PENGEMBANGAN SISTEM OTOMATISASI *FEEDBACK* PADA DIVISI *ENGINEERING* PT UNGGUL LANCAR SEJAHTERA

Rahmat Wahyu Budiyanto*, Kristina Hutajulu, Ning Ratwastuti
Program Studi Manajemen Informatika, Politeknik Astra, Jakarta, Indonesia
Email: rahmatwahyubudiyanto@gmail.com*

Abstrak*-- PT Unggul Lancar Sejahtera (ULS) *is a well-known automotive company in Indonesia with high production targets at each plant to meet customer demand. To achieve this, the company ensures that production machines operate in optimal condition. If the machines are not running properly, downtime may occur, which can disrupt production. To monitor machine conditions more effectively, ULS developed the System of Feedback Automation (SOF), which automatically sends notifications and repair suggestions when certain machine parameters MTTR, MTBF, and CO_2 exceed their threshold values. The system sends these notifications via email and Telegram. The SOF was developed using the Waterfall methodology, as the system requirements were clearly defined from the start and only minor changes were expected during development. It was implemented using the Java Spring framework and Oracle Database. The testing phase including unit test, SIT, UAT, and QAT showed that the system successfully sent 102 notifications (12 MTTR, 36 MTBF, and 54 CO_2) according to the defined schedule. These results prove that SOF effectively automates machine condition monitoring, allowing the monitoring process to be carried out more organized.

***Keywords: Production Target, Downtime, Waterfall*, Java, MotionBoard.**

Abstrak*-- PT Unggul Lancar Sejahtera (ULS) adalah perusahaan otomotif ternama di Indonesia dengan target produksi yang tinggi pada setiap *plant* untuk memenuhi permintaan konsumen. Dalam memenuhi permintaan tersebut, perusahaan memastikan mesin produksi dalam kondisi yang optimal untuk memproduksi kendaraan. Jika kondisi mesin tidak optimal maka hal terburuk yang dapat terjadi adalah *downtime*, akibatnya berbagai upaya dilakukan untuk memantau kondisi mesin secara lebih teratur. PT ULS berupaya mengembangkan Sistem Otomatisasi *Feedback* (SOF) untuk mengirimkan notifikasi berupa *feedback* terkait potensi kerusakan mesin beserta usulan perbaikannya pada parameter data MTTR, MTBF, dan CO_2 melalui email dan Telegram. Pengembangan SOF menggunakan metodologi *waterfall* karena kebutuhan sistem telah jelas sejak awal dan kemungkinan perubahan selama proses pengembangan sangat kecil. Pada pengembangan SOF memanfaatkan *framework* Java Spring dan Oracle *Database*. Hasil pengujian SOF meliputi *unit test*, SIT, UAT, dan QAT menunjukkan bahwa sistem berhasil mengirimkan 102 notifikasi (12 MTTR, 36 MTBF, dan 54 CO_2) sesuai jadwal yang ditentukan. Hasil ini membuktikan bahwa SOF berfungsi dalam mengotomatisasi pemantauan kondisi mesin sehingga proses pemantauan dapat dilakukan secara lebih teratur.

***Kata Kunci:* Target Produksi, *Downtime*, *Waterfall*, Java, MotionBoar.**

I. PENDAHULUAN

PT Unggul Lancar Sejahtera (ULS) adalah perusahaan otomotif ternama di Indonesia. Pada tahun 2024, perusahaan mencatat penjualan jutaan kendaraan yang menunjukkan bahwa produk mereka sangat diterima oleh masyarakat Indonesia. Dalam memenuhi permintaan tersebut, perusahaan mengoperasikan enam *plant* produksi yang tersebar di berbagai lokasi untuk memproduksi berbagai jenis kendaraan guna memenuhi permintaan konsumen.

Pada tahun 2022 – 2024, perusahaan mencatat rata-rata waktu mesin berhenti pada periode tertentu (*downtime)* pada suatu *plant* akibat mesin yang tidak beroperasi secara optimal dalam kurun waktu satu tahun mencapai enam jam. Selama periode ini, diperlukan perbaikan mesin yang membutuhkan waktu sehingga berpotensi menyebabkan target produksi tidak tercapai. Jika hal ini terjadi, diperlukan waktu kerja tambahan di luar jam kerja normal (*overtime)* untuk mengajar ketertinggalan produksi sehingga menimbulkan biaya yang cukup besar.

