

Aplikasi Transaksi Penjualan Makanan Dan Pemesanan Tempat Duduk Berbasis Android Yang Diintegrasikan Dengan Iot For Canteen

1st Dimas Aji Sukmandaru
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

dimasajisukmandaru@student.telkomu
niversity.ac.id

2nd Periyadi
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

periyadi@staff.telkomuniversity.ac.id

3rd Lisda Meisaroh
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

lisdameisaroh@staff.telkomuniversity.a
c.id

Abstrak — Universitas Telkom memiliki kantin di setiap fakultas yang ada. Kantin yang biasa digunakan oleh mahasiswa dan karyawan sebagai pelanggan kadang memiliki antrian yang panjang untuk membayar ataupun memesan. Sehingga menyita waktu untuk para pelanggan mengantri saja apalagi ketika kantin sedang penuh. Kadang pelanggan juga tidak mengetahui apakah menu yang diinginkan masih tersedia atau sudah habis. Dari sisi penjual, tidak jarang mereka melewatkan pesanan yang telah dipesan oleh pelanggan sehingga pelanggan harus menunggu lagi makanan mereka untuk dibuat. Sering pula terjadi bahwa seluruh tempat duduk yang tersedia di kantin juga penuh, sehingga mahasiswa maupun pegawai yang ingin makan kesusahan untuk mencari tempat duduk.

Proyek ini diintegrasikan dengan sebuah sistem untuk meningkatkan kenyamanan pelanggan ketika berada di area kantin yaitu IoT For Canteen. Sistem ini terhubung dengan beberapa sensor yang menangkap data dari kondisi yang dapat mempengaruhi kenyamanan kantin. Sistem ini diintegrasikan dengan aplikasi berbasis Android yang memiliki fitur reservasi dan dua metode pembayaran, yaitu secara langsung dan melalui pembayaran secara daring. Sistem dan aplikasi ini dihubungkan dengan Firebase untuk melakukan pertukaran data antara sistem dan aplikasi.

Proyek ini diharapkan mampu mengurangi jumlah antrian dengan menambah efisiensi transaksi antara penjual dan pembeli dengan implementasi pembayaran daring serta mampu menambah kenyamanan serta tingkat pelayanan yang ada di kantin.

Kata kunci— *Android, Kantin, Internet Of Things, Firebase*

I. PENDAHULUAN

Teknologi pada masa saat ini sangat penting dalam kehidupan sehari-hari, baik dalam bidang pendidikan, informasi, bahkan dalam bidang ekonomi. Dalam bidang ekonomi juga masyarakat tidak bisa lepas dari industri kuliner. Industri kuliner di Indonesia sendiri berkembang pesat dengan aneka masakan yang ada di Indonesia. Industri kuliner ini dapat kita temukan dimana saja karena makanan merupakan salah satu kebutuhan pokok yang dibutuhkan oleh manusia.

Industri ini dapat berbentuk pabrik, rumah makan, maupun kios yang menjual makanan. Kios-kios ini mampu kita jumpai diberbagai tempat, bahkan kampus pun menyediakan kantin yang berisi kios-kios dengan berbagai pilihan makanan yang ada. Di Universitas Telkom memiliki kantin di setiap fakultas yang ada. Kantin yang biasa digunakan oleh mahasiswa kadang memiliki antrian yang panjang untuk membayar ataupun memesan. Sehingga menyita waktu untuk para mahasiswa mengantri saja apalagi ketika kantin sedang penuh. Kadang mahasiswa juga tidak mengetahui apakah menu yang diinginkan masih tersedia atau sudah habis. Dari sisi penjual, tidak jarang mereka melewatkan pesanan yang telah dipesan oleh pelanggan sehingga pelanggan harus menunggu lagi makanan mereka untuk dibuat. Sering pula terjadi bahwa seluruh tempat duduk yang tersedia di kantin juga penuh, sehingga mahasiswa maupun pegawai yang ingin makan kesusahan untuk mencari tempat duduk, tidak jarang pula mahasiswa ataupun pegawai tidak jadi untuk memesan makanan karena hal tersebut.

Dengan adanya aplikasi ini mahasiswa mampu mendapat informasi mengenai tingkat keramaian dari kantin, menu yang tersedia di setiap kios, dan dapat melakukan pemesanan baik tempat duduk maupun makanan melalui handphone mereka masing masing dan mempermudah pelanggan dan penjual dalam bertransaksi.

Adapun tujuan dari Proyek Akhir ini, adalah membuat sistem transaksi melalui aplikasi dengan 2 metode pembayaran yaitu membayar langsung melalui kasir dan membayar secara online, membuat aplikasi android yang diintegrasikan dengan IoT for Canteen untuk menampilkan informasi kondisional dari area kantin dan mengatur komponen output dan input, sistem reservasi yang terhubung dengan sensor proximity bertujuan agar pelanggan dapat memesan tempat duduk dan memeriksa apakah meja kantin sedang penuh atau tidak.

