

SISTEM REMOTE DESKTOP BERBASIS WEB UNTUK MONITORING DAN KENDALI MULTI-KLIEN DI LABORATORIUM KOMPUTER

Antonius Saputra¹, Lusiana Citra Dewi², Alvin Chandra³, Indra Dwi Rianto⁴, Renny Triana⁵

^{1,2,3,4} Program Studi Teknik Informatika Universitas Bina Nusantara, Jakarta 11480, Indonesia

⁵ Program Studi Sistem Informasi Universitas Bina Nusantara, Jakarta 11480, Indonesia

Keywords:

manajemen remote desktop; aplikasi web monitoring secara real-time; *Wake-on-LAN*, *blocking* perangkat USB.

Correspondent Email:

asaputra@binus.edu

Abstrak. Paper ini mempresentasikan perancangan dan implementasi sistem manajemen remote desktop berbasis web menggunakan *framework* Node.js. Sistem dikembangkan untuk mengatasi keterbatasan *tools* akses jarak jauh tradisional dan memungkinkan administrator memantau serta mengendalikan banyak komputer klien dalam jaringan lokal secara real-time tanpa instalasi perangkat lunak di sisi klien. Fitur utama yang disediakan meliputi pemantauan multi-klien, eksekusi perintah jarak jauh, *Wake-on-LAN*, pengambilan tangkapan layar, serta pemblokiran perangkat input/USB. Proses pengembangan menggunakan Dynamic Systems Development Method (DSDM) yang memanfaatkan komunikasi asinkron dan *event-driven programming* untuk meningkatkan responsivitas dan skalabilitas sistem. Berdasarkan pengujian di laboratorium komputer, sistem menunjukkan *latency* rendah (kurang dari 3 detik), *reliability* tinggi, dan mendapatkan umpan balik positif dari 23 administrator. Secara keseluruhan, solusi ini memberikan alternatif yang fleksibel dan ringan untuk manajemen *remote desktop* yang aman, khususnya pada lingkungan pendidikan dan institusi.



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. This paper presents the design and implementation of a web-based remote desktop management system using the Node.js framework. The system is developed to address the limitations of traditional remote access tools and enables administrators to monitor and control multiple client computers within a local network in real time without installing software on the client side. The main features provided include multi-client monitoring, remote command execution, *Wake-on-LAN*, screenshot capture, and input/USB device blocking. The development process adopts the Dynamic Systems Development Method (DSDM), which leverages asynchronous communication and event-driven programming to improve system responsiveness and scalability. Based on testing in a computer laboratory environment, the system demonstrates low latency (less than 3 seconds), high reliability, and receives positive feedback from 23 administrators. Overall, this solution offers a flexible and lightweight alternative for secure remote desktop management, particularly suitable for educational and institutional settings.

1. PENDAHULUAN

Dalam ekosistem teknologi informasi, khususnya pada institusi pendidikan, laboratorium riset, dan lingkungan administrasi, kebutuhan akan pengelolaan banyak komputer klien secara terpusat dan real-time menjadi

semakin penting. Alat akses jarak jauh tradisional seperti TeamViewer, AnyDesk, dan VNC memang umum digunakan, tetapi masih memiliki beberapa keterbatasan. Perangkat lunak tersebut biasanya mengharuskan instalasi di sisi klien, mengonsumsi sumber daya sistem

yang relatif tinggi, dan tidak mendukung kendali banyak klien secara terintegrasi. Keterbatasan ini mengurangi fleksibilitas dan membuat proses pemeliharaan menjadi lebih sulit, terutama pada lingkungan dengan beragam sistem operasi.

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan sistem *remote desktop* berbasis web yang ringan yang dikembangkan dengan menggunakan Node.js dan Socket.IO. Sistem ini memungkinkan administrator untuk memantau dan mengendalikan banyak komputer klien dalam jaringan lokal secara *real-time* tanpa perlu menginstal perangkat lunak di sisi klien. Fitur utama yang disediakan meliputi pemantauan multi-klien, eksekusi perintah jarak jauh, Wake-on-LAN, pengambilan tangkapan layar, pemblokiran perangkat *input/USB*, serta antarmuka web yang responsif sehingga mudah digunakan.

Pentingnya arsitektur berlatensi rendah dan skalabel dalam sistem pemantauan jarak jauh telah dibahas dalam berbagai penelitian sebelumnya. Kopetz [2] menekankan bahwa perilaku deterministik dan toleransi kesalahan merupakan prinsip dasar dalam perancangan sistem terdistribusi *real-time*. Rajkumar et al. [3] juga menjelaskan peran *cyber-physical system* dalam memungkinkan pengendalian yang responsif pada perangkat-perangkat yang saling terhubung dalam jaringan. Pada implementasi praktis, Gupta et al. [1] menunjukkan bagaimana sistem pemantauan jarak jauh dapat diterapkan di bidang pertanian melalui mekanisme akuisisi data dan kendali secara *real-time*. Studi lain pada sistem *remote desktop* berbasis TCP menunjukkan bahwa pengaturan ulang mekanisme komunikasi dapat menurunkan waktu respons interaksi hingga berada pada batas yang masih nyaman digunakan pengguna, sehingga menegaskan pentingnya optimasi jalur komunikasi pada aplikasi kendali jarak jauh [16].

