

## **INTERNAL TESTING RESULT MANAGEMENT SYSTEM (Studi Kasus : PT. Solmit Bangun Indonesia)**

Rini Nuraini Sukmana<sup>1</sup>, Abdurrahman<sup>2</sup>, Karina Khairiyah<sup>3</sup>

<sup>1,2,3</sup>STMIK BANDUNG

Sekolah Tinggi Manajemen Informatika dan Komputer Bandung  
JL. Cikutra no 113, Bandung – Jawa Barat, INDONESIA

<sup>1</sup>[rnurainisukmana@gmail.com](mailto:rnurainisukmana@gmail.com), <sup>2</sup>[mr.indonesian@gmail.com](mailto:mr.indonesian@gmail.com), <sup>3</sup>[karinatyyah@gmail.com](mailto:karinatyyah@gmail.com),

---

### **Abstrak**

Salah satu tahapan yang penting dalam proses pengembangan perangkat lunak adalah tahapan pengujian. Pengujian pada perangkat lunak dilakukan dengan tujuan untuk memastikan fungsi – fungsi pada perangkat lunak telah bekerja dengan baik. PT. Solmit Bangun Indonesia sebagai suatu perusahaan yang bergerak dibidang jasa pelayanan teknologi informasi memiliki kewajiban kepada kliennya untuk memberikan pelayanan terbaik dengan menghasilkan suatu produk perangkat lunak yang sesuai dengan kebutuhan. Salah satu upaya yang dilakukan untuk memastikan produk perangkat lunak telah berjalan dengan baik dan sesuai kebutuhan adalah dengan diterapkannya suatu sistem yang dapat mengelola dan memonitor *bugs* pada perangkat lunak. *Internal testing result management system* merupakan suatu sistem yang membantu pengelolaan dan monitoring data penyelesaian isu – isu (*bugs*) pada perangkat lunak baik yang masih dalam tahap pengembangan maupun yang telah diimplementasikan.

Metodologi penelitian yang digunakan dalam melaksanakan penelitian ini adalah metodologi berorientasi objek dengan menggunakan *Unified Modeling Language (UML)* sebagai bahasa pemodelan. Untuk pengembangan sistemnya menggunakan metode *prototype*. Bahasa pemrograman yang digunakan dalam perancangan sistem ini adalah *PHP Hypertext Preprocessor* yang dibangun dengan menggunakan *framework* codeigniter (CI) dengan *database MySQL*.

Hal yang telah tercapai pada penelitian ini yaitu, telah dibuatnya suatu sistem yang membantu dalam pengelolaan dan monitoring penyelesaian isu – isu (*bugs*) pada perangkat lunak di PT. Solmit Bangun Indonesia

**Kata kunci:** *Bugs, Pengujian Perangkat Lunak, Sistem Manajemen*

---

### **Abstract**

*One of the most important stages in the software development process is the testing phase. Tests on the software are performed with the aim of ensuring the functions of the software are working properly. PT. Solmit Bangun Indonesia as a company engaged in information technology services has obligations to it is clients to provide the best service by producing a software product that suits the needs. One effort made to ensure the software product has been running properly and as needed is the implementation of a system that can manage and monitor bugs in the software. Internal testing result management system is a system that helps the management and monitoring of data resolving issues (bugs) on software both in development stage and that have been implemented.*

*The methodology used in this study is the method of by using Unified Modeling Language (UML) as a modeling language. For the development of the system using the prototype method. The programming language used in the design of this system is PHP Hypertext Preprocessor which is built using the framework code igniter (CI) with MySQL database.*

*The thing that has been achieved in this research is has made a system that assist in management and monitoring of solution of issues (bugs) in software at PT. Solmit Bangun Indonesia.*

*Keyword: Bugs, Management System, Software Testing*

---

## 1. Pendahuluan

Pengujian perangkat lunak bertujuan untuk memastikan fungsi – fungsi pada perangkat lunak bekerja dengan baik. Pengujian juga dilakukan untuk mencari kesalahan yang mungkin terjadi pada perangkat lunak.

