

TUGAS AKHIR

MONITORING TEMPERATURE, GETARAN, DAN KECEPATAN MOTOR INDUKSI 1 PHASE BERBASIS WEBSITE



Disusun Oleh :

ACHMAD SHOLIKHUL FAJRI

NIM.1814111018

**BIDANG STUDI SISTEM TENAGA
PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS BHAYANGKARA SURABAYA
2023**

LEMBAR PERSETUJUAN
MONITORING TEMPERATURE, GETARAN, DAN
KECEPATAN MOTOR INDUKSI 1 PHASE
BERBASIS WEBSITE

Diajukan Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana
(S1) Teknik Elektro Universitas Bhayangkara Surabaya

Disusun Oleh :

ACHMAD SHOLIKHUL FAJRI
NIM : 1814111018

Diperiksa dan disetujui oleh :

Pembimbing I

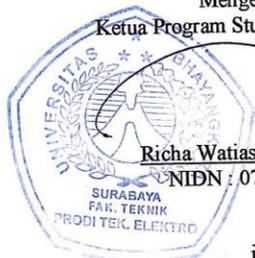


Prof. Dr. Ir. Saidah, MT.
NIDN : 0712066101

Mengetahui,
Ketua Program Studi Teknik Elektro



Richa Watiasih, S.T., M.T.
NIDN : 0714047601



iii

LEMBAR PENGESAHAN

**MONITORING TEMPERATURE, GETARAN, DAN
KECEPATAN MOTOR INDUKSI 1 PHASE BERBASIS
WEBSITE**

**Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar
Sarjana (S1) Teknik Elektro Universitas Bhayangkara Surabaya**

Disusun Oleh :

ACHMAD SHOLIKHUL FAJRI
NIM : 1814111018

Tanggal Ujian : 10 Juli 2023
Periode Wisuda :

Pembimbing I

Prof. Dr. Ir. Saidah MT.
NIDN. 0712066101

Disetujui oleh Tim penguji Tugas Akhir

Penguji I

Penguji II

Penguji III

Agus Kiswanto, ST., MT.
NIDN. 0715087101

Dr. Bambang Purwahyudi, ST., MT.
NIDN : 0025057001

Dr. Hasti Afianti, ST., MT.
NIDN : 0730037402

Mengetahui,
Dekan Fakultas Teknik

Dr. Muhammad Ghazi, S.T., M.T.
NIDN : 0028127003

Menyetujui,
Ketua Program Studi Teknik Elektro

Richa Watiyah, ST., MT.
NIDN : 0714047601

LEMBAR PERNYATAAN KEASLIAN PENELITIAN

Yang bertandatangan di bawah ini :

Nama : Achmad Sholikhul Fajri
Tempat/Tanggal Lahir : Ngawi, 3 Mei 2000
NIM : 1814111018
Fakultas / Jurusan : Teknik / Teknik Elektro

Dengan ini saya menyatakan bahwa Tugas Akhir kami dengan judul "MONITORING TEMPERATURE, GETARAN, DAN KECEPATAN MOTOR INDUKSI 1 PHASE BERBASIS WEBSITE" beserta seluruh isinya adalah karya saya sendiri dan bukan merupakan karya tulis orang lain, baik sebagian maupun seluruhnya, kecuali dalam bentuk kutipan yang telah disebutkan sumbernya.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila kemudian di temukan adanya pelanggaran terhadap apatika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini maka saya siap menanggung segala bentuk resiko/sanksi yang berlaku.

Surabaya, 10 Juli 2023
Yang membuat pernyataan,



(Achmad Sholikhul Fajri)
NIM : 1814111018

ABSTRAK

Peran motor induksi di dunia industri sangatlah penting. Dalam prosesnya banyak motor induksi yang dinyalakan secara terus menerus yang akan dapat menyebabkan beberapa permasalahan serius pada motor induksi tersebut antara lain mudah panas, kecepatan mulai menurun dan kondisi shaft yang mulai bergoyang krn bearing sudah rusak. Dari banyak kondisi kerusakan yang mungkin terjadi tersebut maka akan menyebabkan banyak hal serius dan dapat menghasilkam efek beruntut dari suatu proses baik produksi maupun manufaktur. Dalam menanggulangi hal tersebut sangat dibutuhkan adanya data kondisi motor yang dapat dipelajari sebagai referensi untuk membuat rencana perawatan motor secara berkala agar tidak sampai ada motor yang bermasalah yang akan mengganggu jalannya suatu proses. Dengan melihat problem yang ada maka dibuat suatu system monitoring terhadap motor yang dapat dipantau kapanpun secara online. Dengan memanfaatkan sensor temperature DS1820, photoelectric untuk tachometer dan akselerometer MPU6050 untuk pembacaan getaran. Serta sistem *Internet of Think (IoT)* yang mampu memberi notifikasi dan data kepada tim perawatan motor.

Kata Kunci : Motor induksi, getaran, temperature, kecepatan, *Internet of Think (IoT)*

ABSTRACT

The role of induction motors in the industrial world is very important. In the process, many induction motors are turned on continuously which will cause some serious problems with the induction motor, including easy overheating, speed begins to decrease and the condition of the shaft that begins to sway because the bearing has been damaged. Of the many damage conditions that may occur, it will cause many serious things and can produce demanding effects from a process, both production and manufacturing. In overcoming this, it is very necessary to have motor condition data that can be studied as a reference to make a motorcycle maintenance plan regularly so that there is no problematic motorbike that will interfere with the course of a process. By looking at the existing problems, a monitoring system for the motorcycle is made that can be monitored at any time online. By utilizing the DS1820 temperature sensor, photoelectric for the tachometer and accelerometer MPU6050 for vibration reading. As well as an Internet of Think (IoT) system that is able to provide notifications and data to the motor maintenance team.

Keywords: *Induction motor, vibration, temperature, speed, Internet of think (IoT)*

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT, atas limpaha nrahmat dan hidayah-Nya sehingga peneliti dapat menyelesaikan Tugas Akhir yang berjudul **“MONITORING TEMPERATURE, GETARAN, DAN KECEPATAN MOTOR INDUKSI 1 PHASE BERBASIS WEBSITE”** Oleh karena itu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada :

1. Allah SWT yang telah melimpahkan Rahmat dan KaruniaNya
2. Ibu Prof. Dr.Ir. Saidah, MT. selaku Pembimbing Tugas Akhir
3. Bapak Dr. Mohammad Ghozi, ST., MT., selaku Dekan Fakultas Teknik Universitas Bhayangkara Surabaya
4. Ibu Richa Watiasih ST., MT., selaku Ketua Program Studi Teknik Elektro Universitas Bhayangkara Surabaya
5. Bapak Agus Kiswantonono ST., MT., selaku Koordinator Tugas Akhir
6. Bapak Heru selaku staff TU TE yang telah membantu melengkapi administrasi tugas akhir
7. Orang tua dan keluarga dan kerabat yang senantiasa memberi dukungan kepada kami, mendoakan kami kapanpun dan dimanapun kami berada

Peneliti menyadari sepenuhnya bahwa dalam penulisan Tugas Akhir ini masih banyak kekurangan dan keterbatasan, oleh karena itu peneliti mengharapakan saran yang membangun agar tulisan ini bermanfaat bagi perkembangan ilmu pengetahuan kedepan.

Surabaya, 10 Juli 2023
Penulis

Achmad Sholikhul Fajri

DAFTAR ISI

COVER.....	i
LEMBAR PERSETUJUAN.....	iii
LEMBAR PENGESAHAN.....	v
LEMBAR PERNYATAAN KEASLIAN.....	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah.....	1
1.4 Tujuan Penelitian.....	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Motor Induksi.....	5
2.2 Sensor DS1820.....	7
2.3 NodeMCU ESP32.....	8
2.4 Arduino IDE.....	10
2.5 Internet of Things.....	16
2.6 Thingspeak.....	21
2.7 Tachometer atau RPM meter.....	24
2.8 Accelerometer MPU6050.....	26
BAB III PERANCANGAN SISTEM.....	29
3.1 Perencanaan sistem.....	29
3.2 Perencanaan dan perancangan perangkat keras elektronik ..	30
3.2.1 Node MCU.....	31
3.2.2 DS1820.....	32
3.2.3 Photoelectric.....	33
3.2.4 MPU6050.....	34
3.3 Thingspeak.....	36
BAB IV ANALISA DAN PENGUJIAN.....	41
4.1 Pengujian sensor temperature.....	39
4.2 Pengujian sensor kecepatan motor.....	40
4.3 Pengujian pembacaan getaran motor.....	42
4.4 Hasil monitoring thingspeak.....	44

BAB V KESIMPULAN DAN SARAN	51
5.1 Kesimpulan	51
5.2 Saran	51
DAFTAR PUSTAKA	53
LAMPIRAN	55
SURAT BIMBINGAN TUGAS AKHIR	67
BIODATA PENULIS	69
FORM BIMBINGAN TUGAS AKHIR.....	71

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi kontruksi motor induksi	5
Gambar 2.2 Medan magnet pada stator	6
Gambar 2.3 Sensor DS1820	7
Gambar 2.4 Modul ESP32 devkit-C	8
Gambar 2.5 Tampilan software arduino IDE.....	7
Gambar 2.6 Tampilan depan website thingspeak.....	21
Gambar 2.7 Tampilan sign up thingspeak.....	22
Gambar 2.8 Tampilan saat akan membuat chanel baru.....	23
Gambar 2.9 Pengaturan chanel baru	23
Gambar 2.10 Setting API key.....	24
Gambar 2.11 Photoelectric untuk tachometer.....	25
Gambar 2.12 Ilustrasi cara kerja tachometer photoelectric.....	26
Gambar 2.13 Accelerometer.....	27
Gambar 3.1 Blok diagram sistem.....	29
Gambar 3.2 Flowchart sistem monitoring	30
Gambar 3.3 Rancangan skematik	31
Gambar 3.4 Hasil jadi motor induksi dengan perangkat monitoring.....	32
Gambar 3.5 Desain layout PCB rangkaian monitoring	33
Gambar 3.6 Konfigurasi pin I/O ESP32 ke interface	33
Gambar 3.7 Aplikasi sensor temperature DS1820 pada motor	34
Gambar 3.8 Aplikasi photoelectric untuk pengukuran kecepatan.....	35
Gambar 3.9 Aplikasi sensor getaran MPU6050 pada motor.....	36
Gambar 3.10 Tampilan data temperature motor 1 pada thingspeak.....	37
Gambar 3.11 Tampilan data getaran motor 1 pada thingspeak.....	37
Gambar 3.12 Tampilan data kecepatan motor 1 pada thingspeak.....	38
Gambar 3.13 Tampilan data temperature motor 2 pada thingspeak.....	38
Gambar 3.14 Tampilan data getaran motor 2 pada thingspeak.....	39
Gambar 3.15 Tampilan data kecepatan motor 2 pada thingspeak.....	39
Gambar 4.1 Grafik perbandingan pembacaan thermometer dengan sensor getaran DS1820.....	42
Gambar 4.2 Grafik perbandingan aktual kecepatan vs tachometer	45
Gambar 4.3 Tampilan hasil grafis temperature motor 1 pada thingspeak	47
Gambar 4.4 Tampilan hasil grafis getaran motor 1 pada thingspeak	48
Gambar 4.5 Tampilan hasil grafis kecepatan motor 1 pada thingspeak	48

Gambar 4.6 Tampilan hasil grafis temperature motor 2 pada thingspeak49
Gambar 4.7 Tampilan hasil grafis getaran motor 2 pada thingspeak49
Gambar 4.8 Tampilan hasil grafis kecepatan motor 2 pada thingspeak 49

DAFTAR TABEL

Tabel 4.1 Hasil perbandingan sensor temperature DS1820 dengan thermometer	42
Tabel 4.2 Hasil perbandingan aktual kecepatan dengan tachometer	44
Tabel 4.3 Hasil pengambilan data perbandingan getaran dan kecepatan dengan motor 1 dan motor 2	46

HALAMAN INI SENGAJA DIKOSONGKAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Motor induksi sangatlah dibutuhkan mayoritas pelaku industri, baik kecil menengah maupun besar. Sistem pengecekan berkala pada suatu motor sangatlah diperlukan untuk mencegah terjadinya macet produksi karena kerusakan motor.

Permasalahan pada sistem pemeliharaan dalam suatu industri menjadi latar belakang dari penelitian ini. Dimana sistem pencatatan dari monitoring suatu motor masih banyak yang dilakukan secara manual, belum *realtime*, tidak tercatat dengan baik, bahkan belum ada pemeliharaan. Sehingga kondisi kerusakan motor masih sulit dideteksi dan dapat mempengaruhi jalannya proses dari suatu industri.

Dengan adanya permasalahan tersebut, maka banyak industri yang seharusnya sudah menerapkan suatu rencana pemeliharaan (*Prediktive maintenance*) yang datanya dapat termonitor secara online melalui website. Oleh karena itu, akan dirancang sebuah ***Monitoring Temperature, Getaran, dan Kecepatan Motor Induksi 1 Phase Berbasis Website***.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang maka dapat dirumuskan beberapa permasalahan antara lain sebagai berikut :

1. Bagaimana cara akses sensor kecepatan motor?
2. Bagaimana cara akses sensor temperature?
3. Bagaimana cara akses sensor getaran?
4. Bagaimana keefektifan *monitoring* dalam penelitian ini dalam penentuan *prediktive maintenance*?

1.3 Batasan Masalah

Untuk membatasi ruang lingkup tugas akhir supaya lebih terarah dan penyesuaian dengan kemampuan yang ada maka pembahasan dibatasi beberapa hal, antara lain:

1. Menggunakan motor induksi pompa air 1 phase.
2. *Web server* yang digunakan adalah web server tidak berbayar yaitu thingspeak.

1.4 Tujuan Penelitian

Sesuai dengan uraian latar belakang, maka tujuan yang ingin dicapai dari penelitian ini adalah :

1. Mengetahui kondisi motor secara *realtime*.
2. Membantu tim pemeliharaan suatu industri untuk menentukan penjadwalan suatu proses pemeliharaan

1.5 Metodologi Penelitian

Dalam pengerjaan proyek akhir ini penulis menerapkan metodologi melalui beberapa tahap sebagai berikut :

1.5.1 Studi Literatur

Dalam proses perancangan dan pengujian sistem, perlu mempelajari tentang pengukuran getaran, temperature, kecepatan, dan spesifikasi dari motor induksi. Literatur didapatkan dari buku – buku acuan, makalah-makalah acuan, dan beberapa forum diskusi di internet.

