

Pengembangan Sistem Monitor dan Laporan Mesin PT. Numalos Abadi Menggunakan *Amazon Web Services*

Erico¹, Stephanie Wijaya², Bryan Herberth Tambela³, Irpan Adiputra Pardosi⁴, Sunaryo Winardi⁵

^{1,2,3,4,5}Universitas Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4567789

^{1,2,3,4,5}Fakultas Informatika, Program Studi Teknik Informatika, Universitas Mikroskil, Medan

¹181110301@students.mikroskil.ac.id, ²181111471@students.mikroskil.ac.id,

³181112442@students.mikroskil.ac.id, ⁴irpan@mikroskil.ac.id, ⁵sunaryo.winardi@mikroskil.ac.id

Abstrak

PT. Numalos Abadi merupakan perusahaan distributor dan layanan purna jual mesin, salah satunya adalah *decanter* Flottweg Z5E. PT. Numalos Abadi memerlukan penerapan *Internet of Things* untuk mendapatkan informasi mesin, salah satu informasi mesin yang menjadi fokus utama adalah *running hour* yang merupakan penjumlahan durasi mesin berjalan. *Running hour* yang tidak terpantau mengakibatkan penurunan kinerja mesin yang berdampak pada hasil produksi maupun kerusakan mesin. Tidak adanya pemberitahuan dini waktu perawatan mesin mengakibatkan tidak adanya persiapan materi maupun waktu yang merugikan bagi *customer*. Berdasarkan masalah tersebut, dikembangkanlah sebuah solusi berupa sistem *monitoring* dan laporan mesin untuk mengelola waktu perawatan mesin dan mengurangi potensi kerugian bagi *customer*. Pengembangan sistem monitor dan laporan mesin diimplementasikan menggunakan *Amazon Web Services* dalam aplikasi *web* dan *mobile*. Dalam penelitian ini dikembangkan aplikasi *web* dan *mobile* menggunakan metodologi *Waterfall* untuk menghasilkan kualitas sistem yang baik dan dokumen pengembangan sistem yang terorganisasi. Pengujian dilakukan secara simulasi dan *running hour* yang telah melewati batas akan menampilkan indikasi alarm serta notifikasi. Pengujian *Black Box* yang dilakukan juga menunjukkan secara fungsional aplikasi dapat bekerja dengan baik dan mengeluarkan hasil yang diharapkan.

Kata kunci: *internet of things (IoT), Amazon Web Services (AWS), running hour mesin, perawatan mesin, decanter Flottweg Z5*

Abstract

PT Numalos Abadi is a distributor and after-sales service company for machinery, one of which is the Flottweg Z5E decanter. PT. Numalos Abadi required using the Internet of Things to obtain information on the machines it distributes. One of the main focuses is the running hour of a machine which is the number of hours a machine is in operational mode, if not monitored could cause a decrease in machine performance which impacts the production results and machine damage. Due to the absence of a notification Customers often requests sudden maintenance without any preparation. Based on this problem, a solution was developed in the form of a machine Monitoring and Reporting system to manage maintenance time and reduce potential losses. The development of the system is implemented using Amazon Web Services in web and mobile applications. Waterfall methodology is used in this research to produce good system quality and organized system development documents. Testing is done by simulation and running hours that have exceeded the limit will display alarm indications and notifications. Black Box testing shows that functionally the application works and produces the expected results.

Keywords: *internet of things (IoT), Amazon Web Services (AWS), running hour machine, service machine, decanter Flottweg Z5E*

