

Rancang Bangun Aplikasi Resep Masakan Tanpa Kontak Dengan Perintah Suara

Jeremy Endratno¹, Caecilia Citra Lestari²

Abstrak - Sering tidak kita sadari bahwa ketika kita menggunakan aplikasi saat memasak, terjadi banyak kontak fisik antara perangkat dan pengguna. Dengan menggunakan speech recognition, pengguna dapat memberikan perintah menggunakan perintah suara. Seperti perintah untuk melanjutkan ke tahap berikutnya pada resep, pengguna hanya akan berkata "berikutnya". Banyak perintah lainnya dapat pengguna berikan untuk memperjelas resep pada saat memasak, contoh lainnya adalah "kembali", "ulangi", "apa itu tomat", "bagaimana cara menumis", dan banyak lainnya. Dengan mengucapkan perintah suara, teknologi speech recognition akan menangkapnya dan mengubah perintah suara menjadi sebuah text yang dapat dimengerti oleh komputer. Perintah akan dicari kebenarannya menggunakan Natural Language Processing. Dengan begitu komputer dapat menjalankan perintah yang berikan lebih akurat. Semua penjelasan resep dijelaskan dengan rapi pada 1 halaman tanpa scrolling sehingga pengguna dapat melihat instruksi dengan jelas. Instruksi juga dijelaskan melalui penjelasan suara menggunakan teknologi Speech Synthesis. teknologi ini membuat komputer dapat mengubah text menjadi suara yang dapat dimengerti oleh manusia. Dengan adanya perantara ini, pengguna dapat menghindari kontak fisik yang dapat menyebarkan bakteri pada bahan masakan. Aplikasi dibuat menggunakan lingkungan xcode dan bahasa pemrograman swift. Seluruh teknologi dapat diakses menggunakan library yang telah disediakan oleh xcode. Teknologi akan dirancang sedemikian rupa agar dapat berfungsi dengan baik pada aplikasi. Hasil uji coba mengatakan bahwa dalam kondisi normal sistem dapat dipakai dengan baik.

Kata Kunci: Speech recognition, speech synthesis, pengguna, perangkat, kontak fisik, natural language processing, xcode.

Abstract - As home cooks, we very often use recipes as a cooking guide. We can get recipes on various platforms, one of which is a recipe application. However, we often don't realize that when we use the application while cooking, there is a lot of physical contact between the device and the user. By using speech recognition, users can give commands using voice commands. As prompts to continue to the next step of the recipe, the user will just say "next". Many other commands the user can give to clarify the recipe when cooking, other examples are "return", "repeat", "what is a tomato", "how to sauté", and many others. By saying a voice command, speech recognition technology will capture it and convert the voice command into text that can be understood by a computer. The command will be looked and judged for the rightness by Natural Language Processing. That way the computer will run the command given more accurately. All recipe explanations are explained neatly on 1 page without scrolling so that users can see the instructions clearly. Instructions are also explained through voice explanations using Speech Synthesis technology. This technology allows computers to convert text into sounds that humans can understand. With this intermediary, users can avoid physical contact that can spread bacteria to cooking ingredients. The application is built using the xcode environment and swift programming language. All technologies can be accessed using a library provided by xcode. The technology will be designed in such a way as to function properly in the application. The test results show that under normal conditions the system can be used properly.

Keyword: Speech recognition, speech synthesis, user, device, physical contact, natural language processing, Xcode.

I. PENDAHULUAN

A. Latar Belakang

Banyak orang yang memasak akan menggunakan resep untuk menjadi pemandunya dalam memasak. Salah satu platform yang sering digunakan untuk mendapat panduan resep masakan adalah aplikasi seluler. Dalam proses memasak menggunakan aplikasi seluler, terjadi banyak sekali kontak fisik antara user dengan perangkatnya, terutama aplikasi resep yang menggunakan metode langkah demi langkah. Pada umumnya terdapat rata-rata 17.000 bakteri pada perangkat pengguna. Oleh karena itu disarankan untuk tidak menyentuh perangkat di antara menyiapkan bahan masakan.

Sebuah solusi sudah diterapkan diberbagai macam platform, akan tetapi setiap solusi memiliki kekurangan

¹ Mahasiswa Program Studi Informatika, Fakultas Teknologi Informasi Universitas Ciputra Surabaya, CitraLand CBD Boulevard, 60219, Indonesia (tlp: 6231-7451699; fax: 6231-7451698; e-mail: jendratno@student.ciputra.id)

² Dosen Program Studi Informatika, Fakultas Teknologi Informasi Universitas Ciputra Surabaya, CitraLand CBD Boulevard, 60219, Indonesia (tlp: 6231-7451699; fax: 6231-7451698; e-mail: caecilia.citra@student.ciputra.id)

dan kelebihannya sendiri. Youtube merupakan salah satu solusi para koki untuk menghindari kontak fisik. Dengan menonton video kontak fisik dapat dihindari, akan tetapi terdapat kekurangan dalam menonton video. Kekurangan terdapat pada kecepatan video dan instruksi bergerak, dimana untuk para koki yang masih amatir akan kesusahan untuk mengikuti instruksi jika tidak menghentikan video. Oleh karena itu, dalam penelitian ini akan diambil pendekatan yang berbeda.

Dikarenakan masalah diatas, maka dari itu penelitian ini bertujuan untuk menciptakan sebuah fitur yang mampu memberikan perantara interaksi antara user dan perangkat tanpa harus melakukan kontak fisik. Penggunaan teknologi *speech recognition* untuk mengubah perintah suara pengguna menjadi *text based* pada aplikasi agar dapat dikelola dan dimengerti oleh aplikasi. Dan penggunaan teknologi Speech Synthesis agar dapat mengubah perintah aplikasi yang berupa *text based* menjadi sebuah perintah suara artifisial yang dapat dimengerti user.

B. Rumusan Masalah

Bagaimana cara merancang dan membangun aplikasi resep masakan tanpa kontak fisik dengan menggunakan perintah suara?

