BAB I

PENDAHULUAN

A. Latar Belakang

Standar air bersih dan sehat berlaku pada Standar No. 2 Menteri Kesehatan Republik Indonesia. 492/MENKES/PER/IV/2010 Persyaratan kualitas air minum, antara lain harus bersih, tidak berasa, tidak keruh, tidak beracun, tidak berbau, memenuhi batasan jumlah padatan terlarut, dan bebas dari zat kimia yang berlebihan. Peraturan menetapkan bahwa kekeruhan maksimum air minum adalah 5 NTU (*Nephelometric Turbidity Unit*) dan tingkat maksimum padatan terlarut dalam air adalah 500 PPM (*Part Per Million*). Cara paling umum untuk mendeteksi parameter kualitas air adalah dengan mengambil sampel secara manual dan mengirimkannya ke laboratorium untuk dideteksi dan dianalisis. Metode ini membutuhkan banyak waktu, ataupun memiliki waktu analisis yang lama.

Seiring dengan berkembangnya teknologi sensor dan sistem *monitoring* berbasis data, terdapat peluang untuk mengembangkan sistem monitoring yang dapat memantau secara *real-time* tingkat PH dan kekeruhan air dengan menggunakan sensor digital yang terintegrasi. Sistem ini tidak hanya memberikan data yang akurat, tetapi juga dapat digunakan untuk mempercepat proses penentuan kelayakan konsumsi air tanpa bergantung pada laboratorium konvensional. Dengan menggunakan metode pemantauan berbasis sensor,

hasil yang didapatkan dapat lebih cepat dan lebih mudah diakses, sehingga dapat digunakan untuk pengambilan keputusan yang lebih tepat dan efisien.

Maka kiranya diperlukan sebuah alat yang mampu mendeteksi, menentukan tingkat kelayakan air itu dibutuhkan, dimana rancangan monitoring ini yang menggunakan sensor PH (Sensor PH SEN0161-V2) dan sensor kekeruhan (Sensor *DFRobot*-SEN0189) dengan *mikrokontroler* ESP 32 diharapkan dapat menjadi salah satu pengembangan dengan penggunaan dan pemanfaatan *IOT* yang sifatnya memudahkan, cepat dan efisien dalam mendeteksi kualitas air yang akan di distribusikan sehingga mampu didapatkan air yang berkategori aman untuk konsumennya sedini mungkin.

B. Rumusan Masalah

Adapun rumusan masalah yang akan dibahas adalah "Bagaimana membangun sistem *monitoring* tingkat PH dan kekeruhan air untuk menentukan kelayakan tingkat konsumsi air ?".

C. Tujuan Penelitian

Adapun tujuan penelitian dari proposal ini yakni, sebagai berikut :

1. Penggunaan Sensor PH SEN0161-V2dan sensor DF*Robot*-SEN0189 dengan *microkontroler* ESP 32 untuk menentukan kelayakan konsumsi air.
2. Memberikan informasi secara realtime dalam monitoring bermediakan *LCD* dan Aplikasi *Android*.
3. Merancang aplikasi *android* yang dapat *memonitoring* tingkat ph dan kekeruhan air.

D. Batasan Masalah

1. Rancangan ini tidak memuat dan terikat pada administrasi pelanggan.
2. Aplikasi *android* yang rancang hanya sebagai sistem monitoring tidak untuk mengontrol kerja alat.
3. Rancangan alat ini tidak membaca jumlah volume air.

E. Manfaat Penelitian

Adapun manfaat penelitian dari proposal ini, yaitu :

1. Manfaat bagi penulis

   Menambah ilmu dan pengetahuan mengenai IOT (*internet of things*) serta mampu untuk mengimplementasikan penerapan IOT sebagai pengejewantahan pengetahuan yang didapat selama berstatus mahasiswa dan sejalan dengan penulisan proposal ini.

2. Manfaat bagi pengguna

   Memberikan informasi dengan memonitoring tingkat ph dan kekeruhan air untuk melihat tingkat kelayakan konsumsi air.

3. Manfaat bagi pembaca

   Memberikan edukasi seputar penerapan *IOT* terkhusus dalam perancangan alat yang mampu membaca nilai ph dan kekeruhan air

4. Manfaat bagi Instansi

   Penelitian ini berguna untuk sekolah-sekolah dimana sekolah menjadi salah satu pengonsumsi air dengan kapasitas yang cukup besar dengan penggunaan harian seluruh *stakeholder* sekolah maka dengan penggunaan

rancangan ini sekolah dapat mengetahui kualitas air untuk digunakan secara umum disekolah.

F. Sistematika Penulisan

Adapun sistem penulisan pada proposal ini sebagai berikut :

1. Bab I : Pendahuluan

   Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan.

2. Bab II : Tinjauan Pustaka

   Bab ini berisi tentang kajian teori yang memuat refensi penunjang yang menjelaskan tentang definisi dan fungsi dari perangkat-perangkat yang digunakan dalam pembuatan tugas akhir, bab ini juga memuat kajian penelitian terdahulu yang beresensikan poin-poin kajian sebelumnya sebagai referensi dan pembanding dalam pembuatan tugas akhir ini.

3. Bab III : Metode Penelitian

   Bab ini memuat poin-poin berupa lokasi dan waktu, jenis penelitian, alat dan bahan, rancangan sistem, dan metode pengumpulan data.

4. Bab IV : Hasil dan Pembahasan

   Bab ini menjelaskan mengenai analisis sistem, implementasi perangkat, rancangan aplikasi *android* dan mengenai metode pengujian serta teknik pengujian.

5. Bab V : Penutup

   Bab ini menjelaskan mengenai kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Internet of Things

IOT muncul sebagai isu besar di *Internet*. Diharapkan bahwa miliaran hal fisik atau benda akan dilengkapi dengan berbagai jenis sensor terhubung ke internet melalui jaringan serta dukungan teknologi seperti tertanam sensor dan aktualisasi , *frekuensi radio Identifikasi* (RFID), jaringan *sensor nirkabel*, *real-time* dan layanan *web*, *IOT* sebenarnya *cyber* fisik sistem atau jaringan dari jaringan. Dengan jumlah besar hal / benda dan sensor / aktuator yang terhubung ke *internet*, besar-besaran dan dalam beberapa kasus aliran data *real-time* akan otomatis dihasilkan oleh hal-hal yang terhubung dan sensor. Dari semua kegiatan yang ada dalam *IOT* adalah untuk mengumpulkan data mentah yang benar dengan cara yang efisien, tapi lebih penting adalah untuk menganalisis dan mengolah data mentah menjadi informasi lebih berharga.

Internet of Things dalam penerapannya juga dapat *mengidentifikasi*, menemukan, melacak, memantau objek dan memicu *event* terkait secara otomatis dan real time, Pengembangan dan penerapan komputer, Internet dan teknologi informasi dan komunikasi lainnya (TIK) membawa dampak yang besar pada masyarakat manajemen ekonomi, operasi produksi, sosial manajemen dan bahkan kehidupan pribadi.

Internet of Things menggunakan beberapa teknologi yang secara garis besar di gabungkan menjadi satu kesatuan diantaranya sensor sebagai pembaca data, koneksi internet dengan bebarapa macam topologi jaringan, *radio frequency identification* (RFID), wireless sensor network dan teknologi yang terus akan bertambah sesuai dengan kebutuhan (C. Wang et al., 2013).

2. Monitoring

Monitoring, dalam bahasa Indonesia dikenal dengan istilah pemantauan. *Monitoring* merupakan sebuah kegiatan untuk menjamin akan tercapainya semua tujuan organisasi dan manajemen. Dalam kesempatan lain, *monitoring* juga didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, *mengidentifikasi* masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan. Dengan kata lain, *monitoring* merupakan salah satu proses didalam kegiatan organisasi yang sangat penting yang dapat menentukan terlaksana atau tidaknya sebuah tujuan organisasi. Tujuan dilakukannya *monitoring* adalah untuk memastikan agar tugas pokok organisasi dapat berjalan sesuai dengan rencana yang telah ditentukan.

Internet of Things menggunakan beberapa teknologi yang secara garis besar di gabungkan menjadi satu kesatuan diantaranya sensor sebagai pembaca data, koneksi *internet* dengan bebarapa macam topologi jaringan, *radio*

frequency identification (RFID), *wireless sensor network* dan teknologi yang terus akan bertambah sesuai dengan kebutuhan (Apri Junaidi, 2015).

3. Kekeruhan

Kekeruhan (*turbidity*) adalah keadaan dimana transparansi suatu zat cair berkurang akibat kehadiran zat-zat tak terlarut. Sebagaimana kita ketahui, air keruh merupakan salah satu ciri air yang tidak bersih dan tidak sehat. Pengkonsumsian air keruh dapat mengakibatkan timbulnya berbagai jenis penyakit seperti cacingan, *diare* dan penyakit kulit. Air bersih sangat dibutuhkan khususnya daerah perkotaan yang menggunakan fasilitas PDAM (Perusahaan Daerah Air Minum) untuk sumber air bersih (Fajri Rachmansyah, Satrio Budi Utomo, Sumardi, 2014).

4. PH

PH adalah jumlah *konsentrasi ion Hidrogen* (H+) pada larutan yang menyatakan tingkat keasaman dan kebasaan yang dimiliki. PH merupakan besaran fisis dan diukur pada skala 0 sampai 14. Bila PH < 7 larutan bersifat asam, PH > 7 larutan bersifat basa dan PH = 7 larutan bersifat netral . Pengukuran pH biasanya dilakukan dengan menggunakan PH meter (Muchamad Ngafifuddin,Dkk, 2017).

Pada prinsipnya pengukuran suatu PH adalah didasarkan pada potensial elektro kimia yang terjadi antara larutan yang terdapat didalam *elektroda* gelas (*membrane g*elas) yang telah diketahui dengan larutan yang terdapat diluar elektroda gelas yang tidak diketahui. Hal ini dikarenakan lapisan tipis dari gelembung kaca akan berinteraksi dengan *ion hidrogen* yang ukurannya relatif

kecil dan *aktif*, *elektroda* gelas tersebut akan mengukur *potensial elektrokimia* dari *ion hidrogen* atau diistilahkan dengan *potential of hidrogen* (Fanny Astria,Dkk, 2014).

PH pada air juga dapat mengalami kenaikan dan penurunan.Salah satu komponen penurun ph adalah pasir silica yang berfungsi sebagai penyaring pertikel kotoran halus dan juga penurun PH pada air, selain dari itu daun ketapang juga dapat menurunkan nilai ph pada air (Marco Johanes Steward Pinangkaan,Dkk, 2019).

Skala PH bukanlah skala *absolut*. Beberapa dampak kesehatan jika kadar PH air tidak seimbang adalah keseimbangan keasaman dan alkalinitas tubuh, mempertahankan tingkat *elektrolit*, dan pH yang rendah kurang dari 7 (netral) maka akan dapat mengakibatkan air tidak stabil dan mengalami perubahan warna, bau dan rasa (Jufriadi Karangan,Dkk, 2019).

5. Turbidity sensor

Turbidity meter adalah alat yang digunakan sebagai alat uji standar untuk mengetahui tingkat kekeruhan air. Keberadaan alat ini sebenarnya sudah umum dan mudah dicari. Namun, karena harganya relative mahal menjadikan alat ini hanya dimiliki oleh pihak–pihak tertentu. Untuk menguji apakah air yang kita punya mempunyai standar atau tidak harus pergi ke Laboratorium pengujian air minum, hal ini menyebabkan kurang efektif dan efisien. Dasar pembacaan kekeruhan air itu, di alat sensor tersebut ada sejenis sensor sumber cahaya dan penangkap cahaya, yang kemudian dilewatkan ke bagian air yang akan di lakukan pengukuran atau pengecekan kekeruhan. Sensor ini bisa kita

hubungkan ke perangkat pengolah *instrument* pengukuran seperti ke *mikrokontroler* ataupun ke *arduino*. Agar dapat digunakan untuk pembacaan data sensor melalui ADC, pada sensor kekeruhan ini diperlukan dua buah *resistor* yang di hubungkan di pin 2 dan pin 3. Dan kita juga membutuhkan kabel penghubung dari sensor menuju *unit processing* / *mikrokontroller* / *arduino*. Berikut bentuk fisik Sensor kekeruhan air *GE Turbidity* (Annga Satria Wicaksan, 2018).



Gambar 2. 1 Sensor *Turbidity DF Robot*-SEN0189

Tabel 2. 1 *Spesifikasi DF Robot*-SEN0189

Parameter	Spesification
Operating Voltage	5V DC
Operating Current	40mA (*MAX*)
Response Time	<500ms
Insulation Resistance	100M (*Min*)
Operating Temperature	5℃~90℃
Storage Temperature	-10℃~90℃
Weight	30g

6. Sensor PH *Meter*

Sebuah PH *meter* terdiri dari sebuah *elektroda* (*probe* pengukur) yang terhubung ke sebuah alat *elektronik* yang mengukur dan menampilkan nilai

PH. Prinsip kerja utama PH *meter* adalah terletak pada sensor *probe* berupa *elektrode* kaca dengan cara mengukur jumlah *ion* H3O+ di dalam larutan. Ujung *elektrode* kaca adalah lapisan kaca setebal 0.1 mm yang berbentuk bulat (*bulb*). *Bulb* ini dipasangkan dengan *silinder* kaca non *konduktor* atau plastik memanjang. Inti sensor PH terdapat pada permukaan *bulb* kaca yang memiliki kemampuan untuk bertukar *ion* positif (H+) dengan larutan terukur (Al Qalit,Dkk, 2017).



Gambar 2. 2 Sensor PH SEN0161-V2.

Tabel 2. 2 Spesifikasi Sensor PH SEN0161-V2.

Parameter	Spesifikasi
Tegangan kerja antara	3.3 ~ 5.5V
Output tegangan *analog*	0 ~ 3.0V
Jenis konektor *probe*	*BNC*
Tingkat akurasi pengukuran	±0.1
Signal Connector	PH2.0-3P
Dimensi *board*	42mm x 32mm
Range deteksi pH	0 ~ 14
Suhu kerja	5 ~ 60 ° C
Titik netral	pH 7 ± 0,5
Internal Resistance	< 250MΩ
Panjang kabel *probe*	100cm

7. *Mikrokontroler* ESP32

ESP 32 adalah *mikrokontroler* yang dikenalkan oleh *Espressif System* merupakan penerus dari *mikrokontroler* ESP8266. Pada *mikrokontroler* ini sudah tersedia modul *WiFi* dalam *chip* sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things* (Muliadi, dkk, 2020).



GAMBAR 2. 3 ESP 32

TABEL 2. 3 SPESIFIKASI ESP 32

Parameter	Spesifikasi
Tegangan	3.3 Volt
CPU	*Xtens a Dual Core* LX6-160m Hz
Arsitektur	32 Bit
Flash Memory	16 MB
SRAM	512 KB
GPIO Pin	36(18/2)
Bluetooth	Tersedia
WiFi	Tersedia
SPI/12c/UAR	4/2/2

8. Pompa Air

Pompa adalah mesin atau peralatan mekanis yang digunakan untuk menaikkan cairan dari dataran rendah ke dataran tinggi atau untuk mengalirkan

cairan dari daerah bertekanan rendah ke daerah yang bertekanan tinggi dan juga sebagai penguat laju aliran pada suatu sistem jaringan perpipaan.

