



# Membuat chatbot menggunakan Rasa

---

**Mujahid Al Mutaz Billah**

<http://linkedin.com/in/mujahid-al-mutaz-billah>

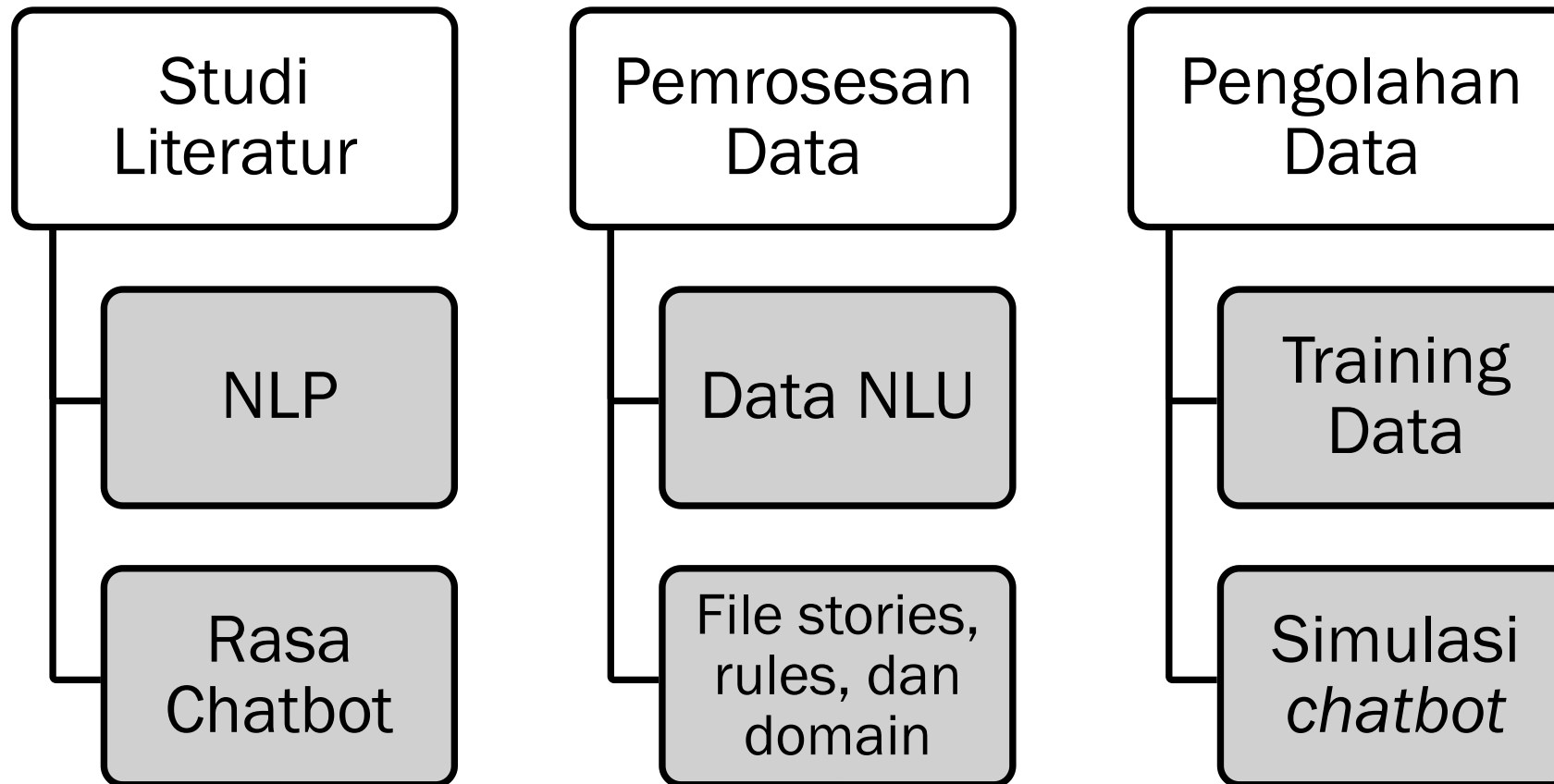
# Tujuan Penelitian

1. Membuat *chatbot* untuk menjawab FAQ (*Frequently Asked Questions*).
2. Membuat *chatbot* yang dapat memanggil nama pelanggan.
3. Membuat *chatbot* untuk pelanggan melakukan pemesanan.
4. Membuat *chatbot* yang dapat melakukan cek pemesanan.

