



**TUGAS AKHIR - TE145561**

**PERANCANGAN *POWER DISTRIBUTION* PADA OPERATOR  
CONTROL UNIT (OCU) RHINO ROBOT**

Ahmad Nur Riza  
NRP. 1031150000043

Pembimbing  
Imam Arifin, S.T.,  
Mahardika Pratama S.T., M.Sc.,  
Lidikri Mohammad Fasha, S.T.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





## **TUGAS AKHIR - TE145561**

PERANCANGAN *POWER DISTRIBUTION* PADA OPERATOR  
CONTROL UNIT (OCU) RHINO ROBOT

Ahmad Nur Riza  
NRP. 1031150000043

Pembimbing  
Imam Arifin, S.T., M.T.  
Mahardika Pratama, S.T., M.Sc., Ph.D.  
Lidikri Mohammad Fasha, S.T.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018

-----Halaman ini sengaja dikosongkan-----



## **FINAL PROJECT - TE145561**

**DESIGN OF POWER DISTRIBUTION ON OPERATOR CONTROL  
UNIT (OCU) RHINO ROBOT**

**Ahmad Nur Riza  
NRP. 10311500000056**

**Supervisors  
Imam Arifin, S.T., M.T.  
Mahardika Pratama, S.T., M.Sc., Ph.D.  
Lidikri Mohammad Fasha, S.T.**

**DEPARTMENT OF ELECTRICAL AUTOMATION ENGINEERING  
Faculty of Vocations  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018**

-----Halaman ini sengaja dikosongkan-----

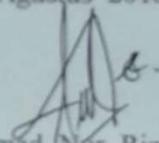
## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "*Perancangan Power Distribution pada Operating Control Unit (OCU) Rhino Robot*" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya saya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 2 Agustus 2018



Ahmad Nur Riza

NRP. 10311500000043

-----Halaman ini sengaja dikosongkan-----

PERANCANGAN *POWER DISTRIBUTION* PADA OPERATOR  
CONTROL UNIT (OCU) RHINO ROBOT

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Ahli Madya Teknik

Pada

Departemen Teknik Elektro Otomasi  
Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Pembimbing I,

Am Arifin, S.T., M.T.  
197302222002121001



Menyetujui,

Pembimbing II,

Mahardhika P., S.T., M.Sc., Ph.D.

Pembimbing III,

Lidikri Mohammad F., S.T.  
NIK. 020332016

**SURABAYA**  
**JULI, 2018**

**----Halaman ini sengaja dikosongkan----**

PERANCANGAN *POWER DISTRIBUTION* PADA *OPERATOR*  
*CONTROL UNIT (OCU) RHINO ROBOT*  
DI  
PT. BHIMASENA RESEARCH AND DEVELOPMENT

TUGAS AKHIR

Disusun oleh:

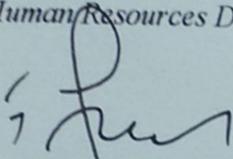
Ahmad Nur Riza

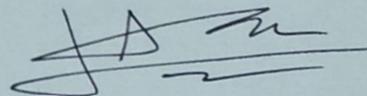
NRP. 1031150000043

Menyetujui,

Kepala *Human Resources Department*,

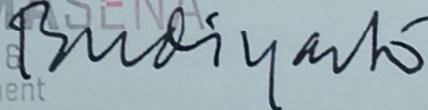
Pembimbing Perusahaan,

  
Fadli Tirmissi  
NIK. 020492016

  
Lidkri Mohmmad F., S.T.  
NIK. 020332016

*Chief Executive Officer,*





Dipl. -Ing. Aris Budiarto

**-----Halaman ini sengaja dikosongkan-----**

# **PERANCANGAN *POWER DISTRIBUTION* PADA *OPERATOR CONTROL UNIT (OCU) RHINO ROBOT***

Ahmad Nur Riza  
1031150000043

Pembimbing I : Imam Arifin, S.T., M.T.  
Pembimbing II : Mahardika Pratama, S.T., M.Sc., Ph.D.  
Pembimbing III : Lidikri Mohammad F., S.T.

## **ABSTRAK**

*Operator Control Unit (OCU)* tersusun dari beberapa komponen dengan kebutuhan tegangan yang berbeda-beda. Untuk mengatasi masalah tersebut, maka perlu dibuat manajemen sumber dengan melakukan perancangan *power distribution*. Dalam sistem tersebut dibuat sistem *powering* dengan fungsi untuk mengolah daya dari baterai dan menghasilkan tegangan keluaran yang sesuai dengan kebutuhan.

Baterai yang digunakan dipasang didalam OCU, sehingga sulit untuk mengetahui kondisi dari sumber. Untuk itu, dibuat sistem indikator yang memiliki hubungan dengan sumber, dimana indikator yang ditampilkan ada dua, yaitu indikator baterai dan *power*. Saat sumber habis dapat dilakukan pengisian ulang namun, untuk menggunakannya baterai harus dilepas dari perangkat terlebih dahulu. Melalui permasalahan tersebut, maka dibuat konsep sistem *charging* agar proses pengisian dapat dilakukan dengan mudah dengan menggunakan Metode *Constant Current Constant Voltage*.

Melalui pengujian yang telah dilakukan, sistem *powering* mampu menyalakan tiap-tiap beban dengan lama penggunaan selama 4 jam 44 menit. Dari pembacaan nilai SoC, dibandingkan dengan perhitungan manual kapasitas baterai. Melalui perbandingan tersebut diperoleh selisihnya, dimana rata-rata yang didapat kurang dari 1%. Acuan tegangan dan SoC untuk sistem *charging* saat baterai penuh sebesar 16.35 VDC dan 107%, sedangkan saat habis sebesar 11.9 VDC dan 0%.

Kata Kunci : *Operator Control Unit*, manajemen , *Power Distribution*, SoC.

**-----Halaman ini sengaja dikosongkan-----**

# **DESIGN OF POWER DISTRIBUTION ON OPERATOR CONTROL UNIT (OCU) RHINO ROBOT**

Ahmad Nur Riza  
10311500000043

*Supervisor I* : Imam Arifin, S.T., M.T.  
*Supervisor II* : Mahardika Pratama, S.T., M.Sc., Ph.D.  
*Supervisor III* : Lidikri Mohammad F., S.T.

## **ABSTRACT**

*Operator Control Unit (OCU) contain from some component with different voltage requirement. For resolve the problem, so need to be made source management with doing design power distribution. In the system is be made powering system with function for management power from battery and equal output voltage is suitable with requirement.*

*Battery need to installed in the OCU, so difficult for knowing condition from source. Therefor, be made indicator system is have relation with source, where displayed battery and power indicator. When source is empty, can did recharge battery, but for using it battery must to be removed from device previously. Based on the problem, so be made charging system concept so charging process can did easily using constant current constant voltage methods.*

*Based on the testing, powering system can activate each load with duration about 4 hours and 44 minutes. From reading SoC value, compared with manual calculate of capacity battery. By means of comparing obtained difference, where average obtained less than 1%. Voltage and SoC reference for charging system when battery is full amount 16.35VDC and 107%, while battery is empty amount 11.9VDC and 0%.*

*Keyword : Operator Control Unit, Management , Power Distribution, SoC.*

**-----Halaman ini sengaja dikosongkan-----**

## KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Allah SWT, karena berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan pengerjaan hingga menyusun buku Tugas Akhir dengan judul “Perancangan *Power Distribution* pada *Operator Control Unit (OCU)* Rhino Robot”. Pada kesempatan kali ini, penulis ingin menyampaikan rasa terimakasih kepada orang tua serta keluarga yang telah memberi dukungan dari awal hingga saat ini. Dosen pembimbing Bapak Imam Arifin, S.T., M.T. dari Institut Teknologi Sepuluh Nopember, Surabaya dan Bapak Mahardika Pratama S.T., M.Sc., Ph.D dari Nanyang Technology University, Singapura yang telah memberi bimbingan dalam proses mengerjakan. Bapak Lidikri Mohammad Fasha, S.T. selaku pembimbing yang telah memberikan bimbingan selama diperusahaan dengan sabar membimbing dari awal proses pengerjaan hingga akhir. Seluruh karyawan PT Bhimasena Research and Development, khususnya untuk Divisi Rhino Robot karena telah banyak membantu dalam proses mengerjakan. Segenap Direksi PT Bhimasena Research and Development karena telah memberi ijin untuk melakukan penelitian ditempat tersebut. Kepala Departemen dan segenap civitas akademik Teknik Elektro Otomasi yang telah mendidik mulai dari awal hingga akhir perkuliahan. Alief Ardiansyah, Tata Tanjung Tamara, Livian Tjandra, Varisa Rahmawati, dan Amiroatul Khoiro selaku teman seperjuangan dalam melakukan penelitian di PT Bhimasena Research and Development. Anggota Laboratorium Sistem Komputer dan Otomasi yang telah menemani baik dalam keadaan susah dan senang

Penulis menyadari bahwa dalam penyusunan laporan ini tidak lepas dari kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diperlukan demi kebaikan dan kesempurnaan penyusunan laporan dimasa yang akan datang. Semoga laporan ini dapat memberikan manfaat bagi penulis khususnya dan pembaca umumnya.

Surabaya, Juni 2018

Ahmad Nur Riza

**-----Halaman ini sengaja dikosongkan-----**

## DAFTAR ISI

LEMBAR PENGESAHAN .....	iii
LEMBAR PENGESAHAN PERUSAHAAN .....	v
ABSTRAK.....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL .....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan.....	3
1.5 Metodologi Penelitian.....	4
1.6 Sistematika Laporan.....	5
BAB II MANAJEMEN DAYA PADA OCU.....	7
2.1 EOD Robot .....	7
2.2 Rhino Robot.....	7
2.3 OCU.....	8
2.4 <i>Power Distribution</i> .....	9
2.5 Baterai Lithium Ion.....	10
2.6 Smart Battery .....	12
2.7 <i>System Management Bus (SMBus)</i> .....	12
2.8 <i>Voltage Regulator</i> .....	14
2.9 Efisiensi Daya .....	15
2.10 Inrush Current .....	16
2.11 Filter.....	17
2.12 State of Charge (SoC) .....	17
2.13 Charging System.....	18
2.14 Constant Current Constant Voltage (CCCV).....	19
BAB III PERANCANGAN <i>POWER DISTRIBUTION</i> .....	21
3.1 <i>Design Requirement</i> .....	21
3.2 Blok Fungsional Sistem .....	22
3.3 Perancangan Sistem Powering .....	23
3.3.1 Sistem Proteksi .....	24
3.3.2 Voltage Regulator.....	26

3.3.3	Pemilihan Kapasitor dan Induktor .....	28
3.3.4	Sistem Switching .....	29
3.3.5	Konfigurasi <i>Board Power</i> Ke Beban .....	30
3.4	Sistem Indikator .....	32
3.4.1	Indikator Power .....	32
3.4.2	Indikator Baterai .....	32
3.4.3	Pembuatan Host .....	33
3.4.4	Konfigurasi Multiplexer dan Host .....	34
3.4.5	Konfigurasi Board Indikator Baterai .....	35
3.4.6	Pembuatan Program Indikator Baterai .....	36
3.5	Perancangan Sistem Charging .....	38
3.5.1	LTC4100 .....	38
3.5.2	Algoritma Charging .....	39
3.5.3	Konfigurasi Charger .....	42
BAB IV HASIL DAN ANALISIS .....		45
4.1	Sistem Powering .....	45
4.1.1	Tegangan Keluaran .....	45
4.1.2	Efisiensi .....	48
4.2	Indikator Baterai .....	51
4.2.1	Program Indikator Baterai .....	52
4.2.2	Data SoC .....	54
4.3	Hubungan Tegangan dan SoC .....	54
4.4	Validasi Hasil Pengujian .....	55
BAB V PENUTUP .....		57
DAFTAR PUSTAKA .....		59
LAMPIRAN .....		63

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Contoh Robot EOD .....	8
<b>Gambar 2.2</b> Pembagian Tugas Pengerjaan.....	9
<b>Gambar 2.3</b> Reaksi Kimi Baterai Lithium Ion .....	10
<b>Gambar 2.4</b> Topologi SMBus .....	13
<b>Gambar 2.5</b> Kondisi <i>Start</i> dan <i>Stop</i> .....	13
<b>Gambar 2.6</b> Konfigurasi Dasar <i>Linier Voltage Regulator</i> dengan <i>Feedback</i> .....	15
<b>Gambar 2.7</b> Grafik Arus dan Tegangan dari CCCV <i>Charging</i> .....	19
<b>Gambar 3.1</b> Konfigurasi Sistem <i>Power Distribution</i> OCU Menggunakan Konsep Sistem <i>Charging</i> .....	22
<b>Gambar 3.2</b> Konfigurasi Sistem <i>Power Distribution</i> OCU .....	23
<b>Gambar 3.3</b> Diagram Blok Sistem <i>Powering</i> .....	24
<b>Gambar 3.4</b> <i>Flowchart</i> Aktivasi OCU.....	25
<b>Gambar 3.5</b> Rangkaian <i>Voltage Regulator</i> LM2576/LM2596.....	27
<b>Gambar 3.6</b> Grafik Menentukan Induktor Pada Keluaran 5VDC .....	28
<b>Gambar 3.7</b> Grafik Menentukan Induktor Pada Keluaran 12VDC .....	29
<b>Gambar 3.8</b> Rangkaian <i>Switching On/Off Voltage Regulator</i> Menggunakan <i>Optocoupler</i> .....	30
<b>Gambar 3.9</b> <i>Switching On/Off</i> dan Indikator <i>Power</i> Menggunakan Relay .....	30
<b>Gambar 3.10</b> Konfigurasi <i>Board Power</i> .....	31
<b>Gambar 3.11</b> Konfigurasi <i>Multiplexer</i> dengan Baterai dan <i>Host</i> .....	34
<b>Gambar 3.12</b> Konfigurasi <i>Board</i> Indikator Baterai .....	35
<b>Gambar 3.13</b> Pengaturan Port I2C .....	36
<b>Gambar 3.14</b> Pengaturan USART .....	37
<b>Gambar 3.15</b> Rangkaian <i>Charger</i> Menggunakan LTC4100 .....	39
<b>Gambar 3.16</b> <i>Flowchart</i> Sistem <i>Charging</i> .....	42
<b>Gambar 3.17</b> Konfigurasi <i>Charger</i> pada Sistem .....	43
<b>Gambar 4.1</b> Board Sistem <i>Powering</i> .....	45
<b>Gambar 4.2</b> Grafik <i>Start</i> Tegangan pada Keluaran <i>Voltage</i> <i>Regulator</i> .....	47
<b>Gambar 4.3</b> Uji Coba Indikator Baterai.....	52

**Gambar 4.4** Grafik Hubungan SoC dan Tegangan Baterai .....55

## DAFTAR TABEL

<b>Tabel 2.1</b> Kelebihan dan Kekurangan Penggunaan Baterai Lithium Ion.....	11
<b>Tabel 2.3</b> Perbedaan Karakteristik Elektrik I2C dan SMBus.....	14
<b>Tabel 3.1</b> Estimasi <i>Fuse</i> pada Masing-masing Beban.....	26
<b>Tabel 3.2</b> Pengelompokan Regulator dan Beban .....	27
<b>Tabel 4.1</b> Tegangan Keluaran Voltage Regulator Tanpa Beban .....	46
<b>Tabel 4.2</b> Efisiensi <i>Voltage Regulator</i> 1 (Ubec) .....	48
<b>Tabel 4.3</b> Data Efisiensi <i>Board</i> Sistem <i>Powering</i> .....	49
<b>Tabel 4.4</b> Kebutuhan Daya pada Beban OCU.....	50
<b>Tabel 4.5</b> Sampel Nilai SoC dan Kapasitas Baterai .....	54
<b>Tabel 4.6</b> Validasi Desain .....	56

**-----Halaman ini sengaja dikosongkan-----**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

PT Bhimasena Research and Development merupakan perusahaan yang bergerak pada bidang penelitian dan pengembangan. Fokus penelitian berada pada alat-alat pendukung dalam militer. Perusahaan ini memiliki beberapa divisi yaitu Rhino Robot, *Ground Vehicle* (GV), *Underwater Vehicle* (UV), dan *Unmanned Aerial Vehicle* (UAV). Pada Divisi UV dibagi menjadi dua bidang, yaitu Kendaraan Tsempur Bawah Air (KTBA) dan *Diver Propulsion Vehicle* (DPV). Pada UAV dibagi menjadi tiga bidang, yakni Helikopter, *Vertical Take Off Landing* (VTOL) dan UAV *Backpack*.

Salah satu produk yang dihasilkan yaitu Kendaraan Jihandak. Produk tersebut merupakan kendaraan pelengkap pada tim penjinak bahan-bahan peledak. Didalamnya telah dilengkapi dengan beberapa peralatan yang mampu mendukung dalam pelaksanaan tugas. Salah satunya adalah Robot *Explosive Ordnance Disposal* (EOD).

Pada saat ini, untuk melengkapi kendaraan tersebut dengan Robot EOD, perusahaan harus mendatangkan produk dari luar negeri. Kelemahan dari cara tersebut adalah saat terjadi kerusakan pada robot, perbaikan harus dilakukan oleh perusahaan pembuat robot dan tidak diperbolehkan ditangani oleh pihak PT Bhimasena Research and Development. Namun, untuk melakukan perbaikan harus menunggu dalam waktu yang lama yakni sekitar 2 hingga 3 bulan. Selain itu, dari segi biaya yang dibutuhkan cukup mahal.

Untuk itu, perusahaan ini mengembangkan Rhino Robot yang ditangani oleh Divisi Rhino Robot. Fungsi robot tersebut adalah untuk mendekati dan mengambil benda-benda yang dianggap berbahaya [1]. Dengan alat tersebut diharapkan mampu mengurangi kemungkinan kecelakaan yang dialami oleh pengguna. Terdapat dua bagian pada robot tersebut, yaitu *chasis* dan *arm*. Masing-masing bagian tersebut

memiliki fungsi sendiri-sendiri, *chasis* digunakan untuk mobilitas robot dan *arm* digunakan untuk mengambil benda yang diinginkan. Selain itu, pada *chasis* juga dilengkapi *flipper*, fungsinya adalah untuk membantu robot agar dapat melewati medan yang tidak rata seperti naik dan turun tangga.

Untuk mengendalikan robot digunakan modul yang bernama Operator Control Unit (OCU) [2]. Pada versi pertama, perangkat tersebut dibuat menggunakan laptop *toughbook* dan *gamepad*. Untuk mengendalikan robot digunakan *gamepad* yang telah terhubung dengan *Graphical User Interface* (GUI) yang ada pada layar laptop. Kelemahan dari sistem ini adalah terlalu banyak kombinasi tombol yang harus dihafalkan oleh pengguna. Selain itu, dari pihak konsumen lebih menyukai kendali robot menggunakan tombol-tombol dan *joystick* yang memiliki sedikit kombinasi tombol.

Berdasarkan permasalahan yang telah disebutkan pada paragraf sebelumnya, maka dibuat OCU versi 2. Pada versi ini perangkat dibuat menggunakan modul yang disusun pada pelican *case*. Didalamnya telah dilengkapi dengan beberapa tombol serta layar yang digunakan untuk menampilkan GUI dan video atau gambar yang telah diambil melalui kamera robot.

Alat tersebut tersusun dari beberapa komponen seperti LCD, dan *main controller* dimana pada sistem ini digunakan LattePanda. Dari setiap beban membutuhkan *supply* daya agar dapat bekerja, sehingga perlu disambungkan dengan sumber. Disisi lain, dalam transmisi daya masalah yang sering menjadi perhatian adalah efisiensi dan kerugian energi [3]. Melalui proses penyambungan tersebut, perlu dilakukan manajemen sumber ke masing-masing beban agar daya yang diberikan sesuai dengan kebutuhan. Untuk melakukan pengelolaan, maka perlu dilakukan perancangan *power distribution*.

Dalam perancangan OCU sendiri terdapat tiga bagian yang dikerjakan yaitu pembuatan GUI, sistem *interface* dan sistem *power distribution*. Sistem *interface* berfungsi untuk mengkoordinasi tombol-tombol dan *joystick* yang ada pada panel. Sedangkan, pada penelitian ini, dibuat sistem yang dapat memberikan *supply* daya sesuai dengan

kebutuhan komponen. Selanjutnya dibuat sistem tambahan yang memiliki hubungan dengan sumber agar dapat membantu mempermudah pengguna saat menggunakan alat.

## **1.2 Permasalahan**

Dalam pembuatan OCU terdapat beberapa komponen yang membutuhkan *supply* daya dengan tegangan berbeda-beda. Kebutuhan tersebut harus terpenuhi agar sistem dapat bekerja. Baterai dipasang didalam alat, sehingga pengguna sulit untuk mengetahui kapasitas sumber yang sedang digunakan. Dari perusahaan pembuat baterai, menyediakan *charger* berupa modul. Oleh karena itu, jika pengguna ingin melakukan proses pengisian, maka baterai harus dilepas terlebih dahulu dari alat.

## **1.3 Batasan Masalah**

Terdapat beberapa batasan dalam penelitian yang dilakukan, yaitu agar sistem dapat bekerja dilakukan pembagian *supply* daya sesuai dengan kebutuhan komponen, dimana dalam hal ini dibutuhkan tegangan sebesar 5VDC dan 12VDC. Dalam penampilan indikator baterai, digunakan 4 led dengan menggunakan parameter SoC dari kedua bagian baterai. Sedangkan, pada pengambilan data, diambil dari satu bagian baterai. Untuk sistem *charging* dibuat secara konsep menggunakan Metode *Constant Current Constant Voltage* (CCCV).

