

**SISTEM KONTROL DAN PEMANTAUAN KUALITAS AIR
SUNGAI MAHAKAM MENGGUNAKAN NAÏVE BAYES DAN
SIMPLE EXPONENTIAL SMOOTHING**

CONTROL AND MONITORING SYSTEM WATER QUALITY
OF MAHAKAM RIVER USING NAÏVE BAYES AND SIMPLE
EXPONENTIAL SMOOTHING

HIDAYATUL MUTTAQIEN



PROGRAM PASCASARJANA

STMIK HANDAYANI

MAKASSAR

2019

TESIS

**SISTEM KONTROL DAN PEMANTAUAN KUALITAS AIR
SUNGAI MAHAKAM MENGGUNAKAN NAÏVE BAYES DAN
SIMPLE EXPONENTIAL SMOOTHING**

Disusun dan diajukan oleh:

HIDAYATUL MUTTAQIEN

2018130001

Telah dipertahankan di depan Panitia Ujian Tesis

Pada Tanggal 10 Desember 2019

Dan dinyatakan telah memenuhi syarat

Menyetujui

Komisi Penasehat


Dr. Ir. Zahir Zainuddin, M.Sc

Ketua


Dr. Abdul Latief Arda, M.Si, M.Kom

Anggota

Ketua Program Studi

Sistem Komputer




Dr. Ir. Zahir Zainuddin, M.Sc

Direktur Program Pascasarjana

STMIK Handayani,




Dr. Eng Yuyun, S.Kom, MT.

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Hidayatul Muttaqien


Nomor Mahasiswa : 2018130001

Program studi : Sistem Komputer

Menyatakan dengan sebenarnya bahwa tesis/disertasi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan tulisan atau pemikiran orang lain. Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tesis/disertasi ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 10 Desember 2019

Yang menyatakan,



Hidayatul Muttaqien

PRAKATA

Assalamu Alaikum Warrahmatullahi Wabarakatuh.

Segala puji bagi Allah yang Maha Mengetahui segala sesuatu yang ada pada alam semesta ini. Atas berkah dan rahmat-Nya sehingga penelitian ini dapat diselesaikan dalam rangka mencapai gelar Magister. Shalawat serta salam kepada Nabi Muhammad Shalallahu alaihi wassalam, beserta keluarganya, sahabatnya dan para pengikutnya hingga akhir zaman.

Pada Kesempatan ini penulis penulis ingin mengucapkan terima kasih sebesar-besarnya kepada ayahanda tercinta Ir. H. Sidik Tarmidji, Ibunda tercinta Hj. Ida Hairina, Istri tercinta Andi Hutami Endang, M.Kom dan seluruh pihak yang telah membantu dalam penyusunan ini terlebih lagi kepada bapak **Dr. Ir. Zahir Zainuddin, M.Sc** dan **Dr. Abdul Latief Arda, M.Si, M.Kom** selaku pembimbing dalam penyusunan penelitian ini, tak lupa pula penulis haturkan terima kasih setinggi – tingginya kepada:

1. Bapak Dr. Eng. Yuyun, S.Kom, MT selaku direktur Program Pascasarjana STMIK Handayani.
2. Bapak Dr. Ir. Zahir Zainuddin, M.Sc selaku Ketua Program Studi Sistem Komputer STMIK Handayani Makassar.
3. Segenap Dosen yang telah memberikan ilmunya kepada penulis selama mengikuti Pendidikan di Program Magister STMIK Handayani Makassar.

4. Adeeva Afshien Myesha dan Azkayra Yasna Malika selaku putri – putri tersayang yang selalu memberikan dorongan, semangat dan do'a kepada penulis
5. Seluruh saudaraku Miftahul Jannah, S.Pd.I., Miftahurrizki, S.H.I., Nur Fauzan Hidayat, S.Kom. dan Hidayatul Muhtadin yang selalu menjadi pendukung dan teman diskusi penulis.
6. Segenap staf dan rekan – rekan pada Program Magister STMIK Handayani
7. Semua pihak yang tidak dapat disebutkan satu-persatu, atas semua bantuan dan dukungan yang telah diberikan kepada penulis.

Walaupun dalam penelitian dan penyusunan tesis ini, kami telah berusaha secara maksimal, namun jika masih ada kekurangan baik dari segi penulisan maupun dari segi isi dan metodologi penelitian kami mohon maaf dan mohon petunjuk berupa kritik dan saran, demi menjadi intropeksi dan perbaikan kami dalam proses penyusunan dan penelitian berikutnya agar lebih baik lagi. Akhir kata saya selaku penyusun mengucapkan terima kasih.

Makassar, 10 Desember 2019

PENULIS

SISTEM KONTROL DAN PEMANTAUAN KUALITAS AIR SUNGAI MAHAKAM MENGGUNAKAN NAÏVE BAYES DAN SIMPLE EXPONENTIAL SMOOTHING

ABSTRAK

Hidayatul Muttaqien, 2019. Air sungai memiliki peran penting dalam kehidupan manusia. Banyak peradaban di dunia mulai berkembang di daerah tepi sungai. Pemantauan terhadap kualitas air sungai yang dilaksanakan secara periodik dan berkesinambungan merupakan salah satu upaya pengelolaan lingkungan. Sistem kontrol perangkat pompa yang mengacu pada baku mutu air perlu dilakukan agar sumber air yang akan diolah sesuai dengan ketentuan – ketentuan yang ada. Hal ini menghindari dampak – dampak yang tidak baik apabila air tidak sesuai dengan baku mutu air.

IoT (Internet of Things) merupakan solusi yang memanfaatkan internet yang terhubung secara terus menerus untuk melakukan Kontrol perangkat pompa sehingga dapat dengan cepat melakukan tindakan sesuai dengan pemantauan air sungai. Protokol komunikasi berperan penting dalam meningkatkan kinerja sistem kontrol dalam penelitian ini *MQTT (Message Queue Telemetry Transport)* digunakan sebagai protokol komunikasi karena efisien dalam penggunaan *bandwidth*, *reliable*, dan sangat ringan. *Naïve Bayes* merupakan salah satu teknik probabilistik sederhana yang dapat mengklasifikasikan kondisi kualitas air berdasarkan pengukuran parameter - parameter kualitas air yaitu *pH*, *TSS (Total Suspended Solid)*, *(Dissolved Oxygen)*, *Nitrate*. *Simple Exponential Smoothing* digunakan untuk memperkirakan kualitas air sungai secara kuantitatif dan menjadi sumber informasi untuk keputusan selanjutnya.

Penerapan *IoT* dengan protokol *MQTT* menghasilkan sistem kontrol yang *responsive* dengan tingkat akurasi mencapai 100%. *Naïve Bayes* mampu menentukan kondisi kualitas air dengan baik serta cepat sehingga dapat memberikan keputusan sistem kontrol yang cepat pula. *Simple Exponential Smoothing* mampu memperkirakan parameter – parameter kualitas air sungai dengan baik. Hasil pengujian menggunakan *MAPE (Mean Absolute Percentage Error)* menunjukkan rata – rata error untuk pH sebesar 4.28%, TSS sebesar 7.94%, DO sebesar 4.42% dan Nitrate 8.68% hal ini menunjukkan hasil yang cukup baik.

Kata kunci: *Internet of Things (IoT)*, *Message Queue, Telemetry, MQTT, Naïve Bayes, Simple Exponential Smoothing, Kualitas Air.*

CONTROL AND MONITORING SYSTEM WATER QUALITY OF MAHAKAM RIVER USING NAÏVE BAYES AND SIMPLE EXPONENTIAL SMOOTHING

ABSTRACT

Hidayatul Muttaqien, 2019. River water has an important role in human life. Many civilizations in the world began to develop in the river. Monitoring of river water quality which is carried out periodically and continuously is one of the efforts to manage the environment. Pump device control system that refers to water quality standards needs to be implemented so that the source of water to be treated is in accordance with the existing provisions. This avoids adverse effects if the water does not meet water quality standards.

IoT (Internet of Things) is a solution that utilizes a continuously connected internet to control pumping devices so that they can quickly action in accordance with river water monitoring. Communication protocol an important role in improving the performance of the control system, in this research MQTT (Message Queue Telemetry Transport) is used as a communication protocol because it is efficient in the use of bandwidth, reliable, and very lightweight. Naïve Bayes is a simple probabilistic technique that can classify water quality conditions based on measurement of water quality parameters, namely pH, TSS (Total Suspended Solid), (Dissolved Oxygen), Nitrate. Simple Exponential Smoothing is used to estimate river water quality quantitatively and becomes a source of information for future decisions .

The implementation of IoT with the MQTT protocol results in a responsive control system with 100% accuracy. Naïve Bayes is able to determine water quality conditions well and quickly so that it can provide a quick control system decision as well. Simple Exponential Smoothing is able to estimate the parameters of river water quality well. The test results using MAPE (Mean Absolute Percentage Error) showed an average error for pH of 4.28%, TSS of 7.94%, DO of 4.42% and Nitrate of 8.68%, this showed quite good results

Keyword: Internet of Things (IoT), Message Queue, Telemetry, MQTT, Naïve Bayes, Simple Exponential Smoothing, Water Quality

DAFTAR ISI

PRAKATA	ii
ABSTRAK	iv
ABSTRACT	v
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Rumusan Masalah	5
C. Tujuan Penelitian.....	5
D. Manfaat Penelitian.....	5
E. Batasan Masalah.....	6
F. Definisi dan Istilah	6
BAB II TINJAUAN PUSTAKA.....	9
A. Tinjauan Umum Tentang Air.....	9
B. Sungai Mahakam	11
C. Proses Pengolahan dan Pendistribusian Air	12
D. Sistem Kontrol berbasis <i>IoT (Internet of Things)</i>	17
E. <i>MQTT (Message Queuing Telemetry Transport)</i>	18
F. Metode Naive Bayes	21
G. <i>Simple Exponential Smoothing</i>	22
BAB III METODE PENELITIAN.....	24
A. Jenis Penelitian	24
B. Waktu dan Lokasi Penelitian	27
C. Tahapan penelitian	27
D. Perancangan Sistem	28
E. Kebutuhan Perangkat Keras	29
F. Kebutuhan Perangkat Lunak	34
G. Use Case.....	34

H. Pengujian Sistem	35
BAB IV HASIL DAN PEMBAHASAN.....	37
A. Sistem Kontrol berbasis <i>IoT</i> protokol <i>MQTT</i> dengan Algoritma <i>Naïve Bayes</i>	37
B. Skema Komunikasi Mikrokontroller dengan <i>MQTT</i> Server.....	38
C. Menentukan Kualitas Air Menggunakan Algoritma <i>Naïve Bayes</i> ...	41
D. <i>Forecasting</i> parameter – parameter Kualitas Air menggunakan <i>Simple Exponential Smoothing</i>	46
E. Antarmuka Sistem Kontrol dan Pemantauan.....	48
F. Pengujian Sistem	51
BAB V PENUTUP	55
A. Kesimpulan.....	55
B. Saran.....	56
DAFTAR PUSTAKA.....	57

DAFTAR GAMBAR

Gambar 1. Posisi Intake PDAM Tirta Kencana Samarinda.....	15
Gambar 2. Intake Teluk Lerong Samarinda	15
Gambar 3. Pipa Saluran ke Pengolahan.....	16
Gambar 4. Posisi intake pada <i>Googlemaps</i>	16
Gambar 5. <i>MQTT</i> Skema (Jerald, Anand, & Deepika, 2019)	19
Gambar 6. Skema Broker dalam <i>MQTT</i> (Jerald et al., 2019).....	20
Gambar 7. Metode <i>Waterfall</i> (Bassil, 2012)	25
Gambar 8. Skema Sistem Kontrol dan Monitoring	29
Gambar 9. Use Case sistem.....	35
Gambar 10. Skema Komunikasi Mikrokontroller ke <i>MQTT Server</i>	38
Gambar 11. Rangkaian Sistem Kontrol.....	39
Gambar 12. Flowchart Program Tertanam pada Mikrokontroller	40
Gambar 13. Flowchart Naive Bayes.	42
Gambar 14. <i>Code</i> langkah 1 (satu) Implementasi <i>Naive Bayes</i>	42
Gambar 15. <i>Code</i> Langkah 2 (Dua) Implementasi <i>Naive Bayes</i>	43
Gambar 16. <i>Code</i> Langkah 3 Implementasi <i>Naive Bayes</i>	43
Gambar 17. <i>Code</i> Langkah 4 (empat) Implementasi <i>Naive Bayes</i>	44
Gambar 18. Pemberitahuan Kualitas Air Melalui SMS dan Email.....	45
Gambar 19. <i>Code</i> mengontrol pompa.....	45
Gambar 20. <i>Flowchart Simple Exponential Smoothing</i>	46
Gambar 21. Grafik Proses Peramalan pH	47
Gambar 22. Grade SSL Konfigurasi	49
Gambar 23. Halaman Login.....	49
Gambar 24. Halaman Pemantauan dan Kontrol	50
Gambar 25. Halaman peramalan parameter kualitas air.	51
Gambar 26. Grafik Pengujian <i>Simple Exponential Smoothing</i>	54

DAFTAR TABEL

Tabel 1. Kriteria Mutu Air Berdasarkan Kelas (PERATURAN PEMERINTAH REPUBLIK INDONESIA, n.d.)	11
Tabel 2. Spesifikasi Sensor	32
Tabel 3. Kebutuhan Perangkat Lunak.....	34
Tabel 4. Percobaan Sistem dan Mesin Pompa.....	52
Tabel 5. Confusion Matrix	53

BAB I

PENDAHULUAN

A. Latar Belakang

Air sungai memiliki peran penting dalam kehidupan manusia. Banyak peradapan di dunia mulai berkembang di daerah tepi sungai. Manfaat sungai bagi kehidupan tidak pernah lepas dari kebutuhan makhluk hidup akan air. Di Provinsi Kalimantan Timur terdapat Sungai Mahakam dengan panjang ± 920 Km. (*PERATURAN PEMERINTAH REPUBLIK INDONESIA, No. 82 Tahun 2001*) Bahwa air merupakan salah satu sumber daya alam yang memiliki fungsi sangat penting bagi kehidupan dan perikehidupan manusia, serta untuk memajukan kesejahteraan umum, sehingga merupakan modal dasar dan faktor utama pembangunan. Bahwa air merupakan komponen lingkungan hidup yang penting bagi kelangsungan hidup dan kehidupan manusia dan makhluk hidup lainnya. 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. Di satu pihak, usaha dan atau kegiatan manusia memerlukan air yang berdaya guna, tetapi di lain pihak berpotensi menimbulkan dampak negatif, antara lain berupa pencemaran yang dapat mengancam ketersediaan air, daya guna, daya dukung, daya tampung, dan produktivitasnya. Agar air dapat bermanfaat secara lestari dan pembangunan dapat berkelanjutan, maka dalam pelaksanaan

pembangunan perlu dilakukan pengelolaan kualitas air dan pengendalian pencemaran air. Dampak negatif pencemaran air mempunyai nilai (biaya) ekonomik, di samping nilai ekologis, dan sosial budaya. Upaya pemulihan kondisi air yang cemar, bagaimanapun akan memerlukan biaya yang mungkin lebih besar bila dibandingkan dengan nilai kemanfaatan finansial dari kegiatan yang menyebabkan pencemarannya. Demikian pula bila kondisi air yang cemar dibiarkan (tanpa upaya pemulihan) juga mengandung ongkos, mengingat air yang cemar akan menimbulkan biaya untuk menanggulangi akibat dan atau dampak negatif yang ditimbulkan oleh air yang cemar. Berdasarkan definisinya, Pencemaran air yang diindikasikan dengan turunnya kualitas air sampai ke tingkat tertentu yang menyebabkan air tidak dapat berfungsi sesuai dengan peruntukannya. Yang dimaksud dengan tingkat tertentu tersebut di atas adalah baku mutu air yang ditetapkan dan berfungsi sebagai tolok ukur untuk menentukan telah terjadinya pencemaran air, juga merupakan arahan tentang tingkat kualitas air yang akan dicapai atau dipertahankan oleh setiap program kerja pengendalian pencemaran air.

Penetapan baku mutu air selain didasarkan pada peruntukan (*designated beneficial water uses*), juga didasarkan pada kondisi nyata kualitas air yang mungkin berada antara satu daerah dengan daerah lainnya. Oleh karena itu, penetapan baku mutu air dengan pendekatan golongan peruntukkan perlu disesuaikan dengan menerapkan pendekatan klasifikasi kualitas air (kelas air). Penetapan baku mutu air yang didasarkan

pada peruntukan semata akan menghadapi kesulitan serta tidak realistis dan sulit dicapai pada air yang kondisi nyata kualitasnya tidak layak untuk semua golongan peruntukan.

