

## BAB III

### ANALISIS DAN DESAIN SISTEM

#### III.1. Analisis Masalah

PT. Ouzen Anugerah Indonesia merupakan perusahaan *startup* yang bergerak dibidang *E-Commerce*. PT. Ouzen Anugerah Indonesia menjual berbagai macam produk kecantikan dan juga produk kesehatan seperti pembersih wajah, suplemen pemutih collagen, suplemen pelangsing, dll. Perusahaan ini mempromosikan produknya melalui berbagai macam *social media* seperti *Facebook*, *Instagram*, dan *Google*. Untuk satu orang *beauty consultant* dalam sehari mereka bisa mendapatkan 100 *lead* baru melalui iklan, semakin banyak *lead* masuk tentu saja akan semakin bagus karena *lead* tersebut bisa diprospek sehingga dapat menghasilkan keuntungan, tetapi tugas utama sebagai *beauty consultant* adalah memprioritaskan pelanggan, daripada dengan calon pelanggan. Hal ini tentu membuat *beauty consultant* menjadi kewalahan dan tidak efisien dalam membalas pertanyaan dari calon pelanggan baru sekaligus dengan melayani pelanggan lama setiap harinya. Maka peneliti merekomendasikan pengembangan aplikasi untuk mempermudah pelanggan dalam bertanya tentang produk OuzenMart, sehingga dengan adanya aplikasi ini calon pelanggan jadi lebih cepat mendapatkan informasi produk lebih cepat tanpa harus menunggu lama balasan dari *beauty consultant* OuzenMart.

### III.1.1. Analisa *Input*

Sebelum dibuatnya aplikasi layanan informasi pada PT. Ouzen Anugerah Indonesia, perusahaan ini masih menggunakan cara seperti biasa yaitu calon pelanggan mengirim pesan melalui *whatsapp* untuk tanya tanya mengenai produk yang di iklankan oleh team iklan. Berikut adalah salah satu contoh gambar tangkapan layar dari percakapan antara calon pelanggan dengan *beauty consultant* melalui *Whatsapp*.



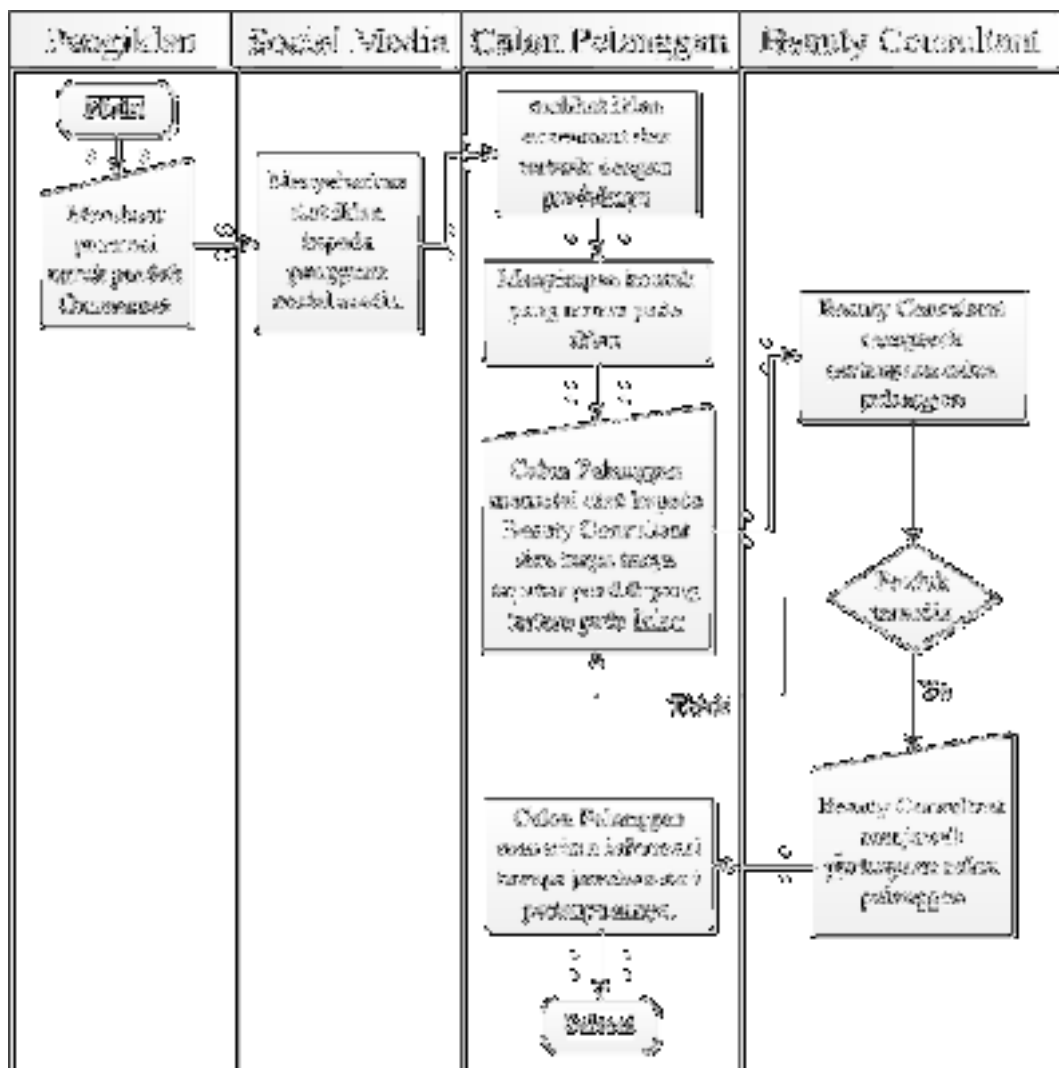
**Gambar III.1. Analisa *Input* Percakapan**

Pada gambar III.1 bisa dilihat calon pelanggan ada yang bertanya mengenai harga dari produk, dan ada yang ingin membeli produk. Selanjutnya *beauty consultant* akan menjawab pertanyaan dari calon pelanggan tersebut.

### III.1.2. Analisa Proses

Adapun analisa proses yang penulis tampilkan dapat dilihat pada Gambar

III.2 *Flow Of Diagram* berikut ini :



**Gambar III.2. *Flow Of Diagram* Analisa Proses**

Gambar III.2. menjelaskan FOD Analisa Proses dari perusahaan PT. Ouzen Anugerah Indonesia sebelum dibuatnya Aplikasi Layanan Informasi Penjualan pada PT. Ouzen Anugerah Indonesia Menggunakan API.AI berbasis Android.

### III.1.3. Analisa Output

Analisa *output* sebelum dibuat aplikasi layanan informasi penjualan pada PT. Ouzen Anugerah Indonesia adalah *beauty consultant* mampu menjawab pertanyaan dari calon pelanggan. Berikut adalah contoh gambar percakapan sebagai analisa output dari *beauty consultant*.



**Gambar III.3. Analisa Output Percakapan**

### III.1.4. Strategi Pemecahan Masalah

Berdasarkan evaluasi dari aplikasi yang berjalan, maka penulis memberika strategi pemecahan masalah sebagai berikut :

1. Membuat aplikasi yang memudahkan pelanggan mendapatkan informasi yang dijual serta membantu *beauty consultant* dalam menjawab pertanyaan yang sering diajukan oleh pelanggan

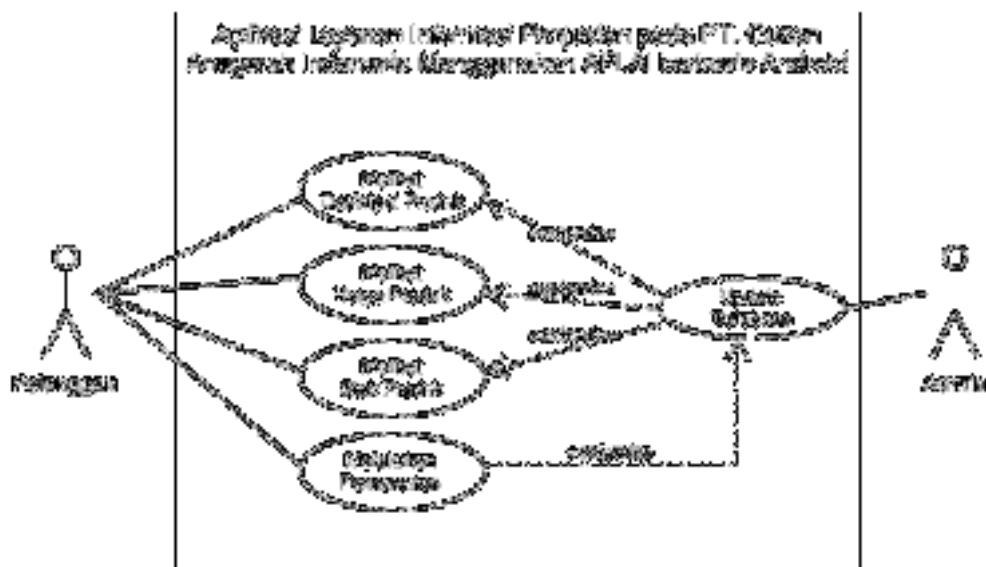
2. Merancang aplikasi layanan informasi penjualan menggunakan API.AI berbasis *Android*.

### III.2. Desain Sistem

Secara teori peneliti menggunakan pemodelan UML dan dengan menggunakan bahasa pemrograman *Java Android*, XML, HTML, PHP, *Javascript* dan *database Firebase* untuk memudahkan dalam perancangan aplikasi.