## **1.4 Tujuan**

Dari hasil perancangan sistem ini, diharapkan mampu memberikan *supply* daya pada tiap komponen agar tiap komponen dapat bekerja. Selanjutnya menampilkan indikator pada panel OCU agar dapat mempermudah pengguna untuk mengetahui kondisi sumber yang digunakan. Serta mampu memberikan sebuah konsep sistem pengisian yang mudah dilakukan, sehingga untuk melakukan pengisian ulang baterai, pengguna tidak perlu melepas baterai terlebih dahulu.

## 1.5 Metodologi Penelitian

Dalam proses mengerjakan penelitian, ada beberapa tahap yang dilakukan, yaitu: tahap persiapan, tahap perancangan, tahap pembuatan, tahap pengujian dan analisis, serta tahap pembautan, serta tahap penyusunan laporan.

Pada tahap persiapan ini dilakukan studi literatur mengenai materi-materi yang memiliki hubungan dengan topik penelitian. Sumber yang digunakan bermacam-macam mulai dari web hingga paper. Diskusi dengan pembimbing yang berada di perusahaan juga dilakukan, tujuannya adalah untuk meluruskan jika terjadi pemahaman materi yang kurang sesuai. Hasil yang didapat pada tahap ini digunakan sebagai modal untuk membuat perancangan sistem.

Dilanjutkan dengan tahap perancangan, langkah yang dilakukan adalah perancangan sistem, yang meliputi perangkat keras dan perangkat lunak. Untuk perancangan perangkat keras dibuat *board* indikator baterai serta sistem powering untuk OCU, mulai dari bagaimana menyalakan hingga mematikannya. Selanjutnya, untuk perancangan perangkat lunak dibuat program indikator baterai.

Tahap pembuatan, terdapat dua bagian yaitu pembuatan perangkat keras dan perangkat lunak. Perangkat keras yang dibuat meliputi *board* sistem *powering* dan *board* indikator baterai. Untuk pembuatan perangkat lunak dibuat program indikator baterai.

Selanjutnya dilakukan tahap pengujian dan analisis, dari hasil pembuatan sistem, tahap selanjutnya adalah melakukan pengujian terhadap alat yang telah dibuat. Kemudian dilakukan analisis dari data yang telah diperoleh.

Yang terakhir adalah tahap penyusunan laporan, setelah alat berhasil dibuat dan dapat bekerja dengan baik, pengambilan data serta analisis terpenuhi selanjutnya dapat mengatasi permasalahan yang diambil. Tahap selanjutnya adalah pembuatan laporan untuk buku Tugas Akhir. Diharapkan buku ini dapat bermanfaat bagi pembaca dan bisa dijadikan pedoman dalam melanjutkan dalam pengembangan ide ini.

## 1.6 Sistematika Laporan

Sistematika penulisan Laporan ini, terbagi menjadi lima bab yang diuraikan sebagai berikut:

### BAB I PENDAHULUAN

Berisi tentang latar belakang, perumusan masalah, batasan masalah, tujuan, metodologi penelitian, sistematikan laporan buku Tugas Akhir.

### BAB II PENUNJANG *POWER DISTRIBUTION*

Menjelaskan teori-teori yang digunakan sebagai landasan dalam penelitian ini.

### BAB III PERANCANGAN *POWER DISTRIBUTION*

Membahas perencanaan dan pembuatan sistem. Dalam hal ini dibagi menjadi dua bagian yaitu perangkat keras yang meliputi pembuatan konsep dan desain PCB, serta perencanaan serta pembuatan perangkat lunak yang meliputi *flowchart* dan program.

### BAB IV PENGUJIAN DAN ANALISIS

Memaparkan mengenai hasil pengujian dan analisis data yang telah diperoleh.

### BAB V PENUTUP

Dalam bagian ini dijelaskan mengenai kesimpulan yang didapat dari uji coba dan saran-saran untuk pengembangan alat lebih lanjut.

-----Halaman ini sengaja dikosongkan-----

## **BAB II**

### **MANAJEMEN DAYA PADA OCU**

Dalam proses mengerjakan, dilakukan studi literatur mengenai materi-materi yang dapat menunjang dalam tahap perancangan dan implementasi sistem. Terdapat beberapa teori dengan berbagai macam sumber yang digunakan.

#### **2.1 EOD Robot**

Perkembangan robot pada era saat ini sangat pesat, salah satunya adalah dengan munculnya Robot *Explosive Ordnance Disposal* (EOD) dalam dunia militer. Robot tersebut merupakan robot yang dapat menggantikan manusia untuk mengintai, menghapus, dan menangani bahan peledak atau barang berbahaya lainnya dilingkungan berbahaya secara langsung [4].

Robot EOD sendiri pertama kali ditemukan pada tahun 1972 oleh Letnan Kolonel Peter Miller. Robot tersebut digabung dengan kerangka yang terdapat gerobak anti ledakan. Gerobak tersebut dirancang dengan dilengkapi pengait, fungsinya adalah untuk dikaitkan dengan mobil, sehingga dapat ditarik ke lokasi yang aman tanpa membahayakan petugas [5].

#### **2.2 Rhino Robot**

Pada era saat ini, hampir seluruh kegiatan manusia menggunakan bantuan robot. Begitu pula dalam dunia militer, robot menjadi bagian penting dalam menyelesaikan tugas. Contohnya adalah Rhino Robot, dimana robot tersebut merupakan robot EOD yang dikembangkan oleh PT Bhimasena Research and Development. Tujuan dibuat robot ini adalah untuk menggantikan manusia dalam menangani benda-benda yang berbahaya seperti bom [1]. Dengan adanya robot tersebut diharapkan mampu mengurangi kecelakaan yang terjadi pada pengguna.

Robot ini masih dalam tahap pengembangan, sehingga gambarnya belum dapat dipublikasikan. Namun, desian yang telah dibuat hampir

sama dengan desain robot EOD yang ada dipasaran, seperti yang telah ditunjukkan pada Gambar 2.1. Terdapat dua bagian utama yaitu *chasis* dan *arm*.



**Gambar 2. 1** Contoh Robot EOD

Bagian *chasis* berfungsi untuk menyangga *arm* dan mobilisasi robot. Disisi kanan kiri dilengkapi dengan *belt* dan digerakkan menggunakan motor yang telah dipasang didalamnya. Selain itu, pada bagian depan dilengkapi dengan *flipper*, fungsinya adalah untuk melewati medan yang tidak rata seperti tangga. Pada bagian depan dan belakang dilengkapi dengan kamera. Gambar yang telah diambil akan dikirimkan ke pengguna, sehingga dapat diketahui kondisi medan robot berada.

*Arm* berfungsi untuk mengambil objek yang diinginkan. Untuk mengambilnya pada bagian ujung telah dilengkapi dengan capit. Selain itu, pada bagian ini juga dilengkapi dengan kamera. Gambar yang telah diambil oleh kamera tersebut selanjutnya akan dikirimkan ke pengguna, sehingga dapat diketahui apakah objek berhasil diambil atau tidak

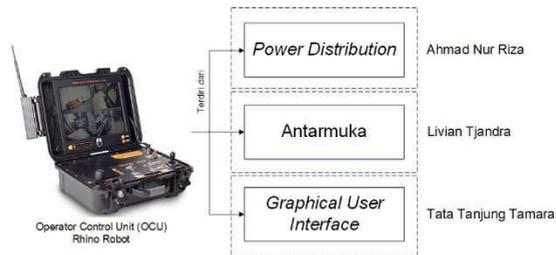
### 2.3 OCU

Perangkat tersebut merupakan antarmuka kontrol perangkat keras atau perangkat lunak untuk robot. Untuk mengendalikan Rhino Robot digunakan modul yang bernama OCU [2]. Dengan alat tersebut, pengguna dapat mengontrol robot dari jarak jauh. Untuk mengendalikan robot, alat tersebut telah dilengkapi dengan LCD yang digunakan untuk menampilkan *graphic user interface* dan video yang diterima dari

kamera robot. Disediakan beberapa tombol serta *joystick* untuk mengendalikan pergerakan robot.

Didalamnya terdiri dari mini PC dan beberapa mikrokontroler yang difungsikan untuk menyimpan program serta mengambil data yang akan dikirim pada robot. Selain itu, juga terdapat beberapa rangkaian yang digunakan untuk manajemen sumber. Terdapat *transmitter*, fungsinya adalah untuk mengirim dan menerima data dari robot.

Dalam pengerjaan perangkat ini terdapat tiga bagian utama yang dikerjakan, yaitu pembuatan *power distribution*, *graphical user interface*, dan sistem antarmuka, seperti yang ditunjukkan pada gambar 2.2. Dimana penulis mendapat tugas untuk mengerjakan bagian *power distribution* sedangkan untuk bidang lain dikerjakan oleh anggota tim yang lain.



**Gambar 2. 2** Pembagian Tugas Pengerjaan

## 2.4 *Power Distribution*

Dalam literatur [6] menyebutkan bahwa *DC distribution* terdiri dari sumber, konduktor, dan komponen. *Power Distribution* sendiri merupakan tahap pengaliran daya dari sumber kepada konsumen atau komponen. Pada penelitian ini, sumber yang digunakan adalah baterai. Untuk menghubungkan komponen dengan sumber digunakan konduktor, dimana pada sistem ini dilakukan pengolahan sumber terlebih dahulu sebelum disalurkan pada tiap-tiap komponen.

Efisiensi dan gangguan tegangan menjadi aspek penting dalam perancangan sistem ini [3]. Dimana nilai efisiensi menentukan berapa

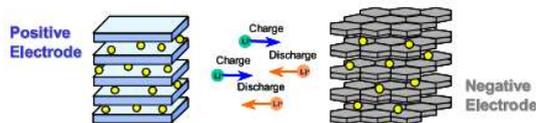
lama alat dapat digunakan. Keluaran dari tegangan DC biasanya terdapat gangguan, hal tersebut dapat dilihat melalui pengukuran menggunakan osiloskop. Biasanya terdapat komponen yang membutuhkan *supply* dengan gangguan pada tegangan bernilai kecil.

Dari pembuatan sistem ini dapat diketahui berapa lama alat dapat dioperasikan. Untuk menentukannya dapat dihitung daya masukan total dengan memperhatikan nilai efisiensi. Selanjutnya dilakukan perhitungan berapa besar arus yang dibutuhkan untuk memenuhi kebutuhan komponen, kemudian nilai tersebut dijadikan pembagi dari kapasitas baterai yang digunakan.

## 2.5 Baterai Lithium Ion

Baterai jenis ini telah dikembangkan sebagai solusi untuk menyediakan energi yang lebih besar dalam kemasan lebih kecil dan ringan [7]. Sehingga sering menjadi pilihan untuk digunakan pada kendaraan listrik dan perangkat elektronik *portable* [8], [9]. Dalam beberapa literatur bahan yang digunakan untuk menyusun baterai lithium ion terdiri dari *lithium cobalt cathode*, *graphite anode*, pelarut elektrolit organik termasuk garam lithium dan separator [10]–[12]. Fungsi utama separator baterai adalah untuk menyekat positif dan negatif elektroda, menahan elektrolit dan mengirimkan lithium ion.

Reaksi kimia *lithium ion battery* dapat dilihat pada gambar 2.3, dimana kedua elektroda mempunyai struktur lapisan. Ketika *charging*, lithium ion keluar dari katoda dan bergerak melalui elektrolit lalu disimpan diantara lapisan graphit anoda, sehingga menimbulkan beda potensial. Ketika melepaskan energi (*discharge*) reaksinya berubah. Lithium ion merupakan media dimana elektron dibawa dan dimasukkan dalam bahan elektroda positif.



Gambar 2. 3 Reaksi Kimi Baterai Lithium Ion

Baterai ini memiliki efisiensi *charge/discharge* hampir 100%, kecuali pada siklus pertama. Berbeda dengan baterai dari *Nickel Cadmium* (Ni-Cd) dan *Nickel Metal* (Ni-MH) yang efisiensinya dibawah 100%. Selain itu, dalam lithium ion menggunakan pelarut organik pada elektrolit, dimana hal tersebut membedakan dengan baterai isi ulang *alkaline* [13].

**Tabel 2.1** Kelebihan dan Kekurangan Penggunaan Baterai Lithium Ion

<b>Kelebihan</b>	<ul style="list-style-type: none"> <li>• Spesifikasi energi dan komponen yang tinggi dengan <i>power cell</i></li> <li>• Siklus panjang dan memperpanjang umur simpan serta bebas perawatan</li> <li>• Kapasitas tinggi, resistansi internal rendah dan efisiensi <i>coloumbic</i> yang baik</li> <li>• Algoritma pengisian sederhana dan waktu yang dibutuhkan cukup singkat</li> <li>• <i>Self discharge</i> rendah (kurang dari setengah NiCd dan NiMH)</li> </ul>
<b>Kekurangan</b>	<ul style="list-style-type: none"> <li>• Membutuhkan rangkaian proteksi untuk mencegah pelarian pelarian panas jika terjadi <i>stress</i></li> <li>• Degradasi pada suhu tinggi dan ketika disimpan pada tegangan tinggi</li> <li>• Tidak dapat melakukan pengisian cepat pada suhu beku (&lt;0°C, &lt;32°F)</li> <li>• Pengaturan transmisi diperlukan jika terjadi pengiriman dalam jumlah besar</li> </ul>

Secara umum, akhir siklus hidup baterai sekunder ditetapkan saat kapasitasnya berada dibawah 60% kapasitas total dan tidak pulih pada siklus berikutnya. Sebagian siklus bergantung pada beberapa kondisi seperti *charge*, *depth of discharge*, arus dan suhu lingkungan Kondisi-kondisi yang dapat mengurangi siklus hidup baterai yaitu *over-charging* dengan tegangan yang terlalu besar, *charging* terus menerus, *Charging* dan *discharging* dengan arus yang berlebihan dan Suhu lingkungan melebihi dari yang disarankan [14]. Penggunaan baterai jenis ini, memiliki beberapa kelebihan dan kekurangan, diantaranya dapat dilihat pada Tabel 2.1 [15].

## 2.6 Smart Battery

Perangkat ini merupakan baterai yang memiliki sistem manajemen sendiri. Didalamnya terdapat rangkaian elektronik dan sensor yang mendeteksi tingkat tegangan, arus serta parameter lain seperti keadaan kesehatan dan kemudian dikomunikasikan dengan perangkat lain [16]. Baterai jenis ini seringkali digunakan pada perangkat elektronik *portable* seperti laptop, *handphone* dan lain sebagainya.

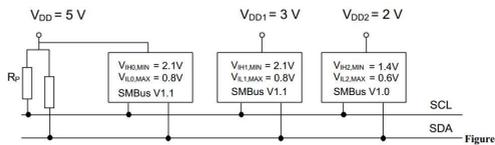
*Smart battery* terdiri dari kumpulan sel yang mampu menyediakan daya, rangkaian elektronik yang disertakan didalamnya harus dapat memantau lingkungan tertentu untuk menghitung nilai data yang sesuai [17]. Baterai ini berkomunikasi dengan perangkat lain (seperti *host* dan *smart battery charger*) dengan dua cara, yaitu:

- Menggunakan SMBus *clock* dan *data* yang merupakan saluran komunikasi utama *smart battery* dan perangkat SMBus lainnya.
- Menggunakan mekanisme pemberian sinyal sekunder atau *safety signal*, yang sebelumnya digambarkan sebagai pin “T” pada konektor baterai.

## 2.7 System Management Bus (SMBus)

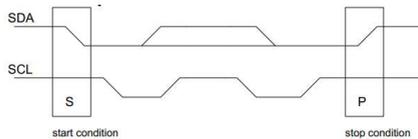
Untuk melakukan komunikasi antara *master* dan *slave* digunakan protokol SMBus. Protokol tersebut merupakan antarmuka dua kawat melalui sistem sederhana dan manajemen daya yang berhubungan dengan *chip* untuk berkomunikasi dengan seluruh sistem, Operasinya didasarkan pada prinsip pengoperasian *Inter Integrated Circuit (I2C)* [18]. SMBus menyediakan bus kontrol sistem untuk mengirim pesan ke dan dari perangkat dengan menggunakan jalur kontrol individual, sehingga dapat membantu mengurangi jumlah pin yang digunakan [19]. Dengan menggunakan sistem ini, perangkat dapat menyediakan informasi mengenai pembuatan perangkat, memberitahu model sistem/*part number*, menyimpan suatu kondisi untuk menghentikan kejadian, mengirimkan laporan dari berbagai macam jenis kesalahan dan menerima parameter kontrol.

Pada sistem ini, terdapat tiga jenis perangkat yaitu *host*, *master* dan *slave*. Perangkat *host* merupakan *master* khusus yang menyediakan antarmuka dengan CPU sistem. *Master* adalah perangkat yang mengeluarkan perintah, menghasilkan *clock*, dan mengakhiri komunikasi. Sedangkan *slave* adalah perangkat yang menerima dan menanggapi perintah dari *master* [19]. Untuk penyambungan pin *Serial Data* (SDA) dan *Serial Clock* (SCL) ditambahkan resistor *pull up*, seperti yang terlihat pada Gambar 2.4, fungsinya adalah untuk membuat nilai *floating* (berada diantara 1 dan 0) menjadi 1.



**Gambar 2. 4** Topologi SMBus

Seperti I2C, dua bus didefinisikan untuk mengirim *start* dan *stop* kondisi. Kondisi *start* terjadi pada saat SDA mengalami transisi dari *high* ke *low* dan SCL dalam kondisi *high*. Dan kondisi *stop* terjadi saat SDA mengalami transisi dari *low* ke *high* dan SCL dalam kondisi *high*, seperti yang telah ditunjukkan pada Gambar 2.5.



**Gambar 2. 5** Kondisi *Start* dan *Stop*

Secara umum I2C dan SMBus sama, namun ada sedikit perbedaan diantara keduanya. Perbedaan tersebut dapat dilihat pada Tabel 2.2 dan 2.3 [20].

**Tabel 2. 2** Perbedaan *Clock* I2C dan SMBus

	I2C	SMBus
<i>Minimum</i>	Tidak ada	10 KHz
<i>Maximum</i>	100 KHz (Mode standart)	100 KHz

	I2C	SMBus
	400 KHz (Mode Cepat) 2 MHz (Mode Kecepatan Tinggi)	
Timeout	Tidak ada	35 ms

**Tabel 2. 3** Perbedaan Karakteristik Elektrik I2C dan SMBus

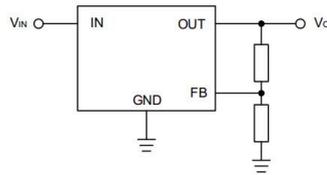
	I2C	SMBus
$V_{High}$	Fixed Voltage : 3.0 – VDD + 0.5V VDD Relative : 0.7 – VDD + 0.5V	2.1 - VDD
$V_{Low}$	Fixed Voltage : -0.5 – 1.5V VDD Relative : -0.5 – 0.3VDD	Hingga 0.8V
Max Current	3mA	350 $\mu$ A

## 2.8 Voltage Regulator

Setiap rangkaian elektronik dirancang untuk beroperasi dari beberapa tegangan *supply*, yang biasanya diasumsikan konstan [21]. *Voltage regulator* merupakan sebuah komponen yang mampu menyediakan tegangan keluaran konstan terlepas dari perubahan beban atau tegangan masukan. Komponen ini hadir untuk memberikan solusi terhadap masalah tersebut, dimana keluaran tegangan yang dihasilkan tetap. Berdasarkan metode konversinya, rangkaian ini dibagi menjadi dua jenis yaitu *linier* dan *switching* [22]. Sedangkan berdasarkan nilai keluarannya terdapat dua jenis diantaranya adalah *fixed* (bernilai tetap) dan *adjustable* (nilai dapat diubah-ubah).

*Linier regulator* pada dasarnya terdiri dari masukan, keluaran dan *ground*. Untuk tipe *adjustable* terdapat pin *feedback* yang digunakan untuk mengembalikan tegangan, [18] seperti yang terlihat pada Gambar 2.6.

Dalam penggunaan *regulator* jenis ini terdapat beberapa keuntungan yaitu konfigurasi rangkaian sederhana, komponen yang digunakan sedikit, dan gangguan rendah. Selain itu juga terdapat kekurangan diantaranya efisiensi relatif buruk, panas yang dihasilkan cukup besar dan hanya terdapat operasi *step down*.



**Gambar 2. 6** Konfigurasi Dasar *Linier Voltage Regulator* dengan *Feedback*

*Regulator* jenis ini menggunakan elemen *switching* untuk mengubah suplai daya yang masuk menjadi tegangan pulsa, dan kemudian dihaluskan menggunakan kapasitor, induktor dan komponen lainnya [19]. Pada rangkaian ini memiliki efisiensi yang lebih tinggi dari pada *linear regulator*.

Terdapat 4 jenis *regulator* ini yang umum digunakan, diantaranya adalah *buck*, *boost*, *buck-boost*, dan *flyback* [23]. *Buck* digunakan untuk menurunkan tegangan dengan polaritas yang sama. *Boost* digunakan untuk menaikkan tegangan dengan polaritas yang sama. *Buck Boost (Inverting)* menghasilkan tegangan dengan polaritas terbalik dan nilai yang didapatkan bisa lebih besar maupun lebih kecil dari tegangan masukan. *Flyback* dapat digunakan untuk menghasilkan satu atau lebih keluaran, namun beberapa diantaranya memiliki kemungkinan polaritas yang berbalik dengan masukan.

Sama dengan jenis *linier*, *switching regulator* juga memiliki keuntungan dalam penggunaan yaitu efisiensi tinggi, tidak mudah panas, memungkinkan operasi *buck*, *boost* dan *inverter*. Disisi lain, juga tersedia kekurangannya yaitu komponen yang digunakan lebih banyak, desain rumit, dan gangguan tinggi.