Dalam penerapan sistem informasi manajemen proyek, sebelum perangkat lunak diimplementasikan, perangkat lunak tersebut harus melalui proses pengetesan internal dan setelah perangkat lunak tersebut diimplementasikan atau digunakan oleh pengguna (*user*), perangkat lunak tersebut juga akan melalui tahapan pemeliharaan (*maintenance*) yang dimana pada tahapan ini pengguna (*user*) memastikan efektifitas operasional dari sistem. Baik pada tahapan pengetesan internal maupun pada tahapan pemeliharaan tidak menutup kemungkinan pada perangkat lunak yang telah dibangun tersebut terdapat isu – isu (*bugs*) yang ditemukan. Isu – isu (*bugs*) tersebut perlu dimonitoring dan ditindaklanjuti sampai dengan perangkat lunak dapat berfungsi sesuai dengan kebutuhan pengguna (*user*).

Dari beberapa kebutuhan yang telah dianalisis, maka diperlukan suatu sistem informasi untuk melakukan proses *monitoring* dan tindak lanjut isu – isu (*bugs*) perangkat lunak di Solmit dapat terorganisir dengan baik.

### 1.1 Rumusan Masalah

Adapun permasalahan yang muncul berdasarkan uraian latar belakang diatas, maka dapat dirumuskan sebagai berikut:

1. Bagaimana membangun sistem yang dapat menyimpan laporan isu – isu (*bugs*) dalam sebuah *database* internal yang keamanannya terjamin?
2. Bagaimana merancang suatu format pencatatan isu – isu (*bugs*) yang tepat untuk menyeragamkan *output* tindaklanjut antara isu yang satu dengan yang lainnya?
3. Bagaimana mengetahui sejauh mana isu – isu (*bugs*) telah selesai ditindaklanjuti?

### 1.2 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk merancang dan mengimplementasikan *internal testing result management system* di PT. Solmit Bangun Indonesia yang diharapkan mampu:

1. Menyimpan laporan isu – isu (*bugs*) dalam sebuah *database* internal yang keamanannya terjamin.
2. Merancang suatu format pencatatan isu – isu (*bugs*) yang tepat untuk menyeragamkan *output* tindaklanjut antara isu yang satu dengan yang lainnya.
3. Mengetahui sejauh mana isu – isu (*bugs*) telah selesai ditindaklanjuti.

### 1.3 Roadmap Penelitian

Agar penelitian yang dilakukan dapat terarah, maka diperlukan *roadmap* penelitian sebagai bahan penelitian agar penelitian yang dilakukan lebih fokus dan lebih baik.

1. Analisis dari jurnal yang pertama, dengan judul "Pendekatan Model *Information Technology Infrastructure Library* (ITIL) Dalam Rekayasa Sistem Informasi Untuk Pencapaian *IT-Governance*". [7]

Pengelolaan *incident & problem management* pada infrastruktur Teknologi Informasi (TI) dengan *framework* ITIL di organisasi dapat menunjukkan kepastian bahwa suatu masalah telah diperbaiki, mencegah terjadinya kembali masalah yang sama dan melakukan perawatan serta pencegahan untuk mengurangi masalah - masalah yang pernah terjadi muncul kembali, serta dapat memberikan *service* terbaik bagi pengguna layanan TI terkait *incident & problem management*.

2. Analisis dari jurnal yang kedua, dengan judul "Analisis Tata Kelola Teknologi Informasi Dengan *Best Practice* ITIL V3 *Service Operation* (Studi Kasus: Pustaka dan Arsip Kampar)". [5]

Kesepakatan layanan dalam ITIL *Service Design* disebut dengan *service level agreement* atau disingkat SLA.

Kategorisasi *service level* diperlukan untuk memetakan kritikalitas dari beragam produk dan layanannya, paling tidak menjadi 3 kategori penting, yaitu: *critical*, *medium* dan *low priority*.