1.5.2 Perancangan Sistem

Setelah mempelajari materi-materi yang ada, dimulailah perancangan sistem, penyediaan komponen, uji coba sebelum pembuatan *hardware* elektronik dan sesudah pembuatan *hardware* elektronik, mengujinya dan setelah itu pembuatan media observasi yaitu berupa motor induksi 1 phase yang dimodifikasi dan disesuaikan terhadap *plant* yg dibuat.

Proyek akhir ini dibuat melalui beberapa tahap, diantaranya mendesain dan merancang alat, membuat *hardware* elektronik, pengecekan *hardware* elektronik, integrasi media tanam dan *hardware*

elektronik, pembuatan program, pembuatan GUI di *thingspeak*, integrasi seluruh sistem.

1.5.3 Pembuatan *Software* sistem

Setelah perancangan sistem selesai, selanjutnya perancangan perangkat lunak (*software*) dari sistem yang dikontrol. Perancangan perangkat lunak yaitu, membuat algoritma untuk mengakses sensor getaran, kecepatan, temperature dan membuat tampilan GUI *thingspeak* sebagai perekam data, integrasi sistem sensor dan motor induksi.

1.5.4 Pembuatan *Hardware* Sistem

Pada tahap ini dilakukan pembuatan *hardware system* yang terbagi menjadi 2 yaitu elektronik dan peraga motor induksi 1 phase yang pada akhirnya juga diuji. Proyek Akhir ini menggunakan motor induksi 1 phase dengan daya 125watt dan kecepatan nominal sebesar 2850 rpm. dan 1400 rpm Motor induksi tersebut kemudian dimodifikasi dengan pemasangan sensor kecepatan menggunakan tachometer, sensor getaran menggunakan accelerometer, dan sensor temperature DS1820 yang ditanam dalam body motor.

Sedangkan untuk *hardware* elektronik alat ini meliputi bagian-bagian seperti berikut.

- NodeMCU ESP32
- DS1820
- Accelerometer
- Photoelectric untuk tachometer

1.5.5 Pengujian Sistem, Integrasi dan Pengambilan Data

Pada tahap ini dilakukan pengujian tiap-tiap bagian dari *system* yang sebelumnya dibuat, setelah dirasa siap selanjutnya seluruh sistem tersebut diintegrasikan dengan rapi dan siap untuk diambil data. Pengujian Proyek Akhir dilakukan untuk mengetahui kelebihan dan kekurangan dari sistem sebagai referensi untuk perbaikan dan pengembangan alat untuk kemudian hari. Untuk kemudian sistem secara keseluruhan diuji kembali sehingga sistem bekerja dengan baik. Berdasarkan tahapan ini dilakukan pula pengambilan data-data yang dibutuhkan untuk dicantumkan pada laporan Proyek Akhir.

1.5.6 Pembuatan Laporan Proyek Akhir

Pada tahap ini dilakukan pembuatan atau penulisan laporan buku Proyek Akhir. Pada laporan tersebut dijelaskan mengenai semua hal yang berkaitan dengan pengerjaan Proyek Akhir, seperti penjelasan komponen yang dipakai, proses pembuatan alat, sistem kerja alat, data-data hasil pengujian alat, dan lain sebagainya. Diharapkan dengan penulisan laporan ini dapat bermanfaat sebagai bahan acuan dalam pembuatan alat serupa diwaktu yang akan datang.

1.6 Sistematika Penulisan

Sistematika dalam pembahasan penyusunan Proyek Akhir direncanakan sebagai berikut:

- Pada Bab I berisikan latar belakang pembuatan alat pada proyek akhir, tujuan yang ingin dicapai, batasan masalah pada proyek akhir, serta sistematika pembahasan.
- Bab II meliputi literatur-literatur, teori dasar, serta referensi yang berguna sebagai acuan dan landasan dalam perencanaan dan pembuatan proyek akhir.
- Pada Bab III membahas tentang perencanaan dan pembuatan perangkat lunak maupun perangkat keras dari proyek akhir.
- Pada Bab IV berisikan pengujian dan analisa hasil yang diperoleh.
- Pada Bab V berisikan kesimpulan dan juga saran serta masukan setelah melihat analisa dari pengerjaan alat dan pengujian alat.

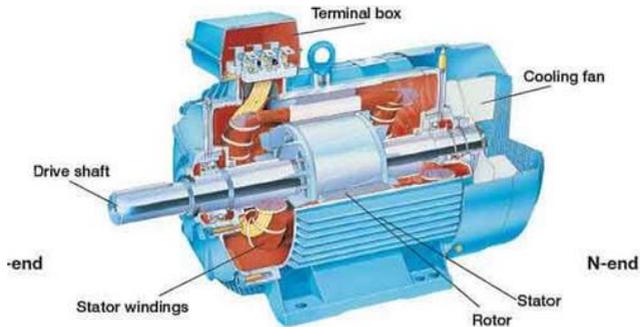
BAB II

TINJAUAN PUSTAKA

2.1 Motor Induksi

Motor Induksi atau mesin juga disebut sebagai Mesin Asinkron. Kata Asynchronous berarti bahwa mesin tidak pernah berjalan dengan kecepatan sinkron. Motor induksi terutama terdiri dari dua jenis. Itu bisa berupa motor induksi satu phase atau tiga phase.

Motor induksi satu phase biasanya dibangun dalam ukuran kecil (hingga 3 HP). Motor induksi tiga phase adalah motor AC yang paling umum digunakan di industri. Mereka sederhana dalam konstruksi, dapat diandalkan.



Gambar 2.1 Ilustrasi konstruksi Motor Induksi

Motor Induksi satu Phase dan Tiga Phase dibedakan pada berbagai faktor dalam artikel ini seperti Supply motor beroperasi, torsi motor awal, pemeliharaan, fitur, efisiensi motor, faktor kekuatan motor dan contoh di mana kedua motor digunakan.

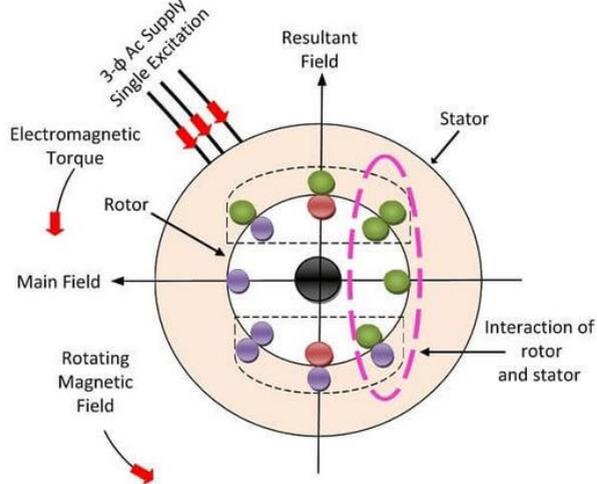
2.1.1 Prinsip kerja Motor Induksi

Motor yang bekerja berdasarkan prinsip induksi elektromagnetik dikenal sebagai motor induksi. Induksi elektromagnetik adalah fenomena di mana gaya gerak listrik menginduksi melintasi konduktor listrik ketika ditempatkan dalam medan magnet yang berputar.

Stator dan rotor adalah dua bagian penting dari motor. Stator adalah bagian yang diam, dan ia membawa belitan yang tumpang tindih

sementara rotor membawa belitan utama atau medan. Gulungan stator sama-sama dipindahkan satu sama lain dengan sudut 120° .

Ketika pasokan tiga fase diberikan ke stator, medan magnet berputar diproduksi di atasnya. Gambar di bawah ini menunjukkan medan magnet berputar yang diatur di stator.



Gambar 2.2 Medan magnet pada stator

Prinsip kerja motor induksi adalah bahwa medan magnet yang berputar mengarah ke arah berlawanan arah jarum jam. Medan magnet yang berputar memiliki polaritas yang bergerak. Polaritas medan magnet bervariasi dengan memperhatikan setengah siklus positif dan negatif dari suplai. Perubahan polaritas membuat medan magnet berputar.

Konduktor rotor tidak bergerak. Konduktor stasioner ini memotong medan magnet berputar dari stator, dan karena induksi elektromagnetik, EMF menginduksi dalam rotor. EMF ini dikenal sebagai EMF yang diinduksi rotor, dan ini disebabkan oleh fenomena induksi elektromagnetik.

Konduktor rotor dihubungkan pendek baik oleh cincin akhir atau dengan bantuan resistansi eksternal. Gerakan relatif antara medan magnet yang berputar dan konduktor rotor menginduksi arus dalam konduktor rotor. Ketika arus mengalir melalui konduktor, fluks menginduksi padanya. Arah fluks rotor sama dengan arah arus rotor.

Sekarang kita memiliki dua fluks satu karena rotor dan yang lain karena stator. Aliran-aliran ini saling berinteraksi. Di satu ujung konduktor fluks saling membatalkan, dan di ujung lainnya, kerapatan fluks sangat tinggi.

Dengan demikian, fluks densitas tinggi mencoba untuk mendorong konduktor rotor menuju daerah fluks densitas rendah. Fenomena ini menginduksi torsi pada konduktor, dan torsi ini dikenal sebagai torsi elektromagnetik.

Arah torsi elektromagnetik dan medan magnet yang berputar adalah sama. Dengan demikian, rotor mulai berputar ke arah yang sama dengan medan magnet yang berputar.

Kecepatan rotor selalu kurang dari medan magnet yang berputar atau kecepatan sinkron. Rotor mencoba berlari dengan kecepatan rotor, tetapi selalu terlepas. Dengan demikian, motor tidak pernah berjalan pada kecepatan medan magnet yang berputar, dan ini adalah alasan karena motor induksi juga dikenal sebagai motor asinkron.

2.2 Sensor DS1820

Sensor yang berfungsi untuk mendeteksi suhu ruangan yang merupakan jenis seri sensor terbaru dari keluaran produsen Maxim. Sensor ini dapat mendeteksi suhu dari -55°C sampai 125°C dengan tingkat keakuratan ($\pm 0.5^{\circ}\text{C}$) dan dengan resolusi 9 – 12-bit. Apabila terdapat banyak sensor yang disusun secara paralel data dari keluaran setiap sensor tersebut dapat dibaca hanya dengan menggunakan 1 kabel data atau (oneWire) saja



Gambar 2.3 sensor DS1820

2.2.1 Fitur DS1820

Berikut ini adalah spesifikasi dan fitur dari sensor DS1820:

- Interface menggunakan 1-wire sebagai komunikasi data
- Terdapat pengenalan unik 64 bit pada setiap sensor
- Dapat mengukur suhu dari range -55°C sampai $+125^{\circ}\text{C}$
- Keakuratan sensor yaitu $\pm 0.5^{\circ}\text{C}$ pada suhu -10°C sampai $+85^{\circ}\text{C}$
- Resolusi sensor yaitu 9 – 12 bit
- Dapat mengkonversi data suhu 12-bit (digital word) hanya membutuhkan waktu 750 ms
- Mempunyai konfigurasi alarm yang dapat disetting
- Pengaplikasiannya yaitu pada sistem industri, termometer, atau sistem apapun yang memerlukan pembacaan suhu

2.3 NodeMCU ESP32

Mikrokontroler ESP32 dibuat oleh perusahaan bernama Espressif Systems. Salah satu kelebihan yang dimiliki oleh ESP32 yaitu sudah terdapat Wi-Fi dan Bluetooth di dalamnya, sehingga akan sangat memudahkan ketika kita belajar membuat sistem IoT yang memerlukan koneksi wireless. Mikrokontroler ESP32 memiliki keunggulan yaitu sistem berbiaya rendah, dan juga berdaya rendah dengan modul WiFi yang terintegrasi dengan chip mikrokontroler serta memiliki bluetooth dengan mode ganda dan fitur hemat daya menjadikannya lebih fleksibel.



Gambar 2.4 Modul ESP32 devkit-C

2.3.1 Spesifikasi

ESP32 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. ESP32 sendiri tidak jauh berbeda dengan ESP8266 yang familiar di pasaran, hanya saja ESP32 lebih kompleks dibandingkan ESP8266. Berikut ini merupakan spesifikasi yang dimiliki oleh mikrokontroler ESP32 :

- Prosesor: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz.
- Memori: 520 KB SRAM.
- Wireless connectivity: Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR and BLE (shares the radio with Wi-Fi).
- Peripheral I/O: 12-bit SAR ADC (up to 18 channels), 2x 8-bit DACs, 10x touch sensors (capacitive sensing GPIOs), 4x SPI, 2x I2S interfaces, 2x I2C interfaces, 3x UART, SD/SDIO/CE-ATA/MMC/eMMC host controller, SDIO/SPI slave controller, Ethernet MAC interface, CAN bus 2.0, infrared remote controller (TX/RX, up to 8 channels), motor PWM, LED PWM (up to 16 channels), hall effect sensor, ultra low power analog pre-amplifier.
- Security : IEEE 802.11 standard security, secure boot, flash, encryption, 1024-bit, OTP (up to 768-bit for customers), cryptographic hardware acceleration (AES, SHA-2, RSA, ECC), random number generator (RNG).

2.3.2 Implementasi

Mikrokontroler ESP32 banyak digunakan dan sangat mendukung untuk pembuatan sistem aplikasi Internet of Things (IoT). Berikut ini merupakan contoh penggunaannya, sebagai berikut :

- Smart Security, Salah satu contoh perangkat smart security yang paling umum adalah kunci rumah yang menggunakan biometrik seperti sidik jari atau kode otentikasi. kegunaan alat tersebut untuk mengurangi pembobolan, kemalingan rumah, dan memperketat keamanan rumah dari orang yang ingin berbuat jahat. Selain itu juga untuk mengatasi beberapa 'kecelakaan' kecil seperti kehilangan kunci, akses rumah oleh tamu, dan akses ruangan yang memerlukan otorisasi khusus seperti kamar orang tua.
- Smart City, Di skala yang lebih luas, teknologi IoT juga menyentuh urusan tata kota. kegunaannya smart city ini untuk mengintegrasikan semua permasalahan kota menjadi satu. contohnya seperti CCTV,

lampu lalu lintas, sampai integrasi sistem transportasi dalam kota. Semuanya bisa diakses langsung lewat smartphone yang terhubung ke internet.

2.4 Arduino IDE

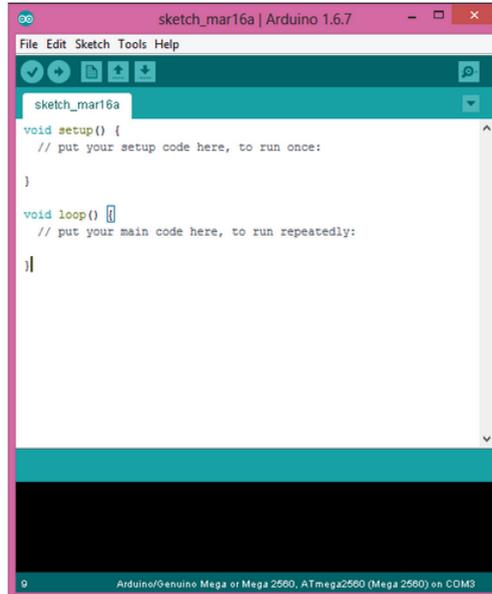
IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui sintaks pemrograman[6]. 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.

