

Implementasi Monitoring Kualitas Udara Menggunakan Peramalan Exponential Smoothing dan NodeMCU Berbasis Mobile

by Eko Budi Setiawan

Submission date: 18-Nov-2021 12:40PM (UTC+0700)

Submission ID: 1706343229

File name:

3441_Eko_Budi_Setiawan_Implementasi_Monitoring_Kualitas_Udara_Menggunakan_Peramalan_Exponential_Smoothing_1757499877.pdf
(2.39M)

Word count: 4135

Character count: 24388

Implementasi Monitoring Kualitas Udara Menggunakan Peramalan Exponential Smoothing dan NodeMCU Berbasis Mobile Android

Kurniansyah Pratama¹, Eko Budi Setiawan²
 Program Studi Teknik Informatika, Universitas Komputer Indonesia
kurniansyah.pratama@gmail.com
eko@email.unikom.ac.id

Diterima 23 Oktober 2017
 Disetujui 20 Desember 2017

Abstract—Poor air quality threatens the health of all living things from humans to plants. Most people don't know that the air they are breathing may be polluted. WHO said that 92% population in the world are living in the area with poor air quality that exceed the safe limit from WHO. Carbondioxide is one of many dangerous gas that must be considered by human. It can cause serious problem to human health. Industries are contribute to carbondioxide concentration in air. The purpose of this system is build the tool using NodeMCU microcontroller to measure the air quality in some place. This system can inform the user about air quality in their place by using Android smartphone. The forecast feature can be used to prevent the dangerous air quality.

Index Terms— Carbondioxide, Air Quality, Android, Forecast

I. PENDAHULUAN

Udara merupakan faktor penting dalam kehidupan. Kualitas udara yang buruk dapat mengancam kelangsungan hidup semua makhluk mulai dari manusia, hewan hingga tumbuhan. Udara yang setiap waktu kita hisap mengandung emisi yang berasal dari berbagai sumber seperti emisi yang keluar dari industri, kendaraan bermotor, juga asap rokok. Hasil studi menunjukkan bahwa polusi udara berbahaya bagi kesehatan manusia, khususnya untuk para usia lanjut dan anak – anak. Seperti berita yang dilansir pada situs WHO bahwa 92% populasi di dunia hidup di area yang memiliki kualitas udara yang melampaui batas aman yang telah ditentukan oleh organisasi WHO [1].

Dengan kemajuan teknologi seperti adanya mikrokontroler dan banyaknya pengguna *smartphone*, teknologi ini dapat dimanfaatkan untuk mengawasi kualitas udara di suatu tempat. Mikrokontroler dapat dimanfaatkan sebagai pengolah data dari sebuah sensor untuk mengambil nilai ppm yang dapat dijadikan sebagai acuan kualitas udara dari suatu tempat. Sedangkan *smartphone* dapat digunakan untuk menampilkan data yang telah diperoleh oleh

mikrokontroler supaya kondisi kualitas udara di tempat tersebut dapat diawasi dari jarak jauh.

NodeMCU adalah sebuah perangkat IoT yang bersifat *open source*, berbasis proyek eLua dan dibangun diatas ESP8266SDK 1.4. Alat ini banyak menggunakan proyek – proyek *open source*, seperti lua-cjson dan spiffs. Didalamnya terdapat *firmware* yang berfungsi untuk menjalankan ESP8266 Wi-Fi SoC, dan *hardware* yang berbasis modul ESP-12 [2]. Dengan menggunakan sensor MQ-135 dan perangkat NodeMCU [3], maka dapat dibuat sebuah alat untuk memonitor kualitas udara yang ada di suatu tempat untuk menyelesaikan permasalahan mengenai keadaan kualitas udara buruk yang telah dijelaskan tadi. Namun, untuk lebih baiknya tidak hanya memonitor keadaan yang telah terjadi dan sedang terjadi saja. Maka dari itu diperlukan sebuah metode yang dapat memprediksi keadaan selanjutnya.

Dari referensi penelitian telah ditemukan bahwa peramalan melalui pola data mengenai indeks kualitas udara dapat dilakukan menggunakan metode *Exponential Smoothing* [4]. Cara kerja metode *Exponential Smoothing* yaitu dengan memanfaatkan data-data yang telah diperoleh sebelumnya untuk kemudian dijadikan model yang dapat digunakan untuk peramalan.

Dengan demikian, aplikasi yang akan dibangun dalam penelitian ini diharapkan dapat diterapkan sebagai indikator untuk mengawasi kualitas udara di wilayah yang ditetapkan oleh penggunanya. Sehingga sistem dapat memberikan informasi kepada pengguna mengenai tingkat kualitas udara di wilayah tersebut.

