

## **DOKUMEN TEKNIS PENGUJIAN DAN ANALISIS PRODUK**

### **Pendahuluan**

#### **1.1 Ringkasan Isi Dokumen**

Dokumen ini membahas tentang pengujian yang dilakukan untuk mengetahui ketercapaian fungsionalitas sistem terhadap spesifikasi yang dijanjikan dan verifikasi hasil implementasi berdasarkan desain rancangan sistem. Pengujian dilakukan untuk setiap subsistem dan sistem secara keseluruhan. Penjelasan pengujian meliputi bagaimana pengujian dilakukan, parameter yang menjadi fokus pengujian, hasil yang didapatkan dan analisis hasil pengujian yang didapatkan. Di bagian akhir dokumen ditambahkan rencana pengembangan produk selanjutnya berdasarkan analisis yang didapatkan.

#### **1.2 Tujuan Penulisan, Aplikasi dan Fungsi Dokumen**

Dokumen ini berlaku untuk pengembangan produk SPICAF untuk:

- 1) Dokumentasi proses pengujian dan verifikasi desain dan implementasi sistem dan subsistem dari produk SPICAF yang dikerjakan.
- 2) Penjabaran proses dan langkah pengujian tiap bagian, tahap per tahap, menyesuaikan dengan spesifikasi yang dijanjikan.
- 3) Menjadi acuan untuk penilaian ketercapaian spesifikasi yang dijanjikan, kesesuaian rancangan dan implementasi, dan ketercapaian penyelesaian masalah yang diangkat.
- 4) Pemenuhan komponen penilaian mata kuliah Skripsi di lingkup Program Studi Teknik Elektro.

## Pengujian dan Analisis Hasil Produk

### 2.6 Pengujian Subsistem *Bubble Sensor*

#### 2.1.1. Lingkup Pengujian

Batasan dan objektif pengujian keseluruhan subsistem:

1. Validasi hasil pembacaan *lux* dari LED yang dibaca sensor BH1750 saat selang berisi cairan.
2. Validasi hasil pembacaan *lux* dari LED yang dibaca sensor BH1750 saat selang berisi udara.
3. Penentuan konstanta dan selisih dari pembacaan sensor.

#### 2.1.2. Konfigurasi Pengujian

Perangkat keras dan lunak yang digunakan antara lain:

1. PC Intel Core i5 dengan OS Windows 10.
2. Arduino IDE v.1.8.18.
3. Arduino MEGA 2560.
4. BH1750 *library* v.1.3.0.
5. Sensor module BH1750.
6. LED putih 5 mm.
7. Kotak hitam akrilik.
8. Selang bening diameter 0,5 mm.
9. Resistor 1k $\Omega$ .

#### 2.1.3. Syarat Pengujian

1. Pengujian subsistem dilakukan secara *open loop*.
2. Pengujian dilakukan di ruang tertutup, ber-Air Conditioner, dan dinyalakan lampu. Tujuannya mensimulasikan ruangan pasien di rumah sakit.
3. *Baud rate* yang digunakan adalah 9600 bps.
4. Pengukuran dalam satuan *lux* minimal 0 *lux* dan maksimal 65535 *lux*.

5. Selang bening diasumsikan sebagai variabel bebas karena kedua faktor pembacaan menggunakan selang.

#### 2.1.4. Prosedur Pengujian dan Verifikasi

1. Pengukuran dilakukan dengan meng-*upload software* ke mikrokontroler setelah perangkat elektronik berhasil dihubungkan dan protokol komunikasi I2C berhasil di-*establish*.
2. LED dinyalakan secara konstan dengan tegangan 5V dari mikorkontroler.
3. Verifikasi isi suntik memang berisi cairan tanpa ada gelembung udara.
4. Pengumpulan data *lux* pada selang berisi cairan setiap 2 detik.
5. Verifikasi isi suntik tidak ada cairan.
6. Pengumpulan data *lux* pada selang berisi udara setiap 2 detik.
7. Perbandingan antara *lux* bercairan dan berudara.

Verifikasi isi memastikan keadaan suntik dan mendapatkan estimasi intensitas *lux*. Estimasi kemudian memiliki selisih yang cukup untuk membedakan antara bercairan dan berudara.

