

## Pengembangan Sistem Pengukur Curah Hujan di Sungai Jakarta Berbasis IoT

### *The Development of IoT-Based Rainfall Measuring System in Jakarta River*

HENDRA RAHMAWAN<sup>1\*</sup>, DARY MUZHAR MUHAMMAD<sup>2</sup>, FARIANTO<sup>3</sup>

#### Abstrak

Daerah Khusus Ibukota Jakarta merupakan dataran rendah dengan intensitas hujan yang tinggi pada periode tertentu. Saat ini curah hujan diukur secara manual oleh petugas dengan mengunjungi secara langsung alat pengukur curah hujan Ombrometer yang ditempatkan di sejumlah titik yang berdekatan dengan beberapa sungai di Jakarta. Dengan prosedur seperti itu maka pengukuran curah hujan menjadi tidak efisien secara waktu dan biaya karena memerlukan petugas yang harus mengunjungi lokasi pengukuran secara langsung dan memerlukan beberapa proses tambahan untuk mendapatkan nilai pengukuran. Hal tersebut akan menyulitkan jika data curah hujan diperlukan secara *real time* seperti untuk keperluan peringatan dini banjir. Untuk mengatasi masalah tersebut maka diperlukan sistem pengukur curah hujan yang dapat mengukur curah hujan secara otomatis dan mengirimkan hasil pengukurannya secara *real time* melalui jaringan. Pada penelitian ini dikembangkan sistem pengukur curah hujan berbasis IoT dengan menggunakan metode *waterfall* yang terdiri atas tahap pendefinisian kebutuhan, perancangan sistem, implementasi sistem, dan pengujian sistem. Sistem tersebut berbasiskan sensor pengukur curah hujan *tipping bucket* yang nilainya dapat dibaca dan diproses langsung oleh perangkat IoT NodeMCU ESP8266. Hasil pengujian telah menunjukkan bahwa seluruh kebutuhan fungsional sistem yang telah didefinisikan dapat berfungsi dengan baik.

Kata Kunci: ESP8266, IoT, sistem pengukur curah hujan, *tipping bucket*

#### Abstract

*DKI Jakarta is a lowland with high rainfall intensity in certain periods. Currently rainfall is measured manually by officers by visiting directly the Ombrometer rainfall gauge which is placed at a number of points close to several rivers in Jakarta. With such a procedure, the measurement of rainfall becomes inefficient in terms of time and cost because it requires officers who have to visit the measurement location directly and require several additional processes to get the measurement value. This will be difficult if rainfall data is needed in real time, such as for flood early warning purposes. To overcome this problem, a rainfall measuring system is needed that can measure rainfall automatically and send the measurement results in real time via the internet. In this research, an IoT-based rainfall measuring system has been developed using the waterfall method which consists of the stages of defining requirements, designing system, implementing system, and testing system. The system is based on a tipping bucket rainfall sensor whose value can be read and processed directly by the IoT NodeMCU ESP8266 device. The test results have shown that all the functional requirements of the system that have been defined can function properly.*

*Keywords: ESP8266, IoT, rainfall measuring system, tipping bucket*

<sup>1</sup>Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor 16680

<sup>2</sup>Sekolah Vokasi IPB

<sup>3</sup>XCAMP PT XL AXIATA

\*Penulis Korespondensi: Surel: [hrahmawan@apps.ipb.ac.id](mailto:hrahmawan@apps.ipb.ac.id)

## PENDAHULUAN

Hujan merupakan salah satu jenis presipitasi, yaitu suatu kejadian ketika *droplet* air atau kristal es turun ke permukaan tanah (Battan 1987). Beberapa variabel dari hujan adalah jumlah, waktu kejadian, bentuk, karakter dan intensitas (World Meteorological Organization 2006). Curah hujan adalah salah satu unsur iklim yang banyak diamati di Indonesia karena variasinya yang cukup besar (Hermawan 2010). Dengan demikian, adalah penting untuk selalu mengamati data curah hujan di berbagai wilayah di Indonesia. Curah hujan diukur dalam satuan milimeter (mm). Makna curah hujan 1 mm adalah jumlah air yang tersimpan dalam sebuah area seluas 1 m<sup>2</sup> adalah sebanyak 1000 ml atau 1 l (Battan 1987).

DKI Jakarta merupakan daerah dataran rendah yang rawan terjadi hujan dengan curah hujan yang tinggi pada periode tertentu. Curah hujan dapat diukur menggunakan alat ukur curah hujan, sehingga curah hujan dapat selalu dipantau dan diinformasikan kepada pihak-pihak yang berkepentingan di DKI Jakarta. Alat pengukur curah hujan yang digunakan di DKI Jakarta saat ini adalah ombrometer yang merupakan alat pengukur curah hujan tipe observasi. Masalah dalam penggunaan ombrometer adalah pengukuran curah hujan harus melibatkan pengamat untuk mengambil sampel air hujan yang telah tertampung untuk selanjutnya digunakan dalam perhitungan nilai curah hujan. Selain itu, akurasi pengukuran curah hujan menggunakan ombrometer sangat tergantung kepada akurasi operator dalam mengukur sampel air hujan yang diambil dari ombrometer (Maftukhah *et al.* 2016).

Selain ombrometer, alat ukur curah hujan jenis lain yang dapat digunakan adalah Hellman dan *tipping bucket*. Alat ukur Hellman bekerja dengan memanfaatkan gerakan naik-turun *float chamber* di dalamnya ketika air hujan masuk dan keluar. Gerakan tersebut akan menggerakkan pena secara otomatis sehingga menghasilkan grafik curah hujan. Meskipun pengukuran curah hujan terjadi secara otomatis pada alat ukur Hellman, tetapi pemrosesan dan penyimpanan data masih terjadi secara manual. Alat ukur hujan lainnya adalah *tipping bucket*. *Tipping bucket* mengukur curah hujan dengan memanfaatkan pergerakan dua bejana di dalamnya yang secara bergantian menampung air hujan yang masuk melalui corong (Maftukhah *et al.* 2016). Pergerakan bejana tersebut akan memicu *reed switch* yang berada di antara kedua bejana (Raghava dan Wani 2014), dan selanjutnya dapat dihubungkan ke perangkat IoT seperti ESP8266 untuk pemrosesan lebih lanjut.

