

Analisis Efisiensi Sistem Smart Charging Eksternal untuk Laptop

Ahmad Afief Amrullah^{*1}, Arief Setyanto², Sudarmawan³

^{1,2,3} Program Pascasarjana Teknik Informatika Universitas Amikom Yogyakarta
E-mail: ^{*1}aseptilena@gmail.com, ²arief_s@amikom.ac.id, ³sudarmawan@amikom.ac.id

Abstrak

Smart charging dalam penelitian-penelitian terdahulu terkait dengan piranti seperti laptop lebih banyak fokus pada persoalan usia baterai dan kerusakannya. Di sisi lain, terdapat penelitian yang memperlebar konteks smart charging kepada kendali berdasarkan kondisi kelistrikan dengan indikator berupa frekuensi. Akan tetapi, efisiensi belum jelas antara model sistem per piranti dibandingkan terpusat. Oleh karena itu penulis hendak menginvestigasi dan fokus pada efisiensi tersebut. Dalam penelitian berjudul “Analisis Efisiensi Sistem Smart Charging Eksternal untuk Laptop” ini penulis menggunakan perangkat dan data-data yang relatif serupa dengan penelitian terdahulu, namun dengan pendekatan dan teknologi yang relatif berbeda. Data diolah dengan beberapa filter sehingga data cukup optimal dipergunakan, khususnya terkait dengan persentase, waktu dan laju pengisian. Data kemudian dipergunakan untuk menghitung nilai efisiensi yang dicari. Adapun data lain yang berkaitan dipergunakan untuk melihat performa dan kecenderungan sistem. Berdasarkan data yang diperoleh dari hasil analisis, disimpulkan bahwa sistem dengan skema individual relatif lebih efisien dibandingkan dengan skema terpusat. Adapun dalam hal performa, sistem dengan skema individual juga relatif lebih baik, karena tidak ditemukan galat selama percobaan.

Kata Kunci— *smart charging, laptop, frekuensi*

1. PENDAHULUAN

Konsep Smart Charging menjadi topik menarik sebagaimana dipaparkan dalam banyak penelitian bahwa dengan penerapan sistem ini memungkinkan charger untuk menghentikan proses pengisian daya baterai secara mandiri (terprogram), memberikan pembatasan SoC (State of Charge) baik pada batas minimal maupun maksimal sehingga dapat menghindari overheating, overcharging (pengisian daya yang terus-menerus pada SoC maksimal yang mengakibatkan temperatur tinggi yang berkelanjutan juga penurunan kapasitas baterai), sampai pada kemungkinan ledakan atau terbakarnya baterai serta mampu mengurangi charge cycle untuk memperlambat penurunan kapasitas baterai, sehingga baterai lebih awet [1]–[3].

Dalam penelitian yang dilakukan oleh Hild dkk. (2013) yang berjudul “Smart Charging Technologies for Portable Electronic Devices”, mereka mengembangkan dua model sistem Smart Charging, pertama adalah Smart Charging berbasis hardware (microcontroller) dan kedua Smart Charging berbasis software [4]–[6].

Kedua model dalam penelitian tersebut disimpulkan dapat dijalankan dengan baik. Namun demikian, Hild dkk. (2013), mencantumkan saran bahwa model smart charging yang disematkan per perangkat (individual/standalone) sebagaimana dua model tersebut pada dasarnya belum tentu merupakan yang terbaik dan paling efisien. Sehingga dimungkinkan sebuah sistem smart charging yang terpusat di mana seluruh perangkat individual yang terhubung dengan jaringan listrik dapat dikendalikan menggunakan algoritma tertentu, boleh jadi model semacam ini lebih efisien [4], [7], [8].

Penelitian Hild dkk. (2013), khususnya terkait dengan sistem smart charging yang lebih efisien telah ditelaah pada beberapa penelitian namun penelitian-penelitian tersebut lebih bersifat

memperluas cakupan objek [4], [9], [10]. Seperti Ghai dkk. (2013) yang menggunakan topologi “DC Picogrid” yakni dengan menambahkan penyimpanan energi secara lokal ke setiap alat DC dan mengelolanya secara cerdas, Al-Mashhadany & Attia (2015) yang mengusulkan desain baru untuk pengisi daya bertenaga surya pada perangkat berdaya rendah dengan hasil dari simulasi dan eksperimen yang menunjukkan kelayakan desain yang memadai untuk implementasi praktis, juga Moghadam (2015) dan Lee dkk. (2016) pada Home Energy Management System [4], [11], [12].

Merujuk pada pentingnya konsep Smart Charging dari berbagai penelitian di atas, serta upaya mengimplementasikan saran penelitian dari Hild dkk. (2013), maka peneliti mengajukan penelitian dengan judul “Analisis Efisiensi Sistem Smart Charging Eksternal untuk Laptop” [13], [14]. Penelitian ini memilih piranti laptop karena selain melanjutkan penelitian Hild dkk. (2013) juga mengacu pada penelitian Tchuisseu dkk. (2017) yang menjelaskan bahwa Algoritma DDC (Dynamic Demand Control) dapat diimplementasikan sebagai kendali eksternal atau terintegrasi dalam peralatan listrik yang dapat diisi ulang seperti laptop atau telepon seluler, atau dengan peralatan untuk aplikasi termal/suhu seperti AC, lemari es, ketel listrik. Perangkat-perangkat ini terkadang dapat menunda atau memajukan operasinya beberapa menit atau bahkan jam tanpa mengganggu kenyamanan pengguna [15]–[17].

