Perencanaan dan Pembuatan Aplikasi Android Pengkonversian Suara Menjadi Teks dalam Bahasa Indonesia dengan Machine Learning untuk Membantu Tunarungu

David Wibisono, Rolly Intan, Endang Setyati
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658
E-mail: davidwibisono95@gmail.com, rintan@petra.ac.id, endang@stss.edu

ABSTRAK

Sistem Pengenalan Pengucapan sejauh ini telah mencapai WER *(Word Error Rate)* hingga 11.85% dalam *dataset* Bahasa Inggris. Dengan dukungan data yang sangat besar *machine learning* menjadi popular karena dengan adanya data yang sangat besar ini dapat membantu *machine learning* untuk mengenali pengucapan dengan lebih baik.

Penelitian ini terinspirasi dari arsitektur *Deep Speech* oleh Baidu, dan mengimplementasikan arsitektur tersebut terhadap *dataset* Bahasa Indonesia. *Dataset* dalam penelitian ini dibuat bervariasi berguna untuk menguji performa *machine learning* dalam mengenali pencucapan. Variasi *dataset* itu berupa suara rekaman dengan kondisi bersih dan berisik, suara hasil sintesis dari Apple dan Bing.

Penelitian ini menunjukkan bahwa semakin besar variasi *dataset,* maka memberikan hasil WER yang relative lebih kecil sehingga dengan kata lain hasil model *machine learning* dapat mengenali kata-kata dengan tepat. Permasalahannya adalah semakin besar *dataset*, maka semakin besar pula ambiguitas model bahasanya (*language model*).

Kata Kunci: *Deep Speech, Machine Learning, Speech Recognition, Neural Network, CTC, Language Model.*

ABSTRACT

The Speech Recognition System has achieved WER (Word Error Rate) up to 11.85% in English Words. Big data in speech can helps machine learning to become popular because it can maintain a good generalization to boost machine learning in speech recognition.

This paper inspired by Baidu (Deep Speech), we will implement its architecture to achieve the same goal in Indonesian Words. For this research, we use many variations of datasets according to its source such as clean environment voice, noise environment voice, and speech synthesizer from Apple and Bing.

The main problem is many variations of datasets influence the results of WER according to its size. Bigger variations of datasets maintain good generalization for the machine learning, but also it has big ambiguity in language model.

***Keywords*:** *Deep Speech, Machine Learning, Speech Recognition, Neural Network, CTC, Language Model.*

1. LATAR BELAKANG

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*. [6]

Menurut Tebelskis [10], 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.

Indonesia memiliki keanekaragaman bahasa, dialek, dan aksen sehingga masalah variasi ini menjadi perhatian khusus. Menurut Badan Pusat Statistik [2], terdapat lebih dari 2.500 bahasa daerah di Indonesia dengan jumlah penduduk saat itu mencapai 237.641.326 jiwa. Dari jumlah penduduk tersebut, hanya 19,94% saja yang secara tepat menggunakan Bahasa Indonesia sebagai Bahasa sehari-hari, melainkan 79,45% masih menggunakan Bahasa Daerah. Keanekaragaman bahasa daerah menghasilkan dialek dan aksen yang berbeda-beda pada tiap daerah, sehingga diperlukan metode yang dapat mengenali suara pengucapan dalam berbagai bentuk dialek dan aksen yang ada.

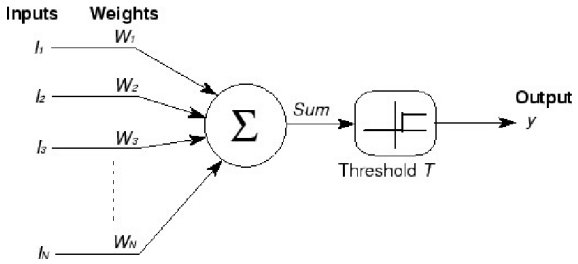
Untuk membuat Sistem *Speech Recognition*, metode HMM / GMM dalam masalah variasi yang disebutkan diatas dapat diminimalisasi tingkat errornya / dimaksimalkan tingkat akurasinya dengan menggunakan Deep Neural Networks. Menurut Baidu Research [1,4], sistem Recurrent Neural Network (RNN) dan Connectionist Temporal Classification (CTC) [3] menunjukkan peningkatan yang cukup signifikan yaitu mereduksi tingkat rata-rata error dalam Bahasa Inggris hingga 43%. Menurut Geofrey Hinton, et all [6] 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 [2,5,7,8] dan menunjukan hasil yang meningkatkan performa *speech recognition* terutama pada model akustik. Dalam penelitiannya Baidu memberi nama *Deep Speech* .

