

PERANCANGAN APLIKASI MOBILE DAN DATABASE PADA SISTEM OTOMASI BUDIDAYA MAGGOT BERBASIS INTERNET OF THINGS

Naufal Wijaya^{1*)}, Enda Wista Sinuraya² dan Eko Handoyo³

¹²³Departemen Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, Semarang, Indonesia

^{*)} E-mail: naufalwijaya@students.undip.ac.id

Abstrak

Maggot BSF merupakan salah satu budidaya peternakan yang dapat sebagai pengurai sampah organik dan sebagai pakan ternak. Pembudidayaan ini juga dapat menjadi salah satu usaha untuk mewujudkan SDG's terutama pada poin pertama, kedua, kelima, dan kedua belas. Selain memiliki faktor pakan, maggot BSF memiliki faktor suhu dan kelembapan yang optimal untuk dapat mempercepat siklus hidupnya. Maka dari itu dibutuhkan sistem otomasi untuk mengatur suhu dan kelembapan dari media kandang maggot BSF dalam waktu malam ataupun siang hari. Permasalahan tersebut dapat diatasi dengan pembuatan sistem kontrol suhu dan kelembapan otomatis berbasis *Internet of Things* sehingga peternak dapat memantau suhu dan kelembapan media tanpa harus ke kandang setiap saat. React Native menyediakan *framework* yang bagus untuk pembuatan aplikasi android sebagai sistem *monitoring* dan *controlling* jarak jauh melalui *internet of things* sehingga dapat membantu peternak maggot BSF. Aplikasi android akan menampilkan *record data* dalam harian dan tahap fase yang sudah diolah menjadi rata-rata pada hari tersebut. Selain itu sistem juga perlu adanya otomasi dalam penyemprotan media lalat sehingga peternak tidak perlu menyemprot terlalu sering.

Kata kunci: Maggot bsf, React Native, aplikasi android, *internet of things*

Abstract

Maggot BSF is one of the livestock cultivation practices that can serve as an organic waste decomposer and animal feed. This cultivation method also contributes to achieving SDGs, especially goals one, two, five, and twelve. In addition to the feed factor, maggot BSF requires optimal temperature and humidity levels to accelerate its life cycle. Hence, an automated system is needed to regulate the temperature and humidity of the maggot BSF breeding medium, both during the day and night. This issue can be addressed by developing an Internet of Things (IoT)-based automated temperature and humidity control system, allowing farmers to monitor these factors remotely without frequent visits to the breeding site. React Native provides an excellent framework for developing an Android application for remote monitoring and controlling through the Internet of Things, thereby assisting maggot BSF farmers. The Android application will display daily data records and processed phase averages for that day. Moreover, the system should incorporate automation for fly media spraying, reducing the need for frequent manual intervention by the farmers.

Keywords: Maggot bsf, React Native, android application, *internet of things*

1. Pendahuluan

Maggot adalah salah satu budidaya peternakan yang memiliki dua keuntungan yaitu bisa sebagai pengurai sampah dan sebagai pakan ternak. Maggot merupakan larva dari lalat BSF (*Black Soldier Fly*) dan bukanlah termasuk hama penyakit bagi manusia. Pembudidayaan maggot BSF juga dapat menjadi salah satu upaya untuk mewujudkan beberapa poin *Sustainable Development Goals*. Poin pertama, kedua, kelima dan kedua belas [1].

Revolusi Industri 4.0 merupakan suatu perkembangan tahap industri setelah 3.0 dimana pertama kali dijalankan pada Hannover Fair, 4-8 April 2011. Istilah ini dipakai

pemerintah Jerman untuk meningkatkan kemajuan di bidang industri ke tingkat selanjutnya, dengan bantuan teknologi terutama *Internet Of Things*. [2] Sekarang perkembangan teknologi internet sudah diterapkan untuk produksi di dunia industri pada negara-negara maju, terutama Amerika Serikat dan negara-negara di Eropa Barat. [3]

Pada penelitian sebelumnya hanya masih ada pemecahan masalah dengan satu parameter saja yaitu suhu dengan lampu sebagai aktuatornya. Sehingga jika suhu suatu media terlalu panas maka tidak ada hal yang bisa dilakukan secara otomatis. [4]

Dari permasalahan kondisi tersebut dan penelitian sebelumnya dalam pembudidayaan maggot BSF perlu diperhatikan kondisi dari suhu serta kelembapan dari kandang maggot. Dalam membantu menyelesaikan permasalahan tersebut, perlu dibuatkan sistem yang mampu menjaga suhu dan kelembapan kandang dalam kondisi stabil. Tujuan dari sistem otomatis ini untuk meningkatkan efektifitas dalam pembudidayaan maggot.

