Buletin Poltanesa Vol. 23 No. 1 (Juni 2022) p-ISSN 2721-5350 e-ISSN 2721-5369
https://doi.org/10.51967/tanesa.v23i1.1285 © 2022 Politeknik Pertanian Negeri Samarinda

Sistem Kendali Jarak Jauh untuk *Smart Home* Melalui Aplikasi Android Menggunakan NodeMCU dan *Firebase*

Daniel Martomanggolo Wonohadidjojo *
Informatika, Universitas Ciputra Surabaya, Surabaya, 60219
daniel.m.w@ciputra.ac.id
**Corresponding author*

Hansel Santoso
Informatika, Universitas Ciputra Surabaya, Surabaya, 60219
hsantoso@student.ciputra.ac.id

Abstrak—Smart home merupakan rumah dengan beberapa perangkat pintar yang terpasang di dalam rumah. Perangkat pintar ini merupakan bagian dari Internet of Things (IoT), dengan tujuan untuk membantu pemilik rumah dalam memantau dan mengendalikan keadaan dalam rumah. Perangkat pintar dapat dioperasikan oleh pemilik rumah di dalam maupun di luar rumah dengan menggunakan cloud storage. Dalam penelitian ini hal tersebut diimplementasikan dengan perangkat keras NodeMCU. Setiap sensor yang terhubung dengan NodeMCU mengirimkan data ke pusat untuk dikirimkan ke cloud storage yang bernama Firebase menggunakan metode WebSocket. Data yang disimpan di Firebase akan dikirim ke aplikasi Android yang dibangun menggunakan Flutter. Dengan demikian sistem ini mampu membantu pemilik rumah untuk memantau dan mengendalikan perangkat elektronik di dalam rumah menggunakan aplikasi Android dimana Aplikasi ini dihubungkan dengan NodeMCU menggunakan cloud real-time database.

Kata Kunci— Smart home, NodeMCU, Flutter, Firebase, real-time database, Cloud

I. PENDAHULUAN

Fungsi utama dari *smarthome* adalah untuk mempermudah pekerjaan rumah (Ruangmom, 2021). Namun, pengguna dapat mengalami kecerobohan seperti lupa melakukan sesuatu kepada perangkat *IoT* tertentu di rumah yang merugikan pengguna misalnya pengguna lupa mematikan perangkat elektronik, saat telah berada di luar rumah (Qader & Sahib, 2021). Teknologi *smart home* mempermudah pengguna namun masih memiliki kekurangan yaitu tidak semua perangkat *IoT* dapat diakses jika pengguna sedang tidak berada di rumah (Alzahrani et al., 2020). Oleh karena itu, sistem *smart home* [i]dalam penelitian ini adalah perangkat *NodeMCU*, dan tidak menggunakan perangkat yang lebih sering digunakan yaitu microcontroller *Arduino* sebab *Arduino* tidak dapat terhubung dengan *Wi-Fi* secara langsung tanpa modul tambahan (Kusumah & Susila, 2020).

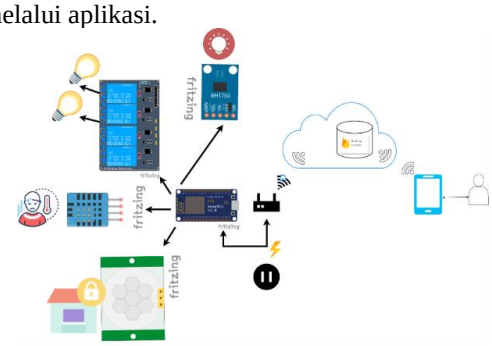
Node MCU bekerja sebagai pusat untuk memantau dan mengendalikan beberapa sensor sekaligus menggunakan *cloud real-time database* (Ada, 2020; Guadagnini et al, 2019) dengan metode *WebSocket* untuk pengiriman dan penerimaan data (Łasocha & Badurowicz, 2021).