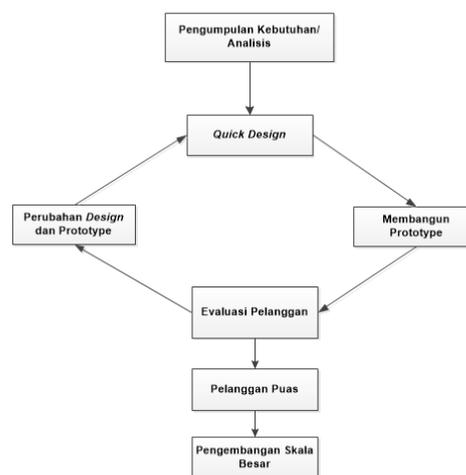
3. Analisis dari jurnal yang ketiga, dengan judul "Analisis Layanan Teknologi Informasi Pada Proses Bisnis Akademik Perguruan Tinggi XYZ". [8]

Tujuan dari analisis layanan teknologi informasi adalah menentukan

fungsional layanan TI, kualitas layanan TI, sumber daya TI, dan kemampuan TI penyedia layanan dalam bentuk dokumen katalog layanan TI.

### 1.4 Metodologi Penelitian

Metode pengembangan perangkat lunak yang digunakan dalam perancangan *internal testing result management system* di PT. Solmit Bangun Indonesia adalah dengan metode *prototype*:



Gambar 1.1 Model *Prototype* [4]

Berikut ini merupakan tahapan dari paradigma *Prototype*:

#### 1. Pengumpulan Kebutuhan/ Analisis

*Prototype* pertama dimulai dengan mengumpulkan kebutuhan-kebutuhan dari perangkat lunak yang akan dibangun. Pada pengumpulan kebutuhan ini peneliti berdiskusi dengan pihak Solmit untuk menentukan kebutuhan – kebutuhan perangkat lunak dan mendefinisikan gambaran umum perangkat lunak yang akan dirancang.

#### 2. Perancangan (*Quick Design*)

Perancangan berisi serangkaian proses desain yang dilakukan secara cepat oleh peneliti untuk memberikan gambaran

atas aspek – aspek perangkat lunak yang akan dibangun.

### 3. Membangun *Prototype*

Proses pembangunan *prototype* perangkat lunak mengacu perancangan desain pada tahapan sebelumnya. Pembangunan *prototype* ini terdiri dari perancangan basis data dan penulisan kode program pada perangkat lunak yang dibangun.

### 4. Evaluasi Pelanggan

Fase selanjutnya setelah pembangunan *prototype* adalah dilakukannya pengujian atau evaluasi oleh pengguna (Solmit). Pengujian dilakukan dengan *black box testing*. Dari hasil pengujian ini, apabila perangkat lunak sudah sesuai dengan yang diharapkan maka akan lanjut ke fase implementasi sistem, namun apabila masih terdapat ketidaksesuaian maka tahapan *prototype* akan kembali lagi pada fase *quick design* (fase nomor 2).

### 5. Pelanggan Puas

Pelanggan (pihak Solmit) merasa puas dengan perangkat lunak yang dibangun, karena telah sesuai dengan yang diharapkan dan perangkat lunak siap diimplementasikan.

### 6. Pengembangan Skala Besar

Dari *prototype* yang telah dibangun dapat menjadi dasar untuk pengembangan perangkat lunak dalam skala yang lebih besar. [5]

## 2. Overview Sistem

*Internal testing result management system* merupakan sebuah sistem yang dirancang untuk membantu perusahaan yang bergerak di bidang jasa proyek pengembangan perangkat lunak dalam mengelola hasil pengetesan isu – isu (*bugs*) aplikasi yang bersumber dari internal perusahaan maupun dari klien atau pengguna. Dalam pengelolaannya *internal*

*testing result management system* dapat menghitung persentase penyelesaian dan tindak lanjut perbaikan isu – isu (*bugs*) secara otomatis dan menampilkan *dashboard* status penyelesaian perbaikan isu – isu (*bugs*). Selain itu, terdapat pula kategori isu – isu (*bugs*) yang nantinya akan menjadi acuan dalam menentukan batas waktu (*due date*) penyelesaian perbaikan isu – isu (*bugs*).

### 2.1 Pengujian Perangkat Lunak

Menurut Muslihudin dan Oktafianto didalam bukunya tentang Analisis Perancangan Sistem Informasi Menggunakan Metode Terstruktur dan UML, “Pengujian perangkat lunak adalah proses mengeksekusi sistem perangkat lunak untuk menentukan apakah perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan sesuai dengan lingkungan yang diinginkan”<sup>[1]</sup>.

Jadi, Pengujian perangkat lunak bertujuan untuk memastikan fungsi – fungsi pada perangkat lunak bekerja dengan baik.

