

**PURWARUPA SISTEM MONITORING SUMBER DAYA AIR DI
WILAYAH PERKOTAAN INFORMAL DENGAN ARDUINO
PADA STUDI KASUS BANK AIR KAMPUNG TERBAN
YOGYAKARTA**



Disusun oleh:

N a m a : RIZKY ARIF WINDIARTO

NIM : 12523212

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2018

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PURWARUPA SISTEM MONITORING SUMBER DAYA AIR DI
WILAYAH PERKOTAAN INFORMAL DENGAN ARDUINO
PADA STUDI KASUS BANK AIR KAMPUNG TERBAN
YOGYAKARTA**



(Ari Sujarwo, S.Kom., MIT (Hons))

HALAMAN PENGESAHAN DOSEN PENGUJI

**PURWARUPA SISTEM MONITORING SUMBER DAYA AIR DI
WILAYAH PERKOTAAN INFORMAL DENGAN ARDUINO
PADA STUDI KASUS BANK AIR KAMPUNG TERBAN
YOGYAKARTA**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk
memperoleh gelar Sarjana Teknik Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 4 April 2018

Tim Penguji

Ari Sujarwo, S.Kom., MIT (Hons)

Anggota 1

Fietyata Yudha, S.Kom., M.Kom.

Anggota 2

Andhika Giri Persada, S.Kom., M.Eng.

Mengetahui,

Ketua Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Indonesia



(Hendrik, S.T., M.Eng.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Rizky Arif Windiarto

NIM : 12523212

Tugas Akhir dengan judul:

**PURWARUPA SISTEM MONITORING SUMBER DAYA AIR DI
WILAYAH PERKOTAAN INFORMAL DENGAN ARDUINO
PADA STUDI KASUS BANK AIR KAMPUNG TERBAN
YOGYAKARTA**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 19 Maret 2018



(Rizky Arif Windiarto)

HALAMAN PERSEMBAHAN

*Alhamdulillah Rabbil'Alamin, puji syukurku kepada-Mu Ya Allah Subhanallahu Wa Ta'ala
atas salah satu dari sekian banyak nikmat yang tak mungkin bisa dihitung dengan
terselesaikannya karya ini.*

*Shalawat serta keselamatan selalu terlimpah kepada Rasulullah Muhammad Salallahu
'Alaihi Wasallam sebagai suri tauladan yang berakhlak mulia.
kupersembahkan karya Tugas Akhirku ini kepada orang-orang yang kusayangi*

Ibuku Heny Arifanti

Ayahku Agus Winarto

Dan ketiga adikku Anugrah, Karuniawan dan Athallah.

Terima Kasih untuk semua dukungan, semangat, do'a, cinta dan kasih sayangnya

HALAMAN MOTO

“We are more often frightened than hurt; and we suffer more from imagination than from reality”

- Lucius Annaeus Seneca

“The cure for boredom is curiosity. There is no cure for curiosity”

-Ellen Parr

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan”

-Qs. Al-Insyirah : 5

KATA PENGANTAR

Assalaamu'alaikum Wr. Wb.

Alhamdulillah Robbil 'Alamin, puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Purwarupa Sistem Monitoring Sumber Daya Air Di Wilayah Perkotaan Informal Dengan Arduino Pada Studi Kasus Bank Air Kampung Terban Yogyakarta”**.

Laporan ini disusun sebagai salah satu persyaratan yang harus dipenuhi dalam rangka menyelesaikan pendidikan pada jenjang Strata 1 di Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Tugas akhir ini dapat terselesaikan atas bantuan, dukungan, dan bimbingan yang diberikan dari berbagai pihak, maka dari itu penulis mengucapkan terimakasih kepada :

1. Allah SWT atas segala limpahan nikmat, rahmat dan hidayah-Nya.
2. Rasulullah Muhammad SAW yang menjadi teladan dan panutan.
3. Bapak Hendrik, S.T, M.Eng., selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Ibu Nur Wijyaning Rahayu, S.Kom., M.Kom., selaku Sekretaris Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Ari Sujarwo, S.Kom., MIT (Hons), selaku dosen pembimbing tugas akhir yang telah memberikan waktunya untuk membimbing penulis dalam menyelesaikan tugas akhir.
6. Bapak dan ibu dosen Jurusan Teknik Informatika yang telah membagi ilmunya kepada penulis.
7. Kedua orang tua, Agus Winarto dan Heny Arifanti yang telah memberikan dukungan moril maupun materiil dalam pembuatan tugas akhir ini.
8. Ketiga adikku Anugrah, Karuniawan dan Athallah yang telah memberikan semangat dan doa.
9. Galih Andyan Anindita yang telah menyediakan tempat untuk penulis menyelesaikan tugas akhir.
10. Teman-teman Otentik Kopi yang menemani penulis saat mengerjakan tugas akhir ini.
11. Seluruh teman-teman SISJARKOM yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan motivasi dan doa.

12. Semua pihak yang tidak dapat penulis sebutkan satu persatu, terimakasih atas bantuan dan do'anya.

Tugas akhir ini tidak lepas dari kekurangan dan ketidaksempurnaan dikarenakan terbatasnya kemampuan dan pengalaman penulis, oleh karena itu kritik dan saran yang membangun sangat penulis harapkan agar dapat lebih baik lagi. Semoga laporan ini dapat diterima dan bermanfaat bagi para pembacanya. Amin.

Wassalaamu 'alaikum Wr. Wb.

Yogyakarta, 19 Maret 2018



(Rizky Arif Windiarto)

SARI

Air merupakan sumber daya yang penting bagi manusia. Keberadaan air sangat dibutuhkan untuk menunjang kehidupan manusia. Indonesia sebagai negara perairan memiliki jumlah sumber air yang hampir tak terbatas. Tidak semua sumber air yang ada di Indonesia dapat dimanfaatkan untuk sumber air minum dan menunjang kegiatan sehari-hari hanya sumber air tawar yang dapat dimanfaatkan.

Namun, Air merupakan zat yang mudah tercemar. Pencemaran air seringkali tidak dapat dideteksi secara langsung. Pencemaran ini dapat diakibatkan oleh berbagai hal seperti limbah pabrik dan rumah tangga atau karena faktor alam. Pemantauan yang dilakukan secara periodik diperlukan untuk mengetahui kualitas sebuah sumber air dari waktu ke waktu.

Smart city merupakan tren teknologi yang sedang berkembang pesat. *Smart city* adalah gabungan teknologi dengan kehidupan sehari-hari. *Smart city* mencakup berbagai bidang seperti otomotif, pertanian, sosial, ekonomi, kesehatan dan sebagainya. Pada *Smart city* seluruh aspek kehidupan manusia di dalamnya terkoneksi dengan jaringan internet. Hal ini disebut sebagai *Internet of Things* atau IoT.

IoT yang merupakan inti dari sebuah *smart city* mengkoneksikan setiap perangkat seperti telepon, pendingin udara bahkan speaker. Perangkat IoT biasanya memiliki fungsi yang spesifik. Contohnya adalah teknologi *energy sensing* yang sedang dikembangkan dapat digunakan untuk memantau penggunaan listrik secara akurat sehingga membantu dalam memantau penggunaan listrik. Menggunakan sensor yang tepat, IoT dapat dimanfaatkan dalam pemantauan kualitas air.

Sistem monitoring sumber daya air merupakan sebuah sistem yang dikembangkan untuk melakukan pemantauan terhadap kualitas sebuah mata air. Perancangan proses sistem dibuat dengan menggunakan UML (*Unified Modeling Language*) yang kemudian menjadi dasar dalam pembuatan purwarupa sistem. Purwarupa dari sistem dibuat dengan menggunakan Arduino dan Raspberry Pi.

Sistem monitoring sumber daya air merupakan solusi yang tepat untuk memantau kualitas mata air yang terdapat di wilayah Terban karena memudahkan proses pemantauan kualitas pada Bank Air AirKami sehingga kualitas air dapat dipantau secara terus menerus.

Kata kunci: Air, Pemantauan, *Smart city*, IoT, Sistem monitoring sumber daya air, UML, Arduino, Raspberry Pi.

GLOSARIUM

Amplifier	penguat sinyal listrik.
ARM	arsitektur mikroprosesor yang digunakan pada berbagai perangkat.
DC	arus listrik searah.
Clock speed	kecepatan pemrosesan perangkat pemroses utama.
CPU	perangkat pemroses utama.
EEPROM	jenis memori penyimpanan hanya baca yang dapat dan dihapus dengan arus listrik.
Electromagnetic	medan magnet yang diakibatkan oleh arus listrik.
Flag	variabel pada pemrograman yang digunakan untuk menandai sebuah aksi.
Flash	jenis memori penyimpanan yang dapat menyimpan data saat arus listrik diputus.
GPIO	pin yang terdapat pada papan sirkuit.
GPU	pemroses grafis pada perangkat komputer.
Internet of things	konsep dimana beberapa perangkat keras dan perangkat lunak dapat saling terhubung dan berkomunikasi satu dengan yang lainnya melalui sebuah jaringan telekomunikasi.
Konduktivitas	kemampuan suatu bahan menghantarkan arus listrik.
Mikrokontroler	sebuah semikonduktor yang dapat diprogram sehingga memiliki fungsi tertentu seperti mengendalikan perangkat lain maupun membaca nilai yang dihasilkan oleh sensor
Ports	konektor
Probe	ujung sensor yang digunakan untuk membaca data
Purwarupa	desain awal sebuah perangkat.
Queue	struktur data yang berbentuk seperti antrian
RAM	jenis memori penyimpanan yang digunakan untuk menyimpan data sementara.
Software repository	tempat penyimpanan perangkat lunak
Salinitas	kadar garam yang terlarut dalam air
Shield	papan sirkuit yang dapat dipasang pada papan sirkuit lainnya dan dapat memberikan fungsi tambahan

SoC	sebuah sirkuit terintegrasi yang menggabungkan berbagai komponen komputer.
Solenoid	kumparan dari kabel yang dililitkan secara rapat.
SRAM	salah satu jenis dari RAM.
Tegangan input	tegangan listrik masukan pada sebuah perangkat.
Tegangan operasi	tegangan listrik yang dibutuhkan sebuah perangkat untuk beroperasi.
Thread	bagian dari sebuah program yang memiliki alur sendiri dan berjalan secara independen.
Turbiditas	tingkat kekeruhan air.
Turbine	mekanisme yang digunakan untuk menghasilkan listrik dari gerakan air melalui baling-baling.
Ultrasonic	gelombang suara frekuensi tinggi.
Waterfall	Metode pengembangan perangkat lunak.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian	4
1.6.1 Analisis Kebutuhan.....	5
1.6.1 Perancangan Sistem	5
1.6.2 Implementasi Sistem.....	5
1.6.3 Pengujian Sistem.....	5
1.6.4 Sistematika Penulisan	5
BAB II KAJIAN PUSTAKA.....	7
2.1 Air.....	7
2.1.1 Air pada wilayah informal	7
2.1.2 Pentingnya pemantauan kualitas air.....	7
2.2 Peran teknologi terkait pemantauan kualitas air.....	8
2.2.1 Sistem benam dan mikrokontroler	8
2.2.2 Pemanfaatan sensor pada sistem benam	9
2.2.3 Teknologi jaringan sensor nirkabel.....	10
2.2.4 Konsep <i>smart city</i> dan IoT	10

2.2.5	Contoh implementasi IoT pada kehidupan nyata.....	11
BAB III METODOLOGI.....		14
3.1	Gambaran umum sistem.....	14
3.2	Analisis kebutuhan	16
3.2.1	Kebutuhan masukan.....	16
3.2.2	Kebutuhan keluaran	16
3.2.3	Kebutuhan proses.....	16
3.2.4	Kebutuhan perangkat keras.....	17
3.3	Perancangan sistem	23
3.3.1	Metode perancangan	24
3.3.2	Diagram aktivitas.....	24
3.3.3	Perancangan diagram alir.....	25
3.3.4	Perancangan diagram alir data	29
3.4	Perancangan basis data	33
3.4.1	Struktur tabel.....	34
3.4.2	Perancangan ERD	35
3.5	Rencana Implementasi.....	36
3.6	Rencana Pengujian Sistem	36
3.6.1	Langkah pengujian fungsi.....	37
BAB IV HASIL DAN PEMBAHASAN		38
4.1	Hasil.....	38
4.1.1	Perakitan perangkat sensor	38
4.1.2	Instalasi sistem operasi Raspbian.....	39
4.1.3	Implementasi basis data	40
4.1.4	Konfigurasi XBee	42
4.2	Pembahasan	43
4.2.1	Inisialisasi perangkat.....	43
4.2.2	Pembacaan sensor	47
4.2.3	Pengiriman dan penerimaan.....	49
4.3	Pengujian fungsi perangkat	50
4.3.1	Inisialisasi perangkat.....	50
4.3.2	Pembacaan dan pengiriman data.....	53
4.3.3	Hasil pengujian fungsi	56
4.4	Pengujian sensor.....	57

4.5 Kelebihan dan kekurangan purwarupa sistem.....	59
BAB V KESIMPULAN DAN SARAN	60
5.1 Kesimpulan.....	60
5.2 Saran.....	60
DAFTAR PUSTAKA	62
LAMPIRAN.....	64

DAFTAR TABEL

Tabel 3.1 Tabel <i>node</i>	34
Tabel 3.2 Tabel Sensor	34
Tabel 3.3 Tabel Location	34
Tabel 3.4 Tabel SensorValue	35
Tabel 3.5 Tabel pengujian fungsi perangkat.....	37
Tabel 4.1 Hasil pengujian fungsi perangkat	56
Tabel 4.2 Hasil pembacaan sampel.....	59

DAFTAR GAMBAR

Gambar 2.1 <i>Smart Citizen Kit</i>	12
Gambar 2.2 Tampilan situs <i>Smart Citizen Kit</i>	12
Gambar 2.3 Alat pakan eFishery	13
Gambar 3.1 Gambaran sistem.....	14
Gambar 3.2 Titik penempatan perangkat.....	15
Gambar 3.3 Bank AirKami	15
Gambar 3.4 Raspberry Pi	17
Gambar 3.5 Arduino UNO.....	18
Gambar 3.6 XBEE PRO dan antena 9dBi	19
Gambar 3.7 XBee <i>Shield</i>	19
Gambar 3.8 Termokopel dengan modul MAX6675	20
Gambar 3.9 Sensor keasaman dan <i>amplifier</i>	20
Gambar 3.10 Sensor aliran air	21
Gambar 3.11 Sensor konduktivitas	21
Gambar 3.12 Sensor <i>turbidity</i>	22
Gambar 3.13 Katup solenoid (kiri) dan Pompa DC (kanan)	22
Gambar 3.14 Relai	23
Gambar 3.15 Aktivitas sistem.....	24
Gambar 3.16 Diagram alir perangkat sensor	25
Gambar 3.17 Diagram alir <i>main thread</i>	26
Gambar 3.18 Diagram alir <i>thread</i> serialWatcher.....	27
Gambar 3.19 Diagram alir <i>thread</i> worker.....	28
Gambar 3.20 Diagram alir <i>thread DbWatcher</i>	29
Gambar 3.21 Diagram alir data level nol.....	30
Gambar 3.22 Diagram alir data level satu	30
Gambar 3.23 Diagram alir data level dua proses dua	31
Gambar 3.24 Diagram alir data level dua proses tiga	32
Gambar 3.25 Diagram alir data level dua proses empat	32
Gambar 3.26 Diagram alir data level dua proses lima.....	33
Gambar 3.27 Diagram relasi tabel	35
Gambar 4.1 Perangkat sensor	38
Gambar 4.2 Sensor kualitas air	39

Gambar 4.3 Perangkat sentral yang terhubung dengan laptop	39
Gambar 4.4 Konfigurasi alamat IP	40
Gambar 4.5 Basis data WaterMonitoring	40
Gambar 4.6 Isi tabel Location.....	41
Gambar 4.7 Isi tabel Node	41
Gambar 4.8 Isi tabel Sensor	41
Gambar 4.9 Isi tabel SensorValue	42
Gambar 4.10 XBee perangkat sentral	42
Gambar 4.11 XBee perangkat.....	43
Gambar 4.12 Kode konfigurasi waktu jeda dan jumlah pembacaan	44
Gambar 4.13 Kode membaca keluaran sensor aliran air	44
Gambar 4.14 Konfigurasi perangkat.....	45
Gambar 4.15 Pengambilan daftar sensor dan perangkat sensor	45
Gambar 4.16 Kode untuk menjalankan <i>thread</i>	46
Gambar 4.17 Kode <i>thread</i> dbWatcher	46
Gambar 4.18 Kode sumber membuang sampel	47
Gambar 4.19 Kode sumber mengisi sampel	48
Gambar 4.20 Kode untuk membaca sensor <i>turbidity</i>	48
Gambar 4.21 Kode untuk membaca sensor suhu	48
Gambar 4.22 Kode untuk membaca sensor pH	48
Gambar 4.23 Kode untuk membaca sensor konduktivitas.....	48
Gambar 4.24 Kode untuk mengirim data.....	49
Gambar 4.25 Kode <i>thread</i> serialWatcher saat menerima data	49
Gambar 4.26 Kode untuk menyimpan data yang dijalankan <i>thread</i> worker	50
Gambar 4.27 Inisialisasi perangkat sensor.....	51
Gambar 4.28 Inisialisasi MainThread.....	51
Gambar 4.29 Tahap menjalankan <i>thread</i>	52
Gambar 4.30 <i>Thread</i> dbWatcher.....	52
Gambar 4.31 Sebelum <i>Thread</i> dbWatcher berjalan.....	52
Gambar 4.32 Sesudah <i>Thread</i> dbWatcher berjalan	53
Gambar 4.33 Output pembuangan dan pengambilan sampel	53
Gambar 4.34 Perangkat membuang sampel.....	53
Gambar 4.35 Perangkat mengambil sampel	53
Gambar 4.36 Output pembacaan nilai sensor	54

Gambar 4.37 Perangkat membaca nilai keluaran sensor	54
Gambar 4.38 Konstruksi dan pengiriman	55
Gambar 4.39 Data diterima dan disimpan	55
Gambar 4.40 Data tersimpan pada basis data	55
Gambar 4.41 Pengujian sensor pada air mineral	57
Gambar 4.42 Hasil pengujian pada air mineral	57
Gambar 4.43 Pengujian sensor pada air perasan jeruk	58
Gambar 4.44 Hasil pengujian pada air perasan jeruk	58

BAB I

PENDAHULUAN

1.1 Latar Belakang

Air merupakan sebuah senyawa kimia yang paling penting bagi kehidupan di bumi. Hampir 70% dari permukaan bumi tertutup oleh air (Angel dan Wolseley, 1992). Air menjadi sebuah ekosistem dengan berbagai macam kehidupan di dalamnya. Air juga merupakan senyawa yang paling banyak terdapat dalam tubuh manusia. Tersedianya sumber air bersih sangat penting bagi manusia. Selain sebagai sumber air minum, air bersih dapat digunakan untuk mandi dan mencuci serta untuk pengairan lahan. Sumber air bersih dapat berasal dari mata air, sumur, danau maupun sungai. Jumlah manusia yang semakin bertambah mengakibatkan kebutuhan terhadap air bersih juga bertambah.

Indonesia sebagai sebuah negara yang besar memiliki banyak sumber air yang tersebar di berbagai daerah. Namun, akses terhadap sumber air bersih di Indonesia masih cukup sulit. Banyak masyarakat di pedesaan atau bahkan di kota besar yang masih mengambil air dari sumber air seperti sumur, danau atau sungai yang mungkin tercemar. Pencemaran ini dapat disebabkan oleh Mikroorganisme, senyawa anorganik, bahan radioaktif, panas, dan nutrisi tanaman. Ciri-ciri air yang mengalami pencemaran sangat bervariasi tergantung dari jenis air dan polutannya dan untuk mengetahui apakah sebuah sumber air terpolusi, diperlukan pengujian untuk mengetahui sifat-sifat air sehingga dapat diketahui apakah terjadi penyimpangan batasan-batasan polusi air (Fardiaz, 1992). Pengujian air biasanya dilakukan secara konvensional dengan mengambil sampel air dan menguji sampel tersebut di laboratorium. Pengujian secara konvensional ini membutuhkan waktu dan biaya sehingga kualitas dari sebuah sumber daya air tidak dapat dipantau secara *real-time*. Pemantauan terhadap kualitas sumber daya air sangat penting untuk mendapatkan gambaran umum terkait kualitas air di sebuah tempat. Informasi yang didapat terkait kualitas air ini dapat digunakan dalam perencanaan penataan kota.

Kepadatan penduduk di Indonesia berada pada nomor empat di dunia setelah Amerika. Data dari Badan Pusat Statistik pada juni 2016 menunjukkan bahwa Indonesia memiliki 257,9 juta penduduk dimana sebagian besar terkonsentrasi di kota. Saat ini, arsitektur kota di Indonesia lebih mirip sebagai sebuah ‘kampungan besar’ (Evers, 2007). Kota-kota di Indonesia pada biasanya terfragmentasi menjadi wilayah yang bersifat formal dan informal. Penduduk Indonesia yang berada di perkotaan, terutama pendatang, kebanyakan

tinggal di wilayah informal yang lokasinya tersebar. Wilayah informal yang tersebar dan mendominasi kota-kota di Indonesia menjadikan penataan dan pengembangan kota memerlukan perencanaan yang matang. Perencanaan yang kurang matang akan menimbulkan munculnya lebih banyak masalah.

