RANCANG BANGUN PEMBUAT KOKTAIL BERBASIS RASPBERRY PI DENGAN KONTROL PERINTAH SUARA

Putu Bagus A. A. Sanjaya
1512201
sanjayaputu90@gmail.com

M. Ibrahim Ashari, ST., M
Pembimbing 1

Dr. Eng I Komang Somawirata, ST., MT.
Pembimbing 2

Abstract— .*Penggunaan teknologi dalam bidang pramusaji sangat sering digunakan baik untuk kepentingan manajemen maupun demi kepentingan kepuasan pelanggan. Salah satu contohnya adalah dalam bidang bartender atau pramutama bar. Bartender atau pramutama bar adalah seorang yang ahli dalam meracik dan menyajikan minuman beralkohol. Pada umumnya bartender akan meracik minuman yang pelanggan pesan dan menyajikannya dengan cara yang menarik serta mampu dalam berinteraksi, untuk itu pelanggan akan memberikan bayaran tambahan untuk servis sang bartender disamping menu yang telah dipesan. Berdasarkan permasalahan tersebut, terbentuk sebuah pemikiran untuk membuat mesin yang mampu meracik minuman koktail dan mampu berinteraksi serta menghibur pengguna tanpa perlu jauh-jauh pergi ke kelab malam untuk menikmati koktail. Maka dari itu tercipta rancang bangun mesin pembuat koktail yang berbasis Raspberry Pi dengan kontrol perintah suara. Dengan mesin ini pengguna cukup meletakkan gelas berisi es dan memberikan perintah kepada mesin dan mesin akan otomatis meracik minuman yang diinginkan. Berdasarkan hasil analisa dan pengujian yang dilakukan, mesin ini mampu meracik minuman tanpa harus memijat tombol serta bersifat portable.*

Kata Kunci— mesin pembuat koktail, smart batender, raspberry pi, kontrol perintah suara, google assistant;

I. PENDAHULUAN

A. Latar Belakang

Pramutama bar (*bartender, barman, barkeep, barmaid, barkeeper, barperson, bar attendant*) adalah orang yang meracik dan menyajikan minuman beralkohol untuk tamu di konter bar. Minuman beralkohol yang disajikan antara lain koktail, anggur, likeur, dan bir. Tempat-tempat minum seperti *pub, tavern*, dan kelab malam juga sering mempekerjakan pramutama bar. Seorang pramutama bar diharapkan mampu menyiapkan minuman koktail klasik, misalnya: *Cosmopolitan, Manhattan, Old Fashioned*, dan *Mojito*. Sebelum menyajikan minuman, ia juga perlu memastikan tamu yang disajikan minuman olehnya sudah cukup usia secara hukum untuk mengonsumsi minuman beralkohol. Ia juga menjaga agar tamu tidak mabuk, dan menolak menyajikan minuman untuk tamu yang telah mengonsumsi terlalu banyak alkohol.

Koktail (bahasa Inggris: *cocktail*) adalah minuman beralkohol yang dicampur dengan minuman atau bahan-bahan lain yang beraroma. Di zaman sekarang, koktail biasanya dibuat dari satu jenis atau lebih minuman beralkohol ditambah perisa berupa *liqueur*, sari buah, gula, madu, air, es, minuman karbonasi, susu, krim susu, rempah-rempah, *bitters*, dan sebagainya. Namun untuk menikmati minuman ini, penikmat harus merogoh isi kantong mereka cukup dalam. Dimana mereka harus membayar *tax service* yang cukup mahal dan apabila *bartender* mampu melayani penikmat dengan baik, mau tidak mau penikmat harus memberikan tip. Terlebih penikmat harus keluar rumah dan mencari *bar, resto, tavern* atau *pub* yang memiliki *bartender* handal sesuai selera penikmat.

Dengan melihat keterbatasan ini, maka salah satu solusi untuk mengatasi keterbatasan tersebut diperlukan sebuah alat atau mesin yang dapat berinteraksi dan menyajikan sebuah koktail yang tanpa harus penikmat meninggalkan rumah/tempat kerjanya dan presisi dalam meramu minuman tersebut. Dengan mesin ini pengguna cukup mengatakan jenis koktail yang diinginkan, maka mesin ini akan mencampurkan dan menuangkan minuman sesuai menu yang dipilih. Pengguna hanya butuh membawa gelas khusus koktail yang diisi es dan meletakkannya pada tempat yang telah disediakan.

Penelitian ini diambil dari referensi jurnal [Siregar E. A., Purwantoro S., Trisnadoli A., 2014], dimana peneliti tersebut menggunakan konveyor sebagai sistem pencampur dan android sebagai sistem kontrolnya. Ada pula penelitian yang digunakan sebagai referensi tambahan oleh peneliti, yaitu rancang bangun alat yang menggunakan PLC sebagai pencampur bahan minuman bersoda berdasarkan kadar keasaman [Kusumawardani R., Sumiharto R., 2015]. Dengan melihat hal tersebut, menurut peneliti menggunakan pompa peristaltik dan sistem perintah suara sebagai kontrol dirasa memberikan pelayanan lebih cepat. Selain pengguna dapat berinteraksi langsung via suara dengan alat, juga lebih mudah dioperasikan dibandingkan dengan kedua referensi tersebut.

Maka dari itu, peneliti melakukan pengembangan dengan membuat alat pencampur minuman koktail berbasis raspberry pi karena dirasa lebih modern karena pin-pin dari raspberry sudah digital serta menggunakan pompa peristaltik dibandingkan dengan konveyor atau dispenser. Rancang bangun ini mengembangkan alat dari referensi [Hacker Shack, 2018], yaitu Smart

bartender lalu ditambahkan control perintah suara oleh peneliti sebagai pengembangannya guna menjadikan alat ini seperti memiliki bartender pribadi di rumah atau tempat dimanapun pengguna berada.

B. Rumusan Masalah

Berdasarkan paparan latar belakang di atas maka dapat dirumuskan masalah sebagai berikut :

1. Bagaimana merancang perangkat pencampur minuman koktail berbasis raspberry dengan kontrol perintah suara?
2. Bagaimana merancang proses pencampuran minuman secara otomatis sesuai menu yang dipilih?
3. Bagaimana mengintegrasikan Raspberry pi dengan Google Assistant?
4. Bagaimana merancang program untuk resep tiap menu?

C. Tujuan

Merancang dan membuat perangkat *Pencampur Minuman Koktail Berbasis Raspberry Pi dengan Kontrol Perintah Suara* yang mampu menyajikan minuman dengan presisi pencampuran yang baik. Manfaat dari alat ini agar dapat mempermudah dalam proses pelayanan, sehingga penikmat tidak perlu membayar seorang *bartender* dan memudahkan dalam pencampuran minuman serta dapat berinteraksi layaknya memiliki seorang *bartender* pribadi di rumah.

II. TINJAUAN PUSTAKA

A. Mikrokontroler

Dalam proses pembuatan alat elektronik yang dapat memindah sumber listrik otomatis ini menggunakan mikrokontroller Raspberry Pi. Raspberry Pi adalah komputer mikro berukuran seperti kartu kredit yang dikembangkan oleh Raspberry Pi Foundation, Inggris. Komputer single board ini dikembangkan dengan tujuan untuk dapat mengajarkan dasar-dasar ilmu komputer dan pemrograman untuk siswa sekolah di seluruh dunia. Meskipun mikrokontroler ini yang memiliki fisik seperti Arduino, yang dimana lebih dikenal untuk proyek-proyek prototyping juga, tidak demikian dengan Raspberry Pi yang sangat-sangat berbeda dari mikrokontroler lainnya kebanyakan, dan sebenarnya, lebih seperti komputer daripada Arduino.

B. Sound Sensor

Sensor suara (*Sound Sensor*) merupakan modul sensor yang mengubah besaran suara menjadi besaran listrik. Sensor suara ini memiliki catu daya sebesar 3,3 sampai 5 VDC [Lapono L. A. S., Pingak R. K., 2018]. Module ini bekerja berdasarkan prinsip kekuatan gelombang suara yang masuk. Dimana gelombang suara tersebut mengenai membran sensor, yang berefek pada bergetarnya membran sensor. Dan pada membran tersebut terdapat kumparan kecil yang dapat menghasilkan besaran listrik. Kecepatan gerak kumparan ini menentukan kuat lemahnya gelombang listrik yang dihasilkan. Sensor ini memiliki kemampuan deteksi suara 48dB hingga 52dB. Sensor ini memiliki digital output untuk dapat mendeteksi ada atau tidaknya suara di dalam suatu ruangan. [Lukman M. P., Junaedy, Rieuwpassa Y. F. Y., 2018].