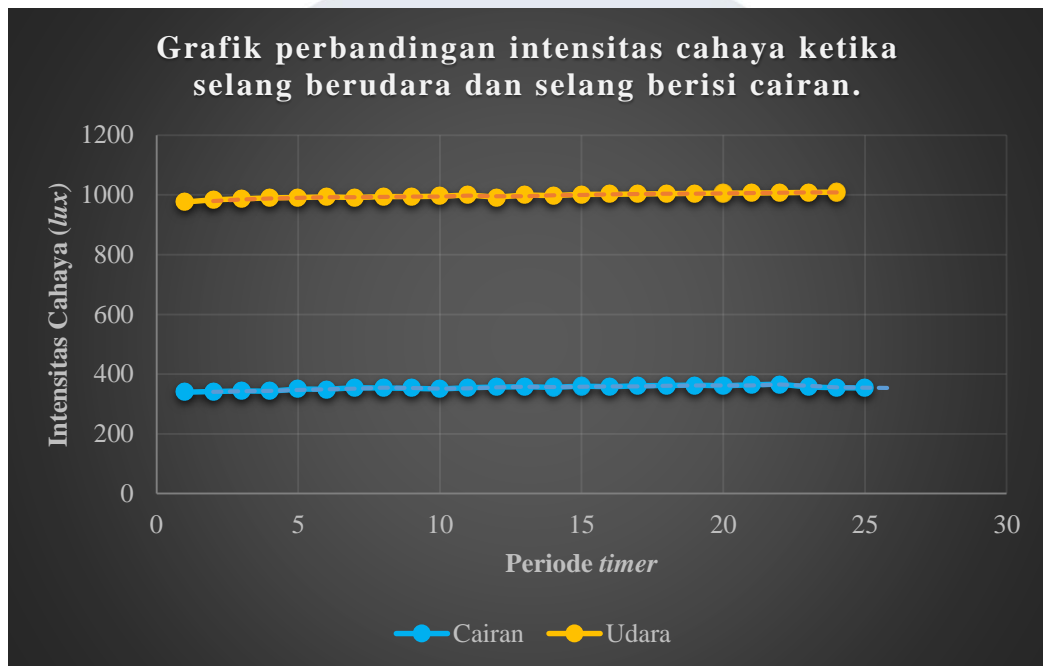


Gambar 5. 1 Kondisi selang berudara.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 2.1.5 Hasil Pengujian

Hasil pengujian subsistem adalah sebagai berikut:



Gambar 5. 2 Grafik perbandingan intensitas cahaya ketika selang berudara dan selang berisi cairan.



Gambar 5. 3 Konfigurasi ketika LED dinyalakan.

### 2.1.6 Analisis

Konsep pendeteksian gelembung dapat dilihat dari gambar berikut. Ketika diberikan cahaya, area yang bercairan cenderung lebih rapat dibandingkan area yang berudara, di mana pembiasannya tidak terhalangi oleh cairan.



Gambar 5. 4 Verifikasi kondisi selang dan beda bias cahaya pada saat bergelembung udara dan saat berisi cairan.

Pada beberapa percobaan sebelumnya dengan konfigurasi yang sama, data yang terbaca berbeda-beda pembacaannya tapi konstan. Dari tiap kali pembacaan memiliki suatu pola yang sama. Data relatif konstan dan naik secara perlahan dalam jangka waktu yang lama. Data intensitas cahaya dalam satuan *lux* menunjukkan pengukuran pada kondisi bercairan selalu lebih rendah dari kondisi berudara.

Dengan data di atas, didapatkan data kondisi berisi cairan memiliki rata-rata 354 *lux* dengan maksimum 366 *lux* sedangkan pada kondisi berisi udara memiliki rata-rata 998 *lux* dengan maksimum 1009 *lux*. Dengan demikian ditentukan konstanta pemisah yang akan dipakai adalah 400 *lux*. Di bawah 400 *lux* menandakan selang beroperasi normal yaitu berisi obat cair. Sedangkan di atas 400 *lux* menandakan ada kendala *bubbling* di dalam selang. Data relatif konstan dan tidak ada fluktuasi data untuk tiap kondisi sehingga data adalah akurat > 99% dan *error* pengukuran adalah < 1% mencakupi akurasi sensor.

Dapat diverifikasi dan sesuai dengan target fitur, subsistem *bubble sensor* berhasil mendeteksi perbedaan ketika selang berisi udara dan berisi cairan dengan cara membedakan intensitas cahaya dalam satuan *lux*. *Lux* di bawah 400 dinyatakan berisi cairan yang menandakan pengoperasian berjalan secara normal.

## 2.7 Pengujian Subsistem Pendorong Suntik

### 2.2.1 Lingkup Pengujian

Batasan dan objektif pengujian keseluruhan subsistem:

1. Validasi rpm *stepper motor* menggunakan pembacaan dari *digital rotary encoder* selama pengoperasian motor untuk mencapai target rpm.
2. Pengukuran rpm ketika menjalankan *softstarter* tanpa *load*.
3. Pengukuran rpm ketika menjalankan *softstarter* dengan *load*.
4. Validasi suntik berhasil terdorong.

### 2.2.2 Konfigurasi Pengujian

Perangkat keras dan lunak yang digunakan antara lain:

1. PC Intel Core i5 dengan OS Windows 10.
2. Arduino IDE v.1.8.18.
3. Arduino MEGA 2560.
4. *AccelStepper library* v.1.61.0.
5. *Digital Rotary Encoder* KY-040.
6. *Stepper Motor NEMA 17*.
7. *Stepper Motor Driver* A4988.
8. *Transmision Panel* dan *gearbox* dengan total *gear ratio* 1:562024.
9. *Rack and pinion*.
10. Suntik berukuran maksimum 60 mL.

### 2.2.3 Syarat Pengujian

1. Pengujian subsistem dilakukan secara *open loop*.
2. Pengujian dilakukan di ruang tertutup, ber-*Air Conditioner*, dan dinyalakan lampu. Tujuannya mensimulasikan ruangan pasien di rumah sakit.
3. *Baud rate* yang digunakan adalah 9600 bps.
4. Pengukuran dalam satuan rpm.
5. *Load* adalah suntik yang berisi air dalam kapasitas penuh.

