

Pengujian Black Box dengan Teknik Equivalence Partitioning pada Aplikasi Monitoring Pemberian Obat Filariasis Berbasis Android

Mohammad Reza Maulana*, Eko Budi Susanto, Paminto Agung Christianto

Program Studi Teknik Informatika, STMIK Widya Pratama, Pekalongan, Indonesia
Email: ¹*reza@stmik-wp.ac.id, ²ekobudi.s@stmik-wp.ac.id, ³p_a_chr@stmik-wp.ac.id
Email Penulis Korespondensi: reza.stmikwp@gmail.com

Abstrak—Penerapan teknologi informasi di berbagai sektor, termasuk dalam bidang kesehatan, telah menjadi suatu kebutuhan. Dinas Kesehatan (Dinkes) Kota Pekalongan menggunakan teknologi informasi untuk mendukung visi Pemerintah Kota Pekalongan sebagai Smart City. Salah satu fokus utama Dinkes adalah penanganan penyakit kaki gajah (filariasis). Meskipun sudah mengadakan program Pemberian Obat Pencegahan secara Massal (POPM), angka mikor filaria (MF) rate masih di atas 1%, sehingga memerlukan POPM tambahan. Salah satu permasalahan adalah monitoring pemberian obat oleh petugas Tenaga Pelaksana Eliminasi (TPE) yang belum optimal. Oleh karena itu, diperlukan mekanisme monitoring, diimplementasikan melalui dashboard eksekutif berbasis online. Data dan grafik di dashboard berasal dari aplikasi pencatatan berbasis Android yang dipasang di smartphone petugas TPE. Aplikasi Android perlu diuji untuk meminimalisir kesalahan atau bug dimana dapat diuji dengan metode Black Box. Penelitian ini bertujuan untuk menguji sistem dalam hal pemberian input dan output tanpa mengetahui detail kode program. Salah satu teknik yang dapat digunakan di dalam pengujian Black Box adalah teknik Equivalence Partitioning, dimana teknik ini akan membagi input menjadi partisi valid dan invalid untuk memastikan hasil sesuai dengan kasus uji. Tahapan penelitian melibatkan penentuan use case, kriteria, partisi uji, pembuatan data uji, dan desain test case. Pengujian dilakukan melalui test case yang menguji proses login, input data obat, dan pengiriman foto sebagai bukti konsumsi obat. Hasil pengujian menunjukkan bahwa dari tujuh test case yang ada, sistem sudah berjalan dengan baik sesuai dengan skenario yang diharapkan. Beberapa rekomendasi perbaikan diperlukan, terutama dalam pesan output kepada pengguna. Secara umum aplikasi monitoring pemberian obat filariasis telah berfungsi dengan baik, sesuai dengan rancangan aplikasi. Evaluasi menghasilkan rekomendasi perbaikan untuk meningkatkan pengalaman pengguna. Penelitian ini dapat menjadi landasan untuk pengembangan selanjutnya, termasuk pengujian menyeluruh pada aplikasi.

Kata Kunci: Pengujian Black Box; Equivalence Partitioning; Android; Dashboard Monitoring; Filariasis

Abstract—The application of information technology in various sectors, including the health sector, has become a necessity. Dinas Kesehatan (Dinkes) of Pekalongan City uses information technology to support the Pekalongan City Government's vision as a Smart City. One of the main focuses of the Health Department is treating elephantiasis (filariasis). Even though we have implemented a Mass Preventive Medication (Pemberian Obat Pencegahan secara Massal – POPM) program, the mycorrhiza filarial (MF) rate is still above 1%, requiring additional POPM. One of the problems is the monitoring of drug administration by Tenaga Pelaksana Eliminasi (TPE) officers which is not optimal. Therefore, a monitoring mechanism is needed, implemented through an online-based executive dashboard. The data and graphs on the dashboard come from an Android-based recording application installed on the TPE officer's smartphone. Android applications need to be tested to minimize errors or bugs which can be tested using the Black Box method. This research aims to test the system in terms of providing input and output without knowing the details of the code. One technique that can be used in Black Box testing is the Equivalence Partitioning technique, which divides the input into valid and invalid partitions to ensure the results match the test case. The research stages involve determining use cases, criteria, test partitions, creating test data, and test case design. Testing is carried out through test cases that test the login process, inputting drug data, and sending photos as proof of drug consumption. The results show that by testing the seven test cases, the system can be used properly according to the expected scenario. Some recommendations for improvements that are needed are adjustments to the output message to users. In general, the application for monitoring filariasis drug administration has functioned well, under the application design. Evaluation produces recommendations for improvements to improve user experience. This research can become a basis for further development, including thorough testing of the application.

Keywords: Black Box Testing; Equivalence Partitioning; Android; Dashboard Monitoring; Filaria

1. PENDAHULUAN

Penerapan teknologi informasi saat ini telah merambah ke hampir semua bidang dan organisasi untuk mendukung proses bisnis yang ada, salah satunya diterapkan di organisasi yang menangani bidang kesehatan [1]. Dengan memanfaatkan teknologi informasi, organisasi dapat menjalankan kegiatan menjadi lebih efisien, kualitas pelayanan meningkat dan menurunkan biaya-biaya yang berkaitan dengan kesehatan [2]. Dinas Kesehatan (Dinkes) Kota Pekalongan merupakan salah satu organisasi yang sangat sadar pentingnya pemanfaatan teknologi informasi dalam menunjang pekerjaan. Hal ini sejalan dengan visi Pemerintah Kota Pekalongan dalam menerapkan konsep *Smart City*, dimana salah satunya untuk mendukung pengembangan ekosistem digital di Kota Pekalongan [3].