C. Google Assistant

Google Assistant adalah asisten virtual yang didukung oleh kecerdasan buatan dan dikembangkan oleh Google yang terutama tersedia di perangkat seluler dan perangkat rumah pintar. Tidak seperti Google Now, Google Assistant dapat terlibat dalam percakapan dua arah. Google Assistant awalnya memulai debutnya pada bulan Mei 2016 sebagai bagian dari aplikasi perpesanan Google Allo, dan pembicara yang diaktifkan suara Google Home. Setelah periode eksklusif pada telefon pintar Pixel dan Pixel XL, lalu mulai dikerahkan di perangkat Android lainnya pada bulan Februari 2017, termasuk telefon pintar pihak ketiga dan Android Wear, dan dirilis sebagai aplikasi yang berdiri sendiri pada operasi iOS sistem pada bulan Mei. Di samping pengumuman perangkat pengembangan perangkat lunak pada bulan April 2017, Google Assistant telah, dan sedang, diperluas untuk mendukung berbagai macam perangkat, termasuk mobil dan peralatan rumah pintar. Fungsi Google Assistant juga dapat disempurnakan oleh pengembang pihak ketiga [Google, 2016].

D. Peristaltic Pump

Peristaltic pump beroperasi dengan memungkinkan cairan manuju ke selang. Cairan ini kemudian mengalir ke dalam casing pompa melalui selang. Sekali di sana, baling-baling dengan sejumlah kompres pengait tabung memaksa cairan pada melalui pompa dan mengarahkannya ke tujuan akhir. Teknik ini dikenal sebagai peristaltik. Dengan demikian, alat ini disebut peristaltic pump. Sering kali, ketika menggunakan peristaltic pump, cairan harus disimpan dalam lingkungan yang steril. Oleh karena itu, selang selalu harus berisi cairan sama. Namun, ini tidak selalu membatasi fungsi pompa, karena selang dapat beralih keluar. Hal ini sering dilakukan, misalnya, dalam kasus dimana ada cairan lain yang sedang dipompa. Pompa juga mempunyai kelebihan lain. Sebagai contoh, karena bagian dalam peristaltic pump selalu kering, jadi tidak perlu khawatir karena harus melindungi pompa dari kelembapan. Kelembapan di dalam tabung akan tetap terjaga [Subroto, Lusmiyanto B., Rahardjo T., 2012]. Variasi diameter roller 6 mm, 8 mm, 10 mm, 12 mm, 14 mm. Hasil penelitian menunjukan bahwa variasi jumlah 3 roller dengan diameter 6 mm menghasilkan debit paling optimal yaitu 354,1 ml/menit. Semakin kecil diameter yang digunakan pada mekanisme maka akan semakin menambah debit yang dihasilkan [Maryanto L. E., Basyirun, Anis S., 2018].

Gambar 1. Peristaltic Pump

Adapun rumus yang dapat digunakan untuk menentukan yang berhubungan dengan fluida, waktu atau volumenya, sebagai berikut:

$$Q = \frac{V}{t} \text{ atau } Q = Av$$

Q	=	Debit aliran fluida (m^3/detik)
V	=	Volume fluida (m^3)
t	=	Waktu (detik)
v	=	Kecepatan aliran fluida (m/detik)
A	=	Luas penampang aliran (m^2)

E. OLED Display

OLED merupakan perangkat solid state atau perangkat elektronik yang umumnya terdiri atas film tipis organik yang diapit di antara dua elektroda konduktif film tipis. Ketika arus listrik diterapkan, cahaya terang dipancarkan. OLED juga menggunakan molekul desainer berbasis karbon yang dapat memancarkan cahaya ketika arus listrik melewatinya. Ini juga bisa disebut electro phosphorescence. Bahkan dengan sistem berlapis, sistem ini sangat tipis. Biasanya bisa kurang dari 500 nm atau sekitar 200 kali lebih kecil dari rambut manusia. Pada saat digunakan untuk menghasilkan display, teknologi OLED menghasilkan tampilan bercahaya sendiri yang tidak memerlukan cahaya latar dan karenanya mampu lebih hemat energi. Properti ini dapat menghasilkan tampilan yang lebih tipis dan sangat kompak. Layar hanya membutuhkan daya yang sangat kecil, yaitu hanya 2-10 volt. Teknologi OLED juga menggunakan zat yang memancarkan cahaya merah, hijau, biru atau putih. Tanpa sumber penerangan lain, bahan OLED bisa menghadirkan video dan gambar yang terang serta jernih yang mudah dilihat di hampir semua sudut. Meningkatkan material organik membantu mengontrol kecerahan dan warna cahaya pada OLED tersebut [Patel B. N., Prajapati M. M., 2014]. Namun pada penelitian ini, peneliti menggunakan *Miniatur OLED Display*. Yang modulnya lebih kecil, simple dan sangat baik digunakan.

F. Relay

Relay dapat berfungsi sebagai saklar otomatis. Prinsip kerjanya adalah elektromagnetik untuk merubah kondisi saklar yang dapat menghantarkan arus listrik dengan tegangan yang lebih tinggi. (Sadewo, Widasari, & Muttaqin, 2017)

Ada dua macam jenis relay yaitu:

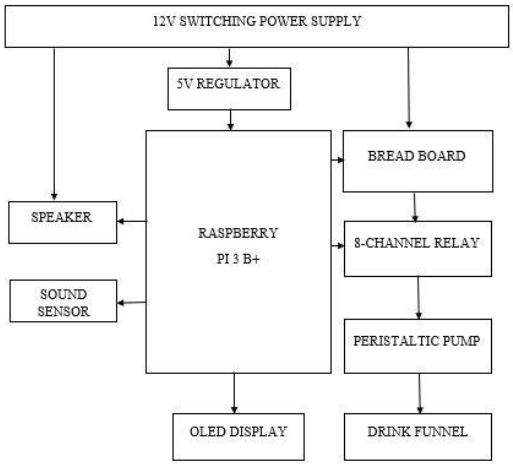
- *Normally Close* (NC), kondisi awaI saklar selalu berada pada posisi tertutup (*close*).
- *Normally Open* (NO), kondisi awaI saklar seIaIu berada pada posisi terbuka (*open*).

Papan antarmuka *Relay* 8-Saluran 5V, Dapat mengontrol berbagai peralatan dengan arus besar, dapat dikontrol langsung oleh *Microcontroller* (*Raspberry Pi*, *Arduino*, 8051, AVR, PIC, DSP, ARM, ARM, MSP430, logika TTL). *Relay* adalah saklar yang dioperasikan secara listrik. Banyak *relay* menggunakan elektromagnet untuk mengoperasikan saklar secara mekanis dan menyediakan isolasi listrik di antara dua sirkuit.

Berikut merupakan beberapa fungsi Relay yang telah umum diaplikasikan kedalam peralatan Elektronika yang sudah biasa digunakan diantaranya adalah:

1. Relay yang dapat digunakan untuk menjalankan fungsi logika (*Logic Function*)
2. Relay yang biasa digunakan untuk memberikan fungsi penundaan waktu (*Time Delay Function*)
3. Relay juga dapat digunakan untuk mengendalikan sirkuit tegangan tinggi dengan bantuan dari sinyal tegangan rendah.
4. Ada juga fungsi Relay yang digunakan untuk melindungi motor ataupun komponen lainnya dari kelebihan tegangan ataupun hubung singkat (*Short*)

G. Block Diagrams



Gambar 2. Block Diagrams

Prinsip Kerja Alat :