Smart city yang saat ini sedang menjadi tren berusaha untuk menyelesaikan permasalahan yang ada pada perkotaan dengan bantuan teknologi. Dalam pengembangan smart city, teknologi komputer menjadi aspek yang bersifat fundamental. Weiser (1993) mengemukakan bahwa di masa depan, komputer bekerja secara tidak kasat mata untuk mendukung aktivitas manusia dalam sebuah infrastruktur untuk memproduksi data, baik secara otomatis maupun dengan bantuan manusia. Karakteristik dari *smart city* adalah banyaknya jaringan nirkabel yang tersebar wilayah perkotaan serta penggunaan sensor yang terdistribusi untuk memantau arus lalu lintas, cuaca hingga saluran air untuk mendapatkan informasi secara real-time (Murakami, 2014). *Ubiquitous computing* dengan pendekatan *internet of things* merupakan sebuah konsep utama yang membentuk *smart city*.

Internet of things (IoT) adalah konsep dimana beberapa perangkat keras dan perangkat lunak dapat saling terhubung dan berkomunikasi satu dengan yang lainnya melalui sebuah jaringan telekomunikasi. IoT mengacu pada benda sehari-hari yang saling terkoneksi serta memiliki sifat *ubiquitous* (Xia et al, 2012). *Ubiquitous* artinya sebuah perangkat IoT dapat ditemukan di berbagai tempat misalnya di rumah, jalan, sungai, atau bahkan di tengah laut. Perangkat IoT bekerja dengan sendirinya mengikuti instruksi yang sudah diprogram sesuai dengan kegunaannya. Perangkat IoT biasanya hanya memerlukan sedikit campur tangan manusia, dimana manusia hanya bertugas mengawasi. Meskipun konsep IoT adalah sebuah hal yang baru namun sudah cukup banyak platform yang mendukung konsep ini. Perangkat mikrokontroler yang biasanya digunakan dalam bidang robotika dapat dikombinasikan dengan berbagai macam sensor dan modul untuk menghasilkan sebuah perangkat IoT.

Mikrokontroler merupakan sebuah semikonduktor yang dapat diprogram sehingga memiliki fungsi tertentu seperti mengendalikan perangkat lain maupun membaca nilai yang dihasilkan oleh sensor. Mikrokontroler dapat dihubungkan dengan modul yang memberi fungsionalitas tambahan seperti komunikasi melalui gelombang radio sehingga mikrokontroler dapat berkiriman data dengan perangkat lain. Modul wifi yang ada dipasaran dapat dipasangkan dengan mikrokontroler sehingga mikrokontroler ini memiliki konektivitas dengan internet.

Mikrokontroler yang saling berkomunikasi dapat digunakan untuk membuat sebuah jaringan sensor nirkabel. Jaringan sensor nirkabel adalah sebuah jaringan yang terdiri atas titik sensor dan titik sentral dimana sekumpulan titik sensor bekerja secara otomatis untuk melakukan pengambilan data kondisi lingkungan yang kemudian dikirimkan ke titik sentral. Fleksibilitas dari mikrokontroler ini membuat perangkat ini dapat digunakan untuk mengembangkan sebuah perangkat IoT seperti purwarupa sistem monitoring sumber daya air.

Pada purwarupa sistem monitoring sumber daya air yang akan dibuat, mikrokontroler digunakan sebagai perangkat yang dapat membaca kualitas air. Mikrokontroler dihubungkan dengan berbagai macam sensor yang dapat membaca kualitas air seperti pH, tingkat kekeruhan dan sebagainya. Data dari kualitas air yang diambil kemudian dapat dikirimkan secara nirkabel melalui modul pengirim data seperti modul wifi. Data ini kemudian akan dikirimkan ke sebuah perangkat yang dapat mengolah dan menampilkan data tersebut.

Penelitian ini merupakan bagian penelitian unggulan Perguruan Tinggi No. 03/ST-DirDPPM/70/DPPM/ Penelitian Terapan Unggulan Perguruan Tinggi Kemristekdikti/IV/2017 yang berjudul: Pengembangan Masterplan, Desain dan Prototip “Prasarana Data Lingkungan” Berteknologi *Internet of Things* untuk Pemantauan dan Perencanaan Kualitas Lingkungan di Wilayah Perkotaan Informal: Studi Kasus Bantaran Sungai Code di Yogyakarta oleh Dr. Ilya Fadjar Maharika, IAI. sebagai ketua penelitian dan, Ari Sujarwo, S.Kom., MIT (Hons), Medila Kusriyanto, ST., M.Eng. and DR.-Ing. Widodo B.Msc. a. sebagai tim peneliti. Penelitian ini akan dilaksanakan selama tiga tahun yaitu tahun 2017 sampai dengan 2019.

Pada tahun pertama, akan difokuskan pada pengembangan model pendataan bentang air yaitu teknologi terapan untuk memonitor perilaku air dan penggunaannya, menciptakan respon berupa notifikasi dan langkah terprogram tertentu (otomasi) yang terintegrasi dengan pengetahuan lokal sehingga menjadi pengetahuan lokal baru berupa data bentang air (*Waterscape Map*) yang merupakan gambaran *real-time* bagaimana “perilaku air” di kawasan tertentu.

Penelitian tahun pertama ini akan dibagi menjadi beberapa bidang. Salah satu dari bidang tersebut adalah *Information Technology* yang diketuai oleh bapak Ari Sujarwo, S.Kom., MIT (Hons). Bidang tersebut berfokus pada: Identifikasi teknologi sensing dan penstruktural data tentang air (*knowledge management*) dan Pengembangan konsep desain. Penelitian pada bidang *Information Technology* dibagi menjadi 4 sub bagian yang kemudian digunakan sebagai tugas akhir mahasiswa.

1.2 Rumusan Masalah

Berdasarkan latar belakang dapat dirumuskan sebuah rumusan yaitu bagaimana membuat purwarupa sebuah sistem yang dapat digunakan untuk melakukan pemantauan kondisi air secara otomatis dan menyimpan hasilnya pada sebuah basis data?

1.3 Batasan Masalah

Batasan masalah dari penelitian ini adalah:

1. Perangkat hanya sebatas purwarupa.
2. Pengujian kualitas air hanya sebatas pada sensor pH, konduktivitas, kekeruhan, suhu.
3. Sensor aliran air hanya untuk menghitung jumlah air yang digunakan.
4. Perangkat sentral harus selalu aktif.
5. Perangkat sentral hanya berfungsi untuk menerima dan menyimpan data.
6. Perangkat sensor hanya membaca dan mengolah dan mengirimkan data sensor ke perangkat sentral.
7. Perangkat sensor hanya dapat mengirimkan data saat terhubung dengan perangkat sentral.
8. Jarak komunikasi maksimal antar perangkat adalah 200 meter.

1.4 Tujuan Penelitian

Tujuan dari pengerjaan tugas akhir yang berjudul “Purwarupa Sistem Monitoring Sumber Daya Air di Wilayah Perkotaan Informal dengan Arduino pada Studi Kasus Bank Air Kampung Terban Yogyakarta” adalah untuk membuat rancangan dan purwarupa dari sistem monitoring kualitas air yang dapat berjalan secara otomatis.

1.5 Manfaat Penelitian

Penelitian ini memiliki beberapa manfaat yang dapat diambil, antara lain:

1. Mempermudah dalam memantau dan menguji kualitas air secara berkala
2. Sebagai media pembelajaran dalam pengembangan perangkat IoT
3. Sebagai purwarupa pembuatan sistem monitoring kualitas air

1.6 Metodologi Penelitian

Untuk menyelesaikan penelitian ini, beberapa tahapan dilakukan untuk mendapatkan hasil sesuai dengan yang diharapkan. Tahap-tahap yang dilakukan adalah sebagai berikut:

1.6.1 Analisis Kebutuhan

Analisis kebutuhan adalah tahap yang dilakukan untuk mengidentifikasi masalah yang ada dan perangkat lunak serta perangkat keras yang dibutuhkan oleh penulis dalam pengerjaan penelitian ini. Analisis kebutuhan dibagi menjadi dua tahap:

Studi Lapangan

Survei lokasi dilakukan dengan tujuan agar penulis mengetahui kondisi dari lokasi alat sistem monitoring air nantinya diimplementasikan sehingga penulis dapat menentukan perangkat keras dan perangkat lunak yang sesuai dengan kondisi lapangan.

Studi Pustaka

Studi pustaka dilakukan untuk mencari materi dan referensi terkait dengan judul penelitian, perangkat keras dan perangkat lunak yang digunakan oleh penulis. Tujuan dari pencarian materi dan referensi adalah agar penulis lebih memahami tentang topik penelitian dan mempermudah dalam pengerjaan penelitian.

1.6.1 Perancangan Sistem

Pada tahap ini, sistem dirancang agar dapat bekerja sesuai dengan harapan penulis. Perancangan dilakukan dengan diagram untuk mempermudah dalam pembuatan sistem.

1.6.2 Implementasi Sistem

Implementasi dilakukan dengan membuat sistem sesuai dengan rancangan yang telah dibuat sebelumnya. Sistem diimplementasikan dalam beberapa bahasa pemrograman yang disesuaikan dengan perangkat yang digunakan.

1.6.3 Pengujian Sistem

Pengujian sistem dilakukan untuk menentukan apakah sistem berjalan sesuai dengan rancangan yang sudah dibuat. Tahap pengujian juga dilakukan untuk menentukan reliabilitas dari sistem yang dibuat.

1.6.4 Sistematika Penulisan

Penulisan laporan tugas akhir ini dibagi menjadi beberapa bagian. Bagian pertama adalah Pendahuluan yang berisi pengantar permasalahan yang akan dibahas. Pendahuluan menjelaskan tentang gambaran umum terkait objek penelitian. Bagian kedua adalah Kajian

Pustaka yang membahas tentang teori-teori yang mendukung penelitian ini. Metodologi membahas tentang langkah-langkah yang akan dilakukan, pengumpulan data, analisis kebutuhan, dan perancangan untuk membuat sistem. Hasil dan pembahasan memaparkan hasil dari purwarupa perangkat monitoring air yang sudah dibuat. Meliputi inti program dan hasil pengujian dari perangkat. Kesimpulan dan saran berisi kesimpulan dari hasil purwarupa perangkat monitoring air yang sudah dibuat. Saran berisi masukan yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dalam pembuatan perangkat.

BAB II

KAJIAN PUSTAKA

2.1 Air

2.1.1 Air pada wilayah informal

Evers (2007) memaparkan bahwa sebagian besar wilayah perkotaan di Indonesia merupakan wilayah perkotaan informal. Secara harafiah kata informal berarti tidak resmi. Kata wilayah dapat didefinisikan sebagai tempat sekeliling atau yang termasuk dalam lingkungan suatu tempat. Wilayah perkotaan informal dapat diartikan sebagai wilayah perkotaan yang tidak terjangkau oleh aturan resmi yang ditetapkan oleh pemerintah. Menurut pengertian tersebut wilayah perkotaan informal dapat diartikan sebagai wilayah perkotaan yang tidak teratur atau kumuh.

Sebagian besar penduduk Indonesia tinggal di pemukiman yang berada di wilayah perkotaan informal. Penduduk di wilayah ini biasanya memanfaatkan air sungai atau sumber mata air untuk keperluan sehari-hari seperti mandi dan mencuci. Beberapa pemukiman bahkan memanfaatkan sumber air ini sebagai sumber air minum. Pertambahan jumlah penduduk artinya bertambahnya kebutuhan hidup, salah satunya adalah air. Penduduk di wilayah perkotaan informal pada umumnya memiliki kesadaran lingkungan yang masih kurang. Penduduk yang tinggal dekat sungai biasanya membuang sampah dan limbah rumah tangga ke sungai. Sumber air bersih sangat mudah tercemar oleh zat-zat yang mengakibatkan perubahan kualitas air seperti sabun dan sampah-sampah organik. Perubahan kualitas ini mengakibatkan air menjadi tidak layak untuk dikonsumsi. Saat ini masalah utama yang dihadapi oleh sumber daya air meliputi kuantitas air yang sudah tidak mampu memenuhi kebutuhan yang terus meningkat dan kualitas air untuk keperluan domestik semakin menurun sehingga kesadaran terkait penghematan dan pelestarian sumber daya air harus ditanamkan di masyarakat (Effendi, 2003). Berdasarkan pemaparan tersebut, Pemantauan kualitas air perlu dilakukan untuk mengetahui perubahan kualitas air sehingga dapat dilakukan tindakan represif.

2.1.2 Pentingnya pemantauan kualitas air

Kualitas air dapat berubah sewaktu-waktu karena faktor lingkungan seperti bencana alam atau juga disebabkan oleh faktor manusia seperti limbah rumah tangga. Pemantauan

terhadap kualitas sumber air dapat memberikan berbagai informasi terkait dengan keadaan masyarakat yang tinggal disekitar sumber air.

Mason (1993) memaparkan bahwa pemantauan terhadap kualitas sebuah sumber air memiliki tiga tujuan utama. Tujuan pertama tentunya untuk mengetahui gambaran kualitas air pada suatu tempat secara umum. Tujuan kedua adalah untuk menentukan kriteria kualitas air berdasarkan variabel-variabel ekologi perairan. Kriteria kualitas air dapat ditentukan dengan mengetahui hubungan sebab akibat antara perubahan variabel berdasarkan pada parameter fisika dan kimia. Nilai parameter fisika dan kimia ini kemudian dijadikan patokan dalam pembentukan kriteria kualitas air. Tujuan ketiga adalah untuk mendeteksi dan mengukur pengaruh yang ditimbulkan oleh sebuah zat pencemar terhadap kualitas lingkungan di sekitarnya. Pemantauan kualitas air juga dilakukan untuk mengetahui perbaikan kualitas lingkungan setelah zat pencemar dihilangkan.

Fardiaz (1992) menjelaskan bahwa tingkat pencemaran air dapat ditentukan melalui beberapa sifat-sifat air. Ada beberapa sifat yang pada umumnya dapat digunakan untuk menentukan tingkat pencemaran yaitu warna, konduktivitas, tingkat kekeruhan, nilai pH dan suhu. Air yang baik tidak berwarna dan tidak keruh. Air yang berwarna dan keruh menandakan terdapat ada padatan atau zat kimia yang terlarut. Nilai pH digunakan untuk menentukan derajat keasaman air, semakin tinggi nilainya maka air semakin asam air yang baik memiliki nilai PH antara 6,5 sampai 8,5. Air yang baik memiliki tingkat konduktivitas yang kecil. Konduktivitas yang tinggi menunjukkan tingkat kesadahan atau kadar garam yang tinggi. Suhu air juga dapat digunakan untuk menentukan adanya zat pencemar. Zat kimia atau aktivitas bakteri dapat meningkatkan suhu air. Menurut PERMENKES RI Nomor 416 Tahun 1990, suhu air yang baik memiliki batas maksimal kurang lebih 3 derajat celsius dari suhu lingkungan.

2.2 Peran teknologi terkait pemantauan kualitas air

2.2.1 Sistem benam dan mikrokontroler

Sistem benam atau sering disebut *embedded system* menjadi kunci dalam kemajuan teknologi saat ini. Ponsel pintar dan kamera digital merupakan sebagian sistem benam yang sering ditemui dalam kehidupan sehari-hari. Corporaal (2006) mendefinisikan sistem benam sebagai sebuah sistem yang memproses informasi dari sensor dan mengendalikan aktuator sesuai dengan fungsi yang dibenamkan. Sistem benam biasanya diimplementasikan dalam bentuk mikrokontroler.

Mikrokontroler pada dasarnya merupakan sebuah perangkat pemroses data dengan perangkat penyimpanan dan memori yang dibenamkan di dalamnya. Mikrokontroler merupakan sebuah pemroses data yang dapat diimplementasikan pada berbagai hal (Singh, 2008). Mikrokontroler menjadi otak dari berbagai perangkat elektronik saat ini. Mikrokontroler memiliki berbagai macam jenis dengan ukuran penyimpanan dan kemampuan pemrosesan yang bervariasi. Mikrokontroler yang ada pada dasarnya berbasis pada dua model arsitektur yaitu arsitektur CISC dan RISC. Kedua arsitektur ini merepresentasikan cara bertukar data antara pemroses data dan memori.

Arsitektur RISC (*Reduced Instruction Set Computing*) dikembangkan oleh Berkeley, IBM dan Stanford sekitar tahun 1960. RISC disebut juga arsitektur Harvard dan menjadi penerus arsitektur CISC. RISC menjadi arsitektur yang banyak digunakan oleh perangkat mikrokontroler saat ini. Berbeda dengan CISC yang menggunakan satu jalur, arsitektur ini menggunakan dua jalur berbeda untuk mengakses memori dan data sehingga pemrosesan data lebih cepat. Arsitektur RISC menggunakan set instruksi yang lebih sederhana, berbeda dengan CISC yang menggunakan set instruksi yang banyak dan memiliki fungsi spesifik (Masood, 2011). Arsitektur RISC ini membuat mikrokontroler yang dibuat dengan arsitektur ini lebih mudah dalam proses pengembangannya.

2.2.2 Pemanfaatan sensor pada sistem benam

Sebuah sistem benam dapat mengambil data melalui suatu sensor. Sensor ini digunakan oleh sistem untuk mengambil data dari lingkungan di sekitarnya. Sensor yang digunakan memiliki fungsi yang spesifik sesuai dengan kegunaannya. Sensor suhu dapat memberikan informasi besaran suhu udara, benda padat atau suatu zat cair. Sensor PH dapat digunakan untuk mengetahui derajat keasaman dari sebuah zat. Setiap sensor menghasilkan sebuah keluaran berbeda yang dapat diolah oleh sistem.

Menurut Budiarmo (2011) sensor adalah peralatan yang digunakan untuk merubah suatu besaran fisik menjadi besaran listrik sehingga dapat dianalisis dengan rangkaian listrik tertentu. Sensor sentuhan pada layar ponsel pintar merupakan salah satu jenis sensor yang sering ditemui. Sensor ini mengubah sentuhan pengguna yang merupakan besaran fisik menjadi besaran listrik yang menjadi data masukan pada sistem.

Seperti yang dipaparkan sebelumnya, besaran listrik hanya dapat dianalisis setelah melalui rangkaian tertentu. Dalam penggunaannya, masukan dan keluaran ini membutuhkan sebuah rangkaian perangkat akuisisi data dalam bentuk konverter analog-digital maupun digital-analog (Febridiani dan Arif, 2012). Mikrokontroler yang merupakan sebuah sistem

benam memiliki perangkat tersebut di dalamnya. Konverter ini mengubah besaran listrik yang merupakan data analog menjadi data digital maupun sebaliknya untuk selanjutnya diproses oleh pemroses data pada sistem.

2.2.3 Teknologi jaringan sensor nirkabel

Menurut Ardiyanto dan Sumiharto (2012), jaringan sensor nirkabel adalah sebuah teknologi nirkabel yang terdiri dari kumpulan titik sensor yang tersebar di suatu area tertentu. Teknologi ini sudah berkembang sejak lama, namun perkembangannya cukup lambat. Namun berdasarkan hasil riset dan pengembangannya, teknologi sensor nirkabel menjadi solusi yang layak untuk diimplementasikan pada sistem yang membutuhkan komunikasi nirkabel (Ferdoush dan Li, 2014).

Teknologi ini memungkinkan suatu sistem untuk saling berkomunikasi melalui gelombang radio. Teknologi sensor nirkabel dapat diaplikasikan pada sebuah sistem benam. Standar yang ditetapkan oleh IEEE, yaitu 802.15.4 RF mendukung hal ini. Standar 802.15.4 berisi tentang spesifikasi protokol dan konsumsi daya dari perangkat komunikasi yang menggunakan teknologi ini. Standar ini menetapkan agar perangkat komunikasi menggunakan konsumsi daya yang rendah sehingga sistem yang menggunakan perangkat ini tidak memerlukan sumber listrik yang besar. Protokol yang digunakan juga memiliki reliabilitas data yang tinggi sehingga kerusakan data saat proses pengiriman dapat diatasi. Proses penambahan titik sensor juga mudah karena perangkat komunikasi secara otomatis akan bergabung dengan jaringan yang sudah ditentukan.

2.2.4 Konsep *smart city* dan IoT

Perkembangan teknologi yang pesat sangat berpengaruh terhadap manusia. Keberadaan internet merupakan salah satu pemicu pesatnya perkembangan teknologi. Internet mengubah cara hidup manusia secara keseluruhan. Teknologi memberikan banyak kemudahan, serta sebagai cara baru dalam melakukan aktivitas manusia (Ngafifi, 2014). Aktivitas yang sebelumnya hanya dilakukan secara luring seperti membaca berita sekarang dapat dilakukan secara daring melalui Internet. Internet bisa dikatakan sudah menjadi bagian dari manusia. Internet yang ada saat ini tidak hanya terhubung dengan komputer personal. Keberadaan ponsel pintar semakin membuat internet semakin mudah diakses. Internet bahkan terhubung dengan perangkat seperti lampu dan pemanggang roti. Hal ini merupakan sebuah konsep yang dinamakan IoT atau *Internet of Things* yang merupakan inti dari *smart city*.

