

IMPLEMENTASI METODE WEIGHTED PRODUCT DALAM MEMONITOR GUDANG PENYIMPANAN ROTI BERBASIS INTERNET OF THING PADA PLATFORM NODE-RED

Anderias Eko Wijaya^{*1}, Haris Nurjaman^{#2}

Program Studi Teknik Informatika, STMIK Subang^{*1}

Program Studi Teknik Informatika, STMIK Subang^{#2}

E-mail: ekowjy09@yahoo.com^{*1}, harisnurjaman@gmail.com^{#2}

ABSTRAK

Metode Weighted Product (WP) merupakan salah satu metode yang dapat digunakan pada proses pengambilan keputusan untuk menentukan kandidat terbaik. Pada penelitian ini Metode WP digunakan untuk menentukan gudang penyimpanan roti untuk Roti Kuro Subang.

Roti Kuro Subang merupakan salah satu unit usaha yang saat ini bergerak dibidang pengolahan produk makanan khususnya roti. Seiring berkembangnya unit usaha maka diperlukan sebuah gudang penyimpanan roti. Untuk membantu pihak Roti Kuro Subang menentukan gudang penyimpanan roti diperlukan sistem yang dapat merekomendasikan gudang penyimpanan roti yang paling sesuai. Metode yang digunakan dalam pembuatan sistem ini adalah metode Weighted Product (WP). Metode ini dipilih karena mampu memberikan pilihan yang terbaik dari sejumlah alternatif. Dengan dimasukkannya sebuah konsep Internet of Things (IoT) pada penelitian kali ini dapat meningkatkan akurasi dalam menentukan nilai kriteria pada sejumlah alternatif.

Platform web yang digunakan adalah Node-red dan basis data yang digunakan adalah MySQL. Hasil dari penelitian ini adalah memberikan alternatif gudang penyimpanan roti yang paling sesuai untuk pihak Roti Kuro Subang.

Kata Kunci : *Gudang Penyimpanan, Internet of Thing (IoT), Metode Weighted Product (SAW)*

ABSTRACT

The weighted Product (WP) method is one method that can be used in the decision-making process to determine the best candidate. In this study, the WP method is used to determine the bread storage for Roti Kuro Subang.

Roti Kuro Subang is one of the business units currently engaged in the processing of food products, especially bread. As the business unit develops, a bread storage warehouse is needed. To help the Roti Kuro Subang determine the bread storage warehouse needed a system that can recommend the most suitable bread storage warehouse. The method used in making this system is the Weighted Product (WP) method. This method was chosen because it can provide the best choice from several alternatives. The inclusion of an Internet of Thing (IoT) concept in this study can improve accuracy in determining the value of criteria in several alternatives.

The web platform used is Node-red and the database used is MySQL. The results of this study are to provide alternative bread storage that is most suitable for the Roti Kuro Subang.

Kata Kunci : *Internet of Thing (IoT), Metode Weighted Product, Warehouse*

1. Pendahuluan

1.1 Latar Belakang

Roti Kuro Subang merupakan salah satu unit usaha yang saat ini bergerak di bidang pengolahan produk makanan khususnya roti. Tidak seperti roti pada umumnya roti yang dijual Roti Kuro Subang adalah roti tawar yang telah diolah kembali sebelum dijual ke konsumen, sehingga produk jadinya berupa roti kukus dan roti panggang. Karena tidak memproduksi bahan baku produksinya sendiri yaitu roti tawar, saat ini Roti Kuro Subang memesan terlebih dahulu sebelum

olah kembali menjadi produk jadi, serta karena jam operasionalnya dimulai dari siang sampai dengan malam maka selang waktu tersebut bahan baku tersebut akan disimpan terlebih dahulu di tempat penyimpanan. Dengan roti tawar sebagai bahan baku utamanya maka perlu penanganan khusus dalam hal penyimpanan. Roti merupakan makanan yang mampu bertahan tanpa adanya tanda-tanda pembusukan selama 5 hari di mana suhu dan kelembaban lingkungan menjadi perbedaan besar. Seiring berkembangnya unit usaha maka jumlah bahan baku yang dipesan akan naik kuantitasnya, tentunya dibutuhkan tempat penyimpanan yang memadai dan baik.

Gudang adalah sebuah bangunan atau ruangan yang digunakan sebagai tempat penyimpanan roti tawar sebelum diolah kembali dan dijual kepada konsumen. Gudang menurut Bowersox [1], Gudang adalah lokasi untuk penyimpanan produk sampai permintaan (demand) cukup besar untuk melaksanakan distribusinya, penyimpanan dianggap perlu untuk menyesuaikan produk dengan kebutuhan konsumen. Dan gudang mempunyai peranan yang sangat penting untuk menyimpan roti sehingga roti mampu bertahan yang cukup lama dengan mencegah pembusukan roti tersebut, karenanya teknik penyimpanan yang salah akan menurunkan kualitas roti. Oleh karena itu penentuan lokasi gudang tersebut sangat penting bagi unitusaha demi kelancaran usahanya.

Metode Weighted Product adalah salah satu metode dari beberapa metode MADM (Multi Attribute Decision Making). Kusumadewi, pada dasarnya proses MADM dilakukan melalui 3 tahap, yaitu penyusunan komponen-komponen situasi, analisis, dan sintesis informasi. Pada tahap penyusunan komponen, komponen situasi akan dibentuk tabel taksiran yang berisi identifikasi alternatif dan spesifikasi tujuan, kriteria dan atribut [2].

Untuk menunjang dalam penentuan nilai suatu atribut atau kriteria maka dalam penelitian ini penulis akan mengaitkannya dengan suatu konsep berupa Internet of Things, atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep baru yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus, adapun kemampuannya seperti berbagi data, remote control, dan sebagainya. Diharapkan dengan dimasukkannya IoT pada penelitian kali ini dapat meningkatkan nilai akurasi atau ketepatan dalam menentukan peringkat suatu tempat yang akan diuji sebagai gudang tempat penyimpanan roti.

Penelitian yang telah dilakukan menggunakan metode Weighted Product yaitu Wahyudin dan Masniah [2] yang meneliti tentang penilaian Eco Office Award pada dinas lingkungan hidup Kabupaten - Kota. Jurnal ini menjelaskan cara untuk menentukan kriteria penilaian, pembobotan penilaian dan pemeringkatan suatu lingkungan dan penghijauan kantor, serta menganalisis dan mengevaluasi implementasi penerapan metode Weighted Product tersebut dalam mendapatkan hasil yang lebih akurat dalam menentukan pemenang Eco Office Award.

Oleh karena itu, penulis berinisiatif untuk merancang suatu sistem dengan menggunakan sebuah metode yaitu metode Weighted Product dan konsep IoT yang diharapkan dapat membantu pihak Roti Kuro Subang dalam menentukan lokasi gudang penyimpanan baru.

