

**Implementasi Sistem *Monitoring Real-time*
Berbasis IoT untuk Analisis *Cycle time* pada *Line*
Produksi *Assembly* Perusahaan Manufaktur**

PROYEK AKHIR

Disusun Oleh:

Elsha Amalia Pusponegoro

3312311042



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
2026**

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, karena atas berkat, rahmat, dan karunia-Nya, penulis dapat menyelesaikan penyusunan Laporan Proyek Akhir ini dengan judul "*Implementasi Sistem Monitoring Real-time Berbasis IoT untuk Analisis Cycle time pada Line Produksi Assembly Perusahaan Manufaktur*".

Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan pada Program Studi Teknik Informatika, Jurusan Teknik Informatika, Politeknik Negeri Batam. Penulis menyadari bahwa selesainya laporan ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Ir. Bambang Hendrawan, ST., MSM., CIPMP., CISCIP. selaku Direktur Politeknik Negeri Batam.
2. Bapak Ahmad Hamim Thohari, S.S.T., M.T. selaku Ketua Jurusan Teknik Informatika.
3. Ibu Yeni Rokhayati, S.Si., M.Sc. selaku Ketua Program Studi Teknik Informatika.
4. Ibu Mir'atul Khusna Mufida, Ph.D. selaku Dosen Pembimbing yang telah meluangkan waktu, tenaga, dan pikiran untuk memberikan arahan serta bimbingan selama pengerjaan proyek akhir ini hingga selesai.
5. Bapak Ahmadi Irmansyah Lubis, S.Kom., M.Kom. dan Ibu Sri Rahayu, M.Pd. selaku Dosen Penguji yang telah memberikan masukan dan saran membangun demi penyempurnaan laporan ini.
6. Pimpinan dan seluruh staf PT Giken Precision Indonesia, khususnya kepada Bapak Agus Irawan dan Bapak Budi Sukmana selaku pembimbing lapangan yang telah memberikan kesempatan, fasilitas, dan ilmu praktis selama penelitian berlangsung.
7. Kedua orang tua dan keluarga tercinta yang senantiasa memberikan doa, semangat, dan dukungan moral maupun materi yang tak terhingga.

8. Teman - teman seperjuangan IF 23, terimakasih banyak sudah mewarnai hari hari penulis dan membuat semua perjalanan kuliah ini menjadi terasa lebih indah dan mudah.
9. *Last but not least*, Terimakasih untuk diri sendiri, Elsha Amalia Pusponegoro. Apresiasi sebesar-besarnya yang telah berjuang untuk menyelesaikan apa yang telah dimulai. Sulit bisa bertahan sampai dititik ini, terimakasih untuk tetap hidup dan merayakan dirimu sendiri, walaupun sering kali putus asa atas apa yang sedang diusahakan. Tetaplah jadi manusia yang mau berusaha dan tidak lelah untuk mencoba. *God thank you for being me independent women, i know there are more great ones but i'm proud of this achievement."*

Penulis menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari semua pihak demi perbaikan di masa mendatang. Akhir kata, semoga laporan Proyek Akhir ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang Teknologi Informasi dan Industri Manufaktur.

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iv
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN	vii
ABSTRAK	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terkait	5
2.2 Landasan Teori	8
2.3 Metode Pengembangan Produk.....	11
BAB III ANALISIS DAN PERANCANGAN	14
3.1 Analisis Kebutuhan	14
3.2 Perancangan.....	18
BAB IV IMPLEMENTASI DAN PENGUJIAN	31
4.1 Hasil Implementasi.....	32
4.2 Pengujian Sistem	37
4.2.1 <i>User Acceptance Testing (UAT)</i>	37
4.2.2 <i>Verifikasi Validitas Data (Functional Verification)</i>	38
4.2.3 <i>Pengujian Efisiensi Kerja (Time Behaviour)</i>	40
BAB V KESIMPULAN.....	42
5.1 Kesimpulan.....	42
5.2 Saran	43
DAFTAR PUSTAKA	44
DAFTAR LAMPIRAN	46

DAFTAR TABEL

Tabel 2. 1 Perbandingan Penelitian Sebelumnya.....	5
Tabel 2. 2 <i>Timeline</i> Pengerjaan.....	13
Tabel 3. 1 Kebutuhan Fungsional	18
Tabel 3. 2 Kebutuhan Non Fungsional	19
Tabel 3. 3 Skenario Usecase <i>Monitoring Cycle time</i>	21
Tabel 3. 4 Skenario Usecase <i>View History Data</i>	22
Tabel 3. 5 Skenario Usecase <i>Export CSV</i>	23
Tabel 4. 1 <i>User Acceptance Testing</i>	37
Tabel 4. 2 Hasil Verifikasi Validitas Data	39
Tabel 4. 3 Hasil Pengujian <i>Time Behaviour</i>	40

DAFTAR GAMBAR

Gambar 2. 1 Metode Prototype	11
Gambar 3. 1 Gambaran Umum Sistem	17
Gambar 3. 2 Usecase Diagram.....	20
Gambar 3. 3 Activity Diagram <i>Monitoring Cycle time</i>	24
Gambar 3. 4 Activity Diagram <i>View Data History</i>	25
Gambar 3. 5 Activity Diagram <i>Export CSV</i>	26
Gambar 3. 6 ER Diagram.....	27
Gambar 3. 7 Perancangan Antarmuka Dashboard <i>Monitoring</i>	28
Gambar 3. 8 Perancangan Antarmuka <i>History Data</i>	29
Gambar 3. 9 Perancangan Logika Program (Firmware)	30
Gambar 3. 10 Perancangan Sistem IoT	31
Gambar 4. 1 Set Up Sensor pada <i>Line</i> Produksi	33
Gambar 4. 2 <i>Dashboard Monitoring</i>	34
Gambar 4. 3 <i>History Data</i>	35

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan <i>Requirement</i>	46
Lampiran B. Link Produk	46
Lampiran C. Dokumen Pengujian UAT.....	46

ABSTRAK

Efisiensi operasional dalam industri manufaktur sangat bergantung pada keakuratan data produksi, khususnya pengukuran *cycle time*. Saat ini, proses pencatatan *cycle time* pada *line* produksi *assembly* di salah satu perusahaan manufaktur elektronik di Batam masih dilakukan secara manual menggunakan stopwatch. Metode ini mengakibatkan tingginya latensi informasi serta rentan terhadap kesalahan pencatatan (*human error*). Proyek akhir ini bertujuan merancang dan mengimplementasikan sistem *monitoring cycle time* secara *real-time* berbasis *Internet of Things* (IoT) untuk mengotomatisasi pengukuran durasi kerja pada setiap stasiun. Metode pengembangan sistem yang digunakan adalah *Prototyping*, yang mengintegrasikan mikrokontroler ESP32 dengan sensor sebagai perangkat akuisisi data, protokol komunikasi MQTT dan WebSocket untuk transmisi data berkecepatan tinggi, serta antarmuka dashboard berbasis web menggunakan React.js dan Node.js. Hasil verifikasi fungsional menunjukkan bahwa sistem mampu mendeteksi data *cycle time* secara otomatis dengan rata-rata deviasi pengukuran sebesar 0,98% dibandingkan metode manual, serta memiliki performa responsif dengan latensi rata-rata 0,035 detik (35 ms). Selain itu, sistem ini dilengkapi fitur penyimpanan data historis dan ekspor laporan dalam format CSV. Penerapan sistem ini memberikan kontribusi signifikan dalam meningkatkan transparansi data produksi, memudahkan supervisor mengidentifikasi hambatan (*bottleneck*) secara dini, serta mendukung pengambilan keputusan manajerial yang lebih cepat.

Kata Kunci: *Cycle time*, *Internet of Things* (IoT), *Real-time Monitoring*, MQTT, ESP32

BAB I PENDAHULUAN

1.1 Latar Belakang

Di era Revolusi Industri 4.0, persaingan di sektor manufaktur menuntut perusahaan untuk terus meningkatkan efisiensi dan produktivitas (Zulfadlillah et al., 2024). Pengambilan keputusan yang cepat dan berbasis data menjadi kunci utama untuk menjaga daya saing. Pada Perusahaan Manufaktur modern, khususnya yang bergerak di bidang perakitan (*assembly*), performa *line* produksi menjadi faktor yang sangat krusial.

Proses perakitan di banyak industri hingga saat ini masih sangat bergantung pada tenaga kerja manusia. Ketergantungan ini menimbulkan tantangan signifikan dalam memantau produktivitas setiap individu secara akurat dan berkelanjutan. Salah satu indikator terpenting untuk menjaga produktivitas agar tetap konsisten dan optimal adalah *cycle time*, yaitu total durasi nyata yang dibutuhkan untuk menyelesaikan satu siklus proses dari awal hingga akhir. Namun demikian, metode pemantauan konvensional memiliki keterbatasan signifikan dalam aspek *data traceability* dan *latency*. Keterlambatan dalam akuisisi data produksi mengakibatkan *gap* informasi antara kejadian aktual di rantai produksi dengan proses pengambilan keputusan manajerial (Zulfadlillah et al., 2024). Hal ini menghambat identifikasi anomali *cycle time* secara *real-time*, yang berdampak langsung pada penurunan efisiensi *line balancing* (Achmadi et al., 2021).

Dari perspektif teknologi informasi, tantangan utama dalam digitalisasi proses ini terletak pada bagaimana mengakuisisi data dari *edge device* (sensor di stasiun kerja) dan mengirimkannya ke antarmuka pengguna (*client*) dengan latensi seminimal mungkin. Pendekatan konvensional yang berbasis *polling* data seringkali membebani server dan menciptakan jeda waktu (*delay*) visualisasi yang tidak dapat ditoleransi dalam pemantauan produksi cepat. Oleh karena itu, diperlukan penerapan arsitektur sistem berbasis *Event-Driven* menggunakan protokol

komunikasi *asynchronous* yang bertujuan untuk menjamin integritas dan *real-time delivery* data dari lapisan fisik ke lapisan aplikasi.

Menyadari kondisi tersebut, implementasi teknologi *Internet of Things* (IoT) menawarkan solusi yang efektif (Chulakit et al., 2023). Arsitektur IoT modern yang memanfaatkan protokol seperti MQTT dan WebSocket mampu menyajikan data secara instan ke *dashboard* berbasis web, yang diharapkan dapat menjadi bahan evaluasi akurat untuk membuat keseluruhan proses lebih efisien (Putra et al., 2023).

Oleh karena itu, penulis mengusulkan sebuah proyek Tugas Akhir berupa “Implementasi Sistem *Monitoring Real-time* Berbasis IoT untuk Analisis *Cycle time* pada *Line Produksi Assembly*”. Proyek ini bertujuan merancang dan menerapkan sebuah sistem yang mampu menyediakan data *cycle time* secara langsung, akurat, dan *live*, sehingga manajemen dapat memantau performa produksi secara lebih efektif dan efisien.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, berikut ini merupakan rumusan masalah dalam penelitian :

1. Belum tersedianya sistem *monitoring* berbasis *Internet of Things* (IoT) yang mampu mengukur *cycle time* pada setiap stasiun kerja di *line produksi assembly* secara otomatis dan *real-time*.
2. Pengukuran *cycle time* di setiap stasiun kerja pada *line produksi* masih dilakukan secara manual menggunakan *stopwatch*. Sehingga memerlukan waktu, rentan terhadap kesalahan pencatatan, serta menyulitkan manajemen untuk mendapatkan data secara cepat dan akurat.

1.3 Tujuan

Sejalan dengan rumusan masalah di atas, tujuan yang ingin dicapai dari proyek akhir ini adalah:

1. Merancang dan membangun sistem *monitoring* berbasis *Internet of Things* (IoT) yang mampu mengukur *cycle time* pada setiap stasiun kerja di *line* produksi *assembly* secara otomatis dan *real-time* dengan tingkat akurasi pengukuran di atas 95% (deviasi < 5% dibandingkan pengukuran manual).
2. Menyajikan data *cycle time* melalui *dashboard* berbasis web dengan latensi pembaruan data di bawah 3 detik, sehingga informasi tersedia secara instan bagi manajemen untuk analisis performa secara efektif

1.4 Batasan Masalah

Agar penelitian ini dapat lebih terarah dan fokus pada tujuan utama, maka ruang lingkup proyek ini dibatasi pada beberapa hal berikut:

1. Sistem yang dikembangkan berfokus pada pemantauan (*monitoring*) *cycle time* dan tidak mencakup sistem kendali (*controlling*) untuk mesin atau *line* produksi.
2. Implementasi perangkat keras dilakukan secara terbatas pada satu stasiun kerja (*workstation*) di *line* produksi *assembly* Perusahaan Manufaktur, sebagai model validasi sistem sebelum penerapan skala penuh
3. Sensor yang digunakan terbatas pada pendeteksian awal dan akhir sebuah siklus kerja untuk menghitung durasi, tidak termasuk sensor untuk pengukuran parameter lain seperti suhu, getaran, atau kualitas produk.
4. Aspek keamanan jaringan dan data tidak menjadi fokus utama dalam penelitian ini.
5. *Output* sistem adalah visualisasi data *cycle time* dan total *throughput*. *Throughput* dihitung saat produk melewati satu titik akhir yang telah ditentukan, tanpa adanya identifikasi produk individual.
6. Aplikasi ini hanya bisa digunakan pada *line* produksi Perusahaan Manufaktur Batam

1.5 Manfaat

Proyek akhir ini diharapkan dapat memberikan manfaat dari sisi praktis maupun teoritis, di antaranya adalah:

1.5.1. Manfaat Praktis

1. Menyediakan data performa produksi yang akurat dan *real-time*, memungkinkan pengambilan keputusan yang lebih cepat dan tepat untuk mengatasi *bottleneck* dan meningkatkan efisiensi *line* produksi.
2. Memudahkan pemantauan kinerja setiap stasiun kerja secara langsung melalui *dashboard* tanpa harus melakukan pencatatan manual, sehingga lebih efisien.
3. Mengubah proses pencatatan produktivitas dari manual menjadi digital, mengurangi potensi *human error* dan menghemat waktu.