II. KAJIAN PUSTAKA

A. Studi Terdahulu

Menurut time.com, terdapat sekitar 17.000 pada perangkat yang kita pakai sehari-hari. Dalam penggunaan perangkat untuk mengakses aplikasi resep masakan, tentu terjadi banyak sekali kontak fisik antara user dan perangkat. Kontak fisik tersebut dapat menyebabkan bahan masakan yang kita pakai terpenuhi oleh bakteri [1].

Sangatlah umum jika terdapat kurangnya kebersihan pada layar perangkat. Apalagi penggunaan perangkat layar sentuh pada publik. "Saya sangat khawatir mengenai perangkat layar sentuh yang digunakan restoran cepat saji untuk memesan makanan dan mereka memakan makanannya dengan tangan kosong tanpa peralatan makan" kata Dr Robert Almer, profesor kesehatan publik di New York Medical Collage [2].

Terdapat berbagai macam bakteri yang dapat diidentifikasi saat ini. Bakteri sangatlah unik, bakteri terkadang dapat menetap di tubuh kita tanpa menyebabkan gangguan apapun. Meskipun bakteri tidak seganas virus dan parasit, akan tetapi juga dapat berkembang dalam tubuh kita dan menyebabkan tumbuhnya penyakit [3].

B. Terminologi

Teknologi yang akan digunakan pada aplikasi adalah *speech recognition*, *speech synthesis*, dan *natural language processing*. Pada sub bab ini akan dijelaskan cara kerja teknologi tersebut.

1) Speech Recognition

Speech recognition adalah sebuah fungsi yang memberikan mesin untuk mengerti ucapan user dan menkonversinya menjadi serangkaian kata yang dapat dibaca oleh komputer, sehingga dapat menciptakan komunikasi natural antara manusia dan mesin. Fungsi ini membagi data suara dan menganalisisnya berdasarkan model bahasa. Hasil analisa akan dikembalikan sebagai *output decoder* yang merupakan kemungkinan terbaik dari kalimat yang diucapkan dalam bentuk text [4]. Dengan user mengucapkan beberapa kalimat perintah, aplikasi akan menangkapnya dan menjalankan sebuah fungsi.

Bentuk algoritma paling simpel dalam *speech recognition* dinamakan "Simple Pattern Matching". Algoritma ini biasa digunakan untuk mendeteksi kumpulan kosa kata yang sangat kecil, seperti "satu", "dua", sampai dengan "sepuluh". dari kosa kata tersebut, tidak ada kata yang memiliki fones yang mirip. Sehingga komputer dapat membedakan pola suara yang memiliki nada fonem yang terdengar berbeda. Algoritma ini menyelesaikan masalah dengan membandingkan dan menyesuaikan seluruh bongkah pola suara dengan pola yang ada pada memorinya [5].

Karena algoritma yang digunakan di atas menyelesaikan kumpulan kosa kata yang relatif kecil, maka peneliti mencari algoritma baru dengan menggunakan Machine Learning. Sistem Speech Recognition dimulai dari mendengarkan sebuah suara yang dibaca melalui microphone. Karena komputer tidak dapat mengerti dengan data suara yang apa adanya, maka langkah pertama yang dilakukan adalah untuk mengubah data suara menjadi data digital menggunakan analog-to-digital converter. Data digital tersebut dapat dilihat dalam bentuk spectrogram menggunakan teknik matematika yang dinamakan Fast Fourier Transform. Data tersebut akan dipecah menjadi beberapa seri bongkahan yang saling tumpang tindih, bongkahan tersebut dinamakan "acoustic frames". bongkahan tersebut akan di proses secara digital dan dianalisis untuk mencari komponen perkataan yang mereka miliki. Dalam sebuah kata terdapat beberapa bongkahan yang bisa menjadi kunci utama dalam mencari tahu kata tersebut. Jadi algoritma kali ini yaitu algoritma "Pattern and Feature Analysis" hanya akan membandingkan kunci utama bongkahan tersebut dengan *phonetic dictionary*, yaitu sebuah kamus kata yang memiliki pecahan kunci utama [6].

Dalam kehidupan nyata, penggunaan Speech Recognition merupakan hal yang lebih kompleks dari yang kita duga. Ada beberapa kelemahan yang kita harus fokuskan pada penggunaan algoritma sebelumnya. Dari sana peneliti membuat sebuah model bahasa yang dinamakan "Natural Language Processing" [7].

Ketika orang berbicara, mereka tidak hanya mengeluarkan kata – kata yang acak. Setiap kata akan bergantung dengan kata yang sebelum atau sesudahnya

diucapkan. bahasa juga mempunyai sebuah gramatika, agar dapat tersusun rapi. Dari logika di atas kita dapat membuat model bahasa yang dapat di implementasikan dalam Speech Recognition dengan menggunakan algoritma Viterbi. Dengan menghitung kemungkinan kata yang akan selanjutnya muncul, komputer dapat memprediksi kata tersebut. Hidden Markov Model merupakan sebuah sistem matematika yang dibuat oleh Andrey Markov untuk menghitung perhitungan prediksi agar komputer dapat menemukan kemungkinan kata terbaik. Akan tetapi dalam sudut pandang komputer, kemungkinan terbaik tidak akan pernah mencapai seratus persen, sehingga jawaban yang benar harus dikonfirmasi oleh user itu sendiri [8].

2) Speech Synthesis

Speech Synthesis adalah sebuah fungsi yang mengurus perancangan sistem komputer yang mensintesis teks tertulis. teknologi yang memperkenankan komputer untuk menkonversi sebuah teks tertulis menjadi ucapan melalui microphone atau telephone. Fungsi ini mengambil sebuah input berupa teks dan dianalisis dengan menggunakan model linguistik. Hasil analisis akan diberikan ilmu prosidi dan digenerasikan sebuah output berupa suara yang dapat dimengerti oleh user [9].

