

Analisis Kinerja Aplikasi Pemantauan dan Pengendalian *Smart Agriculture* Berbasis Android

Helmy¹, Fenny Rahmasari², Arif Nursyahid³, Thomas Agung Setyawan⁴, Ari Sriyanto Nugroho⁵

Intisari—Era digital yang terus berkembang melahirkan revolusi industri budidaya pertanian dan peternakan cerdas berbasis *internet of things* (IoT). Salah satunya adalah penggunaan aplikasi berbasis Android dalam pemantauan dan pengendalian parameter pada proses budidaya di era digital ini. Koneksi internet yang tidak stabil dapat mengganggu proses pemantauan. Oleh karena itu, diperlukan pengintegrasian sistem dalam satu aplikasi yang dapat berjalan walaupun dalam keadaan internet terputus (*offline*), sehingga pengelola dapat melakukan pemantauan dan pengendalian *smart agriculture* berbasis Android dengan dua mode operasi, yaitu *online* dan *offline*. Analisis kinerja juga diperlukan untuk mengetahui keandalan aplikasi dalam pengiriman dan penerimaan data. Integrasi sistem ini menggunakan dua mode, yaitu mode operasi *online* dan *offline*. Mode *online* mengindikasikan aplikasi Android berinteraksi dengan sistem melalui server ketika terkoneksi dengan internet menggunakan metode *representational state transfer application programming interface* (REST API), sedangkan mode *offline* artinya aplikasi Android berinteraksi langsung dengan sistem melalui koneksi *local device* (*access point/AP* lokal). *Smart agriculture* berinteraksi dengan sistem melalui protokol *message queuing telemetry transport* (MQTT), dengan Android bertindak sebagai *client* MQTT. Analisis kinerja aplikasi meliputi pengujian *black box*, *load activity*, dan kinerja aplikasi melalui fitur *profiler* Android. Hasil yang diperoleh berdasarkan pengujian fungsionalitas aplikasi (*black box*) menunjukkan bahwa aplikasi dapat memantau dan mengendalikan *smart agriculture* dengan baik secara *online* maupun *offline* oleh pengelola. Nilai *load time* rata-rata di seluruh aktivitas sebesar 3,507 detik saat *bandwidth* sebesar 4,54 Mbps. Sementara itu, rata-rata *load time* yang dihasilkan saat *bandwidth* 35,35 Mbps adalah sebesar 1,4 detik. Berdasarkan pengujian kinerja, sistem ini cukup ringan dengan penggunaan *central processing unit* (CPU) terbesar 31% dan penggunaan memori terbesar 453,8 MB.

Kata Kunci—Aplikasi Android, Analisis Kinerja, IoT, *Smart Agriculture*.

I. PENDAHULUAN

Sektor pertanian termasuk sektor yang paling menjanjikan karena semakin banyaknya pertumbuhan penduduk yang cukup signifikan dan dibarengi oleh sektor pariwisata, infrastruktur, bisnis *e-commerce*, dan sektor perdagangan/ retail [1].

^{1,3,4,5} Politeknik Negeri Semarang, Jl. Prof. H. Soedarto, S.H., Tembalang, Semarang 50275 (telp: 024 7473417, 7499585; fax: 024 7472396; e-mail: ¹helmy@polines.ac.id, ³arif.nursyahid@polines.ac.id, ⁴thomas@polines.ac.id, ⁵ari.sriyanto@polines.ac.id)

² Jurusan Teknik Elektro Politeknik Negeri Semarang, Jl. Prof. H. Soedarto, S.H., Tembalang, Semarang 50275 (telp: 024 7473417, 7499585; fax: 024 7472396; e-mail: ²fennyrahma@gmail.com)

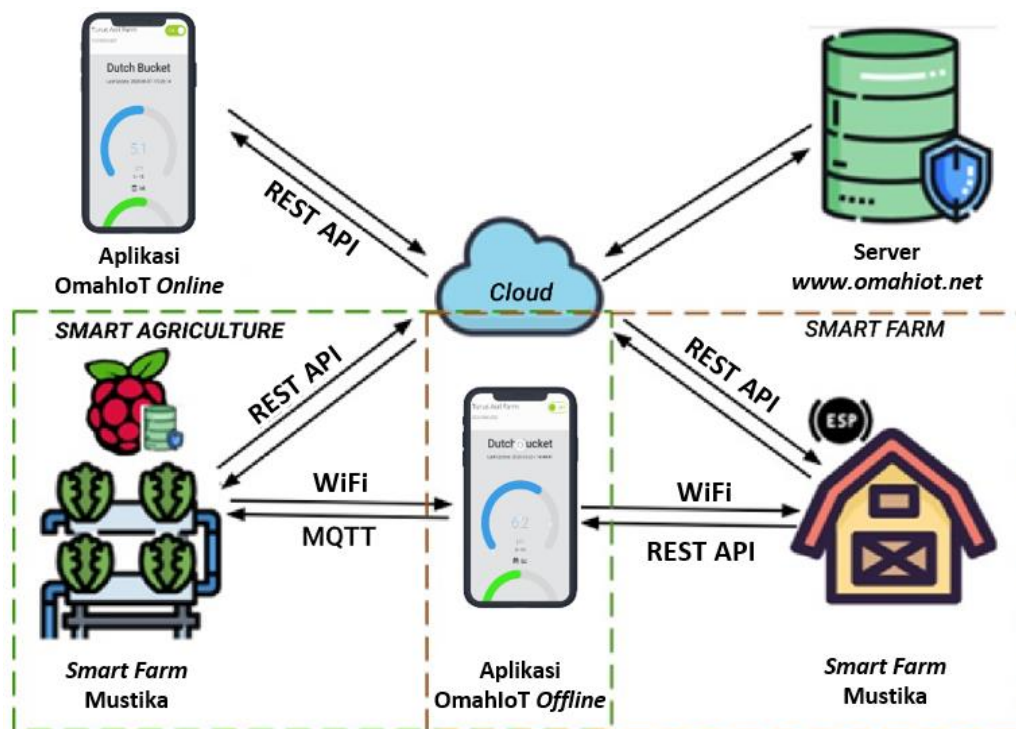
[Diterima: 23 Desember 2021, Revisi: 15 Januari 2022]

Berdasarkan laporan bulanan data sosial ekonomi dari Badan Pusat Statistik Indonesia bulan Desember 2019 lalu, empat nilai tukar petani (NTP) tercatat 104,46 atau naik sebesar 0,35 % dibanding NTP November 2019, yaitu sebesar 104,10 [2].

