

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisa

Proses analisa dilakukan untuk mengidentifikasi masalah yang terjadi di subjek penelitian yang nantinya akan dicari solusi pemecahan masalahnya.

3.1.1 Identifikasi Masalah

Identifikasi masalah merupakan salah satu langkah dalam menganalisis suatu sistem. Masalah dapat didefinisikan sebagai masalah yang diharapkan dapat dipecahkan yang membuat tujuan sistem tidak dapat dicapai.

Tabel 3.1 Permasalahan, Solusi, dan Manfaat.

No.	Permasalahan	Solusi	Manfaat
1	AI yang ada pada IDE hanya memberikan <i>auto-correct</i> , <i>auto-complete</i> , atau <i>code-suggestion</i> .	Membuat aplikasi yang dapat memberikan kode yang sesuai dengan perintah yang diberikan.	Programmer bisa mendapatkan kode yang lebih tepat dan sesuai dengan kebutuhan, meningkatkan efisiensi dalam pengembangan aplikasi.

			<p>Membantu programmer pemula, untuk belajar sintaks dan konsep pemrograman dengan memberikan contoh kode yang kontekstual. Mengurangi kemungkinan kesalahan dalam kode dengan memberikan saran yang lebih akurat dan relevan.</p>
2	Proses konversi dari design UI ke kode program memakan waktu cukup banyak	Membuat aplikasi yang dapat mengenali gambar dan menghasilkan kode program yang sesuai dengan gambar dan perintah yang	<p>Mempercepat proses pengembangan dengan mengurangi waktu yang dibutuhkan untuk mengonversi desain UI menjadi</p>

		diberikan untuk mempercepat proses pengembangan aplikasi	kode program.
3	Proses konversi dari <i>React Native</i> ke <i>Flutter</i> cukup memakan waktu	Membuat aplikasi yang dapat melakukan konversi kode tanpa menghilangkan fungsi aslinya.	Programmer dapat menghemat waktu cukup banyak karena proses konversi dibantu oleh AI

3.1.2 Pemecahan Masalah

Pada beberapa permasalahan yang diidentifikasi pada tabel 3.1 solusi yang dijabarkan akan diimplementasi pada aplikasi yang dikembangkan. Untuk menyelesaikan permasalahan 1 dan 3, pada aplikasi yang dikembangkan akan menggunakan bantuan *API* dari OpenAI agar *bot* dapat memberikan balasan yang kontekstual, sehingga programmer bisa mendapatkan kode program sesuai kebutuhan atau melakukan konversi kode dari bahasa pemrograman yang lain ke bahasa pemrograman yang dibutuhkan dengan lebih cepat. Untuk menyelesaikan permasalahan 2, pada aplikasi yang dikembangkan ada fitur untuk mengirim pesan teks dengan gambar, sehingga *bot* dapat menganalisa gambar dan teks yang dikirim secara bersamaan dan memberikan balasan yang sesuai untuk

pesan dan gambar yang dikirimkan.

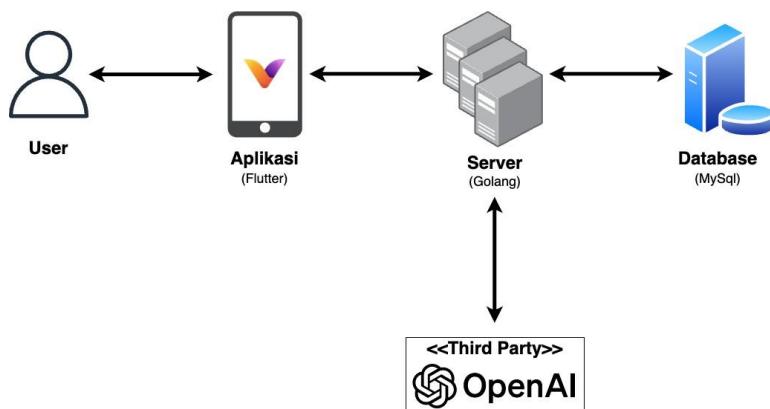
3.2 Perancangan

Perancangan sistem yang dilakukan saat pembuatan aplikasi *chatbot* “Vita” adalah sebagai berikut.

3.2.1 Perancangan Sistem

Dari hasil identifikasi masalah pada sistem yang berjalan dapat menjadi acuan untuk merancang aplikasi *chatbot* “Vita”.

