

BAB II

TEORI PENUNJANG

2.1 Battery Monitoring System

Battery Monitoring System merupakan sebuah sistem yang dibuat untuk melakukan pemantauan terhadap kinerja baterai agar tidak terjadinya aktivitas diluar standar pemakaian baterai. Dengan adanya *Battery Monitoring System*, parameter pada baterai seperti tegangan, arus, temperatur, Status pengisian baterai (*State of Charge*) dan Status kesehatan baterai (*State of Health*) dapat dipantau melalui system ini[4]. Tujuan dari pembuatan *Battery Monitoring System* adalah untuk melakukan tindakan pencegahan kerusakan terhadap baterai supaya baterai tetap bekerja secara optimal.

2.2 Baterai

Baterai adalah sebuah alat yang dapat merubah energi kimia menjadi energi listrik melalui reaksi elektrokimia oksidasi dan reduksi *redox*. Reaksi terjadi dengan perpindahan elektron dari kutub negatif ke kutub positif. Secara umum baterai terdiri dari satu atau lebih sel yang terhubung secara seri, paralel ataupun gabungan seri-paralel tergantung output dari tegangan dan kapasitas.

Berdasarkan jenisnya baterai dikelompokkan menjadi 2 yaitu, baterai primer dan baterai sekunder. Baterai primer (*nonrechargeable battery*) adalah baterai yang tidak dapat diisi muatan listrik kembali. Contoh baterai primer adalah baterai *zinc carbon* dan baterai *alkaline-manganese*. Baterai sekunder (*rechargeable battery*) adalah baterai yang dapat diisi muatan listrik kembali atau dapat dilakukan proses *charge* dan *discharge* berulang-ulang. Contoh baterai sekunder adalah baterai *lead-acid* dan baterai *lithium-ion*[5].

2.2.1 Baterai Valve Regulated Lead-Acid

Baterai *lead-acid* termasuk kategori baterai *rechargeable*. Meskipun baterai *lead-acid* mempunyai energi yang sangat rendah jika dibandingkan

dengan berat dan volumenya, Tetapi baterai *lead-acid* mampu menghasilkan arus yang relatif lebih tinggi jika dibandingkan dengan beratnya.



Gambar 2. 1 Baterai VRLA

Sumber Gambar: <http://www.panasonic.com>

Baterai *lead-acid* diklasifikasikan menjadi baterai *lead-acid* konvensional dan baterai *valve-regulated lead-acid*. Pada baterai *lead-acid* konvensional hasil gas dari proses reaksi kimia akan terbuang langsung ke atmosfer dan pada *valve-regulated lead-acid* (VRLA) gas hasil reaksi kimia akan direkombinasikan lagi menjadi air[5].

Dalam penggunaannya, baterai VRLA mempunyai kelebihan serta kekurangan. Berikut kekurangan serta kelebihan baterai VRLA terangkum dalam tabel:

Tabel 2. 1 Kelebihan dan Kekurangan Baterai VRLA

Kelebihan	Kekurangan
Bebas dari perawatan	Densitas energi relatif kecil
State of Charge dapat ditentukan dari pengukuran voltase baterai	Umur siklus lebih pendek daripada baterai <i>sealed nickel-cadium</i>
Tersedia baterai satu <i>cell</i> 2V hingga 48V	Lebih sensitif terhadap temperatur lingkungan daripada baterai <i>lead acid</i> konvensional

Kelebihan	Kekurangan
Harga relatif murah	
<i>High-rate capability</i>	
<i>Moderate life time on floating service</i>	

Pada penelitian ini digunakan sebuah baterai berjenis VRLA dengan spesifikasi sebagai berikut:

Tabel 2. 2 Spesifikasi Baterai VRLA

Spesifikasi	Keterangan
Merk	Panasonic
Tipe	LC-V127R2
Tegangan	12 V
Kapasitas	7,2 Ah
Dimensi (P X L X T)	151 mm x 64,5 mm x 94mm
Berat	2,30 kg
Terminal	Faston 187 & 250M
Hambatan Dalam	24 mΩ
Masa Hidup	3 Tahun pada 25°C 5 Tahun pada 20°C

2.3 *State of Charge (SoC)*

SoC adalah salah satu parameter penting yang dipantau oleh BMS. Dengan menggunakan SoC, pengguna dapat mengetahui kapan baterai harus melakukan *charging* dalam bentuk persentase. Ada beberapa cara untuk menentukan SoC baterai, yaitu dari tegangan, arus, dan suhu baterai. Metode paling sederhana adalah dengan melakukan pengukuran langsung OCV atau tegangan beban, kemudian memperkirakan SoC dari karakteristik pelepasan yang telah disimpan sebelumnya[6]. Namun, metode ini mempengaruhi estimasi karena kesalahan pengukuran yang kecil. SoC dapat didefinisikan dalam persamaan:

$$SoC = \frac{C_{releaseable}}{C_{max}} \times 100 \quad (2.1)$$

$C_{releaseable}$ adalah kapasitas muatan yang dapat dibuang *discharge* baterai, kemudian C_{max} adalah kapasitas muatan maksimum baterai, lalu SoC adalah rasio dari $C_{releaseable}$ dan C_{max} . Hasil dari SoC akan di tampilkan dalam bentuk persentase.