2.4.1 Menulis Sketch

Program yang ditulis dengan menggunakan 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 Software Arduino IDE, menunjukkan board yang terkonfigurasi beserta COM Ports yang digunakan.



Gambar 2.5 tampilan dari Software Arduino IDE.

Tombol- tombol pada toolbar, sebagai berikut:

- *Verify*
Berfungsi untuk melakukan checking kode yang kamu buat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum
- *Upload*
Berfungsi untuk melakukan kompilasi program atau kode yang kamu buat menjadi bahasa yang dapat dipahami oleh mesin alias si Arduino.
- *New*
Berfungsi untuk membuat *Sketch* baru
- *Open*
Berfungsi untuk membuka *sketch* yang pernah kamu buat dan membuka kembali untuk dilakukan editing atau sekedar upload ulang ke Arduino.
- *Save*
Berfungsi untuk menyimpan *Sketch* yang telah kamu buat.
- *Serial Monitor*
Berfungsi untuk membuka 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.

Fitur-fitur pada menuubar, sebagai berikut

- File

Memiliki sub menu sebagai berikut:

- ✓ **New**, berfungsi untuk membuat membuat sketch baru dengan bare minimum yang terdiri void setup() dan void loop().
- ✓ **Open**, berfungsi membuka sketch yang pernah dibuat di dalam drive.
- ✓ **Open Recent**, merupakan menu yang berfungsi mempersingkat waktu pembukaan file atau sketch yang baru-baru ini sudah dibuat.
- ✓ **Sketchbook**, berfungsi menunjukkan hirarki *sketch* yang kamu buat termasuk struktur foldernya.
- ✓ **Example**, berisi contoh-contoh pemrograman yang disediakan pengembang Arduino, sehingga kamu dapat mempelajari program-program dari contoh yang diberikan.
- ✓ **Close**, berfungsi menutup jendela Arduino IDE dan menghentikan aplikasi.
- ✓ **Save**, berfungsi menyimpan *sketch* yang dibuat atau perubahan yang dilakukan pada *sketch*
- ✓ **Save as...**, berfungsi menyimpan *sketch* yang sedang dikerjakan atau *sketch* yang sudah disimpan dengan nama yang berbeda.
- ✓ **Page Setup**, berfungsi mengatur tampilan page pada proses pencetakan.
- ✓ **Print**, berfungsi mengirimkan file sketch ke mesin cetak untuk dicetak.
- ✓ **Preferences**, disini kam dapat merubah tampilan *interface* IDE Arduino.
- ✓ **Quit**, berfungsi menutup semua jendela Arduino IDE. *Sketch* yang masih terbuka pada saat tombol *Quit* ditekan, secara otomatis akan terbuka pada saat Arduino IDE dijalankan.

- Edit

Memiliki sub menu sebagai berikut:

- ✓ **Undo/Redo**, berfungsi untuk mengembalikan perubahan yang sudah dilakukan pada *Sketch* beberapa langkah mundur dengan *Undo* atau maju dengan *Redo*.

- ✓ **Cut**, berfungsi untuk *remove* teks yang terpilih pada editor dan menempatkan teks tersebut pada *clipboard*.
- ✓ **Copy**, berfungsi menduplikasi teks yang terpilih kedalam editor dan menempatkan teks tersebut pada *clipboard*.
- ✓ **Copy for Forum**, berfungsi melakukan *copy* kode dari editor dan melakukan *formatting* agar sesuai untuk ditampilkan dalam forum, sehingga kode tersebut bisa digunakan sebagai bahan diskusi dalam forum.
- ✓ **Copy as HTML**, berfungsi menduplikasi teks yang terpilih kedalam editor dan menempatkan teks tersebut pada *clipboard* dalam bentuk atau format HTML. Biasanya ini digunakan agar code dapat *diembedded* pada halaman web.
- ✓ **Paste**, berfungsi menyalin data yang terdapat pada *clipboard*, kedalam editor.
- ✓ **Select All**, berfungsi untk melakukan pemilihan teks atau kode dalam halaman editor.
- ✓ **Comment/Uncomment**, berfungsi memberikan atau menghilangkan tanda // pada kode atau teks, dimana tanda tersebut menjadikan suatu baris kode sebagai komen dan tidak disertakan pada tahap kompilasi.
- ✓ **Increase/Decrease Indent**, berfungsi untuk mengurangi atau menambahkan indentasi pada baris kode tertentu. Indentasi adalah “tab”.
- ✓ **Find**, berfungsi memanggil jendela window *find and replace*, dimana kamu dapat menggunakannya untuk menemukan variabel atau kata tertentu dalam program atau menemukan serta menggantikan kata tersebut dengan kata lain.
- ✓ **Find Next**, berfungsi menemukan kata setelahnya dari kata pertama yang berhasil ditemukan.
- ✓ **Find Previous**, berfungsi menemukan kata sebelumnya dari kata pertama yang berhasil ditemukan.
- Sketch
 - Memiliki sub menu sebagai berikut:
 - ✓ **Verify/Compile**, berfungsi untuk mengecek apakah *sketch* yang kamu buat ada kekeliruan dari segi sintaks atau tidak. Jika tidak ada kesalahan, maka sintaks yang kamu buat akan dikompile kedalam bahasa mesin.
 - ✓ **Upload**, berfungsi mengirimkan program yang sudah dikompilasi ke Arduino Board.
 - ✓ **Upload Using Programmer**, menu ini berfungsi untuk menuliskan *bootloader* kedalam IC Mikrokontroler Arduino. Pada kasus ini

kamu membutuhkan perangkat tambahan seperti *USBasp* untuk menjembatani penulisan program *bootloader* ke IC Mikrokontroler.

- ✓ **Export Compiled Binary**, berfungsi untuk menyimpan file dengan ekstensi **.hex**, dimana file ini dapat disimpan sebagai arsip untuk di upload ke board lain menggunakan tools yang berbeda.
- ✓ **Show Sketch Folder**, berfungsi membuka folder *sketch* yang saat ini dikerjakan.
- ✓ **Include Library**, berfungsi menambahkan library/pustaka kedalam *sketch* yang dibuat dengan menyertakan sintaks `#include` di awal kode. Selain itu kamu juga bisa menambahkan library eksternal dari file **.zip** kedalam Arduino IDE.
- ✓ **Add File...**, berfungsi untuk menambahkan file kedalam *sketch* arduino (file akan dikopikan dari drive asal). File akan muncul sebagai tab baru dalam jendela *sketch*.

- Tools

Memiliki sub menu sebagai berikut:

- ✓ **Auto Format**, berfungsi melakukan pengatran format kode pada jendela editor
- ✓ **Archive Sketch**, berfungsi menyimpan *sketch* kedalam file **.zip**
- ✓ **Fix Encoding & Reload**, berfungsi memperbaiki kemungkinan perbedaan antara pengkodean peta karakter editor dan peta karakter sistem operasi yang lain.
- ✓ **Serial Monitor**, berfungsi membuka jendela serial monitor untuk melihat pertukaran data.
- ✓ **Board**, berfungsi memilih dan melakukan konfigurasi board yang digunakan.
- ✓ **Port**, memilih port sebagai kanal komunikasi antara software dengan hardware.
- ✓ **Programmer**, menu ini digunakan ketika kamu hendak melakukan pemrograman chip mikrokontroler tanpa menggunakan koneksi Onboard USB-Serial. Biasanya digunakan pada proses *burning bootloader*.
- ✓ **Burn Bootloader**, mengizinkan kamu untuk mengkopikan program *bootloader* kedalam IC mikrokontroler

- Help

Disini kamu bisa mendapatkan bantuan terhadap kegalauanmu mengenai pemrograman. Menu help berisikan file-file dokumentasi yang berkaitan dengan masalah yang sering muncul, serta penyelesaiannya. Selain itu pada menu help juga diberikan link untuk

menuju Arduino Forum guna menanyakan serta mendiskusikan berbagai masalah yang ditemukan.

2.4.2 Sketchbook

Arduino Software IDE, menggunakan konsep *sketchbook*, dimana *sketchbook* menjadi standar peletakan dan penyimpanan file program. Sketch yang telah kamu buat dapat dibuka dengan dari **File -> Sketchbook**, atau dengna menu **Open**.

2.4.3 Tabs, Multiple Files, dan Compilations

Mekanisme ini mengijinkan kamu untuk melakukan menejemen sketch, dimana lebih dari satu file dibuka dalam tab yang berbeda.

2.4.4 Uploading

Merupakan mekanisme untuk mengkopikan file **.hex** atau file hasil kompilasi kedalam IC mikrokontroler Arduino. Sebelum melakukan *uploading*, yang perlu kamu pastikan adalah jenis board yang kamu gunakan dan COM Ports dimana keduanya terletak pada menu **Tools -> Board** dan **Tools -> Port**.

2.4.5 Library

Library/ Pustaka merupakan file yang memberikan fungsi ekstra dari sketch yang kamu buat, semisal agar Arduino dapat bekerja dengan *hardware* tertentu dan melakukan proses manipulasi data. Untuk menginstal *Library* pihak ketiga alias *Library* bukan dari Arduino, dapat dilakukan dengan Library Manager, Import file **.zip**, atau kopi paste secara manual di folder libraries pada Documents di platform Windows.

2.4.6 Serial Monitor

Serial monitor merupakan suatu jendela yang menunjukkan data yang dipertukaran antara arduino dan komputer selama beroperasi, sehingga kamu bisa menggunakan serial monitor ini untuk menampilkan nilai hasil operasi atau pesan *debugging*. Selain melihat data, kamu juga bisa mengirimkan data ke Arduino melalui serial monitor ini, caranya

dengan memasukkan data pada *text box* dan menekan tombol *send* untuk mengirimkan data. Hal penting yang harus kamu perhatikan adalah menyamakan *baudrate* antara serial monitor dengan Arduino board. Untuk menggunakan kemampuan komunikasi serial ini, pada Arduino, di bagian fungsi void setup(), diawali dengan instruksi **Serial.begin** diikuti dengan nilai *baudrate*.

2.4.7 Preferences

Preferences mengatur tentang beberapa hal dalam penggunaan Arduino Software IDE, seperti ukuran font, lokasi dimana menyimpan sketcbok, bahasa yang digunakan pada Arduino Software IDE, dan masih banyak lagi. Kamu bisa mengatur preferences pada menu file yang dapat dijumpai pada platform Windows dan Linux.

2.4.8 Language Support

Language Support merupakan pilihan bahasa yang dapat disesuaikan pada Software Arduino IDE. Bahasa Indonesia sudah ada loh. Language Support ini dapat ditemukan pada menu **file -> preferences** atau dengan menekan **Ctrl+Comma**

2.5 Internet Of Things

Internet of things adalah suatu konsep atau program dimana sebuah objek memiliki kemampuan untuk mentransmisikan atau mengirimkan data melalui jaringan tanpa menggunakan bantuan perangkat komputer dan manusia. Internet of things atau sering disebut dengan IoT saat ini mengalami banyak perkembangan.

Perkembangan IoT dapat dilihat mulai dari tingkat konvergensi teknologi nirkabel, microelectromechanical (MEMS), internet, dan QR (*Quick Responses*) Code. IoT juga sering diidentifikasi dengan RFID (*Radio Frequency Identification*) sebagai metode komunikasi.

Selain itu, juga mencakup teknologi berbasis sensor, seperti teknologi nirkabel, QR Code yang sering kita jumpai. Kemampuan dari IoT sendiri tidak perlu diragukan lagi. Banyak sekali teknologi yang telah menerapkan sistem IoT, sebagai contoh sensor cahaya, sensor suara dari teknologi Google terbaru, yaitu Google Ai, dan Amazon Alexa.

Dan yang terbaru saat ini, penerapan Smart City yang sudah dilakukan di beberapa negara maju, seperti China dan Jerman. Sehingga, segala

bentuk aktivitas penduduk suatu kota dapat termonitoring dengan baik oleh sistem dengan jaringan basis data berskala besar.

2.5.1 Bagian-bagian IoT

Setelah mengenal apa itu internet of things, selanjutnya masuk pada pembahasan mengenai unsur – unsur IoT. Setidaknya, terdapat lima unsur pembentuk dari internet termasuk juga kecerdasan buatan, konektivitas, sensor, dan lain sebagainya. Berikut merupakan penjabarannya:

- **Artificial Intelligence (AI)**

Artificial Intelligence (AI) atau dalam bahasa Indonesia berarti kecerdasan buatan merupakan merupakan sebuah penemuan yang dapat memberikan kemampuan bagi setiap teknologi atau mesin untuk berpikir (menjadi “smart”). Jadi, AI disini dilakukan dengan mengumpulkan berbagai data, pemasangan jaringan, dan pengembangan algoritma dari kecerdasan buatan.

Sehingga, dari yang awalnya sebuah mesin hanya dapat melaksanakan perintah dari pengguna secara langsung, sekarang dapat melakukan berbagai aktivitas sendiri tanpa menunggu instruksi dari pengguna. Misalnya saja, teknologi AI yang diterapkan pada robot pelayan di sebuah restoran di Jepang.

Dimana, kemampuan robot tersebut dapat berpikir layaknya seorang pelayan manusia asli. Karena di dalam sistem kendali robot tersebut telah menggunakan bantuan AI. Dengan mencakup berbagai sumber data dan informasi secara lengkap dan algoritma yang kompleks.

- **Konektivitas**

Konektivitas atau biasa disebut dengan hubungan koneksi antar jaringan. Di dalam sebuah sistem IoT yang terdiri dari perangkat kecil, setiap sistem akan saling terhubung dengan jaringan. Sehingga dapat menciptakan kinerja yang lebih efektif dan efisien.

Untuk standar biaya pemasangan jaringan tidak selalu membutuhkan jaringan yang besar dan biaya yang mahal. Anda juga dapat merancang sistem perangkat dengan menggunakan jaringan yang lebih sederhana dengan biaya yang lebih murah.

- **Perangkat ukuran kecil**

Di dalam perkembangan teknologi masa kini, semakin kecil sebuah perangkat maka akan menghasilkan biaya yang lebih sedikit,

namun efektifitas dan skalabilitas menjadi tinggi. Sehingga di masa yang akan datang, manusia dapat lebih mudah menggunakan perangkat teknologi berbasis IoT dengan nyaman, tepat, dan efisien.

