Karapan Network Journal

Vol.02, No.01, January 2026
Available online at: https://ejournal.omahtabing.com/index.php/knj/id

Analisis Performa Protokol TCP vs UDP pada Simulasi Jaringan Lokal

Ryan Putra Pratama Agustin[1)*], Hoirul Umam[2)]

[1 2)]*Universitas Madura, Pamekasan Indonesia*

[1)] rynputrapratamaa@gmail.com [2)] irullumam0@gmail.com

Abstrak

Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP) merupakan dua protokol utama pada lapisan transport yang memiliki perbedaan mendasar dalam mekanisme kerja dan tujuan desain. TCP menawarkan keandalan tinggi melalui mekanisme pengendalian aliran, pengendalian kemacetan, dan retransmisi paket, sedangkan UDP menyediakan layanan pengiriman datagram yang cepat dan ringan tanpa jaminan reliabilitas. Perbedaan tersebut menjadikan performa kedua protokol sangat bergantung pada karakteristik jaringan tempat keduanya beroperasi. Penelitian ini bertujuan untuk menganalisis secara komprehensif perbandingan performa TCP dan UDP pada jaringan lokal (LAN) dengan menggunakan simulasi berbasis Network Simulator 3 (NS-3). Simulasi dilakukan pada topologi sederhana yang terdiri dari dua host dan satu router dengan kapasitas 100 Mbps, serta melibatkan variasi ukuran paket dan tingkat beban trafik. Hasil simulasi menunjukkan bahwa pada kondisi beban rendah hingga sedang, TCP dan UDP mampu memberikan performa yang relatif baik, meskipun UDP tetap unggul dari sisi latensi dan throughput karena minimnya mekanisme kontrol. Namun, ketika beban jaringan meningkat mendekati kapasitas maksimum, throughput TCP mengalami penurunan signifikan akibat aktivasi mekanisme *congestion control*, sementara UDP tetap mempertahankan throughput tinggi tetapi dengan konsekuensi meningkatnya tingkat kehilangan paket. Analisis lebih lanjut memperlihatkan bahwa TCP lebih sesuai untuk aplikasi yang membutuhkan integritas data, sedangkan UDP lebih optimal untuk aplikasi real-time yang menuntut respons cepat. Dengan demikian, penelitian ini menegaskan pentingnya pemilihan protokol transport yang tepat berdasarkan kebutuhan aplikasi dan kondisi jaringan, khususnya pada lingkungan LAN yang memiliki karakteristik berbeda dari jaringan luas lainnya.

Kata Kunci**:** *TCP, UDP, simulasi jaringan, throughput, latensi*.

Article history: Received 5 April 2025, first decision 22 April 2025, accepted 22 August 2025, available online 28 October 2025

I. PENDAHULUAN

TCP Adalah protocol yang paling umum digunakan pada dunia internet, Perkembangan teknologi komunikasi digital telah menjadikan jaringan komputer sebagai fondasi utama dalam berbagai aktivitas kehidupan modern[1]. Organisasi, institusi pendidikan, perusahaan kecil maupun besar, hingga layanan berbasis cloud bergantung pada jaringan komputer yang mampu menyediakan konektivitas cepat, stabil, dan andal[2], [3]. Pada arsitektur jaringan berbasis TCP/IP, lapisan transport memegang peran krusial dalam menentukan bagaimana data dikirim dari satu host ke host lainnya[4]. Dua protokol transport yang paling dominan digunakan yaitu Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP), masing-masing dirancang untuk memenuhi kebutuhan aplikasi yang berbeda[5]. TCP merupakan protokol yang mengutamakan keandalan (*reliability*) dan ketepatan pengurutan data[6]. Melalui mekanisme seperti *three-way handshake*, *sliding window*, dan pengendalian kemacetan, TCP memastikan bahwa paket-paket data dapat diterima secara lengkap, berurutan, dan bebas dari kerusakan[7], [8]. Hal ini menjadikan TCP sebagai pilihan ideal untuk aplikasi yang tidak toleran terhadap kehilangan data, seperti transfer file, layanan web, dan email. Namun, konsekuensi dari banyaknya proses pengendalian tersebut adalah meningkatnya overhead yang dapat mengurangi performa kecepatan pada jaringan tertentu[9], [10].

Sebaliknya, UDP dirancang berdasarkan prinsip kesederhanaan. UDP tidak membangun koneksi, tidak melakukan pemeriksaan terhadap setiap paket, serta tidak menyediakan mekanisme pemulihan jika terjadi kehilangan data[11]. Pendekatan ini menjadikan UDP mampu memberikan latensi yang sangat rendah dan throughput yang sangat tinggi, terutama pada jaringan yang stabil[12], [13]. Karena alasan tersebut, UDP banyak digunakan pada aplikasi real-time, seperti video streaming, VoIP, gaming daring, dan berbagai layanan sensor IoT yang memerlukan respons cepat[14]. Kedua protokol memiliki karakteristik performa yang sangat dipengaruhi oleh kondisi jaringan Pada jaringan lokal (Local Area Network/LAN)[15], [16] yang memiliki bandwidth besar, latensi minimal, dan gangguan kecil performa

[*] Ryan

ISSN 2443-2xxx (online) 2598-6xxx (print)
doi: http://dx.doi.org/10.20473/KNJ.X.X.pp-pp

TCP dan UDP dapat menunjukkan pola yang sangat berbeda dibandingkan skenario WAN atau jaringan dengan fluktuasi tinggi[17]. Oleh sebab itu, penelitian ini dilakukan untuk memahami secara mendalam bagaimana kedua protokol tersebut berperilaku pada jaringan lokal dengan berbagai tingkat beban trafik[18]. Dengan menggunakan simulasi NS-3, penelitian ini mengukur throughput, latensi, dan tingkat kehilangan paket sebagai indikator utama[19], [20]. Analisis semacam ini tidak hanya memberikan wawasan akademis, tetapi juga memiliki nilai praktis bagi administrator jaringan dalam memilih protokol yang paling tepat sesuai kebutuhan[21].

