SISTEM MONITORING RPM RODA *SMART WHEELCHAIR* PADA HALAMAN *WEB BERBASIS AJAX* MENGGUNAKAN SENSOR *OPTOCOUPLER*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Afdy Clinton
NIM: 135150300111026

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

SISTEM MONITORING *RPM* RODA *SMART WHEELCHAIR* PADA HALAMAN *WEB* BERBASIS *AJAX* MENGGUNAKAN SENSOR *OPTOCOUPLER*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Afdy Clinton
NIM: 135150300111026

Skripsi ini telah diuji dan dinyatakan lulus pada
16 Januari 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Dahnial Syauqy, S.T., M.T., M.Sc.	Dr. Eng. Fitri Utaminingrum, S.T., M.T.
NIK. 201607 870423 1 002	NIP. 19820710 200812 2 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 16 Januari 2018

Afdy Clinton

NIM: 135150300111026

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan segala rahmatNya sehingga penulis dapat menyelesaikan makalah dengan judul "Sistem *Monitoring RPM* Roda *Smart Wheelchair* pada Halaman *Web* Berbasis *Ajax* Menggunakan Sensor *Optocoupler*"

Penulis menyadari kelemahan serta keterbatasan yang ada sehingga dalam menyelesaikan skripsi ini memperoleh bantuan dari berbagai pihak, dalam kesempatan ini penulis menyampaikan ucapan terimakasih kepada :

1. Papa Sandy, Mama Oza, Anfi Siegel dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian, dan kesabarannya memberikan semangat kepada peneliti, tiada henti memberikan doa demi terselesaikannya skripsi ini, dan telah memberikan kesempatan kepada peneliti untuk menjalani hidup dengan bidang yang peneliti inginkan.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
4. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
5. Bapak Dahnial Syauqy, S.T., M.T., M.Sc. selaku dosen pembimbing I yang telah memberikan pengarahan dan bimbingan kepada peneliti, sehingga peneliti dapat menyelesaikan skripsi ini dengan baik, meskipun peneliti jarang menghadap untuk bimbingan.
6. Ibu Dr. Eng. Fitri Utaminingrum, S.T., M.T. selaku dosen pembimbing II yang telah memberikan pengarahan dan bimbingan kepada peneliti, sehingga peneliti dapat menyelesaikan skripsi ini dengan baik, meskipun peneliti benar-benar jarang menghadap untuk bimbingan.
7. Seluruh civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya dan untuk teman-teman Teknik Komputer Angakatan 2013.
8. Bulan Austenita atas doa, perhatian, cinta, kasih sayang dan bantuannya dalam memotivasi peneliti untuk menyelesaikan skripsi ini.
9. Teman-teman "Lulus Bareng" atas waktu yang telah diluangkan dalam suka maupun duka, dari semester pertama sampai peneliti ditinggal lulus sama kalian.

Malang, 16 Januari 2018

Afdy Clinton

afdyclinton@gmail.com

ABSTRAK

Pengaplikasian motor elektrik saat ini sudah sangat luas, tetapi kurangnya perhatian terhadap kecepatan motor elektrik dapat menghasilkan *output* yang tidak diharapkan. Terdapat beberapa contoh penggunaan motor elektrik, pertama pada *smart wheelchair* dan kedua pada pembuatan robot mobil. Motor elektrik digunakan untuk menggerakan roda dari *smart wheelchair* ataupun robot mobil. *Smart wheelchair* dan robot mobil menggunakan 2 motor elektrik dengan kecepatan yang berbeda, sehingga mengakibatkan robot mobil ataupun *smart wheelchair* tidak bergerak ke arah yang dituju. Berdasarkan permasalahan tersebut, maka dibutuhkan sebuah sistem *monitoring RPM* jarak jauh dengan tujuan untuk mengetahui perbedaan kecepatan antara kedua motor ataupun kedua roda dengan mudah, sehingga kecepatan dari motor elektrik dapat dikontrol sesuai dengan kebutuhan. Dalam penelitian ini berfokus untuk mengetahui nilai *RPM* kedua roda *smart wheelchair* dengan menggunakan teknik *incremental encoder*, yaitu dengan menggunakan Sensor kecepatan FC-03 dan *encoder disk* dengan 20*holes*. Pada *smart wheelchair* diimplementasikan 2 Sensor FC-03 dan 2 *encoder disk* untuk mendeteksi nilai RPM dari masing-masing roda. Sensor FC-03 digunakan untuk mendeteksi berapa banyak *holes* yang terdeteksi dalam 1detik, sehingga dapat dihitung berapa banyak rotasi yang dilakukan oleh motor ataupun roda setiap menitnya. Berdasarkan hasil pengujian nilai *RPM* prototipe Sensor FC-03 sebelah kanan didapatkan rata-rata persentase error dari 50 data sebesar 0.92% dan sebelah kiri sebesar 0.87%. Hasil pengujian nilai *RPM* Sensor FC-03 sebelah kanan setelah diimplementasikan pada *smart wheelchair* memiliki presentase error sebesar 2.00% dan sebelah kiri sebesar 2.06%. Status koneksi *WiFi* NodeMCU saat terhubung dengan laptop pada jarak 10-50meter adalah terhubung dan status koneksi pada jarak 60-70meter adalah terputus.

Kata kunci: Kecepatan motor elektrik, *smart wheelchair,* Sensor Optocoupler, *incremental encoder*

ABSTRACT

The application of electric motors is now very wide, but the lack of attention to the speed of electric motors can produce unexpected output. There are several examples of the use of electric motors, first on the smart wheelchair and second on the manufacture of robot cars. Electric motors are used to move the wheels of a smart wheelchair or a robot car. Smart wheelchair and robot cars use 2 electric motors with different speeds, resulting a robot car or a smart wheelchair does not move in the intended direction. Based on these problems, a remote RPM monitoring system is needed in order to know the speed difference between the two motors or both wheels with ease, so that the speed of the electric motor can be controlled as needed. In this research focused on knowing the value of RPM of both wheel of smart wheelchair by using incremental encoder technique, that is by using FC-03 speed sensor and encoder disk with 20holes. On the smart wheelchair is implemented 2 FC-03 Sensor and 2 disk encoder to detect the RPM value of each wheel. The FC-03 sensor is used to detect how many holes are detected in 1sec, so it can be calculated how much rotation is done by the motor or wheel every minute. Based on the test results RPM prototype Sensor FC-03 to the right obtained the average percentage error of 50 data of 0.92% and the left of 0.87%. The result of RPM Sensor FC-03 test right after implemented on smart wheelchair has percentage error of 2.00% and left side of 2.06%. The status of NodeMCU WiFi connection when connected to a laptop at a distance of 10-50meter is connected and the connection status at a distance of 60-70meter is disconnected.

***Keywords**: Speed of electric motor, smart wheelchair, Optocoupler Sensor, incremental encoder*

DAFTAR ISI

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar belakang

Kecepatan motor elektrik masih kurang diperhatikan, meskipun pengaplikasiannya sudah sangat luas. Pengaplikasian motor elektrik pada mesin *drill* diharapkan dapat berputar dengan kecepatan tinggi, tetapi kebanyakan dari kita tidak memperdulikan seberapa cepat motor elektrik itu berputar (Hughes, 1993). Pengaplikasian lain motor elektrik yaitu dalam pembuatan robot. Robot dibuat dengan suatu fungsi tertentu. Robot diharapkan dapat bergerak cepat dan akurat. Keakuratan suatu pergerakan robot dapat dipengaruhi oleh kecepatan dari motor elektrik yang digunakan (Sigaud & Peters, 2010).

Dalam pembuatan sebuah robot mobil dengan dua buah motor yang akan digunakan pada lingkungan dengan banyak rintangan, membutuhkan *monitoring RPM* (*Revolution Per Minutes*) dari kedua motor secara cermat. *Monitoring RPM* motor digunakan untuk mengetahui apakah kecepatan kedua motor sudah sesuai, sehingga kecepatan motor yang belum sesuai dapat disesuaikan (Chen, 2011).

Sebuah *smart wheelchair* dengan fungsi dapat dijalankan secara otomatis oleh pengguna, menggunakan dua buah motor elektrik dengan kecepatan yang berbeda, menyebabkan *smart wheelchair* bergerak ke arah yang tidak dituju. Untuk mengetahui perbedaan kecepatan dari kedua motor dibutuhkan sebuah sistem *monitoring RPM* (Fitri Utaminingrum, et al., 2016).

Nilai *RPM* dapat diketahui dengan menggunakan teknik *Incremental encoder. Incremental encoder* adalah suatu teknik yang menggunakan sensor dan juga *encoder disk,* untuk mendeteksi putaran motor. *Encoder disk* adalah sebuah piringan pipih dengan beberapa lubang (*hole*) yang mengelilinginya, *encoder disk* idealnya diletakan pada *shaft* dari motor, sehingga ketika motor menyala *encoder disk* akan berputar. Sebuah sensor *optocoupler* dapat digunakan untuk mendeteksi berapa banyak lubang dari *encoder disk,* sehingga dapat ditentukan berapa nilai *RPM* dari sebuah motor. Kelebihan teknik ini yaitu menggunakan *counter* saat menambahkan nilai pembacaan sensor (W. Pessen, 1989).

Berdasarkan beberapa permasalahan di atas, maka *monitoring RPM* penting untuk dilakukan. Dalam penelitian ini dibuat sebuah sistem *monitoring RPM* jarak jauh. *Monitoring RPM* jarak jauh dilakukan agar pengguna dapat lebih leluasa dalam melakukan *monitoring,* oleh karena itu *monitoring RPM* diterapkan pada sebuah halaman web, dengan tujuan agar *monitoring* dapat dilakukan dari mana saja dan kapan saja (Perera, et al., 2012).

Pada penelitian ini *monitoring* kecepatan motor diterapkan pada *smart wheelchair* dengan tujuan untuk mengetahui kecepatan yang dihasilkan oleh setiap motor, karena motor yang digunakan memiliki perberbedaan dari segi usia pemakaian dan bahkan perbedaan spesifikasi, sehinga kedua motor tersebut tidak memiliki kecepatan yang sama dalam rentang waktu yang telah ditentukan.

Monitoring RPM setiap roda dilakukan dengan mengimplementasikan teknik *incremental encoder* dengan cara menempatkan satu *encoder disk* pada setiap gear yang terhubung dengan motor, sehingga ketika motor berputar maka *encoder disk* juga akan berputar dengan kecepatan yang sama dengan motor. Beberapa gear digunakan pada *smart wheelchair* untuk menggerakan masing-masing roda. *Sensor Optocoupler* akan dugunakan untuk mendeteksi *holes* dari *encoder disk* yang telah disesuaikan jumlah *hole*nya dengan satu kali putaran roda. Setiap data yang didapatkan oleh *Sensor Optocoupler* akan dihitung oleh NodeMCU dengan persamaan *RPM*. Setelah data dihitung, maka data dapat dikirimkan ke sebuah halaman *web*site yang telah disiapkan untuk menampilkan data *RPM* setiap roda. Pengiriman data dari perangkat keras ke halaman *web* menggunakan teknik *Ajax.* Setelah data selesai dihitung dan didistribusikan ke halaman *web*, maka *user* dapat melakukan *monitoring* secara langsung terhadap perputaran kedua roda yang digunakan.

Dengan memanfaatkan alat ini diharapkan *monitoring RPM* dari setiap roda yang digunakan pada *smart wheelchair* dapat dilakukan dari mana saja dan kapan saja, karena hanya cukup dilakukan dengan mengakses halaman *web* yang digunakan untuk menampilkan nilai *RPM.*

1.2 Rumusan masalah

Berdasarkan latar belakang di atas dapat dirumuskan masalah yang akan diangkat pada penelitian, yaitu:

1. Bagaimana rancangan dari sistem monitoring *RPM* roda *Smart Wheelchair* pada halaman *web* berbasis *Ajax* menggunakan *Sensor Optocoupler*?
2. Bagaimana kondisi nilai *RPM* bacaan prototipe dibandingkan dengan nilai *RPM* bacaan *tachometer* saat melakukan pembacaan pada sebuah roda yang digerakan oleh sebuah motor DC*?*
3. Bagaimana kondisi nilai *RPM* bacaan sensor FC-03 setelah diimplementasikan pada *smart wheelchair* dibandingkan dengan nilai *RPM* bacaan *tachometer?*
4. Berapa jarak maksimum *WiFi* dari NodeMCU ESP8266-12E saat terhubung dengan sebuah perangkat?

1.3 Tujuan

Adapun tujuan berdasarkan rumusan masalah di atas, yaitu:

1. Mengetahui rancangan yang dibutuhkan untuk membangun sebuah sistem monitoring *RPM* roda *Smart Wheelchair* pada halaman *web* berbasis *Ajax* menggunakan *Sensor Optocoupler.*
2. Mengetahui berapa selisih nilai *RPM* bacaan prototipe dibandingkan nilai *RPM* bacaan *tachometer* saat melakukan pembacaan pada sebuah roda yang digerakan oleh sebuah motor DC.
3. Mengetahui berapa selisih nilai *RPM* bacaan sensor FC-03 setelah diimplementasikan pada *smart wheelchair* dibandingkan dengan bacaan nilai *RPM tachometer.*
4. Mengetahui jarak maksimum koneksi *WiFi* dari NodeMCU ESP8266-12E ketika terhubung dengan sebuah perangkat.

1.4 Manfaat

Manfaat dari penelitian "sistem monitoring *RPM* roda *Smart Wheelchair* pada halaman *web* berbasis *Ajax* menggunakan *Sensor Optocoupler*" yaitu:

1. Mengimplementasikan *incremental encoder* pada *smart wheelchair*, sehingga dapat diketahui perbedaan nilai *RPM* antara roda kiri dan roda kanan.
2. Menjadi referensi untuk peneliti lain agar dapat mengembangkan atau membangun sebuah sistem menggunakan *incremental encoder* untuk mengetahui nilai *RPM* dari motor elektrik atau roda yang digunakan.

1.5 Batasan Masalah

Adapun batasan masalah yang digunakan pada penelitian ini agar tidak melebar dan tidak menyimpang dari apa yang akan dibahas, berikut batasan masalah yang diterapkan:

1. Penelitian menggunakan NodeMCU sebagai perangkat pengolah data dan pengirim data ke halaman *web.*
2. NodeMCU hanya digunakan sebagai *Access Point.*
3. Menggunakan NodeMCU sebagai *web server.*
4. Halaman *web* hanya dapat diakses dari IP lokal pada NodeMCU dengan *URL* 192.168.4.1.
5. Menggunakan sensor FC-03 untuk mendeteksi *holes* yang terbaca.
6. *Holes* dinyatakan sebagai sebuah *pulses* yang diterima sensor.
7. Menggunakan *Disk Encoder* dengan 20 *holes.*
8. Nilai minimum prototipe adalah 3*RPM*.

9. Nilai *RPM* yang ditampilkan protitipe hanya kelipatan 3.
10. Menggunakan satuan *RPM.*
11. Pengiriman data dari *web server* ke halaman *web* menggunakan *Ajax*.

1.6 Sistematika Pembahasan/Laporan

Sistematika pembahasan dalam skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi pembahasan mengenai latar belakang dari pembahasan topik yang dipilih, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan dari Penelitian "Sistem Monitoring *RPM* Roda *Smart Wheelchair* Pada Halaman *Web* Berbasis *Ajax* Menggunakan *Sensor Optocoupler*".

BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi landasan kepustakaan/teori yang digunakan sebagai referensi dalam pembuatan penelitian "Sistem Monitoring *RPM* Roda *Smart Wheelchair* Pada Halaman *Web* Berbasis *Ajax* Menggunakan *Sensor Optocoupler*" yang didapat dari beberapa penelitian yang berkaitan dengan perhitungan *RPM*, NodeMCU, Sensor FC-03, *Ajax* dan HTML.

BAB III METODE PENELITIAN

Bab ini membahas metode yang diterapkan dalam penelitian. Studi literatur, metode pengambilan data, metode perancangan, metode pengujian dan metode analisis biasanya termasuk dalam metode penelitian.

BAB IV REKAYASA KEBUTUHAN

Bab ini membahas kebutuhan-kebutuhan perangkat keras dan perangkat lunak dalam merancang Sistem Monitoring *RPM* Roda *Smart Wheelchair* Pada Halaman *Web* Berbasis *Ajax* Menggunakan *Sensor Optocoupler.*

BAB V PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi pembahasan mengenai perancangan Sistem Monitoring *RPM* Roda *Smart Wheelchair* Pada Halaman *Web* Berbasis *Ajax* Menggunakan *Sensor Optocoupler*.

BAB VI PENGUJIAN DAN ANALISIS

Bab ini menguraikan tentang bagaimana pengujian dilakukan, hasil pengujian, dan analisis hasil pengujian. Penyajian data dan penjelasannya dilakukan secara terurut dan logis menggunakan teks dan ilustrasi lainnya.

BAB VII PENUTUP

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan. Kesimpulan disampaikan berdasarkan hasil pengujian dan analisis yang didapat.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas dan menjelaskan mengenai teori-teori yang digunakan dalam penelitian seperti *electric wheelchair*, perhitungan *RPM,* NodeMCU, Sensor FC-03 dan *incremental encoder*.

2.1 Tinjauan Pustaka

Tinjauan pustaka berupa penelitian-penelitian yang telah ada sebelumnya dan memiliki keterkaitan dengan pembuatan penelitian ini.

