

## Pengembangan Aplikasi Manajemen Antrean Pesanan Menu Restoran dengan Memanfaatkan Teknologi Kode QR

Yahya Sahaja<sup>1</sup>, Agi Putra Kharisma<sup>2</sup>, Tri Afirianto<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>yahyasahaja@student.ub.ac.id, <sup>2</sup>agi@ub.ac.id, <sup>3</sup>tri.afirianto@ub.ac.id

### Abstrak

Permasalahan yang terjadi di sebuah restoran dalam melayani pelanggan sangat beragam. Kebanyakan Restoran di Malang masih memberikan daftar menu secara manual kepada pelanggannya yang membuat pihak restoran harus mencetak baru menunya setiap kali terdapat perubahan daftar menu restoran. Selain itu, pemesanan menu restoran secara manual masih menggunakan kertas, sehingga pelanggan harus menuliskan menu yang ingin dipesan melalui kertas yang diberikan oleh pelayan restoran. Hal ini akan membuang-buang waktu dan tenaga para pelanggan. Dengan keterbatasan karyawan restoran dan jumlah pelanggan yang sangat ramai membuat pelayanan di restoran tersebut sangat lama. Berdasarkan permasalahan tersebut, penulis membuat aplikasi manajemen antrean pesanan menu restoran yang diharapkan dapat mengotomatiskan sistem antrean di restoran. Aplikasi yang dikembangkan berupa dua aplikasi, yaitu aplikasi untuk pelanggan dan aplikasi untuk operator restoran. Penelitian ini menggunakan metode pengembangan perangkat lunak *Agile* dengan menerapkan kerangka kerja Scrum dikarenakan kebutuhan pengguna yang memiliki banyak perubahan di setiap fase pengembangan. Hasil dari penelitian ini berfokus pada penyelesaian masalah dengan menyederhanakan proses manajemen antrean pesanan menu restoran dan menyederhanakan pemesanan menu restoran. Selain itu, peneliti juga melakukan dua pengujian, yaitu pengujian validasi dan *usability* untuk mengetahui tingkat kebergunaan aplikasi bagi pengguna. Dari pengujian validasi didapatkan hasil bahwa validasi yang dilakukan di setiap *sprint* terhadap kebutuhan fungsionalitas dinyatakan valid, sedangkan dari hasil pengujian *usability* didapatkan hasil bahwa aplikasi untuk pelanggan mendapatkan nilai *usability* sebesar 82,25% yang masuk ke dalam kategori B, dengan *adjective rating excellent*. Aplikasi untuk operator restoran mendapatkan nilai *usability* sebesar 96,25% yang masuk ke dalam kategori A, dengan *adjective rating best imaginable*.

**Kata kunci:** Pengembangan aplikasi, manajemen antrean, pesanan menu restoran, teknologi Kode QR, Scrum framework, *usability*.

### Abstract

*The problems happen in a restaurant for serving customers are too diverse. Many of restaurants in Malang are still just giving the menu list manually to the customers, that the restaurant should print the updated menu. Besides, ordering in a restaurant is still just using paper, that the customers should write the menu manually to order, and give it back to the restaurant waiters. The impact is that it can take too much time and costs more energy to the waiter as well as handling the restaurant menu ordering, even if there are so many customers should be served. Based on those problems, the writer wants to make an application that can handle restaurant food ordering queue management, hopefully this application could automate queue system in a restaurant. This application is built with following Agile methodology implementing Scrum framework, because the requirements are still developing within the application development. The result of this research was focused to solve the problems as well as simplify the queue management process and menu ordering in a restaurant. The researcher would also test the system with validation and usability test. The usability test was done for determining the usability level as the customers view point. For the result, validation testing states that the validations for every sprint is valid and usability testing states that the usability value for customers application got 82,25% score belonging in an B category with adjective rating excellent, whereas the application for restaurant operator got 96.25% belonging as an A category with adjective rating best imaginable.*

**Keywords:** *application development, queue management, restaurant menu ordering, QR Code technology, Scrum framework, usability testing.*

---

## 1. PENDAHULUAN

Restoran merupakan bangunan yang digunakan secara komersil yang melayani konsumen baik makanan dan/atau minuman (Marsum, 2005). Bagi orang yang sibuk dengan pekerjaan dan tidak sempat untuk memasak, rumah makan atau restoran dapat menjadi solusinya. Usaha restoran/rumah makan berskala menengah dan besar memiliki rata-rata tamu per harinya sebanyak 227 orang dan tempat duduk yang tersedia sebanyak 131 tempat duduk per usaha pada tahun 2015. Dilihat dari lokasi usaha, sebagian besar usaha restoran/rumah makan bertempat di kawasan pertokoan atau perkantoran, yaitu sebesar 54,57 persen. Sedangkan di lokasi objek wisata hanya sebesar 15,71 persen (BPS, 2017).