1.2 Identifikasi Masalah

Identifikasi permasalahan yang diperoleh dari latar belakang sebagai berikut:

1. Pentingnya keakuratan dalam menentukan kondisi pada gudang penyimpanan roti.
2. Diperlukannya sebuah sistem penunjang keputusan dalam menentukan lokasi gudang penyimpanan roti.

1.3 Tujuan

Tujuan dari penelitian ini adalah membangun sistem penunjang keputusan berbasis Internet of Things dalam menentukan lokasi gudang penyimpanan roti dan memberikan data yang akurat terkait kondisi gudang penyimpanan roti.

1.4 Manfaaat

Manfaat yang dari penelitian ini antara lain:

1. Membantu pengguna dalam meninjau/memonitor kondisi gudang penyimpanan roti.
2. Hasil dari sistem ini memberikan kemudahan bagi pihak Roti Kuro Subang untuk menentukan lokasi gudang penyimpanan roti dengan tingkat akurasi yang tinggi.

1.5 Metodologi Penelitian

Metode penelitian yang digunakan penulis sebagai berikut:

1. Observasi (studi lapangan)
Mempelajari dan mengumpulkan data berupa data yang dihasilkan oleh sensor kemudian diolah menggunakan metode Weigthed Product sehingga menghasilkan peringkat tiap lokasi yang diuji, metode ini akan digunakan untuk proses Sistem Pendukung Keputusan
2. Studi Literatur
Ditahap ini melakukan literatur (jurnal, buku, dan artikel) mengenai metode penelitian yang meliputi konsep Internet of Things dan metode *Weighted Product*.
3. Dokumentasi
Metode ini dilakukan dengan cara mencari dokumen-dokumen tertentu melalui website, kantor, dan lain-lain.
4. Analisa dan Perancangan Sistem
UML telah umum didefinisikan sebagai bahasa standar untuk menentukan, membangun, memvisualisasikan, dan mendokumentasikan sistem perangkat lunak dari awal. Agar dapat sesuai penggunaan dari rancangan pengembangan sistem, UML menyediakan kesamaan bahasa untuk menggambarkan model perangkat lunak. UML tidak dikonseptual harus dalam metodologi yang khusus, tetapi lebih fleksibel dan dapat disesuaikan agar sesuai dengan pendekatan apa pun dan dapat digunakan secara umum dalam berbagai siklus hidup perangkat lunak dan proses pengembangan[3].
5. Uji coba dan Evaluasi Sistem yang dibuat.
Pada tahap ini sistem yang telah dibuat ini akan dilakukan beberapa skenario.

2. Pustaka

2.1 Internet of Things (IoT)

Internet of Things (IoT) adalah tentang bagaimana benda fisik berinteraksi satu sama lain, komunikasi mesin-ke-mesin dan komunikasi orang-ke-komputer akan diperluas menjadi sesuatu objek. Kunci dari teknologi tersebut akan mendorong IoT berkaitan dengan teknologi sensor pintar termasuk wireless sensor network, teknologi nano dan miniaturisasi[4]. Tujuan dasar dari IoT adalah untuk menghubungkan sesuatu yang tidak terhubung. Ini berarti bahwa objek yang saat ini tidak bergabung ke jaringan komputer, yaitu Internet, akan terhubung sehingga mereka dapat berkomunikasi dan berinteraksi dengan manusia dan objek lainnya. IoT adalah transisi teknologi dimana perangkat akan memungkinkan kita untuk merasakan dan mengendalikan dunia fisik dengan membuat objek lebih pintar dan menghubungkannya melalui jaringan internet. Ketika objek dan mesin dapat dirasakan dan dikendalikan dari jarak jauh melalui jaringan, integrasi yang lebih erat antara dunia fisik dan komputer dapat diaktifkan. Ini memungkinkan peningkatan pada bidang efisiensi, akurasi, otomatisasi dan pengembangan aplikasi yang lebih canggih[5].

Awal mula sejarah IoT dimulai pada tahun 2008 ditandai dengan sebuah forum bernama FCC menyetujui penggunaan white space spectrum. Pada akhirnya peluncuran IPv6 di tahun 2011 memicu pertumbuhan besar di bidang Internet of Things, perkembangan ini didukung oleh perusahaan raksasa seperti Cisco, IBM, Ericsson mengambil inisiatif lebih banyak dari sisi pendidikan dan komersial dengan teknologi IoT yang dapat dikembangkan sebagai penghubung antara manusia dan komputer. Dengan berkembangnya Internet of Things, semua peralatan yang kita gunakan dalam kehidupan kita sehari-hari dapat dikendalikan dan dipantau menggunakan IoT. Mayoritas proses dilakukan dengan bantuan sensor pada IoT. Sensor dikerahkan dimana-mana dan sensor ini merubah data fisik mentah menjadi sinyal digital dan mengirimkan mereka ke pusat kontrol. Dengan cara ini kita bisa memonitor perubahan lingkungan jarak jauh dari setiap bagian dari seluruh dunia melalui internet. Arsitektur sistem ini akan didasarkan pada konteks operasi dan proses dalam skenario real-time. Semisal otomatisasi di rumah dengan setiap kotak saklar listrik akan terhubung dengan ponsel pintar (atau terkadang remote)

sehingga dapat dioperasikan dari jarak jauh. Tapi skenario seperti itu tidak perlu prosesor dan perangkat penyimpanan dipasang di setiap kotak saklar. Suresh, hanya dibutuhkan sensor untuk menangkap sinyal dan proses itu (kebanyakan untuk peralihan ON / OFF). Sehingga arsitektur sistem ini bervariasi tergantung pada konteks penerapannya [6].

2.2 Weighting Product

Yoon, metode Weighted Product menggunakan perkalian untuk menghubungkan peringkat atribut, dimana peringkat dari setiap atribut akan dipangkatkan terlebih dahulu dengan bobot atribut yang bersangkutan [7]. Kusumadewi, Weighted Product adalah keputusan analisis multi-kriteria yang populer dan merupakan metode pengambilan keputusan multi-kriteria. Seperti semua metode FMADM. Metode FMADM untuk menyelesaikan kasus-kasus dimana data terdiri atas banyak atribut kepentingan terdiri dari metode Simple Additive Weighting Method (SAW), Weighted Product (WP), ELECTRE, TOPSIS, dan Analytic Hierarchy Process (AHP) [8]. WP adalah himpunan berhingga dari alternatif keputusan yang yang dijelaskan dalam istilah beberapa kriteria keputusan Ningrum [9].

Pemilihan metode Weighted Product didasarkan juga atas kemampuannya dalam memberikan solusi optimal dalam sistem pemeringkatan. Pemilihan metode ini juga didasarkan atas kompleksitas komputasi yang tidak terlalu sulit sehingga waktu yang dibutuhkan dalam menghasilkan perhitungan relatif singkat [10].

