Sistem *Monitoring* Keadaan Ruang Laboratorium Fakultas Komunikasi dan Informatika di Universitas Muhammadiyah Surakarta

Ragil Burhanudin Pamungkas, Bana Handaga

Program Studi Informatika

Universitas Muhammadiyah Surakarta (UMS)

Surakarta, Indonesia

Om.ragil7@gmail.com, Bana.handaga@gmail.com

Abstraksi—*Monitoring* merupakan kegiatan yang dilakukan untuk memantau, mengawasi, dan mengamati hal-hal tertentu pada suatu tempat. Laboratorium Fakultas Komunikasi dan Informatika (FKI) Universitas Muhammadiyah Surakarta (UMS) telah melakukan *monitoring* dengan menggunakan CCTV, namun *monitoring* menggunakan CCTV terbatas hanya pada citra saja. Maka dari itu dibuatlah sebuah sistem yang bertujuan untuk melakukan *monitoring* keadaan ruang Laboratorium FKI UMS yang bekerja selama 24 jam dalam satu hari. Sistem ini dirancang untuk melakukan *monitoring* suhu di dalam ruangan, asap di dalam ruangan serta gerakan untuk mendeteksi apakah terdapat aktifitas di dalam ruangan, selain itu *monitoring* juga dilakukan pada cahaya untuk mengetahui kondisi pencahayaan untuk menentukan kondisi lampu di dalam ruang serta arus listrik di dalam ruang tersebut. Pengguna dapat melakukan *monitoring* secara *real time* melalui *androidsmartphone* dengan menggunakan koneksi internet, selain itu pengguna juga mendapatkan notifikasi melalui aplikasi *androidsmartphone* dan *short message service* (SMS). *Server* yang digunakan pada penelitian ini adalah *firebase* dengan fitur *real time database*. Metode pengembangan yang digunakan dalam sistem ini adalah SDLC (*Software Development Life Cycle*) dengan model *waterfall*. Berdasarkan pengujian yang dilakukan didapatkan tingkat akurasi untuk notifikasi melalui aplikasi berbasis *android* sebesar 100% sedangkan tingkat akurasi notifikasi melalui SMS sebesar 86,67%.

Katakunci—*androidsmartphone*; *firebase*; *monitoring*; *real time*

I. PENDAHULUAN

Monitoring merupakan kegiatan yang dilakukan untuk memantau, mengawasi, dan mengamati hal-hal tertentu pada suatu tempat. Proses *monitoring* telah menjadi hal penting karena dapat mencegah suatu kelalaian ataupun kesalahan yang terjadi. Keadaan dan penggunaan ruang merupakan hal yang paling sering dipantau, biasanya menggunakan alat CCTV.

Laboratorium Fakultas Komunikasi dan Informatika juga telah melaksanakan proses *monitoring* menggunakan CCTV. *Monitoring* tersebut merekam citra bergerak atau video kegiatan yang dilakukan di dalam laboratorium. Hal tersebut bisa saja mencegah pencurian atau kerusakan alat-alat di laboratorium, namun tidak dengan kerusakan maupun kelalaian lain yang mungkin bisa membahayakan. Seperti halnya kelalaian mematikan, AC, dispenser, komputer atau sejenisnya yang dapat menyebabkan kebakaran. *Monitoring* menggunakan CCTV hanya terbatas pada citra dan tidak dapat mendapatkan data tentang kondisi ruangan secara lengkap.

Sistem *monitoring* adalah sistem yang digunakan untuk melakukan pemantauan guna mendapatkan sebuah data yang akurat dalam kurun waktu tertentu. Salah satu aspek yang penting dalam sistem *monitoring* adalah penyajian data secara *real time*. Sebelumnya telah banyak dilakukan penelitian mengenai sistem *monitoring*, salah satu diantaranya adalah membuat sebuah sistem *monitoring* untuk rumah cerdas. Penelitian tersebut menggunakan Arduino sebagai *micro controller* dan *micro-webserver* serta *smartphone* berbasis *android* sebagai User[1]. Selain itu juga telah dilakukan penelitian membuat sebuah rancang bangun sistem *monitoring* ketinggian permukaan air menggunakan mikrokontroler ATMEGA328P berbasis *web service*. Data yang diamati meliputi suhu, kelembaban, dan ketinggian air, kemudian data tersebut dikirim ke sebuah *webserver* yang menggunakan *MySQL* dan *php*[2]. Tak hanya itu, penelitian lainya juga dilakukan yaitu membuat alat *monitoring* suhu kabel dan trafo, data yang diamati adalah suhu pada kabel trafo, kemudian data tersebut dikirimkan melalui SMS ke pengguna ketika suhu berada diatas 40°C[3].