Dengan ditetapkannya baku mutu air pada sumber air dan memperhatikan kondisi airnya, akan dapat dihitung berapa beban zat pencemar yang dapat ditanggung adanya oleh air penerima sehingga air dapat tetap berfungsi sesuai dengan peruntukannya. Beban pencemaran ini merupakan daya tampung beban pencemaran bagi air penerima yang telah ditetapkan peruntukannya.

Pemanfaatan air untuk pemenuhan kebutuhan rumah di Kota Samarinda dan sekitarnya bersumber pada sungai Mahakam. Saat ini sudah dilakukan pemantauan kualitas air pada Sungai Mahakam berupa peletakkan sensor yang menghitung *pH*, *Salinity*, *Ammonium*, *Nitrate* dan lain – lain tetapi untuk pengolahan hasil pengukuran masih dilakukan secara manual serta tidak ada peringatan dini jika terjadi pencemaran sehingga air masih dapat tersebar ke masyarakat. Perlu dilakukan pemantauan yang real-time untuk memantau kualitas air, sehingga jika terjadi kontaminasi pada air dapat dengan cepat memberikan peringatan dini, atau mengontrol perangkat distribusi air tersebut (Kavi Priya, Shenbagalakshmi, & Revathi, 2017).

IoT (Internet of Things) merupakan solusi dalam memperluas manfaat dari konektivitas *internet* yang terhubung secara terus menerus (*Real*

Time)(Pawar & Ghumbre, 2016; A. Singh, Meshram, Gujar, & Wankhede, 2016; T. A. Singh & Chandra, 2018). *IoT* dapat dimanfaatkan untuk mengontrol seperangkat elektronik dari jarak jauh melalui media *internet*. Perangkat elektronik yang terhubung akan saling bertukar data dan informasi secara terus menerus menggunakan protokol *TCP/IP*. Proses komunikasi menggunakan protokol *TCP/IP* memiliki beberapa parameter untuk mengukur kualitasnya. Parameter tersebut adalah *latency*, *bandwidth*, *response time* dan *reliability*(Naik, 2017; T. A. Singh & Chandra, 2018). *Naïve Bayes Algorithm* merupakan teknik berbasis probabilistic sederhana yang dapat mengklasifikasikan kondisi kualitas air berdasarkan parameter-parameter pengukuran kualitas air. *Simple Exponential Smoothing* digunakan untuk memperkirakan kualitas air secara kuantitatif. Dengan memanfaatkan dua algoritma tersebut diharapkan dapat memberikan informasi kualitas air dan prediksi keadaan air sungai pada kondisi tertentu sehingga dapat digunakan untuk proses pengambilan keputusan dan tindakan apa yang akan dilakukan selanjutnya(Lutfi, Muttaqien, Apriliani, Zainuddin, & Yuyun, 2019).

Dari hal tersebut maka penelitian ini memberikan solusi dengan pemanfaatan *IoT* dan protokol komunikasi *MQTT* yang efisien dan memiliki reliabilitas yang baik serta pemanfaatan *Naïve Bayes Algorithm Simple Exponential Smoothing* diharapkan dapat memberikan informasi yang lebih cepat, tepat dan *realtime*.

B. Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat disimpulkan permasalahan yang akan diteliti yaitu bagaimana dapat membangun sistem kontrol berbasis *IoT* yang berkomunikasi menggunakan protokol komunikasi yang ringan, cepat dan memiliki konsumsi *bandwidth* yang kecil dan reliabilitas yang baik yaitu *MQTT (Message Queue Telemetry Transport)* dan membangun sistem monitoring menggunakan *Naïve Bayes Algorithm* dan *Simple Exponential Smoothing* untuk menghasilkan informasi kualitas air dan prediksi keadaan air sungai pada kondisi tertentu.

C. Tujuan Penelitian

Tujuan penelitian ini adalah :

1. Membangun sistem kontrol berbasis *IoT* dan *MQTT Telemetry transport* sehingga menghasilkan sistem yang handal, cepat dan memiliki reliabilitas yang baik.
2. Membuat sistem monitoring yang memanfaatkan *Naïve Bayes Algorithm* dan *Simple Exponential Smoothing* dalam proses menentukan kualitas air sungai dan memprediksi kualitas air sungai.

D. Manfaat Penelitian

Dengan adanya penelitian ini diharapkan dapat memberikan manfaat antara lain:

1. Memberikan referensi dalam pengembangan sistem control berbasis *IoT* dengan protokol komunikasi yang cepat, ringan dan reliabilitas yang baik.
2. Memberikan sistem monitoring yang dapat berguna bagi pengambil kebijakan terkait pemanfaatan air sungai Mahakam dengan memanfaatkan *Naïve Bayes Algorithm* dan *Simple Exponential Smoothing*
3. Terpenuhinya kebutuhan informasi bagi seluruh stakeholder tentang data kualitas air sungai Mahakam secara kontinyu (*Real Time*).

E. Batasan Masalah

Setelah dikemukakan latar belakang dalam penelitian ini, maka masalah yang akan diteliti adalah bagaimana membangun sistem control berbasis *IoT* menggunakan protokol komunikasi *MQTT* dan menyediakan sistem monitoring kualitas air sungai mahakam menggunakan *Naïve Bayes Algorithm* dan *Simple Exponential Smoothing* dengan parameter – parameter yang menjadi acuan adalah *pH*, *TSS (Total Suspended Solid)*, *DO (Dissolved oxygen)*, *Nitrate*.

F. Definisi dan Istilah

Agar memudahkan dalam memahami penelitian ini, maka dikemukakan definisi dan istilah dari beberapa kata yang dianggap perlu, sehingga

penelitian ini dapat lebih jelas dan tidak menimbulkan kesalahan dalam memahami penelitian ini. Beberapa definisi dan istilah kata yang dianggap perlu sebagai berikut:

1. **Sistem Kontrol** adalah Sistem kontrol adalah proses melakukan pengendalian terhadap sesuatu parameter agar berada pada batas – batas yang diinginkan
2. **Intake** adalah bangunan yang digunakan untuk memperoleh air dari sungai (atau mata air terbesar suatu daerah) sesuai dengan jumlah yang diperlukan.
3. **IoT (*Internet of Things*)** merupakan solusi dalam memperluas manfaat dari konektivitas internet yang terhubung secara terus menerus.
4. **MQTT** adalah protokol konektivitas machine-to-machine (M2M)/ *Internet of Things (IoT)* yang berbasis open source (Eclipse) dengan standar terbuka (OASIS) yang dirancang untuk perangkat terbatas dan bandwidth rendah, dengan latency tinggi atau berjalan pada jaringan yang tidak dapat diandalkan
5. **pH (*Power of Hydrogen*)** adalah derajat keasaman yang digunakan untuk menyatakan tingkat keasaman atau kebasaan yang dimiliki oleh suatu larutan.
6. **Total suspended solid atau padatan tersuspensi total (TSS)** adalah residu dari padatan total yang tertahan oleh saringan dengan ukuran partikel maksimal $2\mu\text{m}$ atau lebih besar dari ukuran partikel

koloid. Yang termasuk TSS adalah lumpur, tanah liat, logam oksida, sulfida, ganggang, bakteri dan jamur.

7. **Oksigen terlarut (*dissolved oxygen*, disingkat **DO**)** atau sering juga disebut dengan kebutuhan oksigen (*Oxygen demand*) merupakan salah satu parameter penting dalam analisis kualitas air.
8. **Nitrat adalah** ion poliatomik dengan rumus molekul NO_3^- dan massa molekul 62,0049 g/mol

BAB II

TINJAUAN PUSTAKA

Dalam rangka melakukan penelitian ini, maka dilakukan tinjauan pustaka terkait dengan sistem yang telah dikembangkan berbasis Sistem Kontrol berbasis *IoT*, protokol komunikasi yang digunakan, algoritma dalam menentukan kualitas air sungai dan memprediksi kualitas air sungai yaitu menggunakan *Naïve Bayes Algorithm* dan *Simple Exponential Smoothing*.

A. Tinjauan Umum Tentang Air

Air merupakan sumber daya alam yang sangat penting dalam kehidupan manusia dan digunakan masyarakat untuk berbagai kegiatan sehari-hari, termasuk kegiatan pertanian, perikanan, peternakan, industri, pertambangan, rekreasi, olahraga dan sebagainya. Dewasa ini, masalah utama sumber daya air meliputi kuantitas air yang sudah tidak mampu memenuhi kebutuhan manusia yang terus meningkat dan kualitas air untuk keperluan domestik terus menurun khususnya untuk air minum. Sebagai sumber air minum masyarakat, air harus memenuhi beberapa aspek yang meliputi kuantitas, kualitas dan kontinuitas (WHO, 2004). Menurut Peraturan Pemerintah No. 82 tahun 2001 tentang Pengelolaan Kualitas Air dan Pengendalian Pencemaran Air, yang disebut sumber air adalah wadah air yang terdapat di atas dan dibawah permukaan tanah. Sumber air dikelompokkan menjadi:

1. Air hujan (termasuk es dan salju)
2. Air permukaan (sungai, rawa, danau, situ, waduk, dan muara)
3. Air tanah (akuifer)
4. Mata air

Berdasarkan PP No. 82 tersebut klasifikasi dan kriteria mutu air terbagi menjadi 4 kelas, sebagai berikut:

1. **Kelas satu:** air yang peruntukannya dapat digunakan untuk air baku air minum, dan atau peruntukan lain yang mempersyaratkan mutu air yang sama dengan kegunaan tersebut.
2. **Kelas dua:** air yang peruntukannya dapat digunakan untuk prasaran/sarana rekreasi air, pembudidayaan ikan air tawar, peternakan, air untuk mengairi pertanaman, dan atau peruntukan lain dipersyaratkan mutu air yang sama dengan kegunaan tersebut
3. **Kelas tiga:** air yang peruntukannya dapat digunakan untuk pembudidayaan ikan air tawar, peternakan, air untuk mengairi pertanaman, dan atau peruntukan lain dipersyaratkan mutu air yang sama dengan kegunaan tersebut.
4. **Kelas empat:** air yang peruntukannya dapat digunakan untuk mengairi pertanaman dan peruntukan lain yang mempersyaratkan mutu air yang sama dengan kegunaan tersebut.

Beberapa parameter yang kami gunakan dalam proses penentuan mutu air seperti yang ditunjukkan pada tabel 1 di bawah ini:

Tabel 1. Kriteria Mutu Air Berdasarkan Kelas (PERATURAN PEMERINTAH REPUBLIK INDONESIA, *n.d.*)

Parameter	Satuan	Kelas			
		I	II	III	IV
TSS	Mg/l	50	50	400	400
pH	Ph	6-9	6-9	0-6	5-9
DO	Mg/l	6	4	3	0
Nitrate	Mg/l	10	10	20	20

B. Sungai Mahakam

Di Provinsi Kalimantan Timur sendiri Sungai Mahakam yang merupakan sungai dengan panjang \pm 920 Km dan melewati 5 (lima) kabupaten dan 1 (satu) kota yaitu : Kabupaten Malinau (KALTARA), Mahakam Hulu, Kutai Barat, Kutai Kartanegara, Kutai Timur dan Kota Samarinda dan terdapat 422 aliran sungai (ordo 2 sampai dengan ordo 5) merupakan urat nadi kehidupan masyarakat sekitar, karena banyak masyarakat yang masih tergantung dengan sungai.

Fungsi Sungai Mahakam bagi kehidupan adalah untuk memenuhi kebutuhan air sehari-hari, sarana transportasi, sumber air bagi pertanian, sumber bahan baku air minum, tempat pembudidayaan dan pencarian ikan air tawar, tempat menyuci dan mandi. Selain menunjang kebutuhan hidup manusia, sungai juga digunakan sebagai tempat akhir pembuangan limbah

dari aktifitas rumah tangga maupun aktifitas industri, pembukaan lahan, perkebunan, kehutanan dan pertanian.

C. Proses Pengolahan dan Pendistribusian Air

Di Indonesia Perusahaan Daerah Air Minum (PDAM) merupakan perusahaan resmi milik daerah yang bertugas mendistribusikan air bersih ke masyarakat. Dalam proses pengolahan air sebelum didistribusikan PDAM memiliki fasilitas penunjang antara lain:

1. **Intake** adalah bangunan yang digunakan untuk memperoleh air dari sungai (atau mata air terbesar suatu daerah) sesuai dengan jumlah yang diperlukan.
2. **Menara air baku** adalah sebuah bangunan yang mengontrol air baku supaya tetap konsisten, sehingga proses proses selanjutnya bisa terlaksana maksimal.
3. **Clarifier** menjadi tempat proses bernama koagulasi. Di mesin clarifier, air akan melalui proses pengendapan sehingga kotoran didalamnya akan terpisah dengan air.
4. **Rapid Mixing** bangunan pengaduk cepat ini merupakan tempat pencampuran antara koagulan dan air baku.
5. **Pulsator** tempat ini akan memproses pengadukan lambat, dimana menaikkan dan menurunkan air, jadi flok yang ada bisa bercampur. Lumpur yang mengendap akan dibuang otomatis oleh mesin setiap

20 menit sekali. Proses ini diharapkan dapat mengubah kekeruhan air menjadi 1 *FTU* sehingga siap untuk di filter.

6. **Filtrator atau bangunan filtrasi** merupakan tempat penyaringan kotoran yang tidak mengendap pada 2 proses sebelumnya. Selain kotoran, proses ini juga akan menyaring berbagai bakteri dan mikroorganisme yang larut dalam air. Didalam bangunan ini ada pasir silica hitam yang tebal serta batuan batuan diantaranya kerikil yang akan menyaring air.
7. **Reservoir** Bangunan satu ini adalah bangunan untuk menampung air bersih yang telah diolah dan sudah siap didistribusikan kemasyarakat yang berlangganan.

Adapun tahapan pengolahan air oleh PDAM adalah sebagai berikut:

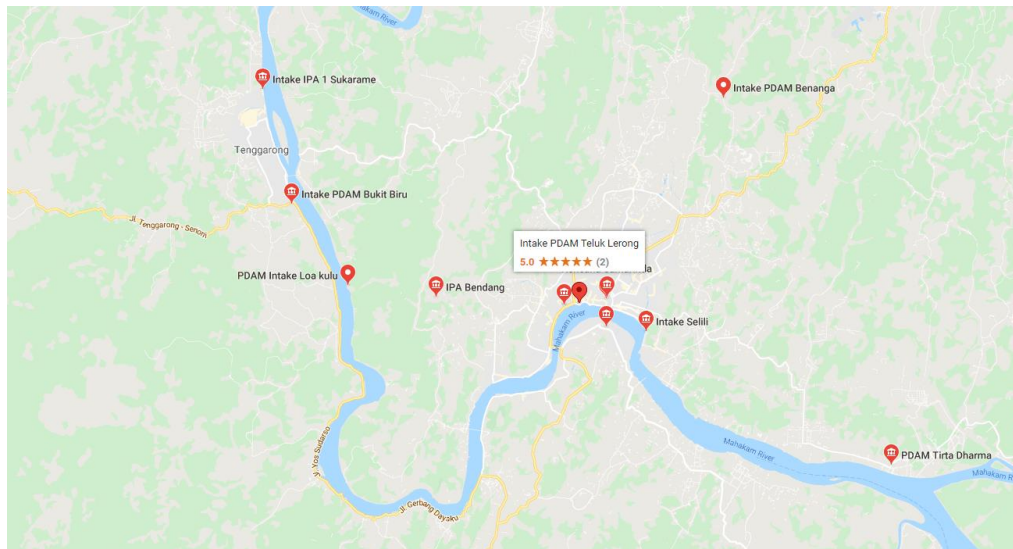
1. **Penyaringan dan Pengendapan** adalah proses awal ketika air sungai datang ke PDAM. Air akan disaring dan diendapkan dengan tujuan memisahkan air baku dari kotoran yang menyertainya seperti dedaunan, pasir, lumpur, batuan kecil hingga sampah.
2. **Koagulasi** merupakan tahap pencampuran bahan kimia $Al_2(SO_4)_3$ atau yang biasa disebut dengan tawas kedalam air yang telah disaring agar kotoran yang berbentuk bakteri, jamur, lumpur halus dan partikel kecil yang tidak dapat disaring dapat mengendap sehingga nantinya air menjadi bersih dan tidak berbau.
3. **Flokulasi** merupakan tahap pembentukan flok setelah koagulasi. Setelah air baku dan koagulan (tawas) dicampur, dilakukan flokulasi

atau pengadukan cepat dan pengadukan lambat agar zat kimia benar benar tercampur rata dalam air.