Istilah *smart city* atau kota pintar memiliki artian yang beragam jika dilihat dari berbagai perspektif. Holland (2008) memandang *smart city* sebagai sebuah fenomena “*Urban Labeling*” karena label smart city merupakan konsep yang bias serta arti dan penggunaannya tidak konsisten. Namun apabila dilihat dari perspektif teknologi, *smart city* dapat diartikan sebagai sebuah kota yang didalamnya terdapat banyak jaringan nirkabel yang menghubungkan berbagai perangkat pintar di penjuru kota. Teknologi merupakan kunci utama untuk menjadi sebuah smart city karena penggunaan teknologi informasi dapat dimanfaatkan untuk mengubah cara hidup dan bekerja manusia menjadi lebih efektif (Holland, 2008). Berdasarkan pemaparan tersebut dapat dikatakan bahwa IoT merupakan konsep dasar dari sebuah *smart city*. Teknologi yang bersifat *mobile* dan *ubiquitous* akan menjadi sangat penting (Nam dan Pardo, 2011).

IoT pertama kali diperkenalkan sekitar tahun 1990. Pada saat itu yang menjadi fokus utama dari IoT adalah pemanfaatan teknologi informasi pada infrastruktur modern di perkotaan. Fokus dari penelitian IoT bergeser ketika Institut California melakukan riset tentang bagaimana sebuah kota dapat mengimplementasikan teknologi informasi di setiap aspek yang ada (Alawadhi et al., 2012). Weiser (1993) mengemukakan bahwa nantinya teknologi akan membantu manusia di berbagai aspek kehidupan. Semua objek dalam kehidupan sehari-hari akan saling terhubung dengan internet dan saling berinteraksi. IoT membuka banyak peluang bagi pengembangan produk yang dapat meningkatkan kualitas hidup (Xia et al, 2012). Perangkat IoT biasanya memiliki fungsi yang spesifik. Contohnya adalah penggunaan sensor pada kendaraan yang dapat mengatur jalannya kendaraan di tengah padatnya lalu lintas. Teknologi *energy sensing* yang sedang dikembangkan dapat digunakan untuk memantau penggunaan listrik secara akurat sehingga membantu dalam memantau penggunaan listrik.

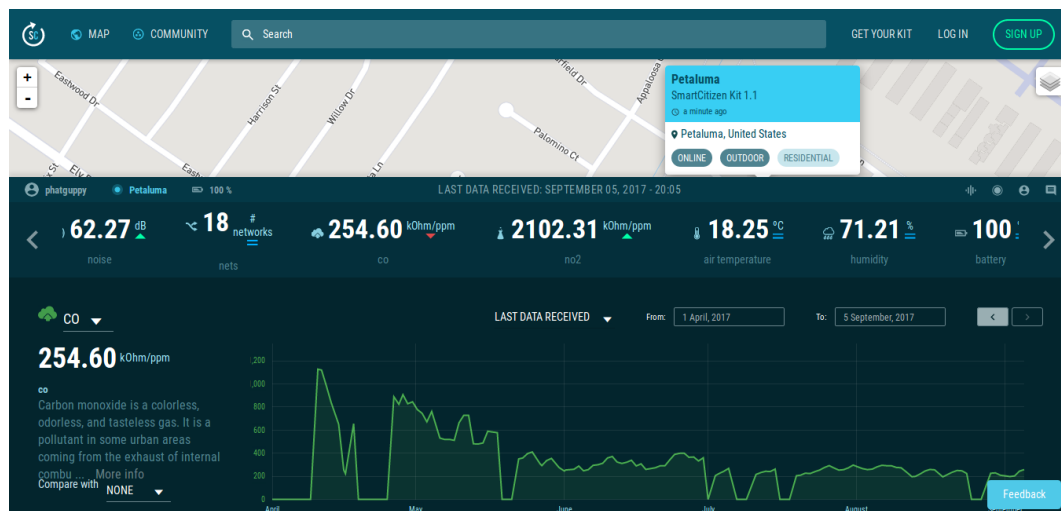
2.2.5 Contoh implementasi IoT pada kehidupan nyata

IoT mulai diaplikasikan di berbagai belahan dunia. Berbagai bidang sudah mulai mengaplikasikan konsep ini. Salah satu contohnya adalah *Smart Citizen Smart Citizen* merupakan sebuah proyek yang dikembangkan oleh Fab Lab Barcelona. Tujuan dari proyek ini adalah untuk mengambil data terkait tingkat pencahayaan, kebisingan suara serta polusi, kelembaban dan suhu udara di suatu tempat secara *real-time*.



Gambar 2.1 *Smart Citizen Kit* (sumber: seedstudio.com)

Proyek ini mengandalkan sukarelawan untuk membawa sebuah perangkat khusus. Perangkat yang disebut dengan *Smart Citizen Kit* ini memiliki berbagai macam sensor yang tertanam di dalamnya. Perangkat ini mengirimkan data yang diambil oleh sensor melalui jaringan internet ke sebuah server pusat untuk kemudian diolah. Data yang sudah diolah selanjutnya ditampilkan ke sebuah website yang dapat diakses oleh setiap orang seperti pada Gambar 2.2.



Gambar 2.2 Tampilan situs *Smart Citizen Kit*

IoT tak hanya diimplementasikan untuk melakukan pemantauan kualitas udara di perkotaan. Di Indonesia, IoT juga diimplementasikan pada bidang perikanan melalui sebuah produk bernama *eFishery*. *eFishery* merupakan sebuah sistem *smart-feeder* yang terdiri atas sebuah perangkat pakan ikan dan aplikasi ponsel pintar. Pemilik alat ini dapat mengatur jumlah pakan yang diberikan serta jadwal pemberian pakan dari setiap alat. Sistem ini juga

akan mencatat setiap data pakan dan menyimpan data tersebut ke internet sehingga pemilik dapat melakukan pemantauan setiap saat.



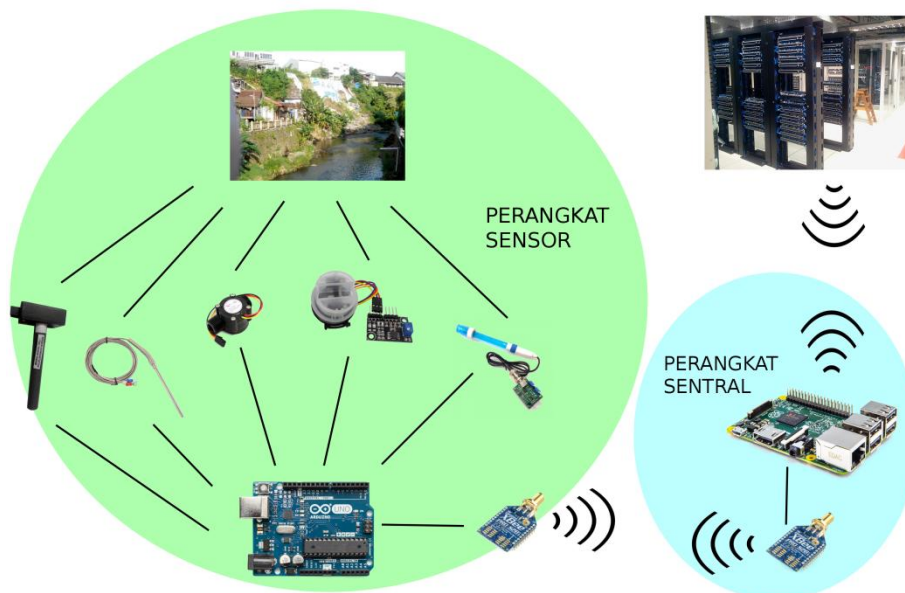
Gambar 2.3 Alat pakan eFishery (Sumber: efishery.com)

Terkait dengan pemantauan terhadap kualitas air, Konsep IoT dapat diaplikasikan dengan membuat sebuah perangkat yang dapat membaca sifat-sifat air melalui sensor. Teknologi jaringan sensor nirkabel nantinya dapat diimplementasikan pada perangkat ini sebagai media komunikasi perangkat dalam mengirimkan data yang sudah diambil untuk kemudian diolah.

BAB III METODOLOGI

3.1 Gambaran umum sistem

Gambar 3.1 menggambarkan bentuk keseluruhan sistem yang akan dibuat. Sistem nantinya terdiri atas dua perangkat yaitu perangkat sensor dan perangkat sentral. Kedua perangkat ini akan saling berkomunikasi melalui jaringan nirkabel. Perangkat akan dipasang pada salah satu dari sembilan mata air yang ada di Terban, Yogyakarta. Perangkat sentral akan diletakkan di sebuah titik yang ada diantara sembilan mata air. Perangkat sensor akan diletakkan di salah satu mata air.



Gambar 3.1 Gambaran sistem

Perangkat sensor terdiri atas kontroler, perangkat komunikasi nirkabel dan sensor. Sensor yang ada pada perangkat ini akan mengambil data secara langsung dari mata air. Data dari mata air yang diambil sensor, diolah oleh kontroler untuk selanjutnya dikirim ke perangkat sentral melalui jaringan nirkabel.

Perangkat sentral terdiri atas sebuah komputer dan perangkat komunikasi nirkabel. Data yang dikirimkan oleh perangkat sensor diterima dan diproses oleh perangkat sentral. Perangkat sentral berfungsi menyimpan data dari perangkat sensor ke dalam basis data untuk

sementara. Basis data pada perangkat sentral kemudian akan direplikasi ke server pusat untuk ditampilkan ke sebuah situs web.



Gambar 3.2 Titik penempatan perangkat



Gambar 3.3 Bank AirKami

Penelitian ini mencakup pembuatan purwarupa perangkat sensor dan perangkat sentral. Kedua perangkat nantinya akan ditempatkan di salah satu mata air yang terdapat di Terban, Yogyakarta seperti yang ditunjukkan pada Gambar 3.2. Lokasi tersebut merupakan salah satu lokasi mata air di Terban yang digunakan oleh warga. Pada lokasi tersebut terdapat

sebuah bank air (Gambar 3.3) yang menjadi sumber air minum masyarakat Terban. Perangkat sensor akan dihubungkan dengan pompa yang terdapat pada bank air. Perangkat sentral akan ditempatkan pada lokasi yang sama namun tidak terhubung langsung dengan perangkat sensor.

3.2 Analisis kebutuhan

Analisis kebutuhan meliputi penentuan kebutuhan yang nantinya harus dipenuhi dalam implementasi sistem monitoring air.

3.2.1 Kebutuhan masukan

Kebutuhan masukan dari sistem monitoring air ini antara lain:

- a. Data konfigurasi
- b. Data sensor

3.2.2 Kebutuhan keluaran

Kebutuhan keluaran dari sistem monitoring air ini antara lain:

- a. Informasi nilai sensor
- b. Informasi perangkat sensor
- c. Informasi log perangkat sentral

3.2.3 Kebutuhan proses

Kebutuhan proses dari sistem monitoring air ini dibagi menjadi dua bagian, yaitu:

- a. Kebutuhan proses Perangkat sentral
 - 1. Proses *parsing* konfigurasi perangkat sentral
 - 2. Proses penerimaan data sensor
 - 3. Proses *parsing* data sensor
 - 4. Proses penyimpanan data sensor
 - 5. Proses penghapusan data sensor lama
- b. Kebutuhan masukan Perangkat sensor
 - 1. Proses *parsing* konfigurasi perangkat sensor
 - 2. Proses pengambilan data sensor
 - 3. Proses konstruksi data sensor
 - 4. Proses pengiriman data sensor

3.2.4 Kebutuhan perangkat keras

Kebutuhan perangkat keras dari sistem monitoring air ini antara lain:

a. Raspberry Pi



Gambar 3.4 Raspberry Pi

Raspberry Pi digunakan sebagai perangkat sentral pada sistem ini. Perangkat sentral berperan sebagai penerima data yang dikirimkan oleh perangkat sensor. Data yang diterima akan diolah dan disimpan pada perangkat ini.

Raspberry Pi dipilih karena memiliki ukuran kecil namun memiliki fungsi sama seperti sebuah komputer *desktop*. Raspberry Pi juga termasuk perangkat yang hemat daya karena hanya membutuhkan daya kurang dari 5 watt (sumber). Raspberry Pi berbasis pada sistem operasi Linux Debian versi ARM yang memiliki *software repository* sendiri. *Repository* yang digunakan memiliki berbagai aplikasi yang sama dengan linux versi desktop. Raspberry Pi memiliki spesifikasi:

1. SoC : Broadcom BCM2837
2. CPU : 4x ARM Cortex-A53, 1.2GHz
3. GPU : Broadcom VideoCore IV
4. RAM : 1GB LPDDR2 (900 MHz)
5. Koneksi : 10/100 Ethernet, 2.4GHz 802.11n *wireless*
6. *Bluetooth* : Bluetooth 4.1 Classic
7. Penyimpanan : microSD
8. GPIO : 40-pin
9. *Ports* : HDMI, 3.5mm *analogue audio-video jack*, 4x USB 2.0, Ethernet, *Camera Serial Interface (CSI), Display Serial Interface (DSI)*

b. Arduino

Arduino merupakan sebuah papan kontroler yang digunakan pada perangkat sensor untuk membaca data dari sensor. Perangkat sensor nantinya berhubungan langsung dengan

alat-alat sensor yang ada di mata air. Perangkat ini membaca data yang merupakan besaran listrik dari alat sensor untuk selanjutnya diubah menjadi satuan tertentu. Perangkat kemudian mengirimkan data tersebut ke perangkat sentral.



Gambar 3.5 Arduino UNO

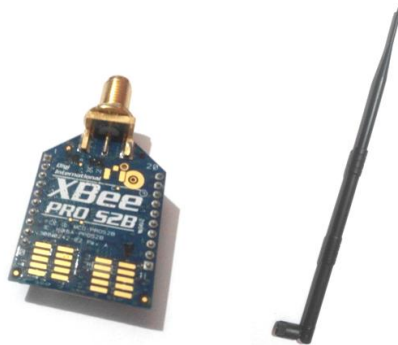
Arduino memiliki banyak varian namun secara umum fungsinya sama. Arduino sangat mudah untuk diprogram, serta memiliki banyak modul dan sensor yang memang khusus dibuat untuk Arduino. Arduino memiliki berbagai varian, namun yang digunakan pada penelitian ini adalah varian UNO R3. Arduino UNO R3 memiliki spesifikasi sebagai berikut:

- | | |
|------------------------|--------------|
| 1. Mikrokontroller | : ATmega328P |
| 2. Tegangan operasi | : 5V |
| 3. Tegangan input | : 6V - 20V |
| 4. Pin Digital | : 14, 6 PWM |
| 5. Pin Analog | : 6 |
| 6. Arus DC per pin I/O | : 20 mA |
| 7. Arus DC pin 3.3V | : 50 mA |
| 8. Memori <i>Flash</i> | : 32 KB |
| 9. SRAM | : 2 KB |
| 10. EEPROM | : 1 KB |
| 11. <i>Clock speed</i> | : 16 Mhz |

c. XBee PRO dan antena 9dBi

XBee merupakan sebuah perangkat yang digunakan untuk komunikasi perangkat. Ada tiga jenis modus yang digunakan pada XBEE yaitu modus *coordinator*, *router* dan *end-device*. Dalam sebuah jaringan XBee, diperlukan adanya sebuah *coordinator* yang mengatur komunikasi. *Coordinator* mengatur PAN ID yang merupakan identitas dari sebuah jaringan

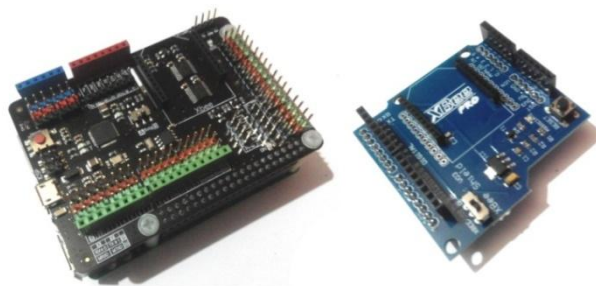
XBee. Setiap Xbee hanya dapat berkomunikasi dengan XBee yang memiliki jaringan (PAN ID) yang sama. Modus *router* dan *end-device* memiliki fungsi yang hampir sama. Kedua XBee dengan modus *router* dan *end-device* dapat berkomunikasi dengan *coordinator* atau XBee lain di jaringan yang sama. Kelebihan modus *router* adalah XBee dengan modus ini dapat memperluas jangkauan sinyal seperti sebuah *repeater* pada jaringan WiFi sehingga perangkat *end-device* yang dapat menjangkau *coordinator* dapat tergabung dalam jaringan melalui XBee dengan modus *router*.



Gambar 3.6 XBEE PRO dan antenna 9dBi

XBee memiliki berbagai varian, namun XBee seri PRO memiliki jangkauan yang paling jauh. XBee dapat digunakan pada Raspberry pi maupun Arduino. Pada sistem ini XBee digunakan untuk mengirimkan data dari perangkat sensor ke perangkat sentral. Antena digunakan untuk memperkuat sinyal yang dipancarkan.

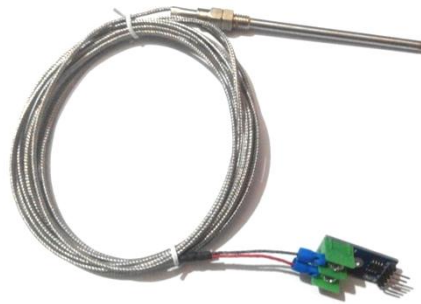
d. XBee Shield



Gambar 3.7 XBee Shield

XBee *Shield* adalah sebuah modul tambahan yang digunakan untuk menghubungkan XBee dengan Raspberry Pi dan Arduino. *Shield* dapat dihubungkan dengan Arduino dan Raspberry Pi melalui pin GPIO.

e. Termokopel dan modul MAX6675



Gambar 3.8 Termokopel dengan modul MAX6675

Termokopel adalah salah satu jenis sensor suhu. Termokopel memiliki berbagai kelebihan jika dibandingkan dengan sensor suhu lainnya, yaitu lebih tahan terhadap lingkungan, memiliki rentang suhu yang luas, respon yang baik terhadap perubahan suhu serta banyak terdapat di pasaran. Modul MAX6675 digunakan untuk memperkuat keluaran yang dihasilkan oleh termokopel sehingga dapat dibaca oleh Arduino.

f. Sensor keasaman



Gambar 3.9 Sensor keasaman dan *amplifier*

Sensor keasaman digunakan untuk mengetahui derajat keasaman atau PH dari air. Sensor PH memiliki *probe* yang dimasukkan ke dalam air dan modul *amplifier* yang berguna untuk memperkuat sinyal yang dikeluarkan oleh sensor sehingga dapat diterima oleh Arduino. *Amplifier* digunakan untuk mengubah output dari sensor dari milivolt menjadi volt sehingga dapat dibaca oleh Arduino. Sensor ini menggunakan persamaan (3.1)

$$pH = 3.5Voltage * 0.004882813 \quad (3.1)$$

g. Sensor aliran air



Gambar 3.10 Sensor aliran air

Sensor ini digunakan untuk mengetahui debit dari aliran air. Pada sistem ini sensor digunakan untuk mengetahui debit air yang dihasilkan oleh sumber mata air. Sensor aliran air yang dapat dihubungkan dengan perangkat mikrokontroler memiliki berbagai macam jenis menurut prinsip kerjanya. Beberapa diantaranya adalah sensor aliran berbasis elektromagnetik, ultrasonik dan turbin. Pada penelitian ini digunakan sensor jenis elektromagnetik karena kemudahannya dalam penggunaan. Sensor ini menggunakan persamaan (3.2) dengan x adalah nilai input analog dari sensor.

$$Flow = \frac{1000 / (time\ since\ last\ read * pulse)}{calibration} \quad (3.2)$$

h. Sensor konduktivitas



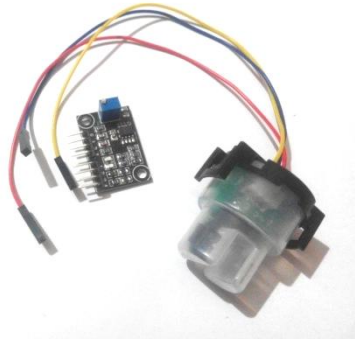
Gambar 3.11 Sensor konduktivitas

Sensor ini digunakan untuk mengetahui konduktivitas air. Sensor ini menggunakan dua batang besi yang berjajar seara paralel. Jarak antara kedua besi kurang lebih 1 cm dengan panjang besi 5.5 cm. Sensor ini akan menghitung tingkat konduktivitas dengan membaca nilai tegangan yang dihantarkan dari satu batang besi ke batang besi lainnya. Nilai

konduktivitas ditentukan melalui persamaan (3.4) dengan x adalah nilai input analog dari sensor. Nilai konduktivitas yang dihasilkan dalam satuan μS

$$Conductivity = 0.518x + 222.4 \quad (3.3)$$

i. Sensor turbiditas

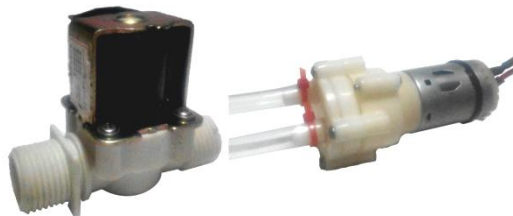


Gambar 3.12 Sensor *turbidity*

Sensor ini digunakan untuk mengetahui turbiditas atau tingkat kekeruhan air dengan menggunakan cahaya yang dipancarkan kemudian diterima kembali oleh sensor. Sensor kemudian akan memperhitungkan tingkat intensitas cahaya yang diterima. Ketika air menjadi keruh, maka tingkat intensitas cahaya semakin kecil. Perhitungan turbiditas pada sensor ini dilakukan dengan persamaan (3.4) yang menghasilkan tingkat kekeruhan dalam persen dimana air yang sangat keruh memiliki nilai 100% dan air jernih 0%.

$$Turbidity = 100 - \left(\frac{Voltage}{calibrationVoltage} \right) * 100 \quad (3.4)$$

j. Katup solenoid dan Pompa DC



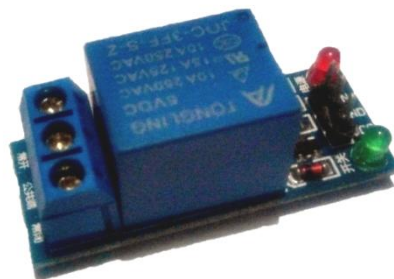
Gambar 3.13 Katup solenoid (kiri) dan Pompa DC (kanan)

Katup Solenoid digunakan untuk sebagai katup yang mengontrol air yang masuk ke dalam penampung sampel. Katup Solenoid memiliki prinsip kerja yang sederhana. Sebuah gulungan solenoid terdapat di dalam perangkat. Pada bagian tengah solenoid terdapat pegas

dan batang besi yang terhubung dengan sebuah katup, fungsi pegas adalah mendorong batang besi agar katup tetap menutup. Saat listrik mengalir melalui perangkat, solenoid akan menghasilkan gaya magnet yang menarik batang besi dan membuka katup sehingga air dapat mengalir.