2. ARTIFICIAL NEURAL NETWORK

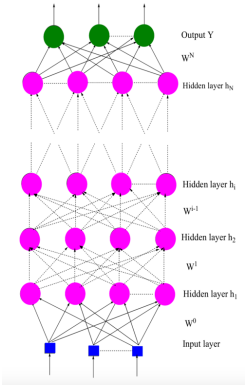
2.1 Multi Layer Perceptron

MLP *(Multi-Layer Perceptron)* atau yang biasa disebut *Feed Forward Neural Network* adalah model *perceptron* yang dikemukakan oleh Rosenblatt pada tahun 1950. Pada model ini berbeda dengan model pertama kali perkembangan *Neural Network* pertama kali dimana neuron terhubung pada satu layer dengan parameter *weight* dan *bias*. Lihat Gambar 1.



Gambar 1. *Neural Network* pertama kali

Perbedaan yang paling menonjol adalah pada *hidden layer* tidak hanya berjumlah satu melainkan banyak *neuron*. Jumlah banyak neuron yang digunakan disesuaikan dengan pola masalah yang akan diselesaikan. Semua layer yang terkoneksi dari *input layer*, *hidden layer*, dan *output layer* disebut juga arsitektur utama dimana akan di-*train,* sehingga dicapai parameter yang optimal sesuai dengan sebuah hasil sistem klasifikasi regresi. Lihat Gambar 2.



Gambar 2. MLP atau disebut juga dengan *Feed Forward Neural Network*

2.2 Bi-directional Neural Network

Bi-RNN (*Bi-directional Recurrent Neural Network*) adalah hasil modifikasi pada layer RNN dimana state tidak hanya terkoneksi dengan *state* berikutnya namun juga pada *state* sebelumnya. Rumus pada RNN dapat dinotasikan sebagai rumus (1)

$$h_t^{(j)} = f\left(W^{(j)T} h_t^{(j-1)} + W^{(f)T} h_{t-1}^{(j)} + b^{(j)}\right) \quad (1)$$

$h_t^{(j)}$ = hasil layer ke-j pada waktu t

T = panjang *frame-t*

Sedangkan rumus pada Bi-RNN perlu dihitung terlebih dahulu tidak hanya *forward* pada rumus (2) namun juga *backward* pada rumus (3)

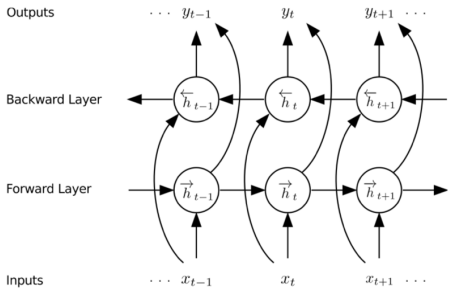
$$h_t^{(f)} = g\left(W^{(j)T} h_t^{(j)} + W^{(j)} h_{t-1}^{(j)} + b^{(j)}\right) \quad (2)$$

$$h_t^{(b)} = g\left(W^{(j)T} h_t^{(j)} + W^{(j)} h_{t-1}^{(j)} + b^{(j)}\right) \quad (3)$$

Dimana pada layer ke-j tidak hanya dihitung pada *forward state* saja, namun dengan *backward state*. Sehingga hasil dari $h_t^{(f)}$ dan $h_t^{(b)}$ dijumlahkan dengan rumus (4).

$$h_t^{(j)} = h_t^{(f)} + h_t^{(b)} \quad (4)$$

Pada rumus RNN (1) dengan Bi-RNN (4), perbedaan yang menonjol adalah adanya perhitungan *backward state,* keduanya sama-sama menghitung *forward state.* (lihat Gambar 3)



Gambar 3. Alur kerja Bi-RNN

2.3 Connectionist Temporal Classification

CTC *(Classification Temporal Classification)* adalah metode pengklasifikasian dengan label yang berurutan / sekuensial tanpa perlu mensegmentasi secara eksplisit. CTC dikemukakan oleh Alex Graves pada tahun 2006 [3]. CTC sangat populer digunakan pada *Speech Recognition,* tidak hanya performa bagus yang dicapai, namun keunggulan utama pada metode ini, tidak perlunya segmentasi pada label target teks.