4. **Sedimentasi**, Setelah melewati tahapan tahapan tersebut, air harus didiamkan lagi selama kurang lebih 15 menit hingga gumpalan kotor dalam air sudah mengendap didasar wadah. Ketika melewati proses ini, keluaran air dari wadah ini tidak akan keruh dan tidak lagi berbau.
5. **Filtrasi**, Setelah mengalami beberapa tahap pengendapan diawal dan di tahap sedimentasi, air ternyata masih mengandung kotoran. Kotoran yang biasanya masih terdapat di air adalah kotoran yang bermassa ringan dan kecil seperti serpihan kayu kecil dan dedaunan kecil yang kering. Penyaringan dilakukan dengan cara mengalirkan air secara vertikal kedalam bak bak penyaring yang berlapis sehingga saat keluar, air akan benar benar bersih.
6. **Disinfeksi**, Setelah melewati berbagai tahapan diatas dan telah siap didistribusikan, air tentunya harus memenuhi beberapa syarat yaitu bersi, tidak berbau, dan sehat. Sehat dalam hal ini adalah erhindar dari berbagai bakteri dan jamur penyebab penyakit. Oleh karena itu, pada tahap terakhir, air akan diberi disinfektan berupa gas khlor sehingga bebas dari bakteri.

Dalam proses pengambilan bahan baku air yang akan diolah PDAM Tirta Kencana Samarinda memiliki beberapa Intake yang tersebar di beberapa titik sepanjang Sungai Mahakam seperti yang ditunjukkan pada gambar 1. Adapun detail posisi masing – masing Intake seperti yang

ditunjukkan pada tabel 1. Dalam penelitian ini hanya melakukan pemrosesan pada Intake Teluk Lerong Samarinda seperti yang ditunjukkan pada gambar 2 dan gambar 3.



Gambar 1. Posisi Intake PDAM Tirta Kencana Samarinda



Gambar 2. Intake Teluk Lerong Samarinda



Gambar 3. Pipa Saluran ke Pengolahan.



Gambar 4. Posisi intake pada Googlemaps.

D. Sistem Kontrol berbasis *IoT (Internet of Things)*

Sistem kontrol adalah proses melakukan pengendalian terhadap sesuatu parameter agar berada pada batas – batas yang diinginkan. Di era revolusi industri 4.0 otomasi sangat membantu dalam hal kelancaran operasional, biaya produksi, keamanan, mutu produk dan lainnya. Sistem kontrol otomatis adalah suatu sistem yang bekerja atau berfungsi mengendalikan proses tanpa adanya campur tangan manusia (otomatis). Ada dua sistem kontrol pada sistem kendali/kontrol otomatis yaitu:

1. *Open Loop* (Loop Terbuka) adalah suatu sistem kontrol yang hasil keluarannya tidak berpengaruh terhadap aksi pengendalian.
2. *Close Loop* (Loop Tertutup) adalah sistem kontrol yang hasil keluarannya berpengaruh terhadap aksi pengendalian yang dilakukan.

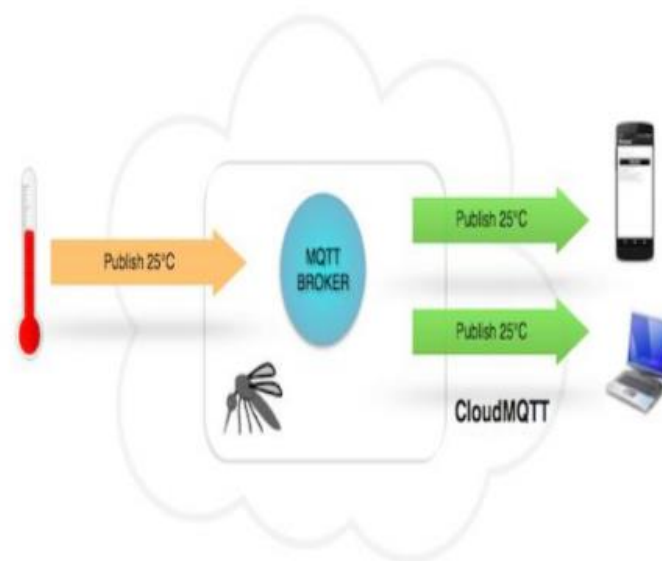
Dalam penelitian ini diterapkan jenis aksi pengontrol *Two Position Controller* atau *on – off* untuk mengontrol perangkat In-Take (pompa). Karakteristik dari jenis aksi pengontrol ini adalah hanya memiliki dua jenis keluaran untuk pengendaliannya. Dalam perkembangan sistem kontrol otomatis adalah salah satunya memanfaatkan sarana internet untuk mengubah parameter – parameter pengontrol yang sering disebut juga *Internet of Things (IoT)*. Skema memberikan ruang perkembangan sistem kendali otomatis jarak jauh dimana saja dan kapan saja sehingga memudahkan dalam proses pengendalian sistem.

IoT (Internet of Things) merupakan solusi dalam memperluas manfaat dari konektivitas internet yang terhubung secara terus menerus. (Depari et al., 2019) mengembangkan sistem monitor instrument alat deteksi penyakit *Neurodegenerative* dengan memanfaatkan konsep *IoT* arsitektur yang diusulkan adalah menghubungkan instrument diagnose ke *cloud* sehingga memungkinkan operator instrument berinteraksi langsung menggunakan sarana *internet*, arsitektur ini memanfaatkan protokol komunikasi pertukaran data yaitu *MQTT* dan *AMQP (Advanced Message Queuing Protocol)*. (Wang, Zhang, Zhang, Zhang, & Li, 2013) mengusulkan sistem control cerdas berbasis pada teknologi internet untuk mengontrol peralatan pada rumah untuk mengatur sensor nirkabel dan jaringan actuator yang menggunakan frekuensi radio 433Mhz. Server, aplikasi, komputer klien dan tablet atau ponsel pintar berkomunikasi dengan pusat control melalui jaringan nirkabel. Sistem ini mencakup fungsi monitor dan manajemen perangkat, keamanan rumah, statistik dan Analisa penggunaan energi. (Pavithra & Balakrishnan, 2015) membangun sistem control peralatan rumah berbasis web yang terhubung ke internet sehingga dapat melakukan optimalisasi daya yang efisien.

E. *MQTT (Message Queuing Telemetry Transport)*

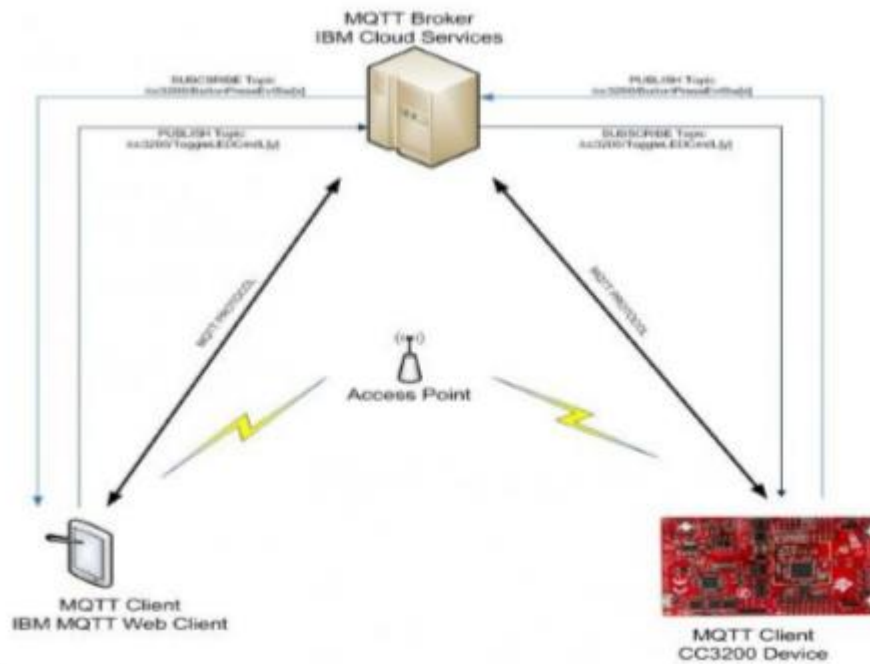
MQTT adalah protokol konektivitas machine-to-machine (M2M)/ *Internet of Things (IoT)* yang berbasis *open source* (Eclipse) dengan standar terbuka (OASIS) yang dirancang untuk perangkat terbatas dan bandwidth rendah, dengan latency tinggi atau berjalan pada jaringan yang tidak dapat

diandalkan (Dizdarevic, Carpio, Jukan, & Masip-Bruin, 2019) (Sarafov, 2018) (Vyas & Rudani, 2018) *MQTT* sangat ideal untuk perangkat yang terhubung dan aplikasi mobile di era *M2M/IoT* dimana *bandwith* dan daya baterai menjadi pertimbangan utama.



Gambar 5. *MQTT* Skema (Jerald, Anand, & Deepika, 2019) .

Protokol *MQTT* menggunakan prinsip publish *subscribe*. komponen (biasanya sensor) yang menghasilkan info tertentu dan menerbitkan info tersebut disebut *publisher*. Client yang tertarik untuk mendapatkan info tertentu mendaftarkan diri minat dari info tertentu, proses ini disebut *subscribe*, client yang berminat disebut *subscriber*. Selain *publisher* dan *subscriber* ada juga broker yang menjamin *subscriber* mendapatkan info yang diinginkan dari *publisher*.



Gambar 6. Skema Broker dalam *MQTT* (Jerald et al., 2019)

Gambar 6 menunjukkan arsitektur jaringan WSN dengan broker menjadi middleware. Broker terletak pada traditional network (Internet/LAN/WLAN). Gateway dibutuhkan untuk menyediakan akses dengan broker. Untuk keamanan pada *MQTT* bisa menggunakan proxy pada *MQTT* tersebut atau menggunakan *HTTP proxy*. Bedanya adalah pada *MQTT proxy* memiliki pendekatan latensi yang lebih rendah dan lebih jelas jika ukuran datanya meningkat. Untuk menjamin pesan terkirim ke client, *MQTT* mendefinisikan 3 level *Quality of Service (QoS)*.

1. 0: Broker/client akan mengirimkan pesan sekali, tanpa konfirmasi.
2. 1: Broker/client akan mengirimkan pesan minimal sekali, diperlukan konfirmasi.

3. 2: Broker/client akan mengirimkan pesan tepat sekali dengan menggunakan four step handshake

Pesan bisa dikirimkan pada level QoS manapun dan client dapat subscribe level QoS manapun juga. Artinya client memilih maksimum QoS yang akan diterima. Contohnya jika info dikirimkan pada QoS 2 dan client menerima dengan QoS 0, pesan akan dikirimkan dengan QoS 0. jika client kedua *subscribe* info yang sama tetapi dengan QoS2, maka pesan yang sama akan dikirimkan namun dengan QoS2.

F. Metode Naive Bayes

Naive Bayes Classifier (NBC) merupakan teknik prediksi berbasis *probabilistic* sederhana yang berdasar pada penerapan teorema Bayes (atau aturan Bayes) dengan asumsi independensi (ketidaktergantungan) yang kuat(naif).

Algoritma *Naïve Bayes* merupakan salah satu algoritma yang terdapat pada teknik klasifikasi. Dalam algoritma ini semua atribut yang terdapat pada dataset dianggap tidak mempunyai hubungan, jadi antara atribut dianggap tidak ada keterhubungannya dengan atribut yang lain. Algoritma ini menghitung probabilitas masing-masing atribut untuk dihitung kembali rata-rata probabilitas atribut-atribut tersebut. Metode *Naïve Bayes* cukup mudah di implementasikan, dan hasil dari metode ini pun baik.

Dari pendapat diatas dapat kita simpulkan bahwa Naive bayes adalah teknik prediksi berbasis probabilistic sederhana dan merupakan salah satu

algoritma yang terdapat pada teknik klasifikasi. Rumus dari Naive Bayes adalah sebagai berikut :

$$P(Y|X) = \frac{P(Y)P(X_1|Y)P(X_2|Y)\dots P(X_n|Y)}{P(X)} \dots \dots \dots (1)$$

Dimana,

$P(Y|X)$ = Probabilitas data dengan vector X pada kelas Y

$P(Y)$ = Probabilitas awal kelas Y

$P(X_i | Y)$ = Probabilitas X_i berdasarkan kondisi hipotesis H

$P(X)$ = Probabilitas X

G. *Simple Exponential Smoothing*

Simple Exponential Smoothing atau metode Exponential Smoothing sederhana. Dengan menggunakan metode peramalan moving average dan *Simple Exponential Smoothing* sehingga menghasilkan sistem peramalan yang baik, yang bisa dimanfaatkan dan memudahkan para pengguna.

Simple Exponential Smoothing digunakan untuk peramalan dengan jangka waktu pendek, biasanya hanya satu bulan ke depan. Model ini mengasumsikan bahwa data berfluktuasi di sekitar rata-rata yang stabil (tidak ada trend atau pertumbuhan pattern yang konsisten). Rumus dari model *Simple Exponential Smoothing* adalah sebagai berikut:

$$F_{t+1} = \alpha D_t + (1 - \alpha) F_t \dots \dots \dots (2)$$

Keterangan:

F_{t+1} = Hasil peramalan

α = inisialisasi

D_t = Data sekarang

F_t = Peramalan Sebelumnya

BAB III

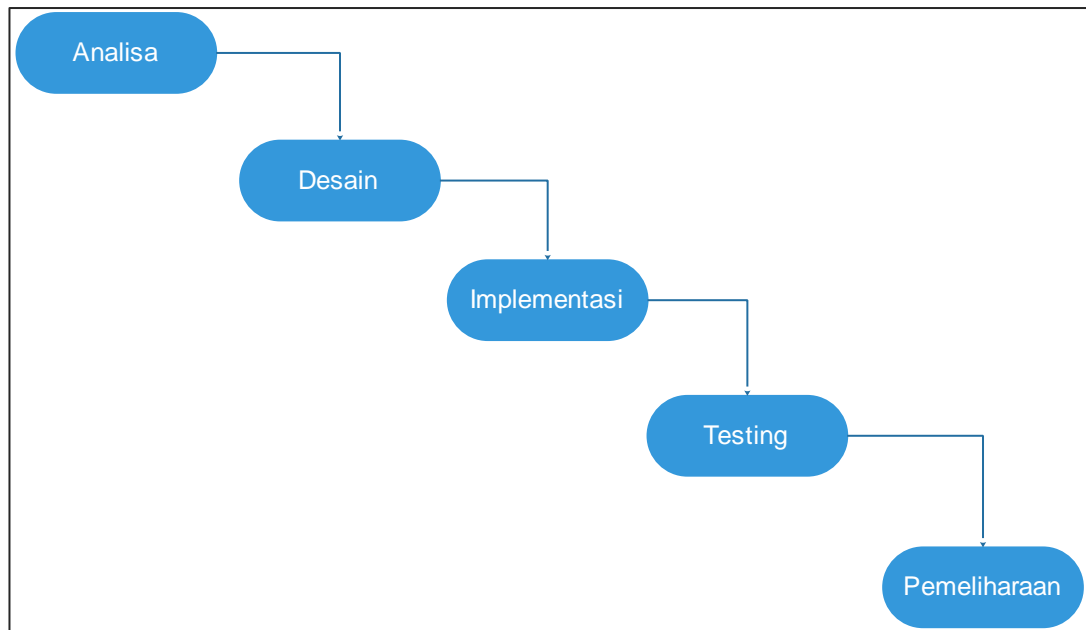
METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian ini penelitian rekayasa (*re-engineering*) adalah penelitian yang menerapkan ilmu pengetahuan menjadi suatu rancangan guna mendapatkan kinerja sesuai dengan persyaratan tertentu. Rancangan tersebut merupakan sintesis unsur-unsur rancangan yang dipadukan dengan metode ilmiah menjadi suatu model yang memenuhi spesifikasi tertentu. Penelitian berawal dari menentukan spesifikasi rancangan yang memenuhi spesifikasi yang ditentukan, memilih alternative terbaik, dan membuktikan bahwa rancangan yang dipilih memenuhi persyaratan yang ditentukan secara efisiensi, efektif dan biaya yang murah. Dalam proses rekayasanya penelitian ini menggunakan model *SDLC* (*Software Development Life Cycle*) adalah pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengambangkan sebuah sistem. *SDLC* juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak yang terdiri dari tahapan: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*).