Pompa DC digunakan untuk membuang sampel yang ada di dalam penampung. Sebuah motor kecil terdapat di bagian ujung pompa. Motor yang terdapat pada pompa DC ini menghasilkan putaran yang cukup untuk menyedot air dari penampung sehingga air dapat dibuang.

k. Relai



Gambar 3.14 Relai

Relai merupakan sebuah sakelar yang bekerja dengan memanfaatkan prinsip elektromagnetik. Pada relai, terdapat sebuah kotak yang berisi seperangkat sakelar yang terhubung dengan pin *trigger*. Pada kotak ini terdapat dua buah titik kontak (NO dan NC), solenoid dan batang besi yang terhubung pada suplai daya dan titik kontak NC (*Normally Closed*). Saat relai aktif, daya magnet solenoid akan menarik batang besi sehingga bersentuhan dengan titik kontak NO (*Normally Open*).

Ada dua jenis relai berdasarkan *trigger* yang digunakan, relai *Low-trigger* dan *High-trigger*. Relai *Low-trigger* akan aktif saat pin *trigger* memiliki tegangan rendah dan sebaliknya pada relai *High-trigger*. Tegangan yang diperlukan berbeda-beda tergantung spesifikasi relai. Pada penelitian ini digunakan relai *Low-trigger*.

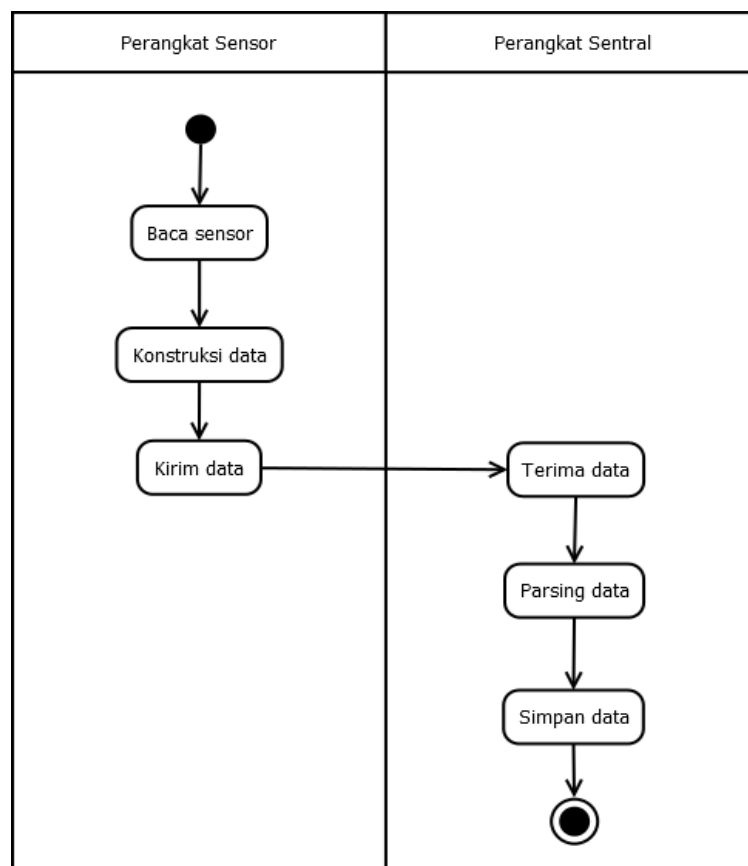
3.3 Perancangan sistem

Perancangan sistem adalah tahap dimana hasil analisis yang telah dilakukan dituangkan dalam bentuk tabel dan diagram yang menggambarkan gambaran umum dari cara kerja sistem yang akan dibuat.

3.3.1 Metode perancangan

Perancangan sistem dilakukan dengan metode perancangan linear atau biasa disebut dengan metode perancangan *waterfall*. Metode ini merupakan metode yang umum digunakan. Metode *waterfall* memiliki urutan yang harus dipenuhi sebelum berlanjut ke tahap berikutnya. Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. (Pressman, Roger S. 2005).

3.3.2 Diagram aktivitas



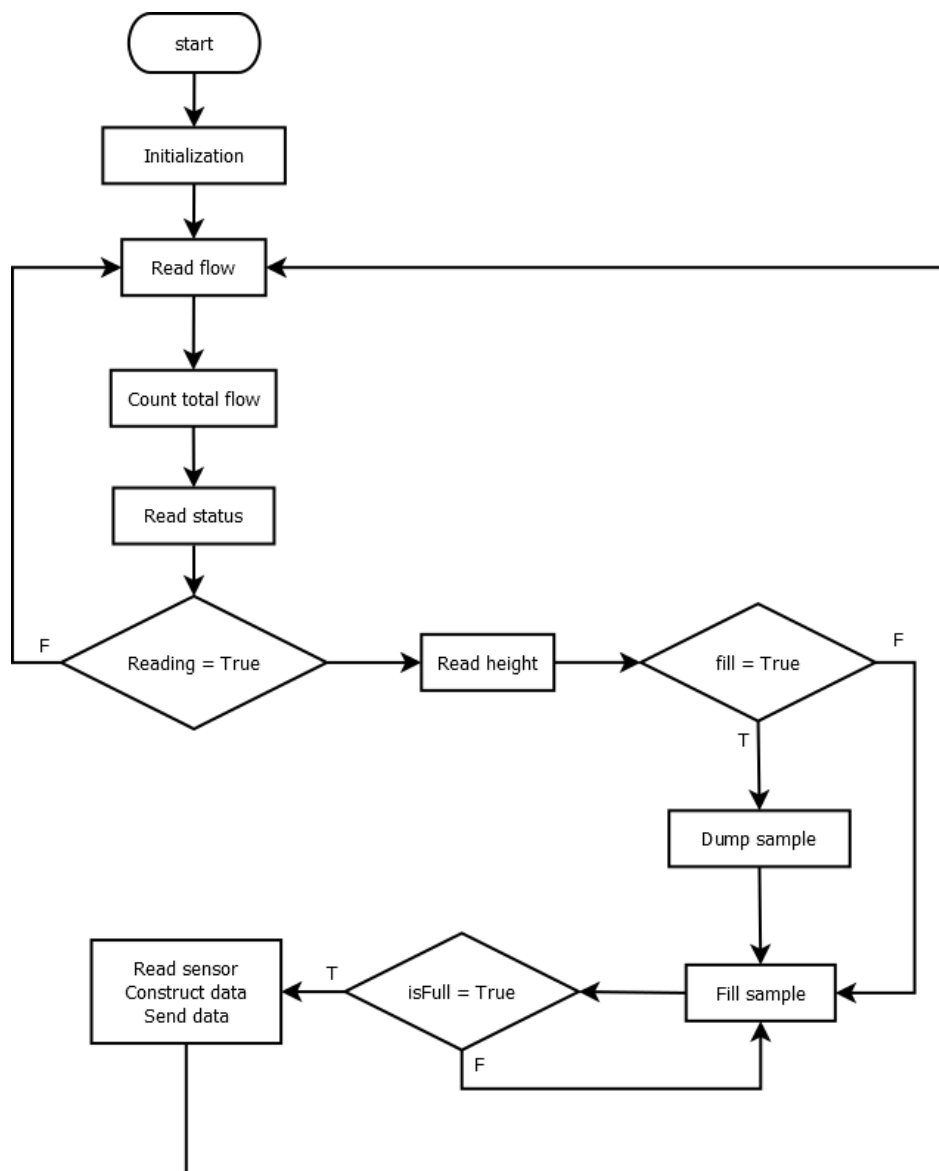
Gambar 3.15 Aktivitas sistem

Gambar 3.15 merupakan diagram aktivitas yang menggambarkan proses kerja dari purwarupa sistem yang akan dibuat. Proses pertama yang dijalankan adalah proses pembacaan sensor yang dilakukan oleh perangkat sensor. Hasil bacaan sensor selanjutnya dikonstruksi menjadi bentuk *string* yang selanjutnya dikirimkan ke perangkat sentral. Alur perangkat sensor secara lengkap dapat dilihat pada Gambar 3.16.

Perangkat sentral akan menerima data yang dikirimkan oleh perangkat sensor. Proses penerimaan data dilakukan oleh *thread* serialWatcher. Alur penerimaan data ditunjukkan pada Gambar 3.18. Data yang berhasil diterima selanjutnya di-*parsing* dan disimpan pada basis data. Proses tersebut dilakukan oleh *thread* worker. Alur *thread* worker dapat dilihat pada Gambar 3.19.

3.3.3 Perancangan diagram alir

a. Diagram alir perangkat sensor

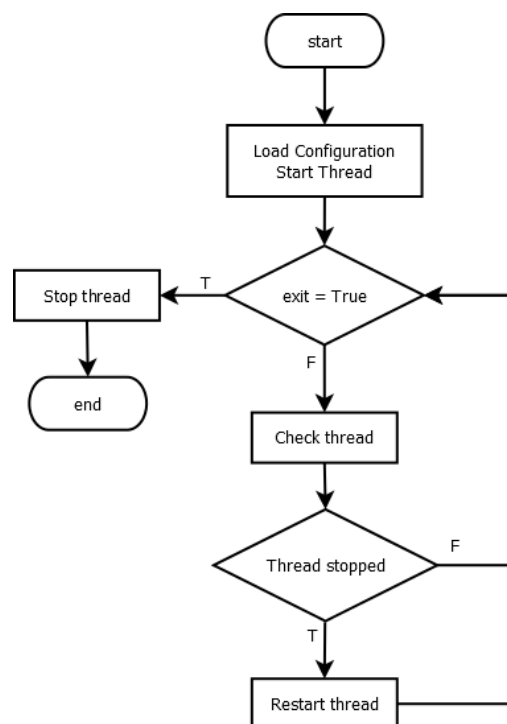


Gambar 3.16 Diagram alir perangkat sensor

Gambar 3.16 menunjukkan alur dari perangkat sensor. Saat dijalankan, perangkat sensor akan melakukan inisialisasi terhadap pin yang ada pada Arduino. Kemudian perangkat

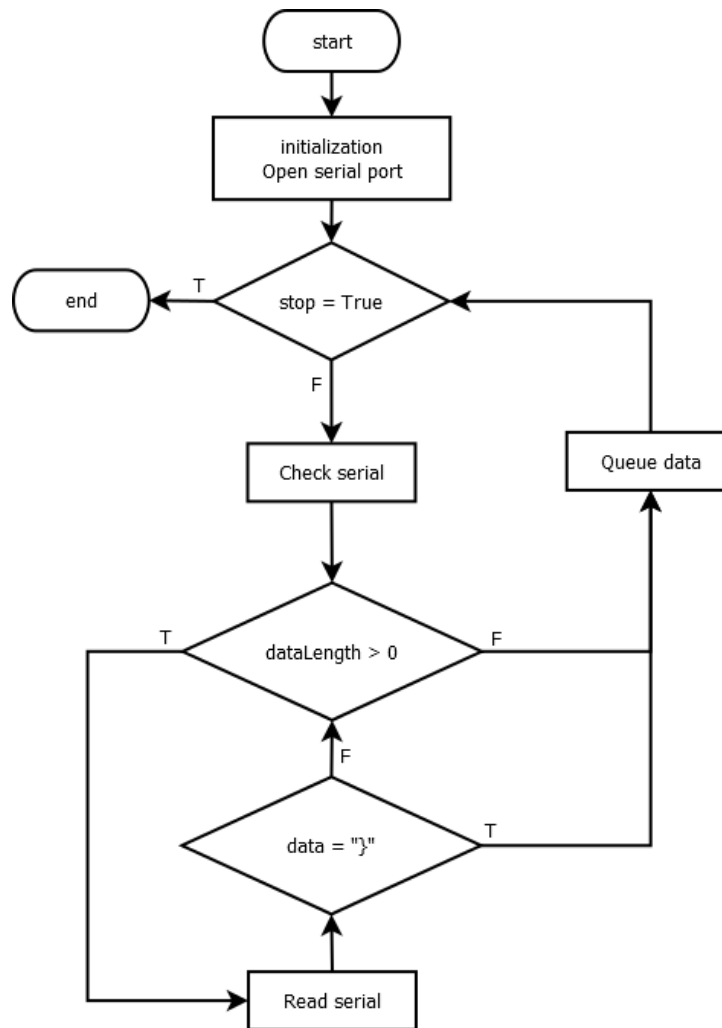
akan terus membaca aliran air dan menghitung jumlah air yang keluar berdasarkan nilai rata-rata aliran air yang dibaca sensor. Perangkat selanjutnya akan mengecek kondisi apakah sensor akan membaca sifat air. Jika tidak, perangkat akan kembali mengecek aliran air. Jika ya, perangkat akan mulai mengambil sampel data pada interval tertentu sampai jumlah sampel yang diambil terpenuhi. Selanjutnya perangkat akan menggabungkan data yang sudah dibaca dan mengirimkannya ke perangkat sentral. Proses akan terus berulang hingga perangkat dimatikan.

b. Diagram alir perangkat sentral



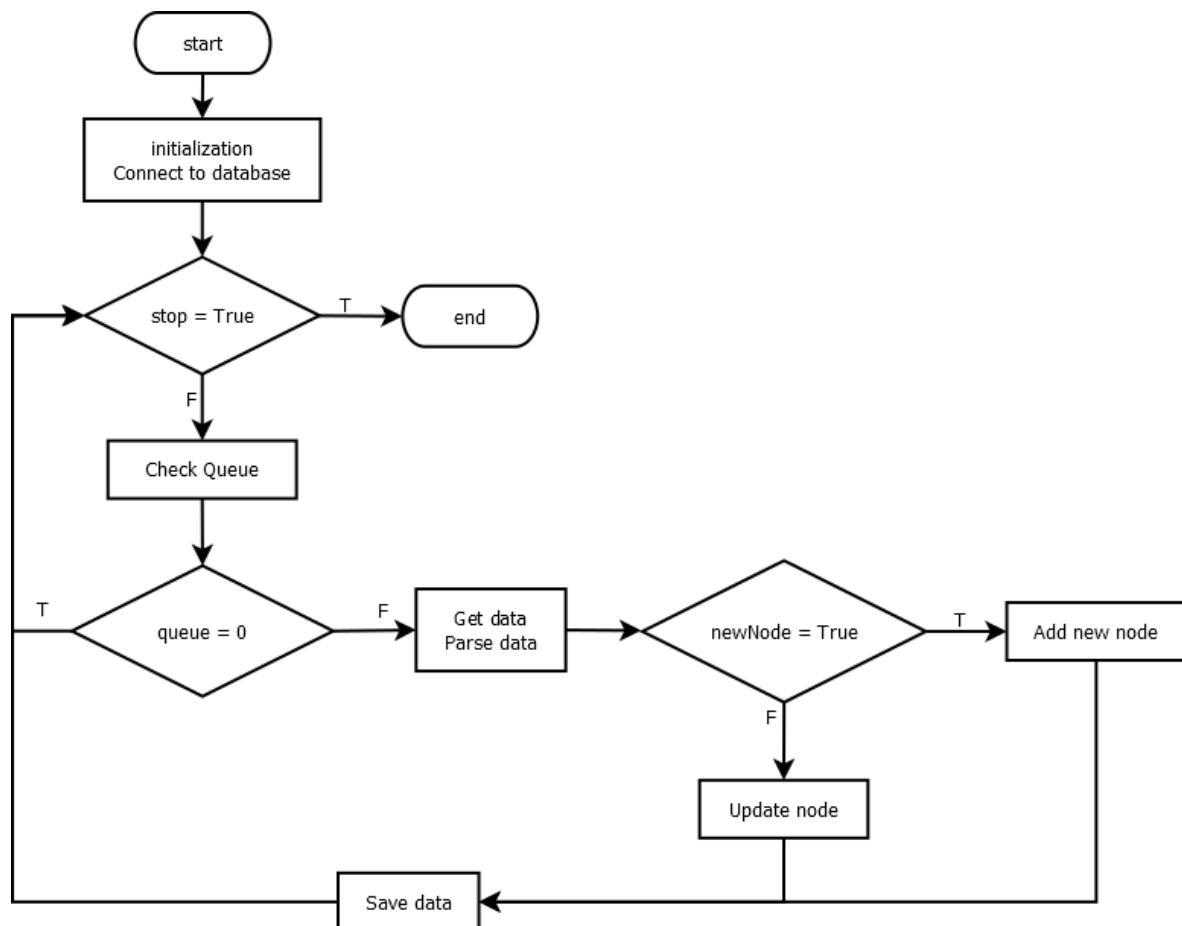
Gambar 3.17 Diagram alir *main thread*

Gambar 3.17 merupakan diagram alir dari program utama yang berjalan di perangkat sentral. Program ini nantinya akan membaca konfigurasi yang sudah ditetapkan kemudian menjalankan beberapa *thread* sesuai dengan konfigurasi. Ada tiga macam *thread* yang dijalankan, yaitu *thread serialWatcher*, *thread worker* dan *thread dbWatcher*. Selanjutnya program ini memeriksa apakah *flag* bernilai *True* dimana program akan berhenti dan menghentikan seluruh *thread*. Jika *flag* bernilai *false* program memeriksa setiap *thread* yang dijalankan. Apabila ditemukan *thread* yang terhenti, program akan menjalankan ulang *thread* yang terhenti. Proses ini terus berlanjut hingga *flag* bernilai *true*.



