



## Perancangan Sistem Informasi Pengelolaan Bug Berbasis Website Di PT. Javan Cipta Solusi

Lilis Emalia<sup>1</sup>, Yudhi Yanuar<sup>2</sup>, Dede Wahyudin<sup>3</sup>

<sup>1</sup>Teknik Informatika, STMIK Mardira Indonesia

<sup>2,3</sup>Manajemen Informatika, Politeknik Praktisi Bandung

[lilis.emalia@gmail.com](mailto:lilis.emalia@gmail.com), [yudhiyanuar2010@gmail.com](mailto:yudhiyanuar2010@gmail.com), [dwahyudin568@gmail.com](mailto:dwahyudin568@gmail.com)

### Abstract

The rapid development of information technology demands that companies continuously innovate in managing projects and improving software quality. PT. Javan Cipta Solusi, as a software development company, faces challenges in bug management within its projects. Although QA Touch has been used as the main tool in testing management, there is a need for a more focused and efficient system for recording and handling bugs. This research aims to design a Web-Based Bug Management Information System that meets the specific needs of the Quality Assurance (QA) team at PT. Javan Cipta Solusi. The system is designed to help the QA team record, track, and report bugs more efficiently and transparently. The methodology used includes requirement analysis, system design, implementation, and system testing. The designed information system has successfully met the QA team's needs in terms of more focused and systematic bug recording and reporting. Key features provided by this system, such as bug status tracking, bug prioritization, and bug analysis reports, have improved transparency and clarity of bug statuses, as well as accelerated the software repair cycle. The research results show that the recording process became faster, requiring an average of only 25 seconds, and weekly reports can be generated automatically in less than 7 minutes. In terms of oversight, the system is able to maintain the accuracy of bug statuses and priorities. As much as 92% of bugs were handled according to priority, and only 8% of bugs were delayed for more than three days. These quantitative data indicate that the designed system not only supports efficiency but also enhances the accountability of the QA process.

Keywords: *Information System, Bug Management, Website, Quality Assurance, Quantitative Data.*

### Abstrak

Perkembangan teknologi informasi yang pesat menuntut perusahaan untuk terus berinovasi dalam mengelola proyek dan meningkatkan kualitas perangkat lunak. PT. Javan Cipta Solusi sebagai perusahaan pengembangan perangkat lunak menghadapi tantangan dalam manajemen bug pada proyek-proyek yang dikerjakannya. Meskipun telah menggunakan QA Touch sebagai alat utama dalam manajemen pengujian, terdapat kebutuhan untuk sistem yang lebih terfokus dan efisien dalam pencatatan serta penanganan bug. Penelitian ini bertujuan untuk merancang Sistem Informasi Pengelolaan Bug Berbasis Website yang dapat memenuhi kebutuhan spesifik tim Quality Assurance (QA) di PT. Javan Cipta Solusi. Sistem ini dirancang untuk membantu tim QA dalam mencatat, melacak, dan melaporkan bug secara lebih efisien dan transparan. Metodologi yang digunakan meliputi analisis kebutuhan, perancangan sistem, implementasi, dan pengujian sistem. Sistem informasi yang dirancang telah berhasil memenuhi kebutuhan tim QA dalam hal pencatatan dan pelaporan bug yang lebih terfokus dan sistematis. Fitur-fitur utama yang disediakan oleh sistem ini, seperti pelacakan status bug, penentuan prioritas bug, dan laporan analisis bug, telah meningkatkan transparansi dan kejelasan status bug, serta mempercepat siklus perbaikan perangkat lunak. Hasil penelitian menunjukkan bahwa proses pencatatan menjadi lebih cepat, hanya memerlukan rata-rata 25 detik, dan pelaporan mingguan dapat dihasilkan secara otomatis dalam waktu kurang dari 7 menit. Dalam hal pengawasan, sistem ini mampu menjaga akurasi status dan prioritas bug. Sebanyak 92% bug ditangani sesuai prioritas, dan hanya 8% bug yang tertunda lebih dari tiga hari. Data kuantitatif tersebut menunjukkan bahwa sistem yang telah dirancang tidak hanya mendukung efisiensi, tetapi juga meningkatkan akuntabilitas proses QA.