#### III.2.1. Use Case Diagram

Perancangan Aplikasi Layanan Informasi Penjualan pada PT. Ouzen Anugerah Indonesia Menggunakan API.AI berbasis Android menggunakan *Use Case Diagram* dapat dilihat pada gambar III.3 sebagai berikut :



Gambar III.4. *Use Case Diagram*

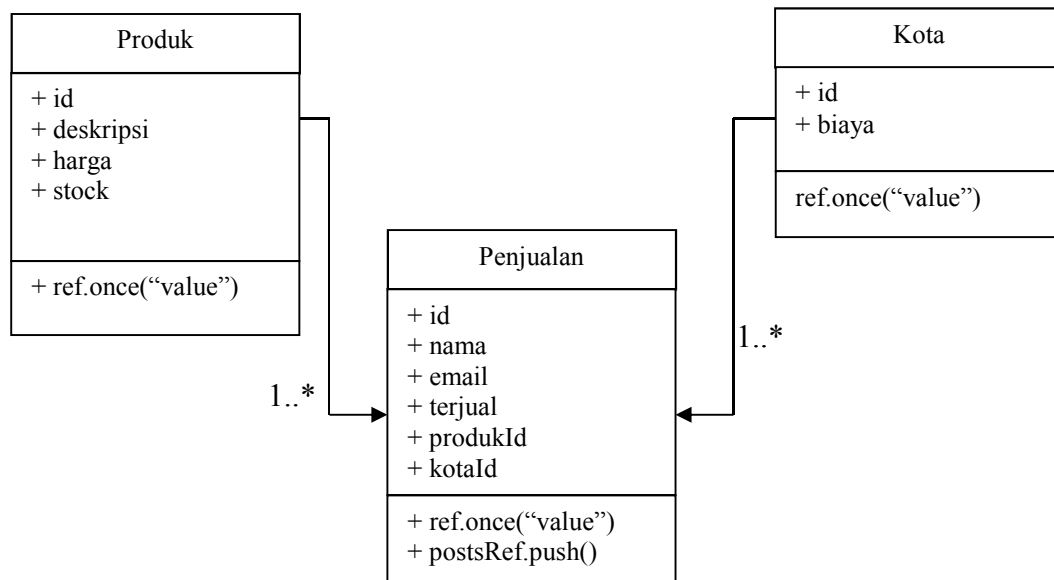
Berikut penjelasan dari *Usecase* Aplikasi Layanan Informasi Penjualan pada PT.

Ouzen Anugerah Indonesia Menggunakan API.AI berbasis Android :

1. Informasi mulai dari deskripsi, harga, dan stok dari produk akan di update oleh Admin secara manual melalui *website firebase database*.
2. Pelanggan yang menggunakan aplikasi ini harus mengetikkan sesuatu terlebih dahulu untuk mulai berinteraksi dengan aplikasi.
3. Setelah memulai chat dengan aplikasi, maka aplikasi akan membalas chat tersebut untuk memberikan panduan pelanggan fitur aplikasi

### III.2.2. *Class Diagram*

Perancangan Aplikasi Layanan Informasi Penjualan pada PT. Ouzen Anugerah Indonesia Menggunakan API.AI berbasis Android menggunakan *Class Diagram* dapat dilihat pada gambar III.5 sebagai berikut :



**Gambar III.5. *Class Diagram***

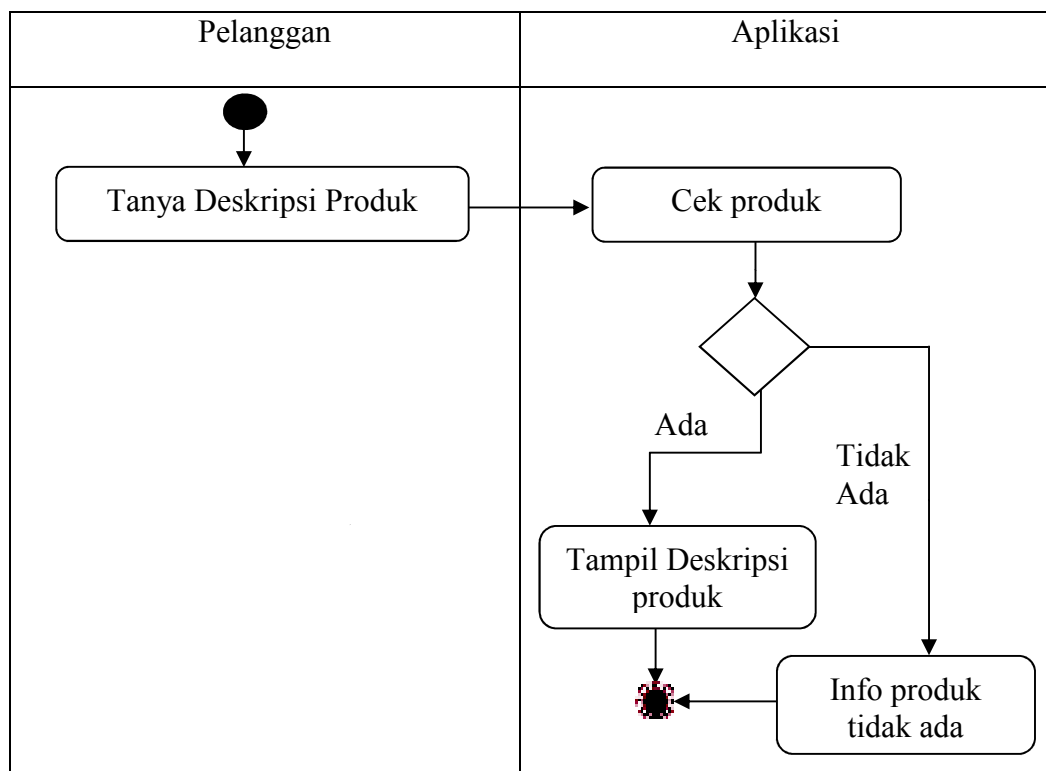
Berikut penjelasan dari *Class Diagram* :

1. Pada *table* Produk berisi *field* id, deskripsi, harga, serta *stock* mengenai produk yang dijual oleh OuzenMart.
2. Pada *table* Kota berisi *field* id serta biaya ongkos kirim untuk beberapa kota besar di Indonesia.
3. Pada *table* Penjualan adalah *table* relasi antara produk dan kota saat calon pelanggan melakukan pemesanan pada Aplikasi.

### III.2.3. Activity Diagram

Berikut ini adalah *activity diagram* Aplikasi Layanan Informasi Penjualan pada PT. Ouzen Anugerah Indonesia Menggunakan API.AI berbasis Android sebagai berikut :

#### III.2.3.1. Activity Diagram Tanya Deskripsi Produk

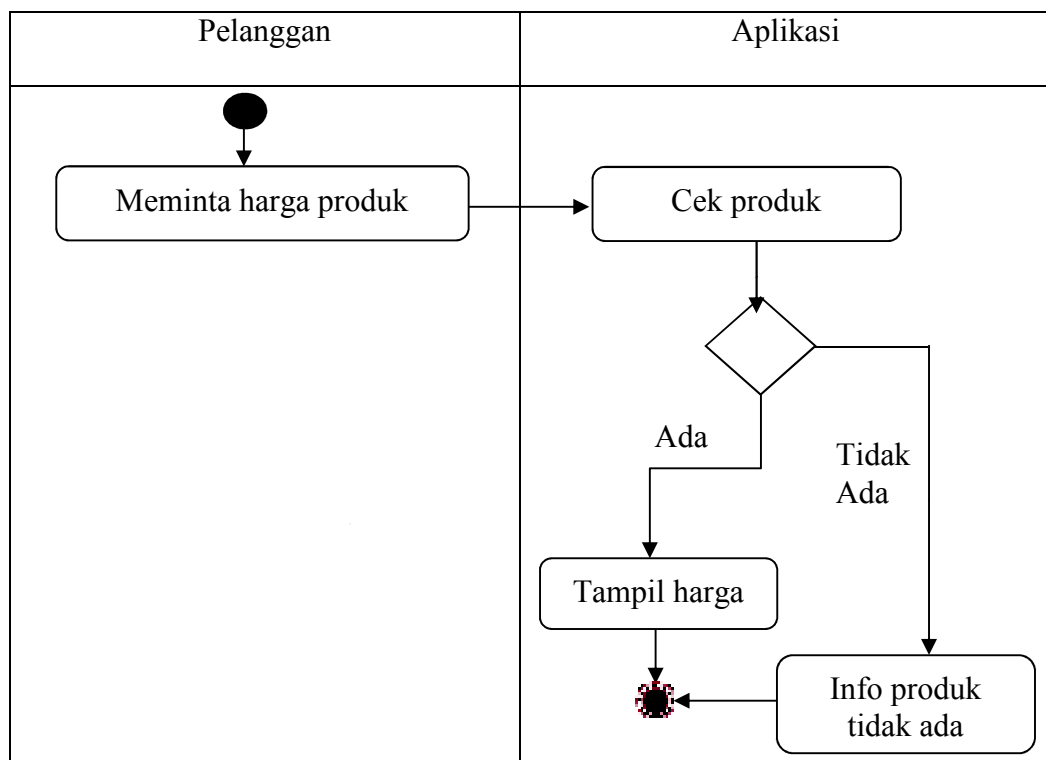


**Gambar III.6. Activity Diagram Tanya Deskripsi Produk**

Berikut penjelasan dari *Activity Diagram* Tanya Deskripsi Produk :

1. Pelanggan akan mengakses aplikasi yang ada di *smartphone* mereka
2. Pelanggan akan dibawa ke tampilan percakapan dan mulai meminta penjelasan dari produk yang ada di OuzenMart
3. Aplikasi akan menanyakan kembali produk apa yang pelanggan minta untuk dijelaskan, jika produk tersebut tersedia di database maka aplikasi akan menampilkan deskripsi produk tersebut kepada pelanggan.