Berdasarkan masalah diatas serta penelitian-penelitian terdahulu maka dilakukan penelitian yang bertujuan untuk membuat suatu sistem *monitoring* ruang Laboratorium Fakultas Komunikasi dan Informatika (FKI) Universitas Muhammadiyah Surakarta (UMS) yang bekerja selama 24 jam dalam satu hari. Laboratorium FKI UMS merupakan laboratorium yang disediakan oleh Universitas

Muhammadiyah Surakarta untuk menunjang proses belajar mengajar. Di dalam laboratorium FKI terdapat PC, proyektor, AC, dll. Sistem ini dirancang untuk melakukan monitor antara lain suhu ruangan, asap dalam ruangan, gerakan didalam ruangan untuk mengetahui terdapat aktifitas atau tidak, cahaya di dalam ruangan, serta arus listrik yang dapat dipantau secara langsung dan *real time* melalui *androidsmartphone*. Sistem ini diberi nama Simo.

II. METODE

Simo adalah sistem *monitoring* ruang laboratorium FKI yang digunakan untuk memantau keadaan ruang laboratorium FKI. Dengan adanya Simo meskipun pengguna berada di tempat yang jauh tetap dapat melakukan *monitoring* terhadap laboratorium FKI dan mengetahui kondisi didalamnya. Hal-hal yang dipantau meliputi suhu ruangan, asap didalam ruangan, hal ini dilakukan untuk memantau adanya asap atau tidak, kemudian gerakan sehingga pengguna dapat mengetahui apakah terdapat aktifitas di dalam ruangan. Selain itu pemantauan juga dilakukan terhadap cahaya untuk mengetahui kondisi pencahayaan ruangan, berdasarkan kondisi yang diberikan oleh sensor maka dapat diketahui apakah lampu di dalam ruangan menyala atau tidak. Pemantauan yang terakhir adalah arus listrik untuk mengetahui apakah masih terdapat perangkat elektronik yang masih hidup. Selain itu simo juga memiliki fitur notifikasi dari SMS (Short Message Service) serta dari aplikasi Simo berbasis *android*. Metode pengembangan pada Simo menggunakan SDLC (Software Development Life Cycle) dengan model *waterfall*. SDLC model *waterfall* adalah siklus hidup yang didasarkan pada ketergantungan langkah-langkah bertahap untuk menyelesaikan implementasi suatu sistem, dimana setiap langkah tergantung pada penyelesaian langkah sebelumnya[4]. Tahapan pengembangan SDLC dengan model *waterfall* pada penelitian ini meliputi analisis kebutuhan, desain, implementasi, dan pengujian.

A. Analisis Kebutuhan

Analisiskebutuhan dilakukandilakukan untukmentukansoftwaremaupun hardware yang digunakandalampengembanganSimo. Software yang digunakandalampengembanganantaralain text editor *VSCode*, *serverfirebase*, *umlet* untukmembuatactivity diagram. Sedangkanuntukhardware diantaranya yaitulaptop (*Processor intel core i3 1.9 Ghz, 2GB DDR3 RAM, 500GB HDD, Nvidia GeForce 820M 2GB*) sebagai tempat pembuatan sistem. *Raspberry Pi 3* digunakan sebagai *base station* yang berfungsi untuk mengolah data dari *sensor node* yang kemudiandikirimkan keserver *firebase*. serta *NodeMcu* sebagai *nsor node* yang berguna untuk mengumpulkan data dari sensor-sensor yang digunakan. Sensor yang digunakan antaralain, DHT11 sebagai sensor suhu, ACS712 sebagai sensor arus listrik, LDR (*Light Dpendant Resistor*) sebagai sensor cahaya, PIR (*Passive Infrared*) sebagai sensor gerak, serta MQ-5 sebagai sensor asap.

B. Desain

1. Use Case Diagram

Use case diagram merupakan pemodelan untuk tingkah laku (*behavior*) sistem informasi yang akan dibuat, selain itu *use case* juga digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [5]. Pengguna dalam aplikasi Simo adalah dosen, staff atau asisten laboratorium yang bekerja di laboratorium FKI. Pengguna aplikasi Simo dapat melihat data dari sensor setelah melakukan *login*. *Use case diagram user* untuk aplikasi Simo dapat dilihat pada Gambar 1.

Gambar 1. Use case diagram User aplikasi Simo

2. Arsitektur Sistem

Gambar 2. Arsitektur sistem Simo

Simo terdiri dari *hardware* dan *software*. *Hardware* dalam Simo meliputi *raspberry pi* sebagai *base station*, kemudian *sensor node* yang terdiri dari *nodemcu* dan 5 sensor antara lain sensor suhu, asap, gerakan, cahaya, arus listrik. Sedangkan untuk *software* yang digunakan Simo meliputi *software* sensor untuk *nodemcu*, *software* pengolah data untuk *raspberry pi*, dan *software* untuk *database server* serta *software* untuk aplikasi Simo untuk *androidsmartphone*. Arsitektur Simo secara umum dapat dilihat pada Gambar 2.