Sebagai bagian dari mitigasi bencana banjir di Daerah Khusus Ibukota (DKI) Jakarta, data curah hujan yang bersifat *real time* dan mudah diperoleh memiliki peran penting karena akan menentukan langkah-langkah penanganan banjir yang perlu dilakukan. Data curah hujan yang bersifat *real time* dan mudah diperoleh akan sulit disediakan jika alat pengukur yang digunakan bersifat manual seperti alat ombrometer yang saat ini digunakan di DKI Jakarta. Dengan demikian, diperlukan sebuah sistem, yang terdiri atas perangkat keras dan perangkat lunak, yang mampu mengukur curah hujan dan segera mengirimkan data curah hujan tersebut secara *real time* ke pengguna melalui internet agar mudah diakses. Data curah hujan selanjutnya dapat diakses menggunakan aplikasi *mobile* sehingga akses data dapat dilakukan di berbagai tempat dan waktu.

Kemudahan untuk memperoleh data *real time* curah hujan yang diperoleh melalui pengukuran di lapangan dapat dicapai dengan menggunakan teknologi *internet of things* (IoT). IoT dapat dipandang sebagai dunia yang berisi milyaran objek yang di dalamnya memiliki kecerdasan tertanam, perangkat untuk berkomunikasi, kemampuan untuk mengindra lingkungan (*sensing*), dan kemampuan untuk melakukan aksi (*actuation*). Seluruh objek tersebut terhubung melalui jaringan *internet protocol* (IP) (Cirani *et al.* 2018). IoT adalah salah satu teknologi canggih terkini yang memiliki beberapa kelebihan yaitu memungkinkan adanya peningkatan utilisasi, memerlukan interaksi pengguna yang minimum, memungkinkan pemeliharaan yang bersifat proaktif, memungkinkan adanya pengurangan biaya, dan membuka peluang pengembangan layanan-layanan baru yang ditingkatkan (Shafique *et al.* 2020).

Salah satu perangkat yang dapat digunakan untuk mengimplementasikan teknologi IoT adalah modul NodeMCU ESP8266. NodeMCU adalah *firmware* berbasis Lua *open source* untuk modul ESP8266 WiFi *system on chip* (SoC) yang diproduksi oleh *Espressif System*. Mulanya, *firmware* ini dikembangkan sebagai pendamping dari modul pengembangan ESP8266, tetapi karena komunitas pengguna NodeMCU sangat banyak, selanjutnya proyek ini didukung oleh komunitas dan dapat digunakan pada modul ESP lainnya (NodeMCU Team 2018). Salah satu kelebihan modul ESP8266 adalah memiliki pengaturan yang memungkinkan konsumsi daya perangkat yang rendah sehingga durasi operasi perangkat dapat bertahan lama. Hal tersebut mengakibatkan modul ESP8266 sangat sesuai untuk dioperasikan dengan baterai yang diisi ulang dengan menggunakan sel surya (Gowda *et al.* 2020).

Beberapa penelitian telah dilakukan terkait dengan pengukuran curah hujan secara otomatis. Alam *et al.* (2021) telah membangun alat pengukur curah hujan *tipping bucket* dan menghubungkannya dengan mikrokontroler Arduino Uno dan perangkat IoT NodeMCU ESP8266 untuk mengirimkan data curah hujan ke *cloud server* ThinSpeak dan Google Drive. Penelitian tersebut telah dapat mengukur curah hujan dan mengirimkannya ke ThingSpeak dan Google Drive namun masih menggunakan mikrokontroler tambahan selain NodeMCU ESP8266, yaitu Arduino Uno, yang tentunya akan memerlukan tambahan daya untuk mengoperasikannya. Selain itu, pada penelitian tersebut juga belum menyediakan aplikasi bagi *end user* untuk melihat hasil pengukuran curah hujan. Begitu juga dengan Ramadhan *et al.* (2020) yang telah mengembangkan sistem pemantauan banjir dan pencegahan dini berbasis IoT. Sistem tersebut menggunakan sensor curah hujan *tipping bucket*, sensor tetes hujan, mikrokontroler Arduino Mega2560, dan mikrokontroler ESP8266-01. Sistem tersebut mengirimkan data curah hujan dan durasi hujan ke pengguna melalui aplikasi *mobile* berbasis Blynk. Penggunaan mikrokontroler Arduino Mega2560 pada penelitian tersebut tentunya akan meningkatkan konsumsi daya secara keseluruhan. Catu daya alat pada penelitian tersebut juga belum dirancang agar dapat dipasang di lapangan tanpa harus terhubung ke jaringan listrik PLN.

Penelitian lainnya yang merancang prototipe perangkat IoT untuk mengukur curah hujan sebagai *early warning indicator* untuk banjir lahar dingin telah dilakukan oleh Katriani dan Darmawan (2021). Seperti pada penelitian sebelumnya, penelitian tersebut juga menggunakan mikrokontroler Arduino Uno dan modul ESP8266 yang berkonsekuensi terhadap konsumsi daya yang lebih besar. Yuwono *et al.* (2018) telah mengembangkan perangkat pemantau cuaca berbasis sensor *tipping bucket* dan komputer mini Raspberry Pi. Pada penelitian tersebut, penggunaan Raspberry Pi memiliki potensi konsumsi daya yang lebih besar dibandingkan penggunaan mikrokontroler seperti Arduino Uno atau ESP8266. Hal tersebut salah satunya disebabkan karena Raspberry Pi menggunakan prosesor dengan kecepatan *clock* yang jauh lebih tinggi dibandingkan Arduino Uno atau ESP8266.