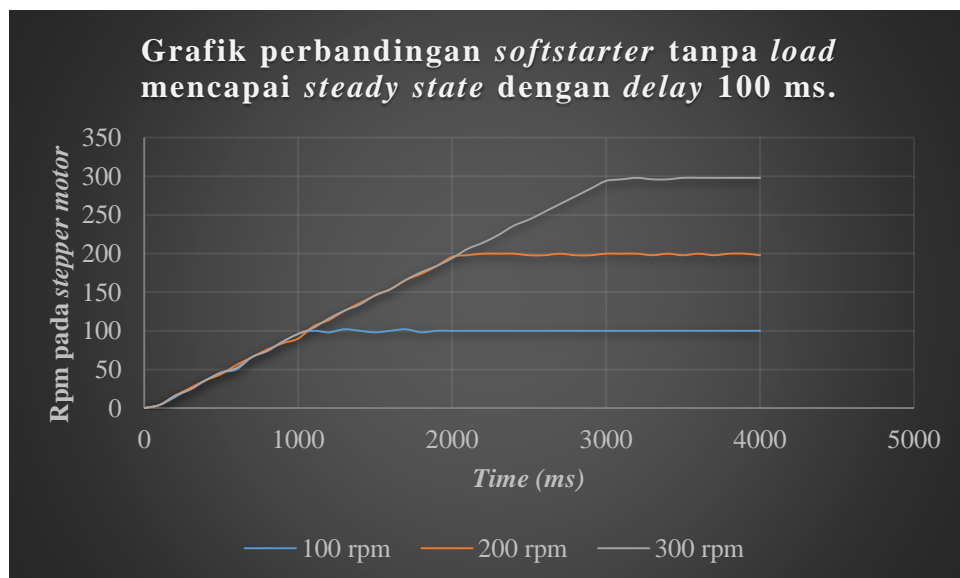
6. Tidak ada interferensi selain *load*.
7. *Driver* A4988 memiliki  $V_{ref}$  0,6 V dimana akan menarik arus sebesar 1,2 A.

#### 2.2.4 Prosedur Pengujian dan Verifikasi

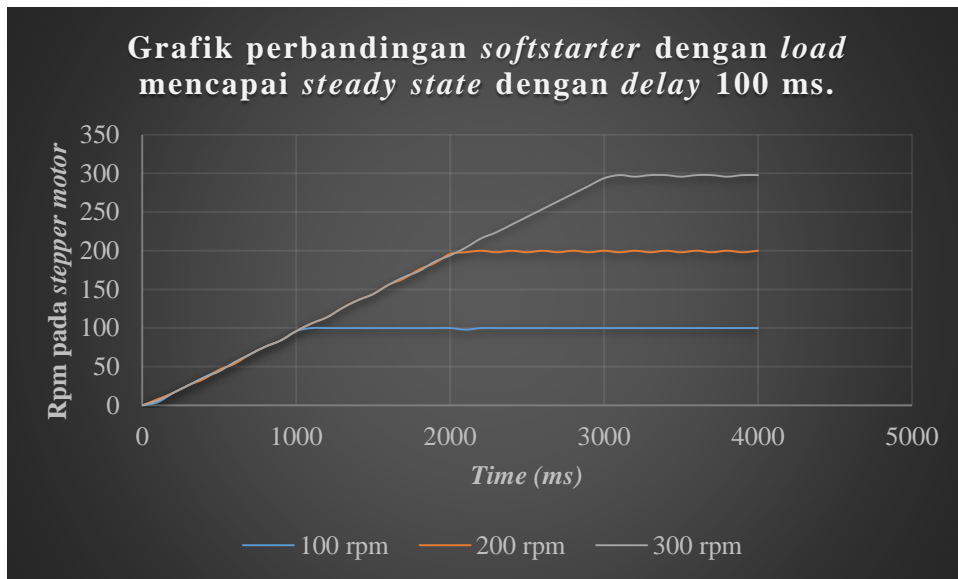
1. Pengukuran dilakukan dengan meng-*upload software* ke mikrokontroler setelah perangkat elektronik berhasil dihubungkan dan saklar sudah dinyalakan.
2. Menjalankan target rpm setiap pengukuran dan membaca rpm ketika *softstart*.
3. Membaca rpm ketika mencapai *steady state* dalam beberapa waktu.
4. Memasukkan *load* kemudian menjalankan prosedur 2 dan 3 kembali.
5. Memperhatikan pergerakan suntik dalam waktu yang lama.

#### 2.2.5 Hasil Pengujian

Berikut adalah grafik data hasil pengukuran *softstart* tanpa *load* dan dengan *load*.

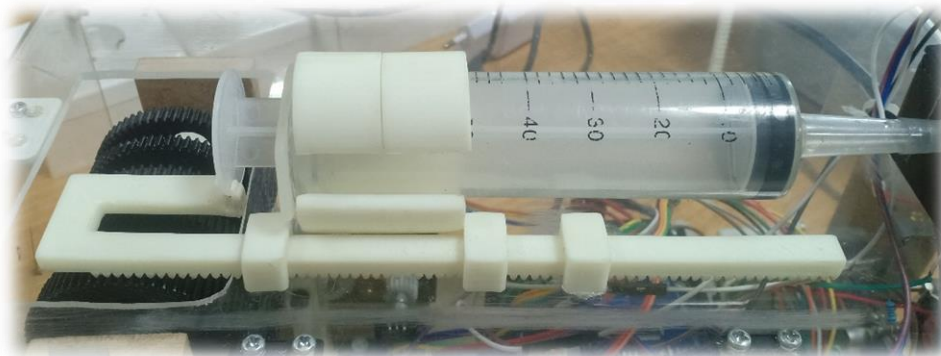


Gambar 5. 5 Grafik perbandingan *softstarter* tanpa *load* mencapai *steady state* dengan *delay* 100 ms.



Gambar 5. 6 Grafik perbandingan *softstarter* dengan *load* mencapai *steady state* dengan *delay* 100 ms.

Setelah mengukur rpm tiap *softstart* dan memperhatikan rpm selama *steady state*, diperhatikan apakah suntik berhasil didorong atau tidak.



Gambar 5. 7 *Rack and pinion* pada penhujung maksimum injeksi.