Menurut data yang didapatkan dari kuesioner yang dibagikan kepada 265 orang, 176 orang menjawab bahwa mengantri tetap menjadi permasalahan yang mereka alami pada saat berada di restoran dan 80 orang menjawab pelayanan yang ada kurang baik. Dari kuesioner tersebut dan wawancara yang telah dilakukan pada restoran juga didapatkan bahwa restoran di Malang masih memberikan daftar menu secara manual kepada pelanggannya yang membuat pihak restoran harus mencetak baru menunya setiap kali terdapat perubahan daftar menu restoran. Selain itu, pemesanan menu restoran secara manual masih menggunakan kertas, sehingga pelanggan harus menuliskan menu yang ingin dipesan melalui kertas yang diberikan oleh pelayan restoran. Hal ini akan membuang-buang waktu dan tenaga para pelanggan apalagi bagi mereka yang hanya memiliki waktu istirahat yang sedikit. Dengan keterbatasan karyawan restoran dan jumlah pelanggan yang sangat ramai membuat pelayanan di restoran tersebut sangat lama. Dengan data tersebut dapat diambil kesimpulan bahwa usaha restoran memang selalu ramai akan pelanggan

Berdasarkan permasalahan yang telah dipaparkan, penulis ingin membuat aplikasi manajemen antrian pesanan menu restoran yang diharapkan dapat mengotomatiskan sistem antrian di restoran dengan membuat sistem terpusat yang dibantu dengan teknologi yang berkembang pada era saat ini. Salah satu teknologi yang dapat menjadi solusi adalah

Progressive Website Application (PWA). PWA menggunakan kapabilitas modern website untuk membawa suatu website menjadi app-like yang membuat pengalaman pengguna lebih menyerupai aplikasi *smartphone*, sehingga akan membuat aplikasi mobile web menjadi lebih cepat, dapat diandalkan, dan engaging. (Karpagam, 2017) Karena pengalaman pengguna sangat penting dalam PWA, maka diperlukan pengujian untuk mengetahui tingkat pengalaman pengguna yang terfokus pada pengalaman penggunaan aplikasi *smartphone*, sehingga pengujian *usability* dapat menjadi opsi untuk menguji pengalaman pengguna dalam menggunakan sistem (Bangor, Staff, Kortum, Miller, & Staff, 2009).

Pengembangan aplikasi ini dalam hal autentikasi restoran akan lebih mudah dengan menggunakan Kode QR. Quick Response Code atau yang biasa disebut sebagai Kode QR adalah gambar digital dua dimensi dimana dapat dengan mudah dibaca oleh kamera pada perangkat mobile manapun. Sekarang ini Kode QR sangat populer karena perkembangan mobile. Penggunaan mobile device mencapai 15.6% pada tahun 2001 hingga 74.9% (2010) (Cata, T., Patel, P. S., & Sakaguchi, 2013). Teknologi akan terus berkembang dari waktu ke waktu. Dengan adanya teknologi, banyak permasalahan dapat diatasi dan dipermudah. Sehingga seharusnya pihak restoran dapat memanfaatkan teknologi tersebut untuk membuat sistem restorannya agar sistem pelayanan dalam restoran dapat terotomatiskan.

Dari permasalahan yang telah diuraikan sebelumnya, kemudian didapatkan empat rumusan masalah, yaitu bagaimana menyederhanakan proses manajemen antrian pesanan menu restoran dengan aplikasi manajemen antrian pesanan menu restoran?; Bagaimana menyederhanakan proses pemesanan menu restoran dengan aplikasi manajemen antrian pesanan menu restoran?; Bagaimana hasil pengujian validasi pada aplikasi manajemen antrian pesanan menu restoran dengan menggunakan pengujian *blackbox*?; Dan bagaimana hasil tingkat *usability* dalam penggunaan aplikasi manajemen antrian pesanan menu restoran dengan pengujian *usability*?

Pengembangan aplikasi dilakukan dengan

metode pengembangan *Agile* dan menggunakan *framework* Scrum. Hal ini dilakukan karena kebutuhan yang masih belum jelas di awal pengembangan, sehingga perlu adanya iterasi untuk mendefinisikan kebutuhan-kebutuhan sistem.

Setelah sistem selesai diimplementasikan, penulis melakukan pengujian untuk menguji kualitas dari sistem. Pengujian yang dilakukan sebanyak 3 pengujian, yaitu pengujian validasi, *regression testing*, dan *usability testing* untuk mengetahui tingkat kebergunaan aplikasi.

## 2. LANDASAN KEPUSTAKAAN

### 2.1 Penelitian Terkait

Dalam melakukan penelitian ini, terdapat beberapa penelitian sebelumnya yang relevan dengan manajemen antrean pesanan menu restoran, yaitu penelitian yang berjudul Aplikasi Reservasi Menu Restoran Berbasis Web dan Mobile Android di Cowek Ireng (Rasid, Supriyono, & Setiawan, 2018). Pada penelitian ini, terdapat masalah antrean panjang di restoran yang diatasi dengan membuat aplikasi pemesanan menu restoran pada restoran Cowek Ireng. Hasil dari penelitian ini dapat menunjang proses manajemen restoran menjadi lebih baik dengan pengelolaan pemesanan menu, reservasi menu dan tempat, dan pengelolaan data user seperti pimpinan, koki, kasir, koki, dan pelanggan.

