

## Prototype Smart Minimarket

Taufik Akbar Wibowo <sup>1</sup>, Rahmi Eka Putri, MT <sup>2</sup>

<sup>1,2</sup> Sistem Komputer Fakultas Teknologi Informasi Universitas Andalas Jl. Universitas Andalas, Limau Manis, Pauh, Kota Padang, Sumatera Barat

### INFORMASI ARTIKEL

Received: February 00, 00  
 Revised: March 00, 00  
 Available online: April 00, 00

### KATA KUNCI

*NodeMCU, minimarket, motor DC, Light Dependent Resistor, Mobile Vision API*

### KONTAK PERSONAL

Phone: +62 81275981780  
 E-mail: [taufikakbarwibowo@outlook.com](mailto:taufikakbarwibowo@outlook.com)

### A B S T R A K

Penelitian ini bertujuan untuk membangun sistem *minimarket* yang dapat bekerja dengan otomatis dengan memanfaatkan *smartphone* sebagai media pemesanannya. Sistem ini bertujuan mempermudah pelanggan dalam membeli barang tanpa harus repot membawa keranjang belanja, dan juga mencegah serta mengurangi ancaman pencurian. Sistem ini terdiri atas tiga bagian yaitu aplikasi *mobile*, *Prototype* alat, dan aplikasi PC. Aplikasi *mobile* digunakan oleh pelanggan untuk memesan barang dengan cara memindai *barcode* barang. *Prototype* alat terdiri dari komponen berupa NodeMCU sebagai inti pemrosesan, motor DC dan *motor driver* sebagai penggerak barang dan keranjang, serta laser diode dan sensor cahaya sebagai pendeteksi barang dan keranjang yang lewat. Dari penelitian yang telah dilakukan aplikasi PC dapat memesan barang secara sukses dengan persentase 100% dan dapat memindai *barcode* dengan kecepatan minimum 0.54 detik pada jarak maksimum 60cm pada *barcode* 3.5cm x 3.5cm. Kemudian *prototype* alat dapat membaca data pemesanan secara sukses dengan persentase 100% dan memproses data pemesanan secara sukses dengan persentase 100%. Terakhir aplikasi PC dapat membaca informasi pemesanan melalui masukan nomor keranjang secara sukses dengan persentase 100%.

### PENDAHULUAN

*Minimarket* sebagai tempat berbelanja yang praktis menjadi salah satu tujuan bagi masyarakat untuk membeli barang. Lokasinya yang terjangkau dan tersebar luas merupakan salah satu nilai positif *minimarket* dibandingkan dengan *supermarket*. Dengan menyediakan cukup banyak variasi barang belanjaan untuk kebutuhan sehari-hari yang biasanya diperlukan dengan cepat, *minimarket* menjadi pilihan yang efisien untuk berbelanja barang-barang tersebut.

*Minimarket* dengan sistem yang diterapkan saat ini memiliki beberapa permasalahan. Salah satu permasalahan tersebut yaitu penggunaan keranjang yang merepotkan. *Minimarket* biasanya menyediakan keranjang untuk menampung barang belanjaan yang akan dibeli. Namun, penggunaan keranjang merepotkan karena pelanggan harus membawa barang belanjaan ketika berkeliling. Permasalahan lainnya yaitu adanya ancaman pencurian. Menurut sumber yang didapat, dalam tiga bulan terdapat empat pencurian yang terjadi di wilayah hukum Polda Metro Jaya[1]. Hal ini menjadikan *minimarket* sebagai target empuk para pelaku pencurian.

Beberapa pendekatan telah dilakukan untuk memberi sebagian solusi untuk masalah diatas. Panasonic mengembangkan sebuah sistem bernama Reji Robo yang berfungsi sebagai alternatif kasir. Barang-barang yang dibeli oleh pelanggan didata oleh alat

embedded *barcode* scanner pada setiap keranjang. Dengan sistem ini, kasir nantinya akan memproses pembayarannya saja tanpa harus ada campur tangan manusia[2]. Adapun kelemahan pada sistem ini yaitu pembuatan perangkat yang mahal karena harus dipasangkan pada setiap keranjang yang tersedia. Pendekatan lain yang diajukan pada penelitian yang telah dilakukan sebelumnya yaitu berupa aplikasi *minimarket* berbasis *web*[3]. Pendekatan berupa aplikasi berbasis *web* ini bisa dikatakan cukup bagus, namun penerapannya kurang maksimal karena belum memiliki sistem yang bisa mengambil barang pesanan secara otomatis. Android merupakan sistem operasi yang perkembangannya sangat masif dan cepat. Saat ini android sudah identik dengan *smart phone*. Perkembangan aplikasi di android juga sangat cepat, bahkan tahun 2010 tiap bulan ada lebih dari 10 ribu aplikasi ditambahkan untuk android. **Error! Reference source not found.**[5]. Android memiliki fitur pada kamera yang dapat digunakan untuk fungsi-fungsi tertentu. Salah satu fungsi kamera tersebut yaitu dapat melakukan pemindaian *barcode*.

Berdasarkan permasalahan diatas, penulis memiliki ide untuk membangun sebuah sistem pada *minimarket* yang terintegrasi dengan aplikasi Android dimana sistem bisa melakukan pemesanan barang dan memproses data pesanan. Sistem ini dapat menghilangkan masalah keranjang yang merepotkan, meminimalisir ancaman dan kerugian atas pencurian, dan menunjang performa *minimarket* dengan sistem 24 jam. Oleh

karena itu, penulis ingin mengajukan sebuah proposal penelitian tugas akhir dengan topik “*Prototype Smart Minimarket.*”

## STUDI PUSTAKA

### Minimarket

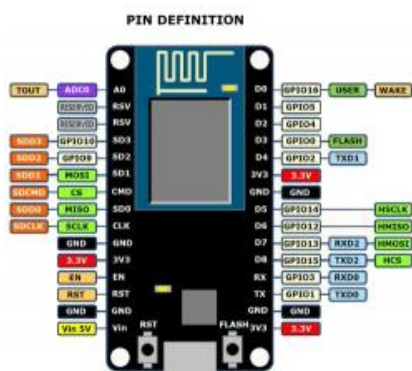
*Minimarket* merupakan salah satu sarana perbelanjaan umum untuk membeli barang kebutuhan sehari-hari. Perbedaan antara *supermarket* dengan *minimarket* terdapat pada ukuran tempat dan jumlah pasokan barang yang disediakan. *Supermarket* memiliki ukuran tempat yang luas dan besar sehingga dapat memasok barang dalam jumlah yang besar. Tetapi, *supermarket* hanya dapat ditemukan pada pusat perbelanjaan. Akses terhadap *supermarket* dinilai cukup rendah karena kurang efisien untuk berbelanja barang kebutuhan harian dalam kuantitas kecil dimana bisa disediakan oleh *minimarket*. *Minimarket* itu sendiri memiliki ukuran tempat yang relatif kecil dan sempit. Dengan karakteristik yang dimiliki oleh *minimarket* menyebabkan pasokan barang yang dapat disediakan berjumlah kecil. Meskipun demikian, *minimarket* memiliki sisi keuntungan tersendiri yaitu lokasi bangunan. *Minimarket* umumnya dapat ditemukan pada lokasi strategis seperti sekitar perumahan, perkantoran, dan tempat keramaian. Lokasi yang strategis menjadikan *minimarket* sebagai tujuan untuk berbelanja kebutuhan harian yang lebih efisien bagi masyarakat dibandingkan *supermarket*.

### NodeMCU

NodeMCU merupakan *open-source firmware* dan *development kit* yang membantu kita untuk membuat *prototype* produk IoT (*Internet of Things*) melalui beberapa baris *script* Lua[5]. NodeMCU menerapkan chip ESP8266 dari Espressif yang memberikan solusi integrasi *Wi-Fi* SoC untuk memenuhi kebutuhan *user* yang berkelanjutan untuk penggunaan daya yang efisien, desain yang pas dan performa yang handal dalam industri *Internet of Things*[7]. Pemrograman pada NodeMCU dapat ditulis dengan bahasa pemrograman Lua maupun C. Adapun untuk bahasa pemrograman C, dapat digunakan tools Arduino IDE sebagai media penulisan, pengompilasian, dan pengunggahan program ke NodeMCU.



Gambar 1 NodeMCU[5]



Gambar 2 Keterangan Pinout NodeMCU[8]

Tabel 1 Spesifikasi ESP8266-12E[7][9]

Categories	Items	Values
WiFi Parameters	WiFi Protocols	802.11 b/g/n
	Frequency Range	2.4GHz-2.5GHz
Hardware Parameters	Peripheral Bus	UART/HSPI/I2C/I2S/Ir Remote Control GPIO/PWM
	Operating Voltage	3.0V~3.6V
	Operating Current	Average Value: 80mA
	Operating Temperature Range	-40°C~125°C
	Ambient Temperature Range	Normal Temperature
	Package Size	16mm*24mm*3mm
	External Interface	N/A
	Processor	Tensilica L106 32-bit RISC
	Memory	RAM < 50 kB, Flash up to 16 MB
	ADC	10-bit (1 pin)
Software Parameters	WiFi mode	Station/softAP/softAP+station
	Security	WEP/WPA/WPA2
	Encryption	TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network) / download and write firmware via host
	Software Development	Support Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
User Configuration	AT Instruction Set, Cloud Server, Android/iOS App	

### Motor DC

Motor DC atau sering disebut motor arus searah lebih sering digunakan untuk keperluan yang membutuhkan pengaturan kecepatan dibandingkan dengan mesin AC. Alasan utama penggunaan mesin DC terutama pada industri-industri modern adalah karena kecepatan kerja motor-motor DC mudah diatur dalam suatu rentang kecepatan yang luas, disamping banyaknya metode-metode pengaturan kecepatan yang dapat digunakan[10].

### Motor driver L298N

L298N adalah contoh IC yang dapat digunakan sebagai *driver motor* DC. IC ini menggunakan prinsip kerja H-Bridge. Tiap H-Bridge dikontrol menggunakan level tegangan TTL yang berasal dari *output* mikrokontroler. L298N dapat mengontrol 2 buah motor DC. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 46 Vdc dan arus mencapai 2 A untuk setiap kanalnya Error! Reference source not found.