- **Sensor**
Sensor merupakan unsur yang menjadi pembeda dari IoT dengan mesin canggih yang lain. Dengan adanya sensor, mampu untuk mendefinisikan sebuah instrumen, yang mana dapat mengubah IoT dari jaringan standar yang cenderung pasif menjadi sistem aktif yang terintegrasi dengan dunia nyata.
- **Keterlibatan Aktif**
Banyak mesin modern yang masih menggunakan keterlibatan (*engagement*) secara pasif. Namun, yang menjadi pembeda dari mesin yang lain, IoT telah menerapkan metode paradigma aktif dalam berbagai konten, produk, serta layanan yang tersedia.

2.5.2 Cara Kerja

Cara kerja internet of things adalah memanfaatkan sebuah argumentasi dari algoritma bahasa pemrograman yang telah tersusun. Dimana, setiap argumen yang terbentuk akan menghasilkan sebuah interaksi yang akan membantu perangkat keras atau mesin dalam melakukan fungsi atau kerja.

Sehingga, mesin tersebut tidak memerlukan bantuan dari manusia lagi dan dapat dikendalikan secara otomatis. Faktor terpenting dari jalannya program tersebut terletak pada jaringan internet yang menjadi penghubung antar sistem dan perangkat keras. Tugas utama dari manusia adalah menjadi pengawas untuk memonitoring setiap tindakan dan perilaku dari mesin saat bekerja.

Kendala terbesar dari pengembangan Internet of things adalah dari sisi sumber daya yang cukup mahal, serta penyusunan jaringan yang sangat kompleks. Biaya pengembangan juga masih terlampau mahal dan tidak semua kota atau negara telah menggunakan IoT sebagai kebutuhan primer mereka.

2.5.3 Penerapan

Banyak sekali contoh dari penerapan IoT dalam kehidupan sehari – hari yang tanpa anda sadari sangat dekat dengan anda. Berikut merupakan beberapa contoh bidang yang telah menerapkan teknologi IoT.

- Kesehatan

Contoh internet of things yang pertama dalam bidang kesehatan. Saat ini, banyak sekali teknologi *advanced* yang dapat membantu kinerja dari dokter maupun tenaga medis. IoT juga membuat sebuah terobosan baru dalam pengembangan mesin dan alat medis untuk mendukung kinerja dari tenaga medis agar lebih efektif, tepat, dan mengurangi resiko kesalahan.

Salah satu contoh dari keberadaan IoT dalam dunia kesehatan adalah membantu dalam proses pendataan detak jantung, mengukur kadar gula tubuh, mengecek suhu tubuh dan lain sebagainya. Data yang diperoleh akan disimpan dalam penyimpanan data berskala besar.

Saat ini lebih dikenal dengan big data. Dengan menggunakan big data mampu membaca informasi dan data yang berupa angka atau teks secara cepat, dan efisien. Tenaga medis tidak perlu lagi untuk mencatat secara manual, karena semua informasi dapat ditampung dalam basis data dan akan dikirimkan pada mesin IoT untuk menjalankan tugas sesuai dengan algoritma yang dikembangkan.

- Energi

Dalam bidang energi, terdapat bervariasi permasalahan yang timbul. Mulai dari polusi atau pencemaran, pemborosan, dan berkurangnya pasokan sumber daya. Oleh karena itu, dengan adanya IoT sendiri mampu untuk mengurangi beberapa resiko tersebut. Misalnya saja, dengan penerapan sensor cahaya mampu untuk mengurangi penggunaan energi listrik.

Dengan sensor tersebut, mampu menangkap partikel cahaya, sehingga saat cahaya tersebut banyak maka lampu akan mati. Namun, saat tidak ada pasokan cahaya, maka lampu akan otomatis menyala.

Kemudian, juga dapat menerapkan pada fungsi penjadwalan yang dilakukan pada mesin oven, mesin pemanas yang telah terintegrasi dengan jaringan internet. Dan contoh konkret yang sering kita jumpai adalah pada smart TV yang telah menerapkan IoT untuk

metode pencarian channel disesuaikan dengan pilihan pengguna (*user*).

- **Transportasi**

Teknologi cerdas juga telah mencapai bidang transportasi umum. Biasanya, anda selalu mengendarai sebuah mobil sendiri sesuai dengan aturan dan kemampuan berkendara yang telah anda pelajari. Namun, apakah anda sudah mengetahui saat ini ada penemuan terbaru, dimana anda dapat menjalankan mobil tanpa mengemudi sendiri.

Mobil tersebut dapat berjalan sendiri sesuai dengan prosedur dan terprogram dengan baik. Jadi, anda dapat merasakan sensasi seperti pada sistem autopilot di pesawat. Tahap pengembangan kendaraan tersebut masih diujicobakan di beberapa negara maju.

Selain kendaraan, sistem lalu lintas juga termasuk dalam cakupan internet of things. Dengan IoT, mampu untuk mengontrol berbagai sistem lalu lintas saat kondisi macet maupun sepi. Sehingga, mampu mengurangi resiko angka kecelakaan dan pelanggaran lalu lintas yang terjadi.

- **Lingkungan Umum**

Contoh internet of things yang terakhir yaitu dalam bidang lingkungan umum. Dimana segala aktivitas manusia, tumbuhan, maupun hewan dapat dipantau dan diawasi dengan menggunakan teknologi IoT. Misalnya saja, untuk melakukan penelitian kualitas air harus dibutuhkan sumber informasi yang akurat dan terpercaya.

Dengan bantuan internet of things, mampu untuk mencari sumber data secara valid dan cepat. Tidak hanya itu, cakupan wilayah geografis yang disajikan juga cukup luas dan dapat menjangkau lebih banyak daerah. Dengan bantuan big data, permasalahan mengenai kecepatan transfer data dan pembacaan data data tertutupi dengan baik.

2.5.4 Manfaat

Setelah mengetahui dengan rinci mengenai contoh internet of things, berikutnya masuk pada pembahasan mengenai manfaat internet of things. Manfaat disini dapat dibagi menjadi tiga bagian.

- Memudahkan proses konektivitas
 Manfaat IoT yang pertama adalah memudahkan dalam proses konektivitas antar perangkat atau mesin. Semakin koneksi antar jaringan baik, maka sistem perangkat dapat berjalan dengan lebih cepat dan fleksibel.
 Anda mungkin masih banyak yang menggunakan alat konvensional, namun apabila anda mencoba untuk mengoperasikan sebuah sistem secara terpusat hanya melalui perangkat mobile, maka jawabannya yang pasti adalah dengan menggunakan teknologi cerdas.
- Ketercapaian efisiensi
 Manfaat internet of things yang kedua adalah tercapainya efisiensi kerja. Semakin banyak konektivitas jaringan yang terbentuk, semakin kecil pula jumlah penurunan waktu untuk melakukan tugas. Sehingga, aktivitas dan kinerja manusia menjadi lebih terbantu dengan adanya IoT.
- Meningkatkan efektivitas monitoring kegiatan
 Dengan menggunakan internet of things, efektivitas untuk mengontrol dan monitoring sebuah pekerjaan menjadi lebih mudah. Selain itu, teknologi cerdas juga mampu untuk memberikan rekomendasi atau alternatif pekerjaan yang lebih mudah bagi pengguna.

2.6 ThingSpeak

Thingspeak.com merupakan platform Internet of Things dibagian cloud dimana kita dapat mengirim atau menerima suatu data dengan protokol komunikasi HTTP dan juga dapat menampilkan nilai data melalui dashboard gratis yang diberikan[11].



Gambar 2.6 Tampilan depan website thingspeak

2.6.1 Fungsi

Thingspeak.com berfungsi sebagai pengumpul data yang berasal dari perangkat node berupa sensor-sensor yang sudah terhubung ke internet dan juga memungkinkan pengambilan data dari perangkat lunak untuk keperluan visualisasi ,notifikasi, kontrol dan analisis historis data.

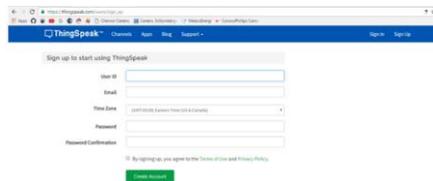
Unsur utama dari ThingSpeak adalah saluran, yang berisi bidang data, bidang lokasi, dan bidang status. Setelah membuat saluranThingSpeak, kita bisa menulis data ke saluran proses dan melihat hasil data lewat MATLAB. Dari situ dapat dilihat reaksi terhadap data dengan *tweet* dan alert lainnya.

2.6.2 Fitur

- Open API
- Real-time data collection
- Geolocation data
- Data processing
- Data Visualizations
- Device status messages
- Plugins

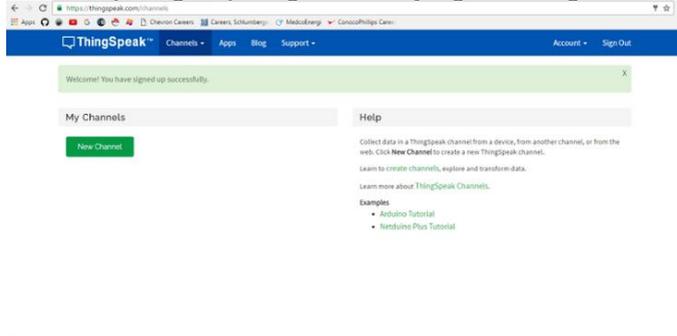
2.6.3 Langkah Penggunaan

- Kita harus memiliki akun Thingspeak.com dengan mengisikan User ID, Email dan juga Password. Pastikan email yang digunakan aktif karena digunakan untuk konfirmasi akun dan pemberitahuan lebih lanjut.



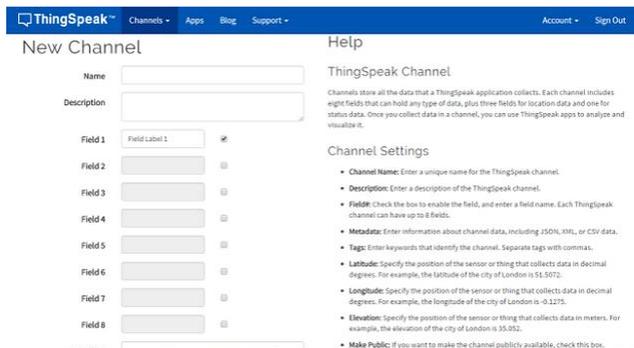
Gambar 2.7 Tampilan sign up thingSpeak

- Membuat Channel baru dengan cara menekan tombol "New Channel" yang nantinya ini digunakan untuk menampilkan data yang dikirim dari perangkat-perangkat sensor yang terhubung ke internet.



Gambar 2.8 Tampilan saat akan membuat channel baru

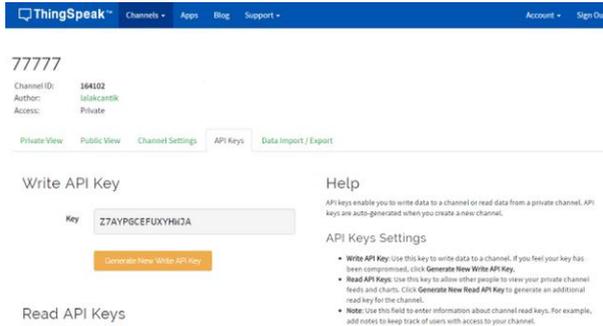
- Langkah ketiga adalah mengisi rincian untuk saluran yang baru kita buat. Nama saluran bebas dalam pelabelannya dan ini sekaligus menjadi nama data tersimpan atau menjadi proyek kerja kita . Kita boleh mengkosongkan beberapa field jika memang tidak digunakan dalam penyimpanan datanya.



Gambar 2.9 Pengaturan channel baru

- Untuk langkah terakhir kita bisa menggunakan API Key yaitu API Write yang berguna untuk keperluan mengirim data ke

thingspeak.com dan juga API Read yang digunakan untuk mengambil data dari thingspeak.com untuk keperluan analisis data maupun kontrol.



Gambar 2.10 Setting API Key

2.7 Tachometer atau RPM meter

Tachometer adalah alat untuk mengukur kecepatan putaran dari piringan, atau rotasi poros motor dan mesin berputar. Ukurannya yang kecil dan mudah dibawa membuat tachometer sangat membantu *engineer* ketika melakukan inspeksi peralatan.

Fungsi pengukuran rotasi pada sebuah unit untuk mengetahui efisiensi kinerja dari unit tersebut. Biasanya dilakukan pada saat inspeksi. Karena, efisiensi dari kinerja unit berputar tentu akan mempengaruhi biaya produksi nantinya. Oleh sebab itu, inspeksi unit kerja berpotar dilakukan secara berkala di industri dengan menggunakan tachometer.



Gambar 2.11 Photoelectric untuk tachometer

Tachometer juga diaplikasikan pada kendaraan bermotor untuk menunjukkan kecepatan putaran poros motor. Dengan demikian, para pengendara dapat mengetahui berapa kecepatan kendaraan yang sedang mereka operasikan. Selain itu, tachometer digunakan untuk memeriksa/mengukur kecepatan aliran darah dalam dunia kesehatan

2.7.1 Jenis tachometer

- Tachometer AC atau DC

Untuk mengukur putaran generator DC, tachometer listrik akan dihubungkan dengan poros putaran generator. Pengukuran putaran dilakukan pada medan magnet konstan dari magnet permanen. Rotasi tersebut menginduksi gaya gerak listrik. Voltmeter kumparan bergerak mengukur besarnya gaya gerak listrik yang diinduksi.

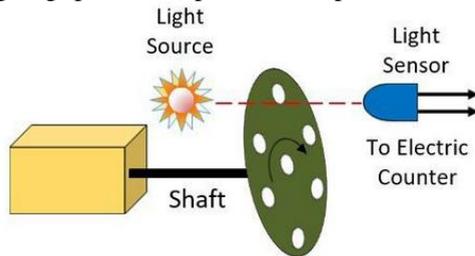
Resistansi diberikan secara seri untuk mengontrol arus besar yang dihasilkan oleh *armature*, dan komutator digunakan untuk mengubah arus bolak-balik menjadi arus searah. Maka dari serangkaian konversi gaya magnet dan gaya gerak listrik tersebut menghasilkan jumlah putaran dari generator.

Pada pengukuran rotasi generator AC, angker tachometer tetap diam di dalam medan magnet berputar. Rotasi medan magnet menginduksi gaya gerak listrik dalam kumparan, dan amplitudo atau frekuensi gaya gerak listrik dipakai untuk

mengukur kecepatan rotasi. Pada awalnya tegangan diperkuat dan kemudian diperbaiki dengan melewati filter tegangan.

- Tachometer kontak dan non kontak

Tachometer jenis kontak akan bekerja dengan menempel secara langsung pada perangkat yang berputar. Sebuah sensor magnetik dipasang untuk menampilkan *output* dalam hal membaca rpm motor. Jenis tachometer ini biasanya terpasang langsung (permanen) pada unit berputar.



