

PROPOSAL

**RANCANG BANGUN APLIKASI *VIRTUAL FOOD ORDERING*
DENGAN *QUICK RESPONSE (QR) CODE* MENGGUNAKAN
ALGORITMA *MULTILEVEL FEEDBACK QUEUE (MLFQ)*
BERBASIS *PROGRESSIVE WEB APPLICATION (PWA)*
(Studi Kasus : Rumah Makan Kampung Bakau)**

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Teknik



TIYAN ADDAR QUTHNI ATTIRMIDZI

E1E1 17 065

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HALU OLEO

2021

LEMBAR PENGESAHAN

Seminar Proposal

**RANCANG BANGUN APLIKASI *VIRTUAL FOOD ORDERING* DENGAN
QUICK RESPONSE (QR) CODE MENGGUNAKAN ALGORITMA
MULTILEVEL FEEDBACK QUEUE (MLFQ) BERBASIS *PROGRESSIVE*
WEB APPLICATION (PWA)
(Studi Kasus : Rumah Makan Kampung Bakau)**

Adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Universitas Halu Oleo ataupun institusi pendidikan lainnya.

Kendari, Maret 2021



Tiyan Addar Quthni Attirmidzi
NIM. E1E1 17 065

Pembimbing I



Stiswaty, S.T., M.MSI.
NIP. 19811107 200812 2 003

Pembimbing II



Rizal Adi Saputra, S.T., M.KOM.
NIP. 19910406 201903 1 021

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	v
DAFTAR TABEL.....	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	4
1.7 Tinjauan Pustaka.....	5
BAB II LANDASAN TEORI.....	7
2.1 Rumah Makan.....	7
2.2 Sistem Antrian.....	7
2.3 Model Antrian.....	8
2.4 Penjadwalan.....	10
2.5 Algoritma <i>Multilevel Feedback Queue</i> (MLFQ).....	10
2.6 <i>Quick Response</i> (QR) <i>Code</i>	17
2.7 <i>Progressive Web Application</i> (PWA).....	19
2.8 <i>Javascript Object Notation</i> (JSON).....	20
2.9 <i>Representational State Transfer</i> (REST).....	20
2.10 <i>Application Programming Interface</i> (API).....	21
2.11 Node.js.....	22
2.12 Mongo DB.....	22
2.13 Express.js.....	22
2.14 React.js.....	23
2.15 Tailwind CSS.....	23

2.16	<i>Insomnia Rest Client</i>	23
2.17	<i>Scanner Quick Response (QR) Code</i>	24
2.18	<i>Library Generate Quick Response (QR) Code</i>	24
2.19	<i>Unified Modelling Language (UML)</i>	24
2.20	<i>Metode Prototype</i>	28
BAB III METODOLOGI PENELITIAN.....		31
3.1	Metode Pengumpulan Data.....	31
3.2	Metode Pengembangan Sistem.....	31
3.3	Waktu dan Tempat Penelitian.....	33
3.3.1	Waktu Penelitian.....	33
3.3.2	Tempat Penelitian.....	33
3.4	Analisis Kebutuhan Sistem.....	33
3.4.1	Analisis Kebutuhan Fungsional.....	33
3.4.2	Analisis Kebutuhan <i>Non-Fungsional</i>	34
3.5	Analisis Perancangan Sistem.....	35
3.5.1	Analisis Sistem Yang Sedang Berjalan.....	35
3.5.2	Analisis Sistem Yang Akan Dikembangkan.....	36
3.5.3	Perancangan Arsitektur Sistem.....	36
3.5.4	Pemodelan <i>Unified Modeling Language (UML)</i>	37
3.5.4	Perancangan Antarmuka.....	45
DAFTAR PUSTAKA.....		51

DAFTAR GAMBAR

Gambar 2.1 <i>Single Channel Single Phase</i>	8
Gambar 2.2 <i>Single Channel Multi Phase</i>	9
Gambar 2.3 <i>Multi Channel Single Phase</i>	9
Gambar 2.4 <i>Multi Channel Multi Phase</i>	9
Gambar 2.5 <i>Multilevel Feedback Queue</i> (Salomo dkk., 2019)	11
Gambar 2.6 Implementasi Antrian <i>Multilevel Feedback Queue</i>	13
Gambar 2.7 Contoh <i>Quick Response (QR) Code</i>	18
Gambar 2.8 Anatomi <i>Quick Response (QR) Code</i> (Widayati, 2017)	18
Gambar 2.9 Contoh data dengan format JSON	20
Gambar 2.10 Metode <i>Prototype</i>	28
Gambar 2.11 Tahapan Metode <i>Prototype</i>	29
Gambar 3.1 Sistem Yang Sedang Berjalan	35
Gambar 3.2 Sistem Yang Berjalan Akan Dikembangkan	36
Gambar 3.3 Arsitektur Sistem	37
Gambar 3.4 Diagram <i>Use Case</i> Sistem	37
Gambar 3.5 Diagram <i>Activity</i> Sistem	41
Gambar 3.6 Diagram <i>Class</i> Sistem	42
Gambar 3.7 Diagram <i>Sequence Cashier</i> Sistem	43
Gambar 3.8 Diagram <i>Sequence Chef</i> Sistem	44
Gambar 3.9 Diagram <i>Sequence Customer</i> Sistem	45
Gambar 3.10 Tampilan Daftar Menu	46
Gambar 3.11 Tampilan Daftar Pesanan (a)	46
Gambar 3.12 Tampilan Daftar Pesanan (b)	47
Gambar 3.13 Tampilan <i>Dashboard Chef</i>	47
Gambar 3.14 Tampilan Daftar Pesanan	48
Gambar 3.15 Tampilan Daftar Pesanan Detail	48
Gambar 3.16 Tampilan Riwayat Pesanan	49
Gambar 3.17 Tampilan Penataan Meja Makan	49
Gambar 3.18 Tampilan Daftar Menu	50

DAFTAR TABEL

Tabel 2.1 Data Pesanan Di Restoran (<i>Dummy Data</i>)	11
Tabel 2.2 Parameter Antrian	12
Tabel 2.3 Pesanan Meja 1	13
Tabel 2.4 Pesanan Meja 3	13
Tabel 2.5 Pesanan Meja 6	14
Tabel 2.6 Pesanan Meja 4	14
Tabel 2.7 Proses-proses yang ada pada Q1	14
Tabel 2.8 Proses-proses yang ada pada Q2	15
Tabel 2.9 Proses-proses yang ada pada Q3	15
Tabel 2.10 Proses-proses yang ada pada Q4	16
Tabel 2.11 Proses-proses yang ada pada Q5	16
Tabel 2.12 Data Pesanan Yang Telah Diproses	17
Tabel 2.13 Simbol-simbol Diagram <i>Use Case</i>	25
Tabel 2.14 Simbol-simbol Diagram <i>Activity</i>	25
Tabel 2.15 Simbol-simbol Diagram <i>Sequence</i>	26
Tabel 2.16 Simbol-simbol Diagram <i>Class</i>	27
Tabel 3.1 <i>Gannt Chart</i> Waktu Penelitian	33
Tabel 3.2 Spesifikasi Perangkat Keras	34
Tabel 3.3 Spesifikasi Perangkat Lunak	35
Tabel 3.4 Deskripsi <i>Use Case Scan Table</i>	38
Tabel 3.5 Deskripsi <i>Use Case Order</i>	38
Tabel 3.6 Deskripsi <i>Use Case Authentication</i>	39
Tabel 3.7 Deskripsi <i>Use Case Manage Orders</i>	40
Tabel 3.8 Deskripsi <i>Use Case Manage The Table</i>	40
Tabel 3.9 Deskripsi <i>Use Case Manage Menu List</i>	41

BAB I

PENDAHULUAN

1.1 Latar Belakang

Rumah makan merupakan salah satu industri yang berkembang dengan cepat. Perkembangan industri ini pun terlihat sangat pesat di kota-kota besar di Indonesia, dimana setiap pengusaha rumah makan berlomba-lomba dalam mendapatkan pelanggan sebanyak-banyaknya dengan meningkatkan produk dan pelayanan. Dalam industri ini pun persaingan bukan hal yang sangat asing untuk ditemui, maka dari itu pihak pengelola dituntut untuk menerapkan strategi yang tepat agar tidak kalah dalam persaingan. Untuk menghadapi persaingan tersebut setiap rumah makan harus mempunyai keunggulan dari rumah makan lainnya, mulai dari pelayanan, suasana, cita rasa, fasilitas dan kecepatan penanganan masalah yang dihadapi oleh pelanggan sehingga pelanggan merasa puas, dihargai, dan menjadi loyal terhadap rumah makan tersebut (Salim, 2013).

Namun dalam pelaksanaannya, kadang muncul beberapa masalah yang membuat pelayanan menjadi tidak maksimal. Seperti waktu menunggu pesanan yang lama, terjadinya penumpukan kertas-kertas yang berisi pesanan pada sisi dapur sehingga terjadi kesalahan urutan pemrosesan pesanan dan tertukarnya pesanan dari pelanggan satu dengan pelanggan lainnya (Prasetyo dkk., 2016).

Rumah makan Kampung Bakau merupakan rumah makan yang berada tepat di atas hutan bakau (hutan *mangrove*) di tepi teluk Kendari. Rumah makan ini pada awalnya hanya menyediakan kudapan seperti ubi goreng dan pisang goreng dan juga hanya digunakan sebagai tempat perkumpulan yang bersifat pribadi oleh pemiliknya. Seiring permintaan dari pengunjung akhirnya rumah makan ini dibuka untuk umum pada tahun 2018 dan sekarang menyediakan berbagai macam menu makanan berdasarkan permintaan dari pengunjung. Rumah makan ini mendapatkan kunjungan sekitar 200 sampai 300 kunjungan setiap harinya.

Berdasarkan hasil pengamatan permasalahan yang dihadapi oleh rumah makan Kampung Bakau dan didasarkan keterangan pemilik yang diwakili oleh

manajer rumah makan tersebut adalah seringnya pesanan yang diantarkan tidak sesuai dengan pesanan pelanggan dan juga terkadang pesanan pelanggan bertukar dengan pelanggan lainnya. Ini membuat kepuasan pelanggan menjadi berkurang akibat permasalahan tersebut.

Untuk mengatasi permasalahan tersebut dibutuhkan sebuah sistem yang dapat membantu rumah makan Kampung Bakau dalam mengelola segala pesanan yang masuk di sisi dapur. Sistem tersebut akan menerapkan algoritma *Multilevel Feedback Queue* (MLFQ). Algoritma MLFQ adalah algoritma yang memiliki level antrian dan dapat berpindah dari level antrian awal ke level antrian berikutnya. Algoritma MLFQ menjadwalkan dengan cara memasukkan proses ke beberapa antrian (*queue*). Masing-masing *queue* memiliki algoritma sendiri seperti *Round Robin* (RR), *First Come First Served* (FCFS) atau algoritma lainnya. Algoritma MLFQ biasa digunakan pada penjadwalan CPU (Prasetyo dkk., 2016).

Sistem ini pun juga akan memanfaatkan teknologi *Quick Response* (QR) *Code* dan *Progressive Web Application* (PWA). *Quick Response* (QR) *Code* merupakan jenis kode matriks yang dikembangkan oleh Denso Corporation yang merupakan sebuah perusahaan besar di negara Jepang dan dipublikasikan pada tahun 1994, dengan tujuan sebagai salah satu teknologi kode penerjemah yang mempunyai kecepatan tinggi (Dodu dkk., 2018). *Progressive Web Application* (PWA) sebuah istilah untuk aplikasi berbasis web yang menggunakan teknologi web paling modern. PWA digambarkan sebagai kumpulan dari teknologi, konsep desain dan web API (*Application Programming Interface*) yang bekerja secara bersama untuk memberikan sentuhan aplikasi pada sebuah *mobile web* (Kurniawan dkk., 2018).

Quick Response (QR) *Code* akan dimanfaatkan sebagai penerjemah kode meja dari keberadaan pelanggan dan untuk *Progressive Web Application* (PWA) akan dimanfaatkan sebagai aplikasi tempat pelanggan melakukan pemesanan. Digunakannya teknologi PWA ini agar pelanggan menggunakan aplikasi ini tanpa harus menginstal terlebih dahulu.

Berdasarkan latar belakang yang telah diuraikan diatas, maka penulis mengambil topik penelitian dengan judul **“RANCANG BANGUN APLIKASI *VIRTUAL FOOD ORDERING* DENGAN *QUICK RESPONSE (QR) CODE* MENGGUNAKAN ALGORITMA *MULTILEVEL FEEDBACK QUEUE (MLFQ)* BERBASIS *PROGRESSIVE WEB APPLICATION (PWA)*”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya maka rumusan masalah dari penelitian ini adalah sebagai berikut :

1. Bagaimana merancang dan membangun Aplikasi *Virtual Food Ordering* dengan *Quick Response (QR) Code* ?
2. Bagaimana menerapkan algoritma *Multilevel Feedback Queue (MLFQ)* pada aplikasi *Virtual Food Ordering* dengan *Quick Response (QR) Code* ?

1.3 Batasan Masalah

Adapun batasan masalah dari sistem ini adalah sebagai berikut :

1. Penelitian ini hanya dilakukan dalam lingkup Rumah Makan Kampung Bakau.
2. Pelanggan hanya dapat melakukan pemesanan apabila memindai *Quick Response (QR) Code* pada meja pelanggan.
3. Pelanggan tidak dapat melakukan pembatalan pesanan ketika pesanan telah diterima oleh pihak dapur atau koki.
4. Perangkat *smartphone* pelanggan harus terhubung dengan jaringan internet.

1.4 Tujuan Penelitian

Adapun tujuan penelitian dari sistem ini adalah sebagai berikut :

1. Untuk merancang dan membangun Aplikasi *Virtual Food Ordering* dengan *Quick Response (QR) Code*.
2. Untuk menganalisis penerapan Algoritma *Multilevel Feedback Queue (MLFQ)* pada aplikasi *Virtual Food Ordering* dengan *Quick Response (QR) Code*.

1.5 Manfaat Penelitian

Adapun manfaat penelitian dari sistem ini adalah sebagai berikut :

1. Diharapkan Algoritma *Multilevel Feedback Queue* (MLFQ) dapat diterapkan dalam pemesanan makanan pada Rumah Makan Kampung Bakau.
2. Diharapkan aplikasi ini meningkatkan pelayanan agar lebih cepat dan efisien.

1.6 Sistematika Penulisan

Adapun sistematika penulisan proposal Tugas Akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

BAB II LANDASAN TEORI

Bab ini berisi teori-teori penunjang yang berhubungan dengan penelitian berupa teori dari metode yang digunakan diantaranya yaitu Rumah Makan, Sistem Antrian, Model Antrian, Penjadwalan, *Multilevel Feedback Queue* (MLFQ), *Quick Response* (QR) Code, *Progressive Web Application* (PWA), *Javascript Object Notation* (JSON), *Representational State Transfer* (REST), *Application Programming Interface* (API), Node.js, Mongo DB, Express.js, React.js, Tailwind CSS, *Insomnia Rest Client*, *Scanner Quick Response* (QR) Code, *Library Generate Quick Response* (QR) Code, *Unified Modelling Language* (UML) dan Metode *Prototype*.

