

**IMPLEMENTASI SENSOR PIR (PASSIVE INFRARED) DALAM
SISTEM KEAMANAN STATIS DENGAN SISTEM PANIC
BUTTON DAN LORA (LONG RANGE) 915 MHZ**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI ELEKTRONIKA

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



**FIRNA FRILANISA
NIM. 175060307111004**

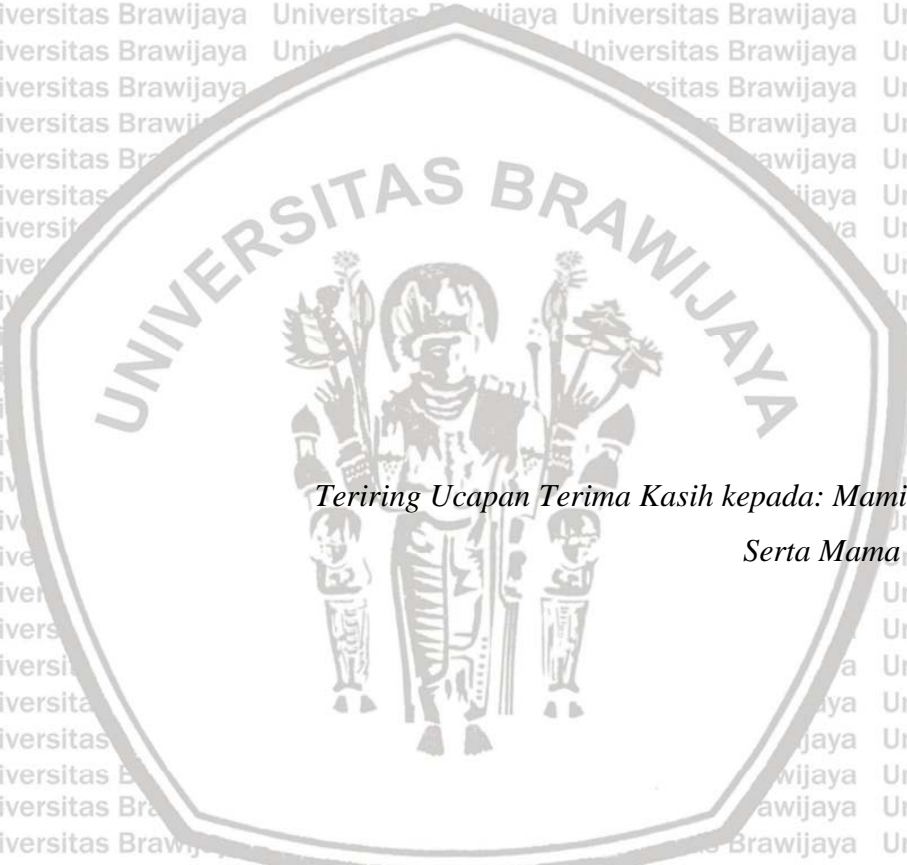
**UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG**

2021









*Teriring Ucapan Terima Kasih kepada: Mami, Nenek tersayang
Serta Mama dan Ayah tercinta*

PERNYATAAN ORISINALITAS SKRIPSI, TESIS, ATAU DISERTASI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Jakarta, 13 Juni 2021
Mahasiswa,

FIRNA FRILANISA
NIM. 175060307111004



RINGKASAN

Firna Frilanisa, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Juni 2021, Implementasi Sensor PIR (*Passive Infrared*) Dalam Sistem Keamanan Statis Dengan Sistem *Panic Button* Dan LoRa (*Long Range*) 915 MHz, Dosen Pembimbing: Rudy Yuwono, S.T., M.Sc. Dan Fakhriy Hario P., ST., MT., Dr., IPM.

Tindakan kriminal sudah semakin marak terjadi terutama pencurian dengan pemeratan dimana semakin banyak terjadi di bangunan ataupun rumah. Pencurian ini seringkali tidak ditangani secara langsung oleh aparat keamanan dikarenakan terlambatnya laporan oleh korban maupun saksi mata sehingga pelaku berhasil lolos dari pengejaran. Dalam kondisi darurat, korban juga akan terjebak dalam situasi kepanikan dan sulit untuk melarikan diri sehingga kondisinya tidak memungkinkan untuk melapor, maka dibutuhkan sistem keamanan yang efektif dan tanggap dalam menghadapi kondisi darurat.

Solusi yang ditawarkan untuk mengatasi permasalahan diatas adalah dengan merancang suatu sistem keamanan yang tanggap dalam memberikan pesan atau sinyal kondisi bahaya di lingkungan sekitar dan aparat keamanan dengan memanfaatkan transceiver LoRa dengan pengaktifan sensor dan *panic button*. Sistem ini menggunakan mikrokontroler Arduino Uno dan ESP32 sebagai pengendali komponen elektronik yang diintegrasikan dengan transceiver LoRa, *Panic Button*, dan sensor PIR sebagai komponen pendukung. Penelitian ini membahas tentang alat pendeteksi gerakan. Alat ini dirancang untuk membantu pemilik bangunan untuk mendeteksi gerakan yang terjadi di bangunan tersebut ketika pemilik bangunan tidak berada di dalam. Pendeteksi gerakan yang dirancang ini bertujuan untuk membantu sistem keamanan bangunan atau rumah. Sensor PIR digunakan sebagai pendeteksi gerakan yang mengirimkan notifikasi kepada user melalui aplikasi Blynk yang sudah di instal pada smartphone.

Hasil penelitian menunjukkan sensor PIR dapat mendeteksi gerakan manusia sejauh 6 meter dan mengirim alamat terjadinya bahaya dengan menekan *push button*. Sistem ini dapat mengirim pembacaan sensor dengan mengirim notifikasi berupa teks menggunakan aplikasi Blynk secara otomatis apabila sensor PIR mendeteksi gerakan atau saat *push button* ditekan.

Kata kunci: Sensor PIR, Deteksi gerakan, LoRa, Blynk, *Panic Button*, Arduino Uno, ESP32

SUMMARY

Firna Frilanisa, Department of Electrical Engineering, Faculty of Engineering Universitas Brawijaya, June 2021, Implementation of PIR (Passive Infrared) Sensors in Static Security Systems With Panic Button Systems and LoRa (Long Range) 915 MHz, Advisor Lecturer: Rudy Yuwono, ST, M.Sc . And Fakhriy Hario P., ST., MT., Dr., IPM.

Criminal acts are increasingly prevalent, especially theft with equity, where more and more occur in buildings or homes. This theft is often not handled directly by the security forces due to late reports by victims and eyewitnesses so that the perpetrators managed to escape from being chased. In an emergency, the victim will also be trapped in a panic situation and find it difficult to escape so that the condition is not possible to report, so an effective and responsive security system is needed in dealing with emergency conditions.

The solution offered to overcome the above problems is to design a security system that is responsive in providing messages or signals of dangerous conditions in the surrounding environment and security forces by utilizing LoRa transceivers with sensor activation and panic buttons. This system uses Arduino Uno and ESP32 microcontrollers as electronic component controllers that are integrated with LoRa transceivers, Panic Buttons, and PIR sensors as supporting components. This study discusses the motion detection device. This tool is designed to help building owners to detect movements that occur in the building when the building owner is not inside. This motion detector is designed to help building or home security systems. The PIR sensor is used as a motion detector that sends notifications to the user through the Blynk application that has been installed on the smartphone.

The results of the study show that the PIR sensor can detect human movement as far as 6 meters and send the address of the occurrence of danger by pressing the push button. This system can send sensor readings by sending text notifications using the Blynk application automatically when the PIR sensor detects motion or when the push button is pressed.

Keywords: PIR Sensor, Motion Detection, LoRa, Blynk, Panic Button, Arduino Uno, ESP32



12. Keluarga Besar HME FT UB, yang bisa membuat penulis mempelajari hal baru dan mengerti bagaimana kehidupan berorganisasi.

13. Seluruh teman-teman serta semua pihak yang tidak mungkin bagi penulis untuk mencantumkan satu-persatu, terimakasih banyak atas bantuan dan dukungannya.

Pada akhirnya, Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat. Penulis sebagai penyusun menyadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini karena keterbatasan pengetahuan dan pengalaman Penulis. Untuk itu Penulis sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan skripsi ini.

Jakarta, 26 Januari 2021

Penulis



DAFTAR ISI

PENGANTAR..... i

DAFTAR ISI.....iii

DAFTAR TABEL..... v

DAFTAR GAMBAR vi

DAFTAR LAMPIRAN vii

BAB I PENDAHULUAN 1

1.1 Latar belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 2

1.4 Tujuan 3

1.6 Manfaat 3

BAB II TINJAUAN PUSTAKA 4

2.1 Sinar Inframerah 4

2.2 Sensor PIR 4

2.3 *Push Button* 6

2.4 Mikrokontroler Arduino Uno 7

2.5 LCD 8

2.6 Buzzer 10

2.7 Mikrokontroler ESP32 11

2.8 SSD1306 OLED Display 12

2.9 LORA RFM95/96/97/98 12

2.10 Aplikasi Blynk 13

BAB III METODE PENELITIAN..... 14

3.1 Deskripsi Umum Sistem 14

3.2 Spesifikasi Alat 14

3.3 Studi Literatur 15

3.4 Perancangan Hardware 15

3.4.1 Perancangan Transmitter 15

3.4.2 Perancangan Receiver 17

3.5 Konfigurasi Rangkaian 17

3.5.1 Konfigurasi Rangkaian PIR 18

3.5.2 Konfigurasi Rangkaian *Panic Button* 19

3.5.3 Konfigurasi LoRa Transmitter 19

3.5.4	Konfigurasi LoRa Receiver	20
3.6	Perancangan Software	21
3.7	Pengujian Alat dan Sistem	22
3.7.1	Pengujian Sub Sistem	23
3.7.2	Pengujian Seluruh Sistem	23
BAB IV HASIL DAN PEMBAHASAN		24
4.1	Hasil dan Pembahasan Pengujian Sub Sistem	24
4.1.1	Pengujian Sensor	24
4.1.2	Pengujian <i>Panic Button</i>	29
4.1.3	Pengujian Aplikasi Blynk	30
4.2	Hasil Dan Pembahasan Pengujian Seluruh Sistem	34
BAB V KESIMPULAN DAN SARAN		35
5.1	Kesimpulan	35
5.2	Saran	35
DAFTAR PUSTAKA		36
LAMPIRAN		38



DAFTAR TABEL

Tabel. 2. 1 Spesifikasi Arduino Uno 8

Tabel. 2. 2 Spesifikasi ESP32..... 11

Tabel. 3. 1 Konfigurasi Pin LoRa..... 20

Tabel. 3. 2 Konfigurasi Pin LoRa Receiver..... 21

Tabel. 4. 1 Hasil Pengujian Jarak Sensor PIR 25

Tabel. 4. 2 Hasil Pengujian Sudut Sensor PIR 26

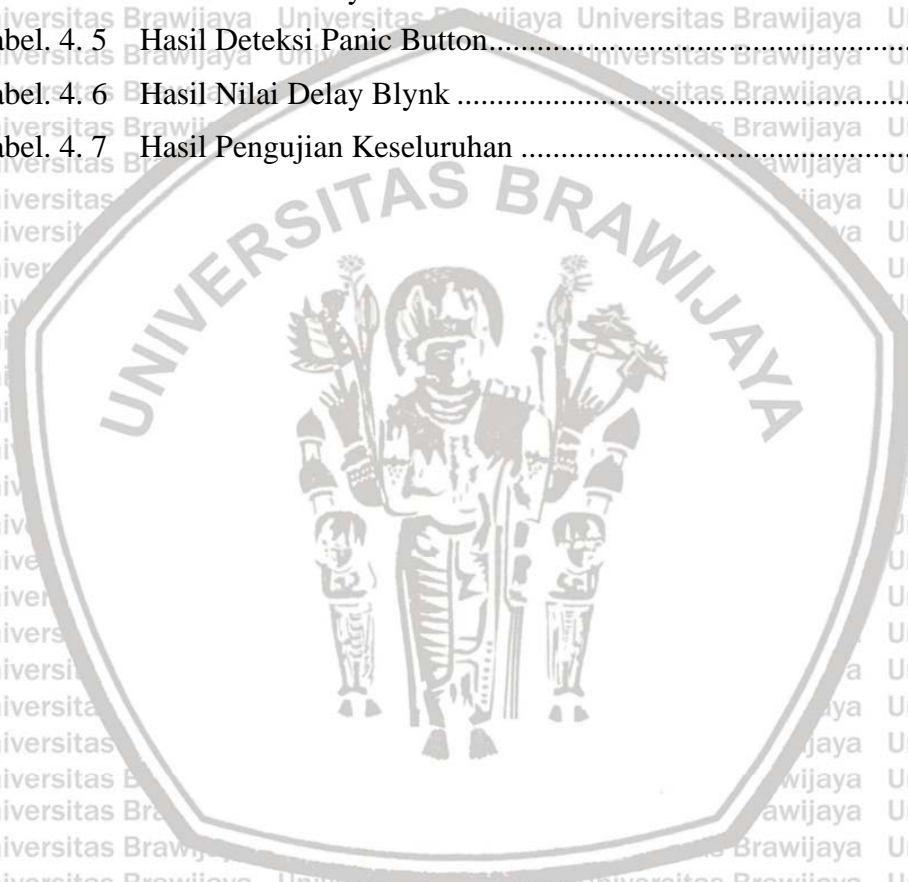
Tabel. 4. 3 Hasil Deteksi Kepekaan Sensor PIR 27

