

RANCANG BANGUN SISTEM KONTROL *ON-OFF* PADA LAMPU BERBASIS *TELEGRAM*

Fahmi Fuadi Maulana Al Akbar^[1], Muhammad Iqbal Muzaki^[2], Dewi Cahya Febrina^[3], Bambang Purwahyudi^[4]

^{[1],[2],[3],[4]}Bidang Elektronika, Teknik Elektro, Fakultas Teknik, Universitas Bhayangkara Surabaya
Jl. Ahmad Yani No.114, Ketintang, Kec. Gayungan, Kota Surabaya, Jawa Timur 60231

e-mail: ^[1]fahmifuadi561@gmail.com,

^[2]iqbalmzk24@gmail.com, ^[3]fefifebrina1993@gmail.com, ^[4]bmb_pur@yahoo.com

ABSTRACT

Through IoT, we can control the desired device anywhere and anytime only through the internet / WiFi network. This research aims to design an on-off control device for lights without direct contact using IoT technology. In this journal we discuss the processes from start to finish of applying the IoT-based tools. First, the MCU ESP32 node as a microcontroller that has been integrated with WiFi mode which will later be connected to the Smartphone and forwarded to the Relay Module which is tasked with disconnecting and connecting the voltage input to the lamp. This module must be programmed first using Arduino software to connect between the Smartphone device and the ESP32 MCU Node. If it is finished, we connect the ESP32 MCU Node with WiFi, then we can easily turn off and turn on the house lights anywhere through the Telegram application.

Keywords: *Internet of Things, Node MCU ESP32, Arduino Uno, Module Relay*

ABSTRAK

Dengan adanya IoT kita dapat mengontrol alat yang diinginkan dimana pun dan kapan pun hanya dengan melalui jaringan internet / WiFi. Penelitian ini bertujuan untuk merancang alat kontrol on - off pada lampu tanpa berkontak secara langsung menggunakan teknologi IoT. Pada jurnal ini dibahas proses dari awal sampai akhir pengaplikasian alat berbasis IoT. Pertama Node MCU ESP32 sebagai mikrokontroler yang telah terintegrasi dengan WiFi mode yang nantinya akan terkoneksi dengan Smartphone dan diteruskan pada Relay Module yang bertugas untuk memutus dan menghubungkan input tegangan pada Lampu. Modul ini harus diprogram dahulu menggunakan software Arduino untuk menghubungkan antara perangkat Smartphone dan Node MCU ESP32. Jika sudah selesai, kita koneksikan Node MCU ESP32 dengan WiFi, maka dengan mudah kita dapat mematikan dan menyalakan lampu rumah dimana pun melalui aplikasi Telegram.

Kata kunci: *Internet of Things, Node MCU ESP32, Arduino Uno, Module Relay*

1. PENDAHULUAN

Internet of Things (IoT) merupakan suatu konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus. Internet of Things (IoT) bisa dimanfaatkan pada gedung untuk mengendalikan peralatan elektronik seperti lampu ruangan yang dapat dioperasikan dari jarak jauh[1]. Perkembangan teknologi yang bisa dimanfaatkan dari adanya koneksi internet ini bisa mengakses peralatan elektronik seperti lampu ruangan yang dapat dioperasikan dengan cara online melalui mobile, sehingga dapat memudahkan pengguna memantau ataupun mengendalikan lampu kapanpun dan dimanapun pada lokasi yang akan diterapkan teknologi kendali jarak jauh[2].

Lampu biasanya digunakan sebagai alat pencahayaan dan biasanya menggunakan cara manual yang mematikan dan menghidupkan secara langsung. Cara seperti ini akan diubah dengan memakai karakteristik pada *speech recognition* atau pengembangan sistem terhadap Smart Phone untuk sistem kendali jarak jauh pada lampu dengan memakai internet modul Node MCU berbasis Sistem On Chip ESP8266 yang merupakan bentuk penerapan *Internet of Things* (IoT)[3].

Teknologi IoT dapat diterapkan untuk mewujudkan konsep terbaru yang terkait dengan kemajuan *smart home* yang akan memberikan kenyamanan bagi para pemiliknya. Selain itu peneliti memiliki tujuan untuk membantu mengembangkan keamanan bagi para pemiliknya karena beberapa alat di rumah telah terkendali secara otomatis, salah satunya pada lampu rumah[4].

2. TEORI

Sebagai landasan teori berikut di bawah ini adalah pembahasan teori dari perangkat yang akan digunakan dalam rancangan sistem kontrol *on-off* pada lampu berbasis telegram.