3.2.1.1 Arsitektur Aplikasi

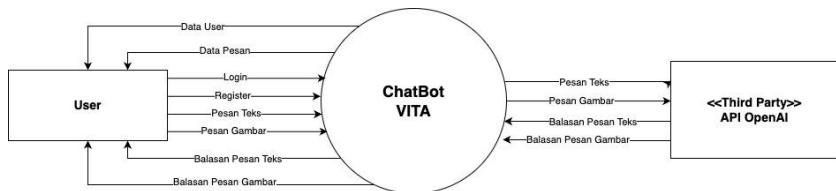


Gambar 3.1 Arsitektur Aplikasi

Pada arsitektur aplikasi digambarkan bagaimana sistem nantinya akan berjalan. Pada sistem yang akan berjalan, *user* dapat mengakses *chatbot* melalui aplikasi yang dikembangkan menggunakan *flutter* yang nantinya dapat diinstal pada perangkat *android* atau *iOS* *user*. Ketika *user*

mengirimkan pesan melalui aplikasi, maka aplikasi akan menggunakan *API* di server, dan server yang melakukan akses ke *third party* yaitu *API* dari *OpenAI* untuk mendapatkan balasan pesan dari *user*. Setelah mendapatkan balasan, server akan menyimpan data pesan yang dikirimkan oleh *user* ke dalam database lalu meneruskan balasan pesan ke aplikasi untuk ditampilkan ke *user*.

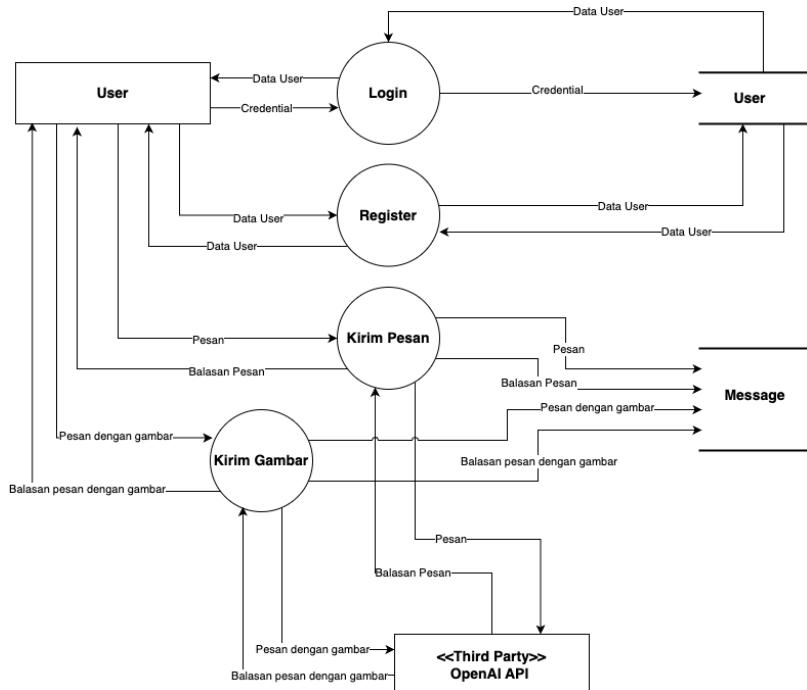
3.2.1.2 DFD Level 0



Gambar 3.2 DFD Level 0

Pada DFD Level 0 terdapat 2 entitas, satu entitas internal yaitu *user* dan satu entitas *external API* dari *OpenAI* yang mana *API* ini akan membantu dalam memberikan balasan dari pesan yang dikirimkan oleh *user*. Entitas *user* dapat melakukan *login* jika sudah melakukan pendaftaran akun, dan *user* dapat melakukan *register* apabila belum memiliki akun. Setelah *user login* atau *register* maka sistem akan memberikan data *user* dari *database* kepada *user* yang berhasil *login* atau *register*, dan *user* dapat mengirimkan pesan ke sistem. Setelah *user* mengirimkan pesan teks atau gambar, maka sistem akan meneruskan ke *OpenAI* untuk mendapatkan balasan pesan yang kontekstual, dan setelah itu balasan pesan akan langsung diteruskan ke *user* yang mengirimkan pesan tersebut.