1. PENDAHULUAN

PT. Numalos Abadi merupakan sebuah perusahaan distributor dan layanan purna jual (*after-sales service*) *engineering* khusus yang mewakili sejumlah merek terkemuka di Indonesia, salah satunya adalah Flottweg SE (Jerman). PT. Numalos Abadi sebagai penyedia kebutuhan *engineering* khusus ingin meningkatkan pelayanan pada mesin yang didistribusikannya, salah satunya adalah mesin *decanter* Flottweg Z5E. Peningkatan yang diharapkan ialah kemudahan dalam mengakses informasi mesin secara cepat yang dapat dilakukan di mana pun, salah satunya adalah informasi *running hour*. Informasi *running hour* ini diperlukan untuk menjaga kinerja mesin dan jika *running hour* telah mencapai waktu yang ditentukan maka perlu dilakukan perawatan (*service*). *Service* ini sangat dibutuhkan karena mesin yang telah bekerja melewati *running hour* dapat menurunkan kinerja mesin yang berdampak pada hasil produksi mesin yang kurang maksimal. Oleh karena itu, sebelum *running hour* mesin mencapai waktu yang ditentukan, diperlukannya peringatan atau notifikasi terlebih dahulu agar pihak *customer* dapat mempersiapkan waktu maupun biaya untuk melakukan perawatan mesin tersebut.

Solusi yang dapat digunakan yaitu dengan memanfaatkan *Internet of Things (IoT)* agar informasi mesin yang didistribusikan oleh PT. Numalos Abadi dapat diakses di mana pun dan kapan pun. *Internet of Things (IoT)* merupakan sebuah konsep yang memungkinkan sebuah objek yang terhubung ke internet memiliki kemampuan untuk mengirimkan data melalui jaringan tanpa memerlukan bantuan baik dari perangkat komputer maupun manusia [1]. Dalam penerapannya pada mesin *decanter* Flottweg Z5E, *Amazon Web Services (AWS)* dimanfaatkan sebagai *server*, *hosting*, dan layanan IoT. *Amazon Web Service* merupakan layanan *cloud computing* dari amazon.com yang setiap fungsi di dalamnya dapat diakses dengan panggilan *web service* [2]. Salah satu layanan dari *Amazon Web Service* ialah *AWS IoT Core* yang merupakan *platform* yang memungkinkan pengguna menghubungkan perangkat IoT ke *cloud* AWS tanpa perlu menyediakan atau mengelola *server*. Pada mesin *decanter* Flottweg Z5E terdapat *controller* yang biasa disebut *Programmable Logic Controller (PLC)* dengan tipe Modicon M221 yang merupakan produk merek Schneider Electric. Untuk menghubungkan *controller* tersebut dengan aplikasi *mobile* dan *web* menggunakan *protocol Message Queue Telemetry Transport (MQTT)* diperlukan sebuah alat sebagai *gateway* antar kedua *platform* tersebut yaitu *Raspberry Pi* dan *middleware* *AWS IoT Core* sebagai *MQTT Broker*.

Namun dalam penelitian ini terdapat beberapa ruang lingkup masalah yang ditentukan, dimulai dari jumlah aktor dalam sistem ini ada 3 yaitu Pemilik, Admin dan Anggota. Dalam pengembangan ini, aplikasi *web* berfungsi untuk dapat melakukan *monitoring* dan membaca laporan dari mesin serta menambahkan pabrik atau anggota baru, sedangkan aplikasi *mobile* berfungsi sebagai *view only client* yaitu hanya dapat melakukan *monitoring* dan membaca laporan dari mesin yang ada di sebuah pabrik, dan layanan AWS yang digunakan merupakan akun *free trial* yang memiliki batas waktu sebanyak 12 bulan, serta dalam mengakses aplikasi diperlukan koneksi internet yang memadai dan pengujian dilakukan dengan *controller (PLC)* mesin yang disimulasikan.

2. TINJAUAN PUSTAKA

2.1 Mesin Decanter

Mesin adalah perkakas untuk menggerakkan atau membuat sesuatu yang dijalankan dengan roda, digerakkan oleh tenaga manusia atau motor penggerak, menggunakan bahan bakar minyak atau tenaga alam. Mesin adalah alat mekanik atau elektrik yang mengirim atau mengubah energi untuk melakukan atau membantu pelaksanaan tugas manusia [3].