CTC terlebih dahulu dimulai dengan perhitungan probabilitas dari tiap-tiap prediksi label yang ditetapkan. Misal saja label = { a, b, c }, maka tiap waktu akan memiliki 3 indeks probabilitas tersebut. Rumus untuk ditribusi daripada probabilitas terdapat pada rumus (5).

$$p(\pi|x) = \prod_{t=0}^{T} y_{\pi_t}^t, \forall \pi \in L'^T \quad (5)$$

$p(\pi|x)$ = distribusi probabilitas label x melalui *path* dinotasikan π

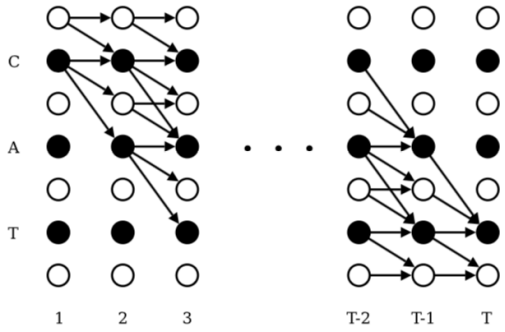
$y_{\pi_t}^t$ = hasil output dari RNN menggunakan operasi softmax

L'^T = elemen dari pada semua kemungkinan *path*

Penghitungan $y_{\pi_t}^t$ dihitung dengan operasi softmax pada rumus (6).

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

Pada Gambar 4 dapat dilihat hasil prediksi dengan kata "*cat*" dengan melihat semua path yang mungkin. Algoritma menghitungnya menggunakan Forward dan Backward Algorithm.

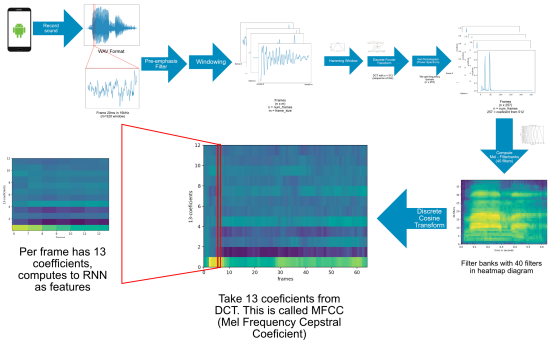


Gambar 4. Semua kemungkinan path untuk kata "cat"

3. ANALISIS SISTEM

3.1 Pengambilan Fitur MFCC (Mel-Frequency Cepstral Coefficient)

Pengambilan fitur / ekstraksi fitur yang digunakan adalah MFCC (*Mel-Frequency Cepstral Coefficients*). MFCC adalah koefisien dari koleksi yang dihasilkan oleh MFC (*Mel-Frequency Cepstrum*). Jumlah koefisien yang digunakan adalah 13. Ukuran *frame* yang digunakan setiap waktunya adalah 0.002 detik / 20 milidetik dengan *striding* / pergeseran sebanyak 0.001 detik / 10 milidetik. Berikut *flowchart* / diagram pengambilan fitur MFCC dari WAV *audio file* pada Gambar 5.



Gambar 5. *Flowchart* / diagram alur pengambilan fitur MFCC dari WAV *audio file*

Setelah pengambilan fitur MFCC dilakukan *past-future context*, artinya ambil fitur c-konteks sebelum dan sesudah pada kolom fitur waktu t.

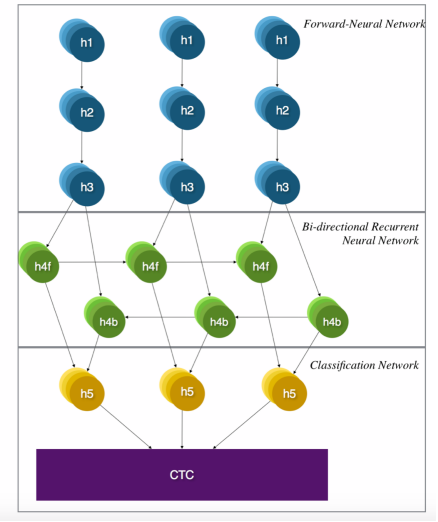
3.2 Model Neural Network

Dataset adalah fitur matriks yang dihasilkan oleh tahap preprocessing data. Konsep sederhana dari *neural network* adalah memasukan inputan *dataset* ke dalam 3 layer, yaitu *input*, *hidden*, dan *output*. Dimana semua layer diberikan *weight*, *bias* dan *activation function*. Pada model *neural network* digunakan *random weight* dan *bias* mengikuti gauss / normal *distribution* dengan *activation function* ReLU (*Rectifier Linear Unit*) pada rumus (7).

