

Perancangan Sistem Pendeteksi Dan Penanggulangan Banjir Menggunakan ESP32 Berbasis IoT

Yuslan Basir^{1*}, M. Rizki Adi Pratama², Moh. Wahyu Aminullah³

^{1,2,3} Program Studi Teknik Elektro, Universitas Tridinanti, Palembang 30129

*Korespondensi penulis: yuslan@univ-tridinanti.ac.id

(Received: 02-02-2023; Revised: 31-05-2023; Accepted: 31-06-2023)

Abstract. Design of a flood detection system using IoT-based ESP32 which functions to monitor water level heights and can be controlled by a smartphone with the blynk application. This tool is designed using ESP32 as a Wi-Fi module, a water level sensor to determine the water discharge level and a relay to function as a switch. The way this tool works is to detect the arrival of floods through a level sensor and the data is directed to the ESP32 microcontroller as a medium and sent by the output indicator lights, LCD displays and smartphones in the blynk application called flood monitoring. The test results use a water level sensor with a variation of 3 levels, namely level 1, level 2, and level 3. The indicator light uses 3 colours, namely green, yellow, and red. The measurement results on the power supply by a multimeter obtained a value of 8.97 V and the calculation results obtained a value of 8.10 V so the difference of 0.87 V is still a reasonable tolerance. If the sensor reaches level 3 it will send a notification in the form of a notification to blynk so that the red indicator light will turn on, and the pump will automatically turn on. The flood detection and control system operates well after a trial of flood monitoring and control through the blynk application.

Keywords: monitoring, sensor level, ESP32, smartphone, blynk.

Abstrak. Perancangan sistem pendeteksi banjir menggunakan ESP32 berbasis IoT yang berfungsi untuk memantau ketinggian level air dan dapat di control melalui smartphone dengan aplikasi blynk. Alat ini dirancang menggunakan ESP32 sebagai modul Wi-Fi, sensor level air untuk menentukan ketinggian debit air serta relay berfungsi sebagai switch. Cara kerja alat ini adalah mendeteksi datangnya banjir melalui sensor level dan data tersebut di arahkan pada mikrokontroler ESP32 sebagai media dan dikirim melalui output indikator lampu, LCD display dan smartphone di aplikasi blynk yang bernama monitoring banjir. Hasil pengujian menggunakan sensor level air dengan variasi 3 level yaitu level 1, level 2, dan level 3. Lampu indikator menggunakan 3 warna yaitu hijau, kuning, dan merah. Hasil pengukuran di catu daya melalui multimeter didapatkan nilai 8.97 V dan hasil perhitungan didapatkan nilai 8.10 V sehingga selisih 0.87 V masih toleransi wajar. Jika sensor mencapai level 3 akan mengirimkan notifikasi berupa pemberitahuan ke blynk sehingga lampu indikator merah akan menyala, dan pompa akan otomatis hidup. Sistem pendeteksi dan pengendalian banjir beroperasi dengan baik setelah dilakukan uji coba monitoring dan pengendalian banjir melalui aplikasi blynk.

Kata Kunci: monitoring, sensor level, ESP32, smartphone, blynk.

PENDAHULUAN

Indonesia merupakan negara kepulauan yang beriklim tropis yang memiliki dua musim yaitu musim kemarau dan musim penghujan. Musim penghujan terutama daerah yang memiliki saluran perairan yang kurang memadai dapat dipastikan akan terjadinya banjir dengan curah hujan yang tinggi. Banjir merupakan bencana bagi setiap daerah dimana air hujan yang turun tidak teralir maupun terserap ke tanah karena saluran perairannya yang kurang memadai. Data Badan Nasional Penanggulangan Bencana (BNPB) tahun 2014 sampai 2023 menunjukkan bencana yang sering terjadi adalah banjir [1]. Total kejadian

banjir yang terjadi tahun 2014 sampai 2023 berjumlah 8.110 kejadian mengakibatkan adanya korban jiwa, rusaknya rumah warga, dan fasilitas umum milik pemerintah.

Bencana banjir sering terjadi masih menjadi salah satu fokus perhatian yang mengakibatkan banyak korban jiwa, dan menimbulkan banyak kerugian, baik kerugian materil maupun psikologis. Tidak ada pencegahan secara efektif terhadap banjir untuk meminimalisir korban jiwa, dan minimnya sistem untuk memberi peringatan sedini mungkin akan datangnya banjir sehingga kerugian dapat dikurangi.

