

RUTE DINAMIS BERBASIS MINIMUM SPANNING TREE UNTUK KOORDINASI NODE BERGERAK DI LINGKUNGAN KAMPUS

Elda Rayhana¹, M. Ikrar Yamin², R.M. Dinata³

¹Fisika, Fakultas Sains Terapan dan Teknologi, ISTN Jakarta

²Teknik Elektronika, Fakultas Sains Terapan dan Teknologi, ISTN Jakarta

³Teknik Informatika, Fakultas Sains Terapan dan Teknologi, ISTN Jakarta

*Penulis korespondensi: elray@istn.ac.id

ABSTRAK

Koordinasi beberapa node bergerak dalam lingkungan dengan infrastruktur jaringan terbatas memerlukan mekanisme rute yang mampu beradaptasi terhadap perubahan topologi secara real-time. Penelitian ini menyajikan prototipe sistem rute dinamis berbasis Minimum Spanning Tree (MST) untuk koordinasi lima node bergerak yang dibangun dengan ESP32 dan modul Global Positioning System (GPS), dengan komputasi MST dijalankan secara terpusat pada cloud server yang diakses melalui tethering WiFi smartphone. Arsitektur yang dikembangkan mengimplementasikan algoritma Kruskal dengan optimasi Union-Find untuk membentuk graf berbobot dari jarak antarnode berbasis koordinat GPS, kemudian mengirimkan kembali instruksi rute berupa tetangga dan arah pergerakan ke setiap node. Pengujian dilakukan pada skenario statis dan gerak linier terkoordinasi yang merepresentasikan jarak antarkampus di wilayah Jakarta Selatan, dengan parameter utama berupa waktu komputasi MST, latency ujung ke ujung, dan indeks kestabilan pohon. Hasil pengujian menunjukkan bahwa waktu komputasi MST berada pada orde beberapa milidetik dengan latency ujung ke ujung sekitar 187–210 milidetik dan nilai Tree Stability Index yang tinggi, sehingga struktur MST relatif stabil terhadap perubahan posisi yang moderat. Temuan ini mengindikasikan bahwa pendekatan terpusat berbasis MST dengan dukungan cloud server layak digunakan sebagai solusi awal untuk koordinasi tim bergerak skala kecil di lingkungan kampus dan dapat dikembangkan lebih lanjut untuk skenario mobilitas yang lebih kompleks.

Kata kunci: Minimum Spanning Tree; Routing Dinamis; ESP32; GPS; Client Server

1 PENDAHULUAN

Pemanfaatan sistem bergerak otonom untuk tugas-tugas seperti pencarian dan penyelamatan, patroli keamanan, pemantauan lingkungan, serta koordinasi operasi taktis menuntut ketersediaan mekanisme komunikasi dan koordinasi yang andal meskipun infrastruktur jaringan tetap tidak selalu tersedia (Asif et al., 2017; Kumar et al., 2020; Ragavan & Ramasamy, 2019). Dalam banyak skenario lapangan, *node* bergerak hanya mengandalkan konektivitas nirkabel berbasis WiFi atau *tethering* ponsel cerdas untuk mengirimkan informasi posisi dan perintah rute secara real-time, sehingga fluktuasi kualitas kanal dan keterbatasan jangkauan menjadi faktor pembatas kinerja sistem (Pantazis et al., 2013). Berbagai studi pada *wireless sensor networks* (WSN) dan *mobile ad-hoc networks* (MANET) menunjukkan bahwa perubahan topologi akibat mobilitas dapat menurunkan kinerja *routing* jika struktur jaringan tidak diperbarui secara adaptif, yang tercermin dalam peningkatan *latency*, konsumsi energi yang lebih besar, serta penurunan *quality of service* (QoS) pada aplikasi yang sensitif terhadap waktu (Akyildiz et al., 2002; Pantazis et al., 2013; Royer & Toh, 1999).

Seiring berkembangnya *microcontroller* berbiaya rendah seperti ESP32 yang terintegrasi dengan modul *Global Positioning System* (GPS) dan antarmuka WiFi, muncul peluang untuk

membangun prototipe sistem koordinasi *node* bergerak yang portabel dan mudah direalisasikan sebagai *testbed* di lingkungan kampus maupun skenario skala kecil lainnya (Kumar et al., 2020). Pada saat yang sama, pemanfaatan komputasi awan memungkinkan pemindahan beban pemrosesan graf dan algoritma *routing* dari perangkat tepi yang terbatas sumber daya ke *server* pusat yang lebih kuat, sehingga perhitungan topologi dapat dilakukan dengan lebih cepat dan konsisten (Lin et al., 2023; Wang et al., 2016). Dalam konteks ini, pendekatan *routing* yang mampu memanfaatkan informasi posisi geografis secara eksplisit dan membentuk struktur jaringan yang efisien menjadi semakin relevan, terutama untuk memastikan konektivitas global dengan biaya komunikasi yang minimal (Lin et al., 2023; Stojmenovic et al., 2008).