1.5.2. Manfaat Teoritis/Ilmiah

1. Memberikan kontribusi keilmuan mengenai implementasi arsitektur *Full-Stack IoT* yang mengintegrasikan mikrokontroler ESP32 dengan antarmuka web modern (React.js) melalui protokol MQTT dan WebSocket untuk menangani kendala *soft real-time* di lingkungan industri.
2. Sebagai referensi akademis dalam penerapan algoritma penghitungan *cycle time* otomatis yang menjembatani data sinyal digital diskrit dari sensor menjadi informasi manajerial yang terstruktur dalam basis data.

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Berikut adalah beberapa kajian atau referensi tinjauan pustaka yang dilihat pada tabel 2.1.

Tabel 2. 1 Perbandingan Penelitian Sebelumnya

Peneliti/Tahun	Judul/Topik Penelitian	Metode/Teknologi	Kelebihan	Kekurangan/Celah
Saranjuu Chulakit, Amirul Syafiq Sadun, Nor Anija Jalaludin, Hairulazwan Hashim, Suziana Ahmad, Nur Aminah Sabarudin, Muhammad Ashraf Fauzi, ZhiWen Wang (2024)	<i>New Approach of Process Cycle time Measurement System with Sensor Application</i>	Sensor IR di tiap stasiun perakitan, ESP32 NodeMCU, Firebase, LabVIEW GUI, conveyor 3D simulasi 3 skenario (balanced, unbalanced, downtime)	<i>Real-time</i> capture waktu siklus, manual & otomatis, identifikasi bottleneck, ekspor Excel, prinsip lean & <i>line</i> balancing	Belum ada notifikasi otomatis, <i>line</i> balancing graph belum <i>real-time</i> , belum gabung data pekerja, masih simulasi skala kecil.
Zulfadlillah, Moh Ayip Fathani, Muhamad Noval, Irwana (2024)	Penerapan Sistem Manufacturing 4.0 dengan <i>Integrasi Internet of Things (IoT)</i> untuk Optimalisasi Efisiensi Produksi	Literature review 45 jurnal Scopus + eksperimen prototipe smart factory (OPC UA, NodeMCU, predictive maintenance, digital twin)	<i>Monitoring real-time</i> (downtime turun 30%), predictive maintenance (biaya perawatan turun 20%), supply chain (hemat energi 15%), studi kasus Siemens & Unilever (akurat 25%, error 0,001%)	Biaya infrastruktur tinggi (\$2,5 juta/pabrik), risiko keamanan siber, tantangan SDM & standar interoperabilitas untuk UKM.
Parhan Oktaria Putra, Herri Setiawan, Zaid	Implementasi Alat <i>Monitoring</i> Suhu Ruangan	Sensor DHT-11, NodeMCU ESP8266, MQTT ke server Adafruit,	<i>Monitoring</i> suhu/kelembaban <i>real-time</i> tanpa hadir di lokasi;	Satu sensor DHT-11 (akurasi terbatas), skalabilitas multi-ruangan belum diuji;

Romegar Mair (2023)	Berbasis <i>Internet of Things</i> (IoT) Menggunakan Metode MQTT dan HTML Pada Ruang Server Universitas Indo Global Mandiri Palembang	web HTML <i>real-time</i> , notifikasi Telegram Bot	notifikasi otomatis $\geq 30^{\circ}\text{C}$, selisih sensor $0,8^{\circ}\text{C}$, delay 5,6 detik; bisa kontrol AC	belum ada keamanan data MQTT, web sederhana, belum analisis tren
Fandi Achmadi, Budi Harsanto, Akhmad Yunani (2021)	Analisis <i>Cycle time</i> Proses Perakitan Senjata di PT Pindad (Persero)	Perhitungan <i>cycle time</i> 56 tugas, 43 stasiun kerja, 155 komponen, analisis dokumen & literatur; usulan integrasi VSM & RPW	Identifikasi bottleneck (253 detik vs takt time 87 detik), basis data actual, rekomendasi metode spesifik sesuai <i>straight-line assembly</i>	Belum implementasi nyata metode VSM+RPW, fokus satu produk, faktor non-teknis belum dibahas mendalam, belum simulasi dampak penerapan
Saranjuu Chulakit, Amirul Syafiq Sadun, Nor Anija Jalaludin, Jamaludin Jalani, Suziana Ahmad, Muhammad Haziq Md Hanapi, Nur Aminah Sabarudin (2023)	<i>A Centralized IoT-Based Process Cycle time Monitoring System for Line Balancing Study</i>	CTMS: IR sensor input/output tiap proses, Arduino + LabVIEW GUI + Google Firebase REST API, Android/mobile app <i>real-time</i> , uji 4 proses, bandingkan dengan stopwatch + korelasi Pearson	<i>Monitoring cycle time real-time & terpusat</i> , data cloud akses PC/smartphones, korelasi tinggi dengan stopwatch (r 0,9465–0,9997), GUI user-friendly, indikator status produk	Hanya <i>monitoring</i> , belum intervensi otomatis, masih simulasi <i>line</i> perakitan kecil yang belum diuji di industri nyata, variabilitas pada proses tertentu, perlu pengembangan kontrol proses

Berdasarkan kajian pustaka yang dirangkum dalam Tabel 2.1, terdapat beberapa kesenjangan (*gap*) teknis dan implementasi yang belum sepenuhnya tertangani oleh penelitian-penelitian terdahulu. Penelitian yang dilakukan oleh Chulakit et al. (2024) dan Chulakit et al. (2023) meskipun telah menerapkan konsep IoT, sebagian besar masih diuji pada lingkungan simulasi (bukan *line* produksi

aktif) dan menggunakan protokol berbasis HTTP/REST API melalui Firebase yang memiliki latensi lebih tinggi dibandingkan protokol komunikasi *real-time* murni. Sementara itu, penelitian Putra et al. (2023) dan Zulfadlillah et al. (2024) lebih berfokus pada pemantauan parameter lingkungan (suhu) dan konsep makro *smart factory*, namun belum secara spesifik menangani pengukuran *cycle time* per proses kerja yang menuntut presisi waktu tinggi.

Penelitian ini hadir untuk mengisi kesenjangan tersebut dengan menerapkan arsitektur *full-stack* IoT yang mengintegrasikan protokol MQTT untuk transmisi data sensor dan WebSocket untuk visualisasi antarmuka. Kebaharuan (*novelty*) utama dalam penelitian ini terletak pada implementasi langsung di *line* produksi pada Perusahaan Manufaktur dengan mekanisme komunikasi data *asynchronous* yang menjamin latensi visualisasi di bawah 3 detik. Pendekatan ini mempermudah supervisor untuk mengidentifikasi *bottleneck* secara visual saat terjadi penyimpangan waktu terhadap *Standard Time*, sebuah kemampuan respons cepat yang sulit dicapai menggunakan metode pengukuran manual seperti yang dibahas oleh Achmadi et al. (2021).

Secara spesifik, kontribusi kebaruan yang ditawarkan dalam penelitian ini dibandingkan penelitian sebelumnya meliputi:

1. Validasi Lingkungan Industri Nyata: Berbeda dengan penelitian sebelumnya yang mayoritas berbasis simulasi, sistem ini diimplementasikan dan diuji langsung pada *line* produksi aktif di Perusahaan Manufaktur.
2. Arsitektur *End-to-End Low Latency*: Penerapan integrasi penuh ESP32 → MQTT → Node.js → WebSocket → React Dashboard untuk mencapai latensi pengiriman data < 3 detik, yang belum menjadi fokus utama pada penelitian terdahulu.
3. Fitur Analisis Historis & Ekspor Data: Penyediaan fitur rekam jejak (*history*) lengkap dan ekspor CSV untuk mendukung analisis jangka panjang manajemen, fitur yang seringkali absen pada purwarupa sistem *monitoring* sederhana

2.2 Landasan Teori

2.2.1 *Internet of Things*

Internet of Things (IoT) adalah konsep di mana perangkat fisik dilengkapi sensor, aktuator, dan kemampuan komunikasi untuk saling terhubung melalui jaringan internet. Integrasi IoT dalam sistem manufaktur memungkinkan pengumpulan, pengiriman, dan analisis data secara *real-time* sehingga mendukung optimalisasi efisiensi produksi dan pengambilan keputusan yang lebih cepat (Zulfadlillah et al., 2024). Dalam penelitian ini, IoT digunakan untuk menghubungkan sensor pada stasiun kerja dengan server agar data *cycle time* dapat dipantau secara langsung.

2.2.2 *Cycle time*

Cycle time adalah waktu yang dibutuhkan untuk menyelesaikan satu siklus proses produksi dari awal hingga akhir pada suatu stasiun kerja atau *line* produksi. Pengukuran *cycle time* yang akurat menjadi indikator utama untuk menjaga stabilitas *line balancing* dan efisiensi produksi (Chulakit et al., 2024). Dengan memantau *cycle time* secara *real-time*, perusahaan dapat mengidentifikasi *bottleneck* (kemacetan proses) yang menghambat target produksi serta melakukan perbaikan berkelanjutan berdasarkan data aktual (Achmadi et al., 2021).

2.2.3 *Real Time*

Dalam konteks *Industrial Internet of Things* (IIoT), sistem *real-time* didefinisikan sebagai kemampuan sistem untuk mengakuisisi dan memvisualisasikan data dengan latensi minimal, sehingga memungkinkan intervensi segera saat terjadi anomali produksi (Chulakit et al., 2024). Sistem yang dikembangkan ini masuk dalam kategori *soft real-time*, di mana toleransi keterlambatan data di bawah 3 detik masih dapat diterima untuk kebutuhan *monitoring* manusia, namun tetap menjaga integritas urutan data (Putra et al., 2023).

2.2.4 **Monitoring Produksi**

Monitoring produksi adalah proses pengawasan berkelanjutan terhadap parameter operasional mesin dan tenaga kerja. Berbeda dengan pengawasan manual, *monitoring* berbasis teknologi terkini memungkinkan akuisisi data performa mesin dan *throughput* secara otomatis dan transparan (Hartanto et al., 2024). Sistem ini memungkinkan supervisor mendeteksi ketidaksesuaian standar produksi secara dini, sehingga strategi perbaikan dapat dieksekusi sebelum target harian terganggu. Pada penelitian ini, sistem dirancang untuk menyediakan data durasi proses dan *history* produksi yang dapat diekspor untuk kebutuhan audit kualitas.

2.2.5 **Line Balancing**

Line balancing adalah proses menyeimbangkan waktu kerja pada setiap proses di *line* produksi agar aliran produksi menjadi lancar dan tidak ada penumpukan pada stasiun tertentu. Ketersediaan data *cycle time* yang presisi sangat krusial dalam menentukan langkah penyeimbangan lintasan (*line balancing*) guna meningkatkan total *throughput* produksi dan meminimalkan waktu menganggur (*idle time*) (Achmadi et al., 2021)

2.2.6 **MQTT**

MQTT (*Message Queuing Telemetry Transport*) adalah protokol komunikasi ringan berbasis *publish/subscribe* yang menjadi standar dalam pengiriman data IoT. Protokol ini terbukti efisien dalam menjaga keandalan data pada jaringan dengan *bandwidth* terbatas, serta memiliki latensi pengiriman yang lebih rendah dibandingkan protokol HTTP konvensional (Pratama et al., 2023).

2.2.7 **WebSocket**

WebSocket adalah protokol komunikasi *full-duplex* yang memungkinkan interaksi dua arah secara terus-menerus antara server dan klien melalui satu koneksi TCP. Dalam penelitian ini, WebSocket digunakan untuk mengirim data *real-time* dari server ke *dashboard* web, memastikan pengguna mendapatkan pembaruan

visualisasi suhu atau data siklus secara instan tanpa perlu memuat ulang halaman, sebagaimana diterapkan dalam sistem *monitoring* modern (Putra et al., 2023)

2.2.8 Dashboard Web

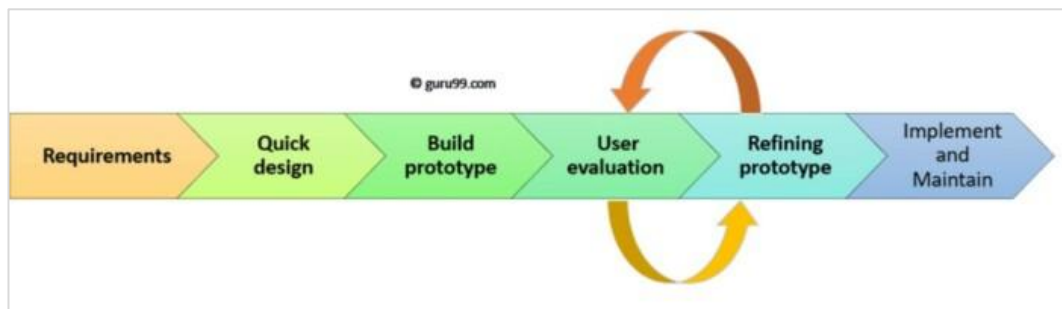
Dashboard web merupakan antarmuka visual untuk menampilkan data dalam bentuk tabel, grafik, atau indikator status. Penggunaan dashboard berbasis web memungkinkan akses data *monitoring* dilakukan dari berbagai perangkat tanpa instalasi aplikasi khusus, memudahkan supervisor dalam memantau parameter produksi secara fleksibel dan responsif (Putra et al., 2023)..

