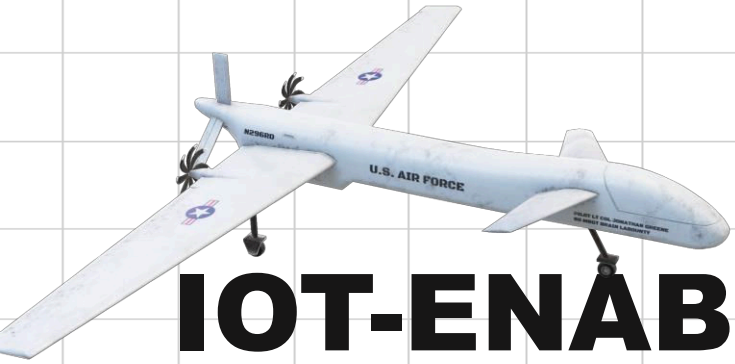


IOT-ENABLED AUTONOMOUS SYSTEM COLLABORATION FOR DISASTER-AREA MANAGEMENT

Disusun Oleh

Anif Maulana R

2042231074

JUDUL YANG DIANGKAT

Dirancang untuk manajemen daerah bencana. Tujuan utama dari sistem ini adalah memanfaatkan teknologi IoT dan sistem otonom untuk mempercepat dan meningkatkan efektivitas dalam penanganan bencana. Dengan memanfaatkan sensor, perangkat pintar, dan robot otonom, sistem ini dapat membantu mengurangi risiko bagi pekerja kemanusiaan dan meningkatkan peluang keselamatan bagi para korban bencana. Teknologi ini memungkinkan kolaborasi yang cepat dan efektif antara berbagai perangkat dan sistem yang saling terhubung untuk memberikan respons yang lebih tepat waktu.

LATAR BELAKANG

Latar belakang ini ada dari tantangan besar yang dihadapi dalam operasi tanggap darurat pascabencana, seperti gempa bumi, banjir, kebakaran, kebocoran bahan kimia, atau ledakan. Situasi pascabencana menuntut tim penyelamat untuk segera terjun ke lapangan guna menyelamatkan korban, memitigasi dampak bahaya, dan meminimalisir kerusakan properti. Namun, kondisi lingkungan yang berubah drastis akibat bencana sering kali menyulitkan tim penyelamat dalam menilai keadaan, serta mengumpulkan informasi secara cepat dan aman. Selain itu, sumbersumber bahaya seperti kebocoran bahan kimia atau emisi panas yang berlebihan harus segera dikenali untuk menghindari kerugian lebih lanjut. Seiring dengan perkembangan teknologi, pemanfaatan kendaraan nirawak (Unmanned Vehicles - UV), seperti UAV (Unmanned Aerial Vehicle) dan UGV (Unmanned Ground Vehicle), menjadi semakin penting dalam membantu operasi pencarian dan penyelamatan di daerah bencana. Kendaraan-kendaraan ini bekerja secara otomatis untuk memantau area bencana, mengumpulkan data lingkungan, dan berkolaborasi dengan tim penyelamat tanpa membahayakan nyawa manusia

RUMUSAN MASALAH

1. Bagaimana cara menciptakan kolaborasi yang efektif antara UAV, UGV, dan tim penyelamat dalam operasi pencarian dan penyelamatan pascabencana?
2. Bagaimana meningkatkan efisiensi komunikasi antar sistem nirawak dan operator manusia di area bencana yang memiliki keterbatasan infrastruktur jaringan?
3. Bagaimana mengatasi keterbatasan daya dan cakupan jaringan pada UAV di area bencana?
4. Bagaimana memastikan sistem yang diusulkan dapat bekerja dengan efektif dalam berbagai skenario bencana yang dinamis dan kompleks?

