

PENGEMBANGAN SISTEM *MONITORING* PENDATAAN APLIKASI BERBASIS WEB PADA KEMENTERIAN PEKERJAAN UMUM DAN PERUMAHAN RAKYAT

Salma Nada Safira^{*1}, Yusi Tyroni Mursityo², Mochamad Chandra Saputra³

^{1,2,3}Universitas Brawijaya, Malang

Email: ¹salmanadasafira@student.ub.ac.id, ²yusi_tyro@ub.ac.id, ³andra@ub.ac.id

^{*}Penulis Korespondensi

(Naskah masuk: 04 Januari 2023, diterima untuk diterbitkan: 26 September 2023)

Abstrak

Kementerian Pekerjaan Umum dan Perumahan Rakyat adalah kementerian Indonesia yang mempunyai tugas menyelenggarakan kegiatan pemerintahan di bidang pekerjaan umum dan perumahan rakyat untuk membantu Presiden. Perangkat lunak yang dimiliki oleh Kementerian PUPR sangat banyak, sehingga membutuhkan *monitoring* pendataan aplikasi. Pada pemantauan aplikasi di kementerian PUPR masih menggunakan Google Workspace, proses tersebut menyebabkan harus melakukan pengecekan setiap aplikasi dan menjadi tidak efisien. Sehingga, membutuhkan adanya solusi dengan pengembangan sistem *monitoring* aplikasi. Aktivitas pada sistem *monitoring* dengan melacak status aplikasi, dan melakukan pengelolaan data perangkat lunak. Metode *Rational Unified Process* (RUP) digunakan untuk membuat aplikasi. Pendekatan dilakukan menggunakan pengembangan berorientasi objek dengan UML. Proses implementasi sistem menggunakan kerangka kerja Laravel. Hasil penelitian ini adalah sistem *monitoring* aplikasi berbasis *website*. Sistem yang berhasil dibangun, kemudian melalui proses pengujian sistem. Pengujian unit dilakukan dengan *white box testing* yang menghasilkan status uji valid berdasarkan pada semua *path*. Pengujian validasi dengan *black box testing* menunjukkan bahwa semua fitur valid. *User Acceptance Testing* menghasilkan nilai akhir 94,67% sehingga sudah sesuai dengan kebutuhan pengguna. Kontribusi sistem terhadap organisasi dapat memperbaiki efisiensi pada *monitoring* aplikasi dan meningkatkan efektivitas, karena data tersimpan di dalam *database*.

Kata kunci: *sistem monitoring, Rational Unified Model (RUP), perangkat lunak, Laravel, pengujian sistem*

DEVELOPMENT OF WEB-BASED APPLICATION DATA COLLECTION MONITORING SYSTEM AT THE MINISTRY OF PUBLIC WORKS AND PEOPLE'S HOUSING

Abstract

Assisting the President in managing government activities pertaining to public works and public housing is the responsibility of the Ministry of Public Works and Public Housing in Indonesia. The Ministry of PUPR owns a considerable amount of software, so it requires monitoring application data collection. In monitoring applications at the PUPR ministry, they still use Google Workspace. This process causes them to have to check each application, which becomes inefficient. Thus, it requires a solution through the development of an application monitoring system. Monitoring activities that can be carried out on this system can track the status of application and manage software data. This application is designed using the Rational Unified Process (RUP) method. The approach is carried out using object-oriented development with UML. The system implementation process uses the Laravel framework. The result of this research is a website-based application monitoring system. The system that is successfully built then goes through the system testing process. Unit testing is done by white box testing, which produces a valid test status based on all paths. Validation testing with black box testing shows that all features are valid. User Acceptance Testing produces a final value of 94.67% so that it is in accordance with user needs. The contribution of the system to the organization can improve efficiency in application monitoring and increase effectiveness because the data is stored in the database.

Keywords: *monitoring system application, Rational Unified Model (RUP), software, Laravel, system testing*

1. PENDAHULUAN

Kementerian Pekerjaan Umum dan Perumahan Rakyat (PUPR) merupakan kementerian Indonesia yang mempunyai tugas menyelenggarakan urusan pemerintahan di bidang pekerjaan umum dan perumahan rakyat. Perangkat lunak yang dimiliki oleh kementerian PUPR sangat banyak, sehingga membutuhkan *monitoring* pendataan aplikasi. Pada pemantauan status aplikasi di kementerian PUPR masih menggunakan Google Workspace, sehingga proses tersebut menyebabkan harus melakukan pengecekan satu persatu aplikasi untuk mengetahui status aplikasi dan server. Hal tersebut menyebabkan proses menjadi lama, tidak otomatis terdeteksi, dan tidak efisien. Oleh karena itu, membutuhkan adanya solusi dengan pengembangan sistem monitoring pendataan aplikasi. Penelitian tentang sistem monitoring aplikasi yang mampu untuk *monitoring* dan *controlling* server dan dapat memeriksa koneksi status server serta memeriksa aplikasi. Sehingga aplikasi yang dibangun dapat memudahkan *system administrator* untuk mengawasi kinerja server (Haykal and Siswanto, 2018).