Selain itu, terdapat 19.590 data yang terdiri dari berbagai parameter data pada 10 mesin yang tercatat melebihi batas aman (*threshold*) dalam kurun

waktu satu bulan, hal ini menunjukkan tingginya indikasi mesin beroperasi dalam kondisi tidak optimal. Oleh sebab itu, pemeliharaan perlu segera dilakukan terhadap mesin yang terindikasi beroperasi dalam kondisi tidak optimal agar kerusakan mesin dapat dihindari.

PT ULS melakukan berbagai upaya dalam memantau kondisi mesin untuk mengurangi potensi terjadinya *downtime*. Salah satunya dengan mengukur parameter data mesin, seperti *Mean Time to Repair* (MTTR), *Mean Time Between Failure* (MTBF), dan karbondioksida (CO_2). Ketiga parameter tersebut merupakan *Key Performance Indicator* (KPI) pada Divisi *Engineering* yang berperan penting dalam menurunkan risiko terjadinya *downtime*. Parameter data MTTR dan MTBF memiliki pengaruh yang signifikan terhadap keandalan dan produktivitas sistem [1]. Sistem juga dapat menghasilkan emisi gas rumah kaca dan emisi tersebut dapat memburuk seiring dengan bertambahnya usia [2].

MTTR merupakan indikator yang digunakan untuk menghitung rata-rata waktu yang diperlukan dalam proses perbaikan mesin setelah mengalami kerusakan [3]. Semakin rendah nilai MTTR, semakin cepat mesin dapat kembali beroperasi. MTBF mengukur rata-rata waktu antara satu kegagalan mesin dengan kegagalan berikutnya, hal ini menunjukkan keandalan dan daya tahan mesin [3]. Semakin tinggi nilai MTBF menandakan mesin lebih jarang mengalami kerusakan. Sementara itu, CO_2 merupakan gas emisi yang terbentuk dari hasil pembakaran bahan bakar fosil dalam menghasilkan listrik, transportasi, dan proses industri [4]. Pada proses produksi PT ULS, terdapat dua jenis CO_2 yaitu CO_2 *release* dan *reduction*. CO_2 *release* adalah pelepasan gas CO_2 ke atmosfer [5]. Sedangkan CO_2 *reduction* didefinisikan sebagai pengurangan emisi CO_2 dari pembakaran bahan bakar fosil [6].

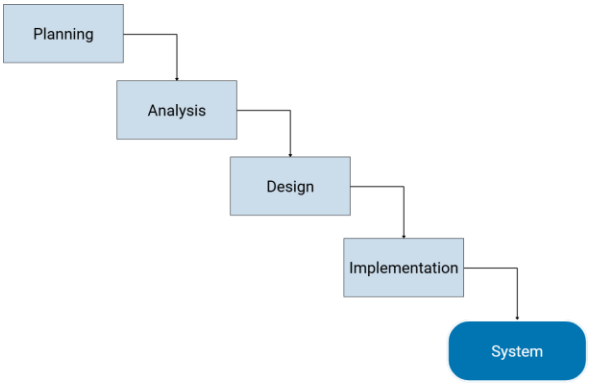
Penelitian Bristy et al. (2025) mengembangkan dasbor pemeliharaan prediktif berbasis IoT dengan indikator kesehatan aset melalui sistem "*health score*" [7]. Namun, penelitian tersebut belum memiliki fitur notifikasi yang dapat mengirimkan peringatan langsung kepada tim operasional sehingga respon terhadap potensi kerusakan masih bergantung pada pemantauan manual. Penelitian ini melengkapi kekurangan tersebut dengan mengembangkan Sistem Otomatisasi *Feedback* (SOF) yang mampu mengirimkan notifikasi perbaikan mesin secara melalui email dan Telegram.

Proses pengiriman notifikasi perbaikan mesin melalui *email* dan Telegram didasarkan oleh beberapa alasan. *Email* digunakan karena mampu mengirim dan menerima pesan dalam waktu singkat, bahkan secara *real-time* sehingga mendukung komunikasi yang cepat dan responsif [8]. Sedangkan, Telegram dipilih karena menghadirkan layanan *cloud* yang dapat mengirim pesan dengan lebih cepat dan menyediakan kapasitas penyimpanan yang besar [9].