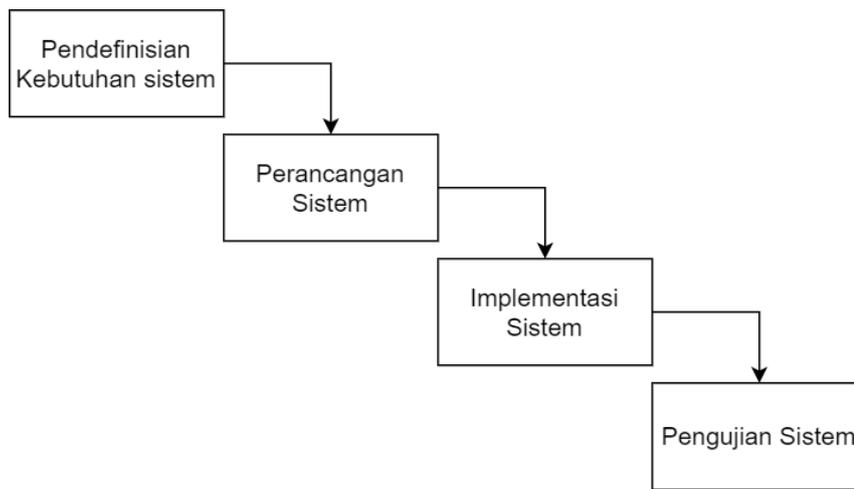
Berdasarkan deskripsi masalah dan penelitian-penelitian sebelumnya yang telah diuraikan maka penelitian ini memiliki tujuan umum untuk membangun sistem pengukur curah hujan otomatis berbasis IoT. Sistem tersebut dapat mengukur curah hujan secara otomatis menggunakan sensor *tipping bucket* dan mengirimkan data hasil pengukuran ke aplikasi *mobile* melalui jaringan internet. Kelebihan sistem yang dibuat pada penelitian ini dibandingkan kedua sistem pada penelitian terdahulu yang telah dijelaskan adalah pada jumlah mikrokontroler yang digunakan, yaitu hanya menggunakan satu unit ESP8266, dan alat sudah dirancang untuk dipasang di lapangan tanpa harus terhubungan ke jaringan listrik PLN. Manfaat dari sistem yang dihasilkan dari penelitian ini adalah mempermudah pengamat melakukan perhitungan curah hujan melalui alat pengukur curah hujan berbasis IoT dan mendapatkan data secara *real time*.

## METODE

Penelitian ini dibatasi pada ruang lingkup berikut:

- 1 Sistem yang dikembangkan bersifat prototipe yang belum digunakan pada situasi nyata.
- 2 Data yang ditampilkan dibatasi pada data curah hujan per hari.
- 3 Sistem diasumsikan akan digunakan oleh petugas pemantau curah hujan, dan bukan oleh masyarakat umum.

Metode yang digunakan untuk mengembangkan sistem ini diadopsi dari metode *Waterfall* (Royce 1987) yang disesuaikan dengan kebutuhan penelitian ini sehingga menjadi empat tahap yaitu pendefinisian kebutuhan sistem, perancangan sistem, implementasi sistem, dan pengujian sistem. Alur dari keempat tahap tersebut dideskripsikan pada Gambar 1 dengan penjelasan sebagai berikut.



Gambar 1 Metode pengembangan sistem pengukur curah hujan berbasis IoT.

### Perancangan Sistem

Berdasarkan kebutuhan-kebutuhan yang telah didefinisikan pada tahap sebelumnya, pada tahap ini dibuat rancangan modul perangkat keras dan perangkat lunak yang membangun sistem pengukur curah hujan berbasis IoT. Rancangan perangkat keras dibuat dengan menggunakan perangkat lunak Fritzing. Adapun algoritme dalam modul perangkat lunak dirancang dalam bentuk diagram alir (*flowchart*).

### Implementasi Sistem

Rancangan-rancangan yang sudah dibuat pada tahap perancangan sistem selanjutnya diimplementasikan pada tahap ini. Proses implementasi perangkat keras berupa perakitan rangkaian elektronik yang dibangun dari beberapa modul sesuai dengan rancangan yang sudah dibuat. Sementara itu, proses implementasi perangkat lunak berupa penulisan kode program dari algoritme yang sudah dirancang.

### Pengujian Sistem

Sistem pengukur curah hujan berbasis IoT yang telah diimplementasikan akan diuji untuk mengetahui kesesuaian implementasi dengan kebutuhan-kebutuhan yang telah didefinisikan. Dari tahap ini akan diketahui apakah sistem dapat berjalan dengan baik atau tidak. Jika ditemukan kesalahan pada sistem, maka langkah perbaikan segera dilakukan. Pada penelitian ini, sistem diuji dengan metode *black box*.

Pengujian yang dilakukan pada penelitian ini dibatasi pada kebutuhan fungsional dan nonfungsional yang didefinisikan. Pengujian terhadap validitas hasil pembacaan curah hujan belum dicakup dalam penelitian ini. Namun demikian, algoritme perhitungan curah hujan

mengikuti manual penggunaan dari sensor pengukur curah hujan, sehingga diharapkan hasil pembacaan tetap akurat sesuai standar sensor tersebut.

## HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan mengenai hasil-hasil dari seluruh tahap pengembangan sistem pengukur curah hujan berbasis IoT. Hasil dari tahap pengumpulan kebutuhan sistem, perancangan sistem, implementasi sistem, dan pengujian sistem secara rinci dijabarkan dalam penjelasan berikut.

