

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Beberapa penelitian terdahulu yang relevan dengan topik ini dirangkum dalam matriks perbandingan. Lima penelitian dipilih berdasarkan keterkaitan dengan aspek utama sistem yang dikembangkan, yaitu penggunaan protokol ESP-NOW, domain olahraga, arsitektur *multi-node*, sinkronisasi waktu, dan analisis performa empiris. Perbandingan kelima penelitian tersebut disajikan pada Tabel 2.1.

Tabel 2.1. Matriks Perbandingan Penelitian Terdahulu

No	Aspek / Fitur Utama	[4]	[15]	[11]	[6]	[13]	Penelitian Ini
1	Protokol ESP-NOW	×	✓	✓	✓	✓	✓
2	Aplikasi olahraga	✓	×	×	✓	×	✓
3	<i>Multi-node / multi-lane</i>	×	✓	✓	✓	✓	✓
4	Sinkronisasi waktu antar <i>node</i>	×	✓	×	×	✓	✓
5	Analisis <i>Packet Delivery Rate</i> (PDR) empiris	×	×	×	×	✓	✓
6	Analisis <i>latency</i> empiris	×	✓	✓	✓	✓	✓
7	Pengujian di lingkungan RFI (<i>Radio Frequency Interference</i>)	×	×	×	×	✓	✓
8	Tanpa <i>router/access point</i>	✓	✓	✓	✓	✓	✓

Keterangan. Simbol ✓ = ada, × = tidak ada.

Pada [4], sistem *timing* atlet portabel berbasis Arduino dikembangkan menggunakan sensor LDR (*Light Dependent Resistor*) dan modul *transceiver* nirkabel. Sistem ini mendeteksi kehadiran atlet pada garis *start* dan *finish* melalui laser yang diarahkan ke sensor LDR, kemudian menghitung selisih waktu keduanya secara otomatis tanpa memerlukan koneksi internet. Sistem ini hanya mendukung satu jalur kompetisi dan tidak menggunakan protokol ESP-NOW, sehingga skalabilitasnya terbatas. Penelitian ini menjadi bukti kelayakan pendekatan *timing* berbasis nirkabel tanpa infrastruktur jaringan, sekaligus menegaskan kebutuhan

perluasan ke arsitektur *multi-lane* yang diimplementasikan di penelitian ini.

Pada [15], protokol FTSP (*Flooding Time Synchronization Protocol*), yaitu protokol sinkronisasi waktu di mana satu *node* referensi menyiarkan *timestamp* secara periodik untuk dikalibrasi oleh seluruh *node* penerima, diimplementasikan pada *Wireless Sensor Network* (WSN) berbasis ESP-NOW dengan topologi bintang menggunakan lima perangkat NodeMCU ESP8266. Evaluasi menunjukkan reduksi *offset* waktu yang konsisten setelah sinkronisasi, dengan rata-rata *offset* akhir berkisar antara 1,062-1,079 ms. Penelitian ini mencakup arsitektur *multi-node* dan sinkronisasi waktu berbasis ESP-NOW. Penelitian ini menjadi referensi pendekatan sinkronisasi waktu berbasis ESP-NOW yang diadopsi pada sistem *timing* perlombaan renang ini.

Pada [11], sistem akuisisi data sEMG (*surface electromyography*, teknik pengukuran aktivitas listrik otot melalui elektroda yang dipasang pada permukaan kulit) *multi-node* nirkabel berbasis ESP-NOW dikembangkan dengan setiap *node* mengirimkan data ke *hub* pusat. Pengujian menunjukkan bahwa tidak ditemukan *packet loss* yang signifikan untuk satu hingga dua *node* aktif secara simultan, dengan jarak operasional efektif hingga 100 m untuk dua *node* dan 70 m untuk seluruh enam *node*. Performa *packet loss* minimal pada jarak tersebut menjadi referensi kelayakan ESP-NOW untuk sistem *timing* perlombaan renang ini pada jarak pengujian yang lebih pendek.