Dinkes Kota Pekalongan memiliki banyak program kerja untuk menjadikan masyarakat Kota Pekalongan hidup lebih sehat dan terlayani dalam hal pelayanan kesehatan. Salah satu pekerjaan yang masih menjadi fokus utama dari Dinkes Kota Pekalongan adalah penanganan penyakit kaki gajah dimana wilayah ini menjadi endemis filariasis [4], [5]. Dinkes Kota Pekalongan telah melaksanakan kegiatan pemberian obat pencegahan secara massal (POPM) filariasis pada tahun 2011 – 2015 dimana cakupan pemberian obat ini masih di bawah 65%. Hal ini masih di bawah standar pengobatan massal filariasis yang telah ditetapkan oleh Kementerian Kesehatan Indonesia sebesar minimal 65% [5]. Lebih lanjut, Kota Pekalongan mengadakan kembali kegiatan POPM pada tahun 2017 dan 2018 [6]. Namun, berdasarkan hasil evaluasi yang telah dilakukan, Kota Pekalongan masih harus melakukan kegiatan POPM lagi dikarenakan angka mikor filaria (MF) rate

masih diatas 1%. Oleh karena itu, Dinkes Kota Pekalongan kembali harus mengulang untuk ketiga kalinya dalam pelaksanaan kegiatan POPM filariasis selama 2 tahun pada tahun 2021 dan 2022 [7].

Salah satu permasalahan yang ada di dalam kegiatan POPM filariasis adalah proses pemantauan pemberian obat yang dilakukan oleh petugas Tenaga Pelaksana Eliminasi (TPE) belum dilakukan secara optimal [7]. Dari permasalahan tersebut, diperlukan sebuah mekanisme monitoring pelaksanaan kegiatan. Salah satu solusi dalam pelaksanaan monitoring dapat dilakukan melalui dashboard eksekutif, dimana sistem informasi ini dapat diakses secara online dan data yang tersedia bersifat *realtime* [7], [8]. Dengan adanya dashboard eksekutif ini, pimpinan dapat menentukan kebijakan secara cepat dan tepat melalui data-data yang ada, sehingga pengawasan-pengawasan dalam pelaksanaan kegiatan akan menjadi lebih baik.

Data-data dan grafik yang ada di dalam dashboard eksekutif ini dikirimkan dari aplikasi pencatatan dan monitoring pemberian obat berbasis Android. Aplikasi ini dipasang di smartphone petugas TPE, dan mereka mencatat aktivitas pemberian obat melalui aplikasi ini. Dengan menggunakan smartphone petugas dapat dengan mudah membawa perangkat tersebut dan langsung menginputkan data pemberian obat seketika itu juga. Platform Android dipilih dikarenakan dari petugas TPE yang ada rata-rata menggunakan smartphone yang memiliki sistem operasi Android, dimana Android ini merupakan platform yang paling banyak digunakan oleh masyarakat Indonesia [9].

Aplikasi yang dijalankan oleh petugas TPE ini sangat penting dikarenakan merupakan sumber pengiriman data yang akan masuk ke dashboard eksekutif. Aplikasi perlu diuji untuk meminimalisir kemungkinan adanya *error* atau *bug* yang bisa muncul saat penggunaan aplikasi. Selain itu, pengujian merupakan salah satu tahapan penting yang ada di dalam siklus pengembangan perangkat lunak, dimana pengujian dilakukan untuk meningkatkan kualitas aplikasi [10], [11]. Salah satu pengujian yang bisa diterapkan untuk menguji sebuah aplikasi adalah pengujian *Black Box*. Pengujian *Black Box* merupakan pengujian yang fokus pada perilaku input dan output dari objek yang diuji [12], [13]. Pengujian *Black Box* sangat efektif untuk diterapkan pada proses pengujian sebuah software karena mudah untuk dilakukan, meskipun penguji tidak tahu sama sekali bagaimana kode program bekerja di dalam aplikasi.

Beberapa teknik atau pendekatan yang dapat digunakan untuk melakukan pengujian *Black Box* yaitu *Equivalence Partitioning*, *Boundary Value Analysis*, *State Transition Testing*, *Decision Table Testing*, *Pair-Wise Testing* dan *Use-Case Testing* [13]. Dari beberapa teknik tersebut, *Equivalence Partitioning* merupakan salah satu teknik yang banyak digunakan oleh peneliti untuk digunakan dalam menguji aplikasi [14], [15], [16], [17], [18], [19]. Teknik *Equivalence Partitioning* merupakan teknik pengujian dimana penguji membagi dua partisi inputan menjadi inputan *valid* dan inputan *invalid* untuk memastikan hasil/output dari sebuah sistem sesuai dengan kasus uji yang ada [13]. Dari satu proses yang sama, dengan menginputkan nilai yang berbeda, output dari sistem dapat berbeda juga.

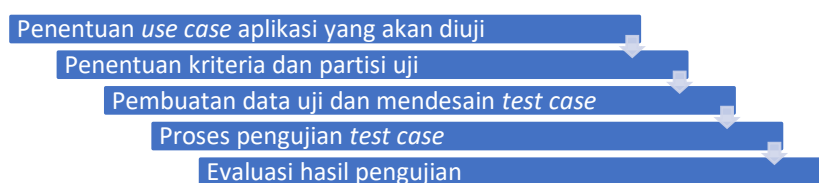
Penelitian yang dilakukan oleh Marfiah menyatakan bahwa pengujian dengan teknik *Equivalence Partitioning* membantu penguji dalam menentukan deskripsi *case* pengujian untuk mendapatkan gambaran kualitas aplikasi dan menemukan kesalahan yang dapat muncul saat pengguna menginputkan data di aplikasi [14]. Selain itu dalam penelitian lain, dalam menjalankan teknik *Equivalence Partitioning* penguji dapat dengan mudah melakukan pengujian berdasarkan data uji yang detail sesuai dengan use case dan kriteria-kriteria yang telah ditentukan [15]. Kemudian, menurut Sasongko di dalam penelitiannya, pendekatan teknik pengujian ini membantu dalam proses pembuatan rancangan *test case* untuk menemukan kesalahan yang tidak muncul saat proses *handling error* [16].

