

PERANCANGAN APLIKASI ANTARMUKA PADA PROTOTIPE SMART TOILET BERBASIS INTERNET OF THINGS (IOT) PADA PERANGKAT ANDROID

Daffa Maheswara Wiryawan^{*1)}, Nasywaan 'Ammar Sahasika²⁾, Nanda Fitri Tsalatsa³⁾, M. Arfan⁴⁾, dan Sumardi⁵⁾, Darjat⁶⁾

^{1,2,3}Program Studi Sarjana Departemen Teknik Elektro, Universitas Diponegoro, Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*})E-mail: daffamw19@gmail.com

Abstrak

Dengan kemajuan teknologi, toilet modern mengalami transformasi signifikan, termasuk pengembangan toilet pintar dengan fitur sensor otomatis dan bahan ramah lingkungan. Meskipun perkembangan ini meningkatkan kenyamanan pengguna, fokus terhadap fasilitas untuk petugas kebersihan masih kurang. Petugas kebersihan memiliki peran penting dalam menjaga toilet agar tetap bersih dan nyaman, kurangnya dukungan dapat membuat toilet menjadi kurang nyaman dan berpotensi sebagai tempat berkembangnya penyakit. Oleh karena itu diberikan dukungan dan fasilitas kepada petugas kebersihan dengan dibuatnya sistem monitoring smart toilet berbasis Internet of Things pada aplikasi android. Aplikasi android yang terintegrasi dengan sistem smart toilet memungkinkan petugas kebersihan untuk memantau kondisi lingkungan pada smart toilet secara jarak jauh. Penggunaan aplikasi android akan menampilkan kondisi lingkungan pada smart toilet dan juga jadwal petugas kebersihan setiap harinya. Penggunaan aplikasi android memberikan akses informasi yang responsif, informatif dan mudah digunakan oleh pengguna. Aplikasi android juga memiliki kompatibilitas terhadap versi android terbaru yaitu android 11 dan 12. Pada aplikasi android telah diuji pada beberapa responden salah satunya yaitu petugas kebersihan dan mendapatkan hasil 44,5% responden sangat setuju, 49,1% responden setuju, dan 6,4% kurang setuju pada tampilan antarmuka aplikasi dan mudah dimengerti serta mudah untuk digunakan.

Kata kunci : Toilet, smart toilet, monitoring, aplikasi android, petugas kebersihan

Abstract

With the advancement of technology, modern toilets are undergoing significant transformations, including the development of smart toilets with automatic sensor features and eco-friendly materials. While these developments improve user comfort, there has been less focus on facilities for janitors. While janitors play an important role in keeping washrooms clean and comfortable, a lack of support can make washrooms less convenient and a potential breeding ground for disease. Therefore, support and facilities are provided to janitors with the creation of an Internet of Things-based smart toilet monitoring system on android applications. Android applications that are integrated with smart toilet systems allow janitors to monitor environmental conditions in smart toilets remotely. The use of the android application will display the environmental conditions in the smart toilet and also the janitor's schedule every day. The use of android applications provides access to information that is responsive, informative and easy to use by users. The android application also has compatibility with the latest android versions, namely android 11 and 12. The android application has been tested on several respondents, one of which is the janitor and gets the results 44.5% of respondents strongly agree, 49.1% of respondents agree, and 6.4% disagree on the appearance of the application interface and is easy to understand and easy to use.

Keywords: Toilets, smart toilet, monitoring, android applications, janitors

1. Pendahuluan

Toilet merupakan sarana sanitasi yang memiliki peran penting bagi masyarakat untuk menjaga kesehatan dan kebersihan. Berbagai keperluan sanitasi dilakukan di

toilet seperti membersihkan tangan, membersihkan muka hingga pengelolaan limbah manusia secara higienis. Peran toilet selain sebagai unsur utama fasilitas sanitasi publik, juga merupakan suatu kebutuhan pokok dalam rumah tangga sebagai bagian dari infrastruktur sanitasi.

Perkembangan teknologi yang saat ini semakin maju berpengaruh terhadap toilet modern yang mulai mengalami transformasi cukup signifikan[1]. Toilet modern tidak hanya memenuhi fungsi dasarnya sebagai tempat pembuangan, tetapi juga dilengkapi dengan berbagai fasilitas tambahan yang meningkatkan kenyamanan dan efisiensi penggunaannya. Salah satu contohnya, toilet pintar yang dilengkapi dengan sensor otomatis, sistem penyiram air yang menggunakan daya dorong, serta penggunaan bahan ramah lingkungan untuk mengurangi dampak negatif terhadap lingkungan[2].

Fasilitas yang disediakan oleh toilet pintar umumnya hanya ditujukan untuk pengguna toilet, sementara fasilitas untuk petugas kebersihan masih minim dan kurang mendapat perhatian. Petugas kebersihan memiliki peran penting dalam menjaga toilet agar terjaga kebersihannya sehingga memberi kenyamanan bagi pengguna toilet. Tugas utama dari seorang petugas kebersihan toilet yaitu membersihkan toilet, wastafel, dan area sekitarnya serta memantau ketersediaan fasilitas toilet seperti sabun, tisu toilet agar selalu tersedia. Tanpa adanya peran dari petugas kebersihan, toilet dengan cepat dapat menjadi tempat yang tidak nyaman dan berpotensi menjadi sarang penyakit. Kehadiran petugas kebersihan yang berkualitas tidak hanya meningkatkan pengalaman pengguna, tetapi juga berkontribusi menjaga kebersihan publik secara keseluruhan[3]. Oleh karena itu, penting untuk memberikan apresiasi dan dukungan fasilitas yang layak terhadap pekerjaan yang mereka lakukan.

