

PROTOTYPE PENGENDALI RUMAH CERDAS MULTIPLATFORM

Beni Krisbiantoro¹, Gading Reno Sasmito²

^{1,2}STT Atlas Nusantara
beni.krisbiantoro@gmail.com

Abstrak. Rumah merupakan bangunan tempat tinggal yang didalamnya terdapat perabot rumah tangga dan juga perangkat elektronik seperti lampu, kipas angin, pompa air, dan lain sebagainya. Sering terjadi pemilik rumah lupa mematikan perangkat elektronik ketika sedang bepergian atau meninggalkan rumah, sehingga membuat biaya listrik akan menjadi lebih mahal. Perlu sebuah sistem yang dapat mengontrol dan memonitoring perangkat elektronik yang ada di rumah. Dengan menggunakan mini komputer Raspberry Pi dapat dibuat pengendali rumah cerdas multiplatform yang terhubung oleh relay dan terkoneksi dengan database Firebase sebagai media penyimpanan online secara realtime. Pemilik rumah dapat memonitoring perangkat elektronik dari jarak jauh dengan mengakses situs web maupun dari aplikasi Android yang dibuat, bisa dari tablet, smartphone, komputer, serta media lainnya. Selain dapat menghidupkan dan mematikan perangkat dari jarak jauh, pemilik rumah juga dapat memantau kondisi dari perangkat Raspberry Pi itu sendiri serta kondisi suhu maupun kelembaban rumah yang ditinggalkan dari pembacaan sensor. Hasil yang didapat dari prototype pengendali rumah cerdas multiplatform ini maka pengguna dapat melakukan kendali dan monitoring perangkat elektronik di rumah secara realtime kapanpun dan dimanapun berada melalui halaman web atau aplikasi Android.

Kata Kunci: rumah cerdas, smarthome, raspberry pi, multiplatform

PENDAHULUAN

Rumah adalah salah satu kebutuhan penting manusia dalam hidup. Rumah adalah bangunan fisik untuk berlindung, yang didalamnya dapat berisi barang-barang yang dimiliki oleh manusia seperti furnitur dan perangkat elektronik. Kebanyakan pemilik rumah saat ini setidaknya memiliki beberapa perangkat dasar di dalam rumahnya, seperti lampu, kipas, ataupun pompa air. Penggunaan perangkat elektronik ini akan mempengaruhi biaya listrik, terutama ketika penghuni rumah tidak dapat mengontrol berapa besar konsumsi pemakaian listrik. Masalah umumnya adalah ketika penghuni harus meninggalkan rumah selama beberapa hari, misalnya karena harus bekerja atau bepergian ke tempat yang jauh. Terkadang, pemilik rumah lebih suka menyalakan perangkat elektronik, terutama lampu, untuk menghindari pencuri di malam hari. Sedangkan pada saat siang hari, karena pemilik rumah tidak dapat mematikan aliran listrik, maka akan mengakibatkan ketidakefisienan dalam pemakaian listrik dan berakibat tagihan untuk biaya listrik yang muncul. Oleh karena itu, pemilik rumah membutuhkan sistem yang dapat mengendalikan secara otomatis perangkat elektronik agar bekerja lebih cerdas serta dapat memonitor kondisi perangkat elektronik tersebut pada saat pemilik rumah sedang meninggalkan rumah.

Internet of Things (IoT) pada saat ini semakin banyak diaplikasikan secara luas dan teknologinya semakin matang menjadi konsep terbaru dan paling hyped di dunia IT. Selama dasawarsa terakhir istilah *Internet of Things* telah menarik perhatian dengan visi unruk menghubungkan semua perangkat yang ada dalam infrastruktur jaringan secara global, yang memungkinkan terjadinya saling koneksi antar perangkat kapanpun, dimanapun, untuk kebutuhan apapun dan bagi siapapun [1]. *Internet of Things* juga dapat dianggap sebagai jaringan global yang memungkinkan komunikasi antara manusia ke manusia, manusia-ke-mesin dan mesin-ke-mesin, apapun itu di seluruh dunia dengan memberikan identitas yang unik untuk setiap obyek [2].