Tabel. 4. 4 Hasil Nilai Delay Sensor PIR 28

Tabel. 4. 5 Hasil Deteksi Panic Button..... 29

Tabel. 4. 6 Hasil Nilai Delay Blynk 32

Tabel. 4. 7 Hasil Pengujian Keseluruhan 34



DAFTAR GAMBAR

Gambar. 1. 1	Jumlah Kasus Pencurian di Indonesia pada Tahun 2018-2019.....	1
Gambar. 2. 1	Sensor PIR.....	5
Gambar. 2. 2	Jangkauan dan Gelombang Sensor PIR.....	5
Gambar. 2. 3	Push Button	6
Gambar. 2. 4	Arduino Uno.....	7
Gambar. 2. 5	Arduino Uno Pin Diagram	8
Gambar. 2. 6	LCD.....	9
Gambar. 2. 7	Buzzer.....	10
Gambar. 2. 8	ESP32	11
Gambar. 2. 9	SSD1306 OLED Display.....	12
Gambar. 2. 10	RFM95W LoRa Module	12
Gambar. 3. 1	Diagram Blok Gambaran Umum Sistem.....	14
Gambar. 3. 2	Diagram Blok Perangkat Transmitter.....	16
Gambar. 3. 3	Skema Transmitter	16
Gambar. 3. 4	Diagram Blok Perangkat Transmitter.....	17
Gambar. 3. 5	Skema Receiver.....	17
Gambar. 3. 6	Konfigurasi Rangkaian Transmitter	18
Gambar. 3. 7	Konfigurasi Rangkaian PIR.....	19
Gambar. 3. 8	Konfigurasi Rangkaian Panic Button	19
Gambar. 3. 9	Konfigurasi Rangkaian LoRa.....	20
Gambar. 3. 10	Diagram Alir Sistem.....	22
Gambar. 4. 1	Grafik Delay Sensor PIR.....	28
Gambar. 4. 2	Serial Monitor.....	30
Gambar. 4. 3	Hasil Pengiriman	31
Gambar. 4. 4	Grafik Delay Blynk	33

DAFTAR LAMPIRAN

Lampiran 1 Implementasi Transmitter 39

Lampiran 2 Implementasi Receiver 39

Lampiran 3 Dokumentasi Alat Pada Transmitter 40

Lampiran 4 Dokumentasi Alat Pada Receiver 40

Lampiran 5 Listing Program Pada Transmitter 41

Lampiran 6 Listing Program Pada Receiver 47

Lampiran 7 Datasheet 51

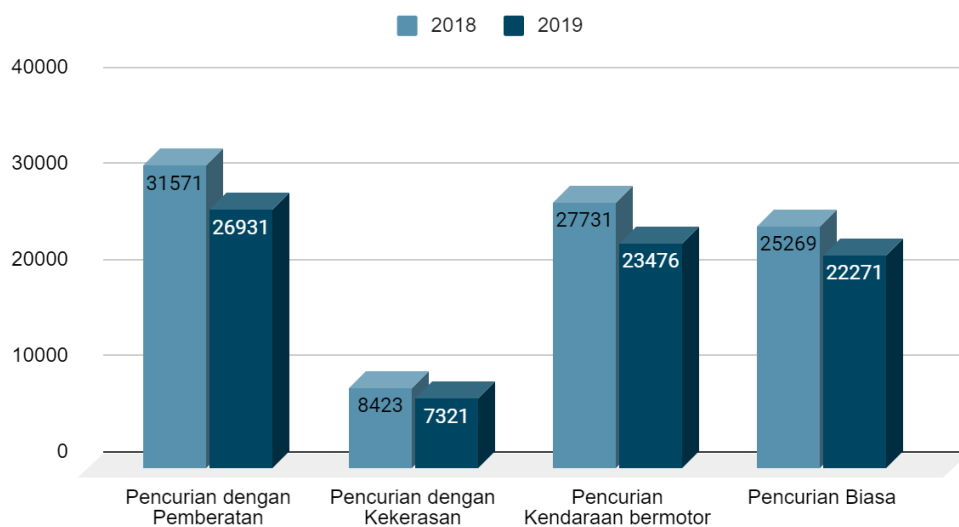


BAB I PENDAHULUAN

1.1 Latar belakang

Terdapat berbagai macam jenis pencurian diantaranya yaitu pencurian dengan pemberatan, pencurian dengan kekerasan, pencurian kendaraan bermotor dan pencurian biasa. Dari 4 jenis pencurian tersebut, pencurian dengan pemberatan adalah yang paling sering terjadi. Pencurian pada gedung atau bangunan masuk dalam kategori pencurian dengan pemberatan, pencurian dengan kekerasan ataupun pencurian biasa. Seperti yang tertera dalam Pasal 363 KUHP pencurian dengan pemberatan antara lain pencurian hewan, pencurian saat bencana, pencurian yang dilakukan dengan cara membongkar, memecah, atau memanjat pada rumah atau pekarangan tertutup (Biro Hukum dan Humas, Badan Urusan Administrasi, Mahkamah Agung RI, 2021).

Kasus Pencurian



Gambar. 1. 1 Jumlah Kasus Pencurian di Indonesia pada Tahun 2018-2019

Sumber : (Badan Pusat Statistik, 2020)

Gambar 1.1 menunjukkan grafik jumlah kasus pencurian di Indonesia pada tahun 2018-2019. Menurut data dari Badan Pusat Statistik, selama tahun 2018 telah terjadi pencurian dengan pemberatan sebanyak 31.571 kasus yang terjadi dan pada tahun 2019 terjadi penurunan jumlah kasus pencurian dengan pemberatan menjadi 26.931 kasus. Melihat besarnya angka pencurian dengan pemberatan maka diperlukan perancangan sistem

keamanan untuk memonitor sekaligus dapat mengirim alarm tanda akan bahaya ke aparat keamanan secara *real time*. (Badan Pusat Statistik, 2020).

Tindakan kriminal seringkali tidak ditangani secara langsung oleh aparat keamanan dikarenakan terlambatnya laporan oleh korban maupun saksi mata sehingga pelaku berhasil lolos dari pengejaran. Ketidakakuratan laporan mengenai alamat korban juga menjadi salah satu penyebab lambatnya penanganan oleh pihak kepolisian atau aparat keamanan. Dalam kondisi darurat, korban juga akan terjebak dalam situasi kepanikan dan sulit untuk melarikan diri sehingga kondisinya tidak memungkinkan untuk melapor, maka dibutuhkan sistem keamanan yang efektif dan tanggap dalam menghadapi kondisi darurat.

Solusi dari permasalahan diatas adalah dengan merancang suatu sistem keamanan yang tanggap dalam memberikan pesan atau sinyal kondisi bahaya di lingkungan sekitar dan aparat keamanan dengan memanfaatkan *transceiver* LoRa dengan pengaktifan sensor dan *panic button*.

Skripsi ini akan merancang sebuah sensor dan *panic button* yang dapat mengirimkan sinyal bahaya saat mendeteksi gerakan atau dalam keadaan darurat melalui transmisi LoRa dan aplikasi blynk ke *smartphone*. Hasil keluaran pada aplikasi blynk berupa indikasi alamat tempat terjadinya bahaya. Sistem ini terdiri dari sensor PIR, *panic button*, mikrokontroler Arduino UNO, mikrokontroler ESP32, LED, *buzzer*, serta *transceiver* LoRa 915 MHz.

1.2 Rumusan Masalah

Berdasarkan masalah yang telah dijelaskan pada latar belakang, dapat dibuat rumusan masalah yaitu :

1. Bagaimana merancang suatu sistem yang dapat mendeteksi gerakan dan keadaan darurat di dalam bangunan?
2. Bagaimana merancang suatu sistem yang cepat dan tanggap dalam melaporkan keadaan darurat ke aparat keamanan setempat?

1.3 Batasan Masalah

Penelitian ini berkaitan dengan perancangan sistem pada bagian keamanan, komunikasi nirkabel, dan antarmuka pengguna. Oleh karena itu, penulis membatasi penelitian ini hanya pada :

1. Perancangan sensor PIR untuk mendeteksi objek yang bergerak.
2. Perancangan *push button* sebagai *panic button* dalam keadaan darurat.
3. Perancangan antarmuka menggunakan Blynk sebagai *monitoring device*.

4. Sistem ini menggunakan mikrokontroler Arduino Uno dan ESP32.

1.4 Tujuan

Berdasarkan rumusan masalah yang telah diuraikan diatas, maka dapat diketahui tujuan penelitian yaitu sebagai berikut :

1. Merancang sensor dan antarmuka dalam sistem keamanan statis dengan sistem *panic button* menggunakan LoRa 915 MHz
2. Mempermudah pelaporan saat terjadi keadaan darurat kepada aparat keamanan setempat.

1.6 Manfaat

Adapun manfaat yang dapat diambil dari penelitian ini adalah untuk menghasilkan rancang bangun sensor dan antarmuka dalam sistem keamanan statis dengan sistem *panic button* menggunakan LoRa 915 MHz dan mempermudah dalam pelaporan serta monitoring pengguna akan adanya bahaya ke aparat keamanan setempat.



BAB II

TINJAUAN PUSTAKA

2.1 Sinar Inframerah

Radiasi inframerah (IR) adalah jenis radiasi elektromagnetik. Cahaya inframerah memiliki panjang gelombang yang lebih panjang daripada cahaya tampak. Inframerah memiliki panjang gelombang 750 nm sampai 100 m. Radiasi inframerah tidak terlihat oleh manusia tetapi kita dapat merasakannya sebagai panas (Soyer, 2009). Inframerah selanjutnya dapat dibagi sesuai jangkauannya sebagai berikut :

1. *Near Infrared* (NIR): 750 nm - 1.5 μm .
2. *Short Wavelength Infrared* (SWIR): 1.5 μm - 3 μm .
3. *Mid Wavelength Infrared* (MWIR): 3 μm - 8 μm .
4. *Long Wavelength Infrared* (LWIR): 8 μm - 15 μm .
5. *Far Infrared* (FIR): Lebih dari 15 μm

MWIR dan LWIR dikenal sebagai inframerah termal. Semua benda memancarkan apa yang dikenal sebagai radiasi benda hitam (radiasi termal). Ini dipancarkan dari permukaan suatu benda yang disebabkan oleh suhunya. Tubuh manusia pada suhu tubuh normal memancarkan IR kira-kira pada panjang gelombang sekitar 9,4 μm (Soyer, 2009).

2.2 Sensor PIR

Sensor PIR (*Passive Infrared*) adalah suatu alat yang berfungsi untuk menangkap suatu besaran fisis (temperatur suhu tubuh manusia) dan merubahnya ke bentuk sinyal listrik. Sesuai dengan namanya, *Passive Infrared* yang berarti memiliki sifat pasif. Pasif disini memiliki arti bahwa sensor tidak memancarkan sinar infra merah tetapi hanya menerima radiasi infra merah yang dipancarkan oleh suatu objek yang bergerak (dalam hal ini tubuh manusia) (Zain, 2013).

Sensor PIR Seperti yang terlihat Gambar 2.1 terdiri dari beberapa bagian yaitu :

1. *Fresnel Lens*
2. *IR Filter*
3. *Pyroelectric sensor*
4. *Amplifier*
5. *Comparator*



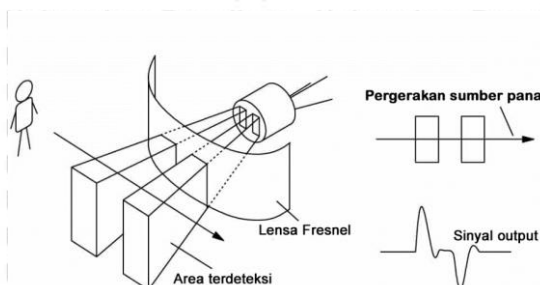
Gambar. 2. 1 Sensor PIR

Sumber : <https://www.tokopedia.com/delta988electro/pir-sensor-hc-sr501>

Sensor PIR dibagi menjadi dua elemen *sensing* yang terhubung dengan masukan. Kedua bagian ini digunakan untuk mendeteksi gerakan (perubahan) bukan tingkat IR rata-rata. Jika salah satu bagian melihat lebih banyak atau lebih sedikit radiasi IR daripada yang lain, *output* akan berubah *High* atau *Low* (Adafruit Industries , 2020).

Saat sensor dalam keadaan *idle*, kedua slot mendeteksi jumlah IR yang sama yang terpancar dari ruangan, dinding, atau luar ruangan. Ketika sumber panas seperti manusia atau hewan lewat, pertama kali memotong setengah dari sensor PIR, maka akan menyebabkan perubahan diferensial positif antara dua bagian. Ketika sumber panas meninggalkan area penginderaan, kebalikannya terjadi, di mana sensor menghasilkan perubahan diferensial negatif. Perubahan *pulse* inilah yang terdeteksi (Adafruit Industries , 2020). Untuk membatasi rentang sensitivitas sensor, terdapat filter IR. Filter ini umumnya dirancang untuk bandwidth $5,5 \mu\text{m}$ hingga $15 \mu\text{m}$ (Soyer, 2009).

Menurut (Wibowo, 2018), Ketika ada sumber panas yang lewat di depan sensor tersebut, maka sensor akan mengaktifkan sel pertama dan sel kedua sehingga akan menghasilkan bentuk gelombang seperti ditunjukkan dalam Gambar 2.2.



Gambar. 2. 2 Jangkauan dan Gelombang Sensor PIR

Sumber : <https://www.hestech.id/2020/03/cara-menggunakan-sensor-gerak-pir-hc.html>

Berdasarkan *Datasheet* HC-SR501, Sensor PIR memiliki fitur dan karakteristik sebagai berikut:

1. Tegangan input bervariasi dari 4.V hingga 12V (disarankan + 5V)
2. Tegangan output Tinggi / Rendah (3.3V/TTL)
3. Dapat membedakan antara gerakan objek dan gerakan manusia
4. Jangkauan sensor sekitar 120 ° dan 7 meter
5. Konsumsi daya rendah 65mA
6. Suhu pengoperasian dari -20 ° hingga + 80 ° Celsius

2.3 Push Button

Push button merupakan saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan *unlock* (tidak mengunci). Sistem kerja *unlock* disini berarti saklar akan bekerja sebagai *device* penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal dapat dilihat bentuk komponen *push button* dalam Gambar 2.3. (SAHIDIN, 2016)



Gambar. 2. 3 Push Button

Sumber : <https://www.tokopedia.com/ctm/push-button-kecil>

Berdasarkan fungsi kerjanya yang menghubungkan dan memutuskan, *push button* mempunyai 2 tipe kontak yaitu sebagai berikut :

1. NO (*Normally Open*), merupakan kontak terminal dimana kondisi normalnya terbuka (aliran arus listrik tidak mengalir). Dan ketika tombol ditekan, kontak yang NO ini akan menjadi menutup (*Close*) dan mengalirkan atau menghubungkan arus listrik. Kontak NO digunakan sebagai penghubung atau menyalakan sistem circuit (*Push Button ON*).

