

# Implementasi Metode *Design Science Research* (DSR) pada *Monitoring Server Voip* Berbasis *Web* dengan Pendekatan *Agile* untuk Optimalisasi Kinerja Sistem (Studi Kasus: PT. XYZ)

Dimas Adjie<sup>1</sup>, Hadi Zakaria<sup>2\*</sup>

<sup>1,2</sup>Teknik Informatika, Ilmu Komputer, Universitas Pamulang  
[official.dimasadjie@gmail.com](mailto:official.dimasadjie@gmail.com), [dosen00274@unpam.ac.id](mailto:dosen00274@unpam.ac.id)

## Abstract

PT. XYZ is a telecommunications technology company that has been providing cloud-based communication services, such as Cloud PBX, VoIP Telephony, Cloud Call Centers, and integrated Omni-Channel CRM systems, since 2013. The scope of this research is focused on infrastructure optimization through the development of a centralized monitoring system to manage thousands of call history records and server performance data. The background of this study is driven by the increasing number of clients and system complexity, which has rendered manual monitoring methods ineffective. This inefficiency leads to risks such as limited performance visibility, potential human error, and undetected security threats. Consequently, this research aims to build an innovative artifact in the form of a web-based VoIP server monitoring system capable of providing automated, centralized, and real-time monitoring. The methodology employed is Design Science Research (DSR) combined with an Agile approach to create a system that is adaptive to the company's changing needs. The system was developed using the Laravel framework, Python for backend processing, Chart.js for interactive visualization, and MySQL for telemetry data storage. The collected monitoring data includes CPU, memory, and disk utilization, connectivity status, and essential port openness, gathered by agents every minute. Implementation results indicate that the system successfully presents current server conditions, detects resource usage anomalies, and provides automated notifications during service disruptions. In conclusion, this system significantly improves operational efficiency, service security, and the quality of VoIP server monitoring, while strengthening the company's position as a reliable and trusted cloud-based communication solution provider in Indonesia.

Keywords: design science research, agile, monitoring, VoIP server, web-based system

## Abstrak

PT. XYZ merupakan perusahaan teknologi telekomunikasi yang sejak 2013 menyediakan layanan komunikasi berbasis *cloud* seperti *Cloud PBX*, *VoIP Telephony*, *Cloud Call Center*, serta sistem CRM *Omni Channel* yang terintegrasi. Ruang lingkup penelitian ini difokuskan pada optimalisasi infrastruktur melalui pengembangan sistem pemantauan yang terpusat untuk mengelola ribuan data riwayat panggilan dan performa *server*. Latar belakang penelitian ini didasari oleh meningkatnya jumlah klien dan kompleksitas sistem yang menyebabkan metode *monitoring* manual menjadi tidak efektif, sehingga menimbulkan risiko keterbatasan visibilitas performa, potensi kesalahan manusia, serta ancaman keamanan yang tidak terdeteksi secara dini. Oleh karena itu, penelitian ini bertujuan untuk membangun artefak inovatif berupa sistem *monitoring server* VoIP berbasis *web* yang mampu menyediakan pemantauan otomatis, terpusat, dan *real-time*. Metode yang digunakan adalah *Design Science Research* (DSR) yang dikombinasikan dengan pendekatan *Agile* untuk menciptakan sistem yang adaptif terhadap perubahan kebutuhan perusahaan. Sistem ini dikembangkan menggunakan *framework Laravel*, pemrosesan backend dengan *Python*, visualisasi interaktif *Chart.js*, dan penyimpanan data telemetri pada *MySQL*. Data *monitoring* yang dikumpulkan mencakup utilisasi CPU, memori, disk, status konektivitas, hingga status keterbukaan port esensial yang diambil oleh agent setiap menit. Hasil implementasi menunjukkan bahwa sistem berhasil menyajikan kondisi *server* secara aktual, mendeteksi anomali penggunaan *resource*, serta memberikan notifikasi otomatis saat terjadi gangguan layanan. Kesimpulannya, sistem ini secara signifikan berhasil meningkatkan efisiensi operasional, keamanan layanan, dan kualitas pemantauan *server* VoIP, serta memperkuat posisi perusahaan sebagai penyedia solusi komunikasi berbasis *cloud* yang andal dan terpercaya di Indonesia.

Kata kunci: design science research, agile, monitoring, server VoIP, sistem berbasis *Web*

©This work is licensed under a Creative Commons Attribution - ShareAlike 4.0 International License

## 1. Pendahuluan

PT. XYZ merupakan perusahaan teknologi telekomunikasi yang telah berfokus pada layanan *Voice over Internet Protocol* (VoIP) sejak tahun 2013. Perusahaan ini menyediakan solusi *Cloud PBX* dan ekosistem komunikasi terpadu untuk membantu transformasi bisnis dari sistem telepon konvensional ke komunikasi digital yang lebih fleksibel [1]. Seiring bertambahnya jumlah klien dan semakin kompleksnya infrastruktur yang dikelola, perusahaan menghadapi tantangan dalam hal pemantauan performa dan keamanan *server* VoIP yang digunakan untuk

mendukung layanan tersebut. Ketergantungan terhadap proses *monitoring* manual serta kurangnya visibilitas *real-time* terhadap kondisi server menyebabkan potensi risiko gangguan layanan, kebocoran data, dan menurunnya efisiensi operasional. Proses *monitoring* manual yang dilakukan selama ini mencakup aktivitas *login* ke masing-masing *server* VoIP secara satu per satu, kemudian memeriksa beberapa aspek penting melalui menu yang berbeda, seperti *monitoring resource usage*, *call history*, *audit log*, *service status*, versi *server* VoIP, serta pengecekan keamanan secara manual melalui pengujian *port* jaringan yang terbuka atau tertutup. Seluruh hasil *monitoring* tersebut

kemudian dicatat secara manual untuk keperluan dokumentasi dan analisis internal. Permasalahan ini memerlukan solusi sistematis agar kualitas layanan komunikasi tetap terjaga dan kebutuhan pelanggan dapat terpenuhi dengan optimal.

