TESIS – TE142599

KLASIFIKASI AIR SUNGAI BERBASIS KOMBINASI TEKNOLOGI IOT-BIG DATA MENGGUNAKAN SVM

RIZQI PUTRI NOURMA BUDIARTI
NRP 2214205202

DOSEN PEMBIMBING
Mochamad Hariadi, S.T., M.Sc., Ph.D.
Prof. Ir. Mauridhi Hery Purnomo, M.Eng., Ph.D.

PROGRAM MAGISTER
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

TESIS – TE142599

KLASIFIKASI AIR SUNGAI BERBASIS KOMBINASI TEKNOLOGI IOT-BIG DATA MENGGUNAKAN SVM

RIZQI PUTRI NOURMA BUDIARTI
NRP 2214205202

DOSEN PEMBIMBING
Mochamad Hariadi, S.T., M.Sc., Ph.D.
Prof. Ir. Mauridhi Hery Purnomo, M.Eng., Ph.D.

PROGRAM MAGISTER
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

oleh:

Rizqi Putri Nourma Budiarti
NRP. 2214205202

Tanggal Ujian : 9 Januari 2017
Periode Wisuda : Maret 2017

Disetujui oleh:

1. Mochamad Hariadi S.T., M.Sc., Ph.D (Pembimbing I)
   NIP: 19691209 199703 1 002

2. Prof. Ir. Mauridhi Hery Purnomo, M.Eng., Ph.D (Pembimbing II)
   NIP: 19580916 198601 1 001

3. Dr. Supeno Mardi Susiki N., S.T., M.T (Penguji)
   NIP: 19700313 199512 1 001

4. Dr. Eko Mulyanto Y., S.T., M.T (Penguji)
   NIP: 19680601 199512 1 009

5. Dr. Diah Puspito Wulandari, S.T., M.Sc (Penguji)
   NIP: 19801219 200501 2 001

an. Direktur Program Pascasarjana
Asisten Direktur

Prof. Dr. Ir. Tri Widjaja, M.Eng.
NIP. 19611023 198603 1 001

Direktur Program Pascasarjana

Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D
NIP. 19601202 198701 1 001

Halaman ini sengaja dikosongkan

Halaman ini sengaja dikosongkan

KLASIFIKASI AIR SUNGAI BERBASIS KOMBINASI TEKNOLOGI IOT-BIG DATA MENGGUNAKAN SVM

Nama mahasiswa : Rizqi Putri Nourma Budiarti
NRP : 2214205202
Pembimbing : 1. Mochamad Hariadi, S.T., M.Sc., Ph.D
2. Prof. Ir. Mauridhi Hery Purnomo, M.Eng., Ph.D

ABSTRAK

Betapa pentingnya peranan air bagi kehidupan makhluk hidup, tidak hanya bagi manusia bahkan makhluk hidup lainnya membutuhkan air sebagai salah satu unsur pendukung keberlangsungan kehidupan pada tiap makhluk hidup. Untuk menjaga keberlangsungan sumber daya air khususnya air sungai diperlukan sistem monitoring yang mampu mengambil parameter kualitas air dengan menggunakan sensor. Telah banyak tersedia perangkat sensor kualitas air, namun masih belum ada sistem monitoring yang mampu melakukan klasifikasi kualitas air tersebut secara interaktif dan akurat.

Untuk mengatasi permasalahan tersebut, pada penelitian ini akan di buat sistem perangkat Internet of Things yang terdiri dari sensor kualitas air YSI 600R, sistem benam *Raspberry Pi* 3 dan perangkat komunikasi 4G. Selain itu telah dibuat juga sistem *Big Data* yang dilengkapi dengan fitur machine learning yang dapat melakukan klasifikasi kualitas air. Proses monitoring dilakukan pada area intake PDAM Karang Pilang dengan klasifikasi menggunakan metode *Support Vector Machine*. Hasil dari sistem ini dapat mengklasifikasi kualitas air sungai tersebut secara interaktif dan akurat.

Hasil penelitian ini menunjukkan bahwa dengan metode *Support Vector Machine* menghasilkan performance nilai akurasi total untuk SVM dengan kernel Linear adalah 0.9138 dan SVM dengan kernel RBF adalah 0.8372.Pengujian hasil validasi telah dilakukan berdasarkan grafik ROC dengan nilai area Under ROC menunjukan 0.93. Dengan begitu dapat dikatakan bahwa unjuk kerja berdasarkan nilai Area Under ROC “Excellent”.

Kata kunci: Klasifikasi Air Sungai, *Internet of Things*, *Big Data*, *Support Vector Machine*.

Halaman ini sengaja dikosongkan

SURFACE WATER CLASIFICATION BASED ON COMBINATION OF IOT-BIG DATA TECHNOLOGIES USING SVM

By : Rizqi Putri Nourma Budiarti
Student Identity Number : 2214 205 202
Supervisor(s) : 1. Mochamad Hariadi, S.T., M.Sc., Ph.D
2. Prof. Ir.Mauridhi Hery Purnomo, M.Eng, Ph.D

ABSTRACT

How important to role of water for the survival of living beings, not only for human but also the other living beings need water as one of the element that supporting the continuity of life in every living creature. In order to maintain the necessary of water resources such as river, recently the need of monitoring system that able to take the parameter of water quality using sensors really important. Nowadays, has been widely the available of water quality sensor devices, but there isn't monitoring system that able to perform the classification of water quality in interactive and accurate.

To overcome these problems, in this thesis we built a Prototype the Internet of Things system consisting of YSI water quality sensors 600R, embedded systems Raspberry Pi 3 and 4G communication device. Additionally, we built also Big-Data system that equipped with machine learning algorithm that can perform water quality classification with Support Vector Machine method. This system monitor every activities on PDAM Karang Pilang and applying classification. The result of this sistem is able to perform the classification of river water quality in interactive and accurate.

The result are, we were able to do classification by using Support Vector Machine with accuracy level 0.9138 by using Linear kernel and 0.8372 by using RBF kernel. From ROC result, we achieved AUC value until 0.93. Its mean we achieved excelent result.

Keywords: River water classification, Internet of Things, Big Data, Support Vector Machine.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Bismillahirahmanirrahim

Alhamdulillah, Segala puji syukur kehadirat Allah SWT terucap atas anugerah, rahmat dan karunia-Nya sehingga pada akhirnya penulis dapat menyelesaikan Tesis yang berjudul :

"KLASIFIKASI AIR SUNGAI BERBASIS KOMBINASI TEKNOLOGI IOT-BIG DATA MENGGUNAKAN SVM"

Penulisan Tesis ini disusun guna memenuhi persyaratan dalam pencapaian gelar Magister Teknik pada Bidang Keahlian Jaringan Cerdas Multimedia – Jurusan Teknik Elektro – Fakultas Teknologi Industri – Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Selain itu, guna memenuhi beban 6 SKS (Satuan Kredit Semester) sesuai dengan sistem perkuliahan studi Magister di ITS.

Dalam penyusunan buku Tesis ini, penulis menyadari masih banyak terdapat kesalahan dan kekurangan. Namun, dalam usaha untuk dapat menyempurnakan buku Tesis ini di masa mendatang, penulis mengharapkan adanya kritikan, ide dan saran yang nantinya buku ini dapat dikembangkan penulis menjadi lebih baik lagi.

Harapan penulis pada buku Tesis ini, semoga buku ini dapat memberikan informasi dan manfaat yang seluas-luasnya bagi pembaca pada umumnya dan para akademisi khususnya mahasiswa Teknik Elektro ITS.

Surabaya, 3 Januari 2017

Penulis
Rizqi Putri Nourma Budiarti

Halaman ini sengaja dikosongkan.

UCAPAN TERIMA KASIH

Bismillahirahmanirrahim

Dengan menyebut nama Allah yang Maha Pengasih lagi Maha Penyayang. Alhamdulillah, Segala puji syukur penulis panjatkan kehadirat Allah SWT atas segala Rahmat dan Karunia-Nya sehingga penulisan buku Tesis ini dapat terselesaikan.

Ucapan terimakasih, penghormatan dan penghargaan setinggi-tingginya penulis ucapkan kepada Bapak Mochamad Hariadi, S.T., M.Sc., Ph.D. selaku pembimbing I yang dengan penuh perhatian memberikan dorongan, bimbingan, saran serta pembelajaran yang sangat berharga. Ucapan terimakasih, penghormatan dan penghargaan setinggi-tingginya pula penulis ucapkan kepada Bapak Prof. Ir. Mauridhi Hery P, M.Eng., Ph.D. selaku pembimbing II yang dengan penuh kesabaran, ketelatenan dan perjuangan memberikan semangat, bimbingan, arahan serta saran yang sangat berguna selama ini.

Dengan terselesaikannya buku tesis ini, perkenankanlah pula saya mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Suami penulis Sritrusta Sukaridhoto atas segala doa, perhatian, waktu dan bantuan selama penulis melakukan penelitian serta Anak-anak penulis tersayang Sritrusta Aulia R.G. dan Sritrusta Ryu R. yang selalu mendoakan penulis.
2. Kedua Orang Tua penulis Kukuh Fauzy dan Sumartini yang telah membesarkan, memberikan pendidikan serta doa yang selalu terucapkan dari kecil hingga sekarang.
3. Mertua penulis, Ibu Ida Rodiah Sukardjono, atas segala doa yang terpanjatkan, bantuan dan perhatian untuk segera menyelesaikan tesis ini.
4. Rektor Institut Teknologi Sepuluh Nopember Surabaya, atas kesempatan dan fasilitas yang diberikan kepada saya untuk mengikuti dan menyelesaikan program magister ini.
5. Segenap dosen-dosen di Teknik Elektro khususnya pada bidang keahlian Jaringan Cerdas Multimedia, penulis mengucapkan terima kasih dan

penghormatan dari lubuk hati yang paling dalam atas bimbingan selama kuliah, ilmu, pengalaman serta masukan-masukan yang sangat bermanfaat.

6. Bapak Anang Tjahjono, yang telah membantu menyediakan perangkat sensor YSI600R untuk penelitian ini.
7. Bapak Nanang Widyatmoko, selaku manager operasional maintance PDAM Surya Sembada dan Tim IT PDAM atas segala bantuan pada penelitian ini.
8. LPDP yang telah memberikan bantuan beasiswa tesis bagi penulis dalam melakukan penelitian ini.
9. Teman-teman Jaringan Cerdas Multimedia angkatan 2014, 2015, 2016 baik ganjil maupun genap, terima kasih atas kebersamaan dalam suka dan duka didalam perkuliahan serta pertemanan yang indah selama ini, semoga persahabatan ini akan tetap terjalin.
10. Teman-teman di Lab B204 dan B401 (Para Senior S3, kawan sesama S2, serta adik-adik S1), teman-teman di Telematika, Telkom, Elektronika, Pengaturan, dan Tenaga Listrik terima kasih atas segala dukungan dan bantuannya selama ini.

Terima kasih penulis ucapkan kepada pihak-pihak yang tidak dapat disebutkan satu per satu, tanpa Bapak, Ibu, dan rekan-rekan sekalian penulis tidak akan mungkin menyelesaikan Tesis ini. Terima kasih atas segala dukungannya. Selain ucapan terima kasih, penulis juga memohon maaf apabila selama menempuh studi magister di ITS terdapat kesalahan penulis dalam bertutur kata, bersikap, dan bertindak yang tidak berkenan. Akhirnya dengan segenap hati, penulis berharap agar Allah Yang Maha Esa selalu melimpahkan Rahmat-Nya dan membalas segala kebaikan yang telah diberikan.

Surabaya, 3 Januari 2017

Penulis

Rizqi Putri Nourma Budiarti

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Betapa pentingnya peranan air bagi kehidupan makhluk hidup, tidak hanya bagi manusia bahkan makhluk hidup lainnya membutuhkan air sebagai salah satu unsur pendukung keberlangsungan kehidupan pada tiap makhluk hidup. Sebenarnya air merupakan sumber daya alam yang tidak terbatas, karena sumber air bisa didapatkan dari beberapa sumber air. Menurut perda Surabaya no.2 tahun 2004 mengenai definisi air bahwa semua air yang terdapat di dalam atau berasal dari sumber baik yang terdapat di atas maupun di bawah permukaan tanah, termasuk air laut yang dimanfaatkan di darat dan sumber sumber air adalah tempat-tempat dan wadah air, baik yang terdapat di atas maupun di bawah permukaan tanah, termasuk akuaifer, mata air, sungai, rawa, danau dan waduk[1]. Sekarang ini, pertumbuhan jumlah penduduk yang semakin meningkat menyebabkan permintaan untuk konsumsi air bersih pun meningkat. Permintaan untuk ketersediaan dan konsumsi air yang terus meningkat apabila tidak diimbangi dengan kualitas air yang sesuai standart dan pencemaran air menyebabkan air menjadi sumber daya alam yang terbatas dan selalu mengalami penurunan kualitas.

Merujuk pada Peraturan Daerah Kota Surabaya No.2 Tahun 2004 mengenai pengelolaan kualitas air dan pengendalian pencemaran bahwa air adalah sumber daya alam yang harus dapat dimanfaatkan untuk memenuhi hajat hidup orang banyak, oleh karena itu perlu dijaga kelestarian dan keberlangsungan fungsi air terutama pada sumber-sumber air untuk meningkatkan kesejahteraan manusia. Untuk melestarikan fungsi air perlu dilakukan pengelolaan kualitas air dan pengendalian pencemaran air secara bijaksana dengan memperhatikan kepentingan generasi sekarang dan mendatang serta keseimbangan ekologis. Untuk mewujudkan keberlangsungan fungsi air, maka berbagai pengendalian pencemaran air sudah banyak dilakukan. Akan tetapi, masih ada segelintir orang yang membuang sampah sembarangan, pembuangan limbah industri yang belum sempurna ke sungai, maupun limbah rumah tangga yang tidak terolah dengan

baik. Berbagai permasalahan air baik yang secara langsung maupun tidak langsung telah menyebabkan pencemaran air meningkat.

Oleh karena itu, salah satu upaya dalam pengendalian pencemaran air dengan menerapkan sistem monitoring yang mampu mendeteksi dan melakukan proses pengelompokan kualitas air. Pengelompokan atau pengklasifikasian dibutuhkan untuk menilai kondisi air pada suatu ekosistem air di setiap titik waktu. Dengan begitu dapat diketahui apakah kondisi air tersebut berkualitas baik (bersih) atau buruk (tercemar). Sehingga sistem ini diharapkan bisa membantu pemerintah dalam merencanakan langkah selanjutnya yang mungkin diperlukan dalam pembenahan ekosistem lingkungan air menjadi lebih baik.

Dalam mewujudkan ekosistem lingkungan yang berperan dalam pengelolaan dan pengolahan sumber air menjadi lebih baik, beberapa penelitian telah dilakukan untuk mengetahui kandungan air. Kandungan air dalam suatu sistem lingkungan dapat diketahui dengan cara manual, dimana prosesnya dilakukan dengan cara mengambil sample air menggunakan alat seperti gelas, gayung, maupun alat lainya seperti botol. Hasil sample air yang diambil, dicek menggunakan laboratorium untuk mengetahui kandungan air tersebut. Proses manual ini membutuhkan waktu sekitar 1 sampai 30 hari tergantung pada parameter apa saja yang ingin diukur dan diteliti. Hasil yang diperoleh dari prosedur manual sangat bergantung pada seberapa terampil dan ahli dalam proses pengukuran air tersebut. Pada kenyataannya, saat ini proses pengukuran parameter-parameter kualitas air yang dilakukan PDAM (Perusahaan Daerah Air Minum) dan BLH (Badan Lingkungan Hidup) masih dilakukan secara manual, serta pengambilan sample air dilakukan secara periodik misalkan seminggu sekali atau sebulan sekali. Selain menggunakan prosedur manual, pengukuran dapat dilakukan secara metode otomatis dengan menggunakan sensor air untuk mengetahui kualitas air.

Metode pengambilan data dengan menggunakan sensor, sistem benam dan memanfaatkan teknologi komunikasi jarak jauh yang selalu terhubung dengan *Internet* merupakan salah satu implementasi dari teknologi *Internet of Things* (IoT). Dimana dengan teknologi ini mampu membantu mempermudah untuk mendapatkan parameter kualitas air.

Kualitas air biasanya digambarkan dalam beberapa parameter. Parameter yang digunakan bermacam-macam seperti pada paper[2][3][4]. Dari parameter yang dihasilkan, beberapa peneliti menggunakannya untuk menentukan prediksi nilai parameter selanjutnya seperti pada paper[5]. Berbagai metode klasifikasi telah dilakukan baik secara manual, maupun komputasional dengan menggunakan *machine learning*. Metode klasifikasi dalam *machine learning* yang bisa digunakan antara lain *Decision Trees, Ruled-Based Classifiers, Artificial Neural Networks,* dan *Support Vector Machines* (SVM). Masing-masing metode memiliki teknik klasifikasi dan kelebihan yang berbeda. Pada Penelitian sebelumnya, paper [2] pengambilan sampling data dilakukan secara periodik tiap 5 hari sekali dan diambil dalam beberapa menit saja. Sedangkan proses klasifikasi belum menggunakan *machine learning*. Hal ini berbeda dengan paper[3] dimana sudah menggunakan proses klasifikasi menggunakan machine learning dengan metode teknik *Artificial Neural Network* dan pada paper[4] menggunakan metode Kombinasi antara *Neural Network* dan *Genetic Algorithms*. Hasil yang didapatkan dari kedua paper tersebut sudah cukup baik, tetapi data yang diambil masih menggunakan log data di tahun-tahun sebelumnya dan pengambilan data dilakukan secara manual dan periodik dilakukan dua kali dalam sebulan, selain itu belum menggunakan environment *Big Data*. Perbandingan antara beberapa metode ANN, ANFIS dan SVM untuk permasalahan air sungai, dimana performansi SVM lebih baik[5].

Metode machine learning yang bisa digunakan dalam mengklasifikasikan data, antara lain menggunakan Regresi Logistik dan *SVM (Support Vector Machine). Regresi Logistik* dan *Support Vector Machine (SVM)* merupakan metode dalam *machine learning* dan *data mining* yang sering digunakan dalam mengolah data dalam jumlah besar[6][7]. SVM bekerja sangat baik pada data set dengan dimensi tinggi, bahkan SVM menggunakan teknik kernel untuk memetakan data asli dari dimensi asalnya menjadi dimensi lain yang lebih tinggi. Pada SVM, hanya sejumlah data terpilih sajalah yang berkonstribusi untuk membentuk model yang digunakan dalam klasifikasi yang akan dipelajari. Data-data yang terpilih ini dinamakan *Support Vector.* Jenis metode ini yang digunakan untuk mengklasifikasikan data-data yang berupa perolehan nilai dari beberapa

parameter air yang diperoleh dari sensor air. Dimana data tersebut yang akan dibaca dan disimpan terlebih dahulu pada sistem benam (*local-offline logging*) sebelum dikirimkan pada environment *Big Data* (*online logging*) menggunakan komunikasi data 4G.

Secara umum environment *Big Data* dapat diartikan sebagai kumpulan data yang berukuran sangat besar (*volume*), sangat cepat perubahan/pertumbuhannya (*velocity*), data beragam dalam berbagai bentuk/format (*variety*), serta memiliki nilai tertentu (*value*)[8]. Sekarang ini, data skala besar yang tersedia tidak mampu dalam memenuhi kebutuhan hanya menggunakan kapasitas data pada mesin tunggal. Oleh Karena itu, diperlukan banyak mesin untuk menyimpan data dalam memori sehingga memenuhi kebutuhan dan efisiensi pada proses komputasi. Era *Big Data* mulai populer dalam pengelolaan informasi dalam beberapa tahun ini, mengingat begitu pesatnya pertumbuhan data di internet, khususnya melalui media sosial.

Oleh karena itu, dengan latar belakang inilah maka penulis tertarik untuk melakukan penelitian secara lebih detail mengenai sistem klasifikasi kualitas air berbasis kombinasi IoT-*Big Data* dengan menggunakan metode SVM dimana dengan penelitian ini diharapkan akan mampu menjadi early warning terhadap kondisi air sungai terkini dan memicu kesadaran berbagai pihak untuk dapat melakukan perubahan dan memanfaatkan penelitian ini sehingga dampak yang ditimbulkan nantinya bisa dihindarkan.

1.2 Rumusan Masalah

Kebutuhan akan sistem *monitoring* interaktif pada kualitas air sungai yang dapat terintegrasi dengan kapasitas yang besar dan mampu untuk melakukan klasifikasi kualitas air dengan akurat sangatlah diperlukan. Karena itu dalam penelitian ini akan difokuskan bagaimana penerapan sistem yang terintegasi dengan teknologi IoT-*Big Data* yang mampu melakukan klasifikasi yang interaktif dan akurat. Objek penelitian akan dilakukan di PDAM Karang Pilang Surabaya. Sehingga rumusan masalah pada penelitian ini adalah: Bagaimana menerapkan sistem klasifikasi kualitas air sungai yang bersifat interaktif dan akurat?

1.3 Tujuan

Menciptakan sistem monitoring yang dapat melakukan pengklasifikasian parameter air untuk mengetahui tingkat status mutu air sungai. Hasil dari pengelompokan data dengan menggunakan kelas-kelas tertentu dapat digunakan untuk membantu memberikan informasi kepada sistem pemrosesan sebelum *Intake* PDAM apabila terjadi perubahan ekosistem lingkungan disekitar Area PDAM Karang Pilang.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut :

a. Sumber data yang digunakan, antara lain: 1) Sensor air YSI 600R yang dipasang pada aliran sungai Kali Surabaya di daerah Karang Pilang, sebelum pintu air PDAM Karang Pilang dengan masa percobaan dari Nopember 2016, 2) Sensor air PDAM dengan masa percobaan dari Maret hingga Agustus 2016 dan 3) Data laporan laboratorium PDAM mulai tahun 2014 hingga Oktober 2016. Semua data yang diolah adalah data air baku.
b. Data yang diambil berasal dari sensor air yang kemudian disimpan pada sistem benam sehingga diperoleh data *logger* yang tersimpan baik *local offline logging* dan *online logging* yang kemudian digunakan sebagai data input penelitian.
c. Parameter-parameter air yang digunakan adalah suhu, pH, DO, NTU, TSS, dan Salinitas dengan menggunakan metode Indeks Pencemaran.
d. Sistem IoT hanya digunakan untuk pengambilan data saja.
e. Perangkat lunak yang digunakan antara lain, Sistem *Big Data* Server yang menggunakan *Spark* dan *Hadoop*, libSVM pada *library Python* yaitu *Scikit-learn* sebagai tool *Machine Learning*, yang diintegrasikan pada *Jupyter* sebagai *web user interface interaktif* .
f. Metode pengklasifikasian yang digunakan yaitu *Support Vector Machine (SVM)* pada Sistem *Big Data*.

g. Pada penelitian ini tidak dibahas permasalahan manajemen dan keamanan data (*Security*) baik pada *Internet of Things (IoT)* maupun *Big Data*.
h. *Big Data* server yang digunakan masih menggunakan *Standalone* server.

1.5 Kontribusi

1. *Prototype* sensor kualitas air yang berbasis IoT dan terkoneksi server *Big Data*.
2. Sistem analisa kualitas air menggunakan metode SVM pada Environtment *Big Data*.

1.6 Metodologi Penelitian

Dalam penelitian ini, alur kerja yang digunakan adalah sebagai berikut :

1. Melakukan survey dan perijinan lokasi penelitian.
2. Mempelajari teori dasar mengenai air sungai dan parameter air yang digunakan dalam penelitian.
3. Mempelajari dan mengimplementasikan perangkat IoT dan perangkat lunak pada *Raspberry Pi* 3 dan *SQLite* untuk database local di embed ke sistem benam.
4. Mempelajari *Spark* untuk implementasi pada Sistem *Big Data*.
5. Menerapkan Aplikasi Klasifikasi *Spark* dan LibSVM pada *Spark*.
6. Melakukan pengumpulan dan pengambilan data parameter kualitas air dengan membuat sistem *logger* pada sistem benam dan pengiriman pada server sistem *Big Data*.
7. Melakukan preprosessing data sebagai penggabungan IoT dan *Big Data* secara langsung.
8. Melakukan pengklasifikasian Kualitas Air yang selanjutnya dilakukan penganalisa hasil dari implementasi sistem IoT-*Big Data* menggunakan metode SVM.
9. Menyusun laporan ke dalam bentuk buku Tesis.

1.7 Sistematika Penulisan

Secara garis besar, penulisan penelitian ini terdiri dari lima bab yang masing – masing bab menjelaskan secara terperinci yang berkaitan dengan penelitian ini. Sistematika penulisan pada buku tesis ini sebagai berikut:

BAB I : PENDAHULUAN

Pada Bab ini, memberikan penjelasan mengenai latar belakang penelitian, rumusan masalah, tujuan, batasan masalah, konstribusi, dan metodologi penelitian serta sistematika pembahasan.

BAB II : KAJIAN PUSTAKA DAN TEORI PENUNJANG

Pada Bab ini, berisi penjelasan mengenai kajian pustaka dan teoritis yang berkaitan tentang penelitian. Diawali dengan Sumber Daya Air, Sungai Kali Surabaya, Parameter Air, Raspberry Pi dan Sensor, *Big Data Framework, Spark, SVM (Support Vector Machine)* serta dasar membangun sistem IoT-*Big Data* dengan menggunakan metode SVM (*Support Vector Machine*) serta metode lain yang bersumber dari publikasi ilmiah maupun buku yang berkaitan dengan materi penelitian ini.

BAB III : METODOLOGI PENELITIAN

Pada Bab ini, mendeskripsikan uraian mengenai metodologi penelitian, perancangan sistem dalam bentuk desain awal sampai sistem desain IoT-*Big Data* yang terintegrasi dengan metode SVM dalam proses pengklasifikasian dalam melakukan penelitian ini.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Pada Bab ini, membahas tentang hasil penelitian dan pengujian sistem secara keseluruhan.

BAB V : PENUTUP

Pada Bab ini, berisikan kesimpulan yang diperoleh dari hasil penelitian dan pengujian sistem serta saran-saran dan masukan untuk penelitian selanjutnya.

Halaman ini sengaja dikosongkan

BAB 2
KAJIAN PUSTAKA DAN TEORI PENUNJANG

2.1 Kajian Penelitian Terkait

Beberapa peneliti telah melakukan penelitian tentang IoT, *Big Data* dan klasifikasi air sungai. Penelitian tersebut antara lain:

Penelitian air sungai telah banyak dilakukan antara lain: C.Veesommai dan Y.Kiyoki. "*Critical Contaminate Detection, Classification of Multiple-water-quality-parameters Value and Real Time Notification by rSPA Processes*"[2]. Penelitian ini membahas tentang parameter klasifikasi kualitas air dengan penggunaan metode rSPA dalam perhitungannya. Sarkar dan Pandey, dalam penelitian mereka yang berjudul "*River Water Quality Modelling Using Artificial Neural Network Technique*"[3] melakukan penelitian tentang pemodelan kualitas air sungai dengan menggunakan artificial neural network. Ding dan groupnya telah melakukan penelitian dengan judul "*The Use of Combined Neural Networks and Genetic Algorithms for Prediction of River Water Quality*"[4]. Namun sistem yang digunakan belum menggunakan machine learning dan framework *Big Data* untuk data analisanya.

Noori, Roohollah, Zhigiang Deng, Amin Kiaghadi and Fatemeh Torabi Kachoosangi, "*How Reliable Are ANN, ANFIS and SVM Techniques for Predicting Longitudinal Dispersion Coefficient in Natural Rivers*?"[5]. Penelitian ini tentang analisa kualitas air dalam beberapa metode yang dilakukan menggunakan ANN, ANFIS, dan SVM dimana hasil yang didapatkan menunjukkan bahwa untuk penggunaan machine learning kualitas air, metode SVM mendapatkan hasil yang lebih baik dan akurat daripada teknik analisis lainnya.