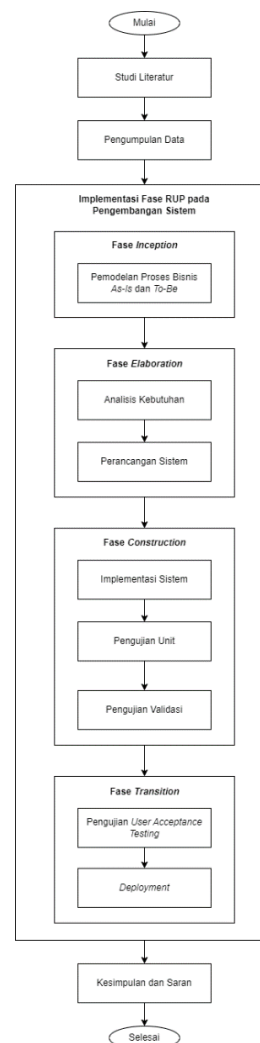
Dari penelitian tersebut, dapat menyimpulkan bahwa dengan membuat sistem *monitoring* yang baik itu memerlukan penggunaan sebuah sistem berbasis web sehingga dapat dengan mudah untuk *monitoring* aplikasi setiap, serta efektif dan efisien. *Monitoring* merupakan aktivitas yang bertindak secara sistematis untuk dapat mengontrol, mengawasi suatu kinerja (Habibi and Karnovi, 2020). Berdasarkan kebutuhan pada Kementerian PUPR dalam adanya sistem *monitoring* pendataan aplikasi, aktivitas yang dilakukan pada sistem, diantaranya dapat melacak (*track*) status server aplikasi, *SSL certification date* pada aplikasi, melihat status aplikasi yang aktif, tidak aktif, serta memerlukan pengembangan, dapat menampilkan data-data yang mendukung pada pendataan perangkat lunak, dapat mengelola data dan file agar tersimpan di *database*. Web server merupakan layanan protokol HTTP atau HTTPS (Husen and Surbakti, 2020), dari akses web server dapat mendeteksi status *up* atau *down*. *SSL certification* memiliki arti dengan sesi komunikasi yang terhubung dengan jaringan internet yang aman (Huda, 2020), *SSL certificate* untuk mengetahui tanggal terakhir SSL web tersebut.

Proses pengembangan sistem *monitoring* aplikasi dilakukan dengan menggunakan metode *Rational Unified Process* (RUP). Dimana memiliki arti berupa suatu pendekatan pada implementasi sistem yang bersifat iteratif (Perwitasari, Afwani and Anjarwani, 2020). Pemodelan kebutuhan menggunakan *Unified Modeling Language* (UML). Selanjutnya dilakukan proses pengujian terlebih dahulu agar aplikasi dapat beroperasi dengan benar. Pada pengujian perangkat lunak, menggunakan pengujian unit, validasi dan *User Acceptance Testing* (UAT) untuk mengetahui respon pengguna terhadap perangkat lunak yang berhasil dibangun sudah sesuai

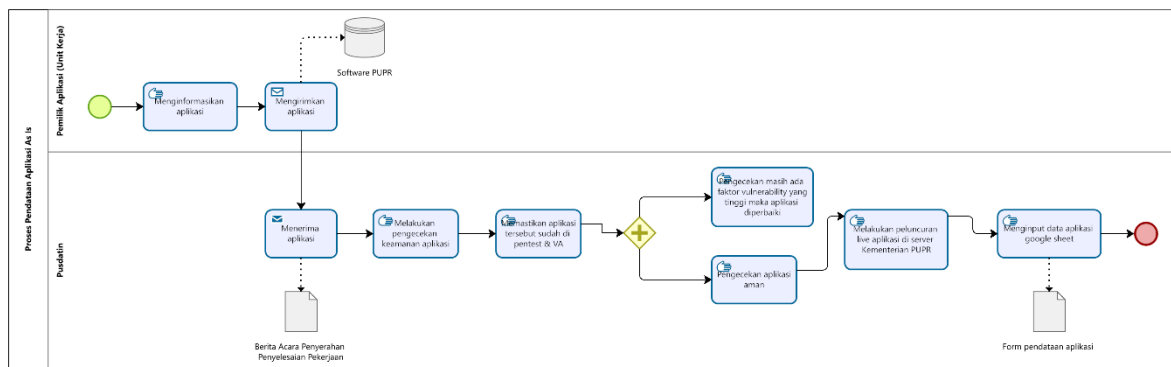
dengan kebutuhan. Diharapkan sistem ini dapat membantu *monitoring* pendataan aplikasi yang terdapat di Kementerian PUPR. Penelitian ini bertujuan untuk melakukan perancangan sistem *monitoring* pendataan aplikasi sesuai dari analisis kebutuhan. Kemudian mengimplementasikan sistem perangkat lunak *monitoring* pendataan aplikasi berbasis web sehingga dapat mempersingkat waktu dalam pengoperasiannya dan mengetahui hasil pengujian setelah membuat sistem *monitoring* pendataan berbasis web agar sesuai dengan kebutuhan dan berjalan dengan efektif.

2. METODE PENELITIAN

Metode *Rational Unified Process* (RUP) digunakan pada penelitian ini. Bab ini memberikan gambaran tentang tahapan penelitian, dimulai dengan mempelajari literatur, dilanjutkan pengumpulan data melalui wawancara dan observasi, pemodelan proses bisnis, analisis kebutuhan, perancangan sistem, implementasi sistem, dan pengujian. Berikut ini merupakan alur penelitian yang dapat diamati pada Gambar 1.