DAMPAK SIGNIFIKAN

Efisiensi Operasional dalam Manajemen Bencana

Sistem kolaborasi otonom berbasis IoT yang melibatkan kendaraan darat tak berawak (UGV) dan kendaraan udara tak berawak (UAV) membawa perubahan signifikan dalam manajemen area bencana. Teknologi ini mampu mempercepat proses evakuasi dan penyelamatan korban dengan memungkinkan penilaian cepat terhadap area bencana tanpa memerlukan intervensi manusia langsung. Sistem ini mengurangi risiko bagi penanggap pertama serta mempercepat pengumpulan data penting seperti keberadaan korban dan sumber bahaya, seperti kebocoran bahan kimia atau nuklir, yang sebelumnya menimbulkan risiko besar bagi tim penyelamat manusia.

Peningkatan Jangkauan Komunikasi di Area Bencana

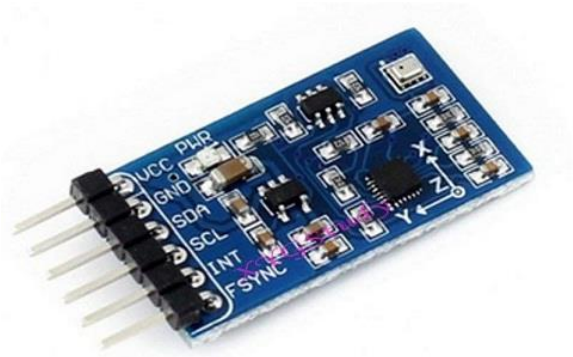
Area yang tidak terjangkau jaringan dapat tetap terhubung. Hal ini menjadi sangat penting ketika jaringan telekomunikasi konvensional lumpuh akibat bencana alam, sehingga UAV dapat mempertahankan konektivitas antar sistem dan memudahkan koordinasi penyelamatan, karena penggunaan UAV sebagai titik relay jaringan memperluas jangkauan komunikasi ke daerah daerah yang terpencil atau rusak akibat bencana. Dengan adanya UAV yang dilengkapi teknologi komunikasi line-of-sight (LoS).