2. NC (*Normally Close*), merupakan kontak terminal dimana kondisi normalnya tertutup (mengalirkan arus listrik). Dan ketika tombol *push button* ditekan, kontak NC ini akan menjadi membuka (*Open*), sehingga memutuskan aliran arus listrik. Kontak NC digunakan sebagai pemutus atau mematikan sistem circuit (*Push Button Off*). (SAHIDIN, 2016)

2.4 Mikrokontroler Arduino Uno

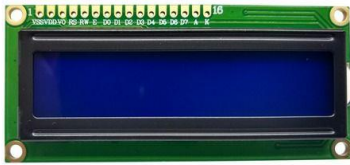
Sistem mikrokontroler menggunakan Arduino yang merupakan sebuah *board* mikrokontroler yang berbasis ATmega328 yang dapat dilihat dalam Gambar 2.4. Arduino memiliki 14 pin *input/output* yang mana 6 pin dapat digunakan sebagai *output* PWM, 6 *analog input*, crystal osilator 16 MHz, koneksi USB, *jack power*, kepala ICSP, dan tombol *reset*. Arduino mampu mensupport mikrokontroler hanya dengan menghubungkan Arduino ke komputer menggunakan kabel USB atau memberi catu daya dari AC to DC adapter atau baterai. (farnell, 2021)



Gambar. 2. 4 Arduino Uno

Sumber : https://www.arduino.cc/en/uploads/Main/ArduinoUno_R3_Front.jpg

Arduino memiliki 20 pin I/O yang terdiri dari 6 pin *input analog* dan 14 pin *digital input/output* yang ditunjukkan dalam Gambar 2.5. Masing-masing dari 14 pin digital pada Uno dapat digunakan sebagai input atau output menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up internal* (terputus secara default) sebesar 20-50 kOhm. Arduino memiliki 6 pin *input analog* yang diberi label A0 hingga A5. Masing-masing memberikan resolusi 10 bit (yaitu 1024 nilai berbeda). Secara default pin ini mengukur dari ground ke 5 volt, meskipun mungkin untuk mengubah ujung atas jangkauan menggunakan pin AREF dan fungsi *analogReference()*. (farnell, 2021)



Gambar. 2. 6 LCD

Sumber : <https://www.tokopedia.com/celectro/lcd-16x2-blue>

Menurut (WIJAYA, 2017), modul LCD (*Liquid Cristal Display*) memiliki mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD (*Liquid Crystal Display*). Mikrokontroler pada suatu LCD (*Liquid Cristal Display*) dilengkapi dengan memori dan register. Memori yang digunakan mikrokontroler internal LCD adalah :

- DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang akan ditampilkan berada.
- CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan.
- CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrik pembuat LCD (*Liquid Cristal Display*) tersebut sehingga pengguna tinggal mengambilnya sesuai alamat memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM.

Register control yang terdapat dalam suatu LCD diantaranya adalah:

- Register perintah yaitu register yang berisi perintah-perintah dari mikrokontroler ke panel LCD (*Liquid Cristal Display*) pada saat proses penulisan data atau tempat status dari panel LCD (*Liquid Cristal Display*) dapat dibaca pada saat pembacaan data.
- Register data yaitu register untuk menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada register akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

Pin, kaki atau jalur input dan kontrol dalam suatu LCD (*Liquid Crystal Display*) diantaranya adalah :

- Pin data adalah jalur untuk memberikan data karakter yang ingin ditampilkan menggunakan LCD (*Liquid Cristal Display*) dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.
- Pin RS (*Register Select*) berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika *low* menunjukan yang masuk adalah perintah, sedangkan logika *high* menunjukan data.
- Pin R/W (*Read Write*) berfungsi sebagai instruksi pada modul jika *low* tulis data, sedangkan *high* baca data.
- Pin E (*Enable*) digunakan untuk memegang data baik masuk atau keluar.
- Pin VLCD berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan dengan trimpot 10 Kohm, jika tidak digunakan dihubungkan ke ground, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

2.6 Buzzer

Buzzer adalah suatu alat yang dapat mengubah sinyal listrik menjadi sinyal suara terlihat dalam Gambar 2.7 bentuk komponen *buzzer*. Pada umumnya *buzzer* digunakan sebagai alarm karena penggunaannya yang cukup mudah yaitu dengan memberikan tegangan *input* maka *buzzer* akan mengeluarkan bunyi. Frekuensi suara yang di keluarkan oleh *buzzer* berkisar antara 1-5 KHz. (DAHKI, 2017)



Gambar. 2. 7 Buzzer

Sumber : <https://sariteknologi.com/product/buzzer/>

Prinsip kerja dari *buzzer* yaitu terdiri dari kumparan yang terpasang secara diafragma. Ketika kumparan tersebut dialiri listrik maka akan menjadi elektromagnet sehingga mengakibatkan kumparan tertarik ke dalam maupun ke luar tergantung dari arah arus dan polaritas magnetnya. Karena kumparan dipasang secara diafragma maka setiap

kumparan akan menggerakkan diafragma tersebut secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara.

2.7 Mikrokontroler ESP32

ESP32 adalah satu chip kombo 2,4 GHz Wi-Fi dan *Bluetooth* yang dirancang dengan teknologi daya ultra-rendah TSMC 40 nm seperti yang terlihat dalam Gambar 2.8. Alat ini dirancang untuk mencapai daya dan kinerja RF terbaik, menunjukkan ketahanan, keserbagunaan, dan keandalan dalam berbagai aplikasi dan skenario daya. (Espressif Systems, 2020)



Gambar. 2. 8 ESP32

Sumber : <https://www.mouser.co.id/ProductDetail/Espressif-Systems/ESP32-S2-Saola-1R?qs=GBLSI2AkirsMIxW%2FKjKKRw%3D%3D>

ESP32 menggunakan CMOS untuk radio dan pita dasar terintegrasi penuh chip tunggal, sementara juga mengintegrasikan sirkuit kalibrasi lanjutan yang memungkinkan solusi untuk menghilangkan ketidaksempurnaan sirkuit eksternal atau menyesuaikan dengan perubahan dalam kondisi eksternal. Dengan demikian, produksi massal solusi ESP32 tidak memerlukan peralatan pengujian Wi-Fi yang mahal dan terspesialisasi. (Espressif Systems, 2020)

ESP32 memiliki beberapa spesifikasi yang dapat dilihat dalam Tabel. 2,2 berikut :

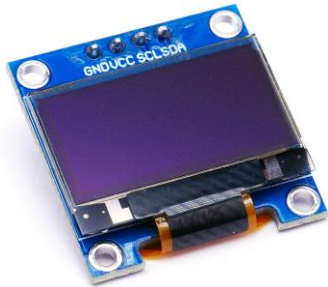
Tabel. 2. 2 Spesifikasi ESP32

Mikrokontroler	ESP32
Tegangan catu daya	1,8-3,6 Volt
Suhu operasi	-40sampai 125 °C
Antarmuka	34 GPIO, 12bit-SAR ADC sampai 18 channel, 2 x 8-bit DAC, 10 x sensor sentuh, 4 x SPI, 2 x I2S, 2 x I2C, 3 x UART.
Clock	8 MHz internal, 2 MHz – 60 MHz
Timer	2 x 64-bit timer, 1 x watchdog timer

Sumber : (Espressif Systems, 2020)

2.8 SSD1306 OLED Display

SSD1306 adalah *driver* CMOS OLED / PLED chip tunggal dengan pengontrol untuk sistem tampilan grafis dot-matrix dioda pemancar cahaya organik / polimer seperti yang terlihat dalam Gambar 2.9. SSD1306 terdiri dari 128 segmen dan 64 umum. IC ini dirancang untuk panel OLED tipe *Common Cathode*. (Solomon Systech Limited, 2008)



Gambar. 2. 9 SSD1306 OLED Display

Sumber : <https://esphome.io/components/display/ssd1306.html>

SSD1306 disematkan dengan kontrol kontras, *display RAM*, dan osilator, yang mengurangi jumlah komponen eksternal dan konsumsi daya. Chip ini memiliki kontrol kecerahan 256 langkah. Data / Perintah dikirim dari MCU umum melalui Antarmuka Paralel yang kompatibel dengan seri 6800/8000, antarmuka I2C atau Antarmuka Periferal Serial yang dapat dipilih perangkat keras. Sangat cocok untuk banyak aplikasi portabel yang ringkas, seperti sub-tampilan ponsel, pemutar MP3 dan kalkulator, dll. (Solomon Systech Limited, 2008)

2.9 LORA RFM95/96/97/98

Transceiver RFM95/96/97/98 (W) memiliki fitur modem jarak jauh LoRa yang menyediakan komunikasi spektrum penyebaran jarak jauh dan kekebalan interferensi tinggi sambil meminimalkan konsumsi saat ini seperti yang ditunjukkan dalam Gambar 2.10.



Gambar. 2. 10 RFM95W LoRa Module

Sumber : <https://www.hoperf.com/modules/lora/RFM95.html>

Menggunakan teknik modulasi LoRa yang dipatenkan dari Hope RF RFM95 / 96/97/98 (W) dapat mencapai sensitivitas lebih dari - 148dBm menggunakan kristal biaya rendah dan *bill of material*. Sensitivitas tinggi yang dikombinasikan dengan penguat daya +20 dBm terintegrasi menghasilkan anggaran tautan terdepan di industri sehingga optimal untuk aplikasi apa pun yang memerlukan jangkauan atau ketahanan. LoRaTM juga memberikan keuntungan signifikan dalam pemblokiran dan selektivitas dibandingkan teknik modulasi konvensional, menyelesaikan kompromi desain tradisional antara jarak, gangguan kekebalan dan konsumsi energi.

Perangkat ini juga mendukung mode FSK kinerja tinggi (G) untuk sistem termasuk WMBus, IEEE802.15.4g. RFM95 / 96/97/98 (W) memberikan gangguan fase yang luar biasa, selektivitas, linieritas penerima, dan IIP3 untuk konsumsi arus yang jauh lebih rendah daripada perangkat pesaing.

2.10 Aplikasi Blynk

Blynk dirancang untuk *Internet of Things*. Aplikasi ini dapat mengontrol perangkat keras dari jarak jauh, menyimpan data, menampilkan data sensor, mengirim data melalui email notifikasi, dan masih banyak lagi. Blynk merupakan platform sistem operasi Android OS version 4.2+ maupun iOS version 9+ sebagai kendali pada modul Arduino, Raspberry Pi, Arduino, NodeMCU, ESP8266 dan perangkat sejenis lainnya melalui internet. Ada tiga komponen utama dalam platform:

- Aplikasi Blynk - memungkinkan pengguna membuat antarmuka luar biasa untuk proyek yang menggunakan berbagai *widget* yang telah disediakan.
- *Blynk Server* - bertanggung jawab atas semua komunikasi antara smartphone dan perangkat keras. pengguna dapat menggunakan *Blynk Cloud* atau menjalankan server Blynk pribadi secara lokal. Blynk merupakan aplikasi open-source yang dapat dengan mudah menangani ribuan perangkat dan bahkan dapat diluncurkan di Raspberry Pi.
- *Blynk Library* - untuk semua platform perangkat keras populer - memungkinkan komunikasi dengan server dan memproses semua perintah yang masuk dan keluar.

(Blynk, 2021)

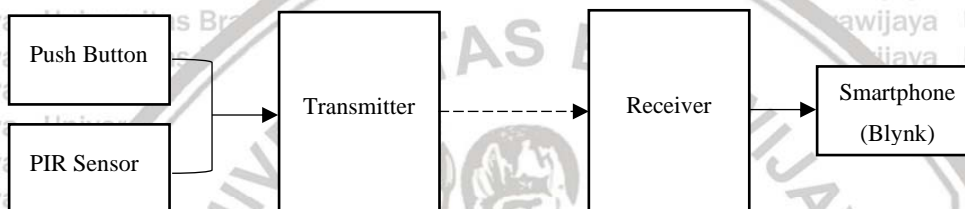
BAB III

METODE PENELITIAN

Metode penelitian yang dilakukan untuk menyelesaikan rumusan masalah dalam penelitian ini adalah menguraikan gambaran umum sistem, penentuan spesifikasi sistem, dan perancangan serta pengujian sistem keamanan menggunakan sensor PIR dan *panic button* berbasis LoRa dan aplikasi Blynk.

3.1 Deskripsi Umum Sistem

Deskripsi umum sistem berisi gambaran umum terkait rancangan alat yang dibuat dan bagaimana sistem bekerja secara sederhana serta mudah dipahami. Diagram blok dari deskripsi umum alat dapat dilihat dalam Gambar 3.1.



Gambar. 3.1 Diagram Blok Gambaran Umum Sistem

Sumber : (Dokumen Pribadi)

Berdasarkan Gambar. 3.1 bahwa alat yang digunakan untuk mendeteksi gerakan adalah sensor PIR. Sensor ini digunakan untuk mendeteksi gerakan apabila ada manusia yang mendekati sensor dengan jarak paling jauh 5 meter, maka sensor PIR akan mendeteksi adanya gerakan manusia kemudian mengirimkan sinyal ke mikrokontroler Arduino Uno pada *transmitter* untuk diproses. Saat pengguna merasa terancam atau terjadi keadaan darurat maka dapat menekan *panic button* yang selanjutnya akan mengirim sinyal bahaya melalui mikrokontroler Arduino Uno pada *transmitter*. Keadaan ini merupakan parameter mikrokontroler untuk memproses dan mengirim sinyal serta mengaktifkan alarm dan LED.

Pada *transmitter* selanjutnya akan mengirim data ke *receiver* melalui transmisi LoRa.

Mikrokontroler ESP32 pada *receiver* akan menerima sinyal dalam bentuk tipe data *String* dan OLED akan menampilkan text yang diterima. ESP32 yang terkoneksi internet akan terhubung dengan aplikasi Blynk dan selanjutnya mengirim data ke *smartphone* melalui aplikasi Blynk.