II. TINJAUAN PUSTAKA

A. Polusi Udara

Polusi udara atau pencemaran udara adalah masuknya atau dimasukkannya zat, energi, dan atau komponen lain ke dalam udara oleh kegiatan manusia, sehingga melampaui baku mutu udara yang telah

ditetapkan [5]. Sumber pencemaran udara dapat dibagi menjadi tiga kategori besar, yaitu sumber perkotaan dan industri, sumber pedesaan/pertanian dan lainnya, emisi alami. Umumnya sumber pencemar udara utama dari industri merupakan sumber titik, walaupun di dalam kawasan industri besar akan ditemui pula sumber garis, misalnya jalan penghubung di dalam kawasan tersebut, maupun sumber area. Sumber titik di industri juga dapat diamati sebagai:

1. Emisi normal, yaitu emisi yang berasal dari sumber-sumber yang terkontrol dan disalurkan melalui cerobong sehingga dapat diukur atau dipantau besarnya.
2. Emisi abnormal, yaitu emisi yang berasal dari sumber-sumber titik kecil. Sumber ini lebih sulit dikontrol dan diukur.
3. Emisi sementara/aksidental yang berasal dari kebocoran dan tumpahan kecil, ledakan dan kebakaran.

Emisi normal dan abnormal masih dapat dikelola walaupun pada emisi abnormal tindakannya lebih sulit, sedangkan emisi aksidental diatasi dengan tindakan tanggap darurat. Emisi abnormal berasal dari sumber kecil sebuah proses berupa kebocoran gas atau uap dari sambungan pipa.

B. Karbondioksida (CO₂)

Karbondioksida adalah gas yang tidak berwarna dan tidak berbau. Karbondioksida beracun pada jantung dan menyebabkan menurunnya gaya kontraktile. Pada konsentrasi tiga persen berdasarkan volume di udara, karbondioksida bersifat narkotik ringan dan menyebabkan peningkatan tekanan darah dan denyut nadi, dan menyebabkan penurunan daya dengar. Pada konsentrasi sekitar lima persen berdasarkan volume, karbondioksida menyebabkan stimulasi pusat pernapasan, pusing-pusing, kebingungan dan kesulitan pernapasan yang diikuti sakit kepala dan sesak napas. Pada konsentrasi delapan persen, dapat menyebabkan sakit kepala, keringatan, penglihatan buram, tremor, dan kehilangan kesadaran setelah paparan selama lima sampai sepuluh menit. Konsentrasi karbondioksida (CO₂) global di atmosfer telah meningkat sejak dimulainya revolusi industri karena pertumbuhan pesat aktivitas manusia [6].

Tabel 1. Indeks Kualitas Udara Karbondioksida

| Indeks Kualitas Udara | ppm | Keterangan |
|-----------------------|-------------|--|
| Good | 350 < 1000 | Rata – rata konsentrasi di dalam ruangan dengan sistem sirkulasi udara yang baik |
| Moderate | 1000 < 2000 | Mengakibatkan kantuk |
| Unhealthy | 2000 < 5000 | Sakit kepala, kantuk, penurunan konsentrasi, |

| Indeks Kualitas Udara | ppm | Keterangan |
|-----------------------|----------|--|
| | | peningkatan denyut jantung |
| Very Unhealthy | 5000 | Batas maksimal konsentrasi karbondioksida dalam lingkungan kerja. Untuk jam kerja 8 jam. Menurut OSHA(Occupational Safety and Health Administration) |
| Dangerous | >= 40000 | Dapat menyebabkan kekurangan oksigen serius yang mengakibatkan kerusakan otak permanen, koma bahkan kematian |

C. API (Application Programming Interface)

API (Application Programming Interface) adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal.

Pada konteks web, API merupakan pemanggilan fungsi lewat *Hyper Text Transfer Protocol* (HTTP) dan mendapatkan respon berupa *Extensible Markup Language* (XML) atau *Javascript Object Notation* (JSON). Pemanggilan fungsi ke suatu situs tertentu akan menghasilkan respon yang berbeda. [7]

D. NodeMCU

NodeMCU adalah sebuah perangkat IoT yang bersifat *open source*. Didalamnya telah terdapat *firmware* yang berjalan diatas ESP8266 Wi-Fi SoC dari sistem espress if, dan perangkat keras yang berbasis modul ESP – 12. Nama NodeMCU sendiri mengacu pada *firmware* yang ada didalamnya. *Firmware* perangkat ini menggunakan bahasa pemrograman Lua. Perangkat ini berbasis pada proyek eLua, dan dibangun diatas espress if Non – OS SDK untuk ESP8266. ESP8266 adalah Wi – Fi SoC yang telah diintegrasikan dengan inti Tensilica Xtensa LX106, yang banyak digunakan untuk aplikasi – aplikasi IoT.

Setelah itu, dilakukan *porting library* klien MQTT dari Contiki ke dalam *platform* ESP8266 SoC, yang kemudian ditetapkan untuk proyek NodeMCU. Dengan itu, NodeMCU mempunyai dukungan untuk protokol MQTT IoT, menggunakan Lua untuk mengakses MQTT broker. Perkembangan lainnya yang tidak kalah penting dilakukan pada tanggal 30 Januari 2015, ketika Devsaurus melakukan porting u8glib untuk proyek NodeMCU, memberikan kemudahan pada NodeMCU untuk menggunakan sebuah LCD, OLED, monitor [3].



