Terbit online pada laman web jurnal : **http://jurnal.iaii.or.id**

JURNAL RESTI

(Rekayasa Sistem dan Teknologi Informasi)

Vol. 1 No. 1 (2017) 50 – 57 | ISSN Media Elektronik : 2580-0760

Purwarupa Sistem Pendataan Pengunjung dan Pengendalian Perangkat Laboratorium Dalam Pengembangan Smart Campus

Heny Pratiwi[a], Alfa Satyaputra[b], Arnold Aribowo[c]

[a]Sistim Komputer, fakultas ilmu komputer, Universitas Pelita Harapan, 06hnmay@gmail.com

[b]Sistim Komputer, fakultas ilmu komputer, Universitas Pelita Harapan, alfa.satyaputra@uph.edu

[c]Sistim Komputer, fakultas ilmu komputer, Universitas Pelita Harapan, arnold.aribowo@uph.edu

Abstract

The laboratory visitors data collection system serves as a timestamp data collection of laboratory visitors, and the control system is set for controlling electrical devices such as air conditioners, lights, and door locks inside the laboratory room according to the visitors amount, both are integrated as one system for implementing the development of Smart Campus and supporting conveniences and security for Lecturers, staff and students. The system uses hardware such as door lock solenoid, relay board, RFID reader, serial-cables and Arduino microcontrollers. Some software are used such as Node.js scripts (JavaScript runtime environment) as the central logic and communication, one Arduino microcontroller programmed with Arduino software program, and another Arduino microcontroller is embedded with firmware Firmata and application software such as server XAMPP for web pages using Bootstrap framework and Javascript library such as AngularJS, JQuery and Javascript Bootstrap. Furthermore, MySQL is used as a database by using a serial protocol, and then WebSocket and Johnny-Five Framework are used as a complement system solution. Overall integration of the systems is expected to run smoothly as designed.

Keywords: Node.Js, MySQL, Websocket, Firmata, Serial Communication

Abstrak

Sistem pendataan pengunjung laboratorium berfungsi sebagai pendataan waktu keluar-masuk pengunjung laboratorium, sedangkan sistem pengendali perangkat mengendalikan perangkat kelistrikan seperti pendingin ruangan, lampu, dan kunci pintu, semua terintegrasi sebagai satu sistem sebagai implementasi dari pengembangan *Smart Campus* yaitu kampus yang memberikan kenyamanan dan keamanan bagi dosen, staff dan mahasiswa. *Hardware* yang digunakan adalah *solenoid lock door,* papan *relay,* pembaca *RFID*, kabel serial dan dua unit mikrokontroler Arduino, sedangkan *software* yang digunakan adalah skrip *Node.js* (*JavaScript runtime environment*) sebagai pusat logika dan komunikasi, program Arduino pada mikrokontroler Arduino pertama, dan *firmware Firmata* pada mikrokontroler Arduino kedua, dan aplikasi *XAMPP* sebagai *server* halaman *web* yang menggunakan *framework* Bootstrap, serta *library Javascript* diantaranya *AngularJS, JQuery* dan *Javascript Bootstrap*. *MySQL*, sebagai basis data, menggunakan *serial protocol*. Selanjutnya pelengkap solusi sistem adalah *websocket* dan *Johnny-Five Framework*. Hasil keluaran integrasi seluruh sistem sesuai rancangan adalah keberhasilan yang diharapkan.

Kata Kunci: Node.Js, MySQL, Websocket, Firmata, komunikasi serial

1. Pendahuluan

Automasi adalah satu sistem solusi untuk segala aktifitas operasi yang memiliki sifat repetitif yang dilakukan secara manual. Solusi dengan sistem automasi memberi kenyamanan, kemudahan dan kepraktisan buat para pemakai yang secara rutin harus melakukan aktifitas tersebut. Salah satu cara penerapan sistem automasi ini adalah dengan menggunakan *Microcontroller*.

Sistem pendataan pengunjung ruang laboratorium, antarmuka dan robotika, di gedung B ruang 241 Universitas Pelita Harapan, Lippo Karawaci, dipilih sebagai tempat untuk penerapan sistim rancangan automasi, dengan alasan operasional dan aktifitas diruang tersebut masih dilakukan secara manual. Pendataan pengunjung dijadikan sebagai masukan untuk pengendalian perangkat kelistrikan ruang laboratorium. Selanjutnya penerapan sistem automasi pendataan pengunjung, diharapkan bisa melayani

kenyamanan para pengunjung yang sering lupa mengisi buku daftar tamu setiap hari. Selama waktu operasional, sistem melakukan pendataan setiap pengunjung yang keluar-masuk ruang laboratorium tersebut. Selain bisa membantu kenyamanan pengunjung, sistem ini juga bisa mengurangi biaya konsumsi enersi listrik. Beberapa bagian sistem dan komponen membutuhkan rancang-bangun *hardware* dan *software*, dengan pengujian keduanya secara terpisah. Pengujian integrasi keseluruhan system merupakan tahap akhir.

Solenoid lock door, papan *relay,* pembaca *RFID*, kabel *serial*, dan dua *microcontroller* Arduino adalah *hardware* yang dipakai. Sedangkan *software* yang di rancang-bangun menggunakan bahasa pemrograman *Node.js* sebagai pusat logika dan komunikasi, bahasa pemrograman Arduino pada *microcontroller* Arduino pertama, dan bahasa pemrograman *firmware Firmata microcontroller* Arduino kedua, serta aplikasi *XAMPP* sebagai *server* pada *webpage*. Sedangkan *webpage* harus dibangun menggunakan *framework Bootstrap* dan beberapa *library Javascript* seperti *AngularJS, JQuery* dan *Javascript Bootstrap*, selanjutnya pendataan menggunakan *MySQL*. Dibutuhkan pula media protokol komunikasi antara *Node.js*, *firmware Firmata,* halaman *web* dan basis-data pendataan dengan menggunakan protokol *serial*, *websocket*, dan *Johnny-Five*.