### Pendefinisian Kebutuhan Sistem

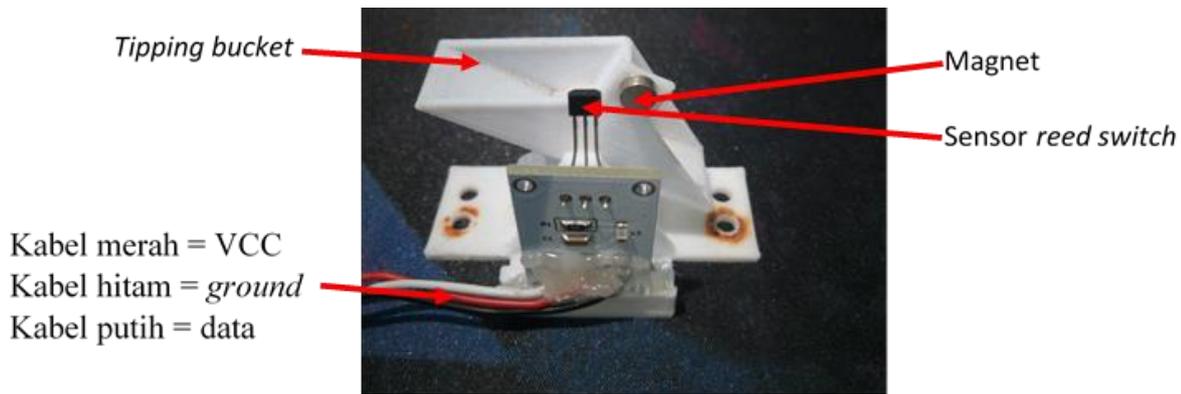
Berdasarkan komunikasi dengan pengguna, pada tahap ini dapat didefinisikan kebutuhan-kebutuhan fungsional dan nonfungsional dari sistem pengukur curah hujan berbasis IoT. Kebutuhan-kebutuhan tersebut dinyatakan pada Tabel 1 dan Tabel 2. Selain kebutuhan-kebutuhan fungsional, pada tahap ini juga diidentifikasi kebutuhan perangkat keras dan perangkat lunak yang diperlukan baik untuk pengembangan maupun operasional sistem. Kebutuhan-kebutuhan tersebut dinyatakan pada Tabel 3 dan 4. Berikut penjelasan ringkas dari beberapa perangkat keras yang digunakan dalam operasional sistem ini.

#### *1 Sensor Curah Hujan Tipe Tipping Bucket*

Pengukur curah hujan yang digunakan pada penelitian ini memiliki kapasitas 0,7 mm per tip. Kapasitas per tip pada pengukur curah hujan *tipping bucket* mungkin saja bersifat variatif tergantung produsen dari sensor tersebut. Secara internal, sensor ini bekerja dengan menggunakan *reed switch* yang dipicu oleh *tipping bucket* bermagnet. Sensor ini memiliki tiga kabel yang umumnya berwarna merah, hitam, dan putih. Kabel berwarna merah dihubungkan ke VCC, kabel berwarna hitam dihubungkan ke *ground*, dan kabel berwarna putih atau kuning dihubungkan ke kaki mikrokontroler untuk mengirimkan nilai (Manullang dan Tamba 2013). Gambar 2 merupakan tampak luar dari sensor curah hujan tipe *tipping bucket*, dan Gambar 3 menyajikan tampak dalam dari sensor.



Gambar 2 Tampak luar sensor curah hujan jenis *tipping bucket*.



Gambar 3 Tipping bucket dan reed switch.

Tabel 1 Kebutuhan fungsional alat pengukur curah hujan berbasis IoT

No	Kebutuhan fungsional
1	Sistem dapat membaca <i>input</i> dari sensor curah hujan <i>tipping bucket</i>
2	Sistem dapat mengirimkan data curah hujan per jam ke <i>database</i>
3	Sistem dapat mengirimkan data total curah hujan ke <i>database</i>
4	Sistem dapat mengirimkan data rata-rata curah hujan ke <i>database</i>

Tabel 2 Kebutuhan nonfungsional alat pengukur curah hujan berbasis IoT

No	Kebutuhan nonfungsional
1	Perangkat keras dapat menyala selama 24 jam
2	<i>Casing</i> dapat melindungi perangkat keras dari air dan terik matahari

Tabel 3 Kebutuhan perangkat keras

No	Nama Komponen	Jumlah	Keterangan
1	Sensor Curah Hujan tipe <i>tipping bucket</i>	1	Sensor pengukur curah hujan
2	NodeMCU ESP8266 Amica	1	Mikrokontroler untuk menjalankan program perhitungan curah hujan dan mengirimkan data ke internet
3	Modul RTC DS3231	1	Modul pencatat waktu <i>real time</i>
4	Kabel Micro USB 20cm	1	Untuk menghubungkan NodeMCU ke komputer dalam proses pemrograman
5	<i>Solar Charge Controller</i> 10A	1	Untuk mengendalikan pengisian ulang baterai aki VRLA
6	Panel Surya 10 WP	1	Untuk mendapatkan daya dari cahaya untuk proses pengisian ulang aki VRLA
7	Aki VRLA 2.3Ah 12Volt	1	Baterai sebagai catu daya alat
8	Box Panel IP65 25x20x15cm	1	<i>Casing</i> bagi alat/perangkat keras yang telah dirakit
9	<i>Notebook</i> Lenovo ideapad 310-14IKB	1	Untuk membuat rancangan sistem dan melakukan implementasi perangkat lunak
	- Intel Core i5 7200U		
	- SSD 256GB		
	- HDD 1000GB		
	- RAM 8GB DDR4 2133MHz		

Tabel 4 Kebutuhan perangkat lunak

No	Nama <i>Software</i>	Versi	Kegunaan
1	Arduino IDE	1.8.13	Untuk menulis kode program dan mengirimkan program ke NodeMCU
2	Fritzing	0.8.7b	Untuk membuat skema rangkaian
3	Kodular		Untuk membuat aplikasi <i>mobile</i> berbasis Android
4	Firestore		Untuk membuat <i>cloud database</i> yang mendukung real time database

## 2 NodeMCU ESP8266

Modul NodeMCU yang digunakan pada penelitian ini memiliki spesifikasi sebagai berikut: ukuran: 57mm x 30mm; tegangan input 3.3V – 5V; GPIO 14 Pin; *Flash memory* 4MB; *Clock speed* 40/26/24 MHz; WiFi IEEE 802.11 b/g/n; frekuensi 2.4 GHz; and Mikro USB.