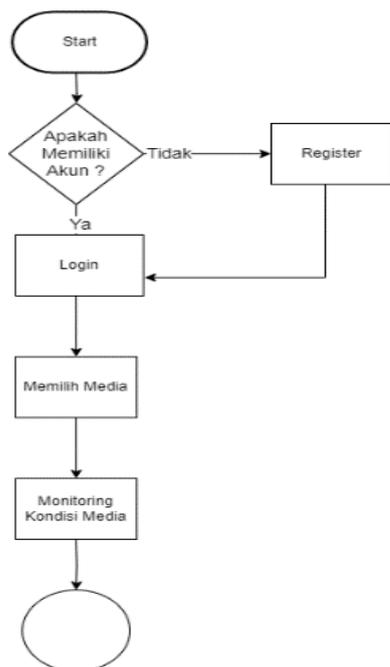
Terdapat dua media untuk penerapan teknologi informasi yaitu melalui aplikasi *mobile* atau melalui *website*. Pada penggunaan antara aplikasi *mobile* dan *website* didapatkan hasil bila aplikasi *mobile* lebih efektif dalam pemakaiannya daripada *website*. [5] Maka sistem ini nantinya akan menggunakan aplikasi *mobile* sebagai *controlling* dan *monitoring*.

Pembuatan sistem ini bertujuan untuk membantu pembudidaya maggot BSF melakukan budidaya dengan lebih efisien. Dengan adanya sistem ini, dapat memudahkan pembudidaya dalam memonitor suhu dan mengontrol lampu pada kandang maggot BSF dengan menggunakan *Smartphone* tanpa harus menyita waktu dan tenaga ataupun mengganggu rutinitas sehari-hari.

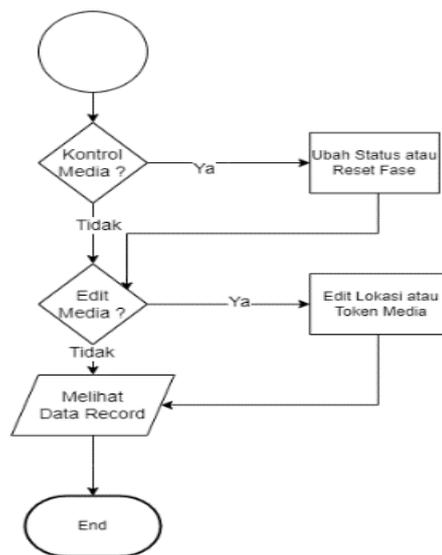
2. Perancangan Sistem

2.1. Arsitektur Software

Berikut *flowchart* aplikasi android yang ditunjukkan pada Gambar 1.



Gambar 1 Flowchart Aplikasi Android



Gambar 1 (Lanjutan)

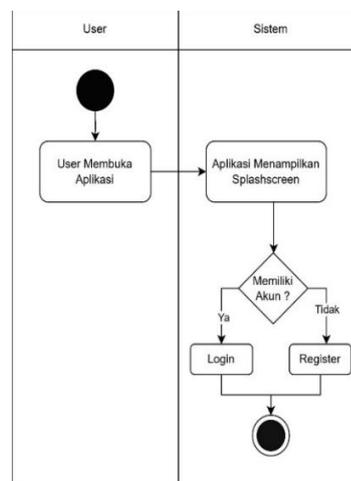
Pada tahap awal *user* aplikasi akan melakukan *login*, jika belum memiliki akun, *user* dapat melakukan *register* terlebih dahulu. Setelah itu *user* akan masuk ke media yang akan dipantau dan dikontrol. Selain itu *user* juga dapat melakukan edit detail media dan juga melihat data *record* pada media tersebut.

2.2. Activity Diagram Software

Pada perancangan aplikasi android MAGGIOT, setelah pembuatan *Use Case Diagram* perlu adanya *Activity Diagram* sehingga dapat diketahui alur proses dari masing-masing proses.

2.2.1. Activity Diagram Proses Awal Pembukaan Aplikasi

Berikut *activity diagram* proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 2.

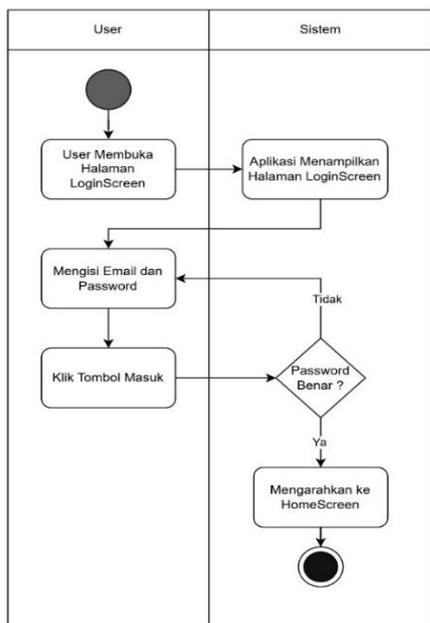


Gambar 2 Activity Diagram pembukaan aplikasi

Setelah aplikasi dibuka oleh *user*, aplikasi akan menampilkan halaman paling awal yaitu *SplashScreen* yang berisi logo dari Aplikasi MAGGIOT. Setelah itu apakah *user* memiliki akun atau tidak, jika sudah maka *user* akan melakukan proses *login* dan jika belum memiliki akun, *user* harus melakukan proses *register*.

2.2.2. Activity Diagram Proses Login

Berikut *activity* diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 3.