Gambar 1. Alur Penelitian



Gambar 2. Proses Bisnis As-Is

Prosedur pada penelitian mengikuti alur sebagai berikut:

1. Studi literatur
Mempelajari literatur melalui *e-book*, jurnal nasional dan internasional, buku, serta penelitian terdahulu sebagai bahan acuan.
2. Pengumpulan data
Wawancara dan observasi dengan *stakeholder* kementerian PUPR.
3. Fase *inception*
Pada tahap *inception*, peneliti melakukan pemodelan proses bisnis *as-is* dan *to-be*.
4. Fase *elaboration*
Melakukan analisis kebutuhan dengan membuat daftar kebutuhan fungsional dan kebutuhan non fungsional.
5. Fase *construction*
Melakukan implementasi sistem dengan pengkodean sistem, kemudian pengujian unit dan pengujian validasi.
6. Fase *transition*
Menerapkan pengujian UAT dengan melibatkan pengguna berdasarkan aktor dari sistem dan melakukan deployment.
7. Kesimpulan dan saran
Menemukan hasil penelitian serta memberikan rekomendasi untuk penelitian selanjutnya.

3. HASIL DAN PEMBAHASAN

3.1. Pemodelan Proses Bisnis

Pemodelan proses bisnis termasuk dalam fase *inception*. Pemodelan proses bisnis dilakukan berdasarkan dari hasil wawancara untuk menggambarkan tahapan aktivitas yang terjadi di dalam perusahaan. Proses bisnis dilakukan untuk menggambarkan aktivitas yang terjadi di dalam perusahaan sesuai kebutuhan dalam mencapai *goal* (Prasojo, 2022).

3.1.1. Proses Bisnis As-Is

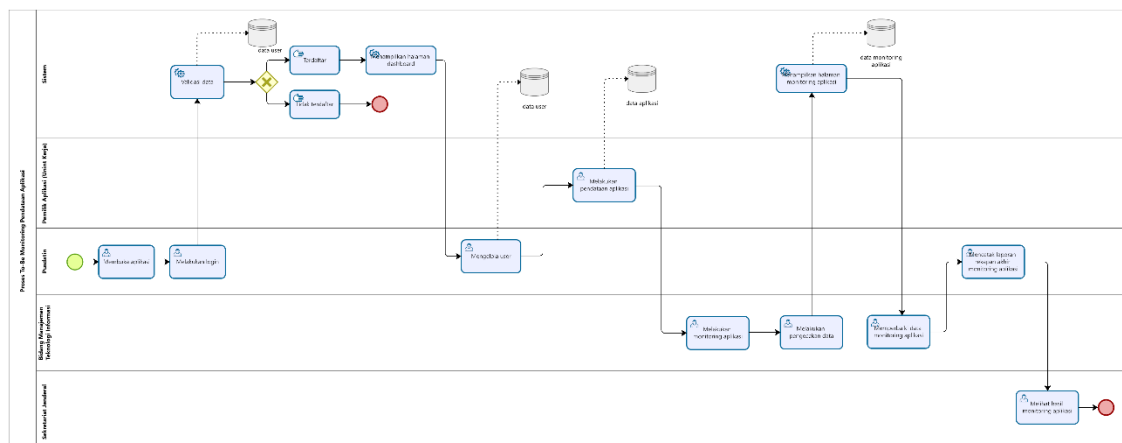
Gambar 2. menjelaskan proses bisnis *as-is* untuk pemodelan proses bisnis monitoring aplikasi yang sedang berjalan di Kementerian PUPR. Pusdatin melakukan monitoring aplikasi secara manual dengan pengecekan aplikasi untuk memeriksa aplikasi. Jika aplikasi sudah aman, maka aplikasi dapat *live* di server Kementerian PUPR. Kemudian melakukan pendataan aplikasi melalui Google Sheet.

3.1.2. Proses Bisnis To-Be

Gambar 3. merupakan proses bisnis *to-be* atau proses bisnis usulan, dengan memanfaatkan sistem *monitoring*. Sehingga dapat memberikan solusi dari proses bisnis *as-is* yang terjadi. Pada sistem *monitoring* juga dapat mencetak laporan rekapan.

3.2. Analisis Kebutuhan

Tahap ini termasuk pada fase *elaboration*. Untuk analisis kebutuhan diawali dengan



Gambar 3. Proses Bisnis To-Be

mengidentifikasi aktor yang terlibat dalam sistem, membuat diagram *use case* dan *activity diagram*.

3.2.1. Use Case Diagram

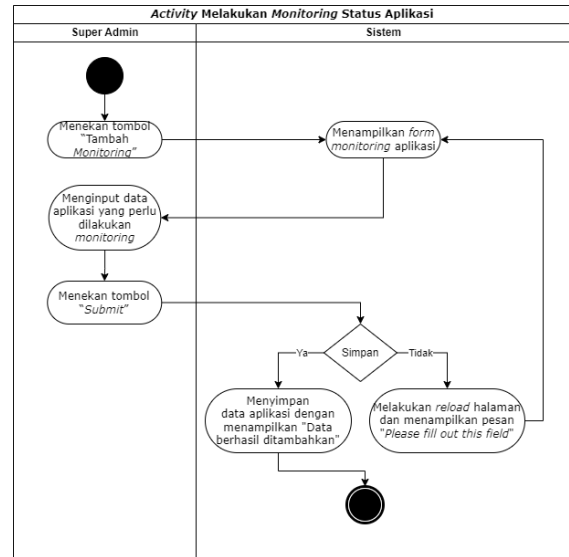
Tahap ini mengidentifikasi hubungan antara sistem dan aktornya (Yang, 2021). Dari *use case diagram* sistem *monitoring* terdapat 4 aktor. Super admin akan melakukan generalisasi kepada admin karena aktivitasnya hampir mirip dengan admin. Kemudian terdapat 18 *use case* pada *use case diagram* sistem *monitoring*. *Use case diagram* dapat diamati pada Gambar 4.