Decanter adalah mesin yang digunakan untuk memisahkan kembali minyak yang tersisa dari cairan *sludge*. Cara kerja dari mesin *decanter* adalah *sludge* yang berat jenisnya lebih besar akan terlempar keluar melalui *nozzle* yang diakibatkan gaya sentrifugal sedangkan minyak yang berat jenisnya lebih kecil akan terkumpul dan akan keluar menuju *recycle oil tank*. *Decanter* merupakan mesin yang digunakan untuk memisahkan *light phase*, *heavy phase* dan padatan (*solid*) [4]. Mesin *decanter*

yang digunakan dalam pengembangan ini adalah mesin *decanter* Flottweg Z5E yang dapat dilihat pada Gambar 1.



Gambar 1. Mesin *decanter* Flottweg Z5E [4]

2.2 *Internet of Things (IoT)*

Internet of Things (IoT) merupakan sebuah konsep yang memungkinkan sebuah objek yang terhubung ke internet memiliki kemampuan untuk mengirimkan data melalui jaringan tanpa memerlukan bantuan baik dari perangkat komputer maupun manusia. *IoT* diekspektasikan dapat mengubah bagaimana dunia berfungsi dengan memungkinkan untuk memonitor dan mengontrol kejadian – kejadian penting yang berhubungan dengan lingkungan sekitar menggunakan bantuan dari objek atau alat. Objek atau alat tersebut sudah memiliki sensor yang mampu untuk mengambil, memproses, dan mengirimkan data secara *wireless* ke dalam sebuah *cloud-based services* yang dapat menganalisis dan mengartikan data yang dikirim merupakan sebuah data yang penting atau berarti [1].

2.3 *Amazon Web Services (AWS)*

Amazon Web Services merupakan layanan *cloud computing* yang disediakan dari Amazon.com yang setiap fungsinya dapat diakses dengan panggilan *web service*. Pada *Amazon Web Service* tersedia lebih dari 200 layanan [2]. Beberapa Layanan Amazon Web Services yang kami gunakan adalah AWS *IoT Core*, AWS *Lambda*, AWS *Relational Database Services (RDS)*, AWS *Amplify*, AWS *Elastic Beanstalk*, dan AWS *Simple Storage Service (S3)*.

AWS *IoT Core* merupakan *platform* yang memungkinkan pengguna untuk menghubungkan perangkat IoT ke *cloud AWS* tanpa menyediakan dan mengelola *server* [5]. AWS *Lambda* merupakan layanan komputasi *serverless* yang disediakan oleh Amazon [6]. Amazon RDS merupakan layanan basis data relasional yang memudahkan untuk menyiapkan, mengoperasikan, dan menskalakan basis data relasional di *cloud* [7]. AWS *Amplify* merupakan alat yang dibuat khusus bagi *developer frontend web* dan seluler untuk mengonfigurasi *backend web* atau aplikasi seluler, menghubungkan aplikasi dalam beberapa menit dan juga membangun *UI Frontend web* secara visual dengan mudah [8]. AWS *Elastic Beanstalk* merupakan layanan untuk menerapkan dan menyesuaikan skala aplikasi mulai dari penyediaan kapasitas, penyeimbangan beban, *auto-scaling* hingga pemantauan kesehatan aplikasi [9]. AWS S3 merupakan layanan penyimpanan data dalam bentuk objek yang juga dapat melindungi data [10].

2.4 Monitor

Monitor adalah alat yang dirancang untuk mengobservasi, mengawasi, mengontrol, atau memverifikasi operasi suatu sistem. Monitor juga berarti pengawasan dan tindakan memverifikasi kebenaran operasi suatu program selama pelaksanaannya berdasarkan rutin diagnostik yang digunakan dari waktu ke waktu untuk menjawab pertanyaan tentang program tersebut.

Kegiatan *monitoring* dimaksudkan untuk mengetahui kecocokan dan ketepatan kegiatan yang dilaksanakan dengan rencana yang telah disusun. *Monitoring* digunakan pula untuk memperbaiki kegiatan yang menyimpang dari rencana, mengoreksi penyalahgunaan aturan dan sumber – sumber, serta untuk mengupayakan agar tujuan dicapai seefektif dan seefisien mungkin [11].

