

RANCANG BANGUN *DASHBOARD* UNTUK PROTOTIPE MODUL LATIH *INTERNET OF THINGS* BERBASIS *FLOW-BASED PROGRAMMING*

Agus Wagyana* dan Dandun Widhiantoro

Program Studi Broadband Multimedia, Jurusan Teknik Elektro, Politeknik Negeri Jakarta
Jalan Prof. Dr. G.A. Siwabessy, Kampus UI Depok 16425, Jawa Barat

^{*)}E-mail: agus.wagyana@elektro.pnj.ac.id

ABSTRAK

Makalah ini membahas hasil penelitian penulis tentang perancangan dan pembuatan *dashboard* (*user interface*) untuk mengoperasikan prototipe modul latihan *Internet of Things* (IoT) dengan menggunakan pemrograman *flow-based*. Metode penelitian mencakup pengkajian rancangan perangkat keras dan pembuatan (pemrograman) *dashboard*, serta uji coba modul. Modul yang sudah dirancang terdiri atas perangkat keras berupa *single-board computer* Raspberry Pi 3 yang dilengkapi dengan sejumlah komponen input (*push button*, *variable resistor*, dan berbagai sensor) dan komponen output (LED, RGB LED, relay magnet, *buzzer*, motor servo, dan LCD karakter). Modul menggunakan Raspberry Pi karena memiliki spesifikasi dasar yang mendukung, seperti kecepatan prosesor dan memori yang cukup tinggi, ketersediaan GPIO untuk *interfacing* ke komponen I/O serta adanya *built-in WiFi adapter* untuk akses ke jaringan Internet. Sedangkan perangkat lunak untuk mengatur modul ini terdiri atas sistem operasi Raspbian Jessie Stretch dan aplikasi *flow-based programming* Node-RED untuk membuat *flow* aplikasi *user interface* IoT. Node-RED digunakan untuk membuat *dashboard* IoT empat *tab*, yaitu *Monitoring*, *Controlling*, *Home Automation*, dan *Smart Garden*. Dari hasil pengujian, *dashboard* yang dibuat sudah berfungsi secara normal dalam mengatur komponen I/O yang tersedia, seperti pemantauan sensor dan pengendalian atau display ke komponen output secara *real time*. Penelitian tahap selanjutnya adalah pengkajian implementasi aplikasi IoT yang sudah dibuat pada halaman web (*hosting*) agar bisa beroperasi secara *online* melalui Internet.

Kata kunci : Internet of Things, IoT, Raspberry Pi, Flow-Based Programming, Node-RED

PENDAHULUAN

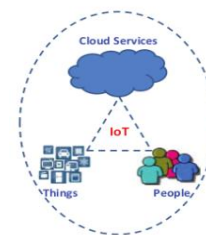
Pada jaringan konvensional, internet merupakan infrastruktur yang menyediakan terminal ke penggunaannya, sedangkan pada jaringan *Internet of Things* (IoT), internet memberikan interkoneksi antar benda pintar (*smart object*) di dalam lingkungan komputasi *ubiquitous* (Miorandi, 2012).

Istilah *Internet of Things* mengacu kepada arsitektur berbasis internet yang mendukung pertukaran layanan, informasi, dan data diantara milyaran benda pintar dan saat ini banyak mendapatkan perhatian dari kalangan industri dan akademisi (Santucci, 2009). IoT memungkinkan koneksi diantara benda-benda dan bisa membuat kehidupan manusia lebih nyaman dan efisien.

ITU mendefinisikan *Internet of Things* (IoT) sebagai “A global infrastructure for the information society enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies” (ITU work on Internet of things, 2015). IoT memungkinkan benda-benda fisik maupun virtual untuk saling berkomunikasi melalui infrastruktur internet dengan menggunakan layanan tertentu.

Salah satu model referensi untuk IoT secara skematik diperlihatkan pada Gambar 1 (Khajenasiri, 2016). Arsitektur sistem merupakan *framework* dimana benda-benda (*things*), manusia, dan layanan awan

(*cloud services*) digabungkan untuk mendukung aplikasi.



Gambar 1. Arsitektur Sistem IoT

Berbagai bidang bisa memanfaatkan layanan IoT, seperti transportasi, *smart city*, *smart domotics*, *smart health*, *e-governance*, *assisted living*, *e-education*, *retail*, logistik, pertanian, otomasi, industri manufaktur, dan manajemen bisnis/proses (Gubbia et al., 2013; Miorandi et al., 2012).