Berkembangnya era digital telah melahirkan pertanian 4.0 dan revolusi dalam industri pertanian pintar pun berkembang. Hal ini membantu memandu tindakan yang diperlukan untuk memodifikasi dan mengarahkan kembali sistem pertanian agar secara efektif mendukung pengembangan dan menjamin ketahanan pangan yang diimplementasikan dengan *internet of things* (IoT) [3]. Pengembangan IoT di bidang pertanian dapat memudahkan petani dalam mengelola lahan pertaniannya tanpa perlu mengoperasikan suatu perangkat secara manual serta dapat mengefisienkan waktu yang biasanya digunakan untuk melakukan tugas yang sama [4]. Pemantauan dan pengendalian dapat dilakukan dari jarak jauh melalui *website* maupun aplikasi berbasis Android [5]. Penggunaan aplikasi berbasis Android pada penelitian ini dilakukan karena banyaknya pengguna *smartphone*, khususnya Android [6], [7].

Beberapa penelitian mengenai sistem pemantauan dan pengendalian *smart agriculture* berbasis Android telah dilakukan. Salah satunya adalah penelitian mengenai kinerja aplikasi Android untuk pemantauan dan pengendalian *urban farming* pada jaringan 4G [8]. Kinerja aplikasi Android sebagai pemantau dan pengendali sistem *urban farming* ini diuji pada waktu responsnya melalui jaringan seluler 4G dengan hasil kisaran waktu respons antara 4-9 detik [9]. Uji aplikasi pada jaringan 4G dimaksudkan untuk mengukur kecepatan aplikasi dalam merespons kendali terhadap proses *urban farming* dari beberapa lokasi. Penelitian lain dilakukan untuk mendeteksi pertumbuhan tanaman yang terhubung dengan perangkat lunak bernama AGRO-TECH untuk mengendalikan beberapa sensor. Sistem notifikasi atau pemberitahuan yang dipakai adalah *short message service* (SMS) yang berisi laporan penghasilan mingguan dan juga *pop-up* pada aplikasinya [10]. Selanjutnya, sebuah penelitian membuat aplikasi *mobile* untuk memantau parameter hidroponik. Penelitian tersebut bertujuan untuk mengendalikan perubahan cuaca dan perubahan tingkat nutrisi karena transpirasi dan penyerapan oleh sistem tumbuhan menggunakan teknologi IoT. Sistem ini menggunakan sensor *light dependent resistor* (LDR), sensor suhu dan kelembapan untuk pemantauan cuaca, sensor pH, sensor konduktivitas listrik (*electrical conductivity/EC*) untuk kontrol kualitas air, dan sensor ultrasonik untuk mengukur ketinggian air. Aplikasi yang digunakan pada penelitian ini adalah MIT App Inventor, yang merupakan *platform online* untuk mengembangkan aplikasi *mobile* [11].

Berdasarkan penelitian-penelitian yang sudah dilakukan sebelumnya, diketahui belum ada penelitian yang membuat aplikasi sistem pemantauan dan pengendalian *smart*



Gbr. 1 Diagram integrasi sistem.

agriculture berbasis Android dengan mode *online* (dengan koneksi internet) dan mode *offline* (tanpa koneksi internet) beserta analisis kinerjanya. Sistem pemantauan dan pengendalian dengan mode *online* ini dapat memudahkan pengguna melakukan pemantauan dan pengendalian dari jarak jauh, sedangkan metode *offline* tetap digunakan agar penyelesaian masalah dapat lebih cepat dilakukan apabila terjadi kendala koneksi internet atau kendala lainnya. Makalah ini tersusun dari empat bagian. Bagian I adalah pendahuluan, sedangkan Bagian II membahas desain. Hasil dan pembahasan dijelaskan pada Bagian III dan yang terakhir, Bagian IV, berisi kesimpulan.

II. METODOLOGI

Metode yang digunakan pada penelitian ini adalah metode *waterfall*. Metode *waterfall* adalah metode klasik dengan sistem yang linier, yaitu keluaran pada tahap sebelumnya menjadi masukan bagi tahap selanjutnya. Metode perancangan dan pengujian sistem pada penelitian ini dimulai dengan spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan perangkat lunak ke pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak yang dihasilkan.

Pada tahap perencanaan, dilakukan pendataan kebutuhan alat dan bahan, baik berupa perangkat lunak maupun perangkat keras. Kemudian, tahap selanjutnya adalah pembuatan rancangan kerja yang berisi jadwal kegiatan. Tahap pemodelan pada penelitian ini berisikan seluruh perancangan desain sistem, meliputi diagram integrasi sistem, yang ditunjukkan pada Gbr. 1, perancangan model perangkat lunak untuk aplikasi OmahIoT, diagram *use case* aplikasi OmahIoT, *use case*