Weighted Product [2] yaitu:

- Penentuan atau perbaikan nilai bobot (W) dengan menggunakan Rumus 2.1 sebagai berikut: Formula untuk melakukan normalisasi adalah sebagai berikut:

Rumus 2.1 Perbaikan bobot

Keterangan:

- W_j adalah indeks bobot.
 - \mathbf{W}_j adalah bobot yang diperbaiki.
 - $\sum \mathbf{W}_j$ adalah jumlah bobot keseluruhan.

Penentuan nilai alternatif dari setiap kriteria (S) yang merupakan pemangkatan atribut penilaian dengan menggunakan Rumus 2.2 sebagai berikut:

Rumus 2.2 Nilai alternatif

Keterangan:

- S_i adalah nilai alternatif ke i
 - W_j adalah bobot yang sudah diperbaiki
 - X_{ij} adalah nilai atribut ke i.

- 2 Penentuan nilai Vektor (V) yang merupakan proses akhir untuk mendapatkan hasil perhitungan dan preferensi pemeringkatan dengan menggunakan Rumus 2.3 sebagai berikut:

$$V_i = \frac{\prod_{j=1}^n X_{ij} W_j}{\prod_{j=1}^n (X_{ij}^*) W_j} \dots \quad \dots \quad 2.3$$

Rumus 2.3 Nilai Vektor

Keterangan:

- V_i adalah nilai vektor ke i.
 - $\prod_j^n = 1 X_{ij} W_j$ adalah nilai alternatif ke i.
 - $\prod_j^n = 1 (X_{ij} *) W_j$ adalah jumlah nilai alternatif pangkat atribut.

2.3 Arduino IDE

Arduino IDE adalah sebuah *software* lingkungan pengembangan yang terintegrasi yang digunakan untuk melakukan pengembangan program pada Arduino. *Software* ini memiliki bahasa pemrograman sendiri yang merupakan turunan dari bahasa C/C++ sehingga terdapat kemiripan dalam penulisan sintaknya. Arduino IDE dikembangkan menggunakan bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library* C/C++ yang biasa disebut *wiring* yang memudahkan dalam operasi *input* dan *output*. Arduino IDE merupakan pengembangan dari *Software Processing* yang telah mengalami perombakan dan menjadi Arduino IDE yang khusus untuk pemrograman dengan Arduino.

Program yang ditulis dengan menggunakan Arduino IDE disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam berkas dengan ekstensi .ino. Teks editor pada *software* Arduino memiliki fitur-fitur seperti *cutting/paste* dan *searching/replacing* sehingga memudahkan dalam menulis kode program [11]. Di dalam *software* Arduino IDE terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program.

2.4 MQTT

MQTT adalah protokol konektivitas mesin-ke-mesin (M2M) / "Internet of Things". Protokol ini dirancang sebagai pengiriman, mempublikasikan dan berlangganan pesan yang sangat ringan. Protokol ini sangat berguna untuk koneksi pada lokasi terpencil dengan pengiriman data yang kecil dimana membangun jaringan *bandwidth* akan sangat mahal sehingga protokol ini dapat menghemat biaya. MQTT telah diaplikasikan dalam berbagai bentuk penggunaan misalnya dalam sensor yang berkomunikasi dengan broker melalui tautan satelit atau melalui koneksi *dial-up*, menyambungkan antara alat penyedia layanan kesehatan, dalam berbagai alat otomatisasi rumah dan skenario perangkat kecil. Protokol ini juga ideal untuk aplikasi seluler karena ukurannya yang kecil, penggunaan daya yang rendah, paket data yang diminimalkan dan distribusi informasi yang efisien ke satu atau banyak penerima [12].

Pesan aplikasi diangkut oleh MQTT akan memiliki sebuah *Quality of Service* dan sebuah *Topic Name* yang terkait [12].

Komponen-komponen yang terdapat dalam MQTT adalah sebagai berikut:

1. Klien adalah program atau perangkat yang menggunakan MQTT. klien selalu dapat membuat koneksi jaringan ke server. Beberapa fitur yang dimiliki klien adalah:
 - Dapat mempublikasikan pesan aplikasi yang mungkin diminati oleh klien lain.
 - Berlangganan untuk meminta pesan aplikasi yang ingin diterima.
 - Berhenti berlangganan untuk menghapus permintaan untuk pesan aplikasi.
 - Memutuskan sambungan dari server.
2. Server adalah program atau perangkat yang bertindak sebagai perantara antara klien yang menerbitkan pesan aplikasi dengan klien yang telah membuat langganan. Beberapa fitur yang dimiliki Server adalah:
 - Server menerima koneksi jaringan dari klien.
 - Menerima pesan aplikasi yang diterbitkan oleh klien.
 - Memproses permintaan berlangganan dan berhenti berlangganan dari klien.
 - Meneruskan pesan aplikasi yang cocok dengan klien yang berlangganan.
3. Berlangganan adalah komponen yang terdiri dari *topic filter* dan *QoS* maksimum. Berlangganan dikaitkan dengan satu sesi. Setiap sesi dapat berisi lebih dari satu langganan. Setiap langganan dalam satu sesi memiliki *topic filter* yang berbeda.
4. *Topic Name* adalah label yang dilampirkan ke pesan aplikasi yang cocok dengan yang berlangganan yang diketahui ke server. Server mengirimkan salinan pesan aplikasi ke setiap klien yang berlangganan.
5. Topic Filter adalah ekspresi yang terkandung dalam berlangganan, untuk menunjukkan minat pada satu atau lebih topik. Topic Filter dapat menyertakan karakter *wildcard*.
6. Session adalah komponen yang berfungsi interaksi *stateful* antara klien dan server. Beberapa *session* hanya berlangsung selama terkoneksi jaringan, yang lain dapat merentang beberapa koneksi jaringan berturut-turut antara klien dan server.

7. Paket Kontrol MQTT adalah paket informasi yang dikirim melalui koneksi jaringan. Spesifikasi MQTT mendefinisikan empat belas jenis paket kontrol yang berbeda, salah satunya (paket *publish*) digunakan untuk menyampaikan pesan aplikasi.

2.5 Node-red

Node-red (<https://nodered.org>) adalah perangkat *Flow-based Programming* yang dibangun berbasis Node.Js, umumnya digunakan di ruang lingkup IoT dan bertujuan untuk menciptakan aplikasi yang digerakkan oleh peristiwa yang dapat dengan mudah mengintegrasikan API dan layanan. Node-red terdiri dari *runtime* berbasis Node.Js yang dapat diakses pada *web browser* untuk mengakses editor aliran. Di dalam *web browser* dapat membuat aplikasi dengan menyeret node dari palet ke ruang kerja dan mulai menyambungkannya. Node-red dikembangkan pada awal 2013 sebagai proyek sampingan oleh Nick O'Leary dan Dave Conway-Jones dari IBM's Emerging Technology Services group. Dimulai sebagai sebuah konsep untuk memvisualisasikan dan memanipulasi pemetaan antara topik MQTT, dengan cepat menjadi alat yang jauh lebih umum yang dapat dengan mudah diperluas ke segala arah. Menjadi proyek *open-source* pada September 2013 dan telah dikembangkan menjadi lebih terbuka sejak itu, sebagai puncaknya menjadi salah satu proyek unggulan dari JS Foundation pada Oktober 2016 [13].