Implementasi merupakan tahapan penerapan dari suatu ide, konsep, kebijakan, atau inovasi ke dalam tindakan nyata yang bertujuan menghasilkan dampak tertentu. Dampak tersebut dapat berupa perubahan dalam pengetahuan, peningkatan keterampilan, serta pergeseran nilai dan sikap individu maupun kelompok yang terlibat. Proses ini berjalan secara sistematis dan dirancang secara terstruktur untuk menjamin tercapainya tujuan dengan efektif. Selain itu, proses implementasi umumnya juga melibatkan kegiatan evaluasi dan penyesuaian guna menghadapi berbagai tantangan serta hambatan yang mungkin muncul, sehingga hasil yang diperoleh dapat lebih optimal [2]. Dalam konteks penelitian ini, implementasi difokuskan pada penerapan sistem *monitoring server* VoIP berbasis *web* dengan pendekatan *Agile* dan metode *Design Science Research* (DSR), guna meningkatkan efisiensi operasional serta keamanan infrastruktur komunikasi pada PT. XYZ.

Metode *Design Science Research* (DSR) merupakan pendekatan pemecahan masalah yang berfokus pada peningkatan pengetahuan melalui penciptaan artefak inovatif. Tujuan utama dari metode ini adalah memperluas basis pengetahuan dalam bidang teknologi dan sains dengan menghasilkan solusi yang tidak hanya praktis, tetapi juga berdampak terhadap peningkatan lingkungan atau sistem tempat artefak tersebut diterapkan. Artefak yang dihasilkan dapat berupa rancangan sistem, model, metode, atau instansi teknologi yang baru, sekaligus memberikan pemahaman teoritis melalui *Design Knowledge* (DK), yaitu pengetahuan mengenai alasan mengapa artefak tersebut dapat memberikan dampak positif terhadap konteks penggunaannya [3]. Dalam penelitian ini, metode DSR diterapkan untuk merancang dan mengembangkan sistem *monitoring server* VoIP berbasis *web* yang bertujuan mengatasi kendala efisiensi pemantauan *server*, meningkatkan keamanan sistem, serta menyediakan visibilitas yang terintegrasi bagi PT. XYZ.

*Monitoring* merupakan kegiatan yang dilakukan secara berkala untuk mengumpulkan data dan mengukur kemajuan terhadap tujuan dari suatu program. Proses ini menekankan pada pelaksanaan dan hasil (*output*) dengan tujuan utama untuk memastikan bahwa aktivitas yang dilakukan sesuai dengan rencana dan prosedur yang telah ditetapkan. *Monitoring* tidak hanya mencatat apa yang telah dilakukan, tetapi juga mengevaluasi kualitas layanan yang diberikan. Informasi dikumpulkan secara reguler melalui indikator tertentu guna mengamati apakah kegiatan yang sedang berjalan tetap berada pada jalur yang benar. Indikator ini mencerminkan inti dari aktivitas serta target yang ingin dicapai berdasarkan

perencanaan awal. *Monitoring* yang dilakukan secara sistematis dan konsisten dapat membantu mendeteksi potensi hambatan atau penyimpangan lebih awal serta memberikan masukan berharga dalam proses evaluasi program [4]. Dalam konteks penelitian ini, sistem *monitoring* difokuskan pada pemantauan sistem VoIP berbasis *web* untuk membantu tim teknis PT. XYZ dalam mengawasi performa, kestabilan layanan, serta mendeteksi aktivitas tidak wajar yang dapat berdampak terhadap kualitas komunikasi dan keamanan data.

*Server* VoIP pada prinsipnya memiliki fungsi yang serupa dengan komputer *server* pada umumnya, yakni sebagai pusat pengelolaan lalu lintas dan aplikasi VoIP dalam suatu jaringan komputer. Dengan dukungan koneksi *internet* atau jaringan Wi-Fi, penggunaan VoIP memberikan efisiensi biaya yang signifikan dibandingkan telepon konvensional. Perangkat yang digunakan pun fleksibel, mulai dari *IP Phone*, *smartphone*, hingga laptop atau komputer. Sebagai media komunikasi suara berkualitas tinggi secara waktu nyata, VoIP sangat memperhatikan aspek *delay*. *Delay* di bawah 150 ms umumnya masih dianggap nyaman, namun *delay* di atas 300 ms dapat mengganggu kualitas percakapan. VoIP umumnya menggunakan protokol *User Datagram Protocol* (UDP) karena sifatnya yang cepat, ringan, dan tidak membebani transmisi, menjadikannya ideal untuk aplikasi *real-time* dengan tuntutan kualitas layanan (QoS) yang tinggi [5]. Seiring dengan meningkatnya kompleksitas infrastruktur dan kebutuhan layanan komunikasi *real-time*, penting bagi perusahaan seperti PT. XYZ untuk memiliki sistem *monitoring* yang mampu memantau performa server VoIP secara efisien, guna menjaga kualitas panggilan serta mendeteksi potensi gangguan secara proaktif.

Web atau yang dikenal sebagai *World Wide Web* (WWW), merupakan halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses diseluruh dunia selama terkoneksi dengan jaringan *internet*. *Website* merupakan salah satu *platform* yang paling sering di akses untuk mencari berbagai informasi dan sarana komunikasi. *Web* berperan sebagai antarmuka yang memudahkan manusia dalam menjelajahi, membaca, serta berinteraksi dengan data yang tersedia. Komponen utama *web* mencakup *frontend*, *backend*, dan infrastruktur. *Frontend* menyajikan tampilan antarmuka pengguna yang menarik dan interaktif, sedangkan *backend* menangani logika aplikasi dan pengelolaan data. Infrastruktur mendukung keberlangsungan operasional *web* melalui penyediaan *server*, *domain*, serta protokol yang menjamin aksesibilitas dan keamanan [6,7]. Dalam konteks sistem *monitoring*, pemanfaatan teknologi berbasis *web* memungkinkan PT. XYZ untuk menyajikan informasi performa server VoIP secara berkala melalui antarmuka yang informatif, sehingga mendukung efisiensi pemantauan, analisis, dan pengambilan keputusan dengan lebih tepat dan terstruktur.