2.1 Smart Home

Smart home adalah sebuah sistem kendali bagi pemilik rumah yang dapat mengatur dan mengendalikan peralatan elektronik yang sudah ada didalam rumah dan perangkat ini juga bisa di kendalikan dengan jarak jangkauan yang jauh dan dapat diaplikasikan dengan smartphone atau android. Pengaplikasian terhadap smart phone atau android termasuk salah satu cara pengaplikasian *Internet of Things* (IoT). Dengan menggunakan smart home pada rumah, sehingga akan lebih efektif dalam pengguna dan sesuai dengan penggunaan untuk menghemat energi listrik yang di pakai pengguna dan dapat juga memberikan sebuah kenyamanan yang lebih baik lagi dan terjamin kualitasnya[5].

2.2 Internet of Things (IoT)

Internet of Things atau IoT memiliki istilah untuk pemakaian pada internet, dalam mengambil perhitungan yang bersifat mobile kemudian menghubungkan kedalam kehidupan sehari hari. Pada model C-IoT dalam bentuk sederhananya terdiri dari Sensing, Gateway, dan Services. penginderaan (sensing) akan memasukan apa yang dianggap begitu penting, Gateway juga akan menambah kecerdasan dan konektifitas untuk sebuah sistem yang akan di ambil baik dari tingkatan sistem atau penyampaian informasinya ke Cloud level, sedangkan Services akan mengumpulkan informasi dan menganalisa, dan dapat mengembangkan wawasan untuk membantu meningkatkan suatu kualitas hidup atau improve business operation[6].

2.3 Node MCU ESP32

ESP32 adalah mikrokontroler yang dikenalkan oleh Espressif System dan merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dan Bluetooth dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi Internet of Things. ESP32 memiliki fitur yang cukup lengkap karena mendukung input/output Analog dan Digital, PWM, SPI, I2C, dll[7].



Gambar 1. Node MCU ESP 32

Hardware yang baik seperti ESP32 akan lebih user friendly jika dapat diprogram lebih dari satu software pendukung (Cross Platform). ESP32 mendukung beberapa environment pemrograman. Beberapa environment pemrograman yang umum digunakan adalah:

1. Arduino IDE
2. PlatformIO IDE (VS Code)
3. LUA
4. MicroPython
5. Espressif IDF (IoT Development Framework)
6. JavaScript

Arduino IDE merupakan environment yang paling sering digunakan, tapi tidak salahnya menggunakan environment yang berbeda sehingga memiliki wawasan yang lebih baik lagi.

2.4 Arduino IDE

IDE adalah singkatan dari Integrated Development Environment. IDE yang termasuk program yang digunakan untuk bisa membuat program pada ESP32 Node MCU. Pembuatan program bisa menggunakan Software Arduino IDE disebut sebagai sketch. Sketch ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi.ino pada Software Arduino IDE, terdapat semacam pesan box berwarna hitam yang fungsinya menampilkan status, seperti pesan error, compile, dan upload pada program. Dibagian bawah paling kanan Software Arduino IDE, menunjukkan board yang terkonfigurasi beserta COM Ports yang digunakan berikut fungsi dari Verify/Compile dan Upload di Arduino IDE.

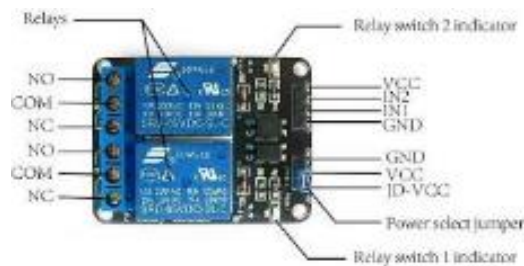
2.5 Modul Relay

Spesifikasi:

1. Modul ini menggunakan relay asli berkualitas tipe Normally Open (NO) dengan maximum load AC 250V/10A, DC 30V/10A
2. Memakai SMD Optocoupler isolation, yang berkinerja stabil dengan arus pemicu (trigger current) hanya sebesar 5mA
3. Tegangan sinyal pemicu sebesar 5V DC
4. Dapat disetting untuk mendeteksi high atau low dengan mengubah jumper
5. Dirancang dengan toleransi keamanan, bahkan jika arus pemicu putus, relay tidak akan bekerja
6. Dilengkapi lampu indikator Power (hijau) dan Status Relay (merah)
7. Mudah dipasang, menggunakan terminal untuk pemasangan kabel.
8. Ukuran: 50x41x18.5mm
9. Dilengkapi 4 lobang baut berdiameter 3.1 mm berjarak 44.5mm x 35.5mm