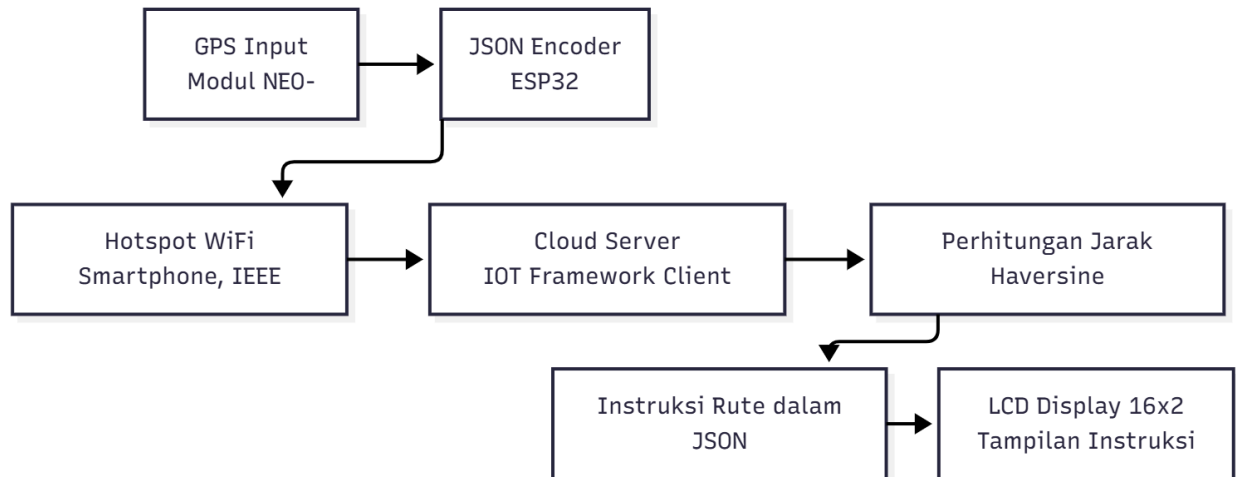
Secara teoritis, algoritma *Minimum Spanning Tree (MST)* menawarkan kerangka graf yang menjamin keterhubungan semua *node* dengan total bobot sisi minimum, sehingga banyak dimanfaatkan sebagai tulang punggung rute komunikasi atau rute pergerakan fisik antarnode pada jaringan nirkabel (Das & Roy, 2012; Lin et al., 2023; Stojmenovic et al., 2008). Sejumlah penelitian telah mengkaji pemanfaatan *MST* untuk *routing* pada jaringan bergerak, baik melalui formulasi kinetik, skema terdistribusi, maupun pendekatan probabilistik yang mempertimbangkan keterbatasan energi dan dinamika kanal (Li & Canovas, 2001; López et al., 2013; Wang et al., 2016). Namun, banyak kajian tersebut masih didominasi oleh simulasi atau uji pada kondisi ideal, sehingga aspek-aspek praktis seperti akurasi *GPS* pada perangkat murah, variasi *latency* ujung ke ujung akibat jalur WiFi–seluler, serta perilaku kestabilan *MST* pada prototipe berbasis *ESP32* dalam lingkungan nyata belum dievaluasi secara menyeluruh (Kumar et al., 2020; Pantazis et al., 2013).

Berdasarkan celah tersebut, penelitian ini berfokus pada perancangan dan evaluasi prototipe sistem rute dinamis berbasis *MST* untuk koordinasi beberapa *node* bergerak yang dibangun dengan *ESP32* dan modul *GPS*, dengan komputasi *MST* dijalankan secara terpusat pada *cloud server* yang diakses melalui *tethering* WiFi *smartphone* (Lin et al., 2023; Wang et al., 2016). Studi kasus difokuskan pada konfigurasi lima *node* yang mewakili lokasi kampus di Jakarta Selatan (ISTN, UI, UNAS, Indraprasta, dan Jayabaya), dengan dua skenario mobilitas utama, yaitu kondisi statis dan gerak linier terkoordinasi yang merepresentasikan perpindahan antarkampus (Li & Canovas, 2001; López et al., 2013; Wang et al., 2016). Tujuan utama penelitian adalah mengukur dan menganalisis waktu komputasi *MST*, *latency* ujung ke ujung, serta *Tree Stability Index* pada kedua skenario tersebut sebagai dasar untuk menilai kelayakan pendekatan *MST* terpusat bagi koordinasi tim bergerak skala kecil. Kontribusi spesifik yang diusulkan meliputi perancangan arsitektur *client–server* sederhana untuk rute *MST* berbasis data posisi *GPS*, implementasi prototipe *ESP32–GPS* yang terintegrasi dengan layanan *cloud*, serta penyediaan analisis empiris mengenai kestabilan struktur *MST* dan komponen *latency* pada lingkungan kampus yang realistis.

2 METODE

2.1 Desain Umum Sistem

Penelitian ini menggunakan pendekatan rekayasa sistem yang mengintegrasikan perancangan arsitektur, implementasi prototipe perangkat keras dan perangkat lunak, serta pengujian eksperimental pada skenario mobilitas yang terkontrol (Lin et al., 2023). Sistem yang dikembangkan mengadopsi arsitektur *client–server*, di mana lima *node* bergerak berbasis *ESP32–GPS* berfungsi sebagai *client* yang secara periodik mengirimkan data posisi ke *cloud server* untuk dilakukan komputasi *Minimum Spanning Tree* dan menerima kembali instruksi rute (Kumar et al., 2020; Lin et al., 2023).



Gambar 1. Diagram alir sistem rute dinamis berbasis Minimum Spanning Tree (MST)

Diagram ini memperlihatkan alur data dari pembacaan koordinat oleh modul *GPS* NEO-6M, proses *encoding* ke format *JSON* di *ESP32*, pengiriman melalui *hotspot* WiFi *smartphone* ke *cloud server* untuk komputasi jarak Haversine dan *MST* dengan algoritma Kruskal–Union–Find, hingga pembentukan instruksi rute dalam *JSON* yang dikembalikan ke *ESP32* dan ditampilkan pada *LCD* 16×2 sebagai panduan koordinasi *node* bergerak.