1. 12V *switching power supply* sebagai sumber tenaga yang akan memberikan energi pada tiap instrumen yang membutuhkan pada rancang bangun alat
2. *Bread board* sebagai pembagi energi ataupun sebagai terminal untuk menghubungkan tiap komponen
3. 5V *regulator* adalah instrumen yang akan merubah (*stepped down*) tegangan dari *power supply* agar mikrokontroler mendapatkan daya sesuai spesifikasi
4. *Speaker* sebagai *feedback* dari *Google Assistant* dan telinga yang mendengar pertanyaan dari pengguna alat yang akan menjadi inputan untuk diproses oleh mikrokontroler
5. Raspberry pi 3 B+ sebagai otak pada rancang bangun yang akan mengatur segala perintah pada alat dan seagai pemrosesan input perintah
6. *Sound sensor* sebagai perekam perintah-perintah untuk mengupgrade *Google Assistant* yang akan digunakan sebagai bahan inputan
7. *8-channel relay* sebagai pemicu pompa-pompa yang akan bekerja sesuai perintah yang diberikan mikrokontroler
8. Pompa peristaltik adalah pompa yang membawa minuman dari botol sampai ke gelas, dan bekerja sesuai resep koktail yang diinginkan
9. *OLED display* akan menampilkan resep/minuman/menu yang pengguna inginkan
10. *Drink Funnel* adalah tempat semua minuman bercampur dan keluar dari pipa ke gelas

III. METODOLOGI

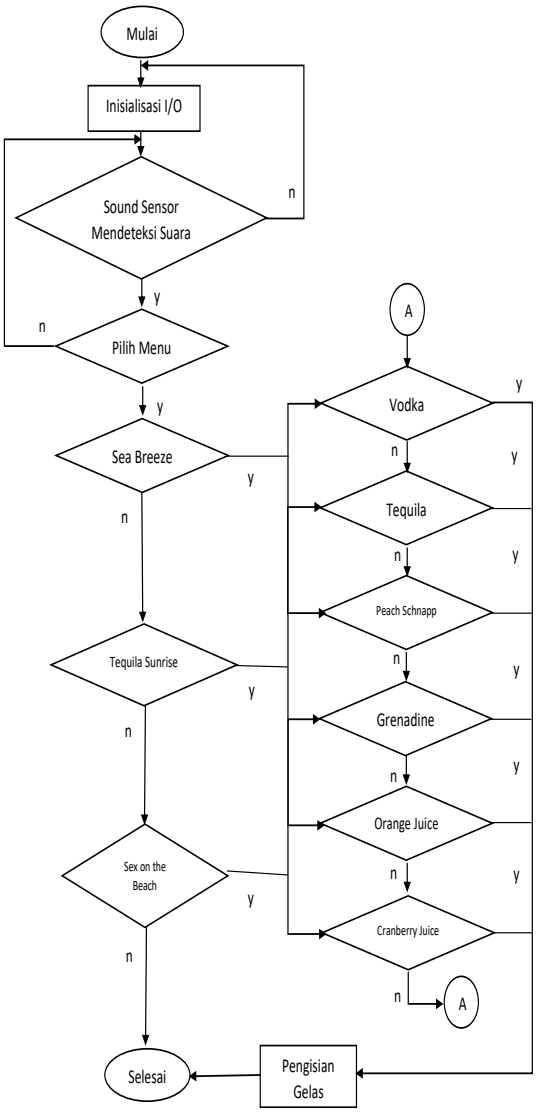
A. *Perancangan*

Dalam perancangan ini metodologi dapat dibagi menjadi dua, yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). Ada beberapa komponen utama yang terdapat pada alat ini, yaitu:

a) *Hardware*:
   1. *Raspberry Pi 3*
   2. *Sound Sensor*
   3. *Speaker*
   4. *Peristaltic Pump*
   5. *Power Distribution Board*
   6. 5V *Regulator*
   7. *OLED Display*
   8. *8-Channel Relay*
   9. 12V *Switching Power Supply*
   10. 2*mm Inner Diameter Silicon Tube*
   11. *Kitchen Funnel*
   12. *Jumper Wires*

b) *Software*:
   1. *Raspbian*
   2. *Google Assistant*

B. *Flowchart Sistem Kerja Alat*

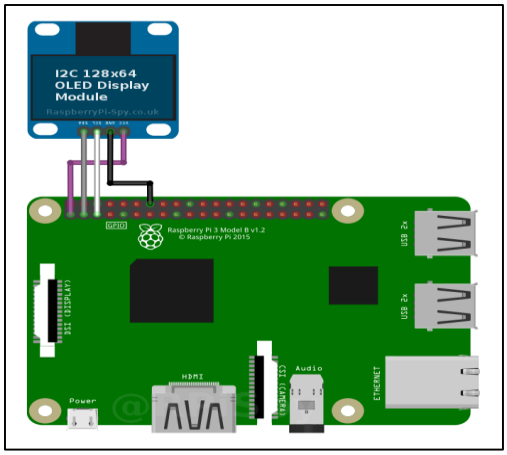


Gambar 2. Flowchart Sistem Kerja Alat

Flowchart diatas merupakan gambaran secara umum tentang sistematika dari rancang bangun alat ini. Dari penelitian ini menjelaskan tentang bagaimana cara kerja alat dari mulai inputan suara terdeteksi hingga bagaimana mikrokontroler memilih pin-pin GPIO yang bekerja dan memproses menu yang dipilih serta bahan mana saja yang akan tertuang sesuai jalur pompa peristaltic.

C. *Pengkabelan Modul Yang Digunakan*

1. OLED Display
   OLED *Display* Ini memiliki empat pin. Dua pin-nya terhubung pada *power* dan *ground* (Vcc dan Gnd) dan dua untuk antarmuka I2C (SDA dan SCL masing-masing pada GPIO 2 dan GPIO 3). OLED *display* dihubungkan langsung ke header GPIO Raspberry Pi menggunakan skema berikut:



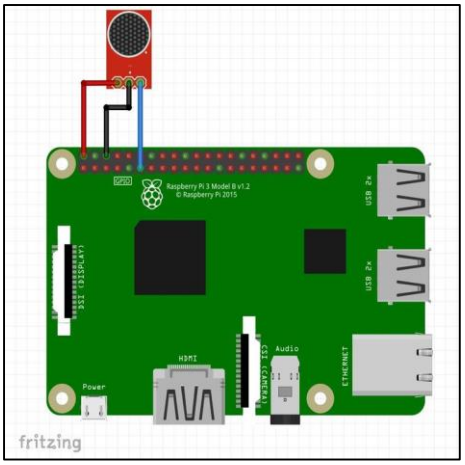
Gambar 3. *Wiring OLED Display* pada Pi

Tabel 1. Skema OLED Display ke Header GPIO

OLED	GPIO Pin	*Notes*
Vcc	1	3.3V
Gnd	6, 9, 14, 20, 25, 30,34 & 39	*Ground*
SCL	5	I2C SCL
SDA	3	I2C SCA

OLED *display* dapat dihubungkan ke pin Vcc pada Pin 1 atau 17 karena keduanya menyediakan 3.3V. OLED *display* dapat dihubungkan ke pin Gnd pada Pin 6, 9, 14, 20, 25, 30, 34 atau 39 karena semuanya menyediakan *ground.*

2. Sound Sensor
   Modul sensor suara ini digunakan sebagai alat perekam atau hanya inputan dalam menambahkan perintah pada *Google Assistant*, supaya setiap perintah yang digunakan sebagai inputan dapat terproses oleh mikrokontroler. Modul sensor yang digunakan pada rancang bangun ini memiliki 4 pin, yaitu untuk VCC, GND dan OUT (*analog output* dan *digital output*). Pin VCC akan dihubungkan pada pin 2 atau 4 karena menyediakan 5V, pin GND pada pin 6,9,14,20,25,30,34,39 karena semuanya menyediakan *Ground* serta pin *OUT* dihubungkan pada GPIO 17 (pin 11). Pada rancang bangun ini atau pada pengujian sersor, pin *analog output* tidak digunakan. Jadi pada skema pengkabelan hanya menggunakan 3 kabel *jumper*.