Keamanan dan skalabilitas menjadi aspek penting dalam pengelolaan sistem jarak jauh. Laghari et al. [5] menyoroti peningkatan risiko keamanan siber pada lingkungan IoT dengan menekankan pentingnya komunikasi terenkripsi dan protokol autentikasi yang kuat. Berdasarkan hal tersebut, Saleem et al. [4] menunjukkan bagaimana arsitektur sistem yang modular dan aman dapat diterapkan secara

efektif, sebagaimana ditunjukkan pada sistem manajemen energi cerdas yang mereka kembangkan. Sebagai perspektif tambahan, Paris et al. [8] memperlihatkan bahwa integrasi protokol SSL/TLS ke dalam sistem IoT berbasis MQTT dapat meningkatkan keamanan komunikasi secara signifikan. Selain itu, evaluasi kinerja akses tampilan jarak jauh pada skenario *mobile cloud* memperlihatkan bahwa variasi algoritma kompresi dan kondisi jaringan berpengaruh langsung terhadap konsumsi daya dan kualitas pengalaman pengguna, sehingga desain protokol harus mempertimbangkan keseimbangan antara latensi [17].

Sistem ini dikembangkan dengan menggunakan Dynamic Systems Development Method (DSDM), yang mengombinasikan pemrograman berbasis peristiwa (*event-driven programming*) dan komunikasi asinkron untuk mencapai kinerja berlatensi rendah serta skalabilitas yang tinggi. Kontribusi utama penelitian ini adalah: (1) merancang arsitektur manajemen *remote desktop* berbasis web yang bersifat *agentless*; (2) mengintegrasikan berbagai fitur kendali *real-time* ke dalam satu antarmuka terpadu; dan (3) mengevaluasi kinerja sistem serta pengalaman pengguna pada lingkungan laboratorium yang sesungguhnya.

2. TINJAUAN PUSTAKA

A. Aplikasi Sistem Pemantauan Jarak Jauh

Sistem pemantauan jarak jauh menunjukkan fleksibilitas tinggi dengan penerapan di berbagai domain industri, pertanian, dan pendidikan. Gupta et al. [1] mengembangkan sistem berbasis web untuk lingkungan pertanian dengan menitikberatkan pada pengumpulan data *real-time* dan kendali jarak jauh. Penelitian mereka menunjukkan bahwa sistem pemantauan jarak jauh dapat meningkatkan efisiensi operasional pada lingkungan terdistribusi. Selain pada lingkungan pertanian, penelitian lain [15] menunjukkan bahwa pemantauan jarak jauh berbasis IoT juga banyak diterapkan pada domain rumah tangga, misalnya untuk memonitor dan mengendalikan konsumsi daya listrik perangkat rumah secara *real-time* melalui koneksi internet. Konsep ini menjadi referensi penting bagi pengembangan sistem pemantauan dan kendali terpusat di lingkungan institusi seperti laboratorium dan ruang kelas.

Dalam ranah pendidikan, Burunkaya et al. [6] mengembangkan sistem kelas cerdas berbasis IoT yang memungkinkan pemantauan dan pengendalian terpusat berbagai perangkat kelas melalui platform berbasis web yang dapat diskalakan. Zhu [10] juga mengeksplorasi sistem kelas berbasis IoT dengan fokus pada integrasi perangkat dan umpan balik real-time. Sistem-sistem tersebut menunjukkan potensi pemantauan jarak jauh di lingkungan akademik. Namun, banyak di antaranya masih bergantung pada konfigurasi yang spesifik perangkat keras. Pada penelitian ini, sistem yang diusulkan berfokus pada independensi platform dan penerapan tanpa agen (agentless), sehingga lebih adaptif dan mudah diimplementasikan pada berbagai lingkungan. Pendekatan serupa juga terlihat pada pengembangan sistem remote desktop berbasis web dengan arsitektur peer-to-peer yang berupaya mengurangi ketergantungan pada klien khusus dan memaksimalkan akses melalui peramban, meskipun fokusnya lebih pada aksesibilitas individu dibandingkan manajemen multi-klien di lingkungan laboratorium [19].

B. Keterbatasan Komparatif Alat yang Ada

Perangkat lunak remote desktop populer seperti TightVNC dan TeamViewer memang menyediakan fitur akses jarak jauh dasar, tetapi masih memiliki sejumlah keterbatasan. TightVNC bersifat ringan dan sumber terbuka, namun memerlukan konfigurasi manual dan tidak mendukung pemantauan banyak klien secara bersamaan. TeamViewer menawarkan antarmuka yang ramah pengguna dan dukungan lintas platform, tetapi bergantung pada infrastruktur proprietari serta membutuhkan instalasi di sisi klien, sehingga kurang praktis untuk penerapan skala besar di lingkungan laboratorium atau institusi.