2. Tinjauan Pustaka

2.1 MySQL

Basis data (*database)* adalah sekumpulan data yang diatur dengan satu system pengaturan, dan disimpan dalam/dengan format yang didefinsikan, kemudian strukturnya ditentukan oleh *metadata*. Ada beberapa model basis data yang dapat digunakan. Pada penelitian ini digunakan basis data *MySQL*, yang mengimplementasi model *relational.* Model ini merupakan model yang paling banyak digunakan karena memiliki beberapa kemudahan dibandingkan dengan model basis data lainnya, seperti: data dapat ditambah dan dikurangi tanpa merubah struktur data secara keseluruhan dan sebuah data dapat memiliki lebih dari satu *parent* [11] [15].

2.2 Node.js

Node.js merupakan *platform* yang dibuat dari *Chrome JavaScript runtime* yang berguna untuk membangun aplikasi jaringan dengan cepat dan *scalable*. *Node.js* menggunakan model *event-driven, non-blocking I/O* yang membuat *Node.js* ringan dan efisien, tepat untuk pembuatan aplikasi *real-time*. Hal ini menjadi salah satu kelebihan dari *Node.js* karena pada model bahasa pemrograman umumnya contohnya sekuensial, suatu proses hanya dapat dikerjakan setelah proses sebelumnya selesai. Namun pada *Node.js* proses tidak perlu menunggu proses sebelumnya selesai. [2][7]

2.3 Websocket dan Firmata

WebSocket pertama kali muncul pada *HTML5*, yang memungkinkan adanya komunikasi antara *server* dan *web browser* secara *real-time*. Untuk memulai sebuah koneksi *WebSocket*, pertama - tama *client* mengirimkan permintaan *handshake* kemudian *server* membalas dengan sebuah respon. Ketika sebuah koneksi telah tercipta, maka komunikasi awal yang berbasis *HTTP* berubah menjadi *komunikasi full-duplex* [2:13].

Firmata adalah sebuah protokol komunikasi antara *software* komputer dengan mikrokontroler. Protokol dapat diimplementasi di dalam *firmware* dari sebuah mikrokontroler atau pada paket perangkat lunak (*software package)*. Secara teori protokol Firmata dapat diimplementasi pada semua jenis mikrokontroler. Namun, pada saat ini implementasi terlengkap terdapat pada mikrokontroler Arduino, termasuk mikrokontroler yang kompatibel dengan arduino [5].

2.4 Komunikasi Serial

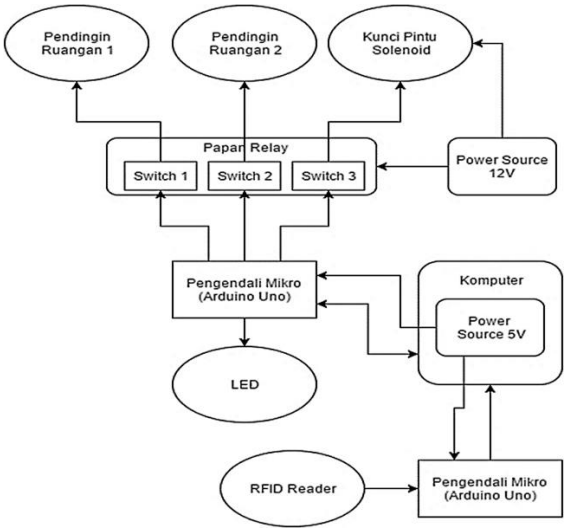
Komunikasi *serial* adalah proses pengiriman data satu *bit* secara berurutan dalam satu satuan waktu, melalui sebuah saluran komunikasi atau *computer bus.* Setiap data yang dikirimkan akan dimulai dengan sebuah *start-bit* untuk menandakan awal dari suatu data dan diakhiri dengan *stop-bit* untuk menandakan data berakhir. Setiap perangkat yang menggunakan komunikasi serial harus memiliki kesepakatan dimana kedua perangkat menggunakan pengaturan kecepatan dan paritas yang sama [14][15].

3. Metodologi Penelitian

Penelitian ini diawali dengan studi pustaka untuk mengumpulkan data dan informasi dari sumber informasi yang ada, baik melalui *e-book*, buku-buku, artikel, atau data-data lain dari internet yang berhubungan dengan permasalahan *solenoid lock door,* papan *relay,* pembaca *RFID*, kabel *serial* dan *microcontroller* Arduino, *Node.js, firmware Firmata*, aplikasi *XAMPP*, *framework Bootstrap*, *library Javascript* seperti *AngularJS, JQuery* dan *Javascript Bootstrap*, dan *MySQL*. Data dan informasi yang didapat kemudian dibaca dan dievaluasi sebelum diterapkan ke dalam sistem yang ingin dibangun. Selanjutnya, dilakukan rancang bangun *hardware* atau perangkat keras, dan dilakukan *troubleshooting* untuk setiap komponen, berikutnya juga dilakukan analisis, dan perancangan *software* atau perangkat lunak yang dibangun untuk mendapatkan keluaran yang tepat setelah pengujian.