Sistem IoT terdiri atas beberapa blok fungsi, yaitu perangkat IoT, komunikasi, layanan, manajemen, sekuriti, dan aplikasi.

- Perangkat IoT** : Perangkat ini meliputi sensor, aktuator, pengontrol, dan pemonitor. Perangkat IoT dapat bertukar data dengan perangkat lain atau dengan aplikasi, atau mengumpulkan data dan mengolahnya baik lokal atau mengirimkannya ke server pusat atau aplikasi berbasis awan (*cloud*).
- Komunikasi** : Blok ini melakukan komunikasi diantara perangkat dan *remote server*. Protokol

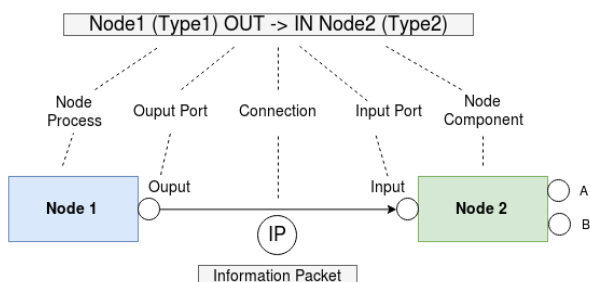
komunikasi IoT umumnya berada pada *data link layer*, *network layer*, *transport layer*, dan *application layer*.

- c. **Layanan** : Layanan IoT memberikan berbagai fungsi seperti pemodelan atau pengontrolan perangkat, publikasi data, dan analisis data.
- d. **Manajemen** : Blok ini memberikan berbagai fungsi untuk mengelola sistem IoT.
- e. **Sekuriti** : Blok keamanan data ini memberikan fungsi seperti otentikasi, otorisasi, privasi, integritas konten, dan keamanan data.
- f. **Aplikasi** : Lapisan aplikasi merupakan fungsi terpenting bagi *user* karena bertindak sebagai *interface* untuk mengendalikan dan memonitor berbagai hal dari sistem IoT. Aplikasi memberikan *user* kemampuan untuk memvisualisasikan dan menganalisis status sistem pada keadaan *real time*.

Penelitian ini membahas rancang bangun aplikasi untuk modul latih IoT sederhana. Aplikasi *user interface* yang berbentuk *dashboard* ini dikembangkan untuk memudahkan *user* dalam memonitor dan mengendalikan komponen-komponen input dan output yang digunakan. *Dashboard* dirancang dan dibuat dengan Node-RED, yang merupakan salah satu software pemrograman visual khusus untuk layanan IoT yang berbasis *Flow-based programming*

Flow-Based Programming dengan Node-RED

Flow-Based Programming atau FBP, yang diinvenisi oleh J. Paul Morrison (www.jpaulmorrison.com/fbp), merupakan pemrograman dengan cara menjelaskan fungsi-fungsi aplikasi sebagai “kotak hitam” jaringan atau “node”. Setiap data memiliki tujuan tertentu yang sudah didefinisikan; diberikan sejumlah data, melakukan sesuatu terhadap data tersebut lalu meneruskan data tersebut. Jaringan bertanggung jawab untuk mengalirkan data dalam bentuk *Information Packet* (IP) diantara node, seperti diperlihatkan pada Gambar 2.



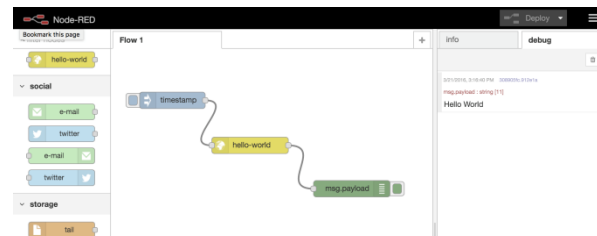
Gambar 2. Konsep dasar *Flow-based Programming*

Konsep FBP sudah digunakan di berbagai *platform*, yaitu Web (misalnya, NoFlo, Facebook Flux, Google TensorFlow, Microsoft Azure Event Hubs), di embedded systems (misalnya, ROS, MicroFlo, dan NodeRED), dan multimedia (misalnya, V4L, Gstreamer, Apple Quartz), (Barbieri, 2016).

