

PEMBUATAN RAK KOMPONEN SEMI OTOMATIS BERBASIS RASPBERRY PI 3 dan ARDUINO NANO

Jonathan Pratama Lukito, Resmana Lim, Handry Khoswanto
Program Studi Teknik Elektro, Universitas Kristen Petra
Jl.Siwalankerto 121-131, Surabaya 60236, Indonesia

E-Mail: jonathan.pratama.lukito@gmail.com ; resmana@petra.ac.id ; handry@petra.ac.id

Abstrak - Ide ini diambil dari masalah ketidakpraktisan toko komponen elektronika dalam manajemen komponen yang dijual pada rak komponen dan sangat mengharapkan ketelitian penjual dalam mengetahui posisi komponen yang terletak.

Sistem yang dibuat ini terdiri dari rak komponen dan simulasi rak komponen yang terdapat LED sebagai indikator drawer dan komponen-komponen pendukung didalamnya. Sistem ini memiliki sebuah halaman web pembelian sederhana sebagai antar muka pengguna yang terhubung didalam satu jaringan lokal. Seluruh sistem ini dikendalikan oleh Raspberry Pi 3 dan Arduino Nano.

Berdasarkan hasil pengujian yang dilakukan, dapat diambil kesimpulan bahwa sistem yang dibuat ini telah berjalan dengan benar sesuai dengan pengujian.

Kata Kunci— Pembuatan Rak Komponen Semi Otomatis, Raspberry Pi 3, Arduino Nano

I. PENDAHULUAN

Perkembangan teknologi dan modernisasi peralatan elektronik telah menyebabkan perubahan yang sangat mendasar didalam aktivitas manusia sehari-hari, dimana manusia selalu menginginkan segala sesuatunya serba otomatis, praktis dan fleksibel [1]. Perkembangan teknologi pada toko baik secara fisik maupun online sangat pesat, semuanya menginginkan kepraktisan dan kemudahan pengelolaan. Jumlah barang yang banyak tentu sangat merepotkan bila tidak termanajemen dengan teknik penyimpanan barang yang tertata, mudah dicari, dan ringkas. Dengan demikian, diperlukan sebuah sistem otomatis untuk penyimpanannya.

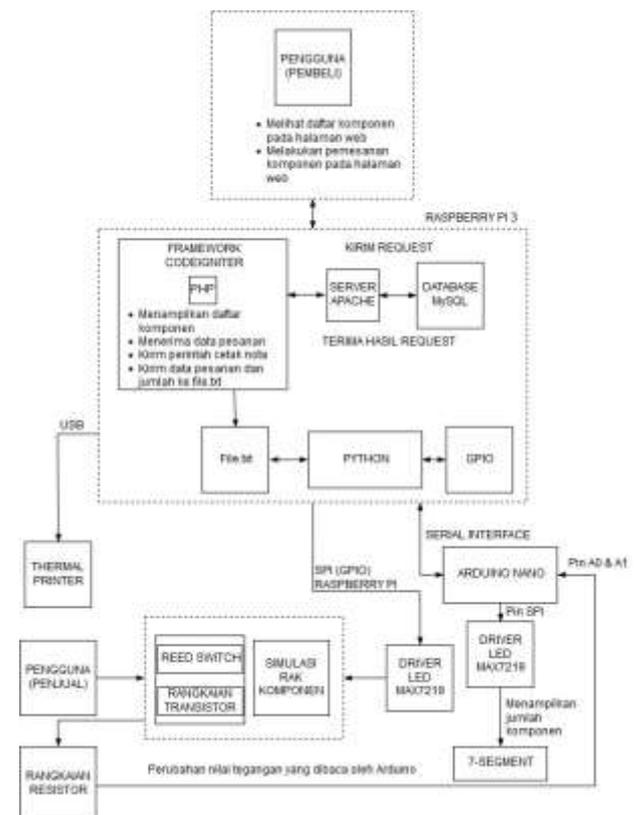
Penulis mengambil kasus toko komponen elektronika sebagai permasalahan yang penulis lihat dan ingin penulis selesaikan dengan sebuah sistem otomatis yang penulis buat. Toko komponen saat ini masih mengandalkan ketelitian mata manusia untuk melihat daftar komponen yang dicari pada rak komponen. Hal ini dapat memakan waktu yang lama, karena harus melihat daftar komponen pada masing-masing *drawer* rak komponen apabila tidak terorganisir dengan baik.