Model *SDLC* yang dipakai dalam penelitian ini adalah model Waterfall. Waterfall Model atau Classic Life Cycle merupakan model yang paling banyak dipakai dalam Software Engineering (SE). Menurut Bassil (2012)

disebut *waterfall* karena tahap demi tahap yang harus dilalui menunggu selesainya tahap sebelumnya dan berjalan berurutan.



Gambar 7. Metode *Waterfall* (Bassil, 2012)

1. **Analisa**, Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Seorang sistem analis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut.
2. **Desain Sistem**, Proses desain akan menerjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada:

struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural.

3. **Implementasi**, merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan pengujian terhadap sistem yang telah dibuat tadi. Tujuan pengujian adalah menemukan kesalahan-kesalahan terhadap system tersebut dan kemudian bisa diperbaiki.
4. **Testing**, Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user
5. **Pemeliharaan**, Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau system operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

B. Waktu dan Lokasi Penelitian

Penelitian ini dilakukan di Dinas Lingkungan Hidup Provinsi Kalimantan Timur yang merupakan bertanggung jawab kepada kualitas lingkungan hidup termasuk dalam hal ini terkait pemantauan air sungai Mahakam. Penelitian ini dilakukan selama 2 bulan, dimulai semenjak disetujuinya proposal penelitian ini.

C. Tahapan penelitian

Penelitian ini merupakan penelitian lanjutan dari berbagai topik *IoT* dan protokol komunikasi *MQTT* serta penerapan proses data mining dengan menggunakan *Naïve Bayes* dan *Simple Exponential Smoothing* sehingga menghasilkan sistem control dan sistem monitoring pada air sungai Mahakam. Tahapan penelitian yang kami usulkan adalah sebagai berikut:

Tahapan ini adalah sebagai berikut:

1. **Studi Literatur**, tahap ini melakukan pencarian informasi terkait pengembangan sistem berbasis *IoT* dengan menggunakan protokol komunikasi *MQTT* serta penerapan algoritma *Naïve Bayes* dan *Simple Exponential Smoothing*, maka perlu dilakukan studi literatur dari berbagai sumber.
2. **Perancangan Sistem**, tahap ini melakukan perancangan sistem baik perangkat keras dan perangkat lunak yang akan mendukung sistem yang diusulkan

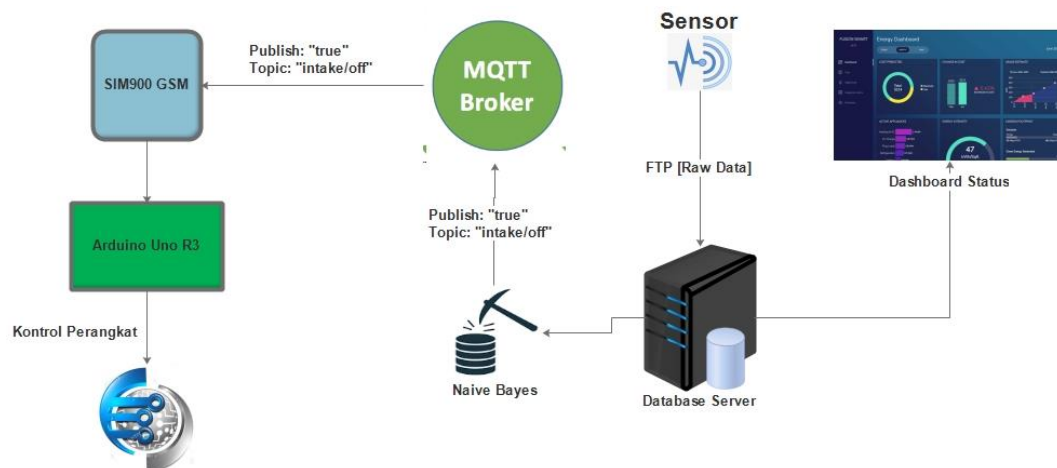
3. **Pembuatan Sistem**, tahap ini melakukan proses pembuatan sistem yang sesuai dengan perancangan sebelumnya.
4. **Uji coba dan Analisis Sistem**, tahap ini melakukan uji coba dari sistem yang dihasilkan dengan menggunakan metode load dan scalability testing yang mencakup *response time*, *throughput* dan *reliabilitas* dari sistem.
5. **Kesimpulan**, tahap ini merupakan tahapan akhir dari penelitian yang dilakukan, yaitu menyimpulkan hal-hal yang dianggap pokok pada penelitian ini

D. Perancangan Sistem

Rancangan sistem yang kami usulkan dalam penelitian ini adalah seperti yang ditunjukkan pada gambar 8. Dengan penjelasan sebagai berikut:

1. Sensor membaca parameter – parameter pengukuran air sungai kemudian mengirim ke *Database server* menggunakan FTP.
2. Hasil pengukuran dari sensor secara *periodik* ditampilkan Dashboard berbasis web
3. Data – data yang diperoleh dari sensor yang tersimpan pada *database server* akan diolah menggunakan *Naïve Bayes* untuk menentukan kondisi air sungai
4. Hasil dari proses tersebut akan dikirim ke *MQTT* server yang selanjutnya akan diteruskan oleh *MQTT* Broker ke semua subscribers.

5. Mikrokontroler yang terhubung ke SIM900 GSM untuk media koneksi ke internet akan melakukan control ke perangkat yang terhubung dan akan menjalankan perintah sesuai dengan command yang diterima dari *MQTT publisher*.



Gambar 8. Skema Sistem Kontrol dan Monitoring

E. Kebutuhan Perangkat Keras

Dalam pengembangan sistem ini diperlukan perangkat – perangkat keras seperti pada tabel di bawah ini

1. Server yang dibutuhkan adalah mengacu pada kebutuhan minimal dari sistem operasi CentOS 8.x.x yaitu
 - a. *Clock Processor* : 2Ghz
 - b. *RAM* : 2GB
 - c. *Harddisk* : 20GB
2. Mikrokontoller yang digunakan adalah ATmega yang telah tertanam pada set Arduino Uno R3 dengan spesifikasi sebagai berikut

- a. Mikrokontroler ATmega328 Tegangan pengoperasian 5V Tegangan input yang disarankan 7-12V Batas tegangan input 6-20V
 - b. Jumlah pin I/O digital
 - c. 14 (6 di antaranya menyediakan keluaran PWM)
 - d. Jumlah pin input analog 6 Arus DC tiap pin I/O 40 mA Arus DC untuk pin 3.3V 50 mA
 - e. Memori Flash
 - f. 32 KB (ATmega328), sekitar 0.5 KB digunakan oleh bootloader
 - g. SRAM 2 KB (ATmega328) EEPROM 1 KB (ATmega328) Clock Speed 16 Hz
3. SIM900A adalah SIM900A adalah GSM/GPRS module yang dapat digunakan untuk mengirim sms, calling, transfer data melalui GPRS & fungsi DTMF. Module GSM – GPRS dengan chip Sim900A dari simcomm yg sudah dilengkapi dengan antenna.
- a. Power supply: 5V
 - b. Tx/Rx level: support 3,3V / 5V / RS232, kompatibel dengan Raspberry pi, Arduino, uC, PC.
 - c. Ukuran: 4,9cm x 4,7cm
 - d. Control: via AT commands
4. *2-Channel Relay Module* digunakan untuk sebagai saklar on/off listrik AC maskimal 250V yang digunakan untuk mengontrol pompa. Dengan

spesifikasi sebagai berikut input 3-5VDC dapat mengontrol arus listrik AC maksimal 250V dan 10A.

5. Step Down Module Modul konverter DC ke DC (DC-DC Converter) ini menggunakan IC LM2596S yang merupakan Integrated Circuit (IC) untuk mengubah tingkatan tegangan (voltage level) arus searah / Direct Current (DC) menjadi lebih rendah dibanding tegangan masukannya. Tegangan masukan (input voltage) dapat dialiri tegangan berapa pun antara 3 Volt hingga 40 Volt DC, yang akan diubah menjadi tegangan yang lebih rendah di antara 1,5 Volt hingga 35 Volt DC. Besar arus berkelanjutan (continuous current) yang dapat ditangani modul elektronika ini sebesar 1,5A dengan arus puncak / momentary peak current 3A.
6. Power Supply 12V 5A merupakan komponen digunakan sebagai catu daya dari komponen – komponen terkait.
7. Sensor yang digunakan adalah seperangkat sensor dari s-can yang telah terstandar dan teruji sehingga hamper dapat dipastikan memberikan hasil yang valid untuk masing – masing parameter yang digunakan dalam penelitian ini, dengan spesifikasi adalah sebagai berikut seperti yang ditunjukkan pada tabel 2.

Tabel 2. Spesifikasi Sensor

Digital Input	:	8 port
Digital Output	:	4 port
Analog Input	:	4 chanel 22-bit ADC dengan differential input
Analog Input Parameter	:	referensi tegangan 2048 – 5000 mV
Analog Input Range	:	0 – 20 mA atau 0 – 5 volt tanpa pembagi tegangan
Analog Output	:	2 chanel 12 bit DAC, 0 – 5 volt atau 0 – 10 volt
Flash memory	:	2 Mbyte
Serial Data Port	:	1 port RS-232 dan 1 port RS-485 atau 3 port RS-232
Catu Tegangan	:	8 – 30 Volt
Konsumsi Arus	:	20 – 70 mA
Suhu Operasional	:	10° – 60° C
Display Data	:	LCD 2 x 16 Character, LED Indicator
Interval Time	:	periodik dan EWS
Sampling Period	:	10, 15, 20, 30, 60 detik
Sampling Method	:	Komunikasi berbasis digital sensor serial data protokol
Data Processing	:	Konversi dari format ASCII ke floating Point
Metode Perekaman Data Periodik : Sistem Alarm (Event Based)	:	Data periodik sesaat tiap 3 – 60 menit Data rata – rata per jam <ul style="list-style-type: none"> ▪ Berdasarkan input digital dan analog, dapat digunakan untuk system pengamanan alat dan power failure monitoring ▪ Berdasarkan input data serial (data sesaat) vs setting threshold value (baku mutu), dapat digunakan untuk deteksi dini pencemaran air (Early Warning System/EWS) ▪ Data alarm dan Time Stamp direkam di flash memori local
Alarm Action	:	Digital Output, SMS
Media Komunikasi	::	GSM via Modem atau internet (Port 80)
Metode Komunikasi Data	:	SMS dua arah atau menggunakan API (metode POST dan Get) dapat lintas platform

Jenis Modem	:	Serial GSM/GPRS
Metode Data Transfer	:	SMS Otomatis, Standby Mode, Direct cable atau Format JSON
Clock	:	Local RTC
Setting Parameter	:	Via SMS
Server Synchronization	:	Direct Cable
Cek Pulsa Prabayar	:	Otomatis Remote
Jumlah Running Tabel	:	2 (dua) independen running table (time based table data dan event based table data)
Jumlah Sensor Parameter	:	6 – 15 parameter
Format Data Sensor	:	Floating Point
Format Parameter Kimia	:	ID Tgl Jam Temp Cond TDS DO pH Turbidity Depth Amonia Nitrat COD
Jenis Parameter Kimia	:	Suhu, DO, pH, Turbidity, TDS, Depth, Amonia, Nitrat, ORP, COD
Format Parameter Fisik	:	ID Tgl Jam Curah Hujan TMA Debit
Jenis Parameter Fisik	:	Curah Hujan, Tinggi Muka Air, Debit Air
Power Monitoring	:	Internal Monitoring tegangan aki kering
Format File Data	:	8 bit PDU
Time Stamp Data Record	:	Tahun, Bulan, Tanggal, Jam, Menit
Casing Material	:	Plastik / Aluminium
Indoor Casing	:	Indoor Casing IP64 / PVC
Fuse Pengaman	:	2A
Output Display	:	Dapat menampilkan hasil sesaat ke running Text
A.2. Power Management		
a). Batere / Aki kering	:	12 VDC, 12 Ah
b). Solar Cell Panel	:	50 WP

F. Kebutuhan Perangkat Lunak

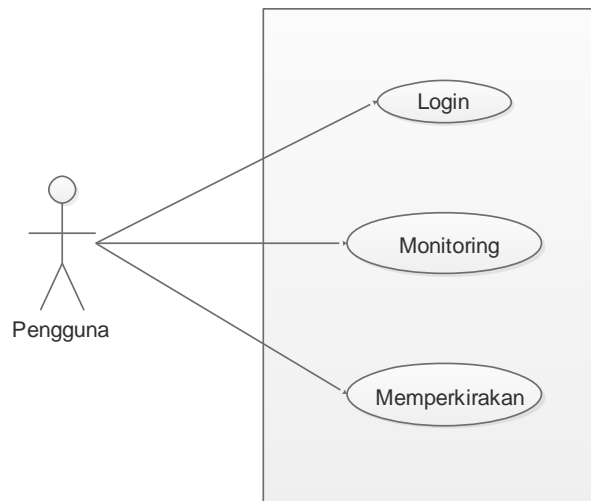
Dalam proses pengembangan sistem dibutuhkan perangkat lunak sebagai berikut:

Tabel 3. Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Operating System: CentOS Linux 7 (Core); CPE OS Name: cpe:/o:centos:centos:7 Kernel: Linux 3.10.0-1062.4.3.el7.x86_64 Architecture: x86-64
2.	<i>Database server</i>	Server version: 5.5.64-MariaDB MariaDB Server
3.	<i>Web Server</i>	Nginx version: nginx/1.16.1; support PHP 7.2.24
4.	<i>MQTT Server</i>	version 1.6.7 MQTT v3.1.1 broker.

G. Use Case

Pada use case sistem terdapat actor pengguna yang melakukan pemantauan dan prediksi parameter – parameter kualitas air dengan terlebih dahulu melakukan login ke sistem.



Gambar 9. Use Case sistem

H. Pengujian Sistem

Pengujian dilakukan dengan membandingkan kualitas air pada data asal dengan kualitas air yang diputuskan dari sistem. Terdapat empat kemungkinan yang terjadi pada saat melakukan penentuan kualitas air dengan menggunakan algoritma *Naïve Bayes* yaitu:

- *True Positive (TP)* adalah kondisi dimana kualitas air baik dan mesin pompa *on*
- *True Negative (TN)* adalah kondisi dimana kualitas air tidak baik dan mesin pompa *off*.
- *False Negative (FN)* adalah kondisi dimana kualitas air tidak baik dan mesin pompa *on*
- *False Positive (FP)* adalah kondisi dimana kua

Penentuan akurasi, presisi, *recall* dan *error rate* ditunjukkan pada persamaan 3,4,5 dan 6.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (3)$$

$$Presisi = \frac{TP}{TP+FP} \dots\dots\dots (4)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (5)$$

$$Error Rate = \frac{FN+FP}{TP+TN+FP+FN} \dots\dots\dots (6)$$

Pengujian sistem yang menggunakan data deret waktu dapat menggunakan *MAPE* (*Mean Absolute Percentage Error*). *MAPE* merupakan ukuran ketepatan relative yang digunakan untuk mengetahui presentase penyimpangan hasil peramalan, dengan persamaan sebagai berikut:

$$MAPE = \frac{1}{n} \sum_{i=1}^n |PEi| \dots\dots\dots (7)$$

BAB IV

HASIL DAN PEMBAHASAN

Tahap ini adalah membangun sistem berdasarkan rancangan yang dibuat. Dalam hal ini menerapkan konsep *IoT* dengan sarana komunikasi menggunakan protokol *MQTT* (*Message Queue Telemetry Transport*). Untuk membantu proses pemantauan diimplementasikan algoritma *Naïve Bayes* untuk menentukan kualitas air, jika dinyatakan tidak baik maka sistem akan mematikan mesin pompa pada In-Take serta menginformasikan hal tersebut kepada pengelola terkait melalui SMS notifikasi dan Email. Sistem ini juga melakukan *forecasting* terhadap nilai – nilai parameter yang menentukan kualitas air menggunakan algoritma *Simple Exponential Smoothing*. Implementasi sistem ini memerlukan perangkat keras dan perangkat lunak yang akan kami paparkan berikutnya. Tahapan akhir pada penelitian ini adalah melakukan pengujian terhadap sistem yang telah dibangun untuk mengetahui atau bahan pertimbangan penelitian – penelitian terkait.