Setelah mengetahui kata-kata yang perlu diucapkan, sistem sekarang harus menghasilkan suara ucapan yang membentuk kata-kata tersebut. Secara teori, ini adalah masalah sederhana: semua yang dibutuhkan komputer adalah daftar alfabetis besar kata-kata dan detail cara melaftalkan masing-masing. Untuk setiap kata, kita memerlukan daftar fonem yang membentuk suaranya. Sistem dapat menggunakan rekaman suara manusia yang harus dimuat sebelumnya dengan potongan kecil suara manusia yang dapat mereka atur ulang. Dengan kata lain, seorang programmer harus merekam banyak contoh orang yang mengatakan hal-hal yang berbeda, memecah kalimat yang diucapkan menjadi kata-kata dan kata-kata menjadi fonem. Jika ada cukup sampel ucapan, komputer dapat mengatur ulang bit dengan berbagai cara berbeda untuk membuat kata dan kalimat yang sama sekali baru. Jenis sintesis ucapan ini disebut *concatenative*. Karena didasarkan pada rekaman manusia, penggabungan adalah jenis sintesis ucapan yang terdengar paling alami dan digunakan secara luas oleh mesin yang hanya memiliki hal-hal terbatas untuk diucapkan. Kelemahan utamanya adalah bahwa itu terbatas pada satu suara dan satu bahasa [10].

3) Natural Language Processing

Natural Language Processing atau singkatnya NLP, secara luas didefinisikan sebagai manipulasi otomatis bahasa alami, seperti ucapan dan teks, oleh perangkat lunak. Studi tentang pemrosesan bahasa alami telah ada

selama lebih dari 50 tahun dan berkembang melampaui bidang linguistik dengan kemunculannya komputer.

Natural Language mengacu pada cara kita, manusia, berkomunikasi satu sama lain. Seperti, cara bicara dan teks. Kita mungkin lebih terbiasa berbicara satu sama lain, lebih daripada menulis. Bahkan mungkin lebih mudah belajar berbicara daripada menulis. Suara dan teks adalah cara kita berkomunikasi satu sama lain. Mengingat pentingnya jenis data ini, kita harus memiliki metode untuk memahami dan bernalar tentang bahasa alami, seperti yang kita lakukan untuk jenis data lainnya.

Linguistik komputasi adalah studi linguistik modern dengan menggunakan alat-alat ilmu komputer. Linguistik kemarin mungkin menjadi ahli bahasa komputasi saat ini sebagai penggunaan alat komputasi dan pemikiran telah mengambil alih sebagian besar bidang studi. Linguistik komputasi adalah studi tentang sistem komputer untuk memahami dan menghasilkan bahasa alami. Satu fungsi alami untuk linguistik komputasi akan menjadi pengujian tata bahasa yang diusulkan oleh ahli bahasa teoretis.

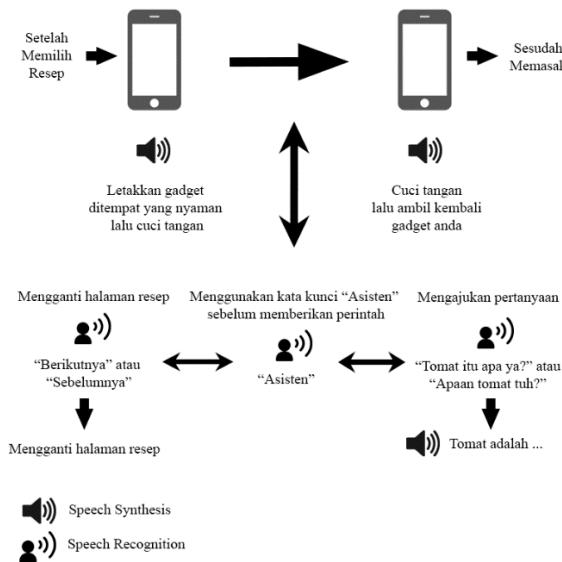
Alat dan metode dari bidang Pengolahan Bahasa Alami belum berkembang. Mungkin salah satu yang lebih dikenal luas buku teks yang ditulis oleh peneliti top di lapangan, mereka merujuk pada subjek sebagai ilmu linguistik, mengizinkan diskusi tentang linguistik klasik dan metode statistik modern. Tujuan dari ilmu linguistik adalah untuk dapat menggambarkan dan menjelaskan orang banyak pengamatan linguistik yang berputar-putar di sekitar kita, dalam percakapan, tulisan, dan lainnya media. Sebagian dari itu berkaitan dengan ukuran kognitif tentang bagaimana manusia memperoleh, berproduksi dan memahami bahasa, sebagian berkaitan dengan memahami hubungan antara ucapan linguistik dan dunia, dan bagian darinya berkaitan dengan pemahaman struktur linguistik yang digunakan bahasa untuk berkomunikasi [11].

NLP biasa dilatih menggunakan bahasa pemrograman Python. Salah satu hal bersahabat tentang Python adalah memungkinkan kita untuk mengetik langsung ke file interpreter interaktif — program yang akan menjalankan program Python. kita dapat mengakses interpreter Python menggunakan antarmuka grafis sederhana yang disebut Interactive Development Environment (IDLE) [12].

III. METODOLOGI PENELITIAN

A. Desain Proses Perintah Tanpa Kontak

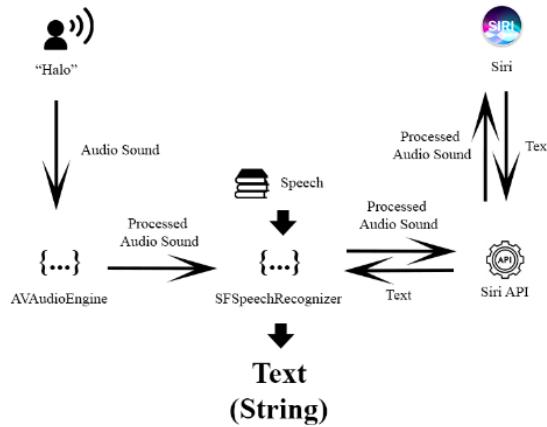
Desain tanpa kontak adalah desain penggunaan Speech Recognition dan Speech Synthesis pada aplikasi untuk menjadi perantara agar menghindari kontak fisik.



Gambar. 1. Desain Proses Perintah Tanpa Kontak

1) Desain Proses Speech Recognition

Speech Recognition yang digunakan pada aplikasi adalah fitur Siri yang dibagikan kepada pengguna Xcode secara gratis dengan menggunakan library Speech. Speech merupakan library yang dapat dengan praktis menghubungkan API Speech Recognition Siri pada project aplikasi. API dapat dipanggil dengan batas 1000 panggilan per jam dan maksimal 1 menit per panggilan.