Gambar 2.12 Ilustrasi cara kerja tachometer photoelectric

Tachometer non-kontak dirancang untuk tidak bersentuhan dengan unit berputar yang akan diukur. Pada jenis ini terdiri dari mekanisme sinar laser atau radiasi inframerah yang diproyeksikan langsung oleh tachometer. Hasil pengukuran akan tampil secara langsung pada layar LCD yang terpasang pada tachometer. Perangkat tachometer jenis ini dapat mengukur rentang kecepatan dengan sudut apapun.

2.8 Accelerometer MPU 6050

Module MPU6050 merupakan sensor yang berfungsi sebagai accelerometer sekaligus gyroscope yang dipackage kedalam 1 module. Accelerometer sendiri Merupakan sensor berfungsi untuk mengukur percepatan suatu benda atau objek yang bergerak yang bersifat dynamic maupun static.



Gambar 2.13 Accelerometer

Pengukuran dynamic merupakan pengukuran percepatan pada objek benda yang bergerak, sedangkan pengukuran static merupakan pengukuran percepatan objek terhadap gravitasi bumi. Untuk hal ini accelerometer lebih sering digunakan untuk mengukur sudut kemiringan (tilt) dan secara dasarnya prinsip kerja dari accelerometer yaitu menggunakan prinsip percepatan (acceleration).

Contoh pada accelerometer di kehidupan sehari – hari yaitu fungsi shake control Handphone. Dimana dengan fungsi ini kita dapat mengikuti kuis – kuis yang misalnya diadakan oleh online shop seperti Shopee, dan tokopedia.

Selain itu fungsi ini dapat diaplikasikan pada music player dimana untuk mengganti lagu ke lagu selanjutnya hanya dengan menggoyakan Handphone kita. Contoh lainnya yaitu mengubah tampilan layar dari semula vertical (portrait) menjadi horizontal (landscape) untuk melihat video youtube misalnya.

2.8.1 Spesifikasi MPU6050

- Menggunakan chip IC MPU6050
- Tegangan operasional di range antara 3Vdc – 5Vdc
- Menggunakan antarmuka komunikasi I2C (SCL, SDA)
- Range dari Gyroscope : 250 500 1000 2000 / s
- Chip built-in 16bit AD converter, 16-bit data output
- Difungsikan sebagai sensor accelerometer dan gyroscope
- Ukuran modul : 2.2cm x 1.7cm

HALAMAN INI SENGAJA DIKOSONGKAN

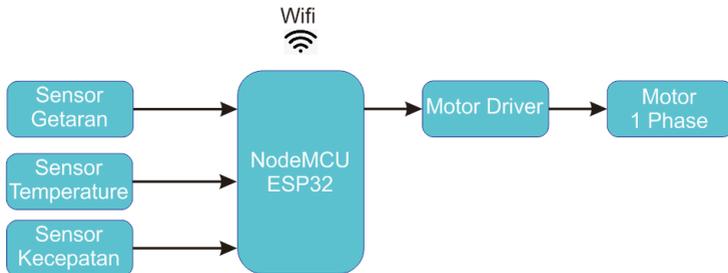
BAB III

PERANCANGAN SISTEM

3.1 Perencanaan Sistem

Perangkat keras secara garis besar terdiri dari NodeMCU ESP32 sebagai mikrokontrol, sensor suhu DS1820, MPU6050 sebagai sensor getaran, dan photo elektrik sebagai tachometer pengukur RPM motor. Sedangkan untuk *Graphical User Interface* pada website, digunakan tampilan bawaan dari *broker iot* yaitu thingspeak.

Perencanaan sistem dalam mengerjakan proyek akhir ini dikerjakan berdasarkan blok diagram yang telah dibuat seperti Gambar 3.1 dibawah ini :



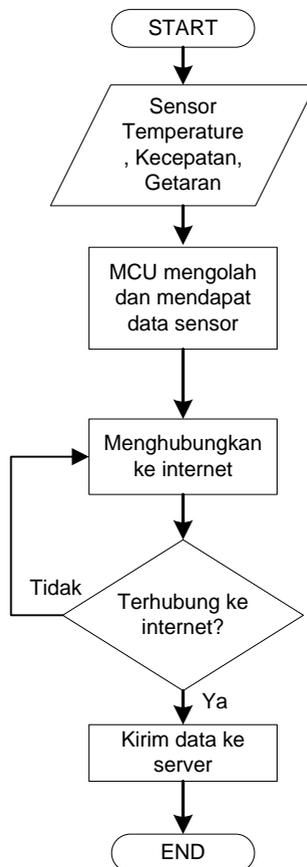
Gambar 3.1 Blok diagram sistem

Berdasarkan pada blok diagram di atas secara keseluruhan gambaran sistem yang dibangun pada perancangan dan pembuatan alat ini terdiri dari beberapa bagian yaitu: *input*, pemroses data, dan *output*.

- a. Blok input ini terdiri dari perangkat yang mengirimkan datanya ke mikrokontroler sebagai referensi eksekusi program. Sensor getaran menggunakan akselerometer MPU6050, sensor temperature menggunakan DS1820 waterproof yang ditempelkan pada bodi motor, dan photoelectric yang difungsikan sebagai tachometer untuk mengukur RPM
- b. Blok kedua adalah blok mikrokontroler atau dapat kita sebut *nodeMCU*, yang berfungsi untuk menjalankan atau mengeksekusi

perintah, mengolah data dari *input* yang berupa sensor dan berbagai aktuator yang digunakan pada sistem ini.

- c. Pada bagian output alat ini menggunakan driver motor yang bertugas untuk mengaktifkan motor. Selain itu output dari alat ini adalah data yang dikirimkan melalui jaringan internet yang kemudian akan disajikan dalam bentuk grafik di website.



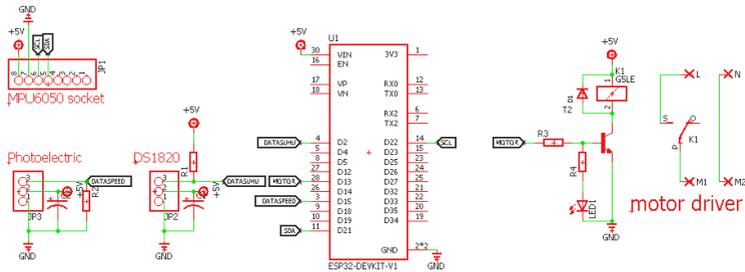
Gambar 3.2 Flowchart Sistem monitoring

Motor induksi dinyalakan melalui relay yang dikendalikan oleh mikrokontroller. Motor dinyalakan secara terus menerus dengan kecepatan konstan yang kemudian dimonitoring oleh sensor getaran,

temperature dan rpm.. data dari sensor-sensor tersebut kemudian dikemas dalam bentuk paket yang selanjutnya data tersebut dikirimkan melalui jaringan internet.

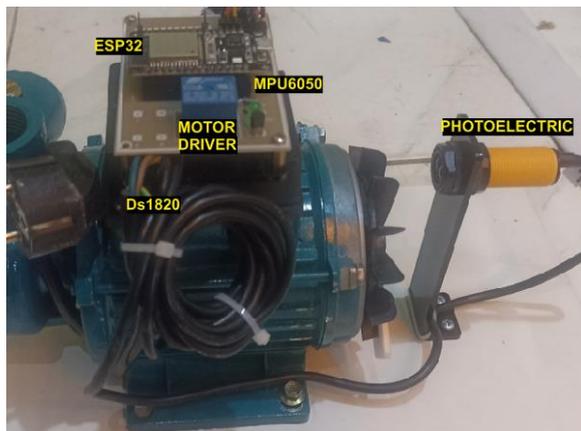
3.2 Perencanaan dan Perancangan Perangkat Keras Elektronik

Perangkat keras elektronik direncanakan terdiri dari minimum sistem nodeMCU, sensor DS1820, MPU6050 dan photoelectric.

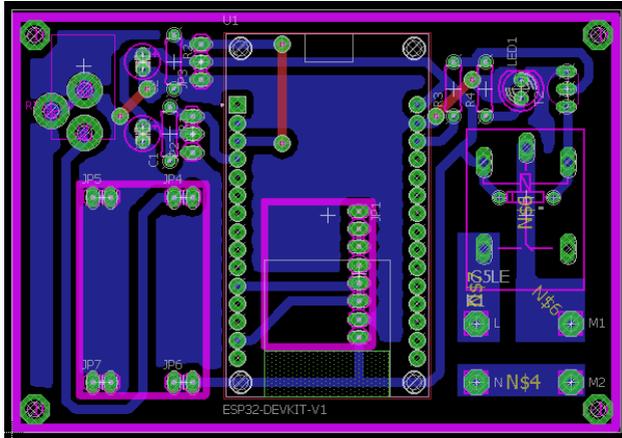


Gambar 3.3 Rancangan skematik

Sensor MPU6050 yang digunakan adalah modul kompak, sehingga yang dibutuhkan dalam perancangan skematik dan PCB hanyalah socket saja, berbeda dengan yang lainnya yang langsung menggunakan komponen secara individu.



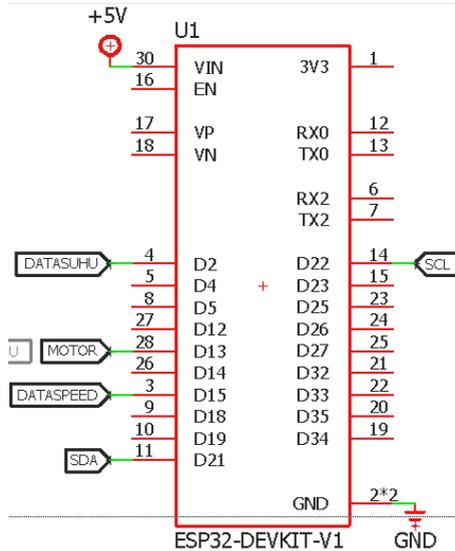
Gambar 3.4 Hasil jadi motor induksi dengan perangkat monitoring



Gambar 3.5 Desain layout PCB rangkaian monitoring

3.2.1 NodeMCU

Sebagai pengolah data, sistem minimum sangat diperlukan, dalam sistem minimum terdapat IC mikrokontroler dan komponen pendukung lainnya. Dalam alat ini mikrokontroler yang digunakan ialah ESP32. Konfigurasi port input/output dari mikrokontroler adalah sebagai berikut:

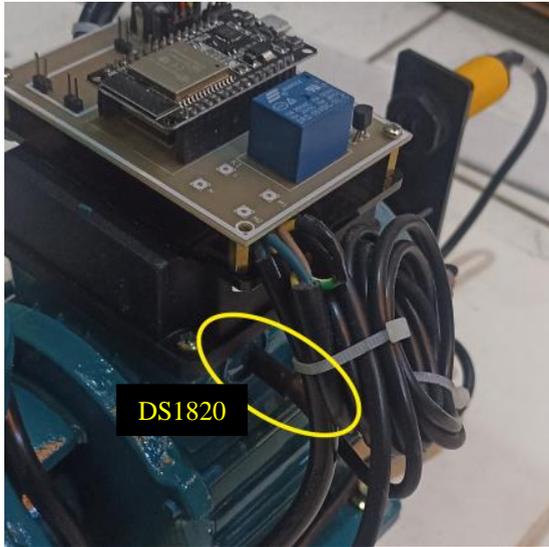


Gambar 3.6 Konfigurasi pin I/O ESP32 ke interface

Dari gambar 3.6 data temperature dari DS1820 dihubungkan pada pin D2, trigger unutmk mengaktifkan motor pada pin D13, pulsa dari photoelectric pada pin D15 dan MPU6050 karena menggunakan jalur data I2C maka membutuhkan 2 pin SDA dan SCL pada pin D21 dan D22.

3.2.2 Sensor DS1820

Sensor DS1820 menggunakan jalur data 1 wire yang hanya memerlukan 1 pin untuk akses sensor tersebut. Pada jalur data tersebut dipasang sebuah resistor yang dihubungkan pada tegangan positif untuk menjaga sinyal high agar tetatp stabil.



Gambar 3.7 Aplikasi sensor temperature DS1820 pada motor

Modul sensor tersebut dihubungkan dengan sumber tegangan 5volt. Daftar komponen yang digunakan untuk sebagai berikut

- DS1820
- Resistor 10K

3.2.3 Photoelectric

Pembacaan RPM dari motor memanfaatkan kipas dari motor induksi. Putaran dari kipas tersebut menghasilkan pulsa 1 dan 0. Dari pulsa tersebut akan dihitung berapa kali pulsa 1 diterima oleh mikrokontrol dalam waktu 1 menit motor berputar. Kemudian data banyaknya rotasi dalam satu menit tersebut di simpan pada suatu paket data bersama data sensor yang lain



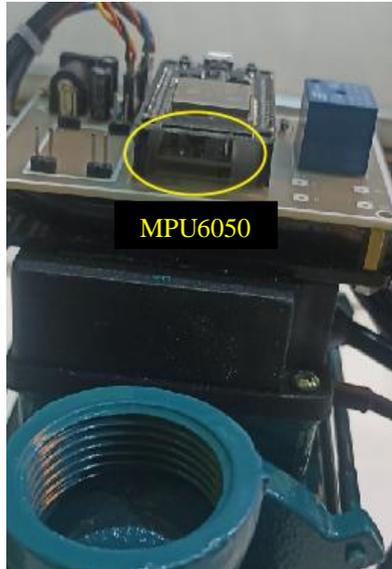
Gambar 3.8 Aplikasi photoelectric untuk pengukuran kecepatan

Modul sensor tersebut dihubungkan dengan sumber tegangan 5volt. Daftar komponen yang digunakan untuk sebagai berikut

- Photoelectric
- Kapasitor filter 10uf/16volt

3.2.4 MPU6050

Pengukuran tegangan memanfaatkan kemampuan akselerometer dalam mengukur percepatan. Nilai percepatan tersebut kemudian diturunkan menjadi suatu nilai getaran. Peletakan sensor ini sangat berpengaruh pada hasil pembacaan dari sensor. Pada alat yang dibut ini, sensor diletakkan pas di bagian tengah badan dari motor induksi.



