

ANALISIS METODE PENGUJIAN PERANGKAT LUNAK *BLACKBOX TESTING* DAN PEMODELAN DIAGRAM UML PADA APLIKASI *VETERINARY SERVICES* YANG DIKEMBANGKAN DENGAN MODEL *WATERFALL*

Muhamad Syarif¹⁾, Eri Bayu Pratama²⁾

^{1,2}Universitas Bina Sarana Informatika

Jl. Abdurrahman Saleh No. 18, Pontianak, Kalimantan Barat

E-mail: muhamad.mdx@bsi.ac.id¹, eri.ebp@bsi.ac.id²

ABSTRACT

Current technological developments can facilitate work in various fields, one of the information systems that need to apply computer technology is in the health sector. Veterinary services produce medical records which are an important component of important patient records that must be managed properly and should be accessed quickly. To develop a computerized system, you must use the right software development model where the waterfall model is one of the models that is often used, the results of the software that has been developed need to be tested before being implemented. Blackbox testing and use in the waterfall phase is considered to reduce the incidence of errors in functionality.

Keywords: *Waterfall, Blackbox testing, Veterinary service*

ABSTRAK

Perkembangan teknologi saat ini dapat memudahkan pekerjaan diberbagai bidang, salah satu sistem informasi yang perlu menerapkan teknologi terkomputer ialah dibidang kesehatan. *Veterinary service* menghasilkan rekam medis yang merupakan komponen penting bagi catatan penting pasien harus dikelola dengan baik dan sebaiknya dapat diakses dengan cepat. Untuk mengembangkan sistem terkomputerisasi harus menggunakan model pengembangan perangkat lunak yang tepat dimana model *waterfall* merupakan salah satu model yang sering digunakan, hasil perangkat lunak yang telah dikembangkan perlu dilakukan pengujian sebelum diimplementasikan. *Blackbox testing* dan digunakan pada fase *waterfall* dianggap mampu mengurangi resiko terjadinya kesalahan dalam fungsionalitas.

Kata Kunci: *Waterfall, Blackbox testing, Veterinary service*

1. PENDAHULUAN

Perkembangan teknologi saat ini berkembang dengan pesat, banyak sistem yang menggunakan dokumen fisik beralih ke pengelolaan data digital, penggunaan data digital terbukti lebih mudah diakses dan dapat menyajikan informasi lebih cepat. Peralihan data fisik ke bentuk digital tentunya harus didukung dengan sebuah perangkat lunak (*software*), perangkat lunak dapat dibangun oleh pengembang (*software developer*) berdasarkan permintaan konsumen (*stakeholder*).

Veterinary adalah seorang dokter yang berperan menangani hewan, baik menangani kesehatan hewan dari segala

penyakit yang diderita hewan serta berperan dalam meningkatkan kesejahteraan hewan dan kesehatan masyarakat [1]. Layanan ini mencakup diagnosa, pengobatan, dan perawatan setelahnya. Hasil pemeriksaan setiap hewannya harus didokumentasikan agar dapat dijadikan pengambilan keputusan dalam pemberian tindakan pengobatan kesehatan hewan. Sistem kesehatan hewan merupakan integrasi seluruh kegiatan aspek kesehatan hewan yang dilakukan oleh semua pihak baik pemerintah, swasta, maupun masyarakat [2].

Sistem pengelolaan dokumen pada klinik layanan *veterinary service* masih

banyak yang menggunakan media fisik, dokumen disimpan pada lemari penyimpanan ataupun ruang penyimpanan dokumen, sehingga semakin lama tentu akan semakin banyak membutuhkan ruang dan berdampak pada sulitnya melakukan pencarian dokumen tertentu, sehingga metode penyimpanan media fisik sebaiknya beralih menjadi penyimpanan dan pengelolaan data secara digital, dengan cara mengembangkan sebuah aplikasi yang dapat memenuhi semua kebutuhan layanan jasa *veterinary service*.

