Pengembangan Aplikasi *Mobile* Pemesanan dan Pembayaran Makanan Berbasis *Cloud Storage*

http://dx.doi.org/10.28932/jutisi.v8i1.4393

Riwayat Artikel
Received: 18 Januari 2022 | *Final Revision*: 9 Maret 2022 | *Accepted*: 11 Maret 2022

Erico Darmawan Handoyo [#1], Sulaeman Santoso [#2], Daniel Jahja Surjawan ✉ [#3]

[#] *S1 Teknik Informatika,Universitas Kristen Maranatha*
Jl. Prof. Drg. Suria Sumantri No. 65, Bandung, Indonesia
[1]erico.dh@it.maranatha.edu
[2]sulaeman.santoso@it.maranatha.edu
[3]daniel.js@it.maranatha.edu

Abstract —**Most dining places or restaurants generally still utilize manual food ordering process. The waiters will approach the customers to inquire about the food to be ordered. However, during this soaring covid pandemic, the owners are afraid of the virus spread through waiters who must approach the customers. For this reason, a mobile application will be made that can provide self-service food orders by customers without having to interact with waiters. Most digitalization efforts only focus on the client aspects of the ordering process and not on the processing of those orders (in the kitchen). This research studied an alternative digital ordering system that integrated the process on the client side and the kitchen side**

Keywords— **Digitalization; Food Ordering; Mobile Application; Restaurant Management**

I. PENDAHULUAN

Pada masa pandemic Covid 19 yang terjadi saat penelitian ini, kebutuhan akan digitalisasi proses manual menjadi sesuatu yang amat penting karena interaksi antara manusia harus dikurangi. Salah satu bidang yang pada saat ini masih banyak menggunakan proses manual dan melibatkan banyak interaksi langsung manusia adalah proses pemesanan di tempat makan/restoran.

Pada umumnya solusi digitalisasi untuk restoran/tempat makan yang ditawarkan hanya mencakup aspek pemesanan makanan saja dan tidak mencakup sistem pemrosesan pesanan di dapur. Saat ini banyak pemilik restoran yang merasa khawatir terjadinya penyebaran Covid melalui pelayan yang berhubungan langsung dengan pelanggan saat menanyakan makanan yang dipesan. Selain itu proses manual memiliki banyak kemungkinan kesalahan pencatatan dan juga melibatkan lebih banyak interaksi manusia yang sebenarnya dapat dibuat efisien.

Proses digitalisasi pemesanan di tempat makan melibatkan beberapa unsur. Unsur pertama adalah perangkat keras dimana berbagai macam jenis perangkat keras telah dicoba seperti misalnya penggunaan NFC, *tablet*, ataupun *web application* pada *desktop* atau perangkat lainnya. Pada penelitian ini akan menggunakan perangkat keras *mobile phone* yang pada saat ini telah menjadi standar perangkat keras yang juga digunakan oleh kebanyakan orang. Sehingga penyediaan perangkat keras tidak menjadi masalah. Selain itu, perangkat keras yang mahal juga dapat menjadi masalah tersendiri bagi rumah makan kecil ataupun kantin, karena penyediaan alat tersebut akan memakan modal yang cukup besar. Unsur kedua adalah integrasi sistem dengan bisnis proses yang terjadi pada sebuah tempat makan/restoran. Sebuah sistem pemesanan digital yang terintegrasi dengan proses bisnis restoran/ tempat makan akan sangat membantu kinerja sumber daya yang ada menjadi lebih efisien. Unsur ketiga yang terlibat adalah portabilitas. Sebuah sistem digital pada sebuah rumah makan/restoran akan memiliki kemiripan satu sama lain. Jadi dibanding mengembangkan sebuah sistem yang unik untuk setiap rumah makan/restoran, akan lebih efisien untuk mengembangkan sebuah sistem umum yang dapat digunakan untuk berbagai rumah makan, dan dapat dipasang dan dipindahkan dengan mudah apabila rumah makan memiliki cabang atau pindah tempat atau mengganti perangkat kerasnya.

Proses pemesanan makanan di sebuah rumah makan melibatkan pencatatan yang pada umumnya terjadi secara manual. Begitu pula untuk pemrosesan pesanan tersebut di bagian dapur juga terjadi secara manual. Tentu saja dengan digitalisasi

proses pemesanan dapat dipermudah dan diefisiensikan, sehingga penelitian untuk mendigitalisasi proses ini telah banyak dilakukan.

Salah satu pendekatan untuk mendigitalisasi proses ini adalah dengan menggunakan teknologi terbaru yang diharapkan memudahkan proses pemesanan. Agita Chrisnawa pada tahun 2009 [1] menggunakan microcontroller AT89552 sebagai perangkat keras pembantu untuk memesan makanan. Pada tahun 2015, Chavan Varsha menggunakan teknologi web untuk membantu pemesanan digital [2], web yang dibuat diharapkan akan diakses menggunakan perangkat telepon genggam.

Kuan Yu Lin [3] dan Annisa Yolanda [4] di tahun 2018 mencoba menggunakan NFC untuk mempermudah pemesanan makanan. Biasanya pendekatan ini memerlukan pengguna membawa alat tambahan yaitu sebuah kartu NFC. Namun pada penelitian ini mereka menggunakan teknologi NFC yang sudah terintegrasi dengan perangkat telepon genggam. Beberapa peneliti bahkan tidak berhenti pada perangkat keras untuk pemesanan tapi hingga menerapkan robot sebagai pelayan [5] [6].

Namun sebuah tema utama yang ditemukan pada semua penelitian ini adalah penggunaan perangkat lunak yang bisa diakses oleh sebanyak mungkin pengguna. Itu sebabnya banyak penelitian mulai memfokuskan pada peralatan telepon genggam sebagai perangkat keras pilihan, karena besarnya pengguna telepon genggam. Sebagai contohnya penelitian [7] [8] [9] [10] semuanya menggunakan media telepon genggam dengan berbagai variasi pada sistem yang dikembangkannya. Sebagai contoh penelitian oleh Vindya Linayage [8] menambahkan penggunaan *machine learning* untuk merekomendasi menu sesuai dengan profil pengguna. Renjith Ravi [9] menggunakan sistem *Bluetooth* untuk memindahkan informasi dari perangkat lunak pemesan ke perangkat lunak toko. Chochiang et al [7] menggunakan penyimpanan *cloud* (azure) untuk menyimpan data yang akan ditawarkan.

Aplikasi yang dikembangkan di penelitian ini akan fokus pada peralatan dengan sistem operasi *android* dan menggunakan sebuah *framework cross-platform* yaitu flutter framework. yang artinya dapat menggunakan telepon genggam, *web* atau pun peralatan sejenis. Flutter sendiri adalah sebuah *framework* pemrograman dengan Bahasa DART yang memudahkan pengembang untuk membuat aplikasi yang dapat berjalan di lebih dari satu jenis peralatan. Pada saat ini flutter dapat menggunakan aplikasi *mobile (*IOS, dan android), *web*, dan *desktop*.

Untuk penyimpanan, penelitian ini akan menggunakan *firebase firestore* dari google sebagai tempat penyimpanan berbasis awan (*cloud*). Pemilihan ini dikarenakan sifat dari firebase yang memberikan akses gratis sampai pemakaian tertentu, sehingga memungkinkan aplikasi ini untuk digunakan oleh rumah makan kecil ataupun kantin yang mungkin tidak memiliki modal yang besar.