Fasilitas yang dapat mendukung petugas kebersihan toilet dalam melaksanakan tugas, salah satunya yaitu dengan perancangan aplikasi antarmuka pada prototipe *smart toilet* berbasis *Internet of Things* (IoT). Aplikasi ini dirancang agar dapat diakses melalui perangkat Android dan menyediakan informasi bagi petugas kebersihan toilet. Informasi yang tersedia dalam aplikasi ini mencakup berbagai parameter penting yang ada dalam toilet seperti kejernihan air, volume air, suhu air, suhu ruangan, kelembapan, tingkat bau, serta ketersediaan sabun dan tisu toilet. Selain itu, aplikasi ini juga dilengkapi dengan fitur jadwal petugas kebersihan yang bertugas setiap harinya. Petugas kebersihan dapat dengan mudah mengakses informasi yang mereka butuhkan melalui aplikasi ini untuk menjalankan tugas mereka dengan lebih efisien.

Penelitian terdahulu mengenai smart toilet oleh Adilia dkk [4] berhasil menciptakan *flush* dan lampu otomatis berbasis mikrokontroler Arduino uno. Penelitian lainnya dilakukan oleh Sari dkk [5] mengenai *smart toilet* menciptakan sistem pemantauan kondisi toilet berbasis *website*. Penelitian dari Miza Alfarisi [6] menggunakan sensor DS18B20 dan MQ-135 dalam pembacaan perubahan suhu dan kadar CO dalam memonitoring sirkulasi udara dengan nilai error kesalahan sebesar 0,85%. Penelitian Erik Prastika [7] berbasis sensor PIR dan sensor temperatur untuk toilet berdiri otomatis, yang pada hasil akhir dari penelitian tersebut menghasilkan keberhasilan alat sebesar 83,3% Pada beberapa penelitian

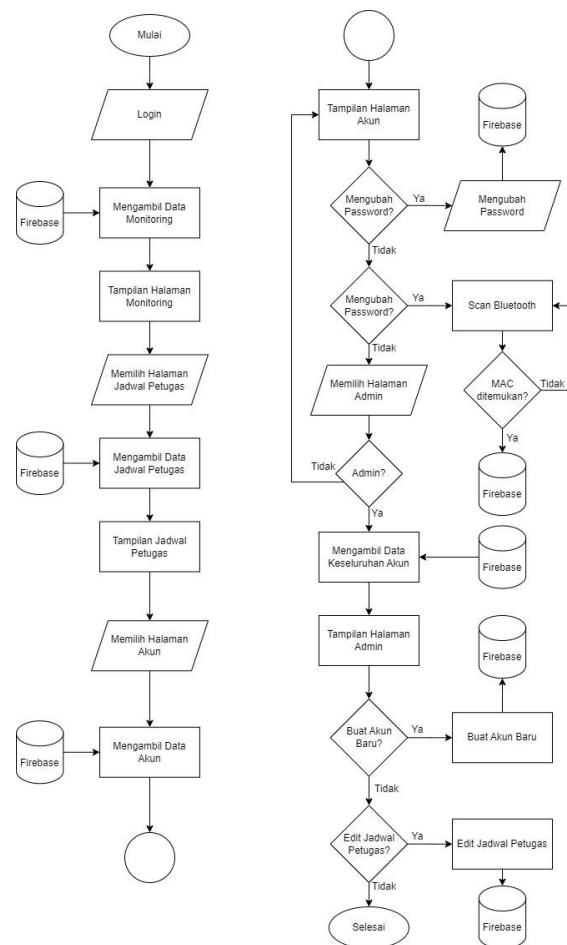
tersebut belum ada bahasan mengenai sistem *monitoring smart toilet* berbasis aplikasi android, sedangkan penggunaan aplikasi android dapat mempermudah petugas kebersihan untuk memantau kondisi toilet. Oleh karena itu, pada makalah ini dirancang sistem *monitoring smart toilet* berbasis aplikasi android menggunakan *flutter*.

2. Perancangan Sistem

2.1 Arsitektur Software

2.1.1 Flowchart Aplikasi Android

Flowchart yang biasa disebut dengan diagram alir adalah suatu jenis diagram yang merepresentasikan algoritme yang berisi langkah-langkah instruksi yang berurutan dalam sistem [8]. Pada perancangan makalah ini merepresentasikan penggunaan *flowchart* pada aplikasi android sehingga merepresentasikan visual dari alur kerja yang terdapat pada android seperti navigasi antar layar, pengelolaan data, penampilan informasi, dan interaksi pengguna [9]. Berikut adalah *flowchart* aplikasi android yang ditunjukkan pada Gambar 1.



Gambar 1. Flowchart Aplikasi Android

Proses *flowchart* dapat dilihat pada Gambar 1. Proses dimulai saat pengguna melakukan *login*. Setelah akses

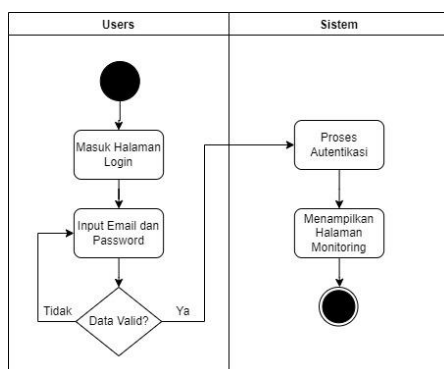
terverifikasi, sistem melakukan pengambilan data *monitoring* dari firebase yang selanjutnya ditampilkan pada halaman *monitoring*. Pengguna memiliki kemudahan untuk navigasi, di mana mereka dapat memilih halaman untuk melihat jadwal petugas, yang mengharuskan sistem untuk mengambil data jadwal petugas dari firebase sebelum ditampilkan. Selanjutnya, pengguna dapat memilih untuk mengakses halaman akun mereka, yang akan menjalankan sistem untuk kembali mengambil data dari firebase. Dalam tampilan halaman akun, pengguna diberi opsi untuk mengubah *password* mereka. Jika pilihan ini dipilih, proses perubahan *password* akan dilakukan. Namun, jika tidak ada keinginan untuk mengubah *password*, pengguna memiliki pilihan untuk melakukan presensi melalui *scanning bluetooth*. Jika sistem tidak menemukan *MAC address* yang sesuai, presensi dianggap gagal.