Penelitian ini akan berfokus pada pengujian fitur utama di dalam aplikasi mobile yang digunakan oleh petugas TPE, sehingga dapat dipastikan aplikasi dapat berjalan dengan baik. Terdapat tujuh *test case* dari tiga fitur utama yang akan dilakukan pengujian. Dengan menggunakan pendekatan teknik pengujian *Equivalence Partitioning* diharapkan aplikasi dapat berjalan seperti yang diharapkan sesuai dengan skenario yang telah dibuat. Selain itu pengujian ini juga dapat mendeteksi adanya kesalahan proses dan output yang terjadi, sehingga dapat dilakukan perbaikan *bug* pada aplikasi maupun mendapatkan hasil rekomendasi perbaikan aplikasi kedepannya.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan-tahapan penelitian diperlukan agar tujuan dalam melakukan pengujian sistem dapat dilakukan secara terstruktur dan dapat dijalankan dengan baik. Seperti yang telah dijelaskan pada bagian pendahuluan, penelitian ini menggunakan pengujian *Black Box* dengan teknik *Equivalence Partitioning* di dalam menguji aplikasi monitoring pemberian obat filariasis berbasis android. Terdapat beberapa langkah yang dilakukan di dalam penelitian ini seperti yang terlihat pada Gambar 1.



Gambar 1. Tahapan Penelitian pada Pengujian aplikasi monitoring pemberian obat filariasis berbasis android

Dari Gambar 1 dapat dilihat bahwa tahapan di dalam penelitian ini terdapat lima langkah utama. Pada tahap pertama, peneliti melakukan penentuan *use case* proses di dalam aplikasi yang akan digunakan sebagai objek pengujian. Hal ini penting dilakukan agar menjadi jelas bagian-bagian mana saja di dalam aplikasi yang akan dilakukan proses pengujian. Setelah itu, tahap kedua adalah menentukan kriteria apa saja yang akan dijadikan acuan untuk pengujian serta membuat partisi dengan membagi kelompok di setiap kriteria untuk menentukan kondisi nilai *valid* dan *invalid*. Dalam pembagian kelompok ini, nilai input dikelompokkan ke dalam partisi setara, yang dapat diperoleh dengan mempertimbangkan karakteristik input yang sama. Selanjutnya, pada tahap ketiga data uji dibuat berdasarkan kondisi partisi uji yang telah ditentukan di tahap sebelumnya dan pembuatan desain *test case* untuk selanjutnya dilakukan proses pengujian. Nilai input pada data uji yang dibuat harus mewakili partisi yang sama di dalam kelompoknya, sehingga penguji hanya perlu menguji satu kali untuk mewakili semua nilai input dalam partisi tersebut [20]. Dengan penerapan pendekatan seperti hal tersebut, maka dapat mengurangi waktu dan biaya pengujian.

Kemudian tahap utama pada pengujian dilakukan pada tahap keempat, yaitu proses pengujian pada aplikasi monitoring pemberian obat filariasis berbasis android sesuai dengan test case yang sudah ditentukan. Dari pengujian yang dilakukan, penguji akan melakukan pencatatan hasil pengujian pada tabel *test case* yang telah dibuat. Selanjutnya, pada tahap terakhir, peneliti melakukan analisa dan evaluasi hasil pengujian. Tahap ini perlu dilakukan agar dapat diketahui kelayakan dari aplikasi dan rekomendasi-rekomendasi yang dapat diterapkan untuk perbaikan aplikasi.

2.2 Pengujian Black Box

Pengujian perangkat lunak dapat dilakukan dengan berbagai metode, salah satu yang dapat digunakan adalah pengujian *Black Box*. Pengujian *Black Box* dikenal juga dengan pengujian berbasis spesifikasi atau pengujian fungsional dimana pengujian yang dilakukan berfokus pada tingkah laku aplikasi sesuai dengan inputan yang diberikan dan hasil yang didapatkan tanpa mengetahui logika kode program [12], [13]. Pengujian ini dilakukan untuk memastikan aplikasi yang dihasilkan sudah sesuai dengan kebutuhan fungsional yang telah ditentukan.

2.3 Teknik Equivalence Partitioning

Di dalam metode pengujian *Black Box* terdapat beberapa teknik yang dapat digunakan di dalam pendekatan pengujiannya. Salah satu teknik yang dapat digunakan adalah *Equivalence Partitioning*. Teknik *Equivalence Partitioning* adalah metode pengujian di mana penguji akan membagi input menjadi dua bagian yaitu input yang *valid* dan input yang *tidak valid* [13]. Pembagian ini dilakukan untuk memeriksa apakah hasil atau output sistem sesuai dengan skenario pengujian yang telah ditentukan. Dalam satu proses yang sama, berbagai nilai input dapat menghasilkan keluaran yang berbeda dari sistem yang diuji.

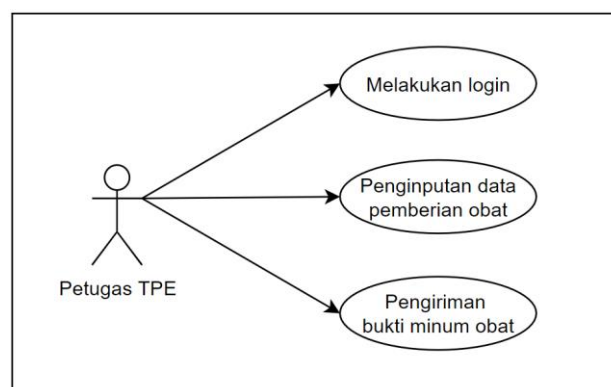
3. HASIL DAN PEMBAHASAN

3.1 Rancangan Pengujian

Sesuai tahapan penelitian yang telah dijabarkan, rancangan pengujian ini perlu dipersiapkan sebelum penguji melakukan pengujian pada aplikasi. Jika dilihat dari langkah-langkah yang ada, rancangan pengujian ini ada di tahap pertama sampai tahap ketiga.