Flow-based Programming diciptakan oleh J. Paul Morrison (www.jpaulmorrison.com/fbp) pada tahun 1970-an, *Flow-based Programming* bertujuan menggambarkan perilaku aplikasi sebagai simpul jaringan, atau *node* sebagaimana mereka disebut dalam Node-red. Setiap *node* memiliki tujuan yang jelas, yaitu menerima data, memproses data dan kemudian meneruskan data itu. Sementara jaringan bertanggung jawab atas aliran data antar *node*. *Flow-based Programming* adalah suatu model yang cocok untuk merepresentasikan secara visual dan membuatnya lebih mudah diakses oleh pengguna yang pada umumnya. Jika seseorang dapat memecah masalah menjadi langkah-langkah terpisah, mereka dapat melihat aliran dan memahami apa yang dilakukannya, tanpa harus memahami setiap baris kode dalam setiap *node* [14].

3. Analisis

3.1 Deskripsi Sistem

Sistem yang bangun adalah sistem pendukung keputusan dengan melakukan pemantauan kondisi gudang menggunakan Mikrokontroler NodeMCU ESP8266. Antarmuka sistem dengan pengguna akan berbasis platform web Node-red. Dalam sistem ini akan dirancang dengan dikelompokkan ke dalam dua bagian, yaitu bagian server dan bagian perangkat IoT.

Bagian server terdiri dari perangkat utama yaitu komputer server dan perangkat IoT berupa Mikrokontroler NodeMCU ESP8266 dengan komponen pendukungnya. Perangkat komputer server berfungsi untuk menyimpan data dihasilkan oleh perangkat IoT dan program web server. Program web server yang digunakan terdiri dari platform Node-red yang berbasis server Node.JS dan basis data MySQL. Konektivitas antara komputer server dan Mikrokontroler NodeMCU ESP8266 menggunakan *Wi-fi Access Point*.

Program web yang akan dibangun ini merupakan sistem yang dapat memonitor dan menyimpan data kondisi lokasi gudang roti berupa variabel suhu, kelembaban dan kualitas udara. Hasil yang dikeluarkan berupa pendukung keputusan dalam menentukan lokasi gudang baru milik Roti Kuro Subang berdasarkan hasil dari analisis yang telah dilakukan menggunakan metode *Weighted Product* dengan data yang telah dikumpulkan oleh perangkat IoT.

3.2 Metode Simple Additive Weighting

Metode penelitian dilakukan dengan studi literatur terhadap beberapa sumber yang relevan, dengan menganalisis terhadap kondisi lokasi gudang. Implementasi metode WP dalam sistem adalah untuk menunjang pendukung keputusan dalam menentukan lokasi gudang lokasi penyimpanan roti yang baru.

Data yang digunakan adalah data lokasi yang telah tentukan sebagai kemungkinan lokasi gudang penyimpanan roti yang baru, yaitu: Pasirkarembi, Sukajadi dan Soklat. Implementasi WP dilakukan dengan cara dibutuhkan beberapa kriteria dalam menentukan peringkat tiap lokasi. kriteria yang ditentukan adalah suhu, kelembaban dan kualitas udara (kadar CO2). Adapun data

hubungan antara lokasi dan kriteria dapat terlihat pada Tabel 3.1. Nilai-nilai tersebut merupakan nilai yang telah dirata-ratakan berdasar hasil dari pemantauan pada masing-masing lokasi Gudang.

Tabel 3.1 Data nilai kriteria.

Lokasi	Suhu	Kelembaban	Kualitas udara
Pasirkarembi	30.73	63.10	1818.15
Sukajadi	31.98	60.22	909.15
Soklat	30.44	66.98	249.53

Setelah mendapatkan data tahapan selanjutnya adalah menentukan bobot. Bobot adalah nilai atau tingkat kepentingan relatif dari setiap kriteria yang diberikan oleh *decision maker*. Adapun bobot dengan perbaikan bobot dari masing-masing kriteria ditampilkan pada Tabel 3.2.

Tabel 3.2 Bobot Kriteria

Kriteria	Bobot	Perbaikan Bobot
Suhu	35%	0.35
Kelembaban	30%	0.30
Kualitas udara	25%	0.25
Jumlah	100%	1

Selanjutnya secara manual, untuk proses nilai alternatif masing-masing lokasi yang dilakukan dengan pangkatkan dan kalikan nilai masing-masing kriteria tersebut dengan bobot yang sudah diperbaiki sebelumnya dengan hasil dapat dilihat pada Tabel 3.3:

Tabel 3.3 Nilai alternatif

Lokasi	Rumus	Hasil
Pasirkarembi	$S_1 = (30.73^{0.35})(63.10^{0.30})(1818.15^{0.25})$	75.095
Sukajadi	$S_1 = (31.98^{0.35})(60.22^{0.30})(909.15^{0.25})$	63.143
Soklat	$S_1 = (30.44^{0.35})(66.98^{0.30})(249.53^{0.25})$	46.378
Jumlah		184.616

Menentukan Nilai vektor yang akan digunakan Menghitung Preferensi (V_i) untuk pemeringkatan dengan rumus berikut:

Tabel 3.4 Nilai Vektor

Lokasi	Rumus	Hasil
Pasirkarembi	$V_1 = \frac{75.095}{75.095 + 63.143 + 46.378}$	0.406
Sukajadi	$V_1 = \frac{63.143}{75.095 + 63.143 + 46.378}$	0.342
Soklat	$V_1 = \frac{46.378}{75.095 + 63.143 + 46.378}$	0.251

Maka hasil dapat dilihat pada Tabel 3.4 lokasi yang memiliki nilai vektor terbesar hingga terkecil secara berurutan adalah Pasirkarembi, Sukajadi dan Soklat

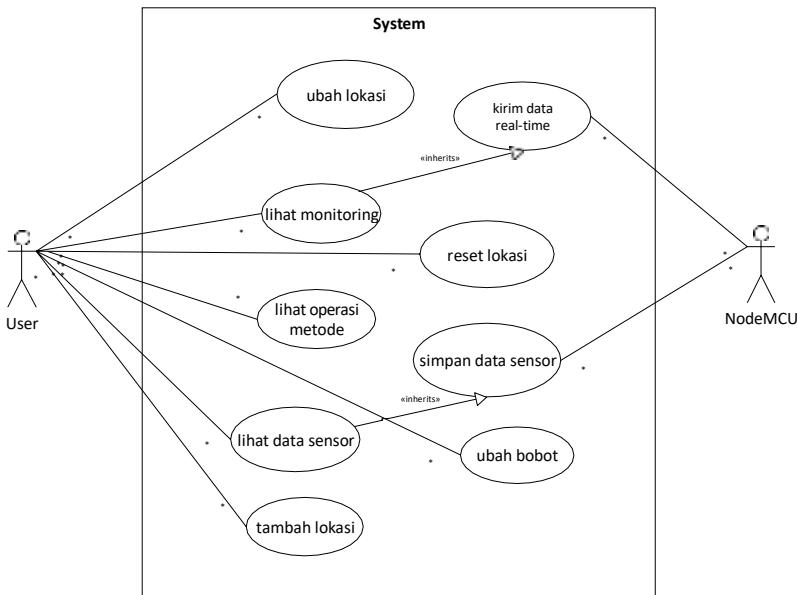
3.3 Functional Requirements

Functional Requirement digunakan untuk mendefinisikan fungsi-fungsi yang dapat diproses pada sistem pendukung keputusan untuk menentukan gudang penyimpanan roti. Sistem dapat melakukan hal berikut:

1. Melihat data *real-time*, data sensor dan operasi metode WP.
2. Menyimpan data sensor dan lokasi.
3. Merubah data bobot.
4. Menghitung data menggunakan metode WP.

3.4 Use Case Diagram

Use case diagram digunakan dalam menggambarkan fungsionalitas dari sebuah sistem. Sebuah diagram *use case* akan mengidentifikasi aktor yang terlibat dalam kegiatan yang dapat dilakukan pada sistem. Dalam hal ini meliputi pengguna dan NodeMCU seperti dapat dilihat pada Gambar 3.1.



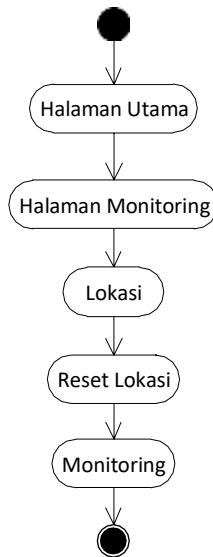
Gambar 3.1 Use case Diagram.

Dari *use case diagram* pada Gambar 3.1 dapat dilihat bahwa sistem ini terdapat dua aktor yaitu pengguna dan NodeMCU, meskipun NodeMCU merupakan perangkat IoT namun pada sistem ini dapat berkomunikasi dengan objek dimana aktor digambarkan dengan gambar orang. pengguna dan NodeMCU dapat melakukan beberapa hal yang digambarkan dengan *use case* (oval).

3.5 Activity Diagram

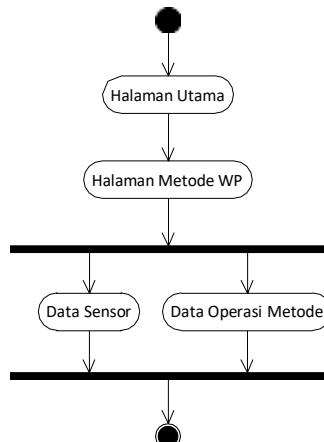
Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang dirancang sesuai dengan metode yang digunakan, maka langkah-langkah yang dilakukan dalam *Activity Diagram* adalah sebagai berikut:

1. Menu Monitoring



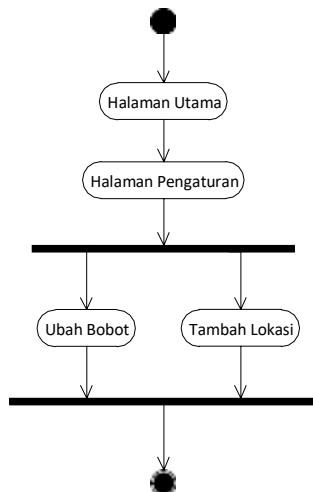
Gambar 3.2 Activity Diagram Menu Monitoring.

2. Menu Metode WP



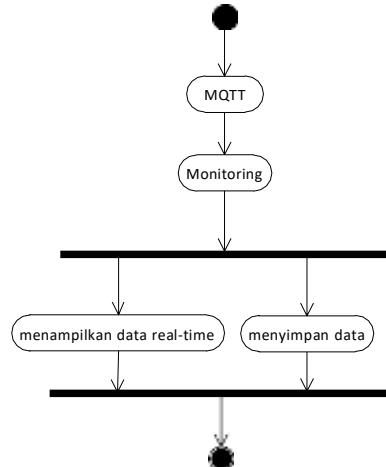
Gambar 3.3 Activity Diagram Menu Metode WP.

3. Menu Pengaturan



Gambar 3.4 Activity Diagram Menu Monitoring.

4. Koneksi IoT

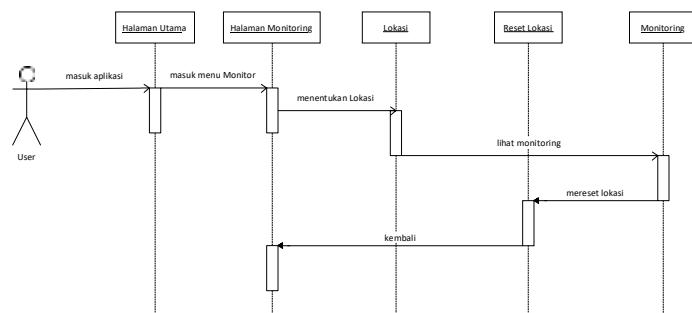


Gambar 3.51 Activity Diagram Koneksi IoT.

3.6 Sequence Diagram

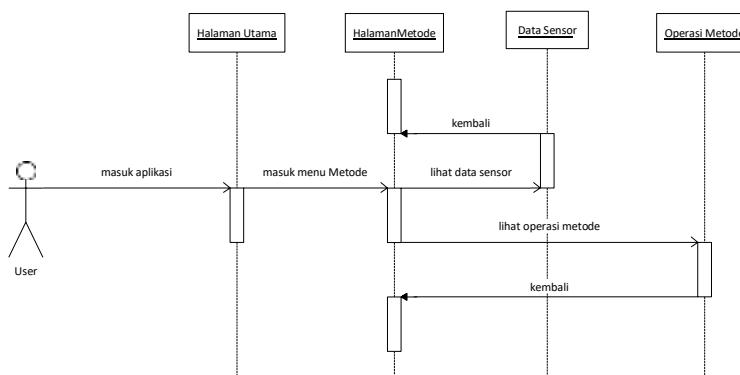
Sequence diagram digunakan untuk menggambarkan urutan-urutan kejadian yang mungkin terjadi pada sistem. Berikut adalah *sequence diagram* dari sistem pendukung keputusan untuk menentukan gudang penyimpanan roti.

