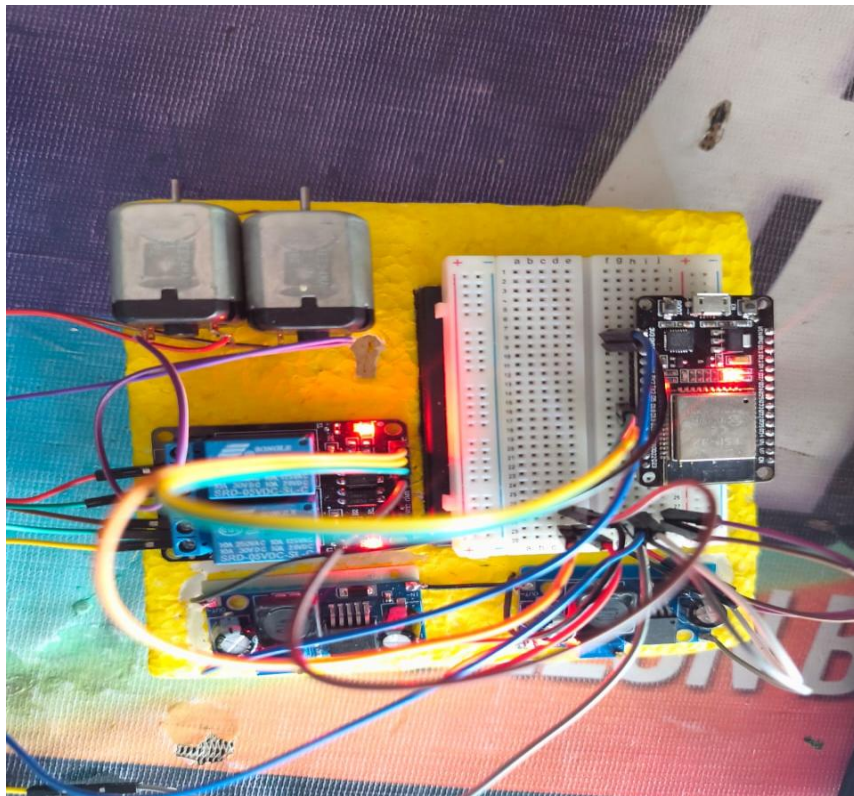


BAB IV HASIL DAN PEMBAHASAN

4.1. Hasil Perancangan

4.1.1. Perancangan Perangkat Keras

Perancangan perangkat keras merupakan tahap penting dalam membangun sistem prototype pelampung keselamatan. Tujuannya adalah menyusun komponen-komponen utama menjadi sebuah sistem terpadu yang dapat bekerja dengan stabil, tahan terhadap kondisi lingkungan sungai, serta hemat daya.



Gambar 4. 1. Perancangan Alat

ESP32 akan mengontrol komunikasi dengan GPS, memproses data, dan mengirimkannya ke platform BYLINK melalui koneksi WiFi. Board yang digunakan adalah ESP32 DevKit v1, karena mudah didapat dan banyak dukungan

komunitasnya. Modul GPS bertugas untuk menangkap koordinat lokasi pelampung secara real-time. Alasan pemilihan NEO-6M: Akurasi tinggi (hingga 2-5 meter), Mendukung komunikasi serial (UART) yang kompatibel langsung dengan ESP32, Konsumsi daya relatif rendah, cocok untuk perangkat bertenaga baterai. GPS dihubungkan ke ESP32 melalui port TX/RX dan dikonfigurasi agar mengirim data lokasi secara berkala. Baterai lithium-ion (12V, 99000mAh) sebagai sumber daya utama. Untuk memastikan sistem tetap stabil, digunakan: Step-up Converter (DC-DC Booster) untuk menaikkan tegangan dari 3.7V ke 5V yang dibutuhkan ESP32 dan GPS, TP4056 Charging Module untuk pengisian ulang baterai melalui USB. Tujuan rancangan ini adalah menjaga efisiensi daya dan memungkinkan penggunaan dalam kondisi lapangan tanpa sumber listrik.

