LAPORAN AKHIR PENELITIAN

Ringkasan penelitian berisi: (i) latar belakang penelitian, (ii) tujuan penelitian, (iii) tahapan metode penelitian, (iv) luaran yang ditargetkan, (v) uraian TKT penelitian yang ditargetkan serta (vi) hasil penelitian yang diperoleh sesuai dengan tahun pelaksanaan penelitian.

RINGKASAN

Asisten virtual merupakan perangkat lunak dengan kecerdasan buatan yang dapat membantu manusia melakukan tugas tertentu. Beberapa contoh dari asisten virtual yang populer yaitu Siri dari Apple, Alexa dari Amazon, Cortana dari Microsoft, dan Google Assistant dari Google [1]. Asisten virtual membutuhkan sensor dan model untuk pengucapan agar dapat berinteraksi dengan manusia. Model pengucapan yang akan dikembangkan dalam penelitian ini yaitu model sintesis suara dan model konversi suara. Model sintesis suara saat ini sudah banyak dikembangkan seperti Wavenet dan Tacotron dari Google, Deep Voice dari Baidu dan lainnya. Dalam model sintesis suara dan konversi suara, permasalahan utamanya yaitu bagaimana menghasilkan pengucapan yang natural [2], [3]. Selain itu, metode sintesis suara hanya dapat mengeluarkan suara pengucapan dari pembicara yang ada dalam dataset pelatihan. Hal ini membuat model tersebut tidak variatif apalagi pengguna mempunyai preferensinya masing-masing. Oleh karena itu, dalam penelitian ini juga dikembangkan model konversi suara untuk mengubah suara pembicara tersebut ke suara pembicara yang lainnya. Model konversi suara tidak hanya melakukan transformasi suara tetapi juga harus bisa menirukan suara target yang sesuai pengucapannya tidak hanya dari segi nada tetapi juga aksennya [4].

Penelitian ini bertujuan untuk mengembangkan model pengucapan bahasa Indonesia bagi asisten virtual yang natural mendekati pengucapan manusia dan dapat menerima masukan berupa teks yang disintesis menjadi suara kemudian mengkonversi suara tersebut ke beberapa tipe suara pembicara yang lainnya.

Model sintesis suara menggunakan baseline dari model Tacotron [5]. Tacotron merupakan model sintesis suara end-to-end yang dilatih menggunakan input dataset pasangan teks dan pengucapan. Dalam penelitian ini, model akan dilatih menggunakan dataset pengucapan bahasa Indonesia. Model Tacotron dibangun berdasarkan model sequence-to-sequence (seq2seq) yang menggunakan attention sehingga tidak memerlukan penyelarasan tingkat fonem atau fitur linguistik seperti pada model WaveNet. Model ini menggunakan rekonstruksi Griffin-Lim untuk mensitesis pengucapannya. Hasil keluaran modelnya berupa spektrogram yang dapat digunakan untuk membuat sinyal suara dari pembicara tersebut berdasarkan masukan teks yang diberikan. Model konversi suara menggunakan baseline MelGAN (Mel-Generative Adversarial Network) yang dilatih menggunakan inputan dari model Tacotron untuk dikonversi ke suara pembicara dari dataset yang diberikan [6].

Luaran yang dicapai pada penelitian ini yaitu satu artikel ilmiah untuk jurnal nasional terindeks SINTA sebagai luaran wajib. TKT penelitian ini pada skala 3. Penelitian berlangsung selama satu tahun dan ditargetkan mendapatkan hasil model sintesis suara dan konversi suara Bahasa Indonesia.

Kata kunci maksimal 5 kata kunci. Gunakan tanda baca titik koma (;) sebagai pemisah dan ditulis sesuai urutan abjad.

sintesis suara; text-to-speech; model pengucapan; asisten virtual.

Hasil pelaksanaan penelitian berisi: (i) kemajuan pelaksanaan penelitian yang telah dicapai sesuai tahun pelaksanaan penelitian, (ii) data yang diperoleh, (iii) hasil analisis data yang telah dilakukan, (iv) pembahasan hasil penelitian, serta (v) luaran yang telah didapatkan. Seluruh hasil atau capaian yang dilaporkan harus berkaitan dengan tahapan pelaksanaan penelitian sebagaimana direncanakan pada proposal. **Penyajian data** dan **hasil penelitian** dapat berupa gambar, tabel, grafik, dan sejenisnya, serta **pembahasan hasil penelitian** didukung dengan sumber pustaka primer yang relevan dan terkini.