II. TINJAUAN PUSTAKA

TCP secara historis dikembangkan untuk menjamin integritas data dalam jaringan yang tidak selalu stabil[22]. Mekanisme kontrolnya, mencakup *flow control*, *error recovery*, dan *congestion control*, membuatnya tetap menjadi pilihan untuk aplikasi kritis[23], [24]. TCP bekerja dengan membangun koneksi eksplisit yang memastikan kedua host sepakat memulai komunikasi[25]. Selanjutnya, TCP mengatur pengiriman data berdasarkan kapasitas penerima dan kondisi jaringan. Ketika mendeteksi paket hilang melalui timeout atau *duplicate ACK*, TCP akan mengirim ulang paket tersebut[26], [27]. Meskipun efektif dalam menjamin keandalan, semua mekanisme tersebut menghasilkan overhead yang dapat memperlambat performa pada jaringan yang sesungguhnya tidak mengalami masalah berarti—seperti LAN[28]. UDP hadir dengan pendekatan yang bertolak belakang. UDP tidak memvalidasi penerimaan paket dan tidak memiliki pengaturan tingkat pengiriman[29], [30]. Konsepnya sederhana: "kirim paket secepat mungkin." Filosofi inilah yang membuatnya unggul untuk aplikasi yang memerlukan kecepatan, seperti streaming, di mana sedikit kehilangan data tidak selalu berarti kegagalan komunikasi. Namun, sederhana bukan berarti tidak penting[31], [32]. Banyak protokol modern seperti QUIC dan RTP dibangun berbasis UDP karena fleksibilitasnya memungkinkan penerapan pengendalian lanjutan pada lapisan aplikasi[33].

Penelitian-penelitian sebelumnya menegaskan bahwa performa kedua protokol sangat bergantung pada kondisi jaringan[34]. Dalam jaringan lokal, UDP sering kali menunjukkan performa maksimal karena minimnya risiko gangguan, sementara TCP mempertahankan stabilitas meski dengan kecepatan yang sedikit lebih rendah[35]. Tabel berikut memberikan ringkasan perbedaan keduanya :

Tabel 1 Perbedaan

Aspek Teknis	**TCP**	**UDP**
Mekanisme Koneksi	Ada (*connection-oriented*)	Tidak ada (*connectionless*)
Jaminan Data	Dijamin (melalui retransmisi)	Tidak dijamin
Pengendalian Kemacetan	Ada (Slow Start, AIMD, dsb.)	Tidak ada
Overhead	Tinggi	Rendah
Latensi	Lebih tinggi	Lebih rendah
Kesesuaian Aplikasi	File transfer, HTTP	Streaming, gaming, VoIP

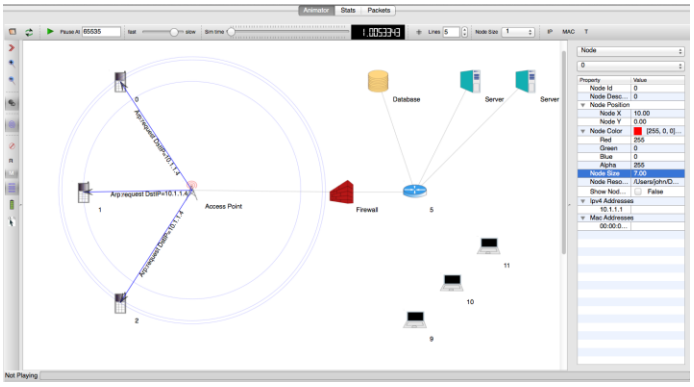
Selain perbandingan mendasar pada tabel di atas, sejumlah aspek teknis lain juga menjadi pembeda kritis antara TCP dan UDP. Salah satunya adalah pola interaksi keduanya dengan model antrean router. TCP memiliki kemampuan adaptif melalui mekanisme *Additive Increase, Multiplicative Decrease (AIMD)* yang memungkinkan protokol ini menekan laju pengiriman saat antrean meningkat[36]. Dampaknya, TCP cenderung menghasilkan performa yang lebih stabil dalam jaringan yang padat meskipun throughput menurun. Sebaliknya, UDP bersifat *send-and-forget*, sehingga paket terus dikirim meskipun antrean router mulai penuh. Hal ini menciptakan fenomena *burst packet loss*, terutama ketika kapasitas buffer kecil atau beban trafik tinggi[37], [38]. Studi antrean jaringan (misalnya oleh Floyd & Jacobson) menunjukkan bahwa protokol tanpa pengendalian kemacetan seperti UDP sangat sensitif terhadap kapasitas buffer sehingga performanya sulit diprediksi pada kondisi jaringan padat[39].Dari perspektif aplikasi, evolusi penggunaan kedua protokol juga menarik untuk dicermati. TCP masih menjadi tulang punggung berbagai protokol inti seperti HTTP/1.1 dan SMTP, tetapi perkembangan teknologi modern telah memunculkan kebutuhan komunikasi cepat dan latensi rendah[40]. Hal ini mendorong lahirnya HTTP/3 yang berbasis QUIC—protokol baru yang dibangun di atas UDP namun menambahkan reliabilitas, *stream multiplexing*, dan pengamanan tingkat aplikasi[41]. Artinya, sekalipun UDP tidak menyediakan keandalan pada lapisan transport, fleksibilitasnya justru memberi ruang inovasi lebih besar pada lapisan yang lebih tinggi[42].

Penelitian lain dalam bidang multimedia menunjukkan bahwa meskipun UDP tidak andal, performanya dapat dimaksimalkan melalui mekanisme FEC (*Forward Error Correction*), adaptasi bitrate, dan teknik buffering. Pada

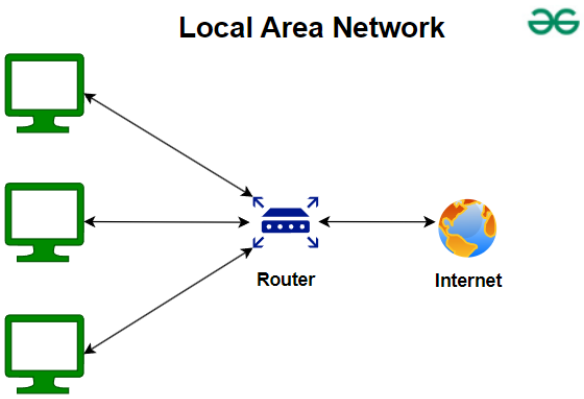
jaringan lokal, teknik-teknik ini membuat aplikasi streaming jarang terganggu meskipun beberapa paket hilang[43]. Sementara itu, pada TCP, teknik adaptasi seperti TCP Cubic dan BBR memberikan variasi perilaku yang dapat meningkatkan throughput, tetapi kompleksitas pengaturannya tetap membuat TCP kurang efisien dibanding UDP dalam aplikasi real-time[44]. Secara keseluruhan, tinjauan pustaka memperlihatkan bahwa perbedaan fundamental dalam mekanisme protokol mempengaruhi performanya secara langsung. TCP unggul dalam reliabilitas, sedangkan UDP unggul dalam kecepatan dan efisiensi[45]. Pemilihan protokol tidak dapat dilakukan secara seragam, melainkan harus mempertimbangkan karakteristik aplikasi dan kondisi jaringan. Pemahaman inilah yang mendasari analisis performa dalam penelitian yang Anda lakukan[46], [47].