Pengembangan teknologi dan sensor terhadap banjir terus dikembangkan. Penelitian sebelumnya terdapat penggunaan alat deteksi banjir menggunakan Radar Doppler [2]. Alat ini memerlukan rancangan perangkat keras rumit dan biaya cukup besar. Sistem deteksi banjir menggunakan sensor ultrasonik berbasis mikrokontroler memiliki respon lambat dan menggunakan media SMS *gateway*. Selanjutnya deteksi banjir menggunakan sensor ultrasonik dan nodemcu dengan variasi jarak kedalaman 3-13 cm [3], atau memanfaatkan *IoT (Internet of Things)* dan sensor ultrasonik [4], serta menggunakan sensor ultrasonik dan ESP32 [5]. *IoT* yang menggunakan NodeMCU8266 dan *water level sensor* [6], atau menggunakan NodeMCU dan protokol MQTT berbasis *IoT* [7].

Pada penelitian ini digunakan *IoT*, dengan memanfaatkan teknologi internet sehingga obyek dapat diakses secara online. Pada dasarnya *IoT* merupakan teknologi kendali atau *monitoring* jarak jauh yang memanfaatkan jaringan internet sebagai penghubungnya, dan menggunakan *gadget* atau android sebagai media monitoringnya sehingga mempermudah *user* dalam pengoperasiannya [8]. Penelitian ini menggunakan *Water level sensor* dan ESP32, hasil ditampilkan pada layar *smartphone*. *Water level sensor* untuk mendeteksi ketinggian air saat ini dari persiapan pengujian yang disimulasikan, dan ESP32 merupakan papan mikrokontroler dengan modul wifi bawaan yang dapat mengirim pembacaan sensor melalui internet ke dalam basis data [9]. *Water level sensor* pada aplikasinya dimanfaatkan untuk *monitor water level* pada *smart water tanks* [10] atau *Water Level Prediction Model* melalui *Machine Learning* [11]. Selain itu, adanya pompa sebagai penanggulangan dini jika curah hujan yang tinggi dan dapat di akses oleh perangkat *smartphone* melalui aplikasi *BLYNK* [12]. Aplikasi *blynk* juga digunakan pada *smart bins* berbasis *IoT* [13].

METODE PENELITIAN

Pendekatan Teori

Prinsip kerja perangkat sistem ini terdiri dari input berupa *access point* dan *smartphone*, prosesnya menggunakan Mikrokontroler ESP32 dan outputnya berupa tampilan LCD dan rele. *Access point* berfungsi sebagai media agar terhubung ke jaringan internet sehingga dapat di monitoring dari jarak jauh, sensor level air digunakan untuk mengetahui ketinggian level air sehingga dapat mengetahui level air melalui LCD display dan Smartphone di aplikasi *blynk* [12]. Pompa berfungsi untuk menyedot air yang tergenang banjir dan mengalirkan ke daerah yang tidak tergenang banjir. Komponen atau rangkaian pendukungnya meliputi unit power supply, papan PCB, Arduino IDE, diode, *converter*, kapasitor, resistor, induktor, dan transistor.

Mikrokontroler ESP32 ditunjukkan pada gambar 1, dan dibuat oleh Espressif Systems. Kelebihannya yaitu biaya yang rendah, berdaya rendah pada seri chip (SoC) dengan Wi-Fi & kemampuan *Bluetooth* dua mode. Keluarga ESP32 termasuk chip ESP32-D0WDQ6 (dan ESP32-D0WD), ESP32-D2WD, ESP32-S0WD, dan sistem dalam paket (SiP) ESP32-PICO-D4 [14].

Fitur dan spesifikasi dari ESP32 :

- 18 ADC (*Analog Digital Converter*, berfungsi untuk mengubah sinyal analog ke digital)
- 2 DAC (*Digital Analog Converter*, kebalikan dari ADC)
- 16 PWM (*Pulse Width Modulation*)

- 10 Sensor sentuh
- 2 jalur antarmuka UART
- pin antarmuka I2C, I2S, dan SPI



GAMBAR 1. Mikrokontroler ESP32.

Water level merupakan seperangkat alat yang digunakan untuk mengukur ketinggian air di tempat yang berbeda sehingga mengetahui perbandingan. Penelitian ini menggunakan sensor level air jenis YL 83. Kelebihannya yaitu *water level* ini memiliki akurasi yang tinggi dalam pengukuran jarak jauh. Untuk hindari kesalahan pengukuran dalam pemakaian *water level*, suhu terhadap air haruslah sama. Sensor level air ditunjukkan pada gambar 2.