Sedang penelitian ini menggunakan frekuensi listrik sebagai acuan smart charging masih berdasar pada penelitian Tchuisseu dkk. (2017), yang menyatakan bahwa jika frekuensi berkurang maka biaya meningkat, menyebabkan pelanggan untuk mematikan peralatan. Sebaliknya, jika frekuensi meningkat maka biaya menurun, mendorong pelanggan untuk menyalakan peralatan [15], [18], [19]. Akhirnya, daripada menggunakan biaya sebagai masukan pada Demand Side Management, cukup hanya menggunakan frekuensi yang universal dan tersedia dengan mudah di titik manapun pada jaringan listrik (Tchawou Tchuisseu dkk., 2019). Ini merupakan ide di balik Dynamic Demand Control (DDC) yang menggunakan algoritma yang terintegrasi pada peralatan seperti HVAC untuk meregulasi konsumsi energi berdasar pada frekuensi instan [15], [20]. Satu keuntungan dari kendali ini adalah koneksi langsung dengan jaringan listrik [15], [21], [22].

Dengan keberhasilan penelitian ini, dapat memberikan manfaat diantaranya memberikan solusi yang efisien dan efektif atas permasalahan smart charging pada piranti elektronik portabel, khususnya laptop. Memberikan peluang pengembangan riset dan produk sistem smart charging untuk piranti elektronik portabel, khususnya laptop di Indonesia. Dan memberikan wawasan terkait potensi pengembangan sistem smart charging pada ruang lingkup smart grid.

2. METODE PENELITIAN

2.1. Jenis, Sifat, dan Pendekatan Penelitian

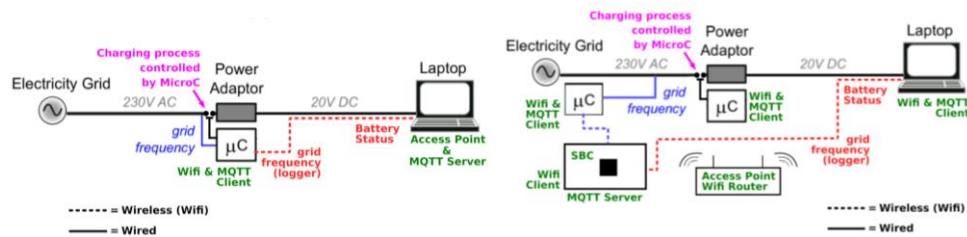
Jenis dari penelitian ini adalah eksperimental, di mana terdapat skema pengujian dari prototipe sistem yang dikembangkan. Penelitian ini bersifat deskriptif dengan pendekatan kuantitatif karena terdapat data-data kuantitatif seperti persentase pengisian baterai dan nilai nominal frekuensi jaringan listrik yang berbentuk angka namun dengan hasil akhir yang bersifat deskriptif.

Adapun mengenai metode eksperimen, dikutip oleh [9] pada bukunya yang berjudul “Metode Penelitian Kuantitatif Kualitatif dan R&D” mengemukakan bahwa secara umum metode penelitian diartikan sebagai cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu dan menurut Arikunto (2010) dalam [9] pada bukunya berjudul “Prosedur Penelitian Suatu Pendekatan Praktik”, Revisi ke-V, metode penelitian adalah cara yang digunakan oleh peneliti dalam mengumpulkan data penelitiannya. Eksperimen menurut Sugiyono (2008) dalam [9] adalah suatu penelitian yang digunakan untuk mencari perlakuan tertentu terhadap yang lain

dalam kondisi yang terkendalikan dan menurut Arikunto (2010) dalam [9] adalah suatu cara untuk mencari hubungan sebab-akibat antara dua faktor yang sengaja ditimbulkan oleh peneliti dengan mengurangi atau menyisihkan faktor-faktor lain yang mengganggu.

2.2. Metode Pengumpulan Data

Pada penelitian ini, data dikumpulkan dengan menggunakan instrumen yang terdiri dari beberapa komponen. Adapun menurut Arikunto (2010) dalam [9] instrumen adalah alat ukur yang dipergunakan untuk memperoleh data penelitian. Instrumen pada penelitian ini terdiri dari 2 jenis komponen, yaitu hardware dan software. Selain itu, penulis menggunakan 2 model skema sistem dalam penelitian ini, yaitu skema I: individual/standalone dan skema II: terpusat.



Gambar 1. Model prototipe skema I (individual/standalone) & skema II (terpusat/centralized)

Secara prosedural, terkait software, pengumpulan data dilakukan dengan beberapa bagian program, yaitu 1) program yang tertanam (*embedded program*) pada hardware modul mikrokontroler ESP-8266, 2) program pada server (khusus skema II) dan 3) program pada laptop (data logger). Bagan alir dari masing-masing program terlampir. Adapun algoritma kendali utama, penulis paparkan pada sub-bab Alur Penelitian.