Shuxiu Liang, Songlin Han, Zhaochen Sun, "*Parameter optimization method for the water quality dinamic model based on data-driven theory*"[6]. Penelitian ini membahas tentang penelitian sungai yang terletak di china, serta menggunakan sistem pengklasifikasian SVM untuk menentukan klasifikasi dari

hasil parameter kualitas air yang didapatkan, namun untuk implementasinya masih belum menggunakan framework *Big Data*.

Yue Liao, Jianyu Xu, Wenjing Wang. "*A method of water quality assessment based on biomonitoring and multiclass Support Vector Machine*"[7]. Pada penelitian ini membahas penggunaan parameter biologi dari kualitas air dengan Interpretasi pada klasifikasi biota air dan menggunakan metode SVM multiclass. Namun proses pengklasifikasian datanya belum menggunakan framework *Big Data*.

Chen dan group, melakukan survey tentang aplikasi yang dapat digunakan pada teknologi *Big Data*. Penelitian mereka berjudul "*Big Data: A Survey*"[8]. Menyebutkan monitoring kualitas air sangat memerlukan teknologi *Big Data* karena mampu digunakan untuk menyimpan data yang besar dan dapat dilakukan klasifikasi.

Rao, Aravinda S., Stephen M, Jayavardhana G., Marimuthu P., Richard S.,dan Vincent P. "*Design of low-cost autonomous water quality monitoring system*"[9]. Penelitian ini membahas tentang perangkat sensor kualitas air, namun untuk perangkat sensor kualitas air yang digunakan untuk meneliti kualitas air dari parameter fisika dan kimia, dan digunakan untuk memonitoring kebiasaan habitat hewan di dalam air berkaitan dengan polusi air. Hasilnya menitikberatkan dari harga sensor yang dirangkai diperoleh dengan biaya yang lebih murah dan menitikberatkan pada proses normalisasi datanya. Pada penelitian ini masih belum ada proses klasifikasi dan belum menggunakan framework *Big Data*.

Herwindra B, bersama rekan-rekannya, melakukan penelitian tentang "*Design and Implementation of Smart Environment Monitoring and Analytics in Real-time System Framework Based on Internet of Underwater Things and Big Data*"[10], membangun *framework* untuk analisa dan monitoring lingkungan berbasis *Internet of Things* dan *Big Data*. Namun sistem mereka belum memberikan integrasi antara sistem *Big Data* dan analisa menggunakan machine learning.

Shifeng Fang bersama rekan-rekannya, membangun sistem monitoring linkungan untuk memantau iklim perubahan di daratan China di daerah XinJiang. Sistem ini menggunakan sensor udara, kelembaban dan pengendapan hujan. Paper

mereka berjudul "*An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things*"[11].

Kunwar P Singh bersama rekan-rekannya, melakukan pembandingan dalam manajemen sistem kualitias air dengan menggunakan SVM[12]. Dimana pada penelitian tersebut menggunakan data sensor BOD dan melakukan perbandingan antara SVM *linear* dan *non-linear*. Dari penelitian mereka didapatkan *non-linear* memberikan akurasi yang lebih baik dibanding dengan linear.

Aris-Kyriakos Koliopoulus bersama rekan-rekannya, melakukan penelitian tentang "*A Parallel Distributed Weka Framework for Big Data Mining using Spark"*[13], telah melakukan penelitian integrasi dengan menggunakan Weka dalam pemrosesan data *mining* dengan menggabungkannya pada pemrosesan data *Spark* dan memperoleh hasil mendekati skala *linear* dengan nilai 91,4 % dengan workloadnya bisa lebih eficien 4x daripada penggunaan Hadoop.

2.2 Sumber Daya Air Sungai

Berdasarkan Undang-Undang Republik Indonesia mengenai Sumber Daya Air No.7 Tahun 2004 menyebutkan bahwa sumber daya air adalah air, sumber air, dan daya air yang terkandung didalamnya. Air adalah semua air yang terdapat pada, di atas, ataupun di bawah permukaan tanah,termasuk dalam pengertian ini air permukaan, air tanah,air hujan dan air laut yang berada di darat. Sumber Air adalah tempat atau wadah air alami dan/atau buatan yang terdapat pada, di atas, ataupun di bawah permukaan tanah. Daya air adalah potensi yang terkandung dalam air dan/atau pada sumber air yang dapat memberikan manfaat ataupun kerugian bagi kehidupan dan penghidupan manusia serta lingkungannya[14].

Beberapa sumber air dapat diperoleh dari lokasi yang berbeda-beda dan untuk pemanfaatan yang berbeda pula seperti yang tertuang pada acuan Surat Keputusan Peraturan Daerah Kota Surabaya tentang Pengelolaan Kualitas Air dan Pengendalian Pencemaran Air No.2 di Jawa Timur Tahun 2004. Penggolongan kriteria air dibagi menjadi empat kelas[15], antara lain :

1. Kelas 1, yaitu air yang peruntukannya dapat digunakan untuk air bahan baku minum, dan atau peruntukan lain yang mensyaratkan mutu air yang sama dengan kegunaan tersebut.
2. Kelas 2, yaitu air yang peruntukannya dapat digunakan untuk sarana/prasarana rekreasi air, pembudidayaan ikan air tawar dan air payau, peternakan, air untuk mengairi taman, dan /atau peruntukan lain yang mensyaratkan mutu air yang sama dengan kegunaan tersebut.
3. Kelas 3, yaitu air yang peruntukannya dapat digunakan untuk pembudidayaan ikan air tawar dan air payau, peternakan, air untuk mengairi pertamanan, dan/atau peruntukan lain yang mensyaratkan mutu air yang sama dengan kegunaan tersebut.
4. Kelas 4, air yang peruntukannya dapat digunakan untuk mengairi pertamanan dan/atau peruntukan lain yang mensyaratkan mutu air yang sama dengan kegunaan tersebut.

Pembagian penggolongan kategori air ke beberapa kelas diatas tergantung pada nilai kualitas air yang digambarkan pada beberapa parameter-parameter air. Pada penelitian ini digunakan standar baku berdasarkan Peraturan Daerah Kota Surabaya No 02 Tahun 2004, Peraturan Pemerintah Republik Indonesia No 82 Tahun 2001 tentang peruntukan nilai baku kelas 1 dan untuk nilai kekeruhan menggunakan Peraturan Menteri Kesehatan No 492/Menkes/Per/IV/2010.

2.2.1 Parameter Kualitas Air

Pembagian penggolongan kategori air ke beberapa kelas tergantung pada parameter-parameter air. Pada penelitian ini digunakan aturan nilai baku kelas 1. Pembagian penggolongan kategori air ke beberapa kelas tergantung pada parameter-parameter air[16]. Penilaian parameter air dapat ditinjau dari beberapa perubahan indikator pencemaran air. Indikator yang sering kali digunakan sebagai tanda bahwa ekosistem lingkungan air telah tercemar dapat digolongkan menjadi beberapa pengamatan yaitu :

1. Pengamatan secara fisis yaitu pengamatan pencemaran air berdasarkan tingkat kejernihan air (kekeruhan), perubahan suhu, warna, dan adanya perubahan warna, bau dan rasa.
2. Pengamatan secara kimiawi yaitu pengamatan pencemaran air berdasarkan zat kimia yang terlarut, perubahan pH.
3. Pengamatan secara biologis, yaitu pengamatan pencemaran air berdasarkan microorganism yang ada di dalam air, terutama ada tidaknya bakteri pathogen.

Indikator yang sering kali digunakan dalam pemeriksaan pencemaran air adalah pH atau konsentrasi ion hydrogen, oksigen terlarut (*Dissolved Oxygen*, DO), kebutuhan oksigen biokimia (*Biochemical Oxygen Demand*, BOD) serta kebutuhan oksigen kimiawi (*Chemical Oxygen Demand*, COD). Namun pada penelitian yang digunakan penulis untuk pemeriksaan pencemaran air menggunakan beberapa parameter yang terdapat pada sensor air YSI 600R yaitu pH, DO, suhu, salinitas. Sebagai tambahan dalam penelitian ini, penulis menggunakan beberapa sumber data air baku PDAM, yaitu: Sensor air PDAM dengan masa percobaan dari Maret hingga Agustus 2016 Data laporan laboratorium PDAM mulai tahun 2014 hingga Oktober 2016. Untuk sensor air PDAM menggunakan dua area, yaitu Ngagel dan Karang Pilang dengan parameter input yang digunakan pada sensor ngagel yaitu suhu, NTU, TSS, pH, dan DO sedangkan pada sensor karang pilang yaitu NTU saja. Sedangkan Data laporan laboratorium PDAM mulai tahun 2014 hingga Oktober 2016, menggunakan data laboratorium dari Ngagel dan Karang Pilang yang hanya menggunakan 5 parameter dari 20 parameter yang ada, yaitu suhu, NTU, TSS, pH, dan DO.

Berikut penjelasan untuk masing-masing parameter yang akan digunakan peneliti dalam penelitian ini :

1. Pengukuran pH, merupakan pengukuran kosentrasi ion Hidrogen pada air sungai. Pada range pH antara 6.5 – 8.5 menunjukkan kondisi alami sungai dalam keadaan belum tercemar. Kondisi sungai yang mengalami pencemaran air, apabila kondisi pH yang dihasilkan lebih rendah dari 6.5 atau lebih tinggi dari 8.5. pH dikatakan dalam kondisi

asam apabila nilainya kurang dari 7. Sedangkan pH dikatakan dalam kondisi basa bila nilainya lebih dari 7. Perubahan kondisi air menjadi lebih asam, apabila terdapat kandungan bahan-bahan organik. Begitu pula sebaliknya, kondisi air bisa menjadi basa apabila terdapat kandungan kapur. Penyebab terjadinya perubahan pada kondisi air sangat tergantung pada kandungan bahan pencemarnya. Terjadinya perubahan nilai pH air memiliki peranan yang penting bagi organisme air, sehingga nilai pH yang terlalu basa ataupun terlalu asam dapat menyebabkan pencemaran.

2. DO, merupakan oksigen yang terlarut dalam air. Nilai DO biasanya diukur untuk menunjukkan jumlah oksigen dalam air. Semakin besar nilai DO pada airnya mengindikasikan air tersebut memiliki kualitas yang bagus. Sebaliknya jika nilai DO rendah, dapat diketahui bahwa air tersebut telah tercemar. Tanpa adanya oksigen yang terkandung didalamnya maka banyak microorganism dalam air yang tidak dapat hidup karena oksigen yang terlarut digunakan untuk proses degredasi senyawa organic dalam air. Oksigen terlarut akan menurun apabila banyak limbah, terutama limbah organik yang masuk ke sistem perairan.
3. Suhu, digunakan sebagai indikator dikarenakan perubahan kecil pada temperature dapat mempercepat proses biologis tumbuhan dan hewan bahkan menentukan tingkat kejenuhan oksigen yang terkandung didalamnya. Suhu air yang relative tinggi ditandai munculnya ikan-ikan dan hewan air ke permukaan untuk mencari oksigen. Dalam pengukuran DO tanpa adanya indikator suhu airnya maka kurang berguna, karena penurunan tingkat oksigen dihitung dari perbedaan tingkat kejenuhan dan DO terukur tidak dapat ditentukan karena suhu air tidak diketahui.
4. TSS, residu tersuspensi (*Total Suspended Solid*) adalah semua zat padat (pasir, lumpur, dan tanah liat) atau partikel-partikel yang tersuspensi dalam air dan dapat berupa komponen hidup (biotik) seperti fitoplankton, zooplankton, bakteri, fungi, ataupun komponen

mati (abiotik) seperti detritus dan partikel-partikel anorganik. Zat padat tersuspensi merupakan tempat berlangsungnya reaksi-reaksi kimia yang heterogen, dan berfungsi sebagai bahan pembentuk endapan yang paling awal dan dapat menghalangi kemampuan produksi zat organik di suatu perairan. Penetrasi cahaya matahari ke permukaan dan bagian yang lebih dalam tidak berlangsung efektif akibat terhalang oleh zat padat tersuspensi, sehingga fotosintesis tidak berlangsung sempurna. Sebaran zat padat tersuspensi di laut antara lain dipengaruhi oleh masukan yang berasal dari darat melalui aliran sungai, ataupun dari udara dan perpindahan karena resuspensi endapan akibat pengikisan.

5. Salinitas, adalah tingkat keasinan atau kadar garam terlarut dalam air. Salinitas juga mengacu pada kandungan garam dalam tanah.
6. NTU, Kekeruhan menggambarkan sifat optik air yang ditentukan berdasarkan banyaknya penyerapan cahaya dan pancaran cahaya yang berasal dari bahan-bahan yang terdapat di dalam air. Kekeruhan disebabkan oleh adanya bahan organik dan anorganik baik yang tersuspensi ataupun yang terlarut maupun bahan anorganik dan organik yang berupa plankton dan mikroorganisme lainnya. Pada suatu perairan, zat anorganik yang menyebabkan kekeruhan dapat berasal dari pelapukan batuan dan logam, sedangkan zat organik berasal dari lapukan hewan dan tumbuhan. Sedangkan untuk bakteri, dapat dikategorikan sebagai materi *organic* tersuspensi yang menambah kekeruhan air. Satuan kekeruhan dalam pengukuran nilai kekeruhan dengan pengukuran nephelometer dinyatakan dalam NTU (*Nephelometric Turbidity Unit*). Pengukuran dengan nephelometer sering digunakan karena ketepatan, sensitifitas dan dapat digunakan dalam rentang turbiditas yang besar. Pada penelitian ini, untuk nilai baku kekeruhan yang digunakan berdasarkan Peraturan Menteri Kesehatan Nomor: 492/Menkes/Per/IV/2010 pada tanggal 19 April 2010, dimana untuk nilai baku yang disarankan pada kekeruhan sebesar 5[17].

2.2.2 Sungai Kali Surabaya

Bagi masyarakat Surabaya sumber air yang memiliki peranan penting sebagai sumber air baku dan dikonsumsi masyarakat sebagian besar berasal dari air sungai. Salah satunya adalah sungai Kali Surabaya. Hampir lebih dari 90% sumber air baku berasal dari Kali Surabaya. Namun pada kenyataannya, keluhan mengenai kualitas air yang diperoleh masyarakat tidak semuanya masuk kategori layak untuk dikonsumsi. Hal ini dipengaruhi oleh banyak faktor yang menyebabkan permasalahan pencemaran air baik dari limbah rumah tangga, limbah pabrik, sampah yang dibuang ke sungai secara langsung dan kurang kesadaran masyarakat sekitar untuk menjaga ekosistem sungai.

Kali Surabaya merupakan percabangan dari Kali Brantas ini memiliki panjang kira-kira 41 km dan mengalir dari kota Mojokerto ke kota Surabaya yang memiliki potensi air tawar yang cukup besar. Namun, saat ini berbagai banyak indikasi tekanan yang berlebihan terhadap ekosistem kali Surabaya, dimana disekitar lokasi Kali Surabaya terdapat ratusan industry berskala kecil sampai berskala besar padahal sumber air baku PDAM ini memberikan manfaatnya sekitar lebih dari tiga juta penduduk Surabaya. (Sumber: BLH Surabaya).

2.2.3 Metode Indeks Pencemaran

Indeks pencemaran (*Pollution Index*) digunakan untuk menentukan tingkat pencemaran relatif terhadap parameter yang diijinkan[15]. Indeks Pencemaran (IP) memiliki konsep yang berlainan dengan Indeks Kualitas Air (*Water Quality Index*). Indeks Pencemaran (IP) ditentukan untuk suatu peruntukan, kemudian dapat dikembangkan untuk beberapa peruntukan bagi seluruh bagian badan air atau sebagian dari suatu sungai.

Perhitungan penentuan status mutu air menggunakan metode Indeks Pencemaran diatur pada Keputusan Menteri Negara Lingkungan Hidup No 115 Tahun 2003 tentang Pedoman Penentuan Status Mutu Air[18].

Pengelolaan kualitas air atas dasar Indeks Pencemaran (IP) dapat memberikan masukan pada pengambil keputusan agar dapat menilai kualitas

badan air untuk suatu peruntukan. Kemudian melakukan tindakan untuk memperbaiki kualitas jika terjadi penurunan kualitas akibat kehadiran senyawa pencemar.

Jika L_{ij} merupakan konsentrasi parameter kualitas air yang dicantumkan dalam baku peruntukan air (j), dan C_i dinyatakan sebagai konsentrasi parameter kualitas air (i) yang diperoleh dari hasil analisis dari suatu lokasi pengambilan dari suatu alur sungai, maka IP_j adalah Indeks Pencemaran bagi peruntukan (j) yang merupakan fungsi dari $\frac{C_i}{L_{ij}}$ dan ditentukan dari resultan nilai maksimum (M) dan nilai rerata (R) rasio konsentrai perparamter terhadap nilai baku mutunya. Metode Indeks Pencemaran [17] dapat dihitung dengan:

$$IP_j = \sqrt{\frac{(C_i/L_{ij})_M^2 + (C_i/L_{ij})_R^2}{2}} \qquad (2.1)$$

Evaluasi terhadap nilai IP_j dapat menentukan kategori kelas Indeks Pencemaran:

1) $0 \leq IP \leq 1{,}0$, memenuhi baku mutu (kondisi baik)
2) $1{,}0 \leq IP \leq 5{,}0$, tercemar ringan
3) $5{,}0 \leq IP \leq 10$, tercemar sedang
4) $IP \geq 10$, tercemar berat

Pada penelitian ini, penulis menggunakan pembagian kelas berdasarkan pada hasil perhitungan Indeks Pencemaran. Implementasi kelas Indeks Pencemaran dapat dilihat pada Tabel 2.1.

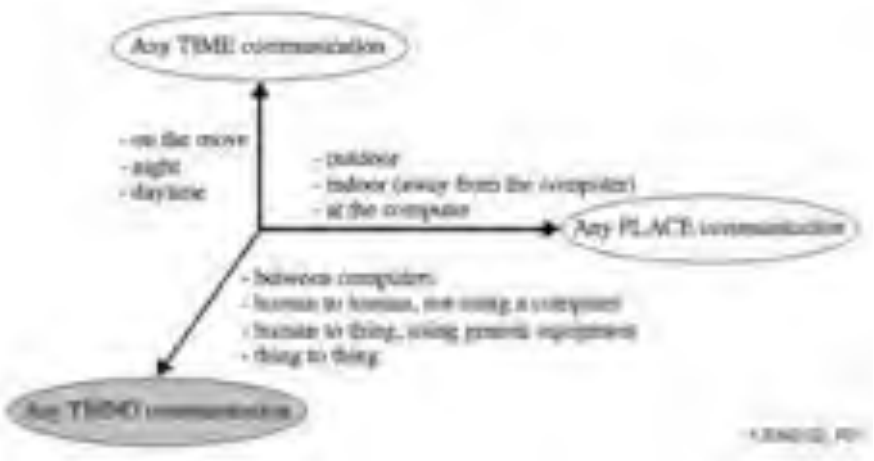
Tabel 2.1. Indeks Pencemaran.

Kelas	Kondisi	*Score*	Keterangan	Label pada SVM (output)
1	Baik Sekali	$0 \leq IP \leq 1{,}0$	Memenuhi baku mutu (kondisi baik)	0
2	Baik	$1{,}0 \leq IP \leq 5{,}0$	Tercemar ringan	1
3	Sedang	$5{,}0 \leq IP \leq 10$	Tercemar sedang	2

4	Buruk	$IP \geq 10$	Tercemar berat	3

2.3 Internet of Things

Perkembangan *Internet of Things* (IoT), tidak lepas dari konsep dasar-nya yaitu untuk menghubungkan objek yang satu dengan objek yang lainnya (*Things*) secara bersama-sama, sehingga memungkinkan untuk saling berkomunikasi satu sama lain. Menurut definisi[19], Internet of Things digambarkan sebagai *" ...a global insfrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies(ICT)"*. Dengan penerapan konsep teknologi IoT dengan interkoneksi antara dunia fisik dengan dunia maya, baik melalui eksploitasi identifikasi, pengambilan data, pengolahan data, dan kemampuan dalam berkomunikasi membuka peluang baru dalam dimensi IoT untuk mengakses apapun, setiap saat dan dari tempat manapun. Seperti yang digambarkan pada Gambar 2.1 mengenai dimensi *Internet of Things* menurut ITU-T Y.2060.



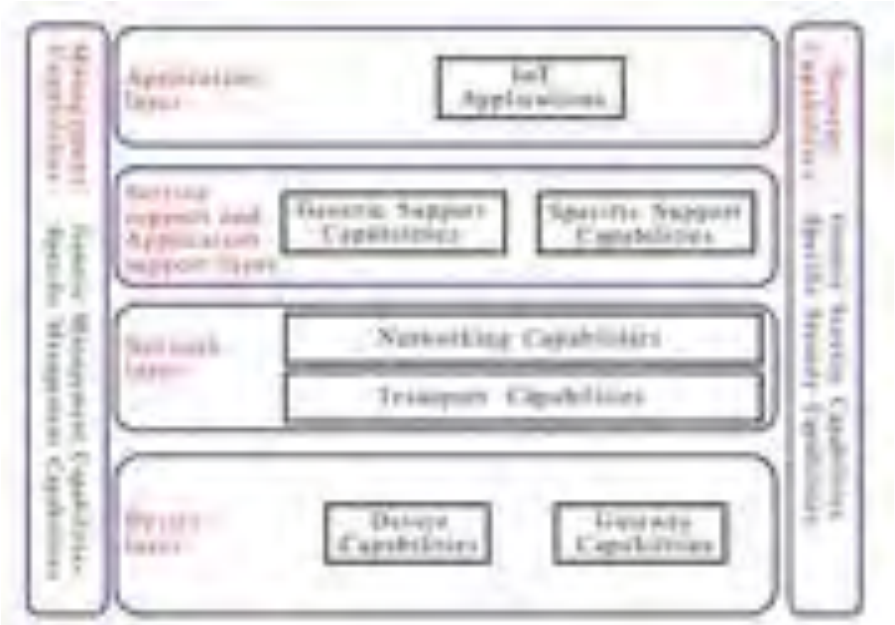
Gambar 2.1 Dimensi *Internet of Things* menurut ITU-T Y.2060.

Berikut adalah karakteristik dasar dari IoT, yang memberikan gambaran yang lebih jelas mengenai Internet of Things :

a. Interkonektivitas, di dalam IoT, apapun (*things*) bisa saling berhubungan dengan informasi global dan juga infrastruktur komunikasi.

b. Layanan yang berkaitan pada objek (*things*), dimana IoT mampu memberikan layanan terkait dengan objek-objek yang saling berhubungan termasuk dalam hal keterbatasan seperti perlindungan privasi maupun kosistensi semantik baik teknologi informasi di dunia fisik maupun dunia maya.
c. Heterogenitas, meliputi interaksi perangkat IoT yang heterogen karena berdasarkan platform hardware yang berbeda dan jaringan yang digunakan.
d. Perubahan dinamis, dimana keadaan perangkat berubah secara dinamis misalkan saat perangkat terhubung atau terputus maupun kondisi "*sleep*" serta kondisi "*waking up*". Selain itu, dapat ditinjau dari jumlah perangkat dapat berubah secara dinamis.
e. Skala "*Enormous*": Jumlah perangkat yang perlu dikelola dan berkomunikasi satu sama lainnya dapat berkembang lebih besar dengan perangkat yang terhubung ke internet saat ini. Bahkan yang lebih penting, pengelolaan data yang dihasilkan dan interpretasi untuk tujuan aplikasi.

ITU-T Y 2060, juga telah mendefinisikan model referensi untuk IoT, seperti yang digambarkan pada Gambar 2.2 mengenai model referensi dari IoT.



Gambar 2.2. Model Referensi dari Internet of Things menurut ITU-T Y.2060.

Model referensi dari IoT terbagi menjadi empat lapisan layer, antara lain:

1. Layer Aplikasi (*Application Layer*)

   Pada layer aplikasi, semua aplikasi IoT berada dalam layer ini.

2. Layer Layanan dan Pendukung Aplikasi (*Service Support and Application Support Layer*)

   Pada layer layanan dan pendukung aplikasi, terdiri dari dua kemampuan yaitu:

   - Kemampuan dukungan generic, merupakan kemampuan umum yang dapat digunakan oleh aplikasi IoT, contohnya untuk pengolahan data atau penyimpanan data.
   - Kemampuan dukungan khusus, merupakan kemampuan tertentu selain kemampuan generic, dimana kemampuan ini yang dibutuhkan untuk membuat dukungan untuk aplikasi yang beragam.

3. Layer Jaringan (*Network Layer*)

   Pada layer jaringan, terbagi menjadi dua bagian yaitu:

   - Kemampuan membangun jaringan, didalamnya memyediakan fungsi kontrol yang relevan untuk konektivitas jaringan, seperti akses dan control kemampuan transportasi, managemen mobilitas atau otentikasi, otorisasi dan akuntansi (AAA).
   - Kemampuan Transportasi, pada kemampuan ini focus pada penyediaan konektivitas untuk pengankutan layanan IoT dan aplikasi informasi data khusus, serta transportasi kontrol IoT yang meliputi kontrol terkait dan manajemen informasi.

4. Layer Perangkat *(Device Layer*)

   Pada layer perangkat, mencakup dua kemampuan yaitu:

   - Kemampuan Perangkat, yaitu kemampuan yang termasuk dalam:
     - Interaksi langsung dan tidak langsung dengan jaringan komunikasi.
     - Jaringan Ad-hoc, perangkat yang dapat membangun jaringan secara ad-hoc yang diperlukan dalam peningkatan skalabilitas dan penyebaran yang cepat

- Kondisi "*Sleep*" dan "*Waking up"*, kemampuan perangkat yang mendukung kondisi tertentu sebagai mekanisme dalam penghematan energi.

- Kemampuan *Gateway* yaitu kemampuan yang termasuk pada:
  - Dukungan beberapa interface, yang meliputi dukungan perangkat yang terhubung melalui berbagai jenis teknologi kabel dan nirkabel, seperti ZigBee, Wi-Fi, Bluetooth, maupun *Controller Area Network*(CAN). Sehingga, pada layer Jaringan, *gateway* memiliki kemampuan untuk dapat berkomunikasi dengan berbagai teknologi seperti PSTN (*Public Swithed Telephone Network*), generasi 2G atau 3G, jaringan LTE (*Long-Term Evolution*), Ethernet atau DSL (*Digital Subscriber Lines*).
  - Konversi Protokol, dalam beberapa situasi konversi protocol diperlukan untuk mendukung komunikasi antar perangkat yang menggunakan protocol yang berbeda pada layer perangkat dan layer jaringan, misalkan sebuah protocol teknologi *ZigBee* pada lapisan perangkat, dan teknologi 3G pada lapisan jaringan.

Selain empat lapisan layer yang telah disebutkan diatas, terdapat dua tambahan kemampuan pendukung dalam IoT yaitu

1. Kemampuan Management

   Kemampuan management IoT, meliputi kesalahan tradisional (*traditional fault*), konfigurasi, *accounting*, performansi dan keamanan (contohnya kesalahan manajemen, konfigurasi manajemen, akutansi manajemen, performansi dan keamanan manajemen.

2. Kemampuan Keamanan

   Dalam kemampuan keamanan, terbagi keamanan generik dan keamanan khusus, dimana pada setiap layer dijelaskan sebagai berikut:

- Pada layer aplikasi, meliputi otorisasi, otentifikasi, aplikasi kerahasiaan data dan perlindungan integritas, perlindungan privasi, audit keamanan dan anti-virus.
- Pada layer jaringan, meliputi otorisasi, otentikasi, penggunaan data, dan data yang menandakan kerahasiaan serta sinyal perlindungan integritas.
- Pada layer perangkat, meliputi otentikasi, otorisasi, kontrol akses, kerahasiaan data, dan perlindungan integritas.

Untuk kemampuan keamanan dapat digabungkan dengan kebutuhan aplikasi tertentu, misalkan *mobile payment*, ataupun persyaratan keamanan.