Node-RED (<https://nodered.org>) merupakan *flow-based programming tool* untuk membantu pengembang IoT mengintegrasikan berbagai komponen I/O dengan *Application Program Interface* (API) dan layanan-layanan IoT. Node-RED menggunakan *node.js* sehingga bisa dijalankan pada jaringan atau di *cloud*.

Saat ini telah tersedia ribuan *flow* dan *node* di Node-RED *library* sehingga bisa dibuat berbagai koneksi dengan banyak perangkat dan layanan.

Node-RED terdiri atas *Node.js-based runtime* yang mengarahkan ke suatu *web browser* untuk mengakses *flow editor*. Melalui *web browser* tersebut, aplikasi bisa dibuat dengan *men-drag node-node* yang tersedia di *palette* ke *workspace* dan kemudian membuat rangkaian. Aplikasi kemudian dijalankan dengan *men-deploy* kembali ke *runtime*-nya. Gambar 3 memperlihatkan kumpulan palet (*social*, *storage*, dan lain-lain) dan *workspace* tempat *flow* program dibuat.



Gambar 3. Palet node dan *workspace flow* Node-RED

Palet node bisa diperluas dengan meng-*install* *node-node* baru yang dibuat komunitas dan *flow* yang sudah dibuat bisa di-*shared* sebagai file JSON.

Node-RED bisa di-*install* dan dijalankan secara lokal (laptop atau komputer), pada perangkat (Raspberry Pi, BeagleBone Black, atau Android), dan pada cloud (IBM Bluemix, SenseTecnica FRED, Amazon Web Services, atau Microsoft Azure).

METODE PENELITIAN

Rancang bangun *dashboard* untuk modul latih IoT ini meliputi rancang bangun system dalam bentuk prototyping dan rancang bangun serta pengujian *dashboard* Node-RED.

Perancangan Aplikasi

Modul latih IoT ini dirancang untuk membantu menjelaskan konsep dasar dan aplikasi sistem IoT khususnya dalam pemrograman Node-RED. Modul dirancang untuk 4 (empat) aplikasi yang berbeda, yaitu *monitoring*, pengontrolan, *Home Automation*, dan *Smart Garden*, seperti diperlihatkan pada Tabel 1.

Tabel 1. Aplikasi Modul Latih IoT

No	Aplikasi	Komponen I/O
1	Monitoring (memantau nilai sensor di Raspberry Pi dari Dashboard)	Sensor suhu, kelembaban, cahaya, jarak, gas, dan gerak; potensiometer, dan switch
2	Pengontrolan (mengontrol aktuator di Raspberry Pi dari Dashboard)	LED, RGB LED, Motor Servo, LCD karakter, Buzzer, dan Relay magnet
3	Home Automation (simulasi otomasi rumah sederhana)	Sensor suhu, cahaya, gerak, gas, relay magnet, dan buzzer
4	Smart Garden (simulasi kebun pintar)	Sensor suhu, cahaya, kelembaban tanah, relay magnet, dan buzzer, dan button

Fungsi setiap aplikasi Node-RED adalah sebagai berikut :

1. Aplikasi Monitoring

Flow Node-RED ini digunakan untuk pembacaan nilai-nilai sensor (analog dan digital) yang tersambung ke Raspberry Pi 3 dan memperlihatkannya di Dashboard.

2. Aplikasi Controlling

Flow Node-RED digunakan sebagai pengendali komponen output yang tersambung ke Raspberry Pi 3 melalui Dashboard.

3. Home Automation

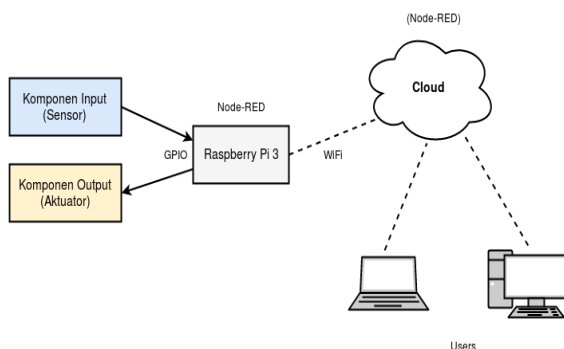
Flow ini digunakan untuk menunjukkan sistem otomasi rumah sederhana dengan menggunakan beberapa sensor dan komponen output.