2.2.9 Studi Kasus

Proyek ini diimplementasikan pada *line* produksi di Perusahaan Manufaktur. Selama ini pencatatan *cycle time* dilakukan secara manual sehingga berpotensi menimbulkan keterlambatan dan kesalahan data. Dengan adanya sistem *monitoring* berbasis IoT ini, data *cycle time* tiap stasiun dapat tercatat secara otomatis, *real-time*, dan dapat diakses melalui *dashboard* web untuk evaluasi produktivitas.

2.3 Metode Pengembangan Produk

Penelitian ini menggunakan metode *Prototyping* sebagai pendekatan pengembangan perangkat lunak. Metode ini berfokus pada pembuatan model awal (*prototype*) yang dapat dievaluasi oleh pengguna untuk memperoleh masukan sebelum sistem akhir dibangun. Alur tahapan metode *Prototyping* yang digunakan dalam penelitian ini dapat dilihat pada gambar dibawah ini.



Gambar 2. 1 Metode Prototype

Menurut Ridla dan Rahman (2024), metode *prototype* merupakan teknik pengembangan sistem yang menggunakan model awal (*prototype*) sebagai gambaran bagi pengguna mengenai sistem yang akan dibangun, sehingga evaluasi dapat dilakukan lebih awal untuk memastikan sistem akhir sesuai dengan kebutuhan.

2.3.1 Pengumpulan Kebutuhan (*Requirement*)

Tahap ini merupakan langkah awal untuk mengidentifikasi kebutuhan pengguna dan mendefinisikan fungsi utama sistem. Menurut Ridla dan Rahman (2024), pada tahap ini peneliti harus memahami permasalahan yang ada sebelum melangkah ke tahap desain. Peneliti melakukan observasi langsung pada proses produksi di *line* produksi Perusahaan Manufaktur serta wawancara dengan supervisor. Hal ini bertujuan untuk mendapatkan parameter apa saja yang krusial dalam *monitoring cycle time*.

2.3.2 Perancangan Cepat (*Quick Design*)

Tahap ini bertujuan untuk menerjemahkan kebutuhan menjadi rancangan konseptual sederhana. Aprilisa dan Aulia (2024) menjelaskan bahwa *quick design* berfokus pada penyajian gambaran umum sistem kepada pengguna, seperti alur data dan antarmuka, bukan detail teknis yang mendalam. Peneliti membuat rancangan arsitektur IoT, skema topologi jaringan MQTT–WebSocket, dan *mockup* antarmuka (*dashboard*) agar pengguna mendapat gambaran visual bagaimana data akan ditampilkan.

2.3.3 Pembuatan Prototype

Pada tahap ini, rancangan cepat diterjemahkan menjadi perangkat lunak atau perangkat keras yang dapat beroperasi. Tahap ini adalah proses pembangunan sistem versi awal yang siap untuk diuji coba (Aprilisa & Aulia, 2024). Peneliti membangun sistem *monitoring cycle time* menggunakan mikrokontroler ESP32 dengan sensor proximity/IR, serta mengembangkan backend menggunakan Express.js dan *dashboard* berbasis React untuk menampilkan data secara *real-time*.

2.3.4 Evaluasi Prototype (*User Evaluation*)

Tahap ini melibatkan pengguna secara langsung untuk menguji *prototype* yang telah dibuat. Menurut Ridla dan Rahman (2024), evaluasi ini bertujuan untuk mendeteksi kesalahan fungsi dan mendapatkan umpan balik (*feedback*) guna penyempurnaan sistem. Pengujian dilakukan langsung di rantai produksi. Operator dan supervisor diminta menggunakan sistem untuk menilai apakah tampilan *dashboard* mudah dipahami, apakah data *cycle time* akurat, dan apakah *delay* pengiriman data masih dalam batas toleransi.

2.3.5 Perbaikan dan Implementasi Sistem (*Refining & Implementation*)

Berdasarkan hasil evaluasi, pengembang melakukan perbaikan pada *prototype*. Proses ini dilakukan berulang hingga sistem memenuhi spesifikasi yang diharapkan sebelum diimplementasikan secara penuh (Aprilisa & Aulia, 2024). Masukan dari pengguna digunakan untuk memperbaiki sistem, seperti penambahan

fitur ekspor data ke CSV, fitur riwayat (*history*) produksi, dan optimasi koneksi MQTT agar lebih stabil saat digunakan dalam jangka waktu lama.

Berikut adalah jadwal pelaksanaan pengembangan sistem *monitoring cycle time real-time* berbasis IoT, dimulai dari analisis kebutuhan hingga sistem siap digunakan.

Tabel 2. 2 Timeline Pengerjaan

Tahapan Kegiatan	Juni 2025	July 2025	Agustus 2025	Sept 2025	Okt 2025	Nov 2025
Requirements						
Observasi & Wawancara						
Quick Design						
Design Arsitektur						
Design Mockup						
Build Prototype						
Coding ESP32						
Coding Frontend						
Coding Backend						
User Evaluation						
Uji coba						
Feedback						
Refining Prototype						
Revisi Bug						
Penambahan Fitur						
Implement & Maintenance						
Finalisasi Sistem						

BAB III ANALISIS DAN PERANCANGAN

Analisis dan perancangan pada bagian ini difokuskan untuk menghasilkan rancangan sistem *monitoring cycle time real-time* berbasis IoT yang sesuai dengan ruang lingkup penelitian. Perancangan dilakukan agar sistem yang dibangun benar-benar dapat digunakan sebagai solusi dalam pemantauan *cycle time* di setiap stasiun kerja *line* produksi. Setiap tahapan analisis diarahkan untuk mengidentifikasi kebutuhan fungsional maupun non-fungsional, sehingga sistem yang dihasilkan tidak hanya memenuhi aspek teknis, tetapi juga mampu meningkatkan efektivitas, efisiensi, dan akurasi dalam mendukung kegiatan operasional produksi. Selain itu, perancangan ini juga mempertimbangkan aspek keamanan data, kemudahan penggunaan, serta integrasi dengan teknologi pendukung (MQTT, WebSocket, React), sehingga sistem dapat berfungsi optimal dan berkelanjutan.

3.1 Analisis Kebutuhan

Proses pencatatan *cycle time* di *line* produksi tempat penelitian masih berjalan secara manual dan menghadapi sejumlah kendala. Saat ini pencatatan waktu mulai dan selesai proses dilakukan oleh engineering atau supervisor menggunakan *stopwatch* untuk mengukur durasi proses pada setiap stasiun kerja. Hasil pengukuran tersebut kemudian dicatat pada kertas atau spreadsheet dan direkap untuk dibandingkan dengan *Standard Time* (ST) yang telah ditetapkan.

Mekanisme ini memakan waktu lama dan rentan kesalahan pencatatan, sehingga evaluasi produksi tidak dapat dilakukan secara *real-time*. Supervisor maupun manajer juga kesulitan mengidentifikasi *bottleneck* atau anomali secara cepat karena data tidak langsung tersedia. Akibatnya, ketika terjadi keterlambatan pada suatu proses (misalnya proses A jauh lebih lama daripada standar), penyebab pastinya apakah karena operator kurang efisien, ada masalah mesin, atau faktor lain tidak segera diketahui sehingga penanganan masalah menjadi lambat.

Sistem yang akan dikembangkan bertujuan untuk mengotomatisasi pencatatan *cycle time* secara *real-time* sehingga supervisor dan manajer dapat

langsung melihat durasi tiap proses melalui dashboard. Dengan data *real-time* tersebut, mereka dapat lebih cepat mengidentifikasi proses yang mengalami keterlambatan, menganalisis penyebabnya (misalnya operator lambat atau mesin error), dan mengambil tindakan korektif segera untuk meminimalkan *downtime* dan meningkatkan produktivitas *line* produksi.

3.1.1 Proses Bisnis Berjalan

Sebelum adanya sistem *monitoring cycle time* berbasis IoT, proses pencatatan durasi kerja pada setiap stasiun produksi masih dilakukan secara manual. Operator hanya fokus melaksanakan pekerjaan sesuai instruksi di masing-masing stasiun kerja, sedangkan engineering atau supervisor bertugas mengamati jalannya proses dan mengukur waktu mulai serta selesai setiap siklus menggunakan *stopwatch*. Hasil pengukuran tersebut kemudian dicatat secara manual di kertas atau spreadsheet dan direkap secara berkala untuk dibandingkan dengan *Standard Time* (ST) yang telah ditetapkan perusahaan. Setelah semua data terkumpul, supervisor menyusunnya menjadi laporan evaluasi produksi.

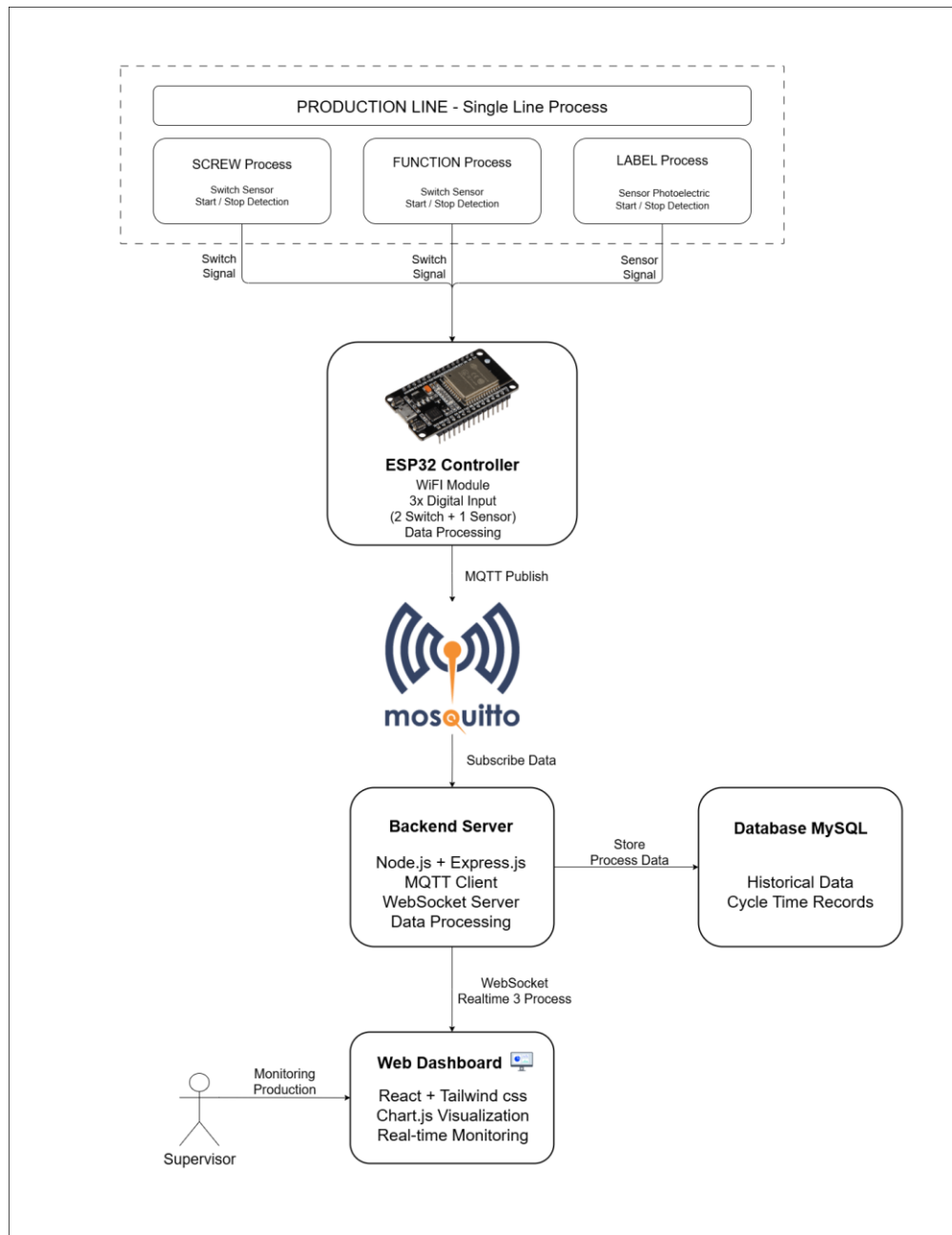
Proses manual ini menimbulkan sejumlah kendala, antara lain: rawan kesalahan pengukuran atau pencatatan, memerlukan waktu lama untuk rekapitulasi, serta sulitnya mengidentifikasi *bottleneck* secara cepat akibat tingginya latensi informasi, karena data tidak terpusat dan tidak tersedia secara *real-time*. Akibatnya, ketika terjadi keterlambatan pada suatu proses, supervisor atau manajer tidak dapat segera mengetahui penyebabnya—apakah karena operator, kondisi mesin, atau faktor lain—sehingga penanganan masalah menjadi lambat. Selain itu, data historis tidak tersimpan secara sistematis sehingga analisis tren jangka panjang menjadi sulit dilakukan.