Tujuan penelitian ini adalah membangun sistem smart home yang dapat digunakan untuk mengendalikan lampu, memantau temperatur, kelembaban ruang dan gerakan melalui aplikasi Android yang terhubung dengan NodeMCU melalui cloud storage. Hal tersebut diimplementasikan dengan cara memantau parameter pada aplikasi yang ditangkap oleh sensor melalui NodeMCU dan cloud storage.

II. METODOLOGI

A. Desain Arsitektur

Gambar 1 menjelaskan bagian-bagian yang dibuat pada penelitian ini. Arsitektur sistem ada penelitian ini terdiri dari 2 bagian yaitu bagian perangkat *smart home* menggunakan sensor dan *NodeMCU*, dan bagian aplikasi yang akan berinteraksi dengan pengguna untuk memantau dan mengendalikan perangkat *smart home* mereka melalui *Firebase real-time database*. Sistem ini bekerja menggunakan aplikasi. Aplikasi ini memiliki beberapa perintah yang dapat digunakan pengguna untuk memantau dan mengendalikan perangkat mereka. Perintah ini dikirim melalui aplikasi dan kemudian diterima oleh *Firebase*. Perintah yang diterima oleh *Firebase* selanjutnya diteruskan ke *NodeMCU*. Dari *NodeMCU*, perintah ini akan dilaksanakan menggunakan perangkat sensor yang sudah terpasang. Setelah perangkat melaksanakan perintah yang diterima, sensor akan memberikan hasil kepada *NodeMCU* yang akan di unggah ke *Firebase* untuk ditampilkan ke pengguna melalui aplikasi.

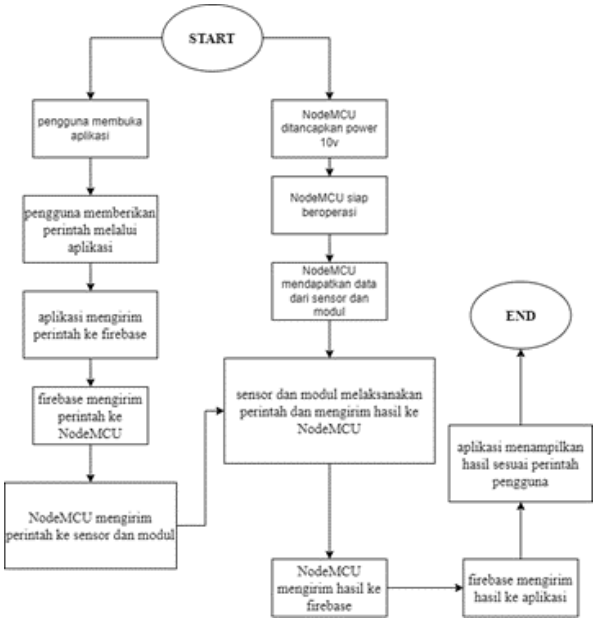


Gambar 1. Arsitektur Sistem

https://doi.org/10.51967/tanesa.v23i1.1285 © 2022 Politeknik Pertanian Negeri Samarinda

B. Desain Alur Kerja

Flowchart pada Gambar 2 dimulai dengan pengguna membuka aplikasi yang tersambung pada internet, dan *NodeMCU* siap beroperasi. Ketika pengguna ingin melakukan sesuatu, pengguna mengirimkan perintah yang telah disediakan dalam aplikasi. Perintah ini akan diterima oleh *Firebase* yang diteruskan ke *NodeMCU*. Dari *NodeMCU*, perintah akan diteruskan ke sensor dan modul untuk dilaksanakan. Setelah perintah dilaksanakan, hasil akan di unggah ke *Firebase* agar hasil dapat ditampilkan ke pengguna melalui aplikasi. Apabila, pengguna sudah selesai memberi perintah, maka *NodeMCU* akan tetap siaga untuk menerima perintah baru dari pengguna melalui aplikasi.

