

# Finite State Machine Pada Perancangan Line E-Commerce Chatbot Menggunakan NodeJs MongoDB dan Express

Farid Angga Pribadi<sup>1</sup>, Annisa Taufika Firdausi<sup>2</sup>, Putra Prima Arhandi<sup>3</sup>

<sup>1,2,3</sup> Jurusan Teknologi Informasi, Politeknik Negeri Malang

<sup>1</sup>faridangga@polinema.ac.id, <sup>2</sup>annisa.taufika@polinema.ac.id, <sup>3</sup>putraprima@polinema.ac.id

---

## Abstrak

Perkembangan ilmu pengetahuan dan teknologi di bidang teknologi informasi mengakibatkan terjadinya perubahan perilaku pengguna di internet. Saat ini media sosial mengambil peranan penting dalam pemasaran dan penjualan. Salah satu media sosial yang populer di Indonesia adalah Line messenger. Line memiliki fitur chatbot yang memungkinkan untuk membuat chat berbasis bot yang dapat dijalankan pada aplikasi Line messenger. Namun terdapat kekurangan pada chatbot yaitu manajemen menu dan fungsi yang ditampilkan ke user tidak termanajemen dengan baik. Masalah tersebut dapat diselesaikan dengan menggunakan finite state machine. Metode finite state machine digunakan untuk menentukan alur pengguna saat memilih menu yang tampil pada chatbot. Penelitian ini bertujuan untuk merancang dan membangun chatbot yang akan dimanfaatkan untuk penjualan dan pembelian serta menerapkan metode finite state machine untuk manajemen menu yang ditampilkan pada chatbot. Model pengembangan Software development life cycle (SDLC) yang digunakan adalah model Rapid Application Development (RAD). RAD memiliki beberapa tahapan yaitu analisa (analysis), perancangan (design), implementasi (implementation), pengujian (testing), pemeliharaan (maintenance). Pada penelitian ini dilakukan dua jenis pengujian, yang pertama pengujian finite state, pengujian dilakukan dengan melakukan input terhadap state chatbot dan membandingkan response chatbot dengan hasil yang diharapkan. Yang kedua Usability Testing yaitu pengujian dilakukan menggunakan kuisioner Computer System Usability Questionnaire (CSUQ). Hasil penelitian ini menunjukkan bahwa 87% pengguna dapat dengan mudah untuk menggunakan sistem chatbot yang digunakan untuk penjualan dan pembelian serta metode finite state machine dapat menampilkan menu sesuai state yang ditentukan.

**Kata kunci :** Line Chatbot, Finite State Machine, E-Commerce.

---

## 1. Pendahuluan.

Berkembangnya ilmu pengetahuan dan teknologi di bidang teknologi informasi mengakibatkan terjadinya perubahan perilaku pengguna di internet di Indonesia. Saat ini media sosial mengambil peranan penting dalam pemasaran dan penjualan. Berdasarkan survey Asosiasi E-Commerce Indonesia (idEA) yang dilaksanakan pada tahun 2017, penggunaan platform media sosial untuk menjual dan membeli barang mencapai 66%. Hanya 16% penjual menggunakan platform marketplace dan sisanya memilih menggunakan website sendiri (Nugraha, 2020). Hal ini menunjukkan bahwa penggunaan platform media sosial untuk e-commerce masih populer di masyarakat.

Line messenger adalah satu diantara aplikasi komunikasi yang populer di Indonesia. Hingga saat ini dari total 220 juta penggunanya di Dunia, 90 juta diantaranya berasal dari Indonesia. Line memiliki fitur Line chatbot yang memungkinkan untuk membuat chat berbasis bot yang dapat dijalankan pada aplikasi Line messenger. Sehingga dengan adanya fitur Line chatbot dan penggunanya yang cukup banyak, Line messenger dapat memberikan

fleksibilitas dalam memasarkan dan penjualan produk.

Chatbot adalah sebuah software yang dibuat untuk dapat berkomunikasi dengan konsumen (manusia) melalui percakapan tertulis, suara, dan visual. Chatbot dapat menanggapi ribuan percakapan dengan cepat. Chatbot banyak dimanfaatkan oleh perusahaan karena perusahaan dapat meminimalkan biaya operasional dan memberikan pelayanan yang aktif selama 24 jam. Sehingga chatbot dapat meminimalkan jumlah pekerja dan memaksimalkan kualitas layanan terhadap konsumen (Amalia & Wibowo, 2019).

Pada chatbot terdapat objek-objek menu yang dapat menampilkan informasi dan atau feedback sesuai dengan keinginan user. Namun dengan banyaknya menu dan fungsi yang berada pada chatbot membuat manajemen setiap menu dan fungsi didalam chatbot tidak termanajemen dengan baik.

Salah satu solusi untuk melakukan manajemen menu dan fungsi pada chatbot dapat digunakan sebuah Finite State Machine (FSM). FSM adalah sebuah metodologi yang memiliki tiga komponen utama yaitu State, Event dan Action, dengan menggunakan tiga komponen ini dapat dibuat sebuah

sistem kontrol yang mengatur proses kerja pada sebuah sistem (Setiawan, 2006) (Tambahan 4). FSM dapat diimplementasikan pada berbagai bidang yang bervariasi baik pada game dimana pada pembuatan game FSM sering digunakan sebagai pengatur menu, user interface, atau perpindahan antara satu state ke state lain (Hariadi, 2019; Hormansyah et al., 2018; Kurniawardhani et al., 2014; Saputra et al., 2019), ataupun pada sistem pemantauan polusi udara (tambahan 3)(Prahardis et al., 2018), vending machine (Tambahan 1, Tambahan 2).