Gambar 4. Use Case Diagram

3.2.2. Activity Diagram

Proses perwakilan secara vertikal dan aliran pengembangan *use case* dilakukan pada *activity diagram* (Rahayu et al., 2021). Pada Gambar 5. merupakan *activity diagram* melakukan monitoring status aplikasi, menjelaskan aktivitas aktor dalam melakukan pengecekan terhadap aplikasi yang akan dilakukan monitoring.



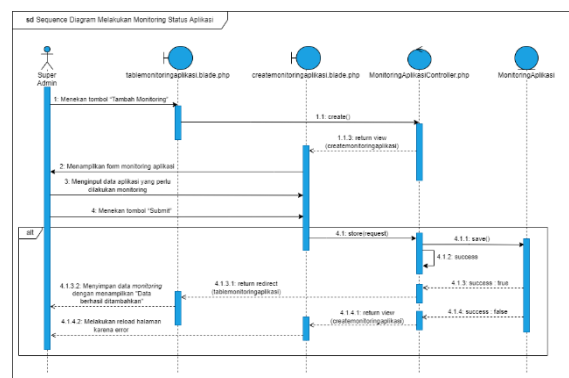
Gambar 5. Activity Diagram Melakukan Monitoring Status Aplikasi

3.3. Perancangan Sistem

Perancangan sistem termasuk dalam tahap *elaboration*. Perancangan sistem yang diimplementasikan terdiri dari perancangan arsitektur dengan *sequence diagram*. Dilanjutkan dengan perancangan basis data, dengan membuat *physical data model* (PDM). Perancangan komponen merupakan proses logis dari sistem dengan menjelaskan algoritma perangkat lunak. dan terakhir perancangan antarmuka, dengan desain layar antarmuka pengguna sebagai ilustrasi.

3.3.1. Sequence Diagram

Tindakan yang diambil sebagai respon terhadap suatu peristiwa untuk membuat keluaran dilakukan pada *sequence diagram* (Henderi, Rahardja and Rahwanto, 2021). Pada Gambar 6. merupakan *sequence diagram* melakukan monitoring status aplikasi dengan aktor super admin yang memiliki 4 objek yang saling berhubungan, dan entity Monitoring Aplikasi.

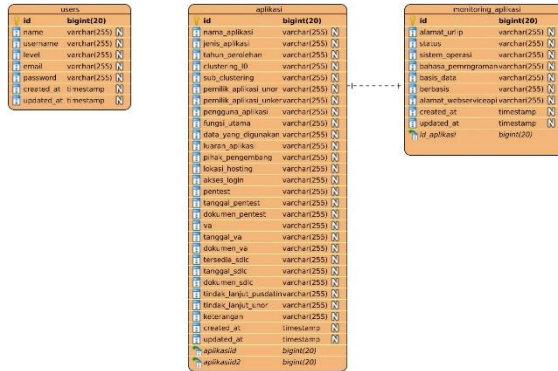


Gambar 6. Sequence Diagram Melakukan Monitoring Status Aplikasi

3.3.2. Physical Data Model (PDM)

Physical data model merupakan hubungan antara tabel dengan menggambarkan data secara fisik

berdasarkan ruang lingkup struktur data (Sylvia and Vigil, 2021). Pemodelan ini membantu dalam memvisualisasikan struktur basis data beserta relationship yang menghasilkan skema. Total tabel pada PDM yaitu berjumlah 3 tabel, diantaranya users, aplikasi, dan monitoring aplikasi. *Physical data model* dapat diamati pada Gambar 7.



Gambar 7. *Physical Data Model (PDM)*

3.3.3. Perancangan Komponen

Aktivitas untuk memproses algoritma dalam bentuk *pseudocode* sesuai dengan *method* dilakukan pada tahap ini, sebelum melakukan implementasi sistem. *Method* createMonitoringAplikasi memiliki tujuan untuk menambahkan data *monitoring* aplikasi baru. Perancangan komponen *method* createMonitoringAplikasi dapat diamati pada Tabel 1.

Tabel 1. Perancangan Komponen *Method* CreateMonitoringAplikasi

No	<i>Pseudocode Method</i> CreateMonitoringAplikasi
1	BEGIN
2	monitoringaplikasi = Request
3	
4	validate = [
5	monitoringaplikasi field
6	idAplikasi is required,
7	monitoringaplikasi field
8	alamatURL is required,
9	monitoringaplikasi field
10	status is required,
11	monitoringaplikasiData field
12	sistemOperasi is required,
13	monitoringaplikasiData field
14	bahasa Pemrograman is
15	required,
16	monitoringaplikasiData field
17	basisData is required,
18	monitoringaplikasiData field
19	berbasis is required,
20	monitoringaplikasiData field
21	alamat Webservicesapi is
22	required,
23]
24	
25	IF (validate is FALSE)
26	
27	RETURN response (Please fill
28	out this field)
29	END IF
30	monitoringaplikasi = create data
31	to database table

