

**SISTEM INFORMASI MANAJEMEN ENERGI BERBASIS  
INTERNET OF THINGS (IOT) DENGAN METODE RAPID  
APPLICATION DEVELOPMENT (RAD)**

**SKRIPSI**

Diajukan sebagai salah satu syarat  
Untuk memperoleh gelar  
Sarjana Teknik



Oleh:

KEVIN SEBASTIAN ARIEF  
NIM. I0716018

**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS SEBELAS MARET  
SURAKARTA  
2020**

**HALAMAN SURAT PENUGASAN**

**SURAT PERNYATAAN**  
**ORISINALITAS KARYA ILMIAH**

Saya mahasiswa Program Studi Teknik Elektro Universitas Sebelas Maret yang bertanda tangan dibawah ini :

Nama : Kevin Sebastian Arief  
NIM : I0716018  
Judul Tugas Akhir : Sistem Informasi Manajemen Energi berbasis Internet of Things (IoT) dengan metode Rapid Application Development (RAD)

Dengan ini menyatakan bahwa Tugas Akhir atau Skripsi yang saya susun tidak mencontoh atau melakukan plagiat dari karya tulis orang lain. Jika terbukti Tugas Akhir yang saya susun tersebut dinyatakan batal dan gelar sarjana yang saya peroleh dengan sendirinya dibatalkan atau dicabut.

Demikian surat pernyataan ini saya buat dengan sebenarnya dan apabila di kemudian hari terbukti melakukan kebohongan maka saya sanggup menanggung segala konsekuensinya.

Surakarta, Juli 2020

**Kevin Sebastian Arief**  
**NIM. I0716018**

**HALAMAN PENGESAHAN TIM PEMBIMBING DAN TIM PENGUJI  
SISTEM INFORMASI MANAJEMEN ENERGI BERBASIS *INTERNET OF  
THINGS (IOT)* DENGAN METODE *RAPID APPLICATION  
DEVELOPMENT (RAD)***

**Disusun Oleh**

**KEVIN SEBASTIAN ARIEF**

**NIM I0716018**

**Pembimbing 1**

**Pembimbing 2**

**Meiyanto Eko Sulistyo S.T., M.Eng.**  
**NIP. 197705132009121004**

**Agus Ramelan S.Pd., M.T.**  
**NIP. 199203152019031017**

Telah dipertahankan di hadapan Tim Dosen Penguji pada hari....Tanggal.....2020

1. **Meiyanto Eko Sulistyo S.T., M.Eng.**  
NIP. 197705132009121004 .....  
.....
2. **Agus Ramelan S.Pd., M.T.**  
NIP. 199203152019031017 .....  
.....
3. **Muhammad Hamka Ibrahim ST., M.Eng.**  
NIP. 198812292019031011 .....  
.....
4. **Sutrisno S.T., M.Sc, Ph.D.**  
NIP. 198705062019031009 .....  
.....

Mengetahui,

Kepala Prodi Teknik Elektro

Koordinator Tugas Akhir

**Feri Adriyanto, Ph.D.**  
NIP. 196801161999031001

**Muhammad Hamka I., S.T, M.Eng.**  
NIP. 1988122920161001

**SISTEM INFORMASI MANAJEMEN ENERGI BERBASIS INTERNET OF THINGS  
(IOT) DENGAN METODE RAPID APPLICATION DEVELOPMENT (RAD)**

**Kevin Sebastian Arief**

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Sebelas Maret

Email : [kscdh6@gmail.com](mailto:kscdh6@gmail.com)

**ABSTRAK**

Pengembangan Sistem Informasi Manajemen Energi dilakukan untuk mempermudah dalam *monitoring* dan *controlling* suatu alat elektronik yang berada pada suatu tempat. *Monitoring* dilakukan dengan bersifat *real time* dengan data yang diambil dari sebuah *device* dengan cara pengiriman data adalah menggunakan *Internet of Things* (IoT) yaitu dengan mengirimkan data ke dalam *Application Programming Interface* (API) yang ada dalam *Framework Laravel* yang kemudian akan diolah oleh sistem untuk dimasukkan ke dalam *database MySQL*. Sedangkan untuk *controlling* dilakukan juga secara *real time* dengan menekan tombol on/ off yang ada pada sistem informasi ini. Sistem Informasi Manajemen Energi ini dikembangkan dengan metode *Rapid Application Development* (RAD) serta menggunakan *Framework Laravel*. Penggunaan metode ini digunakan supaya pengembangan Sistem Informasi ini dapat dilakukan secara cepat dan terstruktur. Penelitian ini menghasilkan Sistem Informasi bernama *Building Energy Management System* yang menampilkan informasi secara *real time* kepada *user* berupa penggunaan energi dan biayanya, baik total maupun dari tiap ruangan.

Kata kunci : Sistem Informasi, Laravel, MySQL, RAD, IoT, API.

**INTERNET OF THINGS (IOT) BASED ENERGY MANAGEMENT INFORMATION  
SYSTEM WITH RAPID APPLICATION DEVELOPMENT (RAD) METHOD**

**Kevin Sebastian Arief**

Electrical Engineering Study Program, Faculty of Engineering, Universitas Sebelas Maret

Email : [kscdh6@gmail.com](mailto:kscdh6@gmail.com)

**ABSTRACT**

*The Development of Energy Management Information System is carried out to facilitate monitoring and controlling an electronic device that is in someplace. Monitoring is done in real-time with data taken from a device by sending data using Internet of Things, sending data into Application Programming Interface (API) in Laravel Framework which will then be processed by the system to be entered into the MySQL database. Whereas controlling is also done in real-time by pressing the on/ off button on this information system. This Energy Management Information System is developed using the Rapid Application Development (RAD) method and uses the Laravel Framework. The use of this method is used so that the development of this Information System can be done quickly and structured. This research produces Information System called Building Energy Management System that displays real-time information to users in the form of energy usages and costs, both total and from each room.*

*Keywords : Information System, Laravel, MySQL, RAD, IoT, API.*

## KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa, atas berkat dan anugerah-Nya, penulis dapat menyelesaikan penyusunan skripsi dengan judul Sistem Informasi Manajemen Energi berbasis *Internet of Things* (IoT) dengan metode *Rapid Application Development* (RAD) di Fakultas Teknik Universitas Sebelas Maret (UNS) Surakarta. Dalam penyusunan skripsi ini, tentu perlu melewati serangkaian proses. Sebagaimana kita ketahui bahwa dalam berproses tidak selamanya jalan yang dilalui itu mulus dan sesuai dengan harapan kita. Namun oleh karena itulah, kita semakin mendapat banyak pengalaman yang berharga. Pengalaman – pengalaman yang memberikan kita sebuah bekal untuk menjalani kehidupan kita di masa depan.

Dalam penyusunan skripsi ini, empat tahun perkuliahan di Program Studi Teknik Elektro telah penulis lalui. Skripsi bukan tentang menyelesaikan sebuah proyek dan menuliskannya, namun di dalamnya terkandung segala aspek materi dan kebijaksanaan yang telah diasah menjadi lebih runcing seama empat tahun leih perkuliahan di Teknik Elektro. Oleh karena itu, skripsi ini penulis persembahkan bagi segala pihak yang telah menjadi bagian dalam perjalanan empat tahun leih dalam dunia perkuliahan.

Menyelesaikan tugas akhir atau skripsi ini tentu bukanlah perkara mudah. Sebagai wujud apresiasi, melalui kata pengantar ini, penulis hendak menyampaikan ucapan terima kasih kepada :

1. Orang tua dan keluarga tercinta yang selalu memberikan motivasi, semangat, dan doa sehingga penulisan skripsi ini dapat terselesaikan.
2. Bapak Meiyanto Eko Sulistyo S.T., M.Eng. selaku Pembimbing I yang memberikan dukungan, ide, arahan, bimbingan, dan motivasi selama perkuliahan di Teknik Elektro.
3. Bapak Agus Ramelan S.Pd., M.T. selaku Pembimbing II yang selalu memberikan dukungan, ide, arahan, bimbingan, dan motivasi selama penggerjaan skripsi ini.

4. Bapak Subhan selaku staff ICT yang telah memberikan bantuan dan ilmu selama penggerjaan skripsi ini.
5. Seluruh Dosen Program Studi Teknik Elektro yang telah memberikan ilmu yang bermanfaat, motivasi, dan inspirasi yang luar biasa selama menjalani masa perkuliahan selama kurang lebih 4 tahun ini.
6. Yudhi Prabowo Kusuma dan Oki Setiawan yang telah menjadi tester dalam Sistem Informasi *Building Energy Management System* dan memberikan masukan yang sangat bermanfaat.
7. Seluruh teman-teman Teknik Elektro terutama angkatan 2016 yang tidak bisa disebutkan satu persatu yang telah sama-sama berjuang selama 4 tahun ini atas bantuannya yang luar biasa dan semangat yang diberikan untuk segera mendapatkan gelar Sarjana Teknik ini.

Surakarta, Juli 2020

Kevin Sebastian Arief  
I0716018

## DAFTAR ISI

HALAMAN SURAT PENUGASAN .....	ii
SURAT PERNYATAAN ORISINALITAS KARYA ILMIAH .....	iii
HALAMAN PENGESAHAN TIM PEMBIMBING DAN TIM PENGUJI .....	iv
ABSTRAK .....	v
ABSTRACT .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Penelitian terdahulu.....	5
2.2 <i>Internet of Things (IoT)</i> .....	9
2.3 Thingspeak .....	9
2.4 Sistem Informasi .....	10
2.5 <i>Web Application</i> .....	10
2.6 <i>Database Management System (DBMS)</i> .....	10
2.7 <i>Framework</i> .....	12
2.8 Laravel.....	12
2.9 <i>Application Programming Interface (API)</i> .....	13
2.10 PHP .....	13
2.11 <i>Hyper Text Markup Language (HTML)</i> .....	14
2.12 <i>Cascading Style Sheet (CSS)</i> .....	14
2.13 XAMPP .....	16
2.14 MySQL.....	16
2.15 <i>Data Flow Diagram (DFD)</i> .....	18

2.16 Metode Pengembangan .....	19
BAB III METODOLOGI PENELITIAN.....	21
3.1 Metode dan Lokasi Penelitian .....	21
3.2 Sumber Data .....	22
3.3 Metode Pengumpulan Data.....	22
3.4 Instrumen Penelitian .....	22
3.5 Metode Rapid Application Development (RAD) .....	24
3.6 Teknik Pengujian.....	28
BAB IV HASIL DAN PEMBAHASAN .....	31
4.1. Desain dan Implementasi Sistem Manajemen Energi .....	31
4.2. Pengujian Sistem menggunakan metode <i>Black Box</i> .....	43
4.3. Pengujian Sistem menggunakan metode <i>User Acceptance Testing</i> (UAT) .....	50
4.4. Performa Sistem Informasi Building Energy Management System.....	54
BAB V KESIMPULAN DAN SARAN.....	57
5.1. Kesimpulan.....	57
5.2. Saran .....	57
DAFTAR PUSTAKA .....	59
LAMPIRAN .....	61

## DAFTAR GAMBAR

Gambar 2.1 Konsep IoT .....	9
Gambar 2.2 Konsep MVC pada Laravel [21] .....	13
Gambar 2.3 XAMPP Control Panel .....	16
Gambar 3.1 Flowchart Metode Penelitian .....	21
Gambar 3.2 Tahapan Metode RAD .....	24
Gambar 3.3 Entity Relationship Diagram .....	25
Gambar 3.4 DFD Level 0 Sistem Informasi BEMS .....	25
Gambar 3.5 DFD Level 1 Sistem Informasi BEMS .....	26
Gambar 3.6 Business Process Diagram .....	27
Gambar 3.7 Flowchart Sistem Informasi BEMS .....	27
Gambar 4.1 Halaman Dashboard .....	31
Gambar 4.2 (a) Indikator dan (b) Teks “Lost Connection to Hardware”.....	32
Gambar 4.3 Halaman Usages .....	33
Gambar 4.4 Halaman Control .....	34
Gambar 4.5 Mematikan Control Master di lantai 1 .....	35
Gambar 4.6 Halaman Cost .....	35
Gambar 4.7 Tampilan ketika user memilih periode.....	36
Gambar 4.8 Halaman Statistic.....	36
Gambar 4.9 Halaman Settings.....	37
Gambar 4.10 Tampilan ketika user memilih protokol MQTT .....	39
Gambar 4.11 Tampilan ketika user memilih protokol HTTP .....	39
Gambar 4.12 Notifikasi yang akan muncul ketika user salah menginputkan password lama.....	40
Gambar 4.13 Notifikasi ketika user berhasil mengubah password .....	40
Gambar 4.14 Modal Admin List .....	40
Gambar 4.15 Tampilan modal ketika tombol tambah admin baru ditekan.....	41
Gambar 4.16 Tampilan modal ketika tombol edit ditekan.....	42
Gambar 4.17 Tampilan konfirmasi ketika menekan tombol lupa password.....	42
Gambar 4.18 Tampilan konfirmasi ketika menekan tombol delete .....	43
Gambar 4.19 User menginputkan di Modal Login .....	44
Gambar 4.20 Redirect ke Halaman dashboard.....	44
Gambar 4.21 Tampilan notifikasi ketika user salah menginputkan email atau password.....	44
Gambar 4.22 Proses pengujian menampilkan data ke halaman dashboard .....	45
Gambar 4.23 Proses pengujian menampilkan data ke halaman usages .....	45
Gambar 4.24 Proses pengujian menampilkan data ke halaman control.....	46
Gambar 4.25 Proses pengujian menampilkan data ke halaman cost .....	46
Gambar 4.26 Proses pengujian menampilkan data ke halaman statistic.....	47
Gambar 4.27 Proses pengujian menampilkan data ke halaman statistic.....	47
Gambar 4.28 Proses pengujian menampilkan data ke tabel Admin List .....	47
Gambar 4.29 Proses pengujian pembaharuan data admin .....	48
Gambar 4.30 Proses pengujian pengontrol device sebelum control dilakukan ....	48
Gambar 4.31 Proses pengujian pengontrol device setelah control dilakukan.....	49
Gambar 4.32 Menu Logout.....	49

Gambar 4.33 Proses pengujian logout .....	50
Gambar 4.34 Hasil uji pada website GTmetrix.....	55
Gambar 4.35 Hasil uji pada website Google PageSpeed Insight .....	55
Gambar 4.36 Hasil uji dari website Pingdom .....	56

## **DAFTAR TABEL**

Tabel 2.1 Tabel Penelitian Terdahulu .....	5
Tabel 2.2 Batasan Thingspeak tanpa license.....	9
Tabel 2.3 Simbol-simbol ERD .....	11
Tabel 2.4 Tipe Data MySQL.....	17
Tabel 2.5 Simbol-simbol DFD .....	18
Tabel 3.1 Rancangan Tabel Uji Black Box.....	28
Tabel 3.2 Rancangan Tabel Uji UAT .....	29
Tabel 4.1 Bobot Nilai pada tiap Jawaban .....	50
Tabel 4.2 Data Jawaban yang diberikan .....	50
Tabel 4.3 Penghitungan Data berdasarkan bobot nilai pada tiap jawaban.....	51
Tabel 4.4 Hasil akhir dari UAT.....	54

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Energi Listrik telah menjadi kebutuhan utama bagi manusia untuk menjalankan hampir setiap aktivitasnya. Dengan begitu maka kebutuhan akan energi listrik juga akan semakin meningkat. Dengan penggunaan energi listrik yang tidak terkontrol dan terus menerus tentu juga akan berdampak buruk pada kerusakan lingkungan dan habisnya cadangan energi. Selain itu penggunaan energi listrik yang boros dapat memicu gejolak sosial ekonomi yang berkepanjangan. Salah satu cara untuk mencegah hal ini adalah pembuatan sebuah sistem informasi untuk memonitoring energi listrik [1].

Menurut data ASEAN Centre for Energy (ACE) pada beberapa tahun terakhir, Indonesia tercatat sebagai negara dengan tingkat pemborosan energi listrik paling tinggi. Badan Pengkajian dan Penerapan Teknologi (BPPT) menunjukkan, intensitas pemborosan penggunaan energi di Indonesia empat kali lebih besar dibandingkan di Jepang.

Sistem Informasi ini dibuat untuk memonitoring pemakaian energi listrik supaya pemakaian energi listrik menjadi lebih efisien dan tidak ada pemakaian energi listrik yang tidak diperlukan. Sistem ini bersifat *real time* dengan mengukur *variable* energi dan biaya per kWh (*kilowatt hour*) yang dapat diubah. Pengukuran ini selanjutnya disimpan ke dalam *database* dan dapat memperlihatkan *record* dari periode atau waktu tertentu. Selain itu, *user* dapat mengeksport data tersebut ke dalam format .xls atau .csv.

Sistem Informasi ini juga digunakan sebagai *controller* yang digunakan untuk mengatur *device* yang ada. *Controller* disini dilakukan dengan menekan tombol on/ off pada sistem ini, setelah itu *value* akan dikirimkan ke *device*.

Sistem Informasi ini memanfaatkan *Application Programming Interface* (API) dalam pengiriman data dari *Hardware* ke *Database*. API merupakan salah satu implementasi dari konsep Internet of Things (IoT).

Sistem Informasi Building Management System (BEMS) menggunakan *framework* Laravel, yang sangat bermanfaat dalam mempermudah pembuatan sistem. *Framework* Laravel memiliki konsep MVC (*Model, View, Controller*), dimana memiliki fungsi masing – masing, yaitu *Model* yang berhubungan dengan *database*, *View* yang berhubungan dengan *user interface*, dan *Controller* yang berfungsi sebagai *control* antara *Model* dan *View*. Dengan adanya konsep ini, maka setiap komponen akan terbagi dan dapat tersusun dengan baik. Tujuan dalam penggunaan *framework* Laravel ini adalah dengan menggunakan *framework* ini maka pembuatan sistem informasi akan lebih cepat dan adanya beberapa fitur dari *framework* Laravel yang sangat mudah untuk diaplikasikan.

Metode pengembangan yang digunakan adalah RAD (*Rapid Application Development*). Tujuan penggunaan metode RAD pada sistem ini adalah supaya pengembangan sistem ini dapat dilakukan dalam waktu yang relatif cepat [2]. Metode pengembangan sangat diperlukan untuk meningkatkan kualitas dari sistem dan mengikuti kebutuhan dari *user*.

## 1.2 Rumusan Masalah

Berdasarkan uraian di atas, maka penulis merumuskan permasalahan yang ada antara lain sebagai berikut:

- a. Bagaimana mendesain Sistem Informasi Manajemen Energi agar mudah dimengerti oleh user?
- b. Bagaimana implementasi Sistem Informasi Manajemen Energi?
- c. Bagaimana Prosedur dan Hasil pengujian pada Sistem Informasi Manajemen Energi?

## 1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini antara lain sebagai berikut:

- a. Menghasilkan sebuah sistem informasi yang *user-friendly*.
- b. Mengimplementasikan Sistem Informasi Manajemen Energi dengan tepat.

- c. Melakukan pengujian *User Acceptance Testing* (UAT) dan *Black Box* pada Sistem Informasi Manajemen Energi.

#### **1.4 Manfaat Penelitian**

- a. Bagi Fakultas Teknik, menghasilkan sebuah sistem yang berguna untuk *monitoring* pemakaian listrik
- b. Bagi Akademisi, menghasilkan sebuah penelitian yang bisa digunakan sebagai bahan referensi bagi penelitian selanjutnya
- c. Bagi Pemerintah, data yang tersimpan dalam *database* dapat dijadikan acuan untuk menerapkan kebijakan-kebijakan pemerintah

#### **1.5 Batasan Masalah**

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini fokus pada proses pengembangan Sistem Informasi Manajemen Energi yang digunakan untuk *monitoring* penggunaan energi listrik dan *controlling device* yang ada. Bagian sistem *hardware* merupakan sebuah penelitian terpisah.
2. Data yang ditampilkan pada Sistem Informasi Manajemen Energi menyesuaikan *hardware* yang ada, yaitu dengan protokol komunikasi dari platform Thingspeak *free version*. Versi ini memiliki spesifikasi *update* data setiap 15 detik, hanya memiliki 4 *channel*, jumlah MQTT *subscriptions* terbatas hanya 3 *subscriptions*, jumlah pesan per tahun adalah 3 juta pesan atau 8.200 pesan per hari.

#### **1.6 Sistematika Penulisan**

Susunan dan pembahasan yang direncanakan pada penulisan Skripsi ini adalah sebagai berikut:

#### **BAB I Pendahuluan**

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, tujuan, dan manfaat penelitian, serta sistematika penulisan laporan.

#### **BAB II Tinjauan Pustaka**

Bab ini berisi hasil studi penulis dari beberapa literatur tentang teori-teori yang berkaitan dengan topik pembahasan. Pada penelitian ini, fokus utama pembahasan adalah perancangan Sistem Informasi yang digunakan untuk *monitoring* energi pada Program Studi Teknik Elektro UNS.

### **BAB III Metodologi Penelitian**

Bab ini menguraikan tahap-tahap dan prosedur penelitian yang dilakukan. Tahapan yang dilakukan mulai persiapan, desain, pembuatan code, dan tata cara pengujian.

### **BAB IV Hasil dan Pembahasan**

Bab ini menyajikan uraian data, serta analisis pembahasan mengenai pembuatan Sistem Informasi Manajemen Energi yang terdiri dari pengujian fitur aplikasi, keamanan data, serta ketepatan waktu metode *Rapid Application Development* (RAD).

### **BAB V Penutup**

Bab ini berisi tentang kesimpulan dan saran, sebagai hasil dan analisis pembuatan Sistem Informasi Manajemen Energi dengan metode *Rapid Application Development* (RAD).

### **Daftar Pustaka**

Berisi tentang sumber-sumber pada referensi yang digunakan dalam penyusuan skripsi ini.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian terdahulu**

Terdapat beberapa penelitian terdahulu yang berkaitan dengan Sistem Informasi atau yang berkaitan dengan metode yang penulis gunakan, baik metode pengembangan ataupun metode pengiriman data.

Seperti penulis Yu, H. R. [5] yang melakukan penelitian mengenai *Framework Laravel* atau Sagdeo P. [4] yang dalam penelitiannya dapat diketahui bahwa API dapat digunakan untuk komunikasi antar perangkat lunak dengan baik, dan beberapa penelitian yang lainnya. Adapun penelitian terdahulu tersebut, antara lain.

Tabel 2.1 Tabel Penelitian Terdahulu

No	Pengarang	Judul	Metoda	Hasil
1	Meidyan Permata Putri, Hendra Effendi [2]	Implementasi Metode Rapid Application Development Pada Website Service Guide “Waterfall Tour South Sumatra”	Implementasi metode RAD pada suatu website	Pengujian setiap sistem perangkat lunak dengan metode RAD dapat berjalan dengan baik dan lancar
2	Khaoula Benmoussa, Majida Laaziri, Samira Khoulji, Mohamed Larbi	A new model for the selection of web development frameworks: application to PHP frameworks	Penggunaan PHP Framework dalam mengembangkan web	Penggunaan PHP Framework memudahkan dalam memprogram web

	Kerkeb, Abir El Yamami [8]			
3	Akash Jadhav, Rohit Jadhav & Shivganga Gavhane [14]	Implementation of Web Application using Laravel Framework	Penggunaan Framework Laravel pada Web Application	Aplikasi web yang dibuat dengan framework laravel dapat mengakses informasi bisnis, komunikasi bisnis-klien yang mudah dan pemesanan dan manajemen produk
4	Zainab H. A., Hesham A. A., Mahmoud M. B. [3]	Internet of Things (IoT): Definitions, Challenges and Recent Research Directions	Penelitian mengenai <i>Internet of Things</i> (IoT)	Penggunaan IoT memiliki pengaruh yang besar dalam pengimplemen tasian
5	Sagdeo P. [4]	Application Programming Interfaces and The Standardization- Value	Penelitian mengenai <i>Application Programming Interface</i> (API)	API dapat digunakan untuk komunikasi antar komponen perangkat

		Appropriation Problem		lunak dengan baik
6	Yu, H. R. [5]	Design and Implementation of web based on Laravel framework	Penggunaan <i>Framework</i> Laravel dalam suatu <i>website</i>	Desain <i>web</i> menggunakan <i>framework</i> laravel memiliki skalabilitas yang kuat sehingga dapat meningkatkan efisiensi pengembangan
7	Soegoto, E. S. [9]	Implementing Laravel Framework website as brand image in higher-education institution	Pengimplementasi <i>an website</i> <i>Framework</i> Laravel sebagai <i>brand image</i>	Penggunaan <i>framework</i> laravel pada <i>web</i> sebagai <i>brand image</i> menunjukkan pengaruh yang signifikan jika dibandingkan dengan situs <i>web</i> lama tanpa <i>framework</i> laravel
8	Mohammed Thakir Mahmood, Osama	Design and Implementation of Web	Menganalisis, mendesain, mengimplementasikan website	Menggunakan MVC dalam mengembangkan aplikasi

	Ibrahem Ashour, Osman Ucan, Oguz Bayat [15]	Based For Intermediate Online Shop with Laravel Framework	untuk membelikan konsumen produk dari toko secara global menggunakan teknologi MVC	lebih aman karena terdapat class seperti authentication, routing, database managing, dll
9	Sabah Al-Fedaghi [7]	Developing Web Applications	Penelitian berbasis UML yang digambarkan dalam aliran diagram aktivitas UML	Konseptualisasi berbasis aliran diagram aktivitas UML menjanjikan untuk memberikan dasar yang kuat untuk pengembangan aplikasi web pada tahap awal desain.

