

## Sistem Siaga Karhutla Berbasis IoT dan Energi Terbarukan Terintegrasi LoRa

### *IoT-Based Wildfire Early Warning System Integrated with LoRa and Renewable Energy*

Dhavi Novrizal<sup>1</sup>, Radja Reno Simamora<sup>2</sup>, Yuri Rahmanto<sup>3</sup>

<sup>1,2,3</sup> Teknik Komputer, Fakultas Teknik Dan Ilmu Komputer, Universitas Teknokrat Indonesia  
Jl. ZA. Pagar Alam No.9 -11, Labuhan Ratu, Kec. Kedaton, Kota Bandar Lampung, Lampung  
35132

<sup>1</sup>dhavi\_novrizal@teknokrat.ac.id, <sup>2</sup>radja\_reno\_simamora@teknokrat.ac.id,  
<sup>3</sup>yurirahmanto@teknokrat.ac.id

**Abstrak** – Kebakaran hutan dan lahan (Karhutla) menjadi ancaman tahunan di Indonesia dengan dampak besar terhadap lingkungan dan manusia. Contohnya pada tahun 2023, hilangnya keanekaragaman hayati ±1.958,71 ha lahan terbakar di Bangka Belitung, pencemaran udara akibat kabut asap PM2.5 [1], serta dampak pada manusia berupa gangguan pernapasan/ISPA yang tercatat lebih dari 200 ribu warga terkena infeksi saluran pernapasan akibat kabut asap [2], kerugian ekonomi sebesar Rp 150 miliar di Bangka Belitung dan Sumatera [1], dan ancaman keselamatan jiwa terutama pada anak-anak dan balita. Berdasarkan masalah diatas, penulis mengusulkan sistem deteksi dini Karhutla yang dapat dioperasikan di wilayah yang jauh dari pemukiman. Sistem deteksi konvensional berbasis sensor suhu atau asap sering terlambat dan tidak menyediakan data secara real-time. sistem yang diusulkan yaitu sistem pemantauan Karhutla berbasis Artificial Intelligence of Things (AIoT) yang menggabungkan kamera, Raspberry Pi 4, Model AI YOLOv5, ESP32-S3, dan komunikasi LoRa. Kamera menangkap citra lingkungan, diproses oleh Raspberry Pi menggunakan YOLOv5 untuk mendeteksi api. Hasil deteksi dikirim ke ESP32-S3, yang kemudian meneruskan data melalui LoRa ke perangkat penerima. ESP32-S3 penerima menampilkan data ke perangkat mobile untuk pemantauan. Sensor DHT22 digunakan sebagai validasi suhu ekstrem ketika Kebakaran sudah mendekat. Sistem didukung panel surya untuk operasional mandiri di area yang jauh dari pemukiman warga. Integrasi IoT dan deep learning memungkinkan sistem memberikan peringatan dini secara cepat dan akurat, membantu mitigasi kebakaran dan mengurangi potensi kerugian ekonomi maupun jiwa.

**Kata Kunci:** kebakaran hutan, Artificial intelligence of things, LoRa, Peringatan Dini, Solar Panel.

**Abstract** – Forest and land fires (known in Indonesian as Karhutla) are an annual threat in Indonesia, causing severe impacts on the environment and human health. In 2023, biodiversity loss reached about 1,958.71 hectares of burned land in Bangka Belitung, with PM2.5 haze [1] causing air pollution and more than 200,000 respiratory infection (ARI) cases [2]. Economic losses totaled IDR 150 billion in Bangka Belitung and Sumatra [1], with serious safety risks, especially for children and infants. To address this, researchers propose an early detection system for Karhutla that operates in remote areas far from settlements. Conventional systems using temperature or smoke sensors are often slow and lack real-time data. The proposed system is an AIoT-based monitoring solution combining a camera, Raspberry Pi 4, YOLOv5 AI model, ESP32-S3, and LoRa communication. The camera captures environmental images, processed by the Raspberry Pi with YOLOv5 to detect fire. Detection results are sent to the ESP32-S3, which transmits data via LoRa to a receiver. The receiving ESP32-S3 displays information on a mobile device for monitoring. A DHT22 sensor validates extreme temperature when a fire has occurred and is approaching. Solar panels power the system for independent operation in remote areas. Integrating IoT

*and deep learning enables fast, accurate early warnings, supporting fire mitigation and reducing economic losses and risks to human lives.*

**Keywords:** *Forest fire, Artificial Intelligence of Things (AIoT), LoRa, Early Detection, Solar Panel.*

## 1. Pendahuluan

Hutan memiliki fungsi vital dalam menjaga keseimbangan kehidupan makhluk hidup. Namun, kebakaran hutan (Karhutla) kini menjadi isu yang serius karena dapat mengganggu hubungan timbal balik dan kelangsungan ekosistem. Permasalahan ini sudah menjadi perhatian bagi masyarakat, pemerintah maupun dunia. Fenomena ini terjadi setiap tahun yang di akibatkan oleh faktor alam dan 99% ulah dari manusia itu sendiri [4]. Data kebakaran hutan dan lahan (Karhutla) menunjukkan tren yang fluktuatif dari tahun ke tahun. Pada tahun 2020 tercatat luas Karhutla mencapai 296.942 hektare, meningkat menjadi 358.867 hektare pada 2021, lalu menurun menjadi 204.894 hektare pada 2022. Namun, terjadi lonjakan signifikan pada 2023 dengan luas kebakaran mencapai 1.161.192,90 hektare. Di tahun 2024, angka tersebut kembali menurun menjadi 376.805,05 hektare, dan hingga Juni 2025, Karhutla telah mencakup 8.594,49 hektare. Untuk mengatasi permasalahan ini, pemerintah telah melakukan berbagai upaya, baik berupa himbauan maupun sanksi hukum terhadap tindakan yang mengancam kelestarian hutan, baik dilakukan oleh individu maupun kelompok. Namun, upaya tersebut masih belum sepenuhnya efektif, karena tingkat kerusakan dan kebakaran hutan masih menunjukkan angka yang memprihatinkan, sehingga diperlukan langkah pencegahan yang lebih proaktif dan terukur.

Salah satu langkah penting tersebut adalah membangun sistem peringatan dini Karhutla yang mampu menghindarkan potensi bencana lanjutan. Sistem ini dimaksudkan supaya masyarakat dan pihak terkait mendapat informasi lebih awal terkait Karhutla [6]. Banyak kasus kebakaran hutan diketahui pada saat sudah memakan lahan dalam durasi waktu yang cukup lama. Informasi mengenai terjadinya kebakaran umumnya berasal dari masyarakat di sekitar lokasi kejadian. Keterlibatan warga diperlukan untuk melaporkan kebakaran kepada petugas pemadam agar penanganan dapat segera dilakukan. Namun, pendekatan konvensional yang mengandalkan partisipasi masyarakat ini memiliki kelemahan, terutama dalam hal keterlambatan penyampaian informasi. Oleh sebab itu, dibutuhkan sistem yang mampu memberikan peringatan dini secara otomatis terkait kejadian kebakaran hutan atau lahan (Karhutla).

Jika melihat penerapan di lapangan yang sudah ada, sistem deteksi kebakaran konvensional umumnya menggunakan sensor suhu atau asap untuk mendeteksi adanya api, namun sistem ini memiliki beberapa kelemahan, seperti keterlambatan dalam mendeteksi api yang baru muncul dan ketidakmampuan memberikan informasi visual mengenai lokasi kebakaran secara langsung. Hal ini menyulitkan petugas dalam menentukan titik api secara akurat dan cepat, serta memantau perkembangan kebakaran secara real-time.

