492...">https://www.slideshare.net/slideshow/penyajian-data-dalam-diagram/2427492...</a>	●
INTERNET SOURCE		
61.	0.08% ejournal.uin-suska.ac.id <a href="https://ejournal.uin-suska.ac.id/index.php/sitekin/article/download/13934/6909">https://ejournal.uin-suska.ac.id/index.php/sitekin/article/download/13934/6909</a>	●
INTERNET SOURCE		
62.	0.07% biztechacademy.id <a href="https://biztechacademy.id/8-alat-populer-dalam-analisis-data-yang-harus-diket...">https://biztechacademy.id/8-alat-populer-dalam-analisis-data-yang-harus-diket...</a>	●
INTERNET SOURCE		
63.	0.07% bakri.uma.ac.id <a href="https://bakri.uma.ac.id/ram-dan-rom-perbedaan-keduanya-pada-laptop-dan-ha..">https://bakri.uma.ac.id/ram-dan-rom-perbedaan-keduanya-pada-laptop-dan-ha..</a>	●
INTERNET SOURCE		
64.	0.07% idwebhost.com <a href="https://idwebhost.com/blog/pengertian-ram-adalah/">https://idwebhost.com/blog/pengertian-ram-adalah/</a>	●





REPORT #21954173

INTERNET SOURCE		
65.	0.07% <a href="http://www.divasoft.net">www.divasoft.net</a>	●
	<a href="https://www.divasoft.net/blog/13-framework-pemrograman-web-terbaik-untuk...">https://www.divasoft.net/blog/13-framework-pemrograman-web-terbaik-untuk...</a>	
INTERNET SOURCE		
66.	0.07% <a href="http://cloudraya.com">cloudraya.com</a>	●
	<a href="https://cloudraya.com/blog/apa-itu-bandwidth/">https://cloudraya.com/blog/apa-itu-bandwidth/</a>	
INTERNET SOURCE		
67.	0.07% <a href="http://blog.wowrack.co.id">blog.wowrack.co.id</a>	●
	<a href="https://blog.wowrack.co.id/2023/08/11/apa-itu-server-memahami-peran-dan-fu...">https://blog.wowrack.co.id/2023/08/11/apa-itu-server-memahami-peran-dan-fu...</a>	
INTERNET SOURCE		
68.	0.06% <a href="http://id.linkedin.com">id.linkedin.com</a>	●
	<a href="https://id.linkedin.com/pulse/aplikasi-gratis-open-source-untuk-membuat-dash...">https://id.linkedin.com/pulse/aplikasi-gratis-open-source-untuk-membuat-dash...</a>	
INTERNET SOURCE		
69.	0.06% <a href="http://eprints.upj.ac.id">eprints.upj.ac.id</a>	●
	<a href="https://eprints.upj.ac.id/id/eprint/3482/11/BAB%20III.pdf">https://eprints.upj.ac.id/id/eprint/3482/11/BAB%20III.pdf</a>	
INTERNET SOURCE		
70.	0.05% <a href="http://media.neliti.com">media.neliti.com</a>	●
	<a href="https://media.neliti.com/media/publications/409688-sistem-pemantauan-aktivi...">https://media.neliti.com/media/publications/409688-sistem-pemantauan-aktivi...</a>	
INTERNET SOURCE		
71.	0.05% <a href="http://animator.uho.ac.id">animator.uho.ac.id</a>	●
	<a href="https://animator.uho.ac.id/index.php/journal/article/download/30/15/106">https://animator.uho.ac.id/index.php/journal/article/download/30/15/106</a>	
INTERNET SOURCE		
72.	0.05% <a href="http://eprints.upj.ac.id">eprints.upj.ac.id</a>	●
	<a href="https://eprints.upj.ac.id/id/eprint/6640/11/BAB%20IV.pdf">https://eprints.upj.ac.id/id/eprint/6640/11/BAB%20IV.pdf</a>	
INTERNET SOURCE		
73.	0.04% <a href="http://eprints.itn.ac.id">eprints.itn.ac.id</a>	●
	<a href="http://eprints.itn.ac.id/view/subjects/INF.type.html">http://eprints.itn.ac.id/view/subjects/INF.type.html</a>	
INTERNET SOURCE		
74.	0.04% <a href="http://www.domainesia.com">www.domainesia.com</a>	●
	<a href="https://www.domainesia.com/berita/pengertian-erd-adalah/">https://www.domainesia.com/berita/pengertian-erd-adalah/</a>	