3.1.1 Penentuan Use case pada Aplikasi

Penentuan *use case* aplikasi ini penting dilakukan agar scope dari pengujian yang dilakukan dapat dibatasi sehingga tujuan pengujian dapat tercapai. Gambar 2 merupakan *use case* aplikasi yang dijadikan dasar untuk proses pengujian.



Gambar .2 Use Case Aplikasi untuk Proses Pengujian

Pada Gambar 2 dapat dilihat bahwa use case yang diambil untuk pengujian aplikasi adalah proses melakukan login aplikasi, penginputan data pemberian obat yang telah dibagikan kepada warga dan pengiriman bukti minum obat berupa foto warga yang sedang minum obat.

3.1.2 Penentuan Kriteria dan Partisi Uji

Setelah *use case* ditentukan, tahap selanjutnya yaitu menentukan kriteria. Tabel 1 merupakan kriteria yang digunakan dalam pengujian, dimana diambil dari sebagian fitur yang ada di dalam *use case*.

Tabel 1. Kriteria Uji

Kode	Kriteria
CR001	Mengisi data login aplikasi
CR002	Mengisi data jumlah obat yang diberikan
CR003	Mengupload bukti minum obat

Dari data yang ada di Tabel 1 tersebut selanjutnya akan dibagi menjadi 2 partisi dengan nilai kelompok *valid* dan *invalid* untuk setiap kriteria yang telah dibuat.

Untuk nilai partisi dengan kode CR001 dan CR003 memiliki nilai boolean *True* atau *False*, dimana nilai *True* masuk kelompok *valid* dan nilai *False* masuk kelompok *invalid*. Kemudian untuk nilai partisi kriteria dengan kode CR002 memiliki nilai dengan tipe integer dengan range value 0 sampai 4. Selain nilai tersebut, maka nilai yang diinputkan masuk kelompok nilai *invalid*. Tabel 2 merupakan partisi uji yang digunakan untuk penentuan jenis hasil uji *valid* atau *invalid* pada kriteria yang telah ditentukan.

Tabel 2. Partisi Uji

Kode Partisi Uji	Paramater	Jenis Hasil Uji	Equivalence Partitioning
TP001A	Input login	<i>valid</i>	Login berhasil
TP001B	Input login	<i>invalid</i>	Login gagal
TP002A	Input data obat	<i>valid</i>	$0 \leq x \leq 4$
TP002B	Input data obat	<i>invalid</i>	$x < 0$
TP002C	Input data obat	<i>invalid</i>	$x > 4$
TP001A	Upload foto	<i>valid</i>	Upload berhasil
TP001B	Upload foto	<i>invalid</i>	Upload gagal

Dari Tabel 2 dapat dilihat dari tiga kriteria dibagi menjadi 2-3 partisi uji dimana terdapat jenis hasil uji *valid* dan *invalid*. Selain itu terdapat keterangan untuk masing-masing partisi uji pada kolom *Equivalence Partitioning* yang menjadikan jenis hasil uji tersebut apakah *valid* atau *invalid*.

3.1.3 Pembuatan Data Uji dan Test Case Pengujian

Setelah pembuatan kriteria dan partisi uji, tahap selanjutnya yaitu pembuatan data uji. Data uji yang dibuat merujuk pada partisi uji yang sebelumnya telah ditentukan pengelompokan nilainya. Tabel 3 merupakan data uji yang digunakan untuk penentuan *test case* pada tahap berikutnya.

Tabel 3. Data Uji

Kode Data Uji	Paramater	Sampel Data Uji	Jenis Hasil Uji	Kode Partisi Uji
DT001A	Input login	username: 933633 password: 933633	<i>valid</i>	TP001A
DT001B	Input login	username: 012345 password: 012345	<i>invalid</i>	TP001B
DT002A	Input data obat	3	<i>valid</i>	TP002A
DT002B	Input data obat	-2	<i>invalid</i>	TP002B
DT002C	Input data obat	6	<i>invalid</i>	TP002C
DT003A	Upload foto	Memilih foto	<i>valid</i>	TP001A
DT003B	Upload foto	Tidak memilih foto	<i>invalid</i>	TP001B

Pada Tabel 3 sampel data uji didefinisikan sebagai inputan yang akan diberikan ke dalam sistem. Saat proses pengujian dilakukan, penguji akan memasukkan nilai-nilai tersebut kemudian memprosesnya dan mencatat hasilnya.

Setelah data uji ditentukan, proses berikutnya merupakan pembuatan *test case* yang dijadikan penguji untuk dasar melakukan pengujian di aplikasi dan mencatat hasilnya.

Tabel 4. Test Case

Kode Test Case	Parameter (Kode Data Uji)	Jenis Hasil Uji	Output	Hasil
TC001A	Input login (DT001A)	<i>valid</i>		
TC001B	Input login (DT001B)	<i>invalid</i>		
TC002A	Input data obat (DT002A)	<i>valid</i>		
TC002B	Input data obat (DT002B)	<i>invalid</i>		
TC002C	Input data obat (DT002C)	<i>invalid</i>		
TC003A	Upload foto (DT003A)	<i>valid</i>		

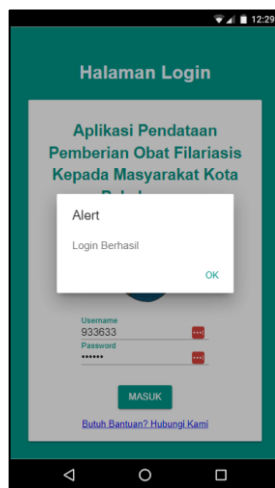
TC003B	Upload foto (DT003B)	invalid
--------	----------------------	---------

Tabel 4 merupakan *test case* yang digunakan oleh penguji dalam melaksanakan proses pengujian dan mencatat hasil pengujian. Penguji akan melakukan pengujian satu per satu sesuai kode *test case* dan langsung mencatat hasilnya.

