Mimic Human Speech in Bahasa Indonesia Using Speech Recognition and Speech Synthesis

Valens Prabagita Ivan Susilo
Department of Computer Science
President University
Cikarang, Bekasi, 17550, Indonesia
prabagita12@gmail.com

Abstrak

Sehari - hari orang menggunakan speech recognition dan speech synthesis secara tidak sadar. Teknologi ini sangat membantu dalam kegiatan mereka. Masing - masing teknologi sendiri dapat menghasilkan bermacam - macam software yang berkaitan dengan suara. Menggabungkan kedua teknologi akan menghasilkan lebih bermacam lagi software. Salah satu kombinasi itu adalah mimic speech, atau menirukan suara. Penelitian ini akan membhasa tentang Speech Recognition menggunakan Convolutional Neural Network sebagai model machine learning dan Speech Synthesis yang menggunakan Concatenative Synthesis dengan suku kata sebagai satuannya. Tujuan dari penelitian ini adalah untuk mengembangkan aplikasi untuk mengumpulkan, melatih, dan menirukan suara dalam Bahasa Indonesia. Pengguna dapat berpartisipasi dengan merekam suara mereka. Suara yang dikumpulkan akan dilatih untuk digunakan dalam aplikasi untuk mengenali suara. Setelah suara yang dikumpulkan dilatih, Pengguna dapat menggunakan aplikasi menirukan suara dengan mengidentifikasi suara dahulu lalu menghasilkan suara yang diinginkan dengan suara yang sudah teridentifikasi. Aplikasi untuk mengumpulkan dan menirukan suara akan dikembangkan dalam website.

1 Pendahuluan

"Ok Google, play some music". "Siri, what should I eat for lunch?". Sehari-hari orang menggunakan asisten buatan untuk membantu kegiatan mereka. Orang - orang sangat menyukai keberadaan asisten buatan ini karena dalam hitungan detik, apa yang mereka inginkan dapat terbenuhi. Secara kasat mata, orang - orang seperti berbicara dengan komputer atau ponsel mereka. Tetapi sebenarnya, speech recognition memegang perang penting didalamnya dengan bantuan machine learning. Google Assistance, Apple Siri, Microsoft Cortana, Amazon Alexa, dan asisten buatan lainnya mempunya ribuan hingga jutaan lebih suara yang dapat dianalisan dan mereka mendapatkannya dengan mudah dari suara orang - orang yang menggunakan aplikasi mereka dengan perizinan yang mereka terima.

Jika speech recognition adalah proses untuk mendapatkan data dari menganalisa suara, kebalikannya adalah speech synthesis, yaitu proses untuk membuat suara buatan. Maka dari itu speech recognition dikenal dengan istilah speech-to-text dan speech synthesis dengan isitilah text-to-speech. "Hey Cortana, read my email" adalah perintah agar asisten buatan menghasilkan suara, membacakan isi email. Masing - masing teknologi dapat menghasilkan bermacam - macam software yang berkaitan dengan suara. Menggabungkan kedua teknologi akan menghasilkan lebih bermacam lagi software. Salah satu kombinasi itu adalah mimic speech, atau menirukan suara.

Penelitian ini bertujuan untuk mengembangkan aplikasi untuk mengumpulkan suara - suara, melatih machine learning dengan suara yang terkumpul, dan menirukan suara dalam Bahasa Indonesia. Aplikasi untuk mengupulkan dan menirukan suara dikembangkan dalam website. Aplikasi dapat mengenali suara dan menghasil suara dari teks.

2 Batasan

Batasan aplikasi ini adalah sebagai berikut:

- Ada 9 suku kata yang dipilih untuk digunakan dalam aplikasi, a, i, na, ma, mu, di, ri, dan ku.
- Suara direkam dalam 1 detik dengan sample rate 16000 dan mono.
- Data speech recognition diambil dari suara yang direkam dalam Bahasa Indonesia.
- Data speech synthesis diambil dari suara yang disimpan, hasil dari analisa speech recognition.
- Aplikasi dikembangkan dalam website.

3 Metodologi

Pendekatan yang digunakan untuk mencapai tujuan penelitian ini menggunakan teknik speech synthesis concatenative synthesis, speech recognition Mel Frequency Cepstral Coefficients (MFCC), dan machine learning Convolutional Neural Network (CNN).