### 2.2.6 Analisis

Algoritma *softstarter* menggunakan fungsi *millis()* untuk menjalankan adisi setiap target rpm yang diinginkan yakni untuk setiap 100 ms. *Softstarter* dimulai dari 0 rpm kemudian diadisi terus menerus hingga mencapai target rpm dan *steady*



*state*. Pembacaan *encoder* pada *millis()* memiliki periode *delay* 1000 ms. Dengan adisi 100 ms pada *softstarter*, *encoder* masih sempat membaca rpm *stepper motor*. Bisa dilakukan adisi yang lebih cepat namun *encoder* tidak akan mampu membaca rpm dengan baik karena *delay* 1000 ms adalah 1/10 dari adisi *softstarter*. Selain perbedaan periode pembacaan, fitur *softstarter* berhasil diimplementasi 100%.

Pengukuran dilakukan dalam ranah 100 rpm, 200 rpm, dan 300 rpm. *Softstarter* 100 rpm dicapai paling cepat dan 300 rpm dicapai paling lama. Dapat dilihat pembacaan data rpm berselisih 0-3 nilai dari *true value* karena *encoder* memiliki *error* akurasi sebesar nilai demikian yang dijelaskan pada *datasheet*. Pembacaan rpm didapat dari pembacaan

$$rpm = encoder * 2$$

Dimana dikali 2 karena fasa perpindahan pada *encoder* terjadi selama 2 kali. Berdasarkan data pembacaan rpm, dapat dikatakan pembacaan rpm oleh *encoder* adalah berhasil dengan *error* 1-3% dari keterbatasan modul *digital rotary encoder*.

Untuk perbandingan antara *load* dan *no load*, relatif sama nilai pembacaan rpm. Hal ini berarti isi suntik tidak memengaruhi kecepatan putaran *stepper motor*. Faktor yang bisa membantu poin ini adalah penggunaan *gearbox* membantu menaikkan torsi gerak dorong yang didapat dari perubahan cepat rotasi menjadi torsi. Semakin tinggi *gear ratio* dan rpm maka akan semakin tinggi pula torsi dorong pada *load*. Sebagai verifikasi, gambar akhir menunjukkan suntik berhasil didorong oleh subsistem.

## 2.8 Pengujian Subsistem Antarmuka Pengguna (UI)

### 2.3.1 Lingkup Pengujian

Batasan dan objektif pengujian keseluruhan subsistem:

1. Validasi hasil masukkan data *keypad* sesuai dengan yang ditampilkan ke layar LCD.
2. Validasi data yang tidak termasuk dalam *range* operasi.

### 2.3.2 Konfigurasi Pengujian

Perangkat keras dan lunak yang digunakan antara lain:

1. PC Intel Core i5 dengan OS Windows 10.
2. Arduino IDE v.1.8.18.
3. Arduino MEGA 2560.
4. *LiquidCrystal library* v.1.0.7.
5. *LiquidCrystal\_I2C library* v.1.1.2.
6. *Keypad library* v.3.1.1.
7. *Wire.h*.
8. *Keypad matrix* 4x4.
9. LCD 16x2.
10. *I2C Serial Adapter*.

### 2.3.3 Syarat Pengujian

1. LCD 16x2 menggunakan *I2C serial adapter*.
2. Karena keterbatasan ukuran layar, perpindahan layar menggunakan *delay()* dan *lcd.clear()*.

### 2.3.4 Prosedur Pengujian dan Verifikasi

1. Pengujian dilakukan dengan meng-*upload software* ke mikrokontroler setelah perangkat elektronik berhasil dihubungkan dan saklar sudah dinyalakan.
2. LCD menampilkan arahan kepada pengguna.
3. *Keypad* dimasukkan *input* oleh pengguna dan ditunjukkan oleh LCD.
4. Jika *input* tidak dalam ranah, LCD menunjukkan tidak dapat memproses.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 2.3.5 Hasil Pengujian



Gambar 5. 8 Layar awal pemilihan ukuran suntik.



Gambar 5. 9 Pemilihan suntik 50 cc berhasil.



Gambar 5. 10 Arahan memasukkan *volumetric flowrate*.



Gambar 5. 11 Memasukkan *volumetric flowrate* dibawah spesifikasi.



Gambar 5. 12 Menekan tombol C untuk menghapus *input* dan mengisi ulang *volumetric flowrate*.



Gambar 5. 13 Masukkan *volumetric flowrate* dihapus.



Gambar 5. 14 Layar setelah *input* dihapus.



Gambar 5. 15 Masukkan kurang dari spesifikasi minimal, masukkan ditolak.



Gambar 5. 16 Masukkan yang memenuhi spesifikasi.



Gambar 5. 17 Masukkan berhasil diterima alat.

### 2.3.6 Analisis

Pengujian subsistem antarmuka teruji melalui hasil pengujian. Keypad mampu menerima semua masukkan tombol yang ditekan pengguna. LCD berhasil mengarahkan pengguna apa yang akan di-*input* selanjutnya, berhasil menunjukkan hasil masukkan seperti ukuran suntik, volume, dan *volumetric flowrate*, berhasil hanya memperbolehkan masukkan angka ke mikrokontroler dan layar, berhasil

menolak masukkan yang tidak sesuai dengan spesifikasi, berhasil menghapus masukkan dengan tombol 'C', dan berhasil menerima masukkan akhir untuk diolah selanjutnya oleh mikorkontroler dengan tombol 'D'. Keberhasilan pengujian alat adalah 100%.