Berkaitan dengan permasalahan tersebut, telah ada beberapa penelitian dan buku yang dipublikasikan, antara lain adalah buku yang ditulis oleh (Chand, 2017) dimana pada buku ini finite state machine menjadi salah satu metode populer yang digunakan untuk membuat chatbot. Banyak lagi penelitian lain yang memanfaatkan finite state machine pada chatbot, antara lain chatbot untuk merekomendasikan ban pada konsumen (Colace et al., 2017) pada penelitian ini finite state machine dikombinasikan dengan petri net, pada penelitian lain mengkombinasikan antara finite state machine dengan information retrieval untuk membuat sebuah sistem percakapan yang efisien antara pengguna dan chatbot ggowo 2, selain itu finite state machine juga digunakan dalam chatbot pada bidang kesehatan (Fadhil & Gabrielli, 2017).

Berdasarkan luasnya implementasi finite state machine pada chatbot dengan berbagai bidang, pada penelitian ini dibuat sebuah chatbot pada platform Line Messenger, yang dapat digunakan sebagai alternatif pembeli untuk pemesanan produk dan mempermudah penjual dalam melayani pelanggan. Pada penelitian ini chatbot ini akan mengimplementasikan Finite State Machine untuk mengatasi masalah state management dan pengaturan setiap menu dan fungsi pada chatbot.

## 2. Studi Literatur

### 2.1 Chatbot

Chatbot adalah sebuah sistem yang memungkinkan percakapan yang lancar antara pengguna dan komputer dimana pengguna merasa sedang berkomunikasi dengan manusia, ada banyak contoh penerapan chatbot antara lain pada bidang perbankan, asuransi dan bisnis lainnya. Chatbot yang dibuat pada umumnya digunakan untuk melayani konsumen dengan keunggulan pelayanan dapat berjalan setiap hari tanpa harus menggunakan tenaga manusia dan menurunkan biaya produksi.

Pada umumnya terdapat dua pendekatan dalam membuat chatbot yaitu Rule Based Approach (Menu Based Approach) dan AI Based Approach. Pendekatan pertama merupakan cara yang paling mudah dan sederhana, sedangkan pada pendekatan kedua membutuhkan komputasi dan metode NLP

yang mumpuni untuk memahami maksud dari pengguna (Abhishek et al., 2019).

### 2.2 NodeJs

NodeJs adalah sebuah platform baru yang dapat digunakan untuk berbagai tujuan antara lain aplikasi web, aplikasi server dan tujuan umum lainnya. NodeJs di desain untuk memiliki skalabilitas yang baik pada aplikasi yang menggunakan jaringan dengan menggunakan kombinasi fitur yang dimilikinya antara lain server side javascript, asynchronous IO dan asynchronous programming. NodeJs cukup populer terbukti dengan digunakannya NodeJs pada perusahaan besar seperti Paypal yang sudah mengkonversi servis nya dari bahasa pemrograman java ke bahasa NodeJs (Herron, 2020).

### 2.3 Mongo DB

MongoDB adalah jenis database yang menggunakan konsep berkas (NoSQL). NoSQL adalah tipe basis data yang memiliki skema fleksibel, dengan demikian MongoDB tidak menggunakan perintah SQL seperti pada database relasional. MongoDB memiliki keunggulan yaitu data tidak disimpan menggunakan tabel. Akan tetapi, di simpan dalam bentuk dokumen terstruktur berformat JSON. Sehingga performa yang di hasilkan oleh MongoDB akan lebih cepat (Maksimovic, 2017). Pada implementasinya MongoDB dapat dipasangkan dengan berbagai macam stack javascript salah satu yang populer adalah MERN stack dimana pada stack ini digunakan MongoDB, Express Js, React dan NodeJs (Wilson & Koroliov, 2018).

### 2.4 Express Js

Express JS adalah sebuah framework node.js untuk membuat aplikasi web yang bersifat minimal dan fleksibel, dapat digunakan untuk membuat single page app, multipage web app dan hybrid web app. Express dibuat oleh TJ Holowaychuk dan terinspirasi dari framework lain yaitu Sinatra dimana Sinatra berbasis Ruby (Ethan Brown, 2014).

### 2.5 REST (*Representational State Transfer*)

REST adalah sebuah konsep yang dikemukakan oleh Roy Fielding pada disertasi PhD nya. Prinsip utama dari REST adalah penggunaan protokol HTTP untuk komunikasi data yang menggunakan konsep resource dimana setiap komponen dianggap sebagai sebuah resource dan diakses menggunakan metode HTTP. Konsep ini kemudian diadopsi sebagai metode untuk bertukar data oleh berbagai framework populer seperti express, go framework, spring boot dan python RESTfull API (Subramanian & Raj, 2019).

Nomor urut tabel ditulis di bagian atas tabel dan disertai judul atau nama tabel, contoh: Tabel 1, Hasil pengukuran keluaran sistem, Tabel 2(a). Hasil pengukuran pada titik A, dll. Tabel-tabel yang

memerlukan tempat melebihi satu kolom, dapat disajikan di bagian akhir makalah dalam dua kolom dengan tetap diberi nomor urut yang sesuai.