4. Smart Garden

Flow ini digunakan untuk menunjukkan sistem kebun pintar sederhana dengan menggunakan beberapa sensor dan komponen output.

Rancang Bangun Sistem

Diagram sistem modul latih lengkap diperlihatkan pada Gambar 4. Sistem terdiri atas *single board computer* Raspberry Pi 3 (<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>) sebagai pengendali utama yang di-interfacing melalui slot *General Purpose Input Output* (GPIO) ke berbagai komponen input (sensor) dan output. Raspberry Pi disambungkan melalui koneksi WiFi ke *cloud* (internet). Data-data dari komponen input/output bisa dikirim ke/dari internet untuk disimpan di suatu *platform* layanan IoT. User bisa memantau dan mengendalikan komponen input/output melalui *web browser* di perangkat laptop. Aplikasi Node-RED digunakan untuk mengatur tampilan *user interface* (dashboard) di halaman web yang dituju.



Gambar 4. Diagram Sistem IoT

Raspberry Pi 3 dipilih karena pertimbangan ketersediaan adapter WiFi untuk akses internet secara langsung tanpa perlu tambahan apapun, Gambar 5 memperlihatkan Raspberry Pi 3 dan *pinout* GPIO-nya.



Pin No.	Pin No.	Pin No.
3.3V	1	2
GPIO2	3	4
GPIO3	5	6
GPIO4	7	8
GND	9	10
GPIO17	11	12
GPIO27	13	14
GPIO22	15	16
3.3V	17	18
GPIO10	19	20
GPIO9	21	22
GPIO11	23	24
GND	25	26
DNC	27	28
GPIO5	29	30
GPIO6	31	32
GPIO13	33	34
GPIO19	35	36
GPIO26	37	38
GND	39	40
5V	1	2
5V	3	4
GND	5	6
GPIO14	7	8
GPIO15	9	10
GPIO18	11	12
GND	13	14
GPIO23	15	16
GPIO24	17	18
GND	19	20
GPIO25	21	22
GPIO8	23	24
GPIO7	25	26
DNC	27	28
GND	29	30
GPIO12	31	32
GND	33	34
GPIO16	35	36
GPIO20	37	38
GPIO21	39	40

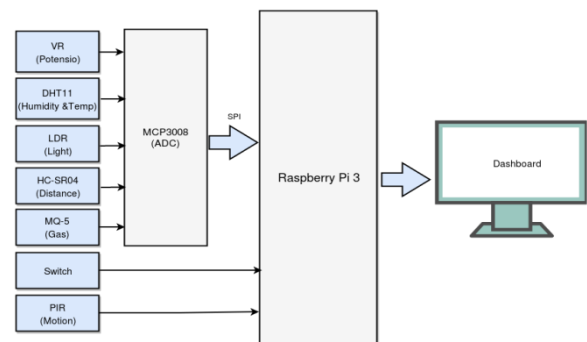
Gambar 5. Raspberry Pi 3 dan *pinout* GPIO

Pada penelitian ini, aplikasi Node-RED masih di-install dan dijalankan di Raspberry Pi 3 sehingga koneksi data masih berlangsung secara lokal antara komponen I/O dan *web browser* yang dijalankan di Raspberry Pi 3 ke URL lokal. Alamat URL program Node-RED berada di URL 127.0.0.1:1880. Sedangkan *dashboard*-nya berada pada URL 127.0.0.1:1880/ui.

Berikut ini akan dibahas rancangan antarmuka dari setiap aplikasi.

A. Aplikasi Monitoring

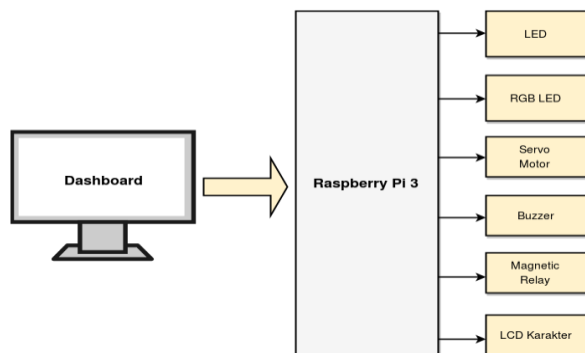
Diagram blok rangkaian *interfacing* untuk aplikasi monitoring diperlihatkan pada Gambar 6. MCP3008 merupakan chip 8 kanal (Ch0 sampai dengan Ch7) dengan koneksi SPI (*Serial Peripheral Interface*). Chip ini untuk mengubah nilai input analog dari sensor menjadi nilai digital 10 bit atau dari 0 sampai 1023 di outputnya. Chip ini diperlukan karena GPIO Raspberry Pi tidak memiliki pin analog input.