Penelitian pertama adalah milik Prateek Mishra, Shikhar Pradhan, Siddartha Sethiya, Vikas Chaudhary dengan judul "*Contactless Tachometer with Auto Cut Off".* Penelitian ini membahas tentang pembuatan *contactless tachometer* untuk mengukur kecepatan motor DC. Penelitian ini menggunakan mikrokontroler Arduino UNO untuk mengolah data, *IR Sensor* digunakan untuk menerima pantulan cahaya dari *propeller*, LCD menampilkan hasil kalkulasi nilai *RPM.* Penelitian ini melakukan pengujian terhadap *propeller* yang dipasangkan pada *shaft* motor DC, sehingga akan mendapatkan *pulse* dan akan dikalkulasikan menjadi nilai *RPM*. *IR Sensor* terdiri dari *LED* dan *photodiode, LED* akan menembakan cahaya yang akan dipantulkan oleh *propeller* dan akan diterima oleh *photodiode* dan akan dianggap sebagai *pulse* dari *propeller* yang berputar. *Pulse* yang diterima akan dikirimkan ke Aruino Uno dan akan ditampilkan pada LCD berdasarkan nilai rata-rata dari 3 *pulses* secara berurutan. Setelah nilai *RPM* telah ditampilkan maka sebuah *voltage regulator* akan melakukan pembatasan terhadap daya dari motor yang berputar, sehingga penggunaan daya lebih sedikit (MISHRA, et al., 2017).

Penelitian kedua adalah milik Salice Peter, Naveen N M, Nidheesh M N, Swetha Annu James dan Seril Joseph dengan judul "*Design of a Contactless Tachometer*". Penelitian ini membahas tentang sebuah desain *tachometer* secara *contactless.* Dalam penelitian ini terdapat beberapa perangkat keras yang digunakan yaitu ATMega-16, *IR Transceiver, LED* dan *buzzer*. Penelitian ini memanfaatkan *IR transmitter* untuk menembakan *infra red* dan akan diterima oleh *IR receiver.* Diantara *IR transmitter* dan *IR receiver* terdapat sebuah *propeller*, ketika *IR receiver* tidak mendeteksi *infra red* akibat terhalang oleh *propeller,* maka akan terbaca sebagai *pulse* dan akan dikalkulasikan menjadi nilai *RPM.* Buzzer digunakan sebagai tanda peringatan ketika kecepatan motor mencapai nilai yang terlalu tinggi (Peter, et al., 2014).

Tabel 2.1 Kajian Pustaka

No.	Nama Penulis, Tahun, dan Judul	Persamaan	Perbedaan	
			Penelitian terdahulu	Rencana Penelitian
1.	Prateek Mishra, Shikhar Pradhan, Siddartha Sethiya, & Vikas Chaudhary [2017] *Contactless Tachometer with Auto Cut Off*	Menggunakan *propeller* atau *encoder disk*	Berbasis Arduino dan menggunakan *IR Sensor* untuk mendeteksi *pulse*	Berbasis NodeMCU, Web server, menggunakan *Sensor Optocoupler* untuk menteksi *pulse*
2.	Salice Peter, Naveen N M, Nidheesh M N, SwethaAnnu James dan Seril Joseph [2014] *Design of a Contactless Tachometer*	Menggunakan *propeller* atau *encoder disk*	Berbasis ATMega-16 dan menggunakan *IR Transceiver* untuk mendeteksi *pulse*	Berbasis NodeMCU, Web server, menggunakan *Sensor Optocoupler* untuk menteksi *pulse*

2.2 Dasar Teori

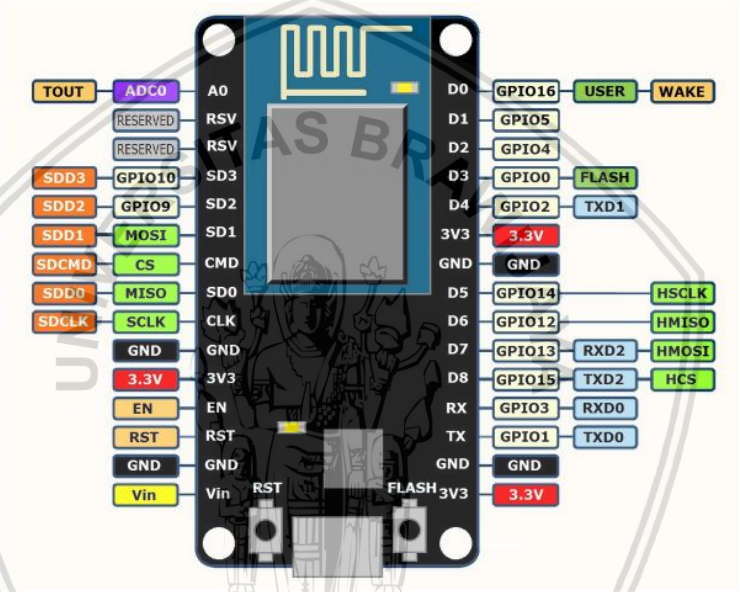
2.2.1 Electric-Powered Wheelchair

Electric-Powered Wheelchair adalah sebuah *wheelchair* yang digerakan oleh motor elektrik. *Electric-Powered Wheelchair* berguna bagi pengguna yang tidak mampu menggerakan roda secara manual. *Electric-Powered Wheelchair* tidak hanya diperuntukan untuk pengguna dengan gangguan mobilitas tetapi juga untuk pengguna yang berbasis *kardiovaskular* (Anirudh & Satpathy, 2014). Gambar 2.1 adalah contoh dari *Electric-Powered Wheelchair.*

Gambar 2.1 *Electric-Powered Wheelchair*
Sumber: (Meyra, n.d.)

2.2.2 NodeMCU ESP8266-12E

NodeMCU adalah sebuah *IoT platform* yang didalamnya terdapat modul *WiFi* ESP8266-12E dan IC AMS1117. modul *WiFi* ESP8266-12E hanya dapat menerima tegangan maksimal 3.6V sehingga pada board NodeMCU disertakan IC AMS1117 yang berfungsi sebagai *voltage regulator*, sehingga dapat menurunkan tegangan masuk sebesar 5V ke 3.3V, sehingga tidak merusak modul *WiFi* ESP8266-12E. Terdapat 4 buah Vout yang dapat digunakan pada NodeMCU, 3 buah Vout dari NodeMCU mengalirkan tegangan 3.3V dan 1 buah Vout mengalirkan tegangan 5V. Pada NodeMCU hanya terdapat 1 buah pin ADC yang ditunjukan dengan label pin A0. NodeMCU memiliki beberapa pin digital yang ditunjukan dengan label pin D0, D1, D2, D3, D4, D5, D6, D7 dan D8. Pada Gambar 2.2 diilustrasikan sebuah NodeMCU dengan menunjukan deskripsi dari setiap pin yang ada pada NodeMCU.

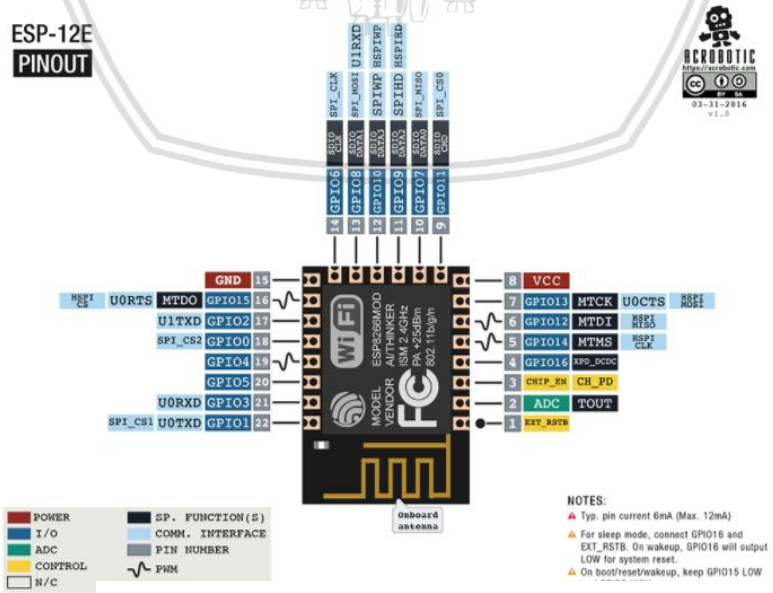


Gambar 2.2 NodeMCU
Sumber: (Currey, 2015)

ESP8266-12E atau biasa disebut ESP-12E adalah sebuah modul *WiFi* yang dikembangkan oleh Ai-thinker Team dari modul ESP8266. ESP-12E memiliki antena yang sudah terdapat pada modul, *WiFi* 2.4GHz, terdapat beberapa mode operasi yaitu STA/AP/STA+AP, sudah mendukung penggunaan *WiFi* dengan *standard* IEEE802.11 b/g/n, mendukung protokol TCP/IP dan terdapat *internal SRAM* dan *ROM*. ESP-12E dapat dimanfaatkan sebagai sebuah alat yang memancarkan sinyal *WiFi*, sehingga perangkat lain dapat terhubung dengan ESP-12E dengan memanfaatkan mode operasi yang sudah tersedia di ESP-12E. Terdapat berbagai macam fitur-fitur yang dapat dimanfaatkan dari ESP8266-12E, berikut adalah daftar fitur yang terdapat pada ESP8266-12E (Ai-Thinker, 2015).

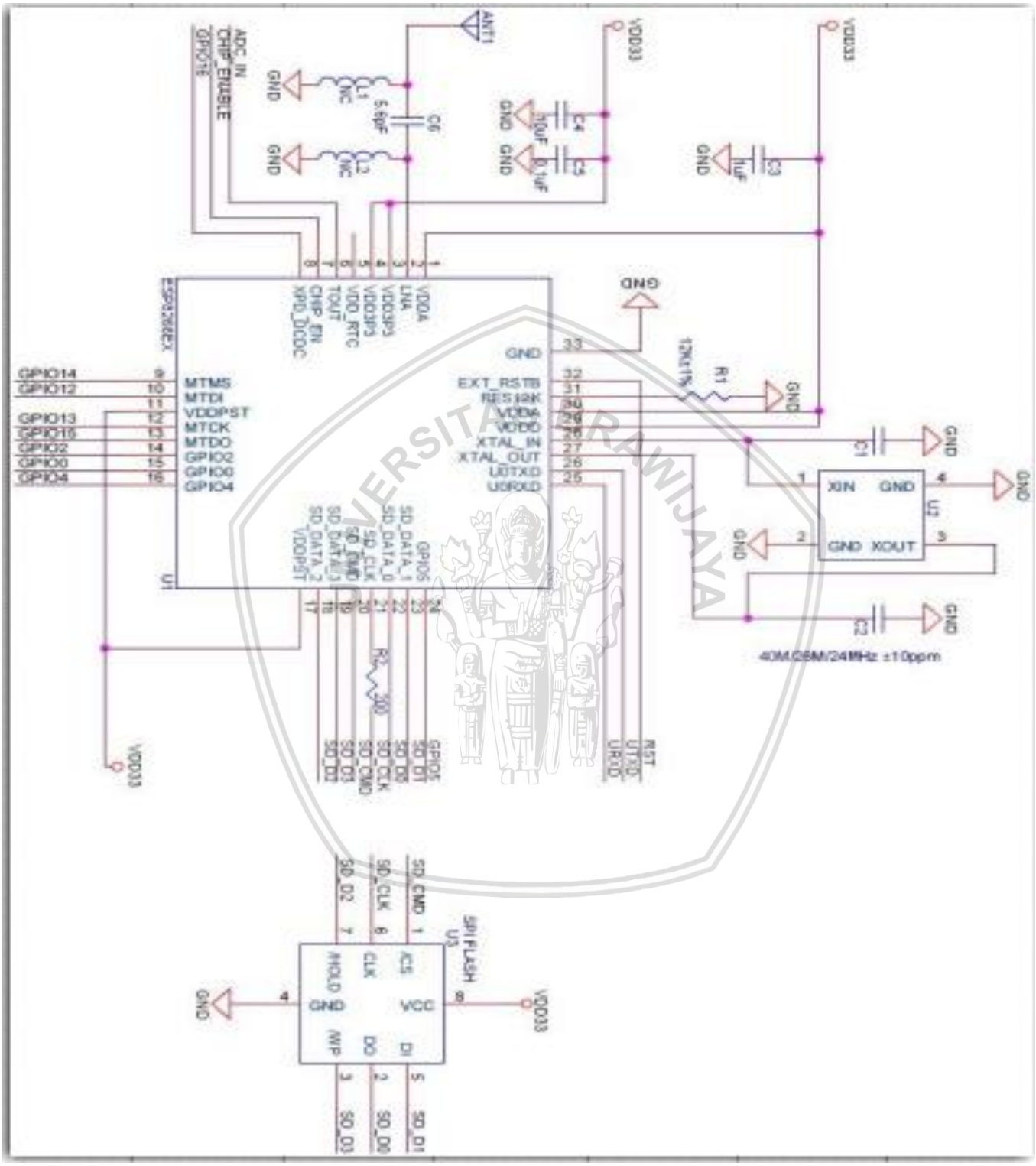
- Mendukung *WiFi* 802.11 b/g/n
- 32-bit MCU berdaya rendah
- 10-bit ADC
- TR *switch*, LNA, *amplifier* daya dan penyelaras jaringan
- PLL, dan regulator
- Mendukung *antenna diversity*
- *WiFi* 2.4Ghz, Mendukung WPA/WPA2
- Mendukung mode operasi STA/AP/STA+AP
- Mendukung *Smart Link Function* untuk perangkat Android dan iOS
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- Agregasi A-MPDU & A-MSDU dan 0.4s *guard interval*
- *Deep sleep power* <10uA, *Power down leakage current* < 5uA
- *Wake up* dan *transmit* paket dalam < 2ms
- Konsumsi daya *standby* dari < 1.0mW (DTIM3)
- +20dBm daya output pada mode 802.11b

ESP8266-12E diilustrasikan pada gambar 2.3 beserta dengan deskripsi singkat mengenai nama pin dan kegunaannya.



Gambar2.3 ESP8266-12E
Sumber: (learn.acrobotic.com, n.d.)

Rangkaian dari ESP8266-12E diilustrasikan pada Gambar 2.4 dengan mengilustrasikan jalur yang terhubung dengan *chip* inti dari ESP8266-12E.



Gambar 2.4 Skematik ESP8266-12E
Sumber: (Ai-Thinker, 2015)

Terdapat beberapa parameter yang menjelaskan tentang kemampuan dari perangkat ESP8266-12E. Tabel 2.2 menujukan parameter penting dari ESP8266-12E.

Tabel 2.2 Parameter ESP8266-12E

WiFi Parameter	*WiFi Protocol*	802.11 b/g/n
	Frequency Range	2.4GHz-2.5GHz (2400M-2483.5M)
Hardware Parameter	*Peripheral Bus*	*GPIO/PWM*
		UART/HSPI/I2C/I2S/Ir Remote Control
	Operating Temperature Range	*-40˚~125˚*
	Operating Voltage	*3.0-3.6 Volt*
	Operating Current	*Avg value:* 80mA
	Ambient Temperature Range	*Normal Temperature*
	Package Size	*16mm*24mm*3mm*
	External Interface	*N/A*
Software Parameter	*Security*	WPA/WPA2
	WiFi Mode	Station/SoftAP/SoftAP+Station
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART *download*/OTA
	Software Development	Mendukung *Cloud Server Development*/SDK untuk custom *software development*
	User Configuration	Set Perintah AT, *Cloud Server, Android/iOS app*
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP

Sumber: (AI-Thinker, 2015)

Pada ESP8266-12E terdapat 22 pin. Setiap pin tersebut memiliki fungsi masing-masing, sehingga pengguna dapat menyesuaikan kebutuhan dengan pin yang ada. Terdapat pin GPIO (*General-purpose input/output)* yang dapat digunaka sebagai *input* maupun *output* sistem. Pin GPIO dapat digunakan untuk menjalan sebuah fungsi tertentu menyesuaikan program yang telah dibuat. Terdapat 1 buah pin ADC yang berfungsi untuk mengkonversi nilai analog dari menjadi digital. Terdapat beberapa pin lain dengan fungsi tertentu yang tertera pada tabel 2.3.

Tabel 2.3 Pin ESP8266-E12

No.	Nama Pin	Fungsi
1	RST	Reset module
2	ADC	A/D konversi, *range* voltase masuk 0-1v*, scope* 0-1024
3	EN	Chip enable
4	IO16	GPIO16; Dapat digunakan untuk wake up
5	IO14	GPIO14; HSPI_CLK
6	IO12	GPIO12; HSPI_MISO
7	IO13	GPIO13; HSPI_MOSI; UART0_CTS
8	VCC	3.3V power supply (VDD)
9	CS0	Chip selection
10	MISO	Salve output Main input
11	IO9	GPIO9
12	IO10	GPIO10
13	MOSI	Main output slave input
14	SCLK	Clock
15	GND	GND
16	IO15	GPIO15; MTDO; HSPICS; UART0_RTS
17	IO2	GPIO2; UART1_TXD
18	IO0	GPIO0
19	IO4	GPIO4
20	IO5	GPIO5
21	RxD	UART0_RXD; GPIO3
22	TxD	UART0_TXD; GPIO1

Sumber: (AI-Thinker, 2015)

2.2.3 *Sensor Optocoupler*

Sensor Optocoupler adalah sebuah sensor yang mengandung sebuah sumber cahaya dan pendeteksi cahaya. Untuk meminimalisir interferensi dari sumber cahaya lainnya, maka sumber cahaya pada sensor menghasilkan cahaya dengan frequensi yang jarang digunakan seperti *infrared*. Dan pendeteksi cahaya pada sensor ini juga dibuat agar hanya dapat menerima cahaya infrared sehingga tidak akan memberikan respon ketika cahaya lain yang terdeteksi (Namaru, 2009). *Sensor Optocoupler* dapat digunakan untuk mengukur kecepatan motor dengan cara mendeteksi banyak *hole* yang melalui *infrared*. Sensor FC-03 ditunjukan pada Gambar 2.5.

Gambar 2.5 Sensor FC-03
Sumber: (V2.0, 2017)

Nama pin yang terdapat pada Sensor FC-03 beserta dengan fungsinya ditunjukan pada Tabel 2.4.