A. Sistem Kontrol berbasis *IoT* protokol *MQTT* dengan Algoritma

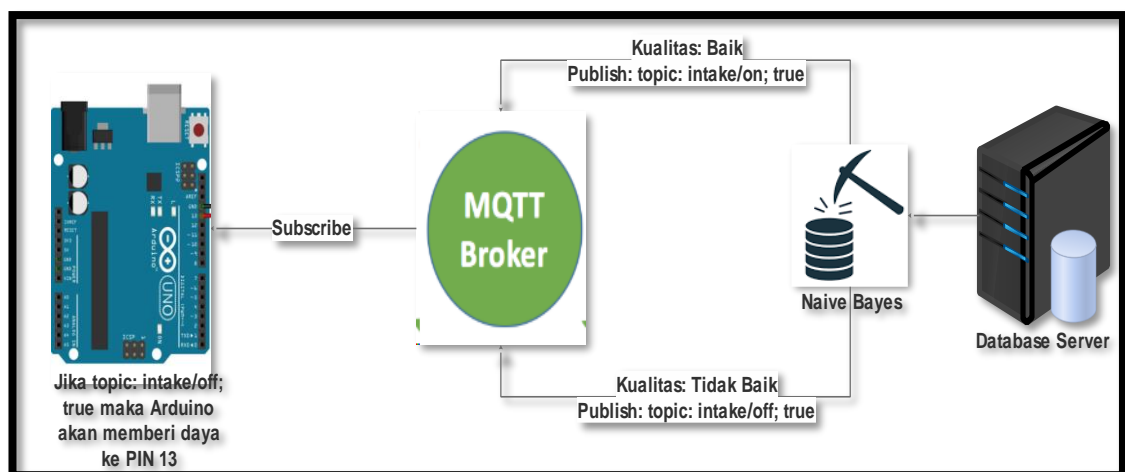
Naïve Bayes

Sistem yang dibangun menggunakan mikrokontroler *Arduino Uno R3* yang terhubung ke *module SIM900A* sebagai sarana agar terhubung ke server *MQTT* pada alamat *MQTT.haitek.id* yang berjalan pada port 1883 menggunakan jaringan 2G GSM. Relay terhubung dengan mikrokontroler dan catu daya sebagai kontrol on/off dari pompa. Catu daya yang

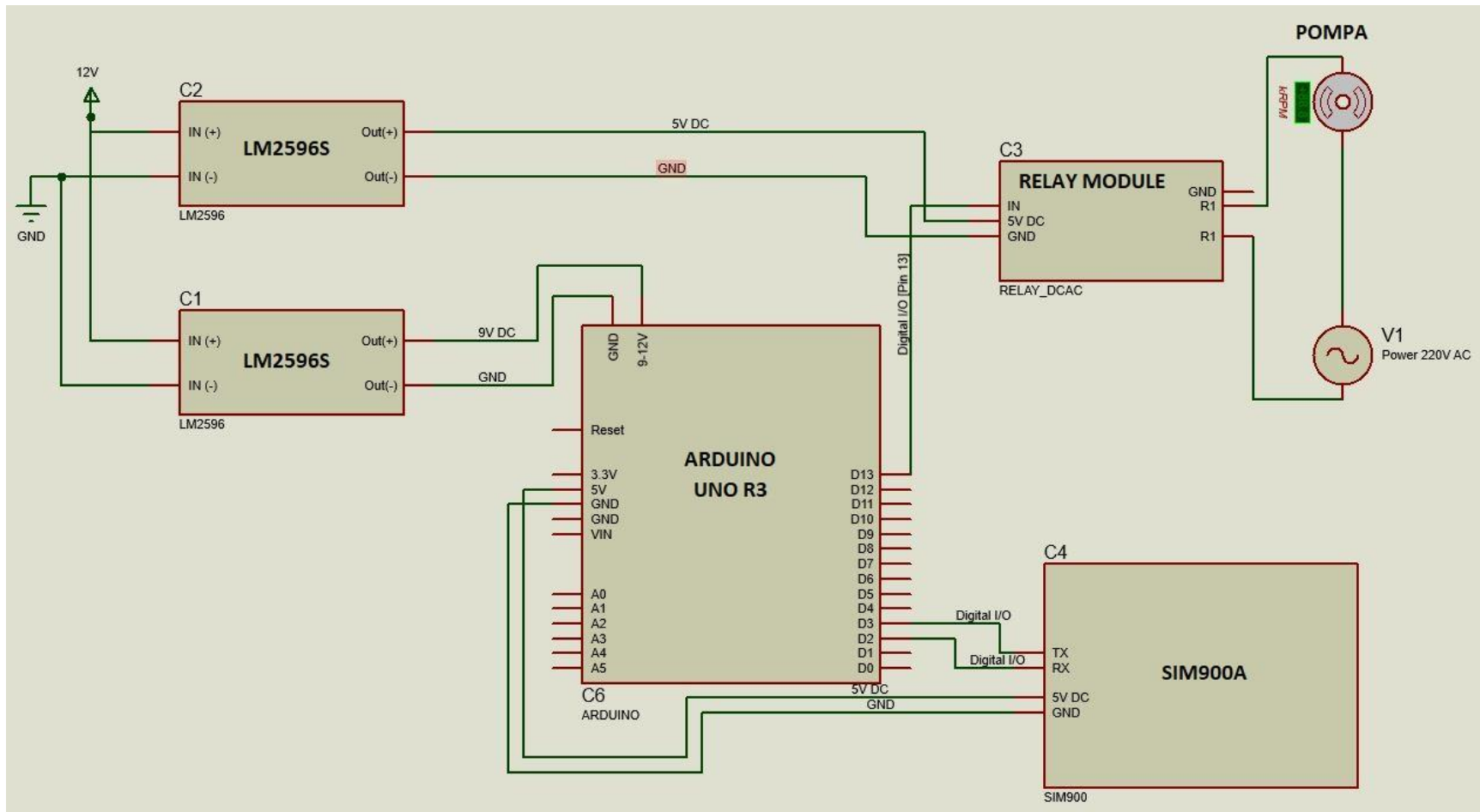
digunakan bersumber dari power supply dengan input 220VAC, output 12VDC yang dihubungkan secara paralel ke 2 Step-Down LM2596S, 1 (satu) diturunkan ke kisaran 9V untuk sumber daya Mikrokontroler dan 1 (satu) diturunkan ke kisaran 5V untuk sumber daya dari relay. Rangkaian dari berbagai perangkat tersebut ditunjukkan pada gambar 11.

B. Skema Komunikasi Mikrokontroler dengan MQTT Server

Skema komunikasi antara mikrokontroler dengan MQTT Server dilakukan dengan cara mikrokontroler sebagai *subscriber*, yang nanti akan menerima *message* yang akan di-*publish* oleh aplikasi server menggunakan algoritma *Naïve Bayes*, jika kualitas air “tidak baik” maka akan mempublish message untuk mematikan mesin pompa. Jaringan yang digunakan adalah menggunakan 2G GSM dengan memanfaatkan module SIM900A yang telah terhubung ke mikrokontroler. Skema komunikasi tersebut ditunjukkan pada gambar 10.

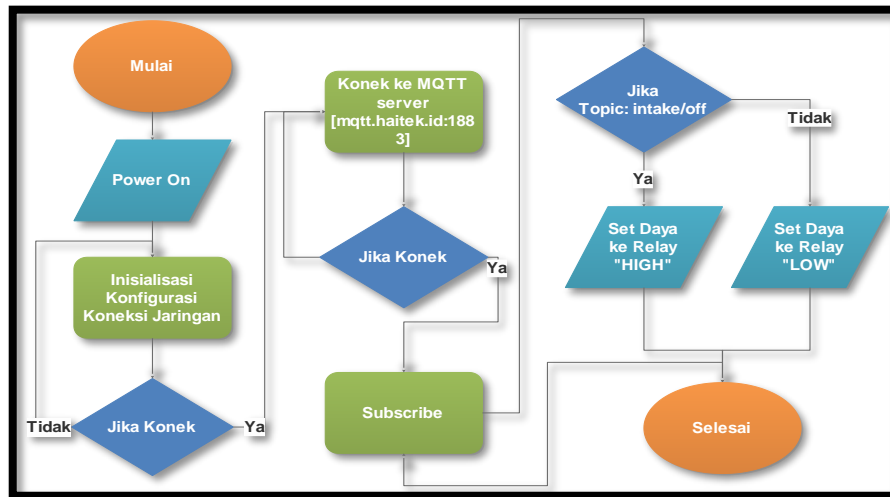


Gambar 10. Skema Komunikasi Mikrokontroler ke MQTT Server



Gambar 11. Rangkaian Sistem Kontrol

Alur program yang tertanam pada mikrokontroler seperti yang ditunjukkan pada flowchart pada gambar 12 dengan detail tahapan adalah sebagai berikut:



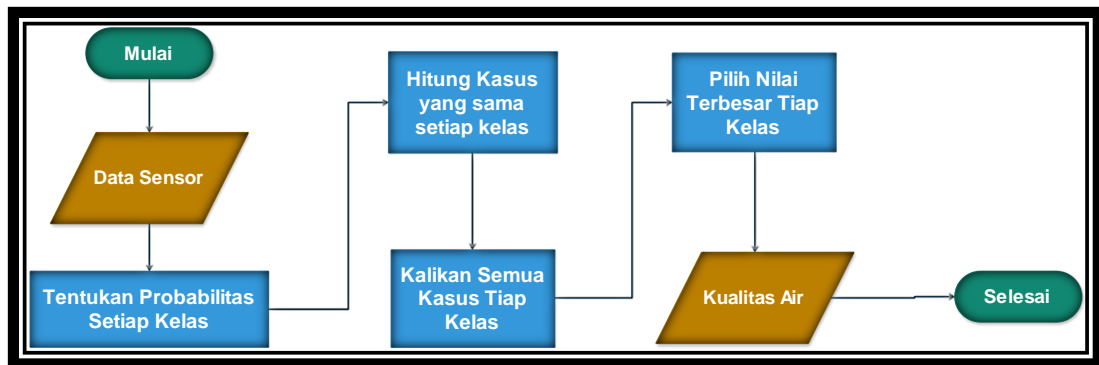
Gambar 12. Flowchart Program Tertanam pada Mikrokontroler

1. **Power on:** memberi daya pada perangkat kontrol dengan 220V AC
2. **Inisialisasi Konfigurasi Koneksi Jaringan:** tahap ini melakukan setting parameter – parameter koneksi jaringan GSM yaitu penentuan APN, jika terhubung maka akan dilanjutkan ke proses berikutnya, jika tidak terhubung maka dilakukan pengulangan sampai jaringan terhubung.
3. **Koneksi ke MQTT Server:** tahapan ini melakukan koneksi ke MQTT Server dengan alamat sebagai berikut MQTT.haitek.id dan port 1883, jika berhasil maka akan dilanjutkan proses subscriber, jika gagal maka akan dilakukan pengulangan sampai terhubung.

4. **Subscribe:** tahapan ini melakukan *subscribe* ke topik untuk selalu membaca message yang sesuai dengan topik secara terus menerus atau dengan kata lain melakukan *streaming*, jika ada message pada topic “intake/off” maka akan memberikan nilai “HIGH” pada pin 13 Arduino sesuai dengan skema pada gambar 5 yang terhubung ke relay hal ini akan melakukan pemutusan daya ke pompa sehingga pompa akan berhenti. Jika ada message pada topic “intake/on” maka akan memberikan nilai “LOW” pada pin 13 hal ini akan meyambung daya ke pompa sehingga pompa aktif kembali.

C. Menentukan Kualitas Air Menggunakan Algoritma *Naïve Bayes*

Proses penentuan Kualitas Air dilakukan dengan menerapkan Algoritma *Naïve Bayes* pada data – data pembacaan sensor yang memiliki siklus per jam yang dikirim melalui protokol FTP kemudian disimpan ke *Database server*. Flowchart dari penerapan Algoritma *Naïve Bayes* ditunjukkan pada gambar 13 terdapat empat proses atau langkah dalam menentukan kualitas air. Adapun langkah – langkah yang dilakukan dalam penerapan *Naïve Bayes* dengan parameter pH = Baik, TSS = Baik, DO = Baik, Nitrate = Baik (sesuai dengan pengukuran sensor update terakhir) adalah sebagai berikut:



Gambar 13. Flowchart Naive Bayes.

1. Menentukan probabilitas dari kualitas air “Baik (B)” dan “Tidak Baik (T)”

Baik : 692 / 1069

Tidak Baik : 377 / 1069

```

echo "Langkah 1: Menghitung Jumlah Status Kualitas Air 'Baik' dan 'Tidak Baik'<br>";
echo "<ul><li>Baik : $jum->jum_b / $jum->jum </li> <li>Tidak Baik : $jum->jum_t / $jum->jum </li></ul>";
  
```

Gambar 14. Code langkah 1 (satu) Implementasi Naive Bayes

2. Menghitung jumlah kasus yang sama dengan kelas yang sama

$P(\text{Kondisi PH} = \text{Baik} \mid Y = \text{Baik})$: 692 / 692

$P(\text{Kondisi PH} = \text{Baik} \mid Y = \text{Tidak Baik})$: 0 / 377

$P(\text{Kondisi TSS} = \text{Baik} \mid Y = \text{Baik})$: 677 / 692

$P(\text{Kondisi TSS} = \text{Baik} \mid Y = \text{Tidak Baik})$: 377 / 377

$P(\text{Kondisi DO} = \text{Baik} \mid Y = \text{Baik})$: 692 / 692

P (Kondisi DO = Baik | Y= Tidak Baik): 377 / 377

P (Kondisi Nitrate = Baik | Y= Baik): 619 / 692

P (Kondisi Nitrate = Baik | Y= Tidak Baik): 1 / 377

```
if($v2->k_ph=='B'){
  echo "<ul><li>P (Kondisi PH = Baik | Y= Baik) : $jum->jum_ph_b_b / $jum->jum_b </li> "
  . "<li>P (Kondisi PH = Baik | Y= Tidak Baik) : $jum->jum_ph_t_b / $jum->jum_t </li></ul>";

  $step2_ph_b = $jum->jum_ph_b_b / $jum->jum_b;
  $step2_ph_t = $jum->jum_ph_t_b / $jum->jum_t;
} else {
  echo "<ul><li>P (Kondisi PH = Tidak Baik | Y= Baik) : $jum->jum_ph_b_t / $jum->jum_b </li> "
  . "<li>P (Kondisi PH = Tidak Baik | Y= Tidak Baik) : $jum->jum_ph_t_t / $jum->jum_t </li></ul>";

  $step2_ph_b = $jum->jum_ph_b_t / $jum->jum_b;
  $step2_ph_t = $jum->jum_ph_t_t / $jum->jum_t;
}
```

Gambar 15. Code Langkah 2 (Dua) Implementasi Naive Bayes

3. Kalikan semua hasil perhitungan parameter Status Kualitas Air 'Baik' dan 'Tidak Baik'

$$P [\text{Baik}] = 1 * 0.97832369942197 * 1 * 0.89450867052023 * (692 / 1069) \\ = 0.56649426561478$$

$$P [\text{Tidak Baik}] = 0 * 1 * 1 * 0.0026525198938992 * (377 / 1069) = 0$$

```
echo "<br>Langkah 3: kalikan semua hasil variable Status Kualitas Air 'Baik' dan 'Tidak Baik'<br>";
$p_baik = $step2_ph_b*$step2_tss_b*$step2_do_b*$step2_nitrate_b*($jum->jum_b / $jum->jum);
$p_tidakbaik = $step2_ph_t*$step2_tss_t*$step2_do_t*$step2_nitrate_t*($jum->jum_t / $jum->jum);
```

Gambar 16. Code Langkah 3 Implementasi Naive Bayes

4. Bandingkan Hasil P[Baik] jika lebih besar P [Tidak Baik] maka Status Kualitas Air 'Baik' jika sebaliknya maka Status Kualitas Air 'Tidak Baik' Status Air **Baik** Karena $0.56649426561478 > 0$ hasil tersebut sistem

akan mem-*publish message* ke *MQTT Server* yaitu topik “intake/on” “true”. Begitu sebaliknya jika setelah dilakukan penentuan kualitas air menghasilkan **Tidak Baik** maka sistem akan mem-*publish message* ke *MQTT Server* yaitu topik “intake/off” “true”. Setelah diperoleh kualitas air maka akan dilakukan pemberitahuan ke *staceholders* berupa pesan singkat melalui SMS dan informasi yang lebih detail melalui *email* hal ini dilakukan dikarenakan SMS hanya dapat menampung 161 karakter, informasi yang dikirimkan seperti yang ditunjukkan pada gambar 18. Code untuk mengontrol pompa ditunjukkan pada gambar 19.