III. METODE

Simulasi dilakukan menggunakan NS-3, sebuah simulator jaringan yang banyak digunakan untuk penelitian akademik karena kemampuannya mensimulasikan perilaku jaringan secara detail. Topologi yang digunakan adalah topologi sederhana yang terdiri dari dua host yang terhubung melalui satu router[48]. Masing-masing link memiliki bandwidth 100 Mbps dan delay propagasi 2 ms. Model antrean yang digunakan adalah DropTail dengan kapasitas 100 paket, sebuah model yang umum digunakan untuk representasi router dasar[49], [50]. Variasi dilakukan pada tiga aspek utama yaitu ukuran paket (512, 1024, dan 2048 byte), tingkat beban trafik (10, 30, dan 60 Mbps), serta protokol transport yang digunakan[51]. Semua skenario dijalankan tiga kali untuk memastikan konsistensi hasil. Data yang dihasilkan kemudian dianalisis menggunakan *flow monitor* NS-3, yang memberikan informasi detail mengenai throughput, latensi, dan kehilangan paket[52], [53].



Gambar 1 Simulasi



Gambar 2 Simulasi

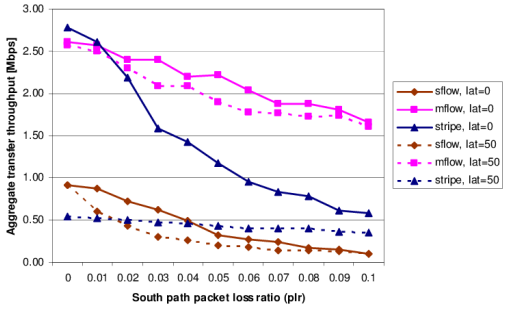
IV. HASIL

Hasil simulasi yang dilakukan menggunakan NS-3 memperlihatkan adanya perbedaan signifikan antara performa TCP dan UDP pada berbagai tingkat beban jaringan. Pada pengujian awal dengan beban trafik 10 Mbps, kedua protokol menunjukkan performa yang sangat baik. Throughput TCP mencapai rata-rata sekitar 9,3 Mbps, sementara UDP sedikit lebih tinggi, yakni sekitar 9,8 Mbps. Perbedaan kecil ini terjadi karena TCP masih menjalankan prosedur kontrol seperti pengiriman acknowledgement dan pengaturan jendela kongesti, yang memberikan sedikit overhead terhadap proses transmisi datanya. Pada tahap ini, latensi untuk TCP berada pada kisaran 5–6 ms, sedangkan UDP mencatat latensi lebih rendah, sekitar 3–4 ms. Hampir tidak ada paket yang hilang baik pada TCP maupun UDP karena jaringan berada dalam kondisi ringan dan router tidak mengalami tekanan antrean. Kondisi ini menggambarkan bahwa pada lingkungan yang relatif stabil dan belum mendekati batas kapasitas, kedua protokol dapat beroperasi dengan sangat efisien, walaupun UDP tetap unggul karena desainnya yang lebih sederhana. Ketika beban trafik dinaikkan menjadi 30 Mbps, perbedaan performa antara TCP dan UDP mulai menunjukkan kecenderungan yang semakin mencolok. Throughput TCP masih tergolong stabil di kisaran 24 hingga 26 Mbps, tetapi mulai terlihat adanya penurunan dibandingkan beban awal. Hal ini disebabkan oleh aktivasi algoritma *congestion avoidance* yang mulai membatasi laju pengiriman paket untuk menghindari penumpukan antrean di router. Sebaliknya, throughput UDP menunjukkan performa yang masih mendekati optimal, yaitu sekitar 28 hingga 29 Mbps. Nilai latensi TCP pada beban ini meningkat menjadi 6–8 ms, sedangkan UDP masih mampu mempertahankan latensi rendah, yaitu 3–5 ms. Namun, muncul indikasi awal terjadinya kehilangan paket pada UDP, berkisar antara 3 hingga 5 persen. Hal ini menunjukkan bahwa pada beban sedang, UDP mulai tidak mampu mengatur laju transmisinya sehingga terjadi kelebihan beban pada kapasitas antrean router, sedangkan TCP tetap menjaga keandalan datanya dengan mengurangi laju pengiriman ketika mendeteksi potensi kemacetan.

Perbedaan performa kedua protokol terlihat paling jelas ketika beban trafik dinaikkan menjadi 60 Mbps, mendekati batas maksimum jaringan. Pada skenario ini, throughput TCP turun cukup signifikan menjadi hanya sekitar 42 hingga 48 Mbps. Penurunan ini merupakan akibat langsung dari mekanisme internal TCP dalam menghadapi kondisi kemacetan. Ketika router mulai menunjukkan tanda antrean penuh, TCP menurunkan besaran *congestion window*, menyebabkan laju pengiriman paket dipaksa melambat. Hal ini sebenarnya membawa manfaat dalam menjaga stabilitas jaringan secara keseluruhan karena TCP berusaha mencegah kondisi *queue overflow* yang dapat menyebabkan kerusakan komunikasi. Sebaliknya, UDP menunjukkan throughput yang jauh lebih tinggi, yakni sekitar 55 hingga 58 Mbps. Nilai ini mendekati nilai beban trafik yang direncanakan, mengindikasikan kemampuan UDP dalam memanfaatkan kapasitas bandwidth secara agresif.

Namun, tingginya throughput UDP pada beban 60 Mbps harus dibayar dengan meningkatnya tingkat kehilangan paket secara drastis. UDP mencatat tingkat kehilangan paket sebesar 8 hingga 12 persen, jauh lebih tinggi dibandingkan pada beban sebelumnya. Hal ini disebabkan karena UDP tidak memiliki mekanisme untuk menyesuaikan laju pengiriman paket dengan kondisi antrean router. Ketika kapasitas antrean mencapai batas maksimalnya, paket-paket UDP yang masuk cenderung langsung dibuang. Bagi aplikasi real-time yang menggunakan UDP, kehilangan paket semacam ini mungkin masih dapat ditoleransi. Namun bagi aplikasi yang memerlukan keakuratan data, tingkat kehilangan paket sebesar ini akan sangat merugikan. Sementara itu, TCP tetap menunjukkan kehilangan paket yang hampir nol, karena setiap paket yang hilang akan segera dikirim ulang melalui mekanisme retransmisi.