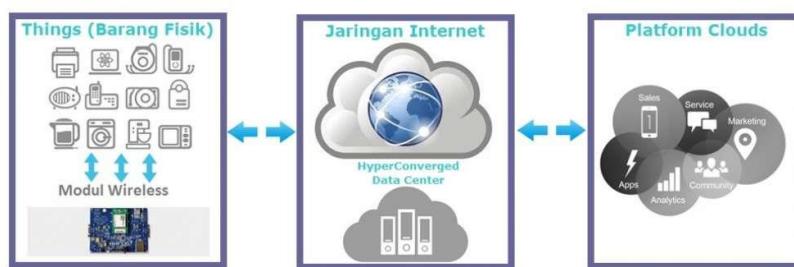
Dari penelitian terdahulu diatas tidak terdapat sebuah sistem yang menerapkan atau menggabungkan teknologi *Internet of Things* (IoT) dengan sebuah sistem informasi. Penelitian yang dikembangkan oleh penulis menggabungkan teknologi tersebut, dengan menggunakan *Application Programming Interface* (API) sebagai sarana yang digunakan untuk mengumpulkan data dari suatu *hardware* ke *software* yang ada pada Sistem Informasi Manajemen Energi. Dan sistem ini akan bersifat *real time* sehingga sangat berguna bagi *user* dalam melakukan proses *monitoring*. Selain itu, sistem informasi ini juga berfungsi sebagai *controller* untuk mengontrol *device* yang ada pada tiap tempat.

## 2.2 Internet of Things (IoT)

IoT berarti kemampuan untuk membuat semua disekitar kita (perangkat, mesin, *handphone*, mobil, kota, jalan, dan lain-lain) terkoneksi ke dalam internet dengan kecerdasannya [3].

IoT ini menjadi sebuah teknologi yang mendasari dalam Sistem Informasi Manajemen Energi, dimana dengan teknologi ini, data dari suatu *device* dapat dikirimkan ke *device* yang lain dengan mudah yaitu hanya melalui Internet.

Konsep dari IoT cukup sederhana dengan cara kerja mengacu pada 3 elemen utama pada arsitektur IoT, yaitu sebuah barang fisik yang dilengkapi dengan modul IoT, perangkat koneksi ke internet seperti modem dan *Cloud Data Center* yang menjadi tempat untuk menyimpan aplikasi beserta *database*.



Gambar 2.1 Konsep IoT [22]

## 2.3 Thingspeak

Thingspeak adalah sebuah *platform open-source API* dan IoT yang berbasis *web* yang digunakan untuk menyimpan data sensor [23].

Pada Sistem Informasi Manajemen Energi, Thingspeak digunakan sebagai tempat hardware dalam menyimpan data dan kemudian data tersebut akan diteruskan dan disimpan ke dalam *database* server.

Tabel 2.2 Batasan Thingspeak tanpa license

Fitur	Keterangan
Jumlah pesan	3 juta/tahun (~8200/hari)
Batas <i>interval</i> pembaruan pesan	Setiap 15 detik
Jumlah <i>channel</i>	4
Jumlah MQTT <i>subscriptions</i>	Terbatas hingga 3

Berbagi <i>private channel</i>	Terbatas hingga 3
--------------------------------	-------------------

## 2.4 Sistem Informasi

Sistem Informasi adalah sistem kerja yang proses dan kegiatannya dikhkusukan untuk memproses informasi, yaitu menangkap, mentransmisikan, menyimpan, mengambil, memanipulasi, dan menampilkan informasi. Dengan demikian, sistem informasi adalah sistem dimana peserta manusia dan atau mesin melakukan pekerjaan (proses dan kegiatan) menggunakan informasi, teknologi, dan sumber daya lainnya untuk menghasilkan produk dan atau layanan informasi untuk pelanggan internal atau eksternal [6].

## 2.5 Web Application

*Web Application* atau Aplikasi *Web* merujuk pada sebuah aplikasi yang diakses melalui *Web Browser* melalui jaringan dan dikembangkan menggunakan bahasa yang didukung *Browser*, seperti HTML dan JavaScript. Aplikasi *Web* diperlukan di bidang interaksi bisnis ke bisnis melalui jaringan, misalnya untuk perusahaan di luar negeri yang melakukan *outsourcing* proyek satu sama lain. Penggunaan aplikasi *web* dapat memberikan proses penting seperti transfer dana dan pembaruan informasi harga [7].

## 2.6 Database Management System (DBMS)

*Database* atau basis data adalah kumpulan terpadu dari file-file terkait, bersama dengan rincian interpretasi dari data yang terkandung di dalamnya. Pada dasarnya, basis data tidak lebih dari sistem penyimpanan catatan berbasis komputer yaitu sistem yang keseluruhan tujuannya adalah untuk merekam dan memelihara inforamsi atau data [12].

*Database Management System* (DBMS) adalah sistem perangkat lunak yang memungkinkan akses ke data yang terkandung dalam *database*. Tujuan dari DBMS adalah untuk menyediakan metode yang mudah dan efektif untuk mendefinisikan, menyimpan dan mengambil informasi yang terkandung dalam *database*. DBMS berinteraksi dengan program aplikasi, sehingga data yang terkandung dalam *database* dapat digunakan oleh banyak aplikasi dan

pengguna. Selain itu, DBMS memberikan kontrol terpusat terhadap *database*, mencegah pengguna yang curang atau tidak sah mengakses data, dan memastikan privasi data [12].

### 2.5.1. Entity Relationship Model

*Entity Relationship Model* (ERD) adalah sebuah model data yang didasarkan pada persepsi dunia nyata, yang terdiri atas *object* dasar yang disebut dengan entitas dan hubungan antara *object* tersebut.

Ada tiga macam relasi dalam hubungan atribut dalam satu file, antara lain sebagai berikut:

1. *One to one relationship*

Di dalam *one to one relationship*, setiap baris data pada tabel pertama dihubungkan dengan hanya ke satu baris pada tabel kedua. Hubungan antara file pertama dan kedua adalah satu banding satu.

2. *One to many relationship*

Di dalam *one to many relationship*, setiap baris data pada tabel pertama dapat dihubungkan ke satu baris atau lebih data pada tabel kedua. Hubungan antara file pertama dan kedua adalah satu berbanding banyak.

3. *Many to many relationship*

Di dalam *many to many relationship*, satu baris atau lebih data pada tabel pertama dapat dihubungkan ke satu atau lebih baris data pada tabel kedua. Artinya ada banyak baris di tabel satu dan tabel kedua yang saling berhubungan satu sama lain. Hubungan tabel pertama dan tabel kedua adalah banyak berbanding banyak.

Tabel 2.3 Simbol-simbol ERD

Gambar	Keterangan
	Entitas
	Hubungan

	Penghubung ( <i>link</i> )
	One to Many
	One to One
	Many to Many

## 2.7 Framework

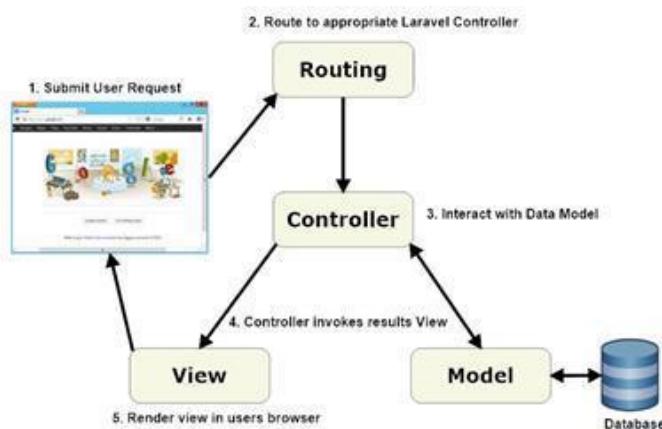
*Framework* memfasilitasi pemrograman *Web* dan membuatnya lebih teratur dalam beberapa cara. Pertama, *Framework* meningkatkan produktivitas pemrograman karena menulis sepotong kode yang biasanya memakan waktu berjam-jam dan mengambil ratusan baris kode dapat dilakukan dalam hitungan menit menggunakan fungsi *Framework* yang tertanam. Di sisi lain, *Framework* biasanya dilengkapi dengan *support team*, dokumentasi atau forum penting dimana pengguna dapat dengan cepat mendapatkan jawaban atas kendala yang dialaminya [10].

*Framework* biasanya menggunakan paradigma *Model View Controller* (MVC). Hal ini memastikan pemisahan antara “*model*” data yang bertanggung jawab atas akses dan permintaan ke *database*, “*view*” yang merupakan deskripsi penyajian data, dan “*controller*” yang mengimplementasikan logika bisnis dengan mengambil data “*model*” [10].

## 2.8 Laravel

Laravel adalah sebuah *framework* dengan sintaks yang elegan dan memiliki fungsi luas seperti keamanan, penyimpanan kata sandi, pengingat dan reset kata sandi, enkripsi dan validasi. Menurut sebuah survei yang dilakukan pada bulan Maret 2015 tentang popularitas *Framework* PHP, Laravel adalah *Framework* yang paling populer, berturut-turut diikuti oleh Symfony2, Nette, CodeIgniter, Yii2, dan lain-lain [9].

Laravel dirancang untuk meningkatkan kualitas perangkat lunak, menyederhanakan otentikasi, memudahkan perutean, memudahkan akses, dan meningkatkan daya dalam *framework website* [9].



Gambar 2.2 Konsep MVC pada Laravel [21]

## 2.9 Application Programming Interface (API)

API adalah seperangkat antarmuka yang mengekspos layanan atau data yang disediakan oleh aplikasi perangkat lunak melalui serangkaian *resources* yang telah ditentukan, seperti metode, objek, atau URI (Stylos, Faulring, Yang, & Myers, 2009). Dengan menggunakan *resources* ini, aplikasi lain dapat mengakses data tersebut [18].

Sebuah API menentukan bagaimana komponen perangkat lunak dapat berinteraksi satu dengan yang lain [4]. Dengan adanya API ini maka akan memudahkan dalam mengirimkan dan menerima data yang diinginkan dari satu perangkat ke perangkat yang lainnya.

## 2.10 PHP

PHP adalah bahasa pemrograman yang secara khusus cocok untuk pengembangan *web* dan dapat disematkan ke dalam HTML. PHP hampir sama dengan bahasa C untuk pengembangan *web*. Struktur sintaksis pada dasarnya sama, sehingga fleksibel dan mudah untuk mengimplementasikannya pada aplikasi *web* apapun [13].

## 2.11 Hyper Text Markup Language (HTML)

HTML merupakan singkatan dari *Hyper Text Markup Language* yang digunakan untuk mengembangkan struktur halaman *web* [12].

*Tag* adalah penanda perintah yang ada di dalam HTML. *Tag* digunakan untuk menentukan tampilan dari dokumen HTML. Berikut adalah beberapa *tag* yang sering digunakan pada HTML [12]:

1. <!DOCTYPE HTML> : Digunakan untuk mendeskripsikan versi dari HTML
2. <HTML> : Merupakan elemen *root* pada halaman *Web* HTML
3. <HEAD> : Memiliki informasi mengenai dokumen
4. <TITLE> : Digunakan untuk memberikan judul pada Dokumen *Web*
5. <BODY> : Berisi semua informasi dalam dokumen
6. <H1> : Mendefinisikan *heading* ukuran besar
7. <P> : Mendefinisikan paragraf

## 2.12 Cascading Style Sheet (CSS)

CSS adalah bahasa yang merepresentasikan halaman *website*. Seperti warna, *layout*, dan *font* [16]. Dengan menggunakan CSS, *web developer* dapat membuat halaman *website* menjadi lebih menarik dan indah. Pembuatan CSS dapat disisipkan di dalam halaman HTML.

Ada beberapa metode penulisan CSS ke dalam halaman HTML, yaitu:

### 1. External Style Sheet

Metode penulisan CSS secara terpisah dengan memberikan ekstensi .css untuk nama *filenya*. *File* akan disematkan atau *diload* dengan kode HTML seperti berikut.

```
<head>
<meta charset="UTF-8">
<title>Jenis CSS</title>
<link rel="stylesheet" href="style.css" type="text/css">
</head>
```

## 2. Internal Style Sheet

Metode penulisan CSS langsung di dalam *file* HTML tepatnya berada diantara *tag head* dengan menggunakan *tag style*. Contoh penggunaan *Internal Style Sheet*.

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Jenis CSS</title>
    <style type="text/css">
      body {font-family: arial;}
      h1 {
        background-color: red;
        color: white;
        padding: 5px;
      }
    </style>
  </head>
  <body>
    <h1>Cascading Style Sheet</h1>
  </body>
</html>
```

## 3. Inline Style Sheet

Metode penulisan CSS langsung pada *tag* HTML yang ingin diberikan *style* CSS. Metode ini sangat tidak disarankan karena akan membuat *tag* pada HTML akan penuh dengan *attribute* HTML dan membuat ukuran *file* pada HTML akan bertambah besar. Contoh penggunaan *Inline Style Sheet*.