Gambar 6. Diagram Aplikasi Monitoring

B. Aplikasi Pengontrolan

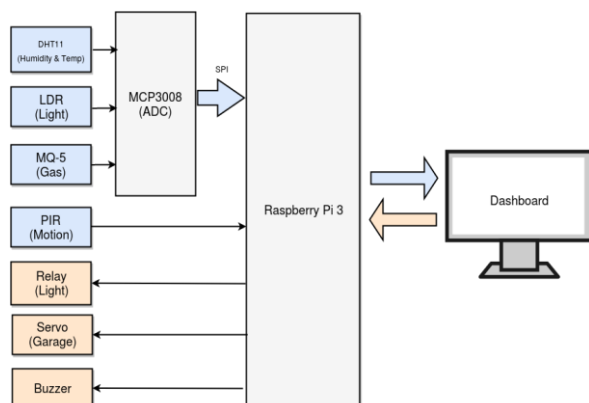
Diagram blok rangkaian *interfacing* untuk aplikasi pengontrolan diperlihatkan pada Gambar 7. *Dashboard* akan menampilkan tiga mode pengontrolan, yaitu *Analog Control*, *Digital Control*, dan *Character display*. *Analog control* menggunakan pin PWM pada Raspberry Pi untuk mengatur intensitas cahaya LED, warna RGB LED, dan sudut putaran motor servo. *Digital control* digunakan untuk mengatur suara buzzer dan meng-on/off-kan relay magnet. Sedangkan *character display* berfungsi untuk mengirimkan string karakter LCD karakter 16x2.



Gambar 7. Diagram Aplikasi Pengontrolan

C. Aplikasi Home Automation

Diagram blok rangkaian *interfacing* untuk aplikasi *Home Automation* sederhana ini diperlihatkan pada Gambar 8. Aplikasi ini akan memonitor nilai-nilai sensor yang ditempatkan di dalam rumah, yaitu suhu, cahaya, gas, dan gerakan kemudian akan mengontrol secara otomatis (kondisi relay dan buzzer) atau manual (servo).



Gambar 8. Diagram Rangkaian Home Automation

Cara kerja sistemnya diperlihatkan pada Tabel 2. Pengontrolan terdiri atas 2 mode, yaitu otomatis dan manual. Kontrol otomatis terjadi pada saat cahaya rendah atau gas tinggi, atau adanya gerakan. Sedangkan kontrol manual untuk membuka garasi rumah melalui motor servo..

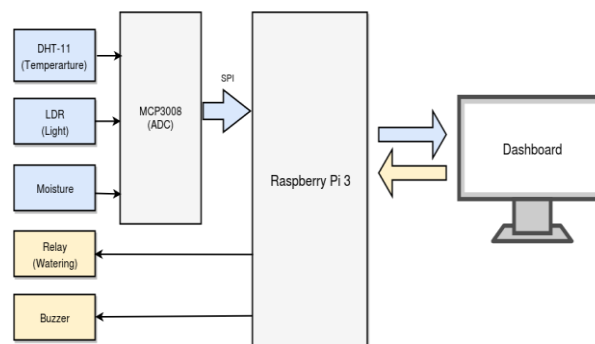
Tabel 2. Cara Kerja Aplikasi Home Automation

Sensor	Aktuator	Kontrol
Intensitas cahaya rendah	Relay magnet menghidupkan lampu	Otomatis
Kandungan gas tinggi	Buzzer hidup dan kirim e-mail	Otomatis
Gerakan	Buzzer hidup dan kirim e-mail	Otomatis
-	Motor servo (Garasi)	Manual

D. Aplikasi Smart Garden

Diagram blok rangkaian *interfacing* aplikasi *Smart Garden* sederhana ini diperlihatkan pada Gambar 9.