Pada [6], sistem pemantauan gerakan dayung pada olahraga air terbuka (*stand-up paddleboarding*) diimplementasikan menggunakan *node* berbasis ESP32 dan protokol ESP-NOW dalam arsitektur *master-slave*. Pengujian lapangan di danau terbuka menunjukkan komunikasi nirkabel dengan 7 *node* yang stabil dalam jarak hingga 100 m dengan *packet loss* minimal serta daya tahan baterai lebih dari 3 jam. Penelitian ini menjadi bukti kelayakan ESP-NOW untuk sistem *multi-node* di lingkungan olahraga air terbuka, yang kondisinya memiliki kemiripan dengan kolam renang kompetisi.

Pada [13], metode Sync-ESP-NOW diusulkan sebagai pendekatan sinkronisasi terjadwal berbasis lapisan aplikasi pada jaringan ESP-NOW dengan 8 *node* ESP32-WROOM dalam area 30×30 m. Pengujian di bawah interferensi Wi-Fi eksternal 40 Mbps menunjukkan tingkat penerimaan paket sebesar 99,2-99,6% dengan waktu transmisi per *round* di bawah 4 ms. Penelitian ini menjadi referensi performa ESP-NOW di bawah kondisi interferensi nirkabel bagi sistem *timing* perlombaan renang pada penelitian ini.

Kelima penelitian tersebut masing-masing hanya mencakup sebagian aspek

sistem *timing* perlombaan renang *multi-lane*, sebagaimana terlihat pada matriks perbandingan. Aspek tersebut mencakup penerapan ESP-NOW pada *timing* renang *multi-lane*, sinkronisasi waktu antar *node*, serta pengujian empiris PDR dan *latency* di bawah kondisi RFI yang bervariasi. Sejauh kajian literatur yang ditelusuri, belum ditemukan penelitian yang menggabungkan seluruh aspek tersebut secara bersamaan.

2.2 ESP-NOW

ESP-NOW adalah protokol komunikasi *peer-to-peer* yang dikembangkan oleh Espressif, beroperasi di atas lapisan *Physical Layer* (PHY) Wi-Fi tanpa memerlukan *router* [7]. Setiap perangkat diidentifikasi melalui alamat MAC (*Media Access Control*) miliknya [13]. Beberapa protokol komunikasi nirkabel jarak pendek yang umum digunakan pada sistem IoT (*Internet of Things*) adalah Bluetooth, ZigBee, Wi-Fi, dan ESP-NOW [7, 8]. Perbandingan karakteristik umum keempat protokol tersebut disajikan pada Tabel 2.2.

Tabel 2.2. Perbandingan Protokol Nirkabel Jarak Pendek

Parameter	Bluetooth	ZigBee	Wi-Fi	ESP-NOW
Standar IEEE	802.15.1	802.15.4	802.11	802.11
Frekuensi	2,4 GHz	2,4 GHz	2,4/5 GHz	2,4 GHz
<i>Data rate</i>	~784 Kbps	250 Kbps	54 Mbps	1 Mbps
Jangkauan maksimum	100 m	100 m	100 m	220 m
<i>Payload</i> maksimum	251 B	128 B	2312 B	250 B
Konsumsi daya	1-35 mA	1-10 mA	100-350 mA	60-100 mA

Sumber [7, 8]. Nilai jangkauan merupakan batas maksimum teoritis. Jangkauan efektif Bluetooth dalam kondisi tipikal berkisar 10-20 m dan Wi-Fi 30-100 m [8].

ESP-NOW dipilih untuk sistem ini karena memiliki beberapa keunggulan berikut.

1. Tidak memerlukan *router* atau *access point* sehingga sistem dapat beroperasi sepenuhnya mandiri di lokasi manapun [7].

2. Mendukung mode *broadcast* ke semua *node* sekaligus tanpa proses *pairing* satu-per-satu [9, 13].
3. Jangkauan hingga 220 m di lapangan terbuka, memberikan margin yang cukup untuk berbagai geometri *venue* kompetisi [7].
4. Waktu transmisi rendah yang sesuai dengan kebutuhan *latency real-time* pada sistem *timing* [13].