## 2.9 Efisiensi Daya

Dalam Kamus Besar Bahasa Indonesia (KBBI) menyebutkan bahwa, dalam bidang energi efisiensi merupakan bagian energi yang disalurkan menjadi proses yang berguna. Tidak semua daya masukan ( $P_{\text{masukan}}$ ) yang diberikan pada suatu sistem diubah menjadi daya

keluaran ( $P_{\text{keluaran}}$ ). Perbedaan antara  $P_{\text{masuk}}$  dan  $P_{\text{keluaran}}$  karena dalam suatu sistem pasti memiliki disipasi (kehilangan) daya. Untuk menghitung besar nilai efisiensi dapat digunakan rumus pada Persamaan 2.1. Berdasarkan rumus tersebut efisiensi daya dapat diartikan sebagai perbandingan antara daya keluaran dengan daya masukan.

$$Efisiensi = \frac{P_{out}}{P_{in}} \times 100\% \quad (2.1)$$

## 2.10 Inrush Current

Dalam perancangan *power distribution* rawan terjadi *inrush current*. Kejadian tersebut merupakan lonjakan arus yang ditarik oleh catu daya dihidupkan [24]. Hal ini sering terjadi pada beban induktif seperti trafo, induktor dan motor listrik. Arus awal ini dapat melonjak sedikit lebih tinggi dari arus operasional normal (*steady state*) [25]. Efek lonjakan arus ini dapat merusak komponen elektronika yang digunakan pada rangkaian [26].

*Inrush current* dapat dikurangi dengan cara meningkatkan waktu naik tegangan pada kapasitansi beban dan memperlambat laju muatan pada kapasitor. Terdapat tiga solusi berbeda yang dapat digunakan untuk menyelesaikan permasalahan ini [27], yaitu:

- *Soft start*, pada *voltage regulator*, *DC/DC converter*, dan *Low Dropout Regulator (LDO)*, terdapat fungsi *soft start*. Dengan fasilitas tersebut, *rise time* dapat ditingkatkan dimana lonjakan arus dapat berkurang.
- *Discrete implementation*, pemindahan daya dengan kenaikan waktu yang terkontrol dapat dicapai dengan menggunakan sirkuit diskrit dan dapat dilakukan dengan berbagai macam cara.
- *Integrated load switch*, sistem ini dapat digunakan sebagai pengganti sirkuit diskrit.

Dari ketiga solusi tersebut, yang memiliki integritas tinggi adalah *discrete implementation*, dibandingkan dengan *integrated load switch* yang membutuhkan komponen lebih banyak. Namun, solusi yang memiliki integritas paling tinggi adalah *voltage regulator* dengan *soft start* yang sudah terpasang [25].

## 2.11 Filter

Dalam catu daya, kapasitor digunakan untuk menghaluskan (*filter*) keluaran DC yang memiliki gangguan. Namun, juga dapat ditambahkan induktor atau resistor pada rangkaian [28]. Pada pembuatan *voltage regulator* selalu ada dua kapasitor yang berada pada masukan ( $C_{in}$ ) dan keluaran ( $C_{out}$ ). Masing-masing memiliki fungsi tersendiri,  $C_{in}$  difungsikan untuk menekan gangguan yang diterima dari sumber sedangkan  $C_{out}$  berfungsi untuk menghaluskan keluaran sebelum masuk ke beban [29]. Pada LC *filter* berperan untuk mempertahankan tegangan keluaran tetap konstan dan membatasi lonjakan tegangan. Secara umum, untuk menghitung kebutuhan kapasitor keluaran dapat digunakan persamaan pada 2.2. Dimana nilai  $V_{rms}$  dapat diukur menggunakan *multimeter*, sedangkan untuk frekuensi dapat digunakan osiloskop.

$$C = \frac{1}{V_{rms} \times 4f} \quad (2.2)$$

Pada kapasitor terdapat dua unsur parasitik yaitu *Equivalent Series Resistance* (ESR) dan *Equivalent Series Inductance* (ESL). ESR adalah hambatan secara seri terhadap pelat kapasitor. Resistansi tersebut memiliki pengaruh dalam pembuatan rangkaian *voltage regulator*. Nilai tersebut sensitif terhadap perubahan frekuensi, umumnya untuk kebanyakan kapasitor nilai ESR akan meningkat seiring dengan naiknya frekuensi operasi [30].

Untuk mendapatkan nilai gangguan yang kecil, diperlukan kapasitor keluaran dengan ESR yang kecil. Dan lebih baik digunakan induktor yang bernilai antara  $1\mu\text{H}$  hingga  $5\mu\text{H}$  dengan resistansi yang rendah, gunanya adalah untuk mempertahankan regulasi beban yang baik [31].

## 2.12 State of Charge (SoC)

Dalam baterai terdapat sebuah parameter yang disebut SoC. Nilai tersebut didefinisikan sebagai rasio total kapasitas energi yang dapat digunakan dari sebuah baterai dengan kapasitas total. Satuan yang

digunakan adalah persen. Cara mengukur nilai tersebut dapat digunakan tiga cara [32], yaitu:

- Secara langsung, dapat dilakukan jika baterai bisa melakukan *discharge* pada nilai yang konstan.
- *Specific gravity*, cara ini bergantung pada perubahan pengukuran dari berat bahan kimia aktif.
- Perkiraan SoC berdasarkan tegangan, dengan mengukur tegangan sel baterai sebagai dasar untuk perhitungan. Cara ini dipengaruhi oleh bahan sel dan suhu, sehingga hasil yang didapatkan bisa kurang akurat.

Kebalikan dari nilai ini adalah *depth of discharge* (DoD). Nilai tersebut menunjukkan prosentasi kapasitas yang telah digunakan [33].

### 2.13 Charging System

Setiap perangkat elektronik umumnya dilengkapi dengan *charger* yang digunakan untuk mengisi ulang baterai. Sistem *charging* merupakan sistem yang memproduksi tenaga listrik untuk mengisi baterai. Sistem ini digunakan untuk mengembalikan kondisi baterai agar selalu siap digunakan. Pada dasarnya sistem ini memiliki tiga fungsi utama [34]. Fungsi tersebut adalah mengisi baterai (*charging*), mengoptimalkan tingkat pengisian (*stabilizing*), dan mengetahui kapan harus berhenti (*terminating*)

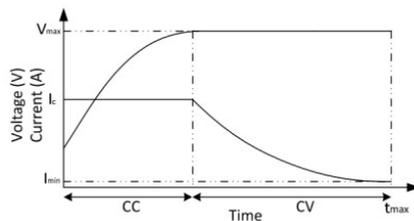
Sistem ini memiliki beberapa jenis, berdasarkan kecepatan pengisiannya dapat dibedakan menjadi 3 yaitu:

- *Slow charge*, sistem ini membutuhkan waktu 14 – 16 jam untuk mengisi baterai hingga penuh. Tingkat arus yang digunakan sebesar  $0.1C$  (0.1 dari kapasitas baterai). Keuntungan menggunakan sistem ini adalah baterai tidak panas.
- *Quick charge*, untuk mengisi baterai hingga penuh, diperlukan waktu 3 – 6 jam. Tingkat arus yang digunakan sebesar  $0.3C$  (0.3 dari kapasitas baterai). Pada sistem ini rawan terjadi panas pada baterai, sehingga suhu pada baterai perlu untuk diawasi.

- *Fast charge*, untuk mengisi baterai hingga penuh, diperlukan waktu kurang dari 1 jam. Tingkat arus yang digunakan sebesar 1.0C (setara dengan kapasitas baterai). Pada sistem ini juga rawan terjadi panas pada baterai, sehingga suhu pada baterai perlu untuk diawasi

## 2.14 Constant Current Constant Voltage (CCCV)

Metode ini merupakan cara yang paling banyak digunakan dilapangan, karena sederhana dan mudah untuk diimplementasikan. Proses ini merupakan kombinasi antara mode *constant current* dan *constant voltage charging* [35]. Oleh karena itu, sistem ini menjamin proses *fast charging* ketika baterai dapat menerima tingkat arus tinggi dan polarisasi baterai terbatas pada akhir *charging*, sehingga mampu mencegah kerusakan sel [36].



**Gambar 2. 7** Grafik Arus dan Tegangan dari CCCV Charging

*Constant current* merupakan kemampuan memvariasikan tegangan pada rangkaian elektronika untuk mempertahankan arus agar tetap konstan [37]. Mode ini bekerja dengan menggunakan resistansi internal yang sangat tinggi dan resistansi beban jauh lebih rendah [38].

*Constant voltage* merupakan kebalikannya, yaitu kemampuan untuk memvariasikan arus keluaran guna untuk mempertahankan tegangan pada nilai tetap [39]. Sumber yang biasa digunakan untuk mode ini adalah baterai dan *power supply* [40].

Dari Gambar 2.5 dapat dilihat bahwa proses pengisian dimulai dengan menggunakan mode *constant current* hingga tegangan baterai mencapai nilai maksimum. Kemudian akan beralih pada mode *constant*

*voltage*, pada tahap tersebut baterai sudah terisi 85% dari kapasitas penuhnya. Selain itu, dalam mode ini pengisian arus selalu dimonitor tujuannya adalah untuk menentukan kapan proses *charging* dapat dihentikan. Biasanya baterai dianggap penuh ketika pengisian arus turun dibawah  $0.1C$  [41].

## **BAB III**

### **PERANCANGAN *POWER DISTRIBUTION***

Pada Bab ini dijelaskan mengenai proses perancangan dan pembuatan sistem *power distribution* pada OCU. Terdapat beberapa poin yang dijelaskan yaitu penjelasan mengenai target-target yang ingin didapatkan dari perancangan sistem ini, blok fungsional sistem, perancangan sistem *powering*, sistem indikator, dan sistem *charging*.

#### **3.1 *Design Requirement***

Sesuai dengan yang telah dijelaskan pada Bab II bagian subbab 2.3, bahwa penulis mendapat tugas untuk mengerjakan bagaian *power distribution*. Untuk itu, pada Bab ini dibahas mengenai proses perancangan dan pembuatan sistem *power distribution* pada OCU. Secara garis besar terdapat tiga bagian yang dikerjakan, yaitu sistem *powering*, sistem indikator dan pembuatan konsep sistem *charging*. Masing-masing sistem tersebut dirancang sesuai dengan kebutuhan.

Sistem *powering* digunakan untuk mengelola daya dari baterai sebelum disalurkan ke komponen yang digunakana. Pada sistem ini akan dihasilkan daya dengan tegangan yang sesuai dengan kebutuhan tiap-tiap komponen. Target yang ingin dicapai dalam perancangan sistem ini adalah sebagai berikut:

- Mampu menghasilkan tegangan sebesar 5VDC dan 12VDC
- Komponen penyusun OCU dapat diaktifkan menggunakan *supply* yang dihasilkan dari sistem ini

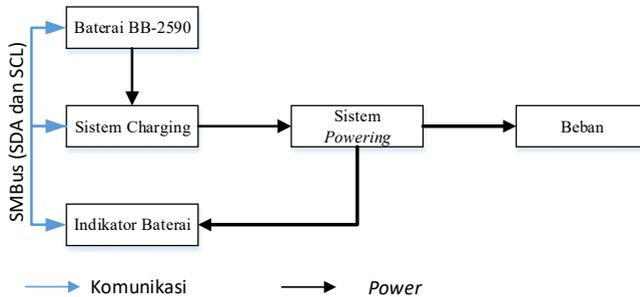
Dalam perancangan sistem indikator, terdapat dua indikator yang ditampilkan yaitu kapasitas baterai dan *power*. Pada indikator baterai digunakan nilai SoC, dengan ketentuan bahwa nilai tersebut memiliki selisih yang kecil dengan perhitungan manual kapasitas.

Pada pembuatan konsep sistem *charging*, terdapat parameter-parameter yang digunakan untuk mempengaruhi proses pengisian.

Target dalam perancangan ini yaitu diketahui rentang nilai dari parameter-parameter yang diizinkan dalam proses pengisian.

### 3.2 Blok Fungsional Sistem

Pada sistem ini terdapat tiga bagian utama yaitu sistem *powering* yang berfungsi untuk melakukan pembagian *supply* dari baterai ke beban. Selanjutnya adalah sistem indikator digunakan untuk menampilkan kondisi-kondisi yang berhubungan dengan sumber. Terakhir adalah sistem *charging* untuk melakukan pengisian pada baterai.



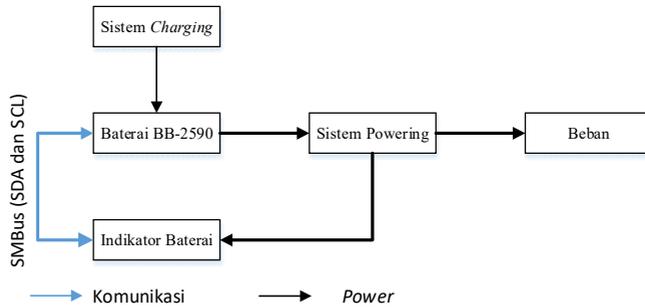
**Gambar 3.1** Konfigurasi Sistem *Power Distribution* OCU Menggunakan Konsep Sistem *Charging*

Sumber yang digunakan adalah baterai jenis lithium ion dari Ben Tronics, dengan tipe BB 2590. Tipe tersebut telah dilengkapi *smart battery system*, dimana didalamnya terdapat BMS dan SMBus V1.1 [1]. Didalam satu baterai terdapat dua bagian dimana masing-masing kapasitasnya adalah 7.5 Ah dengan nilai tegangan sebesar 15VDC. Pada sistem ini kedua bagian tersebut disusun secara paralel, sehingga didapat kapasitas total 15 Ah.

Sebelum sumber dari baterai masuk ke beban, dilakukan pengolahan terlebih dahulu melalui sistem *powering*. Pada sistem tersebut akan dihasilkan keluaran tegangan yang sesuai dengan kebutuhan beban.

Dapat dilihat pada Gambar 3.1, merupakan diagram blok fungsional sistem menggunakan konsep sistem *charging* yang telah dibuat.

Didalamnya terdapat SMBus yang menghubungkan antara baterai, indikator baterai dan sistem *charging*, fungsinya adalah untuk melakukan komunikasi, dimana mikrokontroler berperan sebagai *master* sedangkan yang lain sebagai *slave*. Dalam sistem ini juga dapat dilakukan pembuatan indikator baterai dengan menggunakan mikrokontroler.



**Gambar 3.2** Konfigurasi Sistem *Power Distribution OCU*

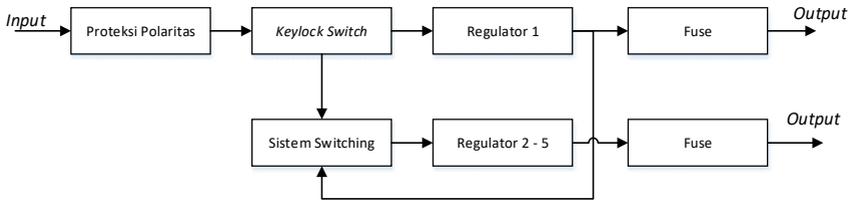
Untuk diagram blok sistem tanpa menggunakan konsep sistem *charging* dapat dilihat pada Gambar 3.2. Pada sistem ini, digunakan *charger* yang telah tersedia dari perusahaan pembuat baterai. Proses pengisian dapat dilakukan dengan melepas baterai dari OCU terlebih dahulu.

### 3.3 Perancangan Sistem Powering

Pada tahap ini, dilakukan pengolahan sumber dari baterai ke beban. Pada Gambar 3.3 dapat dilihat, bahwa dalam sistem ini terdapat 3 bagian yaitu sistem proteksi, *voltage regulator* dan *switching*. Selain itu, terdapat *keylock switch* yang digunakan untuk memutus dan menghubungkan baterai ke sistem.

OCU aktif dapat dilihat *flowchart* pada Gambar 3.4, dimana sebelum menyalakannya terdapat dua *switch* yang harus dioperasikan secara berurutan. Pertama yang dilakukan adalah mengubah posisi *keylock switch* pada posisi *on*, fungsinya adalah untuk menghubungkan

sumber dari baterai ke sistem. Pada tahap ini, indikator baterai akan menyala dan *voltage regulator* 1 aktif untuk memberi *supply* pada *main controller*. Perangkat tersebut merupakan mini PC sehingga didalamnya telah dilengkapi dengan *operating system*. Untuk mengaktifkannya digunakan *push button* yang telah disediakan.



**Gambar 3.3** Diagram Blok Sistem Powering

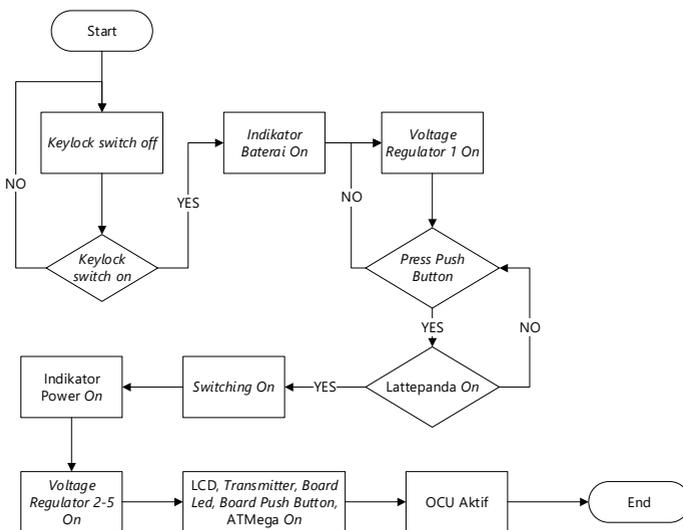
Langkah selanjutnya adalah menekan *push button power*, fungsinya adalah untuk mengaktifkan *main controller*. Setelah itu, *switching* akan aktif dan menyalakan led indikator power serta *voltage regulator* 4 hingga 5. Dalam kondisi ini, semua beban OCU telah mendapat *supply*, sehingga alat dapat dioperasikan.

Untuk mematikan OCU, dapat digunakan fungsi *shut down* dari *operating system*. Namun jika terjadi masalah, dimana alat tidak berjalan dapat dilakukan *force shut down*, caranya adalah dengan menekan *push button power* dan ditahan beberapa saat. Akan tetapi, untuk cara tersebut hanya disarankan jika alat dalam keadaan macet dan tidak dapat dioperasikan sama sekali.

### 3.3.1 Sistem Proteksi

Proteksi yang diterapkan pada sistem ini digunakan untuk melindungi rangkaian dari berbagai macam kemungkinan yang memiliki hubungan dengan sumber sehingga dapat merusak beban atau sistem. Terdapat dua jenis yang digunakan, yaitu proteksi polaritas dan arus berlebih (*overcurrent*). Masing-masing diletakkan pada awal dan akhir sistem. Pada awal perancangan digunakan p-MOSFET dan dioda untuk proteksi polaritas. Namun, dirasa dari segi harga terlalu mahal

akhirnya diganti hanya dengan menggunakan dioda. Jenis yang dipakai adalah dioda *scottkky*, karena memiliki nilai *drop* tegangan lebih kecil dibanding dengan jenis lainnya.



**Gambar 3.4** Flowchart Aktivasi OCU

Pembuatan proteksi polaritas bertujuan agar tidak terjadi kerusakan pada sistem ketika pemasangan sumber terbalik. Hal ini akan sangat bermanfaat pada saat melakukan uji coba dan *maintenance* (*socket* belum dipasang). Untuk proteksi arus berlebih digunakan *fuse* yang ditempatkan pada keluaran *voltage regulator*. Untuk menentukan nilainya, terlebih dahulu dicek arus yang dibutuhkan dengan cara mengukur secara langsung. Dari hasil tersebut dapat ditentukan ukuran *fuse* yang akan digunakan, jika nilainya tidak ada maka digunakan satu tingkat yang berada di atasnya. Dalam hal ini, digunakan salah satu *datasheet* dari Littlefuse. Besar yang digunakan disesuaikan dari kebutuhan beban, ukuran tersebut dapat dilihat pada tabel 3.1.

**Tabel 3.1** Estimasi *Fuse* pada Masing-masing Beban

No.	Beban	Fuse (A)
1	Lattepanda	2
2	ATMEGA	0.032
3	Board Push Button	0.032
4	Board Led	0.8
5	LCD	0.5
6	Transmitter	0.25
7	Led Emergency	0.032
8	Indikator Baterai	0.032

### 3.3.2 Voltage Regulator

Fungsi dari rangkaian ini adalah untuk mengubah nilai tegangan dari baterai sebesar 15VDC dan dihasilkan keluaran yang sesuai dengan kebutuhan beban. Pada perancangan sistem ini, hanya digunakan satu jenis IC keluaran yaitu *fixed voltage regulator*. Jenis komponen yang digunakan adalah *switching regulator*. Dipilih jenis tersebut karena secara teori memiliki efisiensi yang lebih besar dari pada *linier regulator*.

Kebutuhan *supply* pada beban OCU memiliki ukuran yang bermacam-macam. Untuk itu dibuat pengelompokan berdasarkan besar tegangan dan arus yang diperlukan. Masing-masing kelompok digunakan satu *voltage regulator* dan bisa terdapat lebih dari satu beban. Pada tabel 3.2 dapat dilihat hasil pengelompokan, dimana didapat 6 kelompok.