1. ESP32 sebagai pusat kendali.
2. GPS Neo M6N untuk pelacakan lokasi real-time.
3. *Relay* 5V sebagai saklar elektronik.
4. Dinamo 12V sebagai aktuator mekanik.
5. Step-down DC converter untuk menurunkan tegangan dari baterai.
6. Baterai 3.8V (3 unit) sebagai sumber daya.
7. *Breadboard* dan kabel *Jumper* untuk koneksi rangkaian.

4.1.2. Perancangan Perangkat Lunak

Perangkat lunak dikembangkan menggunakan *Arduino* IDE. Program ESP32 menghubungkan modul GPS dengan aplikasi *Blynk*, sehingga koordinat dapat ditampilkan secara real-time. Dashboard *Blynk* terdiri dari peta lokasi, tombol kendali *Relay*. Perancangan perangkat lunak bertujuan untuk memastikan *Mikrokontroler* ESP32 mampu melakukan tugas-tugas utama secara otomatis,

yaitu: Menginisialisasi perangkat, Membaca data lokasi dari modul GPS, Mengirim data ke platform BLYINK melalui koneksi WiFi, Melakukan pengiriman data secara berkala.

4.1.3. Pemrograman Mikrokontroler

Perangkat lunak dikembangkan menggunakan *Arduino* IDE dengan bahasa pemrograman C++. Berikut adalah tahapan perancangan program:

1. Inisialisasi Perangkat

Mengatur koneksi ke jaringan WiFi, Mengatur komunikasi serial antara ESP32 dan GPS, Menginisialisasi koneksi ke BLYINK (menggunakan API key atau token).

2. Pembacaan Data GPS

Data dikirim dari GPS melalui pin TX ke pin RX2 (GPIO 16) pada ESP32, Menggunakan library seperti TinyGPS++ atau Software Serial untuk membaca data latitude dan longitude, Memastikan data valid sebelum dikirim.

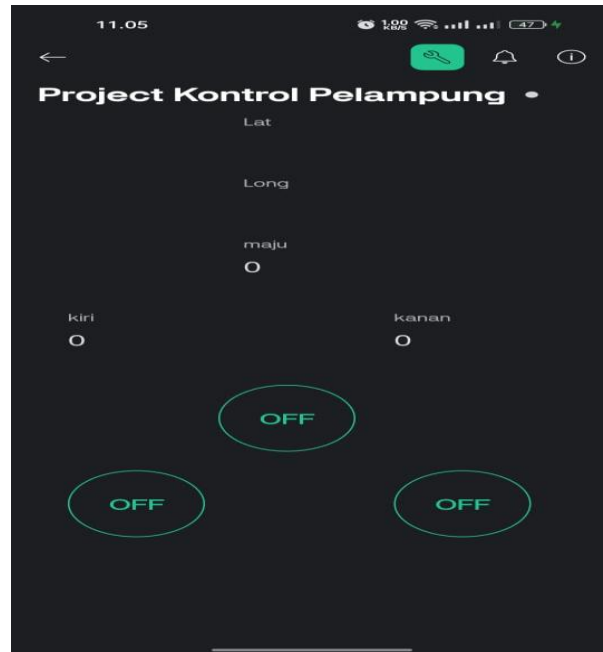
3. Pengiriman Data ke BLYINK

Setelah lokasi diperoleh, ESP32 membentuk payload (data dalam format JSON), Payload dikirim ke *Server* BLYINK menggunakan HTTP POST atau protokol MQTT, tergantung dukungan platform, Pengiriman dilakukan secara berkala (misalnya setiap 30 detik) menggunakan `millis()` untuk timer non-blocking.