Wi-Fi mengharuskan setiap perangkat terhubung ke *access point* yang sama, sehingga sistem memerlukan infrastruktur jaringan tambahan yang tidak selalu tersedia di *venue outdoor* perlombaan renang [7]. Bluetooth memiliki jangkauan efektif 10-20 m [8], jauh di bawah jarak antar *node* yang dibutuhkan pada kolam renang kompetisi. ZigBee memiliki jangkauan efektif hingga 75 m [8], namun pada jarak tersebut di lingkungan terbuka dengan potensi interferensi, margin tersebut tidak cukup untuk menjamin koneksi yang stabil. ESP-NOW beroperasi di atas lapisan PHY 802.11 sehingga menggunakan radio Wi-Fi yang sudah terintegrasi di modul ESP32-WROOM-32U tanpa komponen tambahan, dengan kapasitas *payload* maksimum 250 *byte* per paket [7].

Selain karakteristik fisik tersebut, ESP-NOW mendukung dua mode pengiriman dengan tingkat jaminan pengiriman yang berbeda, yaitu sebagai berikut.

1. Mode *broadcast* mengirimkan paket ke semua perangkat terdekat tanpa konfirmasi pengiriman individual, sehingga cocok untuk distribusi perintah ke banyak *node* sekaligus [9, 13].
2. Mode *unicast* mengirimkan paket ke satu perangkat berdasarkan MAC *address* dengan mekanisme konfirmasi pengiriman (*acknowledgment*) otomatis, sehingga pengirim dapat mengetahui keberhasilan atau kegagalan pengiriman [9].

2.3 Radio Frequency Interference (RFI)

ESP-NOW bekerja pada pita ISM (*Industrial, Scientific, and Medical*) 2,4 GHz yang bersifat medium bersama dan rentan terhadap interferensi [16]. Gangguan paling langsung berasal dari jaringan Wi-Fi 802.11b/g/n pada kanal yang sama, karena setiap *access point* aktif menambah peluang tabrakan paket [13]. Pita ini juga dihuni Bluetooth, ZigBee, dan bocoran daya oven *microwave* yang tidak tampak pada pemindaian nama jaringan Wi-Fi [17].

Kualitas sinyal juga dipengaruhi kehadiran penonton. Tubuh manusia menyerap gelombang 2,4 GHz sehingga menaikkan *path loss* [18], sedangkan jumlah jaringan Wi-Fi terdeteksi cenderung ikut naik saat *venue* lebih ramai [19]. Karena banyaknya sumber gangguan tersebut, performa ESP-NOW pada sistem *timing* perlombaan renang perlu dievaluasi di bawah tingkat RFI yang berbeda.

2.4 Reference Broadcast Synchronization (RBS)

Reference Broadcast Synchronization (RBS) adalah metode sinkronisasi waktu terdistribusi yang menggunakan satu *event broadcast* bersama sebagai referensi “jam nol” (T_0) bagi semua penerima, sehingga seluruh *node* tersinkronisasi tanpa perlu *timestamp* eksplisit maupun protokol sinkronisasi terpisah [20, 21]. Salah satu potensi kesalahan pada RBS adalah perbedaan *propagation delay*, yaitu waktu rambat sinyal *broadcast* hingga sampai ke tiap *node*. Sinyal radio merambat mendekati kecepatan cahaya, yaitu sekitar 3×10^8 m/s, sehingga pada jarak terjauh antar *node* sekitar 60 m perbedaan tersebut hanya $60 \text{ m} \div (3 \times 10^8 \text{ m/s}) \approx 0,2 \mu\text{s}$ dan dapat diabaikan.

Karena *propagation delay* tersebut dapat diabaikan, faktor pembatas akurasi sinkronisasi yang tersisa adalah *crystal drift*, yaitu perbedaan laju jam lokal antar *node* akibat toleransi osilator. Setelah T_0 disinkronkan, setiap *node* mengukur waktu secara independen menggunakan osilatornya masing-masing, sehingga jam antar *node* dapat sedikit berbeda seiring waktu berjalan [14]. Nilai *drift* osilator nirkabel umumnya berada pada rentang 10-100 ppm [21]. Kristal 40 MHz pada modul ESP32-WROOM-32U dispesifikasikan dengan toleransi ± 10 ppm [22]. Dengan nilai 10 ppm, deviasi maksimum selama sesi perlombaan 60 detik dihitung pada Persamaan 2.1.