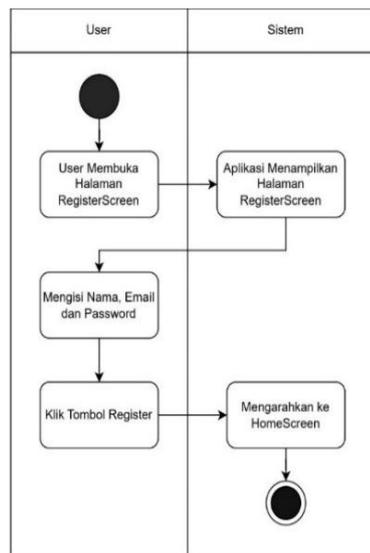


Gambar 3 Activity Diagram proses login

Berdasarkan Gambar 3 dapat diketahui jika proses *login*, *user* dapat membuka halaman *LoginScreen* sehingga aplikasi akan menampilkan tampilan halaman *LoginScreen* dimana *user* harus menuliskan *email* dan *password* yang telah terdaftar lalu menekan tombol masuk. Jika *password* benar maka sistem akan mengarahkan *user* ke halaman *HomeScreen*. Sedangkan jika salah maka *user* harus memasukkan kembali *email* dan *password* yang benar.

2.2.3. Activity Diagram Proses Register

Berikut *activity* diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 4.

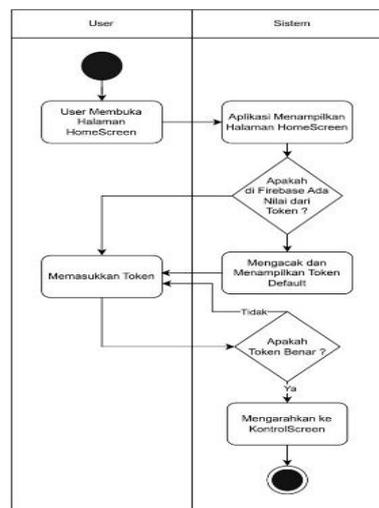


Gambar 4 Activity Diagram proses register

Pada proses *register*, *user* dapat membuka halaman *LoginScreen* sehingga aplikasi akan menampilkan tampilan halaman *RegisterScreen* dimana *user* harus menuliskan nama, *email* dan *password* yang akan didaftarkan terdaftar lalu menekan tombol *register*. Setelah itu sistem akan mengarahkan *user* ke halaman *HomeScreen*.

2.2.4. Activity Diagram Proses Masuk Ke Media

Berikut *activity* diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 5.

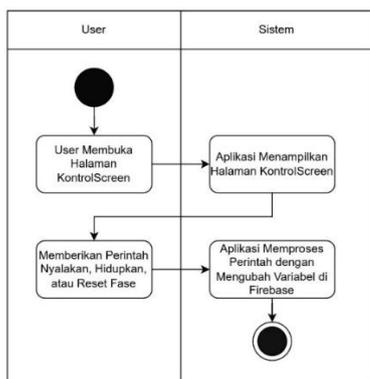


Gambar 5 Activity Diagram proses masuk ke media. Pada proses masuk ke media, *user* dapat membuka halaman *HomeScreen* sehingga aplikasi akan menampilkan tampilan halaman *HomeScreen* dimana *user* harus menuliskan token dari media tersebut, jika token

masih *default* maka akan ditampilkan token tersebut sehingga token tidak akan ditampilkan jika token sudah diubah. Setelah itu *user* dapat menekan tombol masuk sehingga sistem akan mengarahkan *user* ke halaman *KontrolScreen*.

2.2.5. Activity Diagram pada *KontrolScreen*

Berikut *activity* diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 6.

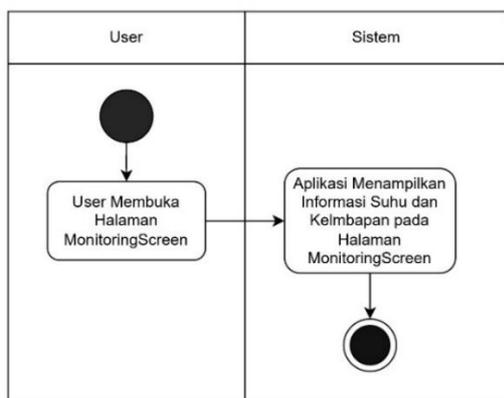


Gambar 6 Activity Diagram halaman *KontrolScreen*

Pada halaman *KontrolScreen*, user dapat membuka halaman *KontrolScreen* sehingga aplikasi akan menampilkan tampilan halaman *KontrolScreen* dimana *user* dapat mengubah status dan *reset* fase maggot BSF.

2.2.6. Activity Diagram pada *MonitoringScreen*

Berikut *activity* diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 7.



