

RANCANG BANGUN SISTEM MONITORING BERBASIS WEB UNTUK PENGAMBILAN DATA REAL-TIME DAN DETEKSI ANOMALI PARAMETER MESIN PADA PT XYZ

Divasetya Pratama Putri, Intan Purnamasari, Dadang Yusup
Informatika, Universitas Singaperbangsa Karawang
Jl. H.S Ronggowaluyo, Telukjambe Timur, Karawang, Indonesia
2110631170009@student.unsika.ac.id

ABSTRAK

Pengendalian parameter suhu pada mesin produksi komponen aluminium merupakan faktor krusial dalam menjaga standar kualitas produk. Saat ini, PT XYZ masih menerapkan pemantauan manual berbasis *memory card* dan Excel yang menyebabkan keterlambatan visualisasi data hingga dua jam, sehingga menghambat deteksi dini anomali dan meningkatkan risiko produk cacat. Penelitian ini bertujuan membangun sistem *monitoring* berbasis *web* untuk akuisisi data dan deteksi anomali secara *real-time*. Pengembangan sistem menerapkan metode *Agile Scrum* yang dilaksanakan dalam lima siklus *sprint*, serta divalidasi melalui pengujian *Black Box* dan *White Box Testing* untuk menjamin kualitas fungsional dan logika kode. Hasil penelitian menunjukkan bahwa sistem mampu memangkas waktu pemrosesan data dari hitungan jam menjadi instan dengan akurasi notifikasi anomali yang valid. Implementasi ini terbukti meningkatkan efisiensi operasional dan mempercepat pengambilan keputusan penanggulangan masalah di lantai produksi.

Kata kunci : sistem monitoring, real-time, deteksi anomali, Agile Scrum, web

1. PENDAHULUAN

PT. XYZ dikenal sebagai perusahaan yang beroperasi di bidang otomotif. Perusahaan ini memproduksi komponen mesin mobil berbahan dasar aluminium, sehingga suhu dari setiap titik tertentu pada mesin produksi sangat mempengaruhi kualitas produk yang dihasilkan. Hal tersebut mengharuskan karyawan memantau parameter mesin produksi secara berkala agar suhu di setiap titik tersebut tetap sesuai dengan standar yang berlaku.

Berdasarkan observasi di lapangan, proses pemantauan suhu saat ini masih berjalan secara manual. Data parameter diambil dari *Programmable Logic Controller* (PLC) menggunakan *memory card*, kemudian dipindahkan dan diolah menggunakan Microsoft Excel untuk menghasilkan grafik *control chart*. Proses manual ini memakan waktu yang cukup panjang, menyebabkan visualisasi data baru tersedia hingga dua jam setelah produksi berlangsung. Keterlambatan informasi ini mengakibatkan penanganan mesin yang mengalami anomali suhu menjadi terhambat. Dampak fatal dari keterlambatan ini adalah risiko lolosnya produk cacat hingga ke tangan konsumen karena anomali tidak terdeteksi sejak dini.

Salah satu kasus yang pernah terjadi di PT. XYZ adalah ketika produk cacat berhasil lolos dari proses *quality check* dan sampai ke tangan konsumen. Untuk mengatasi permasalahan latensi data dan risiko kualitas tersebut, diperlukan peralihan dari metode manual ke sistem terotomatisasi. Penelitian ini mengusulkan pengembangan sistem *monitoring* berbasis *web* yang mampu mengakuisisi data suhu secara *real-time* langsung dari *database* mesin. Sebagai rencana penyelesaian masalah, sistem akan dilengkapi dengan fitur deteksi anomali otomatis yang

memberikan notifikasi visual (*pop-up*) seketika saat suhu melewati ambang batas. Pengembangan sistem ini akan menerapkan metode *Agile Scrum* untuk memastikan fitur-fitur yang dibangun sesuai dengan kebutuhan dinamis di lantai produksi.

Berdasarkan pemaparan permasalahan dan solusi yang diajukan, penelitian ini bermaksud untuk merancang sistem *monitoring* berbasis *web* yang dapat diakses oleh karyawan melalui laptop di kantor. Sistem ini menyediakan alat pengawasan yang cepat dan akurat, sehingga teknisi dapat melakukan perbaikan mesin segera dan meminimalkan potensi produk cacat dalam siklus produksi.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Sistem *monitoring* telah banyak dikembangkan pada penelitian terdahulu. Pada penelitian Putra & Sardiko, metode *Waterfall* digunakan untuk mengembangkan sistem *monitoring* yang berhasil menampilkan data parameter mesin boiler DBU secara *real-time* dan menyimpan riwayat nilai efisiensi mesin [1]. Lalu, penelitian yang dilakukan oleh Ardiansyah & Firdausi, menggunakan metode *Waterfall* dan konsep perhitungan efisiensi *Overall Equipment Effectiveness* (OEE) telah berhasil membangun sistem *monitoring* yang dapat meningkatkan efisiensi dalam menganalisis performa mesin extruder DTE-4 melalui data yang tercatat. Hasil penelitian menunjukkan bahwa sistem mampu menghemat waktu dalam proses pengisian form OEE sebesar 68,16% yang secara manual menghabiskan waktu rata-rata 345 detik menjadi 110 detik dengan bantuan sistem. Sistem juga mampu menghemat waktu dalam memantau kondisi mesin sebesar 95,44% [2].