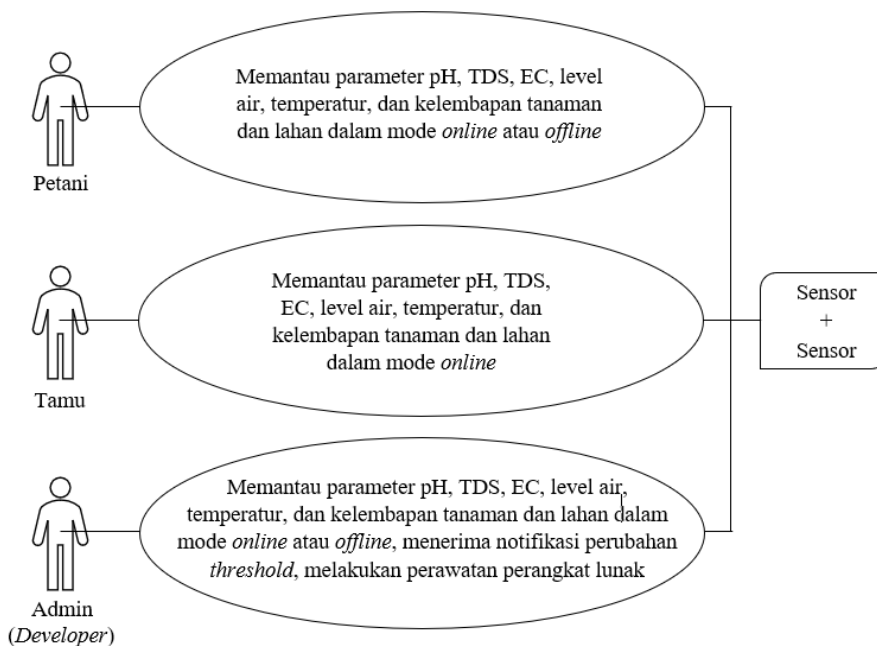
subsystem aplikasi OmahIoT, perancangan basis data berupa *entity relationship diagram* (ERD), dan desain *user interface* (UI) aplikasi Android OmahIoT. Kemudian, tahap konstruksi merupakan tahap pengerjaan aplikasi Android OmahIoT dengan perangkat lunak Android Studio IDE menggunakan tiga tahapan, yaitu tahap eksekusi, tahap pengujian, dan tahap *debugging*.

Tahap pengujian memuat pengambilan data yang dilakukan bersamaan dengan pengujian keseluruhan sistem. Dalam pengujian pada sisi aplikasi yang dilakukan dengan metode *black box*, data yang diambil berupa fungsi-fungsi yang tidak benar atau hilang, kesalahan antarmuka (*interface*), kesalahan dalam struktur data atau akses basis data eksternal, kesalahan kinerja, inisialisasi, dan kesalahan terminasi.

III. DESAIN

Integrasi dua sistem pemantauan dan pengendalian *smart agriculture* dan *smart farm* berbasis Android ditunjukkan pada Gbr. 1. Sistem berbasis Android pada makalah ini mengukung dua mode, yaitu mode *online* dan mode *offline*. Mode *online* mengindikasikan aplikasi Android akan berinteraksi dengan sistem melalui server ketika terkoneksi dengan internet menggunakan metode *representational state transfer application programming interface* (REST API). Server yang digunakan bernama *www.omahiot.net*. Sementara itu, mode *offline* artinya aplikasi Android akan berinteraksi langsung dengan sistem melalui koneksi *local device* (*access point/AP* lokal). *Smart agriculture* berinteraksi dengan sistem melalui protokol *message queuing telemetry transport* (MQTT), dengan Android bertindak sebagai *client* MQTT.

Pada aplikasi ini terdapat tiga aktor, yaitu petani, tamu, dan admin/*developer*. Dalam melakukan pemantauan dan



Gbr. 2 Diagram use case smart agriculture.

TABEL I
USE CASE SUBSISTEM APLIKASI

Subsistem	Pengguna /Aktor	Deskripsi
Management user account	Petani dan Tamu	Pengguna dapat mendaftar sendiri dan juga dapat memperbarui atau melihat detailnya. Pengguna dapat melihat/mengedit detail pengguna dan hapus pengguna. Untuk pengguna petani harus mempunyai SN untuk fitur aplikasi khusus petani.
Management monitoring farmer	Petani	Petani dapat memantau lahan baik secara online maupun offline dan juga tampilan monitoring akan terlihat sesuai username petani.
Management guest	Tamu	Tamu dapat melihat info terkini seputar lahan.
Management controlling farmer	Petani	Petani dapat mengatur threshold yang dapat ter-update langsung ke basis data server/lokal.

pengendalian, petani diberi dua mode, yaitu *online* dan *offline*. Kedua mode dapat berjalan sesuai koneksi internet pengelola di lahan pertanian. *Guest*/pengguna tamu hanya diberi kesempatan memantau dalam mode *online* saja. Data *monitoring* disediakan oleh server web yang kemudian ditampilkan pada Android pengguna. Admin/*developer* bertugas dalam perawatan perangkat lunak dan pengembangan aplikasi.

Diagram *use case* pada *smart agriculture* diperlihatkan pada Gbr. 2. Aktor dapat memantau parameter tanaman, mengatur *threshold* parameter tanaman, dan menerima notifikasi apabila terjadi perubahan *threshold*.

Desain aplikasi lanjutan dilakukan menggunakan metode pemecahan menjadi subsistem. Tabel I menjelaskan *use case* subsistem aplikasi OmahIoT. Subsistem terdiri atas *management user account*, *management monitoring farmer*, *management guest*, dan *management controlling farmer*.

A. Perancangan Pengujian Fungsionalitas Aplikasi (Black Box Testing)

Tahap ini memuat pengambilan data yang dilakukan bersamaan dengan pengujian keseluruhan sistem. Dalam pengujian dari sisi aplikasi, yang dilakukan dengan metode *black box*, data yang diambil berupa fungsi-fungsi yang tidak benar atau hilang, kesalahan antarmuka, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan kinerja, serta inisialisasi dan kesalahan terminasi.