Model *Agile* adalah pendekatan SDLC yang dirancang untuk menghasilkan solusi adaptif dan efisien melalui proses pengembangan yang iteratif. Metode ini mengutamakan fleksibilitas dan kolaborasi intensif untuk merespons kebutuhan pengguna yang dinamis serta perubahan di tengah proyek tanpa mengganggu stabilitas sistem. Dengan membagi pengembangan ke dalam tahapan terstruktur mulai dari perencanaan hingga evaluasi umpan balik, *Agile* memungkinkan tim untuk melakukan perbaikan dan penambahan fitur secara cepat, tepat, dan hemat biaya [8]. Dalam pengembangan sistem *monitoring server VoIP* berbasis *web* di PT. XYZ, pendekatan *Agile* menjadi strategi yang tepat karena memungkinkan pengembangan dilakukan secara bertahap, disesuaikan dengan kebutuhan klien dan kondisi sistem yang dinamis, serta memungkinkan perbaikan berkelanjutan berdasarkan hasil evaluasi tiap iterasi.

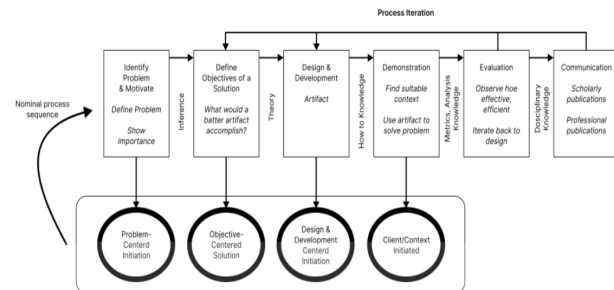
Optimalisasi merupakan suatu proses yang bertujuan untuk mencapai hasil terbaik dari berbagai alternatif solusi yang tersedia, dengan memperhatikan tolok ukur keberhasilan suatu individu, organisasi, maupun sistem. Proses ini dilakukan dengan memaksimalkan fungsi objektif tanpa melanggar batasan yang ada. Optimalisasi memungkinkan peningkatan efisiensi dalam berbagai aspek, seperti peningkatan keuntungan, pengurangan waktu pemrosesan, dan efektivitas penggunaan sumber daya [9]. Dalam konteks sistem *monitoring server VoIP* berbasis *web* di PT. XYZ, proses optimalisasi menjadi penting untuk memastikan bahwa sistem dapat bekerja secara efisien dalam memantau performa *server VoIP*, meminimalkan beban kerja manual, serta meningkatkan kecepatan dalam mendeteksi aktivitas atau gangguan yang tidak biasa demi menjaga kualitas layanan komunikasi Perusahaan.

Kinerja sistem merupakan kemampuan suatu sistem dalam menjalankan tugas atau memproses beban kerja tertentu secara efektif. Kemampuan ini menjadi tolok ukur penting untuk memastikan bahwa sistem dapat beroperasi sesuai harapan dan mendukung tujuan yang diinginkan dalam lingkup penggunaannya [10]. Dalam konteks penelitian ini, sistem *monitoring server VoIP* berbasis *web* diharapkan mampu mendukung peningkatan efisiensi kerja teknisi di PT. XYZ. Dengan tersedianya informasi *monitoring* yang terpusat dan mudah diakses, karyawan dapat lebih cepat mendeteksi aktivitas yang tidak biasa pada *server VoIP*, sehingga dapat mengambil tindakan korektif lebih awal. Hal ini akan berpengaruh langsung terhadap kinerja individu maupun tim, serta menjaga keberlangsungan layanan komunikasi yang andal kepada klien.

## 2. Metode Penelitian

Penelitian ini mengimplementasikan metode *Design Science Research* (DSR) dengan dukungan pendekatan *Agile* untuk merancang dan membangun sistem *monitoring server VoIP* berbasis *web*. Metode DSR dipilih karena berfokus pada penciptaan artefak berupa

produk nyata yang ditujukan untuk menyelesaikan permasalahan riil secara terstruktur dan sistematis. Pendekatan *Agile* digunakan untuk mempercepat proses pengembangan melalui iterasi (*sprint*) yang fleksibel dan berorientasi pada kebutuhan pengguna. Tahapan dari metode *Design Science Research* (DSR) ditunjukkan pada Gambar 1 dibawah ini



Gambar 1. Tahapan Metode *Design Science Research* (DSR)

Berikut adalah penjelasan dari tahap-tahap yang dilakukan pada *Design Science Research* (DSR) dengan model pengembangan *Agile* pada penelitian ini:

### 2.1. Identifikasi Masalah dan Motivasi (*Problem-Centered Initiation*)

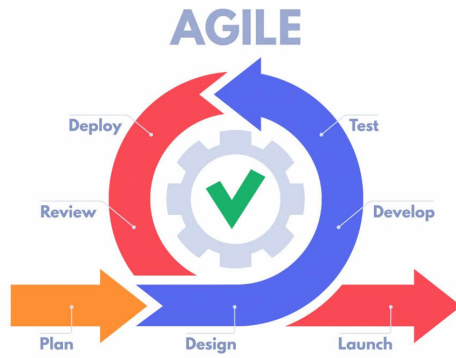
Permasalahan utama yang dihadapi adalah belum adanya sistem *monitoring* yang mampu memberikan pemantauan kondisi *server VoIP* secara menyeluruh, otomatis, dan *realtime*. Hal ini menimbulkan kesulitan dalam mendeteksi *downtime*, beban sumber daya yang berlebihan, serta kualitas panggilan yang menurun. Pada pendekatan *Agile*, tahap ini diwujudkan melalui penyusunan *product backlog* berdasarkan hasil observasi lapangan, studi pustaka, serta wawancara dengan tim operasional.