Pada penelitian ini mengambil konsep teknologi *Internet of Things*, pengembangan dari teknologi internet dimana setiap benda memiliki koneksi jaringan, dan mampu mengirimkan serta menerima suatu data. Komponen dari teknologi *Internet of Things* yang digunakan pada penelitian ini antara lain : 1) sensor, 2) sistem benam, 3) komunikasi data dan 4) pengolahan *Big Data*.

2.3.1 Sensor

Sensor air merupakan alat yang digunakan untuk mendeteksi kandungan kualitas air dalam suatu perairan. Sensor air yang digunakan yaitu YSI 600R[20] dan WTW IQ SensorNet 2020 XT[21].

2.3.1.1 YSI 600R

Untuk gambar sensor YSI 600R terlihat pada Gambar 2.3. Sedangkan untuk spesifikasi pada sensor YSI 600R dijelaskan pada Tabel 2.2.

Gambar 2.3 Sensor Kualitas Air YSI 600R

Tabel 2.2 Spesifikasi Sensor Kualitas Air YSI 600R

Spesifikasi	
Dimensi	Diameter=4,2cm Panjang=35,6cm Berat=0,5kg
Komunikasi	RS-232, SDI-12
Power	12V eksternal
Sensor	
Dissolved Oxygen %Saturasi	*Range*=0-500% Resolusi=0,1% Akurasi=0-200%±2%,200-500±6%
Dissolved Oxygen mg/L	*Range*=0-50mg/L Resolusi=0,01mg/L Akurasi=0-20mg/L±0,2mg/L, 20-50mg/L±6mg/L
pH	*Range*=0-14 unit Resolusi=0,01unit Akurasi=±0,2unit
Suhu	*Range*=-5 s/d 50 C Resolusi=0,01 C Akurasi=0,15 C

Untuk menghubungkan sensor dengan sistem benam menggunakan konektor USB-to-Serial. Dengan dihubungkan ke sistem benam sensor ini dapat mengirimkan data menuju ke data center, karena mampu mengirim data dengan sendirinya kita menyebut sensor ini sebagai sensor aktif.

2.3.1.2 WTW IQ SensorNet 2020 XT

PDAM telah menggunakan sensor WTW IQ SensorNet 2020 XT untuk memonitoring kualitas air di reservoir air Ngagel dan Karang Pilang. Bentuk dari sensor WTW IQ SensorNet 2020 XT dapat dilihat pada Gambar 2.4.

Sensor ini memiliki sensor modular, dimana main sensor bisa dipasangkan dengan beberapa sensor air seperti pH, NTU, chlor dan lainnya. Sensor ini sudah mendukung teknologi Internet Protokol, sehingga dapat diakses dari jarak jauh. Sensor ini juga memiliki internal web informasi yang menampilkan hasil dari sensor yang ada didalamnya. Namun untuk dapat mengambil data dari sensor WTW IQ SensorNet ini perlu dilakukan dengan melakukan web scrapping. Untuk model sensor seperti ini kita menyebutnya sebagai sensor pasif.

Gambar 2.4. Sensor WTW IQ SensorNet 2020 XT.

2.3.1.3 USB-to-Serial

Kabel penghubung antara sensor dan sistem benam yang merubah koneksi dari port USB ke Serial Port DB9.

2.3.2 Sistem Benam (*Embended System*)

Sistem benam yang digunakan untuk penelitian ini adalah *Raspberry Pi* versi 3[22]. Dimana sistem benam ini memiliki gambar seperti yang terlihat pada Gambar 2.5. Sedangkan untuk spesifikasi *Raspberry Pi* 3 Model B dijelaskan pada Tabel 2.3.

Gambar 2.5 *Raspberry Pi* 3 Model B

Tabel 2.3 Spesifikasi *Raspberry Pi* 3 Model B.

Spesifikasi	
Prosessor	A 1.2GHz quad core ARMv8
RAM	1GB
Catu Daya	5V
Sistem Operasi	Raspbian
Bahasa pemrograman	Python
Interface	
USB	4 port
GPIO	40 pin (UART)
HDMI	1
Ethernet	100baseTx
Audio	3.5mm jack output
Camera Interface (CSI)	1
Display Interface	1
Micro SD Slot	1
VGA	VideoCore IV 3D

Untuk mengambil data sensor yang terhubung dengan RS232 digunakan pemrograman data serial dengan python. Library yang digunakan adalah pySerial.

2.3.2.1 Python Serial (pySerial)

Pada penelitian ini untuk mengambil data dari sensor YSI 600R yang terhubung melalui port serial RS-232 digunakan bahasa pemrograman Python[23]. Bahasa pemrograman python memiliki modul tersendiri untuk mengambil data melalui port serial yaitu pySerial.

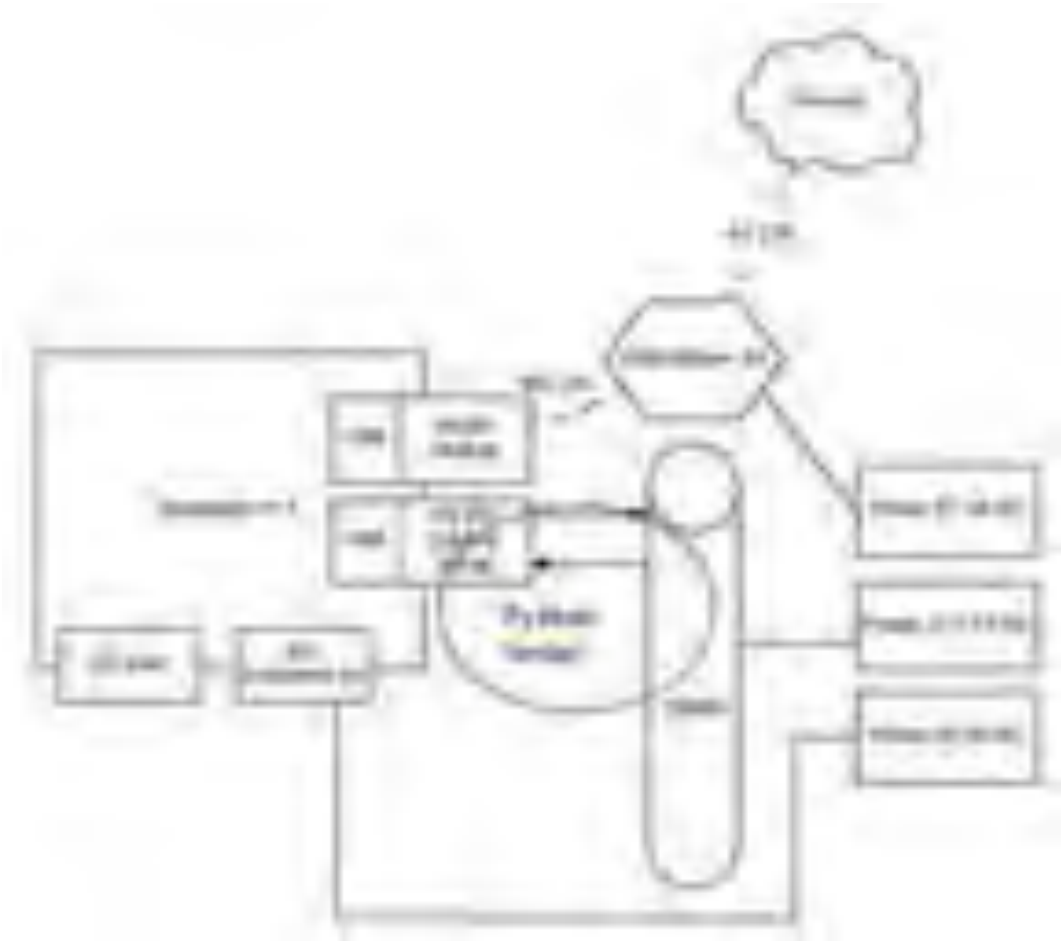
Penggunaan pySerial mengikuti parameter komunikasi serial antara lain 8 bit, Non-parity dan 1 stop bit atau biasa disingkat dengan 8N1. Kecepatan data transfer yang digunakan pada penelitian ini adalah 9600 bps. Pada sistem operasi Raspbian port serial diindentifikasi dengan /dev/ttyUSB0.

Pemrograman pySerial memanggil *module* dengan sintaks "*import serial*". Contoh inisialisasi serial port dengan pySerial dapat dilihat pada Kode Sumber 2.1.

```
	import serial
	ser = serial.Serial()
	ser.port = "/dev/ttyUSB0"
	#ser.port = "/dev/ttyS2"
	ser.baudrate = 9600
	ser.bytesize = serial.EIGHTBITS #number of bits per bytes
	ser.parity = serial.PARITY_NONE #set parity check: no parity
	ser.stopbits = serial.STOPBITS_ONE #number of stop bits
	#ser.timeout = None          #block read
	ser.timeout = 1            #non-block read
	#ser.timeout = 2              #timeout block read
	ser.xonxoff = False     #disable software flow control
	ser.rtscts = False         #disable hardware (RTS/CTS) flow
control
	ser.dsrdtr = False          #disable hardware (DSR/DTR) flow
control
	ser.writeTimeout = 2     #timeout for write
```

Kode Sumber 2.1. Inisialisasi serial port dengan pySerial.

Dalam proses *read write* data dari sensor menggunakan python serial, seperti digambarkan pada Gambar 2.6.



Gambar 2.6 *Read Write* Data menggunakan *Python Serial*

Untuk pengambilan data dilakukan dengan sintaks yang dapat dilihat di Kode Sumber 2.2

```
while(len(response) <= 0):
  ser.write("data\r")
  response = ser.readline().strip('\r\n')
```

Kode Sumber 2.2. Pengambilan data dari sensor dengan pySerial.

Data sensor yang telah diambil, disimpan sementara didalam sistem benam dengan menggunakan *SQLite*.

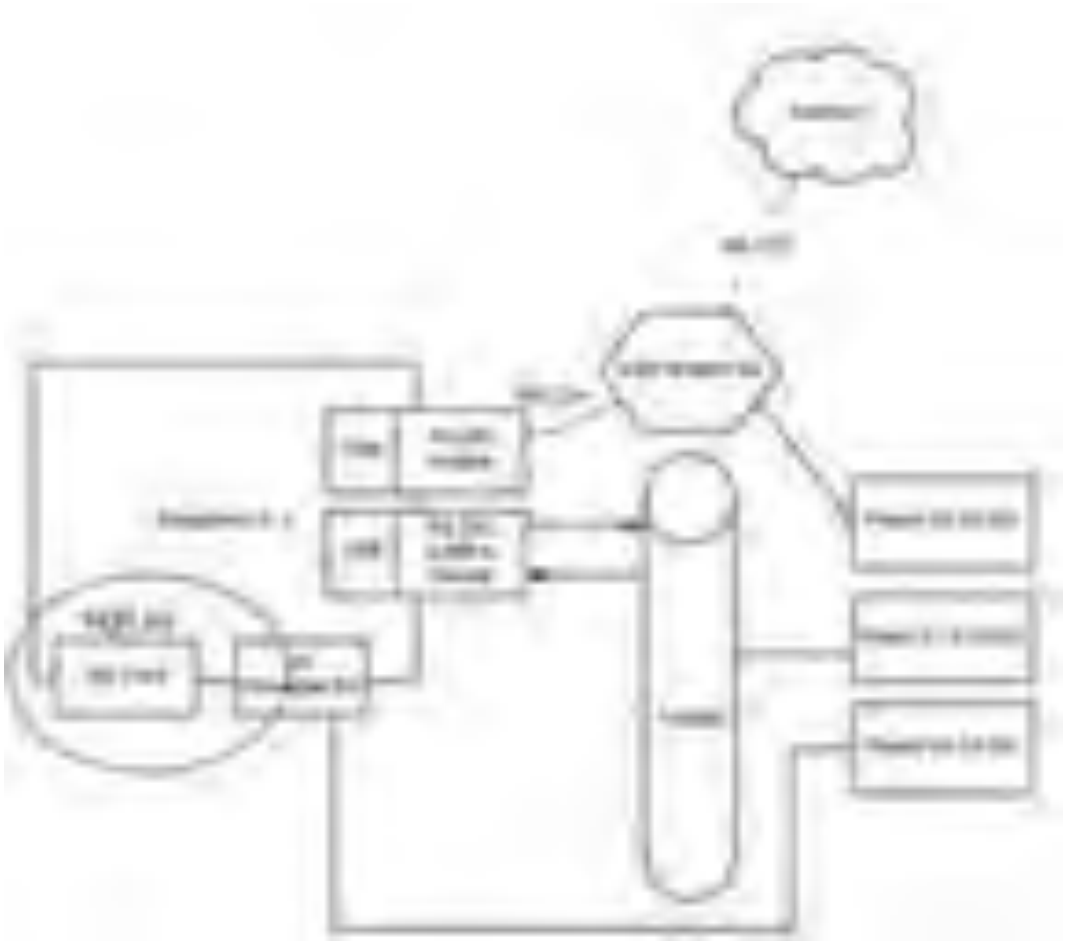
2.3.2.2 *SQLite*

Pada penelitian ini, untuk aplikasi data logging local penulis menggunakan aplikasi RDBMS dalam hal ini adalah *SQLite*[24]. *SQLite* merupakan aplikasi database yang menggunakan struktur bahasa SQL dalam

menyimpan informasi agar dapat diakses dengan mudah dan cepat. *SQLite* merupakan database relasional yang open source dan mendukung sebagian besar standar bahasa SQL.

Menurut Ramez Elmasri dan Shamkant B. Navanthe[25], pada dasarnya SQL atau yang sering disebut dengan "*Structured Query Languange*" merupakan bahasa standar dalam database yang biasanya digunakan untuk sistem manajemen database relasional yang komersial. Bahasa SQL sendiri memiliki kemampuan dalam pendefinisian data, query, serta update. Selain itu SQL bisa digunakan dalam menentukan batasan keamanan maupun otorisasi serta sangat mudah dalam menggabungkan statemen bahasa SQL ke dalam bahasa pemrograman lainnya, misalkan java atau python.

SQLite dikenal sebagai mesin database SQL sederhana yang tertanam dan memiliki keandalan tinggi serta cepat. Hal ini yang membuatnya berbeda dengan database SQL lainnya karena *SQLite* langsung dapat membaca dan menulis pada sebuah disk penyimpanan. Selain itu, dikarenakan *SQLite* merupakan sebuah library yang sangat praktis, sehingga *SQLite* dapat dijalankan cukup baik pada lingkungan dengan memori yang terbatas seperti pada *Raspberry Pi*, ponsel ataupun gadget lainnya yang menggunakan memori rendah. Pada penelitian ini, penulis menggunakan *SQLite* yang terinstall bersama sistem operasi raspbian yang digambarkan pada Gambar 2.7.



Gambar 2.7 *SQLite* yang terinstall pada *Raspberry Pi* 3

2.3.3 Komunikasi Data

Komunikasi data yang digunakan untuk menghubungkan perangkat *Internet of Things* dengan *environment Big Data* adalah komunikasi 4G LTE. Dimana perangkat yang akan digunakan adalah USB Modem 4G/LTE DT-100 Advance Jetz Plus Soft AP [26] seperti yang digambarkan pada Gambar 2.8. Sedangkan untuk spesifikasi USB Modem 4G/LTE DT-100 Plus + Soft AP dijelaskan pada Tabel 2.4.

Gambar 2.8 USB Modem 4G/LTE DT-100 Plus + Soft AP

Tabel 2.4 Spesifikasi USB Modem 4G/LTE DT-100 Plus + Soft AP

	Spesifikasi
.	USB Dongle Modem dan menggunakan USB 2.0 Plug and Play
.	Mendukung data jaringan LTE/DC-HSPA+/HSUPA/HSDPA/UMTS/EDGE/ GPRS
.	Mendukung internet data downlink dengan kecepatan sampai 100Mbps (4G/LTE/WIMAX) dan kecepatan upload sampai 50Mbps
.	Sudah terintegrasi Soft AP (Wifi *Tethering*)
.	MicroSD (TF-Card) Slot untuk penyimpanan data
.	Mendukung sampai 5 koneksi pengguna dalam waktu bersamaan
.	Mendukung OS PC Windows 2000/2003/XP/Vista/7/8/10 dan Mac OS X v10.4 keatas, serta Linux
sumber: https://alnect.net/product/7633/Page-Modem-GSM-4GLTE-Advance-Jetz-DT100-Plus-Soft-AP	

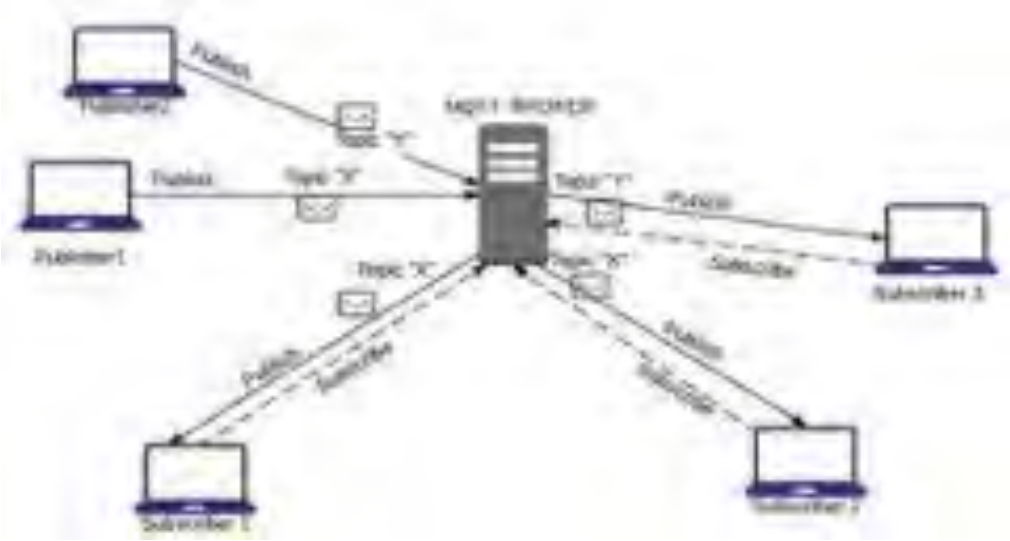
Data yang diambil dari sensor perlu dikirim ke data center melalui komunikasi data 4G menggunakan internet. Protokol yang digunakan untuk mengirimkan data sensor menuju ke data center menggunakan protokol MQTT.

2.3.3.1 MQTT

Untuk cara pentransferan data yang digunakan dalam sistem *Internet of Things* ini sehingga data yang ditransfer cukup efisien, menggunakan protokol MQTT (Message Queue Telemetry Transport Protocol)[27]. Protokol MQTT

adalah protokol jaringan yang menggunakan konsep publish/subscriber dalam pengiriman datanya, biasanya untuk pengiriman pesan antar perangkat "*Internet of Things*". Pada mekanismenya, konsep yang menggunakan cara publish/subscribe pesan dengan menerapkan topik yang sama didalamnya, sekilas prinsipnya menyerupai client-server. MQTT ini sebenarnya diterapkan pada lintas stack protocol TCP/IP yang memiliki ukuran paket data dengan header-nya lebih kecil sehingga sumber daya yang diperlukan relatif kecil. Header pada MQTT lebih sederhana dibandingkan protokol TCP/IP pada penggunaan HTTP. Jenis data yang dikirimkan menggunakan protokol MQTT ini bisa berupa data binary, text, bahkan XML. Salah satu platform yang mengimplementasikan MQTT adalah Mosquitto, platform ini yang nantinya digunakan sebagai MQTT broker.

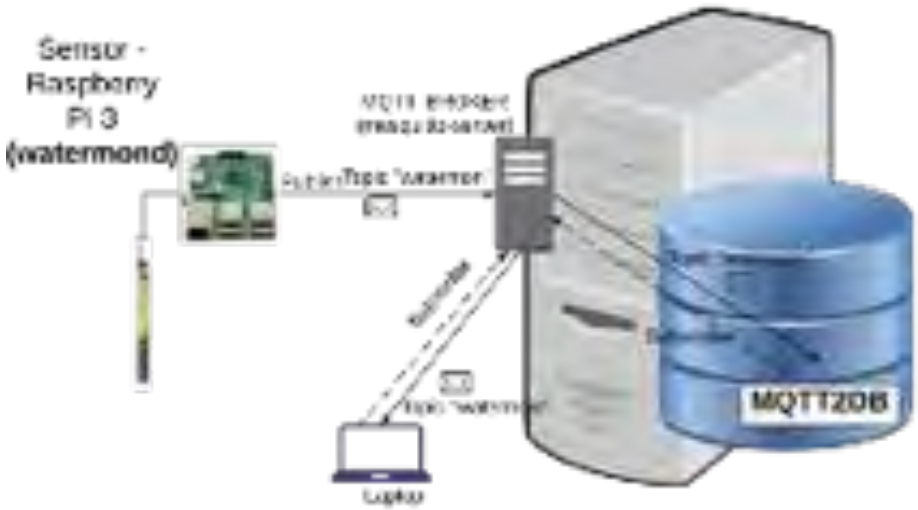
MQTT mengunakan topik dalam mem-publish ataupun men-subscribe pada pengiriman pesan. Dikarenakan terhubung pada berbagai perangkat, maka komunikasi yang terjadi antar perangkat dilakukan dengan pengiriman pesan, dimana setiap pesan selalu memiliki topik yang nantinya digunakan sebagai kata kunci yang berupa string atau authentifikasi user, layaknya *password*. Sama halnya dengan paradigma client dan server, dalam komunikasi di dalam jaringan pada MQTT digunakan istilah MQTT Publiser dan MQTT Subscriber sebagai perangkat-perangkat yang ingin berkomunikasi satu sama lain, istilah dalam jaringan sebagai client nya. Sedangkan MQTT Broker bertindak sebagai pihak yang mengatur dan meneruskan pesan-pesan yang diterima bahkan bisa mempertahankan pesan pada setiap topik yang dikirimkan oleh MQTT Publiser[28]. Alur pengiriman pesan topik pada MQTT digambarkan pada Gambar 2.9.



Gambar 2.9 Alur Pengiriman Pesan Topik pada MQTT

Pada Gambar 2.9 menjelaskan alur pengiriman pesan topik pada MQTT, Pada gambar diatas terlihat *Publisher1* ketika akan mengirimkan pesan tertentu menggunakan *topic* "X" dan ketika ada permintaan dari Subscriber untuk mengakses pesan pada *Publisher1* harus mengetahui topik tertentu dalam hal ini topic "X" sehingga yang mendapatkan pesan dari *publisher1* hanyalah Subscriber1 dan Subscriber2 yang memiliki topik sama yaitu topic "X" sedangkan untuk subscriber yang tidak memiliki topic sama tidak mendapatkan akses kepada pesan tersebut.

Pada penelitian ini, penulis menggunakan alur pengiriman yang digambarkan pada Gambar 2.10.



Gambar 2.10. Alur Pengiriman pesan dengan protokol MQTT pada penelitian ini.

MQTT Publisher memiliki peran yang memberikan suatu pesan kepada topik tertentu. Pada penelitian ini, perangkat *Raspberry Pi* 3 berperan sebagai MQTT *Publisher*, yang akan memberikan pesan berupa data sensor air kepada subscriber apabila memiliki topik yang sama. Dalam menggunakan MQTT ini, *publisher* menggunakan library paho-python. Pada penelitian ini telah dibuat aplikasi yang dapat mengambil data sensor kemudian mengirimkan dengan protokol MQTT, aplikasi ini diberi nama "watermond".

MQTT Subscriber dianalogikan sebagai client yang men-*subscribe* suatu topik, sehingga ketika publiser mengirimkan pesan dengan topik tertentu misalkan *"Topic watermon"*, maka subscriber yang memiliki topik yang sama akan menerima pesan tersebut. Perangkat yang berperan sebagai subscriber pada penelitian ini adalah aplikasi "MQTT2DB" yang berfungsi menjembatani antara data MQTT untuk dapat dimasukkan kedalam DBMS..

MQTT Broker, memiliki fungsi sebagai pihak yang mengatur sekaligus sebagai perantara yang menghubungkan antara *publisher* dengan subscriber, yang selanjutnya akan meneruskan pesan dari *publisher* ke *subscriber*. Platform yang menggunakan protokol MQTT sangat banyak, namun yang digunakan sebagai MQTT Broker pada penelitian ini adalah *Mosquitto-server*.

Topic yang digunakan dalam MQTT penggunaannya hampir sama dengan pengiriman pesan pada chatting broadcast tetapi lebih sederhana dan

memiliki fungsi sebagai authentifikasi user dalam pengiriman pesan agar MQTT Broker dengan mudah menghubungkan dan meneruskan pesan dari publisher ke subscriber.

2.3.4 Pengolahan BigData

Menurut Min Cen, dkk, 2014. Secara umum *Big Data* dapat diartikan sebagai sekumpulan data yang ditinjau dari ukurannya yang sangat besar (*volume*), sangat cepat perkembangan/pertumbuhannya (*velocity*), data yang beragam dalam berbagai bentuk/format (*variety*), serta memiliki nilai tertentu (*value*)[8].

Dalam infrastruktur *Big Data*, data akuisisi menunjukkan acuan aliran data yang memiliki kecepatan tinggi dan ragam yang bervariasi, infrastruktur *Big Data* mendukung akuisisi data yang besar sehingga apabila diproses dengan cepat dan dalam lingkungan terdistribusi maka struktur data yang dinamis tersebut dapat menghasilkan prediksi yang lebih baik.

2.3.4.1 Hadoop

Hadoop [29] menjadi sebuah pilihan alternative dalam mengelola data yang cukup besar karena merupakan perangkat open source yang handal, terukur dan komputasi terdistribusi. Kepopuleran *Hadoop* dalam rentan beberapa tahun terakhir ini, menjadikan *Hadoop* sebagai model pemrograman sederhana pada distributed parallel computing dalam mengolah *dataset* yang besar. *Hadoop* merupakan implementasi *open source* yang mengadopsi dari *Google-MapReduce* kaarena dianggap memberikan banyak keuntungan yang signifikan dibandingkan *database parallel*. Sudah banyak perusahaan yang berhasil dalam mengembangkan perusahaannya melalui penggunaaan *Hadoop*. Secara gambaran umum *Apache Hadoop* adalah sebuah perangkat lunak *open source* yang memungkinkan pemrosesan *dataset* yang besar secara terdistribusi dalam *cluster* server. *Hadoop* dirancang untuk dapat bekerja pada server tunggal hingga banyak server, dengan toleransi kesalahan yang sangat tinggi.

Hadoop mendukung pembuatan *software open-source* yang menyediakan sebuah *framework* untuk membangun aplikasi *distributed computing* dengan skalabilitas yang tinggi. *Apache Hadoop* memiliki dua komponen utama yaitu:

1. *MapReduce*, framework yang membagi pekerjaan ke node-node dalam proses peng-clusteringan Hadoop. *MapReduce* merupakan model programing untuk pengolahan data. *MapReduce* dibangun menggunakan konsep *divide* and *conquer*, dengan cara membagi pekerjaan yang besar menjadi bagian-bagian kecil dan memprosesnya secara paralel. Penanganan pada *MapReduce* meliputi pendistribusian data, pengkomputasian secara paralel serta otomatisasi dalam penanganan kegagalan. Dua bagian utama dalam *MapReduce*, antara lain pertama, proses Map. Pada proses Map terjadi pembacaan data dari sekumpulan *record* pada sebuah input file, kemudian menjalankan filtering dan transformasi serta memberikan output berupa sekumpulan data rekam menengah ke dalam bentuk pasangan nilainya. Setelah itu nilai pada data rekam menengah menjadi inputan data pada proses kedua yaitu *Reduce*. Pada proses *Reduce* terjadi penerimaan data nilai rekam menengah sebagai input dan menggabungkan nilai-nilainya menjadi sebuah *key* yang utuh sehingga menjadi nilai ringkasan sebagai outputnya.
2. HDFS, istilah singkat dari *Hadoop Distributed Filesystem*. Dimana sebuah file system digunakan untuk menyimpan file yang besar dengan membagi-bagi dan menyimpan lalu mendistribusikan ke dalam banyak node yang salaung berhubungan. HDFS merupakan perancangan filesystem yang menggunakan *MapReduce* dalam pembacaan data, mengolah dan menuliskan output keluarannya. Untuk menjaga reliabilitas, data dalam HDFS akan diduplikasi ke dalam beberapa node. Dua bagian utama dalam HDFS yaitu *Namenode* dan *Datanode*. *Namenote* digunakan dalam pengaturan metadata *filesystem*, sedangkan *Datanode* merupakan blok penyimpanan.