Beberapa penelitian telah dilakukan untuk mengatasi kelemahan sistem deteksi kebakaran konvensional. Penelitian tersebut merancang sistem pendeteksi kebakaran berbasis mikrokontroler dan aplikasi peta menggunakan *IoT*, yang dapat memberikan informasi lokasi kebakaran secara *real-time*[7], namun jarak deteksi nya bisa dikatakan kecil sampai 10-60 cm saja. Zidifaldi mengembangkan sistem deteksi dini menggunakan sensor api dan suhu berbasis Arduino, dengan *notifikasi* ke *smartphone* melalui aplikasi *Blynk*[8]. Selain itu, sistem berbasis *computer vision* juga mulai dikembangkan, seperti pada penelitian Widharma yang menggunakan algoritma *YOLO* untuk mendeteksi api secara *real-time* dan memberikan *notifikasi* melalui Telegram serta *buzzer*[9].

Penelitian lain oleh Pratama dan Kurniawan mengimplementasikan algoritma *deep learning* untuk deteksi kebakaran pada lingkungan industri[10], sedangkan Sari dan Yusuf mengembangkan sistem pendeteksi berbasis sensor MQ-2 dan *platform IoT*[11]. Lestari dan Wicaksono menggabungkan sensor suhu dan kamera untuk deteksi dini kebakaran[13]. Hidayat

dalam penelitiannya menerapkan *YOLOv4* untuk deteksi api sebagai bagian dari sistem keamanan rumah[14]. Rahman dan Dewi menggunakan *Raspberry Pi* dan kamera sebagai basis sistem deteksi otomatis[15]. Sementara itu, Santoso dan Widodo menguji kinerja sistem deteksi kebakaran berbasis *IoT* di gedung perkantoran[16].

Berdasarkan masalah dan penelitian-penelitian yang sudah dilakukan tersebut, peneliti mengusulkan pengembangan sistem pendeteksi kebakaran berbasis *IoT* dan *computer vision* yang lebih optimal dan akurat serta menggunakan panel surya sebagai sumber daya untuk implementasi di hutan. Sistem yang diusulkan akan mengintegrasikan kamera untuk mendeteksi kebakaran dengan algoritma deteksi objek berbasis *YOLO* untuk meningkatkan akurasi dan kecepatan deteksi. Ketika kamera mendeteksi adanya api atau asap, sistem akan otomatis mengirimkan *notifikasi* peringatan dini kepada operator sebagai cara awal menanggulangi karhutla. Perbedaan utama antara sistem yang diusulkan dengan sistem yang sudah ada terletak pada integrasi teknologi *IoT* dan *computer vision* yang lebih canggih, penggunaan algoritma *deep learning* untuk meningkatkan akurasi deteksi, serta penggunaan panel surya sebagai sumber daya. Dengan demikian, diharapkan sistem ini dapat memberikan *respons* yang lebih cepat dan tepat dalam penanggulangan kebakaran, sehingga risiko kerugian material dan korban jiwa dapat diminimalkan.

## 2. Metode Penelitian

Metode penelitian ini membahas dasar konseptual dan teknis perancangan sistem deteksi kebakaran hutan dan lahan (Karhutla) berbasis kecerdasan buatan. Pembahasan mencakup definisi, penyebab, serta tren Karhutla di Indonesia, dan metode deteksi berbasis *YOLOv5* meliputi prinsip kerja, pembagian *grid*, *bounding box*, *NMS*, dan *tensor* output sebagai landasan ilmiah pengembangan sistem mitigasi bencana yang efisien dan terintegrasi.

### 2.1. Kebakaran Hutan dan Lahan (Karhutla)

Kebakaran hutan dan lahan (Karhutla) merupakan peristiwa terbakarnya area hutan maupun lahan yang menyebabkan kerusakan ekosistem, pencemaran udara, serta gangguan kesehatan dan sosial ekonomi. Menurut *FAO* (2007), kebakaran hutan dapat terjadi karena sebab alami maupun ulah manusia dan sering kali tidak terkendali. Di Indonesia sendiri, Karhutla seringkali terjadi secara berulang tiap tahun, terutama di wilayah yang memiliki lahan gambut serta tingkat aktivitas pembukaan lahan yang tinggi.

Penyebab Karhutla dapat dijelaskan melalui beberapa pendekatan, seperti teori segitiga api menyatakan bahwa api membutuhkan tiga elemen utama, yaitu bahan bakar, oksigen, dan panas. kondisi musim kemarau, vegetasi kering menjadi bahan bakar alami yang sangat mudah terbakar, selain itu, teori antropogenik menyebutkan bahwa faktor manusia merupakan penyebab utama Karhutla, seperti pembukaan lahan dengan metode bakar, perburuan, atau kelalaian lainnya. Sementara itu, teori iklim menunjukkan bahwa fenomena seperti *El Niño* yang memperpanjang musim kemarau turut memperburuk risiko Karhutla [19].

Berdasarkan data Sistem Pemantauan Hutan Indonesia (SiPongi), Karhutla masih menjadi ancaman serius dalam beberapa tahun terakhir. Luas area terbakar tercatat 296.942 hektare pada 2020, meningkat menjadi 358.867 hektare pada 2021, lalu menurun ke 204.894 hektare pada 2022. Tahun 2023 terjadi lonjakan signifikan hingga 1.161.192,90 hektare, sebelum kembali turun menjadi 376.805,05 hektare pada 2024. Hingga Juni 2025, tercatat 8.594,49 hektare lahan terbakar. Fluktuasi ini menunjukkan bahwa intensitas Karhutla sangat dipengaruhi oleh kondisi cuaca dan aktivitas manusia.

Secara umum, Karhutla di Indonesia lebih banyak disebabkan oleh aktivitas manusia, seperti pembukaan lahan dengan metode pembakaran yang dianggap cepat dan murah. Namun, saat musim kemarau panjang, api yang semula terkendali dapat dengan mudah meluas menjadi

bencana besar. Oleh karena itu, penanggulangan Karhutla perlu mencakup pencegahan berbasis teknologi, penegakan hukum, edukasi masyarakat, dan pemantauan iklim secara berkelanjutan.

## 2.2. Metode *You Only Look Once* (YOLO)

*YOLO* (*You Only Look Once*) adalah metode deteksi objek berbasis *convolutional neural network* (CNN) yang memandang deteksi objek sebagai regresi tunggal dari gambar *input* ke *bounding box* dan kelas objek. Joseph Redmon et al. (2016) memperkenalkan metode ini untuk menghasilkan deteksi cepat dan akurat dalam satu tahap pemrosesan. *YOLO* memprediksi posisi dan klasifikasi objek dalam satu langkah *forward propagation* tanpa tahap proposal seperti pada *R-CNN*.

Metode *YOLO* dapat digunakan untuk mendeteksi nyala api dengan pola visual khas, seperti warna merah-oranye, bentuk tidak beraturan, dan intensitas cahaya tinggi. Model *YOLO* dilatih untuk mengenali ciri visual nyala api dan melakukan deteksi real-time menggunakan dataset kebakaran. Pendekatan ini efektif untuk sistem pemantauan bencana berbasis kamera atau *drone* karena efisien dalam waktu inferensi, penting untuk peringatan dini dan respons cepat terhadap kebakaran hutan.

### A. Pemotongan Gambar Menjadi *Grid*

Membagi gambar *input* menjadi *grid* berukuran  $S \times S$ , seperti  $13 \times 13$  atau  $19 \times 19$ , merupakan salah satu metode utama dalam arsitektur *YOLO*. Setiap sel bertanggung jawab mendeteksi objek jika pusat objek berada dalam area sel tersebut. Pembagian ini memungkinkan deteksi paralel dan mempercepat proses inferensi. Misalnya, gambar berukuran  $416 \times 416$  piksel yang dibagi menjadi  $13 \times 13$  *grid* menghasilkan sel berukuran  $32 \times 32$  piksel. Setiap sel mewakili beberapa *anchor box*, yaitu template ukuran objek untuk mempermudah lokalisasi dan klasifikasi.