IoT mendeskripsikan sebuah dunia di mana hampir semua hal dapat dihubungkan dan saling berkomunikasi dengan cara yang cerdas dari apa yang pernah ada sebelumnya. Konsep *smarthome* jika dikaitkan dengan teknologi menurut salah satu definisi diberikan oleh Satpathy adalah “*sebuah rumah yang cukup cerdas untuk membantu pemiliknya untuk hidup mandiri dan lebih nyaman dengan bantuan teknologi yang disebut sebagai Smarthome. Dalam Smarthome, semua perangkat mekanik dan digital saling terhubung membentuk jaringan, dapat berkomunikasi satu sama lain serta dengan pengguna untuk menciptakan ruang yang interaktif*” [3]. Deskripsi ini menegaskan bahwa tujuan utama *smarthome* adalah untuk meningkatkan kenyamanan penghuni dan membuat kehidupan sehari-hari menjadi lebih mudah [4].

Berdasarkan konsep IoT, program pengendali untuk *smarthome* dibuat bertujuan untuk dapat menyelesaikan masalah ini. Dengan adanya kontroler *smarthome* multiplatform yang dibuat menggunakan mini komputer Raspberry Pi yang terhubung dengan relai, selanjutnya dihubungkan dengan basis data Firebase – basis data yang berada pada lingkungan cloud yang bersifat *real time*, pemilik rumah dapat memantau dan mengendalikan perangkat elektronik dari jarak jauh. Akses dapat dilakukan dari halaman web atau aplikasi Android yang dibuat dari menggunakan perangkat tablet, ponsel cerdas, komputer, ataupun gadget lain yang dimiliki oleh pengguna. Fitur yang dapat diakses oleh pemilik rumah antara lain: dapat menyalakan, mematikan perangkat elektronik dari jarak jauh, memantau kondisi perangkat Raspberry Pi itu sendiri, serta memantau kondisi suhu dan kelembaban di dalam lingkungan rumah. Sistem juga dapat secara cerdas mematikan dan menyalakan perangkat elektronik secara otomatis sesuai dengan kondisi rumah tanpa perlu banyak memerlukan interaksi dengan pemilik rumah.

1. KAJIAN PUSTAKA

1.1 Raspberry Pi

Raspberry Pi adalah mini komputer berharga murah dengan ukuran hanya sebesar kartu kredit yang dapat dihubungkan dengan monitor komputer atau TV, serta dapat menerima input dari keyboard dan mouse. Meskipun berukuran kecil, Raspberry Pi dapat digunakan untuk melakukan kegiatan komputasi, untuk mempelajari bahasa pemrograman seperti Scratch atau Python. Pengguna juga dapat menggunakan perangkat ini untuk melakukan kegiatan seperti di komputer desktop, antara lain browsing halaman web di Internet, memutar video kualitas tinggi, menjalankan aplikasi lembar kerja elektronik, pengolahan kata, serta untuk bermain game. [5]

Beberapa model dari:

1. Raspberry Pi Model A
2. Raspberry Pi Model A+
3. Raspberry Pi Model B
4. Raspberry Pi Model B+
5. Raspberry Pi 2
6. Raspberry Pi 3



Gambar 1 Raspberry Pi 3 Model B

1.2 Firebase

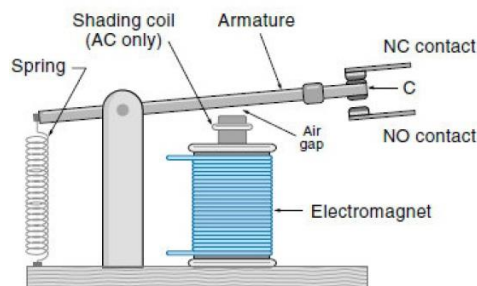
Firebase adalah layanan DbaaS (Database as a Service) dengan konsep *real time*. Firebase adalah penyedia layanan cloud untuk layanan *backend* berbasis di San Francisco, California. Firebase menyediakan fitur-fitur yang biasanya sesuai dengan kebutuhan pengguna. Firebase menyediakan juga perangkat dan infrastruktur untuk membangun aplikasi lebih baik.