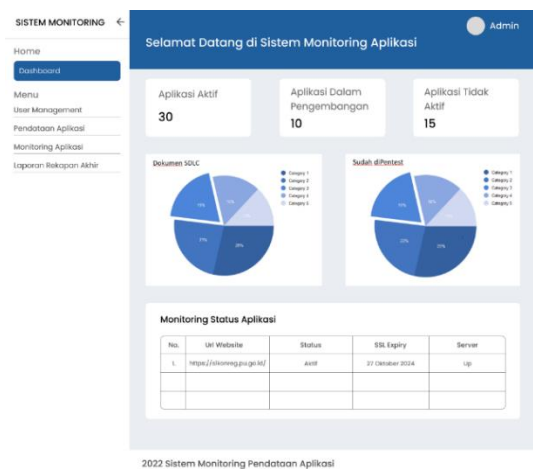
```

MonitoringAplikasi with
value(monitoringaplikasiData all
atribute)
20    RETURN response (Data Berhasil
      Ditambahkan!)
21
22    END

```

3.3.4. Perancangan Antarmuka

Suatu desain dengan memvisualisasikan komponen pada setiap halaman. Pada tahap ini melakukan perancangan antarmuka sebagai gambaran untuk implementasi sistem. Perancangan antarmuka *dashboard* merupakan halaman awal sistem monitoring pendataan aplikasi setelah user berhasil melakukan autentikasi. Perancangan antarmuka *dashboard* dapat diamati pada Gambar 8.



Gambar 8. Perancangan Antarmuka *Dashboard*

3.4. Implementasi Sistem

Proses ini termasuk dalam tahap *construction*. Implementasi sistem merupakan pembuatan sistem dengan bahasa pemrograman sesuai dari perancangan desain antarmuka. Implementasi sistem pada penelitian ini dilakukan menggunakan kerangka kerja Laravel. Proses ini melakukan implementasi kode program dan antarmuka.

3.4.1. Implementasi Kode Program

Implementasi ini dilakukan berdasarkan perancangan komponen. Proses dimulai dengan implementasi kode dalam bahasa pemrograman dengan *framework* Laravel. Implementasi kode program *method* createMonitoringAplikasi dapat diamati pada Tabel 2.

Tabel 2. Implementasi Kode Program *Method* CreateMonitoringAplikasi

No	Implementasi Data Tabel Monitoring Aplikasi
1	public function store(Request \$request)
2	{
3	\$this->validate(\$request, [
4	'id_aplikasi' =>
5	'required exists:aplikasi,
6	'id',

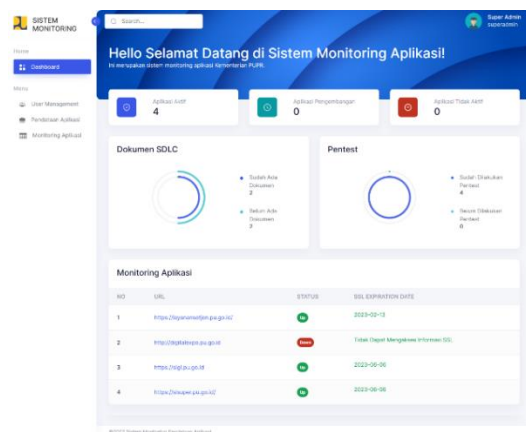

```

5      'alamat_urลิป' =>
6      'required|string',
7      'status' =>
8      'required|string|in:Aktif, Dalam
Pengembangan,Tidak Aktif',
9      'sistem_operasi' =>
10     'required|string',
11     'bahasa_pemrograman' =>
12     'required|string',
13     'basis_data' =>
14     'required|string',
15     'berbasis' =>
16     'required|in:Desktop,Web,Mobile'
17
18     'alamat_webservicesapi' =>
19     'nullable',
20
21 ];
22
23 $monitoring_aplikasi = new
MonitoringAplikasi;
24 $monitoring_aplikasi->alamat_urลิป =
$request-> alamat_urลิป;
25 $monitoring_aplikasi->status =
$request->status;
26 $monitoring_aplikasi->sistem_operasi =
$request->
sistem_operasi;
27 $monitoring_aplikasi->bahasa_pemrograman =
$request->
bahasa_pemrograman;
28 $monitoring_aplikasi->basis_data =
$request->basis_data;
29 $monitoring_aplikasi->berbasis =
$request->berbasis;
30
31 if ($request->
has('alamat_webservicesapi') &&
$request->
32     alamat_webservicesapi != '') {
33     $monitoring_aplikasi->alamat_webservicesapi =
34     $request->alamat_webservicesapi;
35 }

```

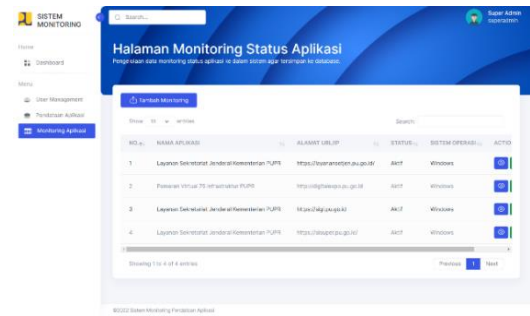
3.4.2. Implementasi Antarmuka

Pada implementasi antarmuka yaitu melakukan implementasi berdasarkan perancangan antarmuka. Tahap implementasi antarmuka menjelaskan implementasi yang dilakukan berdasarkan dari penulisan *source code* dengan bahasa pemrograman. Gambar 9. merupakan implementasi antarmuka dashboard, untuk menampilkan halaman awal atau beranda pada sistem monitoring setelah user berhasil melakukan autentikasi.



