BAB 1. PENDAHULUAN

1.1. Latar Belakang Permasalahan

Metode *Automatic Speech Recognition* (ASR) seperti *Hidden Markov Model (HMM)* dan *Gaussian Mixture Model* (GMM) dapat mengklasifikasikan setiap ucapan menjadi teks dengan cara menghitung probabilitas maksimum sehingga didapatkan *model* / *states* yang diinginkan. Namun kekurangan GMM adalah pendekatan statistik tidak efektif memodelkan data yang berada pada non-linear manifold dalam space data. [1]

Menurut Joe Tebelskis [2], pengucapan dapat sangat bervariasi tergantung dari aksen, lafal, artikulasi, dinamika, nasalitas, tangga nada, volume, dan kecepatan. Adapun kategori kondisi yang mempengaruhi tingkat akurasi sistem ASR adalah jumlah kosakata, *speaker dependence* vs. *independence* (satu speaker vs. lebih), jeda / ketidaksinambungan suara, struktur bahasa, suara yang tidak memiliki arti seperti "uh" / "um", noise dari lingkungan sekitar.

Masalah variasi ini menjadi perhatian khusus, melihat keanekaragaman bahasa, dialek, dan aksen yang dimiliki Indonesia. Menurut Wikipedia, terdapat lebih dari 700 bahasa daerah di Indonesia. Walaupun bahasa Nasional adalah Indonesia, setiap daerah tidak bisa terlepas dari dialek, aksen, dan bahasa daerahnya. Bahkan menurut statistik sensus tahun 2000 terdapat 84.3 juta penduduk Indonesia mayoritas adalah Bahasa Jawa. Kedua diikuti dengan Bahasa Sunda sekitar 34 juta penduduk. Ketiga diikuti dengan Bahasa Madura sekitar 13.6 juta penduduk.

Untuk membuat Sistem *Speech Recognition*, metode HMM / GMM dalam masalah variasi yang disebutkan diatas dapat diminimalisasi tingkat errornya / dimaksimalkan tingkat akurasinya dengan menggunakan Deep Neural Networks. Menurut Baidu Research [4], sistem Recurrent Neural Network (RNN) dan Connectionist Temporal Classification (CTC) [5] menunjukkan peningkatan yang cukup signifikan yaitu mereduksi tingkat rata-rata error dalam Bahasa Inggris hingga 43%. Menurut Geofrey Hinton, et all [1] sistem ASR menggunakan

pendekatan Neural Network memiliki keunggulan saat dibandingkan dengan GMM dengan pengambilan fitur model akustik untuk mengatasi variasi dataset yang sangat besar dan memiliki kosakata yang besar.

ASR menggunakan neural network sudah banyak dilakukan dalam penelitian [6,7,8,9] dan menunjukan hasil yang mengingkatkan performa speech recognition terutama pada model akustik. Baidu adalah salaDalam penelitiannya Baidu memberi nama Deep Speech. Deep Speech terdapat versi 1 [3] dan 2 [4]. Deep Speech menggunakan spectrogram dalam pengambilan fitur dari signal suara. Fitur ini di hitung dengan weight dan bias ke dalam Recurrent Neural Network. Untuk pengklasifikasian urutan huruf yang dihasilkan, digunakan Connectionist Temporal Classification [5]. Dalam implementasinya, Baidu menggunakan framework yang dibuat sendiri bernama Warp-CTC. Framework ini menggabungkan Torch (Lua) dan TensorFlow (Python). Penelitian Baidu menginspirasi penelitian ini untuk mengutamakan metode RNN dan CTC dalam mengenali pola dataset suara dalam Bahasa Indonesia. Sistem akan ditraining dengan training data yang mengambil sample suara mahasiswa dari 3 daerah berbeda di Universitas Kristen Petra. Sistem ditraining dengan arsitektur model yang dijelaskan di bagian Metodologi Penelitian. Sistem *voice recognizer* yang sudah dibuat sesuai dengan model yang sudah di training akan menghasilkan parameter weight dan bias.