Di sisi lain, jika pengguna merupakan admin, akan diarahkan ke halaman admin setelah memilih opsi tersebut. Di halaman tersebut, admin dapat melihat data keseluruhan akun dari firebase. Halaman admin memberikan pilihan untuk membuat akun baru atau mengedit jadwal petugas. Setiap pilihan ini, jika dilaksanakan, akan menimbulkan perubahan atau penambahan yang sesuai dalam database firebase.

2.2 Activity Diagram Aplikasi

Activity diagram merupakan suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan cermat dalam suatu sistem [10]. Pada perancangan aplikasi android perlu adanya perancangan *activity diagram* agar dapat mengetahui alur kerja dari masing-masing proses pada aplikasi.

2.2.1 Activity diagram proses login

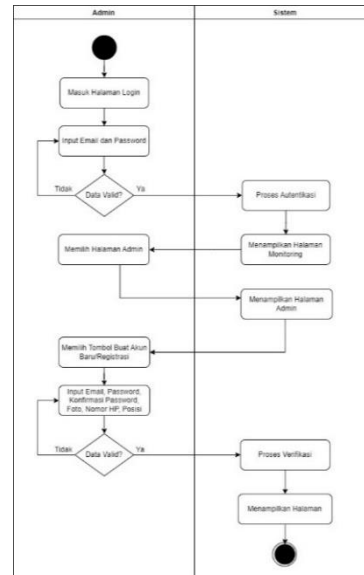


Gambar 2. Activity diagram proses login

Activity diagram proses login dapat dilihat pada Gambar 2. Halaman *login* merupakan halaman awal aplikasi, pada proses ini pengguna masuk dalam aplikasi dan muncul tampilan layar *login* dimana pengguna diminta untuk memasukkan alamat *e-mail* dan *password* yang telah

terdaftar. Apabila *e-mail* dan *password* yang dimasukkan benar maka pengguna akan dialihkan ke halaman *monitoring*.

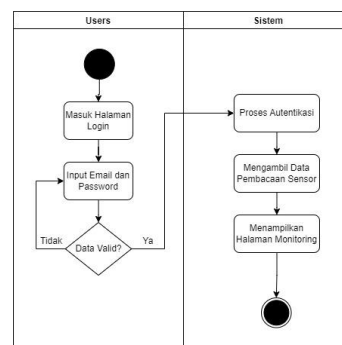
2.2.2 Activity diagram proses register



Gambar 3. Activity diagram proses register

Activity diagram proses register dapat dilihat pada Gambar 3. Admin dapat memilih tombol buat akun baru atau registrasi. Pada proses ini admin akan diminta untuk meng-input data berupa *E-mail*, *password*, konfirmasi *password*, foto, nomor HP serta posisi dari pengguna yang akan di daftarkan.

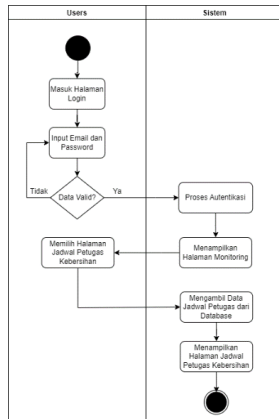
2.2.3 Activity diagram proses monitoring



Gambar 4. Activity diagram proses monitoring

Activity diagram proses monitoring dapat dilihat pada Gambar 4. Pada proses *monitoring*, pengguna yang telah berhasil login dengan meng-input *E-mail* dan *password* selanjutnya akan diarahkan menuju halaman *monitoring*. Sistem secara otomatis akan mengambil data pembacaan sensor untuk tiap paramater dan menampilkan hasilnya pada layar *monitoring*.

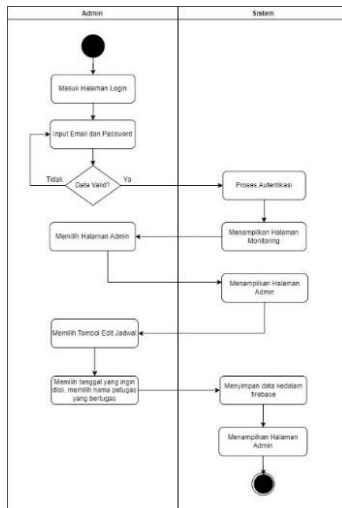
2.2.4 Activity diagram proses menampilkan halaman jadwal petugas



Gambar 5. Activity diagram proses menampilkan informasi pegawai

Activity diagram proses menampilkan jadwal petugas dapat dilihat pada Gambar 5. Pada proses menampilkan halaman jadwal petugas kebersihan, setelah pengguna berhasil login dan layar menampilkan halaman monitoring, pengguna dapat memilih halaman jadwal petugas kebersihan, kemudian sistem secara otomatis mengambil data jadwal petugas kebersihan dari firebase dan ditampilkan pada halaman jadwal petugas kebersihan.

2.2.5 Activity diagram proses edit jadwal petugas

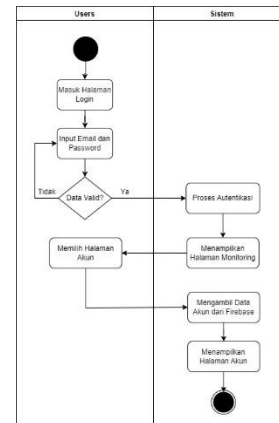


Gambar 6. Activity diagram proses edit jadwal petugas

Activity diagram proses edit jadwal petugas kebersihan dapat dilihat pada Gambar 6. Pada proses edit jadwal petugas kebersihan, admin sebagai pengguna terlebih dahulu diminta untuk login, kemudian setelah autentikasi berhasil dan muncul halaman monitoring admin dapat memilih halaman admin. Setelah halaman admin tampil di layar, selanjutnya dipilih tombol edit jadwal dimana

admin akan diminta untuk memilih tanggal sesuai yang diinginkan dan mengisi nama petugas yang akan bertugas pada tanggal tersebut. Data tersebut kemudian disimpan dalam firebase kemudian akan ditampilkan pada halaman jadwal petugas.