# Ruang Lingkup Penelitian

1. Perusahaan yang akan dibuatkan *chatbot* bernama Harvestcakes.com.
2. *Framework* yang digunakan adalah Rasa versi 2.x.
3. Data NLU diambil dari dokumen soal rekrutmen yang diberikan.
4. Service hanya berupa console *command line interface*.

# Metode Penelitian



# Hasil Penelitian

---

# 1. Chatbot untuk FAQ

Akan diinputkan terlebih dahulu data NLU yang terdapat pada soal ke dalam file nlu.yml. Setelah itu, akan dibuat text response pada file domain.yml agar bot bisa menjawab pertanyaan sesuai yang diharapkan.

```
responses:
  utter_ask_pricelist:
    - text: "Hai kak, untuk informasi lengkap tentang produk The Harvest silakan kunjungi webiste kami di harvestcakes.com ya Kak, terimakasih"

  utter_ask_holiday:
    - text: "Outlet kami buka setiap hari ya kak, termasuk hari Minggu dan hari Libur dari pukul 09.00 - 21.00 WIB, Terima kasih"

  utter_ask_operating_hour:
    - text: "Hai Kak, dapat kami informasikan bahwa jam operasional outlet kami yaitu pukul 09.00 - 21.00 WIB ya Kak, Terima kasih"

  utter_ask_halal_item:
    - text: "Hai Kak, untuk saat ini kami sedang dalam proses sertifikasi yaa. Namun, sebagai informasi kami menggunakan bahan-bahan yang halal da

  utter_ask_custom_cake:
    - text: "Baik Kakak, untuk minimal waktu pemesanan Custom Cake adalah satu minggu sebelumnya ya dan diharapkan untuk langsung dipesan dengan p
```

# 1. Chatbot untuk FAQ

```
rules:

- rule: Menanyakan harga
  steps:
  - intent: ask_pricelist
  - action: utter_ask_pricelist

- rule: Menanyakan libur outlet
  steps:
  - intent: ask_holiday
  - action: utter_ask_holiday

- rule: Menanyakan jam operasional
  steps:
  - intent: ask_operating_hour
  - action: utter_ask_operating_hour

- rule: Mengenai sertifikat halal
  steps:
  - intent: ask_halal_item
  - action: utter_ask_halal_item

- rule: Mengenai kue kostum
  steps:
  - intent: ask_custom_cake
  - action: utter_ask_custom_cake
```

Agar bot bisa menjawab pertanyaan sesuai dengan apa yang ditanyakan oleh customer, akan diatur logika jawaban dan pertanyaan yang terdapat pada file rules.yml

# 1. Chatbot untuk FAQ

Berikut hasil simulasi *chatbot* untuk FAQ

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
```

```
Your input -> Price list nya ada
```

```
Hai kak, untuk informasi lengkap tentang produk The Harvest silakan kunjungi webiste kami di harvestcakes.com ya Kak, terimakasih
```

```
Your input -> apakah hari minggu buka
```

```
Outlet kami buka setiap hari ya kak, termasuk hari Minggu dan hari Libur dari pukul 09.00 - 21.00 WIB, Terima kasih
```

```
Your input -> Hari ini buka jam brp
```

```
Hai Kak, dapat kami informasikan bahwa jam operasional outlet kami yaitu pukul 09.00 - 21.00 WIB ya Kak, Terima kasih
```