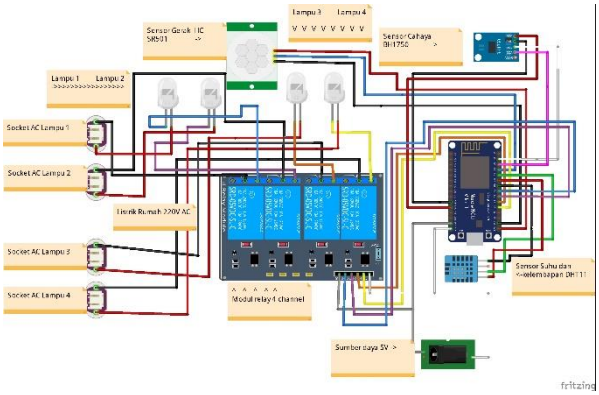


Gambar 2. Diagram *Flowchart* sistem secara keseluruhan

C. Desain Skema Diagram

Gambar 3 merupakan skema rancangan yang akan digunakan dalam penelitian ini. Rancangan berjalan dimulai dengan memberikan daya kepada *NodeMCU*. Setelah *NodeMCU* mendapatkan daya, *NodeMCU* dapat menjalankan beberapa sensor dan modul yang terhubung. Setiap sensor dan modul yang terhubung, memerlukan daya listrik yang diambil dari *NodeMCU*. Oleh karena itu, sangat penting untuk *NodeMCU* mendapatkan daya listrik yang cukup sesuai dengan jumlah daya listrik perangkat yang digunakan. Sensor dan modul yang digunakan dalam skema adalah sensor suhu dan kelembaban *DHT11* untuk mengetahui suhu dan kelembaban ruangan (Hakiki et al, 2020), sensor pengukur cahaya *BH1750* untuk mengetahui intensitas cahaya (Harding et al, 2018), Sensor Gerak *HC SR501* untuk mendeteksi gerakan dan modul *relay* 4 *Channel* untuk memutus dan menyambung aliran listrik kutub negatif lampu rumah (Rahmatulloh, 2020).

Tiap sensor dan modul yang terhubung ke *NodeMCU* menggunakan kabel yang harus terhubung pada 3 pin penting di setiap perangkat, yaitu kutub daya positif, kutub daya negatif dan kutub untuk transmisi data. Hal ini juga berlaku pada modul, dengan perbedaan pada jumlah *channel* dan berapa banyak c*hannel* yang digunakan. Semakin banyak jumlah *channel* yang dibutuhkan, maka semakin banyak kutub transmisi data yang perlu dihubungkan (Septama et al, 2018; Susilo et al 2021).



Gambar 3. Diagram skema perangkat keras

III. HASIL DAN PEMBAHASAN

A. Implementasi Hardware

1. Kode NodeMCU

NodeMCU merupakan komponen utama penelitian ini yang terhubung ke *Firebase* untuk dikendalikan melalui *smartphone* pengguna. *Smartphone* menggunakan sistem operasi Android. Segmen kutipan kode yang berfungsi untuk menghubungkan *NodeMCU* ke *Firebase* dapat dilihat sebagai berikut:

```
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#define FIREBASE_HOST "https://fir-smarthome-
c52b0-default-rtdb.asia-
southeast1.firebasedatabase.app/"
#defineFIREBASE_AUTH "secret"
#define WIFI_SSID "    "
#define WIFI_PASSWORD "     "
const long interval=10000;
FirebaseData firebaseData;
void setup() {
 pinMode(LED_BUILTIN, OUTPUT);
 Serial.begin(9600);
 // sambung ke wifi
 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
 Serial.print("connecting");
 while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
 }
 Serial.println();
 Serial.print("connected: ");
 Serial.println(WiFi.localIP());
 Firebase.begin(FIREBASE_HOST,
FIREBASE_AUTH);
```