Produk utama dari Firebase adalah basis data yang memberikan API yang memungkinkan developer untuk menyimpan dan melakukan sinkronisasi data ke banyak client. Bagi developer yang mengembangkan aplikasi web dengan teknologi HTML, CSS dan JS, Firebase selain menyediakan teknologi server side dan basis data, juga menyediakan layanan hosting dan penyimpanan file statis dalam bentuk layanan CDN. Beberapa contoh aplikasi yang bersifat real time antara lain BBM, Whatsapp, Facebook ataupun aplikasi lain agar dapat bersifat real time dapat menggunakan basis data dan *tool* lain di dalamnya yang telah disediakan oleh Firebase. Firebase menyediakan library untuk berbagai macam platform client. Sebagai contoh untuk browser menggunakan JavaScript dan untuk platform mobile menggunakan Objective-C atau Android API [6].

1.3 Relay

Relay dalam rangkaian bersifat menjadi pelindung utama – dengan cara kerja seperti halnya switch – dalam sebagian besar proses dalam kontrol atau peralatan baik yang bersifat elektronik atau elektromekanik. Relay memberikan respon terhadap adanya perubahan kondisi listrik, baik voltase atau arus, yang mengakibatkan kontak relay berstatus open atau close. Relay adalah perangkat switch yang dapat memutus atau mengubah state dari rangkaian listrik dari satu kondisi ke kondisi yang lain. Relay memungkinkan satu sirkuit untuk beralih ke sirkuit kedua yang dapat benar-benar terpisah dari yang pertama. Tidak ada sambungan listrik di dalam relay antara dua sirkuit tersebut - tautan bersifat magnetis dan mekanis saja.

Pada dasarnya relay terdiri dari kumparan elektromagnet, lengan, pegas dan serangkaian kontak listrik. Kumparan elektromagnet mendapat daya melalui switch atau driver relay dan menyebabkan lengan terhubung sehingga beban mendapatkan catu daya. Gerakan lengan disebabkan karena adanya pegas. Dengan demikian, relai terdiri dari dua sirkuit listrik terpisah yang terhubung satu sama lain hanya melalui koneksi magnetik, dan relai dikontrol dengan mengendalikan kondisi elektromagnet.



Gambar 2 Cara Kerja Relay

Kontak pada realy biasanya terdiri dari *common* (COM), *normally open* (NO) dan *normally closed* (NC). Kontak NC terhubung ke kontak COM jika daya tidak diberikan ke koil. Kontak NO terbuka jika daya tidak diberikan ke koil. Ketika kumparan diberi energi, kontak COM terhubung ke kontak NO, dan kontak yang NC pada kondisi mengambang. Versi *double-pole* sama dengan versi *single-pole* kecuali ketika dua switch terbuka dan tertutup bersamaan. [7] Modul relay terdiri dari banyak relay dan dimungkinkan untuk dikontrol dari Raspberry Pi.



Gambar 3 Modul Relay

2. METODE PENELITIAN

2.1 Analisa Masalah

1. Ketika pemilik rumah sedang meninggalkan rumah, pemilik rumah tidak dapat mengontrol nyala lampu dan perangkat elektronik lain yang ada di rumah.
2. Pemilik rumah sering lupa untuk mematikan perangkat elektronik yang ada di rumah ketika meninggalkan rumah, sehingga perangkat elektronik tersebut terus menyala dan menggunakan daya listrik.