3.1 Concatenative Synthesis

Concatenative synthesis menghubungkan suara atau ucapan yang telah direkam. Concatenative synthesis merupakan cara termudah untuk menghasilkan suara buatan yang terdengar jelas dan alami. Salah satu aspek terpenting dalam concatenative synthesis adalah menentukan satuan suara yang benar.

Panjang dan pendek satuan mempunyai untung dan rugi masing - masing dalam penentuan ini. Dengan satuan yang panjang, lebih alami, dan titik konvergensi yang lebih sedikit, serta kontrol artikulasi yang baik terperoleh. Akan tetapi kebutuhan satuan dan memorinya akan besar. Dengan satuan yang pendek, memori yang sedikit sudah cukup, tetapi pengumpulan dan pelabelan sample suara akan lebih sulit dan kompleks [Hande, 2014].

Dalam sistem sekarang ini, satuan yang digunakan biasanya berupa kata, suku kata, *demisyllables*, fonem,

diphones, dan kadang pula triphones. Karena tidak ditemukan jumlah fonem atau satuan yang lebih kecil di Bahasa Indonesia, suku kata dapat menjadi pilihan terbaik sebagai satuan suara.

3.2 MFCC

MFCC adalah salah satu metode ekstraksi fitur yang paling umum digunakan dalam speech recognition yang diperkenalkan oleh Davis dan Mermelstein pada 1980-an [practicalcrypthography.com, 2012].

3.2.1 Framing dan Windowing

Langkah yang dapat dilakukan sebelum framing dan windowing adalah menerapkan filter pre-emphasis pada sinyal untuk memperkuat frekuensi tinggi. Filter pre-emphasis dapat diterapkan ke sinyal $\boldsymbol{x}$ menggunakan filter urutan pertama dalam persamaan berikut di mana nilai tipikal untuk koefisien filter ($\boldsymbol{\alpha}$) adalah 0.95 atau 0.97 [haythamfayek.com, 2016]:

$$\boldsymbol{y(t) = x(t) - \alpha x(t-1)}$$

Framing dilakukan karena sinyal audio terus berubah, sehingga untuk menyederhanakan banyak hal, dengan asumsi bahwa dalam skala waktu singkat sinyal audio tidak berubah. Biasanya, framing sinyal dilakukan dalam 20-40ms bagian (25ms standar). Selang setiap bagian biasanya 10ms, yang membuat bagian satu dengan yang lainnya memiliki beberapa data yang sama. Jika, file audio tidak terbagi menjadi bagian frame yang pas, sisa bagian tersebut akan diisi dengan nol.

Setelah memotong sinyal kedalam frame, windowing atau menerapkan fungsi window seperti Hamming window kesetiap bagian frame dapat dilakukan dimana, $\boldsymbol{0 \leq n \leq N-1}$, $\boldsymbol{N}$ adalah panjang frame:

$$\boldsymbol{w[n] = 0.54 - 0.46\, cos\left(\frac{2\pi n}{N-1}\right)}$$

3.2.3 Discrete Fourier Transform dan Power Spectrum

Menghitung setiap bagian dengan Discrete Fourier Transform (DFT) atau Fast Fourier Transform (FFT). Lalu dapat diikuti dengan menghitung power spectrum (periodogram). Periodogram menggunakan persamaan dimana, $\boldsymbol{x_i}$ adalah frame ke $\boldsymbol{i^{th}}$ dari sinyal $\boldsymbol{x}$ dan $\boldsymbol{N}$ adalah ukuran FFT hasil dari pangkat dua lebih besar ato sama dengan jumlah sampel dalam satu bagian frame:

$$\boldsymbol{P = \frac{|FFT(x_i)|^2}{N}}$$

3.2.4 Mel Filterbank

Periodogram spekral diperkirakan masi mengandung banyak informasi yang tidak diperlukan untuk speech recognition. Mengambil segumpal tempat periodogram dan mejumlahkannya dapat mendapatkan gambaran tentang seberapa banyak energi yang ada di berbagai daerah frekuensi. Hal inilah yang dilakukan oleh mel filterbank.