```
<body>
  <h1 style="background-color: red; color: white; padding: 5px">Cascading Style Sheet</h1>
```

</body>

## 2.13 XAMPP

XAMPP adalah singkatan dari *Cross-Platform* (X), Apache (A), MySQL (M), PHP (P) dan Perl (P). XAMPP adalah distribusi Apache yang sederhana dan ringan yang membuatnya sangat mudah bagi pengembang untuk membuat *web server* lokal untuk keperluan pengujian. Semua yang diperlukan untuk menyiapkan *web server* – aplikasi *server* (Apache), *database* (MySQL), dan bahasa pemrograman (PHP) – disertakan dalam file sederhana yang dapat diekstraksi. XAMPP juga *cross-platform* yang artinya berfungsi dengan baik di Linux, Mac, dan Windows. Pengembangan *web* menggunakan XAMPP terutama karena XAMPP berukuran kecil dan *beginner-friendly* [15].



Gambar 2.3 XAMPP Control Panel

## 2.14 MySQL

MySQL adalah sebuah *Relational Database Management System* (RDBMS) *open source* dan *multithreaded* yang dibuat oleh Michael “Monty” Widenius pada tahun 1995 [17]. MySQL memiliki banyak fitur yang membuatnya tetap menjadi sistem basis data yang hebat. Kecepatan adalah salah satu fitur yang menonjol.

Tabel 2.4 Tipe Data MySQL

JENIS TIPE	TIPE	KETERANGAN
NUMERIK	INT	-2147483648 sampai 2147483647 Untuk yang bertanda, 0 sampai 4294967295 Untuk yang tidak bertanda
	TINYINT	128-127 Untuk yang bertanda, 0-255 Untuk yang tidak bertanda
	SMALINT	31768-32767 Untuk yang bertanda, 0-65535 Untuk yang tidak bertanda
	MEDIUMINT	-8388608-8388607 atau 0-16777215
	BIGINT	-9223372036854775808 sampai dengan 9223372036854775807 atau 0 – 1844674407370955161
	FLOAT(M,D)	Bilangan pecahan presisi tunggal
	DOUBLE(M,D)	Bilangan pecahan presisi ganda
	DECIMAL(M,D)	Bilangan dengan decimal
DATE	DATE	Tanggal dengan format YYYY-MM-DD
	DATE TIME	Tanggal dan waktu dengan format: YYYY-MM-DD HH:MM:SS
	TIMESTAMP	Tanggal dan waktu dengan format: YYYYMMDDHHMMSS
	TIME	Waktu dengan format HH:MM:SS
	YEAR(M)	Waktu dengan format YYYY
STRING	CHAR(M)	0-255 karakter dengan ukuran tetap.
	VARCHAR(M)	0-255 karakter dengan ukurang bervariasi.
	TEXT	String dengan Panjang maksimum 65535 karakter.
	BLOB	Tipe data untuk menyimpan data binary dalam jumlah besar. String dengan Panjang maksimum 65535

	ENUM	Enumerasi atau sebuah list (daftar)
--	------	-------------------------------------

## 2.15 Data Flow Diagram (DFD)

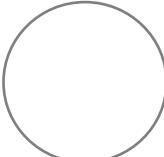
DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan: darimana asal data, dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut [20].

Salah satu manfaat dari DFD adalah memungkinkan penganalisis sistem memahami keterkaitan antara subsistem yang satu dengan subsistem yang lain.

Ada empat buah simbol pada DFD, yang masing – masingnya digunakan untuk:

- External entity* : digunakan untuk menyatakan sesuatu yang berhubungan dengan sistem. Contohnya adalah kantor, departemen atau divisi dalam perusahaan, orang, sistem informasi yang lain. Sesuai dengan namanya, entitas ini berada diluar sistem
- Data flow* : digunakan untuk menunjukkan arus dari data yang dapat berupa masukan untuk sistem ataupun hasil dari proses sistem. *Data flow* sebaiknya diberi nama yang jelas dan mempunyai arti
- Process* : digunakan untuk menunjukkan kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang keluar dari proses. Suatu proses harus menerima arus data dan menghasilkan arus data
- Data store* : digunakan untuk menunjukkan simpanan dari data yang dapat berupa suatu file atau *database* di sistem komputer

Tabel 2.5 Simbol-simbol DFD

No	Gambar	Nama
1		<i>Process</i>

2		<i>Data Flow</i>
3		<i>Data Store</i>
4		<i>External Entity</i>

## 2.16 Metode Pengembangan

Metode Pengembangan adalah suatu rangkaian aktivitas yang dilakukan untuk merancang, membuat, mengaplikasikan, dan mendukung atau meningkatkan fungsi perangkat lunak.

### 2.15.1 *System Development Life Cycle (SDLC)*

Metode pengembangan perangkat lunak tradisional yang sistematis. Memiliki lima tahapan proses, diantaranya *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*.

### 2.15.2 *Rapid Application Development (RAD)*

Metode ini menggunakan pendekatan interaktif dan inkremental, tetapi lebih menekankan pada tenggat waktu dan efisiensi biaya yang sesuai dengan kebutuhan.

Proses pengembangan dengan metode RAD lebih singkat, karena semua pihak yang terlibat dengan pengembangan sistem terus terlibat aktif dalam setiap proses hingga hasil dapat tercapai. Tahapan atau alur pada metode ini juga lebih sedikit.

### 2.15.3 *Prototype*

Metode ini memiliki 5 tahapan, dimulai dengan pengumpulan informasi dari *user* ke tim pengembang. Kemudian tim pengembang akan merencanakan sistem dan mengerjakan *prototype* dari perangkat lunak. *Prototype* tersebut akan diserahkan kepada *user* untuk dilakukan evaluasi. Tim pengembang akan merevisi sistem sampai memenuhi kebutuhan dari *user*. Ketika revisi sudah selesai dan telah diterima oleh *user*, maka perangkat lunak siap untuk diterjemahkan ke dalam perangkat keras. Kemudian, dilakukan proses uji coba dan berbagai revisi sebelum mulai dapat digunakan.

### 2.15.4 *Spiral*

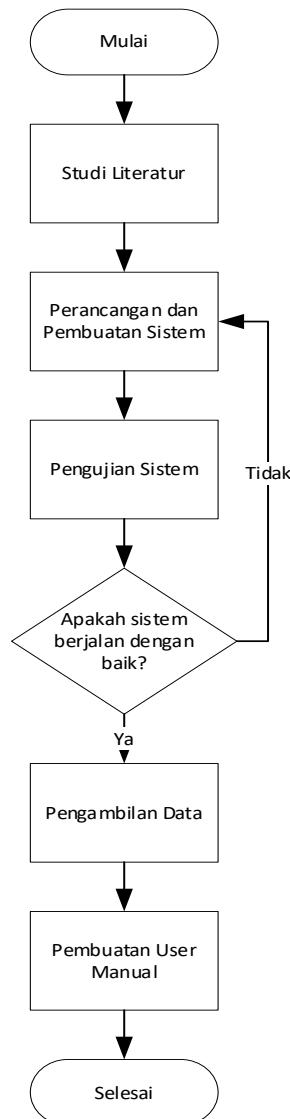
Metode ini menggabungkan metode *Waterfall* dan *prototype*. Metode ini umumnya diterapkan dalam pengembangan perangkat lunak berskala besar, sekaligus membutuhkan sistem yang kompleks. Setiap prosesnya disertai dengan analisis mendalam mengenai tingkat risiko dan keberhasilan pengembangan. Terdapat 5 tahapan diantaranya adalah komunikasi, perencanaan, analisis, pengembangan perangkat lunak dijalankan, evaluasi dari *user*.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Metode dan Lokasi Penelitian

Dalam penelitian ini, penulis menggunakan metode eksperimen dengan analisis data menggunakan pendekatan kualitatif. Lokasi penelitian ini dilakukan di Program Studi Teknik Elektro Universitas Sebelas Maret. Adapun metode penelitian dapat dijelaskan dalam *flowchart* berikut.



Gambar 3.1 *Flowchart* Metode Penelitian

### **3.2 Sumber Data**

Sumber data diperoleh dari lapangan secara langsung, yaitu dari *device* yang dibuat untuk mengambil data. *Device* ini akan mengirimkan data, dan akan diterima pada sistem yang kemudian akan dimasukkan ke dalam *database* dan ditampilkan pada Sistem Infromasi secara *real time*.

### **3.3 Metode Pengumpulan Data**

Motode pengumpulan data yang digunakan dalam penelitian ini, yaitu:

#### **1. Studi Literatur**

Salah satu metode pengumpulan data dengan cara mempelajari dan mengumpulkan sumber pustaka yang diperoleh dari buku, hasil penelitian, skripsi, dan media lain sebagai referensi penulis dalam melakukan penelitian.

#### **2. Wawancara**

Wawancara dilakukan dengan memberikan pertanyaan kepada orang yang berkaitan dengan sistem. Dalam penelitian ini, pihak yang diwawancarai adalah Dosen yang berkaitan dengan sistem ini, dan juga mahasiswa yang mengerjakan *hardware*.

#### **3. Observasi**

Observasi adalah teknik pengumpulan data dengan cara langsung mengamati permasalahan yang terjadi di lapangan. Dalam penelitian ini, pengamatan dilakukan langsung terhadap bagaimana data akan dikirimkan dari *device* yang telah dibuat dan dimasukkan ke dalam *database* sebagai sumber data dari sistem ini.

#### **4. Dokumentasi**

Dokumentasi digunakan untuk mengumpulkan dan mempelajari berbagai sumber dokumentasi, baik itu transkrip, buku tamu, agenda, laporan yang berkaitan dengan sistem. Yaitu mempelajari cara pengambilan data dari suatu *device* ke dalam sistem.

### **3.4 Instrumen Penelitian**

Adapun instrumen penelitian yang digunakan dalam penelitian, yaitu:

## 1. Perangkat Keras

Perangkat keras yang digunakan untuk mengembangkan dan mengumpulkan data, adalah sebagai berikut:

- Laptop ASUS dengan spesifikasi *Processor i5-6200U, HDD 1TB, Memory 8GB*

## 2. Perangkat Lunak

Perangkat lunak yang digunakan untuk mengembangkan sistem, adalah sebagai berikut:

- a. Sistem Operasi Windows 10
- b. XAMPP
- c. Atom
- d. Brave *Browser*
- e. *Framework* Laravel
- f. HTML
- g. CSS
- h. PHP
- i. Bootstrap
- j. API
- k. Javascript
- l. MySQL
- m. MQTT.fx

### 3.5 Metode Rapid Application Development (RAD)

RAD adalah model proses perangkat lunak yang menekankan pada daur pengembangan hidup yang singkat. Dari definisi konsep RAD, metode ini dapat dilakukan dalam waktu yang relatif lebih cepat.



Gambar 3.2 Tahapan Metode RAD [2]

Tahapan RAD terdiri dari 3 tahap yang terstruktur dan saling bergantung di setiap tahap, yaitu:

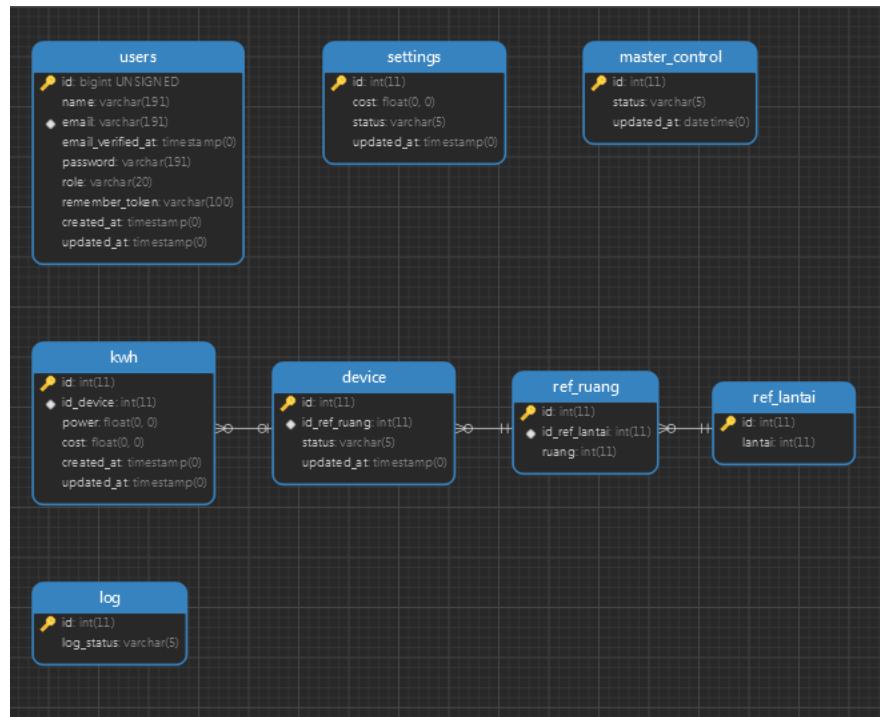
1. *Requirements Planning*
  - a. *User* dan *analyst* bertemu untuk mengidentifikasi tujuan dari aplikasi atau sistem.
  - b. Berorientasi pada pemecahan masalah bisnis.
2. *Design Workshop*
  - a. Fase desain dan penyempurnaan.
  - b. *Programmer* dan *analyst* membangun dan menunjukkan tampilan visual desain dan alur kerja pengguna.
  - c. *Analyst* menyempurnakan modul berdasarkan tanggapan *user*.
3. *Implementation*
  - a. Menerapkan dan menguji aplikasi atau sistem yang telah dibuat.

#### 3.5.1 Requirements Planning

Tahapan ini bertujuan untuk mengidentifikasi kebutuhan, batasan dan objektifitas dari suatu sistem yang akan dibangun dengan mengumpulkan data dari *stakeholder*.

a. *Entity Relationship Diagram*

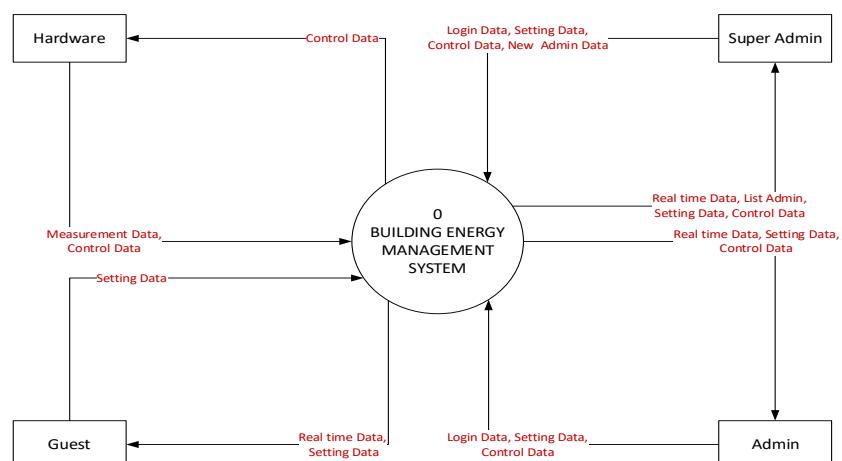
*Entity Relationship Diagram* (ERD) adalah diagram yang memperlihatkan entitas yang terlibat dalam suatu sistem serta relasi antar entitas tersebut.



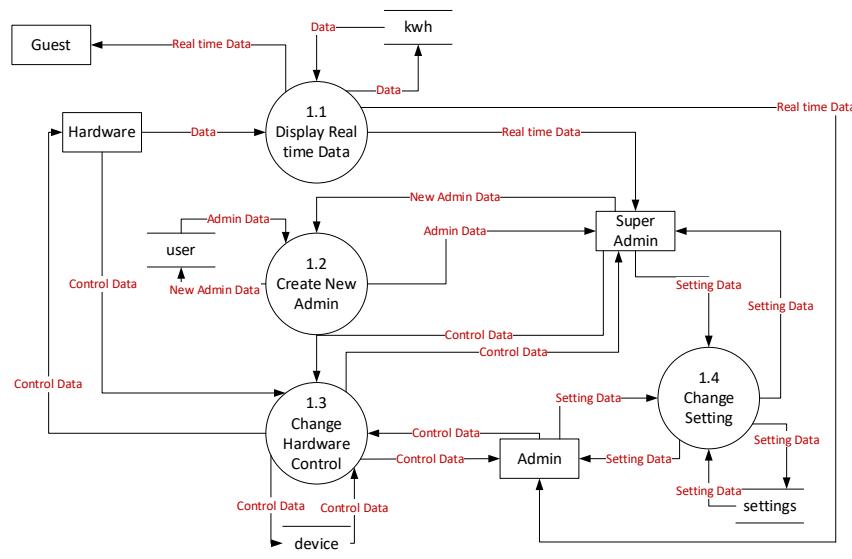
Gambar 3.3 *Entity Relationship Diagram*

b. *Data Flow Diagram*

*Data Flow Diagram* (DFD) memberikan tampilan secara visual tentang aliran data dan informasi dari suatu sistem.



Gambar 3.4 DFD Level 0 Sistem Informasi BEMS



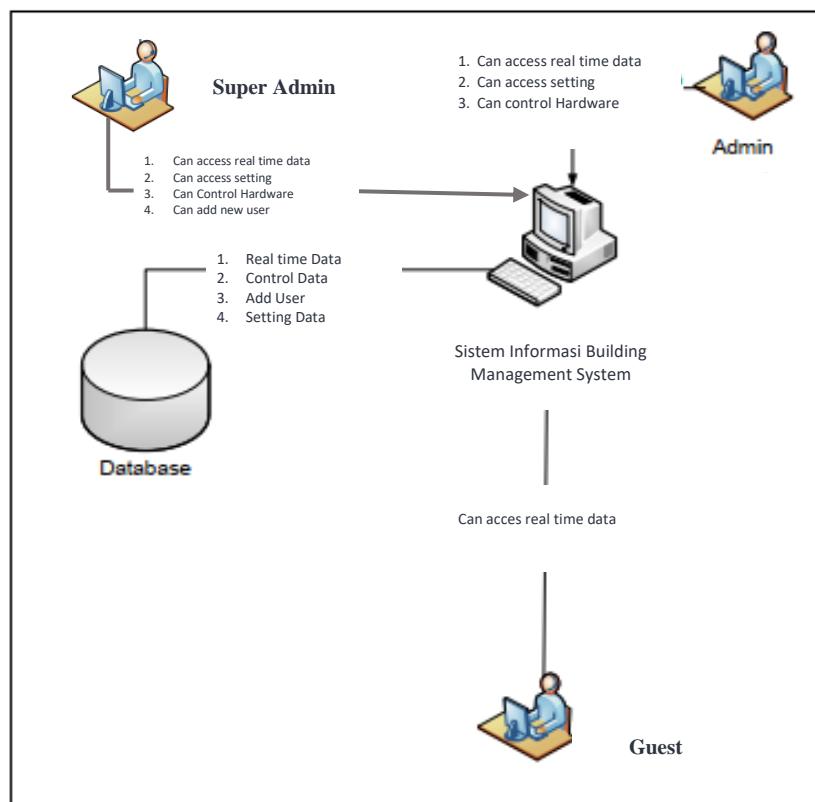
Gambar 3.5 DFD Level 1 Sistem Informasi BEMS

### 3.5.2 Design Workshop

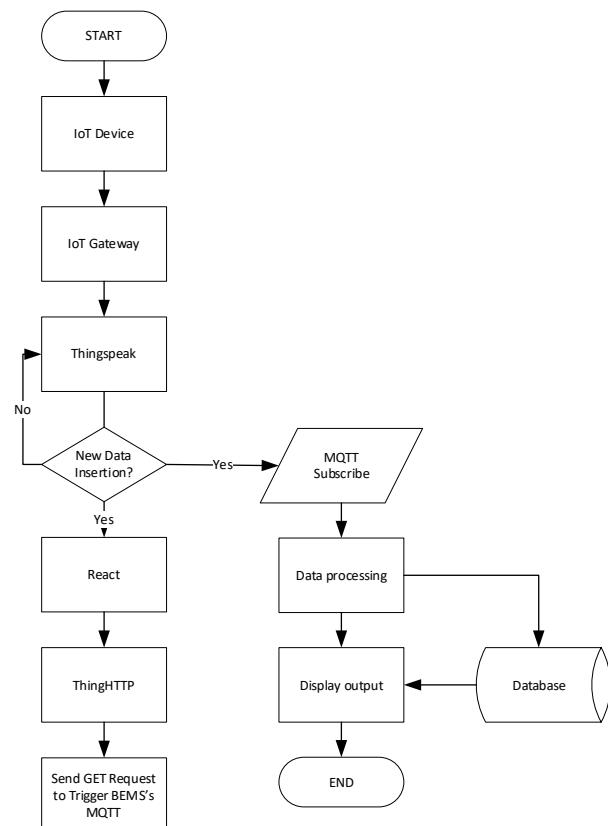
Pada tahapan *Design Workshop* bertujuan untuk merancang semua kegiatan dalam arsitektur sistem secara keseluruhan dan meningkatkan pemahaman atas masalah berdasarkan analisis yang telah dilakukan.

Pada proses ini melakukan proses desain dan melakukan perbaikan apabila masih terdapat ketidaksesuaian desain antara user dan analis.

Pada Sistem Informasi *Building Energy Management System*, *user* memberikan desain yang diinginkan dan sistem akan dibuat berdasarkan desain dari *user* tersebut. Dan ketika *user* memberikan komentar terhadap ketidaksesuaian dalam desain, maka sistem akan diperbaiki sesuai dengan permintaan dari *user*. *User* juga menentukan aliran data yang terjadi pada Sistem Informasi *Building Energy Management System*, yaitu menggunakan API dari Thingspeak. Adapun output lain dari tahap ini, adalah *Business process* yang terjadi seperti pada Gambar 3.5.



Gambar 3.6 *Business Process Diagram*



Gambar 3.7 *Flowchart Sistem Informasi BEMS*

### 3.5.3 Implementation

Tahapan *implementation* yaitu pengimplementasian sistem dan penerapan metode dalam pemrograman terhadap hasil kebutuhan sistem dan dapat dijelaskan dalam tahap implementasi *database* dan *coding* program. Tujuan dari tahapan ini adalah mengkonstruksi sistem dan penerapan metode dalam pemrograman.