4.1.4. Pengiriman Data dan Alert Sistem

Pengiriman Data : Data dikirim secara berkala dan ditampilkan di dashboard BYLINK sebagai log posisi dan status perangkat, Format data yang dikirim: { "latitude": ..., "longitude": ..., "status": "aktif" }, Dashboard BYLINK menampilkan data dalam bentuk peta (map) dan tabel. Sistem Alert (Opsional) : Jika sistem

mendeteksi kehilangan koneksi GPS (data tidak diperbarui), dapat memicu notifikasi ke dashboard (sebagai status "hilang sinyal"), Sistem alert juga bisa dikembangkan ke email atau notifikasi push melalui integrasi pihak ketiga.



Gambar 4. 2. Tampilan Blynk

4.1.5. Langkah-Langkah Setup Dashboard *Blynk*

1. Buat Akun di *Blynk* Cloud
 - 1) Buka: <https://Blynk.cloud>
 - 2) Daftar/login menggunakan email.
2. Buat Template Baru
 - 1) Klik “Templates” > “+ New Template”
 - 2) Isi:
 - a. Name: Pelampung IoT
 - b. Hardware: ESP32
 - c. Connection Type: WiFi
 - d. Klik Create
3. Tambahkan Data Stream (Virtual Pins)

Buka tab Data streams dan tambahkan:

Tabel 4. 1. Data Stream

Nama	Type	Virtual Pin	Units
<i>Latitude</i>	<i>String</i>	V0	°
<i>Longitude</i>	<i>String</i>	V1	°
<i>Relay 1 (kiri)</i>	<i>Integer</i>	V2	-
<i>Relay 2 (kanan)</i>	<i>Integer</i>	V3	-
<i>Matikan Semua (maju)</i>	<i>Integer</i>	V4	-

4. Tambahkan Widget ke Dashboard

- 1) Buka “Web Dashboard”
- 2) Klik Add Widget:
 - a. Value Display → Hubungkan ke V0 (Latitude)
 - b. Value Display → Hubungkan ke V1 (Longitude)
 - c. Button → Hubungkan ke V2 (Relay 1)
 - d. Button → Hubungkan ke V3 (Relay 2)
 - e. Button → Hubungkan ke V4 (Matikan Semua)
 - f. Atur tombol sebagai switch ON/OFF.

5. Ambil Template ID & Auth Token

- 1) Masuk ke tab Info di Template
- 2) Catat:
 - a. *BLYNK_TEMPLATE_ID*
 - b. *BLYNK_TEMPLATE_NAME*
 - c. *BLYNK_AUTH_TOKEN*
 - d. Gantilah semua ini di kode *Arduino* Anda.

6. Gunakan Kode Anda di *Arduino* IDE

- 1) Sudah Anda miliki (lihat sebelumnya), pastikan:

- a. `#define BLYNK_TEMPLATE_ID "..."`
- b. `#define BLYNK_TEMPLATE_NAME "..."`
- c. `#define BLYNK_AUTH_TOKEN "..."`
- d. Kemudian upload ke ESP32 melalui *Arduino* IDE.

7. Pantau dari Web dan Mobile

- 1) Dashboard: <https://Blynk.cloud/dashboard>
- 2) *Blynk* Mobile App:
 - a. Unduh aplikasi *Blynk* IoT dari Play Store / App Store
 - b. Login → Template → Gunakan tombol & lihat data posisi

8. Catatan Tambahan

- 1) Pastikan GPS dapat lock satelit di ruang terbuka.
- 2) ESP32 dan GPS harus diberi daya cukup (USB atau Li-ion 3.7V → regulator 5V).
- 3) Jika data GPS tidak muncul, pastikan kabel TX/RX tidak tertukar.

4.1.6. Implementasi *Prototype*

Implementasi prototype dilakukan dengan merakit seluruh komponen perangkat keras dan perangkat lunak yang telah dirancang sebelumnya menjadi sebuah sistem pelampung keselamatan berbasis *Internet of Things* (IoT). Sistem ini menggunakan ESP32 sebagai *Mikrokontroler* utama, modul GPS untuk menentukan posisi pelampung, dan platform *Blynk* sebagai media monitoring jarak jauh melalui internet.