B. Perancangan Load Testing Aktivitas Aplikasi

Load testing aplikasi merupakan pengujian waktu yang diperlukan data untuk berpindah dari aktivitas satu ke aktivitas lainnya. Selain itu, dilakukan juga analisis *load time* aktivitas *monitoring* dan *load time* aktivitas *controlling*. Tujuan utamanya adalah untuk menguji kemampuan aplikasi menangani permintaan dari sejumlah pengguna secara objektif, seperti mempertahankan waktu respons terhadap permintaan pengguna. Beberapa aplikasi dapat mengalami permasalahan, seperti *long load time*, yaitu kondisi *load time* aplikasi lebih dari satu menit. *Load time* harus di bawah 10 detik jika memungkinkan. Ketentuan *load time* yang dihasilkan berdasarkan pengambilan saat *bandwidth* buruk adalah tidak lebih dari 5 detik, sedangkan untuk *bandwidth* yang bagus, *load time* yang dihasilkan adalah tidak lebih dari 3 detik.

C. Perancangan Pengujian Kinerja Aplikasi

Pengujian kinerja aplikasi dilakukan secara menyeluruh dengan memanfaatkan fitur Profiler pada Android Studio IDE.

TABEL II
PENGUJIAN FUNGSIONALITAS APLIKASI

No.	Aktivitas Pengujian	Tampilan yang Diharapkan	Tampilan (Status)
1	<i>Intro: Splash Welcome</i>	<i>Splash</i> : menampilkan 3-4 detik logo dan slogan OmahIoT. <i>Welcome</i> : menampilkan info seputar OmahIoT berbentuk <i>carousel</i> .	<i>Splash</i> (sesuai) <i>Welcome</i> (sesuai)
2	<i>User authentication: Login Register Profile</i>	<i>Login</i> : pengguna dapat melakukan <i>login</i> dengan menggunakan <i>email</i> dan <i>password</i> yang telah terdaftar. <i>Register</i> : pengguna dapat mendaftarkan akun dengan mengisi nama, <i>email</i> , <i>password</i> , dan juga kontak. <i>Profile</i> : pengguna dapat mengubah data dan nama, <i>email</i> , alamat, kontak, dan foto profil.	<i>Login</i> (sesuai) <i>Register</i> (sesuai) <i>Profile</i> (sesuai)
3	<i>Menu dashboard: Smart Agriculture</i>	Menampilkan tampilan menu awal berisi <i>Monitoring</i> , <i>Controlling</i> , <i>Profile</i> , dan <i>Apps Guide</i> . Konten dalam setiap menu akan berbeda di setiap jenisnya yang telah terdaftar sebelumnya di web OmahIoT penggunanya.	<i>Dashoard Smart Agriculture</i> (sesuai)
4	<i>Menu monitoring: Monitoring Offline Monitoring Online</i>	<i>Mode offline</i> : dengan cara mengoneksikan <i>smartphone</i> pada WiFi lokal <i>device</i> sistem terdekat. Menampilkan nilai sensor sebenarnya dari sistem secara <i>real time</i> melalui <i>edge</i> sesuai dengan <i>username</i> . <i>Mode online</i> : dengan cara mengoneksikan <i>smartphone</i> pada WiFi reguler. Menampilkan pembacaan sensor dengan bentuk <i>gauge chart</i> melalui metode <i>WebView</i> .	<i>Monitoring Offline</i> (sesuai) <i>Monitoring Online</i> (sesuai)
5	<i>Menu controlling: Controlling Offline Controlling Online</i>	<i>Setting</i> beberapa <i>threshold</i> parameter perangkat/sensor melalui aplikasi Android. Jalur komunikasi untuk <i>controlling</i> adalah <i>mode offline</i> dengan koneksi WiFi lokal + protokol MQTT perangkat sekitar. <i>Mode online</i> dilakukan dengan mengirimkan data dalam <i>mode offline</i> terakhir ke basis data pusat.	<i>Controlling Smart Agriculture</i> (sesuai)
6	<i>Notifikasi: Smart Agriculture</i>	Pada <i>smart agriculture</i> notifikasi <i>pop up</i> akan muncul ketika sedang melakukan <i>controlling</i> . Notifikasi berupa pemberitahuan bahwa data <i>threshold</i> telah masuk ke perangkat.	<i>Notifikasi Smart Agriculture</i> (sesuai)
7	<i>Apps Guides</i>	Menampilkan sejumlah informasi dan petunjuk dari fitur yang sesuai dengan <i>user role</i> .	<i>Apps Guides</i> (sesuai)



Gbr. 3 Tampilan *monitoring online* gagal.

Semakin responsif suatu aplikasi *mobile*, semakin bagus aplikasi tersebut. Semakin cerdasnya aplikasi mengenali segala bentuk tindakan yang diinginkan pengguna akan menambah nilai kualitas aplikasi tersebut.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Fungsionalitas Aplikasi (Black Box Testing)

Pengujian dalam segi aplikasi (*black box testing*) menguji sukses atau tidaknya aplikasi berjalan sesuai skenario.



Gbr. 4 Tampilan *monitoring offline* gagal.

Pengujian ini hanya berfokus pada fungsionalitas dan keluaran. Pengujian lebih ditujukan pada desain aplikasi sesuai dengan pemodelan dan reaksi-reaksi apabila terdapat celah-celah *bug*/