Keterbatasan tersebut juga terkonfirmasi pada studi kelayakan awal dalam penelitian ini, di mana para administrator melaporkan kesulitan dalam mengelola banyak komputer secara efisien dengan alat yang ada. Beberapa penelitian lain mengembangkan aplikasi remote desktop manager yang menghubungkan perangkat seluler dengan komputer pribadi dan menyediakan fitur seperti screen sharing, file transfer, Wake-on-LAN, dan kendali input jarak jauh [18]. Namun, hasil analisis performa menunjukkan bahwa meskipun fitur yang

disediakan cukup lengkap, implementasi tersebut masih menghadapi tantangan terkait latency dan kebutuhan konfigurasi di sisi klien [18]. Berdasarkan celah tersebut, sistem yang diusulkan dirancang dengan arsitektur sepenuhnya berbasis web dan agentless yang mendukung pemantauan multi-klien secara real-time, kendali jarak jauh, dan fungsi administratif tanpa instalasi tambahan maupun ketergantungan eksternal di sisi klien.

C. Perancangan Sistem Real-Time dan Pertimbangan Kinerja

Perancangan sistem real-time penting untuk memastikan responsivitas dan keandalan pada aplikasi pemantauan jarak jauh. Kopetz [2] menjelaskan prinsip-prinsip dasar pada sistem tertanam terdistribusi, termasuk determinisme dan toleransi kesalahan. Rajkumar et al. [3] melanjutkan gagasan ini pada sistem siber-fisik dengan menekankan pentingnya komunikasi berlatensi rendah dan keandalan sistem yang tinggi.

Loja dan Maita [11] melakukan perbandingan kinerja kerangka kerja web dan menyimpulkan bahwa Node.js memiliki efisiensi temporal dan skalabilitas yang lebih baik. Hal ini mendukung arsitektur yang digunakan dalam penelitian ini, di mana Node.js digunakan untuk menangani banyak koneksi klien dan komunikasi data real-time.

D. Teknologi Komunikasi untuk Kendali Jarak Jauh

Protokol komunikasi yang efektif penting untuk mendukung kendali dan pemantauan real-time. Nakazato et al. [7] mengevaluasi kinerja WebRTC untuk kolaborasi jarak jauh dan menunjukkan bahwa kendali berbasis peramban dimungkinkan pada lingkungan seluler. Namun, sistem mereka tidak mencakup fitur lanjutan seperti Wake-on-LAN dan pemblokiran USB, yang sudah diimplementasikan dalam penelitian ini.

Paris et al. [8] menunjukkan bahwa MQTT yang dipadukan dengan protokol SSL/TLS efektif dalam sistem IoT, serta membuktikan bahwa arsitektur berbasis peristiwa efisien dan aman pada lingkungan dengan sumber daya terbatas. Deshpande et al. [9] juga memvalidasi penggunaan Socket.IO untuk komunikasi real-time, yang turut digunakan dalam

penelitian ini untuk memastikan interaksi berlatensi rendah antara server dan klien.

E. Keamanan dan Integritas dalam Arsitektur Pemantauan

Keamanan menjadi aspek penting dalam sistem pemantauan jarak jauh, khususnya pada lingkungan institusional dan multi-pengguna. Laghari et al. [5] mengidentifikasi tantangan keamanan siber pada aplikasi IoT dan merekomendasikan komunikasi terenkripsi serta mekanisme autentikasi yang kuat. Saleem et al. [4] juga menjelaskan pentingnya arsitektur yang aman dan modular pada sistem pemantauan energi berbasis IoT.

Maurya et al. [13] mengusulkan kerangka keamanan berbasis blockchain untuk jaringan IoT yang menawarkan pendekatan terdesentralisasi terhadap integritas data dan kontrol akses; hal ini dapat dipertimbangkan sebagai pengembangan di masa depan bagi sistem yang diusulkan. Chakraborty dan Kundan [12] juga memaparkan arsitektur sistem pemantauan modern yang berfokus pada modularitas, skalabilitas, dan interoperabilitas, yang juga diterapkan dalam sistem yang dikembangkan pada penelitian ini.

Ulasan pustaka ini menjadi landasan teoretis dan teknis bagi sistem pada penelitian ini, serta memandu perancangan arsitektur, pemilihan teknologi, dan integrasi fitur yang akan dijelaskan pada bagian berikutnya. Selain itu, studi-studi terkait pengelolaan dan pengamanan jaringan lokal [14], seperti implementasi sistem kendali jarak jauh dan manajemen pengguna berbasis OpenWRT pada jaringan RT/RW Net, menunjukkan bahwa pendekatan terpusat pada sisi infrastruktur jaringan dapat melengkapi solusi pemantauan dan kendali di level desktop yang diusulkan dalam penelitian ini.

3. METODE PENELITIAN

Metode Dynamic Systems Development Method (DSDM) dipilih dalam penelitian ini karena menyediakan pendekatan yang terstruktur dengan fokus pada *iterative prototyping*, keterlibatan aktif pengguna, dan pengelolaan waktu. Dibandingkan dengan metode agile lain seperti *Scrum* dan *Extreme Programming* (XP), DSDM menawarkan kerangka yang lebih formal, termasuk teknik pemodelan dan mekanisme pelibatan pemangku kepentingan. Hal ini membuat DSDM cocok

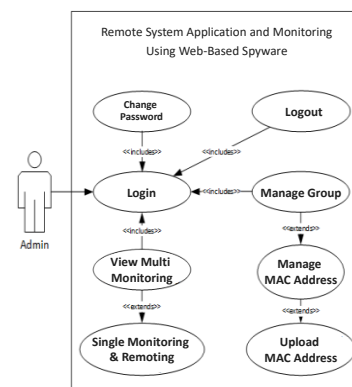
untuk sistem institusional yang melibatkan banyak peran pengguna dan memerlukan pengawasan administratif. Metodologi ini dibagi menjadi lima fase utama.