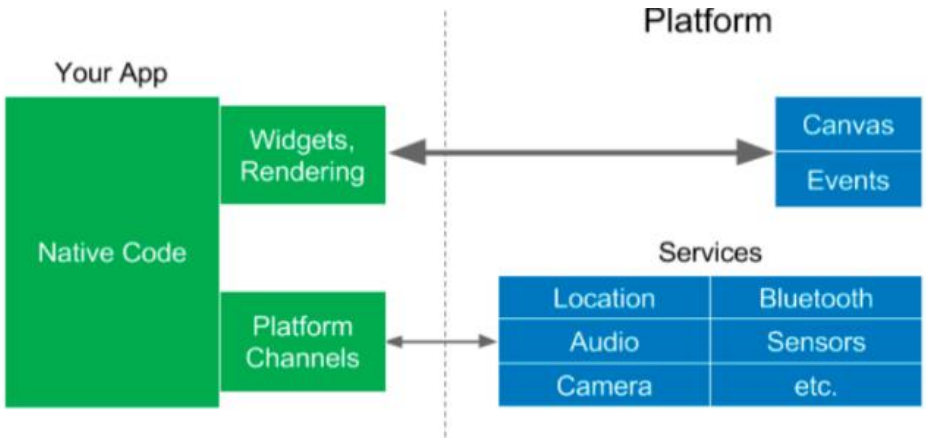
Menurut Nazaruddin [11] android merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam-macam peranti bergerak. Android umum digunakan di *smartphone* dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple, dan BlackBerry OS. Android dikembangkan oleh *Google Inc*. Android merupakan sistem operasi berbasis Linux dan bersifat *open source*. Awalnya sistem operasi android dikembangkan untuk ponsel pintar dan komputer tablet dengan antarmuka layar sentuh. Saat ini sistem operasi android juga dikembangkan untuk perangkat keras lain seperti kamera digital, jam tangan, perangkat navigasi, televisi, dan kacamata pintar.Android banyak diminati pengguna ponsel pintar karena sifatnya yang *open source* membuat pengembang aplikasi tertarik untuk mengembangkan aplikasi berbasis sistem operasi android. Saat ini terdapat lebih dari 1 juta aplikasi yang dapat diunduh pengguna android melalui layanan toko aplikasi yang dinamakan *Google Play Store*.Menurut Teguh Arifianto [12] android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis Linux.Akan tetapi, sistem operasi yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli dari ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

Flutter adalah sebuah framework multi platform yang dikembangkan oleh tim di Google [13]. Flutter bertujuan untuk menyederhanakan pengembangan perangkat lunak multiplatform dengan satu *code base*. Hal ini juga berlaku untuk pemisahan UI dan code yang biasa terdapat pada bahsa bahasa pengembangan yang lain. Flutter membuat satu codebase yang cukup untuk UI dan logic. Flutter mengimplementasikan kodenya dengan *widget*. *Widget* di dalam flutter dapat berupa komponen *visual* maupun sekedar penampung bagi widget yang lainnya. Dengan demikian, flutter memiliki kode yang bersifat hierarki. Gambar 1 menunjukkan contoh kode pada flutter.

```
@override
Widget build(BuildContext context) {
  return Scaffold(
  ├─appBar: AppBar(
  │ └─title: Text(widget.title),
  │ ), // AppBar
  ├─body: Center(
  │ └─child: Column(
  │   │ mainAxisAlignment: MainAxisAlignment.center,
  │   │ children: <Widget>[
  │   ├──Text(
  │   │     'You have pushed the button this many times:',
  │   │   ), // Text
  │   └──Text(
  │         '$_counter',
  │         style: Theme.of(context).textTheme.display1,
  │       ), // Text
  │     ], // <Widget>[]
  │   ), // Column
  │ ), // Center
```

Gambar 1. Contoh Kode Pada Flutter

Yang menjadi pembeda antara flutter dengan solusi multiplatform yang lain adalah karena flutter tidak menggunakan penyambung seperti pendekatan multiplatform yang lain. Gambar 2 menunjukkan gambaran besar dari pendekatan flutter terhadap solusi multiplatform Dengan struktur ini perubahan platform tidak akan mengganggu kinerja dari aplikasi. Dan tidak ada jembatan yang berpotensi menjadi *bottleneck.*



Gambar 2. Struktur Arsitektur Flutter [14]

Keunggulan flutter yang lain adalah ketersediaannya pustaka pustaka kode siap pakai yang dapat disertakan beserta dengan kode flutter yang dimiliki untuk menambahkan fungsionalitas khusus. Pustaka ini tersedia dalam bentuk *widget* yang dapat ditambahkan dengan menambahkan pustaka tersebut pada file .yaml yang ada pada setiap proyek flutter. Sehingga dengan demikian pengembang *flutter* dapat saling bekerjasama dan tidak perlu membuat segala sesuatu dari awal kembali. Salah satu pustaka yang digunakan pada penelitian ini adalah pustaka yang menyambungkan aplikasi *flutter* dengan *firebase database*.

Dalam sebuah aplikasi yang memiliki data di *cloud*, pembaharuan data terjadi pada umumnya ketika user memberikan *request* ke penyimpanan *cloud*. Sehingga data yang tampil pada aplikasi pengguna belum tentu merupakan data yang paling baru. Dengan menggunakan stream, perubahan pada *database cloud* akan langsung terlihat pada aplikasi pengguna.

Apabila diibaratkan, sebuah *stream* di flutter adalah sebuah pipa dimana ketika sebuah nilai dimasukkan di ujung kesatu maka nilai tersebut akan mengalir keluar di ujung yang lain yang kita sebut *listener* [15]. Sebuah *stream* dapat memiliki lebih dari ujung dan setiap ujung itu akan dapat menggunakan nilai yang sama yang telah disampaikan dari sumber.

Gambar 3 menunjukkan penggunaan *stream builder* di dalam flutter. *Streambuilder* ini akan menerima *stream* yang dapat diambil dari database atau *service* di *cloud*. Ketika sebuah data diterima karena perubahan yang terjadi di *database* atau *service* di *cloud*, maka *streambuilder* akan membangun kembali *widget* yang ada di bawahnya sehingga data di aplikasi pengguna akan serupa dengan data pada cloud.

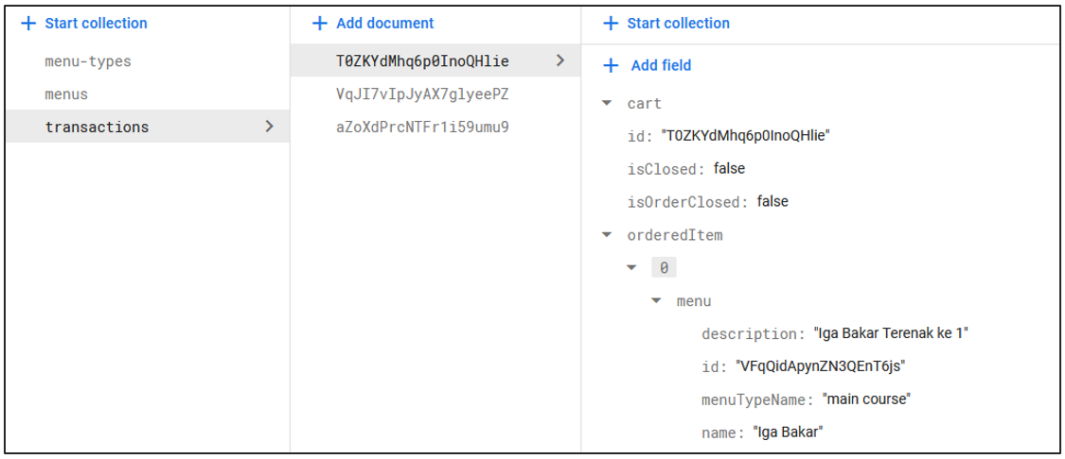
```
if (documentSnapshot != null)
  StreamBuilder<firestore.DocumentSnapshot>(
      stream: documentSnapshot,
      builder: (_, snapshot) {
        if (snapshot.hasData) {
          Transaction tr = TransactionJson(
                  snapshot.data!.data() as Map<String, dynamic>)
              .toTransaction();

          return Column(
            children: [
                  Text(tr.id.toString()),
                  Text(tr.isClosed.toString()),
                  Text('CART')
                ] +
```