2.2 Perangkat Keras dan Perangkat Lunak

Node client dibangun menggunakan modul *ESP32-WROOM* yang terhubung dengan modul *GPS* u-blox NEO-6M dan *LCD* 16×2 berbasis antarmuka *I²C* sebagai media tampilan instruksi rute (Kumar et al., 2020). Setiap *node* memperoleh catu daya dari adaptor 5 V USB untuk meminimalkan variasi tegangan selama pengukuran, sementara konektivitas jaringan disediakan oleh *hotspot* WiFi *smartphone* yang beroperasi pada standar IEEE 802.11n pita 2,4 GHz (Kumar et al., 2020).

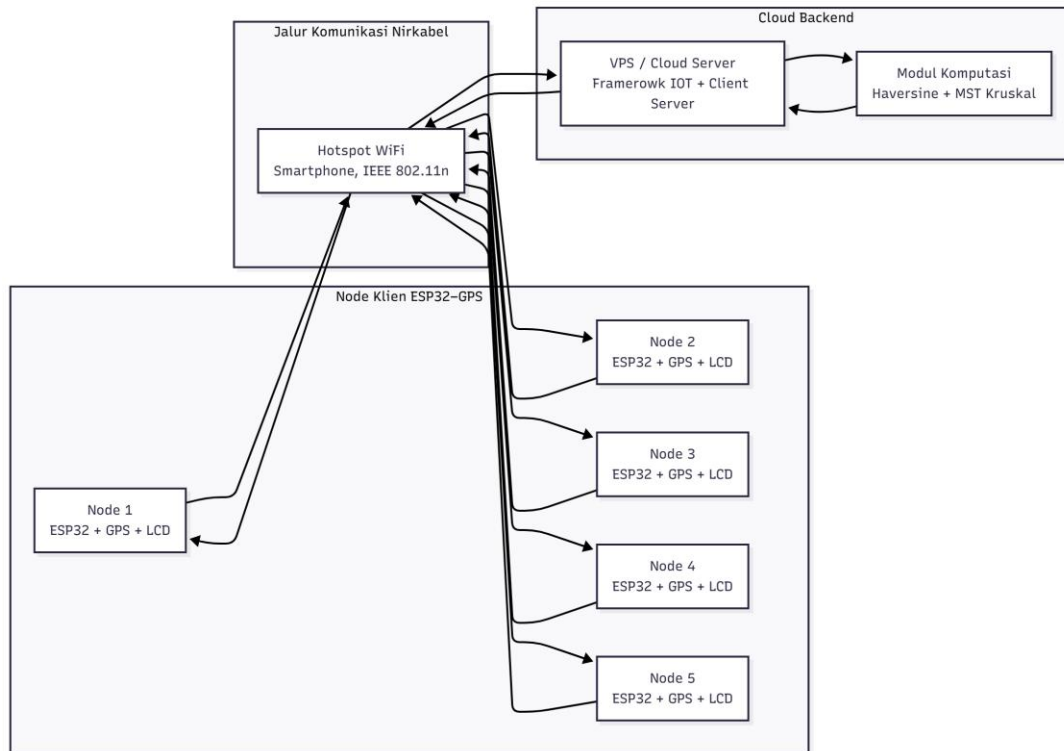
Perangkat lunak sisi *client* dikembangkan dengan *Arduino IDE* menggunakan pustaka *TinyGPS++* untuk pemrosesan data *GPS*, *ArduinoJson* untuk serialisasi dan deserialisasi *JSON*, serta pustaka *LCD* berbasis *I²C* untuk menampilkan informasi tetangga dan arah pergerakan. Pada sisi *server*, *backend* diimplementasikan pada *virtual private server* (*VPS*) berbasis Linux yang menjalankan layanan web menggunakan bahasa Python dan kerangka kerja ringan untuk menerima permintaan HTTP dari *node* (Hu et al., 2021; Lin et al., 2023).

2.3 Arsitektur dan Alur Komunikasi

Pada *server*, data koordinat *GPS* yang diterima dari seluruh *node* digunakan untuk membangun graf lengkap berbobot dengan menghitung jarak antarnode menggunakan rumus Haversine. Selanjutnya, algoritma *MST* Kruskal dengan struktur data *Union–Find* dijalankan untuk memperoleh pohon rentang minimum dan menghasilkan instruksi rute per *node* yang berisi daftar tetangga, jarak, serta *bearing* (Lin et al., 2023; Wang et al., 2016). Instruksi ini dikemas kembali dalam format *JSON* dan dikirimkan ke masing-masing *node* melalui jalur WiFi–seluler yang sama, untuk kemudian ditampilkan pada *LCD*.

Secara garis besar, alur komunikasi sistem dapat diringkas sebagai berikut: *ESP32* membaca data koordinat dari modul *GPS* dan membentuk paket *JSON*, paket dikirim melalui jaringan WiFi ke *cloud server*, *server* menghitung jarak dan membentuk graf berbobot, menjalankan algoritma *MST*, lalu mengonversi hasilnya menjadi instruksi rute per *node* yang dikirim kembali

untuk ditampilkan pada *LCD*. Alur ini dapat direpresentasikan dalam sebuah diagram alir untuk memperjelas hubungan antara komponen *client* dan *server*.