### Laser Diode

Laser diode merupakan salah satu komponen elektronik yang berfungsi memancarkan cahaya. Berbeda dengan LED (*Light Emitting Diode*) yang pancaran cahayanya menyebar, laser diode memancarkan cahaya laser yang pancarannya lurus menuju satu titik. Dengan pancaran cahaya yang terfokus pada satu titik, tingkat intensitas cahaya yang dihasilkan relatif lebih tinggi dari cahaya LED[12]. Laser diode memiliki sifat yang sama dengan

cahaya pada umumnya yaitu tidak dapat menembus objek yang tidak tembus pandang.

### Sensor Cahaya LDR

Sensor cahaya LDR (*Light Dependent Resistor*) adalah salah satu jenis *resistor* yang dapat mengalami perubahan resistansinya apabila mengalami perubahan penerimaan cahaya. Besarnya nilai hambatan pada sensor cahaya LDR tergantung pada besar kecilnya cahaya yang diterima oleh LDR itu sendiri. LDR sering disebut dengan alat atau sensor yang berupa resistor yang peka terhadap cahaya [13].

### ADC ADS1115

ADC (*Analog Digital Converter*) merupakan perangkat elektronik yang dapat memproses nilai tegangan analog menjadi nilai digital. Nilai digital yang didapat dari ADC bisa diproses oleh mikrokontroler menjadi sebuah data. ADS1115 merupakan modul ADC yang dikeluarkan oleh Adafruit. Perangkat ini dapat digunakan pada mikrokontroler yang memiliki akses terbatas terhadap ADC.

Tabel 2 Spesifikasi ADS1115 [14]

Resolution	16 Bits
Programmable Sample Rate	8 to 860 Samples/Second
Power Supply/Logic Levels	2.0V to 5.5V
Low Current Consumption	Continuous Mode: Only 150µA Single-Shot Mode: Auto Shut-Down
Low-Drift Voltage Reference	Internal
Oscillator	Internal
Internal PGA	up to x16
I2C Interface	4-Pin-Selectable Addresses
Mode	Four Single-Ended or 2 Differential Inputs
Comparator	Programmable

### Web server Apache

*Web server* merupakan sebuah *software* yang bertugas menerima permintaan HTTP dari *client*, yang dikenal sebagai *Web browsers*, dan melayani mereka dengan respon HTTP dengan tambahan konten data opsional, dimana biasanya berupa halaman *web* seperti dokumen HTML dan objek yang terhubung (gambar, dll) [15]. *Web server* juga bisa disebut sebagai komputer yang berfungsi menangani permintaan untuk memberikan dan menampilkan halaman *web* pada *client web browser*. *Web browser* itu sendiri merupakan aplikasi yang berfungsi sebagai perantara pengguna dengan *web server* dimana *web browser* bekerja menampilkan halaman *web* yang diminta oleh pengguna ke *web server*. Salah satu perangkat lunak yang menyediakan layanan *web server* yaitu Apache. Apache dikembangkan oleh ASF (Apache Software Foundation) dan merupakan perangkat lunak *cross platform* yang dapat digunakan di hampir berbagai platform *device*.

### Database MySQL

*Database* merupakan media penyimpanan data yang terstruktur dan tersimpan secara sistematis di dalam komputer. *Database*

menyimpan data ke dalam bentuk *record* yang tersimpan dalam tabel. Untuk dapat mengelola data pada *database* diperlukan perangkat lunak yang disebut DBMS (*Database Management System*). Salah satu perangkat lunak yang berfungsi sebagai DBMS yaitu MySQL. *Database MySQL* merupakan suatu perangkat lunak *database* yang berbentuk *database* relasional atau disebut *Relational Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (*Structured Query Language*) [16]. Selain itu, MySQL juga dapat mengatur izin akses *database* pada *user* yang sudah terdaftar sehingga data lebih aman.

### Arduino IDE

Arduino IDE (*Integrated Development Environment*) merupakan perangkat lunak *open source* yang dikembangkan oleh perusahaan penyedia *software tools* dan *hardware platforms*, Arduino. Arduino IDE digunakan untuk menulis, mengompilasi, dan meng-upload kode program yang akan digunakan pada mikrokontroler. Arduino IDE menggunakan bahasa C sebagai bahasa pemrogramannya. Meskipun dikembangkan oleh Arduino, tools ini mendukung sebagian besar mikrokontroler lain, salah satunya yaitu chip ESP8266.

### Android Studio

Android Studio merupakan sebuah *tools* yang digunakan untuk membuat aplikasi Android. Android Studio merupakan perangkat lunak yang awalnya dikembangkan oleh JetBrains bernama IntelliJ IDEA. Google melakukan akuisisi terhadap *tools* untuk dikomersilkan dengan Android kepada para pengembang supaya para pengembang dapat mengembangkan aplikasi Android secara praktis dan mudah. Android Studio memberikan beberapa kelebihan dari *tools* lain untuk mengembangkan aplikasi Android.

### Mobile Vision API

*Mobile Vision API* merupakan API (*Application Programming Interface*) yang dikembangkan oleh Google dan ditargetkan untuk pemrograman aplikasi Android. *Mobile Vision API* memberikan fitur layaknya *image processing* pada umumnya, namun lebih berfokus pada fungsi-fungsi tertentu. Karena hanya ditargetkan untuk Android, *Mobile Vision API* tidak dapat diterapkan pada platform OS selain Android. *Mobile Vision API* memiliki tiga fungsi utama yaitu *detect faces*, *scan barcodes*, dan *recognize text*.

*Scan barcodes* disebut dengan *Barcode Scanner API* yang berfungsi mendeteksi *barcode* secara *realtime* pada orientasi apapun. Kita juga bisa mendeteksi dan mengartikan beberapa *barcode* dalam format yang berbeda pada waktu yang sama [17]. *Barcode* itu sendiri terdiri atas dua macam yaitu *barcode* satu dimensi dan *barcode* dua dimensi. *Barcode* satu dimensi merupakan *barcode* yang dibentuk secara horizontal ataupun vertikal. Adapun jenis-jenis pada *barcode* satu dimensi yaitu EAN-13, EAN-8, UPC-A, UPC-E, Code-39, Code-93, Code-128, ITF, Codabar [18]. *Barcode* dengan model ini bisa ditemukan pada produk kemasan yang dijual di toko. *Barcode* dua dimensi merupakan *barcode* yang dibentuk sedemikian rupa secara horizontal dan vertikal. Adapun jenis-jenis pada *barcode* dua

dimensi yaitu QR-Code, Data Matrix, PDF-417, AZTEC[18]. *Barcode* dengan model ini biasa ditemukan pada aplikasi *smartphone* berbasis sosial seperti LINE dan WhatsApp.

### Visual Studio

Visual Studio merupakan sebuah IDE (*Integrated Development Environment*) yang dikembangkan oleh Microsoft sebagai *tools* untuk mengembangkan aplikasi pada berbagai *platform* seperti *desktop*, *web*, dan *mobile*. Visual Studio mendukung banyak bahasa pemrograman untuk berbagai *platform* yang telah disebutkan sebelumnya seperti bahasa C, C++, C#, Python, F#, .NET, dan Javascript. Untuk *desktop platform*, pengembang dapat menggunakan bahasa C# dimana bahasa ini merupakan pengembangan bahasa C++ yang sudah terintegrasi dengan GUI (*Graphical User Interface*) dimana pada C++ tidak disediakan secara langsung tanpa pihak ketiga. Selain dukungan bahasa pemrograman yang beragam, Visual Studio juga dilengkapi fasilitas untuk menerapkan *extension* atau *plugin* yang bertujuan untuk menunjang pengembangan aplikasi.

## PERANCANGAN SISTEM

### Analisa Kebutuhan Sistem

Untuk memahami kebutuhan sistem secara keseluruhan akan diuraikan secara fungsional maupun non-fungsional.

#### Kebutuhan Fungsional Sistem

Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem, yaitu

1. Sistem dapat membaca dan memperbarui data pesanan pada *database* secara realtime
2. Sistem dapat mengaktifkan motor DC untuk mengambil barang pesanan
3. Sistem dapat membaca masukan dari sensor cahaya sebagai data acuan untuk *counter* barang yang diambil

#### Kebutuhan Non-Fungsional Sistem

Kebutuhan non-fungsional adalah kebutuhan yang menitik beratkan pada properti perilaku yang dimiliki oleh sistem, yaitu

1. Tersedianya koneksi jaringan yang stabil dan cepat
2. Tersedianya sumber daya listrik untuk mengaktifkan sistem ini

### Rancangan Proses

Perancangan proses dilakukan dengan menspesifikasikan fungsionalitas sebuah sistem, dimulai dari pemesanan barang oleh *user* pada aplikasi Android, pemrosesan data pesanan oleh sistem, hingga penampilan data pesanan pada kasir.

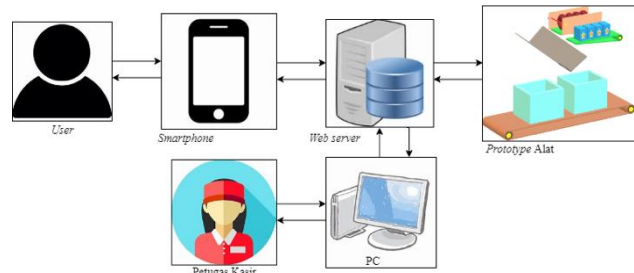
### Perancangan Sistem

Berikut merupakan rancangan sistem yang akan dibuat yang digambarkan pada Gambar 3. Berdasarkan gambar 3, alur proses sistem terdiri atas:

1. *User* mengakses aplikasi pada *smartphone* untuk melakukan pemesanan barang. Pemesanan barang dilakukan dengan membaca *barcode* barang yang ingin dipesan. Aplikasi pada *smartphone* dapat mengakses *web server* untuk membaca

data barang dan menambah data pesanan pada *database*. Jika proses pemesanan telah dilakukan, data pesanan akan dikirim melalui *web server* untuk disimpan pada *database*.