BAB III METODOLOGI PENELITIAN

Bab ini berisi metode pengumpulan data, metode pengembangan sistem, waktu dan tempat penelitian, analisis kebutuhan sistem dan analisis perancangan sistem.

1.7 Tinjauan Pustaka

Hasil penelitian sebelumnya yang menjadi acuan dan referensi penyusunan tugas akhir ini, penulis mengambil dari penelitian yang pernah dilakukan :

Penelitian yang dilakukan oleh (Prasetyo dkk., 2016) dengan judul “Pemodelan Penjadwalan *Multilevel Feedback Queue* Menggunakan *Dynamic Time Quantum* Pada Kasus Pemesanan Makanan Di Restoran”. Hasil dari 19 percobaan dapat diperoleh, rata-rata *response time* MLFQ lebih cepat dibanding FCFS, baik pada parameter koki berjumlah 3, 4, maupun 5. Pada percobaan yang sama disimpulkan bahwa rata-rata *response time* MLFQ DTQ dominan lebih cepat dibandingkan MLFQ, dengan perbandingan 15 percobaan pada koki berjumlah 3 orang, 14 percobaan pada koki berjumlah 4 orang, dan 17 percobaan pada koki berjumlah 5 orang. Hasil dari 19 percobaan dapat diperoleh, rata-rata *response time* MLFQ vertikal lebih cepat dibandingkan dengan FCFS. Pada percobaan yang sama disimpulkan bahwa rata-rata *response time* MLFQ lebih cepat dibandingkan dengan MLFQ vertikal dan disimpulkan pula rata-rata *response time* MLFQ DTQ lebih cepat dibandingkan dengan MLFQ DTQ vertikal. Pengerjaan secara horizontal atau biasa memiliki keunggulan pada segi *response time*, sedangkan pengerjaan secara vertikal memiliki keunggulan pada segi *fairness*. MLFQ vertikal dianggap paling bagus dari pada algoritma pembanding yang lain karena paling unggul dari segi *fairness* dan memiliki rata-rata *response time* yang lebih baik jika dibandingkan dengan FCFS.

Penelitian yang dilakukan oleh (Manesia dkk., 2020) dengan judul “Perancangan Aplikasi Daftar Menu Restoran Berbasis Android Menggunakan Metode *Multilevel Queue*”. Prosedur antrian pemesanan makanan dapat dilakukan pada aplikasi android sehingga pelanggan tidak lagi harus mencatat pesanan secara manual, karna aplikasi sudah tersedia pada meja masing-masing pelanggan. Algoritma *multilevel queue* dapat diterapkan dalam perancangan aplikasi daftar menu makanan dengan melakukan pengurutan pemesanan berdasarkan waktu pemesanan dan menghitung besar nilai *quantum time* pada setiap proses, sehingga diperoleh hasil yang maksimal. Aplikasi daftar menu makanan dirancang dengan menggunakan *Software Eclipse Juno* sebagai editor

yaitu menggunakan bahasa pemograman Java dan menggunakan SQLite sebagai *database* untuk kemudian dapat di jalankan pada aplikasi berbasis android.

Penelitian yang dilakukan oleh (Salomo dkk., 2019) dengan judul “*Multilevel Algorithm Feedback Queue for Restaurant Order System*”. Hasil Cara mendesain informasi antrian pesanan sistem yang menggunakan desain UML telah menemukan 5 modul yaitu login, pemesanan, pembayaran, *quiz*, laporan terstruktur sehingga dapat mengatasi antrian pemesanan makanan di era globalisasi ini. Hasil Uji Coba 19 dan Perbandingan Rata-rata *Response Time* perslot, pada Tabel 4 terlihat rata-rata waktu respon MLFQ lebih cepat dari pada FCFS, baik parameter chef adalah 3, 4, dan 5.

BAB II

LANDASAN TEORI

2.1 Rumah Makan

Rumah makan adalah salah satu industri yang berkembang dengan cepat, khususnya di area perkotaan, karena daerah perkotaan dianggap tempat yang strategis dan mempunyai penduduk yang banyak, semakin banyak jumlah penduduk kemungkinan untuk mendapatkan pangsa pasar makin besar. Semakin diterimanya rumah makan oleh masyarakat, maka persaingan antar rumah makan terhadap kualitas produk dan layanan akan semakin terlihat di masa mendatang, perusahaan diharapkan lebih mengerti kebutuhan pelanggannya, sehingga konsumen tetap percaya terhadap produk yang ia pakai dan penyedia jasa perlu menggunakan strategi khusus agar perusahaan tersebut tidak tertinggal dari yang lain (Saputra, 2014).

2.2 Sistem Antrian

Sistem antrian adalah hasil pengembangan yang mengatur pelayanan yang berurutan sesuai kedatangan pelanggan, untuk mencapai sebuah kinerja yang efektif dan efisien. Antrian terjadi disebabkan oleh jumlah kebutuhan akan layanan yang melebihi kapasitas fasilitas layanan, sehingga pengguna fasilitas layanan yang tiba tidak bisa segera mendapat layanan dikarenakan kesibukan layanan. Disiplin antrian adalah konsep yang membahas mengenai kebijakan dalam menentukan pelanggan mana yang dipilih dari antrian untuk dilayani, berdasarkan urutan kedatangan pelanggan. Ada empat bentuk disiplin pelayanan yang biasa digunakan dalam praktek (Prasetyo dkk., 2016), yaitu :

1. *First In First Out* (FIFO) atau *First Come First Served* (FCFS)

Aturan pelayanan ini menerapkan pelanggan pertama yang datang maka pelanggan tersebut yang pertama dilayani. Misal, antri membeli bensin, antri melakukan *service* sepeda motor, antri membeli tiket bioskop, dan lain-lain.

2. *Last In First Out (LIFO)* atau *Last Come First Served (LCFS)*

Aturan pelayanan ini menerapkan pelanggan yang terakhir datang akan dilayani pertama kali. Misalnya antrian dalam lift, pelanggan yang terakhir naik lift adalah pelanggan yang akan pertama kali keluar dari lift.

3. *Service In Random Order (SIRO)* atau *Random Selection for Service (RSS)*

Aturan pelayanan ini menerapkan setiap pelanggan yang datang dan mengantri dalam sistem memiliki kesempatan yang sama untuk dilayani terlebih dahulu. Artinya pada disiplin antrian ini menggunakan pelayanan secara acak. Misalnya adalah antrian dalam arisan karena pelayanan yang dilakukan menggunakan undian di mana setiap orang yang ikut dalam antrian arisan tersebut memiliki kesempatan atau peluang yang sama untuk memenangkan arisan.

4. *Priority Service (PS)*

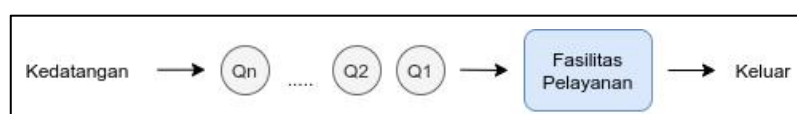
Aturan pelayanan ini menerapkan pemberian pelayanan kepada pelanggan yang memiliki prioritas lebih tinggi dibandingkan pelanggan yang memiliki prioritas lebih rendah meskipun pelanggan yang memiliki prioritas lebih tinggi tersebut datang paling akhir di garis tunggu. Misalnya pada antrian pelayanan rumah sakit, di mana rumah sakit akan melayani pasien yang memiliki tingkat penyakit yang lebih serius.

2.3 Model Antrian

Ada empat model struktur antrian dasar yang umum terjadi dalam seluruh sistem antrian (Bahar dkk., 2018) :

1. *Single Channel Single Phase*

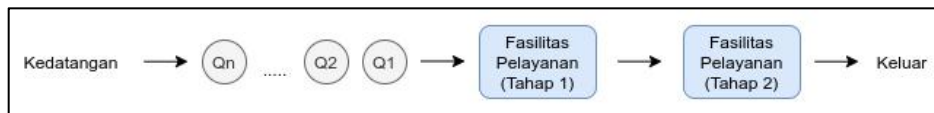
Single Channel Single Phase memiliki arti satu jalur satu tahap berarti bahwa dalam sistem tersebut hanya terdapat satu pemberi layanan serta satu jenis layanan yang diberikan, sehingga yang telah menerima pelayanan dapat langsung keluar dari sistem antrian.



Gambar 2.1 *Single Channel Single Phase*

2. *Single Channel Multi Phase*

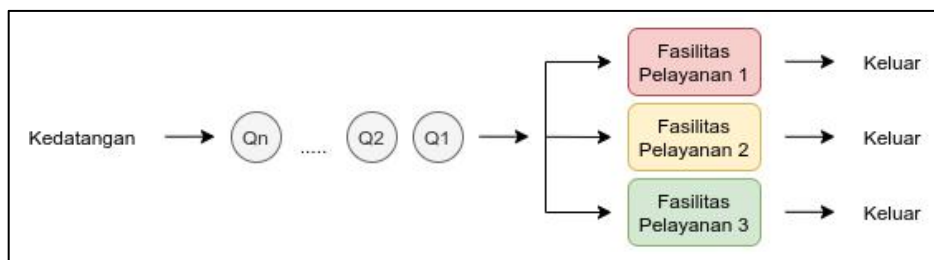
Single Channel Multi Phase memiliki arti satu jalur banyak tahap berarti dalam sistem antrian tersebut terdapat lebih dari satu jenis layanan yang diberikan, tetapi dalam setiap jenis layanan hanya terdapat satu pemberi layanan.



Gambar 2.2 *Single Channel Multi Phase*

3. *Multi Channel Single Phase*

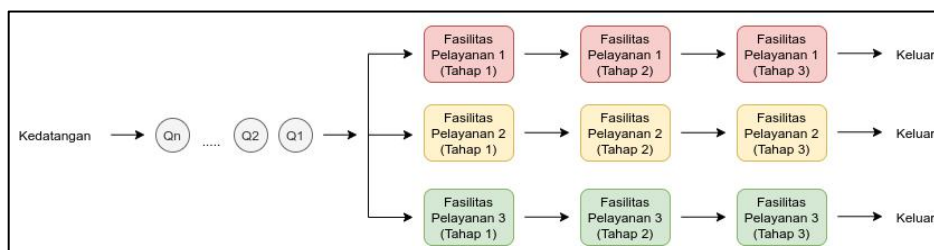
Multi Channel Single Phase memiliki arti banyak jalur satu tahap adalah sistem antrian dimana terdapat satu jenis layanan dalam sistem antrian tersebut, namun terdapat lebih dari satu pemberi layanan.



Gambar 2.3 *Multi Channel Single Phase*

4. *Multi Channel Multi Phase*

Multi Channel Multi Phase memiliki arti banyak jalur banyak tahap adalah sistem antrian dimana terdapat lebih dari satu jenis layanan dan terdapat lebih dari satu pemberi layanan dalam setiap jenis layanan.



Gambar 2.4 *Multi Channel Multi Phase*

2.4 Penjadwalan

Penjadwalan adalah aktivitas perencanaan untuk menentukan kapan dan dimana setiap operasi sebagai bagian dari pekerjaan secara keseluruhan harus dilakukan pada sumber daya yang terbatas, serta pengalokasian sumber daya pada suatu waktu tertentu dengan memperhatikan kapasitas sumber daya yang ada. Penjadwalan dapat diartikan sebagai pengalokasian sejumlah sumber daya (*resource*) untuk melakukan sejumlah tugas atau operasi dalam jangka waktu tertentu dan merupakan proses pengambilan keputusan yang peranannya sangat penting dalam industri manufaktur dan jasa yaitu mengalokasikan sumber-sumber daya yang ada agar tujuan dan sasaran perusahaan lebih optimal (Baker and Trietsch, 2009).

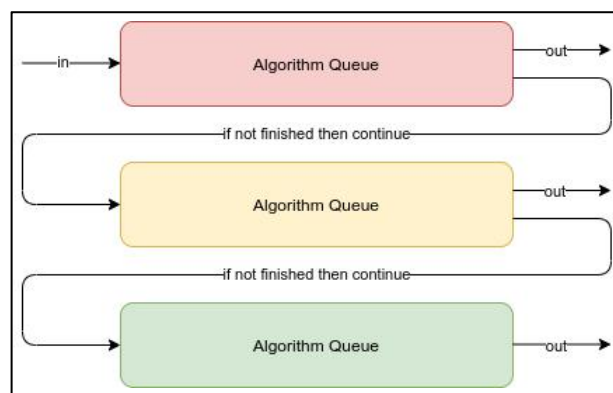
2.5 Algoritma *Multilevel Feedback Queue* (MLFQ)

Multilevel Feedback Queue atau biasa disingkat dengan MLFQ, menjadwalkan proses dengan cara memasukkan proses-proses ke dalam beberapa antrian (*queue*). Masing-masing antrian (*queue*) memiliki algoritma sendiri seperti *Round Robin* (RR), *First Come First Served* (FCFS) dan algoritma lainnya. Berikut ini merupakan penjelasan dari algoritma-algoritma yang sering diterapkan pada algoritma *Multilevel Feedback Queue* (Prasetyo dkk., 2016) :

1. *First Come First Served* (FCFS), merupakan aturan dimana pelanggan yang datang lebih awal yang akan dilayani terlebih dahulu. Algoritma ini tidak cocok bila diterapkan pada kondisi yang *realtime*. Ini dikarenakan suatu proses yang dapat memonopoli dengan eksekusi yang panjang dan lama sehingga menghalangi proses pendek untuk dieksekusi. Selain itu, jika proses yang sedang dikerjakan memakan waktu yang lama maka akan berakibat meningkatnya *response time* dan *waiting time* proses selanjutnya.
2. *Round Robin* (RR), merupakan versi *preemptive* dari *First Come First Served*. Pada algoritma *Round Robin* ini, proses dimasukkan pada *first in first out* tetapi setiap proses diijinkan untuk berjalan dalam waktu yang terbatas. Interval waktu ini biasa disebut *time slice* atau *quantum time*. Jika suatu proses tidak selesai dieksekusi hingga *time slice* habis, maka proses tersebut

preempted atau berhenti hingga mendapat giliran kembali dieksekusi untuk dieksekusi.

Algoritma *Multilevel Feedback Queue* (MLFQ) juga merupakan salah satu algoritma yang didasarkan pada model antrian *Multi Channel Single Phase* yang dimana pada antriannya terdapat beberapa tingkat. Hal ini dapat dimanfaatkan untuk menempatkan pelayanan pada tiap-tiap tingkat antrian. Adapun gambaran dari algoritma *Multilevel Feedback Queue* (MLFQ) dapat dilihat pada Gambar 2.5, berikut.



Gambar 2.5 *Multilevel Feedback Queue* (Salomo dkk., 2019)

Cara kerja dari algoritma *Multilevel Feedback Queue* (MLFQ) akan diterapkan dengan mendefinisikan terlebih dahulu parameternya, yaitu :

- 1.) Jumlah antrian.
- 2.) Algoritma tiap antrian.
- 3.) Kapan menaikkan proses ke antrian yang lebih tinggi.
- 4.) Kapan menurunkan proses ke antrian yang lebih rendah.
- 5.) Antrian mana yang akan dimasuki proses membutuhkan.