3.1.2 Gambaran Umum Sistem

Sistem *monitoring cycle time real-time* yang dikembangkan melibatkan beberapa komponen utama, yaitu perangkat IoT (ESP32 beserta sensor) di tiap stasiun kerja, server backend berbasis Node.js/Express, dashboard web sebagai antarmuka pengguna, broker MQTT, serta *database* MySQL untuk penyimpanan data.

- Perangkat IoT (ESP32 + Sensor) berperan sebagai pengirim data. Sensor mendeteksi kejadian mulai/selesai proses di stasiun kerja. Data durasi kemudian dipublish oleh ESP32 ke broker MQTT.
- Broker MQTT menerima data yang dipublish dari semua ESP32. Server backend Node.js/Express berlangganan (*subscribe*) ke broker MQTT untuk mengambil data yang masuk secara *real-time*.
- Server backend memproses data yang diterima, menyimpannya ke *database* MySQL, dan sekaligus meneruskan data terbaru ke dashboard web melalui koneksi WebSocket.
- *Dashboard* Web berperan sebagai antarmuka pengguna (*client*). Melalui dashboard ini pengguna dapat melihat data *cycle time real-time* per stasiun, melihat data historis, dan mengekspor data ke CSV.

Dengan mekanisme ini, alur data berjalan dari sensor → ESP32 (*publish* MQTT) → broker MQTT → server backend (*subscribe* MQTT) → simpan ke MySQL & kirim ke *dashboard* via WebSocket → supervisor melihat data *real-time*. Sistem ini mempermudah supervisor/manajer memantau *cycle time* setiap stasiun kerja dan mengambil tindakan bila ditemukan proses yang lebih lama dari standar.



Gambar 3. 1 Gambaran Umum Sistem

Gambar 3.1 merupakan gambaran umum sistem *monitoring cycle time* berbasis IoT yang dibangun. Sistem ini terdiri dari beberapa komponen utama yaitu ESP32 Controller sebagai pengolah sinyal dari sensor, Mosquitto MQTT Broker sebagai perantara pengiriman data, Backend Server untuk pemrosesan data, Database MySQL untuk penyimpanan data *historis* dan catatan *cycle time*, serta Web Dashboard untuk visualisasi dan *monitoring real-time* oleh supervisor.

3.2 Perancangan

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan sekumpulan layanan atau fungsi utama yang wajib disediakan oleh sistem agar dapat berjalan sesuai dengan tujuan yang telah ditetapkan. Pada penelitian ini, fitur-fitur dirancang untuk mendukung aktivitas pemantauan *cycle time* secara digital. Berikut adalah kebutuhan fungsional pada sistem *monitoring cycle time real-time* studi kasus di perusahaan tempat penelitian.

Tabel 3. 1 Kebutuhan Fungsional

Kode	Kebutuhan Fungsional
FR-01	Sistem mampu menerima dan menyimpan data <i>cycle time</i> dari sensor IR/Proximity pada ESP32 secara otomatis melalui protokol MQTT.
FR-02	Sistem menampilkan data <i>cycle time</i> aktual tiap proses secara <i>real-time</i> melalui dashboard web.
FR-03	Sistem menyediakan fitur pencarian dan penampilan data historis <i>cycle time</i> berdasarkan periode tertentu.
FR-04	Sistem menyediakan fitur ekspor data <i>cycle time</i> ke dalam format CSV untuk keperluan analisis lanjutan.

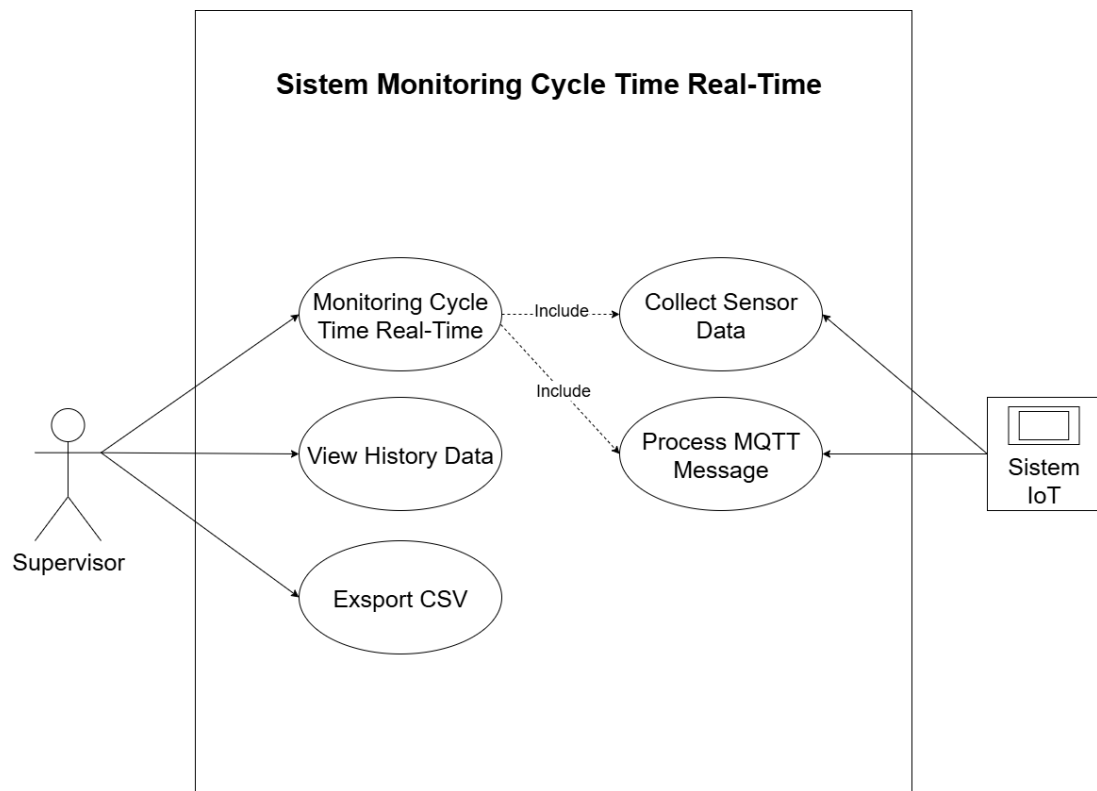
3.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kriteria kualitas yang harus dipenuhi agar sistem bekerja dengan baik, seperti kinerja, keandalan, kemudahan penggunaan, dan ketersediaan. Berikut adalah kebutuhan non-fungsional pada sistem *monitoring cycle time real-time* studi kasus di perusahaan tempat penelitian.

Tabel 3. 2 Kebutuhan Non Fungsional

Kode	Aspek	Kebutuhan Fungsional
NFR-01	Performance Efficiency	Sistem mampu menampilkan pembaruan data sensor ke <i>dashboard</i> dengan latensi < 3 detik (<i>Time Behaviour</i>).
NFR-02	Reliability	Sistem mampu beroperasi stabil dan menangani <i>trigger</i> dari 3 stasiun kerja secara simultan tanpa kegagalan data (<i>packet loss</i>).
NFR-03	Usability	Antarmuka pengguna dapat mudah dipahami oleh pengguna
NFR-04	Capacity	Basis data sistem mampu menampung penyimpanan data historis <i>cycle time</i> minimal untuk periode 1 tahun produksi.

3.2.3 Diagram Use Case



Gambar 3. 2 Usecase Diagram

Gambar 3.2 merupakan diagram *Use Case* sistem *monitoring cycle time real-time*. Diagram ini menggambarkan interaksi aktor Supervisor dengan sistem untuk melakukan fungsi utama seperti *Monitoring Cycle time Real-time*, *View History Data*, dan *Export CSV*, serta proses internal sistem seperti *Collect Sensor Data* dan *Process MQTT Message* dari Sistem IoT.

3.2.4 Skenario Use Case

3.2.4.1 Skenario Usecase *Monitoring Cycle time*

Tabel 3. 3 Skenario Usecase *Monitoring Cycle time*

Kode Use Case	UC 001
Nama Use Case	Monitoring Cycle time
Aktor	Supervisor
Deskripsi	Supervisor melakukan <i>monitoring cycle time</i> produksi secara <i>real-time</i> untuk memantau performa dan efisiensi operasional <i>line</i> produksi
Kondisi Awal	Supervisor sudah mengakses sistem dan sensor IoT sudah aktif dan terhubung ke sistem
Kondisi Akhir	Supervisor berhasil melihat data <i>cycle time real-time</i> dan dapat memantau performa produksi
Alur Kejadian Normal	
Aktor	Sistem
1. Supervisor mengakses <i>dashboard monitoring cycle time</i>	
	2. Sistem menampilkan <i>interface monitoring real-time</i> dengan grafik dan indikator performa <i>in</i>
	3 Sistem secara otomatis mengumpulkan data dari sensor IoT di <i>line</i> produksi
	4 Sistem memproses data sensor dan menghitung <i>cycle time</i> secara <i>real-time</i>
	5. Sistem menampilkan data <i>cycle time</i> terkini dalam bentuk grafik dan angka
6. Supervisor memantau performa produksi melalui <i>dashboard</i> yang terus diperbarui	
Alur Kejadian Alternatif	
1. Jika koneksi sensor IoT terputus, sistem menampilkan peringatan dan menggunakan data terakhir yang <i>valid</i>	

3.2.4.2 Skenario Usecase *View History Data*

Tabel 3. 4 Skenario Usecase *View History Data*

Kode Use Case	UC 002
Nama Use Case	<i>View History Data</i>
Aktor	<i>Supervisor</i>
Deskripsi	Supervisor melihat data historis <i>cycle time</i> untuk analisis trend dan performa produksi periode sebelumnya
Kondisi Awal	Supervisor sudah mengakses sistem dan data historis tersedia di <i>database</i>
Kondisi Akhir	Supervisor berhasil melihat data historis <i>cycle time</i> dalam bentuk tabel
Alur Kejadian Normal	
Aktor	Sistem
1. Supervisor mengakses menu " <i>History Data</i> "	
	2. Sistem menampilkan <i>filter</i> (tanggal, shift, <i>line</i> produksi)
3. Supervisor mengisi kriteria filter yang diinginkan	
	4. Sistem melakukan <i>query database</i> berdasarkan kriteria <i>filter</i>
	5. Sistem menampilkan data historis dalam bentuk tabel dan grafik trend
6. Supervisor menganalisis trend <i>cycle time</i> periode yang dipilih	
Alur Kejadian Alternatif	
1. Jika data terlalu banyak, sistem menampilkan dengan sistem <i>paging</i>	