1. Menu Monitoring



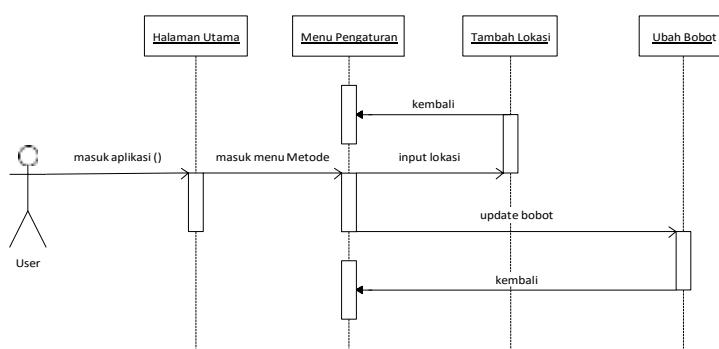
Gambar 3.6 Sequence Diagram Menu Monitoring.

2. Menu Metode WP



Gambar 3.7 Sequence Diagram Menu Metode WP.

3. Menu Pengaturan

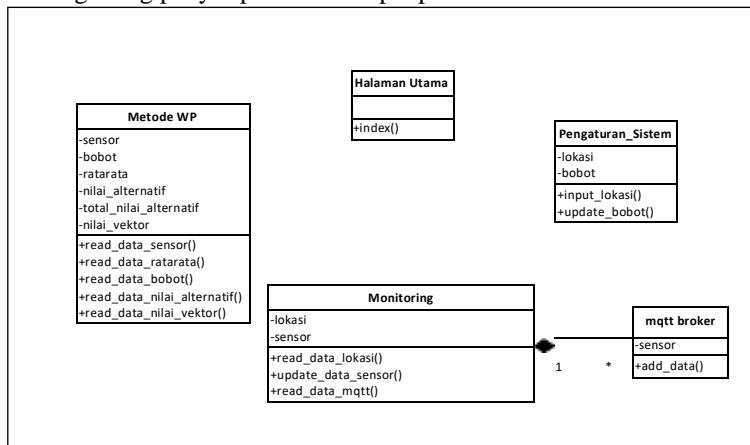


Gambar 3.8 Sequence Diagram Menu Pengaturan.

3.7 Class Diagram

Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas yang ada dalam sistem perangkat lunak yang akan dikembangkan. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk

mencapai suatu tujuan. Berikut penggambaran *class diagram* dari sistem pendukung keputusan untuk menentukan gudang penyimpanan roti dapat pada Gambar 3.9.



Gambar 3.92 Class Diagram

4. Implementasi

4.1 Node-Red Platform

Instalasi Node-red. Node-red merupakan *framework* atau perangkat kerja berupa aplikasi berbasis web yang dirancang untuk lingkungan pengembangan konsep IoT dimana di dalamnya memungkinkan pengguna berinteraksi dengan perangkat IoT. Instalasi Node-red dapat diawali membuka Command Prompt kemudian mengetik:

```
npm install -g --unsafe-perm node-red
```

Selanjutnya tahap proses instalasi akan selesai dengan menampilkan pesan *succes*, seperti terlihat pada Gambar 4.1.

```

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Tester>node --version && npm --version
v0.16.2
6.9.0

C:\Users\Tester>npm install -g --unsafe-perm node-red
C:\Users\Tester>AppData\Roaming\npm\node-red-> C:\Users\Tester\AppData\Roaming\npm\node_modules\node-red\binnode-red-pi
C:\Users\Tester>AppData\Roaming\npm\node-red-> C:\Users\Tester\AppData\Roaming\npm\node_modules\node-red\node-red.js

> bcrypt@3.0.6 install C:\Users\Tester\AppData\Roaming\npm\node_modules\node-red\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build
node-pre-gyp [ERR] Being request for node-pre-gyp https download
[bcrypt] Success: "C:\Users\Tester\AppData\Roaming\npm\node_modules\node-red\node_modules\bcrypt\lib.binding.lib.node" is installed via remote
+ node-red@0.20.7
added 368 packages from 352 contributors in 87.316s

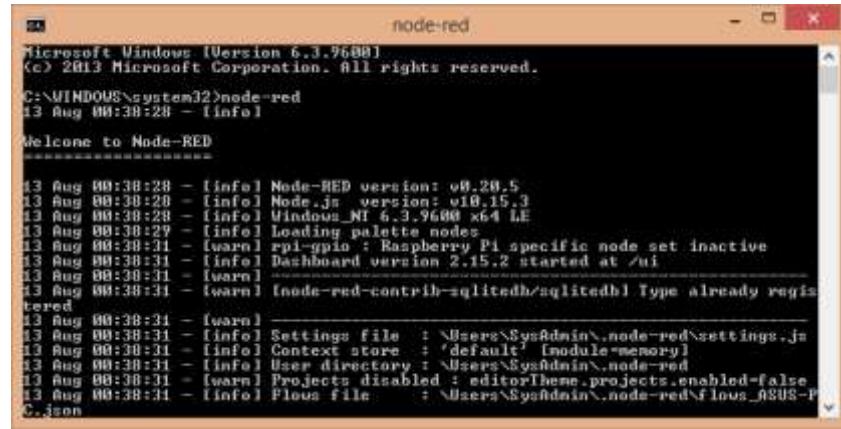
C:\Users\Tester>
  
```

Gambar 4.1 Instalasi Node-red.

Untuk menjalankan node-red dapat dilakukan dengan cara membuka Command Prompt kembali dan mengetik:

```
node-red
```

Kemudian akan tampil log yang menampilkan bahwa Node-red telah berjalan, dan dapat diakses pada halaman <http://localhost:1880/> di *web browser* seperti ditampilkan pada Gambar 4.2 dan Gambar 4.3.