Menghitung mel filterbank dengan cara menerapkan filter segitiga, biasanya sebanyak 20-40 (26 atau 40 standar) filter, pada mel-scale sampai pada power spectrum untuk mengekstrak pita frequensi. Formula untuk mengkoversi antara Hertz ($\boldsymbol{f}$) dan Mel ($\boldsymbol{m}$) menggunakan persamaan berikut:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$
$$f = 700\left(10^{\frac{m}{2595}} - 1\right)$$

Setiap filter dalam filterbank berbentuk segitiga memiliki respons 1 pada frekuensi tengah dan menurun secara linear menuju 0 hingga mencapai frekuensi tengah dari dua filter yang berdekatan di mana responsnya adalah 0. Ukuran filter yang bagus adalah memulai filter dari 300Hz untuk yang lebih rendah dan hingga 8000Hz untuk frekuensi atas. Filterbank dapat dimodelkan dengan persamaan berikut:

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)}, & f(m-1) \le k < f(m) \\ 1, & k = f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)}, & f(m) < k \le f(m+1) \\ 0, & k > f(m+1) \end{cases}$$

3.2.5 Logarithm

Setelah menghitung mel filterbank, selanjutnya cukup menghitung logaritmanya. Secara umum, hal ini dilakukan untuk menggandakan volume suara yang dirasakan, memasukkan 8 kali lebih banyak energi ke dalamnya.

3.2.6 Discrete Cosine Transform

Langkah terakhir adalah menghitung Discrete Cosine Transform (DCT). DCT menghias energi yang artinya matriks kovarians diagonal dapat digunakan untuk memodelkan fitur. Hanya 12 koefisien DCT yang disimpan, Ini dilakukan, karena koefisien DCT yang lebih tinggi mewakili perubahan cepat yang menurunkan kinerja speech recogntion, jadi menghilangkan hingga hanya tersisa 12 akan mendapat peningkatan kecil.

3.3 Convolutional Neural Network

Convolutional neural network (CNN atau ConvNet) adalah salah satu varian model neural network yang dikenal untuk mengenali gambar [wikipedia.org, 2018]. Model ini dirancang untuk mengenali objek tidak peduli dimana permukaan objek itu. Model ini tidak harus mempelajari kembali perkiraan setiap permukaan yang mungkin muncul [Geitgey, 2016].

3.3.1 Convolution Layer

Convolution layer adalah lapisan untuk memasukan gambar yang telah di process sebelumnya atau output lapisan lain kedalam neural network kecil. Neural network ini memperlakukan setiap gambar atau output lapisan lain secara merata. Ia akan menandai jika sesuatu yang menarik muncul sebagai pembelajaran.

3.3.2 Max-pooling Layer

Max-pooling atau down sampling adalah lapisan untuk mengurangi output dengan cara menemukan nilai maksimum dalam output. Output dipecah menjadi kelompok dengan yang sama dan melangkah dari kelompok ke kelompok sampai mencakup seluruh output. Kemudian, setiap kelompok akan ditemukan nilai maksimum.

3.3.3 Fully-connected Layer

Fully-connected layer adalah lapisan untuk penalaran dan pertimbangan

tingkat tinggi dalam neural network yang padat untuk mengenali gambar.

4 Hasil Percobaan

Untuk mengevaluasi efektivitas metode yang diusulkan di bagian sebelumnya dalam aplikasi ini, percobaan dilakukan untuk memastikan aplikasi berjalan dengan baik.

Karena speech synthesis hanya menggunakan metode concatenative synthesis, atau menggabungkan, makana evaluasi akan fokus pada speech recognition. Dataset yang digunakan selama evaluasi sebanyak 10000 suara laki - laki dan perempuan, yang masing - masing 500 suara pada setiap suku kata pada suasana yang tidak berisik dan masing - masing 500 suara pada suara tidak dikenal atau suara acak. Hasil yang dianggap baik ditunjukan dari akurasi lebih dari 75% dengan suku kata yang bersangkutan. Evaluasi juga dilakukan di suku kata diluar batasan dan suara acak seperti kondisi diam, suku kata o dan mi.