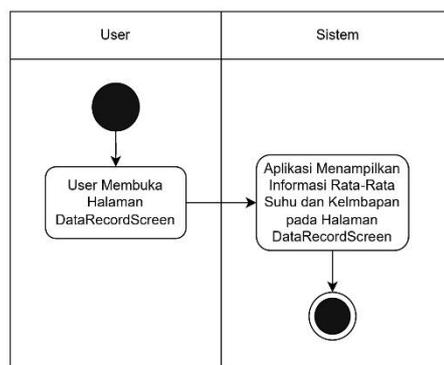
Gambar 7 Activity Diagram halaman *MonitoringScreen*

Pada halaman *MonitoringScreen*, user dapat membuka halaman *MonitoringScreen* sehingga aplikasi akan menampilkan tampilan halaman *MonitoringScreen* dimana

user dapat melihat kondisi suhu dan kelembapan maggot serta penjelasan pada nilai suhu dan kelembapan yang cocok untuk maggot BSF. Setelah itu user dapat menekan tombol untuk pindah ke fitur lain sehingga sistem akan mengarahkan user ke halaman yang dipilih.

2.2.7. Activity Diagram pada *DataRecordScreen*

Berikut *activity* diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 8.

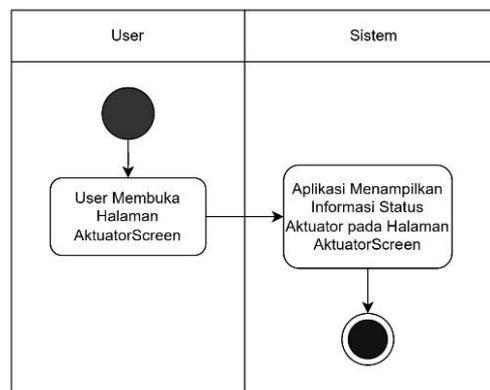


Gambar 8 Activity Diagram halaman *DataRecordScreen*

Pada halaman *DataRecordScreen*, user dapat membuka halaman *DataRecordScreen* sehingga aplikasi akan menampilkan tampilan halaman *DataRecordScreen* dimana *user* dapat melihat rata-rata suhu dan kelembapan maggot pada masing-masing hari dan fase hidup dari Maggot BSF.

2.2.8. Activity Diagram pada *AktuatorScreen*

Berikut *activity* diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 9.

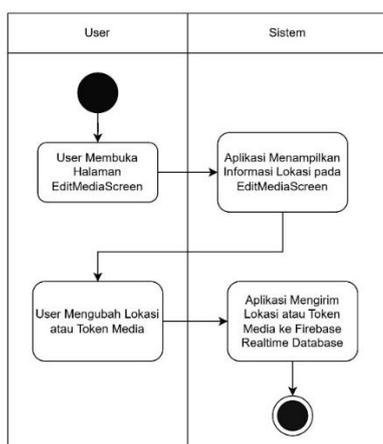


Gambar 9 Activity Diagram halaman *AktuatorScreen*

Pada halaman *AktuatorScreen*, user dapat membuka halaman *AktuatorScreen* sehingga aplikasi akan menampilkan tampilan halaman *AktuatorScreen* dimana user dapat mengetahui status dari 4 aktuator yaitu lampu, kipas, mistmaker dan pompa. Setelah itu user dapat menekan tombol untuk pindah ke fitur lain sehingga sistem akan mengarahkan user ke halaman yang dipilih.

2.2.9. Activity Diagram pada *EditMediaScreen*

Berikut *activity* diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 10.



Gambar 10 Activity Diagram halaman *EditMediaScreen*

Pada halaman *EditMediaScreen*, user dapat membuka halaman *EditMediaScreen* sehingga aplikasi akan menampilkan tampilan halaman *EditMediaScreen*. Pada halaman ini user dapat mengubah lokasi atau token media, sehingga saat sudah dilakukan perubahan token media, maka *HomeScreen* tidak akan menampilkan token default kembali.

2.3. Perancangan Database pada Firebase

Database yang digunakan pada firebase memanfaatkan dua fitur yang telah disediakan firebase yaitu firebase *authentication* untuk manajemen data user dan firebase *realtime database* untuk manajemen data dari media.

Pada firebase *authentication* berisikan data dari user yang diisi oleh user sendiri berupa *email*, *displayName*, dan *password*. Selain itu firebase juga menyediakan data langsung yang dimiliki user yaitu *lastSignInTime*, *creationTime*, *providerId*, *isAnonymous*, *phoneNumber*, *photoURL*, *emailVerified*, dan *tenantId*. Tangkapan layar tampilan *database* pada firebase *authentication* untuk *database user* ditunjukkan pada Gambar 11.



Gambar 11 Tampilan *database* pada *firebase authentication*

Firestore *realtime database* berfungsi sebagai database untuk penyimpanan database dari media yang dijalankan. Terdapat dua *path* pertama untuk penyimpanannya, yaitu *path* untuk detail media dan *path* untuk *record* data dari media. Tangkapan layar tampilan pada *firebase realtime database* untuk *database* dari media maggot ditunjukkan pada Gambar 12.



Gambar 12 Tampilan *database* pada *firebase realtime database*

Pada *path* *sistem1* berisikan detail dari media berupa *devicetoken*, hari, id, kelembapan, kipas, lampu, lokasi, mistmaker, peringatankelembapan, peringatan suhu, periode, sprayer, status, suhu, timereset, dan token. *Path* *devicetoken* berisikan token dari masing-masing device yang terhubung serta total jumlah device yang telah terhubung.