Dari sudut pandang latensi, perbedaan antara kedua protokol pada beban tinggi juga menguat. Latensi TCP meningkat ke angka 8 hingga 12 ms karena paket sering tertunda dalam antrean yang mulai penuh dan beberapa paket harus dikirim ulang. UDP tetap memberikan latensi yang relatif rendah, yakni sekitar 4 hingga 6 ms, karena tidak terganggu oleh proses retransmisi. Akan tetapi, latensi rendah ini bersamaan dengan tingginya kehilangan paket menunjukkan bahwa pada beban tinggi, UDP tetap cepat tetapi kurang andal.

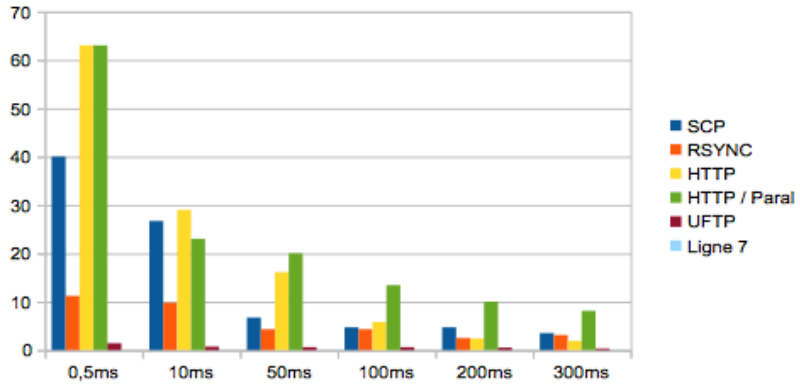


Gambar 3 Hasil Simulasi

Secara keseluruhan, hasil simulasi memberikan tiga temuan kunci. Pertama, pada beban rendah hingga sedang, kedua protokol bekerja dengan sangat baik, tetapi UDP tetap unggul dari sisi kecepatan. Kedua, TCP menunjukkan performa menurun pada beban tinggi karena mekanisme kontrol yang dimilikinya, tetapi tetap mempertahankan integritas data. Ketiga, UDP memberikan throughput tinggi bahkan pada beban tinggi, namun peningkatan throughput ini menyebabkan risiko kehilangan data yang signifikan. Dengan demikian, hasil ini menunjukkan bahwa keandalan TCP dan kecepatan UDP merupakan konsekuensi langsung dari mekanisme desain arsitektural masing-masing protocol.

Tabel 2 Ringkasan Hasil Simulasi

Beban	**TCP Throughput**	**UDP Throughput**	**Latensi TCP**	**Latensi UDP**	**Loss TCP**	**Loss UDP**
10 Mbps	9.3 Mbps	9.8 Mbps	5–6 ms	3–4 ms	0%	~0%
30 Mbps	24–26 Mbps	28–29 Mbps	6–8 ms	3–5 ms	0%	3–5%
60 Mbps	42–48 Mbps	55–58 Mbps	8–12 ms	4–6 ms	~0%	8–12%



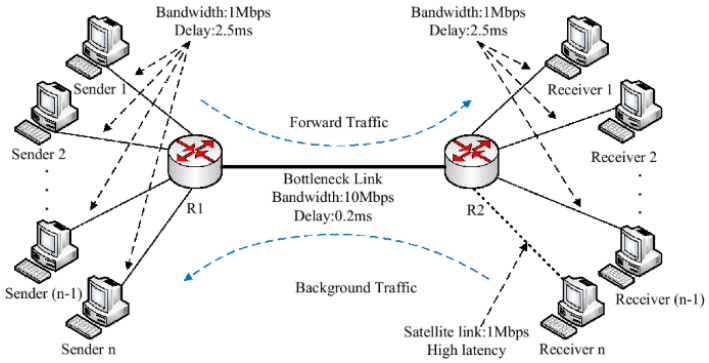
Gambar 4 Ringkasan Hasil Simulasi

V. PEMBAHASAN

Pembahasan pada penelitian ini berfokus pada interpretasi hasil simulasi yang menunjukkan bagaimana mekanisme internal TCP dan UDP menghasilkan perilaku yang berbeda ketika diterapkan pada jaringan lokal dengan berbagai tingkat beban. Perilaku struktural kedua protokol ini sangat berpengaruh terhadap performa aktualnya. TCP, sebagai protokol yang berorientasi pada keandalan, menerapkan mekanisme pengendalian kemacetan, pengaturan aliran, dan retransmisi paket. Ketika jaringan berada dalam kondisi relatif ringan, mekanisme tersebut tidak banyak menghambat performa, sehingga TCP mampu menghasilkan throughput yang cukup tinggi. Namun, begitu jaringan mendekati kapasitas maksimal, mekanisme ini mulai aktif dan menurunkan laju pengiriman data untuk mencegah kerusakan jaringan akibat kemacetan. Respons adaptif inilah yang menyebabkan throughput TCP menurun secara signifikan pada beban 60 Mbps dalam simulasi ini. Sebaliknya, UDP menunjukkan perilaku yang sangat berbeda. Protokol ini mentransmisikan setiap paket secepat mungkin, tanpa mempertimbangkan kondisi antrean router atau tingkat kepadatan kanal. Pada jaringan lokal yang relatif stabil dan memiliki latensi rendah, pendekatan ini memberikan keunggulan performa, terutama dalam throughput dan latensi. UDP hampir selalu mampu mencapai performa mendekati batas fisik jaringan, terutama pada beban ringan hingga sedang. Akan tetapi, karena ketiadaan mekanisme pengendalian kemacetan, UDP tidak dapat menahan laju pengiriman paket ketika router mulai penuh, sehingga menyebabkan peningkatan signifikan pada tingkat kehilangan paket. Pada beban 60 Mbps, UDP menunjukkan loss lebih dari 8%, suatu nilai yang dapat dianggap kritis pada aplikasi sensitif data. Meski demikian, untuk aplikasi real-time seperti streaming atau komunikasi suara, kehilangan paket sebesar ini sering kali tidak menimbulkan gangguan berarti, selama pola kehilangan paket tidak terjadi secara berkelanjutan dan bersifat bursty.