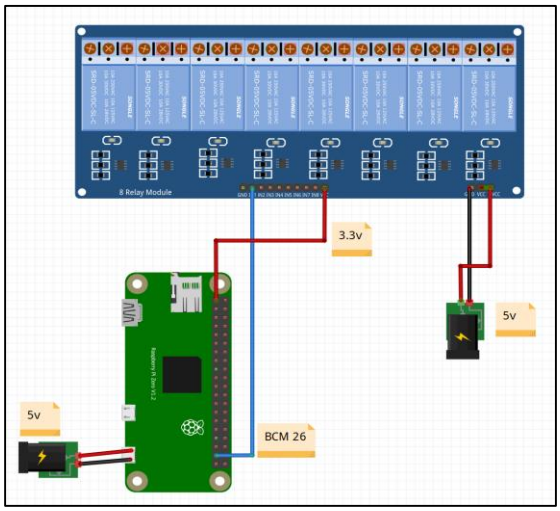
Gambar 4. *Wiring Sound Sensor* pada Pi

Sensor suara dihubungkan langsung ke header GPIO Raspberry Pi menggunakan skema berikut:

Tabel 2. Skema Sensor Suara ke Header GPIO

Sound Sensor	GPIO Pin	*Notes*
Vcc	2 & 4	5V
Gnd	6, 9, 14, 20, 25, 30,34 & 39	Ground
Digital Output	11	*Input to Pi*
Analog Output	-	*Not needed*

3. 8-Channel Relay
   Macam-macam *channel* (saluran) *relay* yang tersedia ada 2, 4, 8 dan bahkan 16 saluran modul, tergantung pada apa yang dibutuhkan. Pada rancang bangun alat ini, menggunakan 6 saklar (*relay*) untuk memicu 6 pompa peristaltik yang digunakan. Pada kasus ini, rancang bangun menggunakan *8-Channel Relay* sebagai pemicu pompa dan sengaja menyisakan 2 slot kosong pada *relay* sebagai cadangan jika ada penambahan pompa.



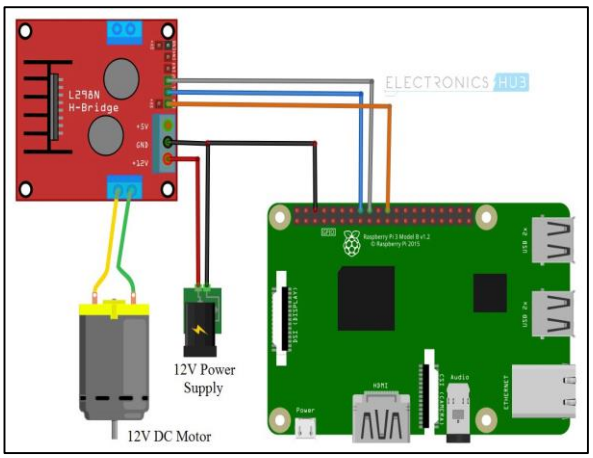
Gambar 5. *Wiring Relay* pada Pi

Strukturnya sangat sederhana karena semua pin diberi label. Bergantung pada berapa banyak relay yang akan dikontrol, kita perlu menghubungkan jumlah GPIO yang sesuai ke pin *IN*. Pada perancangan ini *relay* menggunakan inputan tenaga 5V, namun untuk program dari Pi diantarkan dengan tenaga 3.3V.

Tabel 3. Skema *Relay* ke Header GPIO

Relay	**GPIO Pin**	***Notes***
JD-Vcc	2 & 4	5V
Gnd	6, 9, 14, 20, 25, 30,34 & 39	*Ground*
Vcc	1 & 17	3.3V
IN	37	*Input to Pi*

4. Peristaltic Pump
   Pompa ini pada dasarnya adalah motor DC, sehingga memiliki banyak torsi. Di dalam pompa terdapat pola "semanggi" dari rol. Saat motor berputar, semanggi menekan tabung untuk menekan fluida. Terdapat PWM motor untuk mempercepat atau memperlambat laju aliran dan jika dihubungkan motor ke arah lain itu akan memindahkan cairan ke arah lain. Bekerja dengan baik dengan transistor daya (dasar hidup / mati) atau chip driver motor seperti L293D dan juga L298N. Untuk pengujian dan pengkabelannya dapat dilihat pada Gambar 6. Pengujian dengan modul L298N akan memudahkan kita untuk mengatur arah putar motor atau mengatur kecepatan putar pada motor.



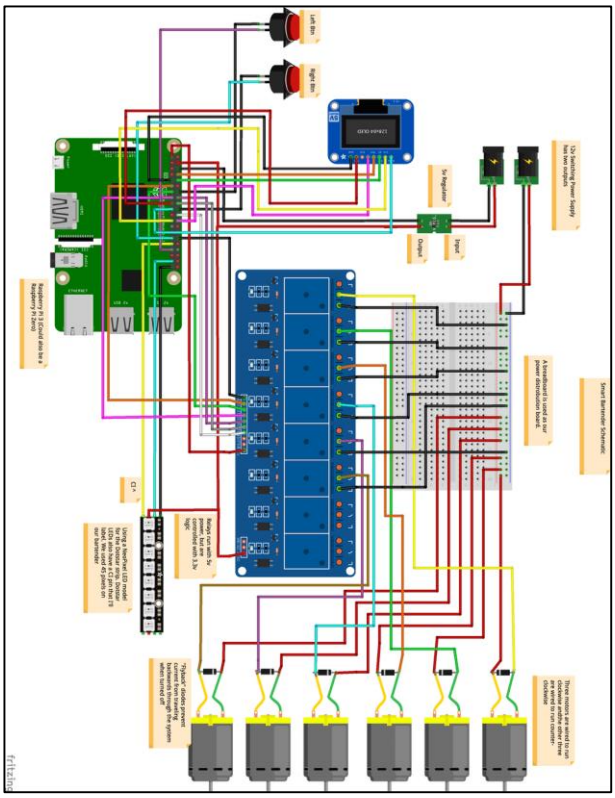
Gambar 6. *Wiring* Pompa pada Pi

Pada rancang bangun alat ini, pompa peristaltik sebelum disambungkan pada *relay* dan *Power Distribution Board* (pada rancang bangun ini menggunakan *breadboard* sebagai gantinya karna susah mencari modul alat yang diinginkan) terlebih dahulu disambungkan pada dioda (1N4007 Diode) untuk mencegah arus bergerak mundur melalui sistem saat dimatikan, namun jika *power supply* yang digunakan outputnya DC maka tidak perlu untuk menambahkan dioda. Pada rancang bangun ini, 3 motor pada pengkabelannya dipasang searah jarum jam dan 3 lainnya dipasang berlawanan arah jarum jam.

Tabel 4. Skema Pompa ke Header GPIO

H-bridge	**GPIO pin**	***Notes***
Vcc	-	12V to *Power Supply*
Gnd	6, 9, 14, 20, 25, 30,34 & 39	*Ground* Pi & *Power supply*
ENA	22	To Power the Pump
IN1	16	To Turn Clockwise
IN2	18	To Turn Counter-Clockwise

5. Wiring and Circuit Diagrams
   Dari beberapa alat-alat yang digunakan, adapun gambaran atau perencanaan pada pengkabelan modul-modul yang terhubung pada mikrokontroler. Gambaran ini adalah sebuah skema dari beberapa modul yang digunakan sebagai acuan pengkabelan pada rancang bangun sesungguhnya. Dari skema pengkabelan yang ada pada sub bab sebelumnya adalah sambungan pada pin mikrokontroler hanya digambarkan dalam skala umum atau hanya untuk pengujian modul atau *hardware* yang akan terpakai pada rancang bangun mesin pembuat koktail, namun pengkabelan yang digunakan pada rancang bangun yang sesungguhnya.



Gambar 7. Circuit Diagram

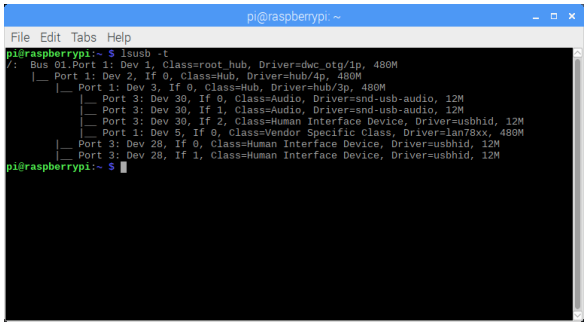
D. *Integrasi Pi dengan Audio Device dan Asisten Virtual*

Pada integrasi Pi dengan Audio Device, yang dimana pada tahap ini menghubungkan atau menambahkan fitur asisten pada rancang bangun. Audio Device ini berfungsi sebagai pembantu kita dalam melakukan sesuatu melalui perintah suara.

Asisten virtual yang digunakan pada rancang bangun ini menggunakan salah satu fitur dari *Google*, yaitu *Google Asisstant.* Langkah pertama yang dilakukan adalah membuat akun *Google* (jika belum memilikinya). Untuk menyelesaikan integrasi, adapun alat-alat yang dibutuhkan sebagai berikut:

1. *Raspberry Pi 3 B/B+*
2. *Speaker* dengan 3,5mm koneksi AUX
3. USB *microphone* (bisa juga dengan menambahkan USB *soundcard* dan *clip-on mic*)

Untuk perakitannya cukup simple karena pada raspberry sudah terdapat port usb dan aux. Jadi tinggal dipasangkan saja dan langsung mendownload program agar audio terdeteksi oleh pi.