## **2.9 Pengujian Notifikasi Pengguna (UN)**

### **2.4.1 Lingkup Pengujian**

Batasan dan objektif pengujian keseluruhan subsistem:

1. Validasi nilai kuat *decibel* dari bunyi *buzzer* di atas 45 dB.

### **2.4.2 Konfigurasi Pengujian**

Perangkat keras dan lunak yang digunakan antara lain:

1. PC Intel Core i5 dengan OS Windows 10.
2. Arduino IDE v.1.8.18.
3. Arduino MEGA 2560
4. *Passive buzzer module*.
5. *LiquidCrystal library* v.1.0.7.
6. *LiquidCrystal\_I2C library* v.1.1.2.
7. *Wire.h*.
8. LCD 16x2.
9. *I2C Serial Adapter*.
10. *App Spectroid v1.1.1*. di *Android*.

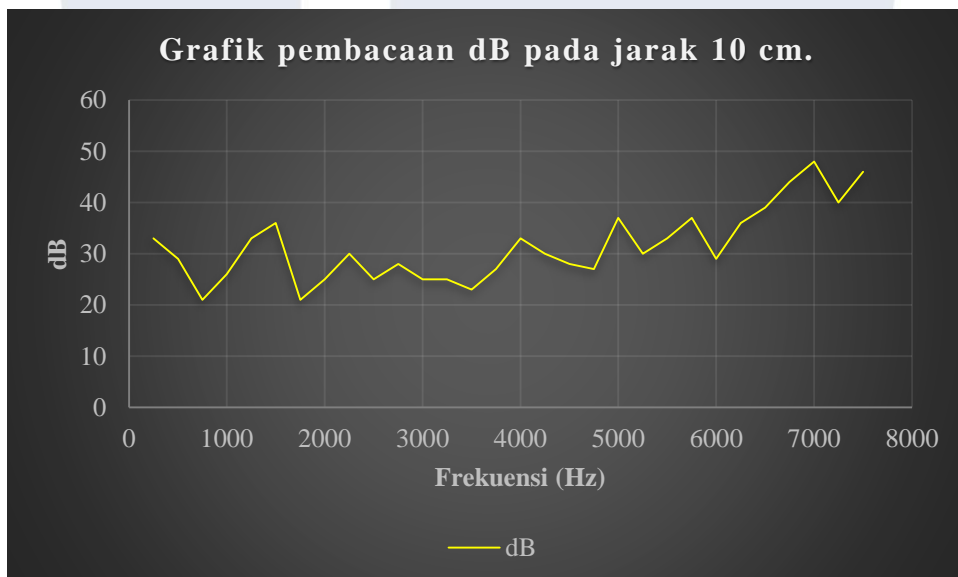
### **2.4.3 Syarat Pengujian**

1. Pengukuran menggunakan aplikasi *Spectroid*.
2. Handphone diletakkan pada jarak 10 cm dan 8 meter dari kepala *buzzer*. Delapan meter diasumsikan cukup untuk mensimulasikan di luar ruangan.

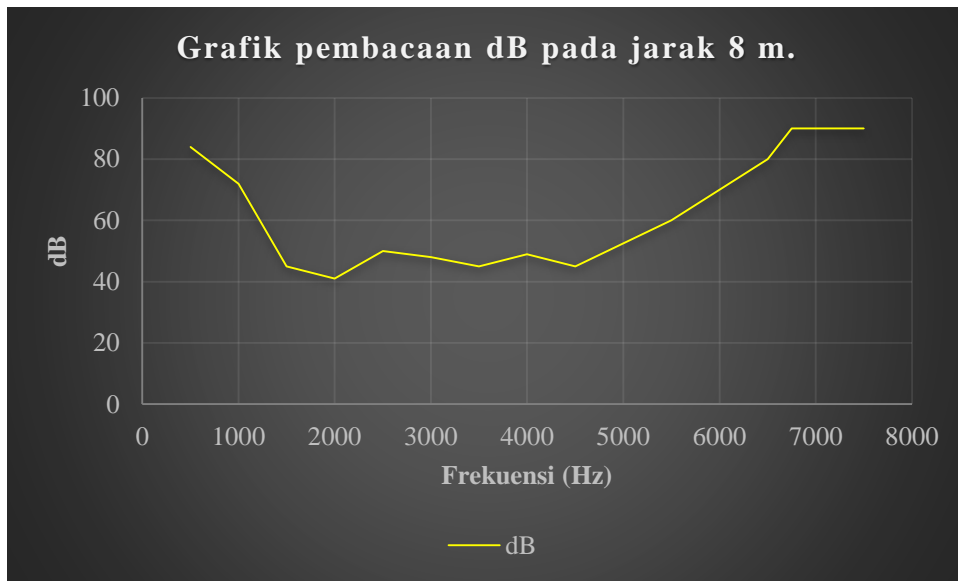
#### 2.4.4 Prosedur Pengujian dan Verifikasi

1. Pengukuran dilakukan dengan meng-*upload software* ke mikrokontroler setelah perangkat elektronik berhasil dihubungkan dan saklar sudah dinyalakan.
2. *Buzzer* dijalankan dengan secara bertahap menaikkan tinggi frekuensi menggunakan fungsi *millis()*.
3. *Spectroid* mengukur besar dB setiap kenaikan frekuensi pada *buzzer*.