F. Feasibility Study

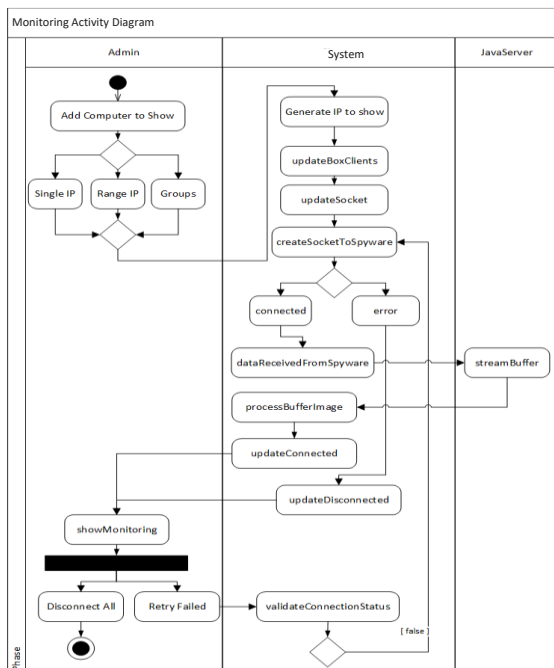
Studi kelayakan dilakukan untuk mengidentifikasi tantangan dalam mengelola banyak komputer klien di dalam jaringan lokal. Berdasarkan wawancara dan kuesioner, ditemukan adanya ketidakefisienan dalam proses remote troubleshooting, terutama ketika komputer berada di lokasi yang berbeda atau digunakan oleh pengguna dengan kemampuan teknis yang terbatas. Temuan ini menjadi dasar perlunya pengembangan aplikasi berbasis web yang mendukung pemantauan multi-klien, kendali jarak jauh, dan fitur Wake-on-LAN.

G. Business Study

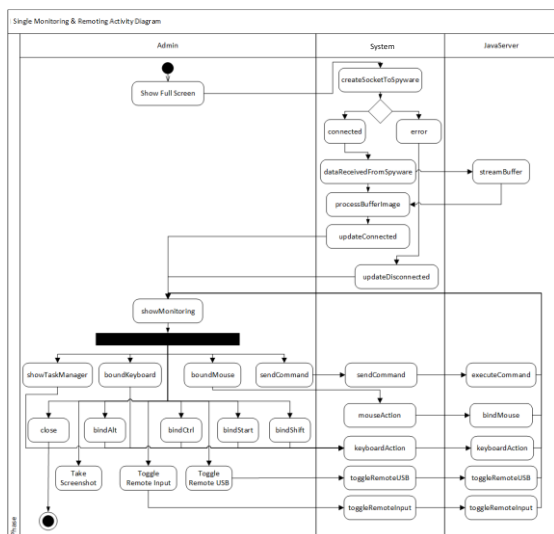
Fase ini berfokus pada pemilihan teknologi dan pendefinisian kebutuhan sistem. Sistem dirancang dengan menggunakan Node.js, Express.js, dan Socket.IO dengan arsitektur Model-View-Controller (MVC). Berbagai diagram Unified Modeling Language (UML), seperti use case diagram, activity diagram, dan sequence diagram, disusun untuk memodelkan perilaku sistem dan memvalidasi keputusan perancangan.



Gambar 1. Use Case Diagram



Gambar 2. Activity Diagram – Monitoring



Gambar 3. Activity Diagram – Remote Control

H. Functional Model Iteration

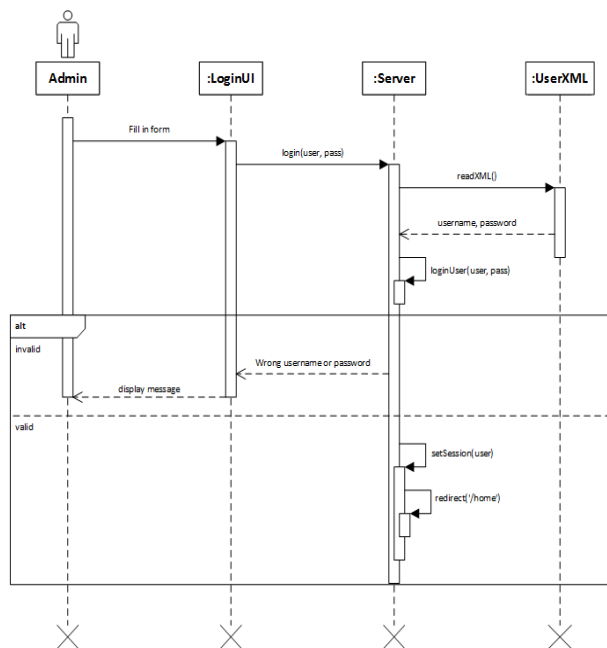
Prototipe dikembangkan untuk mendemonstrasikan fitur-fitur utama seperti pemantauan multi-klien, kendali jarak jauh, pengambilan tangkapan layar, dan Wake-on-LAN. Prototipe tersebut diuji secara internal dan disempurnakan berdasarkan masukan dari staf IT.