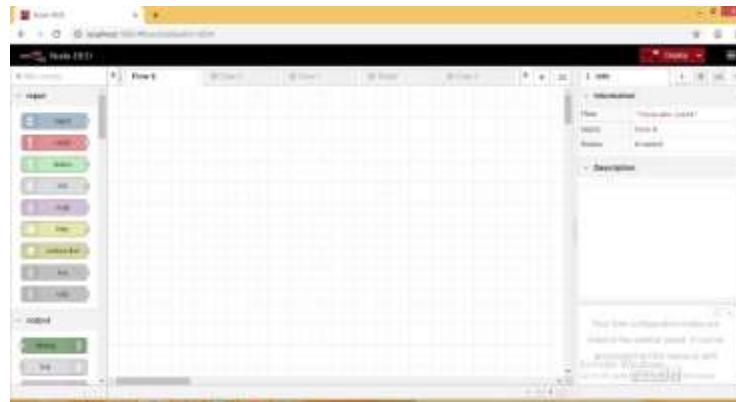


```
Microsoft Windows [Version 6.1.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>node-red
13 Aug 00:38:28 - [info] Welcome to Node-RED

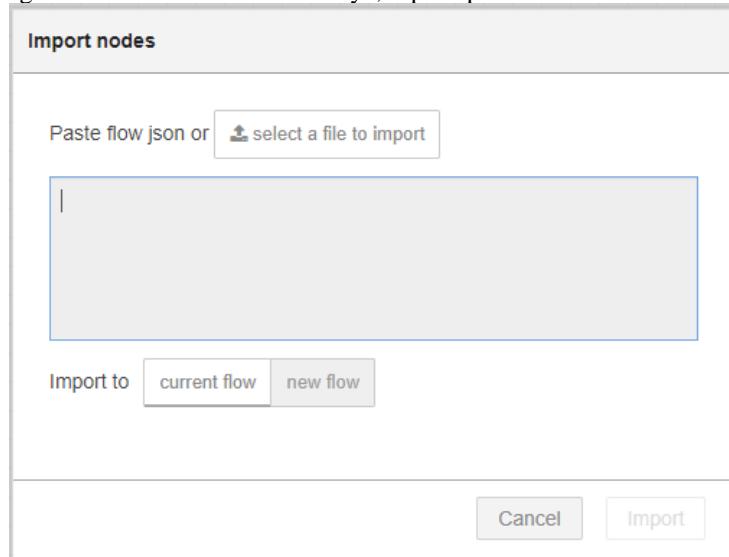
13 Aug 00:38:28 - [info] Node.js version: v10.15.3
13 Aug 00:38:28 - [info] Windows_NT 6.1.9600 x64 LE
13 Aug 00:38:29 - [info] Loading palette nodes
13 Aug 00:38:31 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
13 Aug 00:38:31 - [info] Dashboard version 2.15.2 started at /ui
13 Aug 00:38:31 - [warn] 
13 Aug 00:38:31 - [warn] Inode-red-contrib-sqlitedb/sqlitedb Type already registered
13 Aug 00:38:31 - [warn] 
13 Aug 00:38:31 - [info] Settings file : C:\Users\SysAdmin\.node-red\settings.js
13 Aug 00:38:31 - [info] Context store : 'default' [module=memory]
13 Aug 00:38:31 - [info] User directory : C:\Users\SysAdmin\.node-red
13 Aug 00:38:31 - [warn] Projects disabled : editorTheme.projects.enabled=false
13 Aug 00:38:31 - [info] Flows file : C:\Users\SystemAdmin\.node-red\flows_ASUS-P
C.json
```

Gambar 4.2 Node-red pada Command Prompt.



Gambar 4.3 Node-red pada web browser.

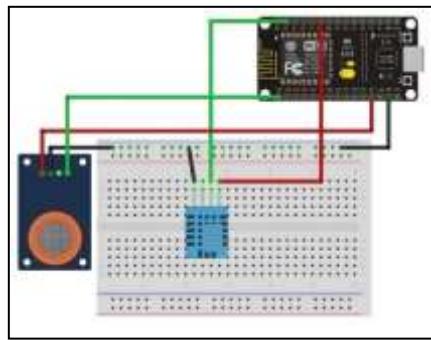
Selanjutnya adalah mengatur isi *node* pada Node-red, yaitu dengan cara mengimpor *flow* yang telah diprogram terlebih dahulu sebelumnya, seperti pada Gambar 4.4.



Gambar 4.4 Impor flow pada Node-red.

4.2 Arsitektur Hardware

Rancangan rangkaian elektronik adalah alat untuk perancangan yang menggambarkan secara visual hubungan antar komponen elektronik digunakan dalam perakitan peralatan IoT dengan spesifikasi yang telah dijelaskan pada analisis kebutuhan, seperti dapat dilihat pada Gambar 4.5.



Gambar 4.5 Rancangan rangkaian elektronik

4.3 Perangkaian Perangkat IoT dan Jaringan Access Point

Selain satu unit komputer penulis juga menggunakan perangkat IoT guna menunjang penelitian, berikut komponen perangkat IoT yang terdiri dari:

1. NodeMCU ESP8266
2. DHT11
3. MQ-135

Pada tahap ini dilaksanakan implementasi dari rancangan rangkaian elektronik kemudian melakukan pemrograman ke mikrokontroler menggunakan Arduino IDE. Berikut adalah hasil dari implementasi dapat dilihat pada Gambar 4.13 dan Gambar 4.14

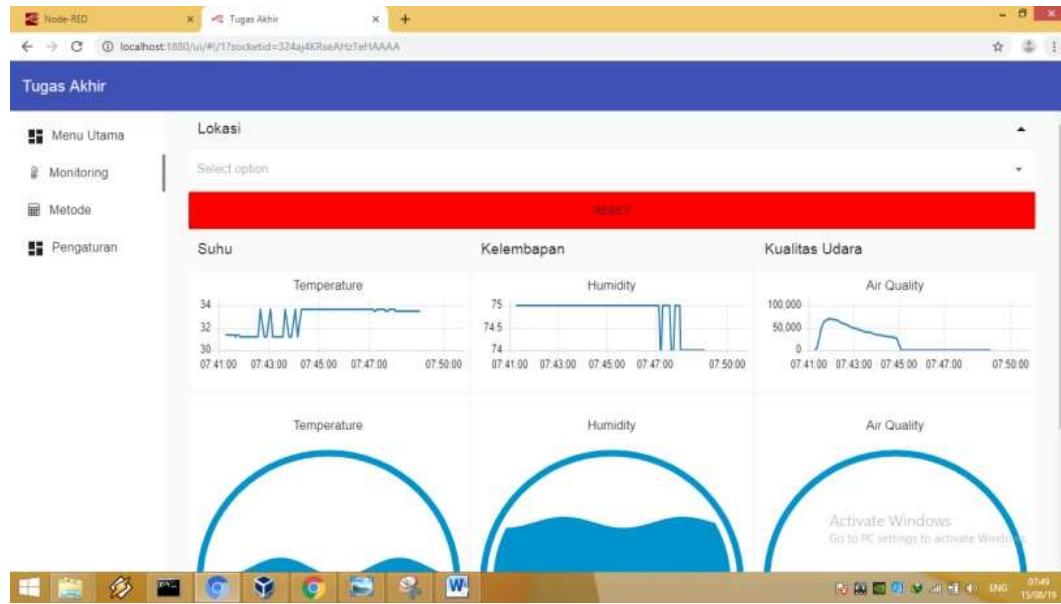


Gambar 4.6 Rangkaian perangkat IoT.

Untuk menyambungkan perangkat IoT dengan komputer server maka diperlukan sebuah perangkat *Wifi Access Point*, dalam hal ini penulis menggunakan media berupa *tethering* dari perangkat *Android Smartphone*.

