

**PENGEMBANGAN APLIKASI MANAJEMEN ANTRIAN  
RESTORAN DENGAN MEMANFAATKAN TEKNOLOGI NEAR-  
FIELD COMMUNICATION (NFC) DAN KODE QR**

**PROPOSAL SKRIPSI**

Disusun oleh:  
Yahya Sahaja  
NIM: 165150207111086



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019

## DAFTAR ISI

DAFTAR ISI .....	ii
DAFTAR TABEL.....	iv
DAFTAR GAMBAR .....	v
DAFTAR LAMPIRAN .....	vi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan Masalah .....	2
1.6 Sistematika Pembahasan .....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Kajian Pustaka .....	5
2.2 Manajemen Antrian Restoran .....	5
2.3 Aplikasi Perangkat Bergerak .....	6
2.4 Progressive Website Application .....	6
2.5 React JS .....	7
2.6 NFC.....	8
2.7 QR Code .....	8
2.8 GraphQL.....	8
2.9 Node JS.....	8
2.10 Model Pengembangan Perangkat Lunak.....	9
2.10.1 Scrum .....	9
2.11 MVVM <i>Design Pattern</i> .....	11
2.12 Pengujian Perangkat Lunak .....	11
2.12.1 <i>Blackbox Testing</i> .....	11
2.12.2 <i>Usability Testing</i> .....	12
2.12.3 <i>Regression Testing</i> .....	12
BAB 3 METODOLOGI .....	13
3.1 Diagram Alir Metode .....	13

3.2 Analisis Kebutuhan .....	13
3.2.1 Pembuatan <i>Backlog</i> Produk dan Perancangan <i>Sprint</i> .....	14
3.3 Perancangan dan Implementasi .....	14
3.4 Evaluasi .....	15
3.4.1 Evaluasi dan Demonstrasi Produk.....	15
3.4.2 Retrospektif dan Perancangan <i>Sprint</i> Selanjutnya .....	15
3.5 Pengujian .....	15
3.6 Pengambilan Kesimpulan dan Saran .....	16
DAFTAR REFERENSI .....	17

## DAFTAR TABEL

No table of figures entries found.

## DAFTAR GAMBAR

Gambar 2.1 Graphql diagram.....	8
Gambar 2.2 Node JS flow .....	9
Gambar 2.3 Alur Pengembangan dalam Metode Scrum .....	10
Gambar 2.4 Pattern MVVM .....	11
Gambar 3.1 Alur Metodologi Penelitian .....	13

## **DAFTAR LAMPIRAN**

**No table of contents entries found.**

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

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 menu secara manual yang membuat pihak restoran harus mencetak baru menunya. Ini juga yang menyebabkan pihak restoran harus mencetak ulang dan pelanggan juga harus menulis manual menu yang dipesan. Sebagian besar restoran juga masih memberikan menu secara manual yang membuat antrian yang sangat panjang. 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 “Sans” yang diharapkan dapat mengotomasi sistem pelayanan 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)

Pengembangan aplikasi ini dalam hal autentikasi restoran akan lebih mudah dengan menggunakan QR Code. *Quick Response Code* atau yang biasa disebut sebagai QR Code adalah gambar digital dua dimensi dimana dapat dengan mudah dibaca oleh kamera pada perangkat *mobile* manapun. Sekarang ini QR Code 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). Selain QR Code, sistem autentikasi restoran juga dapat ditangani dengan NFC. NFC adalah sistem transmisi data yang menggunakan prinsip

teknologi RFID. NFC merupakan *short-range* dan teknologi *high frequency* (13.56 MHz) yang membuatnya data mengirimkan data antar *devices* (Fernández, Fernández, Aguilar, Selvi, & Crespo, 2013). Dengan menggunakan NFC, autentikasi restoran cukup hanya dengan meletakkan perangkat *smartphone* pada kartu NFC.

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 terotomasi.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah penulis jabarkan sebelumnya, maka dapat dirumuskan rumusan masalahnya sebagai berikut:

1. Apakah kebutuhan fungsional dan non-fungsional aplikasi Sans?
2. Bagaimana hasil perancangan aplikasi Sans?
3. Bagaimana hasil implementasi aplikasi Sans?
4. Bagaimana hasil pengujian dalam aplikasi Sans?

## **1.3 Tujuan**

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengetahui apa saja kebutuhan fungsional dan non-fungsional aplikasi Sans.
2. Mengetahui bagaimana hasil perancangan aplikasi Sans.
3. Mengetahui bagaimana hasil implementasi aplikasi Sans.
4. Mengetahui bagaimana hasil pengujian dalam aplikasi Sans.