Gambar 8. Communicating Pi with Audio Device

Langkah selanjutnya untuk menyambungkan *Google Asisstant* pada *Pi* adalah dengan mendownload software *arecord* dengan perintah **sudo apt-get install arecord.** Setting dengan perintah *arecord -1* dan contoh hasilnya akan seperti gambar di bawah ini.

```
pi@raspberrypi:~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Aud:
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Gambar 9. *Installing arecord*

Kemudian buka file **/etc/asound.conf** dengan VIM atau dengan NANO lalu tambah perintah seperti gambar dibawah, lalu buat file *asoundrc* dengan perintah **sudo nano .asoundrc.** Setelah itu setting speaker dengan menjalankan software *raspi-config* kemudian test dengan speaker dan mic.

```
pcm.!default {
  type asym
  capture.pcm "mic"
  playback.pcm "speaker"
}
pcm.mic {
  type plug
  slave {
    pcm "hw:1,0"
  }
}
pcm.speaker {
  type hw
  card 0
}
ctl.!default {
 type hw card 0
```

Gambar 10. Adding code to arecord

Untuk respon *Google Assistan* akan diawali dengan sapa dari pengguna seperti, *"Hello Google"* atau mungkin bisa diganti dengan nama yang diinginkan supaya membuat lebih spesifik untuk menghidupkan alat seperti contoh *"Hello Jenny",* lalu *cocktail maker* akan membalas *"Good day Boss"* atau,*"Hello Boss, How are you?"* seperti itu, dan pada sistem ini *Google Assistant* akan merespon dengan suara perempuan sesuai nama diberikan pada sistem.

Adapula beberapa command dan respon yang akan ditambahkan pada sistem:

- **Start command:**
  1. Hello Jenny
  2. I am thirsty
  3. Give me shot
  4. Thank you Jenny
- **Jenny respond:**
  1. Hello Boss
  2. Good day Sir
  3. What can I offer you?
  4. What would you like?
  5. Anytime Sir

Untuk memasukan perintah dan respon ini pertama-tama instal dulu software yang dibutuhkan **sudo apt-get install libasound2-dev bison python-dev espeak –y** selanjutnya coba jalankan test espeak dengan lib bahasa inggris dan merubah ke suara perempuan **espeak –vid+f2 "Hello Boss"** maka suara akan keluar pada speaker dan menandakan espeak berfungsi dengan baik.

Selanjutnya menginstal *pocketsphinx*, apk ini sedikit rumit karena dibutuhkan 2 module yang ada pada *sphinx* yaitu *sphinxbase* dan *pocketsphinx.* 2 module ini disimpan pada *Pi* di folder **"~"** kemudian di extract dan diinstal masing-masing dengan perintah seperti gambar dibawah ini.

```
cd ~/sphinxbase-5prealpha
sudo ./configure --enable-fixed
sudo make
sudo make install
```

Gambar 11. Install Sphinxbase

```
cd ~/pocketsphinx-5prealpha
sudo ./configure
sudo make
sudo make install
```

Gambar 12. Install Pocketsphinx

Setelah selesai penginstalan harus menginfokan kepada *Pi* tentang keberadaan *pocketsphinx* dengan command **sudo nano /etc/ld.so.conf**. Lalu tambahkan */usr/local/lib* dengan command **include /etc/ld.so.conf.d/*.conf** enter lalu **/usr/local/lib** kemudian *reboot* sistem.

Kemudian membuat language model dengan file .txt lalu upload ke ***http://www.speech.cs.cmu.edu/tools/lmtool-new.html*** dan download hasilnya masukan pada *Pi*. Pada kasus ini disimpan dalam folder **"~"**, kemudian masuk pada folder tersebut *~/pocketsphinx-5prealpha/src/programs* dan jalankan programnya.

```
*Untitled - Notepad
File  Edit  Format  View  Help
Hello Jenny
I am thirsty
Give me shot
Thank you Jenny

Hello Boss
Good day Sir
What can I offer you?
What would you like?
Anytime Sir
```

Gambar 13. Voice Command List

Selanjutnya setelah test voice command tinggal membuat logicnya. Untuk membuat *Pi* menjalankan apa yang kita perintah perlu memodifikasi *~/pocketsphinx-5prealpha/src/programs/continuous.c* seperti contoh jika pengguna mengatakan, *"Hello Jenny"* pada mic maka *Pi* akan menjawab, *"Hello Boss"* ke speaker. Namun sebelumnya harus di back-up terlebih dahulu pada **cd ~/pocketsphinx-5prealpha/src/programs** lalu enter **cp continuous.c continuous.c.old** kemudian edit dengan **nano continuous.c** lalu cari baris seperti gambar di bawah ini kemudian dirubah (bagian yang di block adalah yang ditambahkan).

```
...
        if (in_speech && !utt_started) {
            utt_started = TRUE;
            E_INFO("Listening...\n");
        }
        if (!in_speech && utt_started) {
            /* speech -> silence transition, time to start new utterance  */
            ps_end_utt(ps);
            hyp = ps_get_hyp(ps, NULL );
            if (hyp != NULL) {
                printf("%s\n", hyp);
                fflush(stdout);
                char word[30];
                if (hyp) {
                        sscanf(hyp, "%s", word);
                        if (strcmp(word, "Hello Jenny") == 0)
                        {
                        system ("espeak -vid+f2 'Hello Boss'");
                        }
                }
        }
...
```

Gambar 14. Edit Command Logic

Lalu selanjutnya save dan build ulang dengan cara **cd ~/pocketsphinx-5prealpha** enter kemudian **sudo make.** Kemudian langkah terakhirnya adalah menjalankan commandnya **sudo ./pocketsphinx_continuous -dict ~/4314.dic -lm ~/4314.lm -inmic yes -adcdev plughw:1,0** dan jangan lupa untuk dijalankan pada background dengan menggunakan **sudo nohup ./pocketsphinx_continuous -dict ~/4314.dic -lm ~/4314.lm -inmic yes -adcdev plughw:1,0 >/dev/null 2>&1 &** dan program sudah bisa dijalankan.

IV. HASIL DAN PEMBAHASAN

A. *Pengujian Mikrokontroler*

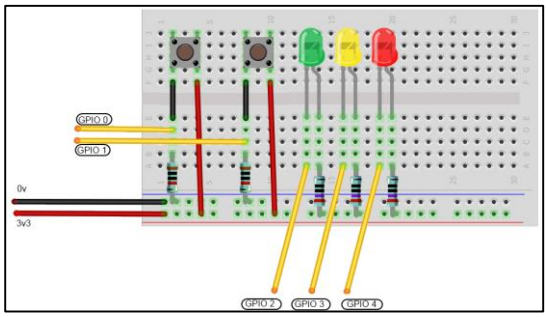
Peralatan yang digunakan pada pengujian ini adalah:

- *Raspberry Pi 3* model *B+*
- *3 Resistor 270* Ω
- *2 Resistor 10 k*Ω
- *Bread board*
- *LED*
- *9 Jumper wire male-female*
- *2 push button*

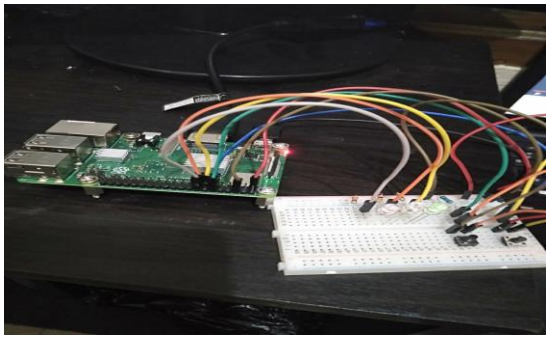
Langkah-langkah pengujiannya, pada *bread board* dirangkai menggunakan kabel *jumper* antara lampu LED (merah, kuning dan hijau), resistor beserta *push-button* seperti pada Gambar 7. Dimana *push-button* dihubungkan dengan *resistor 10 k*Ω masing-masing pada pin GPIO 0 dan pin GPIO 1. Sedangkan untuk lampu *LED* dihubungkan pada *resistor 270* Ω masing-masing pada pin GPIO 2, GPIO 3 dan GPIO 4. Untuk menjalankan pengujian ini *bread board* dihubungkan pada input *GPIO* pin 1 atau 17 (3,3V) atau pada pin 2 atau 4 (5V), sedangkan untuk ground dihubungkan pada pin 6, 9, 14, 20, 25, 30, 34 atau 39. Pada pengujian ini pin GPIO dapat dirubah-rubah sesuai fungsinya guna menguji tiap *header* apakah bekerja dengan baik atau tidak.

Pengujian mikrokontroler ini menggunakan 2 metode, yaitu:

- *Blink*
- *In-out*



Gambar 15. *Wiring Diagram* Pengujian GPIO pada B*read Board*



Gambar 16. Pengujian GPIO

Dimana pada cara *blink*, pada saat program dijalankan lampu LED akan berkedip bergantian sedangkan pada cara *in-out* sama seperti menyalakan lampu pada umumnya, dimana ketika tombol *push-button* dipijat maka lampu akan menyala. Program yang digunakan pada *Raspi* dapat dilihat pada Gambar 17 dan Gambar 18.