**3. Data**

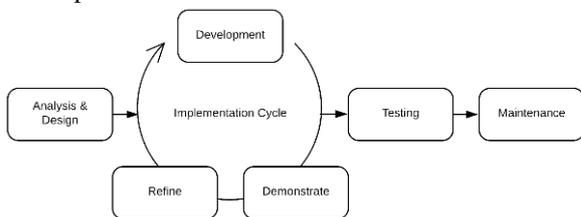
Untuk mengumpulkan data pada penelitian ini dilakukan dengan melakukan wawancara dan melakukan pengamatan pada sistem yang sudah berjalan kepada pemilik UMKM Juwalan, media yang digunakan adalah chat dan komunikasi video call melalui internet.

**4. Lokasi Penelitian**

Pelaksanaan penelitian di lakukan di Politeknik Negeri Malang Lab Pemrograman 01 Jalan Soekarno Hatta no 9 Kota Malang.

**5. Metode Penelitian**

Model pengembangan Software development life cycle (SDLC) yang digunakan adalah model Rapid Application Development (RAD). RAD memiliki beberapa tahapan yaitu analisa (analysis), perancangan (design), implementasi (implementation), pengujian (testing), pemeliharaan (maintenance) detail langkah langkah RAD dapat dilihat pada Gambar 1.



Gambar 1 Metode Pengembangan Perangkat Lunak RAD

**5.1 Analysis**

Pada tahap ini mengumpulkan dan menganalisis kebutuhan perangkat lunak yang diinginkan oleh pengguna. Hasil dari analisa pada tahap ini menghasilkan kebutuhan fungsional dan kebutuhan non fungsional.

**5.2 Design**

Setelah mengumpulkan kebutuhan yang digunakan pada sistem, maka dibuat perancangan sistem berupa: Desain sistem, use case diagram, activity diagram, wireframe website chatbot, wireframe chatbot, desain database sistem.

**5.3 Implementation**

Tahap ini adalah tahap pembuatan sistem dari perancangan sistem yang telah direncanakan pada tahap sebelumnya. Pada penelitian ini digunakan fullstack javascript nodeJs, Express, React dan MongoDB.

**5.4 Testing**

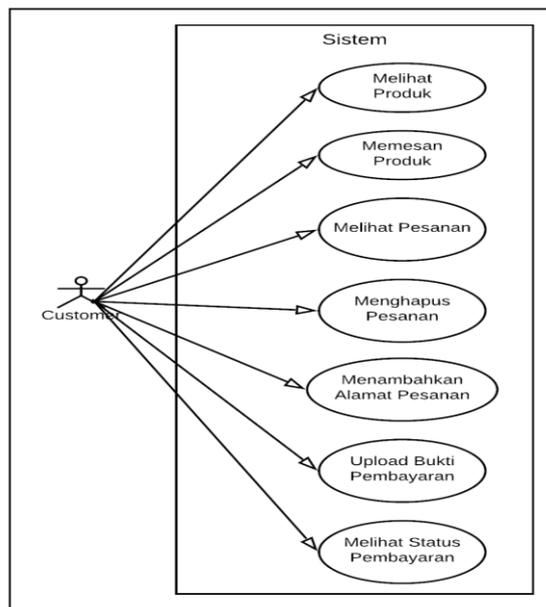
Setelah selesai pada tahap pembuatan sistem. Sistem akan diuji menggunakan dua metode yaitu Black Box Testing dan Usability Testing. Black Box Testing merupakan pengujian yang menguji spesifikasi fungsional dari sistem yang dibuat (Mustaqbal et al., 2015). Pengujian tersebut dilaksanakan untuk memastikan apakah sistem telah berhasil memenuhi kebutuhan sesuai dengan kebutuhan yang ada pada tahap analisis. Usability testing merupakan pengujian yang berfokus pada pengguna, pengujian ini menilai apakah pengguna dapat menggunakan sistem yang dibuat dengan mudah atau tidak.

**6. Hasil dan Pembahasan**

**6.1 Desain Perangkat Lunak**

**1. Usecase Diagram Customer**

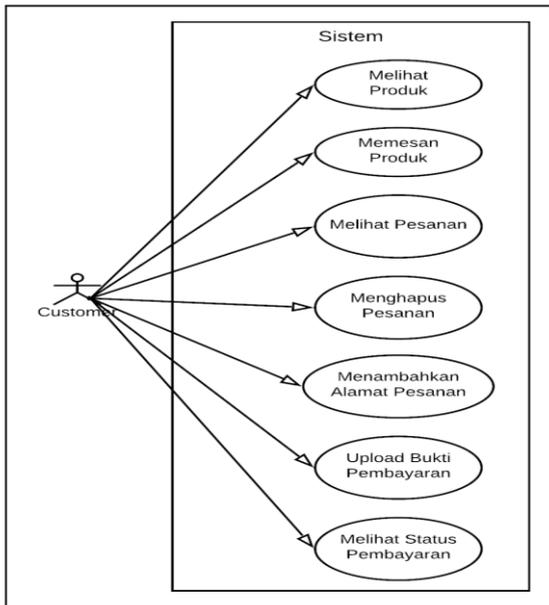
Gambar 2 merupakan gambaran use case customer pada chatbot. Pada chatbot customer dapat melihat produk, memesan produk, melihat pesanan yang telah dipilih, menghapus pesanan, menambahkan alamat pesanan, mengupload bukti pembayaran dan melihat status pembayaran setelah dikonfirmasi oleh admin.