II. KAJIAN TEORI

A. STUDI LITERATUR

Pada tahun 2018 terdapat sebuah proyek yang dilakukan di The Deloitte Building di kota Oslo. Bangunan tersebut

memiliki sebuah kantin yang setiap harinya dipakai oleh karyawan maupun pengunjung dari bangunan tersebut untuk makan. Banyak orang yang mengeluh ketika mengantri di kantin tersebut, karena cukup banyak orang yang mengantri dan banyak orang yang menunggu terlalu lama dan membatalakan untuk makan disitu meskipun makanan di kantin tersebut terkenal lezat. Proyek tersebut dilakukan untuk mendeteksi antrian pada kantin di Bangunan tersebut, karena setiap hari kantin tersebut cukup ramai sehingga banyak orang keluar dengan perut lapar karena tidak mendapatkan makanan. Proyek ini menggunakan sensor suara untuk mendeteksi banyaknya orang pada antrian dan data akan dikirimkan ke aplikasi yang terpasang di handphone, sehingga pelanggan yang akan ke kantin mampu mengawasi seberapa banyak antrian yang ada.[1]

Salah satu perusahaan yaitu Bosch juga menerapkan konsep Crowd Sensors atau sensor keramaian yang mampu diterapkan pada kantin-kantin. Karena kantin memiliki daya tampung yang besar sebagai sebuah fasilitas umum yang kerap dipakai setiap orang. Sehingga dengan semakin banyak orang yang masuk di area kantin suhu dalam ruangan akan menjadi semakin panas. Sistem ini mengatur suhu yang ada pada ruangan antara 24.5 C sampai 26 C. Sehingga pelanggan didalam kantin mampu merasa nyaman didalam kantin.[2]

Pada tahun 2021 dibangun sebuah proyek yang dilakukan di daerah Palladam, India. Proyek ini bernama IoT based Smart Cafeteria Management System. Proyek ini memiliki tujuan untuk memudahkan pekerjaan pegawai kafetaria dengan memberikan fitur pemesanan melalui sebuah aplikasi Android. Aplikasi ini dibantu dengan sebuah sensor IR Proximity yang terhubung dengan sebuah microcontroller. Sensor mengirim data tempat duduk yang kosong. Aplikasi dan microcontroller terhubung dengan sebuah modul Wi-fi yang diatur melalui IoT.[3]

Salah satu universitas di India yaitu KL University membuat sebuah aplikasi Billing untuk kantin mereka. Aplikasi tersebut dibuat guna mengurangi penggunaan nota kertas yang biasa digunakan untuk bertransaksi dan menambah efisiensi serta menambah tingkat pelayanan kepada pelanggan. Pada aplikasi tersebut pelanggan mampu melakukan pembayaran secara daring dan mampu memberikan ulasan pada makanan yang dijual.[4]

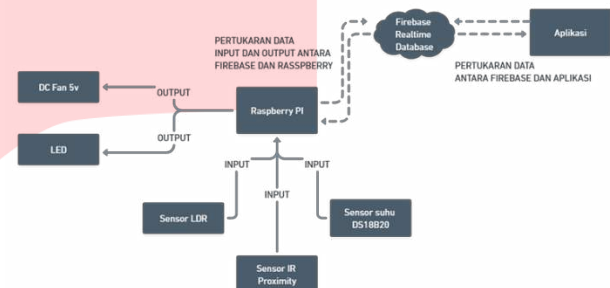
Sebuah purwarupa kantin elektronik berbasis Internet of Things dirancang disebuah kantin kejujuran. Kantin kejujuran merupakan sebuah kantin dengan konsep pelanggan mengambil makanan pilihan mereka dan membayar sendiri di tempat pembayaran yang telah disiapkan. Purwarupa tersebut berbentuk sebuah mesin penjual makanan ringan dan minuman otomatis yang diatur oleh administrator. Administrator dan penjual mampu menginventarisasi produk yang dijual didalam mesin. Untuk menangani transaksi dibuat sebuah sistem pembayaran dengan menggunakan RFID. Dengan menggunakan sistem ini pembayaran dapat menjadi semakin mudah dan administrator maupun penjual dapat memonitoring transaksi yang ada dengan mudah.[5]

Proyek ini mengintegrasikan aplikasi berbasis Android dan sistem kenyamanan IoT For Canteen dengan menggunakan realtime database yang dimiliki oleh Firebase API. Aplikasi ini dibangun menggunakan Android Studio sebagai integrated development environment(IDE) serta menggunakan Kotlin sebagai bahasa pemrograman.