2. *Prototype* alat akan membaca data pesanan dan mengambil barang pesanan sesuai dengan data tersebut. Jika barang pesanan sudah lengkap, maka alat akan memperbarui data *no\_keranjang* pada *database* melalui *web server* dan akan membaca data pesanan berikutnya.
3. PC petugas kasir bertujuan untuk menjalankan aplikasi kasir untuk memproses pembayaran *user* kepada petugas tersebut. Aplikasi kasir akan membaca data pesanan dari *database* melalui *web server* dan menampilkannya sehingga petugas kasir dapat mengetahui data pesanan *user* yang telah selesai.

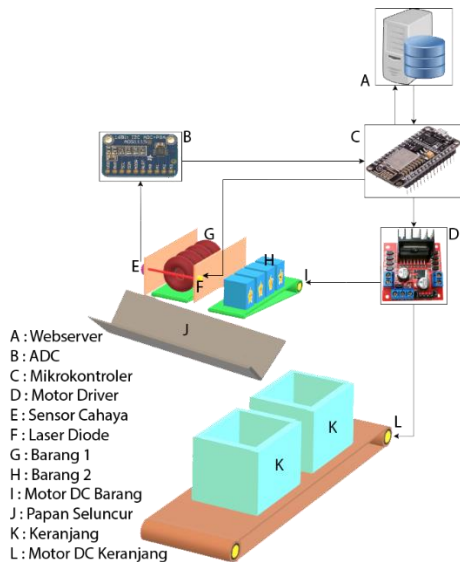


Gambar 3 Blok Diagram Sistem

Adapun gambaran *Prototype* alat dapat dilihat pada gambar 4. Berdasarkan gambar 4, alur kerja alat terdiri atas:

1. Mikrokontroler membaca data pesanan dari *database*.
2. Mikrokontroler menggerakkan “motor DC barang” yang dikendalikan melalui *motor driver* untuk menjalankan “konveyor barang” dengan tujuan menjalankan barang yang sesuai dengan pesanan. Barang pesanan akan dijalankan oleh “motor DC barang” supaya jatuh ke papan seluncur yang mengarah tepat ke keranjang.
3. Laser diode dan sensor cahaya berfungsi mendeteksi barang yang melewati laser tersebut dan digunakan sebagai *counter* barang yang lewat. Mekanisme *counter* bekerja dengan cara mengambil nilai intensitas cahaya laser ketika tidak terhalang dan menjadikannya sebagai *setpoint*. Jika nilai sensor yang dibaca tidak sesuai dengan *setpoint*, maka barang akan dianggap melewati laser dan akan men-trigger *counter*. Pembacaan nilai sensor membutuhkan ADC eksternal untuk memfasilitasi banyak sensor yang dipakai dikarenakan NodeMCU hanya memiliki satu buah pin untuk masukkan data analog. Jika jumlah barang yang diambil sudah sesuai dengan data pesanan, maka mikrokontroler secara otomatis akan mematikan “motor DC barang”.
4. Mikrokontroler menjalankan “motor DC keranjang” untuk menempatkan keranjang berikutnya ke depan papan seluncur untuk pesanan berikutnya. Supaya keranjang berhenti tepat di depan papan seluncur, digunakan laser diode dan sensor cahaya dengan melakukan mekanisme *counter* barang yang lewat seperti yang telah dijelaskan di atas.

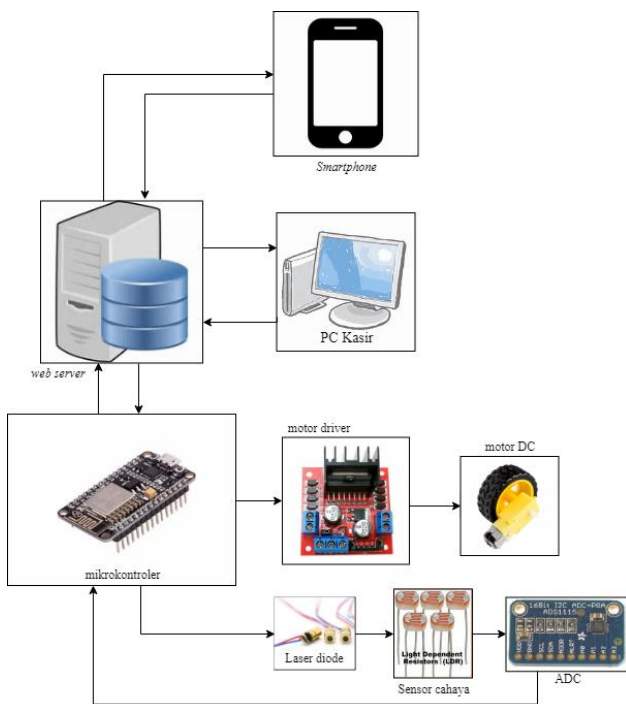




Gambar 4 Gambaran *Prototype* Alat

**Perancangan Perangkat Keras**

Pembentukan dan alur kerja sistem akan dijelaskan pada perancangan perangkat keras yang dapat dilihat pada Gambar 5.



Gambar 5 Perancangan Perangkat Keras

Berdasarkan gambar 5, perancangan perangkat keras terdiri dari:

1. NodeMCU  
NodeMCU merupakan mikrokontroler yang bertugas menjadi inti *Prototype* alat dan mengatur jalan kerja komponen yang memproses pemesanan.
2. Motor driver  
Motor driver berfungsi mengontrol motor DC sesuai dengan masukan yang diterima dari mikrokontroler.
3. Motor DC  
Motor DC pada barang bekerja menjalankan barang pesanan ke papan seluncur yang mengarah ke keranjang, sedangkan motor

DC pada keranjang bekerja menggeser keranjang untuk pemrosesan pesanan berikutnya.

4. Laser Diode

Laser diode berfungsi memancarkan cahaya laser ke sensor cahaya supaya dapat mendeteksi barang yang lewat. Nilai ADC sensor cahaya yang dipancarkan oleh laser diode memiliki nilai tertentu yang menjadi set point sehingga jika cahaya laser tersebut terblokir akan ada perbedaan nilai dan perbedaan nilai ini menjadi acuan untuk *counter* barang yang lewat.

5. Sensor Cahaya

Sensor cahaya berfungsi mengukur intensitas cahaya. Pada sistem, laser diode memancarkan cahaya laser ke sensor cahaya untuk mendeteksi adanya benda yang lewat.

6. ADC

ADC berfungsi memproses data analog menjadi data digital yang dapat diolah oleh mikrokontroler. ADC diperlukan karena NodeMCU hanya memiliki satu buah pin ADC yang mengakibatkan terbatasnya sensor yang dapat dihubungkan.

7. *Smartphone*

*Smartphone* berfungsi sebagai media untuk menjalankan aplikasi pemesanan barang oleh *user*. Aplikasi yang berjalan pada *smartphone* dapat digunakan untuk berkomunikasi dengan *database* melalui *web server*.

8. *Web server*

*Web server* berfungsi sebagai perantara *database* dengan mikrokontroler dan *smartphone* untuk dapat bertukar data. Data yang disimpan dan dikirim pada *database* melalui *web server* digunakan untuk mengolah pesanan pada sistem.

9. PC (Personal Computer)

PC berfungsi menjalankan aplikasi kasir yang dapat menerima data pesanan pada *database*, menampilkan barang yang dipesan berdasar data pesanan, dan menampilkan total harga pesanan.

**Perancangan Perangkat Lunak**

Pada perancangan perangkat lunak terdapat beberapa poin seperti perancangan *database*, perancangan perangkat lunak *user*, perancangan perangkat lunak sistem, perancangan perangkat lunak kasir, dan perancangan *user interface*.

**Perancangan Database**

Pada *database* terdapat tiga buah tabel yang akan digunakan yaitu tabel barang, tabel pesanan, dan tabel info\_pesanan. Tabel barang merupakan tabel yang dirancang untuk menyimpan semua informasi barang. Informasi pada rancangan tabel barang dapat dilihat pada tabel 3. Pada rancangan tabel barang terdapat field *id\_barang*, nama, jumlah, dan url. Field *id\_barang* sebagai primary key menunjukkan id dari barang, field nama menunjukkan nama barang, field jumlah menunjukkan informasi jumlah stok barang, field url merupakan informasi halaman *web* yang merujuk pada gambar dari masing-masing barang, dan field harga menunjukkan informasi harga barang.

Tabel 3 Rancangan Tabel Barang

id_barang	Nama_barang	Jumlah	url	harga
<i>varchar</i> (10)	<i>varchar</i> (20)	<i>int</i> (1)	<i>varchar</i> (50)	<i>int</i> (6)

Tabel pelanggan merupakan tabel yang dirancang untuk menyimpan semua data pesanan yang bisa diproses oleh sistem. Informasi rancangan tabel pelanggan dapat dilihat pada tabel 4. Field *id\_user* merupakan informasi mengenai nomor pesanan yang masuk ke tabel pelanggan dan digunakan untuk mencari

data pesanan yang sesuai serta dijadikan sebagai nomor antrian pelanggan. Field *data\_user* merupakan informasi mengenai data pesanan. Informasi pesanan hanya disimpan dalam bentuk angka untuk dapat diproses oleh sistem nantinya. Tanggal merupakan kolom untuk informasi waktu pesanan masuk. Terakhir yaitu field *no\_keranjang* yang menunjukkan informasi nomor pesanan yang sedang/akan diproses.

**Tabel 4 Rancangan Tabel Pelanggan**

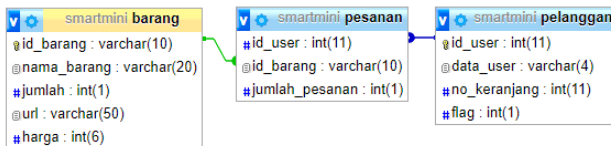
id_user	data_user	no_keranjang	flag	Tanggal
int(11)	varchar(4)	int(11)	int(1)	date

Tabel pesanan merupakan tabel yang berisi info pesanan secara rinci sesuai dengan *id\_user* yang bersangkutan. Tabel pesanan akan berelasi dengan tabel pelanggan dan tabel barang supaya informasi lengkap dapat diambil ketika PC kasir mengakses data pada tabel ini. Disini terdapat field *id\_user* sebagai primary key berisi nomor antrian pelanggan, field *id\_barang* yang berisi *id\_barang* yang dipesan, dan field *jumlah\_pesanan* yang berisi jumlah barang yang dipesan oleh *user* tersebut. Tabel pesanan digunakan untuk menampilkan informasi pada aplikasi kasir. Rancangan tabel pesanan dapat dilihat pada tabel 5.

