

IoT-Based Real-Time Security System Implementation at Kaka Game Center Internet Café

Muh Alvian Bagus Prastya¹, Joni Maulindar², Eko Purwanto³

^{1,2,3} Teknik Informatika, Fakultas Ilmu Komputer, Universitas Duta Bangsa, Surakarta, 57155, Indonesia

Informasi Artikel

Diterima : 17 Juli 2025
Revisi : 30 Juli 2025
Publikasi : 30 September 2025

Kata Kunci:

Internet of Things
Sistem Keamanan
ESP32
Warnet
ESP-32CAM

ABSTRAK

Keamanan ruang publik seperti warnet (warnet) menjadi perhatian penting, terutama saat lokasi dalam kondisi tidak beroperasi. Penelitian ini mengembangkan sistem keamanan real-time berbasis Internet of Things (IoT) dengan mengintegrasikan mikrokontroler ESP32 dan ESP32-CAM untuk meningkatkan efektivitas pengawasan di warnet. Sistem memanfaatkan sensor PIR untuk mendeteksi gerakan, kemudian mengirimkan notifikasi otomatis ke aplikasi Telegram, sekaligus memicu pengambilan gambar melalui ESP32-CAM. Selain deteksi otomatis, pengguna juga dapat mengontrol sistem secara manual melalui Telegram, seperti menyalakan buzzer, mengambil snapshot, atau mengakses streaming video secara real-time menggunakan ngrok. Sistem dikembangkan menggunakan metode Waterfall yang mencakup analisis kebutuhan, desain, implementasi, dan pengujian. Hasil pengujian menunjukkan bahwa seluruh fitur sistem bekerja dengan baik, responsif, dan mendukung pemantauan jarak jauh secara efisien. Dengan biaya terjangkau dan integrasi teknologi yang optimal, sistem ini layak digunakan sebagai alternatif sistem keamanan modern di lingkungan usaha kecil. Kontribusi ilmiah dari penelitian ini adalah menghadirkan solusi integratif dengan fitur monitoring, notifikasi real-time, dan kendali jarak jauh berbasis Telegram yang dapat diadaptasi secara luas pada pengembangan sistem keamanan IoT untuk berbagai lingkungan usaha skala kecil.

ABSTRACT

Security in public spaces such as internet cafés (warnet) is a major concern, especially during non-operational hours. This study develops a real-time security system based on the Internet of Things (IoT) by integrating the ESP32 and ESP32-CAM microcontrollers to enhance surveillance effectiveness in warnet environments. The system uses a PIR sensor to detect motion, then automatically sends a notification to the Telegram application and triggers image capture via the ESP32-CAM. In addition to automatic detection, users can also manually control the system via Telegram, such as activating the buzzer, requesting snapshots, or accessing real-time video streaming using ngrok. The system was developed using the Waterfall method, which includes requirement analysis, design, implementation, and testing. The results show that all system features operate properly, are responsive, and support efficient remote monitoring. With its low cost and optimal technology integration, the system is a viable alternative to conventional security systems for small-scale businesses. The scientific contribution of this research lies in the development of an integrative solution combining real-time monitoring, instant notification, and remote control via Telegram, which can be widely adopted to enhance IoT-based security systems for various small-scale business environments.

This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



*Penulis Koresponden

Email: alvian.bagus2@gmail.com

Cara sitasi IEEE::

M.A.B. Prastya, J. Maulindar, & E. Purwanto, "IoT-Based Real-Time Security System Implementation at Kaka Game Center Internet Café," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 3, pp. 1096-1108, September 2025, doi: 10.30811/jaise.v5i3.7327

1. PENDAHULUAN

Perkembangan teknologi Internet of Things (IoT) telah membawa dampak signifikan dalam berbagai aspek kehidupan, termasuk dalam bidang keamanan [1]. IoT memungkinkan integrasi antara perangkat keras dan perangkat lunak untuk menciptakan sistem yang dapat memantau, mendeteksi, dan merespons kejadian secara real-time [2]. Dalam konteks keamanan, penerapan IoT dapat meningkatkan efisiensi dan efektivitas sistem pengawasan [3], terutama pada area yang memerlukan pemantauan terus-menerus seperti warung internet (warnet).

Warnet merupakan penyedia layanan sewa komputer beserta internet bagi masyarakat. Warnet bisa digunakan sebagai tempat bekerja maupun tempat untuk bermain bagi para penggunanya [4]. Sistem keamanan tradisional pada warnet umumnya masih mengandalkan pengawasan manual atau penggunaan Closed-Circuit Television (CCTV) yang memerlukan pemantauan aktif oleh operator. Hal ini kurang efektif dalam memberikan respons cepat terhadap kejadian yang tidak diinginkan, seperti pencurian atau tindakan kriminal lainnya [5]. Selain itu, sistem konvensional belum mendukung notifikasi real-time dan akses jarak jauh secara efisien.

Berbagai penelitian sebelumnya telah mengembangkan sistem keamanan ruangan berbasis Internet of Things (IoT) yang mengintegrasikan sensor PIR, alarm buzzer, dan layanan WhatsApp melalui Twilio API untuk memberikan notifikasi real-time saat terdeteksi gerakan mencurigakan. Penelitian ini menitikberatkan pada efisiensi notifikasi serta kemudahan penerapan sistem di berbagai lingkungan seperti kost, rumah, dan kantor. Dengan waktu respons rata-rata 4 detik dan akurasi tinggi, sistem ini terbukti efektif memberikan peringatan dini serta fleksibel dalam penggunaan daya dan konfigurasi perangkat [6].

Selanjutnya, penelitian oleh Luthfi Cakrayuda dan rekan (2025) mengembangkan sistem keamanan cerdas bernama SiCEMOT yang mengintegrasikan ESP32-CAM, sensor gerak PIR, dan notifikasi Telegram untuk mendeteksi gerakan dan mengirimkan gambar secara real-time di lingkungan asrama. Penelitian ini menekankan pentingnya respons cepat, efisiensi energi melalui mode deep sleep, serta ketepatan deteksi objek manusia dengan akurasi hingga 95% dalam pencahayaan normal. [7].