Gambar 2. Arsitektur *client-server* prototipe ESP32-GPS-cloud

Diagram ini menggambarkan lima *node* klien berbasis ESP32-GPS yang terhubung ke *cloud backend* melalui *hotspot* WiFi *smartphone*: setiap *node* mengirimkan koordinat ke VPS untuk komputasi jarak Haversine dan MST Kruskal, lalu menerima kembali instruksi rute yang ditampilkan pada *LCD* masing-masing.

2.4 Skenario Pengujian

Pengujian dilakukan pada dua skenario mobilitas utama. Pada skenario statis, lima *node* ditempatkan pada posisi tetap yang merepresentasikan koordinat lima kampus di Jakarta Selatan (ISTN, UI, UNAS, Indraprasta, dan Jayabaya) dan dibiarkan diam selama 10 menit dengan frekuensi pembaruan 2 Hz, sehingga dihasilkan 600 siklus pembaruan posisi dan perhitungan MST (Li & Canovas, 2001; López et al., 2013). Skenario ini digunakan sebagai *baseline* untuk mengevaluasi waktu komputasi MST, *latency* ujung ke ujung, dan kestabilan struktur pohon ketika topologi jaringan tidak berubah.

Pada skenario gerak linier terkoordinasi, posisi *node* disimulasikan bergerak sepanjang lintasan yang merepresentasikan perpindahan antarkampus dengan kecepatan mendekati 10 m/s selama 15 menit. Data posisi yang telah direkam diputar ulang (*replay*) ke dalam sistem sehingga *server* memperoleh deret koordinat yang berubah secara teratur dan melakukan komputasi MST pada setiap siklus, sehingga efek perubahan jarak relatif antarnode terhadap kestabilan pohon dapat diamati secara sistematis (Wang et al., 2016).

Tabel 1 merangkum komponen utama perangkat keras dan perangkat lunak yang digunakan dalam prototipe sistem rute dinamis berbasis MST, sehingga memudahkan pembaca melihat spesifikasi dan peran masing-masing elemen secara ringkas.

Tabel 1. Konfigurasi perangkat keras dan perangkat lunak sistem prototipe

Komponen	Spesifikasi Utama	Fungsi Utama
<i>Microcontroller</i>	ESP32-WROOM, terintegrasi WiFi	Akuisisi data <i>GPS</i> , pengiriman/ penerimaan <i>JSON</i> , kontrol <i>LCD</i>
Modul <i>GPS</i>	u-blox NEO-6M, laju pembaruan 5 Hz, akurasi $\pm 2\text{--}3$ m	Menyediakan koordinat posisi <i>real-time</i> node bergerak
<i>Display</i>	LCD 16×2, antarmuka I ² C	Menampilkan instruksi rute (tetangga, jarak, <i>bearing</i>)
Konektivitas nirkabel	<i>Hotspot</i> WiFi <i>smartphone</i> , IEEE 802.11n, 2,4 GHz	Menghubungkan semua node ke <i>cloud server</i> melalui jaringan seluler
Daya	Adaptor USB 5 V stabil	Menyediakan catu daya konstan untuk setiap node
<i>Backend server</i>	VPS Linux, Flask (Python), koneksi internet publik	Menerima data posisi, mengelola permintaan <i>client</i>
Modul komputasi graf	Python + <i>NetworkX</i> + implementasi Kruskal + <i>Union-Find</i>	Menghitung jarak Haversine, menyusun graf, dan menghitung <i>MST</i>
Pustaka <i>client</i>	<i>Arduino IDE</i> , <i>TinyGPS++</i> , <i>ArduinoJson</i> , pustaka <i>LCD</i> I ² C	Memproses data <i>GPS</i> , membentuk/ mem-parsing <i>JSON</i> , dan mengendalikan tampilan
Protokol komunikasi	HTTP sederhana berbasis <i>REST</i>	Mengirim dan menerima data posisi serta instruksi rute antarsisi <i>client-server</i>

Tabel ini menunjukkan bahwa seluruh rantai sistem dibangun dari komponen *off-the-shelf* berbiaya rendah yang dipadukan dengan *backend* berbasis Python dan pustaka graf, sehingga konfigurasi prototipe relatif mudah direplikasi di lingkungan kampus untuk keperluan uji coba koordinasi *node* bergerak skala kecil.

2.5 Parameter Pengukuran

Beberapa parameter utama diukur secara konsisten pada kedua skenario untuk menilai kinerja sistem. Waktu komputasi *MST* didefinisikan sebagai selang waktu di sisi *server* dari saat seluruh data posisi diterima hingga algoritma Kruskal selesai menghasilkan himpunan sisi *MST*. *Latency* ujung ke ujung dihitung sebagai selisih antara waktu pengambilan data *GPS* di *node ESP32* dan waktu instruksi rute muncul pada *LCD*, sehingga mencakup komponen keterlambatan akuisisi *GPS*, transmisi WiFi–seluler, pemrosesan *server*, dan *parsing* di sisi *client* (Kumar et al., 2020).

Selain itu, dicatat pula panjang total *MST* sebagai indikator efisiensi topologi rute, *Tree Stability Index* sebagai ukuran proporsi siklus pembaruan di mana himpunan sisi *MST* tidak berubah, serta kekuatan sinyal WiFi dan kejadian kehilangan paket untuk memastikan hasil tidak didominasi oleh gangguan jaringan ekstrem (Pantazis et al., 2013; Wang et al., 2016).

