Implementasi Scheduling Pengiriman Data Detak Jantung Melalui Sensor Pulse

1st Muhammad Rifaldi Septa Pratama
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
muhammadrifaldisp@student.telkomuniversity.ac.id

2nd Ida Wahidah
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
wahidah@telkomuniversity.ac.id

3rd Vinsensius Sigit W.P
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
vinsensiusvsw@telkomuniversity.ac.id

***Abstrak* — Otak merupakan salah satu organ tubuh manusia yang sangat penting karena merupakan pusat dan tempat keseimbangan manusia. Pengujian aktivitas otak manusia sangat dianjurkan karena dapat meningkatkan kondisi fisik dan mental seseorang, seperti pemantauan kesehatan otak dan mental, identifikasi dini penyakit dan *screening* masalah kognitif serta mental dan emosional. Dengan kemungkinan besar peningkatan jumlah penderita gangguan fungsi otak, dapat disimpulkan bahwa solusi sistematis yang diusulkan untuk masalah ini adalah desain EEG (electroencephalogram). EEG (*electroencephalography*) adalah alat yang mengukur aktivitas listrik otak manusia. Kondisi detak jantung dibutuhkan dalam proses pengujian aktivitas otak ini. Maka dari itu Sensor Pulse merupakan salah satu perangkat yang digunakan untuk mengukur dan mendeteksi detak jantung seseorang. Scheduling adalah proses mengatur urutan eksekusi tugas atau pekerjaan dalam suatu sistem atau lingkungan yang memiliki sumber daya terbatas, seperti waktu, tenaga kerja, mesin, atau ruang. Tujuannya yaitu mengoptimalkan penggunaan sumber daya yang ada, mengurangi waktu tunggu, meningkatkan efisiensi, dan mencapai berbagai tujuan yang relevan sesuai dengan konteksnya. Proses *scheduling* ini digunakan menggunakan metode *priority scheduling*. Di mana pengujian pada scheduling ini membandingkan nilai rata – rata perubahan detak jantung yang didapatkan dari Sensor Pulse. Dengan menggunakan metode *priority scheduling* maka dapat menilai tingkat keakurasian dari sebuah sensor.**

Kata kunci*— *EEG, Sensor Pulse, Scheduling.

I. PENDAHULUAN

Electroencephalography (EEG) adalah metode non-invasif yang telah lama digunakan untuk merekam aktivitas listrik otak manusia. Teknik ini memungkinkan kita untuk memahami bagaimana sinyal listrik yang dihasilkan oleh neuron di otak berhubungan dengan fungsi kognitif, gangguan neurologis, perubahan emosi dan banyak masalah kesehatan fisik, dan gangguan roh lainnya. Seiring kemajuan teknologi, analisis EEG menjadi lebih kompleks dan informatif, memungkinkan kita mengeksplorasi lebih dalam kompleksitas aktivitas otak manusia. Penelitian sebelumnya telah mengidentifikasi pola EEG berbeda yang terkait dengan perubahan kesehatan mental, proses kognitif, dan respons terhadap rangsangan eksternal. Akan tetapi, masih dibutuihkan penelitian lebih lanjut supaya bisa memahami lebih detail bagaimana sebenarnya sinyal EEG tersebut diinterpretasikan dan bagaimana penggunaannya dalam berbagai konteks medis dan non-medis.

II. KAJIAN TEORI

Pada pembahasan ini implementasi scheduling dilakukan terhadap mikrokontroller ESP 32 dan sensor detak jantung yaitu Sensor Pulse. Proses scheduling dilakukan menggunakan Arduino Ide sebagai monitor terhadap data yang akan di-*scheduling*. Lalu pengujian scheduling diukur menggunakan Wireshark.

A. ESP32

ESP32 adalah mikrokontroler yang sangat populer dalam dunia pengembangan perangkat keras (*hardware development*). ESP 32 ini dikembangkan oleh perusahaan Tiongkok Espressif Systems. ESP 32 ini dirancang untuk mengontrol berbagai fungsi dalam suatu sistem.