### Perancangan Sistem

Tahap perancangan sistem pengukur curah hujan berbasis IoT dimulai dengan mendefinisikan arsitektur global dari sistem seperti dinyatakan pada Gambar 4. Sistem tersebut terdiri atas empat komponen utama yaitu alat pengukur curah hujan, perangkat koneksi internet melalui *wifi*, *cloud database*, dan aplikasi pada perangkat *mobile*. Secara umum sistem bekerja dengan membaca dan menghitung data curah hujan lalu mengirimkannya ke *cloud database* Firebase melalui jaringan *wifi* yang terhubung ke internet. Aplikasi *mobile* selanjutnya akan membaca data curah hujan yang telah terimpan di Firebase dan menampilkan informasi terkait curah hujan ke pengguna.



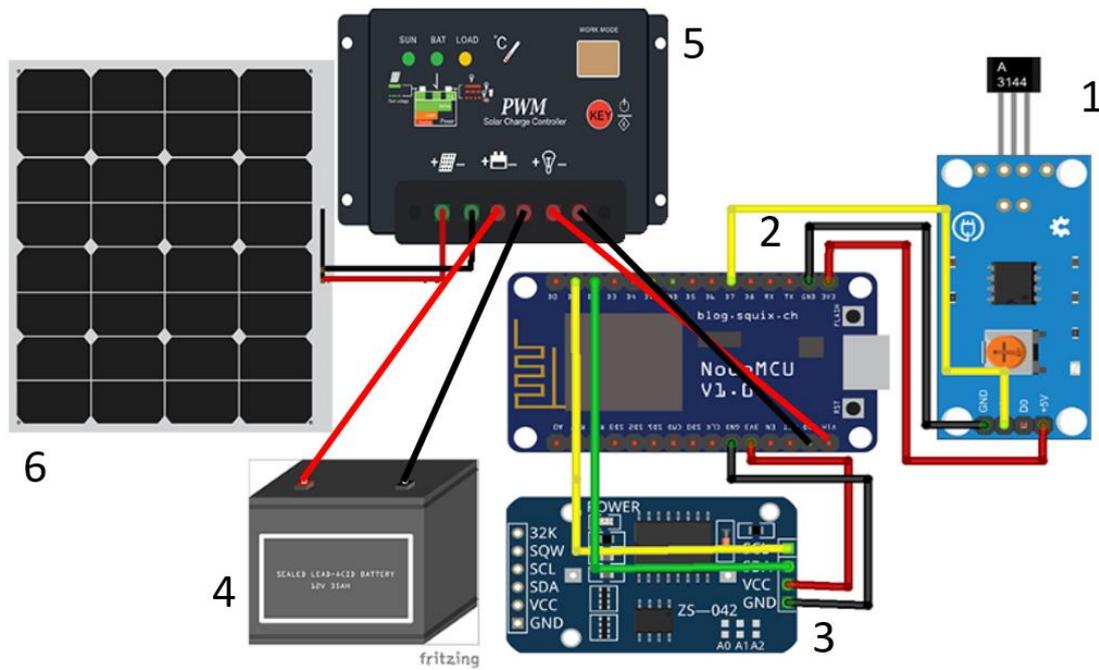
Gambar 4 Arsitektur umum sistem pengukur curah hujan berbasis IoT.

## 3 Perancangan Perangkat Keras

Rancangan perangkat keras dibuat dengan menggunakan perangkat lunak Fritzing. Rancangan yang dihasilkan berupa skema rangkaian yang dapat dilihat pada Gambar 5. Pada rangkaian tersebut panel surya dan baterai terhubung ke *solar charge controller* untuk mengatur daya yang keluar-masuk baterai. Sensor curah hujan terhubung ke kaki pin D7 pada NodeMCU untuk data, VCC ke kaki pin 3V3, dan GND ke kaki GND. Modul RTC DS3231 kaki SCL terhubung ke pin D1, SDL terhubung ke pin D2, VCC terhubung ke 3V3, GND terhubung ke GND. Modul RTC DS3231 berfungsi sebagai pengatur tanggal dan waktu untuk pengiriman data dari NodeMCU ke database. Berikut keterangan modul pada Gambar 5:

- 1 Sensor curah hujan (*reed switch*),
- 2 Nodemcu ESP8266,
- 3 RTC DS3231,
- 4 Baterai/Aki 12Volt,
- 5 *Solar Charge Controller*, dan
- 6 Panel surya.

Untuk melindungi komponen dari percikan air maupun panas matahari digunakan *casing* berupa *box panel* yang terbuat dari alumunium dengan ukuran 25 x 20 x 15 cm yang memiliki sertifikat IP65 anti air. *Casing* tersebut ditunjukkan pada Gambar 6.



Gambar 5 Skema rangkaian alat pengukur curah hujan berbasis IoT.