## 3.6 Teknik Pengujian

Pengujian perangkat lunak adalah suatu elemen yang menjadi jaminan kualitas dari perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean. Contoh dari teknik pengujian adalah teknik pengujian *Black Box* dan *User Acceptance Testing* (UAT) yang akan digunakan untuk menguji Sistem Informasi Manajemen Energi.

*Black Box* adalah suatu pengujian yang menguji eksternal dari pekerjaan sistem seperti *input/ output* dari sistem, hasil yang diharapkan, dan *user interface* apakah sistem yang telah dibuat sesuai dengan desain [19].

Tabel 3.1 Rancangan Tabel Uji *Black Box*

No.	Menu/ Fungsi	Pengujian
1.	Proses Login	Menginputkan <i>email</i> dan <i>password user</i> kemudian memeriksa data <i>user</i> berdasarkan <i>input</i> yang telah dimasukkan
2.	Proses Menampilkan Data	Mengambil dan menampilkan data yang diperlukan dari dalam database ke halaman <i>web</i>
3.	Proses Pembaharuan Data	Melakukan pembaharuan data
4.	Proses Pengontrolan <i>Device</i>	Mengontrol <i>device</i> yang ada pada sistem

5.	Proses Logout	Menghapus <i>session user</i> dan kembali ke halaman <i>dashboard</i>
----	---------------	---

*User Acceptance Testing* adalah suatu pengujian yang melibatkan *end user*. Tujuannya untuk mengetahui apakah sistem yang digunakan sudah diterima dan sudah memenuhi kebutuhan yang diminta [19].

Tabel 3.2 Rancangan Tabel Uji UAT

No.	Pertanyaan	SA	A	N	D	SD
1	Apakah tampilan sistem informasi ini menarik?					
2	Apakah sistem informasi ini mudah dipahami dan digunakan?					
3	Apakah sistem ini memiliki performa yang bagus?					
4	Apakah sistem ini memenuhi fungsi <i>monitoring</i> ?					
5	Apakah sistem ini memenuhi fungsi sebagai <i>controller</i> ?					
6	Apakah sistem ini memiliki fitur yang berguna?					
7	Apakah sistem ini membantu dalam <i>monitoring</i> energi listrik?					
8	Apa saran Saudara/ri terhadap sistem ini?					

Pada Tabel 3.3 adalah Rancangan tabel untuk pengujian UAT, dimana *user* akan mengisi kuisioner berdasarkan pertanyaan yang ada. Dengan memberikan centang pada kolom pilihan jawaban, dimana terdapat 5 kategori yaitu SA (*Strongly Agree*), A (*Agree*), N (*Netral*), D (*Disagree*), dan SD (*Strongly Disagree*). Yang kemudian akan dihitung persentase untuk tiap

pertanyaan dari tiap pengisi kuisioner. Untuk pertanyaan ke 8 yaitu mengenai saran/ *feedback* tidak akan dihitung, tetapi jawaban akan dianggap sebagai masukkan untuk sistem yang kemudian akan dipertimbangkan pada sistem.

## BAB IV

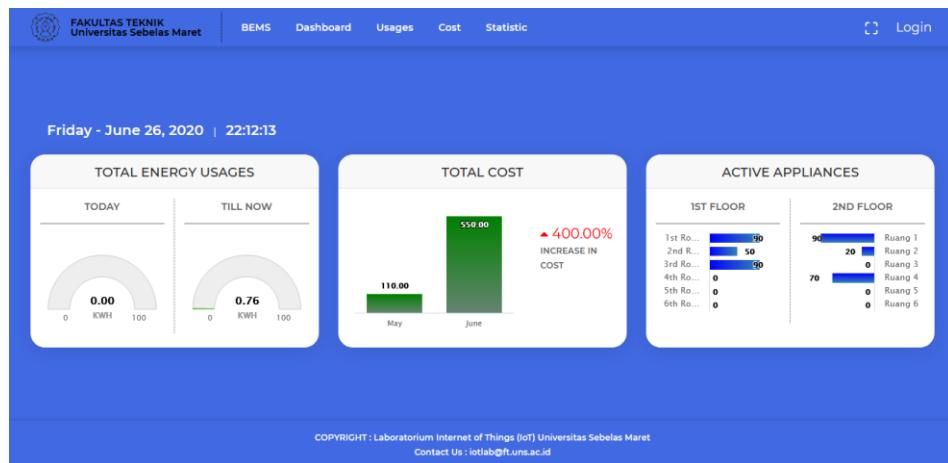
# HASIL DAN PEMBAHASAN

Dalam bab ini ditampilkan hasil dari implementasi metode *Rapid Application Development* (RAD) pada keadaan yang sesungguhnya.

### 4.1. Desain dan Implementasi Sistem Manajemen Energi

Implementasi *user interface* pada Sistem Informasi *Building Energy Management System* ini dilakukan berdasarkan desain yang telah dibuat oleh *user*. Implementasi diambil dari *screenshot* untuk tiap halaman yang ada pada *website*.

#### 1. Halaman Dashboard



Gambar 4.1 Halaman *Dashboard*

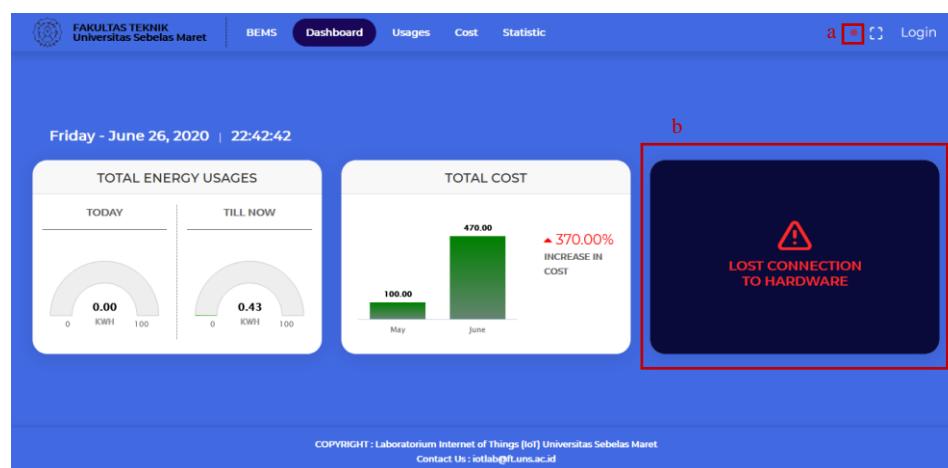
Halaman *Dashboard* ini merupakan halaman pertama yang akan dikunjungi oleh *user*. Pada halaman *dashboard* terdapat beberapa informasi yaitu *Total Energy Usages*, *Total Cost*, dan *Active Applicances*. Yang masing – masing dari informasi tersebut berada di dalam *card* dan *value* digambarkan dengan menggunakan grafik. Pada *card Total Energy Usages* berisi informasi mengenai penggunaan energi yang dibedakan menjadi 2, yaitu penggunaan energi pada hari ini (*Today*) dan sampai saat

ini (*Till Now*), data yang dikeluarkan oleh grafik *Till Now* adalah dimulai dari tanggal 1 bulan ini sampai detik sekarang.

*Card Total Cost* berisi informasi mengenai total biaya yang dikeluarkan, Total biaya yang dikeluarkan di sini adalah dari bulan lalu dan bulan ini, hal ini dilakukan supaya *user* bisa mengetahui perbandingan biaya yang dikeluarkan dari bulan kemarin dan bulan ini. Perbandingan ini digambarkan dengan persentase dan ada teks “*Increase in Cost*” yang akan muncul ketika total biaya bulan ini lebih besar daripada bulan lalu. Dan teks “*Decrease in Cost*” yang akan muncul ketika total biaya bulan lalu lebih besar daripada bulan ini.

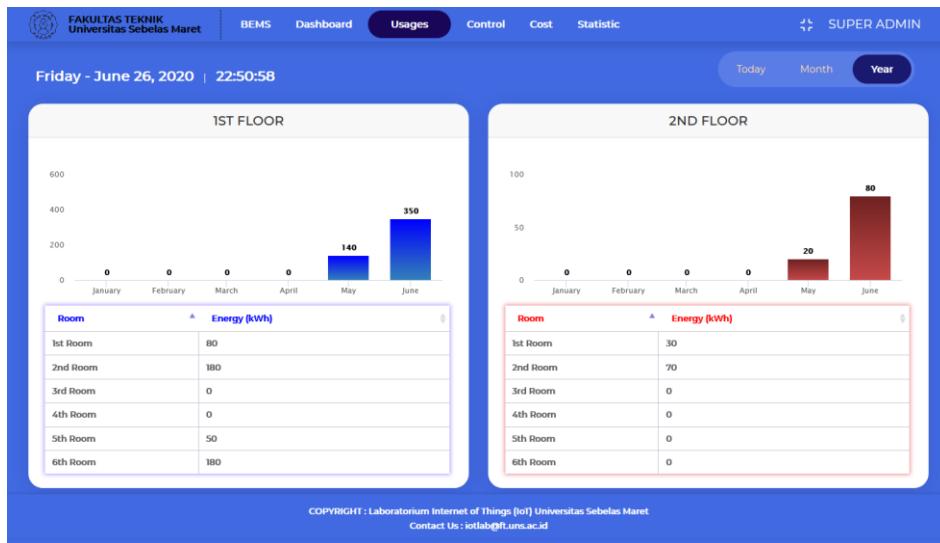
*Card Total Appliances* berisi informasi mengenai energi yang digunakan untuk setiap ruang di setiap lantai yang ada. Hal ini dilakukan supaya *user* dapat melihat penggunaan energi pada setiap ruangan.

Halaman *dashboard* ini mengambil data secara *real time*, hal ini dilakukan karena halaman *dashboard* ini akan digunakan untuk *monitoring*. Selain itu, terdapat sebuah fitur yang disematkan pada halaman *dashboard* ini, yaitu ketika *Hardware* tidak mengirimkan data baru ke dalam *database*, maka akan muncul sebuah indikator pada *navbar* dan juga pada *Card Total Appliances* akan tertutup dengan teks “*Lost Connection to Hardware*”. Fitur ini disematkan ke dalam sistem supaya *user* mengetahui keadaan dari *Hardware*. Dan karena indikator berada pada *navbar* maka indikator ini dapat dilihat *user* disaat *user* berada di halaman apa saja.



Gambar 4.2 (a) Indikator dan (b) Teks “*Lost Connection to Hardware*”

## 2. Halaman *Usages*



Gambar 4.3 Halaman *Usages*

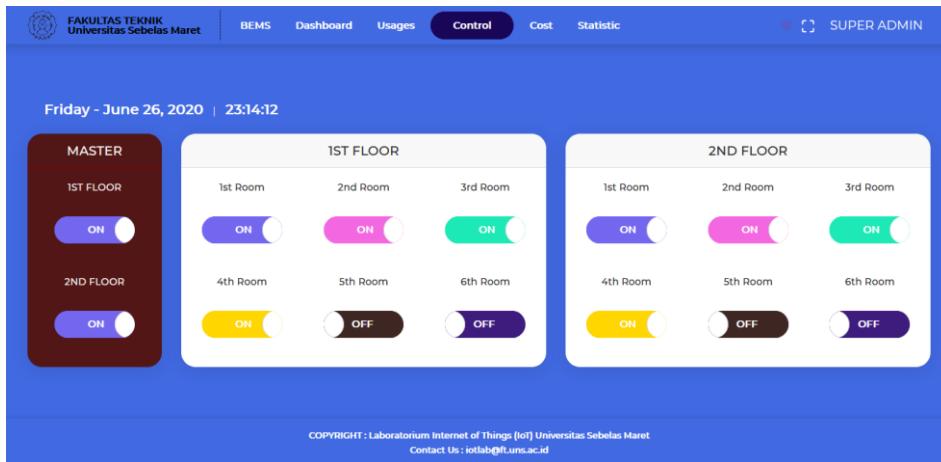
Halaman *Usages* merupakan sebuah halaman yang berisi tentang informasi penggunaan energi secara *detail*. Halaman *Usages* ini dapat memilih 3 periode penggunaan energi (*Today*, *Month*, *Year*), dan untuk setiap periodenya memiliki informasi yang *detail*.

Pada setiap periodenya terdapat informasi mengenai total penggunaan energi pada tiap ruang yang digambarkan dengan menggunakan tabel. Selain itu, informasi juga digambarkan dengan grafik yang dibedakan untuk tiap periodenya. Untuk periode hari ini (*Today*), maka data yang dikeluarkan adalah data untuk setiap jam dan dimulai dari jam 12 malam sampai pada saat halaman *usages* dibuka. Contoh ketika *user* membuka halaman *usages* pada saat jam 12 siang maka data penggunaan energi untuk hari ini akan mengeluarkan data dari jam 12 malam sampai jam 12 siang.

Untuk periode bulan (*Month*), maka data yang dikeluarkan adalah data untuk tiap minggu dari minggu pertama di bulan itu sampai pada saat halaman *usages* dibuka. Contoh ketika *user* membuka halaman *usages* pada saat tanggal 26 Juni, maka data penggunaan energi untuk bulan ini akan mengeluarkan data dari tanggal 1 bulan Juni sampai tanggal 26 Juni.

Untuk periode tahun (*Year*), maka data yang dikeluarkan adalah data untuk tiap bulan dari bulan pertama sampai pada saat halaman *usages* dibuka. Contoh ketika *user* membuka halaman *usages* pada saat bulan Juni, maka data penggunaan energi untuk bulan ini akan mengeluarkan data dari bulan Januari sampai bulan Juni.

### 3. Halaman *Control*

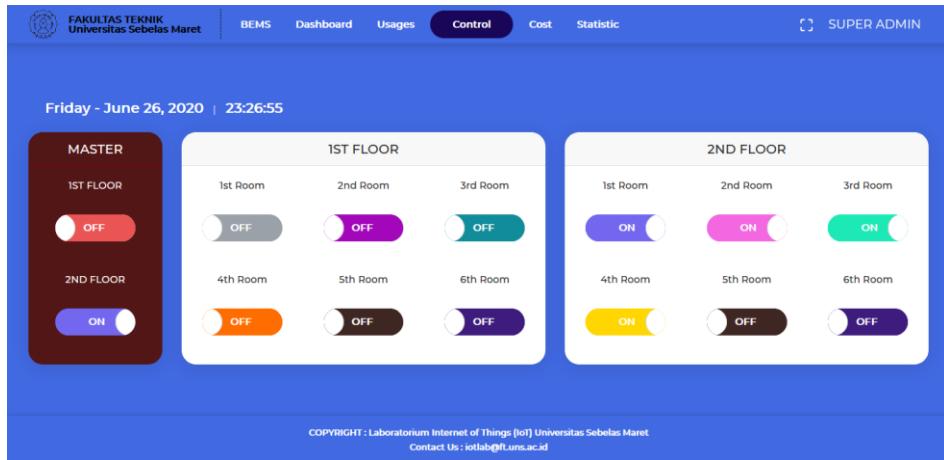


Gambar 4.4 Halaman *Control*

Halaman *Control* ini hanya dapat diakses oleh *Admin* dan *Super Admin*. Halaman *Control* berisi tombol on/ off untuk setiap ruang di setiap lantai dan tombol on/ off untuk *Control Master*.

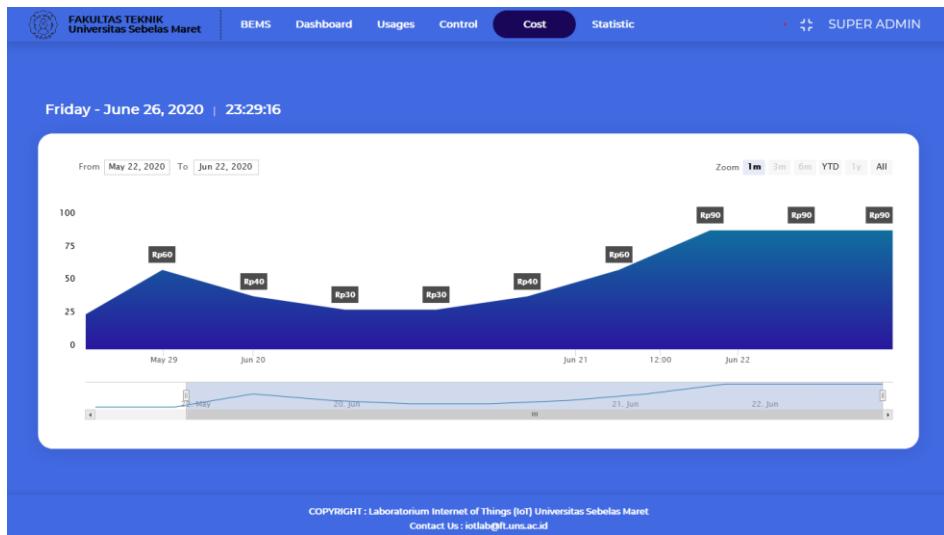
Warna setiap tombol untuk setiap ruang berbeda – beda, hal ini dilakukan supaya dapat memudahkan *user* dalam membedakan antara satu ruang dengan ruang lainnya.

Untuk *Control Master* dibuat untuk memudahkan *user* dalam menghidupkan dan mematikan *device* dalam satu lantai. Karena ketika *user* mematikan *Control Master* pada lantai 1, maka semua *device* pada lantai 1 akan mati begitu juga ketika *user* mematikan *Control Master* pada lantai 2, maka semua *device* pada lantai 2 akan mati. Sama halnya dengan mematikan *device*, hal ini juga berlaku ketika *user* menghidupkan *Control Master*.



Gambar 4.5 Mematikan *Control Master* di lantai 1

#### 4. Halaman *Cost*



Gambar 4.6 Halaman *Cost*

Halaman *Cost* berisi informasi mengenai biaya yang dikeluarkan dalam periode tertentu. Informasi ini digambarkan dalam bentuk grafik dengan periode yang dapat diubah – ubah oleh *user*, hal ini dilakukan supaya *user* dapat melihat data pada periode tertentu dengan cepat tanpa harus mencarinya terlebih dahulu. Ketika membuka halaman *cost*, *default* periode adalah 1 bulan, jadi grafik akan menampilkan data selama 1 bulan.



Gambar 4.7 Tampilan ketika *user* memilih periode

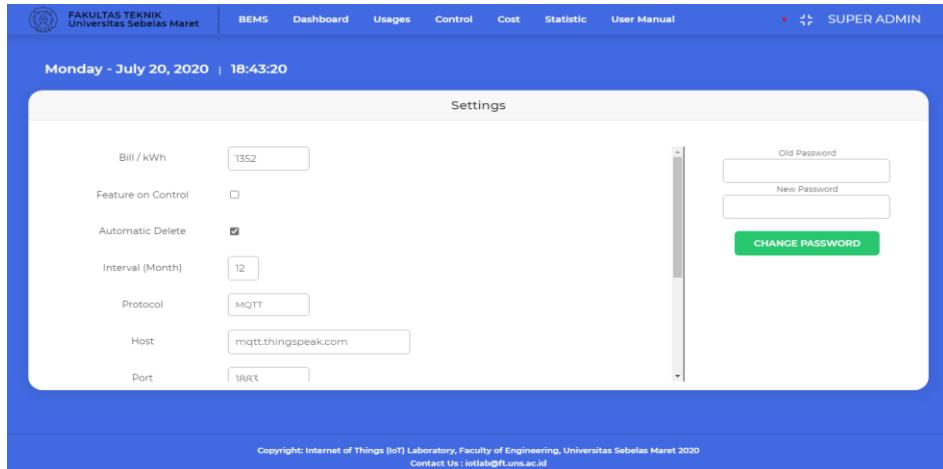
## 5. Halaman *Statistic*



Gambar 4.8 Halaman *Statistic*

Halaman *Statistic* berisi informasi mengenai penggunaan energi dan biaya selama periode tertentu. Periode dapat diubah – ubah oleh *user*, hal ini dilakukan supaya *user* dapat melihat data pada periode tertentu dengan cepat, tanpa perlu mencarinya terlebih dahulu. Pada saat membuka halaman *statistic*, *default* periode adalah 1 bulan, jadi grafik menampilkan data selama 1 bulan.

## 6. Halaman *Settings*



Gambar 4.9 Halaman *Settings*

Halaman *Settings* ini hanya dapat diakses oleh *Admin* dan *Super Admin* saja. Hal ini dilakukan karena dalam halaman *Settings* ini terdapat informasi yang penting, yaitu informasi mengenai biaya per kWh, *feature on Control*, fitur penghapusan otomatis, pengaturan mengenai protokol komunikasi yang digunakan dan juga penggantian *password*.

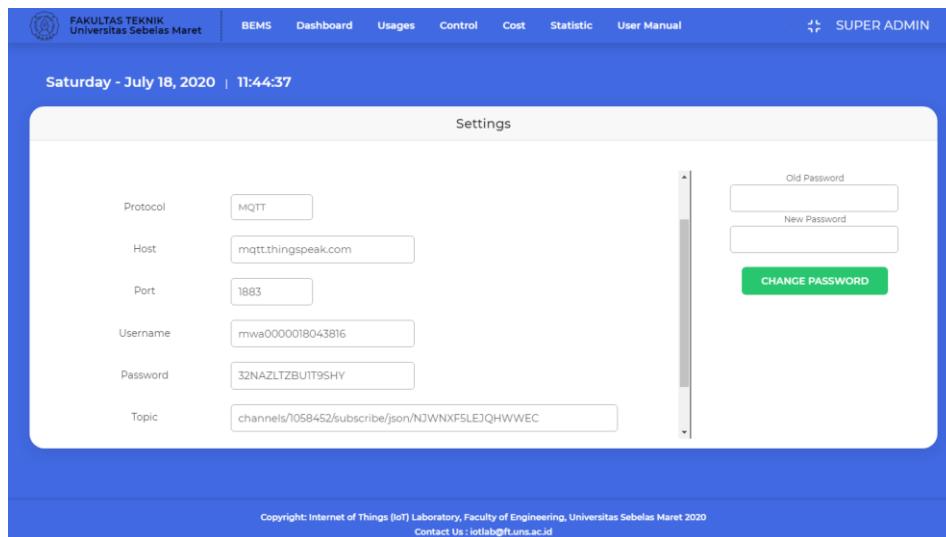
Biaya per kWh secara *default* bernilai 1352, *Admin* dan *Super Admin* dapat mengubah nilai tersebut. Biaya per kWh ini akan berpengaruh pada data yang akan masuk setelah pengubahan nilai ini. Jadi data yang sudah masuk sebelum pengubahan akan menggunakan nilai sebelum ini. Contoh sebelum pengubahan (biaya per kWh adalah 1352), data masuk dari *device* yang ada pada masing – masing ruang, yang menunjuk nilai 0,3 kWh. Maka *cost* yang akan didapatkan adalah 0,3 kWh dikali dengan 1352, yaitu sebesar 405,6. Setelah data masuk, ada pengubahan dalam halaman *settings* yang mengubah nilai biaya per kWh dari 1352 menjadi 1400. *Cost* sebesar 405,6 tadi tidak akan mengalami pengubahan dan akan tetap bernilai 405,6 tetapi setelah pengubahan ketika ada data yang masuk maka akan dikalikan dengan 1400, yang adalah nilai baru dari biaya per kWh.

*Feature on Control* adalah sebuah *setting* yang digunakan untuk mengatur fitur pada halaman *control*. Jika fitur ini diaktifkan, maka ketika *Control Master* pada lantai tertentu dihidupkan, *value/* nilai yang terdapat

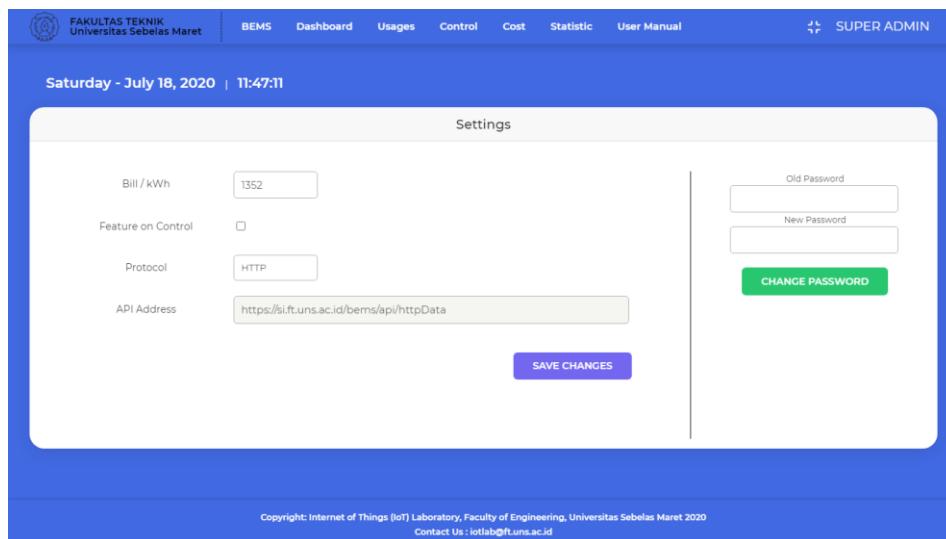
pada tiap *control* dalam tiap ruangan akan *direstore*/ dikembalikan seperti sebelum *Control Master* tersebut dimatikan. Contoh pada halaman *control* terdapat ruang 1 dan ruang 2 di lantai 1 yang memiliki nilai ON/ *control* ON dan setelah itu *Control Master* lantai 1 dimatikan, maka ketika *Control Master* lantai 1 dihidupkan kembali, hanya ruang 1 dan ruang 2 pada lantai 1 saja yang akan ON. Berbeda jika fitur dimatikan, ketika *Control Master* lantai 1 hidup maka semua ruang yang terdapat pada lantai 1 akan hidup. Fitur ini diterapkan supaya memudahkan *user* yang setiap hari hanya mengaktifkan ruang tertentu saja, selain memudahkan, fitur ini juga akan menghemat waktu *user*.