HASIL PELAKSANAAN PENELITIAN

Model pengucapan terdiri dari model sintesis suara dan model konversi suara. Model sintesis suara menggunakan baseline dari model Tacotron [3], [5]. Tacotron merupakan model sintesis suara *end-to-end* yang dilatih menggunakan input dataset pasangan teks dan pengucapan. Dalam penelitian ini, model akan dilatih menggunakan dataset pengucapan bahasa Indonesia. Model Tacotron dibangun berdasarkan model sequence-to-sequence (seq2seq) yang menggunakan attention [7] sehingga tidak memerlukan penyelarasan tingkat fonem atau fitur linguistik seperti pada model WaveNet [2]. Model ini menggunakan rekonstruksi Griffin-Lim [8] untuk mensitesis pengucapannya. Hasil keluaran modelnya berupa spektrogram yang dapat digunakan untuk membuat sinyal suara dari pembicara tersebut berdasarkan masukan teks yang diberikan secara otomatis.

Model konversi suara menggunakan baseline model Generative Adversarial Network (GAN) [6], [9], [10] yang dilatih menggunakan inputan dari model Tacotron untuk dikonversi ke suara pembicara dari dataset yang diberikan. GAN merupakan model generatif yang biasanya digunakan untuk pembuatan citra dan transfer style dari satu citra/suara ke citra/suara yang lain. Model konversi suara juga menggunakan rekonstruksi Griffin-Lim untuk menghasilkan sinyal suara dari spektrogramnya.

Pada penelitian ini dikembangkan model pengucapan Bahasa Indonesia bagi asisten virtual yang meliputi model sintesis suara dan model konversi suara. Model sintesis suara sudah banyak dikembangkan oleh peneliti terutama dari Google [2], [3], [5] dan Baidu [11]–[13] dan mendapatkan hasil yang baik dari segi kealamian pengucapan untuk sintesis suara Bahasa Inggris dan Mandarin. Model konversi suara juga telah banyak dikembangkan dan mendapatkan hasil yang baik terutama menggunakan tipe-tipe model GAN [4], [9], [10] yang memang unggul untuk menangani kasus style transfer.

Data

Data yang digunakan dalam penelitian ini yaitu data suara yang diambil dari dataset publik dan data suara yang di ambil dari internet. Dataset publik yang digunakan yaitu LJSpeech [14] sebagai benchmark dataset. Dataset suara Bahasa Indonesia untuk pelatihan awal model sintesis suara dicuplik dari suara artis yang diambilkan dari film pendek, vlog, dan talkshow yang diunduh dari internet. Dataset ini digunakan untuk mengembangkan arsitektur dan konfigurasi model sintesis suara yang disesuaikan dengan data suara Bahasa Indonesia. Setelah model teruji dengan baik, dataset suara Bahasa Indonesia dengan durasi waktu yang panjang dari beberapa pengisi suara akan diambil kemudian. Tabel 1 menunjukkan struktur dari dataset yang digunakan.

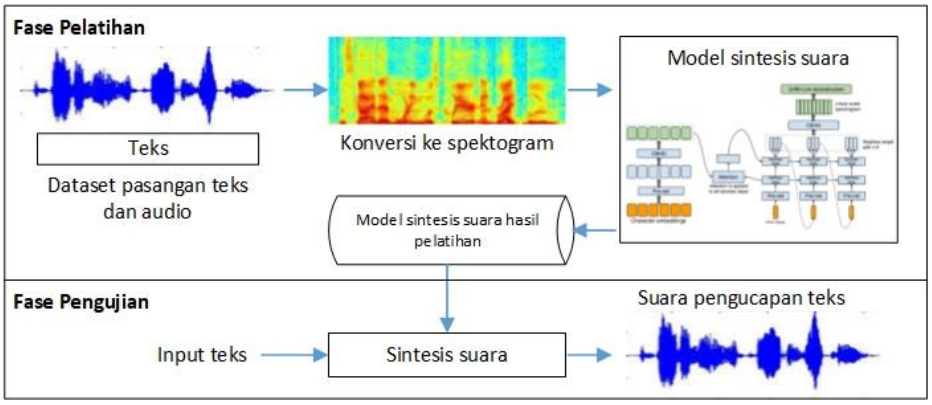
Tabel 1. Informasi dataset

No	Dataset	Jumlah Data	Format Data	Lama Waktu
1	LJSpeech (Bahasa Inggris)	13.100	WAV, each audio file is a single-channel 16-bit PCM WAV with a sample rate of 22050 Hz	Setiap data sekitar 1-10 detik dengan total durasi hampir 24 jam