Gambar 3.18 Diagram alir *thread* serialWatcher

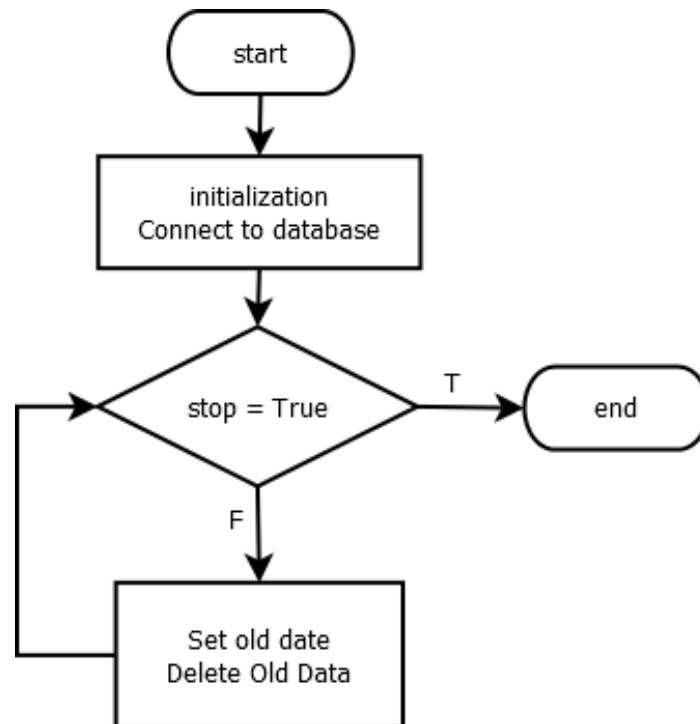
Gambar 3.18 merupakan diagram alir dari *thread* serialWatcher. *Thread* ini bertugas mengambil data dari *port* serial dan memasukkan data ke dalam *queue*. Saat awal dijalankan, *thread* melakukan inisialisasi dan membuka *port* serial sesuai dengan konfigurasi yang ditetapkan. Pada awal perulangan, *thread* memeriksa *flag* yang diatur oleh program utama. Jika *flag* bernilai *True*, *thread* akan berhenti. *Thread* kemudian memeriksa apakah terdapat data yang diterima pada *port* serial. *Thread* akan terus membaca *port* serial hingga menemukan “}” atau hingga seluruh data sudah dibaca. Data yang diterima disimpan ke dalam *queue* untuk diproses oleh *thread* worker



Gambar 3.19 Diagram alir *thread* worker

Gambar 3.19 menunjukkan alur kerja *thread* worker. *Thread* ini bertugas mengambil data dari *queue* dan menyimpan data ke dalam basis data. Saat awal dijalankan, *thread* melakukan inisialisasi berdasarkan konfigurasi yang ditetapkan oleh program utama. Pada awal perulangan, *thread* memeriksa *flag* yang diatur oleh program utama. Jika *flag* bernilai *True*, *thread* akan berhenti. *Thread* kemudian memeriksa apakah terdapat data pada *queue* dan akan mengambil data untuk diproses jika terdapat data.

Thread selanjutnya melakukan pengecekan pada data yang sudah di-*parsing* apakah informasi tentang perangkat sensor yang mengirimkan data terdapat pada basis data. *Thread* akan memperbarui basis data jika informasi perangkat sensor ditemukan.. Jika tidak, maka *thread* akan menambahkan informasi perangkat sensor ke dalam basis data. *Thread* kemudian menyimpan data yang sudah di-*parsing* ke dalam basis data. Proses tersebut kemudian akan diulang oleh *thread* sampai program utama menghentikan *thread*.



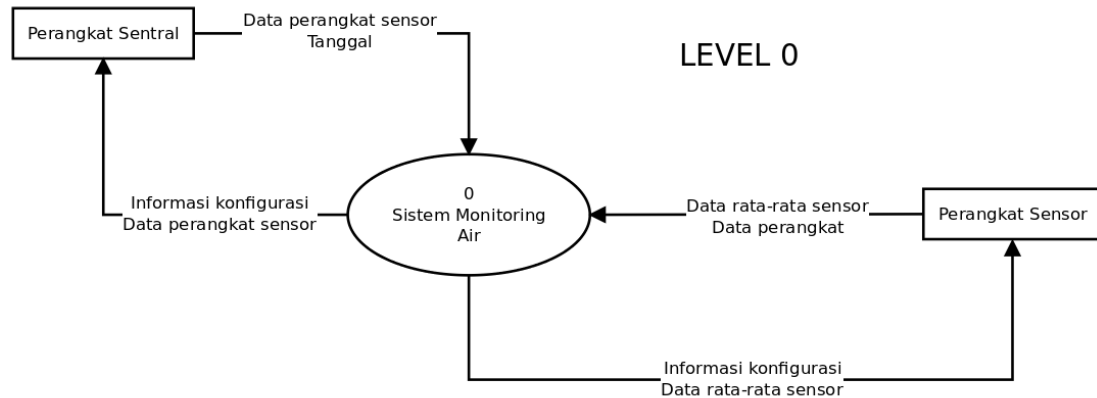
Gambar 3.20 Diagram alir *thread DbWatcher*

Gambar 3.20 adalah alur kerja dari *thread watcher*. *Thread* ini berfungsi menghapus data dari basis data di perangkat sentral yang sudah dianggap tidak berlaku. Pada setiap perulangan, *thread* ini akan menetapkan batas tanggal data yang masih berlaku, selanjutnya menghapus data yang berumur lebih tua dari tanggal tersebut. Proses akan diulang sampai program utama menghentikan *thread*.

3.3.4 Perancangan diagram alir data

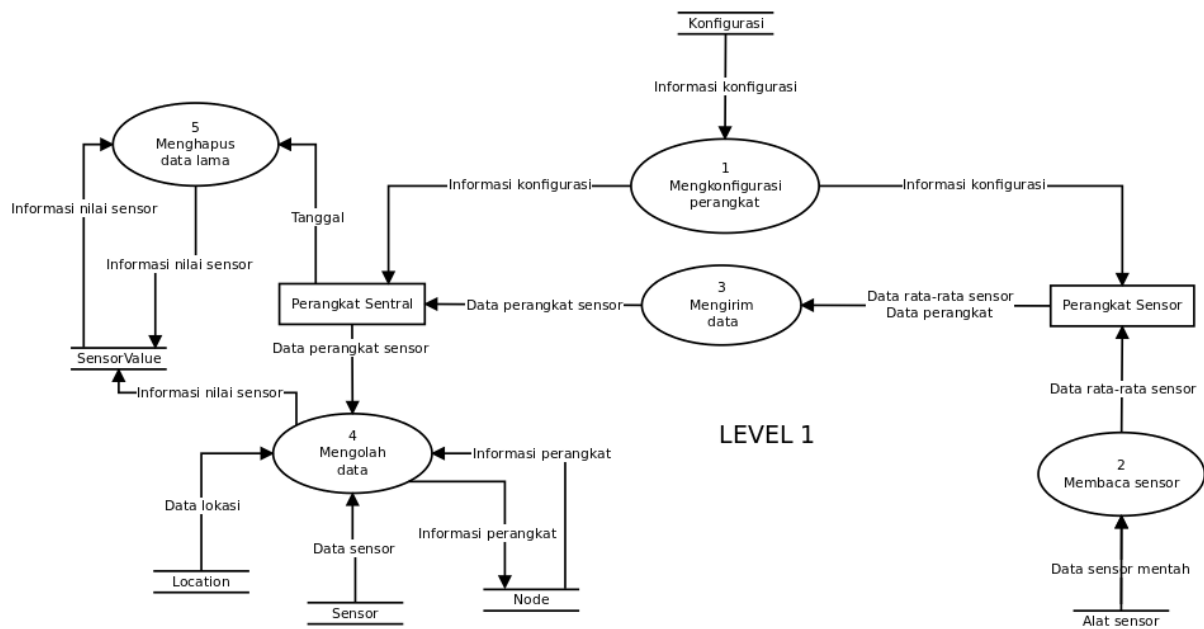
a. Diagram alir data level 0

Pada level nol, terdapat dua entitas luar yaitu Perangkat Sentral dan Perangkat Sensor. Perangkat Sentral dapat memasukkan Data perangkat sensor dan Tanggal. Perangkat Sentral akan menerima Informasi konfigurasi dan Data perangkat sensor. Perangkat Sensor dapat memasukkan Data rata-rata sensor dan Data perangkat. Perangkat Sensor akan menerima Informasi konfigurasi dan Data rata-rata sensor.



Gambar 3.21 Diagram alir data level nol

b. Diagram alir data level 1



Gambar 3.22 Diagram alir data level satu

Pada diagram level satu (Gambar 3.22), terdapat empat proses yaitu Mengkonfigurasi perangkat, Membaca sensor, Mengirim data, Mengolah data, Menghapus data lama.

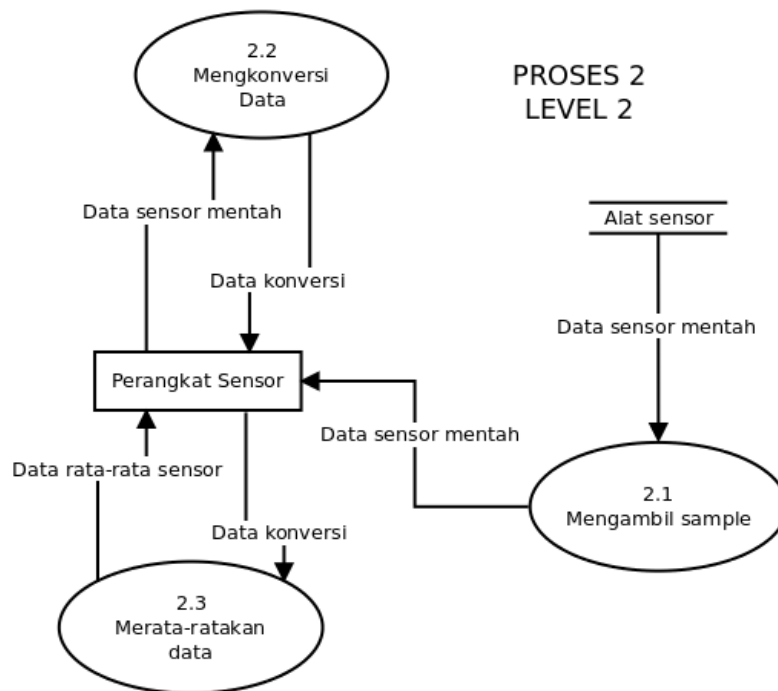
Proses Mengkonfigurasi perangkat dilakukan oleh Perangkat Sentral dan Perangkat Sensor. Proses ini mengambil Informasi konfigurasi dari penyimpanan Konfigurasi. Perangkat Sentral dan Perangkat Sensor menerima Informasi konfigurasi dari proses ini.

Proses Membaca sensor dilakukan oleh Perangkat Sensor. Proses mengambil Data sensor mentah dari penyimpanan Alat sensor. Perangkat Sensor menerima Data rata-rata sensor dari proses ini.

Proses Mengirim data dilakukan oleh Perangkat Sentral dan Perangkat Sensor. Perangkat Sensor dapat memasukkan Data rata-rata sensor dan Data perangkat. Perangkat Sentral menerima Data perangkat sensor.

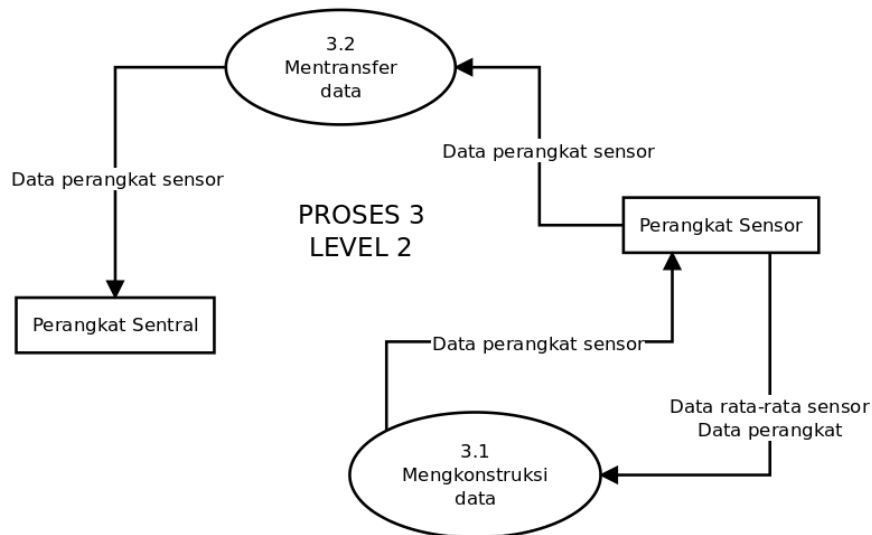
Proses Menghapus data lama dilakukan oleh Perangkat Sentral. Perangkat Sentral dapat memasukkan Tanggal. Proses ini dapat mengambil dan memasukkan Informasi nilai sensor ke penyimpanan *SensorValue*.

c. Diagram alir data level 2



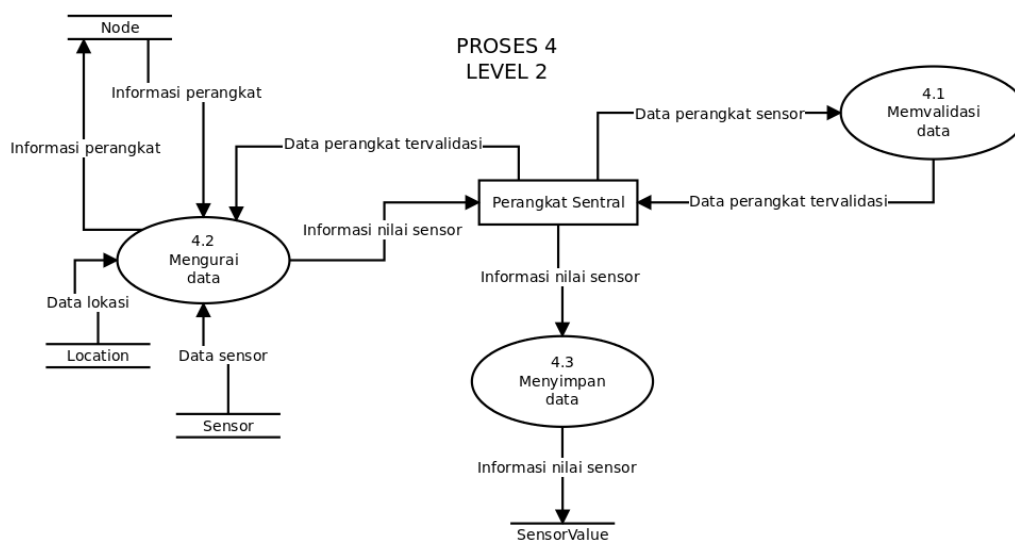
Gambar 3.23 Diagram alir data level dua proses dua

Gambar 3.23 merupakan diagram level dua dari proses Membaca sensor. Pada diagram ini terdapat tiga proses yaitu proses Mengambil sample, Mengkonversi data, dan Merata-ratakan data. Proses Mengambil sample mengambil Data sensor mentah dari penyimpanan Alat sensor. Perangkat Sensor akan menerima Data sensor mentah. Perangkat Sensor dapat memasukkan Data sensor mentah ke dalam proses Mengkonversi data dan menerima Data konversi. Perangkat Sensor dapat memasukkan Data konversi ke proses Merata-ratakan data dan menerima Data rata-rata sensor.



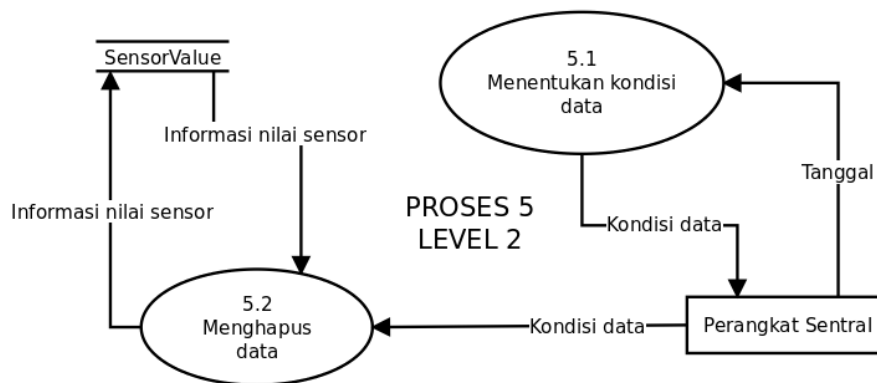
Gambar 3.24 Diagram alir data level dua proses tiga

Gambar 3.24 merupakan diagram level dua dari proses Mengirim data. Pada diagram ini terdapat dua proses yaitu proses Mengkonstruksi data dan proses Mentransfer data. Proses mengkonstruksi data dilakukan oleh Perangkat Sensor. Perangkat Sensor dapat memasukkan Data rata-rata sensor dan Data perangkat ke dalam proses dan menerima Data perangkat sensor. Proses Mentransfer data dilakukan oleh Perangkat Sensor dan Perangkat Sentral. Perangkat Sensor dapat memasukkan Data perangkat sensor. Perangkat Sentral akan menerima Data perangkat sensor dari proses ini.



Gambar 3.25 Diagram alir data level dua proses empat

Gambar 3.25 merupakan diagram level dua dari proses Mengolah data. Pada diagram ini terdapat tiga proses yaitu proses Memvalidasi data, Mengurai data, dan Menyimpan data. Pada proses Memvalidasi data, Perangkat Sentral dapat memasukkan Data perangkat sensor dan menerima Data perangkat tervalidasi. Perangkat Sentral dapat memasukkan Data perangkat tervalidasi dan menerima Informasi nilai sensor. Proses Mengurai data dapat mengambil Data lokasi dari penyimpanan *Location*, Data sensor dari penyimpanan *Sensor*, dan Informasi perangkat dari penyimpanan *Node*. Proses ini dapat menyimpan Informasi perangkat pada penyimpanan *Node*. Perangkat Sentral dapat memasukkan Informasi nilai sensor ke proses Menyimpan data. Proses Menyimpan data akan menyimpan Informasi nilai sensor ke penyimpanan *SensorValue*.



Gambar 3.26 Diagram alir data level dua proses lima

Gambar 3.26 merupakan diagram level dua dari proses Menghapus data lama. Pada diagram ini terdapat dua proses yaitu proses Menentukan kondisi data dan proses Menghapus data. Perangkat Sentral dapat memasukkan Tanggal ke dalam proses Menentukan kondisi data dan menerima Kondisi data. Perangkat sentral dapat memasukkan Kondisi data ke dalam proses Menghapus data. Proses Menghapus data dapat mengambil dan menyimpan Informasi nilai sensor ke penyimpanan *SensorValue*.

3.4 Perancangan basis data

Basis data merupakan komponen penting dalam pembuatan suatu sistem sebagai tempat penyimpanan data. Pembuatan Sistem Monitoring Air ini menggunakan basis data MySQL. MySQL digunakan karena kemudahan dalam penggunaannya serta memiliki kompatibilitas yang baik dengan berbagai bahasa pemrograman. Basis data nantinya disimpan pada perangkat sentral. Pada sistem ini terdapat empat tabel yaitu Tabel *Node*, Tabel *Sensor*, Tabel *SensorValue*, dan Tabel *Location*.

3.4.1 Struktur tabel

a. Tabel Node

Tabel Node menyimpan informasi terkait perangkat sensor seperti lokasi dan waktu pengiriman data terakhir. Tabel Node memiliki beberapa atribut yaitu id, name, location, latitude, longitude, dan lastUpdate. Struktur tabel Node dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel *node*

Atribut	Tipe Data	Keterangan
id	varchar(6)	not null, auto increment, primary key
name	varchar(10)	not null
location	varchar(4)	not null
latitude	varchar(10)	
longitude	varchar(11)	
lastUpdate	datetime	

b. Tabel Sensor

Tabel Sensor menyimpan informasi terkait sensor yang digunakan pada perangkat sensor. Tabel ini memiliki atribut dataHeader dan type. Struktur tabel dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tabel Sensor

Atribut	Tipe Data	Keterangan
dataHeader	varchar(2)	not null, primary key
type	varchar(15)	not null
unit	varchar(3)	

c. Tabel Location

Tabel Location menyimpan informasi terkait lokasi dan kode lokasi yang digunakan pada perangkat sensor. Tabel ini memiliki atribut id dan locationName. Struktur tabel dapat dilihat pada Tabel 3.3.

Tabel 3.3 Tabel Location

Atribut	Tipe Data	Keterangan
id	varchar(4)	not null, primary key
locationName	varchar(50)	not null

d. Tabel SensorValue

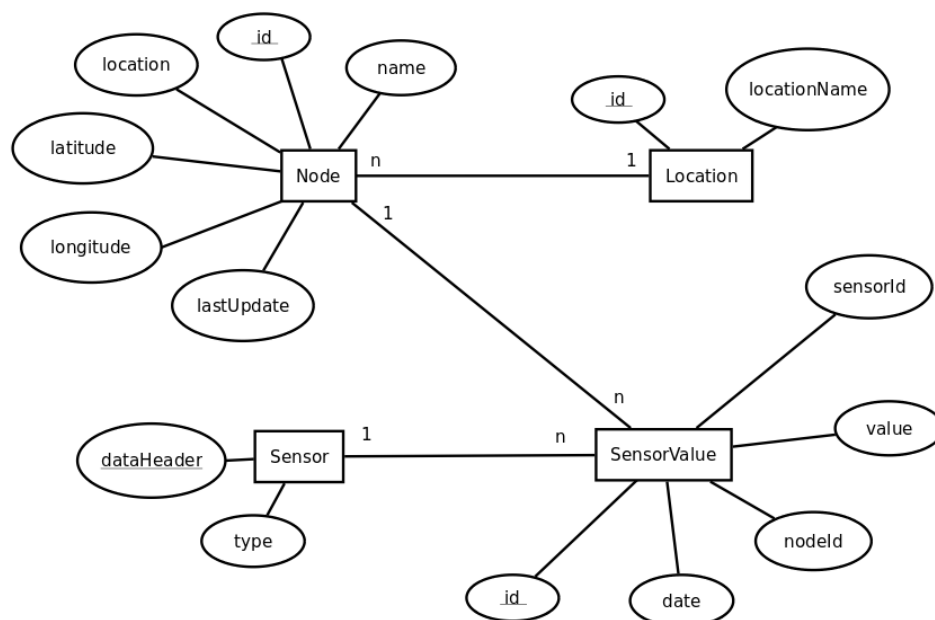
Tabel SensorValue menyimpan informasi terkait data sensor yang sudah diolah oleh perangkat sensor. Tabel ini memiliki atribut id, date, nodeId, sensorId dan value. Atribut nodeId dan sensorId merupakan foreign key. Atribut nodeId diambil dari tabel Node. Atribut sensorId diambil dari tabel Sensor. Struktur tabel dapat dilihat pada Tabel 3.4.