Fitur penghapusan otomatis merupakan sebuah fitur yang memudahkan *user* dalam mengelola data pada *database*. Dengan fitur ini, maka data yang ada pada *database* dapat terhapus dengan otomatis dan *user* dapat menentukan *interval* dari data yang akan dihapus tersebut. Jadi contoh *user* mengaktifkan fitur ini dan menginputkan *value* 12 pada *interval*, maka data yang berumur 12 bulan akan terhapus dari database. Fitur ini akan berjalan setiap sebulan sekali, yaitu pada tanggal 1 setiap bulannya.

Pengaturan mengenai protokol komunikasi yang digunakan dibagi menjadi 2 yaitu menggunakan protokol MQTT dan HTTP. Jika *user* memilih untuk menggunakan protokol MQTT, maka *user* harus menginputkan beberapa pengaturan pada protokol MQTT, yaitu berupa *Host*, *Port*, *Username*, dan *Password*. Hal ini dilakukan supaya MQTT pada BEMS dapat terkoneksi dengan benar dengan MQTT *Publisher*. Jika *user* memilih untuk menggunakan protokol HTTP, maka *user* dapat melihat API *Address* yang dapat digunakan sebagai url penerimaan data dari *hardware*. Dengan pengaturan mengenai protokol komunikasi ini diharapkan BEMS dapat secara fleksibel melakukan penerimaan data dari *hardware* yang ada.



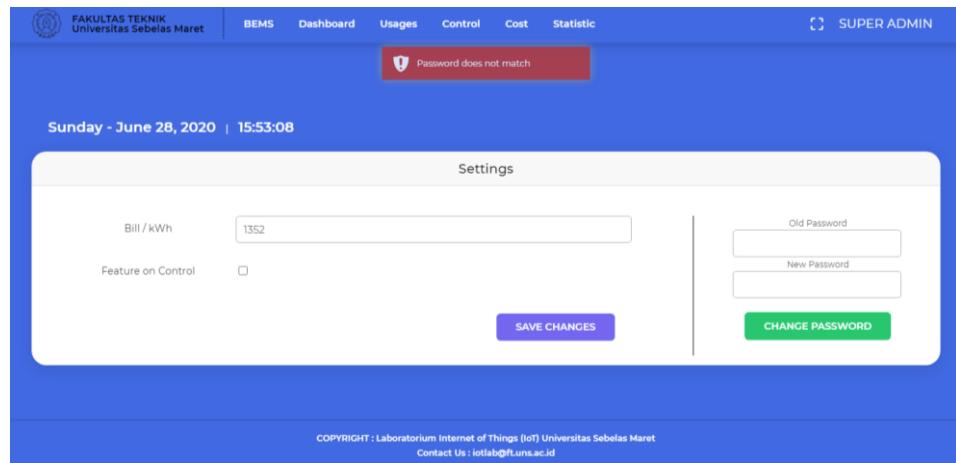
Gambar 4.10 Tampilan ketika *user* memilih protokol MQTT



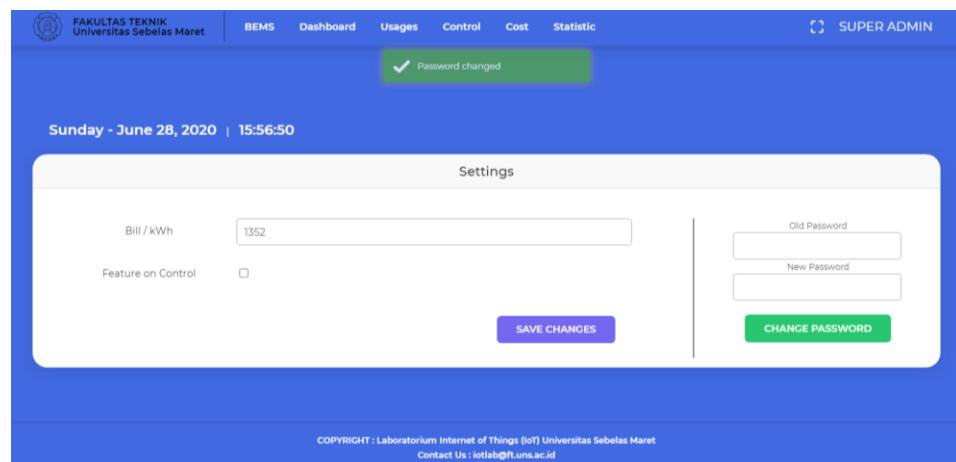
Gambar 4.11 Tampilan ketika *user* memilih protokol HTTP

Penggantian *password* di halaman *settings* ini membutuhkan 2 inputan/ masukan, yaitu *password* lama dan juga *password* baru. Hal ini dilakukan sebagai konfirmasi bahwa yang mengganti *password* adalah pemilik akun sendiri/ *user* sendiri. Jika *user* salah menginputkan *password* lama, maka akan ada notifikasi yang memberitahu *user* bahwa *user* salah menginputkan *password*.

Begitu juga ketika *user* berhasil mengubah *password*, maka *user* akan melihat notifikasi bahwa pengubahan *password* telah berhasil dilakukan.

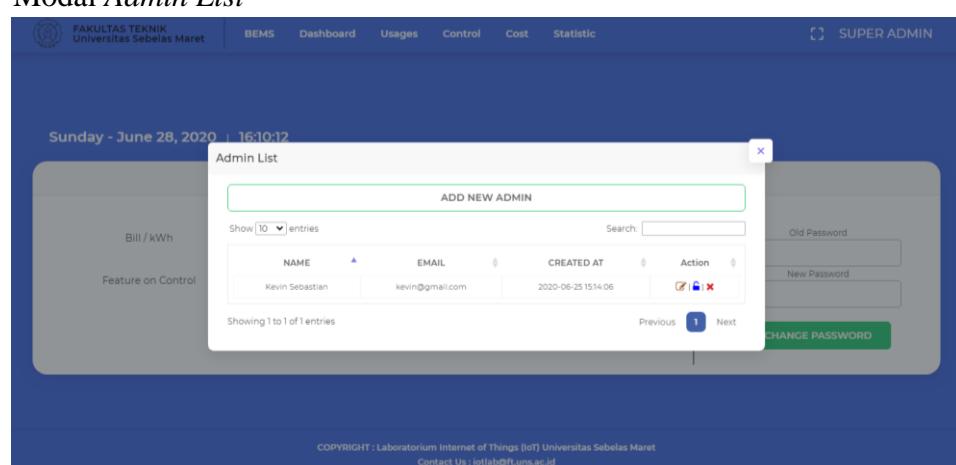


Gambar 4.12 Notifikasi yang akan muncul ketika *user* salah menginputkan *password* lama



Gambar 4.13 Notifikasi ketika *user* berhasil mengubah *password*

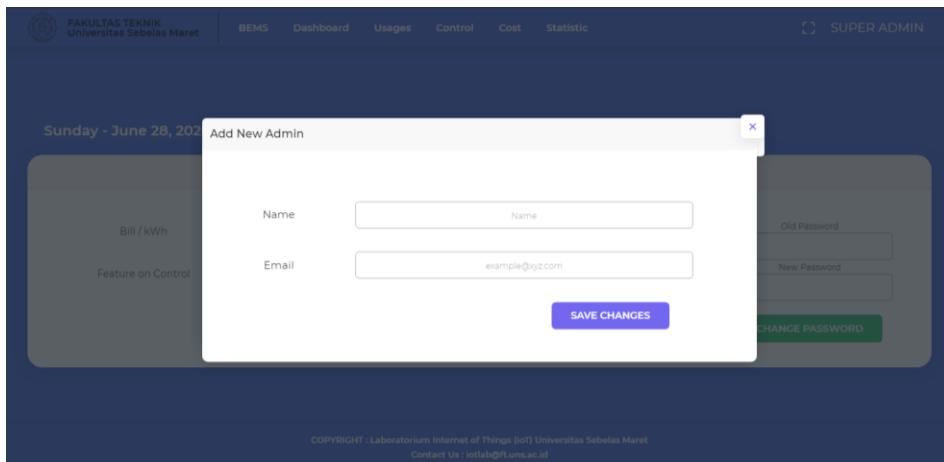
## 7. Modal Admin List



Gambar 4.14 Modal Admin List

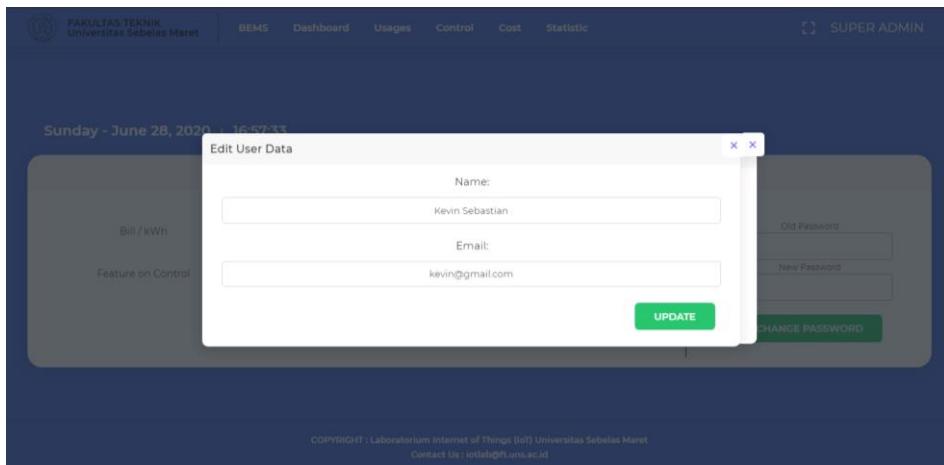
Modal ini hanya dapat diakses oleh *Super Admin* saja. Seperti namanya, modal ini berisi informasi mengenai semua *admin* yang ada. Informasi tersebut yaitu mengenai nama, *email*, tanggal pembuatan, dan tombol – tombol *action* seperti *edit*, lupa *password* dan *delete admin*.

Di modal ini juga terdapat tombol untuk membuat *Admin* baru, yang terletak di bagian atas modal. Jika tombol tersebut ditekan maka akan muncul modal lagi yang berisi form tentang nama dan *email* dari *admin* yang akan dibuat, *Admin* yang dibuat akan memiliki *default password* yaitu admin123. Hal ini dilakukan supaya memudahkan *Super Admin* dalam pembuatan *Admin* baru.



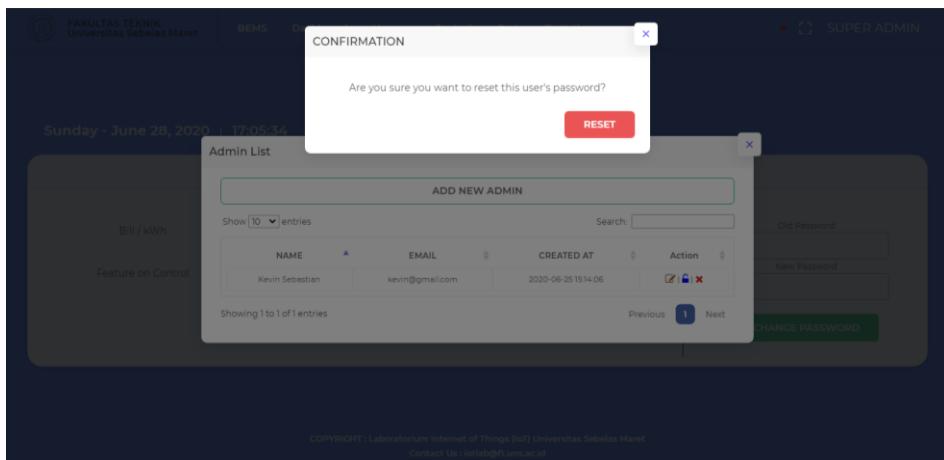
Gambar 4.15 Tampilan modal ketika tombol tambah *admin* baru ditekan

Tombol *edit* memiliki fungsi untuk mengubah informasi mengenai *admin*. Informasi yang diubah adalah tentang nama dan *email* dari *admin* tersebut, *password* dapat diubah oleh *admin* sendiri. Hal ini dilakukan supaya memudahkan *admin* dalam pengubahan *password* tanpa harus melalui *super admin*.



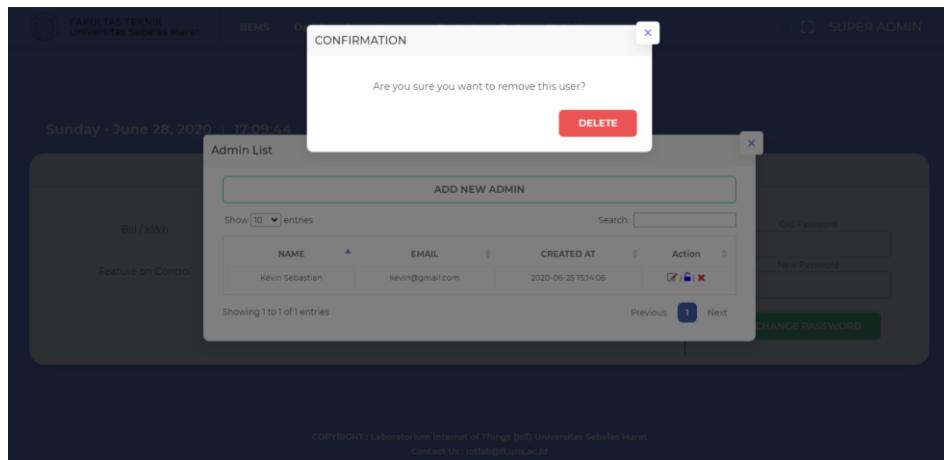
Gambar 4.16 Tampilan modal ketika tombol *edit* ditekan

Tombol lupa *password* digunakan untuk mereset *password* menjadi *default password*. Hal ini dilakukan supaya membantu *admin* yang lupa dengan *password* yang dimilikinya. Ketika *super admin* menekan tombol ini, maka akan muncul konfirmasi, hal ini dilakukan untuk mencegah *super admin* yang tidak sengaja dalam menekan tombol.



Gambar 4.17 Tampilan konfirmasi ketika menekan tombol lupa *password*

Tombol *delete* digunakan untuk menghapus *admin*, juga terdapat konfirmasi yang akan keluar jika menekan tombol *delete*. Hal ini dilakukan untuk mencegah *super admin* yang tidak sengaja menekan tombol.



Gambar 4.18 Tampilan konfirmasi ketika menekan tombol *delete*

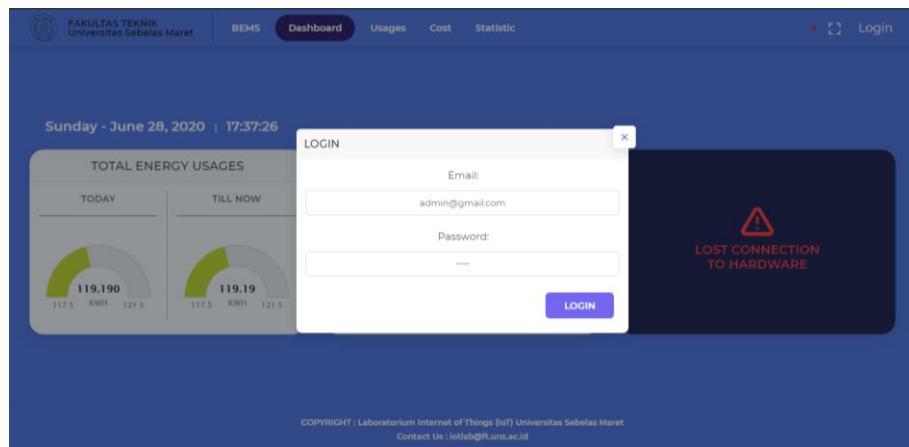
## 4.2. Pengujian Sistem menggunakan metode *Black Box*

Pengujian sistem ini dilakukan untuk memastikan apakah sistem berjalan dengan semestinya dan sesuai dengan fungsi atau tujuan yang diinginkan. Pengujian sistem ini dilakukan dalam rangka mencari adanya *bug*, kesalahan dalam pemrograman, dan ketidaksesuaian program terhadap tujuan yang diinginkan.

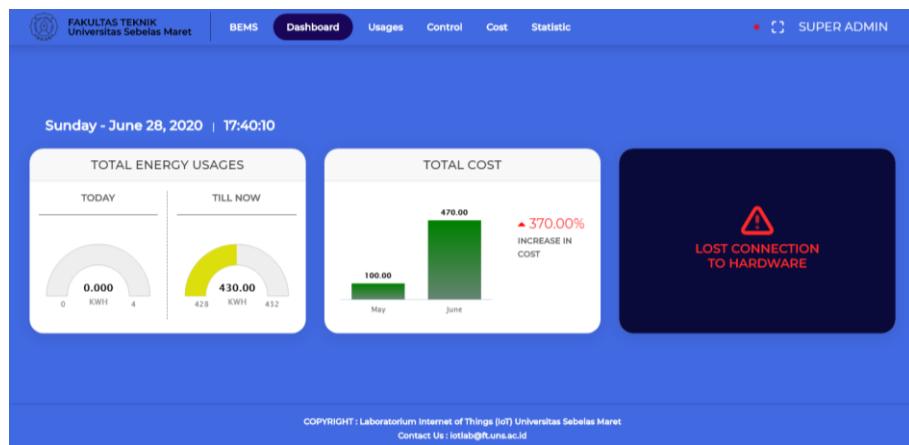
Pengujian sistem dilakukan dengan menguji setiap proses dan kesalahan yang mungkin terjadi. Pengujian sistem *Black Box* memfokuskan fungsi – fungsi, masukan, dan keluaran dari sistem apakah sesuai dengan yang diinginkan.

### 4.2.1. Proses *Login*

Proses *Login* adalah proses dimana *user* memasukkan/ menginputkan *email* dan *password* ke dalam modal *Login* dan akan *redirect* ke dalam halaman *dashboard* ketika *user* menginputkan data dengan benar. *Login* disini dapat dilakukan dengan menggunakan akun *admin* atau *super admin*.

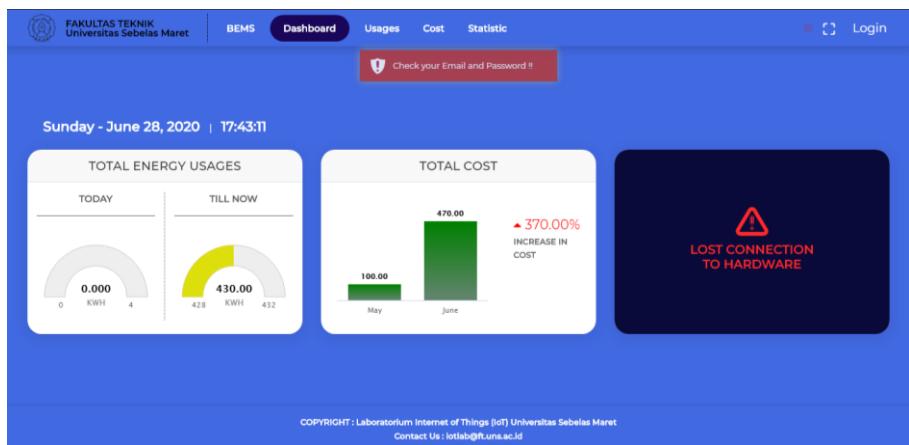


Gambar 4.19 User menginputkan di Modal Login



Gambar 4.20 Redirect ke Halaman dashboard

Jika *user* salah menginputkan *email* ataupun *password*, maka akan muncul sebuah notifikasi yang memberitahu *user* untuk mengecek *email* dan *password*.



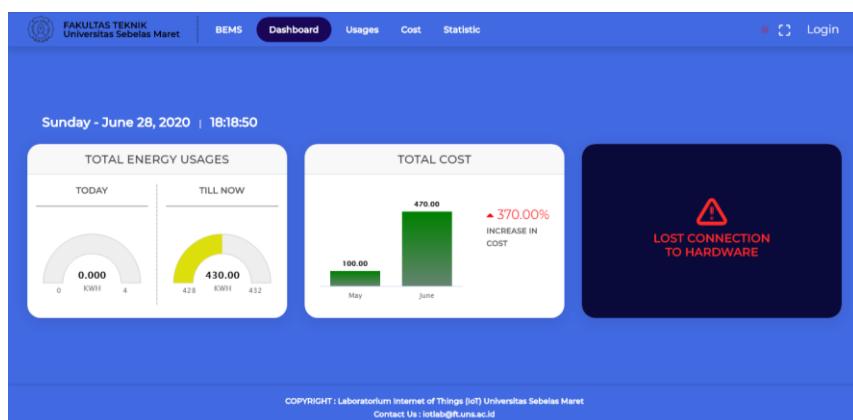
Gambar 4.21 Tampilan notifikasi ketika user salah menginputkan email atau password

#### 4.2.2. Proses Menampilkan Data

Proses menampilkan data ini merupakan proses pengambilan data dari *database* dan ditampilkan pada halaman yang ada pada *website*. Pengujian ini dilakukan untuk menguji apakah data yang dikeluarkan benar dan sesuai dengan yang diinginkan. Pengujian dilakukan untuk setiap halaman yang menampilkan data.

##### 1. Halaman *dashboard*

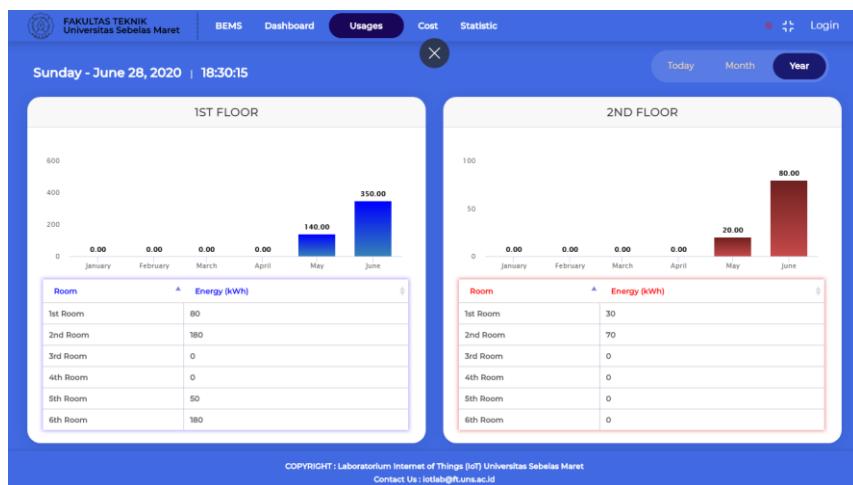
Di halaman *dashboard*, data diambil/ ditampilkan secara real time, dan dibagi menjadi 3.



Gambar 4.22 Proses pengujian menampilkan data ke halaman *dashboard*

##### 2. Halaman *usages*

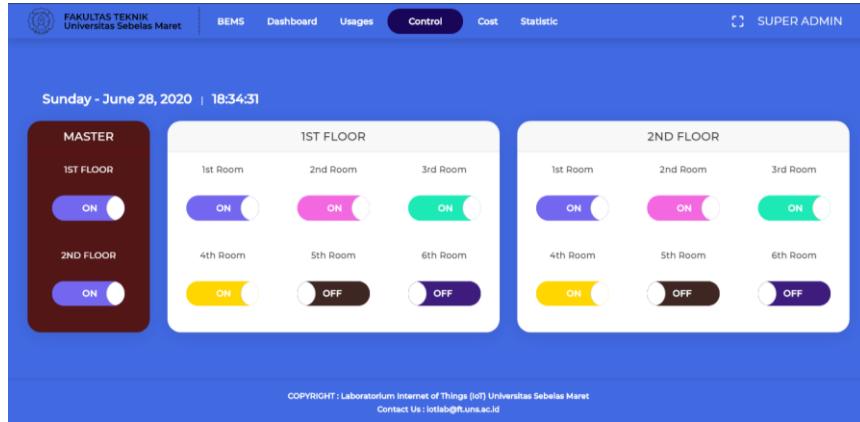
Di halaman *usages*, data yang ditampilkan dibagi menjadi 3 periode, yaitu *Today*, *Month*, dan *Year*.



Gambar 4.23 Proses pengujian menampilkan data ke halaman *usages*

### 3. Halaman *control*

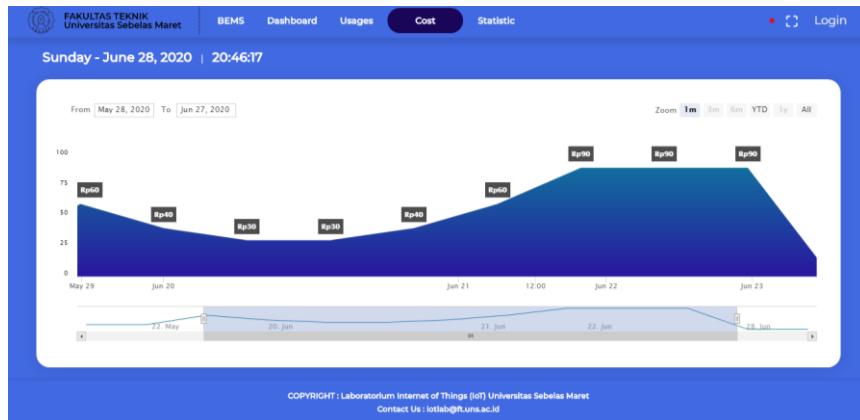
Data yang ditampilkan pada halaman *control* ini berupa nilai ON atau OFF, yang digambarkan dengan tombol ON/ OFF.



Gambar 4.24 Proses pengujian menampilkan data ke halaman *control*

### 4. Halaman *cost*

Data yang ditampilkan pada halaman *cost* ini berupa grafik dan diambil dari awal data.



Gambar 4.25 Proses pengujian menampilkan data ke halaman *cost*

### 5. Halaman *statistic*

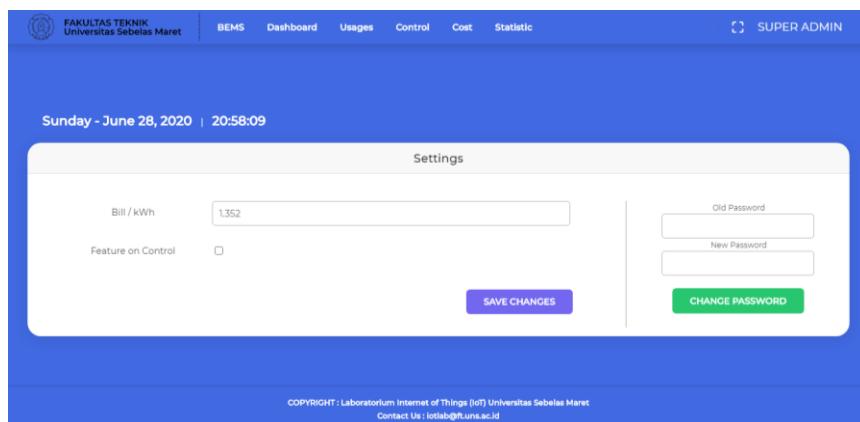
Data yang ditampilkan pada halaman *statistic* ini berupa grafik dan dibagi menjadi 2.



Gambar 4.26 Proses pengujian menampilkan data ke halaman *statistic*

## 6. Halaman *settings*

Data yang ditampilkan berupa value yang bisa diubah oleh *user*.



Gambar 4.27 Proses pengujian menampilkan data ke halaman *statistic*

## 7. Admin List

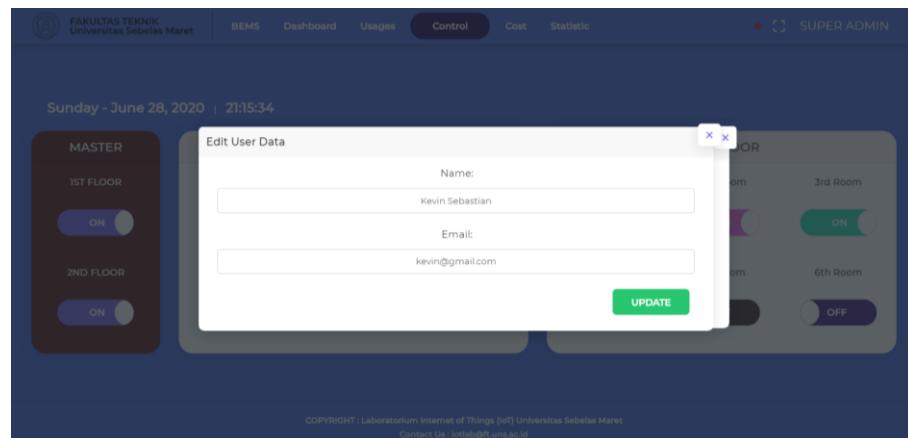
Data yang ditampilkan berupa tabel.

Admin List					
ADD NEW ADMIN Show 10 entries Search: <input type="text"/> NAME: <input type="text"/> EMAIL: <input type="text"/> CREATED AT: <input type="text"/> Action: <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>					
Bill / kWh	Kevin Sebastian	kevin@gmail.com	2020-06-28 15:02:37	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Feature on Control				<input type="checkbox"/>	<input type="checkbox"/>
Showing 1 to 1 of 1 entries Previous <input type="button" value="1"/> Next <input type="button" value="2"/>					

Gambar 4.28 Proses pengujian menampilkan data ke tabel *Admin List*

#### 4.2.3. Proses Pembaharuan Data

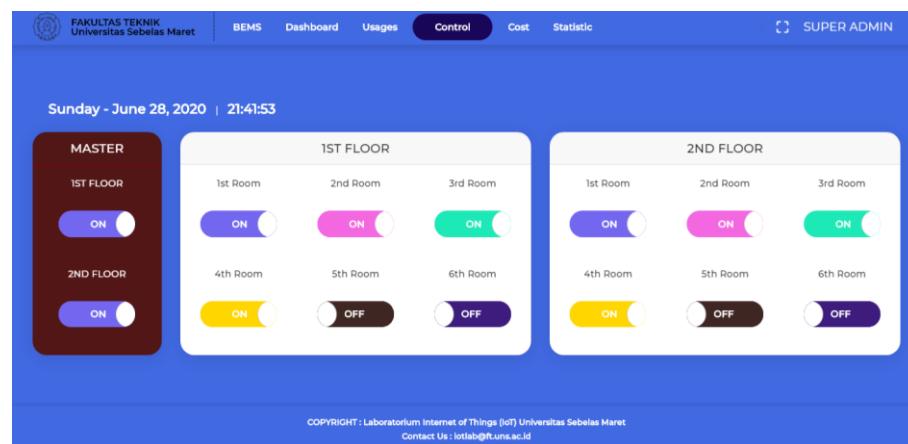
Proses pembaharuan data adalah proses mengubah data dan menyimpan ke dalam *database*. Pengujian ini dilakukan untuk mengetahui apakah *user* dapat mengubah data yang telah ditampilkan. Pada BEMS, proses pembaharuan data terjadi di halaman *settings* dan pengubahan data *Admin*.



Gambar 4.29 Proses pengujian pembaharuan data *admin*

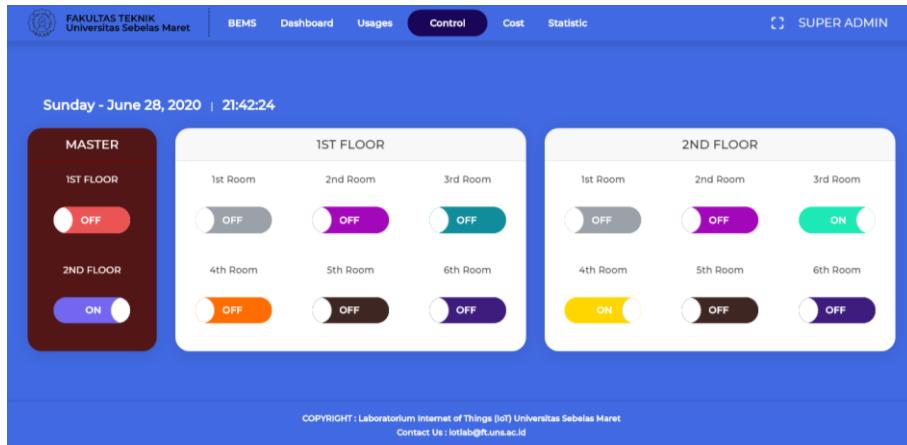
#### 4.2.4. Proses Pengontrolan Device

Proses pengontrolan *device* adalah proses mengontrol *device* pada tiap ruang di setiap lantai yang dapat dilakukan dari *website* BEMS. Pengujian ini dilakukan untuk mengetahui apakah fungsi *control* pada sistem informasi ini berjalan sesuai dengan yang semestinya atau tidak.



Gambar 4.30 Proses pengujian pengontrol *device* sebelum *control* dilakukan

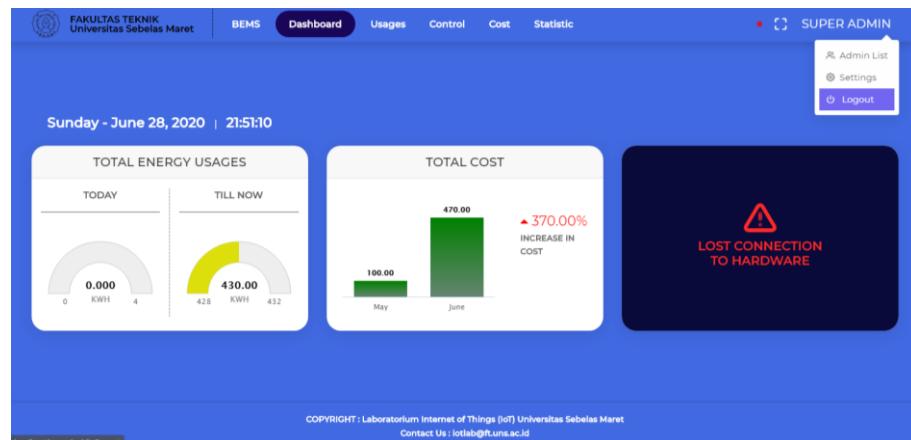
Proses pengontrol *device* ini hanya terjadi di halaman *control*. Pengontrolan ini langsung terhubung dengan *device* yang ada pada tiap ruang.



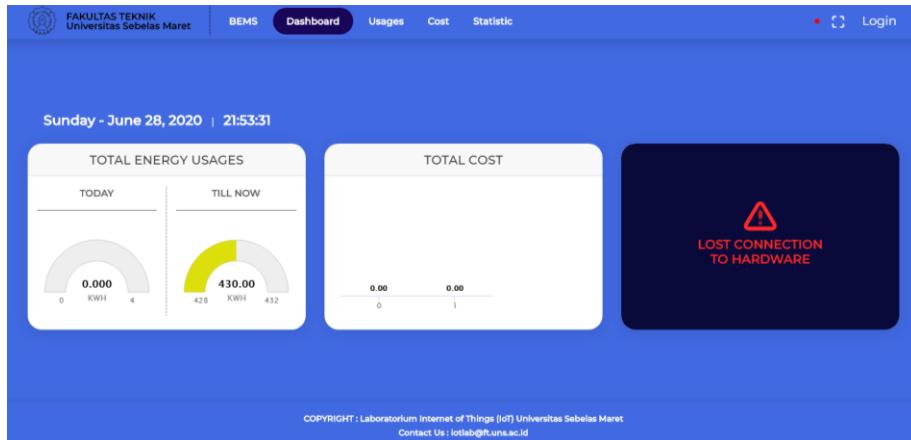
Gambar 4.31 Proses pengujian pengontrol *device* setelah *control* dilakukan

#### 4.2.5. Proses *Logout*

Proses *Logout* adalah proses keluar dari sistem BEMS. Proses pengujian ini dilakukan untuk mengetahui apakah *user* dapat keluar dari sistem atau tidak.



Gambar 4.32 Menu *Logout*



Gambar 4.33 Proses pengujian *logout*

#### 4.3. Pengujian Sistem menggunakan metode *User Acceptance Testing* (UAT)

Pengujian sistem ini dilakukan untuk mengetahui apakah sistem telah berjalan sesuai yang diharapkan dan memenuhi tujuan sistem tersebut dibuat atau tidak. Pengujian sistem dengan menggunakan metode UAT umumnya dilakukan oleh *client* atau pengguna akhir (*end user*). Dari setiap aspek pertanyaan yang ada pada UAT akan dianalisis dengan membuat persentase pada tiap aspek pertanyaan yang ada. Pengujian ini dilakukan oleh 20 orang/responden dengan menggunakan Google *Form*.

Tabel 4.1 Bobot Nilai pada tiap Jawaban

Jawaban	Bobot
SA: Strongly Agree	5
A: Agree	4
N: Netral	3
D: Disagree	2
SD: Strongly Disagree	1

Tabel 4.2 Data Jawaban yang diberikan

No.	Pertanyaan	SA	A	N	D	SD
1	Apakah tampilan sistem informasi ini menarik?	6	12	2	0	0

2	Apakah sistem informasi ini mudah dipahami dan digunakan?	5	13	2	0	0
3	Apakah sistem ini memiliki performa yang bagus?	4	12	4	0	0
4	Apakah sistem ini memenuhi fungsi <i>monitoring</i> ?	8	11	1	0	0
5	Apakah sistem ini memenuhi fungsi sebagai <i>controller</i> ?	7	8	4	0	1
6	Apakah sistem ini memiliki fitur yang berguna?	10	9	1	0	0
7	Apakah sistem ini membantu dalam <i>monitoring</i> penggunaan energi listrik?	10	8	2	0	0
8	Apa saran Saudara/ri terhadap sistem ini?					

Tabel 4.3 Penghitungan Data berdasarkan bobot nilai pada tiap jawaban

No.	Pertanyaan	SA	A	N	D	SD	Jumlah
1	Apakah tampilan sistem informasi ini menarik?	30	48	6	0	0	84
2	Apakah sistem informasi ini mudah dipahami dan digunakan?	25	52	6	0	0	83
3	Apakah sistem ini memiliki performa yang bagus?	20	48	12	0	0	80
4	Apakah sistem ini memenuhi fungsi <i>monitoring</i> ?	40	44	3	0	0	87