Berdasarkan permasalahan tersebut, PT ULS berupaya mengembangkan SOF untuk mendukung proses pemantauan kondisi mesin. Sistem diharapkan dapat mengotomatisasi notifikasi berupa *feedback* terkait potensi kerusakan mesin beserta usulan perbaikannya. Tujuan penelitian ini adalah mengembangkan sistem yang mampu mengotomatisasi pengiriman notifikasi perbaikan mesin berdasarkan parameter MTTR, MTBF, dan CO_2 sesuai *threshold* yang ditetapkan sehingga pemantauan kondisi mesin dapat dilakukan secara lebih teratur.

II. METODOLOGI PENELITIAN

Metodologi merupakan kerangka kerja atau pendekatan sistematis yang digunakan untuk mengelola proses pengembangan perangkat lunak [10]. Pengembangan SOF menggunakan metodologi *waterfall* karena kebutuhan sistem sudah jelas sejak awal dan kemungkinan terjadinya perubahan selama proses pengembangan sangat kecil. Gambaran metodologi *waterfall* yang terapkan dapat diamati pada gambar 1.



Gambar 1. Metodologi *Waterfall* [10]

III. HASIL DAN PEMBAHASAN

Pengembangan SOF dilakukan sesuai dengan tahapan metodologi *waterfall*. Tahap pertama pada *planning* menghasilkan dokumen *system request* yang dapat diamati pada gambar 2 serta pada *activity plan* yang ditampilkan dalam gambar 3.

System Request – SOF		
***Project* Sponsor**	:	Divisi *Engineering* PT ULS.
Business Need	:	Sistem Otomatisasi *Feedback* Berbasis *Threshold* dibuat untuk mengotomatisasi pengiriman *feedback* apabila terdapat parameter data MTTR, MTBF, atau CO_2 yang melewati *threshold*, hal ini berfungsi untuk mendukung kegiatan pemantauan kondisi mesin pada Divisi *Engineering* yang sebelumnya dilakukan secara manual dan berulang sehingga menyebabkan berbagai kendala saat pemantauan, seperti keterlambatan dalam proses *maintenance* yang berpotensi mengakibatkan kerusakan mesin.
Business Requirement	:	Berikut merupakan *business requirement* untuk sistem ini: 1. Dapat mengelola *threshold warning* dan *alert* pada parameter data MTBF, MTTR, dan CO_2. 2. Dapat mengelola pesan *feedback* pada email maupun Telegram. 3. Dapat mengirim pesan *feedback* apabila parameter data melewati *threshold* melalui email maupun Telegram secara otomatis.
Business Value	:	Pengembangan sistem ini ditergetkan untuk membantu Divisi *Engineering* dalam melakukan pemantauan mesin secara otomatis melalui *feedback* yang dikirim melalui email dan Telegram sehingga mengurangi risiko keterlambatan dalam melakukan *maintenance* sehingga mengurangi risiko kerusakan mesin yang memerlukan banyak biaya.
Special Issues or Constraint	:	-

Gambar 2. *System Request*

No.	Aktivitas	Progress	Dec-24	Jan-25	Feb-25	Mar-25	Apr-25	May-25
1	Planning	100%						
2	Analysis	100%						
3	Design	100%						
4	Development	100%						
5	Testing	100%						
6	Deployment	100%						

No.	Warna	Deskripsi
1		Perencanaan
2		Aktual

Gambar 3. *Activity Plan*

Pada tahap analisis menghasilkan proses bisnis sebelum dan sesudah adanya sistem yang dapat diamati pada gambar 4 dan 5, selain itu terdapat dokumen *requirement definition* yang berisi kebutuhan fungsional dan non fungsional sistem, dokumen ini dapat dilihat pada gambar 6 dan 7. Berdasarkan analisis proses bisnis, diperlukan perbaikan dengan menggunakan *Business Process Automation* (BPA). BPA merupakan strategi yang memanfaatkan perangkat lunak dalam melakukan otomatisasi proses bisnis yang kompleks dan dilakukan secara berulang [11]. Melalui BPA, sistem dapat mengurangi langkah-langkah yang manual dan berulang untuk menjalankan proses bisnis.