Penelitian yang relevan lainnya adalah penelitian yang berjudul Online Food Ordering System (Singh & Kanade, 2018). Penelitian ini membahas industri makanan selalu menjadi lahan yang mengundang profit bukan hanya bagi pemilik restoran, namun juga kepada pengguna/customer dan distributor. Perubahan yang besar di industri makanan dan juga semakin meningkatnya teknologi dengan memanfaatkan internet dapat membuat pemesanan menu makanan secara online di restoran menjadi kebutuhan masyarakat. Pengguna dapat dengan mudah memesan menu makanan di restoran menggunakan real-time online food ordering, menelusuri proses pembuatan makanan, dan memberikan feedback kepada menu makanan dan juga restoran agar dapat meningkatkan produktivitas restoran. Inisial implementasi system dilakukan pada 2 restoran/mess pada 5 area.

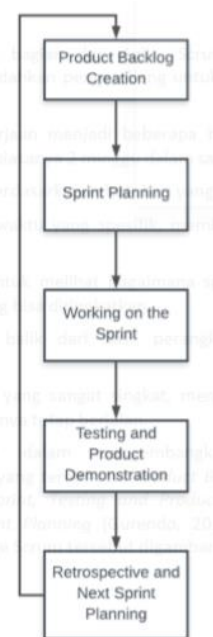
Penelitian tentang Customer-Oriented Restaurant Management System (CORMS) juga

membahas tentang manajemen antrean pesanan menu restoran. Penelitian ini menjadikan sebuah sistem multiplatform (terutama pada tablet) yang dapat mengatur pemesanan di sebuah restoran tidak hanya pada pemesanannya saja, namun juga mengatur dan mengubah menu-menu di restoran dengan user interface yang baik. Selain itu dengan menampilkan rekomendasi menu makanan pada restoran akan memberikan user experience yang baik kepada customer (Davis, Francis, Sukumaran, E, & Nair, 2017).

### 2.2 Progressive Website Application

*Progressive Website Application* atau biasa disebut PWA menggabungkan yang terbaik dari aplikasi website dan *mobile*. PWA dibangun menggunakan menggunakan teknologi *website application* yang bertindak selayaknya seperti sebuah aplikasi *native mobile*. Ide tentang PWA ini pertama kali disokong oleh *engineer* Google, Alex Russel, pada Juni 2015. Konsep dari PWA ini adalah teknologi, konsep desain, dan Web APIs yang membuat aplikasi website yang biasanya diakses melalui browser menyediakan pengalaman *app-like* seperti *push notification*, *work offline*, tampilan atau *app-shell* terlihat dan terasa seperti aplikasi *native mobile*, dan *load from home screen*

### 2.3 SCRUM



Gambar 1. Pengembangan dalam Metode *Scrum*

Pengembangan sistem manajemen antrean pesanan menu restoran menggunakan metodologi *Agila* dengan memanfaatkan

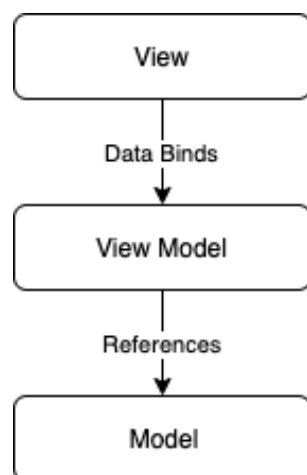
*framework* Scrum karena metode ini dapat digunakan dalam pengembangan sistem yang memiliki tingkat kompleksitas tinggi dan perubahan terus-menerus. Scrum adalah salah satu bagian dari *Agile*. Alur pengembangan dengan metode Scrum dapat dilihat pada Gambar 1.

Pada Gambar 1 menunjukkan bahwa alur pengembangan berawal dari pembuatan produk *backlog*, kemudian dilanjutkan dengan perencanaan *sprint*, pengerjaan *sprint* yang dilakukan selama durasi *sprint*, melakukan *testing* dan demonstrasi produk kepada *stakeholder*, dan terakhir melakukan retrospektif dan perencanaan *sprint* selanjutnya. Setelah itu jika alurnya akan mengulang dari awal kembali sesuai dengan kebutuhan sistem.

## 2.4 React JS

React JS merupakan suatu *UI library* yang dikembangkan oleh Facebook yang memiliki fasilitas *interactive*, *stateful*, dan *reusable UI components*. *Library* ini telah digunakan oleh Facebook. React JS menjadi *library* yang paling baik dalam proses *rendering user interface* yang sangat kompleks namun dengan performa yang tinggi. Dasar dari React adalah konsep yang menggunakan *Virtual DOM* dimana dapat digunakan untuk *rendering* pada *client-side* ataupun *server-side*. Cara kerjanya adalah dengan manipulasi DOM sesuai dengan perubahan *state* secara *up-to-date* (Kumar & Singh, 2016)

## 2.5 MVVM Design Pattern



Gambar 2. Pattern MVVM

Model View ViewModel (MVVM) adalah *design pattern* atau *architectural pattern* yang digunakan untuk memisahkan antara *User*