4. Hasil dan Pembahasan

Sistem ini merupakan purwarupa atau *prototype,* sehingga perangkat laboratorium yang digunakan dalam penelitian ini bisa menjadi *framework* untuk direalisasikan kemudian hari, dan sudah menggunakan komponen yang mewakili produk nyata. Bagan purwarupa sistem yang dirancang dapat dilihat pada Gambar 1.



Gambar 1. Prototipe Sistem Yang Dirancang

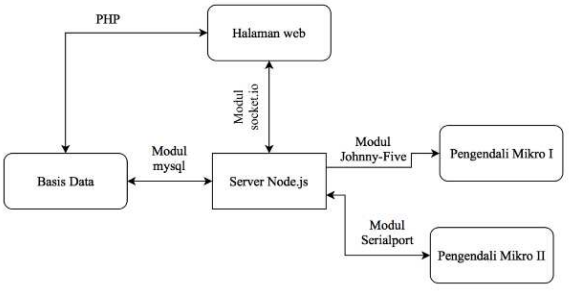
Pada perangkat pertama yaitu pendingin ruangan atau *air conditioner* menggunakan sebuah *relay*, yang pada aplikasi system dihubungkan ke pendingin ruangan tersebut. Karena ada tiga buah relay yang digunakan, maka sebuah papan yang memuat beberapa *relay* bisa membantu memudahkan proses aplikasi. *Relay* pertama dan kedua digunakan untuk simulasi mesin pendingin ruangan. *Relay* ketiga dihubungkan ke sebuah pengunci pintu *solenoid*. Sedangkan sumber tenaga *relay* tersebut didapatkan dari dua buah baterai jenis ukuran AA yang memiliki daya 12 *Volt*.

Perangkat kedua adalah dimaksudkan untuk mewakili lampu penerangan ruang, dalam hal ini digantikan oleh enam buah *LED* yang dirangkai membentuk tiga baris, setiap baris terdiri dari dua buah *LED* (dirancang seperti ruang laboratorium UPH gedung B ruang 241) dan dimaksudkan untuk mensimulasikan tiga baris lampu di ruang laboratorium B241. Setiap *pin* arus positif pada *LED* dihubungkan ke sebuah *pin digital* pada mikrokontroler Arduino untuk supaya *LED* tersebut mendapat sumber tenaga listrik dari Arduino tersebut dan sekaligus pengendalian *LED* juga bisa dilakukan melalui Arduino. Sedangkan semua *pin* arus negatif atau *ground* pada *LED* dihubungkan secara paralel pada *breadboard* dan kemudian dihubungkan ke *pin ground* pada Arduino.

Perangkat yang utama dalam sistem pendataan dan pengendalian seluruh *system*, adalah sebuah komputer. Selain komputer tersebut menjadi sumber tenaga listrik bagi mikrokontroler Arduino, komputer tersebut menjadi penting karena berfungsi sebagai *server*, yaitu *server Node.js.* yang sudah di unduh dan tersimpan dalam komputer tersebut. *Server* ini selanjutnya berfungsi sebagai pusat pengontrol, pengolahan data, dan mengelola komunikasi antara mikrokontroler Arduino, halaman *web* dan basis data.

Dalam penelitian ini melibatkan juga basis-data, halaman *web*, dan *server Apache*. Basis-data berfungsi untuk mengorganisasi data yang masuk. Halaman *web* diperlukan untuk memonitor aplikasi secara *mobile,* sedangkan *server Apache* yang sangat umum dipakai untuk halaman web dan basis data. Mikrokontroler yang digunakan untuk aplikasi sistim ini sebanyak dua buah Arduino Uno. Mikrokontroler Arduino Uno yang pertama berfungsi sebagai pengendali *relay* dan *LED*, sedangkan Arduino Uno yang kedua berfungsi sebagai penerima masukan data dari *RFID reader.* Pada aplikasi sistim ini juga perlu digunakan *firmware Firmata* yang berfungsi untuk menjembatani kondisi *pin* Arduino Uno supaya dapat dikendalikan oleh *server Node.js*. Namun *firmware* tersebut tidak mendukung *library MFRC522*, yang dihubungkan ke Arduino dimaksudkan sebagai media komunikasi dengan *modul RFID reader*, untuk mengatasi hal ini maka diperlukan Arduino Uno kedua yang menyimpan *library MFRC522* sehingga bisa menjembatani komunikasi dengan *modul RFID Reader*.

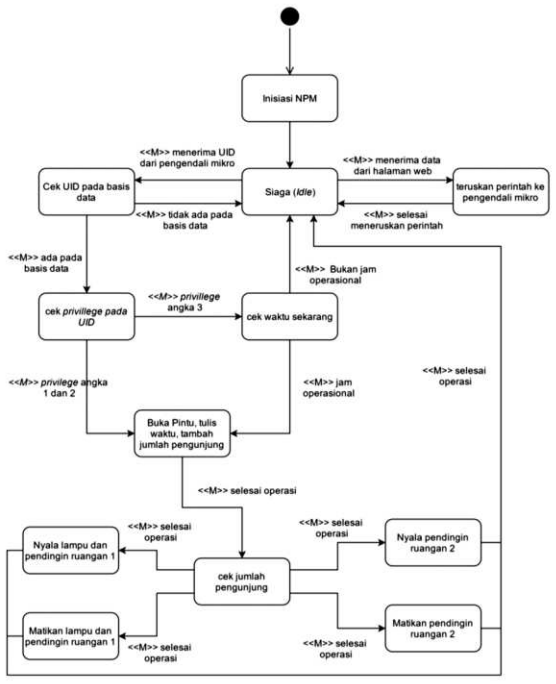
Ada dua *software* yang dikembangkan,yaitu *server Node.js,*berupa skrip dan aplikasi Arduino. Skrip *Node.js* dijalankan pada terminal / *Command Prompt* sistem operasi. Hampir semua sistem operasi yang ada sekarang ini mendukung *platform Node.js*, diantaranya adalah sistem operasi *Linux, Unix, Mac OS* dan *Windows*. Secara umum *server Node.js* memiliki dua tugas utama, yaitu sebagai pusat komunikasi sistem dan sebagai pengolah data. Gambar 2 adalah diagram sistem komunikasi yang dipakai dalam penelitian ini:



Gambar 2. Diagram Sistem Komunikasi

Adapun komunikasi yang dikelola oleh *server Node.js* antara lain: Pertama, komunikasi antara *server* dan halaman *web* dengan menggunakan modul *Node.js* bernama *Socket.io. Socket.io* merupakan modul *Node.js,* yang menggunakan protokol komunikasi

WebSocket. Kedua, komunikasi antara *server* dan basis-data dengan menggunakan modul *mySQL*. Modul ini merupakan implementasi *MySQL* pada *Node.js*, sehingga *syntaxes* yang digunakan pada modul ini sama dengan *syntaxes* pada *MySQL*. Ketiga, sedangkan untuk komunikasi antara *server* dan mikrokontroler Arduino yang pertama, modul *Johnny-Five* digunakan sebagai media untuk menjembatani komunikasi tersebut. Modul ini merupakan *framework* berbasis *Firmata*, yang berfungsi untuk mempermudah komunikasi antara *Node.js* dan mikrokontroler Arduino. *Firmware Firmata* harus diunduh dan disimpan dalam memory Arduino, untuk penggunaan modul tersebut adalah dengan melalui akses yang sudah tersedia pada *Arduino IDE*. Keempat, modul *serialport* digunakan untuk komunikasi antara *server* dan mikrokontroler Arduino yang kedua. Modul ini digunakan, karena perbedaan jenis komunikasi yang ada pada mikrokontroler Arduino pertama, yang sudah menyimpan *firmware Firmata*, dan tidak mendukung *library MFRC522*. Sehingga modul *serialport* ditambahkan untuk mengatasi hal tersebut. Modul *serialport* menerima data dari *port serial*, sehingga data yang dikirimkan oleh mikrokontroler Arduino yang kedua, melalui *serial port,* diterima oleh *server Node.js*.



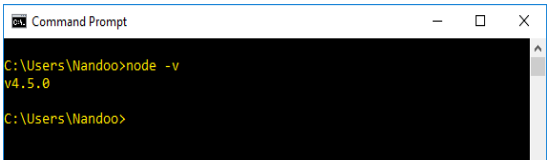
Gambar 3. Diagram cara kerja skrip server Node.js

Diagram kerja skrip *server Node.js* bisa dilihat pada Gambar 3. Ketika skrip dijalankan, maka skrip melakukan inisiasi modul, yaitu *modul NPM*. Kemudian skrip memasuki fase *event-loop*, *event* ditandai dengan munculnya bentuk bangun dua-dimensi yang berbingkai lebih tebal, lalu skrip melakukan instruksi sesuai dengan *event* yang terjadi. Sebagai contoh, bila skrip menerima instruksi dari halaman *web* (*website)*, maka skrip melakukan proses untuk meneruskan instruksi tersebut ke Arduino secara terus menerus (seperti digambarkan oleh diagram yang tidak terdapat *end*).

Tahap pemasangan *Node.js*, pertama unduh *installer Node.js* pada halaman *web* resmi *Node.js* yaitu https://nodejs.org/en/download/, lalu unduh *installer* sesuai dengan pada sistem operasi yang ada di komputer. Pada penelitian ini digunakan sistem operasi *Windows 64-bit*, maka, *installer* yang diunduh adalah *installer Windows 64-bit.* Untuk lebih jelas dapat dilihat pada Gambar 4.

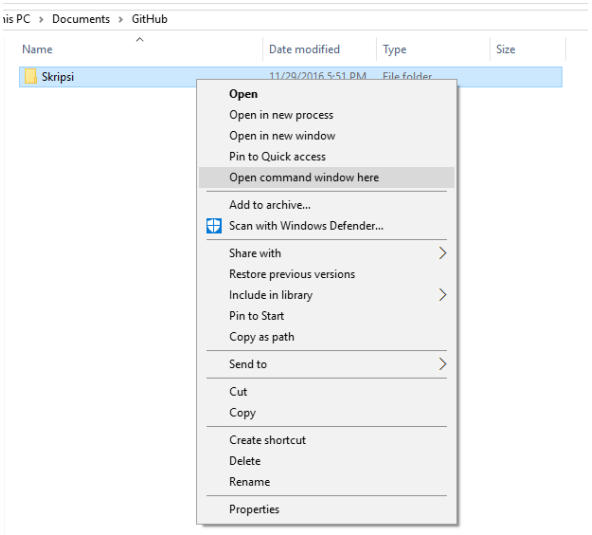
Gambar 4. Pemasangan Node.js

Setelah *installer* berhasil diunduh, kemudian dilakukan proses instalalisasi hingga selesai. Lalu dilakukan pengujian apakah *Node.js* telah terpasang dengan baik, dengan mengetikan *"node –v"* pada aplikasi *Command Prompt*. Jika *Node.js* telah terpasang dengan benar maka *Command Prompt* mengembalikan keluaran yang berisi versi dari *Node.js* yang terpasang. Hasil keluaran terlihat pada Gambar 5.