Interface pemicu:

1. DC+: power +5V DC
2. DC-: power -5V DC
3. IN1: sinyal low atau high pada channel 1 (stel jumper)
4. IN2: sinyal low atau high pada channel 2 (stel jumper)
5. Interface output relay:
6. NO1: normally open channel 1 (relay tidak hidup sampai ada sinyal baru hidup)
7. COM1 : common interface channel 1
8. NC1: normally close channel 1 (relay hidup sampai ada sinyal baru mati)
9. NO2: normally open channel 2 (relay tidak hidup sampai ada sinyal baru hidup)
10. COM2 : common interface channel 2
11. NC2: normally close channel 2 (relay hidup sampai ada sinyal baru mati)



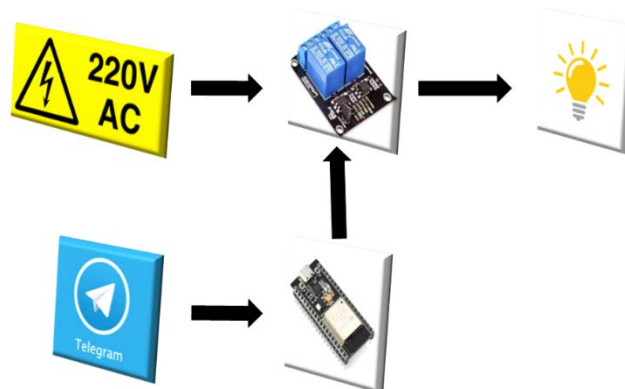
Gambar 2. Module Relay 2 Chanel

3. METODE

Dalam menghasilkan suatu sistem kontrol *on-off* pada rumah tinggal dengan menggunakan kontroler ESP32 berbasis IoT, ada beberapa tahapan yaitu meliputi pengumpulan data, perancangan perangkat keras, perancangan perangkat lunak, pengujian sistem untuk mendapatkan data-data yang berhubungan dengan kerja sistem kontrol.

3.1 Perancangan Hardware

Perancangan *hardware* ini bertujuan untuk mengesekusi perintah yang sudah dikirim lewat Telegram. Perintah yang dikirim lewat *telegram* akan diterima oleh Node MCU ESP32 kemudian dikirim ke Relay, pada Relay ini lah yang akan menghantar dan memutus aliran listrik yang tersambung kepada perangkat yaitu ke lampu.

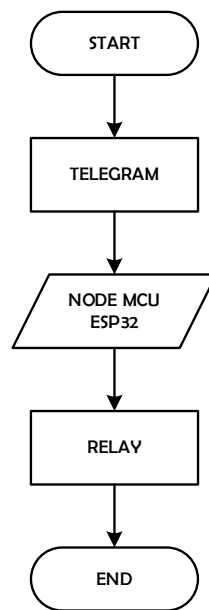


Gambar 3. Blok Diagram Rancangan Hardware

Gambar di atas ialah blok diagram untuk perangkat keras (*hardware*) yang digunakan dalam penelitian ini.

3.2 Perancangan Software

Perancangan *software* merupakan suatu media pengendali pemograman dengan suatu perintah program yang nantinya perintah program itu berfungsi sebagai pengendali suatu rangkaian yang telah dibuat yang meliputi; pemograman modul ESP32, program arduino ide berisi perintah berupa input (coding) yang berurutan kemudian diproses untuk diupload pada module wifi ESP32 dan menghasilkan keluaran *actuator* tampilan pada aplikasi *telegram* yang ada di smartphone.



Gambar 4. Flowchart Rancangan Software




```
#include <WiFi.h>
#include <UniversalTelegramBot.h>
#include <WiFiClientSecure.h>
const char* WIFI_SSID = "Wifi.Id";
const char* WIFI_PASSWORD = "12345678a";
const char* BOT_TOKEN = "6540063575:AAH_f3t6kkygqG0G1R01n67QmP37ye-j2SY";
String chatId = "1793493581";
WiFiClientSecure client;
UniversalTelegramBot bot(BOT_TOKEN, client);
const int relay1 = 25; // Pin yang digunakan untuk mengendalikan lampu
void setup() {
  Serial.begin(115200);
  pinMode(relay1, OUTPUT);
  // Menghubungkan ke WiFi
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
    digitalWrite(relay1, HIGH);
  }
  Serial.println("Connected to WiFi");
  // Menginisialisasi pin untuk lampu
  digitalWrite(relay1, LOW);
  // Memulai koneksi ke Telegram
  client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Ini opsional, bergantung pada
  versi pustaka yang digunakan
}
void loop() {
  int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  for (int i = 0; i < numNewMessages; i++) {
    chatId = bot.messages[i].chat_id; // Menyimpan chat_id dari pesan terbaru
    String text = bot.messages[i].text;
```

```
Serial.println(text);
if (text == "Say") {
  bot.sendMessage(chatId, "ApaKabar!");
  delay(100);
  bot.sendMessage(chatId, "Apa yang bisa dibantu!");
} else if (text == "Nyalakan lampu") {
  digitalWrite(relay1, LOW);
  bot.sendMessage(chatId, "Lampu telah dihidupkan!");
} else if (text == "Matikan lampu") {
  digitalWrite(relay1, HIGH);
  bot.sendMessage(chatId, "Lampu telah dimatikan!");
}
}
delay(100);
}
```