2.2. Sistem Monitoring Berbasis Web

Sistem adalah beberapa komponen berbeda yang saling berhubungan untuk membentuk sebuah fungsi tertentu [3]. *Monitoring* merupakan kegiatan yang bertujuan untuk mengumpulkan dan menganalisis informasi yang telah didapatkan dengan tujuan untuk memahami cara kerja dari suatu program atau sistem [4]. Dengan perkembangan teknologi, kegiatan *monitoring* saat ini dapat dilakukan melalui sebuah sistem digital, salah satunya adalah sistem berbasis *web* yang dapat diakses melalui *browser*. Sistem *monitoring* berbasis *web* adalah sistem yang digunakan untuk mengumpulkan dan menganalisis informasi dengan menggunakan teknologi berbasis *internet* [5].

2.3. Unified Modelling Language (UML)

Unified Modeling Language (UML) merupakan bahasa berbasis notasi dan diagram yang digunakan untuk memvisualisasikan dan mendokumentasikan sebuah sistem dalam proses pengembangan perangkat lunak [6]. *Use case diagram* merupakan salah satu jenis diagram UML yang menjelaskan fitur atau tugas yang dapat dilakukan sebuah sistem dari sudut pandang pengguna [7]. *Activity diagram* merupakan diagram yang digunakan untuk menjelaskan aktivitas yang terjadi di dalam sistem [6]. *Sequence diagram* berfungsi sebagai alat untuk menjabarkan kebutuhan sistem, mendokumentasikan alur proses, dan menggambarkan kerja sistem saat sedang beroperasi [8].

2.4. Agile Scrum

Agile merupakan model pengembangan perangkat lunak yang memiliki sistem kerja secara iteratif. Prinsip utama dalam pengembangan sistem *Agile* adalah mampu menanggapi kebutuhan sistem yang sering berubah dan pengerjaan dalam periode singkat. Metode *Agile* juga mengutamakan kolaborasi dengan pengguna sistem [9].

Scrum sering digunakan dalam pengembangan sistem yang memiliki tingkat kompleksitas tinggi. Sifatnya yang adaptif memungkinkan fitur, persyaratan, atau fungsi yang sedang dikembangkan dalam proyek dapat diubah sesuai dengan kebutuhan pengguna [10].

2.5. White Box Testing

White Box Testing adalah pengujian yang dilakukan dengan fokus terhadap logika internal dan struktur kode. Dalam implementasinya, pemrogram memiliki pemahaman penuh terhadap desain, arsitektur, dan implementasi internal kode program. Tujuannya adalah untuk memastikan bahwa semua jalur kode telah dieksekusi setidaknya sekali, setiap kondisi logis berfungsi dengan benar, dan potensi kerentanan atau *bug* yang tersembunyi dapat teridentifikasi sebelum sistem digunakan oleh pengguna akhir [11].

Teknik *Basis Path Testing* adalah salah satu teknik pengujian yang melibatkan identifikasi seluruh *independent path* dari *flowgraph* kode. Selanjutnya adalah menghitung *cyclomatic complexity* dari *flowgraph* yang telah dibuat dan memastikan setiap jalur independen telah diuji setidaknya satu kali [12]. *Cyclomatic complexity* dihitung dengan menggunakan rumus berikut:

$$V(G) = E - N + 2 \quad (1)$$

Dimana:

$V(G)$ = Cyclomatic Complexity

E = Edge

N = Node

Selain persamaan (1), *Cyclomatic Complexity* juga bisa dihitung menggunakan rumus yang melibatkan jumlah *predicate node* pada *flowgraph* yang dirumuskan pada persamaan (2) [13].

$$V(G) = P + 1 \quad (2)$$

Dimana:

P = Predicate Node

2.6. Black Box Testing

Black Box Testing adalah pengujian yang dilakukan untuk mengetahui fungsi dari sebuah sistem berjalan sesuai dengan spesifikasi yang telah ditetapkan. Fokus utama dari pengujian ini adalah mendeteksi fitur yang tidak berjalan sesuai dengan fungsinya, ketidaksesuaian antarmuka sistem, ketidaksesuaian dalam struktur data, permasalahan pada performa sistem, dan potensi kesalahan selama proses inisialisasi dan terminasi [14].

3. METODE PENELITIAN

3.1. Analisis Kebutuhan Sistem

Pada tahap ini, penulis melakukan wawancara dengan *product owner* untuk mengumpulkan kebutuhan pengguna yang akan digunakan dalam menetapkan spesifikasi berupa fitur utama pada sistem yang akan dikembangkan. Hasil dari wawancara dijadikan sebagai dasar dalam proses pemodelan *use case* dan *sprint backlog*.