Organisasi dan lalu lintas data yang terjadi semua diolah melalui Firebase. Untuk bagian komponen yang akan digunakan dan diintegrasikan dengan aplikasi ini adalah Raspberry Pi 3 Model B, sensor cahaya LDR, sensor suhu DS18B20, motor kipas angin DC 5V, dan Lampu LED. Aplikasi ini akan memiliki fitur berupa reservasi tempat duduk, informasi kondisi kantin, dan pembayaran secara online dengan menggunakan payment gateway dari Midtrans.

III. METODE

Pada bagian ini akan membahas mengenai metode yang digunakan untuk membangun proyek ini. Metode yang diimplementasikan adalah metode SDLC (System Development Life Cycle). Metode ini digunakan karena SDLC merupakan metode yang mencakup analisa, perancangan, pengembangan, dan perawatan berkala pada perangkat lunak yang sedang dibuat.



GAMBAR 1

A. ANALISA KEBUTUHAN SISTEM

Dari tujuan dan hasil akhir yang ingin didapatkan pada proyek ini didapatkan analisa dan kebutuhan perangkat keras untuk membangun sistem sebagai berikut.

No	Nama Perangkat
1	Raspberry Pi 3 Model B
2	Sensor LDR
3	Sensor DS18B20
4	Kipas Angin DC 5V
5	Lampu LED

Perangkat lunak yang dibutuhkan untuk membantu pengembangan dan digunakan agar sistem dapat berjalan adalah sebagai berikut.

No	Nama Perangkat Lunak
1	Android Studio
2	Firebase
3	Midtrans

B. PERANCANGAN

Pada gambar 1 sistem dibagi menjadi 2 bagian, yaitu sistem IoT For Canteen yang terdiri dari Raspberry Pi, lampu LED, motor DC, sensor LDR, dan sensor suhu DS18B20. Sistem selanjutnya adalah sistem aplikasi. Sistem IoT For Canteen terdiri atas 3 input dan 2 output. Sensor LDR, sensor proximity dan sensor suhu sebagai input, lalu lampu LED dan motor DC menjadi output. Input dan output yang ada terhubung pada Raspberry Pi. Disini Raspberry Pi terhubung

dengan Firebase Database dan pertukaran data yang ada harus terjadi realtime. Data yang disimpan dari Raspberry Pi pada database adalah data dari sensor suhu, sensor LDR, konfigurasi lampu LED, dan konfigurasi kecepatan motor DC. Selanjutnya pada sistem aplikasi, aplikasi terhubung dengan Firebase database. Aplikasi menyimpan dan mengolah data dari pengguna. Pengguna dibagi menjadi 2, yaitu pelanggan dan penjual. Masing – masing dari pengguna memiliki interface yang berbeda pada aplikasi. Data pelanggan, penjual, dan transaksi yang terjadi selanjutnya disimpan di Firebase database. Pelanggan mampu berinteraksi dengan sistem IoT For Canteen pada interface yang telah disiapkan di dalam aplikasi. Preferensi konfigurasi kenyamanan dari pelanggan akan disimpan di Firebase database lalu data tersebut akan diambil oleh Raspberry Pi dan selanjutnya Raspberry Pi akan mengatur output sesuai preferensi pelanggan.

C. PENGEMBANGAN

Pengembangan dilakukan sesuai dengan hasil perancangan yang telah didapatkan. Aplikasi dibangun dengan menggunakan Android Studio sebagai perangkat lunak. Firebase digunakan sebagai database untuk proyek.

D. TESTING

Uji coba dilakukan dengan melakukan percobaan langsung menguji performa, efektifitas, dan error yang ada pada setiap fitur pada aplikasi dan sistem. Uji coba ini dilakukan untuk memastikan bahwa sistem berjalan dengan semestinya sesuai dengan hasil perancangan yang telah dilakukan. Uji coba dilakukan dengan menggunakan White-box Testing.

E. MAINTENANCE

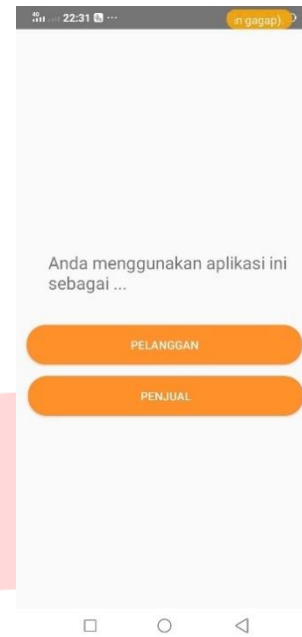
Melakukan maintenance pada semua komponen sistem. Maintenance pada aplikasi dilakukan dengan melakukan bug testing secara berkala. Pada Firebase dilakukan dengan memantau lalu lintas data yang terjadi. Dan pada sistem IoT For Canteen dilakukan dengan melakukan pengecekan secara teratur pada komponen yang anda untuk mendeteksi adanya kerusakan yang terjadi.