```
Your input -> Mau tanya harvest halalkah
```

```
Hai Kak, untuk saat ini kami sedang dalam proses sertifikasi yaa. Namun, sebagai informasi kami menggunakan bahan-bahan yang halal dan tidak menggunakan bahan-bahan yang dilarang seperti rhum, gelatin babi dsb yaa Kak.
```

```
Your input -> Mau tanya terkait custom cake
```

```
Baik Kakak, untuk minimal waktu pemesanan Custom Cake adalah satu minggu sebelumnya ya dan diharapkan untuk langsung dipesan dengan pastry chef-nya di outlet kami yaa Kak, Terima Kasih
```

```
Your input -> 
```



## 2. Chatbot memanggil nama pelanggan

Untuk memanggil nama, akan dibuat list nama pada file nlu.yml dan diberikan entities. Entities tersebut kita set pada file domain.ymlu dan akan dibuatkan slot dengan nama yang sama agar dapat kita panggil.

```
- intent: customer_name_list
  examples: |
    - [Agus](name)
```

```
entities:
- name
- item
- qty
- rc

slots:
  name:
    type: unfeaturized
  item:
    type: text
```

## 2. Chatbot memanggil nama pelanggan

Untuk memanggil nama pelanggan, kita atur text response agar dapat memanggil slot nama lalu kita membuat logika percakapan kembali pada file rules.yml dan stories.yml

```
utter_ask_help:  
- text: "Hi kak {name}, dengan The Harvest ada yang bisa kami bantu?"  
  
utter_sufficient:  
- text: "Baik Kak {name}, terimakasih sudah menghubungi The Harvest. Jika ada yang ingin ditanyakan bisa langsung menghubungi kami kembaliya"
```

```
- story: Greet  
  steps:  
  - intent: greet  
  - action: utter_ask_customer_name  
  - intent: customer_name_list  
  - action: utter_ask_help
```

```
- rule: Closing  
  steps:  
  - intent: sufficient  
  - action: utter_sufficient
```

## 2. Chatbot memanggil nama pelanggan

Berikut hasil simulasi *chatbot* untuk memanggil nama pelanggan pada saat pembukaan dan penutupan.

```
Your input -> hi
Hai kak boleh tahu siapa Namanya?
Your input -> Agus
Hi kak Agus, dengan The Harvest ada yang bisa kami bantu?
Your input -> cukup
Baik Kak Agus, terimakasih sudah menghubungi The Harvest. Jika ada yang ingin ditanyakan bisa langsung menghubungi kami kembaliya
Your input -> 
```

Terdapat keterbatasan yaitu jika nama pelanggan tidak terdapat pada `nlu.yml`, maka bot tidak bisa memanggil nama tersebut. Hal itu bisa diatas dengan menggunakan form dan membutuhkan waktu untuk penelitian lebih lanjut

### 3. Chatbot melakukan pemesanan

Untuk melakukan pemesanan, akan dibuat list item dan kuantitas pada file nlu.yml dan diberikan entities. Entities tersebut kita set pada file domain.yml dan akan dibuatkan slot dengan nama yang sama agar dapat kita panggil.

```
- intent: order_qty
  examples: |
    - saya pesan [dua](qty)
    - [2](qty)
    - [dua](qty)
    - [1](qty) saja
    - [satu](qty)
    - pesan [satu](qty)
    - [1](qty)
    - 1 aja
    - [tiga](qty)
```

```
- intent: store_item
  examples: |
    - [Macaroon](item)
    - [Tiramisu](item)
```

```
entities:
- name
- item
- qty
- rc

slots:
  name:
    type: unfeaturized
  item:
    type: text
```

### 3. Chatbot melakukan pemesanan

Untuk memanggil jenis pemesanan dan kuantitasnya, kita atur text response agar dapat memanggil slot nama lalu kita membuat logika percakapan kembali pada file rules.yml dan stories.yml