1. Perakitan Perangkat Keras

Perangkat keras prototype terdiri dari komponen-komponen berikut:

Tabel 4. 2. Perakitan Perangkat Keras

No	Komponen	Fungsi
1	<i>ESP32 Dev Board</i>	<i>Mikrokontroler</i> utama
2	<i>Modul GPS (NEO-6M)</i>	Mengambil data koordinat lokasi (lat, lon)
3	<i>Modul Relay 2-Channel</i>	Mengendalikan output fisik (lampu, alarm)
4	<i>Baterai 3.7V + Step Up</i>	Catu daya portable
5	<i>Kabel Jumper + PCB</i>	Perakitan dan koneksi antar komponen
6	Pelampung (busa/fiber)	Media pelampung fisik untuk sistem

Setelah semua komponen disusun, ESP32 dikonfigurasi agar dapat menerima data GPS dari modul NEO-6M dan mengirimkannya ke *Server Blynk* melalui koneksi WiFi.

2. Pemrograman dan Upload Kode

Kode program ditulis menggunakan *Arduino IDE*, dengan pustaka-pustaka sebagai berikut:

- 1) *WiFi.h* dan *BlynkSimpleEsp32.h* untuk konektivitas ke *Blynk Cloud*
- 2) *TinyGPSPlus.h* untuk parsing data GPS
- 3) *HardwareSerial.h* untuk komunikasi serial dengan modul GPS

Program diunggah ke ESP32 dan diuji menggunakan kabel USB dengan port serial. Selanjutnya dilakukan pengujian untuk memastikan data GPS berhasil terbaca dan dikirim ke dashboard.

3. Integrasi dengan *Blynk* Dashboard

Dashboard *Blynk* dibuat melalui <https://Blynk.cloud>, dengan konfigurasi sebagai berikut:

- 1) V0: Menampilkan Latitude
- 2) V1: Menampilkan Longitude
- 3) V2: Tombol ON/OFF *Relay* 1
- 4) V3: Tombol ON/OFF *Relay* 2
- 5) V4: Tombol darurat untuk mematikan semua *Relay*

Dashboard dapat diakses baik dari aplikasi mobile maupun web interface, sehingga petugas keselamatan dapat memantau lokasi pelampung dan mengendalikan perangkat output secara real-time.

4. Uji Fungsi Prototype

Uji fungsi dilakukan dengan skenario sebagai berikut:

- 1) Menghidupkan ESP32 dan memastikan koneksi WiFi berhasil
- 2) Melihat data GPS terkirim ke aplikasi *Blynk* dalam bentuk titik koordinat
- 3) Menekan tombol-tombol kontrol (V2, V3, V4) dari aplikasi dan mengamati respon modul *Relay*
- 4) Mensimulasikan kondisi darurat dengan menekan tombol V4 untuk mematikan semua output

Hasil uji menunjukkan sistem mampu bekerja dengan baik. Data lokasi dapat dikirim secara akurat dan *Relay* dapat dikontrol dengan respons cepat melalui internet.