IV. HASIL DAN PEMBAHASAN

A. Implementasi Basis Data

Aplikasi dibangun dengan menggunakan Firebase sebagai basis data. Dengan 4 *node* utama. Setiap *node* memiliki fungsi yang berbeda. “canteenStatus” terhubung dengan sistem perangkat keras IoT For Canteen. *Node* ini mengolah *input* dan *output* yang diatur oleh Raspberry Pi berupa pengaturan kipas, pengaturan lampu, data dari sensor LDR, dan data dari sensor suhu. Interaksi dari aplikasi dengan *node* ini adalah menampilkan data dari sensor LDR dan sensor suhu serta mengatur *output* kipas dan lampu melalui aplikasi. “reservationBill” adalah *node* yang mengatur reservasi yang dilakukan oleh pelanggan pada aplikasi. “seller” adalah *node* yang menyimpan data dari penjual. Data yang disimpan berupa *username*, *password*, data kantin, data makanan yang dijual, dan nota. “sitStatus” adalah *node* yang mengatur status reservasi dari tempat duduk yang ada.

B. Implementasi Proses Aplikasi



GAMBAR 2

Pada gambar 2 pengguna aplikasi dapat memilih untuk menggunakan aplikasi sebagai pelanggan atau penjual.

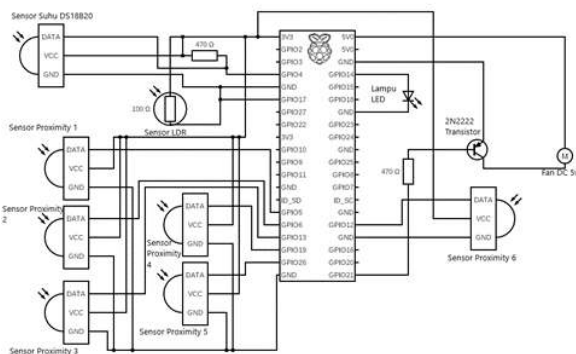
1. Pengguna Sebagai Penjual

Jika pengguna memilih penjual sebagai pilihan maka akan diarahkan menuju halaman *login* untuk penjual. Disini penjual harus memasukkan *username* dan *password* yang telah diberikan. Setiap penjual akan diberikan *username* dan *password* yang unik. Lalu jika *username* dan *password* yang dimasukkan benar dan menekan tombol login, penjual akan diarahkan ke menu utama untuk penjual. Pada menu utama ini, penjual dapat merubah foto kantinnya sesuai dengan keinginannya. Foto dapat diambil menggunakan kamera gawai atau dipilih melalui galeri. Penjual juga dapat merubah nama kantin dan merubah status buka atau tutupnya kantin. Dengan menekan salah satu makanan, penjual dapat mengubah data makanan yang dipilih. Penjual dapat mengubah gambar makanan, nama makanan, harga makanan, ketersediaan makanan, dan penjual juga dapat menghapus makanan tersebut dari menu. Dengan menekan tombol untuk menambahkan makanan, akan muncul tampilan untuk menambahkan makanan. Tampilan ini kurang lebih sama dengan tampilan mengubah makanan namun tanpa adanya data makanan yang ada. Ketika menekan tombol nota maka penjual akan diarahkan ke tampilan menu nota. Pada menu nota menampilkan seluruh nota pembeli di kios penjual yang bersangkutan. Nota dibagi menjadi 3 status yang berbeda yaitu butuh konfirmasi, sedang berjalan, dan selesai. Tampilan ini secara otomatis menyortir nota dari Firebase database ke kategori masing – masing sesuai statusnya. Setiap nota pada tampilan ini menampilkan nama pemesan, waktu nota dikonfirmasi, dan waktu pesanan selesai dikerjakan. Ketika penjual menekan tombol detail pada nota yang dipilih maka akan menampilkan detail dari nota yang terpilih. Tampilan ini akan menampilkan, makanan yang dipilih, jumlah makanan yang dipilih, harga setiap makanan, total dari seluruh harga, tempat duduk yang dipilih, status dari nota, dan nama dari pemesan. Penjual dapat mengubah status

dari nota pada tampilan ini dengan menekan tombol ubah status.