Base station pada Simo menggunakan *raspberry pi*. *Raspberry pi* adalah perangkat komputer kecil yang menggunakan sistem operasi linux dengan harga terjangkau, fleksibel, dan dapat dimodifikasi serta diprogram menjadi sebuah sistem embedded[6]. Arsitektur Simo menggunakan *raspberry pi 3* dengan spesifikasi *System on Chip (SoC) Broadcom BCM2837, CPU 4x ARM Cortex-A53* dengan frekuensi 1,2 GHz, GPU *Broadcom VideoCore IV*, RAM 1 GB LPDDR2 (900MHz). Kemudian untuk *networking* memiliki 10/100 Ethernet, 2.4 GHz 802.11n *wireless protocol*, Bluetooth 4.1. Media penyimpanan dengan tipe *MicroSD*, GPIO dengan 40-pin header, populated. Selain itu *raspberry pi 3* juga menyediakan port yang meliputi HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI). Simo menggunakan *raspberry pi 3* karena sudah mendukung

koneksi *wireless* tanpa perlu melakukan penambahan modul *wireless* seperti pada *raspberry pi* versi 1 dan 2.

Kemudian pada bagian *sensor node*, Simo menggunakan *nodemcu*. *NodeMcu* merupakan papan pengembang yang telah dilengkapi dengan *ESP8266 low-cost IEEE 802.11 chip* dengan *full TCP/IP stack* dan kemampuan mikrokontroler yang berjalan pada 80Mhz serta memiliki memori sebesar 128KB[7]. Microcontroller yang digunakan papan pengembangan ini adalah Tensilica 32-bit RISC CPU Xtensa LX106. Daya yang diperlukan 3.3 volt. Memiliki 16 Digital I/O pins, dan 1 analog input pin. Flash memory 4 MB, dan SRAM 64 KB. Mendukung koneksi *wireless* dengan *protocol IEEE 802.11 b/g/n*. Simo menggunakan *nodemcu* dikarenakan selain dengan harga yang terjangkau *nodemcu* juga sudah mendukung koneksi *wireless* dengan *protocol IEEE 802.11 b/g/n* yang dapat langsung berkomunikasi dengan *raspberry pi* 3. Selain *nodemcu* *sensor node* memiliki sensor yang meliputi sensor suhu (DHT11), sensor asap (MQ-5), sensor gerakan (PIR), sensor cahaya (ldr), serta sensor arus (ACS712).

Gambar 3. Denah implementasi laboratorium FKI

Berdasarkan arsitektur sistem pada gambar 2, maka dibuat sebuah denah implementasi terhadap ruang laboratorium FKI yang dapat dilihat pada gambar 3. Pada ruang laboratorium FKI ditempatkan 2 *sensor node*. Penggunaan 2 *sensor node* dikarenakan ukuran ruang yang luas sehingga apabila hanya menggunakan 1 *sensor node* kondisi ruangan untuk bagian tertentu tidak dapat terpantau. Sensor node yang pertama berada dekat dengan *base station*, yaitu berada di dalam laboratorium bagian depan. Sedangkan *sensor node* yang kedua diletakkan didepan pintu masuk. Penempatan sensor didepan pintu masuk karena pintu tersebut sebagai jalan masuk serta keluar untuk mahasiswa maupun asisten laboratorium dan dosen. Koneksi UMS Line ke *base station* menggunakan kabel UTP dan konektor RJ45. Sedangkan koneksi dari *sensor node* ke *base station* menggunakan koneksi *wireless* dengan *protocol IEEE 802.11*.

Kemudian untuk bagian *software* dari *sensor node* bekerja dengan mengumpulkan data dari masing-masing sensor. Setelah terkumpul data kemudian dikirim ke *base station* menggunakan media transmisi *wireless* dengan *protocol IEEE 802.11*. Sedangkan untuk *base station* terpasang *software* yang

memiliki fungsi untuk melakukan pengolahan data. *Software* pada *base station* menerima data melalui komunikasi *wireless* dengan *protocol IEEE 802.11*. Data yang diterima kemudian dikirimkan ke *database server*.

Sedangkan untuk *server* menggunakan *firebase*. *Firebase* merupakan sebuah cloud yang disediakan oleh Google dengan menggunakan bahasa *JSON (Java Script Object Notation)*[8]. Selain itu *Firebase* merupakan sebuah teknologi yang dapat membantu untuk membuat sebuah *webapplication* tanpa harus melakukan *server-side programming* sehingga pengembangan menjadi lebih cepat dan mudah [9]. *Firebase* juga mendukung sinkronisasi data secara *real time* dan memberikan tempat penyimpanan *database* sebesar 1 GB dan *downstreambandwith* sebesar 10 GB. Fitur *firebase* lainnya yang digunakan adalah *firebasecloud messaging (FCM)* untuk mengirimkan notifikasi ke aplikasi Simo. FCM dirancang untuk menyediakan koneksi ke perangkat *smartphone* melalui pesan dan notifikasi [10]. Selain itu, fitur *authentication* juga digunakan untuk mengelola perijinan dalam mengakses *database*. User menggunakan aplikasi Simo berbasis *android* sebagai antar muka untuk mendapatkan informasi.