Berkaitan dengan sistem pengenalan suara (ASR), komunikasi menjadi masalah utama jika pihak penerima pesan tidak dapat menerima pesan dengan baik. Salah satu masalah yang dihadapi adalah ketika orang awas berkomunikasi dengan orang tunarungu. Sehingga untuk relevansi penelitian selanjutnya, sistem pengenalan suara ini dapat dikembangkan untuk menjadi aplikasi yang secara universal dapat digunakan orang Tunarungu di Indonesia.

1.2. Perumusan Masalah

Adapun perumusan masalah dari skripsi ini adalah:

- Bagaimana caranya untuk memodelkan machine learning agar dapat mempelajari variasi pada aksen suara berbahasa Indonesia.

- Bagaimana performa akurasi dalam mengklasifikasi suara dengan tidak mengorbankan kecepatan.

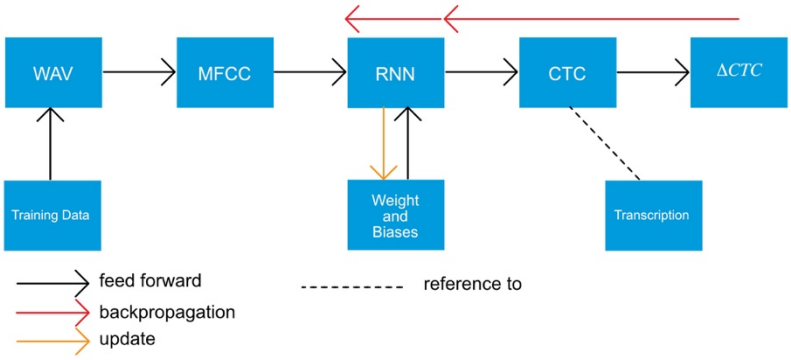
1.3. Tujuan Skripsi

Penelitian ini bertujuan untuk menerapkan penggunaan metode RNN dan CTC dalam mengenali pola dataset suara dalam Bahasa Indonesia. Meninjau arsitektur layer pada RNN dan parameter fitur berbeda dengan Baidu yaitu MFCC. Meninjau penggunaan gradient descent untuk peng-update-an weight dan bias. Parameter uji dengan menghitung WER (Word Error Rate) dan Kecepatan dalam mengenali kata - kata sebagai tolak ukur penelitian selanjutnya.

1.4. Ruang Lingkup

- Studi Literatur
  1. Mel-Frequency Cepstral Coefficient
  2. Deep Speech Baidu
  3. Connectionist Temporal Classification
- Pengumpulan Data

  Pengumpulan data suara diambil sample mahasiswa dari 3 daerah berbeda di Universitas Kristen Petra. Kamus Bahasa Indonesia diambil dari source internet (*github*).
- Input - Proses – Output



Gambar 1.1. Diagram alur secara keseluruhan dalam *training dataset*.

1. *Training*

Input :

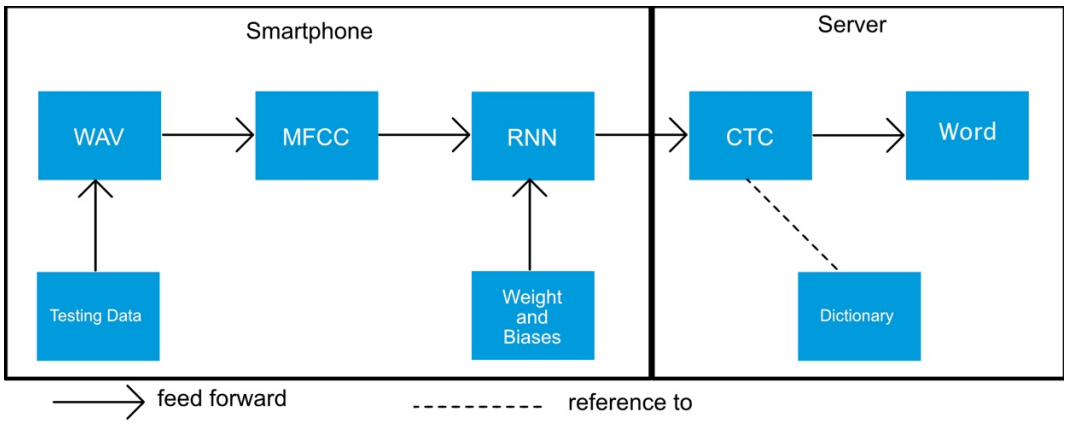
Training akan dijalankan melalui *server* dengan mengambil *training dataset* dengan merekam suara pembicara melalui *smartphone* sebanyak 30 orang. Setiap pembicara membacakan teks transkrip yang dibuat dengan rata - rata total 5 menit pembicaraan. Setiap *raw* suara dengan *format* WAV, diambil *frame sample* 20ms. Sehingga didapatkan 300.000 *frame*. Konfigurasi suara 16kHz dan 16-bit.