Tabel 3.4 Tabel SensorValue

Atribut	Type Data	Keterangan
id	int(4)	not null, auto increment
date	datetime	not null
nodeId	varchar(6)	not null
sensorId	varchar(2)	not null
value	float	

3.4.2 Perancangan ERD

ERD menunjukkan relasi antar tabel. Gambar 3.27 menunjukkan relasi antara empat tabel pada sistem ini, yaitu tabel Node, tabel Location, tabel Sensor dan tabel SensorValue.



Gambar 3.27 Diagram relasi tabel

Tabel Node dan tabel SensorValue memiliki relasi 1:n yang artinya sebuah node atau perangkat sensor dapat memiliki banyak data sensor. Tabel Sensor dan tabel SensorValue memiliki relasi 1:n yang artinya sebuah sensor dapat memiliki banyak data sensor. Tabel

Location dan tabel SensorValue memiliki relasi 1:n yang artinya terdapat banyak data sensor pada satu lokasi.

3.5 Rencana Implementasi

Purwarupa sistem dibuat berdasarkan pada rancangan sistem yang sudah dibuat. Purwarupa yang dibuat nantinya akan ditempatkan langsung di lokasi mata air. Purwarupa dibuat sesuai dengan lingkup pada penelitian ini yaitu proses pengambilan data air, komunikasi antar perangkat serta penyimpanan data pada basis data perangkat sentral. Secara ideal, perangkat sensor yang dibuat seharusnya dapat mengambil data air melalui sensor secara akurat. Data yang diambil dapat diolah dan dikirimkan melalui jaringan nirkabel oleh perangkat sensor tanpa adanya *loss*. Perangkat sentral dapat menerima data secara utuh tanpa ada kerusakan data. Data yang diterima oleh perangkat sentral dapat diproses dan disimpan dalam basis data.

Pengambilan data air dilakukan oleh perangkat sensor yang dibuat menggunakan Arduino yang terhubung dengan sensor. Komunikasi antar perangkat dilakukan dengan menggunakan XBee yang dipasang pada Arduino dan Raspberry Pi melalui XBee *shield*. Program utama yang terdapat pada perangkat sentral (Raspberry Pi) dibuat dengan menggunakan bahasa pemrograman Python 2.7.

3.6 Rencana Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui apakah sistem yang dibuat dapat diimplementasikan dalam skala sebenarnya. Pengujian yang dilakukan adalah pengujian fungsi dan pengujian sensor.

Pengujian fungsi dilakukan untuk mengetahui apakah sistem dapat berjalan sesuai dengan rancangan sistem yang dibuat. *Black box* adalah metode pengujian perangkat lunak yang menguji fungsionalitas tanpa mengetahui proses di dalamnya. Pengujian *black box* terfokus pada masukan dan keluaran program apakah hasil sesuai dengan rancangan. Pada sistem ini dilakukan beberapa pengujian fungsi pada fungsi utama sistem. Fungsi yang diujikan ditampilkan pada Tabel 3.5. Kolom Hasil diisi dengan “sesuai” jika fungsi alat berjalan seperti hasil yang diharapkan dan “tidak sesuai” jika perangkat tidak berjalan seperti hasil yang diharapkan.

Pengujian sensor dilakukan untuk mengetahui apakah data yang diambil oleh perangkat sensor benar-benar membaca sifat air. Pengujian sensor dilakukan dengan melakukan pembacaan beberapa sampel air.

Tabel 3.5 Tabel pengujian fungsi perangkat

No	Perangkat	Fungsi	Hasil yang diharapkan	Hasil	Keterangan
1	Perangkat sensor	Membuang sampel	Perangkat sensor dapat membuang sampel air		
2	Perangkat sensor	Mengambil sampel	Perangkat sensor mengambil sampel air pada periode tertentu.		
3	Perangkat sensor	Mengisi penampung	Perangkat sensor mengisi penampung air hingga ketinggian yang ditentukan		
4	Perangkat sensor	Membaca sensor	Perangkat sensor dapat mengambil data air.		
5	Perangkat sensor	Mengirim data	Perangkat sensor dapat mengirimkan data ke perangkat sentral.		
6	Perangkat sentral	Menerima data	Perangkat sentral dapat menerima data dari perangkat sensor.		
7	Perangkat sentral	Mengurai data	Perangkat sentral dapat melakukan <i>parsing</i> data dari perangkat sensor.		
8	Perangkat sentral	Menyimpan data	Perangkat sentral dapat menyimpan data ke basis data.		
9	Perangkat sentral	Mengapus data	Perangkat sentral dapat menghapus data pada periode tertentu.		

3.6.1 Langkah pengujian fungsi

Pengujian fungsi yang akan dilakukan pada kedua perangkat adalah sebagai berikut:

- Melakukan konfigurasi dan pengecekan konektivitas kedua perangkat.
- Perangkat sensor dan perangkat sentral dihubungkan dengan komputer.
- Penampung sampel diisi dengan air.
- Pipa yang terhubung dengan penampung disambungkan dengan sumber air.
- Perangkat sensor dan perangkat sentral dinyalakan.
- Melakukan pengamatan terhadap keluaran kedua perangkat pada komputer dan melakukan pencocokan dengan aktivitas yang dilakukan kedua perangkat.

BAB IV

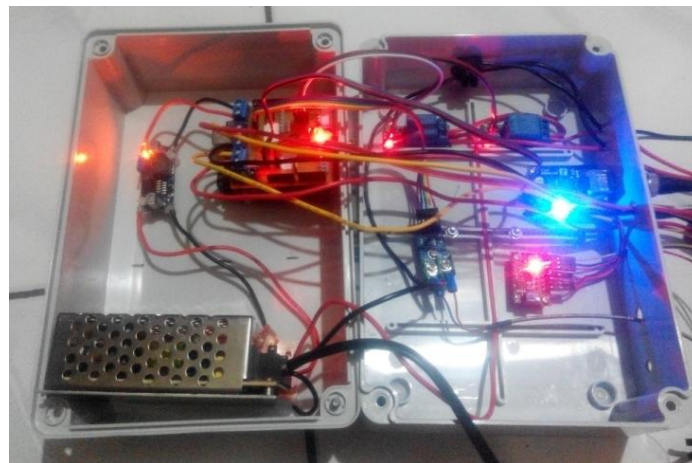
HASIL DAN PEMBAHASAN

4.1 Hasil

Hasil berisi implementasi dari rancangan yang sudah dibuat. Implementasi Proses implementasi membutuhkan satu buah komputer, Raspberry Pi 3 B, Arduino UNO, XBEE, sensor dan beberapa komponen lainnya. Implementasi dilakukan dalam beberapa langkah. Langkah pertama adalah membuat perangkat sensor sesuai dengan rancangan dengan menggunakan Arduino. Langkah kedua adalah konfigurasi Raspberry Pi sebagai perangkat sentral.

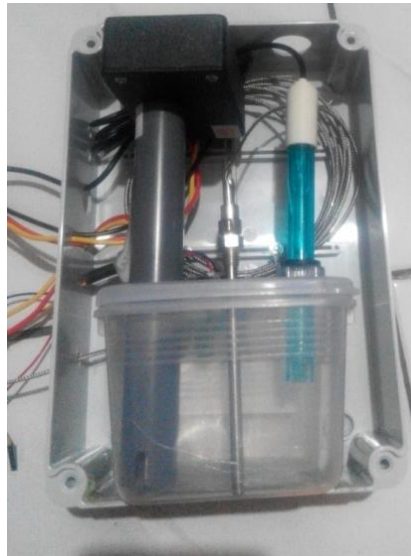
4.1.1 Perakitan perangkat sensor

Perangkat sensor diimplementasikan dengan menggunakan Arduino UNO. Arduino dan sensor yang ada dipasang pada sebuah kotak berukuran 22 x 15 x10 cm. Ada dua kotak yang digunakan. Kotak pertama (Gambar 4.1) berisi Arduino UNO, suplai daya dan modul yang terhubung dengan sensor dan Arduino. Suplai daya Arduino nantinya akan diambil dari aki.



Gambar 4.1 Perangkat sensor

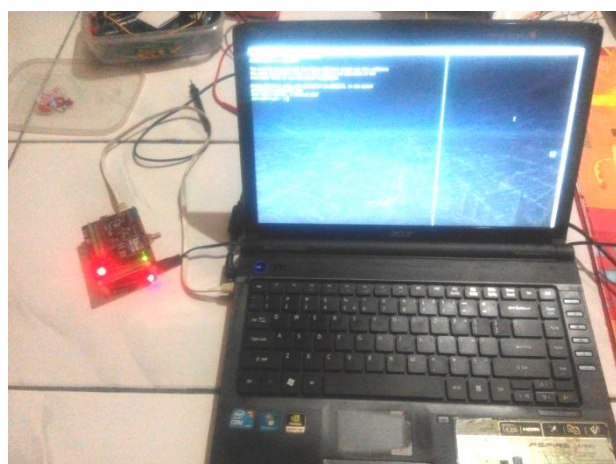
Kotak kedua (Gambar 4.2) berisi pompa DC dan sensor yang dipasang pada sebuah penampung. Penampung ini digunakan untuk menampung sementara sampel yang akan diambil datanya. Air sampel dimasukkan ke dalam penampung melalui pipa yang terhubung dengan katup solenoid. Pembuangan sampel dilakukan dengan selang yang dipasang pada pompa DC.



Gambar 4.2 Sensor kualitas air

4.1.2 Instalasi sistem operasi Raspbian

Sebelum melakukan konfigurasi, dilakukan instalasi sistem operasi pada Raspberry Pi. Sistem operasi yang digunakan adalah sistem operasi Raspbian. Raspbian merupakan sistem operasi yang dibuat secara khusus untuk Raspberry Pi. Sistem operasi ini berbasis pada Linux Debian. Versi Raspbian yang digunakan adalah Raspbian Jessie. Konfigurasi pada Raspberry Pi dilakukan dengan menggunakan sebuah komputer. Komputer dan Raspberry Pi dapat dihubungkan melalui kabel (Gambar 4.3) atau secara nirkabel melalui wifi.



Gambar 4.3 Perangkat sentral yang terhubung dengan laptop

Pengaturan konfigurasi dilakukan melalui koneksi SSH (*Secure Shell*). SSH adalah sebuah protokol *remote access* yang secara *native* didukung oleh sistem operasi berbasis Linux. Secara *default* pengalamatan IP pada Raspberry Pi menggunakan DHCP, maka perlu

dilakukan konfigurasi alamat IP untuk mempermudah proses implementasi. Konfigurasi alamat ditunjukkan pada Gambar 4.4

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
NETWORK=192.168.1.0
NETMASK=255.255.255.0
IPADDR=192.168.1.1
GATEWAY=192.168.1.2
```

Gambar 4.4 Konfigurasi alamat IP

4.1.3 Implementasi basis data

Basis data yang sebelumnya sudah dirancang diimplementasikan pada Raspberry Pi sebagai perangkat sentral. DBMS yang digunakan adalah MariaDB. DBMS merupakan perangkat lunak yang digunakan dalam proses manajemen basis data. MariaDB dipilih karena merupakan perangkat lunak terbuka dan memiliki kemampuan proses data dan penggunaan memori yang baik.

Rancangan basis data diimplementasikan pada MariaDB melalui *query*. Basis data dibuat dengan nama WaterMonitoring. Sesuai dengan rancangan yang telah dibuat, basis data diimplementasikan dalam 4 tabel yang bernama Location, Node, Sensor dan SensorValue ditunjukkan pada Gambar 4.5.

```
MariaDB [WaterMonitoring]> SHOW TABLES;
+-----+
| Tables_in_WaterMonitoring |
+-----+
| Location                    |
| Node                        |
| Sensor                      |
| SensorValue                 |
+-----+
4 rows in set (0.00 sec)
```

Gambar 4.5 Basis data WaterMonitoring

a. Tabel Location

```
MariaDB [WaterMonitoring]> DESCRIBE Location;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | varchar(4) | NO   | PRI | NULL    |       |
| locationName | varchar(50) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Gambar 4.6 Isi tabel Location

b. Tabel Node

```
MariaDB [WaterMonitoring]> DESCRIBE Node;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | varchar(6) | NO   | PRI | NULL    |       |
| name       | varchar(10) | NO   |     | NULL    |       |
| location   | varchar(4) | NO   |     | NULL    |       |
| latitude   | varchar(10) | YES  |     | NULL    |       |
| longitude  | varchar(11) | YES  |     | NULL    |       |
| lastUpdate | datetime   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Gambar 4.7 Isi tabel Node

c. Tabel Sensor

```
MariaDB [WaterMonitoring]> DESCRIBE Sensor;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| dataHeader | varchar(2) | NO   | PRI | NULL    |       |
| type       | varchar(15) | NO   |     | NULL    |       |
| unit       | varchar(3) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Gambar 4.8 Isi tabel Sensor

d. Tabel SensorValue

```
MariaDB [WaterMonitoring]> DESCRIBE SensorValue;
```

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	NULL	auto_increment
date	datetime	NO		NULL	
nodeId	varchar(6)	NO		NULL	
sensorId	varchar(2)	NO		NULL	
value	float	YES		NULL	

5 rows in set (0.00 sec)

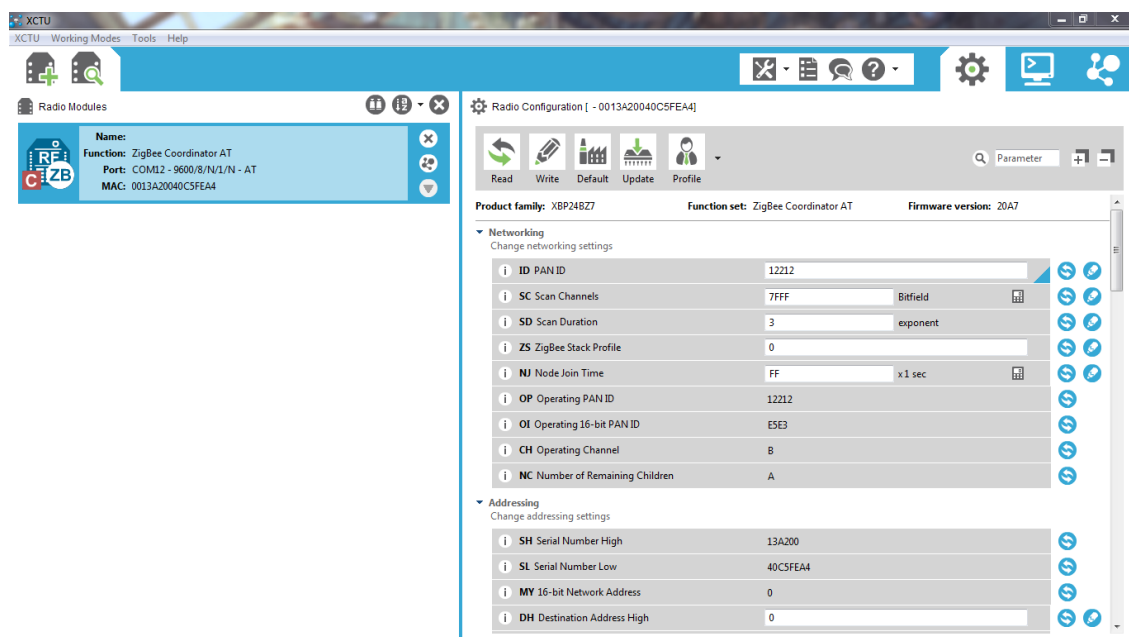
Gambar 4.9 Isi tabel SensorValue

4.1.4 Konfigurasi XBee

Konfigurasi XBee dilakukan dengan perangkat lunak XCTU. XCTU merupakan perangkat lunak yang dibuat khusus untuk keperluan konfigurasi XBee. Perangkat lunak yang mendukung semua jenis XBee ini juga dapat digunakan untuk melakukan pembaruan versi.

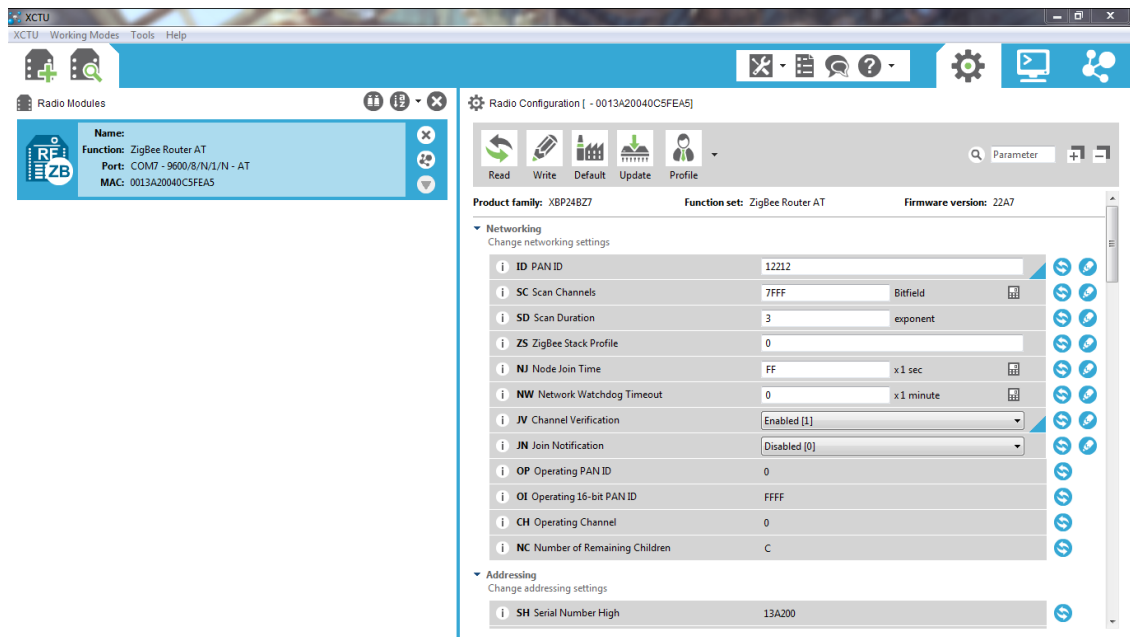
XBee pada perangkat sentral dibuat sebagai *Coordinator* sedangkan XBee pada perangkat sensor sebagai *router*. PAN ID pada kedua XBee diatur menjadi 12212. =

a. Konfigurasi XBee perangkat sentral



Gambar 4.10 XBee perangkat sentral

b. Konfigurasi XBee perangkat sensor



Gambar 4.11 XBee perangkat

4.2 Pembahasan

Pembahasan berisi penjelasan singkat dari potongan kode program yang telah dibuat. Kode program dibuat berdasarkan rancangan yang ada. Pembahasan dibagi menjadi tiga bagian yaitu proses inisialisasi perangkat, pembacaan sensor serta pengiriman dan penerimaan.

4.2.1 Inisialisasi perangkat

a. Perangkat sensor

Gambar 4.12 adalah proses yang dilakukan pada tahap inisialisasi perangkat sensor. Tahap ini dilakukan saat perangkat dinyalakan. Pada tahap ini, perangkat sensor akan membaca konfigurasi yang terdapat pada EEPROM. Perangkat kemudian berjalan sesuai dengan konfigurasi. Setelah konfigurasi selesai, perangkat akan mulai membaca sensor aliran air.