Dalam penyimpanan token masing-masing perangkat, token akan dimasukkan kedalam *firebase realtime database* dengan *path* *sistem1/devicetoken*, pada *path* tersebut, terdapat keterangan token dari masing-masing perangkat dengan *path* *device1*, *device2*, dan seterusnya. Selain itu terdapat *path* *totaldevice* sebagai jumlah perangkat yang sudah terdaftar. Pendaftaran

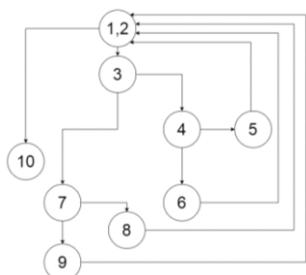
Pada *path* *recorddata1p1* berisikan rekaman data suhu dan kelembapan pada media dengan id 1 dan pada periode 1. Sehingga jika sudah memasuki periode selanjutnya, maka rekaman data suhu dan kelembapan akan berada pada *path* *recorddata1p2* dan seterusnya. Hasil pembacaan hari dari RTC akan menjadi *path* selanjutnya pada penyimpanan data, jika hari pertama maka *path* akan menjadi h1 dan seterusnya. Setelah itu data suhu dan kelembapan akan tersimpan pada *path* suhu dan *path* kelembapan.

3. Pengujian dan Analisis

3.1. Pengujian Aplikasi Android dengan White Box

3.1.1. Pengujian White Box Login User

Berikut *flowgraph* proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 13.



Gambar 13 Flowgraph login user

Pada *flowgraph* yang ditampilkan pada Gambar 13, dapat disimpulkan jika proses *login* memiliki 9 *node* atau titik dan 12 *edge* atau garis pada *flowgraph* sehingga dapat dihitung *Cyclomatic Complexity* (CC) seperti pada Tabel 1.

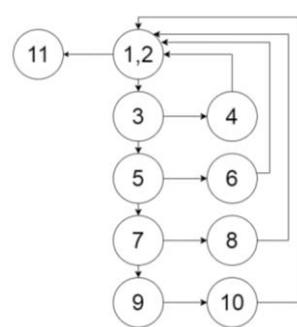
Tabel 1 White Box Login User

Nilai <i>Cyclomatic Complexity</i> (CC)	Path pada <i>flowgraph</i>	Hasil Uji
$E = 12$	Path 1 = 1-2-10	Skenario <i>login</i> berhasil
$N = 9$	Path 2 = 1-2-3-4-5-1-2-11	Skenario <i>login</i> gagal karena <i>email</i> tidak diisi
$V(G) = E - N + 2 = 12 - 9 + 2 = 5$	Path 3 = 1-2-3-4-6-1-2-11	Skenario <i>login</i> gagal karena <i>password</i> tidak diisi
	Path 4 = 1-2-3-7-8-1-2-11	Skenario <i>login</i> gagal karena <i>email</i> tidak ditemukan
	Path 5 = 1-2-3-7-9-1-2-11	Skenario <i>login</i> gagal karena <i>password</i> salah

Berdasarkan Tabel 1 dapat disimpulkan jika proses *login* memiliki nilai CC yaitu 5 *path*.

3.1.2. Pengujian White Box Register User

Berikut *flowgraph* proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 14.



Gambar 14 Flowgraph register user

Pada *flowgraph* yang ditampilkan pada Gambar 14, dapat disimpulkan jika proses *register* memiliki 10 *node* atau titik dan 13 *edge* atau garis pada *flowgraph* sehingga dapat dihitung *Cyclomatic Complexity* (CC) seperti pada Tabel 2.

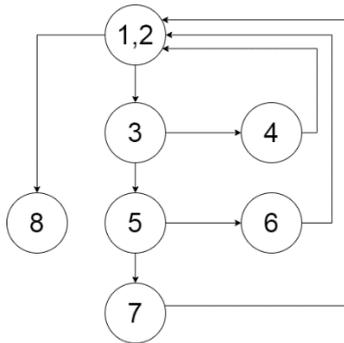
Tabel 2 White Box Register User

Nilai <i>Cyclomatic Complexity</i> (CC)	Path pada <i>flowgraph</i>	Hasil Uji
$E = 13$	Path 1 = 1-2-11	Skenario <i>register</i> berhasil
$N = 10$	Path 2 = 1-2-3-4-1-2-11	Skenario <i>register</i> gagal karena nama tidak diisi
$V(G) = E - N + 2 = 13 - 10 + 2 = 5$	Path 3 = 1-2-3-5-6--1-2-11	Skenario <i>register</i> gagal karena <i>email</i> tidak diisi
	Path 4 = 1-2-3-5-7-8-1-2-11	Skenario <i>register</i> gagal karena <i>password</i> tidak diisi
	Path 5 = 1-2-3-5-7-9-10-1-2-11	Skenario <i>register</i> gagal karena <i>email</i> telah digunakan

Berdasarkan Tabel 2 dapat disimpulkan jika proses *register* memiliki nilai CC yaitu 5 *path*.