3.2 Spesifikasi Alat

Spesifikasi alat yang digunakan adalah sebagai berikut :

1. Mikrokontroler digunakan untuk pengolahan data.
2. Sensor *Passive InfraRed* (PIR) digunakan sebagai alat untuk mendeteksi objek bergerak (manusia).
3. *Push button* digunakan sebagai *panic button* ketika dalam keadaan darurat.
4. LoRa digunakan untuk mengirim data dari mikrokontroler Arduino Uno ke mikrokontroler ESP32.
5. LCD dan OLED *display* digunakan sebagai keluaran yang dapat menampilkan data informasi dan *user interface*.
6. *Buzzer* dan LED digunakan sebagai keluaran yang dapat mengindikasikan jika terjadi bahaya atau keadaan darurat.
7. Aplikasi Blynk sebagai keluaran yang digunakan untuk memantau data informasi dan *user interface*.

3.3 Studi Literatur

Tahapan ini dilakukan pengumpulan referensi untuk menentukan konsep penelitian yang akan dilakukan. Referensi yang diperlukan berupa kumpulan jurnal, *datasheet* dan prinsip kerja.

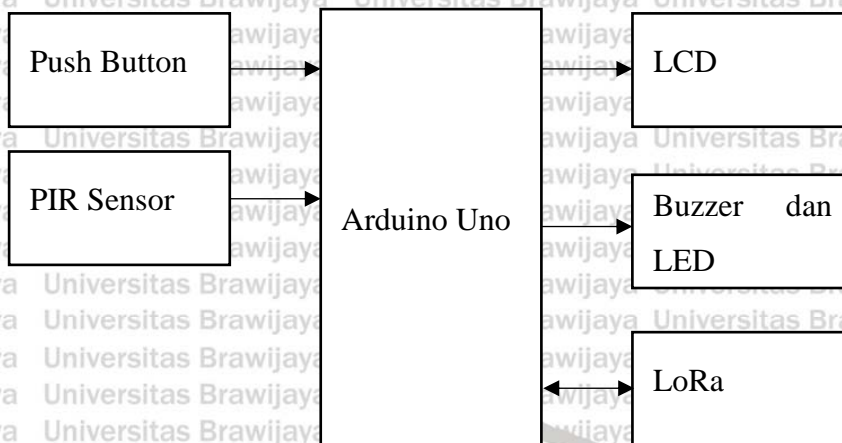
3.4 Perancangan Hardware

Perancangan *hardware* merupakan tahap untuk merakit dari pengumpulan komponen yang telah ditentukan. Dalam perancangan *hardware* harus memenuhi kriteria terhadap perancangan *software*. Setelah perancangan *hardware* maka selanjutnya dilakukan pengujian sistem yang dibuat.

Sistem alat pada penelitian ini terbagi menjadi dua bagian utama yaitu perangkat transmitter dan perangkat *receiver*. Perangkat *transmitter* akan menerima data sensor dan *push button* kemudian mengirimkannya ke perangkat *receiver* menggunakan transmisi LoRa. Perangkat *receiver* akan meneruskan data yang diterima ke *smartphone* untuk ditampilkan melalui blynk.

3.4.1 Perancangan Transmitter

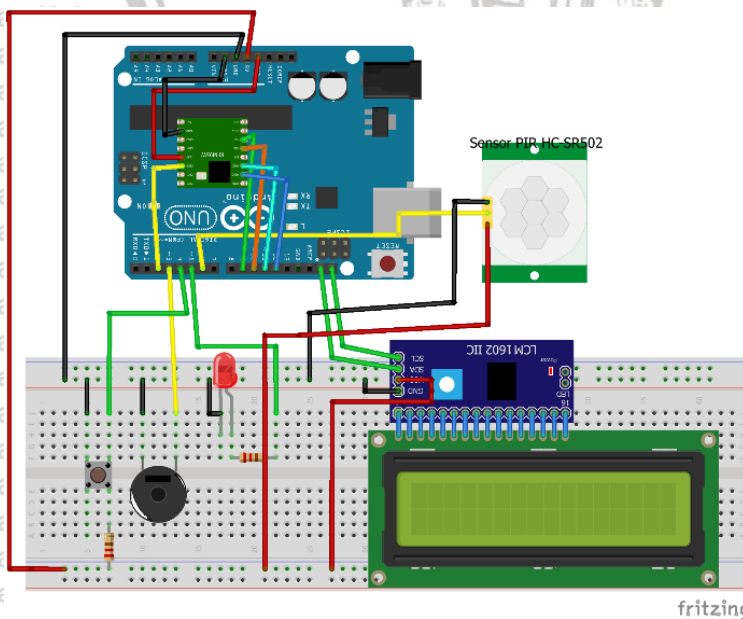
Perangkat *transmitter* dirancang menggunakan beberapa komponen elektronik utama yaitu meliputi : Mikrokontroler Arduino Uno dan Modul LoRa RFM95. Seluruh komponen utama dan beberapa komponen pendukung dirangkai pada papan PCB sesuai dengan rancangan yang tercantum pada Gambar 3.2.



Gambar. 3. 2 Diagram Blok Perangkat Transmitter

Sumber : (Dokumen Pribadi)

Perangkat *Transmitter* dalam Gambar 3.3 terdapat 1 buah mikrokontroler yaitu Arduino Uno dengan sensor PIR sebagai alat pendeteksi gerakan dan *push button* sebagai *panic button*. Terdapat komponen pendukung seperti LoRa, LCD, LED, dan *Buzzer* sebagai alat transmisi dan indikator. Mikrokontroler Arduino Uno digunakan sebagai otak dari sistem yang dihubungkan pada masing-masing komponen dengan diberi tegangan sebesar 5V pada sensor PIR, *Push Button*, LCD, LED, *Buzzer* namun pada LoRa diberi tegangan sebesar 3V.

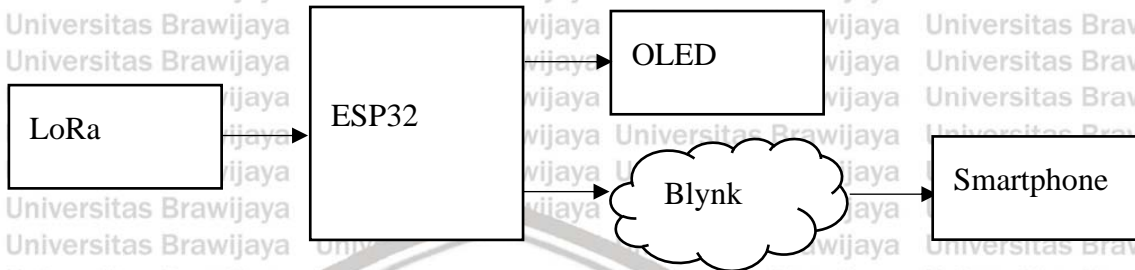


Gambar. 3. 3 Skema Transmitter

Sumber : (Dokumen Pribadi)

3.4.2 Perancangan Receiver

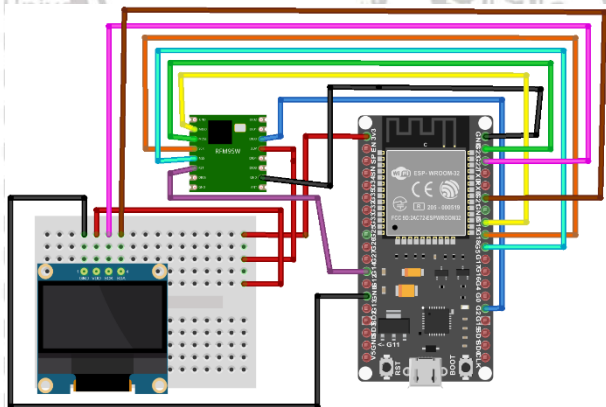
Perangkat *receiver* dirancang menggunakan komponen elektronik utama meliputi : Mikrokontroler ESP-32 dan Modul LoRa RFM95. Seluruh komponen utama dan beberapa komponen pendukung dirangkai pada papan PCB sesuai dengan rancangan yang tercantum pada Gambar 3.4.



Gambar. 3. 4 Diagram Blok Perangkat Transmitter

Sumber : (Dokumen Pribadi)

Perangkat *Receiver* dalam Gambar 3.5 terdapat 1 buah mikrokontroler yaitu ESP32 dengan fitur Wi-Fi sehingga dapat terkoneksi dengan aplikasi Blynk. Terdapat komponen pendukung seperti LoRa dan OLED *display* sebagai alat transmisi dan indikator. Mikrokontroler ESP32 digunakan sebagai otak dari sistem yang dihubungkan pada masing-masing komponen dengan diberi tegangan sebesar 3.3V pada LoRa dan OLED.



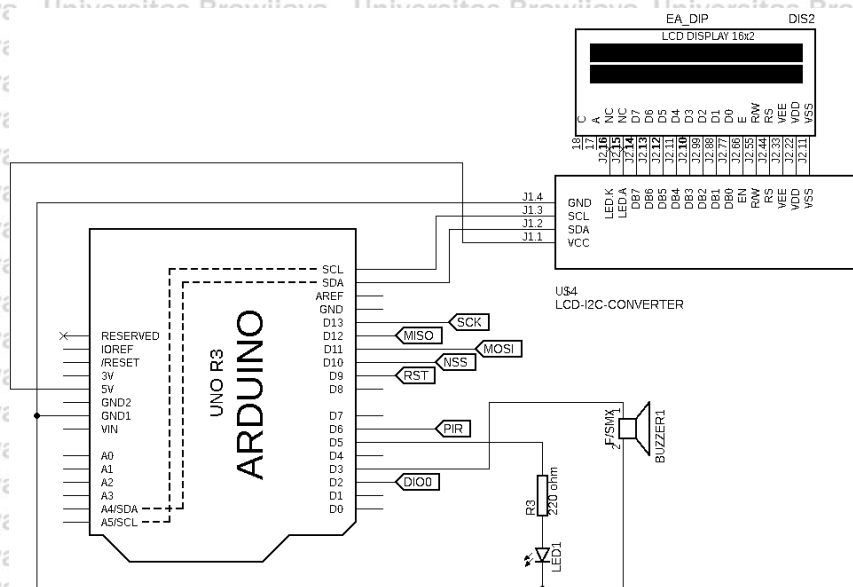
Gambar. 3. 5 Skema Receiver

Sumber : (Dokumen Pribadi)

3.5 Konfigurasi Rangkaian

Perangkat *Transmitter* menggunakan Arduino Uno sebagai Mikrokontroler sebagai pengolah data yang memiliki 14 *pin input/output digital* dan 6 *pin analog* yang dapat digunakan untuk menerima data hasil pembacaan sensor maupun mengaktifkan indikator

yang digunakan. Konfigurasi rangkaian *transmitter* menggunakan mikrokontroler Arduino Uno dapat dilihat dalam Gambar 3.6.



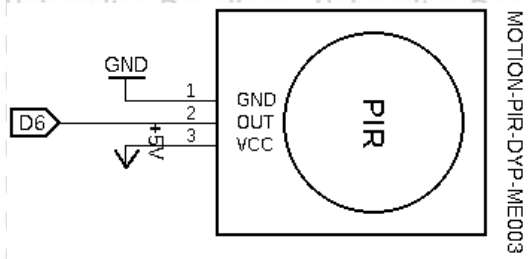
Gambar. 3. 6 Konfigurasi Rangkaian Transmitter

Sumber : (Dokumen Pribadi)

Pada rangkaian Arduino Uno dalam Gambar 3.6 terdapat rangkaian pendukung yang terhubung seperti LCD, I2C *converter*, *buzzer*, LED, dan resistor 220 Ω . I2C pada LCD digunakan untuk meminimalisir pin I/O yang digunakan pada LCD sehingga hanya menggunakan 4 pin yaitu pin VCC, GND, SCL, dan SDA. LED pada *transmitter* digunakan sebagai indikator saat *panic button* ditekan atau saat sensor PIR mendeteksi objek yang bergerak. LED terdapat 2 kaki, kaki polaritas negatif tersambung dengan GND dan kaki polaritas positif tersambung seri dengan resistor 220 Ω ke pin D5 pada Arduino Uno untuk menghambat arus sehingga tidak terjadi *short circuit*. *Buzzer* digunakan sebagai indikator saat terjadi keadaan darurat. *Buzzer* memiliki 2 pin yang tersambung dengan GND dan D3 pada Arduino Uno.

3.5.1 Konfigurasi Rangkaian PIR

Sensor PIR diimplementasikan pada perangkat *transmitter* sebagai masukan yang digunakan dalam mendeteksi objek manusia. Sensor PIR dipasang dengan pin digital Arduino Uno. Berikut adalah skema sensor PIR yang ditunjukkan dalam Gambar 3.7 terdapat 3 pin yaitu GND, VCC, dan Out. Pada pin GND Sensor PIR dihubungkan dengan pin Ground Arduino Uno, Pin VCC Sensor PIR dihubungkan dengan tegangan +5V pada Arduino Uno dan pin OUT dihubungkan dengan pin D6 Arduino Uno.

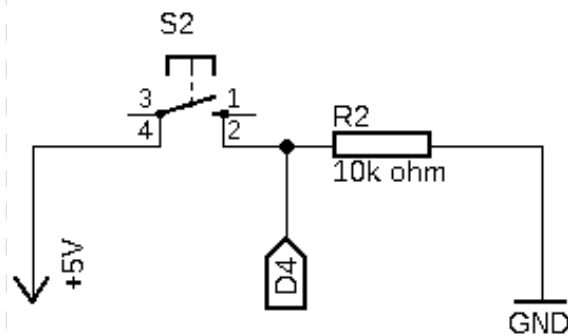


Gambar. 3. 7 Konfigurasi Rangkaian PIR

Sumber : (Dokumen Pribadi)

3.5.2 Konfigurasi Rangkaian Panic Button

Penggunaan *push button* pada sistem ini diimplementasikan pada perangkat *transmitter* sebagai *panic button* yang memudahkan pengguna dalam pengaktifan kerja alat untuk melaporkan kondisi darurat. *Push button* menggunakan sistem kerja *unlock* yang berarti tombol akan bekerja sebagai *device* penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka tombol akan kembali pada kondisi normal. Pada Gambar 3.8 Terdapat kabel GND yang tersambung dengan 10k *pull down resistor*, pin 1 tersambung dengan pin D4 pada Arduino Uno, pada pin 4 tersambung dengan pin VCC/+5V pada Arduino Uno. *Pull down resistor* dapat memberikan pembacaan *LOW* saat tombol *switch* tidak ditekan. Ketika tombol *switch* ditekan maka 10k resistor akan dilewati dan pembacaan akan menjadi *HIGH*.