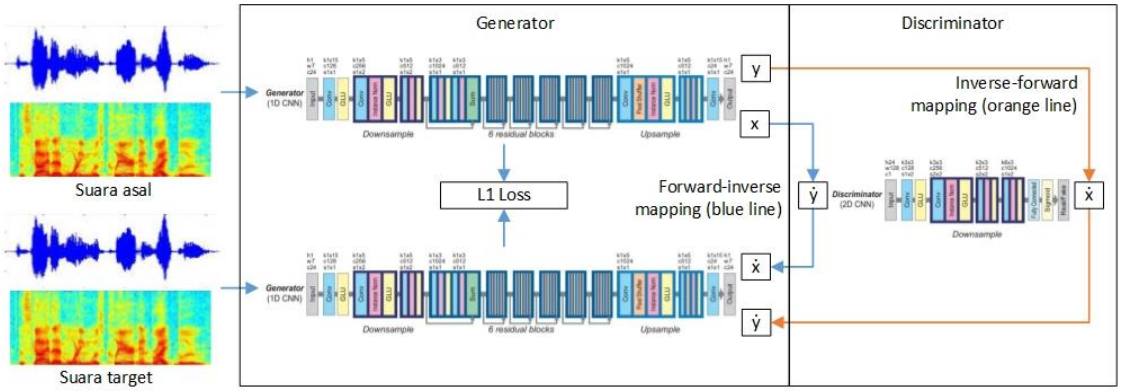
No	Dataset	Jumlah Data	Format Data	Lama Waktu
2	Suara Artis (Bahasa Indonesia)	203	WAV, each audio file is a single-channel 16-bit PCM WAV with a sample rate of 22050 Hz	Setiap data sekitar 1-10 detik dengan total durasi hampir 60 menit

Hasil Analisis

Dari hasil analisis kebutuhan perangkat untuk mengolah data dan membangun model sinstesis suara, model ini akan dijalankan pada Google Colab dengan Notebook berbahasa Python dan komputer dengan spesifikasi prosesor Intel Core i7, RAM 16 GB, dan kartu grafis Nvidia GeForce GTX1070. Untuk menerapkan arsitektur model digunakan tambahan library tensorflow-tts, soundfile, tensorflow, scikit-learn, panda, matplotlib dan tensorboard. Pengembangan model pengucapan asisten virtual ditunjukkan pada Gambar 1. Prosedur dibagi menjadi dua bagian yaitu bagian model sintesis suara dan bagian model konversi suara untuk pengucapan Bahasa Indonesia. Pada bagian model sintesis suara, masukan untuk fase pelatihan adalah pasangan data teks dan audio pengucapan ahasa Indonesia dari beberapa pembicara. Data kemudian masuk ke bagian encoder, decoder dengan atensi dan menghasilkan spektogram yang bisa direkonstruksi menjadi sinyal suara menggunakan metode Griffin-Lim. Pada fase test, data teks yang diinputkan akan diproses oleh model yang sudah dilatih tersebut menjadi sinyal suara pengucapan pembicara tersebut. Pada bagian model konversi suara, masukan untuk fase pelatihan yaitu pasangan spektogram dari suara asal dan suara target. Data kemudian masuk ke bagian generator dan discriminator untuk dipelajari style dari suara target. Pada fase test, data suara asal akan dikonversi ke suara target menggunakan model yang sudah dilatih tersebut.



(b) Model sintesis suara



(b) Model konversi suara

Gambar 1. Prosedur dari (a) model sintesis suara dan (b) model konversi suara.