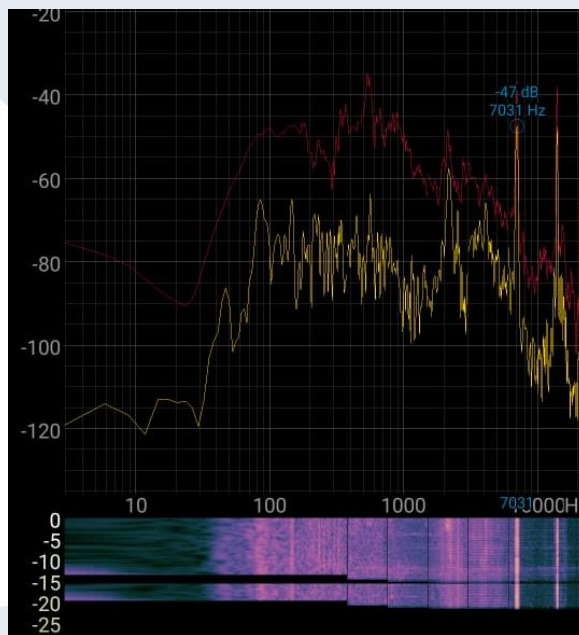
#### 2.4.5 Hasil Pengujian



Gambar 5. 18 Grafik pembacaan dB pada jarak 10 cm.



Gambar 5. 19 Grafik pembacaan dB pada jarak 8 m.



Gambar 5. 20 Hasil pembacaan pada *Spectroid* dengan frekuensi 7000 Hz.

#### 2.4.6 Analisis

Dari hasil pengukuran pada *Spectroid* dapat disimpulkan buzzer berbunyi diatas 45 dB ketika diberikan frekuensi minimal 7000 Hz di jarak 10 cm. Ketika 6750 Hz sudah mendekati 45 dB, namun supaya berjaga-jaga tidak kurang dan mempertimbangkan *error*, akan digunakan 7000 Hz. Pada jarak 8 m, bunyi dari



*buzzer* masih cukup terdengar, namun banyak dipengaruhi oleh suara lingkungan. Pada 1750 Hz, dB yang terbaca adalah 45 dB, namun pada frekuensi itu, jarak dekat tidak mencapai dB tersebut. 7000 Hz pada jarak 8 m mencapai 90 dB yang menjadi penentu utama apakah bisa terdengar sampai diluar sekaligus tetap memenuhi standar WHO. Dari pengujian ini, dapat diverifikasi lewat pengukuran pada *Spectroid* adalah relatif konstan di posisi yang sama. Maka, pengujian subsistem notifikasi pengguna lewat *buzzer* adalah berhasil diimplementasikan sekaligus memenuhi spesifikasi yang ditentukan WHO.

## **2.10 Pengujian Power Management Subsystem**

### **2.5.1 Lingkup Pengujian**

Batasan dan objektif pengujian keseluruhan subsistem:

1. PMS berhasil bertransisi dari daya *adaptor* ke daya baterai dan sebaliknya.

### **2.5.2 Konfigurasi Pengujian**

Perangkat keras dan lunak yang digunakan antara lain:

1. *Automatic Transfer Switch* dengan *threshold* 12 V.
2. Module *step-up* MT3608 dengan *MicroUSB port*.
3. *Adaptor* 18 watt 9 V 2 A dengan kepala *MicroUSB*.
4. Baterai 18650.
5. Module *step-up* XL6009.

### **2.5.3 Syarat Pengujian**

1. Baterai 18650 terdiri dari 2 buah diserikan, dari kapasitas 3,7 V menjadi 7,4 V.
2. Module *step-up* dari MT3608 an XL6009 dinaikkan hingga 12 V.

#### 2.5.4 Prosedur Pengujian dan Verifikasi

1. Pengujian dilakukan dengan memasang baterai 18650 ke *casing* dan mencolok *adaptor* ke colokan AC.
2. Ketika kondisi menyala dari *adaptor*, cabut *adaptor* agar daya ditarik dari baterai oleh ATS. Ketika kondisi menyala dari baterai, pasang *adaptor* agar daya ditarik oleh *adaptor* kembali.
3. Verifikasi dilakukan menggunakan *multimeter*.

#### 2.5.5 Hasil Pengujian



Gambar 5. 21 Tampak kondisi ATS menarik daya dari *adaptor*.



Gambar 5. 22 Tampak kondisi ATS menarik daya dari baterai.

<i>ATS truth table</i>	<i>Adaptor</i>	<i>Battery</i>
<i>AC on</i>	✓	✗
<i>AC off</i>	✗	✓

Tabel 5. 1 Tabel kebenaran pengujian ATS.

### 2.5.6 Analisis

Pengujian subsistem ATS berfungsi sesuai target fungsi dimana daya ditarik dari *adaptor* apabila ada dan ditarik dari baterai apabila daya dari adaptor tidak ada. Hal ini dapat diverifikasikan melalui tabel kebenaran pengujian serta kondisi ATS dalam pengujian.

Kendala yang terjadi pada saat transisi adalah saat berpindah dari daya adaptor ke daya baterai, mikrokontroler mati dan mereset sistem. Hal ini dikarenakan ATS menggunakan *relay* untuk melakukan fungsinya. Transisi yang dilakukan relay memiliki *delay* yang cukup untuk mikrokontroler menganggap tidak ada daya lagi. Solusi yang bisa diimplementasikan adalah pendekatan memberikan daya cepat dan singkat cukup untuk menutupi *delay*, yaitu menggunakan *super capacitor*.

## 2.11 Pengujian Subsistem Data Logging

### 2.6.1 Lingkup Pengujian

Batasan dan objektif pengujian keseluruhan subsistem:

1. Subsistem berhasil menyimpan data masukkan secara otomatis antara lain volume suntik, volume cairan, *volumetric flowrate*, dan total waktu per operasi.