Ide ini penulis dapatkan dari halaman web kickstarter, dengan judul Chipseasy Smart Electronic Component Organizer by 3DFactory [2] dan ketika melihat video dari halaman YouTube yang berjudul [D.I.Y.] How to Store and Organise Electronic Parts (Arduino Project) [3]. Ide dari YouTube yang berjudul [D.I.Y.] How to Store and Organise Electronic Parts (Arduino Project) [3] memiliki kemiripan dengan yang penulis buat dimana rak komponen

yang dirakit menggunakan indikator LED untuk mengetahui posisi komponen yang dituju. Perbedaannya terletak pada desain hardware yang digunakan dan desain rangkaian LED yang digunakan.

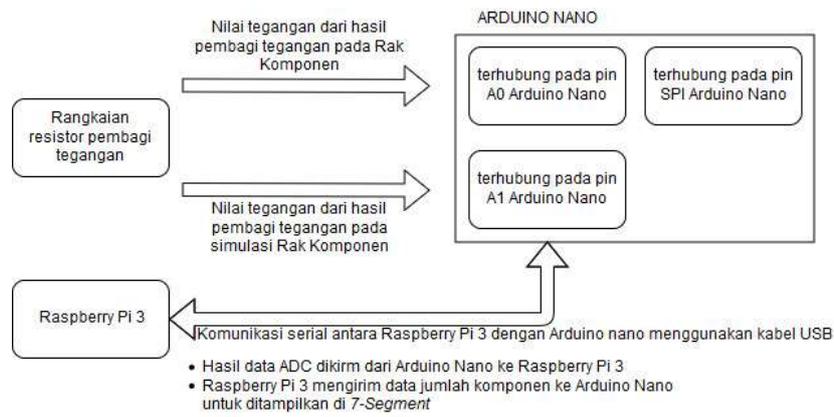
Kelebihan dari sistem yang penulis buat adalah pengguna dapat mengetahui berapa banyak komponen yang harus diambil ketika drawer tersebut dibuka dan dapat membuka drawer secara acak sesuai LED yang menyala. Selain itu, pengguna juga menerima cetakan nota dari *thermal printer* sehingga desain sistem yang penulis buat ini untuk digunakan pada toko komponen elektronika.

II. PERENCANAAN DESAIN SISTEM DAN ALAT



Gambar 1. Desain Sistem

Pada Gambar 1, merupakan desain koneksi antara raspberry pi 3 dengan hardware penunjang. Desain sistem pada Gambar 2 dari rak komponen semi otomatis ini bertujuan untuk memudahkan pengguna dengan cepat menemukan komponen yang ingin dicari pada sebuah rak komponen. Sistem ini dapat diterapkan pada toko komponen elektronika karena memudahkan pekerjaanya

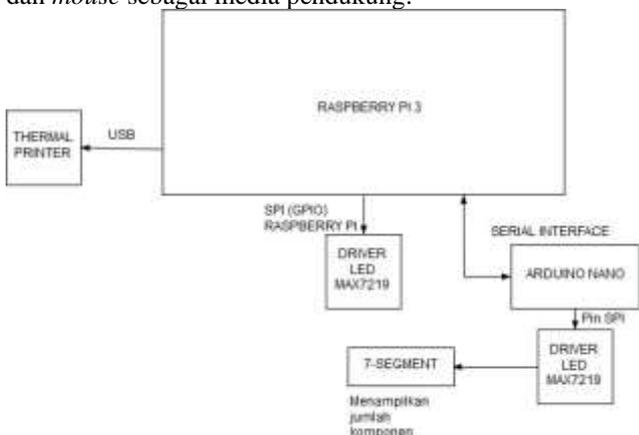


Gambar 2 Desain Diagram Koneksi Antara Arduino Nano Dengan Hardware

untuk mencari komponen pada rak yang besar. Secara garis besar, sistem ini terdiri dari input, proses, dan output. Pada desain input, pengguna membuat sebuah pesanan komponen pada halaman web, setelah pengguna membuat sebuah pesanan, maka langsung berpindah pada bagian proses yang terjadi ketika ada pesanan masuk. Proses dari *output* pertama adalah LED pada rak komponen atau LED pada simulasi rak komponen menyala sesuai pesanan pengguna dan *thermal printer* akan mencetak nota sesuai pesanan. Pada kondisi ini, mulai terjadi interaksi antara pengguna dengan *hardware*, yaitu ketika pengguna menarik *drawer* rak komponen atau pengguna menekan tombol pada simulasi rak komponen. Data interaksi tersebut diterima dan diolah oleh Arduino Nano, kemudian data tersebut dikirimkan secara serial ke Raspberry Pi 3. Saat Raspberry Pi 3 menerima data tersebut, Raspberry Pi 3 mematikan LED yang menyala sesuai interaksi antara pengguna dengan *hardware*.