Kata kunci: *Sistem Informasi, Pengelolaan Bug, Website, Quality Assurance, Data Kuantitatif.*



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

## 1. Pendahuluan

Perkembangan dunia teknologi, terutama dalam pengelolaan perangkat lunak, menjadi hal yang sangat signifikan dalam mendukung efisiensi dan produktivitas suatu perusahaan. Dalam konteks ini, PT. Javan Cipta Solusi, sebagai perusahaan pengembangan perangkat lunak, menghadapi tantangan efisiensi dalam manajemen *bug* pada proyek-proyeknya. Kesalahan atau *bug* dapat menyulitkan pemakai, karena *bug* dapat menghambat suatu pekerjaan. Dalam proses pengembangan perangkat lunak suatu *development* team ataupun seorang *developer* berkemungkinan besar mendapatkan *bug* dan *issue* (masalah). *Bug* adalah suatu *flaw* (cacat), *error*, atau *a fault* (gagal) pada komputer program yang menghasilkan sesuatu yang tidak diinginkan atau hasil yang tidak diprediksi[1]. Sebuah *bug* perlu untuk dihilangkan, tetapi *bug* tidak dapat dihindari dalam perangkat lunak. Berbagai cara dilakukan untuk mengurangi jumlahnya seperti, manajemen proyek perlu memperkirakan waktu yang dibutuhkan untuk menangani suatu *bug* agar dapat membuat perencanaan proyek yang baik, sehingga kualitas perangkat lunak yang diterima pengguna menjadi lebih baik[2]. Kualitas perangkat lunak telah menjadi salah satu pilar utama dalam kesuksesan aplikasi dan sistem di era digital saat ini. Dalam lingkungan yang semakin kompleks dan kompetitif, pemenuhan standar kualitas yang tinggi menjadi kunci untuk memastikan kepuasan pengguna, efisiensi operasional, dan reputasi yang baik bagi organisasi pengembang[3]. Saat ini, meskipun telah menggunakan QA *Touch* sebagai alat manajemen pengujian utama, terdapat kebutuhan spesifik untuk meningkatkan efektivitas dan efisiensi dalam pencatatan serta penanganan *bug*. Perubahan ini diperlukan guna memastikan fokus yang lebih terarah pada proses ini, bahkan tanpa memerlukan integrasi penuh dengan alat manajemen pengujian lainnya.

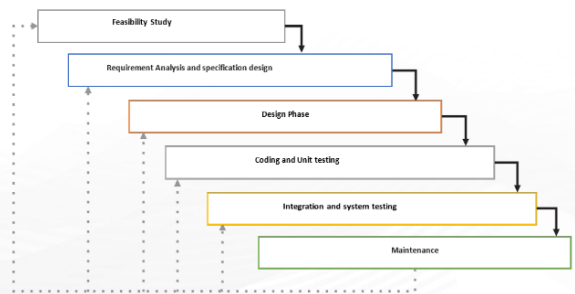
Dalam konteks ini, perancangan Sistem Informasi Pengelolaan *Bug* Berbasis *Website* diharapkan dapat menjadi solusi terkonsentrasi. Solusi ini diharapkan mampu memberikan kemudahan bagi tim *Quality Assurance* (QA) yang bertanggung jawab untuk memastikan bahwa aplikasi memenuhi semua kebutuhan teknis dan fungsional[4]. Tim QA di PT. Javan Cipta Solusi bertugas untuk mencatat, melacak, dan melaporkan *bug* secara lebih efisien. Implementasi sistem ini diharapkan dapat membawa dampak positif, seperti peningkatan kecepatan respons terhadap *bug*, transparansi status *bug* yang lebih baik, serta percepatan siklus perbaikan perangkat lunak.

Pentingnya perhatian terhadap pengelolaan bug diprioritaskan mengingat PT. Javan Cipta Solusi menegaskan kualitas perangkat lunak sebagai salah satu fokus utama perusahaan. Berdasarkan latar belakang tersebut penelitian ini bertujuan untuk mengidentifikasi

fitur utama yang dapat mendukung pengawasan tim QA dalam proses pelaporan dan penanganan *bug*, serta meningkatkan kejelasan status dan prioritas *bug* di PT. Javan Cipta Solusi.

## 2. Metode Penelitian

Pemilihan metode pengembangan model *waterfall* yang ditunjukkan pada gambar 1 didasarkan pada kebutuhan akan perencanaan yang ketat dan struktur tahapan pengembangan yang jelas.



Gambar 1. Metode Model *Waterfall*

Keunggulan utama dari model ini adalah dokumentasi yang jelas dan terstruktur, sehingga mempermudah proses pemeliharaan serta pengembangan lebih lanjut. Selain itu, model *waterfall* meminimalkan risiko kesalahan karena setiap tahap harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya, memastikan sistem dikembangkan dengan landasan yang kuat[5].