$$ReLU(x) = \max\{0, min\{x, 20\}\} \quad (7)$$

x = input fungsi ReLU

Alur jaringan *neural network* pada Gambar 6 dibagi 3 bagian besar, yaitu *Forward-Neural Network* (*hidden layer*), *Bi-directional Recurrent Neural Network* (*hidden layer*), *Classification Network* (*output layer*).

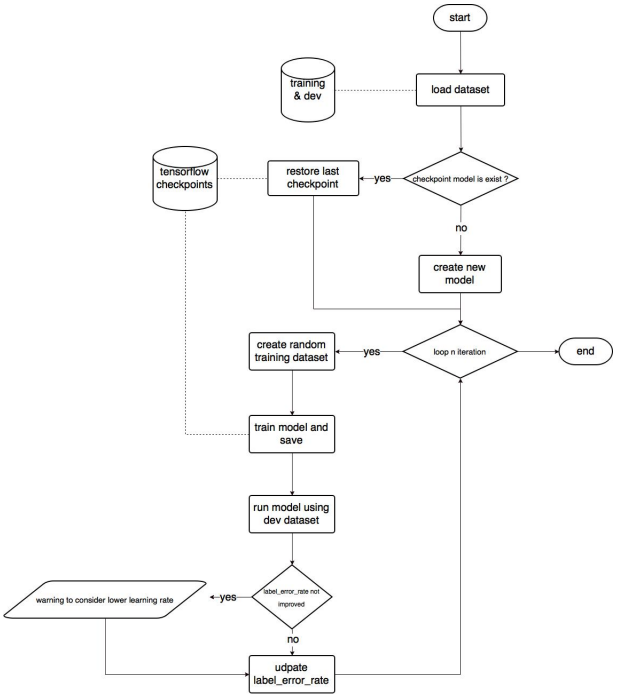


Gambar 6. Alur Jaringan Model *Neural Network*

3.3 Sistem *Training* dan *Testing*

Sistem dibagi menjadi dua bagian, sistem *training* dan sistem implementasi. Pada sistem *training*, dilakukan proses berulang dengan N iterasi untuk dilakukan *backpropagation* sehingga didapatkan perubahan *weight* dan *bias* yang optimum. *Backpropagation* yang digunakan adalah Adam Optimizer.

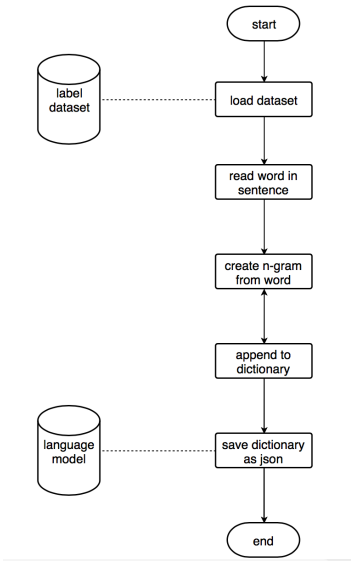
Untuk data *training* dibagi menjadi 2 bagian, *training* dan *dev dataset*. Dengan menggunakan *dev* untuk *dev dataset* untuk menghitung performa training model dengan tolak ukur *Edit Distance* / *Label Error Rate*. *Flowchart* / diagram alur untuk menjelaskan sistem *training* pada Gambar 7.