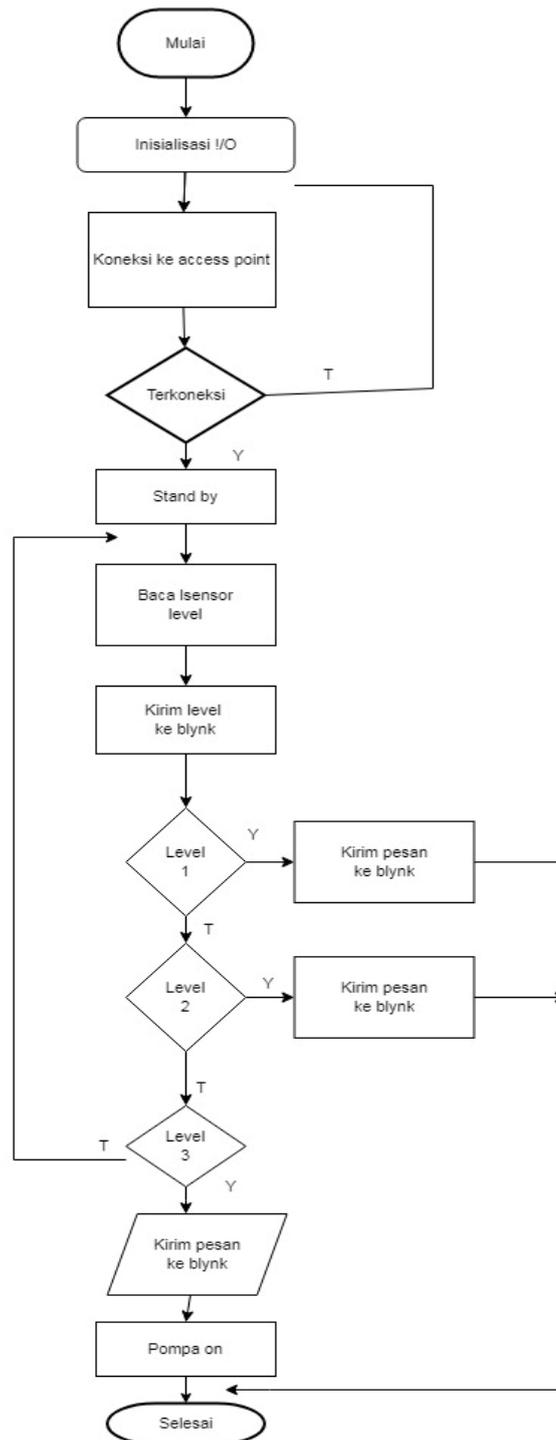


GAMBAR 2. Sensor level air.

ArduinoIDE (*Integrated Development Environment*) merupakan software yang digunakan untuk memprogram di arduino, dan sebagai media untuk memprogram *board* Arduino. *Blynk* adalah aplikasi platform *IoT* yang dapat diinstal di OS seluler yang memungkinkan pengguna berinteraksi dengan mikrokontroler seperti Arduino, Raspberry Pi, NodeMCU, dan modul lainnya melalui Internet. Artinya, *blynk* dapat mengontrol dan memantau proyek *IoT* bahkan dari jauh selama aplikasi *blynk* dan proyek *IoT* terhubung dengan internet [13], [15].

Rancang Bangun Alat

Perancangan adalah tahap terpenting dari seluruh proses pembuatan alat. Tahap pertama yaitu membuat diagram blok rangkaian, selanjutnya memilih komponen dengan karakteristik yang sesuai dengan kebutuhan. *Data book* dan petunjuk lain digunakan pemilihan komponen merupakan pilihan yang tepat bagi alat yang akan dibuat. Tahap awal dimulai dari pembuatan diagram blok rangkaian, pemilihan komponen, *flow chart*, sampai dengan *finishing* pemasangan komponen yang akan digunakan. *Flow chart* merupakan langkah awal dari program yang akan dibuat. Karena, *Flow chart* adalah hal terpenting dalam menggunakan sebuah mikrokontroler pada suatu alat yang dirancang [16]. *Flow chart* rancang bangun ini ditunjukkan pada gambar 3.



GAMBAR 3. Flow chart rancang bangun

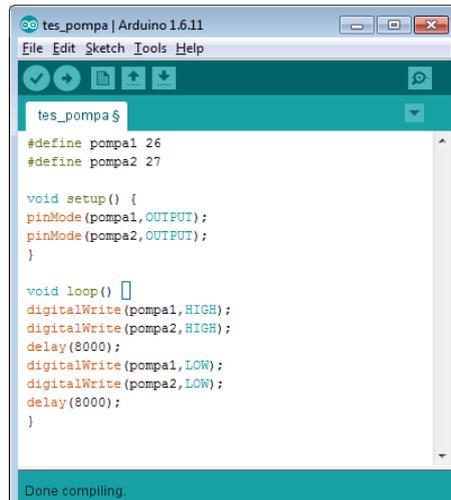
Perancangan Software

Sebelum bisa dijalankan, arduino memerlukan program yang menentukan arah dari arduino uno bekerja. Bahasa pemrograman yang digunakan pada arduino uno adalah bukan bahasa assembler yang relatif sulit tetapi bahasa yang sederhana dengan bantuan pustaka pustaka (*libraries*) arduino yang bisa membantu membuat program. Hal ini bertujuan untuk mempermudah *programmer* dalam mengembangkan programnya. Pada perancangan pemrograman ini salah satu komponen terpenting adalah mikrokontroler yang terdapat pada arduino uno merupakan komponen utama yang berperan sebagai otak dibalik pengaturan sistem rangkaian pada alat ini. Pada bagian ini penginputan *command*

ke dalam mikrokontroler menggunakan *software* IDE arduino. Adapun jenis bahasa yang digunakan adalah bahasa C, sedangkan untuk aplikasi android berupa *blynk*[17].