3.2.1.3 DFD Level 1



Gambar 3.3 DFD Level 1

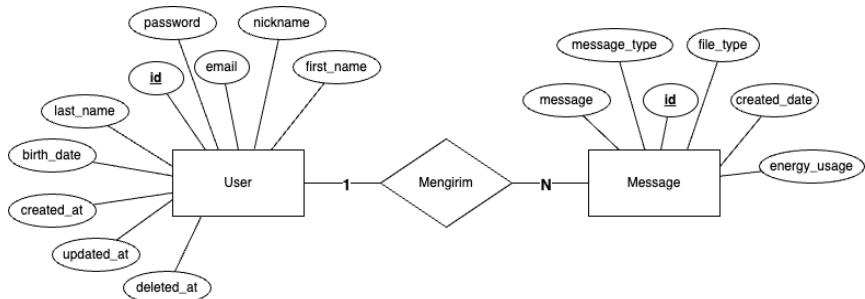
Pada DFD Level 1, teridentifikasi 2 entitas, yakni satu entitas internal berupa *user* dan satu entitas eksternal berupa *API* dari *OpenAI*, yang mana *API* ini akan berfungsi dalam menyediakan balasan dari pesan yang dikirimkan oleh *user*. Selain itu, terdapat penyimpanan informasi yaitu *user* dan *message*. Kemudian, terdapat pula 4 proses, yaitu *login*, *register*, *kirim pesan*, dan *kirim gambar*. Entitas *user* dapat melakukan proses *login*, dan *user* dapat melakukan *register* apabila belum

mendaftarkan akun. Ketika user melakukan proses *login* atau *register*, maka sistem akan meneruskan data ke penyimpanan data user untuk memastikan validitas data, dan apabila data terbukti sesuai, maka sistem akan memberikan data user kepada user yang akan *login* atau *register*. Setelah user *login* atau *register*, maka user dapat mengirimkan pesan teks atau gambar ke sistem. Saat user mengirimkan pesan teks ke sistem, maka sistem memanfaatkan bantuan *API* dari *OpenAI* untuk memperoleh balasan yang sesuai. Setelah tahapan tersebut, sistem menyimpan pesan ke database message dan melanjutkan dengan meneruskan balasan ke user. Pada saat user mengirimkan pesan gambar ke sistem, maka sistem pun memanfaatkan *API* dari *OpenAI* untuk memperoleh balasan yang sesuai, selanjutnya menyimpan pesan ke penyimpanan data message dan kemudian secara langsung meneruskan balasan ke user.

3.2.2 Perancangan Data

Berikut merupakan perancangan data dengan menggunakan *Entity Relationship Diagram* (ERD) untuk menggambarkan relasi antar data yang ada pada aplikasi.

3.2.2.1 Entity Relationship Diagram



Gambar 3.4 *Entity Relationship Diagram (ERD)*

Pada *Entity Relationship Diagram (ERD)* ini terdapat 2 Entitas yaitu *User* dan *Message*. *User* merupakan *user* yang terdaftar dan dapat menggunakan aplikasi *chatbot*. *Message* merupakan pesan yang dikirimkan oleh *user*.

3.2.3 Perancangan Aplikasi

Pada bagian perancangan aplikasi, akan ditunjukkan *User Interface* dari aplikasi yang dikembangkan. Pada perancangan aplikasi akan ditampilkan perancangan dari halaman *splash*, *login*, *register*, *chat*, kirim gambar, dan profil.

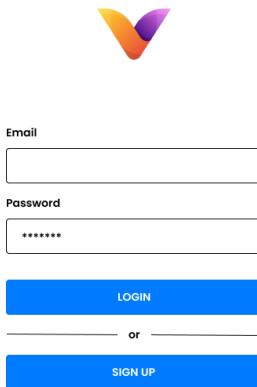
3.2.3.1 Perancangan Halaman *Splash*



Gambar 3.5 Halaman *Splash*

Pada halaman *splash*, hanya terdapat logo dari aplikasi, dan juga indikator *loading* karena pada halaman ini dilakukan proses inisiasi data. Sekaligus proses pengecekan *autentikasi user*, jika proses pengecekan *autentikasi user* berhasil dan *autentikasi* masih valid, maka user dianggap masih *login* dan akan diarahkan ke halaman *chat*. Jika proses pengecekan *autentikasi* gagal, maka data *autentikasi user* sebelumnya dihapus, dan user harus melakukan *login* kembali.

3.2.3.2 Perancangan Halaman Login



Gambar 3.6 Halaman Login

Pada halaman *login* terdapat dua *text field* yang pertama untuk *email* dan yang kedua untuk *password*, pada *text field email* dapat menerima *input* berupa teks, nomor, dan simbol, dan juga terdapat validasi apabila *text field* kosong, lalu apakah teks yang di-*input* oleh *user* sudah sesuai dengan format *email* atau belum. Sedangkan untuk *text field password*, *user* dapat melakukan *input* teks apapun, dan hanya terdapat validasi untuk mengecek apakah *text field password* kosong atau tidak.