Prinsip kerja pompa adalah dengan melakukan penekanan dan penghisapan terhadap *fluida*. Pada sisi hisap pompa (*suction*), elemen pompa akan menurunkan tekanan dalam ruang pompa sehingga akan terjadi perbedaan tekanan antara permukaan *fluida* yang dihisap dengan ruang pompa.



Gambar 2. 4 Pompa air DC12V 240L

Tabel 2. 4 Spesifikasi pompa air DC12V 240L

Parameter	Spesifikasi
Tegangan Kerja	**12** VDC
Kecepatan Pompa	240 L/h
Daya Angkat Maksimum	300 cm
Tingkat Kebisingan	< 35 dB
Diameter Luar	9 mm
Diameter Dalam	6 mm
Kemampuan Tahan Air	IP68
Dimensi	52 x 47 mm

9. Relay

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (*Coil*) dan Mekanikal (seperangkat Kontak Saklar/*Switch*). *Relay* menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.

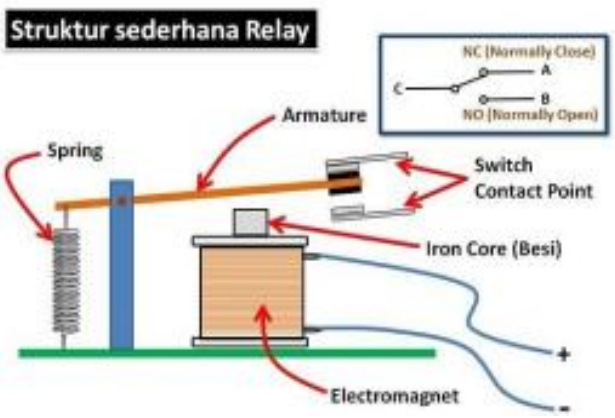
Kontak Poin (*Contact Point*) *Relay* terdiri dari 2 jenis yaitu , *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *CLOSE* (tertutup). *Normally Open* (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *OPEN* (terbuka) Karena *Relay* merupakan salah satu jenis dari Saklar, maka istilah *Pole* dan *Throw* yang dipakai dalam Saklar juga berlaku pada *Relay*.

Banyaknya kondisi yang dimiliki oleh sebuah Kontak (*Contact*) Berdasarkan penggolongan jumlah *Pole* dan *Throw-nya* sebuah *relay*, maka *relay* dapat digolongkan menjadi, *Single Pole Single Throw* (SPST) *relay* golongan ini memiliki 4 Terminal, 2 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *Coil*. *Single Pole Double Throw* (SPDT) *relay* golongan ini memiliki 5 Terminal, 3 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *Coil*. *Double Pole Single Throw* (DPST) *relay* golongan ini memiliki 6 Terminal, diantaranya 4 Terminal yang terdiri dari 2 Pasang Terminal Saklar sedangkan 2 Terminal lainnya untuk *Coil*. *Relay* DPST dapat dijadikan 2 Saklar yang dikendalikan oleh 1 *Coil*. *Double Pole Double Throw* (DPDT) :

Relay golongan ini memiliki Terminal sebanyak 8 Terminal, diantaranya 6 Terminal yang merupakan 2 pasang *Relay* SPDT yang dikendalikan oleh 1 (*single*) *Coil*. Sedangkan 2 Terminal lainnya untuk *Coil* (Muhamad Saleh,Dkk, 2017).



GAMBAR 2. 5 RELAY



Gambar 2. 6 Struktur Relay

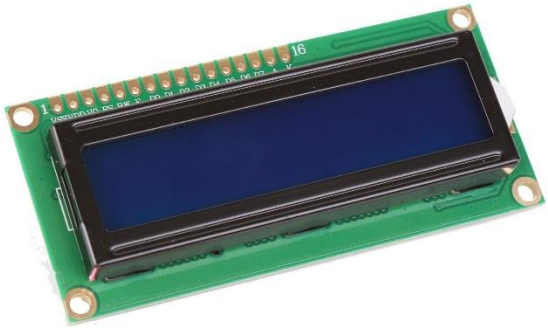
Tabel 2. 5 Spesifikasi Relay SPST

Parameter	Spesifikasi
upply Voltage	3.75 to 6 V
Supply Current with Relay De-Energized	2 mA
Supply Current with Relay Energized	70 to 72 mA
Input Control Signal	Active High
Input Control Signal Current	1.5 to 1.9 mA

Relay Max Contact Voltage	250 VAC or 30 VDC
Relay Max Contact Current	10 A
Dimensions	
Length	34 mm (1.50")
Width	26 mm (1.02")
Height	18 mm (0.71")

10. LCD

LCD (Liquid Cristal Display) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi *CMOS logic* yang bekerja dengan tidak menghasilkan cahaya, tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. *LCD (Liquid Cristal Display)* berfungsi sebagai penampil data, baik dalam bentuk karakter, huruf, angka ataupun grafik. *LCD* adalah lapisan dari campuran organik antara lapisan kaca bening dengan *elektroda* transparan *indium oksida* dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Lapisan *sandwich* memiliki *polarizer* cahaya vertikal depan dan *polarizer* cahaya horisontal belakang yang diikuti dengan lapisan *reflektor*. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan (Vike Tiffani Bawotong, 2015).



Gambar 2. 7 LCM 1602B COB LCD 2 x 16

Tabel 2. 6 Spesifikasi LCM 1602B COB LCD 2 x16

Parameter	Spesifikasi
Display Format	16x2
Display Mode	*STN, Transmissive, Negative*
LCD Color	*White; Background: Blue*
Display Type	Character COB
Outline Dimension	80.0(W) x36.0(H) x 13.1 mm
Viewing Area	64.5MIN(W) x15.8MIN(H)mm
Drive IC	ST7066U-0A *English / Japan Font*
Viewing Direction	6 O'*clock*
Backlight	*White*
Driving Method	1/16 *Duty*, 1/5 *Bias*

11. Selenoid

Solenoid valve merupakan katup yang dikendalikan dengan arus listrik baik AC maupun DC melalui kumparan / *selenoida*. *Solenoid valve* ini merupakan elemen kontrol yang paling sering digunakan dalam sistem *fluida*. Seperti pada sistem *pneumatik*, sistem *hidrolik* ataupun pada sistem kontrol mesin yang membutuhkan elemen kontrol otomatis. pada sistem *pneumatik*, *solenoid valve* bertugas untuk mengontrol saluran udara yang bertekanan menuju aktuator *pneumatik* (*cylinder*). Atau pada sebuah tandon air yang

membutuhkan *solenoid valve* sebagai pengatur pengisian air, sehingga tandon tersebut tidak sampai kosong (Rina Latuconsin, dkk, 2017).



Gambar 2. 8 *Selenoid Valve* 1/2" 12V DC

Tabel 2. 7 Spesifikasi *Selenoid Valve* 1/2" 12V DC

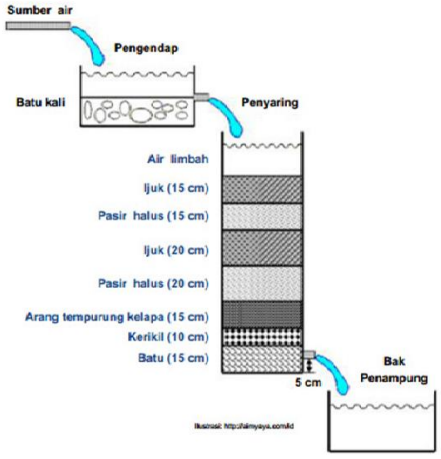
Parameter	Spesifikasi
SKU	APL-1/2-12VDC
Port Size	1/2" Male NPT
Voltage	12V DC
Orifice Size	8.5 mm
Temp Range	32 to 125° F / 0 to 50°C
PSI Range	1.5 - 90 PSI (*Minimum Required*)
Power	6 *Watts*
Coil Connection	*Terminal*
Weight	4.10 oz
Height	2.39
Length	2.85" *port to port*
Width	1.65"

12. *Filter* Air

Filtrasi merupakan proses penjernihan atau penyaringan air melalui media (pada penelitian ini digunakan pasir), dimana selama air melalui media akan terjadi perbaikan kualitas. Hal ini disebabkan adanya pemisahan partikel-partikel *tersuspensi* dan *koloid*, reduksi bakteri dan organisme lainnya dan pertukaran konstituen kimia yang ada dalam air. *Filtrasi* adalah salah satu bentuk untuk menghasilkan *effluent* dengan *efisiensi* tinggi. Faktor yang perlu

diperhatikan untuk menjaga *efisiensi filtrasi* adalah untuk menghilangkan *partikula*t dan *koloidal* yang tidak mengendap setelah *flokulasi* biologis atau kimia ,menaikkan kehilangan *suspensi solid*, kekeruhan, *phospor*, BOD, COD, bakteri dan lain-lain, serta mengurangi biaya desinfektan.

Dalam proses *filtrasi* terdapat kombinasi antara beberapa proses yang berbeda. Proses-proses tersebut meliputi *mechanical straining*, Merupakan proses penyaringan partikel tersuspensi yang terlalu besar untuk dapat lolos melalui ruang antara butiran media. Kemudian sedimentasi Merupakan proses mengendapnya partikel tersuspensi yang berukuran lebih kecil dari lubang pori-pori pada permukaan butiran, Selanjutnya *adsorpsi* Prinsip proses ini adalah akibat adanya perbedaan muatan antara permukaan butiran dengan partikel tersuspensi yang ada di sekitarnya sehingga terjadi gaya tarik-menarik.Kemudian aktifis kimia Merupakan proses dimana partikel yang terlarut diuraikan menjadi *substansi* sederhana dan tidak berbahaya atau diubah menjadi partikel tidak terlarut, sehingga dapat dihilangkan dengan proses penyaringan, *sedimentasi* dan *adsorpsi* pada media berikutnya. Dan aktifis biologi Merupakan proses yang disebabkan oleh aktifitas *mikroorganisme* yang hidup di dalam *filter* (Ajeng Ari Novia, 2019).



Gambar 2. 9 Sistem filter Air

Tabel 2. 8 Komponen filter

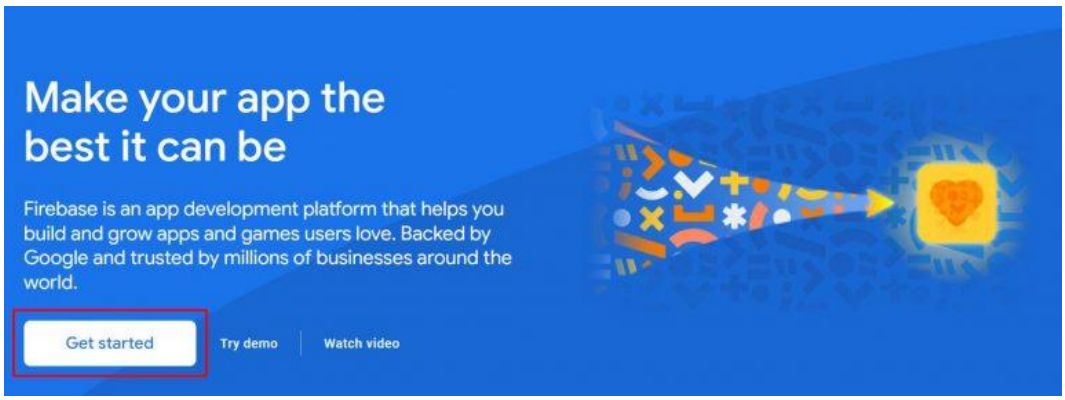
Bahan	Ketebalan
Ijuk	±15 cm
Pasir halus(pasir silica)	±20 cm
Ijuk	±15 cm
Pasir halus	±15 cm
Arang tempurung kelapa	±15 cm
Kerikil	±15 cm
Batu	±15 cm
kapas	±15 cm

13. Firebase

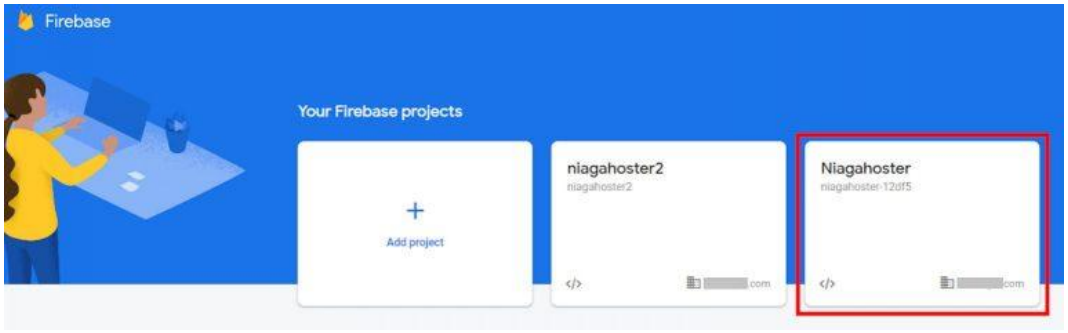
Firebase adalah penyedia layanan *cloud* dengan *back-end* sebagai servis yang berbasis di San Fransisco, California. *Firebase* membuat sejumlah produk untuk pengembangan aplikasi *Mobile* ataupun *web*. *Firebase* didirikan oleh Andrew Lee dan James Tamplin pada tahun 2011 dan diluncurkan dengan *cloud database* secara *realtime* di tahun 2012. Produk utama dari *Firebase* yakni suatu database yang menyediakan API untuk memungkinkan pengembang menyimpan dan *mensinkronisasi* data lewat *multiple client*.

Perusahaan ini diakusisi oleh *Google* pada Oktober 2014 (Anisya Sonita,Dkk, 2018).