Pembahasan Hasil

Dari hasil analisis, model sintesis suara kemudian diterapkan menggunakan bahasa pemrograman Python dan library tensorflow untuk diukur kinerjanya. Penelitian yang dilakukan hanya sampai pada model sintesis suara dan belum mencapai tahapan konversi suara. Hal ini dikarenakan hasil dari sintesis suara yang masih rendah sehingga belum memungkinkan untuk dilanjutkan ke tahapan selanjutnya.

Hasil Model Sintesis Suara

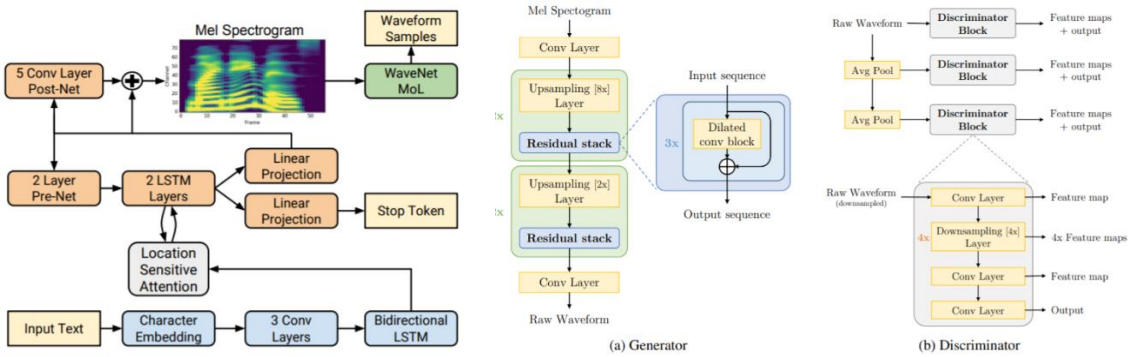
Dataset pada Tabel 1 digunakan sebagai input untuk pelatihan bagi baseline model Tacotron 2 [5] dan MelGAN [6] yang diambilkan dari library Tensorflow-TTS [15]. Prosesnya terdiri dari 3 tahapan yaitu 1) pra-proses data suara menjadi fitur mel spektogram, 2) pelatihan menggunakan model Tacotron 2 dan pelatihan menggunakan model MelGAN dan 3) hasil pelatihan digunakan untuk melakukan prediksi output spektogram oleh model Tacotron 2 yang kemudian disintesis menjadi suara oleh model model MelGAN. Detil dari tahapan tersebut dijelaskan sebagai berikut.

1. Pra-proses dataset

Pra-proes dataset akan mengekstrak beberapa fitur dari data suara yaitu ID data, mel spektogram, dan mel spektogram yang ternormalisasi ke rentang [-1,1]. Fitur tersebut dibagi menjadi data latih dan data validasi kemudian dihitung rerata dan standar deviasi untuk menstandarkan mel spektogramnya.

2. Pelatihan

Pelatihan terbagi menjadi dua tahapan yaitu pelatihan menggunakan model Tacotron 2 dan pelatihan menggunakan model MelGAN pada dataset nomor 2 yaitu suara artis (lihat Tabel 1). Pelatihan model Tacotron 2 menghasilkan model untuk menerima input teks yang diprediksi menjadi mel spektogram dari suara pembicara yang digunakan di data latih. Arsitektur model (a) Tacotron 2 dan (b) MelGAN yang digunakan pada penelitian ini ditunjukkan pada Gambar 2 dan detilnya ditunjukkan pada Gambar 3.



(a) Tacotron 2 [5] (b) MelGAN [6]