Gambar 1. NodeMCU V3

E. Android

Android adalah sebuah sistem operasi berbasis kernel linux yang dikembangkan oleh Google dan digunakan untuk perangkat-perangkat *mobile*. Android memungkinkan pengembang untuk membuat *programming* dengan menggunakan bahasa Java, juga dapat mengontrol perangkat melalui *library* Java yang telah dikembangkan oleh Google. Langkah pertama dalam pengembangan aplikasi android, pengembang membuat aplikasi menggunakan bahasa Java, kemudian aplikasi-aplikasi tersebut diunduh melalui situs pihak ketiga atau toko aplikasi *online* [8].

F. Ionic

Ionic adalah sebuah *framework* dapat digunakan untuk membangun sebuah aplikasi *mobile* dengan menggunakan teknologi web seperti HTML5, CSS dan JavaScript. *Framework* ionic dibangun menggunakan AngularJS, yang merupakan sebuah *framework* JavaScript yang telah banyak digunakan. Angular juga menjadikan aplikasi-aplikasi yang dibangun lebih mudah dikelola untuk pengembangan dan bekerja secara *teamwork* karena ionic memungkinkan pengembang untuk dapat dengan mudah menambah, mengkostumisasi fitur-fitur atau *library-library* aplikasi [9].

G. Peramalan (Forecast)

Peramalan adalah suatu proses memperkirakan secara sistematis tentang apa yang mungkin terjadi dimasa yang akan datang berdasarkan informasi masa lalu dan sekarang yang dimiliki agar kesalahannya dapat diperkecil. Peramalan tidak memberikan jawaban pasti tentang apa yang akan terjadi, melainkan berusaha mencari pendekatan tentang apa yang akan terjadi sehingga dapat memberikan kontribusi dalam menentukan keputusan terbaik [10].

H. Metode Pemulusan Eksponensial Musiman

Sebagaimana halnya dengan persamaan pemulusan eksponensial linear yang dapat digunakan untuk memprakirakan serial data yang memiliki pola trend, bentuk persamaan lain dapat digunakan apabila pola dasar serial data berbentuk musiman. Salah satu metode prakiraan yang khusus untuk data yang berpola musiman adalah metode pemulusan eksponensial linear dan musiman dari *winter*. Metode ini didasarkan atas tiga persamaan, yaitu masing – masing untuk unsur stasioner, trend dan musiman, yang dirumuskan seperti pada gambar 2.

$$S_t = \alpha \left(\frac{X_t}{I_{t-L}} \right) + (1 - \alpha)(S_{t-1} + T_{t-1})$$

$$T_t = \beta(S_t - S_{t-1}) + (1 - \beta)T_{t-1}$$

$$I_t = \gamma \left(\frac{X_t}{S_t} \right) + (1 - \gamma)I_{t-L}$$

$$F_{t+m} = (S_t + T_t \cdot m)I_{t-L+m}$$

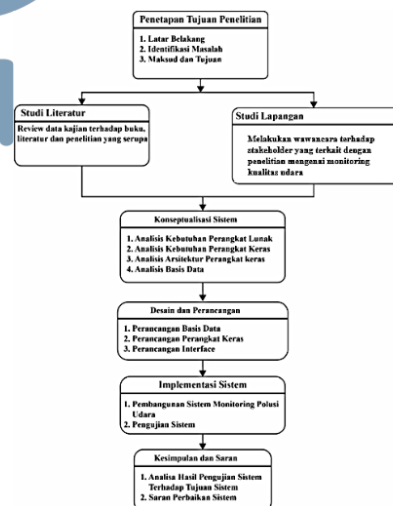
Gambar 2. Persamaan Pemulusan Eksponensial Musiman

Keterangan :

- S_t = Nilai Pemulusan *Level*
- T_t = Nilai Pemulusan *Trend*
- I_t = Nilai Pemulusan Musiman
- F_{t+m} = Prakiraan untuk periode t
- α, β, γ = Konstanta pemulusan
- L = Jumlah periode dalam satu siklus musim
- I = Faktor penyesuaian musim.

III. METODOLOGI PENELITIAN

Metodologi penelitian dalam ilmu komputer/sistem informasi/teknologi informasi merupakan “langkah-langkah/tahapan perencanaan dengan bantuan beberapa metode, teknik, alat (*tools*) dan dokumentasi dengan tujuan untuk membantu peneliti dalam meminimalkan resiko kegagalan dan menekankan pada proses/sasaran penelitian di bidang CS/IS/IT”. Adapun langkah-langkah penelitian yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 3.



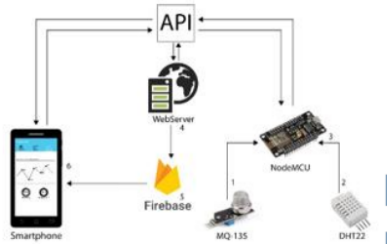
Gambar 3 Tahapan Penelitian

IV. HASIL DAN PEMBAHASAN

Pada tahapan ini terdiri dari tahapan penelitian yang dilakukan. Tahapan tersebut terdiri dari analisis dan perancangan sistem, analisis teknologi yang digunakan.