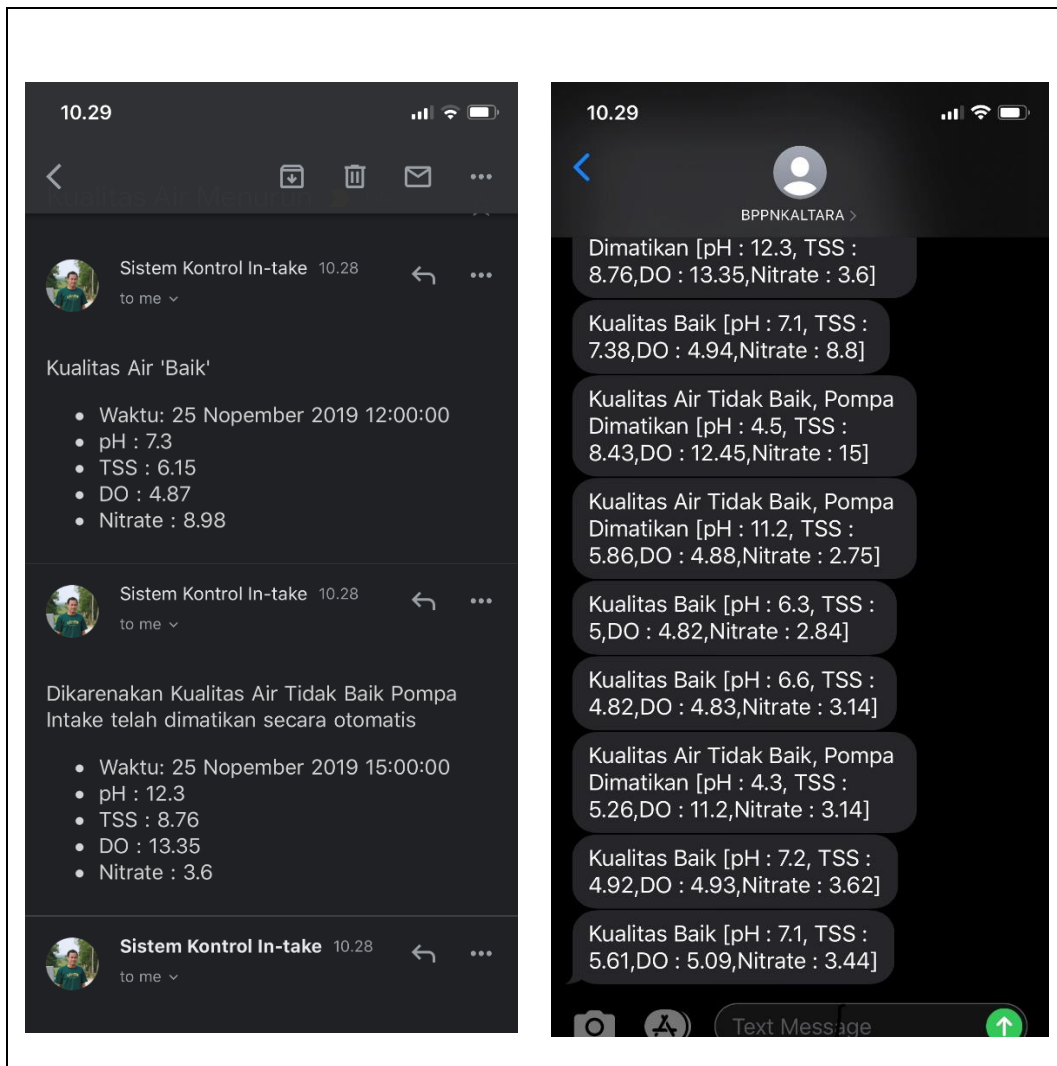
```

if($p_baik > $p_tidakbaik) {
    echo "Status Air <span style='font-weight:bold'>Baik</span> Karena $p_baik > $p_tidakbaik ";
    $status="B";
    shell_exec('mosquitto_pub -h mqtt.haitek.id -t "intake/on" -m "true" -u "mqtt" -P "Test#123"');
    $msgEmail="<p>Kualitas Air 'Baik'</p>"
        . "<ul><li>Waktu: ".DateToIndo($v2->sensorlog)."</li> "
        . "<li>pH : $v2->at5_ph</li>"
        . "<li>TSS : $v2->at6_tss</li>"
        . "<li>DO : $v2->at6_do</li>"
        . "<li>Nitrate : $v2->at5_nitrate</li>"
        . "</ul>";

    $msgSMS = "Kualitas Baik [pH : $v2->at5_ph, TSS : $v2->at6_tss,DO : $v2->at6_do,"
        . "Nitrate : $v2->at5_nitrate]";
    $penerima = $this->model_naivebayes->get_penerima();
    if($penerima!=false){
        foreach ($penerima as $vp) {
            if (filter_var($vp->email, FILTER_VALIDATE_EMAIL)) {
                sendgmail($vp->email, "Kualitas Air Menurun", $msgEmail);
            }
            $this->serviceclient->sentSMS($vp->nohp,$msgSMS);
            $param = array(
                "nohp"=>$vp->nohp,
                "email"=>$vp->email,
                "msgemail"=> addslashes($msgEmail),
                "msghp"=> addslashes($msgSMS),
            );
            // $this->model_master->insert("tb_send",$param);
        }
    }
} else {

```

Gambar 17. Code Langkah 4 (empat) Implementasi *Naive Bayes*.



Gambar 18. Pemberitahuan Kualitas Air Melalui *SMS* dan *Email*

```

void mqttCallback(char* topic, byte* payload, unsigned int len) {
  if (String(topic) == "intake/off") {
    SerialMon.println("Engine [Off]");
    relayStatus = HIGH;
  } else{
    SerialMon.println("Engine [On]");
    relayStatus = LOW;
  }
  digitalWrite(RELAY_PIN, relayStatus);
}

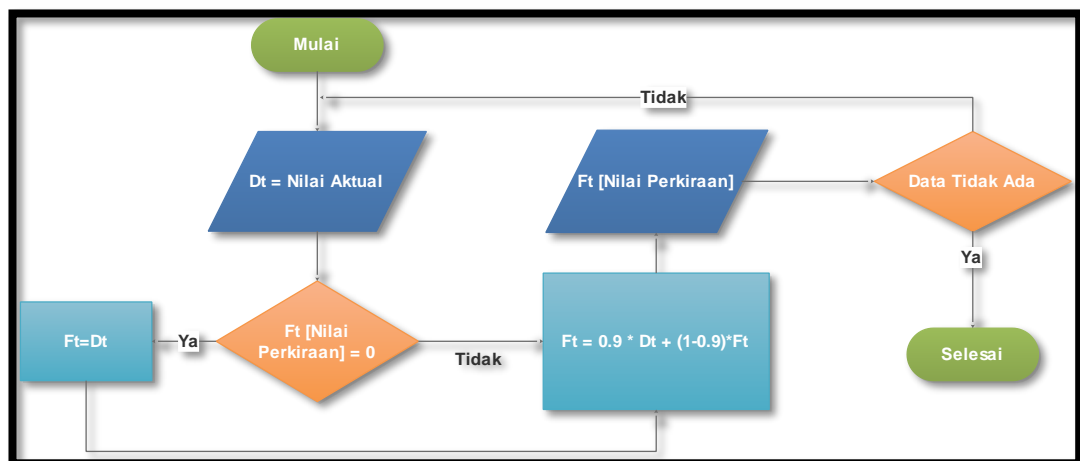
```

Gambar 19. *Code* mengontrol pompa

**D. Forecasting parameter – parameter Kualitas Air menggunakan
*Simple Exponential Smoothing***

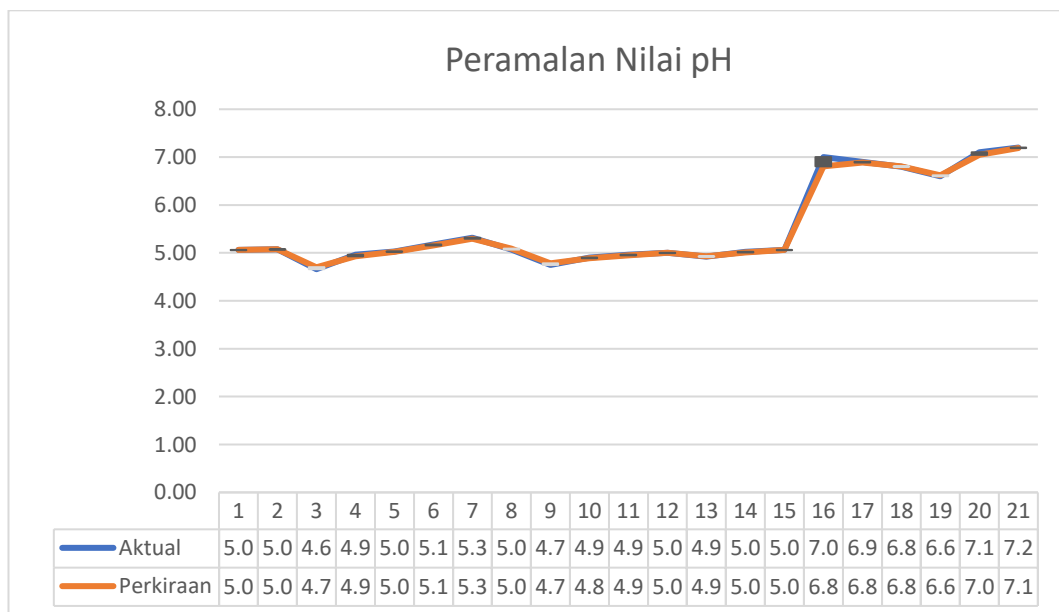
Algoritma adalah prosedur langkah-langkah untuk penghitungan. Algoritma digunakan untuk penghitungan, pemrosesan data, dan penalaran otomatis. Algoritma peramalan yang digunakan dalam sistem ini adalah *Simple Exponential Smoothing*. Metode tersebut adalah prosedur yang secara terus menerus memperbaiki peramalan dengan merata-rata (menghaluskan = smoothing) nilai masa lalu dari suatu data runtut waktu dengan cara menurun (exponential).

Dalam penelitian ini dilakukan peramalan terhadap parameter – parameter kualitas air yaitu nilai pH, TSS, DO, Nitrate sebagai contoh proses peramalan nilai pH untuk besok harinya pada pukul 00:00 adalah sebagai berikut:



Gambar 20. Flowchart Simple Exponential Smoothing.

Dari Flowcart diatas dilakukan perhitungan sebanyak 21 data berurutan seperti yang ditunjukkan pada grafik gambar 15, maka diperkirakan bahwa nilai pH pada tanggal 25 November 2019 pukul 00:00 adalah sebesar 7.10 (Tujuh koma Satu). Skema ini telah kami terapkan pada sistem berbasis web. Algoritma *Simple Exponential Smoothing* diterapkan untuk data – data sebelumnya yang telah tersimpan pada *database server* dengan siklus per jam dan periode *forecasting* per hari. Sistem akan menampilkan data hasil perhitungan berdasarkan tanggal yang dipilih sebanyak 24 jam per parameter kualitas air.



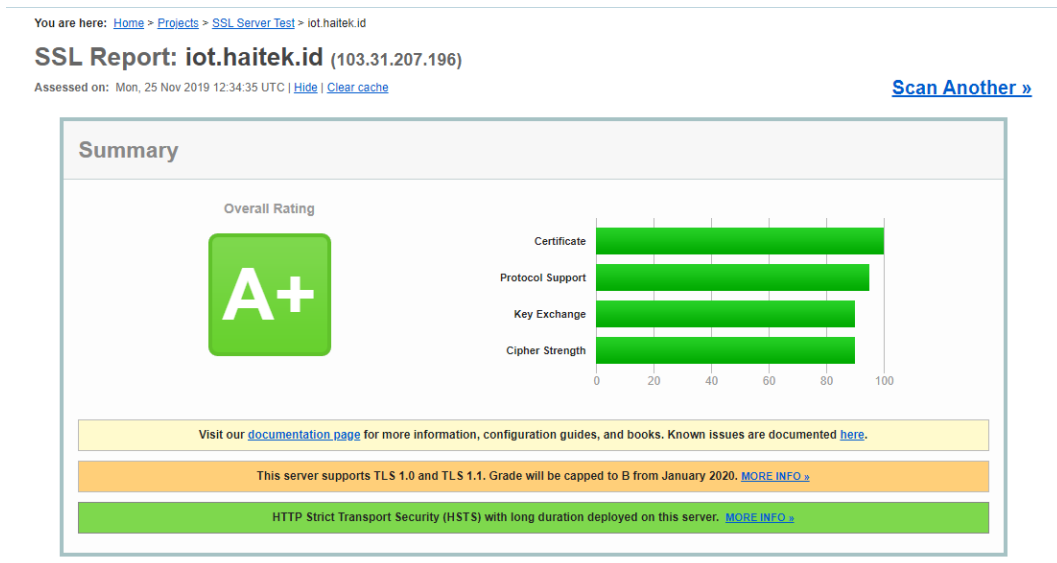
Gambar 21. Grafik Proses Peramalan pH

Hasil perkiraan ini akan menjadi pertimbangan dalam rangka pemantauan bahan baku air yang akan diolah. Jika diperkirakan akan mengalami penurunan kualitas air pada waktu yang diperkirakan maka tindakan yang

diambil bersiaga pada waktu tersebut agar dapat mengalihkan pengambilan bahan baku air melalui sumber lainnya. Hasil tersebut juga dapat menjadi tindakan preventif untuk mematikan pompa sebelum waktu yang diperkirakan kualitas air akan menurun untuk menghindari mengambil air yang kualitasnya tidak memenuhi standar baku mutu. Kemudian jika ternyata hasil perkiraan berbeda dengan kondisi nyata yaitu diperkirakan kualitas air tidak baik dan faktanya kualitas air baik, maka sistem control pompa otomatis akan menyalakan pompa dan kembali mendapatkan kualitas air yang baik.

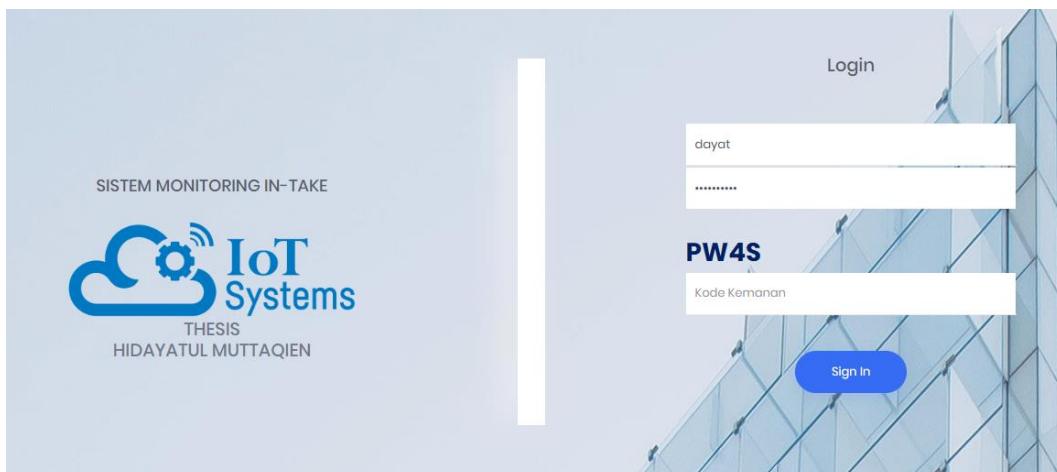
E. Antarmuka Sistem Kontrol dan Pemantauan

Antarmuka Sistem kontrol dan Pemantauan Kualitas Air Sungai dibangun untuk memudahkan pengguna dalam menggunakan sistem. Antarmuka dari sistem ini adalah berbasis web yang mendukung mobile responsive yang dapat menyesuaikan layout dengan layar perangkat. Dalam proses transaksi data pada website telah menggunakan *SSL (Secure Socket Layer)* untuk melakukan enkripsi data agar meningkatkan keamanan sistem antarmuka yang dibangun. Konfigurasi keamanan web mendapat grade "A+" seperti yang ditunjukkan pada gambar 22 di bawah ini.