## **1.4 Manfaat**

1. Manfaat bagi penulis

Dapat mengembangkan sistem aplikasi perangkat bergerak sesuai dengan materi perkuliahan yang telah dipelajari sebelumnya.

2. Manfaat bagi peneliti selanjutnya

Dapat menjadi acuan untuk pengembangan aplikasi yang serupa.

3. Manfaat bagi pengguna

Dapat membantu untuk melakukan pemesanan makanan di restoran tanpa harus mengantre di depan kasir.

## **1.5 Batasan Masalah**

Pengembangan dalam penelitian ini memiliki beberapa batasan, diantaranya adalah:

1. Sistem yang akan dibuat menggunakan menggunakan *framework React JS*.



2. Sistem yang akan dibuat menggunakan konsep *Progressive Website Application*.
3. Sistem yang akan dibuat hanya dapat dijalankan pada *operating system* dengan *version* minimal pada *platform* Android Jelly Bean versi 4.0 dan *platform* iOS versi 9.0.
4. Sistem yang akan dibuat harus terkoneksi dengan internet
5. Data yang diolah bersumber dari restoran 7Seven Chicken di Malang
6. Target pengguna sistem adalah mahasiswa yang sering menggunakan 7Seven Chicken sebagai tempat melakukan kegiatan seperti mengerjakan tugas maupun rapat

## 1.6 Sistematika Pembahasan

Dokumen pada skripsi ini disusun menjadi beberapa bab yang terdiri dari:

1. Bab 1 – PENDAHULUAN  
Berisi tentang latar belakang, rumusan masalah, tujuan dari penelitian, manfaat dari penelitian, batasan penelitian, dan sistematika pembahasan.
2. Bab 2 – LANDASAN KEPUSTAKAAN  
Memuat kajian-kajian kepastakaan yang relevan yang akan digunakan sebagai referensi dalam melakukan penelitian ini.
3. Bab 3 – METODOLOGI PENELITIAN  
Memuat alur kerja penelitian sebagai proses penyelesaian masalah dalam penelitian ini.
4. Bab 4 – ANALISIS KEBUTUHAN  
Memuat hal-hal yang terkait seputar proses penggalian kebutuhan dalam proses pengembangan sistem.
5. Bab 5 – PERANCANGAN DAN IMPLEMENTASI  
Memuat hal-hal yang berkaitan dengan perancangan dan pemodelan sistem berdasarkan data yang telah didapat di tahap analisis kebutuhan serta memuat hal-hal yang berkaitan dengan implementasi pengembangan sistem berdasarkan pemodelan yang telah dilakukan sebelumnya.
6. Bab 6 – PENGUJIAN  
Memuat hal-hal yang berkaitan dengan pengujian sistem yang dilakukan oleh responden dan menganalisis hasil yang telah didapat. Pengujian terdiri dari pengujian *Black Box* dan analisis hasil pengujian.
7. Bab 7 – PENUTUP

Pada bab ini berisi kesimpulan dari keseluruhan uraian bab-bab sebelumnya. Serta saran-saran dari hasil yang diperoleh dan yang diharapkan dapat bermanfaat dalam pembelajaran selanjutnya.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Dalam melakukan penelitian ini, terdapat beberapa penelitian sebelumnya yang relevan dengan manajemen antrian 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 antrian 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. Menggunakan *real-time online food ordering*, pengguna dapat dengan mudah memesan menu makanan di restoran, menelusuri proses pembuatan makanan, dan memberikan feedback kepada menu makanan dan juga restoran agar dapat meningkatkan produktivitas restoran. Inisial implementasi system dilakukan pada 2 *restaurant/mess* pada 5 area.

Penelitian tentang *Customer-Oriented Restaurant Management System* (CORMS) juga membahas tentang manajemen antrian restoran. Penelitian ini menjadikan sebuah system 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 Manajemen Antrian Restoran

Tingginya minat pembeli pada sebuah restoran dapat mengakibatkan panjangnya antrian, khususnya pada hari libur. Antrian pada sebuah restoran seringkali membuat pelanggan tidak puas dan bahkan akhirnya memilih untuk meninggalkan restoran. Untuk mengatasi hal tersebut, perlu dilakukan manajemen antrian restoran. Manajemen antrian pemesanan menu di sebuah restoran dapat menggunakan satu kasir, yang membuat pelanggannya masuk pada *waiting line*, lalu pelanggan menunggu dan mendapatkan pelayanan. Untuk mengatasi waktu tunggu yang terlalu lama, restoran dapat memberikan fasilitas

tambahan untuk pelayanan kasir restoran, sehingga membuat sistem antrian menjadi *Multichannel-Single Phase* yang mana terdapat dua jalur antrian dengan dua fasilitas pelayanan kasir. (Susila, Panji, & Prima, 2007)