Ada 4 penguji dalam skenario ini. Tester 1 yaitu pengguna laki - laki yang suaranya telah digunakan dalam pembelajaran machine learning. Tester 2 yaitu pengguna perempuan yang suaranya telah digunakan dalam pembelajaran machine learning. Tester 3 yaitu pengguna laki - laki yang suaranya belum pernah digunakan dalam pembelajaran machine learning. Tester 4 yaitu pengguna perempuan yang suaranya belum pernah digunakan dalam pembelajaran machine learning. Setiap penguji akan melakukan evaluasi dalam keadaan seperti berikut:

- Suasana berisik (Musik keras atau orang sekitar sedang mengobrol).
- Suasana lumayan berisik (Suara hujan atau suara dari ruangan sebelah).
- Suasana tidak berisik.

Tabel 1: Hasil Tester 1 pada suasana berisik.

No	Spoken Syllable	Noisy Background			Correct Result
		1	2	3	
1	a	52 *(a)*	100 *(a)*	100 *(unknown)*	1/3
2	i	62 *(na)*	85 *(ri)*	96 *(ma)*	0/3
3	na	100 *(unknown)*	86 *(kan)*	100 *(unknown)*	0/3
4	ma	100 *(a)*	84 *(unknown)*	100 *(a)*	0/3
5	mu	99 *(kan)*	71 *(ku)*	88 *(unknown)*	0/3
6	di	59 *(mu)*	100 *(ri)*	97 *(ri)*	0/3
7	ri	50 *(na)*	100 *(ri)*	92 *(a)*	1/3
8	ku	100 *(ku)*	85 *(a)*	100 *(kan)*	1/3
9	kan	72 *(unknown)*	61 *(kan)*	79 *(kan)*	3/3
10	Unknown 1 *(silent)*	99 *(unknown)*	100 *(a)*	99 *(unknown)*	2/3
11	Unknown 2 *(o)*	100 *(na)*	100 *(a)*	100 *(ku)*	0/3
12	Unknown 3 *(mi)*	100 *(a)*	51 *(kan)*	96 *(ma)*	1/3

Tabel 2: Hasil Tester 1 pada suasana lumayan berisik.

No	Spoken Syllable	Semi Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	100 *(a)*	100 *(a)*	3/3
2	i	100 *(i)*	36 *(ri)*	100 *(i)*	2/3
3	na	97 *(mu)*	48 *(ma)*	99 *(na)*	1/3
4	ma	49 *(i)*	98 *(ri)*	79 *(mu)*	0/3
5	mu	55 *(ri)*	99 *(ku)*	50 *(ri)*	0/3
6	di	57 *(i)*	100 *(di)*	100 *(i)*	1/3
7	ri	100 *(ri)*	75 *(ri)*	100 *(ri)*	3/3
8	ku	92 *(ku)*	68 *(ku)*	91 *(ku)*	2/3
9	kan	100 *(kan)*	94 *(na)*	100 *(kan)*	2/3
10	Unknown 1 *(silent)*	100 *(a)*	99 *(unknown)*	100 *(ma)*	2/3
11	Unknown 2 *(o)*	100 *(ku)*	89 *(ku)*	72 *(ku)*	1/3
12	Unknown 3 *(mi)*	100 *(ri)*	94 *(ri)*	100 *(ri)*	0/3

Tabel 3: Hasil Tester 1 pada suasana tidak berisik.

No	Spoken Syllable	Not Noisy Background			Correct Result
		1	2	3	
1	a	100 (a)	100 (a)	100 (a)	3/3
2	i	94 (i)	48 (kan)	94 (i)	2/3
3	na	80 (na)	99 (na)	100 (na)	3/3
4	ma	96 (di)	90 (i)	57 (di)	0/3
5	mu	100 (i)	86 (ku)	95 (i)	0/3
6	di	99 *(ri)*	100 *(i)*	80 *(ri)*	0/3
7	ri	100 *(ri)*	96 *(ri)*	91 *(ri)*	3/3
8	ku	85 *(ku)*	96 *(ku)*	100 *(ku)*	3/3
9	kan	92 *(kan)*	100 *(kan)*	100 *(kan)*	3/3
10	Unknown 1 *(silent)*	98 *(unknown)*	98 *(unknown)*	98 *(unknown)*	3/3
11	Unknown 2 *(o)*	100 *(ku)*	98 *(ku)*	65 *(ku)*	1/3
12	Unknown 3 *(mi)*	54 *(mu)*	100 *(ri)*	67 *(mu)*	2/3

Tabel 4: Hasil Tester 2 pada suasana berisik.