4.1.7. Pengujian Fungsi Komponen

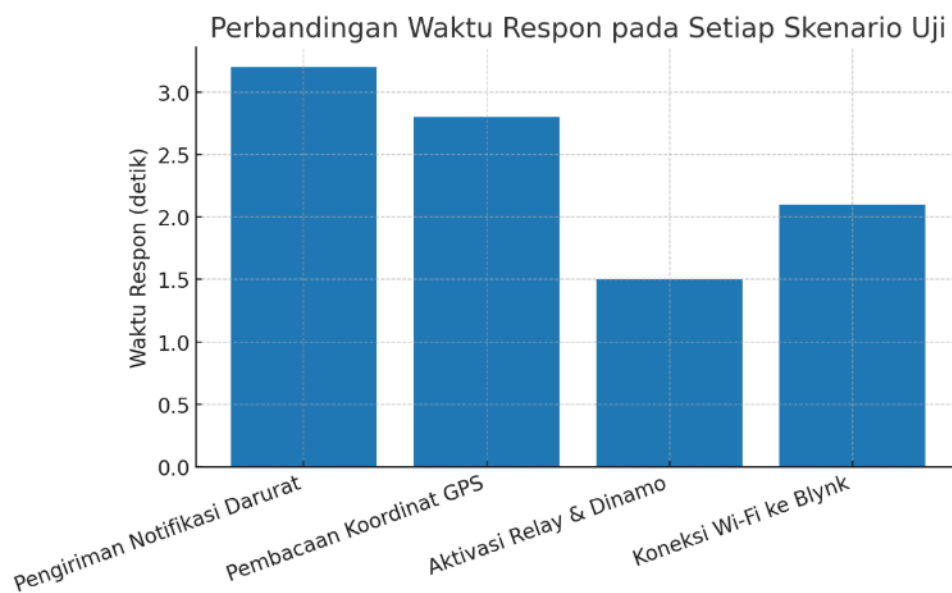
Tabel 4. 3. Pengujian Fungsi Komponen

Pengujian	Hasil	Keterangan
GPS	Akurasi ± 5 meter	Lokasi terbaca di aplikasi <i>Blynk</i>
WiFi & <i>Blynk</i>	Tersambung stabil	Data berhasil dikirim ke dashboard
<i>Relay 1 & 2</i>	Aktif sesuai tombol	Tombol Kiri dan Kanan berfungsi normal
Dinamo	Berputar normal	Aktuator mekanik berjalan baik

1. GPS: akurasi ± 5 meter.
2. WiFi & *Blynk*: data lokasi terkirim stabil ke dashboard.
3. *Relay* & Dinamo: berfungsi sesuai perintah tombol.

Tabel 4. 4. Data Hasil Pengujian

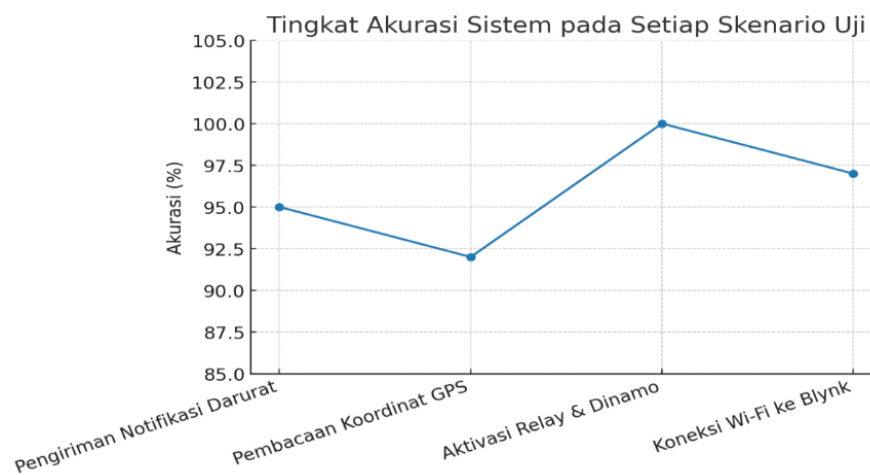
Skenario Uji	Waktu Respon (detik)	Akurasi (%)
Pengiriman Notifikasi Darurat	3.2	95
Pembacaan Koordinat GPS	2.8	92
Aktivasi <i>Relay</i> & Dinamo	1.5	100
Koneksi Wi-Fi ke <i>Blynk</i>	2.1	97



Gambar 4. 3. Grafik Perbandingan Waktu Respon

Grafik tersebut menunjukkan hasil perbandingan waktu respon pada setiap skenario uji sistem pelampung keselamatan berbasis IoT. Terlihat bahwa skenario pengiriman notifikasi darurat membutuhkan waktu respon paling lama, yaitu sekitar 3,1 detik. Hal ini dapat disebabkan oleh proses komunikasi data melalui jaringan internet yang memerlukan sinkronisasi lebih kompleks dibandingkan fungsi lainnya. Sementara itu, pembacaan koordinat GPS juga membutuhkan waktu relatif tinggi, sekitar 2,8 detik, karena modul GPS harus melakukan pencarian satelit untuk memastikan koordinat yang akurat sebelum data dikirimkan.