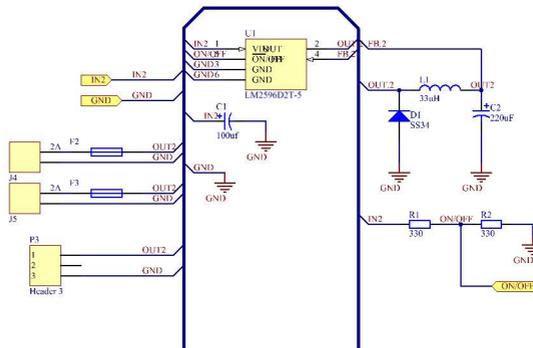
Berdasarkan hasil pengelompokan terdapat satu kelompok beban yang membutuhkan arus dengan ukuran mendekati 3A, sehingga dipilih IC dengan spesifikasi dapat menghasilkan keluaran yang mampu mencapai nilai tersebut. Melalui pertimbangan tersebut akhirnya didapatkan IC dengan tipe LM2576. Komponen ini memiliki kesesuaian spesifikasi dengan yang dibutuhkan. Namun, terdapat kendala dimana untuk mencari bentuk smd dari komponen tersebut dipasaran yang ada diIndonesia terlalu sulit akhirnya diganti dengan IC LM2596. Dipilih komponen ini karena memiliki spesifikasi dan bentuk rangkaian yang

sama dengan komponen sebelumnya [2][3]. Karena alasan tersebut, tidak perlu dilakukan perubahan pada desain *board* yang telah dibuat.

**Tabel 3.2** Pengelompokan Regulator dan Beban

No.	Regulator Ke-	Keluaran Tegangan	Beban
1	1	5V	Lattepada
2	2	5V	ATMEGA dan Board Push Button
3	3	12V	Board LED
4	4	12V	LCD
5	5	12V	Transmitter, Led Emergency
6	6	5V	Indikator baterai

Pada IC ini terdapat pin *on/off*, fungsinya adalah untuk mengaktifkan dan mematikan *regulator*. Rangkaian akan aktif ketika tegangan pada pin tersebut bernilai 0, begitu juga sebaliknya jika melebihi nilai tersebut rangkaian akan mati, sehingga keluaran yang dihasilkan sama dengan nol.



**Gambar 3.5** Rangkaian *Voltage Regulator* LM2576/LM2596

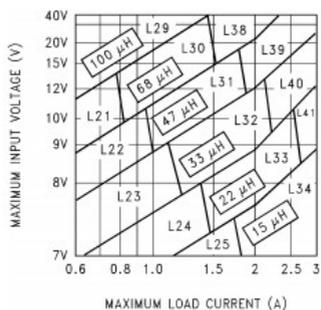
Fasilitas tersebut dimanfaatkan pada sistem ini. Pada *board power*, *regulator* 2 hingga 5 pin *on/off* tidak langsung dihubungkan dengan *ground*. Tujuannya adalah untuk mengatur beban agar aktif ketika *main controller* dinyalakan. Saat *keylock switch* diputar pada posisi *on*, yang aktif hanya *regulator* 1 dan 6. Dengan konsep tersebut diharapkan dapat mengurangi penggunaan daya pada alat ini, dimana saat *push button*

power belum ditekan, maka sumber hanya memberikan *supply* pada indikator baterai dan Lattepanda.

Untuk *voltage regulator* 1 digunakan produk dari Ubec, karena Lattepanda membutuhkan *supply* dengan gangguan kecil dan keluaran arus konstan. Digunakan jenis tersebut karena IC LM2596, tidak mampu menghasilkan arus konstan.

### 3.3.3 Pemilihan Kapasitor dan Induktor

Dalam pemilihan komponen ini, dilakukan berdasarkan rekomendasi dari *datasheet*. Untuk menentukan besar induktor dan kapasitor keluaran *voltage regulator* 5VDC dan 12VDC, dapat disesuaikan dengan tabel rekomendasi dari *datasheet*. Variable yang digunakan untuk mencari adalah tegangan masukan maksimum dan arus beban maksimum. Karena masing-masing didesain dengan keluaran arus sebesar 3A dan masukan 15VDC, maka ukuran komponen yang digunakan adalah  $33\mu\text{H}$  untuk keluaran 5VDC dan  $22\mu\text{H}$  untuk keluaran 12VDC.

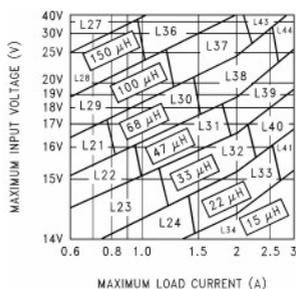


**Gambar 3.6** Grafik Menentukan Induktor Pada Keluaran 5VDC

Namun, beda jenis kapasitor yang digunakan, berbeda juga ukuran yang dipakai. Untuk menghasilkan keluaran arus yang konstan disarankan untuk menggunakan induktor dengan kapasitas arus dengan minimal setengah dari arus beban.

Pemilihan kapasitor digunakan tabel yang telah tersedia dari *datasheet*, dimana beda jenis bahan yang digunakan, berbeda pula

ukuran yang dipakai. Untuk memilih parameter yang digunakan hampir sama dengan pemilihan induktor, yaitu tegangan keluaran, tegangan masukan maksimum, dan arus beban. Dari Gambar 3.6 dan 3.7 dapat ditentukan besar induktor yang digunakan. Ketentuan yang digunakan sama dengan sebelumnya, sehingga didapat kebutuhan kapasitor pada keluaran 5VDC sebesar  $330\mu\text{F}/35\text{V}$  dengan bahan elektrolit serta  $220\mu\text{F}/10\text{V}$  atau  $330\mu\text{F}/10\text{V}$  untuk bahan tantalum. Sedangkan pada rangkaian keluaran 12VDC digunakan kapasitor sebesar  $470\mu\text{F}/25\text{V}$  dengan bahan elektrolit serta  $100\mu\text{F}/16\text{V}$  atau  $180\mu\text{F}/16\text{V}$  untuk bahan tantalum.



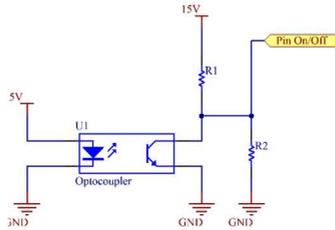
**Gambar 3.7** Grafik Menentukan Induktor Pada Keluaran 12VDC

### 3.3.4 Sistem *Switching*

Nyala dan mati OCU dibuat bergantung pada Lattepada, karena beban tersebut berperan sebagai inti dari sistem ini. Tujuannya adalah untuk meminimalisir penggunaan daya, ketika *keylock switch* dialihkan pada posisi *on*. Untuk itu, pada saat menekan *push button power*, *main controller* akan aktif dan memberi sinyal pada sistem *switching* untuk mengaktifkan pin *on/off regulator* 2 hingga 5. Melalui proses tersebut, beban yang lain akan ikut aktif. Sinyal yang dikirimkan diambil dari pin *digital output*.

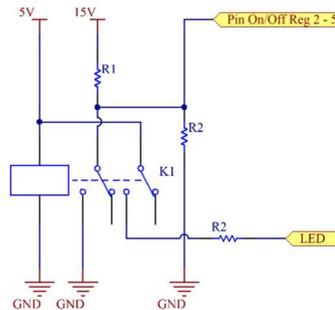
Terjadi dua kali perancangan pada sistem ini. Yang pertama digunakan optocoupler sebagai *switching*-nya. Pada konsep ini, tiap rangkaian menggunakan satu optocoupler. Rangkaian yang digunakan

dapat dilihat pada Gambar 3.7. Nilai resistor yang digunakan pada R1 dan R2 dibuat sama, tujuannya adalah untuk membagi tegangan.



**Gambar 3.8** Rangkaian Switching On/Off Voltage Regulator Menggunakan Optocoupler

Saat dilakukan uji coba *voltage regulator* secara fungsional, sistem dapat bekerja dengan baik. Namun pada integrasi sistem terjadi masalah, dimana tegangan pada pin *on/off* tidak bisa bernilai nol. Hal tersebut disebabkan karena *optocoupler* tidak dapat melakukan *switching*. Oleh karena itu, komponen tersebut diganti dengan menggunakan relay. Untuk konsep ini, semua pin *on/off* digabungkan menjadi satu, rangkaian dapat dilihat pada gambar 3.8.



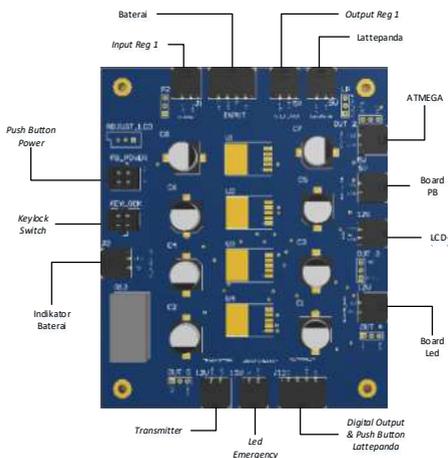
**Gambar 3.9** Switching On/Off dan Indikator Power Menggunakan Relay

### 3.3.5 Konfigurasi *Board Power* Ke Beban

*Regulator* 1 dipasang secara terpisah dengan *board power*, namun masukan yang diterima tetap diambil dari *board* tersebut. Dibuat

demikian karena pada *board* regulator 1 telah tersedia kabel untuk masukan dan keluaran. Disisi lain, hal tersebut juga bisa mempermudah pemasangan. Tegangan masukan yang digunakan adalah 15VDC, sehingga dalam pemasangan didapat dari *connector* yang tertulis *input reg 1*. Keluaran yang dihasilkan disambungkan lagi dengan *board power*, didalamnya akan dibagi menjadi dua yang digunakan untuk memberi *supply* pada Lattepada dan relay.

Keterangan baterai pada Gambar 3.9, disambungkan pada sumber baterai dimana terdapat 4 koneksi yaitu 2 VCC dan 2 GND. *Digital output* dan *push button* diambil dari *main controller*, konfigurasi tersebut digunakan sebagai *push button* dan indikator *power* untuk OCU. Sedangkan untuk keterangan Lattepada, ATMEGA, *board pb*, LCD, *board led*, *led emergency*, dan *transmitter* merupakan *connector* yang terhubung pada beban sesuai dengan nama yang tertera.



**Gambar 3.10** Konfigurasi *Board Power*

Indikator baterai pada *board* ini digunakan untuk memberi *supply* sebesar 15VDC pada *board* indikator baterai. Untuk *keylock switch* dan *push button power* digunakan untuk menarik kabel dari *board* ke panel dan disambungkan pada masing-masing komponen sesuai dengan keterangan.

### 3.4 Sistem Indikator

Perancangan indikator bertujuan agar pengguna bisa lebih mudah mengetahui kondisi sumber yang digunakan. Dalam sistem ini, yang perlu ditampilkan adalah power OCU digunakan untuk mengetahui bahwa alat aktif atau tidak, dan level baterai digunakan untuk menunjukkan sisa kapasitas baterai. Untuk menampilkan kedua indikator tersebut digunakan led yang dipasang pada panel. Untuk indikator power digunakan led yang ada pada *push button power*. Sedangkan untuk level baterai digunakan 4 led yang telah dilengkapi *panel mount* dan disusun secara *vertical* pada *panel*.

#### 3.4.1 Indikator Power

Indikator ini berfungsi untuk memberi tahu pada pengguna bahwa OCU dalam keadaan menyala atau mati. Led akan menyala ketika *push button power* ditekan dan akan mati ketika OCU dimatikan.

Rangkaian yang digunakan digabungkan menjadi satu dengan sistem *switching* yang terdapat pada sistem *powering*. Ketika *push button power* ditekan, relay akan melakukan *switching* dan menyalakan led yang terdapat pada *push button* tersebut. Pada sistem ini komponen yang digunakan adalah relay 5VDC. Dimana kontaknya ada dua yang satu digunakan untuk indikator power dan satu lagi untuk mengendalikan pin *on/off* regulator 2 sampai 5, seperti yang terlihat pada gambar 3.6.

#### 3.4.2 Indikator Baterai

Fungsi dari indikator ini adalah untuk menampilkan kapasitas yang tersisa dari baterai, sehingga pengguna dapat mengetahui kapan harus melakukan *charging*. Serta mampu memperkirakan berapa lama alat bisa digunakan. Dalam *board* ini terdapat dua bagian utama yaitu sistem *host* dan *voltage regulator*.

Indikator ditampilkan dengan menggunakan 4 led yang disusun secara *vertical* pada panel. Terdapat 3 warna yang digunakan yaitu

merah, kuning dan 2 hijau. Masing-masing menunjukkan kapasitas baterai yang tersedia. Dimana merah menunjukkan rating 0 – 25%, kuning 26 – 50%, dan hijau 51 – 100%. Komponen yang digunakan telah dilengkapi dengan *panel mount* sehingga dapat mempermudah dalam pemasangan. Pada sistem ini terdapat dua bagian yaitu *host* dan *switching*. Digunakan mikrokontroler sebagai *host*, fungsinya adalah untuk mengakses data yang terdapat pada baterai. Dipilih *multiplexer* sebagai *switching*-nya, dimana digunakan untuk melakukan peralihan pin I2C baterai. Karena pada satu baterai tersedia dua bagian dengan *port* terpisah dan alamat yang digunakan sama.

### 3.4.3 Pembuatan Host

Fungsi sistem ini adalah untuk membaca dan mengolah data dari kedua baterai dan menampilkannya kedalam bentuk led. Pada perancangannya, dilakukan dua kali pembuatan mikrokontroler. Hal tersebut terjadi berdasarkan masukan yang diberikan oleh pembimbing dari perusahaan.

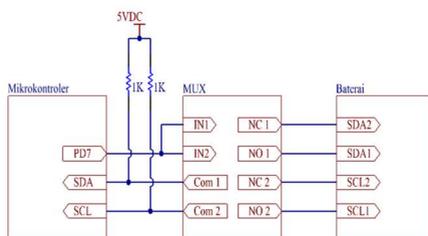
Pertama komponen yang digunakan adalah ATtiny2313. Penentuan tipe tersebut dilakukan dengan mempertimbangkan jumlah pin yang dibutuhkan sedikit. Untuk memprogram komponen tersebut diperlukan pengaturan *clock* yang cukup rumit agar komunikasi serial dapat berjalan. Selain itu, pin I2C yang tersedia berbenturan dengan ISP (*In-System Chip Programming*) dan ditakutkan program tidak bisa terdownload. Melalui pertimbangan tersebut, pembimbing dari perusahaan menyarankan untuk mengganti tipe komponen yang digunakan.

Pada pembuatan kedua digunakan ATMEGA8 untuk menggantikan sistem *host* sebelumnya. Dipilih komponen tersebut, karena lebih mudah digunakan dan ukurannya tidak terlalu besar, sehingga desain cukup tidak terlalu memakan tempat.

### 3.4.4 Konfigurasi Multiplexer dan Host

Dalam pembacaan data pada baterai, digunakan pin SDA dan SCL. Pada baterai sendiri terdapat dua bagian dimana masing-masing memiliki pin tersebut. Untuk mengakses datanya pada komunikasi ini digunakan alamat. Pada masing-masing bagian baterai memiliki alamat yang sama. Untuk itu diperlukan *switching* guna memindahkan pin komunikasi.

Sebagai solusi digunakan *multiplexer* untuk mengalihkan pin tersebut. IC yang diterapkan pada sistem ini adalah 74HC157, namun dalam realisasi terjadi masalah dimana nilai keluaran berkurang dari masukan yang diberikan. Hal tersebut diuji dengan memberikan tegangan 5VDC pada masukan dan hasil ukur pada keluaran menjadi sekitar 2VDC. Karena alasan tersebut akhirnya diputuskan untuk diganti dengan menggunakan TS5A23157. Komponen tersebut beroperasi seperti relay, dimana terdapat dua pasang masukan dan kontak yang terdiri dari *normally close* dan *normally open*. Keluaran yang dihasilkan berada pada pin *com*. Pada pin *normally open* disambungkan pada port SDA. Selanjutnya untuk pin *normally close* disambungkan pada port SCL. Pada masing-masing sambungan, baterai bagian B dihubungkan dengan keluaran MUX pertama (NO1 dan NC1), begitu juga sebaliknya untuk baterai bagian A dihubungkan dengan keluaran kedua (NO2 dan NC2).



**Gambar 3.11** Konfigurasi Multipelxer dengan Baterai dan *Host*

Pada pin *com 1* dihubungkan dengan port SDA dan *com 2* disambungkan dengan port SCL dari *host*. Masing-masing sambungan

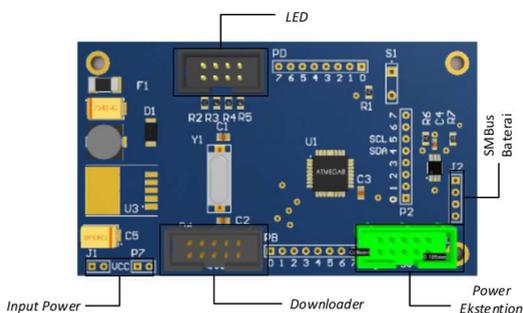
tersebut, dipasang *pull up* resistor dengan nilai 1K dan VCC sebesar 5VDC. Fungsinya adalah untuk mengubah kondisi *floating* menjadi *high*. Untuk pin IN1 dan IN2 diparalel lalu disambungkan pada PORTD.5. Pada pin tersebut, *switching* diatur untuk memindahkan pembacaan data dari baterai bagian A ke bagian B atau sebaliknya.

### 3.4.5 Konfigurasi Board Indikator Baterai

Dari desain yang telah dibuat, sistem ini memiliki koneksi dengan *board* sistem *powering* dan baterai. Dari Gambar 3.11 telah disediakan keterangan pada tiap *connector*. *Input power* disambungkan dari *board power* yang memiliki keterangan indikator, namun jika digunakan sistem *charging* dihubungkan pada konektor indikator baterai.

SMBus baterai dihubungkan langsung dengan pin SDA dan SCL baterai, dipasang sesuai dengan keterangan yang telah tertera. Dalam sistem ini telah dilengkapi resistor *pull up*, sehingga dalam pemasangan tidak perlu ditambahkan resistor. Ketika sistem *charging* diterapkan, koneksi tersebut diparalel dengan konektor SMBus yang terdapat pada *board* sistem *charging*.

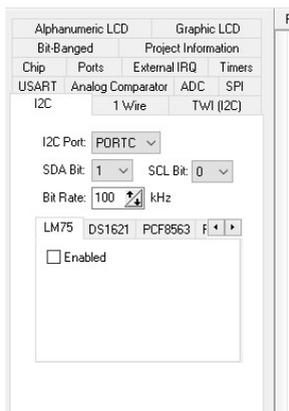
Selanjutnya adalah koneksi untuk Led dengan keterangan led pada gambar. Untuk menyalakan led digunakan pin *digital output* dari mikrokontroler. Dan terakhir adalah *downloader* yang digunakan untuk *upload* program pada IC.



Gambar 3.12 Konfigurasi Board Indikator Baterai

### 3.4.6 Pembuatan Program Indikator Baterai

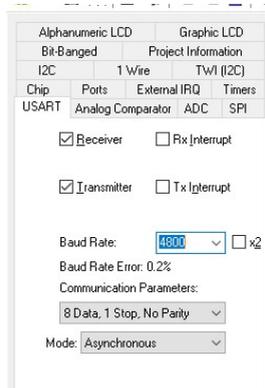
Untuk membaca data yang terdapat pada baterai digunakan *port* I2C. Program yang digunakan untuk membaca data baterai, dibuat dengan menggunakan protokol SMBus, dimana telah disediakan pada SMBus *specification revision* 1.1. Dalam pedoman tersebut telah dijelaskan sebelum melakukan pembacaan data, terlebih dahulu *host* mengirim permintaan pada *slave* dan kemudian data yang diminta akan dikirimkan per *byte*.



Gambar 3.13 Pengaturan Port I2C

Tahap awal yang dilakukan yaitu membuat project baru pada Code Vision AVR. Dalam pembuatan project tersebut, terdapat beberapa konfigurasi yang harus dilakukan. Pertama tentukan jenis *chip* yang digunakan, dimana dalam sistem ini digunakan ATmega 8 dan diikuti mengatur *clock*. Besar *clock* yang dituliskan tidak boleh melebihi besar Kristal yang digunakan. Selanjutnya, mengaktifkan pin I2C dengan memilih tab I2C, selanjutnya mengatur *port* I2C pada *port* C, seperti yang telah ditunjukkan pada Gambar 3.13. Tahap selanjutnya yaitu mengatur pin sebagai keluaran atau masukan pada tab ports. Pada pengaturan ini PORTD.0 hingga PORTD.6 dan diatur sebagai keluaran, kecuali PORTD.2 diatur sebagai masukan. Berikutnya perlu diatur USART, setelah memilih tab tersebut centang box *receiver* dan

*transmitter*. Selanjutnya, atur *baudrate* dengan nilai *error* yang paling kecil, seperti pada Gambar 3.14. Fungsi dari pengaturan tersebut adalah untuk menampilkan data pada *serial monitor*. Langkah terakhir yaitu memilih menu *generate* untuk mengubah aturan yang telah dibuat kedalam program.



**Gambar 3.14** Pengaturan USART

Urutan pembuatan program dimulai menambahkan *library delay* dan dilanjutkan dengan menginisialisasi variabel-variabel yang digunakan. Dilanjutkan dengan inialisasi kondisi *start* dan diikuti *slave address* dimana pada baterai yang digunakan menggunakan alamat *0x0B*. Selanjutnya mengirimkan *command code* data yang ingin dibaca dengan keterangan *write* (0). Setelah itu, diinisialisasi kondisi *repeated start*, kemudian dilanjutkan dengan meminta data yang telah dikirim sebelumnya menggunakan *command code* serta keterangan *read* (1). Dan yang terakhir adalah menginisialisasi kondisi *stop*.