Gambar 2 Use Case Diagram Chatbot Customer

**2. Usecase Diagram Admin**

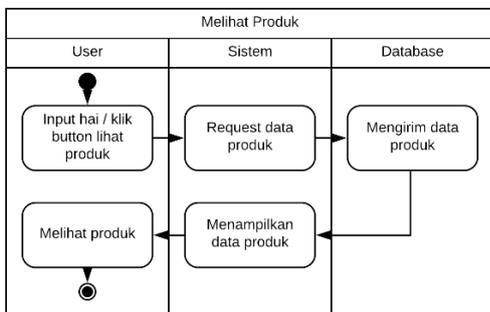
Gambar 3 merupakan gambaran use case admin pada chatbot. Pada chatbot admin dapat melihat produk, memesan produk, melihat pesanan yang telah di pilih, menghapus pesanan, menambahkan alamat pesanan, mengupload bukti pembayaran, melihat daftar pesanan yang telah dipesan oleh customer, melihat daftar bukti pembayaran yang telah di upload oleh customer, mengkonfirmasi pembayaran customer dan melihat status pembayaran.



Gambar 3 Use Case Diagram Chatbot Admin

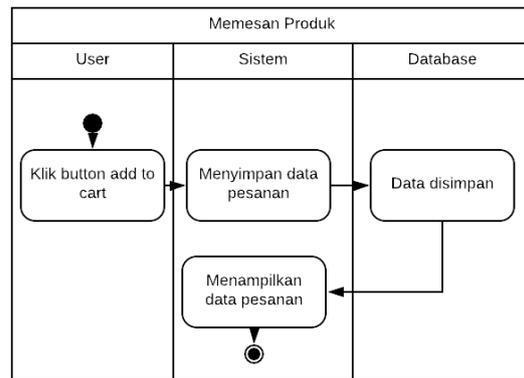
### 3. Activity Diagram

Diagram aktivitas Gambar 4 menerangkan proses untuk aktivitas melihat produk. Proses dimulai dengan user menginputkan kata hai atau menekan tombol lihat produk, kemudian sistem request data produk dan sistem menampilkan data produk dari database.



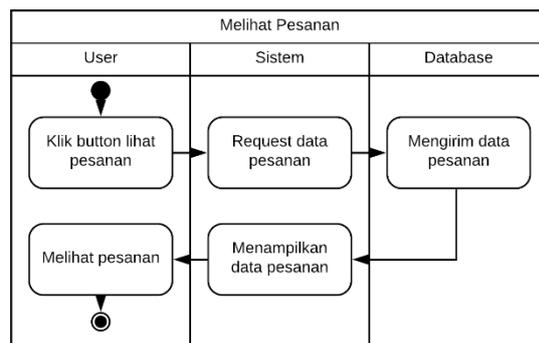
Gambar 4 Activity Diagram melihat produk

Diagram aktivitas Gambar 5 menerangkan proses untuk aktivitas memesan produk. Proses dimulai dengan user menekan tombol add to cart kemudian sistem menyimpan data pesanan ke database dan menampilkan data pesanan dari database.



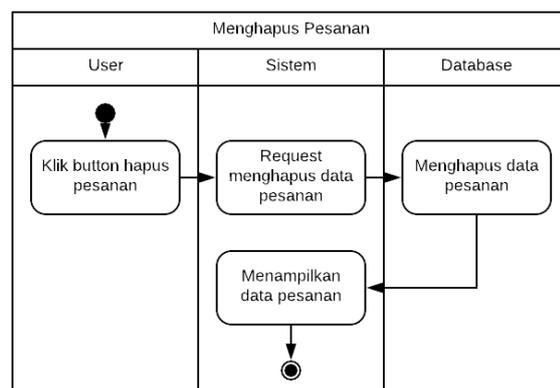
Gambar 5 Activity Diagram memesan produk

Diagram aktivitas Gambar 6 menerangkan proses untuk aktivitas melihat pesanan. Proses dimulai dengan user menekan tombol lihat pesanan yang tersedia, kemudian sistem merequest data pesanan ke database, database mengirim data ke sistem, kemudian sistem menampilkan data pesanan dari database sesuai request yang diterima oleh database.



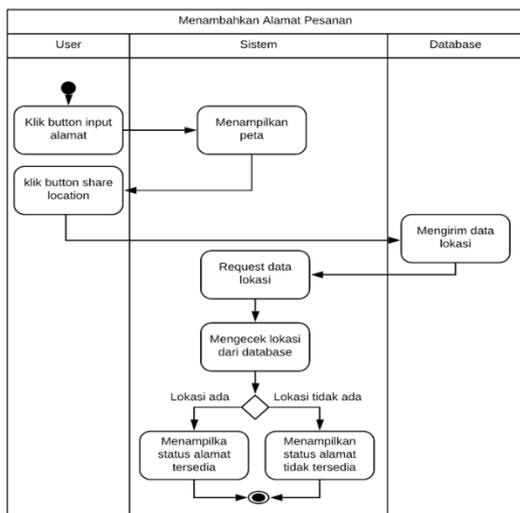
Gambar 6 Activity Diagram Melihat Pesanan

Diagram aktivitas Gambar 7 menerangkan proses untuk aktivitas menghapus pesanan. Proses dimulai dengan user menekan tombol hapus pesanan yang tersedia, kemudian sistem request menghapus data ke database, database menghapus data pesanan sesuai request dan sistem menampilkan data pesanan kosong.