A. Desain Hardware Koneksi Raspberry Pi 3

Desain *hardware* yang digunakan pada sistem ini terdiri dari Raspberry Pi 3 yang terdapat rangkaian *breakout board*, Arduino Nano yang terdapat rangkaian modul resistor, driver LED MAX7219, *step down* DC-DC converter, rak komponen yang terdiri dari rangkaian *reed switch*, rangkaian LED rak komponen dan rangkaian modul transistor, simulasi rak komponen yang terdiri dari rangkaian LED simulasi rak komponen dan rangkaian tombol, dan yang terakhir adalah *thermal printer*, *monitor* dan *mouse* sebagai media pendukung.



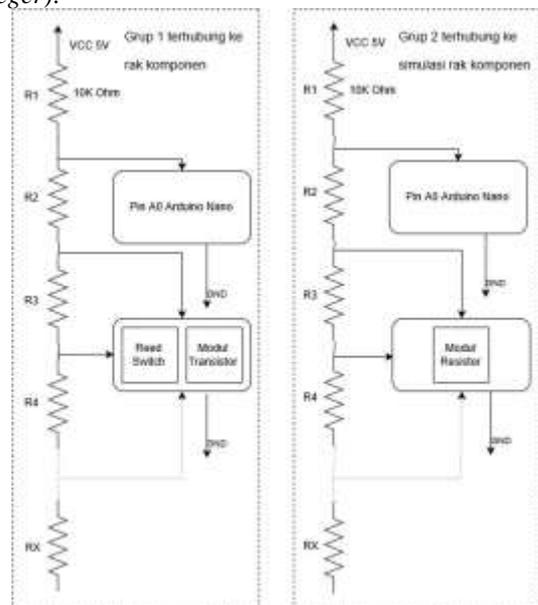
Gambar 3 Desain Diagram Koneksi Raspberry Pi 3 Dengan Hardware

B. Desain Hardware Koneksi Arduino Nano

Arduino Nano digunakan untuk menerima nilai tegangan yang terbaca, sebagai ADC (*Analog to Digital Converter*), data ADC tersebut diolah menjadi nilai *integer* dan mengirim data tersebut ke Raspberry Pi 3 menggunakan USB. Mengingat Raspberry Pi 3 tidak memiliki fitur ADC, sehingga digunakan Arduino Nano sebagai pengganti IC (*Integrated Circuit*) ADC. Arduino yang digunakan pada pembuatan proyek ini adalah Arduino Nano. Gambar 3 adalah desain diagram koneksi dari Arduino Nano.

C. Desain Rangkaian Modul Resistor

Rangkaian modul resistor didesain untuk membagi nilai tegangan 5V DC menjadi nilai tegangan yang diinginkan. Nilai tegangan tersebut disesuaikan dengan perhitungan dari rangkaian seri pada resistor sehingga tegangan terbagi diantara resistor tersebut. Alasan digunakan rangkaian pembagi tegangan ini agar Raspberry Pi 3 mengetahui posisi dari *drawer* rak komponen yang dibuka atau posisi dari tombol yang ditekan pada simulasi rak komponen. Gambar 4 adalah skematik secara garis besar dari rangkaian modul resistor. Tabel 1 adalah tabel hasil perhitungan nilai pembagi tegangan untuk mendapatkan data ADC yang sesuai. Data yang dikirimkan ke Raspberry Pi 3 adalah data dari kolom nilai ADC dalam bentuk INT (*integer*).



Gambar 4 Desain Rangkaian Pembagi Tegangan

Pada Gambar 4 terlihat pembagi tegangan terjadi ketika salah satu dari R2, atau R3, atau R4, atau RX terhubung ke ground. Hasil pembagi tegangan terbaca diantara resistor R1 dan R2 yang terhubung ke pin A0 dan A1 pada Arduino Nano. Dengan resolusi dari Arduino Nano sebesar 10 bit dengan nilai terbesar adalah 1023, hasil pembagi tegangan tersebut dapat dijelaskan dengan rumus berikut:

$$\frac{R_2}{R_1+R_2} \times V_{IN}$$

Keterangan:

R2 = resistor pembagi

R1 = resistor utama

Vin = tegangan input pada R1

Nilai R2 hingga RX akan terakumulasi terus menerus tergantung resistor mana yang terhubung ke *ground*. Sebagai contoh bila R2 terhubung ke *ground*, maka

$$\begin{aligned} V_{R2} &= \frac{R_2}{R_1+R_2} \times V_{IN} \\ &= \frac{100}{10100} \times 5000mV \\ &= 49.5mV \end{aligned}$$