Proses :

1. Suara yang terekam di Android *smartphone* disimpan sebagai *training dataset*.
2. Pengambilan fitur dari *raw* suara menjadi MFCC (*Mel-Frequency Ceptrals Coeficient*).
3. Pengolahan fitur untuk dimasukkan ke dalam sistem *RNN* (*Recurrent Neural Network*) dihitung dengan parameter *weight* dan *bias* untuk mengklasifikasi grafem *(graphneme)* / huruf.
4. Menghitung CTC loss dengan mereferensikan urutan huruf yang dihasilkan *RNN* terhadap teks transkrip bacaan.
5. Proses diulang terus dengan membagi 3 *batch* (100.000 *frame*) dengan iterasi 100,500,1000, dan 2000 kali.

Output :

Dihasilkan model dengan *weight* dan *bias*. *Weight* dan *bias* yang di training.



Gambar 1.2. Diagram alur untuk *testing dataset*.

2. *Testing*

   Input :

   Suara diambil melalui *smartphone* Android berupa *buffer bit*.

   Proses :

   1. *Buffer* yang diterima di ambil per 20 *miliseconds sample*.
   2. *Buffer* dimasukkan ke dalam sistem RNN untuk di komputasi dengan *weight* dan *bias* yang sudah di training.
   3. *Model* yang sudah ditrain akan mengklasifikasi *frame* untuk menghasilkan urutan karakter.
   4. Urutan karakter dikirim ke *server* untuk diklasifikasi dengan CTC.
   5. CTC akan mengklasifikasikan urutan karakter sesuai kamus dalam mem-validasi sebuah kata.
   6. Server mengembalikan ke *smartphone*.

   Output :

   Kata yang dikenali oleh sistem RNN dan CTC.

- Spesifikasi Teknis

  Bahasa Pemograman yang digunakan untuk training : Python. Bahasa Pemograman untuk mobile device Android : Java / C Library / Framework yang digunakan untuk training : Warp-CTC, Numpy, Scipy, dan Matplotlib, Tensorflow. Library / Framework untuk mobile device Android : AAudio / OpenSL ES, ARM NEON / Project Ne10. Bahasa yang dikenali : Bahasa Indonesia

1.5. Tinjauan Pustaka

1. MFCC (*Mel-Frequency Cepstral Coeficient*)

   Data kolektif dari MFC (*Mel-frequency cepstrum*). MFC adalah data reseprentasi dari *short-term power spectrum* (spektrum energy) terhadap suara berdasarkan transformasi *linear cosine* dari *log power spectrum* pada *non-linear mel scale* frekuensi. Transformasi raw audio menjadi MFCC adalah sebagai berikut :

1. Diambil *sample frame* berkisar 20-40 ms. Berarti jika *sample rate* 16kHZ dan diambil *frame* 20 ms maka akan diambil 320 *sample*. Kita notasikan setiap *frame* dengan $s_i(n)$ dengan n adalah *sample* 0-320, i adalah range hingga *frame* terakhir.
2. Hitung tiap frame ke dalam *Discrete Fourier Transform*, mengikuti rumus (1.1).

$$S_i(k) = \sum_{n=0}^{N} S_i(n)h(n)e^{\frac{-2\pi kn}{N}}, 1 \leq k \leq K \quad (1.1)$$

   $S_i(k)$ = hasil DFT pada frame ke-i

   N = ukuran *window*

   $h(n)$ = *hamming window*

   K = panjang DFT
3. Hitung *periodogram-based power spectrum* setiap frame dengan rumus (1.2).