3.1.3. Pengujian White Box User Masuk Ke Media

Berikut *activity* diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 15.



Gambar 15 Flowgraph user masuk ke media

Pada flowgraph yang ditampilkan pada Gambar 15, dapat disimpulkan jika proses masuk ke media memiliki 7 node atau titik dan 9 edge atau garis pada flowgraph sehingga dapat dihitung Cyclomatic Complexity (CC) seperti pada Tabel 3.

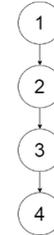
Tabel 3 White Box User Masuk ke Media

Nilai Cyclomatic Complexity (CC)	Path pada flowgraph	Hasil Uji
$E = 9$	Path 1 = 1-2-8	Skenario masuk ke media berhasil dengan token default
$N = 7$		
$V(G) = E - N + 2$		
$= 9 - 7 + 2 = 4$		
	Path 2 = 1-2-3-4-1-2-8	Skenario masuk ke media berhasil dengan token yang telah diubah
	Path 3 = 1-2-3-5-6-1-2-8	Skenario masuk ke media gagal karena token media yang diisi salah
	Path 4 = 1-2-3-5-7-1-2-8	Skenario masuk ke media gagal karena token media yang tidak diisi

Berdasarkan Tabel 3 dapat disimpulkan jika proses masuk ke media memiliki nilai CC yaitu 4 path.

3.1.4. Pengujian White Box User Melakukan Kontrol Status

Berikut activity diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 16.



Gambar 16 Flowgraph user melakukan kontrol status

Pada flowgraph yang ditampilkan pada Gambar 16, dapat disimpulkan jika proses masuk ke media memiliki 4 node atau titik dan 3 edge atau garis pada flowgraph sehingga dapat dihitung Cyclomatic Complexity (CC) seperti pada Tabel 4.

Tabel 4 White Box User Melakukan Kontrol Status

Nilai Cyclomatic Complexity (CC)	Path pada flowgraph	Hasil Uji
$E = 3$	Path 1 = 1-2-3-4	Skenario mengubah status media berhasil
$N = 4$		
$V(G) = E - N + 2$		
$= 3 - 4 + 2 = 1$		

Berdasarkan Tabel 4 dapat disimpulkan jika proses kontrol status memiliki nilai CC yaitu 1 path.

3.1.5. Pengujian White Box User Melakukan Kontrol Reset Fase

Berikut activity diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 17.



Gambar 17 Flowgraph user melakukan reset fase

Pada flowgraph yang ditampilkan pada Gambar 17, dapat disimpulkan jika proses masuk ke media memiliki 5

node atau titik dan 4 edge atau garis pada flowgraph sehingga dapat dihitung Cyclomatic Complexity (CC) seperti pada Tabel 5.

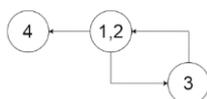
Tabel 5 White Box User Melakukan Reset Fase

Nilai Cyclomatic Complexity (CC)	Path pada flowgraph	Hasil Uji
$E = 4$	Path 1 = 1-2-3-4-5	Skenario memulai ulang fase kontrol media berhasil
$N = 5$		
$V(G)$		
$= E - N + 2$		
$= 4 - 5 + 2$		
$= 1$		

Berdasarkan Tabel 5 dapat disimpulkan jika proses reset fase memiliki nilai CC yaitu 1 path.

3.1.6. Pengujian White Box User Mengubah Detail Media

Berikut activity diagram proses awal pembukaan aplikasi android yang ditunjukkan pada Gambar 17.



Gambar 17 Flowgraph user mengubah detail media

Pada flowgraph yang ditampilkan pada Gambar 17, dapat disimpulkan jika proses masuk ke media memiliki 3 node atau titik dan 3 edge atau garis pada flowgraph sehingga dapat dihitung Cyclomatic Complexity (CC) seperti pada Tabel 6.

Tabel 6 White Box Login User

Nilai Cyclomatic Complexity (CC)	Path pada flowgraph	Hasil Uji
$E = 3$	Path 1 = 1-2-3-4	Skenario mengubah detail media berhasil
$N = 3$		
$V(G)$	Path 2 = 1-2-3-2-4	Skenario mengubah detail media gagal karena tidak ada perubahan
$= E - N + 2$		
$= 3 - 3 + 2$		
$= 2$		

Berdasarkan Tabel 6 dapat disimpulkan jika proses reset fase memiliki nilai CC yaitu 1 path.

3.2. Pengujian Aplikasi Android dengan Black Box

Pengujian Black Box dilakukan secara fungsionalitas pada semua fitur. Berikut merupakan hasil dari Black Box Testing yang dilakukan menggunakan 3 perangkat yaitu Samsung Galaxy M51, Realme 5i, dan Samsung A02S. Pengujian Black Box ditunjukkan pada Tabel 7.