3. Activity Diagram

Gambar 4. Activity diagram

Simo bekerja dengan mengumpulkan data-data sensor melauhi *sensor node*. Kemudian *sensor node* melakukan pengecekan apakah sudah terhubung dengan *base station*. Apabila tidak maka *sensor node* melakukan pengambilan data sensor baru, namun apabila *sensor node* sudah terhubung dengan *base station* maka data kemudian dikirimkan ke *base station*. Data yang diterima oleh *base station* kemudian diolah dan diubah dalam bentuk *JSON* untuk dikirimkan ke *server firebase* setelah melalui tahap autentikasi. Selanjutnya pada aplikasi Simo berbasis *android*, apabila *user* sudah login dan sukses dalam autentikasi *server database firebase*, maka *user* akan menerima data sensor pada aplikasi Simo berbasis *android*. Activity diagram dari Simo dapat dilihat pada Gambar 4.

4. Desain database

Simo menggunakan *firebase* sebagai *server* penyimpanan *database* pada *firebase* menggunakan *NoSQL*. Data yang disimpan dalam *database* berbentuk *JSON* yang hanya terdiri dari *key* and *value*. Berbeda dengan *MySQL*, pada *NoSQL* penyimpanan *database* tidak memiliki struktur data. Contoh rancangan data dalam bentuk *MySQL* dapat dilihat pada Gambar 5.

data current	data ldr	data users
id : int(11)	id : int(11)	pengenal : varchar(255)
value : varchar(255)	value : varchar(255)	dibuat : varchar(255)
time : varchar(255)	time : varchar(255)	masuk : varchar(255)

Gambar 5. Contoh rancangan data dalam bentuk *MySQL*

Berdasarkan Gambar 5, data akan diubah dalam bentuk *JSON*. contoh kasus pengolahan data, *base station* akan mengirimkan data sensor temperature dengan *id*="LQ8N30kyEEdmk9-52x5", *value*=38, *time*="2018-10-31 09:05:78.67483", maka data tersebut akan disimpan dalam format *JSON* sebagai "temperature":{"time":"2018-10-31 09:06:59.510735","value":38}}. Selain itu *firebase* juga akan mengirimkan data ke aplikasi Simo dalam format *JSON*.

Pengenal ↓	Penyedia	Dibuat tanggal	Masuk	UID pengguna
om.ragil7@gmail.com				ncmR1QQA..

Gambar 6. Fitur *authentication firebase*

Kemudian untuk data *user* yang digunakan untuk login akan disimpan pada fitur *authentication firebase*. Contoh data *user* dapat dilihat pada Gambar 6. Pengenal merupakan email yang digunakan untuk login, penyedia berisi jenis akun yang digunakan untuk masuk, pada gambar 6 disimbolkan dengan ikon surat yang berarti email, sedangkan UID pengguna berisi gabungan huruf yang digunakan sebagai id pengenal unik.

5. Desain user interface

Gambar 7. Desain *User interface* Simo

Desain antar muka aplikasi Simo dapat dilihat pada gambar 7, yang meliputi antarmuka login (a), daftar (b), home (c), riwayat (d). Antar muka home akan berisi keterangan suhu, gerakan, asap, arus dan cahaya terbaru. Sedangkan antarmuka riwayat akan berisi riwayat sesuai dengan kategori meliputi suhu, gerakan/aktifitas, asap, arus, serta cahaya.

III. HASIL DAN PEMBAHASAN

A. Implementasi

Simo akan dikembangkan berdasarkan analisis kebutuhan beserta desain yang telah dibuat. Pembuatan aplikasi Simo berbasis *android* menggunakan framework *react-native* dengan Bahasa pemrograman JavaScript dan *software* untuk *sensor node* menggunakan *arduino ide* berbasis C, sedangkan *software* pada *raspberry pi* menggunakan bahasa *python3*.

```

22 | const ref = firebase.database().ref("data/");
23 | ref.child("current").limitToLast(1).on("child_added", (snapshot) => this.setState({ current: snapshot.toJSON() }));

```

Gambar 8. Kode mengambil data terbaru

Pengembangan Simo dimulai dengan mengubah data dari analisis kebutuhan, desain, beserta *activity diagram* sebagai acuan. Antarmuka *home* pada simo bertujuan untuk menampilkan data terbaru sehingga apabila terjadi penambahan data untuk setiap sensor maka antarmuka tersebut akan menampilkan data terbaru, kode untuk melakukan pengambilan data terbaru dapat dilihat pada gambar 8. Kode *ref.child("current")* digunakan untuk memilih data sensor dari *server firebase*. Sedangkan untuk menampilkan data secara *real time* ini menggunakan fungsi *on("child added")* yang berfungsi untuk melakukan pengambilan data apabila ada perubahan data pada *server firebase* sesuai dengan kategori sensor, sedangkan fungsi *limitToLast(1)* untuk menampilkan satu data terbaru.

```

62 | const ref = firebase.database().ref("data/");
63 | ref.child("temperature").orderByChild("time").startAt(startDate).endAt(endDate)
64 | .once("value").then((snapshot) => {
65 |   let arrayOfKeys = Object.keys(snapshot.val())
66 |   .sort()
67 |   .reverse();
68 |   let results = arrayOfKeys
69 |     .map((key) => snapshot.val()[key]);
70 |   let arrayOfKeys = string[]
71 |   let referenceToOldestTime = snapshot.val()[arrayOfKeys[arrayOfKeys.length - 1]].time;
72 |   this.setState({ data: results, timeReference: referenceToOldestTime, loading: false });