Didalam *Apache Hadoop* terdapat beberapa aplikasi seperti yang digambarkan dalam *Hadoop* Ekosistem, dimana salah satu diantaranya mendukung fungsi Machine Learning yang terlihat pada Gambar 2.11.

Gambar 2.11. Hadoop Ekosistem dengan Machine Learning [30].

Setelah data disimpan tahap selanjutnya adalah penganalisaan data yang besar dan beraneka ragam pada lingkungan terdistribusi dengan menggunakan analisa data mining maupun analisa statistik. Ada dua jenis analisa dalam *Big Data* ditinjau dari pendekatan *machine learning* yaitu :

1. Analisa penggambaran data yang tersimpan di BigData dan menampilkan data tanpa memerlukan algoritma *Artificial Intelegence*.
2. Analisa dalam pencarian informasi yang tersembunyi dari data yang tersimpan di *Big Data*. Pada analisa ini, tidak hanya menampilkan data yang lama namun menampilkan data yang baru yang dihasilkan dari proses analisa *Big Data*, dimana sudah menggunakan *Artificial Intelegence* atau *Machine Learning*. Sifat umumnya biasanya diikuti adanya proses *training* untuk meningkatkan keakurasian data.

Big Data menawarkan berbagai solusi yang bias dikategorikan sebagai berikut :

1. *Social data analysis* yang dikembangkan dalam pemrosesan data dari social media, yang bisa dikembangkan untuk sentimen analisis.
2. *Historical data analysis*, misalkan pemrosesan analisa pada masa lalu sehingga bisa memberikan gambaran di masa lalu.
3. *Prediktif analysis*, penggabungan data historical analysis dengan kombinasi *Artificial Intelegence* dalam memprediksi kejadian yang akan datang sehingga tindakan antisipasi dapat dilakukan.

2.3.4.2 *Spark*

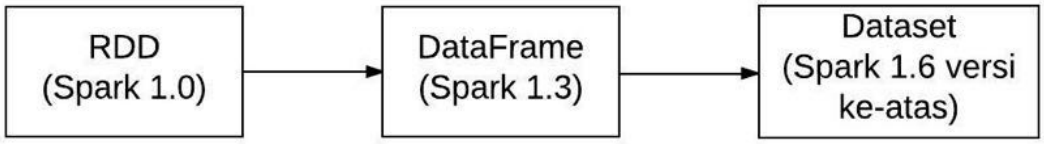
Apache Spark [31] adalah sebuah mesin pengolahan data hadoop memiliki fungsi analisis data yang kompleks dan fungsi machine learning serta algoritma grafik. *Apache Spark* bisa menjalankan program 100 kali lebih cepat dalam memori dan 10 kali lebih cepat pada disk dibandingkan *MapReduce*. *Apache Spark* dapat berjalan beriringan dengan sistem penyimpanan Hadoop dan HDFS. Aplikasi pada *Apache Spark* dapat dilihat pada Gambar 2.5.

Gambar 2.12 Apache *Spark*

Beberapa aplikasi yang bisa digunakan untuk menganalisa data dan pengimplementasian data mining menggunakan machine learning untuk beberapa proses klasifikasi, peng-cluster-an maupun untuk prediksi. Salah satunya framework *Big Data* adalah *Spark.*

2.3.4.3 Tipe Data Abstraksi pada *Spark*

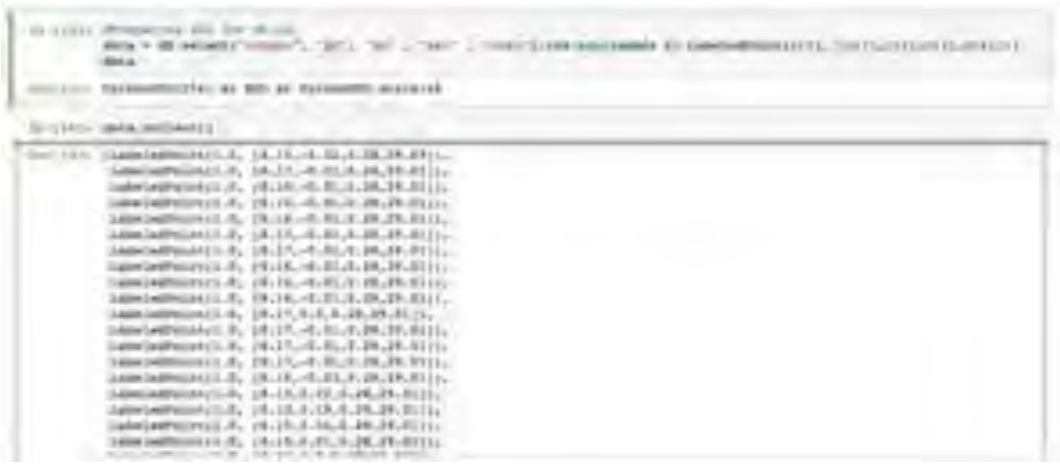
Pada *Spark* versi 2.0 terdapat 3 tipe data abstraksi yang disediakan *Spark* dan dapat digunakan, yaitu *RDD (Resilient Distribution Dataset)*, DataFrame dan *Dataset*. Evolusi tipe data ini dapat dilihat seperti yang digambarkan pada gambar



Gambar 2.13 Tipe data abstraksi pada Spark

1. RDD (Resilient Distribution Dataset)
2. DataFrame
3. Dataset

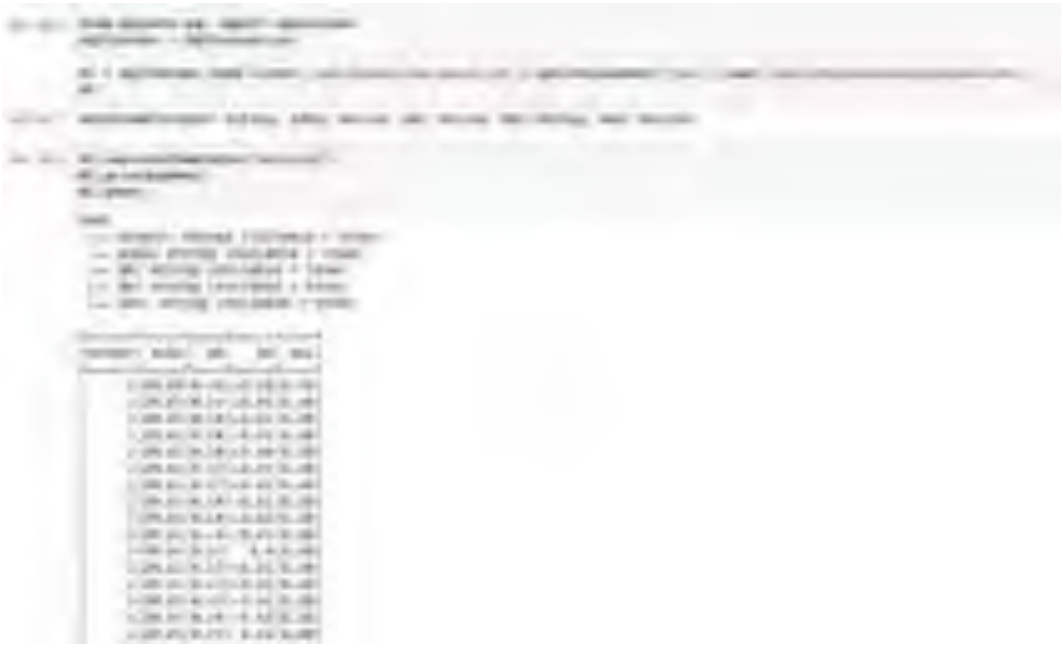
RDD merupakan tipe data utama yang digunakan *Spark*, dimana RDD ini digunakan untuk tipe data yang *unstructured*. RDD bersifat blok data, dimana untuk aplikasi *Spark* komputasi dilakukan dengan menggunakan RDD. RDD dapat diparalel dengan menggunakan fungsi lambda. RDD memiliki sifat simple dimana memiliki API dengan OOP untuk dapat mengakses data pada RDD. Namun RDD memiliki kelemahan berupa banyak data tambahan yang pada jumlah data yang besar bisa mengakibatkan sampah data pada objek memory JVM. Contoh penggunaan RDD dapat dilihat pada Gambar 2.14.



Gambar 2.14. Contoh penggunaan RDD.

DataFrame merupakan tipe data terstruktur yang digunakan *Spark* untuk dapat dianalisa. Dataframe pada *Spark* sudah mendukung SQL, dimana

pembacaan dataframe bisa lebih cepat dari RDD. Contoh penggunaan dataframe dapat dilihat pada Gambar 2.15.



Gambar 2.15. Contoh penggunaan DataFrame.

Dataset merupakan dataframe yang sudah di pilah menjadi suatu baris baru. *Spark* tidak menyediakan fungsi khusus untuk python dalam mengakses suatu *dataset,* namun struktur data python sudah dapat langsung mengakses dataframe sehingga dapat dibentuk menjadi *dataset*. Contoh penggunaan dataset dapat dilihat pada Gambar 2.16.

Gambar 2.16. Contoh penggunaan *Dataset*.

Spark menggunakan RDD mulai versi awal, pada tahun 2011 dikembangkan tipe data DataFrame yang mulai digunakan pada *Spark* versi 1.3, dan mulai *Spark* 1.6 dan terbaru dikenalkan tipe data *Dataset*.

2.3.4.4 *pySpark*

pySpark merupakan salah satu modul python yang digunakan untuk apache *Spark*. *PySpark* adalah Python API untuk aplikasi *Spark*. *pySpark* menggunakan bahasa pemrograman python dalam mengimplementasikan machine learning yang digunakan untuk pengklasifikasian.

Untuk penelitian ini *pySpark* diintegrasikan dengan *jupyter-notebook* untuk dapat menjalankan script python secara interaktif melalui user interface web.

2.3.4.5 *Jupyter* Notebook

Jupyter notebook [32] adalah aplikasi antar muka berbasis web yang dapat digunakan untuk membuat program analisis yang dapat diintegrasikan dengan *Spark*. *Jupyter* notebook juga dapat membuat program analisis yang menggunakan metode machine learning sehingga dapat memproduksi grafik analisa.

Jupyter notebook ini dapat diintegrasikan dengan library pemrograman python yang menyediakan metode machine learning.

2.3.4.6 MariaDB

MariaDB merupakan implementasi RDBMS versi opensource dari MySQL Enterprise. Pada penelitian ini menggunakan MariaDB untuk penyimpanan data. Data sensor yang diterima oleh MQTT2DB dimasukkan kedalam tabel MariaDB.

2.3.4.7 Library Python pendukung machine learning

Untuk dapat menggunakan machine learning, pada penelitian ini menggunakan pustaka bahasa pemrograman Python. Pustaka yang digunakan antara lain *Scikit-Learn*, SciPy, NumPy, dan matplotlib.

Scikit-Learn (http://www.scikit-learn.org) [33] adalah Python library yang mendukung algoritma machine learning. Dapat digunakan untuk analisa data dan data mining. Dibangun dengan integrasi dengan library Python lainnya seperti NumPy, SciPy, dan matplotlib. *Scikit-Learn* menggunakan LibSVM untuk menjalankan algoritma *Support Vector Machine* [34].

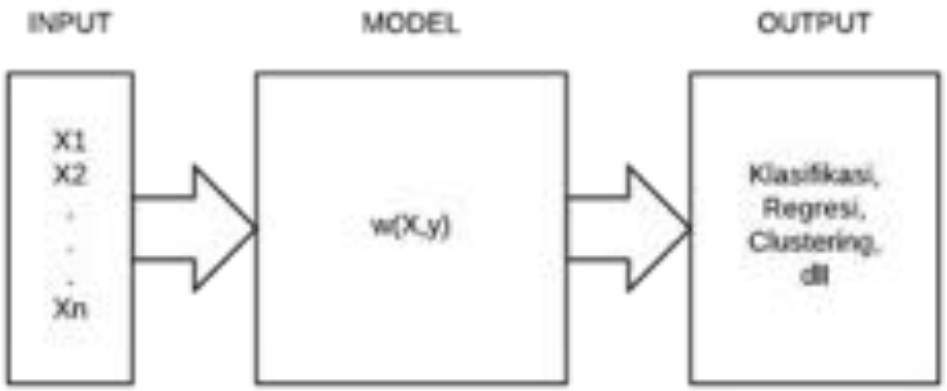
SciPy (http://www.scipy.org) [35] adalah ekosistem perangkat lunak python untuk matematika, sains dan enginering. Dimana paket SciPy terdiri dari NumPy, matplotlib, dan Panda.

NumPy (http://www.numpy.org) [36] adalah paket library python untuk komputasi ilmiah, dimana mampu mengolah N-dimensi objek array, penggunaan untuk aljabar linier, fourier transform, dan pembangkitan bilangan random.

Matplotlib (http://www.matplotlib.org) [37] adalah library python untuk ploting 2D dan menghasilkan hasil gambar dengan resolusi tinggi dan dapat digunakan bersama library python lainnya.

2.4 Machine Learning (secara umum)

Pembelajaran mesin (*Machine Learning*)[38] merupakan salah satu cabang dari kecerdasan buatan yang membahas mengenai pembangunan sistem yang didapat berdasarkan pada pembelajaran data, atau sebuah studi yang mempelajari cara untuk memprogram sebuah komputer untuk belajar. Inti dari pembelajaran mesin adalah representasi dan generalisasi.



Gambar 2.17. Machine Learning.

Pada Gambar 2.13 menunjukkan pembelajaran mesin secara umum, dimana Input merupakan representasi data dan pemilihan ekstraksi fitur-fitur tertentu, dimana sekumpulan fitur-fitur yang memberikan deskripsi sebuah objek dinyatakan dengan $X = [X1, X2, \dots, Xn]$. Dalam mengenali sebuah objek

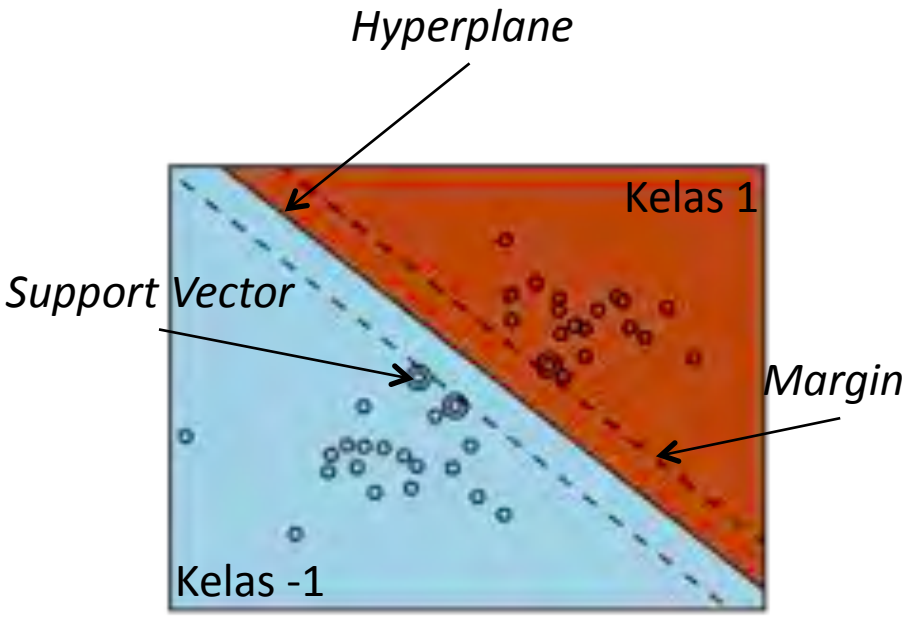
sehingga mendapatkan informasi data dari pembagian area fitur-fitur tersebut, maka diperlukan model $w = [X, y]$ tertentu yang digunakan dan berfungsi sebagai model pembelajaran mesin yang menggunakan algoritma-algoritma tertentu sehingga diperoleh Output yang berupa informasi dari data yang telah diproses, sehingga menghasilkan klasifikasi data, Regresi, ataupun pengklasteran data.

2.4.1 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) [39] adalah salah satu metode dalam *machine learning* yang sangat popular saat ini. *Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop* on *Computational Learning Theory*. Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane* oleh Vapnik tahun 1964, kernel diperkenalkan oleh *Aronszajn* tahun 1950, dan demikian juga dengan konsep-konsep pendukung yang lain. Akan tetapi hingga tahun 1992, belum pernah ada upaya merangkaikan komponen-komponen tersebut. Prinsip dasar SVM adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada problem *non-linear*, dengan memasukkan konsep kernel trick pada ruang kerja yang berdimensi tinggi.

Support Vector Machine (SVM) juga dikenal sebagai teknik pembelajaran mesin paling mutakhir setelah pembelajaran mesin sebelumnya yang dikenal *neural network* (NN). Kedua Metode tersebut menunjukkan keberhasilan dalam penggunaan pengklasifikasian dan pengenalan pola. Proses pembelajaran dilakukan dengan menggunakan pasangan data input dan data output sebagai sasaran yang diinginkan. Proses pembelajaran ini dikenal dengan *Supervised Learning*. Dari proses inilah akan diperoleh fungsi yang menggambarkan ketergantungan input dan outputnya. Selanjutnya, diharapkan fungsi yang dihasilkan dapat mengeneralisasi data dengan baik. Sehingga fungsi yang digunakan nantinya untuk data input diluar data pembelajaran.

Konsep SVM digambarkan secara sederhana sebagai upaya mencari *hyperplane* dengan menggunakan margin yang maksimal agar memberikan generalisasi yang lebih baik pada metode klasifikasi. *Hyperplane* berfungsi sebagai pemisah dua set data dari dua kelas yang berbeda. *Hyperplane* terbaik antara kedua kelas dapat ditemukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* tersebut dengan data terdekat dari masing-masing kelas. Data yang paling dekat ini disebut sebagai *Support Vector*. Hal ini dapat digambarkan pada Gambar 2.18 dimana terlihat lingkaran ganda. Usaha untuk mencari lokasi *hyperplane* merupakan inti dari proses pelatihan pada SVM. Contoh implementasi SVM dapat dilihat pada Gambar 2.18. SVM dengan *hyperplane* dan margin.



Gambar 2.18. SVM dengan *hyperplane* dan margin.

2.4.1.1 Support Vector Classification (SVC)

SVM yang digunakan untuk melakukan klasifikasi disebut dengan Support Vector Classification (SVC). Dimana untuk melakukan klasifikasi menggunakan persamaan matematika. Misalkan data yang tersedia dinotasikan (X_i, y_i) dengan $X_i = \{X_{i1}, X_{i2}, \dots, X_{iq}\}^T$ untuk data vektor $X \in R^p$ dalam dua kelas dan $y_i \in \{1, -1\}$ menyatakan label kelas yang akan digunakan. Dimana $i = 1,2, \dots, n$ dimana n adalah banyaknya data. Kelas positif dinotasikan sebagai 1 dan kelas

negatif dinotasikan dengan -1. Dengan asumsi bahwa kedua kelas ini dapat terpisah secara sempurna oleh *hyperplane* berdimensi p.

Hyperplane pada SVM[40], sebenarnya merupakan hyperplane linear yang hanya bekerja pada data yang dapat dipisahkan secara linear. Namun, untuk data yang distribusi kelasnya tidak linear biasanya menggunakan pendekatan kernel pada fitur data awal set data. SVM linear menggunakan hyperplane klasifikasi linear SVM dengan persamaan :

$$w.X_i + b = 0 \qquad (2.1)$$

dimana :

w dan b adalah parameter model

$w.X_i$ adalah inner-product dari vector w dan X_i

X_i merupakan atribut (fitur) set untuk data latih ke-i yang dinyatakan oleh (X_i, y_i) dengan $i = 1,2, \dots, n$ dan y_i menyatakan label kelas.

Data vektor X_i termasuk dalam kelas -1 adalah data yang memenuhi pertidaksamaan berikut:

$$w.X_i + b \leq -1 \qquad (2.2)$$

Data vektor X_i termasuk dalam kelas 1 adalah data yang memenuhi pertidaksamaan berikut:

$$w.X_i + b \geq +1 \qquad (2.3)$$

Klasifikasi kelas data pada SVM pada persamaan (2.2) dan (2.3) dapat digabungkan menjadi:

$$y_i(w.X_i + b) \geq 1, i = 1,2, \dots n \qquad (2.4)$$

Margin optimal dihitung dengan memaksimalkan jarak antara hyperplane dan data terdekat. Jarak maksimal dinotasikan dengan vektor bobot $\|w\|$. Selanjutnya masalah ini diformulasikan ke dalam problem *Quadratic Programming* (QP) dengan meminimalkan invers vektor bobot menjadi $\frac{1}{2}\|w\|^2$, dengan syarat seperti pada persamaan (2.4).

Dalam menyelesaikan permasalahan ini dapat diselesaikan dengan Lagrange multiplier:

$$Lp = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i y_i (w.X_i + b) - 1 \quad (2.5)$$

Dimana α_i adalah Lagrange multiplier yang berkorespondensi dengan x_i. Nilai α_i adalah nol atau positif. Persamaan (2.5) dapat dihitung dengan meminimalkan Lp terhadap vektor w dan b dengan memaksimalkan Lp terhadap α_i. Sehingga SVC mampu menyelesaikan dengan persamaan (2.6):

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \quad (2.6)$$

syarat: $y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i$

$$\zeta_i \geq 0, i = 1, \ldots, n$$

Parameter C berguna untuk mengontrol *trade-off* antara margin dan error klasifikasi. Apabila variabel w dan b tidak ada nilainya diperlukan variable slack ζ_i untuk menyelesaikan persamaan. Dimana untuk persamaan (2.6) digunakan untuk *primal problem*.

Dengan menurunkan persamaan *dual problem* pada data linier, maka *dual problem* untuk data non-linier didapat persamaan (2.7):

$$\min_{\alpha} \frac{1}{2} \alpha^T Q_\alpha - e^T \alpha \quad (2.7)$$

syarat: $y^T \alpha = 0$

$$0 \leq \alpha_i \leq C, i = 1, \ldots, n$$

Dimana e vektor keseluruhan, C > 0, batas atas, Q adalah n x n positif matrik, $Q_{ij} = y_i y_j K(x_i, x_j)$, dimana dalam pemetaan fungsi kernel ke dimensi baru maka *Dot-product* kedua buah vektor (x_i) dan (x_j) dinotasikan sebagai $\phi(x_i)^T \phi(x_j)$ sehingga $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, teknik komputasi ini dikenal dengan *kernel trick*.

Untuk persamaan prediksi pada dimensi fitur yang baru dapat dihitung dengan persamaan (2.8):

$$sgn(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x_j) + b) \quad (2.8)$$

Untuk pemodelan SVM pada thesis ini sudah menggunakan LibSVM.

2.4.1.2 Penerapan model kernel

SVM Non Linear menggunakan pendekatan kernel pada fitur datanya. Kernel didefinisikan sebagai suatu fungsi yang memetakan fitur data dari dimensi rendah (ruang input) ke dimensi yang lebih tinggi (ruang fitur). Contoh, suatu fungsi kernel k yang mana untuk semua vector input x, y akan memenuhi kondisi persamaan (2.2):

$$k(x, y) = \emptyset(x)^T \emptyset(y) \quad (2.2)$$

dimana :

$\emptyset(.)$ merupakan fungsi pemetaan dari ruang input ke ruang fitur.

Salah satu fungsi kernel yang banyak digunakan adalah Gaussian radial basis function (RBF), menggunakan persamaan (2.3):

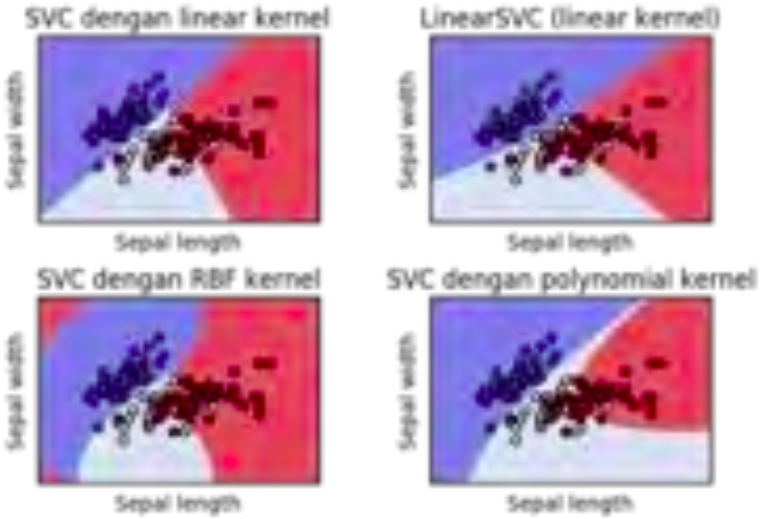
$$k(x, y) = \emptyset(||x - y||) = \exp\left[\frac{-\|x - y\|^2}{2.\sigma^2}\right] \quad (2.3)$$

Fungsi kernel polynomial dapat dilihat pada persamaan (2.4).

$$k(x, y) = (x.y + c)^d \quad (2.4)$$

Pada SVM untuk membentuk hyperplane dalam dilakukan dengan menentukan kernel yang digunakan. Kernel yang dapat digunakan untuk penelitian ini adalah: SVC dengan linier kernel, LinearSVC (linier kernel), SVC dengan RBF kernel, dan SVC dengan polynomial kernel.

Contoh penggunaan untuk masing-masing kernel dapat dilihat pada Gambar 2.19.



Gambar 2.19. Contoh penggunaan kernel pada SVM.

2.4.1.3 Proses klasifikasi

Klasifikasi yang digunakan pada penelitian ini menggunakan model klasifikasi SVM. Klasifikasi SVM termasuk dalam "Supervised Learning" dimana pada data pelatihan (*training*) disertai target, misalkan fungsi target y yang memetakan setiap data/vektor (set fitur) X ke dalam satu dari sejumlah label kelas output yang tersedia. Tujuan penggunaan model SVM ini adalah untuk membangun model yang dapat menghasilkan output yang benar untuk suatu data input dalam hal ini untuk penklasifikasi data baru dengan tepat.

Proses klasifikasi yang digunakan pada penelitian ini, digambarkan pada Gambar 2.15. Pada Gambar 2.15, menunjukan proses klasifikasi dimana proses learning dan *testing* dilakukan menggunakan input data. Proses learning merupakan proses pemilihan model dan penentuan parameter data berdasarkan data pelatihan (*training* data), misalkan (X_train, y_train). Sedangkan *testing* merupakan proses pengujian metode dengan menggunakan data penguji (*testing* data), misalkan $(X\_test, ?)$, sehingga diperoleh nilai estimasi untuk kapabilitas generalisasi dari model prediksi dari algoritma yang digunakan. Pada fit *training* , dilakukan proses pembangunan model dimana selama proses pelatihan diperlukan algoritma pelatihan salah satunya adalah SVM (*Support Vector Machine*). Pada akhirnya model yang dibangun pada saat pelatihan digunakan dalam memetakan

setiap vektor /data input ke label kelas dari data baru yang belum diketahui label kelasnya sehingga mendapatkan label kelas keluaran yang benar.

Gambar 2.20. Proses Klasifikasi.

2.4.1.4 Multi-Class Clasification

SVM hanya dapat melakukan klasifikasi biner (dua kelas), akan tetapi permasalahan di dunia nyata umumnya mempunyai banyak kelas. Sehingga membutuhkan klasifikasi lebih dari dua kelas. Oleh karena itu SVM sudah mendukung untuk melakukan klasifikasi dengan multi class. Metode yang digunakan untuk pendekatan multi-kelas salah satunya menggunakan pendekatan "one-against-one" (OAO).