A. Analisis dan Perancangan Sistem

Sistem yang akan dibangun yaitu aplikasi monitoring kualitas udara yang disertai fitur peramalan untuk Android. Arsitektur sistem yang akan dibangun dapat dilihat pada Gambar 4.



Gambar 4. Arsitektur Sistem

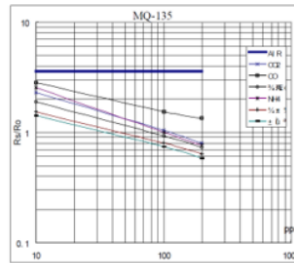
NodeMCU berfungsi sebagai pengolah dan alat untuk mengambil data kandungan CO₂ serta data kelembaban dan temperatur dari work station untuk kemudian dikirim ke webserver melalui API. Setelah data dikirim, data dapat dilihat melalui aplikasi smartphone Android atau web display yang telah disediakan. Aplikasi pada smartphone memiliki fitur-fitur untuk melihat kondisi terbaru dari work station, histori, serta peramalan untuk membuat prakiraan beberapa periode ke depan. Firebase pada sistem akan mengirimkan push notification pada smartphone ketika kualitas udara mencapai kondisi unhealthy.

B. Analisis Teknologi yang digunakan

Analisis teknologi yang digunakan adalah proses analisis mengenai cara kerja teknologi-teknologi yang akan digunakan oleh sistem. Dalam penelitian ini ada beberapa teknologi yang akan dijelaskan yang terdiri dari analisis cara kerja sensor kualitas udara, cara kerja sensor kelembaban dan temperatur DHT22 dan cara kerja analog to digital converter(ADC) yang ada pada mikrokontroler NodeMCU, analisis pengiriman data menggunakan NodeMCU dan analisis API yang akan dibangun.

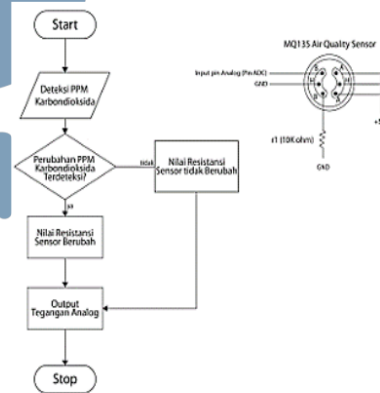
1. Analisis cara kerja sensor MQ - 135

Pada dasarnya, cara kerja dari sensor kualitas udara MQ135 adalah dengan mendeteksi keberadaan gas-gas berbahaya di sekitarnya. Sensor ini akan digunakan sebagai pendeteksi kadar gas CO₂ yang ditunjukkan oleh satuan ppm.



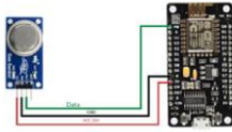
Gambar 5. Karakteristik Sensitivitas sensor MQ135

Dari grafik yang ditunjukkan pada Gambar 5, dapat diketahui bahwa tahanan sensor (Rs/Ro) akan mengecil seiring dengan tingkat konsentrasi gas karbondioksida (CO₂) yang terdeteksi oleh sensor sehingga input tegangan analog yang akan digunakan sebagai input pin ADC pada mikrokontroler NodeMCU akan semakin besar. Sebaliknya, semakin kecil tingkat kepadatan gas yang dapat terdeteksi oleh sistem maka tahanan sensor akan semakin besar sehingga tegangan analog yang masuk ke dalam pin ADC mikrokontroler akan semakin kecil.



Gambar 6. Alur Kerja Sensor MQ135

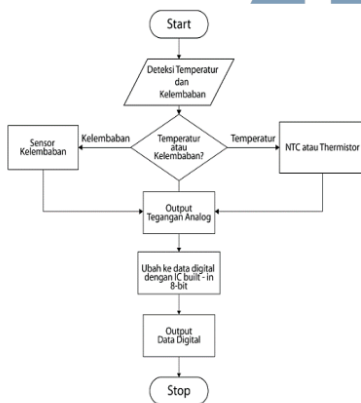
Sensor ini mengambil hasil deteksi kualitas udara berupa perubahan nilai resistansi analog pada pin keluarannya. Pin ini dapat dihubungkan dengan pin ADC (analog-to-digital converter) pada perangkat Mikrokontroler NodeMCU. Dengan demikian, peneliti dapat menyimpan data tersebut pada database server untuk kemudian dapat diolah dan dikirimkan pada perangkat smartphone yang digunakan oleh pengguna. Koneksi antara sensor MQ-135 dengan mikrokontroler NodeMCU dapat dilihat pada Gambar 7.