GAMBAR 1
ESP32

Gambar di atas merupakan contoh ESP32 yang berfungsi sebagai tempat pengolahan data Sensor Pulse dan scheduling data.

B. Sensor Pulse

Sensor Pulse adalah perangkat yang digunakan untuk mengukur denyut nadi seseorang. Sensor ini mengidentifikasi perubahan dalam aliran darah yang disebabkan oleh kontraksi jantung dan mengubahnya menjadi sinyal elektrik yang dapat diukur. Informasi ini kemudian dapat digunakan untuk berbagai aplikasi tergantung pada konteks penggunaannya.

GAMBAR 2
Sensor Pulse

Pada gambar diatas merupakan Sensor Pulse yang digunaka untuk mendeteksi detak jantung pada pengguna.

C. Arduino IDE

Yakni suatu *software* yang digunakan dalam pembuatan *sketch* pemrograman sekaligus berperan sebagai media untuk pemrograman pada *board* yang hendak diprogram. Arduino IDE digunakan dalam kegiatan mengedit, membuat, meng-*upload* ke board yang ditentukan, dan meng-coding program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA yang dilengkapi dengan beberapa *library* sehingga dapat mempermudah dalam penggunaannya. Proses pengimplementasian terhadap scheduling data Sensor Pulse ini dilakukan pada Arduino IDE.

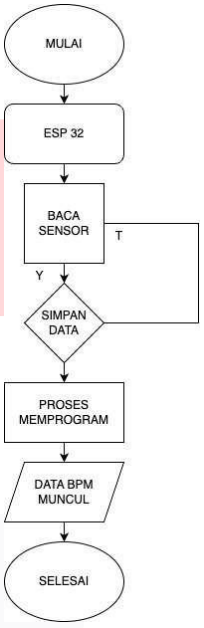
D. Wireshark

Wireshark adalah perangkat lunak pemantauan jaringan (network monitoring) yang digunakan untuk menganalisis, merekam, dan memeriksa lalu lintas jaringan. Kemudian juga dikenal sebagai salah satu alat yang paling populer dan kuat dalam domain analisis jaringan. Wireshark ini digunakan untuk pengujian parameter QoS untuk mengetahui kualitas jaringan yang digunakan pada pengiriman data. Parameter yang diujikan adalah delay, troughput, dan paket loss.

III. METODE

Scheduling adalah proses pengaturan dan pengalokasian sumber daya komputer untuk mengeksekusi tugas atau proses secara efisien dalam suatu sistem komputasi. Tujuannya adalah untuk mengoptimalkan penggunaan sumber daya, meningkatkan kinerja sistem, dan memenuhi berbagai persyaratan waktu yang mungkin ada dalam konteks tertentu. Contoh metode yang dipilih dalam topik ini adalah *priority scheduling*.

A. Flowchart Sensor Pulse



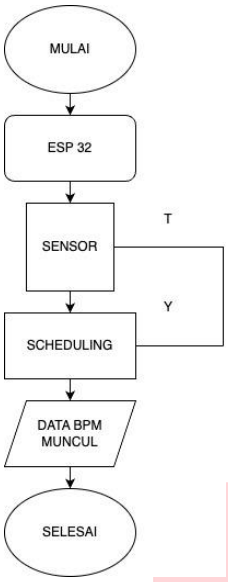
GAMBAR 3
Flowchart Sensor Pulse

Gambar di atas merupakan *flowchart* dari Sensor Pulse. Sensor Pulse dihubungkan ke ESP 32 melalui kabel. Setelah sensor berhasil disambungkan maka ESP 32 akan segera membaca dan mengolah data hasil deteksi dari Sensor Pulse.