**Tabel 5 Rancangan Tabel Pesanan**

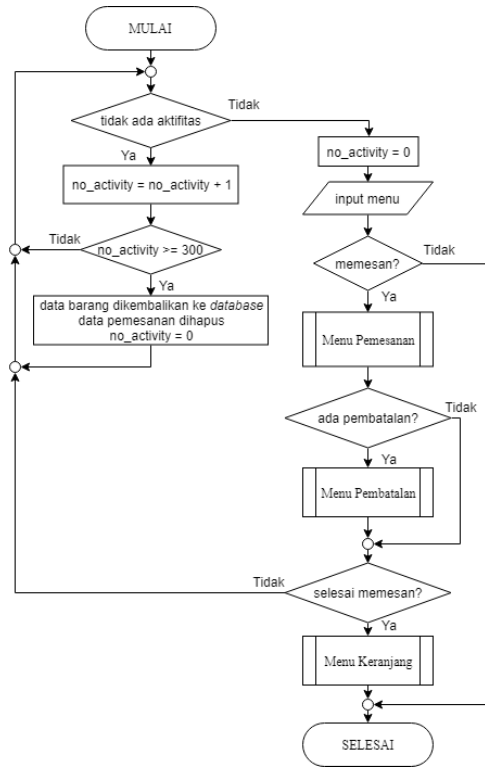
id_user	id_barang	jumlah_pesanan
int(11)	varchar(10)	int(1)

Adapun bentuk relasi dari ketiga tabel dapat dilihat pada gambar 3.4.



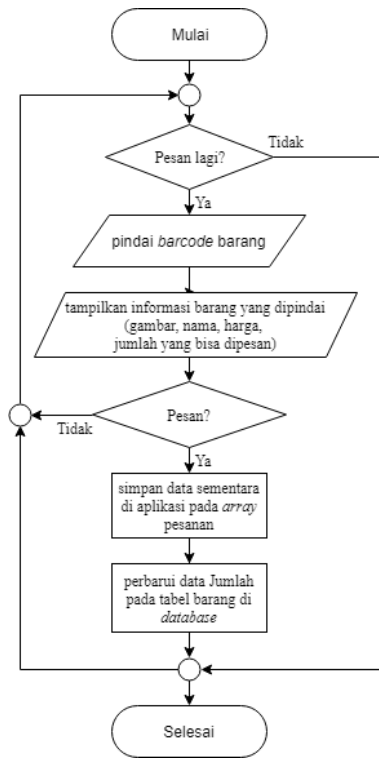
**Gambar 6 Relasi Antar Tabel**

**Perancangan Perangkat Lunak User**

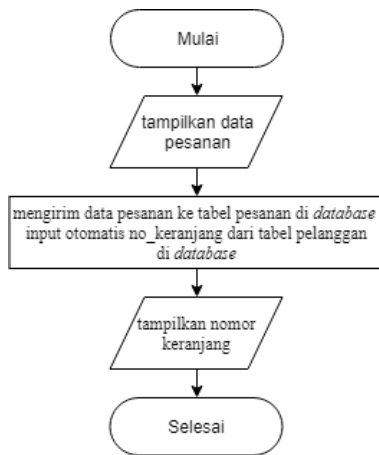


**Gambar 7 Flowchart User**

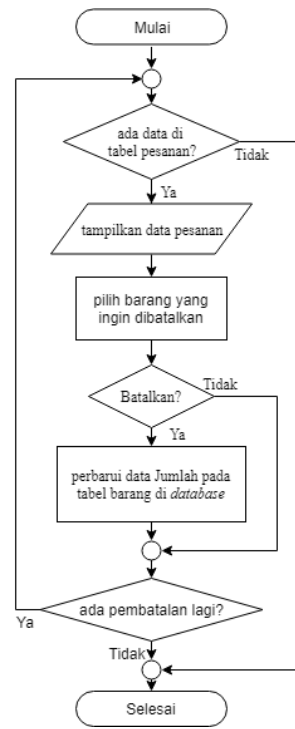
Perancangan perangkat lunak *user* menggambarkan tahapan yang dilalui oleh *user* untuk dapat menggunakan sistem. Adapun *flowchart* perangkat lunak *user* dapat dilihat pada gambar 7. Berdasarkan gambar 7, alur proses *user* pada aplikasi dimulai dengan menu pemesanan. Kemudian jika *user* ingin membatalkan pesanan maka akan dilakukan menu pembatalan. Jika pesanan sudah siap, maka akan dilanjutkan ke menu keranjang. Adapun penjelasan dari prosedur yang terdapat pada gambar 7 *flowchart* dapat dilihat pada gambar 8, 9, dan 10.



Gambar 8 Flowchart Menu Pemesanan



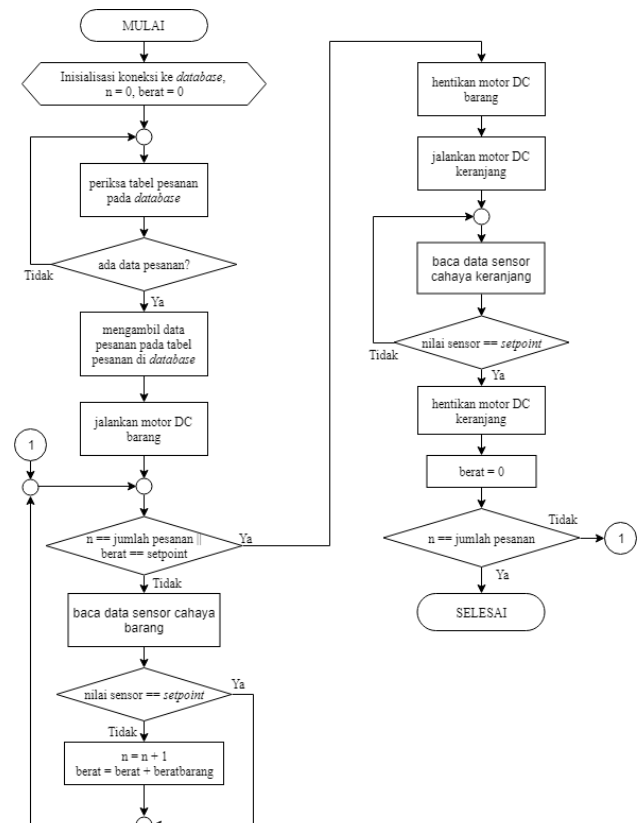
Gambar 9 Flowchart Menu Keranjang



Gambar 10 Flowchart Menu Pembatalan

**Perancangan Perangkat Lunak Sistem**

Perancangan perangkat lunak sistem menggambarkan tahapan proses kerja sistem. Adapun flowchart perangkat lunak sistem dapat dilihat pada gambar 11



Gambar 11 Flowchart Sistem Tertanam NodeMCU

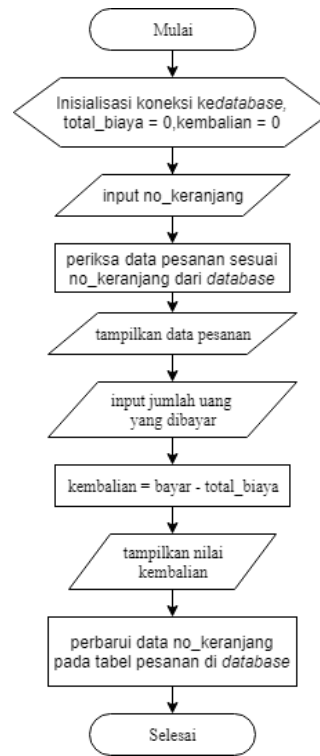
Berdasarkan gambar 11, alur kerja sistem dapat dijelaskan sebagai berikut

1. Inisialisasi koneksi ke *database* melalui *web server* dan set *variable* *n* bernilai 0. *Variable* *n* digunakan untuk membandingkan jumlah barang yang dipesan dengan barang yang sudah dijatuhkan.
2. Periksa data pada tabel pelanggan di *database* melalui *web server*. Jika tidak ada data baru, maka proses pemeriksaan akan dilakukan kembali. Namun, jika terdapat data baru maka sistem akan mengambil data baru tersebut dan akan memprosesnya. Setelah itu, sistem akan mengaktifkan motor DC barang yang bertujuan untuk menjatuhkan barang yang dipesan ke dalam keranjang.
3. Menjatuhkan barang dimana proses memiliki kondisi yang memungkinkan sistem untuk dapat menghitung barang yang telah dijatuhkan dengan nilai *n*. Jika nilai *n* belum sesuai dengan jumlah pesanan maka akan dilakukan looping pengecekan sensor cahaya. Jika sensor cahaya memberikan nilai yang tidak sesuai dengan *setpoint*, maka hal tersebut menandakan bahwa ada barang yang lewat sehingga nilai *n* akan bertambah satu. Pengecekan sensor berlanjut hingga jumlah pesanan sesuai dengan nilai *n* dan proses pengecekan jumlah pesanan akan berhenti dan motor DC barang akan dihentikan. Proses menjatuhkan barang berhenti jika *n* sudah sesuai jumlah pesanan atau total berat barang yang dijatuhkan sudah mencapai *setpoint*.
4. Mengaktifkan motor DC keranjang yang bertujuan menggeser keranjang untuk memposisikan keranjang berikutnya ke depan papan seluncur dan mengeluarkan keranjang yang sudah selesai dari konveyor keranjang. Supaya posisi keranjang berada tepat di depan papan seluncur, digunakan kondisi yang sama pada pengecekan barang yang jatuh dengan membaca nilai sensor cahaya. Jika nilai sensor cahaya belum mencapai *setpoint* maka motor DC keranjang belum bisa dihentikan. Namun jika nilai sensor cahaya sudah mencapai *setpoint*, maka motor DC keranjang akan dihentikan. Proses yang terakhir yaitu memperbarui *no\_keranjang* pada tabel pesanan di *database* supaya alat dapat memproses pesanan berikutnya.