2. Kode BH1750

Sensor *BH1750* merupakan sensor yang terhubung dengan *NodeMCU* menggunakan kabel bertujuan untuk mendapatkan data cahaya dalam satuan *lux*. Segmen kutipan kode yang berfungsi untuk menghubungkan *BH1750* ke *NodeMCU* dapat dilihat sebagai berikut:

https://doi.org/10.51967/tanesa.v23i1.1285 © 2022 Politeknik Pertanian Negeri Samarinda


```
Void loop() {
//awal kode pembacaan cahaya sekaligus push ke
firebase
 float lux = lightMeter.readLightLevel();
 lux = lux - 1.5;
 Serial.print("Light: ");
 Serial.print(lux);
 Serial.println(" lx");
 delay(1000);
if(Firebase.setFloat(firebaseData,
"/Data/Light:", lux)){
 Serial.println("lux cahaya terkirim");
 } else{
  Serial.println("lux cahaya tidak terkirim");
  Serial.println("Karena:          "          +
firebaseData.errorReason());
  }
  delay(1000);
 //akhir kode pembacaan cahaya sekaligus push
ke firebase
}
```

3. Kode HC SR501

HC SR501 merupakan sensor yang terhubung dengan *NodeMCU* menggunakan kabel bertujuan untuk mendapatkan data gerak menggunakan *infrared*. Segmen kutipan kode yang berfungsi untuk menghubungkan *HC SR501* ke *NodeMCU* dapat dilihat sebagai berikut:

```
 void loop() {
 float pir = digitalRead(piroutpin);
 if (pir == HIGH)
 {
  Firebase.setFloat(firebaseData,
 "/Data/Motion:", pir);
  Serial.println("Gerakan Terdeteksi!");
```

4. Kode DHT11

DHT11 merupakan sensor yang terhubung dengan *NodeMCU* menggunakan kabel bertujuan untuk mendapatkan data suhu dengan satuan *lux* dan kelembaban dengan satuan persen. Segmen kutipan kode yang berfungsi untuk menghubungkan *DHT11* ke *NodeMCU* dijabarkan sebagai berikut:

```
void loop() {
float pir = digitalRead(piroutpin);
if (pir == HIGH)
{
 Firebase.setFloat(firebaseData,
"/Data/Motion:", pir);
 Serial.println("Gerakan Terdeteksi!");
 }
```

5. *Kode Modul relay*

Modul *relay* merupakan perangkat yang terhubung dengan *NodeMCU* menggunakan kabel bertujuan untuk mendapatkan memantau dan mengendalikan lampu yang terhubung dengan perangkat. Segmen kutipan kode yang berfungsi untuk menghubungkan Modul *relay* ke *NodeMCU* dapat dilihat sebagai berikut: 5.

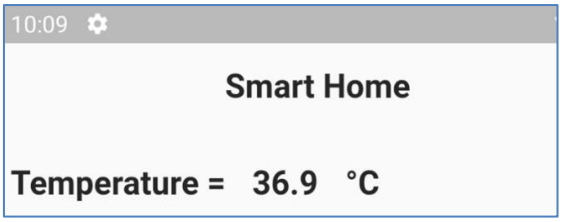
```
 void loop() {
 //awal kode pembacaan dan kendali status 4
 lampu sekaligus push ke firebase
   if
(Firebase.get(firebaseData,"/LightState1/swit
ch")) {//Lampu1
    String lampu1 = firebaseData.stringData();
      if (lampu1=="false"){
        digitalWrite(lampurelay1,HIGH);
//Device1 is ON
      }
      else if (lampu1=="true"){

digitalWrite(lampurelay1,LOW);//Device1 if
OFF}}}
```

.B. Implementasi Software

1. Memantau Suhu

Fitur untuk memantau suhu dapat berjalan sesuai dengan rencana yang dibuat menggunakan *Flutter*. Hasil dari implementasi dapat dilihat pada Gambar 4 dan cuplikan kode yang digunakan adalah sebagai berikut:

```
Padding(
   padding: const EdgeInsets.all(8.0),
   child: Text(
snapshot.data.snapshot.value["Humidity:"].toS
tring(),
 style: TextStyle(
 fontSize: 20, fontWeight: FontWeight.bold),
 ),
),
```