### 2.2. Definisi Tujuan Solusi (*Objective Centered Solution*)

Tujuan yang ingin dicapai adalah membangun sistem *monitoring server VoIP* berbasis *web* yang dapat menampilkan informasi status *server*, penggunaan *CPU*, *memory*, dan *disk*, *status port*, histori panggilan, hingga perhitungan *Service Level Agreement* (SLA) secara otomatis. Dengan pendekatan *Agile*, tujuan ini diturunkan menjadi *user stories*, misalnya sebagai admin, saya ingin melihat grafik penggunaan CPU secara *realtime* dan juga sebagai operator, saya ingin menerima notifikasi otomatis ketika server mengalami *downtime*.

### 2.3. Desain dan Pengembangan Artefak (*Design & Development*)

Tahap ini merupakan inti dari metode DSR karena menghasilkan artefak utama berupa sistem *monitoring server VoIP* berbasis *web*. Pada penelitian ini digunakan pendekatan *Agile* karena memungkinkan pengembangan sistem secara iteratif dan adaptif. Tahapan metode *Agile* ditunjukkan pada Gambar 2 dibawah ini



Gambar 2. Metode Agile

#### A. Planning (Perencanaan)

Menentukan tujuan utama sistem *monitoring*, kebutuhan pengguna (*administrator server*), serta fitur yang harus tersedia. Dari hasil analisis diperoleh kebutuhan antara lain: perhitungan SLA, *resource usage*, *peak usage*, *threshold breach*, *downtime total*, *call volume*, rata-rata durasi panggilan, *uptime* kontinu, dan persentase penggunaan *port*.

#### B. Design (Perancangan)

Membuat rancangan awal sistem berupa arsitektur, alur data *monitoring*, desain *database*, serta *wireframe dashboard*. Pada tahap ini juga ditentukan integrasi antara *server VoIP* dengan sistem *monitoring* berbasis *web*.

#### C. Development (Pengembangan)

Implementasi kode program secara bertahap sesuai prioritas fitur. Pengembangan dilakukan dengan pembagian *Sprint*, misalnya *Sprint* awal fokus pada pengumpulan data server, *Sprint* berikutnya pada *dashboard*, lalu pengembangan fitur analitik seperti SLA, *call monitoring*, dan notifikasi *threshold*.

#### D. Test (Pengujian)

Setiap hasil pengembangan diuji menggunakan metode *black box testing* untuk memverifikasi fungsionalitas sistem. Misalnya, apakah perhitungan SLA sudah sesuai rumus, apakah grafik *resource usage* muncul dengan benar, atau apakah *alert threshold* terkirim saat beban melebihi batas.

#### E. Deployment (Penerapan)

Sistem yang sudah diuji kemudian dideploy pada server perusahaan sehingga dapat digunakan dalam lingkungan nyata. Proses ini memastikan artefak dapat berjalan sesuai dengan kondisi infrastruktur VoIP PT. XYZ.

#### F. Review (Evaluasi & Umpan Balik)

Tahap terakhir setiap iterasi adalah melakukan *review* bersama pengguna. *Feedback* digunakan untuk memperbaiki sistem di *Sprint* berikutnya. Misalnya, jika *dashboard* dianggap kurang informatif, maka ditambahkan visualisasi tambahan; jika notifikasi terlalu lambat, maka dilakukan optimasi.

Dengan tahapan *Agile* di atas, dihasilkan artefak berupa sistem *monitoring server VoIP* berbasis *web* yang dapat

membantu administrator dalam memantau *resource usage*, menganalisis *call*, menghitung SLA, dan mendeteksi *downtime*. Sistem ini mendukung tujuan penelitian untuk mengoptimalkan kinerja server VoIP secara efektif dan efisien.

#### 2.4. Demonstrasi (*Client/Context Initiated*)

Artefak diuji langsung pada infrastruktur *server VoIP* di PT. XYZ. Sistem dijalankan dalam kondisi operasional nyata untuk memastikan keakuratan data *monitoring*. Dalam pendekatan *Agile*, setiap *sprint* diakhiri dengan *sprint review*, di mana tim operasional mencoba fitur yang telah selesai dibangun dan memberikan umpan balik untuk perbaikan pada *sprint* berikutnya.

#### 2.5. Evaluasi (*Metrics, Analysis, Knowledge*)

Tahap evaluasi sistem *monitoring server VoIP* dilakukan melalui beberapa metode yang terstruktur. Pertama, dilakukan uji fungsionalitas sistem menggunakan metode *black box testing* untuk memastikan seluruh fitur berjalan sesuai harapan. Kedua, dilakukan pengukuran kecepatan pembaruan data *monitoring* yang ditetapkan pada interval setiap satu menit.

Selain itu, evaluasi mencakup analisis mendalam terhadap berbagai indikator kinerja, seperti perhitungan *Service Level Agreement (SLA)*, durasi *downtime*, rata-rata penggunaan sumber daya (*resource usage*), volume panggilan, hingga pemantauan *uptime* secara kontinu. Dalam konteks pendekatan *Agile*, seluruh proses evaluasi ini dilakukan sejalan dengan tahapan *sprint retrospective*, yang berfungsi sebagai wadah refleksi atas kinerja *sprint*, identifikasi area perbaikan, serta penentuan rencana pengembangan pada iterasi berikutnya.

#### 2.6. Komunikasi (*Disciplinary Knowledge*)

Hasil penelitian dikomunikasikan dalam bentuk laporan akademis (skripsi), dokumentasi teknis sistem, serta penyampaian hasil kepada pihak PT. XYZ sebagai pengguna. Dalam pendekatan *Agile*, komunikasi juga dilakukan melalui *release notes* pada setiap *sprint*, yang menjelaskan fitur baru, perbaikan, atau pembaruan sistem.