$$\delta_{\max} = 10 \times 10^{-6} \times 60 \text{ s} \times 10^6 \frac{\mu\text{s}}{\text{s}} = 600 \mu\text{s} \quad (2.1)$$

Nilai 600 μs ini di bawah 1 ms dan masih lebih kecil daripada resolusi waktu 0,01 detik (setara 10 ms) yang digunakan pada perlombaan renang resmi [2], sehingga pengaruhnya terhadap hasil waktu dapat diabaikan. RBS sesuai untuk sistem yang sudah memiliki paket *broadcast* dalam mekanisme komunikasinya karena paket tersebut dapat berfungsi ganda sebagai sinyal sinkronisasi T_0 tanpa menambah *overhead* komunikasi. Hal ini berbeda dengan metode seperti *Flooding*

Time Synchronization Protocol (FTSP) yang memerlukan pertukaran *timestamp* berulang [15].

2.5 ESP32-WROOM-32U

ESP32-WROOM-32U adalah modul ESP32 dengan konektor IPEX/U.FL untuk antena eksternal [22]. ESP32 adalah keluarga mikrokontroler berbasis arsitektur Xtensa 32-bit dengan prosesor *dual-core* berfrekuensi hingga 240 MHz dan konektivitas Wi-Fi/Bluetooth terintegrasi [23, 24]. Tampilan modul ditunjukkan pada Gambar 2.1.



Gambar 2.1. Modul ESP32-WROOM-32U

Beberapa fitur ESP32-WROOM-32U menunjang kebutuhan pencatatan waktu presisi dan respons cepat pada sistem *timing* ini, yaitu sebagai berikut.

1. Arsitektur *dual-core* memungkinkan satu *core* menangani operasi *time-critical* seperti manajemen protokol jaringan (Wi-Fi/ESP-NOW *stack*), sementara *core* lainnya menjalankan pemrosesan tingkat aplikasi termasuk tugas pencatatan waktu yang kritis [23].
2. *Hardware interrupt* pada pin *General Purpose Input Output* (GPIO) mendukung deteksi event eksternal dengan *latency* rendah via *Interrupt Service Routine* (ISR) [22].
3. Timer beresolusi 1 μ s memungkinkan pengukuran waktu dengan presisi mikrodetik [24].

2.6 Parameter Pengukuran

Evaluasi performa sistem menggunakan lima parameter, yaitu *Packet Delivery Rate* (PDR), *retry rate*, *latency*, *jitter*, dan deviasi sinkronisasi. PDR dan *retry rate* mengukur keandalan dan kelancaran pengiriman paket. *Latency* mengukur karakteristik waktu transmisi, *jitter* mengukur variasi waktu transmisi antar paket, dan deviasi sinkronisasi mengukur akurasi sinkronisasi antar Lane.

2.6.1 Packet Delivery Rate (PDR) dan Retry Rate

PDR yang dihitung adalah *effective* PDR, yaitu tingkat keberhasilan pengiriman paket setelah mekanisme pengiriman ulang selesai bekerja, yang diformulasikan pada Persamaan 2.2.

$$\text{PDR} = \frac{\sum \text{paket diterima}}{\sum \text{Lane siap}} \times 100\% \quad (2.2)$$

Setiap Lane yang siap dalam suatu perlombaan diharapkan mengirim satu paket *finish*. Lane yang sudah siap tetapi paket *finish*-nya tidak diterima Receiver tetap ikut dihitung sebagai Lane yang seharusnya mengirim. Kegagalan pengiriman tersebut menurunkan PDR.

Untuk mengukur seberapa banyak percobaan pengiriman ulang yang diperlukan akibat kondisi RFI, digunakan *retry rate*, yaitu rata-rata *retry_count* per paket yang diterima. Nilai *retry_count* = 0 berarti paket tiba pada percobaan pertama. Nilai tinggi menunjukkan kanal yang memburuk meskipun paket akhirnya sampai. Dua kondisi dengan PDR identik dapat memiliki *retry rate* yang sangat berbeda, dan perbedaan ini dapat menunjukkan seberapa berat RFI yang terjadi.