Dalam pembuatan indikator ini digunakan nilai SoC yang telah disediakan oleh baterai. *Command code* dari data tersebut adalah *0x0E*. Konsep yang digunakan adalah membaca data pada bagian *A* terlebih dahulu kemudian pin *SDA* dan *SCL* dipindahkan ke bagian *B* menggunakan *multiplexer*. Tahap selanjutnya membaca data dari bagian tersebut. Kedua nilai yang telah didapatkan akan dijumlahkan dan dibagi dua, seperti yang ditampilkan pada Persamaan 3.1. Hasil yang didapat

akan ditampilkan pada led panel. Proses tersebut berlangsung secara berulang kali, sehingga yang ditampilkan pada led akan berubah seiring dengan perubahan nilai SoC.

$$SoC = \frac{SOC_{Section A} + SOC_{Section B}}{2} \quad (3.1)$$

### 3.5 Perancangan Sistem Charging

Dalam sistem ini terjadi perubahan dari rencana awal, dimana pada proposal dicantumkan hingga uji coba. Namun, dari hasil rapat tim yang telah disepakati, pada OCU versi ini tidak dimasukkan sistem *charging* terlebih dahulu. Alasannya adalah ingin memaksimalkan fungsi dasar terlebih dahulu. Dengan kesepakatan tersebut, pembimbing perusahaan menyarankan untuk membuat konsep dan menentukan jenis komponen yang dapat digunakan.

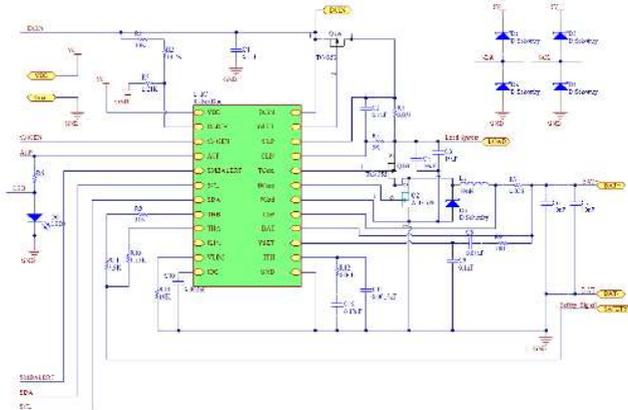
#### 3.5.1 LTC4100

Komponen yang digunakan adalah LTC4100, dimana didalamnya telah dilengkapi SMBus untuk melakukan komunikasi. Dengan IC ini proses *charging* dapat diatur sesuai dengan kebutuhan. Dalam perancangan ini, rangkaian yang digunakan sesuai dengan rekomendasi dari *datasheet*.

Pada pin  $I_{LIM}$  dan  $V_{LIM}$  digunakan untuk membatasi arus serta tegangan yang keluar, dengan cara memasang resistor antar pin tersebut dengan *ground*. Untuk rangkaian yang telah dibuat, tidak diberi resistor pada pin tersebut sehingga, maksimal keluaran yang bisa dikeluarkan adalah 4A dan 27VDC. Dibuat demikian, karena diinginkan arus yang dapat mengisi baterai mencapai nilai 4A dengan kapasitas baterai 15Ah sehingga proses pengisian bisa berlangsung kurang lebih dalam 3 jam. Untuk tegangan dapat diatur mulai dari 16 – 27VDC, karena baterai didesain dengan tegangan 15VDC. Berdasarkan pengaturan tersebut, ukuran minimal *power supply* atau *adaptor* yang dapat digunakan adalah 18VDC, 4A.

Dalam sistem ini juga dilengkapi *switching*, fungsinya adalah untuk memindahkan baterai dari bagian *A* ke bagian *B* atau sebaliknya.

Pada indikator baterai digunakan untuk mengalihkan sumber yang semula dari baterai dipindahkan langsung dari *charger*, sehingga pada proses *charging* indikator tersebut akan menyala. Serta juga untuk memutus sistem dari baterai dan mengalihkannya ke *charger*. Terdapat dua pilihan komponen yang digunakan untuk melakukan peralihan tersebut yaitu MOSFET dan relay.



**Gambar 3.15** Rangkaian *Charger* Menggunakan LTC4100

Dalam keadaan normal, untuk mengisi baterai hingga penuh dibutuhkan waktu selama kurang lebih 3 jam 20 menit. Nilai tersebut didapatkan dengan cara membagi kapasitas arus baterai dengan arus yang digunakan untuk mengisi.

### 3.5.2 Algoritma Charging

Metode yang digunakan pada sistem ini adalah *constant current constant voltage (CCCV)*. Dalam komponen yang digunakan dapat diatur nilai keluaran arus dan tegangan dengan menggunakan *command code chargingcurrent (0x14)* serta *chargingvoltage (0x15)*. Data yang diperoleh dari perintah tersebut menggunakan satuan *millivolt (mV)* atau *miliampere (mA)*. Terdapat tiga parameter yang digunakan yaitu *SoC*,

tegangan dan suhu baterai, dimana nilai tersebut dapat mempengaruhi jalannya proses *charging*.

Pada IC yang digunakan terdapat syarat-syarat kapan *charger* aktif dan mati. Untuk memulai pengisian fungsi *chargingvoltage()* dan *chargingcurrent()* tidak boleh bernilai nol. Selanjutnya, masukan dari adaptor harus tersambung, hal ini akan ditunjukkan dengan perbandingan nilai, dimana  $DCDIV > V_{ACP}$ .

Beberapa faktor yang dapat menghentikan proses pengisian pada algoritma dari IC. Faktor tersebut adalah sebagai berikut:

- Fungsi *chargingcurrent()* dan *chargingvoltage()* tidak dituliskan
- Daya AC tidak tersambung
- *Alarm inhibited* telah diatur dalam fungsi *chargerstatus()*
- *Inhibit\_charge* diatur dalam fungsi *chargermode()*
- *Reset to zero* diatur dalam fungsi *chargermode()*
- Pin CHGEN bernilai *low*
- Tegangan DCIN tidak mencukupi untuk mengisi baterai
- Fungsi *chargingcurrent()* bernilai 0
- Fungsi *chargingvoltage()* bernilai 0

Yang mengatur kerja dari sistem ini adalah *host*, dimana menggunakan mikrokontroler yang terdapat pada *board* indikator baterai. Fungsinya adalah untuk membaca nilai-nilai yang digunakan sebagai parameter *charging*. Serta mengatur dan memberi perintah pada *charger* agar bekerja sesuai dengan masukan parameter yang telah diterima dari baterai. Pada Baterai BB-2590 memiliki rentang nilai suhu rekomendasi yang diizinkan untuk melakukan pengisian yaitu sebesar 5 – 38°C.

Proses pengisian dimulai dengan menggunakan mode *constant current*. Pada mode tersebut berlangsung hingga tegangan baterai mencapai tegangan 15VDC. Kemudian akan beralih ke mode *constant voltage* hingga baterai terisi penuh. Urutan tersebut dibuat berdasarkan konsep yang umum digunakan.

Proses *charging* dilakukan pada tiap bagian baterai secara bergantian. Sebelum dimulai, *host* akan membaca parameter-parameter

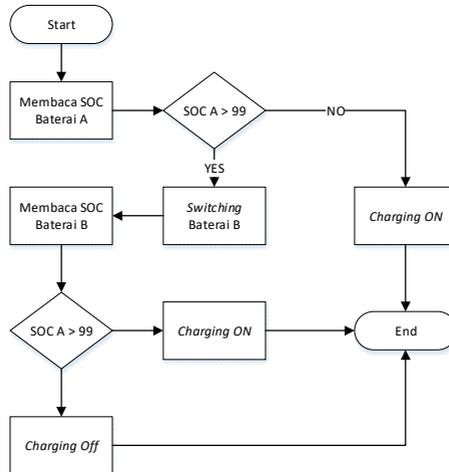
pada bagian A terlebih dahulu yang memiliki pengaruh untuk mengaktifkan dan mematikan *charger*. Jika SoC kurang dari 100% dan suhu berada pada rentang nilai yang diizinkan, *charger* akan mengisi bagian tersebut terlebih dahulu. Namun jika yang terjadi sebaliknya, *host* akan berpindah membaca parameter pada bagian B. Selanjutnya apabila nilai parameter berada pada kondisi yang telah disebutkan sebelumnya, maka *charger* akan aktif. Proses *charging* akan berhenti ketika kedua bagian sudah terisi penuh atau suhu berada dalam rentang nilai yang tidak diizinkan.

Ketika pengisian sedang berlangsung, *host* akan selalu memantau parameter yang terdapat pada baterai. Dengan cara tersebut perintah pada *charger* dapat diubah kapan saja saat terjadi perubahan pada parameter. Nilai yang memiliki pengaruh paling sensitif adalah suhu, dimana setiap ada perubahan kapasitas *charging* arus akan selalu berkurang.

Untuk mengantisipasi terjadinya kecelakaan dalam proses pengisian, pada IC telah dilengkapi proteksi berupa fungsi yang dapat mengirimkan sinyal dan dapat digunakan untuk mengatur proses pengisian. Dalam hal ini, *host* hanya mengirimkan fungsi tersebut dalam bentuk bit, namun tidak semuanya menghambat proses pengisian. Hanya bit *over charged alarm*, *terminate charge alarm*, *reserved*, dan *over temp alarm* yang dapat menghambat proses pengisian dan mengatur *alarm inhibited* dalam fungsi *chargerstatus()*. Sedangkan yang lain tidak memiliki pengaruh pada LTC4100. Untuk mengirimnya digunakan *command code alarmwarning()* (0x16).

*Overcharge* dapat diperoleh dengan membandingkan nilai kapasitas baterai yang telah diisi dengan desain kapasitas baterai (dalam hal ini bisa digunakan SoC). Ketika kondisi ini terjadi pada proses pengisian maka *host* akan diatur untuk mengirimkan bit *over charged alarm* pada *charger*. *Over temperature* didapat dengan membaca nilai suhu baterai dan membandingkannya dengan suhu yang diijinkan oleh baterai. Jika kondisi tersebut terjadi, maka akan dikirimkan bit *over temp alarm*.

Untuk memulai sistem *charging* diperlukan tambahan resistor sebesar 470 *ohm* yang dipasang pada pin *enable* baterai dan dihubungkan dengan 5VDC. Rangkaian ini akan digunakan untuk mengendalikan *on/off charger* dengan menggunakan *host*, dimana 5VDC akan didapat dari *digital output*.

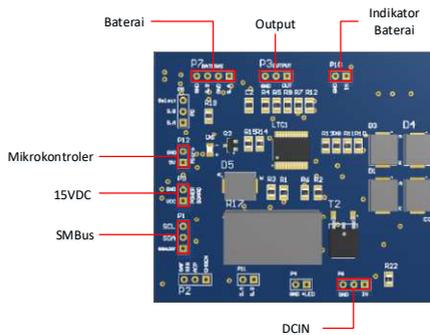


**Gambar 3.16** Flowchart Sistem Charging

### 3.5.3 Konfigurasi Charger

Pada *board charger* terdapat beberapa koneksi yang dihubungkan sesuai dengan keterangan pada Gambar 3.15 DCIN disambungkan dengan keluaran adaptor, dimana bisa digunakan dengan ukuran 24VDC 4A. Pada pin tersebut dipasang *connector* yang biasa digunakan untuk laptop. Tujuannya adalah agar pengguna dapat melakukan pengisian baterai tanpa harus melepasnya terlebih dahulu dari OCU. Untuk SMBus dipasang paralel dengan baterai dan *host* menggunakan pin SDA serta SCL. 15VDC didapat dari *output board power* dengan keterangan indikator baterai. Mikrokontroller dihubungkan dengan VCC dari *host*, fungsinya adalah untuk memberi *supply* IC LTC4100 dan *pull up* resistor pin SMBus. Baterai disambungkan pada keterangan baterai,

dimana telah disediakan 4 koneksi yang terdiri dari sambungan per bagian. *Output* digunakan untuk memberi *supply* pada *board power*. Proses pengisian dapat dilakukan dengan kondisi alat dalam keadaan menyala. Indikator baterai dihubungkan dengan *board* indikator baterai, dimana sebelumnya telah terdapat *switching* yang berfungsi untuk mengalihkan sumber dari baterai atau *charger*.



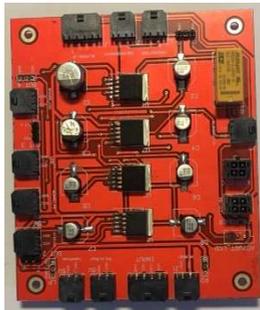
**Gambar 3.17** Konfigurasi *Charger* pada Sistem

**-----Halaman ini sengaja dikosongkan-----**

## BAB IV PENGUJIAN DAN ANALISIS

### 4.1 Sistem Powering

Dalam pengujian ini, akan diambil keluaran dari masing-masing *regulator* dengan menggunakan *power supply* dan baterai. Hasil yang didapat akan dianalisis gangguan pada tegangan dan apa yang mempengaruhi masalah tersebut. kemudian dihitung besar efisiensi dan diperkirakan berapa lama alat dapat dioperasikan.



**Gambar 4. 1** Board Sistem *Powering*

Pada tahap ini juga dilakukan uji coba sistem *powering* disambungkan ke beban. Tujuannya adalah untuk memastikan bahwa beban dapat dinyalakan dengan menggunakan *supply* dari *board* sistem *powering* yang telah dibuat.

#### 4.1.1 Tegangan Keluaran

Pengambilan data pada sistem ini dilakukan dengan menggunakan osiloskop. Dengan alat tersebut bisa didapatkan nilai tegangan keluaran serta gangguan yang dihasilkan. Pada tahap ini dilakukan pengambilan data dengan kondisi keluaran tidak disambungkan dengan beban. Hasil yang didapat telah disediakan pada Tabel 4.1. Grafik yang didapatkan telah disediakan pada halaman lampiran. Untuk data yang telah diambil,

akan dianalisis mengenai faktor apa saja yang dapat mempengaruhi gangguan.

**Tabel 4. 1** Tegangan Keluaran *Voltage Regulator* Tanpa Beban

No	Rangkaian	Keluaran (V)		Gangguan (mV)
		V <sub>MIN</sub>	V <sub>MAX</sub>	
1	Regulator 1	5.1	5.2	160
2	Regulator 2	4.8	5.0	160
3	Regulator 3	11.8	12.0	200
4	Regulator 4	11.8	12.0	200
5	Regulatro 5	12.0	12.2	200

Dari data keluaran *voltage regulator* yang telah diperoleh pada Tabel 4.1, dapat diketahui bahwa masing-masing rangkaian memiliki nilai gangguan yang berbeda-beda. Untuk mengurangi besar nilai gangguan tersebut telah dilakukan uji coba dengan menggunakan berbagai ukuran kapasitor dan induktor, namun hasil yang diperoleh tidak terjadi perubahan.

Hal tersebut diperkirakan dipengaruhi oleh desain rangkaian, dimana tata letak komponen dan jalur *layout* sangat berpengaruh. Disebutkan demikian karena pada uji coba *voltage regulator* dengan menggunakan ukuran yang kecil bisa didapatkan nilai gangguan lebih kecil dibanding dengan ukuran yang lebih besar. Disisi lain pada *datasheet* juga merekomendasikan spesifikasi untuk membuat *board* sesuai dengan yang telah dijelaskan.

Untuk mendapatkan hasil yang lebih baik, komponen tambahan seperti kapasitor, dioda dan induktor harus diletakkan sedekat mungkin dengan IC. Jalur *ground* dibuat selebar dan sependek mungkin serta jumlah titik seminimal mungkin. Untuk *layout* (jalur) VCC dibuat menjauhi induktor. Karena pada komponen ini terbuat dari lilitan, sehingga mampu menghasilkan induksi. Dimana dia dapat menginduksi tegangan pada jalur yang dekat dengannya.

Selain dari desain PCB, besar nilai ESR kapasitor juga memiliki pengaruh terhadap gangguan yang dihasilkan. Nilai tersebut mampu mempengaruhi rangkaian sehingga efisiensi berkurang serta hasil yang didapatkan tidak stabil. Dalam *datasheet* LM2596 telah disediakan

rumus seperti pada persamaan 4.1. Fungsinya adalah untuk menghitung besar nilai ESR yang diperlukan agar gangguan yang dihasilkan bisa lebih kecil.  $\Delta I_{IND}$  merupakan besar nilai dari *ripple current* yang dihasilkan.

$$ESR \text{ of } C_{OUT} = \frac{\text{Gangguan pada Keluaran Tegangan } (\Delta V_{OUT})}{\Delta I_{IND}} \quad (4.1)$$

Dalam hal ini, tidak dapat dilakukan analisis mengenai besar ESR dari kapasitor yang digunakan. Karena data *ripple current* tidak dapat diambil, disebabkan keterbatasan alat. Nilai tersebut bisa diperoleh dengan menggunakan osiloskop dan terdapat rangkaian tambahan yang dihubungkan pada keluaran dan probe osiloskop. Selain itu, juga dapat digunakan *probe* penjepit pada osiloskop.

Dari rumus yang telah dituliskan pada persamaan 4.1, dapat diperkirakan dengan menggunakan nilai  $\Delta I_{IND}$  konstan. Dengan ketentuan tersebut dapat ditarik kesimpulan, jika ingin mendapatkan nilai gangguan yang kecil, maka ESR kapasitor juga harus kecil. Hal tersebut dikarenakan oleh hubungan antara nilai tersebut dan gangguan berbanding lurus.



**Gambar 4. 2** Grafik *Start* Tegangan pada Keluaran *Voltage Regulator*

Kondisi start sumber sangat berpengaruh pada beban, dimana jika terjadi lonjakan pada permulaan mampu memperpendek usia pemakaian *device* yang digunakan. Pada Gambar 4.2, dapat kita lihat bahwa kondisi *start* tegangan pada keluaran *voltage regulator*. Hasil tersebut didapat dari pengukuran tegangan dengan menggunakan mode *rise* pada osiloskop. Dari grafik terlihat proses *start* terjadi secara *soft* dan tidak terdapat *overshoot*. Dari hasil tersebut dapat dipastikan kemungkinan

terjadi *inrush current* sangat kecil, karena hubungan antara arus dan tegangan adalah berbanding lurus. Dimana jika terjadi *inrush current* maka grafik tegangan juga akan mengalami *overshoot*.

Hasil dari pengujian ini, telah sesuai dengan ketentuan yang dibutuhkan. Pada sistem ini berhasil menghasilkan keluaran tegangan sebesar 5VDC dan 12VDC serta mampu memberi *supply* pada komponen, sehingga dapat diaktifkan, walaupun untuk beban Lattepanada harus menggunakan *voltage regulator* dengan tipe yang berbeda.

#### 4.1.2 Efisiensi

Pada tahap ini, dilakukan pengukuran daya masukan dan keluaran, untuk menghitung besar efisiensi. Pengukuran tersebut dilakukan dengan cara menghubungkan masing-masing keluaran dari *voltage regulator* dengan 2 led yang telah dilengkapi resistor 1500 *ohm*. Efisiensi dihitung secara keseluruhan karena, pada masukan telah terhubung pada semua IC. Sehingga perhitungan efisiensi tidak dapat dilakukan pada satu rangkaian, karena disipasi daya yang terjadi disebabkan oleh 4 rangkaian tersebut. Hal ini berbeda dengan *voltage regulator* 1, karena dapat diukur secara terpisah.

**Tabel 4. 2** Efisiensi *Voltage Regulator* 1 (Ubec)

$V_{IN}(V)$	$I_{IN}(A)$	$P_{IN}(W)$	$V_{OUT}(V)$	$I_{OUT}(mA)$	$P_{OUT}(W)$	Efisiensi(%)
15.0	0.02	0.3	5.38	21.28	0.1144864	38.16

Data efisiensi pada *voltage regulator* 1 disediakan pada Tabel 4.2. Untuk data efisiensi *board* sistem *powering* telah disediakan pada Tabel 4.3. Proses pengukuran dilakukan dengan menggunakan *power supply*, dimana telah terdapat LCD yang menampilkan tegangan dan arus keluaran. Pada keluaran *voltage regulator* digunakan multimeter untuk mengukur tegangan dan arus yang masuk ke beban. Data yang didapat telah dituliskan pada Tabel 4.3.

Perhitungan nilai efisiensi dilakukan dengan menggunakan rumus yang telah dituliskan pada persamaan 2.1 yang ada pada pembahasan efisiensi daya di Bab 2. Untuk data pada Tabel 4.3, nilai  $P_{out}$  diperoleh

dengan menjumlahkan semua daya keluaran dari masing-masing *voltage regulator*.

Dari data yang telah diperoleh dapat dihitung besar nilai disipasi pada *board power*. Nilai tersebut dapat dihitung dengan cara daya masukan dikurangi dengan daya keluaran. Hasil yang didapat adalah sebesar 0.472383 watt. Hal ini, disebabkan oleh relay yang digunakan serta disipasi dari masing-masing IC.

**Tabel 4.3** Data Efisiensi *Board Sistem Powering*

Masukan			Rangkaian	Keluaran			Efisiensi (%)
Tegangan (V)	Arus (mA)	Daya (W)		Tegangan (V)	Arus (mA)	Daya (W)	
15.3	60.8	0.93024	Reg 2	5.03	3.02	0.0151906	49.21
			Reg 3	11.93	12.02	0.1433986	
			Reg 4	12.09	12.22	0.1477398	
			Reg 5	12.22	12.40	0.151528	

Hasil dari pengukuran dan perhitungan didapat efisiensi rendah. Hal tersebut dapat disebabkan nilai ESR kapasitor, seperti yang telah dijelaskan pada bagian analisis keluaran sistem *powering*. Jika komponen yang digunakan memiliki toleransi kecil, dapat dipastikan nilai efisiensi yang diperoleh akan meningkat.