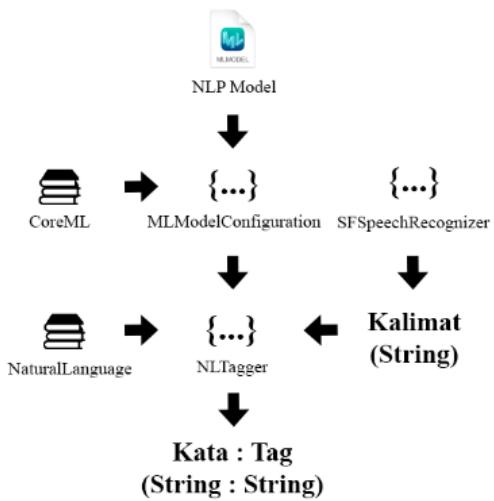


Gambar. 2. Desain Proses Speech Recognition

SFSpeechRecognizer merupakan sebuah function pada library speech yang digunakan untuk memanggil API Siri. Saat function SFSpeechRecognizer dipanggil, selama 1 menit kita dapat memberikan function tersebut sebuah data suara agar dikembalikan sebagai data text.

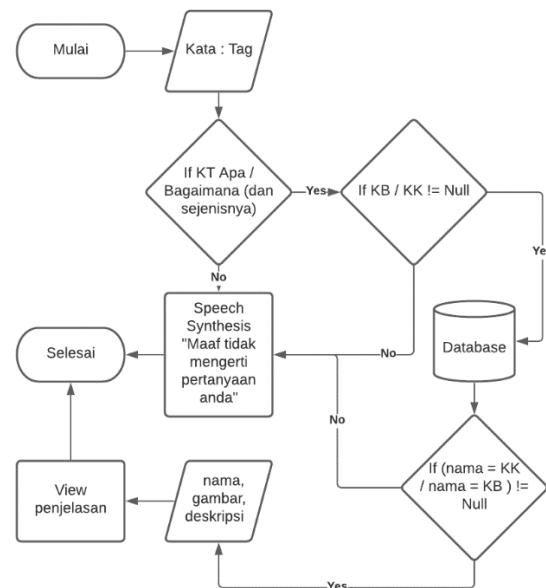
2) Desain Proses Natural Language Processing

Setelah mendapatkan model NLP, dibuat sebuah logika penggunaan model yang intregasikan pada fungsi tagger dengan menggunakan 2 library yaitu NaturalLanguage dan CoreML.



Gambar. 3. Desain Proses Natural Language Processing

Sebelum NLP model dipakai oleh NLTagger, model harus dikonfigurasi oleh CoreML menggunakan fungsi MLModelConfiguration. Setelah itu model akan dapat digunakan pada NLTagger. Hasil kalimat yang sudah terekam oleh Speech Recognition menggunakan fungsi SFSpeechRecognizer, akan menghasilkan sebuah data text berbentuk kalimat. Data tersebut akan dijadikan input pada NLTagger. NLTagger akan memproses sehingga menghasilkan output berbentuk kata yang sudah terpecah dan tertag.



Gambar. 4. Desain Algoritma Penggunaan Natural Language Processing

Setelah kalimat sudah terpecah dan tertag, data tersebut akan dimasukan ke algoritma sederhana untuk mencari konten dari pertanyaan pengguna pada database. Algoritma ini menggunakan if untuk memastikan

kebenaran pertanyaan yang ditanyakan. Dan memastikan juga bahwa konten yang dicari ada pada database. Jika memang ada pada database maka database akan memberikan data berupa nama, gambar, dan deskripsi yang akan dijabarkan pada view penjelasan. Jika konten yang dicari tidak ada, maka aplikasi akan memberitahu pengguna melalui speech synthesis bahwa aplikasi tidak mengerti pertanyaannya.

3) Desain Proses Speech Synthesis

Fungsi speech synthesis diambil dari library bernama AVFoundation. Library tersebut bisa dipakai secara gratis oleh developer yang menggunakan Xcode. Pada library tersebut terdapat fungsi bernama AVSpeechSynthesizer yang menjalankan fungsi speech synthesis. Fungsi tersebut hanya meminta sebuah input string yang akan dijadikan sebuah suara pada aplikasi. Sehingga penjelasan konten dari pertanyaan maupun lainnya diberikan dalam bentuk suara dengan menggunakan fungsi tersebut.

B. Desain Model Natural Language Processing

POS Tag atau Part of Speech Tagging merupakan sebuah bentuk umum dari NLP yang sering digunakan. Sistem kerja Postag adalah dengan memberi tag pada kata dalam sebuah kalimat. Tag merupakan istilah digunakan untuk kelas kata, sebagai contoh, "memasak" merupakan "kata kerja". Jadi jika dalam kalimat seperti "Saya akan bekerja besok" akan menghasilkan tag, "subjek, kata sambung, kata kerja, keterangan waktu".

Dalam implementasi NLP pada aplikasi, terjadi beberapa masalah yang mengharuskan untuk melatih model NLP sendiri. Pada kumpulan library milik Apple sendiri, terdapat library NLP yang dapat dipakai secara gratis bernama NaturalLanguage. Tentu saja pada NaturalLanguage terdapat fungsi Postag yang dapat digunakan secara bebas dan praktis. Akan tetapi pada NaturalLanguage hanya terbatas pada bahasa Inggris saja, sedangkan aplikasi yang dikembangkan akan menggunakan bahasa Indonesia.