Nilai tersebut adalah nilai hasil tegangan yang terbagi, ketika nilai tegangan tersebut masuk ke Arduino Nano, maka ADC pada Arduino Nano akan menerima nilai tersebut sesuai resolusi ADC yang dimiliki oleh Arduino Nano. Untuk mengetahui perubahan nilai ADC setiap kenaikan tegangan, dapat menggunakan rumus:

$$\frac{5000mV}{1023} = 4.88mV$$

Keterangan:

5000mV = Vin

1023 = hasil konversi dari resolusi Arduino (10 bit)

Dari rumus tersebut, dapat diketahui setiap kenaikan tegangan 4,88mV, maka ADC pada Arduino Nano akan bertambah satu kali. Nilai perubahan ADC tersebut *dicompare* dengan nilai pembagi tegangan pada resistor dengan rumus:

$$\left(\frac{4,88}{V_x}\right)/10$$

Keterangan:

4,88 = hasil pembagian tegangan input dan hasil konversi resolusi Arduino

Vx = hasil dari nilai tegangan yang terbagi

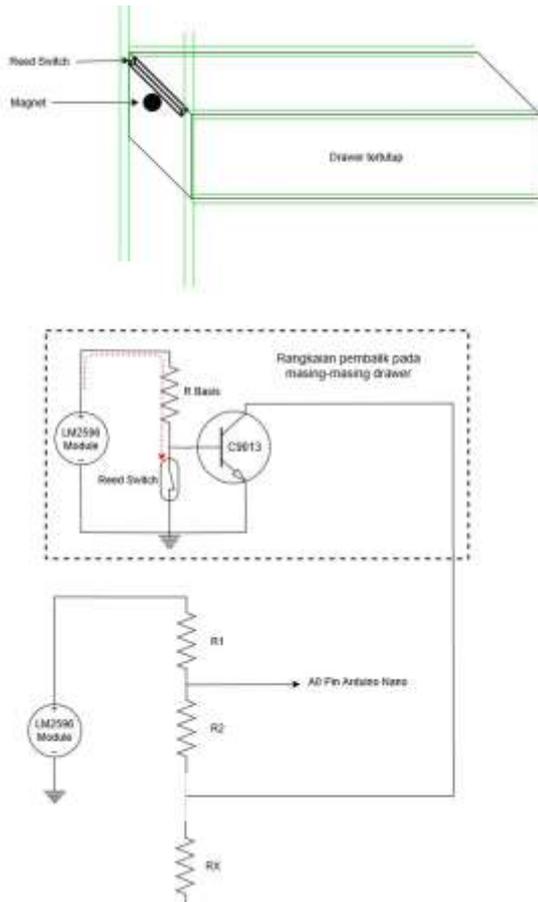
Melalui rumus tersebut, dapat diketahui tegangan yang telah diolah oleh ADC Arduino Nano. Nilai tersebut adalah hasil akhir dari nilai yang diterima dari pembagi tegangan sehingga nilai tersebut siap untuk dikirimkan ke Raspberry Pi 3. Tabel 1 adalah hasil perhitungan nilai ADC dengan kondisi ideal sampai mendapatkan nilai akhir pada Arduino Nano. Untuk simulasi rak komponen, nilai resistor yang digunakan sama. Perbedaannya terletak pada nilai akhir yang

diolah ditambah dengan angka 15, karena nomor pada simulasi rak komponen adalah lanjutan dari nomor rak komponen.

no rak	R Seri	R1	Nilai tegangan	Range	Step	Nilai ADC/10
1	100	10000	49,50	1023	4,88	1
2	100	10000	99,01	1023	4,88	2
3	100	10000	148,51	1023	4,88	3
4	100	10000	198,02	1023	4,88	4
5	100	10000	247,52	1023	4,88	5
6	100	10000	297,03	1023	4,88	6
7	100	10000	346,53	1023	4,88	7
8	100	10000	396,04	1023	4,88	8
9	100	10000	445,54	1023	4,88	9
10	100	10000	495,05	1023	4,88	10
11	100	10000	544,55	1023	4,88	11
12	100	10000	594,06	1023	4,88	12
13	100	10000	643,56	1023	4,88	13
14	100	10000	693,07	1023	4,88	14
15	100	10000	742,57	1023	4,88	15