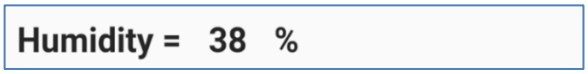


Gambar 4. Aplikasi memantau suhu secara *real-time*

2. Memantau Kelembaban

Fitur untuk memantau kelembaban dapat berjalan sesuai dengan rencana yang dibuat menggunakan *Flutter*. Hasil dari implementasi dapat dilihat pada Gambar 5 dan cuplikan kode yang digunakan dapat dilihat sebagai berikut:

```
Padding(
   padding: const EdgeInsets.all(8.0),
   child: Text(
snapshot.data.snapshot.value["Humidity:"].toS
tring(),
 style: TextStyle(
 fontSize: 20, fontWeight: FontWeight.bold),
),
),
```



Gambar 5. Aplikasi memantau kelembaban secara *real-time*

3. Memantau dan mengendalikan Gerak

Fitur untuk memantau dan mengendalikan sensor gerak dapat berjalan sesuai dengan rencana yang dibuat menggunakan *Flutter*. Hasil dari implementasi dapat

https://doi.org/10.51967/tanesa.v23i1.1285

dilihat pada Gambar 6 dan cuplikan kode yang digunakan dapat dilihat sebagai berikut:

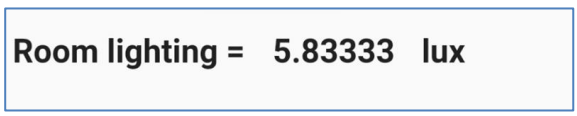
```
Padding(
padding: const EdgeInsets.all(8.0),
child: Text(

snapshot.data.snapshot.value["Motion:"].toStr
ing(),
style: TextStyle(
fontSize: 20, fontWeight: FontWeight.bold),
            ),
           ),
 FlatButton(
 child: Text('RESET'),
 color: Colors.blue,
 textColor: Colors.white,
 onPressed: () async {
 motionReset();
 },
 ),
```

Gambar 6. Aplikasi memantau dan mengendalikan sensor gerak secara *real-time*

4. Memantau Cahaya

Fitur untuk memantau cahaya dapat berjalan sesuai dengan rencana yang dibuat menggunakan *Flutter*. Hasil dari implementasi dapat dilihat pada Gambar 7 dan cuplikan kode yang digunakan dapat dilihat sebagai berikut:

```
Padding(
padding: const EdgeInsets.all(8.0),
child: Text(
snapshot.data.snapshot.value["Light:"].toStri
ng(),
style: TextStyle(
fontSize: 20, fontWeight: FontWeight.bold),
),
),
```

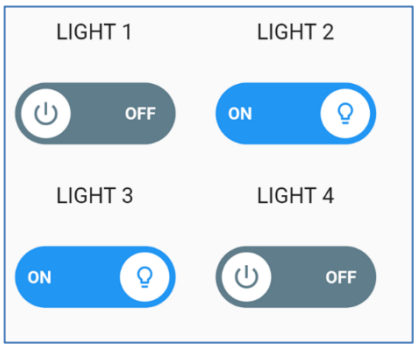


Gambar 7. Aplikasi memantau cahaya secara *real-time*

5. *Memantau dan mengendalikan Modul relay*

Fitur untuk memantau dan mengendalikan Modul *relay* dapat berjalan sesuai dengan rencana yang dibuat menggunakan *Flutter*. Hasil dari implementasi dapat dilihat pada Gambar 4 dan cuplikan kode yang digunakan dapat dilihat sebagai berikut:

```
Column(
 children: [
 Text('LIGHT 1'),
 SizedBox(
 height: 20,
 ),
 Transform.scale(
 scale: 0.8,
 child: LiteRollingSwitch(
 value: isSwitched1,
 onChanged: (value) async {
_handleSwitch1(value);
},
textOn: 'ON',
textOff: 'OFF',
colorOn: Colors.blue,
colorOff: Colors.blueGrey,
iconOn: Icons.lightbulb_outline,
iconOff: Icons.power_settings_new,),),],),
```