*Interface* dan *Application Logic*. *Design pattern* MVVM membentuk *linear* yang terdiri dari View-View Model-Model, seperti diagram pada Gambar 2. (Vanus et al., 2016)

Konsep dasar dari MVVM adalah View Model yang menangani komunikasi antara View dan Model. View akan selalu mengamati perubahan data yang dilakukan pada logic yang terdapat di View Model, sedangkan pengelolaan data pada View Model akan mereferensi data yang tersimpan pada Model. Sehingga View Model tidak akan memerdukan View yang melakukan *binding* ke View Model. (Kouraklis, 2016)

Penerapan *design pattern* yang memisahkan antara bisnis dengan *logic* dan meningkatkan *reusability* pada kode program dapat memudahkan pengembangan, pengujian, dan *maintain* aplikasi. Hal ini dikarenakan *developer* dan *page designer* dapat saling kooperatif, sehingga ketika terdapat perubahan UI, tidak perlu lagi mengubah *logic* yang sudah ada untuk memanipulasi data. (Li et al., 2015)

### 2.4.2. Kode QR

Denso Wave menciptakan *Kode QR* pada tahun 1994. Denso Wave menggunakan Kode QR sebagai trademark yang telah terregistrasi untuk melacak produknya. Perluasan penggunaan *Kode QR* kemudian dipromosikan oleh Denso Wave sebagai metode yang cepat dan mudah untuk *tracking* pada suatu produk.

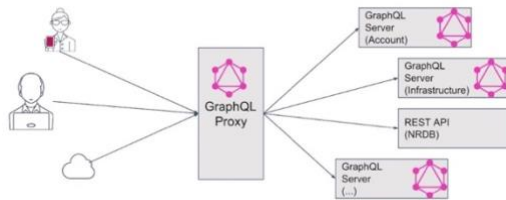
Kode QR telah digunakan dalam skala besar pada pemasaran atau kampanye sejak awal tahun 1990-an untuk menciptakan interaksi dengan konsumen. Denso Wave menggunakan ekstensi teknologi ini karena potensinya dalam perdagangan otomotif. Kode QR Pertama kali dating ke pasar sebagai iklan produk pada tahun 2011 ketika industri telekomunikasi sedang menaik. Hingga saat ini, Kode QR telah menjadi teknologi yang populer pada *smartphone* (Ozdenizci, Aydin, Coskun, & Ok, n.d.)

## 2.6 GraphQL

Graphql adalah sebuah konsep baru dalam membangun sebuah *application programming interface* (API) dan diimplementasikan pada sisi *server*. Posisi Graphql berada pada sisi klien dan *server* yang berhubungan dalam suatu API untuk mengambil dan memanipulasi data. Graphql didesain untuk berkolaborasi dengan bahasa



pemrograman *server* yang lain dengan baik (Buna, 2016).

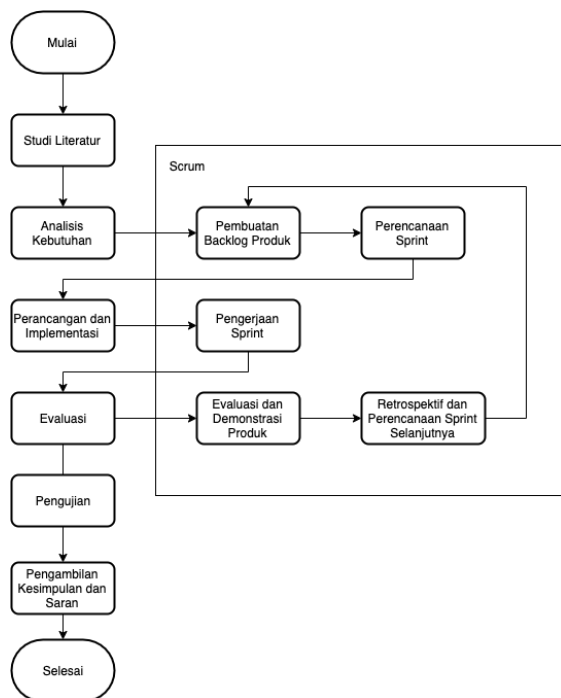


Gambar 3. *Graphql diagram*

Sumber: (Weitzel, n.d.)

Pada Gambar 3 menunjukkan bahwa client mengirimkan request ke GraphQL server melalui proxy GraphQL. Kemudian request tersebut diolah oleh GraphQL server.

### 3. METODOLOGI



Gambar 4. Alur Metodologi Penelitian

Metodologi penelitian dibutuhkan dalam mengembangkan perangkat lunak agar proses dapat terarah dengan baik dan sesuai dengan tujuan. Metode dalam pengembangan perangkat lunak sistem manajemen antrean pesanan menu restoran adalah dengan metode SDLC Scrum. Tahapan-tahapan yang dilakukan pada penelitian ini digambarkan dalam Gambar 4.