Metodelogi pengembangan perangkat lunak adalah sebuah pendekatan formal atau rangkaian dari tindakan untuk mengimplementasikan *System development life cycle* (SDLC) yang merupakan suatu proses pemahaman tentang bagaimana sebuah sistem informasi dapat mendukung kebutuhan bisnis, mendisain sistem, membangun dan menyajikannya kepada sebuah organisasi [3]. Ketidakjelasan ruang lingkup, kompleksitas produk, ukuran proyek, serta kemampuan *programmer*, perubahan ruang lingkup atas perubahan lainnya dapat berdampak pada penambahan waktu serta biaya pengembangan. [4].

Salah satu cara meminimalkan kesalahan pengembangan yaitu dengan cara memberikan bentuk visual *software design* yang ada ke *stakeholder*, karena bentuk visual dianggap lebih mampu memberikan pemahaman lebih cepat, sehingga meminimalkan terjadinya ketidaksesuaian antar kebutuhan dan hasil pengembangan.

Pemodelan *software design* dapat menggunakan *Unified Modelling Language* (UML) *diagram* seperti *usecase diagram* yang dapat dikomunikasikan ke *stakeholder* berdasarkan kebutuhan yang diperlukan pada setiap aktornya. Setelah pemodelan *software design* selesai dan telah memenuhi kebutuhan serta ruang lingkup sistem maka akan dilanjutkan ke fase berikutnya yaitu *implementation and unit testing*, dimana salah satu metode pengujian adalah *blackbox testing* guna

menguji fungsionalitas perangkat lunak yang dikembangkan.

2. METODELOGI PENELITIAN

2.1 Waterfall

Metode Waterfall memiliki tahapan-tahapan sebagai berikut, Sommerville dalam (Sasmito, 2017):

1. *Requirements analysis and definition*,
2. *System and software design*,
3. *Implementation and unit testing*,
4. *Integration and system testing*,
5. *Operation and maintenance*.

2.2 Unified Modelling Language (UML)

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem berorientasi obyek yang telah menjadi standar dalam industri untuk visualisasi dalam merancang dan mendokumentasikan sistem piranti lunak". Richard dalam [6]

1. *Usecase Diagram*, merupakan fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau aktor. Sugiarti dalam (Sasmito, 2017)
2. *Class diagram*, merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem[7]
3. *Activity diagram*, adalah memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Rosa dan Salahuddin dalam [5]
4. *Sequence diagram*, adalah suatu diagram interaksi yang menekankan pada pengaturan waktu dari pesan-pesan. Booch dalam [6]

2.3 Blackbox Testing

Blackbox-testing merupakan salah satu metode untuk menguji perangkat lunak yang telah dibangun, baik pengujian pada unit-unit kecil maupun hasil yang telah terintegrasi untuk menguji fungsional perangkat lunak. Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk

mengetahui apakah fungsi, masukaffn dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan[8].

3. HASIL DAN PEMBAHASAN

3.1 Requirement Analysis and Definition

Analisa dan definisi kebutuhan adalah fase awal model *waterfall* dimana fase ini terjadinya komunikasi langsung antara *stakeholder* dan *software developer*.

3.2 System and Software Design

Desain sistem dan perangkat lunak adalah fase yang merubah hasil analisa kebutuhan kedalam bentuk visual. Pemodelan menggunakan UML diagram yang terdiri dari *usecase diagram*, *class diagram* dan *activity diagram*.

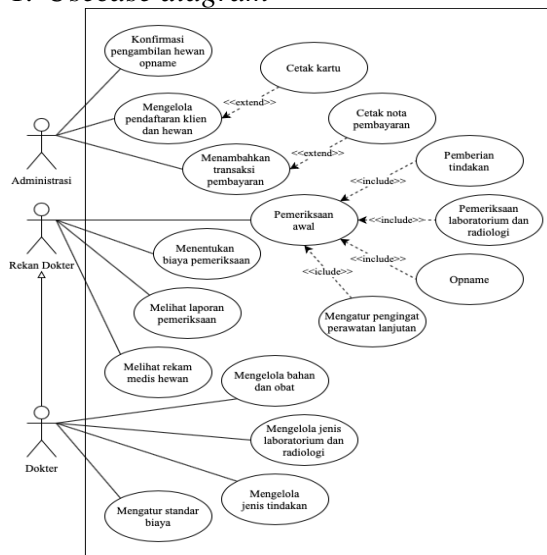
1. Inisialisasi Proyek

Tahap pertama dalam membangun *software* adalah menentukan tentang *tools* dan teknologi yang akan digunakan dalam pengembangan.