No	Spoken Syllable	Noisy Background			Correct Result
		1	2	3	
1	a	99 *(unknown)*	99 *(a)*	99 *(a)*	2/3
2	i	82 *(unknown)*	98 *(ma)*	99 *(ri)*	0/3
3	na	52 *(ri)*	99 *(ma)*	61 *(ma)*	0/3
4	ma	50 *(unknown)*	37 *(unknown)*	95 *(ri)*	0/3
5	mu	99 *(unknown)*	52 *(di)*	99 *(unknown)*	0/3
6	di	99 *(i)*	99 *(unknown)*	99 *(unknown)*	0/3
7	ri	86 *(i)*	20 *(kan)*	57 *(di)*	0/3
8	ku	99 *(ku)*	68 *(kan)*	90 *(unknown)*	1/3
9	kan	84 *(kan)*	99 *(kan)*	78 *(a)*	2/3
10	Unknown 1 *(silent)*	100 *(unknown)*	100 *(unknown)*	100 *(unknown)*	3/3
11	Unknown 2 *(o)*	99 *(ma)*	99 *(ku)*	99 *(ku)*	0/3
12	Unknown 3 *(mi)*	36 *(di)*	96 *(mu)*	99 *(di)*	1/3

Tabel 5: Hasil Tester 2 pada suasana lumayan berisik.

No	Spoken Syllable	Semi Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	100 *(unknown)*	99 *(a)*	2/3
2	i	92 *(ri)*	97 *(unknown)*	96 *(ri)*	0/3
3	na	83 *(ma)*	62 *(unknown)*	75 *(kan)*	0/3
4	ma	87 *(unknown)*	64 *(ma)*	90 *(ng)*	2/3
5	mu	76 *(ma)*	78 *(ku)*	66 *(di)*	0/3
6	di	58 *(mu)*	48 *(mu)*	30 *(kan)*	0/3
7	ri	99 *(kan)*	89 *(kan)*	99 *(ri)*	1/3
8	ku	99 *(unknown)*	99 *(ku)*	88 *(unknown)*	1/3
9	kan	99 *(kan)*	65 *(kan)*	85 *(kan)*	2/3
10	Unknown 1 *(silent)*	100 *(unknown)*	79 *(a)*	99 *(ri)*	1/3
11	Unknown 2 *(o)*	99 *(a)*	82 *(ku)*	99 *(ng)*	0/3
12	Unknown 3 *(mi)*	89 *(kan)*	35 *(ku)*	96 *(ri)*	1/3

Tabel 6: Hasil Tester 2 pada suasana tidak berisik.

No	Spoken Syllable	Not Noisy Background			Correct Result
		1	2	3	
1	a	88 *(a)*	99 *(a)*	99 *(a)*	3/3
2	i	99 *(i)*	99 *(i)*	52 *(i)*	2/3
3	na	97 *(a)*	54 *(kan)*	99 *(a)*	0/3
4	ma	99 *(ng)*	99 *(ng)*	98 *(ng)*	0/3
5	mu	81 *(i)*	95 *(i)*	44 *(i)*	0/3
6	di	99 *(i)*	36 *(i)*	96 *(i)*	0/3
7	ri	47 *(ri)*	32 *(unknown)*	48 *(i)*	0/3
8	ku	98 *(ku)*	67 *(kan)*	99 *(ku)*	2/3
9	kan	84 *(ng)*	85 *(ng)*	87 *(ng)*	0/3
10	Unknown 1 *(silent)*	99 *(unknown)*	99 *(unknown)*	100 *(unknown)*	3/3
11	Unknown 2 *(o)*	100 *(ku)*	99 *(ku)*	95 *(ku)*	0/3
12	Unknown 3 *(mi)*	67 *(ri)*	78 *(di)*	77 *(i)*	1/3

Tabel 7: Hasil Tester 3 pada suasana berisik.