B. Metode Scheduling

Metode scheduling yang digunakan pada topik ini adalah *priority scheduling,* yakni suatu algoritma penjadwalan dalam sistem komputer yang digunakan untuk mengatur eksekusi tugas berdasarkan prioritas yang diberikan pada masing-masing tugas. Dalam sistem dengan *priority scheduling,* tugas atau proses dengan tingkat prioritas yang lebih tinggi akan mendapatkan waktu CPU lebih banyak daripada tugas dengan prioritas lebih rendah.

C. Flowchart Scheduling



GAMBAR 4
Flowchart Scheduling

Priority Scheduling merupakan suatu metode yang cukup mudah dalam melakukan proses penjadwalan data. Penjadwalan data dilakukan dengan masing-masing sensor mengirimkan 5 data secara sekaligus lalu di rata – ratakan. Setelah datanya berhasil di rata – ratakan maka sensor dapat menentukan tingkat keakurasian data. Berikut merupakan langkah - langkah yang dilakukan untuk proses scheduling:

1. Langkah pertama adalah ESP 32 dihubungkan ke Sensor Pulse.
2. Setelah masing-masing ESP 32 dihubungkan terhadap Sensor Pulse, sensor tersebut mengeluarkan data.
3. Kemudian setelah mendapatkan BPM masing–masing ESP 32 melakukan perhitungan untuk mengetahui rata – rata perubahan BPM yang didapatkan.
4. Setelah mendapatkan rata – rata perubahan BPM pada masing–masing ESP 32, maka ESP 32 kedua akan mengirimkan hasil rata – rata perubahan BPM yang didapatkan ke ESP 32 ketiga untuk dibandingkan dengan hasil rata – rata pada ESP 32 ketiga.
5. Pada metode penjadwalan ini menggunakan priority scheduling, yang dilakukan dengan mengatur penjadwalan pengiriman data ESP 32 ke MQTT dengan membandingkan hasil rata – rata perubahan pada data BPM.
6. Data yang diprioritaskan berdasarkan hasil rata-rata perubahan yang paling stabil yaitu data yang memiliki nilai Error paling sedikit.

D. Paramater Pengujian QoS

Quality of Service (QoS) merupakan suatu pengukuran dan penilaian seberapa baik suatu jaringan atau layanan jaringan dapat menyediakan pengalaman yang dapat diandalkan dan sesuai dengan kebutuhan pengguna. Untuk mengukur QoS, berbagai parameter dan metrik digunakan. Berikut beberapa parameter pengukuran QoS yang digunakan :

1. Delay

Yakni waktu yang dibutuhkan untuk paket data atau sinyal untuk melakukan perjalanan dari sumber ke tujuan dalam jaringan. Di bawah ini merupakan rumus dalam mencari delay:

$$\text{Rata – rata delay} = \frac{Total\ Delay}{Jumlah\ Paket\ Yang\ diterima} \quad (1)$$

2. Jitter

Jitter adalah variasi dalam waktu respons atau delay yang terjadi antara kedatangan paket data. Jitter yang rendah penting untuk aplikasi *real-time* seperti VoIP dan video streaming. Berikut rumus dalam mencari jitter :

$$\text{Jitter} = \frac{\text{Total Variasi Delay}}{Jumlah\ Paket\ Yang\ diterima - 1} \quad (2)$$

3. Throughput

Yakni *bandwidth* sebenarnya (aktual) yang diukur melalui satuan waktu tertentu yang digunakan dalam kegiatan transfer data dengan ukuran tertentu. Berikut rumus dalam mencari throughput:

$$\text{Throughput} = \frac{Paket\ Yang\ Diterima}{Lama\ Pengamatan} \quad (3)$$

4. Packet Loss

Yakni suatu parameter yang menggambarkan kondisi yang menunjukkan jumlah total paket yang hilang, hal tersebut bisa terjadi karena beberapa kemungkinan antara lain terjadinya *overload* di dalam suatu jaringan, *error* yang terjadi pada media fisik. Berikut adalah rumus dalam mencari *packet loss*:

$$\text{Packet Loss} = \frac{Paket\ Data\ Dikirim - Paket\ Data\ Diterima}{Paket\ Data\ Diterima} \times 100\ \% \quad (4)$$

IV. HASIL DAN PEMBAHASAN

Pengimplementasian scheduling pada Sensor Pulse ini menggunakan metode *Priority Scheduling*. Kemudian untuk mengukur parameter jaringan menggunakan wireshark.