Tabel 2.4 Pin sensor FC-03

No	Nama Pin	Fungsi
1	VCC	3.3V-5V power supply (VDD)
2	GND	Ground
3	D0	Menerima nilai berdasarkan sinyal digital
4	A0	Menerima nilai berdasarkan sinyal analog

2.2.4 *Encoder disk*

Encoder disk dapat digunakan untuk mendeteksi jumlah putaran yang dihasilkan suatu perangkat seperti putaran motor. *Encoder disk* ini digunakan untuk menghitung jumlah putaran yang dihasilkan dalam rentang waktu yang telah ditentukan dengan memanfaatkan lubang-lubang yang terdapat pada *encoder disk* (Xie, 2003). Terdapat beberapa macam *encoder disk* dengan jumlah lubang berbeda yang mengelilinginya, antara lain 20, 36 dan 100. Pada penelitian ini digunakan *encoder disk* dengan 20 *hole* seperti pada Gambar 2.6.

Gambar 2.6 Encoder disk

2.2.5 Teknik Pengukuran *RPM* (Revolution Per Minute)

Penggunaan teknik *incremental encoder* dengan menempatkan sebuah *toothed rotor* atau *encoder disk* di antara sumber cahaya dari sensor dengan penerima cahaya dari sensor. Cahaya yang ditembakan oleh sensor melalui lubang-lubang yang ada pada *encoder disk* dan diterima oleh penerima cahaya disebut sebagai pulse. Setiap pulse yang diterima selama satu detik akan dibagi dengan banyak lubang yang terdapat pada *encoder disk* dan dikalikan dengan enam puluh detik sehingga *RPM* dapat diperoleh. (Uday A. Bakshi, 2009)

Sensor Optocoupler dapat melakukan *sensing* dengan tingkat rendah maupun medium, dengan menggunakan sebuah *encoder disk* dengan lubang yang telah terpisah dengan jarak yang sama antara masing-masing lubang. Ketika menggunakan *Sensor Optocoupler* cara termudah untuk menentukan nilai *RPM* dengan melakukan *monitoring* pada *pulse frequency* yang didapat oleh sensor ketika melakukan *sensing* terhadap *encoder disk*, **Persamaan 2.1** digunakan untuk melakukan perhitungan *RPM* (Arabaci & Bilgin, 2012).

$$RPM = \frac{60\ sec}{Tpr\ \cdot\ Nnoc}\ \cdot\ Ncon \qquad (2.1)$$

Keterangan :

Tpr = waktu pengukuran yang telah ditetapkan

Nnoc = *holes* pada *encoder disk*

Ncon = banyak *holes* yang dideteksi

2.2.6 Arduino Software (IDE)

Arduino IDE adalah sebuah *software* yang digunakan untuk menulis kode program yang dibuat oleh Arduino. Program yang ditulis menggunakan Arduino Software (IDE) disebut *sketch*. *Sketch* ini ditulis dalam editor teks dan disimpan dengan ekstensi file .ino. Editor memiliki fitur untuk memotong / menempel dan mencari / mengganti teks. Area pesan memberi umpan balik sambil menyimpan dan mengekspor dan juga menampilkan kesalahan. Konsol menampilkan output teks oleh Arduino Software (IDE), termasuk pesan kesalahan dan informasi lainnya yang lengkap. Sudut kanan bawah jendela menampilkan papan dan port serial yang dikonfigurasi. Tombol toolbar memungkinkan Anda untuk memverifikasi dan mengunggah program, membuat, membuka, dan menyimpan sketsa, dan membuka monitor serial (Arduino, n.d.).

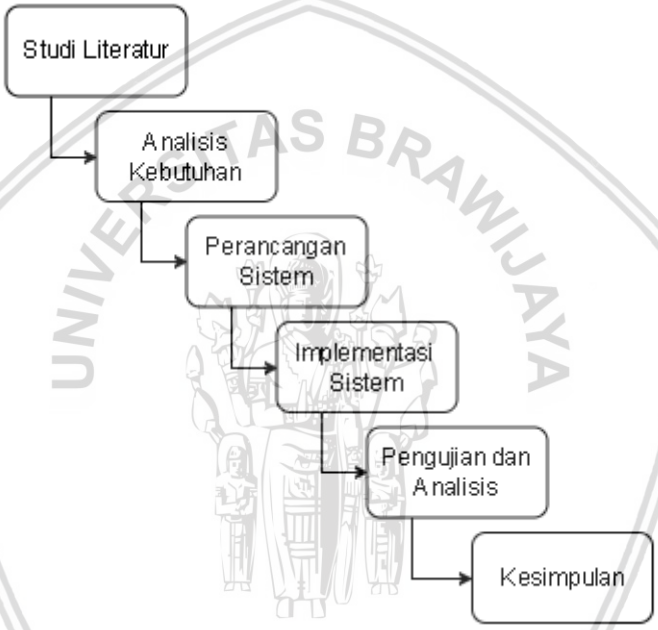
2.2.7 Ajax

Ajax kependekan dari *asynchronous JavaScript and XML*, adalah sebuah teknik yang digunakan untuk membuat sebuah *web* maupun aplikasi yang bersifat interaktif. *Ajax* menitik beratkan kegunaannya dalam melakukan *update* dalam sebuah halaman *web*, menggunakan pengambilan data dari *web server* tanpa melakukan refresh terhadap halaman *web* tersebut. *asynchronous JavaScript and XML* merepresentasikan sebuah kemungkinan mendasar yang dapat diterapkan dalam *web. Asynchronous* dalam *Ajax* berarti ketika *JavaScript* melakukan request ke sebuah server, *JavaScript* akan tetap melakukan eksekusi tanpa menunggu balasan dari server, dan ketika server membalas maka *JavaScript* akan segera menjalankan sebuah *event* tertentu yang berkaitan dengan data yang diterima. *Ajax* adalah sebuah gabungan dari beberapa teknologi yang ada seperti XHTML dan CSS yang digunakan sebagai *standard* tampilan mendasar, *Document Object Model* yang digunakan menampilkan dan berinteraksi dengan user secara dinamis, XML yang digunakan untuk melakukan pertukaran dan memanipulasi data, *XMLHttpRequest* digunakan untuk menerima data secara *asynchronous* dan terakhir adalah *JavaScript* yang digunakan untuk mengikat semua hal teknologi tersebut menjadi suatu kesatuan (Holzner, 2009).

Cara kerja *Ajax* adalah *JavaScript* menggunakan sebuah objek yang sudah tertanam dalam browser, objek tersebut adalah *XMLHttpRequest*, *XMLHttpRequest* digunakan untuk membuka sebuah koneksi antara *JavaScript* dan *web* server *XMLHttpRequest* akan mengunduh data yang dibutuhkan (biasanya berupa data XML) dan setelah data diunduh, maka *JavaScript* akan melakuka update dan menampilkan data tersebut pada halaman *web* yang sedang digunakan. Cara kerja *Ajax* dapat digambarkan seperti contoh berikut: ketika seseorang sedang mengakses google dan mengetikan sesuatu dalam *search box*, maka *JavaScript* akan melakukan komunikasi dengan *web server* untuk mendapatkan informasi yang dibutuhkan tanpa melakukan *refresh* terhadap halaman *web* yang sedang digunakan (Holzner, 2009).

BAB 3 METODOLOGI

Pada bab ini penulis mendeskripsikan studi literatur yang telah didapatkan dari membaca dan mengumpulkan berbagai macam sumber yang dapat digunakan sebagai materi pendukung dari penelitian ini. Literatur yang didapat berupa buku, jurnal, laporan, artikel dan materi lain dari internet. Setelah mempelajari literatur yang telah didapat, maka akan dibuat beberapa poin yang diharapkan dapat tercapai sebagai hasil dari penelitian ini. Dalam bab ini juga akan mendeskripsikan analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan analisis, dan kesimpulan yang dapat diambil dari penelitian ini. Alur metode penelitian dapat dilihat pada Gambar 3.1 di bawah ini.



Gambar 3.1 Proses Metodologi Penelitian

3.1 Studi literatur

Mempelajari dan memahami literatur yang berkaitan dengan pembuatan penelitian ini. Literatur berupa buku, jurnal, laporan, situs dan lain lain. Literatur yang didapat membahas cara menggunakan NodeMCU sebagai *Access Point,* menggunakan *web server* pada NodeMCU, cara menggunakan *Ajax* sebagai teknik pengambilan data dari *web server* dan meng*upadate* data yang tersebut, menggunakan HTML untuk menyesuaikan dan memperbaiki tampilan output pada halaman web, menghitung nilai *RPM* berdasarkan *pulse* dari *encoder disk* yang didapat oleh sensor, mengaplikasikan *interrupt* pada NodeMCU.

3.2 Analisis Kebutuhan

Terdapat beberapa kebutuhan yang dapat mendukung penelitian ini, oleh karena itu dilakukanlah analisis kebutuhan. Kebutuhan pada penelitian ini terbagi menjadi dua yaitu kebutuhan komponen perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*). Pada penelitian *Monitoring RPM* Motor Pada *Smart wheelchair* Berbasis Web Menggunakan *Sensor Optocoupler*, seluruh komponen perangkat keras akan disatukan menjadi sebuah perangkat yang telah diprogram menggunakan perangkat lunak, sehingga dapat bekerja sesuai dengan fungsinya.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah fungsi sistem yang harus dicapai dan dibutuhkan sebagai parameter keberhasilan. Berikut beberapa kebutuhan fungsionalitas dari penelitian ini.

1. Sistem dapat menghitung nilai *RPM* dari masing-masing motor pada *Smart wheelchair* saat motor berputar atau diaktifkan.
2. Sistem dapat menampilkan nilai *RPM* dari masing-masing motor dalam sebuah halaman *web* pada *URL* http://192.168.4.1.
3. Sistem dapat melakukan pembaruan pada halaman web terhadap nilai yang didapat tanpa melakukan *refresh*.
4. Sistem dapat digunakan sebagai *access point* sehingga *client* dapat terhubung dengan perangkat.
5. Sistem dapat menampilkan data XML pada http://192.168.4.1/xml.

3.2.2 Kebutuhan Perangkat Keras

Berikut adalah beberapa perangkat keras yang dibutuhkan dalam pembuatan penelitian ini:

1. NodeMCU	:	untuk mengkalulasi nilai dari pembacaan sensor, memancarkan sinyal *WiFi* dan sebagai *web server*.
2. Sensor FC-03	:	untuk membaca pulse dari *encoder disk*.
3. Encoder disk	:	untuk dibaca oleh sensor.

3.2.3 Kebutuhan Perangkat Lunak

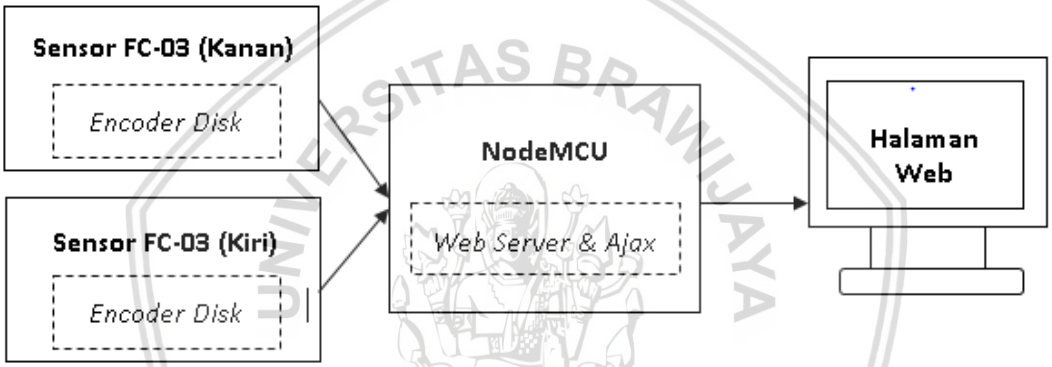
Berikut adalah beberapa perangkat lunak yang dibutuhkan untuk menerapkan *Ajax* dan *Web Server* dalam penelitian ini :

1. Sistem Operasi	:	Windows 10 sebagai pendukung untuk menjalankan Arduino IDE.
2. Arduino IDE	:	Untuk memprogram NodeMCU.

3. *Library* ESP8266*WiFi* : Untuk menggunakan *NodeMCU.*
4. *Library* ESP8266WebServer : Untuk membuat *Web Server*.
5. *Library* ESP8266Client : Untuk menerima *client*.

3.3 Perancangan Sistem

Rancangan sistem ini berupa blok diagram yang menggambarkan hubungan setiap perangkat yang ditunjukan pada Gambar 3.2. Dapat dilihat bahwa NodeMCU digunakan sebagai inti dari perangkat ini. Sensor FC-03 dan *encoder disk* akan ditempatkan di masing-masing motor pada *smart wheelchair*. NodeMCU akan menerima nilai yg didapat setelah kedua sensor FC-03 melakukan *sensing* terhadap *pulse* dari *encoder disk.* Setelah NodeMCU menerima kedua nilai dari masing-masing sensor, maka akan diletakan pada *web server*, lalu akan dikirim dan ditampilkan pada halaman web.



Gambar 3.2 Blok Diagram Sistem

3.4 Implementasi

Implementasi akan dibagi menjadi dua bagian pertama bagian perangkat keras dan kedua bagian perangkat lunak, sehingga menjadi suatu sistem seperti pada Gambar 3.2.

3.4.1 Perangkat keras

Pada bagian ini akan dirancang sebuah rangkaian dari seluruh komponen perangkat keras sehingga menjadi sebuah perangkat. NodeMCU akan disambungkan dengan dua buah Sensor FC-03 dengan menggunakan kabel. *Encoder disk* akan diletakan pada sisi kanan dan kiri dari *smart wheelchair* dan diletakan Sensor FC-03 dengan tempat menyesuaikan kemampuan *sensing* dari sensor.

3.4.2 Perangkat lunak

Pada bagian ini dilakukan instalasi *software* Arduino IDE pada sebuah laptop. Setelah itu, akan dibuat sebuah program menggunakan Bahasa C yang akan diupload dan digunakan pada sistem *monitoring RPM* dengan menyertakan perhitungan *RPM* berdasarkan nilai *pulse* yang didapat dan menyertakan *Ajax* sebagai teknik pengambil data dan pengirim data ke sebuah halaman web.

3.5 Pengujian

Terdapat beberapa pengujian yang dilakukan terhadap sistem *monitoring RPM* untuk mengetahui kualitas sistem. Berikut beberapa pengujian yang dilakukan.

1. Pengujian nilai Sensor FC-03 terhadap nilai *tachometer.*
2. Pengujian waktu tampil data.
3. Pengujian sistem m*onitoring RPM* pada *Smart wheelchair*.
4. Pengujian jarak *WiFi* maksimum NodeMCU.

3.6 Kesimpulan

Tahap pengambilan kesimpulan dapat dilakukan setelah menyelesaikan seluruh tahap pada bab 3 ini. Kesimpulan dapat dibuat berdasarkan hasil pengujian dan analisis terhadap sistem yang telah dibangun. Kesimpulan diharapkan dapat menjadi sebuah bahan referensi dalam hal pengemangan dan penyempurnaan sistem. Pada akhir penulisan ini ada bagian saran yang bertujuan untuk memperbaik kekurangan dari sistem yang telah dibangun.

BAB 4 REKAYASA KEBUTUHAN

4.1 Deskripsi Umum

Pada bab ini akan dijelaskan secara detail mengenai kebutuhan-kebutuhan yang harus dipenuhi terkait perancangan dan implementasi. Dengan melakukan rekayasa kebutuhan diharapkan sistem yang telah dibangun dapat bekerja dengan baik dan sesuai fungsi.

4.1.1 Perspektif Sistem

Sistem *monitoring RPM* roda *smart wheelchair* pada halaman *web* berbasis *Ajax* menggunakan *Sensor Optocoupler* merupakan sebuah alat yang digunakan untuk melakukan *monitoring RPM* dari motor dari masing-masing sisi pada *smart wheelchair* sehingga dapat diketahui jika salah satu motor berputar lebih cepat. Sistem akan dinyatakan berhasil jika dapat menampilkan nilai *RPM* dari masing-masing motor pada sebuah halaman web, melakukan update nilai pada halaman web tanpa melakukan *refresh* dan nilai yang ditampilkan tidak memiliki perbedaan yang signifikan ketika dibandingan dengan *tachometer* yang biasa digunakan untuk menghitung nilai *RPM.*

4.1.2 Ruang Lingkup

Ruang lingkup dari Sistem *monitoring RPM* roda *smart wheelchair* pada halaman *web* berbasis *Ajax* menggunakan *Sensor Optocoupler* adalah perhitungan nilai *RPM* dan penerapan *Ajax* sebagai teknik pengambilan data dari *web server* dan melakukan update terhadap halaman web tanpa adanya *refresh.* Berikut adalah ruang lingkup *sensor* dan *output* dari sistem *monitoring RPM*:

4.1.2.1 Ruang Lingkup Sensor

Ruang lingkup sensor yang digunakan dalam penelitian "*Monitoring RPM* Motor Pada *Smart wheelchair* Berbasis Web Menggunakan *Sensor Optocoupler"* adalah sensor FC-03, terdapat sebuah *optocoupler* dengan sebuah pemancar *infra red* dan *receiver* dengan jarak antara keduanya sebesar 0.6cm.

4.1.2.2 Ruang Lingkup *Output*

Ruang lingkup sensor yang digunakan dalam penelitian "*Monitoring RPM* Motor Pada *Smart wheelchair* Berbasis Web Menggunakan *Sensor Optocoupler"* adalah sebuah Halaman web yang akan menampilkan nilai *RPM* dengan cara mengakses http://192.168.4.1.

4.1.3 Karakteristik Pengguna

Sistem ini berupa menggunakan jaringan lokal dari NodeMCU sehingga memungkinkan beberapa pengguna secara bersamaan. Pengguna hanya melakukan *monitoring* terhadap halaman web saja.