### III.2.3.2. *Activity Diagram* Tanya Harga Produk



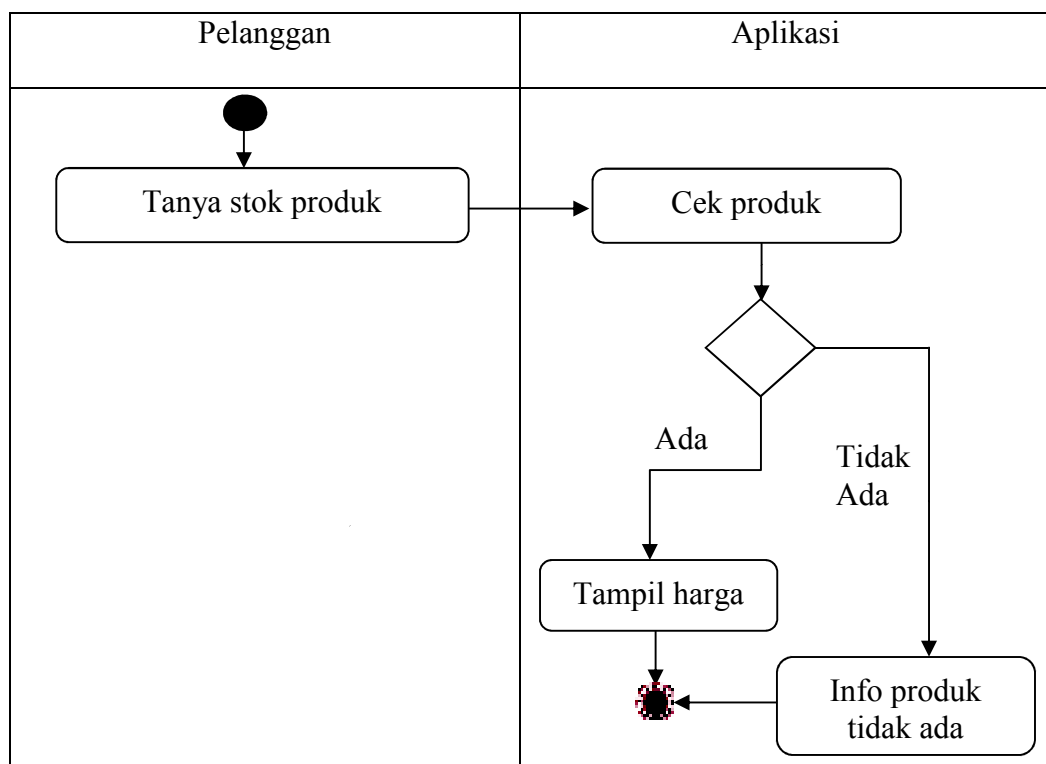
**Gambar III.7. *Activity Diagram* Tanya Harga produk**

Berikut penjelasan dari *Activity Diagram* Tanya Harga Produk :

1. Pelanggan akan mengakses aplikasi yang ada di *smartphone* mereka

2. Pelanggan akan dibawa ke tampilan percakapan dan mulai meminta keterangan harga dari produk yang ada di OuzenMart
3. Aplikasi akan menanyakan kembali produk apa yang pelanggan minta untuk dijelaskan, jika produk tersebut tersedia di database maka aplikasi akan menampilkan harga produk tersebut kepada pelanggan.

### III.2.3.3. *Activity Diagram* Tanya Stok Produk



**Gambar III.8. *Activity Diagram* Tanya Stok Produk**

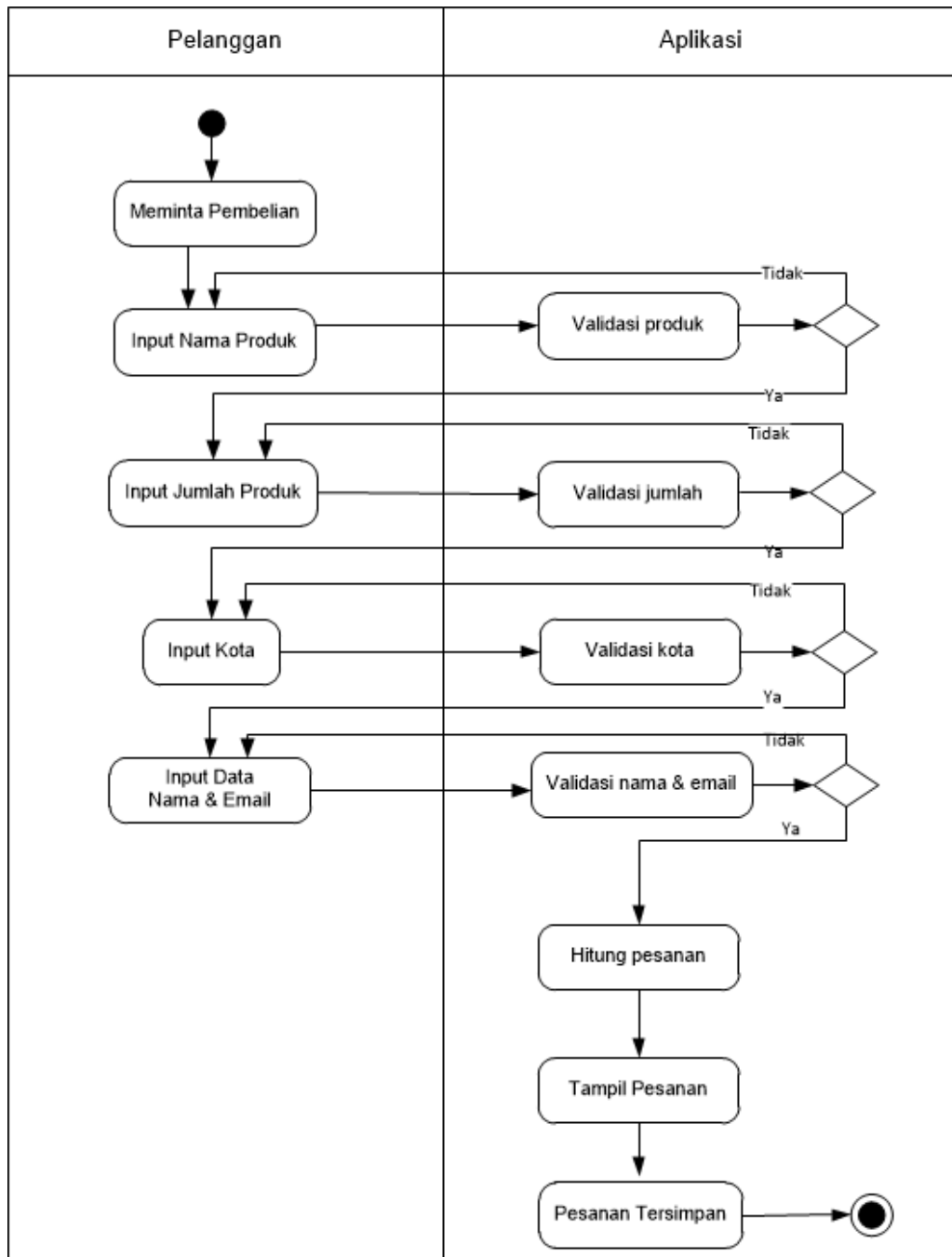
Berikut penjelasan dari *Activity Diagram* Tanya Stok Produk :

1. Pelanggan akan mengakses aplikasi yang ada di *smartphone* mereka
2. Pelanggan akan dibawa ke tampilan percakapan dan mulai tanya stok dari produk yang ada di OuzenMart

3. Aplikasi akan menanyakan kembali produk apa yang pelanggan minta untuk dijelaskan, jika produk tersebut tersedia di database maka aplikasi akan menampilkan stok produk tersebut kepada pelanggan.

#### **III.2.3.4. *Activity Diagram* Pemesanan**

*Activity Diagram* Pemesanan untuk melakukan pengolahan data pemesanan yang dapat dilihat seperti pada gambar III.9.



**Gambar III.9. Activity Diagram Pemesanan**

Berikut penjelasan dari *Activity Diagram* Pemesanan :

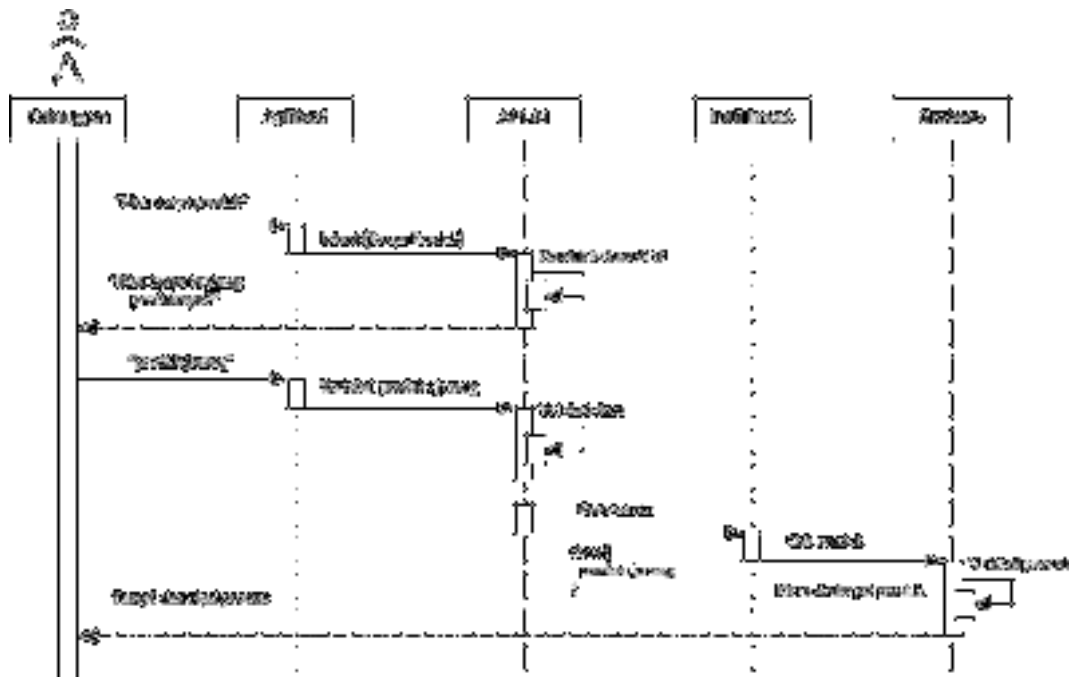
1. Setelah pelanggan membuka aplikasi dan berada di tampilan awal, pelanggan akan melakukan pemesanan produk dengan melakukan chat

2. Aplikasi akan meminta pelanggan untuk memasukkan produk terlebih dahulu, kemudian memvalidasi input tersebut cocok atau tidak
3. Jika tidak cocok aplikasi akan mengirimkan kembali pertanyaan kepada pelanggan untuk memasukkan produk. Jika cocok aplikasi akan meminta jumlah dari produk yang ingin dibeli pelanggan.
4. Jika jumlah yang diminta melebihi stok yang ada, maka aplikasi akan mengirim pesan bahwa stok tidak cukup. Jika jumlah yang diminta kurang dari stok maka aplikasi akan melanjutkan untuk meminta pelanggan memasukkan kota untuk menghitung ongkos kirim.
5. Jika kota yang dimasukkan pelanggan tidak cocok dengan database, maka aplikasi akan mengirimkan pesan bahwa kota yang dimasukkan tidak ada pada database. Jika kota yang dimasukkan tersedia, aplikasi akan melanjutkan dengan meminta pelanggan untuk memasukkan nama & email.
6. Jika validasi berhasil, aplikasi akan menghitung keseluruhan dari total biaya ongkir, harga produk, dan jumlah produk kemudian mengirimkan informasi pemesanan kepada pelanggan.