JENIS SENSOR

Sensor Ultrasonik

JENIS SENSOR

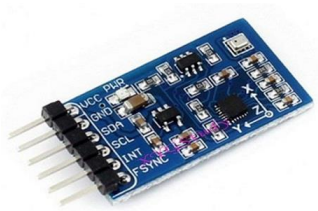
Sensor Ultrasonik



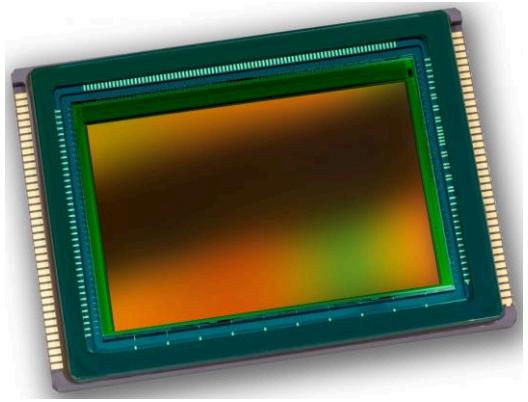
Sensor IMU

JENIS SENSOR

Sensor Ultrasonik



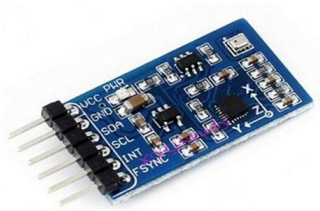
Sensor IMU



Sensor Camera

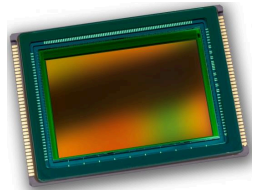
JENIS SENSOR

Sensor Ultrasonik



Sensor IMU

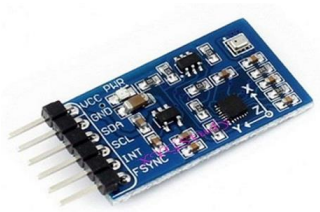
Sensor Sonar



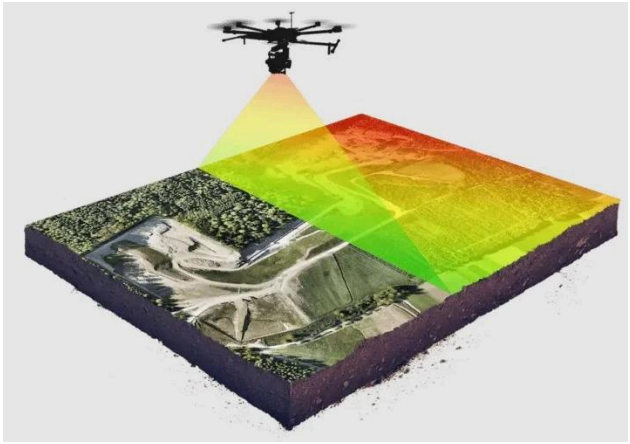
Sensor Camera

JENIS SENSOR

Sensor Ultrasonik

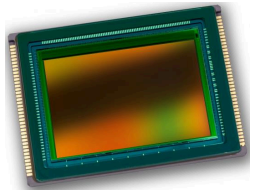


Sensor IMU



Sensor Lidar

Sensor Sonar



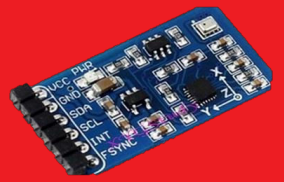
Sensor Camera

MODEL MATEMATIS

$$d = \frac{v \cdot t}{2}$$

Sensor Ultrasonik

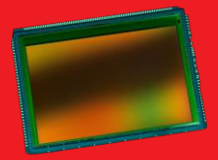
MODEL MATEMATIS



$$\theta(t) = A \cdot \sin(2\pi f t + \phi)$$

Sensor IMU

MODEL MATEMATIS



$$\text{Frame}(t) = A \cdot \sin(2\pi f t + \phi)$$

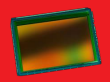
Sensor Camera

MODEL MATEMATIS

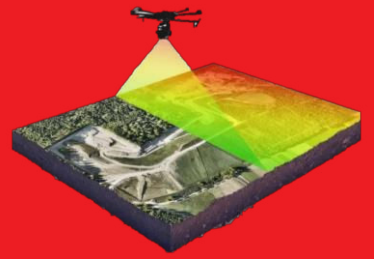
$$d = \frac{c \cdot t}{2}$$

Sensor Sonar



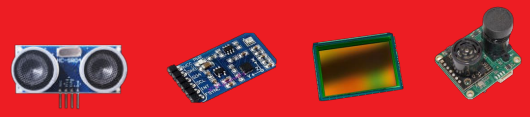


MODEL MATEMATIS



$$d = \frac{v \cdot t}{2}$$

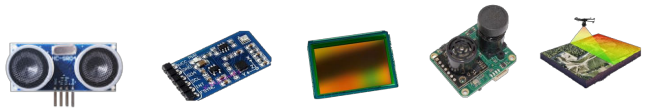
Sensor Lidar



CODINGAN PYTHON

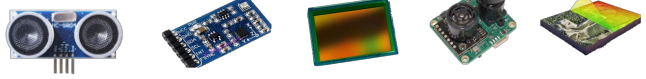
```python
mport tkinter as tk
from tkinter import ttk
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import numpy as np

# Fungsi matematis sensor
def generate_signal(sensor_type, amplitude=1, frequency=1, sampling_rate=1000, duration=2, phase=0):
    t = np.arange(0, duration, 1/sampling_rate)
    if sensor_type == 'Ultrasonic':
        speed_of_sound = 343  # m/s
        signal = amplitude * np.sin(2 * np.pi * frequency * t + phase)
        distance = speed_of_sound * t / 2
        return t, signal, distance
    elif sensor_type == 'IMU':
        angle = amplitude * np.sin(2 * np.pi * frequency * t + phase)
        return t, angle
    elif sensor_type == 'Camera':
        frames = amplitude * np.sin(2 * np.pi * frequency * t + phase)
        return t, frames
    elif sensor_type == 'LIDAR':
        speed_of_light = 3e8  # m/s
        signal = amplitude * np.sin(2 * np.pi * frequency * t + phase)
        distance = speed_of_light * t / 2
        return t, signal, distance
    elif sensor_type == 'SONAR':
        speed_of_sound = 1500  # m/s (in water)
        signal = amplitude * np.sin(2 * np.pi * frequency * t + phase)
        distance = speed_of_sound * t / 2