Manajemen antrian restoran dengan berfokus kepada *Customer* dapat dengan menggantikan sistem yang sudah ada yaitu penggunaan kertas untuk pemesanan dengan sistem baru yang menggunakan teknologi sebagai pengganti penggunaan kertas dalam pemesanan menu di restoran. Dorongan untuk sistem *online food ordering* semakin meningkat dari hari ke hari, hal ini dikarenakan dapat memudahkan akses dan fleksibilitas dari pekerjaan sehari-hari di sebuah restoran. Dengan teknologi manajemen antrian restoran, pihak restoran dapat dengan mudah mengganti menu restorannya dan mengatur pemesanan yang ada di restoran tersebut. (Davis et al., 2017)

### 2.3 Aplikasi Perangkat Bergerak

Aplikasi perangkat bergerak (*Mobile application*) adalah suatu set program atau *software* yang berjalan atau beroperasi pada sebuah perangkat bergerak atau *mobile device*. Aplikasi tersebut akan melakukan tindakan atau tugas-tugas khusus yang diberikan kepada penggunanya. *Mobile Application* adalah suatu segmen yang baru dan cepat dalam Teknologi Informasi dan Komunikasi. Aplikasi perangkat bergerak memiliki kapasitas yang ringan, ramah bagi pengguna, dapat diunduh, dan dijalankan pada *mobile phone*.

Beberapa aplikasi *mobile* merupakan aplikasi *pre-installed* yang sudah terinstall dalam sebuah smartphone, dan aplikasi perangkat bergerak lainnya dapat diunduh oleh pengguna melalui internet pada sebuah *market* yang disediakan oleh *developer* dan kemudian di-*install* pada perangkat. Dengan banyaknya aplikasi perangkat bergerak yang dikembangkan oleh *developer*, maka *market* dari aplikasi perangkat bergerak pun semakin meningkat. Sehingga beberapa aplikasi dapat di-*install* pada *platform* yang berbeda-beda seperti iPhone, Blackberry, Android, Symbian, dan Windows. Walaupun aplikasi *mobile* memiliki beberapa batasan seperti layar yang kecil, akses navigasi, dan kecepatan yang lambat; penggunaan aplikasi perangkat bergerak terus meningkat dari hari ke hari, banyak masyarakat lebih memilih untuk menggunakan aplikasi perangkat bergerak dibandingkan dengan aplikasi *desktop* untuk melakukan hal yang ringan (Islam & Mazumder, 2010).

### 2.4 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*.

*Progressive Website Application* memiliki beberapa fitur, seperti:

1. Dapat diandalkan

Memuat aplikasi secara instan, karena dapat berjalan secara *offline*, sehingga ketika aplikasi dibuka, maka *service worker* yang bekerja selayaknya *client proxy* akan handle seluruh request sesuai yang sudah termuat di dalam *cache* saat aplikasi pertama kali dimuat atau diperbarui.

2. Cepat

Merespon interaksi pengguna dengan cepat dengan animasi yang halus serta tanpa *Janky Scrolling*. Hal ini dikarenakan waktu untuk memuat membutuhkan lebih dari 3 detik, 53% pengguna akan meninggalkan situs tersebut.

3. Mengikat pengguna dan dapat diinstall

Banyak cara untuk mengikat pengguna, seperti *user experience* yang baik dengan menerapkan *design* seperti aplikasi *native mobile*, fitur instalasi aplikasi ke *home screen* tanpa perlu instalasi melalui *store*, dan dengan adanya *push notification* maka dapat mengikat kembali penggunaannya. *Developer* dapat menggunakan *app manifest* untuk mengatur *icon* pada *home screen*, mengontrol bagaimana tampilan ketika aplikasi akan dibuka, *screen orientation*, dan pengaturan *window*.

4. *Fresh*

Selalu *up-to-date* dengan fitur yang terbaru, hal ini dapat diatur menggunakan *service worker*. Selain itu, terdapat fitur *background sync* untuk sinkronisasi data pengguna yang prosesnya berada di *background*.

*Progressive Website Application* akan meningkatkan performa website dengan membuatnya menjadi lebih cepat, dapat diandalkan, dan mengikat pengguna (Karpagam, 2017).