3.2.3.3 Perancangan Halaman Register



Email

Nickname

First Name

Last Name

Birth Date
 

Password

Repeat Password

I have read and agree to Vita's [user agreement](#) and [privacy policy](#)

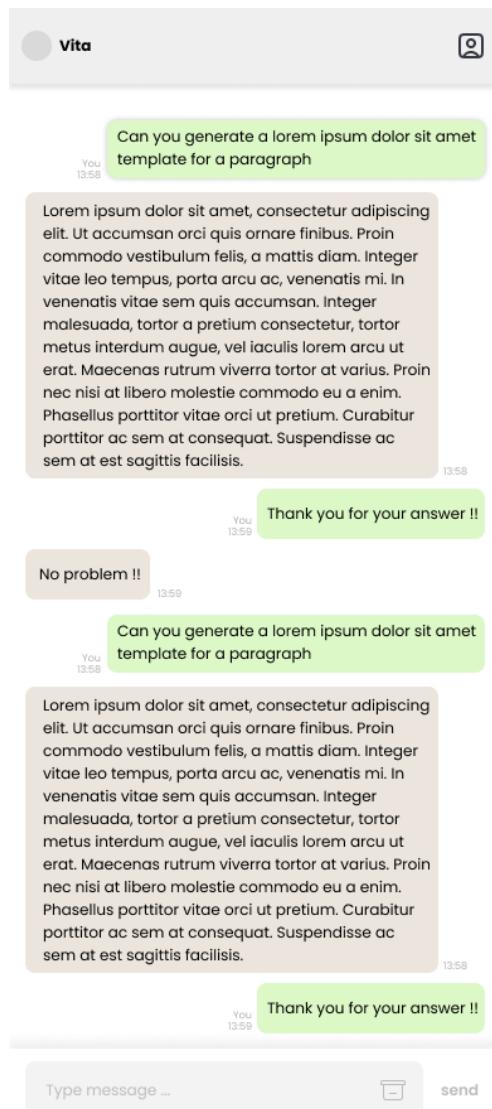
SIGN UP

Gambar 3.7 Halaman Register

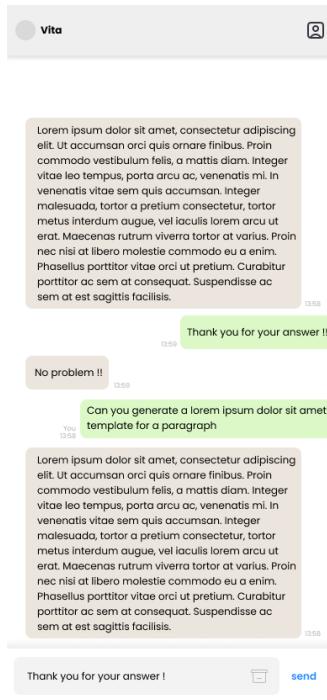
Pada halaman *register*, terdapat logo aplikasi di bagian atas. Lalu terdapat *text field email* yang menerima *input* berupa teks, nomor, dan simbol. Lalu pada *text field email* juga terdapat validasi untuk teks kosong dan jika format email tidak sesuai. Lalu terdapat *text field nickname* untuk *nickname* dari *user*, *text field nickname* ini hanya menerima *input* berupa teks, dan terdapat validasi apabila teks kosong dan jika *user* memberikan

input selain text. Lalu terdapat *text field* first name untuk nama depan dari user *text field* first name ini hanya menerima *input* berupa text, dan terdapat validasi apabila teks kosong dan jika *user* memberikan *input* selain text. Lalu terdapat *text field* last name untuk nama belakang dari user, *text field* last name ini hanya menerima *input* berupa teks, dan terdapat validasi apabila teks kosong dan jika user memberikan *input* selain teks. Lalu ada *birth date* untuk tanggal lahir dari *user*, jika *field* ini ditekan oleh *user*, maka akan menampilkan *dialog* untuk *date picker*. Lalu ada *text field* untuk *password* dan *confirm password*. Pada *text field password*, *user* dapat memberikan *input* berupa teks, nomor, dan simbol, dan terdapat validasi jika *text field* ini belum diisi. Lalu pada *text field confirm password* cukup sama dengan *text field password*, *text field* ini menerima *input* berupa teks, nomor, dan simbol, dan terdapat validasi apabila teks kosong atau jika teks tidak sama dengan *password*. Lalu juga terdapat tombol *register* untuk mengirim data yang sudah di-*input* oleh *user* ke server.