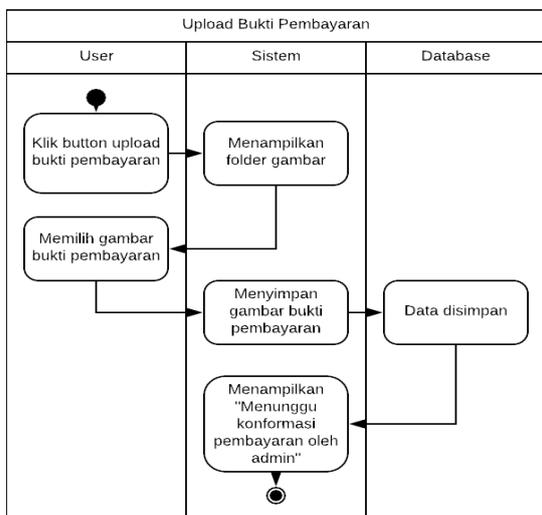
Gambar 7 Activity Diagram Menghapus Pesanan

Diagram aktivitas Gambar 8 menerangkan proses untuk aktivitas menambahkan alamat pesanan. Proses dimulai dengan user menekan tombol input alamat, kemudian sistem menampilkan peta, user menekan tombol share location, sistem mengecek data di database. Jika data alamat tersedia di database, sistem akan menghitung ongkos kirim dan menampilkan ongkos kirim serta status alamat tersedia. Jika alamat tidak tersedia, sistem akan menampilkan status alamat tidak tersedia.



Gambar 8 Activity Diagram Menambahkan Alamat Pesanan

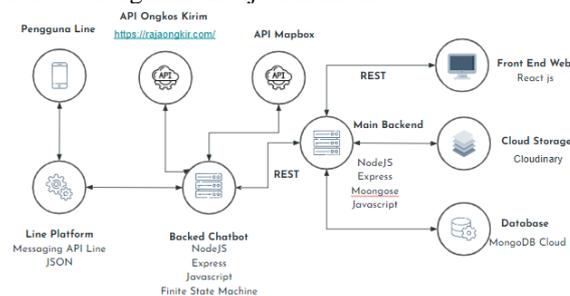
Diagram aktivitas Gambar 9 menerangkan alur proses untuk aktivitas upload bukti pembayaran. Proses dimulai dengan user menekan tombol upload bukti pembayaran, kemudian sistem menampilkan folder gambar, setelah itu user memilih bukti pembayaran, sistem akan menyimpan data ke database dan sistem akan menampilkan status menunggu konfirmasi pembayaran oleh admin.



Gambar 9 Activity Diagram Mengupload Bukti Pembayaran

4. Desain Sistem

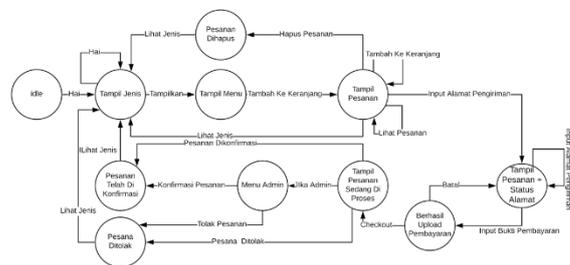
Pada Gambar 10 merupakan arsitektur sistem yang dibuat. Sistem ini menggunakan Line Platform Messaging API yang merupakan fitur pada line chatbot yang memungkinkan komunikasi dua arah antara pengguna line dengan backend chatbot yang di buat oleh developer. Line Platform Messaging API akan mengirimkan informasi berbentuk JSON melalui protokol HTTPS ke URL yang ditetapkan sebagai webhook. Backend chatbot kemudian mengirimkan respon berdasarkan request yang dikirim pengguna. Pada main backend digunakan untuk mengelola data yang diinputkan oleh user menggunakan web, untuk penyimpanan data gambar menggunakan cloud storage cloudinary dan untuk penyimpanan data teks menggunakan mongoDb cloud. Komunikasi main backend dengan backend chatbot menggunakan REST API dengan framework expressjs. API ongkos kirim digunakan oleh backend chatbot untuk mengambil data ongkos kirim dan API Mapbox digunakan untuk mengkonfersi koordinat latitude longitude menjadi nama kota.



Gambar 10 Desain Sistem

5. Desain State Transisi

Pada Gambar 11 menjelaskan rancangan alur finite state machine yang akan diterapkan di line e-commerce chatbot.



Gambar 11 Desain Transisi State

Pada tahap desain state dan transisi di Gambar 12 menggunakan state machine diagram. Keuntungan state machine diagram dibanding dengan flowchart yaitu state machine diagram pada sistem ini menunjukkan perubahan tiap state(keadaan) berdasarkan event(kondisi) yang dipilih pengguna. Sedangkan pada flowchart menunjukkan proses yang di jalankan pada sistem. Detail proses state dan

transisi dapat dilihat pada Tabel 1 Penjelasan Desain State dan Transisi.