2. Pengguna Sebagai Pembeli

Jika pengguna memilih untuk menggunakan aplikasi sebagai pelanggan maka akan diarah ke tampilan menu utama pelanggan. Pada tampilan ini pelanggan dapat memilih kantin yang diinginkan. Dengan menekan info kondisi kantin, maka akan memunculkan tampilan dari informasi kondisi kantin. Pada tampilan ini menampilkan suhu dari kantin, dan tingkat keterangan dari area kantin. Dengan menekan tombol ubah pada informasi suhu maka pembeli dapat mengatur kipas angin mati atau tidak. Dengan menekan tombol ubah pada tingkat keterangan area kantin maka pembeli dapat mengatur tingkat keterangan pada lampu kantin. Dengan menekan tombol reservasi pada menu utama akan menampilkan tampilan untuk melakukan reservasi. Jika pelanggan belum melakukan reservasi, maka pelanggan dapat melakukan reservasi dengan menekan tombol reservasi pada tampilan. Jika pelanggan sudah melakukan reservasi, maka akan ditampilkan kursi yang dipeservasi dan waktu reservasinya. Setelah pelanggan memilih kios yang dipilih, maka pelanggan akan diarahkan ke tampilan menu makanan. Pada tampilan ini pelanggan dapat memilih makanan yang tersedia pada kantin yang dipilih. Setelah memilih makanan, pelanggan dapat merubah jumlah pembelian makanan yang sudah dipilih. Jika pelanggan sudah menentukan pilihan makanannya dan menekan tombol konfirmasi, maka pelanggan akan diarah ke tampilan pilih tempat.



GAMBAR 3

Pada tampilan ini pelanggan memilih tempat pada kantin sesuai pada pemetaan tempat yang ada pada aplikasi. Ketika pelanggan akan melakukan reservasi, maka juga akan diarahkan ke tampilan ini. Setelah pelanggan menekan tombol konfirmasi dari tampilan pemilihan tempat duduk, maka akan diarah ke tampilan nota. Pada tampilan ini, pelanggan ditampilkan pilihan makanan yang telah dipilih beserta jumlah, total harga, dan tempat duduk yang telah dipilih. Untuk melanjutkan ke proses transaksi pelanggan harus memasukkan nama. Setelah pelanggan memasukkan nama dan menekan tombol konfirmasi, maka akan muncul *dialog box* yang memberi pilihan kepada pelanggan untuk membayar langsung ke kios atau melalui pembayaran secara *online*. Jika pelanggan memilih pembayaran melalui kios, maka pelanggan harus membayar di kios yang dipilih. Jika pelanggan memilih untuk membayar secara *online*, maka akan diarahkan ke tampilan pembayaran *online*. Pada tampilan ini pelanggan dapat membayar sesuai dengan *e-payment* yang digunakan oleh pelanggan. Setelah pelanggan melakukan transaksi baik melalui kios atau secara *online*. Maka akan muncul notifikasi yang menunjukkan tempat

duduk yang dipilih, proses pengerjaan pesanan, dan batas waktu pelanggannya bisa menempati tempat duduk.

3. Implementasi Rangkaian

Pada gambar 3 menunjukkan rangkaian, terdapat 3 jenis sensor yaitu sensor suhu DS18B20, sensor Infra Red Proximity dan sensor LDR sebagai input. Dan terdapat 2 output yaitu lampu LED dan kipas angin DC 5V. Transistor 2N2222 berfungsi sebagai pemutus dan penyambung kipas angin. VCC pada sensor DS18B20 disambung ke pin power 3,3V. Pada sensor suhu DS18B20, VCC dan DATA disambung menggunakan resistor 470 Ohm agar data suhu yang diambil oleh sensor dapat terbaca oleh Raspberry Pi. Input dari suhu yang didapat oleh sensor disambung ke Raspberry Pi melalui pin 7 atau GPIO4. Sensor LDR disambungkan pada pin power 3,3V. Pada bagian ground sensor LDR terhubung dengan pin ground dan pin 11 atau GPIO17 pada Raspberry Pi. Untuk lampu LED, pada sisi positif terhubung dengan pin 8 atau GPIO14 dan sisi negatif terhubung pada pin ground. Untuk kipas angin DC 5V, pada kabel power terhubung dengan pin power 5V dan pada kabel ground terhubung dengan bagian emitter transistor 2N2222. Pada bagian base transistor terhubung dengan pin 40 atau GPIO21 melalui resistor 470 Ohm dan pada bagian collector transistor terhubung dengan ground. Terdapat 6 sensor Infra Red Proximity. Sensor ini memiliki 3 pin penghubung yaitu VCC, data, dan *ground*. Setiap pin data pada sensor terhubung ke pin 29, 31, 32, 33, 35, 37, 39. VCC terhubung pada power 3.3V dan pin *ground* terhubung dengan *ground*.