3.2.3.4 Perancangan Halaman Chat



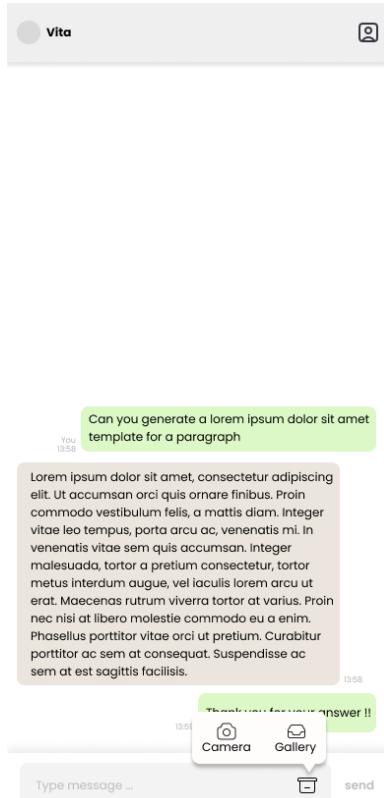
Gambar 3.8 Halaman Chat



Gambar 3.9 Halaman chat saat *textfield* diisi

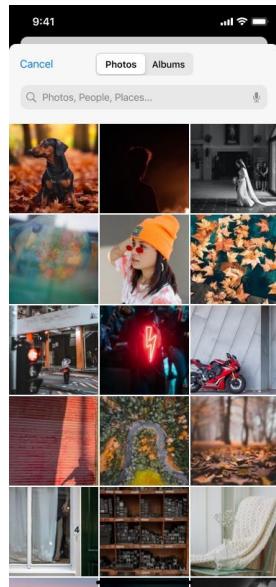
Setelah *user login*, maka *user* akan langsung diarahkan ke halaman *chat*. Pada bagian atas dari halaman *chat* terdapat nama aplikasi “Vita” dan juga di bagian kanan atas terdapat tombol untuk pindah ke halaman profil. Lalu terdapat daftar dari pesan yang pernah dikirimkan *user* ke *chatbot* sekaligus dengan tanggal pengirimannya. Lalu di bagian bawah terdapat *text-field* untuk mengirimkan pesan kepada *chatbot*. Jika *text-field* masih kosong maka tombol untuk mengirim pesan akan dinonaktifkan dan berwarna abu-abu, jika *text-field* sudah diisi maka tombol kirim pesan berubah menjadi berwarna biru. Dan pesan dapat dikirimkan.

3.2.3.5 Perancangan Halaman Kirim Gambar



Gambar 3.10 Memilih Sumber Gambar

Pada bagian ujung kanan dari *text-field* terdapat sebuah *icon* yang saat diklik akan menampilkan suatu dialog yang berisi pilihan untuk sumber dari gambar yang akan dikirim yaitu kamera dan galeri.



Gambar 3.11 Mengambil gambar dari galeri

Jika klik pada icon galeri maka *user* akan diarahkan ke halaman galeri dan user dapat memilih gambar mana yang akan dikirim.



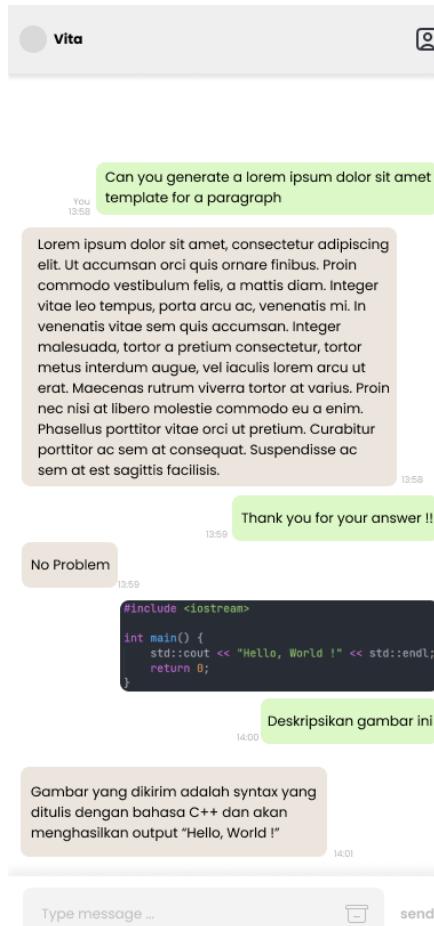
Gambar 3.12 Mengambil gambar dari kamera