2.4 *State of Health (SoH)*

SoH merupakan indikator penurunan kinerja karena penuaan pada baterai. Faktor yang mempengaruhi penuaan baterai disebabkan oleh suhu ruangan dan baterai serta *internal resistance*. Keadaan baterai ditentukan berdasarkan persamaan[6]:

$$SoH = \frac{C_{actual}}{C_{fresh}} \times 100 \quad (2.2)$$

C_{actual} adalah kapasitas baterai dalam kondisi saat ini dan C_{fresh} adalah kapasitas pengenal baterai seperti yang dinyatakan pada *datasheet* baterai. Baterai dalam keadaan rusak, saat SoH mencapai 80%. Hasil dari SoH akan di tampilkan dalam bentuk persentase. Penulis mendefinisikan kurang dari 90% SoH sebagai aman, 50% SoH sebagai waspada, dan kurang dari 20% SoH sebagai kesalahan.

2.5 *Internet of Things (IoT)*

Internet of things (IoT) didefinisikan sebagai sebuah jaringan dengan masing-masing benda yang tertanam dengan sensor yang terhubung kedalam jaringan internet. Konsep IoT mencakup 3 elemen utama yaitu: benda fisik yang telah diintegrasikan pada modul sensor, koneksi internet, dan pusat data pada server untuk menyimpan data ataupun informasi dari aplikasi. Penggunaan benda yang terkoneksi ke internet akan menghimpun data yang kemudian terkumpul menjadi *big data* untuk kemudian diolah, dianalisa baik oleh instansi pemerintah, perusahaan terkait, maupun instansi lain kemudian dimanfaatkan bagi kepentingan masing-masing [7][8].

2.6 *Object Oriented Programming (OOP)*

Teknik *Object Oriented Programming* merupakan sebuah teknik pemrograman yang menggantikan teknik pemrograman berbasis prosedur.

OOP adalah cara atau paradigma pemrograman yang berorientasi objek dan kelas, dimana setiap objek memiliki atribut beserta fungsi yang saling berinteraksi seperti halnya objek. Dengan menggunakan OOP, program dapat dibuat lebih modular, rapi, mudah dipahami dan dikembangkan lebih lanjut tanpa banyak sintaks yang diubah.

Programmer lain dapat dengan mudah menggunakan *class* yang sudah dibuat tanpa mengetahui proses yang sedang berjalan dari *class* tersebut. Jika dibandingkan dengan kode program prosedural dimana kode dimodelkan menggunakan fungsi, maka pengembangannya cukup rumit karena kode-kode program tidak tertata dengan baik dan tidak dipisahkan sesuai fungsinya[9].

2.7 Edge Computing

Edge Computing adalah teknologi yang dapat melakukan komputasi di *network edge*, sehingga komputasi terjadi di dekat sumber data. Dalam *Edge Computing*, perangkat akhir tidak hanya menggunakan data tetapi juga menghasilkan data. *Edge Computing* mempunyai penyimpanan substansial disebut sebagai *cloudlets*, *micro datacentres* atau *fog node*. Biasanya ditempatkan di *Edge Internet* dekat perangkat seluler atau sensor[10]. *Edge Computing* mempunyai kelebihan dan kekurangan dapat dilihat pada table berikut:

Tabel 2. 2 Keuntungan dan Kekurangan Edge Computing

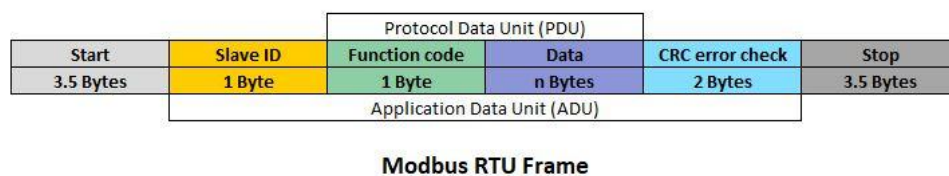
Keuntungan	Kekurangan
Respon waktu nyata.	Kapasitas penyimpanan terbatas.
<i>Latency</i> rendah.	<i>Edge Computing</i> membutuhkan jaringan berpemilik.
Edge dapat bekerja tanpa <i>cloud</i> dan meningkatkan keamanan data.	Perangkat IoT memiliki kualitas yang tinggi.
Struktur terdistribusi <i>Edge Computing</i> dapat mengurangi lalu lintas jaringan, penyimpanan dan biaya <i>bandwidth</i> .	Sulit menjaga keamanan data.
	Biaya tinggi untuk penyimpanan dan transmisi data.