4. Pemrograman Pada Raspberry Pi

Pada Raspberry Pi 3 Model B ditulis sebuah program untuk mendapatkan data *input* dari sensor dan data *output* dari *Firestore Database*.

```
1 import RPi.GPIO as GPIO
2 import time
3 import pyrebase
4 from w1thermsensor import W1ThermSensor
5
6 GPIO.setmode(GPIO.BOARD)
7
8 fan = 40
9 ldr = 11
10 sensor = W1ThermSensor()
11 led = 8
```

GAMBAR 4

Kode pada gambar 4 adalah kode untuk menginisialisasi pin yang digunakan pada Raspberry Pi. Diawal dilakukan import beberapa *library* agar kode dapat berjalan. Lalu menulis `GPIO.setmode(GPIO.BOARD)` agar kode membaca pin yang dipakai melalui urutan angka yang tertera pada tiap pin. Selanjutnya menuliskan variabel yang berisi angka *integer* sesuai dengan pin yang dipakai.

```
config = {
    "apiKey": "AIzaSyB...",
    "authDomain": "...",
    "databaseURL": "https://...",
    "storageBucket": "..."
}

firebase = pyrebase.initialize_app(config)
db = firebase.database()
```

GAMBAR 5

Setelah dilakukan inisialisasi Langkah selanjutnya adalah menuliskan data konfigurasi yang digunakan untuk menghubungkan *Firestore Database* dan Raspberry Pi. Data – data yang

ditulis adalah *API key* dari *database*, *authorization domain*, *database URL*, dan *database storage bucket URL*. Lalu melakukan inisialisasi pada data konfigurasi yang sudah ditulis. Selanjutnya adalah kode untuk menginisialisasi fungsi PWM (*Pulse Width Modulation*) untuk lampu LED. Penggunaan PWM pada kode difungsikan untuk mengatur tingkat kecerahan dari lampu LED. Tingkat kecerahan lampu LED dapat diatur dari jangkauan 0 sampai 100 persen.

```

31 def ldr_value (ldr):
32     count = 0
33
34     GPIO.setup(ldr,GPIO.OUT)
35     GPIO.output(ldr,GPIO.LOW)
36
37     time.sleep(1)
38
39     GPIO.setup(ldr,GPIO.IN)
40
41     while (GPIO.input(ldr) == GPIO.LOW):
42         count +=1
43
44     return count

```

GAMBAR 6

Selanjutnya, membuat sebuah fungsi untuk mendapatkan data tingkat keterangan cahaya dari sensor LDR. Data dari sensor LDR didapatkan dengan cara mengkalkulasi besar resistansi dari sensor LDR.

```

46 try:
47     while True :
48         print (ldr_value(ldr))
49         temp = sensor.get_temperature()
50         print (temp)
51         db.child("canteenStatus").child("temp").set(temp)
52         if ldr_value(ldr) > 400 :
53             db.child("canteenStatus").child("terang").set("Terang")
54         else :
55             db.child("canteenStatus").child("terang").set("Redup")
56         fan_db = db.child("canteenStatus").child("fan").get()
57         if fan_db.val() == 0 :
58             GPIO.output(fan, GPIO.LOW)
59             print ("Kipas Mati")
60         elif fan_db.val() == 1 :
61             GPIO.output(fan, GPIO.HIGH)
62             print ("Kipas Menyala")
63         temp_db = db.child("canteenStatus").child("temp").get()

```

GAMBAR 7

Pada gambar 7 semua data yang didapat baik dari Firebase dan sensor diolah dan ditampilkan

No	Pilihan makanan	Tempat duduk	Total harga	Nama pelanggan	Status	Data
Nama	Jumlah	Harga makanan				
1	Food 3	2	30000	A1	30000	Percobaan 1 Berhasil Cocok
2	yul	2	30000	A2	45000	Percobaan 2 Berhasil Cocok
	Food 2	1	15000			
3	Food 2	1	15000	A3	75000	Percobaan 3 Berhasil Cocok
	Food 3	4	60000			
4	yul	5	75000	A1	150000	Percobaan 4 Berhasil Cocok
	Food 3	5	75000			
5	yul	1	15000	B2	45000	Percobaan 5 Berhasil Cocok
	Food 2	1	15000			
	Food 3	1	15000			

pada *console* dari Raspberry Pi.

C. Pengujian

1. Pengujian Pengiriman Data Dari Pembeli ke Penjual

Tujuan dari pengujian ini untuk memastikan bahwa data dari pelanggan terkirim ke aplikasi penjual dan memiliki data yang sama. Pengujian ini dilakukan dengan Melakukan pengiriman data dari aplikasi untuk pembeli ke aplikasi untuk penjual Hasil kesimpulan dari 5 kali pengujian yang dilakukan adalah dari data dari aplikasi untuk pembeli yang dikirim ke aplikasi untuk penjual memiliki hasil yang sama tanpa ada kekurangan data atau kesalahan pada proses pengiriman data.