$$P_i(k) = \frac{1}{N}|S_i(k)|^2 \quad (1.2)$$

   $P_i(k)$ = hasil periodogram pada frame ke-i
4. Hitung semua energi filterbank dengan algoritma berikut :

   Hitung *lower* dan *upper* frekuensi dari sample rate terhadap Mel scale dengan rumus (1.3).

$$M(f) = 1125 \ln\left(1 + \frac{f}{700}\right) \quad (1.3)$$

   $M(f)$ = hasil skala mel pada frekuensi f

   f = besaran frekuensi

   Sehingga dengan rumus diatas kita dapat menghasilkan *lower* dan *upper*. Dengan n filterbank, kita akan membuat range mulai dari *lower* hingga *upper*, sehingga kita akan mendapat *range of mel scale*. Range tersebut akan dikonversikan kembali terhadap frekuensi dengan rumus (1.4).

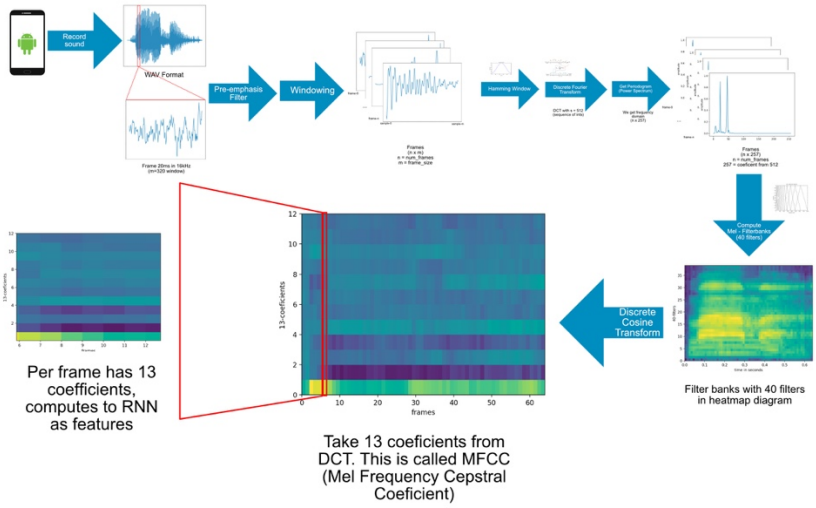
$$M^{-1}(m) = 700(\exp\left(\frac{m}{1125}\right) - 1 \quad (1.4)$$

   $M^{-1}(m)$ = hasil invers dari skala mel menjadi frekuensi

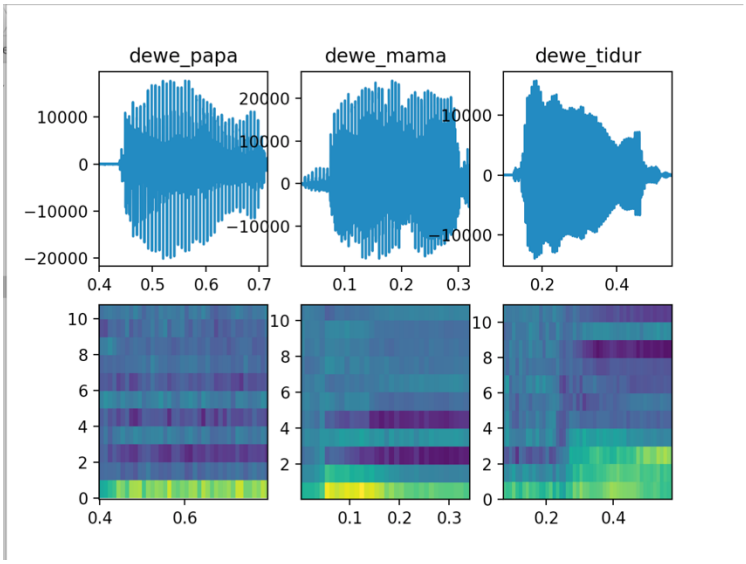
   m = besaran skala mel

Dari hasil range frekuensi di atas, dibuat ke dalam filterbanks.