2.3. Metode Analisis Data

Analisis data dilakukan menggunakan 2 bentuk pendekatan, yaitu 1) matematis non-visual dengan rumus dan 2) matematis visual dengan grafik. Pendekatan (1) menggunakan rumus yang menghitung perbandingan laju pengisian baterai dan pengosongannya. Kinerja keseluruhan dari setiap teknik pengisian pintar untuk perangkat elektronik sangat tergantung pada kapasitas baterai dan tarif muatan dan debitnya [4]. Hild dkk. (2013) mendefinisikan rasio laju pengisian dan pengosongan menjadi:

$$R = \text{Laju Pengisian} / \text{Laju Pengosongan}$$

atau

$$R = \text{Rate of Charge} / \text{Rate of Discharge}$$

Tabel 1. Daftar laju (titik-titik persentase per detik) dari pengisian R_{Charge} dan pengosongan $R_{\text{Discharge}}$, dan rasio di antara keduanya [4]

Laptop	R_{Charge}	$R_{\text{Discharge}}$	R
Laptop 1 (Dell E5520)	0.0213	0.0115	1.8522
Laptop 2 (Macbook Pro®)	0.0128	0.0056	2.2857

Adapun nilai laju baik pengisian maupun pengosongan dapat diperoleh dengan menghitung perubahan persentase baterai laptop dalam rentang waktu tertentu, dengan kata lain,

$$\text{Laju Pengisian} = \text{Total \% Naik} / \text{Total Waktu Dibutuhkan}$$

$$\text{Laju Pengosongan} = \text{Total \% Turun} / \text{Total Waktu Dibutuhkan}$$

Terkait dengan efisiensi sistem, dalam hal pengisian dan pengosongan baterai pada masing-masing skema, dibutuhkan nilai kenaikan SoC yang menyamai penurunannya. Hal ini mengacu pada teori efisiensi dalam buku “Leiden’s Handbook of Batteries” oleh Reddy & Linden (2008) di mana efisiensi dapat diformulasikan menjadi:

$$\text{Efisiensi Pengisian Daya} = \frac{\text{Energi yang Dipakai}}{\text{Energi Pengembalian SoC}}$$

Jika nilai SoC sudah setimbang, maka yang pasti berbeda adalah waktu yang dibutuhkan untuk mencapai kesetimbangan. Maka, efisiensi dapat diformulasikan menjadi:

$$\text{Efisiensi Pengisian Daya} = \frac{\text{Waktu Penurunan SoC}}{\text{Waktu Pengembalian SoC}}$$

Metode khusus untuk menghitung efisiensi ini terdiri dari beberapa tahap, yaitu:

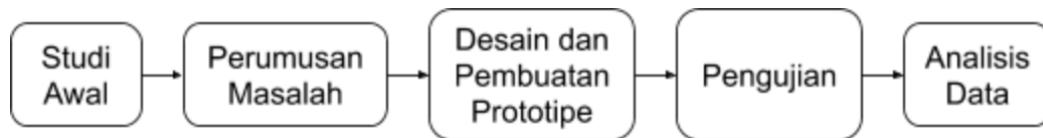
a. menyamakan nilai persentase pengisian dan pengosongan, dengan cara mencuplik sebagian besar data dan mengurangi jumlah baris data yang dipergunakan, hingga didapatkan nilai persentase yang sama.

b. menghitung total waktu yang dibutuhkan untuk pengisian dan pengosongan.

total waktu yang diperoleh kemudian dipergunakan pada formula di atas.

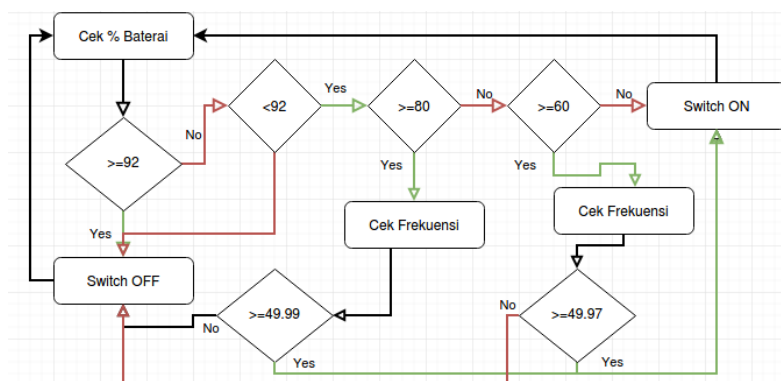
Alur Penelitian

Secara umum alur penelitian eksperimental ini dapat digambarkan sebagai berikut.



Gambar 2. Alur penelitian

Algoritma dari pengendalian charger (ON-OFF) ditunjukkan oleh gambar 5 berikut.



Gambar 3. Algoritma kendali ON-OFF atau switching charger

2.4. Studi Awal

Pada tahap studi awal, penulis telah melakukan pencarian, pengumpulan serta klasifikasi terkait karya tulis ilmiah dan pekerjaan-pekerjaan yang telah dilakukan oleh para akademisi dan peneliti sebelumnya. Adapun topik atau tema yang menjadi acuan utama penulis adalah terkait dengan kata “*smart charging*” atau “*smart charge*” dan “laptop”.

2.5. Perumusan Masalah

Setelah penulis melakukan studi awal terkait topik “*smart charging*” hingga pada tahap akhir melakukan klasifikasi terhadap pekerjaan-pekerjaan dan karya tulis ilmiah terkait, penulis menemukan bahwa terdapat sebuah usulan pekerjaan lanjutan dari penelitian Hild dkk. (2013) yang belum pernah terjamah, pun pada penelitian-penelitian yang secara langsung merujuk pada Hild dkk. (2013). Hal ini terkait dengan ketidakjelasan efisiensi antara sistem *smart charging* dengan model individual atau *standalone* dan sistem dengan model terpusat.

2.6. Desain dan Pembuatan Prototipe

Pada tahapan desain dan pembuatan prototipe, penulis melakukan beberapa modifikasi dan penyesuaian yang mengacu pada pekerjaan-pekerjaan dan karya tulis terdahulu.