2.5 Simulasi

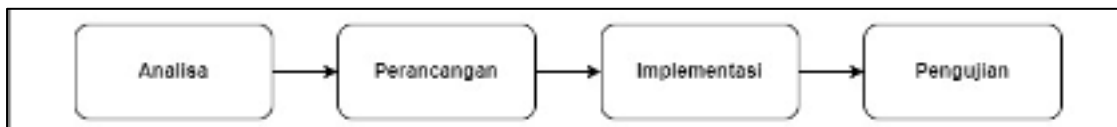
Simulasi merupakan sebuah proses peniruan dari sesuatu yang nyata beserta keadaan sekelilingnya (*state of affairs*). Dalam simulasi dibutuhkan komputer untuk dapat mempelajari sistem

dan digunakan untuk pengumpulan data agar estimasi statistik dapat dilakukan untuk dapat mengetahui bagaimana karakteristik asli dari sistem tersebut [12].

Adapun tahapan dalam melakukan sebuah simulasi yaitu dimulai dengan mendefinisikan suatu masalah yang akan menjadi tujuan untuk disimulasikan beserta variabel penting yang berkaitan dengan masalah tersebut, setelah itu perlu dikembangkan sebuah kejadian yang mungkin terjadi pada saat dilakukan pengujian dengan nilai variabel tertentu, setelah semua data yang diperlukan sudah tersedia dilakukan percobaan dan mempertimbangkan hasil yang didapatkan dari percobaan tersebut dan mengambil keputusan atau kesimpulan dari hasil tersebut [12].

3. METODE PENELITIAN

Adapun kerangka kerja dalam pengembangan ini sebanyak 4 tahapan yang digambarkan sebagai berikut:

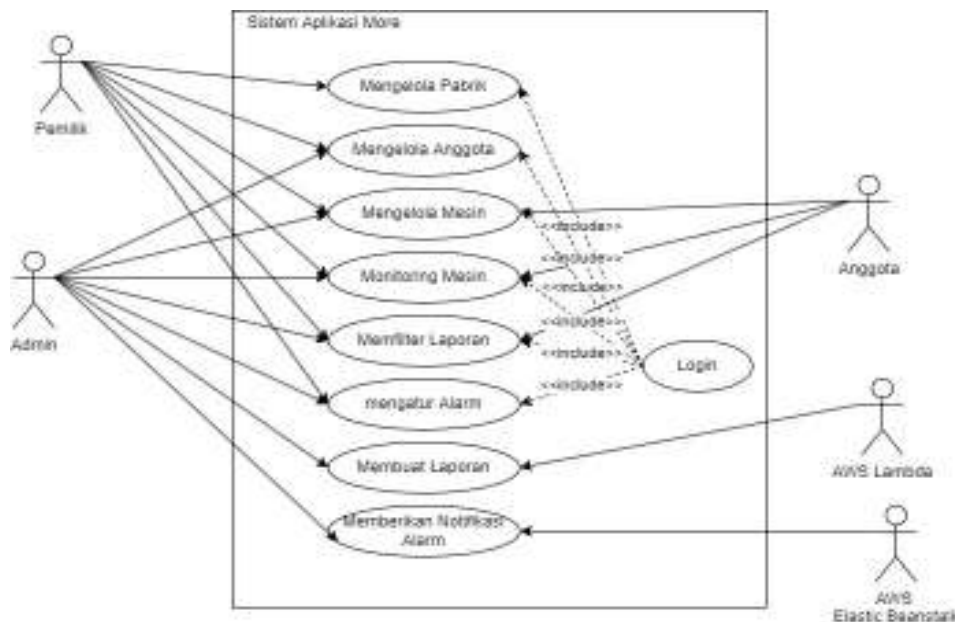


Gambar 2. Kerangka Kerja

Berdasarkan kerangka kerja pada Gambar 2, maka masing-masing tahapan dapat dijelaskan sebagai berikut:

3.1 Analisis

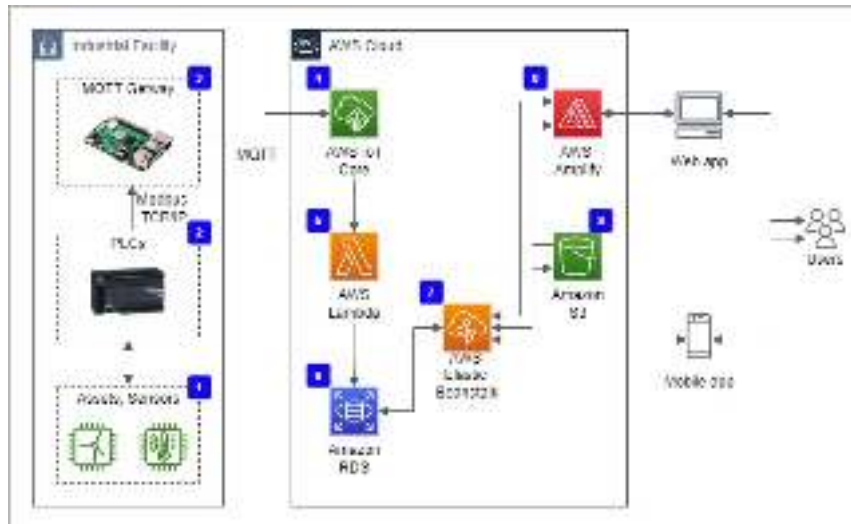
Analisis terdiri dari 2 bagian yaitu analisis kebutuhan fungsional dan non fungsional. Kebutuhan fungsional dimodelkan menggunakan *use case diagram* dan kebutuhan non fungsional dimodelkan menggunakan pemodelan PIECES. Analisis kebutuhan fungsional dimodelkan menggunakan diagram *use case*. Diagram ini mencakup fitur utama dari sistem dan digunakan untuk 5 (lima) aktor, yaitu Pemilik, Admin, Anggota, AWS Lambda dan AWS Elastic Beanstalk. Diagram *use case* dapat dilihat pada Gambar 3.



Gambar 3. Diagram *use case* Aplikasi MoRe

3.2 Perancangan

Perancangan aplikasi menggunakan Amazon Web Services yang merupakan penyedia layanan *cloud computing* yang menyediakan lebih dari 200 layanan. Pada perancangan aplikasi MoRe hanya menggunakan beberapa layanan yaitu *AWS IoT Core*, *AWS Lambda*, *Amazon RDS*, *AWS Elastic Beanstalk*, *AWS Amplify*, *AWS S3*. Rancangan arsitektur yang di buat pada AWS merupakan sebuah ekosistem *server* yang memiliki beberapa tugas seperti membaca data dari perangkat keras dengan *protocol MQTT*, *database server*, *API server*, dan *web hosting*. Perancangan arsitektur aplikasi pada Amazon Web Services (AWS) dapat dilihat pada Gambar 4.



Gambar 4. Rancangan aplikasi AWS

Pada rancangan AWS aplikasi MoRe terbagi menjadi 2 bagian yaitu arsitektur pada *Industrial Facility* (pabrik) dan arsitektur pada *AWS Cloud*. Pada *Industrial Facility assets, sensor* (1) merupakan perangkat keras yang terpasang pada mesin, berfungsi untuk mengumpulkan informasi mesin. PLCs Modicon M221 (2) merupakan *controller* mesin yang terpasang pada panel kontrol mesin berfungsi sebagai penerima informasi *Assets, Sensor* dan untuk menjalankan mesin. MQTT Gateway (3) merupakan *Raspberry Pi 4 model B* yang di program menggunakan *Node-Red*. *Raspberry Pi* membaca informasi dari PLC (2) melalui jaringan LAN menggunakan *protocol Modbus TCP/IP* dan mengirimnya ke AWS melalui internet menggunakan *protocol MQTT*.