TABEL III
PENGUJIAN LOAD TESTING APLIKASI

No.	Bandwidth (Mbps)	Objek	Waktu (detik)
1	4,54 (buruk)	Splash screen	3,000
		Welcome activity	5,053
		Login activity	3,553
		Dashboard Turus Asri	8,123
		Dashboard Mustika	4,159
		Monitoring online Turus Asri	6,306
		Monitoring offline Turus Asri	1,453
		Monitoring offline Mustika	7,458
		Controlling Turus Asri	0,300
		Controlling Mustika	0,200
		Profil Turus Asri	1,637
		Profil Mustika	3,747
		Logout Turus	0,237
Logout Mustika	1,790		
Rata-rata load time per aktivitas			3,507
2	35,15 (baik)	Splash screen	3,000
		Welcome activity	2,847
		Login activity	1,127
		Dashboard Turus Asri	1,420
		Dashboard Mustika	0,358
		Monitoring online Turus Asri	1,702
		Monitoring online Mustika	3,620
		Monitoring offline Turus Asri	1,410
		Monitoring offline Mustika	4,320
		Controlling Turus Asri	0,020
		Controlling Mustika	0,030
		Profil Turus Asri	0,388
		Profil Mustika	0,504
Logout Turus	0,060		
Logout Mustika	0,205		
Rata-rata load time per aktivitas			1,400

vulnerability pada program aplikasi. Alat yang digunakan untuk melakukan pengujian fungsionalitas ini adalah *smartphone* Android. Begitu pula dengan simulatornya yang menggunakan *real device* berupa *smartphone* Android. API yang dipakai dalam pembuatan aplikasi Android OmahIoT ini adalah minimum SDK 29, sedangkan versi Android minimal Jelly Bean/6. Dalam pengujiannya, digunakan SDK API 30 dan Android versi 10. Tabel II menyajikan hasil pengujian fungsionalitas aplikasi.

Aplikasi Android ini dibuat untuk memantau *smart agriculture* dengan koneksi internet (*online*) maupun tanpa koneksi internet (*offline*). Mode operasi *online* memiliki tugas membaca data sensor berupa pH, *total dissolved solids* (TDS), cahaya, suhu *greenhouse*, kelembapan, volume air tandon, dan suhu nutrisi serta membaca data sensor berupa suhu kandang ayam, kadar amonia, kelembapan, kipas, pendingin (*cooler*), dan penghangat (*heater*). Data sensor tersebut dikirim dari perangkat basis data pusat terlebih dahulu, kemudian Android menerimanya lewat *cloud* dengan WebView. Mode operasi *online* sangat bergantung pada koneksi internet. Indikasi kegagalan *monitoring* mode *online* akibat koneksi internet terputus ditunjukkan pada Gbr. 3.



Gbr. 5 Tampilan monitoring, (a) offline, (b) online.

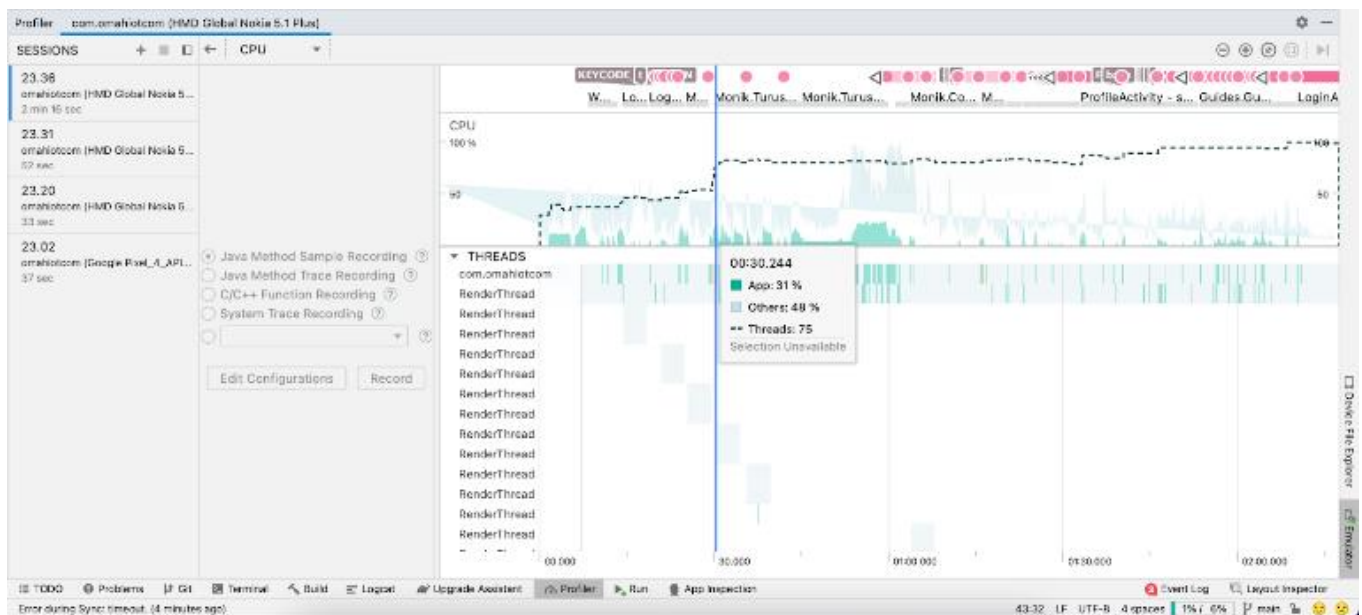
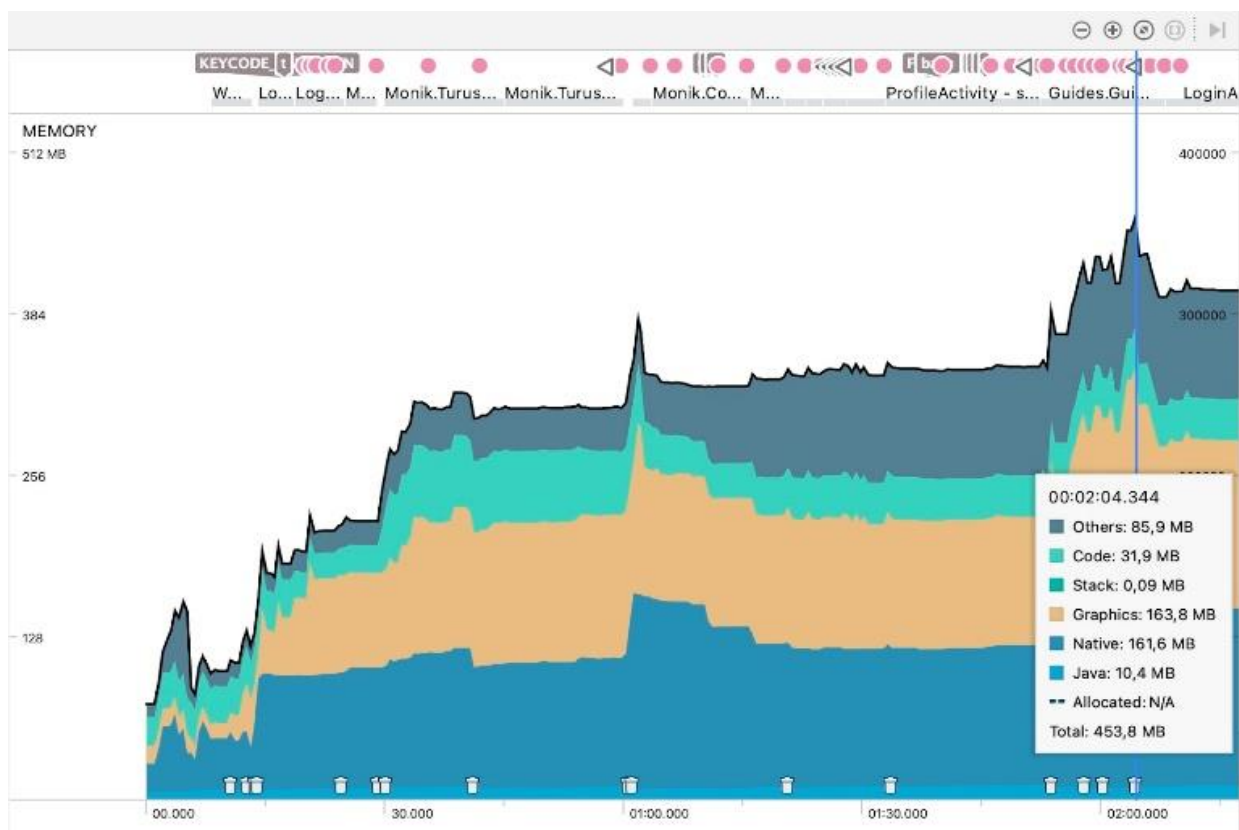