5. Dari fasil filterbanks dilakukan Discrete Cosine Transform (DCT), diambil 13 index pertama daripada hasil DCT untuk dijadikan sebagai MFCC. Keseluruhan algoritma dapat dilihat pada Gambar 1.3



Gambar 1.3. Proses pengambilan fitur dari RAW *audio* (WAV) menjadi MFCC. Contoh Data Representasi fitur MFCC dari tiga input wav dengan sample rate berbeda - beda (lihat gambar 1.4).



Gambar 1.4. Data representasi berdasar *wavelength* (atas) dan MFCC dengan 12 koefisien (bawah) dengan konfigurasi dari kiri ke kanan 44.1kHz, 16kHz, 8kHz.

2. *Recurrent Neural Networks*

Recurrent Neural Networks dipakai dalam kasus yang berurutan (*sequence*), biasanya relatif terhadap waktu. Sehingga penggunaan daripada arsitektur RNN sangat tepat digunakan untuk *Speech Recognition* yang linear terhadap waktu. Berikut arsitektur RNN yang digunakan oleh Baidu Research pada Deep Speech :

Dinotasikan training set $X = \{(x_i^{(i)}, y_i^{(i)}),\ldots\}$ $X = \left\{\left(x_i^{(i)}, y_i^{(i)}\right)\right\}$, dimana x adalah inputan relatif terhadap waktu t *(time-series)* dan y adalah target grafem / huruf yang dilabelkan.Untuk 3-layer pertama merupakan *non-recurrent layer* dimana setiap data adalah *independent*, dihitung rumus (1.5) dengan model h :

$$h_t^{(l)} = g\left(W^{(l)} h_t^{(l-1)} + b^{(l)}\right) \tag{1.5}$$

$h_t^{(l)}$ = hidden layer ke-l dengan waktu t

$W^{(l)}$ = *weigth* pada layer ke-l

$b^{(l)}$ = *bias* pada layer ke-l

dimana $g(z) = min\{max\{0, z\}, 20\}$, sehingga $g(z)$ adalah *clipped rectified-linear* (ReLu) sebagai *activation function*. Untuk layer ke-4 menggunakan *bi-directional recurrent layer*, dimana rumus perhitungan untuk model h dimana $h^{(f)}$sebagai *forward* pada rumus (1.6) dan $h^{(b)}$ sebagai *backward* pada rumus (1.7).

$$h_t^{(f)} = g\left(W^{(4)} h_t^{(3)} + W_r^{(f)} h_{t-1}^{(f)} + b^{(4)}\right) \tag{1.6}$$

$$h_t^{(b)} = g\left(W^{(4)} h_t^{(3)} + W_r^{(b)} h_{t-1}^{(b)} + b^{(4)}\right) \tag{1.7}$$

dengan menghitung untuk *forward* dan *backward* untuk model$h$, dapat dihitung dengan menjumlahkan keduanya untuk dihasilkan model layer ke-4 $h_t^{(4)}$pada rumus (1.8).

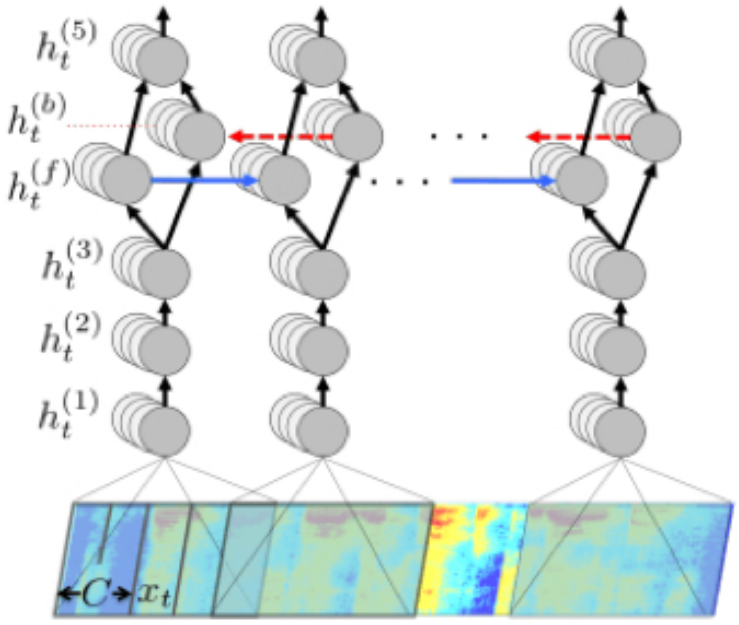
$$h_t^{(4)} = h_t^{(f)} + h_t^{(b)} \tag{1.8}$$