Gambar 3. Penggunaan *StreamBuilder* di dalam Flutter

Penyimpanan data tradisional biasa dilakukan pada komputer yang terdapat di lokal. Penyimpanan data lokal ini memiliki beberapa kekurangan yaitu bahwa data yang tersimpan rawan untuk hilang ataupun rusak karena perangkat keras. Hal ini dapat diatasi dengan menggunakan penyimpanan data berbasis *cloud*. Data yang disimpan di dalam *cloud* akan tersimpan pada beberapa komputer *server* dari penyedia. Hal ini menyebabkan data tersebut lebih aman terhadap kerusakan dan dapat selalu tersedia dimanapun (tidak tergantung oleh lokasi). Beberapa penyedia layanan *cloud storage* diantaranya adalah *Amazon AWS*, *Google Firebase*, dan lain lain.

Google Firebase adalah sebuah layanan *cloud* yang disediakan oleh google. Layanan ini mencakup penyimpanan *cloud (firestore)*, *hosting* aplikasi, layanan *push messaging*, layanan otentikasi, dan lain lain [16]. Pada penelitian ini produk yang akan digunakan adalah *real time database* atau *firestore*.

Firestore adalah layanan google *cloud* untuk penyimpanan data dalam bentuk *database no-sql*. *Database no-sql* ini adalah tipe *database* tanpa relasi. Berbeda dengan *relational-database* dimana setiap data memiliki hubungan dengan data lainnya, *database no-sql* ini memiliki data yang tidak secara langsung berhubungan dengan data lainnya. Database tipe ini dirancang untuk menangani sejumlah besar data yang tidak terstruktur dengan cepat [17] [18]. Dengan data tersimpan di *cloud firestore data* akan dapat diakses dimanapun dan dengan stream, data dapat berubah secara *real-time*. Data pada *firestore* disimpan dalam bentuk *collection* yang bertingkat. Gambar 4 menunjukkan contoh sebuah struktur data pada *firestore*, dapat terlihat bahwa pada sebuah data transaksi terdapat data yang bertingkat sifatnya. Data transaksi akan memiliki sejumlah dokumen yang masing masing dokumennya akan memiliki lagi data yang bertingkat sifatnya. Contohnya adalah data orderedItem disini ordered item adalah barang yang dipesan oleh pengguna. Data ini terdiri dari menu yang dipesan dan juga harga serta jumlah pesanan. Sehingga data ordered item ini memiliki banyak data.

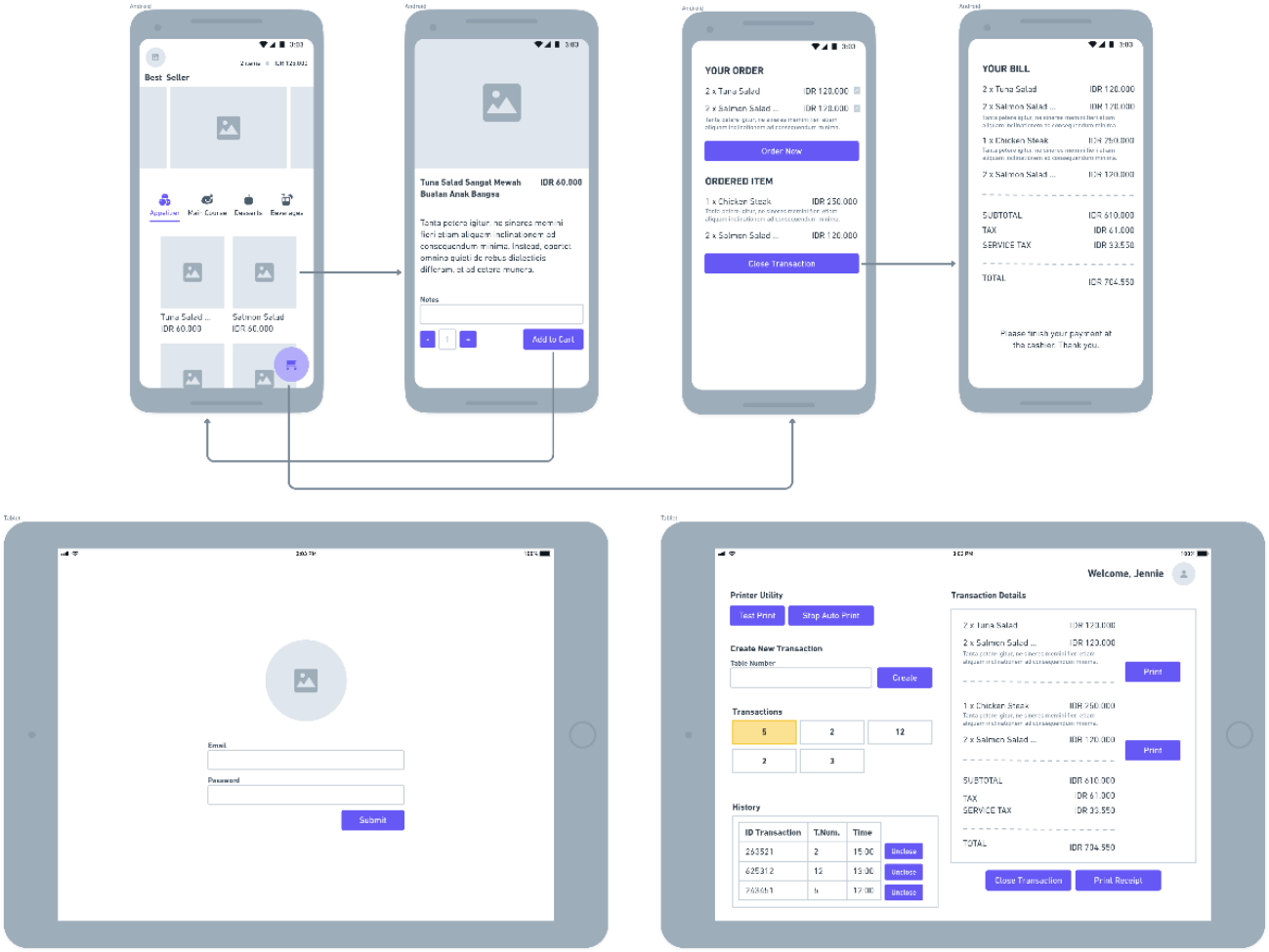


Gambar 4. Bentuk data pada Firestore

II. METODE PENELITIAN

A. Alur Sistem

Gambar 5 merupakan urut-urutan tampilan proses secara keseluruhan yang berjalan pada aplikasi mobile. Ada 2 pengguna yaitu kasir dan pelanggan pada aplikasi yang dibuat. Menu yang dapat diakasi oleh Kasir yaitu: *Submit Account, Test Print, Stop Auto Print, Create Table Number, Choose Table Number, Unclose Transaction History, Print Receipt, Close Transaction*. Sedangkan menu yang dapat diakses oleh pengguna yaitu: *Scan QR-Code Menu, Appetizer, Main Course, Dessert, Beverages, Write Notes, Add to Cart, Order Now, Close Transaction*.