**Perancangan Perangkat Lunak Kasir**

Pada perancangan perangkat lunak kasir akan dijelaskan alur kerja dari aplikasi kasir. Adapun *flowchart* perangkat lunak kasir dapat dilihat pada gambar 12.

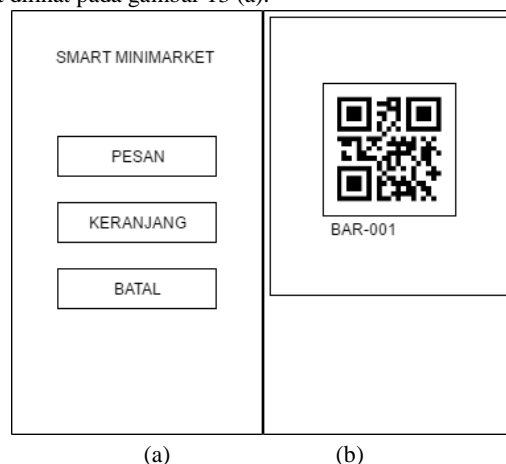
Berdasarkan gambar 12, aplikasi kasir dimulai dengan inisialisasi dan koneksi ke *database*. Kasir akan meng-*input* *no\_keranjang* *user* yang akan diproses dan aplikasi akan menampilkan data pesanan dalam bentuk tabel. Kemudian petugas kasir akan meng-*input*-kan nilai pembayaran yang diberikan untuk diproses supaya bisa diketahui kembalinya. Ketika proses untuk menghitung kembalian selesai, maka nilai kembalian akan ditampilkan untuk dibaca oleh petugas kasir.



**Gambar 12 Flowchart Kasir**

**Perancangan Interface**

Perancangan *interface* menggambarkan rancangan awal *interface* pada perangkat lunak yang terkait dengan sistem. Terdapat dua *interface* yang akan dirancang yaitu *interface user* dan kasir. Pada perancangan *interface user*, terdapat empat halaman utama yaitu halaman awal, halaman pemindaian *barcode*, halaman informasi barang dan halaman keranjang. Halaman awal merupakan halaman pertama yang ditampilkan ketika membuka aplikasi Android. Halaman awal memuat tiga buah tombol yang akan menjalankan proses tersendiri. Tiga tombol tersebut yaitu tombol ‘Pesanan’, ‘Keranjang’, dan ‘Keluar’. Rancangan halaman awal dapat dilihat pada gambar 13 (a).

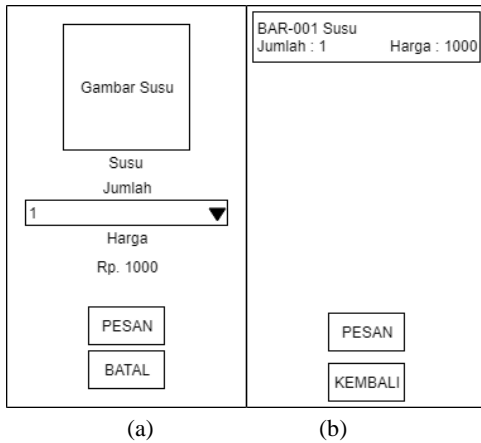


**Gambar 13 (a) Rancangan Halaman Awal, (b) Rancangan Halaman Pemindaian Barcode**

Adapun rancangan halaman pemindaian *barcode* dapat dilihat pada gambar 13 (b). Pada halaman pemindaian *barcode* akan ditampilkan gambar yang berasal dari kamera untuk digunakan sebagai *barcode* scanner. Jika kamera diarahkan pada gambar

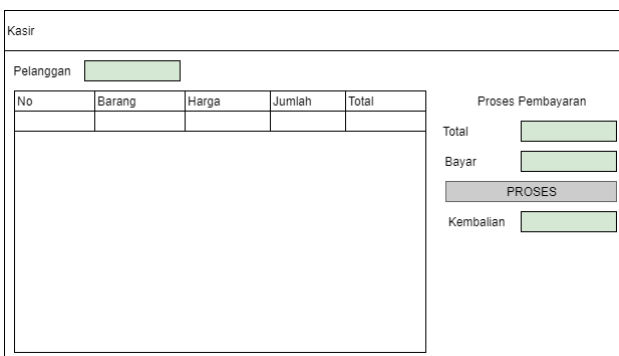


barcode dan aplikasi sukses membaca barcode tersebut, maka user akan secara otomatis dialihkan ke halaman informasi barang. Kemudian rancangan halaman informasi barang dapat dilihat pada gambar 14 (a). Pada halaman informasi barang akan ditampilkan nama barang, gambar barang, serta jumlah barang yang bisa dipesan. Terdapat tombol pesan untuk memesan barang dan batal untuk membatalkan pesanan. Terakhir yaitu rancangan halaman keranjang dapat dilihat pada gambar 14 (b). Pada halaman keranjang, akan ditampilkan data barang-barang sesuai dengan data pesanan. Terdapat tombol pesan untuk mengirimkan data pesanan ke database untuk diproses oleh sistem dan tombol kembali untuk kembali ke halaman awal.



Gambar 14 (a) Rancangan Halaman Informasi Barang, (b) Rancangan Halaman Keranjang

Adapun bentuk rancangan interface kasir dapat dilihat pada gambar 15. Pada gambar 15, di sisi kiri terdapat kolom tabel yang akan memuat seluruh pesanan oleh semua user. Kolom tabel akan secara otomatis terisi jika pesanan user sudah diproses oleh sistem. Pada sisi kanan, ada kolom pelanggan sebagai tempat memasukkan nomor antrian user untuk dapat diproses pembayarannya. Di bawah kolom pelanggan, ada kolom total yang akan menampilkan total harga pesanan user. Kemudian ada kolom bayar untuk memasukkan berapa jumlah uang yang dibayarkan user kepada kasir. Ketika kolom bayar sudah dimasukkan, kasir dapat menekan tombol selesai dan kolom kembalian akan menampilkan nilai kembalian yang akan dikembalikan kepada user.



Gambar 15 Rancangan Interface Kasir

## HASIL DAN ANALISA

### Implementasi

Implementasi Prototype Smart Minimarket dapat dilihat pada gambar 16.



Gambar 16 Implementasi Prototype Smart Minimarket

Implementasi perangkat keras terdiri dari komponen yang meliputi NodeMCU, sensor cahaya, laser diode, ADC, motor driver, dan motor DC. Implementasi perangkat keras terdiri atas tiga bagian yaitu bagian konveyor barang, bagian konveyor keranjang, dan papan seluncur. Konveyor barang berjumlah empat buah untuk menampung masing-masing empat barang yang berbeda. Konveyor keranjang terletak di bawah dan berfungsi untuk menjalankan keranjang yang diletakkan di atasnya. Terakhir papan seluncur untuk mengarahkan barang yang dijatuhkan tepat ke keranjang. Komponen-komponen perangkat keras memiliki fungsi sebagai berikut

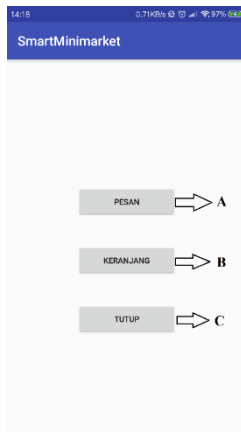
- Motor DC konveyor barang berfungsi menjalankan konveyor barang untuk menjatuhkan barang ke papan seluncur
- Motor DC konveyor keranjang berfungsi menjalankan konveyor keranjang ketika barang yang dipesan sudah sesuai dengan data pemesanan
- Motor driver berfungsi mengontrol masukan dari mikrokontroler untuk menjalankan motor DC
- Laser diode dan sensor cahaya merupakan satu kesatuan yang memiliki satu fungsi yaitu mendeteksi barang yang lewat
- ADC berfungsi memproses masukan analog menjadi data digital untuk dapat diproses oleh mikrokontroler
- NodeMCU berperan sebagai mikrokontroler untuk mengontrol komponen Prototype alat, membaca data pemesanan dari database, serta mengirimkan masukan kembali ke database

### Implementasi Perangkat Lunak

Implementasi perangkat lunak terbagi ke dalam 3 bagian yaitu aplikasi mobile, aplikasi PC, dan webserver.

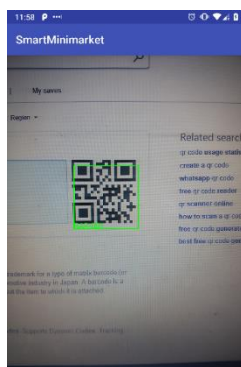
#### Implementasi Perangkat Lunak Mobile

Pada aplikasi mobile, terdapat lima halaman dengan interface dan perancangan yang berbeda yaitu halaman menu utama, halaman pemindaian barcode, halaman pemesanan barang, halaman penyuntingan jumlah barang dan halaman keranjang.



**Gambar 17 Tampilan Halaman Menu Utama**

Halaman ini merupakan halaman awal dari aplikasi. Pada halaman ini, *user* akan diberikan tiga buah pilihan tombol. Tampilan dari halaman menu utama dapat dilihat pada gambar 17. Pada gambar 17, terlihat bahwa ada tiga tombol menu yang dapat dipilih oleh *user*. Pertama yaitu menu pesan dimana akan mengalihkan *user* ke halaman pemindaian *barcode*. Kemudian yang kedua yaitu menu keranjang dimana akan mengalihkan *user* ke halaman keranjang. Terakhir yaitu menu tutup dimana berfungsi menutup aplikasi.

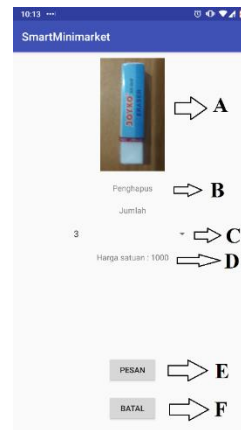


**Gambar 18 Tampilan Halaman Pemindaian Barcode**

Halaman pemindaian *barcode* berfungsi untuk memindai *barcode* barang yang akan dipesan oleh pelanggan. Halaman ini akan menampilkan layar penangkapan kamera. Disini, *user* harus mengarahkan kamera ke *barcode* barang yang akan dipesan. Jika aplikasi berhasil mendeteksi *barcode*, maka *user* akan dialihkan ke halaman pemesanan barang. Untuk dapat kembali ke halaman menu utama, *user* dapat menekan tombol back pada navigation button. Tampilan dari halaman pemindaian *barcode* dapat dilihat pada gambar 18.