```

Gambar 9. Kode untuk mengambil data riwayat

Kemudian antarmuka riwayat pada simo bertujuan untuk menampilkan riwayat data yang disimpan pada *serverfirebase*. Pengambilan data riwayat memiliki fitur untuk menampilkan data sesuai dengan tanggal yang ditentukan, kode untuk mengambil data riwayat dapat dilihat pada gambar 9. Pengambilan data menggunakan fungsi *filter* yaitu, *startAt(startDate)* pada baris 63 untuk batas tanggal awal, dan *endAt(endDate)* pada baris 63 untuk batas tanggal akhir, dimana *startDate* merupakan *variable* yang berisi tanggal awal, dan *endDate* adalah *variable* yang berisi batas tanggal akhir, kemudian fungsi *once("value")* pada baris 64 berfungsi untuk melakukan pengambilan data satu kali, sehingga apabila pengguna ingin mendapatkan data terbaru perlu menekan fungsi *filter* lagi. Kemudian data yang telah di proses akan disimpan dalam *variable* data.

B. Hasil Penelitian

Hasil dari penelitian ini merupakan sebuah sistem monitoring keadaan ruang laboratorium FKI UMS. Antarmuka aplikasi dapat dilihat pada gambar 10 dan 11.

Gambar 10. Antarmuka Simo untuk riwayat

Antarmuka riwayat pada gambar 10, Menampilkan riwayat data yang tersimpan pada server sesuai dengan kategori yang telah ditentukan meliputi gerakan, arus, asap, cahaya, serta suhu.

Gambar 11. Antarmuka Simo untuk grafik suhu

Data yang telah dikirim ke *server* akan ditampilkan dalam aplikasi Simo berbasis *android*. Salah satu bentuk penyajian data adalah dengan menggunakan grafik yang dapat dilihat pada gambar 11. Pada gambar tersebut terdapat grafik yang dapat dipilih sesuai dengan tanggal. Data yang ditampilkan pada grafik merupakan rerata dari tiap 6 data untuk sumbu y, sedangkan untuk data pada sumbu x didapatkan dari data terakhir berdasarkan 6 data yang telah di rerata.

C. Pengujian

Pengujian dilakukan melalui 6 tahap, tahap ke-1 hingga ke-5 untuk melakukan pengujian sensor. Sedangkan tahap ke-6 untuk melakukan pengujian sistem Simo secara keseluruhan. Pengujian tahap ke-1 adalah pengujian sensor suhu, pengujian ini dilakukan untuk mengetahui tingkat akurasi suhu dengan cara membandingkan suhu yang dihasilkan oleh sensor dengan suhu yang dihasilkan oleh termometer konvensional. Pengujian dilakukan 2 kali dengan masing masing data yang diambil sebanyak 100 data. Pengujian sensor suhu pertama dilakukan di dalam ruangan dengan suhu 20 – 22 °C, sedangkan pengujian kedua dilakukan di dalam ruangan dengan suhu 29-30 °C. berdasarkan pengujian pertama dan kedua akan diambil nilai suhu minimum, maksimum beserta rerata. Hasil pengujian dapat dilihat pada gambar 12.

Gambar 12. Grafik hasil pengujian sensor suhu

Berdasarkan hasil pengujian pada gambar 12, diketahui bahwa untuk pengujian pertama suhu minimum untuk sensor DHT 11 adalah 21 °C, suhu maksimum adalah 23 °C, sedangkan reratanya adalah 21,48 °C kemudian untuk termometer konvensional suhu minimumnya adalah 21 °C, suhu maksimum sebesar 24 °C, dan rerata suhu sebesar 21,92 °C, berdasarkan hasil tersebut maka diketahui bahwa selisih dari rerata suhu yang terbaca oleh sensor dengan suhu yang terbaca oleh thermometer konvensional sebesar 0,44 °C. Sedangkan untuk pengujian kedua suhu minimum untuk sensor DHT 11 adalah 29 °C, suhu maksimum adalah 30 °C, sedangkan reratanya adalah 29,03 °C kemudian untuk termometer konvensional suhu minimumnya adalah 29 °C, suhu maksimum sebesar 30 °C, dan rerata suhu sebesar 29,55 °C, berdasarkan hasil tersebut maka diketahui bahwa selisih

dari rerata suhu yang terbaca oleh sensor dengan suhu yang terbaca oleh termometer konvensional sebesar 0,55 °C. berdasarkan hasil tersebut maka diketahui bahwa rerata selisih antara suhu yang terbaca oleh sensor dan termometer konvensional dari dua pengujian yang dilakukan sebesar 0.50 °C, dengan kesalahan yang cukup kecil maka sensor DHT 11 dapat digunakan untuk mendeteksi suhu secara akurat. Selain itu berdasarkan pada gambar 11, ditentukan suhu untuk batas normal sebesar 30 °C, apabila suhu lebih dari 30 °C maka sistem akan mendeteksi adanya bahaya.