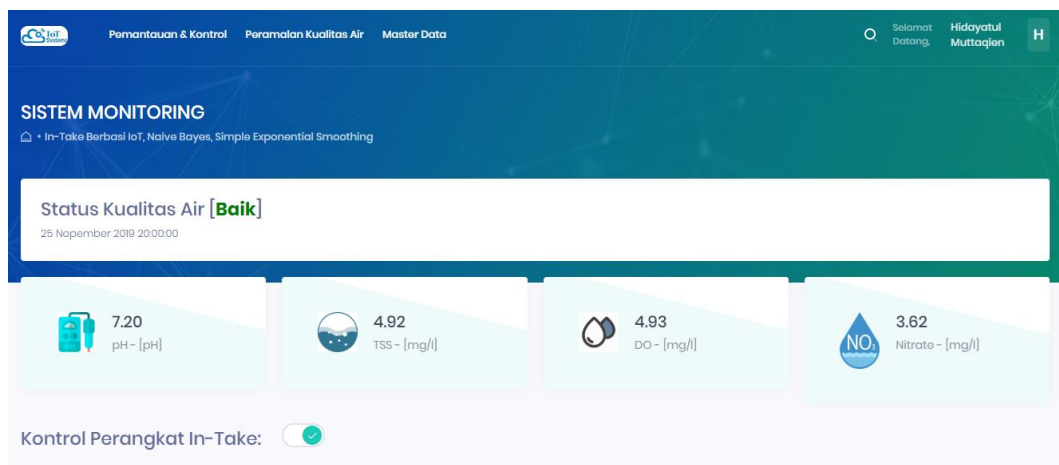
Gambar 22. Grade SSL Konfigurasi

Sebelum dapat menggunakan antarmuka sistem ini maka pengguna diwajibkan *Login* ke sistem dengan menggunakan username dan password yang telah terdaftar halaman login seperti ditunjukkan pada gambar 23 dibawah ini.



Gambar 23. Halaman Login

Jika username, password dan kode keamanan sesuai maka akan tampil halaman kontrol dan pemantauan kualitas air yang menampilkan kualitas air dan parameter – parameter pengukuran sensor seperti PH, TSS, DO, Nitrate berdasarkan nilai pengukuran terakhir. Pengukuran parameter – parameter tersebut di update setiap 1 (satu) jam sekali. Penentuan kualitas air menggunakan algoritma *Naïve Bayes* yang proses dilakukan setiap ada update data terbaru setiap jamnya. Sistem akan secara otomatis mematikan mesin pomp ajika hasil dari proses penentuan kualitas air “tidak baik” pada antarmuka berbasis web ini juga dapat dilakukan mematikan atau menghidupkan pompa pada halaman “Pemantauan & Kontrol” seperti ditunjukkan pada gambar 24 di bawah ini.



Gambar 24. Halaman Pemantauan dan Kontrol

Untuk melakukan perkiraan parameter – parameter kualitas air dapat dilakukan pada menu “Peramalan Kualitas Air” dengan cara memilih tanggal yang diinginkan maka sistem akan menampilkan perkiran nilai dari

parameter – parameter tersebut sebanyak 24 jam yang dibagi per jam seperti yang ditunjukkan pada gambar 25 di bawah ini.

Peramalan Kualitas Air

Pilih Tanggal

26/11/2019 Proses

Tanggal dan Waktu	Perkiraan Parameter Kualitas Air			
	pH	TSS	DO	Nitrate
26 Nopember 2019 [00:00:00]	7.199	8.237	4.783	5.036
26 Nopember 2019 [01:00:00]	7.366	7.647	4.896	6.750
26 Nopember 2019 [02:00:00]	7.147	6.440	4.963	4.133
26 Nopember 2019 [03:00:00]	6.662	5.569	4.976	3.115
26 Nopember 2019 [04:00:00]	6.305	5.930	4.977	2.184
26 Nopember 2019 [05:00:00]	6.324	5.347	4.863	2.175
26 Nopember 2019 [06:00:00]	6.362	4.406	4.682	2.170

Gambar 25. Halaman peramalan parameter kualitas air.

F. Pengujian Sistem

Pengujian dilakukan dengan membandingkan kualitas air yang diputuskan sistem dengan kondisi perangkat pompa, dimana jika kualitas air “Baik” maka pompa menyala dan sebaliknya jika kualitas air “Tidak Baik” maka pompa akan tidak menyala. Pengujian dilakukan sebanyak sepuluh kali seperti yang ditunjukkan pada tabel 5 di bawah ini Terdapat empat

kemungkinan yang terjadi pada saat melakukan penentuan kualitas air dengan menggunakan algoritma *Naïve Bayes*.

Tabel 4. Percobaan Sistem dan Mesin Pompa

Percobaan	Kualitas Air	Pompa	Confusion Matrix
Percobaan 1	Baik	On	TP
Percobaan 2	Tidak Baik	Off	TN
Percobaan 3	Baik	On	TP
Percobaan 4	Tidak Baik	Off	TN
Percobaan 5	Tidak Baik	Off	TN
Percobaan 6	Baik	On	TP
Percobaan 7	Baik	On	TP
Percobaan 8	Tidak Baik	Off	TN
Percobaan 9	Baik	On	TP
Percobaan 10	Tidak Baik	Off	TN

Konsumsi bandwidth dengan menggunakan protokol *MQTT* cukup kecil yaitu 14KB untuk *sending* dan 12KB receiving, hal ini membuat komunikasi antara sistem kontroler dengan server *MQTT* sangat reliable hal ini membuat perangkat menjadi responsive dalam melakukan proses on – off pompa. Dari tabel diatas maka menghasilkan *confusion matrix* seperti disajikan pada tabel di bawah ini.

Tabel 5. Confusion Matrix

<i>Kualitas Air</i>	On	Off
<i>Baik</i>	TP (5)	FN (0)
<i>Tidak Baik</i>	FP (0)	TN (5)

Dari tabel diatas dapat dihitung akurasi, presisi, *recall* dan *error rate* dengan menggunakan persamaan 3,4,5 dan 6 sebagai berikut:

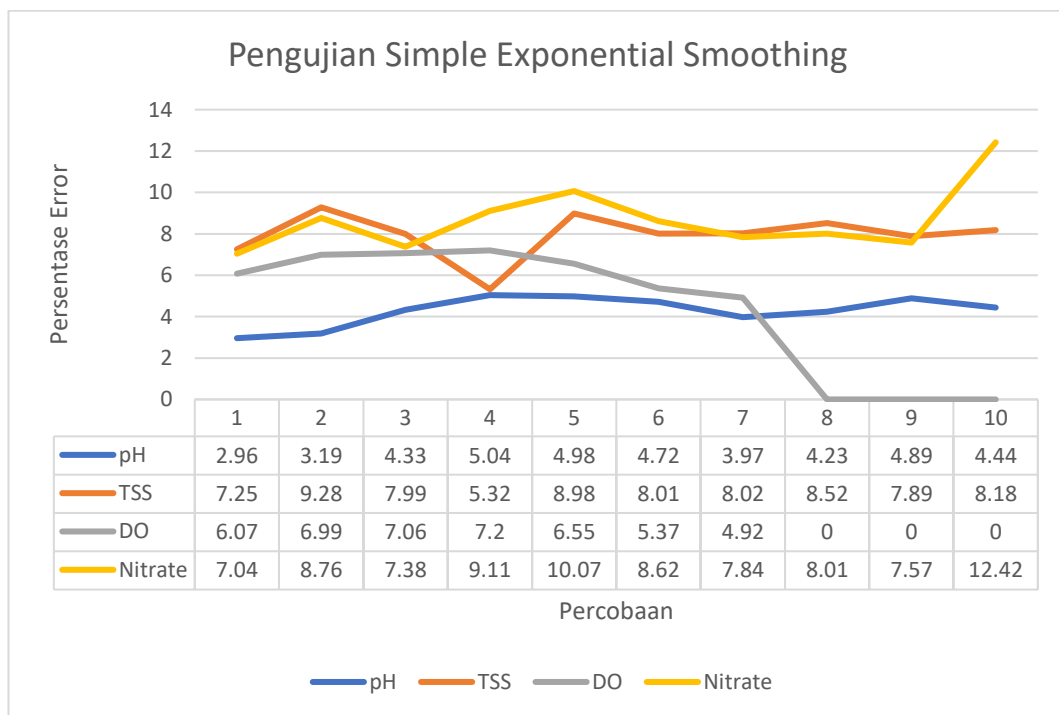
$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5 + 5}{5 + 5 + 0 + 0} = \frac{10}{10} = 1.00$$

$$Error Rate = \frac{FN + FP}{TP + TN + FP + FN} = \frac{0 + 0}{5 + 5 + 0 + 0} = \frac{0}{10} = 0$$

Dari hasil perhitungan diatas maka Akurasi dari sistem mencapai 100% dan error rate sekitar 0 %. Dari hasil tersebut sistem penentuan kualitas air cukup dapat dijadikan acuan karena memiliki akurasi yang cukup tinggi dan tidak terjadi error.

Dilakukan sebanyak 10 (sepuluh) kali pengujian nilai perkiraan dari penerapan algoritma *Simple Exponential Smoothing* dengan menggunakan *MAPE (Mean Absolute Percentage Error)* dengan membandingkan nilai perkiraan dengan nilai aktual. Seperti yang ditunjukkan pada gambar 26 maka diperoleh nilai untuk parameter **pH** 2.96% error terkecil dan 5.04% error terbesar dengan rata – rata error untuk 10 (sepuluh) kali percobaan adalah 4.28%, untuk parameter **TSS** 5.32% error terkecil dan 9.28% error

terbesar dengan rata – rata error untuk 10 (sepuluh) kali percobaan adalah 7.94%, untuk parameter **DO** 0% error terkecil dan 7.20% error terbesar dengan rata – rata error untuk 10 (sepuluh) kali percobaan adalah 4.42% dan untuk parameter **Nitrate** 7.04% error terkecil dan 12.42% error terbesar dengan rata – rata error untuk 10 (sepuluh) kali percobaan adalah 8.68%. dari hasil pengujian tersebut maka dapat disimpulkan bahwa perkiraan untuk setiap parameter – parameter kualitas air menunjukkan hasil yang baik.



Gambar 26. Grafik Pengujian Simple Exponential Smoothing

BAB V

PENUTUP

A. Kesimpulan

Dalam penelitian ini telah dapat diimplementasikan sistem pemantauan dan kontrol jarak jauh menggunakan jaringan internet atau disebut juga *Internet of Things (IoT)* berbasis website yang menggunakan protokol komunikasi yang efisien, ringan dan reliable yaitu *MQTT (Message Queue Telemetry Transport)* yang memang telah banyak digunakan dalam penerapan *IoT*. Penerapan *IoT* dengan menggunakan protokol *MQTT* menghasilkan sistem kontrol yang cukup baik dengan tingkat akurasi mencapai 100% dan error rate 0%. Penggunaan *Naïve Bayes* dirasa cukup baik dalam menyelesaikan permasalahan yang diangkat dalam penelitian ini. Penggunaan *Simple Exponential Smoothing* dalam melakukan prediksi dengan data deret waktu memberikan hasil yang baik, setelah diuji dengan menggunakan *MAPE (Mean Absolute Percentage Error)* diperoleh nilai rata-rata error maksimal 8.68% hal ini menunjukkan bahwa hasil perkiraan yang baik dan dapat digunakan untuk sumber informasi yang dapat dipertimbangkan untuk dapat menjadi acuan dalam proses pengambilan keputusan terkait pemantauan kualitas air sungai Mahakam. Secara umum penelitian ini telah berhasil membangun sistem pemantauan dan kontrol yang efisien dan reliable dengan memanfaatkan teknologi saat ini.

B. Saran

Adapun saran untuk penelitian dan pengembangan berikutnya serta diharapkan akan menambah referensi dalam penerapan sistem kontrol jarak jauh yaitu:

1. Perlu ditambahkan parameter lain seperti curah hujan, suhu udara yang kemungkinan akan mempengaruhi kualitas air sehingga proses prediksi lebih baik lagi.
2. Perlu dipertimbangkan untuk menerapkan sistem yang diajukan dan telah diuji pada penelitian ini untuk dapat diterapkan dalam dunia nyata.
3. Perlu diteliti lebih lanjut terkait skema pemantauan kualitas air yang terintegrasi dengan titik - titik lainnya serta memperhatikan kuat arus air sungai sehingga dapat dilakukan tindakan preventif pada saat diperoleh kualitas air yang menurun pada hulu sungai sebelum mencapai titik pompa air.

DAFTAR PUSTAKA

- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *ArXiv:1205.6904 [Cs]*. Retrieved from <http://arxiv.org/abs/1205.6904>
- Depari, A., Fernandes Carvalho, D., Bellagente, P., Ferrari, P., Sisinni, E., Flammini, A., & Padovani, A. (2019). An *IoT* Based Architecture for Enhancing the Effectiveness of Prototype Medical Instruments Applied to Neurodegenerative Disease Diagnosis. *Sensors*, *19*(7), 1564. <https://doi.org/10.3390/s19071564>
- Dizdarevic, J., Carpio, F., Jukan, A., & Masip-Bruin, X. (2019). Survey of Communication Protocols for Internet-of-Things and Related Challenges of Fog and Cloud Computing Integration. *ACM Computing Surveys*, *51*(6), 1–29. <https://doi.org/10.1145/3292674>
- Jerald, F., Anand, M., & Deepika, N. (2019). *Design of an Industrial IOT Architecture Based on MQTT Protocol for End Device to Cloud Communication*. *7*(6), 3.
- Kavi Priya, S., Shenbagalakshmi, G., & Revathi, T. (2017). Design of smart sensors for *Real Time* drinking water quality monitoring and contamination detection in water distributed mains. *International Journal of Engineering & Technology*, *7*(1.1), 47. <https://doi.org/10.14419/ijet.v7i1.1.8921>

- Lutfi, M., Muttaqien, H., Apriliani, A., Zainuddin, H., & Yuyun, Y. (2019). Application of the *Naïve Bayes Algorithm* and *Simple Exponential Smoothing* for Food Commodity Prices Forecasting. *Proceedings of the 1st International Conference on Science and Technology, ICOST 2019, 2-3 May, Makassar, Indonesia*. Presented at the 1st International Conference on Science and Technology, ICOST 2019, 2-3 May, Makassar, Indonesia, Makassar, Indonesia. <https://doi.org/10.4108/eai.2-5-2019.2284613>
- Naik, N. (2017). Choice of effective messaging protocols for *IoT* systems: *MQTT, CoAP, AMQP and HTTP*. *2017 IEEE International Systems Engineering Symposium (ISSE)*, 1–7. <https://doi.org/10.1109/SysEng.2017.8088251>
- Pavithra, D., & Balakrishnan, R. (2015). *IoT* based monitoring and control system for home automation. *2015 Global Conference on Communication Technologies (GCCT)*, 169–173. <https://doi.org/10.1109/GCCT.2015.7342646>
- Pawar, A. B., & Ghumbre, S. (2016). A survey on *IoT* applications, security challenges and counter measures. *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, 294–299. <https://doi.org/10.1109/CAST.2016.7914983>
- PERATURAN PEMERINTAH REPUBLIK INDONESIA. (n.d.). 28.
- Sarafov, V. (2018). *Comparison of IoT Data Protocol Overhead*. https://doi.org/10.2313/net-2018-03-1_02

- Singh, A., Meshram, S., Gujar, T., & Wankhede, P. R. (2016). Baggage tracing and handling system using RFID and *IoT* for airports. *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, 466–470. <https://doi.org/10.1109/CAST.2016.7915014>
- Singh, T. A., & Chandra, J. (2018). *IOT* Based Green House Monitoring System. *Journal of Computer Science*, 14(5), 639–644. <https://doi.org/10.3844/jcssp.2018.639.644>
- Vyas, D. J., & Rudani, N. N. (2018). *MQTT & IOT Based Control and Monitoring of Smart Green House*. 6(1), 6.
- Wang, M., Zhang, G., Zhang, C., Zhang, J., & Li, C. (2013). An *IoT*-based appliance control system for smart homes. *2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, 744–747. <https://doi.org/10.1109/ICICIP.2013.6568171>