Tabel 2.1 Data Pesanan Di Restoran (*Dummy Data*)

Nomor Meja	Waktu Pesanan (OT) (H)	Kode	Pesanan	Jumlah	Waktu Proses (BT) (M)
1 (P1)	10.00	P1M1	Cumi Goreng Tepung	1	18
		P1M2	Ikan Katamba Bakar	2	24
		P1M3	Cah Pakis	2	13
		P1M4	Es Teh	3	5
		P1M5	Nasi Bakul Besar	1	4

3 (P2)	10.01	P2M1	Ikan Putih Bakar	3	22
		P2M2	Cah Kangkung	1	12
		P2M3	Cah Pakis	1	13
		P2M4	Nasih Bakul Besar	1	4
		P2M5	Air Mineral	3	2
6 (P3)	10.12	P3M1	Ikan Putih Bakar	4	22
		P3M2	Cah Kangkung	2	12
		P3M3	Es Teh	2	5
		P3M4	Es Jeruk	2	6
		P3M5	Nasi Bakul Kecil	2	3
4 (P4)	10.20	P4M1	Ikan Baronang Bakar	1	25
		P4M2	Terong Goreng Tepung	1	12
		P4M3	Cumi Goreng Tepung	1	18
		P4M4	Cah Kangkung	2	12
		P4M5	Air Mineral	1	2
		P4M6	Es Teh	1	5
		P4M7	Nasi Bakul Besar	1	4

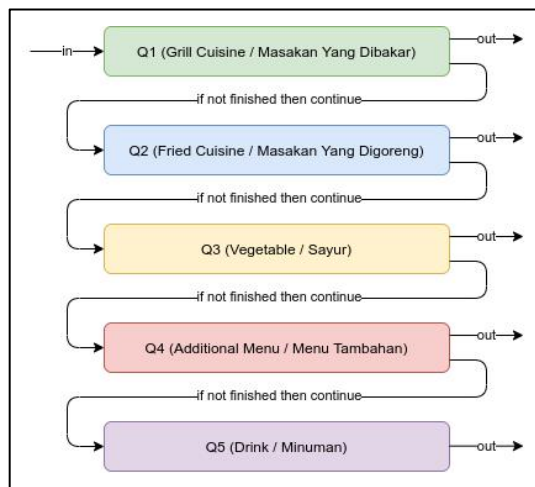
Adapun contoh kasus berdasarkan data dari Tabel 3.1 dan langkah-langkah dalam implementasi metode *Multilevel Feedback Queue* adalah sebagai berikut :

1. Menentukan parameter antrian

Tabel 2.2 Parameter Antrian

Parameter	Value
Jumlah Antrian	5
Algoritma Tiap Antrian	- Q1 <i>First Come First Served</i> (FCFS) - Q2 <i>First Come First Served</i> (FCFS) - Q3 <i>First Come First Served</i> (FCFS) - Q4 <i>First Come First Served</i> (FCFS) - Q5 <i>First Come First Served</i> (FCFS)
Kapan Menaikkan Proses Ke Antrian Yang Lebih Tinggi	-
Kapan Menurunkan Proses Ke Antrian Yang Lebih Rendah	Jika proses tidak sesuai dengan layanan antrian satu maka proses dipindahkan ke layanan antrian berikutnya.
Antrian Mana Yang Akan Dimasuki Proses Membutuhkan	<i>High priority - low priority</i> (Grill Cuisine / Pembakaran, <i>Fried Cuisine</i> / Penggoreng, <i>Vegetable</i> / Sayuran, <i>Additional Menu</i> / Menu Tambahan dan <i>Drink</i> / Minuman)

2. Memasukkan proses ke dalam antrian



Gambar 2.6 Implementasi Antrian *Multilevel Feedback Queue*

Adapun pesanan yang pertama kali masuk ke dalam antrian adalah sebagai berikut :

Tabel 2.3 Pesanan Meja 1

P1M5	P1M4	P1M3	P1M2	P1M1
------	------	------	------	------

Pada meja 1 memiliki 1 pesanan menu goreng, 1 pesanan menu bakar, 1 pesanan menu sayur, 1 menu minuman dan 1 pesanan menu tambahan yang dipesan pada pukul 10.00. Pesanan dengan kode P1M1 masuk ke Q2 (*fried cuisine*), untuk kode P1M2 masuk ke Q1 (*grill cuisine*), untuk kode P1M3 masuk ke Q3 (*vegetable*), untuk kode P1M4 masuk ke Q5 (*drink*) dan untuk kode P1M5 masuk ke Q4 (*additional menu*). Pada 1 menit kemudian tepat pada pukul 10.01 meja 3 melakukan pemesanan dengan pesanan sebagai berikut :

Tabel 2.4 Pesanan Meja 3

P2M5	P2M4	P2M3	P2M2	P2M1
------	------	------	------	------

Pada meja 3 memiliki 1 pesanan menu bakar, 2 pesanan menu sayur, 1 pesanan menu tambahan dan 1 pesanan menu minuman yang dipesan pada pukul 10.03. Pesanan dengan kode P2M1 masuk ke Q1 (*grill cuisine*), untuk kode P2M2 dan P2M3 masuk ke Q3 (*vegetable*), untuk kode P2M4 masuk ke Q4 (*additional menu*) dan untuk kode P2M5 masuk ke Q5 (*drink*). Pada 11 menit

kemudian tepat pada pukul 10.12 meja 6 melakukan pemesanan dengan pesanan sebagai berikut :

Tabel 2.5 Pesanan Meja 6

P3M5	P3M4	P3M3	P3M2	P3M1
------	------	------	------	------

Pada meja 6 memiliki 1 pesanan menu bakar, 1 pesanan menu sayur, 2 pesanan menu minuman dan 1 pesanan menu tambahan yang dipesan pada pukul 10.12. Pesanan dengan kode P3M1 masuk ke Q1 (*grill cuisine*), untuk kode P3M2 masuk ke Q3 (*vegetable*), untuk kode P3M3 dan P3M4 masuk ke Q5 (*drink*) dan untuk kode P3M5 masuk ke Q4 (*additional menu*). Pada 8 menit kemudian tepat pada pukul 10.20 meja 4 melakukan pemesanan dengan pesanan sebagai berikut :

Tabel 2.6 Pesanan Meja 4

P4M7	P4M6	P4M5	P4M4	P4M3	P4M2	P4M1
------	------	------	------	------	------	------

Pada meja 4 memiliki 1 pesanan menu bakar, 2 pesanan menu goreng, 1 pesanan menu sayur, 2 pesanan menu minuman dan 1 pesanan menu tambahan yang dipesan pada pukul 10.20. Pesanan dengan kode P4M1 masuk ke Q1 (*grill cuisine*), kode P4M2 dan P4M3 masuk ke Q2 (*fried cuisine*), untuk kode P4M4 masuk ke Q3 (*vegetable*), untuk kode P4M5 dan P5M6 masuk ke Q5 (*drink*) dan untuk kode P4M7 masuk ke Q4 (*additional menu*).

Jadi untuk proses-proses yang ada disetiap antrian-antrian adalah sebagai berikut :

- 1) Q1 masak yang dibakar (*grill cuisine*)

Tabel 2.7 Proses-proses yang ada pada Q1

<i>Order Time</i>	10.20	10.12	10.01	10.00
<i>Process</i>	P4M1	P3M1	P2M1	P1M2

Analisa waktu tunggu tambahan pada Q1 yang memiliki 3 koki yang akan memproses pesanan tersebut :

- a. P1M2 akan diproses oleh koki A tanpa adanya waktu tunggu tambahan (*additional waiting time*), karena koki sedang tidak memproses apapun.
- b. P2M1 akan diproses oleh koki B tanpa adanya waktu tunggu tambahan.

- c. P3M1 akan diproses oleh koki C tanpa adanya waktu tunggu tambahan.
 - d. P4M1 memiliki 3 menit waktu tunggu tambahan karena menunggu koki B selesai memproses P2M1 yang memiliki waktu proses (*burst time*) 22 menit.
- 2) Q2 masakkan yang digoreng (*fried cuisine*)

Tabel 2.8 Proses-proses yang ada pada Q2

<i>Order Time</i>	10.20	10.20	10.00
<i>Process</i>	P4M3	P4M2	P1M1

Analisa waktu tunggu tambahan pada Q2 yang memiliki 3 koki yang akan memproses pesanan tersebut :

- a. P1M1 akan diproses oleh koki A tanpa adanya waktu tunggu tambahan (*additional waiting time*), karena koki sedang tidak memproses apapun.
 - b. P4M2 akan diproses oleh koki B tanpa adanya waktu tunggu tambahan.
 - c. P4M3 akan diproses oleh koki C tanpa adanya waktu tunggu tambahan.
- 3) Q3 sayur (*vegetable*)

Tabel 2.9 Proses-proses yang ada pada Q3

<i>Order Time</i>	10.20	10.12	10.01	10.01	10.00
<i>Process</i>	P4M4	P3M2	P2M3	P2M2	P1M3

Analisa waktu tunggu tambahan pada Q3 yang memiliki 3 koki yang akan memproses pesanan tersebut :

- a. P1M3 akan diproses oleh koki A tanpa adanya waktu tunggu tambahan (*additional waiting time*), karena koki sedang tidak memproses apapun.
- b. P2M2 akan diproses oleh koki B tanpa adanya waktu tunggu tambahan.
- c. P2M3 akan diproses oleh koki C tanpa adanya waktu tunggu tambahan.
- d. P3M2 memiliki 1 menit waktu tunggu tambahan karena menunggu koki A selesai memproses P1M3 yang memiliki waktu proses (*burst time*) 13 menit.

4) Q4 menu tambahan (*additional menu*)**Tabel 2.10 Proses-proses yang ada pada Q4**

<i>Order Time</i>	10.20	10.12	10.01	10.00
<i>Process</i>	P4M7	P3M5	P2M4	P1M5

Analisa waktu tunggu tambahan pada Q4 yang memiliki 2 koki yang akan memproses pesanan tersebut :

- P1M5 akan diproses oleh koki A tanpa adanya waktu tunggu tambahan (*additional waiting time*), karena koki sedang tidak memproses apapun.
- P2M4 akan diproses oleh koki B tanpa adanya waktu tunggu tambahan.
- P2M3 akan diproses oleh koki A tanpa adanya waktu tunggu tambahan, karena telah menyelesaikan proses P1M5.
- P4M7 akan diproses oleh koki B tanpa adanya waktu tunggu tambahan, karena telah menyelesaikan proses P2M4.

5) Q5 minuman (*drink*)**Tabel 2.11 Proses-proses yang ada pada Q5**

<i>Order Time</i>	10.20	10.20	10.12	10.12	10.01	10.00
<i>Process</i>	P4M6	P4M5	P3M4	P3M3	P2M5	P1M4

Analisa waktu tunggu tambahan pada Q5 yang memiliki 2 koki yang akan memproses pesanan tersebut :

- P1M4 akan diproses oleh koki A tanpa adanya waktu tunggu tambahan (*additional waiting time*), karena koki sedang tidak memproses apapun.
- P2M5 akan diproses oleh koki B tanpa adanya waktu tunggu tambahan.
- P3M3 akan diproses oleh koki B tanpa adanya waktu tunggu tambahan, karena telah menyelesaikan proses P2M5.
- P3M4 akan diproses oleh koki A tanpa adanya waktu tunggu tambahan, karena telah menyelesaikan proses P1M4.
- P4M5 akan diproses oleh koki B tanpa adanya waktu tunggu tambahan, karena telah menyelesaikan proses P3M3.
- P4M6 akan diproses oleh koki A tanpa adanya waktu tunggu tambahan, karena telah menyelesaikan proses P3M4.

Terdapat beberapa proses yang mendapatkan waktu tunggu tambahan, karena adanya proses yang lebih dulu masuk kedalam antrian. Adapun data pesanan yang telah diproses dapat dilihat pada Tabel 2.12.

Tabel 2.12 Data Pesanan Yang Telah Diproses

Nomor Meja	OT (H)	Kode	BT	WT (M)	AWT (M)	DT (H)
1 (P1)	10.00	P1M1	18	24	0	10.24
		P1M2	24			
		P1M3	13			
		P1M4	5			
		P1M5	4			
3 (P2)	10.01	P2M1	22	22	1	10.24
		P2M2	12			
		P2M3	13			
		P2M4	4			
		P2M5	2			
6 (P3)	10.12	P3M1	22	22	0	10.32
		P3M2	12			
		P3M3	5			
		P3M4	6			
		P3M5	3			
4 (P4)	10.20	P4M1	25	25	3	10.45
		P4M2	12			
		P4M3	18			
		P4M4	12			
		P4M5	2			
		P4M6	5			
		P4M7	4			

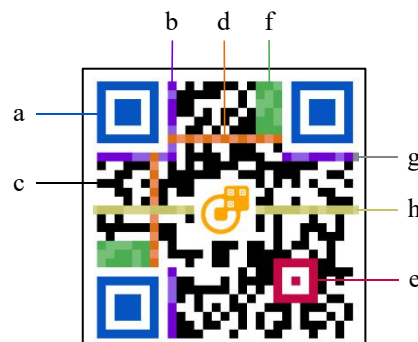
2.6 *Quick Response (QR) Code*

Quick Response Code atau yang dikenal dengan *QR-Code*, merupakan teknik yang mengubah data tertulis menjadi kode-kode 2 dimensi yang tercetak kedalam suatu media yang lebih ringkas. *Quick Response (QR) Code* diperkenalkan pertama kali oleh perusahaan Jepang Denso-Wave pada tahun 1994. *Barcode* jenis ini awalnya digunakan untuk melacak inventaris dalam pembuatan suku cadang kendaraan dan sekarang digunakan di berbagai industri (Rouillard, 2008).



Gambar 2.7 Contoh Quick Response (QR) Code

QR-Code mampu menyimpan semua jenis data, seperti data angka atau *numeric*, *alphanumeric*, *biner*, dan *kanji/kana*. Selain itu *QR-Code* memiliki tampilan yang lebih kecil daripada *barcode*. Hal ini dikarenakan *QR-Code* mampu menampung data secara horizontal dan vertikal, jadi secara otomatis ukuran dari tampilan gambar *QR-Code* bisa hanya sepersepuluh dari ukuran sebuah *barcode*. Tidak hanya itu *QR-Code* juga tahan terhadap kerusakan, sebab *QR-Code* mampu memperbaiki kesalahan sampai dengan 30% tergantung dengan ukuran atau versinya. Oleh karena itu, walaupun sebagian simbol *QR-Code* kotor ataupun rusak, data tetap dapat disimpan dan dibaca. Tiga tanda berbentuk persegi di tiga sudut memiliki fungsi agar simbol dapat dibaca dengan hasil yang sama dari sudut manapun (Rahmawati dan Rahman, 2011).



Gambar 2.8 Anatomi Quick Response (QR) Code (Widayati, 2017)

Adapun penjelasan dari anatomi *Quick Response (QR) Code* adalah sebagai berikut (Widayati, 2017) :

- a. *Finder Pattern*, berfungsi untuk identifikasi letak *QR-Code*.
- b. *Format Information*, berfungsi untuk informasi tentang *error correction level* dan *mask pattern*.
- c. *Data*, berfungsi untuk menyimpan data yang dikodekan.