Dengan demikian, penerapan metode DSR yang dikombinasikan dengan *Agile* menghasilkan sebuah artefak berupa sistem *monitoring server VoIP* berbasis *web* yang dapat menyajikan data secara *real-time*, akurat, dan efisien, serta terbukti membantu tim teknis dalam meningkatkan efektivitas pemantauan dan optimalisasi kinerja sistem VoIP.

### 3. Hasil dan Pembahasan

Pada bagian ini akan dijelaskan hasil penelitian berdasarkan tahapan DSR dan pengembangan sistem yang digunakan.

Analisis sistem merupakan langkah awal yang krusial dalam pengembangan sistem informasi, di mana dilakukan pengkajian dan penilaian terhadap sistem

yang sedang digunakan. Tujuannya adalah untuk memahami cara kerja sistem tersebut, menemukan permasalahan yang muncul, serta merumuskan kebutuhan untuk sistem baru atau perbaikan dari sistem yang telah ada. Hasil dari proses ini menjadi landasan penting dalam perancangan dan pembangunan sistem selanjutnya [11].

### 3.1. Analisa Sistem Usulan

Analisis sistem yang diusulkan bertujuan untuk membangun solusi *monitoring server* VoIP berbasis *web* yang dapat memberikan kemudahan bagi administrator dalam melakukan pengawasan secara terpusat di PT. XYZ. Sistem ini dirancang untuk menggantikan metode lama yang membutuhkan pengecekan manual pada masing-masing *server*, yang selama ini memakan waktu, tenaga, serta berpotensi terjadi kesalahan.

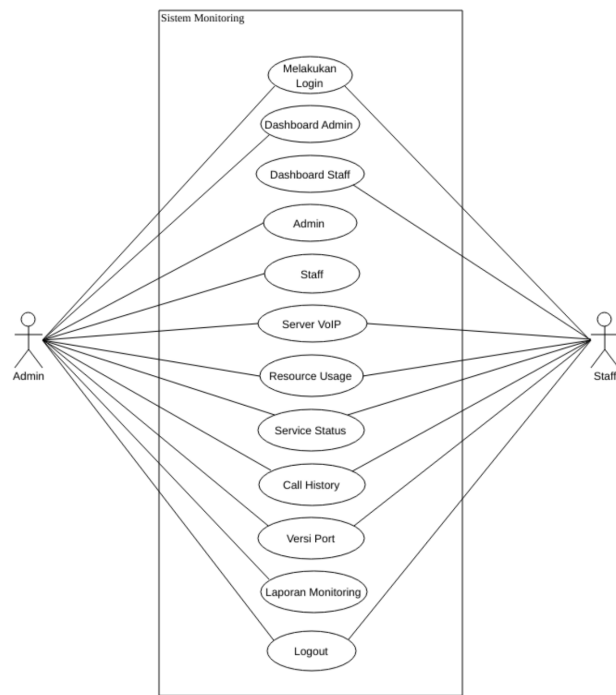
Melalui sistem yang dikembangkan, *administrator* dapat memantau berbagai kondisi dan performa *server* secara *visual* pada satu *platform web*. Informasi yang ditampilkan selalu diperbarui secara berkala sehingga memberikan gambaran kondisi terbaru dari masing-masing *server* yang terhubung. Selain itu, sistem dilengkapi dengan fitur pemberitahuan otomatis yang akan muncul secara langsung pada halaman *dashboard* ketika ditemukan kondisi yang memerlukan perhatian segera. Notifikasi ini dirancang untuk memberikan tanda peringatan yang mudah terlihat oleh administrator agar tindakan yang diperlukan dapat segera dilakukan tanpa keterlambatan.

Dengan sistem yang terpusat ini, administrator tidak perlu lagi melakukan pengecekan secara terpisah pada masing-masing *server*, sehingga meningkatkan efisiensi pengawasan dan meminimalkan potensi keterlambatan dalam mendeteksi gangguan. Sistem ini juga mendukung visualisasi data yang mudah dipahami, sehingga mempermudah proses pengambilan keputusan yang cepat dan tepat dalam menjaga kualitas layanan PT. XYZ kepada klien.

### 3.2. Use Case Diagram

*Use Case Diagram* adalah model visual yang menggambarkan perilaku sistem, alur proses bisnis, dan interaksi antara aktor dengan sistem yang akan dikembangkan. Diagram ini berfungsi untuk mengidentifikasi seluruh fungsionalitas di dalam sistem sekaligus menentukan pengguna atau entitas mana saja yang berhak mengakses fungsi-fungsi tersebut guna memastikan keselarasan kebutuhan fungsional [12].

*Use Case Diagram* yang digunakan ditunjukkan pada Gambar 3 dibawah ini



Gambar 3 Use Case Diagram

### 3.3. Metode Design Science Research (DSR)

Penelitian ini mengadopsi metode *Design Science Research* (DSR) untuk merancang dan membangun artefak berupa sistem *monitoring server* VoIP berbasis *web*. Pemilihan DSR didasarkan pada fokusnya untuk menghasilkan solusi teknologi inovatif yang menjawab permasalahan praktis di dunia nyata. Permasalahan yang diidentifikasi adalah tidak adanya sistem pemantauan terpusat yang mampu menyediakan data komprehensif, otomatis, dan *real-time* mengenai kinerja sumber daya *server*, status layanan, riwayat panggilan (*call history*), status ketersediaan port, serta perhitungan *Service Level Agreement* (SLA).

Proses *Design Science Research* (DSR) dalam penelitian ini diimplementasikan melalui tiga tahapan sistematis yang saling terintegrasi. Tahap pertama diawali dengan identifikasi masalah dan motivasi, yang dilakukan melalui studi literatur, observasi langsung pada infrastruktur jaringan yang sedang berjalan, serta analisis kebutuhan pengguna melalui wawancara mendalam dengan tim teknis dan operasional. Selanjutnya, penelitian memasuki tahap perancangan dan pengembangan artefak, di mana dilakukan penyusunan arsitektur serta pengembangan komponen-komponen sistem berdasarkan kebutuhan yang telah didefinisikan sebelumnya. Terakhir, dilakukan tahap evaluasi dan validasi dengan cara menguji artefak di dalam lingkungan operasional nyata untuk memastikan fungsionalitas dan efektivitas sistem yang telah dibangun.