Kemudian, Fajar Abdurrokhman dan Adiyanto (2021) merancang sistem keamanan berbasis *Internet of Things* (IoT) khusus untuk lingkungan warnet Tara Komputer, dengan memanfaatkan ESP32-CAM, sensor infrared E18-D80NK, dan aplikasi Telegram sebagai media pemantauan. Penelitian ini menitikberatkan pada keamanan warnet yang ditinggal dalam kondisi tutup, serta memberikan solusi hemat biaya dibandingkan sistem CCTV konvensional [8].

Berbeda dengan penelitian-penelitian sebelumnya, penelitian ini secara khusus difokuskan pada sistem keamanan real-time berbasis Internet of Things (IoT) yang diterapkan di lingkungan warnet, dengan menggabungkan fitur-fitur komprehensif yang belum sepenuhnya diakomodasi oleh penelitian terdahulu. Sistem yang dikembangkan tidak hanya mencakup deteksi gerakan dan notifikasi seperti pada studi Silvia Adetia dkk. maupun Luthfi Cakrayuda dkk., tetapi juga menghadirkan integrasi antara ESP32 dan ESP32-CAM, pengambilan gambar (snapshot), streaming video real-time, serta kontrol perangkat (buzzer dan LED) secara langsung melalui perintah di aplikasi Telegram.

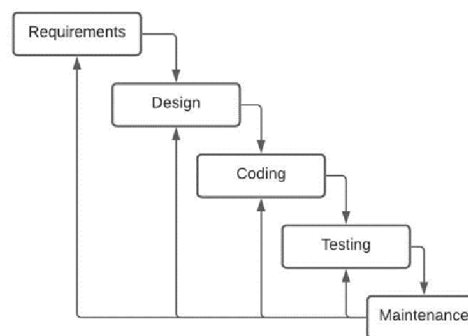
Penelitian ini bertujuan menghasilkan solusi keamanan IoT yang aplikatif, ekonomis, dan fleksibel untuk kebutuhan usaha kecil seperti warnet. Dengan memanfaatkan perangkat terjangkau dan integrasi penuh dengan Telegram sebagai media notifikasi dan kontrol, sistem ini diharapkan dapat meningkatkan efisiensi pengawasan, memberikan rasa aman bagi pemilik, serta menjadi alternatif modern bagi sistem keamanan konvensional yang umumnya bersifat pasif, mahal, dan kurang adaptif terhadap kendali jarak jauh.

Namun, berdasarkan hasil studi literatur, mayoritas penelitian sebelumnya lebih menekankan pada notifikasi gerakan sederhana atau penggunaan satu jenis sensor tanpa integrasi komprehensif dengan fitur kendali jarak jauh, dokumentasi visual (snapshot), serta layanan streaming real-time yang mudah diakses. Penelitian ini mengisi celah tersebut dengan mengembangkan sistem keamanan IoT yang tidak hanya memberikan notifikasi otomatis, tetapi juga mengintegrasikan kontrol perangkat secara real-time, bukti visual,

serta pemantauan jarak jauh berbasis Telegram, sehingga dapat meningkatkan efektivitas dan fleksibilitas sistem keamanan pada lingkungan usaha kecil seperti warnet.

2. METODE

Penelitian ini menggunakan metode pengembangan sistem perangkat lunak model Waterfall, yang dikenal sebagai pendekatan sekuensial dalam proses rekayasa perangkat lunak. Model ini terdiri dari lima tahapan utama, yaitu analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan [21]. Pendekatan ini dipilih karena memberikan alur kerja yang terstruktur dan cocok untuk proyek dengan kebutuhan sistem yang telah didefinisikan secara jelas sejak awal. Berikut ilustrasi dari model pengembangan sistem metode waterfall.



Gambar 1. Metode Waterfall
(Sumber researchgate.net)

Adapun tahapan-tahapan pengembangan sistem dengan metode waterfall adalah sebagai berikut.

Tahap awal dilakukan dengan menganalisis kebutuhan sistem, baik dari sisi perangkat keras maupun fitur perangkat lunak yang diperlukan. Dalam penelitian ini, kebutuhan perangkat keras meliputi mikrokontroler ESP32 sebagai pusat kendali, sensor PIR untuk mendeteksi gerakan, ESP32-CAM sebagai kamera untuk pengambilan gambar dan streaming video, buzzer sebagai indikator suara alarm, serta LED sebagai indikator visual. Dari sisi perangkat lunak, digunakan Arduino IDE atau PlatformIO untuk pemrograman, Telegram Bot untuk notifikasi dan kontrol jarak jauh, serta ngrok untuk tunneling streaming video agar dapat diakses dari mana saja. Kebutuhan fungsional sistem mencakup deteksi gerakan otomatis, notifikasi real-time ke Telegram, kontrol buzzer dan LED dari Telegram, pengiriman foto ke Telegram, streaming video real-time, serta pencatatan log aktivitas pengguna.

Setelah kebutuhan teridentifikasi, dilakukan desain sistem yang meliputi perancangan skematik rangkaian elektronik, blok diagram sistem, dan algoritma logika kerja sistem. Algoritma utama sistem dapat dijelaskan sebagai berikut: sensor PIR membaca adanya gerakan di area yang dipantau; jika terdeteksi gerakan, sistem mengirim notifikasi ke Telegram, mengambil foto menggunakan ESP32-CAM, dan mengirim foto tersebut ke Telegram. Selain itu, pengguna dapat mengontrol buzzer dan LED secara manual melalui perintah di Telegram. Sistem juga menyediakan akses streaming video real-time yang dapat diakses dari mana saja melalui ngrok, serta mencatat log aktivitas akses streaming dan pengambilan foto ke Telegram.

Setelah rancangan selesai, penerapan sistem dimulai dengan menggabungkan seluruh bagian perangkat keras dan melakukan pemrograman pada ESP32 serta ESP32-CAM menggunakan Arduino IDE atau PlatformIO. Di Telegram, sebuah bot dibangun yang mencakup fitur notifikasi otomatis saat terdeteksi gerakan, perintah untuk mengontrol buzzer dan LED, perintah untuk mengambil dan mengirim foto, serta log aktivitas yang mencatat siapa saja yang mengakses streaming atau meminta snapshot. Semua aktivitas penting dicatat dan dikirimkan ke Telegram untuk memudahkan pemantauan.