Gambar 4. Proses Bisnis Sebelum

Gambar 5. Proses Bisnis Sesudah

Functional Requirement	
Kelola *Threshold* MTTR MTBF	Berikut merupakan *requirement* yang terdapat pada fungsi kelola *threshold* MTTR MTBF: 1. *Factory Industry Engineering* (FIE) dan *Industrial System Development* (ISD) dapat melihat *threshold* MTTR dan MTBF. 2. FIE dan ISD dapat melakukan *upload* data *threshold* MTBF dan MTTR. 3. FIE dan ISD dapat melakukan *export excel* data *threshold* MTBF dan MTTR.
Kelola *Threshold* CO_2	Berikut merupakan *requirement* yang terdapat pada fungsi kelola *threshold* CO_2: 1. *Engineering and Safety* (ENSA) dan ISD dapat melihat data *threshold* CO_2. 2. ENSA dan ISD dapat menambahkan *threshold* CO_2. 3. ENSA dan ISD dapat memperbaharui *threshold* CO_2. 4. ENSA dan ISD dapat menghapus *threshold* CO_2. 5. ENSA dan ISD dapat melakukan *export excel* data *threshold* CO_2.
Kelola Pesan *Feedback* Email dan Telegram	Berikut merupakan *requirement* yang terdapat pada fungsi kelola pesan *feedback* email dan Telegram: 1. FIE, ENSA, dan ISD dapat melihat pesan *feedback* email dan Telegram. 2. FIE, ENSA, dan ISD dapat menambahkan pesan *feedback* email dan Telegram. 3. FIE, ENSA, dan ISD dapat memperbaharui pesan *feedback* email dan Telegram. 4. FIE, ENSA, dan ISD dapat melakukan *export excel* pesan *feedback* email dan Telegram.
Pemantauan Kondisi Mesin	Berikut merupakan *requirement* yang terdapat pada fungsi pemantauan kondisi mesin: 1. Background job dapat mengirimkan *feedback* secara otomatis jika terdapat MTTR, MTBF, atau CO_2 yang melewati *threshold warning* atau *alert* pada jadwal yang ditentukan melalui email dan Telegram.

Gambar 6. *Functional Requirement*

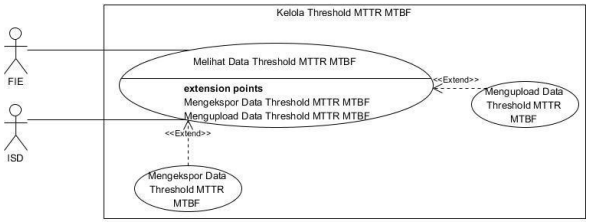
Non-Functional Requirement	
Operational	1. Sistem dapat beroperasi pada sistem operasi seperti windows. 2. Sistem dapat beroperasi di berbagai web browser (Mozilla Firefox, Google Chrome, Microsoft Edge) 3. Sistem dikembangkan menggunakan bahasa pemrograman Java dengan *framework* Spring Boot untuk Sistem Otomatisasi *Feedback*. 4. Sistem dikembangkan dengan menggunakan Oracle Database untuk Sistem Otomatisasi *Feedback*. 5. Sistem dikembangkan dengan *tools* Netbeans 8.2 untuk Sistem Otomatisasi *Feedback*.
Performance	Sistem dapat beroperasi selama 24 jam.
Security	1. Sistem hanya dapat beroperasi apabila *device* terhubung dengan jaringan internal PT ULS. 2. *User* hanya dapat mengakses sistem apabila memiliki hak akses *login* ke dalam sistem. 3. *User* hanya dapat mengakses fitur sesuai dengan *role* yang dimiliki.
Cultural and Political	-

Gambar 7. *Non-Functional Requirement*

Berdasarkan dokumen *requirement*, terdapat parameter data mesin berupa MTTR, MTBF, dan CO_2 yang diperoleh dari sistem internal PT ULS. Data tersebut dikumpulkan melalui sensor pada mesin, kemudian diolah dan disimpan dalam *database* perusahaan.

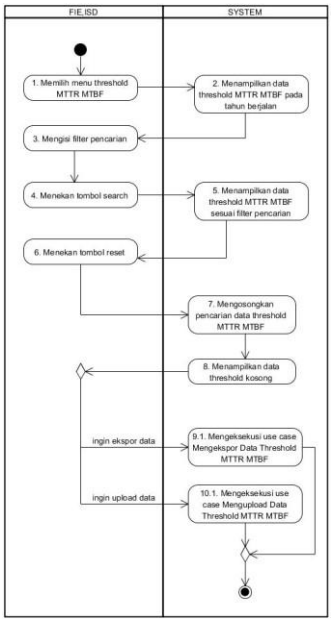
Selanjutnya akan dimodelkan melalui *use case diagram* untuk menggambarkan interaksi antara aktor dengan sistem. Pada SOF, ditemukan lima aktor yang berinteraksi dengan sistem, yaitu FIE, ENSA, ISD, *Engineering*, dan *Background Job*. *Background Job* atau *Cron Job* bertugas untuk menjalankan tugas terjadwal yang dapat mengotomatisasi alur kerja data di lingkungan *cloud* [12].