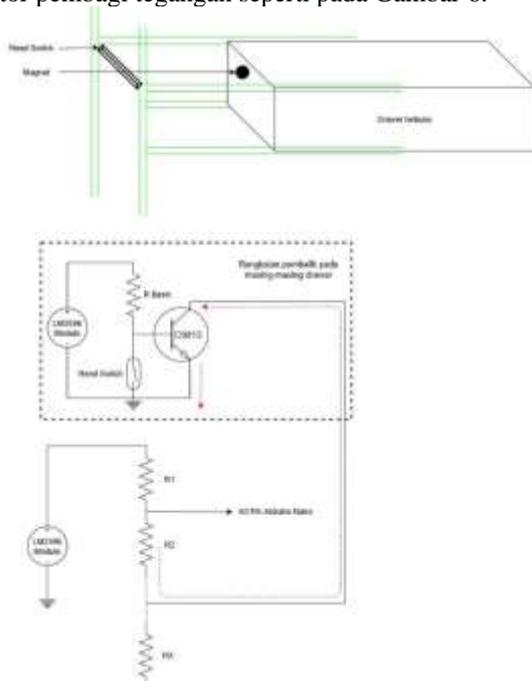
Tabel 1 Hasil Perhitungan Nilai ADC (Dengan Asumsi Ideal)

D. Desain Rangkaian Modul Transistor



Gambar 5 Arah Arus Saat Drawer Tertutup

Saat posisi *drawer* tertutup seperti pada Gambar 5, arus pada R basis akan mengalir ke *ground* sehingga kolektor dan emitor akan terbuka. Saat posisi *drawer* terbuka seperti pada Gambar 6, arus akan mengalir ke basis yang menyebabkan kolektor dan emitor tertutup, kondisi ini menyebabkan transistor tersebut akan melewati arus dari resistor pembagi tegangan seperti pada Gambar 6.

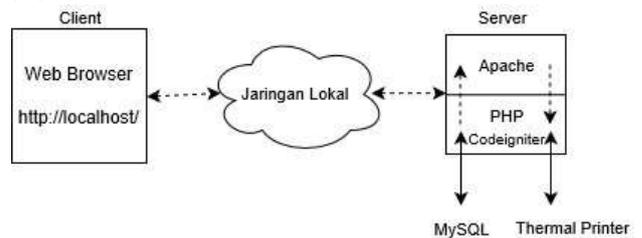


Gambar 6 Arah Arus Saat Drawer Terbuka

Posisi *reed switch* dan magnet bersentuhan saat kondisi awal membuat kondisi *reed switch* menjadi *normally close*, sehingga digunakan rangkaian pembalik logika menggunakan komponen transistor.

E. Desain Halaman Web

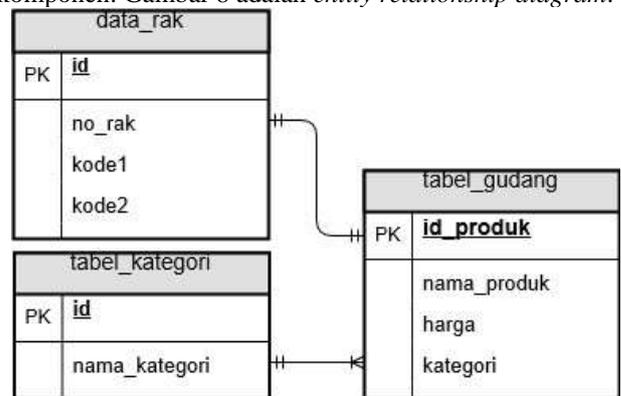
Halaman web digunakan untuk menampilkan data dari *database*. Program PHP pada halaman web ini juga memberi perintah ke *thermal printer* untuk mencetak nota dan mengirimkan perintah ke program Python untuk menyalakan LED pada rak komponen dan simulasi rak komponen yang akan dijelaskan lebih lanjut pada pemrograman Python. PHP dan MySQL tidak dapat berjalan sendiri karena membutuhkan sebuah servis yang bekerja melayani permintaan dari HTTP *client* atau *web browser*.



Gambar 7 Blok Diagram Desain Halaman Web

F. Desain Database

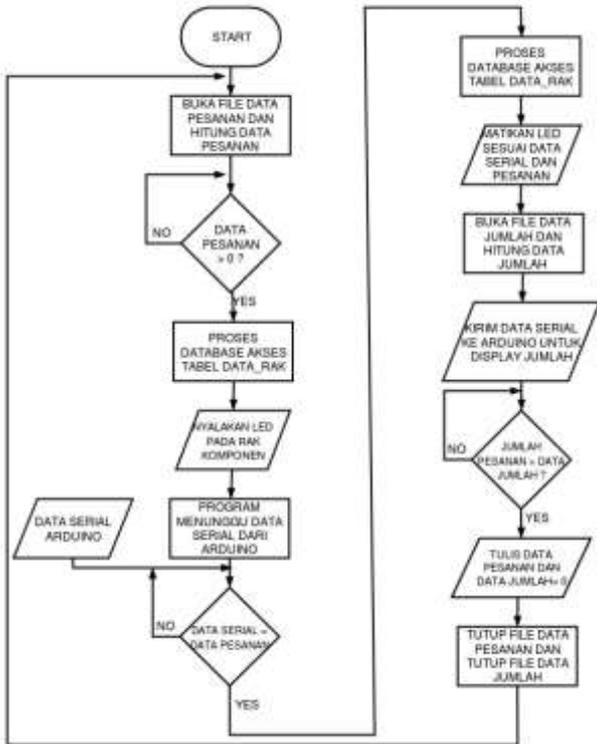
Database MySQL digunakan untuk menyimpan data komponen pada rak, data kategori komponen, data kode masing-masing LED pada rak komponen dan simulasi rak komponen. Gambar 8 adalah *entity relationship diagram*.