#### 3.1 Analisis Kebutuhan

Analisis kebutuhan merupakan tahap dalam

mempelajari kebutuhan calon pelanggan agar mendapatkan definisi kebutuhan sistem atau tahap untuk menentukan kebutuhan dari seluruh elemen sistem. Tahap ini merupakan masa analisis kebutuhan sistem. Semua kebutuhan sistem akan dideskripsikan secara lengkap dalam fase ini.

##### 3.1.1 Pembuatan *Backlog* dan Perancangan *Sprint*

Pada fase analisis kebutuhan, hal yang harus dilakukan pertama kali adalah mendeskripsikan aplikasi secara umum, sehingga pembuatan aplikasi yang akan dibuat dapat lebih mudah untuk dipahami. Selanjutnya adalah penentuan pengguna sistem yang kemudian akan menjadi aktor di dalam sistem. Setelah itu, peneliti mendapatkan *user story* dari masing-masing aktor yang sudah ditentukan. *User story* yang telah didapatkan dapat digunakan untuk menentukan daftar fungsionalitas dari sistem. Daftar fungsionalitas ini akan menjadi rujukan utama dalam mengimplementasikan sistem.

Setelah mendapatkan fungsionalitas dari sistem, kemudian peneliti membagi *backlog* atau fungsionalitas menjadi beberapa *sprint*. Pembagian *backlog* produk ke beberapa *sprint* ini dilakukan dengan mengurutkan *backlog* sesuai dengan prioritas yang dibutuhkan sistem untuk dapat bekerja terlebih dahulu. Setelah itu peneliti menentukan durasi *sprint* yang akan digunakan oleh tim berdasarkan tingkat kompleksitas dari fungsionalitas yang telah dibuat. Selanjutnya dilakukan pembagian *backlog* produk berdasarkan durasi *sprint*.

##### 3.2 Perancangan dan Implementasi

Tahapan ini adalah tahap dalam menentukan perangkat keras (*hardware*), arsitekturnya, dan sistem persyaratan yang dibutuhkan. Dalam tahapan ini, dilakukan segala persiapan akan sesuatu yang dibutuhkan dalam melakukan implementasi pada aplikasi manajemen antrean pesanan menu restoran. Tahapan Perancangan Sistem diantaranya adalah:

##### 1. Perancangan Arsitektur

Perancangan arsitektur dilakukan dengan pemodelan sistem menggunakan diagram *Unified Modelling Language*. Diagram UML yang digunakan adalah sequence diagram dan class diagram.

##### 2. Perancangan Komponen

Perancangan komponen adalah melakukan perancangan berupa sampel algoritme utama dalam bentuk *pseudocode* yang diambil pada class *controller*, sehingga akan mendefinisikan *logic* dari suatu *program*.

### 3. Perancangan Antarmuka

Proses perancangan antarmuka dari sistem terdiri dari *layout* atau tata letak komponen berdasarkan kebutuhan dari sistem. Rancangan antarmuka ini akan digunakan dalam implementasi antarmuka yang dihasilkan setelah *program* diimplementasikan.

Pada tahap implementasi, peneliti menuliskan serangkaian kode program untuk mewujudkan perancangan yang telah dirancang sebelumnya. Kode program tersebut menggunakan bahasa pemrograman Javascript dengan menggunakan React JS sebagai framework untuk aplikasi pengguna, sedangkan pengolahan data dilakukan menggunakan Node JS sebagai Javascript runtime untuk memproses data dan GraphQL sebagai *resolver* dan *query language*. Tahap implementasi sistem diantaranya adalah menjelaskan spesifikasi sistem, dilanjutkan dengan mendefinisikan batasan implementasi, dan kemudian implementasi kode program dan antarmuka.

### 3.3 Evaluasi

Evaluasi digunakan untuk menentukan telah terselesaikannya semua *backlog* yang telah dibuat pada setiap *sprint*. Tim harus menyelesaikan seluruh *backlog* yang sudah direncanakan untuk dikerjakan selama satu *sprint*, jika ada produk *backlog* yang belum selesai, maka *sprint* selanjutnya dapat menangani produk *backlog* tersebut.

#### 3.3.1 Evaluasi dan Demonstrasi Produk

Merupakan tahap evaluasi sistem yang akan dilakukan terhadap sistem yang telah dikembangkan, pengujian yang dilakukan adalah dengan menggunakan metode pengujian Black Box Validation dan Regression Test. Evaluasi sistem bertujuan untuk mengetahui jalannya kebutuhan fungsional yang didefinisikan sebelumnya apakah berjalan sesuai yang diinginkan. Tahapan evaluasi sistem diantaranya adalah pengujian Black Box (Validasi) dan Regression Test.

#### 3.3.2 Retrospektif dan Perancangan *Sprint*

### Selanjutnya

Setelah *sprint* pada suatu iterasi telah selesai dilakukan sesuai tahapan maka retrospektif merupakan tahap berikutnya, yang bertujuan untuk mengevaluasi secara utuh proses dalam suatu *sprint*. Evaluasi tersebut akan menghasilkan kesimpulan dari *sprint*. Kemudian hasil retrospektif akan menjadi bahan rujukan untuk merancang kembali *sprint* yang akan dilakukan pada fase iterasi berikutnya.