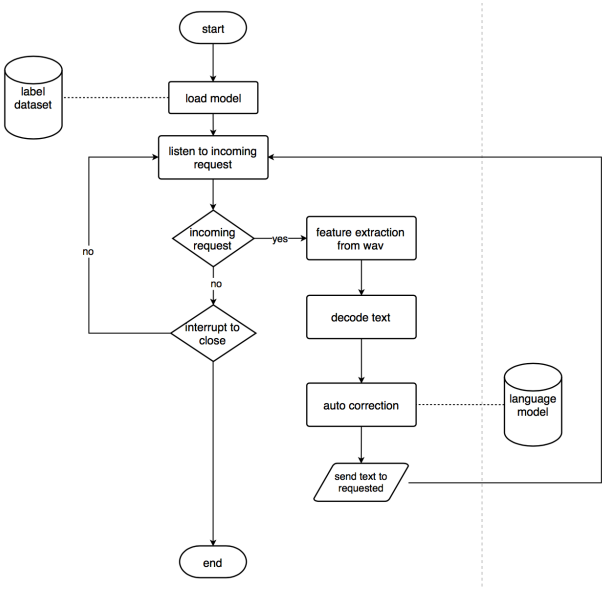
Gambar 7. *Flowchart* / diagram alur *training model*

Pada sistem implementasi, dilakukan dengan cara mengambil data suara yang direkam melalui *smartphone* Android untuk di kirim ke *server* yang sudah dihasilkan saat sistem *training*. *Server* akan mengolah data *raw audio file* yang dikirim oleh *smartphone* Android dan akan menghasilkan prediksi teks untuk ditampilkan di *smartphone* Android. Dalam memprediksi teks ditambahkan satu proses lagi yang dinamakan *language model*.

Dalam proses *language model* dilakukan pengecekan / pengkoreksian kata dalam kalimat yang dihasilkan oleh sistem *training*. Sebelum pengecekan terlebih dahulu akan dilakukan proses pembuatan model / *dictionary* untuk mempermudah proses pengecekan / pengkoreksian kata. *Flowchart* pembuatan model / *dictionary* adalah pada Gambar 8. Berikutnya setelah *language model* telah dibuat akan diimplementasikan pada *flowchart* Gambar 9.



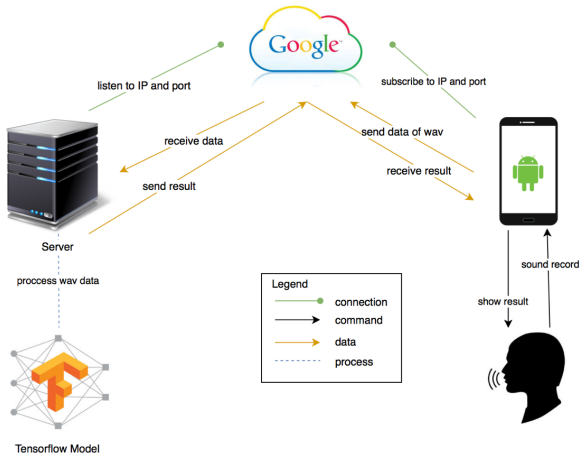
Gambar 8. *Flowchart* / alur diagram pembuatan model / *dictionary* untuk *language model*



Gambar 9. *Flowchart* / diagram alur sistem implementasi pada *server*

3.4 Sistem Aplikasi Android

Pada sistem aplikasi Android BeMyEar menggunakan *socket port* sebagai jalur komunikasi antara sistem *testing* dan aplikasi. Sistem testing disebut sebagai *server-side*, sedangkan sistem aplikasi disebut sebagai *client-side*. Pada *server-side* dibutuhkan API melalui *socket port* dengan mode *listen* terhadap IP dan *port* tertentu, sedangkan *client-side* akan *subscribe* terhadap IP dan *port* yang telah di-*assign* untuk *listen*. Keduanya berkomunikasi melalui jaringan internet (*cloud*) dengan memanfaatkan *Google Cloud Computing*. Berikut gambaran arsitektur kedua sistem dapat dilihat pada Gambar 10.

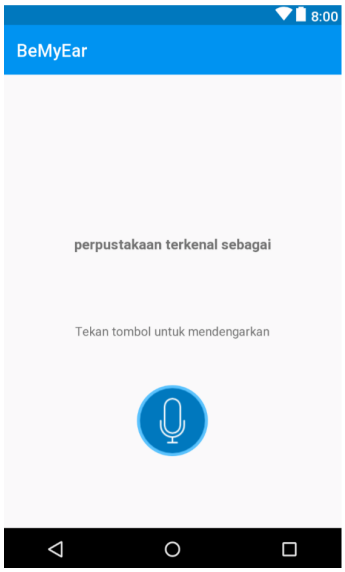


Gambar 10. Arsitektur Keseluruhan Aplikasi Android BeMyEar

4. DESAIN SISTEM

4.1 Desain Aplikasi Android

Pada penelitian ini dibuat desain *interface* untuk program Android bernama BeMyEar. Aplikasi ini akan mengimplementasikan sistem implementasi pada Gambar 10 untuk sistem *testing*. Tampilan aplikasi ini akan dibuat seperti pada Gambar 11.