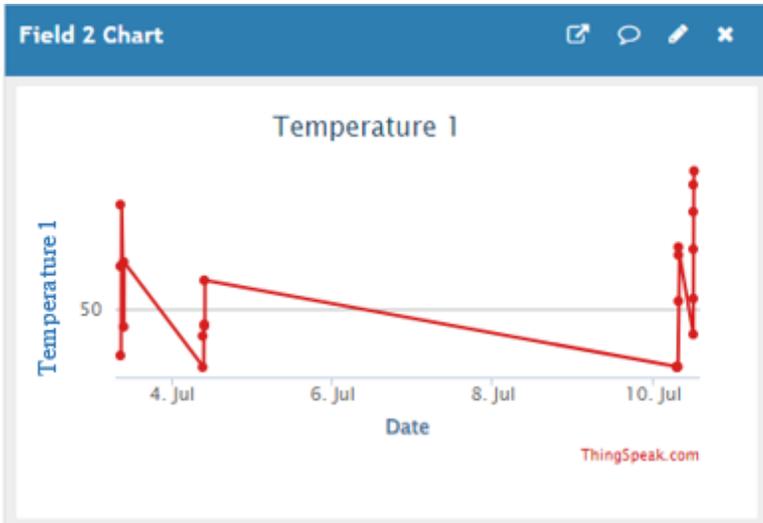
Gambar 3.9 Aplikasi Sensor getaran MPU6050 pada motor

Modul sensor tersebut dihubungkan dengan sumber tegangan 5volt. Daftar komponen yang digunakan untuk sebagai berikut

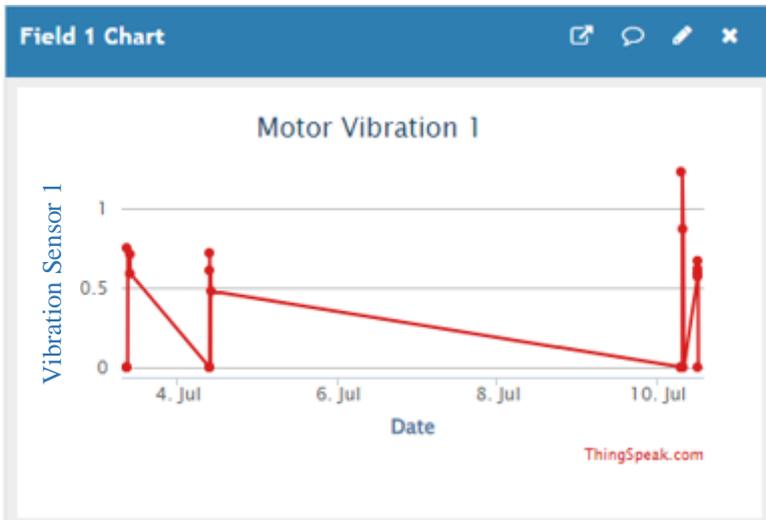
- Sensor modul MPU6050
- Kapasitor filter 10uf/16volt

3.3 ThingSpeak

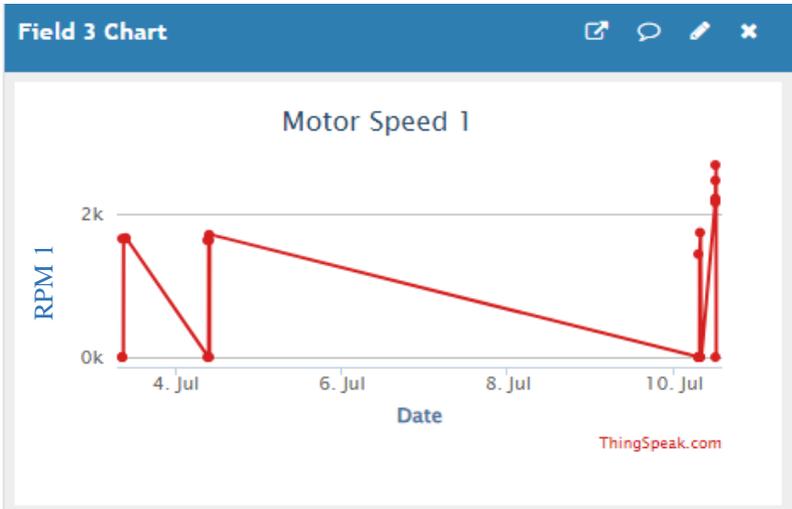
Dalam penelitian ini untuk memonitoring nilai tegangan dan arus digunakan sebuah interface berupa sebuah website yang nantinya akan menyimpan dan menampilkan data yang dikirim oleh perangkat keras. Selain dari website, juga dapat diakses melalui apk pada smartphone, pada penelitian ini digunakan aplikasi pihak ke 3 yaitu Thingshow yang mengambil data dari thingspeak dengan cara memasukkan channel ID dari field yang buat di thingspeak.



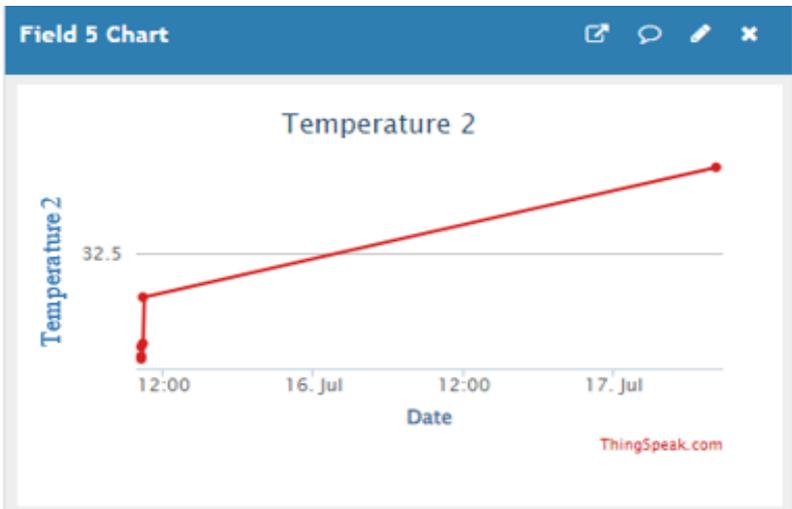
Gambar 3.10 Tampilan data temperature motor 1 pada thingspeak



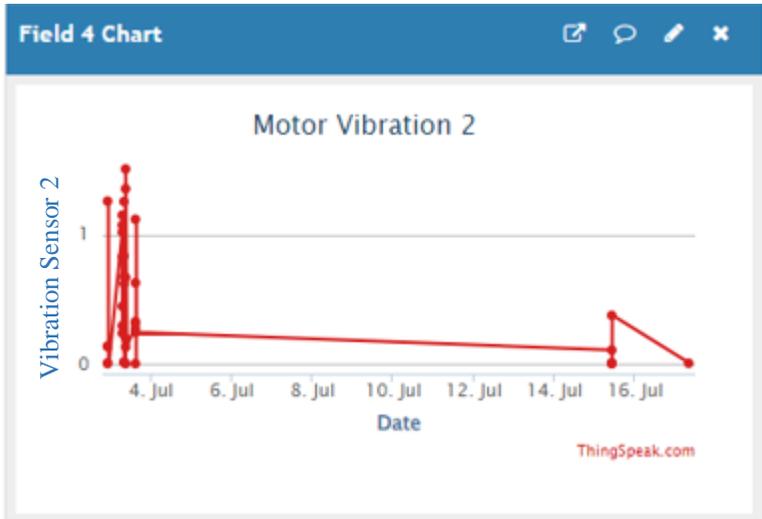
Gambar 3.11 Tampilan data getaran motor 1 pada thingspeak



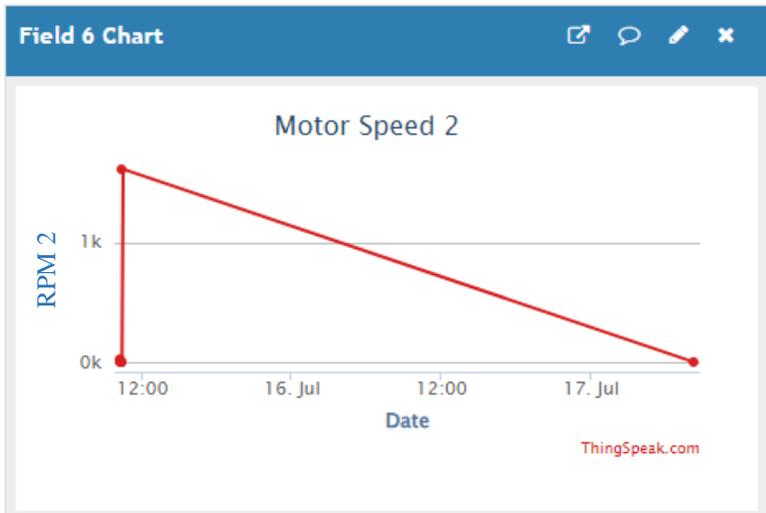
Gambar 3.12 Tampilan data kecepatan motor 1 pada thingspeak



Gambar 3.13 Tampilan data temperature motor 2 pada thingspeak



Gambar 3.14 Tampilan data getaran motor 2 pada thingspeak



Gambar 3.15 Tampilan data kecepatan motor 2 pada thingspeak

HALAMAN INI SENGAJA DIKOSONGKAN

BAB IV

ANALISA DAN PENGUJIAN

Setelah perencanaan dan pembuatan alat dilakukan maka dilaksanakan pengujian untuk mengetahui sistem yang sudah dirancang berjalan sesuai atau tidak dengan teori maupun hasil perhitungan. Pengujian tersebut dilakukan dalam beberapa tahap. Pengujian pertama dilakukan secara terpisah, kemudian pengujian secara keseluruhan atau sistem yang sudah terintegrasi. Pengujian yang dilakukan secara terpisah adalah :

1. Pengujian Sensor temperature
2. Pengujian Sensor kecepatan motor
3. Pengujian pembacaan getaran motor
4. Hasil monitoring Thingspeak

4.1 Pengujian Sensor Temperature

4.1.1 Tujuan

- Untuk mengetahui karakteristik sensor DS1820.
- Untuk memastikan data dari sensor dapat terbaca oleh mikrokontroler

4.1.2 Proses Pengambilan Data

Pengambilan data temperature dilakukan dengan menyalakan motor secara kontinyu dan data temperature diambil melalui serial monitor. Untuk pembandingan nilai pembacaan dari sensor digunakan sebuah termokouple bawaan dari multimeter untuk mengetahui % error dari pembacaan temperature.

4.1.3 Hasil dan Analisa

DS1820 merupakan sensor yang diakses melalui sistem digital dengan sistem 1 wire. Dengan selisih pembacaan/error hingga 0,5°C. Besarnya error pembacaan menurut *datasheet* dari produsen cukup baik. Dalam sistem yang dibangun ini spesifikasi dari sensor sudah sangat memenuhi karena nilai error pembacaan suhu tidak berpengaruh spesifik terhadap perubahan suhu hingga ketelitian nol koma.

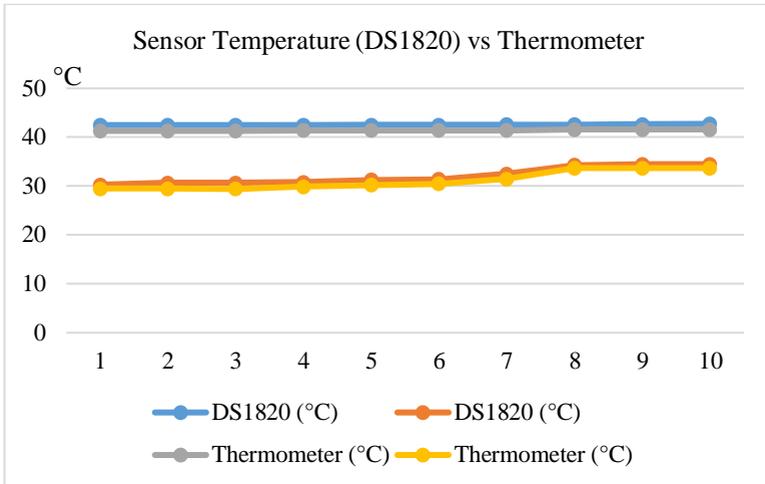
Data diambil setiap 1 menit sekali dengan rentang waktu hingga 10 menit. Dari hasil yang ada terlihat bahwa hasil pembacaan nilai suhu dari sensor dapat dikatakan bagus karena selisihnya dengan pembacaan thermometer hanya berkisar 1,2°C hingga 1,9°C dengan nilai variasi error 2,45% hingga 2,95%. Dari selisih pembacaan atau besarnya error yang didapat, menunjukkan bahwa sensor DS1820 dalam sistem monitoring sangat cukup kemampuannya apabila digunakan sebagai referensi proses perencanaan maintenance motor.

Tabel 4.1 Perbandingan sensor DS1820 dengan Thermometer

Data ke	DS1820 (°C)	DS1820 (°C)	Thermometer (°C)	Thermometer (°C)	%Error
1	42.5	30.3	41.3	29.5	2.91
2	42.5	30.7	41.3	29.5	2.91
3	42.5	30.7	41.3	29.4	2.91
4	42.5	30.8	41.4	29.9	2.66
5	42.56	31.3	41.4	30.2	2.80
6	42.56	31.4	41.4	30.5	2.80
7	42.62	32.5	41.4	31.5	2.95
8	42.62	34.3	41.6	33.7	2.45
9	42.69	34.5	41.6	33.7	2.62
10	42.75	34.5	41.6	33.7	2.76
					27.76
			Rata % error		2.78

$$\%Error = \frac{Ds1820(c)-Thermometer}{Thermometer} \times 100\%$$

$$\begin{aligned}
 &= \frac{42,5-41,3}{41,3} \\
 &= \frac{1,2}{41,3} = 0,02 \\
 &= 0,02 \times 100\% \\
 &= 0,02
 \end{aligned}$$



Gambar 4.1 Grafik terbandingan pembacaan thermometer dengan sensor temperature (DS1820)

4.2 Pengujian Sensor kecepatan motor

4.2.1 Tujuan

- Untuk mengetahui hasil pembacaan sensor kecepatan.
- Untuk memastikan data dari sensor dapat terbaca oleh mikrokontroler.

4.2.2 Proses Pengambilan Data

Data diambil dengan mengaktifkan motor dan memantaunya melalui serial terminal dan dibandingkan secara langsung dengan tachometer. Data diambil selama 10 menit dengan 10 pengambilan data.

4.2.3 Hasil dan Analisa

Sistem pembacaan sensor kecepatan yang digunakan pada alat ini ialah system pembacaan kecepatan putar motor menggunakan optikal sensor dengan memanfaatkan putaran kipas motor yang kemudian menghasilkan sinyal digital yang kemudian dihitung secara software dengan memanfaatkan berapa banyak sensor tersebut menghasilkan pulsa

positif pada dalam 1 menit yang kemudian menghasilkan nilai rotasi per menit. Kecepatan nominal yang ertera pada spesifikasi motor yang digunakan ialah 2850 Rpm dan 1400 Rpm. Dimana hasil yang didapatkan sangat bervariasi.