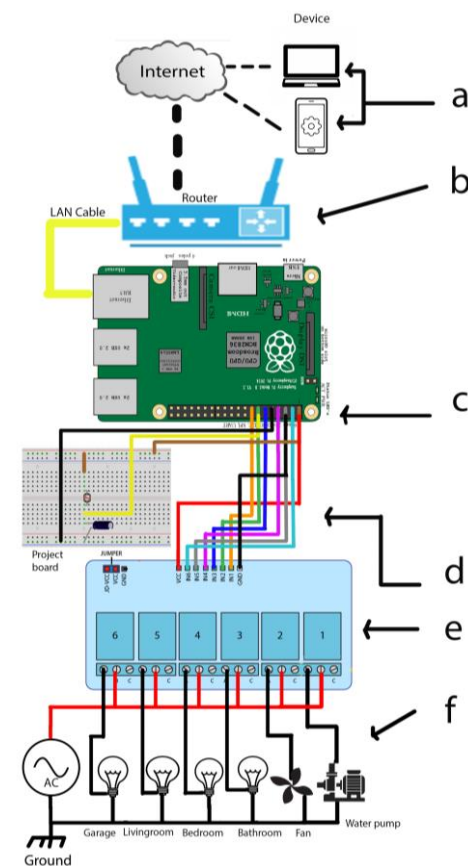
2.2 Kebutuhan Fungsional

1. Nyala lampu listrik dalam rumah harus dapat dikendalikan oleh pemilik rumah dari jarak jauh ketika sedang tidak berada di rumah. Kendali dapat dilakukan dari halaman web atau aplikasi Android.
2. Lampu teras dan garasi di luar rumah secara otomatis dapat menyala dan padam sesuai dengan kondisi cahaya matahari.
3. Sistem juga dapat mengendalikan kipas dan pompa air.
4. Pengguna harus login terlebih dahulu ke sistem untuk dapat mengendalikan perangkat.
5. Sistem juga dapat memberikan informasi yang up-to-date mengenai status dari semua perangkat elektronik.

2.3 Solusi

1. Perlu dibuat aplikasi Android dan halaman web yang memungkinkan pengguna (pemilik rumah) untuk mengendalikan perangkat elektronik lampui, kipas dan juga pompa air.
2. Dirancang sistem yang dapat memberikan fasilitas kepada pengguna untuk mengontrol kondisi perangkat elektronik dalam kondisi mati atau menyala secara manual dari tombol switch maupun dari aplikasi.
3. Sistem juga dirancang agar pengguna dapat melihat informasi secara up-to-date mengenai status dari perangkat elektronik yang dikontrol maupun kondisi perangkat Raspberry itu sendiri.

2.4 Perancangan Sistem

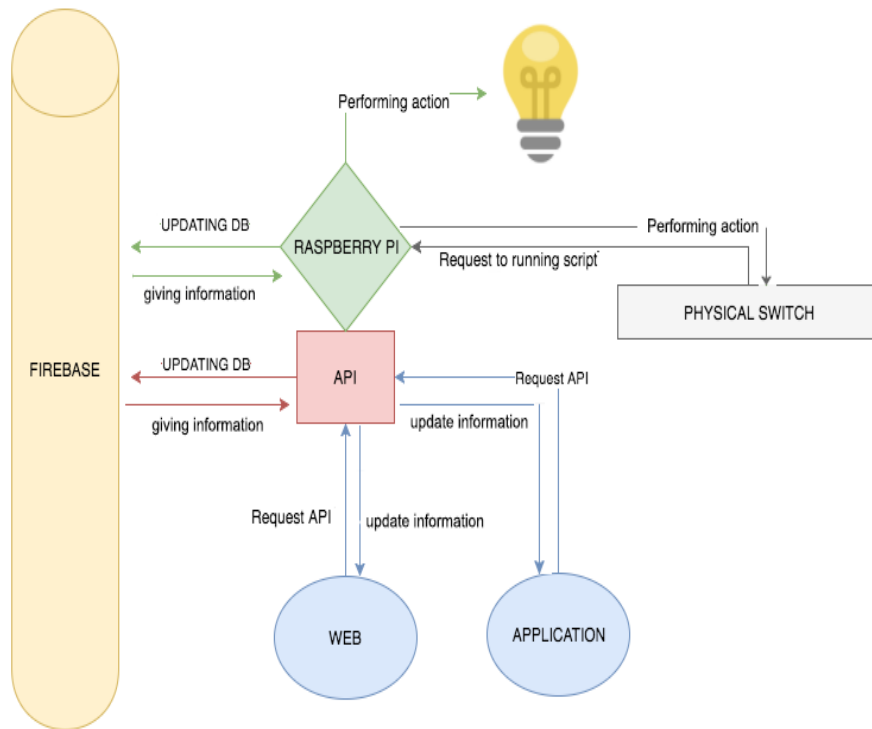


Gambar 4. Setup Perangkat dan Pengkabelan

Gambar 4 menunjukkan rancangan setup perangkat keras dari sistem *smarthome* yang dibuat. Berikut adalah penjelasan dari setiap bagian:

1. Pemilik rumah melalui perangkat gadget yang terhubung ke jaringan Internet akan mengendalikan perangkat elektronik melalui sistem *smarthome* ini. Akses dilakukan melalui halaman web dan aplikasi Android yang terinstall di gadget pemilik rumah.
2. Raspberry Pi dihubungkan melalui kabel LAN ke router agar dapat terhubung ke Internet dan dapat mengakses basis data Firebase, serta mengupdate informasi terbaru ke aplikasi web.
3. Raspberry Pi terhubung ke modul relay melalui port *General Purpose Input and Output* (GPIO).
4. Kabel jumper *female to female*.
5. Modul terdiri dari 6 buah relay yang diset ke *normally open* (NO), setiap relay akan digunakan untuk mengontrol satu perangkat elektronik.
6. Perangkat elektronik yang ada di rumah yang akan dikontrol.
7. Sensor cahaya dan suhu.