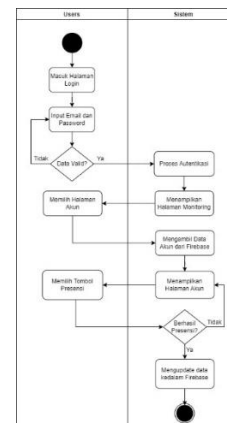
2.2.6 Activity diagram proses menampilkan halaman akun



Gambar 7. Activity diagram proses menampilkan halaman akun

Activity diagram proses menampilkan halaman akun dapat dilihat pada Gambar 7. Pada proses ini menampilkan halaman akun, pengguna yang sudah berhasil login dan masuk halaman monitoring dapat memilih halaman akun, kemudian sistem akan otomatis mengambil data akun dari firebase. Data tersebut selanjutnya akan tampil pada halaman akun pengguna.

2.2.7 Activity diagram proses presensi

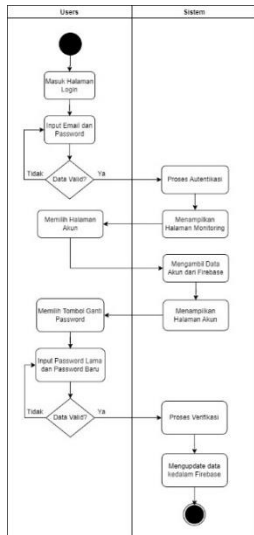


Gambar 8. Activity diagram proses presensi

Activity diagram proses presensi petugas dapat dilihat pada Gambar 8. Pada proses presensi petugas, pengguna yang sudah berhasil login dan masuk ke halaman monitoring dapat memilih halaman akun. Lalu sistem akan menampilkan data akun yang diambil dari firebase, lalu agar pengguna dapat melakukan presensi maka harus menekan tombol presensi. Setelah menekan tombol

presensi maka sistem akan melakukan verifikasi dengan *bluetooth* yang sudah terintegrasi dengan *smart toilet* lalu jika sudah berhasil verifikasi maka sistem akan mengirimkan data presensi ke *firebase*.

2.2.8 Activity diagram proses ganti password



Gambar 9. Activity diagram proses ganti password

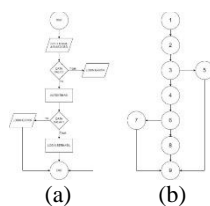
Activity diagram proses ganti password dapat dilihat pada Gambar 9. Pada proses ganti password, pengguna yang sudah berhasil login dan masuk ke halaman monitoring dapat memilih halaman akun. Lalu sistem akan menampilkan data akun yang diambil dari firebase, lalu agar pengguna dapat mengubah password maka harus menekan tombol ganti password. Setelah menekan tombol password maka pengguna harus mengisi form yang berisikan password lama yang ingin diubah dan password baru sebagai password pengganti.

3 Hasil dan Pembahasan

3.1 Pengujian White-Box Aplikasi Android

3.1.1 Pengujian White-Box Login

Pengujian white box bertujuan sebagai pengujian struktural atau pengujian perangkat lunak di mana pengujian dilakukan dengan memahami struktur internal dari kode sumber program, dengan memeriksa logika internal, alur kontrol, dan struktur data dari program yang diuji [11]. Berikut merupakan diagram flowchart dan flowgraph pada proses login yang ditunjukkan pada Gambar 10



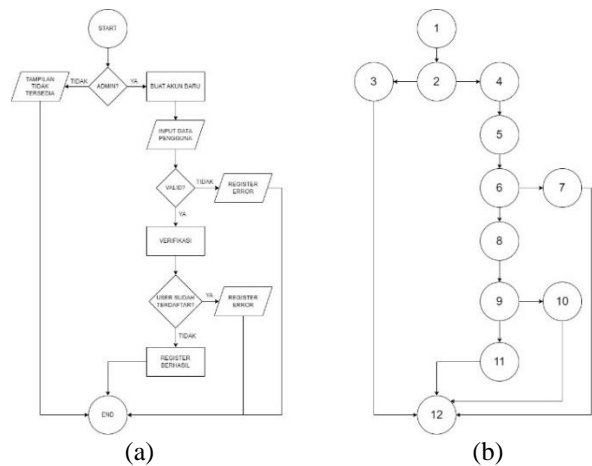
Gambar 10. (a) Flowchart, dan (b) flowgraph pada proses login

Pada Gambar 10 (b) terdapat 9 node dan 10 edge pada flowgraph proses login, sehingga dapat dihitung Cyclomatic Complexity (CC) yang ditunjukkan pada Tabel 1 dibawah ini.

Tabel 1. Hasil pengujian white-box login

Nilai Cyclomatic Complexity (CC)	Path pada flowgraph	Hasil Uji
E = 12	Path 1 = 1-2-3-4-6-8-9	Login berhasil dilakukan
N = 9	Path 2 = 1-2-3-5-9	Login tidak berhasil dilakukan karena data yang dimasukkan tidak valid
= E - N + 2		
= 10 - 9 + 2		
= 3	Path 3 = 1-2-3-4-6-7-9	Login tidak berhasil karena autentikasi gagal dan muncul alert dialog "Terjadi kesalahan saat masuk. Silakan coba lagi."