3 HASIL DAN PEMBAHASAN

3.1 Kinerja Skenario Statis

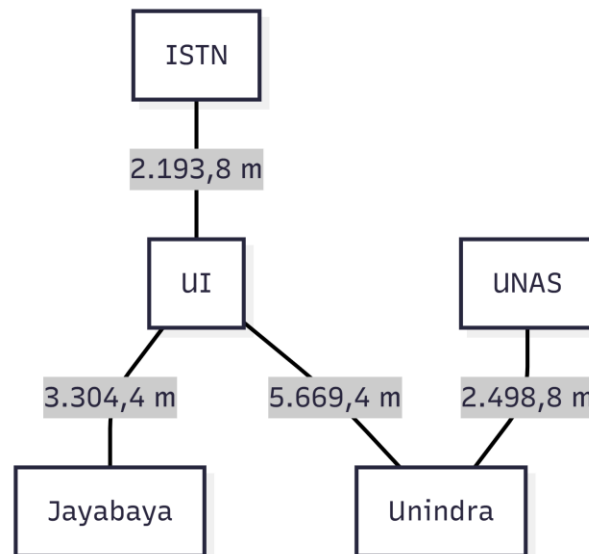
Pada skenario statis, lima *node* ditempatkan pada posisi tetap yang merepresentasikan koordinat lima kampus di Jakarta Selatan (ISTN, UI, UNAS, Indraprasta, dan Jayabaya) selama 10 menit dengan frekuensi pembaruan 2 Hz sehingga dihasilkan 600 siklus pembaruan posisi dan perhitungan *Minimum Spanning Tree (MST)*. Hasil pengukuran menunjukkan bahwa waktu komputasi *MST* di *cloud server* berada pada kisaran 2–4 milidetik per siklus dengan nilai rata-rata 2,847 milidetik, sedangkan *latency* ujung ke ujung berkisar antara 142,8–289,5 milidetik dengan rata-rata sekitar 187,3 milidetik.

Tabel 2. Kinerja skenario statis (S1) selama 600 siklus

Metrik	Rata-rata	Simpangan Baku	Minimum	Maksimum
Waktu komputasi <i>MST</i> (ms)	2,847	0,394	2,103	4,256
<i>Latency</i> ujung ke ujung (ms)	187,3	23,6	142,8	289,5
<i>Tree Stability Index</i>	1,000	0,000	1,000	1,000
Jumlah peristiwa rekonfigurasi	0	–	–	–

Secara struktural, *MST* yang diperoleh pada skenario ini konsisten sepanjang 600 siklus dengan *Tree Stability Index* sebesar 1,0, yang berarti himpunan sisi pohon tidak pernah berubah dari satu pembaruan ke pembaruan berikutnya. Panjang total *MST* berkisar sekitar 13,7 km, dengan UI berperan sebagai simpul pusat yang menghubungkan tiga cabang utama (ISTN–UI, UNAS–Unindra, dan UI–Jayabaya), selaras dengan sebaran geografis kampus di wilayah Jakarta Selatan. Hasil ini menunjukkan bahwa pada topologi yang tidak berubah, beban komputasi *MST* relatif kecil dibandingkan komponen *latency* jaringan, dan sistem mampu mempertahankan rute yang stabil untuk koordinasi tim kecil.

Diagram posisi lima kampus dengan sisi-sisi *MST* yang menghubungkan ISTN, UI, UNAS, Indraprasta, dan Jayabaya, serta label jarak tiap sisi Adalah sebagai berikut:



Gambar 3. Topologi MST pada skenario statis kampus Jakarta Selatan

Gambar 3 menggambarkan struktur *Minimum Spanning Tree (MST)* yang terbentuk pada skenario statis ketika lima node ditempatkan di lokasi kampus ISTN, UI, UNAS, Unindra, dan Jayabaya di Jakarta Selatan. UI berperan sebagai simpul pusat yang menghubungkan tiga cabang utama, yaitu ke ISTN ($\approx 2.193,8$ m), ke Unindra ($\approx 5.669,4$ m), dan ke Jayabaya ($\approx 3.304,4$ m), sementara UNAS terhubung ke Unindra dengan jarak sekitar 2.498,8 m, sehingga seluruh kampus terhubung dalam satu pohon dengan total panjang rute minimum. Struktur ini menunjukkan bahwa pada kondisi statis, topologi rute yang dibangun oleh algoritma *MST* bersifat efisien dan stabil, sekaligus konsisten dengan sebaran geografis kampus yang membentuk pola hub-and-spoke dengan UI sebagai pusat.