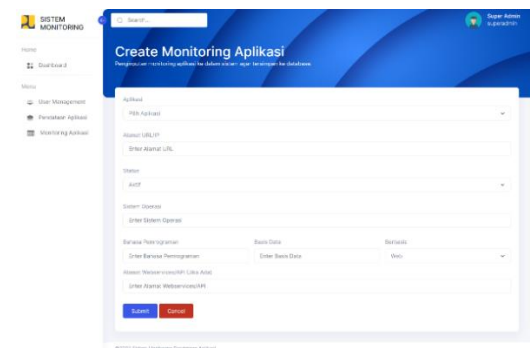
Gambar 9. Implementasi Antarmuka Dashboard

Implementasi antarmuka *monitoring* status aplikasi dapat diamati pada Gambar 10. Implementasi ini digunakan untuk menampilkan daftar *monitoring* aplikasi. Terdapat beberapa komponen pada halaman *monitoring* aplikasi, yaitu sidebar sebagai menu pada sistem *monitoring*. Bagian body terdapat tombol untuk tambah *monitoring* dan data tabel yang menunjukkan daftar *monitoring* aplikasi yang berhasil diinput dan tersimpan di *database*.



Gambar 10. Implementasi Antarmuka *Monitoring* Status Aplikasi

Implementasi antarmuka *create monitoring* aplikasi digunakan untuk membuat data *monitoring* aplikasi baru. Komponen utama pada implementasi ini yaitu form data monitoring aplikasi. Tampilan implementasi antarmuka *create monitoring* aplikasi dapat diamati pada Gambar 11.



Gambar 11. Implementasi Antarmuka *Create Monitoring* Aplikasi

3.5. Pengujian Sistem

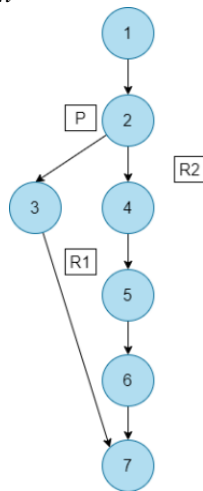
Pengujian perangkat lunak yang dirancang dapat membuktikan pada sistem yang dibuat sesuai dengan spesifikasi, desain, dan implementasi sistem. Pengujian sistem dipisah menjadi 2 fase pada metode RUP. Untuk pengujian unit dan validasi termasuk dalam fase *construction*, sedangkan untuk pengujian *User Acceptance Testing* (UAT) termasuk dalam fase *transition*.

3.5.1. Pengujian Unit

Pengujian unit dapat diartikan sebagai jenis pengujian perangkat lunak di mana masing-masing unit diuji untuk memastikan bahwa dapat berfungsi seperti yang diharapkan. Pengujian unit dilakukan dengan menggunakan *white box testing*. Pengujian

white box merupakan upaya untuk menguji perangkat lunak dengan menganalisis dan memeriksa struktur internal dan kode sistem informasi (Purba *et al.*, 2022). Pengujian ini menggunakan basis *path testing* berdasarkan *method*. Pada jurnal ini yang dicantumkan hanya *method* createMonitoringAplikasi. Untuk *flowgraph* dapat diamati pada Gambar 12.

a. *Flow Graph*



Gambar 12. *Flow Graph Method* CreateMonitoringAplikasi

b. Hasil Pengujian Unit

Hasil pengujian unit method createuser dilakukan berdasarkan dari *basis path testing* yang dilakukan, sehingga mengetahui jumlah jalur untuk melakukan pengujian unit (Shaik *et al.*, 2021). Berikut merupakan hasil pengujian unit method createMonitoringAplikasi, yang dapat diamati pada Tabel 3.

Tabel 3. Hasil Pengujian Unit *Method* createMonitoringAplikasi

Prosedur Uji	Expected Result	Result
Class Driver memanggil method createMonitoringAplikasi tanpa memasukkan data di field.	Sistem memanggil method createMonitoringAplikasi dengan melakukan validasi terhadap field yang bersifat <i>required</i> , dan menampilkan pesan <i>error</i>	Sistem memanggil method createMonitoringAplikasi dengan melakukan validasi terhadap field yang bersifat <i>required</i> , dan menampilkan pesan <i>error</i>
Class Driver memanggil method createMonitoringAplikasi dengan memasukkan data di semua field.	Sistem memanggil method createMonitoringAplikasi dengan melakukan validasi terhadap field yang bersifat <i>required</i> , dan menampilkan pesan <i>success</i>	Sistem memanggil method createMonitoringAplikasi dengan melakukan validasi terhadap field yang bersifat <i>required</i> , dan menampilkan pesan <i>success</i>

3.5.2. Pengujian Validasi

Pada pengujian validasi dilakukan untuk memastikan bahwa perangkat lunak yang dapat berfungsi sesuai dengan implementasi dan kebutuhan pengguna. Pengujian validasi yang dilakukan menggunakan pengujian *black box*. Pengujian *black box* merupakan teknik pengujian fungsionalitas program atau disebut pengujian *behavioral* (Rianto, 2021). Dalam pengujian *black box* melakukan percobaan sistem dengan menjalankan semua fitur apakah sudah sesuai dengan hasil yang diharapkan. Berikut merupakan *test case* hasil dari pengujian *black box* melakukan monitoring status aplikasi yang dapat diamati pada Tabel 4.