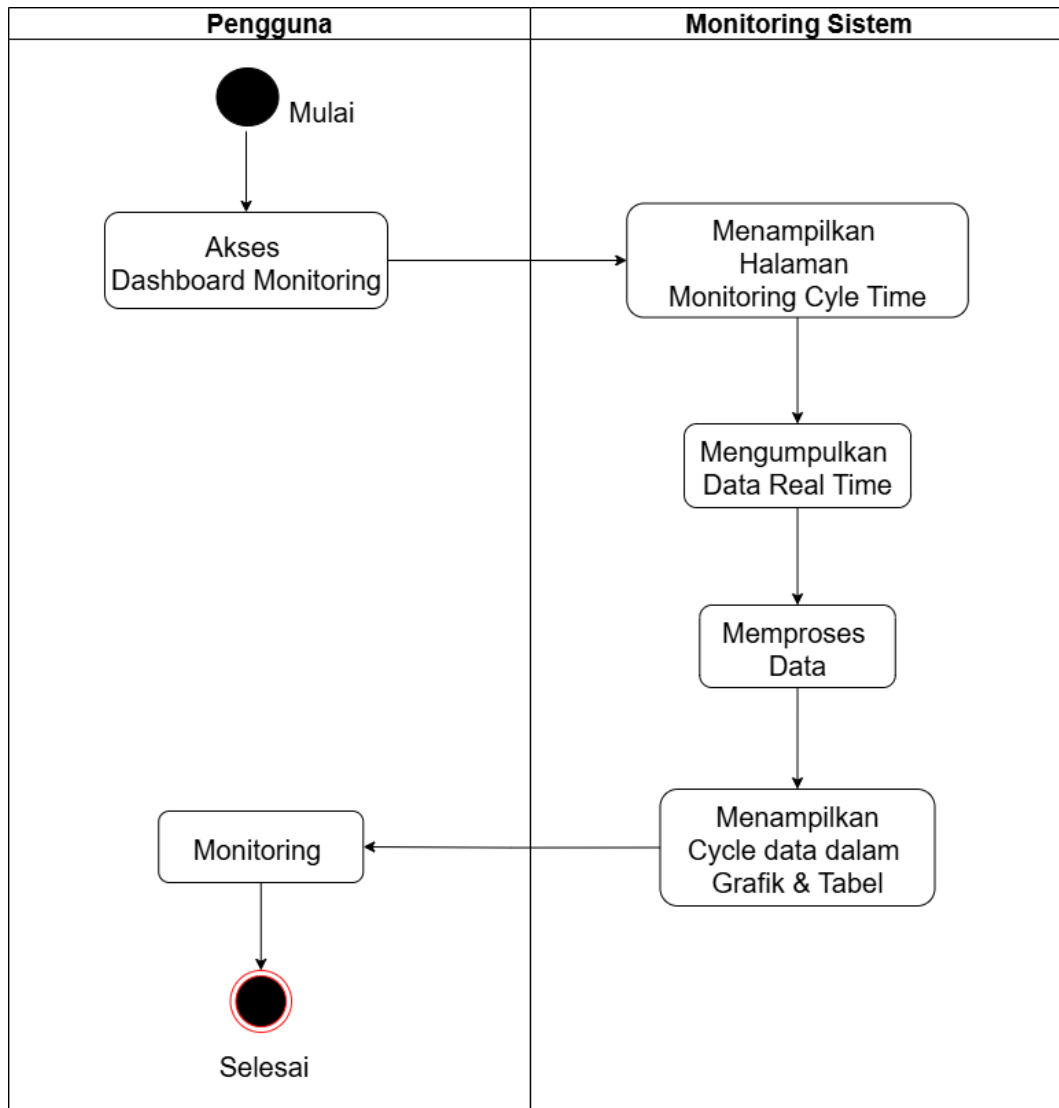
3.2.4.3 Skenario Usecase *Export CSV*

Tabel 3. 5 Skenario Usecase *Export CSV*

Kode Use Case	UC 003
Nama Use Case	Export CSV
Aktor	Supervisor
Deskripsi	Supervisor meng-export data cycle time ke format CSV untuk analisis lebih lanjut atau pembuatan laporan
Kondisi Awal	Supervisor sudah mengakses sistem dan data yang akan di-export sudah tersedia
Kondisi Akhir	File CSV berhasil dibuat dan diunduh oleh supervisor
Alur Kejadian Normal	
Aktor	Sistem
1. Supervisor berada di halaman history data	
2. Supervisor memilih periode data yang akan di-export dengan menggunakan filter	
3. Supervisor menekan tombol "Export to CSV"	
	4. Sistem memproses dan generate file CSV dari data yang dipilih
5. Supervisor mengunduh file CSV ke komputer lokal	
Alur Kejadian Alternatif	
-	

3.2.5 Activity diagram

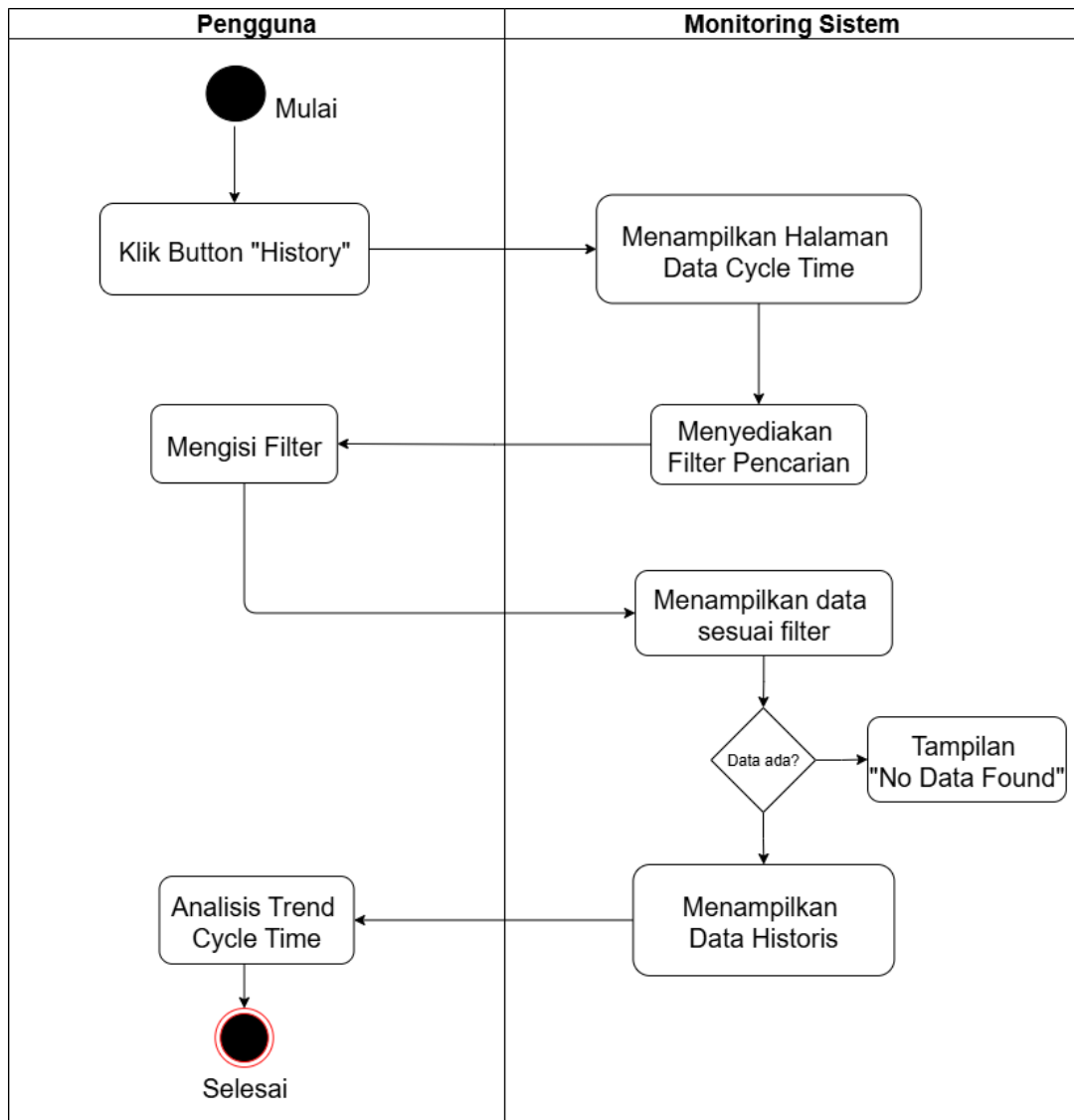
3.2.5.1 Activity Diagram *Monitoring Cycle time*



Gambar 3. 3 Activity Diagram *Monitoring Cycle time*

Gambar 3.3 merupakan *Activity Diagram Monitoring Cycle time*. Diagram ini menggambarkan alur aktivitas pengguna ketika mengakses dashboard *monitoring*, menampilkan halaman *monitoring cycle time*, mengumpulkan data *real-time*, memproses data, hingga menampilkan hasil *cycle data* dalam grafik dan tabel.

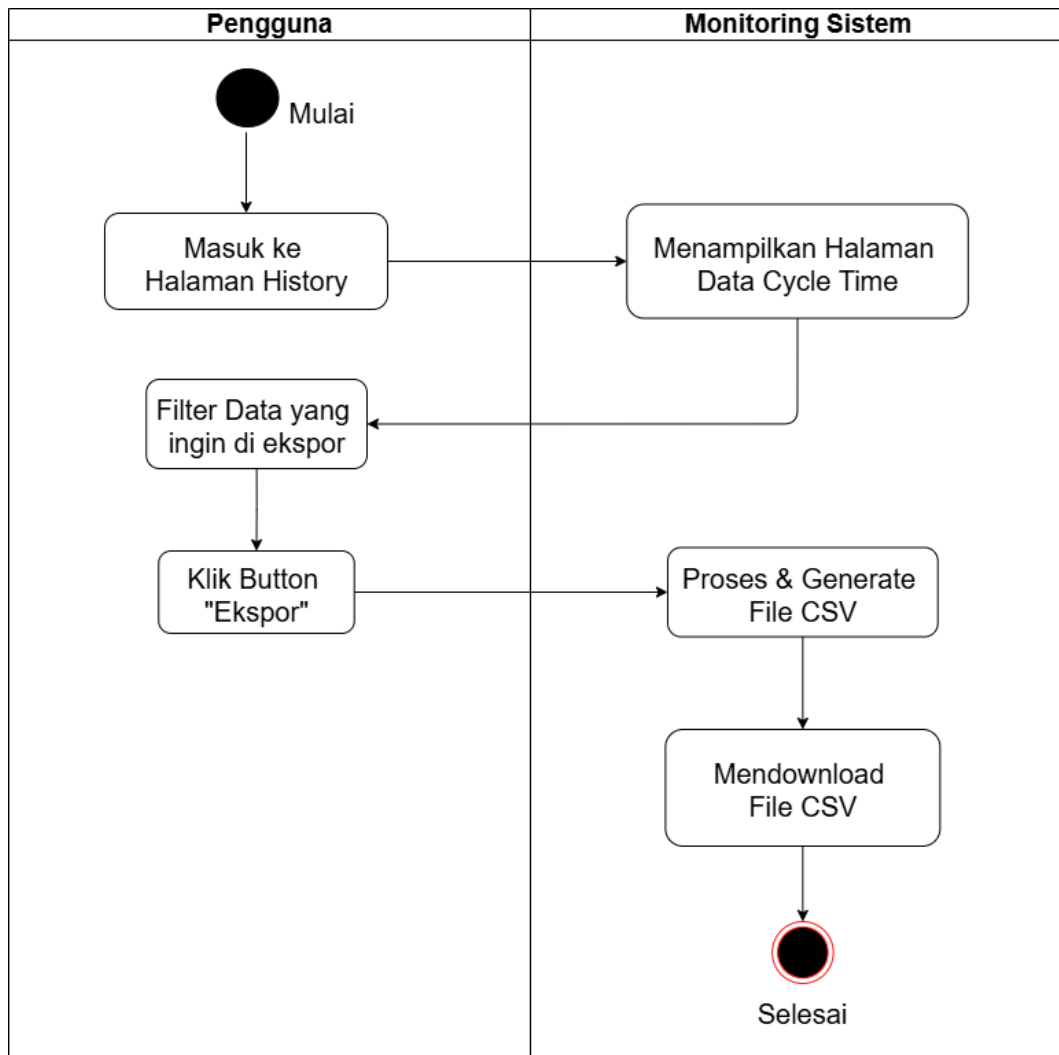
3.2.5.2 Activity Diagram View Data History



Gambar 3. 4 Activity Diagram View Data History

Gambar 3.4 merupakan *Activity Diagram View Data History*. Diagram ini menunjukkan proses pengguna dalam melihat data historis *cycle time* mulai dari mengklik tombol “History”, mengisi filter pencarian, menampilkan data sesuai filter, hingga menganalisis tren *cycle time*.

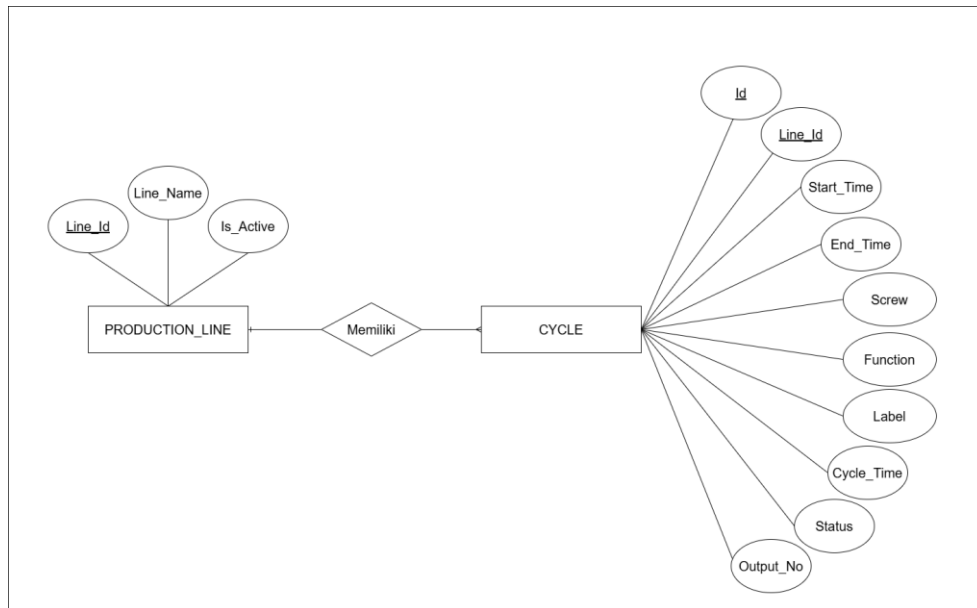
3.2.5.3 Activity Diagram Export CSV



Gambar 3. 5 Activity Diagram Export CSV

Gambar 3.5 merupakan *Activity Diagram* Export CSV. Diagram ini menjelaskan alur pengguna saat mengekspor data *cycle time*, mulai dari masuk ke halaman *history*, memfilter data yang ingin diekspor, mengklik tombol “Ekspor”, hingga sistem memproses dan mengunduh file CSV.