2. Pengujian Fitur Reservasi

Tujuan dari pengujian ini adalah memastikan bahwa data reservasi terkirim pada Firebase database dan melakukan pencocokan antara data dari aplikasi dan data dari Firebase database. Pengujian ini dilakukan dengan melakukan reservasi di aplikasi dan mencocokkan data dari aplikasi dan Firebase.

TABEL 1

Pengujian ke-	Tempat duduk	Waktu mulai	Waktu selesai
1	B1	10:00	11:00
2	A3	08:00	09:00
3	B3	13:30	15:00
4	B2	13:00	13:15
5	A1	08:00	08:30

Tabel diatas merupakan data pilihan reservasi dari aplikasi. Data dipilih berbeda pada setiap pengujian. Data-data tersebut selanjutnya dikirim untuk dilakukan validasi dengan data pada Firebase dan informasi reservasi pada aplikasi.

TABEL 2

Pengujian ke-	Tempat duduk	Waktu mulai	Waktu selesai	Validasi data
1	B1	Senin Agustus 01 10:00:18 GMT+07:00 2022	Senin Agustus 01 11:00:18 GMT+07:00 2022	Cocok
2	A3	Senin Agustus 01 08:00:41 GMT+07:00 2022	Senin Agustus 01 09:00:18 GMT+07:00 2022	Cocok
3	B3	Selasa Agustus 02 13:30:41 GMT+07:00 2022	Selasa Agustus 02 15:00:18 GMT+07:00 2022	Cocok
4	B2	Kamis Agustus 04 08:00:14 GMT+07:00 2022	Kamis Agustus 04 13:15:14 GMT+07:00 2022	Cocok
5	A1	Kamis Agustus 04 08:00:14 GMT+07:00 2022	Kamis Agustus 04 08:30:14 GMT+07:00 2022	Cocok

Tabel diatas merupakan data reservasi yang didapat dari Firebase setelah data dari aplikasi dikonfirmasi dan dikirim menuju *database*.

Hasil analisa dari 5 kali pengujian yang telah dilakukan adalah data reservasi yang dikirimkan aplikasi memiliki kecocokan yang sama dengan data yang diperoleh pada Firebase database.

3. Pengujian Pengiriman dan Validasi Data Input

Tujuan pengujian ini untuk menguji pengiriman data dari input sensor LDR dan sensor DS18B20 menuju Raspberry Pi, Firebase database, dan aplikasi. Lalu melakukan validasi dengan mencocokkan data dari Raspberry Pi, Firebase database, dan aplikasi. Pengujian ini dilakukan dengan menjalankan program pada Raspberry Pi dan memonitoring data yang ada pada Raspberry Pi lalu mencocokkan data tersebut dengan data yang didapat oleh aplikasi.

Pengujian ke-	Data Sensor LDR	Data Sensor suhu	Data suhu Firebase	Data cahaya Firebase	Data suhu aplikasi	Data cahaya aplikasi	Validasi Data
1	1472	31.1875	31.1875	Terang	31.1875	Terang	Cocok
2	1269	31.4375	31.4375	Terang	31.4375	Terang	Cocok
3	276	31.5	31.5	Redup	31.5	Redup	Cocok
4	674	31.625	31.625	Redup	31.62	Redup	Tidak Cocok
5	3113	31.75	31.75	Terang	31.75	Terang	Cocok
6	1066	31.8125	31.75	Terang	31.75	Terang	Tidak Cocok
7	3301	31.875	31.875	Terang	31.875	Terang	Cocok
8	198	31.875	31.875	Redup	31.875	Redup	Cocok
9	206	31.9375	31.9375	Redup	31.9375	Redup	Cocok
10	956	32.0	32.0	Terang	32.0	Terang	Cocok

Tabel berikut merupakan tabel hasil pengujian yang dilakukan. Hasil analisa pada pengujian ini mendapati 8 dari 10 data cocok dengan satu sama lain. Ketidacocokan data yang terjadi akibat faktor kecepatan konektivitas jaringan internet. Ketika kecepatan internet yang didapatkan oleh perangkat Raspberry Pi lambat dapat menyebabkan data input pada Raspberry Pi tidak dikirimkan secara normal menuju Firebase database.