Tahap berikutnya melibatkan evaluasi sistem, yang terdiri dari pengujian masing-masing komponen secara terpisah, pengujian integrasi di antara komponen, serta pengujian keseluruhan sistem pada kondisi lapangan. Pengujian dilakukan untuk memastikan bahwa setiap fitur bekerja sesuai tujuan, termasuk respons sistem terhadap deteksi gerakan, keakuratan notifikasi, efektivitas kontrol perangkat, kualitas pengiriman foto, kestabilan streaming video, serta validitas log aktivitas. Data yang diperoleh dari pengujian, seperti waktu respons, keakuratan deteksi gerakan, dan kualitas streaming, dianalisis untuk mengevaluasi performa sistem serta efektivitas pengendalian dan pemantauan jarak jauh melalui Telegram.

Dalam tahapan terakhir, perawatan dilaksanakan guna menjamin sistem berfungsi dengan baik dalam jangka waktu yang panjang. Hal ini meliputi kalibrasi ulang sensor, pembaruan firmware pada mikrokontroler jika diperlukan, serta evaluasi daya tahan komponen. Secara keseluruhan, penelitian ini didukung oleh konsep-konsep rekayasa perangkat lunak modern dan pengembangan sistem IoT yang sesuai dengan perkembangan teknologi keamanan digital.

3. HASIL DAN PEMBAHASAN

3.1. Analisis Kebutuhan

Tahap analisis kebutuhan bertujuan untuk merumuskan spesifikasi sistem secara detail agar sistem keamanan real-time di warnet dapat bekerja secara otomatis dan sesuai dengan sasaran yang telah ditetapkan. Dari sisi perangkat keras, sistem ini mengandalkan sejumlah komponen utama, di antaranya mikrokontroler ESP32 yang berperan sebagai pusat kendali, sensor PIR (Passive Infrared Receiver) untuk mendeteksi gerakan manusia di area yang dipantau, serta modul ESP32-CAM yang berfungsi sebagai kamera untuk pengambilan gambar dan streaming video real-time. Selain itu, sistem juga dilengkapi dengan buzzer sebagai indikator suara alarm, LED sebagai indikator visual, serta modul Wi-Fi untuk konektivitas internet.

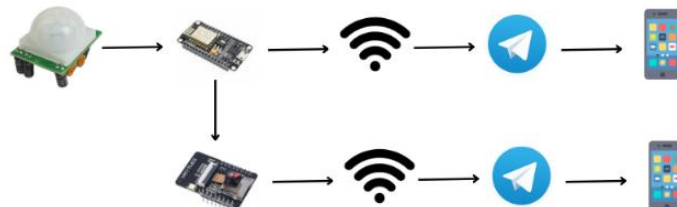
Pada sisi perangkat lunak, digunakan Arduino IDE atau PlatformIO sebagai alat bantu untuk menulis dan mengunggah program ke mikrokontroler, serta Telegram Bot yang berfungsi sebagai media notifikasi dan kontrol jarak jauh berbasis smartphone. Kebutuhan sistem secara fungsional mencakup kemampuan mendeteksi gerakan secara otomatis, mengirimkan notifikasi real-time ke Telegram saat terdeteksi gerakan, mengontrol buzzer dan ESP32 dari Telegram, mengambil foto serta mengirimkannya ke Telegram, menyediakan akses streaming video real-time yang dapat diakses dari mana saja melalui ngrok, serta mencatat dan mengirimkan log aktivitas ke Telegram. Sedangkan untuk kebutuhan non-fungsional, sistem dirancang agar mudah digunakan, memiliki koneksi jaringan yang stabil, serta mampu merespons perubahan kondisi secara cepat dan efisien.

3.2. Desain Sistem

Pada tahapan ini, dijelaskan rancangan sistem mulai dari blok diagram yang menunjukkan hubungan antar komponen utama, flowchart yang mengilustrasikan alur logika kerja sistem, skema rangkaian elektronik yang merepresentasikan koneksi antar perangkat keras, hingga tampilan antarmuka pengguna yang digunakan untuk monitoring dan pengendalian melalui aplikasi. Blok diagram sistem menggambarkan interaksi antara ESP32, ESP32-CAM, sensor PIR, buzzer, LED, serta modul Wi-Fi yang terhubung ke jaringan internet. Komponen-komponen ini saling bekerja sama untuk mendeteksi gerakan, mengambil foto, mengaktifkan alarm, serta mengirimkan notifikasi dan foto ke Telegram Bot sebagai pusat monitoring dan pengendalian jarak jauh.

3.2.1. Diagram Blok Sistem

Gambaran umum hubungan antar komponen ditunjukkan melalui diagram blok di bawah.



Gambar 2. Blok Diagram

Sistem dimulai dengan sensor PIR (Passive Infrared Receiver) yang bertugas mendeteksi gerakan manusia di area yang dipantau. Data hasil deteksi dari sensor PIR kemudian dikirimkan ke mikrokontroler ESP32, yaitu perangkat yang dilengkapi konektivitas Wi-Fi. ESP32 memproses sinyal tersebut dan melakukan beberapa tindakan, seperti mengaktifkan buzzer atau LED sebagai indikator lokal, serta mengirimkan notifikasi ke Telegram Bot melalui jaringan internet. Selain itu, ESP32 juga dapat memberikan perintah ke modul ESP32-CAM untuk mengambil foto atau memulai streaming video.