## 2.5 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).

## 2.6 NFC

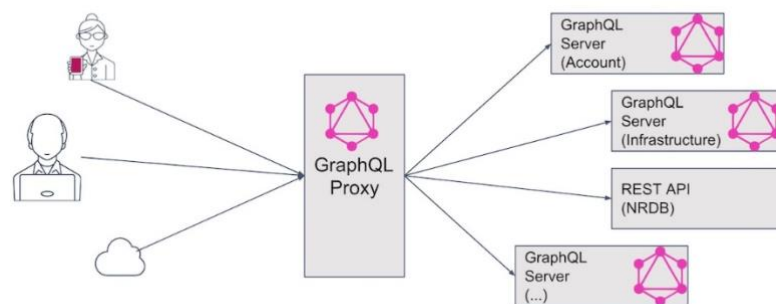
(Fernández et al., 2013) melakukan penelitian tentang pengontrolan kehadiran/absensi pada *higher education* menggunakan NFC. Penelitian ini dilakukan di Eropa karena adanya masalah guru yang kehabisan waktu untuk melakukan cek absensi setiap mengajar. Hasil dari penelitian ini membawa keuntungan baik bagi bidang edukasi, maupun siswa yang dapat mengikuti pembelajaran secara lebih efisien.

## 2.7 QR Code

(Cata, T., Patel, P. S., & Sakaguchi, 2013) melakukan penelitian tentang penggunaan QR Code pada market. Penelitian ini membahas tentang sejarah dan latar belakang QR Code, kasus yang tepat untuk penggunaan QR Code, dan diakhiri dengan memberikan saran untuk penggunaan yang potensial terhadap QR Code pada praktik market.

## 2.8 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 8ahasa pemrograman *server* yang lain dengan baik (Buna, 2016).

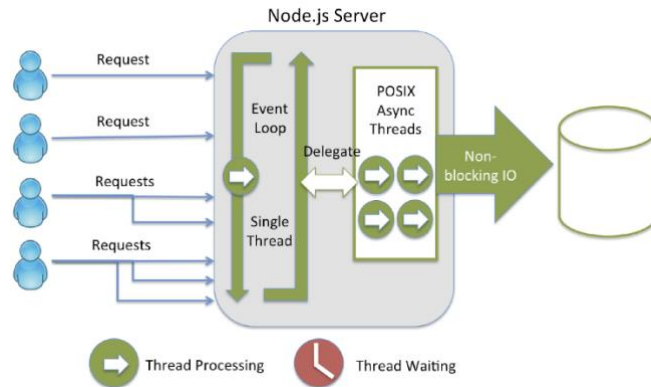


**Gambar 2.1 Graphql diagram**

(sumber: [blog.newrelic.com](http://blog.newrelic.com))

## 2.9 Node JS

Node js merupakan suatu perangkat lunak untuk pengembangan aplikasi dari bahasa pemrograman Javascript yang memungkinkan Javascript dijalankan pada sisi *server*. Node.js berjalan dengan basis event dimana pada suatu kode program dijalankan hingga selesai. Setelah kode program selesai dijalankan, baru dialihkan ke kode program selanjutnya. Node JS memiliki banyak modul yang berguna sehingga tidak perlu menulis semua kode dari awal (Kiessling, 2015).



**Gambar 2.2 Node JS flow**

(sumber: strongloop.com)

## 2.10 Model Pengembangan Perangkat Lunak

Dalam proses pengembangan perangkat lunak, terdapat beberapa model pengembangan yang ditawarkan, salah satunya adalah model yang dipakai dalam penelitian ini yaitu Scrum. Model-model proses untuk *software engineering* seperti *model sekuensial linier* atau *model waterfall*, *model spiral*, *model prototype*, *model RAD*, *model inkremental*, *model pengembangan konkuren* dan *model metode formal* (Roger, 1997).