Melalui data tersebut, dapat dianalisis berapa alat dapat dioperasikan dengan sumber yang telah disediakan. Untuk menentukannya dihitung daya total beban yang digunakan. Selanjutnya melakukan perhitungan, dimana nilai kapasitas total dari sumber dibagi dengan daya beban, namun nilai efisiensi juga dipertimbangkan. Dalam hal ini, hanya dilakukan analisi pada kondisi alat dioperasikan secara penuh.

Selanjutnya dapat diketahui kebutuhan daya total dengan menjumlahkan *list* daya beban yang telah disediakan pada Tabel 4.4. Untuk menghitung operasi waktu penggunaan OCU, terlebih dahulu dihitung kebutuhan daya masukan secara keseluruhan. Untuk beban yang terhubung pada *board* sistem *powering* dapat dihitung dengan menggunakan persamaan 4.4. Namun, khusus untuk beban LattePanda

dihitung secara terpisah, karena *supply* yang digunakan berasal dari *regulator* Ubec. Selanjutnya untuk indikator baterai data yang digunakan merupakan daya masukan dengan menggunakan tegangan 15VDC dari sumber, sehingga tidak perlu dilakukan perhitungan daya masukan terlebih dahulu.

**Tabel 4. 4** Kebutuhan Daya pada Beban OCU

No.	Beban	Daya (W)
1.	Lattepanda	10
2.	<i>Transmitter</i>	2.5
3.	Led <i>Emergency</i>	0.192
4.	Indikator <i>Power</i>	0.24
5.	<i>Push Button Input</i>	0.08
6.	Indikator Baterai	0.5
7.	Led	0.96
8.	LCD	6
9.	ATMega	0.24

$$P_{in} = \frac{P_{out} \times 100}{Efisiensi} \quad (4.4)$$

Berikut adalah perhitungan untuk menentukan daya masukan yang dibutuhkan oleh *regulator* 1. Dari perhitungan pada persamaan 4.5 hingga 4.6 didapat daya masukan yang dibutuhkan oleh *regulator* Ubec sebesar 26.20 *watt*.

$$P_{in} = \frac{10 \times 100}{38.16} \quad (4.5)$$

$$P_{in} = 26.20 \text{ watt} \quad (4.6)$$

Total daya yang terhubung pada *board* sistem *powering* adalah 25.712 *watt*. Dari nilai tersebut dapat dilakukan perhitungan kebutuhan daya masukan, yang telah ditunjukkan pada persamaan 4.7 sampai 4.8 dimana didapat hasil sebesar 92.89 *watt*.

$$P_{in} = \frac{10.212 \times 100}{49.21} \quad (4.7)$$

$$P_{in} = 20.75 \text{ watt} \quad (4.8)$$

Selanjutnya dilakukan perhitungan daya total, disini daya masukan *board* sistem *powering*, *regulator* Ubec dan indikator baterai dijumlahkan. Dari perhitungan pada persamaan 4.9 hingga 4.11 didapat total daya masukan sebesar 47.45 watt.

$$P_{total} = P_{in\ regulator\ Ubec} + P_{in\ sistem\ powering} + P_{ind.baterai} \quad (4.9)$$

$$P_{total} = 26.20 + 52.24 + 2.5 \quad (4.10)$$

$$P_{total} = 47.45\ watt \quad (4.11)$$

Dari daya total tersebut dihitung nilai arus dengan ketentuan tegangan 15VDC, seperti pada persamaan 4.12 sampai 4.14.

$$I = \frac{P}{V} \quad (4.12)$$

$$I = \frac{47.45}{15} \quad (4.13)$$

$$I = 3.16\ A \quad (4.14)$$

Selanjutnya dilakukan perhitungan durasi waktu alat dapat digunakan, dimana telah ditunjukkan pada persamaan 4.15 hingga 4.20. Dari perhitungan tersebut didapat hasil alat mampu dioperasikan secara penuh dalam waktu sekitar 4:44 jam. Dengan demikian, target untuk mengetahui perkiraan durasi alat dapat dioperasikan telah terpenuhi.

$$t = \frac{I_{Baterai}}{I_{Beban}} \quad (4.15)$$

$$t = \frac{15}{3.16} \quad (4.16)$$

$$t = 4.74\ jam \quad (4.17)$$

$$t = 4.74 \times 60 \quad (4.18)$$

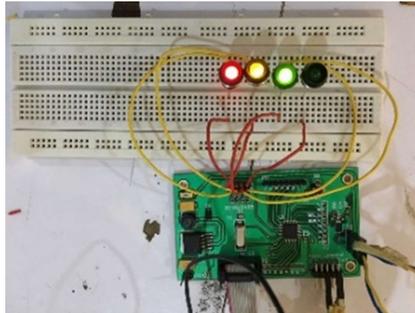
$$t = 284.4\ menit \quad (4.19)$$

$$t = 4:44 \quad (4.20)$$

## 4.2 Indikator Baterai

Dalam tahap ini, dilakukan pengujian dengan cara membaca nilai SoC baterai. Dari nilai tersebut akan dibandingkan dengan perhitungan manual menggunakan nilai *design capacity* dan *remaining capacity*.

Kemudian dihitung nilai *error* dengan acuan perhitungan manual sebagai nilai aktual.



**Gambar 4.3** Uji Coba Indikator Baterai

Hasil uji coba yang ditampilkan pada Gambar 4.3 dilakukan saat nilai SoC 51%. Sehingga dapat dilihat terdapat 3 led yang aktif yaitu merah, kuning dan hijau. Pada pengujian ini, digunakan program yang dijelaskan pada bagian 4.2.1.

#### 4.2.1 Program Indikator Baterai

Secara garis besar program terdiri dari dua proses, yaitu membaca data dan menyalakan led. Pada tahap awal dilakukan pembacaan nilai SoC dari kedua bagian baterai. Selanjutnya, dari nilai rata-rata tersebut digunakan acuan untuk menyalakan led.

Dalam pembuatan program membaca nilai SoC diawali dengan mikrokontroller mengirim permintaan data ke baterai seperti yang ditunjukkan pada program berikut:

```
i2c_start();  
i2c_write(SMB_Write);  
i2c_write(z);  
i2c_stop();  
delay_ms(50);
```

SMB\_Write pada program tersebut, sebelumnya telah dideklarasikan dengan alamat 0x16. Alamat tersebut diperoleh dari alat baterai yang dijumlahkan dengan 0. Untuk z pada program diisi dengan *command*

*code* atau kode dari data yang ingin diambil, dimana dalam hal ini data yang diinginkan adalah data SoC dengan kode 0x0E.

Setelah mengirim permintaan, selanjutnya mikrokontroller dapat membaca data dari baterai dengan menggunakan program seperti berikut:

```
i2c_start();
i2c_write(SMB_Read);
data[0] = i2c_read(1);
data[1] = i2c_read(1);
i2c_stop();
delay_ms(50);
return ((int)data[1]<<8) | (int)data[0];
```

SMB\_Read pada program telah dideklarasikan dengan alamat 0x17. Alamat tersebut diperoleh dengan cara alamat baterai di dijumlahkan (*and*) dengan 1. Pembacaan dilakukan dua kali, karena data yang diminta dalam bentuk *word* sedangkan SMBus sendiri mengirim data dalam bentuk *byte*. Setelah kedua *byte* data diterima, selanjutnya data yang diterima dengan inisial data[1] digeser ke kiri sebanyak 8 bit dan *and* dengan data[0].

Seperti yang telah dijelaskan pada paragraf pertama, bahwa untuk menyalakan led digunakan parameter dari hasil perhitungan nilai rata-rata SoC dari kedua bagian baterai. Selanjutnya, nilai tersebut akan dieksekusi oleh program berikut:

```
if(val>=75)
    PORTD = 0b00001111;
else if (val>=50 && val<75)
    PORTD = 0b00000111;
else if (val>=25 && val<50)
    PORTD = 0b00000011;
else if (val>=10 && val<25)
    PORTD = 0b00000001;
else if (val>=1 && val<10)
    PORTD = 0b00000001;
    delay_ms(500);
    PORTD = 0b00000000;
    delay_ms(500);
```

## 4.2.2 Data SoC

Data diambil dari satu bagian baterai dan dilakukan secara acak, dimana tidak semua kondisi diambil datanya. Semua data yang dibaca pada pengujian ini ditampilkan menggunakan *serial monitor* Arduino. Untuk data lebih lengkap telah disediakan pada bagian lampiran.

**Tabel 4. 5** Sampel Nilai SoC dan Kapasitas Baterai

No.	SoC (%)	Remaining Capacity (mAh)	Design Capacity (mAh)
1	100	6800	6800
2	90	6118	6800
3	70	4749	6800
4	60	4079	6800
5	50	3392	6800
6	40	2716	6800
7	30	2029	6800
8	20	1339	6800
9	10	677	6800

Dari data yang telah diperoleh pada Tabel 4.5, dilakukan perhitungan manual nilai kapasitas. Untuk menghitung kapasitas dapat dilakukan dengan menggunakan Persamaan 4.21.

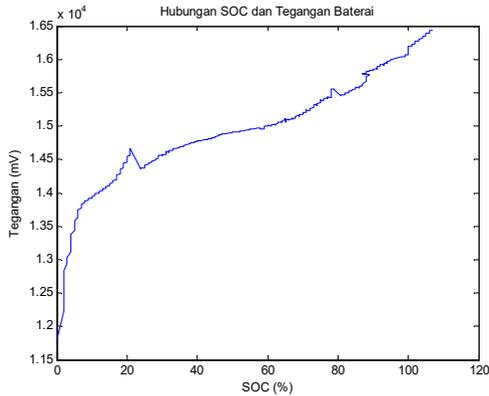
$$Capacity = \frac{Remaining\ Capacity}{Design\ Capacity} \times 100\% \quad (4.21)$$

*Remaining capacity* adalah nilai arus yang tersisa pada baterai dan *design capacity* merupakan kapasitas total arus yang ada pada baterai. Dari data yang telah didapat dan hasil perhitungan, dapat dihitung selisih antara keduanya. Hasil yang diperoleh selisih yang didapatkan rata-rata kurang dari 1, dimana data tersebut dalam versi lengkap telah disediakan pada bagian lampiran.

## 4.3 Hubungan Tegangan dan SoC

Selanjutnya, diambil data hubungan antara SoC dan tegangan, tujuannya adalah untuk mengetahui berapa tegangan baterai ketika dalam kondisi penuh dan kosong. Dari pengambilan data didapat ketika

baterai terisi penuh, SoC yang didapatkan sebesar 107%, dan nilai tegangan sekitar 16.35VDC. Sedangkan dalam keadaan kosong, besar tegangan sekitar 11.9VDC.



**Gambar 4.4** Grafik Hubungan SoC dan Tegangan Baterai

Pada nilai SoC yang didapatkan lebih dari 100%, hal tersebut telah dijelaskan pada literatur [17]. Dimana dijelaskan pada penggunaan *command code 0x0E* bisa didapatkan nilai melebihi 100%.

Dari pembacaan data tersebut, maka dapat digunakan acuan dalam pembuatan sistem *charging* untuk OCU versi selanjutnya. Dimana tegangan *threshold* diatur sebesar 16.35VDC. Sehingga acuan yang digunakan adalah nilai tersebut, tujuannya adalah untuk memaksimalkan proses pengisian. Sedangkan untuk nilai SoC dapat diatur sesuai data yang telah dibaca yaitu 107%.

#### 4.4 Validasi Hasil Pengujian

Pada bagian ini, dilakukan validasi dari hasil pengujian terhadap target-target yang telah dijelaskan pada Bab III. Pada Tabel 4.6 disajikan hasil dari validasi sistem yang telah dilakukan. Berdasarkan keterangan tersebut, dapat dipastikan bahwa sistem berhasil dibuat.

**Tabel 4.6** Validasi Desain

Target	Sistem	Hasil
Menghasilkan Keluaran 5VDC dan 12VDC	Sistem <i>Powering</i>	Dapat diperoleh keluaran sebesar 5VDC dan 12VDC seperti yang telah disajikan pada Tabel dan Lampiran 1 hingga 5.
Dapat mengaktifkan komponen yang digunakan dalam OCU		Sistem <i>powering</i> dapat mengaktifkan komponen-komponen penyusun OCU seperti yang ditampilkan pada Lampiran 6 hingga 8.
Selisih Nilai SoC dan perhitungan kapasitas secara manual kecil	Indikator Baterai	Diperoleh selisih dari kedua nilai tersebut rata-rata kurang dari 1%, seperti yang telah disajikan pada Lampiran 12 dan 13.
Mengetahui rentang nilai parameter yang digunakan pada konsep sistem <i>charging</i>	Sistem <i>Charging</i>	Terdapat 3 parameter yang digunakan yaitu suhu, tegangan dan SoC. Rentang suhu baterai yang diijinkan untuk proses pengisian sebesar 5 – 38°C. Nilai tegangan dan SoC saat baterai penuh sebesar 16.35VDC dan 107%, sedangkan saat baterai habis sebesar 11.9VDC dan 0%

## **BAB V**

### **PENUTUP**

Berdasarkan data yang diperoleh dari hasil pengujian dan analisis didapatkan nilai gangguan pada sistem *powering* 160mV untuk keluaran 5VDC, dan 200mV pada keluaran 12VDC. Dari hasil uji coba komponen yang digunakan mampu bekerja dengan ketentuan *supply* seperti yang telah disebutkan. Selain itu, juga didapat kondisi *soft start* pada keluaran *voltage regulator*, sehingga aman untuk komponen yang digunakan serta didapat efisiensi pada *board* sistem *powering* sebesar 49.21% dan *regulator* 1 dari Ubec 38.16%. Dari nilai tersebut, alat mampu beroperasi selama 4 jam 44 menit. Dalam pembuatan indikator baterai ditampilkan menggunakan 4 led dengan parameter yang digunakan adalah SoC baterai, dimana dari hasil pengujian didapat nilai 107% saat baterai dalam keadaan penuh dan 0% saat baterai pada keadaan habis. Untuk mempermudah pengguna dalam melakukan proses pengisian, digunakan *connector* pada OCU. Parameter yang digunakan pada sistem *charging* adalah tegangan, SoC, dan suhu. Tegangan pada baterai saat penuh bernilai 16.35VDC dan ketika habis adalah 11.9VDC. Nilai tersebut dapat digunakan sebagai acuan untuk pembuatan sistem pengisian baterai. Suhu yang diijinkan untuk mengisi baterai sebesar 5 – 38<sup>0</sup>C.

Untuk pengembangan kedepan, komponen yang digunakan pada sistem *powering* harus memiliki nilai toleransi kecil. Ukuran *board* juga perlu dibuat sekecil mungkin dengan posisi komponen saling berdekatan. Tujuannya adalah untuk mendapat nilai gangguan yang kecil dan arus konstan. Selain itu, cara tersebut juga mampu meningkatkan efisiensi, sehingga durasi pemakaian alat dapat bertambah. Saat melakukan pembuatan sistem indikator baterai dan sistem *charging* parameter yang ada pada baterai terlebih dahulu dibaca. Karena tiap baterai memiliki nilai yang berbeda-beda sehingga dengan dilakukan langkah tersebut, mampu memaksimalkan performa dari sistem.

**-----Halaman ini sengaja dikosongkan-----**

## DAFTAR PUSTAKA

- [1] J. J. Zeng, R. Q. Yang, W. J. Zhang, X. H. Weng, and Q. Jun, "Research on semi-automatic bomb fetching for an EOD robot," *Int. J. Adv. Robot. Syst.*, vol. 4, no. 2, pp. 247–252, 2007.
- [2] N. Checka, S. Schaffert, D. Demirdjian, J. Falkowski, and D. H. Grollman, "Handheld operator control unit," *Proc. seventh Annu. ACM/IEEE Int. Conf. Human-Robot Interact. - HRI '12*, p. 137, 2012.
- [3] F. Dastgeer, "Direct current distribution systems for residential areas powered by distributed generation," no. December, p. 4, 2011.
- [4] L. Fan, X. Yao, H. Qi, L. Jiang, and W. Wang, "A single-hand and binocular visual system for EOD robot," *Proc. IEEE Int. Conf. Autom. Logist. ICAL 2007*, pp. 1930–1935, 2007.
- [5] L. Dams, *Bomb Disposal Robot History*. [www.youtube.com](http://www.youtube.com), 2014.
- [6] B. T. Patterson, "DC Power Distribution System for Rural Applications," no. October, p. 2, 2012.
- [7] \_\_\_\_\_, *Lithium Ion Technical Handbook.pdf*.
- [8] A. P. Hartmunt . A, Joseph . D, Frederique . T, "Lifetime Analysis of Four Different Lithium Ion Batteries for Plug In Electric Vehicle.pdf," 2014.
- [9] C. A. Lundgren, K. Xu, T. R. Jow, J. Allen, and S. S. Zhang, "Lithium-Ion B."
- [10] E. Wikner, "Lithium ion Battery Aging: Battery Lifetime Testing and Physics-based Modeling for Electric Vehicle Applications," Chalmers University of Technology, 2017.
- [11] R. Scipioni, S. H. Jensen, J. Hjelm, and P. Norby, "Performance and Lifetime Limiting Effects in Li-ion Batteries," *Ph.D. Thesis (DTU)*, 2016.
- [12] T. R. M, Celina . K, Michael . W, Kevin . L, "Lithium Ion Batteries Hazard and Use Assessment.pdf," Quincy, Massachusetts, USA, 2011.
- [13] \_\_\_\_\_, "Lithium Ion." Available: [http://www.rathboneenergy.com/articles/sanyo\\_lionT\\_E.pdf](http://www.rathboneenergy.com/articles/sanyo_lionT_E.pdf). [Accessed: 23-Mar-2018].
- [14] \_\_\_\_\_, "How to Prolong Lithium based Batteries," 2018. Available:[http://batteryuniversity.com/learn/article/how\\_to\\_pr](http://batteryuniversity.com/learn/article/how_to_pr)

- along\_lithium\_based\_batteries. [Accessed: 23-Mar-2018].
- [15] \_\_\_\_\_, “How do Lithium Batteries Work?,” 2017.. Available:[http://batteryuniversity.com/learn/article/lithium\\_based\\_batteries](http://batteryuniversity.com/learn/article/lithium_based_batteries). [Accessed: 23-Mar-2018].
- [16] \_\_\_\_\_, “Smart Battery.”. Available: <https://www.techopedia.com/definition/13294/smart-battery>. [Accessed: 26-Mar-2018].
- [17] Intel and Duracell, “Smart Battery Data Specification,” vol. 1.1, 1998.
- [18] I. F. SBS, “System Management Bus Specification,” vol. 1.1, no. December, 2000.
- [19] R. Fan, “SMBus Quick Start Guide,” AN4471, 2012.
- [20] \_\_\_\_\_, “Differences between I2C and SMBus.”. Available: <https://www.totalphase.com/support/articles/200349186-Differences-between-I2C-and-SMBus>. [Accessed: 26-Mar-2018].
- [21] C. Simpson, “Linear and Switching Voltage Regulator Fundamental part 1,” SNVA558, 2011.
- [22] \_\_\_\_\_, “What is the Difference Between Linear and Switching Regulators?.”. Available: <https://www.rohm.com/electronics-basics/dc-dc-converters/linear-vs-switching-regulators/>. [Accessed: 15-Mar-2018].
- [23] Texas Instruments, “Switching Regulator Fundamentals,” 2016.
- [24] D. Converters, “Inrush Current in DC-DC Converters,” pp. 1–9.
- [25] \_\_\_\_\_, “What is the difference between Inrush Current and Peak Current?.”. Available: <https://pacificpower.com/2017/05/difference-inrush-current-peak-current/>. [Accessed: 26-Mar-2018].
- [26] R. Fay, “Inrush Current.”. Available: <https://forum.digikey.com/t/inrush-current/679>. [Accessed: 25-Mar-2018].
- [27] A. Kaknevicus, Alek; Hoover, “Managing Inrush Current,” 2015.
- [28] M. L. T. Cossio *et al.*, “Filtering,” *Uma ética para quantos?*, vol. XXXIII, no. 2, 2012.
- [29] P. Wang, “Effective use of filter capacitors to clean up voltage source signals in portable consumer designs.”. Available: <https://www.embedded.com/design/real-time-and->

- performance/4007179/Effective-use-of-filter-capacitors-to-clean-up-voltage-source-signals-in-portable-consumer-designs. [Accessed: 27-Mar-2018].
- [30] Sunu Pradana, "Adaptasi Peralatan Penguji Kondisi Esr Pada Kapasitor," no. May 2013, 2013.
- [31] Texas Instruments, "LM2596 SIMPLE SWITCHER ® Power Converter 150-kHz 3-A Step-Down Voltage Regulator," 2016.
- [32] A. F. Farizy and D. A. Asfani, "Desain Sistem Monitoring State of Charge Baterai pada Charging Station Mobil Listrik Berbasis Fuzzy Logic Dengan Mempertimbangkan Temperature," vol. 5, no. 2, 2016.
- [33] K. Thoubboron, "Depth of discharge (DoD): What does it mean for your battery, and why is it important?," 2018.. Available: <https://news.energysage.com/depth-discharge-dod-mean-battery-important/>. [Accessed: 25-Mar-2018].
- [34] \_\_\_\_\_, "Battery Chargers and Charging Methods," 2005.. Available: <http://www.mpoweruk.com/chargers.htm>. [Accessed: 13-Mar-2017].
- [35] C. Zhang, Y. Zheng, L. Li, X. Wang, and J. Yu, "Design and Simulation of EV Charging Device Based on Constant Voltage-Constant Current PFC Double Closed-Loop Controller," no. Icaees, 2015.
- [36] W. Wu, "Charging time estimation and study of charging behavior for automotive Li-ion battery cells using a Matlab / Simulink model," School of Electrical Engineering, 2016.
- [37] \_\_\_\_\_, "Constant Current." Available: <http://www.amadamiyachi.com/glossary/glossconstantcurrent>. [Accessed: 27-Mar-2018].
- [38] \_\_\_\_\_, "What is Constant Current Source?" Available: <http://www.learningaboutelectronics.com/Articles/What-is-a-constant-current-source.php>. [Accessed: 25-Mar-2018].
- [39] \_\_\_\_\_, "Constant Voltage." Available: <http://www.amadamiyachi.com/glossary/glossconstantvoltage>. [Accessed: 25-Mar-2018].
- [40] \_\_\_\_\_, "Constant Voltage," 2014.. Available: <https://www.sunpower-uk.com/glossary/constant-voltage/>. [Accessed: 26-Mar-2018].
- [41] M. Elias, A. Arof, and K. Nor, "Design of high energy lithium-ion battery charger," *Australas. Univ. Power Eng. Conf.*, no.