#### **III.2.4. *Sequence Diagram***

Adapun bentuk rancangan *sequence diagram* yang penulis rancang adalah sebagai berikut :

### III.2.4.1. *Sequence Diagram* Tanya Deskripsi Produk

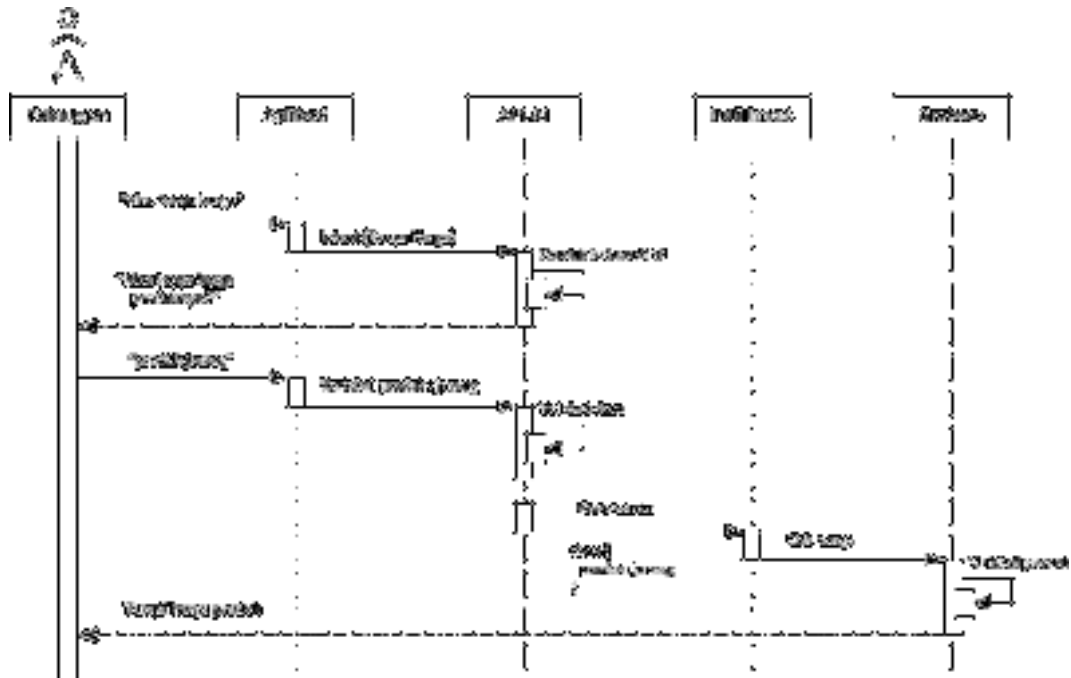


**Gambar III.10.** *Sequence Diagram* Tanya Deskripsi Produk

Berikut penjelasan dari *Sequence Diagram* Tanya Deskripsi Produk :

1. Pelanggan tanya produk kepada aplikasi tentang produk, jika pelanggan tidak menyebutkan nama produk. Maka aplikasi akan bertanya untuk meminta pelanggan menyertakan nama produk yang ingin ditanya
2. Pelanggan memberitahu nama produk yang ingin ditanya, aplikasi kemudian memproses data tersebut ke *API.AI* untuk di cek, jika ada maka aplikasi akan mengirimkan deskripsi produk tersebut kepada pelanggan

### III.2.4.2. *Sequence Diagram Tanya Harga Produk*

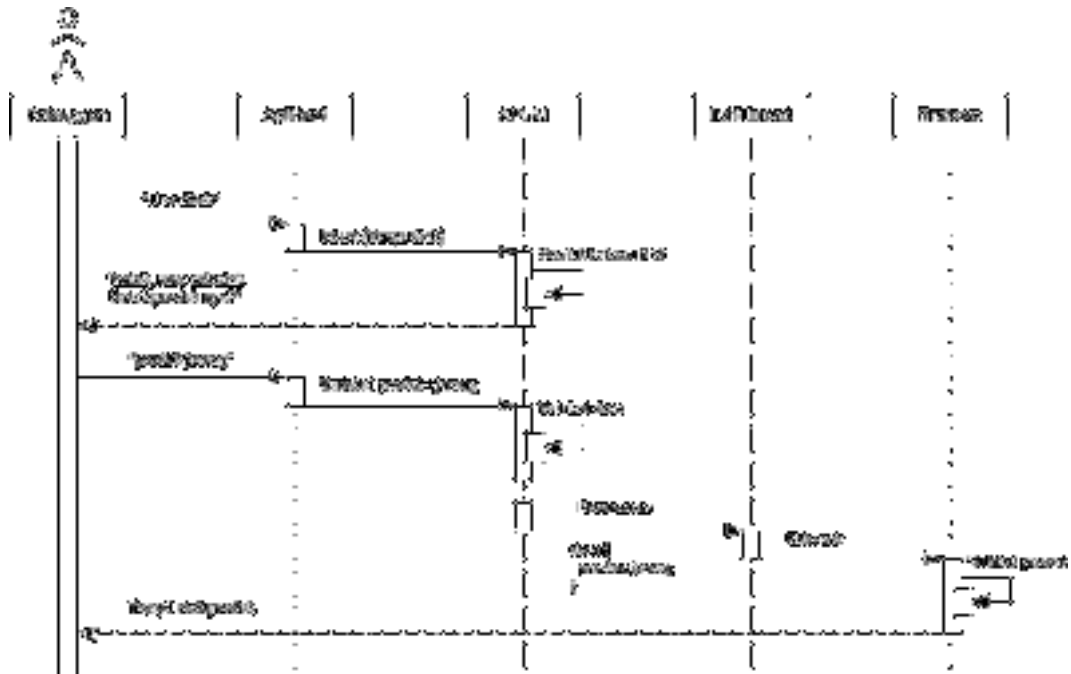


**Gambar III.11. *Sequence Diagram Tanya Harga Produk***

Berikut penjelasan dari *Sequence Diagram Tanya Harga Produk* :

1. Pelanggan tanya harga produk kepada aplikasi tentang produk, jika pelanggan tidak menyebutkan nama produk. Maka aplikasi akan bertanya untuk meminta pelanggan menyertakan nama produk yang ingin ditanya
2. Pelanggan memberitahu nama produk yang ingin ditanya, aplikasi kemudian memproses data tersebut ke *API.AI* untuk di cek, jika ada maka aplikasi akan mengirimkan harga produk tersebut kepada pelanggan

### III.2.4.3. *Sequence Diagram Tanya Stok Produk*

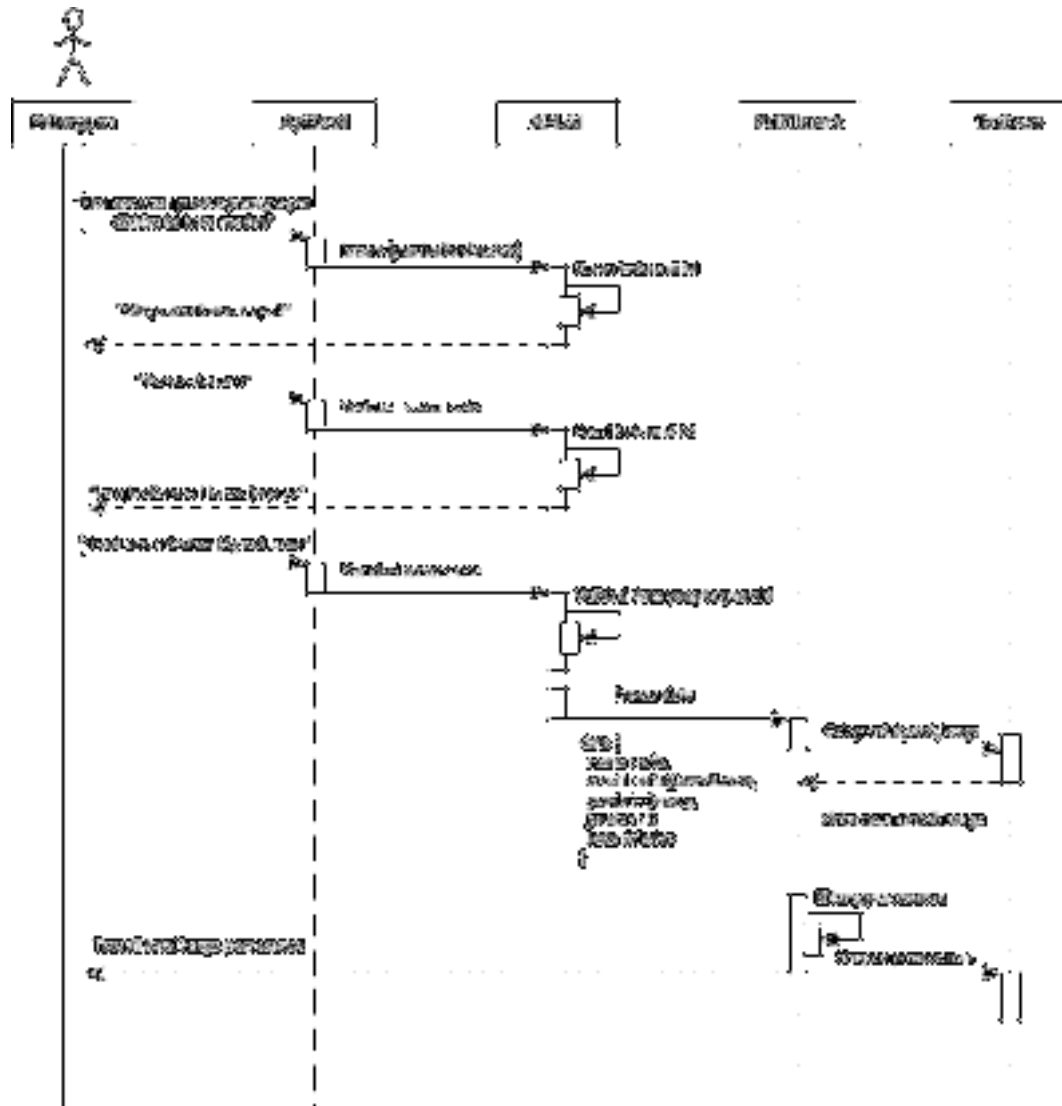


**Gambar III.12. *Sequence Diagram Tanya Stok Produk***

Berikut penjelasan dari *Sequence Diagram Tanya Stok Produk* :