Gambar 5 adalah arsitektur dari keseluruhan sistem *smarthome* ini beserta penjelasan interaksinya.



Gambar 5. Diagram Arsitektur

3. HASIL DAN PEMBAHASAN

Pembahasan pada bagian ini akan menjelaskan perancangan perangkat keras yang telah dilakukan, pengkabelan dan juga hasil pengujian dari sistem. Prototype pengendali *smarthome* ini mempunyai ukuran panjang 80cm, lebar 60cm dan tinggi 11 cm, serta di dalamnya terdapat lampu LED, kipas, pompa air, sensor suhu dan cahaya.

3.1 Prototype Smarthome

Gambar 6 adalah bentuk akhir dari prototype *smarthome* ini berdasarkan rancangan yang telah dijelaskan sebelumnya pada Sub Bab 3. Material dari prototype ini dibuat dengan mempertimbangkan kebutuhan perangkat keras dan dibuat dari bahan triplex dengan ketebalan 5mm dan bahan akrilik dengan tebal 2mm untuk bagian atap.



Gambar 6 Prototype Smarthome (Tampak Atas)



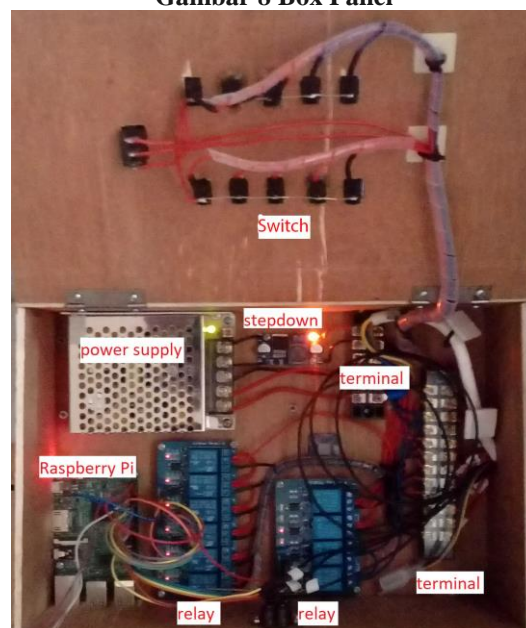
Gambar 7 Prototype Smarhome (Tampak Samping)

Prototype *smarhome* ini dilengkapi dengan box panel dengan ukuran 20cm x 30cm x 11cm. Box panel berisi Raspberry Pi, modul relay, switch, trafo stepdown, power supply 12V/10A dan terminal.

Pada sisi kiri dari box panel terdapat switch untuk memberikan 2 pilihan pengendalian perangkat elektronik yang ada di rumah, secara manual dengan menggunakan switch atau secara otomatis oleh sistem pengendali.



Gambar 8 Box Panel



Gambar 9 Bagian Dalam Box Panel

3.2 Web dan Aplikasi Android Pengendali

Halaman web disediakan agar pemilik rumah dapat mengendalikan dan juga memonitor kondisi dari perangkat elektronik. Halaman web dapat diakses pada alamat <http://knockknock.sttar.ac.id>.

3.2.1 Halaman Login

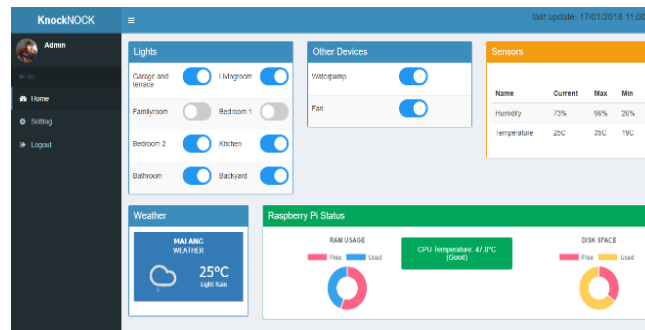
Sebelum pengguna dapat mengakses menu-menu pada halaman pengendali, diharuskan untuk melakukan login terlebih dahulu melalui web browser.