```
import RPi.GPIO as GPIO
GPIO.setup(13, GPIO.OUT)
GPIO.setup(11, GPIO.IN)
GPIO.setup(15, GPIO.OUT)
GPIO.setup(12, GPIO.IN)
while 1:
        if GPIO.input(11):
                GPIO.output(13, False)
        else:
                GPIO.output(13, True)

        if GPIO.input(12):
                GPIO.output(15, False)
        else:
                GPIO.output(15, True)
```

Gambar 17. Program Pengujian GPIO Cara *In-out*

```
from time import sleep
import RPi.GPIO as GPIO
GPIO.setup(13, GPIO.OUT)
GPIO.setup(15, GPIO.OUT)
GPIO.setup(16, GPIO.OUT)
while 1:
     GPIO.output(13, False)
     sleep(1)
     GPIO.output(13, True)
     sleep(1)

     GPIO.output(15, False)
     sleep(1)
     GPIO.output(15, True)
     sleep(1)

     GPIO.output(16, False)
     sleep(1)
     GPIO.output(16, True)
     sleep(1)
```

Gambar 18. Program Pengujian GPIO Cara *Blink*

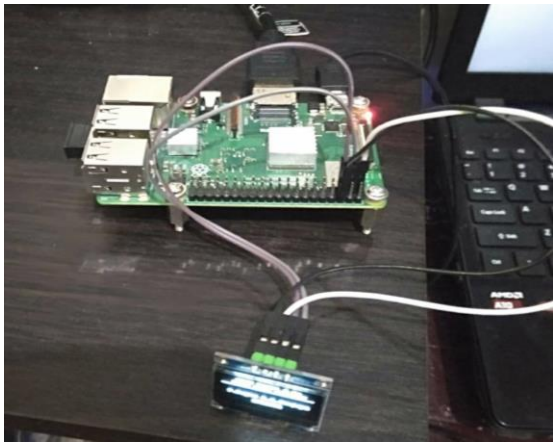
Setelah melakukan banyak pengujian dengan merubah-rubah pin GPIO menunjukkan, bahwa semua pin-pin *header* berjalan dan berfungsi dengan baik sesuai penggunaannya.

B. *Pengujian Display*

Pada pengujian *display* menggunakan modul *OLED display I2C 0,96 inch.* Pengujian *display* ini bertujuan untuk memunculkan karakter yang diprogram pada *raspi*. Modul *OLED display I2C 0,96 inci* yang digunakan dapat memunculkan karakter dengan resolusi 128×32 dan 128×64 piksel. Yang lebih murah memiliki piksel warna tunggal yang berwarna putih, kuning atau biru. Perangkat yang digunakan pada pengujian ini memiliki piksel putih dan menggunakan antarmuka *I2C* yang hanya membutuhkan empat kabel untuk dihubungkan ke *Pi.*

Peralatan yang digunakan pada pengujian ini antara lain:

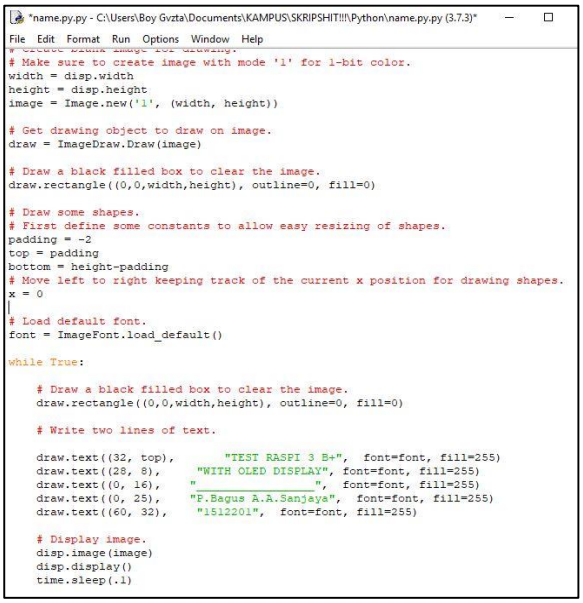
- Modul *OLED display I2C 0,96 inch*
- *Raspberry Pi 3* model B+
- *Jumper wire male-male*



Gambar 19. *Wiring OLED Display*

Langkah-langkah pengujian :

- Menghubungkan modul *OLED display I2C 0,96 inch* ke pin 3 atau GPIO 8 (SDA1(I2C)) dan pin 5 atau GPIO 9 (SCL1(I2C)) pada *Pi*.
- Mengubah program dari *Adafruit* yang sudah ter-*instal* pada *Pi* untuk menampilkan karakter yang diinginkan (pada pengujian ini menampilkan karakter nama dan NIM penguji).
- Mengamati tampilan pada LCD



Gambar 20. Edit Program *Adafruit*

Pada *Adafruit* menggunakan program yang bekerja pada layar yang memiliki resolusi 128×32. Program yang dijalankan pada pengujian ini masih berjalan dengan resolusi layar 128×64 piksel, tetapi lebih baik untuk mengubahnya sebelum berpindah ke sesuatu yang lebih rumit. Untuk melakukannya pada pengujian ini cukup mengedit skrip dan menonaktifkan baris konfigurasi 128×32 dengan menempatkan karakter # di depan, dan mengaktifkan baris 128×64 dengan menghapus karakter # dari depan. Langkah ini menjadi penting jika pengujian yang diinginkan memuat gambar sendiri untuk ditampilkan di layar.

```
# 128x32 display with hardware I2C:
#disp = Adafruit_SSD1306.SSD1306_128_32(rst=RST)