Dari masalah diatas, diharuskan untuk melatih model NLP sendiri yang sesuai untuk penggunaan pada aplikasi. Untuk melatih model NLP, kita dapat menggunakan library Machine Learning milik Apple bernama CreateML. Pada CreateML kita memerlukan data list yang berisi sebuah string yang merupakan kata yang membentuk kalimat. Kita juga memerlukan tag dari kata tersebut yang sesuai dengan urutan katanya. Contoh data adalah sebagai berikut.

```
"tokens":["Apa","itu","tomat","?"],  
"labels":["KT","TDT","KB","TT"]
```

Terdapat banyak dataset tersedia di internet untuk melatih model NLP Postag Indonesia. Sebagian besar dataset diambil dari sebuah novel atau berita yang tiap

kalimat sudah di tag satu per satu. Akan tetapi dataset tersebut hanya mewakili kata dan kalimat formal saja, sedangkan untuk aplikasi ini dibutuhkan interaksi formal maupun informal untuk melancarkan proses memasak. Oleh karena itu penulis membuat dataset sendiri dengan menggunakan kata – kata dan kalimat formal maupun informal.

Dikarenakan dataset dibuat untuk aplikasi resep masakan, maka kalimat dan kata serta tag yang digunakan akan dibuat seminimal mungkin yang mencangkupi fungsional aplikasi. Kalimat yang digunakan hanya kalimat tanya, dan kosa kata yang digunakan hanya diambil dari seputar kegiatan memasak, lebih spesifik adalah bahan dan metode masakan yang sudah ditentukan melalui survei.

Dalam percobaan melatih model NLP, dataset beberapa kali diubah atau ditambahkan untuk mendapatkan hasil yang sesuai, dan meningkatkan akurasi postag. Berikut perubahan dataset yang terbagi menjadi 3 fase.

Dataset Training 1	Dataset Training 2	Dataset Training 3
Berisi kalimat tanya formal maupun informal Sepengetahuan penulis dengan kosa kata yang sangat terbatas	Menambahkan kosa kata berupa bahan dan metode masakan yang sudah diambil pada survei	Menambahkan kalimat tanya formal maupun informal yang diambil dari internet

```

"tokens": ["Apa", "itu", "tomat", "?"],  
"labels": ["KT", "TDT", "KB", "TT"]  
  

"tokens": ["Apa", "itu", "penyet", "?"],  
"labels": ["KT", "TDT", "KB", "TT"]  
  

"tokens": ["Apakah", "Andrea", "sudah",  
"memasak", "bahan", "?"],  
"labels": ["KT", "TDT", "KB", "TT"]

```

Gambar. 5. Dataset Training

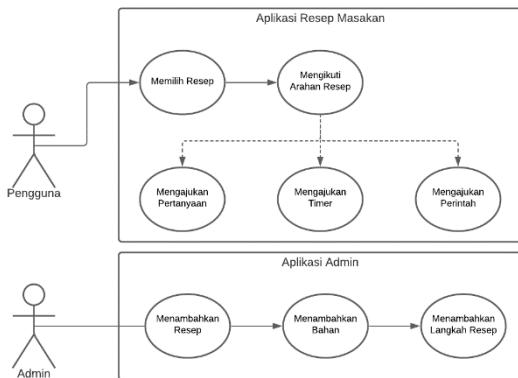
Tiap dataset ditambahkan data sesuai dengan kekurangan dari dataset sebelumnya. Menghasilkan hasil yang lebih sepadan dengan fungsional dari aplikasi. Hasil model memiliki prediksi yang berbeda, berikut adalah tabel prediksi model.

Tiap dataset ditambahkan data sesuai dengan kekurangan dari dataset sebelumnya. Menghasilkan hasil yang lebih sepadan dengan fungModel dataset training 2 merupakan model yang cukup untuk mendukung fungsional aplikasi dengan baik. Akan tetapi dengan menambahkan data dari internet, dapat memperluas kemungkinan cara bertanya orang. Karena alasan tersebut untuk proyek aplikasi ini akan menggunakan model dataset training dari aplikasi. Hasil model memiliki prediksi yang berbeda, berikut adalah tabel prediksi model.

C. Desain Aplikasi

Desain aplikasi dijelaskan menggunakan Unified Modeling Language yaitu sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek.

1) Use Case Diagram

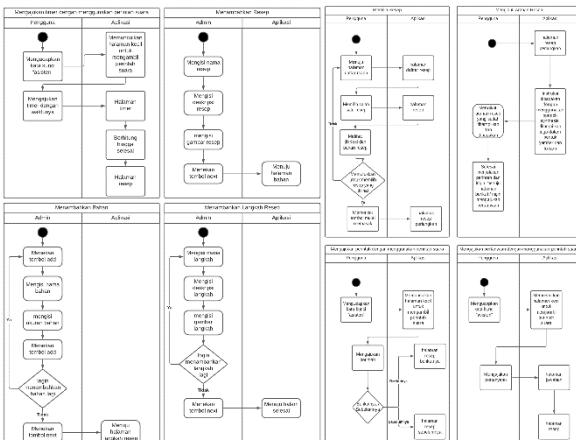


Gambar. 6. Use Case Diagram

Pengguna memilih resep pada aplikasi dan mengikuti arahan. Pada saat memasuki halaman perlangkah resep, pengguna dapat mengajukan perintah, pertanyaan, ataupun timer dengan perintah suara agar tidak terjadi kontak fisik antara pengguna dan gadget. Sedangkan admin dapat menambahkan resep, bahan, dan langkah resep pada aplikasi admin.

2) Activity Diagram

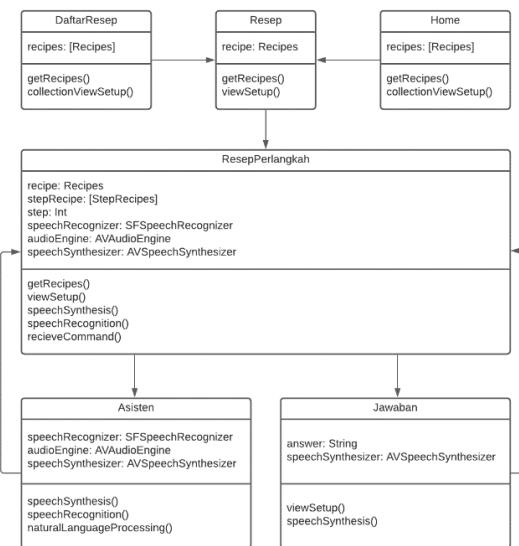
Activity diagram akan menjelaskan secara detail fitur – fitur yang dialami oleh pengguna pada use case diagram. berikut adalah activity diagram dari aplikasi.