sehingga untuk perhitungan di layer ke-5 $h_t^{(5)}$ pada rumus (1.9)

$$h_t^{(5)} = g\left(W^{(5)} h_t^{(4)} + b^{(5)}\right) \tag{1.9}$$

Pada layer ke-5, dilakukan softmax function dimana dihitung $h_{t,k}^{(6)}$ dengan rumus (1.10).

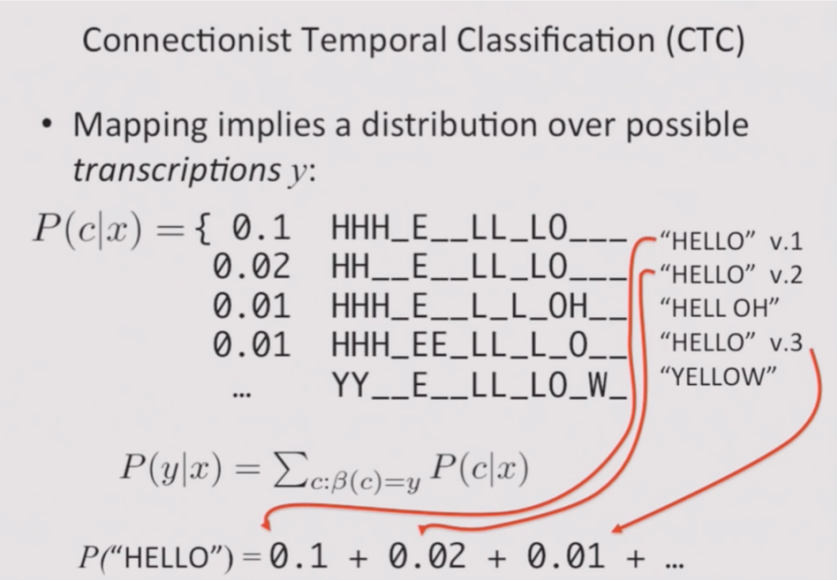
$$h_{t,k}^{(6)} = \hat{y}_{t,k} \equiv P(c_t = k|x) = \frac{\exp\left(W_k^{(6)} h_t^{(5)} + b_k^6\right)}{\sum_j \exp\left(W_j^{(6)} h_t^{(5)} + b_j^6\right)} \quad (1.10)$$



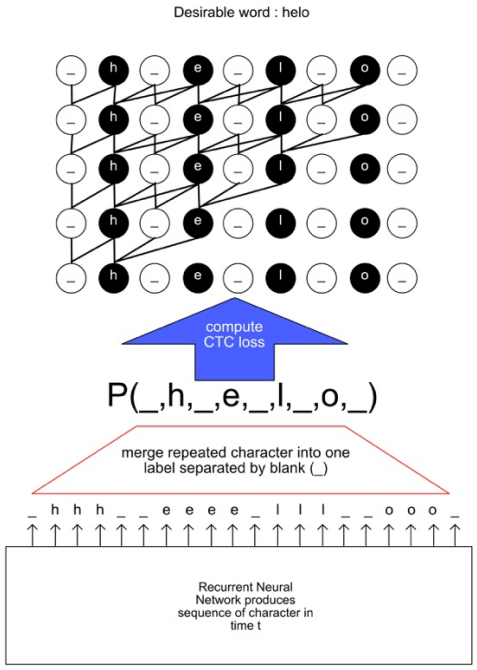
Gambar 1.5. Arsitektur mulai dari fitur spectrogram hingga mendapatkan model h yang diajukan oleh Baidu Research.