Tabel 7 Indikator Pengujian semua fitur

No	Fitur yang Diuji	Hasil Pengujian		
		Peran gkat 1	Peran gkat 2	Peran gkat 3
1	Install Aplikasi	V	V	V
2	Menampilkan SplashScreen	V	V	V
3	Melakukan Login pada LoginScreen	V	V	V
4	Melakukan Register pada RegisterScreen	V	V	V
5	Memasukkan token media di HomeScreen	V	V	V
6	Mengubah status sistem kontrol di KontrolScreen	V	V	V
7	Mengulang kembali fase menjadi hari pertama di KontrolScreen	V	V	V
8	Menampilkan kondisi media pada KontrolScreen	V	V	V
9	Menampilkan kondisi detail suhu dan kelembapan pada MonitoringScreen	V	V	V
10	Menampilkan rata-rata suhu dan kelembapan setiap hari dan tahap fase dalam periode sekarang	V	V	V
11	Menampilkan status aktuator pada AktuatorScreen	V	V	V
12	Mengubah lokasi atau token media pada EditMediaScreen	V	V	V
13	Push Notification	V	V	V

Berdasarkan hasil pengujian pada Tabel 7 dapat dilihat bahwa semua fitur pada aplikasi dapat berjalan dengan lancar pada semua perangkat yang memiliki versi Android 9, 10 dan 12. Fitur yang dilakukan pengujian yaitu install aplikasi, menampilkan *SplashScreen*, *login* dan *register*, masuk ke media, *controlling* dan *monitoring* serta notifikasi.

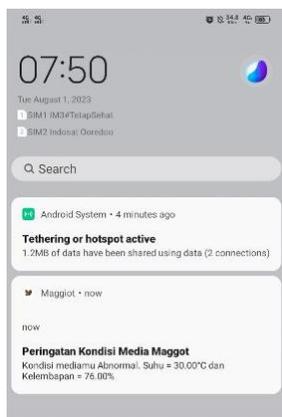
3.3. Pengujian Penerimaan Notifikasi pada Aplikasi Android

Pada pengujian penerimaan notifikasi dilakukan dengan mengamati Notifikasi dikirimkan jika suhu atau kelembapan sedang dalam kondisi abnormal. Pada senarai program di ESP8266 untuk mengetahui jika kondisi abnormal adalah dengan menggunakan variabel peringatansuhu dan peringatankelembapan. Jika peringatansuhu atau peringatankelembapan bernilai 1 maka variabel peringatan akan bernilai 1. Dalam pengujiannya akan dipantau dari ESP8266 melalui serial monitor seperti pada Gambar 18.



Gambar 18 Serial monitor pengiriman notifikasi

Dalam Gambar 18 terlihat jika notifikasi dikirimkan saat kelembapan dalam kondisi abnormal dibuktikan dengan hasil pembacaan variabel peringatankelembapan bernilai 1. Setelah itu variabel peringatan akan bernilai 1 juga. ESP8266 akan memanggil data token device dari masing masing perangkat dimana notifikasi akan dikirimkan. Setelah itu diperlihatkan pada serial monitor jika notifikasi dikirimkan dengan status “*success*” : 1 dan “*failure*” : 0 yang menandakan jika notifikasi sukses dikirim. Pada device android notifikasi akan muncul seperti pada Gambar 19.



Gambar 19 Tangkapan layar penerimaan notifikasi

Pada Gambar 19 diketahui jika notifikasi masuk pada menit yang sama dengan ESP8266 mengirim notifikasi

saat dipantau melalui serial monitor yaitu menit 07.50 seperti pada gambar 19. Notifikasi yang muncul memiliki *title* “Peringatan Kondisi Media Maggot” dan *body* yang bertuliskan “Kondisi mediamu Abnormal. Suhu = 30.00°C dan kelembapan = 76.00%” Pengujian pengiriman notifikasi dilakukan sebanyak 3 kali dengan hasil pada Tabel 8.

Tabel 8 Latensi pengiriman notifikasi

No	Waktu Pengiriman dari ESP8266	Waktu Pengiriman dari ESP8266
1	07.50	07.50
2	08.00	08.00
3	08.10	08.10

Berdasarkan Tabel 8, diketahui jika pengiriman notifikasi dari ESP8266 ke aplikasi android memiliki keterlambatan dibawah satu menit.

3.4. Pengujian User Acceptance Test (UAT)

Pengujian UAT dilakukan dengan melibatkan pengguna dari beberapa mahasiswa jurusan Teknik Elektro Universitas Diponegoro dan pegawai dari *Farm* Rispro menggunakan google *form* untuk mahasiswa Parameter yang digunakan untuk indikator terdiri dari 5 tingkat yaitu SS (Sangat Setuju), S (Setuju), KS (Kurang Setuju), TS (Tidak Setuju), dan STS (Sangat Tidak Setuju) yang ditunjukkan pada Tabel 9.