Pada kasus kebakaran hutan, ukuran api dapat bervariasi di berbagai area gambar. Metode *grid* ini membuat *YOLO* mampu memantau seluruh area dan mendeteksi nyala api yang tersebar, jauh, atau berkelompok tanpa mengorbankan akurasi maupun kecepatan.

### B. *Bounding Box*

Beberapa prediksi *bounding box*. Masing-masing *bounding box* berisi lima parameter:

1. Koordinat pusat objek ( $x, y$ ).
2. Lebar ( $w$ ) dan tinggi ( $h$ ) *bounding box*.
3. Nilai confidence ( $C$ ), yaitu skor keyakinan bahwa terdapat objek di dalam box tersebut dan bahwa box tersebut akurat.

Nilai *confidence* diperoleh dari kombinasi probabilitas keberadaan objek dan nilai *Intersection over Union* (*IoU*) antara box prediksi dan *ground truth*. Secara teoritis, *bounding box* bertujuan melokalisasi objek secara tepat dalam gambar. Dalam sistem deteksi kebakaran hutan, *bounding box* menandai posisi nyala api yang terdeteksi. Akurasi prediksi ini penting untuk menentukan koordinat kebakaran, memperkirakan luas area terbakar, serta mendukung pengambilan keputusan dalam pengiriman bantuan atau *drone* pemadam.

### C. *Non-Max Suppresion* (NMS)

*Non-Maximum Suppression* (NMS) adalah algoritma yang digunakan untuk menghilangkan prediksi *bounding box* yang berlebihan dan tumpang tindih terhadap objek yang sama. Proses ini dilakukan dengan memilih *bounding box* dengan *confidence score* tertinggi, lalu menghilangkan box lain yang memiliki nilai *Intersection over Union* (*IoU*) lebih besar dari ambang batas tertentu (misalnya 0.5) terhadap box terpilih.

Secara formal, *IoU* dihitung dengan rumus:

$$IoU = \frac{\text{Luas tumpang tindih antara dua bounding box}}{\text{Luas gabungan bounding box tersebut}}$$

Tujuan utama *NMS* adalah menghindari duplikasi deteksi, yaitu saat satu objek api terdeteksi dengan beberapa *bounding box* yang tumpang tindih. Dalam deteksi kebakaran hutan, *NMS* penting untuk menyederhanakan informasi visual dan mencegah kesalahan estimasi jumlah titik api. Tanpa *NMS*, sistem dapat melaporkan lebih banyak titik api dari yang sebenarnya, sehingga berisiko menimbulkan alarm palsu (*false positive*) yang mengganggu pengambilan keputusan operasional.

#### D. Tensor Output YOLO

Dalam metode *YOLO*, hasil dari proses inferensi jaringan saraf konvolusional (*CNN*) tidak langsung berupa gambar atau koordinat yang dapat dilihat secara visual, melainkan berbentuk *tensor*, yaitu struktur data multidimensi yang menyimpan semua prediksi yang dihasilkan oleh model. *Tensor output* dari *YOLO* memuat informasi penting seperti *bounding box*, *confidence score*, dan probabilitas kelas untuk setiap prediksi yang dilakukan oleh sel *grid*.

Secara umum, *tensor output YOLO* memiliki dimensi:

$$S \times S (B \cdot 5 + C) \quad (1)$$

Keterangan:

- $S \times S$ : jumlah sel *grid* dalam gambar (contoh:  $13 \times 13$ )
- $B$ : jumlah *bounding box* yang diprediksi per sel (contoh: 3)
- 5: terdiri dari 4 parameter *bounding box* ( $x, y, w, h$ ) dan 1 *confidence score*
- $C$ : jumlah kelas objek (misalnya: api, asap, manusia, pohon, dll)

Setiap baris dalam *tensor* menyimpan hasil prediksi dari satu sel *grid*. Misalnya, jika *YOLOv5* digunakan dengan  $13 \times 13$  *grid*, 3 *bounding box* per sel, dan 1 kelas (Api), maka dimensi *tensor output*-nya adalah:

$$13 \times 13 (3 \cdot 5 + 1) = 13 \times 13 \times 16 \quad (2)$$

Dalam konteks pendeteksian kebakaran hutan, *tensor* ini menyimpan semua kemungkinan lokasi api yang terdeteksi, lengkap dengan ukuran area api, posisi dalam gambar, dan tingkat keyakinan bahwa objek tersebut adalah api. *Tensor output* ini akan diproses lebih lanjut oleh fungsi *Non-Max Suppression (NMS)* untuk menyaring prediksi terbaik.

*Tensor* memungkinkan sistem menyimpan seluruh prediksi secara efisien sebelum divisualisasikan atau dikirim ke pengendali. Perannya penting dalam *deployment model YOLO* di sistem *IoT*, seperti kamera *drone*, ESP32, atau Raspberry Pi, karena mempermudah integrasi antara deteksi visual dan sistem aksi otomatis, seperti menyalakan alarm, mengirim koordinat, atau mengaktifkan alat pemadam.

#### 2.3. Metode Perancangan

Metode perancangan ini mengintegrasikan kecerdasan buatan, sistem benam, dan *Internet of Things (IoT)* untuk mitigasi kebakaran hutan dan lahan (Karhutla). Cakupan metode meliputi perancangan model, pemilihan komponen, arsitektur sistem yang mendukung transmisi data *real-time*, panel surya sebagai sumber daya mandiri, sistem dirancang untuk beroperasi berkelanjutan di wilayah terpencil dan memberikan peringatan dini guna mengurangi dampak Karhutla.

#### 2.4. Model dan Spesifikasi

Bagian ini menjelaskan model sistem yang digunakan serta spesifikasi perangkat keras dan perangkat lunak yang membentuk keseluruhan alat. Pemilihan komponen didasarkan pada

kebutuhan sistem dalam menjalankan proses deteksi api secara akurat dan efisien. Setiap bagian, mulai dari kamera, unit pemroses (Raspberry Pi 4), hingga modul komunikasi *LoRa* pada bagian *transmitter* (pengirim) dan *receiver* (penerima), memiliki peran penting dalam mendukung kinerja sistem secara keseluruhan.

#### 2.4.1. Komponen Alat

Komponen alat yang digunakan dalam sistem ini ditunjukkan pada Tabel 1 dan Tabel 2.

**Tabel 1.** Komponen Transmitter (Pengirim)

Nama	Spesifikasi	Jumlah	Keterangan
<i>Mini PC</i> (Raspberry 4)	5 V DC (USB-C)	1	Komputer kecil sebagai pemroses utama untuk menjalankan algoritma <i>YOLO</i>
ESP32-S3	3.3V	1	Mikrokontroller sebagai penyalur data antara <i>Mini PC</i> ke <i>LoRa</i>
Kamera	5V (USB bus)	1	Mengambil Video untuk proses deteksi api berbasis <i>computer vision</i>
<i>LoRa</i> SX1278 (Transmitter)	3.3 V	1	Modul komunikasi untuk mengirim data secara nirkabel ke <i>receiver</i>
Panel Surya	~18 V Voc, 30–50 W	1	Sumber daya berbasis energi matahari
<i>Charge Controller</i>	Panel Voc ~18 V+; Output ~14.1 V (12 V system)	1	Mengatur pengisian baterai dari panel surya agar aman dan efisien
DHT 22	3.3–5 V	1	Mengukur suhu dan kelembaban lingkungan
Baterai 18650	14.8 V nominal; 16.8 V penuh	4	Menyimpan energi listrik untuk daya sistem

