BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Evaluasi dan analisis yang telah dilakukan terhadap purwarupa Sistem *Smart Hydroponic* Fakultas Ilmu Komputer UPNVJ menghasilkan kesimpulan yang secara langsung menjawab rumusan masalah penelitian, yaitu sebagai berikut:

1. Rancangan Arsitektur Integrasi Sistem

   Arsitektur sistem Internet of Things (IoT) *Smart Hydroponic* berhasil dirancang dan diimplementasikan melalui struktur tiga lapisan utama, yaitu Layer Perangkat IoT, Layer Protokol Komunikasi, dan Server Backend dengan Basis Data. Pada sisi *backend* , integrasi protokol *CoAP* dan *WebSocket* direalisasikan secara asynchronous menggunakan bahasa pemrograman *Python*. Framework FastAPI bertindak sebagai pusat kendali untuk menangani permintaan *WebSocket* dan *HTTP* yang mendukung koneksi full-duplex, sementara pustaka aiocoap diintegrasikan melalui mekanisme lifespan FastAPI. Pendekatan arsitektural ini terbukti berhasil memungkinkan *server CoAP* berjalan berdampingan dan terintegrasi secara mulus dengan *server WebSocket* di dalam satu ekosistem aplikasi.

2. Tingkat Performa Protokol

   Berdasarkan pengujian transmisi data *real-time*, tidak ada satu protokol yang secara absolut unggul di semua kondisi, melainkan sangat bergantung pada volume beban data (*payload*) dan karakteristik pemrosesan.

   - Dari segi *latency* dan *jitter*: Protokol *CoAP* menunjukkan performa waktu transmisi yang lebih baik dibandingkan dengan *WebSocket* pada sebagian besar skenario pengujian. Pada skenario pengiriman data normal, *CoAP* menghasilkan rata-rata *latency* sebesar 269,45 ms, lebih rendah dibandingkan *WebSocket* yang mencapai 996,61 ms. Nilai *jitter CoAP* juga lebih rendah, yaitu 238,45 ms, sedangkan *WebSocket* mencapai 1.339,93 ms. Pada pengujian variasi ukuran *payload*, *CoAP* memperoleh *latency* sebesar 337,37 ms (512 *byte*), 143,06 ms (1024 *byte*), dan 108,83 ms (2048 *byte*), serta *jitter* sebesar 310,72 ms, 73,83 ms, dan 114,86 ms. Sementara itu, *WebSocket* menghasilkan *latency*

sebesar 159,84 ms, 162,76 ms, dan 338,05 ms, serta *jitter* sebesar 175,78 ms, 127,80 ms, dan 274,64 ms pada ukuran *payload* yang sama. Analisis lalu lintas jaringan juga menunjukkan adanya kejadian *TCP* Retransmission dan Duplicate *ACK* pada komunikasi *WebSocket* yang mengindikasikan keterlambatan atau kehilangan paket selama proses transmisi. Kondisi tersebut diduga berkontribusi terhadap peningkatan *latency* dan variasi waktu kedatangan paket.

- Dari segi *throughput*: Protokol *CoAP* secara konsisten menghasilkan *throughput* yang lebih rendah dibandingkan dengan *WebSocket*. Pada skenario pengiriman data normal, *throughput CoAP* sebesar 0,09 kbps, sedangkan *WebSocket* mencapai 0,31 kbps. Pada pengujian variasi ukuran *payload*, *throughput CoAP* meningkat dari 0,15 kbps (512 *byte*) menjadi 0,29 kbps (1024 *byte*) dan 0,50 kbps (2048 *byte*). Sebaliknya, *WebSocket* menghasilkan *throughput* yang lebih tinggi, yaitu 0,19 kbps, 0,34 kbps, dan 0,61 kbps pada masing-masing ukuran *payload*. Hasil ini menunjukkan bahwa *WebSocket* mampu mentransmisikan data dalam jumlah yang lebih besar per satuan waktu, sedangkan *CoAP* menghasilkan *throughput* yang lebih rendah, yang mencerminkan karakteristik protokol dengan overhead komunikasi yang lebih kecil.
- Dari segi *packet loss*: Pada skenario pengiriman data normal, *CoAP* menghasilkan *packet loss* sebesar 15,69%, sedikit lebih rendah dibandingkan dengan *WebSocket* sebesar 17,93%. Pada pengujian variasi ukuran *payload*, *CoAP* memperoleh *packet loss* sebesar 4,76% (512 *byte*) dan menurun menjadi 1,40% (1024 *byte*), namun meningkat drastis menjadi 67,79% pada ukuran 2048 *byte* akibat terjadinya kegagalan komunikasi (Internal Server Error) pada *Node Environment* dan *Node Plant*. Sebaliknya, *WebSocket* menunjukkan nilai *packet loss* yang relatif stabil, yaitu 1,96% (512 *byte*), meningkat menjadi 8,40% (1024 *byte*), kemudian kembali menjadi 1,96% (2048 *byte*). Temuan ini menunjukkan bahwa *CoAP* memiliki performa yang baik pada pengiriman data berukuran kecil hingga sedang, tetapi mengalami penurunan keandalan ketika ukuran *payload* melebihi kapasitas yang