Gbr. 6 Controlling smart agriculture.

Monitoring dalam mode operasi *offline* sangat bergantung pada WiFi lokal perangkat yang dijadikan AP. Apabila koneksi pada AP perangkat terputus, *monitoring offline* akan gagal menerima data. Indikasi *monitoring offline* gagal akibat koneksi AP terputus ditunjukkan pada Gbr. 4.

Tampilan *monitoring offline* dan *online* ditunjukkan pada Gbr. 5. Menu *controlling* dalam aplikasi Android ini mengusung mode *offline* atau melalui koneksi AP perangkat. *Controlling* pada *smart agriculture* ditunjukkan oleh Gbr. 6.

Controlling untuk *smart agriculture* memanfaatkan protokol MQTT, sedangkan aplikasi Android bertindak sebagai *client* dan koneksi AP Raspberry Pi. Pengelola hanya perlu memasukkan data nilai *threshold* berupa nilai TDS dan pH yang dikehendaki sesuai format penulisan yang ada pada *hint form*. Indikasi *controlling* berhasil adalah pengguna mendapat

Gbr. 7 Hasil *profiling* CPU aplikasi OmahIoT.Gbr. 8 Hasil *profiling* memori aplikasi OmahIoT.

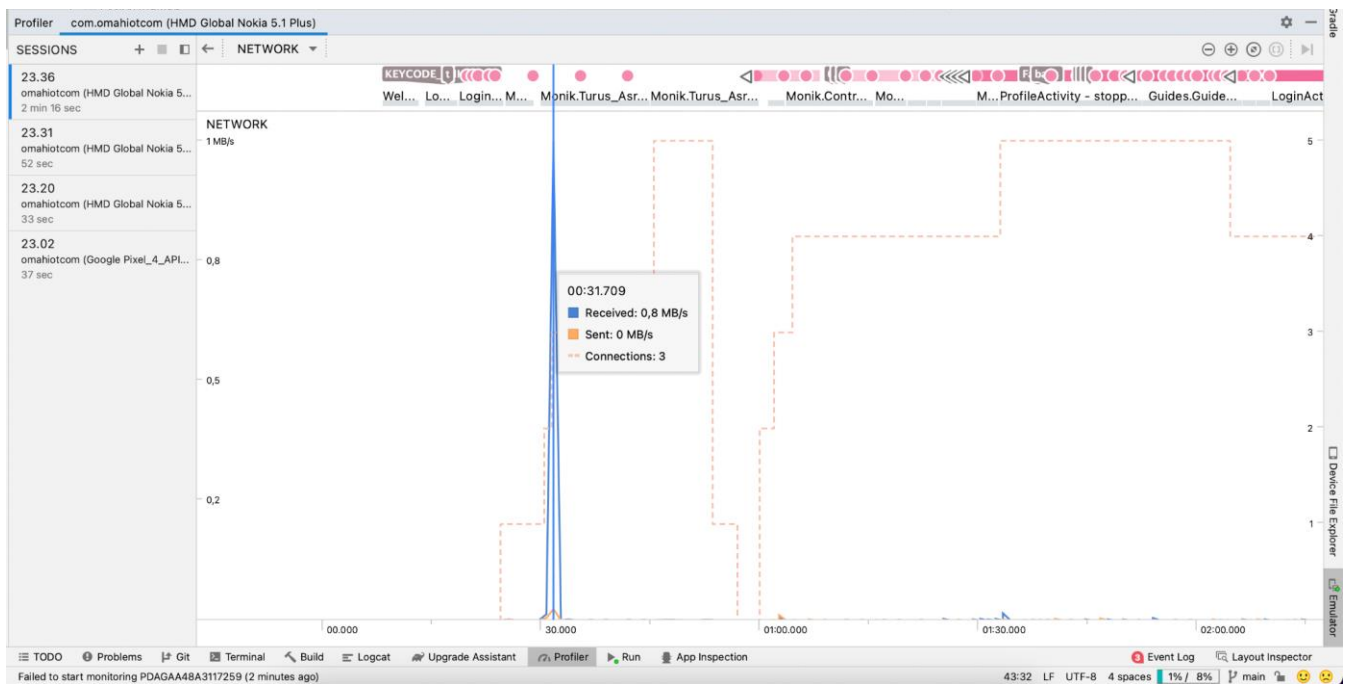
pesan “connected” ketika masuk ke aktivitas *controlling* dan menerima pesan data telah ter-*publish* setelah memasukkan nilai *threshold* sesuai format yang benar kemudian menekan tombol *publish*.