3.1.2 Pengujian White-Box Register



Gambar 11. (a) Flowchart, dan (b) flowgraph pada proses register

Pada Gambar 11 (b) terdapat 12 node dan 14 edge pada flowgraph proses registrasi, sehingga dapat dihitung Cyclomatic Complexity (CC) yang ditunjukkan pada Tabel 2 dibawah ini.

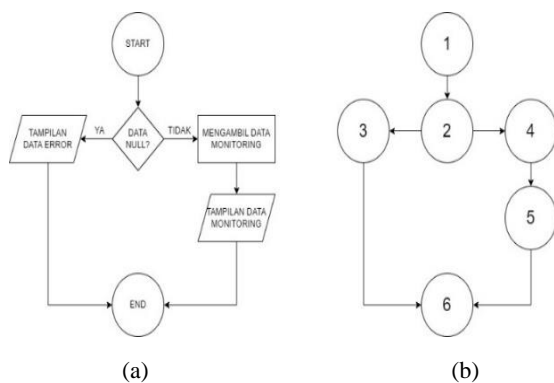
Tabel 2. Hasil pengujian *white-box register*

Nilai <i>Cyclomatic Complexity</i> (CC)	<i>Path</i> pada <i>flowgraph</i>	Hasil Uji
E = 12 N = 9 CC = E - N + 2 = 14 - 12 + 2 = 4	<i>Path</i> 1 = 1-2-4-5-6-8-9-11-12 <i>Path</i> 2 = 1-2-3-12 <i>Path</i> 3 = 1-2-4-5-6-7-12 <i>Path</i> 4 = 1-2-4-5-6-8-9-10-12	<i>Registrasi</i> berhasil dilakukan <i>Registrasi</i> tidak dapat dilakukan karena posisi pengguna bukan admin <i>Registrasi</i> gagal dilakukan karena data yang dimasukkan tidak valid <i>Register</i> gagal dilakukan karena <i>email</i> pengguna yang didaftarkan sudah terdaftar sebelumnya

Hasil pengujian *white-box register* dapat dilihat pada Tabel 2. Dapat disimpulkan jika proses *register* memiliki nilai CC yaitu 4 *path*.

3.1.3 Pengujian *White-Box* Pengambilan Data Monitoring

Berikut merupakan diagram *flowchart* dan *flowgraph* pada proses *login* yang ditunjukkan pada Gambar 12.



Gambar 12. (a) *Flowchart*, dan (b) *flowgraph* pada proses *monitoring*

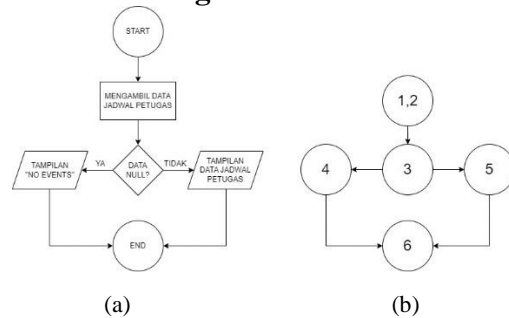
Pada Gambar 12 (b) terdapat 6 *node* dan 6 *edge* pada *flowgraph* proses *registrasi*, sehingga dapat dihitung *Cyclomatic Complexity* (CC) yang ditunjukkan pada Tabel 3 dibawah ini.

Tabel 3 Hasil pengujian *white-box monitoring*

Nilai <i>Cyclomatic Complexity</i> (CC)	<i>Path</i> pada <i>flowgraph</i>	Hasil Uji
E = 12 N = 9 CC = E - N + 2 = 6 - 6 + 2 = 2	<i>Path</i> 1 = 1-2-4-5-6 <i>Path</i> 2 = 1-2-3-6	Pengambilan data <i>monitoring</i> berhasil dilakukan Pengambilan data <i>monitoring</i> gagal dilakukan karena terdapat data <i>monitoring</i> pada <i>firebase</i> yang <i>null</i>

Hasil pengujian *white-box monitoring* dapat dilihat pada Tabel 3. Dapat disimpulkan jika proses *monitoring* memiliki nilai CC yaitu 2 *path*.

3.1.4 Pengujian *White-Box* Pengambilan Data Jadwal Petugas



Gambar 13 (a) *Flowchart*, dan (b) *flowgraph* pada proses *monitoring*

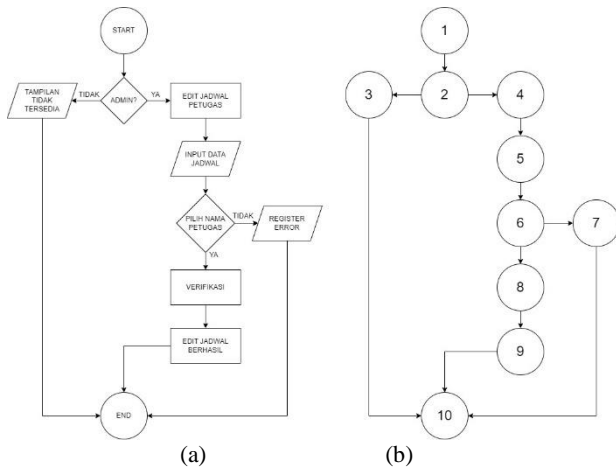
Pada Gambar 13 (b) terdapat 5 *node* dan 5 *edge* pada *flowgraph* proses *jadwal* petugas, sehingga dapat dihitung *Cyclomatic Complexity* (CC) yang ditunjukkan pada Tabel 4 dibawah ini.