Tabel 4. *Test Case* Validasi Melakukan *Monitoring Status* Aplikasi

Test Code	VBT-021
Scenario	Skenario 1
Requirement Code	SRS-F-013
Objective	Pengujian untuk membuktikan bahwa sistem dapat menjalankan fungsi input <i>monitoring</i> status aplikasi.
Test Procedure	<ol style="list-style-type: none"> 1. Penguji telah berhasil masuk ke dalam sistem <i>monitoring</i>. 2. Penguji telah masuk ke dalam halaman <i>monitoring</i> aplikasi. 3. Penguji menekan tombol "Tambah <i>Monitoring</i>". 4. Penguji memasukkan data pada <i>form create monitoring</i> aplikasi dengan penuh dan benar. 5. Penguji menekan tombol "Submit".
Expected results	Sistem berhasil menyimpan data <i>monitoring</i> aplikasi dan menampilkan notifikasi data berhasil ditambah.
Result	Sistem berhasil menyimpan data <i>monitoring</i> aplikasi dan menampilkan notifikasi data berhasil ditambah.
Status	Valid

3.5.3. Pengujian *User Acceptance Testing* (UAT)

Pengujian *User Acceptance Test* (UAT) merupakan pengujian untuk mengetahui respon pengguna terhadap perangkat lunak yang berhasil dibangun (Priyatna *et al.*, 2020). Untuk mengetahui apakah sistem *monitoring* yang dirancang sudah memenuhi kebutuhan, maka dilakukan pengujian UAT. Responden dapat memilih pilihan jawaban berdasarkan daftar pernyataan yang telah dibuat. Setelah itu responden dapat memberikan penilaian terhadap pertanyaan berdasarkan *skala likert*. Berikut merupakan daftar pernyataan yang dapat diamati pada Tabel 5.

Tabel 5. Daftar Pernyataan UAT

Kategori	Definisi	Daftar Pernyataan
Usability	Pengujian yang dilakukan untuk mengevaluasi layanan dan memastikan pengguna dapat menggunakan	<ol style="list-style-type: none"> 1. Sistem <i>monitoring</i> pendataan aplikasi dapat dengan mudah digunakan. 2. Tampilan sistem <i>monitoring</i> pendataan aplikasi

	sistem dengan nyaman ketika mudah.	3. Sistem monitoring pendataan aplikasi mempermudah dalam melakukan monitoring aplikasi.
		4. Sistem monitoring pendataan aplikasi mempermudah dalam mencari informasi data aplikasi dan monitoring aplikasi.
		5. Laporan rekapan akhir mempermudah dalam melakukan pengujian.
<i>Functional correctness and completeness</i>	Pengujian dilakukan sesuai dengan spesifikasi kebutuhan dan memastikan bahwa sistem dapat berfungsi.	1. Sistem monitoring pendataan aplikasi membantu dalam pendataan aplikasi dengan baik.
<i>Confidentially and Availability</i>	<ul style="list-style-type: none"> • <i>Confidentially</i> merupakan aturan dalam membatasi hak akses informasi • <i>Availability</i> merupakan kemampuan pengguna untuk mengakses informasi yang mempunyai hak akses dan wewenang. 	1. Sistem monitoring pendataan aplikasi membantu dalam monitoring status aplikasi dengan baik. 2. Sistem monitoring pendataan aplikasi membantu dalam pembuatan laporan rekapan akhir pendataan aplikasi. 3. Sistem monitoring pendataan aplikasi dapat diakses tiap waktu asalkan sudah terkoneksi internet dan database. 4. Sistem monitoring pendataan aplikasi memerlukan login sistem sebagai akses dalam sistem untuk menjaga keamanan informasi.

Berikut merupakan perhitungan untuk *User Acceptance Testing*:

- a. Melakukan perhitungan nilai berdasarkan dari jawaban 6 responden, hasil perhitungan pilihan jawaban responden dapat diamati pada Tabel 6.

Tabel 6. Hasil Skala Likert

Jawaban	Bobot	Jumlah Jawaban (N)	Hasil (Y x N)
Sangat Setuju	5	44	220
Setuju	4	16	64
Netral	3	0	0
Tidak Setuju	2	0	0
Sangat Tidak Setuju	1	0	0
Total			284

- b. Melakukan perhitungan nilai X (Jumlah keseluruhan)

$$X = \frac{\text{Jumlah Skala Likert} \times \text{Jumlah Responden}}{\text{Jumlah Pernyataan}}$$

$$X = 5 \times 6 \times 10 = 300$$

- c. Melakukan perhitungan nilai persentase

$$\text{Rumus} = \frac{\text{Total Nilai}}{X} \times 100\%$$

$$\text{Rumus} = \frac{284}{300} \times 100\% = 94,67\%$$

Berdasarkan dari hasil perhitungan *User Acceptance Testing* dengan skala *likert* yang telah dilakukan bersama dengan enam penguji dari Kementerian PUPR, menunjukkan bahwa hasil akhir memperoleh nilai 94,67%. Dari hasil tersebut, dapat disimpulkan bahwa pada sistem *monitoring* aplikasi, responden menyatakan sangat setuju dalam membantu *monitoring* aplikasi yang ada di Kementerian PUPR dan sudah sesuai dengan kebutuhan.