### 2.2 Isu

Menurut Prayudi dalam jurnalnya yang berjudul “Manajemen Isu dan Tantangan Masa Depan Pendekatan Public Relations”, isu muncul ketika ada ketidaksesuaian antara pengharapan publik dengan praktek organisasi yang jika diabaikan bisa berdampak merugikan bagi organisasi. Isu bisa meliputi masalah, perubahan peristiwa, situasi, kebijakan atau nilai <sup>[2]</sup>.

### 2.3 Manajemen

Menurut Soegoto dalam bukunya yang berjudul “*Entrepreneurship* Menjadi Pebisnis Ulung”, untuk membangun suatu bisnis setiap orang harus memshsmi tentang manajemen dan fungsi – fungsi dasar manajemen <sup>[3]</sup>.

Manajemen adalah proses perencanaan, pengorganisasian, pengarahan dan pengawasan sumber daya organisasi untuk mencapai tujuan organisasi.

### 2.4 Manajemen Isu

Manajemen isu merupakan proses proaktif dalam mengelola isu – isu, tren atau peristiwa potensial, eksternal dan internal yang memiliki dampak baik negatif maupun positif terhadap organisasi dan menjadikan isu sebagai peluang meningkatkan reputasi perusahaan [2].

### 2.5 Software Quality Assurance

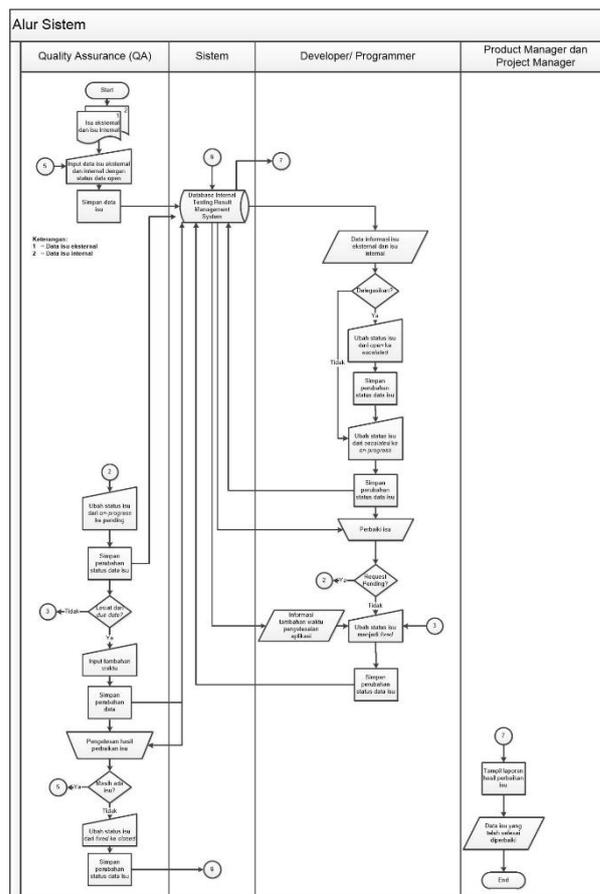
Jaminan kualitas perangkat lunak (*Software Quality Assurance/SQA*) adalah aktifitas pelindung yang diaplikasikan kepada seluruh proses perangkat lunak [6].

Analisis terhadap dokumen bisnis, pembuatan skenario *testing* dan memahami *scope* adalah bagian penting yang harus dilakukan QA sejak awal suatu proyek dimulai.

## 3. Analisis Sistem

### 3.1 Analisis Alur Sistem

Prosedur yang diusulkan pada aplikasi *internal testing result management system* dapat digambarkan sebagai berikut.



Gambar 3.1 Alur Sistem Yang Diusulkan

### 3.2 Kategori Isu

Kategori isu – isu (*bugs*) ini nantinya akan menjadi acuan dalam menentukan batas waktu (*due date*) penyelesaian perbaikan isu – isu (*bugs*).