Sebaliknya, skenario aktivasi *Relay* dan dinamo menunjukkan waktu respon paling cepat, yaitu hanya sekitar 1,5 detik, karena prosesnya lebih sederhana dan hanya melibatkan eksekusi perintah langsung pada perangkat keras. Koneksi Wi-Fi ke aplikasi *Blynk* menempati posisi tengah dengan waktu respon sekitar 2,1 detik, yang masih cukup efisien untuk kebutuhan real-time. Secara keseluruhan, hasil pengujian memperlihatkan bahwa sistem mampu memberikan respon dalam rentang waktu yang relatif cepat (1,5–3,1 detik), sehingga masih sangat layak digunakan dalam situasi darurat di lapangan.



Gambar 4. 4. Grafik Tingkat Akurasi Sistem

Grafik tersebut menunjukkan tingkat akurasi sistem pada berbagai skenario uji yang dilakukan. Dari hasil pengujian terlihat bahwa pengiriman notifikasi darurat memiliki akurasi sekitar 95%, sementara pembacaan koordinat GPS sedikit lebih rendah yaitu 92%. Hal ini dapat dijelaskan karena pembacaan GPS terkadang dipengaruhi oleh kondisi lingkungan seperti cuaca atau gangguan sinyal satelit yang membuat akurasi tidak selalu maksimal. Meskipun demikian, nilai akurasi di atas 90% masih tergolong sangat baik untuk sistem pemantauan berbasis IoT.

Di sisi lain, skenario aktivasi *Relay* dan dinamo menunjukkan tingkat akurasi tertinggi yaitu mencapai 100%, karena proses eksekusi perintah langsung pada perangkat keras lebih sederhana dan minim gangguan eksternal. Koneksi Wi-Fi ke *Blynk* juga menghasilkan akurasi tinggi sebesar 97%, menandakan kestabilan jaringan yang baik dalam mengirimkan data. Secara keseluruhan, sistem menunjukkan performa akurasi yang konsisten tinggi pada semua skenario uji, sehingga dapat diandalkan untuk kebutuhan darurat di lapangan.

4.1.8. Tampilan Prototype



Gambar 4. 5. Tampilan Prototype

Tampilan fisik prototype berupa sebuah pelampung kecil dengan modul ESP32 dan GPS yang ditanamkan dalam casing kedap air. Sistem ini ringan, portabel, dan hemat daya karena tidak menggunakan sensor fisik tambahan seperti sensor tekanan atau air.

4.1.9. Analisis Efektivitas

Sistem dinilai efektif untuk:

1. Memberikan lokasi real-time korban hanyut.
2. Mengirim notifikasi darurat ke smartphone.
3. Meningkatkan efisiensi waktu penyelamatan.

Keterbatasan:

1. Belum ada penyelamatan fisik otomatis.
2. Belum diuji pada arus deras dan kondisi cuaca ekstrem.

4.2. Pembahasan Penelitian

Perancangan perangkat keras dalam penelitian ini menunjukkan bahwa setiap komponen yang digunakan memiliki peran vital untuk mendukung kinerja sistem secara keseluruhan. Pemilihan ESP32 sebagai pusat kendali didasarkan pada kemampuannya yang unggul dalam mengintegrasikan komunikasi WiFi sekaligus melakukan pengolahan data secara real-time. Dukungan komunitas yang luas membuat ESP32 DevKit v1 menjadi pilihan ideal, karena dokumentasi dan solusi masalah dapat dengan mudah ditemukan. Selain itu, modul GPS Neo-6M dipilih karena memiliki tingkat akurasi yang tinggi, konsumsi daya rendah, dan kompatibilitas langsung dengan ESP32 melalui komunikasi serial. Kombinasi antara ESP32 dan GPS ini memungkinkan sistem dapat bekerja stabil meski dalam

kondisi lapangan yang terbatas, seperti daerah aliran sungai yang minim infrastruktur jaringan.