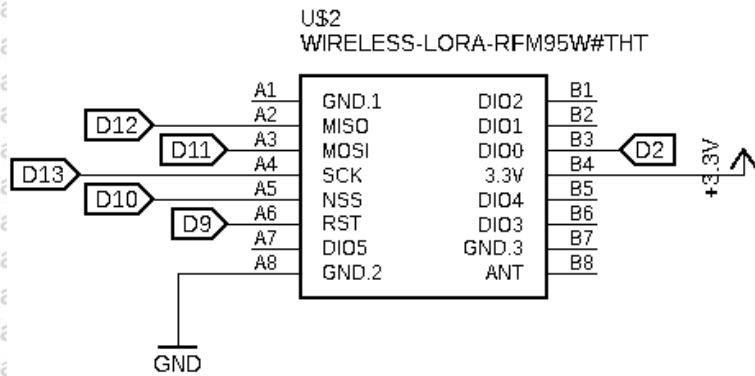


Gambar. 3. 8 Konfigurasi Rangkaian Panic Button

Sumber : (Dokumen Pribadi)

3.5.3 Konfigurasi LoRa Transmitter

Transmisi LoRa digunakan sebagai alat mengirim data sensor dari mikrokontroler Arduino Uno ke mikrokontroler ESP32. Modul LoRa RFM95 memiliki 16 pin dan bekerja dengan tegangan 3,3V sehingga modul LoRa dihubungkan ke tegangan 3,3V pada Arduino Uno terlihat dalam Gambar 3.9.



Gambar. 3. 9 Konfigurasi Rangkaian LoRa

Sumber : (Dokumen Pribadi)

Konfigurasi pin LoRa RFM95 menggunakan mikrokontroler Arduino Uno dapat dilihat dalam Tabel 3.1.

Tabel. 3. 1 Konfigurasi Pin LoRa

Pin LoRa	Pin Arduino Uno
3,3 V	3,3 V
GND	GND
DIO0	D2
RST	D9
NSS	D10
MOSI	D11
MISO	D12
SCK	D13

Sumber : (Dokumen Pribadi)

3.5.4 Konfigurasi LoRa Receiver

Transmisi LoRa pada *receiver* digunakan sebagai alat menerima data sensor dari mikrokontroler Arduino Uno ke mikrokontroler ESP32. Modul LoRa RFM95 memiliki 16 pin dan bekerja dengan tegangan 3,3V sehingga modul LoRa dihubungkan ke tegangan 3,3V pada ESP32 terlihat dalam Tabel. 3.2.

Tabel. 3. 2 Konfigurasi Pin LoRa Receiver

Pin LoRa	Pin ESP32
3,3 V	3,3 V
GND	GND
DIO0	GPIO2
RST	GPIO14
NSS	GPIO5
MOSI	GPIO23
MISO	GPIO19
SCK	GPIO18

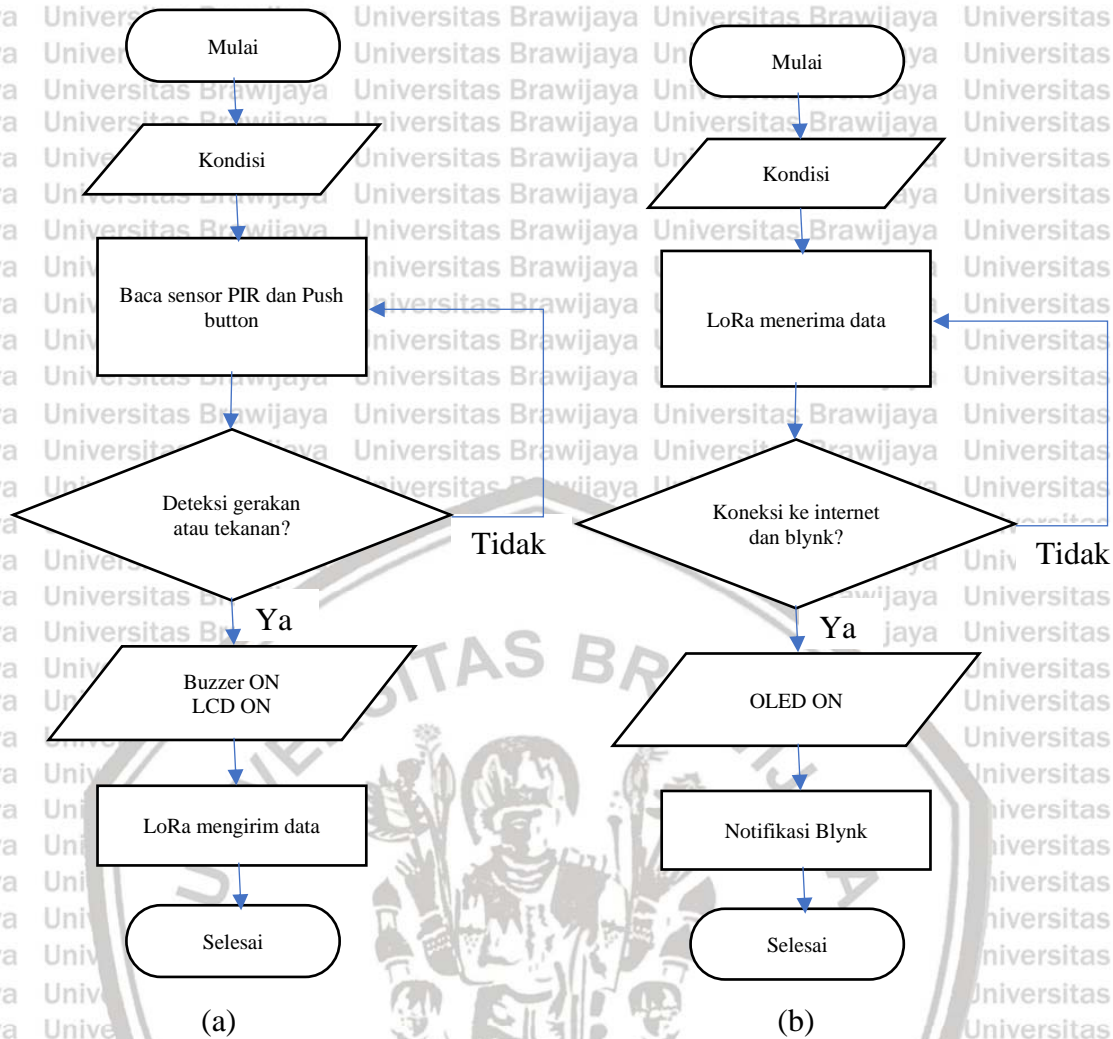
Sumber : (Dokumen Pribadi)

3.6 Perancangan Software

Dalam perancangan *software* ini dilakukan pada rangkaian mikrokontroler dengan operasi Software IDE Arduino. *Software* ini bekerja dengan berbasiskan pemrograman IDE. Pemrograman dilakukan dengan mengetikkan kode program berupa instruksi-instruksi yang akan mengaktifkan pin-pin yang terdapat pada mikrokontroler. Program yang dimasukkan ke rangkaian mikrokontroler yaitu :

1. Mendeteksi pergerakan manusia oleh sensor PIR.
2. *Push button* dapat mengirimkan sinyal bahaya saat ditekan.
3. Menampilkan data informasi melalui LCD dan OLED *display*.
4. Mengirim data informasi secara *online* melalui aplikasi Blynk.

Berikut *flowchart* gambaran umum sistem yang menggambarkan bagaimana sistem bekerja secara sederhana dan mudah dipahami ditunjukkan dalam Gambar 3.10 (a) dan (b).



Gambar. 3. 10 Diagram Alir Sistem untuk (a) Alat Transmitter, (b) Alat Receiver

Sumber : (Dokumen Pribadi)

3.7 Pengujian Alat dan Sistem

Pengujian alat dan sistem dilakukan untuk melihat hasil peralatan yang dibuat apakah sesuai dengan referensi dan konsep yang diinginkan. Pengujian alat ini dilakukan pada masing-masing bagian. Pengujian dilakukan secara bertahap kemudian dilakukan pengujian secara keseluruhan. Pengujian yang dilakukan sebagai berikut:

1. Pengujian Sub Sistem
2. Pengujian Seluruh Sistem

3.7.1 Pengujian Sub Sistem

1. Pengujian Sensor

Pengujian sensor PIR dilakukan untuk mengetahui sensitivitas sensor PIR, jarak terjauh serta sudut sensor dalam mendeteksi gerakan, dan *delay*. Pengujian ini dilakukan dengan mengukur jarak 1 – 7 m dan sudut 0° – 90° dengan memberikan sebuah gerakan di area deteksi sensor. Jika sensor mendeteksi gerakan maka LED dan LCD akan menyala sebagai keluaran indikasi bahaya.

2. Pengujian *Panic Button*

Pengujian *Panic button* dilakukan dengan cara menekan *push button* dengan rentang waktu yang ditentukan. Hasil pengujian ini untuk melihat apakah *panic button* bekerja saat ditekan atau tidak. Jika *push button* ditekan maka LED, LCD, dan *buzzer* akan menyala sebagai keluaran indikasi bahaya.

3. Pengujian Aplikasi Blynk

Pengujian ini bertujuan untuk mengetahui apakah aplikasi Blynk dapat menerima data dalam bentuk teks peringatan. Tampilan pada aplikasi akan memberikan informasi berupa kalimat “*motion detected*” atau alamat terjadinya keadaan darurat serta nilai RSSI dari transmisi LoRa. Pengujian ini dilakukan dengan cara membandingkan data yang dikirim dengan data yang diterima.

Hasil dari pembacaan sensor yang ditampilkan pada aplikasi berupa kalimat yang menandakan bahwa sensor mendeteksi gerakan. Hasil dari pembacaan *push button* berupa alamat terjadinya keadaan darurat.

3.7.2 Pengujian Seluruh Sistem

Pengujian seluruh sistem dilakukan dengan menjalankan seluruh sistem. Pengujian secara keseluruhan dimulai dari pembacaan sensor terhadap gerakan dan pengujian *push button* ketika ditekan. Setelah data diolah oleh arduino maka selanjutnya LoRa akan mengirim data tersebut ke penerima. ESP32 akan membaca dan mengirim data ke aplikasi Blynk sebagai *monitoring* sistem keamanan.

BAB IV

HASIL DAN PEMBAHASAN

Pada pembahasan dilakukan pengujian untuk mengetahui apakah perangkat keras dan perangkat lunak berfungsi dengan baik. Pengujian dilakukan pada Sensor PIR (*Passive Infrared*), *Panic Button*, dan Aplikasi Blynk. Pengujian dilakukan dengan beberapa variasi untuk mengetahui parameter terbaik yang dapat diterapkan. Berikut adalah beberapa aspek yang akan diuji :

1. Pengujian Sub Sistem

Sub sistem terbagi menjadi beberapa pengujian yaitu:

- a. Pengujian Sensor
- b. Pengujian *Panic Button*
- c. Pengujian Aplikasi Blynk

2. Pengujian Seluruh Sistem

4.1 Hasil dan Pembahasan Pengujian Sub Sistem

4.1.1 Pengujian Sensor

Prosedur pengujian sensor ini terdapat empat pengujian, pertama menentukan jarak pembacaan sensor terhadap gerakan dari berbagai macam objek. Kedua yaitu menentukan sudut pembacaan sensor terhadap gerakan. Ketiga yaitu menentukan kepekaan pembacaan sensor terhadap tegangan. Keempat yaitu menentukan nilai *delay* sensor saat mulai mendeteksi hingga berhenti mendeteksi.

4.1.1.1 Jarak Pembacaan Sensor

Pengujian pertama untuk sensor yaitu menentukan jarak pembacaan sensor terhadap gerakan agar pemasangan sensor dapat bekerja dengan baik saat terdapat keadaan bahaya.

Tujuan pengujian jarak ini agar pemasangan sensor terhadap gerakan dapat diperkirakan untuk mengoptimalkan pembacaan. Pengujian ini dilakukan dengan menggunakan beberapa

objek yaitu manusia, hewan, dan benda mati. Hasil pengujian dapat dilihat dalam Tabel 4.1

Hasil Pengujian Jarak Sensor PIR.

Tabel. 4. 1 Hasil Pengujian Jarak Sensor PIR

Jarak (m)	Manusia	Hewan	Benda Mati
1	Terdeteksi	-	-
2	Terdeteksi	-	-
3	Terdeteksi	-	-
4	Terdeteksi	-	-
5	Terdeteksi	-	-
6	Terdeteksi	-	-
7	-	-	-

Sumber : (Dokumen Pribadi)

Pada Tabel. 4.1 dapat dilihat pada jarak diatas 6 meter ditemukan hasil yang berbeda dari hasil percobaan sebelumnya, hasil percobaan jarak 1 m sampai 6 m dapat mendeteksi manusia sehingga area deteksi maksimal sensor hanya sampai 6 m. Hasil pengujian juga menunjukkan bahwa sensor PIR dapat bekerja dengan baik. Hasil percobaan menunjukkan sensor dapat mendeteksi gerakan manusia namun tidak dapat mendeteksi benda mati maupun hewan. Semakin jauh objek bergerak menjauhi sensor maka semakin kecil pula sinar inframerah yang dideteksi oleh sensor sehingga kepekaan sensor dalam mendeteksi menurun.

4.1.1.2 Sudut Pembacaan Sensor

Pengujian kedua untuk sensor yaitu menentukan sudut pembacaan sensor terhadap gerakan agar pemasangan sensor dapat bekerja dengan baik saat terdapat keadaan bahaya.

Tujuan pengujian sudut ini agar pemasangan sensor terhadap gerakan dapat diperkirakan untuk mengoptimalkan pembacaan. Pengujian ini dilakukan dengan memberikan masukan dari jarak 1 – 6 meter dan dengan sudut 0°, 30°, 45°, 60°. Hasil pengujian dapat dilihat dalam Tabel 4.2 Hasil Pengujian Sudut Sensor PIR.