- d. *Timing Pattern*, merupakan pola yang berfungsi untuk identifikasi koordinat pusat *QR-Code*, berbentuk modul hitam putih.
- e. *Alignment Pattern*, merupakan pola yang berfungsi memperbaiki penyimpangan *QR-Code* terutama distorsi *non linier*.
- f. *Version Information*, adalah versi dari sebuah *QR-Code*.
- g. *Quiet Zone*, merupakan daerah kosong di bagian terluar *QR-Code* yang mempermudah mengenali pengenalan *QR* oleh sensor *CCD*.
- h. *QR-Code version*, adalah versi dari *QR-Code* yang digunakan.

2.7 *Progressive Web Application (PWA)*

Progressive Web App atau biasa disingkat dengan istilah PWA, merupakan penggabungan antar unsur *Web* dan *App*. Di tahun 2018 pada acara tahunan yang diadakan oleh Google yaitu Google I/O, Google mengadakan kampanye penggunaan PWA. Disini Google berusaha menjembatani antara kemudahan manajemen *web application* dan kehandalan *mobile application* yang biasa berjalan di sistem operasi *mobile*, yaitu Android dan iOS. PWA sepenuhnya mengandalkan *browser* pengguna dan teknologi yang ada didalamnya. Berikut beberapa layanan yang disediakan pada *Progressive Web App (PWA)* (Syarifudin et al., 2019), antara lain :

- 1) *Service Worker*, merupakan salah satu jenis *web worker*, yaitu *script* yang berjalan di belakang *browser* pengguna. *Service worker* pada dasarnya adalah berkas JavaScript yang berjalan pada *thread* yang berbeda dengan *main thread browser*, menangani *network request*, *caching*, mengembalikan *resource* dari *cache*, dan bisa mengirimkan *push message*.
- 2) *Add To Homescreen*, merupakan layanan yang dapat menambahkan *icon* aplikasi pada *home screen smartphone*.
- 3) *Responsive, Progressive Web App* sangat *responsive* terhadap beberapa aplikasi baik berupa *smartphone*, *mobile*, *desktop* maupun *tablet*.

2.8 *Javascript Object Notation (JSON)*

Javascript Object Notation atau JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data (Herdiana, 2014).

Sebuah objek JSON adalah format data *key-value* yang biasanya di *render* di dalam kurung kurawal. Saat kita bekerja dengan JSON, kita akan sering melihat objek JSON disimpan di dalam sebuah file .json, tapi mereka juga dapat disimpan sebagai objek JSON atau string di dalam sebuah program. Adapun contoh dari JSON sendiri dapat dilihat pada Gambar 2.9 berikut.

```
{
  - name: {
    first_name: "Tiyan Addar Quthni",
    last_name: "Attirmidzi"
  },
  place_of_birth: "Unaaha",
  date_of_birth: "18 March 2000",
  sex: "Male",
  citizenship: "Indonesian",
  Profession: "Software Engineer"
}
```

Gambar 2.9 Contoh data dengan format JSON

2.9 *Representational State Transfer (REST)*

Representational State Transfer adalah suatu arsitektur metode komunikasi yang sering diterapkan dalam pengembangan layanan berbasis web. Arsitektur REST yang umumnya dijalankan via *Hypertext Transfer Protocol* (HTTP), melibatkan proses pembacaan laman web tertentu yang memuat sebuah file XML atau JSON. File inilah yang menguraikan dan memuat konten yang hendak disajikan. REST memiliki ciri khas pada interaksi *client* dan *server* yang difasilitasi oleh sejumlah tipe operasional (verba) dan *Universal Resource Identifiers* (URIs) yang unik bagi tiap-tiap sumber daya. Masing-masing verba

GET, *POST*, *PUT* dan *DELETE* memiliki makna operasional khusus untuk menghindari ambiguitas. REST sering dipergunakan dalam *mobile application*, situs web jejaring sosial, *mashup tools*, dan *automated business processes* (Firdaus dkk., 2019).

Pada REST terdapat beberapa kode respon HTTP yang sering digunakan ketika berinteraksi dengan server (Perkasa dan Setiawan, 2018), yaitu :

- 1) *200 : OK*, perintah yang dikirim ke server benar dan berhasil dijalankan.
- 2) *400 : Bad Request*, perintah yang dikirim ke server berisi isian yang salah.
- 3) *401 : Unauthorized*, pengirim mengirimkan kode yang salah.
- 4) *403 : Forbidden*, Pengirim tidak memiliki hak akses ke dalam resource yang dituju.
- 5) *404 : Not Found*, resource yang dituju tidak ditemukan didalam server.
- 6) *429 : Too Many Request*, pengirim mengakses lebih dari limit yang telah ditentukan dari batas waktu tertentu.
- 7) *500 : Internal Server Error*, server atau program dalam resource mengalami kesalahan.

2.10 Application Programming Interface (API)

Application programming interface atau biasa disingkat dengan API merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan *programmer* untuk “membongkar” suatu software, kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan *programmer* menggunakan sistem *function*. Proses ini dikelola melalui sistem operasi. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi (Ramadhani, 2015).

2.11 Node.js

Node.js adalah sebuah *runtime environment opensource* dan *cross platform* yang biasa digunakan untuk mengembangkan aplikasi web, walaupun node.js bukan merupakan *framework* javascript namun banyak modul dasarnya yang ditulis menggunakan javascript. *Runtime environment* yang dimiliki oleh node.js menerjemahkan javascript menggunakan *google V8 javascript engine* (Rismanto dkk., 2016).

2.12 Mongo DB

Mongo DB adalah sistem basis data yang berbasis *Document Oriented Database* dan termasuk sistem basis data yang menganut paham NoSQL. NoSQL singkatan dari *Not Only Structured Query Language*, artinya sebuah sistem basis data tidak hanya harus menggunakan perintah SQL untuk melakukan proses manipulasi data.

Mongo DB tidak memiliki tabel, kolom, dan baris. Pada MongoDB yang ada hanyalah koleksi dan dokumen. Dokumen yang terdapat dalam MongoDB dapat memiliki atribut yang berbeda dengan dokumen lain walaupun berada dalam satu koleksi. Hal ini tidak dapat dilakukan dalam *Relational Database Management System* (RDBMS), dimana sebuah baris dalam tabel tidak mungkin memiliki kolom yang berbeda dengan baris yang lain jika berada dalam satu tabel (Putra and Rahmayeni, 2016).

2.13 Express.js

Express JS merupakan *framework* web berbasis Node JS yang memiliki berbagai fitur-fitur dasar untuk membangun aplikasi web, baik itu *fullstack* maupun *web service*. Fitur-fitur tersebut diantaranya: *routing*, *middleware*, dan *template engine*. Di samping popularitasnya yang tinggi, membuat *framework* ini banyak *resource* yang mendukungnya. Kelebihan lain dari *framework* ini adalah *simple*, *minimalist* dan *to the point*, artinya kita bisa membuat sesuatu dengan kode yang lebih singkat. Hal ini tentu akan memudahkan kita mempelajarinya.

Framework ini cukup solid sehingga menjadi dasar dari banyak *framework* berbasis Node JS lainnya (Azamuddin & Mukhlisin, 2020b).

2.14 React.js

React.js merupakan salah satu pustaka dalam bahasa pemrograman javascript yang digunakan secara spesifik untuk membangun tampilan antar muka pengguna (*user interface*) pada aplikasi berbasis web. Pada umumnya pengembang akan menyebut pustaka ini sebagai React JS, karena teknologi React juga dikembangkan untuk membangun *mobile apps* melalui React Native. Untuk membedakan kedua hal ini maka digunakanlah istilah React JS untuk menyebut pustaka React yang digunakan untuk membangun aplikasi web (Azamuddin & Mukhlisin, 2020a).

2.15 Tailwind CSS

Tailwind CSS atau Tailwind *Cascading Style Sheet* adalah *utility-first* CSS *framework* yang digunakan untuk membuat *User Interface* (UI) atau desain dengan cepat. Maksud dari *utility-first* adalah Tailwind CSS menyediakan banyak *utility class* yang dapat digunakan untuk membuat *custom* desain tanpa harus menulis CSS (Agae, 2020). Tailwind CSS memiliki pendekatan yang berbeda dengan *framework* CSS lainnya seperti *Bootstrap*, *Foundation* atau *Bulma*. Jika *framework* CSS lainnya telah menyediakan komponen yang utuh, tinggal diinisiasikan lalu digunakan, maka pada Tailwind CSS memberikan *class-class* kecil yang bisa digunakan dengan sangat fleksibel. Hal ini digunakan untuk membangun antarmuka yang sesuai keinginan tanpa harus *override styles* seperti di *framework* CSS lainnya (Ihsan, 2020).

2.16 Insomnia Rest Client

Insomnia adalah aplikasi desktop lintas *platform* gratis yang menghilangkan rasa sakit saat berinteraksi dengan API berbasis HTTP. Insomnia menggabungkan antarmuka yang mudah digunakan dengan fungsionalitas lanjutan seperti *authentication helpers*, *code generation*, dan *environment*

variables. Ada juga opsi untuk berlangganan paket berbayar untuk mendapatkan akses ke sinkronisasi data terenkripsi dan kolaborasi tim (Kong Inc., 2020).

2.17 *Scanner Quick Response (QR) Code*

Scanner Quick Response (QR) Code pada umumnya telah tersedia pada berbagai merk *smartphone*, untuk *smartphone* berbasis android umumnya terletak pada *browser* utama. Namun ada *smartphone* android yang memiliki aplikasi *Scanner QR-Code* tersendiri yaitu Xiaomi dan ada juga *smartphone* android yang telah menginstal secara *default* aplikasi Google Lens yang bisa dijadikan alat untuk menerjemahkan *QR-Code*. Pada *smartphone* berbasis iOS (*iPhone Operating System*) telah memiliki aplikasi tersendiri untuk menerjemahkan *QR-Code* yang telah terinstal secara *default*.

2.18 *Library Generate Quick Response (QR) Code*

Library ini bernama *qrcode.react*, yang dikembangkan oleh *Engineer* Facebook bernama Paul O'Shannessy. *Library* ini merupakan komponen ReactJS untuk menghasilkan *Quick Response (QR) Code* dan berlaku *open-source* yang dimana hingga saat ini memiliki 12 kontributor dan lebih dari 13.000 telah digunakan oleh pengembang.

2.19 *Unified Modelling Language (UML)*





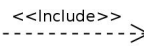
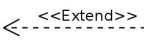
Unified Modeling Language (UML) merupakan bahasa dalam mendesain perangkat lunak secara visual. Dengan UML, desainer dapat melihat konsep global suatu desain. Desain kemudian dapat dijadikan panduan dalam proses pengembangan dan rekayasa perangkat lunak. Selain itu, UML dapat menjadi media komunikasi gagasan antara pengembang perangkat lunak dengan pengguna (Shalahuddin dan Sukamto, 2011). Ada beberapa jenis diagram dalam UML yaitu :

1. *Diagram Use Case*

Diagram Use Case menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi

fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam diagram *activity*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat pengguna) dan bukan bagaimana fungsi yang ada di dalam sistem.



Tabel 2.13 Simbol-simbol Diagram *Use Case*





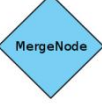


No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Merupakan peran orang, sistem yang lain, atau alat ketika berhubungan dengan <i>use case</i> .
2.		<i>Use Case</i>	Deskripsi dari uraian aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
3.		<i>Association</i>	Abstraksi dari penghubung antara <i>actor</i> dengan <i>use case</i> .
4.		<i>Generalisasi</i>	Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i> .
5.		<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
6.		<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

2. Diagram *Activity*

Diagram *Activity* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana alir berakhir. Diagram *Activity* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Tabel 2.14 Simbol-simbol Diagram *Activity*


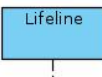


No.	Simbol	Nama	Keterangan
1.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
2.		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan diakhiri.

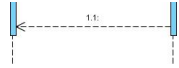


3.		<i>Control Flow</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.
4.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
5.		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
6.		<i>Decision</i>	Percabangan, dimana ada pilihan aktivitas yang lebih dari satu.
7.		<i>Merge</i>	Penggabungan, dimana ada banyak aktivitas digabung menjadi satu.
8.		<i>Join</i>	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
9.		<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

3. Diagram Sequence

Diagram *Sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu objek dan pesan yang dikirimkan dan diterima antar objek. Jumlah diagram *sequence* yang akan digambar adalah minimal sama dengan jumlah *use case* yang didefinisikan.

Tabel 2.15 Simbol-simbol Diagram Sequence

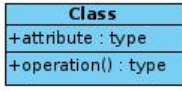




No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Mewakili peran yang dimainkan oleh pengguna (manusia), perangkat keras eksternal, atau subjek lainnya.
2.		<i>Lifeline</i>	Mewakili peserta individu dalam Interaksi.
3.		<i>Activation Box</i>	Berbentuk sebuah persegi panjang tipis pada <i>lifeline</i> , mewakili periode di mana suatu elemen melakukan operasi.
4.		<i>Call Message</i>	Jenis pesan yang mewakili permintaan operasi dari target


			lifeline.
5.		<i>Return Message</i>	Jenis pesan yang mewakili informasi yang dikirimkan kembali ke pengirim pesan atau pemanggil berdasarkan pesan sebelumnya.
6.		<i>Self Message</i>	Jenis pesan yang mewakili permohonan pesan dari lifeline yang sama.
7.		<i>Note</i>	Catatan atau lampiran berbagai komentar ke elemen. Sebuah komentar tidak memiliki kekuatan semantik, tetapi dapat berisi informasi yang berguna bagi pemodel.

4. Diagram Class

Diagram *Class* adalah kumpulan objek-objek yang mempunyai struktur umum, *behavior* umum, relasi umum, dan *semantic* atau kata yang umum. *Class-class* ditentukan atau ditemukan dengan cara memeriksa objek-objek dalam diagram *sequence* dan *collaboration diagram*. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan.

Tabel 2.16 Simbol-simbol Diagram Class

No.	Simbol	Nama	Keterangan
1.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
3.		<i>Directed Association</i>	Relasi antarkelas dengan makna kelas yang atau digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		<i>Generalisasi</i>	Relasi antarkelas dengan makna generalisasi - spesialisasi (umum khusus).
5.		<i>Dependency</i>	Relasi antarkelas dengan makna ketergantungan antarkelas.

6.		<i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)
----	---	--------------------	---

2.20 Metode *Prototype*

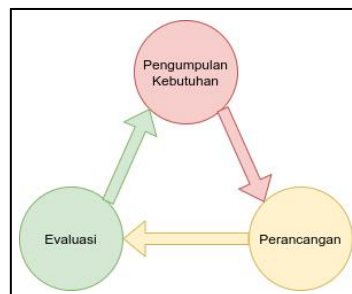
Metode *Prototype* merupakan metode pengembangan perangkat lunak yang terstruktur dan memiliki beberapa tahapan yang harus dilalui pada pengembangannya. Namun jika tahap final dinyatakan bahwa sistem yang telah dibuat belum sempurna atau masih memiliki kekurangan, maka sistem akan dievaluasi kembali dan akan melalui proses dari awal. Pendekatan *prototyping* adalah proses *iterative* yang melibatkan hubungan kerja yang dekat antara perancang dan pengguna (Lugina, 2015).