4.1.4 Lingkungan Operasi Sistem

Sistem akan diletakan pada sebuah *smart wheelchair* yang menggunakan dua motor berbeda. *Encoder disk* akan ditempatkan disetiap motor dari *smart wheelchair* dan sensor akan diletakan menyesuaikan dengan posisi *encoder disk*.

4.1.5 Batasan Perancangan dan Implementasi

Berikut adalah batasan perancangan dan implementasi sistem yang telah ditetapkan :

1. NodeMCU hanya digunakan sebagai *access point* dan *web server.*
2. NodeMCU hanya menggunakan jaringan lokal dengan IP 192.168.4.1.
3. *Encoder disk* ditempelkan pada setiap ujung dari setiap motor sisi kanan dan kiri.
4. Banyak *holes* yang dideklarasikan pada variabel "pulsesperturn" menyesuaikan dengan dengan berapa banyak *holes* yang terbaca sensor dalam satu putaran roda *encoder disk.*
5. *Monitoring* hanya dilakukan pada roda sisi kanan dan sisi kiri dari *smart wheelchair*.
6. Pembaruan nilai *RPM* dari setiap sisi pada halaman web hanya dilakukan setiap 1 detik sekali.

4.1.6 Asumsi dan Ketergantungan

Beberapa asumsi dan ketergantungan dalam penerapan sistem adalah sebagai berikut:

1. Nilai variabel "pulsesperturn" pada program berdasarkan banyak *holes* pada *encoder disk* yang terbaca oleh sensor telah disesuaikan dengan 1 putaran roda pada *smart wheelchair.*
2. Pin-pin yang menghubungkan NodeMCU dengan sensor harus sesuai dengan inisialisasi pin yang telah diprogram.
3. Sumber daya menggunakan sebuah *power bank 5600*mAh.

4.2 Rekayasa Kebutuhan

Sub bab ini menjelaskan tentang kebutuhan sistem secara menyuluruh agar sistem *monitoring RPM* dapat berjalan sesuai fungsi dan tujuan. Pada sub bab ini akan dijabarkan kebutuhan fungsional, kebutuhan perangkat keras dan kebutuhan perangkat lunak dari sistem.

4.2.1 Kebutuhan Fungsional

1. Sistem dapat menghitung nilai *RPM* dari masing-masing motor pada *Smart wheelchair* saat motor berputar atau diaktifkan.

   a. Penjelasan dan prioritas

      Fungsi ini berguna untuk mendapatkan nilai *RPM* dari setiap motor. Nilai *RPM* didapatkan dari pembacaan sensor FC-03 terhadap *holes* yang ada pada masing-masing *encoder disk* dari setiap sisi pada kursi roda yang kemudian akan dilakukan perhitungan sehingga didapat nilai *RPM* dari masing-masing motor. Fungsi ini memiliki prioritas tinggi.

   b. Stimulus atau respon system

      Pada saat NodeMCU terhubung dengan daya, maka system akan merespon dengan mengaktifkan sensor dan *interrupt,* dan sensor akan mulai melakukan s*ensing.* Respon dari sistem ini tidak dapat dilihat oleh pengguna, tetapi respon dapat dilihat dalam *serial monitor* pada Arduino IDE.

   c. Kebutuhan fungsional

      Fungsi harus dicapai atau terpenuhi, karena fungsi ini digunakan untuk mendapatkan nilai *RPM* dari roda sisi kanan dan kiri. Nilai yang didapat akan ditampilkan dalam sebuah halaman *web* sehingga dapat diketahui nilai *RPM* paling besar terdapat pada roda sisi kanan ataupun kiri.

2. Sistem dapat menampilkan nilai *RPM* dari masing-masing motor dalam sebuah halaman *web* pada *URL* http://192.168.4.1.

   a. Penjelasan dan prioritas

      Fungsi ini berguna untuk menampilkan nilai *RPM* dari setiap motor pada sebuah web dengan *URL* http://192.168.4.1, yaitu IP lokal dari NodeMCU. Nilai *RPM* yang ditampilkan pada halaman web akan dibedakan menjadi "*RPM* kanan : " dan "*RPM* kiri : ". Tampilnya halaman web pada http://192.168.4.1 memiliki prioritas tinggi dalam sistem.

   b. Stimulus atau respon system

      Pada saat pengguna mengakses *URL* http://192.168.4.1 pada *browser*, maka NodeMCU akan mengirimkan data ke *browser* dan *browser* menampilkan sebuah halaman *web* sebagai sebuah respon yang menampilkan Judul "*Monitoring RPM*" pada bagian tengah atas halaman *web* dan di bawah judul akan ditampilkan nilai dari *RPM* kanan dan *RPM* kiri.

c. Kebutuhan fungsional

Fungsi ini harus tercapai atau terpenuhi, karena fungsi ini digunakan sebagai *output* dari data yang telah diproses oleh NodeMCU untuk menampilkan nilai *RPM* dari setiap motor pada halaman *web*.

3. Sistem dapat menampilkan data XML pada http://192.168.4.1/xml.

a. Penjelasan dan prioritas

Fungsi ini berguna untuk menampilkan perubahan nilai dari masing-masing sensor yang telah dimasukan kedalam sebuah file xml. Tampilnya file XML pada http://192.168.4.1/xml memiliki prioritas tinggi dalam sistem.

b. Stimulus atau respon system

Pada saat pengguna mengakses *URL* http://192.168.4.1/xml pada *browser*, respon yang didapat adalah NodeMCU akan mengirimkan data XML ke *browser* dan *browser* menampilkan sebuah halaman dengan keterangan bahwa data yang ditampilkan berupa data XML (keterangan hanya akan ditampilkan pada browser *laptop*), nilai dari data tersebut akan terus berubah sesuai dengan nilai *RPM* yang telah dikalkulasi oleh NodeMCU pada setiap sisi roda.

c. Kebutuhan fungsional

Fungsi ini harus tercapai atau terpenuhi, karena fungsi ini digunakan sebgai tanda bahwa data XML dapat digunakan dan halaman *web* dapat menggunakan data XML untuk melakukan pembaruan pada nilai *RPM.*

4. Sistem dapat melakukan pembaruan pada halaman web terhadap nilai yang didapat tanpa melakukan *refresh*.

a. Penjelasan dan prioritas

Fungsi ini berguna untuk melakukan pembaruan terhadap nilai *RPM* yang ditampilkan pada halaman *web* tanpa melakukan *refresh.* Pembaruan dilakukan dengan memanfaatkan teknik *Ajax* yaitu melakukan pembaruan terhadap halaman *web* secara asinkron, sehingga hanya data yang membutuhkan pembaruan yang akan berubah pada halaman *web*.

b. Stimulus atau respon system

Pada saat pengguna mengakses *URL* http://192.168.4.1 pada *browser,* maka respon yang akan didapat berupa tampilan sebuah halaman *web* dengan nilai *RPM* kanan dan kiri yang terus berubah mengikuti perubahan dari nilai data pada XML. Halaman web tidak akan melakukan *refresh* dan hanya akan memperbarui nilai pada data data tertentu yaitu *RPM* kanan dan kiri.

c. Kebutuhan fungsional

Fungsi ini harus tercapai dan terpenuhi, karena fungsi ini digunakan untuk menampilkan perubahan data secara terus menerus tanpa melakukan *refresh* pada halaman *web*.

5. Sistem dapat digunakan sebagai *access point* sehingga *client* dapat terhubung dengan perangkat.

a. Penjelasan dan prioritas

Fungsi ini berguna untuk menghubungkan *client* dengan perangkat. Penggunaan mode *access point* membuat NodeMCU dapat berdiri sendiri tanpa harus terhubung dengan jaringan lain yang ada di sekitarnya. Pemberian *SSID* tertentu dan *Password* tertentu dibutuhkan sebagai penanda bahwa jaringan ini akan menghubungkan sebuah perangkat dengan NodeMCU. Fungsi ini memiliki prioritas tinggi, karena pengguna hanya dapat melihat *output* hanya melalui jaringan lokal NodeMCU.

b. Stimulus atau respon system

Pada saat pengguna menghubungkan sebuah perangkat dengan *SSID* dan *Password* dari NodeMCU, maka munculnya keterangan terhubung dengan jaringan NodeMCU adalah sebuah respon dari fungsi ini.

c. Kebutuhan fungsional

Fungsi ini harus tercapai dan terpenuhi karena fungsi hanya dengan terhubung dengan jaringan lokal pengguna dapat mengakses halaman *web.*

4.2.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras terdiri dari NodeMCU, Sensor FC-03, *Encoder disk* dan laptop. Pada Tabel 4. 1 akan dijabarkan setiap perangkat keras yang dibutuhkan beserta penjelasannya.

Tabel 4.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras	Fungsi
NodeMCU	1. Untuk melakukan kalkulasi terhadap nilai yang didapat oleh sensor FC-03. 2. Sebagai *web server.* 3. Untuk mengirimkan data yang telah dikalkulasi ke sebuah halaman web dengan jaringan lokal menggunakan teknik *Ajax*.

Kebutuhan perangkat keras	Fungsi
Sensor FC-03	Untuk membaca *pulse* yang dihasilkan dari *encoder disk*.
Encoder disk	Sebagi sebuah alat yang akan dilakukan pembacaan oleh sensor.
Laptop	Untuk memprogram NodeMCU melalui software Arduino IDE.

4.2.3 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak terdiri dari Windows 10 dan Arduino IDE dan beberapa *library* pendukung. Pada Tabel 4. 2 akan dijabarkan setiap perangkat lunak yang dibutuhkan beserta penjelasannya.

Tabel 4.2 Kebutuhan Perangkat Lunak

Kebutuhan	Fungsi
Arduino IDE	Adalah sebuah *software* yang digunakan untuk membuat sebuah program. Program yang telah dibuat menggunakan Arduino IDE akan di*compile* terlebih untuk mengetahui apakah ada kesalah dalam penulisan program, jika sudah tidak ada kesahalan maka program akan di*upload* ke NodeMCU.
Windows 10	Sebagai sistem operasi yang digunakan laptop.
ESP8266*WiFi*.h	Disertakan dalam program agar dapat di*upload* ke perangkat NodeMCU
ESP8266WebServer.h	Disertakan dalam program untuk membuat sebuah *web server* dalam NodeMCU dengan *port* 80
ESP8266Client.h	Disertakan dalam program untuk melakukan handle terhadap client.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Gambaran Umum Sistem

Gambaran umum dari Sistem *monitoring RPM* roda *smart wheelchair* pada halaman *web* berbasis *Ajax* menggunakan *Sensor Optocoupler* ditunjukan pada Gambar 5. 1. Sistem ini menggunakan 2 buah *encoder disk* dan 2 buah *sensor* FC-03. Menggunakan mikrokontroler NodeMCU yang mengandung ESP8266-12E. Sistem ini menggunakan *web server* dari ESP8266 *library. Encoder disk* akan diletakan pada masing-masing *gear* motor dan diletakan diantara *optocoupler* yang terdapat dalam modul *sensor* FC-03, sehingga setiap *hole* dari *encoder disk* dapat dideteksi. *Sensor* dihubungkan dengan mikrokontroler NodeMCU, sehingga data yang telah diakuisisi dapat diproses oleh NodeMCU. *Web Server* digunakan untuk menangani *request* dari client dan mengirimkan data yang telah diproses NodeMCU ke halaman *web.*

5.1.1 Fungsi Perangkat Keras

Fungsi dari masing-masing perangkat keras yang digunakan dalam sistem ini akan dijelaskan sebagai berikut:

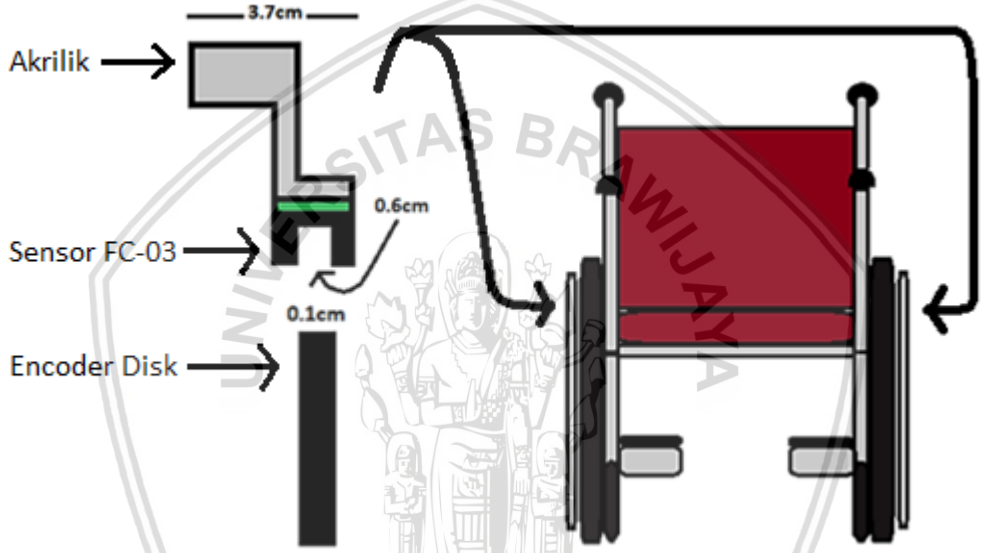
1. *Sensor* FC-03 berfungsi untuk mendeteksi setiap *hole* dari *encoder disk.* menggunakan sinyal digital untuk menentukan pembacaan *hole* dari *encoder disk*.
2. *Encoder disk* memiliki 20 lubang yang dapat dideteksi oleh sensor untuk menentukan perputaran roda.
3. NodeMCU adalah mikrokontroler yang berfungsi untuk menerima sinyal yang didapat oleh sensor dan melakukan perhitungan terhadap nilai tersebut. NodeMCU juga berfungsi untuk mengirimkan data pada halaman *web.*

5.2 Perancangan Sistem

Dalam sub bab perancangan sistem akan dibahas tentang perancangan perangkat keras dan perancangan perangkat lunak yang digunakan.

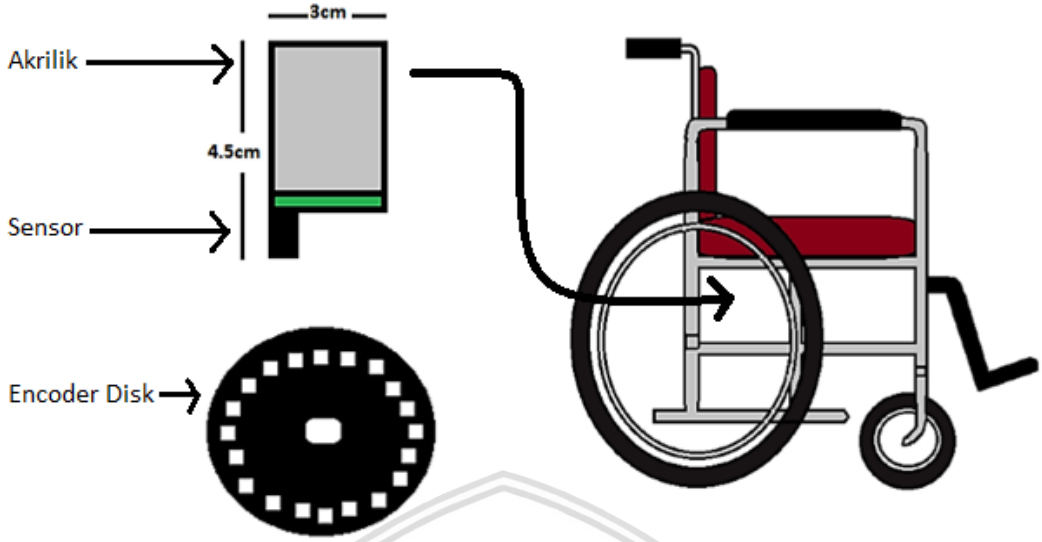
5.2.1 Perancangan Mekanik

Pada setiap sisi *smart wheelchair* hanya terdapat sedikit ruang yang dapat digunakan sebagai tempat untuk meletakan sensor dan juga *encoder disk*, oleh karena itu akrilik dirancang menggunakan ukuran yang sudah ditentukan, ukuran dari akrilik ditunjukan pada Gambar 5. 1 dan 5. 2. Pada Gambar 5. 1 diilustrasikan sebagai bagian depan dari *smart wheelchair* dan juga ilustrasi dari penempatan sensor, yaitu berada di dalam bagian dari roda yang ada pada *smart wheelchair.* Penempatan *sensor* dan *encoder disk* harus sesuai, yaitu *encoder disk* harus berada di tengah pembaca dari *sensor* dan *encoder disk* harus ditempatkan sejajar dengan pembacaan *sensor*, agar sensor dapat membaca setiap *hole* yang ada pada *encoder disk*.



Gambar 5.1 Ilustrasi Bagian Depan

Pada Gambar 5. 2 diilustrasikan penempatan akrilik dari salah satu sisi dari *smart wheelchair.* Penempatan akrilik pada bagian dalam roda hanya membutuhkan sedikit bahan dan ruang, sehingga tidak memrubah tampilan dari *smart wheelchair*, meskipun rancangan penempatan akrilik membutuhkan pengukuran yang sesuai. Ukuran dari akrilik ditunjukan pada Gambar 5.2. Pengukuran akrilik disesuaikan dengan tempat yang tersedia dengan memperhatikan jeruji dari roda tidak boleh bersentuhan dengan akrilik ketika roda berputar. Sistem ini akan digunakan untuk menghitungan nilai *RPM* yang dihasilkan oleh roda ketika digerakan oleh motor, oleh karena itu banyak *hole* dari *encoder disk* disesuaikan dengan roda ketika berputar sebanyak 1 kali.