Gambar 7. Koneksi Sensor MQ135 dan NodeMCU

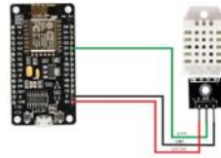
2. Analisis cara kerja sensor DHT22

Sensor DHT22 atau AM2302 adalah sensor kelembaban dan temperatur yang mempunyai tingkat presisi yang sangat tinggi. Keluaran dari sensor DHT22 adalah sinyal digital. Didalam komponen sensor DHT22 terdiri dari komponen sensor kelembaban dan sensor temperatur atau *thermistor*. Kedua sensor ini telah terhubung dengan IC 8-bit di dalamnya yang mengkonversi keluaran menjadi sinyal digital. Sehingga sensor DHT22 dapat dihubungkan secara langsung dengan perangkat NodeMCU ataupun dengan perangkat mikrokontroler lainnya. Alur kerja sensor DHT22 dapat dilihat pada gambar 8.



Gambar 8. Alur Kerja Sensor DHT22

Dikarenakan didalam sensor DHT22 telah terdapat sebuah IC 8-bit yang berfungsi untuk mengubah keluaran dari sensor, maka keluaran dari sensor dapat diolah secara langsung melalui pin GPIO yang ada pada mikrokontroler NodeMCU. Untuk penelitian ini, sensor akan dihubungkan melalui pin GPIO2 atau D4 pada *board* NodeMCU ESP8266. Seperti yang ditunjukkan pada Gambar 9.



Gambar 9. Koneksi NodeMCU dan Sensor DHT22

3. Sintaks Pengambilan data Sensor NodeMCU

Berikut adalah sintaks untuk mendapatkan data dari sensor MQ135 dan DHT22.

```

1 #define ANALOGPIN A0
2 #define DHTTYPE DHT22
3 #define DHTPIN 2
4 MQ135 gasSensor = MQ135(ANALOGPIN);
5 DHT dht(DHTPIN, DHTTYPE);
6 float ppm = gasSensor.getPPM();
7 float kelembaban = dht.readHumidity();
8 float temperatur = dht.readTemperature();
9 if(!isnan(kelembaban) || !isnan(temperatur) ||
10 !isnan(temperaturFahren)) {
11 Serial.println("pembacaan sensor gagal!");
12 return;
  
```

Gambar 10. Sintaks Pengambilan Data Sensor

Sintaks tersebut terdiri dari proses inialisasi pin yang digunakan pada mikrokontroler NodeMCU dan dilanjutkan dengan pengambilan data ppm CO₂ serta pengambilan nilai kelembaban dan temperatur dengan memanfaatkan *method* dari *library* MQ135.h dan DHT.h yang telah disediakan secara *online* pada *website* Arduino.

4. Perancangan RESTful API

RESTful API merupakan implementasi dari API (*Application Programming Interface*). REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk dikembangkan. Komponen yang dibangun dalam RESTful API adalah URL design, dan verbs yang akan digunakan. Bentuk yang akan digunakan untuk URL design untuk API pada sistem ini adalah *'airquality/api/sensor'* sedangkan untuk *verbs* yang digunakan adalah *verbs* POST dan GET. *Verbs* POST digunakan untuk pengiriman data dari alat ke *webserver*. Sedangkan *verbs* GET digunakan untuk pengambilan data oleh aplikasi *Smartphone* dan *Web display*.

6. Analisis *Multiplatform* pada Ionic

Bentuk dari aplikasi ionic berupa aplikasi hybrid. Dimana sebuah aplikasi yang dibuat menggunakan ionic dapat dirilis di lebih dari 1 platform alias *multiplatform*. Tahapan dalam membangun aplikasi

multiplatform pada ionic adalah dengan cara menambahkan *platform* yang akan digunakan pada *project*. Untuk penambahan *platform* pada *project* ionic dapat menggunakan CLI dengan *command* seperti pada Gambar 3.

```
ionic cordova platform add $platform
```

Gambar 3. *Command* tambah *platform*

Platform diganti dengan *platform* yang akan digunakan seperti ios, android, windows atau browser. Contoh untuk menambahkan *platform* browser seperti pada Gambar 4.

```
ionic cordova platform add browser
```

Gambar 4. Tambah *platform* browser

Untuk melihat daftar *platform* yang telah ditambahkan pada *project* dapat dilihat pada *ProjectFolder\platforms*.

Setelah penambahan *platform* selesai, maka aplikasi dapat dibangun sesuai dengan *platform* yang dibutuhkan. Tetapi ada beberapa kebutuhan yang harus dipenuhi sebelum *project* dibangun kedalam aplikasi untuk *platform* yang diperlukan. Ionic menyediakan dokumentasi untuk kebutuhan dalam membangun aplikasi untuk *platform* yang diinginkan. Dengan menggunakan CLI, *command* pada Gambar 5, dapat digunakan untuk membangun *project* menjadi aplikasi yang dapat digunakan oleh *platform*.

```
ionic cordova build $platform
```

Gambar 5. *Command* Build Aplikasi

Contoh *command* untuk meng-*compile* aplikasi web dapat dilihat pada Gambar 6.

```
ionic cordova build Browser
```

Gambar 6. *Build* Aplikasi browser

Hasil keluaran dari *command* diatas telah menghasilkan aplikasi web yang dapat langsung di gunakan sebagai aplikasi web. Untuk hasilnya dapat diambil pada *ProjectFolder\platforms\platforms\browser\www*. Hasil ini dapat langsung di hosting kedalam *webserver*.