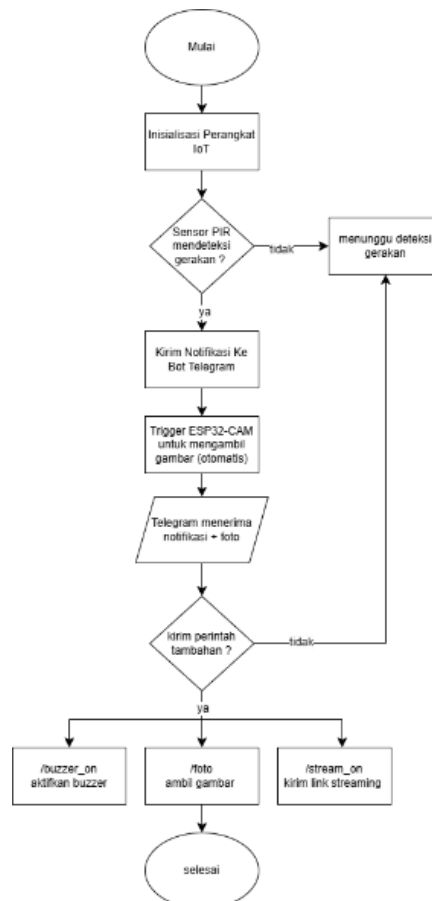
ESP32-CAM menerima perintah dari ESP32, lalu menggunakan kameranya untuk mengambil gambar atau menyediakan streaming video real-time. Data gambar dan video yang dihasilkan kemudian dikirimkan kembali ke Telegram Bot melalui jaringan Wi-Fi, sehingga pengguna dapat menerima notifikasi, melihat foto, atau memantau kondisi warnet secara langsung melalui aplikasi Telegram di smartphone. Telegram Bot berperan sebagai antarmuka utama yang menampung dan mengelola informasi kejadian secara real-time, serta menyediakan log aktivitas sistem.

Dengan sistem ini, pengguna dapat memantau kondisi keamanan warnet dari jarak jauh secara langsung, mengontrol perangkat seperti buzzer & ESP32 melalui perintah di Telegram, serta

menerima bukti visual berupa foto atau streaming video saat terdeteksi gerakan, sehingga keamanan lingkungan dapat terjaga secara lebih efektif dan responsif.

3.2.2. Flowchart Sistem

Alur logika kerja sistem dapat digambarkan melalui flowchart berikut.



Gambar 3. Flowchart

Flowchart di atas menggambarkan alur kerja sistem keamanan real-time berbasis IoT menggunakan ESP32 dan ESP32-CAM yang diterapkan di lingkungan warnet. Sistem dimulai saat perangkat diaktifkan, di mana ESP32 melakukan proses inisialisasi terhadap koneksi WiFi, sensor PIR, buzzer, LED, serta koneksi dengan bot Telegram.

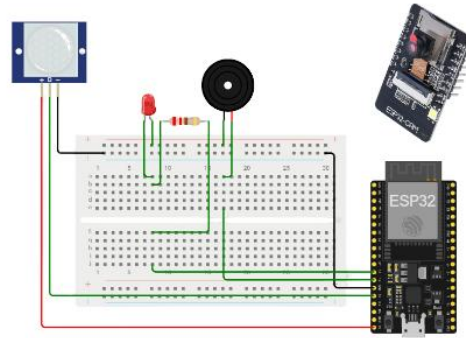
Setelah inisialisasi selesai, sistem masuk ke mode pemantauan gerakan. Sensor PIR secara terus-menerus mendeteksi apakah terdapat gerakan di area pemantauan. Jika tidak ada gerakan, sistem akan tetap dalam kondisi siaga. Namun jika terdeteksi adanya gerakan, maka ESP32 akan langsung mengirimkan notifikasi otomatis ke bot Telegram, sekaligus mentrigger ESP32-CAM untuk mengambil gambar secara otomatis dan mengirimkannya ke Telegram.

Setelah notifikasi dan gambar diterima oleh pengguna melalui Telegram, sistem memberikan kemungkinan interaksi lanjutan. Pengguna dapat mengirim perintah tambahan melalui bot Telegram, seperti: /buzzer_on untuk mengaktifkan alarm buzzer sebagai respons cepat, /foto untuk mengambil gambar secara manual kapan pun diperlukan, /stream_on untuk mengirimkan link streaming dari esp32-cam melalui layanan ngrok yang sudah siaga untuk di pakai

Jika tidak ada perintah tambahan, sistem akan kembali ke kondisi awal dan melanjutkan pemantauan gerakan, sehingga menciptakan loop sistem yang real-time dan berkesinambungan hingga perangkat dimatikan.

3.2.3. Skema Rangkaian

Ilustrasi hubungan perangkat keras ditampilkan melalui skema rangkaian pada gambar berikut.



Gambar 4. Skema Rangkaian

Gambar di atas menunjukkan skema rangkaian dari sistem keamanan real-time berbasis Internet of Things (IoT) yang menggunakan mikrokontroler ESP32 sebagai pusat kendali. Sistem ini terdiri dari beberapa komponen utama, yaitu sensor PIR, buzzer, LED, serta modul ESP32-CAM sebagai unit kamera terpisah. Sensor PIR berfungsi untuk mendeteksi gerakan berdasarkan perubahan radiasi inframerah, dan ketika gerakan terdeteksi, sinyal digital akan dikirim ke ESP32 melalui pin input. ESP32 kemudian akan mengirimkan notifikasi secara otomatis ke pengguna melalui bot Telegram, serta memberikan perintah ke ESP32-CAM untuk mengambil gambar secara real-time.

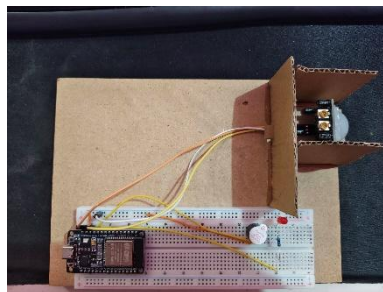
Selain itu, sistem ini juga memungkinkan pengguna untuk mengontrol buzzer sebagai alarm suara melalui perintah `/buzzer_on`, serta memicu kamera secara manual dengan perintah `/foto`. LED digunakan sebagai indikator visual tambahan yang menyala bersamaan dengan buzzer. Semua komponen terhubung melalui breadboard, dengan suplai tegangan dari pin 3.3V dan GND pada ESP32. Untuk komunikasi antara ESP32 dan ESP32-CAM dapat menggunakan protokol HTTP sebagai media penghubung antar perangkat. Dengan skema ini, sistem mampu bekerja secara otomatis maupun manual, memberikan keamanan yang responsif, fleksibel, dan dapat dipantau dari jarak jauh.