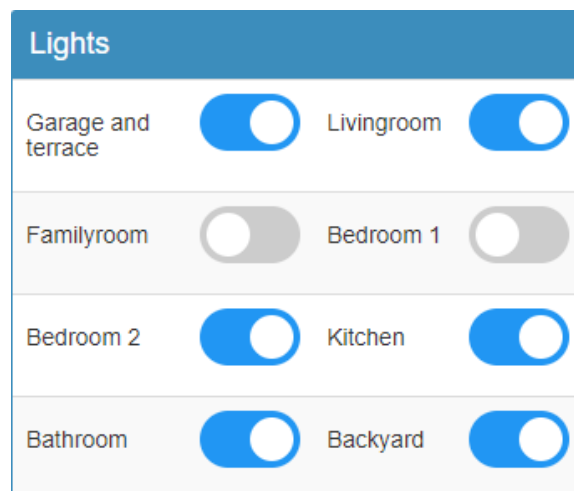
Figure 10. Halaman Login

3.2.2 Halaman Utama

Halaman utama dari web pengendali menampilkan status dari setiap perangkat elektronik yang dikontrol, seperti lampu, pompa air, kipas, kondisi suhu dan kelembaban lingkungan rumah, serta status dari perangkat Raspberry itu sendiri.

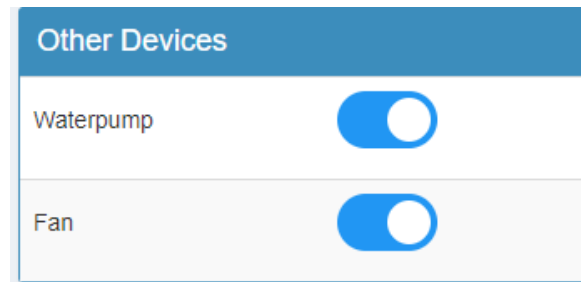


Gambar 11. Halaman Utama



Gambar 12. Panel Kendali Lampu

Sebagai contoh pada Gambar 12 adalah tampilan panel kendali lampu. Terlihat status lampu Familyroom dan Bedroom 1 adalah tidak aktif, yang artinya dalam kondisi real lampu tersebut dalam kondisi sedang tidak menyala. Status dari semua lampu yang lain juga dapat ditampilkan. Pemilik rumah juga dapat mengubah status dari lampu yang dikehendaki (dari padam menjadi menyala atau sebaliknya) dengan melakukan klik pada tombol panel lampu di halaman aplikasi.



Gambar 13. Panel Kendali Perangkat Lain

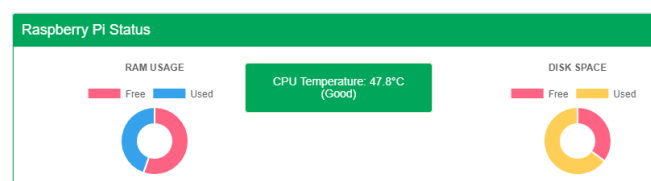
Panel pengenal untuk perangkat lain pada sistem ini berisi pengendali untuk perangkat pompa air dan kipas. Sebagaimana panel kendali lampu, perangkat di sini juga dapat dilihat dan diubah statusnya oleh pemilik rumah.

The image shows a table titled 'Sensors' with the following data:

Name	Current	Max	Min
Humidity	73%	96%	26%
Temperature	25C	35C	19C

Gambar 14. Panel Sensor

Pada panel sensor ini terdapat empat kolom untuk menampilkan nama, nilai aktual (Current), nilai maksimum (Max) dan nilai minimum (Min) dari sensor kelembaban dan suhu. Status "Current" menampilkan data suhu yang dibaca oleh sensor secara aktual. Status Max akan menampilkan data tertinggi yang pernah dibaca. Ketika data yang dibaca sekarang lebih tinggi dari nilai data tertinggi yang tersimpan, maka data Max akan diupdate dan disimpan. Sedangkan status Min akan menampilkan data terendah dari data yang pernah dibaca. Jika data yang terbaca sekarang lebih rendah dari nilai terendah yang tersimpan, maka data Min akan diupdate dan disimpan.