September, 2004.

## LAMPIRAN

**Lampiran 1:** Grafik Keluaran *Voltage Regulator 1* (Ubec)

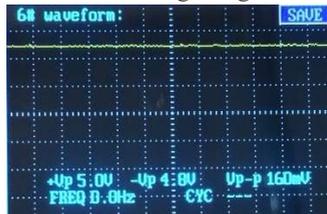


Keterangan :

*Volt per div* = 2V

*Time per div* = 200ms

**Lampiran 2:** Grafik Keluaran *Voltage Regulator 2* (5VDC)



Keterangan :

*Volt per div* = 2V

*Time per div* = 200ms

**Lampiran 3:** Grafik Keluaran *Voltage Regulator 3* (12VDC)



Keterangan :

*Volt per div* = 5V

*Time per div* = 200ms

**Lampiran 4:** Grafik Keluaran *Voltage Regulator 4* (12VDC)



Keterangan :

*Volt per div* = 5V

*Time per div* = 200ms

**Lampiran 5:** Grafik Keluaran *Voltage Regulator 5* (12VDC)

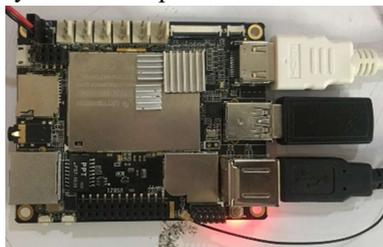


Keterangan :

*Volt per div* = 5V

*Time per div* = 200ms

**Lampiran 6:** Menyalakan LattePanda



Keterangan :

Berhasil jika Led warna merah nyala secara stabil dan bisa diaktifkan

### **Lampiran 7:** Menyalakan *Transmitter*



Keterangan :

Berhasil jika led indikator yang terdapat pada *transmitter* menyala

### **Lampiran 8:** Menyalakan LCD



**Lampiran 9:** *Listing* Program Indikator Baterai

```
#include <mega8.h>
```

```
#include <i2c.h>
```

```
#include <delay.h>
```

```
#define SMB_Write 0x16
```

```
#define SMB_Read 0x17
```

```
unsigned char data[4];
```

```
unsigned int value[8];
```

```
int avrVal;
```

```
int readWord(int z);
```

```
void LED(int val);
```

```
void main(void)
```

```
{
```

```
PORTB =0x00;
```

```
DDRB =0x00;
```

```
PORTC =0x00;
```

```
DDRC =0x00;
```

```
Func0 =Out
```

```
PORTD =0x00;
```

```
DDRD =0x3F;
```

```
TCCR0 =0x00;
```

```
TCNT0 =0x00;
```

```
TCCR1A=0x00;
```

```
TCCR1B=0x00;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H =0x00;
```

```
ICR1L =0x00;
```

```
OCR1AH=0x00;
```

```
OCR1AL=0x00;
```

```
OCR1BH=0x00;
```

```
OCR1BL=0x00;
```

```
ASSR =0x00;
```

```
TCCR2 =0x00;
```

```

TCNT2 =0x00;
OCR2 =0x00;
MCUCR =0x00;
TIMSK =0x00;
UCSRB =0x00;
ACSR =0x80;
SFIO =0x00;
ADCSRA=0x00;
SPCR =0x00;
TWCR =0x00;

i2c_init();

while (1)
{
    PORTD.5 = 1;
    value[0] = readWord(0x0E);

    PORTD.5 = 0;
    value[1] = readWord(0x0E);

    avrVal = (value[0] + value[1])/2;
    LED(avrVal);

    delay_ms(100);
}
}

int readWord(int z)
{
    i2c_start();
    i2c_write(SMB_Write);
    i2c_write(z);
    i2c_stop();
    delay_ms(50);
}

```

```

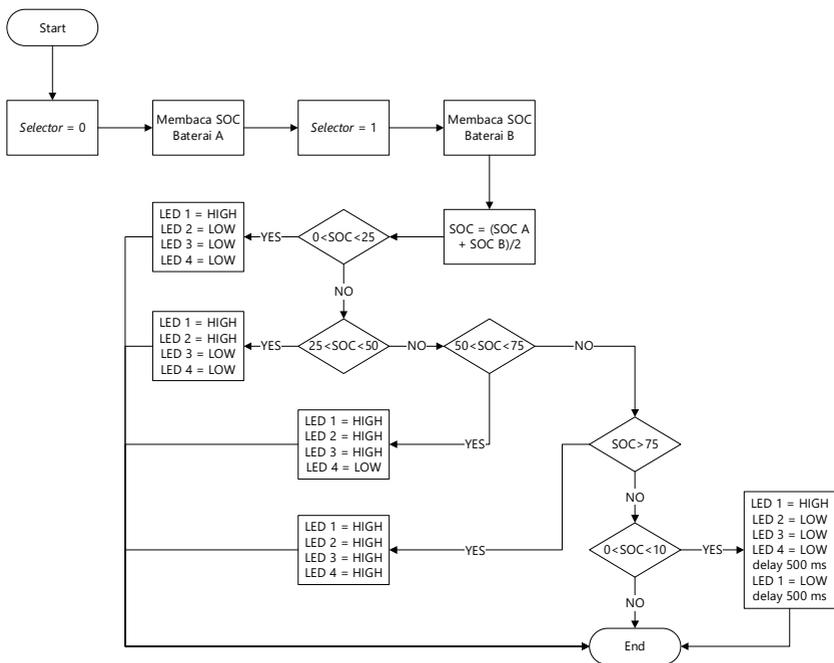
i2c_start();
i2c_write(SMB_Read);
data[0] = i2c_read(1);
data[1] = i2c_read(1);
i2c_stop();
delay_ms(50);

return ((int)data[1]<<8) | (int)data[0];
}

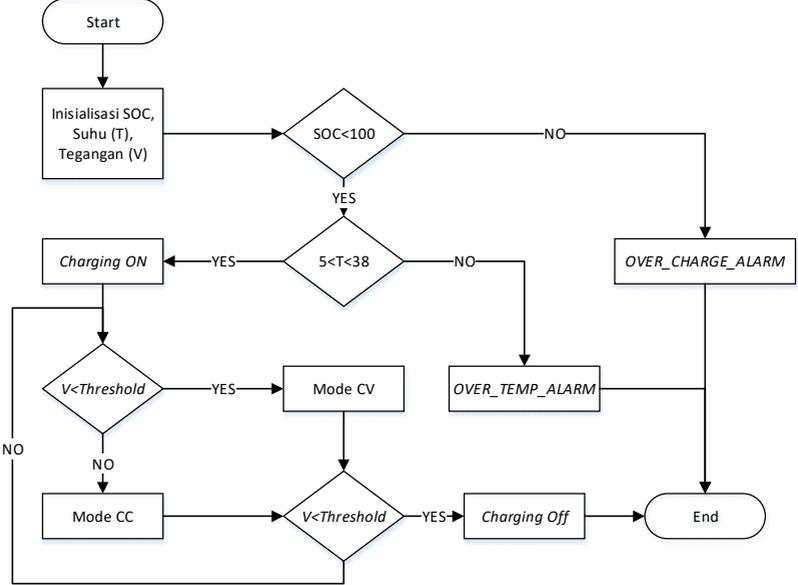
void LED(int val)
{
  if(val>=75)
    PORTD = 0b00001111;
  else if (val>=50 && val<75)
    PORTD = 0b00000111;
  else if (val>=25 && val<50)
    PORTD = 0b00000011;
  else if (val>=10 && val<25)
    PORTD = 0b00000001;
  else if (val>=1 && val<10)
    PORTD = 0b00000001;
    delay_ms(500);
    PORTD = 0b00000000;
    delay_ms(500);
}

```

## Lampiran 10: Folwchart Indikator Baterai



**Lampiran 11: Algoritma Proses Pengisian Baterai**



**Lampiran 12: Data SoC**

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
107	16431	7212	6800	106.0588235	0.941176471
106	16428	7206	6800	105.9705882	0.029411765
106	16422	7201	6800	105.8970588	0.102941176
106	16420	7195	6800	105.8088235	0.191176471
106	16417	7189	6800	105.7205882	0.279411765
106	16412	7184	6800	105.6470588	0.352941176
106	16409	7178	6800	105.5588235	0.441176471
106	16405	7173	6800	105.4852941	0.514705882
106	16399	7166	6800	105.3823529	0.617647059
106	16398	7161	6800	105.3088235	0.691176471
106	16391	7149	6800	105.1323529	0.867647059
106	16388	7144	6800	105.0588235	0.941176471
105	16384	7138	6800	104.9705882	0.029411765
105	16380	7132	6800	104.8823529	0.117647059
105	16378	7127	6800	104.8088235	0.191176471
105	16374	7121	6800	104.7205882	0.279411765
105	16371	7115	6800	104.6323529	0.367647059
105	16369	7109	6800	104.5441176	0.455882353
105	16364	7104	6800	104.4705882	0.529411765
105	16362	7098	6800	104.3823529	0.617647059
105	16357	7092	6800	104.2941176	0.705882353
105	16354	7087	6800	104.2205882	0.779411765

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
105	16352	7081	6800	104.1323529	0.867647059
105	16348	7075	6800	104.0441176	0.955882353
104	16345	7070	6800	103.9705882	0.029411765
104	16341	7064	6800	103.8823529	0.117647059
104	16338	7058	6800	103.7941176	0.205882353
104	16335	7052	6800	103.7058824	0.294117647
104	16331	7047	6800	103.6323529	0.367647059
104	16329	7041	6800	103.5441176	0.455882353
104	16326	7035	6800	103.4558824	0.544117647
104	16323	7030	6800	103.3823529	0.617647059
104	16319	7024	6800	103.2941176	0.705882353
104	16317	7018	6800	103.2058824	0.794117647
104	16313	7013	6800	103.1323529	0.867647059
104	16310	7007	6800	103.0441176	0.955882353
103	16307	7001	6800	102.9558824	0.044117647
103	16303	6996	6800	102.8823529	0.117647059
103	16300	6990	6800	102.7941176	0.205882353
103	16297	6984	6800	102.7058824	0.294117647
103	16294	6978	6800	102.6176471	0.382352941
103	16291	6973	6800	102.5441176	0.455882353
103	16288	6967	6800	102.4558824	0.544117647
103	16284	6961	6800	102.3676471	0.632352941
103	16281	6955	6800	102.2794118	0.720588235

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
103	16278	6949	6800	102.1911765	0.808823529
103	16276	6944	6800	102.1176471	0.882352941
103	16273	6938	6800	102.0294118	0.970588235
102	16269	6932	6800	101.9411765	0.058823529
102	16265	6927	6800	101.8676471	0.132352941
102	16264	6921	6800	101.7794118	0.220588235
102	16260	6915	6800	101.6911765	0.308823529
102	16257	6909	6800	101.6029412	0.397058824
102	16255	6904	6800	101.5294118	0.470588235
102	16251	6898	6800	101.4411765	0.558823529
102	16249	6892	6800	101.3529412	0.647058824
102	16245	6886	6800	101.2647059	0.735294118
102	16242	6881	6800	101.1911765	0.808823529
102	16240	6875	6800	101.1029412	0.897058824
102	16236	6869	6800	101.0147059	0.985294118
101	16234	6863	6800	100.9264706	0.073529412
101	16231	6858	6800	100.8529412	0.147058824
101	16228	6852	6800	100.7647059	0.235294118
101	16225	6846	6800	100.6764706	0.323529412
101	16221	6841	6800	100.6029412	0.397058824
101	16219	6835	6800	100.5147059	0.485294118
101	16215	6829	6800	100.4264706	0.573529412
101	16212	6823	6800	100.3382353	0.661764706

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
101	16209	6818	6800	100.2647059	0.735294118
101	16207	6812	6800	100.1764706	0.823529412
101	16204	6806	6800	100.0882353	0.911764706
100	16201	6800	6800	100	0
100	16197	6794	6800	99.91176471	0.088235294
100	16106	6799	6800	99.98529412	0.014705882
100	16097	6792	6800	99.88235294	0.117647059
100	16096	6784	6800	99.76470588	0.235294118
100	16091	6776	6800	99.64705882	0.352941176
100	16088	6768	6800	99.52941176	0.470588235
100	16082	6760	6800	99.41176471	0.588235294
100	16078	6752	6800	99.29411765	0.705882353
100	16070	6744	6800	99.17647059	0.823529412
100	16069	6735	6800	99.04411765	0.955882353
99	16065	6727	6800	98.92647059	0.073529412
99	16059	6719	6800	98.80882353	0.191176471
99	16058	6711	6800	98.69117647	0.308823529
99	16053	6703	6800	98.57352941	0.426470588
99	16046	6695	6800	98.45588235	0.544117647
99	16045	6686	6800	98.32352941	0.676470588
99	16041	6679	6800	98.22058824	0.779411765
99	16038	6671	6800	98.10294118	0.897058824
98	16045	6663	6800	97.98529412	0.014705882

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
98	16046	6656	6800	97.88235294	0.117647059
98	16045	6650	6800	97.79411765	0.205882353
98	16044	6644	6800	97.70588235	0.294117647
95	15999	6446	6800	94.79411765	0.205882353
95	15991	6438	6800	94.67647059	0.323529412
95	15991	6429	6800	94.54411765	0.455882353
95	15990	6421	6800	94.42647059	0.573529412
95	15979	6413	6800	94.30882353	0.691176471
95	15980	6405	6800	94.19117647	0.808823529
95	15975	6397	6800	94.07352941	0.926470588
94	15968	6381	6800	93.83823529	0.161764706
94	15964	6373	6800	93.72058824	0.279411765
94	15962	6364	6800	93.58823529	0.411764706
94	15970	6357	6800	93.48529412	0.514705882
94	15960	6349	6800	93.36764706	0.632352941
94	15953	6342	6800	93.26470588	0.735294118
94	15941	6334	6800	93.14705882	0.852941176
94	15945	6325	6800	93.01470588	0.985294118
93	15938	6318	6800	92.91176471	0.088235294
93	15932	6309	6800	92.77941176	0.220588235
93	15931	6301	6800	92.66176471	0.338235294
93	15932	6293	6800	92.54411765	0.455882353
93	15940	6285	6800	92.42647059	0.573529412

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
93	15934	6278	6800	92.32352941	0.676470588
93	15914	6270	6800	92.20588235	0.794117647
93	15920	6262	6800	92.08823529	0.911764706
92	15917	6255	6800	91.98529412	0.014705882
92	15924	6248	6800	91.88235294	0.117647059
92	15903	6239	6800	91.75	0.25
92	15899	6231	6800	91.63235294	0.367647059
92	15898	6223	6800	91.51470588	0.485294118
92	15894	6214	6800	91.38235294	0.617647059
92	15900	6207	6800	91.27941176	0.720588235
92	15898	6200	6800	91.17647059	0.823529412
92	15892	6193	6800	91.07352941	0.926470588
91	15897	6186	6800	90.97058824	0.029411765
91	15873	6177	6800	90.83823529	0.161764706
91	15870	6169	6800	90.72058824	0.279411765
91	15872	6161	6800	90.60294118	0.397058824
91	15860	6152	6800	90.47058824	0.529411765
91	15857	6143	6800	90.33823529	0.661764706
91	15856	6134	6800	90.20588235	0.794117647
91	15865	6126	6800	90.08823529	0.911764706
90	15845	6118	6800	89.97058824	0.029411765
90	15852	6110	6800	89.85294118	0.147058824
90	15835	6101	6800	89.72058824	0.279411765

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
90	15839	6092	6800	89.58823529	0.411764706
90	15848	6085	6800	89.48529412	0.514705882
90	15850	6078	6800	89.38235294	0.617647059
90	15849	6072	6800	89.29411765	0.705882353
90	15848	6066	6800	89.20588235	0.794117647
90	15845	6059	6800	89.10294118	0.897058824
90	15842	6053	6800	89.01470588	0.985294118
89	15839	6046	6800	88.91176471	0.088235294
89	15837	6040	6800	88.82352941	0.176470588
89	15834	6034	6800	88.73529412	0.264705882
89	15832	6027	6800	88.63235294	0.367647059
89	15829	6021	6800	88.54411765	0.455882353
89	15827	6015	6800	88.45588235	0.544117647
89	15825	6008	6800	88.35294118	0.647058824
89	15822	6002	6800	88.26470588	0.735294118
89	15820	5996	6800	88.17647059	0.823529412
89	15818	5990	6800	88.08823529	0.911764706
88	15815	5983	6800	87.98529412	0.014705882
88	15812	5977	6800	87.89705882	0.102941176
88	15809	5971	6800	87.80882353	0.191176471
88	15807	5965	6800	87.72058824	0.279411765
88	15806	5958	6800	87.61764706	0.382352941
88	15803	5952	6800	87.52941176	0.470588235

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
88	15800	5946	6800	87.44117647	0.558823529
88	15797	5940	6800	87.35294118	0.647058824
88	15796	5933	6800	87.25	0.75
88	15792	5927	6800	87.16176471	0.838235294
88	15790	5921	6800	87.07352941	0.926470588
87	15787	5915	6800	86.98529412	0.014705882
87	15784	5908	6800	86.88235294	0.117647059
87	15782	5902	6800	86.79411765	0.205882353
87	15780	5896	6800	86.70588235	0.294117647
89	15777	6044	6800	88.88235294	0.117647059
89	15774	6038	6800	88.79411765	0.205882353
89	15772	6031	6800	88.69117647	0.308823529
89	15770	6025	6800	88.60294118	0.397058824
89	15768	6019	6800	88.51470588	0.485294118
89	15763	6006	6800	88.32352941	0.676470588
89	15760	6000	6800	88.23529412	0.764705882
89	15757	5994	6800	88.14705882	0.852941176
89	15755	5988	6800	88.05882353	0.941176471
88	15754	5981	6800	87.95588235	0.044117647
88	15750	5975	6800	87.86764706	0.132352941
88	15747	5969	6800	87.77941176	0.220588235
88	15744	5963	6800	87.69117647	0.308823529
88	15743	5957	6800	87.60294118	0.397058824

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
88	15740	5950	6800	87.5	0.5
88	15738	5944	6800	87.41176471	0.588235294
88	15735	5938	6800	87.32352941	0.676470588
88	15732	5932	6800	87.23529412	0.764705882
88	15676	5920	6800	87.05882353	0.941176471
87	15634	5906	6800	86.85294118	0.147058824
87	15636	5890	6800	86.61764706	0.382352941
87	15631	5876	6800	86.41176471	0.588235294
87	15603	5861	6800	86.19117647	0.808823529
86	15595	5844	6800	85.94117647	0.058823529
86	15583	5828	6800	85.70588235	0.294117647
86	15581	5813	6800	85.48529412	0.514705882
86	15580	5797	6800	85.25	0.75
86	15575	5781	6800	85.01470588	0.985294118
85	15586	5766	6800	84.79411765	0.205882353
85	15580	5752	6800	84.58823529	0.411764706
85	15580	5739	6800	84.39705882	0.602941176
85	15552	5724	6800	84.17647059	0.823529412
84	15546	5708	6800	83.94117647	0.058823529
84	15549	5693	6800	83.72058824	0.279411765
84	15542	5678	6800	83.5	0.5
84	15536	5664	6800	83.29411765	0.705882353
84	15529	5649	6800	83.07352941	0.926470588

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
83	15525	5634	6800	82.85294118	0.147058824
83	15514	5619	6800	82.63235294	0.367647059
83	15505	5603	6800	82.39705882	0.602941176
83	15503	5588	6800	82.17647059	0.823529412
82	15496	5573	6800	81.95588235	0.044117647
82	15491	5558	6800	81.73529412	0.264705882
82	15481	5543	6800	81.51470588	0.485294118
82	15479	5527	6800	81.27941176	0.720588235
82	15468	5512	6800	81.05882353	0.941176471
81	15466	5496	6800	80.82352941	0.176470588
81	15459	5481	6800	80.60294118	0.397058824
79	15549	5311	6800	78.10294118	0.897058824
79	15553	5305	6800	78.01470588	0.985294118
78	15555	5300	6800	77.94117647	0.058823529
78	15553	5294	6800	77.85294118	0.147058824
78	15553	5288	6800	77.76470588	0.235294118
78	15551	5282	6800	77.67647059	0.323529412
78	15531	5273	6800	77.54411765	0.455882353
78	15459	5259	6800	77.33823529	0.661764706
78	15430	5245	6800	77.13235294	0.867647059
77	15436	5228	6800	76.88235294	0.117647059
77	15416	5213	6800	76.66176471	0.338235294
77	15419	5198	6800	76.44117647	0.558823529