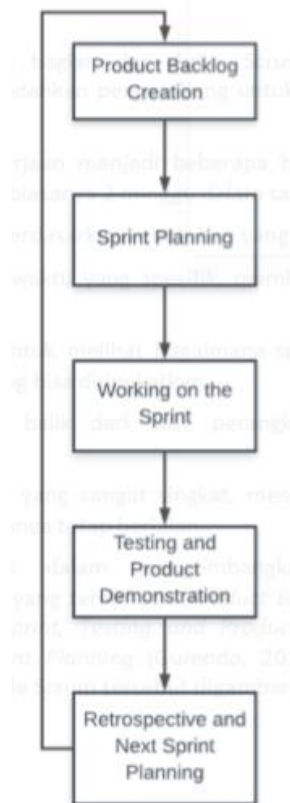
### 2.10.1 Scrum

Pengembangan sistem Sans menggunakan metode Scrum karena metode ini dapat digunakan digunakan dalam pengembangan sistem yang memiliki tingkat kompleksitas tinggi dan perubahan terus-menerus. Scrum adalah salah satu bagian dari *Agile*. Scrum memiliki beberapa karakteristik yang akan memudahkan pengembang untuk bekerja. Karakteristik dari Scrum adalah:

1. Memecah durasi pekerjaan menjadi beberapa bagian yang kemudian disebut dengan *sprint* (biasanya 2 minggu dalam satu *sprint*).
2. Merencanakan *sprint* berdasarkan kebutuhan yang penting pada saat itu.
3. Tidak memperkirakan waktu yang spesifik, membandingkan banyaknya pekerjaan.
4. Mengevaluasi *sprint* untuk melihat bagaimana *sprint* itu berjalan, dan melihat apakah ada yang bisa ditingkatkan.
5. Mendapatkan umpan balik dari hasil perangkat lunak yang telah dikerjakan.
6. Pertemuan setiap hari yang sangat singkat, mengidentifikasi halangan, dan memastikan semuanya tetap berjalan.

Ada beberapa tahapan dalam mengembangkan aplikasi dengan menggunakan metode Scrum yang terdiri atas *Product Backlog Creation*, *Sprint Planning*, *Working on the Sprint*, *Testing and Product Demonstration*, dan

*Restrospective and Next Sprint Planning* (Gurendo, 2015). Adapun alur dari tahapan-tahapan dalam metode Scrum tersebut digambarkan pada Gambar 2.3.



**Gambar 2.3 Alur Pengembangan dalam Metode Scrum**

*Product Backlog* adalah sebuah daftar yang terdiri atas fitur-fitur yang akan diimplementasikan selama proses pengembangan aplikasi. Fitur-fitur tersebut diurutkan berdasarkan prioritas. Dalam fase *product backlog creation* daftar fitur-fitur dibuat berdasarkan hasil wawancara kepada calon pelanggan yang akan dijadikan *User Story*.

*Sprint Planning* merupakan tahap perencanaan *sprint* yang akan dilakukan dalam masa pengembangan aplikasi. Durasi *sprint* tersebut harus ditentukan terlebih dahulu. *Sprint* yang singkat akan membuat aplikasi tersebut rilis dengan frekuensi yang lebih sering. Setelah itu, tim akan menentukan *User Story* yang paling penting untuk diselesaikan terlebih dahulu. Setiap anggota tim harus menentukan berapa lama mereka akan menyelesaikan suatu task.

*Working on the Sprint* merupakan proses implementasi dari aplikasi yang dikembangkan. Untuk melacak kinerja tim, biasanya sebuah *Task Board* digunakan. *Task Board* berisi *User Story* yang telah direncanakan pada tahap sebelumnya. Ada beberapa kolom dalam *Task Board* yang bisa merepresentasikan status pengerjaan dari *User Story* yang bersangkutan seperti *To Do*, *In Progress*, *Testing*, dan *Done*.

*Testing and Product Demonstration* merupakan hasil dari setiap *sprint* yang telah dilakukan. Tim membuat ulasan dan mendemonstrasikan hasil pekerjaan

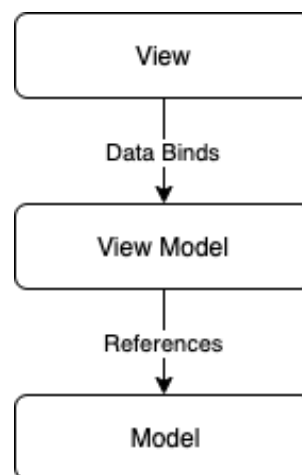


mereka. Setelah itu *stakeholder* dapat menentukan keputusan selanjutnya yang berkaitan dengan kelanjutan pengembangan aplikasi selanjutnya.

*Retrospective and Next Sprint Planning* bertujuan untuk mendiskusikan hasil *sprint* dan menentukan cara untuk memperbaiki proses pengembangan pada *sprint* selanjutnya. Tim memberikan kesimpulan apa yang berjalan lancar dan apa yang bisa diperbaiki pada iterasi berikutnya.