Gambar 4. Interface – Login Page

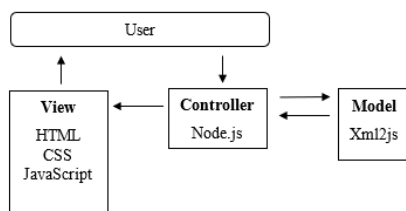
Gambar 5. Interface – Monitoring Page

Gambar 6. Interface – Remote Control Page

Gambar 7. Interface – Group Management



Gambar 8. Sequence Diagram – Login



Gambar 9. MVC Architecture Diagram

I. Design and Build Iteration

Sistem dikembangkan secara bertahap dengan pendekatan modular. Backend logic diimplementasikan menggunakan Node.js, sedangkan antarmuka frontend dibangun dengan HTML, CSS, dan JavaScript. Komunikasi real-time ditangani menggunakan WebSocket untuk memastikan interaksi berlatensi rendah antara server dan klien.

```

var websocket = 0,
    isWantToLogin = false;

/*User Variables for WebSocket Connections*/
var webUsers = {};

/*Log Type Variable for type of log*/
var typeLog = {};
typeLog['tcp'] = "TCP: ", typeLog['web'] = "Web: ", typeLog['udp'] = "UDP: ";

/*NodeJs Web Server listening to port*/
server.listen(1219);

/*Set static folder locations to be accessed*/
app.use(express.static(__dirname + '/public/assets'));

/*Set ExpressJS that can use bodyParser (to parse POST and GET variable)*/
app.use(bodyParser.urlencoded({ extended: true }));

/*Config ExpressJS session*/
app.use(session({
  secret: '4e84e499e272325able54a50e59044f3', //auble
  resave: false,
  saveUninitialized: false
})));
  
```

Gambar 10. Server-side Code Snippets (App.js, Socket Config)

J. Implementasi

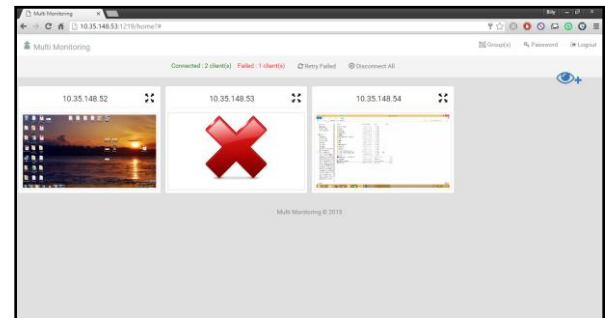
Sistem akhir diimplementasikan pada lingkungan laboratorium komputer dan dievaluasi melalui pengujian oleh pengguna. Umpan balik dikumpulkan menggunakan kuesioner pasca-uji untuk mengukur aspek keterpakaianya, kinerja, dan keandalan sistem.

4. HASIL DAN PEMBAHASAN

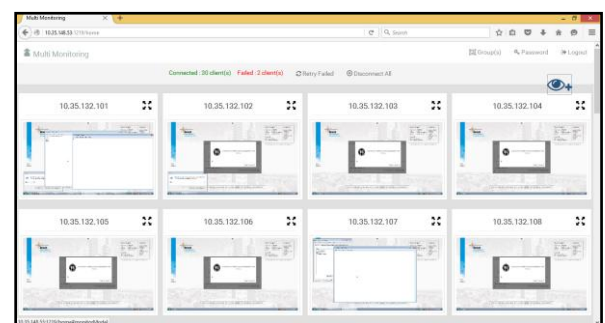
A. Hasil Implementasi Sistem

Aplikasi yang dikembangkan memungkinkan administrator untuk memantau dan mengendalikan banyak komputer klien secara real-time tanpa memerlukan instalasi perangkat lunak di sisi klien. Fitur utama yang disediakan antara lain:

- **Pemantauan multi-klien:** Menampilkan status terkini dari seluruh komputer yang terhubung secara *real-time*.



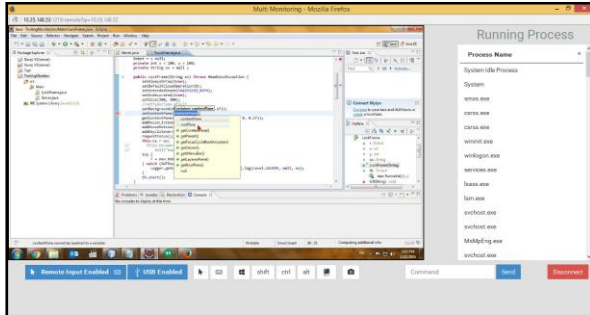
Gambar 11. Range-Based IP Monitoring Interface



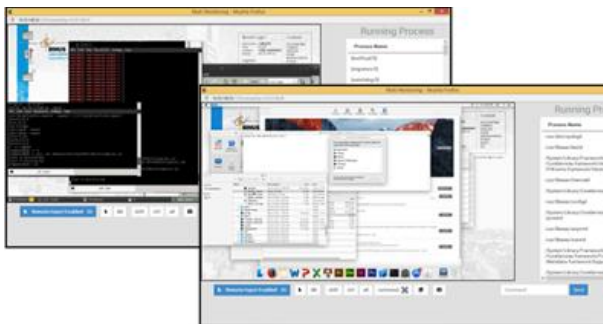
Gambar 12. Group-Based Monitoring Interface

- **Kendali jarak jauh:** Menjalankan perintah pada komputer klien, seperti shutdown, restart, dan pemblokiran input.
- **Pengambilan tangkapan layar:** Menyediakan umpan balik visual untuk keperluan audit dan troubleshooting.