Tabel 1. Arsitektur Teknologi

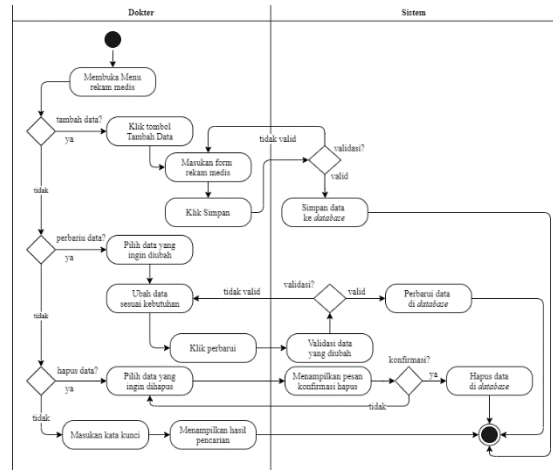
Aspek	Spesifikasi
Basis Software	Web Application
Platform	Php
Framework	CodeIgniter
Database	MariaDB
Browser	Chrome, Firefox, Edge, Safari
Device Used	Personal Computer, Tablet Android OS

1. Usecase diagram



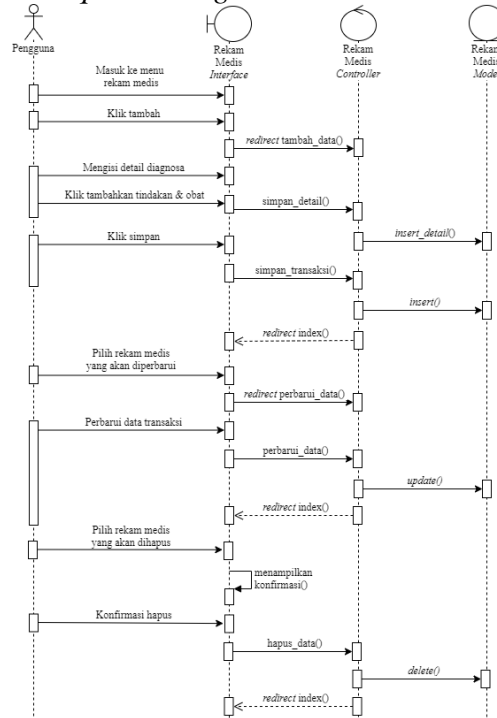
Gambar 1. Usecase diagram

2. Activity diagram



Gambar 2. Activity diagram

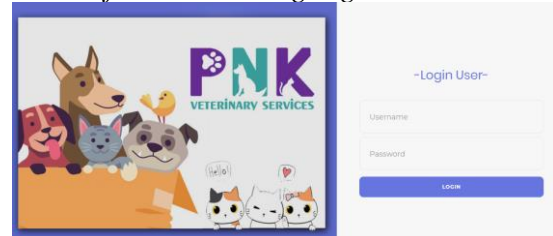
3. Sequence diagram



Gambar 3. Sequence diagram

3.3 Implementation and Unit Testing

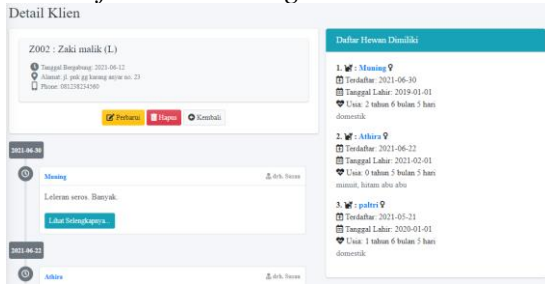
1. Interface dan testing login



Gambar 9. Halaman login

Function test	Status
Validasi	(√) succeeded ()