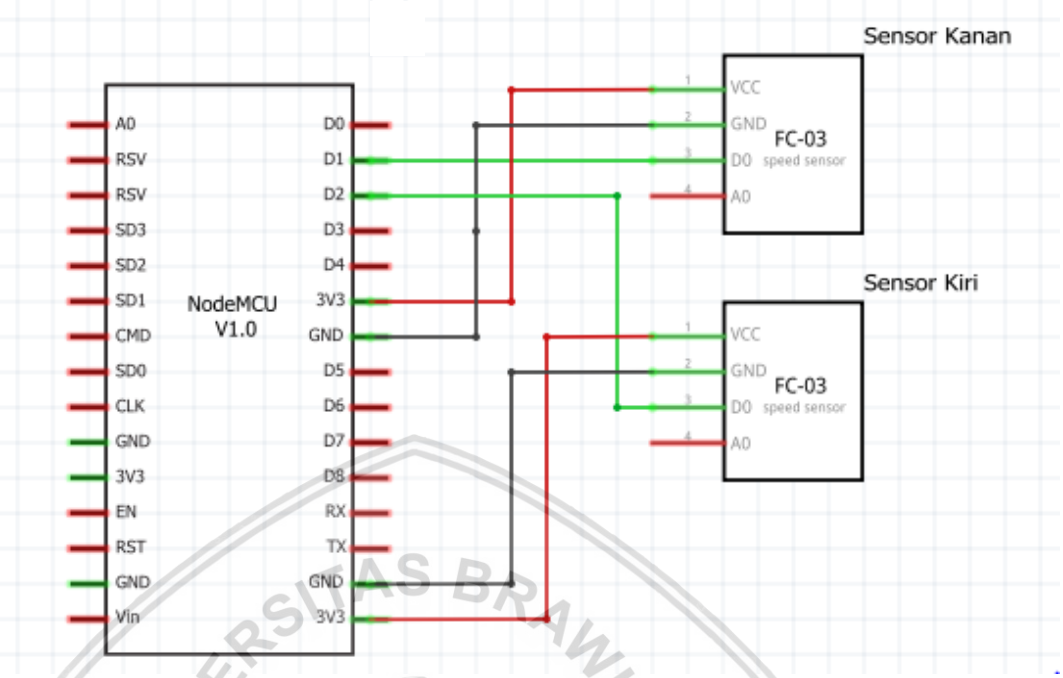


Gambar 5.2 Ilustrasi Bagian Samping

5.2.2 Perancangan Perangkat Keras

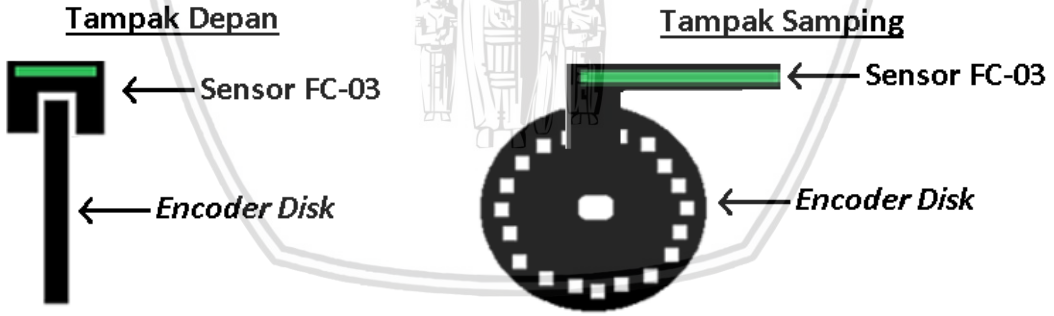
Perancangan perangkat keras dari sistem yaitu menggunakan 2 buah sensor FC-03, 2 buah *encoder disk,* sebuah mikrokontroler NodeMCU, kabel jumper sepanjang 2 meter dan sebuah *power bank* dengan voltase *output* sebesar 5v. *Sensor* FC-03 diletakan pada masing-masing sisi *smart wheelchair* dan akan dihubungkan dengan mikrokontroler NodeMCU menggunakan tegangan 3.3v. Sensor FC-03 digunakan untuk mendeteksi setiap *hole* dari *encoder disk. Hole* yang terdeteksi akan dibaca dan dikirimkan sebagai sinyal *digital.* Setiap *sensor* akan melakukan pembacaan terhadap *hole* dari *encoder disk* selama 1 detik, setiap 1 *hole* yang dideteksi oleh *sensor* akan berupa sinyal digital dengan nilai 1 dan akan dikirimkan ke NodeMCU sebagai tanda adanya perputaran dari roda pada *smart wheelchair* dan mengirimkan nilai 0 ke NodeMCU sebagai tanda tidak adanya perputaran dari roda pada *smart wheelchair*. Banyak *hole* menentukan nilai *RPM* dari masing-masing roda. Banyak *hole* untuk menentukan 1 putaran roda pada *smart wheelchair* disesuaikan dengan 1 kali putaran roda.

NodeMCU akan dihubungkan dengan kedua *sensor* FC-03 menggunakan jumper masing-masing dengan panjang 1 meter. *Sensor* FC-03 memiliki 4 buah pin yaitu Vcc, GND, *Digital Output* dan *Analog Output.* Pada *sensor* FC-03 hanya akan digunakan 3 pin yaitu Vcc, GND dan *Digital Output. Sensor* FC-03 yang pertama akan dihubungkan dengan pin Vcc(3.3v), GND dan D1 pada NodeMCU dan *Sensor* FC-03 yang kedua akan dihubungkan dengan Vcc(3.3v), GND dan D2 pada NodeMCU seperti pada Gambar 5.3. *Sensor* yang digunakan terhubung ke pin Vcc dan GND yang berbeda pada NodeMCU untuk menghindari pengiriman sinyal secara bersamaan dari kedua *sensor*.



Gambar 5.3 Skematik Sistem *Monitoring RPM*

Perancangan pembacaan *hole encoder disk* oleh Sensor FC-03 yaitu dengan menempatnkan *encoder disk* pada celah yang terdapat pada *optocoupler* dari Sensor FC-03. Rancangan penempatan *encoder disk* dan juga Sensor FC-03 dapat dilihat Gambar 5.4.



Gambar 5.4 Penempatan Sensor FC-03 dan *Encoder Disk*

Kalkulasi nilai *RPM* dilakukan dengan cara pertama menetapkan banyak *hole* dalam 1 putaran roda sebanyak 20, kedua melihat berapa banyak *hole* yang terdeteksi dalam setiap detiknya, ketiga melakukan kalkulasi nilai *RPM*. Pada **Persamaan 5.1** dimisalkan terdeteksi 20 *holes* dalam 1detik, maka nilai *RPM* yang didapat adalah 60*RPM,* nilai tersebut didapat dari 20 *holes* yang terdeteksi setiap detiknya yang berarti 1 putaran roda setiap detiknya, maka akan didapat nilai 60*RPM*.

$$RPM = \frac{60\ sec}{Tpr\ \cdot\ Nnoc}\ \cdot\ Ncon \tag{5.1}$$

Keterangan :

Tpr = waktu pengukuran yang telah ditetapkan

Nnoc = banyak *holes* pada *encoder disk*

Ncon = banyak *holes* yang dideteksi

Kalkulasi RPM :

$$RPM = \frac{60\ sec}{1sec\ \cdot\ 20holes}\ \cdot\ 20holes/sec$$

$$RPM = 60$$

5.2.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada sistem ini yaitu memprogram mikrokontroler NodeMCU agar berfungsi untuk menerima nilai *pulses* yang diterima masing-masing sensor dan menampilkannya pada sebuah halaman web. Dalam program dibuat sebuah fungsi *counter* pada masing-masing sensor. Fungsi *counter* akan aktif ketika adanya intrupsi yang diterima setiap sensor, ketika setiap sensor mendeteksi 1 *hole* dari *encoder disk,* maka nilai dari variabel *pulseskanan* dan *pulseskiri* akan bertambah 1. Pembacaan masing-masing sensor akan diaktifkan selama 1000ms. Setelah pembacaan masing-masing sensor selesai dilakukan, maka nilai *incremental* yang telah didapat pada setiap variabel *pulseskanan* dan *pulseskiri* akan dikalkulasikan dengan rumus penghitung *RPM*.

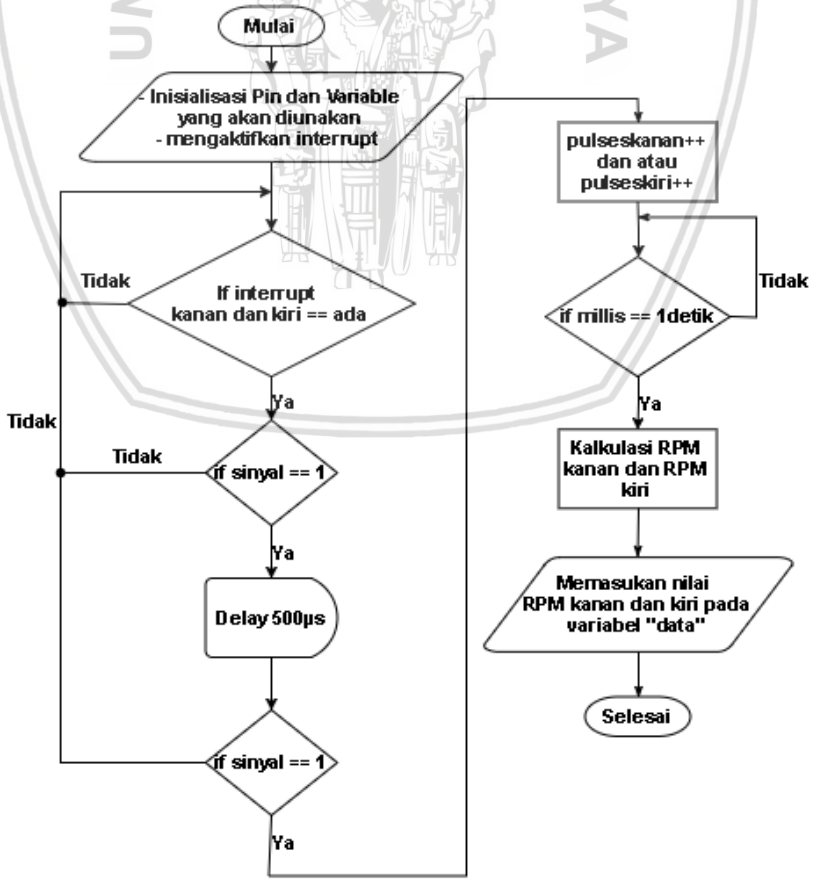
NodeMCU juga diprogram untuk menjadi sebuah *Access Point.* NodeMCU sebagai *Access Point* bertujuan untuk terhubung dengan *client* yang ingin melakukan *monitoring RPM* dari masing-masing pada roda *smart wheelchair. Access Point* NodeMCU dibuat dengan menggunakan sebuah SSID, *password* dan IP lokal dari NodeMCU yaitu 192.168.4.1. SSID dibuat dengan nama "*Monitoring RPM Smart wheelchair*" dan dengan *password* "12345678". *Client* yang ingin melakukan *monitoring* harus menghubungkan perangkatnya dengan jaringan yang telah dibuat oleh NodeMCU dengan SSID dan *password* yang telah ditetapkan.

Web server ESP8266 akan diterapkan pada NodeMCU. *Web server* berfungsi untuk menerima request dari setiap *client* yang telah terhubung. Teknik *Ajax* yang memanfaakan *XMLHttpRequest Object* dan *Javascript* juga diterapkan dalam NodeMCU. *Ajax* berfungsi untuk menerima sebuah objek yang di-*request* oleh *client* dan mengirimkannya kembali ke *client.* Setelah *client* dapat terhubung dengan jaringan lokal dari NodeMCU dan membuka sebuah *browser* untuk melakukan *monitoring,* maka Ajax akan melakukan pertukaran data dengan *web server*. *Ajax* melakukan pertukaran data berupa *request* dari *browser* dan nilai *RPM* dari setiap roda yang terletak pada *server*.

Pada sisi *browser* tampilan halaman *web* dibuat menggunakan HTML. Terdapat kemungkinan dimana layar dari *device* yang digunakan oleh setiap *client* memiliki perberbedaan ukuran, maka tampilan pada halaman *web* dibuat dengan menyesuaikan layar dari *device* yang digunakan.

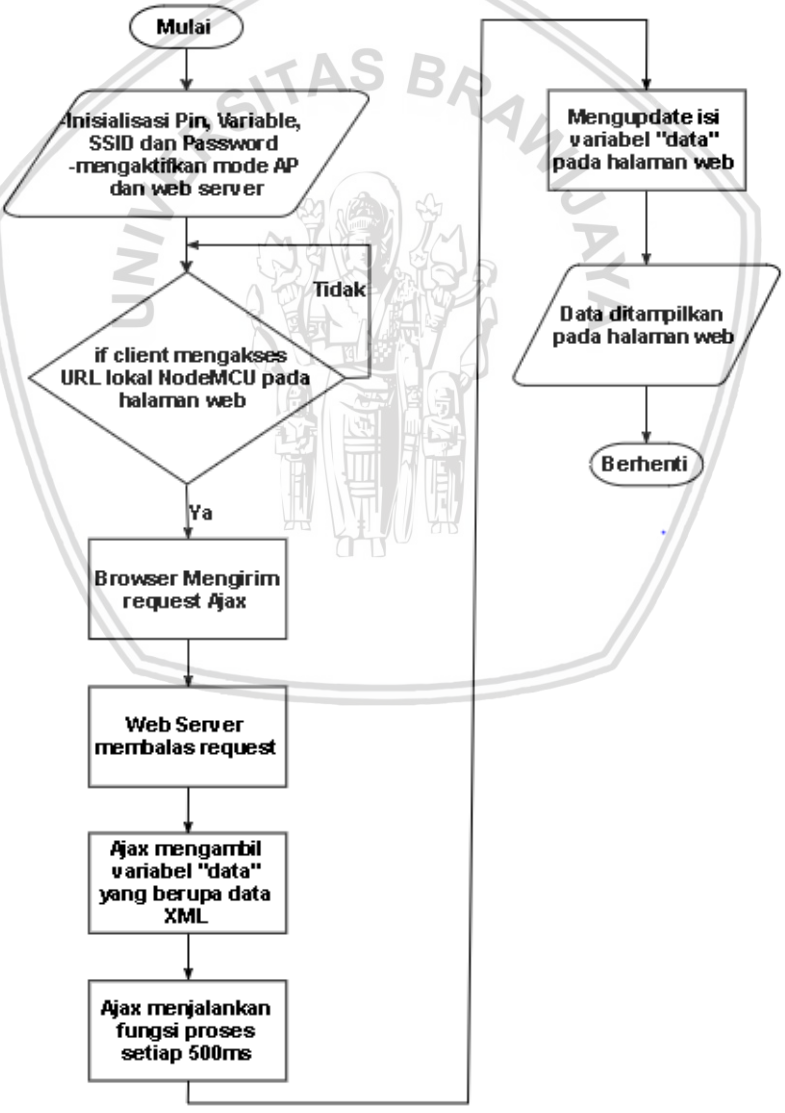
5.2.3.1 Alur Kerja Sistem

Alur kerja sistem terbagi menjadi dua bagian pertama alur akuisisi data dan pengolahan data kedua alur pengiriman data ke halaman web. Alur akuisisi data dan pengolahan data ditunjukan oleh Gambar 5.5, ketika sistem mulai berjalan maka akan dilakukan inisialisasi dari pin dan variabel yang akan digunakan dan juga menjalankan *interrupt* pada pin yang terhubung dengan pin *digital output* dari kedua sensor. *Interrupt* akan langsung berjalan ketika sistem siap beroperasi, lalu akan dicek apakah terdapat ada sinyal dengan nilai 1 yang diterima oleh NodeMCU dari masing-masing sensor yang menandakan adanya *interrupt*. Setelah mengetahui adanya sinyal yang diterima, maka akan dilakukan delay selama 500µm, lalu nilai variabel pulseskanan dan atau pulseskiri akan ditambahkan dengan nilai 1. Pada saat sudah mencapai waktu yang ditentukan yaitu 1detik, maka nilai dari kedua variabel pulseskanan dan pulseskiri akan dimasukan ke dalam sebuah fungsi yang melakukan perhitungan nilai *RPM*. Setelah perhitungan selesai dilakukan maka nilai *RPM* dari masing-masing roda akan dimasukan pada variabel data.