Gambar 8. Aplikasi memantau dan mengendalikan lampu secara *real-time*

C. Pengujian Aplikasi

1. Pengujian Alpha

Pengujian *Alpha* adalah pengujian awal yang dilakukan pengembang untuk mengetahui apakah fitur yang sudah diimplementasikan pada aplikasi dapat berjalan sesuai dengan rencana menggunakan prosedur *White Box Testing* dan *Black Box Testing* (Lee-Jayaram et al., 2019).

a. White Box Testing

White Box Testing adalah uji coba yang digunakan pengembang untuk mengetahui apakah seluruh fitur yang telah diimplementasikan di tingkatan integrasi, unit dan sistem telah berjalan dengan baik dan sesuai dengan rencana (Firman et al., 2021). Pengujian W*hite Box Testing* telah digunakan untuk menangani permasalahan pada tombol untuk mengendalikan lampu dalam aplikasi *smart home* yang telah dirancang. Uji coba Telah dilakukan dengan hasil akhir mengubah kode pada tombol lampu dari yang awal menggunakan *FloatingActionButton.extended* menjadi *LiteRollingSwitch* agar struktur kode ketika ingin mengambil data dari *real-time database* dapat berjalan dengan baik.

b. Black Box Testing

Black Box Testing adalah uji coba yang digunakan pengembang untuk mengetahui apakah seluruh fitur yang telah diimplementasikan sudah sesuai dengan rencana dan kebutuhan pengguna dengan cara melakukan test dari sudut pandang pengguna (Firman et al., 2021). Pengujian *Black Box Testing* bertujuan untuk memastikan bahwa setiap fitur dapat berjalan dengan baik dari sudut pandang pengguna. Hasil dari setiap fitur pada penelitian ini dapat dilihat pada Tabel 1.

https://doi.org/10.51967/tanesa.v23i1.1285 © 2022 Politeknik Pertanian Negeri Samarinda

Tabel 1. Hasil *Black Box Testing* pengguna

Fitur	Syarat	Tujuan	Hasil	Status uji
Login	Pengguna sudah memiliki akun dalam aplikasi	Pengguna berhasil masuk ke halaman *home*	Pengguna berhasil masuk ke halaman *home*	Valid
Register	Pengguna belum memiliki akun dalam aplikasi	Pengguna berhasil membuat akun	Pengguna berhasil membuat akun	Valid
Tetap *login*	Ketika pengguna belum pernah *logout* manual	Pengguna tidak perlu melakukan *login* lagi	Pengguna tidak perlu melakukan *login* lagi	Valid
Memantau suhu	Pengguna di halaman *home*	Pengguna dapat melihat nilai suhu ruangan rumah mereka	Pengguna dapat melihat nilai suhu ruangan rumah mereka	Valid
Memantau kelembaban	Pengguna di halaman *home*	Pengguna dapat melihat nilai kelembaban ruangan rumah mereka	Pengguna dapat melihat nilai kelembaban ruangan rumah mereka	Valid
Memantau cahaya	Pengguna di halaman *home*	Pengguna dapat melihat nilai cahaya ruangan rumah mereka	Pengguna dapat melihat nilai cahaya ruangan rumah mereka	Valid
Memantau dan mengendalikan nilai gerak	Pengguna di halaman *home*	Pengguna dapat melihat dan mengendalikan nilai gerakan yang didapat oleh sensor dalam ruangan rumah mereka	Pengguna dapat melihat dan mengendalikan nilai gerakan yang didapat oleh sensor dalam ruangan rumah mereka	Valid
Memantau dan mengendalikan modul *relay*	Pengguna di halaman *home*	Pengguna dapat melihat dan mengendalikan nilai lampu dalam ruangan rumah mereka	Pengguna dapat melihat dan mengendalikan nilai lampu dalam ruangan rumah mereka	Valid

2. Pengujian Beta

Pengujian *Beta* adalah pengujian lanjutan yang ditujukan kepada pengguna tertentu, yang memiliki pengalaman menggunakan peralatan *smart home* untuk membantu memberikan saran dan kritik terhadap proyek *smart home* ini. Proses pengujian ini, membutuhkan *minimal* 3 orang yang dapat memberikan saran dan kritik melalui proses wawancara dan skenario yang diperlukan untuk mengetahui tingkat keakuratan sensor suhu, kelembaban dan cahaya dalam sistem yang sedang dibuat (Lee-Jayaram et al., 2019).