3.2. Sprint

Setiap siklus *sprint* berjalan selama tiga minggu, di mana setiap siklus mencakup tiga aktivitas utama, yaitu perancangan sistem, implementasi desain, dan pengujian. Proses perancangan sistem divisualisasikan menggunakan diagram *Unified Modeling Language* (UML) meliputi *Activity Diagram*, dan *Sequence Diagram*. Pada tahap implementasi, desain sistem diimplementasikan ke dalam *web* interaktif menggunakan HTML, CSS, JavaScript, dan PHP

Pengujian sistem yang dilakukan menggunakan metode *Black Box Testing* dan *White Box Testing*. Pada pengujian *black box*, *product owner* akan menguji fungsionalitas dari sistem yang dikembangkan, sedangkan pengujian *white box*, dilakukan oleh pengembang untuk memastikan bahwa

struktur internal, logika, dan alur kode program telah berjalan sesuai dengan yang dirancang.

3.3. Sprint Review

Sprint review berfokus pada mempresentasikan potongan fungsionalitas sistem yang telah selesai dikerjakan dan siap digunakan kepada *stakeholder*. Saran dan masukan yang didapatkan dari demonstrasi menjadi dasar untuk memperbarui dan menyesuaikan kembali prioritas item dalam *product backlog*.

3.4. Sprint Retrospective

Evaluasi kinerja pengembang dilakukan dalam tahap *sprint retrospective* ini. Tahapan ini bertujuan untuk mengidentifikasi apa yang sudah berjalan baik, apa yang menjadi kendala, serta mencari potensi perbaikan untuk *sprint* selanjutnya. Hasil dari tahapan digunakan untuk meningkatkan efektivitas dan kualitas kerja di siklus *sprint* selanjutnya.

4. HASIL DAN PEMBAHASAN

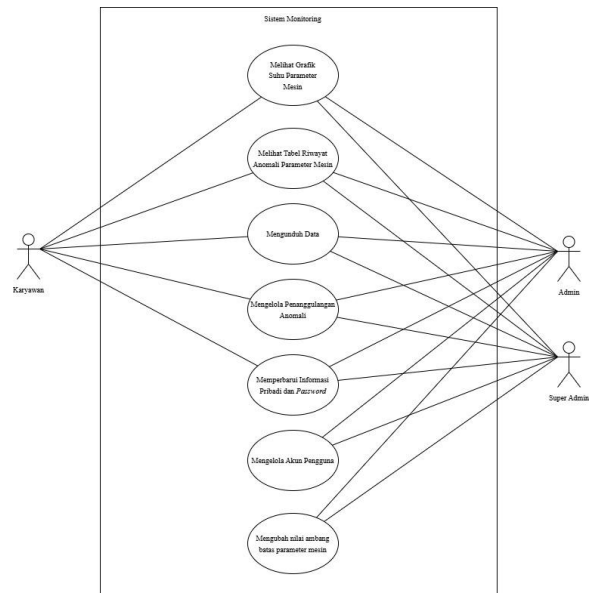
4.1. Hasil Analisis Kebutuhan Sistem

Tabel 1 merupakan hasil dari proses wawancara dan *sprint planning* yang sudah diringkas sesuai dengan kebutuhan utama sistem.

Tabel 1. *Sprint Backlog* Sistem

User Story (Fitur)	Sprint Ke-
Sebagai pengguna, saya ingin sistem ini menerapkan tahapan <i>login</i> agar data <i>monitoring</i> tetap aman dan hanya bisa diakses oleh yang berwenang.	1
Sebagai pengguna, saya ingin bisa <i>logout</i> dari sistem agar orang lain tidak bisa menyalahgunakan akun saya saat perangkat tidak diawasi.	1
Sebagai pengguna, saya ingin melihat grafik suhu parameter mesin setiap satu siklus produksi yang masuk setiap 5 menit agar saya dapat memantau tren suhu selama satu siklus produksi penuh.	1
Sebagai pengguna, saya ingin mendapatkan notifikasi di layar ketika suhu mesin melewati ambang batas agar saya bisa segera melakukan tindakan penanganan.	2
Sebagai pengguna, saya ingin grafik <i>monitoring</i> suhu parameter mesin berubah warna saat suhu parameter mesin melewati ambang batas agar saya bisa segera melakukan tindakan penanganan.	2

Setiap *user story* pada Tabel 1 merepresentasikan fitur kunci yang dirancang untuk menggantikan proses manual di PT XYZ. Secara keseluruhan, sistem ini dikembangkan dengan total 22 fitur (*product backlog items*) yang diselesaikan dalam lima siklus *sprint*, di mana setiap *sprint* berdurasi tiga minggu.