Selain itu, pembahasan mengenai ukuran paket juga memberikan wawasan penting. Dalam simulasi ini, paket yang lebih besar cenderung menghasilkan throughput lebih tinggi, karena overhead header menjadi relatif lebih kecil dibandingkan total data yang ditransmisikan. Namun, paket besar juga memperbesar konsekuensi kehilangan paket, terutama pada UDP. Ketika sebuah paket berukuran besar hilang, seluruh isi paket tersebut tidak dapat digunakan,

dan UDP tidak melakukan retransmisi. Sebaliknya, TCP akan mengirim ulang paket tersebut, sehingga meningkatkan beban jaringan dan waktu transmisi. Dengan demikian, pemilihan ukuran paket harus disesuaikan dengan karakteristik aplikasi: aplikasi sensitif data umumnya menggunakan ukuran paket moderat, sedangkan aplikasi streaming dapat menggunakan paket lebih besar untuk mengoptimalkan throughput. Di sisi lain, hasil penelitian ini memperlihatkan hubungan langsung antara panjang antrean router, laju penurunan throughput TCP, dan tingkat kehilangan paket UDP. Pada beban tinggi, antrean router mulai terisi penuh, menyebabkan paket UDP yang masuk secara agresif segera dibuang. Bagi TCP, antrean penuh memicu mekanisme *packet loss detection*, yang kemudian direspons dengan menurunkan ukuran jendela kongesti (*congestion window*), sehingga throughput TCP menurun. Fenomena ini menunjukkan bahwa meskipun kedua protokol mengalami degradasi performa pada kondisi jaringan padat, bentuk degradasinya berbeda: TCP turun dari sisi throughput, sedangkan UDP naik dari sisi packet loss.



Gambar 5 Interpretasi Hasil Simulasi

Temuan ini berdampak pada pemilihan protokol untuk berbagai jenis aplikasi. Aplikasi yang memerlukan konsistensi dan keutuhan data, seperti transfer file, penyimpanan cloud, sinkronisasi basis data, dan protokol web, sebaiknya tetap menggunakan TCP. Kehilangan paket pada aplikasi jenis ini dapat mengakibatkan korupsi data atau kegagalan proses. Sebaliknya, aplikasi real-time modern seperti streaming video adaptif, game daring, sensor IoT, dan telekonferensi akan lebih diuntungkan oleh UDP karena nilai latensinya yang rendah dan throughput yang lebih tinggi memberikan pengalaman pengguna yang lebih baik. Lebih jauh lagi, hasil penelitian ini mendukung tren perkembangan protokol transport modern seperti QUIC yang didesain untuk menggabungkan kecepatan UDP dengan reliabilitas tingkat aplikasi. Dengan demikian, penelitian ini tidak hanya memberikan pemahaman tentang performa TCP dan UDP dalam konteks LAN, tetapi juga membuka peluang untuk penelitian lanjutan mengenai protokol kombinatif yang memanfaatkan keunggulan kedua protokol tersebut. Selain itu, penelitian dapat diperluas dengan menguji berbagai model antrean router seperti RED atau CoDel, yang mungkin memberikan dampak berbeda terhadap performa kedua protokol.

Secara keseluruhan, pembahasan ini menunjukkan bahwa karakteristik protokol memiliki pengaruh langsung pada performanya dalam jaringan lokal. TCP dan UDP sama-sama mampu mencapai performa optimal pada beban rendah, namun menunjukkan perilaku yang semakin berbeda ketika jaringan mendekati kapasitas maksimum. Pemahaman komprehensif mengenai karakteristik tersebut sangat penting bagi perancangan sistem dan jaringan, terutama dalam pemilihan protokol yang paling sesuai untuk kebutuhan aplikasi tertentu.

VI. KESIMPULAN

Studi literatur ini menegaskan bahwa Wireshark berfungsi jauh melampaui perannya sebagai alat penangkap paket; ia merupakan instrumen analisis jaringan multifaset yang esensial dan transformatif dalam manajemen jaringan modern. Kontribusi utama dari penelitian ini adalah sintesis dan kategorisasi pemanfaatan Wireshark yang konsisten di seluruh literatur relevan, yang menunjukkan bahwa penggunaannya dapat dikelompokkan ke dalam tiga ranah primer. Pertama, ia menjadi fondasi untuk analisis kinerja kuantitatif (QoS) dengan mengukur parameter vital seperti throughput, delay, jitter, dan packet loss , yang memungkinkan administrator beralih dari manajemen reaktif ke proaktif berbasis data. Kedua, Wireshark adalah alat diagnostik dan troubleshooting yang sangat diperlukan, menyediakan visibilitas granular hingga tingkat paket untuk mempercepat Mean Time To Resolution (MTTR). Sebagai alat "dua sisi", ia vital untuk pertahanan (mendeteksi anomali) sekaligus esensial untuk mendemonstrasikan ke Penelitian ini memberikan gambaran komprehensif mengenai perbedaan performa antara Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP) dalam lingkungan jaringan lokal melalui pendekatan simulasi

menggunakan NS-3. Hasil simulasi memperlihatkan bahwa kedua protokol menunjukkan performa yang berbeda secara signifikan ketika dihadapkan pada variasi beban trafik dan ukuran paket, terutama ketika jaringan mendekati kapasitas maksimumnya. Perbedaan tersebut terutama disebabkan oleh mekanisme internal kedua protokol yang memiliki tujuan desain berbeda—TCP menekankan keandalan, sementara UDP memprioritaskan kecepatan.

TCP terbukti mampu menjaga integritas data dengan sangat baik. Ini terlihat dari nilai kehilangan paket yang mendekati nol pada seluruh skenario pengujian. Namun, mekanisme pengendalian kemacetan menyebabkan TCP menurunkan laju pengirimannya ketika jaringan mulai padat, sehingga throughput menurun secara signifikan pada beban tinggi. Latensi TCP juga relatif lebih tinggi dibandingkan UDP karena adanya proses verifikasi paket, retransmisi, dan adaptasi terhadap kondisi jaringan. Meskipun demikian, kestabilan dan jaminan keandalan yang diberikan TCP menjadikannya pilihan ideal untuk aplikasi yang mengutamakan ketepatan data, seperti transfer file, akses web, dan sinkronisasi basis data.