REPORT #21954173

● QUOTES

INTERNET SOURCE

1. **0.3%** [informatika.untag-sby.ac.id](https://informatika.untag-sby.ac.id)  
[https://informatika.untag-sby.ac.id/backend/uploads/pdf/Achmad\\_Syarif\\_1461...](https://informatika.untag-sby.ac.id/backend/uploads/pdf/Achmad_Syarif_1461...)

INTERNET SOURCE

2. **0.24%** [jahorinaforum.com](https://jahorinaforum.com)  
<https://jahorinaforum.com/page/4/>

INTERNET SOURCE

3. **0.22%** [informatika.untag-sby.ac.id](https://informatika.untag-sby.ac.id)  
[https://informatika.untag-sby.ac.id/backend/uploads/pdf/Muhammad\\_Fikri\\_Ba...](https://informatika.untag-sby.ac.id/backend/uploads/pdf/Muhammad_Fikri_Ba...)

INTERNET SOURCE

4. **0.15%** [static.buku.kemdikbud.go.id](https://static.buku.kemdikbud.go.id)  
<https://static.buku.kemdikbud.go.id/content/pdf/bukuteks/kurikulum21/Inform...>

INTERNET SOURCE

5. **0.12%** [ejournal.itn.ac.id](https://ejournal.itn.ac.id)  
<https://ejournal.itn.ac.id/index.php/jati/article/view/2310>

INTERNET SOURCE

6. **0.1%** [media.neliti.com](https://media.neliti.com)  
<https://media.neliti.com/media/publications/300977-monitoring-aplikasi-meng...>

INTERNET SOURCE

7. **0.09%** [katadata.co.id](https://katadata.co.id)  
<https://katadata.co.id/digital/teknologi/62a8764e85565/diagram-konteks-adala...>

INTERNET SOURCE

8. **0.09%** [ejournal.itn.ac.id](https://ejournal.itn.ac.id)  
<https://ejournal.itn.ac.id/index.php/jati/article/download/9859/5672/>

INTERNET SOURCE

9. **0.08%** [eprints.upj.ac.id](https://eprints.upj.ac.id)  
<https://eprints.upj.ac.id/id/eprint/6245/11/BAB%20IV.pdf>

INTERNET SOURCE

10. **0.08%** [repository.uinjkt.ac.id](https://repository.uinjkt.ac.id)  
<https://repository.uinjkt.ac.id/dspace/bitstream/123456789/71766/1/FITRI%20Y...>

INTERNET SOURCE

11. **0.07%** [prakerlap.smkn10-mlg.sch.id](https://prakerlap.smkn10-mlg.sch.id)  
[https://prakerlap.smkn10-mlg.sch.id/manual\\_book/\(Guru\)-Buku%20Petunjuk%](https://prakerlap.smkn10-mlg.sch.id/manual_book/(Guru)-Buku%20Petunjuk%...)



REPORT #21954173

INTERNET SOURCE

**12. 0.06%** widuri.raharja.info

<https://widuri.raharja.info/index.php?title=SI1122466661>

INTERNET SOURCE

**13. 0.02%** verihubs.com

<https://verihubs.com/blog/aplikasi-verifikasi-pendapatan/>

INTERNET SOURCE

**14. 0.01%** repository.bsi.ac.id

[https://repository.bsi.ac.id/repo/files/288887/download/File\\_16-Bab-IV-Peranca...](https://repository.bsi.ac.id/repo/files/288887/download/File_16-Bab-IV-Peranca...)