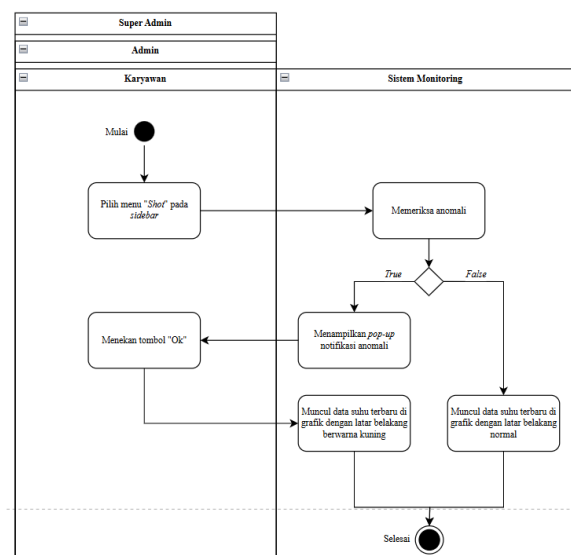


Gambar 1. Use Case Diagram Sistem

Gambar 1 adalah *use case diagram* sistem yang menggambarkan keseluruhan fitur yang dapat diakses melalui sistem *monitoring* ini. Terdapat tiga aktor dengan hak akses ke fitur yang berbeda.

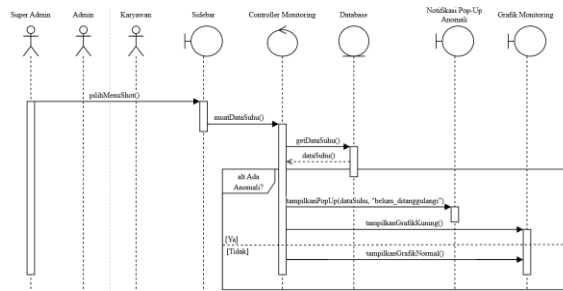
4.2. Hasil Sprint 1

Pada siklus *sprint* 1 ini, fitur yang diselesaikan adalah fitur *login*, *logout*, dan grafik *monitoring* parameter suhu. Berikut ini adalah hasil dari pengembangan sistem pada *sprint* 1.



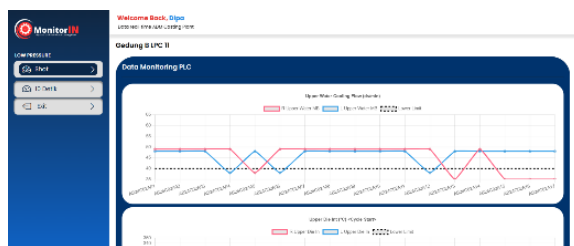
Gambar 2. Activity Diagram Fitur Utama Sistem

Gambar 2 menunjukkan alur proses untuk mengakses fitur grafik *monitoring*, notifikasi *pop-up* anomali, dan indikator visual grafik. Alur dari ketiga fitur tersebut saling berdampingan, sehingga penulis menggabungkannya ke dalam satu diagram.



Gambar 3. Sequence Diagram Fitur Utama Sistem

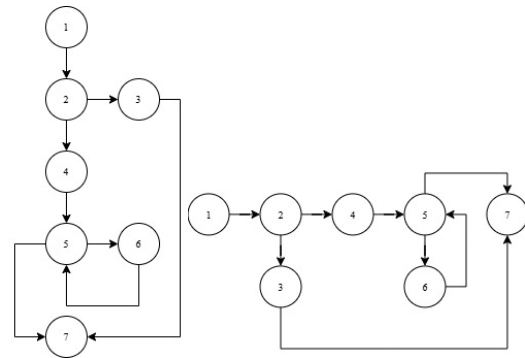
Berdasarkan *activity diagram* pada Gambar 2, *sequence diagram* dari ketiga fitur juga digabungkan menjadi satu diagram utuh yang menunjukkan interaksi antar objek pada fitur grafik *monitoring*, notifikasi *pop-up* anomali, dan indikator visual grafik.



Gambar 4. Hasil Pengembangan Fitur pada Sprint 1

Gambar 4 menampilkan antarmuka dari halaman utama yang langsung ditampilkan saat pengguna berhasil melakukan tahap *login*. Pada bagian utama,

terdapat grafik *monitoring* parameter mesin yang akan diperbarui setiap 5 menit, yaitu setiap satu siklus produksi selesai. Pada *sidebar menu*, terlihat tombol “Exit” yang memungkinkan pengguna untuk keluar dari sistem (*logout*). Saat pengguna menekan tombol tersebut, pengguna akan langsung diarahkan ke halaman *login* sistem.



Gambar 5. Flowgraph Fitur Grafik Monitoring

Gambar 5 adalah *flowgraph* dari logika yang digunakan pada fitur grafik *monitoring*. *Flowgraph* di sebelah kiri adalah logika untuk mengambil nilai data parameter mesin dari *database*, sedangkan *flowgraph* di sebelah kanan adalah logika untuk mengambil nilai ambang batas suhu dari *database*. Nilai *cyclomatic complexity* dari kedua *flowgraph* tersebut adalah 3. Berikut ini adalah *test case* untuk fitur grafik *monitoring*.