Pendekatan OAO membentuk klasifikasi SVM biner sebanyak K(K-1)/2, dimana setiap klasifikator digunakan untuk membedakan di antara pasangan label kelas (y_i, y_j). Vektor yang tidak menjadi anggota kelas y_i ataupun y_j diabaikan ketika pembentukan klasifikator biner (y_i, y_j).

Implementasi SVC pada penelitian ini menggunakan pendekatan OAO.

BAB 3
METODOLOGI PENELITIAN

3.1 Desain Sistem

Dalam penelitian yang akan dilakukan ini memiliki desain sistem seperti yang digambarkan pada Gambar 3.1, lokasi pengambilan data pada sumber air sungai Kali Surabaya diarea sebelum intake pada pintu air PDAM KARANG PILANG.

Gambar 3.1. Desain sistem.

Pada desain sistem diatas menggabungkan teknologi *Internet of Things* yang terdiri dari sensor air YSI 600R, sistem benam *Raspberry Pi 3,* dan komunikasi 4G. Kemudian data tersebut dikirim melalui Internet ke sistem *Big Data*. Pada sistem *Big Data* tersebut menggunakan *Hadoop framework* dan *Spark* untuk digunakan sebagai *machine learning* yang akan mengklasifikasikan parameter kualitas air dengan menggunakan metode SVM.

3.2 Metode Penelitian

Pada Metode penelitian menjelaskan garis besar langkah-langkah penelitian yang akan dilakukan. Pada penelitian ini menggunakan metode penelitian seperti yang digambarkan pada Gambar 3.2.

Gambar 3.2. Metodologi penelitian.

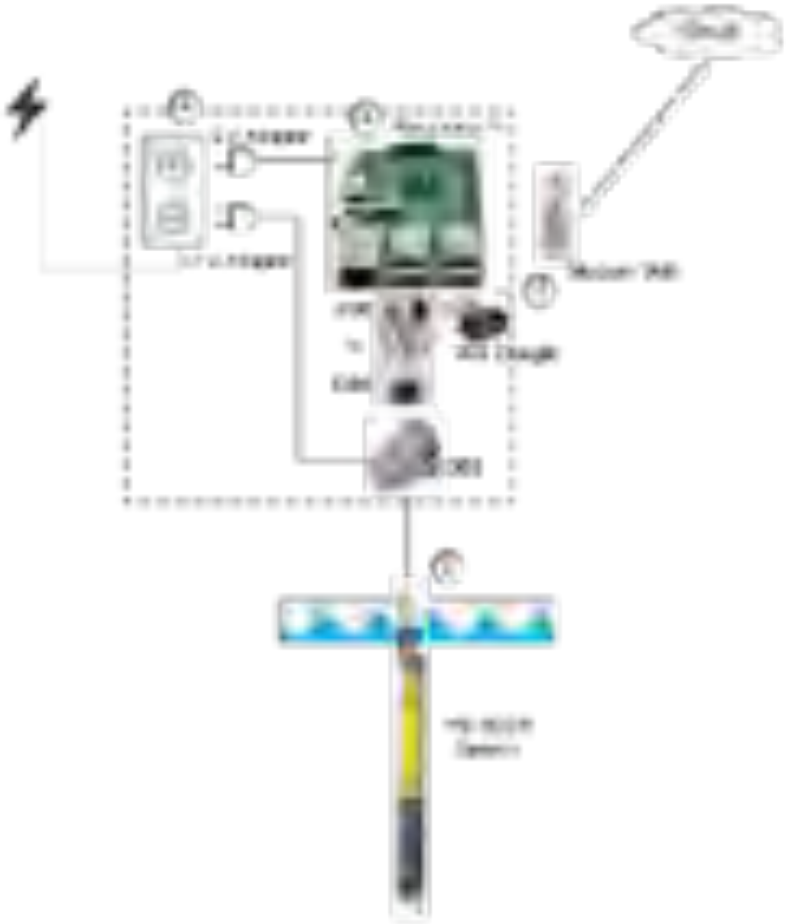
3.2.1 Studi Literatur, *Survey* dan Perijinan Lokasi

Pada tahap ini dilakukan survey di lokasi PDAM Karang Pilang dan pengurusan ijin untuk pemasangan perangkat sistem benam dan sensor kualitas air. Pada penelitian ini juga kami telah mendapatkan data dari PDAM yang berisi tentang laporan kondisi kualitas air baku pada lokasi Karang Pilang dan Ngangel.

3.2.2 Pengembangan perangkat sensor berbasis *Internet of Things* (Sensor Air, *Rapberry Pi*, dan 4G modem)

Untuk mendapatkan data kualitas air, pada penelitian ini telah dibuat perangkat Internet of Things yang terdiri dari rangkaian sensor yang menggunakan sensor kualitas air dan dihubungkan dengan sistem benam yang dilengkapi dengan perangkat lunak yang berfungsi sebagai unit pemrosesan data dari sensor dan mengirim data menuju ke suatu data center dengan menggunakan

koneksi 4G. Rangkaian sensor ini diletakkan di sungai Karang Pilang. Desain perangkat sensor IoT dapat dilihat pada Gambar 3.3.



Gambar 3.3. Desain perangkat sensor air berbasis *Internet of Things*.

Pada subbab berikut akan dijelaskan mengenai implementasi dari perangkat sensor IoT.

3.2.2.1 Pengembangan Sistem Benam beserta Perangkat Lunak

Sistem Benam *Raspberry Pi* 3 digunakan sebagai pusat pemrosesan data. Sistem benam ini menggunakan sistem operasi Raspbian, dimana sistem operasi ini berbasis Debian GNU/Linux yang sudah dimodifikasi untuk dapat berjalan pada sistem benam *Raspberry Pi*.

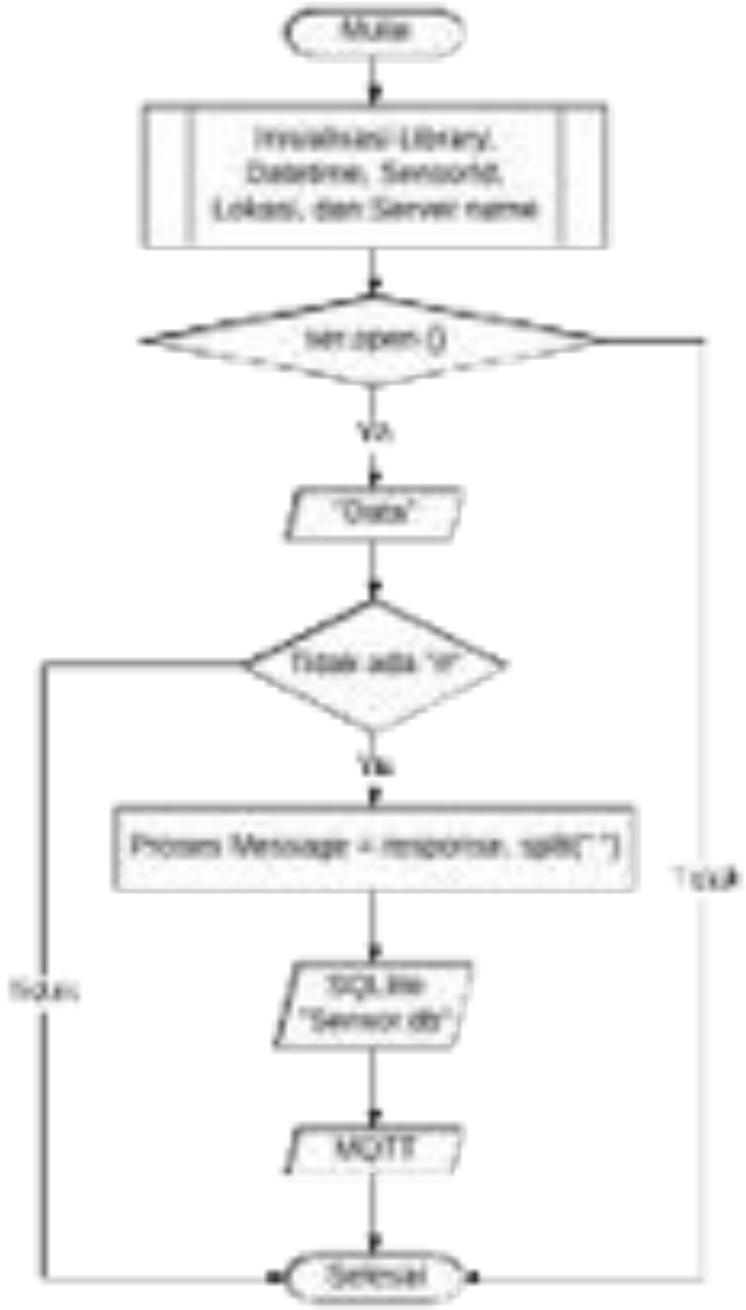
Perangkat lunak yang dibangun pada sistem benam ini dapat dilihat seperti pada *Flowchart* pada Gambar 3.4.

Aplikasi watermond atau water monitoring daemon ini berfungsi untuk mengambil data kualitas air dari sensor, kemudian diproses untuk supaya data dapat disimpan pada sistem database lokal yang bebasis *SQLite* dan mengirimkan data sensor ke data center dengan menggunakan protokol MQTT. Aplikasi ini berjalan pada proses background. Pada aplikasi ini menggunakan bahasa pemrograman Python, dimana dilengkapi dengan beberapa library pendukung antara lain: pySerial, *SQLite* dan MQTT.

Untuk membaca data dari sensor YSI 600R, aplikasi watermond perlu untuk mengirim karakter “data”. Kemudian sensor akan memberikan feedback berupa data-data dari sensor. Contoh kode sumber untuk mengambil data menggunakan pySerial dapat dilihat pada Kode Sumber 3.1.

```
import serial

ser = serial.Serial()

ser.port = /dev/ttyUSB0

ser.open()

while(len(response) <= 0)

      ser.write(“data\r”)

      response = ser.readline().strip(“\r\n”)
```

Kode Sumber 3.1. Kode sumber pembacaan data serial dari sensor.



Gambar 3.4. *Flowchart* aplikasi watermond: pengambilan, pemrosesan dan pengiriman data sensor.

Data yang sudah diambil dari sensor akan di simpan dalam bentuk SQL pada sistem database lokal dengan menggunakan *SQLite*. Dimana untuk menyimpan data tersebut menggunakan sumber kode pada Kode Sumber 3.2.

```
import sqlite3 as lite

con = lite.connect(sensordb)

with con:

      cur = con.cursor()

      cur.execute("INSERT INTO sensor_master (datetime, sensorid, lat, long, data1, data2, data3, data6, data8, data11, data13) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);", (datetimenya, sensorid, lat, long, temp, cond, tds, sal, dosat, do, ph))

      con.commit()
```

Kode Sumber 3.2. Menggunakan SQLite untuk menyimpan data sensor secara lokal.

Setelah data disimpan secara lokal, sistem benam mengirimkan data ke data center dengan menggunakan protokol MQTT. Dimana untuk pengiriman datanya dapat dilihat pada Kode Sumber 3.3.

```
import paho.mqtt.client as mqtt

response = "Data >> "+datetimenya+" id:"+sensorid+" loc:"+lat+","+long+" Temp(C):"+temp+" Conductivity(uS/cm):"+cond+" TDS(g/L):"+tds+" Salinity(ppt):"+sal+" DOsat(%):"+dosat+" DO(mg/L):"+do+" pH:"+ph+" >"

client = mqtt.Client()

client.connect(“202.182.58.3”,”1883”,60)

client.publish("watermon", response);

client.disconnect();
```

Kode Sumber 3.3. Pengiriman data menggunakan MQTT.

Pada sistem benam telah disetting supaya aplikasi watermond dapat berjalan terus pada proses background. Selain itu apabila terjadi kerusakan koneksi, aplikasi watermond dapat menghentikan pengiriman data dan hanya menyimpan secara lokal. Apabila koneksi dapat dilakukan secara otomatis aplikasi watermond dapat mengirimkan data kembali ke data center.

3.2.2.2 Perakitan perangkat sensor dengan sistem benam

Perangkat sensor YSI 600R telah dimodifikasi sehingga memiliki output konektor DB9. Dimana komunikasi yang digunakan adalah komunikasi data serial. Untuk dapat terhubung dengan sistem benam, sensor YSI 600R dihubungkan dengan kabel USB-to-serial.

Sensor YSI 600R bekerja dengan catu daya 12V DC. Dimana pada sistem ini menggunakan adaptor DC 12 V.

3.2.2.3 Menghubungkan perangkat komunikasi nirkabel

Untuk dapat mengirimkan data ke data center, sistem benam diberikan adapter Wireless via USB. Dimana komunikasi ini terhubung dengan Modem USB 4G melalui protokol IEEE802.11n. Sistem pengalamatan IP yang digunakan menggunakan DHCP yang dikontrol melalui modem.

Modem USB 4G bekerja sebagai router yang menghubungkan koneksi WAN dengan menggunakan LTE dengan jaringan LAN yang terhubung melalui Wireless LAN.

3.2.2.4 Catu daya

Perangkat sensor ini memerlukan catu daya, dimana catu daya yang digunakan adalah 5V DC 3A digunakan untuk memberikan daya pada sistem benam, 12V DC 1A dgunakan untuk memberikan daya pada sensor YSI 600R, dan 5V DC 1 A digunakan untuk memberikan daya pada modem USB 4G. Dimana sumber untuk perangkat ini diambil dari kediaman penduduk di sekitar lokasi sungai Karang Pilang.

3.2.2.5 Implementasi Sensor IoT

Dalam penelitian ini, sistem benam dirangkai dalam suatu *outdoor* box yang dapat tahan air seperti yang dapat dilihat pada Gambar 3.5 yang merupakan implementasi dari Sensor IoT yang dilengkapi sistem benam dan catu daya.

Gambar 3.5. Sensor IoT yang dilengkapi dengan sistem benam dan catu daya.

Sensor air perlu diletakkan di area sungai Kali Surabaya, untuk pengambilan nilai parameter air yang diperlukan pada proses penelitian, seperti yang dapat dilihat pada Gambar 3.6.

Gambar 3.6. Pemasangan sensor di lokasi.

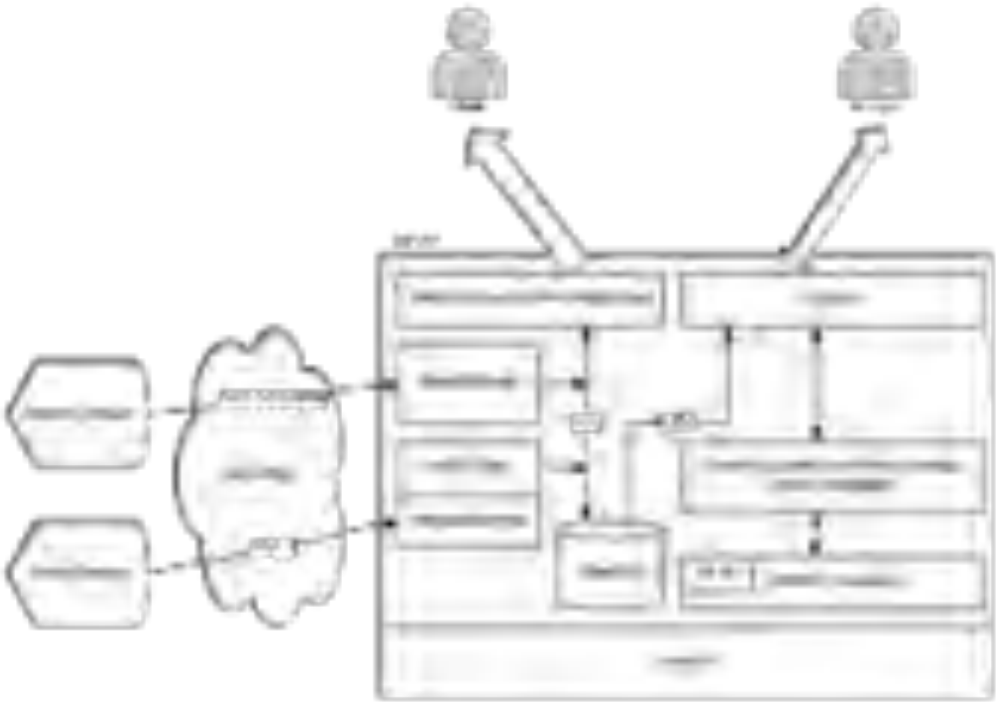
Posisi pengambilan data dan penempatan area untuk pemasangan sensor dapat dilihat pada Gambar 3.7.

Gambar 3.7. Peta lokasi sensor.

3.2.3 Pengembangan data center berbasis *Big Data*

Pada penelitian ini kami telah membangun IoT platform berbasis *Big Data*, dimana sistem ini telah mampu menerima data dari sensor yang telah dikirimkan menggunakan protokol MQTT, platform ini juga bisa mengambil data dari sensor pasif dengan menggunakan teknologi web scrapping, data yang telah dikirim ke data center dapat diolah dengan tool berbasis web interaktif dan dilakukan klasifikasi dengan machine learning.

Sistem IoT Platform berbasis *Big Data* yang telah kita bangun memiliki desain yang dapat dilihat pada Gambar 3.8.



Gambar 3.8. Arsitektur IoT Platform server berbasis *Big Data*.

Subbab berikut akan menjelaskan implementasi dari IoT Platform berbasis *Big Data*.

3.2.3.1 Installasi *Spark* dan DBMS di mesin Linux OS

Server IoT Platfrom yang digunakan adalah Debian GNU/Linux dengan kernel 4.4.21-1-pve. Dimana server yang digunakan terhubung dengan Internet menggunakan IP Publik. Sehingga sensor dapat mengirimkan data melalui Internet dengan mengakses IP dari server tersebut.

Sistem *Big Data* yang digunakan pada server ini menggunakan *Spark*-2.0.2-bin-haddop2.7. Dimana aplikasi *Spark* ini sudah dilengkapi dengan aplikasi Hadoop *Big Data* dengan versi 2.7. Untuk dapat mengakses sistem *Spark*, pada penelitian ini menggunakan bahasa pemrograman Python yaitu dengan perintah *pySpark*. *Spark* sudah mendukung beberapa library yang dapat digunakan untuk Machine Learning.

Untuk menyimpan data dengan ukuran yang cukup besar, pada penelitian ini digunakan DBMS server MariaDB. MariaDB adalah pengembangan sistem

DBMS dari MySQL yang digunakan untuk ukuran enterprise. Dimana MariaDB ini sudah mendukung high availabitiy, keamanan, interopability, dan performa dengan kemampuan untuk *Big Data*. Versi MariaDB yang digunakan adalah 10.1.19 dari distribusi Debian GNU/Linux.

3.2.3.2 Pembuatan aplikasi MQTT2DB untuk sensor aktif

Pada penelitian ini digunakan sensor kualitas air yang terhubung dengan sistem benam *Raspberry Pi*. Sistem sensor ini mampu mengirim data dari sensor dengan menggunakan protokol MQTT, sensor ini disebut juga dengan sensor aktif.

Sensor aktif bertindak sebagai MQTT publisher dan mengirim data dari sensor kualitas air, diterima oleh MQTT Broker - *mosquitto* server dengan topik “watermon”. Untuk dapat menyimpan data yang masuk ke MQTT Broker kedalam database, diperlukan penghubung. Pada penelitian ini telah dibuat aplikasi MQTT2DB berbasis pemrograman Python yang berfungsi untuk menyalurkan data dari sensor yang diterima di MQTT Broker kedalam DBMS.

Aplikasi MQTT2DB bertindak sebagai MQTT subcriber. Pada saat menerima message dari MQTT broker yang menerima data dari MQTT Publisher, MQTT2DB memilah pesan yang masuk dan melakukan SQL Insert kedalam DBMS. Kode sumber yang digunakan pada saat menerima pesan dan memasukkan kedalam DBMS dapat dilihat di Kode Sumber 3.4.

```
import paho.mqtt.clients as paho

import pymysql

def on_message(client, userdata, msg):

      data = filter(None,msg.payload.split(" "))

      cursor = db.cursor()

      sql="INSERT  INTO  `sensordb`  (`datetime`,  `sensorid`,
`lat`,  `long`,  `temp`,  `cond`,  `tds`,  `sal`,  `dosat`,  `do`,
`ph`) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s);"

      try:

                  cursor.execute(sql,     (datetime,   sensorid,
lat, long, temp, cond, tds, sal, dosat, do, ph))

                  db.commit()

      except:

                  db.rollback()

db
pymysql.connect(host="localhost",user="watermon",password="wate
rmon",db="watermon" )


client = paho.Client()

client.on_message = on_message

client.connect("localhost", 1883)

client.subscribe("watermon", qos=1)

client.loop_forever()

db.close()
```

Kode Sumber 3.4. MQTT2DB.

Aplikasi MQTT2DB berjalan pada proses background, dimana dengan bantuan cron pada Linux proses ini akan diperiksa apakah tetap berjalan atau tidak. Dan apabila aplikasi ini tidak berjalan akan dihidupkan kembali oleh shell script yang dijalankan melalui cron.

3.2.3.3 Pengembangan aplikasi WebScraping untuk sensor pasif

Pada penelitian ini menggunakan juga sensor pasif WTW IQ SensorNet 2020 XT, dimana sensor tidak dapat mengirimkan data sensor menuju ke data center. Sehingga data center perlu mendownload data yang berada pada sensor untuk dapat dimasukkan kedalam database di server data center. Sensor pasif pada penelitian ini memiliki luaran berbasis web yang berjalan di sensor. Data yang ditampilkan berformat HTML dan terstruktur dalam suatu tabel. Sehingga diperlukan suatu mekanisme yang dapat memilah data sensor dari tabel yang berbasis HTML dan menyimpan data yang diambil dalam database. Keluaran dari sensor dapat dilihat pada Gambar 3.9.

Gambar 3.9. Keluaran sensor dengan format HTML.

Mekanisme yang bisa dilakukan adalah dengan Web Scrapping, web scrapping adalah suatu mekanisme pendownloadan suatu web dan memilah data yang diinginkan dari struktur bahasa HTML. Untuk dapat melakukan web scrapping, pada penelitian ini menggunakan library BeautyfulSoap dengan pemrograman bahasa Python. Implementasi web scrapping yang telah dilakukan

antara lain: 1) terhubung dengan sensor, 2) Memilah informasi tanggal, 3) preprosessing pemilahan data pada tabel HTML, 4) pengolahan data dan memasukkan kedalam database.

Implementasi aplikasi web scrapping untuk mengambil data dari sensor menggunakan library urllib dapat dilihat pada Kode Sumber 3.5.

```
r = urllib.urlopen("http:// 192.168.7.80/").

read()

soup = BeautifulSoup(r, "html.parser")
```

Kode Sumber 3.5. Koneksi ke sensor dengan urllib.

Setelah mendownload konten HTML dari sensor dilakukan pemilahan konten tanggal. Implementasi kode sumber untuk memilah dapat dilihat di Kode Sumber 3.6.

```
for string in  soup.p.stripped_strings:

      date = (repr(string))[8:28]
```

Kode Sumber 3.6. Pemilahan konten tanggal.

Preprosessing untuk memilah data sensor pada tabel HTML dilakukan dengan implementasi kode sumber pada Kode Sumber 3.7. Langkah-langkah yang dilakukan antara lain: *BeautifulSoap* melakukan fungsi extract(), dilakukan pencarian unsur “table” dengan sintak soup.find(‘table’) dan mencari elemen “tr” dengan table.findAll(‘tr’). Langkah berikutnya adalah mengambil semua elemen pada setiap kolom dengan menjalankan sintak tr.findAll(‘td’).

```
soup.thead.extract()

table = soup.find('table')

rows = table.findAll('tr')

try:

for tr in rows:

        cols = tr.findAll('td')

        text_data = []

        for td in cols:

            text = ''.join(td)

            utftext = str(text.encode('utf-8'))

            text_data.append(utftext) # EDIT

        text = ','.join(text_data)

        print(date+" "+ text + '\n')     #print all data

        data = text.split(",",10)
```

Kode Sumber 3.7. Pemilahan data pada tabel HTML.

Elemen data telah didapatkan pada tahap preprosessing dan dilanjutkan dengan memasukkan kedalam database dengan kode sumber pada Kode Sumber 3.8. Elemen data dimasukkan kedalam database dengan query INSERT.

```
cur = con.cursor()

       query =  "INSERT INTO ngagel2(sid, state, type, serno,
name, mval, munit, mpara, sval, sunit, info) VALUES (%s, %s,
%s, %s, %s, %s, %s, %s, %s, %s, %s);"

        values = (data[0], data[1], data[2], data[3], data[4],
data[5], data[6], data[7], data[8], data[9], data[10])



        cur.execute(query, values)

        con.commit()
```

Kode Sumber 3.8. Penyimpanan kedalam database.

3.2.3.4 Installasi Python Library untuk Machine Learning

Untuk dapat menjalankan fungsi machine learning, pada penelitian ini menggunakan beberapa library daru pemrograman Python. Library yang digunakan antara lain: *pySpark*, jupyter, *Scikit-learn*, scipy, numpy, dan matplotlib.

3.2.3.5 Integrasi *Spark*-*Jupyter* Notebook dan pengembangan Web-UI

Implementasi user interface (UI) pada penelitian ini terdiri dari 2 bagian, yaitu implementasi *Jupyter-notebook* dan web-ui. *Jupyter-notebook* adalah user interface berbasis web interaktif yang digunakan untuk dapat menjalankan perintah machine learning dan memberikan hasil analisa terhadap suatu data. Web-UI adalah user interface berbasis web dinamis yang digunakan untuk menampilkan data dan grafik secara real-time.

Jupyter-notebook (http://www.jupyter.org) adalah aplikasi open source yang dapat digunakan secara interaktif untuk menganalisa data sains dan komputer dan mendukung 40 bahasa pemrograman. *Jupyter* ini dapat di integrasikan dengan *Spark*. Untuk menjalankan jupyter yang sudah diintegrasikan

dengan *Spark* dapat dipanggil dengan perintah "*pySpark*" seperti yang terlihat pada Gambar 3.10.

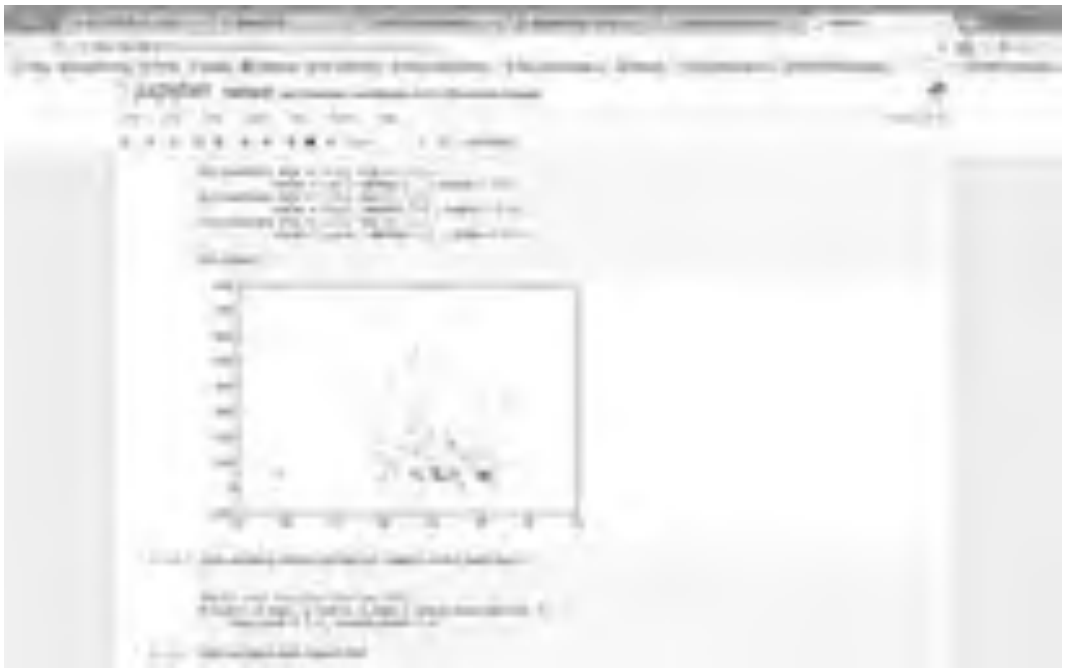
Gambar 3.10. Menjalankan pyspark

Jupyter ini diperuntukkan untuk analyst yang akan melakukan analisa terhadap data yang ada. Untuk melakukan integrasi *Spark* dan *Jupyter-notebook* dapat dilakukan dengan menambahkan syntax pada file .bash_profile. Isi syntax dapat dilihat pada Kode Sumber 3.9.

```
#setting path for spark

export SPARK_PATH='/Users/rizqi/spark-2.0.1-bin-hadoop2.7'

export PYSPARK_DRIVER_PYTHON="jupyter"

export PYSPARK_DRIVER_PYTHON_OPTS="notebook"

alias pyspark='$SPARK_PATH/bin/pyspark --master local[2]'
```

Kode Sumber 3.9. Integrasi Spark dan Jupyter Notebook.