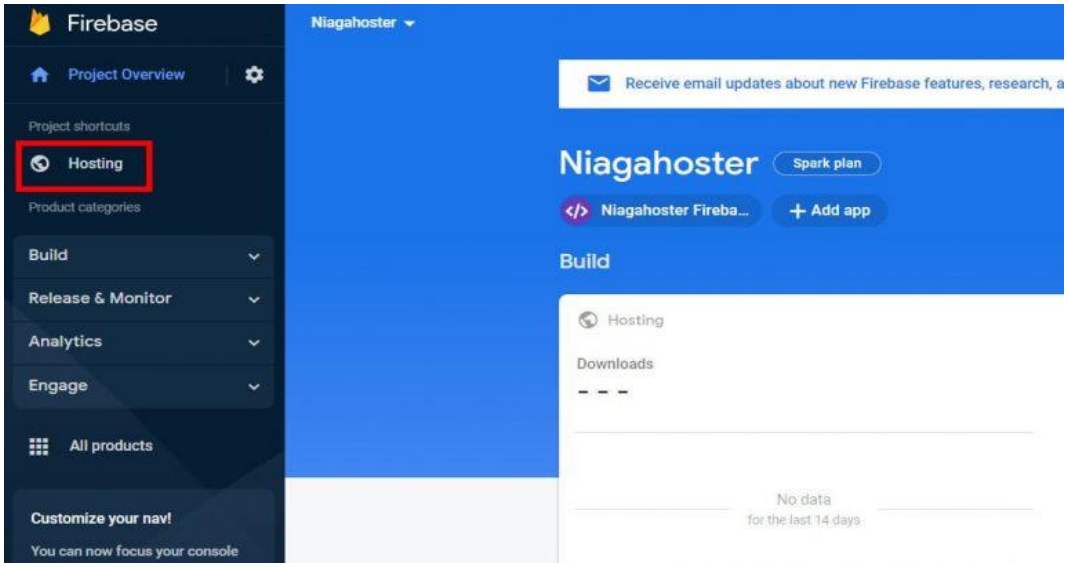
Firebase adalah penyedia layanan *realtime database* dan *backend* sebagai layanan. Suatu aplikasi yang memungkinkan pengembang membuat API untuk *disinkronisasikan* untuk *client* yang berbeda-beda dan disimpan pada *cloud-nya firebase*. *Firebase* memiliki banyak *library* yang memungkinkan untuk m*engintegrasikan* layanan ini dengan *android*, *Ios, Javascript, java, Objetive-C dan NodeJS*. *Database Firebase* juga bersifat bisa diakses lewat *REST API. REST API* tersebut menggunakan *protocol server-sent event* dengan membuat koneksi *HTTP* untuk menerima *push notification* dari *server*. Pengembangan menggunkan *RES* API untuk post data yang selanjutnya *firebase Client library* yang sudah 23 diterapkan pada aplikasi yang dibangun akan mengambil data secara *realtime* (Ahmad Fadil, 2021).



Gambar 2. 10 Halaman Awal Firebase



Gambar 2. 11 Halaman Login Firebase



Gambar 2. 12 Halaman Dashboard Firebase

14. Android

Android merupakan suatu sistem operasi yang berbasis *linux* untuk telepon seluler seperti telepon pintar dan *computer tablet* (Eko Priyo Utomo, 2012). Sederhananya android bisa diartikan sebagai salah satu atau merupakan gabungan dari tiga pengertian yang disebutkan yaitu :

a. Sistem operasi untuk *mobile* bersifat gratis.

b. *Platform* untuk pengembangan aplikasi berbasis *mobile*. Peralatan/hardware, terutama mobile phone yang menjalankan sistem operasi dan aplikasi yang dibuat dengan *platform android*.

Secara lebih spesisfik, *android* dirancang dan mempunyai beberapa bagian yang saling bergantung untuk beberapa keperluan, antara lain :

a. *Kemel sistem operasi linux* yang menyediakan antarmuka untuk tingkat rendah (*low-level*), manajemen memori, dan proses *control*. *Android* dibuat dengan dasar *kernel linux* 2.6.
b. *Hardware* yang didesain untuk memenuhi kebutuhan untuk *mobile device* atau perangkat *mobile*.
c. Dilengkapi dengan *library* yang bersifat *open-source* untuk mengembangkan aplikasi seperti *SQLite*, *Webkit*, *OpenGL* dan media manager.
d. *Framework user-interface* yang digunakan *host* dan *launch* aplikasi.
e. *Framework* aplikasi yang menggunakan sistem *service* ke lapisan aplikasi seperti *window-manager*, *content-provide*, telepon dan sensor.
f. Sebuah *bundle* pengembangan aplikasi yang termasuk didalam *fitur* untuk membuat aplikasi, yaitu berupa *tools, plug-in* dan dokumentasi (Ayu Yudha, 2012).

Setiap pemrograman pasti mempunyai suatu struktur dan metode tertentu untuk menuliskan kode. Dalam pemrograman *android* pembuatan *layout UI (User Interface)* dipisahkan dari kode utama *java*. Pembuatan *UI*

dilakukan pada sebuah *file XML* yang berisikan kode-kode *XML* untuk membangun *UI*.

a. *File Manifest* adalah *file XML* yang *mendeskripsikan* tentang komponen yang digunakan dalam aplikasi. File ini berada pada *root director project android*.
b. *Activity* komponen *File activity* adalah *file java* yang berisikan kelas yang *meng-extends* kelas *activity*. File ini berasal pada direktori *src project android*.
c. *File XML Layout UI* File ini merupakan *file xml* yang digunakan untuk mendesain *UI*. File ini berada pada direktori *res. UI* termasuk *widget android*.
d. Kelas biasa/*library File* ini merupakan file *java* biasa yang berisikan kelas untuk melakukan tugas tertentu. File ini berada pada direktori *assets*.
e. *Resource* Lain *Resource* lain seperti *image*, *font* disimpan pada *directory assets*.

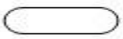
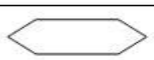
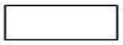
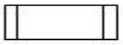
Ada beberapa keuntungan yang akan di dapatkan ketika memutuskan untuk belajar mengenai pemrograman *android*, antara lain .

a. Pasar aplikasi *mobile* yang masih sangat luas.
b. Tidak memerlukan *sertifikasi* untuk menjadi pengembang *android*.
c. Adanya *android market/ google play* untuk tempat distribusi aplikasi yang dibuat sehingga tidak terlalu sulit bagi pengembang yang akan menjual aplikasi yang telah dibuat.

d. *Open-platform*, artinya tidak bergantung pada satu *manufaktur/vendor* tertentu saja sehingga dapat digunakan oleh berbagai macam *vendo*.

15. *Flowchart*

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan *algoritma* atau langkah-langkah *instruksi* yang berurutan dalam sistem. seorang *analis* sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada *programmer* (Rizqi Rosaly, Andy Prasetyo, 2019).

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Gambar 2. 13 simbol *flowhart*

16. UML (Unified Modelling Language)

UML (Unified Modeling Language) merupakan pengganti dari metode *analisis* berorientasi *object* dan *design* berorientasi *object* (*OOAD&D/object*

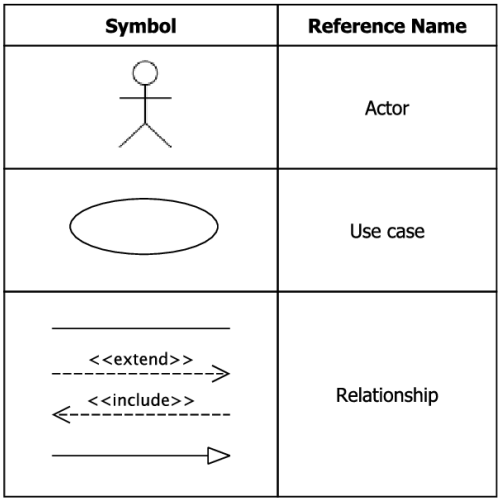
oriented analysis and design) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. *UML* merupakan gabungan dari metode *Booch,Rumbaugh (OMT)* dan *Jacobson*. Tetapi *UML* mencakup lebih luas daripada *OOAD*. Pada pertengahan saat pengembangan *UML*, dilakukan standarisasi proses dengan *OMG(Object Management Group)* dengan harapan *UML* bakal menjadi bahasa standar pemodelan pada masa yang akan datang (yang sekarang sudah banyak dipakai oleh berbagai kalangan).

UML digunakan untuk memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep *berorientasi object*. Dan juga untuk menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

UML adalah bahasa untuk *menspesifikasi*, m*emvisualisasi*, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, *deskripsi* atau perangkat lunak)dari sistem perangkat lunak,seperti pada pemodelan bisnis dan sistem *non* perangkat lunak lainnya. Selain itu *UML* adalah bahasa pemodelan yang menggunakan konsep *orientasi object*.*UML* dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corps*. *UML* menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai prespetktif. *UML* tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

a. Use Case Diagram

Use case diagram adalah satu dari berbagai jenis diagram *UML* (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya.

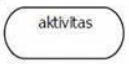
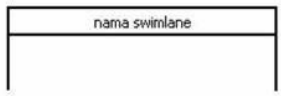


Gambar 2. 14 simbol pada use case diagram

b. Activity Diagram

Activity diagram, dalam bahasa Indonesia diagram *aktivitas*, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur *aktivitas*. Alur atau *aktivitas* berupa bisa berupa runtutan menu-menu atau proses bisnis yang terdapat di dalam sistem tersebut. Dalam buku Rekayasa Perangkat Lunak karangan Rosa A.S mengatakan, “Diagram *aktivitas* tidak menjelaskan kelakuan aktor. Dapat diartikan bahwa dalam

pembuatan *Activity Diagram* hanya dapat dipakai untuk menggambarkan alur kerja atau *aktivitas* sistem saja.

Simbol	Deskripsi
status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / *decision* 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / *join*	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
fork, 	digunakan utk menunjukkan kegiatan yg dilakukan secara paralel
join, 	digunakan utk menunjukkan kegiatan yg digabungkan

Gambar 2. 15 Simbol *activity diagram*

c. Class Diagram

Class diagram atau *diagram* kelas adalah salah satu jenis *diagram* struktur pada *UML* yang menggambarkan dengan jelas struktur serta *deskripsi class*, *atribut*, metode, dan hubungan dari setiap objek. Ia bersifat *statis*, dalam artian *diagram* kelas bukan menjelaskan apa yang terjadi jika kelas kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi. *Diagram* kelas ini sesuai jika diimplementasikan ke proyek yang

menggunakan konsep *object-oriented* karena gambaran dari *class diagram* cukup mudah untuk digunakan.

d. Sequence Diagram

Sequence diagram atau *diagram* urutan adalah sebuah *diagram* yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagaram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek - objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan (Agung Feby Prasetya, 2022).

17. HTML

HyperText Markup Language (HTML) adalah bahasa *web*. Saat kita membuka halaman *web* di *browser web* seperti *Internet Explorer*, *Firefox*, atau Safari, *browser* akan mengunduh dan menampilkan *HTML*. Pada intinya, *HTML* hanyalah sebuah dokumen, sama seperti dokumen yang dibuat di *word*. Program seperti *Microsoft Word* digunakan untuk melihat dokumen pengolah kata karena *microsoft word* tahu cara membaca dan menampilkan teks. Demikian juga dengan *web*, *browser web* adalah program yang mengetahui cara membaca dan menampilkan dokumen yang dibuat dengan *HTML*. Dokumen *Word Process* dapat dibuat dan dibaca dengan satu program. Di sisi lain, dokumen *HTML* membutuhkan program yang berbeda untuk pembuatan dan pembacaan; Kita tidak dapat membuat dokumen *HTML* menggunakan browser. Untuk membuat dokumen *HTML* kita membutuhkan program yang disebut *editor*. *Editor* ini bisa sesederhana program *Notepad* yang disertakan

dengan *Microsoft Windows* atau *serumit Eclipse* dan *Microsoft Visual Studio*. Umumnya kita dapat menggunakan program yang sama untuk membuat dokumen *HTML* yang digunakan untuk membuat program *PHP*. Dokumen *HTML* dapat disimpan di folder *Documents* dalam komputer. Namun jika dibuka, hanya kita saja yang bisa melihat isinya. Untuk menyiasati hal ini, kita memerlukanlebih banyak sumber daya, sebut saja sebagai *server web*. Menyimpan dokumen di *server web* memungkinkan orang lain untuk dapat melihat dokumen kita. *Server web* adalah komputer yang menjalankan perangkat lunak khusus yang mengetahui cara mengirim (atau menyajikan) halaman *web* ke banyak orang pada saat yang bersamaan. Dokumen *HTML* diatur dalam urutan tertentu, dengan bagian-bagian tertentu. Mereka terstruktur sedemikian rupa agar *browser web* dapat membaca dan menampilkannya. Karena hal tersebut, ketika kita membuatdokumen *HTML*, kita harus mengikuti strukturnya dan menyiapkan dokumen agar *browser* dapat membacanya.

Elemen	*Deskripsi*	*Jenis penggunaan*
<a>	Anchor	Membuat link ke halaman lain atau bagian dari dokumen yang sama.
 	Jeda baris	Memasuki jeda baris atau karakter kembali.
	Bagian dari halaman	Membuat area keseluruhan atau pembagian logis pada halaman, seperti bagian heading/menu, area konten, atau footer.
<form>	Form web	Membuat formulir web untuk menerima input pengguna.
<h1> hingga <h6>	Heading	Membuat wadah untuk judul, seperti teks judul.
<hr>	Hard rule	Membuat garis horizontal.
<img>	Image	Wadah untuk gambar.
<input>	Input	Elemen untuk menerima input pengguna.
<link>	Link sumber	Tautan ke sumber daya untuk halaman tersebut; jangan bingung dengan elemen Anchor
<p>	Paragraf dalam halaman	Membuat paragraf tekstual atau area dan wadah lain untuk teks.
<script>	Tag skrip	Menunjukkan skrip web atau program. Juga sering ditemukan di bagian kepala.
	Span	Membuat wadah untuk elemen. Sering digunakan bersama dengan informasi gaya.

Gambar 2.16 Elemen dan penggunaannya

18. Arduino IDE

IDE merupakan singkatan dari *Integrated Developtment Enviroenment*, atau secara bahasa mudahnya merupakan lingkungan *terintegrasi* yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah *Arduino* dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui *sintaks* pemrograman. *Arduino* menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman *Arduino* (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, *IC mikrokontroler Arduino* telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler Arduino* dengan *mikrokontroler*. *Arduino IDE* dibuat dari bahasa pemrograman

JAVA. *Arduino IDE* juga dilengkapi dengan *library C/C++* yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. *Arduino IDE* ini dikembangkan dari *software Processing* yang dirombak menjadi *Arduino IDE* khusus untuk pemrograman dengan *Arduino*.

Program yang ditulis dengan menggunaan *Arduino Software (IDE)* disebut sebagai *sketch*. *Sketch* ditulis dalam suatu *editor* teks dan disimpan dalam *file* dengan *ekstensi ino*. Teks *editor* pada *Arduino Software* memiliki fitur seperti *cutting/paste* dan *seraching/replacing* sehingga memudahkan kamu dalam menulis kode program.

Pada *Software Arduino IDE*, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan *Sotware Arduino IDE*, menunjukan *board* yang terkonfigurasi beserta *COM Ports* yang digunakan. Terdapat juga *serial monitor*. *Serial monitor* disini merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara *arduino* dengan *sketch* pada *port serialnya*. *Serial Monitor* ini sangat berguna sekali ketika kamu ingin membuat program atau melakukan *debugging* tanpa menggunakan *LCD* pada *Arduino*. *Serial monitor* ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan *error*.



Gambar 2.17 *Arduino IDE*

19. Java

Java adalah sebuah bahasa pemrograman *scripting* yang sering digunakan dalam pembuatan aplikasi berbasis *handphone* dan juga dapat digunakan untuk menyediakan akses objek yang disisipkan di aplikasi lain. *Java* berfungsi sebagai penambah tingkah laku agar *widget* dapat tampil lebih *atraktif* (Achmad Fikri Sallaby, Dkk, 2015).