Tabel 3.1 Kategori Isu

Kategori Isu	Keterangan	Response Time	Resolution Time	Closing Time
Low	Sistem dapat digunakan, akan tetapi gejala kesalahan ditemukan namun tidak terlalu mengganggu dan tidak mempengaruhi operasi aplikasi.	240 Menit	1440 Menit	4320 Menit
Medium	Sistem dapat digunakan dengan gangguan minor	180 Menit	1440 Menit	2880 Menit

Kategori Isu	Keterangan	Response Time	Resolution Time	Closing Time
	(dengan catatan kondisi tersebut tidak terlalu mengganggu operasi aplikasi.			
High	Sistem tidak dapat beroperasi sebagian (yang mengakibatkan kritis sehingga tidak dapat dilanjutkannya suatu proses/flow pada aplikasi.	120 Menit	60 Menit	240 Menit

- a *Response Time*  
Waktu paling lama bagi Tim *Programmer* untuk mulai merespon dan atas adanya isu – isu (*bugs*) yang mereka terima, dihitung sejak terjadinya laporan dan/atau gangguan.
- b *Resolution Time*  
Waktu paling lama bagi Tim *Programmer* untuk menganalisa permasalahan dan menemukan solusinya, dihitung sejak terjadinya laporan dan/atau gangguan.
- c *Closing Time*  
Waktu paling lama bagi Tim *Programmer* untuk mengembalikan kondisi perangkat lunak serta waktu bagi QA untuk memastikan bahwa perangkat lunak berfungsi dan berjalan dengan baik, dihitung sejak terjadinya laporan dan/atau gangguan.

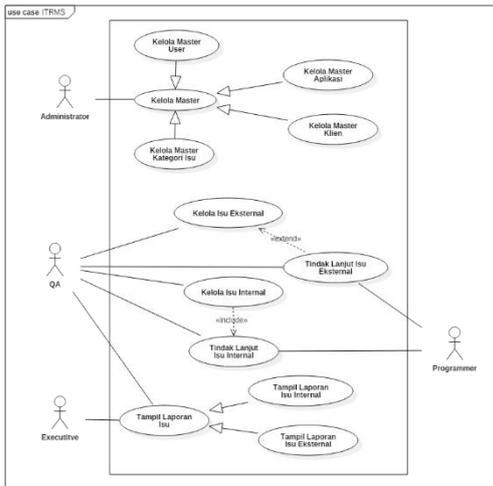
Setiap proses tindak lanjut dan perbaikan isu – isu (*bugs*) yang dilakukan akan tercatat secara otomatis oleh sistem dengan menampilkan *history* perubahan status proses tindak lanjut isu – isu (*bugs*). Status – status yang ditampilkan di dalam sistem adalah sebagai berikut.

Tabel 3.2 Tipe – Tipe Status Tindak Lanjut Isu

No	Nama Status	Keterangan
1	<i>Open</i>	Merupakan status yang diberikan oleh QA ketika ditemukannya isu baik itu isu eksternal maupun isu internal.
2	<i>On-Progress</i>	Merupakan status yang diberikan oleh <i>Programmer</i> ketika isu sudah mulai dilakukan perbaikan.
3	<i>Fixed</i>	Merupakan status yang diberikan oleh <i>Programmer</i> ketika isu telah selesai dilakukan perbaikan dan siap dites oleh QA.
4	<i>Closed</i>	Merupakan status yang diberikan oleh QA terhadap hasil perbaikan isu yang sudah dilakukan pengetesan dan tidak ditemukan lagi permasalahan.
5	<i>Escalated</i>	Merupakan status yang diberikan oleh QA ketika ada perubahan <i>Programmer</i> yang sebelumnya telah didelegasikan.
6	<i>Request Pending</i>	Merupakan status yang diberikan oleh <i>Programmer</i> ketika ada isu yang tidak dapat diselesaikan karena adanya permasalahan seperti perbaikan yang dilakukan memerlukan waktu lebih untuk koordinasi dengan PIC.
7	<i>Pending</i>	Merupakan status yang diberikan oleh QA ketika menyetujui data isu yang statusnya <i>Request Pending</i> .
8	<i>Request For Extra Time</i>	Merupakan status yang diberikan oleh <i>Programmer</i> ketika proses penyelesaian isu melebihi <i>due date</i> .
9	<i>Add extra Time</i>	Merupakan status yang diberikan oleh QA ketika akan menambahkan waktu lebih dalam proses penyelesaian isu yang telah melebihi <i>due date</i> .
10	<i>Canceled</i>	Merupakan status yang diberikan oleh QA ketika ada isu yang diterima baik dari pihak eksternal maupun internal namun isu tersebut merupakan penambahan fitur diluar dari <i>scope</i> proyek.