3.2.6 ER Diagram



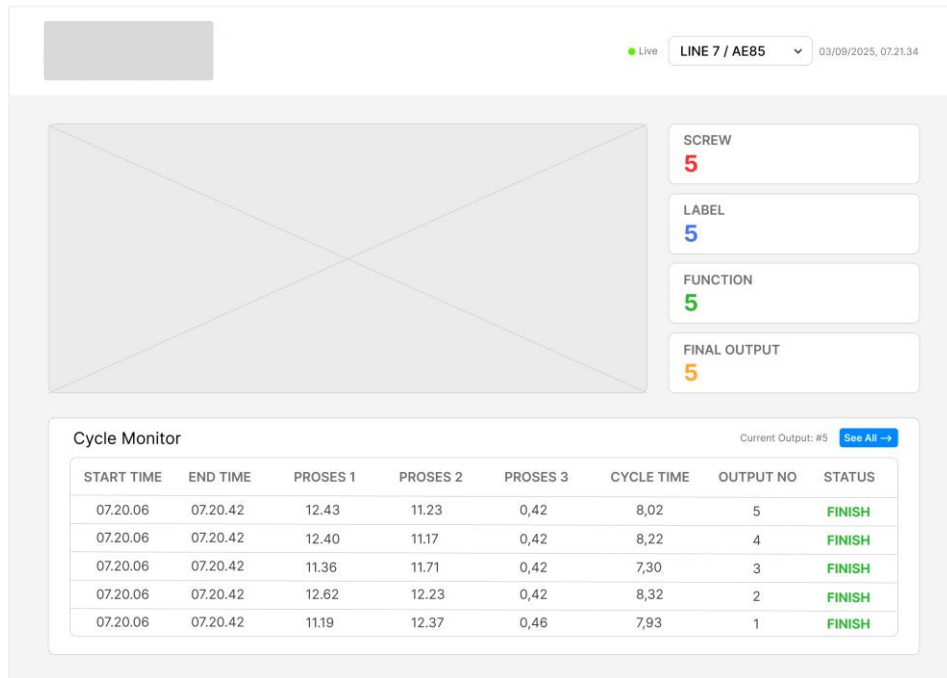
Gambar 3. 6 ER Diagram

Gambar 3.6 merupakan *Entity Relationship (ER) Diagram* sistem *monitoring cycle time*. Diagram ini memperlihatkan hubungan antara entitas PRODUCTION_LINE dan CYCLE beserta atribut-atribut yang dimiliki masing-masing entitas seperti *Line_Id*, *Line_Name*, *Start_Time*, *Screw*, *Function*, *Label*, dan atribut lainnya.

3.2.7 Perancangan Desain Perangkat Lunak

Pada tahapan ini, akan dipaparkan rancangan sistem untuk perancangan perangkat lunak (*software*) seperti tampilan antarmuka dan logika program.

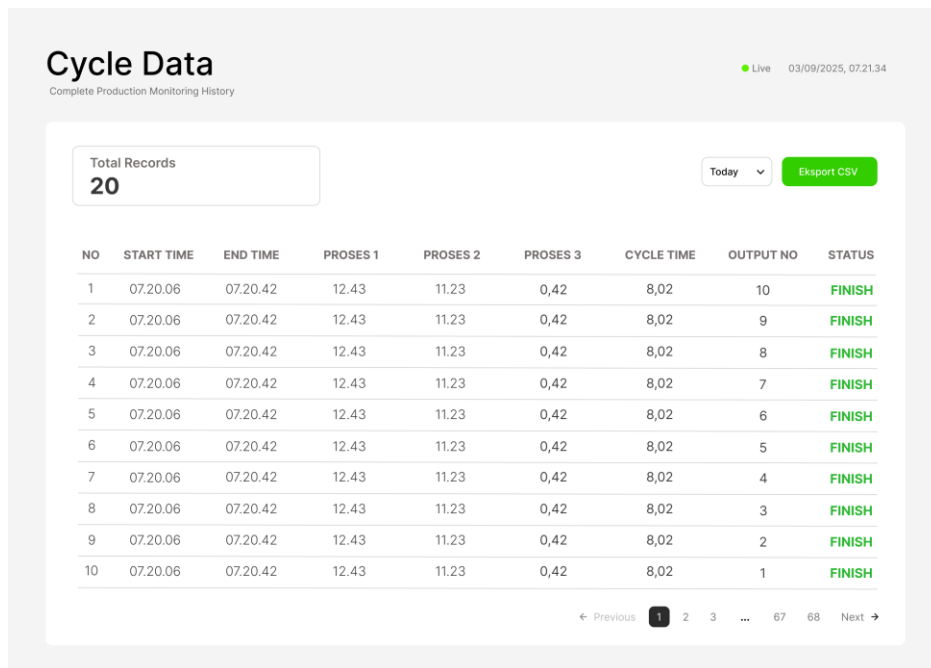
3.2.7.1 Perancangan Antarmuka Dashboard *Monitoring*



Gambar 3. 7 Perancangan Antarmuka Dashboard *Monitoring*

Gambar 3.7 merupakan rancangan antarmuka *Dashboard Monitoring* pada sistem *monitoring cycle time*. Antarmuka ini menampilkan grafik proses individu (*Individual Process Graph*) serta data *cycle time* secara *real-time*, termasuk informasi *Screw*, *Function*, *Label*, dan *Final Output* pada setiap siklus produksi.

3.2.7.2 Perancangan Antarmuka History Data



Cycle Data
Complete Production Monitoring History

● Live 03/09/2025, 07:21:34

Total Records
20

Today ▾ Eksport CSV

NO	START TIME	END TIME	PROSES 1	PROSES 2	PROSES 3	CYCLE TIME	OUTPUT NO	STATUS
1	07.20.06	07.20.42	12.43	11.23	0,42	8,02	10	FINISH
2	07.20.06	07.20.42	12.43	11.23	0,42	8,02	9	FINISH
3	07.20.06	07.20.42	12.43	11.23	0,42	8,02	8	FINISH
4	07.20.06	07.20.42	12.43	11.23	0,42	8,02	7	FINISH
5	07.20.06	07.20.42	12.43	11.23	0,42	8,02	6	FINISH
6	07.20.06	07.20.42	12.43	11.23	0,42	8,02	5	FINISH
7	07.20.06	07.20.42	12.43	11.23	0,42	8,02	4	FINISH
8	07.20.06	07.20.42	12.43	11.23	0,42	8,02	3	FINISH
9	07.20.06	07.20.42	12.43	11.23	0,42	8,02	2	FINISH
10	07.20.06	07.20.42	12.43	11.23	0,42	8,02	1	FINISH

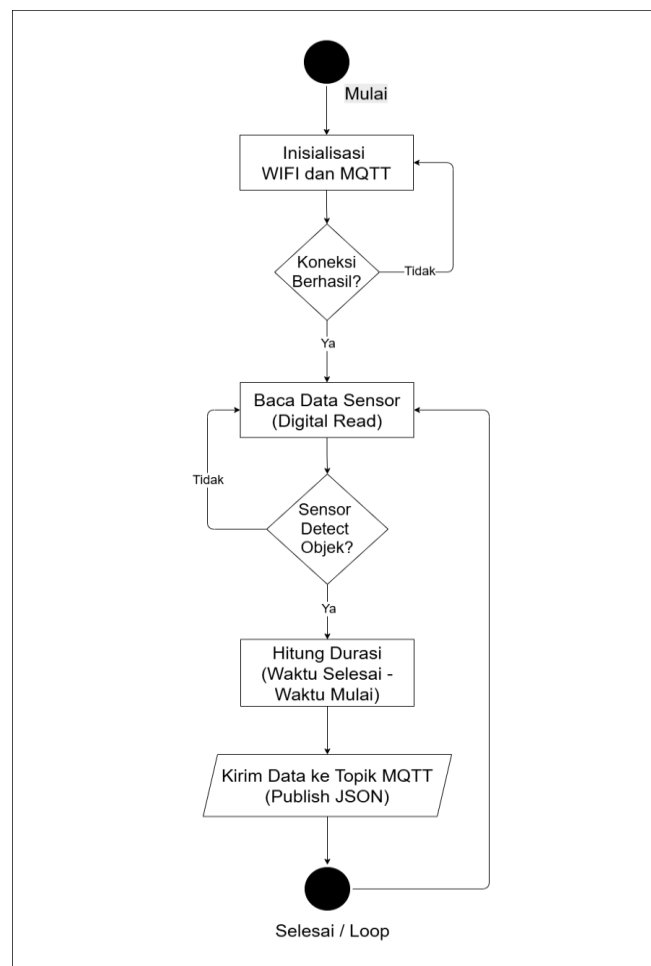
← Previous 1 2 3 ... 67 68 Next →

Gambar 3. 8 Perancangan Antarmuka History Data

Gambar 3.8 merupakan rancangan antarmuka *History Data* pada sistem *monitoring cycle time*. Antarmuka ini menampilkan riwayat lengkap data produksi beserta detail waktu mulai (*Start Time*), waktu selesai (*End Time*), durasi tiap proses, *Cycle time*, *Output No*, serta status proses, dengan fitur *Export CSV* untuk mengunduh data.

3.2.7.3 Perancangan Logika Program (*Firmware*)

Perancangan logika program (*firmware*) menggambarkan algoritma yang ditanamkan pada mikrokontroler ESP32. Alur program dimulai dari inisialisasi koneksi jaringan, pembacaan status sensor pada stasiun kerja, perhitungan *cycle time*, hingga pengiriman data ke server melalui protokol MQTT. Diagram alir logika program dapat dilihat pada Gambar dibawah ini



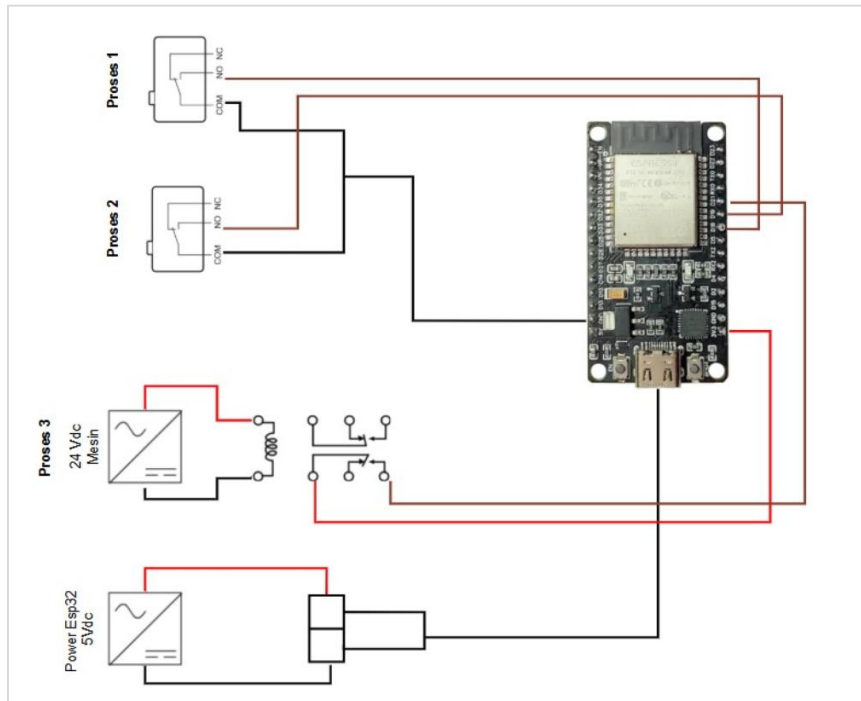
Gambar 3. 9 Perancangan Logika Program (Firmware)

Berdasarkan Gambar 3.9 di atas,

Apabila sensor mendeteksi keberadaan objek, sistem akan mencatat waktu mulai dan waktu selesai untuk menghitung durasi *cycle time*. Data hasil perhitungan tersebut kemudian dikemas dalam format JSON dan dikirimkan (*publish*) ke topik MQTT yang telah ditentukan agar dapat diproses lebih lanjut oleh *server*. Proses

ini akan terus berulang (looping) selama sistem dalam keadaan aktif untuk memantau produksi secara *real-time*.

3.2.8 Perancangan Desain Perangkat Keras



Gambar 3. 10 Perancangan Sistem IoT

Gambar 3.10 Rangkaian sistem IoT untuk *monitoring cycle time*. Rangkaian ini menunjukkan skema koneksi perangkat keras pada tiap proses produksi (Proses 1, Proses 2, Proses 3) yang masing-masing terhubung ke modul ESP32 sebagai pusat pengendali. Sensor pada setiap proses menerima sinyal 24 V/AC, lalu sinyal tersebut dikondisikan dan diteruskan ke pin input ESP32 untuk pencatatan waktu siklus. ESP32 diberi suplai daya 5 V/USB dan mengirimkan data hasil pembacaan sensor ke server melalui jaringan Wi-Fi. Dengan rancangan ini sistem mampu memantau dan mengukur *cycle time* secara otomatis dan *real-time*.

BAB IV IMPLEMENTASI DAN PENGUJIAN

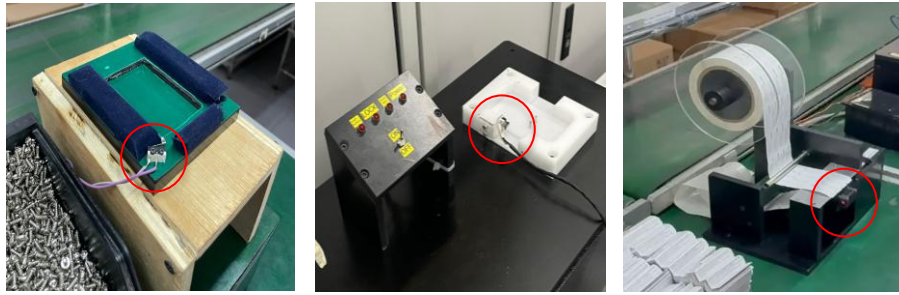
4.1 Hasil Implementasi

Implementasi sistem ini merupakan realisasi dari perancangan arsitektur IoT modern yang memanfaatkan ESP32, protokol MQTT, Node.js, dan WebSocket. Sistem ini bertujuan untuk mendigitalisasi proses *monitoring* di *line* produksi guna menggantikan pencatatan manual. Berikut adalah rincian hasil implementasi sistem

4.1.1 Implementasi Perangkat Keras (Hardware)

Implementasi perangkat keras melibatkan pemasangan mikrokontroler ESP32 dan sensor pada tiap proses di *line* produksi *assembly*. Perangkat keras ini berfungsi untuk mendeteksi status proses dan mengirimkan data secara nirkabel.