Aplikasi ini memonitor nilai-nilai sensor yang ditempatkan di kebun (suhu, cahaya, dan kelembaban tanah) kemudian mengontrol otomatis atau manual kondisi relay untuk pengairan.



Gambar 9. Diagram Aplikasi Smart Garden

Cara kerja sistemnya diperlihatkan pada Tabel 3. Pengontrolan terdiri atas mode otomatis atau manual yaitu untuk menghidupkan relay untuk mengairi tanah pada kondisi kering dan buzzer berbunyi sebagai alarm.

Tabel 3. Cara Kerja Aplikasi Smart Garden

Sensor	Aktuator	Kontrol
Kelembaban tanah rendah (tanah kering)	Relay magnet menghidupkan pengairan dan buzzer berbunyi serta mengirim e-mai ke user	Otomatis atau manual

Rancangan Dashboard

Dashboard terdiri atas 4 menu, yaitu *Monitoring*, *Controlling*, *Home Automation*, dan *Smart Garden*. *Dashboard Monitoring* menampilkan nilai-nilai sensor dalam tiga bentuk visualisasi, yaitu numerik, *chart*, dan *gauge*. *Dashboard Controlling* menampilkan *slider* (*analog control* untuk RGB LED, dan motor servo), *button* (*digital control* untuk LED, relay, dan buzzer), serta teks dan *button* pengiriman string karakter ke LCD karakter. *Dashboard Home Automation* menampilkan nilai-nilai suhu, cahaya, jarak, gas, dan gerakan dalam bentuk *gauge* (*automatic control*) dan *button* (*manual control*) untuk mengontrol relay dan servo. *Dashboard Smart Garden* menampilkan nilai suhu, cahaya, dan kelembaban tanah dalam bentuk *gauge* (*automatic control*) dan *button* (*manual control*) untuk mengontrol relay magnet.

Palet node tambahan di-install dari <https://flows.nodered.org/> melalui *Palette Manager*. Tabel 5 memperlihatkan node-node tambahan untuk mengendalikan komponen yang digunakan.

Tabel 4. Node Library Tambahan

Komponen	Node
LCD	node-red-node-pi-lcd 0.0.5
Dashboard	node-red-dashboard
MCP3008	node-red-node-pi-mcp3008 0.0.7
DHT11	node-red-contrib-dht-sensor 1.0.1

HASIL DAN PEMBAHASAN

Berikut ini akan disajikan hasil dan pembahasan rancang bangun sistem dan tampilan akhir *dashboard*.

Protoboarding

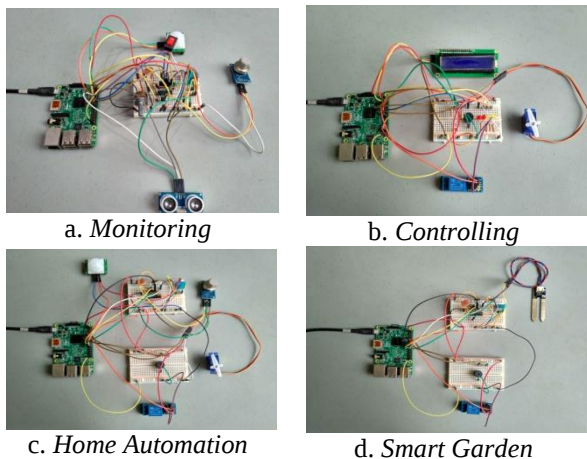
Setiap aplikasi pada modul latih direalisasikan dalam bentuk *protoboarding* rangkaian antarmuka Raspberry Pi 3 dan komponen-komponen input dan output. *Protoboard* digunakan untuk menempatkan komponen input dan output serta menyambungkannya ke pin-pin GPIO Raspberry Pi 3.

Aplikasi *monitoring* (Gambar 10.a) menggunakan sensor (DHT11, LDR, PIR, MQ-5, HC-SR04, potensiometer, dan switch) yang dihubungkan dengan pin-pin GPIO. LDR, MQ-5, dan potensiometer dihubungkan dengan kanal 0, 1, dan 2 dari MCP3008 sebelum dihubungkan ke GPIO. Sedangkan DHT11, PIR, dan switch langsung disambungkan ke pin digital input GPIO.

Aplikasi *controlling* (Gambar 10.b) menggunakan 4 LED, RGB LED, relay magnet, motor servo, LCD karakter, dan buzzer. LED, relay magnet dikendalikan oleh pin GPIO digital output, sedangkan RGB LED dan servo diatur oleh pin PWM.