Pengujian tahap ke-2 adalah pengujian sensor MQ-5. Pengujian dilakukan dalam 2 kondisi dengan masing-masing pengambilan data sebanyak 100 kali, berdasarkan data tersebut akan diambil nilai minimum, maksimum dan reratanya. Pengujian dilakukan dengan cara membaca 10 bit ADC (*Analog Digital Converter*) untuk setiap kondisi. Pengujian kondisi pertama sensor MQ-5 untuk mengetahui 10 bit ADC didalam ruangan dengan sumber asap normal atau tidak ada, pengujian kondisi kedua untuk mengetahui nilai dari 10 bit ADC dengan sumber asap kertas. Hasil dari 10 bit ADC tersebut akan digunakan untuk menentukan batas normal asap di dalam ruangan. Hasil pengujian sensor MQ-5 dapat dilihat pada gambar 13.

Gambar 13. Grafik hasil pengujian sensor asap MQ-5

Gambar 14. Kondisi asap pada nilai maksimum kondisi 2

Berdasarkan hasil pengujian pada gambar 13, untuk kondisi pertama didapatkan data 10 bit ADC dengan nilai minimum sebesar 152, nilai maksimum sebesar 156, dan

rerata sebesar 153,36. Sedangkan kondisi kedua dengan nilai minimum sebesar 430, nilai maksimum 807, dan rerata 544,60. Nilai maksimum pada pengujian kedua terjadi ketika tingkat asap sangat pekat, hal tersebut dapat dilihat pada Gambar 14. Berdasarkan data tersebut maka batas bahaya minimal untuk 10 bit ADC yang terbaca adalah 430 diambil dari nilai minimum pengujian kedua, sehingga apabila 10 bit ADC yang terbaca mulai dari 430 maka sistem Simo akan mendeteksi bahwa ruangan dalam keadaan bahaya.

Pengujian tahap ketiga adalah pengujian sensor gerak PIR, pengujian sensor gerak PIR ini untuk menentukan pada jarak berapa sensor ini masih dapat mendeteksi gerakan. Pengujian dilakukan dengan jarak 30 cm dan 5 m, sedangkan pengambilan data untuk tiap jarak sebanyak 30 kali. Hasil pengujian dapat dilihat pada tabel 1.

TABEL 1. HASIL PENGUJIAN SENSOR GERAK PIR

Jarak (cm)	Sensor 1		Sensor 2	
	Berhasil	Gagal	Berhasil	Gagal
30	30	0	30	0
500	28	2	30	1

Tabel 1 Menunjukkan hasil dari pengujian yang berupa jumlah keberhasilan dan kegagalan pada percobaan. Hasil dari pengujian tersebut kemudian digunakan untuk menghitung tingkat akurasi sensor PIR dengan persamaan 1:

$$\text{persentaseakurasi} = \frac{\text{jumlah keberhasilan}}{\text{jumlah percobaan}} \times 100 \quad (1)$$

Hasil perhitungan menghasilkan persentase akurasi pada jarak 30 cm untuk sensor 1 dan sensor 2 sebesar 100 %, sedangkan pada jarak 500 cm tingkat akurasi untuk sensor 1 sebesar 96,67% dan sensor 2 sebesar 93,33%. Berdasarkan data tersebut diketahui bahwa sensor PIR masih dapat bekerja secara efektif pada jarak 500 cm atau 5 m.

Gambar 15. Grafik hasil pengujian sensor LDR

Pengujian tahap keempat adalah pengujian sensor cahaya yaitu LDR. Pengujian sensor ini bertujuan untuk menentukan aktif atau tidaknya lampu di dalam ruangan. Pengujian dilakukan dengan mengambil data hasil dari pembacaan sensor yang berupa 10 bit ADC pada keadaan lampu aktif sebanyak 100 kali. Hasil pengujian dapat dilihat pada gambar 15. Berdasarkan data yang diperoleh diketahui bahwa dalam

keadaan lampu ruangan laboratorium menyala didapatkan 10 bit ADC untuk sensor 1 dengan nilai minimum sebesar 759, nilai maksimum sebesar 817. Sedangkan 10 bit ADC untuk sensor 2 dengan nilai minimum sebesar 766, dan nilai maksimum sebesar 826. Berdasarkan data tersebut maka ditentukan nilai minimum 10 bit ADC untuk keadaan lampu menyala sebesar 759. Apabila 10 bit ADC yang terbaca mulai dari 759 maka sistem akan memberikan status lampu aktif.