Proses pengerjaannya dapat diuraikan sebagai berikut:

- a. Perencanaan (*Planning*)  
Pada tahap ini, penulis merinci kebutuhan, menetapkan jadwal, dan menentukan sumber daya yang dibutuhkan. Perencanaan menjadi landasan bagi seluruh proses pengembangan.
- b. Analisis Kebutuhan (*Requirements Analysis*)  
Penulis melakukan pengumpulan dan analisis kebutuhan pengguna untuk merancang sistem yang sesuai. Memahami batasan atau fungsi perangkat lunak yang sesuai dengan harapan pengguna. Caranya yaitu mendapatkan data kebutuhan pengguna dengan cara melakukan wawancara atau kuesioner, kemudian melakukan analisis terhadap data tersebut[6]. Identifikasi kebutuhan inti dan fungsionalitas sistem menjadi fokus utama, dengan pemahaman yang didapat dari interaksi langsung dengan pengguna.
- c. Desain (*Design*)  
Pembuatan diagram-diagram UML yang digunakan dalam pemodelan yaitu *Use case*, *Activity*, *Sequence*, dan *Class* di mana masing-masing diagram akan menggambarkan alur sistem yang dirancang[7].

desain UI/UX[8], dan perencanaan rinci untuk implementasi menjadi bagian integral dari proses ini.

d. Pengkodean (*Coding*)

Bahasa pemrograman yang digunakan adalah PHP[9] dan *framework CodeIgniter*[10]. Berdasarkan penelitian sebelumnya menyebutkan bahwa Sistem informasi berbasis web yang dikembangkan menggunakan *Framework CodeIgniter 4*, merupakan salah satu *framework PHP* yang terkenal dan dapat diandalkan dalam pengembangan aplikasi web[11].

e. Pengujian (*Testing*)

Penulis melakukan *blackbox testing* untuk memastikan bahwa sistem berfungsi sesuai kebutuhan yang telah diidentifikasi sebelumnya. Identifikasi dan perbaikan *bug* atau kesalahan dalam kode menjadi langkah krusial.

f. Pemeliharaan (*Maintenance*)

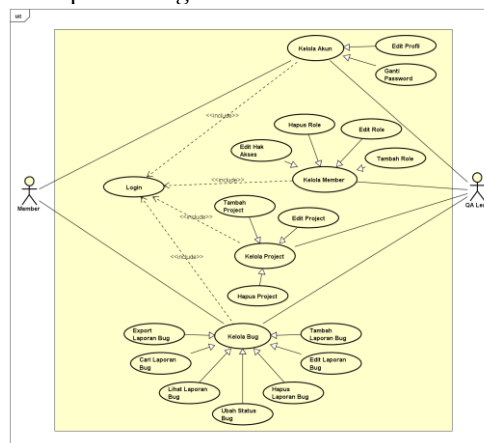
Pemeliharaan melibatkan pemantauan kinerja sistem, pemecahan masalah, dan peningkatan sistem jika diperlukan. Pemeliharaan dapat mencakup perbaikan *bug*, peningkatan fungsionalitas, dan penyesuaian dengan perubahan kebutuhan penggunaannya[12].

prioritas, yang mempengaruhi efisiensi tim QA dalam menangani bug.

### 3.2. Perancangan Sistem Baru

a. *Use Case Diagram*

Diagram ini menunjukkan bagaimana pengguna seperti QA, pengembang, dan manajer proyek berinteraksi dengan sistem untuk mencatat, melacak, dan melaporkan *bug*.



Gambar 2. *Use Case Diagram* Pengelolaan Bug Berbasis Website

## 3. Hasil dan Pembahasan

### 3.1 Analisis Sistem Lama

Proses pengelolaan *bug* di PT. Javan Cipta Solusi dimulai dengan penerimaan dan analisis *bug*. Langkah kerja yang pertama yang dilakukan dalam tahapan ini adalah proses penerimaan dan pelaporan *bug* ke dalam sistem oleh QA, pengisian detail bug seperti judul, deskripsi, dan prioritas oleh QA serta analisis dan penentuan prioritas perbaikan oleh QA. Langkah kedua yaitu penugasan dan perbaikan *bug* melalui kegiatan penugasan *bug* kepada tim pengembang yang bertanggung jawab dan tahapan pengembangan perbaikan *bug* oleh Tim Pengembang. Langkah ketiga adalah pengujian dan validasi *bug* dengan melalui kegiatan pengujian untuk memastikan kualitas perbaikan oleh QA. Serta verifikasi oleh QA dan validasi oleh pelapor *bug* atau pemilik fitur terkait.