Gambar 5.5 Diagram Alir Sistem Akuisisi Data dan Pengolahan Data

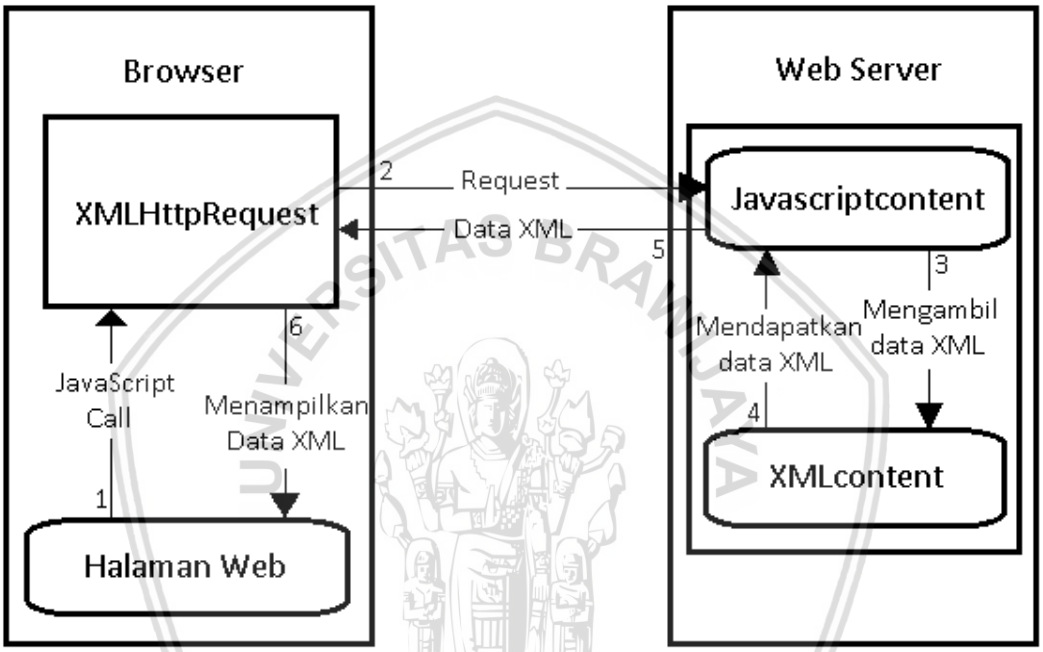
Alur pengiriman data ke halaman web dimulai dengan inisialisasi SSID dan *password* dengan tujuan untuk memberikan nama dari jaringan *WiFi* lokal yang dipancarkan oleh perangkat keras dan juga beberapa variabel yang akan digunakan. Mode *Access point* dan *web server* dengan port 80 juga diaktifkan, sehingga dapat memancarkan jaringan *WiFi* lokal dan menerima *request* dari *client* yang menggunakan protokol HTTP. Pada Gambar 5.6 menunjukan setelah sistem sudah aktif dan terdapat *client* yang telah terhubung dan mengakses halaman web dengan IP 192.168.4.1 maka akan ditampilkan halaman *web* sebanyak 1 kali sebagai respon *web server* terhadap *client* dengan isi nilai *RPM* terakhir dari masing-masing sisi kanan dan kiri. Setelah itu jika masih terdapat *request*, maka akan ditunggu selama 500ms dan sistem mengirimkan isi variabel data terbaru ke halaman *web*, dan jika setelah data terbaru dikirimkan *client* masih melakukan *request* maka akan dilakukan proses yang sama yaitu mengirimkan isi variabel data terbaru ke halaman *web* sampai *client* memutus koneksinya.



Gambar 5.6 Diagram Alir Pengiriman Data

5.2.3.2 Alur Kerja Ajax

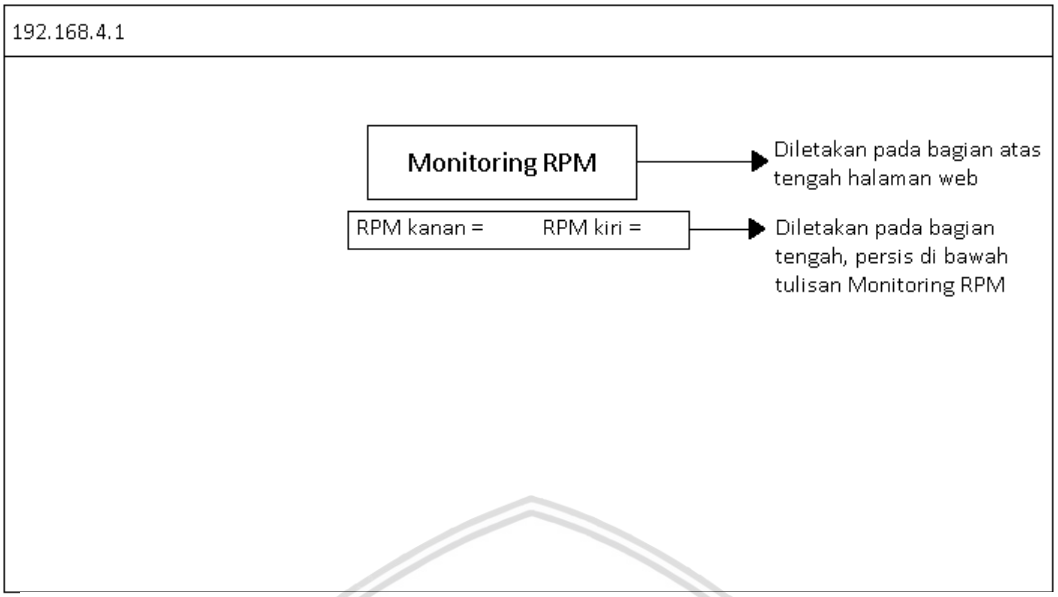
Alur kerja *Ajax* dimulai dengan *client* yang mengakses halaman *web* menggunakan *browser* untuk melakukan *monitoring*, lalu *browser* akan melakukan *Javascript call* dengan mengirim XMLHttpRequest kepada *web server*. *web server* akan menjalankan fungsi WebsiteContent untuk menampilkan halaman *web* sekaligus menjalankan teknik *Ajax* untuk melakukan akuisisi data XML pada fungsi XMLcontent dengan menjalankan Fungsi Javascriptcontent. Data XML yang telah didapat akan dikirimkan ke halaman *web* sebagai respon terhadap XMLHttpRequest. Alur kerja *Ajax* diilustrasikan pada Gambar 5.7.



Gambar 5.7 Alur Kerja Ajax

5.2.3.3 Perancangan Tampilan Halaman Web

Halaman *web* dalam Sistem *Monitoring RPM* Roda *Smart Wheelchair* pada Halaman *Web* Berbasis *Ajax* Menggunakan Sensor *Optocoupler* ini dirancang dengan menempatkan tulisan “Monitoring RPM” pada bagian tengah atas dari halaman web. Nilai RPM kanan dan kiri yang akan direpresentasikan dengan tulisan “RPM kanan = “ dan “RPM kiri = ” yang akan diletakan di bagian bawah tengah dari tulisan “Monitoring RPM”. Ilustrasi dari tampilan halaman web dapat dilihat pada Gambar 5.8.



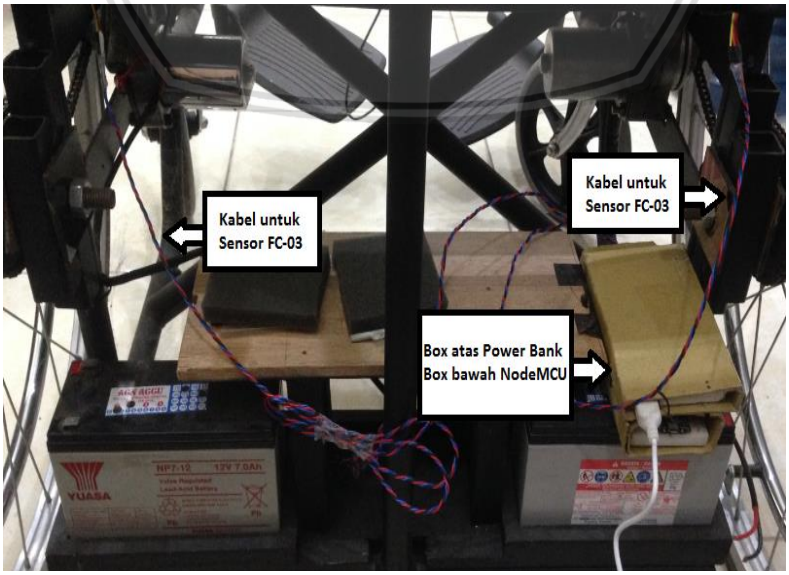
Gambar 5.8 Rancangan Tampilan Halaman Web

5.3 Implementasi Sistem

Dalam implementasi sistem membahas tentang langkah-langkah yang dilakukan untuk membangun sistem ini yang terdiri dari perangkat keras dan perangkat lunak.

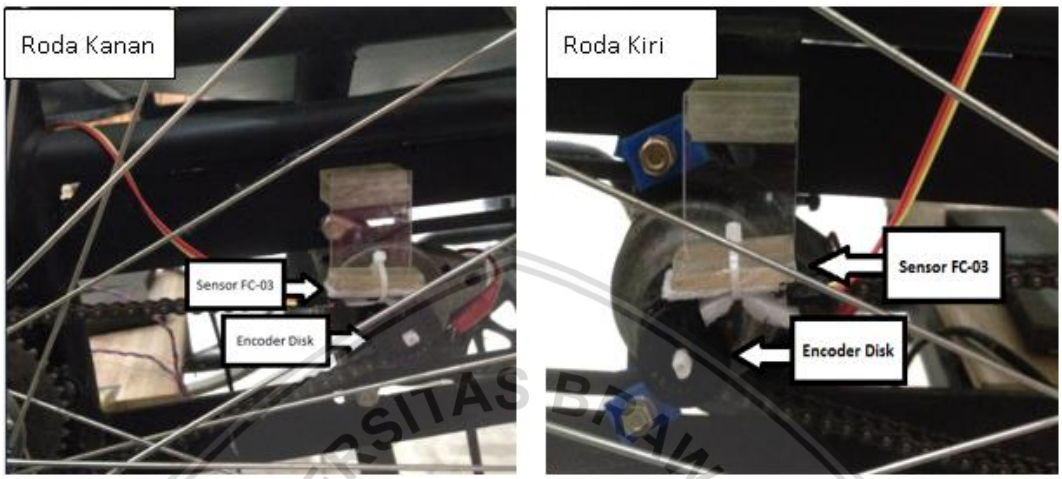
5.3.1 Implementasi Perangkat Keras

Implementasi perangkat keras pada Gambar 5.9 menjelaskan implementasi sistem pada sisi belakang dari *smart wheelchair*. Terdapat box sebagai tempat NodeMCU dan juga *power bank*. NodeMCU terhubung dengan kabel yang masing-masingnya terhubung dengan sensor FC-03.



Gambar 5.9 Implementasi perangkat keras

Implementasi perangkat keras pada Gambar 5.10 menunjukan implementasi Sensor FC-03 pada setiap sisi dari *smart wheelchair*. Terdapat akrilik yang dipasang pada *smart wheelchair* dengan tujuan agar Sensor FC-03 dapat diletakan pada tempat yang seharusnya. Terdapat pula *encoder disk* dengan 20 holes yang akan dibaca oleh sensor, sehingga NodeMCU dapat mengetahui nilai RPM roda pada *smart wheelchair*.



Gambar 5.10 Implementasi Sensor FC-03 dan *Encoder Disk*

5.3.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak dilakukan dengan cara memprogram perangkat keras agar dapat berjalan sesuai fungsi dan tujuannya. Pemrograman dilakukan menggunakan bahasa C dengan menggunakan aplikasi Arduino IDE. Pada bagian awal program beberapa *library* diaplikasikan agar dapat mempermudah dan mempermudah penulisan program yang dibuat pada baris-baris berikutnya. Pada Gambar 5.11 menunjukan *library* yang digunakan pada sistem ini

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <WiFiClient.h>
```

Gambar 5.11 Inisialisasi library

5.3.2.1 Implementasi Pembacaan Sensor, Mode *Access Point* dan *Web Server*

Pada Gambar 5.12 menunjukan implementasi sensor dan mode *access point, Web Server* dan Ajax pada NodeMCU. Pertama dapat dilihat kedua sensor deprogram untuk menggunakan pin 5 dan pin 4 pada NodeMCU sebagai input dan memasangkan fungsi *interrupt* dalam kondisi *RISING* pada kedua pin tersebut, ketika terdapat *interrupt* maka fungsi counterkanan dan counter kiri akan dijalankan. Implementasi *Access Point* pada NodeMCU ditunjukan pada *WiFi*.softAP(ssid, password) yaitu menggunakan variabel ssid dan password yang telah dibuat pada variabel global. Pada server.on("/", WebsiteContent) adalah implementasi dari webserver yang digunakan untuk merespon *client,* ketika *client*

mengakses URL 192.168.4.1/ pada *browser* maka fungsi WebsiteContent akan digunakan. Pada server.on("/xml", XMLcontent) adalah implementasi dari webserver yang digunakan untuk merespon *client,* ketika *client* mengakses URL 192.168.4.1/xml pada *browser* maka fungsi XMLcontent akan digunakan. Mengaktifkan *web server* pada sistem ini dilakukan dengan menggunakan perintah server.begin().

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(5, INPUT);
  pinMode(4, INPUT);
  attachInterrupt(5, counterkanan, RISING);
  Serial.println("Interrupt pada pin 5 aktif");
  attachInterrupt(4, counterkiri, RISING);
  Serial.println("Interrupt pada pin 4 aktif");
  Serial.println("..............................");
  Serial.print("Configuring access point...");
  WiFi.mode(WIFI_AP);//Set menjadi Access point
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP Address : ");
  Serial.print(myIP);
  server.on("/", WebsiteContent);
  server.on("/xml", XMLcontent);
  server.begin();
  Serial.println("HTTP server started");
}
```

Gambar 5.12 Setup sistem

Pada Gambar 5.13 menjelaskan mengenai *interrupt* yang digunakan pada pin 4 dan pin 5. Fungsi counterkanan dan counterkiri dipanggil saat terdeteksi *interrupt* dari masing-masing sensor, di saat salah satu sensor mendeteksi adanya *interrupt* maka akan dilakukan delay sebesar 500µs setelah itu melakukan *increment* terhadap variabel pulseskanan dan pulseskiri.

```
void counterkanan() {
  if (  digitalRead (pinKanan) && (micros() - debouncekanan > 500) && digitalRead (pinKanan) )
    debouncekanan = micros();
    pulseskanan++; //menambah 1 pada pulses kanan
  }
}
void counterkiri() {
  if (  digitalRead (pinKiri) && (micros() - debouncekiri > 500) && digitalRead (pinKiri) ) {
    debouncekiri = micros();
    pulseskiri++; //menambah 1 pada pulses kiri
  }
}
```

Gambar 5.13 Fungsi Interrupt

Pada gambar 5.14 menjelaskan tentang penggunaan *library* ESP8266WebServer agar dapat membuat sebuah *web server. Web server* dibuat pada variabel server dengan menggunakan port 80 yang berarti hanya akan

digunakan pada protocol HTTP saja. Variabel ssid dan password menunjukan nama dan sandi dari jaringan *WiFi* yang akan dipancarkan oleh NodeMCU, pemberian nama dan sandi digunakan sebagai penanda jaringan NodeMCU.

```
ESP8266WebServer server(80); //menggunakan server HTTP

const char* ssid = "Monitoring RPM";
const char* password = "12345678";
```

Gambar 5.14 Web server, SSID dan password

5.3.2.2 Implementasi Perhitungan *RPM*

Pada gambar 5.15 menjelaskan tentang perhitungan *RPM*. Perhitungan *RPM* dari masing-masing fungsi *RPM*kanann dan *RPM*kirii akan dilakukan ketika telah menyentuh 1detik. Saat fungsi *RPM*kanann atau *RPM*kirii sudah aktif maka fungsi *interrupt* pada pin 4 maupun pin 5. Setelah *interrupt* dimatikan maka perhitungan *RPM* akan dilakukan berdasarkan perhitungan yang diterapkan dan dengan menggunakan nilai dari variabel pulseskanan maupun pulseskiri yang telah didapat pada fungsi counterkanan dan counterkiri. Nilai *RPM* diletakan pada variabel *RPM*kanan dan *RPM*kiri. Nilai pulseskanan dan pulseskiri akan diubah menjadi 0 setelah perhitungan selesai dilakukan dengan tujuan untuk mengulang pendeteksian pada sensor, karena nilai *RPM* pada detik tersebut telah didapatkan.

```
int rpmkanann() {
  if (millis() - timeoldkanan >= 1000) {
    /*Uptade every one second, this will be equal to reading frecuency (Hz).*/
    //Don't process interrupts during calculations
    detachInterrupt(4);
    //Note that this would be 60*1000/(millis() - timeold)*pulses if the interrupt
    //happened once per revolution
    rpmkanan = 60 * pulseskanan  /  pulsesperturn;
    //to represent the next second
    timeoldkanan = millis();
    pulseskanan = 0;
    Serial.println("RPM kanan : " + (String)rpmkanan);
    //Restart the interrupt processing
    attachInterrupt(4, counterkanan, RISING);
  }
  return rpmkanan;
}
  if (millis() - timeoldkiri >= 1000) {
    /*Uptade every one second, this will be equal to reading frecuency (Hz).*/
    //Don't process interrupts during calculations
    detachInterrupt(5);
    //Note that this would be 60*1000/(millis() - timeold)*pulses if the interrupt
    //happened once per revolution
    rpmkiri = 60 * pulseskiri /  pulsesperturn;
    //to represent the next second
    timeoldkiri = millis();
    pulseskiri = 0;
    Serial.println("RPM kiri : " + (String)rpmkiri);
    //Restart the interrupt processing
    attachInterrupt(5, counterkiri, RISING);
  }
  return rpmkiri;
```

Gambar 5.15 Perhitungan RPM

Setelah nilai *RPM* telah ditampung oleh variabel *RPM*kanan dan *RPM*kiri, maka nilai tersebut akan dipanggil oleh variabel data yang berapa pada fungsi seperti pada Gambar 5.16.

```
void loop() {
  // put your main code here, to run repeatedly:
  //  inckn = inckn +1;
  //  inckr = inckr +1;
  //  data = "inc kn = "+(String)inckn+" inc kr = "+(String)inckr;

  data = "RPM Kanan =  " + String(rpmkanann()) + " RPM kiri =  " + String(rpmkirii());
  server.handleClient();
}
```

Gambar 5.16 Variabel data pada fungsi Loop

5.3.2.3 Implementasi Ajax

Pembuatan halaman *web* Pada Gambar 5.17 dengan menggunakan fungsi WebsiteContent menggunakan html. Fungsi WebsiteContent akan menampilkan sebuah halaman web dengan konten nilai dari variabel data. Setelah nilai dari variabel data yang pertama telah ditampilkan maka fungsi Javascriptcontent akan dipanggil.

```
void WebsiteContent() {
  Javascriptcontent();
  website = "<html>";
  website += "<head><meta name=\"viewport\" content=\"width=device-width, minimumscale=1.0, ma
  website += "<body onload='process()'><div align=\"center\" id='div1'>" + data + "</div></boc
  website += javascript;
  server.send(200, "text/html", website);
}
```