Aplikasi *Jupyter-notebook* yang berjalan di server dapat dilihat seperti pada Gambar 3.11. *Jupyter* jalan di port 8888.



Gambar 3.11. *Jupyter* notebook pada server.

Web-UI yang telah diimplementasi dapat menampilkan grafik dari sensor dalam rentang waktu tertentu. Implementasi web-ui dapat dilihat pada Gambar 3.12. Web-UI berbasis PHP dan *HighChart javascript* yang digunakan untuk menampilkan grafik dari sensor secara live maupun dalam rentang waktu.

Gambar 3.12 Web-UI.

3.2.4 Pengumpulan dan Pengambilan data

Penelitian ini menggunakan beberapa data yang telah dikumpulkan dan diambil dari sensor. Pada subbab ini akan dijelaskan tentang data yang dipakai pada penelitian ini.

Data sensordb, data ini diperoleh dari sensor YSI 600R yang dipasang di perairan sungai di daerah Karang Pilang. Lokasi ini berdekatan dengan intake air PDAM Karang Pilang. Data ini diambil dengan tujuan untuk mendapatkan status kondisi air baku yang digunakan PDAM. Data diambil 24 jam terus-menerus dengan frekuensi pengambilan data 4-5 detik untuk sekali pembacaan data dari sensor, penyimpanan lokal di database yang terdapat di sistem benam, dan pengiriman menuju ke data center. Format data yang dikirimkan dapat dilihat pada Gambar 3.13. Format data yang dikirim berisikan informasi: tanggal dan waktu, id sensor, lokasi (*latitute, longitude*), *temperatur*, *conductivity*, *TDS*, *salinitas*, *DO saturasi*, *DO,* dan *pH*.

Gambar 3.13 Data sensordb yang diterima di data center

Data kedua yang digunakan pada penelitian ini adalah data dari sensor yang berada di reservoir PDAM untuk lokasi Karang Pilang dan Ngagel. Data ini diambil dari sensor WTW IQ SensorNet 2020 XT yang dipasang di PDAM. Sensor berjalan selama 24 jam, data diambil dengan teknik web scraping dengan waktu pengambilan 4-5 detik. Data dari Ngagel memiliki parameter antara lain: Turbidity untuk air baku, *Chlorine* untu air produksi, TSS, pH air baku, DO air baku, Turbidity air produksi. Data dari Karang Pilang memiliki parameter Turbidity air baku, Turbidity air produksi dan *Chlorine* air produksi. Data Karang Pilang selanjutnya disebut dengan datasenkp, dan data dari sensor yang berada di Ngagel disebut dengan datasenng.

Gambar 3.14. Data dari sensor PDAM yang dapat dilihat dari web browser.

Data ketiga yang diperoleh untuk penelitian ini adalah data laporan laboratorium dari PDAM untuk data dengan lokasi Karang Pilang dan Ngagel. Data ini berisikan laporan harian yang dikeluarkan oleh Laboratorium yang dimiliki oleh PDAM. Data berisikan tentang hasil analisa yang dilakukan oleh laboratorium dengan data sample yang diambil setiap harinya. Data ini memiliki paramter antara lain: suhu, kekeruhan, warna, ss, pH, Alkalinitas, CO2 Bebas, DO, Nitrit, Amonia, Tembaga, *Phospat*, Sulfida, Besi, Krom Heksavalen, Mangan, Seng, Timbal, COD, dan Detergen. Selanjutnya data laboratorium

dengan lokasi Karang Pilang disebut dengan datalabkp, dan data laboratorium dari lokasi Ngagel disebut dengan datalabng.

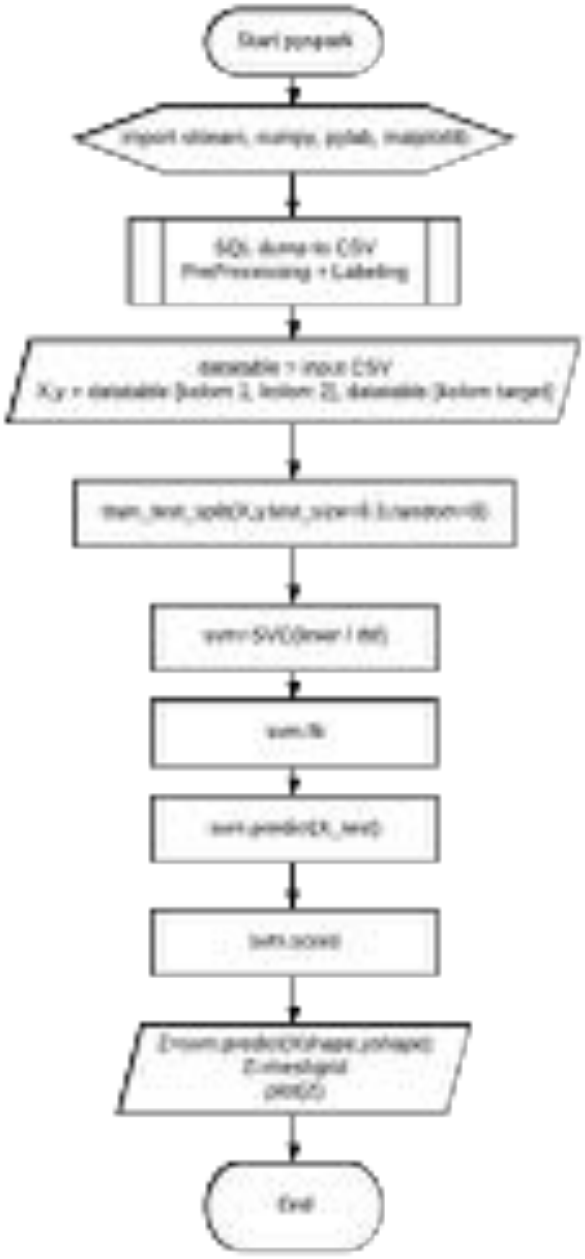
Gambar 3.15. Data dari laboratorium.

3.2.5 Pengembangan aplikasi klasifikasi berbasis *Spark* dan LibSVM

Pada tahap ini telah dikembangkan aplikasi klasifikasi kualitas air berbasis *Spark* dan LibSVM. Alur aplikasi dapat dilihat pada Gambar 3.16. Pada subbab ini akan dijelaskan tentang pengembangan aplikasi klasifikasi kualitas air.

3.2.5.1 Memulai *pySpark*

Untuk memulai pembuatan aplikasi kualitias air berbasis *Big Data*, diperlukan aplikasi *Spark*-Hadoop yang sudah diintegrasikan dengan *jupyter-notebook*. Untuk memulainya dipanggil dengan perintah pada CLI "*pySpark* –*Spark*://202.182.58.3:7077 –executor-memory=4G". Contoh pemanggilan *pySpark* dapat dilihat pada Gambar 3.10. Setelah *pySpark* berjalan akan keluar web interaktif *Jupyter*.



Gambar 3.16. *Flowchart* aplikasi klasifikasi kualitas air berbasis *Spark* dan LibSVM.

3.2.5.2 Import library pendukung machine learning

Untuk dapat melakukan klasifikasi pada data kualitas air, pada penelitian ini menggunakan library dari Python. Untuk dapat menggunakan library tersebut dapat dilakukan dengan perintah *import*. Contoh kode sumber untuk *import* dapat

dilihat pada Kode Sumber 3.10. Sintaks "*import*" library pendukung machine learning.

```
In [2]: %pylab inline
        from sklearn import datasets
        import matplotlib.pyplot as plt
        import numpy as np
        import pylab as pl

        sc

        Populating the interactive namespace from numpy and matplotlib
Out[2]: <pyspark.context.SparkContext at 0x10b528910>
```

Kode Sumber 3.10. Sintaks "*import*" library pendukung machine learning.

3.2.5.3 Persiapan data, preprosessing dan pemberian label

Pada penelitian ini semua data dari sensor maupun laporan disimpan di DBMS dengan format SQL. Data dari SQL tersebut dipindahkan ke format CSV untuk dapat dilakukan preprosessing dan pemberian label. Preprosessing dan pelabelan dilakukan untuk dapat diklasifikasikan dengan metode SVM.

Preprosessing masih dilakukan secara manual dengan bantuan perangkat lunak lain. Data dengan format CSV disusun berdasarkan urutan tanggal dengan pada kolom awal ditambahkan id untuk mempermudah perhitungan jumlah data. Data pada CSV diurutkan secara kolom dengan urutan mengikuti data sensor hasil pengukuran. Urutan data pada file CSV disusun sebagai berikut: id, tanggal, suhu, kekeruhan, TSS, pH, DO dan output. Urutan data pada file CSV dapat dilihat pada Gambar 3.17.

Pada kolom paling kanan ditambahkan kolom “output” yang digunakan sebagai pelabelan target untuk perhitungan metode SVM. Perhitungan kolom output menggunakan Indeks Pencemaran (IP) sesuai dengan formula pada subbab 2.2.3 Metode Indeks Pencemaran.

Dengan perhitungan Indeks Pencemaran tersebut didapatkan kategori untuk pencemaran air, dimana untuk kategori 0, adalah air jernih, kategori 1 adalah air tersemar ringan, kategori 2 adalah kondisi air tercemar sedang dan kategori 3 adalah kondisi air tercemar berat. Angka 0 s/d 3 digunakan sebagai pelabelan untuk digunakan pada metode SVM.

Gambar 3.17. Preprosessing data pada file CSV.

3.2.5.4 Input file data (CSV) dan pemisahan data set dan data class

Setelah persiapan data selesai, data pada file CSV perlu dimasukkan kedalam aplikasi klasifikasi kualitas air. Setelah memanggil file CSV, data dipisahkan menjadi data set dan data class. Data set terdiri dari 2 kolom data fitur (variabel X), sedangkan pada data class (variabel y) terdiri dari 1 kolom. Cara untuk memanggil file CSV dan memilah data set dan data class dapat dilakukan seperti pada Kode Sumber 3.11.

```
In [8]: DataTable = numpy.genfromtxt('stip-kp2016svm.csv',delimiter=',',dtype=None)[1:]
        X,y = (DataTable[:,[2,3] ]).astype(numpy.float), (DataTable[:,7]).astype(numpy.int)
```

Kode Sumber 3.11. Memanggil file CSV dan misahkan data set dan data class.

3.2.5.5 Memisahkan data untuk *training* dan data untuk *testing*

Dari data yang didapat perlu dipisahkan antara data untuk *training* dan data untuk *testing*. Sebelum dipisahkan data yang ada perlu diacak terlebih dahulu, kemudian dipisahkan. Untuk dapat melakukan itu dapat dilakukan dengan perintah "*train_test_split*". Penggunaan sintaks tersebut dapat dilihat di Kode Sumber 3.12. Keluaran dari fungsi *train_test_split* adalah X_train, X_test, y_train dan y_test.

```
In [12]: from sklearn.cross_validation import train_test_split

         #Split into training and test sets
         X_train, X_test, y_train, y_test = train_test_split(X, y,
              test_size = 0.3, random_state = 0)
```

Kode Sumber 3.12. Penggunaan "*train_test_split*".

3.2.5.6 Klasifikasi dengan *Support Vector Machine*

Dalam melakukan klasifikasi dengan menggunakan metode SVM pada *Big Data Framework*, alur yang digunakan dapat dilihat padaGambar 3.18.

Gambar 3.18. Alur klasifikasi SVM pada *Big Data Framework.*

Pada Gambar 3.18 menjelaskan alur klasifikasi menggunakan SVM pada *Big Data framework* menggunakan Input Data yang berupa nilai-nilai parameter air yang digunakan pada penelitian ini. Setelah data raw input yang didapatkan kemudian dilakukan proses preprosessing dimana pada proses ini dilakukan proses pemilihan data yang diperlukan dan membuang data yang tidak diperlukan semisal data yang berupa satuan dari nilai parameter air. Setelah itu dilakukan

proses labelling data dimana nilai-nilai parameter tersebut diberi label tersendiri. Penggabungan proses preprosessing dan labelling data inilah pada *Big Data* yang disebut dengan proses data cleansing. Kemudian dengan menggunakan dataframe, data tersebut diubah kedalam bentuk file CSV yang selanjutnya file CSV ini yang digunakan sebagai input data pada klasifikasi dengan melakukan *training* dan *testing*. Sebelum melakukan prediksi data yang sudah diambil dilakukan *training* terlebih dahulu. Untuk melakukan *training* dengan menggunakan SVM, perlu ditentukan terlebih dahulu jenis kernel SVM yang akan dipakai. Sehingga langkah-langkah untuk melakukan SVM antara lain: 1) menentukan jenis kernel yang akan digunakan, 2) melakukan fit *training*, 3) melakukan prediksi terhadap data test dan tambahan yang bisa dilakukan 4) menghitung *Score*, MSE, dan plot. Setelah itu, dilakukan analisa terhadap model klasifikasi yang digambarkan dalam bentuk kurva ROC.

Jenis kernel SVM yang bisa dipakai antara lain: 1) Linier, 2) RBF dan 3) *Polynomial*. Untuk menggunakan jenis kernel pada *Big Data* dapat dilakukan seperti pada Kode Sumber 3.13.

```
In [32]: from sklearn.svm import SVC

         #svm = SVC(kernel='linear')
         svm = SVC(kernel='rbf')
```

Kode Sumber 3.13. Pemilihan jenis kernel SVM.

Setelah ditentukan jenis kernel yang dipakai, dilakukan fit *training* dengan data X_train dan y_train dari hasil perintah "*train_test_split*". Fit *training* dapat dilakukan dengan kode sumber seperti pada Kode Sumber 3.14

```
svm.fit(X_train, y_train)
```

Kode Sumber 3.14. SVM fit *training*.

Setelah data dilakukan fit *training*, langkah selanjutnya adalah melakukan prediksi terhadap data test. Untuk melakukan prediksi terhadap data test dapat dilakukan dengan menggunakan kode sumber "svm.predict". Contoh prediksi dapat dilihat pada Kode Sumber 3.15

```
pred = svm.predict(X_test)
print("Number of mislabeled points out of a total %d points : %d"
      % (X_test.shape[0],(y_test != pred).sum()))
```

Kode Sumber 3.15 SVM prediksi.

Sebagai tambahan pada penelitian ini dihitung "*Score*" nilai akuasi dan "mse". Untuk melakukan perhitungan tesrebut dapat dilakukan seperti pada Kode Sumber 3.16.

```
score = svm.score(X_test, y_test)
print('Model Accuracy: %.3f' % (score))


mse = mean_squared_error(y_test, svm.predict(X_test))
print("MSE: %.4f" % mse)
```

Kode Sumber 3.16. Perhitungan *Score* dan mse.

Sebelum melakukan plot, perlu dicari terlebih dahulu range daerah data. Untuk mencari range daerah data dapat dilakukan dengan kode sumber seperti pada Kode Sumber 3.17

```
In [41]: x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
         y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
         xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.1),
                              np.arange(y_min, y_max, 0.1))

         Z = svm.predict(np.c_[xx.ravel(), yy.ravel()])
         Z = Z.reshape(xx.shape)
```

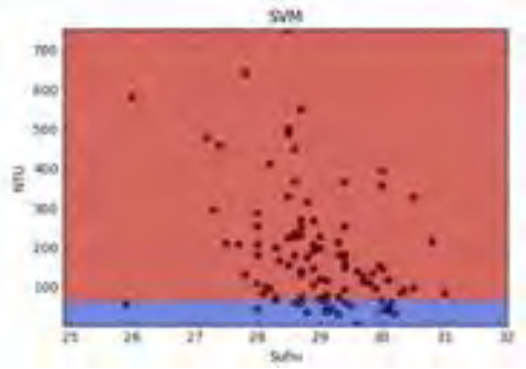
Kode Sumber 3.17. Mencari penyebaran data.

Setelah ditemukan daerah penyebaran data dilakukan plot untuk menghasilkan gambar klasifikasi. Untuk melakukan plot dapat dilakukan seperti pada Kode Sumber 3.18.

```
In [40]: plt.contourf(xx, yy, Z, cmap=plt.cm.coolwarm, alpha=0.8)

         plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.coolwarm)
         plt.xlabel('Suhu')
         plt.ylabel('NTU')
         plt.xlim(xx.min(), xx.max())
         plt.ylim(yy.min(), yy.max())
         #plt.xticks(())
         #plt.yticks(())
         plt.title("SVM")

         plt.show()
```

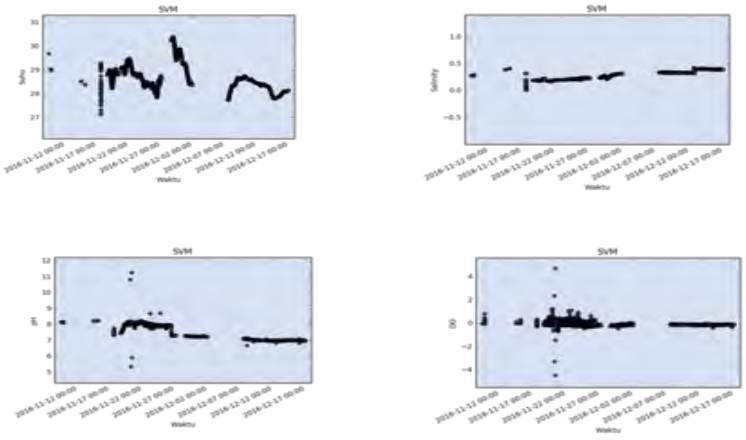
Kode Sumber 3.18. Plot hasil prediksi SVM.

Hasil plot untuk klasifikasi SVM dapat dilihat pada Gambar 3.19.



Gambar 3.19. Hasil plot klasifikasi.

Dari hasil klasifikasi yang dilakukan berikut merupakan contoh beberapa hasil plot yang diperoleh dari beberapa parameter air yang dapat dilihat pada Gambar 3.19.



Gambar 3.20 Hasil plot data dari beberapa parameter air

BAB 4
HASIL DAN PEMBAHASAN

Pada bab ini akan membahas mengenai penggunaan metode SVM (*Support Vector Machine*) terkait dengan proses klasifikasi air sungai dengan menggunakan framework *Big Data* sebagaimana pada penelitian tesis ini digunakan sebagai salah satu kebermanfaatan penggunaan *Big Data* dengan mengubah data yang bersifat pengukuran manual bisa di proses secara *online*.

Sistematika pembahasan pada penelitian ini secara garis besar terbagi menjadi 4 bagian, yaitu :

1. Ruang lingkup sistem, menjelaskan tentang perangkat yang digunakan pada ekperimen, versi perangkat lunak, dan penjelasan tentang data input yang digunakan.
2. Eksperimen klasifikasi air sungai berbasis SVM, dimana pada eksperimen ini menggunakan kernel Linear dan RBF serta pembahasan analisa dengan ROC.
3. Eksperimen performa SVM, didalam eksperimen ini akan dibandingkan prosentase dengan jumlah tahap pengujian (*training*) dan tahap pengujian (*testing*) yang berbeda.
4. Eksperimen penggunaan *Machine Learning* dengan MLlib-RDD dan Spark-*Scikit learn.*

4.1 Ruang Lingkup Sistem

4.1.1 Spesifikasi Sistem

Pada subbab ini dibahas performa klasifikasi SVM yang dilakukan pada data yang dimiliki. Metode SVM yang digunakan adalah Linear dan RBF. Dimana sistem yang digunakan untuk melakukan klasifikasi dapat dilihat pada Tabel 4.1.

Tabel 4.1. Spesifikasi Sistem.

Spesifikasi Sistem	
CPU	Intel Xeon E3-1220 v3 3.1Ghz
Core	4
RAM	16GB
HDD	1 TB
Perangkat Lunak	
OS	Debian GNU/Linux 8.6
Spark	*Spark*-2.0.2-bin-hadoop-2.7
Jupyter	4.2.0
Scikit-Learn	0.14.1
NumPy	1.8.2
Pandas	0.14.1
matplotlib	1.4.2

Koneksi pada *Big Data* server yang digunakan pada penelitian ini memiliki kecepatan data hingga 100Mbps untuk koneksi Internet dan IIX.

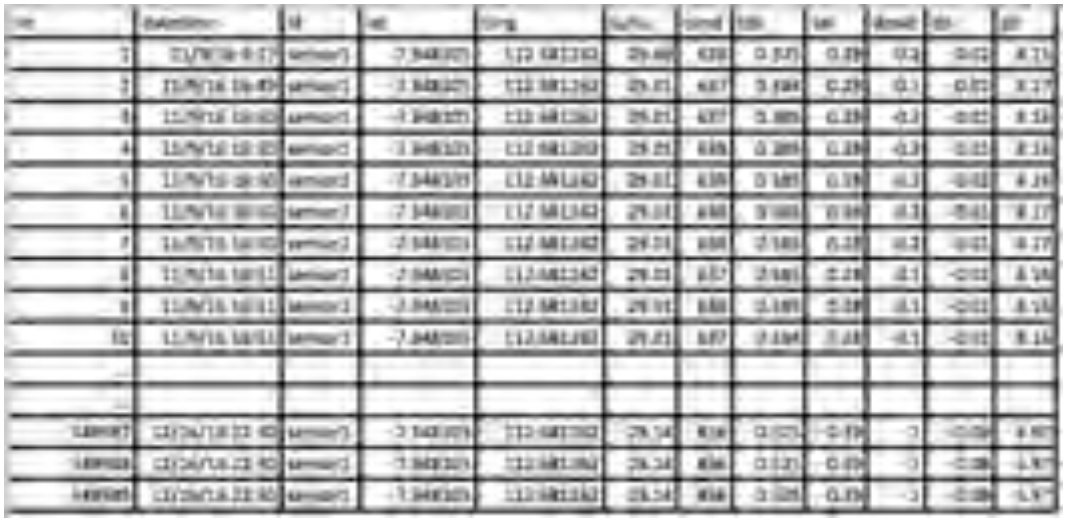
4.1.2 Input Data

Untuk data yang digunakan pada sistem ini terdapat beberapa jenis, yaitu: 1) Data dari sensor YSI 600R yang terpasang di daerah sungai Karang Pilang, data ini disebut dengan sensordb, 2) Data dari sensor reservoir air baku yang terdapat di PDAM dimana digunakan 2 lokasi yaitu Karang Pilang dan Ngagel, data ini disebut juga dengan datasenkp dan datasenng. 3) Data dari laporan tahunan laboratorium perusahaan PDAM tentang kondisi air baku, dimana data ini disebut dengan datalabkp dan datalabng.

4.1.2.1 Data sensordb

Data sensordb didapat dari sensor YSI 600R yang dipasang di daerah Karang Pilang dan terhubung dengan Internet menuju ke data center. Data yagn didapat dapat dilihat seperti pada Tabel 4.2. Data sensordb. Pada klasifikasi yang dilakukan pada penelitian ini parameter yang digunakan adalah pH, DO, dan salinity.

Tabel 4.2. Data sensordb



4.1.2.2 Data datasenkp

Data ini diambil dari sensor yang berada di perusahaan PDAM. Dimana sensor ini bersifat pasif, sehingga server yang perlu menarik data ke data center. Data datasenkp memiliki arti data dari sensor yang berada di PDAM Karang Pilang. Data dari sensor ini dapat dilihat pada Tabel 4.3. Data datasenkp.

Tabel 4.3. Data datasenkp.

1	1/10/16 12:23	S01	Measuring	MIQIC2 REC1	12511111	Air Pro KP1	0.93	ppm	Chlo	8.4	mA	0x0
2	1/10/16 12:23	S02	Measuring	VisoTurb700IQ	13021005	Air Pro KP1	0.81	NTU	Turb			0x0
3	1/10/16 12:23	S03	Measuring	VisoTurb700IQ	13021004	Air Baku KP	259	NTU	Turb			0x0
4	1/10/16 12:27	S01	Measuring	MIQIC2 REC1	12511111	Air Pro KP1	0.9	ppm	Chlo	8.31	mA	0x0
5	1/10/16 12:27	S02	Measuring	VisoTurb700IQ	13021005	Air Pro KP1	0.85	NTU	Turb			0x0
6	1/10/16 12:27	S03	Measuring	VisoTurb700IQ	13021004	Air Baku KP	266	NTU	Turb			0x0
..												
..												
501568	7/6/16 15:47	S01	Measuring	MIQIC2 REC1	12511111	Air Pro KP1	0.69	ppm	Chlo	5.75	mA	0x0
501569	7/6/16 15:47	S02	Measuring	VisoTurb700IQ	13021005	Air Pro KP1	1.46	NTU	Turb			0x0
501570	7/6/16 15:47	S03	Measuring	VisoTurb700IQ	13021004	Air Baku KP	134	NTU	Turb			0x0

Data ini masih bersifat *semi-unstructured*, dimana untuk 1 (satu) kali pengambilan data memiliki 3 baris data sensor. Contoh untuk data Tabel 4.3 urutan S01, S02, S03 kemudian kembali ke S01 dan seterusnya. Sebelum data dapat dipakai untuk klasifikasi, terlebih dahulu data dikumpulkan berdasarkan tanggal dan waktu yang sama, sehingga didapat data dengan pembacaan sensor pada 1 kolom yang sama. Paramter yang digunakan untuk klasifikasi dari data sensor datasenkp adalah data NTU dari Air Baku KP, sehingga diperlukan preproses untuk dapat digunakan pada klasifikasi SVM.

4.1.2.3 Data datasenng

Data ini diambil dari sensor yang berada di PDAM pada area Ngagel. Data ini bersifat pasif, artinya sensor tidak dapat mengirimkan ke data center sehingga perlu dilakukan mekanisme web scraping untuk dapat mengambil data sensor tersebut. Data datasenng dapat dilihat pada Tabel 4.4. Data datasenng.

Tabel 4.4. Data datasenng.

Data ini masih bersifat *semi-unstructured*, dimana untuk 1 (satu) kali pengambilan data memiliki 6 baris data sensor. Contoh untuk data Tabel 4.4 urutan S01, S03, S04, S05, S06, S07 kemudian kembali ke S01 dan seterusnya. Sebelum data dapat dipakai untuk klasifikasi, terlebih dahulu data dikumpulkan berdasarkan tanggal dan waktu yang sama, sehingga didapat data dengan pembacaan sensor pada 1 kolom yang sama. Data sensor yang diperlukan dari datasenkp adalah data NTU dari Air Baku NG, dan pH dari Air Baku NG, sehingga diperlukan preproses untuk dapat digunakan pada klasifikasi SVM.

4.1.2.4 Data datalabkp

Data datalabkp adalah data yang didapat dari laporan harian kondisi air baku PDAM untuk area Karang Pilang. Dimana data tersebut berupa file Excel yang berisi data laporan harian PDAM dengan range data dari tahun 2014 hingga september 2016. Data datalabkp dapat dilihat pada Gambar 4.1.