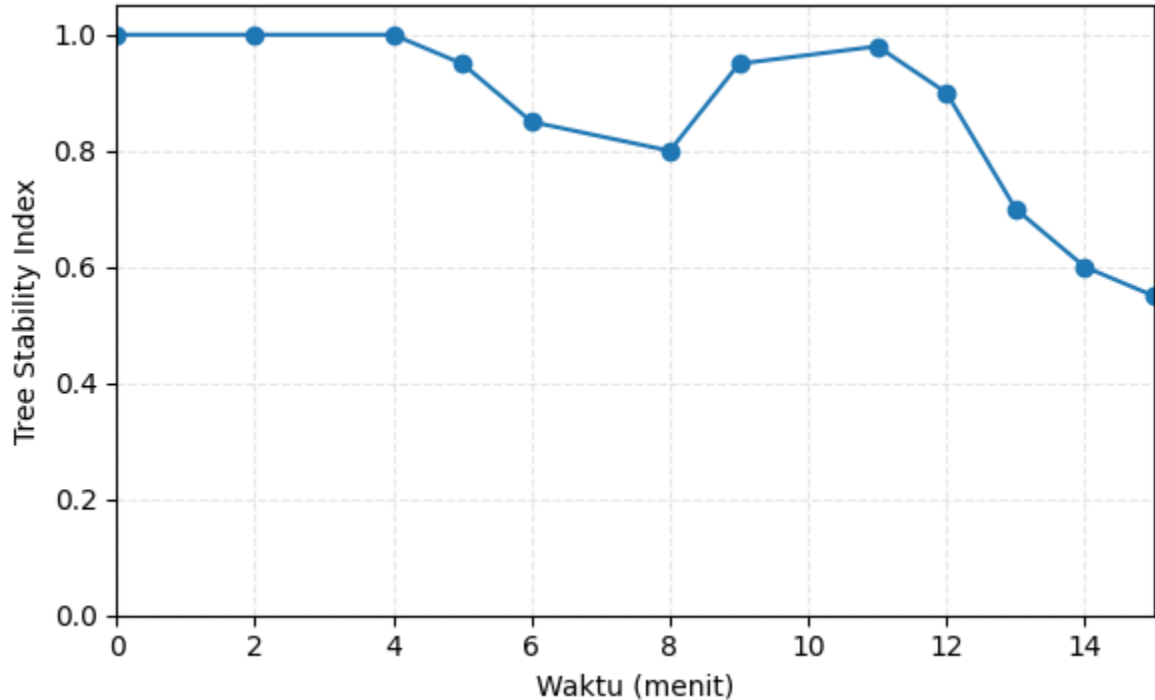
3.2 Kinerja Skenario Gerak Linier

Pada skenario gerak linier terkoordinasi, data posisi yang merepresentasikan perpindahan antarkampus dengan kecepatan sekitar 10 m/s selama 15 menit diputar ulang (*replay*) melalui sistem sehingga *server* menerima deret koordinat yang berubah secara teratur dan melakukan perhitungan *MST* pada setiap siklus, menghasilkan 900 siklus pembaruan. Rata-rata waktu komputasi *MST* meningkat sedikit menjadi 3,124 milidetik dengan simpangan baku 0,521 milidetik, sedangkan *latency* ujung ke ujung naik menjadi sekitar 201,7 milidetik dengan rentang 156,3–342,8 milidetik.

Tabel 3. Kinerja skenario gerak linier (S2) selama 900 siklus

Metrik	Rata-rata	Simpangan Baku	Minimum	Maksimum
Waktu komputasi <i>MST</i> (ms)	3,124	0,521	2,254	5,893
<i>Latency</i> ujung ke ujung (ms)	201,7	31,4	156,3	342,8
<i>Tree Stability Index</i>	0,947	0,142	0,500	1,000
Jumlah peristiwa rekonfigurasi	48	–	–	–

Nilai *Tree Stability Index* sebesar 0,947 menunjukkan bahwa dalam 94,7% siklus pembaruan, struktur *MST* yang terbentuk tetap sama, dan hanya 48 dari 900 siklus (5,3%) yang mengalami rekonfigurasi sisi akibat perubahan jarak relatif antarnode. Perubahan struktur pohon terutama terjadi ketika beberapa *node* melintasi batas keputusan, yaitu kondisi ketika dua atau lebih jalur memiliki panjang yang hampir sama sehingga pergantian sisi menjadi menguntungkan secara metrik jarak. Peningkatan *latency* rata-rata yang relatif kecil dibandingkan skenario statis mengindikasikan bahwa mobilitas linier yang terkoordinasi tidak menimbulkan lonjakan keterlambatan yang signifikan, dan sistem masih mampu mengakomodasi perubahan topologi secara real-time.



Gambar 4. Evolusi Tree Stability Index selama skenario gerak linier

Gambar 4 menggambarkan perubahan *Tree Stability Index* sepanjang 15 menit pengujian skenario gerak linier, yang dibagi secara konseptual menjadi empat fase. Pada menit 0–5, nilai *TSI* berada sangat dekat dengan 1,0 sehingga struktur *MST* sepenuhnya stabil; pada menit 5–8, *TSI* mulai berfluktuasi di rentang sekitar 0,8–0,95 yang mencerminkan zona transisi ketika beberapa sisi alternatif mulai bersaing; pada menit 8–12, *TSI* kembali meningkat mendekati 1,0 dan pohon relatif stabil; sedangkan pada menit 12–15, *TSI* turun ke kisaran 0,55–0,70 akibat frekuensi rekonfigurasi yang lebih sering ketika geometri relatif antarnode menjadi lebih seimbang. Grafik ini memperlihatkan bahwa perubahan rute terkonsentrasi pada beberapa jendela waktu tertentu, bukan tersebar merata sepanjang eksperimen.

3.3 Analisis Perbandingan dan Implikasi

Perbandingan antara skenario statis dan gerak linier menunjukkan bahwa komponen waktu komputasi *MST* pada *cloud server* selalu jauh lebih kecil daripada interval pembaruan posisi (2 Hz), sehingga tidak menjadi faktor pembatas dalam *latency* ujung ke ujung. Peningkatan rata-rata waktu komputasi dari 2,847 milidetik pada skenario statis menjadi 3,124 milidetik pada skenario linier masih berada dalam orde beberapa milidetik, jauh di bawah komponen keterlambatan jaringan WiFi–seluler dan proses *parsing* di sisi *client*. Dengan *latency* ujung ke ujung yang konsisten berada di bawah sekitar 250 milidetik, prototipe ini memenuhi kebutuhan banyak aplikasi koordinasi tim skala kecil yang memerlukan respon sub-detik, seperti latihan pencarian di kampus atau uji coba rute patroli.