Kode Program Untuk Menguji Nyala Pompa

Awal mula dua buah pompa dihubungkan dengan pin ESP32 pada pin 26 dan 27, selanjutnya pompa dijadikan sebagai output. Kemudian untuk menyalakan pompa digunakan perintah *HIGH* sedangkan untuk mematikan pompa digunakan perintah *LOW*, pada program berikut pompa dihidupkan selama 8 detik selanjutnya mati selama 8 detik. Berikut kode program pada arduino IDE untuk uji nyala pompa ditunjukkan gambar 4.



```
tes_pompa | Arduino 1.6.11
File Edit Sketch Tools Help

tes_pompa$

#define pompa1 26
#define pompa2 27

void setup() {
  pinMode(pompa1, OUTPUT);
  pinMode(pompa2, OUTPUT);
}

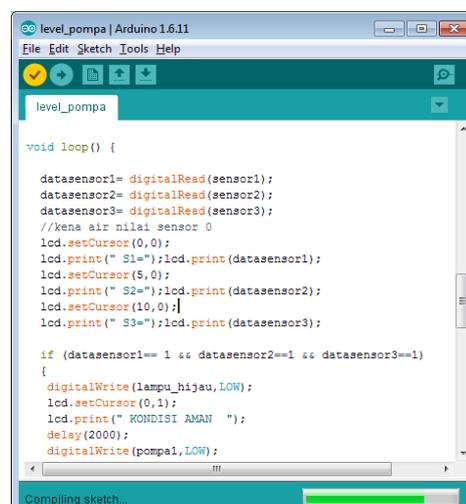
void loop() {
  digitalWrite(pompa1, HIGH);
  digitalWrite(pompa2, HIGH);
  delay(8000);
  digitalWrite(pompa1, LOW);
  digitalWrite(pompa2, LOW);
  delay(8000);
}

Done compiling.
```

GAMBAR 4. Kode program uji pompa

Kode Program Untuk Membaca Level Sensor

Terdapat 3 buah sensor dengan masing-masing sensor mempunyai batasan/level. Sensor 1 terhubung pada pin 33, sensor 2 pada pin 32 dan sensor 3 pada pin 34. Untuk mempermudah dalam pengujian sensor ditambahkan tampilan pada display LCD ukuran 2*16. Pembacaan setiap sensor berupa perintah *digital read*. Nilai-nilai sensor selanjutnya disimpan pada masing-masing *variable* berupa data sensor 1,2 dan 3. Selain itu juga terdapat lampu indikator sebagai penanda level air. Apabila setiap sensor tidak mendeteksi air berupa logic 1, maka semua lampu indikator tidak menyala dan menampilkan pesan pada LCD berupa “KONDISI AMAN”. Berikut kode program pembacaan sensor level air ditunjukkan pada gambar 5.



```
level_pompa | Arduino 1.6.11
File Edit Sketch Tools Help

level_pompa

void loop() {

  datasensor1= digitalRead(sensor1);
  datasensor2= digitalRead(sensor2);
  datasensor3= digitalRead(sensor3);
  //karena air nilai sensor 0
  lcd.setCursor(0,0);
  lcd.print(" S1=");lcd.print(datasensor1);
  lcd.setCursor(5,0);
  lcd.print(" S2=");lcd.print(datasensor2);
  lcd.setCursor(10,0);}
  lcd.print(" S3=");lcd.print(datasensor3);

  if (datasensor1== 1 && datasensor2==1 && datasensor3==1)
  {
    digitalWrite(lampu_hijau,LOW);
    lcd.setCursor(0,1);
    lcd.print(" KONDISI AMAN ");
    delay(2000);
    digitalWrite(pompa1,LOW);
  }

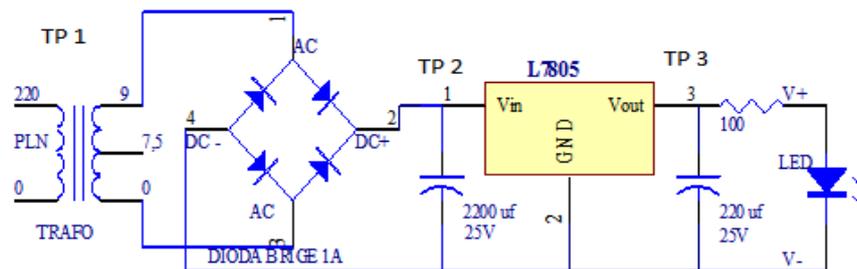
Compiling sketch...
```