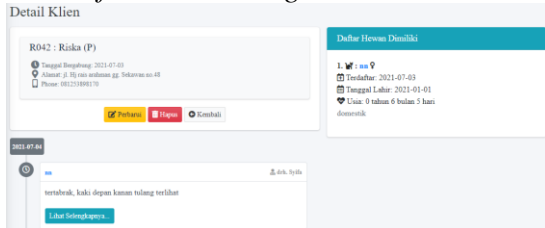
2. Interface dan testing Klien



Gambar 4. Informasi Klien

Function test	Status
Insert	(√) succeeded ()
Read	(√) succeeded ()
Update	(√) succeeded ()
Delete	(√) succeeded ()
Select hewan by klien	(√) succeeded ()
Print Id Card	(√) succeeded ()

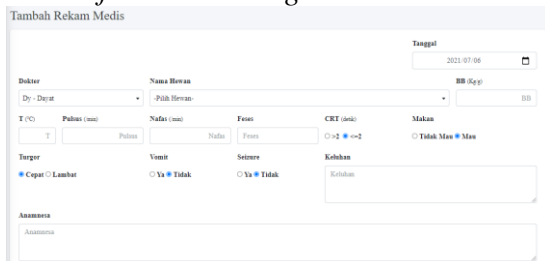
3. Interface dan testing hewan



Gambar 5. Informasi hewan

Function test	Status
Insert	(√) succeeded ()
Read	(√) succeeded ()
Update	(√) succeeded ()
Delete	(√) succeeded ()
Select rekam medis by hewan	(√) succeeded ()

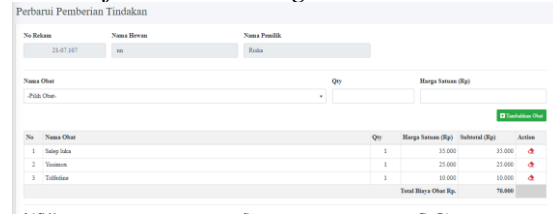
4. Interface dan testing rekam medis



Gambar 6. Informasi rekam medis

Function test	Status
Insert	(√) succeeded ()
Read	(√) succeeded ()
Update	(√) succeeded ()
Delete	(√) succeeded ()
Insert pemeriksaan fisik	(√) succeeded ()
Print by month	(√) succeeded ()
Print by dokter	(√) succeeded ()

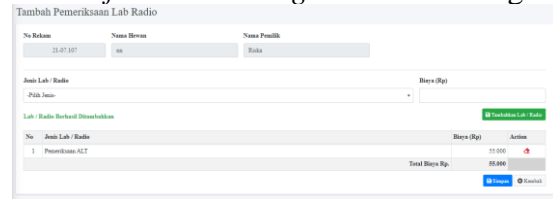
5. Interface dan testing tindakan medis



Gambar 7. Pemberian tindakan

Function test	Status
Insert	(√) succeeded ()
Read	(√) succeeded ()
Update	(√) succeeded ()
Delete	(√) succeeded ()
Insert pemberian obat	(√) succeeded ()
Insert pemberian tindakan	(√) succeeded ()
Delete pemberian obat	(√) succeeded ()
Delete pemberian tindakan	(√) succeeded ()
Upload foto tindakan	(√) succeeded ()

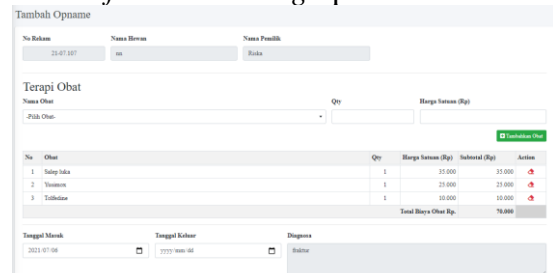
6. Interface dan testing lab dan radiologi



Gambar 8. Pemeriksaan lab dan radio

Function test	Status
Insert	(√) succeeded ()
Read	(√) succeeded ()
Update	(√) succeeded ()
Delete	(√) succeeded ()
Insert pemeriksaan lab dan radio	(√) succeeded ()
Delete pemeriksaan lab dan radio	(√) succeeded ()
Upload hasil lab dan radio	(√) succeeded ()