Berdasarkan analisis yang telah dilakukan terhadap sistem pengelolaan *bug* di PT. Javan Cipta Solusi, dapat disimpulkan bahwa sistem pengelolaan *bug* yang berjalan saat ini, meskipun telah menggunakan QA Touch sebagai alat manajemen pengujian utama, masih memiliki beberapa kekurangan, di antaranya adalah:

a. Metode Pencatatan dan Pelaporan

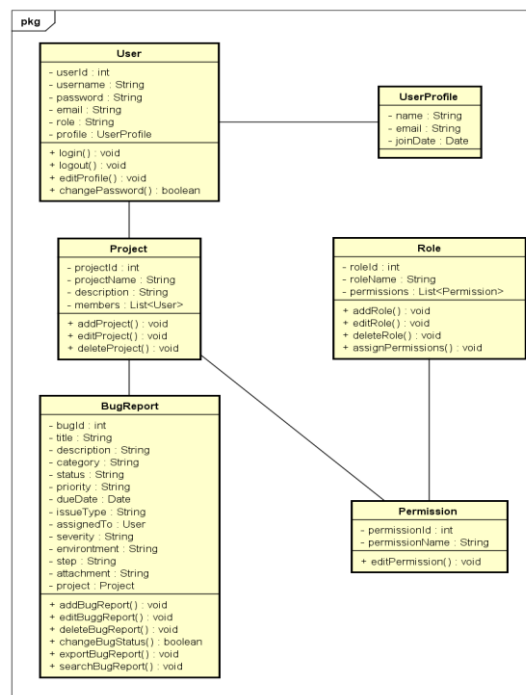
Proses pencatatan dan pelaporan *bug* saat ini tidak terfokus, sehingga mengurangi efisiensi dalam penanganan bug.

b. Pengawasan dan Manajemen Bug

Pengawasan dan manajemen *bug* saat ini masih mengalami kendala dalam hal kejelasan status dan

b. *Class Diagram*

Menunjukkan struktur kelas dari sistem yang akan dibangun, termasuk atribut dan metode dari setiap kelas serta hubungan antar kelas. Diagram ini membantu dalam memahami bagaimana data dan fungsi diorganisasikan dalam sistem.



Gambar 3. *Class Diagram* Pengelolaan Bug Berbasis Website

c. Relasi Tabel

Menggambarkan hubungan antara tabel-tabel dalam basis data. Dengan adanya relasi tabel, data dalam database dapat terstruktur dengan baik sehingga memudahkan pengelolaan dan akses data. Berikut adalah relasi tabel pada sistem yang diusulkan

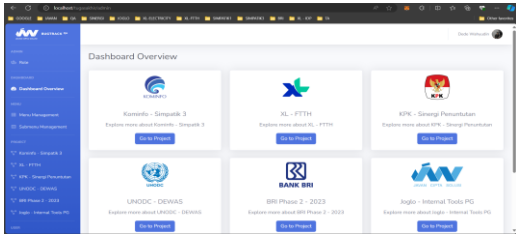


Gambar 4. Relasi Tabel Pengelolaan Bug Berbasis Website

3.3 Implementasi Fitur

Merupakan tahapan untuk merealisasikan rancangan dialog yang telah dibuat. Tahap ini mencakup beberapa komponen yang memungkinkan pengguna untuk berinteraksi dan melakukan berbagai aksi dalam aplikasi pengelolaan bug. Berikut adalah implementasi antarmuka dari aplikasi ini.

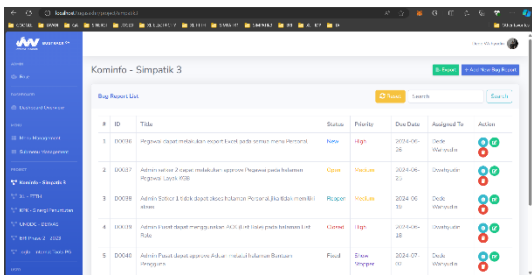
1) Dashboard



Gambar 5. Halaman Dashboard

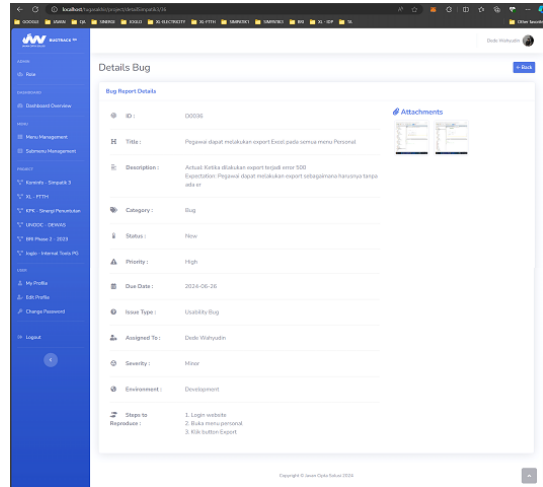
Tampilan ini memberikan daftar proyek yang sedang aktif dalam sistem. Informasi yang disajikan meliputi nama proyek dan opsi untuk mengeksplorasi lebih lanjut tentang proyek tersebut.