Gambar 5. Tampilan Urutan Proses Sistem Kasir

B. Metode Penelitian

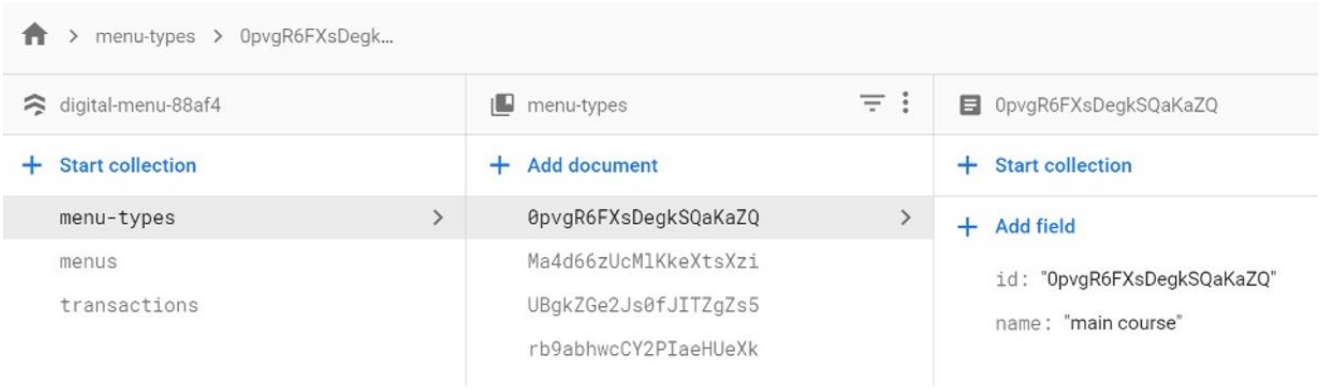
Metode penelitian yang digunakan dalam pembuatan sistem pemesanan digital ini terbagi menjadi beberapa tahap, yaitu sebagai berikut:

1. Pembentukan aplikasi *client*. Analisa dan pengembangan aplikasi *client* dengan menggunakan *flutter framework* yang akan dibagi menjadi dua bagian yaitu bagian pengguna dan bagian dapur.
2. Implementasi sistem. Tahap implementasi teknologi dan pembuatan sistem sesuai dengan analisis sistem operasi prosedur dan data yang sudah didapatkan dari tahapan metode sebelumnya.

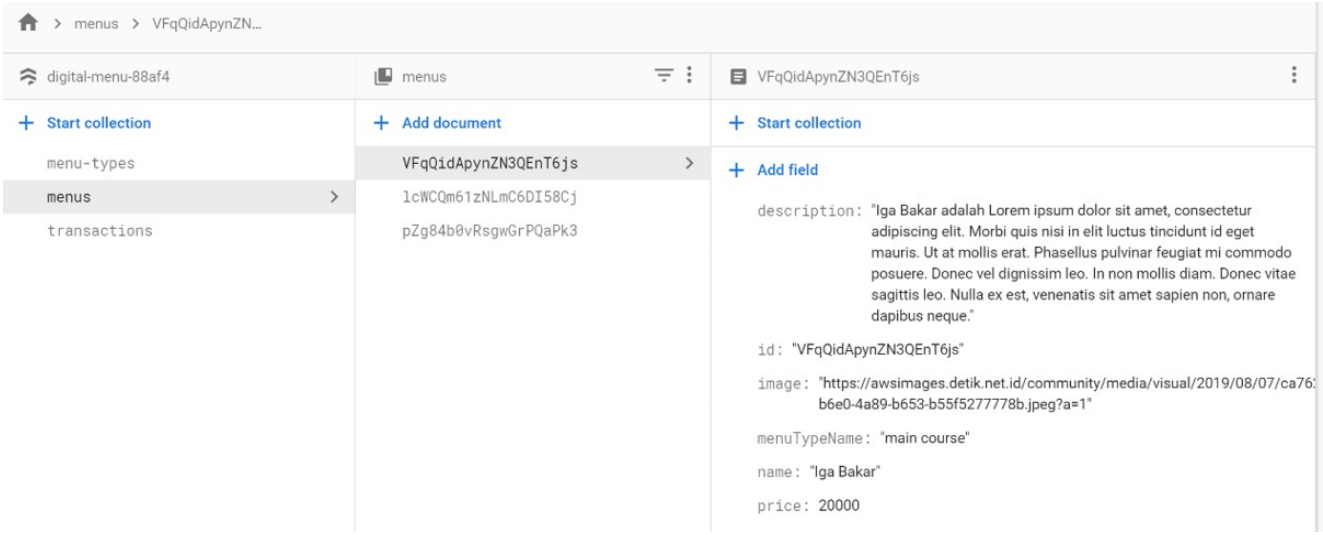
Evaluasi dan revisi system. Pengujian aplikasi akan dilakukan secara internal, sedangkan evaluasi akan dilakukan dengan cara mengundang relawan untuk berpartisipasi menggunakan sistem yang diusulkan. Revisi kemudian akan dilakukan secara bertahap sesuai dengan masukan pada tahap ini.

C. Basis Data

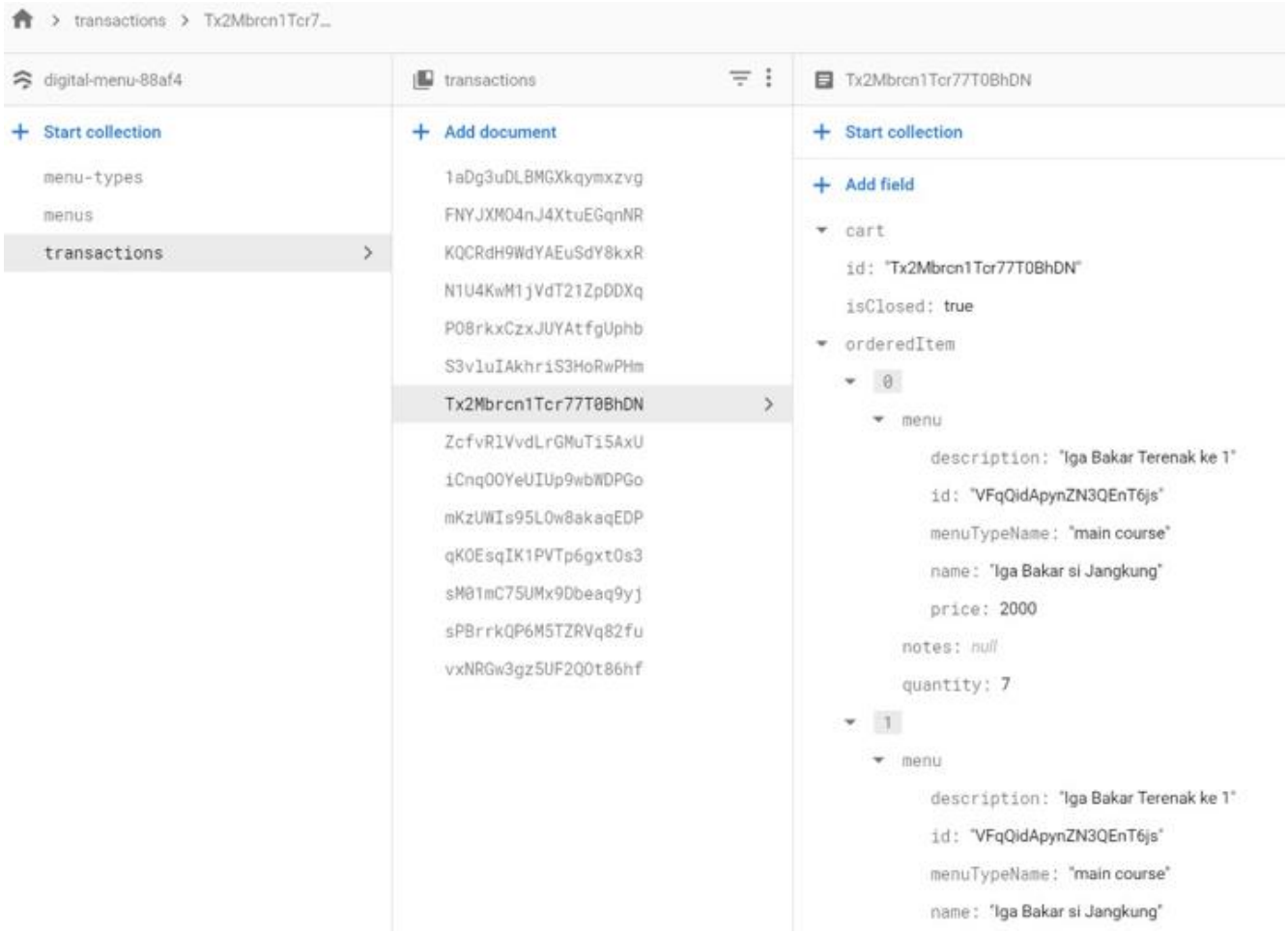
Perancangan data pada aplikasi ini menggunakan *no-sql database* seperti ditunjukkan pada Gambar 6, Gambar 7, Gambar 8. Saat ini *no-sql database* banyak digunakan oleh developer karena pertimbangan kecepatan yang lebih penting dibandingkan *storage*. Data yang redundan pun tidak menjadi masalah selama memiliki kecepatan yang lebih besar dari *storage*. Untuk *edit* dengan *no-sql database* memang lebih lama, namun kemampuan pencarian datanya sangat cepat dan akurat.