**Tabel 2.** Komponen Transmitter (Pengirim)

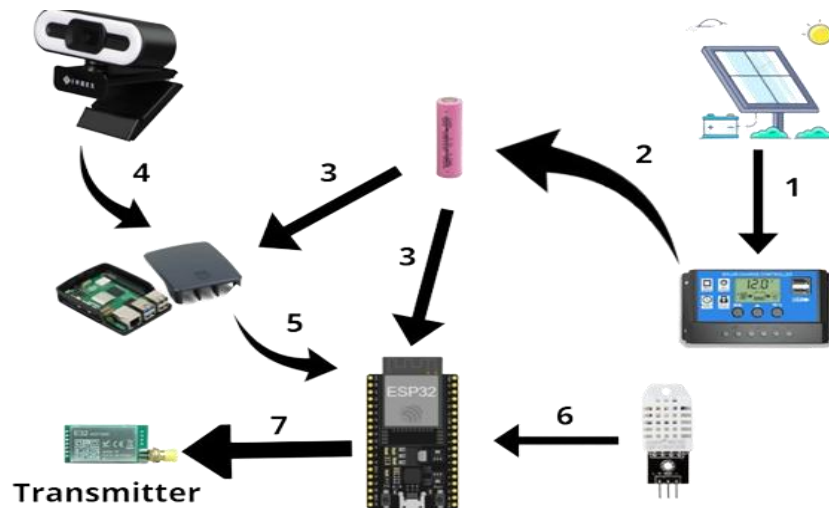
Nama	Spesifikasi	Jumlah	Keterangan
ESP32-S3	3.3V	1	Mikrokontroller sebagai pengelola data yang sudah diterima oleh <i>receiver</i> dan ditampilkan ke mobile
<i>LoRa</i> SX1278 (Receiver)	3.3 V	1	Modul komunikasi untuk menerima data secara nirkabel dari <i>transmitter</i>
<i>Adaptor</i> 5v	Out 5V DC, In 100-240V AC	1	Mengubah Arus menjadi 5v agar cocok dengan ESP32-S3 dan <i>LoRa</i>

Tabel 1 dan 2 menampilkan komponen sistem deteksi dan pengiriman data berbasis IoT. Raspberry Pi 4 berfungsi sebagai pemroses data visual dari kamera, sedangkan ESP32-S3 dan *LoRa* SX1278 mengirim dan menerima data nirkabel. Kamera menangkap gambar lingkungan, sensor DHT22 mencatat suhu dan kelembaban. Panel surya, *charge controller*, dan baterai 18650 menjadi sumber daya mandiri, sementara adaptor menyalakan perangkat di sisi penerima.

### 2.4.2. Skematik Alat

Untuk memastikan koneksi kelistrikan antar perangkat berjalan aman dan stabil, dibuat wiring diagram yang menunjukkan hubungan fisik antara kamera, Raspberry Pi 4, ESP32-S3, dan *LoRa*. Fokus utama ada pada kestabilan daya serta pengaturan jalur *input/output* agar sistem beroperasi optimal. Rangkaian *wiring* lengkap untuk *Transmitter* dan *Receiver* ditampilkan pada gambar berikut.

#### 2.4.2.1. Skematik Sistem Benam



Gambar 1. Skematik Sistem Benam Bagian Transmitter

Pada Gambar 1, menjelaskan alur dari skematik sistem benam bagian transmitter (Pengirim) yaitu:

1. Panel surya menghasilkan listrik dari cahaya matahari. Otomatisasi dan Keberlanjutan Energi dengan Memanfaatkan Energi Surya.
2. Listrik dari panel dialirkan ke *solar charge controller* untuk di manajemen.
3. Daya dari solar controller disimpan ke baterai dan menyuplai kamera, Raspberry Pi & ESP32-S3.
4. Kamera merekam dan mengirim data visual ke Raspberry Pi.
5. Raspberry Pi memproses gambar menggunakan *YOLO* dan mengirim ke ESP32-S3.
6. Sensor suhu & kelembaban mengirimkan data langsung ke ESP32.
7. ESP32-S3 mengirim semua data ke *LoRa Transmitter* untuk dikirim ke *Receiver*.

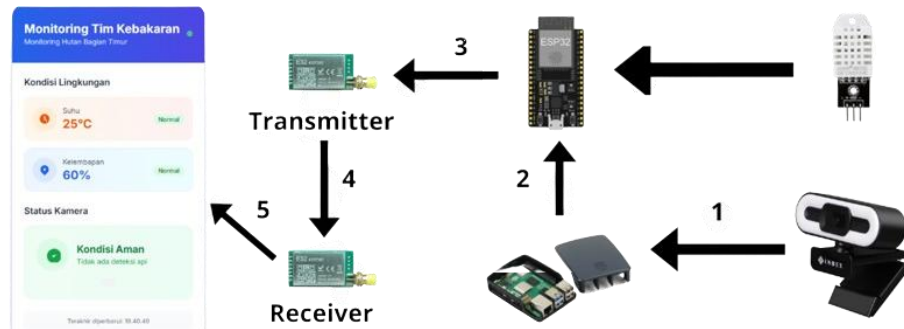


Gambar 2. Skematik Sistem Benam bagian Receiver

Pada Gambar 2, menjelaskan alur dari skematik sistem benam bagian *Receiver* (Penerima) yaitu:

1. Sumber Aliran listrik AC 220 V menjadi sumber daya untuk alat *receiver* (penerima).
2. Adaptor 5V mengubah Sumber listrik AC 220 V menjadi 5V.
3. ESP32-S3 menerima Sumber daya, *LoRa Receiver* ikut menerima sumber daya dan menerima data dari *Transmitter* dan ESP32-S3 mengelola data tersebut.

#### 2.4.2.2. Skematik Sistem *Internet of Things* (IoT)

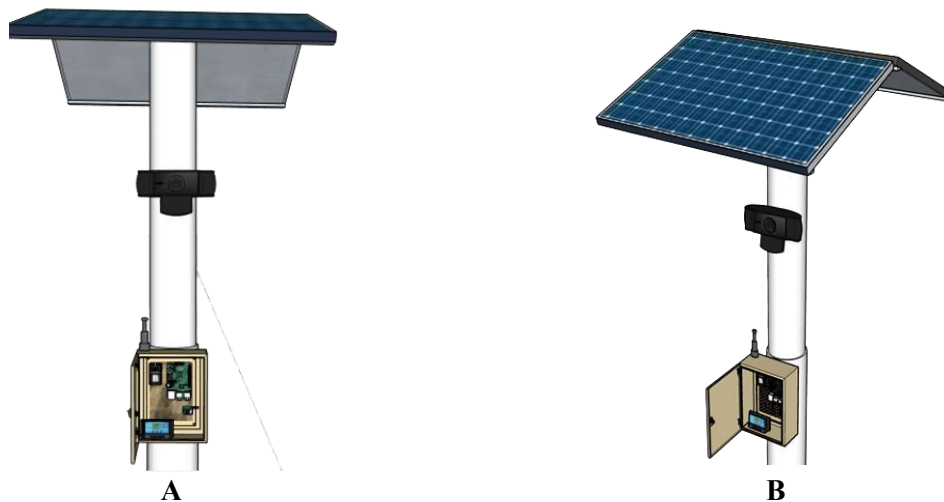


**Gambar 3.** Skematik Sistem *Internet of Things* (IoT) Transmitter dan Receiver

Pada Gambar 3, menjelaskan alur dari skematik sistem *Internet of Things* (IoT) yaitu:

1. Sensor DHT → ESP32 dan Kamera → Raspberry Pi untuk pengambilan data awal.
2. Raspberry Pi kirim hasil deteksi ke ESP32.
3. ESP32 proses semua data dan kirim ke *LoRa Transmitter*.
4. Data dikirim secara *wireless* ke *LoRa Receiver*.
5. *Mobile* menerima data dari *Receiver* dan menampilkannya serta memberikan notifikasi bila Api Terdeteksi.