Tabel 2. Test Case Logika Fitur Grafik Monitoring

Test Case	Hasil yang Diharapkan	Status
Pengambilan Data Parameter Mesin		
Ubah kredensial di kode logika koneksi ke <i>database</i> menjadi salah.	Script berhenti. Pengguna langsung diarahkan ke Halaman <i>Login</i> (<i>index.php</i>).	Berhasil
Pastikan koneksi <i>database</i> berhasil, kosongkan tabel <i>datasensor</i> .	Grafik <i>monitoring</i> kosong.	Berhasil
Pastikan koneksi <i>database</i> berhasil, dan tabel <i>datasensor</i> berisi setidaknya 1 baris data.	Grafik <i>monitoring</i> terisi data.	Berhasil
Pengambilan Nilai Ambang Batas Suhu		
Ubah kredensial di kode logika koneksi ke <i>database</i> menjadi salah.	Script berhenti. Pengguna langsung diarahkan ke Halaman <i>Login</i> (<i>index.php</i>).	Berhasil
Pastikan koneksi <i>database</i> berhasil, kosongkan tabel <i>sensor_limits_lpc11</i> di <i>database</i> .	Garis ambang batas tidak muncul di grafik <i>monitoring</i> per siklus produksi.	Berhasil
Pastikan koneksi <i>database</i> berhasil dan tabel <i>sensor_limit</i> di <i>database</i> terisi setidaknya 1 baris data.	Garis ambang batas muncul dengan nilai yang sesuai di grafik <i>monitoring</i> per siklus produksi.	Berhasil

Berdasarkan hasil pengujian menggunakan *test case* yang tertera pada Tabel 2, logika fitur grafik *monitoring* menunjukkan keberhasilan dan tidak menunjukkan adanya *bug* atau *error* pada program.

Selanjutnya adalah tahap *Black Box Testing* yang hasilnya tertera pada

Tabel 3.

Tabel 3. Pengujian Black Box Sprint 1

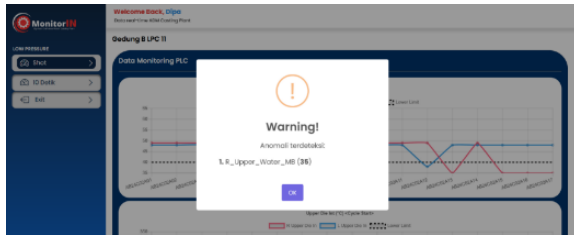
Fitur	Skenario Pengujian	Hasil yang Diharapkan	Status
Grafik <i>Monitoring</i> Per Siklus Produksi	Masukkan data suhu baru ke dalam <i>database</i> secara manual.	Grafik menunjukkan perubahan data suhu di <i>web</i> dalam rentang waktu 5 menit.	Sesuai

Berdasarkan hasil pengujian pada

Tabel 3, fitur *login*, *logout*, dan grafik *monitoring* telah memenuhi kebutuhan fungsional yang dapat divalidasi dengan status dari pengujian yang dilakukan terhadap fitur.

4.3. Hasil Sprint 2

Sprint 2 berhasil menyelesaikan fitur notifikasi *pop-up* anomali dan indikator visual. Hasil implementasi fitur ke dalam bentuk *web* fungsional dijelaskan pada pembahasan berikut.



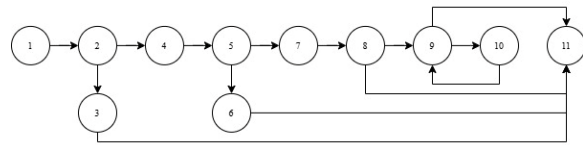
Gambar 6. Fitur Notifikasi *Pop-Up* Anomali

Gambar 6 adalah tampilan fitur notifikasi *pop-up* yang hanya muncul saat data suhu yang masuk mengalami anomali. Fitur ini bertugas untuk memberikan peringatan secara *real-time* kepada pengguna saat sistem mendeteksi adanya penyimpangan atau anomali pada suhu parameter mesin.



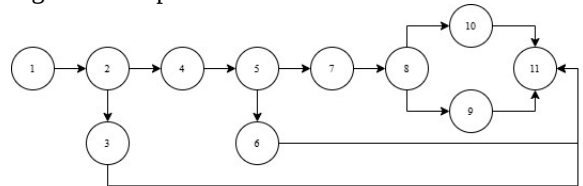
Gambar 7. Fitur Indikator Visual Grafik

Gambar 7 menunjukkan indikator visual dari grafik suhu dengan sensor suhu yang mengalami anomali. Fitur ini berfungsi sebagai umpan balik visual, di mana perubahan warna diterapkan pada latar belakang grafik ketika salah satu parameter suhu terdeteksi mengalami anomali.



Gambar 8. *Flowgraph* Logika Fitur Notifikasi *Pop-Up* Anomali

Gambar 8 adalah *flowgraph* dari logika kode untuk fitur Notifikasi *Pop-Up* Anomali. Nilai *cyclomatic complexity* dari *flowgraph* tersebut adalah 5. Nilai tersebut menunjukkan bahwa terdapat 5 *independent path* yang harus diuji. Selanjutnya adalah *flowgraph* dari logika fitur indikator visual grafik yang digambarkan pada Gambar 9.