GAMBAR 5. Kode program uji sensor

Prosedur Pengujian

Setelah rangkaian alat sistem pendeteksi dan penanggulangan banjir menggunakan ESP32 berbasis IoT berhasil dilakukan, selanjutnya dilakukan pengujian untuk mengamati output perangkat. Pengujian meliputi pengujian catu daya, sensor level air dan lampu indikator, IoT dan blynk. Pengujian yang dimaksudkan salah satunya pengujian tegangan AC dimana untuk memastikan bahwa sensor tersebut bekerja dengan baik dalam membaca tegangan AC yang diukur. Tegangan input yang masuk ke mikrokontroler ESP32 berkisar antara 5- 12V DC sehingga tegangan dari PLN 220 V dapat dikondisikan dan diubah menjadi Volt DC. Rangkaian catu daya ditunjukkan pada gambar 6.

Pengujian fungsional merupakan pengujian yang dilakukan untuk mengetahui apakah alat yang telah di buat dapat bekerja sesuai dengan yang diinginkan. Pengujian dilakukan pada beberapa bagian dari prototype yaitu pada rangkaian catu daya, pembacaan sensor level air dan pompa pada prototype. Berikut hasil data dari masing-masing pengujian :



GAMBAR 6. Rangkaian catu daya

– Blok rangkaian A.

Blok rangkaian A merupakan transformator yang berfungsi menaikkan atau menurunkan nilai tegangan. Transformator yang digunakan yaitu jenis transformator step down, maka tegangan input dari PLN sebesar 220 V AC diturunkan menjadi 9 V AC, nilai arus sebesar 500 mA, maka V_s yang dihasilkan sebagai berikut :

$$V_s = \sqrt{2} \times V_{rms}$$

$$V_s = \sqrt{2} \times 9 \text{ Volt} = 12.72 \text{ V}$$

– Blok rangkaian B

Blok rangkaian B merupakan bagian penyearah (*rectifier*) terdiri dari 1 buah *diode bridge* berfungsi mengubah arus bolak-balik (AC) dari PLN menjadi arus searah (DC). Besarnya tegangan keluaran dari *rectifier* dapat diketahui dengan menggunakan persamaan :

$$V_{dc} = \frac{2 \times V_s}{\pi}$$

$$V_{dc} = \frac{2 \times 12.72 \text{ Volt}}{3.14} = 8.10 \text{ Volt}$$

– Blok rangkaian C

Blok rangkaian C merupakan bagian penyaring (*filter*) yang terdiri dari 1 kapasitor elco (*polar*). berfungsi meminimalkan efek-efek arus AC pada arus DC konstan. Rangkaian tersebut diketahui output *transformator* (V_{trafo})= 9 V AC, arus *transformator* (I_{trafo}) = 500 MaH, Tegangan minimum (V_L) dioda sebagai penyearah 1 V, Frekuensi penyearah penuh (F_{penuh}) = 120 Hz.

– Blok rangkaian D

Blok rangkaian D merupakan bagian *voltage regulator* yang berfungsi penstabil tegangan keluaran dari *rectifier* yang konstan walaupun terdapat fluktuasi baik arus beban maupun

tegangan input sumber. Pada blok ini terdapat IC L7805 berfungsi mengubah tegangan sebelumnya menjadi 5 V. Terdapat juga kapasitor 220 μ F, 25V berfungsi menjaga tegangan keluaran 5V dari IC L7805 agar tetap stabil.

HASIL DAN PEMBAHASAN

Sesuai dengan lokasi titik pengujian rangkaian catu daya (*power supply*), maka hasil pengujian atau pengukuran pada masing-masing titik referensi dan titik uji dirangkum dalam Tabel 1.

TABEL 1. Pengujian rangkaian catu daya.

Lokasi pengujian	Hasil pengujian (V)	
Rangkaian Catu Daya	TP 1	227
	TP 2	8.97
	TP 3	4.94

Hasil pengukuran catu daya pada masing-masing TP menunjukkan bahwa TP1 merupakan tegangan sumber dari PLN yaitu sebesar 227 V AC, TP2 yang merupakan tegangan input IC regulator 8.97 V DC, sedangkan tegangan output yaitu pada TP3 sebesar 4.94 V DC. Pada TP3 nantinya digunakan untuk mensuplai komponen elektronika pada prototype seperti ESP32, *display* LCD maupun sensor level air. Selisih pengukuran dengan perhitungan pada hasil pengukuran Input IC regulator dan perhitungan Blok rangkaian B yaitu 8.97 V dan 8.10 V, selisih 0.87 V DC. Persentase kesalahan sebesar 1.1 % dari hasil pengukuran dapat ditoleransi.