Selanjutnya, terdapat 13 *use case* yang dikelompokkan ke dalam empat bagian utama, yaitu kelola *threshold* MTTR MTBF, kelola *setting feedback*, kelola *threshold* CO_2, dan pemantauan kondisi mesin. Salah satu *use case diagram* yang dihasilkan dapat dilihat pada gambar 8, yaitu *use case diagram* kelola *threshold* MTTR MTBF.



Gambar 8. *Use Case Diagram* Kelola *Threshold MTTR MTBF*

Setiap *use case* akan dibuat menjadi *activity diagram* untuk menggambarkan detail aliran kerja dari *use case* tersebut. Pada SOF, terdapat 13 *activity diagram*, salah satunya dapat dilihat pada gambar 9. Pemodelan fungsional melalui pembuatan *use case diagram* dan *activity diagram* diselesaikan dengan penyusunan *use case description* untuk mendokumentasikan secara rinci proses yang terjadi pada setiap fungsi dalam suatu *use case*. Jumlah *use case description* yang dibuat sebanding dengan jumlah *activity diagram*, yaitu 13 *use case description* yang salah satu hasilnya dapat diamati pada gambar 10.



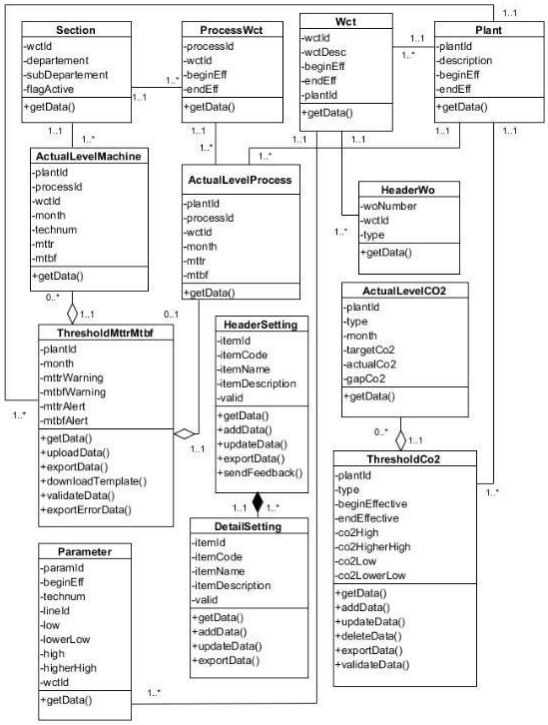
Gambar 9. *Activity Diagram* Melihat Data *Threshold* MTTR MTBF

Use Case Name : Melihat Data *Threshold* MTTR MTBF	**ID : SOFD01**	***Importance Level :*** *Low*
Primary Actor : FIE, ISD	***Use Case Type :*** *Detail, Essential*	
Stakeholder and Interest : FIE, ISD dapat melihat data *threshold* MTTR MTBF yang tersimpan		
Brief Description : *Use case* ini menjelaskan bagaimana FIE, ISD melihat data *threshold* MTTR MTBF yang tersimpan		
Trigger : FIE, ISD ingin melihat data *threshold* MTTR MTBF		
Relationships : *Association* : FIE, ISD *Include* : - *Extend* : Mengekspor Data *Threshold* MTTR MTBF, Mengupload Data *Threshold* MTTR MTBF *Generalization* : -		
Pre-Condition : *Use case* dimulai ketika FIE atau ISD ingin melihat data *threshold* MTTR MTBF		
Normal Flow of Events : 1. USER memilih menu *threshold* MTTR MTBF 2. SYSTEM menampilkan data *threshold* MTTR MTBF pada tahun berjalan 3. USER mengisi *filter* pencarian 4. USER menekan tombol *search* 5. SYSTEM menampilkan data *threshold* MTTR MTBF sesuai *filter* pencarian 6. USER menekan tombol *reset* 7. SYSTEM mengosongkan pencarian data *threshold* MTTR MTBF 8. SYSTEM menampilkan data *threshold* kosong 9. IF USER ingin ekspor data 9.1. SYSTEM mengeksekusi use case Mengekspor Data *Threshold* MTTR MTBF END IF 10. USER ingin upload data 10.1. SYSTEM mengeksekusi use case Mengupload Data *Threshold* MTTR MTBF END IF		
Sub Flow : -		
Alternate/Exceptional Flows : -		
Post-Condition : *Use case* berakhir ketika FIE atau ISD berhasil melihat data *threshold* MTTR MTBF		