Aplikasi *Home Automation* (Gambar 10.c) menggunakan sensor (PIR, MQ-5, DHT11, dan LDR) di *protoboard* bagian atas serta komponen output (relay, motor servo, dan buzzer) di *protoboard* bagian bawahnya.

Aplikasi *Smart Garden* (Gambar 10.d) menggunakan sensor (LDR, DHT11, dan sensor kelembaban tanah) serta komponen output, yaitu relay magnet dan buzzer.



a. Monitoring

b. Controlling

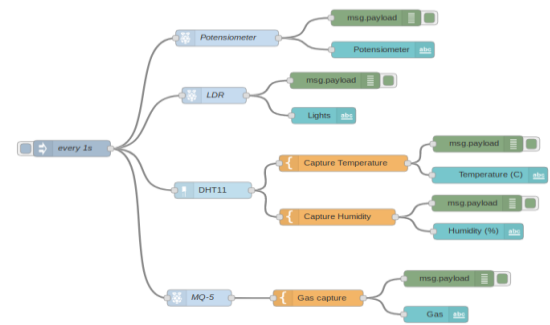
c. Home Automation

d. Smart Garden

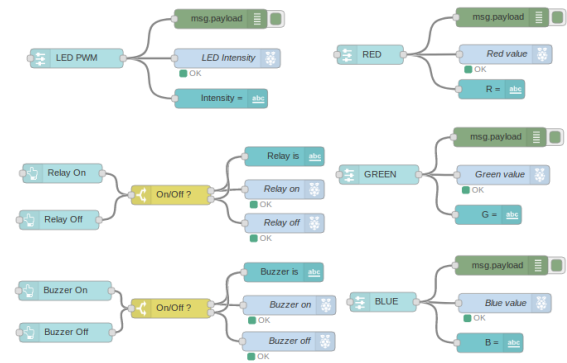
Gambar 10. Protoboarding Rangkaian

Flow Node-RED

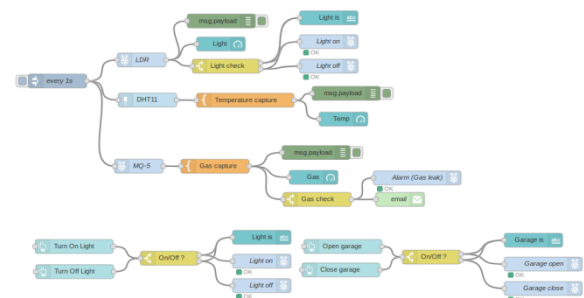
Flow Node-RED semua aplikasi diperlihatkan pada Gambar 11. Flow dibuat dengan menggunakan sejumlah palet yang sudah ada saat (*default*) ditambah beberapa palet kontribusi yang di-install melalui *Palette Manager*. Flow tersebut hanya menampilkan sebagian rangkaian node. Setiap flow dilengkapi juga dengan node *debug* untuk melacak nilai-nilai yang sedang digunakan (untuk keperluan *trouble shooting*).