2) Bug Report



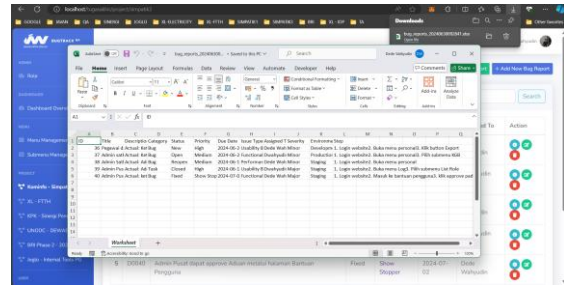
Gambar 6. Halaman Bug Report

Form Bug Report digunakan untuk melaporkan bug yang ditemukan dalam sistem. Form ini mencakup berbagai informasi yang diperlukan untuk mendeskripsikan bug secara lengkap dan akurat



Gambar 7. Halaman Details Bug

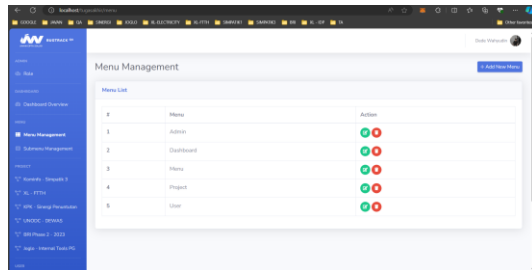
Detail Bug memberikan informasi mendalam mengenai suatu bug tertentu. Halaman ini muncul ketika pengguna mengklik salah satu data dari laporan bug.



Gambar 8. Export Bug Report

Fitur ini memungkinkan pengguna untuk mengekspor laporan bug yang terdeteksi dalam sistem ke dalam format Excel. Fitur ini dirancang untuk memudahkan analisis dan pelaporan bug secara offline.

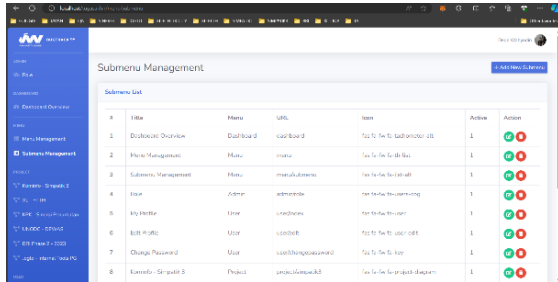
3) Menu Management



Gambar 9. Halaman Menu Management

Form Menu digunakan untuk menambahkan atau mengedit menu dalam sistem. Ini membantu dalam pengelolaan navigasi sistem.

#### 4) Submenu Management



Gambar 10. Halaman Submenu Management

Form Submenu digunakan untuk menambahkan atau mengedit submenu dalam sistem. Ini membantu dalam pengelolaan navigasi sistem di bawah menu utama.

### 3.4. Pengujian Sistem

#### 3.4.1. Black Box Testing

Pengujian menggunakan metode *black box testing*, yaitu melakukan validasi *output* apakah sudah sesuai yang diharapkan dari data *input* yang diberikan, supaya jumlah *test case* tidak terlalu banyak, maka jumlah *test case* perlu dibatasi dengan tidak menghilangkan *test case* yang memiliki potensi untuk menemukan kesalahan[13]. Berikut adalah tabel pengujian beserta hasil pengujiannya.