7. Interface dan testing opname

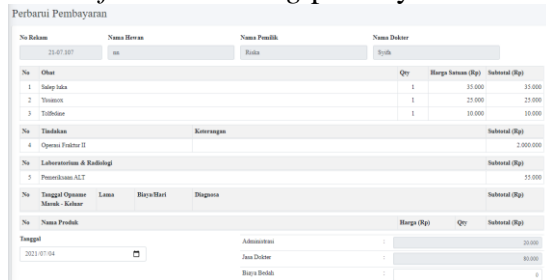


Gambar 9. Opname

Function test	Status
Insert	(√) succeeded ()

Read	(√) <i>successed</i> ()
Update	(√) <i>successed</i> ()
Delete	(√) <i>successed</i> ()
Update status pengambilan	(√) <i>successed</i> ()
Insert pemberian obat	(√) <i>successed</i> ()
Delete pemberian obat	(√) <i>successed</i> ()


8. Interface dan testing pembayaran



Gambar 10. Form pembayaran

Function test	Status
Insert	(√) <i>successed</i> ()
Read	(√) <i>successed</i> ()
Update	(√) <i>successed</i> ()
Delete	(√) <i>successed</i> ()
Update status pembayaran	(√) <i>successed</i> ()
Print by month	(√) <i>successed</i> ()
Select by status pembayaran	(√) <i>successed</i> ()

9. Interface dan testing nota



Jl. Pangeran Natakusuma, Gg. Selamat Bersama, No. 34
081348353435 @klinikhevanpnk klinikhevanpnk@gmail.com

Nomor : 21-07.107
 Nama Pasien : nn (Kucing)
 Nama Pemilik : Riska
 Alamat : jl. HJ rais arahaman gg. Sekawan no.48
 No. Telp : 081253898170

INVOICE

No	Item	Qty	Harga Satuan (Rp)	Subtotal (Rp)
1	Antibiotik	1	35.000	35.000
2	Antibiotik	1	25.000	25.000
3	Antipiretik	1	10.000	10.000
4	Operasi Fraktur II			2.000.000
5	Pemeriksaan ALT			55.000
Subtotal (Rp)				2.125.000
Pembayaran dapat dilakukan melalui:				Administrasi : Rp. 20.000
BCA: 7155049618 A.n (Noor Asy Syifa)				Jasa Dokter : Rp. 80.000
Mandiri: 1390004194282 A.n (Noor Asy Syifa)				Total Biaya : Rp. 2.225.000
Pontianak, 2021-07-04				Potongan : Rp. - 1.200.000
				Grand Total : Rp. 1.025.000
				Uang Bayar : Rp. 300.000
drh. Syifa				Sisa Pembayaran : Rp. - 725.000

Gambar 11. Cetak nota pembayaran

Function test	Status
Read by nomor rekam	(√) <i>successed</i> ()

3.4 Integration and System Testing

Pengujian Integrasi (*Integration Testing*) melakukan pengujian yang difokuskan pada gabungan unit-unit atau modul-modul yang membentuk kesatuan fungsional. Pengujian ini didasarkan pada

informasi dari deskripsi perancangan awal perangkat lunak. Pengujian dilakukan untuk menemukan kesalahan.

Pengujian sistem dirancang untuk meyakinkan bahwa:

1. Semua kebutuhan fungsional perangkat lunak terpenuhi
2. Kinerja perangkat lunak telah sesuai dengan kebutuhan
3. Dokumentasi sudah benar

3.5 Operation and Maintenance

Tahapan ini berguna untuk mempertahankan kinerja sistem dengan baik, memungkinkan agar kinerja sistem berjalan secara maksimal dalam jangka panjang. Sistem dapat dimanfaatkan dengan baik oleh end-user dan mencegah terjadinya kerusakan sistem yang berdampak pada penurunan kinerja atau hilangnya data.

4. KESIMPULAN

Berdasarkan pembahasan yang dipaparkan diatas, berikut kesimpulan yang didapat.