B. Hasil Load Testing Aktivitas Aplikasi

Load testing aplikasi adalah uji kinerja (*performance testing*), yaitu respons sistem diukur dalam berbagai kondisi

kerja (*load condition*). Target nilai *load* aplikasi adalah rata-rata *load time* per aktivitas kurang dari 5 detik untuk kondisi *bandwith* yang buruk dan kurang dari 3 detik untuk kondisi *bandwith* yang bagus. Tabel III menyajikan hasil pengujian fungsionalitas aplikasi.

Tabel III menunjukkan bahwa pada kondisi *bandwith* yang buruk, yaitu 4,54 Mbps, rata-rata *load* aplikasi adalah 3,507



Gbr. 9 Hasil *profiling network* aplikasi OmahIoT.

detik, sedangkan pada kondisi *bandwith* yang cukup bagus, rata-rata *load time* sebesar 1,4 detik, yang artinya sesuai target nilai rata-rata yang ditentukan. Aktivitas dengan *load time* yang buruk ada pada aktivitas *login* ke *dashboard* menu Turus Asri, yaitu sebesar 8,123 detik, sedangkan *load time* terbaik ada pada aktivitas *controlling* Mustika, yang hanya membutuhkan waktu 0,2 detik dalam kondisi *bandwith* buruk. Untuk kondisi *bandwith* yang bagus, yaitu sebesar 35,15 Mbps, aktivitas dengan *load time* terburuk ada pada aktivitas *monitoring offline* Mustika, yaitu sebesar 4,32 detik, sedangkan *load time* terbaik terjadi pada aktivitas *controlling* Turus Asri, yaitu 0,02 detik saja. Perbedaan *load time* ini disebabkan oleh kondisi *bandwith* yang tidak stabil.

C. Perancangan Pengujian Kinerja Aplikasi

Fitur Profiler pada Android Studio menyediakan data *real time* untuk membantu memahami cara aplikasi menggunakan *resource central processing unit* (CPU), memori, dan jaringan. Gbr. 7 menunjukkan hasil *profiling* CPU aplikasi OmahIoT. Optimalisasi penggunaan CPU aplikasi memiliki banyak keuntungan, misalnya memberikan pengalaman penggunaan yang lebih cepat dan lebih lancar serta menghemat masa pakai baterai perangkat. Gbr. 7 juga menunjukkan bahwa tidak ada penggunaan CPU yang mencapai 50% ketika aplikasi OmahIoT berjalan. Penggunaan CPU yang paling tinggi adalah saat aplikasi menampilkan aktivitas *monitoring*, yaitu 31% saja. Hal ini terjadi karena dalam melakukan *load web* di aktivitas *monitoring*, pengguna secara tidak langsung berinteraksi dengan server basis data sehingga penggunaan CPU akan meningkat.

Gbr. 8 menunjukkan hasil *profiling* memori aplikasi OmahIoT. Pengujian ini dapat membantu mengidentifikasi kebocoran memori dan *churn* memori (menunjukkan jumlah pengalokasian objek sementara yang terjadi dalam rentang

waktu tertentu) yang dapat menyebabkan aplikasi tersebut tersendat, berhenti merespons, dan bahkan *foreclose* (berhenti bekerja). Komponen ini menampilkan grafik *real time* atas penggunaan memori aplikasi yang memungkinkan perekaman *heap dump*, memaksa pembersihan sampah memori, dan melacak alokasi memori. Seperti yang terlihat pada Gbr. 8, memori yang dihabiskan untuk aplikasi OmahIoT paling besar adalah 453,8 MB, yaitu pada saat aktivitas *monitoring online* Turus Asri berpindah ke *dashboard* menu *Activity*. Dari rincian memori, memori dari objek yang dialokasikan dari kode Java adalah sebesar 10,4 MB; memori yang digunakan oleh *stack* Java dan *native* dalam aplikasi berdasarkan jumlah *thread* yang digunakan sebesar 0,09 MB; memori yang digunakan oleh aplikasi untuk kode dan *resource*, misalnya *bytecode dex*, kode *dex* yang telah dikompilasi atau dioptimalkan, *library .so*, dan *font*, sebesar 25 MB; dan memori yang digunakan oleh aplikasi tetapi sistem tidak yakin cara mengategorikannya sebesar 85,9 MB.

Gbr. 9 menunjukkan hasil *profiling* jaringan aplikasi OmahIoT. *Network profiler* menampilkan aktivitas jaringan secara *real time* pada lini masa, yang menunjukkan data terkirim dan diterima serta jumlah koneksi saat ini. Hasil *profiling* jaringan menunjukkan bahwa pada aplikasi OmahIoT terdapat tiga *file* yang telah dikirim atau diterima selama aktivitas yang dipilih dalam daftar di seluruh *thread* CPU aplikasi. Tiga *file* tersebut berjalan berdasarkan *thread* yang digunakan, misalnya *file* 192.168.4.1 berjalan dengan menggunakan *thread async task* (*method* GET) dan membutuhkan waktu sebesar 31,07 detik untuk berkomunikasi (pengiriman dan penerimaan). Hasil ini merupakan hasil yang terburuk di antara yang lainnya. Hal ini terjadi karena koneksi yang dibutuhkan adalah koneksi lokal, sedangkan dalam melakukan pengujian digunakan koneksi internet.