Tabel 4 Hasil pengujian *white-box* jadwal petugas

Nilai <i>Cyclomatic Complexity</i> (CC)	<i>Path</i> pada <i>flowgraph</i>	Hasil Uji
E = 12 N = 9 CC = E - N + 2 = 5 - 5 + 2 = 2	<i>Path</i> 1 = 1-2--5-6 <i>Path</i> 2 = 1-2-3-4-6	Pengambilan data <i>jadwal</i> petugas berhasil dilakukan Pengambilan <i>jadwal</i> petugas gagal dilakukan karena <i>jadwal</i> petugas pada <i>firebase</i> <i>null</i>

3.1.5 Pengujian White-Box Edit Jadwal Petugas

Berikut merupakan diagram *flowchart* dan *flowgraph* pada proses *edit* jadwal petugas yang ditunjukkan pada Gambar 14.



Gambar 14 (a) *Flowchart*, dan (b) *flowgraph* pada proses *edit* jadwal petugas

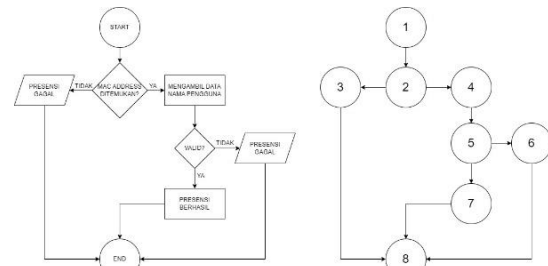
Tabel 5 Hasil pengujian *white-box edit* jadwal

Nilai Cyclomatic Complexity (CC)	Path pada <i>flowgraph</i>	Hasil Uji
$E = 11$	Path 1 = 1-2-4-5-6-8-9-10	<i>Edit</i> jadwal petugas berhasil dilakukan
$N = 10$	Path 2 = 1-2-3-10	<i>Edit</i> jadwal petugas tidak berhasil dilakukan karena posisi pengguna bukan admin
$CC = E - N + 2 = 11 - 10 + 2 = 3$	Path 3 = 1-2-4-5-6-7-10	<i>Edit</i> jadwal petugas tidak berhasil karena pengguna tidak mengisi nama petugas

Hasil pengujian *white-box edit* jadwal dapat dilihat pada Tabel 5. Dapat disimpulkan jika proses *monitoring* memiliki nilai CC yaitu 3 *path*.

3.1.6 Pengujian White-Box Presensi

Berikut merupakan diagram *flowchart* dan *flowgraph* pada proses *edit* jadwal petugas yang ditunjukkan pada Gambar 15.



Gambar 15 (a) *Flowchart*, dan (b) *flowgraph* pada proses presensi

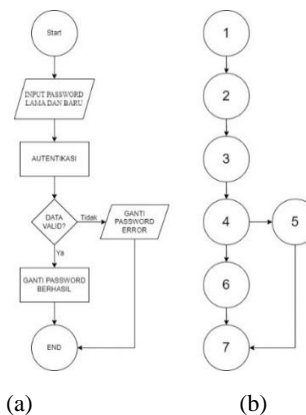
Tabel 6 Hasil pengujian *white-box presensi*

Nilai Cyclomatic Complexity (CC)	Path pada <i>flowgraph</i>	Hasil Uji
$E = 11$	Path 1 = 1-2-4-5-7-8	Presensi berhasil dilakukan
$N = 10$	Path 2 = 1-2-3-8	Presensi tidak berhasil dilakukan karena <i>MAC Address smart toilet</i> tidak ditemukan
$CC = E - N + 2 = 11 - 10 + 2 = 3$	Path 3 = 1-2-4-5-6-8	Presensi tidak berhasil karena pengguna tidak cocok

Hasil pengujian *white-box presensi* dapat dilihat pada Tabel 6. Dapat disimpulkan jika proses *monitoring* memiliki nilai CC yaitu 3 *path*.

3.1.7 Pengujian White-Box Ganti Password

Berikut merupakan diagram *flowchart* dan *flowgraph* pada proses *edit* jadwal petugas yang ditunjukkan pada Gambar 14.



Gambar 16 (a) *Flowchart*, dan (b) *flowgraph* pada proses presensi

Pada Gambar 16 (b) terdapat 7 *node* dan 7 *edge* pada *flowgraph* proses *registrasi*, sehingga dapat dihitung

Cyclomatic Complexity (CC) yang ditunjukkan pada Tabel 7 dibawah ini.

Tabel 7 Hasil pengujian *white-box* presensi

Nilai Cyclomatic Complexity (CC)	Path pada flowgraph	Hasil Uji
E = 12 N = 9 CC = E - N + 2 = 7 - 7 + 2 = 2	Path 1 = 1-2-3-4-6-7 Path 2 = 1-2-3-6	Ganti <i>password</i> berhasil dilakukan Ganti <i>password</i> gagal dilakukan karena data yang diisi tidak valid

Hasil pengujian *white-box* presensi dapat dilihat pada Tabel 7 dapat disimpulkan jika proses *monitoring* memiliki nilai CC yaitu 2 *path*.

3.2 Pengujian *Black-Box* Aplikasi Android

Pengujian *black box* adalah jenis pengujian perangkat lunak dimana dilakukan tanpa memperhatikan struktur internal kode dari program. Tujuannya adalah mengevaluasi kualitas dan keandalan perangkat lunak dari perspektif pengguna akhir, sehingga memastikan bahwa perangkat lunak berfungsi dengan benar dan memenuhi kebutuhan pengguna [12].

Tabel 8 Hasil pengujian *black-box* Aplikasi Android

No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1	Login	- Memasukkan <i>email</i> dan <i>password</i> yang terdaftar - Jika berhasil, pengguna dapat masuk monitoring	Berhasil
2	Registrasi	- Mengisi semua data dengan valid - Berhasil <i>register</i> , dan akun tersimpan firebase	Berhasil
3	Edit Jadwal	- Mengisi tanggal, dan nama petugas yang ingin dipilih.	Berhasil
4	Presensi	- Presensi dengan jarak yang terjangkau dengan <i>smart toilet</i>	Berhasil
5	Ganti Password	- Mengisi <i>password</i> lama dengan benar - Mengisi <i>password</i> baru dengan format yang benar	Berhasil

Hasil pengujian *black-box* dapat dilihat pada Tabel 8. Dapat disimpulkan bahwa seluruh hasil pengujian menggunakan metode *black-box* dapat berjalan secara keseluruhan.