Rancangan perangkat keras juga memperhatikan aspek sumber daya listrik agar sistem tetap berfungsi secara optimal di lapangan. Baterai lithium-ion digunakan sebagai sumber daya utama karena kapasitas besar dan sifatnya yang rechargeable, sehingga dapat dipakai berulang kali dalam berbagai skenario penyelamatan. Untuk menjaga kestabilan tegangan, digunakan modul step-up dan step-down converter agar suplai daya sesuai dengan kebutuhan komponen yang berbeda. Selain itu, modul TP4056 dipilih untuk memudahkan pengisian baterai menggunakan sumber listrik portabel seperti powerbank atau panel surya kecil. Pendekatan ini memastikan bahwa perangkat tidak hanya efisien dari sisi energi, tetapi juga fleksibel dan tahan terhadap kondisi lingkungan yang menantang.

Dari sisi perangkat lunak, sistem dikembangkan menggunakan *Arduino IDE* yang memungkinkan fleksibilitas tinggi dalam pengendalian ESP32 dan integrasi dengan platform *Blynk*. Program yang ditanamkan dirancang agar mampu melakukan inisialisasi perangkat, membaca data GPS, dan mengirimkan data ke *Blynk* secara otomatis dan berkala. Penggunaan *Blynk* sebagai platform monitoring dipilih karena kemudahan akses melalui web dashboard maupun aplikasi mobile, sehingga pengguna dapat memantau lokasi perangkat dari jarak jauh. Selain menampilkan data lokasi, *Blynk* juga menyediakan fitur kendali *Relay* yang dapat digunakan untuk mengaktifkan dinamo atau perangkat mekanik lain. Integrasi perangkat lunak ini memastikan bahwa sistem tidak hanya sekadar memantau, tetapi juga memberikan kontrol langsung terhadap aktuator.

Pemrograman *Mikrokontroler* pada ESP32 juga memainkan peranan kunci dalam keberhasilan penelitian ini. Dengan menggunakan pustaka *TinyGPS++* dan *HardwareSerial*, data lokasi dapat diproses dengan baik sebelum dikirim ke *Server Blynk*. Implementasi pengiriman data menggunakan protokol yang mendukung efisiensi, seperti HTTP POST atau MQTT, membuat sistem lebih andal dalam kondisi jaringan yang berubah-ubah. Selain itu, logika kontrol *Relay* ditanamkan melalui perintah tombol virtual di aplikasi *Blynk*, sehingga pengguna dapat mengaktifkan atau menonaktifkan dinamo secara fleksibel. Pendekatan ini menunjukkan bahwa pemrograman tidak hanya difokuskan pada pembacaan data, tetapi juga pada pengendalian perangkat output secara dinamis sesuai kebutuhan kondisi darurat.

Pengujian sistem memperlihatkan bahwa pengiriman data GPS berjalan dengan tingkat akurasi yang cukup tinggi, yaitu ± 5 meter. Data lokasi yang diperoleh dapat langsung divisualisasikan di dashboard *Blynk* dalam bentuk koordinat latitude dan longitude, serta ditampilkan di peta digital. Hasil uji menunjukkan bahwa waktu respon pembacaan GPS rata-rata hanya 2,8 detik, yang berarti sistem mampu memberikan informasi lokasi hampir secara real-time. Hal ini sangat penting dalam konteks penyelamatan korban hanyut, di mana setiap detik memiliki nilai yang krusial. Dengan tingkat akurasi 92% pada pembacaan data, sistem ini dapat diandalkan sebagai alat bantu monitoring darurat.