Gambar 5. Pengujian Node.js

Environment dari *Node.js* dikelompokkan dalam bentuk *project* yang ditempatkan pada suatu berkas/*folder*. Sehingga setelah sebuah nama *folder* dibuat, kemudian buka aplikasi *Command Prompt* dan tujukan *Command Prompt* pada *folder* ini dengan cara menahan tombol *shift* pada keyboard dan klik kanan pada *folder project*, klik pilihan *"open command window here"* pada menu *drop-down* yang muncul. Untuk lebih jelas dapat dilihat pada Gambar 6.



Gambar 6. Menu Dropdown

Setelah itu, terbuka aplikasi *Command Prompt* yang tertuju pada berkas proyek, lalu ketikan perintah berikut pada *Command Prompt "npm install johnny-five mysql serialport socket.io"*. Maka *NPM (Node Package Module)* memasang empat modul yang tertera pada perintah. Pastikan komputer terhubung ke internet, karena *NPM* memerlukan koneksi internet untuk melakukan pemasangan modul. Setelah selesai, *Command Prompt* memberitahukan pada keluaran yang menyatakan pemasangan berhasil.

Berikut cara memasang *firmware Firmata,* pada mikrokontroler Arduino yang pertama, sambungkan mikrokontroler ke komputer, lalu buka aplikasi *Arduino IDE*. Jika aplikasi belum terpasang, maka *installer* aplikasi dapat diperoleh di halaman resmi Arduino di *https://www.arduino.cc/en/Main/Software*. Lalu pilih menu *file* > *example* > *Firmata* > *StandardFirmata.* Maka terbuka *source code StandardFirmata*. Kemudian *upload source code* tersebut ke Arduino dengan menekan tombol *upload*. Pemasangan *Firmware Firmata,* pada mikrokontroler Arduino yang pertama, selesai.

Sambungkan mikrokontroler Arduino yang kedua ke komputer, melalui kabel *usb-to-serial*. Pilih menu, *file* > *open* >, lalu tujukan ke berkas yang telah di unduh. Buka berkas *"Arduino"*, di dalamnya terdapat *source code* bernama *"rfid.ino"*. Klik dua kali *file* tersebut. Kemudian unggah *source code* tersebut dengan menekan tombol *upload.* Pemasangan *source code* Arduino, pada mikfrokontroler Arduino yang kedua, selesai. Aplikasi Arduino, yang dipasang pada mikrokontroler Arduino yang kedua, berfungsi untuk membaca *numeric* didalam kartu *RFID* (*smart card*), yang dimiliki oleh pengunjung lab, melalui *RFID Reader*.

Basis data, secara keseluruhan, terdiri dari sebuah basis data bernama "siskomlab" dengan empat buah table, yaitu tabel buku tamu, tabel pengunjung terdaftar, tabel *login* halaman *web* dan tabel jumlah pengunjung.

Tabel Buku Tamu, berfungsi seperti layaknya buku tamu yang biasa digunakan, berisikan nama pengunjung, dan rekam waktu masuk/keluar ruang lab. Contoh tabel dapat dilihat pada Tabel 1, Tabel 2, dan Tabel 3.

Tabel 1. Tabel Buku Tamu -1

UID	Jam masuk	Jam keluar
2322723052	2016-10-12 16:50:15	2016-10-12 16:50:19
2525118514	2016-10-12 16:50:26	2016-10-12 16:50:33

Tabel 2. Tabel Buku Tamu -2

Secara keseluruhan:

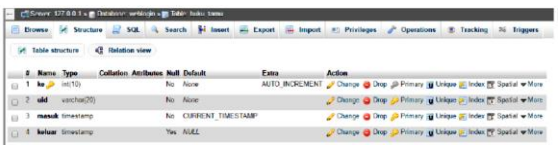
Tabel 3. Tabel Buku Tamu -3

Tabel pengunjung terdaftar berisikan pengunjung (mahasiswa), penanggung jawab ruang lab, dan dosen yang telah terdaftar atau dengan kata lain orang yang memiliki izin untuk masuk dan keluar ruang lab. Hanya orang yang tercatat pada tabel ini yang dapat masuk/keluar ruang lab. Tabel terdiri dari empat kolom yaitu kolom nama, *UID*, *privillege,* dan status. Kolom nama berisi nama pengunjung, kolom *UID* berisi kode identitas dari pengunjung tersebut, kolom *privilege* atau dalam bahasa Indonesia berarti hak istimewa, berisikan angka antara satu sampai tiga. Kolom terakhir adalah kolom status yang berupa angka antara nol atau satu. Angka ini menandakan status posisi pengunjung, angka satu berarti pengunjung sedang berada di dalam lab, angka nol berarti pengunjung sedang berada di luar lab. Format tabel dapat dilihat pada contoh Tabel 4 dan Tabel 5.

Tabel 4. Tabel Pengujung Terdaftar -1

Nama	UID	Privilege	Status
Andre Young	10628203226	1	0
Pak Arnold	2322723052	3	0

Tabel 5. Tabel Pengujung Terdaftar -1



Tabel *login* halaman *web,* berisi para pengguna halaman web, yang memiliki izin untuk melakukan *login* ke halaman web. Tabel terdiri dari dua buah kolom, kolom pertama pada basis data bernama kolom *user* berisi nama pengguna halaman web, dan kolom kedua pada basis data bernama *pass* berisi kata sandi atau *password* dari pengguna yang bersangkutan. Untuk lebih jelas dapat dilihat pada Tabel 6 dan Tabel 7.