3.3 Pengujian Kompabilitas Aplikasi

Uji kompabilitas adalah proses pengujian perangkat lunak yang bertujuan untuk memastikan bahwa aplikasi berfungsi dengan baik di berbagai perangkat keras, sistem operasi, dan kondisi jaringan yang berbeda, sehingga pengguna dapat menggunakan secara konsisten dan optimal di berbagai *platform* [13]. Berikut merupakan hasil pengujian aplikasi pada 3 perangkat Android yang berbeda seperti yang ditunjukkan pada Tabel 9.

Tabel 9 Hasil pengujian kompabilitas aplikasi

No	Skenario Pengujian	Hasil Pengujian Perangkat		
		Realme Narzo 30A	Redmi K20 Pro	OPPO Reno 5F
1	Install Aplikasi	V	V	V
2	Menampilkan <i>SplashScreen</i>	V	V	V
3	Melakukan <i>login</i> pada halaman <i>login</i>	V	V	V
4	Menampilkan data <i>monitoring</i> pada halaman <i>monitoring</i>	V	V	V
5	Menampilkan data jadwal petugas pada halaman jadwal petugas	V	V	V
6	Menampilkan informasi pengawai pada halaman admin	V	V	V
7	Melakukan <i>edit</i> jadwal melalui <i>dialog edit</i> jadwal pada halaman admin	V	V	V
8	Melakukan buat akun baru/ <i>register</i> melalui <i>dialog register</i> pada halaman admin	V	V	V
9	Menampilkan informasi pribadi pada halaman akun	V	V	V
10	Melakukan ganti <i>password</i> melalui <i>dialog</i> ganti <i>password</i> pada halaman akun	V	V	V

3.4 Pengujian User Acceptance Test

Pengujian *user acceptance test* (UAT) adalah jenis pengujian perangkat lunak yang dilakukan oleh pengguna akhir untuk menentukan apakah aplikasi sudah sesuai dengan kebutuhan. Pengujian ini dilakukan sebelum aplikasi dirilis kepada pengguna akhir [14]. Dalam tahap pengujian ini, diperlukan partisipasi dari 10 responden yang bertindak sebagai pengguna aplikasi yang dapat dilihat pada Tabel 10.

Tabel 10 Responden *User Acceptance Test*

No	Nama Responden	Usia	Pekerjaan
1	Savero Pakavi Zahrudin	22	Software Engineer
2	Naufal Wijaya	23	Software Engineer
3	Warsito	29	Petugas Kebersihan
4	Misbakhul Munir	34	Petugas Kebersihan
5	Sugiyanto Irawan	28	Petugas Kebersihan
6	M. Sehatian Tawa'qi	22	Mahasiswa
7	Tobias Kusuma W.	22	Mahasiswa
8	Aditya Radian Tanata	21	Mahasiswa
9	Tristan Rizky Setyawan	22	Mahasiswa
10	Briyantama Wahab	29	Mahasiswa

Hasil pengujian UAT dapat dilihat pada Tabel 11. Responden terdiri dari *software engineer*, petugas kebersihan teknik elektro dan juga mahasiswa yang diberikan 10 pernyataan bernilai positif tentang aplikasi *monitoring smart toilet*.

Tabel 11 Hasil Pengujian *User Acceptance Test*

No	Pertanyaan	SS	S	KS	TS	STS
1	Tampilan pada <i>login</i> sesuai dengan yang saya harapkan	3	7			
2	Tampilan pada halaman <i>monitoring</i> sesuai dengan yang saya harapkan	5	5			
3	Tampilan pada halaman jadwal petugas sesuai dengan yang saya harapkan	3	7	-	-	-
4	Tampilan pada halaman admin sesuai dengan yang saya harapkan	5	4	1	-	-

Tabel 11 Tabel Lanjutan

No	Pertanyaan	SS	S	KS	TS	STS
5	Tampilan pada <i>dialog edit</i> jadwal sesuai dengan yang saya harapkan	4	6	-	-	-
6	Tampilan pada <i>register</i> sesuai dengan yang saya harapkan	5	5	-	-	
7	Tampilan pada halaman akun sesuai dengan yang saya harapkan	5	5	-	-	-
8	Tampilan pada <i>dialog</i> presensi sesuai dengan yang saya harapkan	4	4	2	-	-
9	Tampilan pada <i>dialog</i> ganti <i>password</i> sesuai dengan yang saya harapkan	4	5	1	-	-
10	Apakah aplikasi ini memiliki tampilan yang menarik serta responsif?	4	4	2	-	-
9	Apakah aplikasi ini dapat dengan mudah dimengerti dan digunakan?	5	4	-	-	-
10	Tampilan pada halaman jadwal petugas sesuai dengan yang saya harapkan	9	1	-	-	-
	Total	49	54	7		

3.5 Pengujian Penerimaan Notifikasi

Pengujian notifikasi ini adalah proses pengujian perangkat lunak yang difokuskan untuk memastikan bahwa notifikasi yang dihasilkan berupa pemberitahuan, pesan atau peringatan kepada *user* untuk memberikan informasi penting atau menerima tindakan tertentu [15]. Pengujian ini akan dijalankan sebanyak empat kali dengan kondisi yang berbeda, dan durasi waktu akan dihitung dengan data yang diambil dari *Postman* dan *terminal Visual Studio Code* selama proses *debugging*. Mulai dari saat pengiriman hingga notifikasi muncul pada perangkat, hasil perhitungan durasi ditunjukkan dalam Tabel 12.