Keluaran dari sensor aliran air dibaca dengan fungsi `pulseCounter()` seperti ditunjukkan pada Gambar 4.13. Fungsi ini akan menjumlahkan setiap keluaran yang dihasilkan oleh sensor. Hasil dari keluaran sensor selanjutnya diubah menjadi satuan liter dengan fungsi `addFlow()` yang dijalankan setiap detik.

```

174 int pointer, len, temp[3];
175   for (pointer = 0; pointer < 3; pointer++) {
176       temp[pointer] = EEPROM.read(pointer);
177   }
178   samplingInterval = temp[0]*1000;
179   waitTime = temp[1]*1000;
180   readTimes = temp[2];
181
182   len = EEPROM.read(pointer);
183   len += pointer; pointer++;
184   for (pointer; pointer < len; pointer++) {
185       nodeId += (char)EEPROM.read(pointer);
186   }
187   nodeId.trim();
188
189   len = EEPROM.read(pointer);
190   len += pointer; pointer++;
191   for (pointer; pointer < len; pointer++) {
192       coordinate += (char)EEPROM.read(pointer);
193   }
194   coordinate.trim();

```

Gambar 4.12 Kode konfigurasi waktu jeda dan jumlah pembacaan

```

366 void pulseCounter() {
367     pulseCount++;
368 }
369
370 float addFlow() {
371     float flowRate = ((1000.0 / (millis() - prevFlow)) * pulseCount)
372     / calibrationFlow;
373     float flowLitres = (float)(flowRate / 60);
374     buff[0] += flowLitres + offsetFlow;
375     Ser.print("Total usage: "); Ser.print(buff[0]); Ser.println("L");
376     pulseCount = 0;
377 }

```

Gambar 4.13 Kode membaca keluaran sensor aliran air

b. Perangkat sentral

```

123  """Load config"""
124  self.hostname = readConf.get("Database", "hostname")
125  self.dbUser = readConf.get("Database", "username")
126  self.dbPass = readConf.get("Database", "password")
127  self.dbName = readConf.get("Database", "dbname")
128  self.location = readConf.get("Global", "Location").upper()
129  self.runWatcher = readConf.getboolean("Global", "rundbWatcher")
130  self.workerThreads = readConf.getint("Global", "workerthreads")
131  if (self.workerThreads < 0): raise Exception("Threads number must >=
132  0")
133  self.setOld(readConf.get("Database", "oldData"))
134  self.port = readConf.get("Serial", "device")
135  self.baud = readConf.get("Serial", "baudrate")

```

Gambar 4.14 Konfigurasi perangkat

Program yang berjalan pada perangkat sentral diimplementasikan sesuai dengan rancangan yang telah dibuat. Program dibuat dengan bahasa pemrograman Python 2.7. Saat pertama dijalankan program akan menjalankan proses MainThread. Proses MainThread akan membaca *file* konfigurasi seperti ditunjukkan pada Gambar 4.14. *File* konfigurasi berisi berbagai konfigurasi yang dibutuhkan seperti konfigurasi *port* dan basis data.

Setelah berhasil melakukan konfigurasi, MainThread akan mengambil daftar perangkat sensor serta sensor yang digunakan dari basis data. Daftar ini nantinya digunakan saat dilakukan *parsing* data. Proses ini ditunjukkan pada Gambar 4.15.

```

158  """Get sensor list"""
159  log.info("Getting list of sensors")
160  cur.execute("SELECT dataHeader FROM Sensor")
161  for sensor in cur.fetchall():
162      threadClass.putSensor(sensor)
163
164  """Get node list"""
165  log.info("Getting list of nodes")
166  cur.execute("SELECT id from Node")
167  for node in cur.fetchall():
168      threadClass.putNode(node[0])

```

Gambar 4.15 Pengambilan daftar sensor dan perangkat sensor

```

83 def runWatcherThread(self):
84     t = threadClass.dbWatcher(timeDiff = self.oldData, dbInfo =
85         (self.hostname, self.dbUser, self.dbPass, self.dbName))
86     t.start()
87     self.threadList.append(t)
88
89 def runSerialWatcherThread(self):
90     t = threadClass.serialWatcher(port = self.port, baud =
91         self.baud)
92     t.start()
93     self.threadList.append(t)
94
95 def runWorkerThread(self, worker):
96     t = threadClass.worker(name=worker, location = self.location,
97         dbInfo=(self.hostname, self.dbUser, self.dbPass,
98             self.dbName))
99     t.start()
100    self.threadList.append(t)

```

Gambar 4.16 Kode untuk menjalankan *thread*

Selanjutnya, MainThread akan menjalankan beberapa sub-proses atau disebut juga *thread*. *Thread* ini yang nantinya akan melakukan pengolahan data. MainThread hanya berfungsi untuk membaca konfigurasi, dan menjalankan *thread*. Proses menjalankan *thread* ditunjukkan pada Gambar 4.16.

```

274 log.info("Checking db")
275 oldTime = time.strftime("%Y-%m-%d %H:%M:%S",
276     time.localtime(time.time() - self.oldData))
277 self.cur.execute("SET sql_bin_log = 0")
278 self.cur.execute("DELETE FROM SensorValue WHERE Date < '%s'" %
279     (oldTime))
280 self.db.commit()
281 self.cur.execute("SET sql_bin_log = 1")
282 log.info("Done")

```

Gambar 4.17 Kode *thread* dbWatcher

Ada tiga macam *thread* yang berjalan yaitu *thread* serialWatcher yang menerima data dari *port* serial, *thread* worker yang melakukan *parsing* dan menyimpan data ke basis data serta *thread* dbWatcher yang melakukan penghapusan data. *Thread* dbWatcher menjalankan

fungsiya setiap beberapa waktu sekali sesuai konfigurasi. Gambar 4.17 menunjukkan proses yang dilakukan saat thread melakukan penghapusan data.

4.2.2 Pembacaan sensor

```

206 void dumpSample() {
207     Ser.print(F("Dumping Sample at ")); Ser.println(millis());
208     digitalWrite(7, HIGH);
209     if (digitalRead(7) == 0) {
210         digitalWrite(6, LOW); ////Open Relay 2 (PUMP)
211         unsigned long wait = millis();
212         while (millis() - wait <= 8500);
213         digitalWrite(6, HIGH); ////Close Relay 2 (PUMP)
214     }
215     digitalWrite(7, LOW);
216 }

```

Gambar 4.18 Kode sumber membuang sampel

Proses pembacaan hanya dilakukan oleh perangkat sensor. Sebelum membaca sifat air, perangkat sensor akan melakukan pengecekan apakah terdapat air pada penampung sampel. Jika terdapat air pada penampung, maka perangkat akan menjalankan proses dumpSample() seperti ditunjukkan pada Gambar 4.18. Perangkat akan menyalakan relai yang terhubung pada pompa pembuangan selama 8,5 detik untuk memastikan air pada penampung terbuang.

Perangkat selanjutnya akan menjalankan fungsi fillSample() untuk mengisi air ke dalam penampung sampel (Gambar 4.19). Perangkat akan menyalakan relai yang terhubung dengan katup solenoid. Saat katup terbuka, perangkat akan membaca ketinggian air. Ketika ketinggian air sampai pada titik yang ditentukan, perangkat akan menutup katup dan mulai membaca sensor. Apabila air sampel pada penampung tidak mencapai ketinggian yang ditentukan saat waktu tunggu habis, perangkat akan menganggap penampung sampel kosong dan mengirimkan data kosong kepada perangkat sentral.

```

220 void fillSample() {
221     unsigned long timer = millis();
222     Ser.print(F("Filling Sample at ")); Ser.println(millis());
223     digitalWrite(5, LOW);          ////Open Relay 1 (SOLENOID)
224     while (readHeight() != 0 && millis() - timer < waitTime);

```

```

225     digitalWrite(5, HIGH);          /////Close Relay 1 (SOLENOID)
226     unsigned long wait = millis();
227     while (millis() - wait <= 5000);
228 }

```

Gambar 4.19 Kode sumber mengisi sampel

Proses pembacaan sensor dilakukan hanya jika penampung sampel tidak kosong. Masing-masing sensor dibaca sebanyak jumlah pembacaan yang diambil saat proses konfigurasi. Proses pembacaan dari setiap sensor dilakukan dengan menggunakan fungsi terpisah karen perbedaan rumus yang digunakan. Fungsi yang dijalankan ditunjukkan pada beberapa gambar berikut.

```

355 float voltage = (float)sortAverage(reading) * 0.004882813;
356 int turbidity = 100 - (voltage / calibrationTur) * 100 + offset;
357 if (turbidity < 0) turbidity = 0;
358 Ser.print(F("Turbidity= ")); Ser.println(turbidity);

```

Gambar 4.20 Kode untuk membaca sensor *turbidity*

```

399 float tempResult = sortAverage(reading) + offsetTemp;
340 Ser.print(F("Temperature= ")); Ser.println(tempResult);
341 return tempResult;

```

Gambar 4.21 Kode untuk membaca sensor suhu

```

426 float phResult = (float) 3.5 * (sortAverage(reading) * 0.004882813 +
427 offsetPH) ;
428 Ser.print(F("PH= ")); Ser.println(phResult);
429 return phResult;

```

Gambar 4.22 Kode untuk membaca sensor pH

```

329 float conductivity = 0.518 * sortAverage(reading) + 222.4 +
330 offsetConductivity;
331 Ser.print(F("Conductivity= ")); Ser.println(conductivity);
332 return conductivity;

```

Gambar 4.23 Kode untuk membaca sensor konduktivitas

4.2.3 Pengiriman dan penerimaan

a. Perangkat sensor

Setelah proses pembacaan selesai, data selanjutnya akan disusun dan dikirimkan ke perangkat sentral melalui XBEE. Arduino dan XBEE berkomunikasi melalui komunikasi serial. Untuk mengirimkan data ke perangkat sentral, Arduino hanya cukup mengirimkan data yang sudah disusun melalui komunikasi serial. XBEE yang sudah dikonfigurasi secara otomatis akan mengirimkan data ke perangkat sentral. Proses pengiriman data ditunjukkan pada Gambar 4.24. Seluruh proses akan terus berulang hingga perangkat sensor dimatikan

```

297 void sendData(float *buff) {
298     String data = constructData(buff);
299     Ser.print(F("Data: "));
300     Ser.println(data);
301     Serial.print(data);
302     Ser.println();
303 }
```

Gambar 4.24 Kode untuk mengirim data

a. Perangkat sentral

```

68 if (serialDevice.inWaiting() > 0):
69     serialLen = serialDevice.inWaiting()
69     while (serialDevice.inWaiting > serialLen):
70         temp = serialDevice.read()
71         data += temp
72         if (temp == "}" and len(data) > 1 ): break
73     putQueue(data)
```

Gambar 4.25 Kode *thread* serialWatcher saat menerima data

Data yang dikirim oleh perangkat sensor akan diterima akan diterima oleh XBEE pada perangkat sentral yang mengirimkan data ke Raspberry Pi melalui komunikasi serial. *Thread* serialWatcher akan membaca data yang dikirimkan dan menyimpan data tersebut ke dalam sebuah queue atau antrian. Proses yang dijalankan ditunjukkan pada Gambar 4.25.

Data yang disimpan pada antrian selanjutnya akan diproses oleh *thread* worker. *Thread* ini melakukan parsing data berdasarkan daftar perangkat sensor dan sensor yang sudah diambil saat proses konfigurasi. Gambar 4.26 adalah proses saat *thread* worker menyimpan data ke basis data. Apabila perangkat sensor yang mengirimkan data sudah

terdapat pada daftar, *thread* akan langsung menyimpan data ke basis data. Perangkat sensor yang belum terdaftar akan secara otomatis ditambahkan ke dalam daftar dan basis data.

```

181 if (nodeId in nodeList):
182     """Update node information"""
183     self.cur.execute("UPDATE Node SET Latitude = '%s', Longitude =
184 '%s', lastUpdate = '%s' WHERE id = '%s'" % (lat, long, dateTime,
185 nodeId))
186     self.db.commit()
187 else:
188     """Automatically add new node to DB"""
189     self.cur.execute("INSERT INTO Node values ('%s', 'Node %s',
190 '%s', '%s', '%s', '%s')" % (nodeId, nodeId[3:],
191 nodeId[:3].lstrip('0'), lat, long, dateTime))
192     self.db.commit()
193     if (lock.acquire()):
194         if not nodeId in nodeList:
195             nodeList.append(nodeId)
196         lock.release()
197
198 """Insert data to DB"""
199 for record in dataParsed:
200     self.cur.execute("INSERT INTO SensorValue(date, nodeId,
201 sensorId, value) values ('%s', '%s', '%s', %s)" % (dateTime, nodeId,
202 record[0], record[1]))
203     self.db.commit()

```

Gambar 4.26 Kode untuk menyimpan data yang dijalankan *thread* worker

4.3 Pengujian fungsi perangkat

Pengujian fungsi dilakukan untuk mengetahui apakah sistem dapat berjalan sesuai dengan rancangan sistem yang dibuat. Pengujian dilakukan dengan menguji setiap fungsi sistem berdasarkan pada *form* pengujian yang telah dibuat. Pengujian fungsi dilakukan pada perangkat sensor dan perangkat sentral.

4.3.1 Inisialisasi perangkat

a. Perangkat sensor

Gambar 4.27 menunjukkan keluaran perangkat sensor saat proses inisialisasi. Perangkat sensor menampilkan konfigurasi perangkat yang diambil dari EEPROM.

Konfigurasi yang diambil yaitu id, koordinat, jeda pembacaan, jeda pengisian dan jumlah pembacaan yang dilakukan. Perangkat selanjutnya mulai membaca sensor aliran air.

```
Automated Water Monitoring System
Observer Node v0.5
ID: YOG212
Coordinate: -7.776706,110.371119
Interval: 30000
Fill Wait: 25000
Read: 100

Reading Flow
Total usage: 0.00L
```

Gambar 4.27 Inisialisasi perangkat sensor

b. Perangkat sentral

INFO	17-12-06 13:19:20 MainThread	Initializing node
INFO	17-12-06 13:19:20 MainThread	Connecting to db server
INFO	17-12-06 13:19:20 MainThread	Connected to db server
INFO	17-12-06 13:19:20 MainThread	Loading configuration
INFO	17-12-06 13:19:20 MainThread	Getting list of sensors
INFO	17-12-06 13:19:20 MainThread	Getting list of nodes
DEBUG	17-12-06 13:19:20 MainThread	Location: YOG
DEBUG	17-12-06 13:19:20 MainThread	Port: /dev/ttyACM0
DEBUG	17-12-06 13:19:20 MainThread	Baudrate: 9600
DEBUG	17-12-06 13:19:20 MainThread	Run Watcher: True
DEBUG	17-12-06 13:19:20 MainThread	Worker Threads: 2
DEBUG	17-12-06 13:19:20 MainThread	Old Data: 60s
INFO	17-12-06 13:19:20 MainThread	Done

Gambar 4.28 Inisialisasi MainThread

Keluaran tahap inisialisasi pada perangkat sentral ditampilkan pada Gambar 4.28. Proses MainThread yang merupakan proses utama mengambil konfigurasi dari sebuah *file* yang bernama awm.conf. Konfigurasi yang diambil dari file ini diterapkan ke dalam program. Setelah proses konfigurasi, proses MainThread menjalankan beberapa *thread*. Gambar 4.29 menunjukkan saat MainThread menjalankan *thread*. MainThread menjalankan thread dbWatcher, serialWatcher dan dua *thread* Worker.

INFO	17-12-06 13:19:20	MainThread	Done
INFO	17-12-06 13:19:20	MainThread	Starting Thread
INFO	17-12-06 13:19:20	dbWatcher	Ready
INFO	17-12-06 13:19:20	SerialWatcher	Opening Serial port
INFO	17-12-06 13:19:20	MainThread	Connected to db server
INFO	17-12-06 13:19:20	SerialWatcher	Ready
INFO	17-12-06 13:19:20	SerialWatcher	Waiting for data
INFO	17-12-06 13:19:20	Worker 0	Ready
INFO	17-12-06 13:19:20	MainThread	Connected to db server
INFO	17-12-06 13:19:20	Worker 1	Ready

Gambar 4.29 Taham menjalankan *thread*

Pada Gambar 4.30, *thread* dbWatcher melakukan pengecekan basis data di awal program berjalan. *Thread* akan menghapus data yang melebihi batas waktu yang ditentukan. Batas waktu pada pengujian ini adalah 60 detik. Gambar 4.31 menunjukkan isi basis data sebelum program berjalan. Setelah *thread* melakukan pengecekan, data yang melebihi batas waktu dihapus. Gambar 4.32 menunjukkan isi tabel sesudah data dihapus.

INFO	17-12-06 13:19:20	MainThread	Connected to db server
INFO	17-12-06 13:19:20	dbWatcher	Checking db
DEBUG	17-12-06 13:19:20	dbWatcher	60
DEBUG	17-12-06 13:19:20	dbWatcher	17-12-06 13:19:20
INFO	17-12-06 13:19:20	dbWatcher	Done

Gambar 4.30 *Thread* dbWatcher

mysql> select * from SensorValue;					
+	-----+	-----+	-----+	-----+	-----+
	id		date		nodeId sensorId value
+	-----+	-----+	-----+	-----+	-----+
	6		2017-12-01 14:37:08		YOG0212 FL 1
	7		2017-12-01 14:37:08		YOG0212 TU 50.0
	8		2017-12-01 14:37:08		YOG0212 TE 25.36
	9		2017-12-01 14:37:08		YOG0212 PH 5.97
	10		2017-12-01 14:37:08		YOG0212 CO 448.76
+	-----+	-----+	-----+	-----+	-----+
5 rows in set (0.00 sec)					

Gambar 4.31 Sebelum *Thread* dbWatcher berjalan

```
mysql> select * from SensorValue;  
Empty set (0.01 sec)
```

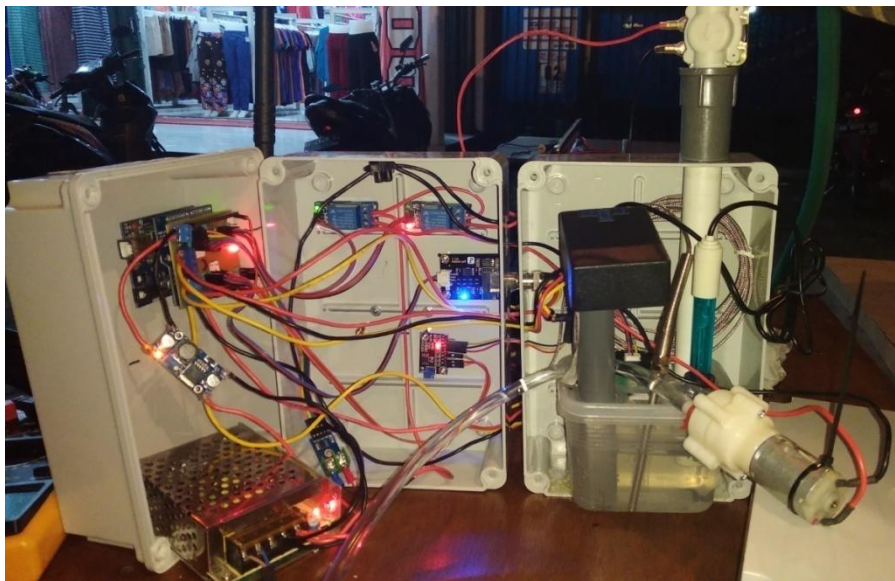
Gambar 4.32 Sesudah *Thread* dbWatcher berjalan

4.3.2 Pembacaan dan pengiriman data

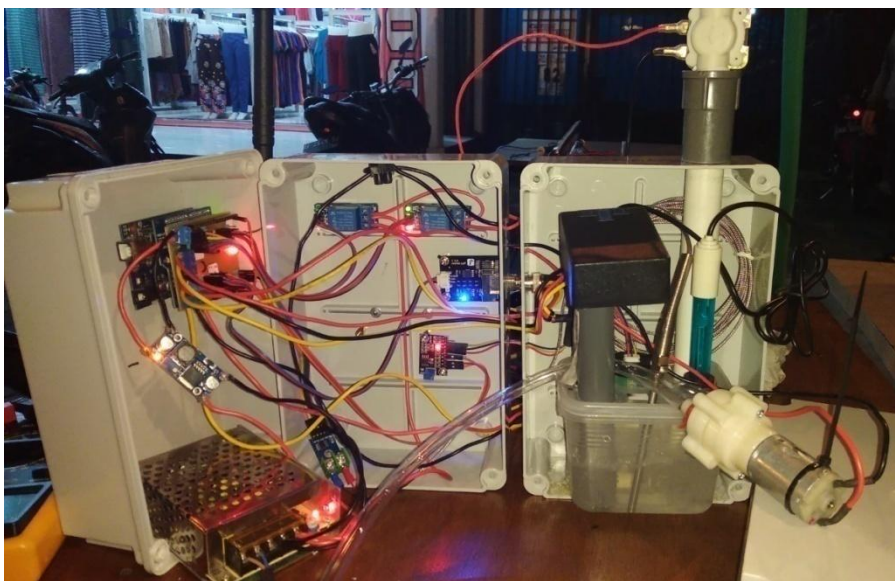
a. Perangkat sensor

```
Dumping Sample 33000  
Filling Sample 41500
```