3.3. Implementasi

Setelah proses perancangan selesai dilakukan, tahap selanjutnya adalah implementasi sistem secara menyeluruh. Implementasi ini mencakup dua aspek utama, yaitu perangkat keras (hardware) dan perangkat lunak (software) yang saling terintegrasi.

3.3.1. Perangkat Keras

Perangkat keras, atau hardware, memiliki peran vital dalam sistem keamanan berbasis IoT karena berfungsi untuk mendeteksi, memproses, serta mengirimkan informasi secara real-time. Dalam sistem ini, komponen utama yang digunakan adalah ESP32, ESP32-CAM, sensor PIR, buzzer, dan LED. Setiap komponen memiliki tugas khusus yang saling terintegrasi dalam menjalankan proses keamanan otomatis. Sensor PIR bertindak sebagai indera pendeteksi gerakan, ESP32 sebagai otak pemroses utama dan pengendali aktuator, sedangkan ESP32-CAM berfungsi sebagai unit kamera yang mengambil gambar dan mengirimkannya ke Telegram. Semua komponen ini harus dirangkai dengan baik agar sistem dapat berjalan optimal dan sesuai fungsinya.



Gambar 5. Rangkaian Perangkat Keras

Gambar 5 menunjukkan perakitan sistem deteksi gerakan yang terdiri dari modul ESP32, sensor PIR, buzzer, dan LED. Dalam konfigurasi ini, sensor PIR mendeteksi adanya gerakan di sekitar area pengawasan dan mengirimkan sinyal ke ESP32. Mikrokontroler ESP32 kemudian memproses sinyal tersebut, mengirimkan notifikasi ke Telegram, serta dapat mengaktifkan buzzer dan LED sebagai alarm visual dan suara. Selain itu,

pengguna juga dapat mengontrol buzzer dan LED dari jarak jauh melalui bot Telegram dengan perintah tertentu seperti /buzzer_on. Rangkaian ini ditata di atas breadboard dengan sambungan yang rapi, serta sumber daya diambil dari pin VIN dan GND milik ESP32. Modul ini menjadi sistem awal yang bertugas memicu seluruh alur keamanan dalam proyek.



Gambar 6. Rangkaian Perangkat Keras

Gambar 6 memperlihatkan perakitan ESP32-CAM sebagai komponen utama dalam pengambilan gambar dan pemantauan visual. Modul ESP32-CAM bekerja dengan menerima perintah dari ESP32 melalui protokol HTTP, seperti perintah /snap yang akan memicu pengambilan gambar saat gerakan terdeteksi. Modul ini juga mendukung pengambilan gambar secara manual dan fitur live streaming yang dapat diakses melalui link ngrok. ESP32-CAM hanya memerlukan sambungan daya dan koneksi WiFi untuk dapat menjalankan fungsinya, menjadikannya modul yang efisien namun sangat penting dalam mendukung sistem keamanan secara menyeluruh.

3.3.2. Penulisan Program

1. Inisialisasi awal

Adapun penjelasan inisialisasi awal disajikan pada gambar di bawah.

```
#define PIR_PIN 13
#define BUZZER_PIN 12
#define LED_PIN 14

#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <HTTPClient.h>
#include <time.h>
```

Gambar 7. Inisialisasi awal

Bagian ini mendefinisikan pin-pin perangkat yang terhubung ke ESP32: sensor PIR, buzzer, dan LED. Kemudian, program mengimpor pustaka-pustaka penting seperti WiFi untuk koneksi internet, UniversalTelegramBot untuk interaksi dengan Telegram, HTTPClient untuk komunikasi HTTP ke ESP32-CAM, serta time.h untuk mendapatkan waktu nyata.

2. Konfigurasi Koneksi

Adapun penjelasan Konfigurasi Koneksi sebagai berikut.

```
const char* ssid = "fh_32aad0";
const char* password = "a1140600";
```

Gambar 8. Konfigurasi Koneksi

ESP32 terhubung ke WiFi menggunakan SSID dan password yang disediakan. Ini wajib agar ESP32 bisa mengakses Telegram dan mengirim perintah ke ESP32-CAM.

3. Setup Bot Telegram

Penjelasan Setup Bot Telegram disajikan dalam gambar di bawah.

```
#define BOT_TOKEN "7875290988:AAHdk0TMA1AD-bzC9qnJ0LbQ5NkoZVno6w0"
#define CHAT_ID "1326995060"

UniversalTelegramBot bot(BOT_TOKEN, client);
```

Gambar 9. Setup Bot Telegram

Bot Telegram dikonfigurasi agar ESP32 bisa mengirim dan menerima pesan. Bot ini nanti akan digunakan untuk memberi notifikasi gerakan, snapshot, serta kontrol buzzer dan status sistem.

4. Pengaturan Waktu

Disajikan penjelasan Pengaturan Waktu pada gambar di bawah.

```
String getNow() {
  time_t now;
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) return "WAKTU ERROR";
  char buf[24];
  strftime(buf, sizeof(buf), "%Y-%m-%d %H:%M:%S", &timeinfo);
  return String(buf);
}
```

Gambar 10. Pengaturan Waktu

Fungsi ini digunakan untuk mengambil waktu lokal menggunakan NTP server. Waktu ini ditampilkan di setiap pesan notifikasi Telegram agar pengguna tahu kapan kejadian berlangsung..