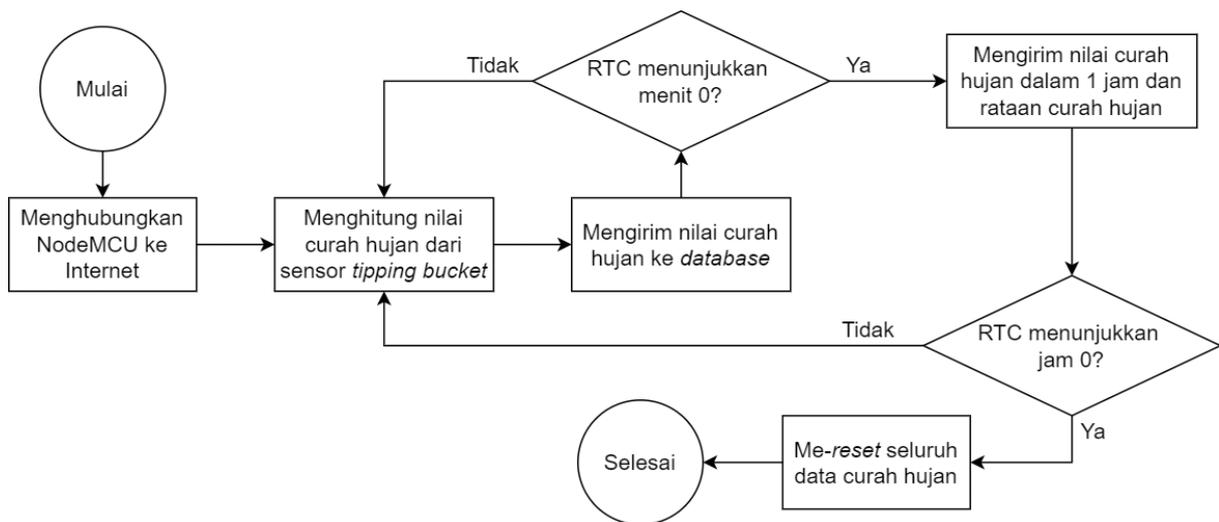
Gambar 6 Casing tampak depan, samping, dan belakang.

#### 4 Perancangan Perangkat Lunak

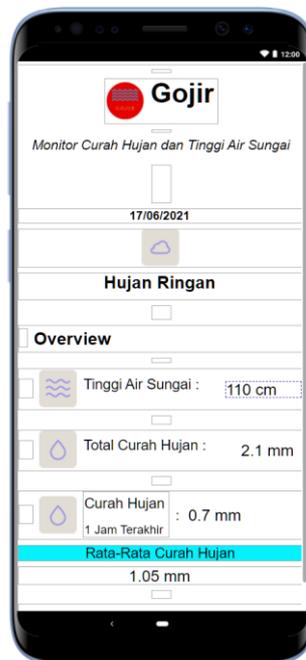
Perangkat lunak yang dirancang pada penelitian ini terdiri atas program penghitung curah hujan dan pengirim data curah hujan yang selanjutnya diimplementasikan pada mikrokontroler NodeMCU ESP8266. Algoritme penghitung curah hujan dan pengirim data curah hujan secara global dinyatakan pada Gambar 7. Alur kerja dari perangkat dimulai dengan menghubungkan NodeMCU ke internet melalui jaringan *wifi*. NodeMCU akan menghitung nilai curah hujan yang diperoleh dari sensor *tipping bucket*. Nilai curah hujan akan dikirim oleh NodeMCU ke *database* Firebase sebagai curah hujan pada hari ini dan akan terkalkulasi sampai menunjukkan pukul 00.00. Jika RTC menunjukkan jam 00.00 maka nilai curah hujan hari ini berubah menjadi nol. Jika RTC menunjukkan menit ke-0, maka nilai yang diterima dari sensor selama satu jam terakhir akan dikirim ke *database* sebagai curah hujan per jam.

Antarmuka aplikasi *mobile* berbasis Android dari sistem pengukur curah hujan dirancang dengan menggunakan aplikasi web Kodular. Aplikasi *mobile* tersebut dapat menampilkan total curah hujan per hari, curah hujan satu jam terakhir, dan cuaca berdasarkan perhitungan dari total curah hujan. Aplikasi tersebut diberi nama Gojir dan tidak hanya memantau curah hujan,

tetapi sudah terintegrasi dengan alat pengukur ketinggian air sungai yang sudah dikembangkan sebelumnya. Gambar 8 menunjukkan tampilan antarmuka aplikasi *mobile* Gojir.



Gambar 7 Flowchart untuk menghitung dan mengirim data curah hujan.



Gambar 8 Rancangan antarmuka aplikasi Gojir.

## Implementasi Sistem

Berdasarkan rancangan yang telah dibuat, selanjutnya dilakukan perakitan perangkat keras dan implementasi aplikasi *mobile* untuk membangun sistem pengukur curah hujan berbasis IoT.

### 1 Implementasi Perangkat Keras

Berdasarkan skema rangkaian yang telah dibuat, maka dilakukan perakitan alat pengukur curah hujan berbasis IoT. Alat yang telah dirakit kemudian ditempatkan di dalam *casing*. Alat yang telah dirakit dan ditempatkan di dalam *casing* ditunjukkan pada Gambar 9. Terdapat lubang pada bagian bawah *casing* sebagai jalur keluar-masuk kabel dari panel surya dan sensor curah hujan. Dua komponen tersebut diharuskan berada di luar *casing* agar dapat berfungsi dengan baik. Alat pengukur curah hujan telah terintegrasi dengan pengukur ketinggian air sungai sehingga alat dapat diletakan di sungai. Gambar 10 merupakan tampak luar dari

perangkat pengukur curah hujan. Urutan dari kiri ke kanan, yaitu: sensor curah hujan tipe *tipping bucket*, *casing* (berisi mikrokontroler Node MCU, baterai aki, dan *charger controller*), dan panel surya.

## 2 Implementasi Perangkat Lunak

Perangkat lunak diimplementasikan berdasarkan *flowchart* dan rancangan antarmuka yang sudah dibuat pada tahap sebelumnya. Gambar 11 merupakan tampilan antarmuka dari aplikasi *mobile* berbasis Android yang telah terpasang di *smartphone*. Sebagian kode program pengukur dan pengirim data curah hujan pada NodeMCU dinyatakan pada Gambar 12.



Gambar 9 Alat yang telah ditempatkan di dalam *casing*.



Gambar 10 Susunan alat-alat (tampak luar).