2.8 Protokol Modbus

Protokol Modbus merupakan struktur pesan pada lapisan aplikasi yang banyak digunakan untuk komunikasi *master-slave* antar perangkat kendali. Pesan Modbus dikirim dari *master* ke *slave* yang memuat alamat *slave*, perintah, data dan *checksum*.

Protokol Modbus mendefinisikan *Protocol Data Unit* (PDU) terdiri atas alamat, *function code*, data dan *checksum*. Pesan dapat dibedakan menjadi dua yaitu *Request* dan *Respons*. *Request* berisi *function code* yang memerintahkan *slave* dengan alamat yang dituju untuk melakukan aksi atas perintah yang diberikan, dapat berupa perintah untuk membaca atau menulis data pada register yang dituju. Sedangkan *Respons* merupakan tanggapan *slave* atas perintah yang diberikan *master*. Protokol *Modbus* memungkinkan transmisi serial dengan dua mode yaitu RTU dan ASCII. Keuntungan dari *Modbus* RTU adalah lebih efisien dalam komunikasi karena dapat mengirim lebih banyak data dalam *baud rate* yang sama. Pada penelitian ini penulis menggunakan *Modbus* mode RTU[11].

Modbus RTU (*Remote Terminal Unit*) merupakan jenis *Modbus* yang digunakan pada komunikasi serial. Format *frame* pada *Modbus* RTU terdiri dari *start bit*, *slave ID*, *function code*, data, *Cyclic Redundant Check* (CRC), dan *end bit*[12].



Gambar 2. 2 Frame Modbus RTU

Sumber: picotech.com

Berikut merupakan fungsi dari setiap bagian pada *frame* Modbus:

1. *Start*, berfungsi sebagai penanda awal *frame* baru.
2. *Address*, alamat dari *slave*, yaitu dari 1 sampai dengan 247.
3. *Function Code*, merupakan perintah yang harus dikerjakan oleh *slave*.

4. *Data*, berisi alamat register, jumlah data, dan data yang akan ditulis. Jumlah *byte* data bergantung pada jumlah data yang akan dituliskan di *slave*.
5. *CRC*, untuk mendeteksi kerusakan data yang dikirim.
6. *End*, sebagai penanda akhir dari *frame* data.

Protocol Data Unit pada Modbus mempunyai sebuah *function code*. *Function code* merupakan perintah dari *master* yang harus dilakukan oleh *slave*[12]. Berikut merupakan tabel *Function code* pada Modbus.

Tabel 2. 3 Function Code Modbus

Jenis Data	Nilai Hex	Awal register
<i>Coil</i>	01 (01 hex)	00001
<i>Discrete Input</i>	02 (02 hex)	10001
<i>Input Register</i>	04 (04 hex)	30001
<i>Holding Register</i>	03 (03 hex)	40001

2.9 Protokol *Message Queue Telemetry Transport* (MQTT)

Protokol MQTT pertama kali di kembangkan oleh IBM pada akhir 1990-an. Pada awalnya protokol ini dikembangkan untuk menghubungkan sensor pada jalur pipa minyak dengan satelit. Protokol ini digunakan untuk pertukaran pesan secara asinkron, yang mana antar pengirim dan penerima terpisahkan skenario baik dalam konfigurasi perangkat maupun layanan yang diberikan[11].

Protokol MQTT menggunakan model komunikasi *publish* dan *subscribe*. Pengguna dibedakan menjadi dua macam yaitu *message broker* dan sejumlah *client*. *Broker* adalah server yang menampung semua pesan dari semua *client* dan kemudian merutekan pesan ke *client* yang dituju. *Client* adalah semua benda yang dapat berinteraksi dengan *broker* untuk mengirim atau menerima pesan. *Client* tersebut dapat berupa sensor yang mengirimkan data variable terukur, maupun pusat data yang mengolah data secara keseluruhan [13][14].

2.10 Mikrokontroler

Arduino adalah sebuah platform elektronik yang bersifat *open source* serta mudah digunakan. Hal tersebut ditunjukkan agar siapapun dapat membuat proyek interaktif dengan mudah dan menarik, beberapa alasannya sebagai berikut[15]:

- a. Murah, harga *board* Arduino relatif murah dibandingkan dengan *platform* mikrokontroler lain.
- b. *Cross-Platform*, Arduino Software IDE dapat dijalankan pada Sistem Operasi Windows, Machintosh OSX, dan juga Linux.
- c. *Simple*, Perangkat lunak Arduino IDE sangat mudah digunakan untuk pemula, namun cukup fleksibel untuk pengguna tingkat lanjut.
- d. Perangkat lunak Arduino diterbitkan sebagai *tools open source*.
- e. Arduino *board* diterbitkan dibawah lisensi *creative commons*, sehingga perancang sirkuit yang berpengalaman dapat membuat modul versi mereka sendiri, memperluasnya dan meningkatkannya.