1. Penerapan pemodelan UML diagram dalam pengembangan perangkat lunak model *waterfall* dapat meminimalkan kesalahan yang terjadi pada fase *implementation* dan *unit testing*, karena *stakeholder* dapat memahami hasil visualisasi lingkup sistem pada *usecase diagram*, dan meminimalkan kesalahan fungsional yang dikerjakan oleh *programmer* karena terbantuan oleh *class diagram* dan *sequence diagram* dijabarkan dengan baik.
2. Pengujian *blackbox testing* menunjukkan hasil yang *positive* karena semua fungsionalitas sistem dapat berjalan dengan baik sehingga kebutuhan *stakeholder* dapat terpenuhi oleh sistem.
3. Penerapan pemodelan UML diagram memberikan dampak *positive*, hasil perhitungan estimasi dan kebutuhan dapat dicapai dengan baik dan sesuai dengan permintaan *stakeholder* serta memberikan gambaran yang jelas

kepada *programmer* untuk membuat sistem sesuai kebutuhan.

4. Peralihan dokumentasi fisik ke bentuk digital menjadikan pengelolaan data ke bentuk informasi menjadi lebih cepat, proses penyipanan dan alur sistem menjadi terstandarisasi, dan tidak membutuhkan ruang penyimpanan yang besar.

5. SARAN

Adapun saran dari penelitian ini berkaitan dengan analisa metode pengujian perangkat lunak *blackbox testing* dan pemodelan UML diagram adalah sebagai berikut.

1. Pembahasan penelitian dapat diperluas dengan penerapan metode pengujian perangkat lunak *whitebox testing*, dan *graybox testing*.
2. Fase *requirement analysis and definition* dapat ditambahkan beberapa metode lainnya seperti metode pengumpulan data *sampling*, studi dokumentasi atau kuisioner.
3. Menjabarkan estimasi waktu dan biaya berdasarkan teknik perhitungan *function point*.
4. Membandingkan aplikasi *veterinary service* ini menggunakan pengembangan perangkat lunak lainnya.

DAFTAR PUSTAKA

- [1] I. M. A. Sanjaya, P. W. Buana, And I. K. A. Purnawan, "Sistem Informasi Manajemen Veteriner Berbasis Mobile Platform Android," *Merpati*, Vol. 4, No. Vol. 4, No. 3 Desember 2016, Pp. 205–213, 2016, [Online]. Available: <https://ojs.unud.ac.id/index.php/merpati/article/view/26878>.
- [2] M. Iqbal, "Strategi Penguatan Kinerja Pelayanan Kesehatan Hewan Dalam Mendukung Sistem Kesehatan Hewan Nasional," *Anal. Kebijak. Pertan.*, Vol. 9, No. 1, P. 53, 2016, Doi: 10.21082/Akp.V9n1.2011.53-71.
- [3] M. Bolung And H. R. K. Tampangela, "Analisa Penggunaan Metodologi Pengembangan Perangkat Lunak," *J. Eltikom*, Vol. 1, No. 1, Pp. 1–10, 2017, Doi: 10.31961/Eltikom.V1i1.1.
- [4] O. Prasimsha And H. P. Putro, "Manajemen Proyek Pengembangan Perangkat Lunak Kerja Sama Perguruan Tinggi Dan Software House (Studi Kasus : Pengembangan Sistem Penilaian Capaian Pembelajaran Lulusan Program Studi Informatika)" *Automata*, Vol. 1, No. 2, 2020.
- [5] G. Wiro Sasmito, "Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal," *J. Inform. Pengemb. It*, Vol. 2, No. 1, Pp. 6–12, 2017.
- [6] F. Agustini, "Sistem Informasi Penyewaan Kamar Menggunakan Metode Waterfall Dengan Konsep Pemrograman Berbasis Objek (Studi Kasus : Hotel Bonita Cisarua Bogor)," *J. Tek. Komput. Amik Bsi*, Vol. Iii, No. 1, Pp. 114–123, 2017.
- [7] A. Hendini, "Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang Barang (Studi Kasus: Distro Zhezha Pontianak)," *J. Khatulistiwa Inform.*, Vol. Iv, No. 2, Pp. 107–116, 2016.
- [8] W. N. Cholifah, S. M. Sagita, And S. Knowledge, "Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android," Vol. 3, No. 2, Pp. 206–210, 2018.