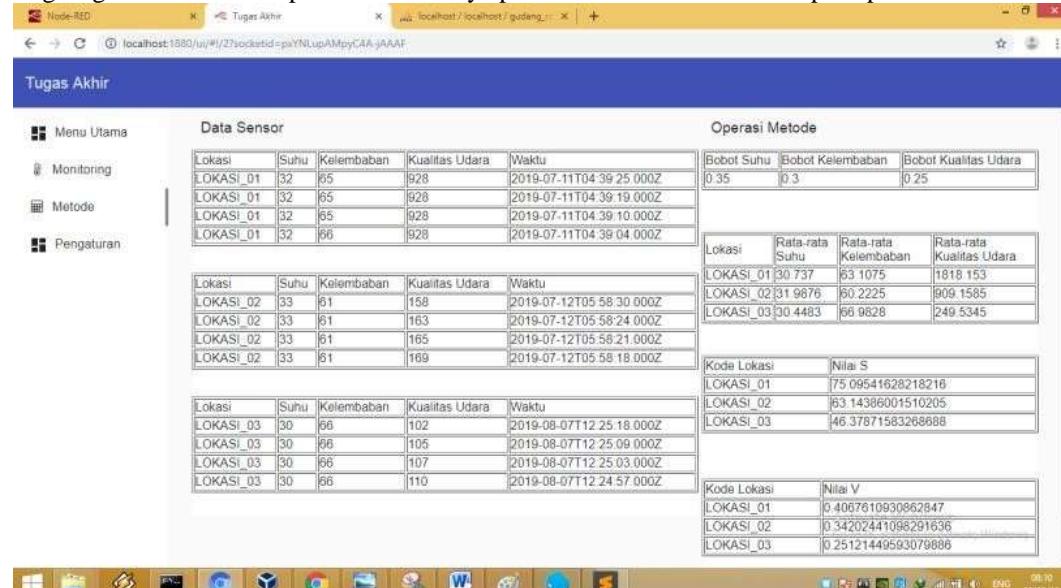
4.4 Pengujian Sistem

Pada tahap ini pengujian sistem dilakukan dengan menentukan lokasi gudang pada halaman Monitoring sebelum dilakukan pemantauan pada lokasi gudang penyimpanan roti yang akan ditinjau dan tekan tombol *reset* jika hendak berhenti memasukan data kondisi lokasi gudang roti ke dalam basis data seperti pada Gambar 4.8.



Gambar 4.8 Pengujian pada pemantauan lokasi

Setelah melakukan pemantauan pada lokasi gudang penyimpanan roti, selanjutnya data akan langsung dianalisis dan dapat dilihat hasilnya pada halaman Metode WP seperti pada Gambar 4.9.



Gambar 4.9 Pengujian pada metode WP

Dari hasil analisis tersebut diketahui bahwa perbandingan antara tiap-tiap lokasi gudang seperti ditampilkan pada Gambar 4.9. Dapat disimpulkan bahwa lokasi gudang penyimpanan roti yang direkomendasikan oleh sistem adalah lokasi 01 dengan nilai vektor tertinggi yaitu 0.406761093.

5. Kesimpulan

5.1 Hasil

Berdasarkan hasil penulisan uraikan, maka dapat ditarik kesimpulan sebagai berikut:

1. Sistem yang telah dibuat dapat menunjang keputusan dalam menentukan lokasi gudang penyimpanan roti untuk Roti Kuro Subang berdasarkan peringkat masing-masing lokasi.
2. Sistem yang telah dibuat dapat mengumpulkan dan menampilkan data secara *real-time*.

3. Sistem yang telah dibuat telah menunjukkan bahwa konsep IoT dapat diterapkan dengan berbagai konteks dalam hal ini penunjang pengambilan keputusan.
4. Dari pengujian sistem diketahui bahwa hasil pengambilan keputusan dari menentukan lokasi gudang Roti Kuro Subang dengan menggunakan metode *Weighted Product* berbasis IoT adalah lokasi 01 dengan nilai vektor tertinggi yaitu 0.406761093, sehingga lokasi gudang tersebut direkomendasikan oleh sistem.

5.2 Saran

Saran untuk mengembangkan sistem ini antara lain:

1. Ditambahkan tombol *on/off* pada perangkat IoT.
2. Dapat menambahkan kriteria penilaian sesuai kebutuhan sistem.
3. Ditambahkan tingkat galat atau *error* dalam pengambilan data nilai kriteria.
4. Menggunakan metode lain sebagai pembanding untuk menentukan tingkat akurasi sistem

Daftar Pustaka

- [1] D. J. Bowersox and D. J. Closs. *Manajemen Logistik: Integrasi Sistem-Sistem Manajemen Distribusi Fisik dan Manajemen Material* (terjemahan A. Hasymi Ali). Jakarta: Penerbit Bumi Aksara. 1986.
- [2] K. Wahyudin dan Masniah, “*Penerapan Metode Weighted Product Untuk Penilaian Eco Office Award Pada Dinas Lingkungan Hidup Kabupaten – Kota*,” Jutisi : Jurnal Ilmiah Teknik Informatika dan Sistem Informasi Vol 6, No 3, 1655-1666, 2017.
- [3] T. Connolly, and C. Begg, *Database Systems a practical approach to design, implementation, and management* (6 ed.). USA: Pearson Education, 2015
- [4] S. D. Kelly, M. K. Suryadevara, and C. S. Mukhopadhyay, *Towards the Implementation of IoT for Environmental Condition Monitoring in Homes*. IEEE, 2013.
- [5] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, and J. Henry, *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*. Indianapolis: Cisco Press, 2017.
- [6] A. Junaidi, “*Internet Of Things, Sejarah, Teknologi Dan Penerapannya : Review*,” Jurnal Ilmiah Teknologi Informasi Terapan Volume I, No 3, 62-66, 2015.
- [7] Irwansyah, “*Penggunaan Metode Weighting Product (WP) Pada Sistem Pendukung Keputusan Dalam Penentuan Barang Konveksi*,” Just TI Volume 7 No 2, 889-896, 2015.
- [8] S. Lestari, “*Penerapan Metode Weighted Product Model Untuk Seleksi Calon Karyawan*,” Jurnal Sistem Informasi (JSI), VOL. 5, NO. 1, 540-545, 2013.
- [9] Basri, “*Metode Weighted Product (WP) Dalam Sistem Pendukung Keputusan Penerimaan Beasiswa Prestasi*,” Jurnal Insypro (Information System and Processing), vol. 2, no. 1, 2017.
- [10] A. Ahmadi, and D. T. Wiyanti, *Implementasi Weighted Product (WP) dalam Penentuan Penerima Bantuan Langsung Masyarakat PNPM Mandiri Perdesaan*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 19-22, 2014.
- [11] A. Riska, Mengenal Arduino Software IDE, March 16, 2016. Accessed on: July 6, 2019. [Online]. Available: <https://www.sinauarduino.com/artikel/mengenal-arduino-software-ide/>
- [12] M. Thakur, NodeMCU ESP8266 Communication Methods and Protocols: Programming with Arduino IDE. Amazon Digital Services LLC, 2018.
- [13] IoT4Beginners, *What is Node-RED?*, Accessed on: July 17, 2019. [Online]. Available: <https://iot4beginners.com/what-is-node-red/>
- [14] G. S. Barbieri, *Flow-Based Programming Applied to IoT Development, OpenIoT & ELC Europe2, 2016*. Access on: 12 July, 2019. [Online]. Available: <http://events.linuxfoundation.org/sites/events/files/slides/openiot2016-Flow%20Based%20Programming%20Applied%20to%20IoT%20Development.pdf>