Bahasa pemrograman *Java* pertama kali muncul dari sebuah project "*The Green Project*" di *Sun Microsystem*, sebuah perusahaan perangkat lunak di Amerika.

Proyek itu dimotori oleh James Gosling, Patrick Naughton, Mike Sheridan, dan Bill Joy. Awalnya, proyek bertujuan untuk menciptakan sebuah peralatan pintar. Namun, karena tak puas dengan hasil dari bahasa pemrograman C++ dan C, mereka memutuskan untuk membuat bahasa pemrograman sendiri yang lebih canggih lagi.

Setelah 18 bulan proyek berjalan, James Gosling akhirnya berhasil menciptakan bahasa pemrograman baru yang dinamai Oak.

Sayangnya, nama Oak ternyata telah digunakan perusahaan Oak Technology. Sehingga, tahun 1995 Oak diganti menjadi Java yang dikabarkan terinspirasi oleh kopi Jawa. Oleh sebab itu, logo bahasa pemrograman Java berupa secangkir kopi.

20. C++

Bahasa Pemrograman C++ adalah bahasa Pemrograman Komputer Tingkat Tinggi (*High Level Language*), tapi C++ juga dimungkinkan untuk menulis Bahasa Pemrograman Tinggkat Rendah (*Low Level Language*) di dalam pengkodingan karena C++ merupakan peluasan dari Bahasa Pemrograman C yang tergolong dalam Bahasa Pemrograman Tingat Menengah (*Middle Level Language*), yang berarti Bahasa Pemrograman C++ memiliki semua fitur dankelebihan yang bahasa pemrograman C miliki, termasuk kelebihan Bahasa C yaitu kita dimungkinkan untuk menggunakan Bahasa Pemrograman *Assembly* di dalam pengkodingan C++, dan juga menyediakan fasilitas untuk memanipulasi memori tingkat rendah (Ilmi Ramadhana, 2018).

B. Kajian Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk mendapatkan bahan *referensi* serta bahan perbandingan dan acuan. Maka peneliti akan mencantumkan beberapa hasil penelitian terdahulu sebagai berikut :

Penelitian terdahulu dilakukan oleh Fajri Rachmansyah, Dkk (2014) dengan judul "Perancangan dan Penerapan Alat Ukur Kekeruhan Air Menggunakan Metode Nefelometrik Pada Instalasi Pengolahan Air Dengan *Multi Media Card* (Mmc) Sebagai Media Penyimpanan (Studi Kasus Di Pdam Jember)". Pada penelitian ini Hasil penelitian yang pertama didapatkan dari hasil pengujian sensor intensitas cahaya yang dibuat.Pada sensor intensitas ini menggunakan *led* berwarna merah sebagai sumber cahaya dan menggunakan fotodioda sebagai penerima cahaya.Ketika sensor mendeteksi air jernih maka fotodioda akan lebih banyak menerima pancaran cahaya dari led sehingga akan mempengaruhi tegangan dari fotodioda dalam mendeteksi tingkat kekeruhan air yang berbeda-beda. Untuk kemudian *output* yang di tampilkan berupa kolom pertama berisi data waktu RTC, kolom kedua berisi nilai ADC dari sensor sedangkan kolom ketiga berisi nilai kekeruhan air dalam satuan NTU (*Nephelometric Turbidity Unit*). Persamaan pada penelitian ini dengan penelitian yang akan di jewantahkan pada proposal ini adalah kesamaan *index* yang ingin di baca yakni tingkat kekeruhan namun, tentu terdapat juga perbedaan diantaranya yaitu penggunaan alat dalam mendeteksi kekeruhan, kemudian tambahan pada pembacaan nilai ph dan *output* yang akan di *monitoring*.

Penelitian terdahulu dilakukan oleh Dendy Ramdani, Dkk (2020) dengan judul "Rancang Bangun Sistem *Otomatisasi* Suhu Dan *Monitoring* pH Air *Aquascape* Berbasis IoT (*Internet Of Thing*) Menggunakan *Nodemcu* Esp8266 Pada Aplikasi Telegram". Penelitian ini membahas perancangan

sistem otomatisasi suhu dan *monitoring* PH air *Aquascape* berbasis *internet of thing* menggunakan *nodemcu* esp8266 pada aplikasi telegram. Hasil dari penelitian ini berupa alat yang mampu melakukan proses otomatisasi suhu dengan menggunakan sensor suhu DS18B20 dan *memonitoring* nilai PH menggunakan sensor PH pada *Bot* telegram. Penelitian ini menghasilkan alat yang digunakan untuk otomatisasi suhu dan *monitoring* PH *Aquascape* . Rangkaian bagian dalam antara modul sensor dan *mikrokontroller*, diantaranya terdiri dari *nodemcu*, modul sensor PH, sensor DS18B20 dan Relay. Alat ini berfungsi guna mengotomatisasi lampu dan cooling fan pada *Aquascape* apabila suhu diluar dari batas 25°C-28 °C. Sensor PH akan mengambil data berupa nilai PH pada *Aquascape*, apabila nilai PH yang didapat <6 atau >8 maka akan data akan diproses oleh *nodemcu* untuk diteruskan kepada *Bot* telegram. Pengujian alat ini dengan perbandingan hasil antara sensor suhu dengan termometer dan sensor PH dengan PH *buffer*. Persamaan dengan penelitian yang akan di wujudkan pada proposal penelitian ini adalah penggunaan basis *IOT* dalam *memonitoring* untuk di teruskan keluaran nya pada aplikasi, namun berbeda dalam aplikasi yang akan digunakan sebagai alat untuk menampilkan *index* dan parameter nya, juga perbedaan terdapat pada sistem kerja dimana pada penelitian terdahulu ini melakukan otomatisasi suhu ,sedangkan pada proposal yang dibangun *memonitoring* otomatisasi penjernihan dan penetralisir tinggi nilai PH air dengan sisitem *filterisasi*.

BAB III

METODE PENELITIAN

A. Lokasi dan Waktu

Lokasi penelitian ini dilakukan di UM Parepare dan waktu pelaksanakan penelitian dilakukan kurang lebih selama 2 (dua) bulan.

B. Jenis Penelitian

Pada penelitian ini menggunakan metode *eksperimen* dan pengembangan dengan menguji rangkaian yang dibuat dengan mengacu pada *referensi* yang telah ada.

C. Alat dan Bahan

Dalam pelaksanaan perancangan alat pada penilitian yang akan dilakukan, dibutuhkan beberapa alat dan bahan . Adapun alat dan bahan yang dibutuhkan dapat dilihat pada tabel dibawah ini.

Tabel 3.1 Alat

No	Alat	Spesifikasi
1	Laptop	*Thinkpad* X220 *Intel*(R) Core(TM) i5-2520M CPU
2	Ponsel Android	Xiomi redmi note 12 pro

Tabel 3.2 Bahan

No	Bahan	Spesifikasi
1	*Mikrokontroler*	ESP 32
2	Bahasa Program	C++,*Java*
3	Ponsel Android	Redmi note 12 pro 5G
No	**Bahan**	**Spesifikasi**
4	Mikrokontroler	ESP 32
5	Kabel	Type c , Kabel Jumper male, female
6	Saklar Swicth	KY-019 Relay

7	Pompa Air	Pompa air DC12V 240L
8	Layar	LCD 2 x 16
9	Pipa Air	¾ inc
10	Bak Penampungan Air	3
11	Katup Kendali	Selenoid Valve ½ 12V DC

D. Metode Pengumpulan Data

Pengumpulan data yang dilakukan pada penelitian ini terdiri dari beberapa tahapan. Tahapan pertama dilakukan studi literatur dengan mengumpulkan data serta informasi menggunakan bahan perpustakaan serta *referensi* sumber-sumber menjadi landasan teori yang mengenai perancangan sistem *monitoring* tingkat ph dan kekeruhan air dalam menentukan kelayakan konsumsi air .Tahapan kedua yaitu *Desain* dan perancangan sistem dilakukan dengan menggambarkan rencana serta membuat sketsa serta pengaturan beberapa bagian terpisah kedalam satu kesatuan yang utuh yang saling berkesinambungan.

BAB IV

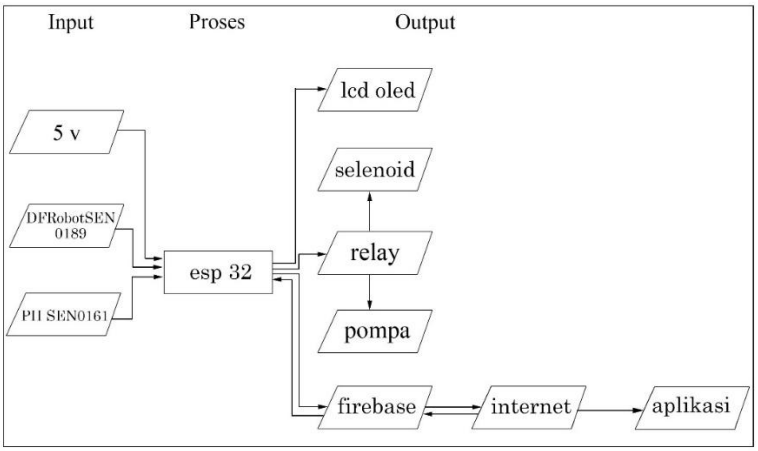
HASIL DAN PEMBAHASAN

Pada Pengembangan *evidence monitoring* tingkat ph dan kekeruhan air untuk menentukan kelayakan konsumsi air ini perlu diperhatikan dengan baik pada saat pembuatan dan perancangan setiap komponen untuk memastikan integrasi yang optimal dan kinerja yang maksimal. Sistem ini mencakup beberapa komponen utama yang saling terintegrasi untuk mencapai kinerja yang sebagaimana mestinya.

A. Rancangan Sistem

Perancangan pengembangan *evidence monitoring* tingkat ph dan kekeruhan air untuk menentukan kelayakan konsumsi air di pdam kota parepare ini meliputi Perancangan Perangkat keras dan perancangan perangkat lunak. Adapun *block diagram* dan sistem yang akan dibuat dapat dilihat pada gambar berikut.

1. *Block Diagram* Sistem



Gambar 4. 1 *Block diagram* sistem

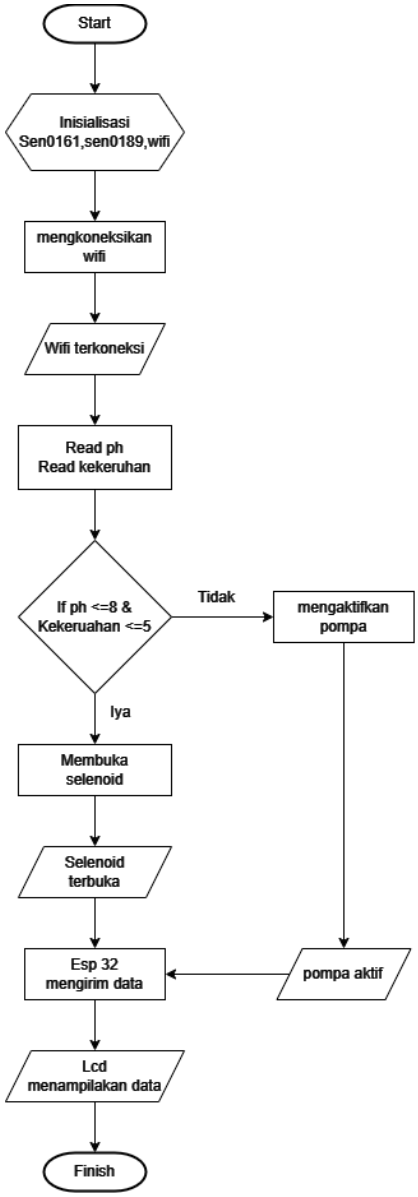
Berdasarkan gambar di atas, maka diperoleh keterangan sebagai berikut:

a. Memberikan daya dengan tegangan 5V kepada esp 32.

b. Sensor PH (Sensor PH SEN0161-V2) dan sensor kekeruhan (Sensor *DFRobot*-SEN0189) di hubungkan dengan esp 32.
c. Selanjutnya sensor PH dan sensor kekeruhan mengirimkan hasil pembacaan PH dan kekeruhan kepada *Microkontroler* ESP 32.
d. Ketika data diterima oleh ESP 32, jika data yang didapat tidak sesuai dengan kadar kualitas air yang di parameterkan, maka *relay* akan menjalankan arus listrik untuk menghidupkan pompa air yang selanjutya air yang ada pada bak distribusi akan dipompa kedalam bak *filter*.
e. Ketika air berada dalam kadar yang memenuhi kelayakan sesuai dengan yang diparameterkan maka *solenoid* akan terbuka dan mengalirkan air untuk di distribusikan.
f. ESP 32 Melanjutkan proses pengiriman data ke *interface* LCM 1602B COB LCD.
g. ESP 32 juga mengirimkan data aplikasi *android* yang ada di ponsel bermediakan *Firebase* melalui internet sebagai jembatan untuk dapat sampai pada aplikasi *android*.
h. Pada aplikasi *android* yang berfungsi sebagai *interface*, aplikasi akan menampilkan tampilam dengan keterangan nilai ph, nilai kekeruhan, beserta dengan keadaan ketika *solenoid* telah terbuka.