Dari sisi kestabilan topologi, *Tree Stability Index* yang mencapai 1,0 pada skenario statis dan tetap tinggi (0,947) pada skenario linier menunjukkan bahwa struktur pohon *MST* relatif tidak sensitif terhadap perubahan posisi yang moderat, selama geometri relatif antarnode tidak berubah secara drastis dibandingkan ketidakpastian pengukuran *GPS*. Hal ini sejalan dengan karakteristik *MST* pada jaringan bergerak yang dilaporkan pada literatur, di mana mayoritas siklus pembaruan mempertahankan himpunan sisi yang sama dan hanya sebagian kecil yang memicu

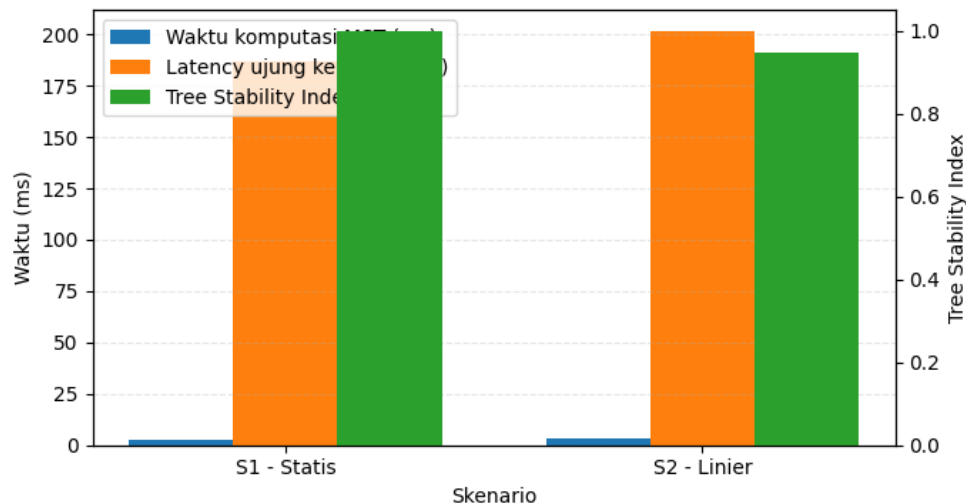
rekonfigurasi ketika *node* melintasi batas keputusan rute (Pantazis et al., 2013; Wang et al., 2016). Temuan ini mengindikasikan bahwa untuk jumlah *node* yang terbatas, pendekatan terpusat berbasis *MST* dengan dukungan *cloud server* cukup tangguh menghadapi perubahan topologi moderat tanpa memerlukan mekanisme *routing* reaktif penuh seperti pada *MANET* yang sangat dinamis.

Tabel 4 merangkum perbandingan kinerja sistem pada skenario statis (S1) dan gerak linier (S2) berdasarkan metrik waktu komputasi *MST*, *latency* ujung ke ujung, *Tree Stability Index*, dan jumlah peristiwa rekonfigurasi.

Tabel 4. Ringkasan perbandingan kinerja skenario S1 dan S2

Skenario	Rata-rata waktu komputasi <i>MST</i> (ms)	Rata-rata <i>latency</i> (ms)	<i>Tree Stability Index</i>	Jumlah peristiwa rekonfigurasi
S1 – Statis	2,847	187,3	1,000	0
S2 – Linier	3,124	201,7	0,947	48

Tabel ini menunjukkan bahwa pergeseran dari topologi statis ke gerak linier hanya meningkatkan waktu komputasi *MST* dan *latency* rata-rata secara marginal, sementara *Tree Stability Index* tetap tinggi meskipun terjadi 48 peristiwa rekonfigurasi pada skenario linier. Hal ini menguatkan bahwa komputasi *MST* pada *cloud server* bukan sumber utama keterlambatan dan bahwa struktur pohon cukup stabil untuk mendukung koordinasi *node* bergerak skala kecil di kedua skenario.



Gambar 5. Grafik perbandingan kinerja skenario S1 dan S2

Gambar 5 menampilkan diagram batang berkelompok untuk dua skenario, S1 (statis) dan S2 (linier), dengan tiga metrik utama: rata-rata waktu komputasi *MST*, rata-rata *latency* ujung ke ujung, dan *Tree Stability Index*. Batang menunjukkan bahwa waktu komputasi *MST* dan *latency* pada S2 sedikit lebih tinggi dibanding S1, tetapi masih dalam orde yang sama, sedangkan *Tree Stability Index* tetap tinggi pada kedua skenario meskipun sedikit menurun pada S2 karena adanya 48 peristiwa rekonfigurasi pohon. Grafik ini menguatkan bahwa beban komputasi *MST* di *cloud server* relatif kecil dan struktur pohon tetap stabil meskipun terjadi pergerakan linier terkoordinasi.

4 KESIMPULAN

Penelitian ini telah menyajikan perancangan dan evaluasi prototipe sistem rute dinamis berbasis *Minimum Spanning Tree (MST)* untuk koordinasi beberapa *node* bergerak berbasis *ESP32-GPS* yang terhubung ke *cloud server* melalui *tethering WiFi smartphone* pada skenario topologi kampus di Jakarta Selatan. Sistem yang dikembangkan mengadopsi arsitektur *client-server* sederhana, di mana komputasi graf dan algoritma *MST* dijalankan secara terpusat di *backend* berbasis Python, sementara *node* tepi menangani akuisisi data posisi, pertukaran pesan *JSON*, dan penampilan instruksi rute pada *LCD*.