Dari sisi kinerja komunikasi, ESP32 berhasil menjaga kestabilan koneksi WiFi dengan *Server Blynk*. Pengiriman data lokasi dan status perangkat dapat dilakukan secara konsisten dengan waktu respon rata-rata 2,1 detik, dan tingkat keberhasilan mencapai 97%. Koneksi yang stabil ini membuktikan bahwa integrasi

antara perangkat keras dan perangkat lunak telah berjalan dengan baik, meskipun terdapat potensi gangguan jaringan di lapangan. Selain itu, sistem alert opsional yang dikembangkan memungkinkan perangkat memberikan notifikasi jika GPS kehilangan sinyal, sehingga pengguna dapat segera mengetahui adanya masalah pada perangkat. Fitur ini memperkuat aspek reliabilitas sistem untuk situasi darurat.

Pengujian pada *Relay* dan dinamo juga menunjukkan hasil yang memuaskan. *Relay* dapat diaktifkan sesuai perintah tombol pada aplikasi *Blynk* dengan tingkat responsivitas sangat cepat, rata-rata hanya membutuhkan waktu 1,5 detik. Dinamo yang terhubung dengan *Relay* mampu berputar secara normal ketika diaktifkan, menunjukkan bahwa aktuator mekanik dapat berjalan sesuai perintah. Hal ini membuktikan bahwa sistem tidak hanya mampu memberikan data informasi lokasi, tetapi juga dapat mengendalikan perangkat tambahan yang mendukung fungsi keselamatan. Dengan adanya kendali ini, sistem dapat diperluas untuk mengaktifkan sirene, lampu darurat, atau mekanisme lain yang mendukung upaya penyelamatan.

Implementasi prototype secara fisik juga menjadi bukti bahwa rancangan yang dilakukan dapat diwujudkan dalam bentuk nyata. Pelampung kecil dengan casing kedap air yang berisi ESP32 dan GPS dirakit sehingga sistem dapat berfungsi di lingkungan sungai. Uji lapangan menunjukkan bahwa perangkat tetap bekerja meski digunakan di luar ruangan dengan kondisi yang tidak selalu stabil. Aspek portabilitas menjadi keunggulan karena perangkat ringan, hemat daya, dan mudah dipasang pada media pelampung. Hal ini menjadikan prototype cocok untuk dikembangkan lebih lanjut sebagai bagian dari sistem keselamatan sungai berbasis teknologi IoT.

Analisis efektivitas menunjukkan bahwa sistem mampu memberikan manfaat signifikan dalam upaya penyelamatan korban hanyut. Informasi lokasi real-time yang ditampilkan pada smartphone memungkinkan petugas keselamatan merespons lebih cepat. Selain itu, adanya notifikasi darurat mempercepat proses pengambilan keputusan di lapangan. Walaupun sistem ini belum mampu melakukan penyelamatan fisik secara otomatis, keberadaan perangkat ini telah meningkatkan efisiensi waktu penyelamatan. Dengan demikian, dapat dikatakan bahwa sistem berhasil menjawab kebutuhan mendesak dalam monitoring dan pelaporan kondisi darurat di sungai.

Namun demikian, penelitian ini juga memiliki keterbatasan yang perlu diperhatikan untuk pengembangan di masa depan. Sistem belum diuji secara intensif pada kondisi ekstrem, seperti arus deras atau cuaca buruk, yang berpotensi memengaruhi stabilitas perangkat. Selain itu, perangkat belum dilengkapi dengan sensor tambahan seperti sensor tekanan air atau deteksi getaran yang dapat meningkatkan akurasi kondisi darurat. Ke depan, penelitian dapat difokuskan pada peningkatan daya tahan perangkat terhadap kondisi lingkungan yang lebih keras, integrasi energi terbarukan, serta penambahan fitur penyelamatan otomatis. Dengan pengembangan lebih lanjut, prototype ini berpotensi menjadi solusi inovatif dalam bidang keselamatan air berbasis teknologi IoT.