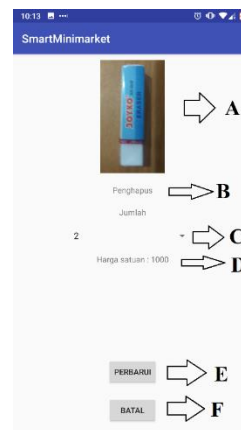
Halaman pemesanan barang merupakan halaman yang berfungsi untuk menampilkan informasi barang dari *barcode* yang telah dipindai sebelumnya. Pada halaman ini, terdapat kolom yang memuat gambar barang sesuai dengan *barcode* yang dipindai. Kemudian dibagian bawah gambar terdapat nama barang. Kemudian dibawah nama barang terdapat kolom jumlah dimana *user* bisa memilih jumlah barang yang akan dipesan. Dibawah kolom jumlah terdapat informasi harga barang per-satuannya. Pada bagian bawah halaman, terdapat dua tombol yaitu tombol pesan dan tombol batal. Jika *user* memilih tombol pesan, maka data barang yang dipesan akan disimpan secara lokal. Namun, jika *user* memilih tombol batal maka tidak akan ada perubahan. Kedua tombol pesan dan tombol batal akan mengalihkan *user*

kembali ke halaman pemindaian *barcode*. Tampilan dari halaman pemesanan barang dapat dilihat pada gambar 19.



**Gambar 19 Tampilan Halaman Pemesanan Barang**

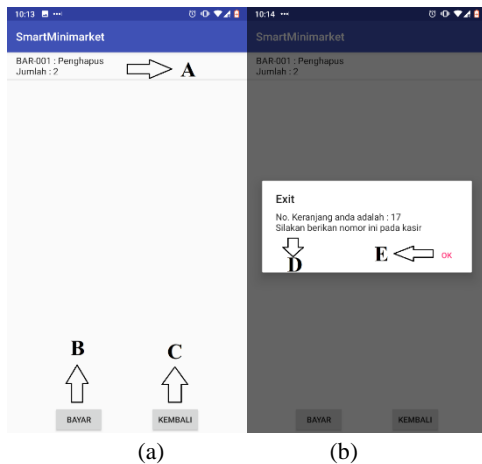
Halaman penyuntingan jumlah barang pada dasarnya tidak jauh berbeda dengan halaman pemesanan barang, hanya saja pada halaman ini tombol 'pesan' diubah tulisannya menjadi 'perbarui'. Pada implementasinya, halaman penyuntingan jumlah barang dan halaman pemesanan barang hanya memakai satu aset halaman dan tombol pesan/perbarui akan berubah sesuai dengan ketentuan yang telah ditetapkan. Halaman penyuntingan jumlah barang dapat diakses ketika *user* memilih barang dari halaman keranjang. Sesuai namanya, halaman penyuntingan jumlah barang berfungsi untuk menyunting jumlah barang yang telah dipesan oleh *user* sebelumnya. Pilihan jumlah barang yang disunting terbatas pada jumlah barang yang dipesan oleh *user* sebelumnya, sehingga halaman ini fungsi utamanya mengurangi atau membatalkan jumlah barang yang telah dipesan. Tampilan dari halaman penyuntingan jumlah barang dapat dilihat pada gambar 20.



**Gambar 20 Tampilan Halaman Penyuntingan Jumlah Barang**

Halaman keranjang merupakan halaman yang berfungsi menampilkan semua barang yang dipesan oleh *user*. Halaman ini memuat kode barang, nama barang, dan jumlah barang yang dipesan. Di bagian bawah halaman terdapat dua tombol yaitu tombol pesan dan tombol batal. Jika *user* sudah yakin dengan barang-barang yang dipesan, maka *user* dapat menekan tombol bayar supaya pesanan dapat diproses oleh *Prototype* alat. Jika *user* menekan tombol kembali, maka *user* akan dialihkan ke halaman menu utama. *User* juga dapat menyunting jumlah barang yang telah dipesan dengan mengklik barang dari daftar keranjang,

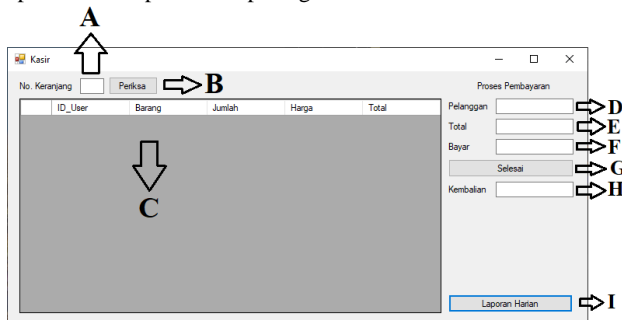
lalu *user* akan dialihkan ke halaman penyuntingan jumlah barang. Tampilan dari halaman keranjang dapat dilihat pada gambar 21.



Gambar 21 (a) Tampilan Halaman Keranjang, (b) Tampilan Pop-Up Ketika User Menekan Tombol Bayar

**Implementasi Perangkat Lunak PC**

Pada aplikasi PC untuk kasir, terdapat satu buah halaman yang digunakan untuk memproses pemesanan pelanggan. Tampilan aplikasi PC dapat dilihat pada gambar 22.



Gambar 22 Tampilan Perangkat Lunak PC Kasir

Pada gambar 22, dapat dilihat ada beberapa komponen di dalam aplikasi tersebut. Pertama ada kolom untuk memasukkan nomor keranjang yang nantinya akan didapat dari pelanggan. Kemudian tombol periksa dapat digunakan untuk memeriksa data pemesanan sesuai dengan masukan dari kolom nomor keranjang. Selanjutnya ada kolom daftar barang dimana akan menampilkan tabel yang berisi informasi barang yang dipesan sesuai dengan masukan pada kolom nomor keranjang. Kemudian ada kolom pelanggan untuk menampilkan nomor id pelanggan dan kolom total untuk menampilkan total harga pemesanan barang. Selanjutnya ada kolom bayar dimana kasir bisa memasukkan nilai biaya yang dibayarkan oleh pelanggan pada kasir. Di bawah kolom bayar ada tombol selesai dimana akan memproses nilai kembalian dari biaya yang dibayarkan dan total harga pemesanan. Kemudian ada kolom kembalian yang akan menampilkan nilai kembalian ketika tombol selesai sudah ditekan.

**Implementasi Sistem**

Sistem terdiri atas aplikasi *mobile*, *Prototype* alat, dan aplikasi PC. Sistem akan memproses data pemesanan pelanggan melalui tahap-tahap yang telah ditentukan. Adapun alur kerja dari sistem yaitu

- a) Pelanggan memesan barang melalui aplikasi *mobile*.

- b) *Prototype* alat membaca data pemesanan pada *database*. Jika terdapat data pemesanan maka data tersebut akan diproses untuk selanjutnya dieksekusi. Data yang didapat dari *database* masih bersifat mentah sehingga perlu diekstrak supaya data dapat diolah dengan mudah.
- c) *Prototype* alat mengekstrak data mentah yang didapat dengan cara membagi data dan memasukkan data ke dalam array supaya pengolahan data pemesanan menjadi lebih mudah.
- d) *Prototype* alat menjalankan motor DC. Jika barang yang diproses sudah sesuai dengan data pemesanan, maka fungsi ini akan menghentikan motor DC konveyor.
- e) *Prototype* alat membaca data sensor cahaya pada konveyor barang untuk mendeteksi apakah barang melewati sensor atau tidak.
- f) Pelanggan mengunjungi kasir untuk membayar dan mengambil barang. Kasir akan memeriksa pemesanan berdasarkan nomor keranjang yang diberikan pada aplikasi PC dan memproses pembayaran tersebut.

**Pengujian**

Pengujian dan analisa yang dilakukan dalam penelitian ini terdapat tiga tahap, yaitu pengujian terhadap perangkat keras, perangkat lunak dan sistem secara keseluruhan.

**Pengujian Perangkat Keras**

Pengujian perangkat keras dilakukan pada *Prototype* alat untuk menguji apakah alat dapat bekerja. Pada pengujian ini terdapat dua komponen yang diuji yaitu sensor cahaya dan motor DC. Tabel hasil pengujian perangkat keras ditampilkan pada tabel 6.

Tabel 6 Hasil Pengujian Sensor Cahaya

Percobaan	Sensor 1		Sensor 2		Sensor 3		Sensor 4		Sensor 5	
	Ada	Tidak ada	Ada	Tidak ada	Ada	Tidak ada	Ada	Tidak ada	Ada	Tidak ada
1	4389	19762	1737	24785	2140	20683	1075	22803	492	1024
2	4245	19610	1763	24782	2167	20702	1091	22815	470	1024
3	4276	19789	1779	24684	2159	20681	1087	22797	493	1024
4	4284	19720	1761	24786	2176	20697	1063	22635	495	1024
5	4298	19753	1785	24791	2174	20703	1054	22804	475	1024

Pada pengujian sensor cahaya dapat dilihat nilai dari sensor pada saat ada barang yang menghalangi cahaya laser atau tidak. Sensor 1 sampai dengan sensor 4 merujuk pada sensor cahaya pada konveyor barang, sedangkan sensor 5 merujuk pada sensor cahaya konveyor keranjang. Karena sensor 1 sampai sensor 4 dihubungkan pada ADC ADS1115 maka keluaran data yang dihasilkan bisa mencapai 16-bit (nilai maksimum 65.536), sedangkan sensor 5 langsung terhubung dengan ADC built-in pada NodeMCU yang hanya memiliki keluaran data 10-bit (nilai maksimum 1024). Pada tabel 4.1 terlihat bahwa nilai sensor akan rendah ketika ada barang menghalangi cahaya laser ke sensor cahaya dengan nilai tertinggi yaitu 4.386 pada sensor cahaya konveyor barang dan 495 pada sensor cahaya konveyor keranjang. Sedangkan nilai sensor akan tinggi ketika cahaya laser ke sensor cahaya tidak dihalangi oleh apapun dengan nilai terendah yaitu 19.610 pada sensor cahaya konveyor barang dan 1.024 pada sensor cahaya konveyor keranjang. Nilai yang didapat ini akan menjadi acuan pada program *Prototype* alat untuk mendeteksi apakah ada barang yang lewat atau tidak.