Gambar 11 Antarmuka aplikasi *mobile* Gojir yang sudah diimplementasikan.

```

void loop()
{
  DateTime now = rtc.now();
  if (bucketPositionA == true) {
    curah_hujan += bucketAmount;
    jumlah_tip++;
    delay(500);
    bucketPositionA = false;
  }
  curah_hujan_hari_ini = jumlah_tip * bucketAmount;
  temp_curah_hujan_per_menit = curah_hujan;

  if (curah_hujan_hari_ini >= 0.00 && curah_hujan_hari_ini <= 0.50)
    cuaca = "Berawan";
  if (curah_hujan_hari_ini > 0.50 && curah_hujan_hari_ini <= 20.00)
    cuaca = "Hujan Ringan";
  if (curah_hujan_hari_ini > 20.00 && curah_hujan_hari_ini <= 50.00)
    cuaca = "Hujan Sedang";
  if (curah_hujan_hari_ini > 50.00 && curah_hujan_hari_ini <= 150.00)
    cuaca = "Hujan Lebat";
  if (curah_hujan_hari_ini > 150.00)
    cuaca = "Hujan Ekstrim";

  if (now.second() == 0) {
    temp_curah_hujan_per_jam += curah_hujan_per_menit;
    if (now.minute() == 0){
      curah_hujan_per_jam = temp_curah_hujan_per_jam;
      temp_curah_hujan_per_hari += curah_hujan_per_jam;
      temp_curah_hujan_per_jam = 0.00;
      jumlah_data++;
      rata_rata_hujan = curah_hujan_hari_ini / jumlah_data;
    }
    if (now.minute() == 0 && now.hour() == 0){
      curah_hujan_per_hari = temp_curah_hujan_per_hari;
      temp_curah_hujan_per_hari = 0.00;
      curah_hujan_hari_ini = 0.00;
      jumlah_tip = 0;
      jumlah_data = 0;
      rata_rata_hujan = 0.00;
    }
    temp_curah_hujan_per_menit = 0.00;
    curah_hujan = 0.00;
    delay(1000);
  }
  if ((jumlah_tip != temp_jumlah_tip) || (now.second() == 0)){
    kirimData();
  }
  temp_jumlah_tip = jumlah_tip;
}

```

Gambar 12 Potongan kode program penghitung dan pengirim data curah hujan pada NodeMCU.

## Pengujian Sistem

Pengujian kebutuhan fungsional alat dan aplikasi *mobile* serta kebutuhan non fungsional alat dilakukan menggunakan metode *blackbox*. Mengingat sistem yang dikembangkan masih dalam tingkat prototipe dan keterbatasan hujan yang turun di Jakarta, maka pengujian tidak dilakukan di sungai, melainkan dilakukan di rumah peneliti dan hujan disimulasikan dengan air yang disemprotkan melalui selang.

Untuk keperluan pengujian dirancang skenario-skenario pengujian sesuai dengan kebutuhan-kebutuhan fungsional dan nonfungsional yang telah didefinisikan. Tabel 5, Tabel 6, dan Tabel 7 menyatakan skenario pengujian fungsional alat, skenario pengujian fungsional aplikasi *mobile*, dan kebutuhan nonfungsional alat. Pengujian dilakukan pada kurun waktu 5-8 April 2021.

Hasil pengujian kebutuhan fungsional alat dan aplikasi *mobile* serta kebutuhan non fungsional alat ditunjukkan pada Tabel 8, Tabel 9, dan Tabel 10. Dari hasil pengujian tersebut dapat dilihat alat pengukur curah hujan berhasil membaca input sensor, mengirimkan data ke *database*, dan aplikasi *mobile* berhasil membaca data yang tersimpan di *database*.

Pengujian perangkat yang dilakukan yaitu pengukuran nilai didapat oleh sensor curah hujan dan nodemcu dapat mengirimkan nilai yang terbaca oleh sensor ke *database*. Aplikasi *mobile*

dapat menampilkan data curah hujan per jam, rata-rata curah hujan, dan total curah hujan seperti yang ditunjukkan pada Gambar 13.

Tabel 5 Skenario pengujian kebutuhan fungsional alat pengukur curah hujan

No. kebutuhan fungsional	Skenario pengujian	Luaran yang diharapkan
1	Menampilkan nilai yang dibaca sensor curah hujan ke <i>console</i>	Nilai curah hujan muncul di <i>console</i>
2	Membaca data curah hujan per jam dari <i>database</i>	<i>Database</i> berhasil menyimpan data curah hujan per jam
3	Membaca data total curah hujan dari <i>database</i>	<i>Database</i> berhasil menyimpan data total curah hujan
4	Membaca data rata-rata curah hujan dari <i>database</i>	<i>Database</i> berhasil menyimpan data rata-rata curah hujan

Tabel 6 Skenario pengujian kebutuhan fungsional aplikasi *mobile*

No. kebutuhan fungsional	Skenario pengujian	Luaran yang diharapkan
1	Menampilkan data curah hujan per jam pada aplikasi <i>mobile</i>	Data curah hujan per jam berhasil diunduh dan ditampilkan pada aplikasi <i>mobile</i>
2	Menampilkan data total curah hujan pada aplikasi <i>mobile</i>	Data total curah hujan berhasil diunduh dan ditampilkan pada aplikasi <i>mobile</i>
3	Menampilkan data rata-rata curah hujan dari <i>database</i>	Data rata-rata curah hujan berhasil diunduh dan ditampilkan pada aplikasi <i>mobile</i>
4	Menampilkan data cuaca dari <i>database</i>	Data cuaca berhasil diunduh dan ditampilkan pada aplikasi <i>mobile</i>

Tabel 7 Skenario pengujian kebutuhan non fungsional alat pengukur curah hujan

No. kebutuhan nonfungsional	Skenario pengujian	Hasil yang diharapkan
1	Menyalakan alat selama 24 jam	Alat tetap menyala dalam 24 jam
2	Menyiram dan menjemur <i>casing</i> yang di dalamnya terdapat alat pengukur curah hujan	Alat di dalam <i>casing</i> tetap terjaga dengan baik