Gambar 9. *Flowgraph* Logika Fitur Indikator Visual Grafik

Nilai *cyclomatic complexity* dari *flowgraph* pada Gambar 9 adalah 4, yang berarti terdapat 4 *independent path* yang harus diuji. Daftar *test case* yang diujikan pada fitur notifikasi *pop-up* anomali dan indikator visual grafik tertera pada tabel berikut ini.

Tabel 4. *Test Case* Fitur *Sprint 2*

Test Case	Hasil yang Diharapkan	Status
Notifikasi <i>Pop-Up</i> Anomali		
Ubah kredensial di kode logika koneksi ke <i>database</i> menjadi salah.	Script berhenti. Pengguna langsung diarahkan ke Halaman <i>Login</i> (<i>index.php</i>).	Berhasil
Pastikan koneksi ke <i>database</i> berhasil, namun ubah <i>query</i> menjadi <i>query</i> yang salah.	Muncul pesan kesalahan “ <i>Query error</i> ”	Berhasil
Pastikan koneksi ke <i>database</i> berhasil, <i>query</i> benar, pastikan tidak ada data anomali dengan status ‘belum ditanggulangi’	Halaman tidak menampilkan notifikasi <i>pop-up</i> anomali.	Berhasil
Pastikan koneksi ke <i>database</i> berhasil, <i>query</i> benar, isi tabel <i>abnormalities</i> dengan satu baris data berstatus ‘belum ditanggulangi’.	Halaman menampilkan notifikasi <i>pop-up</i> anomali berisi satu data anomali.	Berhasil
Pastikan koneksi ke <i>database</i> berhasil, <i>query</i> benar, dan pastikan tabel <i>abnormalities</i> memiliki lebih dari satu baris data dengan status ‘belum ditanggulangi’.	Halaman menampilkan notifikasi <i>pop-up</i> anomali berisi data anomali yang terdaftar.	Berhasil
Indikator Visual Grafik		
Menguji dengan memberikan <i>array</i> sensor yang kosong saat memanggil fungsi.	Fungsi akan langsung mengembalikan nilai <i>false</i> .	Berhasil
Menguji dengan memberikan <i>array</i> sensor yang ada di tabel <i>database</i> , namun koneksi ke <i>database</i> gagal.	Fungsi akan mengembalikan nilai <i>false</i> .	Berhasil
Menguji dengan <i>array</i> sensor yang tidak ditemukan di <i>database</i> .	Fungsi akan mengembalikan nilai <i>false</i> .	Berhasil
Menguji dengan <i>array</i> sensor, yang ada di <i>database</i>	Fungsi akan mengembalikan nilai <i>true</i> .	Berhasil

Berdasarkan hasil pengujian menggunakan *test case* yang ada pada Tabel 4, kedua fitur berhasil dijalankan tanpa menunjukkan adanya *bug* atau *error*

pada program. Selanjutnya adalah tahap *Black Box Testing* yang disajikan pada Tabel 5.

Tabel 5. Hasil Pengujian *Black Box Testing Sprint 2*

Fitur	Langkah Skenario Pengujian	Hasil yang Diharapkan	Status
Notifikasi <i>Pop-Up</i> Anomali	Masukkan data suhu mesin baru yang melewati ambang batas yang telah ditetapkan.	Muncul notifikasi di Halaman Utama yang memberitahukan sensor suhu yang mengalami anomali beserta nilai aktualnya.	Sesuai
Indikator Visual Anomali Grafik	Masukkan data suhu mesin baru yang melewati ambang batas yang telah ditetapkan.	Warna latar Grafik <i>Monitoring</i> Per Siklus Produksi berubah menjadi kuning.	Sesuai

Berdasarkan hasil pengujian pada Tabel 5, fitur notifikasi *pop-up* anomali dan indikator visual grafik telah memenuhi kebutuhan fungsional yang dapat dibuktikan dengan hasil *Black Box Testing* fitur. Kedua fitur tersebut dapat berfungsi dengan baik saat terdapat data parameter suhu yang mengalami anomali.

4.4. Hasil Sprint Review

Sprint review dilakukan di setiap hari Senin dengan durasi selama 1 jam dan dilakukan setiap satu minggu sekali. Melalui kegiatan *review* ini, tim mengidentifikasi satu penyesuaian besar pada *product backlog* yang disajikan pada Tabel 6.

Tabel 6. Penyesuaian *Product Backlog*

Siklus Sprint	Fitur	Tambahan	Penyesuaian
<i>Sprint</i> ke-2	Notifikasi <i>Pop-Up</i> Anomali	Perlu indikator yang bisa memberitahu pengguna grafik sensor mana yang mengalami anomali.	Menambahkan <i>product backlog</i> baru ke dalam siklus <i>sprint</i> yang sedang berjalan, yaitu <i>sprint</i> 2.

Penyesuaian pada Tabel 6 dilakukan untuk mengakomodasi umpan balik dari *stakeholder* yang menginginkan peningkatan pada aspek visualisasi saat muncul data parameter suhu yang mengalami anomali. Penambahan warna yang kontras pada grafik diperlukan agar karyawan dapat mengidentifikasi sensor spesifik yang mengalami anomali dengan lebih cepat sehingga karyawan tidak perlu membaca detail notifikasi satu per satu.