2.7. Pengujian

Pada tahapan ini, penulis melakukan 2 jenis pengujian terhadap sistem yang diusulkan, yaitu 1) pengujian fungsional dan 2) pengujian algoritma.

2.8. Pengujian Fungsional

Pengujian fungsional penulis lakukan dengan menjalankan program sederhana yang menjadi basis dari program untuk penerapan algoritma sistem yang diusulkan. Penulis mendapatkan 2 program yang dapat memberikan verifikasi fungsional untuk pengujian algoritma. Program pertama penulis dapatkan dari akun instructables.com yang beralamatkan pada URL <https://www.instructables.com/member/padeath/>. Program ini mencakup pemanfaatan protokol MQTT untuk mengalihkan (*switch*) saklar berupa relay berdasarkan *string* dari *payload* pesan yang dikirimkan ke modul ESP8266. Program kedua penulis dapatkan dari Bertrik Sikken dan Peter Hazenberg yang mempublikasikan pekerjaan mereka secara online pada alamat URL <https://revspace.nl/MainsFrequency>. Program kedua ini mencakup pemanfaatan protokol MQTT untuk mengirimkan informasi dari sensor frekuensi jaringan listrik melalui modul wifi *transceiver* ESP8266 yang dihubungkan dengan server MQTT online atau *cloud*. Berdasarkan data yang diperoleh, penulis bisa memastikan bahwa sistem yang diusulkan dapat berfungsi sebagaimana yang dibutuhkan. Adapun sampel dari data pengujian fungsi dapat dilihat pada tabel 2 dari total 25385 baris data untuk frekuensi dan gambar 6 untuk *switching*.

Tabel 2. Data pengujian pembacaan frekuensi jaringan listrik

No	MF	Datetime
1	49.96	2019-11-19 Selasa 22:32:18:459906476 WIB

Keterangan: MF=*Mains Frequency* / Frekuensi Jaringan Listrik.

Gambar 4. Pengujian fungsi switching

2.9. Pengujian Algoritma

Penulis melakukan beberapa tahap perlakuan pada prototipe sistem yang diusulkan untuk pengujian algoritma. Pengujian ini mencakup 1) representasi *delay* pada jaringan wireless prototipe sistem yang diusulkan dan 2) data pengisian baterai serta frekuensi jaringan listrik untuk masing-masing prototipe (baik model *standalone* maupun model terpusat).

2.10. Representasi Delay pada Jaringan Sistem

Untuk merepresentasikan *delay* pada jaringan wifi lokal, penulis menggunakan perintah ping untuk mengukur *delay* maksimal yang mungkin terjadi. Berikut cuplikan dari hasil perintah ping yang penulis lakukan dengan durasi lebih dari 3 menit. Hasil ping dapat dilihat pada gambar 7 dan gambar 8.

```
--- 192.168.100.1 ping statistics ---
411 packets transmitted, 411 received, 0% packet loss, time 411606ms
rtt min/avg/max/mdev = 0.810/11.068/801.839/60.984 ms|
```

Gambar 5. Hasil perintah ping I pada jaringan skema I

```
--- 192.168.12.93 ping statistics ---
380 packets transmitted, 380 received, 0% packet loss, time 379421ms
rtt min/avg/max/mdev = 1.027/8.843/967.557/56.500 ms
```

Gambar 6. Hasil perintah ping II pada jaringan skema I

Berdasarkan dua kali percobaan dengan perintah ping tersebut, penulis mendapatkan informasi bahwa *delay* jaringan maksimal yang mungkin terjadi adalah lebih kurang 1 detik. Oleh karena itu, penulis menggunakan acuan waktu 1 detik atau 1000 milidetik sebagai representasi *delay* yang sesuai.

Pengujian untuk representasi *delay* ini penulis lakukan dengan menambahkan waktu delay pada algoritma *switching* program yang dijalankan pada modul ESP8266. Adapun skema yang penulis gunakan adalah skema I (*standalone*). Penambahan ini lebih tepatnya pada posisi tepat sebelum instruksi perubahan nilai “RELAY_PIN” atau saklar pemutus charger baik dari “LOW (ON)” ke “HIGH (OFF)” maupun dari “HIGH (OFF)” ke “LOW (ON)”. Sampel data dari pengujian dengan penambahan delay ini terdapat pada tabel 3. Adapun tabel 3 menunjukkan bahwa setiap terjadi kenaikan maupun penurunan persentase baterai, penundaan atau delay sebesar 1000 milidetik atau 1 detik tidak berpengaruh signifikan terhadap pengisian daya. Hal ini

terlihat dari tidak adanya perbedaan persentase dari peralihan (*switching*) charger baik dari status *charging* ke *discharging* di mana didahului oleh nilai 0 pada kondisi charger maupun dari *discharging* ke *charging* di mana didahului oleh nilai 1 pada kondisi charger.

Tabel 3. Data pengujian algoritma sistem (representasi delay jaringan)

No	SCh	SS	SCp
18	0	Discharging	88
19	0	Discharging	88
20	1	Charging	88
21	1	Charging	88
22	1	Charging	88
23	0	Unknown	88

Keterangan: SCh (SysFS_Charger), SS (SysFS_Status), SCp (SysFS_Capacity(RSoC)).