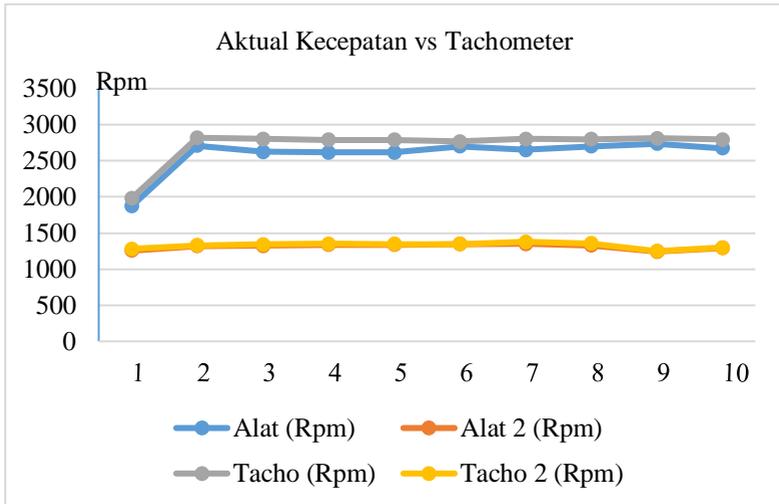
Tabel 4.2 Perbandingan aktual kecepatan vs Tachometer

Data ke	Alat (Rpm)	Alat 2 (Rpm)	Tacho (Rpm)	Tacho 2 (Rpm)	%Error
1	1872	1260	1980	1282	5.77
2	2712	1345	2820	1360	3.98
3	2628	1345	2802	1360	6.62
4	2616	1345	2790	1360	6.65
5	2616	1345	2790	1360	6.65
6	2700	1322	2770	1330	2.59
7	2652	1322	2804	1330	5.73
8	2700	1231	2800	1249	3.70
9	2736	1231	2815	1249	2.89
10	2676	1231	2794	1249	4.41
Total	25908	12977	27165	13129	49.00
Rata-rata	2590.8	1297.7	2716.5	1312.9	4.90

$$\begin{aligned}
 \%Error &= \frac{\text{Alat} - \text{Tacho}}{\text{Tacho}} \times 100\% \\
 &= \frac{1872 - 1980}{1980} \\
 &= \frac{-108}{1980} = 0,05 \\
 &= 0,05 \times 100\% = 0,05
 \end{aligned}$$

Data dari tabel tersebut menunjukkan adanya perbedaan yang cukup besar antara actual dengan alat ukur, besarnya presentase error rata-rata hampir mencapai 5% atau terjadi perbedaan sekitar kurang lebih diangka 100-200. Hal ini menunjukkan bahwa metode pengukuran kecepatan yang dimodifikasi membutuhkan sensor optic yang lebih presisi. Untuk rata rata pembacaan anara actual dengan alat ukur terjadi perbedaan sekitar 126 Rpm. Jadi untuk melakukan monitoring terhadap kecepatan, terlebih dahulu harus menentukan nilai-nilai RPM yang

seharusnya dan nilai RPM minimum dimana motor tersebut membutuhkan suatu perawatan. Maka dengan adanya error pembacaan aktual ini, pengguna dapat melakukan setting sesuai nilai aktual yang ada.



Gambar 4.2 Grafik perbandingan aktual kecepatan vs tachometer

4.3 Pengujian pembacaan getaran motor

4.3.1 Tujuan

- Untuk mengetahui hasil pendeteksian vibrasi pada motor

4.3.2 Proses Pengambilan Data

Data diambil dengan mencatat secara realtime nilai kecepatan dan getaran pada motor saat motor menyala. Data diambil melalui erial terminal.

4.3.3 Hasil dan Analisa

Hasil pembacaan getaran dihubungkan dengan kecepatan motor, karena diasumsikan makin cepat motor maka getaran yang timbul juga semakin besar. Dan nilai getaran digolongkan menurut zona yang sesuai dengan ISO 10816. Data diambil dengan motor tanpa beban. Dengan nilai

getaran di bawah 0,72 maka motor masih digolongkan Good sedangkan diatas 0,72 motor digolongkan pada zona satisfactory. Dalam 20 pengambilan data getaran. Nilai rata-rata getaran yang di dapat dari pembacaan sensor ialah 0,63 mm/s dengan variasi nilai getaran dari yang terkecil 0,45 mm/s hingga terbesar 1,50. Sedangkan hubungan antara kecepatan dan getaran cenderung variatif, hal ini menunjukkan bahwa getaran suatu motor dapat dihubungkan dengan beban yang membebani rotor dari motor.

Tabel 4.3 klasifikasi data perbandingan getaran dan kecepatan pada motor 1 dan 2

Data ke	RPM	RPM (2)	Getaran mm/s	Getaran (2) mm/s	Kategori Zona
1	2532	1260	0.49	0.24	Good
2	2676	1345	0.75	1.01	Satisfactory
3	2580	1345	0.61	0.56	Good
4	2760	1345	0.45	0.68	Good
5	2664	1345	0.48	0.68	Good
6	2568	1322	0.59	0.54	Good
7	2568	1322	0.6	0.55	Good
8	2340	1231	0.61	0.58	Good
9	2568	1231	0.6	0.60	Good
10	2544	1231	0.62	0.63	Good
11	2436	1231	0.46	0.67	Good
12	2580	1231	0.63	0.67	Good
13	2652	1231	0.68	0.62	Good
14	2520	1231	0.8	1.14	Satisfactory
15	2664	1231	0.83	1.25	Satisfactory
16	2532	1128	0.71	0.13	Good
17	2448	1128	0.62	0.21	Good
18	2400	1187	0.67	0.32	Good
19	2328	1187	0.72	1.35	Satisfactory
20	2436	1187	0.74	1.50	Satisfactory

4.4 Hasil monitoring thingspeak

4.4.1 Tujuan

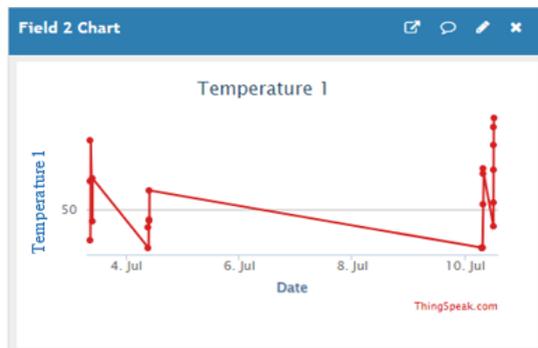
- Untuk mengetahui hasil pengiriman data secara online pada website thingspeak

4.4.2 Proses pengambilan data

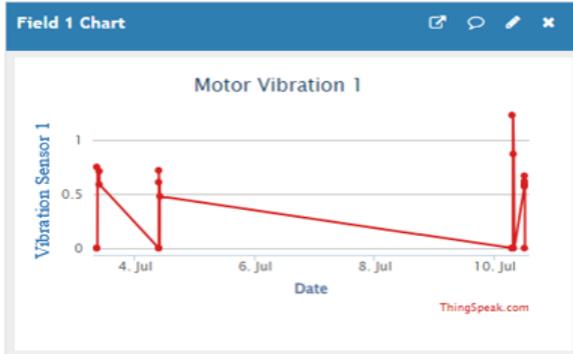
Data diambil dengan mengaktifkan motor beberapa kali. Dan melakukan pemantauan secara online

4.4.3 Hasil dan analisa

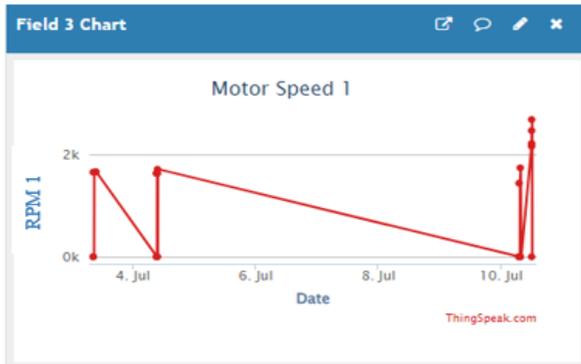
Data dari esp32 dikirimkan melalui jaringan internet setiap 1 menit sekali, dimana dari beberapa proses pengiriman data, data tersebut terkirim dan diterima dengan utuh oleh server. data grafis dari monitoring motor ini berguna untuk kebutuhan pemantauan secara realtime, sedangkan untuk hasil logging data yang dapat diekspor berupa .csv berformat tabel akan sangat berguna untuk evaluasi kondisi motor sebagai referensi penentuan atau pembuatan jadwal perawatan rutin pada suatu motor.



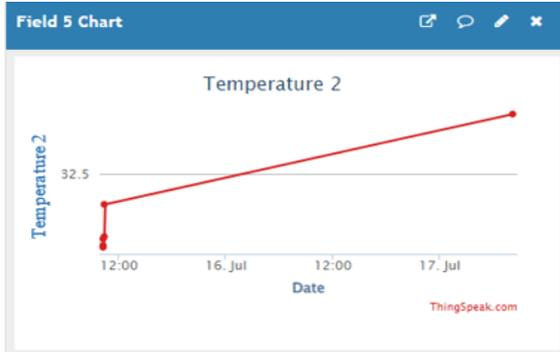
Gambar 4.3 Tampilan hasil grafis temperature motor 1 pada Thingspeak



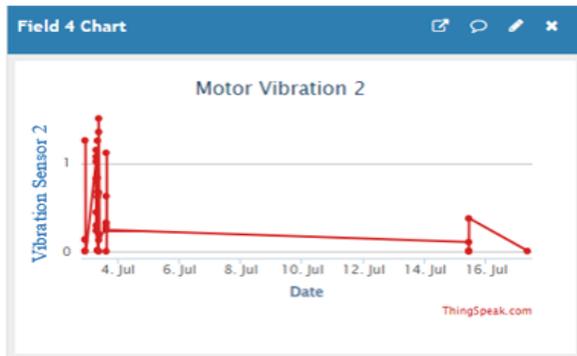
Gambar 4.4 Tampilan hasil grafis getaran motor 1 pada thingspeak



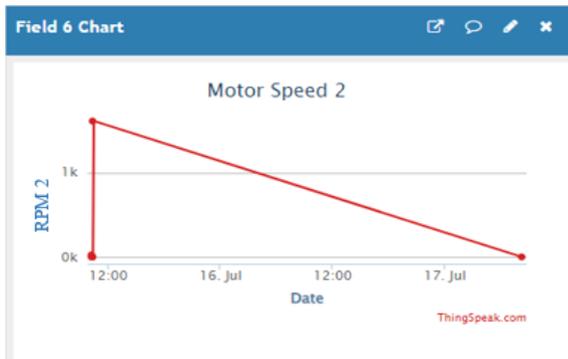
Gambar 4.5 Tampilan hasil grafis kecepatan motor 1 pada thingspeak



Gambar 4.6 Tampilan hasil grafis temperature motor 2 pada thingspeak



Gambar 4.7 Tampilan hasil grafis getaran motor 2 pada thingspeak



Gambar 4.8 Tampilan hasil grafis kecepatan motor 2 pada thingspeak

HALAMAN INI SENGAJA DIKOSONGKAN

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan percobaan dan penelitian terhadap alat yang dibuat maka menghasilkan kesimpulan sebagai berikut:

1. Aplikasi IoT pada sistem monitoring motor menggunakan jaringan internet untuk melakukan pengiriman data ke server dengan melalui nomor API.
2. Sensor temperature DS1820 maupun thermocouple adalah sensor yang sangat cocok digunakan untuk pengukuran permukaan suatu benda, dalam kasus ini ialah body dari motor.
3. Penentuan wajar atau tidaknya getaran motor didasarkan pada standart ISO 10816.
4. Proses pembacaan kecepatan memanfaatkan pulsa yang diterima oleh Esp32 sebagai trigger interrupt untuk memulai perhitungan kecepatan

5.2 Saran

Saran untuk perbaikan di penelitian selanjutnya untuk menggunakan lisensi berbayar agar update data dapat cepat secara real time dari sensor ke ThinkSpeak dan dapat langsung di prediksi menggunakan algoritma machine learning.

HALAMAN INI SENGAJA DIKOSONGKAN

DAFTAR PUSTAKA

- [1] sinuarduino. (2016, Maret 16). *Mengenal Arduino Software (IDE)*. Retrieved Juni 24, 2022, from SINAUARDUINO:<https://www.sinuarduino.com/artikel/mengenal-arduino-software-ide/>
- [2] Permana, A., & Setiono, I. (2017). SISTEM PENGENDALIAN SUHU DAN PEMANTAUAN KELEMBABAN BIJI KOPI. *GEMA TEKNOLOGI*, 19, 2.
- [3] Musyaffa, I. F. (2019). *MONITORING TEMPERATUR DAN GETARAN MOTOR INDUKSI 3 PHASE MENGGUNAKAN ARDUINO UNO* (Doctoral dissertation, Universitas Islam Indonesia).
- [4] *Sensor Kelembaban Tanah atau Soil Moisture*. (2020, Januari 3). Retrieved Juni 24, 2020, from Algorista: <http://www.algorista.com/2020/01/sensor-soil-moisture.html>
- [5] Suyoko, N., Afroni, M. J., & Basuki, B. M. (2019). SISTEM PENGAMAN DAN MONITORING MOTOR INDUKSI SATU FASA SECARA ONLINE. *SCIENCE ELECTRO*, 11(1)./
- [6] Sibarani, A. N. (2020). Studi Analisis Perubahan Putaran Motor Induksi Satu Fasa Akibat Output Plts Aplikasi Kipas Angin.
- [7] Rifky, I. (2021, November 16). *MIKROKONTROLER ESP32*. Retrieved Juni 2022, 24, from <https://raharja.ac.id/2021/11/16/mikrokontroler-esp32-2/>
- [8] Nasution, H. (2021, Oktober). Implementasi Logika Fuzzy pada Sistem Kecerdasan Buatan. *Jurnal ELKHA*, 4, 4. Retrieved Juni 24, 2022.
- [9] Priyandoko, G. (2021). Klasifikasi Kerusakan Motor Induksi Menggunakan Metode Transformasi Wavelet Diskrit Dan K-Nearest Neighbor. *JOINTECS (Journal of Information Technology and Computer Science)*, 6(2), 109-116

- [10] Winston (Ed.). (2021, Desember 26). *PENGERTIAN POMPA DIAFRAGMA, PENGGUNAAN DAN PRINSIP KERJANYA*. Retrieved Juni 24, 2022, from WINSTON ENGINEERING:https://www.winstonengineering.com/id/id/events/39_pompa-diafragma.html
- [11] Maulana, K. Y. (2021, Desember 9). *Thingspeak.com, Web Server IoT Gratis Buat Kamu Yang Suka Gratisan*. Retrieved Juni 24, 2024, from ANAK TEKNIK INDONESIA:<https://www.anakteknik.co.id/krysnayudhamaulana/articles/thingspeakcom-web-server-iot-gratis-buat-kamu-yang-suka-gratisan/>
- [12] MAULANA, R. F. (2022). *SIMULASI SISTEMONITORING GETARAN, SUHU DAN ARUS PADA MOTOR INDUKSI* (Doctoral dissertation).