5. Komunikasi ke ESP32-CAM

Adapun penjelasan Komunikasi ke ESP32-CAM disajikan dalam gambar berikut.

```
const char* cam_ip = "192.168.1.11";
const int cam_port = 80;

void requestSnapshot() {
  String url = "http://" + String(cam_ip) + ":" + String(cam_port) + "/snap";
  HTTPClient http;
  http.begin(url);
  int httpCode = http.GET();
  if (httpCode > 0) {
    Serial.println("Permintaan snapshot ke ESP32-CAM berhasil.");
  } else {
    Serial.println("Permintaan snapshot ke ESP32-CAM gagal.");
  }
  http.end();
}
```

Gambar 11. Ambang Batas dan Reset Mode

ESP32 mengirim permintaan HTTP GET ke ESP32-CAM melalui alamat IP lokal dan port 80. Endpoint /snap digunakan untuk mengambil gambar. Ini memungkinkan integrasi otomatis antar ESP melalui jaringan lokal.

6. Deteksi Gerakan & Notifikasi

Penjelasan Deteksi Gerakan & Notifikasi disajikan dalam gambar di bawah.

```
if (pirState == HIGH && !motionDetected) {
  motionDetected = true;
  digitalWrite(LED_PIN, HIGH);

  if (alarmActive && (millis() - lastMotionTime > motionNotifyInterval)) {
    lastMotionTime = millis();
    String message = "▲ Gerakan terdeteksi!\nWaktu: " + getNow();
    if (bot.sendMessage(CHAT_ID, message, "")) {
      Serial.println("Pesan Telegram terkirim");
    } else {
      Serial.println("Gagal kirim pesan Telegram");
    }
    // Minta snapshot ke ESP32-CAM
    requestSnapshot();
  }
} else if (pirState == LOW) {
  motionDetected = false;
  digitalWrite(LED_PIN, LOW);
}
```

Gambar 12. Deteksi Gerakan & Notifikasi

ESP32 secara berkala memeriksa sensor PIR. Jika mendeteksi gerakan, LED akan menyala, pesan telegram akan dikirim dengan timestamp, dan sistem akan menunggu interval 5 detik sebelum mengirim notifikasi lagi.

7. Respons Telegram

Penjelasan Respons Telegram disajikan dalam gambar di bawah.

```

int numMessages = bot.getUpdates(bot.last_message_received + 1);
for (int i = 0; i < numMessages; i++) {
  String text = bot.messages[i].text;
  String chat_id = String.valueOf(messages[i].chat_id);

  if (chat_id == CHAT_ID) {
    if (text == "/buzzer_on") {
      Serial.println("Buzzer ON command received");
      digitalWrite(BUZZER_PIN, HIGH);
      bot.sendMessage(CHAT_ID, "🔔 Buzzer dihidupkan.");
    } else if (text == "/buzzer_off") {
      Serial.println("Buzzer OFF command received");
      digitalWrite(BUZZER_PIN, LOW);
      bot.sendMessage(CHAT_ID, "🔕 Buzzer dimatikan.");
    } else if (text == "/alarm_on") {
      alarmActive = true;
      bot.sendMessage(CHAT_ID, "🟢 Sistem diaktifkan.");
    } else if (text == "/alarm_off") {
      alarmActive = false;
      bot.sendMessage(CHAT_ID, "🔴 Sistem dimatikan.");
    } else if (text == "/status") {
      String statusMsg = "📊 Status sistem:\n";
      statusMsg += "Alarm notifikasi: " + String(alarmActive) + " 🟢/🔴" + "\n";
      statusMsg += "Buzzer: " + String(digitalRead(BUZZER_PIN) & "ON" : "OFF") + "\n";
      statusMsg += "Waktu: " + getLocalTime();
      bot.sendMessage(CHAT_ID, statusMsg);
    } else if (text == "/foto") {
      bot.sendMessage(CHAT_ID, "📷 Mengambil snapshot dari kamera...");
      requestSnapshot();
    }
  }
}

```

Gambar 13. Respons Telegram

Jika pengguna mengirim pesan Telegram, ESP32 akan memproses perintah berikut, /buzzer_on dan /buzzer_off untuk menghidupkan/menonaktifkan buzzer, /alarm_on dan /alarm_off untuk menghidupkan sistem / mematikan sistem, /status untuk menampilkan status sistem, termasuk buzzer dan waktu, /foto untuk meminta ESP32-CAM mengambil gambar secara manual.

3.3.3. Pengembangan Antarmuka

1. Pembuatan Bot Telegram

Kita bisa lihat di gambar 14 cara membuat bot telegram untuk sistem yang mendeteksi gerakan menggunakan botfather.



Gambar 14. Pembuatan Bot

Untuk membuat bot baru di BotFather, kita perlu mengikuti beberapa langkah, pertama ketik perintah “/newbot”. Lalu, masukkan nama dan username yang kamu inginkan untuk botmu, setelah itu, kamu akan mendapat token yang bisa digunakan untuk mengontrol botmu. Gambar 15 menunjukkan tampilan bot yang sudah jadi.

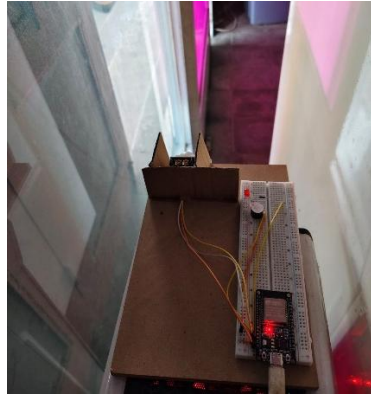


Gambar 15. Tampilan Profile Bot

Bot Telegram dikembangkan menggunakan fitur Bot Father yang disediakan oleh platform Telegram Messenger. Desain antarmuka bot yang menyerupai kotak chat konvensional memudahkan pengguna untuk berinteraksi dengannya.

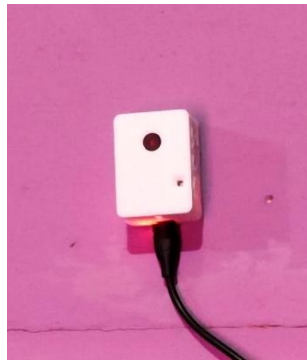
3.4. Pengujian

3.4.1. Pengujian Rangkaian Alat



Gambar 16. Pengujian ESP32

Pengujian ESP32 dilakukan untuk memastikan bahwa mikrokontroler utama dapat menjalankan fungsi deteksi gerakan, pengendalian perangkat output (buzzer dan LED), serta komunikasi dengan pengguna melalui aplikasi telegram. Sensor PIR yang terhubung ke ESP32 akan aktif memindai gerakan di sekitarnya. Ketika terdeteksi adanya pergerakan, ESP32 akan menyalakan LED sebagai indikator visual dan secara otomatis mengirimkan notifikasi ke telegram berisi waktu kejadian serta peringatan gerakan.