## 2.11 MVVM Design Pattern

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 2.4 (Vanus et al., 2016)



**Gambar 2.4 Pattern MVVM**

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 memerdulikan 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)

## 2.12 Pengujian Perangkat Lunak

### 2.12.1 Blackbox Testing

*Black Box Testing* merupakan pengujian perangkat lunak untuk mengetahui fungsi, masukan dan keluaran dari perangkat lunak telah sesuai dengan

spesifikasi yang dibutuhkan atau belum dari segi spesifikasi fungsional tanpa menguji desain dan kode program (Rosa & Salahuddin, 2011) . Uji coba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya yaitu fungsi-fungsi yang salah atau hilang, *interface*, struktur data atau akses *database* eksternal, kesalahan performa ataupun inisialisasi dan terminasi (Ayuliana, 2009).

### **2.12.2 Usability Testing**

*Usability testing* atau uji ketergunaan sebuah aplikasi merupakan salah satu metode evaluasi yang digunakan untuk mengukur tingkat kemudahan dan kenyamanan penggunaan dan interaksi pengguna terhadap sebuah sistem informasi (Henriyadi & Mulyati, 2016). Usability menurut standar internasional ISO 9241 (Ergonomic Requirements for Office Work with Visual Display Terminals) adalah pengukuran sejauh mana suatu produk dapat digunakan oleh yang sudah ditentukan untuk mencapai suatu tujuan tertentu sesuai dengan efektivitas, efisiensi, dan kepuasan penggunaan yang telah ditetapkan (Standard, 1998).