Pengujian tahap kelima adalah pengujian sensor arus ACS712. Pengujian dilakukan tiga kali dengan kondisi yang berbeda. Kondisi pertama yaitu pengandilan data ketika sensor arus tidak terdapat peralatan yang terhubung, kondisi kedua dilakukan ketika terdapat sebuah kipas angin terhubung, sedang kondisi ketiga yaitu ketika setrika terhubung dengan sensor. Untuk setiap kondisi dilakukan pengambilan data sebanyak 50 kali. Untuk menghitung arus digunakan persamaan 2 yang didapatkan dari datasheet sensor ACS712.

$$\text{Arus} = \left(\left(\frac{10 \text{ bit adc terbaca}}{\text{total 10 bit adc}} \times 3,3 \right) - 2,5 \right) : 0,1 - 0,32 \quad (2)$$

Angka 0,32 didapatkan dari hasil pengujian ketika sensor membaca arus tanpa beban. Berdasarkan persamaan tersebut didapatkan hasil yang dapat dilihat pada Gambar 16.

Gambar 16. Hasil pengujian sensor arus

Berdasarkan hasil pengujian sensor arus pada Gambar 16. Untuk kondisi pertama tanpa beban, arus terendah yang terbaca adalah 0 A, sedangkan untuk kondisi kedua dengan beban kipas angin arus terendah yang terbaca adalah 0,10 A dan arus tertinggi sebesar 0,26 A. Kemudian untuk kondisi ketiga dengan beban setrika arus terendah yang terbaca adalah 0 A, hal ini terjadi ketika bimetal didalam setrika melengkung menjauhi kontak karena diatas suhu yang ditetapkan sehingga arus listrik terputus, sedangkan arus tertinggi yang terbaca sebesar 1,20 A.

TABEL II. KONDISI UNTUK MENGIRIMKAN NOTIFIKASI

Kondisi	Arus	Asap (10 bit ADC)	Suhu (°C)	PIR	LDR (10 bit ADC)	Notifikasi	
						Aplikasi	SMS
1	0-20A	<430	<30	0-1	0-1023	Tidak	Tidak

2	0-20A	<430	>30	0-1	0-1023	Ya	Ya
3	0-20A	>430	<30	0-1	0-1023	Ya	Ya
4	0-20A	>430	>30	0-1	0-1023	Ya	Ya

Hasil dari pengujian tahap pertama hingga tahap kelima kemudian digabungkan untuk menentukan kondisi yang akan membuat sistem mengirimkan notifikasi kepada pengguna. Hasil tersebut dapat dilihat pada tabel 2. notifikasi hanya dipengaruhi oleh data dari suhu dan asap. Berdasarkan tabel 2 diketahui bahwa, arus berisi data dalam satuan Ampere, data ini tidak mempengaruhi notifikasi. Kemudian asap berisi data dalam bentuk 10 bit ADC, apabila 10 bit ADC asap diatas 430 (>430) maka sistem akan mengirimkan notifikasi, kemudian untuk suhu berisi data dalam bentuk derajat celcius, apabila suhu diatas 30 °C maka sistem akan mengirimkan notifikasi. PIR berisi data 0 dan 1 apabila 0 tidak terdapat aktifitas sedangkan 1 maka terdapat aktifitas, data dalam PIR tidak mempengaruhi fungsi notifikasi. Selanjutnya LDR berisi data dalam bentuk 10 bit ADC dengan rentang nilai 0-1023. Data 10 bit ADC pada bagian LDR ini tidak mempengaruhi notifikasi sistem.

Pengujian tahap ke-6 merupakan pengujian sistem Simo secara keseluruhan. Pengujian ini bertujuan untuk memastikan apakah Simo dapat bekerja sesuai dengan yang diharapkan seperti pada tabel 2. Notifikasi pada sistem Simo hanya dipengaruhi oleh sensor asap dan suhu. Pengujian tahap ke-6 dilakukan 4 kali dengan kombinasi kondisi sensor asap normal dan bahaya, serta sensor suhu normal dan bahaya dengan pengambilan data sebanyak 30 kali pada masing masing kombinasi. Hasil dari pengujian sistem Simo secara keseluruhan dapat dilihat pada tabel 3.

TABEL III. HASIL PENGUJIAN SISTEM SIMO KESELURUHAN

Kondisi	Asap	Suhu	Notifikasi Berhasil		Notifikasi Gagal	
			Aplikasi	SMS	Aplikasi	SMS
1	<403	<30	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
2	<403	<30	30	25	0	5
3	>403	>30	30	25	0	5
4	>403	>30	30	26	0	2

Tabel 3 menunjukkan hasil dari pengujian yang berupa jumlah notifikasi yang berhasil diterima oleh pengguna. Hasil dari pengujian tersebut kemudian digunakan untuk menghitung tingkat akurasi notifikasi dengan persamaan 3:

$$\text{persentase akurasi} = \frac{\text{notifikasi berhasil}}{\text{jumlah percobaan}} \times 100 \quad (3)$$

Hasil perhitungan menghasilkan persentase akurasi yang ditunjukkan pada tabel 4.