Pada *AWS Cloud*, *AWS IoT Core* (4) merupakan layanan AWS yang berfungsi sebagai *server* MQTT yang menerima *Publish (Subscribe)* data dari MQTT Gateway (3). Setelah *AWS IoT Core* *unsubscribe* dari MQTT Gateway, *AWS Lambda* (5) akan *trigger* untuk membaca dan mengelola data MQTT dan menyimpannya ke *database Amazon RDS* (6). *Amazon RDS* (6) menggunakan *database* relasional PostgreSQL untuk menyimpan data. *AWS Elastic Beanstalk* (7) sebagai *server REST API* dengan *protocol HTTP* yang dapat diakses melalui domain yang disediakan oleh AWS. *AWS Amplify* (8) berfungsi sebagai *client* untuk *deploy* aplikasi *Front-End Web* dan dapat diakses melalui domain yang disediakan oleh AWS. *Amazon S3* (9) layanan penyimpanan data *object* yang diakses oleh *AWS Elastic Beanstalk* (7) dan *AWS Amplify* dengan akses CRUD.

3.3 Implementasi

Implementasi dibagi menjadi 3 bagian yaitu implementasi perangkat, implementasi aplikasi di AWS dan implementasi *user interface*. Implementasi perangkat menggunakan PLC Modicon M221 yang di program sedemikian rupa agar dapat menstimulasikan aktivitas mesin. MQTT Gateway menggunakan *Raspberry Pi* dengan *tool node-red* untuk mengirim data dari PLC ke AWS. Infrastruktur aplikasi pada AWS diimplementasikan sesuai dengan perancangan aplikasi AWS. Implementasi aplikasi *mobile* dan *web* sebagai *user interface* aplikasi.

3.4 Pengujian

Pengujian perangkat dilakukan dengan melakukan simulasi pada PLC mesin yang telah di program berdasarkan data yang diperoleh dari PT. Numalos Abadi. PLC dihubungkan pada *Raspberry Pi* sebagai *MQTT Gateway* agar dapat mengirimkan data ke AWS. Setelah data diterima oleh AWS, data diproses dan ditampilkan pada aplikasi *mobile* dan *web*.

4. HASIL DAN PEMBAHASAN

Hasil dari aplikasi terbagi menjadi 2, yaitu *web* dan *mobile*. Simulasi pengujian dilakukan dengan membangun perangkat sederhana yang mewakili keadaan mesin sebenarnya (Gambar 5). Saat tombol *AUTO ON* (tombol merah) ditekan, maka PLC yang telah disimulasikan mengirimkan data ke *Raspberry Pi* sebagai *Gateway MQTT*. Data dari *Raspberry Pi* diterima oleh *AWS IoT Core* dan diproses. Ada 4 data yang diproses oleh AWS, yaitu *status*, *speed*, *flow inlet*, dan *running hour*. Data *speed* diperoleh dari *inverter* dan *flow inlet* diperoleh dari *flow meter*. Proses penjumlahan *running hour* dimulai saat status mesin *START UP* hingga *RUN*. Saat tombol *AUTO OFF* (tombol hijau) ditekan, maka status berubah menjadi *COOLDOWN* hingga *STOP* dan proses penjumlahan *running hour* berhenti.



Gambar 5. Alat - Alat Yang Digunakan Pada Pengujian Simulasi

Gambar 5 merupakan perangkat sederhana yang dibangun untuk melakukan simulasi. Nomor (1) merupakan tombol *AUTO ON*. Nomor (2) merupakan tombol *AUTO OFF*. Nomor (3) merupakan kabel LAN untuk menghubungkan PLC ke *Raspberry Pi*. Nomor (4) merupakan PLC Modicon M221. Nomor (5) merupakan *Raspberry Pi 4 Model B*.



Gambar 6. *Inverter* mesin

Gambar 6 merupakan *inverter* mesin yang digunakan untuk mengatur kecepatan mesin. Pada pengujian simulasi, kecepatan dari *inverter* disimulasikan berdasarkan data yang diperoleh dari PT. Numalos Abadi.



Gambar 7. *Flow meter* mesin

Gambar 7 merupakan *flow meter* mesin yang digunakan untuk mengukur jumlah aliran masuk. Pada pengujian simulasi ini, jumlah *flow inlet* dari *flow meter* disimulasikan berdasarkan data yang diperoleh dari PT. Numalos Abadi.