### 3.3 Pengujian

Setelah semua proses pengembangan perangkat lunak dalam seluruh *sprint* telah dilakukan, maka sistem harus diuji kepada calon pengguna yang akan menggunakan sistem ini. Pengujian dalam penelitian ini menggunakan Usability Testing yang akan mengevaluasi tingkat kebergunaan aplikasi yang berdampak pada tingkat kepuasan pengguna terhadap sistem yang telah dikembangkan, sehingga pengguna dapat dengan nyaman dan puas berinteraksi terhadap sistem.

### 3.3 Pengambilan Kesimpulan dan Saran

Hasil pengujian dan analisis sistem akan menjadi bahan untuk pengambilan kesimpulan. Kesimpulan diambil untuk mengetahui keberhasilan dan menjawab permasalahan yang telah dirumuskan dalam penelitian. Setelah pengambilan kesimpulan dilanjutkan dengan memberikan saran berdasarkan hasil evaluasi dari penelitian ini.

## 4. PENGEMBANGAN METODOLOGI KERANGKA KERJA SCRUM

Penelitian ini dilakukan dengan mengimplementasi metodologi Agile menggunakan kerangka kerja Scrum. Tahap pertama dalam implementasi Scrum adalah analisis kebutuhan. Analisis kebutuhan didapatkan dengan membuat user story berdasarkan wawancara yang dilakukan kepada narasumber. Setelah menganalisis kebutuhan sistem, kemudian didapatkan 26 kebutuhan fungsional dan 1 kebutuhan non-fungsional. Adapun hasil use case diagram untuk menggambarkan interaksi sistem dan lingkungannya dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Use case diagram aplikasi manajemen antrian pesanan menu restoran untuk pelanggan



Gambar 6. Use case diagram aplikasi manajemen antrian pesanan menu restoran untuk restoran.

Pada Gambar 5 terdapat dua aktor, yaitu Tamu Pelanggan dan Pelanggan. Aktor Tamu Pelanggan berhubungan dengan *use case* Login, sedangkan aktor Pelanggan berhubungan dengan seluruh sisa *use case* yang terdapat dalam sistem tersebut.

Pada Gambar 6 terdapat dua aktor, yaitu Tamu Operator Restoran dan Operator Restoran.

Aktor tamu operator restoran berhubungan dengan *use case* Login Operator Restoran, sedangkan aktor Operator Restoran berhubungan dengan seluruh sisa *use case* yang terdapat di dalam sistem tersebut.

Kebutuhan tersebut kemudian diidentifikasi menjadi *backlog* yang kemudian dipecah untuk diimplementasikan menjadi tiga *sprint*. Durasi *sprint* yang dilakukan adalah selama dua minggu.

Pada setiap *sprint* dilakukan perancangan antarmuka dan kode program; implementasi antarmuka dan kode program; dan evaluasi menggunakan *blackbox testing* dan *regression testing* untuk masing-masing *backlog* pada *sprint* tersebut. Kemudian setelah satu *sprint* selesai, dilakukan demonstrasi produk kepada pelanggan untuk melakukan retrospektif dan perancangan *sprint* selanjutnya.

Adapun satu kebutuhan fungsional utama dari penelitian ini adalah mengubah status pesanan yang dilakukan oleh operator restoran. Implementasi metodologi menggunakan *framework* Scrum pada kebutuhan tersebut dilaksanakan pada *sprint* kedua. *Use Case Scenario* untuk kebutuhan fungsional tersebut dapat dilihat pada Tabel 1, hasil implementasi antarmuka dapat dilihat pada Tabel 2, dan hasil implementasi kode programnya dapat dilihat pada Gambar 7.

Pada Tabel 1 merupakan *use case scenario* Mengubah Pesanan yang dilakukan oleh aktor operator restoran. Tujuan dari *scenario* tersebut adalah untuk mengubah pesanan pelanggan. Setelah alur *scenario* dijalankan, maka kondisi akhir yang diharapkan adalah operator restoran telah mengubah pesanan pelanggan.

Tabel 1. Skenario Use Case Mengubah Status Pesanan

Use Case	Mengubah Pesanan
Kode terkait kebutuhan	SMA-F-003
Actor	Operator Restoran
Target	Operator restoran dapat mengubah pesanan pelanggan
Pre-Condition	Operator restoran telah berada di halaman daftar pesanan dan daftar pesanan telah ditampilkan oleh sistem

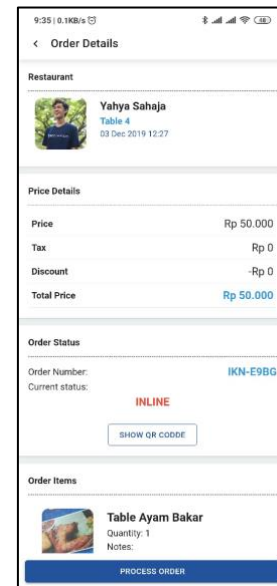
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Operator restoran menekan tombol “Process Order”</li> <li>2. Sistem mengubah status pesanan dari “INLINE” menjadi “PROCESS”</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Operator restoran menekan tombol “Complete Order”</li> <li>2. Sistem mengubah status pesanan dari “PROCESS” menjadi “COMPLETED”</li> </ol>
<i>Post Condition</i>	Operator restoran telah mengubah pesanan pelanggan

Pada Tabel 2 merupakan perancangan algoritme *process order*. Algoritme tersebut bertujuan untuk mengubah status pesanan menjadi berstatus “PROCESS”. Langkah yang dilakukan adalah melakukan *request set* ke Firebase dengan referensi status pesanan pada “id” sesuai *input*.