Artefak yang dihasilkan dalam penelitian ini merupakan sebuah sistem terdistribusi yang terdiri dari tiga komponen utama yang saling berinteraksi secara terintegrasi. Komponen pertama adalah *Agent Python*, yaitu sebuah skrip yang diimplementasikan pada setiap

*server* VoIP untuk mengumpulkan data secara periodik setiap 60 detik. Data telemetri yang dikumpulkan meliputi utilisasi CPU, memori, dan disk, serta status konektivitas *server* (*up/down*) dan status keterbukaan port esensial (*open/closed*) yang kemudian dikirimkan ke *server monitoring* secara terjadwal.

Komponen kedua adalah *Backend Laravel* (PHP) yang berfungsi sebagai inti sistem sekaligus *endpoint* penerima data dari seluruh agent. *Backend* ini bertanggung jawab untuk melakukan parsing, validasi, dan penyimpanan data ke dalam basis data, serta menyediakan serangkaian API untuk kebutuhan visualisasi. Selain itu, sistem ini secara periodik mengambil data riwayat panggilan dari tabel *cdoutput* pada basis data *server VoIP* untuk dianalisis lebih lanjut. Terakhir, terdapat *Dashboard* Visualisasi berbasis *web* yang dikembangkan menggunakan *Laravel* dan *JavaScript*. Antarmuka ini menyajikan seluruh data monitoring dalam bentuk grafik, tabel, dan indikator visual interaktif, sehingga memudahkan administrator dalam menganalisis tren performa serta mengidentifikasi anomali secara efisien.

Dibawah ini adalah perhitungan yang digunakan pada sistem monitoring

*Service Level Agreement* (SLA) Salah satu logika inti dalam sistem ini adalah perhitungan SLA berdasarkan status *server* yang dikirimkan oleh agent secara berkala. SLA dihitung sebagai persentase waktu *server* dalam kondisi *Up* dibandingkan dengan total waktu pemantauan.

Rumus:

$$SLA = \frac{Uptime}{Uptime + Downtime} \times 100\% \quad (1)$$

Contoh Perhitungan SLA Misalkan dalam 1 hari (1440 menit), *server* mengirimkan status setiap 1 menit, sehingga total 1440 titik data. Dari data yang tercatat: Jumlah status "*Up*": 1200 titik (setara  $1200 \times 1 = 1200$  menit)

Jumlah status "*Down*": 240 titik (setara  $240 \times 1 = 240$  menit)

Maka perhitungannya:  $SLA = (1200 / (1200 + 240)) \times 100 = (1200 / 1440) \times 100 \approx 83.33\%$

Rata-rata Pemakaian Resource (CPU, Memori, Disk) Selain SLA, sistem juga menghitung rata-rata pemakaian CPU, Memory, dan Disk per jam.

Rumus:

$$Rata - rata Usage = \frac{\sum_{i=1}^n Usage_i}{n} \quad (2)$$

Contoh Perhitungan CPU Usage Rata-Rata:

Data CPU selama 1 jam (setiap 1 menit): [35.2, 41.0, 32.8, 45.3, ..., 39.5]

Rata-rata CPU =  $(35.2 + 41.0 + 32.8 + 45.3 + \dots + 39.5) / \text{jumlah data}$

Contoh hasil: = 39.1%

*Peak Usage* (Penggunaan Tertinggi) adalah menentukan penggunaan tertinggi selama periode tertentu.

Rumus :

$$Peak Usage = (Usage_1, Usage_2, \dots, Usage_n) \quad (3)$$

Contoh Perhitungan

Data CPU: [35.2, 41.0, 32.8, 45.3, 39.5]

Peak CPU = 45.3%

*Threshold Breach Count* (Jumlah Melebihi Ambang Batas) yaitu menghitung berapa kali pemakaian resource melewati ambang batas tertentu (misalnya CPU > 80%)

Rumus:

$$Threshold Breach Count = \sum_{i=1}^n [Usage_i > Threshold] \quad (4)$$

Contoh: Data CPU: [60, 83, 89, 75, 95]

Count = 3 (yaitu 83, 89, 95)

*Downtime Total* adalah Menghitung total downtime dari jumlah status *Down* dan interval.

Rumus:

$$Total Downtime = Jumlah Status Down \times Interval \quad (5)$$

Contoh: Status *Down* = 25 kali, Interval = 1 menit

Total Downtime = 25 menit = 0.25 jam

*Call Duration Average* adalah menghitung rata-rata durasi semua panggilan.

Rumus:

$$Call Duration average = \frac{\sum_{i=1}^n Durasi_i}{n} \quad (6)$$

Contoh: Durasi: [300, 200, 400, 100]

Rata-rata =  $1000 / 4 = 250$  detik.

### 3.4 Pengujian Sistem

Tahap pengujian dilakukan untuk memastikan bahwa artefak berupa sistem *monitoring server VoIP* berbasis *web* dapat berjalan sesuai dengan kebutuhan pengguna, tujuan penelitian, serta memenuhi kriteria fungsional dan non-fungsional. Pengujian dilakukan melalui beberapa pendekatan sebagai berikut:

Pengujian fungsional (*Black Box Testing*) pada sistem ini dilakukan dengan menggunakan metode *black box testing*, yang bertujuan untuk memastikan setiap fitur bekerja sesuai dengan ekspektasi tanpa meninjau kode program di dalamnya. Proses pengujian ini mencakup beberapa aspek krusial, di antaranya adalah pengujian fitur *monitoring resource usage* untuk memverifikasi apakah data *CPU*, *memory*, dan *disk* muncul secara *real-time* di halaman *dashboard*. Selain itu, dilakukan pengujian pada fitur *monitoring status port* dan *server* guna memastikan status *port* serta kondisi *server* dapat ditampilkan dengan benar.