Gambar 8. Tab pemantauan mesin *web* dan *mobile*

Gambar 8 merupakan data – data yang telah diproses oleh AWS dan ditampilkan pada tab pemantauan mesin. Data yang ditampilkan antara lain status mesin (*Start-Up*, *Run*, *Cooldown*, dan *Stop*), *speed* mesin, *flow inlet* mesin, dan *running hour* mesin. Data – data yang ditampilkan akan diperbaharui secara otomatis setiap 2 detik.



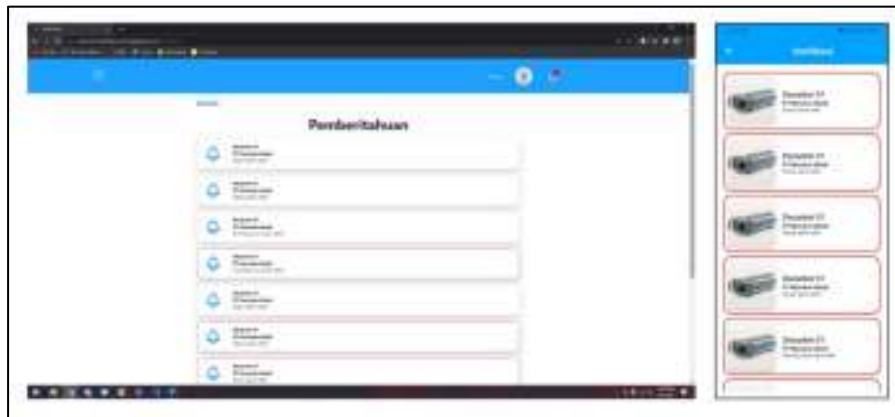
Gambar 9. Tab laporan mesin *web* dan *mobile* (tabel)

Gambar 9 merupakan tampilan tab laporan mesin. Data – data laporan mesin ditampilkan menggunakan tabel. Untuk menampilkan data laporan mesin dapat dilakukan dengan cara memfilter

berdasarkan variabel (*status*, *speed*, *flow inlet*, dan *running hour*) dan tanggal (tanggal mulai dan tanggal berakhir). Selain visualisasi laporan dalam bentuk tabel, pengguna juga dapat menampilkan laporan dalam bentuk grafik yang dapat dilihat pada Gambar 10.



Gambar 10. Tab laporan mesin *web* dan *mobile* (grafik)



Gambar 11. Visualisasi alarm pada halaman notifikasi *web* dan *mobile*

Gambar 11 merupakan visualisasi alarm pada halaman notifikasi. Visualisasi alarm muncul saat *running hour* telah mencapai batas yang telah ditetapkan. Visualisasi alarm akan menampilkan data mesin yang telah mencapai batas *running hour* berupa nama mesin dan nama perusahaan/pabrik mesin. Selain pada halaman notifikasi, visualisasi alarm juga dapat dilihat pada tab pemantauan mesin berupa warna merah pada bagian *running hour* mesin yang ditunjukkan pada Gambar 12.



Gambar 12. Visualisasi alarm pada tab pemantauan mesin *web* dan *mobile*

Selain pengujian simulasi, dilakukan pula pengujian *Black Box* untuk menguji secara fungsional aplikasi *web* dan *mobile* MoRe telah bekerja dengan baik dan mengeluarkan hasil yang sesuai dengan yang diharapkan. Pengujian *Black Box* dilakukan oleh 3 orang perwakilan PT. Numalos Abadi yaitu *Lead Technician Development (user 1)*, *Staff Electrical Engineering (user 2)*, dan *Staff Technician Development (user 3)*. Pengujian *Black Box* dilakukan pada fitur pemantauan mesin, laporan mesin, notifikasi alarm, dan visualisasi alarm. Pengujian pada 4 variabel tersebut dilakukan untuk memastikan bahwa konektivitas berfungsi dengan baik dan pengolahan data yang ditampilkan sesuai dengan yang diharapkan. Hasil pengujian *Black Box web* dan *mobile* MoRe dapat dilihat sebagai berikut:

Tabel 1. Hasil pengujian *Black Box web* dan *mobile* MoRe

	Pemantauan mesin	Laporan mesin	Notifikasi alarm	Visualisasi alarm
<i>User 1</i>	<i>Success</i>	<i>Success</i>	<i>Success</i>	<i>Success</i>
<i>User 2</i>	<i>Success</i>	<i>Success</i>	<i>Success</i>	<i>Success</i>
<i>User 3</i>	<i>Success</i>	<i>Success</i>	<i>Success</i>	<i>Success</i>

5. KESIMPULAN

Kesimpulan yang dapat disampaikan dari hasil penelitian adalah hasil pengujian simulasi menunjukkan bahwa secara fungsional aplikasi telah bekerja dengan baik dimulai dari informasi *running hour* dapat terpantau melalui aplikasi *web* dan juga *mobile*, serta visualisasi alarm yang dimunculkan ketika batas *running hour* telah tercapai.

6. SARAN

Saran yang dapat disampaikan melalui penelitian ini yaitu sistem yang sudah dikembangkan dapat diimplementasikan secara langsung terhadap mesin *decanter* Flottweg Z5E serta dilakukan pengujian kembali dan *survey* kepuasan pelanggan setelah sistem ini diimplementasikan.

DAFTAR PUSTAKA

- [1] M. R. Sahay, M. K. Sukumaran, S. Amarnath and T. N. D. Palani, "Environmental Monitoring System Using IoT and Cloud Service at Real-Time," *EasyChair Preprint*, vol. 968, p. 2, 2019.
- [2] Amazon Web Services, Inc., "AWS," Amazon Web Services, Inc., 2022. [Online]. Available: https://aws.amazon.com/id/about-aws/global-infrastructure/?nc1=h_ls. [Accessed 23 8 2022].
- [3] Mashuda, "Rancang Bangun Mesin Pencacah Usus Unggas Untuk Pakan Ternak Ikan," *eprints Politeknik Negeri Sriwijaya*, p. 5, 2015.
- [4] "FLOTTWEG DECANTER CENTRIFUGES," Flottweg SE, [Online]. Available: <https://www.flottweg.com/product-lines/decanter/>. [Accessed 17 11 2021].
- [5] "AWS IoT Core," Amazon Web Services, Inc., 2021. [Online]. Available: <https://aws.amazon.com/id/iot-core/>. [Accessed 1 11 2021].
- [6] "AWS Lambda," Amazon Web Services, Inc., [Online]. Available: <https://aws.amazon.com/id/lambda/>. [Accessed 28 Maret 2022].
- [7] "Amazon Relational Database Service," Amazon Web Services, Inc., 28 Maret 2022. [Online]. Available: <https://aws.amazon.com/id/rds/>. [Accessed 28 Maret 2022].
- [8] "AWS Amplify," Amazon Web Services, Inc., [Online]. Available: <https://aws.amazon.com/id/amplify/>. [Accessed 28 Maret 2022].
- [9] "AWS Elastic Beanstalk," Amazon Web Services, Inc., [Online]. Available: <https://aws.amazon.com/id/elasticbeanstalk/>. [Accessed 28 Maret 2022].

- [10] "AWS Simple Storage Service," Amazon Web Services, Inc., [Online]. Available: <https://aws.amazon.com/id/s3/>. [Accessed 28 Maret 2022].
- [11] Kepala Pusat Pendidikan dan Pelatihan Jalan, Perumahan, Permukiman dan Pengembangan Infrastruktur Wilayah, "BPSDM Kementerian PU dan Perumahan Rakyat," 5 2018. [Online]. Available: https://bpsdm.pu.go.id/center/pelatihan/uploads/edok/2018/05/6953a_Modul_Monitoring_dan_Evaluasi_Perkotaan.docx. [Accessed 19 10 2021].
- [12] S. Hamali, "BINUS UNIVERSITY BUSINESS SCHOOL," BBS-Management, 29 11 2017. [Online]. Available: <https://bbs.binus.ac.id/management/2017/11/simulasi/>. [Accessed 23 3 2022].