#### 2.5. Desain 3D



**Gambar 4.** Desain 3D, Tampak Depan (A), Tampak Samping (B)

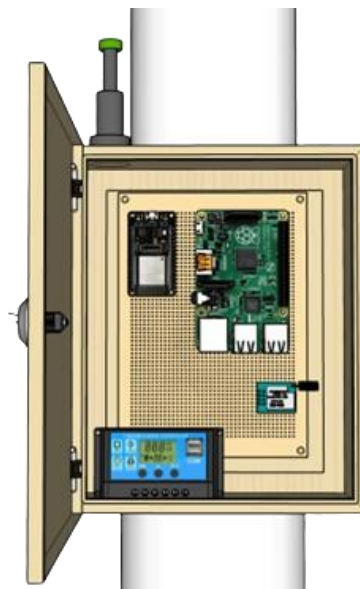
Gambar 4 menampilkan bentuk keseluruhan alat pendeteksi api yang dikembangkan. Alat ini terdiri dari kamera sebagai sensor utama, panel surya sebagai sumber daya mandiri, dan panel box yang menampung seluruh rangkaian elektronik. Desainnya disesuaikan untuk pemasangan di area terbuka seperti hutan atau lahan rawan kebakaran, dengan mempertimbangkan efisiensi energi dan ketahanan terhadap cuaca.





**Gambar 5.** Desain 3D Kamera

Gambar 5 menampilkan kamera yang digunakan untuk menangkap kondisi lingkungan secara real-time. Kamera ini menjadi komponen utama dalam proses deteksi api, karena video yang diambil akan diproses oleh algoritma *YOLOv5* untuk mengidentifikasi keberadaan api secara cepat dan akurat.



**Gambar 6.** Desain 3D Panel Box & Komponen Utama Lainnya

Pada gambar 6 diperlihatkan bagian dalam panel box yang berisi komponen inti sistem, yaitu Raspberry Pi 4 yang menjalankan *model AI YOLOv5*, modul ESP32-S3 untuk komunikasi serial, modul *LoRa* sebagai pengirim data jarak jauh, dan *charge controller* yang mengelola aliran daya dari panel surya. Seluruh komponen ini dirancang agar dapat mendukung sistem deteksi api secara mandiri dan *real-time*.

## 2.6. Analisis Sistem

Sistem deteksi kebakaran konvensional berbasis sensor suhu atau asap memiliki keterbatasan kecepatan dan akurasi, terutama pada tahap awal kebakaran. Akibatnya, peringatan sering terlambat sehingga meningkatkan risiko kerugian. Pendekatan berbasis *computer vision* dan *IoT* menjadi solusi menjanjikan karena mampu mengenali tanda visual kebakaran secara real-time

dengan jangkauan lebih luas. Algoritma *You Only Look Once (YOLO)* unggul karena mendeteksi objek dalam satu tahap pemrosesan, menghasilkan deteksi cepat dan presisi tinggi.

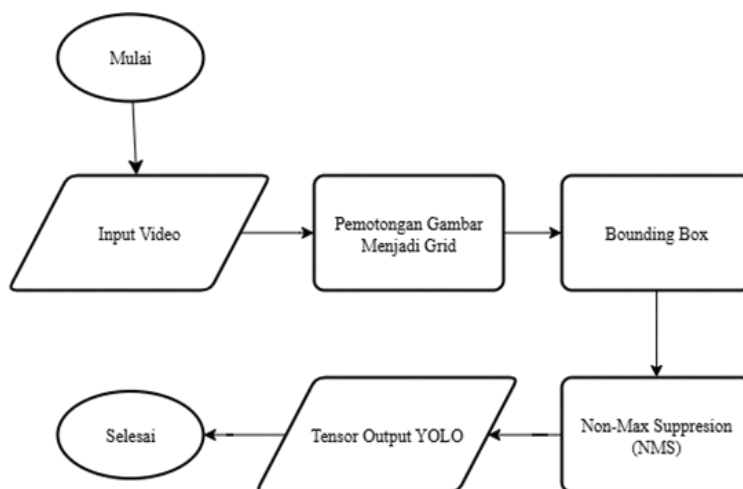
Bahhar melaporkan bahwa model *YOLOv8* mencapai presisi 95,7% dengan kemampuan generalisasi baik [20]. Firdaus mengintegrasikan *YOLOv8* dengan *Blynk* dan Telegram, meningkatkan kecepatan serta keandalan notifikasi real-time dengan mAP50 sebesar 97% [21]. Sistem ini menggabungkan kamera sebagai sensor visual, *mini PC* untuk pemrosesan *YOLO*, dan mikrokontroler ESP32-S3 untuk pengiriman data melalui *IoT*. Kombinasi deteksi *real-time* dan *IoT* ini memberikan solusi deteksi kebakaran yang cepat, adaptif, dan responsif.

### 3. Metode Dasar Pengembangan

Metode Pengembangan Dasar menjelaskan cara kerja sistem secara menyeluruh, mulai dari akuisisi data hingga penyampaian informasi ke pengguna. Sistem ini menggabungkan *AI (YOLOv5)* untuk deteksi api menggunakan kamera, sistem benam berbasis Raspberry Pi 4 dan Mikrokontroler ESP32-S3 untuk pemrosesan lokal, serta modul *LoRa* sebagai bagian dari *IoT* untuk mengirim hasil deteksi ke perangkat penerima secara *real-time*.

#### 3.1. Artificial Intelligence (AI)

Pada bagian ini, *Flowchart* menggambarkan alur kerja sistem *Artificial Intelligence (AI)* dalam mendeteksi api secara otomatis. Proses dimulai dari input video yang dibagi menjadi *grid* dan ditangkap oleh kamera pengawas di area hutan. Data visual kemudian diproses menggunakan algoritma *YOLO*, model *deep learning* berbasis *object detection* yang mendeteksi api secara cepat dan akurat. Setiap *frame* dianalisis untuk menentukan keberadaan api; jika terdeteksi, sistem menandainya dengan *bounding box* serta menampilkan hasil klasifikasi dan *confidence score*. Hasil deteksi selanjutnya dikirim ke sistem benam sebagai dasar pengambilan keputusan otomatis.



Gambar 7. Flowchart Artificial Intelligence (AI)

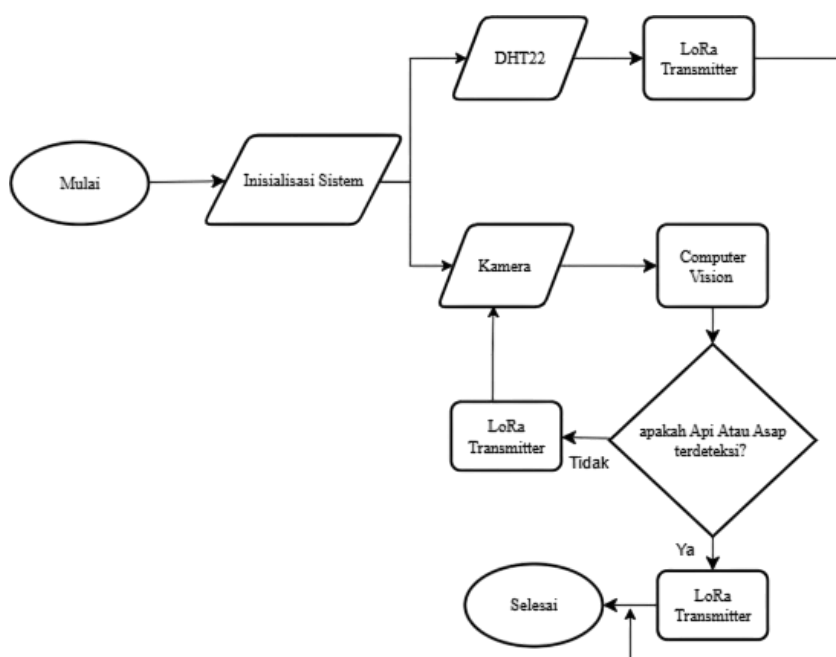
Pada gambar 7, Menjelaskan alur dari *Flowchart Artificial Intelligence (AI)* yaitu sebagai berikut:

1. Mulai - Sistem mengatur semua perangkat keras agar siap digunakan, termasuk sensor, kamera, dan komunikasi *LoRa*.
2. *Input Video* - Video dapat ditangkap secara *real-time* melalui kamera, kemudian setiap *frame* dianalisis satu per satu.