Lebih lanjut, pengujian dilakukan terhadap fitur *call monitoring* untuk memastikan histori panggilan dapat tersimpan dan divisualisasikan dengan tepat, serta fitur perhitungan SLA guna memverifikasi akurasi data *uptime* dan *downtime* sesuai dengan rumus yang telah

ditetapkan. Terakhir, pengujian juga mencakup fitur notifikasi untuk menguji kemampuan sistem dalam mengirimkan peringatan otomatis saat terjadi *downtime* atau ketika pemakaian sumber daya melebihi ambang batas (*threshold breach*).

Selain pengujian fungsional, dilakukan pula pengujian performa sistem untuk menjamin keandalan artefak yang dikembangkan. Pengujian ini difokuskan pada tiga aspek utama, yaitu kecepatan pembaruan data untuk memastikan sistem mampu memperbarui informasi *monitoring* setiap 1 menit sesuai dengan interval yang ditetapkan pada *agent* Python. Selanjutnya, dilakukan pengukuran terhadap respons dashboard guna menilai kecepatan sistem dalam menampilkan visualisasi grafik dan tabel secara optimal. Terakhir, pengujian mencakup aspek ketepatan data yang dilakukan dengan membandingkan hasil pemantauan sistem terhadap data asli pada server, seperti melalui perintah *top* atau *htop*, untuk memastikan akurasi informasi yang disajikan.

Pengujian *User Acceptance* (UAT) dilakukan dengan melibatkan administrator server PT. XYZ sebagai pengguna langsung untuk memastikan sistem memenuhi standar operasional. Dalam tahap ini, dilaksanakan *Sprint Review* di setiap akhir iterasi, di mana pengguna mencoba secara langsung fitur-fitur yang telah selesai dibangun. Selain itu, dikumpulkan *Feedback* Pengguna yang mencakup masukan terkait kemudahan penggunaan, kecepatan akses, kelengkapan data, serta aspek visual pada tampilan dashboard. Sebagai tahap akhir, dilakukan evaluasi Penerimaan Sistem, di mana artefak dinyatakan layak untuk digunakan apabila mayoritas kebutuhan pengguna telah terpenuhi dengan nilai ambang batas  $\geq 80\%$  berdasarkan hasil *checklist User Acceptance Testing* (UAT).

Dokumentasi hasil pengujian, setiap pengujian dicatat dalam Tabel 1 dibawah ini:

Tabel 1 Hasil Pengujian Sistem

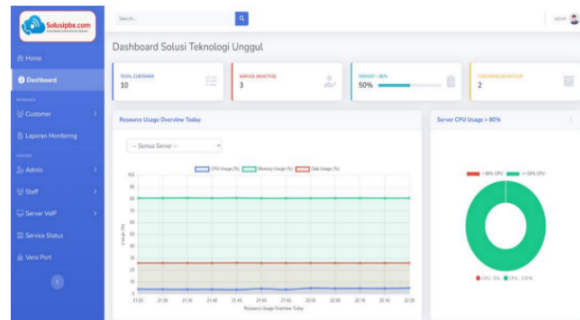
No	Fitur yang diuji	Hasil yang diharapkan	Hasil aktual	Status
1	Menampilkan penggunaan CPU	Grafik muncul <i>realtime</i>	Sesuai	Valid
2	Notifikasi <i>downtime</i> server	Notifikasi terkirim otomatis	Sesuai	Valid
3	Perhitungan SLA	Persentase uptime sesuai rumus	Sesuai	Valid

### 3.5 Implementasi Model ke Aplikasi Web

Implementasi model ke aplikasi *web* adalah proses menerapkan rancangan atau konsep yang telah dibuat seperti model metode, algoritma, atau *system* ke dalam bentuk aplikasi berbasis *web* yang dapat digunakan secara nyata. Tahapan ini mencakup penerjemahan kebutuhan dan desain ke dalam kode program, integrasi

dengan *database*, serta pengujian agar sistem berjalan sesuai tujuan.

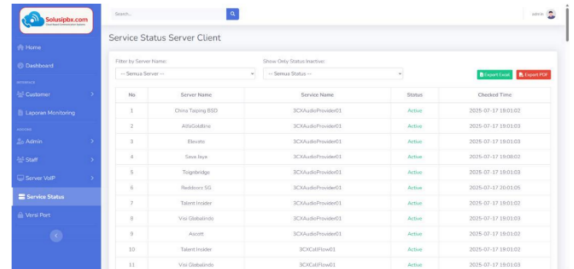
Dengan implementasi ini, model yang sebelumnya hanya berupa konsep atau perhitungan teoretis dapat diakses dan dimanfaatkan oleh pengguna melalui antarmuka web secara lebih praktis, interaktif, dan *real-time*.



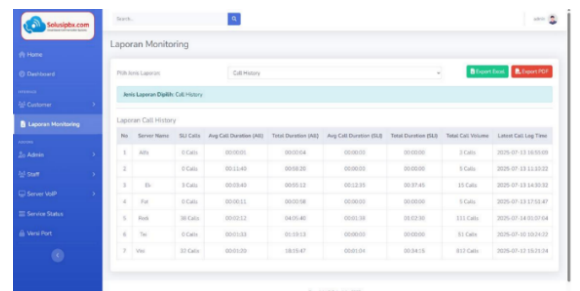
Gambar 4. Halaman Dashboard



Gambar 5. Halaman Server VoIP



Gambar 6. Halaman Service Status



Gambar 7. Halaman Laporan Monitoring

## 4. Kesimpulan

Berdasarkan penelitian mengenai implementasi metode *Design Science Research* (DSR) di PT. XYZ, dapat disimpulkan bahwa sistem *monitoring server VoIP* berbasis *web* telah berhasil dirancang dan diimplementasikan untuk meningkatkan efisiensi pemantauan pada seluruh klien perusahaan. Sistem yang dibangun menggunakan Laravel sebagai *backend*