4.5. Hasil Sprint Retrospective

Sprint retrospective dilakukan setiap satu minggu sekali, di hari Rabu, dengan durasi selama 15-30 menit. Dalam salah satu siklus *sprint*, tim mengidentifikasi bahwa frekuensi komunikasi antar anggota tim masih perlu ditingkatkan. Hal ini dicatat sebagai kendala yang berdampak pada kurangnya transparansi progres pengembangan antar anggota. Selanjutnya, tim internal mengadakan diskusi perencanaan *sprint* berikutnya dengan fokus utama meminimalkan semua kendala dan kesulitan yang telah teridentifikasi. Hal ini memastikan proses pengembangan di *sprint* selanjutnya berjalan lebih efisien.

4.6. Pembahasan

Berdasarkan hasil pengembangan yang telah melalui lima siklus *sprint*, kualitas teknis sistem telah teruji validitasnya. Pengujian *White Box* menggunakan teknik *Basis Path Testing* pada fitur-fitur kritis seperti logika grafik *monitoring* dan notifikasi *pop-up* menghasilkan nilai *cyclomatic complexity* yang efisien, memastikan tidak adanya *looping* tak terbatas atau logika yang ambigu. Selanjutnya, pengujian *Black Box* mengonfirmasi bahwa kelima fitur utama (*login*, *logout*, grafik *monitoring*, notifikasi, dan indikator visual) berjalan sesuai skenario dan mampu menangani data anomali secara responsif.

Implementasi sistem ini memberikan dampak langsung terhadap efisiensi operasional di departemen produksi PT XYZ. Sebelum adanya sistem ini, ketergantungan pada pemindahan data manual dari *memory card* menyebabkan jeda waktu yang signifikan dalam mendeteksi penyimpangan suhu. Perbandingan kondisi operasional sebelum dan sesudah implementasi sistem disajikan pada Tabel 7.

Tabel 7. Perbandingan Proses *Monitoring*

Proses Manual	Implementasi Sistem <i>Monitoring</i>	Dampak Signifikan
Karyawan harus menunggu <i>shift</i> kerja selesai. Proses meliputi menyortir data dan mengolahnya di Excel yang menghabiskan waktu sekitar 2 jam.	Data diperbarui secara <i>real-time</i> . Grafik <i>monitoring</i> per siklus produksi diperbarui selama 5 menit, grafik <i>monitoring</i> per parameter diperbarui selama 10 detik.	Efisiensi waktu pemrosesan data hingga >90%.
Karyawan harus mengecek <i>control chart</i> secara manual setelah data diolah.	Sistem langsung memberikan notifikasi <i>pop-up</i> dan indikator warna visual.	Percepatan respons penanganan masalah.
Terbatas pada laptop yang memegang file Excel.	Dapat diakses dari laptop manapun dalam jaringan lokal (<i>Intranet</i>) via <i>browser</i> .	Peningkatan transparansi dan kemudahan <i>monitoring</i> jarak jauh.

Berdasarkan Tabel 7, transformasi ke sistem berbasis *web* terbukti mengatasi kendala latensi data. Waktu visualisasi yang sebelumnya membutuhkan durasi sekitar dua jam kini dapat dilakukan secara *real-time* dengan pembaruan data per siklus produksi. Hal ini tidak hanya meningkatkan efisiensi waktu hingga lebih dari 90%, tetapi juga mengubah mekanisme pengawasan dari yang bersifat pasif (menunggu laporan akhir *shift*) menjadi aktif (notifikasi instan), sehingga tindakan pencegahan terhadap produk cacat dapat dilakukan segera saat suhu mesin melewati ambang batas.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan membangun sistem *monitoring* parameter mesin berbasis *web* menggunakan metode *Agile Scrum* yang diselesaikan dalam lima siklus *sprint*. Sistem ini terbukti mampu memvisualisasikan data suhu secara *real-time* dan mendeteksi anomali secara otomatis melalui fitur notifikasi *pop-up* serta indikator visual pada grafik. Keberhasilan fungsi-fungsi tersebut telah divalidasi melalui pengujian *White Box* dan *Black Box*, yang menunjukkan bahwa logika sistem berjalan efisien dan sesuai kebutuhan pengguna. Implementasi sistem ini secara signifikan mengubah alur kerja pemantauan suhu di PT XYZ, memangkas waktu pemrosesan data dari hitungan jam menjadi instan, serta meminimalkan risiko lolosnya produk cacat akibat keterlambatan deteksi.

Untuk pengembangan selanjutnya, disarankan agar sistem dapat diintegrasikan langsung dengan perangkat *Internet of Things* (IoT) guna menghilangkan ketergantungan pada infrastruktur basis data perantara dan meminimalkan latensi data. Selain itu, penerapan algoritma *Machine Learning* untuk *predictive maintenance* dapat ditambahkan agar sistem tidak hanya mendeteksi anomali yang sedang terjadi, tetapi juga memprediksi potensi kerusakan mesin. Penambahan fitur notifikasi melalui pesan instan seperti WhatsApp atau Telegram juga disarankan agar peringatan anomali dapat diterima oleh penanggung jawab teknis secara *mobile* dan fleksibel.