2.11. Data Pengisian Baterai dan Frekuensi Jaringan Model Sistem

Sebagai acuan penulis melakukan pengujian terhadap algoritma pada skema model I (*standalone*). Lebih dari 50 kolom terdapat pada data yang dihasilkan untuk pengujian. Sampel data pengujian algoritma dapat dilihat pada tabel 4.

Tabel 4. Data pengujian algoritma sistem model I (*standalone*)

No	SCh	SC	SV	SCr	CPow	ABa	ATh	ADC	AFC	A%	IB	Ich	MF	SD
1	0	90	12260 000	1232 000	151043 20	Discha rging	48.5	5405 mAh	3322 mAh	61	91	87	49.98	2020-01-10 Jumat 20:15:28 WIB

Keterangan: No (Row_Number), SCh (SysFS-AC-Online (Charger)), SysFS-Capacity (RSoC), SysFS-Voltage_Now, SysFS-Current_Now, Calculated_Power_Now, ACPI-Battery-Status, ACPI-Thermal-degrees_C, ACPI-Details-Design_Capacity, ACPI-Details-Last_Full_Capacity, ACPI-Details-Last_Full_Capacity_%, IBAM_BIOS_%, IBAM_Charge_%, Mains Frequency (Sensor), System Datetime.

3. HASIL DAN PEMBAHASAN

Penulis melakukan pengumpulan data selama lebih kurang 8 jam 20 menit 9 detik yang ditunjukkan oleh tabel 4.5. Waktu mulai dan akhir dengan format standar U.S. (Amerika Serikat) dari pengumpulan data dapat dilihat sampelnya pada tabel 4.4.

Tabel 5. Periode pengujian sistem dalam format standar U.S.

START	END	Period
1/10/2020 8:15:28 PM	1/11/2020 4:35:37 AM	8:20:0 9

Berdasarkan data yang diperoleh, penulis melakukan perhitungan terhadap kenaikan dan penurunan persentase baterai (selanjutnya disebut SoC). Dari data yang telah ditampilkan sampelnya pada tabel 4, penulis menghitung selisih dari setiap baris data terhadap data pada 1 baris sebelumnya. Jika nilai SoC 1 baris data lebih besar dari baris sebelumnya, maka dimunculkan selisihnya bernilai positif. Kemudian apabila lebih kecil dari baris sebelumnya, maka dimunculkan selisihnya bernilai negatif. Adapun jika nilai nya sama, maka nilai yang

muncul pasti 0 (nol). Karena perhitungan yang penulis lakukan adalah per baris dengan bergantung pada 1 baris sebelumnya, maka hasil nilai positif pasti bernilai 1 (satu) dan hasil nilai negatif pasti bernilai -1 (minus satu).

Selain itu, penulis juga menghitung *time gap* (jarak waktu) dari setiap kenaikan maupun penurunan SoC. Nilai jarak waktu yang masih dalam format “jam:menit:detik” penulis konversi menjadi waktu dalam satuan detik, sehingga dapat memudahkan perhitungan maupun analisis lainnya. Setelah itu, dengan jarak waktu yang telah dikonversi tersebut, penulis menghitung kecepatan pengisian daya (*charging*) dengan membagi kenaikan dengan waktu yang dibutuhkan (menggunakan jarak waktu). Tabel 6 menunjukkan sampel dari hasil perhitungan yang penulis lakukan.

Tabel 6. Sampel hasil perhitungan perubahan SoC, jarak waktu dan kecepatan (skema I)

Row Number	SysFS AC Adapter	SysFS SoC	Mains Frequency	Time	SoC Change	Time Gap	Time Gap (Seconds)	Speed
64	0	89	49.95	20:16:53	-1	0:01:25	85	-0.012

Berdasarkan data tersebut, penulis melakukan perhitungan kenaikan dan penurunan SoC dengan menjumlahkan setiap perubahan dari awal hingga akhir. Tabel 7 menunjukkan hasil penjumlahan dari kenaikan SoC dan penurunan SoC untuk kedua skema.

Tabel 7. Hasil perhitungan (*charged %*) dan (*discharged %*) dari kedua skema sistem

SKEMA I		SKEMA II	
Charged %	Discharged %	Charged %	Discharged %
685	693	239	254

Untuk mendapatkan nilai yang setimbang antara kenaikan dan penurunan SoC, penulis melakukan penjumlahan perubahan SoC baik dari kenaikan maupun penurunan. Perubahan waktu setiap kenaikan dan penurunan juga penulis jumlahkan. Parameter penjumlahan ini adalah ketika nilai perubahan SoC lebih dari nol (kenaikan) atau kurang dari nol (penurunan). Tabel 8 menunjukkan sampel dari hasil penjumlahan untuk setiap perubahan nilai SoC.

Tabel 8. Sampel hasil penjumlahan perubahan SoC dan jarak waktu berdasarkan perubahan SoC (skema I)

Row_ Number	SysFS_ SoC(%)	Freq. (Hertz)	SoC_ Change (%)	Time Gap (Second s)	SUM_of _SoC_ Change	SUMIF (>0)_of _SoC_ Change	SUMIF (<0)_of _SoC_ Change	SUMIF (SoC_ch ange>0)_of_Tim e_Gaps	SUMIF (SoC_ch ange<0)_of_Tim e_Gaps
126	88	49.98	-1	84	-1	0	-1	0	84

Dengan menggunakan data seperti pada tabel 4.8 dan 4.9 tersebut, penulis melakukan filter terhadap data. Parameter filter ini adalah nilai pada kolom “SUM_of_Change” atau penjumlahan perubahan SoC (gabungan antara kenaikan dan penurunan), di mana data yang diambil adalah ketika nilai pada kolom ini menunjukkan angka 0. Hal ini berarti titik kesetimbangan di mana kenaikan SoC sama dengan penurunannya. Tabel 9 dan tabel 10 menunjukkan hasil filter dari data seperti pada tabel 8.