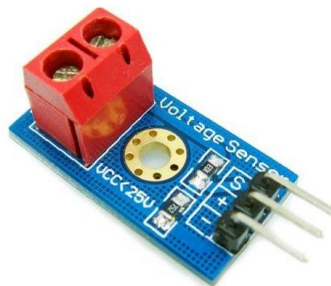
Gambar 2. 3 Arduino Nano

Sumber Gambar: store.arduino.cc/usa/arduino-nano

Arduino merupakan perpaduan dari hardware, bahasa pemrograman dan IDE *Integrated Development Environment*. Arduino mempunyai macam-macam *board*, yaitu Arduino UNO, Arduino NANO, Arduino MEGA, dan lain-lain. Pada penelitian ini, penulis akan menggunakan jenis Arduino NANO.

2.11 Sensor Tegangan

Sensor tegangan adalah suatu alat yang mengukur tegangan pada alat elektronik. Sensor tegangan umumnya berupa sebuah rangkaian pembagi tegangan atau yang biasa disebut *voltage divider*[16].



Gambar 2. 4 Sensor Tegangan

Sumber gambar: arduinolearning.com

2.12 Sensor Arus

ACS712 merupakan suatu IC terpakat berfungsi sebagai sensor arus untuk menggantikan transformator arus yang relatif besar dalam hal ukuran. Prinsipnya ACS712 sama seperti sensor *efek hall* lainnya yaitu memanfaatkan medan magnetik disekitar arus kemudian dikonversi menjadi tegangan yang linier dengan perubahan arus[17].



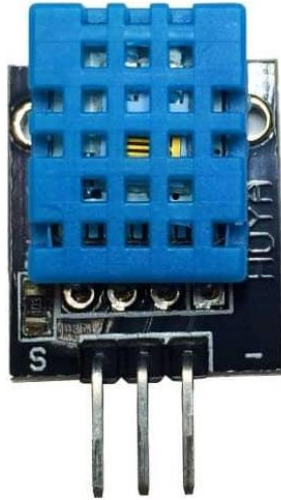
Gambar 2. 5 Sensor Arus ACS712

Sumber gambar: components101.com

Pada penelitian ini, digunakan sensor arus dengan tipe ACS712ELCTR-30A-T yang mempunyai jangkauan pengukuran = ± 30 A dengan sensitivitas sensitivitas = 66 mv.

2.13 Sensor Suhu dan Kelembapan

DHT-11 adalah sensor suhu dan kelembapan dengan tingkat kestabilan output yang cukup tinggi dan mempunyai keandalan jangka panjang.



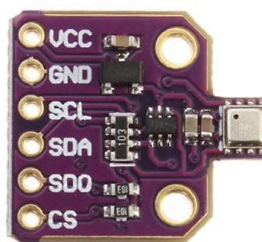
Gambar 2. 6 Sensor DHT11

Sumber gambar: circuitbasics.com

Sensor ini mengukur suhu disekitarnya dengan mengeluarkan sinyal digital pada pin data sehingga tidak memerlukan sinyal input analog lain dalam pengoperasiannya[18].

2.14 Sensor Environment

BME680 adalah sensor digital yang dibuat oleh Bosch. Sensor ini bereaksi terhadap sebagian besar senyawa *volatile* yang mencemari udara di dalam ruangan (Kecuali CO2).



Gambar 2. 7 Sensor BME680

Sumber gambar: sensate.io

BME680 mengukur jumlah kontaminan VOC di udara sekitar dan hasilnya dapat ditampilkan pada laptop atau situs web sebagai penghitung waktu, suhu, kelembaban, tekanan, dan hambatan yang berkorelasi dengan kualitas udara[19].

2.15 Modul Komunikasi RS485

RS-485 adalah teknik komunikasi data serial yang dikembangkan di tahun 1983 dimana dengan teknik ini, komunikasi data dapat dilakukan pada jarak yang cukup jauh yaitu 1,2 km.

Selain dapat digunakan untuk komunikasi *multidrop* yaitu berhubungan secara *one to many* dengan jarak yang jauh, komunikasi RS485 juga dapat digunakan untuk menghubungkan 32 unit beban sekaligus hanya dengan menggunakan dua buah kabel saja tanpa memerlukan referensi ground yang sama antara unit yang satu dengan unit lainnya. RS-485 menggunakan komunikasi *half-duplex* yaitu dapat melakukan pengiriman data dua arah dengan waktu yang tidak bersamaan[20].