Jika klik pada icon kamera maka *user* akan diarahkan ke halaman kamera dari ponselnya untuk dapat melakukan pengambilan gambar



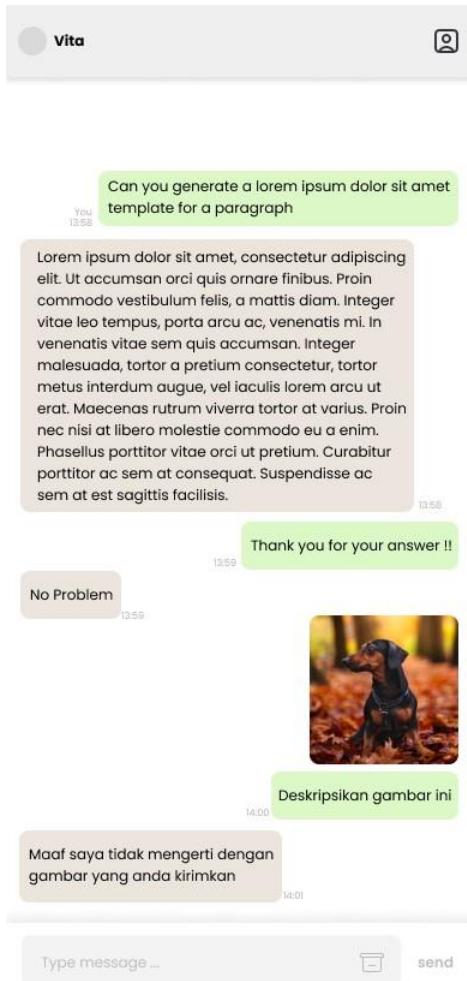
Gambar 3.13 Menambahkan pesan saat mengirim gambar

Setelah mengambil gambar atau memilih gambar. Maka user akan diarahkan ke halaman *preview* gambar, pada halaman ini terdapat *text-field* yang harus diisi pesan yang mendeskripsikan apa yang diperlukan oleh *user*. Dan terdapat tombol kirim pesan yang akan aktif setelah *user* mengisi *text-field*. Setelah mengirim gambar maka *user* akan diarahkan kembali ke halaman *chat*.



Gambar 3.14 Kirim Gambar Berhasil

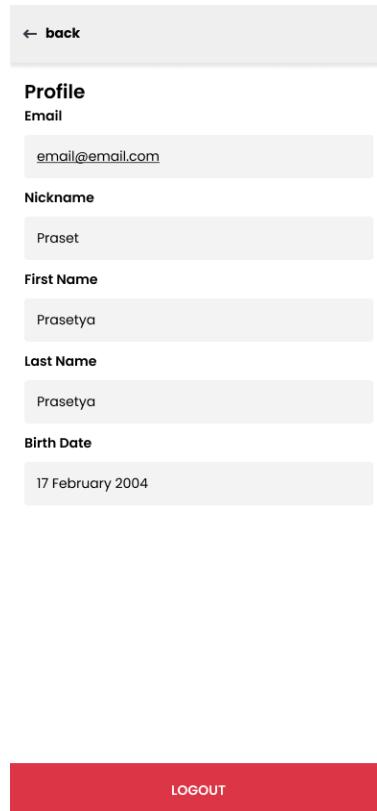
Ketika gambar yang dikirim sesuai dengan tujuan aplikasi yaitu untuk membantu *user* dalam pemrograman, maka *bot* akan merespon sesuai dengan gambar dan pesan yang dikirimkan oleh *user*.



Gambar 3.15 Kirim gambar tidak sesuai konteks

Jika gambar yang dikirim diluar pemrograman, maka bot akan membalas dengan pesan yang menggambarkan jika bot tidak mengerti dengan gambar yang dikirimkan oleh *user*.

3.2.3.6 Perancangan Halaman Profil



← back

Profile

Email

email@email.com

Nickname

Prasetya

First Name

Prasetya

Last Name

Prasetya

Birth Date

17 February 2004

LOGOUT

Gambar 3.16 Halaman Informasi User

Pada halaman informasi user ini terdapat beberapa informasi dari user seperti *email*, *nickname*, *first name*, *last name*, dan tanggal lahir dari user yang *login*. Dan juga terdapat button *logout* untuk mengeluarkan akun dari aplikasi.