Tabel 9. Hasil filter data untuk kenaikan SoC = penurunannya (skema I)

Charged & Discharged %	Time Gaps of Charged	Time Gaps of Discharged
------------------------	----------------------	-------------------------

2	174	166
255	2966	1312
...	...	...
681	14560	13506

Tabel 10. Hasil filter data untuk kenaikan SoC = penurunannya (skema II)

Charged & Discharged %	Time Gaps of Charged	Time Gaps of Discharged
1	5	9
4	336	239
...	...	...
237	11875	16361

Berdasarkan tabel 9, pada skema I nilai waktu dari penurunan SoC tidak ada yang melebihi nilai waktu dari kenaikan SoC, ketika terjadi kesetimbangan. Hal ini menunjukkan semua data yang ada dapat dipergunakan untuk perhitungan efisiensi. Namun demikian, hal yang berbeda terlihat pada tabel 10, di mana pada skema II nilai waktu dari penurunan SoC, hampir semua melebihi nilai waktu dari kenaikan SoC. Dapat dikatakan bahwa data pada skema II menunjukkan hal yang tidak wajar, di mana tidak mungkin output melebihi input. Oleh karena itu penulis memilih salah satu baris data yang dapat dinilai wajar atau logis, yaitu baris ke-2 pada tabel 10. Adapun data yang dipilih pada tabel 9 untuk skema I adalah pada baris pertama, karena perbedaan data dibandingkan dengan data terpilih pada tabel 10 adalah yang terkecil yaitu 2 %.

Dengan demikian, data yang penulis penggunaan selanjutnya adalah data pada baris pertama dari tabel 9 dan data pada baris ke-2 dari tabel 10. Data terpilih ditunjukkan dan penulis tegaskan pada tabel 11.

Tabel 11. Data terpilih untuk perhitungan efisiensi

	Charged & Discharged %	Time Gaps of Charged	Time Gaps of Discharged
Skema I	2	174	166
Skema I	4	239	239

Untuk perhitungan efisiensi, nilai pada kolom “Time Gaps of Discharged” dibagi dengan “Time Gaps of Charged”. Dengan demikian didapatkan nilai efisiensi sebesar 0,9540229885 atau 95,4 % (sembilan puluh lima koma empat persen) untuk skema I dan 0,7113095238 atau 71,1 % (tujuh puluh satu koma satu persen) untuk skema II. Data hasil perhitungan ini ditunjukkan pada tabel 12.

Tabel 12. Data hasil perhitungan efisiensi untuk kedua skema sistem

Efisiensi Skema I	Efisiensi Skema II
0.9540229885	0.7113095238
95.4%	71.1%

Dengan demikian dalam hal efisiensi pengisian daya, skema I relatif lebih baik daripada skema II. Adapun dalam hal kecepatan pengisian daya, perhitungan diperlukan dengan menggunakan data “Charged %” dan “Discharged %” pada tabel 4.8. Penulis menghitung total waktu yang dibutuhkan untuk kenaikan dan penurunan SoC secara keseluruhan. Data hasil perhitungan ini ditunjukkan pada tabel 13.

Tabel 13. Hasil perhitungan kecepatan pengisian daya dari dua skema

SKEMA I		SKEMA II	
Charged % 685	Discharged % 693	Charged % 239	Discharged % 254
Total Time	Total Time	Total Time	Total Time
14952	14994	12088	17906
Rate of Charge 0.04581326912787590	Rate of Discharge 0.04621848739495800	Rate of Charge 0.01977167438782260	Rate of Discharge 0.01418518932201500

Berdasarkan data “Rate of Charge” dan “Rate of Discharge”, dapat dihitung rasio yang menunjukkan perbandingan performa dari dua buah laptop yang menjadi objek dalam penelitian ini. Tabel 14 menunjukkan perbandingan rasio pengisian daya dari laptop pada masing-masing skema. Adapun grafik dari kecepatan pengisian daya pada masing-masing skema terlampir.

Tabel 14. Rasio kecepatan pengisian daya dari kedua skema

Ratio (RoC / RoD) SKEMA I	Ratio (RoC / RoD) SKEMA II
0.9912325502	1.393825203

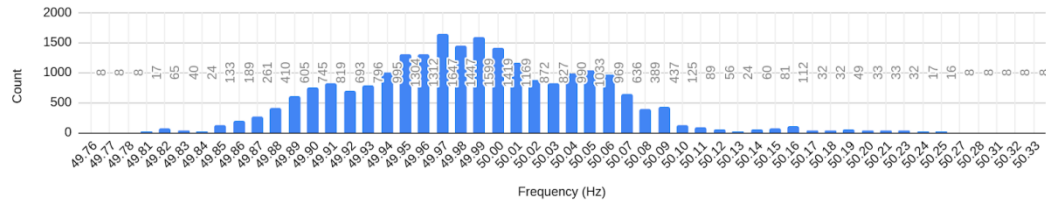
Berdasarkan data dari tabel 14, dapat dikatakan bahwa laptop pada skema I lebih lambat dalam pengisian daya jika dibandingkan dengan laptop pada skema II. Sebagai informasi tambahan, penulis menyertakan beberapa tabel perbandingan dari kedua skema. Di samping efisiensi dan kecepatan pengisian daya, penulis juga menghitung data dari sebaran atau kemunculan pembacaan nilai frekuensi jaringan listrik pada sensor dari masing-masing skema. Hal ini mungkin diperlukan untuk melihat kecenderungan tertentu dari masing-masing skema. Tabel 15 menunjukkan sampel dari data perhitungan kemunculan nilai frekuensi.