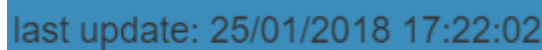
Gambar 15. Status Raspberry Pi

Ada tiga penunjukan informasi di dalam tampilan status di Raspberry Pi, yaitu penggunaan RAM, suhu CPU dan sisa kapasitas media penyimpanan. Informasi ini dapat digunakan untuk memantau kondisi real dari perangkat Raspberry Pi itu sendiri.



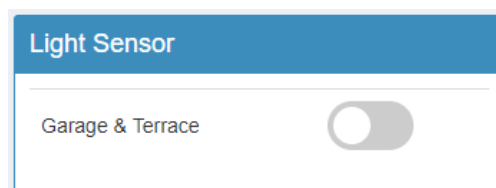
Gambar 16. Panel Cuaca

Panel cuaca berisi widget yang menampilkan perkiraan cuaca dari penyedia layanan cuaca berdasarkan kota yang dipilih.

A screenshot of a 'last update' widget. The text is 'last update: 25/01/2018 17:22:02'.

Gambar 17. Last Update

Informasi Last Update pada navbar menampilkan kapan terjadi update terbaru dari pemcaaan sensor dan status perangkat Raspberry Pi. Informasi tampil dalam bentuk tanggal dan waktu update dengan format dd/mm/yy hh:mm:ss. Status Last update akan berubah jika ada perubahan nilai hasil pembacaan salah satu sensor yang terhubung ke Raspberry Pi, meliputi sensor kelembaban, suhu, penggunaan RAM, suhu CPU dan sisa kapasitas penyimpanan.



Gambar 18. Sensor Cahaya

Gambar 18 adalah adalah bagian pengendali sensor cahaya, dalam hal ini statusnya tidak aktif. Ini menandakan lampu garasi dan teras hanya dapat dikendalikan menyala ataupun padam secara manual melalui switch panel lampu. Ketika panel sensor pada Gambar 18 diaktifkan, maka lampu garasi dan teras akan menyala dan padam secara otomatis tergantung kondisi cahaya matahari.

3.2.3 Pengujian Program

Tujuan dari pengujian program ini adalah untuk melihat hasil pembacaan sistem secara realtime dan memastikan nilai yang sama antara data yang dibaca oleh perangkat dengan yang terdapat di web serta di basis data Firebase ketika program dijalankan.

Ketika program berjalan, Raspberry Pi akan mencoba melakukan sinkronisasi data update terakhir dengan membaca semua nilai yang tersimpan di basis data Firebase pada semua fungsi dan sensor. Dengan mengakses Raspberry Pi menggunakan protokol SSH ke port 22 Raspberry Pi menggunakan IP yang sudah dipasang dimungkinkan untuk melakukan eksekusi program utama yang dibuat dengan menggunakan bahasa Python.

```

25/01/2018 17:22:00
Temperature: Current[25], Max[27.20], Min[19.00]
Humidity: Current[81], Max[96.10], Min[37.30]
CPU temperature: 45.6
RAM total[945.0 MB], RAM used[319.0 MB], RAM free[626.0 MB]
DISK total[7.1G], free[2.5G], perc[64%]
## Update finished successfully ##
=====

```

Gambar 19. Status Raspberry Pi

Gambar 19 menunjukkan output program hasil pembacaan sensor dan kondisi perangkat Raspberry Pi, meliputi suhu, kelembaban, status CPU, penggunaan RAM dan sisa kapasitas penyimpanan yang tersimpan di Firebase. Begitu data dibaca, maka perangkat yang dikontrol juga akan diupdate statusnya.

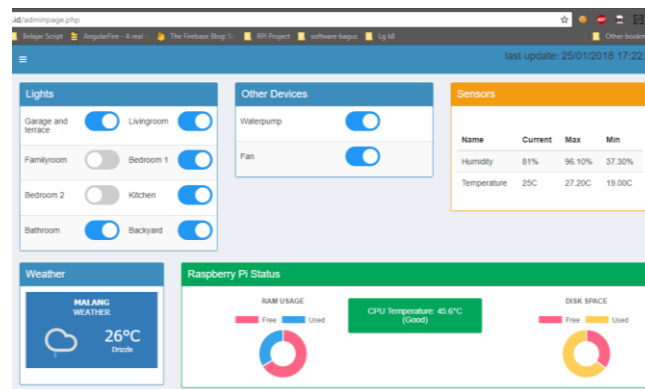
```

=====
garage on
livingroom on
familyroom off
bedroom2 off
bedroom1 on
kitchen on
bathroom on
backyard on
waterpump on
fan on
ldr on
=====

```

Gambar 20. Update Status Perangkat

Gambar 20 adalah status terbaru dari perangkat yang dikontrol yang berhasil diupdate dan disimpan ke dalam basis data Firebase.