- **Kirim perintah (Send Command):** Menjalankan perintah pada komputer klien tanpa diketahui oleh pengguna di sisi klien.
- **USB Blocking:** Mencegah perpindahan data yang tidak sah melalui media penyimpanan eksternal.

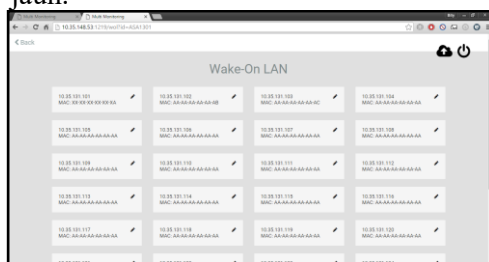


Gambar 13. Remote Control Interface for Windows Clients



Gambar 14. Remote Control Interface for Linux & MacOS Clients

- **Wake-on-LAN:** Mengaktifkan komputer yang sedang dalam kondisi mati dari jarak jauh.



Gambar 15. Wake-on-LAN Activation Interface

```

websocket.on('wakeUp', function (data){
    var jsonWOL = readXML('wol.xml');
    var arrWOL = parseJSONToArray(jsonWOL['WOL'][data][0]);

    var jsonGroups = readXML('groups.xml');

    for(var i=0; i<arrWOL.length; i++){
        wol_wake(arrWOL[i][0], {
            address: jsonGroups['GroupsList'][data][0]['network'][0] + '.255',
            port: 7
        }, function(error){
            if(error){
                console.log('gagal');
                return;
            }
            console.log('sukses');
            return;
        });
        wol.createMagicPacket(arrWOL[i][0]);
    }
});
    
```

Gambar 16. Wake-on-LAN Function

```

/*REMOTING HANDLER*/
var canvas = $('#remoting-canvas')[0],
    image;
canvas.width = canvas.offsetLeft;
canvas.height = canvas.offsetTop;
var ctx = canvas.getContext('2d');

var MouseLocation = {
    x: 0,
    y: 0
};

/*Data Coming*/
socket.on('imagebuffer-0', function (idx, data) {
    if(idx == 0) {
        image.src = 'data:image/jpeg;base64,' + data;
    }
});

ctx = canvas.getContext("2d");
canvasWidth = canvas.width;
image = new Image();
image.onload = onLoadedImage();

function onLoadedImage(i){
    return function(){
        /*Auto Fill Image and resize on Canvas*/
        ctx.drawImage(image, 0, 0, canvasWidth, canvasWidth * image.height / image.width);
    }
}

/*Mouse*/
canvas.addEventListener('mousemove', function(e){
    MouseLocation.x = e.clientX;
    MouseLocation.y = e.clientY;
    setLocation();
});

document.oncontextmenu = document.body.oncontextmenu = function() {return false;}
canvas.onmousedown = function(e){
    /*
    0 - Left Click
    1 - Middle Click
    2 - Right Click
    actid,x => x = Right | Left | Middle
    */
    console.log(e.x + " - " + e.y);
    if(isBoundMouse){
        socket.emit('mouseAction', getURLParam('ip'), "actid," + e.button);
    }
};

canvas.onmouseup = function(e){
    /*
    0 - Left Click
    1 - Middle Click
    2 - Right Click
    actid,x => x = Right | Left | Middle
    */
    if(isBoundMouse){
        socket.emit('mouseAction', getURLParam('ip'), "actiu," + e.button);
    }
};

canvas.onmousewheel = function(e){
    /*
    0 - Left Click
    1 - Middle Click
    2 - Right Click
    actid,x => x = Right | Left | Middle
    */
    //var mouseWheel = (e.wheelDelta/120); //Wheel seperti di mac
    var mouseWheel = -(e.wheelDelta/120); //Wheel seperti di windows
    if(isBoundMouse){
        socket.emit('mouseAction', getURLParam('ip'), "scroll:" + mouseWheel);
    }
};
    
```

Gambar 17. Server-side Code Snippets (App.js, Socket Handling)

B. Evaluasi Teknis

Sistem diuji pada lingkungan laboratorium yang merepresentasikan kondisi dunia nyata. Metrik kinerja utama yang diukur meliputi:

- **Kecepatan akses situs:** sekitar 556 ms pada jaringan internal.

15 requests | 4.3 KB transferred | Finish: 556 ms | DOMContentLoaded: 354 ms | Load: 354 ms

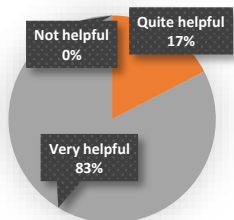
Gambar 18. Network Status

- **Wake-on-LAN:** kurang dari 1 detik untuk menyalakan 33 komputer.
- **Monitoring delay:** 2–3 detik untuk pembaruan tampilan layar secara real-time.
- **Remote control latency:** sekitar 1 detik, terutama dipengaruhi oleh interval pengiriman citra dari klien ke server.