Tabel 12 Hasil pengujian penerimaan notifikasi

Kondisi	Waktu Pengiriman Notifikasi	Waktu Penerimaan Notifikasi	Selisih Waktu
Sabun dibawah 10%	01/18/2024 02:30:21:29	01/18/2024 02:30:22:06	00:765 detik
	6	1	

Tabel 12 Tabel Lanjutan

Kondisi	Waktu Pengiriman Notifikasi	Waktu Penerimaan Notifikasi	Selisih Waktu
Tisu dibawah 10%	01/18/2024 02:32:38:15	01/18/2024 02:32:39:60	01:455 detik
Status kenyamanan dibawah 40%	01/18/2024 02:33:32:06	01/18/2024 02:33:32:95	00:888 detik
	2	0	
		Rata-rata	02:633

4. KESIMPULAN

Aplikasi Monitoring Smart Toilet telah berhasil dirancang berbasis Android, menyediakan fitur yang mendukung petugas kebersihan smart toilet dalam melakukan pemantauan kondisi keseluruhan toilet pintar, termasuk jadwal petugas dan presensi menggunakan teknologi Bluetooth. Hasil pengujian White Box menunjukkan bahwa setiap fungsi program pada aplikasi Flutter memiliki nilai Cyclomatic Complexity (CC) rata-rata sebesar 2,7. Pengujian Black Box menyatakan bahwa setiap fitur aplikasi berjalan dengan baik, sementara uji kompatibilitas menunjukkan aplikasi berfungsi optimal pada tiga perangkat dan dua sistem operasi yang berbeda. Penerimaan notifikasi dalam waktu kurang dari 5 detik, dan hasil User Acceptance Test menunjukkan bahwa sebagian besar responden (93,6%) setuju atau sangat setuju dengan aplikasi ini. Dengan demikian, antarmuka Aplikasi Monitoring Smart Toilet dengan Firebase dapat dianggap berjalan dengan baik berdasarkan hasil uji White Box, Black Box, Kompabilitas, dan User Acceptance.

REFERENSI

- [1] D. Iossifova, "Urban (Sanitation) Transformation in China: a Toilet Revolution and its socio-ecotechnical entanglements," in *Urban transformations and public health in the emergent city*, Manchester University Press, 2020, pp. 102–122.
- [2] F. H. Wibowo and P. Sudira, "Prototype Smart Bathroom based Arduino UNO," *E-JPTE (Jurnal Elektronik Pendidikan Teknik Elektronika)*, vol. 6, no. 8, pp. 81–89, 2017.
- [3] A. Rahmania, "The Risk Matrix of Occupational Health and Safety on Cleaning Service Occupation in Universitas X Ponorogo," *Medical Technology and Public Health Journal*, vol. 7, no. 1, pp. 79–88, 2023.
- [4] F. Adilia, A. Rakhmatsyah, and A. G. Putrada, "Implementasi Toilet Pintar Berbasis Mikrokontroler," *eProceedings of Engineering*, vol. 3, no. 3, 2016.
- [5] M. N. Sari, A. Rahmawati, S. Widodo, and S. S. Hidayat, "Smart Toilet Dilengkapi Sistem Pemantauan Berbasis Website," in *Prosiding Seminar Nasional NCIET*, 2021, pp. 1–12.
- [6] M. Alfarisi, F. Y. Limpraptono, and M. I. Ashari, "Rancang Bangun Sistem Monitoring dan Pembersih Udara Otomatis pada Toilet Umum Berbasis IoT," *Prosiding SENIATI*, vol. 6, no. 1, pp. 46–53, 2022.
- [7] E. Prastika and A. Hadi, "Toilet Berdiri Otomatis Berbasis Sensor PIR (Passive Infrared) dan Sensor Temperatur," in *Seminar Nasional Industri dan Teknologi*, 2019, pp. 365–373.
- [8] R. Rosaly and A. Prasetyo, "Pengertian Flowchart Beserta Fungsi dan Simbol-simbol Flowchart yang Paling Umum Digunakan," *Program Studi Teknik Informatika Politeknik Purbaya*, 2019.
- [9] Y. Yunitasari and S. Sintaro, "Pengerak Kamera Dengan 2in1 Control (Manual Dan Otomatis) Menggunakan Aplikasi Android," *Jurnal Teknologi dan Sistem Tertanam*, vol. 2, no. 2, pp. 36–45, 2021.
- [10] T. Arianti, A. Fa'izi, S. Adam, and M. Wulandari, "Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram UML (Unified Modelling Language)," *Jurnal Ilmiah Komputer Terapan dan Informasi*, vol. 1, no. 1, pp. 19–25, 2022.
- [11] H. Gusdevi, S. Kuswayati, M. Iqbal, M. F. A. Bakar, N. Novianti, and R. Ramadan, "Pengujian White-Box Pada Aplikasi Debt Manager Berbasis Android," *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika*, vol. 4, no. 1, pp. 11–22, 2022.
- [12] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Pengujian black box testing pada aplikasi action & strategy berbasis android dengan teknologi phonegap," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 3, no. 2, pp. 206–210, 2018.
- [13] A. K. Almasyhur, H. Tolle, and A. H. Brata, "Pengembangan aplikasi mobile tracer study menggunakan platform Android," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 11, pp. 5402–5409, 2018.
- [14] M. A. Bastari, D. Darmansah, and D. P. Rakhmadani, "Sistem Informasi Jasa Cuci Interior Rumah dan Mobil Menggunakan Metode User Acceptance Test," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 2, pp. 305–315, 2022.
- [15] A. Lubis, H. Syahfitri, I. R. Munthe, and R. Pane, "Infus Desain Notifikasi Dengan Aplikasi Media Sosial Berbasis Internet of Things (IOT)," *Jurnal Teknik Informatika UNIKA Santo Thomas*, vol. 6, no. 1, pp. 117–125, 2021.