3. Pemotongan Gambar menjadi *Grid* - Setiap *frame* gambar dibagi menjadi *grid-grid* kecil (misalnya 13x13 atau 19x19, tergantung versi *YOLO*), di mana setiap *grid* bertanggung jawab mendeteksi objek yang berada di dalamnya.
4. *Bounding Box* - Untuk setiap *grid*, *YOLO* memprediksi beberapa *bounding box* yang menunjukkan lokasi objek, masing-masing dilengkapi dengan *confidence score* (tingkat keyakinan model terhadap keberadaan objek), koordinat posisi, dan kelas objek seperti api atau asap.
5. *Non-Max Suppression (NMS)* - *YOLO* biasanya menghasilkan banyak *bounding box* yang saling tumpang tindih, sehingga digunakan metode *Non-Maximum Suppression (NMS)* untuk menghapus duplikasi dan mempertahankan *box* dengan *confidence score* tertinggi.
6. *Tensor Output YOLO* - Model menghasilkan *output* akhir berupa *tensor* yang memuat koordinat objek, *confidence score*, dan *class label* (“Api” atau “Asap”). Data ini kemudian dapat dikirim ke sistem lain, seperti ESP32-S3, untuk diproses lebih lanjut.
7. Selesai - Proses satu *frame* selesai, dilanjutkan ke *frame* berikutnya.

### 3.2. Sistem Benam

Diagram alir (*Flowchart*) sistem benam menggambarkan alur kerja perangkat utama yang bertugas mendeteksi keberadaan api di lokasi hutan secara mandiri. Sistem ini terdiri dari kamera sebagai sensor input, Raspberry Pi 4 sebagai pemroses utama, serta algoritma *YOLOv5* yang bertugas menangkap citra terhadap objek api maupun asap. Setelah gambar dari kamera diproses, sistem akan mengambil keputusan berdasarkan hasil deteksi: apakah terdapat indikasi kebakaran atau tidak. Jika terdeteksi adanya api, informasi ini akan dikirimkan melalui modul *LoRa* yang nantinya dibuat sebagai bentuk output data ke sistem monitoring. Seluruh proses ini dilakukan secara lokal tanpa keterlibatan jaringan internet, sehingga termasuk dalam kategori sistem benam yang bersifat cerdas dan *real-time*.

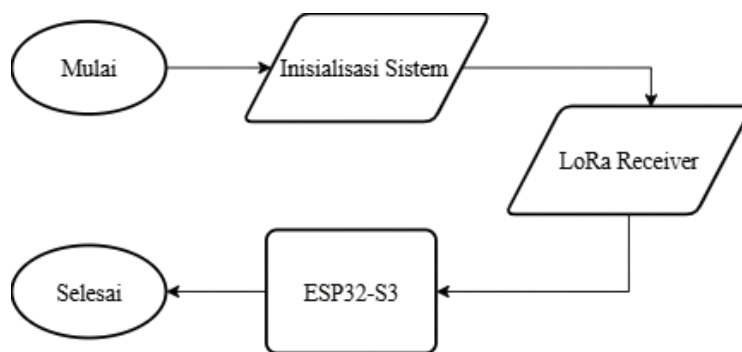


Gambar 8. Flowchart Sistem Benam Transmitter (Pengirim)

Pada Gambar 8, menjelaskan alur dari Flowchart sistem benam bagian Transmitter (Pengirim) yaitu:

Berikut adalah alur dari Flowchart Sistem Benam Transmitter (Pengirim) :

1. Inisialisasi Sistem - Sistem mengatur semua perangkat keras agar siap digunakan, termasuk sensor, kamera, dan komunikasi *LoRa*.
2. Pada Jalur atas, Pembacaan Sensor DHT22 - Sensor DHT membaca suhu & kelembaban udara di lingkungan sekitar
3. Kirim data DHT 22 ke *LoRa Transmitter* - Data suhu & kelembaban dikirim secara nirkabel melalui modul *LoRa Transmitter*.
4. Jalur Bawah, Kamera mengambil video - Kamera mengambil video area yang dipantau untuk dianalisis menggunakan *Computer Vision*
5. Proses *Computer Vision* - Sistem pengolahan gambar menganalisis hasil dari kamera menggunakan model *AI YOLOv5* untuk mendeteksi api atau asap
6. Deteksi Api/Asap - *Decision node* menentukan apakah terdapat api atau asap. Jika tidak terdeteksi, data tetap dikirim melalui *LoRa transmitter* dan sistem kembali melakukan deteksi dengan kamera. Jika terdeteksi, data hasil deteksi juga dikirim melalui *LoRa transmitter*.
7. Selesai - Proses selesai, sistem siap untuk menjalankan pemantauan berikutnya.



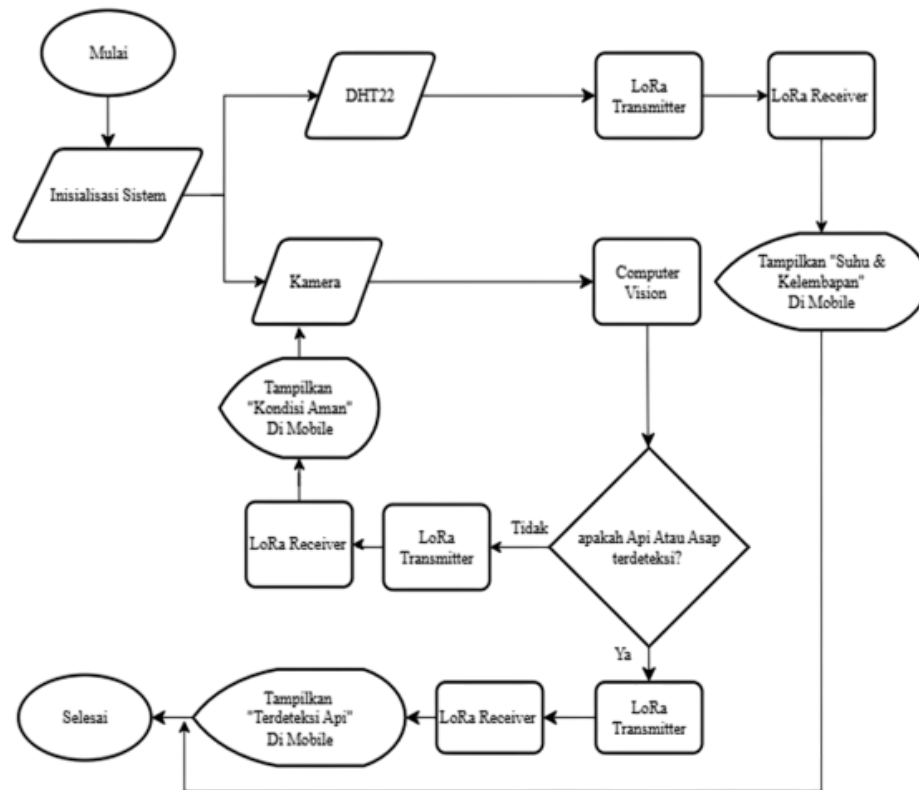
**Gambar 9.** Flowchart Sistem Benam Receiver (Penerima)

Pada Gambar 9, menjelaskan alur dari *Flowchart* sistem benam bagian *Receiver* (Penerima) yaitu:

1. Inisialisasi Sistem - Sistem mengatur semua perangkat keras agar siap digunakan, termasuk sensor, kamera, dan komunikasi *LoRa*.
2. Menerima data dari *LoRa Receiver*
3. Memproses data dari *LoRa Receiver*
4. Selesai - Proses selesai, sistem siap untuk menjalankan pemantauan berikutnya.