# 128x64 display with hardware I2C:
disp = Adafruit_SSD1306.SSD1306_128_64(rst=RST)
```

Gambar 21. Edit Konfigurasi Resolusi Layar

Gambar 15. Hasil Pengujian *Display*

C. Pengujian Pompa

Pompa yang digunakan pada pengujian ini adalah sistem pompa peristaltik yang memakai input 12V dan rata-rata alir (flow rate) 0-80mL per menit dengan keepatan putar 1 sampai 300 rpm.

Pada pengujian ini peralatan yang digunakan adalah:

- Pompa peristaltik
- Power Supply 12V
- Gelas ukur
- Stopwatch

Pengujian pompa ini untuk mengukur lama waktu pompa mengalirkan cairan sesuai takar dalam milliliter pada gelas ukur. Pada pengujian dilakukan 5 kali percobaan, lama waktu untuk 50mL, 100mL, 150mL, 200mL, dan 250mL. Dari percobaan tersebut ditemukan persamaan yang dapat menenemukan lama waktu motor mengalirkan cairan dengan menggunakan rumus fluida dinamis, dimana debit aliran fluida (Q) adalah jumlah volume fluida (V) yang mengalir persatuan waktu. Untuk menentukan debit ditemukan formula sebagai berikut:

$$Q = {}^{V}/_{t} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Dimana:

Q = Debit aliran fluida (m^3/detik)

V = Volume fluida (m^3)

t = Waktu (detik)

Karena dalam *datasheet* motor yang digunakan sudah tercantum debit aliran fluida yaitu 80mL per menit, maka perubahan rumus pada percobaan yang digunakan menjadi:

$$t = {}^{V}/_{Q} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Dengan rumus tersebut, maka pada 5 percobaan yang dilakukan akan mendapatkan perbandingan waktu pada pengujian nyata. Selisih waktu yang didapatkan akan menentukan baik atau tidaknya pompa bekerja pada rancang bangun alat.

Tabel 5. Pump Test Result I

Volume Air (mL)	**Pengujian waktu**		**Selisih (detik)**	***Error* (%)**
	Rumus (detik)	***Stopwatch* (detik)**		
10	7,5	7,6	0,1	1,3
50	37,5	40,4	2,9	7,2

100	75	82,7	7,7	9,3
150	112,5	121,3	8,8	7,3
200	150	160,1	10,1	6,3
Rata-rata error				**6,3**

Rumus perhitungan *error*:

$$Error = \frac{Selisih}{Pembacaan\ Stopwatch} \times 100\%$$

$$Rata - rata\ error = \frac{\sum error}{\sum pengujian}$$

Dari data pada tabel, percobaan pada tabel pertama hasilnya terhitung sangat lama dengan hasil rata-rata error **6,3%.** Wajar saja waktu error rata-ratanya masih terhitung tinggi karena selang pada pompa masih kosong belum terisi air. Jadi, untuk meminimalisir rata-rata error dilakukan lagi 3 percobaan dengan selang pompa yang sudah terisi air atau siap pakai untuk membuktikan hipotesis rata-rata error akan berkurang apabila pompa siap pakai.

Tabel 6. Pump Test Result II

Volume Air (mL)	**Pengujian waktu**		**Selisih (detik)**	***Error* (%)**
	Rumus (detik)	***Stopwatch* (detik)**		
10	7,5	7,5	0	0
50	37,5	39,2	1,7	4,3
100	75	80,4	5,4	6,7
150	112,5	118,8	6,3	5,3
200	150	157,4	7,4	4,7
Rata-rata error				**4,2**

Tabel 7. Pump Test Result III

Volume Air (mL)	**Pengujian waktu**		**Selisih (detik)**	***Error* (%)**
	Rumus (detik)	***Stopwatch* (detik)**		
10	7,5	7,5	0	0
50	37,5	38,4	0,9	2.3
100	75	77,2	2.2	2.8
150	112,5	115,4	2,9	2,5
200	150	154,4	4,4	2,8
Rata-rata error				**2,1**

Tabel 8. Pump Test Result III

Volume Air (mL)	**Pengujian waktu**		**Selisih (detik)**	***Error* (%)**
	Rumus (detik)	***Stopwatch* (detik)**		
10	7,5	7,5	0	0
50	37,5	38,4	0,9	2.3
100	75	77,2	2.2	2.8
150	112,5	115,4	2,9	2,5
200	150	154,4	4,4	2,8
Rata-rata error				**2,1**

Rumus perhitungan *error*:

$$Error = \frac{Selisih}{Pembacaan\ Stopwatch} \times 100\%$$

$$Rata - rata\ error = \frac{\sum error}{\sum pengujian}$$

Dari percobaan pada tabel pertama berbeda dengan table kedua karena pada percobaan pertama selang masih ada sedikit angin yang berbentuk *bubble*, sehingga pada perconbaan berikutnya selang semakin penuh dengan air dan membuat ketepatan waktu pengisian semakin baik serta lebih mendekati akurat. Dapat disimpulkan pula bahwa pompa peristaltik mampu bekerja dengan baik, karena perbandingan waktu perhitungan menggunakan rumus dan pengujian menggunakan *stopwatch* terhitung tipis dengan hasil rata-rata error **2,1%**

D. Pengujian Sistem Keseluruhan

Pada pengujian alat pembuat koktail ini menggunakan 6 bahan sebagai resep untuk membuat beberapa jenis minuman koktail yang digunakan sebagai menu. 6 bahan minuman tersebut adalah, vodka, peach schnapps, tequila, cranberry juice, orange juice dan grenadine syrup.

Dengan bahan-bahan tersebut dapat membuat 6 menu koktail yang digunakan pada sistem:

1. *Madras:*
   - 5omL *vodka*
   - 50mL *cranberry juice*
   - 50mL *orange juice*

2. *Sex on the Beach:*
   - 10mL *peach schnapps*
   - 35mL *vodka*
   - 35mL *orange juice*
   - 35mL *cranberry juice*

3. *Tequila Sunrise:*
   - 50mL *tequila*
   - 100mL *orange juice*
   - 10mL *grenadine syrup*

4. *Screwdriver:*
   - 25mL *vodka*
   - 125mL *orange juice*

5. *Woo Woo:*
   - 25mL *peach schnapps*
   - 25mL *vodka*
   - 100mL *cranberry juice*

6. *Jamaican Sunrise:*
   - 50mL *vodka*
   - 100mL *orange juice*
   - 10mL *grenadine syrup*

Adapun contoh sistem kerja program pada *cocktail maker* dengan hanya menggunakan tombol pada alat dan LCD sebagai indikatornya:

I. Sistem kerja tombol
   1. *Tombol kanan (select)*
      Adalah tombol untuk memilih menu/minuman.
   2. *Tombol kiri (menu)*
      Adalah tombol untuk memilih daftar menu. Pada sistemnya ada 7 pilihan daftar menu.

Lebih jelasnya bisa dilihat pada:
- **Web**: *hackster.io/smart bartender*
- **Youtube**: *rapberry smart bartender*
- **Github**: *hacker shack/smart bartender*

II. Tampilan pada LCD 16x2:
- Saat sebelum memilih menu:
  1. *Fancy Cocktail?* (baris pertama)
  2. *Please Select!* (baris kedua)
- Saat memilih menu:
  1. *MENU:* (baris pertama)
  2. Pada baris kedua akan terlihat nama menu cocktail dan akan berganti saat tombol dipencet, contoh: *Madras, Sex on the Beach, Tequila Sunrise, dll.*
- Saat menu terpilih:
  1. (baris pertama) MENU:
  2. (baris kedua) MIXING!

- Saat selesai mencampur:
  1. *VOILAAA!!!* (baris pertama)
  2. *Have a Drink!* (baris kedua)
- Daftar menu pada LCD:
  1. *Madras*
  2. *Sex on the Beach*
  3. *Tequila Sunrise*
  4. *Screwdriver*
  5. *Woo Woo*
  6. *Jamaican Sunrise*
  7. *CLEAR PUMP!*

Dari menu yang sudah disiapkan akan mendapatkan jumlah volume yang tercampur dalam wadah atau gelas ukur. Hasil pengujian akan mencatat waktu pencampuran dengan *stopwatch* dan membandingkan jumlah volume pencampuran pada gelas ukur.

Tabel 9. System Test Result

Menu	***Volume***		**Waktu**		**Selisih**		***Error* (%)**	
	Prog ram (mL)	**Gelas Ukur (mL)**	**Rumus (s)**	**Stop watch (s)**	**V**	**t**	**V**	**t**
Madras	150	150	112,5	115,4	0	2,9	0	2,5
Sex on the Beach	115	115	86,4	88,7	0	2.3	0	2,5
Tequila Sunrise	160	160	120,3	123,2	0	2,9	0	2,3
Screw driver	150	150	112,5	115,4	0	2,9	0	2,5
Woo Woo	150	150	112,5	115,4	0	2,9	0	2,5
Jamaica n Sunrise	160	160	120.3	123,2	0	2,9	0	2,3
Rata-rata error							**0**	**2,43**