Gambar 6. Menu Types



Gambar 7. Menus



Gambar 8. Transactions

Database disini dibagi menjadi 3 dokumen besar yaitu *menu-types* (Gambar 6), *menus* (Gambar 7), dan *transaction* (Gambar 8). *Menu-types* adalah *database* yang menyimpan kategori menu / makanan yang ditawarkan. Sehingga rumah makan dapat menambahkan tipe menu tambahan atau mengubah tipe menu setiap saat (apabila diperlukan). Namun pengelolaan tipe menu ini tidak termasuk pada lingkup penelitian yang berfokus pada pemesanan kali ini.

Dokumen kedua menyimpan menu yang ditawarkan. Pada dokumen ini di simpanlah nama makanan, harga, deskripsi, dan detil detil lain yang diperlukan pelanggan. Penyimpanan gambar menu sendiri disimpan pada penyimpanan yang terpisah pada *firebase* sehingga yang disimpan pada dokumen ini adalah hanya *link* terhadap penyimpanan tersebut.
Dokumen terakhir menyimpan informasi transaksi yang dilakukan. Transaksi terbagi menjadi dua bagian besar yaitu transaksi yang terbuka (*open*) dimana pelanggan dapat memasukkan pesanan baru atau transaksi yang tertutup (*closed*) dimana pelanggan hanya dapat melihat data transaksi. Selain dari hal itu dokumen transaksi juga menyimpan dua bagian besar yaitu *orderedItem* dan *cart*. *Cart* atau kereta belanja akan menyimpan pesanan sementara dari pelanggan, sedangkan *orderedItem* akan menyimpan pesanan pelanggan yang sudah terkonfirmasi.

D. Delivery & Feedback

Hasil dari aplikasi *mobile* pemesanan dan pembayaran makanan berbasis *cloud storage* selanjutnya dilakukan survey untuk mendapatkan *feedback* dari pengguna dalam hal ini yaitu pembeli & kasir. Survey yang dilakukan berhubungan fungsionalitas dan tampilan dari aplikasi. Tabel 1 merupakan *feedback* dari Pembeli dan Tabel 2 adalah *feedback* dari Kasir.

TABEL 1
FEEDBACK PEMBELI

Komponen	Feedback Pengguna	Kesimpulan
Halaman Awal	Tampilan dan informasi restoran sudah jelas.*	Diterima
Halaman Menu Utama	- Fungsi tambah kurang jumlah makanan belum ada - Fungsi search belum bisa digunakan - Fungsi tombol keranjang belum berjalan - Fungsi filter kategori sudah berjalan - Menyarankan untuk menambahkan lebih banyak variasi kategori makanan	Perlu Dievaluasi
Halaman Detail Makanan	- Fungsi tambah kurang jumlah makanan belum ada - Fungsi *add to cart* sudah berjalan	Perlu Dievaluasi
Halaman Keranjang	- Fungsi tambah kurang jumlah makanan belum ada - Fungsi hapus makanan belum ada - Fungsi *order now* sudah berjalan	Perlu Dievaluasi
Halaman Pembayaran	Halaman sudah informatif dengan adanya rincian pesanan dan data lainnya yang mudah dibaca	Perlu Dievaluasi

TABEL 2
FEEDBACK KASIR

Komponen	Feedback Pengguna	Kesimpulan
Halaman Login	Input *email* dan *password* sudah memadai untuk kasir melakukan login. Informasi nama dan gambar restoran juga jelas.*	Diterima
Halaman Menu Kasir	- Menyarankan validasi apakah nomor meja yang diinput masih tersedia atau tidak ketika membuat transaksi - Fungsi printer sudah berjalan dengan baik - Daftar transaksi mudah dilihat dan digunakan - Daftar *history* informatif, tetapi disarankan dapat ditambah filter rentang waktu untuk memudahkan pencarian - Fungsi *unclose* sudah baik dengan adanya konfirmasi *no/yes*	Perlu dievaluasi
Halaman Detail Transaksi	- Detail transaksi sudah informatif dan mudah dibaca - Fungsi *print receipt* sudah berjalan dengan baik - Menyarankan penambahan konfirmasi *no/yes* ketika melakukan *close transaction*	Perlu dievaluasi

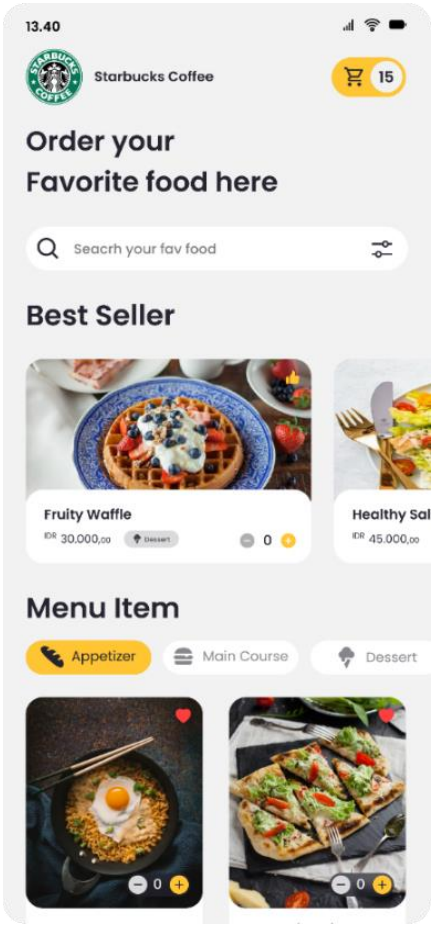
III. HASIL DAN PEMBAHASAN

A. Tampilan Pengguna

Penelitian telah sampai pada perancangan sistem dan pengembangan awal dari perangkat lunak. Perangkat lunak akan berpindah platform dari aplikasi *mobile* menjadi aplikasi web. Perubahan ini dilakukan untuk mengakomodasi pengguna baru yang mungkin merasa enggan untuk menginstall aplikasi untuk melakukan pemesanan makanan. Ketika seorang pelanggan memasuki tempat makan, ia akan terlebih dahulu menghampiri / menunggu kasir untuk mendapatkan *barcode*.