7. Analisis Peramalan *Exponential Smoothing*

Metode pemulusan eksponensial atau *Exponential Smoothing* merupakan pengembangan dari metode *Moving Averages*. Cara kerja dari metode peramalan ini dilakukan dengan cara mengulang perhitungan secara terus menerus dengan menggunakan data-data sebelumnya, dimana setiap data akan diberi bobot. Untuk metode pemulusan eksponensial tiga parameter dari winter menggunakan nilai bobot *alpha* (α), *beta*

(β) dan *gamma* (γ). Nilai bobot ini berfungsi untuk melakukan penghalusan terhadap nilai peramalan. Besaran bobot ditentukan secara acak (*trial and error*) sampai ditemukan nilai yang menghasilkan kesalahan peramalan paling kecil. Nilai bobot berkisar antara 0 dan 1.

Tabel 2. Sampel Data

| Tahun | Kuartal | Periode (t) | Nilai Observasi X | Nilai Prakiraan | | | F |
|-------|---------|-------------|-------------------|--------------------|-------------------|--------------------|--------|
| | | | | S ($\alpha=0,6$) | T ($\beta=0,2$) | I ($\gamma=0,5$) | |
| 2001 | 1 | 1 | 460 | | | 0,96 | |
| | 2 | 2 | 484 | | | 1,01 | |
| | 3 | 3 | 530 | | | 1,11 | |
| | 4 | 4 | 441 | 441,00 | 10,25 | 0,92 | |
| 2002 | 1 | 5 | 492 | 488,00 | 17,60 | 0,98 | 48,538 |
| | 2 | 6 | 509 | 504,62 | 17,40 | 1,01 | 52,724 |
| | 3 | 7 | 588 | 526,65 | 18,33 | 1,11 | 60,492 |
| | 4 | 8 | 490 | 537,55 | 16,84 | 0,92 | 51,005 |
| 2003 | 1 | 9 | - | | | | 54,558 |
| | 2 | 10 | - | | | | 57,658 |
| | 3 | 11 | - | | | | 65,469 |
| | 4 | 12 | - | | | | 55,398 |

Langkah-langkah yang dilakukan dalam metode peramalan ini adalah:

1. Menentukan nilai pemulusan awal (St) yang dapat disamakan dengan nilai aktual periode tersebut. Contoh dari sampel data pada Tabel 2 adalah nilai 441,0
2. Hitung inisialisasi nilai pemulusan trend dengan persamaan seperti berikut.

$$T_4 = \frac{1}{4} \left\{ \frac{492 - 460}{4} + \frac{509 - 484}{4} + \frac{588 - 530}{4} + \frac{490 - 441}{4} \right\} = 10,25$$

3. Hitung inialisasi pemulusan musiman satu musim awal dengan cara membagi setiap data pengamatan (X) dengan rata - rata data pengamatan pada siklus itu.

Rata - rata tahun 2001:

$$X_{2001} = (460 + 484 + 530 + 441) / 4 = 478,75$$

Sehingga nilai inisial indeks musimannya

$$I_1 = 460 / 478,75 = 0,96$$

$$I_2 = 484 / 478,75 = 1,01$$

$$I_3 = 530 / 478,75 = 1,11$$

$$I_4 = 441 / 478,75 = 0,92$$

4. Hitung nilai level(S) pada periode - periode selanjutnya dengan persamaan berikut.

$$S_5 = 0,6 \left(\frac{492}{0,96} \right) + (1 - 0,6)(441,0 + 10,25)$$

5. Hitung trend(T) pada periode - periode selanjutnya dengan seperti berikut.

$$T_5 = 0,2(488,0 - 441,0) + (1 - 0,2)10,25$$

6. Hitung nilai musiman pada periode - periode selanjutnya seperti berikut.

$$I_5 = 0,5 \left(\frac{492}{488} \right) + (1 - 0,5)0,96$$

7. Hitung nilai peramalan yang digunakan untuk pencocokan model dan data aktual dengan persamaan berikut.

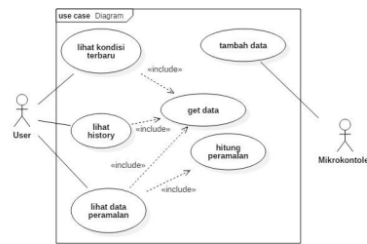
$$F_5 = (488 + 10,25)0,96$$

8. Hitung peramalan untuk beberapa periode yang akan datang seperti berikut.

$$F_9 = (490 + 16,84 * 1)0,98$$

D. Use Case Diagram Aplikasi Mobile Android

Diagram usecase dari aplikasi mobile yang dibangun terdiri dari enam fungsional, yaitu lihat kondisi terbaru, lihat histori, lihat peramalan, get data, hitung peramalan dan tambah data. Sedangkan actor yang terlibat yaitu mikrokontroler dan user. seperti yang dapat dilihat pada Gambar 7.