3.3 Perancangan Kerangka dan Mekanisme

Pada perancangan mekanisme akan dijelaskan beberapa komponen yang digunakan untuk mengontrol lampu menggunakan aplikasi *telegram*. Komponen dan bahan yang diperlukan terdapat pada tabel 1:

Tabel 1. Komponen yang Digunakan sebagai Kontrol Lampu

Nama Komponen	Gambar
Node MCU ESP32	
Module Relay	
Kabel Jumper	

Lampu	
Fiting Lampu	

4. HASIL DAN PEMBAHASAN

Data hasil pengujian sistem kontrol *on-off* pada lampu menggunakan *telegram* dapat dilihat sebagai berikut:

4.1 Hasil Perakitan Alat

Pada hasil perakitan alat dari komponen-komponen yang telah disebutkan di atas telah dirangkai menjadi satu secara keseluruhan. Berikut hasilnya.

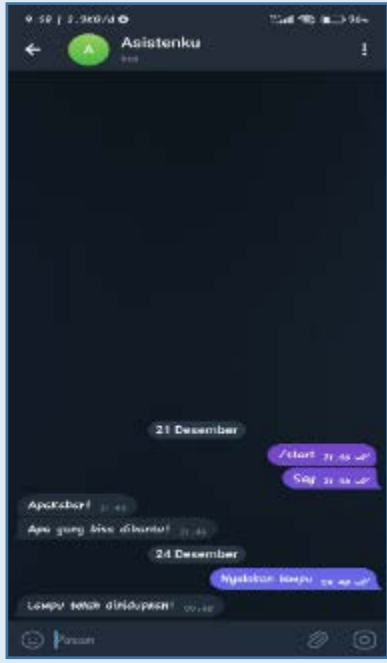
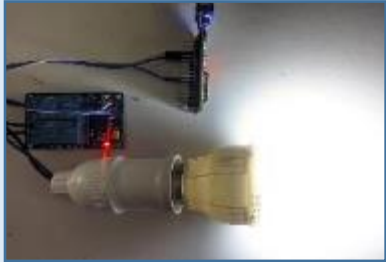
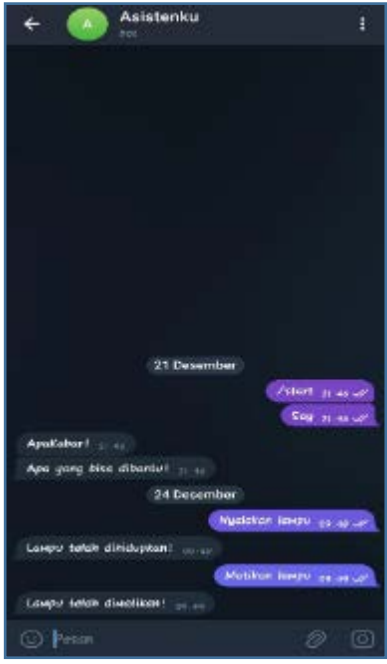
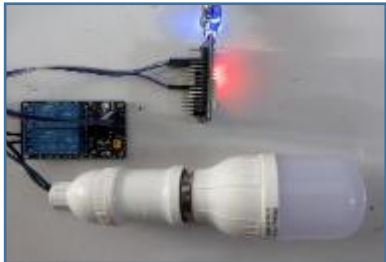


Gambar 5. Hasil Perakitan Alat

4.2 Hasil Pengujian

Pengujian fungsi diperlukan untuk menunjukkan keberhasilan rancang bangun sistem kontrol *on-off* pada lampu menggunakan *telegram*, hasil pengujian dapat dilihat pada tabel 2.