Pada datalabkp terdapat beberapa parameter kondisi air antara lain: suhu, kekeruhan, warna, ss, pH, Alkalinitas, CO2 Bebas, DO, Nitrit, Amonia, Tembaga,

Phospat, Sulfida, Besi, Krom Heksavalen, Mangan, Seng, Timbal, COD, dan Detergen.

Pengambilan data pada datalabkp ini dilakukan tiap hari, namun apabila bukan hari kerja tidak dilakukan pengambilan sampel data. Untuk 1 (satu) file excel terdiri dari 12 bulan laporan pengambilan sampel.

Untuk dapat diklasifikasikan format data tersebut terlebih dahulu dilakukan preproses, antara lain data berupa baris dirubah menjadi bentuk kolom, dsb.

Gambar 4.1. Data datalabkp.

4.1.2.5 Data datalabng

Data datalabng adalah data yang didapat dari laporan harian kondisi air baku PDAM untuk area Ngagel. Dimana data tersebut berupa file Excel yang berisi data laporan harian PDAM dengan range data dari tahun 2014 hingga september 2016. Data datalabng dapat dilihat pada Gambar 4.2.

Pada datalabng terdapat beberapa parameter kondisi air antara lain: suhu, kekeruhan, warna, ss, pH, Alkalinitas, CO2 Bebas, DO, Nitrit, Amonia, Tembaga, *Phospat*, Sulfida, Besi, Krom Heksavalen, Mangan, Seng, Timbal, COD, dan Detergen.

Pengambilan data pada datalabng ini dilakukan tiap hari, namun apabila bukan hari kerja tidak dilakukan pengambilan sampel data. Untuk 1 (satu) file excel terdiri dari 12 bulan laporan pengambilan sampel.

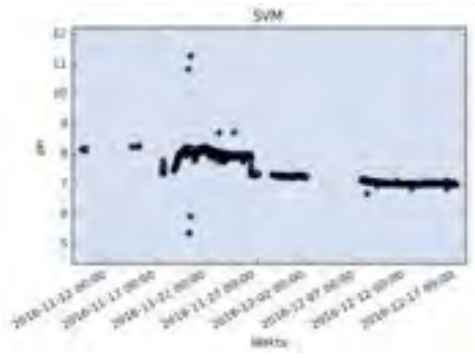
Gambar 4.2. Data datalabng.

Untuk dapat diklasifikasikan format data tersebut terlebih dahulu dilakukan preproses, antara lain data berupa baris dirubah menjadi bentuk kolom, dsb.

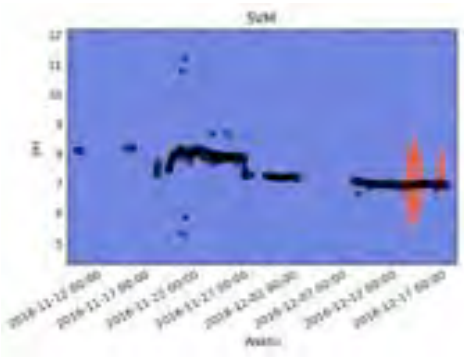
4.2 Eksperimen klasifikasi air sungai berbasis SVM

4.2.1 Perbandingan Kernel SVM Linear dengan RBF

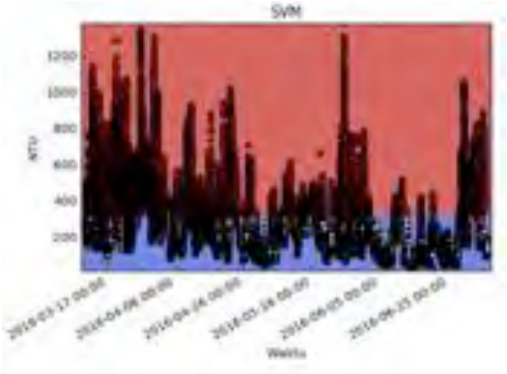
Pada ekperimen ini, kami melakukan klasifikasi dari ke-5 input data. Kami melakukan klasifikasi dengan menggunakan kernel SVC Linear dan juga RBF. Hasil dari perbandingan klasifikasi dapat dilihat pada Gambar 4.3.



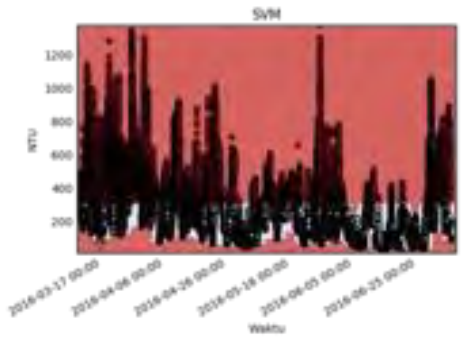
a) SVM Linear dengan data sensordb



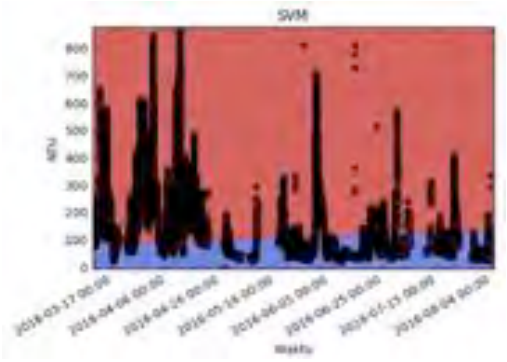
b) SVM RBF dengan data sensordb



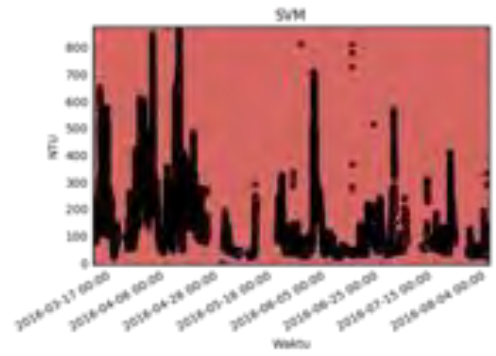
c) SVM Linear dengan data datasenkp



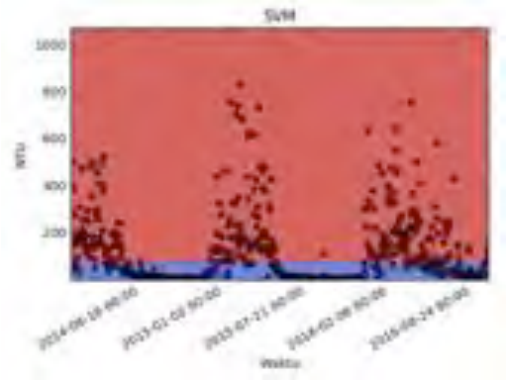
d) SVM RBF dengan data datasenkp



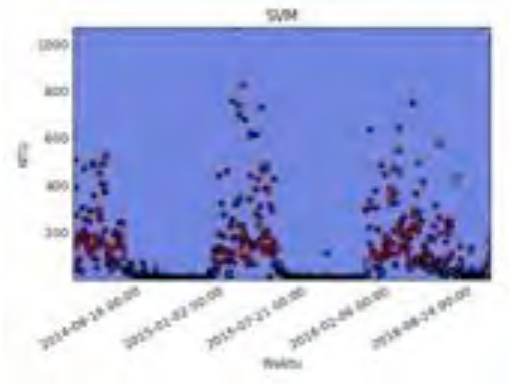
e) SVM Linear dengan data datasenng



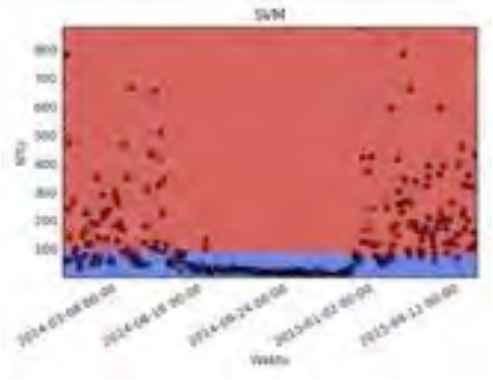
f) SVM RBF dengan data datasenng



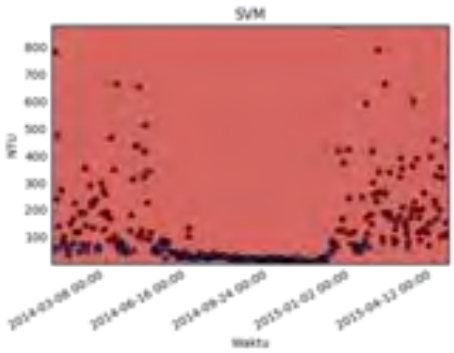
g) SVM Linear dengan datalabkp



h) SVM RBF dengan datalabkp



i) SVM Lienar dengan datalabng



j) SVM RBF dengan datalabng

Gambar 4.3. Perbandingan SVM Lienar dengan SVM RBF dengan input data.

4.2.2 *Mislabel, Score,* dan MSE

Dalam melakukan klasifikasi, sebuah sistem diharapkan dapat melakukan klasifikasi semua set data dengan benar. Sebenarnya, kinerja suatu sistem klasifikasi tidak bisa bekerja secara 100% benar, namun kinerja sistem klasifikasi dapat diukur. Dalam mengukur kinerja sistem klasifikasi dapat menggunakan *confussion matrix*. Berikut tabel matriks confusion yang digunakan pada penelitian ini, dapat dilihat pada Tabel 4.5.

Tabel 4.5. Tabel Confussion Matrix

f_{ij}		*Predicted Class*			
		0	**1**	**2**	**3**
Actual Class	**0**	f_{00}	f_{01}	f_{02}	f_{03}
	1	f_{10}	f_{11}	f_{12}	f_{13}
	2	f_{20}	f_{21}	f_{22}	f_{23}
	3	f_{30}	f_{31}	f_{32}	f_{33}

Pada setiap sel f_{ij} dalam matriks menyatakan jumlah data dari kelas i yang hasil prediksi masuk ke dalam kelas j. Misalkan f_{00} adalah jumlah data kelas 0 yang secara benar dipetakan ke dalam kelas 0. Sedangkan pada f_{00}, f_{01}, f_{02} adalah jumlah data pada kelas 0 yang dipetakan secara salah ke kelas selain 0. Berdasarkan confussion matrix diatas, maka data yang diklasifikasi dengan benar adalah $f_{00}, f_{11}, f_{22}, f_{33}$. Sedangkan data yang diklasifikasi dengan salah adalah $f_{01}, f_{02}, f_{03}, f_{10}, f_{12}, f_{13}, f_{20}, f_{21}, f_{23}, f_{30}, f_{31}, f_{32}$ sehingga dapat dihitung nilai *Mislabel*, *Score* dan MSE.

Mislabel adalah hasil jumlah data yang diprediksi secara salah terhadap jumlah data test yang akan diprediksi setelah dilakukan *training* data. Dimana pada eksperimen ini menggunakan perbandingan data *training* dan data test sebesar 70%:30%. *Score* adalah nilai akurasi hasil prediksi untuk mengetahui jumlah data yang diklasifikasikan secara benar. Untuk menghitung akurasi digunakan persamaan (4.1).

$$Score = \frac{Jumlah\ data\ yang\ diprediksi\ benar}{Jumlah\ data\ prediksi\ total} \qquad (4.1)$$

MSE adalah nilai *error rate* yang digunakan untuk mengetahui jumlah data yang diklasifikasikan secara salah sehinggga mengetahui laju error pada prediksi yang dilakukan. Untuk menghitung nilai *error rate* digunakan persamaan (4.2).

$$MSE = \frac{Jumlah\ data\ yang\ diprediksi\ salah}{Jumlah\ data\ prediksi\ total} \quad (4.2)$$

Berdasarkan hasil eksperimen yang dilakukan terhadap semua Input Data dengan perbandingan data *training* dan data test sebesar 70%:30% , diperoleh tabel Nilai *Mislabel, Score*, dan MSE yang dapat dilihat pada Tabel 4.6.

Tabel 4.6. Nilai *Mislabel, Score* dan MSE.

Data	SVM	*Mislabel*	*Score*	MSE
sensordb	Linear	11106/104997	0.894	0.1058
sensordb	RBF	552/104997	0.995	0.0053
datasenkp	Linear	1259/49785	0.975	0.0253
datasenkp	RBF	13/49785	1	0.0004
datasenng	Linear	7765/45841	0.831	0.1694
datasenng	RBF	44/45841	0.999	0.001
datalabkp	Linear	10/200	0.95	0.05
datalabkp	RBF	95/200	0.525	0.475
datalabng	Linear	8/99	0.919	0.0808
datalabng	RBF	33/99	0.667	0.3333

Percobaan berikut adalah menghitung *Mislabel*, *Score* dan MSE. *Mislabel* adalah jumlah pemberian label kepada data test setelah dilakukan *training* dengan perbandingan data *training* dan test sebesar 70%:30%. *Score* adalah nilai akurasi dalam menentukan prediksi. Mean Square Error (MSE) adalah rata-rata kesalahan.

4.2.3 ROC

Kurva ROC merupakan salah satu cara melakukan analisa terhadap model klasifikasi yang telah dibuat dalam menentukan parameter model yang diinginkan sesuai dengan karakteristik dari model klasifikasi yang diinginkan. Penggunaan kurva ROC seringkali digunakan dalam mengevaluasi proses klasifikasi, dikarenakan mempunyai kemampuan evaluasi secara menyeluruh dan cukup baik[44]. Misalkan pada table yang terdiri dari dua buah kelas data yaitu data kelas yang dihasilkan dari classifier (*Predicted Class*) dan data kelas asli yang telah diketahui (*Actual Class*) dimana data *Predicted Class* yang sama dengan *Actual Class* maka termasuk Tssitive (TP), sedangkan data *Predicted Class* yang

tidak sama dengan *Actual Class* tapi termasuk dari data hasil klasifikasi maka termasuk False Positive(FP).

Kurva ROC digunakan sebagai grafik perbandingan antara *True Positive Rate* (TPR) pada sumbu vertikal dengan *False Positive Rate* (FPR) pada sumbu horizontal. TPR merupakan proporsi data yang digunakan positif yang teridentifikasi dengan benar antara data *Predicted Class* dengan *Actual Class* sedangkan FPR merupakan proporsi data negative yang teridentifikasi salah sebagai positif pada suatu model klasifikasi[45]. *True Positive Rate* dan *False Positive Rate*, dapat dihitung menggunakan persamaan :

$$TPR = \frac{TP}{TP + FP}$$

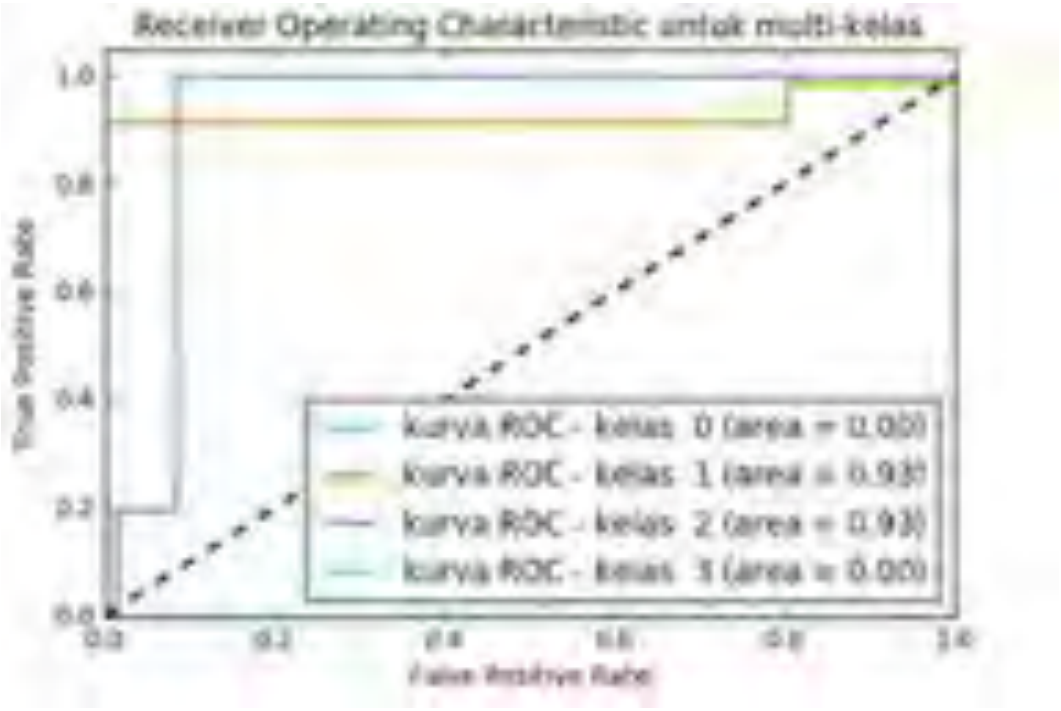
$$FPR = \frac{FP}{TP + FP} \qquad (4.3)$$

Pada kurva ROC, luas area dibawah kurva dikenal dengan AUC (Area Under the ROC *Curve*), dimana nilai AUC berkisaran antara 0 sampai dengan 1, semakin mendekati 1 maka semakin baik nilai uji pada karakteristik klasifikasi tersebut. Nilai kategori AUC[46], sebagai berikut :

Tabel 4.7. Tabel nilai AUC

Range Nilai AUC	Keterangan
0.5 - 0.6	Fail
0.6 - 0.7	Poor
0.7 - 0.8	Fair
0.8 – 0.9	Good
0.9 – 1	Excellent

Hasil eksperimen pencarian nilai ROC untuk klasifikasi SVM dapat dilihat pada Gambar 4.4.



Gambar 4.4. ROC pada datasensordb.

Dengan hasil eksperimen yang dilakukan pada datasensordb, dapat diperoleh nilai area dibawah kurva ROC dengan nilai untuk kelas 1 sebesar 0.93 dan kelas 2 sebesar 0.93, maka nilai hasil kinerja sistem klasifikasi menunjukkan "Excellent".

4.3 Eksperimen performa SVM

Pada eksperimen ini dilakukan dua macam percobaan untuk mengukur performa SVM. Percobaan pertama mengukur waktu yang dibutuhkan untuk melakukan *Training* dan Prediksi, dan yang kedua dengan mengukur akurasi dengan variasi learning rate.

4.3.1 Perbandingan waktu proses

Pada eksperimen ini dilakukan percobaan dengan mengukur perbandingan waktu yang dibutuhkan untuk melakukan *training* dan prediksi. Eksperimen ini dilakukan dengan menggunakan klasifikasi SVM dengan kernel linear dan RBF. Diujicobakan pada 5 input data dengan perbandingan data *training* dan test sebesar 70%:30%. Hasil dari ekperimen dapat dilihat pada Tabel 4.8.

Tabel 4.8. Perbandingan waktu proses.

Data	SVM Kernel	Jml Data	Waktu	
			Training	Prediksi
sensordb	Linear	349989	0:04:59	0:00:08
sensordb	RBF	349989	0:02:04	0:00:05
datasenkp	Linear	165949	0:00:40	0:29:31
datasenkp	RBF	165949	0:10:39	1:22:56
datasenng	Linear	152802	0:01:02	0:46:33
datasenng	RBF	152802	0:03:42	0:26:07
datalabkp	Linear	666	0:00:00	0:01:48
datalabkp	RBF	666	0:00:00	0:10:31
datalabng	Linear	330	0:00:00	0:00:38
datalabng	RBF	330	0:00:00	0:02:06

4.3.2 Eksperimen Akurasi dengan variasi nilai Learning Rate

Pada eksperimen ini kami memantau performa klasifikasi SVM dengan mengganti nilai perbandingan antara data *training* dan data test. Dimana nilai perbandingannya dimulai dari 10% hingga 90%. Eksperimen ini menghitung akurasi yang didapat dari nilai SVM.*Score*, MSE dan waktu yang dibutuhkan untuk melakukan *training* dan prediksi. Eksperimen ini menggunakan data sensordb sebesar 31MB dengan jumlah index sebanyak 349000. Hasil eksperimen dapat dilihat pada Tabel 4.9.

Tabel 4.9. Tabel performa klasifikasi SVM dengan variasi nilai learning rate.

Training/Test									
SVM	10:90	20:80	30:70	40:60	50:50	60:40	70:30	80:20	90:10
Akurasi									
Linear	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.895
RBF	0.979	0.979	0.997	0.996	0.994	0.994	0.995	0.995	0.996
MSE									
Linear	0.1065	0.1068	0.1062	0.1061	0.106	0.1058	0.1058	0.1056	0.1053
RBF	0.0215	0.0214	0.003	0.0041	0.0063	0.0055	0.0053	0.0048	0.0043
Waktu									
Linear	0:00:16	0:00:37	0:01:06	0:01:40	0:02:30	0:03:52	0:04:59	0:06:04	0:08:43
RBF	0:00:14	0:00:27	0:00:43	0:00:56	0:01:55	0:01:42	0:02:04	0:03:43	0:03:41
Mislabel									
Linear	33532/314991	19771/279992	26014/244993	22277/209994	18551/174995	14816/139996	11106/104997	7392/69998	3686/34999
RBF	6766/314991	6881/279992	725/244993	861/209994	1058/174995	826/139996	552/104997	338/69998	149/34999

Dengan perbandingan nilai *training* dan test yang berbeda didapatkan bahwa semakin tinggi nilai prosentasi *training* akan didapatkan akurasi yang lebih baik dan nilai kesalahan MSE yang semakin kecil. Namun proses *training* yang semakin tinggi juga akan memakan waktu yang semakin lama.

4.4 Eksperimen penggunaan *Machine Learning* dengan MLlib-RDD dan *Spark-Scikit Learn*

Pada eksperimen ini telah dilakukan percobaan klasifikasi SVM dengan dua library yang berbeda. Dari ekperimen tersebut dapat dibuat kesimpulan yang dapat dilihat pada

	Spark Mllib	*Spark Scikit-Learn*
Bahasa pemrograman	*Python*	*Python*
Tipe data	RDD	*Dataframe*
SVM	*Linear*	*Linear* dan *Non-Linear* (RBF)
Multiclass	-	Mendukung
Parallel	Mendukung	-
Visualisasi	-	Matplot

BAB 5
KESIMPULAN

5.1 Kesimpulan

Pada penelitian ini bertujuan untuk klasifikasi air sungai dengan menggunakan dua buah teknologi IoT dan *Big Data*, dimana IoT menggunakan beberapa perangkat IoT yang berbeda yang terdiri dari sensor *monitoring* kondisi air, sistem benam yang mampu mengolah data sensor dan mengirimkan ke data center yang berbasis *Big Data* serta untuk pengiriman data dengan protokol MQTT, dimana perangkat IoT ini bertujuan untuk mengakuisisi data sedangkan *Big Data* menggunakan data sensor sebagai data *cleansing* dan menggunakan SVM sebagai analisa klasifikasinya.

Hasil klasifikasi dengan menggunakan metode SVM menggunakan kernel Linear lebih tinggi akurasinya dibandingkan kernel RBF untuk datalabkp dan datalabng. Sedangkan pada data sensordb, datasenkp dan datasenng adalah sebaliknya. Nilai akurasi total (*Score*) untuk SVM dengan kernel Linear adalah 0.9138 dan SVM dengan kernel RBF adalah 0.8372.

Rata-rata MSE untuk SVM dengan kernel Linear memiliki nilai 0.08626 dimana SVM dengan kernel RBF memiliki nilai rata-rata MSE lebih tinggi yaitu 0.163. Dimana artinya adalah klasifikasi air sungai lebih bagus hasilnya dengan menggunakan metode SVM dengan kernel Linear.

Pengujian hasil validasi yang telah dilakukan pada datasensordb berdasarkan grafik ROC dengan nilai *Area Under ROC* menunjukkan 0.93. Dengan begitu dapat dikatakan bahwa unjuk kerja nilai *Area Under ROC* menunjukkan "*Excellent*".

5.2 Penelitian Lanjutan

Saran-saran yang diperlukan untuk pengembangan penelitian Tesis ini di masa depan :

1. Desain alat IoT yang digunakan pada penelitian ini hanya diperuntukkan untuk 1 buah sensor. Oleh karena itu, diharapkan kedepannya lebih

banyak penggunaan sensor dengan data yang lebih banyak dan akurat serta tersebar di berbagai lokasi agar implementasi sistem ini dapat lebih terlihat kebermanfaatannya karena penggunaan proses *training* secara *online* dan dalam kondisi *real*.

2. Perlu dikembangkan sebuah metode yang efisien untuk mengetahui distribusi data dengan metode yang lebih baik dan efisien lagi khususnya untuk diimplementasikan di masa mendatang.
3. Diperlukan algoritma yang sangat cepat dalam menentukan Indeks Pencemaran jika jumlah data yang digunakan sebagai pelatihan dan pengujian lebih kompleks serta pemodelan yang lebih efisien untuk menentukan prediksi kedepannya.

.

DAFTAR PUSTAKA

[1] Pemerintah Kota Surabaya, “Pengelolaan Kualitas Air dan Pengendalian Pencemaran Air”, Peraturan Daerah Kota Surabaya Nomor 2 Tahun 2004.

[2] C. Veesommai and Y.Kiyoki. ”Critical Contaminate Detection, Classification of Multiple-water-quality-parameters Values and Real-time Notification by rSPA Processes”. Surabaya ; IEEE International Electronics Sysposium, 2015.

[3] A. Sarkar dan P.Pandey, ”River Water Quality Modelling Using Artificial Neural Network Technique”. Aquatic Procedia, Vol.4, 2015, Pages 1070-1077.

[4] Y.R.Ding, Y.J.Cai, P.D.Sun and B.Chen. ”The Use of Combined Neural Networks and Genetic Algorithms for Prediction of River Water Quality” Journal of Applied Research and Technology. Vol.12, Issue 3, June 2014, Pages 493-499.

[5] Noori, Roohollah, Zhiqiang Deng, Amin Kiaghadi, and Fatemeh Torabi Kachoosangi. "How Reliable Are ANN, ANFIS, and SVM Techniques for Predicting Longitudinal Dispersion Coefficient in Natural Rivers?." Journal of Hydraulic Engineering , 2015: 04015039.

[6] Shuxiu Liang, Songlin Han, Zhaochen Sun, “Parameter optimization method for the water quality dinamic model based on data-driven theory”. China, 2011.

[7] Yue Liao, Jianyu Xu, Wenjing Wang , “A method of Water Quality Assessment Based on Biomonitoring and Multiclass *Support Vector Machine*”. International Conferences on ESIAT, 2011.

[8] Chen, Min, Shiwen Mao, and Yunhao Liu. "*Big Data*: a survey." *Mobile Networks and Applications* 19, no. 2 , 2014: 171-209.

[9] Rao, Aravinda S., Stephen Marshall, Jayavardhana Gubbi, Marimuthu Palaniswami, Richard Sinnott, and Vincent Pettigrovet. "Design of low-cost autonomous water quality monitoring system." In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, pp. 14-19. IEEE, 2013.

[10] M. Herwindra Berlian, T. E. Rindang Sahputra, B. Ardi, L. Wahya Dzatmika, A. R. Anom Besari, R. Sudibyo, S. Sukaridhoto, "Design and Implementation of Smart Environment Monitoring and Analytics in Real-Time System Framework Based on Internet of Underwater Things and *Big Data*", in *2016 Int. Electron. Symp.* IES, 2016.

[11] Fang, Shifeng, Li Da Xu, Yunqiang Zhu, Jiaerheng Ahati, Huan Pei, Jianwu Yan, and Zhihui Liu. "An integrated system for regional environmental monitoring and management based on internet of things." *IEEE Transactions on Industrial Informatics* 10, no. 2, 2014: 1596-1605.

[12] Singh, Kunwar P., Nikita Basant, and Shikha Gupta. "*Support Vector Machine*s in water quality management." *Analytica chimica acta* 703, no. 2, 2011: 152-162.

[13] Koliopoulos, Aris-Kyriakos, Paraskevas Yiapanis, Firat Tekiner, Goran Nenadic, and John Keane. "A Parallel Distributed Weka Framework for *Big Data* Mining using *Spark*." In *2015 IEEE International Congress on Big Data*, pp. 9-16. IEEE, 2015.