Gambar. 7. Activity Diagram

3) Class Diagram

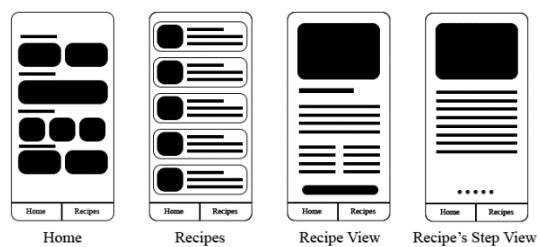
Class Diagram adalah diagram yang menunjukkan class, variable, serta fungsi yang membangun fitur – fitur dari aplikasi. Berikut adalah class diagram dari aplikasi.



Gambar. 8. Class Diagram

4) Desain UI

Dari yang sudah diambil dari kesimpulan komparasi, desain aplikasi menggunakan desain simplistic dan layout card and grid dengan menggunakan warna dominan jingga dan warna sekunder putih.



Gambar. 9. Desain UI

Desain aplikasi terbagi menjadi 4 halaman utama yaitu home, recipes, recipe view, dan recipe's step view. Desain navigasi menggunakan Bottom Navigation Bar dengan konten home dan recipes.

5) Desain Database

Database digunakan untuk menyimpan data bahan dan metode masakan serta resep masakan.



Gambar. 10. Desain Database

IV. IMPLEMENTASI

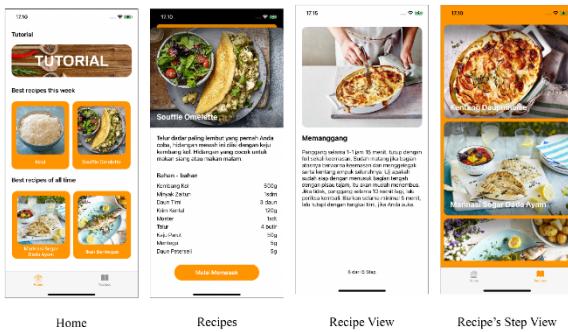
Semua proses rancang bangun aplikasi resep masakan tanpa kontak dibuat dalam lingkungan Xcode. Dengan menggunakan bahasa pemrograman swift dan library yang telah disediakan.

A. User Interface

Dengan desain wireframe yang sudah dibuat pada bab tiga, telah implementasikan dengan menggunakan berbagai macam object pada Xcode.

1) User Interface Aplikasi

Aplikasi dibagi menjadi 4 halaman utama yaitu home, recipes, recipes view, dan recipe's step view. Berikut adalah gambar *user interface* aplikasi yang sudah diimplementasikan.



Gambar. 11. User Interface

2) Collection View

CollectionView merupakan sebuah tampilan daftar yang digunakan untuk daftar resep pada halaman home maupun recipe. Untuk membuat collectionview hal pertama yang harus kita lakukan adalah membuatnya pada storyboard dan menyambungkannya pada class viewcontroller.

```
@IBOutlet weak var recipesCollectionView: UICollectionView!
```

Setelah tersambung kita dapat mengisinya dengan data resep.

```
func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int { return recipes.count - 1 }
```

Fungsi ini digunakan untuk menentukan jumlah resep yang akan ditampilkan.

```
func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
    let cell = recipesCollectionView.dequeueReusableCell(withIdentifier: "RecipesCell", for: indexPath) as! RecipesCollectionViewCell
    cell.setCell(data: recipes[indexPath.row + 1])
    return cell
}
```

Fungsi ini digunakan untuk mengisi data resep pada setiap baris sel.

```
func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath: IndexPath) {
    clickedIndex = indexPath
    recipesCollectionView.deselectItem(at: indexPath, animated: true)
    performSegue(withIdentifier: "DetailRecipe", sender: self)
}
```

Fungsi ini digunakan untuk menampilkan halaman baru sesuai dengan baris sel yang sudah dipencet.

B. Sistem Tanpa Kontak

Sistem tanpa kontak dibuat dengan menggunakan 3 macam teknologi yaitu speech recognition, natural language processing, dan speech synthesis. Dalam bab ini, akan dijabarkan cara penggunaan fungsi tersebut.

1) Speech Recognition

Speech recognition dapat diakses menggunakan fungsi `SFSpeechRecognizer` dalam library Speech.

Import Speech

```
private var recognitionRequest:
```

```
SFSpeechAudioBufferRecognitionRequest?
```

```
private var audioEngine: AVAudioEngine!
```

```
private var inputNode: AVAudioInputNode!
private var audioSession: AVAudioSession!
```

Langkah pertama yang harus dilakukan untuk mengimport library lalu membuat variabel yang dibutuhkan untuk menjalankan fungsi.

```
SFSpeechRecognizer.requestAuthorization {}
```

Setelah mendapatkan izin, kita akan menyalakan API Siri dengan menggunakan SFSpeechRecognizer. Dengan menggunakan recognitionRequest sebagai input yang telah menampung permintaan.

```
recognizer?.recognitionTask(with:
recognitionRequest!) { (result, error) in
print(result.bestTranscription.formattedString, "--final \(result.isFinal)") }
```

Meskipun API sudah menyala tentu saja kita belum mendapatkan output. Pertama kita harus membuat permintaan dan menginputkannya kepada recognitionRequest. Sebelumnya kita harus mengatur format data suara yang akan diterima serta menginputkannya kedalam variabel audioEngine. Jika sudah diinput maka audioEngine dapat masuk pada kondisi siap siaga.

```
inputNode = audioEngine.inputNode
let recordingFormat =
inputNode.outputFormat(forBus: 0)
inputNode.installTap(onBus: 0, bufferSize: 1024,
format: recordingFormat) {
self.speechRecognizer.append(buffer) }
audioEngine.prepare()
```

Jika alat rekam sudah siap siaga, maka kita dapat memulai sesi rekaman dengan menggunakan audioSession.

```
audioSession = AVAudioSession.sharedInstance()
try audioSession.setCategory(.record, mode:
.spokenAudio, options: .duckOthers)
try audioSession.setActive(true, options:
.notifyOthersOnDeactivation)
try audioEngine.start()
```

Dengan menyalanya sesi rekaman maka setiap suara yang kita ucapkan akan direkam oleh device. Hasil rekaman tersebut akan langsung masuk pada recognitionRequest yang menjadi input SFSpeechRecognizer. SFSpeechRecognizer akan meminta siri untuk mengenal data suara, dan dikembalikan dalam bentuk text.