2. Flowchart

a. *Flowchart* Proses Menampilkan di Lcd

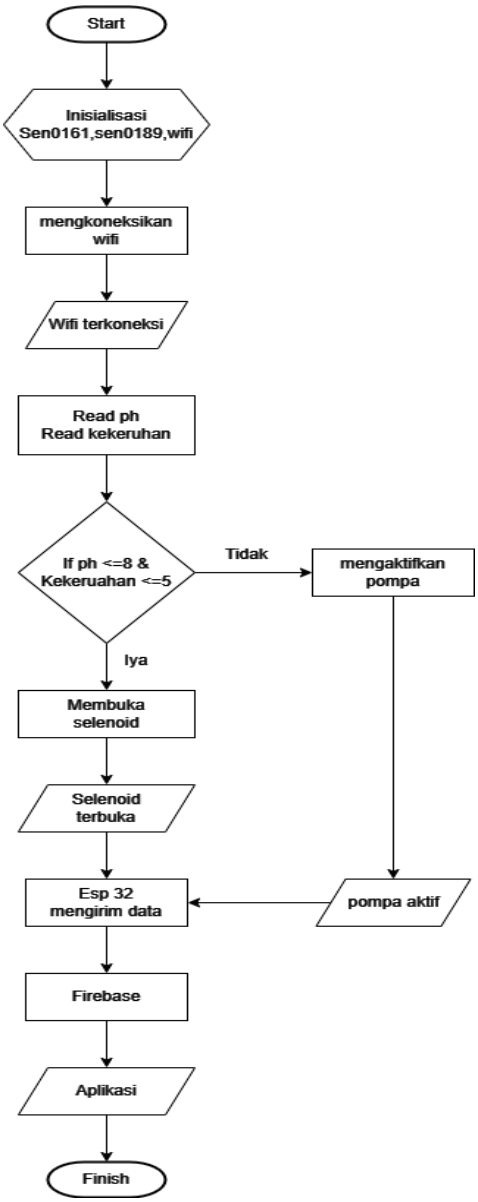


Gambar 4. 2 *Flowchart* proses menampikan di lcd

Pada diagram alir atau *flowchart* diatas diawali dengan *start* atau mulai seperti syarat pembuatan *flowchat* pada umumnya. Kemudian *penginisialisasian*, dilanjutkan dengan proses pengkoneksian *wifi*, selanjutnya Pembacaan nilai ph (Sensor PH SEN0161-V2) dan kekeruhan (Sensor *DFRobot*-SEN0189) . Kemudian program akan membaca kondisi jika ph <=8 dan kekeruhan <=5 maka katup atau *solenoid* akan terbuka untuk mengalirkan air, namun jika tidak memenuhi kondisi maka air akan di

pompa untuk masuk ke dalam bak *filter* air sampai memenuhi kondisi yang ditentukan dan Esp 32 akan mengirimakan data kepada LCD untuk ditampilkan.

b. *Flowchart* Proses menampilkan di Aplikasi

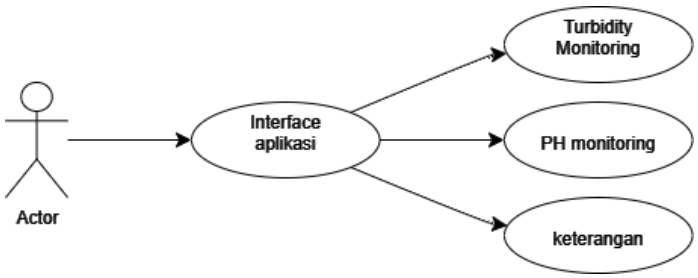


Gambar 4. 3 *Flowchart* Proses Menampilkan di Aplikasi

Pada diagram alir atau *flowchart* diatas diawali dengan *start* atau mulai seperti syarat pembuatan *flowchat* pada umumnya. Kemudian *penginisialisasian*, dilanjutkan dengan proses pengkoneksian *wifi*, selanjutnya Pembacaan nilai ph (Sensor PH SEN0161-V2) dan kekeruhan (*Sensor DFRobot*-SEN0189) . Kemudian program akan membaca kondisi jika ph <=8 dan kekeruhan <=5 maka katup atau *solenoid* akan

terbuka untuk mengalirkan air, namun jika tidak memenuhi kondisi maka air akan di pompa untuk masuk ke dalam bak *filter* air sampai memenuhi kondisi yang ditentukan dan Esp 32 akan mengirimakan data ke *Firebase* dan kemudian *Firebase* akan mengirimakan data ke aplikasi.

3. *Use Case Diagram*



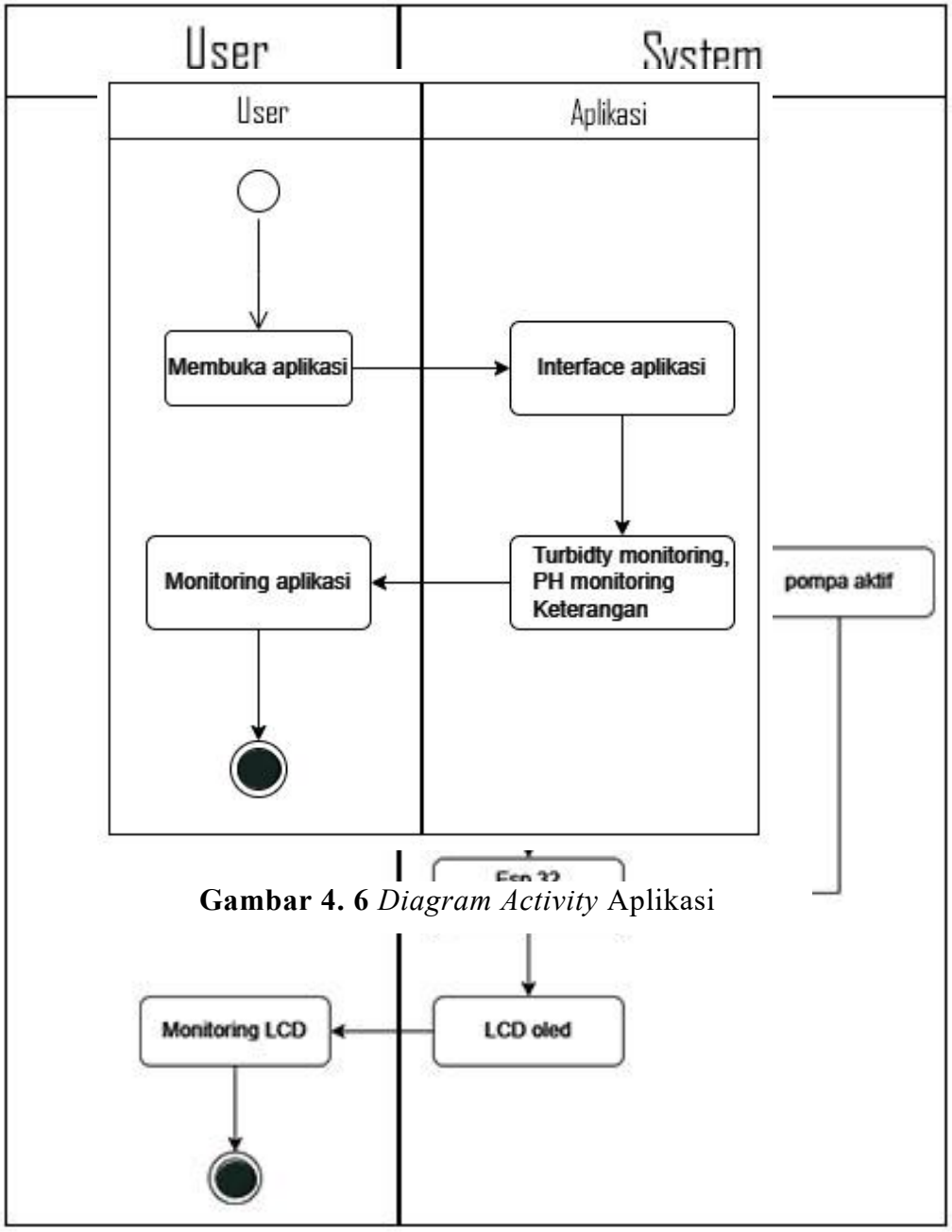
Gambar 4. 4 *Use Case Diagram*

Tabel 4. 1 *Use Case Diagram*

Nama Use Case	**Keterangan**
Interface	*Interface* merupakan tampilan halaman pada aplikasi.
Turbidity monitoring	*Turbidity monitoring* berupa *card box* yang memberikan value kekeruhan dan status kondisi air.
PH *monitoring*	PH *monitoring* berupa *card box* yang memberikan *value* PH dan status PH.
Nama Use Case	**Keterangan**
Keterangan	Keterangan merupakan penjelasan yang dalam hal ini memberikan data alat yang sedang bekerja

4. *Activity Diagram*

a. *Diagram Activity System*



Gambar 4. 5 *Diagram Activity System*

b. *Diagram Activity* Aplikasi

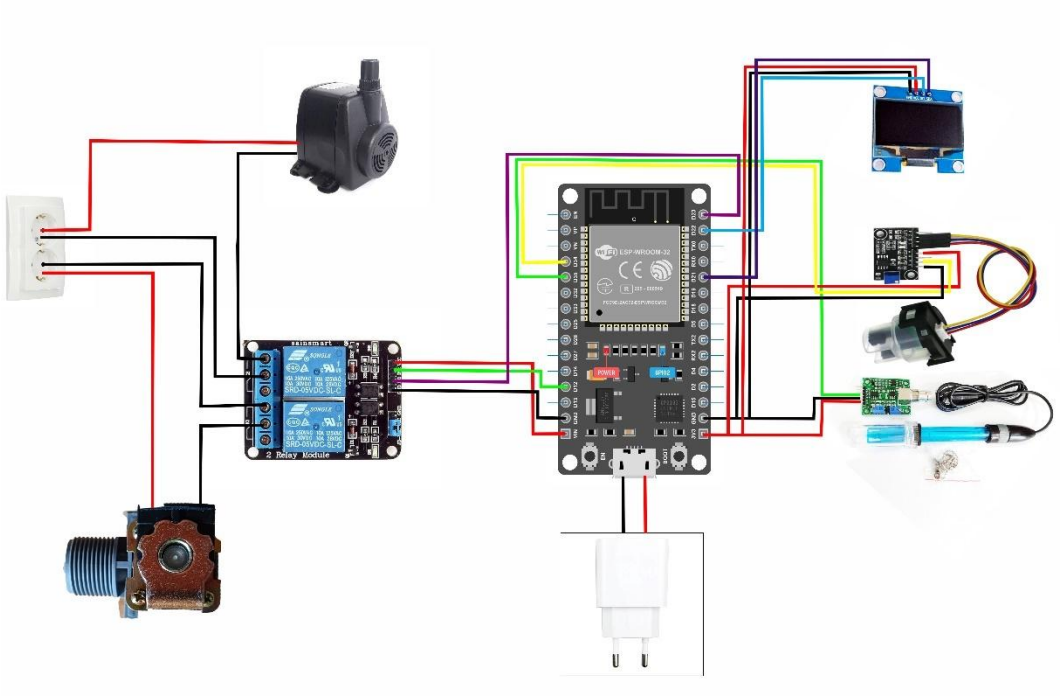
Gambar 4. 6 *Diagram Activity* Aplikasi

B. Rancangan Sistem Perangkat Keras

Perancangan dan perakitan perangkat keras merupakan langkah taktis untuk dapat menghasilkan sebuah rangkaian alat yang tersinergi dari beberapa komponen yang menghasilkan suatu kesatuan. Pada rancangan ini akan memberikan sebuah penjelasan mengenai penggunaan komponen dan perakitannya untuk dapat menghasilkan sebuah alat

untuk mendeteksi nilai ph dan kekeruhan sebagai parameter untuk menentukan kerlayakan konsumsi air.

Rangkaian keseluruhan alat :



Gambar 4. 7 Rancangan sistem perangkat keras

Pada gambar 4.7 memperlihatkan alur rangkaian dari komponen yang digunakan mulai dengan penghubungan *pin* esp 32 ke komponan lain hingga perancangan kabel pompa dan *selenoid* ke *relay* untuk dapat mengatur kerja pompa dan *selenoid*.

1. Perancangan dilakukan dengan penghubungan sensor PH (Sensor PH SEN0161) yang menghubungkan 3 *pin* nya ke ESP 32 DEVKIT V1, yakni *pin* (VCC) ke (3V3), *pin* (GND) dihubungkan dengan (GND) dan *pin* ph *output* (PO) dihubungkan dengan (GPIO35).
2. Menghubungkan 3 pin sensor kekeruhan (Sensor *DFRobot*-SEN0189) ke ke esp 32 *devkit* v1, yakni *pin* (VCC) ke (3V3), *pin* (GND) dihubungkan dengan (GND) dan *pin analog* (A) dihubungkan dengan (GPIO34).

3. Menghubungkan 4 *pin* LCD OLED ke ESP 32 DEVKIT V1, yaitu *pin* (VDD) ke (3V3), *pin* (GND) dihubungkan dengan (GND), *pin* (SDA) dihubungkan dengan *pin* (GPIO21) dan *pin* (SCK) dihubungkan dengan *pin* (GPIO22).
4. Menghubungkan 4 *pin relay* 2 *chanel* ke ke ESP 32 *DEVKIT* V1, yaitu *pin* (VCC) ke *pin* (VIN), *pin* (GND) ke *pin* (GND), *pin chanel* 1 (IN1) ke (GPIO23) dan *pin chanel* 2 ke (GPIO13).
5. Menghubungkan 4 lubang *relay* 2 *chanel*, yaitu lubang K1 dan K2. Pada lubang K1 dihubungkan ke *selenoid* dengan memutus salah satu kabel dan masing-masing kabel dimasukkan pada 2 lubang K1 dengan penyesuaian kondisi *relay normaly close*. Untuk K2 dihubungkan ke pompa dengan memutus salah satu kabel dan masing-masing kabel dimasukkan pada 2 lubang K2 dengan penyesuaian kondisi *relay normaly close*.
6. Menghubungkan kabel selenoid dan pompa ke terminal listrik dengan jenis arus ac.
7. Menghubungkan ESP 32 *DEVKIT* V1 dengan dengan terminal listrik menggunakan kabel data *type* b dan *adaptor* 5V.

Tabel 4.2 Penghubungan Kabel Sensor PH SEN0161 dan ESP 32

NO	SENSOR PH SEN0161	ESP 32 DEVKIT V1
1	VCC	3V3
2	GND	GND
3	PO	GPIO 35

Tabel 4.3 Penghubungan Kabel Sensor *DFRobot*-SEN0189 dan ESP 32

NO	Sensor *DFRobot*-SEN0189	ESP 32 *DEVKIT* V1
1	VCC	3V3
2	GND	GND

3	A	GPIO 34

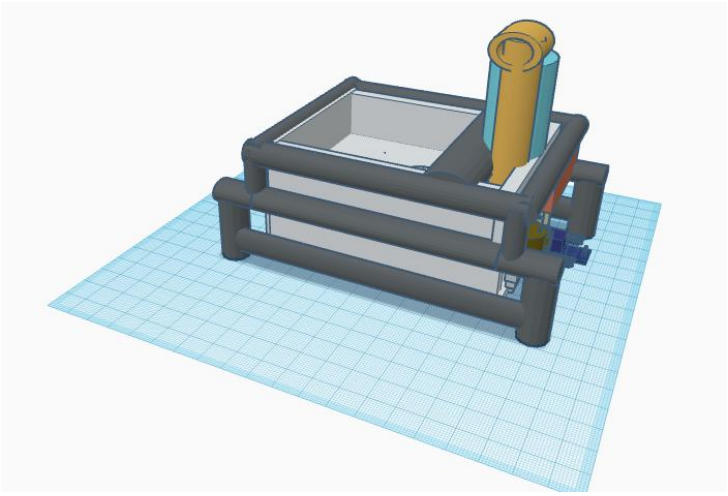
Tabel 4.4 Penghubungan LCD OLED dan ESP 32

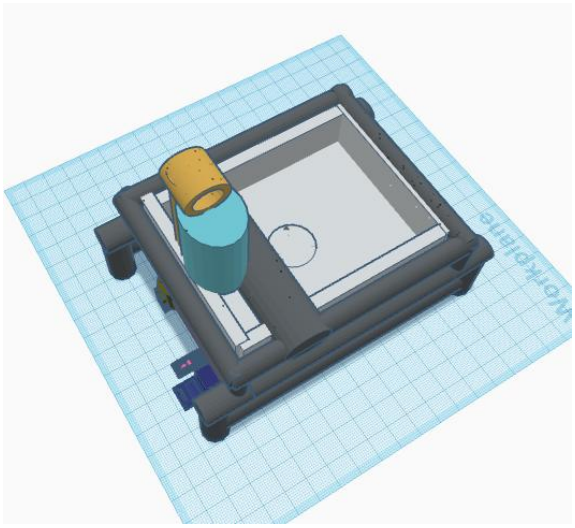
NO	**LCD *OLED***	**ESP 32 *DEVKIT* V1**
1	VDD	3V3
2	GND	GND
3	SDA	GPIO 21
4	SCK	GPIO 22