Hasil yang didapat dari wawancara yang dilakukan kepada 3 pengguna (Winston F, Satrio B.Y, Muhammad H), berakhir dengan tanggapan cukup positif terhadap fitur, penggunaan, dan dapat diandalkan. Tanggapan diatas membuktikan bahwa sistem yang telah dibuat sudah cukup memadai untuk digunakan. Namun, ada beberapa saran dan tanggapan yang dapat digunakan untuk mengembangkan aplikasi yaitu: sebaiknya aplikasi dapat bekerja pada perangkat yang menggunakan sistem operasi *IOS*, memiliki penambahan fitur secara dinamis di mana pengguna dapat menambahkan sensor secara langsung dalam aplikasi, serta memperbaiki UI agar lebih rapi dan bekerja di semua ukuran layar smartphone dan OS.

3. Pengujian keseluruhan sistem IOT

a. Uji coba pertama

Setelah dilakukan uji coba pertama, didapatkan hasil menggunakan rumus (1)

$$E = \frac{(X-Y)}{X} x\ 100\% \quad (1)$$

$$X = \frac{t1+t2+t3+t4+t5}{j} \quad (2)$$

$$Y = \frac{t1+t2+t3+t4+t5}{j} \quad (3)$$

Deskripsi:
E = persentase error untuk pengukuran sensor.
X = rata-rata dari Pembanding sensor.
Y = rata-rata dari Sensor sistem.
t = hasil *test* sensor.
j = jumlah total *test* sensor.

Nilai persentase *error* untuk nilai suhu menggunakan rumus sebesar 2.13%, nilai kelembaban sebesar 38.70% dan nilai cahaya sebesar 25%. Hasil yang didapat oleh uji coba pertama kemudian digunakan untuk uji coba kedua.

b. Uji coba kedua

Uji coba kedua mendapatkan hasil berdasarkan dari hasil uji coba pertama dengan menggunakan rumus (1), (2), dan (3). Nilai persentase *error* untuk nilai suhu menggunakan rumus sebesar 0.3%, nilai kelembaban sebesar 13.45% dan nilai cahaya sebesar 10%. Hasil yang didapat oleh uji coba kedua merupakan hasil yang diberikan setelah memberikan konfigurasi kepada *NodeMCU* terhadap pembacaan nilai sensor.

IV. KESIMPULAN

Berdasarkan hasil implementasi yang telah dilakukan berdasarkan metode dan teknologi yang telah direncanakan, dapat dibuktikan bahwa penelitian telah berhasil membuat sebuah sistem yang dapat memantau dan mengendalikan lampu, memantau temperatur, kelembaban dan gerakan pada sistem *smart home* menggunakan aplikasi Android melalui modul *NodeMCU dan cloud storage.* Berdasarkan beta testing

https://doi.org/10.51967/tanesa.v23i1.1285 © 2022 Politeknik Pertanian Negeri Samarinda

yang telah di lakukan, aplikasi dapat berfungsi dengan baik, namun terdapat beberapa kekurangan yang dapat diperbaiki untuk pengembangan ke depannya.

Fitur implementasi *WebSocket* telah berhasil diimplementasikan dengan fitur-fitur: dapat mengendalikan lampu dengan menggunakan aplikasi Android serta sensor dapat mengirim dan menerima data dari *NodeMCU* ke *Firebase* secara *real-time*. Sistem ini dapat memantau sensor pada *NodeMCU* pada aplikasi Android melalui *Firebase*, serta dapat menghubungkan *NodeMCU* dengan *Cloud*.