2) Speech Synthesis

Speech Synthesis dapat diakses menggunakan fungsi AVSpeechSynthesizer dari library AVFoundation.

Import AVFoundation

Setelah mengimport library, kita akan membuat variable AVSpeechUtterance untuk menampung string dan pengaturan pengucapan.

```
let utterance = AVSpeechUtterance(string: "Halo")
utterance.voice =
AVSpeechSynthesisVoice(language: "id-ID")
utterance.rate =
AVSpeechUtteranceDefaultSpeechRate
```

Kemudian dengan utterance sebagai input, kita akan menggunakan fungsi AVSpeechSynthesizer untuk mem sintesiskan text menjadi suara.

```
let synthesizer = AVSpeechSynthesizer()
synthesizer.speak(utterance)
```

Setelah baris tersebut device akan mengeluarkan suara sesuai dengan text dan pengaturan yang sudah di atur.

3) Natural Language Processing

Natural Language Processing diakses dengan menggunakan fungsi NLTagger dari library NaturalLanguage. Serta dengan bantuan CoreML untuk mengkonfigurasi modelnya.

Import NaturalLanguage

Import CoreML

Setelah mengimport kedua library yang pertama kita harus lakukan adalah memanggil model dan mengkonfigurasikannya.

```
let mlModel = try indonesia1(configuration:
MLModelConfiguration()).model
```

```
let customModel = try NLModel(mlModel:
mlModel)
```

Langkah kedua adalah untuk membuat variable tagger yang berisi fungsi NLTagger. Serta menginputkan model dan text yang ingin di tag.

```
let tagger = NLTagger(tagSchemes: [.nameType,
customTagScheme])
```

```
tagger.string = "Halo apa kabar?"
```

```
tagger.setModels([customModel], forTagScheme:
customTagScheme)
```

Langkah terakhir adalah untuk meminta tagger untuk memulai ngetag kalimat yang sudah diinputkan.

```

tagger.enumerateTags(in:
text.startIndex..<text.endIndex, unit: .word,
scheme: customTagScheme, options:
.omitWhitespace) { tag, tokenRange in

if let tag = tag {
    print("\(text[tokenRange]): \(tag.rawValue)") }
}

```

Setelah itu kita akan mendapatkan output berupa kata – kata pada kalimat yang sudah ditag.

C. Pelatihan Model Natural Language Processing

Pelatihan model natural language precessing menggunakan fungsi MLWordTagger yang dapat diambil pada library CreateML.

Import CreateML

Setelah mengimport librarynya, kita akan memanggil dataset yang telah dibuat.

```

let trainingData = try MLDataTable(contentsOf:
Bundle.main.url(forResource: "train",
withExtension: "json")!)

```

Langkah kedua adalah untuk membuat variable yang berisi fungsi MLWordTagger. Variable tersebut diinputkan dataset dan nama kepala kolom. Serta membuat metadata untuk informasi model yang akan dibuat. Lalu model dapat dioutput dengan jalur yang sudah ditentukan

```

let model = try MLWordTagger(trainingData:
trainingData, tokenColumn: "tokens", labelColumn:
"labels")

let metadata = MLModelMetadata(author: "Jeremy
Endratno", shortDescription: "POS Tag Model",
version: "1.0")

try model.write(to: URL(fileURLWithPath:
"/Users/mac/Desktop/indonesia.mlmodel"),
metadata: metadata)

```

V. HASIL PENGUJIAN DAN PEMBAHASAN

A. Pengujian Sistem Tanpa Kontak

1) Tujuan Pengujian

Pengujian dilakukan untuk melihat akurasi sistem, apakah teknologi yang sudah implementasikan dapat berjalan sesuai keinginan. Pengujian dilakukan juga untuk mengidentifikasi semua kemungkinan masalah dan bug pada sistem.

2) Metode Pengujian

Pengujian dilakukan seperti alpha testing, yaitu dengan mencoba sistem secara terus menerus dan

mencatat proses berjalanannya sistem. Dari data yang sudah dicatat akan dicari tingkat keberhasilan sistem dari aplikasi. Serta apakah sistem dapat menjalankan tugasnya dengan baik atau tidak.

3) Skenario Pengujian

Pengujian dilakukan dengan menggunakan fitur tanya jawab dan perintah asisten. Dari fitur tersebut, dapat dilihat apakah sistem speech recognition, natural language processing dapat dijalankan dengan baik atau tidak.

4) Hasil Pengujian

Berikut adalah tabel hasil dari pengujian sistem tanpa kontak yang dibagi menjadi 3 kondisi.

Perintah	Hasil Pengujian Perintah		
	Keberhasilan Kondisi Ramai	Keberhasilan Kondisi Cepat	Keberhasilan Kondisi Normal
Berikutnya	0 dari 10	10 dari 10	10 dari 10
Sebelumnya	0 dari 10	10 dari 10	10 dari 10

Tabel X
Hasil Pengujian Perintah

Bahan	Hasil Pengujian Bahan Masakan		
	Keberhasilan Kondisi Ramai	Keberhasilan Kondisi Cepat	Keberhasilan Kondisi Normal
Genjer	1 dari 10	2 dari 10	9 dari 10
Katuk	3 dari 10	2 dari 10	10 dari 10
Okra	0 dari 10	1 dari 10	9 dari 10
Kenikir	1 dari 10	1 dari 10	9 dari 10
Lawang	1 dari 10	0 dari 10	9 dari 10
Zucchini	0 dari 10	0 dari 10	0 dari 10
Bit	0 dari 10	0 dari 10	0 dari 10
Pakcoy	0 dari 10	0 dari 10	0 dari 10
Kelor	3 dari 10	4 dari 10	10 dari 10
Pakis	4 dari 10	3 dari 10	10 dari 10