Tabel. 4. 2 Hasil Pengujian Sudut Sensor PIR

Jarak (m)	Nilai Sudut			
	0°	30°	45°	60°
1	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
2	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
3	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
4	Terdeteksi	Terdeteksi	Terdeteksi	-
5	Terdeteksi	Terdeteksi	-	-
6	Terdeteksi	-	-	-

Sumber : (Dokumen Pribadi)

Pada Tabel 4.2 dapat dilihat bahwa nilai sudut dan jarak dalam pembacaan sensor PIR berpengaruh terhadap kepekaan pembacaan sensor. Pada nilai sudut 0 dapat mendeteksi objek yang bergerak sejauh 6 meter. Pada sudut 30 dapat mendeteksi objek yang bergerak sejauh 5 meter. Pada sudut 45 dapat mendeteksi objek yang bergerak sejauh 4 meter. Pada sudut 60 dapat mendeteksi objek yang bergerak sejauh 3 meter. Hal ini menandakan bahwa semakin besar nilai sudut maka semakin kecil jangkauan area deteksi sensor PIR dan semakin kecil pula pergerakan yang terbaca.

4.1.1.3 Kepekaan Pembacaan Sensor

Pengujian ketiga yaitu untuk menentukan kepekaan pembacaan sensor terhadap gerakan. Tujuan pengujian ini mengetahui pembacaan sensor yang ingin dideteksi dan daya tangkap sensor. Pengujian ini dilakukan dengan melakukan gerakan berpindah tempat dan diam ditempat. Hasil pengujian akan melihat apakah sensor aktif dan dapat mengirim data ke aplikasi blynk atau tidak. Hasil pembacaan dapat dilihat dalam Tabel 4.3 Hasil Deteksi Gerakan Sensor PIR.

Tabel. 4. 3 Hasil Deteksi Kepekaan Sensor PIR

Uji Coba	Waktu (detik)	Masukan	PIR	Hasil
1	16:28:26.315	Terjadi Gerakan	Aktif	Terkirim
2	16:28:26.362	Bergerak	Aktif	Terkirim
3	16:28:34.475	Diam Ditempat	Mati	Tidak Terkirim
4	16:28:44.520	Berhenti Bergerak	Mati	Tidak Terkirim
5	16:30:12.696	Terjadi Gerakan	Aktif	Terkirim
6	16:30:22.830	Diam Ditempat	Mati	Tidak Terkirim

Sumber : (Dokumen Pribadi)

Pada Tabel. 4.3 dapat dilihat saat terjadi gerakan maka PIR sensor akan aktif dan mengirim data. Selama masih terjadi pergerakan maka sensor PIR akan terus aktif sampai objek diam dan berhenti bergerak maka pir sensor akan mati dan tidak mengirim sinyal. Saat objek yang bergerak di area jangkauan PIR sensor diam ditempat maka PIR sensor tidak akan lagi mendeteksi karena tidak terjadi pergerakan. Hal ini menunjukkan PIR sensor hanya mendeteksi gerakan (perubahan) bukan tingkat IR rata-rata.

4.1.1.4 Delay Sensor PIR

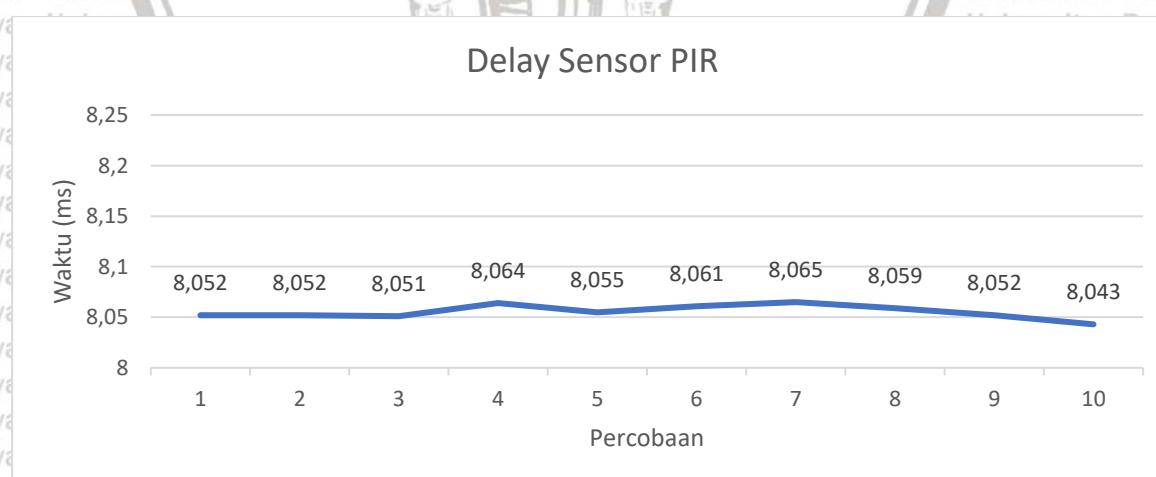
Pengujian Keempat yaitu untuk menentukan nilai *delay* sensor. Tujuan pengujian ini mengetahui berapa lama sensor PIR akan aktif setelah mendeteksi gerakan dan mengirimkan sinyal. Pengujian ini diambil 10 *sample* untuk dihitung nilai *delay* dengan waktu yang telah didapatkan. Hasil dari pengujian ini terdapat waktu dengan satuan detik dan nilai *delay* yang telah dihitung dengan membandingkan selisih dari waktu pengiriman data sensor PIR. Hasil pembacaan dapat dilihat dalam Tabel 4.4 Hasil Nilai *Delay* Sensor PIR.

Tabel. 4. 4 Hasil Nilai Delay Sensor PIR

Uji Coba	Waktu PIR Mengirim (detik)	Delay PIR Mengirim (detik)
1	18:49:07.609	8,052
2	18:49:15.661	8,052
3	18:49:23.712	8,051
4	18:49:31.776	8,064
5	18:49:39.831	8,055
6	18:49:47.892	8,061
7	18:49:55.957	8,065
8	18:50:04.016	8,059
9	18:50:12.068	8,052
10	18:50:20.111	8,043

Sumber : (Dokumen Pribadi)

Tabel 4.4 menunjukkan bahwa dari 10 percobaan didapatkan nilai *delay* sensor PIR yaitu 8,052 detik yang keluar sebanyak 3 kali. Hal ini menunjukkan bahwa selama sensor PIR mendeteksi gerakan maka sensor PIR akan mengirimkan data setiap 8,052 detik ke *receiver* sampai sensor PIR berhenti mendeteksi gerakan. *Delay* sensor PIR dalam mengirim data dapat di temukan dengan menghitung selisih waktu sensor PIR dalam mengirimkan data sehingga didapatkan hasil *delay* sensor seperti dalam Gambar 4.1. Grafik *Delay* Sensor PIR.

**Gambar. 4.1 Grafik Delay Sensor PIR**

Sumber : (Dokumen Pribadi)

Berdasarkan data yang tampak pada Gambar 4.1 Grafik *Delay* Sensor PIR, maka dapat diketahui nilai *error* dari nilai rata – rata *delay* terhadap nilai acuan yaitu 8,052 detik,

pengambilan nilai acuan 8,052 detik karena dalam data paling sering muncul sebanyak 3 kali.

Contoh perhitungan rata – rata dan *error* pada Sensor PIR :

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (4-1)$$

$$\bar{X} = \frac{8,052+8,052+8,051+8,064+8,055+8,061+8,065+8,059+790+8,052+8,043}{10}$$

$$\bar{X} = 8,055 \text{ detik}$$

$$\text{error} = \frac{|\bar{X} - X_{Set}|}{X_{Set}} \times 100\% \quad (4-2)$$

$$\text{error} = \frac{|8,055 - 8,052|}{8,052} \times 100\%$$

$$\text{error} = \frac{0,003}{8,052} \times 100\%$$

$$\text{error} = 0,422\%$$

Persamaan (4 - 1) dan (4 - 2) digunakan untuk menghitung nilai rata-rata *delay* sensor PIR dan *error* maka didapatkan nilai rata-rata *delay* sebesar 8,055 detik dengan nilai *error* 0,422%. Nilai *error* 0,422% menunjukkan seberapa besar perbedaan antara hasil nilai *time delay* yang keluar pada percobaan dengan nilai *time delay* estimasi melalui rata-rata. Hal ini memastikan bahwa nilai estimasi *delay* 8,055 detik meleset sebanyak 0,422% dari nilai asli 8,052 detik.

4.1.2 Pengujian *Panic Button*

Pengujian *panic button* dilakukan dengan cara menekan *push button* selama beberapa detik. Hasil pengujian akan mendeteksi apakah setelah ditekan atau tidak ditekan *push button* akan aktif dan mengirim data. Hasil pengujian dapat dilihat dalam Tabel 4.5 Hasil Deteksi *Panic Button*.

Tabel. 4.5 Hasil Deteksi *Panic Button*

Uji.Coba	<i>Panic Button</i>	Hasil
1	Ditekan selama 1 detik	Terkirim
2	Ditekan selama 2 detik	Terkirim
3	Ditekan Selama 3 detik	Terkirim
4	Tidak Ditekan	Tidak Terkirim
5	Tidak Ditekan	Tidak Terkirim

Sumber : (Dokumen Pribadi)

Pada Tabel. 4.5 dapat dilihat bahwa saat *panic button* ditekan mulai dari 1 detik hingga 3 detik maka akan aktif, namun jika tidak ditekan maka akan mati dan tidak mengirim data.

4.1.3 Pengujian Aplikasi Blynk

Pengujian aplikasi Blynk bertujuan mengetahui keberhasilan mengirimkan sinyal bahaya dari arduino pada ESP32 melalui transmisi LoRa. Tampilan pada aplikasi Blynk akan memberikan informasi berupa alamat terjadinya bahaya atau peringatan deteksi gerakan.

4.1.3.1 Presentase Keberhasilan

Pengujian Pertama ini dilakukan dengan cara membandingkan data yang dikirim dengan data yang diterima ditunjukkan dalam Gambar 4.2 (a) dan (b) pada serial monitor Arduino IDE.

(a) Transmitter Serial Monitor Output:

```

16:26:41.203 -> LoRa Sender
16:26:41.203 -> LoRa Initializing OK!
16:26:41.237 -> calibrating sensor ..... done
16:26:51.246 -> SENSOR ACTIVE
16:27:48.140 -> ---
16:27:48.140 -> motion detected at 68 sec
16:27:48.234 -> Sending packet: 1
16:27:56.328 -> Sending packet: 2
16:28:08.132 -> Sending packet: 3
16:28:21.238 -> motion ended at 96 sec
16:28:26.315 -> ---
16:28:26.315 -> motion detected at 106 sec
16:28:26.362 -> Sending packet: 4
16:28:34.475 -> Sending packet: 5
16:28:44.520 -> motion ended at 119 sec
16:28:50.565 -> ---
16:28:50.612 -> motion detected at 130 sec
16:28:50.659 -> Sending packet: 6
16:28:58.736 -> Sending packet: 7
16:29:12.186 -> Sending packet: 8
16:29:25.246 -> motion ended at 160 sec
16:29:31.979 -> ---
16:29:31.979 -> motion detected at 171 sec
16:29:32.025 -> Sending packet: 9
16:29:40.133 -> Sending packet: 10
16:29:49.441 -> Sending packet: 11
16:30:02.511 -> motion ended at 197 sec
16:30:12.696 -> ---
16:30:12.696 -> motion detected at 212 sec
16:30:12.790 -> Sending packet: 12
16:30:22.830 -> motion ended at 217 sec
16:30:26.220 -> ---
16:30:26.220 -> motion detected at 226 sec
16:30:35.592 -> motion ended at 230 sec
16:30:56.712 -> ---

```

(b) Receiver Serial Monitor Output:

```

16:27:39.142 -> LoRa Receiver
16:27:39.142 -> LoRa Initializing OK!
16:27:39.142 -> [2076] Connecting to Iocream
16:27:41.767 -> [4694] Connected to WiFi
16:27:41.767 -> [4694] IP: 192.168.43.9
16:27:41.767 -> [4694]
16:27:41.767 ->
16:27:41.767 ->
16:27:41.767 ->
16:27:41.814 -> / / v1.0.0-beta.3 on ESP32
16:27:41.814 ->
16:27:41.814 -> [4700] Connecting to blynk-cloud.com:80
16:27:41.907 -> [4829] Ready (ping: 49ms).
16:27:48.281 -> Received packet 2021-05-06T16:28:31Z
16:27:48.281 -> Motion Detected
16:27:48.421 -> with RSSI -26
16:27:56.375 -> Received packet 2021-05-06T16:28:39Z
16:27:56.375 -> Motion Detected
16:27:56.516 -> with RSSI -28
16:28:08.178 -> Received packet 2021-05-06T16:28:51Z
16:28:08.225 -> Motion Detected
16:28:08.366 -> with RSSI -26
16:28:26.409 -> Received packet 2021-05-06T16:29:09Z
16:28:26.455 -> Motion Detected
16:28:26.596 -> with RSSI -26
16:28:34.522 -> Received packet 2021-05-06T16:29:17Z
16:28:34.522 -> Motion Detected
16:28:34.663 -> with RSSI -29
16:28:50.706 -> Received packet 2021-05-06T16:29:33Z
16:28:50.706 -> Motion Detected
16:28:50.846 -> with RSSI -26
16:28:58.783 -> Received packet 2021-05-06T16:29:41Z
16:28:58.783 -> Motion Detected
16:28:58.924 -> with RSSI -29

```

(a)

(b)

Gambar. 4.2 Serial Monitor Pada (a) Transmitter, (b) Receiver

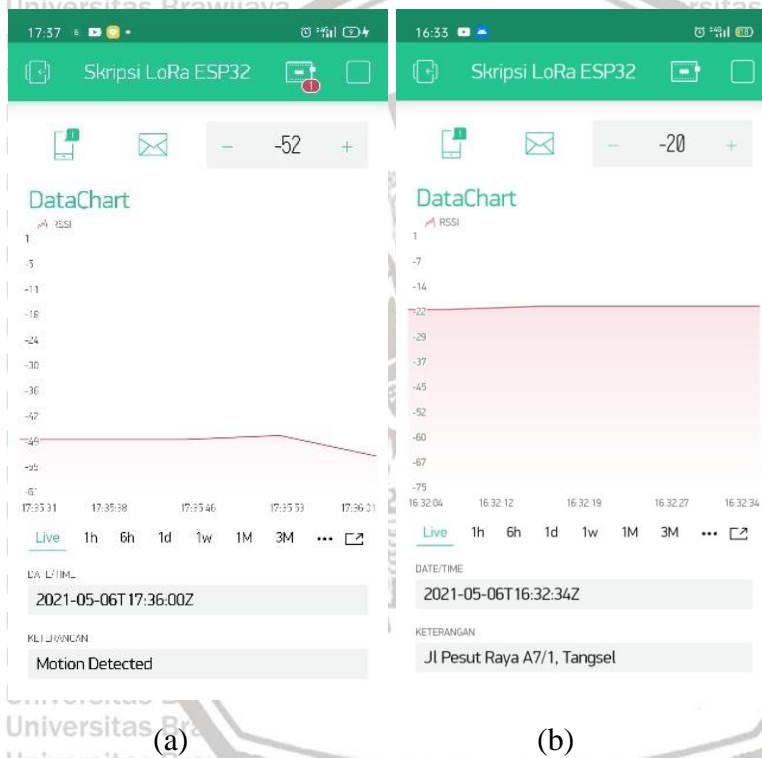
Sumber : (Dokumen Pribadi)

Berdasarkan Gambar 4.2 (a) pada *serial monitor transmitter* terdapat hasil dari percobaan kepekaan sensor PIR. Saat sensor PIR mendeteksi gerakan maka akan menampilkan notifikasi pada *serial monitor* bertuliskan “*motion detected at*” dan selanjutnya LoRa akan mengirim data dan muncul notifikasi bertuliskan “*sending packet*”.