3.6. Kontribusi Sistem Terhadap Organisasi

Kontribusi sistem dalam organisasi merupakan ide atau strategi yang dihasilkan setelah membuat sistem kepada organisasi. Kontribusi memiliki peran penting kepada organisasi karena dapat memberikan dampak yang optimal dan mentransformasi kinerja organisasi (Sudarmanto *et al.*, 2021).

Dalam penelitian ini, kontribusi atau peran sistem *monitoring* terhadap Kementerian PUPR adalah:

- 1) Meningkatkan inovasi organisasi karena ketika sudah menginputkan data aplikasi yang diperlukan, sistem akan langsung mendeteksi status aplikasi beserta *SSL expiration date*.
- 2) Memperbaiki efisiensi suatu aktivitas yang memerlukan waktu yang cukup lama, karena sebelumnya menggunakan Google Workspace. Dengan pembuatan sistem *monitoring* ini prosesnya lebih mudah sehingga menjadi efisien karena melibatkan suatu sistem yang sudah bersifat otomatis.
- 3) Meningkatkan efektivitas karena informasi sudah tersimpan dengan aman di sistem, sehingga proses menjadi lebih efektif dan akurat.
- 4) Meningkatkan peluang dengan kolaborasi antar sesama, karena di dalam sistem sudah terlibat masing-masing peran aktor dalam melakukan *monitoring* pendataan aplikasi.

4. KESIMPULAN

Pada penelitian ini telah berhasil mengembangkan sistem *monitoring* pendataan aplikasi berbasis web pada Kementerian PUPR. Sistem dibangun dengan menggunakan *framework*

Laravel untuk membantu proses implementasi, kemudian melakukan implementasi basis data dengan menggunakan MySQL. Dengan adanya sistem *monitoring* yang telah dikembangkan dapat membantu *stakeholder* dalam permasalahan yang ada sebelumnya, sehingga dengan menggunakan sistem dapat melakukan pemantauan status. Pada pengujian sistem dilakukan dengan pengujian unit yang mendapatkan status valid dari semua *method*. Untuk pengujian validasi menggunakan pengujian *black box*, menghasilkan nilai valid pada semua fitur sehingga sistem dapat berjalan sesuai fungsionalitas. Hasil pengujian *user acceptance testing*, menghasilkan nilai akhir 94,67%. Sehingga dapat diambil kesimpulan bahwa responden menerima sistem monitoring dengan baik dan setuju bahwa sistem dapat membantu melakukan monitoring status aplikasi yang ada di Kementerian PUPR dan sudah sesuai dengan kebutuhan.

Untuk melakukan penelitian selanjutnya pada pengembangan sistem *monitoring* aplikasi, dapat melakukan penambahan fitur *real time monitoring* seperti *log system*. Sehingga pemantauan status aplikasi dapat dilakukan secara *real time* dan mengetahui aplikasi yang memerlukan perbaikan. Kemudian fitur notifikasi yang *up to date* untuk aplikasi yang statusnya tidak aktif atau *offline* atau masih dalam pengembangan, sehingga admin dapat meninjau aplikasi tersebut lebih lanjut.

DAFTAR PUSTAKA

- HABIBI, R. H. & KARNOVI, R. 2020. Tutorial Membuat Aplikasi Sistem Monitoring Terhadap Job Desk Operational Human Capital. Jakarta: Kreatif.
- HAYKAL, A. & SISWANTO, I. M. 2018. Aplikasi Monitoring dan Controlling Server dengan Notifikasi Email Berbasis Web pada PT. Tanabe Indonesia, Jurnal SKANIKA, 1(1), pp. 193–198.
- HENDERI, RAHARDJA, U. & RAHWANTO, E. 2021. UML Powered Design System Using Visual Paradigm. Malang: Literasi Nusantara Abadi.
- HUDA, M. 2020. Keamanan Informasi. Jakarta: Nulisbuku.
- HUSEN, Z. & SURBAKTI, M. S. 2020. Membangun Server dan Jaringan Komputer dengan Linux Ubuntu. Aceh: Syiah Kuala University Press.
- PERWITASARI, R., AFWANI, R. & ANJARWANI, S. E. 2020. Penerapan Metode Rational Unified Process (RUP) dalam Pengembangan Sistem Informasi Medical Check Up pada Citra Medical Centre, Jtika, 2(1), pp. 76–88.
- PRASOJO, B. 2022. Pengenalan Dunia Bisnis untuk Umum. Yogyakarta: Victory Pustaka Media.
- PRIYATNA, B. 2020. Application of UAT (User Acceptance Test) Evaluation Model in Minggon E-Meeting Software Development, Systematics, 2(3), pp. 110–117.
- PURBA, R. A. 2022. Pengembangan Sistem Informasi: Analisis, Pemodelan, dan Perangkat Lunak. Medan: Yayasan Kita Menulis.
- RAHAYU, E. 2021. Analisis Proses Bisnis. Medan: Yayasan Kita Menulis.
- RIANTO, I. 2021. Rekayasa Perangkat Lunak. Klaten: Lakeisha.
- SHAIK, N. 2021. Methodologies for Software Testing. India: One Press Publishing.
- SUDARMANTO, E. 2021. Manajemen Strategi Kontemporer. Medan: Yayasan Kita Menulis.
- SYLVIA, M. L. & VIGIL, I. 2021. Understanding Population Health Analytics. Massachusetts: Jones & Bartlett Learning.
- YANG, H. 2021. UML Tutorials. Taipei: HerongYang.com.

Halaman ini sengaja dikosongkan