Gambar 5.17 Fungsi WebsiteContent

Fungsi Javascript content dijelaskan Pada Gambar 5.18 dimulai dengan membuat sebuah variabel sebagai XMLHttpObject yang akan digunakan untuk mengecek apakah browser yang digunakan oleh *client* sudah mendukung XMLHttpRequest Object atau belum. Setelah itu maka fungsi response akan dijalankan, fungsi response digunakan untuk mendapatkan data berupa data XML yang ada pada fungsi XMLcontent dan data tersebut akan di*update* setiap 500ms.

```
void Javascriptcontent() {
  javascript = "<SCRIPT>\n";
  //javascript += "alert('hello');\n"; //debug javascirpt jalan atau tidak
  javascript += "var xmlHttp=createXmlHttpObject();\n";
  javascript += "function createXmlHttpObject(){\n";
  javascript += "if(window.XMLHttpRequest){\n";
  javascript += "xmlHttp=new XMLHttpRequest();\n";
  javascript += "}else{\n";
  javascript += "xmlHttp=new ActiveXObject('Microsoft,XMLHTTP')\n";
  javascript += "}\n";
  javascript += "return xmlHttp;\n";
  javascript += "}\n";
  javascript += "\n";
  javascript += "function response(){\n";
  javascript += "xmlResponse=xmlHttp.responseXML;\n";
  javascript += "xmldoc = xmlResponse.getElementsByTagName('data');\n";
  javascript += "message = xmldoc[0].firstChild.nodeValue;\n";
  javascript += "document.getElementById('div1').innerHTML=message;\n";
  javascript += "}\n";
  javascript += "function process(){\n";
  javascript += "xmlHttp.open('PUT','xml',true)\n";
  javascript += "xmlHttp.onreadystatechange=response\n";
  javascript += "xmlHttp.send(null);\n";
  javascript += "setTimeout('process()',500);\n";
  javascript += "}\n";
  javascript += "</SCRIPT>\n";
```

Gambar 5.18 Fungsi Javascriptcontent

Fungsi XMLcontent yang dijelaskan pada Gambar 5.19 adalah sebuah fungsi yang menampung data berupa nilai *RPM* kanan maupun kiri. Fungsi ini dibutuhkan dalam proses penampilan data pada halaman *web.*

```
void XMLcontent() {
  XML = "<?xml version='1.0'?>";
  XML += "<data>";
  XML += data;
  XML += "</data>";

  server.send(200, "text/xml", XML);
}
```

Gambar 5.19 Fungsi XMLContent

BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian Nilai *RPM* Bacaan Prototipe Sensor FC-03

Dalam sistem *monitoring RPM* menggunakan 2 buah sensor FC-03, masing-masing sensor merepresentasikan sisi kanan dan sisi kiri pada prototipe. Pengujian kedua sensor dilakukan pada waktu yang berbeda, tetapi menggunakan prosedur yang sama. Pengujian dilakukan dengan membandingkan nilai *RPM* bacaan Sensor FC-03 setelah dikalkulasi dengan niai *RPM* bacaan *tachometer* terhadap sebuah roda yang digerakan oleh sebuah motor DC yang diberi tegangan 5V, dengan 10 tingkat kecepatan berbeda berdasarkan nilai *PWM*. *Tachometer* dan kedua sensor akan mengakuisisi data dan menampilkan nilai *RPM* berdasarkan setiap tingkat kecepatan dari motor DC. Nilai *tachometer* dan kedua sensor akan dibandingkan sehingga didapat persentase error dari kedua sensor.

6.1.1 Tujuan Pengujian

Pengujian dilakukan dengan tujuan untuk mengetahui selisih nilai dari masing-masing sensor ketika dibandingkan dengan nilai *tachometer* sehingga dapat ditampilkan persentase error dari masing-masing sensor.

6.1.2 Prosedur Pengujian

Berikut langkah-langkah pengujian Pengujian Nilai Sensor FC-03 Terhadap Nilai *Tachometer.*

1. Memberi tanda titik awal pada roda yang terhubung dengan motor DC.
2. Menghubungkan NodeMCU dengan laptop.
3. Membuka serial monitor pada Arduino IDE.
4. Menyalakan dan mengarahkan *tachometer* menghadap ke titik awal pada roda.
5. Merekam nilai yang ditampilkan *tachomter* dan sensor pada serial monitor.
6. Mencatat dan membandingkan nilai yang sudah direkam.

Perhitungan persentase error menggunakan **Persamaan 6.1**. Berikut adalah persamaan yang digunakan.

$$Persentase\ Error = \frac{\text{Nilai Sensor} - \text{nilai } Tachometer}{\text{nilai } Tachometer}\ x\ 100 \qquad (6.1)$$

6.1.3 Hasil dan Analisis Nilai *RPM* Sensor FC-03 (kiri)

Tabel 6.1 Pengujian Sensor FC-03 (Kiri)

RPM Prototipe Kiri					
No	Nilai PWM	Nilai RPM Bacaan Sistem (RPM)	Pulse	Nilai RPM Bacaan Tachometer (RPM)	Persentase Error
1	120	90	30	89,6	0.44%
2	120	90	30	89,6	0.44%
3	120	90	30	88,5	1.69%
4	120	87	29	88,5	1.69%
5	120	90	30	88,7	1.46%
6	135	99	33	100,4	1.39%
7	135	102	34	100,8	1.19%
8	135	99	33	100,8	1.78%
9	135	102	34	100,6	1.39%
10	135	99	33	99,9	0.90%
11	150	108	36	107,4	0.55%
12	150	108	36	107,4	0.55%
13	150	105	35	106,8	1.68%
14	150	108	36	107,1	0.84%
15	150	108	36	107,1	0.84%
16	165	114	38	114,3	0.26%
17	165	114	38	113,3	0.61%
18	165	114	38	113,3	0.61%
19	165	114	38	113,4	0.52%
20	165	111	37	113,1	1.85%
21	180	120	40	120,1	0.08%
22	180	123	41	120,1	2.41%
23	180	117	39	119,6	2.17%
24	180	120	40	119,6	0.33%
25	180	120	40	119,4	0.50%
26	195	126	42	124,8	0.96%
27	195	126	42	124,8	0.96%
28	195	123	41	125,2	1.75%
29	195	126	42	125,2	0.63%
30	195	126	42	125,3	0.55%
31	210	129	43	129,1	0.07%
32	210	126	42	128,7	2.09%
33	210	129	43	128,6	0.31%
34	210	129	43	128,6	0.31%
35	210	129	43	128,6	0.31%
36	225	132	44	132,1	0.07%
37	225	132	44	132,1	0.07%

38	225	132	44	132,1	0.07%
39	225	132	44	132,1	0.07%
40	225	132	44	132,2	0.15%
41	240	135	45	135,6	0.44%
42	240	135	45	135,7	0.51%
43	240	135	45	136,1	0.80%
44	240	135	45	135,9	0.66%
45	240	138	46	135,9	1.54%
46	255	144	48	142,5	1.05%
47	255	141	47	142,5	1.05%
48	255	144	48	142,5	1.05%
49	255	141	47	142,4	0.98%
50	255	144	48	142,6	0.98%
Average Percent Error :					0.87%

Dengan **Persamaan 6.2** maka dapat dibuktikan kebenaran dari setiap nilai yang ditampilkan oleh sistem *monitoring RPM*. Berikut adalah perhitungan dari data ke 50 dari Tabel 6.1.

$$RPM = \frac{60\ sec}{Tpr\ \cdot\ Nnoc} \cdot Ncon \qquad (6.2)$$

$$RPM = \frac{60\ sec}{1sec\ \cdot 20\text{holes}} \cdot 48holes$$

$$RPM = 144$$

Dengan menggunakan **Persamaan 6.3**, maka dapat menghasilkan nilai error dari setiap data yang telah diperoleh. Berikut adalah perhitungan persentase error dari data ke 50 dari Tabel 6.1.

$$Persentase\ Error = \frac{\text{Nilai Sensor} - \text{nilai } Tachometer}{\text{nilai } Tachometer}\ x\ 100 \qquad (6.3)$$

$$Persentase\ Error = \frac{144 - 142.6}{142.6}\ x\ 100$$

$$Persentase\ Error = 0.98\%$$

Setelah mendapatkan persentase error dari setiap data pada Tabel 6.1 maka dapat ditentukan nilai rata-rata persentase error sebesar 0.87%.

6.1.4 Hasil dan Analisis Nilai *RPM* Sensor FC-03 (kanan)

Tabel 6.2 Pengujian Sensor FC-03 (Kanan)

RPM Prototipe Kanan					
No	Nilai PWM	Nilai RPM Bacaan Sistem (RPM)	Pulse	Nilai RPM Bacaan Tachometer (RPM)	Persentase Error
1	120	84	28	85,4	1.63%
2	120	87	29	85,4	1.87%
3	120	84	28	85,4	1.63%
4	120	87	29	86,1	1.04%
5	120	87	29	86,1	1.04%
6	135	96	32	96,3	0.31%
7	135	99	33	96,3	2.80%
8	135	96	32	96,7	0.72%
9	135	96	32	96,8	0.82%
10	135	96	32	96,8	0.82%
11	150	105	35	104,0	0.96%
12	150	105	35	105,2	0.19%
13	150	105	35	105,2	0.19%
14	150	105	35	105,4	0.37%
15	150	105	35	105,4	0.37%
16	165	111	37	112,9	1.68%
17	165	111	37	112,9	1.68%
18	165	114	38	112,6	0.97%
19	165	111	37	112,1	0.98%
20	165	111	37	112,1	0.98%
21	180	117	39	118,9	1.59%
22	180	120	40	118,7	1.09%
23	180	120	40	118,7	1.09%
24	180	117	39	118,9	1.59%
25	180	120	40	119,2	0.67%
26	195	123	41	124,2	0.96%
27	195	126	42	124,2	1.44%
28	195	123	41	124,2	0.96%
29	195	126	42	124,2	1.44%
30	195	123	41	124,5	1.20%
31	210	129	43	128,3	0.54%
32	210	126	42	128,1	1.63%
33	210	129	43	127,9	0.86%
34	210	129	43	127,9	0.86%
35	210	126	42	127,8	1.40%
36	225	132	44	131,3	0.53%
37	225	129	43	131,0	1.52%

38	225	132	44	131,0	0.76%
39	225	132	44	131,2	0.60%
40	225	132	44	131,3	0.53%
41	240	135	45	135,2	0.14%
42	240	135	45	135,2	0.14%
43	240	135	45	134,8	0.14%
44	240	135	45	134,8	0.14%
45	240	135	45	134,5	0.37%
46	255	144	48	141,3	1.91%
47	255	141	47	141,2	0.14%
48	255	141	47	141,4	0.28%
49	255	141	47	141,4	0.28%
50	255	141	47	141,3	0.21%
Average Percent Error :					0.92%

Dengan **Persamaan 6.4** maka dapat dibuktikan kebenaran dari setiap nilai yang ditampilkan oleh sistem *monitoring RPM*. Berikut adalah perhitungan dari data ke 50 dari Tabel 6.2.

$$RPM = \frac{60\ sec}{Tpr\ \cdot\ Nnoc} \cdot Ncon \qquad (6.4)$$

$$RPM = \frac{60\ sec}{1sec\ \cdot 20\text{holes}} \cdot 47holes$$

$$RPM = 141$$

Dengan menggunakan **Persamaan 6.5**, maka dapat menghasilkan nilai error dari setiap data yang telah diperoleh. Berikut adalah perhitungan persentase error dari data ke 50 dari tabel 6.2.

$$Persentase\ Error = \frac{\text{Nilai Sensor} - \text{nilai } Tachometer}{\text{nilai } Tachometer}\ x\ 100 \qquad (6.5)$$

$$Persentase\ Error = \frac{141 - 141.3}{141.3}\ x\ 100$$

$$Persentase\ Error = 0.21\%$$

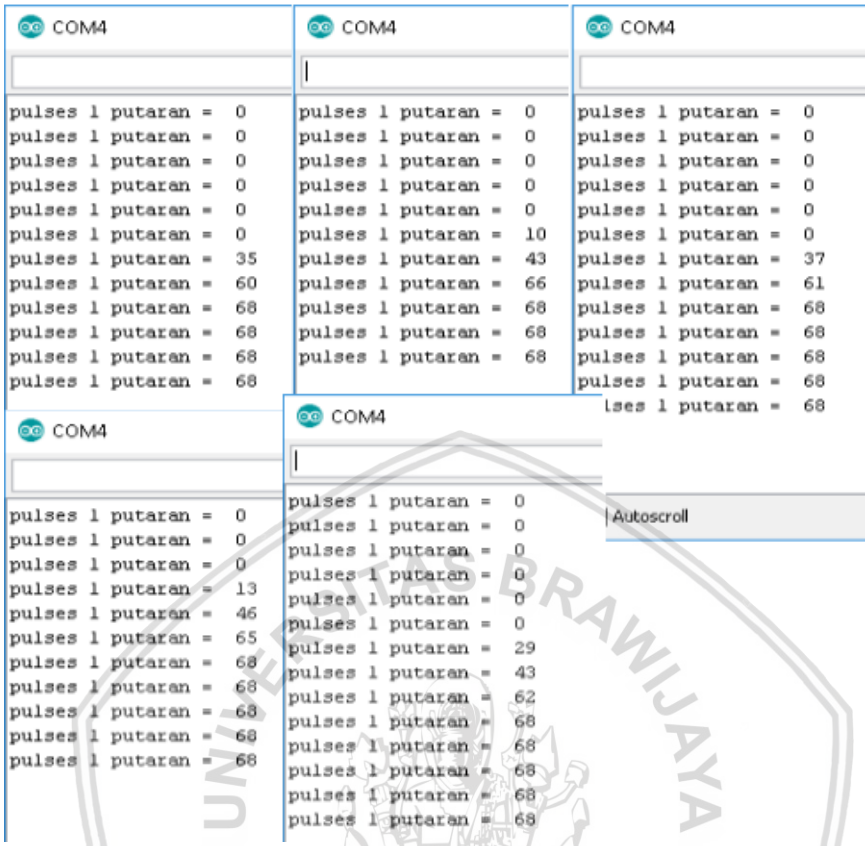
Setelah mendapatkan persentase error dari setiap data pada Tabel 6.2 maka dapat ditentukan nilai rata-rata persentase error sebesar 0.92%.

6.2 Pengujian Sistem *Monitoring RPM* pada *Smart wheelchair*

Pengujian dilakukan pada roda kanan dan kiri dari *smart wheelchair*. Pengujian dilakukan pada waktu yang berbeda dan juga dilakukan dengan kecepatan yang berbeda. Pengujian dilakukan dengan cara membandingkan nilai pada *tachometer* dan nilai pada sistem *monitoring RPM*. Pada pengujian ini *encoder disk* diletakan 1 poros dengan gear yang menggerakan roda, oleh karena itu dibutuhkan penyesuaian mengenai banyak hole pada 1 putaran roda. Jumlah hole untuk 1 putaran disesuaikan dengan cara menentukan titik awal roda pada seperti pada Gambar 6.1.

Gambar 6.1 Titik Awal roda

Setelah titik awal ditentukan maka roda akan diputar sebanyak 1 kali, sehingga didapat berapa banyak hole yang terdeteksi oleh sensor. Hole terdeteksi sebanyak 68 dalam 5 kali percobaan ditunjukan pada Gambar 6.2.



Gambar 6.2 Banyak Hole 1 putaran Roda

6.2.1 Tujuan Pengujian

Menunjukan bahwa sistem tetap dapat berfungsi meskipun menggunakan ukuran roda yang berbeda.

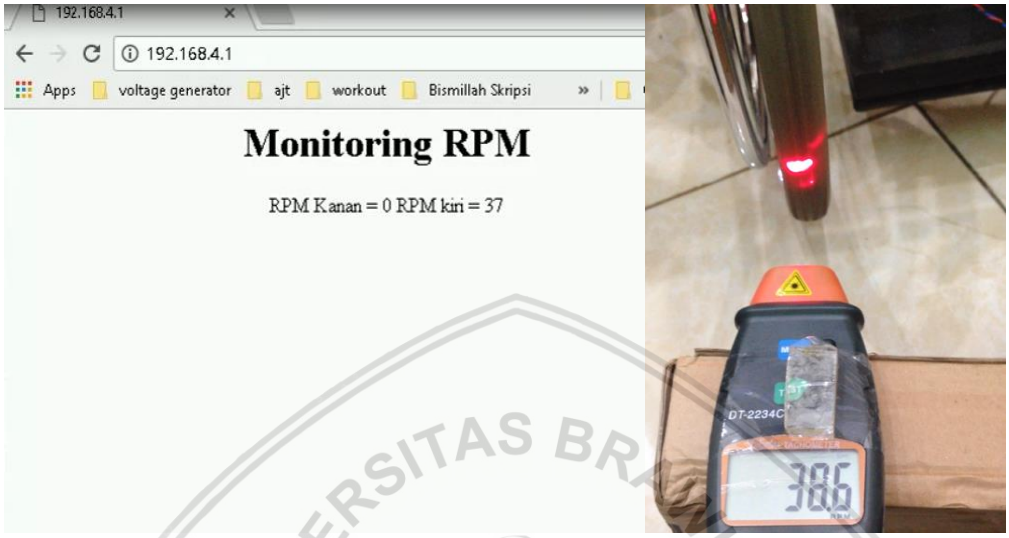
6.2.2 Prosedur Pengujian

Berikut langkah-langkah pengujian Pengujian Nilai Sensor FC-03 Terhadap Nilai *Tachometer.*

1. Memberi tanda titik awal pada roda pada *smart wheelchair*.
2. Menghubungkan NodeMCU dengan laptop.
3. Membuka serial monitor pada Arduino IDE.
4. Menyalakan dan mengarahkan *tachometer* menghadap ke titik awal roda.
5. Merekam nilai yang ditampilkan *tachomter* dan sensor pada serial monitor.
6. Memutar roda secara manual.
7. Mencatat dan membandingkan nilai yang sudah direkam.