dapat ditangani secara optimal, sedangkan *WebSocket* lebih mampu mempertahankan keberhasilan transmisi pada ukuran *payload* yang lebih besar.

Meskipun *CoAP* menunjukkan performa yang lebih baik pada parameter *latency* dan *jitter*, protokol ini memiliki keterbatasan dalam menangani pengiriman data berukuran besar (2048 *byte*). Pada skenario tersebut, data tidak dapat diterima secara utuh oleh *server* sehingga proses parsing mengalami kegagalan dan menghasilkan Internal Server Error, yang berdampak pada tingginya nilai *packet loss*. Berdasarkan hasil pengujian, pemilihan protokol komunikasi sebaiknya disesuaikan dengan ukuran *payload* yang ditransmisikan. *CoAP* lebih sesuai digunakan untuk paket data berukuran kurang dari 1 KB, sedangkan *WebSocket* lebih sesuai untuk paket data berukuran lebih dari 1 KB karena berjalan di atas protokol *TCP* yang memiliki mekanisme segmentasi data (*Maximum Segment Size*/*MSS*) sehingga mampu menangani pengiriman data berukuran besar dengan lebih andal.

5.2 Saran

Dalam rangka penyempurnaan dan pengembangan lebih lanjut terhadap penelitian Sistem *Smart Hydroponic* ini, terdapat beberapa saran yang dapat dipertimbangkan untuk penelitian di masa mendatang:

1. Implementasi *Block-Wise Transfer* pada *CoAP*

   Untuk mengatasi kelemahan kegagalan pengiriman *payload* besar, penelitian selanjutnya sangat disarankan untuk menggunakan pustaka *CoAP* pada mikrokontroler yang mendukung mekanisme *Block-Wise Transfer* (RFC 7959), sehingga perangkat dapat mengirimkan paket data kompleks secara bertahap tanpa memicu penolakan dari *server back-end*.
2. Peningkatan Aspek Keamanan Jaringan

   Penelitian ini belum mencakup pengujian pada lapisan keamanan (*security layer*). Pengembangan selanjutnya dapat mengimplementasikan enkripsi *Datagram Transport Layer Security* (*DTLS*) untuk protokol *CoAP* dan *WebSocket Secure* (WSS) untuk protokol *WebSocket*, guna melindungi data operasional hidroponik dari potensi intersepsi di jaringan nirkabel.

3. Pengujian Stres Skala Penuh di Lingkungan Nyata

   Disarankan untuk melakukan pengujian operasional (*stress testing*) secara terus-menerus (24/7) di *Smart Hydroponic* Fakultas Ilmu Komputer UPNVJ secara langsung. Hal ini bertujuan untuk mengobservasi dampak interferensi sinyal fisik, suhu perangkat keras, serta stabilitas koneksi *server* dalam jangka waktu yang lebih panjang.

4. Evaluasi Proses Pembacaan Data Sensor

   Penelitian ini belum mengevaluasi proses pembacaan data sensor terhadap performa komunikasi. Oleh karena itu, penelitian selanjutnya dapat menganalisis waktu eksekusi pembacaan sensor, penggunaan memori, serta mekanisme penjadwalan tugas (*task scheduling*) pada mikrokontroler untuk mengatahui pengaruhnya tehadap *latency , jitter, throughput*, dan *packet loss*.