Tabel 4.5 Penghubungan kabel *Relay* dan ESP 32

NO	***RELAY***	**ESP 32 *DEVKIT* V1**
1	VCC	VIN
2	GND	GND
3	IN1	GPIO 23
4	IN2	GPIO 13

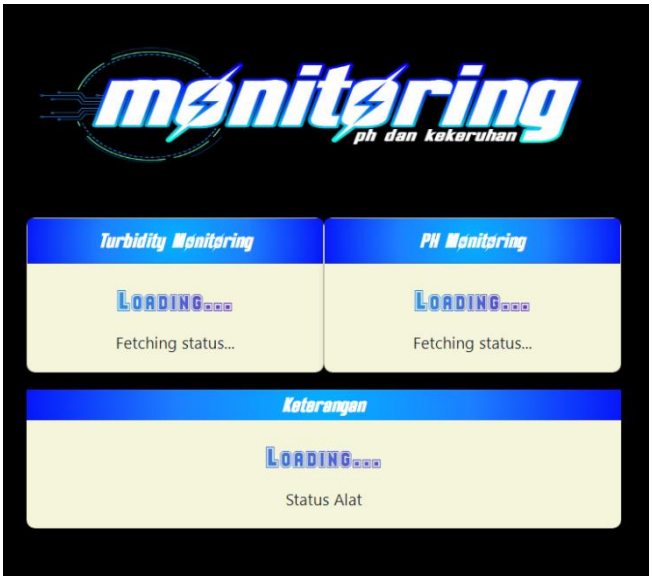
C. Rancangan *Prototype System*





Gambar 4. 8 *Prototype System*

D. Rancangan Perangkat Lunak



Gambar 4.9 *Interface* Aplikasi

Gambar di atas merupakan tampilan *interface* aplikasi dimana terdapat tiga *box* yang memiliki muatan nilai masing-masing, yaitu *box* pertama yang menampilkan nilai kekeruhan dengan status air, *box* kedua yang memuat nilai ph dengan status ph nya dan *box* ketiga berisikan keterangan mengenai status alat.

1. *Source Code Arduino*

```
#define WIFI_SSID "Redmi Note 12 Pro 5G"
#define WIFI_PASSWORD "difa1234"
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Connecting to Wi-Fi");
while (WiFi.status() != WL_CONNECTED)
```

Gambar 4.10 *Script wifi*

Gambar diatas merupakan *script* untuk mengkoneksikan *wifi*, dimana dimulai dengan mendifinisakan *id wifi* dan *pasword*, yang kemudian dilakukan pemanggilan dilanjutkan dengan proses pengkoneksian dan keluaran status *wifi* .

```
sensorValue = analogRead(pHSensorPin);
float voltage = sensorValue * (3.3 / 4095.0);
int val = analogRead(34);
teg = val*(3.3 /
kekeruhan = 100.00 - (teg / 1.63)*100.00;
```

Gambar 4.11 *Script* pembacaan sensor

Gambar diatas merupakan *script* pembacaan untuk sensor dimana dilakukan pendefinisan *pin* yang sebelumnya telah dideklarasikan. Pada kedua sensor di hubungkan pada *pin input* untuk sesnsor kekeruhan di deklarasikan *integer* dengan *pin* 34 dan untuk sensor ph di deklarasikan *float* pada *pin* 35 kemudian dilanjutkan dengan rumus pembacaan sensor utuk mendapatkan nilai yang diinginkan.

```
if (kekeruhan <= 5.0 && pHValue <=8.0) {
  digitalWrite(pompa, HIGH);
  digitalWrite(valve, LOW);
else if (kekeruhan > 5.0 && pHValue <=8.0) {
   digitalWrite(pompa,LOW);
   digitalWrite(valve, HIGH);
}else if (kekeruhan > 5.0 && pHValue >8.0) {
   digitalWrite(pompa,LOW);
   digitalWrite(valve, HIGH);
```

Gambar 4.12 *Script* kondisi alat

Gambar diatas merupakan *script* untuk memberikan penyesuaian kondisi kerja alat dengan memasang parameter dimana ketika *index* mencapai atau tidak terhadap parameter maka alat dengan otomatis melakuka kerjan sebagaimana yang telah diinstruksikan.

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
   Serial.println(F("Gagal memulai SSD1306"));
for(;;);
display.print("NILAI KEKERUHAN:");
display.setCursor(0, 10);
display.print(kekeruhan);
display.setCursor(0, 20);
display.print("NILAI PH:");
display.setCursor(0, 30);
display.print(pHValue, 2);
```

Gambar 4.13 *Script* LCD

Gambar diatas merupakan *script arduino* untuk lcd dimana awalnya dilakukan penginisialisasian objek yang kemudian dilanjutkan dengan *script* untuk memulai lcd dan

```
if (Firebase.signUp(&config, &auth, "", "")) {
    Serial.p
(Firebase.RTDB.
```

Gambar 4.14 *Script arduino* ke *firebase*

adapun *script* selanjutnya yang merupakan *script* untuk menampilkan nilai ph dan kekeruhan.

Gambar diatas merupakan *script arduino* untuk *firebase* dimana dilakukan pemanggilan alamat *firebase* kemudian dilakukan instruksi untuk menghubugkan *firebase* kemudian ketika telah berhasil terhubung dilakukan pengiriman *value* kepada *firebase*.

2. *Source Code* Aplikasi



Gambar 4.15 *Interface* aplikasi

```
<img src="stiker monitoring.jpg" alt="Deskripsi Gambar" class="img">
```

Gambar 4.16 *Script* memasukkan gambar

Gambar 4.15 diatas merupakan gambar *interface* pada aplikasi sebagai upaya untuk menambah daya tarik aplikasi dimana gambar tersebut telah didesain agar memiliki tampilan yang menarik dilihat dan pada gambar 4.16 merupakan *script* memanggil gambar untuk dapat masuk di *interface* aplikasi dimana sebelumnya gambar tersebut telah dimasukkan ke dalam satu folder yang sama dengan folder aplikasi.

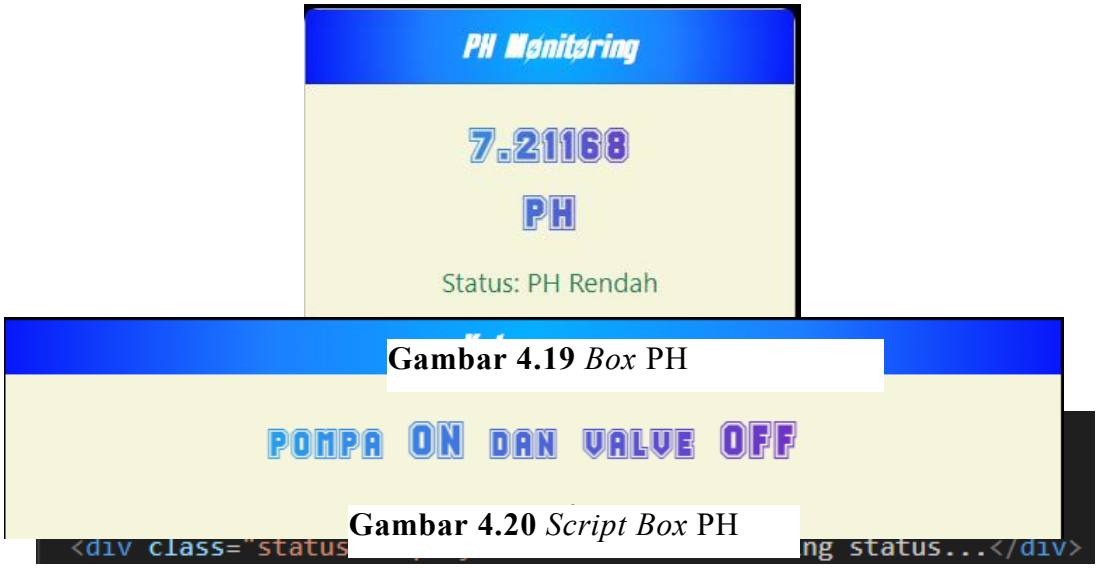


Gambar 4.17 *Box Turbidity*

```
<div class="card-he
<div class="card-body">
  <div class="value-display" id="turbidity">Loading...</div>
  <div class="sta                                          ..</div>
```

Gambar 4.18 *Script box turbidity*

Gambar 4.17 merupakan *box interface* untuk manampilkan keterangan *box* nilai kekeruhan beserta dengan status air, sedangkan pada gambar 4.18 merupakan *script* yang digunakan untuk *box turbidity* dengan menggunakan fungsi *class* dan *id*.



Gambar 4.19 *Box* PH

Gambar 4.20 *Script Box* PH

Gambar 4.19 merupakan *box interface* untuk manampilkan keterangan *box* nilai ph beserta dengan status ph air, sedangkan pada gambar 4.20 merupakan script yang digunakan untuk *box* ph dengan menggunakan fungsi *class* dan *id*.

```
const turbidityRef = database.ref('keterangan/data/turbidity');
```

Gambar 4.22 *Script Box* Keterangan

Gambar 4.21 merupakan *box* keterangan untuk manampilkan

Gambar 4.21 *Box* Keterangan

```
<div class="card-header">Keterangan</div>
<div class="card-body">
  <div class="value-display" id="status_alat">Loading...</div>
  <div class="status-display" id="status-card-3">Status Alat</div>
```

```
turbidityRef.on('value', (snapshot) => {
const value = snapshot.val();
document.getElementById('turbidity').innerText = `${value}
NTU`;
```

keterangan alat yang sedang bekerja, sedangkan pada gambar 4.22 merupakan *script* yang digunakan untuk *box* keterangan dengan menggunakan fungsi *class* dan *id*.

Gambar di atas merupakan *script* aplikasi untuk mendapatkan nilai kekeruhan dimana dilakukan *pereferensian* nilai kekeruhan yang berdasar pada alamat *firebase*, yang kemudian dilakukan pembacaan perubahan nilai pada *firebase* agar nilai yang tampil pada *interface* aplikasi selalu menunjukan nilai terkini yang aada pada *firebase*.

Gambar 4.23 *Script* Mendapatkan Nilai Kekeruhan

```
const PHRef = database.ref('keterangan/data/PH');
PHRef.on('value', (snapshot) => {
const value2 = snapshot.val();
document.getElementById('PH').innerText = `${value2}
PH`;
```

Gambar 4.24 *Scrip* Mendapatkan Nilai PH

Gambar di atas merupakan *script* aplikasi untuk mendapatkan nilai ph dimana dilakukan *pereferensian* nilai ph yang berdasar pada yang berdasar pada alamat *firebase*, yang kemudian dilakukan pembacaan perubahan nilai pada *firebase* agar nilai yang tampil pada *interface* aplikasi selalu menunjukan nilai terkini yang ada pada *firebase*.