Hasil pengujian pada dua skenario mobilitas menunjukkan bahwa waktu komputasi *MST* di *cloud server* selalu berada pada orde beberapa milidetik, dengan rata-rata sekitar 2,847 milidetik pada skenario statis dan 3,124 milidetik pada skenario gerak linier, sehingga jauh lebih kecil dibandingkan komponen keterlambatan jaringan. *Latency* ujung ke ujung konsisten berada di bawah sekitar 250 milidetik (187,3 milidetik pada kondisi statis dan 201,7 milidetik pada skenario linier), yang masih memadai untuk aplikasi koordinasi tim bergerak skala kecil yang membutuhkan respon sub-detik. Dari sisi kestabilan topologi, *Tree Stability Index* mencapai 1,0 pada skenario statis dan tetap tinggi (0,947) pada skenario linier, dengan hanya 48 peristiwa rekonfigurasi dari 900 siklus, yang mengindikasikan bahwa struktur *MST* relatif stabil terhadap perubahan posisi yang moderat.

Temuan-temuan tersebut mengarah pada kesimpulan bahwa pendekatan terpusat berbasis *MST* dengan dukungan *cloud server* layak digunakan sebagai solusi awal untuk koordinasi tim bergerak skala kecil di lingkungan kampus atau skenario serupa, selama jumlah *node* dan tingkat dinamika mobilitas masih dalam batas yang moderat. Meskipun demikian, prototipe yang dikembangkan masih memiliki keterbatasan, antara lain ketergantungan pada satu *cloud server* dan cakupan skenario mobilitas yang hanya mencakup konfigurasi statis dan gerak linier. Penelitian lanjutan perlu mengkaji skenario mobilitas yang lebih kompleks, mengevaluasi konsumsi energi setiap *node*, serta mengeksplorasi pendekatan hierarkis atau prediktif untuk meningkatkan skalabilitas dan ketahanan sistem pada kondisi jaringan yang lebih menantang, sejalan dengan rekomendasi kajian *routing* energi-efisien dan topologi dinamis pada jaringan sensor nirkabel.

DAFTAR PUSTAKA

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4), 393–422. [https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4)[2]
- Asif, M., Khan, S., Ahmad, R., Sohail, M., & Singh, D. (2017). Quality of service of routing protocols in wireless sensor networks: A review. *IEEE Access*, 5, 1913–1933. <https://doi.org/10.1109/ACCESS.2017.2654527>
- Das, S., & Roy, S. (2012). Distributed algorithms for minimum spanning tree in wireless sensor networks. *Wireless Networks*, 18(4), 405–419. <https://doi.org/10.1007/s11276-011-0394-2>
- Hu, H., Han, Y., Yao, Y., & Wang, L. (2021). Trust-based secure and energy efficient routing protocol for wireless sensor networks. *IEEE Access*, 9, 84132–84141. <https://doi.org/10.1109/ACCESS.2021.3087697>
- Kumar, A., Zhao, M., Wong, K.-J., Guan, Y. L., & Chong, P. H. J. (2020). A comprehensive study of IoT and WSN MAC protocols: Research issues, challenges and opportunities. *IEEE Access*, 8, 94183–94203. <https://doi.org/10.1109/ACCESS.2020.2995357>

- Li, X., & Canovas, F. (2001). Kinetic spanning trees for minimum-power routing in MANETs. In *Proceedings of the IEEE Conference* (Document 1002849). IEEE.
- Lin, D., Kumar, P., & Zhang, Y. (2023). CMSTR: A constrained minimum spanning tree based routing protocol for wireless sensor networks. *Computer Communications*, 204, 89–101. <https://doi.org/10.1016/j.comcom.2023.03.015>
- López, I., Yáñez, C., & Camacho, O. (2013). BDD-based algorithm for the minimum spanning tree in wireless ad-hoc network routing. *Wireless Networks*, 19(6), 1293–1305. <https://doi.org/10.1007/s11276-012-0529-1>
- Pantazis, N. A., Nikolidakis, S. A., & Vergados, D. D. (2013). Energy-efficient routing protocols in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 15(2), 551–591. <https://doi.org/10.1109/SURV.2012.062612.00084>
- Ragavan, P. S., & Ramasamy, K. (2019). Optimized routing in wireless sensor networks by establishing dynamic topologies based on genetic algorithm. *Cluster Computing*, 22(Suppl 5), 12119–12125. <https://doi.org/10.1007/s10586-017-1565-1>
- Royer, E. M., & Toh, C.-K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2), 46–55. <https://doi.org/10.1109/98.760423>
- Stojmenovic, I., Ruhil, A. P., & Lobiyal, D. K. (2008). Localized minimum spanning tree-based multicast routing with energy-efficient guaranteed delivery in ad hoc and sensor networks. In *Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS 2008)* (Document 4594821). IEEE. <https://doi.org/10.1109/IPDPS.2008.4536482>
- Wang, X., Huang, M., & Chen, Y. (2016). Minimum spanning tree-based multicast routing protocol for dynamic spectrum access networks: A multi-layer probabilistic approach. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* (pp. 1–6). IEEE. <https://doi.org/10.1109/PIMRC.2016.7549462>