Gambar 17. Pengujian ESP32-CAM

Setelah ESP32 berhasil mendeteksi gerakan, langkah selanjutnya adalah pengujian terhadap ESP32-CAM sebagai modul pengambil gambar. ESP32 mengirimkan perintah melalui protokol HTTP ke ESP32-CAM, yang kemudian memproses perintah tersebut untuk mengambil gambar (snapshot) dan mengirimkannya ke pengguna melalui Telegram, bisa di lihat di gambar 17. Selain fungsi otomatis, ESP32-CAM juga dapat dikontrol langsung oleh pengguna melalui beberapa perintah bot.

3. 4. 2. *Pengujian Tampilan Antarmuka*

Adapun gambar pengujian tampilan antarmuka disajikan sebagai berikut.



Gambar 18. Pengujian Telegram Bot

Hasil pengujian menunjukkan bahwa seluruh perintah dan fitur pada bot Telegram KakaSecurityBot berjalan dengan baik, responsif, dan tanpa kesalahan fungsi. Sistem berhasil merespons perintah seperti /status, /buzzer_on, /foto, dan /alarm_on secara real-time, serta mengirimkan notifikasi otomatis saat sensor PIR mendeteksi gerakan. Gambar hasil tangkapan ESP32-CAM juga dapat diterima dengan jelas, dan peralihan antara mode manual dan otomatis dapat dilakukan tanpa hambatan. Selain itu, setelah permintaan pengambilan foto dikirim, sistem secara otomatis mengirimkan tautan live streaming dari ESP32-CAM menggunakan layanan ngrok, sehingga pengguna dapat langsung memantau kondisi area secara real-time. Oleh karena itu, dapat disimpulkan bahwa pengujian antarmuka Telegram Bot berhasil dan mampu mendukung sistem keamanan real-time secara optimal, efisien, dan praktis untuk kebutuhan pemantauan jarak jauh.

3. 4. 3. Hasil Pengujian Sistem

Melalui pengujian, kita dapat memastikan kualitas sistem, fungsinya, dan kebebasan dari kesalahan. Hal ini penting untuk mencegah masalah saat digunakan oleh pengguna.

Tabel 1. Hasil Pengujian Bot Telegram

Aktivitas Pengujian	Relasi yang diharapkan	Hasil Pengujian	Relasi yang terjadi
/alarm_on	Menyalakan sistem	berhasil	Sistem aktif & siap mendeteksi gerakan
/alarm_off	Menonaktifkan sistem	berhasil	Sistem tidak merespon gerakan
/buzzer_on	Menyalakan buzzer	Berhasil	buzzer menyala sesuai perintah
/buzzer_off	Mematikan buzzer	Berhasil	Buzzer mati setelah perintah
/status	Menampilkan status sistem	berhasil	Status alarm, buzzer, dan waktu ditampilkan
/foto	Mengambil gambar manual	berhasil	Gambar terkirim melalui telegram

Tabel 2. Hasil Pengujian Jarak sensor PIR

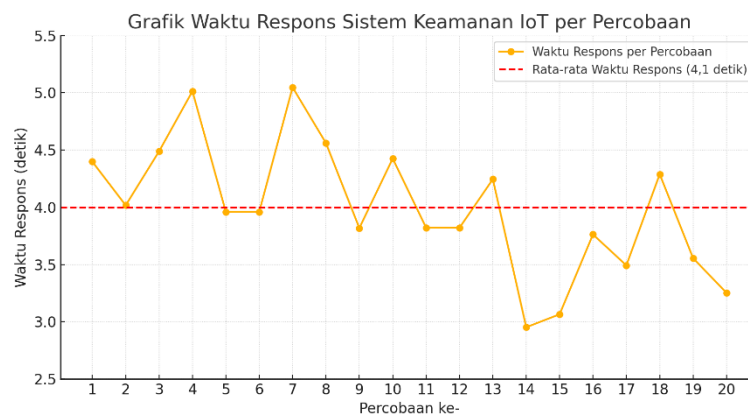
No	Objek	Jarak	Notifikasi	Keterangan
1	Manusia	1m	Terkirim	Mendeteksi
2	Manusia	2m	Terkirim	Mendeteksi

3	Manusia	3m	Terkirim	Mendeteksi
4	Manusia	4m	Terkirim	mendeteksi
5	Manusia	5m	Terkirim	mendeteksi
6	Manusia	6m	Tidak terkirim	Tidak mendeteksi

Tabel 3. Metrik Performa Sistem Keamanan IoT berbasis ESP32 dan Telegram

Metrik	Nilai rata-rata	Standar Deviasi	Keterangan
Waktu Respons (detik)	4.1	0.6	Dari deteksi gerak sampai notifikasi masuk
Error Rate (%)	2.5	-	Gagal kirim notifikasi/foto
Latensi komunikasi	320	40	Rata-rata keterlambatan koneksi Telegram

Berdasarkan pengujian sebanyak 20 kali, sistem mampu mengirim notifikasi dalam waktu rata-rata 4.1 detik. Error rate sebesar 2.5% umumnya disebabkan gangguan koneksi, dan latensi komunikasi rata-rata 320 ms membuktikan sistem tetap responsif untuk aplikasi real-time.



Gambar 19. Grafik waktu respons sistem keamanan IoT pada 20 kali percobaan

Grafik pada Gambar 19 menunjukkan bahwa waktu respons sistem relatif stabil pada kisaran 4 detik untuk setiap percobaan, sehingga membuktikan sistem mampu bekerja secara real-time dan konsisten.