Gambar 10. *Use Case Description* Melihat Data *Threshold* MTTR MTBF

Setelah itu, pemodelan dilanjutkan dengan pembuatan *class diagram* yang merupakan model statis untuk menunjukkan hubungan antara kelas, atribut, dan operasi dari setiap kelas. Pembuatan *class*

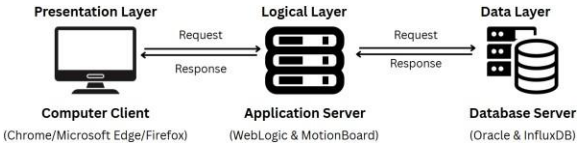
diagram diawali dengan menganalisis elemen seperti nama kelas, atribut, dan operasi, yang ditentukan berdasarkan *requirement definition*. Pada SOF, dihasilkan 14 kelas dan 31 operasi yang dapat diamati pada gambar 11.



Gambar 11. *Class Diagram*

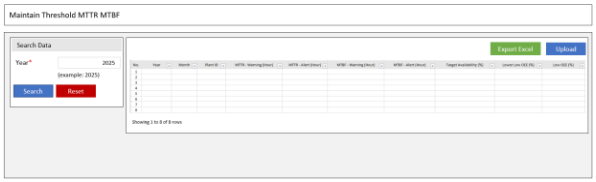
Setelah seluruh pemodelan fungsional dan struktural dilakukan, tahap analisis telah selesai dilakukan. Selanjutnya dilanjutkan dengan tahap desain meliputi desain arsitektur, desain antarmuka, dan dekripsi data.

Desain arsitektur menggambarkan perangkat keras (*hardware*) dan perangkat lunak (*software*) sehingga dapat menghasilkan sistem yang dibutuhkan. Desain arsitektur yang digunakan oleh SOF adalah *three-tiered architecture*, arsitektur ini membagi sistem menjadi tiga lapisan yang saling berhubungan, yaitu *presentation layer*, *logical layer*, dan *data layer*. Desain arsitektur dari pengembangan sistem terdapat pada gambar 12.



Gambar 12. Desain Arsitektur

Pada tahap desain juga terdapat pembuatan desain antarmuka dari aplikasi yang dibuat. Desain antarmuka merupakan perpaduan antara informasi, konsep desain interaksi, serta desain visual yang dirancang untuk memberikan kemudahan dan kenyamanan bagi *user* [13]. Salah satu desain antarmuka SOF dapat dilihat pada gambar 13, yaitu desain antarmuka *view threshold MTTR MTBF*.



Gambar 13. Desain Antarmuka *View Threshold MTTR MTBF*

Selanjutnya perancangan basis data melalui *Physical Data Model* (PDM) sebagai bentuk pendeskripsian data yang diperoleh dari pemodelan *class diagram*. Perancangan basis data berperan penting dalam menentukan struktur informasi yang dibutuhkan secara rinci sebelum sistem dikembangkan sehingga mendukung kinerja sistem secara keseluruhan [14]. PDM digunakan untuk menggambarkan secara detail struktur tabel, seperti nama tabel, kolom tabel, tipe data, panjang data, dan keterbatasan pada data. PDM yang diterapkan pada SOF dapat diamati pada gambar 14.