        return t, signal, distance
    return t, []

```

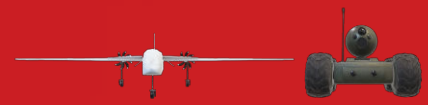
```python
# Fungsi untuk memperbarui plot berdasarkan sensor yang dipilih
def update_plot(sensor_type):
    fig.clear()
    ax = fig.add_subplot(111)

    if sensor_type == 'Ultrasonic':
        t, signal, distance = generate_signal(sensor_type, frequency=40000)
        ax.plot(t, signal, label='Ultrasonic Signal')
        ax.plot(t, distance, label='Distance (m)', linestyle='--')
    elif sensor_type == 'IMU':
        t, angle = generate_signal(sensor_type, frequency=100)
        ax.plot(t, angle, label='IMU Angle (deg)')
    elif sensor_type == 'Camera':
        t, frames = generate_signal(sensor_type, frequency=30)
        ax.plot(t, frames, label='Camera Frames')
    elif sensor_type == 'LIDAR':
        t, signal, distance = generate_signal(sensor_type, frequency=20000)
        ax.plot(t, signal, label='LIDAR Signal')
        ax.plot(t, distance, label='Distance (m)', linestyle='--')
    elif sensor_type == 'SONAR':
        t, signal, distance = generate_signal(sensor_type, frequency=5000)
        ax.plot(t, signal, label='SONAR Signal')
        ax.plot(t, distance, label='Distance (m)', linestyle='--')

    ax.set_title(f'{sensor_type} Sensor Signal')
    ax.set_xlabel('Time (s)')
    ax.set_ylabel('Amplitude / Distance')
    ax.legend()
    ax.grid(True)
    canvas.draw()

```



CODINGAN PYTHON

```python
# Inisialisasi GUI
root = tk.Tk()
root.title("Sensor Signal Visualizer")

# Figure untuk matplotlib
fig = plt.Figure(figsize=(8, 5))
canvas = FigureCanvasTkAgg(fig, master=root)
canvas_widget = canvas.get_tk_widget()
canvas_widget.pack(side=tk.RIGHT, fill=tk.BOTH, expand=1)

# Frame untuk tombol
frame = ttk.Frame(root)
frame.pack(side=tk.LEFT, fill=tk.Y, padx=20, pady=20)  # Isi sepanjang vertikal (fill=tk.Y)

# Tambahkan label judul di atas tombol
label = tk.Label(frame, text="Pilih Sensor", font=("Arial", 14, "bold"))
label.pack(pady=10)

# Gaya untuk tombol
style = ttk.Style()
style.configure("TButton", font=("Arial", 12), anchor="center", padding=(5, 10))

# Tombol untuk setiap sensor
sensors = ['Ultrasonic', 'IMU', 'Camera', 'LIDAR', 'SONAR']
for sensor in sensors:
    btn = ttk.Button(frame, text=sensor, style="TButton", command=lambda s=sensor: update_plot(s))
    btn.pack(pady=10, fill="x")  # Isi horizontal tombol untuk lebar konsisten

# Jalankan aplikasi
root.mainloop()
```





TERIMA KASIH