3. Connectionist Temporal Classification

   Klasifikasi dengan menghitung distribusi semua dari kemungkinan urutan huruf yang dihasilkan oleh RNN. Klasifikasi yang lebih dikenal dengan CTC ini sangat baik dalam mengatasi variasi kecepatan suara yang diucapkan. Contoh pada distribusi kata "*HELLO*" dari setiap training data berbeda - beda. Sehingga dari probabilitas yang ada akan di jumlahkan untuk mendapatkan transkrip yang paling mendekati benar (lihat Gambar 1.6 dan Gambar 1.7).



Gambar 1.6. Contoh presentasi oleh Baidu mengenai penjelasan pengklasifikasian urutan karakter menjadi kata dengan *Connectionist Temporal Classification*.



Gambar 1.7. Penghitungan CTC *Loss* dengan menggunakan *Forward Backward Algorithm* dimana memapingkan semua kemungkinan probabilitas n-gram karakter helo.

1.6. Metodologi Penelitian

Hipotesa

Dengan menerapkan metode RNN dan CTC, dimana merupakan salah satu cabang dari *Machine Learning* sehingga dalam penelitian ini dapat menghasilkan model yang dapat belajar mengenali pola dataset suara Bahasa Indonesia.

Parameter Uji

Parameter uji yang dipakai adalah *Word Error Rate* (WER) dengan rumus (1.11).

$$WER = \frac{S + D + I}{S + D + C} \tag{1.11}$$

WER = hasil perhitungan *error* kata

S = banyak kata yang diganti dengan tidak tepat

D = banyak kata yang dihapus / tidak dihasilkan oleh sistem

I = banyak kata yang ditambah dengan tidak tepat

C = banyak kata yang tepat

Parameter uji kedua adalah kecepatan dalam sistem dalam mengenali kata. Sistem akan mencetak timestamp pada saat mulai pengenalan hingga selesai. Sehingga didapatkan rumus (1.12).

$$v(W_n) = \frac{t}{n} \tag{1.12}$$

$v(W_n)$ = kecepatan dalam n kata

n = banyak kata

t = waktu dalam detik

Metode Pengujian

Sistem akan ditraining dengan dataset melalui system operasi Linux. Dataset yang sudah ditraining menghasilkan model sistem. Model sistem akan diuji untuk *voice recognizer system* melalui smartphone. Penguji sistem adalah 3 mahasiswa dengan aksen berbeda, bahasa Jawa, Batak, Papua. Ketiga mahasiswa akan membacakan sebuah paragraf yang ditentukan untuk dikenali oleh sistem.

1.7. Relevansi

Penelitian ini akan menghasilkan sebuah sistem voice recognizer baru untuk dapat diaplikasikan menjadi sebuah API / *Framework* untuk pengguna Android. Sehingga aplikasi yang bisa diciptakan adalah *Be My Ear* yaitu sebuah aplikasi yang membantu Tunarungu untuk berkomunikasi dengan orang awas.

1.8. Sistematika Penulisan

Penulisan laporan Skripsi ini dibagi menjadi beberapa bab, yaitu:

BAB I : PENDAHULUAN

Bab I berisikan judul, latar belakang, perumusan masalah, ruang lingkup, tujuan skripsi, metodologi penelitian, dan sistematika penulisan yang akan digunakan

BAB II : LANDASAN TEORI

Bab II berisikan teori-teori serta metode-metode yang digunakan dalam pembuatan skripsi

BAB III : ANALISIS DAN DESAIN SISTEM

Bab III berisikan analisis dan desain sistem yang dibuat

BAB IV : IMPLEMENTASI SISTEM

Bab IV berisikan tentang implementasi sistem berdasarkan desain sistem seperti pada Bab III

BAB V : PENGUJIAN SISTEM

Bab V berisikan pengujian sistem yang telah dibuat pada Bab IV

BAB VI : KESIMPULAN DAN SARAN

Bab VI berisikan kesimpulan yang dapat diambil terhadap hasil yang dicapai, dan saran – saran yang berguna bagi pengembangan selanjutnya.