Gambar 7. Diagram Use Case Aplikasi Android

E. Implementasi Sistem

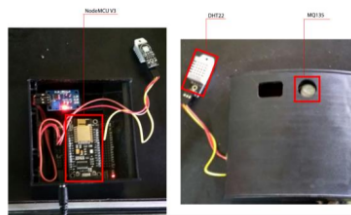
Tahap implementasi merupakan tahap dimana aplikasi monitoring kualitas udara berbasis android dengan bantuan mikrokontroler NodeMCU siap dioperasikan. Tahapan ini juga akan menjelaskan mengenai implementasi perangkat keras, implementasi perangkat lunak, dan implementasi program serta antarmuka aplikasi untuk monitoring kualitas udara.

E.1. Implementasi Perangkat Mikrokontroler

Perangkat keras yang digunakan dalam pembangunan alat untuk mengambil data ppm dari karbondioksida, suhu dan kelembaban pada lokasi studi kasus. Berikut adalah daftar komponen lengkap yang digunakan dalam implementasi perangkat keras alat yang dibangun.

1. Mikrokontroler NodeMCU V3 LoLin
2. Base Board NodeMCU
3. Adaptor 12V 1A
4. Sensor MQ135
5. Sensor DHT22

Untuk bentuk hasil akhir dari alat dapat dilihat pada Gambar 16.



Gambar 16. Implementasi Alat

E.2. Implementasi Perangkat Lunak

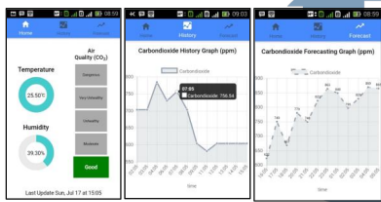
Perangkat lunak pendukung yang digunakan untuk mengimplementasikan aplikasi ini dapat dilihat pada Tabel 3.

Tabel 3. Implementasi Perangkat Lunak

| No. | Jenis Perangkat Lunak | Spesifikasi |
|-----|-----------------------|---|
| 1. | Sistem Operasi | Windows 10 Pro 64-bit |
| 2. | Editor | Arduino IDE, Sublime Text 3, Visual Studio Code |
| 3. | Framework | Ionic, Yii 2.0 |
| 4. | Bahasa pemrograman | php, AngularJS, Arduino C |
| 5. | Database | MySQL |

E.3. Implementasi Aplikasi

Aplikasi monitoring kualitas udara berbasis android dan NodeMCU ini membutuhkan proses instalasi pada perangkat android. Pengguna harus mempunyai aplikasi yang telah terinstall pada perangkat *smartphone*. Untuk hasil implementasi tampilan pada aplikasi Android dapat dilihat pada Gambar 17.



Gambar 8. Antarmuka Android

Dari Gambar 8. Antarmuka untuk *home* ditunjukkan pada gambar paling kiri. Pada halaman ini terdapat indikator kondisi kualitas udara, serta data kelembaban dan temperatur yang terakhir dikirimkan oleh alat. Halaman histori ditunjukkan oleh gambar kedua, dimana dalam halaman ini terdapat grafik histori nilai ppm karbondioksida. Sedangkan untuk halaman peramalan terdapat pada gambar paling kanan. Halaman ini menampilkan hasil perhitungan peramalan nilai ppm karbondioksida yang telah diolah kedalam bentuk grafik.

F. Pengujian User Acceptance

Pengujian *user acceptance* dilakukan pada bulan Juli 2017 dengan menyebarkan kuesioner terhadap 19 responden sebagai pengguna aplikasi. Berikut merupakan hasil dari pengujian kuesioner dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian Kuesioner

| Pernyataan Pertama: Sistem dapat memberikan informasi mengenai indeks kualitas udara. | | | | |
|---|---|----|----|-----|
| SS | S | RR | TS | STS |
| 11 | 8 | 0 | 0 | 0 |
| Rata – Rata = (55+32+0+0+0)/19 = 4.58 | | | | |

Pernyataan Kedua: Dengan adanya sistem ini, pengguna dapat memonitor kondisi udara di suatu tempat dari jarak jauh.

| SS | S | RR | TS | STS |
|---------------------------------------|----|----|----|-----|
| 5 | 13 | 1 | 0 | 0 |
| Rata – Rata = (25+42+3+0+0)/19 = 3.68 | | | | |

Pernyataan Ketiga: Ramalan sistem dapat dimanfaatkan untuk mengantisipasi kualitas udara buruk yang akan terjadi.

| SS | S | RR | TS | STS |
|---------------------------------------|----|----|----|-----|
| 4 | 11 | 3 | 1 | 0 |
| Rata – Rata = (20+44+9+2+0)/19 = 3.95 | | | | |

Pernyataan Keempat: Tampilan aplikasi *smartphone* mudah untuk dipahami dan digunakan.