Gambar 21. Status Update Pada Web Pengendali

Data yang ditampilkan di Raspberry Pi dengan yang ditampilkan di halaman web dapat dibandingkan seharusnya adalah sama.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan pembahasan dari sebelumnya mengenai “Prototype Pengendali Rumah Cerdas Multiplatform” ini maka dapat disimpulkan sebagai berikut:

1. Prototype Pengendali Rumah Cerdas Multiplatform dapat dibuat dengan menggunakan mini komputer Raspberry Pi. Raspberry Pi dihubungkan ke perangkat elektronik yang akan

dikontrol melalui modul relay, meliputi lampu, kipas dan pompa air. Dengan bahasa Python dapat dilakukan pembacaan data dan pengendalian perangkat yang dikontrol.

2. Basis data Firebase digunakan untuk menampung data real time agar dapat ditampilkan dan diubah nilainya melalui halaman web dan aplikasi Android.
3. *Smarthome* ini dapat dikendalikan melalui dua cara: secara manual dan otomatis. Mode manual berarti lampu dan perangkat elektronik lain dioperasikan menggunakan switch. Sedangkan pada mode otomatis, perangkat dapat dikendalikan melalui web dan aplikasi Android. Lampu garasi dan teras mempunyai 2 metode untuk menyalakan dan memadamkannya, melalui tombol yang disediakan di halaman web dan aplikasi Android, atau secara otomatis berdasarkan kondisi cahaya matahari yang dibaca oleh sensor cahaya.
4. Untuk monitoring dari *smarthome* ini, pengguna dapat memantau secara real time suhu dan kelembaban ruangan, perkiraan kondisi cuaca, status perangkat Raspberry Pi itu sendiri, serta kondisi semua lampu di rumah, kipas dan pompa air.

4.2 *Saran*

1. Berdasarkan dari pengujian yang telah dilakukan, disarankan untuk membuat kode program lebih efisien dan menggunakan basis data realtime lain untuk menurunkan nilai delay ketika pembacaan dan update data dengan ketika ditampilkan.
2. Inisialisasi port GPIO di Raspberry Pi tidak menggunakan library `rpi.gpio` tetapi dapat menggunakan library `gpiozero` yang dapat melakukan update ke setiap perangkat ketika program dijalankan.
3. Raspberry Pi diberikan pendingin tambahan yang dapat bekerja otomatis ketika berada di atas suhu threshold agar supaya kondisi suhu CPU tetap terjaga.
4. Sistem ini dapat dikembangkan lebih lanjut agar dapat mengenali dan diperintah dengan menggunakan suara manusia, sebagai contoh untuk mematikan atau menyalakan perangkat.
5. Perangkat yang dikendalikan dapat dikembangkan lebih banyak serta lebih cerdas dengan mengacu kepada perangkat yang banyak terdapat di rumah modern saat ini, misalnya: Air Conditioner yang dapat bekerja sesuai dengan kondisi suhu ruangan.

6 DAFTAR PUSTKA

- Aggarwal, R., & Das, L. (2012). First International Conference on Security of Internet of Things. *RFID Security in the Context of "Internet of Things"*, 51-56.
- Cook, D., & Das, S. (2007). Pervasive Mob. Comput. *How smart are our environments? An updated look at the state of the art*, 53-73.
- Dira. (n.d.). *Komunitas TIK Bandung Jawa Barat Indonesia*. Retrieved November 12, 2017, from <http://www.saungit.org/2016/07/apa-itu-firebase.html>
- Edgefx (1997). Know about different types of relays. Eedgefx.
- Foundation, R. (2012). *Raspberry pi*. (Raspberry Pi) Retrieved march 2016, from <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
- Kosmatos, E. T., & Boucouvalas, A. (2011). *Integrating RFIDs and Smart Objects into a Unified Internet of Things Architecture*. Retrieved from Internet of Things: Scientific Research: <http://dx.doi.org/10.4236/ait.2011.11002>
- Satpathy, L. (2006). Smart Housing: Technology to Aid Aging in Place. *New Opportunities and Challenges*.