5	Apakah sistem ini memenuhi fungsi sebagai <i>controller</i> ?	35	32	12	0	1	79
6	Apakah sistem ini memiliki fitur yang berguna?	50	36	3	0	0	89
7	Apakah sistem ini membantu dalam <i>monitoring</i> penggunaan energi listrik?	50	32	6	0	0	88

8	Apa saran Saudara/ri terhadap sistem ini?	
---	---	--

#### 4.3.1. Analisis pada Tiap Pertanyaan

1. Tampilan sistem informasi ini menarik

Pertanyaan pertama memiliki jumlah 84 dari 20 responden yang berarti memiliki persentase sebesar 84%. Hasil ini didapat dari:

- Membagi jumlah skor dari tiap pertanyaan dan membaginya dengan jumlah skor ideal dari tiap pertanyaan dan mengalikan dengan 100%
- Jumlah skor ideal dari tiap pertanyaan didapatkan dari mengalikan 5 dengan jumlah responden, yang akan mendapatkan nilai 100
- Jadi penghitungannya menjadi  $\frac{84}{100} \times 100\%$  yang memiliki hasil yaitu 84%

Dari persentase yang didapatkan, dapat diketahui bahwa sistem informasi ini memiliki tampilan yang menarik karena memiliki persentase sebesar 84%.

2. Sistem informasi ini mudah dipahami dan digunakan

Pertanyaan kedua ini memiliki jumlah 83 dari 20 responden yang berarti memiliki persentase sebesar 83%, yang didapatkan dari  $\frac{83}{100} \times 100\% = 83\%$ . Dari hasil persentase tersebut dapat diketahui bahwa 83% dari responden setuju bahwa sistem informasi ini mudah dipahami dan digunakan

3. Sistem informasi ini memiliki performa yang bagus

Pertanyaan ketiga ini memiliki jumlah 80 dari 20 responden yang berarti memiliki persentase sebesar 80%, yang didapatkan dari  $\frac{80}{100} \times 100\% = 80\%$ . Dari hasil persentase tersebut dapat diketahui bahwa 80% dari responden setuju bahwa sistem informasi ini memiliki performa yang bagus.

4. Sistem informasi ini memenuhi fungsi *monitoring*

Pertanyaan keempat ini memiliki jumlah 87 dari 20 responden yang berarti memiliki persentase sebesar 87%, yang didapatkan dari  $\frac{87}{100} \times 100\% = 87\%$ . Dari hasil persentase tersebut dapat diketahui bahwa 87% dari responden setuju bahwa sistem informasi ini memenuhi fungsi *monitoring*.

5. Sistem informasi ini memenuhi fungsi sebagai *controller*

Pertanyaan kelima ini memiliki jumlah 79 dari 20 responden yang berarti memiliki persentase sebesar 79%, yang didapatkan dari  $\frac{79}{100} \times 100\% = 79\%$ . Dari hasil persentase tersebut dapat diketahui bahwa 79% dari responden setuju bahwa sistem informasi ini memenuhi fungsi sebagai *controller*.

6. Sistem informasi ini memiliki fitur yang berguna

Pertanyaan keenam ini memiliki jumlah 89 dari 20 responden yang berarti memiliki persentase sebesar 89%, yang didapatkan dari  $\frac{89}{100} \times 100\% = 89\%$ . Dari hasil persentase tersebut dapat diketahui bahwa 89% dari responden setuju bahwa sistem ini memiliki fitur yang berguna.

7. Sistem informasi ini membantu dalam *monitoring* penggunaan energi listrik

Pertanyaan ketujuh ini memiliki jumlah 88 dari 20 responden yang berarti memiliki persentase sebesar 88%, yang didapatkan dari  $\frac{88}{100} \times 100\% = 88\%$ . Dari hasil persentase tersebut dapat diketahui bahwa 88% dari responden setuju bahwa sistem informasi ini membantu dalam *monitoring* penggunaan energi listrik.

8. Apa saran saudara/ri terhadap sistem ini

Pertanyaan ini tidak memiliki nilai karena mengandung *feedback* dari *user*. *Feedback* ini digunakan sebagai masukan pada sistem informasi. *Feedback* yang didapatkan bermacam-macam, dari yang

membahas mengenai desain, keresponsive-an sistem sampai ke hal-hal mendetail seperti pemberian informasi keterangan terhadap suatu nilai yang ditampilkan pada sistem.

#### 4.3.2. Analisa Akhir dari UAT

Tabel 4.4 Hasil akhir dari UAT

No.	Indikator	(%)	Keterangan
1	Desain	84%	Setuju
2	Kemudahan	81,5%	Setuju
3	Fungsi/ Kegunaan	85,75%	Setuju

Hasil yang ada pada tabel 4.4 didapatkan dari penjumlahan persentase tiap pertanyaan dengan indikator yang sama lalu membagi dengan jumlah pertanyaan yang ada pada indikator tersebut.

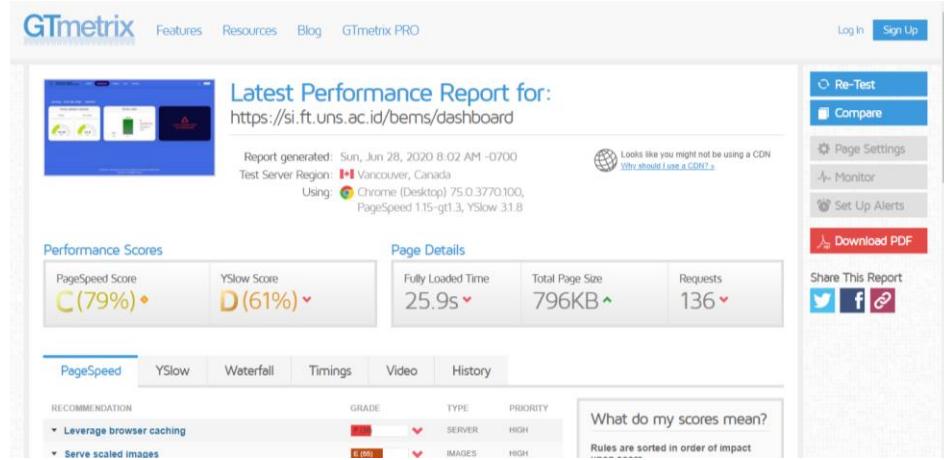
Hasil yang didapatkan untuk indikator desain adalah  $\frac{84}{1} \times 100\% = 84\%$ , untuk indikator kemudahan adalah  $\frac{83+80}{2} \times 100\% = 81,5\%$ , untuk indikator fungsi/ kegunaan adalah  $\frac{87+79+89+88}{4} \times 100\% = 85,75\%$ .

Dari analisa yang telah dilakukan diperoleh hasilnya yaitu dari 20 responden yang telah melakukan pengujian UAT, memberikan respon yang positif pada sistem informasi ini, baik itu dari sisi desain (pertanyaan 1), sisi kemudahan (pertanyaan 2-3), sisi fungsi/ kegunaan (pertanyaan 4-7), yaitu masing-masingnya sebesar 84%, 81,5%, dan 85,75% yang berarti responden setuju mengenai pertanyaan yang telah diberikan.

#### 4.4. Performa Sistem Informasi Building Energy Management System

Performa website BEMS dicek melalui website khusus yang digunakan untuk melihat performa suatu website dan memberikan saran bagaimana meningkatkan performa pada website.

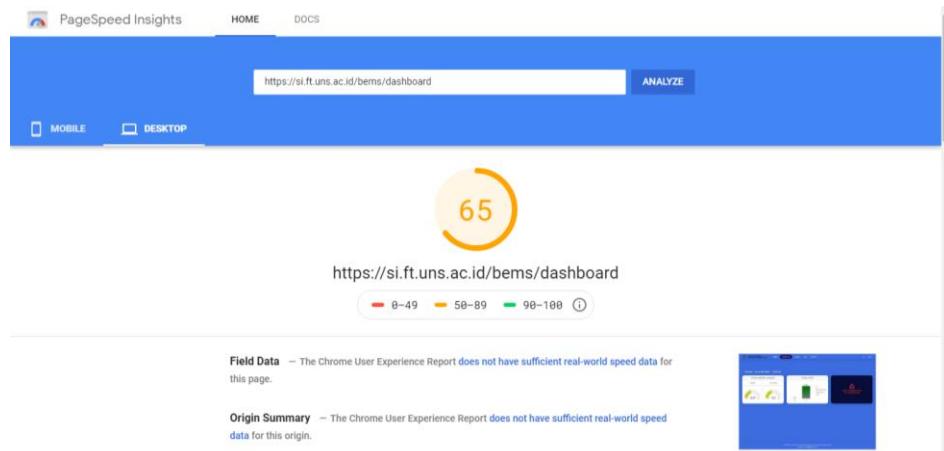
## 1. GTmetrix



Gambar 4.34 Hasil uji pada website GTmetrix

Sumber: <https://gtmetrix.com/>

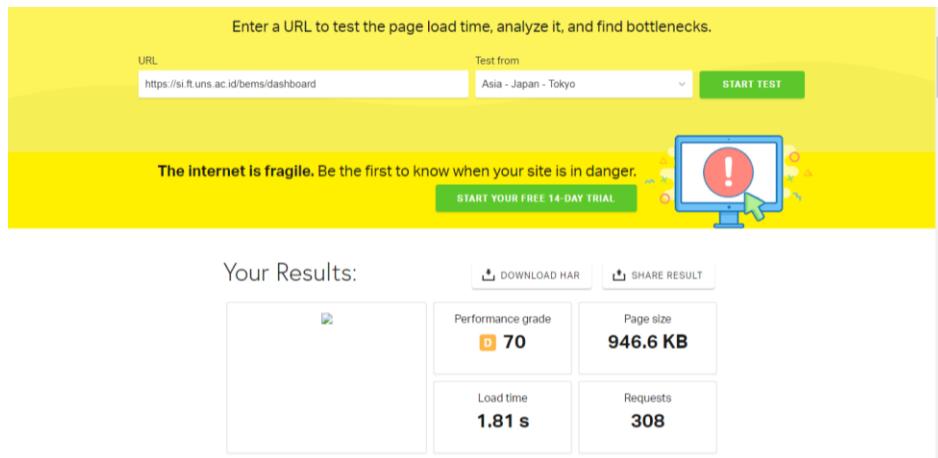
## 2. Google PageSpeed Insight



Gambar 4.35 Hasil uji pada website Google PageSpeed Insight

Sumber: <https://developers.google.com/speed/pagespeed/insights/>

### 3. Pingdom



Gambar 4.36 Hasil uji dari *website* Pingdom

Sumber: <https://tools.pingdom.com/>

Dari hasil uji dari beberapa *website*, diperoleh hasil yang bervariasi. Hasil yang diperoleh ini dipengaruhi oleh beberapa hal, antara lain:

1. Karena pada *website* BEMS, sistem akan melakukan *request* setiap detik ke API untuk mendapatkan data secara *real time*. Hal ini dapat dilihat pada hasil uji pada 3 *website* yang berbeda.
2. Terdapat banyaknya *file javascript* untuk tiap halaman yang menyebabkan sistem harus mengeloa*n* *file* tersebut terlebih dahulu.

Pada *website* Pingdom dapat kita lihat bahwa *Load time* untuk *website* BEMS ini terbilang bagus, karena hanya membutuhkan waktu 1,81 detik sampai halaman terbuka dengan sempurna.

## **BAB V**

### **KESIMPULAN DAN SARAN**

Bab ini berisi mengenai kesimpulan dari keseluruhan penelitian dan terdapat saran untuk penelitian selanjutnya supaya dapat menghasilkan penelitian yang lebih berkualitas dan berbobot.

#### **5.1. Kesimpulan**

Dari penjelasan dan pembahasan yang sudah dilakukan pada bab-bab sebelumnya, dapat disimpulkan beberapa hal, sebagai berikut:

1. Sistem Informasi *Building Energy Management System* (BEMS) dibuat dengan menggunakan *desain* yang diajukan atau diminta oleh *user*. Dengan beberapa perbaikan pada sistem yang sesuai dengan permintaan *user*.
2. Sistem Informasi *Building Energy Management System* (BEMS) diimplementasikan pada suatu bangunan dengan terdapat suatu monitor yang dapat menampilkan sistem informasi ini. Implementasi yang dilakukan sesuai dengan permintaan dari *user*.
3. Hasil pengujian pada Sistem Informasi *Building Energy Management System* (BEMS) dengan menggunakan pengujian *User Acceptance Testing* (UAT) menunjukkan respon yang positif dari *user* dan hasil pengujian *Black Box* yang menunjukkan bahwa fungsi sistem berjalan dengan baik.

#### **5.2. Saran**

Adapun beberapa saran untuk pengembangan sistem selanjutnya supaya mendapatkan hasil yang lebih berkualitas lagi, diantaranya:

1. Membuat Sistem Informasi *Building Energy Management System* menjadi responsive, sehingga *user* dapat melihat *update* penggunaan energi dari *mobile* atau *gadget* yang lainnya.
2. Membuat fitur *real time* menjadi lebih lagi, bisa dilakukan dengan menggunakan metode yang lain, seperti penggunaan teknologi *websocket*.

3. Memperbaiki *user interface* dan disesuaikan dengan kebutuhan *user* kedepannya.

## DAFTAR PUSTAKA

- [1] Lestari R. M., Baihaqi I., Persada S. F., “Praktik Manajemen Energi pada Industri Manufaktur”, *Jurnal Teknik ITS*, Vol. 7, No. 1, 2018.
- [2] Putri M. P., Effendi H., “Implementasi Metode Rapid Application Development Pada Website Service Guide “Waterfall Tour South Sumatra”, *Jurnal SISFOKOM*, vol. 7 No. 2, September 2018.
- [3] Zainab H. A., Hesham A. A., Mahmoud M. B., “Internet of Things (IoT): Definitions, Challenges and Recent Research Directions”, *International Journal of Computer Applications*, Vol. 128, No. 1, October 2015.
- [4] Sagdeo P., “Application Programming Interfaces and The Standardization-Value Appropriation Problem”, *Harvard Journal of Law & Technology*, Vol. 32, No. 1, Fall 2018.
- [5] Yu, H. R., “Design and Implementation of web based on Laravel framework”, *International Conference on Computer Science and Electronic Technology*, ICCSET 2014.
- [6] Alter, S., “Defining Information Systems as Work Systems: Implication for the IS Field”, *European Journal of Information Systems*, Oct 2018.
- [7] Al-Fedaghi, S., “Developing Web Applications”, *International Journal of Software Engineering and Its Applications*, Vol. 5 No.2, April, 2011.
- [8] Benmoussa, K., Laaziri, M., Khoulji, S., Kerkeb, M. L., Yamami, A. E., “A new model for the selection of web development frameworks: application to PHP frameworks”, *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 9 No. 1, February 2019.
- [9] Soegoto, E. S., “Implementing Laravel Framework website as brand image in higher-education institution”, *IOP Conf. Series: Materials Science and Engineering*, 2018
- [10] Bhojaraju G., M. M. Koganurmath., “Database Management : Concepts and Design”, *Database System : Concepts and Design*, February 2014.
- [11] Mishra, A., “Critical Comparison of PHP and ASP.NET For Web Development”, *International Journal of Scientific & Technology Research*, Vol. 3 No.7, July 2014.
- [12] Sharma, A., Aakanksha, “Introduction to HTML (Hyper Text Markup Language)”, *International Journal of Science and Research (IJSR)*, ISSN: 2319-7064.
- [13] Er. Saurabh Walia, Er. Satinderjit Kaur Gill, “A Framework for Web Based Student Record Management System using PHP”, *International Journal of Computer Science and Mobile Computing*, Vol. 3 No. 8, August 2014.
- [14] Jadhav, A., Jadhav, R., & Gavhane, S., “Implementation of Web Application Using Laravel Framework”, *International Journal of Engineering Sciences & Management Research*, ISSN: 2349-6193, May 2017.
- [15] Mahmood, M. T., Ashour, O. I., Ucan, O., Bayat, O., “Design and Impelementation of Web Based for Intermediate Online Shop with Laravel Framework”, *International Journal of Computer Science and Mobile Computing*, Vol. 8 No. 3, March 2019.

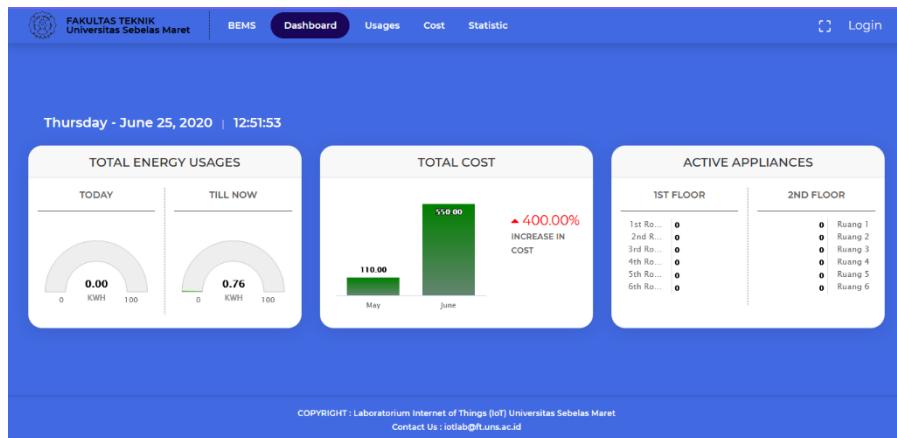
- [16] Meng M., Steinhardt S., & Schubert A., “Application Programming Interface Documentation: What do Software Developers Want?”, *Journal of Technical Writing and Communication*, Vol. 48 No. 3, July 2018.
- [17] Pahlevi O., Mulyani A., Khoir M., “Sistem Informasi Inventori Barang Menggunakan Metode Object Oriented di PT. Livaza Teknologi Indonesia Jakarta”, *Jurnal PROSISKO*, Vol. 5 No. 1, Maret 2018.
- [18] Silalahi M., Wahyudi D., “Perbandingan Performansi Database MongoDB dan MySQL dalam Aplikasi File Multimedia berbasis Web”, *Computer Based Information System Jorunal*, Vol. 6 No. 1, Februari 2018.