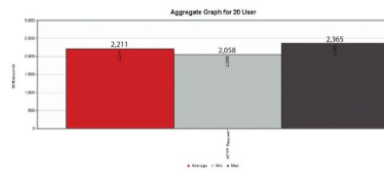
| SS | S | RR | TS | STS |
|--|----|----|----|-----|
| 7 | 10 | 2 | 0 | 0 |
| Rata – Rata = (35+40+6+0+0)/19 = 4.26 | | | | |
| Rata – Rata Akhir = (4.58+3.68+3.95+4.26)/4 = 4.12 | | | | |

Berdasarkan hasil pengujian beta pada Tabel 4. Dapat diambil kesimpulan bahwa mayoritas responden menjawab setuju mengenai fungsi sistem *monitoring* yang dibangun dalam memberikan informasi indeks kualitas udara.

G. Pengujian Performa API

Pengujian ini dilakukan untuk mengetahui kecepatan pengambilan data dari API yang telah dibangun. Pengujian dilakukan dengan bantuan perangkat pendukung Apache JMeter.

Dalam pengujian ini, diasumsikan bahwa ada 20 user. Jeda antar user mengakses API adalah 1 detik. Dapat dilihat dalam grafik bahwa kecepatan rata – rata *response* dari API adalah 2,211ms. Untuk waktu paling minimum atau paling cepat adalah 2,058ms dan waktu maksimum adalah 2,365ms. Gambar 18 merupakan hasil pengujian performa API dari aplikasi yang dibangun.



Gambar 18. Hasil pengujian performa API

V. SIMPULAN

Adapun kesimpulan dan saran yang didapatkan dari hasil penelitian ini yaitu:

A. Kesimpulan

Berdasarkan hasil yang didapat dari penelitian yang dilakukan maka dapat ditarik kesimpulan sebagai berikut:

1. Sistem dapat memberikan informasi mengenai indeks kualitas udara di area sekitar yang telah ditentukan oleh penggunanya.
2. Sistem dapat memberikan ramalan polusi udara untukantisipasi adanya kualitas udara yang buruk.

B. Saran

Adapun saran-saran untuk pengembangan Aplikasi monitoring kualitas udara ini adalah sebagai berikut:

1. Penambahan sensor yang dimonitor tidak hanya gas karbondioksida (CO₂).
2. Pencarian nilai parameter *alpha*, *beta*, dan *gamma* agar dapat dicari secara langsung oleh aplikasi.

DAFTAR PUSTAKA

- [1] E. M. Wyrwal, G. Dziubanek, M. Rusin, K. Oleksiuk and M. Kubasiak, "Impact of Air Pollution on Public Health," *The Impact of Air Pollution on Health, Economy, Environment and Agricultural Sources*, 2011.
- [2] R. Amrutkar, S. Vikharankar and L. Ahire, "Security: Smart Homes Using Internet of Things (IoT)," *International Engineering Research Journal (IERJ)*, vol. 2, no. 2, pp. 558-561, 2016.
- [3] S. S. Devi and G. Vijaykiran, "Things of Internet Based Smart Environmental Monitoring Using Node MCU," *International Journal of Scientific Engineering and Technology Research*, vol. 6, no. 4, 2017.
- [4] R. Jin, T. Liu and S. Wang, "The Application of Holt Exponential Smoothing Model on 2014 Air Quality Index in Miyun County, Beijing, China," *Bulletin of Mathematics*, vol. 3, no. 4, 2015.
- [5] W. E. Cahyono, "Kajian Tingkat Pencemaran Sulfur Dioksida dari Industri di Beberapa Daerah di Indonesia," *Berita Dirgantara*, vol. 12, no. 4, pp. 133-134, 2011.
- [6] Y. Wicaksono and A. Suismoni, "Deteksi Gas Berbahaya CO, CO₂, NOX Dengan Penampil Dot Matriks dan level Bahaya Serta Besarnya," p. 3, 2010.
- [7] R. R. Hardani and Sarwosri, "Rancang Bangun Aplikasi Perangkat Bergerak berbagi Foto Berbasis Android menggunakan API Facebook, Flickr dan Picasa," *Jurnal Teknik Pomits*, vol. 1, no. 1, pp. 1-2, 2012.
- [8] S. Bhardwaj, P. Chouhan, R. Sharma and P. Sharma, "Android Operating Systems," *International Journal of Engineering Technology & Management Research*, vol. 1, no. 1, 2015.
- [9] S. Yusuf, in *Ionic Framework By Example*, Birmingham, Packt Publishing, pp. 5-10, 2016.
- [10] Jonnius, J., & Ali, A. Analisis Forecasting Penjualan Produk Perusahaan. *Kutubkhanah*, vol 15 No.2, pp. 129-137, 2012.

Implementasi Monitoring Kualitas Udara Menggunakan Peramalan Exponential Smoothing dan NodeMCU Berbasis Mobile

ORIGINALITY REPORT

| | | | |
|------------------|------------------|--------------|----------------|
| 5% | % | 0% | 5% |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|----|
| 1 | Submitted to Universitas Brawijaya Student Paper | 2% |
| 2 | Submitted to Academic Library Consortium Student Paper | 2% |
| 3 | Submitted to State Islamic University of Alauddin Makassar Student Paper | 2% |

Exclude quotes On

Exclude matches < 2%

Exclude bibliography On