DAFTAR PUSTAKA

- [1] A. K. Putra and O. Sardiko, "Rancang Bangun Sistem Monitoring Efisiensi Pada Mesin Boiler Dbu Berbasis Web," *J. Instrumentasi dan Teknol. Inform.*, vol. 2, no. 2, pp. 63–68, 2021, [Online]. Available: <https://jurnal.poltek-gt.ac.id/index.php/jiti/article/view/12>
- [2] I. Ardiansyah and M. K. Firdausi, "Rancang Bangun Sistem Monitoring pada Mesin Extruder DTE-4 Berbasis Web," *J. instrumentasi dan Teknol. Inform.*, vol. 2, no. 2, pp. 88–94, 2021, [Online]. Available: <https://jurnal.poltek-gt.ac.id/index.php/jiti/article/view/16>
- [3] B. Prabaningrum, A. Voutama, and N. Heryana, "Rancang Bangun Sistem Manajemen Keuangan Berbasis Website Dalam Pengelolaan Laba Rugi," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 1, pp. 671–680, 2023, doi: 10.36040/jati.v7i1.6483.
- [4] N. Sari and D. Cahyani, "Perancangan Sistem Informasi Monitoring Sertifikat Menggunakan Extreme Programming," *J. Ilm. Comput. Sci.*, vol. 1, no. 1, pp. 1–6, 2022, doi: 10.58602/jics.v1i1.1.
- [5] A. A. Azis and A. Voutama, "RANCANGAN SISTEM MONITORING SISWA BERBASIS WEB DENGAN METODE WATERFALL," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 3, pp. 2072–2079, 2025.
- [6] Suharni, E. Susilowati, and F. Pakusadewa, "Perancangan Website Rumah Makan Ninik Sebagai Media Promosi Menggunakan Unified Modelling Language," *J. Rekayasa Inf.*, vol. 12, no. 1, pp. 1–12, 2023, [Online]. Available: <https://ejournal.istn.ac.id/index.php/rekayasainformasi/article/view/1527>
- [7] L. Setiyani, "Desain Sistem : Use Case Diagram Pendahuluan," *Pros. Semin. Nas. Inov. Adopsi Teknol. 2021*, vol. 1, no. 1, pp. 246–260, 2021, [Online]. Available: <https://ejournal.rosma.ac.id/index.php/inotek/article/view/183/142>
- [8] R. Rohmanto and T. Setiawan, "Perbandingan Efektivitas Sistem Pembelajaran Luring dan Daring Menggunakan Metode Use case dan Sequence Diagram," *Intern. (Information Syst. Journal)*, vol. 5, no. 1, pp. 53–62, 2022, doi: 10.32627/internal.v5i1.506.
- [9] F. Hardiansyah, A. Rizal, and I. Purnamasari, "Implementasi Metode Agile Scrum Dalam Pengembangan Aplikasi Pembelajaran Olahraga," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 2, pp. 1242–1247, 2023, doi: 10.36040/jati.v7i2.6734.
- [10] R. W. P. Pamungkas, A. N. Azizah, and B. S. Zebua, "Analisis penerapan metode scrum untuk meningkatkan efektivitas dalam pembuatan aplikasi melalui literature review," *J. Pendidik. Inform. dan Sains*, vol. 11, no. 2, pp. 156–164, 2022, doi: 10.31571/saintek.v11i2.4650.
- [11] D. Wintana, D. Pribadi, and M. Y. Nurhadi, "Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing," *J. Larik Ldng. Artik. Ilmu Komput.*, vol. 2, no. 1, pp. 8–16, 2022, doi: 10.31294/larik.v2i1.1382.
- [12] M. F. Londjo, "Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Login," *J. Siliwaangi*, vol. 7, no. 2, pp. 35–40, 2021, [Online]. Available: <https://jurnal.unsil.ac.id/index.php/jssainstek/article/view/4086>
- [13] I. M. A. O. Gunawan, G. Indrawan, and Sariyasa, "PENGEMBANGAN SISTEM INFORMASI

KEMAJUAN AKADEMIK MENGGUNAKAN MODEL INCREMENTAL BERBASIS EVALUASI USABILITY DAN WHITE BOX TESTING,” *SINTECH (Science Inf. Technol. J.*, vol. 4, no. 1, pp. 67–78, 2021, [Online]. Available: [https://ejournal.instiki.ac.id/index.php/sintechjo](https://ejournal.instiki.ac.id/index.php/sintechjournal/article/view/661)

urnal/article/view/661
[14] I. Firaldi, I. Purnamasari, and R. Mayasari, “Perancangan Sistem Layanan Pengaduan Masyarakat Di Desa Limusnunggal Berbasis Web,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 4, pp. 7615–7623, 2024, doi: 10.36040/jati.v8i4.10333.