Tabel 8 Hasil pengujian kebutuhan fungsional alat pengukur curah hujan

No	Kebutuhan fungsional alat pengukur curah hujan	Hasil pengujian
1	NodeMCU dapat membaca <i>input</i> sensor dari sensor curah hujan	BERHASIL
2	NodeMCU dapat mengirimkan data curah hujan per jam ke <i>database</i>	BERHASIL
3	NodeMCU dapat mengirimkan data total curah hujan ke <i>database</i>	BERHASIL
4	NodeMCU dapat mengirimkan data rata-rata curah hujan ke <i>database</i>	BERHASIL

Tabel 9 Hasil pengujian kebutuhan fungsional aplikasi *mobile*

No	Kebutuhan fungsional aplikasi <i>mobile</i>	Hasil pengujian
1	Aplikasi <i>mobile</i> dapat membaca data curah hujan per jam dari <i>database</i>	BERHASIL
2	Aplikasi <i>mobile</i> dapat membaca data total curah hujan dari <i>database</i>	BERHASIL
3	Aplikasi <i>mobile</i> dapat membaca data rata-rata curah hujan dari <i>database</i>	BERHASIL
4	Aplikasi <i>mobile</i> dapat membaca data cuaca dari <i>database</i>	BERHASIL

Tabel 10 Hasil pengujian kebutuhan non fungsional alat pengukur curah hujan

No	Kebutuhan non fungsional alat pengukur curah hujan	Hasil pengujian
1	Alat dapat menyala selama 24 jam	BERHASIL
2	<i>Casing</i> dapat melindungi komponen dari air dan terik matahari	BERHASIL



Gambar 13 Antarmuka aplikasi *mobile* hasil pengujian.

## SIMPULAN

Berdasarkan hasil pengujian terhadap sistem pengukur curah hujan berbasis IoT dapat disimpulkan bahwa sistem dapat berjalan sesuai dengan kebutuhan fungsional dan nonfungsional yang telah didefinisikan. Curah hujan dapat diukur oleh sistem tersebut dan laporannya berhasil ditampilkan secara *real time* di aplikasi *mobile* yang digunakan oleh pengguna. Mengingat implementasi pada penelitian ini masih berada pada tingkat prototipe dan belum digunakan pada lingkungan nyata, maka pada penelitian selanjutnya dapat dilakukan instalasi sistem pada lingkungan nyata dan dilakukan evaluasi terhadap validitas hasil pengukuran sistem.

## UCAPAN TERIMA KASIH

Terimakasih kami ucapkan kepada XCAMP PT XL AXIATA yang telah memfasilitasi penelitian ini.

## DAFTAR PUSTAKA

- Alam MJ, Rafi SA, Badhan AA, Islam MN, Shuvo SI, Saleque AM. 2021. Design and implementation of a low-cost IoT based tipping bucket rain gauge. Di dalam: *3<sup>rd</sup> International Conference on Emerging Smart Computing and Informatics (ESCI)*; Pune, 2021 Mar 5-7. New York(US): IEEE. hlm. 279-283.
- Battan LJ. 1987. Precipitation. In *J.E. Oliver and R. W. Fairbridge (Eds.) The Encyclopedia of Climatology*. New York(US):Van Norstrand Reinhold, 688.
- Cirani S, Ferrari G, Picone M, Veltri L. 2018. *Internet of Things: Architectures, Protocols and Standards*. New Jersey(US): John Wiley & Sons.
- Gowda M, Gowda J, Iyer S, Pawar M, Gaikwad V. 2020. Power consumption optimization in iot based wireless sensor node using ESP8266. Di dalam: *International Conference on*

- Automation, Computing and Communication 2020*; Nerul, 2020 Jun 27-28. Navi Les Ulis Cedex A (UK): EDP Sciences. 32: 03048.
- Hermawan E. 2010. Pengelompokan pola curah hujan yang terjadi di beberapa kawasan P. Sumatera berbasis hasil analisis teknik spektral. *Jurnal Meteorologi dan Geofisika*. 11(2):75-85.
- Katriani L, Darmawan D. 2021. Design of automatic Rain Gauge Prototype (ARG) as an early warning indicator for cold lava flood based on the Internet of Things (IoT). *Journal of Physics: Conference Series*. 1805(1):012013.
- Maftukhah T, Wijonarko S, Rustandi D. 2016. Comparison and correlation among measurement results of observatory, Hellman, and tipping bucket sensors. *Instrumentasi*. 40(1):7-14.
- Manullang VS, Tamba T. 2013. Modifikasi penakar hujan otomatis tipe *tipping bucket* dengan *hall effect sensor* ATS276. *Saintia Fisika*. 6(1).
- NodeMCU Team. 2018. NodeMCU documentation [Internet]. [diakses 2022 Apr 24]; <https://nodemcu.readthedocs.io/en/release/>.
- Raghava TKV, Wani SP. 2014. Internet enabled tipping bucket rain gauge. Di dalam: *2014 International Conference on Computer Communication and Informatics*; Coimbatore, 2014 Jan 3-5. New York(US): IEEE. pp. 1-5.
- Ramadhan MF, Kurniawan E, Sugiana A. 2020. Perancangan sistem pemantauan banjir dan pencegahan dini berbasis *Internet of Things* (IoT). *eProceedings of Engineering*. 7(1): 162-169.
- Royce WW. 1987. Managing the development of large software systems: concepts and techniques. Di dalam: *Proceedings of the 9th international conference on Software Engineering*; Monterey California, 1987 Mar. Washington(US): IEEE Computer Society Press. hlm. 328-338.
- Shafique K, Khawaja BA, Sabir F, Qazi S, Mustaqim M. 2020. Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. *IEEE Access*. 8: 23022-23040.
- World Meteorological Organization. 2006. *Technical regulations*. Secretariat of the World Meteorological Organization.
- Yuwono T, Nahar A, Ruzardi, Aminudin A. 2018. Design of IoT for weather monitoring using Raspberry. Di dalam: *AIP Conference Proceedings*; Maharashtra, 2018 Jul 5-6. New York(US): AIP Publishing LLC. hlm. 020298.