1. Pelanggan tanya stok produk kepada aplikasi tentang produk, jika pelanggan tidak menyebutkan nama produk. Maka aplikasi akan bertanya untuk meminta pelanggan menyertakan nama produk yang ingin ditanya
2. Pelanggan memberitahu nama produk yang ingin ditanya, aplikasi kemudian memproses data tersebut ke *API.AI* untuk di cek, jika ada maka aplikasi akan mengirimkan jumlah stok produk tersebut kepada pelanggan

### III.2.4.3. *Sequence Diagram Pemesanan*



**Gambar III.13. *Sequence Diagram Pemesanan***

Berikut penjelasan dari *Sequence Diagram Pemesanan* :

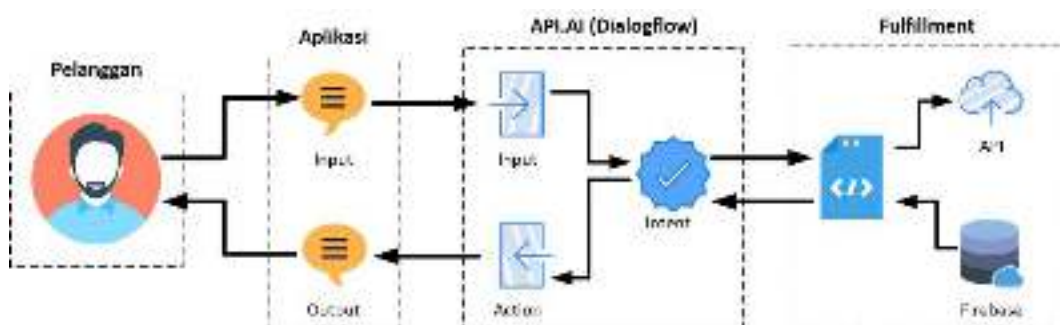
1. Pelanggan mengirim pesan kepada aplikasi untuk membeli produk jamug sebanyak 2 pcs dan dikirim ke kota Medan kepada aplikasi. Maka aplikasi

akan menerima data tersebut dan meneruskan ke *API.AI* untuk dicek apakah semua data sudah terisi.

2. Aplikasi mengharuskan pelanggan untuk mengisi juga Nama dan *Email* untuk melanjutkan pembelian tersebut. Saat semua data sudah lengkap, *API.AI* akan memvalidasi data tersebut dan mengirimkan ke sebuah *Webhook (Fulfillment)*.
3. Dari *webhook* kemudian data tersebut dicek ke *Firestore Database* apakah data yang dikirim ada di database atau tidak. Jika ada maka *webhook* akan menghitung pemesanan dari pelanggan, dan pemesanan tersebut disimpan ke *database* penjualan pada *Firestore*.
4. Setelah itu aplikasi akan menampilkan total pemesanan yang dilakukan oleh pelanggan.

### III.3. Perancangan Struktur Aplikasi

Adapun bentuk rancangan struktur aplikasi yang penulis rancang adalah sebagai berikut :



**Gambar III.14. Perancangan Struktur Aplikasi**

Pada gambar III.14 adalah rancangan struktur dari Aplikasi Layanan Informasi Penjualan pada PT. Ouzen Anugerah Indonesia Menggunakan API.AI

berbasis Android. Rancangan diawali dengan memasukkan teks dari pengguna dalam hal ini adalah pelanggan yang kemudian *input* dari pelanggan ini diolah menggunakan *platform* API.AI dengan cara mengidentifikasi *input* dari pelanggan dengan *intent* yang sudah dibuat. Jika pelanggan memberikan pertanyaan tentang tanya stok produk, maka API.AI akan mengidentifikasi bahwa maksud dari pelanggan adalah *intent* tanya stok produk. Untuk *intent* yang memerlukan akses *database* misalnya cek stok produk, diperlukan penggunaan fitur *fulfillment* dari API.AI itu sendiri untuk mengkoneksikan ke firebase database. Dengan *fulfillment* ini juga penulis bisa membuat berbagai macam fungsi seperti pengecekan stok, serta perhitungan pesanan dari pelanggan.

### **III.3.1. Perancangan Alur Percakapan pada API.AI (*Dialogflow*)**

Perancangan alur percakapan ini dibuat untuk menjelaskan proses bagaimana terjadinya sebuah percakapan antara pelanggan dengan aplikasi. Untuk perancangan alur percakapan pada aplikasi ini, penulis memanfaatkan data penelitian dari percakapan yang terjadi pada perusahaan PT. Ouzen Anugerah Indonesia yaitu berupa pertanyaan – pertanyaan yang sering ditanyakan oleh pelanggan kepada *beauty consultant* OuzenMart.

Sebelum membuat alur percakapan, penulis sudah membuat terlebih dahulu akun API.AI (*Dialogflow*) dan membuat *agent* baru bernama OuzenMartFAQ. Setelah *agent* dibuat barulah penulis bisa melanjutkan untuk membuat *intent* baru untuk alur percakapan yang akan dibuat.

### III.3.1.1. Dialog Percakapan Tanya Deskripsi Produk

Alur percakapan tanya deskripsi produk ini terjadi ketika pelanggan ingin mengetahui apa kegunaan serta manfaat dari produk yang ingin ditanya. Dialog percakapan tanya deskripsi produk ini bisa dilihat sebagai berikut :

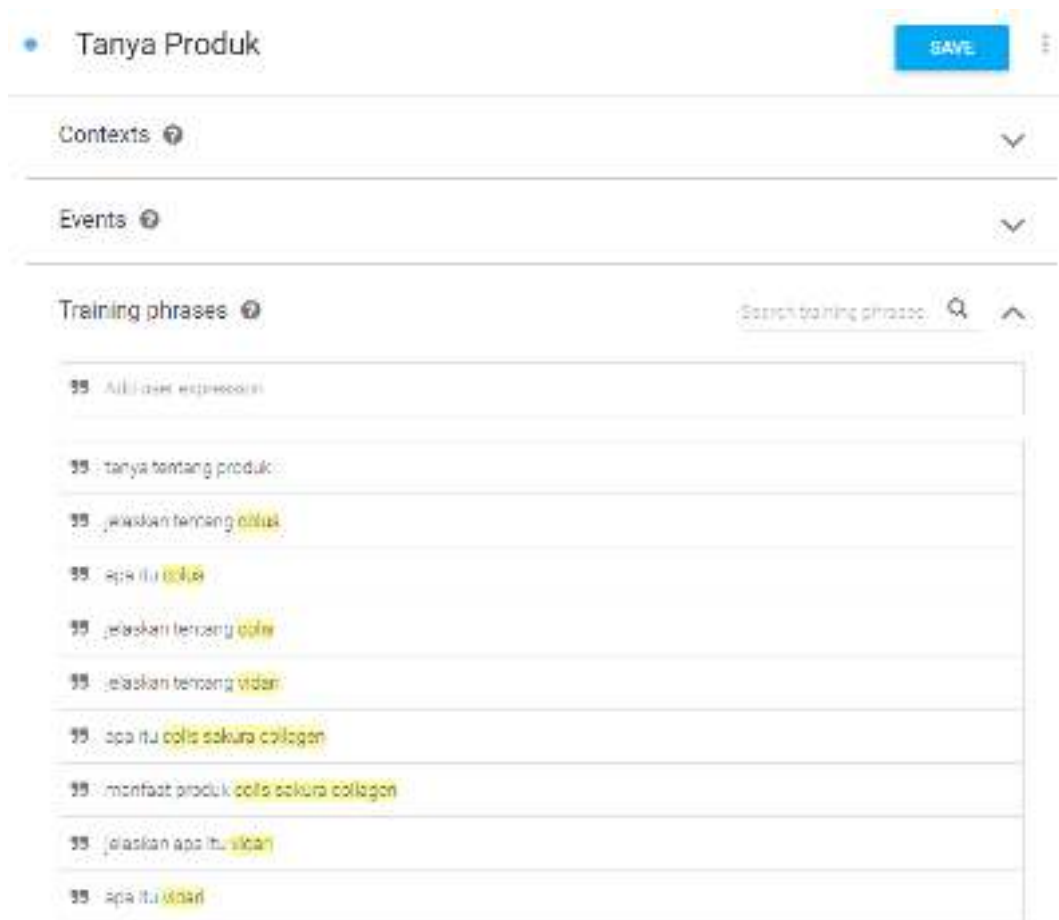
**Pelanggan** : “Mau tanya produk”

**Aplikasi** : “Mau tanya produk apa ?”

**Pelanggan** : “Produk jamug”

**Aplikasi** : “ Penjelasan produk, Jamug adalah suplemen dengan kandungan herbal yang membantu menambah nafsu makan dan berat badan”

Setelah membuat dialog percakapan tanya produk diatas, selanjutnya penulis mengimplementasi dialog percakapan tersebut ke sebuah *intent* pada API.AI seperti gambar III.15 berikut :



**Gambar III.15. Intent Tanya Produk**

Pada Gambar III.15 di atas terdapat *training phrases* yang berfungsi untuk melatih aplikasi menjadi pintar dalam memahami atau menangkap maksud pertanyaan yang diajukan oleh pelanggan. Pada *training phrases* tersebut juga terdapat sebuah *entity* “produk” yang berfungsi untuk memudahkan aplikasi dalam mengenal apa produk yang dimaksud dari pelanggan. Berikut ini merupakan pembuatan *entities* “produk” yang ditunjukkan pada Gambar III.16.