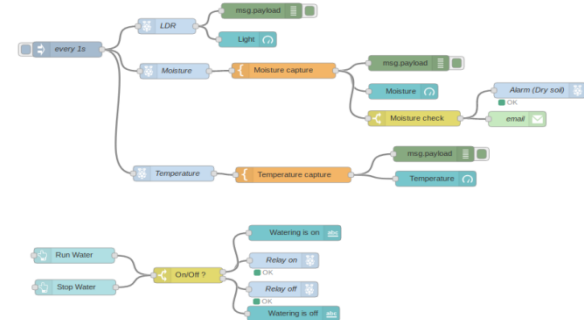
a. Aplikasi Monitoring



b. Aplikasi Controlling



c. Aplikasi Home Automation



d. Aplikasi Smart Garden

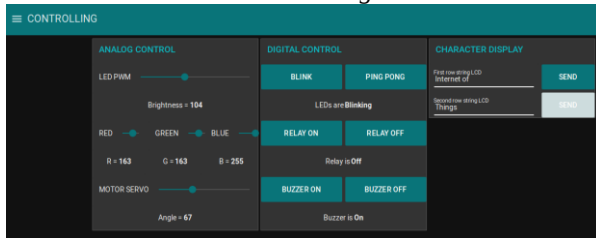
Gambar 11. Flow Aplikasi Node-RED

Tampilan Dashboard

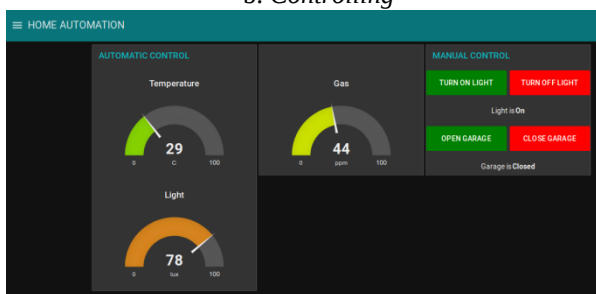
Tampilan *dashboard* menunjukkan sejumlah node untuk keperluan *user interface*, *output text*, *chart*, dan *gauge (monitoring)* serta *slider*, *button*, dan *input text (controlling)*. Semua *dashboard* dirancang menjadi *layout* tiga kolom supaya bisa muat di satu halaman saja tanpa perlu melakukan *scrolling* ke bawah.



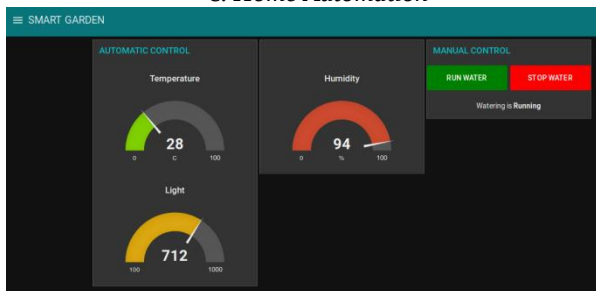
a. Monitoring



b. Controlling



c. Home Automation



d. Smart Garden

Gambar 12. Tampilan Dashboard

KESIMPULAN

Prototipe modul latihan *Internet of Things* ini berbasis Raspberry Pi 3 dan aplikasi *flow-based programming* Node-RED untuk membuat *flow dashboard (graphical user interface)* layanan IoT.

Node-RED digunakan untuk membuat *dashboard* empat aplikasi, yaitu *Monitoring*, *Controlling*, *Home Automation*, dan *Smart Garden*. Dari hasil pengujian, *dashboard* yang dibuat sudah berfungsi secara normal dalam mengatur komponen I/O yang tersedia, seperti pemantauan sensor dan pengendalian atau display ke komponen output secara *real time*. Penelitian tahap selanjutnya adalah pengkajian implementasi aplikasi IoT yang sudah dibuat pada halaman web (*hosting*) agar bisa beroperasi secara *online* melalui Internet.

REFERENSI

- Barbieri, G. S. 2016. Flow Based Programming Applied to IoT Development, *OpenIoT & ELC Europe2*.
<http://events.linuxfoundation.org/sites/events/files/slides/openiot2016-Flow%20Based%20Programming%20Applied%20to%20IoT%20Development.pdf>
- Gubbi, J., Buya, R., Marusic, S., Palaniswami, M. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems Volume 29, Issue 7, September 2013*, p.1645-1660.
<https://pdfs.semanticscholar.org/b9cb/47508b56fe34537fbf56321ff6ded99dd8ba.pdf>
- ITU Work on Internet of Things. 2015. ICTP workshop.
http://wireless.ictp.it/school_2015/presentations/secondweek/ITU-WORK-ON-IOT.pdf
- Khajenasiri, I., Estebasari, A., Verhelst, M., Gielen, G. 2016. A review on Internet of Things solutions for intelligent energy control in buildings for smart city applications.
- Miorandi, D., Sicari, S., De Pellegrini, Chlamtac, F., 2012. Internet of things : Vision, applications and research challenges. *Journal of Adhoc Networks Volume 10, Issue 7, September 2012*, Pages 1497-1516.
http://www.dicom.uninsubria.it/~sabrina.sicari/public/documents/journal/2012_IoT_vision.pdf
- Santucci, G. 2009. From internet of data to internet of things. *Paper for the International Conference on Future Trends of the Internet 2009* (Vol. 28).
http://cordis.europa.eu/pub/fp7/ict/docs/enet/20090128-speech-iot-conference-lux_en.pdf