Tabel 1. Penjelasan Desain State dan Transisi

<i>Current State</i>	<i>Input</i>	<i>State Transition</i>
idle	Input “Hai”	Tampil Jenis
Tampil Jenis	Input “Hai”	Tampil Jenis
Tampil Jenis	Klik button “Tampilkan”	Tampil Menu
Tampil Menu	Klik button “Tambah Ke Keranjang”	Tampil Pesanan
Tampil Pesanan	Klik button “Lihat Pesanan”	Tampil Pesanan
Tampil Pesanan	Klik button “Tambah Ke Keranjang”	Tampil Pesanan
Tampil Pesanan	Klik button “Hapus Pesanan”	Pesanan Dihapus
Pesanan Dihapus	Klik button “Lihat Jenis”	Tampil Jenis
Tampil Pesanan	Klik button “Lihat Jenis”	Tampil Jenis
Tampil Pesanan	Input alamat pengiriman	Tampil Pesanan Status Alamat
Tampil Pesanan Status Alamat	Input alamat pengiriman	Tampil Pesanan Status Alamat
Tampil Pesanan Status Alamat	Input gambar bukti pembayaran	Berhasil Upload Pembayaran
Berhasil Upload Pembayaran	Klik button “Checkout”	Pesanan Di Proses
Berhasil Upload Pembayaran	Klik button “Batal”	Tampil Pesanan Status Alamat
Pesanan Di Proses	Jika user admin	Menu Admin
Pesanan Di Proses	Jika pesanan dikonfirmasi	Pesanan Di Konfirmasi
Pesanan Di Proses	Jika pesanan ditolak	Pesanan Ditolak
Menu Admin	Klik button “Konfirmasi Pesanan”	Pesanan Di Konfirmasi

Menu Admin	Klik button “Tolak Pesanan”	Pesanan Ditolak
Pesanan Ditolak	Klik button “Lihat Jenis”	Tampil Jenis
Pesanan Di Konfirmasi	Klik button “Lihat Jenis”	Tampil Jenis

### 6.2 Implementasi

Penerapan metode finite state machine pada line e-commerce chatbot ini memiliki kelebihan seperti chatbot dapat membatasi menu yang ditampilkan saat user berada di state yang telah di tentukan. Membatasi menu yang dimaksud adalah dapat dilihat pada contoh salah satu state pada Gambar 12 berikut ini.



Gambar 12 Implementasi Keranjang Chatbot

#### 1. Implementasi Wireframe

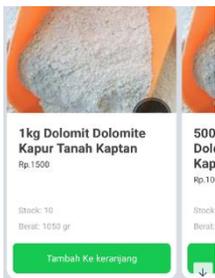
Pada tahap ini dilakukan implementasi dari wireframe yang dibuat pada bab sebelumnya. Tabel 2. Menunjukkan beberapa gambar implementasi dari wireframe beserta penjelasannya. Berikut implementasi yang telah dibuat:

Pada saat user berdada di state tampil jenis kemudian user menginputkan gambar bukti pembayaran, chatbot akan menampilkan pesan “Anda tidak diperbolehkan mengakses button lain pada state ini. Gunakan button “Tampilkan” untuk melanjutkan order”. Hal tersebut dikarenakan input gambar bukti pembayaran diperbolehkan pada saat user berada di state tampil pesanan status alamat.

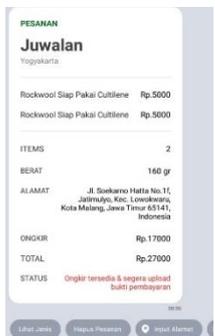
Tabel 2. Implementasi Wireframe

Gambar	Keterangan
	Pada list jenis produk, pengguna dapat melihat carousel message yang berisi list data jenis produk. Button tampilkan digunakan untuk menampilkan data produk berdasarkan

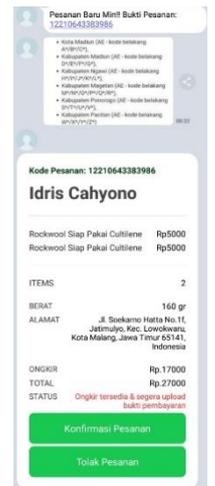
jenis produk yang dipilih.



List produk merupakan pesan yang tampil setelah pengguna klik button produk. Pada list produk pengguna dapat melihat carousel message yang berisi list produk berdasarkan jenis produk yang dipilih. Button tambah keranjang digunakan untuk menambahkan produk yang dipilih ke keranjang.



Keranjang merupakan pesan yang tampil setelah pengguna klik button tambah ke keranjang. Pada keranjang terdapat button quick reply berisi list jenis produk, hapus pesanan untuk menghapus pesan, input alamat untuk menginputkan alamat pengiriman, dan upload pembayaran untuk memilih gambar bukti pembayaran di gallery.



Keranjang admin merupakan pesan yang tampil setelah pengguna klik button checkout. Pada keranjang admin terdapat pesan gambar pembayaran, button konfirmasi untuk mengkonfirmasi pembayaran dan button tolak pesanan untuk menolak pesanan.