Tabel 6. Tabel Login -1

user	pass
admin	1122334455667788
nandoo	bulan

Tabel 7. Tabel Login -2

Sebagai contoh, pada baris pertama, nama pengguna bernama "admin" dan memiliki kata sandi "1122334455667788".

Tabel jumlah pengunjung berisikan jumlah pengunjung yang sedang berada di dalam lab. Tabel hanya terdiri dari satu kolom yaitu kolom "jumlah". Pada basis data tabel dinamakan "jumlah_pengunjung" yang dapat dilihat pada Gambar 7.

Gambar 7. Jumlah_Pengunjung

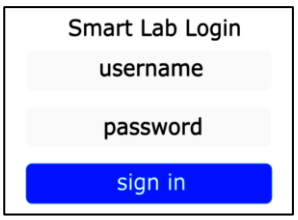
Sistem ini juga di rancang untuk melayani *privilege*, pelayanan sebuah privilege harus dilengkapi dengan *rules* atau aturan. Berikut adalah penjelasan arti angka pada kolom *privilege* dalam tabel pengunjung yang terdaftar, karena angka ini mempengaruhi aturan yang berlaku terhadap setiap pengunjung. Pada kolom *privilege* berisi angka satu sampai tiga. Angka tersebut menandakan nama pengunjung dan hak diberikan kepada pengunjung tersebut, arti angka-angka tersebut adalah angka satu berarti mahasiswa, angka dua berarti mahasiswa yang mendapat hak istimewa sementara, angka tiga berarti dosen atau asisten dosen. Dengan kata lain, "pengunjung angka satu" berarti yang dimaksudkan adalah pengunjung yang memiliki *privilage* tingkat satu, jika disebutkan "pengunjung angka dua" berarti yang dimaksudkan adalah pengunjung yang memiliki *privilege* tingkat dua, begitu juga untuk "pengunjung angka tiga" yang berarti pengunjung yang memiliki *privilege* tingkat tiga.

Implementasi *privilege* pada system dengan aturan sebagai berikut: aturan pertama, hanya pengunjung yang terdaftar pada "Tabel Pengunjung Terdaftar" yang mendapatkan izin dari sistem untuk masuk ruang lab. Aturan kedua, penentuan jam operasional lab pada hari Senin-Jumat pukul 07:00-16:00, di luar jam operasional hanya pengunjung terdaftar dengan *privilege* angka tiga dan dua yang bisa memasuki ruang lab. Ketentuan aturan ke-tiga adalah ketika pengunjung yang terdaftar dengan *privilege* kode angka tiga dan dua memasuki ruang lab, lampu dan pendingin ruangan menyala secara otomatis jika kondisi awal dalam keadaan tidak menyala. Ketentuan aturan ke-empat adalah ketika ruang lab kosong, sistem dengan otomatis mematikan lampu dan pendingin ruangan. Ketentuan aturan ke-lima adalah ketika pengunjung lab berjumlah kurang dari sepuluh orang, hanya satu pendingin ruangan yang dinyalakan oleh sistem. Jika pengunjung lab berjumlah lebih dari sepuluh, sistem menyalakan dua buah pendingin ruangan.

Fungsi utama dari halaman *web* adalah melakukan *interfacing* / antarmuka pengolahan data, menampilkan data, dari basis data, dan antarmuka untuk operasi pengendalian perangkat kelistrikan secara langsung. Halaman *web* difungsikan secara sepsifik kepada penanggung jawab lab (asisten dosen) atau dosen yang bersangkutan. Halaman *web* dialokasikan pada *Apache server* yang beroperasi di komputer yang sama dimana mikrokontroler Arduino terhubung. *Apache server* yang digunakan, adalah dari *installer XAMPP*, yang merupakan aplikasi pemasangan yang di dalamnya terdapat *Apache server,* basis data *MySql*, bahasa skrip *PHP* dan *Perl* (dalam penelitian ini, bahasa skrip *Perl* tidak digunakan).

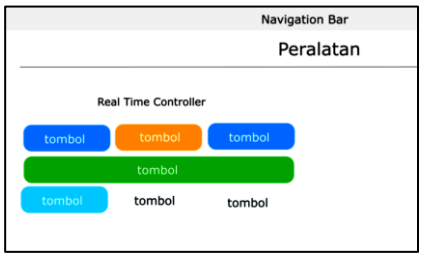
Tampilan halaman *web* dibuat dengan menggunakan *framework Bootstrap*. Semua tampilan, dari *menu drop-down*, judul, teks, dan tombol, menggunakan *style default* milik *Bootstrap*.

Untuk memungkinkan halaman *web* dapat mengolah data dari basis data digunakan skrip *PHP*. Selain digunakan untuk mengolah data, *PHP* juga digunakan untuk menjaga keamaan halaman *web*. Salah satu fitur *PHP* yaitu sesssion yang berfungsi untuk mencatat sesi *login* pengguna halaman *web*, sehingga halaman *web* tidak dapat diakses oleh pengguna yang tidak melakukan *login*. Halaman *login* dapat dilihat pada Gambar 8.