LAMPIRAN

```
#include <WiFi.h>
#include <Wire.h>
#include "ThingSpeak.h"
#include "MPU6050_tockn.h"
#include "OneWire.h"
#include "DallasTemperature.h"

#include "soc/sens_reg.h" // needed for manipulating ADC2
control register

#define Sens_Speed_Pin 18
#define Relay_Pin 19
#define DSTemp_Pin 2

MPU6050 mpu6050(Wire); // SDA 21, SCL 22

OneWire oneWire(DSTemp_Pin); // 2
DallasTemperature sensors(&oneWire);

#define Detik 1000
#define TOKirim 60
#define SelisihAcY 0.2

char ssid[] = "Pohon Belimbing";
char password[] = "klentang25";

int keyIndex = 0;

WiFiClient client;
```

```

unsigned long myChannelNumber = 2052098;
const char * myWriteAPIKey = "HQ3NCEJFZ1FPXMJ";
float Vo, getaran = 0.0;
unsigned long prevmillis = 0, prevmillisV = 0;
bool TimeToSend, Kirim = false;
uint8_t detik = 0;
float temperature;
int cacah;
float zeroAcX, zeroAcY, zeroAcZ, AcX, AcY, AcZ, Selisih;
int timee = 0, RPM = 0;
float gettCal;
void setup() {
    Serial.begin(115200);
    Wire.begin();
    mpu6050.begin();
    sensors.begin();
    pinMode(Sens_Speed_Pin, INPUT);
    pinMode(Relay_Pin, OUTPUT);
    delay(100);
    mpu6050.calcGyroOffsets(true);
    zeroAcX = mpu6050.getAccX();
    zeroAcY = mpu6050.getAccY();
    zeroAcZ = mpu6050.getAccZ();

```

```

// Serial.printf("\r\nZero acX: %0.2f |acY: %0.2f |acZ:
%0.2f\r\n", zeroAcX, zeroAcY, zeroAcZ);

WiFi.mode(WIFI_STA);

uint8_t x = 0;

do {

    if(WifiStatus()) {

        break;

    }

    else {

        x++;

        delay(1000);

    }

}

while(x < 5);

ThingSpeak.begin(client);

delay(1000);

digitalWrite(Relay_Pin, HIGH);

delay(3000);

Vo = 0;

mpu6050.update();

float xx = mpu6050.getAccX();

float y = mpu6050.getAccY();

float z = mpu6050.getAccZ();

```

```

    gettCal = sqrt(xx*xx+y*y+z*z);
    Serial.printf("\r\n getCal: %0.2f\r\n", gettCal);
}
#define interval 10
void loop() {
    timee = 0;
    RPM = 0;
    timee = delay1();
    RPM = (timee*12)/1;
    mpu6050.update();
    AcX = mpu6050.getGyroX();// mpu6050.getAccX();
    AcY = mpu6050.getGyroY();// mpu6050.getAccY();
    AcZ = mpu6050.getGyroZ();// mpu6050.getAccZ();
    getaran = (abs(AcX) + abs(AcY) + abs(AcZ)) / 100.0;
    float x = mpu6050.getAccX();
    float y = mpu6050.getAccY();
    float z = mpu6050.getAccZ();
    float gett = abs(sqrt(x*x+y*y+z*z) - gettCal);
// counter();
    if(millis() - prevmillisV >= interval) {
        prevmillisV = millis();
// getaran = Vo + (AcZ *interval);
        Vo = getaran;

```

```

}

if(millis() - prevmillis >= Detik) {
    prevmillis = millis();
    detik++;

//   getaran = (mpu6050.getGyroX()/131);
    sensors.requestTemperatures();
    temperature = sensors.getTempCByIndex(0);

    Serial.printf("RPM:%d|T:%0.2f|G:%0.2f|g:%0.2f|", RPM,
temperature, getaran, gett);

//   Serial.printf("acX:%0.2f|acY:%0.2f|acZ:%0.2f|S:%0.2f|",
AcX, AcY, AcZ, Selisih);

//   Serial.printf("gyroX:%0.2f|gyroY:%0.2f|gyroZ:%0.2f|",
mpu6050.getGyroX(), mpu6050.getGyroY(),
mpu6050.getGyroZ());

//   Serial.printf("accAngleX:%0.2f|accAngleY:%0.2f|",
mpu6050.getAccAngleX(), mpu6050.getAccAngleY());

//
Serial.printf("gyroAngleX:%0.2f|gyroAngleY:%0.2f|gyroAngleZ:
%0.2f|", mpu6050.getGyroAngleX(),
mpu6050.getGyroAngleY(), mpu6050.getGyroAngleZ());

//
Serial.printf("getAngleX:%0.2f|getAngleYX:%0.2f|getAngleZ:%
0.2f|", mpu6050.getAngleX(), mpu6050.getAngleY(),
mpu6050.getAngleZ());

    Serial.println();

    if(detik > TOKirim) { detik = 0; TimeToSend = true; }

    if(TimeToSend) {

```

```

    if(WifiStatus()) {
        TimeToSend = false;
        Kirim = true;
    }
    else { TimeToSend = false; }
}
}
if(Kirim) {
    SendToThinspeak();
    Kirim = false;
}
}
void CekGetar() {
//30 derajat
// if((AcY > 15000)&&(AcY < 20000)){
//  digitalWrite(led1,LOW); //MERAH
//  digitalWrite(led2,HIGH); //BIRU
//  digitalWrite(led3,LOW);
//  digitalWrite(led4,LOW);
// }
// if((AcY > SelisihGyro)&&(AcY < SelisihGyro)) { counter(); }
//60 derajat
// if((AcY > 10000)&&(AcY < 13000)) {

```

```

// digitalWrite(led1,LOW); //MERAH
// digitalWrite(led2,LOW); //BIRU
// digitalWrite(led3,LOW);
// digitalWrite(led4,HIGH);
// }
// if(AcY < 10000){ counter(); }
//   if(Selisih > SelisihAcY) { counter(); }
// }
void counter() {
  AcX = mpu6050.getAccX();
  AcY = mpu6050.getAccY();
  AcZ = mpu6050.getAccZ();
  Selisih = AcY - zeroAcY;
  if(Selisih < 0.0) Selisih *= -1;
  // Serial.print("Selisih = "); Serial.println(Selisih);
  // Serial.print(" | AcY = "); Serial.print(AcY);
  // Serial.print(" | AcZ = "); Serial.print(AcZ);
  // Serial.print(" | getaran = "); Serial.println(getaran);
  // if(AcY < 10000) {
  //   if(Selisih > SelisihAcY) {
  //     delay(200);
  //     getaran++;
  //   }
}

```

```

// if((AcY > 13000)&&(AcY < 15000)) {
////  delay(200);
//  getaran++;
//  }
  if(getaran > 3) {
//    digitalWrite(led1,HIGH); //MERAH
//    digitalWrite(led2,LOW); //BIRU
//    digitalWrite(led3,HIGH);
//    digitalWrite(led4,LOW);
  }
  if(cacah > 50) {
    cacah = 0;
//    getaran = 0;
    return;
  }
  cacah++;
//  delay(100);
  counter();
}
int delay1() {
  int i,j;
  unsigned int count=0;
  for(i=0;i<1000;i++) {

```

```

for(j=0;j<1000;j++) {
  if(digitalRead(Sens_Speed_Pin)) {
    count++;
    while(digitalRead(Sens_Speed_Pin));
  }
}
}
return count;
}

void SendToThinspeak() {
  ThingSpeak.setField(1, getaran);
  ThingSpeak.setField(2, temperature);
  ThingSpeak.setField(3, RPM);

  int x = ThingSpeak.writeFields(myChannelNumber,
myWriteAPIKey);
  // if(x == 200) { Serial.println("Channel update successful."); }
  // else { Serial.println("Problem updating channel. HTTP error
code " + String(x)); }
}

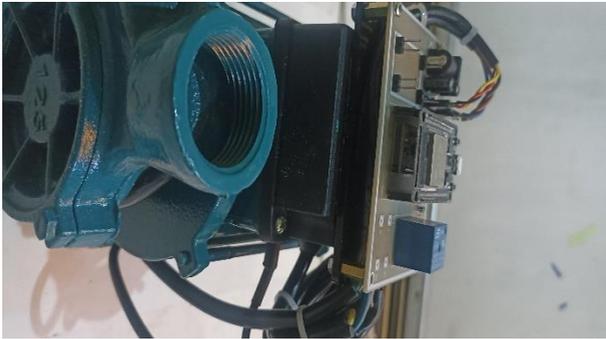
bool WifiStatus() {
/* // Jika harus dipastikan sampai terkoneksi dulu
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");

```

```

Serial.println(SECRET_SSID);
while(WiFi.status() != WL_CONNECTED){
    WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network.
Change this line if using open or WEP network
    Serial.print("."); //delay(1000);
}
Serial.println("\nConnected.");
}
*/
/* // Jika hanya cek saja sekali, lalu kembali ke loop
*/
WiFi.begin(ssid, password);
if(WiFi.status() != WL_CONNECTED) {
// Serial.println("Connecting to WiFi...");
return false;
}
else {
// Serial.println("Connected to WiFi");
return true;
}
}
}
...

```



MOTOR 1



MOTOR 2

SURAT BIMBINGAN TUGAS AKHIR



YAYASAN BRATA BHAKTI DAERAH JAWA TIMUR
UNIVERSITAS BHAYANGKARA SURABAYA
FAKULTAS TEKNIK

Kampus : Jl. A. Yani 114 Surabaya Telp. 031 - 8285602, 8291055, Fax. 031 - 8285601

SURAT TUGAS

Nomor : TUG / 20 / FTK/10/2022

- Pertimbangan :
- Bahwa dalam rangka kelancaran pelaksanaan bimbingan tugas akhir mahasiswa Program Studi Teknik Elektro Fakultas Teknik Universitas Bhayangkara Surabaya.
 - Bahwa sehubungan dengan hal tersebut diatas maka dipandang perlu untuk mengeluarkan Surat Tugas ini.
- Dasar :
- Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Nomor 17 Tahun 2013 tentang Jabatan Fungsional dan Angka Kreditnya, sebagaimana telah diubah dengan Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Nomor 46 Tahun 2013.
 - Buku Pedoman Akademik Universitas Bhayangkara Surabaya Tahun Akademik 2022 / 2023.
 - Kurikulum Program Studi Teknik Elektro Fakultas Teknik Universitas Bhayangkara Surabaya.

DITUGASKAN

- Kepada : Dr., Ir. Saidah, MT. (Pembimbing tunggal)
- Untuk :
- Membimbing tugas akhir mahasiswa Program Studi Teknik Elektro Fakultas Teknik Universitas Bhayangkara Surabaya atas nama ACHMAD SHOLIKHUL FAJRI (1814111018) berjudul *MONITORING TEMPERATURE, GETARAN, DAN KECEPATAN MOTOR INDUKSI 1 PHASE BERBASIS WEBSITE*.
 - Melaksanakan tugas ini dengan seksama dan penuh rasa tanggung jawab.
 - Tugas ini berlaku sejak tanggal ditetapkan sampai selesainya kegiatan.
- Selesai.

Dikeluarkan Di : Surabaya

Pada Tanggal : 31 Oktober 2022



Dekan

Dr. Muhammad Ghazi, S.T., M.T.

HALAMAN INI SENGAJA DIKOSONGKAN

PROFIL PENULIS



- | | |
|--------------------------|--|
| 1. NIM | : 1814111018 |
| 2. NAMA | : Achmad Sholikhul Fajri |
| 3. Tempat, Tanggal Lahir | : Ngawi, 3 Mei 2000 |
| 4. Program Studi | : Teknik Elektro (Sistem Tenaga) |
| 5. Alamat Rumah | : Jl. Dukuh Menanggal 8 No 3
Surabaya |
| 6. Pekerjaan | : |
| 7. Alamat Kantor | : |
| 8. Nomor Ponsel | : 08973361234 |
| 9. Email | : solikul02@gmail.com |
| Judul Tugas Akhir | : MONITORING TEMPERATURE,
GETARAN, DAN KECEPATAN
MOTOR INDUKSI 1 PHASE
BERBASIS WEBSITE |
| Pembimbing | : Prof. Dr.Ir. Saidah, MT. |
| Tanggal Daftar | : 27 Juni 2023 |
| Tanggal Ujian | : 10 Juli 2023 |

Surabaya, 10 Juli 2023
Penulis,

(Achmad Sholikhul Fajri)

HALAMAN INI SENGAJA DIKOSONGKAN

FORM BIMBINGAN TUGAS AKHIR



YAYASAN BRATA BHAKTI DAERAH JAWA TIMUR
UNIVERSITAS BHAYANGKARA SURABAYA
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
 Kampus: Jl. Ahmad Yani 114 Surabaya Telp. 031 - 8285602, Fax. 031 - 8285601

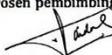
FORM BIMBINGAN TUGAS AKHIR

Nama : ACHMAD SHOLIKHUL FAJRI NIM : 1814111018
 Judul Tugas Akhir : MONITORING TEMPERATURE, GETARAN, DAN KECEPATAN MOTOR
 INDUKSI 1 PHASE BERBASIS WEBSITE
 Pembimbing 1 : Prof. Dr., Ir. Saidah, MT. NIDN : 0712066101
 Pembimbing 2 : NIDN :

NO	TANGGAL	CATATAN REVISI	ACC
1.	03-05-2023	BAB I Konsultasi dan revisian Penulisan	✓
2.	07-05-2023	BAB II Revisi tata letak	✓
3.	12-05-2023	BAB III Revisi blok diagram dan Flowchart	✓
4.	18-05-2023	ACC BAB I, II, III	✓
5.	26-05-2023	BAB IV Revisi Tabel dan Perumusan	✓
6.	13-06-2023	BAB IV Revisi Keterangan Gambar & Hasil monitoring	✓
7.	19-06-2023	BAB V Revisi Tambah daftar Pustaka	✓
8.	22-06-2023	ACC BAB IV, V	✓
9.	27-06-2023	Pengujian Alat	✓

Surabaya, 17 Oktober 2022

Disetujui pada tanggal :
22-06-2023
 untuk mengikuti ujian: **Proposal / Tugas Akhir***
 Dosen pembimbing 1,


 Prof. Dr., Ir. Saidah, MT.

Disetujui pada tanggal :

 untuk mengikuti ujian: **Proposal /
 Tugas Akhir***
 Dosen pembimbing 2,