LAMPIRAN

Source code Arduino

```
#define TINY_GSM_MODEM_SIM900
#define SerialMon Serial
#include <SoftwareSerial.h>
SoftwareSerial SerialAT(3, 2); // RX, TX
#define TINY_GSM_DEBUG SerialMon
#define GSM_AUTOBAUD_MIN 9600
#define GSM_AUTOBAUD_MAX 115200
#define TINY_GSM_USE_GPRS true
#define TINY_GSM_USE_WIFI false
#define GSM_PIN ""
const char apn[] = "internet";
const char gprsUser[] = "";
const char gprsPass[] = "";
const char* broker = "mqtt.haitek.id";
const char* topicIntake = "intake";
#include <TinyGsmClient.h>
#include <PubSubClient.h>
TinyGsm modem(SerialAT);
TinyGsmClient client(modem);
PubSubClient mqtt(client);
#define RELAY_PIN 13
int relayStatus = LOW;
long lastReconnectAttempt = 0;

void mqttCallback(char* topic, byte* payload, unsigned int len) {
  if (String(topic) == "intake/off") {
    SerialMon.println("Engine [Off]");
```

```
        relayStatus = HIGH;
    } else{
        SerialMon.println("Engine [On]");
        relayStatus = LOW;
    }
    digitalWrite(RELAY_PIN, relayStatus);
}
```

```
boolean mqttConnect() {
    SerialMon.print("Koneksi ke");
    SerialMon.print(broker);
    boolean status = mqtt.connect("mqtt","mqtt","Test#123");
    if (status == false) {
        SerialMon.println("Gagal");
        return false;
    }
    SerialMon.println("Sukses");
    mqtt.subscribe("intake/off");
    mqtt.subscribe("intake/on");
    return mqtt.connected();
}
```

```
void setup() {
    // Set console baud rate
    SerialMon.begin(115200);
    delay(10);
    pinMode(RELAY_PIN, OUTPUT);
    SerialMon.println("Dalam Proses...");
    SerialAT.begin(9600);
}
```

```
delay(3000);
SerialMon.println("Inisialisasi modem...");
modem.restart();
String modemInfo = modem.getModemInfo();
SerialMon.print("Modem Info: ");
SerialMon.println(modemInfo);

if ( GSM_PIN && modem.getSimStatus() != 3 ) {
    modem.simUnlock(GSM_PIN);
}

modem.gprsConnect(apn, gprsUser, gprsPass);
SerialMon.print("Konek Ke Jaringan...");
if (!modem.waitForNetwork()) {
    SerialMon.println(" gagal");
    delay(10000);
    return;
}

SerialMon.println(" sukses");
if (modem.isNetworkConnected()) {
    SerialMon.println("Jaringan Terhubung");
}

SerialMon.print(F("Koneksi Data ke "));
SerialMon.print(apn);
if (!modem.gprsConnect(apn, gprsUser, gprsPass)) {
    SerialMon.println(" gagal");
    delay(10000);
    return;
}

SerialMon.println(" Sukses");
```

```

if (modem.isGprsConnected()) {
    SerialMon.println("GPRS terkoneksi");
}
mqtt.setServer(broker, 1883);
mqtt.setCallback(mqttCallback);
}

void loop() {
    if (!mqtt.connected()) {
        SerialMon.println("=== MQTT Server Tidak Terhubung ===");
        // Reconnect every 10 seconds
        unsigned long t = millis();
        if (t - lastReconnectAttempt > 10000L) {
            lastReconnectAttempt = t;
            if (mqttConnect()) {
                lastReconnectAttempt = 0;
            }
        }
        delay(100);
        return;
    }
    mqtt.loop();
}

```

Naïve Bayes

```
<?php
```

```
defined('BASEPATH') OR exit('No direct script access allowed');
```

```

class naivebayes extends CI_Controller
{

    function __construct()
    {
        parent::__construct();
        $this->load->model('model_master',"TRUE");
        $this->load->model('model_naivebayes',"TRUE");
        $this->load->helper('sendgmail');
        $this->session->unset_userdata('logged_in');
    }

    public function index()
    {

        $jum = $this->model_naivebayes->get_jumlah();
        $datas = $this->model_naivebayes->get();

        if($datas !=false){
            foreach ($datas as $v2) {
                echo "<br>-----
<br>";
                echo "Langkah 1: Menghitung Jumlah Status Kualitas Air 'Baik'
dan 'Tidak Baik'<br>";
                echo "<ul><li>Baik : $jum->jum_b / $jum->jum </li> <li>Tidak
Baik : $jum->jum_t / $jum->jum </li></ul>";
                echo "<br>Langkah 2: menghitung jumlah kasus yang sama
dengan kelas yang sama<br>";
                if($v2->k_ph=='B'){

```

```

        echo "<ul><li>P (Kondisi PH = Baik | Y= Baik) : $jum-
>jum_ph_b_b / $jum->jum_b </li> "
        ."<li>P (Kondisi PH = Baik | Y= Tidak Baik) : $jum-
>jum_ph_t_b / $jum->jum_t </li></ul>";

        $step2_ph_b = $jum->jum_ph_b_b / $jum->jum_b;
        $step2_ph_t = $jum->jum_ph_t_b / $jum->jum_t;
    } else {
        echo "<ul><li>P (Kondisi PH = Tidak Baik | Y= Baik) : $jum-
>jum_ph_b_t / $jum->jum_b </li> "
        ."<li>P (Kondisi PH = Tidak Baik | Y= Tidak Baik) : $jum-
>jum_ph_t_t / $jum->jum_t </li></ul>";

        $step2_ph_b = $jum->jum_ph_b_t / $jum->jum_b;
        $step2_ph_t = $jum->jum_ph_t_t / $jum->jum_t;
    }

    if($v2->k_tss=='B'){
        echo "<ul><li>P (Kondisi TSS = Baik | Y= Baik) : $jum-
>jum_tss_b_b / $jum->jum_b </li> "
        ."<li>P (Kondisi TSS = Baik | Y= Tidak Baik) : $jum-
>jum_tss_t_b / $jum->jum_t </li></ul>";

        $step2_tss_b = $jum->jum_tss_b_b / $jum->jum_b;
        $step2_tss_t = $jum->jum_tss_t_b / $jum->jum_t;
    } else {
        echo "<ul><li>P (Kondisi TSS = Tidak Baik | Y= Baik) : $jum-
>jum_tss_b_t / $jum->jum_b </li> "
        ."<li>P (Kondisi TSS = Tidak Baik | Y= Tidak Baik) : $jum-
>jum_tss_t_t / $jum->jum_t </li></ul>";

        $step2_tss_b = $jum->jum_tss_b_t / $jum->jum_b;

```

```

    $step2_tss_t = $jum->jum_tss_t_t / $jum->jum_t;
}

if($v2->k_do=='B'){
    echo "<ul><li>P (Kondisi DO = Baik | Y= Baik) : $jum-
>jum_do_b_b / $jum->jum_b </li> "
        ."<li>P (Kondisi DO = Baik | Y= Tidak Baik) : $jum-
>jum_do_t_b / $jum->jum_t </li></ul>";

    $step2_do_b = $jum->jum_do_b_b / $jum->jum_b;
    $step2_do_t = $jum->jum_do_t_b / $jum->jum_t;
} else {
    echo "<ul><li>P (Kondisi DO = Tidak Baik | Y= Baik) : $jum-
>jum_do_b_t / $jum->jum_b </li> "
        ."<li>P (Kondisi DO = Tidak Baik | Y= Tidak Baik) : $jum-
>jum_do_t_t / $jum->jum_t </li></ul>";

    $step2_do_b = $jum->jum_do_b_t / $jum->jum_b;
    $step2_do_t = $jum->jum_do_t_t / $jum->jum_t;
}

if($v2->k_nitrate=='B'){
    echo "<ul><li>P (Kondisi Nitrate = Baik | Y= Baik) : $jum-
>jum_nitrate_b_b / $jum->jum_b </li> "
        ."<li>P (Kondisi Nitrate = Baik | Y= Tidak Baik) : $jum-
>jum_nitrate_t_b / $jum->jum_t </li></ul>";

    $step2_nitrate_b = $jum->jum_nitrate_b_b / $jum->jum_b;
    $step2_nitrate_t = $jum->jum_nitrate_t_b / $jum->jum_t;
} else {
    echo "<ul><li>P (Kondisi Nitrate = Tidak Baik | Y= Baik) :
$jum->jum_nitrate_b_t / $jum->jum_b </li> "

```

```

        .<li>P (Kondisi Nitrate = Tidak Baik | Y= Tidak Baik) :
$jum->jum_nitrate_t_t / $jum->jum_t </li></ul>";

        $step2_nitrate_b = $jum->jum_nitrate_b_t / $jum->jum_b;
        $step2_nitrate_t = $jum->jum_nitrate_t_t / $jum->jum_t;
    }

    echo "<br>Langkah 3: kalikan semua hasil variable Status
Kualitas Air 'Baik' dan 'Tidak Baik'<br>";

    $p_baik =
$step2_ph_b*$step2_tss_b*$step2_do_b*$step2_nitrate_b*($jum->jum_b
/ $jum->jum);

    $p_tidakbaik =
$step2_ph_t*$step2_tss_t*$step2_do_t*$step2_nitrate_t*($jum->jum_t /
$jum->jum);

    echo "P [Baik] =
$step2_ph_b*$step2_tss_b*$step2_do_b*$step2_nitrate_b*($jum->jum_b
/ $jum->jum) = $p_baik<br>";

    echo "P [Tidak Baik] =
$step2_ph_t*$step2_tss_t*$step2_do_t*$step2_nitrate_t*($jum->jum_t /
$jum->jum) = $p_tidakbaik <br>";

    echo "<br>Langkah 4: Bandingkan Hasil P[Baik] jika lebih
besar P[Tidak Baik] maka Status Kualitas Air 'Baik'<br>"

    . "jika sebaliknya maka Status Kualitas Air 'Tidak Baik'<br>";

    if($p_baik > $p_tidakbaik) {

        echo "Status Air <span style='font-weight:bold'>Baik</span>
Karena $p_baik > $p_tidakbaik ";

        $status="B";

        shell_exec('mosquitto_pub -h mqtt.haitek.id -t "intake/on" -m
"true" -u "mqtt" -P "Test#123");

    } else {

        echo "Status Air <span style='font-weight:bold'>Tidak
Baik</span> Karena $p_baik < $p_tidakbaik ";

```

```

$status="T";

shell_exec('mosquitto_pub -h mqtt.haitek.id -t "intake/off" -m
>true" -u "mqtt" -P "Test#123"');

$msgEmail="<p>Dikarenakan Kualitas Air Tidak Baik Pompa
Intake telah dimatikan secara otomatis</p>"

. "<ul><li>Waktu: ".DateToIndo($v2->sensorlog)."</li> "
. "<li>pH : $v2->at5_ph</li>"
. "<li>TSS : $v2->at6_tss</li>"
. "<li>DO : $v2->at6_do</li>"
. "<li>Nitrate : $v2->at5_nitrate</li>"
. "</ul>";

$msgSMS = "Kualitas Air Tidak Baik, Pompa Dimatikan [pH :
$v2->at5_ph, TSS : $v2->at6_tss,DO : $v2->at6_do,Nitrate : $v2-
>at5_nitrate]";

$penerima = $this->model_naivebayes->get_penerima();
if($penerima!=false){
    foreach ($penerima as $vp) {
        $param = array(
            "nohp"=>$vp->nohp,
            "email"=>$vp->email,
            "msgemail"=> addslashes($msgEmail),
            "msghp"=> addslashes($msgSMS),
        );
        $this->model_master->insert("tb_send",$param);
    }
}

echo "<br>-----
<br>";

$this->db->query("update tb_scan set kondisi='$status' where
idx='$v2->idx'");

```

```
    }  
  }  
  
  }  
}  
?>
```

Simple Exponential Smoothing

```
<?php
```

```
defined('BASEPATH') OR exit('No direct script access allowed');
```

```
class forecast extends CI_Controller {
```

```
    /**
```

```
     * Index Page for this controller.
```

```
     *
```

```
     * Maps to the following URL
```

```
     *         http://example.com/index.php/welcome
```

```
     * - or -
```

```
     *         http://example.com/index.php/welcome/index
```

```
     * - or -
```

```
     * Since this controller is set as the default controller in
```

```
     * config/routes.php, it's displayed at http://example.com/
```

```
     *
```

```
     * So any other public methods not prefixed with an underscore will
```

```
     * map to /index.php/welcome/<method_name>
```

```
     * @see http://codeigniter.com/user_guide/general/urls.html
```

```
    */
```

```
    var $template = 'template';
```

```
private $_path_page = "pages/forecast/";
private $_path_js = "master/";
private $_judul = 'Dashboard';
private $_page_index = 'index';
private $_page_mhs = 'forecast_mhs';
private $_page_dsn = 'forecast_dsn';
private $_page_opr = 'forecast_opr';
```

```
function __construct()
{
    parent::__construct();
    $this->load->model('model_master',"TRUE);
    $this->load->model('model_forecast',"TRUE);
    //$this->load->model('model_daftar',"TRUE);
    $this->load->helper('menu');
    // $this->load->helper('crudakses');
}
```

```
private function get_master($pages)
{
    $session_data = $this->session->userdata('logged_in');
    //print_r($session_data);
    $menu2 = menu($session_data['susrsgrupnama']);

    $data['page'] = $pages;
    $data['susrnama'] = $session_data['susrnama'];
    $data['susrsgrupnama'] =
    $session_data['susrsgrupnama'];
```

```

        $data['sgroupketerangan'] =
$session_data['sgroupketerangan'];
        $data['susprofil'] = $session_data['susprofil'];
        $data['bulan'] = array("", "Jan", "Feb", "Mar",
                                "Apr", "Mei", "Jun",
                                "Jul", "Aug", "Sep",
                                "Okt", "Nov", "Dec");
        $data['bulanP'] = array("-Pilih-", "Januari", "Februari", "Maret",
                                "April", "Mei", "Juni",
                                "Juli", "Agustus",
"September",
                                "Oktober", "Nopember",
"Desember");
        $data['page_response'] = $this->_page_opr;
        $data['menu2'] = $menu2;
        $data['judul'] = $this->_judul;

        return $data;
    }

    public function index()
    {
        if($this->session->userdata('logged_in')) //cek user logged
        {
            $data = $this->get_master($this->_path_page.$this-
>_page_index);
            $data['scripts'] = array($this->_path_js.'forecast');
            $data['datas']=$this->model_forecast->get();
            $this->load->view($this->template, $data);
        }
        else

```

```

        {
            redirect('login','refresh');
        }
    }

public function loaddatas()
{
    if($this->session->userdata('logged_in')) //cek user logged
    {
        $tanggal = $this->input->post("tanggal");
        for($j=0;$j<24;$j++){
            $jam = str_pad($j, 2, "0", STR_PAD_LEFT);
            $datas = $this->model_forecast->get_logscan($tanggal,
$jam);

            if($datas!=false){
                $Ft_N_ph = $datas[0]->at5_ph;
                $Ft_N_tss = $datas[0]->at6_tss;
                $Ft_N_do = $datas[0]->at6_do;
                $Ft_N_nitrate = $datas[0]->at5_nitrate;
                foreach ($datas as $v) {
                    $Ft_N_ph = $this-
>simple_exponential_smoothing(0.9, $v->at5_ph, $Ft_N_ph);
                    $Ft_N_tss = $this-
>simple_exponential_smoothing(0.9, $v->at6_tss, $Ft_N_tss);
                    $Ft_N_do = $this-
>simple_exponential_smoothing(0.9, $v->at6_do, $Ft_N_do);
                    $Ft_N_nitrate = $this-
>simple_exponential_smoothing(0.9, $v->at5_nitrate, $Ft_N_nitrate);
                }
            }
        }
    }
}

```

```

        $ph[$jam]=$Ft_N_ph;
        $tss[$jam]=$Ft_N_tss;
        $do[$jam]=$Ft_N_do;
        $nitrate[$jam]=$Ft_N_nitrate;
    }
    $data['ph']=$ph;
    $data['tss']=$tss;
    $data['do']=$do;
    $data['nitrate']=$nitrate;
    $data['tanggal']=pgtgl($tanggal);
    $this->load->view("pages/forecast/response", $data);
}
else
{
    redirect('login','refresh');
}
}

public function simple_exponential_smoothing($alpha,$Dt,$Ft)
{
    if($this->session->userdata('logged_in')) //cek user logged
    {
        $Ft_N = ($alpha * $Dt) + ((1 - $alpha) * $Ft);
        return $Ft_N;
    }
    else
    {

```

```
redirect('login','refresh');
```

```
}
```

```
}
```

```
}
```