6.2.3 Hasil dan Analisis Nilai *RPM* Sensor FC-03 (kiri)

Bacaan nilai *RPM* antara sistem *Monitoring RPM* dengan *tachometer* ditunjukan pada Gambar 6.3. Kedua nilai tersebut adalah data pertama yang ditampilkan pada Tabel 6.3.



Gambar 6.3 Bacaan Monitoring RPM roda kiri Sistem

Nilai *RPM* roda kiri didapat dengan membandingkan nilai *RPM* sistemdan *tachometer* berdasarkan 10 pengujian yang telah dilakukan. Data ditampilkan pada Tabel 6.3.

Tabel 6.3 RPM roda kiri Smart wheelchair

RPM Kiri Smart Wheelchair			
No	RPM Bacaan Sistem (RPM)	RPM Bacaan Tachometer (RPM)	Persentase Error
1	37	38,6	4.14%
2	39	38,6	1.03%
3	40	40,6	1.47%
4	39	40,6	3.94%
5	40	40,6	1.47%
6	45	44,2	1.81%
7	44	44,2	0.45%
8	39	39,5	1.26%
9	38	39,5	3.79%
10	39	39,5	1.26%
Average Percent Error :			2.06%

Dengan menggunakan **Persamaan 6.6**, maka didapat nilai error dari setiap data yang telah diperoleh. Berikut adalah perhitungan persentase error menggunakan data terakhir Tabel 6.3.

$$Persentase\ Error = \frac{\text{Nilai Sensor} - \text{nilai } Tachometer}{\text{nilai } Tachometer}\ x\ 100 \qquad (6.6)$$

$$Persentase\ Error = \frac{39 - 39.5}{39.5}\ x\ 100$$

$$Persentase\ Error = 1.26\%$$

Setelah mendapatkan persentase error dari setiap data pada Tabel 6.4 maka dapat ditentukan nilai rata-rata persentase error sebesar 2.06%

6.2.4 Hasil dan Analisis Nilai *RPM* Sensor FC-03 (kanan)

Bacaan nilai *RPM* antara sistem *Monitoring RPM* dengan *tachometer* ditunjukan pada Gambar 6.4. Kedua nilai tersebut adalah data pertama yang ditampilkan pada Tabel 6.4.

Gambar 6.4 Bacaan Monitoring RPM roda kanan Sistem

Nilai *RPM* roda kiri didapat dengan membandingkan nilai *RPM* sistemdan *tachometer* berdasarkan 10 pengujian yang telah dilakukan. Data ditampilkan pada Tabel 6.4.

Tabel 6.4 RPM roda kanan Smart wheelchair

RPM Kanan Smart Wheelchair			
No	RPM bacaan Sistem (RPM)	RPM Bacaan Tachometer (RPM)	Persentase Error
1	58	55,7	4.12%
2	56	55,7	0.53%
3	59	59,3	0.50%
4	52	56,4	7.80%
5	56	56,4	0.70%
6	57	56,4	1.06%
7	54	53,7	0.55%
8	53	53,7	1.30%
9	56	57,8	3.11%
10	58	57,8	0.34%
Average Percent Error :			2.00%

Dengan menggunakan **Persamaan 6.7**, maka didapat nilai error dari setiap data yang telah diperoleh. Berikut adalah perhitungan persentase error menggunakan data terakhir Tabel 6.4.

$$Persentase\ Error = \frac{\text{Nilai Sensor} - \text{nilai } Tachometer}{\text{nilai } Tachometer}\ x\ 100 \qquad (6.7)$$

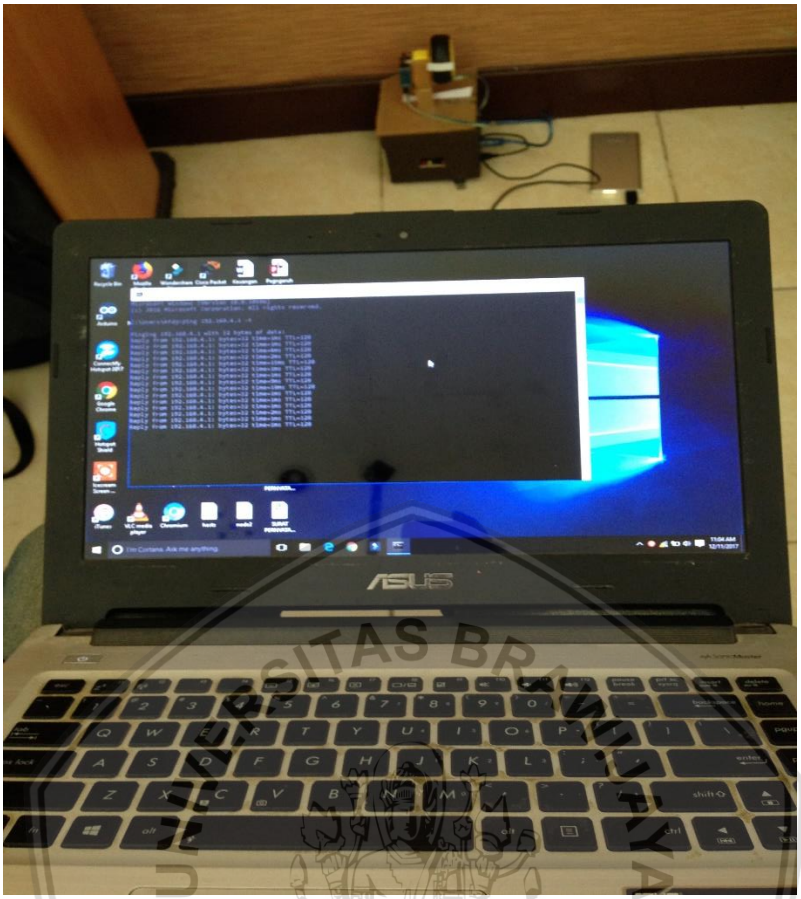
$$Persentase\ Error = \frac{58 - 57.8}{57.8}\ x\ 100$$

$$Persentase\ Error = 0.34\%$$

Setelah mendapatkan persentase error dari setiap data pada Tabel 6.4 maka dapat ditentukan nilai rata-rata persentase error sebesar 2.00%

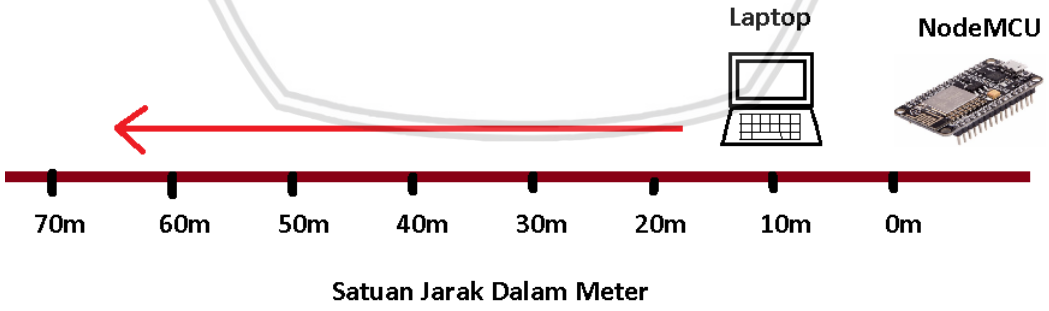
6.3 Pengujian Jarak *WiFi* NodeMCU

Sistem *monitoring RPM* menggunakan NodeMCU sebagai *access point* dan *web server*. Pengguna harus terhubung dengan *WiFi* yang dipancarkan oleh NodeMCU dan mengakses IP local dari NodeMCU, karena kegunaan NodeMCU sebagai *access point* dan *web server*, maka dilakukan pengujian terhadap jarak maksimum terhubungnya NodeMCU dengan perangkat lain saat melakukan transmisi data. Pengujian dilakukan dengan cara menempatkan prototipe pada 7 titik pengujian yaitu 10meter hingga 70meter, lalu laptop akan melakukan ping ke alamat lokal NodeMCU 192.168.4.1 dan menentukan pada jarak berapakah koneksi terputus seperti pada Gambar 6.5.



Gambar 6.5 Pengukuran WiFi NodeMCU

Jarak pengukuran *WiFi* NodeMCU ketika terhubung dengan sebuah laptop diilustrasikan pada Gambar 6.6. Pengukuran dimulai dari jarak 0meter hingga 70meter.



Gambar 6.6 Ilustrasi Titik Pengujian Jarak WiFi NodeMCU dengan Laptop

6.3.1 Tujuan Pengujian

Pengujian jarak *WiFi* dari NodeMCU dilakukan dengan tujuan mengetahui jarak maksimum suatu perangkat dapat terhubung dengan NodeMCU.

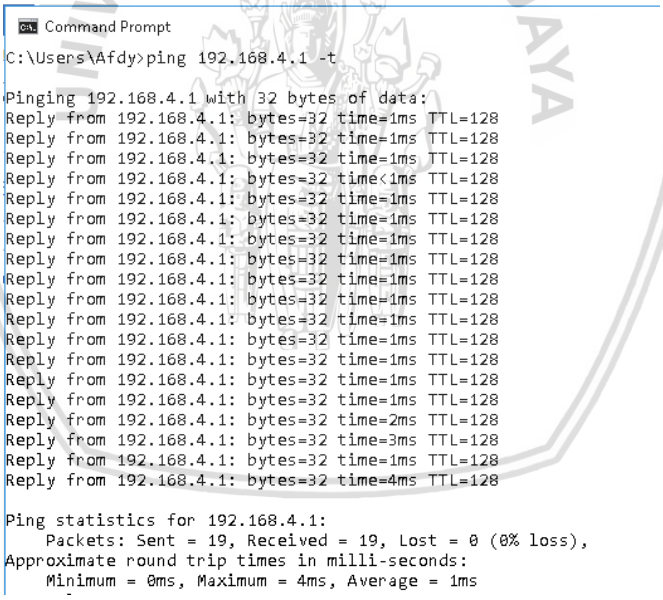
6.3.2 Prosedur Pengujian

Ada beberapa langkah dalam melakukan pengujian jarak *WiFi* dari NodeMCU, berikut adalah langkah-langkah yang dilakukan.

1. Menghubungkan NodeMCU dengan power bank.
2. Menghubungkan laptop dengan *WiFi* dari NodeMCU
3. Melakukan ping terhadap IP 192.168.4.1
4. Melakukan pengamatan pada titik 10meter hingga 70meter

6.3.3 Hasil dan Analisis

Pengujian *WiFi* NodeMCU saat terhubung dengan laptop menghasilkan beberapa data uji dengan cara melakukan ping dari laptop ke alamat local NodeMCU. Pengujian ping dilakukan seperti pada Gambar 6.7.



```
C:\Users\Afdy>ping 192.168.4.1 -t

Pinging 192.168.4.1 with 32 bytes of data:
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time<1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=2ms TTL=128
Reply from 192.168.4.1: bytes=32 time=3ms TTL=128
Reply from 192.168.4.1: bytes=32 time=1ms TTL=128
Reply from 192.168.4.1: bytes=32 time=4ms TTL=128

Ping statistics for 192.168.4.1:
    Packets: Sent = 19, Received = 19, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms
```

Gambar 6.7 Koneksi WiFi NodeMCU

Hasil dari pengujian dapat dilihat pada Tabel 6.5. Dapat dilihat bahwa koneksi antara laptop dengan NodeMCU tetap terhubung pada jarak 0meter hingga 50meter, tetapi pada jarak 60meter dan juga 70meter koneksi terputus.

Tabel 6.5 Pengujian Jarak WiFi NodeMCU

No	Jarak (meter)	Koneksi
1	0	Terhubung
2	10	Terhubung
3	20	Terhubung
4	30	Terhubung
5	40	Terhubung
6	50	Terhubung
7	60	Terputus
8	70	Teputus

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan rumusan masalah, pengujian beserta hasil analisis terhadap data uji didapat beberapa kesimpulan, berikut adalah kesimpulan yang dapat diambil.

1. Proses perancangan implementasi sistem *Monitoring RPM* pada *Smart Wheelchair* Berbasis Web Menggunakan *Sensor Optocoupler* adalah dengan cara menghubungkan dua buah sensor FC-03 dengan sebuah NodeMCU dan memasang *encoder disk* pada *gear* pada *smart wheelchair.* Setiap sensor FC-03 digunakan untuk mengakuisisi bacaan nilai hole, sehingga nodeMCU dapat menghitung nilai *RPM* dari roda kanan dan kiri pada *smart wheelchair.*
2. Berdasarkan analisis dari 50 pengujian dari masing-masing sensor FC-03 pada prototipe yang dilakukan dengan cara membandingkan nilai *RPM* kedua sensor dengan nilai *RPM tachometer* terhadap roda bertenaga motor DC 5V dengan 10 tingkat kecepatan berbeda, diperoleh rata-rata persentase error sebesar 0.92% (Sensor FC-03 sisi kanan) dan 0.87% (Sensor FC-03 sisi kiri).
3. Berdasarkan analisis dari 10 pengujian dari masing-masing sensor FC-03 yang telah diimplementasikan pada *smart wheelchair* dengan cara membandingkan nilai *RPM* kedua sensor dengan nilai *RPM tachometer* terhadap kedua buah roda pada *smart wheelchair* yang diputar secara manual, diperoleh rata-rata persentase error sebesar 2.0% (Sensor FC-03 sisi kanan) dan 2.06% (Sensor FC-03 sisi kiri).
4. Berdasarkan hasil dari pengujian jarak *WiFi* NodeMCU yang dihubungkan dengan sebuah laptop dapat disimpulkan bahwa Laptop masih tetap terhubung mulai dari jarak pengujian 0meter sampai dengan jarak 50meter. Pada jarak 60meter sampai dengan jarak 70meter koneksi antara laptop dengan NodeMCU terputus.

7.2 Saran

Ada pula saran yang bertujuan untuk mengembangkan penelitian ini, berikut beberapa saran yang diusulkan.

1. Menggunakan *encoder disk* dengan 100 *hole* sehingga dapat menghasilkan nilai persentase error yang lebih kecil.
2. *Encoder disk* ditempatkan 1 poros dengan roda pada *smart wheelchair* agar nilai *hole* yang terbaca untuk setiap putaran lebih akurat.

BAB 8 DAFTAR PUSTAKA

Ai-Thinker, 2015. *ESP-12E WiFi Module Version1.0,* s.l.: Ai-Thinker Team.

Ai-Thinker, 2015. *Schematics of Esp-12E WiFi Module.* [Art] (Ai-Thinker Team).

AI-Thinker, 2015. *Schematics of Esp-12E WiFi Module.* [Online] Available at: http://www.kloppenborg.net/images/blog/esp8266/esp8266-esp12e-specs.pdf [Diakses 13 September 2017].

Arabaci, H. & Bilgin, O., 2012. *A novel motor speed calculation method using square wave speed sensor signals via fast Fourier transform.*

Arduino, n.d. *Arduino Software (IDE).* [Online] Available at: https://www.arduino.cc/en/Guide/Environment [Diakses 8 Desember 2017].

Anirudh, T. S. & Satpathy, J. P., 2014. *DESIGN OF MOTORISED WHEELCHAIR.* [Online]Availableat: http://ethesis.nitrkl.ac.in[Accessed 24 Desember 2017].

Chen, Y., 2011. *Design and Developement on Intelligent Robot.* s.l.: LAP Lambert Academic Publishing.

Cook, D., 2009. *Robot Building for Beginners.* New York City: Apress.

Currey, M., 2015. *NodeMCU v1.0 Pin Mapping.* [Art].

Fitri Utaminingrum, H. F. et al., 2016. *Fast Obstacle Distance Estimation using Laser Line Imaging Technique for Smart Wheelchair.*

GodoyI, E. P. et al., 2010. Design and implementation of an electronic architecture for an agricultural mobile robot.

Holzner, S., 2009. *Ajax : A Beginner's Guide.* USA: The McGraw-Hill Companies.

Hughes, A., 1993. *ELECTRIC MOTORS AND DRIVES fundamentals,types and applications.* 2nd ed. Britain: s.n.

learn.acrobotic.com, n.d. *ESP-12E PINOUT.* [Art] (ACROBOTIC Industries).

Meyra, n.d. *iChair XXL 1.614.* [Online] Available at: https://www.meyra.com [Accessed 23 Desember 2017].

MISHRA, P., PRADHAN, S., SETHIYA, S. & CHAUDHARY, V., 2017. International Research Journal of Engineering and Technology (IRJET). *CONTACTLESS TACHOMETER WITH AUTO CUT OFF.*

Namaru, K. K., 2009. NON-CONTACT ROTATIONAL SPEED MEASUREMENT. pp. 7-8.

Nam, K. H., 2010. *AC Motor Control and Electrical Vehicle Applications.* Boca Raton: Taylor & Francis Group.

Perera, S. et al., 2012. Web Based Network Monitoring System.

Peter, S. et al., 2014. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. *Design of a Contactless tachometer.*

Sigaud, O. & Peters, J., 2010. *From Motor Learning to Interaction Learning in Robot.* Berlin: Springer-Verlag Berlin Heidelberg.

Systems, A. M. I., n.d. *AMS1117 1A LOW DROPOUT VOLTAGE REGULATOR,* s.l.: s.n.

Uday A. Bakshi, A. V. B. M. K. A. B., 2009. *Electrical Measurement and Measuring Instruments.* s.l.:Technical Publication Pune.

V2.0, A. R., 2017. *Main parts of the encoder.* [Art].

W. Pessen, D., 1989. *INDUSTRIAL AUTOMATION Circuit Design and Components.* New York: A Wiley-Interscience Publication.

Xie, M., 2003. *FUNDAMENTALS OF ROBOTICS LINKING PERCEPTION TO ACTION.* Singapore: World Scientific Publishing Co. Pte. Ltd..