Berdasarkan data *profiling* jaringan tersebut, jaringan akan bekerja hanya pada aktivitas yang membutuhkan *file* sebagai data transmisi saja dan waktu yang dibutuhkan untuk melakukan pengiriman dan penerimaan respons bergantung pada *thread* yang digunakan, misalkan *thread asyncTask*, dan koneksi jaringan yang dipakai.

Berdasarkan ketiga data *profiling* tersebut, dapat diartikan bahwa aplikasi berjalan sesuai dengan pemodelan. Selain itu, dapat dikatakan juga bahwa aplikasi cukup ringan digunakan karena bergantung pada koneksi internet yang digunakan saat melakukan pengiriman dan penerimaan data.

V. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, aplikasi berbasis Android dapat berjalan sesuai rancangan integrasi yang dapat membantu petani dan peternak untuk memantau dan mengendalikan lahan dalam satu aplikasi saja. Hasil pengujian fungsionalitas (*black box*) menunjukkan bahwa aplikasi berbasis Android dapat berjalan sesuai dengan pemodelan. Aplikasi ini memiliki nilai *load time* rata-rata di seluruh aktivitasnya sebesar 3,507 detik saat *bandwidth* sebesar 4,54 Mbps, sedangkan rata-rata *load time* yang dihasilkan saat kondisi *bandwidth* 35,15 Mbps adalah 1,4 detik. Nilai tersebut telah memenuhi target hasil *load testing* yang diharapkan. Dalam segi kinerja, aplikasi ini cukup ringan, dengan penggunaan CPU paling besar 28%, penggunaan memori terbesar sebesar 236,7 MB, dan kinerja jaringan antara aktivitas mengirim dan menerima *file* data paling lama terjadi saat menggunakan jaringan lokal (*offline*).

KONFLIK KEPENTINGAN

Semua penulis menyatakan bahwa tidak terdapat konflik kepentingan dalam penelitian ini.

KONTRIBUSI PENULIS

Konseptualisasi, Helmy dan Arif Nursyahid; metodologi, Thomas Agung Setyawan; desain sistem, Helmy, Thomas

Agung Setyawan, dan Ari Sriyanto Nugroho; perangkat lunak, Fenny Rahmasari; analisis, Fenny Rahmasari dan Ari Sriyanto Nugroho; penulisan—penyusunan draf asli, Helmy; penulisan—peninjauan dan penyuntingan, Helmy, Fenny Rahmasari, dan Ari Sriyanto Nugroho.

REFERENSI

- [1] R. Khairiyakh, I. Irham, dan J.H. Mulyo, "Contribution of Agricultural Sector and Sub Sectors on Indonesian Economy," *Ilmu Pertan. (Agricultural Sci.)*, Vol. 18, No. 3, hal. 150–159, 2015.
- [2] "Laporan Bulanan Data Sosial Ekonomi," Badan Pusat Statistik, 2019.
- [3] K.A. Patil dan N.R. Kale, "A Model for Smart Agriculture Using IoT," *2016 Int. Conf. Global Trends Signal Process., Inf. Comput., Commun. (ICGTSPICC)*, 2016, hal. 543–545.
- [4] A.P. Atmaja, A. El Hakim, A.P.A. Wibowo, dan L.A. Pratama, "Communication Systems of Smart Agriculture Based on Wireless Sensor Networks in IoT," *J. Robot. Control*, Vol. 2, No. 4, hal. 297–301, Jul. 2021.
- [5] E. Hesti dan Adewasti, "Aplikasi Android sebagai Pengontrol Jarak Jauh Smart home dengan Koneksi Jaringan Internet," *J. Surya Energy*, Vol. 2, No. 2, hal. 157–165, Mar. 2018.
- [6] S.A. Arnomo dan H. Hendra, "Perbandingan Fitur Smartphone, Pemanfaatan dan Tingkat Usability pada Android dan iOS Platforms," *J. Nas. Inform., Teknol. Jaringan*, Vol. 3, No. 2, hal. 184–192, 2019.
- [7] A.T. Sondha, U. Sa'adah, F.F. Hardiansyah, dan M.B.A. Rasyid, "Framework dan Code Generator Pengembangan Aplikasi Android dengan Menerapkan Prinsip Clean Architecture," *J. Nas. Tek. Elektro, Teknol. Inf.*, Vol. 9, No. 4, hal. 327–335, Nov. 2020.
- [8] I.Z.T. Dewi, dkk., "Smart Farming: Sistem Tanaman Hidroponik Terintegrasi IoT MQTT Panel Berbasis Android," *J. Keteknikan Pertan. Trop., Biosist.*, Vol. 9, No. 1, hal. 71–78, 2021.
- [9] R.P. Astutik, "Aplikasi Telegram untuk Sistem Monitoring pada Smart Farming," *J. Teknol., Terap. Bisnis*, Vol. 2, No. 1, hal. 1–6, 2019.
- [10] O. Pandithurai, S. Aishwarya, B. Aparna, dan K. Kavitha, "Agro-Tech: A Digital Model for Monitoring Soil and Crops Using Internet of Things (IOT)," *2017 3rd Int. Conf. Sci. Technol. Eng. Manage.*, 2017, hal. 342–346.
- [11] A.D. Chachadi dan G.R. Rajkumar, "Development of Automated Hydroponic System for Smart Agriculture," *Int. Res. J. Eng. Technol.*, Vol. 8, No. 6, hal. 1273–1278, Jun. 2021.