```
const status_alatRef = database.ref('keterangan/data/status_alat');
status_alatRef.on('value', (snapshot) => {
const value3 = snapshot.val();
document.getElementById('status_alat').innerText = `${value3}`;
```

Gambar 4.25 *Script* Mendapatkan Nilai Keterangan

Gambar di atas merupakan *script* aplikasi untuk mendapatkan keterangan alat dimana dilakukan *pereferensian* keterangan yang berdasar pada yang alamat *firebase*, yang kemudian dilakukan pembacaan perubahan nilai pada *firebase* agar keterangan yang tampil pada

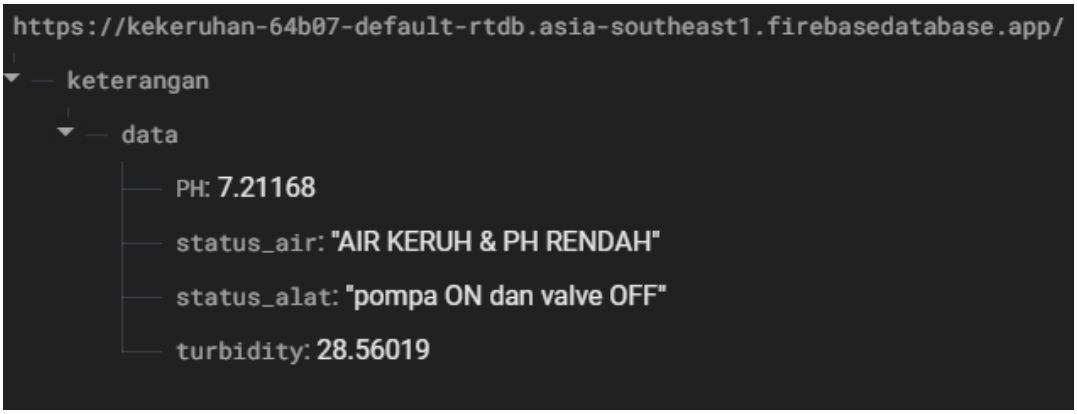
```
background-color: #000000;
```

Gambar 4.26 *Script Backround*

interface aplikasi selalu menunjukan perubahan keterangan terkini yang ada pada *firebase*.

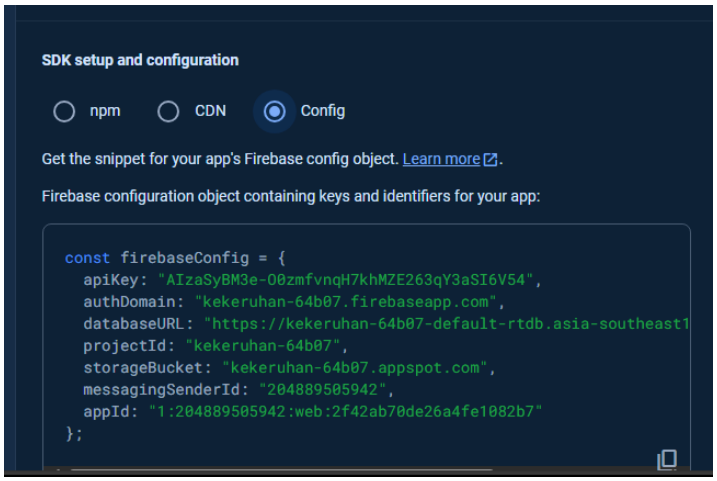
Adapun pada gambar diatas merukan *script* untuk memberikan warna *background* atau warna latar belakang pada tampilan *interface* aplikasi.

E. Firebase



Gambar 4.27 *Interface realtime database*

Gamabar diatas merupakan tampilan *realtime* database pada *firebase* dimana memperlihatkan nilai – nilai dan keterangan yang berhasil dikirimkan oleh alat untuk dapat ditampilakan pada *firebase*.



Gambar 4.28 *Config firebase*

Gambar di atas merupakan *config* pada *firebase* yang sekaligus merupaakan alamat *firebase* yang pada proses perancangan alat kali ini masuk pada 2 program yang pertama menjadi alamat untuk alat agar dapat mengirimkan nilai pada *firebase* dan yang kedua menjadi alamat agar aplikasi dapat mengambil nilai yang ada pada *firebase*. Terdapat 7 *point* yang memuat *config* ini mulai dari *apiKey* hingga *appId*.

F. Rancangan Alat

Pada bagian ini membrikan gambar mengenai hasil perancangan alat *monitoring* ph dan kekeruhan air untuk menuntuklan kelayakan konsumsi air.



Gambar 4.29 Hasil Rancangan alat



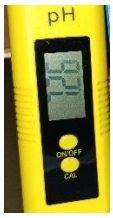
Gambar 4.30 Hasil Rancangan alat

G. Hasil Pengujian

Adapun beberapa hasil percobaan atau hasil pengujian alat agar memastikan perangkat berjalan dengan baik, Bisa dilihat di tabel di bawah ini.

1. Nilai PH di Percobaan Air Keruh

Tabel 4.6 Percobaan air keruh

No	Nilai PH	Nilai PH (alat standar)	Gambar LCD	Gambar alat PH
1	6.52	7.34		
2	7.75	7.26		
3	7.95	7.26		
4	7.78	7.26		
5	8.23	7.26		
6	8.23	7.26		

LanjutanTabel 4.6 Percobaan air keruh

No	Nilai PH	Nilai PH (alat standar)	Gambar LCD	Gambar alat PH

7	7.50	7.58		
8	8.23	7.58		
9	8.71	7.42		
10	6.70	6.86		

Berdasarkan pada data tabel 4.6, maka untuk mendapatkan nilai Standar *Deviasi* adalah sebagai berikut :

Diketahui :

S : Standar Deviasi

xi : nilai x ke i

$\bar{x}$: nilai rata-rata data

n : jumlah data

Penyelesaian :

$$S = \sqrt{\frac{\sum(xi - \bar{x})^2}{n-1}}$$

$$S = \sqrt{\frac{(6{,}25-7{,}73)^2 + (7{,}75-7{,}73)^2 + (7{,}95-7{,}73)^2 + (7{,}78-7{,}73)^2 + (8{,}23-7{,}73)^2 + (8{,}23-7{,}73)^2 + (7{,}50-7{,}73)^2 + (8{,}23-7{,}73)^2 + (8{,}71-7{,}73)^2 + (6{,}70-7{,}73)^2}{10-1}}$$

S = 0.81

Tabel 4. 7 *Presentase* nilai *eror* PH pada percobaan air keruh

No	Nilai PH (A)	Nilai PH alat standar (B)	E ($\left\|\frac{B-A}{B}\right\|$ x 100%)
1	6,52	7,34	11,17%
2	7,75	7,26	6,75%
3	7,95	7,26	9,51%
4	7,78	7,26	7,16%
5	8,23	7,26	13,36%
6	8,23	7,26	13,36%
7	7,50	7,58	1,06%
8	8,23	7,58	8,57%
9	8,71	7,42	17,37%
10	6,70	6,86	2,33%

Untuk mendapatkan nilai *presentasi eror* diatas maka dilakukan penyelesaian sebagai berikut :

Rumus yang digunakan adalah E ($\left|\frac{B-A}{B}\right|$ x 100%) dengan :

| | = Merupakan lambang nilai mutlak atau *absolut* yang digunakan saat menghitung *persentase* kesalahan, *error relatif*, atau selisih *absolut*.
E = eror

B = nilai alat ukur standar

A = nilai ph yang didapat dari pembacan sensor

Sebab mencari presentasi maka dikalikan dengan 100%

Misalnya,

A = 6,52 B = 7,34

Maka,

E = (| $\frac{7,34-6,52}{7,34}$ | x 100%)

E =| $\frac{0,82}{7,34}$ |x 100%

E = 11,17%

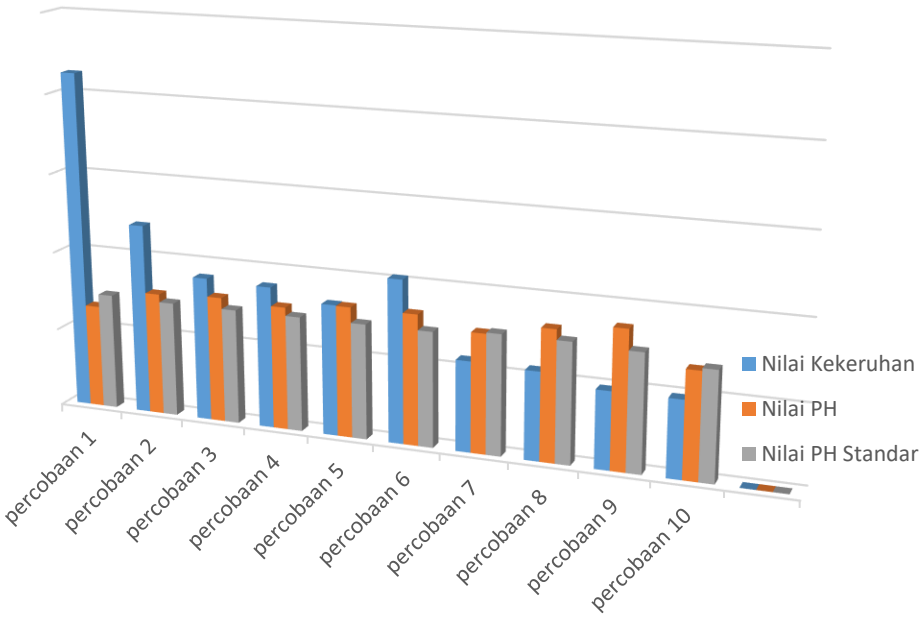
Atau,

A = 7,75 B = 7,26

E = (| $\frac{7,26-7,75}{7,26}$ | x 100%)

E =| $\frac{0,49}{7,26}$ |x 100%

E= 6,75%



	percobaan 1	percobaan 2	percobaan 3	percobaan 4	percobaan 5	percobaan 6	percobaan 7	percobaan 8	percobaan 9	percobaan 10
Nilai Kekeruhan	21,28	12,04	9,09	8,95	8,26	10,27	5,7	5,56	4,87	4,87
Nilai PH	6,52	7,75	7,95	7,78	8,23	8,23	7,5	8,23	8,71	6,7
Nilai PH Standar	7,34	7,26	7,26	7,26	7,26	7,26	7,58	7,58	7,42	6,86

Gambar 4.31 *Grafik* Data 10 Percobaan

2. Nilai Kekeruhan di Percobaan Air Keruh

Tabel 4. 8 Percobaan air keruh

NO	Nilai Kekeruhan	Gambar Layar LCD
1	21.28	
2	12.04	
3	9.09	
4	8.95	
5	8.26	
6	10.27	

Lanjutan Tabel 4.8 Percobaan air keruh

NO	Nilai Kekeruhan	GambarLayar LCD
7	5.70	
8	5.56	
9	4.87	
10	4.87	

3. Percoban Air Laut

Tabel 4. 9 Percobaan air laut

Nilai Kekeruhan	Nilai PH	Nilai PH (alat standar)	Gambar LCD	Gambar alat PH
2.58	7.33	7.82		

Tabel 4. 10 *Presentase eror* ph percobaan air laut

Nilai PH (A)	Nilai PH alat standar (B)	E ($\left\|\frac{B-A}{B}\right\|$ x 100%)
7.33	7.82	0,06

4. Air Sumur Bor

Tabel 4. 11 Percobaan air sumur bor

Nilai Kekeruhan	Nilai PH	Nilai PH (alat standar)	Gambar LCD	Gambar alat PH
2.46	7.02	7.66		

Tabel 4. 12 *Presentase eror* ph percobaan air sumur bor

Nilai PH (A)	**Nilai PH alat standar (B)**	**E ($\left\|\frac{B-A}{B}\right\|$x 100%)**
7.33	7.82	0,06

5. Percobaan Dengan Rentan Waktu

Tabel 4.13 Percobaan dengan Rentan Waktu

No	**Waktu**	**Kekeruhan**	**PH**
1	00:00:16	29.82296	7.98535
2	00:00:38	30.55046	7.82418
3	00:00:59	35.30728	7.84432
4	00:01:20	34.35592	8.16667
5	00:01:42	32.62108	8.28755
6	00:02:03	32.84493	8.40843
7	00:02:24	32.62108	8.50916
8	00:02:45	32.84493	8.71062
9	00:03:07	33.7963	9.49634
10	00:03:28	34.07611	8.63004
11	00:03:49	31.55779	8.93224
12	00:04:10	31.05413	9.01282
13	00:04:32	28.31196	9.05312
14	00:04:53	26.57713	9.27473
15	00:05:14	24.78632	9.77839
16	00:05:35	29.31929	9.49634
17	00:05:56	26.24135	8.79121
18	00:06:17	25.84961	8.91209
19	00:06:38	23.16341	8.79121
20	00:07:00	24.84229	9.03297
21	00:07:21	21.9882	8.77106
22	00:07:42	21.0928	8.85165
23	00:08:03	20.02951	9.49634
24	00:08:25	20.81299	9.25458
25	00:08:46	19.13411	8.89195
26	00:09:07	18.29466	9.1337
27	00:09:28	18.01485	9.15385
28	00:09:50	19.7497	9.1337
29	00:10:12	18.46256	9.03297
30	00:10:33	18.51852	9.09341
31	00:10:54	16.78368	9.09341
32	00:11:16	16.67175	9.15385
33	00:11:37	16.11213	9.19414

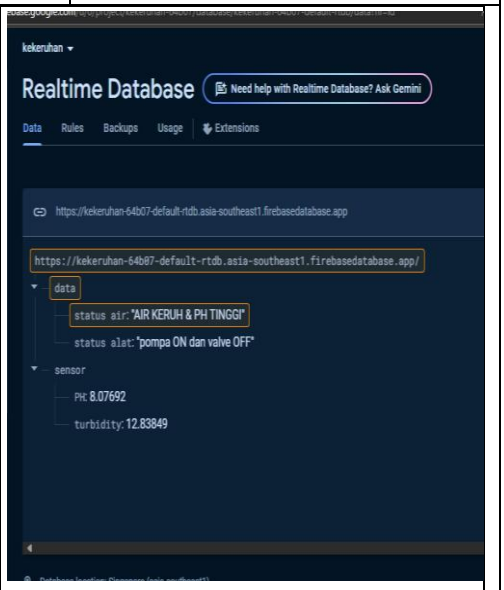
Lanjutan Tabel 4.13 Percobaan dengan Rentan Waktu

No	**Waktu**	**Kekeruhan**	**PH**
34	00:11:59	15.27269	9.15385

35	00:12:20	15.88828	9.25458
36	00:12:41	15.21673	9.1337
37	00:13:02	14.2094	9.25458
38	00:13:24	12.81033	9.19414
39	00:13:45	12.92227	9.15385
40	00:14:06	12.4186	9.23444
41	00:14:27	12.64245	9.29488
42	00:14:48	11.91494	8.97253
43	00:15:09	10.45991	9.15385
44	00:15:31	10.40394	9.47619
45	00:15:52	10.45991	9.21429
46	00:16:13	10.06817	9.174
47	00:16:35	9.78836	9.85898
48	00:16:56	10.40394	9.27473
49	00:17:17	8.66911	9.25458
50	00:17:39	8.33333	9.33517
51	00:18:00	8.05352	9.47619
52	00:18:21	8.33333	9.15385
53	00:18:43	7.77371	9.45605
54	00:19:04	7.60582	9.47619
55	00:19:25	6.99023	9.79854
56	00:19:48	7.21408	8.97253
57	00:20:09	6.87831	9.47619
58	00:20:30	6.71042	9.53663
59	00:20:51	6.15079	9.27473
60	00:21:13	6.26272	9.45605
61	00:21:34	5.59117	9.21429
62	00:21:55	6.09483	9.09341
63	00:22:17	5.64713	9.47619
64	00:22:39	6.03887	9.49634
65	00:23:00	5.64713	9.75825
66	00:23:22	5.87098	9.65751
67	00:23:44	5.59117	9.4359
68	00:24:05	4.58384	9.47619
69	00:24:34	5.2554	9.47619
70	00:24:56	4.80769	9.71795

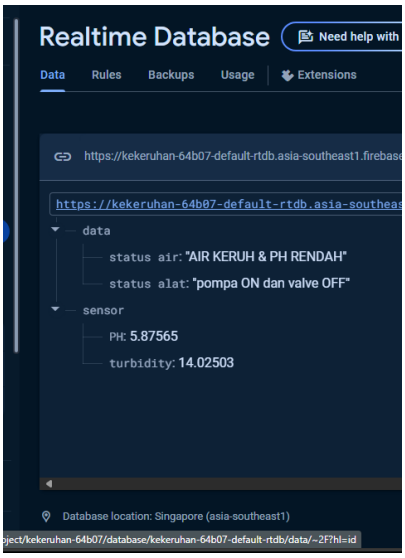
H. Pengujian *Blackbox*

Tabel 4. 14 Pengujian air keruh dan ph tinggi

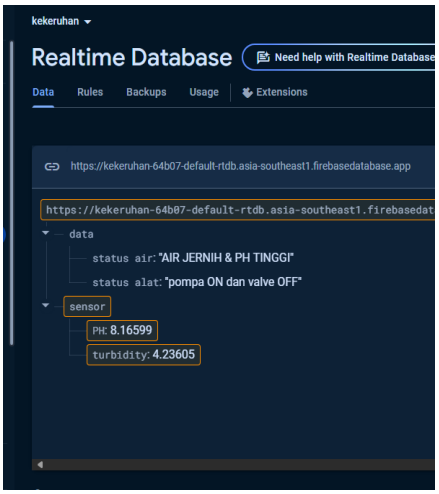
Uji Coba	Hasil	Keterangan