DAFTAR PUSTAKA

Ada, Lady. (2020). PIR Motion Sensor. *Adafruit Learning System*, 1–28. https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf?timestamp=1585441256

Alzahrani, A., Alshamlani, M., Hsu, W. C., Harish, S., & Daim, T. (2020). Personal transformation: Evaluation of smart home hubs. In *Digital Transformation: Evaluating Emerging Technologies* (pp. 1–31). World Scientific Publishing Co. https://doi.org/10.1142/9789811214639_0006

Ardi, A. (2021). Perancangan Aplikasi Android untuk Kegiatan Tahfizh Alquran Daring menggunakan Platform Firebase. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *8*(1), 14–25. https://doi.org/10.35957/jatisi.v8i1.702

Firman, F., Fauziyah, F., & Komalasari, R. T. (2021). Aplikasi Pengingat Dan Pendataan Kenaikan Golongan Gaji Berbasis Web Menggunakan Metode White Box Testing dan Black Box Testing. *Jurnal Teknologi Informasi*, *7*(1), 50–57. https://doi.org/10.52643/jti.v7i1.1387

Guadagnini, P. H., Rocha, F. S. da, & Barlette, V. E. (2019). Um medidor de luminosidade com módulo sensor integrado e aquisição automática de dados com aplicações didáticas. *Revista Brasileira de Ensino de Física*, *41*(3). https://doi.org/10.1590/1806-9126-rbef-2018-0294

Hakiki, M. I., Darusalam, U., & Nathasia, N. D. (2020). Konfigurasi Arduino IDE Untuk Monitoring Pendeteksi Suhu dan Kelembapan Pada Ruang Data Center Menggunakan Sensor DHT11. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *4*(1), 150. https://doi.org/10.30865/mib.v4i1.1876

Harding, A. (2018, July 9). *Every Disgusting Thing That Happens When You Stop Cleaning Your House*. Every Disgusting Thing That Happens When You Stop Cleaning Your House. https://www.cheatsheet.com/culture/every-disgusting-thing-that-happens-when-you-stop-cleaning-your-house.html/

Kusumah, A. A., & Susila, J. (2020). Perancangan Sistem Monitoring Tegangan Dan Arus Berbasis Arduino Uno Dengan Media Wifi. *Jurnal Nasional Aplikasi Mekatronika, Otomasi Dan Robot Industri (AMORI)*, *1*(2). https://doi.org/10.12962/j27213560.v1i2.7709

Łasocha, W., & Badurowicz, M. (2021). Comparison of WebSocket and HTTP protocol performance. *Journal of Computer Sciences Institute*, *19*, 67–74. https://doi.org/10.35784/jcsi.2452

Lee-Jayaram, J. J., Berg, B. W., Sy, A., & Hara, K. M. (2019). Emergent Themes for Instructional Design: Alpha and Beta Testing During a Faculty Development Course. *Simulation in Healthcare*, *14*(1), 43–50. https://doi.org/10.1097/SIH.0000000000000329

Qader, R. A., & Sahib, N. M. (2021). Intelligent agent services in electronic libraries. *Iraqi Journal of Science*, *62*(4), 1349–1363. https://doi.org/10.24996/ijs.2021.62.4.31

Rahmatulloh, A. (2019). Implementasi formula haversine dan komunikasi data real-time menggunakan websocket di sistem pengawasan warga negara asing. *Klik - kumpulan jurnal ilmu komputer*, *6*(2), 143. https://doi.org/10.20527/klik.v6i2.210

Ruangmom, redaksi. (2021, June 21). *Apa itu Smart Home System? Ini Manfaat dan 7 Rekomendasinya*. Apa Itu Smart Home System? Ini Manfaat Dan 7 Rekomendasinya. https://www.ruangmom.com/smart-home-system.html

Septama, H. D., Yulianti, T., Sulistyono, W. E., Yudamson, A., Suhud, R., & Atmojo, T. (2018). Smart Warehouse: Sistem Pemantauan dan Kontrol Otomatis Suhu serta Kelembaban Gudang. In *Seminar Nasional Inovasi*.

Susilo, D., Sari, C., & Krisna, G. W. (2021). Sistem Kendali Lampu Pada Smart Home Berbasis IOT (Internet of Things). *ELECTRA : Electrical Engineering Articles*, *2*(1), 23. https://doi.org/10.25273/electra.v2i1.10504