4. KESIMPULAN

Sistem keamanan real-time berbasis Internet of Things (IoT) yang dibangun menggunakan ESP32 dan ESP32-CAM pada warnet Kaka Game Center berhasil diimplementasikan dan diuji dengan baik. Sistem ini mampu mendeteksi gerakan menggunakan sensor PIR, mengirim notifikasi otomatis ke Telegram, memicu pengambilan gambar secara real-time, serta menyediakan fitur kendali manual seperti buzzer, snapshot, dan live streaming melalui ngrok. Seluruh fungsi berjalan dengan respons cepat dan stabil, baik dalam kondisi siang maupun malam hari. Dengan memanfaatkan perangkat yang ekonomis dan aplikasi Telegram sebagai antarmuka, sistem ini terbukti efektif, fleksibel, dan mudah diakses jarak jauh, sehingga layak dijadikan alternatif sistem keamanan modern untuk usaha skala kecil.

REFERENSI

- [1] J. Wilson Sitopu, M. Sigid Safarudin, M. Wahyu Suryandi Adam, and M. Safar, "Mengenal Internet of Things (IoT): Penerapan Konsep dan Manfaatnya dalam Kehidupan Sehari-hari," *Journal of Human And Education*, vol. 4, no. 4, pp. 827, 2024.
- [2] M. K. Ikhsan and I. P. Sari, "Perancangan Sistem Keamanan Kamar Kos Berbasis Internet of Things (IoT) dengan Menggunakan Wemos D1 Mini dan Sensor Pintu, serta Integrasi Pemberitahuan Melalui Telegram," *Blend Sains Jurnal Teknik*, vol. 3, no. 2, pp. 229–236, 2025, doi: 10.56211/blendsains.v3i2.639.
- [3] N. H. Setiyawan, A. Hariyadi, and A. Kurniawan, "Sistem Pengawasan CCTV Pada ATM Secara Real-Time Berbasis Internet of Things," *Jurnal Janitra Informatika Dan Sistem Informasi*, vol. 5, no. 1, 2025, doi: 10.59395/k0e60m33.

- [4] J. Bisnis Perspektif BIP, P. Mario Ericko Venandra, S. Astrid Ayu, P. Studi Manajemen, F. Ekonomi, and U. Katolik Darma Cendika, "Pengaruh Harga, Fasilitas, Citra Merek Terhadap Kepuasan Pengguna Warnet dan Game Online," *Bisnis Perspektif BIP*, [Online]. Available: <http://jurnal.ukdc.ac.id/index.php/BIP>.
- [5] A. Irawan et al., "Tantangan dan Strategi Manajemen Keamanan Siber di Indonesia berbasis IoT," *Journal ZETROEM*, vol. 6, no. 1, pp. 114-119, 2024.
- [6] S. Adetia, A. Wicaksono, S. Gustina, S. D. Hartiyani, P. Studi, T. Informasi, and F. Teknik, "Penerapan Sistem Keamanan Berbasis Internet of Things (IoT) dengan Sensor Alarm Buzzer," *Media Informatika*, vol. 24, no. 1, 2025.
- [7] L. Cakrayuda, "SICEMOT: Sistem Keamanan Cerdas Berbasis ESP32-CAM, Sensor Gerak, dan Notifikasi Telegram," *Jurnal Informatika Dan Teknik Elektro Terapan*, vol. 13, no. 2, 2025, doi: 10.23960/jitet.v13i2.6420.
- [8] F. Abdurrokhman, M. STMIK Insan Pembangunan, and D. Tetap STMIK Insan Pembangunan, "Implementasi Internet of Things (IoT) untuk Meningkatkan Keamanan pada Warnet Tara Komputer," 9(1), 2021.
- [9] Y. Efendi, "Internet of Things (IoT) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile," *Jurnal Ilmiah Ilmu Komputer*, vol. 4, no. 1, 2018. [Online]. Available: <http://ejournal.fikom-unasman.ac.id>.
- [10] H. Kurniawan, W. Apriliah, I. Kurniawan, and D. Firmansyah, "Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Penggajian pada SMK Bina Karya Karawang," *J. Interkom*, vol. 14, no. 4, pp. 13-23, Jan. 2020, doi: 10.35969/interkom.v14i4.58.
- [11] M. F. Nurpulaela and L. Azhar, "Implementasi Penggunaan ESP32 Sebagai IoT pada Project Smart Charger di PT. Pasifik Satelit Nusantara Bekasi," Skripsi, Fakultas Teknik, Universitas [lengkapi nama universitas], 2024.
- [12] N. Abekiri, A. Rachdy, M. Ajaamoum, B. Nassiri, L. Elmahni, and Y. Oubail, "Platform for hands-on remote labs based on the ESP32 and NOD-red," *Sci. African*, vol. 19, p. e01502, 2023, doi: 10.1016/j.sciaf.2022.e01502.
- [13] A. Juliansyah and D. Nadiani, "Sistem pendeteksi gerak menggunakan sensor PIR dan Raspberry Pi (Motion detection system using PIR sensors and Raspberry Pi)," [Nama Jurnal], vol. 2, no. 4, pp. 199-205, 2021.
- [14] A. D. Mulyanto, "Pemanfaatan Bot Telegram untuk Media Informasi Penelitian," *Matics*, vol. 12, no. 1, p. 49, 2020, doi: 10.18860/mat.v12i1.8847.
- [15] M. Fezari and A. Al Dahoud, "Integrated Development Environment 'IDE' For Arduino: Introduction to Arduino IDE," 2018. [Online]. Available: <https://www.researchgate.net/publication/328615543>.
- [16] A. Isrofi, S. N. Utama, and O. V. Putra, "Rancang bangun robot pemotong rumput otomatis menggunakan wireless kontroler modul ESP32-CAM berbasis Internet of Things (IoT)," *J. Teknoinfo*, vol. 15, no. 1, p. 45, 2021, doi: 10.33365/jti.v15i1.675.
- [17] G. C. Lenardo, "Pemanfaatan Bot Telegram sebagai Media Informasi Akademik di STMIK Hang Tuah Pekanbaru (Utilization of Telegram Bot as Academic Information Media at STMIK Hang Tuah Pekanbaru)," vol. 1, no. 4, pp. 351-357, 2020.