Dari tabel terdapat perbandingan volume dan catatan waktu pengisian. Perbandingan pengisian waktu yang tipis dengan rata-rata error sebesar **2,43%,** untuk volume terisi presisi **0%** rata-rata error sesuai dengan takaran yang dikehendaki. Maka alat yang telah dirancang dapat dikatakan cukup cepat dan presisi dalam pencampuran dan pengisian. Jumlah volume yang tertera pada tabel adalah jumlah volume sebelum ditambahkannya es batu untuk penyajiannya.

Di pengujian yang terakhir ini adalah dimana mencoba sistem dengan menggunakan perintah suara. Jadi pada sub-bab ini sebelum menguji, harus menambahkan dulu perintah-perintah yang diinginkan.

Layanan *Google Assistant* memberikan control penuh atas integrasi dengan asisten dengan menyediakan *Endpoint Streaming*. Streaming kueri audio pengguna ke *endpoint* ini untuk menerima respon audio *Google Assistant.* Ada beberapa langkah-langkah yang dilakukan sebelum menguji sistem, antara lain:

1. Menyiapkan hardware dan akses jaringan
2. Mengkonfigurasi dan menguji audio
3. Mengkonfigurasi project developer dan setelan akun
4. Mendaftarkan model perangkat
5. Menginstal SDK dank ode contoh
6. Menjalankan kode contoh

Untuk menentukan pengujian dengan *Google Assistant* ini berhasil atau tidak, maka dilakukanlah survey pada 5

mahasiswa Teknik Elektro Institut Teknologi Nasional jika command berfungsi atau tidak ataupun mesin bekerja atau tidak. Berdasarkan survey yang disediakan, terdapat 2 tabel yang bisa digunakan. Tabel pertama adalah untuk menguji pengenalan kata oleh 5 mahasiswa untuk mengetahui keakuratan *Google Speech API* memproses perintah.

Tabel 10. Google Assistant System Test Respond

Fungsi	**Perintah**	**Peserta**				
		1	**2**	**3**	**4**	**5**
Sapa	Hallo Google	√	√	√	√	√
	How are you	√	√	√	√	√
	Give me a shot	√	×	√	√	√
	Thank you	√	√	√	√	√
Menu	Madras	√	√	√	√	√
	Sex on the Beach	√	×	√	√	√
	Tequila Sunrise	√	√	√	√	×
	Screwdriver	√	√	√	×	√
	Woo Woo	√	√	√	√	√
	Jamaican Sunrise	√	√	√	√	√
	Clear Pump	√	√	√	√	√

Pada pengujian terhadap perintah suara untuk *cocktail maker* didapatkan hasil bahwa akurasi *Google Speech API* dalam mengkonversi suara untuk perangkat mencapai **92,8%.**

Tabel 11. Google Assistant Time Test Respond

Fungsi	**Perintah**	**Peserta**				
		1*	**2***	**3***	**4***	**5***
Sapa	Hallo Google	2,15	2,32	2,04	2,11	3,25
	How are you	1,26	1,13	2,06	3,49	150
	Give me a shot	3,42	3,12	3,21	3,46	2,42
	Thank you	1,23	1,32	1,50	2,13	1,26
Menu	Madras	3,24	3,21	3,46	3,25	3,54
	Sex on the Beach	3,52	3,54	4,32	3,24	4,08
	Tequila Sunrise	3,15	3,22	3,18	3,46	3,12
	Screwdriver	2,24	3,46	2,56	3,27	3,44
	Woo Woo	3,45	3,44	3,22	3,17	3,42
	Jamaican Sunrise	3,15	3,42	2,54	3,49	3,23
	Clear Pump	2,32	2,13	2,46	2,15	3,05

**hitungan dalam detik*

Sementara hasil pengujian waktu konversi perintah suara ke *Google Speech API* terhadap 5 mahasiswa atau pengguna didapatkah hasil bahwa rata-rata waktu yang dibutuhkan untuk merespon yaitu **5,53** detik.

Jadi, bisa ditarik kesimpulan untuk bisa menghidangkan 1 menu dari proses sapa hingga pencampuran bahan hingga selesai, cocktail maker hanya membutuhkan waktu kurang dari 3 menit saja. Dan dalam hal ini, Pembuat Koktail Berbasis Raspberry Pi dengan Kontrol Perintah Suara mampu mempermudah dan mempercepat pengguna dalam penyajian ataupun membuat koktail.

V. KESIMPULAN

A. Kesimpulan

Setelah melakukan beberapa pengujian pada sistem keseluruhan yang digunakan pada ***"Rancang Bangun Pembuat Koktail Berbasis Raspberry Pi dengan Kontrol Perintah Suara"***, adapun kesimpulan yang dapat diperoleh adalah sebagai berikut:

1. Mesin pembuat koktail dapat dibangun dengan menggunakan Raspberry Pi 3 model B+ dan pompa peristaltic yang diintegrasikan dengan Google Assistant.
2. Pompa mampu bekerja dengan baik dengan rata-rata *error* sebesar **2,1%**.
3. Perbandingan waktu pencampuran bahan dengan rata-rata error sebesar **2,43%**, untuk volume minuman terisi presisi sesuai dengan takaran yang dikehendaki.
4. Akurasi *Google Speech API* dalam mengkonversi suara untuk perangkat mencapai **92,8%.**
5. Rata-rata waktu yang dibutuhkan *Google Speech API* untuk merespon yaitu **5,53** detik.

Pembuat Koktail Berbasis Raspberry Pi dengan Kontrol Perintah Suara mampu mempermudah pengguna dalam penyajian ataupun membuat koktail.

B. Saran

Dari penelitian, perangkat ini bukanlah dalam bentuk sempurna. Untuk pengembangan selanjutnya mungkin dapat dilakukan penambahan *hardware* maupun *software*. Adapun saran yang diperoleh pada penelitian ini adalah sebagai berikut:

1. Menambahkan fungsi notifikasi pada mesin ketika minuman telah selesai dituangkan dan atau ketika botol minuman yang akan segera habis.
2. Menambahkan fitur pengaduk ataupun shaker agar minuman lebih siap minum oleh pengguna.

REFENSI

[1] Siregar E. A., Purwantoro S., Trisnadoli A., Rancang Bangun Mesin Pembuat Minuman Cepat Saji Otomatis Berbasis Arduino Dengan Kontrol Android. Proceeding Applied Business and Engineering Conference (ABEC) (2014) ISSN 2339-2053.

[2] Kusumawardani R., Sumiharto R., Rancang Bangun Sistem Pencampur Minuman Bersoda Berdasarkan Kadar Keasaman Berbasis PLC OMRON CP1H-XA40DR-A. IJEIS, Vol.5, No.1, April (2015) ISSN: 2088-3714.

[3] Giant R. F., Darjat, Sudjadi. (2015). Perancangan Aplikasi Pemantau dan Pengendali Piranti Elektronik pada Ruangan Berbasis Web. Transmisi, 17, (2): e-ISSN 2407–6422.

[4] Subroto, Lusmiyanto B., Rahardjo T., Modifikasi dan perbaikan pompa peristaltik pada mesin pengukur viskositas sol uranium. Penelitian dan Pengelolaan Perangkat Nuklir. Pusat Teknologi Akselerator dan Proses Bahan. Yogyakarta, 26 September 2012 ISSN 1410 – 8178.

[5] Lukman M. P., Junaedy, Rieuwpassa Y. F. Y., (2018). Sistem Lampu Otomatis Dengan Sensor Gerak, Sensor Suhu Dan Sensor Suara Berbasis Mikrokontroler. Jurnal Resistor, 1, (2): p-ISSN 2598-7542, e-ISSN 2598-9650.

[6] Patel B. N., Prajapati M. M., OLED: A modern display technology. International Journal of Scientific and Research Publications, 4, (6), 2014: ISSN 2250-3153.

[7] Ardiansyah R. A., (2016). Perancangan Sistem Pengendali Robot Melalui Internet Menggunakan Raspberry Pi. J.Oto.Ktrl.Inst (J.Auto.Ctrl.Inst), 8, (1): ISSN 2085-2517.

[8] Lapono L. A. S., Pingak R. K., (2018). Rancang Bangun Sound Level Meter Menggunakan Sensor Suara Berbasis Arduino Uno. Jurnal ILMU DASAR, 19, (2): 111-116.

[10] Maryanto L. E., Basyirun, Anis S., (2018). Pengaruh Diameter Roller Terhadap Debit Pompa Peristaltik.

[11] Hacker Shack. Smart bartender. Hackster.io, an Avnet Community. 2018. https://www.hackster.io/hackershack/smart-barteder-5c430e

[12] Google. Google Assistant. Wikipedia, Wikimedia Foundation, Inc. 2016. https://en.wikipedia.org/wiki/Google_Assistant

[13] Nyebarilmu. Cara mengakses sensor suara menggunakan Arduino Uno. Nyebarilmu.com: In Arduino Projects Tutorial, Sensor Arduino. 2017. https://www.nyebarilmu.com/cara-mengakses-sensor-suara-menggunakan-arduino-uno/

[14] Wikipedia. Peristaltic Pump. Wikipedia, Wikimedia Foundation, Inc. 2019. https://en.wikipedia.org/wiki/Peristaltic_pump

[15] Lady Ada, 0.96" mini Colour OLED. Adafruit, Engineered in NYC Adafruit. 2012. https://learn.adafruit.com/096-mini-color-oled/overview