Tabel 15. Sampel data perhitungan kemunculan nilai frekuensi

SKEMA I				SKEMA II			
No.	Frequency (Hz)	Count	Min & Max	No.	Frequency (Hz)	Count	Min & Max
1	49.76	8	49.76	1	49.81	43	49.81
2	49.77	8	50.28	2	49.82	44	64.00

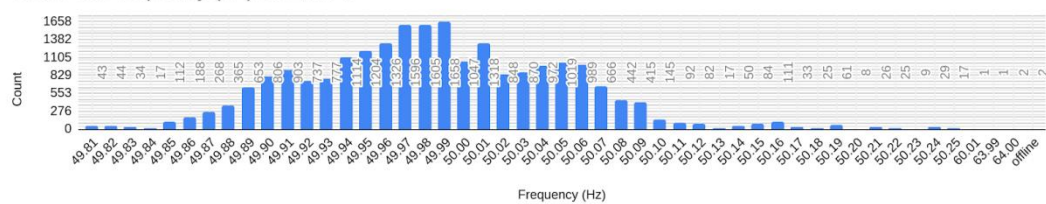
Pada tabel 15, dapat dilihat bahwa terdapat perbedaan dari pembacaan nilai frekuensi pada skema I dan skema I. Skema I menunjukkan pembacaan terendah dari nilai frekuensi pada nilai “49.76” (empat puluh sembilan koma tujuh enam), sementara skema II menunjukkan data tertinggi dari nilai frekuensi yaitu “64”. Namun demikian, angka 64 pada skema II menunjukkan hal yang tidak wajar, karena maksimal nilai frekuensi adalah sekitar 50.2 Hertz. Untuk mengetahui hal-hal tersebut secara lebih jauh, penulis melakukan plot terhadap data perhitungan kemunculan nilai frekuensi ini. Gambar 9 dan gambar 10 menunjukkan detail data dari masing-masing skema.

Count vs. Frequency (Hz) SKEMA I



Gambar 7. Bagan batang dari penghitungan kemunculan nilai frekuensi (SKEMA I)

Count vs. Frequency (Hz) SKEMA II

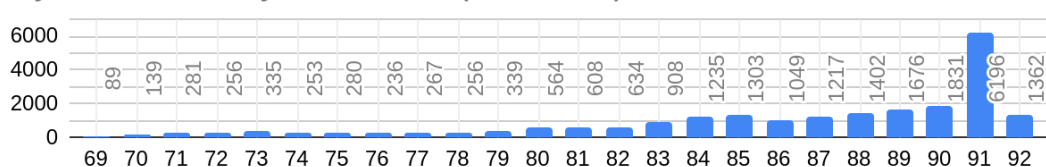


Gambar 8. Bagan batang dari penghitungan kemunculan nilai frekuensi (SKEMA II)

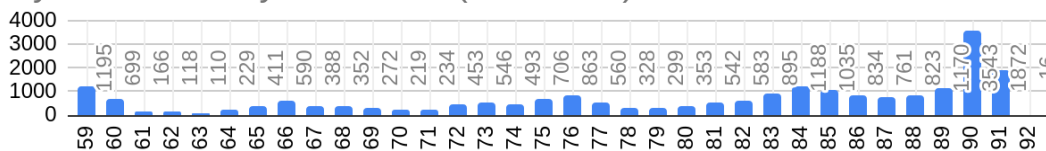
Berdasarkan gambar 9, nilai yang tidak wajar pada skema II dapat dilihat pada data paling kanan, yaitu angka “60.01”, “63.99”, “64.00” dan nilai “offline”. Hal ini menunjukkan bahwa skema II relatif cenderung kurang stabil jika dibandingkan dengan skema I.

Adapun data tambahan terkait dengan data SoC ditunjukkan oleh gambar 11. Nilai SoC dari skema I terlihat lebih cenderung bertahan pada SoC yang tinggi di mana data terbanyak ada pada rentang 84 % hingga 92%. Sementara nilai SoC pada skema II selain pada rentang 84% hingga 92%, tercatat juga nilai SoC rendah cukup banyak terjadi, khususnya pada nilai 59%. Hal ini menunjukkan bahwa kecenderungan skema II adalah juga pada SoC yang rendah.

SysFS % and SysFS Count (SKEMA I)



SysFS % and SysFS Count (SKEMA II)



Gambar 9. Bagan batang dari penghitungan SoC (SKEMA I & SKEMA II)

4. KESIMPULAN

Berdasarkan hasil pengujian serta analisis data yang penulis lakukan, terdapat beberapa hal yang dapat disimpulkan dari penelitian ini. Adapun berikut kesimpulan dari penelitian ini:

1. Sistem dengan Skema Standalone atau Individual relatif lebih efisien daripada sistem dengan Skema Terpusat. Hal ini ditunjukkan oleh efisiensi sebesar 95,4 % (standalone) dan 71,1% (terpusat).
2. Terkait dengan performa hardware dan software, sistem dengan Skema Standalone relatif lebih bisa diandalkan atau reliabel daripada sistem dengan Skema Terpusat.
3. Pada Skema Terpusat, terdapat data yang tidak wajar dalam hal efisiensi, hal ini mungkin dikarenakan faktor dari baterai laptop yang dipergunakan bukan produk asli, melainkan baterai replacement.
4. Pada sistem dengan Skema Terpusat/Centralized, hardware seperti modul ESP8266-01 atau ESP-01 yang belum terintegrasi dengan tombol untuk reset dan penyesuaian tegangan untuk suplai daya, tidak cukup reliabel untuk dipergunakan, karena sulit untuk ditangani pada saat terjadi masalah atau galat insidental.