Tabel XI
Hasil Pengujian Bahan Masakan

Metode	Hasil Pengujian Metode Masakan		
	Keberhasilan Kondisi Ramai	Keberhasilan Kondisi Cepat	Keberhasilan Kondisi Normal
Merebus	6 dari 10	8 dari 10	10 dari 10
Menggoreng	3 dari 10	5 dari 10	10 dari 10
Membakar	3 dari 10	5 dari 10	10 dari 10
Memanggang	2 dari 10	3 dari 10	10 dari 10
Mengasap	2 dari 10	1 dari 10	10 dari 10

5) Analisa Pengujian

Hasil pengujian perintah pada kondisi cepat dan normal adalah 100%, akan tetapi dalam kondisi ramai

akurasi sistem adalah 0%. Hal ini dikarenakan untuk memberikan perintah sistem memerlukan presisi perintah. Dalam arti sistem hanya menerima text dalam kondisi "Berikutnya" atau "Sebelumnya" saja, jika text tidak seperti itu maka sistem tidak akan menjalankan perintah.

Hasil pengujian bahan masakan pada sebagian besar bahan masakan dalam kondisi normal adalah 90 – 100%, akan tetapi untuk beberapa bahan mendapatkan akurasi 0%. Hal ini dikarenakan pada sistem speech recognition tidak dapat mengambil nama bahan yang bukan nama indonesia seperti zucchini. Dan untuk kondisi ramai dan cepat sebagian pertanyaan gagal diambil oleh sistem, hal ini dikarenakan sistem mengambil kata yang mirip akan nama bahan jika tidak diucapkan secara presisi.

Hasil pengujian metode masakan pada kondisi normal adalah 100%. Pada kondisi ramai dan cepat akurasi tidak terlalu tinggi, hal ini sebabkan oleh alasan yang sama pada bahan masakan yaitu sistem mengambil kata yang mirip dengan nama metode jika tidak diucapkan dengan presisi.

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Untuk merancang dan membangun aplikasi resep masakan tanpa kontak fisik, teknologi speech recognition, natural language processing, dan speech synthesis dapat menjadi salah satu alternatif. Speech recognition merupakan teknologi untuk merubah data suara menjadi text (string). Natural language processing adalah sistem linguistik untuk membantu memproses data bahasa. Sedangkan speech synthesis adalah teknologi untuk mengubah text menjadi suara.

Dengan bantuan teknologi tersebut dapat dirancang sedemikian rupa pada sebuah lingkungan xcode dan bahasa pemrograman swift untuk membuat aplikasi resep masakan tanpa kontak fisik. Semua teknologi dapat diakses menggunakan library yang sediakan oleh xcode kepada pengembang. Speech recognition dapat dipakai menggunakan fungsi yang menyambungkan API Siri pada pengembang. Model natural language processing dapat dilatih dan dipakai pada dalam bentuk fungsi untuk membantu proses pengolahan bahasa. Dan speech synthesis dapat dipakai berkat library multimedia yang sudah disediakan.

Dari hasil uji coba yang sudah dilakukan, akurasi sistem tanpa kontak ternilai baik pada kondisi normal. Sistem dapat dipakai untuk membuat aplikasi tanpa kontak yang baik dan fungsional, uji coba aplikasi juga mengatakan bahwa sebagian besar pengguna puas akan sistem yang telah dibuat. Akan tetapi beberapa kelemahan dapat ditemukan dan sistem belumlah sempurna. Beberapa kondisi tidak biasa membuat sistem berada pada akurasi paling rendah, sehingga membuat aplikasi tidak berfungsi dengan baik. Oleh karena itu

penilitian lanjutan dapat dilakukan untuk memperbaiki kelemahan tersebut.

DAFTAR PUSTAKA

- [1] A. Abrahams. (2017, August 23), Your Cell Phone Is 10 Times Dirtier Than a Toilet Seat. Here's What to Do About It [Online]. Available: <https://time.com/4908654/cell-phone-bacteria/>
- [2] J. Ries. (2018, December 2018), Want to Avoid Dangerous Bacteria? Don't Use Public Touch Screen [Online]. Available: <https://www.healthline.com/health-news/want-to-avoid-dangerous-bacteria-dont-use-touch-screens#Good-hygiene-is-your-best-defense>
- [3] S. Doron, S. L. Gorbach, "Bacteria Infection: Overview", International Encyclopedia of Public Health, 10, 1016, 273 - 282, Aug. 2008.
- [4] S. Benkerzaz, Y. Elmer, A. Dennai, "A Study on Automatic Speech Recognition", Journal of Information Technology, 10, 3, 78 - 80, Aug. 2019.
- [5] L Gbadamosi, "Voice Recognition System using Template Matching", International Journal of Research in Computer Science, 3 ,5, 13-17, Sept. 2013.
- [6] A. V. Haridasa, R. Marimuthu, V. G. Sivakumar, "A critical review and analysis on techniques of speech recognition: The road ahead", International Journal of Knowledge-based and Intelligent Engineering Systems, 22, 10, 39 – 57, 2018.
- [7] R. Kavitha, N. Nachammai, R. Ranjani, J. Shafali, "Speech Based Voice Recognition System for Natural Language Processing", International Journal of Computer Science and Information Technologies, 5, 4, 5301 – 5305, 2014.
- [8] Woodford, Chris. (2020, June 3), Speech recognition software [Online]. Available: <https://www.explainthatstuff.com/voicerecognition.html>
- [9] N. Ifeanyi, O. Ikenna, O. Izunna, "Text - To - Speech Synthesis (TTS)", International Journal of Research in Information Technology, 2, 5, 154 - 156, May. 2014.
- [10] Woodford, Chris. 2020, April 8) Speech synthesizers [Online]. Available: <https://www.explainthatstuff.com/how-speech-synthesis-works.html>.
- [11] Jason Brownlee, "Foundations of Natural Language Processing", in Deep Learning for Develop Deep Learning Models for Natural Language in Python, Ed. Jason Brownlee, 2017, pp. 1 – 33.
- [12] Steven Bird, "Language Processing and Python", in Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit, Ed. Sebastopol: O'Reilly Media, 2009, pp. 1 – 35.