Tabel 1. Hasil Pengujian Black Box

No	Nama Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Dashboard	Login website Pengelolaan Bug Akses Dashboard	Menampilkan halaman Dashboard	Berhasil
2.	Bug Report	Buka menu Project Pilih Project terkait	Menampilkan halaman Bug Report	Berhasil
		Klik tombol "Add New Bug Report" Isi dan lengkapi form Klik tombol "Submit Report"	Berhasil tambah Bug Report	Berhasil
		Klik aksi "Edit" Isi dan lengkapi form Klik tombol "Update"	Berhasil edit Bug Report	Berhasil
		Klik aksi "Delete" Klik tombol "Ok"	Berhasil hapus Bug Report	Berhasil
		Klik aksi "Detail"	Menampilkan detail Bug Report	Berhasil
		Klik tombol "Export"	Berhasil export Bug Report	Berhasil
3.	Menu Management	Buka menu "Menu" Pilih submenu "Menu Management"	Menampilkan halaman Menu Management	Berhasil
		Klik tombol "Add New Menu" Isi dan lengkapi form Klik tombol "Add"	Berhasil tambah Menu	Berhasil
		Klik aksi "Edit" Isi dan lengkapi form Klik tombol "Update"	Berhasil edit Menu	Berhasil
		Klik aksi "Delete" Klik tombol "Ok"	Berhasil hapus Menu	Berhasil
4.	Submenu Management	Buka menu "Menu" Pilih submenu "Submenu Management"	Menampilkan halaman Submenu Management	Berhasil
		Klik tombol "Add New Submenu" Isi dan lengkapi form Klik tombol "Add"	Berhasil tambah Submenu	Berhasil
		Klik aksi "Edit" Isi dan lengkapi form Klik tombol "Update"	Berhasil edit Submenu	Berhasil
		Klik aksi "Delete" Klik tombol "Ok"	Berhasil hapus Submenu	Berhasil

#### 3.4.2. Pengujian Data Kuantitatif

Kegiatan analisis data dalam penelitian kuantitatif meliputi pengolahan dan penyajian data, melakukan berbagai perhitungan untuk mendeskripsikan data, dan melakukan analisis untuk menguji hipotesis[14]. Pengujian kuantitatif dalam penelitian ini berupa simulasi berbasis uji coba yang dilakukan secara internal pada tim QA. Dengan hasil pengujian yang tampak pada tabel 2, tabel 3 dan tabel 4.

#### a. Efektivitas Penanganan dan Kejelasan Prioritas

Tabel 2. Data Kuantitatif Menggunakan QA Touch (tanpa penguatan proses)

Indikator	QA Touch (tanpa penguatan proses)
Bug yang tidak diproses sesuai prioritas (%)	21%
Status bug tidak diperbarui tepat waktu (%)	18%
Tim QA tidak mencatat bug kecil/tidak kritis	13%
Bug macet di status "Open" > 3 hari	24%
Bug tidak lengkap deskripsi teknis (%)	16%
Pembuatan laporan QA mingguan	±15 menit

QA Touch sudah menyediakan fitur-fitur canggih, namun jika tidak diikuti SOP yang ketat atau budaya tim

QA yang terarah, maka pelaporan bisa tidak fokus, tidak rapi, dan progres *bug* tidak jelas.

Tabel 3. Data Kuantitatif Sistem Yang Dibangun Dengan Alur Dan Validasi Fokus

Indikator	Sistem Internal
Bug diproses sesuai prioritas (%)	92%
Status selalu diperbarui otomatis/manual	100% (wajib update)
Semua bug, termasuk minor, tercatat	100% (form wajib isi)
Bug tidak mandek >3 hari (%)	Hanya 8%
Deskripsi bug lengkap (%)	100%
Laporan mingguan	Otomatis, <7 menit

Sistem yang dibangun sesuai kebutuhan tim dan proyek internal, sehingga setiap alur kerja QA dapat disesuaikan dengan SOP lokal. Hal ini membuat sistem lebih fokus, dan lebih cepat dalam penanganan masalah nyata di lapangan.

#### b. Fokus Penanganan Bug

Tabel 4. Perbandingan Data Kuantitatif Fokus Penanganan Bug

Aspek	QA Touch (tanpa SOP kuat)	Sistem Internal
Bug tidak sesuai prioritas (%)	21%	8%
Status tidak diperbarui tepat waktu (%)	18%	0%
Deskripsi bug tidak lengkap (%)	16%	0%
Bug macet > 3 hari (%)	24%	8%
Proses pelaporan QA mingguan (durasi)	±15 menit	±7 menit
Cakupan pencatatan semua jenis bug (%)	87%	100%

Tabel perbandingan menunjukkan bahwa sistem yang dibangun lebih unggul dalam konsistensi pelaporan dan pengawasan bug. Dengan tingkat kelengkapan pencatatan 100%, status yang selalu terbaru, dan waktu pelaporan yang lebih singkat, sistem ini lebih efektif mendukung tugas QA secara menyeluruh. Sementara QA *Touch* masih menunjukkan kelemahan pada aspek keterlambatan pembaruan status dan ketidaksesuaian penanganan bug dengan prioritas, terutama bila tidak disertai SOP yang disiplin. Hal ini menegaskan bahwa fungsionalitas sistem internal lebih terfokus dan sesuai dengan kebutuhan operasional tim.