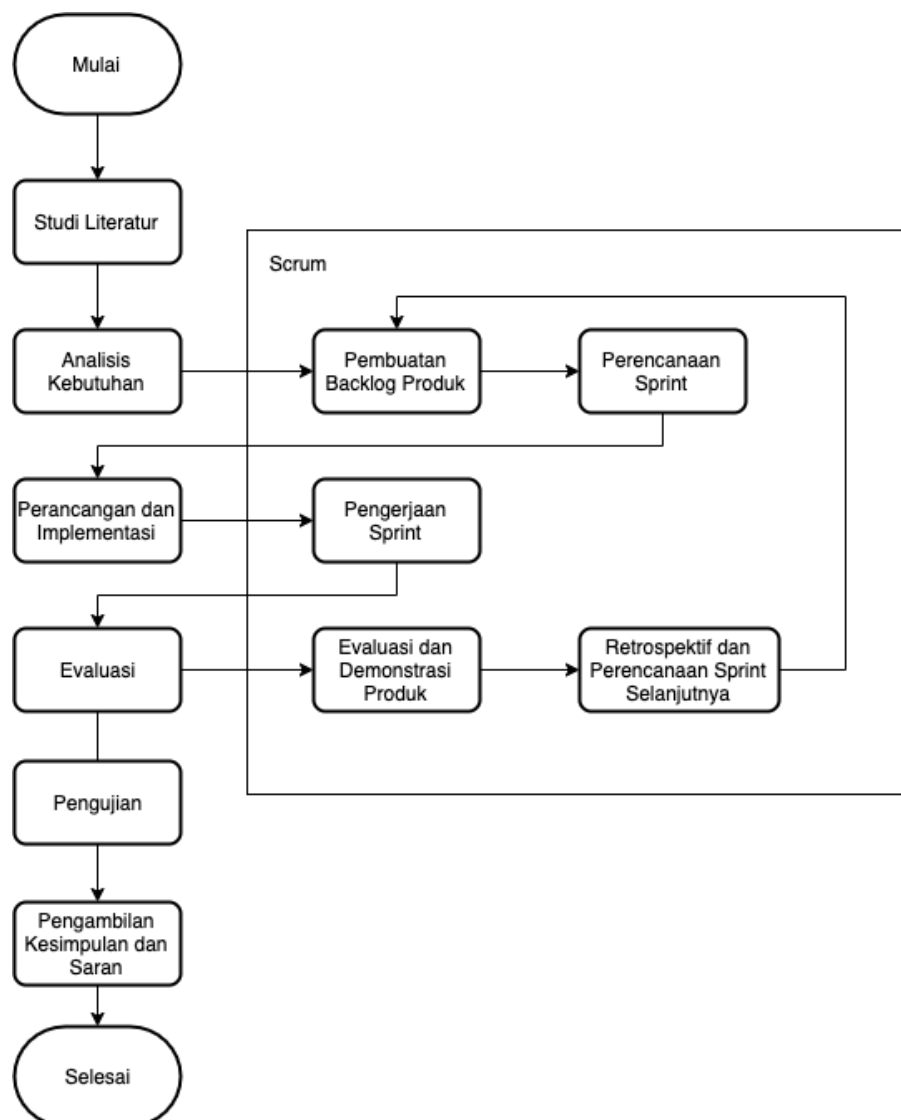
### **2.12.3 Regression Testing**

*Regression testing* adalah pengujian yang berfokus pada variasi yang terjadi dalam daur hidup sebuah perangkat lunak, dan menghasilkan kualitas software yang akan memberikan efek samping. *Regression testing* digunakan untuk memonitor perubahan pada sebuah perangkat lunak dan memberikan timbal balik terhadap perubahan tersebut secara teratur dan konsisten. Ketika perangkat lunak diubah, beberapa aspek dari perangkat lunak seperti konfigurasi dan program sebelumnya juga dapat berubah, maka *regression testing* akan digunakan untuk menjamin bahwa perubahan tersebut tidak akan memberikan sebuah error atau bugs pada fitur lain dikarenakan akan diuji secara keseluruhan setiap penambahan fitur baru (Xiaowen, 2013).

## BAB 3 METODOLOGI

### 3.1 Diagram Alir Metode

Pada bab ini akan membahas alur pelaksanaan atau tahapan-tahapan dalam pengembangan sistem Sans. Hal ini dilakukan agar proses dapat terarah dengan baik dan sesuai dengan tujuan. Metode dalam pengembangan perangkat lunak Sans adalah dengan metode SDLC Scrum. Tahapan-tahapan yang dilakukan pada penelitian ini digambarkan dalam Gambar 3.1 berikut:



Gambar 3.1 Alur Metodologi Penelitian

### 3.2 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. Pada tahap analisis kebutuhan terdapat dua bagian yaitu gambaran umum sistem dan analisis kebutuhan perangkat lunak. Di dalam analisis kebutuhan perangkat lunak diantaranya adalah:

1. Menjelaskan mengenai identifikasi aktor.
2. Menjelaskan mengenai pembuatan *User Story*.
3. Menjelaskan mengenai daftar kebutuhan fungsional.
4. Menjelaskan mengenai pembuatan *backlog product*.
5. Menjelaskan mengenai perencanaan *sprint*.
6. Menjelaskan mengenai analisis data.
7. Menjelaskan mengenai *use case diagram*.
8. Menjelaskan mengenai *use case* skenario.

### **3.2.1 Pembuatan *Backlog* Produk dan Perancangan *Sprint***

Dalam fase analisis kebutuhan, hal yang pertama yang harus dilakukan adalah mendeskripsikan secara umum aplikasi yang akan dibuat agar lebih mudah dipahami. Setelah itu, menentukan target pelanggan aplikasi yang selanjutnya akan menjadi aktor dalam sistem. Kemudian, mendapatkan *User Story* dari aktor yang sudah ditentukan sebelumnya.

Setelah *User Story* didapatkan, selanjutnya adalah membuat daftar fungsionalitas sistem berdasarkan *User Story* yang didapatkan. Fungsionalitas inilah yang akan menjadi acuan utama dalam implementasi selanjutnya.

Selanjutnya adalah menentukan durasi *sprint* yang akan digunakan. Setelah mengetahui durasi *sprint*, maka *backlog* produk kemudian dibagi berdasarkan durasi *sprint* yang telah ditentukan. Durasi *sprint* ditentukan berdasarkan kompleksitas dari fungsionalitas yang dibuat.

## **3.3 Perancangan dan Implementasi**

Tahapan ini adalah tahap dalam menentukan perangkat keras (*hardware*), arsitektur diaya, dan sistem persyaratan yang dibutuhkan. Dalam tahapan ini, dilakukan segala persiapan akan sesuatu yang dibutuhkan dalam melakukan implementasi pada aplikasi Sans. Tahapan Perancangan Sistem diantaranya adalah :

1. Perancangan Arsitektur

Dalam perancangan arsitektur ini akan dilakukan pemodelan dengan menggunakan diagram UML seperti *sequence diagram* dan *class diagram*.

2. Perancangan Komponen

Dalam perancangan komponen ini akan dituliskan beberapa sampel algoritme utama yang diambil dari setiap klas pada *controller*. Algoritme ini akan ditulis dalam bentuk *pseudocode*

### 3. Perancangan Antarmuka

Perancangan antarmuka dari sistem yang akan dibangun ini terdiri dari tata letak komponen yang harus disediakan oleh sistem berdasarkan kebutuhan sistem. Dalam perancangan antarmuka ini akan dituliskan beberapa sampel antarmuka utama berdasarkan level pelanggan.