1. Tujuan

Tujuan metode *prototype* ini adalah mengembangkan model awal software menjadi sebuah sistem yang final.

2. Proses

Pada metode *prototype* ini terdiri dari 3 proses, dapat dilihat pada Gambar 2.10 berikut :



Gambar 2.10 Metode *Prototype*

Adapun penjelasan proses dalam metode *prototyping* adalah sebagai berikut :

a. Pengumpulan Kebutuhan

Pengembang (*developer*) dan pengguna (*user*) akan bertemu terlebih dahulu dan kemudian menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.

b. Perancangan

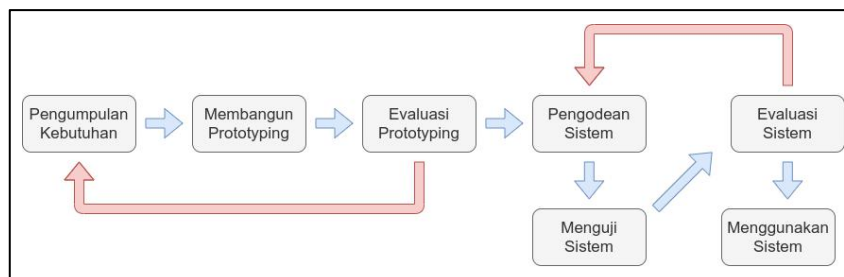
Perancangan dilakukan dengan cepat dan rancangan tersebut mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.

c. Evaluasi

Pada proses ini klien (*user*) akan mengevaluasi *prototyping* yang dibuat untuk memperjelas kebutuhan software.

3. Tahapan

Selain itu untuk memodelkan sebuah perangkat lunak dibutuhkan beberapa tahapan dalam proses pengembangannya, tahapan inilah yang akan menentukan keberhasilan dari sebuah *software* itu sendiri. Tahapan-tahapan pada metode *prototype* adalah sebagai berikut :



Gambar 2.11 Tahapan Metode *Prototype*

a. Pengumpulan Kebutuhan

Pada tahap ini, *developer* dan *user* bersama-sama mendefinisikan format dan kebutuhan keseluruhan perangkat lunak, mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat.

b. Membangun *Prototyping*

Pada tahap ini, *developer* dan *user* bersama-sama membuat format *input* maupun output yang akan dihasilkan oleh sistem yang akan dibuat.

c. Evaluasi *Prototyping*

Pada tahap ini, *developer* dan *user* bersama-sama mendefinisikan format dan kebutuhan keseluruhan perangkat lunak, mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat.

d. Pengodean Sistem

Pada tahap ini, *prototyping* yang telah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

e. Menguji Sistem

Pada tahap ini, pengodean yang telah dilakukan sebelumnya akan diuji apakah dapat berjalan dengan baik atau masih ada bagian-bagian yang perlu diperbaiki dan apakah masih ada bagian yang belum sesuai dengan konsep sebelumnya.

f. Evaluasi Sistem

Evaluasi sistem tidak sama dengan evaluasi *prototyping*, evaluasi sistem ini adalah mengevaluasi sistem atau perangkat lunak yang telah jadi apakah telah sesuai dengan keinginan *user* atau kah belum. Ketika sistem atau perangkat lunak belum sesuai maka sistem akan direvisi dan kembali ke tahap 4 dan 5. Ketika telah sesuai maka sistem atau perangkat lunak dilanjutkan ke tahapan selanjutnya.

g. Menggunakan Sistem

Tahap ini merupakan tahap akhir, yang dimana pada tahapan ini perangkat lunak yang telah jadi dan telah lulus uji, dinyatakan siap untuk digunakan oleh *user*.

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Kajian Pustaka

Kajian pustaka adalah metode pengumpulan data yang dilakukan dengan cara melakukan pengumpulan data pendukung penelitian yang akan dijadikan referensi. Data dapat berupa buku, *paper*, jurnal, skripsi dan sebagainya.

2. Wawancara

Wawancara merupakan percakapan yang dilakukan antara dua orang atau lebih oleh pewawancara kepada narasumber. Tujuan dari metode ini untuk mendapatkan informasi terkait permasalahan pemesanan dan pengantaran makanan serta data yang akan dimasukkan kedalam aplikasi “**RANCANG BANGUN APLIKASI *VIRTUAL FOOD ORDERING* DENGAN *QUICK RESPONSE (QR) CODE* MENGGUNAKAN ALGORITMA *MULTILEVEL FEEDBACK QUEUE (MLFQ)* BERBASIS *PROGRESSIVE WEB APPLICATION (PWA)*”**. Narasumber dari wawancara ini yaitu Bapak Adil selaku Manager di Rumah Makan Kampung Bakau, Kendari.

3.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada perancangan ini adalah metode *prototype*, yang memiliki tujuh tahap dalam pengembangan sistem, yaitu :

1. Pengumpulan Kebutuhan

Pada tahap ini, dilakukan proses identifikasian terhadap sistem yang akan dibuat, mendiskusikan permasalahan pada tempat penelitian, menentukan algoritma yang dapat diterapkan ke dalam sistem dan memuat pemodelan proses (diagram *use case*, diagram *activity*, diagram *class* dan diagram *sequence*).

2. Membangun *Prototyping*

Pada tahap ini, dilakukan pembuatan *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat *input* dan format *output* dan desain rancangan antarmuka).

3. Evaluasi *Prototyping*

Pada tahap ini, dilakukan evaluasi pada *prototyping* yang dibuat pada tahap sebelumnya dan mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat. evaluasi ini dilakukan untuk mengetahui apakah *prototyping* sudah sesuai dengan harapan pelanggan.

4. Pengodean Sistem

Pada tahap ini, dilakukan pembangunan sistem (*development stage*) atau pengimplementasian kode berdasarkan perancangan yang telah dibuat sebelumnya, sehingga menjadi sistem yang dapat digunakan.

5. Menguji Sistem

Pada tahap ini, dilakukan pengujian terhadap sistem yang telah dibangun apakah dapat berjalan dengan baik atau masih ada bagian-bagian yang perlu diperbaiki atau ditambahkan.

6. Evaluasi Sistem

Pada tahap ini, dilakukan evaluasi pada sistem yang telah diuji apakah telah sesuai keinginan pengguna atau belum. Ketika sistem belum sesuai maka sistem akan direvisi dan kembali ke pengodean sistem. Ketika telah sesuai maka sistem dilanjutkan ke tahap produksi (*production stage*) dan di lanjutkan ke tahapan akhir yaitu menggunakan sistem.

7. Menggunakan Sistem

Tahap ini merupakan tahap akhir, yang dimana pada tahapan ini perangkat lunak yang telah jadi dan telah lulus uji, dinyatakan siap untuk digunakan oleh pengguna.

3.3 Waktu dan Tempat Penelitian

3.3.1 Waktu Penelitian

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Januari 2021 sampai dengan Mei 2021, dengan rincian kegiatan dapat dilihat pada **Tabel 3.1** berikut :

Tabel 3.1 Gannt Chart Waktu Penelitian

No.	Uraian	Waktu																			
		2021																			
		Januari				Februari				Maret				April				Mei			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Pengumpulan Kebutuhan	■	■	■	■	■	■														
2.	Membangun <i>Prototyping</i>					■	■	■													
3.	Evaluasi <i>Prototyping</i>							■	■												
4.	Pengodean Sistem									■	■	■	■	■	■	■					
5.	Menguji Sistem															■	■				
6.	Evaluasi Sistem																	■	■		
7.	Menggunakan Sistem																				■

3.3.2 Tempat Penelitian

Penelitian tugas akhir bertempat di Rumah Makan Kampung Bakau, Jl. Buburanda, Anduonohu, Kec. Kambu, Kota Kendari, Sulawesi Tenggara 93231.

3.4 Analisis Kebutuhan Sistem

3.4.1 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah sebuah data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur. Pada analisis kebutuhan fungsional terdapat beberapa kebutuhan-kebutuhan untuk membangun sistem yaitu analisis kebutuhan *input*, analisis kebutuhan proses dan analisis kebutuhan *output*. Adapun kebutuhan-kebutuhan tersebut adalah sebagai berikut :

1. Analisis Kebutuhan *Input*

Adapun analisis kebutuhan input pada proses analisis sistem ini adalah *QR-Code* yang berada disetiap meja makan dan data pesanan yang telah di-assign oleh pelanggan.

2. Analisis Kebutuhan Proses

Adapun analisis kebutuhan proses pada proses analisis sistem ini bertujuan mengidentifikasi permasalahan pada sistem, dimana aplikasi dibangun meliputi perangkat lunak dan analisis terhadap sistem.

3. Analisis Kebutuhan *Output*

Adapun analisis kebutuhan *output* pada proses analisis sistem ini adalah koki yang bertugas pada sisi dapur yang akan memproses data pesanan yang telah dipesan oleh pelanggan.

3.4.2 Analisis Kebutuhan *Non-Fungsional*

Analisis kebutuhan *non-fungsional* bertujuan untuk menganalisis sumber daya yang dibutuhkan untuk membangun sistem. Analisis kebutuhan *non-fungsional* dibagi menjadi dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak yang bertujuan untuk merincikan komponen-komponen yang dibutuhkan dalam membangun sistem. Pada analisis kebutuhan *non-fungsional* terbagi menjadi 2 yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak. Adapun penjelasannya adalah sebagai berikut :

1. Analisis Kebutuhan Perangkat Keras

Adapun perangkat keras yang dibutuhkan untuk membangun aplikasi *Virtual Food Ordering* dapat dilihat pada Tabel 3.2 berikut.

Tabel 3.2 Spesifikasi Perangkat Keras

No.	Nama Perangkat	Spesifikasi
1.	<i>Personal Computer (PC)</i>	HP 110 Desktop PC Series
2.	<i>Processor</i>	Intel® Core™ i3-3240T CPU @ 2.90GHz × 4
3.	<i>Graphics</i>	Intel® HD Graphics 2500 (IVB GT1)
4.	<i>Storage</i>	1 TB HDD
5.	<i>Random Access Memory (RAM)</i>	8192 MB
6.	<i>Monitor</i>	HP W2072a (20 Inch)

2. Analisis Kebutuhan Perangkat Lunak

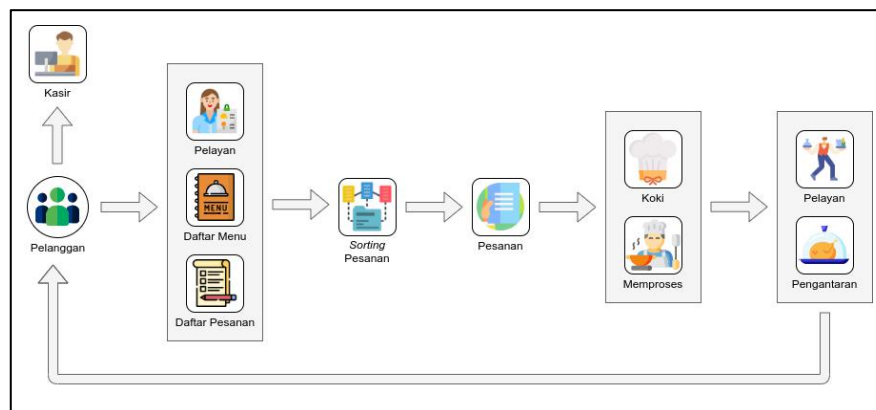
Adapun perangkat lunak yang dibutuhkan untuk membangun aplikasi *Virtual Food Ordering* dapat dilihat pada Tabel 3.3 berikut.

Tabel 3.3 Spesifikasi Perangkat Lunak

No.	Nama Perangkat	Fungsi	Spesifikasi
1.	Linux	<i>Operating System</i>	Ubuntu 20.04.1 LTS
2.	Mongo DB	<i>Database</i>	db version v3.6.8
3.	Node.js	<i>Runtime Environment</i>	v14.15.4
4.	Express.js	<i>Backend Stack</i>	v4.17.1
5.	React.js	<i>Frontend Stack</i>	v17.0.1
6.	Chrome	<i>Web Browser</i>	Version 87.0.4280.88
7.	Visual Studio Code	<i>Code Editor</i>	Version 1.52.1
8.	Insomnia Rest Client	<i>API Rest Client</i>	Version 2020.5.2

3.5 Analisis Perancangan Sistem

3.5.1 Analisis Sistem Yang Sedang Berjalan



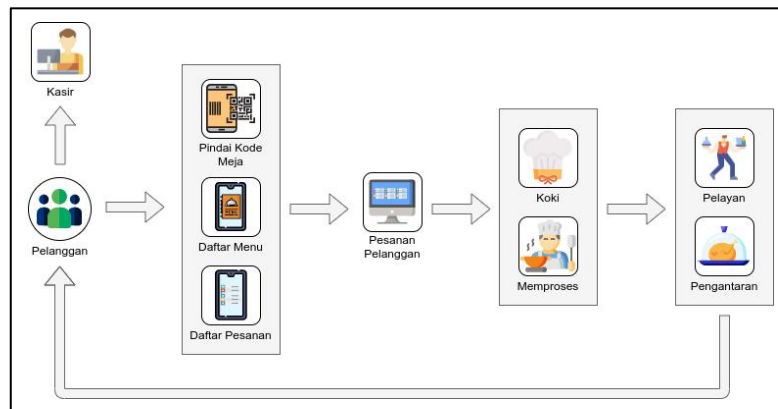
Gambar 3.1 Sistem Yang Sedang Berjalan

Pada Gambar 3.1 mengilustrasikan sistem yang sedang berjalan sekarang, penjelasan lebih lanjut adalah sebagai berikut :

- 1) Pelanggan (*customer*) datang dan dilayani oleh pelayan dengan memberikan daftar menu dan pelayan mencatat pesanan dari pelanggan.
- 2) Pelayan memberikan pesanan ke dapur untuk diproses, tetapi sebelum diproses pesanan tersebut disortir terlebih dahulu. Ini berguna untuk pembagian menu-menu pesanan berdasarkan *section* dari koki.
- 3) Koki memproses pesanan.

- 4) Setelah pesanan selesai diproses, maka pelayan akan mengantarkan pesanan ke pelanggan.
- 5) Terakhir, pelanggan melakukan pembayaran ke kasir.

3.5.2 Analisis Sistem Yang Akan Dikembangkan



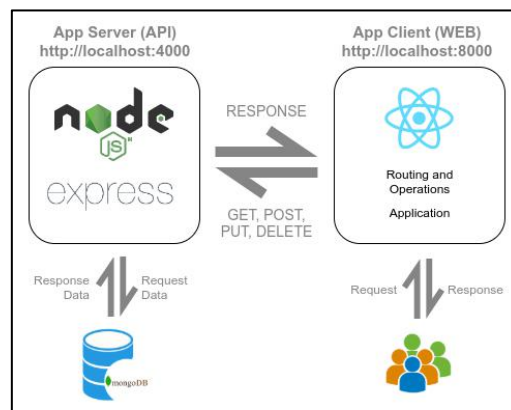
Gambar 3.2 Sistem Yang Akan Dikembangkan

Pada Gambar 3.2 mengilustrasikan sistem yang akan dikembangkan, penjelasan lebih lanjut adalah sebagai berikut :

- 1) Pelanggan (*customer*) datang, kemudian memindai (*scanning*) kode yang berada pada meja.
- 2) Setelah *scanning*, menu makanan dan minuman ditampilkan dan pelanggan melihat dan memilih daftar menu, kemudian pesan.
- 3) Pesanan pelanggan akan terlihat oleh koki yang akan memproses pesanan tersebut.
- 4) Koki memproses pesanan.
- 5) Setelah pesanan selesai diproses, maka pelayan akan mengantarkan pesanan ke pelanggan.
- 6) Terakhir, pelanggan melakukan pembayaran ke kasir.