Gambar 4.33 Output pembuangan dan pengambilan sampel



Gambar 4.34 Perangkat membuang sampel

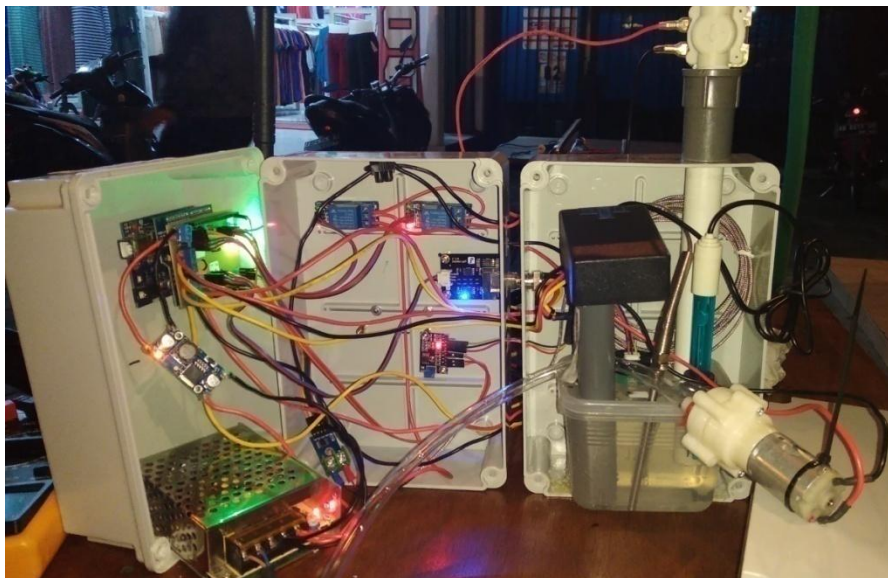


Gambar 4.35 Perangkat mengambil sampel

Gambar 4.33 adalah keluaran perangkat sensor pada tahap dimana perangkat membuang air yang berada pada penampung sampel. Proses pembuangan sampel berjalan selama kurang lebih 8,5 detik. Gambar 4.34 menunjukkan penampakan perangkat saat membuang sampel. Relai di sebelah kiri aktif dan menyalakan pompa air. Setelah seluruh air terbuang, Perangkat sensor akan mengisi air ke dalam penampung sampel hingga ketinggian yang ditentukan. Gambar 4.35 menunjukkan perangkat saat mengisi penampung. Relai di sebelah kanan aktif dan membuka katup solenoid.

```
Start sampling at 41501  
Reading Turbidity  
Turbidity = 0.0  
Reading Temperature  
Temperature = 25.36  
Reading PH  
PH = 7.12  
Reading Conductivity  
Conductivity= 211.76
```

Gambar 4.36 Output pembacaan nilai sensor



Gambar 4.37 Perangkat membaca nilai keluaran sensor

Perangkat sensor selanjutnya membaca nilai keluaran dari sensor seperti yang ditunjukkan Gambar 4.36. Perangkat membaca sensor dengan urutan sensor turbidity, suhu, pH dan konduktivitas. Gambar 4.37 menunjukkan perangkat saat sedang melakukan proses

pembacaan sensor. Setelah proses selesai, data dikirimkan ke perangkat sentral. Data yang sudah dirata-rata dibentuk menjadi sebuah *string* oleh perangkat sensor ditunjukkan pada Gambar 4.38.

```
Data:{YOG0212,, -7.776706,110.371119,,FL1.00,,TU0.0,,TE25.36,,
PH7.12,,CO211.76}
Done Reading at 56428
Sending Data at 56430
```

Gambar 4.38 Konstruksi dan pengiriman

b. Perangkat sentral

```
DEBUG    17-12-06 13:20:07 SerialWatcher {YOG212,, -
7.776706,110.371119,,FL1.00,,TU0.0,,TE25.36,,PH7.12,,CO211.76}
INFO     17-12-06 13:20:07 SerialWatcher Data received
INFO     17-12-06 13:20:07 SerialWatcher Waiting for data
INFO     17-12-06 13:20:08 Worker 1      Processing
DEBUG    17-12-06 13:20:08 Worker 1      {YOG212,, -
7.776706,110.371119,,FL1.00,,TU0.0,,TE25.36,,PH7.12,,CO211.76}
DEBUG    17-12-06 13:20:08 Worker 1      YOG212
DEBUG    17-12-06 13:20:08 Worker 1      -7.776706,110.371119
DEBUG    17-12-06 13:20:08 Worker 1      [('FL', '1.00'), ('TU', '0.0'),
('TE', '25.36'), ('PH', '7.12'), ('CO', '211.76')]
INFO     17-12-06 13:20:08 Worker 1      Update node info
INFO     17-12-06 13:20:08 Worker 1      Done
```

Gambar 4.39 Data diterima dan disimpan

```
mysql> select * from SensorValue;
+-----+-----+-----+-----+-----+
| id  | date                | nodeId | sensorId | value  |
+-----+-----+-----+-----+-----+
| 11  | 2017-12-06 13:20:08 | YOG0212 | FL       | 0      |
| 12  | 2017-12-06 13:20:08 | YOG0212 | TU       | 0.0    |
| 13  | 2017-12-06 13:20:08 | YOG0212 | TE       | 25.36  |
| 14  | 2017-12-06 13:20:08 | YOG0212 | PH       | 7.12   |
| 15  | 2017-12-06 13:20:08 | YOG0212 | CO       | 211.76 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Gambar 4.40 Data tersimpan pada basis data

Gambar 4.39 adalah keluaran dari *thread* serialWatcher dan *thread* Worker saat data diterima oleh perangkat sentral. *Thread* serialWatcher menerima data dan memasukkan data ke dalam *queue*. Data dalam *queue* diambil oleh *thread* worker yang selanjutnya melakukan pengecekan struktur data. Jika struktur data valid maka data akan di-*parsing*. Data yang telah di-*parsing* kemudian disimpan dalam basis data. Gambar 4.40 menunjukkan isi basis data ketika data berhasil disimpan.

4.3.3 Hasil pengujian fungsi

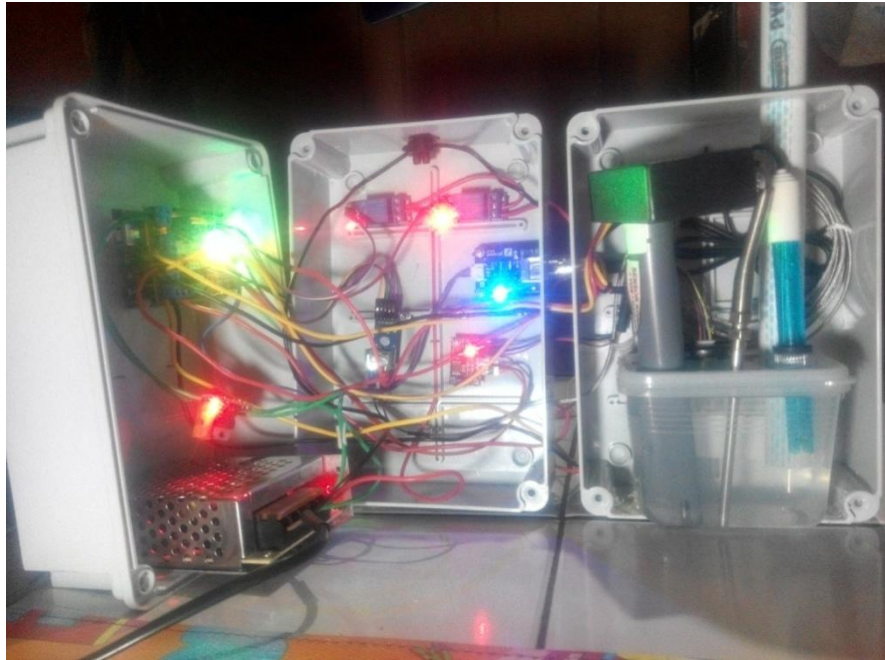
Hasil dari pengujian yang telah dilakukan pada subbab sebelumnya ditunjukkan pada tabel berikut ini:

Tabel 4.1 Hasil pengujian fungsi perangkat

No	Perangkat	Fungsi	Hasil yang diharapkan	Hasil	Keterangan
1	Perangkat sensor	Membuang sampel	Perangkat sensor dapat membuang sampel air	Sesuai	Penampung tidak kosong
2	Perangkat sensor	Mengambil sampel	Perangkat sensor mengambil sampel air pada periode tertentu.	Sesuai	Pipa penampung terhubung dengan sumber air
3	Perangkat sensor	Mengisi penampung	Perangkat sensor mengisi penampung air hingga ketinggian yang ditentukan	Sesuai	
4	Perangkat sensor	Membaca sensor	Perangkat sensor dapat mengambil data air.	Sesuai	
5	Perangkat sensor	Mengirim data	Perangkat sensor dapat mengirimkan data ke perangkat sentral.	Sesuai	Kedua perangkat dalam jangkauan
6	Perangkat sentral	Menerima data	Perangkat sentral dapat menerima data dari perangkat sensor.	Sesuai	Kedua perangkat dalam jangkauan
7	Perangkat sentral	Mengurai data	Perangkat sentral dapat melakukan <i>parsing</i> data dari perangkat sensor.	Sesuai	Data yang di- <i>parsing</i> memiliki struktur data yang sesuai
8	Perangkat sentral	Menyimpan data	Perangkat sentral dapat menyimpan data ke basis data.	Sesuai	Data berhasil di- <i>parsing</i>
9	Perangkat sentral	Menghapus data	Perangkat sentral dapat menghapus data pada periode tertentu.	Sesuai	

4.4 Pengujian sensor

Pengujian sensor dilakukan dengan melakukan pembacaan pada sampel yang memiliki karakteristik yang berbeda. Sensor yang diuji adalah sensor suhu, pH, turbiditas dan konduktivitas karena berkaitan langsung dengan kualitas air. Sampel pertama yang digunakan pada pengujian adalah air mineral (Gambar 4.41). Sampel kedua adalah perasan jeruk yang dicampur dengan air dan gula. Hasil pembacaan disajikan pada Tabel 4.2.



Gambar 4.41 Pengujian sensor pada air mineral

```

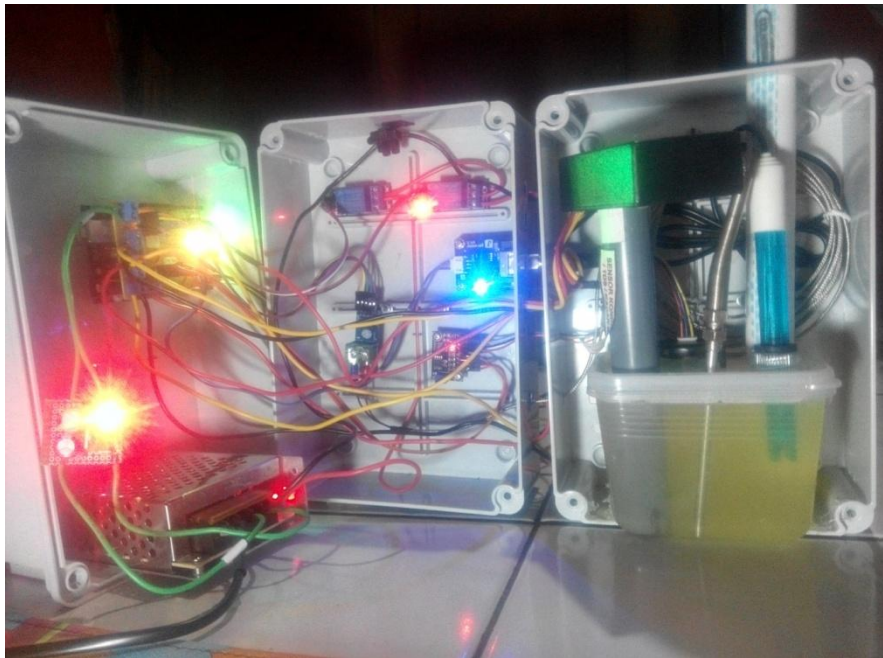
Reading Turbidity
Turbidity = 0.17
Reading PH
PH = 8.03
Reading temperature
Temperature = 28.50
Reading Conductivity
Conductivity = 280.42

```

Gambar 4.42 Hasil pengujian pada air mineral

Hasil pengujian pada air mineral ditunjukkan pada Gambar 4.42. Berdasarkan hasil keluaran sensor, air mineral yang digunakan memiliki turbiditas 0,17%. Sensor pH

menunjukkan bahwa air memiliki pH 8,03. Sensor suhu dan sensor konduktivitas menunjukkan bahwa air memiliki suhu 26,5 °C dan nilai konduktivitas 280,42 uS.



Gambar 4.43 Pengujian sensor pada air perasan jeruk

```
Reading Turbidity
Turbidity = 8.42
Reading PH
PH = 6.05
Reading temperature
Temperature = 34.00
Reading Conductivity
Conductivity = 311.10
```

Gambar 4.44 Hasil pengujian pada air perasan jeruk

Hasil pengujian pada air mineral ditunjukkan pada Gambar 4.44. Berdasarkan hasil keluaran sensor, air perasan jeruk memiliki turbiditas 8,42%. Sensor pH menunjukkan bahwa air memiliki pH 6,05. Sensor suhu dan sensor konduktivitas menunjukkan bahwa air memiliki suhu 34 °C dan nilai konduktivitas 311,10 uS. Perbandingan hasil pembacaan kedua sampel ditunjukkan pada Tabel 4.2.

Tabel 4.2 Hasil pembacaan sampel

No.	Sampel	Suhu	pH	Turbiditas	Konduktivitas
1	Air Mineral	28,5°	8,03	0.17%	280,42 uS
2	Air Jeruk	34°	6,05	8.42%	311,1 uS

4.5 Kelebihan dan kekurangan purwarupa sistem

Purwarupa sistem yang sudah dibuat memiliki beberapa kelebihan dan kekurangan, yaitu:

a. Kelebihan

1. Sistem dapat melakukan pengambilan sampel tanpa bantuan manusia.
2. Pembacaan sifat air dapat dilakukan dalam waktu singkat.
3. Pengolahan data keluaran sensor berjalan secara otomatis.
4. Proses penghapusan data pada basis data berjalan secara otomatis.

b. Kekurangan

1. Sensor yang digunakan belum seakurat sensor yang digunakan pada lab.
2. Beberapa sensor perlu dilakukan penggantian setiap beberapa bulan.
3. Proses pengiriman data masih belum *reliable*.
4. Sifat air yang dapat dibaca masih terbatas pada pH, konduktivitas, turbiditas dan suhu.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat ditarik sebuah kesimpulan yaitu Purwarupa perangkat sensor yang dibuat dengan Arduino UNO dapat melakukan pembacaan beberapa sifat air dengan menggunakan sensor pH, suhu, turbiditas, dan konduktivitas, serta mengirimkan data secara otomatis dan menyimpan hasilnya pada sebuah basis data yang terdapat pada Raspberry Pi sebagai perangkat sentral. Namun sensor yang digunakan terkadang masih mengalami kesalahan pembacaan yang disebabkan oleh beberapa hal seperti gangguan pada daya dan permasalahan pada kabel penghubung sensor. Pengiriman data terkadang masih memiliki permasalahan ketika perangkat berada pada tempat yang memiliki banyak gangguan sinyal.

5.2 Saran

Penelitian yang telah dilakukan masih memiliki banyak kekurangan. adapun saran yang dapat diberikan dari hasil penelitian ini yaitu sebagai berikut:

- a. Pengembang sistem selanjutnya diharapkan dapat mengembangkan sistem dengan menggunakan mikrokontroler yang memiliki lebih banyak pin dan kapasitas RAM yang lebih besar. Arduino varian UNO masih memiliki jumlah pin dan kapasitas RAM yang terbatas. Hal ini menyebabkan pengembangan purwarupa menjadi terbatas. Mikrokontroler dengan jumlah pin yang lebih banyak dan kapasitas RAM yang lebih besar memungkinkan untuk pengembangan lebih lanjut seperti penambahan sensor dan layar untuk menampilkan data.
- b. Pengembang sistem selanjutnya diharapkan dapat memperbaiki kesalahan pada pembacaan sensor dengan menggunakan sumber daya yang stabil serta pemasangan kabel penghubung yang lebih baik.
- c. Masalah gangguan sinyal dapat diatasi dengan perangkat yang menggunakan frekuensi yang berbeda atau antena dengan *gain* lebih besar.
- d. Desain PCB yang digunakan pada purwarupa ini masih memiliki beberapa kesalahan. Pengembang diharapkan dapat memperbaiki kesalahan dan melakukan beberapa peningkatan.
- e. Program utama masih dibuat dengan Python 2.7, pengembang sistem diharapkan dapat melanjutkan pengembangan dengan menggunakan Python 3.

- f. Komunikasi antara perangkat sensor dan perangkat sentral masih hanya satu arah. Pengembang sistem selanjutnya diharapkan dapat mengembangkan komunikasi dua arah sehingga dapat dilakukan sinkronisasi antar perangkat dan konfigurasi perangkat sensor melalui perangkat sentral.

DAFTAR PUSTAKA

- Anonim1, 1990, Peraturan Menteri Kesehatan No. 416/MENKES/PER/IX/1990 tentang Syarat-Syarat Dan Pengawasan Kualitas Air.Arduino. (2017).
- Arduino Uno. Diperoleh 28 Maret dari <https://arduino.co.cc>
- Alawadhi, S., Aldama-Nalda, A., Chourabi, H., Gil-García, J., Leung, S., Mellouli, S., ... & Walker, S. (2012). Building understanding of smart city initiatives. *Electronic government*, 40-53.
- Angel, H. & Wolseley, P. (1992). *The Family of Water Naturalist*. London: Bloomsbury Books.
- Ardiyanto, L., & Sumiharto, R. (2012). Implementasi jaringan sensor nirkabel berbasis XBee studi kasus pemantauan suhu dan kelembaban. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 2(2), 119-130.
- Badan Pusat Statistik. (2016). Kependudukan. Diperoleh 28 Maret dari <https://bps.go.id>
- Budiarso, Z. (2011). SISTEM MONITORING TINGKAT KETINGGIAN AIR BENDUNGAN BEBASIS MIKROKONTROLLER. *Jurnal Dinamika Informatika*, 3(1).
- Corporaal, H. (2006). Embedded system design. *Progress White Papers*, 7-27.
- Effendi, H. (2003). Telaah kualitas air, bagi pengelolaan sumber daya dan lingkungan perairan. Kanisius.
- Evers H.D. (2007) "The End of Urban Involution and the Cultural Construction of Urbanism in Indonesia" *Internationales Asienforum*, vol 38, No. 1-2, pp 51-65.
- Fardiaz, S. (1992). Polusi air dan udara. Kanisius.
- Febriyani, L. D., & Arif, S. (2012). Membangun Embedded Linux pada ALIX3d2 dengan Buildroot pada Aplikasi Stasiun Cuaca. *INKOM Journal*, 5(2), 53-56.
- Ferdoush, S., & Li, X. (2014). Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications. *Procedia Computer Science*, 34, 103-110.
- Hollands, R. G. (2008). Will the real smart city please stand up? Intelligent, progressive or entrepreneurial?. *City*, 12(3), 303-320.
- IEEE 802.15 WPAN Task Group 4 (TG4), diperoleh dari <http://www.ieee802.org/15/pub/TG4.html>
- Mason, C.F. (1993). *Biology of Freshwater Pollution*. 2nd Edition. Longman Scientific and Technical, New York. 351 p.

- Masood, F. (2011). RISC and CISC. arXiv preprint arXiv:1101.5364.
- Murakami Wood, D. (2014, July). Smart City, Surveillance City: Ubiquitous Computing, Big Data and Urban Life. In XVIII ISA World Congress of Sociology (July 13-19, 2014). Isaconf.
- Nam, T., & Pardo, T. A. (2011, June). Conceptualizing smart city with dimensions of technology, people, and institutions. In Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times (pp. 282-291). ACM.
- Ngafifi, M. (2014). KEMAJUAN TEKNOLOGI DAN POLA HIDUP MANUSIA DALAM PERSPEKTIF SOSIAL BUDAYA. *Jurnal Pembangunan Pendidikan: Fondasi dan Aplikasi*, 2(1).
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
- Raspberry Pi. (2017). Raspberry Pi. Diperoleh 3 April dari <https://raspberrypi.org>
- Singh, A. K. (2008). *Microcontroller and Embedded System*. New Age International.
- Weiser, M. (1999). Some computer science issues in ubiquitous computing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3 (July), 12. doi:10.1145/329124.329127.
- Xia, F., Yang, L. T., Wang, L., & Vinel, A. (2012). Internet of things. *International Journal of Communication Systems*, 25(9), 1101.

LAMPIRAN

a. File awm.conf

```
# location = YOG
#       Specify Central node location

# runWatcher = True
#       Run watcher thread for old data auto removal

# workerThreads = 2
#       Specify number of worker threads to run. Value must > 0

# debug = True
#       Show debug data on terminal

# device = /dev/ttyUSB0
#       Serial device port

# baudrate = 9600
#       Serial baudrate

# oldData = 3D
#       Specify the time for data to be considered old
#       Data will be removed when older than specified time
#       m for minutes e.g 10m
#       h for hour e.g 2h
#       D for Days e.g 1D
#       M for Months e.g 3M
#       for second write the number without suffix e.g 10

# hostname = 127.0.0.1
#       Specify database host

# username = root
#       Specify the db login user

# passwd = root
#       Specify the db login pass

# dbname = WaterMonitoring
```

```
#      Name of database to use
```

```
[Global]
```

```
location = YOG
```

```
runWatcher = True
```

```
workerThreads = 2
```

```
debug = True
```

```
[Serial]
```

```
device = /dev/ttyAMA0
```

```
baudrate = 9600
```

```
[Database]
```

```
olddata = 8D
```

```
hostname = localhost
```

```
username = root
```

```
password = elluna
```

```
dbname = Water Monitoring
```