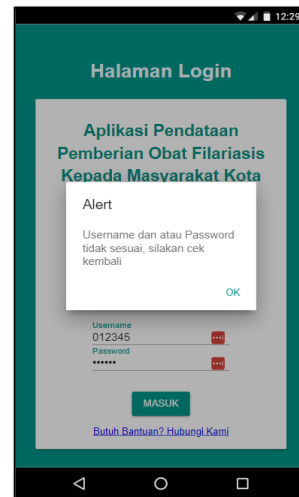
3.2 Hasil Pengujian

Pada bagian ini akan dijabarkan hasil pengujian sesuai dengan rancangan yang telah dibuat sebelumnya. Dari hasil pengujian yang didapatkan, selanjutnya akan dilakukan pembahasan dan evaluasi. Hasil evaluasi dapat digunakan sebagai bahan perbaikan maupun *improvement* aplikasi yang diuji.

Tahapan pengujian dilakukan sesuai dengan *test case* yang telah dirancang pada bagian sebelumnya. Terdapat 7 *test case* yang telah dibuat untuk dilakukan proses pengujian.



Test case TC001A (a)



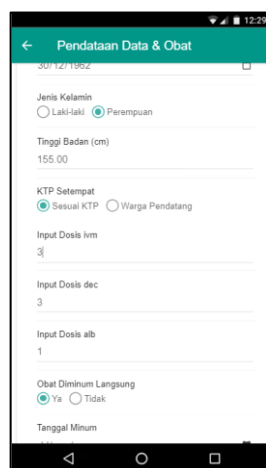
Test case TC001B (b)

Gambar 3. Pengujian untuk Test Case TC001A dan TC001B

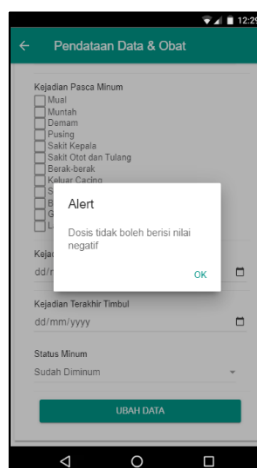
Bagian pengujian proses login terdapat 2 *test case* dimana masing-masing dibuat untuk hasil uji *valid* dan *invalid*. Bagian pertama *test case* dengan kode TC001A diuji dengan data uji DT001A yang menghasilkan hasil uji *valid*. Gambar 3 (a) menunjukkan hasil pengujian pada *test case* ini. Aplikasi akan memunculkan pesan login berhasil setelah menginputkan data username dan password yang sesuai. Setelah pengguna melakukan klik tombol OK pada pesan login berhasil, sistem selanjutnya akan mengarahkan pengguna ke halaman utama dari aplikasi.

Selanjutnya, untuk pengujian *test case* dengan kode TC001B diuji menggunakan data uji DT001B. Pada *test case* ini, jenis hasil uji yang diperoleh adalah hasil *invalid*. Pengguna akan memasukkan data *username* dan *password* yang telah dipersiapkan. Setelah pengguna klik tombol login maka akan memunculkan pesan gagal login dikarenakan *credential* yang dimasukkan tidak terdaftar di database sistem. Pesan yang muncul dapat dilihat pada Gambar 3 (b). Setelah tombol OK pada pesan yang muncul diklik, pengguna akan tetap pada halaman login. Hasil dari dua *test case* untuk mekanisme login ini sudah sesuai dengan rancangan awal pengembangan sistem.

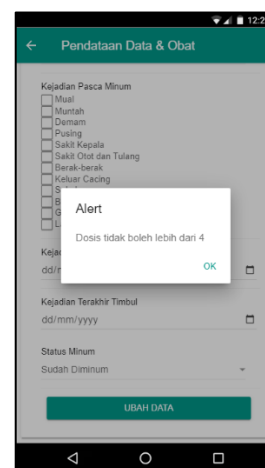
Berikutnya pengujian untuk proses penginputan data obat yang diberikan oleh petugas TPE kepada masyarakat yang hasil pengujiannya dapat dilihat pada Gambar 4.



Test case TC002A (a)



Test case TC002B (b)



Test case TC002C (c)

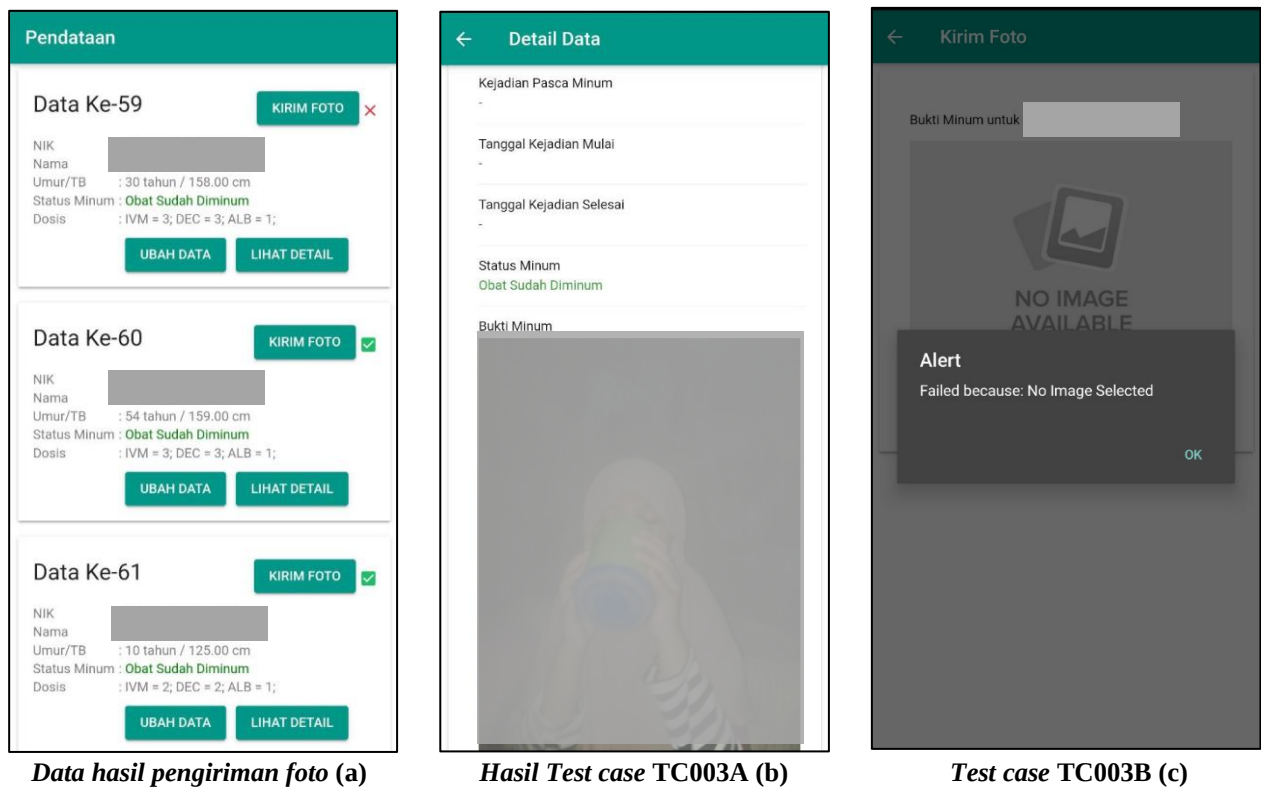
Gambar 4. Pengujian untuk Test Case TC002A, TC002B dan TC002C