### 2.6.2 Konfigurasi Pengujian

Perangkat keras dan lunak yang digunakan antara lain:

1. PC Intel Core i5 dengan OS Windows 10.
2. Arduino IDE v.1.8.18.
3. Arduino MEGA 2560.
4. *MicroSD Card Adapter Module*.
5. *MicroSD Card* dengan *storage* 4 GB.

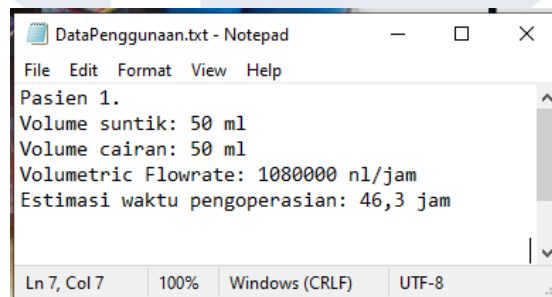
### 2.6.3 Syarat Pengujian

1. Penyimpanan dilakukan setelah semua *input* selesai dimasukkan.
2. Penyimpanan dilakukan setiap pembukaan sebuah *file* yang kemudian diisi dengan data. Setelah dibuka dan diisi data, *file* ditutup.

### 2.6.4 Prosedur Pengujian dan Verifikasi

1. Pengujian dilakukan setelah berhasil memasukkan data untuk mulai pengoperasian.
2. Setelah semua masukkan telah dikonfirmasi, mikrokontroler akan menyimpan data masukkan ke dalam *MicroSD Card*.

### 2.6.5 Hasil Pengujian



Gambar 5. 23 *File .txt* yang diisi data masukkan.

### 2.6.6 Analisis

Pengujian subsistem *data logging* berfungsi sesuai target fungsi dimana subsistem akan menyimpan data masukkan. Data ini biasanya dicatat secara manual oleh petugas kesehatan setiap kali mengoperasikan alat. Tahap fungsi ini adalah sebagai otomatisasi prosedur tersebut. Verifikasi dapat dilihat dari file yang dibuat dan terbukti berhasil.

## 2.12 Pengujian Sistem Keseluruhan

### 2.7.1 Lingkup Pengujian

Keseluruhan sistem akan diuji secara berurutan terutama fitur utamanya:

1. *Keypad* menerima masukkan dan menampilkannya ke layar LCD.
2. Masukkan diolah untuk mendapatkan data lain seperti penentuan lama waktu selesai dan *volumetric flowrate*.
3. Motor berputar dan berhasil mendorong suntik.
4. *Bubble sensor* mendeteksi adanya udara dalam selang yang membunyikan *buzzer*.
5. Sistem kembali menyala apabila tegangan *AC-adaptor* padam dengan menarik daya dari baterai.

### 2.7.2 Konfigurasi Pengujian

Perangkat keras dan lunak yang digunakan antara lain:

1. PC Intel Core i5 dengan OS Windows 10.
2. Arduino IDE v.1.8.18.
3. Arduino MEGA 2560.
4. BH1750 *library* v.1.3.0.
5. Sensor module BH1750.
6. LED putih 5 mm.
7. Kotak hitam akrilik.
8. Selang bening diameter 0,5 mm.
9. Resistor 1k $\Omega$ .
10. *AccelStepper library* v.1.61.0.
11. *Digital Rotary Encoder* KY-040.
12. *Stepper Motor* NEMA 17.
13. *Stepper Motor Driver* A4988.
14. *Transmision Panel* dan *gearbox* dengan total *gear ratio* 1:562024.
15. *Rack and pinion*.
16. Suntik berukuran maksimum 60 mL.

17. *LiquidCrystal library* v.1.0.7.
18. *LiquidCrystal\_I2C library* v.1.1.2.
19. *Keypad library* v.3.1.1.
20. *Wire.h*.
21. *Keypad matrix* 4x4.
22. LCD 16x2.
23. *I2C Serial Adapter*.
24. *Passive buzzer module*.
25. *Automatic Transfer Switch* dengan *threshold* 12 V.
26. Module *step-up* MT3608 dengan *MicroUSB port*.
27. *Adaptor* 18 watt 9 V 2 A dengan kepala *MicroUSB*.
28. Baterai 18650.
29. Module *step-up* XL6009.
30. *MicroSD Card Adapter Module*.
31. *MicroSD Card* dengan *storage* 4 GB.

### 2.7.3 Syarat Pengujian

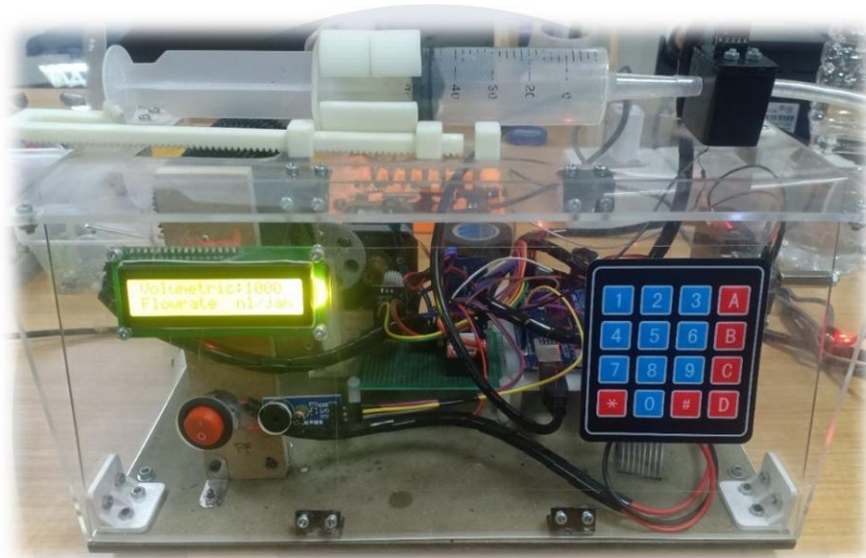
1. Pengintegrasian seluruh subsistem telah berhasil baik dalam perangkat keras, perangkat elektronik, dan perangkat lunak.