3.5.3 Perancangan Arsitektur Sistem

Sistem ini dirancang menggunakan arsitektur *RESTful API*, dimana implementasinya dilakukan secara terpisah antara *app server* dengan *app client* sehingga masing-masing dapat memiliki alamat server yang berbeda. Adapun jika alamat servernya sama pasti nomor *port*-nya yang akan berbeda. Arsitektur sistem dapat dilihat pada Gambar 3.3.



Gambar 3.3 Arsitektur Sistem

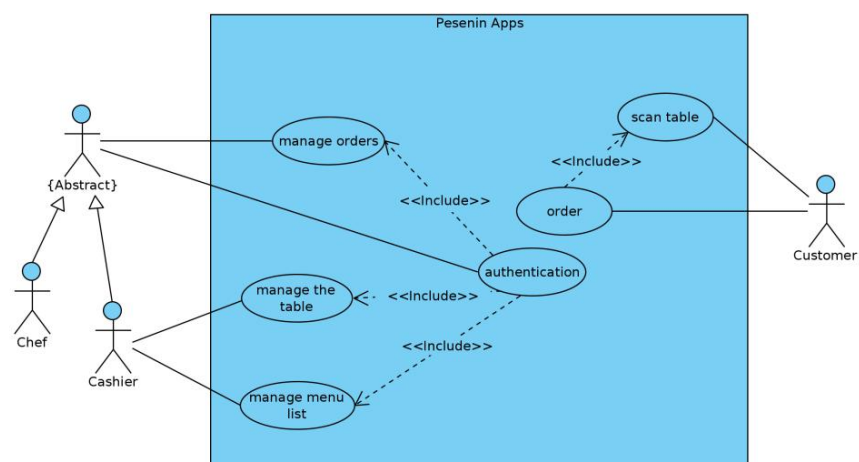
Arsitektur *RESTful API* ini didesain dari *client* dan *server* yang berkomunikasi yang menggunakan protokol *Hypertext Transfer Protocol* (HTTP) untuk pertukaran data. Dengan ini *app server* sebagai penyedia jalur akses *resource* atau data, dan *app client* sebagai *consume* (menampilkan dan mengelola) *resource* yang di sediakan oleh *app server*.

3.5.4 Pemodelan *Unified Modeling Language* (UML)

Pada pemodelan sistem ini, menggunakan pemodelan proses *Unified Modeling Language* (UML) meliputi diagram *use case*, diagram *activity*, diagram *class* dan diagram *sequence*.

1. Diagram *Use Case*

Diagram *use case* digunakan untuk memodelkan fungsionalitas sistem yang dilihat dari pengguna yang ada diluar sistem (aktor). Berikut adalah diagram *use case* untuk sistem yang akan dibangun.



Gambar 3.4 Diagram *Use Case* Sistem

Adapun deskripsi dari masing-masing *use case* yang ada pada Gambar 3.4 adalah sebagai berikut :

a) *Use Case Scan Table*

Adapun deskripsi dari *use case scan table* adalah sebagai berikut :

Tabel 3.4 Deskripsi Use Case Scan Table

Nama	<i>Scan Table</i>
Deskripsi singkat	<i>Customer</i> memindai (<i>scan</i>) <i>QR-Code</i> pada meja makan yang tersedia.
Prasyarat	<i>Customer</i> tersebut berwenang untuk memindai (<i>scan</i>) pada meja makan.
Kondisi akhir	<i>QR-Code</i> meja makan terpindai (<i>ter-scan</i>) dan daftar menu makanan dan minuman ditampilkan.
Situasi <i>error</i>	Memindai (<i>scan</i>) dengan perangkat (<i>device</i>) yang berbeda dengan sebelumnya.
Status sistem saat terjadi kesalahan	<i>Customer</i> belum memindai (<i>scan</i>) <i>QR-Code</i> meja makan.
Aktor	<i>Customer</i>
Pemicu	<i>Customer</i> yang ingin melakukan pemesanan.
Proses standar	(1) <i>Customer</i> memilih meja makan. (2) <i>Customer</i> memindai <i>QR-Code</i> meja. (3) Menu ditampilkan.
Proses alternatif	(1') Meja makan penuh

b) *Use Case Order*

Adapun deskripsi dari *use case order* adalah sebagai berikut :

Tabel 3.5 Deskripsi Use Case Order

Nama	<i>Order</i>
Deskripsi singkat	<i>Customer</i> melihat menu makanan dan minuman dan melakukan pemesanan.
Prasyarat	<i>Customer</i> tersebut berwenang untuk memindai (<i>scan</i>) pada meja makan.
Kondisi akhir	Makanan dan minuman dipesan
Situasi <i>error</i>	Melihat dan melakukan pesanan tanpa memindai (<i>scan</i>) <i>QR-Code</i> meja makan.
Status sistem saat terjadi kesalahan	<i>Customer</i> belum memindai (<i>scan</i>) <i>QR-Code</i> meja makan.
Aktor	<i>Customer</i>

Pemicu	<i>Customer</i> telah memindai (<i>scan</i>) <i>QR-Code</i> meja.
Proses standar	(1) <i>Customer</i> memilih makanan dan minuman. (2) <i>Customer</i> melakukan pesanan. (3) <i>Customer</i> melakukan penambahan pesanan.
Proses alternatif	(2') Pesanan tidak dapat dibatalkan

c) *Use Case Authentication*

Adapun deskripsi dari *use case authentication* adalah sebagai berikut :

Tabel 3.6 Deskripsi Use Case Authentication

Nama	<i>Authentication</i>
Deskripsi singkat	<i>Cashier</i> dan <i>Chef</i> harus melakukan autentikasi (<i>login</i>) terlebih dahulu, sebelum masuk ke proses selanjutnya.
Prasyarat	<i>Customer</i> tersebut berwenang untuk memindai (<i>scan</i>) pada meja makan.
Kondisi akhir	Masuk kedalam <i>dashboard</i>
Situasi <i>error</i>	Akun <i>Cashier</i> atau <i>Chef</i> tidak valid.
Status sistem saat terjadi kesalahan	<i>Username</i> dan <i>password</i> tidak valid, mohon koreksi kembali.
Aktor	<i>Cashier</i> , <i>Chef</i>
Pemicu	<i>Cashier</i> atau <i>Chef</i> yang ingin mengelola menu <i>dashboard</i> (<i>order</i> , <i>table</i> , <i>menu list</i>)
Proses standar	(1) <i>Cashier</i> atau <i>Chef</i> memasukkan <i>username</i> dan <i>password</i> . (2) <i>Cashier</i> atau <i>Chef</i> melakukan pengelolaan pada menu pemesanan (<i>order</i>). (3) <i>Cashier</i> melakukan pengelolaan daftar makanan dan minuman (<i>menu list</i>). (4) <i>Cashier</i> melakukan pengelolaan meja makan (<i>table</i>).
Proses alternatif	(1') <i>Cashier</i> atau <i>Chef</i> tidak memiliki <i>username</i> dan <i>password</i> .

d) *Use Case Manage Orders*

Adapun deskripsi dari *use case manage orders* adalah sebagai berikut :

Tabel 3.7 Deskripsi Use Case Manage Orders

Nama	<i>Manage Order</i>
Deskripsi singkat	<i>Cashier</i> dan <i>Chef</i> mengelola data pesanan (<i>manage order</i>).
Prasyarat	<i>Cashier</i> dan <i>Chef</i> berhasil masuk ke <i>dashboard</i> atau telah melakukan <i>authentication</i> .
Kondisi akhir	Mengelola data pesanan (<i>manage order</i>)
Situasi <i>error</i>	Akun <i>Cashier</i> atau <i>Chef</i> tidak valid.
Status sistem saat terjadi kesalahan	<i>Username</i> dan <i>password</i> tidak valid, mohon koreksi kembali.
Aktor	<i>Cashier</i> , <i>Chef</i>
Pemicu	<i>Cashier</i> atau <i>Chef</i> yang ingin mengelola pesanan (<i>manage order</i>)
Proses standar	(1) <i>Chef</i> menerima pesanan. (2) <i>Chef</i> mengelola pesanan. (3) <i>Chef</i> meneruskan data pesanan ke <i>Cashier</i> . (4) <i>Cashier</i> mengelola data pesanan.
Proses alternatif	-

e) *Use Case Manage The Table*

Adapun deskripsi dari *use case manage the table* adalah sebagai berikut :

Tabel 3.8 Deskripsi Use Case Manage The Table

Nama	<i>Manage The Table</i>
Deskripsi singkat	<i>Cashier</i> mengelola data meja (<i>table</i>).
Prasyarat	<i>Cashier</i> berhasil masuk ke <i>dashboard</i> .
Kondisi akhir	Mengelola data meja (<i>table</i>)
Situasi <i>error</i>	Akun <i>Cashier</i> tidak valid.
Status sistem saat terjadi kesalahan	<i>Username</i> dan <i>password</i> tidak valid, mohon koreksi kembali.
Aktor	<i>Cashier</i>
Pemicu	<i>Cashier</i> yang ingin mengelola data meja (<i>table</i>)
Proses standar	(1) <i>Cashier</i> menambah meja. (2) <i>Cashier</i> cetak <i>QR-Code</i> meja.
Proses alternatif	-

f) *Use Case Manage Menu List*

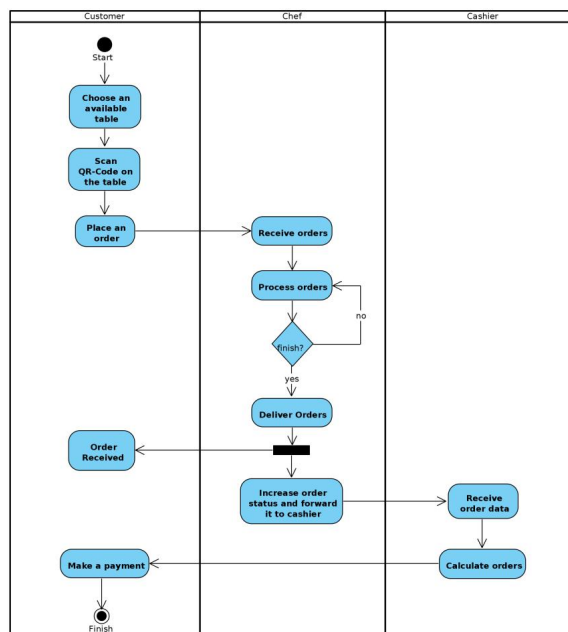
Adapun deskripsi dari *use case manage menu list* adalah sebagai berikut :

Tabel 3.9 Deskripsi *Use Case Manage Menu List*

Nama	<i>Manage Menu List</i>
Deskripsi singkat	<i>Cashier</i> mengelola data menu makanan dan minuman (<i>menu list</i>).
Prasyarat	<i>Cashier</i> berhasil masuk ke <i>dashboard</i> .
Kondisi akhir	Mengelola data menu makanan dan minuman (<i>menu list</i>)
Situasi <i>error</i>	Akun <i>Cashier</i> tidak valid.
Status sistem saat terjadi kesalahan	<i>Username</i> dan <i>password</i> tidak valid, mohon koreksi kembali.
Aktor	<i>Cashier</i>
Pemicu	<i>Cashier</i> yang ingin mengelola data menu makanan dan minuman (<i>menu list</i>)
Proses standar	<i>Cashier</i> mengelola data (tambah, ubah, hapus) menu makanan dan minuman.
Proses alternatif	-

2. *Diagram Activity*

Pada diagram *activity* sistem menjelaskan gambaran inti sistem yang akan dibuat. Diagram *activity* sistem dapat dilihat pada Gambar 3.5 berikut.



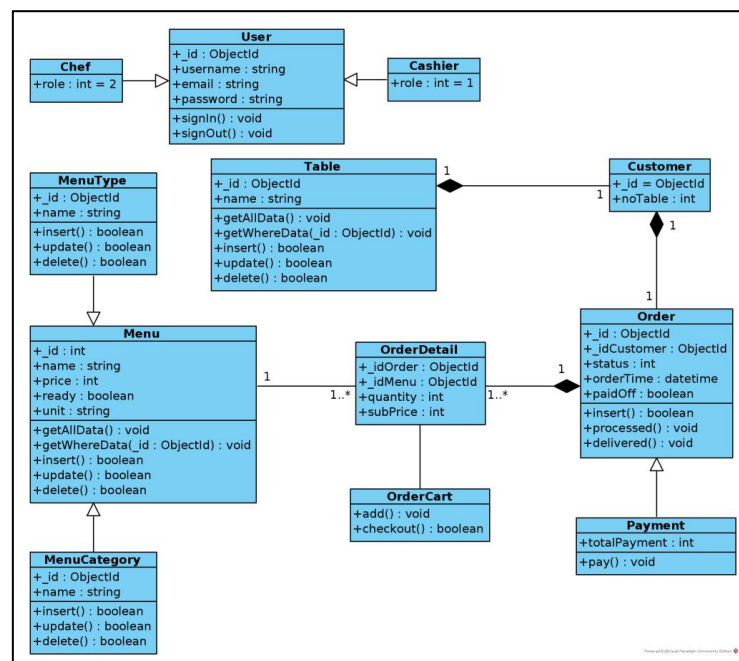
Gambar 3.5 Diagram *Activity* Sistem

Adapun alur dari diagram *activity* ini adalah sebagai berikut :

- a) *Customer* memilih meja makan yang tersedia, kemudian memindai *QR-Code* pada meja tersebut dan melakukan pemesanan.
- b) *Chef* menerima pesanan kemudian mengelolanya (memproses), ketika pesanan tersebut telah selesai maka akan dilanjutkan ke pengantaran ke meja *Customer*. Ketika belum selesai maka akan kembali ke *activity process orders* untuk dikelola.
- c). Bersamaan dengan pengantaran pesanan, data pesanan diteruskan ke *Cashier* untuk dikalkulasi/dihitung agar dapat mengetahui total pembayaran *Customer* diikuti oleh *Customer* yang akan melakukan pembayaran.

3. Diagram Class

Diagram *class* menjelaskan hubungan antara *class* dalam sistem yang dapat dilihat pada Gambar 3.6 berikut.