Gambar 8 Entity Relationship Diagram

G. Pemrograman Python

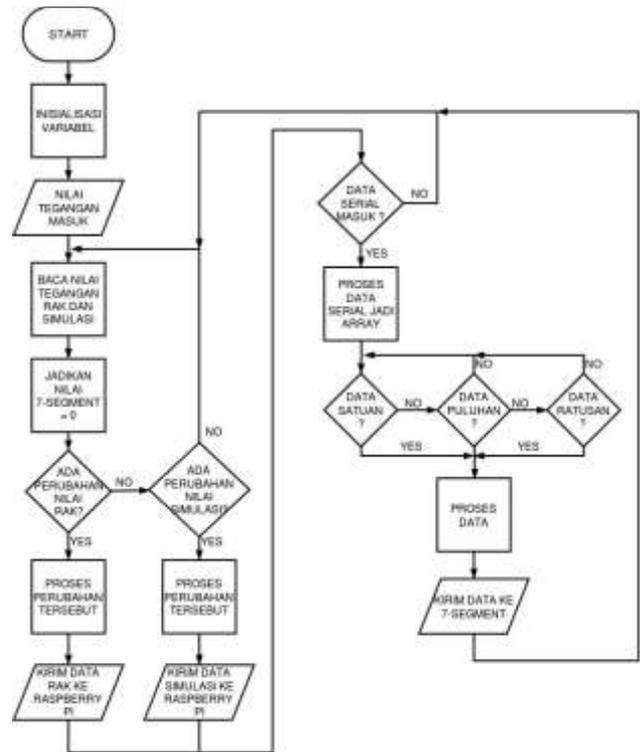
Python digunakan untuk menerima dan mengolah data rak komponen yang dikirim oleh PHP dalam bentuk file.txt. Desain alur program dapat dilihat pada flowchart Gambar 9.



Gambar 9 Flowchart Pemrograman Python

H. Pemrograman Arduino

Program Arduino pada proyek ini bertujuan untuk menerima nilai tegangan dari modul resistor, mengolah hasilnya menjadi data ADC, hasil yang telah diubah menjadi data ADC akan dikirim ke Raspberry Pi 3 melalui komunikasi serial. Gambar 10 adalah flowchart dari program Arduino.



Gambar 10 Flowchart Pemrograman Arduino

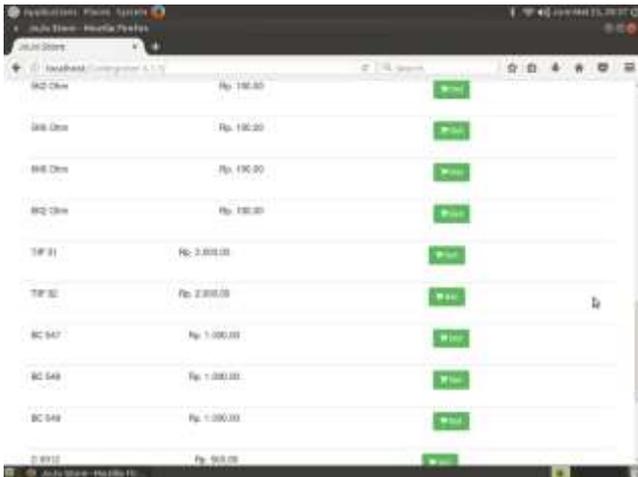
III. PENGUJIAN SISTEM

Sistem ini diuji untuk membuktikan halaman web bekerja dengan baik dan kehandalan program yang telah dibuat pada tugas akhir ini, selain itu kehandalan dari hardware juga diuji pada bab ini. Pengujian yang dilakukan adalah pengujian halaman web dan pengujian program kontrol.