**Pengujian Perangkat Lunak**

Pengujian perangkat lunak dilakukan pada aplikasi *mobile*, aplikasi PC dan program *Prototype* alat untuk menguji apakah aplikasi bekerja sesuai dengan perkiraan.

**Pengujian Perangkat Lunak Aplikasi Mobile**

Pengujian perangkat lunak aplikasi *mobile* terdiri atas pengujian pemindaian *barcode*, pengujian halaman pemesanan barang, dan pengujian halaman keranjang.

Pengujian pemindaian *barcode* pada dasarnya bergantung pada banyak faktor untuk bisa mendeteksi *barcode*. Supaya tidak terjadi perluasan tipe-tipe pengujian, maka diberikan batasan untuk pengujian dengan kriteria:

- a) *Barcode* berukuran 3.5cm x 3.5cm
- b) Percobaan dilakukan dengan pencahayaan ruangan
- c) Kamera memiliki mode *auto-focus*
- d) Posisi *barcode* dan kamera sejajar (tidak ada perbedaan sudut)
- e) Tidak dilakukan zoom kamera saat pemindaian *barcode*

Dengan kriteria sebagai pembatas pengujian diatas, maka didapatkan hasil pengujian yang dapat dilihat pada tabel 7.

**Tabel 7 Hasil Pengujian Jarak Kemampuan Baca Barcode**

Percobaan	Jarak	Terbaca
1.	5cm	Tidak
2.	6cm	Tidak
3.	7cm	Ya
4.	10cm	Ya
5.	20cm	Ya
6.	30cm	Ya
7.	40cm	Ya
8.	50cm	Ya
9.	60cm	Ya
10.	70cm	Tidak

Dari tabel 7 di atas dapat dilihat bahwa pengujian pemindaian *barcode* berhasil dengan jarak minimum yaitu 6cm dan jarak maksimum yaitu 60cm. Pada jarak dibawah 6cm, *barcode* yang akan dipindai tidak tampil secara keseluruhan sehingga pemindaian akan gagal. Sedangkan pada jarak diatas 60cm, *barcode* akan terlihat kecil dan program tidak dapat memproses gambar dari *barcode* karena ukurannya tersebut.

Berikutnya dilakukan pengujian terhadap kecepatan baca *barcode* dimana dapat dilihat hasilnya pada tabel 7 dan 8.

**Tabel 8 Hasil Pengujian Kecepatan Baca Barcode (30cm)**

Percobaan	Waktu respon
1	1.68 detik
2	1.76 detik
3	2.41 detik
4	1.71 detik
5	1.51 detik

**Tabel 9 Hasil Pengujian Kecepatan Baca Barcode (60cm)**

Percobaan	Waktu respon
1	1.43 detik
2	1.52 detik
3	0.54 detik
4	1.36 detik
5	2.13 detik

Berdasarkan tabel 7 dan 8, terlihat bahwa kecepatan baca *barcode* dapat dikatakan relatif cepat dengan waktu respon terendah yaitu 1.51 detik dan waktu respon terlama yaitu 2.41 detik. Hasil pengujian ini dapat mewakili data untuk seluruh jarak kemampuan baca *barcode* dimana selama *barcode* dapat dipindai pada jarak tersebut maka kecepatan bacanya juga relatif sama pada jarak lain. Hal ini dibuktikan pada hasil pengujian kecepatan baca *barcode* pada jarak 60cm yang dapat dilihat pada tabel 8.

Pada halaman pemesanan barang dilakukan pengujian apakah aplikasi dapat membaca data barang pada *database* atau tidak dan aplikasi dapat memperbarui data pada *database* atau tidak. Hasil pengujian halaman pemesanan barang dapat dilihat pada gambar 23.

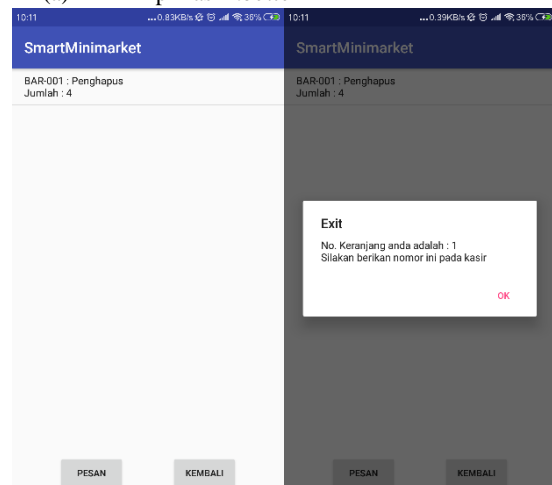


**Gambar 23 Hasil Pengujian Halaman Pemesanan Barang**

Dari gambar 23 dapat dilihat bahwa aplikasi dapat membaca data barang dari *database* dan aplikasi juga dapat memperbarui data jumlah barang pada *database*. Aplikasi dikatakan sukses dalam membaca data barang karena halaman pemesanan barang dapat menampilkan informasi barang. Kemudian aplikasi juga dikatakan sukses dalam memperbarui data jumlah barang pada *database* karena data jumlah barang yang dipesan menjadi berkurang sesuai dengan jumlah barang yang dipilih pada aplikasi.

Pada halaman keranjang, akan diuji apakah aplikasi dapat mengirimkan data pemesanan ke *database* atau tidak dan apakah aplikasi dapat memberikan nomor keranjang sesuai dengan ketentuan atau tidak. Hasil pengujian halaman keranjang yaitu :

(a) Aplikasi mobile



**Gambar 24 Tampilan Aplikasi Mobile Pengujian Halaman Keranjang**



(b) Database

id_user	data_user	no_keranjang	flag
1	4000	1	1

Tabel pelanggan :

id_user	id_barang	jumlah_pesanan
1	BAR-001	4

Tabel pesanan :

**Gambar 25 Tampilan Database Pengujian Halaman Keranjang**

Berdasarkan pengujian diatas dapat dilihat bahwa aplikasi dapat mengirimkan data pemesanan ke *database* dimana hal ini dapat dibuktikan pada *database* terdapat data pemesanan yang sesuai dengan masukan dari aplikasi *mobile*. Selain itu, dari gambar 24 aplikasi juga berhasil dalam memberikan nomor keranjang sesuai dengan ketentuan yang dijelaskan pada implementasi halaman keranjang. Berdasarkan ketentuan yang telah disebutkan dapat dilihat bahwa aplikasi berhasil memberikan nomor keranjang. Perlu diinformasikan bahwa jika keranjang yang diberikan lebih dari satu buah maka nomor keranjang yang disimpan pada *database* hanyalah nomor keranjang terakhir sehingga pada kasir nantinya hanya memasukkan nomor terakhir pada aplikasi PC.

**Pengujian Perangkat Lunak Aplikasi PC**

Pada perangkat lunak aplikasi PC akan diuji apakah aplikasi dapat membaca data pemesanan pada *database* sesuai dengan masukan nomor keranjang. Hasil pengujian perangkat lunak aplikasi PC yaitu:

(a) Database

id_user	data_user	no_keranjang	flag
19	0003	29	0

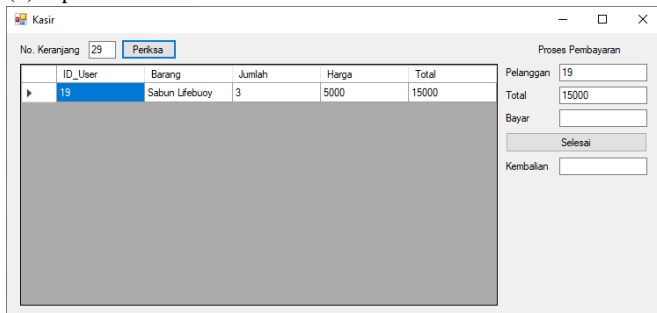
Tabel pelanggan :

id_user	id_barang	jumlah_pesanan
19	BAR-004	3

Tabel pesanan :

**Gambar 26 Tampilan Database Pengujian Aplikasi PC**

(b) Aplikasi PC



**Gambar 27 Tampilan Aplikasi Pengujian Aplikasi PC**

Berdasarkan pengujian diatas dapat dilihat bahwa aplikasi PC mampu membaca data pemesanan barang pada *database* sesuai dengan masukan pada kolom nomor keranjang. Hal ini dapat dilihat pada bagian kolom daftar barang yang menampilkan tabel berisi data pemesanan barang dan data pada tabel ini sesuai dengan data yang berada pada tabel pesanan pada *database*.

**Pengujian Perangkat Lunak Prototype Alat**

Pada perangkat lunak *prototype* alat, akan diuji apakah *prototype* alat dapat membaca data pemesanan barang pada *database* dan *prototype* alat dapat memperbarui data pada *database*. Hasil

pengujian perangkat lunak *Prototype* alat dapat dilihat pada tabel 10.