Setelah objek berhenti bergerak maka sensor PIR akan berhenti mendeteksi dan *serial monitor* akan muncul notifikasi bertuliskan “*motion ended at*”.

Gambar 4.2 (b) pada *serial monitor receiver* selanjutnya akan menerima data yang dikirim dari sensor PIR dan akan muncul notifikasi bertuliskan “*received packet*” yang dilanjutkan dengan tanggal dan jam *packet* diterima. Setelah itu pada *serial monitor receiver* juga memunculkan data berupa string yang di kirim dari transmitter melali LoRA berupa tulisan “*motion detected*” dan nilai RSSI.

Gambar 4.3 merupakan tampilan notifikasi pada Blynk yang *terinstall* pada *smartphone*. Ketika sensor PIR mendeteksi adanya gerakan ditunjukkan dalam Gambar 4.3 (a) dan ketika *panic button* ditekan ditunjukkan dalam Gambar 4.3 (b).



Gambar. 4. 3 Hasil Pengiriman (a) Sensor PIR Pada Aplikasi Blynk, (b) Panic Button Pada Aplikasi Blynk

Sumber : (Dokumen Pribadi)

Gambar 4.3 merupakan tampilan dari hasil pengiriman sensor PIR dan *panic button* pada aplikasi blynk. Tampilan pada aplikasi blynk terdapat grafik dan nilai RSSI, tanggal dan jam paket data diterima, serta teks notifikasi “*motion detected*” dari sensor PIR dan “alamat” dari *panic button*.

Berdasarkan hasil pengujian Persentase keberhasilan dapat dibuat dalam bentuk persamaan (4 - 3) :

$$\frac{\text{Jumlah Data Dikirim}}{\text{Jumlah Data Diterima}} \times 100\% \quad (4-3)$$

$$\text{Jumlah Data Dikirim} = 10$$

$$\text{Jumlah Data Diterima} = 10$$

Maka :

$$\frac{10}{10} \times 100\% = 100\%$$

(4-4)

Dapat dilihat dalam persamaan (4 - 4) bahwa hasil pengujian pengiriman data menggunakan sensor PIR dan *push button* melalui LoRa ke aplikasi blynk bekerja dengan baik dengan presentase keberhasilan 100%.

4.1.3.2 Delay Blynk

Pengujian Kedua ini dilakukan untuk mengetahui nilai *time delay* pengiriman dari masukan ke aplikasi blynk dengan 10 kali percobaan. Pengujian dilakukan dengan mengukur waktu pengirim mengirimkan data dengan waktu blynk menerima data. Hasil pengujian dapat dilihat dalam Tabel 4.6 Hasil Nilai *Delay Blynk*.

Tabel. 4. 6 Hasil Nilai *Delay Blynk*

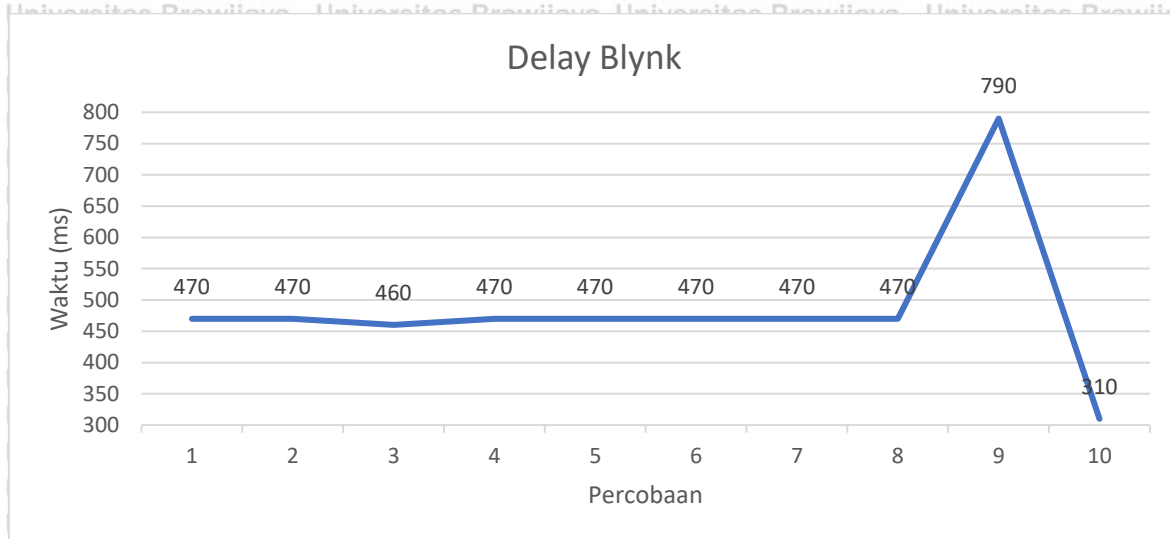
Uji Coba	Waktu Pengirim (detik)	Waktu Penerima (detik)	<i>Delay</i> (ms)
1	16:27:48.234	16:27:48.281	470
2	16:27:56.328	16:27:56.375	470
3	16:28:08.132	16:28:08.178	460
4	16:28:26.362	16:28:26.409	470
5	16:28:34.475	16:28:34.522	470
6	16:28:50.659	16:28:50.706	470
7	16:28:58.736	16:28:58.783	470
8	16:29:12.186	16:29:12.233	470
9	16:29:32.025	16:29:32.104	790
10	16:29:40.133	16:29:40.164	310

Sumber : (Dokumen Pribadi)

Berdasarkan Tabel 4.6 menunjukkan bahwa dari 10 percobaan didapatkan nilai *delay* pengiriman blynk yaitu 470 ms yang keluar sebanyak 7 kali dapat dilihat dalam Gambar 4.4.

Grafik *Delay Blynk*. Hal ini menunjukkan bahwa lama pengiriman dari transmitter ke

receiver adalah 470 ms. *Delay transmitter* mengirim data ke aplikasi blynk dapat di temukan dengan menghitung selisih waktu pengirim dengan waktu penerima.



Gambar. 4. 4 Grafik Delay Blynk

Dari data yang tampak pada Gambar 4.4 Grafik *Delay Blynk*, maka dapat diketahui nilai *error* dari nilai rata – rata *delay* terhadap nilai acuan yaitu 470 ms, pengambilan nilai acuan 470 ms karena dalam data paling sering muncul sebanyak 7 kali.

Contoh perhitungan rata – rata dan *error* pada Blynk :

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (4-5)$$

$$\bar{X} = \frac{470+470+460+470+470+470+470+470+790+310}{10}$$

$$\bar{X} = 485 \text{ ms}$$

$$\text{error} = \frac{|\bar{X} - X_{Set}|}{X_{Set}} \times 100\% \quad (4-6)$$

$$\text{error} = \frac{|485 \text{ ms} - 470 \text{ ms}|}{470 \text{ ms}} \times 100\%$$

$$\text{error} = \frac{15}{470} \times 100\%$$

$$\text{error} = 0,0319\%$$

Dari persamaan (4 - 5) dan (4 - 6) untuk menghitung nilai rata-rata *delay* Blynk dan *error* maka didapatkan nilai rata-rata *delay* sebesar 485 ms dengan nilai *error* 0,0319%.

Nilai *error* 0,0319% menunjukkan seberapa besar perbedaan antara hasil nilai *time delay* yang keluar pada percobaan dengan nilai *time delay* estimasi melalui rata-rata. Hal ini memastikan bahwa nilai estimasi *delay* 485 ms meleset sebanyak 0,0319% dari nilai asli 470 ms.

4.2 Hasil Dan Pembahasan Pengujian Seluruh Sistem

Pengujian keseluruhan sistem dilakukan untuk mengetahui apakah keseluruhan sistem dapat bekerja sesuai perancangan. Pengujian ini dilakukan pada beberapa kondisi sumber gerakan dan *push button* serta pengukuran jarak antara gerakan dan sensor sejauh 1 meter. Hasil pengujian dapat dilihat dalam Tabel 4.7 Hasil Pengujian Keseluruhan.

Tabel. 4. 7 Hasil Pengujian Keseluruhan

Parameter	Masukan	Kondisi	LED	LCD	Buzzer	Blynk
PIR sensor	Terjadi Gerakan jarak 1 meter	Aktif	Menyala	Menampilkan Peringatan	Mati	Terkirim
<i>Push Button</i>	Ditekan	Aktif	Menyala	Menampilkan Peringatan	Menyala	Terkirim

Sumber : (Dokumen Pribadi)

Berdasarkan Tabel 4.7 menunjukkan bahwa semua proses bekerja. Sensor dapat mendeteksi gerakan dan *Push button* dapat menyala ketika ditekan. Pada saat sensor mendeteksi gerakan maka akan menyalakan LED dan LCD pada Arduino akan menampilkan informasi adanya bahaya. Setelah itu aplikasi Blynk dapat menerima data yang telah dikirim.

Pada saat *push button* ditekan maka akan menyalakan LED, buzzer, dan LCD pada Arduino akan menampilkan informasi adanya bahaya. Setelah itu aplikasi Blynk dapat menerima data yang telah dikirim.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil percobaan dapat ditarik kesimpulan sebagai berikut :

1. Berdasarkan hasil pengujian, sensor PIR dapat mendeteksi manusia yang bergerak pada jangkauan jarak sejauh 6 meter.
2. Semakin lebar sudut jangkauan objek bergerak terhadap sensor PIR maka semakin kecil jangkauan deteksi sensor PIR.
3. *Delay* sensor PIR dapat aktif setelah mendeteksi gerakan adalah 8,055 detik dan *error* 0,422%.
4. Berdasarkan hasil pengujian *panic button* dapat mengirim sinyal saat ditekan lebih dari 1 detik dan tidak akan mengirim sinyal saat tidak ditekan.
5. Aplikasi Blynk yang terhubung ke internet dapat menampilkan hasil informasi ketika *panic button* ditekan dan ketika sensor PIR mendeteksi gerakan dengan *delay* pengiriman 485 ms dan *error* 0,0319%.

5.2 Saran

Saran yang dapat digunakan dalam meningkatkan kerja sistem antara lain :

1. Alat dapat dikembangkan dengan menambahkan kamera CCTV untuk memantau Gerakan objek.
2. Sebaiknya pemasangan sensor PIR diatas 1 hingga 2 meter sehingga lebih optimal mendeteksi Gerakan manusia.
3. Sistem keamanan ini dipasang di lorong atau depan pintu masuk dan keluar rumah.

DAFTAR PUSTAKA

Adafruit Industries . (2020). *PIR Motion Sensor*. Retrieved from Adafruit Learning System.

Badan Pusat Statistik. (2020). *Statistik Kriminal 2020*. Retrieved from Badan Pusat Statistik:

<https://www.bps.go.id/publication/2020/11/17/0f2dfc46761281f68f11afb1/statistik-kriminal-2020.html>

Biro Hukum dan Humas, Badan Urusan Administrasi, Mahkamah Agung RI. (2021). *Kitab Undang-Undang Hukum Pidana*. Retrieved from Jaringan Dokumentasi dan

Informasi Hukum (JDIH) Mahkamah Agung RI: <https://jdih.mahkamahagung.go.id/index.php/hukum-acara/4.-Hukum-Acara/Kitab-Undang-Undang-Hukum/>

Blynk. (2021). *Blynk*. Retrieved from Blynk.io: <https://docs.blynk.cc/#intro>

DAHKI, R. H. (2017). *SISTEM PEMANTAU RUANG JARAK JAUH MENGGUNAKAN SENSOR PIR (PASSIVE INFRARED) BERBASIS ATMEGA 8535*. MEDAN: UNIVERSITAS SUMATERA UTARA.

Espressif Systems. (2020). *ESP32 Series Datasheet*.

farnell. (2021). *Arduino Uno Datasheet*. Retrieved from farnell.com: Farnell.com

HOPE Microelectronics CO., Ltd. (n.d.). *RFM95W LORA MODULE*. Retrieved from <https://www.hoperf.com/>: <https://www.hoperf.com/modules/lora/RFM95.html>

SAHIDIN, D. (2016). *SISTEM KENDALI LIFT 3 LANTAI MENGGUNAKAN PLC TWIDO*. UNIVERSITAS MUHAMMADIYAH PURWOKERTO.