2.6.2 Latency dan Jitter

Latency mengukur selisih waktu antara saat Lane *node* mengirim paket *finish* dan saat paket tersebut diterima Receiver. *Latency* juga digunakan sebagai dasar perhitungan deviasi sinkronisasi (ΔT_0) antar Lane. *Jitter* dicatat secara deskriptif tanpa target numerik. Kedua metrik didefinisikan sebagai berikut.

1. *Latency* (L) adalah selisih antara waktu penerimaan paket *finish* di Receiver (t_{rx}) dan waktu pengirimannya di Lane (t_{tx}) dalam satuan μs , dinyatakan pada

Persamaan 2.3.

$$L = t_{rx} - t_{tx} \quad (2.3)$$

Nilai L dapat sedikit negatif apabila perbedaan T_0 antara Lane dan Receiver melebihi waktu transit paket, sehingga nilai absolut $|L|$ digunakan saat mengevaluasi *latency* terhadap target, sedangkan nilai bertandanya tetap dipakai pada perhitungan deviasi sinkronisasi antar Lane (ΔT_0).

2. *Jitter* (J_n) adalah variasi *latency* antar paket berurutan, dinyatakan pada Persamaan 2.4.

$$J_n = |L_n - L_{n-1}| \quad (2.4)$$

2.6.3 Deviasi Sinkronisasi

Deviasi sinkronisasi (ΔT_0) mengukur perbedaan T_0 antar dua Lane dengan membandingkan *latency* paket *finish* keduanya. Apabila T_0 antar Lane berbeda, perbedaan tersebut terbawa ke selisih *latency* paket *finish* yang diterima Receiver. Selisih *latency* tersebut menjadi proksi perbedaan T_0 , dinyatakan pada Persamaan 2.5.

$$\Delta T_{0,A,B} = |L_A - L_B| \quad (2.5)$$

Pengurangan *latency* paket *finish* kedua Lane tersebut mengasumsikan *latency* transit paket *finish* keduanya kira-kira setara, sehingga komponen transit saling menghilangkan dan yang tersisa adalah perbedaan T_0 antar Lane. Perbedaan T_0 ini idealnya hanya mencerminkan perbedaan jam lokal antar Lane apabila kedua Lane menerima paket START dengan waktu transit yang identik. Dalam praktiknya, T_0 di setiap *node* mengandung *latency* nirkabel salinan paket START yang diterimanya, sehingga bila kedua Lane menerima salinan dengan *latency* berbeda, selisih *latency* START itu ikut terhitung dalam ΔT_0 . Nilai yang terukur dengan demikian mencerminkan gabungan perbedaan jam dan perbedaan *latency* START.

2.7 Metode Evaluasi

Evaluasi performa sistem dilakukan dengan membandingkan hasil pengukuran tiap parameter pada setiap skenario terhadap nilai target yang

ditetapkan. Parameter dibagi ke dalam tiga kelompok berdasarkan aspek yang diukur, yaitu keandalan dan kelancaran pengiriman, karakteristik waktu transmisi, dan akurasi sinkronisasi. Setiap kelompok dievaluasi terhadap acuan yang berbeda sesuai dengan karakteristik performa yang relevan.

Latency dievaluasi terhadap target sebagai indikator kualitas transmisi sistem pada setiap skenario pengujian, dengan perbandingan dibatasi pada paket yang terkirim tanpa pengiriman ulang agar yang dinilai adalah waktu transit, bukan waktu tunggu pengiriman ulang. *Jitter* dicatat secara deskriptif tanpa dibandingkan terhadap target. Deviasi sinkronisasi (ΔT_0) dievaluasi terhadap batas toleransi yang diturunkan dari standar presisi sistem *timing* perlombaan renang modern [1, 2], agar perbedaan acuan waktu antar Lane tidak memengaruhi hasil pada resolusi waktu resmi. Evaluasi dilakukan pada skenario dengan kondisi interferensi minimal agar pengaruh RFI terhadap *latency* tidak mendominasi selisih yang diukur. PDR membuktikan keandalan pengiriman akhir di bawah kondisi RFI, sedangkan *retry rate* menunjukkan seberapa berat mekanisme pengiriman ulang bekerja untuk mencapai PDR tersebut.