Gambar 14. *Physical Data Model* (PDM)

Selanjutnya pengembangan sistem dilakukan melalui penulisan kode. Bahasa pemrograman yang

digunakan adalah Java dengan *framework* Java Spring, *framework* ini memiliki 11 direktori, yaitu:

1. Model berfungsi untuk merepresentasikan struktur tabel dalam basis data.
2. DAO berfungsi sebagai antarmuka yang mendefinisikan fungsi-fungsi akses data yang akan diimplementasikan dalam DAO Implementation.
3. DAO implementation merupakan implementasi dari DAO yang berfungsi untuk mengakses dan mengelola data dalam basis data.
4. *Service* berfungsi sebagai antarmuka yang mendefinisikan fungsi-fungsi yang akan diimplementasikan dalam *service* implementation, pada SOF terdapat tiga *service* yang digunakan.
5. *Service implementation* merupakan implementasi dari *service* yang berfungsi sebagai logika bisnis aplikasi.
6. REST berfungsi untuk menghasilkan URL yang dapat diakses oleh user. URL ini memungkinkan user untuk mengakses dan menampilkan output dari fungsi yang tersedia dalam *service*.
7. REST *Advice* yang berisi format *response* umum untuk digunakan oleh berbagai fungsi.
8. Constant berisi query yang diperlukan untuk mengakses data pada database.
9. VO merupakan kelas yang memungkinkan untuk digunakan sebagai penyimpanan data sementara.
10. Job berisi fungsi yang mengatur scheduler untuk menjalankan background job, pada SOF terdapat satu job yang digunakan.
11. View adalah tampilan visual yang memungkinkan user berinteraksi dengan sistem. Pada bagian ini, user dapat mengirim permintaan dan menerima respon dari REST. Tampilan ini dikembangkan menggunakan HTML dan JavaScript.

Selain itu Oracle dipilih sebagai sistem penyimpanan data pada sistem. Oracle merupakan produk *Relational Database Management System* (RDBMS) yang dikembangkan oleh Oracle Corporation. Sistem ini dirancang untuk meningkatkan waktu eksekusi *query* dan operasi basis data menggunakan teknik optimasi kinerja [15].

Setelah pengembangan sistem, tahap berikutnya adalah penguijan sistem yang dilakukan pada SOF mencakup *unit test*, *System Integration Test* (SIT), *User Acceptance Test* (UAT), dan *Quality Assurance Test* (QAT). Seluruh tahapan pengujian ini wajib dilalui guna memastikan standar kualitas pengujian yang tinggi sebagaimana ditetapkan oleh PT ULS. Hasil pengujian menunjukan SOF dapat mengirimkan notifikasi *feedback* perbaikan mesin sesuai dengan jadwal yang ditentukan sehingga proses pemantauan kondisi mesin dapat dilakukan dengan lebih teratur. Tabel 1 menunjukkan hasil pengujian yang telah dilakukan.

Tabel 1. Parameter Pengujian

Parameter Data	**Jumlah Uji**	**Jumlah Notifikasi Terkirim**	**Persentase (%)**
MTTR	12	12	100
MTBF	36	36	100
CO_2	54	54	100

Setelah pengujian, tahap selanjutnya ialah proses *deployment* yang bertujuan untuk menerapkan suatu sistem ke dalam lingkungan operasional [10]. Proses ini dilakukan agar sistem yang telah dikembangkan dapat digunakan oleh *user* di server operasional. Adapun rata- rata nilai *threshold* untuk parameter data MTTR, MTBF, dan CO_2 yang digunakan pada server operasional ditunjukkan pada tabel 2 serta tabel 3.

Tabel 2. *Threshold* MTTR dan MTBF

Parameter Data	***Threshold***	
	Low	***High***
MTTR	1.35	1.37
MTBF	206.9	204.8

Tabel 3. *Threshold* CO_2

Parameter Data	***Threshold***			
	Higher High	***High***	***Low***	***Lower Low***
CO2	0.06	0.03	-0.03	-0.06

IV. KESIMPULAN

Kesimpulan yang dapat diambil berdasarkan pengembangan SOF adalah sistem ini telah dikembangkan dan dilakukan *deployment* ke server operasional, sistem ini dapat mengirimkan notifikasi *feedback* perbaikan mesin sesuai dengan jadwal yang ditentukan sehingga proses pemantauan kondisi mesin dapat dilakukan secara lebih teratur. Namun, ruang lingkup implementasi masih terbatas pada

parameter tertentu seperti MTTR, MTBF, dan CO_2. Pengembangan berikutnya disarankan memperluas ruang lingkup otomatisasi ke parameter lain, seperti *temperature control*, *power consumption*, dan lainnya agar kondisi mesin dapat termonitor dengan lebih lengkap.

V. DAFTAR PUSTAKA

[1] S. Alfionita and F. I. Alifin, "Preventive Maintenance Analysis Based on Mean Time Between Failure (MTBF) and Mean Time to Repair (MTTR)," *Angkasa J. Ilm. Bid. Teknol.*, vol. 15, no. 2, p. 201, 2023, doi: 10.28989/angkasa.v15i2.1833.