dan *dashboard* serta *agent Python* ini mampu melakukan pengiriman data otomatis seperti penggunaan CPU, memori, disk, serta *status port* dan *server* setiap lima menit, sehingga proses *monitoring* menjadi lebih terpusat. Selain itu, sistem ini menyajikan informasi performa secara terkini melalui grafik, tabel, dan indikator visual yang memudahkan tim teknis dalam pengambilan keputusan cepat saat terjadi gangguan, didukung dengan perhitungan SLA, rata-rata penggunaan *resource*, serta *downtime* sebagai indikator evaluasi. Terakhir, ketersediaan fitur pendeteksian aktivitas mencurigakan, seperti anomali *port* dan lonjakan penggunaan sumber daya, sangat membantu dalam mendeteksi potensi gangguan atau penyalahgunaan sejak dini, yang pada akhirnya meningkatkan keamanan serta stabilitas layanan komunikasi VoIP secara keseluruhan.

### Ucapan Terimakasih

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Allah SWT, kedua orang tua, saudara, dan terkhusus kepada kedua dosen pembimbing, serta seluruh dosen prodi Teknik Informatika Universitas Pamulang dan sahabat-sahabat penulis.

### Daftar Rujukan

- [1] Tribunnews, 2024. Telepon Kantor Tak Lagi Relevan, SolusiPBX Hadir Menjawab Kebutuhan Komunikasi Bisnis. [Online] (Updated 27 Jan 2026) Tersedia:<https://www.tribunnews.com/techno/7783548/telepon-kantor-tak-lagi-relevan-solusipbx-hadir-menjawab-kebutuhan-komunikasi-bisnis>. [Accessed 22 April 2026]
- [2] Ina Magdalena, Istiqomah, & Rahma Yunita. (2024). IMPLEMENTASI, EVALUASI, SUMATIF & DIFUSI INOVASI. Sindoro: Cendikia Pendidikan, 2(10), 51–60. <https://doi.org/10.9644/sindoro.v2i10.192>.
- [3] Brocke, J., Hevner, A., & Maedche, A. (2020). Design Science Research. Cases. Springer International Publishing. <https://books.google.co.id/books?id=qdXDwAAQBAJ>.
- [4] Elex Sarmigi, M. S., Muhammad Alfian, M. P., Ravico, M. H., Tiara, M. S., Lia Angela, M. P., Fatnan Asbupel, M. P., Dr. Ahmad Fikri, M. P., & Adab, P. (2025). Instrumen Penelitian Dan Monitoring & Evaluasi (Monev) Di Perguruan Tinggi. Penerbit Adab. [https://books.google.co.id/books?id=o\\_P4EAAAQBAJ](https://books.google.co.id/books?id=o_P4EAAAQBAJ)
- [5] Maryuli Darmawan, S. P. M. E. S. A. M. E. (2021). Teknologi Layanan Jaringan SMK/MAK Kelas XII. Gramedia Widiasarana indonesia. <https://books.google.co.id/books?id=vwWEAAAQBAJ>
- [6] Khana, K. W., & Damayanti, A. (2022). Rancang Bangun Sistem Informasi Training Online Berbasis Web (Studi Kasus: Upt Lab. Komputer STMIK Prabumulih). *Jurnal Fasilkom*, 12(3), 172–178 <https://ejurnal.umri.ac.id/index.php/JIK/article/view/3542/2138>
- [7] Sisehaputra, B., Apriyanto, A., Putra, P. M., Nur, M., Wahyuni, E. D., Ardiada, I. M. D., Najaf, A. R. E., Haryono, H., Sepriano, S., & Sari, I. K. (2025). Buku Ajar Pemrograman Web. PT. Sonpedia Publishing Indonesia. <https://books.google.co.id/books?id=Bnw9EQAAQBAJ>
- [8] Wahyu Mustika Aji, Achmad Syahmi Rasendriya, Alya Putri Salsabila, Keysha Maulina Halimi, Muhammad Nasir, & Aditya Wicaksono. (2025). Pengembangan Website Spherevent untuk Manajemen Event Berbasis Web dengan Pendekatan Agile. *Jurnal Fasilkom*, 15(1), 131–137. <https://doi.org/10.37859/jf.v15i1.9116>
- [9] Mayasari, R., Febriantoko, J., Masnila, N., Ariyanti, I., Jumeilah, F. S., Putra, M. S. G. P., & Tunggal, T. (2024). Buku Referensi Akuntabilitas Alokasi Dana Desa dan Pengelolaan Keuangan Desa. Penerbit NEM <https://books.google.co.id/books?id=zqL6EAAAQBAJ>
- [10] Kharisma, L. P. I., Miftachurohmah, N., Jannah, U. M., Wahyudi, F., Datya, A. I., Syamil, A., Efitra, S. K. M. K., & Juansa, A. (2023). ANALISIS & PERANCANGAN SISTEM : Berbasis Studi Kasus. PT. Sonpedia Publishing Indonesia. <https://books.google.co.id/books?id=DCnBEAAAQBAJ>
- [11] Nirsal, N., Roji, M. F., Syam, S., Selviana, R., Widiyanto, A., Gunawan, I. M. A. O., Arsana, I. N. A., Sutoyo, M. N., Kurniadi, W., Sari, I. K., & others. (2025). Buku ajar Analisis dan Perancangan Sistem. PT. Sonpedia Publishing Indonesia. <https://books.google.co.id/books?id=5UBcEQAAQBAJ>
- [12] Yulia, Y., Anggraini, N. G., & Lisnawanty, L. (2020). Aplikasi Perhitungan PPH 21 Dan PPH 26 Karyawan Berbasis Web. *Jurnal Sistem Informasi Akuntansi*, 1(2). <https://doi.org/10.31294/justian.v1i2.291>