No	Spoken Syllable	Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	99 *(a)*	100 *(a)*	3/3
2	i	71 *(ma)*	99 *(ri)*	99 *(ri)*	0/3
3	na	70 *(ku)*	99 *(a)*	57 *(unknown)*	0/3
4	ma	99 *(a)*	100 *(a)*	100 *(a)*	0/3
5	mu	74 *(mu)*	96 *(kan)*	74 *(mu)*	0/3
6	di	99 *(ri)*	96 *(ri)*	89 *(ri)*	0/3
7	ri	100 *(unknown)*	90 *(ng)*	86 *(a)*	0/3
8	ku	98 *(kan)*	64 *(mu)*	99 *(a)*	0/3
9	kan	99 *(ng)*	100 *(kan)*	98 *(kan)*	2/3
10	Unknown 1 *(silent)*	99 *(a)*	99 *(a)*	100 *(a)*	0/3
11	Unknown 2 *(o)*	92 *(ng)*	84 *(ng)*	61 *(mu)*	1/3
12	Unknown 3 *(mi)*	64 *(kan)*	53 *(ng)*	99 *(ri)*	2/3

Tabel 8: Hasil Tester 3 pada suasana lumayan berisik.

No	Spoken Syllable	Semi Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	99 *(a)*	99 *(a)*	3/3
2	i	99 *(ri)*	90 *(di)*	92 *(di)*	0/3
3	na	100 *(kan)*	99 *(kng)*	92 *(ma)*	0/3
4	ma	99 *(a)*	100 *(kan)*	100 *(kan)*	0/3
5	mu	99 *(mu)*	99 *(mu)*	65 *(ku)*	2/3
6	di	99 *(ri)*	100 *(ri)*	99 *(ri)*	0/3
7	ri	95 *(kan)*	76 *(i)*	78 *(di)*	0/3
8	ku	99 *(a)*	96 *(a)*	100 *(a)*	0/3
9	kan	99 *(a)*	96 *(a)*	99 *(a)*	0/3
10	Unknown 1 *(silent)*	99 *(ri)*	73 *(di)*	60 *(unknown)*	2/3
11	Unknown 2 *(o)*	100 *(a)*	99 *(ku)*	100 *(a)*	0/3
12	Unknown 3 *(mi)*	99 *(ng)*	99 *(ri)*	99 *(i)*	0/3

Tabel 9: Hasil Tester 3 pada suasana tidak berisik.

No	Spoken Syllable	Not Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	100 *(a)*	100 *(a)*	3/3
2	i	92 *(ri)*	95 *(ri)*	41 *(ku)*	0/3
3	na	99 *(na)*	74 *(ku)*	80 *(a)*	1/3
4	ma	59 *(a)*	84 *(ku)*	48 *(kan)*	0/3
5	mu	99 *(mu)*	99 *(ri)*	99 *(ri)*	1/3
6	di	100 *(ri)*	99 *(i)*	99 *(i)*	0/3
7	ri	99 *(ri)*	96 *(ri)*	94 *(i)*	2/3
8	ku	73 *(di)*	100 *(mu)*	99 *(mu)*	0/3
9	kan	100 *(kan)*	99 *(kan)*	96 *(kan)*	3/3
10	Unknown 1 *(silent)*	98 *(unknown)*	98 *(unknown)*	98 *(unknown)*	3/3
11	Unknown 2 *(o)*	99 *(ku)*	68 *(mu)*	100 *(ku)*	1/3
12	Unknown 3 *(mi)*	99 *(ri)*	93 *(i)*	100 *(ri)*	0/3

Tabel 10: Hasil Tester 4 pada suasana berisik.

No	Spoken Syllable	Noisy Background			Correct Result
		1	2	3	
1	a	100 *(unknown)*	99 *(unknown)*	94 *(ma)*	0/3
2	i	97 *(unknown)*	99 *(ri)*	99 *(ri)*	0/3
3	na	100 *(na)*	99 *(ri)*	99 *(ma)*	1/3
4	ma	93 *(na)*	99 *(ma)*	100 *(na)*	1/3
5	mu	55 *(ri)*	99 *(unknown)*	43 *(unknown)*	0/3
6	di	99 *(ri)*	54 *(i)*	99 *(ma)*	0/3
7	ri	97 *(ri)*	74 *(ri)*	100 *(ri)*	2/3
8	ku	68 *(ma)*	92 *(ku)*	52 *(52)*	1/3
9	kan	48 *(ku)*	22 *(ku)*	99 *(a)*	0/3
10	Unknown 1 *(silent)*	100 *(unknown)*	100 *(unknown)*	100 *(unknown)*	3/3
11	Unknown 2 *(o)*	100 *(a)*	100 *(a)*	100 *(a)*	0/3
12	Unknown 3 *(mi)*	55 *(mu)*	99 *(ri)*	94 *(i)*	1/3

Tabel 11: Hasil Tester 4 pada suasana lumayan berisik.