[14] Undang-Undang Republik Indonesia Nomer 7 , “Sumber Daya Air”, Tahun 2004.

[15] Nemerow, Nelson Leonard. *Scientific stream pollution analysis*. McGraw-Hill, 1974.

[16] Presiden, R. I. "Peraturan Pemerintah Nomor 82 Tahun 2001, Tentang Pengelolaan Kualitas Air dan Pengendalian Pencemaran Air.” Lembaran Negara Republik Indonesia Nomor 4161, 2001.

[17] Indonesia, Menteri Kesehatan Republik. “Peraturan Menteri Kesehatan Republik Indonesia Nomor 492”. MENKES/PER/IV/2010 Tentang Persyaratan Kualitas Air Minum. Menteri Kesehatan Republik Indonesia. Jakarta. 16 h, 2010.

[18] Keputusan Menteri Negara Lingkungan Hidup Nomer 115, “Pedoman Penentuan Status Mutu Air”, Tahun 2003.

[19] Recommendation ITU-T Y.2060 “Overview of Internet of Thing” (06/2012)

[20] “600R Multi-Parameter Water Quality Sonde”, https://www.ysi.com/600R, diakses 6 Desember 2015.

[21] “WTW IQ SensorNet TC 2020 XT”, http://www.wtw.com/en/products/product-categories/stationary-meters/iq-sensor-net/miqtc-2020-xt.html, diakses 20 Desember 2016.

[22] “Raspberry – Teach Learn and Make with Rapberry Pi”, http://www.raspberrypi.org, diakses 6 Desember 2015.

[23] “Welcome to Python Programming”, http://www.python.org, diakses 6 Desember 2015.

[24] *SQLite*, https://*SQLite*.org/, diakses 20 Desember 2016.

[25] Elmasri, Ramez. *Fundamentals of database systems*. Pearson Education India, 2008.

[26] USB Modem 4G DT-100, https://alnect.net/product/7633/Page-Modem-GSM-4GLTE-Advance-Jetz-DT100-Plus-Soft-AP, diakses 1 Januari 2016.

[27] MQTT, "Message Queue Telemetry Transport," [*Online*]. Available: http://mqtt.org, diakses 30 November 2016.

[28] MQTT, "The MQTT Protocol Concept," [*Online*]. Available: http://mosquitto.org/man/mqtt-7.html, diakses 30 Nopember 2016.

[29] "Welcome to Apache Hadoop", http://hadoop.apache.org, diakses 6 Desember 2015.

[30] Hadoop Ecosystem Machine Learning, https://biguru.files.wordpress.com/2014/04/cdh.png, diakses 6 Desember 2015.

[31] "Apache *Spark* – Lighting Fast Cluster Computing", http://*Spark*.apache.org, diakses 6 Desember 2015.

[32] Overview of *Jupyter*, http://jupyter.readthedocs.io/en/latest/, diakses 6 Desember 2015.

[33] Scikit-Learn – machine learning in python, http://scikit-learn.org/stable/, diakses 1 Oktober 2016.

[34] Bowles, Michael. Machine learning in Python: essential techniques for predictive analysis. John Wiley & Sons, 2015.

[35] Scipy.org, http://scipy.org/, diakses 1 Oktober 2016.

[36] Numpy, http://numpy.org, diakses 1 Oktober 2016.

[37] Matplotlib - python ploting, http://matplotlib.org/, diakses 1 Oktober 2016.

[38] Bishop, C. M. "Bishop Pattern Recognition and Machine Learning" , 2001.

[39] I. Steinwart and A. Christmann, "*Support Vector Machine*", Information Science and Statistics, Springer, 2008.

[40] Prasetyo, Eko. "Data mining mengolah data menjadi informasi menggunakan matlab." *Yogyakarta: Andi*, *September* (2014).

[41] "Cloudera", http://www.cloudera.com, diakses 6 Desember 2015.

[42] "Basic Of *Big Data*", https://biguru.wordpress.com/tag/cloudera/, diakses 6 Desember 2015.

[43] Jeffrey Shafer, Scott Rixner, and Alan L.Cox. "The Hadoop Distributed Filesystem: Balancing Portability and Performance", IEEE International Symposium,2010.

[44] Cheng, H. D., Juan Shan, Wen Ju, Yanhui Guo, and Ling Zhang. "Automated breast cancer detection and classification using ultrasound images: A survey." *Pattern Recognition* 43, no. 1, 2010: 299-317.

[45] Fawcett, Tom. "An introduction to ROC analysis." *Pattern recognition letters* 27, no. 8, 2006: 861-874.

[46] Mohanty, Aswini Kumar, Swapnasikta Beberta, and Saroj Kumar Lenka. "Classifying benign and malignant mass using GLCM and GLRLM based texture features from mammogram." *International Journal of Engineering Research and Applications* 1, no. 3, 2011: 687-693.

Halaman ini sengaja dikosongkan

Biografi Penulis

Rizqi Putri Nourma Budiarti, menempuh pendidikan S1 pada Teknik Sistem Komputer, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya dan sekarang sedang melanjutkan pendidikan di Magister di Institut Teknologi Sepuluh Nopember Surabaya, Fakultas teknologi Industri, Jurusan Teknik Elektro, Bidang Keahlian Jaringan Cerdas Multimedia. Pernah bekerja sebagai engineer untuk perusahan provider Internet dan Telekomunikasi pada tahun 2005, dan bekerja pada PT. Infomedia Solusi Humanika pada tahun 2013 hingga 2015. Pada saat menempuh pendidikan Magister, penulis menerima Beasiswa LPDP – Beasiswa Tesis. Bidang penelitian penulis antara lain *Internet of Things, BigData dan Machine Learning*. Penulis dapat dihubungi pada Email: rizqi.putri.nb@gmail.com

Daftar riwayat pendidikan:

1. SDN Randuagung II Gresik
2. SMPN 1 Gresik
3. SMAN 1 Gresik
4. D3 Teknik Telekomunikasi Politeknik Elektronika Negeri Surabaya
5. S1 Teknik Sistem Komputer, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya

Daftar publikasi:

1. "*Implementation Naked Object Detection on Firefox Internet Browser using Cascade of Boosted Classifiers Based on Haar Like Features Algorithm.*", In the 18th Indonesian Scientific Conference in Japan, 2010.
2. "*Web Scraping for Automated Water Quality Monitoring System: A case study of PDAM Surabaya*", ISITIA 2016 in Lombok, 2016.

2016 International Seminar on Intelligent Technology and Its Application

Web Scraping for Automated Water Quality Monitoring System: A case study of PDAM Surabaya

Rizqi Putri Nourma Budiarti*, Nanang Widyatmoko†, Mochamad Hariadi‡ and Mauridhi Hery Purnomo§
*Graduate Student of Electrical Engineering, Institute of Technology Sepuluh November, Surabaya, Indonesia
†Department of Maintenance, PDAM Surabaya, Surabaya, Indonesia
‡§Department of Multimedia and Network Engineering, Institute of Technology Sepuluh November, Surabaya, Indonesia
Email: *rizqi.putri14@mhs.ee.its.ac.id, †ms.pemeliharaan@gmail.com, ‡mochar@ee.its.ac.id, §hery@ee.its.ac.id

***Abstract*—The need for a better online water quality monitoring system to control and provide drinking water treatment processes is proven. Water utility company in Surabaya that called as PDAM Surabaya has few reservoirs in their water supply system which are being monitored by WTW IQ SensorNet 2020 XT. However, on that sensor devices while it can provide some of the water quality parameters value information but the sensor is passive and the internal data is still stored in the sensor itself. To solve the problem, we proposed an application of data logger to manage data collections online water quality monitoring system by using web scraping. This application built by using Python language, the application is able to collect data from the sensor by using web scraping. We have utilized BeautifulSoap library and store the data in SQL. In these research, our experiment shows the performance of the application by measuring the accuracy of data that has been scraped from sensors are around 99%, the error rate less than 1%, and MSE (Mean Square Error) around 0.35%. We also calculated the growth of the data size that is gradually increasing and also measured the correlation between the data size with the delay which shows the value of our data is around 0.0000002739. Its means that our application is able to work in real-time and the delay is not affected by the size of data.**

***Index Terms*—Web scraping, water monitoring sensor, online real-time**

I. Introduction

The need for a better online water quality monitoring system to control and provide drinking water treatment processes is proven. Nowadays, the laboratory methods are too slow response in operational methods than the online measurement and also can not preserve a better quality of public health and environment preservation in real time requirement treatment solution. The necessity of rapidly identify the contamination because of the consequences to human health and the environment. In recent years, many researches develop an online water quality monitoring capability which is can provide an early warning of water pollution events[1][2][4]. Ensuring an appropriate and timely response is the right way in detecting contamination of drinking water supplies in real time. The necessity of real-time response and monitoring system should be assessed by water utilities and management area on a case by case basis based on there requirements of an individual water treatment company and environmental agency. Commonly, water quality parameter including pH, dissolve oxygen (DO), turbidity (NTU), temperature, and chlorine are monitored by using the particular instruments of online water equipment[3]. Water utility company using online monitoring equipment as early detection warning system at all stages including intake protection, water treatment, and distribution systems. These are used by early warning environment system and also for contaminant detection of drinking water supply process control and regulatory compliance. In some research, Storey et al have been reviewed and compared several equipment of water quality sensor from many manufactures[4].

Water utility company in Surabaya that called as PDAM Surabaya has few reservoirs in their water supply system which are being monitored by WTW IQ SensorNet 2020 XT. The reservoir has been placed in 5 areas, there are Ngagel I, Ngagel II, Ngagel III, Karang Pilang I, and Karang Pilang II. The WTW IQ SensorNet 2020 XT that used by PDAM Surabaya has many features including Software terminal MIQ/T 2020, networking system, transmission of current measured values directly to the PC, transmission of stored values in offline mode

978-1-5090-1709-6/16/$31.00 ©2016 IEEE

in MIQ/T 2020 for temporary term use for storage [5][6]. However, the sensor provided the information of the water quality parameter's value but it still passively because the internal data is still stored in the sensor itself. When it comes the urgent of viewing data history, to get the data logged the operator must manually retrieve its data by using USB memory. To solve the problem, we proposed an application of data logger to manage data collections online water quality monitoring system by using web scraping.

A range of different techniques has been developed and applied by using web scraping. The application of web scraping technique has been used in many researches for example to exploiting web scraping to approach a collaborative filtering-based web advertising[7]. Some studies also using it in collecting data on consumer electronics and airfares for Italian HICP compilation[8]. Web scraper has also been used to automate the download and collation of Canadian climate data[9]. Web scraping is a technique that can be used when you have a web development background and need to make the crawler download a web page and then achieve some data by extracting and parsing the data from each web page[10]. In this paper, the application built by using Python programming and in this system divided into two stages. The first stage is collecting the data using web scraping tools and the second stage is store the data into the database system. Generally, web scraping tools are typically divided into two main jobs, the first job is having access to sensors that containing the essential data, and the second job is to acquire a structured view of the data information that contained in the retrieved of HTML document pages by utilized Beautiful Soap library. The second stages, the information from web scraping is managed into columns and rows by using PostgreSQL. In these research, we experiment to determining the performance of the application.

The remainder of this paper that we presented is organised as follows. Section I describes introduction, Section II describes system design that discussed into five parts , Section III describes the implementation that divided into two parts , Section IV describes experimental result to evaluate the performance of web scraping, and then Section V describes conclusions of the experiment result.

II. System Design

A. Geographical location of PDAM Surabaya

There are three locations of PDAM Surabaya water treatment area as the study area. The first area

Fig. 1. PDAM Surabaya

called PDAM Surabaya as the data center of the data in all water drinking treatment plant. The location is at 7°17'58.9"S 112°44'50.3"E, which is located along the central coast of Surabaya City, East Java Province, Indonesia. The second area called PDAM IPAM Karangpilang as the place where the reservoir is located. The location is at 7°20'39.6"S 112°40'50.9"E, which is located near the side of Kali Surabaya river. The third area called PDAM IPAM Ngagel as the other place where the reservoir is located. The location is at 7°17'59.3"S 112°44'28.6"E, which is also located in near with Kali Surabaya river. PDAM Surabaya uses raw water from Kali Surabaya river for water drinking treatment plant. Kali Surabaya river flowing from the start point of Mlirip, Mojokerto City and passing through the area Wringin Anom, Driyorejo and getting to Surabaya City, the which is located between 112°30'E-112°45'E and 7°17'15S-7°17'25S. Each area as shown in Fig. 1.

In this research, the water quality indicators were monitored in IPAM Ngagel and IPAM Karang Pilang, but we use IPAM Ngagel II area base of IPAM Ngagel and IPAM Karang Pilang I, II area base of IPAM Karang Pilang.

B. Network Topology Sensor

The topology of this network sensor and the routing mechanisms having a significant influence on network analyze such as performance, reliability and also scalability. This network supports interpersonal systems and having advantages : first, high availability of huge computing and storage resources and second, the network connectivity will be increased. This topology designates the network of connectivity and the internet accessible host to the sensor.

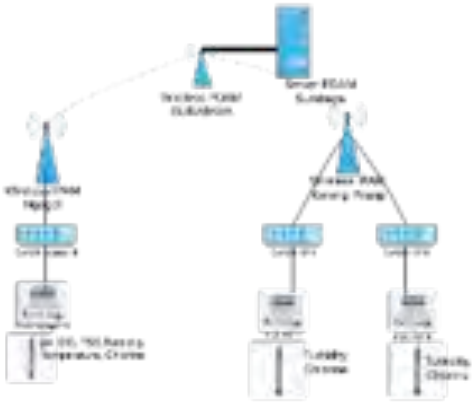


Fig. 2. Network topology Sensor

We used network topology as shown in Fig.2. The WTW IQ SensorNet 2020 XT controller are installed in each area. It has several water quality indicator included pH, dissolve oxygen (DO), turbidity (NTU), total suspended solid (TSS), water temperature and also chlorine. In Ngagel II, included all variables, but in

Fig. 3. WTW IQ SensorNet 2020 XT Controller

Karang Pilang I and II monitored water quality variables included turbidity and chlorine. The WTW IQ SensorNet 2020 XT as it shown in Fig. 3.

C. Beautiful Soup Method

Beautiful Soup is a Python package library that can be used for pulling data out of HTML and XML documents. Beautiful Soup creates a parse tree by modifying, navigating and searching with parsing strategies in idiomatic ways for parsed pages to extract data from HTML[10][11]. Commonly, this library technique is really a popular module that parses a web page and then provides a convenient interface to navigate content. Beautiful Soup using as a Python library for simulating a web browser season. It supports web form, web cookies, web link navigation and uses as a scraping solution on the website content. In this research we utilised Beautiful Soap to scrap data from sensors and parse the information to get sensor values.

D. PostgreSQL

PostgreSQL is an advanced object-relational database management system (ORDBMS) with inherently using SQL's query structure. PostgreSQL is widely considered the most purposeful way for open source database in the world that it freely to obtain the source code, use the program and modify it to fit the particular needs [13][14]. We use the database system with PostgreSQL to begin with setting up into the database server which is the database server located in PDAM Surabaya and we also setting the database users and groups, authentication and using it as a database-management system.

E. Linux Crontab

The crontab is named from the cron derives after chronos, Greek for time process and also tab stands for the table of cron[15][16]. Which is a list of commands that found in Unix and Unix-like operating systems and it used to schedule the job scheduler cron to be executed periodically. Linux support it and it called cron. Crontab has function to running tasks at regular intervals in the background and it can create backups, synchronize files, and schedule updates. In this research, we used it as job scheduler cron to execute task every minute.

III. Implementation

Data in the web based application are data intensive. In this section, we present the full pipeline of web scraping for automated water monitoring sensor, from retrieving and parsing each seed, testing and accessing the data logger. The implementation divided into two parts :

A. Sensor Web Output

In this paper, we used the WTW IQ SensorNet 2020 XT. The sensor provided the information of the water quality parameter's value but it still passively because the internal data is still stored in the sensor itself and can not send its information data to the server. When it comes the urgent of viewing data history, to get the data logger the operator must manually retrieve its data by using USB memory. We utilized web information from the sensor output that connected in the Local Area Network (LAN).

Fig. 4. Screenshot of Sensor web output

To perform data in the element partitioning on the web, we use Python which uses Beautiful Soup to resolve bad markup and then stores the result in a tree using the element tree module. Commonly, because it supports more powerful in searching and traversing, we uses the element tree module over beautiful soup. In this process, we use the sensor web output that defined in the Fig. 4, and used the html document as an ongoing seed document to create the automated water monitoring sensor with web scraping.

B. Web Scraping

Fig. 5 shows the implementation of web scraping for water monitoring sensor logger that consists of:

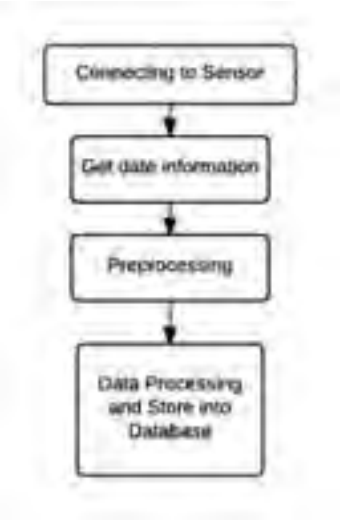


Fig. 5. Methode of web scraping for water monitoring

1) Connecting to Sensor: The water data information that provided by the sensor can be accessed by the required user only. The sensor provides a passive web interface that can not send the information data to the server automatically. In here, we use URLopen to access the sensor from server and parses the information from webpages. In here, the urllib.urlopen command is needed to make the browser pages navigate to the URL provided. These actions are identical to the human user while typing URL in the address bar of the browser.

2) Get date information: In this webpages that consist of the water quality measurement data will be shown by giving the URL and the data displayed in each as a list of strings. After we uses URLopen to access sensor, we like to get the data information by strip parent about date information and takes the form of partitioning in the html document that shown in Listing 1.

Listing 1. The parent form of html document

```
<p>
Controller: NGAGEL 2<br />
Serial: 09430346<br />
Software: 3.59<br />
Time: 11 Apr 2016 08:06:44
</p>
```

3) Preprocessing: In the preprocessing we used table manipulation, the steps for table manipulation are as follow : 1. Search the table part, because the data structure is in the table part. 2. Eliminate the table header part with beautiful soup extract using the soup.t.head.extract command. Furthermore, the parts that are not needed do not interrupt the process. 3. Taking the data row of the sensor, because inside of processor sensor in each sensor there are several modules and the information is written in the row format. In here, we need to get information from tr (table row).

4) Data Processing and Store Into Database: In the data processing, we extract the data from table element. By utilized looping function we scrap each element into array of data. After that, we use insert data query to store data into the database. Listing 2 shows about data processing and store into database method.

Listing 2. Data processing and storing into database

```
con = psycopg2.connect(database='
   sensor_db', user='sensor_user')
cur = con.cursor()

for tr in rows:
        cols = tr.findAll('td')
        text_data = []
```

```
for td in cols:
        text = ''.join(td)
        utftext = str(text.encode
            ('utf-8'))
        text_data.append(utftext)
text = ','.join(text_data)
print(date+" "+ text+ '/')
data = text.split(",",10)

query = "INSERT INTO coba(sid,
    state,type,serno,name,mval
    ,munit,mpara,sval,sunit,
    info) VALUES (%s,%s,%s,%s,
    %s,%s,%s,%s,%s,%s,%s,%s);"
values = (data[0], data[1], data
    [2], data[3], data[4], data
    [5], data[6], data[7], data
    [8], data[9], data[10])

cur.execute(query, values)
con.commit()
```

IV. Experimental Result

In the experiment process, we conducted our experiment to estimate the performance of web scraping when creating data logger from automated water monitoring sensor. This web scraping is being used in PDAM Surabaya application which are currently online in the PDAM Surabaya local area sites. There are tree locations that we monitored, and in each location has several sensors to monitor water condition. In Table I we describe the type of sensor in each location.

TABLE I
Type of Sensor

Locations	# of Sensor	Types of sensor
Ngagel II	6	Turb, Chlor, TSS, pH, O2 + Temp, Turb
Karang Pilang I	3	Chlor, Turb, Turb
Karang Pilang II	2	Turb, Chlor

The experiments were started from March 10th, for evaluation we use the data from March 10th, 2016 until April 5th, 2016. We have collected the data from the sensor for every minute, its means that we have collected data around 37.300 data. In here, we were starting to scrape the collected data. For the seed of data, we automatically scraped the text chunk from sensors. We calculated number of data that has been taken from sensor in each location.

We have calculated the expected data at IPAM Ngagel II around 37343 data for each sensor, but the data logger having around 37319 data which means the accuration of data collection is 99.94%. We also have calculated the expected data at IPAM Karang Pilang I around 37344 data for each sensor, but the data logger having around 37017 datas which means the accuration of data collection is 99.12%. In other areas, we have calculated the expected data at IPAM Karang Pilang II around 37346 data for each sensor, but the data logger having around 37311 datas which means the accuration of data collection is 99.90%. Error rate measures the number of errors data divided by the total number of expected data during a studied time interval. The error rate is a unitless performance measure. In here, we expressed in a percentage. The error rate calculation is defined as :

$$ErrorRate = \frac{|datasensor - dataexpected|}{|dataexpected|} \times 100\% \quad (1)$$

TABLE II
Error Rate of Data Collected

Locations	Start	End	#Data	# Data expected	Error rate
Ngagel II	10/03/2016 12:23	05/04/2016 10:46	37319	37343	0.06%
Karang Pilang I	10/03/2016 12:23	05/04/2016 10:47	37017	37344	0.88%
Karang Pilang II	10/03/2016 12:23	05/04/2016 10:49	37311	37346	0.1%

In Table II we describe the data sensor in each location, the data expected as the theoretical value data that we can get depend on calculation of the timing schedule that we set for every minute, and also the error rate calculation of each location.

In here, we used the result of the error rate to calculate the Mean Square Error (MSE), by using the calculation is define as :

$$MSE = \frac{\Sigma \mathrm{ErrorRate_x}}{\Sigma \mathrm{x}} \times 100\% \quad (2)$$

Using the calculation of MSE, we get the result MSE = 0.35%.

We also conducted experiment process to evaluate the growth of data sensor capacity in each location.

We have measured the data growing every five days start from March 11th, 2016 00:00 until April 4th, 2016 23:59. As Fig. 6 indicates, The growth of data capacity in Ngagel II provides consistently is bigger than Karang Pilang I and Karang Pilang II because the number of its sensors which has more than any other locations. In here, we also measured the correlation between the data

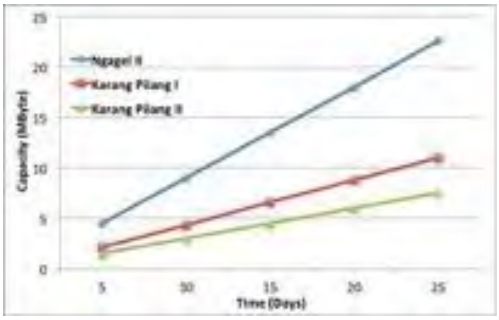


Fig. 6. The Growth of Data Capacity

size per character and the delay to fetch the data size in character is defined as

$$Cor(b,d) = \frac{\mathrm{Cov(b,d)}}{\sqrt{\mathrm{Var(b)} \times \mathrm{Var(d)}}} \quad (3)$$

where Cov(b,d) is covariance between data size in bytes per character and delay, Var(b) is the variance of size data and Var(d) is the variance of the time delay between application starting running to fetch data with the time when application store the data into SQL. Cor(b,d) shows correlation whether the delay to fetch a data to cache varies across data of similar size. if Cor(b,d) value to be close to 1 indicates that delay is affected by size of data. On the other hand, if the Cor(b,d) value to be close to 0, it means almost there are no correlation between size and delay. Our measurement value of Cor(b,d) on our data is 0.0000002739, which is relatively low. The value is low because each time our application fetch the data from the sensor, the size of data is relatively small and also the time delay also very small less than 1 second. It means that our application is able to work almost in real-time.

V. Conclusions

The main contribution of our paper is that our water monitoring application is able to collect data from sensors smoothly. From our experiments shows that the accuracy of data that has been scraped from sensor are around 99%, and error data rate are less then 1%.

In summary, the experimental results indicate that the correlation between data size with delay is relatively low and we had value to close to 0. It means that the delay is not dependent on size, it means also that our application is able to work almost in real-time.

In the future, we have to prepare the storage because the data size is gradually increased. Because of that, the integration of the automated water monitoring system with Big Data architecture which can handle wide dataset is one of our future plans. With Big Data architecture it will be able to process the dataset larger than given memory and also provide a particular method to select the fastest, scalable, and simplest of training the large data.

Acknowledgment

I would like to thanks to PDAM Surabaya Team from Department of Maintenance and Department of IT for support me during this research. I am also grateful to the LPDP scholarship of Republic Indonesia Endowment Funding in Education with number: PRJ-547/LPDP.3/2016 for this research project.

References

[1] W. Yang, J. Nan, and D. Sun, *An online water quality monitoring and management system developed for the Liming River basin in Daqing, China*, Journal of Environmental Management, vol. 88, no. 2, pp. 318-325, Jul. 2008.

[2] C. A. Stedmon, B. Seredy ska-Sobecka, R. Boe-Hansen, N. Le Tallec, C. K. Waul, and E. Arvin, *A potential approach for monitoring drinking water quality from groundwater systems using organic matter fluorescence as an early warning for contamination events*, Water Research, vol. 45, no. 18, pp. 6030-6038, Nov. 2011.

[3] Frey, MM. and Sullivan. L. , *Practical application of on-line monitoring.*, AWWA Research Foundation Report. (Denver), 2004.

[4] M. V. Storey, B. V.D. Gaag and B. P. Burns, *Advances in on-line drinking water quality monitoring and early warning systems.* Journal Elsevier:Water Research, 2010.

[5] IQ SensorNet, *WTW GmbH - Stationary meters - IQ SENSOR NET*. [Online]. Available: http://www.wtw.com/en/products/product-categories/stationary-meters/iq-sensor-net.html. [Accessed: 20-Apr-2016].

[6] IQ SensorNet, *WTW IQ SensorNet 2020 XT Controller - Xylem Analytics UK*. [Online]. Available: http://www.xylemanalytics.co.uk/productsdetail.php?WTW-IQ-SensorNet-2020-XT-Controller-84. [Accessed: 20-Apr-2016]

[7] E. Vargiu and M. Urru, *Exploiting web scraping in a collaborative filtering- based approach to web advertising*, Artificial Intelligence Research, vol. 2, no. 1, Nov. 2012.

[8] F. Polidoro, R. Giannini, R. L. Conte, S. Mosca, and F. Rossetti, *Web scraping techniques to collect data on consumer electronics and airfares for Italian HICP compilation*, Statistical Journal of the IAOS, vol. 31, no. 2, pp. 165-176, May 2015.

[9] C. Bonifacio, T. E. Barchyn, C. H. Hugenholtz, and S. W. Kienzle, *CCDST: A free Canadian climate data scraping tool*, Computers & Geosciences, vol. 75, pp. 13-16, Feb. 2015.

[10] R. Lawson, *Web Scraping with Python.*, 1st ed. Birmingham, Mumbai: Pack Publishing, 2015.

[11] Beautiful Soup 4, *Beautiful Soup Documentation - Beautiful Soup 4.4.0 documentation*. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[12] Beautiful Soup 3, *Beautiful Soup documentation*. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs3/documentation.html. [Accessed: 30-Apr-2016].
[13] J. D. Drake and J. C. Worsley, *Practical PostgreSQL*. O'Reilly Media, 2002.
[14] PostgreSQL, *Postgresql*, 2012, [Online]. Available: http://www.postgresql.org.
[15] Linux crontab, *Linux and UNIX crontab command help and examples*. [Online]. Available: http://www.computerhope.com/unix/ucrontab.htm. [Accessed: 29-Apr-2016].
[16] Cron, *CronHowto - Community Help Wiki*. [Online]. Available: https://help.ubuntu.com/community/CronHowto. [Accessed: 29-Apr-2016].