Pada tahap Implementasi yang dilakukan dalam menyelesaikan laporan ini menggunakan serangkaian kode program. Kode program tersebut menggunakan bahasa pemrograman Javascript. Tahapan implementasi sistem diantaranya adalah menjelaskan spesifikasi sistem, batasan implementasi, implementasi basis data, implementasi kode program, dan implementasi antarmuka.

## 3.4 Evaluasi

Setiap kali sebuah sprint telah dilakukan, evaluasi harus dilakukan untuk menentukan bahwa semua *backlog* produk telah terselesaikan secara menyeluruh atau belum. Jika ternyata masih ada *backlog* produk yang belum terselesaikan, maka *backlog* tersebut akan dimasukkan ke dalam *backlog* produk untuk *sprint* selanjutnya.

### 3.4.1 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*.

### 3.4.2 Retrospektif dan Perancangan *Sprint* Selanjutnya

Setelah semua tahapan selesai dalam sebuah iterasi, maka selanjutnya dilakukan retrospektif. Retrospektif ini bertujuan untuk mengevaluasi seluruh proses *sprint* yang telah dilakukan sebelumnya untuk mengetahui hasil dan kesimpulan dari *sprint* yang telah dilakukan. Setelah itu, semua proses diulang kembali dari awal untuk iterasi selanjutnya dengan ikut mempertimbangkan hasil dan kesimpulan yang telah didapatkan dari fase retrospektif.

## 3.5 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 kepuasan pengguna terhadap sistem

yang telah dikembangkan sehingga pengguna dapat berinteraksi dengan nyaman dan puas terhadap sistem.

### **3.6 Pengambilan Kesimpulan dan Saran**

Pengambilan kesimpulan diambil sesuai dengan hasil dari pengujian dan analisis sehingga dapat diketahui inti dari penelitian ini. Kesimpulan yang ada akan menjadi menjawab untuk permasalahan yang telah dirumuskan dalam penelitian ini.

## DAFTAR REFERENSI

- Ayuliana. (2009). Testing dan Implementasi.
- 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>
- Fernández, M. J. L., Fernández, J. G., Aguilar, S. R., Selvi, B. S., & Crespo, R. G. (2013). Control of attendance applied in higher education through mobile NFC technologies. *Expert Systems with Applications*, 40(11), 4478–4489. <https://doi.org/10.1016/j.eswa.2013.01.041>
- Henriyadi, H., & Mulyati, R. (2016). USABILITY TESTING Sistem Informasi: Studi kasus pada Aplikasi Repositori Publikasi Badan Penelitian dan Pengembangan Pertanian. *Jurnal Perpustakaan Pertanian*, 23(2), 54. <https://doi.org/10.21082/jpp.v23n2.2014.p54-63>
- Islam, R., & Mazumder, T. (2010). Mobile application and its global impact. *International Journal of Engineering & ...*, (06), 72–78. Retrieved from <http://ijens.org/107506-0909 IJET-IJENS.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.
- Kiessling, M. (2015). *The Node Beginner Book: A Comprehensive Node.js Tutorial*.
- 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*, (2015), 1374–1378. <https://doi.org/10.1109/CYBER.2015.7288144>
- Marsum, W. (2005). *Restoran dan Segala Permasalahannya*. Jakarta: Andi.
- Rasid, A., Supriyono, S., & Setiawan, R. (2018). Aplikasi Reservasi Menu Restoran

- Berbasis Web Dan Mobile Android Di Cowek Ireng. *SITECH : Jurnal Sistem Informasi Dan Teknologi*, 1(1), 21–30. <https://doi.org/10.24176/sitech.v1i1.2273>
- Rosa, A., & Salahuddin, M. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- Singh, A., & Kanade, V. (2018). *Online food ordering system*. 374–378.
- Standard, I. (1998). *Iso 9241-11*. 1998.
- Susila, A. nur, Panji, D., & Prima, D. W. A. (2007). *Analisa Sistem Antrian Untuk Menentukan Tingkat Pelayanan Yang Optimal Pada Kasir (Server) Rumah Makan Kober Mie Setan Malang Dengan Metode Simulasi*. 23(39870423), 946–952. <https://doi.org/10.13989/j.cnki.0517-6611.2015.10.011>
- 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>
- Xiaowen, L. (2013). Research on regression testing methods for industry applications. *International Journal of Smart Home*, 7(6), 111–122. <https://doi.org/10.14257/ijsh.2013.7.6.11>