- [19] Utomo D. W., Kurniawan D., Astuti Y. P., “Teknik Pengujian Perangkat Lunak dalam Evaluasi Sistem Layanan Mandiri Pemantauan Haji pada Kementerian Agama Provinsi Jawa Tengah”, *Jurnal SIMETRIS*, Vol. 9, No. 2, November 2018.
- [20] Afyenni R., “Perancangan Data Flow Diagram untuk Sistem Informasi Sekolah (Studi Kasus pada SMA Pembangunan Laboratorium UNP)”, *Jurnal TEKNOIF*, Vol. 2 No.1, April 2014.
- [21] A. L. Yudanto, H. Tolle, and A. H. Brata, “Rancang Bangun Aplikasi Sistem Informasi Manajemen Laboratorium Biomedik Fakultas Kedokteran Universitas Brawijaya,”, *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, Vol. 1, No. 8, Juni 2017.
- [22] Efendi Y., “Internet of Things (IoT) Sistem Pengendalian Lampu menggunakan Raspberry Pi berbasis Mobile”, *Jurnal Ilmiah Ilmu Komputer*, Vol. 4, No. 1, April 2018.
- [23] Sharmad Pasha, “Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis”, *International Journal of New Technology and Research (IJNTR)*, Vol. 2, Issue. 6, June 2016.

# LAMPIRAN

## A. Petunjuk penggunaan untuk *GUEST*

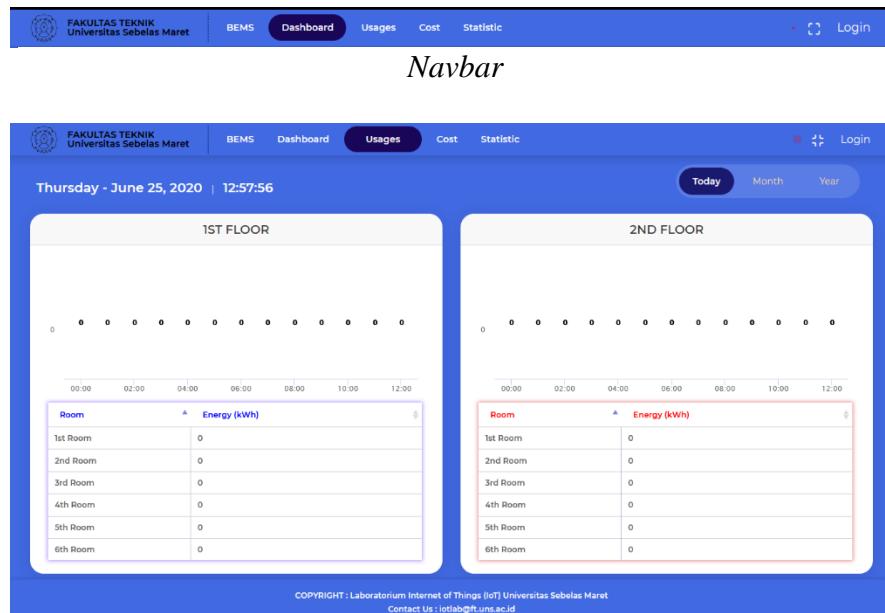
### A.1. Membuka *website BEMS* dan halaman – halamannya

1. Kunjungi *Building Energy Management System* di <https://si.ft.uns.ac.id/bems>.
2. Maka akan tampil halaman *dashboard*.

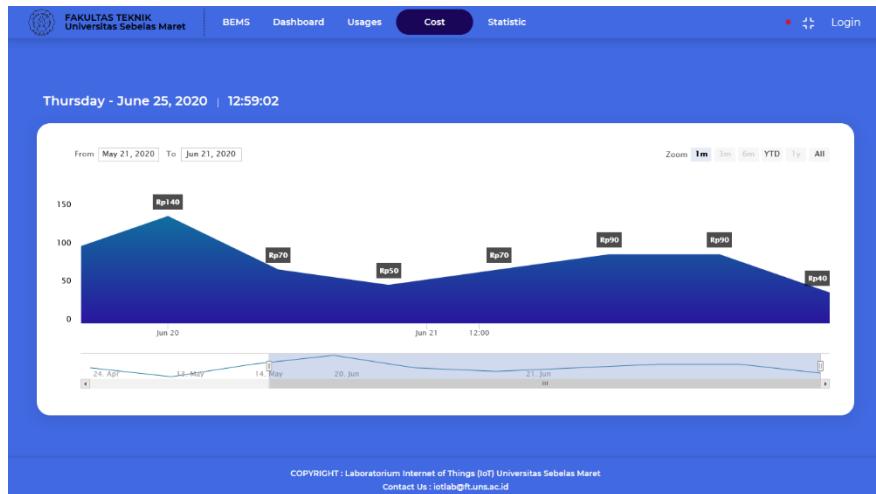


Halaman *Dashboard*

3. Terdapat halaman lainnya yang bisa diakses oleh *Guest*, yaitu halaman *Usages*, *Cost*, dan *Statistic*. Cara untuk mengakses halaman lainnya adalah dengan menekan link yang terdapat pada *navbar* (*navigation bar*).



Halaman *Usages*



Halaman *Cost*



Halaman *Statistic*

## A.2. Membuat *Website BEMS* menjadi layar penuh (*fullscreen*)

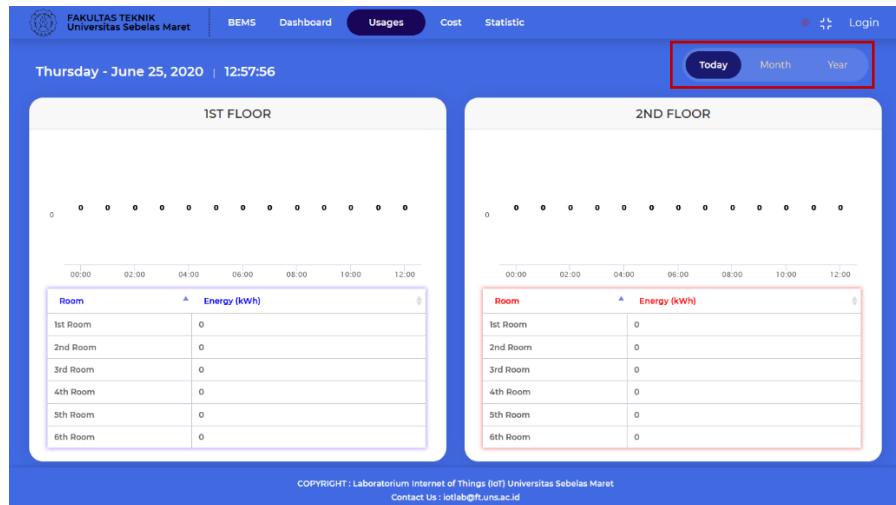
1. Menekan tombol *fullscreen* yang terdapat dalam *navbar*



Letak tombol *fullscreen*

## A.3. Melihat penggunaan Energi dalam 3 periode

1. Masuk ke dalam halaman *Usages*
2. Menekan *navigation button* (*Today*, *Month*, *Year*) yang terdapat di sebelas kanan atas halaman *Usages*



Letak *Navigation button*

#### A.4. Mengubah periode waktu pada halaman Cost

1. Dapat dilakukan dengan 3 cara, yaitu mengubah *range* dari waktu, menggunakan *navigation button* atau dengan menggunakan *navigator*.

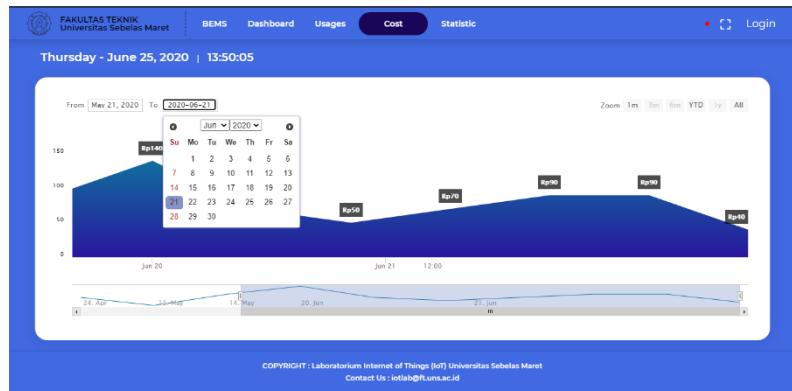
- **Mengubah *range* periode waktu**

- a. Menekan kotak range yang ada di sebelah kiri atas grafik
- b. Untuk mengubah *range* minimum adalah dengan menekan kotak *range* sebelah kiri



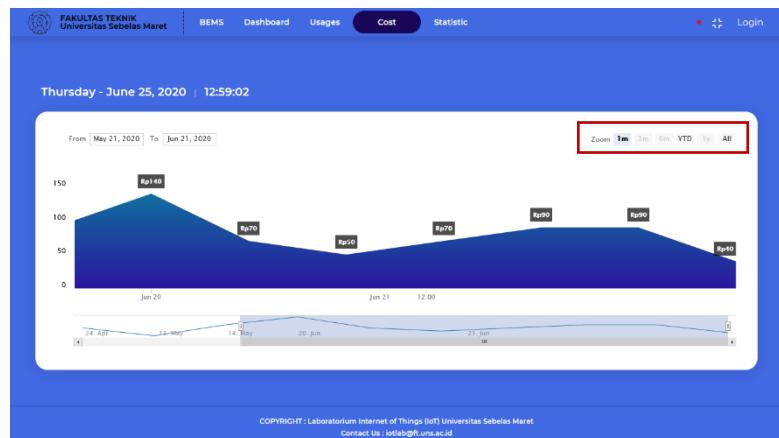
Mengubah *range* minimum

- c. Untuk mengubah *range* maksimum adalah dengan menekan kotak *range* sebelah kanan



Mengubah *range* maksimum

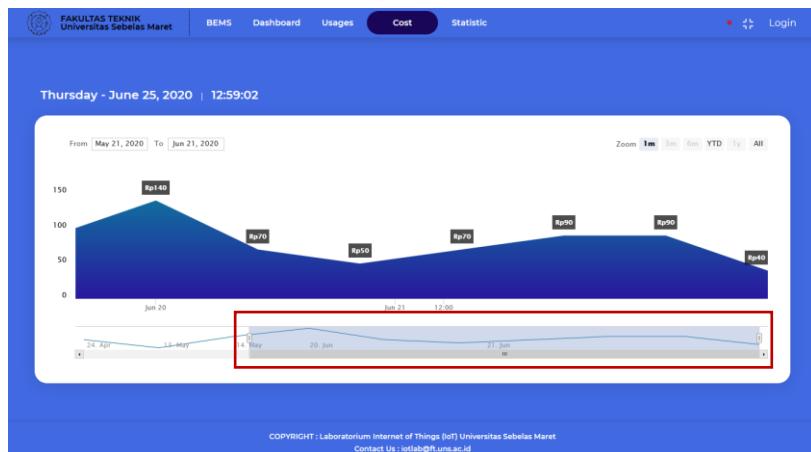
- Menggunakan **navigation button**
  - a. Menekan *navigation button* yang terdapat di sebelah kanan atas grafik
  - b. Terdapat 6 tombol dan terdapat tulisan untuk masing – masing tombolnya, yang menyimbolkan periode, yaitu:
    1. 1m : 1 bulan
    2. 3m : 3 bulan
    3. 6m : 6 bulan
    4. YTD : dari tanggal 1 Januari tahun ini sampai hari ini
    5. 1y : 1 tahun
    6. All : semua data



Letak *navigation button*

- Menggunakan **navigator**
  - a. *Navigator* terletak di bawah grafik
  - b. Mengubah *range* minimum dengan menarik batas kiri dari *navigator*

- c. Mengubah *range* maksimum dengan menarik batas kanan dari *navigator*
- d. Dapat mengubah *range* dengan menarik *navigator* pada bagian tengah
- e. Untuk dapat melihat secara lebih *detail* dari grafik tersebut, maka dapat mengecilkan *range* minimum dan maksimum dari *navigator*



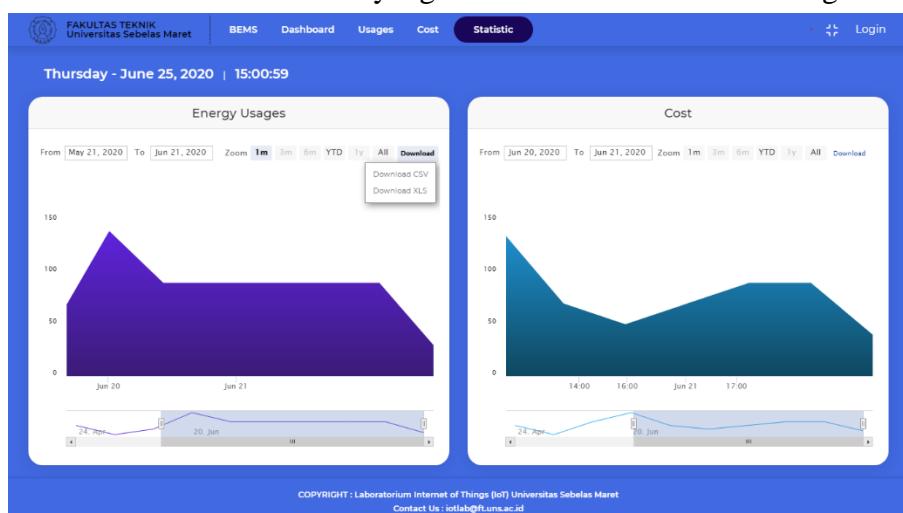
Letak *navigator*

#### A.5. Mengubah periode waktu pada halaman *Statistic*

Cara untuk mengubah periode waktu pada halaman *Statistic* sama dengan yang ada pada halaman *Cost*.

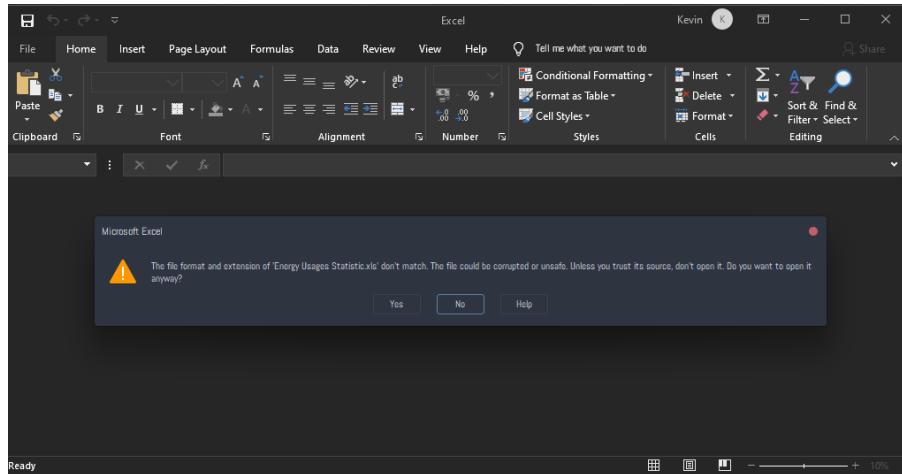
#### A.6. Cara mengeksport data ke dalam excel pada halaman *Statistic*

1. Menekan tombol *Download* yang ada di sebelah kanan atas dari grafik



Pilihan format pada tombol *Download*

2. Memilih format file, bisa berupa CSV atau XLS. Kedua format dapat dibuka di excel
3. Jika sudah berhasil mendownload file, buka file tersebut dan ketika ada peringatan yang terbuka maka pilih yes

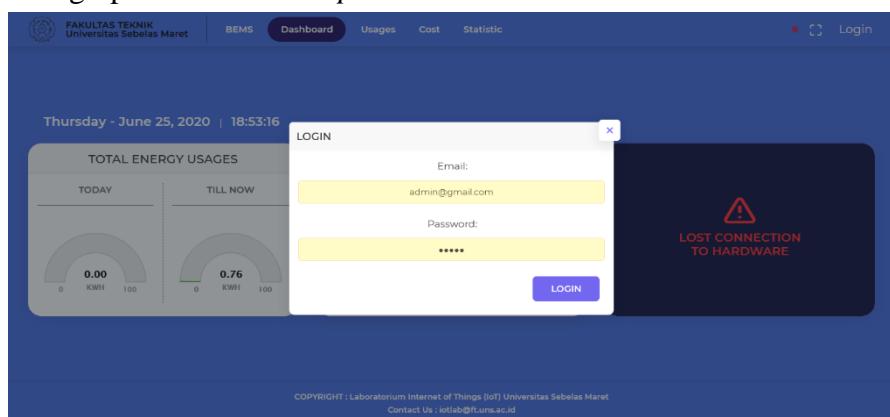


Peringatan pada saat membuka file

## B. Petunjuk Penggunaan untuk ADMIN dan SUPER ADMIN

### B.1. Cara Login di *Building Energy Management System*

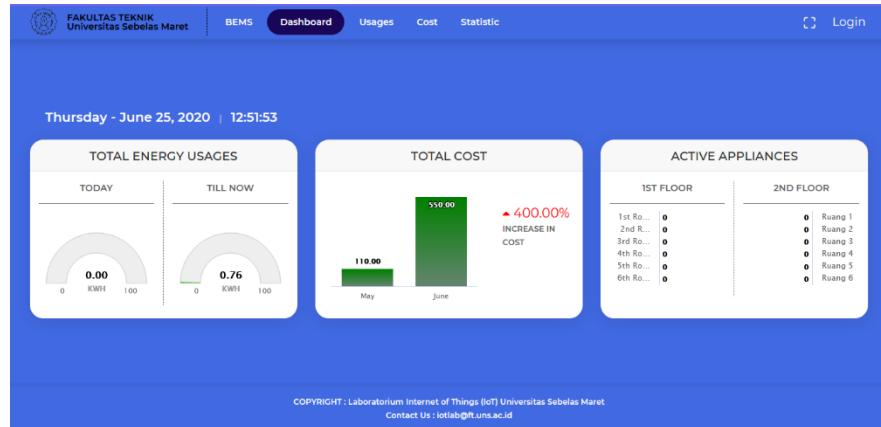
1. Kunjungi *Building Energy Management System* di <https://si.ft.uns.ac.id/bems>
2. Maka akan tampil halaman *dashboard*
3. Menekan tombol *Login* dan akan muncul *pop-up window* atau yang disebut modal.
4. Menginputkan *email* dan *password* dari user



*Login* modal

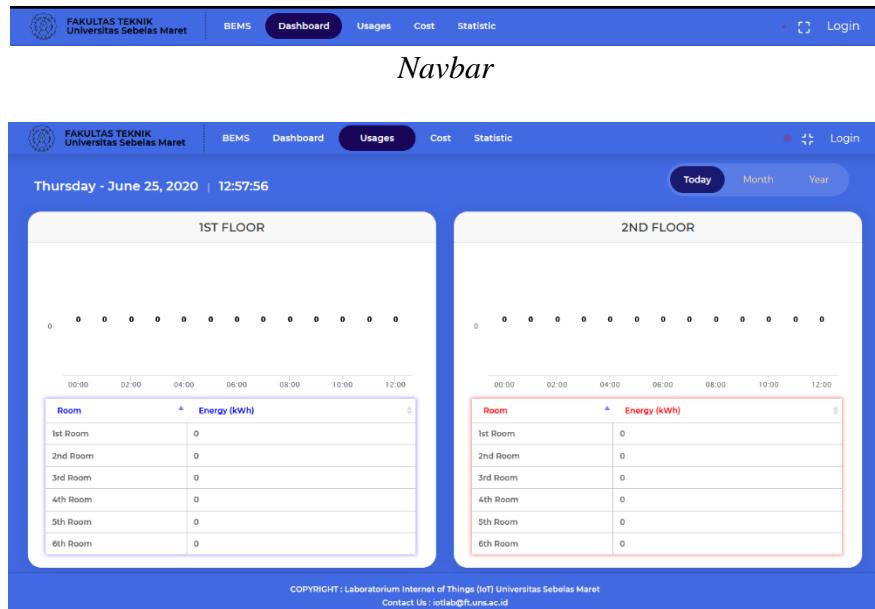
## B.2. Membuka website BEMS dan halaman – halamannya

1. Kunjungi *Building Energy Management System* di <https://si.ft.uns.ac.id/bems>
2. Maka akan tampil halaman *dashboard*

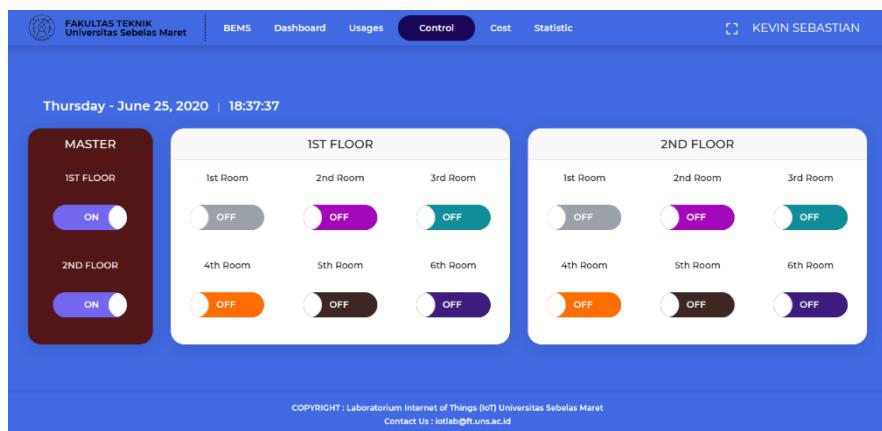


Halaman *Dashboard*

3. Terdapat halaman lainnya yang dapat diakses oleh *admin* dan *super admin*, yaitu halaman *Usages*, *Control*, *Cost*, *Statistic*, yang dapat diakses dengan menekan *link* yang ada di *navbar* (*navigation bar*).



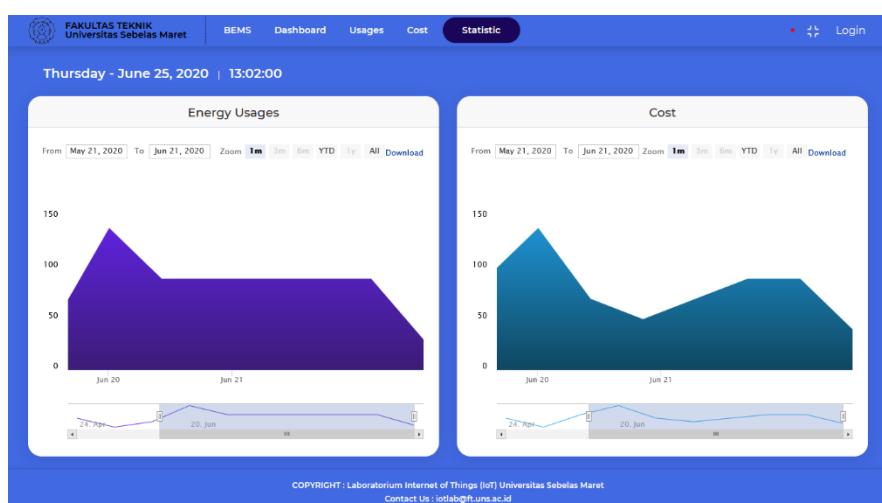
Halaman *Usages*



Halaman *Control*

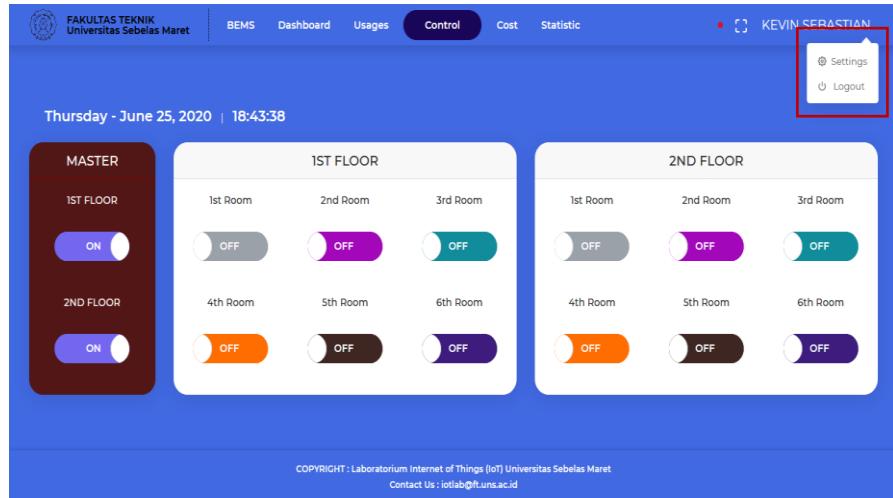


Halaman *Cost*

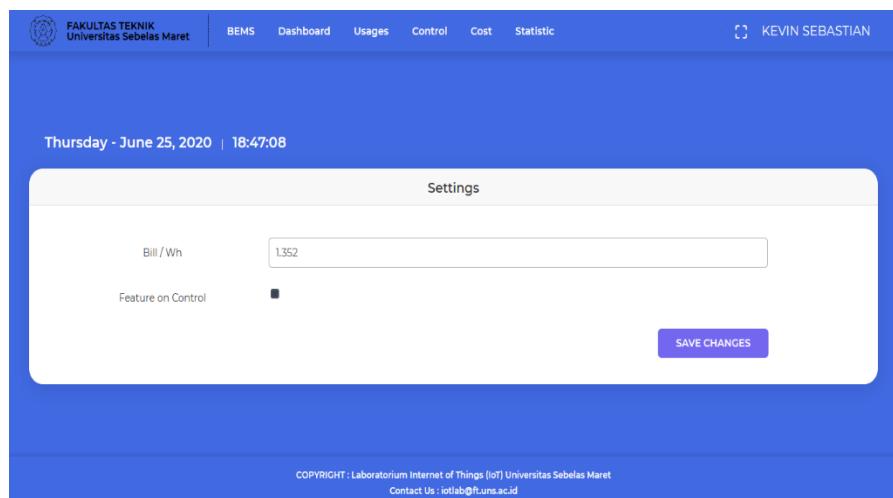


Halaman *Statistic*

4. Dan terdapat halaman *Settings* yang dapat diakses dengan cara menekan nama pada *navbar* dan akan muncul *sub menu* lalu menekan link *Settings*. Halaman ini dapat diakses oleh *admin* dan *super admin*.



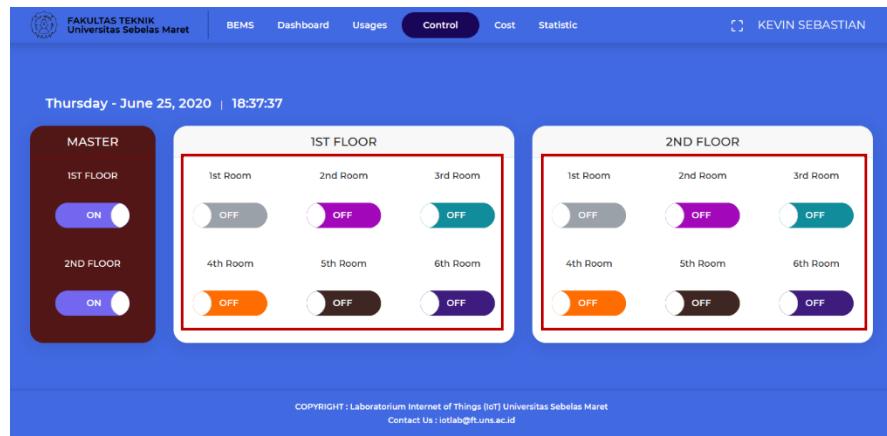
*Sub Menu* yang menampilkan *link Settings*



Halaman *Settings*

### B.3. Mematikan dan menghidupkan *Device* untuk tiap ruang

1. Masuk ke halaman *Control*
2. *Admin* dan *Super Admin* dapat menghidupkan *device* untuk masing – masing ruang dengan menekan tombol *on/ off*

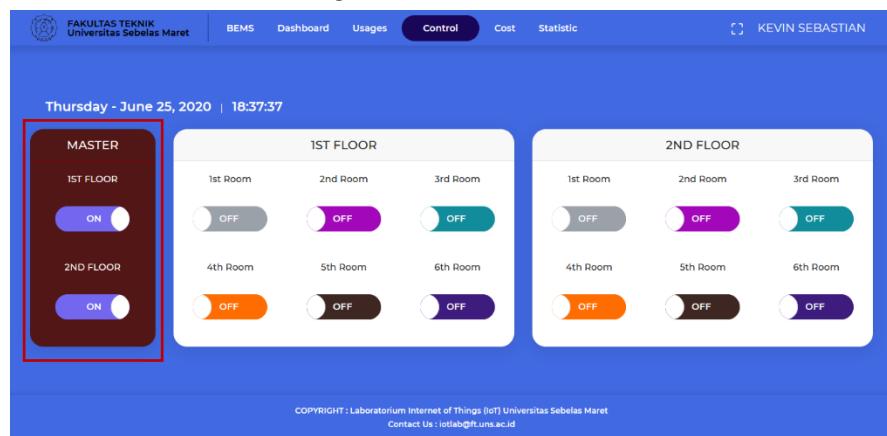


Tombol *on/ off* pada halaman *Control*

3. Terdapat tombol *on/ off* sebagai *control master*, yang digunakan untuk mengontrol device tiap ruang sekaligus (ketika *control master ON* maka semua tombol pada lantai tersebut akan *ON*, ketika *Control Master OFF* maka semua tombol pada lantai tersebut akan *OFF*)

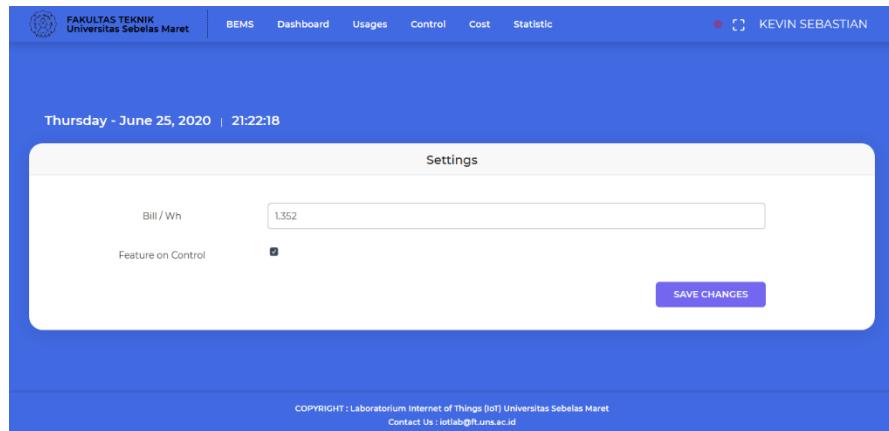
#### B.4. Mengaktifkan feature yang ada di Halaman *Control*

1. Masuk ke halaman *Settings*



*Control Master* yang terdapat di halaman *Control*

2. Menyentang *Feature on Control*
3. Menyimpan dengan menekan tombol *Save Changes*

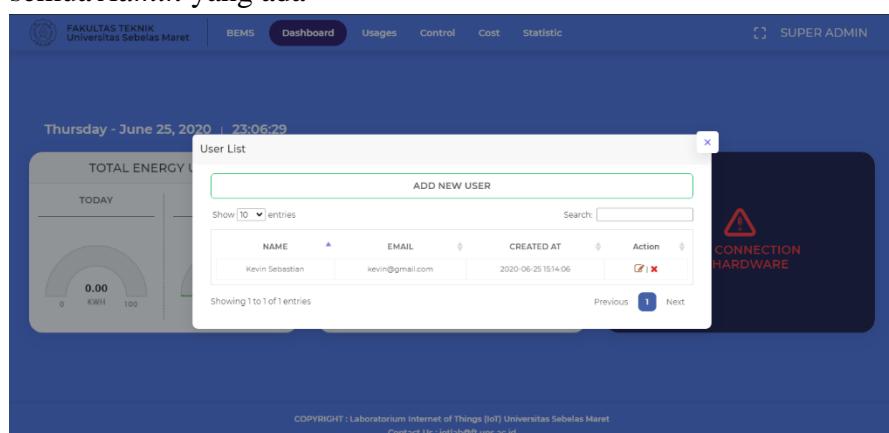


Menyentang *Feature on Control* pada Halaman *Settings*

4. Ketika *Feature on Control* diaktifkan, maka pada saat *Control Master* diaktifkan akan *merestore* value yang ada pada tiap Ruang sebelum *Control Master* tersebut dimatikan. Contoh: Sebelum *Control Master* dimatikan, hanya Ruang 1 dan Ruang 2 saja yang hidup, maka setelah *Control Master* diaktifkan kembali, Ruang 1 dan Ruang 2 saja yang akan hidup, sedangkan Ruang lainnya akan tetap mati.

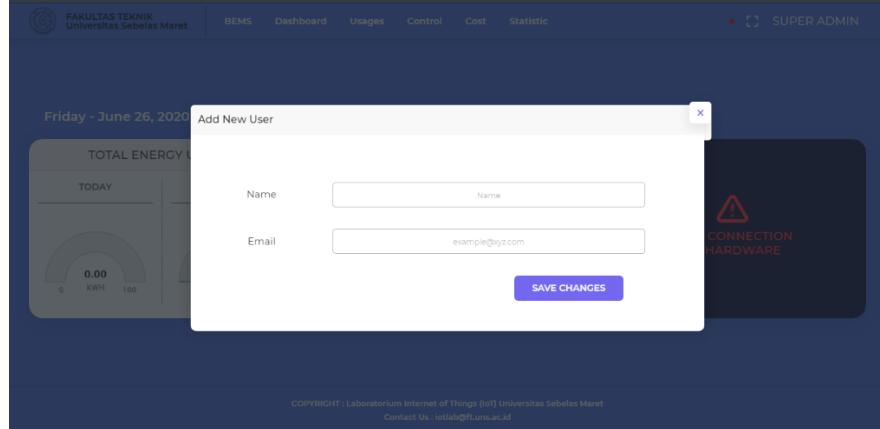
#### B.5. Menambahkan Admin (hanya dapat dilakukan oleh Super Admin)

1. Pastikan Anda Login dengan menggunakan akun **Super Admin**
2. Tekan nama **SUPER ADMIN** yang ada pada *navbar*
3. Akan muncul *sub menu* lalu tekan *link User List*
4. Maka muncul *pop-up window/ modal* yang mencakup informasi dari semua *Admin* yang ada



Modal yang berisi data *Admin* yang ada

5. Tekan tombol ADD NEW USER yang ada di bagian atas modal
6. Akan muncul *pop-up window/ modal* yang berisi *Name* dan *Email*

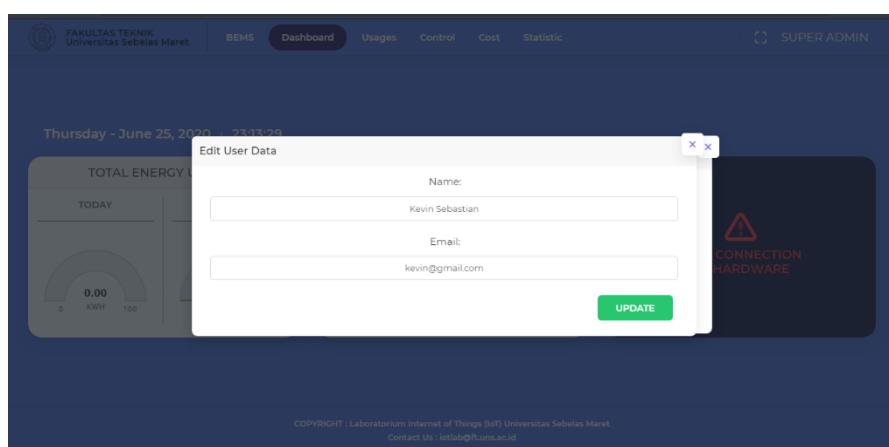


Tampilan Modal untuk menambah *Admin* baru

7. Inputkan Nama dan *Email* lalu tekan tombol *Save Changes* untuk membuat *Admin* yang telah anda inputkan (*Password* : **admin123**)

## B.6. Mengedit Informasi *Admin* (hanya dapat dilakukan oleh *Super Admin*)

1. Pastikan Anda *Login* dengan menggunakan akun *Super Admin*
2. Tekan nama *SUPER ADMIN* yang ada pada *navbar*
3. Akan muncul *sub menu* lalu tekan *link User List*
4. Maka muncul *pop-up window/ modal* yang mencakup informasi dari semua *Admin* yang ada
5. Tekan simbol yang ada pada tabel untuk mengedit informasi mengenai *Admin* tersebut
6. Akan muncul modal yang berisi *Nama* dan *Email*

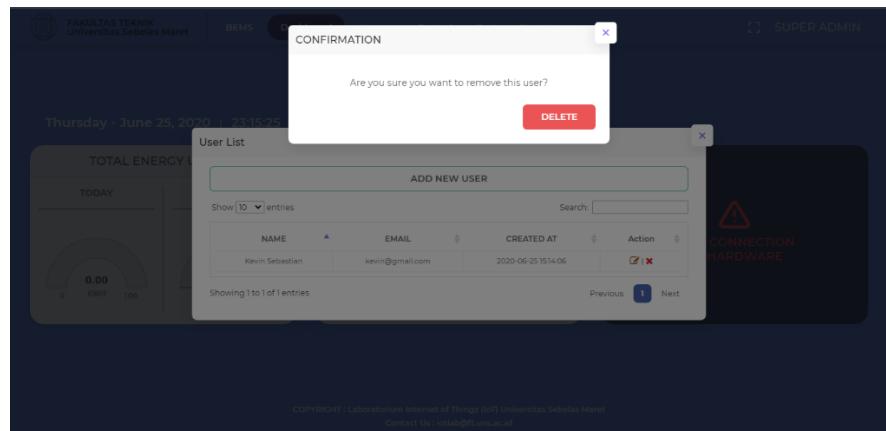


Tampilan Modal untuk mengedit data *Admin*

7. Ubah nama atau email sesuai dengan kebutuhan anda, dan tekan tombol Update

### B.7. Menghapus Admin (hanya dapat dilakukan oleh *Super Admin*)

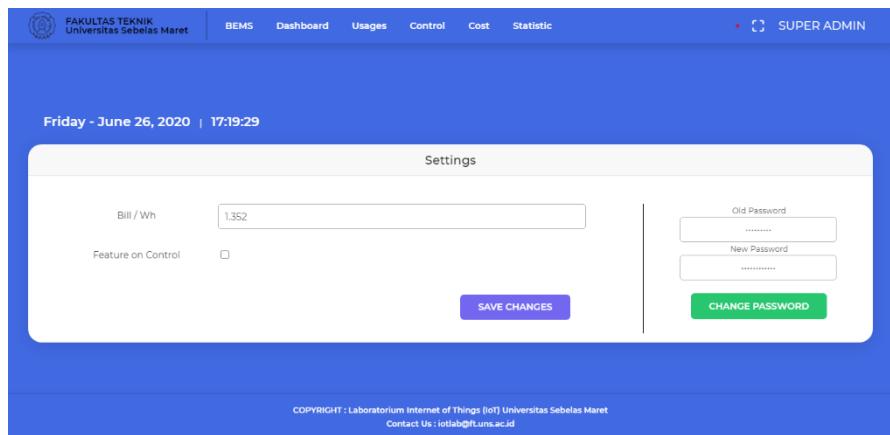
1. Pastikan Anda *Login* dengan menggunakan akun *Super Admin*
2. Tekan nama *SUPER ADMIN* yang ada pada *navbar*
3. Akan muncul *sub menu* lalu tekan *link User List*
4. Akan muncul *pop-up window/ modal* yang mencakup informasi dari semua *Admin* yang ada
5. Tekan simbol sesuai dengan *Admin* yang ingin Anda hapus
6. Akan muncul konfirmasi lalu tekan tombol **DELETE**



Tampilan konfirmasi untuk menghapus *Admin*

### B.8. Mengubah password

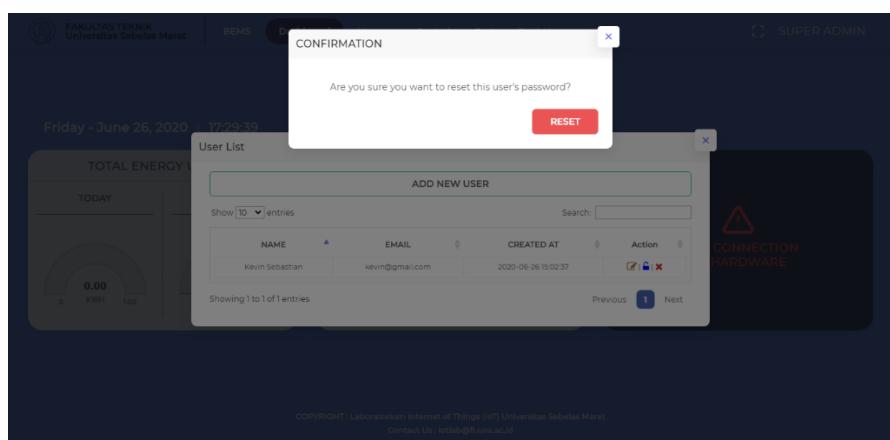
1. Tekan nama Anda yang ada pada *navbar*
2. Akan muncul *sub menu* lalu tekan *link Settings*
3. Setelah Anda masuk ke halaman *Settings*, inputkan password lama Anda dan password baru Anda di *form* yang sudah ada
4. Setelah menginputkan password, tekan **Change Password**



Tampilan pengubahan *password* di Halaman *Settings*

### B.9. Lupa Password (hanya dapat dilakukan oleh *Super Admin*)

1. Pastikan Anda *Login* dengan menggunakan akun *Super Admin*
2. Tekan nama *SUPER ADMIN* yang ada pada *navbar*
3. Akan muncul *sub menu* lalu tekan *link User List*
4. Akan muncul *pop-up window/modal* yang mencakup informasi dari semua *Admin* yang ada
5. Tekan simbol sesuai dengan *Admin* yang ingin Anda reset *passwordnya*
6. Akan muncul konfirmasi lalu tekan tombol **RESET**
7. *Password Admin* tersebut akan tereset ke *default password* yaitu **admin123**



Tampilan konfirmasi ketika akan mereset *Password*

## C. Tutorial untuk Developer

Tutorial ini diperuntukkan untuk *developer* yang akan mengembangkan lebih lanjut BEMS (*Building Energy Management System*). Tutorial berbentuk *case* atau kasus tertentu.

### Menambahkan menu dan grafik di BEMS

1. Buka file BEMS pada *text editor*
2. Buka file navbar.blade.php yang ada di /resources/views/panels/
3. Cari tag `<ul class="nav navbar-nav float-right" id=" sidenav">` dan tuliskan `<li class="nav-item d-none d-lg-block"><a class="nav-link" href="{{ url('menu') }}>Menu</a></li>`. Tulisan “Menu” yang ada di dalam tag `<a></a>` dapat disesuaikan judul menu yang diinginkan.
4. Setelah menulis *code* tersebut, maka akan muncul menu baru pada *navbar*.
5. Arahkan ke /app/Http/Controllers/ dan buat file baru dengan nama MenuController.php atau dengan nama lain yang diinginkan.
6. Buka MenuController.php tersebut dan tuliskan *code* ini  
`<?php`

**namespace App\Http\Controllers;**