No	Spoken Syllable	Semi Noisy Background			Correct Result
		1	2	3	
1	a	73 *(a)*	91 *(unknown)*	99 *(unknown)*	0/3
2	i	81 *(ri)*	99 *(i)*	100 *(ri)*	1/3
3	na	100 *(ma)*	99 *(unknown)*	99 *(unknown)*	0/3
4	ma	99 *(ma)*	99 *(na)*	78 *(ma)*	2/3
5	mu	81 *(unknown)*	99 *(unknown)*	99 *(unknown)*	0/3
6	di	100 *(ri)*	99 *(i)*	99 *(i)*	0/3
7	ri	85 *(ri)*	99 *(ri)*	98 *(ri)*	3/3
8	ku	96 *(ku)*	89 *(ku)*	99 *(a)*	2/3
9	kan	100 *(a)*	99 *(a)*	64 *(ma)*	0/3
10	Unknown 1 *(silent)*	99 *(unknown)*	99 *(ri)*	100 *(unknown)*	2/3
11	Unknown 2 *(o)*	99 *(a)*	100 *(a)*	99 *(a)*	0/3
12	Unknown 3 *(mi)*	43 *(a)*	59 *(ri)*	99 *(unknown)*	3/3

Tabel 12: Hasil Tester 4 pada suasana tidak berisik.

No	Spoken Syllable	Not Noisy Background			Correct Result
		1	2	3	
1	a	99 *(unknown)*	75 *(a)*	100 *(a)*	2/3
2	i	88 *(i)*	99 *(i)*	99 *(ri)*	2/3
3	na	99 *(na)*	99 *(ma)*	99 *(ma)*	1/3
4	ma	55 *(ma)*	72 *(kan)*	99 *(na)*	0/3
5	mu	99 *(unknown)*	94 *(unknown)*	28 *(i)*	0/3
6	di	89 *(i)*	99 *(ri)*	99 *(ri)*	0/3
7	ri	99 *(ri)*	100 *(ri)*	99 *(ri)*	3/3
8	ku	39 *(kan)*	99 *(ku)*	58 *(mu)*	1/3
9	kan	100 *(a)*	99 *(a)*	100 *(a)*	3/3
10	Unknown 1 *(silent)*	40 *(unknown)*	24 *(unknown)*	60 *(unknown)*	3/3
11	Unknown 2 *(o)*	100 *(a)*	99 *(a)*	100 *(a)*	0/3
12	Unknown 3 *(mi)*	60 *(ri)*	100 *(i)*	99 *(ri)*	1/3

Meskipun suasana berisik dan lumayan berisik, setiap tester tidak menunjukuan hasil yang cukup baik. Masih dapat memprediksi sekitar 2 atau 3 suku kata dengan benar dan akurasi yang tinggi. Suasana tidak berisik juga tidak menjamin hasil yang sempurna bahkan pada penguji 1 dan 2. Tetapi, hasilnya lebih baik daripada suasana berisik dan lumayan berisik. Selain itu, pengurangan suasana latar belakang tidak diterapkan ketika proses ekstrasi yang dilakukan sebelum pembelajaran.

Penguji 1 dan 2 seringkali mendapat hasil yang lebih baik dari pada penguji 3 dan 4. Tetapi, pada saat tertentu penguji 3 dan 4 mendapat hasil yang lebih baik. Ini bisa terjadi dikarenakan suasana latar belakangnya atau pun juga kemampuan mikrofon untuk merekam suara.

Suku kata ma, mu, dan di, serta suku kata acak o dan mi memiliki hasil yang buruk hampir setiap waktu. Suku kata acak o dan mi dikarekanan pada pembelajarannya suara acak berisikan kebanyakan suasana latar belakang dari pada suku kata o dan mi. Model machine learning tidak dapat

memprediksi suku kata ma, mu, dan di dengan baik. Seringkali model memprediksi dengan benar tapi dengan akurasi yang rendah atau memprediksi suku kata relatif, contoh hasil ri atau i pada suku kata di. Hasil ini dapat ditingkatkan dengan menambahkan data lagi untuk dipelajari karena data yang digunakan terhitung masih relatif sedikit. Model machine learning yang optimum juga dapat meningkatkan hasil prediksi dan akurasi.