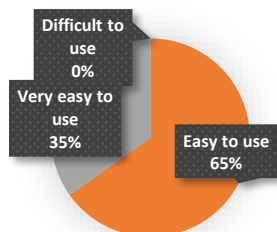
C. Evaluasi Pengguna

Kuesioner pasca-uji didistribusikan kepada 23 pengguna yang terdiri dari administrator jaringan, instruktur laboratorium, dan mahasiswa. Temuan utama yang diperoleh adalah:

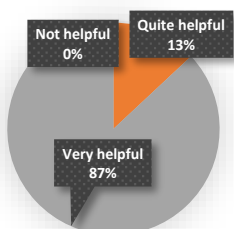
- 83% responden menilai fitur *multi-monitoring* “sangat membantu”.



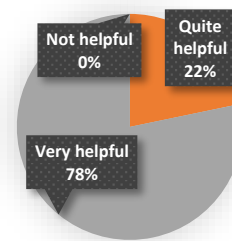
- 100% responden menilai aplikasi “mudah digunakan”.



- 87% responden menilai fitur Wake on LAN “sangat membantu”.



- 78% responden menilai fitur tangkapan layar (screenshot) “sangat membantu”.



Hasil tersebut menunjukkan tingkat kepuasan pengguna yang tinggi, dengan sistem mampu beroperasi secara andal pada kondisi dunia nyata.

D. Pembahasan

Hasil evaluasi menunjukkan bahwa sistem yang dikembangkan mampu menjawab kebutuhan operasional administrator TI. Tingkat kepuasan pengguna yang tinggi mengindikasikan bahwa sistem mudah digunakan, memiliki kinerja yang baik, dan relevan dengan kebutuhan pengguna. Tidak adanya kebutuhan instalasi di sisi klien serta dukungan terhadap berbagai platform (Windows, Linux, dan macOS) membuat sistem ini lebih praktis untuk diterapkan di berbagai lingkungan.

Namun, masukan pengguna juga menyoroti beberapa area yang perlu ditingkatkan, antara lain:

- Integrasi **sistem notifikasi** untuk memberi peringatan kepada administrator ketika terdapat aktivitas klien yang mencurigakan.
- Penambahan fitur **penjadwalan tugas** (*task scheduling*).
- **Antarmuka yang ramah** perangkat mobile untuk memperluas aksesibilitas.
- Kemampuan **file transfer** antara server dan klien.
- Implementasi **enkripsi SSL** untuk meningkatkan aspek keamanan.

Rekomendasi tersebut sejalan dengan arah pengembangan sistem pemantauan modern yang menekankan penguatan keamanan jaringan lokal dan mekanisme manajemen pengguna terpusat, sebagaimana juga ditunjukkan pada penelitian terkait manajemen jaringan dan kendali jarak jauh di tingkat infrastruktur yang dipublikasikan oleh Putra dan Ubaidi di JITET [14].

5. KESIMPULAN

Berdasarkan hasil evaluasi, aplikasi berbasis web untuk pemantauan dan manajemen jarak jauh yang dikembangkan menunjukkan kinerja yang baik dalam membantu tim IT support dan administrator laboratorium memantau komputer klien serta melakukan troubleshooting perangkat lunak. Fitur multi-monitoring memungkinkan pemantauan banyak klien secara bersamaan dan real-time tanpa instalasi di sisi klien, sehingga meningkatkan efisiensi operasional dan mempercepat waktu respons terhadap insiden. Selain itu, kombinasi fitur akses jarak jauh, pemblokiran input, Wake-on-LAN, dan pengambilan tangkapan layar menjadikan sistem ini solusi alternatif yang efektif untuk kebutuhan remote troubleshooting di lingkungan pendidikan dan institusional.

UCAPAN TERIMA KASIH

Lusiana Citra Dewi: *Methodology, Validation, Writing – Review & Editing, Supervision*. Alvin Chandra: *Software, Testing, Visualization, Writing – Review & Editing*. Indra Dwi Rianto: *Writing – Review & Editing*. Renny Triana: *Writing – Review & Editing*.