### 3.3 Usecase Diagram

Berikut ini merupakan pemodelan dari aplikasi menggunakan *use case*.



Gambar 3.2 Use Case Diagram

4. Hasil dan Pembahasan

4.1 Rencana Pengujian

Pengujian yang digunakan dalam penelitian ini menggunakan metode pengujian *black box* dan skenario kasus uji data normal, dimana metode pengujiannya dibagi kedalam beberapa prototipe.

Tabel 4.1 Tabel Pengujian Prototipe I

Kelas Uji	Butir Uji	Deskripsi
Login	Login administrator	Pengecekan keberhasilan login.
	Login QA	
	Login programmer	
	Login executive	
Data Master	Master aplikasi	Pengujian terhadap keberhasilan proses tambah, ubah dan hapus data.
	Master klien	
	Master user	
Isu Eksternal	Data isu eksternal	Pengujian terhadap keberhasilan proses penambahan, perubahan, penghapusan dan pengajuan data isu eksternal serta perubahan status tindak lanjut data isu.
	Tindak lanjut isu eksternal	
Isu Internal	Data isu eksternal	Pengujian terhadap keberhasilan proses penambahan, perubahan, penghapusan dan pengajuan data isu eksternal serta perubahan status tindak lanjut data isu.

Tabel 4.2 Tabel Pengujian Prototipe II

Kelas Uji	Butir Uji	Deskripsi
Data Master	Master kategori isu	Pengujian terhadap penambahan master data yang dibutuhkan dari hasil usulan perbaikan

Kelas Uji	Butir Uji	Deskripsi
		pada pengujian prototipe tahap I. Pengujian dilakukan untuk menguji keberhasilan proses tambah, ubah dan hapus data.
Isu Eksternal	Data isu eksternal	Pengujian terhadap keberhasilan proses penambahan, perubahan, penghapusan dan pengajuan data isu eksternal serta perubahan status tindak lanjut data isu menggunakan batas waktu yang sesuai dengan master kategori isu.
	Tindak lanjut isu eksternal	
Isu Internal	Data isu eksternal	Pengujian terhadap keberhasilan proses penambahan, perubahan, penghapusan dan pengajuan data isu internal serta perubahan status tindak lanjut data isu menggunakan batas waktu yang sesuai dengan master kategori isu.

Tabel 4.3 Tabel Pengujian Prototipe III

Kelas Uji	Butir Uji	Deskripsi
Laporan	Monitoring isu eksternal	Pengujian terhadap kesesuaian format laporan monitoring isu serta keberhasilan proses menampilkan data laporan dan mencetak data laporan.
	Monitoring isu internal	
	Dashboard isu eksternal	Pengujian terhadap keberhasilan proses menampilkan data laporan dan mencetak data laporan.
	Dashboard isu internal	Pengujian terhadap keberhasilan proses menampilkan data laporan dan mencetak data laporan.

Tabel 4.3 Tabel Pengujian Prototipe IV

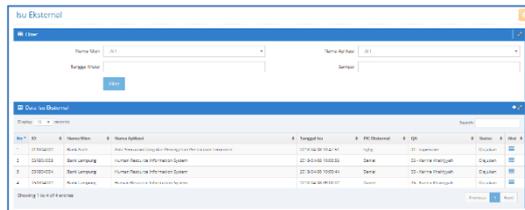
Kelas Uji	Butir Uji	Deskripsi
Laporan	Monitoring isu eksternal	Pengujian terhadap kesesuaian format laporan monitoring isu dari hasil usulan perbaikan pada pengujian prototipe di tahap III.
	Monitoring isu internal	