- **Mikrokontroler:** Menggunakan modul ESP32 yang terhubung ke jaringan Wi-Fi lokal perusahaan.
- **Sensor:** Implementasi dilakukan pada tiga tahapan proses utama yaitu *Screw*, *Function*, dan *Label*. Sinyal input dari mesin/switch (tegangan 24 V/DC) dikondisikan menggunakan rangkaian *level converter* agar dapat dibaca oleh pin *input* digital ESP32 (3.3 V).
- **Topologi:** Sensor pada tiap proses terhubung kedalam satu microcontroller, yaitu ESP32 yang bertugas mem-publish data ke broker MQTT
- **Pemrograman:** Logika kendali pada mikrokontroler diprogram menggunakan bahasa C++ melalui lingkungan pengembangan Arduino IDE. Program dirancang untuk membaca perubahan sinyal digital sensor secara *real-time* dan mengirimkan paket data ke server.



Gambar 4. 1 Set Up Sensor pada *Line* Produksi

Gambar 4.1 memperlihatkan implementasi sensor yang disesuaikan dengan stasiun kerja. Pada stasiun *Screw* dan *Function* (kiri & tengah), digunakan *limit switch* pada *jig* yang akan mengirim sinyal *trigger* secara mekanis saat tertekan oleh produk. Berbeda dengan stasiun *Label* (kanan), sistem memanfaatkan sensor fotoelektrik (*photoelectric*) bawaan mesin untuk mendeteksi keluaran label secara otomatis tanpa penambahan sakelar manual.

4.1.2 Implementasi Backend Server

Backend dibangun menggunakan *environment* Node.js dan *framework* Express.js. Modul MQTT Client digunakan untuk melakukan *subscribe* ke seluruh topik data yang dikirim oleh ESP32. Setiap kali *broker* MQTT menerima data baru, *backend* akan melakukan proses berikut secara sekuensial:

1. Memproses *payload* JSON yang masuk.
2. Menghitung durasi *cycle time* berdasarkan *timestamp*.
3. Menyimpan data rekaman produksi ke dalam *database* MySQL.
4. Meneruskan data terbaru ke *dashboard* klien melalui protokol WebSocket secara *real-time*.

Mekanisme WebSocket ini diimplementasikan untuk memastikan *dashboard* menerima pembaruan visual dalam waktu singkat (< 3 detik) setelah data dipicu di rantai produksi.

4.1.3 Implementasi Antar Muka Pengguna

Sisi perangkat lunak di sisi klien (*client-side*) diimplementasikan dalam bentuk *dashboard* web berbasis React.js yang responsif. *Dashboard* ini menjadi antarmuka utama bagi Supervisor untuk memantau produktivitas.

4.1.3.1 Halaman Dashboard *Monitoring Real-time*

Halaman ini adalah tampilan utama yang menyajikan data secara *live*. Sesuai dengan rancangan antarmuka, halaman ini menampilkan:

- **Grafik Proses Individu:** Visualisasi *cycle time* (detik) untuk proses *Screw*, *Function*, dan *Label* yang bergerak dinamis saat data masuk.
- **Indikator Counter:** Menampilkan jumlah output aktual pada setiap proses dan total output final.
- **Tabel Monitor Siklus:** Menampilkan detail data per produk yang sedang berjalan, mencakup *Start Time*, *End Time*, durasi setiap proses, dan status penyelesaian.



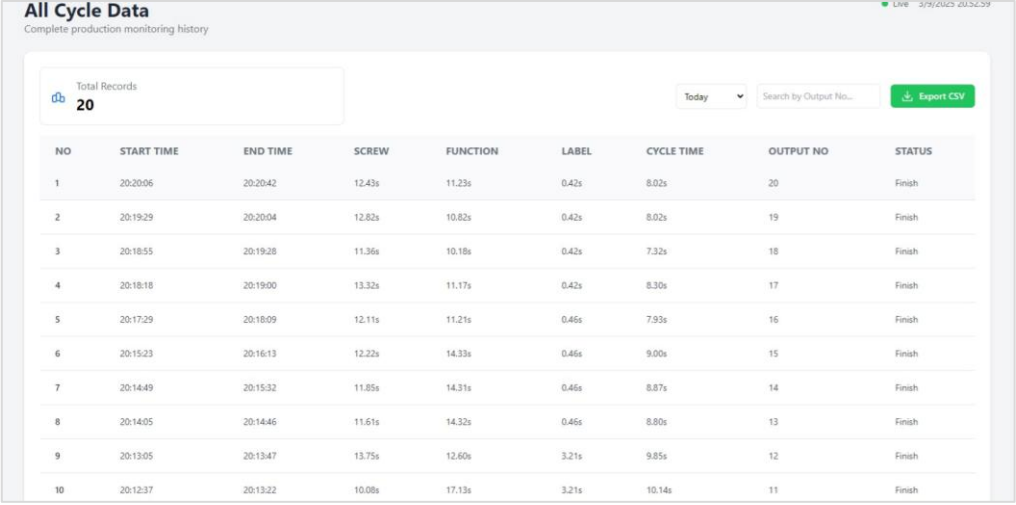
Gambar 4. 2 Dashboard *Monitoring*

Gambar 4.2 merupakan Implementasi *Dashboard Monitoring* pada sistem *monitoring cycle time*. Antarmuka ini menampilkan grafik proses individu (*Individual Process Graph*) serta data *cycle time* secara *real-time*, termasuk informasi *Screw*, *Function*, *Label*, dan *Final Output* pada setiap siklus produksi.

4.1.3.2 Halaman Riwayat Data (History Data)

Fitur ini diimplementasikan untuk memenuhi kebutuhan analisis data historis dan audit jejak produksi. Pada halaman ini, Supervisor dapat:

- Melihat seluruh rekam jejak data produksi yang tersimpan di *database* MySQL.
- Melakukan pemfilteran data berdasarkan rentang tanggal, *shift*, atau nomor *output*.
- Menggunakan fitur Export CSV untuk mengunduh laporan produksi guna analisis lanjutan di *spreadsheet*.



The screenshot displays a web interface titled "All Cycle Data" with the subtitle "Complete production monitoring history". It features a "Total Records" summary showing 20 records, a date filter set to "Today", a search bar for "Output No.", and an "Export CSV" button. Below this is a table with 10 rows of production data. Each row contains the following information: NO, START TIME, END TIME, SCREW, FUNCTION, LABEL, CYCLE TIME, OUTPUT NO, and STATUS. All records in the table show a status of "Finish".

NO	START TIME	END TIME	SCREW	FUNCTION	LABEL	CYCLE TIME	OUTPUT NO	STATUS
1	20:20:06	20:20:42	12.43s	11.23s	0.42s	8.02s	20	Finish
2	20:19:29	20:20:04	12.82s	10.82s	0.42s	8.02s	19	Finish
3	20:18:55	20:19:28	11.36s	10.18s	0.42s	7.32s	18	Finish
4	20:18:18	20:19:00	13.32s	11.17s	0.42s	8.30s	17	Finish
5	20:17:29	20:18:09	12.11s	11.21s	0.46s	7.93s	16	Finish
6	20:15:23	20:16:13	12.22s	14.33s	0.46s	9.00s	15	Finish
7	20:14:49	20:15:32	11.85s	14.31s	0.46s	8.87s	14	Finish
8	20:14:05	20:14:46	11.61s	14.32s	0.46s	8.80s	13	Finish
9	20:13:05	20:13:47	13.75s	12.60s	3.21s	9.85s	12	Finish
10	20:12:37	20:13:22	10.08s	17.13s	3.21s	10.14s	11	Finish

Gambar 4. 3 History Data

Gambar 4.3 merupakan Implementasi *History Data* pada sistem *monitoring cycle time*. Antarmuka ini menampilkan riwayat lengkap data produksi beserta detail waktu mulai (*Start Time*), waktu selesai (*End Time*), durasi tiap proses, *Cycle time*, *Output No*, serta status proses, dengan fitur *Export CSV* untuk mengunduh data.

4.1.4 Alur Penggunaan Sistem

Berdasarkan implementasi di atas, alur penggunaan sistem oleh *user* berjalan sebagai berikut:

1. Supervisor membuka *dashboard monitoring* pada web browser.
2. *Dashboard* secara otomatis menampilkan data *real-time* dari setiap proses pada *line* produksi tanpa perlu melakukan *refresh* halaman.
3. Supervisor memantau indikator *cycle time*
4. Untuk evaluasi harian, Supervisor membuka menu History Data guna melihat rekapitulasi performa produksi.
5. Jika diperlukan laporan fisik atau analisis mendalam, Supervisor melakukan Export CSV untuk mengunduh data *shift* tersebut.

4.2 Pengujian Sistem

Pengujian dilakukan untuk memverifikasi apakah sistem yang dibangun telah memenuhi spesifikasi kebutuhan perangkat lunak dan standar kualitas yang diharapkan. Kerangka kerja pengujian dalam penelitian ini mengacu pada standar kualitas perangkat lunak ISO/IEC 25010, dengan fokus utama pada aspek *Functional Suitability* (Kesesuaian Fungsi) dan *Performance Efficiency* (Efisiensi Kinerja) (Ratumbuisang et al., 2023).

Tujuan utama pengujian ini adalah untuk memvalidasi bahwa logika program yang ditanamkan pada perangkat keras (firmware) dan antarmuka web dapat beroperasi dengan benar (*functional correctness*) serta memiliki respons waktu (*time behaviour*) yang sesuai dengan kebutuhan pemantauan *real-time*.

4.2.1 User Acceptance Testing (UAT)

Pengujian *User Acceptance Testing* (UAT) atau uji penerimaan pengguna adalah proses pengujian yang dilakukan oleh pengguna akhir untuk menghasilkan dokumen yang dijadikan bukti bahwa perangkat lunak yang dikembangkan dapat diterima dan memenuhi kebutuhan pengguna (Khusna et al., 2021). Pengujian ini bertujuan untuk memvalidasi apakah sistem *monitoring cycle time* berbasis IoT yang dibangun telah berjalan sesuai dengan prosedur operasional di Perusahaan Manufaktur

Tabel 4. 1 *User Acceptance Testing*

Kode	Deskripsi Requirements	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
FR-01	Sistem menerima dan menyimpan data <i>cycle time</i> melalui MQTT	Sensor dipicu pada proses produksi	Data <i>cycle time</i> tersimpan otomatis di <i>database</i>	Sesuai	☒ Berhasil ☐ Gagal
FR-02	Sistem menampilkan data <i>cycle time</i> secara <i>real-time</i>	Dashboard dibuka saat produksi berjalan	Data <i>cycle time</i> tampil <i>real-time</i>	Sesuai	☒ Berhasil ☐ Gagal

FR-03	Sistem menampilkan data historis <i>cycle time</i>	Filter data berdasarkan periode	Data historis tampil sesuai periode	Sesuai	☒ Berhasil ☐ Gagal
FR-04	Sistem mengekspor data ke format CSV	Klik tombol ekspor	File CSV berhasil diunduh	Sesuai	☒ Berhasil ☐ Gagal

4.2.2 Verifikasi Validitas Data (*Functional Verification*)

Pengujian ini bertujuan untuk membuktikan aspek *Functional Correctness* (Kebenaran Fungsi) sesuai standar ISO/IEC 25010 (Ratumbuisang et al., 2023). Pengujian dilakukan untuk memverifikasi bahwa sensor dan algoritma pada mikrokontroler ESP32 mampu menangkap *cycle time* dengan benar dan linier terhadap kejadian fisik yang sebenarnya. Mengingat penelitian ini merupakan pengembangan *prototyping*, validasi dilakukan menggunakan metode perbandingan langsung (*direct comparison*) terhadap 20 sampel siklus produksi acak. Jumlah sampel ini digunakan sebagai representasi uji fungsional untuk mendeteksi adanya penyimpangan sistematis (*systematic error*) pada pembacaan sensor, bukan sebagai uji kapabilitas proses statistik jangka panjang. Metode pengujian dilakukan dengan membandingkan data waktu sistem (*System Time*) terhadap alat ukur standar stopwatch (*Manual Time*). Tingkat validitas diukur berdasarkan persentase deviasi (selisih) antara kedua pengukuran tersebut. Adapun rumus persentase selisih (deviasi) yang digunakan untuk melihat kedekatan nilai pengukuran adalah sebagai berikut:

$$Error (\%) = \frac{T_{Manual} - T_{Sistem}}{T_{Manual}} \times 100\%$$

Keterangan:

- T_{Manual} : Waktu pengukuran menggunakan stopwatch (detik)
- T_{Sistem} : Waktu pengukuran oleh sistem IoT (detik)