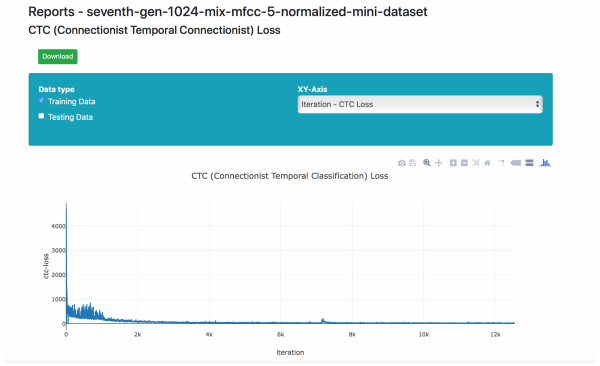


Gambar 11. Tampilan halaman awal jika sudah terkoneksi dengan *server*

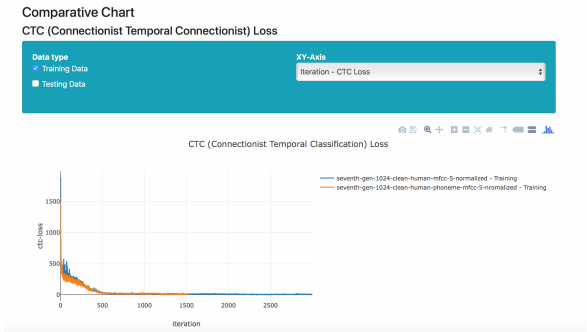
Pada Gambar 11, terdapat tombol  merupakan tombol yang digunakan untuk merekam suara dan mengirimkan kepada *server* melalui *socket* dengan *port* tertentu. Jika setelah mengatakan sebuah kalimat perlu menekan tombol *mic* untuk dilakukan pengenalan suara sehingga akan dihasilkan teks yang akan ditampilkan pada tengah layar perpustakaan terkenal sebagai .

4.2 Desain Web Report

Pada *Web Report* berisikan Menu Model yang telah di *train* oleh sistem. Menu berikutnya adalah *Dataset*, menu ini berisikan dataset yang digunakan pada *training*. Menu berikutnya adalah *Report* dan *Compare*, keduanya sama-sama menampilkan hasil grafik *training*. Namun pada *Compare* terdapat fitur untuk menggabungkan beberapa hasil model grafik. Contoh tampilan Menu *Report* dapat dilihat di Gambar 11. Contoh tampilan Menu *Compare* dapat dilihat Gambar 12.



Gambar 12. Tampilan isi dari Menu *Report*



Gambar 13. Tampilan isi dari Menu *Compare*

5. PENGUJIAN SISTEM

5.1 Variasi Dataset

Dataset yang digunakan pada pengujian adalah *dataset clean synth* dimana suara didapat dari *speech synthesizer* Apple dan Bing, berikutnya *dataset clean human* dimana suara didapat dari rekaman studio, dan terakhir *noise human* dimana suara direkam langsung melalui *smartphone*. Daftar dataset beserta total waktu dapat dilihat pada Tabel 1. Pembagian dataset *training* dan *testing* dengan perbandingan 9 : 1.

Tabel 1. Variasi *Dataset*.

Dataset	Total waktu
clean synth	44 menit 3 detik
clean human	14 menit 6 detik
noise human	58 menit 48 detik

Pada percobaan ini akan dibandingkan hasil dari dataset suara *clean synth*, *clean human*, *noise human, clean* dan *mix*. Masing – masing dataset dijalankan dengan 100 iterasi dengan *weight* dan *bias* yang sama. Dan dihasilkan rata-rata CTC-*loss (training dan testing dataset)* dan minimum CER (*testing dataset*) pada Tabel 2.

Tabel 2. Perbandingan CTC-*loss* dan CER dengan Variasi *Dataset*.