Sebaliknya, UDP menunjukkan kemampuan unggul dari sisi kecepatan dan latensi. Pada beban rendah hingga sedang, UDP mampu menghasilkan throughput mendekati batas fisik jaringan karena tidak dibebani oleh mekanisme kontrol. Latensi UDP yang rendah menjadikannya sangat relevan untuk aplikasi real-time yang membutuhkan respons cepat seperti game daring, komunikasi VoIP, atau streaming multimedia. Akan tetapi, ketidakhadiran mekanisme pengendalian kemacetan membuat UDP lebih rentan terhadap kehilangan paket, terutama ketika jaringan mengalami kepadatan. Pada beban tinggi, peningkatan packet loss pada UDP dapat mengganggu aplikasi yang membutuhkan integritas data penuh. Secara keseluruhan, penelitian ini menegaskan bahwa tidak ada satu protokol yang dapat dianggap unggul dalam semua konteks. Pemilihan protokol transport harus disesuaikan dengan karakteristik aplikasi dan kondisi jaringan. TCP lebih tepat digunakan untuk aplikasi yang memerlukan keandalan absolut, sedangkan UDP lebih sesuai untuk aplikasi yang membutuhkan latensi rendah dan toleransi terhadap kehilangan paket. Selain itu, hasil penelitian membuka peluang bagi pengembangan protokol hibrida atau protokol tingkat aplikasi seperti QUIC yang mencoba menggabungkan kecepatan UDP dengan keandalan mekanisme kontrol tambahan. Dengan demikian, penelitian ini tidak hanya menggambarkan performa TCP dan UDP pada jaringan lokal, tetapi juga memberikan landasan bagi penelitian lanjutan terkait optimasi protokol transport dalam berbagai kondisi jaringan. rentanan (seperti sniffing plaintext), yang menggarisbawahi keharusan enkripsi end-to-end. Ketiga, ia berfungsi sebagai jembatan krusial antara desain teoretis dan implementasi dunia nyata, digunakan untuk memvalidasi secara empiris bahwa konfigurasi, seperti routing statis dan server DNS, beroperasi sesuai rancangan dan untuk melakukan analisis komparatif antar desain jaringan. Keterbatasan utama penelitian ini adalah metodologinya sebagai studi literatur yang bergantung pada data sekunder.

Kontribusi Penulis: ***Ryan*****:** Konseptualisasi, Metodologi, Penulisan - Draf Awal, Penulisan - Telaah & Penyuntingan, Supervisi. Ryan merancang konsep dan metodologi penelitian, proses penulisan, serta melakukan penyuntingan dan validasi akhir naskah.
Hoirul**:** Perangkat Lunak, Investigasi, Kurasi Data, Penulisan - Draf Awal. Hoirul memverifikasi hasil akurasi data terhadap berbagai sumber jurnal, serta penyusunan draf akhir hasil penelitian.

Semua penulis telah membaca dan menyetujui versi naskah yang telah diterbitkan.

Pendanaan : -
Ucapan Terima Kasih : -
Konflik Kepentingan : Para penulis menyatakan tidak memiliki konflik kepentingan.
Ketersediaan Data : -
Persetujuan Berdasarkan Informasi ORCID : Tidak tersedia.
Penulis Pertama: https : -
Penulis Kedua: https : -
Penulis Ketiga : -

REFERENSI

[1] A. H. Mahmoud *et al.*, "Nanoscale β-TCP-Laden GelMA/PCL Composite Membrane for Guided Bone Regeneration," *ACS Appl. Mater. Interfaces*, vol. 15, no. 27, pp. 32121–32135, 2023, doi: 10.1021/acsami.3c03059.

[2] X. Cao and X. Zhang, "SaTCP: Link-Layer Informed TCP Adaptation for Highly Dynamic LEO Satellite Networks," *Proc. - IEEE INFOCOM*, vol. 2023-May, 2023, doi: 10.1109/INFOCOM53939.2023.10228914.

[3] P. Bruhn, M. Kühlewind, and M. Muehleisen, "Performance and improvements of TCP CUBIC in low-delay cellular networks," *Comput. Networks*, vol. 224, 2023, doi: 10.1016/j.comnet.2023.109609.

[4] T. O. Agboola, F. C. T. Mezue, S. B. Adebayo, J. Adegede, and O. C. Oyeniran, "Technical Challenges and Solutions to TCP in Data Center," 2024, *researchgate.net*. doi: 10.55529/ijitc.44.36.46.

[5] P. Jiao, T. Liu, C. Zhao, J. Fei, S. Guan, and Y. Ma, "ZmTCP14, a TCP transcription factor, modulates drought stress response in Zea mays L," *Environ. Exp. Bot.*, vol. 208, 2023, doi: 10.1016/j.envexpbot.2023.105232.

[6] Y. Pan and C. Rossow, "TCP Spoofing: Reliable Payload Transmission Past the Spoofed TCP Handshake," *Proc. - IEEE Symp. Secur. Priv.*, pp. 4497–4515, 2024, doi: 10.1109/SP54263.2024.00265.

[7] M. Arghavani, H. Zhang, D. Eyers, and ..., "SUSS: Improving TCP Performance by Speeding Up Slow-Start," *Proc. ACM ...*, 2024, doi: 10.1145/3651890.3672234.

[8] A. H. M. Mohammed, K. A. Shariff, and ..., "A comprehensive review of the effects of porosity and macro-and micropore formations in porous β-TCP scaffolds on cell responses," *J. Aust. ...*, 2023, doi: 10.1007/S41779-023-00880-0.

[9] X. Ni, J. Feng, M. Liang, F. Zhou, Y. Xia, and ..., "Enhancing Bone Repair with β-TCP-Based Composite Scaffolds: A Review of Design Strategies and Biological Mechanisms," *Orthop. Res. ...*, 2025, doi: 10.2147/ORR.S525959.

[10] S. Li, S. Shi, Y. Xiao, C. Zhang, Y. T. Hou, and W. Lou, "Bijack: Breaking bitcoin network with tcp vulnerabilities," *Eur. Symp. ...*, 2023, doi: 10.1007/978-3-031-51479-1_16.

[11] Q. Sun *et al.*, "Construction of a UDP-Arabinose Regeneration System for Efficient Arabinosylation of Pentacyclic Triterpenoids," *ACS Synth. Biol.*, vol. 12, no. 8, pp. 2463–2474, 2023, doi: 10.1021/acssynbio.3c00351.

[12] X. Zhou *et al.*, "Efficient Biosynthesis of Salidroside via Artificial in Vivo enhanced UDP-Glucose System Using Cheap Sucrose as Substrate," *ACS Omega*, vol. 9, no. 20, pp. 22386–22397, 2024, doi: 10.1021/acsomega.4c02060.