### 3.5. Evaluasi Fungsionalitas

Hasil pengujian menggunakan metode *Black Box* menunjukkan bahwa seluruh fungsi utama dalam sistem informasi pengelolaan bug berbasis website telah berjalan sebagaimana mestinya. Setiap fitur memberikan keluaran yang sesuai dengan masukan yang diberikan, tanpa ditemukan kesalahan fungsional. Dengan demikian, sistem dinyatakan telah berfungsi secara optimal dari sisi fungsionalitas dan siap untuk dilanjutkan ke tahap implementasi lebih lanjut seperti uji coba di lingkungan pengguna sebenarnya.

Metode *Black Box Testing* telah terbukti efektif dalam mengevaluasi fungsionalitas sistem berbasis web,

termasuk kemampuan menangani rentang input kritis[15] serta memastikan keluaran sesuai ekspektasi pengguna[16].

Berdasarkan simulasi internal terhadap fitur pelacakan status, penentuan prioritas, dan pembuatan laporan bug, sistem informasi pengelolaan bug berbasis web menunjukkan efisiensi waktu antara 70% hingga 85% dibandingkan metode manual. Hal ini membuktikan bahwa sistem mampu mempercepat proses pengelolaan bug secara signifikan dan mendukung pengambilan keputusan teknis secara lebih tepat dan efisien.

Berdasarkan data kuantitatif, sistem yang dibangun menunjukkan fungsionalitas yang baik. Proses pencatatan menjadi lebih cepat, hanya memerlukan rata-rata 25 detik, dan pelaporan mingguan dapat dihasilkan secara otomatis dalam waktu kurang dari 7 menit.

Dalam hal pengawasan, sistem ini mampu menjaga akurasi status dan prioritas bug. Sebanyak 92% bug ditangani sesuai prioritas, dan hanya 8% bug yang tertunda lebih dari tiga hari. Hal ini menunjukkan bahwa sistem tidak hanya mendukung efisiensi, tetapi juga meningkatkan transparansi dan akuntabilitas proses QA. Jika dibandingkan dengan QA *Touch*, sistem yang dibangun mungkin tidak memiliki fitur sebanyak itu, namun lebih unggul dalam kesederhanaan, kepatuhan alur, dan relevansi terhadap kebutuhan tim.

### 4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai perancangan Sistem Informasi Pengelolaan Bug Berbasis Website untuk PT. Javan Cipta Solusi, dapat disimpulkan sebagai berikut.

- Sistem informasi yang dirancang telah berhasil memenuhi kebutuhan tim *Quality Assurance* (QA) dalam hal pencatatan dan pelaporan *bug* yang lebih terfokus dan sistematis. Dengan sistem ini, tim QA dapat mencatat dan melaporkan *bug* dengan lebih cepat dan efisien.
- Sistem ini dilengkapi dengan fitur-fitur utama yang mendukung pengawasan dan manajemen *bug*, seperti pelacakan status *bug*, prioritas *bug*, dan laporan analisis *bug*. Hal ini telah meningkatkan transparansi dan kejelasan status *bug*, sehingga tim QA dapat mengelola dan menyelesaikan *bug* dengan lebih baik.

Untuk rencana pengembangan ke depan agar penggunaan sistem pengelolaan *bug* ini dapat di maksimalkan, diantaranya:

- Menambahkan fitur-fitur sesuai dengan kebutuhan pengguna seperti fitur notifikasi instan dengan memanfaatkan teknologi *push notification* atau integrasi ke aplikasi komunikasi telegram
- Menerapkan protokol keamanan yang lebih ketat, termasuk enkripsi data dan otentikasi multi-faktor, untuk melindungi data sensitif dari akses yang tidak sah.

- c. Uji coba sistem secara langsung akan dilakukan dengan melibatkan pengguna dari berbagai divisi yang ada di PT. Javan Cipta Solusi guna mengumpulkan masukan lebih beragam yang dapat dijadikan dasar penyempurnaan sistem ke depan. [9]

56, 2023.  
F. Sinlae, I. Maulana, F. Setiyansyah, and M. Ihsan, "Pengenalan Pemrograman Web : Pembuatan Aplikasi Web Sederhana Dengan PHP dan MYSQL," *J. Siber Multi Disiplin*, vol. 2, no. 2, pp. 68–82, 2024.