Pada pengujian input data obat, terdapat 3 skenario yang akan diuji. Secara prinsip hasil uji untuk input obat akan *valid* jika pengguna memasukkan nilai 0 sampai dengan 4 dengan tipe integer pada field inputan pemberian data obat. Selain rentang nilai tersebut, maka hasil uji akan *invalid*. Gambar 4 (a) merupakan halaman untuk menginputkan pendataan pemberian obat dimana di dalamnya terdapat *field* untuk menginput data dosis obat.

Pada pengujian *test case* dengan kode TC002A pengguna akan menginputkan nilai 3 (data uji DT002A) pada inputan dosis obat. Ketika user klik tombol ubah data, maka sistem akan berhasil menyimpan atau mengupdate nilai ke dalam database. Pada *test case* ini hasil uji bernilai *valid*. Kemudian pada *test case* selanjutnya, pengujian akan dilakukan untuk kode TC002B yang diinput menggunakan data uji DT002B. Pengguna akan menginputkan nilai -2 (minus 2) pada inputan dosis obat. Ketika pengguna klik tombol ubah data, maka sistem tidak akan melakukan penyimpanan atau perubahan data. Sistem akan menampilkan pesan dosis tidak boleh berisi nilai negatif seperti yang terlihat pada Gambar 4 (b). Berikutnya, untuk *test case* dengan kode TC002C diuji menggunakan data uji DT002C dimana pengguna akan menginputkan nilai 6 pada *field* dosis obat. Saat pengguna selesai memasukkan data dan klik tombol ubah data, maka sistem tidak akan melakukan proses simpan/update data dan sistem akan memunculkan pesan dosis tidak boleh diisi melebihi nilai 4 seperti yang terlihat pada Gambar 4 (c).

Proses pengujian pada ketiga *test case* pada proses input data dosis obat yaitu TC002A, TC002B dan TC002C telah dilakukan. Hasil dari pengujian pada ketiga *test case* ini sesuai dengan perancangan awal dimana validasi input nilai telah berjalan sebagaimana mestinya.

Selain pengujian pada proses login untuk masuk ke sistem dan proses pencatatan pemberian obat, pengujian juga dilakukan untuk proses pengiriman gambar atau foto dari warga yang sedang minum obat. Foto ini digunakan sebagai bukti bahwa warga memang sudah meminum obat yang diberikan oleh petugas. Terdapat dua *test case* pada proses pengiriman foto ini yaitu *test case* TC003A dan TC003B.



Gambar 5. Pengujian untuk *Test Case* TC003A dan TC003B

Untuk melakukan pengujian pengiriman foto, pengguna dapat menuju halaman pendataan pada aplikasi seperti yang terlihat pada Gambar 5 (a). Pada halaman tersebut terdapat tombol kirim foto yang digunakan untuk pengiriman bukti bahwa warga telah meminum obat. Di samping tombol kirim, terdapat simbol centang hijau atau silang merah. Simbol centang hijau mengindikasikan bahwa warga tersebut telah difoto, sedangkan silang merah menandakan bahwa warga tersebut belum difoto atau belum dilakukan pendataan.

Pengujian untuk *test case* TC003A diuji dengan data uji DT003A, dimana pengguna akan memilih foto untuk dikirimkan ke server sebagai bukti bahwa penerima obat telah meminum obat yang telah diberikan. Proses pemilihan foto ini ada dua mekanisme yaitu pengguna akan melakukan foto secara langsung menggunakan kamera atau pengguna dapat memilih foto yang sudah tersimpan di gallery. Ketika pengguna sudah memilih foto dan mengirimkan ke server, maka bukti foto akan tersimpan. Bukti foto tersebut dapat dilihat pada halaman detail data seperti yang terlihat pada Gambar 5 (b). Selanjutnya untuk pengujian *test case* TC003B dilakukan dengan menggunakan data uji DT003B. Pada skenario pengujian ini, pengguna menekan tombol untuk mengambil foto lewat kamera atau gallery tetapi tidak jadi memilih foto.

Hasilnya sistem akan memberikan informasi gagal upload ke server dikarenakan tidak ada foto yang dipilih seperti yang terlihat pada Gambar 5 (c).

Pengujian pada kedua *test case* yang berkaitan dengan pengiriman foto telah dilakukan dan mendapatkan hasil yang sesuai dengan rancangan.

Dari semua *test case* yang telah dilakukan pengujian dapat dilihat hasilnya pada Tabel 5.