Nilai ph air > 8 & Nilai Kekeruhan >5	✓	*Relay chanel* 2 akan menyala dan menghidupkan pompa untuk membawa air pada *pemfilteran* & esp 32 akan *mengupload* nilai yang didapat sensor berserta status alat
		

Tabel 4. 15 Pengujian air keruh dan ph rendah

Uji Coba	Hasil	Keterangan
Nilai ph air > 8 & Nilai Kekeruhan =<5	✓	*Relay chanel* 2 akan menyala dan menghidupkan pompa untuk membawa air pada *pemfilteran* & esp 32 akan *mengupload* nilai yang didapat sensor berserta status alat

Tabel 4. 16 Pengujian air jernih dan ph tinggi

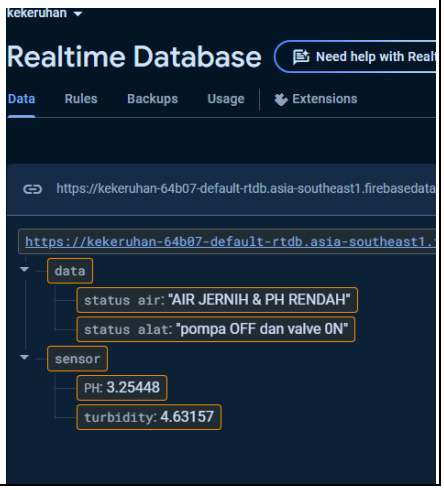
Uji Coba	Hasil	Keterangan
Nilai ph air > 8 & Nilai Kekeruhan <= 5	✓	*Relay chanel* 2 akan menyala dan menghidupkan pompa untuk membawa air pada *pemfilteran* & esp 32 akan *mengupload* nilai yang didapat sensor berserta status alat
		

Tabel 4. 17 Pengujian air jernih dan ph rendah

Uji Coba	Hasil	Keterangan

<table>
<tr><td>Nilai ph air <= 8 &

Nilai Kekeruhan <= 5</td><td>✓</td><td>Relay chanel 1 akan menyala dan menghidupkan selenoid / keran otomatis untuk menyalurkan air & esp 32 akan mengupload nilai yang didapat sensor berserta status alat</td></tr>
<tr><td colspan="3">

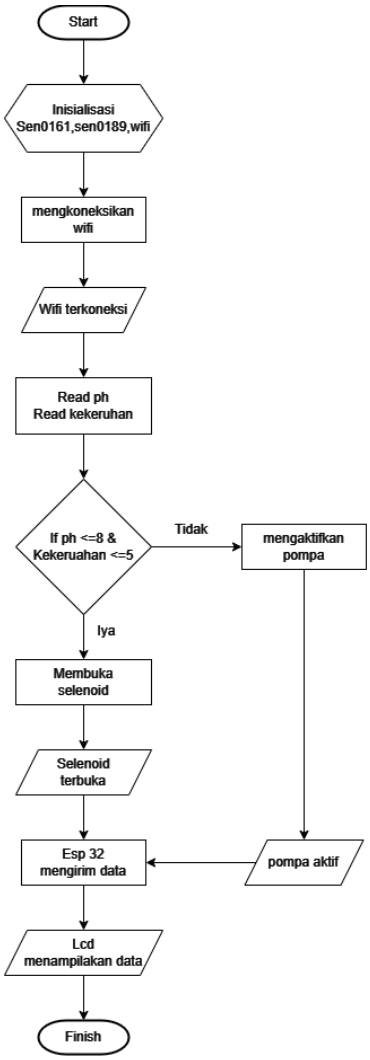

</td></tr>
</table>

I. Pengujian *whitebox*

Pengujian *Whitebox*, adalah suatu metode pengujian aplikasi yang menggunakan penjelasan struktur kontrol sebagai bagian dari *component-level design* untuk membuat *test cases*.

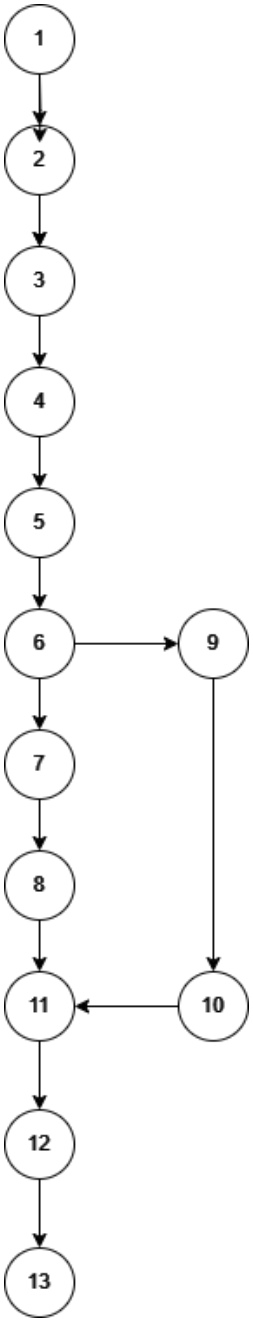
1. Proses menampilkan di LCD

a. *Flowchart*



Gambar 4.32 *Flowchart* Proses menampilkan di LCD

b. *Flowgraph*



Gambar 4.33 *Flowgraph* Proses menampilkan di LCD

Dari gambar *Flowgraph* diatas dapat dilakukan proses perhitungan sebagai berikut :

Diketahui :

N (*node*) = 13 E (*edge*) = 14 P (Predikat node) = 1+1

Penyelesaian :

V(G) = (E – N) + 2

= (13 - 13) + 2

= 2

Path 1 : 1-2-3-4-5-6-7-8-11-12-13

Path 2 : 1-2-3-4-5-6-9-10-11-12-13

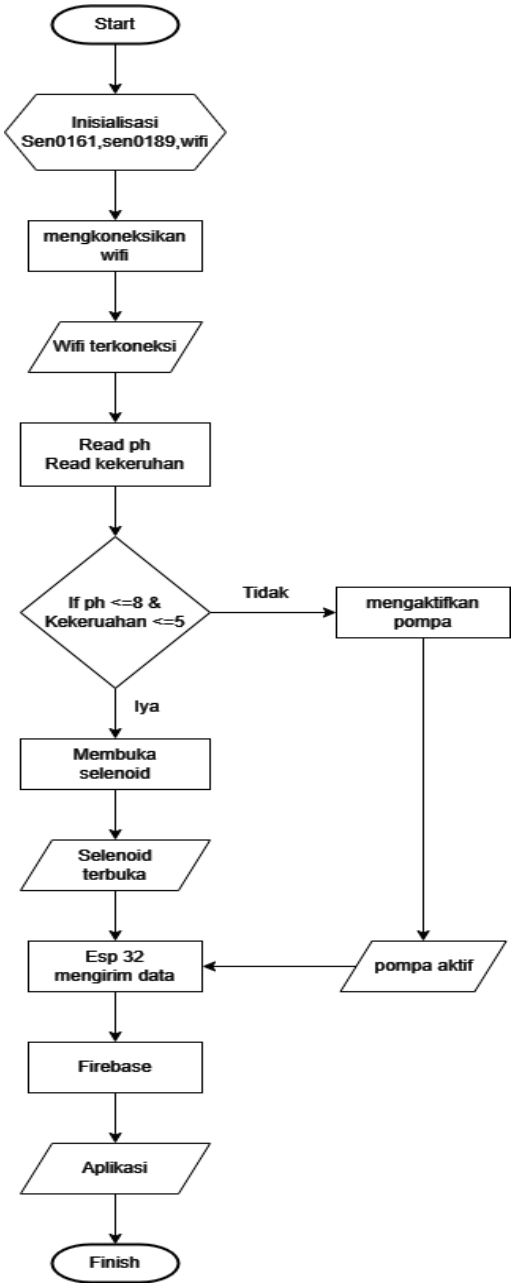
c. *Grafik Matriks*

Tabel 4.18 *Grafik Matriks*

	1	2	3	4	5	6	7	8	9	10	11	12	13	E – 1
1		1												1-1=0
2			1											1-1=0
3				1										1-1=0
4					1									1-1=0
5						1								1-1=0
6							1		1					2-1=1
7								1						1-1=0
8											1			1-1=0
9										1				1-1=0
10											1			1-1=0
11												1		1-1=0
12													1	1-1=0
13														0
	Sum(E+1)													2

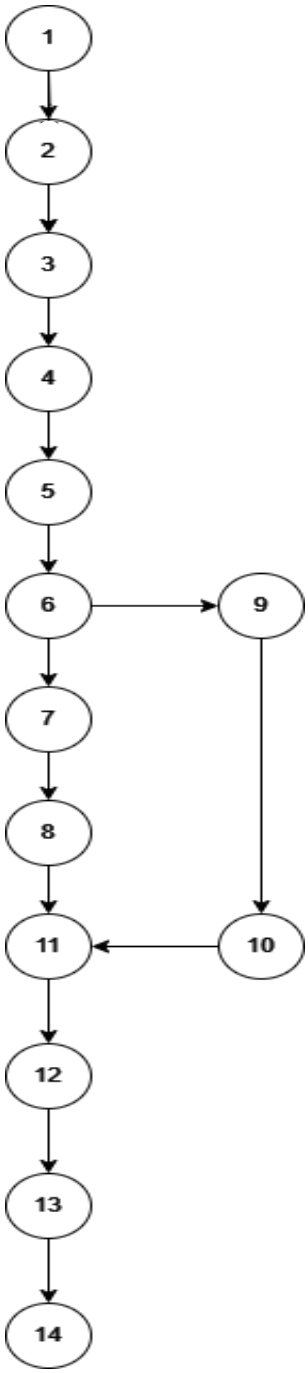
2. Proses Menampilkan di Aplikasi

a. *Flowchart*



Gambar 4.34 *Flowchart* Menampilkan di aplikasi

b. *Flowgraph*



Gambar 4.35 *Flowgraph Interface* Aplikasi

Dari gambar *Flowgraph* diatas dapat dilakukan proses perhitungan sebagai berikut :

Diketahui :

N (*node*) = 14 E (*edge*) = 15 P (Predikat node) = 1 +1

Penyelesaian :

V(G) = (E – N) + 2

= (14 - 14) + 2

= 2

Path 1 : 1-2-3-4-5-6-7-8-11-12-13-14

Path 2 : 1-2-3-4-5-6-9-10-11-12-13-14

c. *Grafik Matriks*

Tabel 4.19 *Grafik Matriks*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	E – 1
1		1													1-1=0
2			1												1-1=0
3				1											1-1=0
4					1										1-1=0
5						1									1-1=0
6							1		1						2-1=1
7								1							1-1=0
8											1				1-1=0
9										1					1-1=0
10											1				1-1=0
11												1			1-1=0
12													1		1-1=0
13														1	1-1=0
14															0
	Sum (E+1)														2

BAB V

PENUTUP

A. Kesimpulan

Penelitian ini merupakan bentuk pengembangan untuk dapat mendesain sebuah alat yang mampu menentukan kualitas air. Bentuk transformasi dari manual ke otomatis memberikan nilai efisien pada penggunaan alat. Alat ini menggunakan Sensor PH (Sensor PH SEN0161) dan sensor kekeruhan (*DFRobot*-SEN0189) yang dapat menjalankan fungsi sebagaimana mestinya ketika telah dilakukan kalibrasi sebagai langkah penyesuaian pembacaan sensor. Penyesuaian pembacaan dilakukan berdasarkan nilai tegangan yang didapat. Alat ini juga menggunakan *mikrokontroler* ESP 32 sebagai otaknya. Adapun bahasa program yang digunakan yaitu *HTML* & *Java Script*.

Setelah dilakukan pengujian pada alat ini, maka diperoleh kesimpulan sebagai berikut :

1. Hasil pengujian menunjukkan perubahan nilai kekeruhan dengan nilai awal 29,8 NTU (tidak layak konsumsi) menjadi 4,8 NTU (layak konsumsi) dengan rentan waktu siklus air selama 24 menit 56 detik.
2. Berdasarkan beberapa sampel pengujian, maka didapatkan *presentase eror* sebesar < 1% pada setiap percobaan.
3. Alat bekerja dengan memanfaatkan fungsi *if-else* dimana jika nilai ph <=8 & nilai kekeruhan <=5 maka alat akan mengaktifkan *selenoid* tetapi jika kondisi di luar itu alat akan mengaktifkan pompa.
4. Penggunaan *realtime database* membantu menampilkan nilai pembacaan sensor yang tampil di aplikasi dengan secara otomatis mengirim data sesuai dengan nilai pembacaan yang didapatkan sensor disertai dengan riwayat nilai dalam sebuah tabel

pada aplikasi, maka dari itu tingkat kestabilan jaringan juga dapat mempengaruhi kerja alat.

B. Saran

Untuk dapat membuat alat yang mampu bekerja dengan maksimal, maka perlu perencanaan dan perancangan yang matang. Hal ini begitu penting sebab pembuatan alat begitu rawan dengan korsleting pada komponen, maka salah satu yang paling penting adalah membaca data sheet setiap komponen agar fungsi komponen dapat berjalan sebagaimana mestinya dan menghindari potensi terjadinya kerusakan komponen, serta melakukan perencanaan yang maksimal agar ketika perancangan ke-sistematisan bekerja mampu terimplementasikan dan ketika alat tidak bekerja dengan maksimal langkah taktis yang perlu dilakukan adalah mencoba setiap komponen satu persatu untuk menemukan masalahnya.

DAFTAR PUSTAKA

Bawotong,V.T. Mamahit,D.J. & Sherwin (2015) Rancang Bangun Uninterruptible Power Supply Menggunakan Tampilan LCD Berbasis Mikrokontroler.

Fadil. A. (2021). Prototype Sistem Monitoring Suhu dan Kelembapan pada gedung Walet berbasis web. Skripsi. Universitas Muhammadiyah Parepare.

Herliana,A. Rasyid P.M. (2016). Sistem Informasi Monitoring Pengembangan Software Pada Tahap Development Berbasis WEB.

Iqtimal,Z Sara,I.D &Syahrizal (2018) Aplikasi Sistem Tenaga Surya Sebagai Sumber Tenaga Listrik Pompa Air.

Junaidi,A. (2015) Internet Of Things, Sejarah Teknologi dan Penerapan Review.

Laisina,L.H & Perman,A. (2017) Pemanfaatan Sensor PIR (Passive Infrared Receiver) dan Mikrokontroler Atmega 16 Untuk Efisiensi Pemakaian Air Wudhu.

Mila, D., Roihan, A., & Permana, A. (2016) Monitoring Kebocoran Gas Menggunakan Mikrokontroler Arduino Uno Dan Esp8266 Berbasis Internet Of Things. Academia.edu.

Muliadi, Imran, A. Rasul,M. (2020) Pengembangan Tempat Sampah Pintar Menggunakan Esp32 Jurnal Media Elektrik.

Ngarfifuddin, M. Susilo & Sunarmo (2017) Penerapan Rancang Bangun Ph Meter Berbasis Arduino Pada Mesin Pencuci Film Radiografi Sinar-X.

Novia Ajeng Ari, (2019) Alat Pengolahan Air Baku Sederhana Dengan Sistem Filtrasi. Widyakala. Volume 6.

Nugraha, D.W., Subito, M., & Astria, F. (2014) Rancang Bangun Alat Ukur Ph Dan Suhu Berbasis Short Message Service (Sms) Gateway.

Qalit,A. Fardian. Rahman, A. (2017) Rancang Bangun Prototipe Pemantauan Kadar pH dan Kontrol Suhu Serta Pemberian Pakan Otomatis pada Budidaya Ikan Lele Sangkuriang Berbasis IoT.

Rachmansyah, F. Utomo, F.R. Sumardi. (2014) Perancangan Dan Penerapan Alat Ukur Kekeruhan Air Menggunakan Metode Nefelometrik Pada Instalasi. Pengolahan Air Dengan Multi Media Card (Mmc) Sebagai Media Penyimpanan (Studi Kasus Di PDAM Jember).

Rosaly Rizqi, & Prasetyo Andy. (2019) Pengertian Flowchart Beserta Fung Simbol-simbol Flowchart yang Paling Umum Digunakan.

Saleh,M & Haryanti,M. (2017) Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay.

Wicaksana, A.S (2018) ,Perancangan Alat Ukur Kekeruhan Pada Air Kolam Menggunakan Optocoupler(Sensor Turbidity)Berbasis Arduino.