Gambar 2. Arsitektur Model (a) Tacotron 2 dan (b) MelGAN

Pelatihan model MelGAN menghasilkan model untuk menerima input mel spektogram menjadi waveform suara pembicara yang digunakan di data latih. Kami melakukan pelatihan dari awal dan tidak menggunakan pre-trained bobot dari model yang dilatih pada dataset LJSpeech karena struktur pengucapan Bahasa Indonesia dan Bahasa Inggris berbeda. Pengaturan hyperparameter menggunakan default dari library Tensorflow-TTS pada dataset LJSpeech seperti yang ditunjukkan pada Tabel 2. Pelatihan untuk Tacotron 2 menggunakan 100 epoch sedangkan pelatihan untuk MelGAN menggunakan 60 epoch karena konsumsi RAM yang cukup besar.

```
Model: "tacotron2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 encoder (TFTacotronEncoder)  multiple                  8218624

 decoder_cell (TFTacotronDec  multiple                  18246402
 oderCell)

 post_net (TFTacotronPostnet  multiple                  5460480
 )

 residual_projection (Dense)  multiple                  41040

=================================================================
Total params: 31,966,546
Trainable params: 31,956,306
Non-trainable params: 10,240
```

```
Model: "melgan_generator"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (None, None, 1)           4260257

=================================================================
Total params: 4,260,257
Trainable params: 4,260,257
Non-trainable params: 0
_________________________________________________________________
Model: "melgan_discriminator"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 melgan_discriminator_scale_  multiple                  5637953
 ._0 (TFMelGANDiscriminator)

 melgan_discriminator_scale_  multiple                  5637953
 ._1 (TFMelGANDiscriminator)

 melgan_discriminator_scale_  multiple                  5637953
 ._2 (TFMelGANDiscriminator)

 average_pooling1d_2 (Averag  multiple                  0
 ePooling1D)

=================================================================
Total params: 16,913,859
Trainable params: 16,913,859
Non-trainable params: 0
```

Gambar 3. Detil parameter setiap layer pada Tacotron 2 dan MelGAN

Tabel 2. Pengaturan hyperparameter model

Tacotron 2	MelGAN

Hop size: 256
Dataset
batch_size: 32
remove_short_samples: true
allow_cache: true
mel_length_threshold: 32
is_shuffle: true
use_fixed_shapes: true
 	
Network hyperparameter:
embedding_hidden_size: 512
initializer_range: 0.02
embedding_dropout_prob: 0.1
n_speakers: 1
n_conv_encoder: 5
encoder_conv_filters: 512
encoder_conv_kernel_sizes: 5
encoder_conv_activation: 'relu'
encoder_conv_dropout_rate: 0.5
encoder_lstm_units: 256
n_prenet_layers: 2
prenet_units: 256
prenet_activation: 'relu'
prenet_dropout_rate: 0.5
n_lstm_decoder: 1
reduction_factor: 1
decoder_lstm_units: 1024
attention_dim: 128
attention_filters: 32
attention_kernel: 31
n_mels: 80
n_conv_postnet: 5
postnet_conv_filters: 512
postnet_conv_kernel_sizes: 5
postnet_dropout_rate: 0.1
attention_type: "lsa"
	Sampling rate: 22050
	Hop size: 256
	Dataset:
	batch_size: 16
	batch_max_steps: 8192
	batch_max_steps_valid: 81920
	remove_short_samples: true
	allow_cache: true
	is_shuffle: true
	
	Network hyperparameter – generator:
	out_channels: 1
	kernel_size: 7
	filters: 512
	upsample_scales: [8, 8, 2, 2]
	stack_kernel_size: 3
	stacks: 3
	is_weight_norm: false
	
	Network hyperparameter – discriminator:
	out_channels: 1
	scales: 3
	downsample_pooling: "AveragePooling1D"
	downsample_pooling_params:
	  pool_size: 4
	  strides: 2
	kernel_sizes: [5, 3]
	filters: 16
	max_downsample_filters: 1024
	downsample_scales: [4, 4, 4, 4]
	nonlinear_activation: "LeakyReLU"
	nonlinear_activation_params:
	  alpha: 0.2
	is_weight_norm: false
	
	Adversarial loss:
	lambda_feat_match: 10.0

Optimizer: initial_learning_rate: 0.001 end_learning_rate: 0.00001 decay_steps: 75 warmup_proportion: 0.02 weight_decay: 0.001 gradient_accumulation_steps: 1 Pelatihan: train_max_steps: 100 save_interval_steps: 50 eval_interval_steps: 50 log_interval_steps: 20 start_schedule_teacher_forcing: 200001 start_ratio_value: 0.5 schedule_decay_steps: 50000 end_ratio_value: 0.0	Optimizer – generator dan discriminator: lr: 0.0001 beta_1: 0.5 beta_2: 0.9 gradient_accumulation_steps: 1 Pelatihan: train_max_steps: 60 save_interval_steps: 10 eval_interval_steps: 2 log_interval_steps: 10 discriminator_train_start_steps: 0

Hasil kinerja pelatihan yaitu loss ditunjukkan pada Tabel 3. Loss untuk pelatihan model Tacotron 2 dan MelGAN masih tinggi tapi terjadi penurunan disetiap epochnya. Dari benchmark dataset LJSpeech diketahui bahwa kinerja model akan baik setelah lebih 65.000-200.000 epoch [15]. Sampai saat ini kami masih melanjutkan pelatihan agar mencapai target tersebut dikarenakan terbatasnya resource Google Colab yang disediakan.