A. I. Ramdhani *et al.*, "PROGRAM APLIKASI INCOMING QUALITY CONTROL CHECKING PROCESS PADA PT . KATOLEC INDONESIA," *Informatics, Sci. Technol. J.*, vol. 12, no. 1, 2022.

I. Ali, H. Fernandy, and N. Fauziyyah, "PENGEMBANGAN SISTEM INFORMASI PENJAMINAN MUTU INTERNAL BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER 4," *DEVICE*, vol. 14, no. 2, pp. 243–250, 2024.

C. Ningki and P. Noviyanti, "Implementasi Aplikasi Penjualan Produk Tradisional Berbasis Website Menggunakan Metode Waterfall," *INFORMATIK*, vol. 19, no. 2, pp. 107–114, 2023.

J. Teknologi, S. Informasi, K. Tgd, S. D. Pratama, and M. N. Dadaprawira, "Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence Dan Boundary Value Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD," *J. Teknol. Sist. Inf. dan Sist. Komput. TGD*, vol. 6, pp. 560–569, 2023.

Sofwatillah, "Tehnik Analisis Data Kuantitatif Dan Kualitatif Dalam Penelitian Ilmiah," *J. Genta Mulia*, vol. 15, no. 2, pp. 79–91, 2024.

C. Edrea, E. Sony, and A. Susanto, "Feasibility Analysis of Bengkel Koding Website Using Black Box Testing and Boundary Value Analysis," *Sinkron*, vol. 8, no. 2, pp. 764–776, 2024.

M. Zen and M. D. P. Ananda, "BULLETIN OF COMPUTER SCIENCE RESEARCH Implementasi dan Pengujian Menggunakan Metode BlackBox Testing Pada Sistem Informasi Tracer Study," *Bull. Comput. Sci. Res.*, vol. 4, no. 4, pp. 327–340, 2024, doi: 10.47065/bulletincsr.v4i4.359.

## Daftar Rujukan

- [1] S. I. Mandala, L. Fitriani, and A. Sutedi, "Rancang Bangun Aplikasi Pengelolaan Bug Tracker Berbasis Web pada Pengembangan Software," *Algoritma*, vol. 19, no. 1, pp. 189–200, 2022, [Online]. Available: <https://jurnal.itg.ac.id/index.php/algoritma/article/view/1046/860>
- [2] S. Sri Wahyuningsih, "Identifikasi Atribut Tingkat Lebih Tinggi untuk Prediksi Umur Bug Identify Higher Level Attributes for Bug Age Prediction," *kolaboratif Sains*, vol. 6, no. 3, pp. 164–180, 2023, [Online]. Available: <https://jurnal.unismuhpalu.ac.id/index.php/JKS>
- [3] Y. Jordan, V. Handrianus Pranatawijaya, and W. Widiatry, "Analisis Kualitas Dan Penerapan Software Quality Assurance Pada Aplikasi Dana Menggunakan Model Iso/Iec 9126," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 4, pp. 7687–7691, 2024, doi: 10.36040/jati.v8i4.10048.
- [4] B. Respiar, "Peran Penting Software Quality Assurance Dalam Pengembangan Aplikasi," *Sci. Mandalika*, vol. 5, no. 12, pp. 535–540, 2024, [Online]. Available: <https://www.ojs.cahayamandalika.com/index.php/jomla/article/view/4038>
- [5] B. K. Pratama, "Pengembangan Learning Management System (Lms) Berbasis Android," *RJTI*, vol. 4, no. 1, pp. 24–34, 2025, [Online]. Available: <https://journal.upp.ac.id/index.php/rjti/article/view/3254/1760>
- [6] F. Khalid, "Rancang Bangun Sistem Informasi Absensi Dan Kegiatan Magang Berbasis Web ( Studi Kasus : Badan Pendapatan Daerah Provinsi Riau )," *RJTI*, vol. 3, no. 3, pp. 61–66, 2024.
- [7] P. Studi, S. Informasi, F. Teknologi, and U. Battuta, "Pemodelan Sistem Penerimaan Anggota Baru dengan Unified Modeling Language ( UML ) ( Studi Kasus : Programmer Association of Battuta )," *Minfo Polgan*, vol. 12, no. 1, pp. 1514–1521, 2023.
- [8] A. Usman and A. Budiman, "Rancang Bangun Desain UI / UX Pada Pembuatan Startup Aplikasi Selfcare Berbasis Website," *Ilmu Komput. dan Sist. Inf.*, vol. 2, no. 1, pp. 42–