Dataset	CTC-loss		CER
	Training	Testing	
clean-synth	74.164	237.842	0.356
clean-human	42.082	335.573	0.494
clean	**11.751**	**76.774**	**0.120**
noise-human	19.114	226.643	0.345
mix	65.194	199.726	0.311

5.2 Variasi Label

Pada percobaan ini akan dilakukan per bagian dataset *clean synth*, *clean human*, *clean*, dan *noise human*. Fonem yang digunakan pada perubahan dari grafem dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan Grafem dan Fonem.

Teks	Grafem	Fonem
e pada menang	e	ə
ng	ng	ŋ
ny	ny	ñ
sy	sy	S
kh	kh	x

Masing-masing percobaan tiap grafem dan fonem dijalankan dengan 100 iterasi dengan *weight* dan *bias* yang sama. Dan dihasilkan rata – rata CTC-loss *(training dan testing dataset)* dan minimum CER (*testing dataset*) dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan CTC-*loss* dan CER dengan Variasi Label. (hasil terbaik dicetak tebal)

Dataset	CTC-*loss*		CER
	Training	***Testing***	
clean-synth-grapheme	74.164	237.842	0.356
clean-synth-phoneme	17.654	312.295	0.413
clean-human-grapheme	42.082	335.573	0.494
clean-human-phoneme	38.640	390.884	0.581
clean-grapheme	11.751	**76.774**	**0.120**
clean-phoneme	**9.652**	96.837	0.143
noise-human-grapheme	19.114	226.643	0.345
noise-human-phoneme	14.932	245.620	0.353

5.3 Variasi n-konteks

Pada percobaan ini dilakukan pada *dataset mix.* Fitur sebagai pembanding adalah MFCC 5 konteks dan 9 konteks. Grafik CTC-loss dan CER dari masing-masing fitur dapat dilihat pada Tabel 5.

Tabel 5. Perbandingan CTC-*loss* dan CER dengan Variasi n-konteks. (hasil terbaik dicetak tebal)

Dataset	CTC-*loss*		CER
	Training	***Testing***	
mix-mfcc-5	65.194	199.726	0.311
mix-mfcc-9	**30.627**	**126.273**	**0.255**

5.4 Pengujian Aplikasi

Pada implementasi ini akan diambil *tensorflow_model* dengan CER terkecil untuk di jadikan sebagai model. *Tensorflow model* yang digunakan adalah *clean*, *noise human*, *mix* MFCC dengan 5 konteks dan 9 konteks.

Percobaan dilakukan dengan merekam dari 5 sample kalimat yang diambil berdasarkan urutan CER terendah pada *testing dataset*. Kalimat disuarakan ulang oleh 1 orang melalui *smartphone* Android yang terkoneksi dengan jaringan dan dikondisikan tidak ada suara penggangu *(noise background)*. Hasil *output* adalah kata yang dihasilkan *neural network (decode text)* dan hasil yang telah di koreksi oleh *language model (LM text)*. Sebagai parameter uji dilakukan penghitungan WER *(Word Error Rate)* dari setiap kalimat setelah dihasilkan oleh *language model*. Hasil uji dapat dilihat pada Tabel 6.

Tabel 6. Hasil Perbandingan Rata-rata WER. (hasil terbaik dicetak tebal)

Tensorflow model	WER
clean dengan MFCC 5 konteks	0.9778
noise human dengan MFCC 5 konteks	**0.4446**
mix dengan MFCC 5 konteks	1.09
mix dengan MFCC 9 konteks	0.9388

Berikut contoh *decode text* dan *LM text* yang berhasil pada *noise human* dengan MFCC 5 konteks dapat dilihat pada Tabel 7.

Tabel 7. Tabel Hasil Uji Sistem pada *Tensorflow Model Dataset Mix* dengan MFCC 5-konteks