Tabel 3. Hasil kinerja pelatihan model.

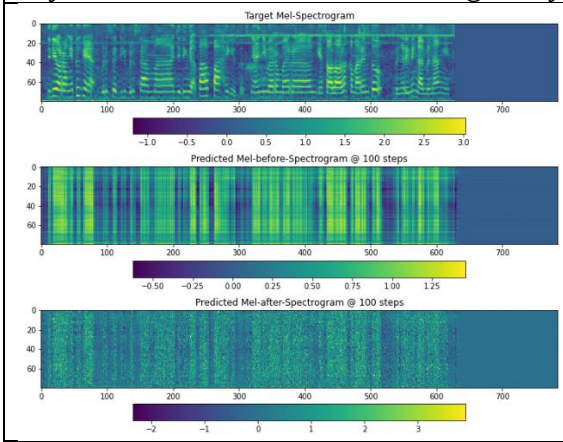
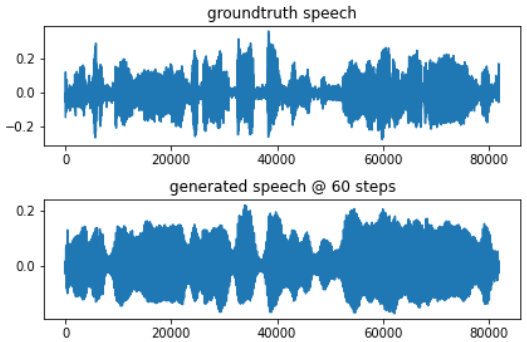
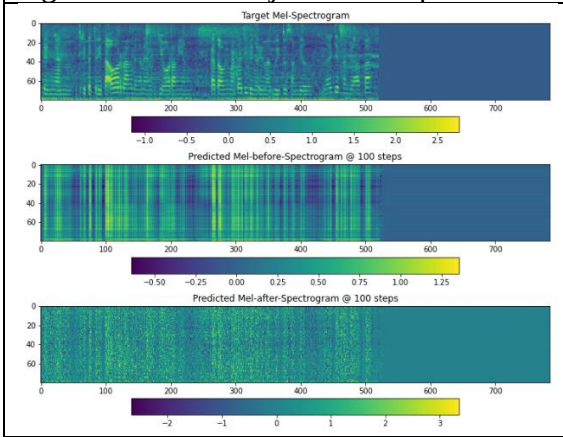
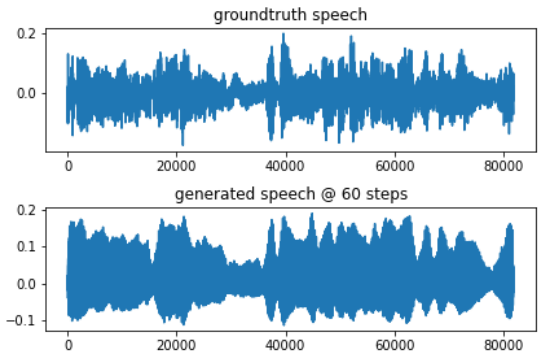
Tacotron 2		**MelGAN**	
Epoch-20 Training	Mel loss before: 0.4049 Mel loss after: 0.7906 Guided attention loss: 0.0146	Epoch-10 Training	fm_loss: 0.0045 Gen loss: 0.9779 Real loss: 0.9333 Fake loss: 0.0022 Dis loss: 0.9355 Mel spectrogram loss: 2.2060
Epoch-40 Training	Mel loss before: 0.2594 Mel loss after: 0.4246 Guided attention loss: 0.0122	Epoch-20 Training	fm_loss: 0.0041 Gen loss: 0.5738 Real loss: 0.5312 Fake loss: 0.0950 Dis loss: 0.6262 Mel spectrogram loss: 1.9270
Epoch-50 Evaluasi	Stop token loss: 0.0137 Mel loss before: 0.2176 Mel loss after: 0.3779 Guided attention loss: 0.0095	Epoch-30 Evaluasi	Adversarial loss: 0.2308 fm_loss: 0.0032 Gen loss: 0.2308 Real loss: 0.1897 Fake loss: 0.3152 Dis loss: 0.5049 Mel spectrogram loss: 2.6192
Epoch-60 Training	Mel loss before: 0.2211 Mel loss after: 0.3496 Guided attention loss: 0.0110	Epoch-40 Training	fm_loss: 0.0051 Gen loss: 0.3310 Real loss: 0.2724 Fake loss: 0.2402 Dis loss: 0.5127 Mel spectrogram loss: 1.8524
Epoch-80 Training	Mel loss before: 0.2058 Mel loss after: 0.3233 Guided attention loss: 0.0106	Epoch-50 Training	fm_loss: 0.0052 Gen loss: 0.3303 Real loss: 0.2754