Gambar 8. Prototipe Sistem Yang Dirancang

Halaman *login*, dimana penanggung jawab lab atau dosen yang bersangkutan melakukan *login*. Halaman "Depan", berisi fungsi dan fitur dari halaman *web*. Pada bagian atas terdapat *navigation bar* yang merupakan *header* navigasi dari halaman *web*, karena fungsinya sebagai navigasi sehingga posisi *navigation bar* bersifat *persistant* yang berarti *navigation bar* tetap ada pada setiap halaman *web* kecuali halaman *login*, berisi tombol - tombol *link* yang mengarah ke halaman – halaman lain, selain itu juga terdapat tombol untuk melakukan *logout*. Kemudian di bagian bawah *navigation bar* terdapat gambar, isi gambar adalah gambar yang bertema *smart laboratorium*. Gambar ini bertujuan untuk memperindah halaman "Depan". Kemudian, dilanjutkan dengan ringkasan fungsi dan fitur halaman *web*, yang dilengkapi dengan beberapa *screenshot* halaman *web*. Untuk tampilan halaman peralatan dapat dilihat pada Gambar 9.

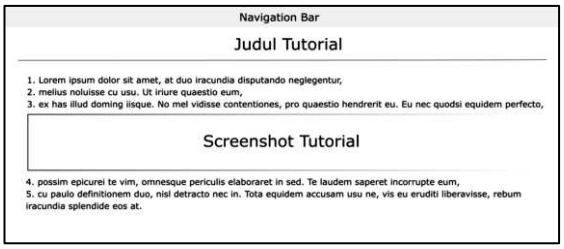


Gambar 9. Tampilan Halaman Peralatan

Halaman "Peralatan" atau *tools* berisikan judul halaman, lalu pada bagian tengah berisi tombol pengendali perangkat kelistrikan, sehingga pengguna dapat menyalakan atau mematikan perangkat secara langsung. Terdapat juga tombol yang mengarahkan ke *phpMyAdmin*, sebuah *tools* yang digunakan untuk pengolahan basis data.

Halaman "Data Pengunjung" berisi judul halaman, di bagian tengah terdapat dua buah tabel yang bersampingan, dan di bagian bawah terdapat tombol untuk memanipulasi data pada tabel pengunjung. Tabel pertama berisi data pengunjung berupa waktu rinci masuk dan keluar lab, tabel kedua berisi pengunjung yang terdaftar.

Halaman "Bagaimana Cara" pada bagian atas terdapat judul halaman, kemudian di bawah judul terdapat penjelasan menggunakan halaman web. Rancangan halaman "Bagaimana Cara" dapat dilihat pada Gambar 10.



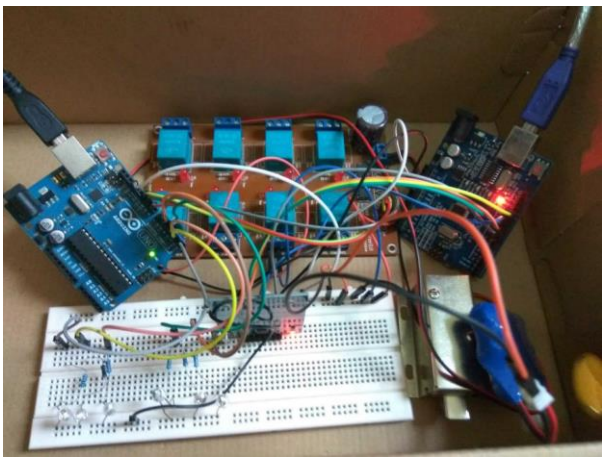
Gambar 10. Tampilan Halaman Bagaimana Cara

Halaman "Tambah Pengunjung" yang berguna untuk menambahkan pengunjung atau *member* baru ke basis data. Penambahan pengunjung dilakukan dengan memasukan nama pengunjung pada *textbox* yang disediakan dan pada *textbox* "*uid*" akan secara otomatis terisi ketika penanggung jawab lab mendekatkan atau *scan* kartu yang disediakan untuk pengunjung atau *member* baru yang akan didaftarkan. Rancangan halaman "Tambah Pengunjung" dapat dilihat seperti Gambar 11.

Gambar 11. Tampilan Halaman Tambah Pengujung

Dalam hal sistem keamanan halaman web, untuk penelitian ini, salah satu fitur dari *php* yang digunakan adalah *session*. Sehingga halaman *web* hanya bisa diakses ketika pengguna halaman *web* telah melakukan *login*. Jika pengguna halaman *web* tidak melakukan *login,* dan mencoba untuk mengakses halaman tertentu, maka pengguna halaman web diarahkan ke halaman *login*. Fitur *session* diterapkan pada semua halaman web, kecuali halaman *login*.

Gambar keseluruhan sistem *hardware* yang sudah terintegrasi dapat dilihat pada Gambar 12.



Gambar 12. Keseluruhan Sistem Hardware

Dalam proses implementasi *software* perlu diingat pada saat *installer Node.js* melakukan pemasangan, tidak hanya *platform Node.js* yang terpasang, tetapi *installer* juga memasang aplikasi *NPM (Node Package Module)* yang berfungsi untuk pemasangan modul *Node.js*.

Pada sistem yang dibangun memiliki tiga macam pengunjung yang memiliki *privilege* tertentu, maka setiap jenis *privilege* pengunjung diuji dengan empat jenis pengujian sehingga total pengujian terdapat 12

jenis pengujian. Setiap jenis pengujian dilakukan sebanyak 30 kali. Hasil dari pengujian ditampilkan pada tabel berikut dibawah ini. Dari hasil keseluruhan didapatkan nilai rata-rata keberhasilan sebesar 94% dan kegagalan sebesar 6%. Kegagalan yang terjadi diakibatkan oleh *bug* yang terdapat pada modul *Node.js* serialport. Modul terkadang tidak merespon atau memberikan respon terlambat. Tabel hasil pengujian dapat dilihat pada Tabel 9.