Pengujian ke-	Total bayar	Metode pembayaran	Status pembayaran	ID Pembayaran
1	9000	ShopeePay	Berhasil	IOT-Bill-1659556941857
2	9000	GoPay	Berhasil	IOT-Bill-1659558289241
3	22000	GoPay	Berhasil	IOT-Bill-1660912360119
4	30000	ShopeePay	Berhasil	IOT-Bill-1660912466215
5	12000	GoPay	Berhasil	IOT-Bill-1660912576138

4. Pengujian Pengiriman dan Validasi Data Output

Tujuan pengujian ini adalah untuk melakukan validasi data output yang dikirimkan melalui aplikasi. Validasi dilakukan dengan mencocokkan pilihan data output dari aplikasi dengan data dari Firebase database dan Raspberry Pi. Pengujian dilakukan dengan memilih opsi konfigurasi *output* yang ada pada aplikasi secara berbeda pada setiap pengujian.

No	Komponen yang dikonfigurasi	Data aplikasi	Data Firebase	Data Raspberry Pi	Kecocokan
1	Kipas Angin	Menyalakan (menyalakan)	1 (menyalakan)	Menyalakan	Cocok
2	Kipas Angin	Mati	0 (mati)	Mati	Cocok
3	Lampu LED	Redup	1 (redup)	Redup	Cocok
4	Lampu LED	Terang	2 (terang)	Terang	Cocok
5	Lampu LED	Mati	0 (mati)	Mati	Cocok

Tabel diatas menunjukkan hasil komparasi antara data yang dikirimkan aplikasi dan data yang didapat pada Raspberry Pi.

Hasil analisis dari pengujian ini adalah data yang dikirim dari aplikasi untuk mengubah hasil *output* dari komponen *output* cocok dengan data yang diperoleh Firebase database dan Raspberry Pi.

5. Pengujian Pembayaran Secara Online

Tujuan pengujian ini adalah untuk menguji tingkat keberhasilan pembayaran secara *online* dengan menggunakan beberapa metode *e-payment*. Pengujian dilakukan dengan memesan makanan dan memilih pembayaran secara *online* untuk metode pembayaran. Hasil analisis dari pengujian kali ini adalah dari 5 hasil pengujian yang dilakukan dengan data yang berbeda berhasil mendapatkan status transaksi sukses. Namun, kecepatan *loading* untuk transaksi dipengaruhi oleh kecepatan jaringan *internet* yang ada pada gawai terkait. *Delay* yang terjadi pada proses bisa berlangsung dari 1 sampai 5 detik tergantung dengan kecepatan koneksi pada gawai.

V. KESIMPULAN

Berikut adalah kesimpulan hasil dari pengujian dan penelitian yang dilakukan pada pengerjaan Proyek Akhir ini:

1. Pembayaran online yang dilakukan melalui Midtrans payment gateway memberi opsi pembayaran melalui ShopeePay dan GoPay. Penggunaan pembayaran secara online dipengaruhi oleh kecepatan koneksi internet pada gawai. Sehingga, gawai yang memiliki koneksi internet lambat dapat mendapatkan error pada aplikasi. Jika pembayaran tidak memungkinkan, pelanggan dapat memilih metode membayar langsung ke kios.
2. Hasil dari pengujian yang dilakukan, pelanggan dapat mengatur konfigurasi output dari beberapa komponen sistem IoT For Canteen. Sehingga pelanggan dapat mengatur komponen output sesuai dengan preferensi pelanggan sendiri.
3. Hasil dari pengujian yang dilakukan, sistem reservasi dapat membantu pelanggan untuk memilih tempat sesuai yang mereka inginkan. Sistem pemilihan tempat dan reservasi juga membantu pelanggan melihat ada berapa meja yang kosong dengan menggunakan sensor proximity.

REFERENSI

- [1] "The Canteen Counter: helping people avoid lunch line since 2018." <https://medium.com/deloittedigitalnorway/the-canteen-counter-helping-people-avoid-lunch-line-since-2018-c6827043f4ca> (accessed Apr. 11, 2020).
- [2] "Bosch Singapore: How to lunch in comfort with the IoT - Bosch.IO." <https://bosch.io/customers/building/bosch-singapore-how-to-lunch-in-comfort-with-the-iot/> (accessed Apr. 11, 2020).
- [3] M. Patil, A. Agrawal, and M. Borkar, "IoT based Smart Cafeteria Management System," *Proc. 5th Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2021*, pp. 155–160, 2021, doi: 10.1109/I-SMAC52330.2021.9641019.
- [4] R. Mothukuri, S. H. Raju, S. Adinarayna, V. C. Jadala, S. F. Waris, and G. S. Rao, "Smart Food Fare Canteen: Automation of Bills and Serving," pp. 475–486, 2022, doi: 10.1007/978-981-16-5157-1_37.
- [5] D. Untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana Teknik Disusun oleh, "PROTOTYPE KANTIN ELEKTRONIK BERBASIS INTERNET OF THINGS (IOT) DI UNIVERSITAS ISLAM MALANG SKRIPSI".