[2] S. Wu, D. Wu, and R. Peng, "Considering greenhouse gas emissions in maintenance optimisation," *Eur. J. Oper. Res.*, vol. 307, no. 3, pp. 1135–1145, 2023, doi: 10.1016/j.ejor.2022.10.007.

[3] "MTBF, MTTR, dan MTTF – Indikator Penting untuk Maintenance." Accessed: Apr. 16, 2025. [Online]. Available: https://amtiss.com/blog/2020/01/21/mtbf-mttr-dan-mttf-indikator-penting-untuk-maintenance/

[4] "What are the sources of carbon dioxide in the atmosphere?" Accessed: Apr. 16, 2025. [Online]. Available: https://www.csiro.au/en/research/environmental-impacts/climate-change/Climate-change-QA/Sources-of CO2#:~:text=Aktivitas manusia menyebabkan karbondi oksida,transportasi%2C dan proses industri).

[5] Y. Widawati and M. Anas, "Determinan Emisi Karbon Dioksida di Indonesia: Analisis Partial Adjustmen," *Univ. Muhammadiyah Surakarta*, vol. 1, no. 1, pp. 1–15, 2024, [Online]. Available: https://eprints.ums.ac.id/122222

[6] "Strategic analysis of diffusion of renewable energy in the Nordic countries." Accessed: Apr. 23, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1364032113001081

[7] I. J. Bristy, M. Tabassum, M. I. Islam, and M. N. Hasan, "IoT-Driven Predictive Maintenance Dashboards in Industrial Operations," *Saudi J. Eng. Technol.*, vol. 10, no. 09, pp. 457–466, 2025, doi: 10.36348/sjet.2025.v10i09.009.

[8] M. Afifah, A. Amaluddin, and R. Soraya, "Pemanfaatan Surat Menyurat Elektronik Dalam Meningkatkan Efektivitas Komunikasi Organisasi," *J. Dialect*, vol. 1, no. 2, pp. 41–49, 2024, doi: 10.46576/dl.v1i2.4630.

[9] H. A. Kusuma, S. B. Wijaya, and D. Nusyirwan, "Sistem Keamanan Rumah Berbasis Esp32- Cam Dan Telegram Sebagai Notifikasi," *Infotronik J. Teknol. Inf. dan Elektron.*, vol. 8, no. 1, p. 30, 2023, doi: 10.32897/infotronik.2023.8.1.2291.

[10] D. T. Barbara Wixom, *Systems Analysis and Design An Object-Oriented Approach with UML Alan Dennis*, vol. 21, no. 1. 2020.

[11] "Apa itu otomatisasi proses bisnis (BPA)?" Accessed: Apr. 20, 2025. [Online]. Available: https://www.ibm.com/id-id/topics/business-process-automation#:~:text=Otomatisasi proses bisnis (BPA) adalah,bisnis tetap berfungsi dengan lancar

[12] A. Wibowo, A. Fathimah, D. Waskito, and R.A. Aprilianto, "Cron Job Implementation for Automated Data Processing and Transfer in Cloud Infrastructure," *Sainteknol J. Sains dan Teknol.*, vol. 23, no. 1, pp. 11–20, 2025, doi: 10.15294/sainteknol.v23i1.22649.

[13] "Apa itu User Interface Design." Accessed: Apr. 16, 2025. [Online]. Available: https://binus.ac.id/binus-digital/2020/01/22/apa-itu-user-interface-design/#:~:text=Desain User Interface (UI) adalah,mudah untuk digunakan dan menyenangkan.

[14] M. Ridha, M. Affandes, E. P. Cynthia, and Pizaini, "Implementasi Treemap untuk Visualisasi Data Angka Kesakitan (Morbiditas) (Studi Kasus: Dinas Kesehatan Indragiri Hilir)," J. Nas. Komputasi dan Teknol. Inf., vol. 5, no. 2, pp. 166–174, 2022, [Online]. Available: https://jnkti.uin-suska.ac.id

[15] S. T. Nurhayati, M. Irwan, and P. Nasution, "Database Management System Pada Perusahaan," *J. Akunt. Keuang. dan Bisnis*, vol. 1, no. 2, pp. 62–64, 2023, [Online]. Available: https://jurnal.ittc.web.id/index.php/jakbs/index