**Tabel 10 Hasil Pengujian Perangkat Lunak Prototype Alat**

Percobaan	Data user pada tabel pelanggan	Data mentah yang dibaca alat prototype	Waktu respon	Data user yang dikirim alat prototype	Data user pada tabel pelanggan setelah diperbarui
1	id_user: 23, data_user: 1100, flag: 1	{"queue": [{"id": "23", "data": "1100"}]}	6.37 detik	23	id_user: 23, flag: 0
2	id_user: 24, data_user: 1200, flag: 1	{"queue": [{"id": "24", "data": "1200"}]}	6.53 detik	24	id_user: 24, flag: 0
3	id_user: 25, data_user: 1220, flag: 1	{"queue": [{"id": "25", "data": "1220"}]}	7.84 detik	25	id_user: 25, flag: 0
4	id_user: 26, data_user: 1531, flag: 1	{"queue": [{"id": "26", "data": "1531"}]}	9.33 detik	26	id_user: 26, flag: 0
5	id_user: 27, data_user: 1584, flag: 1	{"queue": [{"id": "27", "data": "1584"}]}	7.17 detik	27	id_user: 27, flag: 0

Berdasarkan tabel 10 diatas dapat dilihat bahwa *Prototype* alat berhasil dalam membaca data pemesanan barang pada *database* dan juga dapat memperbarui data pada *database*. Indikator keberhasilan *Prototype* alat berhasil membaca data pemesanan barang pada *database* yaitu pada informasi data mentah yang dibaca oleh *Prototype* alat dari *database*. Data mentah ini berisi informasi pemesanan pelanggan berupa id user dan data pemesanan pelanggan. Adapun respon *Prototype* alat dalam membaca data pemesanan dari *database* yaitu dengan waktu tercepat 6.37 detik dan waktu terlama 9.33 detik. Perhitungan waktu dimulai ketika user menekan tombol pesan pada halaman keranjang pada aplikasi dan diakhiri ketika data mentah telah sukses dibaca oleh *Prototype* alat.

Lalu *Prototype* alat juga berhasil dalam memperbarui data pada *database* dimana indikator keberhasilan dapat dilihat pada data flag pada tabel pelanggan. Nilai flag ketika pemesanan belum diproses akan bernilai 1 dan nilai ini akan berubah menjadi 0 ketika pemesanan sudah diproses dimana nilai ini diubah melalui perangkat lunak *Prototype* alat. Dalam mengubah nilai flag dari 1 menjadi 0, *Prototype* alat mengirimkan data ke *database* berupa data id user sehingga pada *database* akan terjadi perubahan nilai flag sesuai dengan id user yang dikirimkan tersebut.

**Pengujian Sistem Keseluruhan**

Pada tahap ini, akan diujikan keselarasan antara kerja aplikasi *mobile*, *Prototype* alat, dan aplikasi PC dalam meng-handle data pemesanan pelanggan. Aplikasi *mobile* bertindak sebagai media masukan dalam menentukan barang-barang yang akan dipesan oleh pelanggan, *Prototype* alat bertindak memproses pemesanan pelanggan dengan menjatuhkan barang yang sesuai dengan data pemesanan ke keranjang, serta aplikasi PC bertindak membaca data pemesanan pelanggan untuk dapat memberikan informasi pada kasir.

Hasil pengujian sistem keseluruhan yaitu:

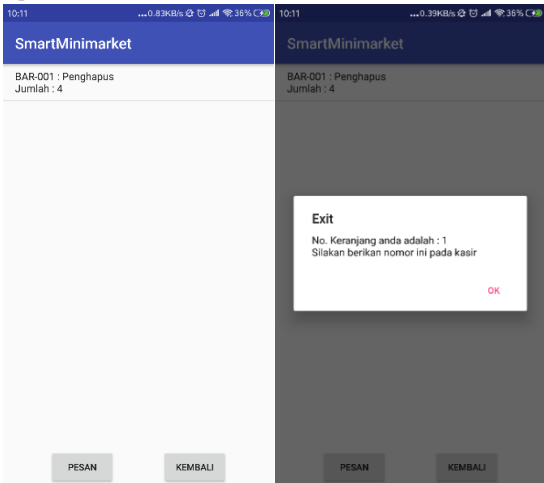
(a) Barang pemesanan





**Gambar 28 Tampilan Pemesanan Pengujian Sistem Keseluruhan**

(b) Aplikasi *mobile*



**Gambar 29 Tampilan Aplikasi *Mobile* Pengujian Sistem Keseluruhan**

(c) *Database*

Tabel pelanggan :

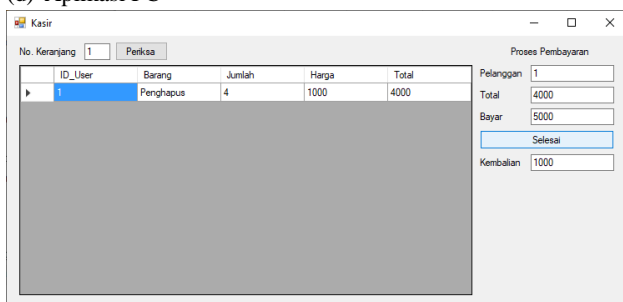
id_user	data_user	no_keranjang	flag
1	4000	1	1

Tabel pesanan :

id_user	id_barang	jumlah_pesanan
1	BAR-001	4

**Gambar 30 Tampilan *Database* Pengujian Sistem Keseluruhan**

(d) Aplikasi PC



**Gambar 31 Tampilan Aplikasi PC Pengujian 1 Sistem Keseluruhan**

Dari percobaan diatas dapat dilihat hasil pengujian dari seluruh bagian-bagian utama sistem. Pada seluruh percobaan dapat dilihat bahwa sistem bekerja dengan baik dalam memproses pemesanan pelanggan baik dari aplikasi *mobile*, *Prototype* alat, maupun aplikasi PC. Pada beberapa kasus, motor DC bekerja dengan baik namun konveyornya tidak berjalan dengan sempurna atau macet. Hal ini dikarenakan konstruksi alat yang tidak begitu optimal sehingga terdapat celah diantara alas konveyor dan motor DC yang menyebabkan motor DC tidak dapat menarik alas konveyor.

Meskipun demikian, secara garis besar sistem sudah bekerja dengan baik dan benar.

**KESIMPULAN**

Berdasarkan perancangan, pengamatan, dan pengujian yang telah dilakukan pada sistem *Prototype Smart Minimarket* maka dapat diambil kesimpulan sebagai Berikut

1. Aplikasi *mobile* dapat memindai *barcode* barang menggunakan *Mobile Vision API* dengan jarak terdekat 7cm, jarak terjauh 60cm, waktu respon tercepat 0.54 detik, dan waktu respon terlama 2.41 detik pada *barcode* berukuran 3.5cm x 3.5cm
2. Sistem dapat memproses pemesanan pelanggan yang dipesan melalui aplikasi *mobile* dengan rata-rata persentase keberhasilan sebesar 100%
3. Aplikasi *mobile* dapat menginformasikan nomor keranjang yang akan diambil pada kasir dengan pengaturan nomor keranjang
4. Kapasitas keranjang mampu menampung barang dengan berat maksimal 100g, jika melebihi akan diletakkan pada keranjang berikutnya
5. Aplikasi PC dapat menginformasikan data pemesanan pelanggan melalui masukan nomor keranjang dengan rata-rata persentase keberhasilan sebesar 100%

**SARAN**

Untuk Pengembangan penelitian selanjutnya dapat dilakukan :

1. Penomoran keranjang juga menggunakan *barcode*.
2. Sistem dapat mengukur ukuran / volume barang yang akan dimasukkan ke dalam keranjang

**REFERENSI**

- [1] Haryani, Orisa Shinta. 2017. Alasan *Minimarket* Jadi Sasaran Empuk Perampokan. <https://kriminologi.id/lapor-waspada/peta-kejahatan/alasan-minimarket-jadi-sasaran-empuk-perampokan>, diakses pada tanggal 20 Maret 2018
- [2] Fukushima, Kurumi. 2016. Panasonic introduces robotic checkout at a grocery store in Osaka. <https://www.cnn.com/2016/12/14/panasonic-introduces-robotic-checkout-at-a-grocery-store-in-osaka.html>, diakses pada tanggal 20 Maret 2018
- [3] Risma, Asep Mulyana, dan Sholekan. 2010. Aplikasi Mini Market Online Berbasis *Web*. Bandung: Telkom University
- [4] Yendri, D., & Putri, R. (2018, March 29). Sistem Pengontrolan Dan Keamanan Rumah Pintar (Smart Home) Berbasis Android. *Journal of Information Technology and Computer Engineering*, 2(01), 1-6.
- [5] Putri, A., & Yendri, D. (2018, March 29). Sistem Pemesanan Makanan dan Minuman Pada Restoran Menggunakan Teknologi NFC Berbasis Android. *Journal of Information Technology and Computer Engineering*, 2(01), 34-40.
- [6] NodeMcu – An open-source firmware based on ESP8266 wifi-soc. [http://nodemcu.com/index\\_en.html](http://nodemcu.com/index_en.html), diakses pada tanggal 17 April 2018
- [7] Espressif. 2018. ESP8266EX Datasheet Version 5.8. Pudong: Espressif Systems (Shanghai) Pte. Ltd

- [8] Node MCU. <http://support.infocenter.gr/portfolio-view/node-mcu/>, diakses pada tanggal 17 April 2018
- [9] Ai-Thinker. 2015. ESP-12E WiFi Module. Bao'an District: Shenzen Ai-ThinkerTechnology Co., Ltd
- [10] Prabowo, Brilliant Adhi. 2013. Pemodelan Sistem Kontrol Motor DC dengan Temperatur Udara sebagai Pemicu. Bandung: LIPI
- [11] Kyohritsu. 2018. HC02-48 Datasheet. Naniwa-ku: Kyoritsu Electrical Instruments Works, Ltd.
- [12] Demik, Randal J, dkk. 2013. Measuring Intensity of Laser Light Penetrating Flight Decks in Laser Illuminations. Romeoville: Lewis University
- [13] Ginting, Filemon J. 2013. Perancangan Alat Ukur Kekeruhan Air Menggunakan Light Dependent Resistor Berbasis Mikrokontroler Atmega 8535. Manado: Universitas Sam Ratulang.
- [14] Texas Instrument. 2018. 16-Bit Analog-to-Digital Converter with Internal Reference. Dallas: Texas Instruments Incorporated
- [15] Ubuntu. 2018. Ubuntu Server Guide. London: Canonical Group Limited.
- [16] Saputro, Haris. 2012. Modul Pembelajaran Praktek Basis Data (MySQL). Semarang: Universitas Dian Nuswantoro
- [17] *Mobile* Vision. <https://developers.google.com/vision/>, diakses pada tanggal 4 April 2018.
- [18] *Barcode* API Overview. <https://developers.google.com/vision/android/barcodes-overview>, diakses pada tanggal 4 April 2018.