Tabel 2. Algoritme *Process Order*

1	Begin
2	Deklarasi id untuk inlineOrder yang ingin diubah
3	statusnya
4	Request <i>set</i> firebase dengan referensi
5	‘/Orders/{id}/status’ menjadi ‘PROCESS’
6	Request <i>set</i> firebase firestore dengan referensi
7	‘/Orders/{id}/updated_at’ menjadi Date saat ini
8	Catch <i>error</i> request
9	Begin
10	Show <i>error</i> message
11	End
12	end
13	

Pada Gambar 7 merupakan implementasi antarmuka Order Detail pada kebutuhan fungsional mengubah status pesanan. Pada hasil implementasi antarmuka Order Detail terdapat tombol “Process Order” untuk mengubah status pesanan pengguna.



Gambar 7. Hasil Implementasi Antarmuka Order Detail pada kebutuhan fungsional mengubah status pesanan.

## 5. PENGUJIAN *USABILITY*

Setelah sistem selesai diimplementasi, kemudian dilakukan pengujian *usability*. Pengujian *usability* dilakukan untuk menguji kebergunaan aplikasi bagi pengguna. Pengujian dilakukan dengan cara memberikan *task scenario* kepada pengguna, kemudian pengguna akan melakukan aktivitas untuk menjalankan *task scenario* yang diberikan. *Task scenario* yang diberikan kepada pelanggan merupakan langkah-langkah dalam menjalankan aplikasi. Langkah-langkah tersebut mengacu pada fungsionalitas aplikasi.

Pada pengujian ini, *task scenario* diberikan kepada dua jenis pengguna, yaitu pelanggan restoran dan operator restoran. Kemudian responden diminta untuk mengisi kuesioner sebagai penilaian tentang usability pada aplikasi ini. Pengujian pada penelitian ini menggunakan kuesioner *SUPER-Qm* yang terdiri dari 16 pertanyaan untuk masing-masing jenis pengguna. Setiap pertanyaan yang diberikan memiliki skor dengan menggunakan skala likert dalam rentang 1 sampai 5 yang dijelaskan pada Tabel 3.

Tabel 3. Skor Skala Likert dari Setiap Pertanyaan

Skor	Keterangan
1	Sangat Tidak Setuju (STS)
2	Tidak Setuju (TS)
3	Netral (N)



4	Setuju (S)
5	Sangat Setuju (SS)

Perhitungan SUPR-Q yang dilakukan untuk mendapatkan hasil kuantitatif dari skor pada pertanyaan kuesioner SUPR-Qm dilakukan dengan menggunakan rumus pada Persamaan 1.

$$\text{Nilai SUPR-Q} = \frac{\text{jumlah nilai diperoleh}}{\text{jumlah nilai maksimal}} \times 100\% \quad (1)$$

Persamaan SUPR-Q pada Persamaan 6.1 digunakan untuk perhitungan nilai SUPR-Qm. Kemudian nilai hasil perhitungan tersebut akan dikonversikan ke dalam kategori penilaian *usability*.

## 6. KESIMPULAN

Berdasarkan hasil dari analisis kebutuhan, perancangan, implementasi, dan pengujian pada penelitian pengembangan aplikasi manajemen antrean pesanan menu restoran didapatkan tiga kesimpulan.

Manajemen antrean pesanan menu restoran yang sebelumnya menggunakan kertas untuk mengatur antrean pesanan menu pelanggan di restoran dapat disederhanakan dengan menggunakan aplikasi manajemen antrean pesanan menu restoran sebagai aplikasi yang mengatur antrean pesanan menu restoran melalui *smartphone*.

Pemesanan menu restoran yang sebelumnya menggunakan kertas untuk mencatat menu yang ingin dipesan dan harus memanggil pelayan untuk melakukan pemesanan dapat disederhanakan dengan menggunakan aplikasi manajemen antrean pesanan menu restoran sebagai aplikasi untuk melakukan pemesanan menu restoran melalui *smartphone*.

Berdasarkan pengujian validasi yang telah dilakukan terhadap implementasi sistem yang dibuat, didapatkan hasil bahwa validasi yang dilakukan di setiap *sprint* terhadap kebutuhan fungsionalitas dinyatakan valid.

Berdasarkan pengujian *usability* yang telah dilakukan kepada pengguna secara langsung, hasil pengujian *usability* untuk aplikasi manajemen antrean pesanan menu restoran untuk pelanggan dapat dilihat pada Tabel 4.

Tabel 4. Hasil pengujian Kuesioner SUPR-Qm untuk pelanggan restoran.