```
class MenuController extends Controller
{
    public function index()
    {
        return view('pages.menu ');
    }
}
```

Terdapat class MenuController, MenuController dapat diganti dan disesuaikan dengan nama file yang telah dibuat.

7. Arahkan ke /resources/views/pages/ lalu buat file baru dengan nama menu.blade.php, lalu tuliskan *code* ini

```

@extends('layouts/contentLayoutMaster')

@section('title', 'Menu')


@section('vendor-style')
<link rel="stylesheet"
      href="//cdn.datatables.net/1.10.21/css/jquery.dataTables.min.css">
@endsection


@section('page-style')
<link rel="stylesheet"
      href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
<link rel="stylesheet" href="https://unpkg.com/bootstrap-datepicker@1.9.0/dist/css/bootstrap-datepicker3.min.css">
<link rel="stylesheet" href="{{ asset('css/pages/styles.css') }}">
@endsection


@section('content')
<section id="menuSection">
  <div id="guts">
    <div class="col-lg-12" id="time">
      <span id='date'></span>
      <span id="timeseparator">|</span>
      <span id='clock'></span>
    </div>
    <br />

    <div class="row">
      <div class="col-lg-12 col-12">
        <div class="card">
          <div class="card-header-chart">
            <h4>Energy Usages</h4>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

```

```

<div class="row-card">
    <div class="col-lg-12 col-12">
        <div id="energy_stat"></div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

</section>

{!! Charts::scripts() !!}
@endsection

@section('vendor-script')
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-
ui.min.js"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.
min.js"></script>
<script
src="https://code.highcharts.com/modules/export-
data.js"></script>
@endsection

@section('page-script')
<script src="{{ asset('js/checkHardware.js') }}"></script>
<script src="{{ asset('js/energy_stat.js') }}"></script>
<script src="{{ asset('js/sidenav.js') }}"></script>
<script src="{{ asset('js/date.js') }}"></script>

```

```

<script src="{{ asset('js/clock.js') }}"></script>
<script src="{{ asset('js/toggle.js') }}"></script>
@endsection

```

8. Buka file web.php di /routes/ dan tuliskan *code* ini

```
Route::get('menu','MenuController@index')->name('menu');
```

9. Untuk mengetahui lebih lanjut dari chart yang digunakan silakan ke <https://api.highcharts.com/highcharts/>, dokumentasi pada website highcharts atau cari pada forum highcharts.



```

<div id="navbar-menu" class="mr-auto float-left bookmark-wrapper d-flex align-items-center" style="padding-left: 10px;">
  <ul class="nav navbar-nav float-right" id="sidenav_bem">
    <li class="nav-item d-none d-lg-block" data-toggle="tooltip" title="Building Energy Management System">
      <a class="nav-link" data-toggle="modal" data-target=".bems-modal">BEMS</a>
    </li>
  </ul>

  <ul class="nav navbar-nav float-right" id="sidenav">
    <li class="nav-item d-none d-lg-block"><a class="nav-link" href="{{ url('dashboard') }}>Dashboard</a></li>
    <li class="nav-item d-none d-lg-block"><a class="nav-link" href="{{ url('usages') }}>Usages</a></li>

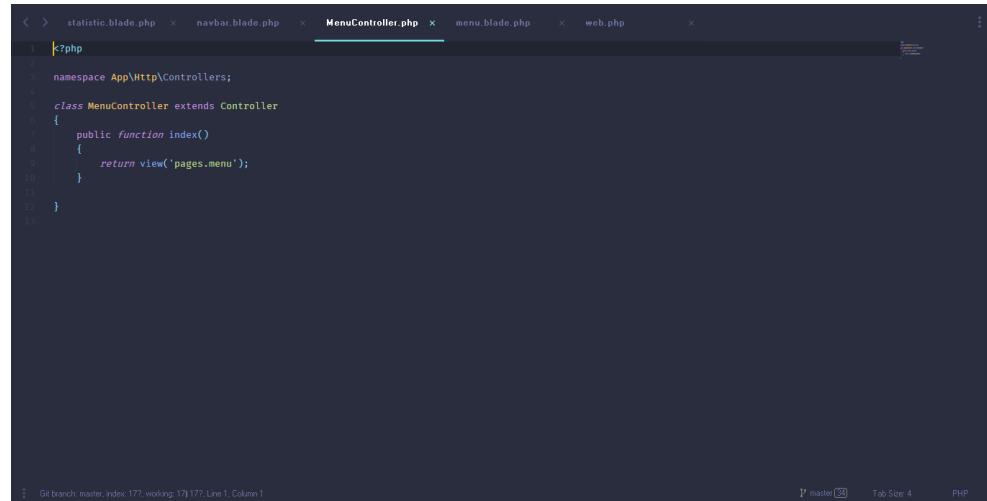
    @if(Auth::user())
      <li class="nav-item d-none d-lg-block"><a class="nav-link" href="{{ url('control') }}>Control</a></li>
    @endif

    <li class="nav-item d-none d-lg-block"><a class="nav-link" href="{{ url('cost') }}>Cost</a></li>
    <li class="nav-item d-none d-lg-block"><a class="nav-link" href="{{ url('statistic') }}>Statistic</a></li>
    <li class="nav-item d-none d-lg-block"><a class="nav-link" href="{{ url('custom') }}>Custom</a></li>
    <li class="nav-item d-none d-lg-block"><a class="nav-link" href="{{ url('menu') }}>Menu</a></li>
    <li class="nav-item d-none d-lg-block"><a class="nav-link" data-toggle="modal" data-target="#manual-modal">Manual</a></li>
  </ul>
</div>

<span class="spinner-grow spinner-grow-sm" role="status" aria-hidden="true" style="color: red; display: none" id="connection_problem" title="LOST CONNECTION TO HARDWARE"></span>
<ul class="nav navbar-nav float-right">
  <li class="nav-item d-none d-lg-block"><a class="nav-link nav-link-expand" data-toggle="tooltip" title="Fullscreen" id="full">
    <i class="ficon feather icon-maximize"></i></a></li>
  @if(Auth()>user())
    <li class="dropdown dropdown-user nav-item"><a class="dropdown-toggle nav-link dropdown-user-link" href="#" data-toggle=""

```

Menambahkan ke navbar.blade.php



```

App\Http\Controllers;
class MenuController extends Controller
{
    public function index()
    {
        return view('pages.menu');
    }
}

```

Tampilan pada MenuController.php

```

< > static.blade.php < > navbar.blade.php < > MenuController.php < > menu.blade.php < > web.php < >
( ) @extends('layouts/contentLayoutMaster')

@section('title', 'Menu')

@section('vendor-style')
<link rel="stylesheet" href="//cdn.datatables.net/1.10.21/css/jquery.dataTables.min.css">
@endsection

@section('page-style')
<link rel="stylesheet" href="https://unpkg.com/bootstrap-datepicker@1.9.0/dist/css/bootstrap-datepicker3.min.css">
<link rel="stylesheet" href="{{ asset('css/pages/styles.css') }}>
@endsection

@section('content')
<section id="menuSection">
<div id="guts">
    <div class="col-lg-12" id="time">
        <span id="date"></span>
        <span id="timeseparator">|</span>
        <span id="clock"></span>
    </div>
    <br />
    <div class="row">
        <div class="col-lg-12 col-12">
            <div class="card">
                <div class="card-header-chart">
                    <h4>Energy Usages</h4>
                </div>
            </div>
        </div>
    </div>
</div>
</section>

```

Git branch: master index 177 working 17/177 Line 1 Column 40

P master (3) Spaces: 2 Laravel Blade

Tampilan menu.blade.php

```

< > static.blade.php < > navbar.blade.php < > MenuController.php < > menu.blade.php < > web.php < >
Route::get('/mqqt/publish/custom2/{any}', 'CusController@SubscribeToTopic2')->where('any', '*');
Route::get('/mqqt/publish/custom3/{any}', 'CusController@SubscribeToTopic3')->where('any', '*');
Route::get('/mqqt/publish/custom4/{any}', 'CusController@SubscribeToTopic4')->where('any', '*');

Route::post('storeData', 'DashDataController@storeData');

Route::get('settings', 'SettingController@index')->middleware('auth')->name('settings');
Route::post('storeSettings', 'SettingController@storeSettings')->middleware('auth')->name('storeSettings');
Route::get('customChartSettings', 'CusController@CustomChartSettings')->name('customChartSettings');
Route::get('clear/{id}', 'CusController@Clear')->name('clear');

Route::get('adduser', 'SettingController@AddUser')->middleware('auth')->name('adduser');
Route::post('storeNewUser', 'SettingController@storeNewUser')->middleware('auth')->name('storeNewUser');

Route::get('edit/{id}', 'SettingController@Edit')->middleware('auth');
Route::post('update', 'SettingController@update')->middleware('auth')->name('update');
Route::get('delete/{id}', 'SettingController@Delete')->middleware('auth')->name('delete');
Route::get('reset/{id}', 'SettingController@Reset')->middleware('auth')->name('reset');
Route::post('changePass', 'SettingController@ChangePass')->middleware('auth')->name('changePass');

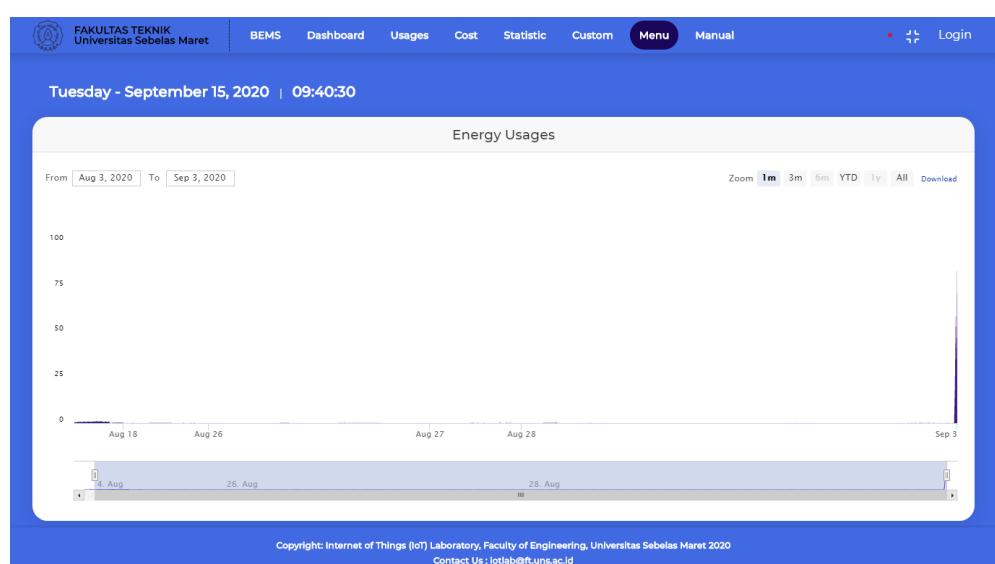
Route::get('menu', 'MenuController@index')->name('menu');


```

Git branch: master index 177 working 17/177 55 characters selected

P master (3) Tab Size: 4 PHP

Menambahkan pada web.php



Tampilan ketika semua langkah telah dilakukan