Gambar 3.6 Diagram Class Sistem

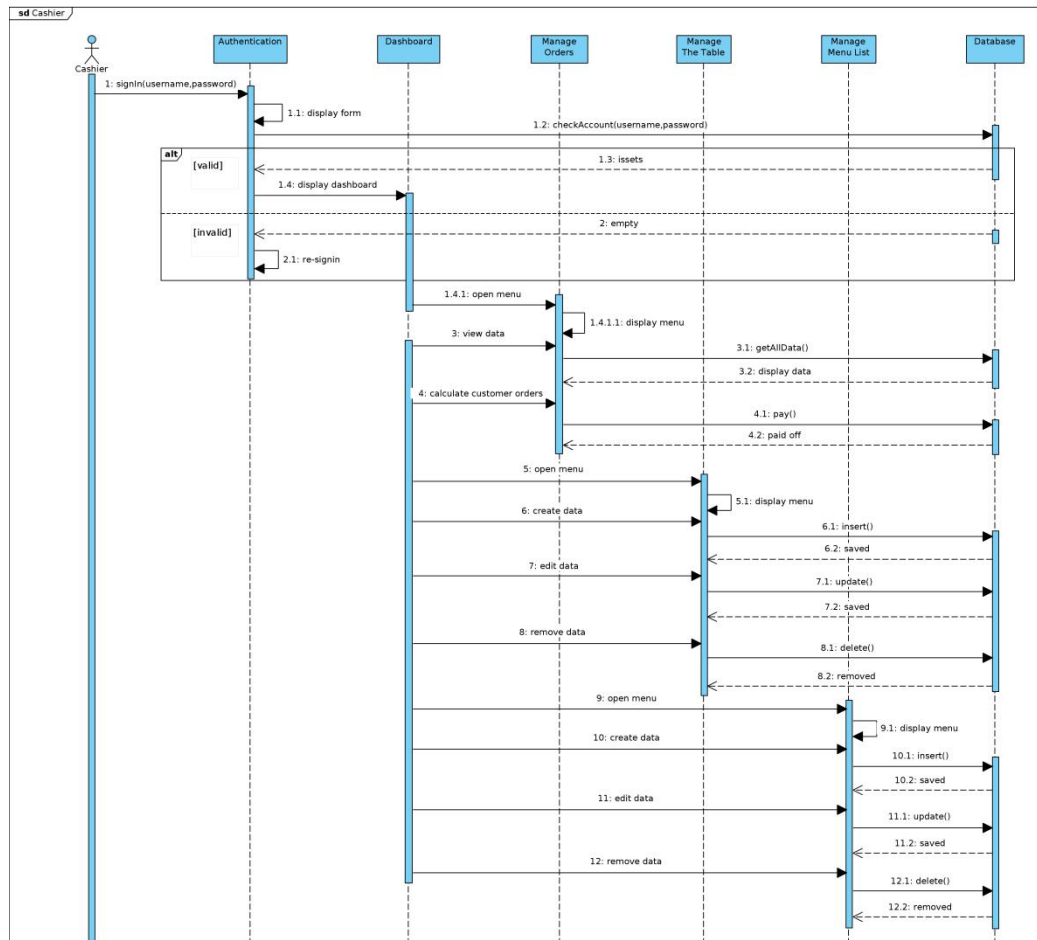
4. Diagram Sequence

Pada diagram *sequence* menggambarkan interaksi antara objek didalam dan disekitar sistem berupa *message* yang digambarkan terhadap waktu. Diagram *sequence* bisa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *even* untuk

menghasilkan *output* tertentu. Adapun diagram *sequence* dari aplikasi yang dibuat adalah sebagai berikut :

a) Diagram *Sequence Cashier*

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequence cashier* yang ditunjukkan oleh Gambar 3.7 berikut.



Gambar 3.7 Diagram *Sequence Cashier* Sistem

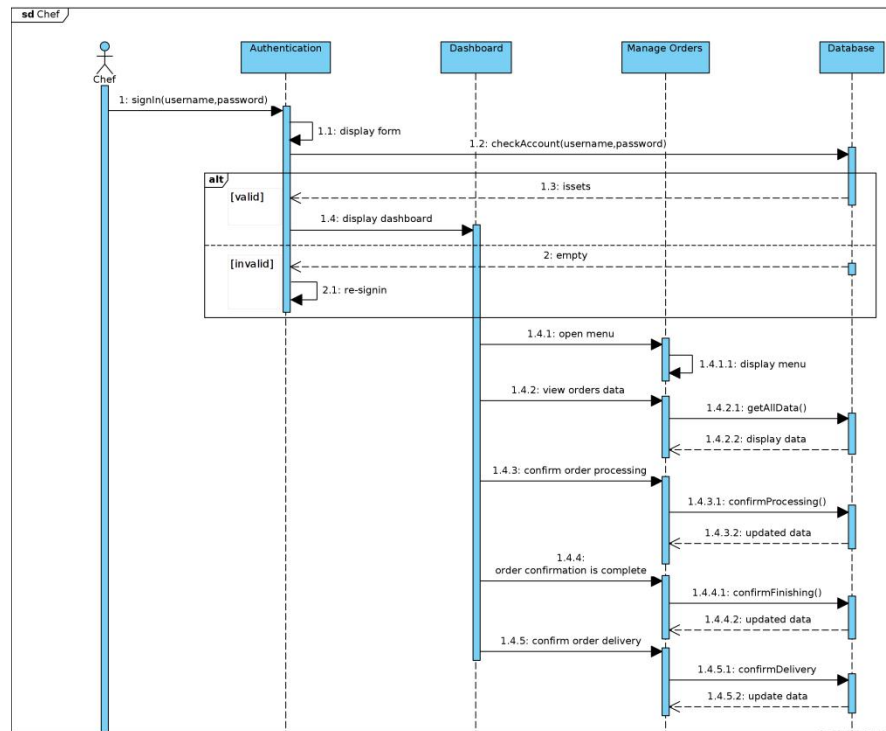
Adapun penjelasan alur kerja *sequence cashier* adalah sebagai berikut :

- 1) *Cashier* mengakses sistem dengan syarat melakukan *authentication* dengan memasukkan *username* dan *password*.
- 2) Ketika *username* dan *password* sah (*valid*) maka *cashier* dapat masuk ke halaman *dashboard* dan dapat mengelola menu-menu yang ada (pesanan, meja dan daftar menu). Ketika *username* dan *password* tidak sah (*invalid*)

maka akan kembali ke halaman *form sign in* dan tidak dapat masuk ke halaman *dashboard*.

b) Diagram *Sequence Chef*

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequence chef* yang ditunjukkan oleh Gambar 3.8 berikut.



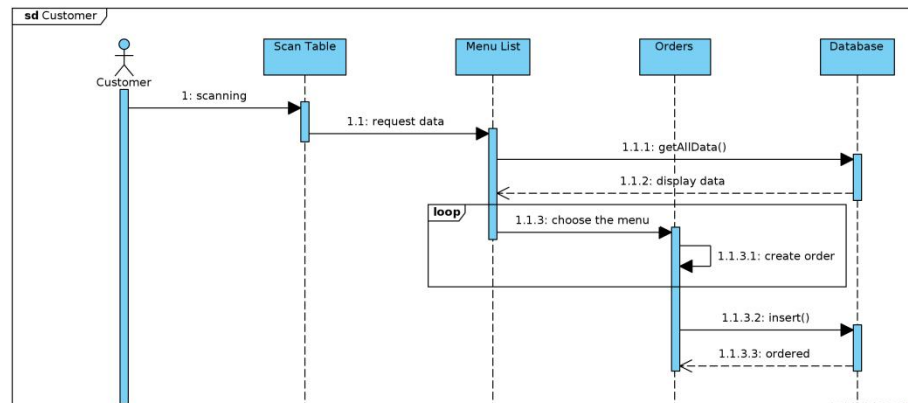
Gambar 3.8 Diagram *Sequence Chef* Sistem

Adapun penjelasan alur kerja *sequence chef* adalah sebagai berikut :

- 1) *Chef* mengakses sistem dengan syarat melakukan *authentication* dengan memasukkan *username* dan *password*.
- 2) Ketika *username* dan *password* sah (*valid*) maka *chef* dapat masuk ke halaman *dashboard* dan dapat mengelola menu pesanan diantaranya konfirmasi proses pesanan, konfirmasi pesanan selesai dan konfirmasi pesanan dalam pengantaran. Ketika *username* dan *password* tidak sah (*invalid*) maka akan kembali ke halaman *form sign in* dan tidak dapat masuk ke halaman *dashboard*.

c) Diagram *Sequence Customer*

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequence customer* yang ditunjukkan oleh Gambar 3.9 berikut.



Gambar 3.9 Diagram *Sequence Customer* Sistem

Adapun penjelasan alur kerja *sequence customer* adalah sebagai berikut :

- 1) *Customer* dapat mengakses sistem ini tanpa harus melakukan *authentication* tetapi dengan syarat harus memindai (*scan*) kode meja.
- 2) Daftar menu ditampilkan.
- 3) *Customer* memilih menu dan melakukan pemesanan.

3.5.4 Perancangan Antarmuka

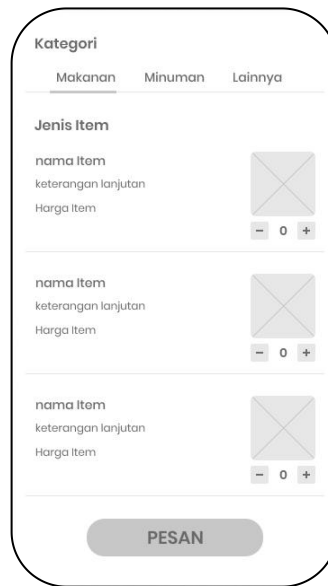
Perancangan antarmuka atau yang lebih sering dikenal dengan istilah *design interface* adalah desain untuk komputer, peralatan, mesin, perangkat komunikasi *mobile*, aplikasi perangkat lunak, dan situs web yang berfokus pada pengalaman dan interaksi pengguna. Adapun perancangan antarmuka dari sistem yang akan dibuat adalah sebagai berikut :

1. *Customer*

Adapun perancangan antarmuka dari sisi *customer* adalah sebagai berikut :

a) Daftar Menu

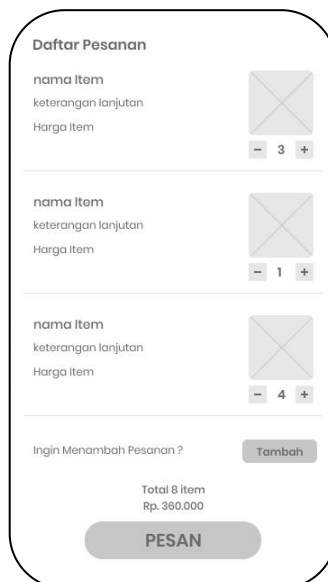
Setelah *customer* telah memindai kode meja, maka akan tampil menu-menu makanan, minuman dan lain-lain. Adapun rancangan antarmuka dapat dilihat pada Gambar 3.10 berikut.



Gambar 3.10 Tampilan Daftar Menu

b) Daftar Pesanan

Setelah melihat daftar menu, *customer* memilih sekaligus menambahkan ke daftar pesanan dan dilanjutkan melakukan pemesanan. Adapun rancangan antarmuka dapat dilihat pada Gambar 3.11 dan Gambar 3.12 berikut.



Gambar 3.11 Tampilan Daftar Pesanan (a)

Pesanan kamu sedang di proses, Mohon di tunggu ya...

Daftar Pesanan

nama Item	keterangan lanjutan	Harga Item
		3
		1

Detail Transaksi

Nama item 1	260.000
Nama item 2	260.000
Nama item 3	260.000
Total	Rp. 260.000

Ingin Menambah Pesanan ? Tambah

Gambar 3.12 Tampilan Daftar Pesanan (b)

2. Chef

Pada perancangan antarmuka dari sisi *chef* menampilkan menu pesanan pelanggan berdasarkan *section chef* yang ada, dan juga menampilkan total pesanan, pesanan diproses, pesanan selesai. Pada tampilan ini juga *chef* dapat mengonfirmasi pesanan tersebut telah selesai dengan menekan *checkbox* yang ada disisi kiri menu pesanan. Adapun rancangan antarmuka dapat dilihat pada Gambar 3.13 berikut.

DAPUR

KASIR

Admin
Kamping Baku

Total pesanan

38

Pesanan Diproses

16

Pesanan Selesai

22

SECTION PENGGORENGAN 08 ANTRIAN

- ☐ 01 ikan karapu (meja 02)
- ☐ 01 ikan putih (meja 03)
- ☐ 01 ikan karapu (meja 04)
- ☐ 01 bebek goreng (meja 02)
- ☐ 01 ayam goreng (meja 06)
- ☐ 01 ikan karapu (meja 04)
- ☐ 01 bebek goreng (meja 02)
- ☐ 01 ayam goreng (meja 06)

SECTION PEMBAKARAN 10 ANTRIAN

- ☐ 01 ikan karapu (meja 02)
- ☐ 01 ikan putih (meja 03)
- ☐ 01 ikan karapu (meja 04)
- ☐ 01 bebek bakar (meja 02)
- ☐ 01 ayam bakar (meja 06)
- ☐ 01 ikan karapu (meja 04)
- ☐ 01 bebek bakar (meja 02)
- ☐ 01 ayam bakar (meja 06)
- ☐ 01 ikan karapu (meja 02)
- ☐ 01 ikan putih (meja 03)
- ☐ 01 ikan karapu (meja 05)
- ☐ 01 bebek bakar (meja 07)
- ☐ 01 ayam bakar (meja 10)
- ☐ 01 ikan karapu (meja 012)
- ☐ 01 bebek bakar (meja 13)
- ☐ 01 ayam bakar (meja 14)
- ☐ 01 bebek bakar (meja 15)
- ☐ 01 ayam bakar (meja 20)

SECTION REBUSAN 28 ANTRIAN

- ☐ 01 ikan palumara (meja 02)
- ☐ 01 ikan palumara (meja 03)
- ☐ 01 ikan palumara (meja 04)
- ☐ 01 ikan palumara (meja 02)
- ☐ 01 sup ayam (meja 06)
- ☐ 01 sup ayam (meja 04)
- ☐ 01 sup ayam (meja 02)
- ☐ 01 ikan palumara (meja 06)

SECTION MINUMAN 10 ANTRIAN

- ☐ 01 es jeruk (meja 02)
- ☐ 01 jus alpukat (meja 03)
- ☐ 01 es teh (meja 04)
- ☐ 01 jus sirsak (meja 02)
- ☐ 01 jus apel (meja 06)
- ☐ 01 es kopi susu (meja 04)

SECTION LAINNYA 10 ANTRIAN

- ☐ 01 nasi bakul (meja 02)
- ☐ 01 nasi bakul (meja 03)
- ☐ 01 nasi (meja 04)
- ☐ 01 nasi (meja 02)
- ☐ 01 air mineral (meja 06)
- ☐ 01 telur rebus (meja 04)