produk SAVE

Define synonyms  Allow automated expansion

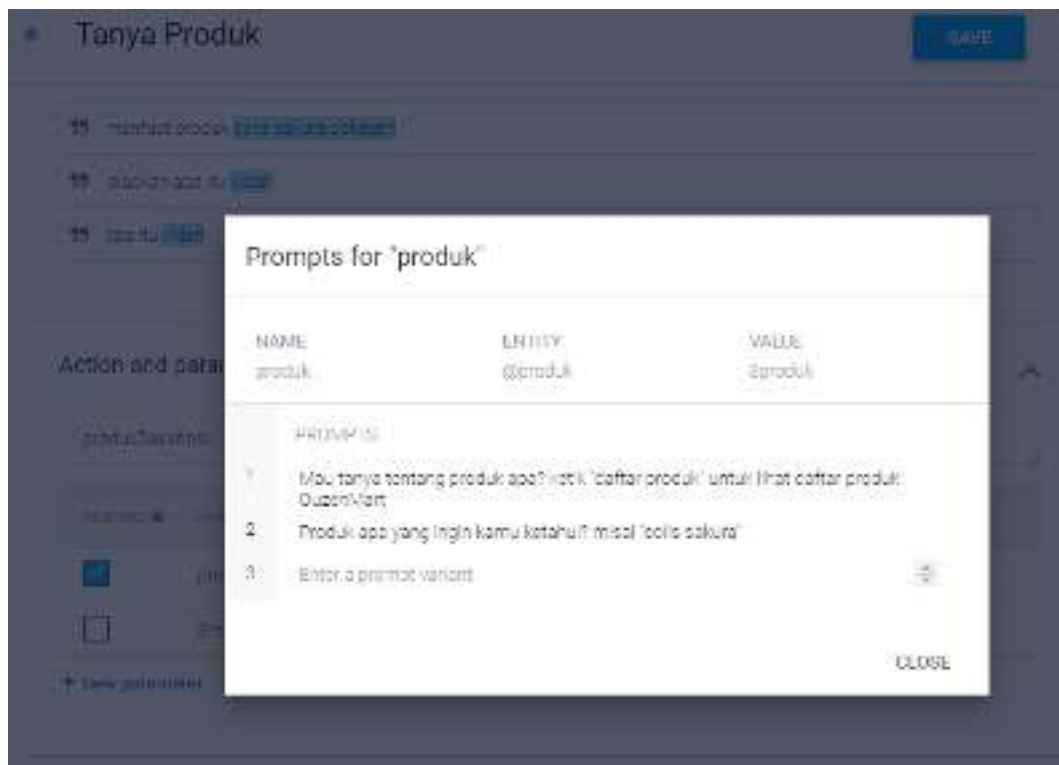
colls sakura collagen	colls sakura collagen, colls sakura, colls collagen, colls sakura besute, Colls Sakura Collagen
vidari	vidari, Midari
jamug	jamug, Jamug, jami
crestine	crestine, Urestine
colls men collagen	colls men collagen, Colls Men Collagen, colls collagen
colls julcy fibre	colls julcy fibre, colls Julcy, colls hare
colls white glow	colls white glow, colls loren, colls glow loren
colls tea tree	colls tea tree
colls whitening treatment	colls whitening treatment
colls	colls
colls	colls, colls :
colls men skincare	colls men skincare, colls men treatment
colls cake soap	colls cake soap, Colls Cake Soap, cake soap, colls cake

[Click here to edit entry](#)

**Gambar III.16. Entities Tanya Produk**

Pada Gambar III.16, tanda kotak biru yang penulis berikan adalah data pada *entities* “produk”, sedangkan tanda kotak merah adalah untuk mendefinisikan sinonim dari *data entity* disebelahnya.

Saat pelanggan tidak memasukkan nama produk, aplikasi akan bertanya kepada pelanggan mengenai produk yang dimaksud. Fitur ini disebut *slot filling*, berikut ini merupakan contoh penggunaan *slot filling* pada *intent* “tanya harga” yang bisa dilihat pada Gambar III.17 :



**Gambar III.17. Slot Filling pada Intent Tanya Produk**

Fitur *slot filling* terletak pada bagian *Action and parameters*, pada praktik yang penulis lakukan, penulis membuat *entity* "produk" menjadi *required* yang artinya *entity* tersebut harus diisi oleh pelanggan yaitu produk. Jika *entity* "produk" tersebut tidak diisi oleh pelanggan, maka aplikasi akan mengirimkan pertanyaan lanjutan mengenai *entity* tersebut.

### III.3.1.2. Dialog Percakapan Tanya Harga

Dialog percakapan tanya harga produk ini terjadi ketika pelanggan ingin mengetahui harga dari produk yang ingin ditanya. Dialog percakapan tanya harga produk ini bisa dilihat sebagai berikut :

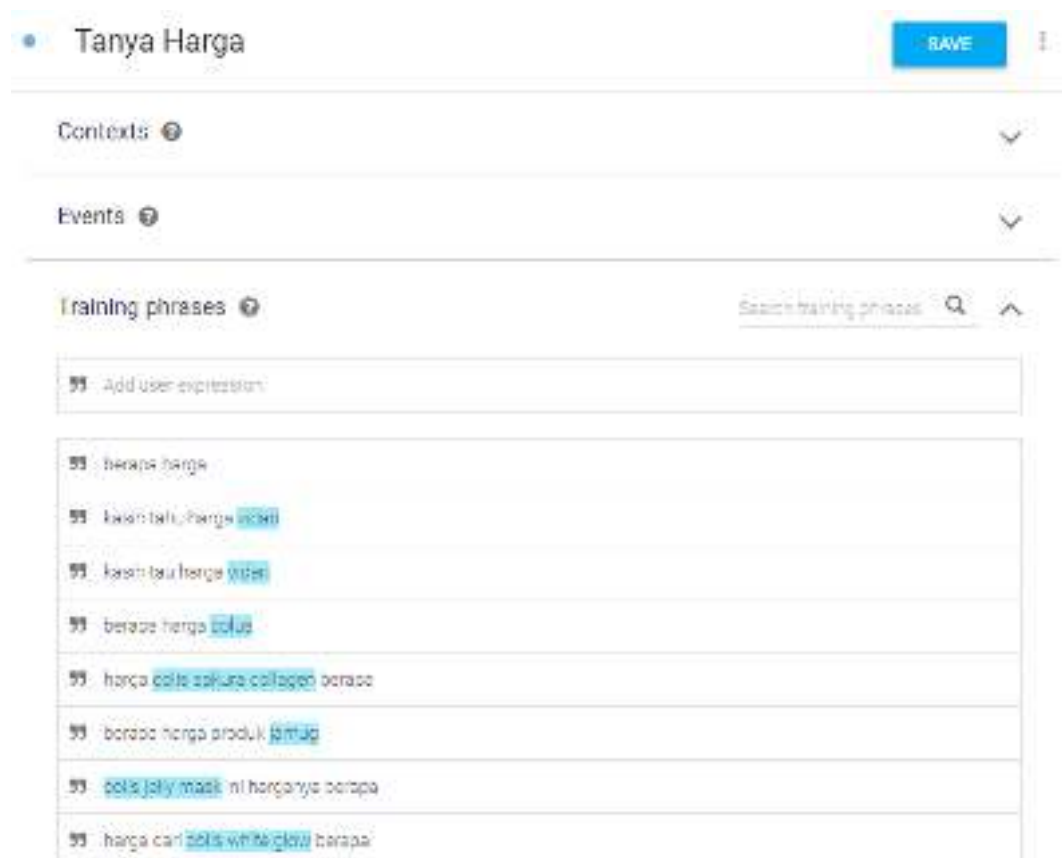
**Pelanggan** : “Mau tanya harga”

**Aplikasi** : “Mau tanya harga untuk produk apa ?”

**Pelanggan** : “Produk jamug”

**Aplikasi** : “ Harga dari produk jamug adalah Rp. 60.000,-”

Setelah membuat dialog percakapan tanya harga produk diatas, selanjutnya penulis mengimplementasi dialog percakapan tersebut ke sebuah *intent* pada API.AI seperti gambar III.18 berikut :



**Gambar III.18. Intent Tanya Harga**

Pada bagian *training phrases* ini terdapat kalimat yang berfungsi untuk melatih aplikasi memahami maksud dari pertanyaan pelanggan mengenai tanya harga produk. Disini juga terdapat *entity* “produk” yang membantu aplikasi mengenal produk yang dimaksud pelanggan. Penulis juga membuat *entity* “produk” menjadi *required* sehingga pelanggan harus mengisi *entity* produk untuk melakukan cek stok produk. Untuk melihat detail *entity* produk yang penulis gunakan untuk intent ini, bisa dilihat pada Gambar III.16.

### III.3.1.3. Dialog Percakapan Tanya Stok

Dialog percakapan tanya stok produk ini terjadi ketika pelanggan ingin mengetahui jumlah stok dari produk yang ingin ditanya. Dialog percakapan tanya stok produk ini bisa dilihat sebagai berikut :

**Pelanggan** : “Cek stok”

**Aplikasi** : “Mau cek stok produk apa ?”

**Pelanggan** : “Produk jamug”

**Aplikasi** : “Stok Jamug tersisa 10 lagi”

Setelah membuat dialog percakapan tanya harga produk diatas, selanjutnya penulis mengimplementasi dialog percakapan tersebut ke sebuah *intent* pada API.AI seperti gambar III.19 berikut :



**Gambar III.19. Intent Tanya Stok**

Pada bagian *training phrases* ini terdapat kalimat yang berfungsi untuk melatih aplikasi memahami maksud dari pertanyaan pelanggan mengenai tanya stok produk. Disini juga terdapat *entity* “produk” yang membantu aplikasi mengenal produk yang dimaksud pelanggan. Penulis juga membuat *entity* “produk” menjadi *required* sehingga pelanggan harus mengisi *entity* produk untuk melakukan cek stok produk. Untuk melihat detail *entity* produk yang penulis gunakan untuk intent ini, bisa dilihat pada Gambar III.16.