Tabel 4. 2 Hasil Verifikasi Validitas Data

No	Waktu Manual (Detik)	Waktu Sistem IoT (Detik)	Selisih (Detik)	Error (%)
1	11.23	11.11	0.12	1.08%
2	10.94	11.15	0.21	1.88%
3	11.41	11.28	0.13	1.15%
4	11.36	11.20	0.16	1.43%
5	11.05	10.98	0.07	0.64%
6	10.92	11.14	0.22	1.97%
7	11.48	11.42	0.06	0.53%
8	11.13	11.29	0.16	1.42%
9	11.45	11.24	0.21	1.87%
10	11.08	10.96	0.12	1.09%
11	11.12	11.34	0.22	1.94%
12	11.34	11.18	0.16	1.43%
13	11.16	11.30	0.01	0.09%
14	11.23	11.04	0.03	0.27%
15	11.31	11.15	0.16	1.43%
16	11.26	11.14	0.12	1.08%
17	11.38	11.23	0.15	1.34%
18	11.02	11.25	0.23	0.35%
19	11.11	11.05	0.06	0.54%
20	11.43	11.29	0.14	1.24%

Berdasarkan data sampel verifikasi pada Tabel 4.2, diperoleh hasil analisis sebagai berikut:

1. **Deviasi Rata-rata:** Sistem menunjukkan rata-rata deviasi pengukuran sebesar 0,98%. Angka ini menunjukkan bahwa algoritma perhitungan waktu pada firmware ESP32 berjalan dengan logika yang benar karena menghasilkan selisih yang sangat kecil ($< 1\%$) terhadap alat ukur manual.
2. **Konsistensi Pembacaan:** Selisih waktu absolut rata-rata berada pada kisaran 0,13 detik. Nilai ini masih berada dalam batas toleransi waktu reaksi manusia (*human reaction time*) saat mengoperasikan stopwatch manual, sehingga perbedaan yang terjadi dapat dikategorikan sebagai variasi wajar dan bukan kesalahan sistem. Dengan demikian, sistem dinyatakan valid secara fungsional untuk digunakan sebagai instrumen pencatat waktu otomatis menggantikan metode manual.

4.2.3 Pengujian Efisiensi Kerja (*Time Behaviour*)

Pengujian ini difokuskan pada aspek Time Behaviour (Perilaku Waktu) dalam standar ISO/IEC 25010 untuk mengukur latensi end-to-end sistem (Ratumbuisang et al., 2023). Latensi didefinisikan sebagai selisih waktu antara saat data dikirim (publish) oleh sensor di lantai produksi hingga data tersebut tampil secara visual di dashboard pengguna. Pengujian dilakukan dengan menyimulasikan 20 kali transmisi data menggunakan protokol MQTT dan WebSocket. Target performa yang ditetapkan adalah latensi di bawah 3 detik untuk memenuhi kriteria pemantauan soft *real-time*.

Tabel 4. 3 Hasil Pengujian Time Behaviour

No	Waktu Trigger (Sensor)	Waktu Tampil (Dashboard)	Latensi (Detik)	Status (< 3 Detik)
1	21:24:26:298	21:24:26:342	0.044	Berhasil
2	21:24:29:300	21:24:29:333	0.033	Berhasil
3	21:24:32:304	21:24:32:337	0.033	Berhasil
4	21:24:35:305	21:24:35:338	0.033	Berhasil
5	21:24:38:308	21:24:38:344	0.036	Berhasil
6	21:24:41:314	21:24:41:343	0.029	Berhasil
7	21:24:44:329	21:24:44:361	0.032	Berhasil
8	21:24:47:330	21:24:47:360	0.030	Berhasil
9	21:24:50:334	21:24:50:370	0.036	Berhasil
10	21:24:53:335	21:24:53:371	0.036	Berhasil
11	21:24:56:340	21:24:56:371	0.031	Berhasil
12	21:24:59:344	21:24:59:379	0.035	Berhasil
13	21:25:02:344	21:25:02:384	0.040	Berhasil
14	21:25:05:346	21:25:05:382	0.036	Berhasil
15	21:25:08:350	21:25:08:382	0.032	Berhasil
16	21:25:11:365	21:25:11:401	0.036	Berhasil
17	21:25:14:372	21:25:14:405	0.033	Berhasil
18	21:25:17:379	21:25:17:417	0.038	Berhasil
19	21:25:20:385	21:25:20:420	0.035	Berhasil
20	21:25:23:397	21:25:23:431	0.034	Berhasil

Hasil pengukuran pada Tabel 4.3 menunjukkan kinerja sistem sebagai berikut:

1. **Responsivitas Tinggi:** Rata-rata latensi sistem tercatat sebesar 0,035 detik (35 ms).

2. **Stabilitas:** Rentang latensi stabil antara 0,029 detik hingga 0,044 detik, yang mengindikasikan koneksi jaringan dan pemrosesan data pada *Backend Server* berjalan efisien tanpa adanya antrean data (*bottleneck*) yang signifikan.

Hasil ini memvalidasi bahwa arsitektur sistem yang dirancang mampu menangani pengiriman data produksi dengan kecepatan tinggi, jauh melampaui batas toleransi maksimal (3 detik) yang ditetapkan.

BAB V KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem *monitoring cycle time real-time* berbasis IoT di Perusahaan Manufaktur, dapat ditarik beberapa kesimpulan utama sebagai jawaban atas tujuan penelitian yang telah ditetapkan:

1. Perancangan dan pengembangan aplikasi berbasis IoT telah berhasil dilakukan untuk menggantikan pencatatan manual. Berdasarkan hasil Verifikasi Validitas Data (*Functional Verification*), sistem menunjukkan kinerja fungsional yang baik dengan rata-rata deviasi pengukuran sebesar 0,98% terhadap pengukuran manual. Hasil ini mengindikasikan bahwa sensor dan logika program pada mikrokontroler telah berfungsi dengan benar (*functionally correct*) dan valid untuk digunakan sebagai alat bantu *monitoring* produksi.
2. Sistem terbukti mampu menyajikan informasi produksi dengan responsivitas tinggi. Hasil Pengujian Efisiensi Kinerja (*Time Behaviour*) menunjukkan bahwa integrasi protokol MQTT dan WebSocket mampu mengirimkan data dengan latensi rata-rata 0,035 detik (35 ms). Nilai ini memenuhi kriteria pemantauan *soft real-time* yang ditetapkan (< 3 detik), sehingga memungkinkan manajemen melihat kondisi lini produksi tanpa jeda visual yang signifikan.
3. Penerapan arsitektur *event-driven* terbukti efektif dalam menangani aliran data produksi yang cepat dibandingkan metode *polling* konvensional. Fitur pendukung seperti visualisasi grafik dinamis, penyimpanan data historis, dan ekspor CSV telah berfungsi dengan baik, memberikan kemudahan bagi manajemen untuk melakukan analisis produktivitas harian secara komprehensif.

5.2 Saran

Demi pengembangan dan penyempurnaan sistem di masa mendatang, serta optimalisasi implementasi di lapangan, penulis menyampaikan beberapa saran sebagai berikut:

1. Pengujian Skala Besar (*Long-Term Testing*) Pengujian dalam penelitian ini dilakukan dalam skala terbatas (*pilot testing*) dengan jumlah sampel 20 siklus untuk memverifikasi fungsionalitas sistem. Disarankan untuk penelitian selanjutnya melakukan uji validasi statistik dengan jumlah sampel yang lebih besar (misalnya >100 siklus) atau pengujian durabilitas dalam jangka waktu panjang (misalnya 1 minggu *non-stop*) untuk mengetahui konsistensi akurasi sensor terhadap faktor lingkungan industri yang dinamis.
2. Integrasi Analisis Prediktif Sistem saat ini fokus pada *monitoring* deskriptif (apa yang sedang terjadi). Kedepannya, data historis yang telah terkumpul dapat diolah menggunakan algoritma *Machine Learning* untuk fitur *Predictive Maintenance* atau prediksi *bottleneck*, sehingga sistem dapat memberikan peringatan dini sebelum masalah produksi terjadi.
3. Skalabilitas dan Infrastruktur Jaringan Untuk implementasi skala penuh di seluruh *line* produksi pabrik, disarankan untuk melakukan evaluasi kapasitas jaringan Wi-Fi industri agar mampu menangani lonjakan trafik data dari puluhan perangkat ESP32 secara simultan tanpa mengalami *packet loss*.

DAFTAR PUSTAKA

- Chulakit, S., Sadun, A. S., Jalaludin, N. A., Hashim, H., Ahmad, S., Sabarudin, N. A., Fauzi, M. A., & Wang, Z. (2024). *New Approach of Process Cycle time Measurement System with Sensor Application*. *Journal of Advanced Research Design*, 123(1), 170–185. <https://doi.org/10.37934/ard.123.1.170185>
- Achmadi, F., Harsanto, B., & Yunani, A. (2021). Analisis cycle time proses perakitan senjata di PT Pindad (Persero). *Operations Excellence: Journal of Applied Industrial Engineering*, 13(2), 159. <https://doi.org/10.22441/oe.2021.v13.i2.015>
- Chulakit, S., Sadun, A. S., Jalaludin, N. A., Jalani, J., Ahmad, S., Hanapi, M. H. M., & Sabarudin, N. A. (2023). A Centralized IOT-Based Process Cycle time Monitoring System for Line Balancing Study. *Journal of Advanced Research in Applied Mechanics*, 105(1), 58–67. <https://doi.org/10.37934/aram.105.1.5867>
- Putra, P. O., Setiawan, H., & Mair, Z. R. (2023). Implementasi Alat Monitoring Suhu Ruangan Berbasis Internet Of Things (IoT) Menggunakan Metode MQTT dan HTML Pada Ruangan Server Universitas Indo Global Mandiri Palembang. *Jurnal Software Engineering and Computational Intelligence*, 1(1), 40–50. <https://doi.org/10.36982/jseci.v1i1.3046>
- Zulfadlillah, Z., Fathani, M. A., Noval, M., & Irwana, I. (2024). Penerapan Sistem Manufacturing 4.0 dengan Integrasi Internet of Things (IoT) untuk Optimalisasi Efisiensi Produksi. *Jurnal Ilmiah Teknik Mesin, Elektro Dan Komputer*, 4(1), 159–164. <https://doi.org/10.51903/juritek.v4i1.4041>
- Wicaksono, A. A., Ningsih, Y. K., & Surjati, I. (2024). Implementation of MQTT Protocol on ESP32-Based OEE Analysis Development Board. *MITOR: Jurnal Teknik Elektro*, 24(2), 169–175. <https://doi.org/10.23917/emitro.v24i2.3908>
- Pratama, R. F., Wicaksono, R. S. R., & Pramudhita, A. N. (2023). Perancangan dan Implementasi Protokol MQTT Pada Sistem Parkir Cerdas Berbasis IoT. *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, 11(3), 475–482. <https://doi.org/10.23960/jitet.v11i3.3191>

- Hartanto, M. B., Putra, A. S., & Fawa'ati, T. M. (2024). Analisis Dampak Implementasi Internet Of Things (IOT) Terhadap Efisiensi Operasional Di Industri Manufaktur. *Jurnal Multimedia dan Android (JMA)*, 5(1), 1–12. <https://doi.org/10.36982/jma.v5i1.1835>
- Ridla, M. A., & Rahman, M. F. (2024). Perancangan Prototype *Monitoring* Suhu Berbasis Internet Of Things (IoT). *JUSIFOR: Jurnal Sistem Informasi dan Informatika*, 3(1), 72–79. <https://doi.org/10.33379/jusifor.v3i1.4367>
- Aprilisa, S., & Aulia, R. (2024). Penerapan metode prototype dalam pengembangan sistem informasi inventory barang berbasis web. *JUTIN: Jurnal Teknik Industri Terintegrasi*, 7(1), 333–340. <https://doi.org/10.31004/jutin.v7i1.24749>
- Khusna, A. N., Delasano, K. P., & Saputra, D. C. E. (2021). Penerapan User-Based Collaborative Filtering Algorithm: Studi Kasus Sistem Rekomendasi untuk Menentukan Gadget Shield. *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 20(2), 293–304. <https://doi.org/10.30812/matrik.v20i2.1124>
- As, A., Hartono, N., & Muin, A. (2024). Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. *Switch: Jurnal Sains dan Teknologi Informasi*, 3(1), 84–100. <https://doi.org/10.62951/switch.v3i1.330>
- Wahyudi, I., Fahrullah, F., Alameka, F., & Haerullah, H. (2023). Analisis Blackbox Testing dan User Acceptance Testing Terhadap Sistem Informasi Solusimedsosku. *Jurnal Teknosains Kodepena*, 4(1), 1–9.
- Ratumbuisang, Y. Y., Kountul, C., & Al Fikri, M. (2023). ISO/IEC 25010 for quality testing of Labor Market Information System (SIPTK). *Jurnal Fokus Elektroda*, 8(3), 153–159.

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement

https://drive.google.com/drive/folders/18WqRZA0QPxPIcKJ4reWa1U80poVbjdb_o?usp=drive_link

Lampiran B. Link Produk

<https://github.com/elshaamalia/giken-project-2.git>

Lampiran C. Dokumen Pengujian UAT

https://drive.google.com/drive/folders/1XhHBILQCV9TffC8fHMa79x5Vjx-hw6vP?usp=drive_link