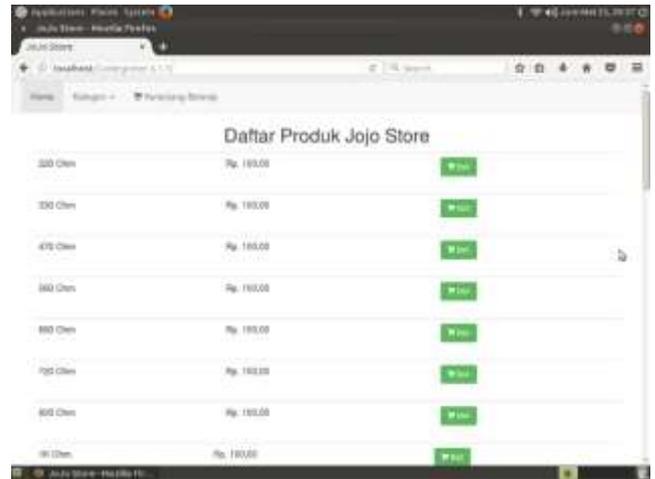


A. Pengujian Halaman Web Daftar Komponen

Pengujian pertama pada proses antar muka dilakukan dengan melihat halaman web utama yang menunjukkan seluruh daftar komponen. Pengujian ini dilakukan untuk membuktikan bahwa seluruh data tabel_gudang pada database MySQL ditampilkan. Gambar 11 membuktikan bahwa data seluruh tabel_gudang ditampilkan dihalaman web utama



Gambar 11 Daftar komponen pada database tabel_gudang



Gambar 12 Hasil Kategori Pada Daftar Komponen

Tabel 2 Pengujian Program Kontrol

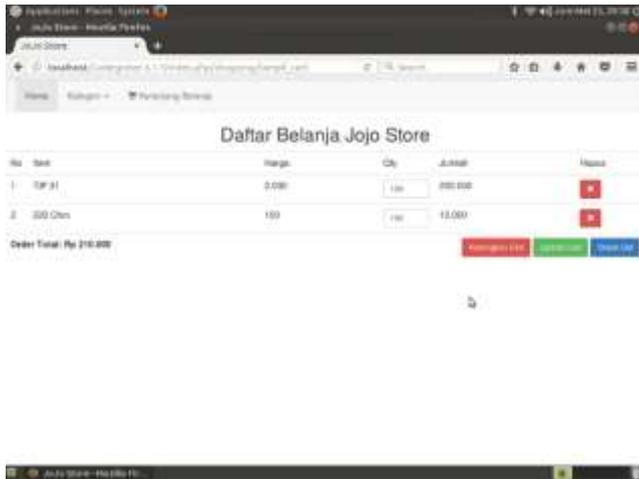
Percobaan	Urutan LED <i>drawer</i> yang menyala	Rak yang ditarik / tombol yang ditekan	Data pada serial <i>monitor</i> Arduino	Urutan LED <i>drawer</i> yang mati	Status
1	1	1	1	1	Berhasil
2	2	2	2	2	Berhasil
3	3,5,7,9,10,15,16	3,5,7,9,10,15,16	3,5,7,9,10,15,16	3,5,7,9,10,15,16	Berhasil
4	4,7,8,9,10,15,17,20	4,7,8,9,10,15,17,20	4,7,8,9,10,15,17,20	4,7,8,9,10,15,17,20	Berhasil
5	25,26,27,28,29	25,26,27,28,29	25,26,27,28,29	25,26,27,28,29	Berhasil
6	24	24	24	24	Berhasil
7	12,13,15,17	12,13,15,17	12,13,15,17	12,13,15,17	Berhasil
8	1,2,3,4,5,6,7,8,9	1,2,3,4,5,6,7,8,9	1,2,3,4,5,6,7,8,9	1,2,3,4,5,6,7,8,9	Berhasil
9	10,11,12,13,14,15	10,11,12,13,14,15	10,11,12,13,14,15	10,11,12,13,14,15	Berhasil
10	16,17,18,19,20	16,17,18,19,20	16,17,18,19,20	16,17,18,19,20	Berhasil

B. Pengujian Halaman Web Kategori Komponen

Pengujian kategori komponen dilakukan dengan interaksi memilih kategori yang ditampilkan antara pengguna dengan halaman web. Kategori yang disediakan pada pembuatan proyek ini terdiri dari transistor dan resistor. Tujuan dari pengujian ini dilakukan untuk mengetahui program PHP dapat berinteraksi dengan tabel_gudang dan tabel_kategori sehingga daftar komponen yang ditampilkan oleh halaman web sesuai dengan permintaan pengguna. Gambar 12 adalah hasil interaksi pengguna dengan kategori komponen.