### 3.3. Internet of Things (IoT)

Perancangan sistem deteksi kebakaran berbasis *IoT* dan *computer vision* dilakukan dengan mempertimbangkan integrasi antara perangkat keras, perangkat lunak, serta antarmuka pengguna agar sistem mampu bekerja secara optimal dan responsif terhadap kondisi darurat. Proses perancangan dimulai dengan pengembangan diagram alir (*flowchart*) yang menggambarkan logika kerja sistem dari awal hingga akhir. Diagram ini menunjukkan bahwa sistem akan melakukan inisialisasi, menangkap citra secara real-time menggunakan kamera, memproses citra dengan algoritma *YOLO*, dan jika api terdeteksi, sistem akan mengirimkan data serial lewat *LoRa* menuju receiver dan memberikan notifikasi serta memperbarui dashboard mobile. Jika tidak terdeteksi api, sistem kembali melakukan pemantauan secara berulang (*looping*) tanpa intervensi manual. Struktur ini digambarkan secara lengkap pada Gambar 10.



**Gambar 10.** Flowchart Internet of Things (IoT)

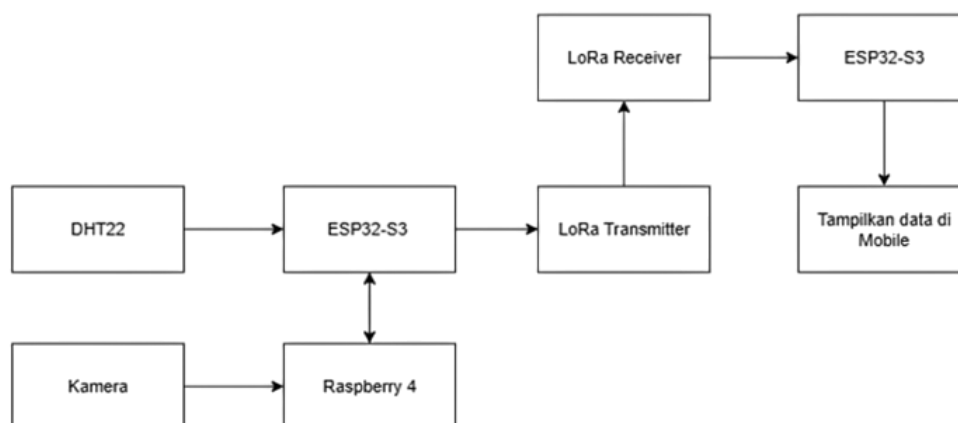
Pada Gambar 10, menjelaskan alur dari Flowchart sistem *Internet of Things (IoT)* yaitu:

1. Inisialisasi Sistem - Sistem mengatur semua perangkat keras dan perangkat lunak agar siap digunakan, termasuk sensor, kamera, dan komunikasi *LoRa*.
2. Pada Jalur atas, Pembacaan Sensor DHT22 - Sensor DHT membaca suhu & kelembaban udara di lingkungan sekitar
3. Kirim data DHT 22 ke *LoRa Transmitter* - Data suhu & kelembaban dikirim secara nirkabel melalui modul *LoRa Transmitter*.
4. Data diterima oleh *LoRa Receiver*
5. Tampilkan “Suhu & Kelembapan” di *mobile*
6. Jalur Bawah, Kamera mengambil video - Kamera mengambil video area yang dipantau untuk dianalisis menggunakan *Computer Vision*.
7. Proses *Computer Vision* - Sistem pengolahan gambar menganalisis hasil dari kamera menggunakan model *AI YOLOv5* untuk mendeteksi api atau asap.
8. Deteksi Api/Asap - Decision node menentukan apakah terdapat api atau asap. Jika tidak terdeteksi, data “aman” dikirim melalui *LoRa transmitter*, diterima oleh *LoRa receiver*, dan aplikasi *mobile* menampilkan status “Kondisi Aman” sebelum sistem kembali melakukan deteksi dengan kamera. Jika terdeteksi, data “deteksi” dikirim melalui *LoRa transmitter*, diterima oleh *LoRa receiver*, dan aplikasi *mobile* menampilkan status “Terdeteksi Api.”
9. Selesai - Proses selesai, sistem siap untuk menjalankan pemantauan berikutnya.

### 3.4. Kesimpulan Metode Pengembangan

Selanjutnya, blok diagram digunakan untuk menyimpulkan keterhubungan antar komponen utama, baik input, pemrosesan, maupun output. Kamera terhubung ke *Mini PC* yang menjalankan *YOLO* sebagai pusat pemrosesan citra. *Output* dari *Mini PC* dikirimkan ke mikrokontroler ESP32-

S3 untuk mengirimkan data terkait tampilan di *mobile* lewat *LoRa*. Hubungan antar komponen ini divisualisasikan dalam Gambar 11.



**Gambar 11.** Blok Diagram Kesimpulan

Pada Gambar 11, menjelaskan alur dari Blok Diagram kesimpulan yaitu sebagai berikut:

1. DHT 22 & Kamera - DHT 22 dan Kamera disini berfungsi sebagai “mata” dari alat ini, DHT dijadikan sebagai pemberi validasi peringatan paling darurat ketika api sudah mendekat pada alat dan Kamera sebagai pemberi data visual yang akan diproses lebih lanjut untuk jarak jauh.
2. Raspberry 4 & ESP32-S3 - Pada proses selanjutnya, Raspberry 4 berfungsi sebagai “otak” dari alat ini, Raspberry 4 menggunakan model *YOLOv5* untuk mendeteksi keberadaan api dari gambar/citra yang diambil dan ESP32-S3 berperan sebagai pengambil keputusan atas dasar informasi yang diberikan oleh Raspberry dan menyalurkan data tersebut.
3. *LoRa (Long Range) Transmitter & Receiver* - Setelah proses data dilakukan oleh raspberry dan ESP32, Maka data tersebut akan dikirim melewati *LoRa Transmitter* (Pengirim) kemudian sampai ke *LoRa Receiver* (Penerima).
4. ESP32-S3 Bagian *Receiver* - Proses Terakhir yaitu data yang telah diterima oleh *LoRa Receiver* maka akan diolah lebih lanjut oleh ESP32-S3 bagian *Receiver* lebih lanjut untuk ditampilkan ke user sebagai monitoring kebakaran hutan.

#### 4. Kesimpulan

Penelitian ini berhasil mencapai tujuan yang dirumuskan pada Bab Pendahuluan, yakni merancang konsep sistem peringatan dini kebakaran hutan dan lahan (Karhutla) berbasis *Artificial Intelligence of Things (AIoT)* yang terintegrasi dengan komunikasi *LoRa* dan sumber energi surya. Sistem ini menggabungkan deteksi visual berbasis algoritma *YOLOv5* dengan mikrokontroler ESP32-S3 dan *mini PC* Raspberry Pi 4 sebagai pusat pemrosesan data. Hasil kajian menunjukkan bahwa rancangan sistem secara konseptual mampu memberikan peringatan dini secara cepat, akurat, dan mandiri di wilayah terpencil tanpa bergantung pada jaringan listrik maupun internet konvensional. Dengan demikian, penelitian ini telah memenuhi tujuannya, yaitu menghadirkan solusi teknologi cerdas yang adaptif dan berkelanjutan untuk mitigasi Karhutla.