Tabel 9. Tabel Hasil Pengujian

Kombinasi Pengujian yang dilakukan				Hasil yang diharapkan			Hasil yang didapat			Persentase hasil
Angka Pengunjung	Posisi	Jam Operasional		Kunci Pintu	Lampu	AC	Kunci Pintu	Lampu	AC	
3	Masuk	Ya		Terbuka	Nyala	Nyala	Terbuka	Nyala	Nyala	90%
	Keluar			Terbuka	Mati	Mati	Terbuka	Mati	Mati	90%
	Masuk	Tidak		Terbuka	Nyala	Nyala	Terbuka	Nyala	Nyala	90%
	Keluar			Terbuka	Mati	Mati	Terbuka	Mati	Mati	100%
2	Masuk	Ya		Terbuka	Nyala	Nyala	Terbuka	Nyala	Nyala	98%
	Keluar			Terbuka	Mati	Mati	Terbuka	Mati	Mati	98%
	Masuk	Tidak		Terbuka	Nyala	Nyala	Terbuka	Nyala	Nyala	96%
	Keluar			Terbuka	Mati	Mati	Terbuka	Mati	Mati	96%
1	Masuk	Ya		Terbuka	Tidak berubah		Terbuka	Tidak berubah		90%
	Keluar			Terbuka			Terbuka			90%
	Masuk	Tidak		Terkunci			Terkunci			94%
	Keluar			Terkunci			Terkunci			96%

5. Kesimpulan

5.1 Simpulan

Telah dibuat purwarupa sistem pendataan pengunjung dengan mencatat waktu masuk/keluar pengunjung dan pengendalian perangkat kelistrikan laboratorium yaitu pendingin ruangan, lampu, dan kunci pintu dengan mengimplementasi aturan yang didefinisikan. Sistem Pendataan Pengunjung telah berhasil mencatat data pengunjung meliputi nama pengunjung, waktu masuk/keluar pengunjung ke laboratorium. Sistem Pengendalian Peralatan Kelistrikan Laboratorium telah mampu mengendalikan lampu 12V, *relay* dan kunci pintu *solenoid*. Hasil pengujian sistem secara keseluruhan diperoleh persentase keberhasilan sebesar 94%. Hasil ini didapat dengan merata -rata nilai hasil pengujian secara keseluruhan. Terdapat dua belas jenis pengujian yang masing-masing dilakukan sebanyak 30 kali.

5.2 Saran

Dari kesimpulan di atas, ada beberapa saran yang dapat diterapkan untuk mengembangkan penelitian ini. Diantaranya, yaitu memperbanyak variasi perangkat kelistrikan yang dapat dikendalikan, seperti kipas angin dan *projector*. Kemudian mengintegrasikan dengan sistem lain, terutama sistem yang menggunakan *Node.js* dan berhubungan dengan *Internet of Things*, seperti sistem pengunci loker elektronik.

6. Daftar Rujukan

[1] Tanenbaum, A., 2014. *Computer networks.* 6th ed. Upper Saddle River, NJ: Prentice Hall PTR.
[2] Rai, R., 2013. *Socket.io Real-time Web Application Development.* 1st ed. Birmingham: Packt Pub.
[3] Internet of things agenda. Rouse, M. *RFID (radio frequency identification).* Updated April 2007. Available at: http://internetofthingsagenda.techtarget.com/definition/RFID-radio frequency-identification. [Accessed: Nov. 14, 2016].
[4] Northwestern University Mechatronics Design Laboratory, 2006. Available at: http://mechatronics.mech.northwestern.edu. [Accessed: Nov. 14, 2016]
[5] Github. *Firmata Protocol Documentation.* Available at: https://github.com/firmata/protocol. [Accessed: Nov. 14, 2016]
[6] Chaniotis, I.K., Kyriakou, K.I.D., Tselikas, N.D., 2015. *Is Node.js a viable option for building modern web applications? A performance evaluation study.* Springer
[7] M Cantelon, M Harter, TJ Holowaychuk, N Rajlich. 2014. *Node.js in Action.* Available at: toc.dreamtechpress.com
[8] Daggett, M.E., 2013. *JavaScript IRL.* Springer
[9] Node.js Foundation, 2016. *Working with Different Filesystems*. Available at: https://nodejs.org/en/docs/guides/working-with-different-filesystems [Accessed: Nov. 14, 2016]
[10] Standard ECMA-414. 2016. *ECMAScript Specification Suite* .2nd ed. Available at: https://www.ecma-international.org/publications/standards/Ecma-414.htm [Accessed: Nov. 16, 2016]
[11] Bell, C., 2014. *MySQL High Availability: Tools for Building Robust Data Centers*. 2nd ed. Sebastopol, CA: O'Reilly Media.
[12] Skvorc, Bruno, 2015. 1st ed. *Jump Start PHP Environment.*Collingwood, Australia: SitePoint Pty.
[13] Lombardi, Andrew, 2015. 1st ed. *Websocket*. Sebastopol, CA: O'Reilly Media.
[14] Frenzel, Louis., 2016. *Handbook of Serial Communication Interfaces.* Waltham, MA: Elsevier.
[15] Henderson, H., 2016. *Encyclopedia of computer science and technology* .2nd ed. New York, NY: Facts On File
[16] Firmata Library. 2016. Available at: https://www.arduino.cc/en/Reference/Firmata [Accessed: Nov.14, 2016]