Berhasil upload pembayaran merupakan pesan yang muncul setelah pengguna upload bukti pembayaran. Pada pesan ini terdapat dua button quick

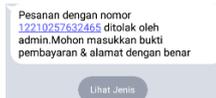
berisi button checkout untuk melanjutkan pemesanan dan button batal untuk menampilkan keranjang.

Pesanan diproses merupakan pesan yang muncul setelah pengguna klik button checkout.

Pesanan dikonfirmasi merupakan pesan yang tampil setelah admin klik button konfirmasi pesanan. Pada pesan ini terdapat button quick reply untuk melihat jenis produk.

Pesanan ditolak merupakan pesan yang tampil setelah admin klik button tolak pesanan. Pada pesan ini terdapat button quick reply untuk melihat jenis produk.

Pesanan dihapus merupakan pesan yang tampil setelah pengguna klik button hapus pesanan. Pada pesan ini terdapat button quick reply untuk melihat jenis produk.



### 6.3 Testing

#### 1. Testing Finite State Machine

Pada tahap ini dilakukan pengujian terhadap finite state machine yang diimplementasikan ke chatbot. Pengujian dilakukan dengan melakukan input terhadap state chatbot dan membandingkan response chatbot dengan hasil yang diharapkan, detail mengenai input, hasil yang diharapkan dan hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Testing Finite State Machine

Jalur State	Input	Hasil yang Diharapkan	Hasil Pengujian
Idle – Tampil Jenis	Input “Hai”	Tampil Jenis	Berhasil
Tampil Jenis – Tampil Jenis	Input “Hai”	Tampil Jenis	Berhasil

Tampil Jenis – Tampil Menu	Klik <i>button</i> “Tampilkan”	Tampil Menu	Berhasil	Pesanan Di Proses – Pesanan Ditolak	Jika pesanan ditolak	Pesanan Ditolak	Berhasil
Tampil Menu – Tampil Pesanan	Klik <i>button</i> “Tambah Ke Keranjang”	Tampil Pesanan	Berhasil	Menu Admin – Pesanan Dikonfirmasi	Klik <i>button</i> “Konfirmasi Pesanan”	Pesanan Di Konfirmasi	Berhasil
Tampil Pesanan – Tampil Pesanan	Klik <i>button</i> “Lihat Pesanan”	Tampil Pesanan	Berhasil	Menu Admin – Pesanan Ditolak	Klik <i>button</i> “Tolak Pesanan”	Pesanan Ditolak	Berhasil
Tampil Pesanan – Tampil Pesanan	Klik <i>button</i> “Tambah Ke Keranjang”	Tampil Pesanan	Berhasil	Pesanan Di Konfirmasi – Tampil Jenis	Klik <i>button</i> “Lihat Jenis”	Tampil Jenis	Berhasil
Tampil Pesanan – Pesanan Dihapus	Klik <i>button</i> “Hapus Pesanan”	Pesanan Dihapus	Berhasil	2. <i>Usability Testing</i> Pada tahap ini pengujian dilakukan menggunakan kuisioner Computer System Usability Questionnaire (CSUQ), kuisioner ini dibuat oleh IBM sebagai parameter usability testing pada perangkat lunak (Lewis, 1995) yang terdiri dari 19 item pertanyaan. Berikut ini merupakan hasil dari kuisioner yang dibagikan kepada 22 responden dan dihitung dengan rumus sebagai berikut:			
Pesanan Dihapus – Tampil Jenis	Klik <i>button</i> “Lihat Jenis”	Tampil Jenis	Berhasil	$Hasil = \frac{Total\ Skor\ Diperoleh}{Skor\ Maksimum} \times 100.$			
Tampil Pesanan – Tampil Jenis	Klik <i>button</i> “Lihat Jenis”	Tampil Jenis	Berhasil	Berdasarkan data pada hasil pengujian System Quality pada Tabel 4 diperoleh rata-rata presentase 87.6% dimana dalam presentase merupakan kategori baik. Aspek Information Quality diperoleh rata-rata presentase 87% dimana dalam presentase merupakan kategori baik. Aspek Interface Quality diperoleh rata-rata presentase 85.7% dimana dalam presentase merupakan kategori baik. Sedangkan total presentase dari keseluruhan aspek Overall diperoleh rata-rata presentase 87% dimana dalam presentase merupakan kategori baik.			
Tampil Pesanan – Tampil Pesanan Status Alamat	<i>Input</i> alamat pengiriman	Tampil Pesanan Status Alamat	Berhasil	Tabel 4. Hasil <i>Usability Testing</i>			
Tampil Pesanan Status Alamat – Tampil Pesanan Status Alamat	<i>Input</i> alamat pengiriman	Tampil Pesanan Status Alamat	Berhasil	<b>Nama Skor</b>	<b>Item</b>	<b>Presentase</b>	<b>Rata - Rata</b>
Tampil Pesanan Status Alamat – Upload Pembayaran	<i>Input</i> gambar bukti pembayaran	Berhasil Upload Pembayaran	Berhasil		P1	89.1%	87.6%
Berhasil Upload Pembayaran	Klik <i>button</i> “Checkout”	Pesanan Di Proses	Berhasil		P2	89.1%	
Berhasil Upload Pembayaran – Pesanan Diproses	Klik <i>button</i> “Batal”	Tampil Pesanan Status Alamat	Berhasil		SY P3	90.9%	
Berhasil Upload Pembayaran – Tampil Pesanan Status Alamat	Jika <i>user</i> admin	Menu Admin	Berhasil		SQ P4	85.5%	
Pesanan Di Proses – Pesanan Dikonfirmasi	Jika pesanan dikonfirmasi	Pesanan Di Konfirmasi	Berhasil		UA P5	88.2%	
					L P6	87.3%	
					P7	89.1%	
					P8	81.8%	
					P9	86.4%	87%
					P10	87.3%	
					IN P11	89.1%	
					FO P12	88.2%	
					QU P13	88.2%	
					AL P14	86.4%	
					P15	83.6%	
					P16	86.4%	85.7%
					INT P17	84.5%	
					QU P18	82.7%	
					AL P19	89.1%	