Tabel 2. Hasil Pengujian

Pengujian	Telegram	Alat Pengujian
Lampu ON	 A screenshot of a Telegram chat interface. The contact is named 'Asistenku'. The chat history shows a date separator for '21 Desember' followed by a purple command message: '/start'. Below it, a grey response message says 'Apakah?'. Another grey message asks 'Apa yang bisa dibantu?'. A second date separator for '24 Desember' is followed by a purple command message: 'Nyalakan lampu'. A grey response message says 'Lampu telah dinyalakan!'.	 A photograph of the experimental setup. A white LED lamp is illuminated, casting a bright white light. In the background, a blue printed circuit board (PCB) with various electronic components is visible, connected to the lamp.
Lampu OFF	 A screenshot of a Telegram chat interface, similar to the one above. It shows the same chat history up to the 'Lampu telah dinyalakan!' message. A purple command message follows: 'Matikan lampu'. A grey response message says 'Lampu telah dimatikan!'.	 A photograph of the experimental setup. The white LED lamp is now unlit and appears as a white object. The blue PCB and other components in the background are visible, but the lamp is not emitting light.

5. SIMPULAN

Sistem *on-off* pada lampu menggunakan Telegram adalah solusi yang inovatif dan dapat memberikan berbagai manfaat. Dengan mengintegrasikan Telegram sebagai platform pengendalian, pengguna dapat mengontrol lampu dari jarak jauh menggunakan aplikasi pesan instan.

Berikut adalah beberapa kesimpulan terkait dengan sistem ini:

1. Kemudahan Penggunaan:
Sistem ini memberikan kemudahan penggunaan karena memanfaatkan antarmuka yang sudah akrab bagi banyak orang, yaitu aplikasi Telegram. Pengguna dapat dengan mudah mengontrol lampu tanpa perlu memahami antarmuka pengendalian khusus.
2. Kontrol Jarak Jauh:
Telegram memungkinkan kontrol jarak jauh, memungkinkan pengguna untuk mengaktifkan atau mematikan lampu dari mana saja yang terhubung dengan internet. Ini memberikan fleksibilitas dan kenyamanan bagi pengguna.
3. Integrasi dengan Sistem Komunikasi:
Telegram tidak hanya berfungsi sebagai pengontrol lampu, tetapi juga sebagai alat komunikasi. Pengguna dapat menerima pemberitahuan atau mengonfirmasi status lampu melalui pesan di Telegram.

Dengan menggabungkan kenyamanan Telegram dengan kendali lampu, sistem ini dapat memberikan solusi yang cerdas dan efisien.

REFERENSI

- [1] R. Rizky, Z. Hakim, A. M. Yunita, and N. N. Wardah, "Implementasi Teknologi IoT (Internet of Think) pada Rumah Pintar Berbasis Mikrokontroler ESP 8266," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 278–281, 2020, doi: 10.36294/jurti.v4i2.1452.
- [2] M. Syahputra Novelan, Z. Syahputra, and P. H. Putra, "Sistem Kendali Lampu Menggunakan NodeMCU dan Mysql Berbasis IOT (Internet Of Things)," *J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 1, pp. 117–121, 2020, [Online]. Available: <https://doi.org/10.30743/infotekjar.v5i1.2976>
- [3] V. 10 N. 3 M. 2020 I. : 2407-3903 Nugroho, Agung, Dzulkha, Alfatan, "SIGMA - Jurnal Teknologi Pelita Bangsa SIGMA - Jurnal Teknologi Pelita Bangsa," *SIGMA - J. Teknol. Pelita Bangsa 167*, vol. 10, no. September, pp. 167–172, 2020.
- [4] M. Usman, "Fakultas ilmu komputer," *J. Sist. Inf. dan Telemat.*, vol. 10, p. 2, 2019.
- [5] D. Susilo, C. Sari, and G. W. Krisna, "Sistem Kendali Lampu Pada Smart Home Berbasis IOT (Internet of Things)," *ELECTRA Electr. Eng. Artic.*, vol. 2, no. 1, p. 23, 2021, doi: 10.25273/electra.v2i1.10504.
- [6] Y. Efendi, "Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile," *J. Ilm. Ilmu Komput.*, vol. 4, no. 2, pp. 21–27, 2018, doi: 10.35329/jiik.v4i2.41.
- [7] S. Supono, T. Rijanto, and J. W. Leksono, "Perancangan Sistem Kendali dan Monitoring Tegangan Motor 3 Fasa Berbasis Internet of Things Menggunakan Aplikasi Blynk," *Indones. J. Eng. Technol.*, vol. 3, no. 1, pp. 38–45, 2020, doi: 10.26740/inajet.v3n1.p38-45.