C. Pengujian Halaman Keranjang Belanja

Pengujian ini dilakukan dengan melakukan pengeditan pesanan pada halaman keranjang belanja dengan membuang komponen yang tidak jadi dipesan atau dengan mengedit jumlah komponen pada kolom jumlah komponen. Pengujian ini dilakukan untuk mengetahui program PHP dengan melihat respon tampilan dari total harga, jumlah komponen, dan daftar komponen apakah sesuai dengan yang diatur oleh pengguna. Pengeditan dilakukan oleh pengguna dengan mengatur daftar komponen yang telah dipesan pada halaman web keranjang belanja lalu klik tombol *update cart*



Gambar 13 Daftar Komponen yang Dipesan Pada Halaman Keranjang Belanja

D. Pengujian Program Kontrol

Pengujian ini bertujuan untuk mengetahui apakah seluruh program berjalan dengan baik mulai dari ketika program Python menerima daftar pesanan komponen untuk menyalakan LED pada rak komponen atau simulasi rak komponen dan menunggu respon dari Arduino melalui komunikasi serial sehingga program Python dapat mengetahui respon yang terjadi dan mematikan LED pada rak komponen atau simulasi rak komponen sesuai data yang diterima.

Selain dari pengujian sisi program Python dan Arduino, pengujian ini juga menguji kehandalan dari *hardware* yang terhubung dengan program. Untuk membuktikan kehandalan program Python dan Arduino dalam menerima, mengolah dan mengirimkan data dilakukan sepuluh kali (10x) percobaan pengiriman data yang dapat dilihat pada Tabel 2.

Tabel 3 Hasil Pengujian Keseluruhan Sistem

Percobaan Pemesanan Komponen					
Percobaan	Jumlah komponen yang dipesan	Urutan LED yang menyala pada <i>drawer</i>	Urutan <i>drawer</i> yang ditarik	Jumlah komponen 7-Segment	Status
1	10,20,30,40,50	1,2,3,4,5	1,5,2,3,4	10,50,20,30,40	Berhasil
2	100	1	1	100	Berhasil
3	1	5	5	1	Berhasil
4	30,40,50	1,2,3	2,3,1	40,50,30	Berhasil
5	100,200,300	6,9,10	10,9,6	300,300,300	Berhasil
6	1,10,100	13,14,15	13,14,15	1,10,100	Berhasil
7	1,12,15,150	8,9,10,11	11,10,9,8	150,150,150,150	Berhasil
8	1,2,22,33	1,5,6,10	10,1,5,6	33,1,2,22	Berhasil
9	200,100,150	1,2,3	1,2,3	200,200,200	Berhasil
10	1,2,3	1,2,3	1,2,3	1,2,3	Berhasil

E. Pengujian Keseluruhan Sistem

Pengujian ini meliputi dari antar muka halaman web sampai pada pemesanan komponen dan pengambilan komponen pada rak komponen. Untuk membuktikan kestabilan program kontrol dan hardware dalam menerima dan mengolah data pesanan, dilakukan pengujian sebanyak sepuluh kali (10x) percobaan pemesanan komponen. Hasil dari percobaan ini dapat dilihat pada Tabel 4.3.

IV. KESIMPULAN

Dari seluruh hasil pembuatan, perakitan, dan pengujian alat yang dibuat pada proyek ini didapatkan beberapa kesimpulan, diantaranya adalah:

1. Daftar komponen yang ditampilkan pada halaman web sesuai dengan data pada *database*.
2. LED pada *drawer* yang menyala telah sesuai dengan *database* yang diakses oleh program Python. Pada Tabel 2, telah membuktikan pengujian program control karena data yang dikirim dan diterima telah sesuai dengan nomor dari rak komponen.
3. Dari hasil pengujian keseluruhan sistem pada Tabel 3, dapat disimpulkan bahwa program PHP, Python dan Arduino telah berkomunikasi dengan baik karena program Arduino secara tepat mengirim data posisi *drawer* dan program Python dapat mengetahui jumlah komponen sesuai pesanan. Sistem berjalan dengan baik yang ditandai dengan status berhasil pada kolom keenam.

V. REFERENSI

- [1] J. Chistiani, "Analisis perbedaan kepuasan konsumen pada pembelian barang elektronik antara toko khusus dan hypermarket di Surabaya," Widya Mandala Catholic University Surabaya Repository, Surabaya, 2014.
- [2] 3DFacture, "Kickstarter," 8 March 2017. [Online]. Available: <https://www.kickstarter.com/projects/3dfacture/chipse-asy-smart-electronic-component-organizer/updates>.

- [3] YouTube, "[D.I.Y.] How to Store and Organise Electronic Parts (Arduino Project)," 5 May 2015. [Online]. Available: <https://www.youtube.com/watch?v=BSGmdM2NbQk>