Rata 87%  
-  
Rata

---

#### 6.4 Kesimpulan

Berdasarkan proses pengujian yang dilaksanakan sebelumnya, dapat diambil kesimpulan Hasil dari pengujian usability pada sistem line e-commerce chatbot menggunakan finite state machine menghasilkan nilai 87%. Sehingga dapat disimpulkan bahwa sistem tersebut dapat dijadikan sebagai penjualan dan pembelian.

Finite State Machine dapat diimplementasikan dalam perancangan sistem ini. Setelah diuji coba Finite State Machine yang dibuat dapat menampilkan menu sesuai dengan state yang ditentukan.

Pada penelitian ini terbatas pada pengujian sistem yang itu sendiri dengan hasil pengujian finite state machine sebesar 87%. Kedepannya dapat diteliti dampak dari penggunaan aplikasi chatbot menggunakan finite state terhadap performa bisnis.

#### Daftar Pustaka:

- Abhishek, S., Karthik, R., & Shrey, S. (2019). *Building an Enterprise Chatbot*. Apress.
- Amalia, E. L., & Wibowo, D. W. (2019). Rancang Bangun Chatbot Untuk Meningkatkan Performa Bisnis. *Jurnal Ilmiah Teknologi Informasi Asia*, 13(2), 137. <https://doi.org/10.32815/jitika.v13i2.410>
- Chand, J. (2017). Chatbot: architecture, design & development. *Scientific Reports*, 19, 46.
- Colace, F., de Santo, M., Pascale, F., Lemma, S., & Lombardi, M. (2017). BotWheels: a Petri Net based Chatbot for Recommending Tires. *Proceedings of the 6th International Conference on Data Science, Technology and Applications*, 350–358. <https://doi.org/10.5220/0006491903500358>
- Ethan Brown. (2014). *Web Development with Node and Express: Leveraging the JavaScript Stack* (1st ed., Vol. 1). O'Reilly Media, Inc.
- Fadhil, A., & Gabrielli, S. (2017). Addressing challenges in promoting healthy lifestyles. *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, 261–265. <https://doi.org/10.1145/3154862.3154914>
- Hariadi, F. (2019). Manajemen User Interface Pada Game 2D Berbasis Finite State Machine. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 3(2), 144. <https://doi.org/10.30865/mib.v3i2.1159>
- Herron, D. (2020). *Node.js Web Development*. Pact.
- Hormansyah, D. S., Ririd, A. R. T. H., & Pribadi, D. T. (2018). IMPLEMENTASI FSM (FINITE STATE MACHINE) PADA GAME PERJUANGAN PANGERAN DIPONEGORO. *Jurnal Informatika Polinema*, 4(4), 290. <https://doi.org/10.33795/jip.v4i4.222>
- Kurniawardhani, A., Suciati, N., & Arieshanti, I. (2014). KLASIFIKASI CITRA BATIK MENGGUNAKAN METODE EKSTRAKSI CIRI YANG INVARIANT TERHADAP ROTASI. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 12(2), 48. <https://doi.org/10.12962/j24068535.v12i2.a322>
- Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1), 57–78. <https://doi.org/10.1080/10447319509526110>
- Maksimovic, Z. (2017). *MongoDB 3 Succinctly*. Syncfusion.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1, 31–36.
- Nugraha, A. (2020). Analisis Faktor-Faktor yang Mempengaruhi Behavioral Intention untuk Mengadopsi Media Sosial sebagai Tempat Belanja Online. *Jurnal Administrasi Niaga Politeknik Negeri Bandung*, 11, 26–27.
- Prahardis, R., Syauqy, D., & Akbar, S. R. (2018). Implementasi Sistem Monitoring Polusi Udara Berdasarkan Indeks Standar Pencemaran Udara Dengan Pemodelan Finite State Machine. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2 No 9.
- Saputra, S. A., Haryanto, H., & Dolphina, E. (2019). Skenario Dinamis Menggunakan Finite State Machine pada Game Pengenalan Tempat dan Peristiwa Bersejarah. *Eksplora Informatika*, 8(2), 112–121. <https://doi.org/10.30864/eksplora.v8i2.155>
- Setiawan, I. (2006). Perancangan Software Embedded System Berbasis FSM. *Jurnal Teknik Elektro*, 1–2.
- Subramanian, H., & Raj, P. (2019). *Hands-On RESTful API Design Patterns and Best Practices: Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs*. Pact.
- Wilson, E., & Koroliov, I. (2018). *MERN Quick Start Guide: Build web applications with MongoDB, Express.js, React, and Node*. Pact.