Tabel 9 Data Diri Responden

No	Nama Responden	Usia (Tahun)	Status
1	Muhammad Hartadi	33	Pegawai <i>Farm</i> Rispro
2	Indah Arga	40	Pegawai <i>Farm</i> Rispro
3	Hafid	43	Pegawai <i>Farm</i> Rispro
4	Ronald Aryawan P.	22	Pegawai <i>Farm</i> Rispro
5	Viorel Ersyad Elhafizh	22	Mahasiswa
6	Luthfia Barliana Sumardi	22	Mahasiswa
7	Tristan Rizky Setiawan	22	Mahasiswa
8	Denny Anggun R.	20	Mahasiswa
9	Aloisius Gonzaga Y. W. A.	22	Mahasiswa
10	Arinda Dwi Wahyu	22	Mahasiswa

Responden terdiri dari pegawai dari *Farm* Rispro dan juga mahasiswa Teknik Elektro Universitas Diponegoro yang

diberikan 10 pernyataan bernilai positif tentang aplikasi MAGGIOT yang ditampilkan pada Tabel 10.

Tabel 10 Hasil Data Responden

No	Pernyataan	SS	S	KS	TS	STS
1	Tampilan pada register sesuai dengan yang saya harapkan	7	3	-	-	-
2	Tampilan pada login sesuai dengan yang saya harapkan	8	2	-	-	-
3	Tampilan pada masuk ke media sesuai dengan yang saya harapkan	5	4	1	-	-
4	Tampilan pada halaman KontrolScreen sesuai dengan yang saya harapkan	9	1	-	-	-
5	Tampilan pada halaman MonitoringScreen sesuai dengan yang saya harapkan	7	3	-	-	-
6	Tampilan pada halaman DataRecordScreen sesuai dengan yang saya harapkan	7	3	-	-	-
7	Tampilan pada halaman AktuatorScreen sesuai dengan yang saya harapkan	6	3	1	-	-
8	Tampilan pada halaman EditMediaScreen sesuai dengan yang saya harapkan	7	2	1	-	-
9	Tampilan antarmuka pada aplikasi menarik	6	3	1	-	-
10	Aplikasi mudah untuk saya pahami dalam pemakaiannya	9	1	-	-	-
Total		71	25	4	0	0

Berdasarkan Tabel 10, maka dari 10 pernyataan bernilai positif diketahui jika parameter Tidak Setuju dan Sangat Tidak Setuju bernilai 0 sehingga dapat disimpulkan bahwa 71% responden sangat setuju jika aplikasi sesuai dengan pernyataan, 25% responden setuju jika aplikasi sesuai dengan pernyataan, dan 4% kurang setuju jika aplikasi sesuai dengan pernyataan.

4. Kesimpulan

1. Pada pengujian Aplikasi MAGGIOT menggunakan metode *White Box Testing* berjalan dengan menunjukkan bahwa setiap senarai program pada aplikasi android menggunakan *framework* React Native memiliki nilai *Cyclomatic Complexity* (CC) 5,5,4,1,1 dan 2 atau dengan rata-rata 3.
2. Pada pengujian Aplikasi MAGGIOT menggunakan metode *Black Box Testing* berjalan dengan menunjukkan bahwa setiap fitur berhasil dijalankan.
3. Pada pengujian notifikasi Aplikasi MAGGIOT jika media dalam kondisi abnormal didapatkan hasil bahwa pengiriman notifikasi berhasil dengan latensi pengiriman notifikasi tidak lebih dari 1 menit.
4. Pada pengujian Aplikasi MAGGIOT menggunakan metode UAT yang diisi oleh pegawai *Farm Rispro* dan mahasiswa dapat disimpulkan bahwa 71% responden sangat setuju, 25% responden setuju, dan 4% responden kurang setuju.
5. Antarmuka aplikasi *mobile* MAGGIOT berhasil untuk *controlling* dan *monitoring* media maggot dengan pengujian *White Box Testing*, *Black Box Testing*, dan *User Acceptance Test*.

Referensi

- [1] N. Azizah, L. Budiman, E. Anggriani, and F. Adiba, "Upaya mendukung global goals melalui pelatihan budidaya maggot sebagai alternative pakan ikan pada kelompok wanita econatural di kota Makassar," vol. 3, no. 1, pp. 211–219, 2022.
- [2] D. Schulz, "FDI and the Industrial Internet of Things," 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg, Luxembourg, 2015, pp. 1-8, doi: 10.1109/ETFA.2015.7301513.
- [3] I. U. Turyadi, "Analisa Dukungan Internet of Things (IoT) terhadap Peran Intelejen dalam Pengamanan Daerah Maritim Indonesia Wilayah Timur," J. Teknol. dan Manaj. Inform., vol. 7, no. 1, pp. 29–39, 2021, doi: 10.26905/jtmi.v7i1.6040.
- [4] I. Rizki Destama Putra, "(Monitoring dan Kontrol Suhu Lampu untuk Budidaya Maggot BSF Berbasis IoT" vol. 9, no. 12, pp. 37–44.
- [5] M. S. Chumairoh, "Perancang Bangun Aplikasi Mobile Pada Platform Android Berbasis Html5 Studi Kasus Layanan Informasi Website Unipdu Jombang," Edutic - Sci. J. Informatics Educ., vol. 1, no. 1, pp. 1–6, 2015, doi: 10.21107/edutic.v1i1.402.