Pengujian Sensor Air dan Lampu Indikator

Sensor level air menggunakan 3 buah yang dijadikan sebagai penentu level air. Saat sensor mengenai air akan menunjukkan logika 0, sedangkan pada saat tidak mengenai air akan menunjukkan logika 1. Indikator lampu sebagai penanda informasi level air. Lampu hijau menyala artinya kondisi level 1, lampu kuning menyala artinya level 2, dan lampu merah menyala artinya level ke 3 (bahaya). Data ini kemudian diolah oleh mikrokontroler ESP32 untuk dijadikan sebagai kondisi level air dan selanjutnya dapat mengendalikan pompa secara otomatis[4], [18]. Hasil pengujian sensor level air ditunjukkan pada gambar 7, 8, dan 9 serta tabel 2.

TABEL 2. Pengujian sensor level air dan lampu indikator.

Sensor level air	Warna lampu Indikator	Keterangan
Level 1	Hijau	Aman
Level 2	Kuning	Waspada
Level 3	Merah	Bahaya



GAMBAR 7. Hasil pengujian sensor air pada level 1.

Gambar 7 menunjukkan ketika air menyentuh level 1, maka akan mendapat notifikasi di LCD berupa tulisan aman.



GAMBAR 8. Hasil pengujian sensor air pada level 2.

Gambar 8 menunjukkan bahwa pengujian sensor di level 2. Ketika air menyentuh sensor level 2, maka akan mendapat notifikasi di LCD peringatan waspada.



GAMBAR 9. Hasil pengujian sensor air pada level 3.

Gambar 9 menunjukkan bahwa pengujian sensor level 3. Ketika air menyentuh di level 3, maka akan mendapat notifikasi di LCD peringatan bahaya.

Pengujian IoT dengan *Blynk*

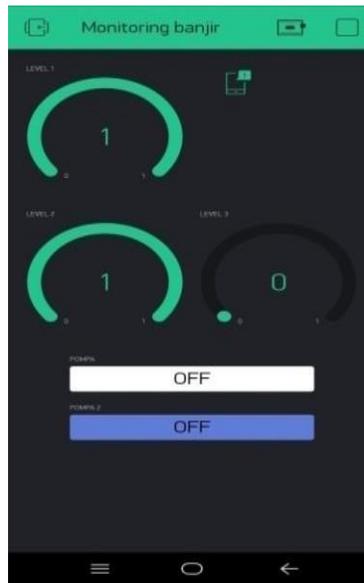
Pada pengujian IoT melalui aplikasi *blynk*, awal mula dilakukan pendaftaran akun pada *blynk*. Selanjutnya membuat rancangan desain aplikasi dengan langkah-langkah berupa pemilihan jalur koneksi melalui *Wifi*, jenis *board* yang digunakan berupa ESP32, dan selanjutnya pemilihan *widget* berupa 3 buah *Gauge*, 2 buah *styled button*, dan sebuah *notification*.

Tampilan menu aplikasi pada *smartphone*

Pada saat mengakses aplikasi *blynk*, *prototype* terlebih dahulu dihubungkan ke jaringan internet melalui *wifi* dengan nama dan *password hotspot* yang sudah ditentukan sehingga secara otomatis akan terhubung. Hasil pembacaan sensor selalu *update* dan proses pengiriman data sangat stabil. Hasil pengujian tampilan level setiap sensor pada aplikasi *blynk* ditunjukkan pada gambar 10, 11, dan 12.



GAMBAR 10. Tampilan notifikasi pengujian level 1 di *blynk*.



GAMBAR 11. Tampilan notifikasi pengujian level 2 di *blynk*.



GAMBAR 12. Tampilan notifikasi pengujian level 3 di *blynk*.