No Pertanyaan	Skor					Total Skor
	1	2	3	4	5	
1	0	0	0	5	0	20
2	0	0	2	3	0	18
3	0	0	1	3	1	20
4	0	0	1	2	2	21
5	0	0	0	4	1	21
6	0	0	0	3	2	22
7	0	1	0	3	1	19
8	0	0	2	1	2	20
9	0	0	2	0	3	21
10	0	0	1	2	2	21
11	0	0	1	3	1	20
12	0	0	0	1	4	24
13	0	0	0	4	1	21
14	0	0	1	3	1	20
15	0	0	2	2	1	19
16	0	0	0	3	2	22
Total Akhir						329
Total Nilai Maksimum						400
Nilai SUPR-Qm						82,25%

Pengujian tersebut mendapatkan nilai *usability* sebesar 82,25% yang masuk ke dalam kategori B, dengan *adjective rating excellent*.

Hasil pengujian *usability* aplikasi manajemen antrean pesanan menu restoran untuk operator restoran mendapatkan dapat dilihat pada Tabel 5.

Pengujian tersebut mendapatkan nilai *usability* sebesar 96,25% yang masuk ke dalam kategori A, dengan *adjective rating best imaginable*. Sehingga dapat disimpulkan bahwa aplikasi manajemen antrean pesanan menu restoran untuk pelanggan dan restoran sangat berguna bagi pelanggan dan operator restoran.

Tabel 5. Hasil pengujian Kuesioner SUPR-Qm untuk pelanggan restoran

No Pertanyaan	Skor					Total Skor
	1	2	3	4	5	
1	0	0	1	0	0	3
2	0	0	0	0	1	5

3	0	0	0	0	1	5
4	0	0	1	0	0	3
5	0	0	0	1	0	4
6	0	0	0	0	1	5
7	0	0	0	0	1	5
8	0	0	0	0	1	5
9	0	0	0	0	1	5
10	0	0	0	0	1	5
11	0	0	0	0	1	5
12	0	0	0	0	1	5
13	0	0	0	0	1	5
14	0	0	0	0	1	5
15	0	0	0	0	1	5
16	0	0	0	0	1	5
Total Akhir						77
Total Nilai Maksimum						80
Nilai SUPR-Qm						96,25%

## 9. DAFTAR PUSTAKA

- Bangor, A., Staff, T., Kortum, P., Miller, J., & Staff, T. (2009). Determining what individual SUS scores mean: adding an adjective rating scale. *Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale*, 4(3), 114–123.
- BPS. (2017). *Statistik Restoran/Rumah Makan 2015*.
- Buna, S. (2016). *Learning GraphQL and Relay*. Birmingham: Packt Publishing Ltd.
- Cata, T., Patel, P. S., & Sakaguchi, T. (2013). *Article ID 813339, 7 pages Cancer. 2013*. <https://doi.org/10.5171/2013>
- Davis, R., Francis, N., Sukumaran, S. K., E, S. J., & Nair, U. (2017). CORMS : an Automated Restaurant Management System. *International Research Journal of Engineering and Technology (IRJET)*, 4(3), 2860–2867. Retrieved from <https://irjet.net/archives/V4/i3/IRJET-V4I3720.pdf>
- Karpagam, V. (2017). Performance Enhancement of Webpage Using Progressive Web App Features. *International Journal of Innovative Research in Advanced Engineering*, 03(4), 2349–2163.
- Kouraklis, J. (2016). MVVM in delphi: Architecting and building model view viewmodel applications. *MVVM in Delphi: Architecting and Building Model View ViewModel Applications*, (October 2016), 1–143. <https://doi.org/10.1007/978-1-4842-2214-0>
- Kumar, A., & Singh, R. K. (2016). Comparative analysis of angularjs and reactjs. *International Journal of Latest Trends in Engineering and Technology*, 7(4), 225–227. <https://doi.org/10.21172/1.74.030>
- Li, X., Chang, D., Pen, H., Zhang, X., Liu, Y., & Yao, Y. (2015). Application of MVVM design pattern in MES. *2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015*, (2012), 1374–1378. <https://doi.org/10.1109/CYBER.2015.7288144>
- Ozdenizci, B., Aydin, M. N., Coskun, V., & Ok, K. (n.d.). *Department of Information Technologies , ISIK University , Istanbul , Turkey { busraozdenizci , mnaydin , vedatcoskun , keremok }@ isikun . edu . tr*.
- Singh, A., & Kanade, V. (2018). *Online food ordering system*. 374–378.
- Vanus, J., Belesova, J., Martinek, R., Bilik, P., Zidek, J., & Koval, L. (2016). Development of Software Tool for Operational and Technical Functions Control in the Smart Home with KNX technology. *IFAC-PapersOnLine*, 49(25), 431–436. <https://doi.org/10.1016/j.ifacol.2016.12.088>
- Weitzel, M. (n.d.). Using the New Relic GraphQL API Explorer. Tersedia di: <<https://blog.newrelic.com/engineering/graphql-api/>> [Diakses 5 Desember 2019]