5 Diskusi

Pada bagian ini, ada beberapa diskusi mengapa metode yang diusulkan digunakan untuk penelitian ini. Poin - poin penting adalah:

- Concatenative Synthesis untuk Speech Synthesis
- MFCC untuk Speech Recognition
- CNN untuk Machine Learning model

Concatenative synthesis adalah apa yang mimic speech butuhkan, rekaman suara yang diambil dari suara yang dianalisa di simpan dan ketika ingin suara dimuat dan di gabungkan. sesuai teks yang dimasukan.

Setiap proses pada MFCC memiliki alasannya tersendiri. Jika di simpulkan MFCC digunakan dan dipilih karena MFCC bertujuan untuk meniru perspektif suara telinga manusia *non-linear* dengan menjadi lebih diskriminatif pada frekuensi yang lebih rendah dan kurang diskriminatif pada suara yang lebih tinggi.

Dua alasan utama menggunakan CNN. Pertama, concatenative synthesis menggunakan satuan suara. Tiap satuan suara adalah bagian kecil dari suatu kalimat atau kata. CNN dianggap sebagai pendekatan optimal untuk menganalisa *small-footprint keyword*, yakni satuan suara itu sendiri, dari pada pendekatan machine learning yang lain [Sainath & Parada, 2015]. Kedua, proses ekstrasi MFCC dapat di plot menjadi spektogram. Dan CNN adalah pendekatan paling umum untuk menyelesaikan masalah tentang visual dan gambar, termasuk spektogram.

6 Conclusion

Ada beberapa kesimpulan yang dapat diperoleh dari penilitian ini. Pertama, aplikasi ini memungkinan pengenalan suara dalam Bahasa Indonesia dari rekaman audio dengan baik menggunakan CNN dan MFCC. Kedua, aplikasi ini memungkinkan untuk menghasilkan suara dalam Bahasa Indonesia sesuai teks dengan baik menggunakan concatenative synthesis. Ini berarti speech recognition dan speech synthesis bekerja dengan baik dan menghasilkan mimic speech.

Di masa yang akan datang, penilitian lebih lanjut dalam speech recognition pada bagian meningkatkan model machine learning. Tidak ada benar dalam memodelkan machine learning, tetapi selalu ada model yang optimum untuk mendapatkan prediksi dan akurasi yang paling baik. Dalam speech synthesis dengan menambakan proses untuk menghilangkan bagian hening atau suara tidak perlu dari rekaman suara dan juga menghilangkan atau mengurangi suasana latar belakang akan menghasilkan pembuatan suara yang lebih lancar dan baik didengar.

Dalam Bahasa Indonesia, penelitian untuk mementukan fonem Bahasa Indonesia dapat menjadi peningkatan yang signifikan dalam speech recognition dan speech synthesis karena aplikasi ini menggunakan suku kata sebagai satuan suara.

7 Ucapan Terima Kasih

Penulis hendak mengucapkan ucapan terima kasih kepada Bapak Tjong Wan Sen sebagai penasihat skripsi untuk saran dan dukungan selama proses skripsi ini dilaksanakan. Dan juga kepada dosen - dosen yang lain untuk dukungan mereka, ilmu, dan pengalaman selama kuliah.

Referensi

[Hande, 2014] Hande, S. S. (2014). A Review on Speech Synthesis an Artificial Voice Production. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 8.

[practicalcrypthography.com, 2012] practicalcryptography.com. (2012). *Mel Frequency Cepstral Coefficient (MFCC) tutorial*. Retrieved from Practical Cryptography: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[haythamfayek.com, 2016] haythamfayek.com. (2016, April 21). *Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between*. Retrieved from Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

[wikipedia.org, 2018] wikipedia.org. (2018, December 26). *Convolutional neural network*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Convolutional_neural_network

[Geitgey, 2016] Geitgey, A. (2016, Juny 14). *Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks*. Retrieved from Medium: https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721

[Sainath & Parada, 2015] Sainath, T. N., & Parada, C. (2015). *Convolutional Neural Networks for Small-footprint Keyword Spotting.* New York, NY, U.S.A: Google, Inc.