Gambar 9. Tampilan Barcode Pengguna

Gambar 9 merupakan barcode yang diberikan oleh kasir saat pelanggan sudah mendapatkan tempat duduk. Tampilan generate barcode oleh kasir akan dijelaskan pada sub bab berikutnya. Barcode ini adalah nomor transaksi pelanggan yang disimpan di *cloud*. Pelanggan akan melakukan scan barcode untuk masuk ke menu makanan yang tersedia di tempat makan dan untuk memesan. Apabila ada lebih dari satu pelanggan yang hendak membayar menggunakan satu bon maka barcode yang sama dapat digunakan. Disini, pelanggan yang melakukan scan pada barcode yang sama akan mendapatkan nomor transaksi yang sama sehingga akan masuk ke dalam satu bon yang sama. Hal ini dilakukan agar sekelompok pengguna dapat berbagai nomor transaksi tanpa perlu berbagai alat untuk memesan. Dengan ini, pelanggan dapat menggunakan alatnya masing masing dan tetap memiliki satu bon yang sama dalam pesanannya.



Gambar 10.Tampilan Menu

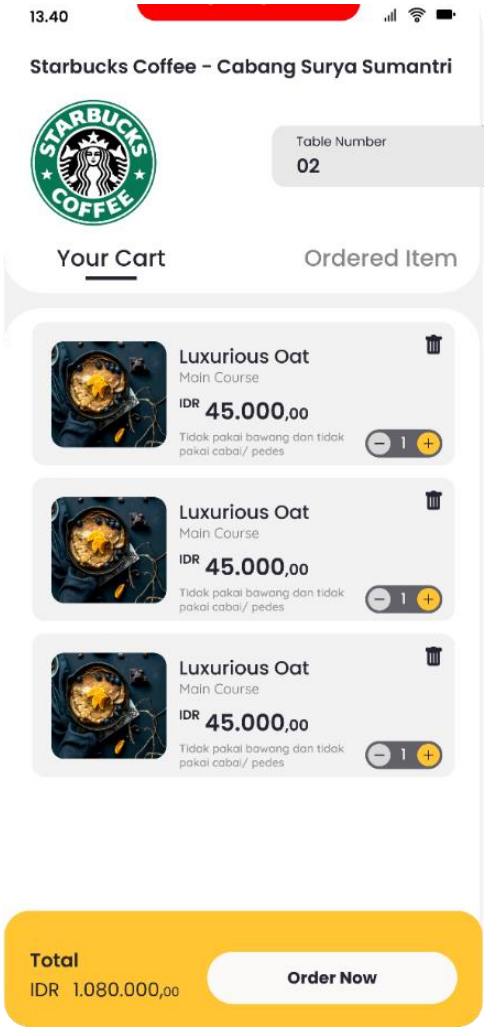
Gambar 10 menunjukkan tampilan awal sesudah pelanggan melakukan scan barcode. Tampilan awal ini ditujukan untuk langsung memberikan pengguna pilihan untuk pesanan. Pada tampilan *Best Seller* menampilkan beberapa menu yang penjualannya paling banyak, dan ditandai dengan tanda 👍 (jempol). Aplikasi secara otomatis akan menentukan menu yang penjualannya paling banyak dan menempatkannya di depan untuk memudahkan pemesanan. Selain itu juga terdapat tanda ❤️ yang berarti menu favorit yang dipilih oleh pelanggan. Pelanggan juga dapat melakukan pencarian makanan secara langsung di kolom *Search your fav food* yang berada di tampilan atas. Selain itu pelanggan juga dapat memilih menu berdasarkan kategori yang disediakan seperti *Appetizer*, *Main Course*, *Dessert*, dan *Beverages.* Ketika pelanggan hendak memesan maka ia dapat melakukannya dengan menekan menu untuk melihat detail menu yang tersedia sebagai pertimbangan pesanan ataupun untuk menyertakan pesanan khusus.

Gambar 11. Tampilan Pilih Menu

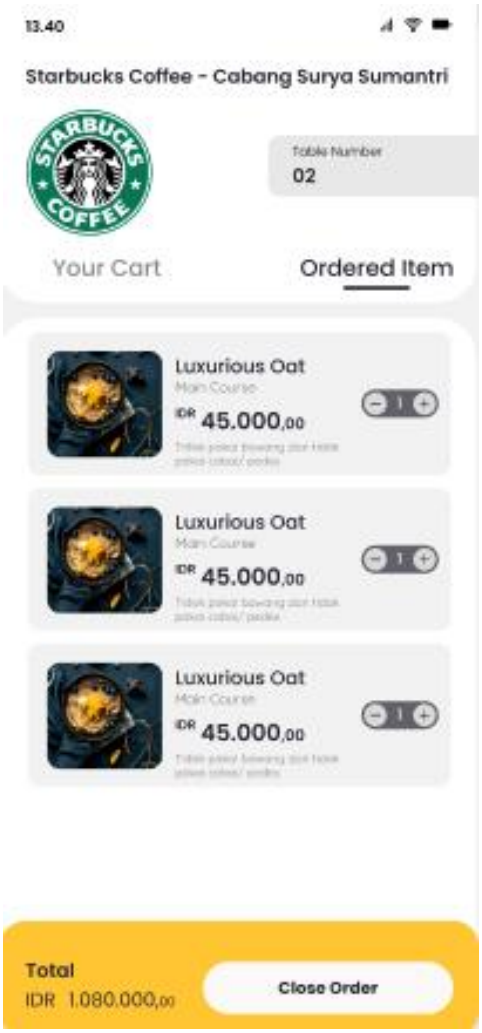
Gambar 11 menampilkan menu yang dipilih oleh pelanggan. Pada tampilan ini pelanggan bisa melakukan:

1. Melihat kategori dari menu yang dipilih, yaitu bisa terdiri dari *Appetizer*, *Main Course*, *Dessert*, dan *Beverages.*
2. Menentukan jumlah yang akan dipesan
3. Mendapatkan informasi lengkap dari menu
4. Menuliskan pesan/permintaan khusus terhadap menu yang akan dipesan.

Setelah itu pelanggan dapat klik tombol *Add to Cart* untuk menambahkan barang pada kereta belanja. Jumlah pesanan disini dan catatan pesanan khusus ini tersimpan dalam bentuk *stream* pada *database*. Yang artinya, apabila terjadi perubahan pesanan maka tampilan tiap alat yang memiliki nomor transaksi yang sama dan sedang melihat menu yang sama akan melihat perubahan yang sama. Hal ini memudahkan pelanggan untuk melakukan pesanan dan pesanan tersebut dapat juga dilihat dan diubah pada pelanggan lain yang memiliki nomor transaksi yang sama walaupun pada alat yang berbeda. Pesanan ini belum langsung diajukan ke kasir untuk dibuat. Tapi hanya ketika pelanggan menekan *order now* barulah pesanan tersebut disampaikan ke bagian kasir. Hal ini dilakukan untuk mencegah terjadinya kesalahan pembelian oleh pelanggan. Pelanggan dapat melihat total pesanan mereka pada halaman kereta belanja atau *cart.*