Target	perpustakaan terkenal sebagai tempat
Decode text	parpustaka toena a suka i tema
LM text	perpustakan terkenal suka tema
WER	0.75
Target	terkenal sebagai perpustakaan tempat
Decode text	tkerkan mang sebat tempa e pustak ks an e
LM text	kerjaan emang hebat tempat pustaka aksi dan
WER	1.5
Target	komisi kesaksian dan pelayanan akan mengadakan persekutuan doa sektor
Decode text	komisi ke sasian dar belayae na a kan akahdaokan uskut ua oas e trhuauh
LM text	komisi ke saksikan dari pelayan nah kan diadakan takut dua pas terhadap
WER	1.2
Target	sebenarnya pantai suluban hanyalah nama lain untuk pantai uluwatu
Decode text	seben ayan panfa sulu ban nyalana mala a in uentuk pan ta unuatu bs
LM text	seberang akan pantai dulu ban nyawanya malam in untuk dan tak uluwatu
WER	1.0
Target	empat kereta pustaka kereta pustaka ini diresmikan pada tahun dua ribu sebelas
Decode text	tmpa e ta pustakara stakainidli s ikan pal ta u ua i pus bela
LM text	tempat tak pustaka mengklarifikasi ikan hal tak dua puas belas
WER	1.0

6. KESIMPULAN

Dari hasil perancangan dan pembuatan aplikasi, dapat diambil kesimpulan antara lain:

- Aplikasi yang dikembangkan menyediakan fitur tombol untuk mendengarkan dan harus menggunakan koneksi internet.
- *Web Report* yang dikembangkan menyediakan empat fitur, antara lain melihat Model yang ada pada *server*, melihat *Dataset* yang ada pada *server*, melihat *Report* yaitu hasil grafik CTC-*loss* dan CER setiap model , dan mengkomparasi tiap-tiap model dalam bentuk grafik gabungan sebagai hasil grafik perbandingan CTC-loss dan CER.
- Sistem *training* telah diuji dengan dibuktikannya hasil CTC-loss yang sangat rendah dari tiap model.
- *Server* untuk melakukan *testing* pada *client* hanya mampu menerima 1 *client* saja.
- Faktor yang mempengaruhi WER adalah pemilihan fitur dan jumlah konteks, pemilihan konfigurasi rekaman suara dan variasi dataset, jumlah kata yang diucapkan pada *training dataset,* pemilihan label grafem / fonem, pemenggalan imbuhan kata untuk labelnya.
- Kecenderungan *machine learning (model neural network)* adalah semakin besar variasi dataset semakin baik ia mengenali kata-kata namun semakin besar juga ambiguitas Bahasa pada *language model*-nya.
- Terkadang dataset *noise* dapat membantu meningkatkan ataupun menurunkan WER.
- WER yang dicapai dalam penelitian ini paling kecil adalah 44,46% dengan kata – kata yang terbatas sesuai dengan *dataset training*.

7. REFERENSI

[1] Amodei, D., Anubhai, R., Battenberg, E., and Case, C., et all. 2015. Deep Speech : End-to-End Speech Recognition in English and Mandarin. *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, 1512.02595. DOI= https://arxiv.org/abs/1512.02595.

[2] Dahl, G., Yu, D., Deng, L., and Acero, A. 2011. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*. 30-42.

[3] Graves, A., Fernandez, S., Gomez, F., Schmidhuber, J. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. *Proceedings of the 23rd International Con- ference on Machine Learning*. ACM, Pittsburgh, PA, 369-376. DOI= https://dl.acm.org/citation.cfm?doid=1143844.1143891.

[4] Hannun, A., Case, C., Casper, J., and Catanzaro, B., et all. 2014. Deep Speech: Scaling up end-to-end speech recognition. 1412.5567. DOI=http://arxiv.org/abs/1412.5567.

[5] Hannun, A. Y., Maas, A. L., Jurafsky, D., and Ng, A. Y. 2014. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. abs/1408.2873. DOI=http://arxiv.org/abs/1408.2873.

[6] Hinton, G., Li, D., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A.,Vanhoucke, V., Nguyen, P., Sainath, T., Kingsbury, B. 2012. Deep Neural Network for Acoustic Model Speech Recognition. *IEEE Signal Processing Magazine,* 1-3.

[7] Lee, H., Pham, P., Largman, Y., and Ng, A.Y. 2009. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems*, 1096–1104.

[8] Mohamed, A., Dahl, G., and Hinton, G. 2011. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 99.

[9] Na'im, A., & Syaputra, H. 2010. *Kewarganegaraan, Suku Bangsa, Agama, Dan Bahasa Sehari-hari Penduduk Indonesia.* Badan Pusat Statistik, Jakarta, Jawa Barat.

[10] Tebelskis, J. 1995. *Speech Recognition using Neural Networks.* Doctoral Thesis. CMU Order Number: CMU-CS-95-142., Carnegie Mellon University.