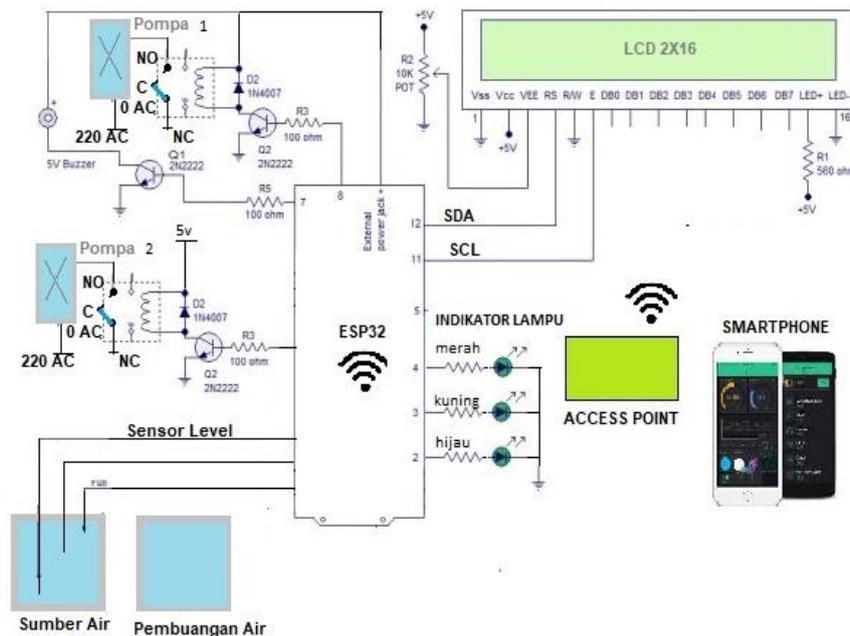
Diagram Rangkaian

Diagram Rangkaian adalah sistem yang diwakili oleh blok yang dihubungkan oleh garis yang menunjukkan hubungan blok. Tujuan pembuatan diagram blok adalah merancang sistem baru atau untuk menggambarkan dan meningkatkan suatu sistem yang sudah ada sebelumnya. Diagram blok rangkaian ditunjukkan pada gambar 13.

Prinsip Kerja Diagram Rangkaian

Berdasarkan desain blok diagram rangkaian secara keseluruhan *access point* berupa *hotspot* untuk koneksi internet yang dijadikan sebagai media penghubung antara *prototype* dengan *smartphone*. Jika sensor level mengenai sumber air lalu akan mengirimkan input ke ESP32 sebagai media lalu diproses dan mengeluarkan output berupa sinyal ke indikator lampu dan LCD *display*, jika masih level 1 lampu akan hidup berwarna hijau dan akan terlihat notifikasi di *smartphone* dan LCD *display*, jika sumber air terus naik dan menyentuh level 2 lampu indikator kuning akan menyala dan mengirimkan notifikasi ke *smartphone* melalui aplikasi *blynk* dan LCD *display*, dan apabila air terus naik menyentuh ke level 3 *relay* akan menyala dan pompa akan hidup

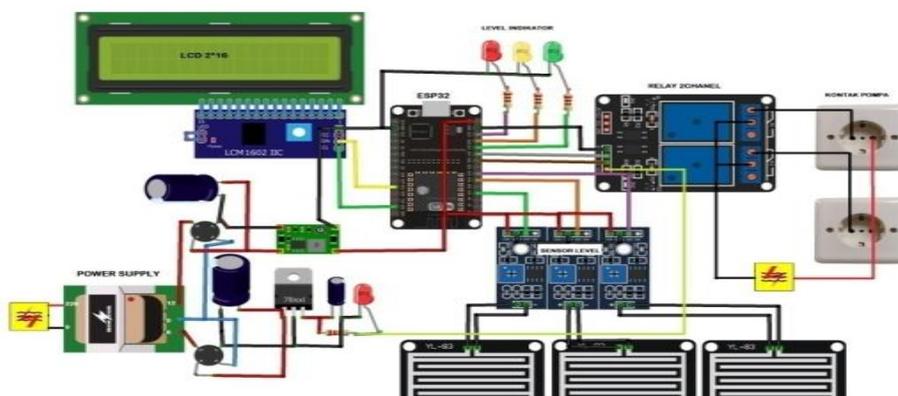
otomatis lalu mengirimkan sinyal berupa lampu indikator berwarna merah dan mendapatkan notifikasi melalui LCD *display* dan *smartphone* [16].



GAMBAR 13. Diagram blok rangkaian.

Rangkaian Keseluruhan

Rangkaian lengkap dari rangkaian rancang bangun sistem pendeteksi dan penganggulan banjir menggunakan ESP32 berbasis IoT ditunjukkan pada gambar 14.



GAMBAR 14. Skema rangkaian keseluruhan.

KESIMPULAN

Setelah melakukan pengujian terhadap rancang bangun deteksi banjir menggunakan ESP32 berbasis IoT, alat tersebut berhasil beroperasi dengan baik. Pembuatan rancang bangun terhubung ke jaringan internet diperlukan aplikasi *blynk* dan pemrograman arduino ide sebagai kode program. Hasil pengukuran di catu daya melalui multimeter didapatkan nilai 8.97 V dan hasil perhitungan didapatkan nilai 8.10 V. Hal ini terdapat selisih 0.87 V disebabkan kapasitor dapat menaikkan tegangan. Pompa dapat dikendalikan secara manual melalui *smarthone* dan apabila air menyentuh pada level 3, maka lampu indikator berwarna merah dan pompa akan hidup otomatis.