Solomon Systech Limited. (2008). *SSD1306 Datasheet*.

Soyer, E. B. (2009). *Pyroelectric Infrared (PIR) Sensor Based Event Detection*. Turkey: The Institute of Egeering and Siences Bilkent University.

Taryudi, Adriano, D. B., & Budi, W. A. (2018). Iot-based Integrated Home Security and Monitoring System. *Journal of Physics: Conference Series*.

Tempongbuka, H., Drs. Elia Kendek Allo, M., & Sherwin R. U. A. Sompie, S. M. (2015). Rancang Bangun Sistem Keamanan Rumah Menggunakan Sensor PIR (Passive Infrared) Dan SMS Sebagai Notifikasi . *E-Journal Teknik Elektro dan Komputer Vol. 4 No. 6*, 10-15.

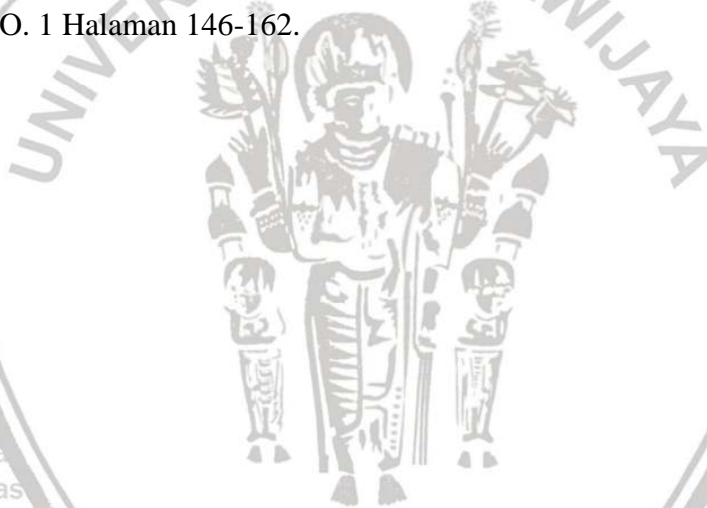
Toyib, R., Bustami, I., Abdullah, D., & Onsardi. (2019). PENGGUNAAN SENSOR PASSIVE INFRARED RECEIVER (PIR) UNTUK MENDETEKSI GERAK BERBASIS SHORT MESSAGE SERVICE GATEWAY. *Jurnal Pseudocode*, 114-124.

Waworundeng, J., Irawan, L. D., & Pangalila, C. A. (2017). Implementasi Sensor PIR sebagai Pendeteksi Gerakan untuk Sistem Keamanan Rumah menggunakan Platform IoT. *Cogito Smart Journal*, 152-163.

Wibowo, P. (2018). *PERANCANGAN SISTEM KEAMANAN RUMAH MENGGUNAKAN SENSOR PIR BERBASIS MIKROKONTROLER*. Medan: Universitas Pembangunan Panca Budi.

WIJAYA, W. F. (2017). *ALAT PENDETEKSI MALING PADA RUMAH MENGGUNAKAN SENSOR PASSIVE INFRARED BERBASIS ARDUINO*. MEDAN: UNIVERSITAS SUMATERA UTARA.

Zain, R. H. (2013). SISTEM KEAMANAN RUANGAN MENGGUNAKAN SENSOR PASSIVE INFRA RED (PIR) DILENGKAPI KONTROL PENERANGAN PADA RUANGAN BERBASIS MIKROKONTROLER ATmega8535 DAN REAL TIME CLOCK DS1307. *JURNAL TEKNOLOGI INFORMASI & PENDIDIKAN*, VOL. 6 NO. 1 Halaman 146-162.





LAMPIRAN



Lampiran 1 Implementasi Transmitter



Lampiran 2 Implementasi Receiver



Lampiran 5 Listing Program Pada Transmitter

```

#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#define rst 9
#define dio0 2
#define NSS 10
#define MOSI 11
#define MISO 12
#define SCK 13

const int buzzer = 3; //Buzzer Pin
const int panicButton = 4; // Push-button pin
const byte redledPin = 5; // Red LED for panic button
int pirPin = 6; //the digital pin connected to the PIR sensor's output
int counter;
int panicState; // To save the last logic state of the button
String Alamat;

//the time we give the sensor to calibrate (10-60 secs according to the datasheet)
int calibrationTime = 10;

//the time when the sensor outputs a low impulse
long unsigned int lowIn;

//the amount of milliseconds the sensor has to be low
//before we assume all motion has stopped
long unsigned int pause = 5000;

```

```

boolean lockLow = true;
boolean takeLowTime;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();

  while (!Serial);
  Serial.println("LoRa Sender");

  if (!LoRa.begin(915E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  LoRa.setSyncWord(0xA5);
  Serial.println("LoRa Initializing OK!");

  pinMode(panicButton,INPUT);
  pinMode(buzzer, OUTPUT); // set up pin as output
  pinMode(redledPin, OUTPUT); // set up pin as output
  digitalWrite(redledPin, LOW); // set pin value LOW

  pinMode(pirPin, INPUT);
  digitalWrite(pirPin, LOW);

  //give the sensor some time to calibrate
  Serial.print("calibrating sensor ");
  lcd.setCursor(0,0);
  lcd.print("calibrating");
  lcd.setCursor(0,1);
  lcd.print("sensor");

```



```

for(int i = 0; i < calibrationTime; i++){
  Serial.print(" ");
  delay(1000);
}

Serial.println(" done");
Serial.println("SENSOR ACTIVE");
lcd.clear();
}

void loop() {
  // put your main code here, to run repeatedly:

  lcd.setCursor(0,0);
  lcd.print("Sistem Keamanan");
  lcd.setCursor(0,1);
  lcd.print("Aktif");

  //panic button

  panicState = digitalRead(panicButton); //Put the reading value of the switch on botao1

  if (panicState == 0) // Pressed button, logic State HIGH (5V)
  {
    lcd.clear();
    Serial.println(panicState); //Shows the logic state of the input on Serial Monitor
    digitalWrite(buzzer,1); //Switch pressed, buzzer on
    tone(buzzer,100,3000);
    digitalWrite(redledPin, HIGH); // Turn the redledPin on
  }

  if (panicState == 0) // Pressed button, logic State HIGH (5V)

```



```

lcd.setCursor(0,0);
lcd.print("INTERRUPT");
lcd.setCursor(0,1);
lcd.print(" Aktif");
delay(3000);
lcd.clear();
digitalWrite(redledPin,LOW);
}
//PIR sensor
if(digitalRead(pirPin) == HIGH){
  lcd.clear();
  digitalWrite(redledPin, HIGH); //the led visualizes the sensors output pin state
  lcd.setCursor(0,0);
  lcd.print("Motion Detected");
  delay(3000);
  lcd.clear();
  if(lockLow){
    //makes sure we wait for a transition to LOW before any further output is made:
    lockLow = false;
    Serial.println("---");
    Serial.print("motion detected at ");
    Serial.print(millis()/1000);
    Serial.println(" sec");
    delay(50);
  }
  takeLowTime = true;
}

```

```

if(digitalRead(pirPin) == LOW){
  digitalWrite(redLedPin, LOW); //the led visualizes the sensors output pin state
  noTone(buzzer);
  if(takeLowTime){
    lowIn = millis(); //save the time of the transition from high to LOW
    takeLowTime = false; //make sure this is only done at the start of a LOW phase
  }
  //if the sensor is low for more than the given pause,
  //we assume that no more motion is going to happen
  if(!lockLow && millis() - lowIn > pause){
    //makes sure this block of code is only executed again after
    //a new motion sequence has been detected
    lockLow = true;
    Serial.print("motion ended at "); //output
    Serial.print((millis() - pause)/1000);
    Serial.println(" sec");
    delay(50);
  }
}
//for Lora Sender
if ((panicState == 0)|| (digitalRead(pirPin) == HIGH)) {
  lcd.setCursor(0,0);
  lcd.print("Sending packet");
  counter++;
  Serial.print("Sending packet: ");
  Serial.println(counter);
  LoRa.println(counter);
  if (panicState == 0) {
    // send packet

```

```
LoRa.beginPacket();
LoRa.println("JI Pesut Raya A7/1, Tangsel");
LoRa.endPacket();
}

if(digitalRead(pirPin) == HIGH){
  // send packet
  LoRa.beginPacket();
  LoRa.println("Motion Detected");
  LoRa.endPacket();
}
delay(5000);
}
```



Lampiran 6 Listing Program Pada Receiver

```

#include <SPI.h>
#include <LoRa.h>

#define BLYNK_PRINT Serial
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

//define the pins used by the transceiver module
#define ss 5
#define rst 14
#define dio0 2

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "MLtM_aBRy4IG6WB6lqSY3t8CuwoVcJ3-";
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Icecream";
char pass[] = "orekihyouka";
int rssi;
String LoRaData;

```

```

// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

// Variables to save date and time
String formattedDate;

void setup() {
  //initialize Serial Monitor
  Serial.begin(115200);
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
  {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  delay(2000);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 0);
  display.println("LoRa Receiver");
  display.display();

  while (!Serial);

  Serial.println("LoRa Receiver");

  LoRa.setPins(ss, rst, dio0); //setup LoRa transceiver module

  while (!LoRa.begin(915E6)) //433E6 - Asia, 866E6 - Europe, 915E6 - North America
  {
    Serial.println(".");
    delay(500);
  }
}

```



```

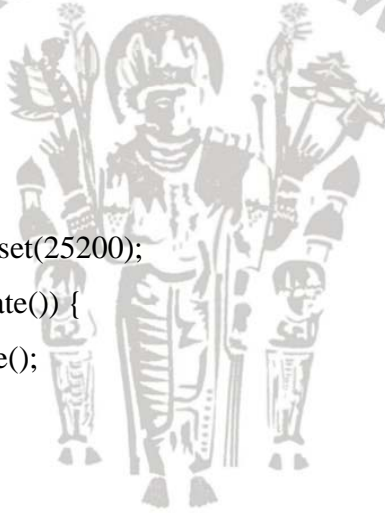
LoRa.setSyncWord(0xA5);
Serial.println("LoRa Initializing OK!");

Blynk.begin(auth, ssid, pass);
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 10);
display.println("Ready");
display.display();

// Initialize a NTPClient to get time
timeClient.begin();
// Set offset time in seconds to adjust for your timezone, for example:
// GMT +1 = 3600
// GMT +8 = 28800
// GMT -1 = -3600
// GMT 0 = 0
timeClient.setTimeOffset(25200);
while(!timeClient.update()) {
timeClient.forceUpdate();
}
}

void loop() {
formattedDate = timeClient.getFormattedDate();
Blynk.run();
// try to parse packet
int packetSize = LoRa.parsePacket();
if (packetSize) {
// received a packet
Serial.print("Received packet ");
Serial.println(formattedDate);
}
}

```



```

display.clearDisplay();
display.setTextSize(1);
display.setCursor(0,10);
display.setTextColor(WHITE);
display.println(formattedDate);
display.display();

// read packet
while (LoRa.available()) {
  LoRaData = LoRa.readString();
  Serial.print((LoRaData));
  Blynk.notify("HELP!! " + (LoRaData));
  Blynk.virtualWrite(V2,LoRaData);
  Blynk.virtualWrite(V3,formattedDate);

  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 30);
  display.println(LoRaData);
}

// print RSSI of packet
Serial.print("with RSSI ");
Serial.println(LoRa.packetRssi());
rssi = LoRa.packetRssi();
Blynk.virtualWrite(V1,rssi);
display.print("RSSI: ");
display.println(rssi);
display.display();
}
}


```

Lampiran 7 Datasheet


★ adafruit learning system

Overview

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.



PIRs are basically made of a [pyroelectric sensor](https://adafru.it/aK0r) (which you can see below as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

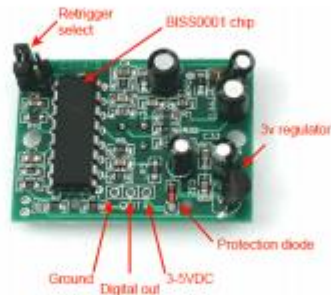


Along with the pyroelectric sensor is a bunch of supporting circuitry, resistors and capacitors. It seems that most small hobbyist sensors use the [BISS0001 \("Micro Power PIR Motion Detector IC"\)](https://adafruit.com/products/109), undoubtedly a very inexpensive chip. This chip takes the output of the sensor and does some minor processing on it to emit a digital output pulse from the analog sensor.

© Adafruit Industries <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor> Page 3 of 28



Our older PIRs looked like this:



Our new PIRs have more adjustable settings and have a header installed in the 3-pin ground/out/power pads.



For many basic projects or products that need to detect when a person has left or entered the area, or has approached, PIR sensors are great. They are low power and low cost, pretty rugged, have a wide lens range, and are easy to interface with. Note that PIRs won't tell you how many people are around or how close they are to the sensor, the lens is often fixed to a certain sweep and distance (although it can be hacked somewhere) and they are also sometimes set off by housepets. Experimentation is key!

Some Basic Stats

These stats are for the PIR sensor in the Adafruit shop which is very much like the Parallax one (<https://adafru.it/aKQ>).

© Adafruit Industries. <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor> Page 4 of 28

Nearly all PIRs will have slightly different specifications, although they all pretty much work the same. If there's a datasheet, you'll want to refer to it

- **Size:** Rectangular
- **Price:** \$10.00 at the Adafruit shop (<https://adafru.it/aIH>)
- **Output:** Digital pulse high (3V) when triggered (motion detected) digital low when idle (no motion detected). Pulse lengths are determined by resistors and capacitors on the PCB and differ from sensor to sensor.
- **Sensitivity range:** up to 20 feet (6 meters) 110° x 70° detection range
- **Power supply:** 5V-12V input voltage for most modules (they have a 3.3V regulator), but 5V is ideal in case the regulator has different specs
- **BIS0001 Datasheet** (<https://adafru.it/cIR>) (the decoder chip used)
- **RE200B datasheet** (<https://adafru.it/cIS>) (most likely the PIR sensing element used)
- **NL11NH datasheet** (<https://adafru.it/cIT>) (equivalent lens used)
- **Parallax Datasheet on their version of the sensor** (<https://adafru.it/cIU>)

More links!