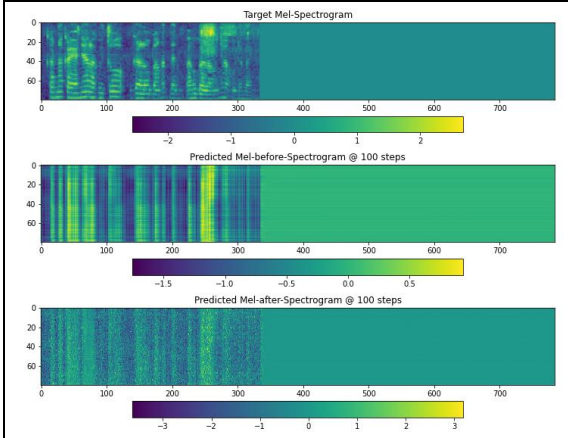
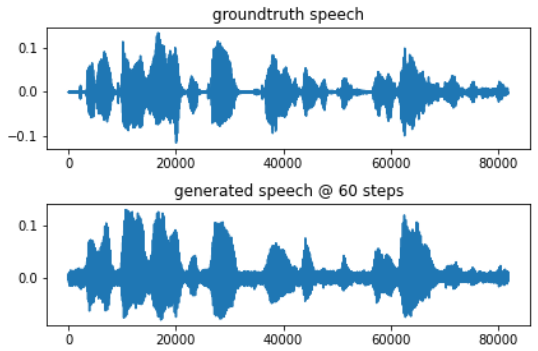
			Fake loss: 0.2392 Dis loss: 0.5146 Mel spectrogram loss: 1.9318
Epoch-100: Evaluasi	Stop token loss: 0.0130 Mel loss before: 0.2238 Mel loss after: 0.3259 Guided attention loss: 0.0087	Epoch-60 Evaluasi	Adversarial loss: 0.2366 fm_loss: 0.0040 Gen loss: 0.2366 Real loss: 0.1896 Fake loss: 0.3199 Dis loss: 0.5095 Mel spectrogram loss: 2.6832

3. Prediksi

Hasil beberapa spektogram dari model Tacotron 2 pada epoch ke-100 dan hasil sinyal suara dari model MelGAN pada epoch ke-60 setelah pelatihan selesai ditunjukkan pada Tabel 4. Berdasarkan Tabel 4 dapat dilihat bahwa hasil dari pelatihan kurang baik karena lossnya masih tinggi. Untuk mengatasi hal itu diperlukan pelatihan lanjutan dengan epoch yang lebih besar yang membutuhkan waktu cukup lama dengan resource komputasi yang terbatas.

Tabel 4. Hasil prediksi pelatihan model.

Nama file: RS001-0016 Teks: Jadi orang itu kayak bersedih bersama jadi ga apa-apa gitu, nyanyi bareng-bareng kayak seolah-olah kemarin tuh dengerinnya ga sambil nangis.	
	