Gambar 12. Tampilan Pesanan Pelanggan

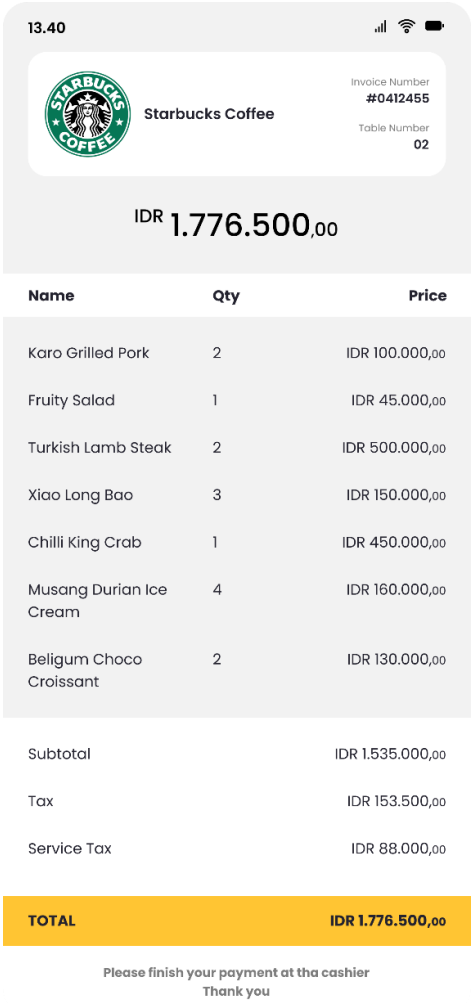


Gambar 13. Tampilan Akhir Pesanan Pelanggan

Gambar 12 merupakan menu *Your Cart* yang menunjukkan tampilan pesanan yang akan di order oleh pelanggan. Disini ketika lebih dari satu pelanggan masuk menggunakan barcode yang sama, maka semua pelanggan akan dapat mengakses status pesanan yang sama dan dapat melihat atau menambah pesanan yang belum dikonfirmasi, sedangkan pesanan yang belum terkonfirmasi hanya dapat dilihat. Pada menu ini juga data terhubung pada sebuah *stream*. Sehingga semua perubahan pada database akan tercerminkan pada aplikasi yang bersangkutan. Seperti yang telah disebutkan sebelumnya, apabila pelanggan dari menu ini menekan tombol order now, maka pesanan akan terkonfirmasi dan dikirimkan untuk dibuat. Setelah pesanan terkonfirmasi, pesanan tidak dapat dibatalkan dari sisi pelanggan.

Pelanggan dapat memastikan pesanan yang terkonfirmasi dengan masuk ke menu *Ordered Item* seperti tampilan padaGambar 13. Disini pelanggan mengakhiri pesanannya dan kemudian membayar pesanannya. Setelah pelanggan melakukan klik *Close Order*, karena pesanan ditutup melalui stream maka pelanggan tidak lagi dapat menambahkan pesanan baru dan setiap pelanggan yang memiliki nomor transaksi yang sama akan juga tidak dapat menambahkan pesanan kembali. Setiap pelanggan yang membuka aplikasi dengan melakukan scan barcode yang sama akan kemudian dialihkan pada tampilan total pembayaran. Pada tampilan ini, setiap pelanggan dapat melihat detail pesanan masing-masing untuk kemudian dapat melakukan pembayaran. Apabila terjadi kesalahan sehingga transaksi tertutup sebelum pelanggan menginginkannya, maka pelanggan dapat menghubungi pelayan untuk membuka kembali transaksi pada aplikasi kasir.

p-ISSN : 2443-2210
e-ISSN : 2443-2229



Gambar 14.Tampilan Total Pembayaran

Gambar 14 merupakan tampilan akhir di pelanggan yang menampilkan total yang harus dibayar oleh pelanggan yang bisa dilakukan langsung di kasir.

B. Tampilan Kasir

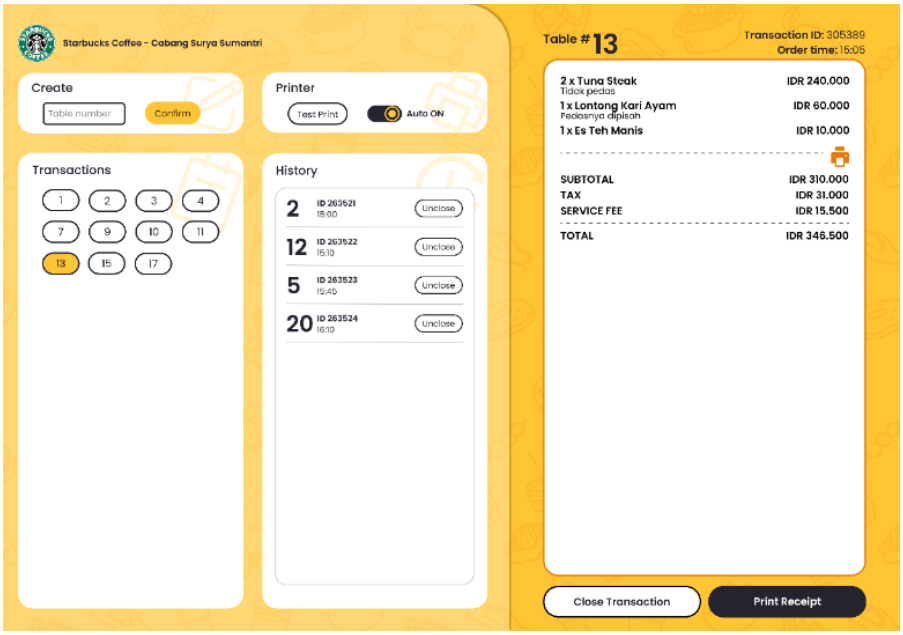
Sistem menu digital ini juga menyertakan aplikasi di sisi kasir, dimana kasir dapat membuka transaksi dan dapat melihat transaksi yang sedang berjalan. Berikut ini adalah beberapa menu dan penjelasan dari tampilan kasir:

1. *Create*. Menu ini digunakan kasir untuk membuat nomor meja yang baru jika terdapat penambahan meja untuk pelanggan. Pada menu inilah, barcode yang akan diberikan kepada pelanggan dibuat. Setelah menekan tombol *create* maka akan di cetaklah barcode transaksi yang baru.
2. *Transaction*. Menu ini akan menghasilkan barcode untuk pelanggan yang duduk di meja yang sudah dipilih. Barcode yang dihasilkan ini yang akan di-*scan* oleh pelanggan untuk melihat menu dan memesan makanan.
3. *Printer*. Menu ini untuk melakukan test cetak, ataupun membuat aktif atau tidak aktif *printer* yang terhubung pada aplikasi kasir.
4. *History*. Menu ini digunakan untuk menampilkan *history* dari semua pelanggan yang sudah selesai melakukan pembayaran dan juga untuk menyediakan pencetakan bon.
5. *Print Receipt*. Menu ini akan mencetak *invoice* saat pelanggan sudah melakukan pembayaran di kasir. Hal ini dapat dilakukan beberapa kali apabila diperlukan.