```
utter_request_order:  
- text: "Hai kak anda ingin memesan apa? Berikut kue yang kami punya \n Macaroon \n Tiramisu \n"  
  
utter_store_item:  
- text: "Boleh dibantu untuk jumlah kue yang ingin dibeli Kak?"  
  
utter_order_qty:  
- text: "Baik kak, kue yang dipesan {item} dengan jumlah {qty}"
```

```
- story: Melakukan pemesanan  
  steps:  
  - intent: request_order  
  - action: utter_request_order  
  - intent: store_item  
  - action: utter_store_item  
  - intent: order_qty  
  - action: utter_order_qty
```

### 3. Chatbot melakukan pemesanan

Berikut hasil simulasi *chatbot* untuk melakukan pemesanan sesuai dengan apa yang dipesankan oleh customer.

```
Your input -> mau pesan
Hai kak anda ingin memesan apa? Berikut kue yang kami punya
Macaroon
Tiramisu
Your input -> Tiramisu
Boleh dibantu untuk jumlah kue yang ingin dibeli Kak?
Your input -> satu
Baik kak, kue yang dipesan Tiramisu dengan jumlah satu
```

## 4. Chatbot melakukan cek pemesanan

Untuk melakukan cek pemesanan, akan dibuat code terlebih dahulu untuk mengambil data pada API yang akan digunakan.

no\_resi didapatkan dari data NLU dengan entities “rc”

```
1  import requests
2
3  def Resi(no_resi):
4      api_address='https://bot-kemenkes-dev.azurewebsites.net/api/checkairwaybillhc?airwaybill='
5      # resi = input('masukan resi:')
6      url = api_address + no_resi
7      json_data = requests.get(url).json()
8      format_add = json_data['result']
9      final = format_add['text']
10     return final
```

## 4. Chatbot melakukan cek pemesanan

Selanjutnya akan membuat fungsi untuk memanggil hasil dari code tersebut ke dalam framework Rasa chatbot pada file actions.py.

```
1  from typing import Any, Text, Dict, List
2
3  from rasa_sdk import Action, Tracker
4  from rasa_sdk.executor import CollectingDispatcher
5  from resi import Resi
6
7
8  class ActionHelloWorld(Action):
9
10     def name(self) -> Text:
11         return "action_resi_api"
12
13     def run(self, dispatcher: CollectingDispatcher, tracker: Tracker, domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
14
15         no_resi = tracker.get_slot("rc")
16         text = Resi(no_resi)
17         dispatcher.utter_message(text=f"{text}")
18         return []
```



## 4. Chatbot melakukan cek pemesanan

Untuk membuat percakapan pada bot, akan dibuat logika percakapan yang memanggil actions tersebut pada stories.yml

```
- story: Melakukan cek pesanan
  steps:
    - intent: track_order
    - action: utter_ask_package
    - intent: ask_package
    - action: action_resi_api
```

## 4. Chatbot melakukan cek pemesanan

Kita set action\_resi\_api pada file domain.yml agar dapat dijalankan.

```
actions:  
- utter_ask_help  
- utter_sufficient  
- action_ask_customer_name  
- utter_ask_package  
- action_resi_api
```

```
action_endpoint:  
  url: "http://localhost:5055/webhook"
```

Sebelum disimulasikan, uncomment pada file endpoints.yml dan jalankan rasa run actions terlebih dahulu pada terminal yang berbeda agar server bisa berjalan.

## 4. Chatbot melakukan cek pemesanan

Berikut hasil simulasi *chatbot* melakukan cek pemesanan.

```
Hai kak silakan inputkan no resi anda
Your input -> cek order dmn
Hai kak silakan inputkan no resi anda
Your input -> 004007700001 ini ya kak
Hi Kak, untuk permintaan pick up paket dengan nomor resi 004007700001 sudah DLV. Mohon ditunggu dalam waktu
1x24 jam ya kak. Terima kasih
Your input -> 
```

Terima kasih.

---