### 2.7.4 Prosedur Pengujian dan Verifikasi

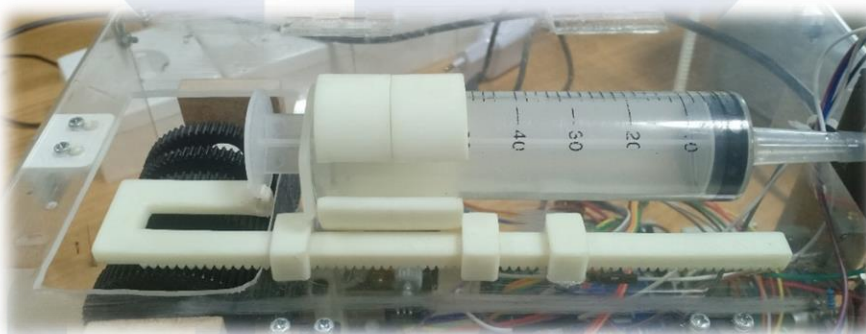
1. Pengguna memasukkan data ukuran volume suntik, isi volume cairan dan *volumetric flowrate* yang diperlukan.
2. Sistem akan berjalan untuk mendorong suntik.
3. Apabila terjadi kendala, *buzzer* akan berbunyi dan LCD akan menampilkannya.

Pengujian diverifikasi *threshold*-nya dengan melihat hasil akhir pendorongan suntik dan apabila terjadi kendala, *buzzer* akan bunyi.

### 2.7.5 Hasil Pengujian



Gambar 5. 24 Pengujian keseluruhan sistem.



Gambar 5. 25 Tampak selesai melakukan injeksi.

### 2.7.6 Analisis

Pengujian keseluruhan sistem adalah integrasi dari semua sistem yang telah dikembangkan dan diuji keberhasilannya. Sistem berhasil melakukan menerima dan menampilkan masukkan dari pengguna, melakukan injeksi sesuai dengan target waktu, dan menyimpan data penting pengoperasian. Ketika terjadi kendala, *buzzer* akan berbunyi dan LCD akan menampilkan kendala tersebut, yaitu pendeteksian udara dalam selang.

## Kesimpulan dan Saran Pengembangan Produk

### 3.1 Kesimpulan Hasil Pengembangan Produk

Dalam dunia kesehatan, diperlukan alat injeksi yang dapat mentransmisikan fluida dalam jangka waktu yang sangat lama. Alat itu adalah *syringe pump*. *Syringe pump* sering digunakan untuk *home care*/rawat jalan, injeksi obat dosis tinggi, dan operasi. Maraknya kasus COVID-19 juga menyebabkan kebutuhan untuk alat ini meningkat.

Idealnya selama injeksi, penyaluran harus hanya cairan yang masuk ke dalam tubuh pasien. Pada praktiknya sering kali terjadi penggelembungan di dalam suntik yang sangat membahayakan pasien, dimana dapat menyebabkan *venous air embolism* yaitu masuknya gelembung udara ke dalam peredaran darah dapat menyebabkan penyumbatan ke jaringan tubuh yang disaluri darah, terutama otak, jantung, dan paru-paru. Kasus terburuknya dapat menyebabkan kematian jika tidak ditindaklanjuti.

Dari permasalahan tersebutlah dikembangkan alat *syringe pump* yang bisa mendeteksi keberadaan udara selama injeksi/transmisi obat cairan. Dengan demikian risiko fatalitas pengobatan menjadi berkurang drastis. Alat yang dikembangkan dinamakan SPICAF.

Pengembangan SPICAF berhasil mendeteksi keberadaan udara dengan menggunakan *bubble sensor*, mendorong suntik dalam jangka waktu yang lama menggunakan *gearbox*, menyimpan data masukkan yang biasa ditulis secara manual oleh petugas secara otomatis, dan fleksibel terhadap keperluan kecepatan injeksi.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



### 3.2 Saran Pengembangan Produk

SPICAF bisa dikembangkan lebih lanjut dengan menambahkan fitur yang bisa memindahkan *rack and pinion* ke baris lain pada *gearbox*. Dengan demikian SPICAF dapat mengubah *range* kerja injeksinya. Fitur ini menyerupai *gearshifting*.

Selain *gearshift*, SPICAF juga dapat dikembangkan pada bagian penyimpanan untuk menggunakan internet modul ESP8266. Penyimpanan melalui internet bisa langsung disimpan pada *database* pengguna, misalnya pihak rumah sakit.

Data yang disimpan dalam *MicroSD Card* ada baiknya juga dapat dilihat oleh petugas tanpa harus mencabut *MicroSD Card* dengan menampilkannya ke LCD ketika tidak beroperasi.

Untuk perbaikan sistem, sebaiknya menggunakan *stepper motor* dengan torsi dan rpm yang lebih tinggi untuk memudahkan pemutaran pada *gearbox* dan meluaskan ranah kecepatan perputaran sehingga bisa melakukan injeksi lebih cepat jika diperlukan.

Penempatan *stepper motor* dan *gearbox* bisa lebih fleksibel jika menggunakan railing sehingga lebih mudah pemasangannya dan bisa diintegrasikan ke fitur *gearshifting*.