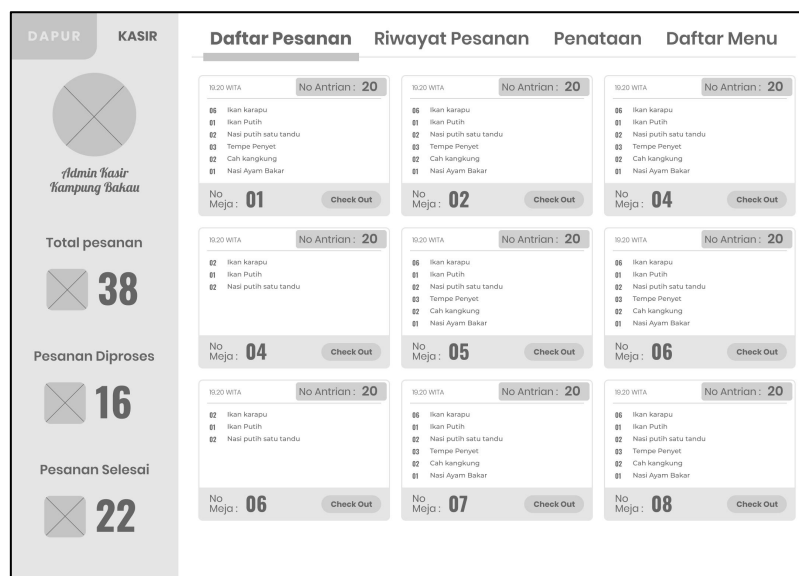
Gambar 3.13 Tampilan Dashboard Chef

3. *Cashier*

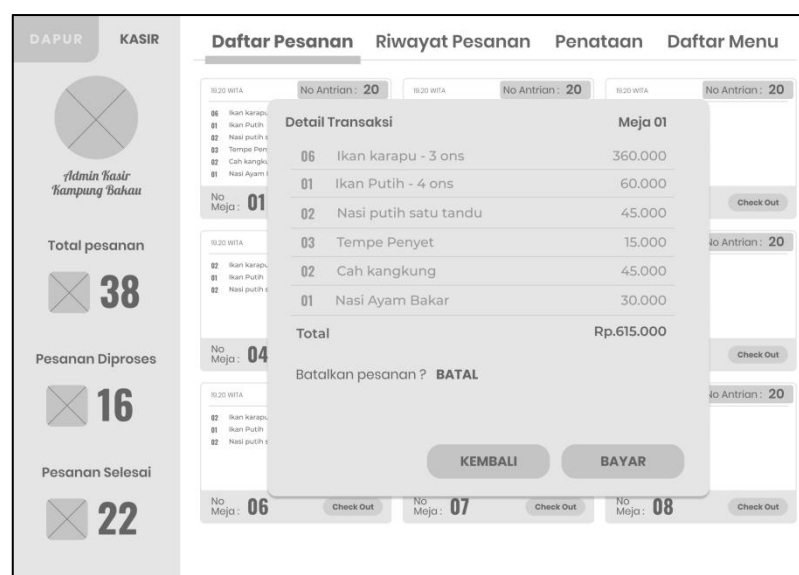
Adapun perancangan antarmuka dari sisi *cashier* adalah sebagai berikut :

a) Daftar Pesanan

Pada perancangan antarmuka ini hampir sama dengan tampilan daftar pesanan pada sisi *chef*. Perbedaannya adalah jika pada sisi *chef* yang ditampilkan pesanan berdasarkan *section chef* maka pada sisi *cashier* ditampilkan pesanan berdasarkan meja. Adapun rancangan antarmuka dapat dilihat pada Gambar 3.14 dan Gambar 3.15 berikut.



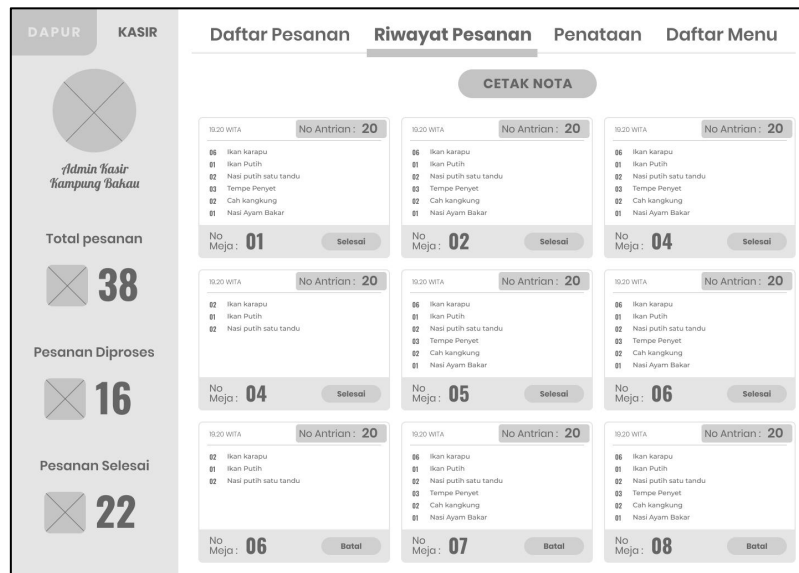
Gambar 3.14 Tampilan Daftar Pesanan



Gambar 3.15 Tampilan Daftar Pesanan Detail

b) Riwayat Pesanan

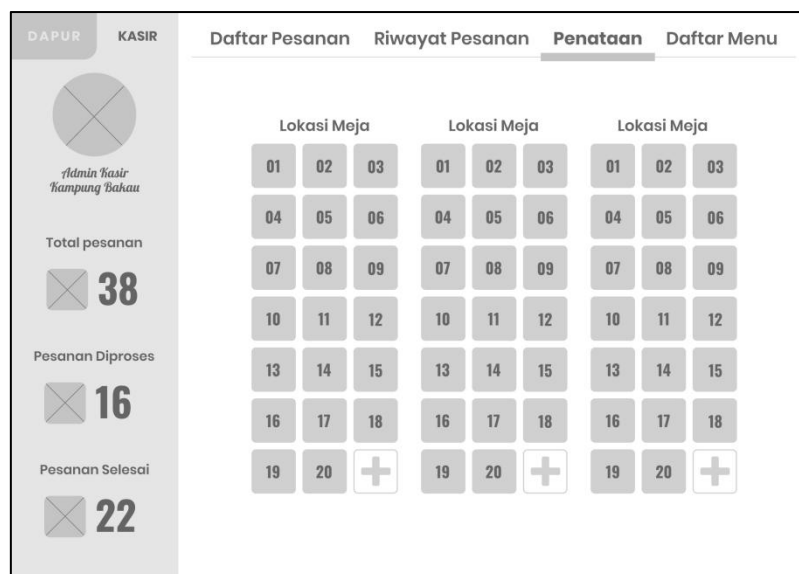
Pada perancangan antarmuka ini menampilkan pesanan-pesanan yang telah diproses dan telah dibayar. Adapun rancangan antarmuka dapat dilihat pada Gambar 3.16 berikut.



Gambar 3.16 Tampilan Riwayat Pesanan

c) Penataan

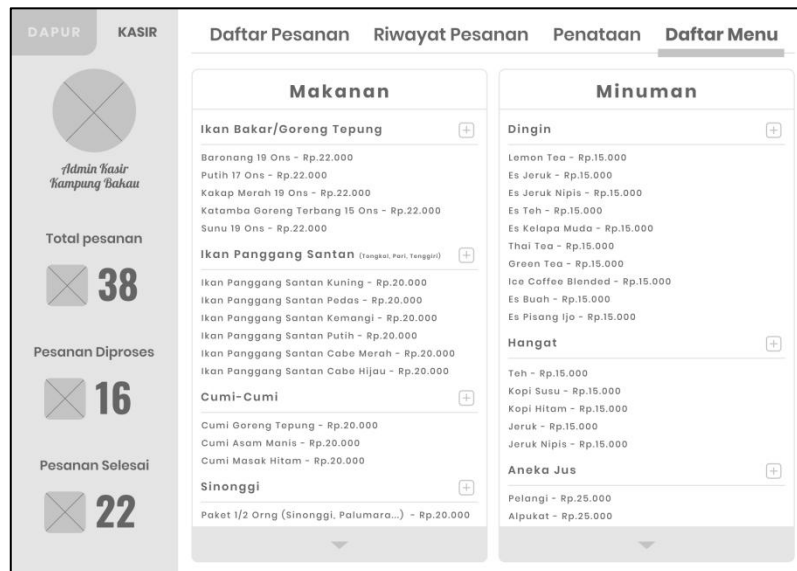
Pada perancangan antarmuka ini menampilkan meja-meja yang ada dan mengelola data-data meja tersebut seperti *generate kode* meja. Adapun rancangan antarmuka dapat dilihat pada Gambar 3.17 berikut.



Gambar 3.17 Tampilan Penataan Meja Makan

d) Daftar Menu

Pada perancangan antarmuka ini menampilkan menu-menu yang ada dan pada halaman ini *cashier* dapat mengelola data-data menu tersebut. Adapun rancangan antarmuka dapat dilihat pada Gambar 3.18 berikut.



Gambar 3.18 Tampilan Daftar Menu

DAFTAR PUSTAKA

- Agae, Yogiokto. *Mengenal Tailwind CSS*. 2020. [Online]. Available: <https://medium.com/devsaurus-class/mengenal-tailwind-css-41a66708863e> [Accessed: 25-Dec-2020].
- Azamuddin, M., & Mukhlisin, H. (2020a). *Modern React Be Frontend Javascript Developer* (M. Azamuddin & H. Mukhlisin (eds.); 1.0.0). KUNGFU KODING. <https://bukureact.id/>
- Azamuddin, M., & Mukhlisin, H. (2020b). *React & Express Be Fullstack Javascript Developer* (M. Azamuddin & H. Mukhlisin (eds.); 1.0.0). KUNGFU KODING. <https://bukureact.id/>
- Bahar, Safril M., Mananohas, Mans Lumiu dan Montolalu, Chriestie E. J. C., 2018, *Model Sistem Antrian dengan Menggunakan Pola Kedatangan dan Pola Pelayanan Pemohon SIM di Satuan Penyelenggaraan Administrasi SIM Resort Kepolisian Manado*, Jurnal Matematika dan Aplikasi deCartesiaN, 7(1), 15. <https://doi.org/10.35799/dc.7.1.2018.19549>.
- Baker, Kenneth R., and Trietsch, Dan., 2009, *Principles Of Sequencing And Scheduling*, New Jersey: John Wiley & Sons.
- Dodu, A. Y. Erwin., Nugraha, Deny Wiria., dan Astuti, Thia Wydia., 2018, *Perbandingan Pembacaan QR Code Menggunakan Metode Standar Dan Metode Euclidean Distance*, semanTIK UHO, 4(2), 99-110. <https://doi.org/10.5281/zenodo.1610994>.
- Firdaus, Ali., Widodo, Slamet., Sutrisman, Adi., Nasution, S. G. F., dan Mardiana, Rina., 2019, *Rancang Bangun Sistem Informasi Perpustakaan Menggunakan Web Service Pada Jurusan Teknik Komputer Polsri*. Jurnal Informatika, 5(2), 83.
- Herdiana, Yana., 2014, *Aplikasi Rumus Matematika Sekolah Menengah Atas Berbasis Mobile*, Jurnal Ilmiah Komputer dan Informatika (KOMPUTA). ISSN: 2089-9033.
- Ihsan, Muhammad Amirul. Apa itu Tailwind CSS. 2020. [Online]. Available: <https://www.kawankoding.id/apa-itu-tailwind-css>. [Accessed: 25-Dec-2020].

- Kong Inc. Getting Started with Insomnia. 2020. [Online]. Available: <https://support.insomnia.rest/article/11-getting-started>. [Accessed: 28-Dec-2020].
- Kurniawan, Awal., Areni, Intan Sari., dan Achmad, Andani., 2018, *Implementasi Progressive Web Application pada Sistem Monitoring Keluhan Sampah Kota Makassar*, JPE-UNHAS, 21(2), 34-38. <https://doi.org/10.25042/jpe.112017.05>.
- Lugina, Restu Setia., “Rancang Bangun Multimedia Pembelajaran Dengan Berbantu Metode Explicit Instruction Untuk Mata Pelajaran Algoritma dan Pemrograman Dasar Pada Materi Sorting,” Universitas Pendidikan Indonesia, 2015.
- Manesia, Pharamita., Hasibuan, Nelly Astuti., dan Saputra, Imam., 2020, *Perancangan Aplikasi Daftar Menu Restoran Berbasis Android Menggunakan Metode Multilevel Queue*, 8(4), pp. 402–407.
- Perkasa, Muhammad Iqbal., dan Setiawan, Eko Budi., 2018, *Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token*, Jurnal ULTIMA Computing, 10(1), 19–26. <https://doi.org/10.31937/sk.v10i1.838>.
- Prasetyo, Tri Wahyu., Wiharto., dan Doewes, Afrizal., 2016, *Pemodelan Penjadwalan Multilevel Feedback Queue Menggunakan Dynamic Time Quantum Pada Kasus Pemesanan Makanan di Restoran*, Jurnal Teknologi & Informasi ITSmart, 4(2), 92. <https://doi.org/10.20961/its.v4i2.1772>.
- Putra, Eko Kurniawanto., dan Rahmayeni, Fitry., 2016, *Implementasi Database MongoDB Untuk Sistem Informasi Bimbingan Konseling Berbasis Web (Studi Kasus: SMPN 1 Sawahlunto)*, Jurnal TeknoIf, 4(1), pp. 67–73.
- Rahmawati, Anita., & Rahman, Arif., 2011, *Sistem Pengamanan Keaslian Ijasah Menggunakan QR-Code dan Algoritma Base64*, Jurnal Sistem Informasi Indonesia, 1(2), 105–112.
- Ramadhani, Mochamed Fajar., 2015, *Pembangunan Aplikasi Informasi, Pengaduan, Kritik, Dan Saran Seputar Kota Cimahi Pada Platform Android*, Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA), 9.

- Rismanto, Ridwan., Arhandi, Putra Prima., dan Prasetyo, Arief., 2016, *Rancang Bangun Aplikasi Ujian Online Real Time Dengan Menggunakan Arsitektur Mean*, Jurnal Teknologi Informasi, 7(2), 149–156.
- Rouillard, Jose., 2008, *Contextual QR codes*. Proc. - The 3rd Int. Multi-Conf. Computing in the Global Information Technology, ICCGI 2008 in Conjunction with ComP2P 2008: The 1st Int. Workshop on Computational P2P Networks: Theory and Practice, July 2008, 50–55. <https://doi.org/10.1109/ICCGI.2008.25>.
- Salim, Boby Hartanto., 2013, *Studi Deskriptif Customer Complaint Di Restoran Hotel Meritus Surabaya*, Jurnal Ilmiah Mahasiswa Universitas Surabaya, 2(2), 1-16.
- Salomo, Ricky., Ranggadara, Indra., Kurnianda, Nia Rahma., and Suhendra, 2019, *Multilevel Algorithm Feedback Queue For Restaurant Order System*, International Journal of Recent Technology and Engineering (IJRTE), 8(3), 5000–5005. <https://doi.org/10.35940/ijrte.C5649.098319>.
- Saputra, Sandika., 2014, *Analisis Kualitas Pelayanan Pada Rumah Makan Metro*, Jurnal Manajemen, 1-60.
- Shalahuddin, M., dan Sukanto, Rosa Ariani., 2011, *Modul Pembelajaran Rekayasa (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- Syaifudin, Kukuh., Pranata, Ego., Amwayandya, Wafa Nafisah., dan Dian, Arum Rawining., 2019, *Analisis Usability pada Perbandingan Web-Native dengan Web Berbasis Progressive Web App*, Informatika.
- Widayati, Yohana Tri., 2017, *Aplikasi Teknologi QR (Quick Response) Code Implementasi Yang Universal*, Jurnal KOMPUTAKI, 3(1), 66-81.