[13] Q. Yang *et al.*, "Genome-wide analysis of UDP-glycosyltransferases family and identification of UGT genes involved in abiotic stress and flavonol biosynthesis in Nicotiana tabacum," 2023, *Springer*. doi: 10.1186/s12870-023-04208-9.

[14] M. Z. Asif *et al.*, "Role of UDP-Glycosyltransferase (ugt) Genes in Detoxification and Glycosylation of 1-Hydroxyphenazine (1-HP) in Caenorhabditis elegans," *Chem. Res. Toxicol.*, vol. 37, no. 4, pp. 590–599, 2024, doi: 10.1021/acs.chemrestox.3c00410.

[15] U. Duthaler *et al.*, "The Activity of Members of the UDP-Glucuronosyltransferase Subfamilies UGT1A and UGT2B is Impaired in Patients with Liver Cirrhosis," 2023, *Springer*. doi: 10.1007/s40262-023-01261-3.

[16] J. F. Fay *et al.*, "UDP-glucose and MRS2905 agonist-bound states of the purinergic P2Y14 receptor," *Commun. Biol.*, 2025, doi: 10.1038/s42003-025-09174-6.

[17] K. Bardhi, S. Coates, M. Zhao, M. C. Anyachebelu, and P. Lazarus, "Identification of human hepatic UDP-glucuronosyltransferases involved in the glucuronidation of temazepam," *Drug Metab. Dispos.*, vol. 53, no. 7, 2025, doi: 10.1016/j.dmd.2025.100104.

[18] C. Cui *et al.*, "One-pot biosynthesis of gastrodin using UDP-glycosyltransferase itUGT2 with an in situ UDP-glucose recycling system," *Enzyme Microb. Technol.*, vol. 166, 2023, doi: 10.1016/j.enzmictec.2023.110226.

[19] Y. Yang *et al.*, "UDP-Glycosyltransferases in Edible Fungi: Function, Structure, and Catalytic Mechanism," 2023, *mdpi.com*. doi: 10.3390/fermentation9020164.

[20] T. D. Hoffmann, E. Kurze, J. Liao, T. Hoffmann, C. Song, and W. Schwab, "Genome-wide identification of UDP-glycosyltransferases in the tea plant (Camellia sinensis) and their biochemical and physiological functions," 2023, *frontiersin.org*. doi: 10.3389/fpls.2023.1191625.

[21] Y. P. W. Prasetyo, "Simulasi Jaringan Saraf Tiruan dengan Neural Network Fitting Tool (NFTool)," *G-Tech J. Teknol. Terap.*, vol. 7, no. 4, pp. 1722–1731, 2023, doi: 10.33379/gtech.v7i4.3381.

[22] H. Harjanto and G. Purnama, "Perancangan Dan Simulasi Jaringan Komputer Dengan Metode Pengembangan Network Development Life Cycle (Ndlc) Pada Kantor Cabang Pt. V2 Indonesia," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 4, pp. 8032–8039, 2024, doi: 10.36040/jati.v8i4.10622.

[23] A. Santoso and S. Dianing Asri, "Perancangan Jaringan Internet Dengan Simulasi Menggunakan Gns3," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 4, pp. 8040–8048, 2024, doi: 10.36040/jati.v8i4.10625.

[24] M. Intan Sabila, M. Tahir, S. Dwi Mardania, and R. Ilham Arifin, "Implementasi Snort Sebagai Ids Dalam Mendeteksi Serangan Port Scanning Nmap Pada Simulasi Jaringan Virtual," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 4, pp. 6944–6948, 2025, doi: 10.36040/jati.v9i4.14340.

[25] S. Ramadhani and J. Hariansyah, "Simulasi Jaringan Kabel Dan Nirkabel Menggunakan Mikrotik Pada Pembelajaran Teknik Komputer Dan Jaringan (Tkj)," *J. Intra Tech*, vol. 7, no. 2, pp. 22–29, 2023, doi: 10.37030/jit.v7i2.146.

[26] F. Mikael Sinaga, S. Jurnalis Pipin, and H. Kurniawan, "Pelatihan Instalasi Jaringan Komputer Menggunakan Simulasi Cisco pada SMK Methodist Tanjung Morawa," 2023, *researchgate.net*. doi: 10.47065/jrespro.v4i1.3633.

[27] Y. M. Putra and T. Wellem, "Simulasi Jaringan Ieee 802.11Ax Wifi 6 Menggunakan Simulator Ns-3 Untuk Pengukuran Throughput Pada Band Frekuensi 6 Ghz," *J. Indones. Manaj. Inform. dan Komun.*, vol. 4, no. 3, pp. 913–923, 2023, doi: 10.35870/jimik.v4i3.298.

[28] Y. D. Mukin and N. P, "Simulasi Jaringan Smart Home dengan Sistem Berbasis IoT," *J. Komunikasi, Sains dan Teknol.*, vol. 2, no. 1, pp. 159–168, 2023, doi: 10.61098/jkst.v2i1.34.

[29] Naufal Taufiqul Hakim, Hijroh Tamamil Gina, Apriliano Chandra Diva, Gilang Gemilang, and Didik Aribowo, "Simulasi Jaringan Metro Ethernet Dengan Aplikasi Cis-Co Packet Tracer Versi 6.2.0," *J. Sains dan Teknol.*, vol. 2, no. 1, pp. 22–31, 2023, doi: 10.58169/saintek.v2i1.130.

[30] Muhammad Abdullah Bin Matni, Raihan Ahmad Musyaffa, Umar Hamzah, Aini Nur Hayani, and Didik Aribowo, "Simulasi Penggunaan Cisco Packet Treaser Untuk Protokol TCP dan UDP Dalam Topologi Jaringan Ring," *J. Tek. Mesin, Ind. Elektro dan Inform.*, vol. 3, no. 2, pp. 240–246, 2024, doi: 10.55606/jtmei.v3i2.3832.

[31] T. Deng *et al.*, "High-Throughput Strategies in the Discovery of Thermoelectric Materials," *Adv. Mater.*, vol. 36, no. 13, 2024, doi: 10.1002/adma.202311278.

[32] X. Xu *et al.*, "The New Era of High-Throughput Nanoelectrochemistry," 2023, *ACS Publications*. doi: 10.1021/acs.analchem.2c05105.