Tabel 5. Hasil Pengujian *Test Case*

Kode <i>Test Case</i>	Parameter (Kode Data Uji)	Jenis Hasil Uji	Output	Hasil
TC001A	Input login (DT001A)	<i>valid</i>	Sistem akan menampilkan pesan login berhasil dan mengarahkan pengguna ke halaman utama.	Berhasil
TC001B	Input login (DT001B)	<i>invalid</i>	Sistem akan menampilkan pesan bahwa login gagal karena username dan atau password tidak sesuai. Sistem akan tetap menampilkan halaman login.	Berhasil
TC002A	Input data obat (DT002A)	<i>valid</i>	Sistem akan melakukan proses simpan nilai dosis yang diinputkan ke dalam database yang ada di server	Berhasil
TC002B	Input data obat (DT002B)	<i>invalid</i>	Sistem akan menampilkan pesan nilai dosis yang diinputkan tidak boleh diisi nilai negatif. Sistem tidak akan melakukan proses simpan.	Berhasil
TC002C	Input data obat (DT002C)	<i>invalid</i>	Sistem akan menampilkan pesan nilai dosis yang diinputkan tidak boleh diatas nilai 4. Sistem tidak akan melakukan proses simpan.	Berhasil
TC003A	Upload foto (DT003A)	<i>valid</i>	Sistem akan melakukan pengiriman data foto yang telah dipilih ke server.	Berhasil
TC003B	Upload foto (DT003B)	<i>invalid</i>	Sistem tidak akan melakukan pengiriman data ke server dikarenakan tidak ada foto yang dipilih.	Berhasil

Dari Tabel 5 dapat dilihat bahwa penguji telah melakukan proses pengujian dan mencatat hasilnya. Dari tujuh *test case* yang ada semuanya berhasil diuji di mana sistem dapat menjalankan proses sesuai dengan kebutuhan fungsional yang telah ditentukan.

3.3 Evaluasi Hasil Pengujian

Langkah terakhir setelah dilakukan pengujian dari *test case* yang ada adalah evaluasi hasil pengujian. Dari hasil pengujian yang ada, dapat dilihat bahwa sistem sudah berjalan dengan baik sesuai dengan rancangan aplikasi yang telah dibuat sebelumnya. Meskipun demikian, terdapat beberapa perbaikan yang dapat dilakukan dari sisi proses bisnis yang ada maupun output yang dihasilkan dari aplikasi agar menjadi lebih baik dari sisi *user experience*.

Tabel 6. Evaluasi Hasil

Kode Evaluasi	Kode <i>Test Case</i>	Output Hasil Uji	Rekomendasi
EV001	TC001B	Sistem akan menampilkan pesan bahwa login gagal karena username dan atau password tidak sesuai	Perbaikan pesan yang muncul : Sistem dapat menampilkan pesan yang lebih sesuai. Apakah username tidak terdaftar atau passwordnya yang tidak sesuai.
EV001	TC003A	Sistem langsung mengirimkan foto ke server tanpa adanya pesan	Penambahan pesan : Sistem dapat menampilkan pesan bahwa foto telah berhasil dikirim ke server

Dari Tabel 6 pada evaluasi hasil pengujian dapat dilihat bahwa terdapat 2 rekomendasi yang dapat diimplementasikan untuk perbaikan selanjutnya. Pertama, untuk kode evaluasi EV001 yang merupakan evaluasi dari hasil *test case* TC001B, dimana hasil pengujian sistem akan menampilkan pesan gagal login karena username dan atau password tidak sesuai. Rekomendasi yang dapat diberikan untuk hasil uji ini adalah dengan membuat pesan lebih spesifik, apakah username tidak terdaftar ataukah passwordnya yang tidak cocok dengan username yang terdaftar. Jadi nantinya

dapat dibuat *test case* lain dengan kondisi username yang tidak terdaftar dan *test case* password yang tidak cocok dengan username yang ada.

Selanjutnya, untuk evaluasi kedua pada kode EV001 dengan hasil uji dari *test case* TC003A, ketika pengguna memilih dan mengupload foto, pengguna tidak diberitahu apakah foto terkirim atau tidak. Perbaikan yang dapat dilakukan adalah dengan memunculkan pesan bahwa foto berhasil atau gagal (bisa karena jaringan) terkirim ke server.

4. KESIMPULAN

Penelitian telah dilakukan dari proses pengumpulan data, pembuatan rancangan pengujian, pelaksanaan pengujian dan evaluasi. Metode pengujian *Black Box* dengan teknik *Equivalence Partitioning* digunakan pada penelitian ini. Berdasarkan penelitian yang telah dilakukan dengan melihat hasil pengujian dan evaluasi, maka dapat disimpulkan bahwa aplikasi monitoring pemberian obat filariasis berbasis android ini telah berfungsi dengan baik. Dari ketujuh *test case* yang ada, setelah dilakukan pengujian baik untuk jenis hasil uji *valid* maupun *invalid*, aplikasi berhasil berjalan sesuai dengan rancangan aplikasi yang telah ditetapkan. Meskipun demikian, berdasarkan evaluasi dari hasil pengujian diperlukan adanya peningkatan pengalaman pengguna dalam memberikan output pesan kepada pengguna. Hasil rekomendasi ini dimaksudkan agar pengguna dapat memahami proses yang ada dengan lebih baik, sehingga dapat menambah intuisi pengguna di dalam menggunakan aplikasi. Untuk penelitian berikutnya, pengujian dapat dilakukan secara menyeluruh pada aplikasi, baik aplikasi untuk kader yang memiliki sistem operasi android maupun aplikasi dashboard monitoring yang berbasis website.