Prospek pengembangan dari penelitian ini masih terbuka luas, mengingat saat ini sistem baru direalisasikan dalam bentuk kajian konseptual dan naskah ilmiah, belum dalam bentuk prototipe alat fisik. Oleh karena itu, langkah pengembangan selanjutnya difokuskan pada implementasi dan pengujian alat secara nyata agar dapat dievaluasi kinerjanya di lapangan. Selain itu, sistem dapat dikembangkan lebih lanjut dengan mengadopsi versi algoritma *YOLO* yang lebih baru dan akurat, seperti *YOLOv8*, untuk meningkatkan kemampuan deteksi terhadap variasi kondisi lingkungan

dan pencahayaan. Pengembangan tersebut diharapkan dapat menjadikan sistem ini sebagai solusi efektif dan inovatif dalam mendukung strategi nasional penanggulangan kebakaran hutan dan lahan di Indonesia.

### Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada semua pihak yang telah berpartisipasi dalam penelitian ini, baik sebagai penyedia fasilitas, narasumber, maupun rekan-rekan yang turut mendukung dan membantu dalam perancangan penelitian ini.

### Referensi

- [1] Kepala BPBD Provinsi Kepulauan Babel Mikron. BPBD: Kerugian karhutla di Babel capai Rp150 miliar. Antaranews. 2023. Accessed August 10, 2025. <https://www.antaranews.com/berita/3803976/bpbd-kerugian-karhutla-di-babel-capai-rp150-miliar>
- [2] Ramdani C. Jangan Anggap Remeh, Inilah Bahaya Karhutla di Kalimantan Selatan. jejakrekam. 2023. Accessed August 10, 2025. <https://jejakrekam.com/2023/09/13/jangan-anggap-remeh-inilah-bahaya-karhutla-di-kalimantan-selatan>
- [3] Sipongi. Rekapitulasi Luas Kebakaran Hutan dan Lahan (Ha) per Provinsi di Indonesia. <https://sipongi.menlhk.go.id/>. 2023. Accessed August 10, 2025. <https://sipongi.menlhk.go.id/indikasi-luas-kebakaran>
- [4] AHAMAD FAJAR. Sinergitas Instansi Pemerintah daerah dalam penanggulangan kebakaran hutan dan lahan di Kabupaten Okan hilir Propinsi Riau Pada Tahun 2015. Skripsi. Universitas Riau. 2015.
- [5] MARGARETA C, & NUGROHO AGUNG, H. Prototipe Sistem Peringatan Dini Kebakaran Hutan Berbasis Parameter Cuaca. Wahana Fisika. Vol 2 Nomor 2. 2017
- [6] BADRI MUHAMMAD, LUBIS P DJUARA, JOKO SUSANTO & SUHARJITO DIDIK. Sistem Komunikasi dini Pencegahan Kebakaran Hutan dan Lahan di Provinsi Riau. Jurnal PIKOM Vol 19 Nomor 1 Juni 2018
- [7] H. H. and R. Candra, "Sistem Deteksi Kebakaran Berbasis Mikrokontroler dan Aplikasi Peta Menggunakan IoT," Jurnal Teknologi dan Sistem Komputer, vol. 10, no. 2, pp. 123–130, 2021, doi: 10.14710/jtsiskom.10.2.123-130
- [8] A. R. Z. Zidifaldi and S. Nugroho, "Sistem Deteksi Dini Kebakaran Menggunakan Sensor Api dan Suhu Berbasis Arduino dengan Notifikasi Blynk," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 7, no. 3, pp. 210–217, 2022, doi: 10.25126/jtiik.2022.7.3.210.
- [9] A. S. P. I. Widharma and M. F. Hidayat, "Deteksi Api Menggunakan Algoritma YOLO Berbasis Computer Vision," Jurnal Informatika, vol. 9, no. 1, pp. 45–52, 2022, doi: 10.31294/ji.v9i1.12345.
- [10] S. P. and D. Kurniawan, "Implementasi Deep Learning untuk Deteksi Kebakaran pada Lingkungan Industri," Jurnal Teknologi dan Rekayasa, vol. 8, no. 2, pp. 89–96, 2023, doi: 10.54321/jtek.v11i4.4567.
- [11] L. S. and M. Yusuf, "Pengembangan Sistem Pendeteksi Kebakaran Berbasis IoT dengan Sensor MQ-2," Jurnal Teknik Elektro dan Komputer, vol. 11, no. 4, pp. 301–308, 2021, doi: 10.54321/jtek.v11i4.4567.
- [12] R. S. A. Nugraha and T. H. Santoso, "Sistem Monitoring Kebakaran Hutan Menggunakan Drone dan YOLOv5," Jurnal Penginderaan Jauh, vol. 5, no. 2, pp. 112–119, 2022, doi: 10.98765/jpj.v5i2.2345
- [13] D. L. and B. Wicaksono, "Integrasi Sensor Suhu dan Kamera untuk Deteksi Dini Kebakaran," Jurnal Sistem Informasi, vol. 14, no. 1, pp. 55–62, 2023, doi: 10.24680/jsi.v14i1.7890.

- [14] R. Hidayat, "Penerapan YOLOv4 untuk Deteksi Api pada Sistem Keamanan Rumah," *Jurnal Keamanan Siber*, vol. 3, no. 3, pp. 150–157, 2022, doi: 10.13579/jks.v3i3.3210.
- [15] M. A. R. and S. Dewi, "Sistem Deteksi Kebakaran Otomatis Menggunakan Raspberry Pi dan Kamera," *Jurnal Teknologi Informasi*, vol. 6, no. 2, pp. 200–207, 2021, doi: 10.24680/jti.v6i2.6543.
- [16] T. S. and L. Widodo, "Analisis Kinerja Sistem Deteksi Kebakaran Berbasis IoT di Gedung Perkantoran," *Jurnal Rekayasa Sistem*, vol. 9, no. 1, pp. 33–40, 2023, doi: 10.12345/jrs.v9i1.9876.
- [17] E. W. and F. H. Prasetyo, "Pengembangan Aplikasi Mobile untuk Monitoring Sistem Deteksi Kebakaran Berbasis IoT," *Jurnal Aplikasi Teknologi*, vol. 7, no. 3, pp. 145–152, 2022, doi: 10.54321/jat.v7i3.4321.
- [18] B. S. and Y. Kurniawan, "Implementasi YOLOv5 untuk Deteksi Api pada Lingkungan Rumah Tangga," *Jurnal Teknologi dan Sistem Informasi*, vol. 10, no. 2, pp. 75–82, 2023, doi: 10.24680/jtsi.v10i2.5678.
- [19] Waisnawa, Putu Gema Bujangga, Novia Mawar Sari, and Agus Bambang Irawan. "Analisis Bahaya Kebakaran Hutan dan Lahan di Jalur Pendakian Gunung Merbabu, Gunung Sindoro dan Gunung Sumbing, Jawa Tengah." *Jurnal Ilmiah Lingkungan Kebumian* 5.2 (2023): 75-83.
- [20] C. B. et al., "A Lightweight Fire Detection Algorithm Based on the Improved YOLOv8 Model," *Applied Sciences*, vol. 14, no. 16, p. 6878, 2024, [Online]. Available: <https://www.mdpi.com/2076-3417/14/16/6878>
- [21] M. A. F. et al., "Prototype smart integrated fire detection based on deep learning YOLO v8 and IoT (internet of things) to improve early fire detection," *International Journal of Applied Mathematics, Sciences, and Technology for National Defense*, vol. 3, no. 2, pp. 53–58, 2023, [Online]. Available: <https://journal.foundae.com/index.php/JASND/article/view/437>