4.2 Tampilan Aplikasi

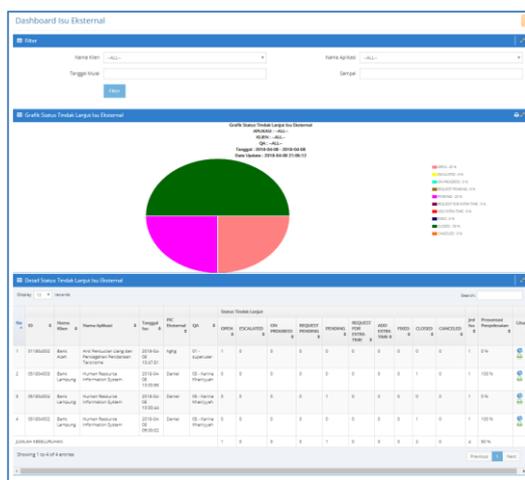
Berikut beberapa *interface* yang terdapat dalam aplikasi *internal testing result management system*.



Gambar 4.1 Tampilan Login



Gambar 4.2 Tampilan Halaman Isu



Gambar 4.3 Tampilan Dashboard

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari berbagai penjelasan yang telah diuraikan dalam laporan penelitian ini, telah dirancang dan diimplementasikan *internal testing result management system* di PT. Solmit Bangun Indonesia yang mampu:

- Menyimpan laporan isu – isu (*bugs*) dalam sebuah *database* internal yang keamanannya terjamin.
- Memiliki format pencatatan isu – isu (*bugs*) yang mampu menyeragamkan

*output* tindaklanjut antara isu yang satu dengan yang lainnya.

- Mengukur sejauh mana isu – isu (*bugs*) telah ditindaklanjuti

### 5.2 Saran

Penulis menyadari bahwa pada pembuatan aplikasi *internal testing result management system* masih memiliki banyak kekurangan. Adapun saran yang dapat diambil guna pengembangan sistem lebih lanjut sebagai berikut:

- Diharapkan adanya pengembangan lagi yang dimana pihak klien Solmit dapat mengakses aplikasi *internal testing result management system* secara langsung untuk mengetahui perkembangan perbaikan dari isu – isu (*bugs*) yang dilaporkan.
- Memungkinkan adanya pengiriman notifikasi tindaklanjut isu – isu (*bugs*) melalui SMS Gateway.

### Daftar Pustaka

- Muslihudin, Oktafianto, 2016, *Analisis Perancangan Sistem Informasi Menggunakan Metode Terstruktur dan UML*, p. 98. Penerbit Andi.
- Prayudi, 2007, *Manajemen Isu dan Tantangan Masa Depan: Pendekatan Public Relations*, v.4, p.26, Jurnal Ilmu Komunikasi.
- Soegoto, 2014, *Entrepreneurship Menjadi Pebisnis Ulung*, p.143, Penerbit Elex Media Komputindo.
- Yunindra, 2017, *Software Engineering*, p.50, Penerbit Deepublish.
- Miata, Akmal, 2016, *Analisis Tata Kelola Teknologi Informasi Dengan Best Practice ITIL V3 Service*

- Operaton (Studi Kasus: Pustakan dan Arsip Kampar)*, v.2. Diperoleh dari: <http://ejournal.uin-suska.ac.id/index.php/RMSI/article/view/1852> [Diakses pada 19/02/2018 pukul 17.05 WIB].
- [6] Shofiah, Shofah, *Analisis Kualitas Perangkat Lunak Terhadap Sistem Informasi STT Wastukencana Purwakarta*, Diperoleh dari: <https://www.stt-wastukencana.ac.id/jurnal/download/5.-ARTIKEL-SQA-SHOFA-15-finish.pdf> [Diakses pada 19/05/2018 pukul 20.51 WIB].
- [7] Surbakti, 2014, *Pendekatan Model Information Technology Infrastructure Library (ITIL) Dalam Rekayasa Sistem Informasi Untuk Pencapaian IT-Governance*”, v.VII. Diperoleh dari: <http://jti.respati.ac.id/index.php/jurnaljti/article/viewFile/52/45> [Diakses pada 19/02/2018 pukul 17.10 WIB].
- [8] Sutomo, Erwin, 2017, *Analisis Layanan Teknologi Informasi Pada Proses Bisnis Akademik Perguruan Tinggi XYZ*, v.2. Diperoleh dari: <http://publications.aisindo.org/index.php/JSII/article/view/61/32> [Diakses pada 26/02/2018 pukul 21.35 WIB].