REFERENCES

- [1] N. K. Ismatullah, A. P. Widodo, and S. A. Nugraheni, "Model EUCS (End User Computing Satisfaction) untuk Evaluasi Kepuasan Pengguna Terhadap Sistem Informasi Bidang Kesehatan : Literature Review," *Media Publikasi Promosi Kesehatan Indonesia*, vol. 5, no. 5, pp. 463–467, 2022.
- [2] N. M. Fadilla and W. Setyonugroho, "Sistem Informasi Manajemen Rumah Sakit Dalam Meningkatkan Efisiensi: Mini Literature Review," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 1, pp. 357–374, 2021.
- [3] S. Eniyati, R. N. Candra, S. Mulyani, and K. A. Martha, "Perhitungan Tingkat Kesiapan Implementasi Smart City dalam Perspektif Smart Governance dengan Metode FIS Mamdani," *Jurnal DINAMIK*, vol. 22, no. 1, 2017.
- [4] M. Ipa, E. Wijayanti, H. Hipokrates, E. P. Astuti, and Y. Yuliasih, "Prevalensi Wuchereria bancrofti Paska Pemberian Obat Pencegahan Massal Filariasis Limfatik di Wilayah Endemis Rendah Kota Pekalongan," *BALABA: JURNAL LITBANG PENGENDALIAN PENYAKIT BERSUMBER BINATANG BANJARNEGARA*, pp. 143–152, Dec. 2021, doi: 10.22435/blb.v17i2.4854.
- [5] H. B. Jorgi, P. Ginandjar, N. Kusariana, and L. Dian Saraswati Peminatan Epidemiologi dan Penyakit Tropik Fakultas Kesehatan Masyarakat, "Peran Tenaga Pelaksana Eliminasi dalam Pelaksanaan Program Pemberian Obat Secara Massal (POPM) Filariasis Di Kota Pekalongan," *JURNAL KESEHATAN MASYARAKAT (e-Journal)*, vol. 7, no. 2, pp. 2356–3346, 2019, [Online]. Available: <http://ejournal3.undip.ac.id/index.php/jkm>
- [6] Portal Berita Info Publik, "Sukseskan POPM Filariasis, Dinkes Terjunks Tim Kader TPE Filariasis Sweeping Door to Door." Accessed: Jan. 02, 2024. [Online]. Available: <https://www.infopublik.id/kategori/nusantara/511573/sukseskan-popm-filariasis-dinkes-terjunks-tim-kader-tpe-filariasis-sweeping-door-to-door>
- [7] M. R. Maulana, E. B. Susanto, and P. A. Christianto, "Pengembangan Dashboard Eksekutif untuk Sistem Monitoring Penanganan Filariasis Kota Pekalongan," *JURNAL LITBANG KOTA PEKALONGAN*, vol. 19, no. 2, 2021.
- [8] M. R. Maulana, A. Syaifudin, H. A. Budijanto, and E. B. Susanto, "Dahboard Eksekutif sebagai Media Koordinasi dan Monitoring Kegiatan untuk Peningkat Kinerja Aparatur Pemerintah," *JURNAL LITBANG KOTA PEKALONGAN*, vol. 18, no. 2, 2020.
- [9] StatCounter, "Mobile Operating System Market Share Indonesia," <https://gs.statcounter.com/os-market-share/mobile/indonesia>. Accessed: Feb. 09, 2024. [Online]. Available: <https://gs.statcounter.com/os-market-share/mobile/indonesia>
- [10] V. Garousi, A. Rainer, P. Lauvås, and A. Arcuri, "Software-testing education: A systematic literature mapping," *Journal of Systems and Software*, vol. 165, Jul. 2020, doi: 10.1016/j.jss.2020.110570.
- [11] N. Sánchez-Gómez, J. Torres-Valderrama, J. A. García-García, J. J. Gutiérrez, and M. J. Escalona, "Model-based software design and testing in blockchain smart contracts: A systematic literature review," *IEEE Access*, vol. 8. Institute of Electrical and Electronics Engineers Inc., pp. 164556–164569, 2020. doi: 10.1109/ACCESS.2020.3021502.
- [12] P. C. Jorgensen and B. DeVries, *Software Testing; A Craftsman's Approach*, 5th ed. Boca Raton, Oxon: CRC Press, 2021.
- [13] A. Spillner and T. Linz, *Software Testing Foundations A Study Guide for the Certified Tester Exam*, 5th ed. Heidelberg: dpunkt.verlag GmbH, 2021.
- [14] Marfuah and S. Adam, "Equivalence Partitions pada BlackBox Testing terhadap Sistem Pemberdayaan Pemerataan UMKM," vol. 6, no. 2, pp. 382–387, 2021, doi: 10.32493/informatika.v6i2.11457.
- [15] R. Pramudita, "Pengujian Black Box pada Aplikasi Ecampus Menggunakan Metode Equivalence Partitioning," *Informatics for Educators and Professionals*, vol. 4, no. 2, pp. 193–202, 2020.
- [16] B. B. Sasongko, F. Malik, F. Ardiansyah, A. F. Rahmawati, F. Dharma Adhinata, and D. P. Rakhmadani, "Pengujian Blackbox Menggunakan Teknik Equivalence Partitions pada Aplikasi Petgram Mobile," 2021.
- [17] E. Novalia and A. Voutama, "Black Box Testing dengan Teknik Equivalence Partitions Pada Aplikasi Android M-Magazine Mading Sekolah," 2022.
- [18] L. Ayu Marlina and S. Sabdo Wibowo, "Pengujian Sistem Informasi Perpustakaan Dengan Teknik Equivalence Partitioning di SMA Nurul Muttaqin Albarokah," *JACIS : Journal Automation Computer Information System*, vol. 3, no. 02, pp. 137–145, 2023.

- [19] A. Amalia, S. W. Putri Hamidah, and T. Kristanto, “Pengujian Black Box Menggunakan Teknik Equivalence Partitions Pada Aplikasi E-Learning Berbasis Web,” *Building of Informatics, Technology and Science (BITS)*, vol. 3, no. 3, pp. 269–274, Dec. 2021, doi: 10.47065/bits.v3i3.1062.
- [20] P. A. D. A. Santi, R. Afwani, Moh. A. Albar, S. E. Anjarwani, and A. Z. Mardiansyah, “Black Box Testing with Equivalence Partitioning and Boundary Value Analysis Methods (Study Case: Academic Information System of Mataram University),” in *Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science)*, Atlantis Press International BV, 2022, pp. 207–219. doi: 10.2991/978-94-6463-084-8_19.