DAFTAR PUSTAKA

- [1] R. SEPTIAN, "SISTEM APLIKASI SMART CHARGER UNTUK PENGOPTIMALAN PENGISIAN BATERAI LAPTOP." University of Muhammadiyah Malang, 2018.
- [2] S. S. Choi and H. S. Lim, "Factors that affect cycle-life and possible degradation mechanisms of a Li-ion cell based on LiCoO₂," *J. Power Sources*, vol. 111, no. 1, pp. 130–136, 2002.
- [3] W. D. Kelton, R. Sadowski, and N. Zupick, "Simulation with Arena. Fifth." McGraw Hill, 2014.
- [4] S. Hild, S. Leavey, C. Gräf, and B. Sorazu, "Smart charging technologies for portable electronic devices," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 328–336, 2013.
- [5] F. A. Ekoanindiyo, "Pemodelan Sistem Antrian Dengan Menggunakan Simulasi," *Din. Tek. Ind.*, 2011.
- [6] Y. I. Al-Mashhadany and H. A. Attia, "High performance for real portable charger through low power PV system," *Int. J. Sustain. Green Energy*, vol. 4, no. 3–1, pp. 14–18, 2015.
- [7] S. K. Ghai, Z. Charbiwala, S. Mylavaram, D. P. Seetharamakrishnan, and R. Kunnath, "DC picogrids: A case for local energy storage for uninterrupted power to DC appliances," in *Proceedings of the fourth international conference on Future energy systems*, 2013, pp. 27–38.
- [8] R. Gunawan, "Perancangan Alat dan Sistem Smart Charger Pada Smartphone Menggunakan Arduino." Universitas Islam Negeri Alauddin Makassar, 2017.
- [9] H. Hanggoro, S. Surisman, and A. Sitepu, "Pengaruh Circuit Training Dan Interval Training Terhadap Hasil Vo₂max Siswa Ekstrakurikuler Bolabasket," *JUPE (Jurnal Penjaskesrek)*, vol. 4, no. 2, 2016.
- [10] X. He, M. Papa, and R. Gamble, "Extending Over-the-Air Libraries to Secure ESP8266 Updates," in *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2019, pp. 1–6.
- [11] B. Kim, "Smart charging architecture for between a plug-in electrical vehicle (PEV) and a smart home," in *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, 2013, pp. 306–307.
- [12] N. Kumar and K. M. Sundaram, "IoT based smart charger: an ESP8266 based automatic charger," in *Proceedings of the International Conference on Big Data and Internet of Thing*, 2017, pp. 153–157.
- [13] M. A. Pasirulloh and E. Suryani, "Pemodelan Dan Simulasi Sistem Industri Manufaktur Menggunakan Metode Simulasi Hybrid (Studi Kasus: PT. Kelola Mina Laut)," *J. Tek. ITS*, vol. 6, no. 2, pp. A234–A238, 2017.

- [14] A. Prastiantari, “SKOPIN (STOP KONTAK PINTAR) PENGENDALI ARUS LISTRIK MENGGUNAKAN TIMER PADA STOP KONTAK BERBASIS ARDUINO.” UNIVERSITAS NEGERI JAKARTA, 2017.
- [15] E. B. T. Tchuisseu, D. Gomila, D. Brunner, and P. Colet, “Effects of dynamic-demand-control appliances on the power grid frequency,” *Phys. Rev. E*, vol. 96, no. 2, p. 22302, 2017.
- [16] S. J. Lee, H. Yang, J. S. Kim, and S. G. Choi, “Supply and demand management system based on consumption pattern analysis and tariff for cost minimization,” in *2016 18th International Conference on Advanced Communication Technology (ICACT)*, 2016, pp. 652–658.
- [17] A. H. Nasution and Y. Prasetyawan, “Perencanaan dan Pengendalian Produksi, Yogyakarta: Graha Ilmu,” 2008.
- [18] T. S. Prayogo, “TA: Sistem Kontrol Charger Handphone Otomatis Berbasis Android.” Institut Bisnis dan Informatika Stikom Surabaya, 2016.
- [19] A. Pröbstl, P. Kindt, E. Regnath, and S. Chakraborty, “Smart2: Smart charging for smart phones,” in *2015 IEEE 21st International Conference on Embedded and Real-Time Computing Systems and Applications*, 2015, pp. 41–50.
- [20] M. Håberg, “Fundamentals and recent developments in stochastic unit commitment,” *Int. J. Electr. Power Energy Syst.*, vol. 109, pp. 38–48, 2019.
- [21] T. B. Reddy, *Linden’s handbook of batteries*. McGraw-Hill Education, 2011.
- [22] M. Rokonzaman and M. Hossam-E-Haider, “Design of a peripheral interface controller based MPPT charge controller,” in *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, 2016, pp. 1–4.