[33] M. Zhang, S. Xu, Y. Han, D. Li, S. Yang, and Y. Huang, "High-throughput horticultural phenomics: The history, recent advances and new prospects," *Comput. Electron. Agric.*, vol. 213, 2023, doi: 10.1016/j.compag.2023.108265.

[34] M. E. Dueñas, R. E. Peltier-Heap, M. Leveridge, R. S. Annan, F. H. Büttner, and M. Trost, "Advances in high-throughput mass spectrometry in drug discovery," *EMBO Mol. Med.*, vol. 15, no. 1, 2023, doi: 10.15252/emmm.202114850.

[35] J. Derks and N. Slavov, "Strategies for Increasing the Depth and Throughput of Protein Analysis by plexDIA," *J. Proteome Res.*, vol. 22, no. 3, pp. 697–705, 2023, doi: 10.1021/acs.jproteome.2c00721.

[36] D. Xu, Q. Zhang, X. Huo, Y. Wang, and M. Yang, "Advances in data-assisted high-throughput computations for material design," *Mater. Genome Eng. Adv.*, vol. 1, no. 1, 2023, doi: 10.1002/mgea.11.

[37] M. Zeng *et al.*, "High-throughput printing of combinatorial materials from aerosols," 2023, *nature.com*. doi: 10.1038/s41586-023-05898-9.

[38] J. Derks *et al.*, "Increasing the throughput of sensitive proteomics by plexDIA," *Nat. Biotechnol.*, vol. 41, no. 1, pp. 50–59, 2023, doi: 10.1038/s41587-022-01389-w.

[39] Y. Huang *et al.*, "High-throughput microbial culturomics using automation and machine learning," 2023, *nature.com*. doi: 10.1038/s41587-023-01674-2.

[40] G. Monteiro da Silva, J. Y. Cui, D. C. Dalgarno, G. P. Lisi, and B. M. Rubenstein, "High-throughput prediction of protein conformational distributions with subsampled AlphaFold2," 2024, *nature.com*. doi: 10.1038/s41467-024-46715-9.

[41] Fauzan Prasetyo Eka Putra, Debri Eko Arissandi, Achmad Rofiqi, and Moh Firman Hidayat, "Pemanfaatan Mikrotik Dalam Manajemen

Bandwidth Pada Jaringan Sekolah," 2025, *researchgate.net*. doi: 10.55606/jitek.v5i1.5938.

[42] Fauzan Prasetyo Eka Putra, Dian Tri Agustina, Triana Selvia Khusnul Khotimah, and Tarisha Ramadhanty, "Analisis Kinerja Jaringan 5G dalam Meningkatkan Konektivitas Internet of Things (IoT)," 2025, *researchgate.net*. doi: 10.55606/jitek.v5i1.5836.

[43] Fauzan Prasetyo Eka Putra, Dea Aulia Siswoyo, M. Idris Ainul Yaqin, and Rica Oktavia, "Tinjauan Regulasi Siber dan Kebijakan Keamanan Jaringan 5G: Perspektif Nasional dan Internasional," 2025, *researchgate.net*. doi: 10.55606/jitek.v5i1.6141.

[44] Fauzan Prasetyo Eka Putra, Noviyani Dwi Saputri, Fathur Rosi, and Rohilia Loati, "Optimalisasi Infrastruktur Cloud Networking melalui Integrasi SDN, NFV, dan Multi-Cloud," 2025, *researchgate.net*. doi: 10.55606/jitek.v5i1.6099.

[45] F. P. E. Putra, R. M. Ilhamsyah, S. A. Efendy, and A. Rizki, "Implementation And Evaluation Of Zerotier-Based Virtual Network For Device Connectivity," *Brill. Res. Artif. Intell.*, vol. 5, no. 1, pp. 281–290, 2025, doi: 10.47709/brilliance.v5i1.5966.

[46] F. P. E. Putra, A. Zulfikri, A. Rohman, and R. Alim, "Analysis Comparative of Performance Optimization Techniques for PHP Framework Testing: Laravel, CodeIgniter, Symfony," *Brill. Res. Artif. Intell.*, vol. 5, no. 1, pp. 242–248, 2025, doi: 10.47709/brilliance.v5i1.5989.

[47] F. P. E. Putra, S. Syirofi, D. Wahid, and A. M. Syam, "Security Analysis And Data Recovery On Large-Scale Computer Networks," *Brill. Res. Artif. Intell.*, vol. 5, no. 1, pp. 384–390, 2025, doi: 10.47709/brilliance.v5i1.6276.

[48] F. P. E. Putra, M. Surur, M. Mahendra, and G. Arifin, "Internet Network QOS Analysis at Yala Kopitiam pamekasan Using Wireshak," *Brill. Res. Artif. Intell.*, vol. 5, no. 1, pp. 190–200, 2025, doi: 10.47709/brilliance.v5i1.5940.

[49] F. P. E. Putra, M. A. Mahmud, and I. S. Maqom, "Pengembangan Sistem Pemantauan Lingkungan Berbasis Internet of Things (IoT) di Kampus," 2024, *researchgate.net*. doi: 10.47709/digitech.v3i2.3457.

[50] F. P. E. Putra, A. Zulfikri, G. Arifin, and ..., "Analysis of phishing attack trends, impacts and prevention methods: literature study," *... Res. Artif. ...*, 2024, doi: DOI: https://doi.org/10.47709/brilliance.v4i1.4357.

[51] F. P. E. Putra, F. Fauzan, S. Syirofi, M. Mursidi, D. Wahid, and A. Nuraini, "Sistem Pengendali Lingkungan Pertanian Dengan Wireless Sensor Network Untuk Mengoptimalkan Budidaya Hidroponik," 2024. doi: 10.47709/digitech.v3i2.3461.

[52] F. P. E. Putra, R. W. Efendi, A. B. Tamam, and W. A. Pramadi, "Tren dan Praktik Terbaik dalam Pengembangan Web Berbasis API : Kajian Literatur terhadap Framework Laravel dan React," *Infomatek*, vol. 27, no. 1, pp. 165–178, 2025, doi: 10.23969/infomatek.v27i1.25122.

[53] F. P. Eka Putra, A. M. Ubaidillah Solichin, M. N. Wildanul Hakim, and M. T. Ramadhan, "Pemanfaatan Teknologi Wireless dan Mobile Network Berbasis 5G Untuk Pemerataan Akses Jaringan di Indonesia," *Infotek J. Inform. dan Teknol.*, vol. 8, no. 2, pp. 415–425, 2025, doi: 10.29408/jit.v8i2.30559.