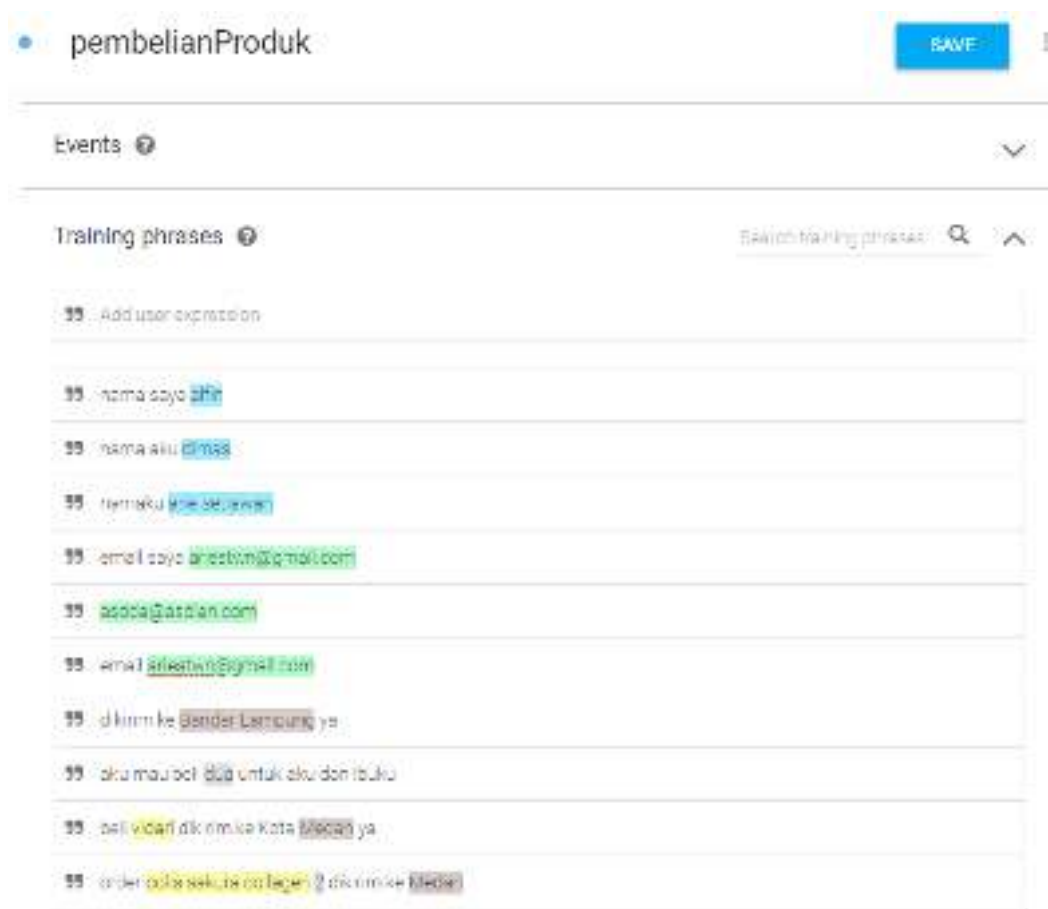
#### **III.3.1.4. Dialog Percakapan Pemesanan**

Dialog percakapan pemesanan ini terjadi ketika pelanggan ingin mengetahui total biaya keseluruhan dari harga produk dan ongkos kirim. Dialog percakapan pemesanan ini bisa dilihat sebagai berikut :

**Pelanggan** : “Aku mau beli jamug 2”

**Aplikasi** : “Oke, ini produknya mau dikirim kemana ?”  
**Pelanggan** : “ke Medan”  
**Aplikasi** : “Oke, nama kamu siapa ?”  
**Pelanggan** : “Nama aku Arie”  
**Aplikasi** : “Oke, nama kamu siapa ?”  
**Pelanggan** : “Emailnya ariestwnblog@gmail.com”  
**Aplikasi** : “Siiap! Kamu membeli jamug sebanyak 2 pcs, harganya Rp.120.000,-. Ditambah ongkir ke Medan, total harganya Rp.145.000,-. Ini nomor pemesanannya : -xxxxxx, Silahkan hubungi nomor WA beauty consultant kami untuk pemesanan lebih lanjut : 08xxxxxxxxx”

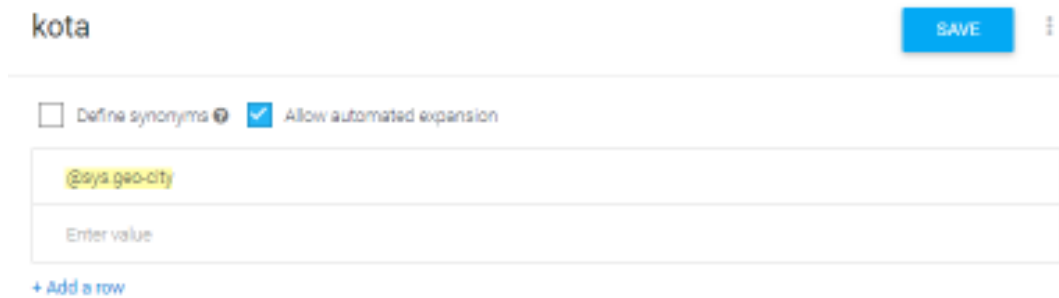
Semakin lengkap *entity* yang diberikan oleh pelanggan kepada aplikasi, maka aplikasi akan langsung memberikan hasil yang diinginkan oleh pelanggan. Jika *entity* yang diberikan oleh pelanggan kurang lengkap, maka aplikasi akan terus meminta pelanggan untuk melengkapi *entity* yang kurang tersebut. Berikut ini adalah *intent* pemesanan yang bisa dilihat pada Gambar III.20 :



**Gambar III.20. Intent Pemesanan**

Pada *intent* pemesanan ini, penulis menambahkan *parameter* `@sys.geocity` sebagai *entity* “kota”, *parameter* `@sys.number-integer` sebagai *entity* “jumlahProduk”, *parameter* `@sys.given-name` dan `@sys.last-name` sebagai *entity* “nama”, dan terakhir *parameter* `@sys.email` sebagai *entity* “email”. *Entity* “kota” ini berfungsi untuk membantu aplikasi dalam memahami nama kota di Indonesia, dan *entity* “jumlahProduk” berfungsi untuk membantu aplikasi dalam memahami jumlah produk yang ingin dibeli oleh pelanggan. Sedangkan untuk *entity* “nama” dan *entity* “email” adalah *entity* tambahan yang berfungsi untuk membantu aplikasi mengenal siapa yang telah melakukan pemesanan tersebut. Berikut ini

merupakan pembuatan *entity* “kota”, *entity* “jumlahProduk”, *entity* “nama”, dan juga *entity* “email” pada API.AI yang bisa dilihat pada Gambar III.21 , Gambar III.22, Gambar III.23, dan Gambar III.24 dibawah ini :



kota

SAVE

Define synonyms  Allow automated expansion

@sys.geo-city

Enter value

+ Add a row

**Gambar III.21 Entity kota**



jumlahProduk

SAVE

Define synonyms  Allow automated expansion

@sys.number.integer

Enter value

+ Add a row

**Gambar III.22 Entity jumlahProduk**

nama SAVE

---

Define synonyms  Allow automated expansion

@sys.given-name
@sys.last-name
@sys.given-name @sys.last-name
@sys.last-name @sys.given-name
Enter value

+ Add a row

**Gambar III.23 Entity Nama**

email SAVE

---

Define synonyms  Allow automated expansion

@sys.email
Enter value

+ Add a row

**Gambar III.24 Entity Email**

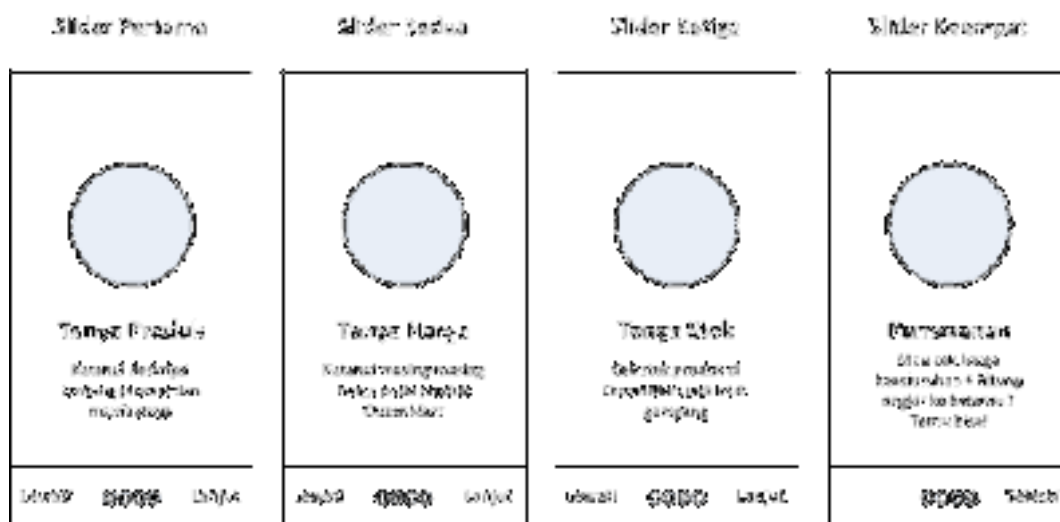
Pada *entity* nama, penggabungan antara *parameter* @sys.given-name dan @sys.last-name adalah untuk melatih aplikasi dalam memahami nama pelanggan baik jika pelanggan hanya memberikan nama depan saja, atau dengan memberikan nama depan dan nama belakang.

### III.4. Perancangan *User Interface*

User interface merupakan hal yang sangat penting bagi pelanggan aplikasi guna lebih mempermudah pelanggan mendapatkan informasi yang diinginkan sesuai dengan perintah yang diberikan kepada aplikasi.