A. Pengujian Sensor Pulse

Dalam mengintegrasikan Pulse Sensor dengan ESP32. Setelah berhasil melakukan penyambungan Pulse Sensor, data yang akan ditampilkan adalah hasil deteksi detak jantung pengguna. Berikut beberapa langkah yang bisa diikuti untuk menghubungkan Pulse Sensor:

GAMBAR 5
Penyambungan Sensor Pulse Ke ESP 32

Sambungkan Pulse Sensor ke ESP32 dengan menggunakan kabel jumper sebagai berikut :
Ground Pulse Sensor ke pin GND pada ESP32. Sambungkan kabel 3,5 Volt pada Pulse Sensor ke pin VCC pada ESP32. Terakhir, hubungkan pin 4 pada Pulse Sensor ke pin VP pada ESP32.

1. Sambungkan ESP32 pertama yang telah dipasangkan dengan Pulse Sensor ke ESP32 kedua. Lakukan langkah-langkah berikut:

Hubungkan pin Rx (16) pada ESP32 pertama ke pin Tx (17) pada ESP32 kedua. Sambungkan pin Tx (17) pada ESP32 pertama ke pin Rx (16) pada ESP32 kedua. Pastikan untuk juga menghubungkan kedua ESP32 tersebut dengan pin Ground yang sesuai.

2. Setelah Pulse Sensor berhasil disambungkan ke ESP 32 maka akan muncul tampilan seperti di bawah ini:

GAMBAR 6
Tampilan Data Sensor Pulse

Gambar di atas merupakan tampilan ketika Pulse Sensor telah berhasil mengeluarkan data yang di mana isi datanya itu adalah hasil dari detak jantung pengguna.

B. Pengujian Scheduling

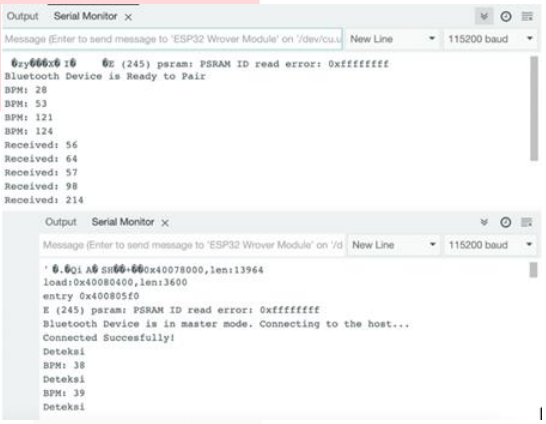
Pada pengujian ini, proses scheduling dilakukan dengan menghubungkan Pulse Sensor ke ESP32 menggunakan metode priority scheduling. Data diambil dari kedua Sensor Pulse sebanyak 5 kali, dan dilakukan perhitungan rata-rata dari data tersebut. Penting untuk dicatat bahwa dalam grafik yang dibuat, sumbu x akan mewakili jumlah delay yang diperoleh, sementara sumbu y akan menggambarkan nilai dari delay yang berhasil diukur.