DAFTAR PUSTAKA

- [1] V. Gupta, A. Garg, and S. Agrawal, "Remote Monitoring and Control Systems in Agriculture and Farming," in *Recent Trends in Artificial Intelligence Towards a Smart World*, R. Arya, S. C. Sharma, A. K. Verma, and B. Iyer, Eds. Singapore: Springer, 2024. [Online]. Available: https://doi.org/10.1007/978-981-97-6790-8_10
- [2] H. Kopetz and W. Steiner, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 3rd ed. Cham, Switzerland: Springer, 2022.
- [3] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-Physical Systems: The Next Computing Revolution," in *Proceedings of the 47th Design Automation Conference (DAC 2010)*, IEEE, 2010. [Online]. Available: <https://doi.org/10.1145/1837274.1837461>
- [4] M. U. Saleem, M. R. Usman, and M. Shakir, "Design, Implementation, and Deployment of an IoT Based Smart Energy Management System," *IEEE Access*, vol. PP, pp. 1–1, Apr. 2021. [Online]. Available: <https://doi.org/10.1109/ACCESS.2021.3070960>
- [5] A. A. Laghari, H. Li, A. A. Khan, Y. Shoulin, S. Karim, and M. A. K. Khani, "Internet of Things (IoT) applications security trends and challenges," *Discover Internet of Things*, vol. 4, Art. no. 36, 2024. [Online]. Available: <https://doi.org/10.1007/s43926-024-00090-5>
- [6] M. Burunkaya, et al., "Design and implementation of an IoT-based smart classroom system," *Applied Sciences*, vol. 12, no. 4, Art. no. 2233, 2022. [Online]. Available: <https://doi.org/10.3390/app12042233>
- [7] J. Nakazato, K. Nakagawa, K. Itoh, et al., "WebRTC over 5G: A study of remote collaboration QoS in mobile environment," *Journal of Network and Systems Management*, vol. 32, no. 1, 2024. [Online]. Available: <https://doi.org/10.1007/s10922-023-09778-5>
- [8] I. L. B. M. Paris, M. H. Habaebi, and A. M. Zyoud, "Implementation of SSL/TLS security with MQTT protocol in IoT environment," *Wireless Personal Communications*, vol. 132, pp. 163–182, 2023. [Online]. Available: <https://doi.org/10.1007/s11277-023-10605-y>
- [9] K. Deshpande, A. Jain, Abhinav, S. Mishra, S. Goel, and R. Garg, "Empowering real-time communication: A seamless chatting system using WebSocket," in *Innovative Computing and Communications (ICICC 2024)*, A. E. Hassanien, S. Anand, A. Jaiswal, and P. Kumar, Eds., *Lecture Notes in Networks and Systems*, vol. 1039. Singapore: Springer, 2025. [Online]. Available: https://doi.org/10.1007/978-981-97-4152-6_29
- [10] Zhu, Q. (2020). Design and Implementation of Smart Classroom System Based on Internet of Things Technology. In: Jain, V., Patnaik, S., Popențiu Vlădicescu, F., Sethi, I. (eds) *Recent Trends in Intelligent Computing, Communication and Devices. Advances in Intelligent Systems and Computing*, vol 1006. Springer, Singapore. https://doi.org/10.1007/978-981-13-9406-5_44
- [11] Loja, A., Maita, T. (2024). Comparative Evaluation of Performance Efficiency in Terms of Temporal Behavior and Resource Utilization, According to the ISO/IEC 25,010 Model, in a Web Application Developed with Angular, React.js, and Vue.js. In: Olmedo Cifuentes, G.F., Arcos Avilés, D.G., Lara Padilla, H.V. (eds) *Emerging Research in*

- Intelligent Systems. CIT 2023. Lecture Notes in Networks and Systems, vol 902. Springer, Cham. https://doi.org/10.1007/978-3-031-52255-0_21
- [12] Chakraborty, M., Kundan, A.P. (2021). Architecture of a Modern Monitoring System. In: Monitoring Cloud-Native Applications. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-6888-9_3
- [13] Maurya, V., Rishiwal, V., Yadav, M. *et al.* Blockchain-driven security for IoT networks: State-of-the-art, challenges and future directions. *Peer-to-Peer Netw. Appl.* 18, 53 (2025). <https://doi.org/10.1007/s12083-024-01812-w>
- [14] A. A. Putra and Ubaidi, "Implementasi Sistem Kendali Jarak Jauh dan Manajemen User Berbasis OpenWRT pada Jaringan RT/RW Net di Desa Laden," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3S1, pp. 3761–3768, 2024. [Online]. Available: <https://doi.org/10.23960/jitet.v12i3S1.5200>
- [15] A. Aska, "Rancang Bangun Sistem Monitoring Daya Listrik Berbasis IoT untuk Rumah Tinggal," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 1, 2025. [Online]. Available: <https://doi.org/10.23960/jitet.v13i2.6160>
- [16] T. Ishihara and H. Tamaki, "Performance Evaluation of Reconfigurable TCP Communication on Remote Desktop System," in *Proc. 2014 IEEE 28th Int. Conf. Advanced Information Networking and Applications*, 2014, pp. 1035–1042. [Online]. Available: <https://ieeexplore.ieee.org/document/6899259>
- [17] T. Kärkkäinen, J. Kämäräinen, and L. Lensu, "Performance evaluation of remote display access for mobile cloud computing," *Computer Communications*, vol. 68, pp. 1–12, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0140366415001619>
- [18] S. Saeed and M. Alam, "Performance Analysis of Secure & Efficient Remote Desktop Manager," *Journal of Independent Studies and Research–Computing*, vol. 11, no. 1, pp. 16–24, 2013. [Online]. Available: <https://jict.ilmauniversity.edu.pk/journal/jict/11.1/4.pdf>
- [19] H. D. Friedland, "An Accessible Web-Based Peer-to-Peer Remote Desktop Control and Access Tool Utilizing WebRTC," M.S. thesis, Dept. of Computer Science and Engineering, Univ. of Tennessee at Chattanooga, Chattanooga, TN, USA, 2024. [Online]. Available: <https://scholar.utc.edu/theses/849>