DAFTAR PUSTAKA

- [1] Badan Nasional Penanggulangan Bencana (BNPB), “Data Informasi Bencana Indonesia dan Badan Nasional Penanggulangan Bencana (BNPB),” 2023.
- [2] C. Umari, E. Anggraini, and R. Z. Muttaqin, “Rancang bangun sistem peringatan dini banjir berbasis sensor ultrasonik dan mikrokontroler sebagai upaya penanggulangan banjir,” *Jurnal Meteorologi Klimatologi dan Geofisika*, vol. 4, no. 2, pp. 35–42, Jul. 2017.
- [3] M. S. Novela, R. S. Hardinata, and R. P. N. D. Cahyo, “Perancangan Sistem Deteksi Banjir Menggunakan Sensor Ultrasonik Dan Nodemcu,” *Seminar Nasional Sosial Humaniora dan Teknologi (Senashtek)*, pp. 32–38, 2022.
- [4] N. Pratama, D. Ucu, and N. D. Nathasia, “Perancangan Sistem Monitoring Ketinggian Air Sebagai Pendeteksi Banjir Berbasis IoT Menggunakan Sensor Ultrasonik,” *Jurnal Media Informatika Budidarma*, vol. 4, no. 1, pp. 117–123, 2020.
- [5] Rusdi. M, M. W. Lestari, Yuvina, and F. N. Hulu, “River Flood Early Warning System Based on Internet of Things in Binjay City,” *International Journal of Research In Vocational Studies (IJRVOCAS)*, vol. 2, no. 4, pp. 42–47, 2023.
- [6] S. P. Windiastik, E. N. Ardhana, and J. Triono, “Perancangan Sistem Pendeteksi Banjir Berbasis IoT (Internet of Thing),” *Seminar Nasional Sistem Informasi 2019 UNMER Malang*, 2019.
- [7] A. Sutarti and S. Rosadi, “Prototype Sistem Pendeteksi Banjir Menggunakan NODEMCU dan Protokol MQTT Berbasis Internet Of Things,” *Jurnal Sistem Informasi dan Informatika (Simika)*, vol. 5, no. 1, pp. 38–48, 2022.
- [8] Z. B. Abilovani, Y. Widhi, and F. A. Bakhtiar, “Implementasi Protokol MQTT untuk Sistem Monitoring Perangkat IoT,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 12, 2018.
- [9] M. H. Khuzaidin and A. S. Sadun, “Development of Centralized Real - Time Flood Detection and Warning System,” *Progress in Engineering Application and Technology*, vol. 4, no. 1, pp. 263–270, 2023.
- [10] F. Jan, N. Min-Allah, S. Saeed, S. Z. Iqbal, and R. Ahmed, “IoT-Based Solutions to Monitor Water Level, Leakage, and Motor Control for Smart Water Tanks,” *Water (Basel)*, vol. 14, no. 3, 2022.
- [11] M. Herath, T. Jayathilaka, Y. Hosino, and U. Rathnayake, “Deep machine learning-Based Water Level Prediction model for Colombo Flood Detention Area,” *Applied Sciences*, vol. 13, no. 4, 2023.
- [12] M. R. Fahlevi and H. Gunawan, “Perancangan Sistem Pendeteksi Banjir menggunakan Internet of Things,” *IT Journal*, vol. 08, no. 1, 2020.
- [13] T. Juwariyah, L. Krisnawati, and S. Sulasminingsih, “Sistem Monitoring Terpadu SMART BINS Berbasis IoT Menggunakan Aplikasi BLYNK,” *Jurnal Informatika & Rekayasa Elektronika*, vol. 3, no. 2, 2020.
- [14] R. A. Pradana, “Mikrokontroler ESP32, apa itu? (bagian 1),” <https://timur.ilearning.me/2019/04/19/mikrokontroler-esp32-apa-itu/>, Apr. 19, 2019.
- [15] A. Wahyudi, “Memulai IoT dengan BLYNK dan Nodemcu,” <https://www.tptumetro.com/2020/05/memulai-iot-dengan-blynk-dan-nodemcu.html>, 2020.
- [16] M. A. Artiyasa, Edwinanto, and Anggi, “Aplikasi Smart Home NODE MCU IoT untuk BLYNK,” *Jurnal Rekayasa Teknologi Nusa Putra*, vol. 7, no. 1, Sep. 2020.

- [17] E. Prasetyo, “Cara Sederhana Memahami Kode Program (Sketch) Pada Arduino IDE,” <https://www.arduinoindonesia.id/2017/02/cara-sederhana-memahami-kode-program.html>, 2017.
- [18] S. Meidianta, A. R. Hakin, and B. Harpad, “Sistem Pendeteksian Peringatan Dini terhadap Banjir Berbasis Mikrokontroller,” *Jurnal Sains Terapan Teknologi Informasi*, vol. 10, no. 2, pp. 30–35, 2018.