GAMBAR 7
Perbandingan Grafik ESP 2 dan ESP 3

GAMBAR 8
Perbandingan Grafik ESP 2 dan ESP 3

GAMBAR 9
Perbandingan Grafik ESP 2 dan ESP 3

Gambar 7 sampai dengan gambar 9 menjelaskan mengenai perbandingan scheduling data yang telah dilakukan. Pada gambar 7 merupakan tampilan grafik yang jumlah errornya lebih kecil atau sedikit. Sehingga dapat disimpulkan bahwa priorty scheduling pada gambar 7 menunjukkan hasil yang lebih baik dari pada perbandingan grafik yang lainnya. Pada pengujian ini, langkah-langkahnya mirip dengan yang dilakukan pada bagian penghubungan Sensor Pulse.



GAMBAR 10
Tampilan Data Scheduling

Gambar di atas merupakan tampilan ketika hasil data Sensor Pulse telah berhasil di-*scheduling*.

C. Pengujian Parameter QoS

Quality Of Service merupakan konsep yang digunakan dalam teknologi jaringan dan komunikasi untuk mengukur, mengelola, dan mengoptimalkan kualitas layanan yang diberikan oleh jaringan atau sistem komunikasi. Pengukuran QoS terhitung dari proses ketika pegiriman data dari ESP32 menuju web server. Dengan percobaan setiap 5 kali pengambilan data. Pengukuram Qos dihitung menggunakan aplikasi Wireshark untuk memfilter IP jaringan yang digunakan dalam pengiriman data yaitu 47.57.143.4 dengan tujuan destinasi web server.

PERCOBAAN	DELAY	THROUGHPUT	PACKET LOSS
1	15.11 ms	1.25 kbps	0.15%
2	41.11 ms	1.72 kbps	0,24 %
3	33.55 ms	0.87 kbps	0,2 %
4	55.67 ms	11.36 kbps	0,21 %
5	20.9 ms	0.66 kbps	0,1%
RATA RATA	33,268	3.172	0.18

GAMBAR 10
Tampilan Tabel QoS

Tabel tersebut merupakan tabel hasil dari pengukur Quality Of Service yang dilakukan pada Wireshark.

1. Delay

Delay merujuk pada waktu tunda data dalam jaringan, dengan delay tertinggi mencapai 55,67 ms, terendah 15,11 ms, dan rata-rata 33,26 ms dari 5 percobaan, sesuai dengan standar ITU-T G.1010.

2. Throughput

Throughput mengukur jumlah data yang berhasil dikirim dalam satuan waktu. Throughput tertinggi adalah 11,36 kbps, terendah 0,66 kbps, dan rata-rata 3,172 kbps dari 5 percobaan, memenuhi standar ITU-T G.1010.

3. Packet Loss

Packet Loss adalah kehilangan paket data selama pengiriman dalam jaringan. Packet loss tertinggi adalah 0,24%, terendah 0,1%, dan rata-rata 0,18% dari percobaan berdasarkan standar ITU-T G.1010.

V. KESIMPULAN

Dalam konteks penghubungan sensor pendeteksi detak jantung ke ESP32, pengguna dapat memperoleh data BPM dengan meletakkan Pulse Sensor pada jari tangan. Selain itu, dalam pengaturan penjadwalan pengiriman data, metode priority scheduling digunakan untuk meningkatkan akurasi pengukuran. Setiap Sensor Pulse mengirimkan data sebanyak lima kali sebelum hasil pengukuran dirata-ratakan, memungkinkan pengambilan data yang lebih akurat. Hasil dari pendekatan ini diharapkan mampu memberikan hasil yang lebih optimal untuk analisis detak jantung.

REFERENSI

[1] S. K. Zulfikri Khakim1, "Dasar - Dasar Electroencephalography (EEG) bagi Riset Psikologi The Basics of Electroencephalography (EEG) for Psychological Research," Yogyakarta,Indonesia, 28 June 2021, p. 115.

[2] Priadi, D. (2018). Pengukuran Quality of Service (QoS) Pada Aplikasi File Sharing dengan Metode Client Server Berbasis Android. *Journal of Telecommunication Network (Jurnal Jaringan Telekomunikasi)*, *6*(1), 39-49.