#### III.4.1. Rancangan *intro slider*

Rancangan *intro slider* saat aplikasi pertama kali dibuka oleh pelanggan dapat dilihat pada gambar III.25. sebagai berikut :



**Gambar III.25. Rancangan *Intro Slider***

Keterangan bagian – bagian rancangan pada *intro slider* dapat dituliskan :

1. Pada masing – masing *slider* terdapat gambar, judul, dan penjelasan yang berguna untuk memberitahu pelanggan kegunaan dari aplikasi ini
2. Terdapat tombol lewati yang berfungsi untuk langsung menuju ke tampilan percakapan
3. Terdapat tombol lanjut yang berfungsi untuk melihat *slider* selanjutnya

4. Terdapat tombol selesai yang hanya aktif ketika pelanggan telah mencapai *slider* terakhir

### III.4.2. Rancangan Percakapan Awal



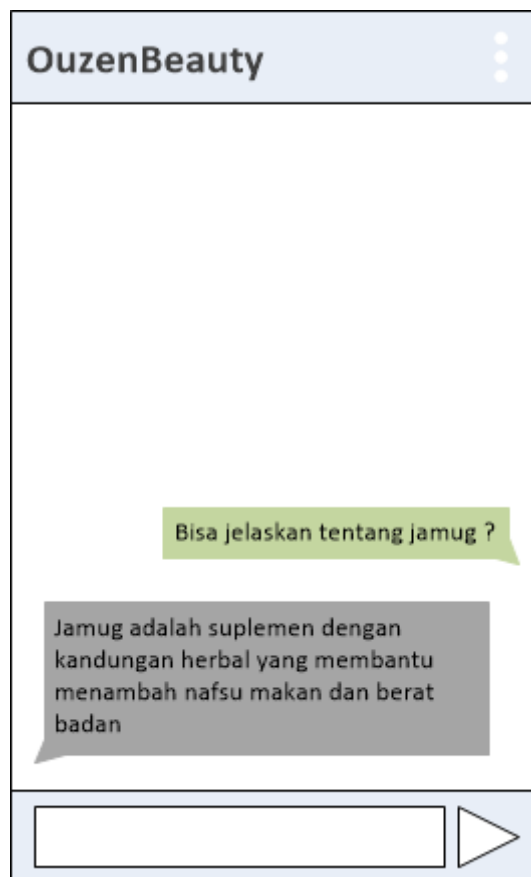
**Gambar III.26. Rancangan Percakapan Awal**

Keterangan bagian – bagian rancangan pada percakapan awal dapat dituliskan :

1. Saat pelanggan mengetikkan percakapan awal berupa “halo” maka aplikasi akan membalas percakapan tersebut berupa instruksi penggunaan aplikasi.
2. Tombol segitiga ke kanan berfungsi untuk mengirim pesan yang dituliskan pelanggan.

3. Untuk melihat tentang aplikasi, pelanggan bisa klik tombol titik tiga disebelah kanan kiri atas.

### III.4.3. Rancangan Percakapan Tanya Deskripsi Produk

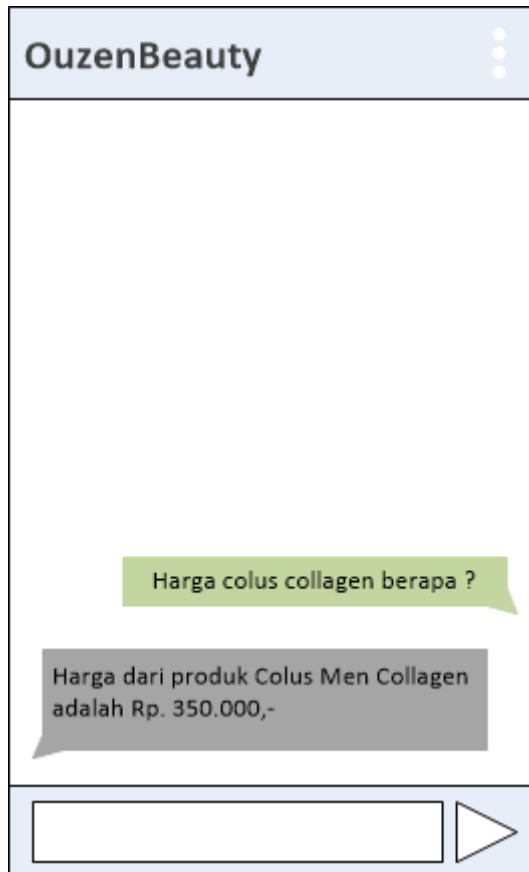


**Gambar III.27. Rancangan Percakapan Tanya Deskripsi Produk**

Keterangan bagian – bagian rancangan pada percakapan awal dapat dituliskan :

1. Saat pelanggan mengetikkan percakapan awal berupa “bisa jelaskan tentang produk” maka aplikasi akan menanyakan kembali produk apa yang ingin diketahui oleh pelanggan
2. jika diawal percakapan pelanggan langsung memberikan nama produk, aplikasi akan langsung menjawab dengan penjelasan tentang produk

#### III.4.4. Rancangan Percakapan Tanya Harga Produk

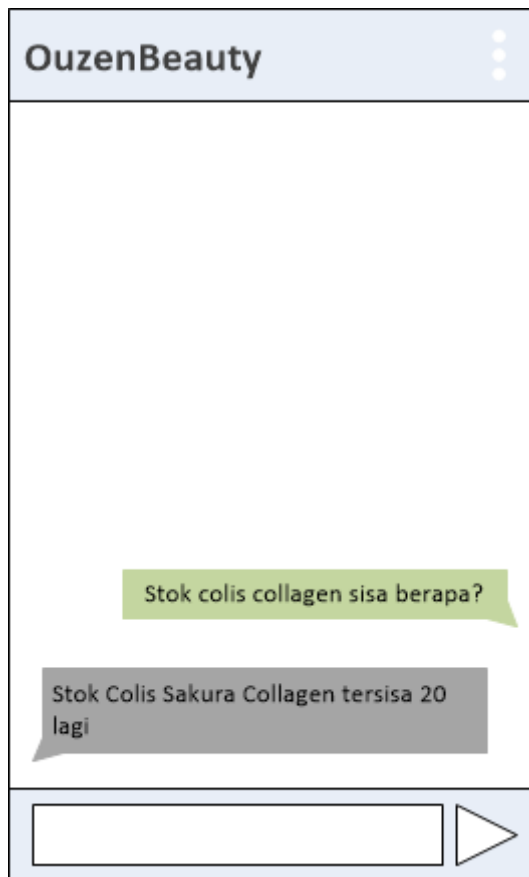


**Gambar III.28. Rancangan Percakapan Tanya Harga Produk**

Keterangan bagian – bagian rancangan pada percakapan awal dapat dituliskan :

1. Saat pelanggan mengetikkan percakapan awal berupa “mau tanya harga produk” maka aplikasi akan menanyakan kembali produk apa yang ingin diketahui oleh pelanggan
2. jika diawal percakapan pelanggan langsung memberikan nama produk, aplikasi akan langsung menjawab dengan harga dari produk yang ditanya

### III.4.5. Rancangan Percakapan Tanya Stok Produk



**Gambar III.29. Rancangan Percakapan Tanya Stok Produk**

Keterangan bagian – bagian rancangan pada percakapan awal dapat dituliskan :

1. Saat pelanggan mengetikkan percakapan awal berupa “cek stok produk” maka aplikasi akan menanyakan kembali produk apa yang ingin diketahui stoknya oleh pelanggan
2. jika diawal percakapan pelanggan langsung memberikan nama produk, aplikasi akan langsung menjawab dengan penjelasan tentang stok produk

### III.4.6. Rancangan Percakapan Pemesanan



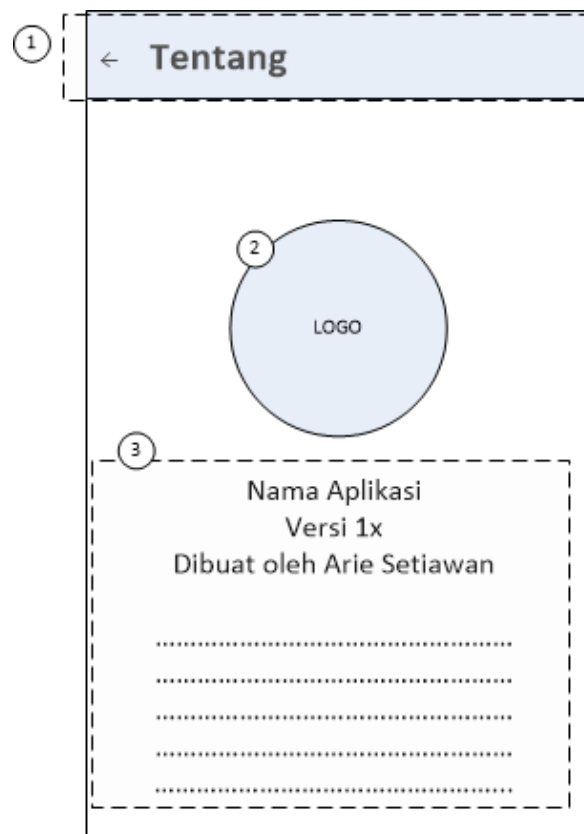
**Gambar III.30. Rancangan Percakapan Pemesanan**

Keterangan bagian – bagian rancangan pada percakapan awal dapat dituliskan :

1. Saat pelanggan mengetikkan percakapan awal berupa “aku mau beli” maka aplikasi akan menanyakan nama dan *email* yang nantinya data tersebut akan disimpan ke database penjualan.
2. Setelah pelanggan memberikan nama dan *email*, aplikasi akan menanyakan kembali produk apa yang ingin dibeli oleh pelanggan. Ketika pelanggan memberikan nama produk yang ingin dibeli, aplikasi akan lanjut menanyakan berapa jumlah produk yang ingin dibeli oleh pelanggan.

3. Kemudian saat pelanggan memberitahu berapa jumlah produk yang ingin dibeli, aplikasi akan bertanya kembali nama kota tempat pelanggan tinggal atau dimana produknya akan dikirim.
4. Setelah calon pelanggan memberikan nama kota, aplikasi akan mulai mengecek ketersediaan produk, stok, dan harga baru kemudian menghitung total pemesanan dari calon pelanggan

#### III.4.7. Rancangan Tentang



**Gambar III.31. Rancangan Tampilan Tentang**

Keterangan bagian – bagian rancangan tentang dapat dituliskan :

1. Tombol kembali
2. Logo aplikasi
3. Keterangan yang menjelaskan tentang dibuatnya aplikasi ini