Gambar 15 merupakan tampilan keseluruhan dari menu kasir. Disini printer yang digunakan untuk percobaan adalah printer *bluetooth* yang terkoneksi terhadap sistem operasi android yang dimiliki aplikasi kasir. Printer ini akan otomatis melakukan *printing* apabila transaksi baru dibuat ataupun ketika melakukan print untuk bon transaksi. Namun apabila

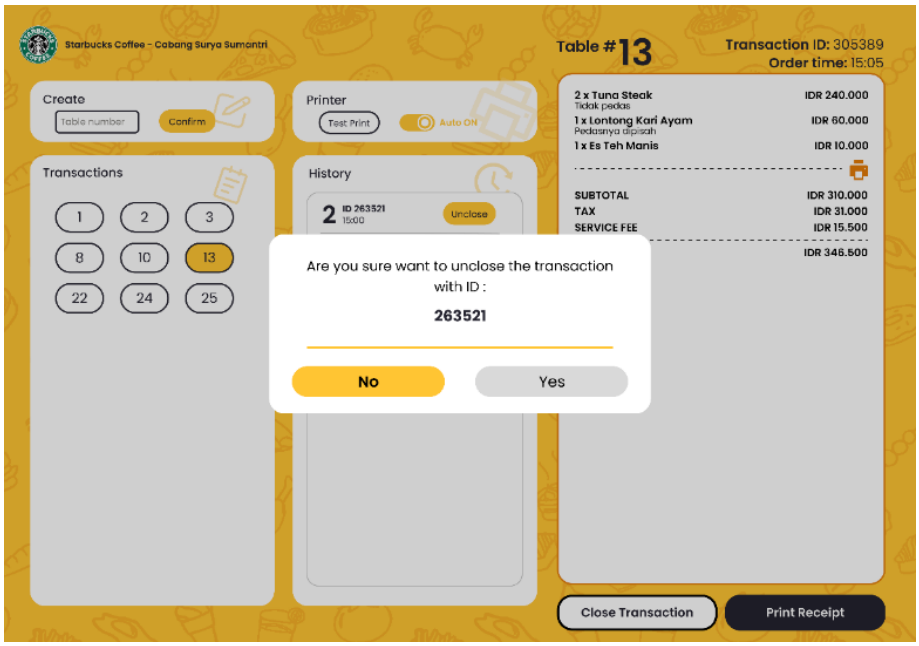
dirasakan tidak perlu, kasir dapat membuat tidak aktif *printer* tersebut dengan *menu printer*. Hal ini dapat dilakukan apabila terjadi masalah dengan printer ataupun kertas habis. Disini printer *bluetooth* perlu dipilih pada saat awal membuka aplikasi untuk memastikan perangkat printer *bluetooth* yang tersedia sudah berjalan dan dapat terhubung dengan sistem operasi android yang berjalan di sisi kasir.

Aplikasi kasir juga memungkinkan kasir untuk dapat melakukan *unclose* terhadap transaksi yang sudah ditutup oleh pelanggan. Hal ini dapat terjadi apabila ada kesalahan di pihak pelanggan ataupun ada kasus kasus lain yang mungkin terjadi.



Gambar 15. Tampilan Kasir

Saat kasir klik tombol *Unclose* akan muncul tampilan seperti pada Gambar 16.



Gambar 16. Tampilan Batal Tutup Pesanan

Selain itu juga, di dalam aplikasi kasir dapat melihat semua transaksi beserta detail transaksi yang tersedia.

IV. SIMPULAN

A. *Simpulan*

Dari hasil penelitian tersebut dapat diambil kesimpulan bahwa sistem pemesanan dan pembayaran makanan berbasis *cloud storage* dapat memberikan beberapa manfaat. Sistem pemesanan digital terintegrasi antara pelanggan dalam melakukan pemesanan makanan sampai dibuatkan pesanannya oleh bagian dapur dan pembuatan *invoice* oleh kasir. Sistem yang dibangun teruji di lapangan dapat menyesuaikan kebutuhan pelanggan, kasir dan bagian dapur secara keseluruhan. *Stream* membantu sinkronisasi data sehingga pelanggan dengan berbagai alat dapat melihat perubahan data yang sama pada waktu yang bersamaan. Berikut ini adalah saran untuk pengembangan penelitian ini lebih lanjut yaitu pengembangan aplikasi *mobile* yang dapat langsung melihat menu untuk setiap tempat makan, pengintegrasian aplikasi pemesanan digital dengan bagian dapur untuk persiapan pesanan, dan sistem pembayaran yang terintegrasi dengan pembayaran digital saat ini.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada LPPM UK Maranatha yang sudah memberikan dukungan dalam penelitian ini.

DAFTAR PUSTAKA

[1] A. C. Resti, "Sistem Pemesanan Otomatis Berbasis Mikrokontroler At89s52 Pada Restoran," , 2009. [Online]. Available: http://eprints.uns.ac.id/2067. [Accessed 31 12 2020].

[2] V. Chavan, P. Jadhav, S. Korade and P. Teli, "Implementing Customizable Online Food Ordering System Using Web Based Application," *IJISET - International Journal of Innovative Science, Engineering & Technology,*, vol. 2, no. 4, p. 722, 2015.

[3] K. Y. Lin, C.-H. Chen , Z.-M. Zhang and S.-C. Ou, "NFC-based mobile application design restaurant ordering system app," in *IEEE international Conference on applied System innovation* , Taiwan, 2018.

[4] A. Y. Putri and D. . Yendri, "Sistem Pemesanan Makanan dan Minuman Pada Restoran Menggunakan Teknologi NFC Berbasis Android," , 2018. [Online]. Available: http://jitce.fti.unand.ac.id/index.php/jitce/article/view/6. [Accessed 31 12 2020].

[5] A. Bankar and S. S. Suresh , "Intelligent Restaurant - Menu Ordering System," *IOSR journal of VLSI and Signal Processing*, vol. 5, no. 5, pp. 47-53, 2015.

[6] K. Kaushal, K. Yadav, V. Vaibhav, C. Sharma, L. Gupta, T. Tripathy and R. Goel, "The E-Restaurant," in *2016 Ninth International Conference on Contemporary Computing (IC3)*, Noida , 2016.

[7] K. Chochiang , P. Ung and N. Bungsaman, "One Stop Restaurant Service Application," in *International Conference on Electrical Engineering / Electronics, Telecommunication and Information (ECTI-CON)*, Phuket, 2020.

[8] V. Liyanage, A. Ekanayake, H. Premasiri and P. Munasinghe, "Foody - Smart Restaurant Management and Ordering System," in *2018 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Malambe, 2018.

[9] R. V. Ravi, A. N.R., A. E., H. P. and J. T., "An Android Based Restaurant Automation System with Touch Screen," in *2019 Third International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, 2019.

[10] A. Bhargave, N. Jadhav, A. Joshi, P. Oke and S. Lahane, "DigitalOrdering System for Restaurant UsingAndroid," *International Journal of Scientific and Research Publications*, vol. 3, no. 4, p. 1, 2013.

[11] N. S. H, Aplikasi Berbasis Android: Berbagai Implementasi dan Pengembangan Aplikasi Mobile Berbasis Android, Bandung: Informatika, 2015.

[12] T. Arifianto, Membuat Interface Aplikasi Android Lebih Keren dengan LWUIT, Yogyakarta: Andi Offset, 2011.

[13] Google, "Flutter dev," Google, [Online]. Available: https://flutter.dev/. [Accessed 1 1 2020].

[14] W. Leler, "hackernoon," 2017. [Online]. Available: https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514. [Accessed 1 1 2020].

[15] "Flutter Stream Basics for Beginners | by Dane Mackier | Flutter Community | Medium," Flutter Community, [Online]. Available: https://medium.com/flutter-community/flutter-stream-basics-for-beginners-eda23e44e32f. [Accessed 14 Desember 2021].

[16] "Firebase Products," Firebase, [Online]. Available: https://firebase.google.com/products-build. [Accessed 14 Desember 2021].

[17] "What is NoSQL? | Nonrelational Databases, Flexible Schema Data Models | AWS," Amazon Web Services, [Online]. Available: https://aws.amazon.com/nosql. [Accessed 14 Desember 2021].

[18] "NoSQL Database - What is NoSQL? | Microsoft Azure," Microsoft Azure, [Online]. Available: https://azure.microsoft.com/en-us/overview/nosql-database. [Accessed 14 Desember 2021].