TABEL IV. PERSENTASI AKURASI NOTIFIKASI

Kondisi	Asap	Suhu	Akurasi(%)	
			Aplikasi	SMS

1	<403	<30	Tidak Ada	Tidak Ada
2	<403	<30	100	83,84
3	>403	>30	100	83,84
4	>403	>30	100	86,67

Tingkat akurasi dapat dilihat pada gambar 17. Kondisi 1 tidak memiliki tingkat akurasi karena pada kondisi 1 data suhu maupun asap dibawah batas bahaya. Sedangkan untuk kondisi 2 tingkat akurasi notifikasi Aplikasi sebesar 100% sedangkan SMS sebesar 83,34%. Kondisi 3 menunjukkan tingkat akurasi fitur notifikasi dari aplikasi Simo berbasis *android* sebesar 100% sedangkan tingkat akurasi dari fitur notifikasi melalui SMS sebesar 83,34%. Kemudian untuk kondisi 4 tingkat akurasi notifikasi aplikasi sebesar 100% dan sms sebesar 86,67%. Berdasarkan kondisi dua, tiga, empat diambil rerata sehingga tingkat akurasi notifikasi berbasis *android* sebesar 100% sedangkan akurasi melalui SMS sebesar 84,78%. Akurasi pengiriman notifikasi melalui SMS dipengaruhi oleh waktu jeda pengiriman notifikasi yang terlalu cepat serta isi konten yang sama di dalam sms, sehingga pesan yang berulang tidak dapat dikirimkan oleh *server* untuk menghindari spam.

Gambar 17. Tingkat Akurasi Simo

IV. KESIMPULAN

Berdasarkan hasil penelitian yang diperoleh, sistem *monitoring* keadaan ruang laboratorium fakultas komunikasi dan informatika di UMS dapat membantu pengguna melakukan pengamatan secara *real time* tentang kondisi ruang tanpa perlu berada di dalam ruangan. Selain itu pengguna juga dapat menerima notifikasi apabila terjadi keadaan yang berbahaya seperti tanda-tanda kebakaran.

Simo memiliki kekurangan yaitu penggunaan sensor asap yang belum dikalibrasi sehingga hanya dapat menampilkan hasil pembacaan sensor yang berupa 10 *bit* ADC, selain itu penggunaan sumber daya listrik tidak menggunakan baterai, apabila listrik dalam ruang tersebut terputus maka fungsi dari Simo akan berhenti. Adapun solusi untuk mengatasinya adalah dengan melakukan kalibrasi sensor asap, serta memberikan sumber daya listrik dari baterai tersendiri. Jadi apabila aliran listrik terputus maka Simo akan tetap dapat bekerja.

DAFTAR PUSTAKA

- [1] Rajeev Piyare and Seong Ro Lee, "Smart Home-Control and Monitoring System Using Smart Phone Smart Home-Control and Monitoring System Using Smart Phone," no. June, 2014.
- [2] R. Fikri *et al.*, "Rancang Bangun Sistem Monitoring Ketinggian Permukaan Air Menggunakan Mikrokontroler ATMEGA328P Berbasis

Web Service," vol. V, no. 2, pp. 42–49, 2015.

- [3] U. Fadlilah and N. Saniya, "Monitoring Suhu Kabel Trafo Melalui Tampilan LCD dan SMS," *Emitor*, vol. 17, no. 02, pp. 01–08, 2017.
- [4] A. M. Langer, *Guide to Software Development: Designing and Managing the Life Cycle*, 2nd ed. Springer Publishing Company, Incorporated, 2016.
- [5] A. Hendini, "Pemodelan UML sistem informasi *Monitoring* Penjualan dan stok barang," *Pemodelan Uml Sist. Inf. Monit. Penjualan Dan Stok Barang (Studi Kasus Distro Zhezha Pontianak)*, vol. IV, no. 2, pp. 107–116, 2016.
- [6] S. G. Nikhade, "Wireless sensor network system using Raspberry Pi and zigbee for environmental monitoring applications," *2015 Int. Conf. Smart Technol. Manag. Comput. Commun. Control. Energy Mater.*, no. May 2015, pp. 376–381, 2015.
- [7] P. A. F. Rezeck, H. Azpurua, and L. Chaimowicz, "HeRo : An Open Platform for Robotics Research and Education," pp. 2–7, 2017.
- [8] P. P. Shah, A. A. Patil, and S. S. Ingleshwar, "IoT based smart water tank with Android application," *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, pp. 600–603, 2017.
- [9] K. N. M. Kumar, K. Akhi, S. K. Gunti, M. Sai, and P. Reddy, "Implementing Smart Home Using Firebase," *Int. J. Res. Eng. Appl. Sci.*, vol. 6573, no. 10, pp. 193–198, 2016.
- [10] L. Moroney, "Firebase Cloud Messaging BT - The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform," L. Moroney, Ed. Berkeley, CA: Apress, 2017, pp. 163–188.