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
77	15424	5183	6800	76.22058824	0.779411765
77	15425	5170	6800	76.02941176	0.970588235
76	15413	5157	6800	75.83823529	0.161764706
76	15418	5144	6800	75.64705882	0.352941176
76	15388	5130	6800	75.44117647	0.558823529
76	15384	5114	6800	75.20588235	0.794117647
75	15384	5098	6800	74.97058824	0.029411765
75	15389	5085	6800	74.77941176	0.220588235
75	15358	5070	6800	74.55882353	0.441176471
75	15347	5053	6800	74.30882353	0.691176471
75	15350	5038	6800	74.08823529	0.911764706
74	15333	5022	6800	73.85294118	0.147058824
74	15335	5006	6800	73.61764706	0.382352941
74	15335	5019	6800	73.80882353	0.191176471
74	15313	5003	6800	73.57352941	0.426470588
74	15308	4987	6800	73.33823529	0.661764706
74	15323	4972	6800	73.11764706	0.882352941
73	15325	4959	6800	72.92647059	0.073529412
73	15287	4945	6800	72.72058824	0.279411765
73	15286	4930	6800	72.5	0.5
73	15281	4914	6800	72.26470588	0.735294118
73	15272	4899	6800	72.04411765	0.955882353
72	15267	4883	6800	71.80882353	0.191176471

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
72	15255	4868	6800	71.58823529	0.411764706
72	15249	4853	6800	71.36764706	0.632352941
72	15242	4837	6800	71.13235294	0.867647059
71	15236	4822	6800	70.91176471	0.088235294
71	15246	4807	6800	70.69117647	0.308823529
71	15247	4794	6800	70.5	0.5
71	15224	4780	6800	70.29411765	0.705882353
71	15207	4764	6800	70.05882353	0.941176471
70	15200	4749	6800	69.83823529	0.161764706
70	15194	4733	6800	69.60294118	0.397058824
70	15189	4717	6800	69.36764706	0.632352941
70	15182	4702	6800	69.14705882	0.852941176
69	15171	4686	6800	68.91176471	0.088235294
69	15166	4671	6800	68.69117647	0.308823529
69	15156	4655	6800	68.45588235	0.544117647
69	15154	4640	6800	68.23529412	0.764705882
69	15160	4625	6800	68.01470588	0.985294118
68	15145	4609	6800	67.77941176	0.220588235
68	15136	4594	6800	67.55882353	0.441176471
68	15124	4578	6800	67.32352941	0.676470588
68	15122	4562	6800	67.08823529	0.911764706
67	15115	4546	6800	66.85294118	0.147058824
67	15111	4530	6800	66.61764706	0.382352941

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
67	15124	4516	6800	66.41176471	0.588235294
67	15122	4502	6800	66.20588235	0.794117647
66	15093	4487	6800	65.98529412	0.014705882
66	15092	4471	6800	65.75	0.25
66	15091	4455	6800	65.51470588	0.485294118
66	15102	4440	6800	65.29411765	0.705882353
66	15101	4427	6800	65.10294118	0.897058824
65	15095	4414	6800	64.91176471	0.088235294
65	15087	4400	6800	64.70588235	0.294117647
65	15064	4384	6800	64.47058824	0.529411765
65	15062	4368	6800	64.23529412	0.764705882
65	15121	4356	6800	64.05882353	0.941176471
64	15065	4341	6800	63.83823529	0.161764706
64	15064	4326	6800	63.61764706	0.382352941
64	15068	4313	6800	63.42647059	0.573529412
64	15068	4299	6800	63.22058824	0.779411765
64	15065	4286	6800	63.02941176	0.970588235
63	15064	4273	6800	62.83823529	0.161764706
63	15059	4260	6800	62.64705882	0.352941176
63	15056	4247	6800	62.45588235	0.544117647
63	15055	4234	6800	62.26470588	0.735294118
63	15049	4220	6800	62.05882353	0.941176471
62	15019	4204	6800	61.82352941	0.176470588

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
62	15016	4188	6800	61.58823529	0.411764706
62	15011	4172	6800	61.35294118	0.647058824
62	15032	4159	6800	61.16176471	0.838235294
61	15014	4145	6800	60.95588235	0.044117647
61	15003	4129	6800	60.72058824	0.279411765
61	15001	4109	6800	60.42647059	0.573529412
61	15000	4094	6800	60.20588235	0.794117647
60	15014	4079	6800	59.98529412	0.014705882
60	15013	4066	6800	59.79411765	0.205882353
60	15011	4053	6800	59.60294118	0.397058824
60	15011	4039	6800	59.39705882	0.602941176
60	15006	4026	6800	59.20588235	0.794117647
60	15004	4013	6800	59.01470588	0.985294118
59	15001	4000	6800	58.82352941	0.176470588
59	14968	3986	6800	58.61764706	0.382352941
59	14966	3970	6800	58.38235294	0.617647059
59	14964	3953	6800	58.13235294	0.867647059
58	14960	3937	6800	57.89705882	0.102941176
58	14958	3921	6800	57.66176471	0.338235294
58	14970	3905	6800	57.42647059	0.573529412
58	14974	3891	6800	57.22058824	0.779411765
58	14974	3878	6800	57.02941176	0.970588235
57	14974	3865	6800	56.83823529	0.161764706

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
57	14972	3851	6800	56.63235294	0.367647059
57	14970	3838	6800	56.44117647	0.558823529
57	14968	3824	6800	56.23529412	0.764705882
57	14966	3811	6800	56.04411765	0.955882353
56	14963	3797	6800	55.83823529	0.161764706
56	14961	3784	6800	55.64705882	0.352941176
56	14960	3770	6800	55.44117647	0.558823529
56	14958	3757	6800	55.25	0.75
56	14956	3743	6800	55.04411765	0.955882353
55	14954	3730	6800	54.85294118	0.147058824
55	14952	3716	6800	54.64705882	0.352941176
55	14950	3703	6800	54.45588235	0.544117647
55	14949	3689	6800	54.25	0.75
55	14945	3676	6800	54.05882353	0.941176471
54	14942	3662	6800	53.85294118	0.147058824
54	14941	3649	6800	53.66176471	0.338235294
54	14939	3635	6800	53.45588235	0.544117647
54	14937	3622	6800	53.26470588	0.735294118
54	14936	3608	6800	53.05882353	0.941176471
53	14932	3594	6800	52.85294118	0.147058824
53	14932	3581	6800	52.66176471	0.338235294
53	14930	3567	6800	52.45588235	0.544117647
53	14929	3554	6800	52.26470588	0.735294118

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
53	14926	3540	6800	52.05882353	0.941176471
52	14926	3527	6800	51.86764706	0.132352941
52	14923	3513	6800	51.66176471	0.338235294
52	14921	3500	6800	51.47058824	0.529411765
52	14921	3486	6800	51.26470588	0.735294118
52	14919	3473	6800	51.07352941	0.926470588
51	14916	3459	6800	50.86764706	0.132352941
51	14915	3446	6800	50.67647059	0.323529412
51	14915	3432	6800	50.47058824	0.529411765
51	14912	3419	6800	50.27941176	0.720588235
51	14910	3405	6800	50.07352941	0.926470588
50	14908	3392	6800	49.88235294	0.117647059
50	14906	3378	6800	49.67647059	0.323529412
50	14905	3364	6800	49.47058824	0.529411765
50	14902	3351	6800	49.27941176	0.720588235
50	14900	3337	6800	49.07352941	0.926470588
49	14898	3323	6800	48.86764706	0.132352941
49	14897	3310	6800	48.67647059	0.323529412
49	14896	3296	6800	48.47058824	0.529411765
49	14895	3283	6800	48.27941176	0.720588235
49	14893	3269	6800	48.07352941	0.926470588
48	14890	3255	6800	47.86764706	0.132352941
48	14890	3242	6800	47.67647059	0.323529412

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
48	14889	3228	6800	47.47058824	0.529411765
48	14885	3235	6800	47.57352941	0.426470588
48	14885	3221	6800	47.36764706	0.632352941
48	14883	3208	6800	47.17647059	0.823529412
47	14882	3194	6800	46.97058824	0.029411765
47	14882	3180	6800	46.76470588	0.235294118
47	14878	3167	6800	46.57352941	0.426470588
47	14876	3153	6800	46.36764706	0.632352941
47	14875	3140	6800	46.17647059	0.823529412
46	14873	3126	6800	45.97058824	0.029411765
46	14871	3112	6800	45.76470588	0.235294118
46	14869	3099	6800	45.57352941	0.426470588
46	14855	3085	6800	45.36764706	0.632352941
46	14848	3069	6800	45.13235294	0.867647059
45	14835	3053	6800	44.89705882	0.102941176
45	14840	3038	6800	44.67647059	0.323529412
45	14832	3022	6800	44.44117647	0.558823529
45	14830	3006	6800	44.20588235	0.794117647
44	14828	2990	6800	43.97058824	0.029411765
44	14824	2974	6800	43.73529412	0.264705882
44	14820	2958	6800	43.5	0.5
44	14814	2942	6800	43.26470588	0.735294118
44	14812	2925	6800	43.01470588	0.985294118

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
43	14813	2910	6800	42.79411765	0.205882353
43	14807	2893	6800	42.54411765	0.455882353
43	14804	2877	6800	42.30882353	0.691176471
43	14801	2861	6800	42.07352941	0.926470588
42	14800	2845	6800	41.83823529	0.161764706
42	14799	2829	6800	41.60294118	0.397058824
42	14792	2813	6800	41.36764706	0.632352941
42	14790	2797	6800	41.13235294	0.867647059
41	14788	2781	6800	40.89705882	0.102941176
41	14785	2765	6800	40.66176471	0.338235294
41	14779	2749	6800	40.42647059	0.573529412
41	14776	2732	6800	40.17647059	0.823529412
40	14777	2716	6800	39.94117647	0.058823529
40	14770	2700	6800	39.70588235	0.294117647
40	14767	2684	6800	39.47058824	0.529411765
40	14763	2668	6800	39.23529412	0.764705882
39	14762	2652	6800	39	0
39	14757	2635	6800	38.75	0.25
39	14753	2619	6800	38.51470588	0.485294118
39	14748	2603	6800	38.27941176	0.720588235
39	14750	2587	6800	38.04411765	0.955882353
38	14747	2571	6800	37.80882353	0.191176471
38	14743	2555	6800	37.57352941	0.426470588

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
38	14738	2539	6800	37.33823529	0.661764706
38	14733	2523	6800	37.10294118	0.897058824
37	14728	2507	6800	36.86764706	0.132352941
37	14725	2491	6800	36.63235294	0.367647059
37	14718	2459	6800	36.16176471	0.838235294
36	14715	2443	6800	35.92647059	0.073529412
36	14712	2427	6800	35.69117647	0.308823529
36	14705	2411	6800	35.45588235	0.544117647
36	14697	2395	6800	35.22058824	0.779411765
35	14689	2379	6800	34.98529412	0.014705882
35	14686	2362	6800	34.73529412	0.264705882
35	14684	2320	6800	34.11764706	0.882352941
34	14671	2304	6800	33.88235294	0.117647059
34	14670	2288	6800	33.64705882	0.352941176
34	14661	2271	6800	33.39705882	0.602941176
34	14666	2255	6800	33.16176471	0.838235294
33	14658	2239	6800	32.92647059	0.073529412
33	14650	2223	6800	32.69117647	0.308823529
33	14643	2206	6800	32.44117647	0.558823529
33	14643	2190	6800	32.20588235	0.794117647
32	14637	2174	6800	31.97058824	0.029411765
32	14606	2157	6800	31.72058824	0.279411765
32	14618	2140	6800	31.47058824	0.529411765

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
32	14621	2124	6800	31.23529412	0.764705882
31	14608	2108	6800	31	0
31	14616	2093	6800	30.77941176	0.220588235
31	14589	2078	6800	30.55882353	0.441176471
31	14596	2062	6800	30.32352941	0.676470588
31	14579	2045	6800	30.07352941	0.926470588
30	14569	2029	6800	29.83823529	0.161764706
30	14570	2013	6800	29.60294118	0.397058824
30	14572	1997	6800	29.36764706	0.632352941
30	14570	1982	6800	29.14705882	0.852941176
29	14556	1967	6800	28.92647059	0.073529412
29	14546	1952	6800	28.70588235	0.294117647
29	14541	1936	6800	28.47058824	0.529411765
29	14543	1920	6800	28.23529412	0.764705882
29	14520	1905	6800	28.01470588	0.985294118
28	14511	1888	6800	27.76470588	0.235294118
28	14503	1872	6800	27.52941176	0.470588235
28	14495	1856	6800	27.29411765	0.705882353
28	14486	1839	6800	27.04411765	0.955882353
27	14482	1823	6800	26.80882353	0.191176471
27	14474	1808	6800	26.58823529	0.411764706
27	14463	1792	6800	26.35294118	0.647058824
27	14457	1776	6800	26.11764706	0.882352941

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
26	14450	1760	6800	25.88235294	0.117647059
26	14442	1744	6800	25.64705882	0.352941176
26	14432	1729	6800	25.42647059	0.573529412
26	14427	1713	6800	25.19117647	0.808823529
25	14414	1697	6800	24.95588235	0.044117647
25	14407	1681	6800	24.72058824	0.279411765
25	14393	1666	6800	24.5	0.5
25	14387	1650	6800	24.26470588	0.735294118
25	14380	1634	6800	24.02941176	0.970588235
24	14367	1619	6800	23.80882353	0.191176471
24	14357	1603	6800	23.57352941	0.426470588
21	14665	1412	6800	20.76470588	0.235294118
21	14593	1397	6800	20.54411765	0.455882353
21	14585	1382	6800	20.32352941	0.676470588
21	14559	1367	6800	20.10294118	0.897058824
20	14544	1353	6800	19.89705882	0.102941176
20	14523	1339	6800	19.69117647	0.308823529
20	14505	1326	6800	19.5	0.5
20	14485	1312	6800	19.29411765	0.705882353
20	14467	1298	6800	19.08823529	0.911764706
19	14448	1285	6800	18.89705882	0.102941176
19	14430	1271	6800	18.69117647	0.308823529
19	14412	1257	6800	18.48529412	0.514705882

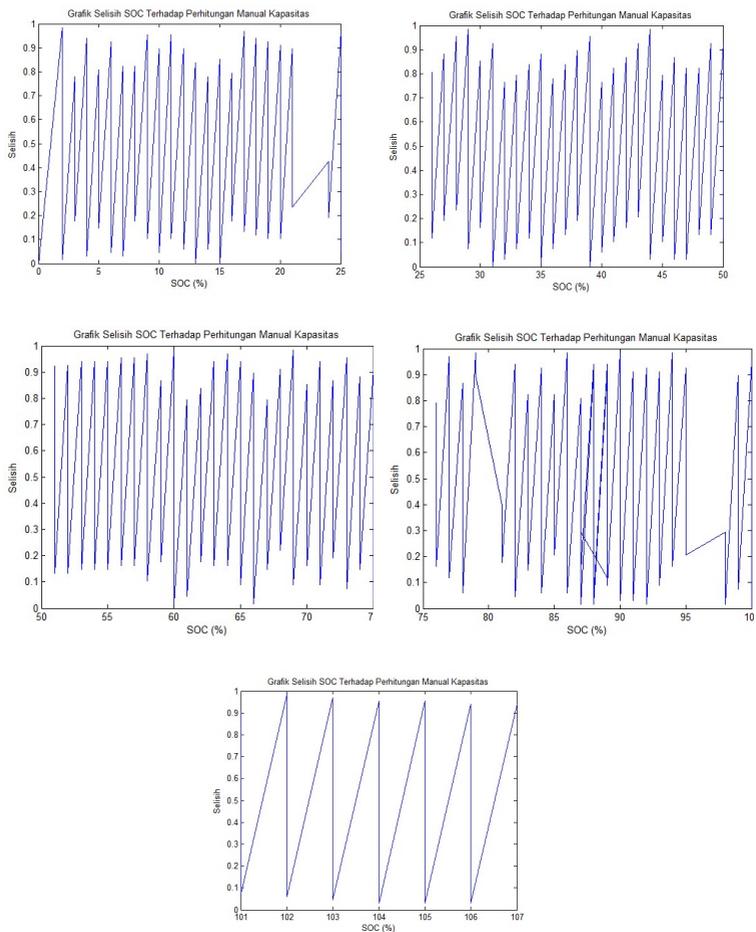
<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
19	14395	1243	6800	18.27941176	0.720588235
19	14377	1229	6800	18.07352941	0.926470588
18	14359	1216	6800	17.88235294	0.117647059
18	14344	1202	6800	17.67647059	0.323529412
18	14326	1188	6800	17.47058824	0.529411765
18	14311	1174	6800	17.26470588	0.735294118
18	14296	1174	6800	17.26470588	0.735294118
18	14280	1160	6800	17.05882353	0.941176471
17	14264	1147	6800	16.86764706	0.132352941
17	14248	1133	6800	16.66176471	0.338235294
17	14232	1118	6800	16.44117647	0.558823529
17	14218	1104	6800	16.23529412	0.764705882
17	14201	1090	6800	16.02941176	0.970588235
16	14188	1076	6800	15.82352941	0.176470588
16	14175	1062	6800	15.61764706	0.382352941
16	14163	1048	6800	15.41176471	0.588235294
16	14152	1034	6800	15.20588235	0.794117647
15	14139	1020	6800	15	0
15	14130	1005	6800	14.77941176	0.220588235
15	14121	991	6800	14.57352941	0.426470588
15	14112	977	6800	14.36764706	0.632352941
15	14104	962	6800	14.14705882	0.852941176
14	14094	948	6800	13.94117647	0.058823529

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
14	14087	934	6800	13.73529412	0.264705882
14	14080	927	6800	13.63235294	0.367647059
14	14074	913	6800	13.42647059	0.573529412
14	14067	899	6800	13.22058824	0.779411765
13	14060	884	6800	13	0
13	14052	870	6800	12.79411765	0.205882353
13	14046	856	6800	12.58823529	0.411764706
13	14039	841	6800	12.36764706	0.632352941
13	14030	827	6800	12.16176471	0.838235294
12	14024	812	6800	11.94117647	0.058823529
12	14015	798	6800	11.73529412	0.264705882
12	14010	784	6800	11.52941176	0.470588235
12	14001	769	6800	11.30882353	0.691176471
12	13996	755	6800	11.10294118	0.897058824
11	13989	741	6800	10.89705882	0.102941176
11	13983	726	6800	10.67647059	0.323529412
11	13975	712	6800	10.47058824	0.529411765
11	13968	697	6800	10.25	0.75
11	13962	683	6800	10.04411765	0.955882353
10	13954	677	6800	9.955882353	0.044117647
10	13945	663	6800	9.75	0.25
10	13941	648	6800	9.529411765	0.470588235
10	13933	634	6800	9.323529412	0.676470588

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
10	13926	619	6800	9.102941176	0.897058824
9	13916	605	6800	8.897058824	0.102941176
9	13909	590	6800	8.676470588	0.323529412
9	13901	576	6800	8.470588235	0.529411765
9	13893	561	6800	8.25	0.75
9	13885	547	6800	8.044117647	0.955882353
8	13877	532	6800	7.823529412	0.176470588
8	13868	518	6800	7.617647059	0.382352941
8	13857	503	6800	7.397058824	0.602941176
8	13844	488	6800	7.176470588	0.823529412
7	13833	474	6800	6.970588235	0.029411765
7	13822	459	6800	6.75	0.25
7	13807	445	6800	6.544117647	0.455882353
7	13789	435	6800	6.397058824	0.602941176
7	13770	420	6800	6.176470588	0.823529412
6	13747	405	6800	5.955882353	0.044117647
6	13721	390	6800	5.735294118	0.264705882
6	13690	375	6800	5.514705882	0.485294118
6	13658	360	6800	5.294117647	0.705882353
6	13620	345	6800	5.073529412	0.926470588
5	13580	330	6800	4.852941176	0.147058824
5	13535	315	6800	4.632352941	0.367647059
5	13486	300	6800	4.411764706	0.588235294

<b>SoC (%)</b>	<b>Tegangan (mV)</b>	<b>Remaining Capacity (mAh)</b>	<b>Desain Capacity (mAh)</b>	<b>Kapasitas (%)</b>	<b>Selisih (%)</b>
5	13435	285	6800	4.191176471	0.808823529
4	13378	270	6800	3.970588235	0.029411765
4	13320	255	6800	3.75	0.25
4	13254	239	6800	3.514705882	0.485294118
4	13184	224	6800	3.294117647	0.705882353
4	13110	208	6800	3.058823529	0.941176471
3	13028	192	6800	2.823529412	0.176470588
3	12938	151	6800	2.220588235	0.779411765
2	12839	135	6800	1.985294118	0.014705882
2	12722	119	6800	1.75	0.25
2	12591	103	6800	1.514705882	0.485294118
2	12428	86	6800	1.264705882	0.735294118
2	12232	69	6800	1.014705882	0.985294118
0	11814	0	6800	0	0
0	11910	0	6800	0	0
0	11952	0	6800	0	0

### Lampiran 13 : Selisih SoC Terhadap Perhitungan Manual Kapasitas



**-----Halaman ini sengaja dikosongkan-----**

## RIWAYAT PENULIS



Ahmad Nur Riza, lahir di Kota Tuban pada tanggal 6 Juli 1997. Anak pertama dari dua bersaudara, bertempat tinggal di Desa Sidokumpul RT 4 RW 5 Kecamatan Bangilan Kabupaten Tuban, Jawa Timur. Pernah menempuh pendidikan di MI Islamiyah Kedungharjo, SMPN 1 Bangilan, dan MAN 1 Bojonegoro. Saat ini sedang menempuh jenjang pendidikan di Institut Teknologi Sepuluh Nopember dengan Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi. Mempunyai pikiran yang realistis. Kegiatan yang disukai adalah jalan-jalan di alam bebas. Memiliki cita-cita untuk menjadi orang yang bisa bermanfaat bagi orang lain.

*E-mail* : rizaahmadnur@gmail.com