Tactoron 2 - Spektogram	MelGAN - Suara
Nama file: RS001-0019 Teks: Dengan cerita-cerita orang dengan energi orang yang kata om Iwan tadi dengerin lagu itu sambil belajar sambil apa	
	
Tactoron 2 - Spektogram	MelGAN - Suara

Nama file: RS005-0100 Teks: Karena kayaknya lagu itu galau banget dan lagu galaunya udah banyak	
	
Tactoron 2 - Spektogram	MelGAN - Suara

Hasil Model Konversi Suara

Hasil model konversi suara belum bisa ditunjukkan karena hasil model sintesis suara yang masih belum baik sehingga langkah selanjutnya yaitu konversi suara dari model sintesis suara ke suara pembicara yang lain tidak bisa dilakukan. Konversi suara akan dilakukan di penelitian selanjutnya setelah penelitian tentang sintesis suara menghasilkan hasil yang baik.

Pengujian

Pengujian kinerja metode pengucapan untuk asisten virtual menggunakan evaluasi subjektif yaitu menghitung *mean opinion score* (MOS) dengan skala 5 dari beberapa responden. Pengujian belum bisa dilakukan karena hasil pelatihan belum mencapai target yang diinginkan.

Capaian Luaran

Saat ini penelitian masih berada pada 80% penyelesaian dimana sudah didapatkan hasil spektogram dari model Tacotron dan hasil suara (waveform) dari model MelGAN. Program juga sudah dibuat dan dilatih modelnya tetapi belum mendapatkan hasil pelatihan yang maksimal. Selanjutnya akan diteruskan pelatihan hingga mencapai 65.000 epoch seperti pada benchmark LJSpeech dataset dan akan ditambah jumlah datanya. Setelah hasil yang didapatkan mendapatkan loss yang kecil dan di pengujian mendapatkan hasil yang baik maka naskah publikasi bisa dibuat.

Status luaran berisi **identitas** dan **status ketercapaian setiap luaran wajib** dan **luaran tambahan** (jika ada) yang dijanjikan. Jenis luaran dapat berupa publikasi, perolehan kekayaan intelektual, hasil pengujian atau luaran lainnya yang telah dijanjikan pada proposal. Uraian status luaran harus didukung dengan **bukti kemajuan** ketercapaian luaran sesuai dengan luaran yang dijanjikan. Lengkapi isian jenis luaran yang dijanjikan serta **lampirkan bukti dokumen** ketercapaian luaran wajib dan luaran tambahan.

STATUS LUARAN

Status luaran wajib: **draft**.

Luaran wajib yaitu satu artikel untuk publikasi di jurnal nasional terindeks Sinta yang berbahasa Inggris. Akan tetapi karena hasil peneltian sinstesis suara masih belum baik dan memerlukan waktu pelatihan model yang lebih panjang maka publikasi luaran penelitian ini ditunda sampai mendapatkan hasil yang baik agar layak untuk dipublikasikan.

Peran Mitra berupa **realisasi kerjasama** dan **kontribusi Mitra** baik *in-kind* maupun *in-cash* (untuk Penelitian Terapan dan Pengembangan). **Bukti pendukung** realisasi kerjasama dan realisasi kontribusi mitra **dilaporkan** sesuai dengan kondisi yang sebenarnya. **Lampirkan bukti dokumen** realisasi kerjasama dengan Mitra.

PERAN MITRA

Tidak ada mitra penelitian.

Kendala Pelaksanaan Penelitian berisi **kesulitan** atau **hambatan** yang dihadapi selama melakukan penelitian dan mencapai luaran yang dijanjikan, termasuk **penjelasan jika** pelaksanaan penelitian dan luaran penelitian **tidak sesuai** dengan yang direncanakan atau dijanjikan.

KENDALA PELAKSANAAN PENELITIAN

Kendala yang dihadapi yaitu keterbatasan akses ke peralatan di Laboratorium karena pandemi Covid-19 yaitu komputer untuk pembuatan dan pelatihan model. Pelatihan model membutuhkan resource yang sangat besar dengan komputasi yang sangat lama sehingga komputer perlu melakukan komputasi selama berhari-hari. Apabila menggunakan server dari Google Colab ketersediaan RAM dan GPU untuk pemrosesan paralel terbatas dan dibatasi juga waktu penggunaan hariannya. Hal ini membuat waktu penyelesaian penelitian menjadi mundur.

Rencana Tindak Lanjut Penelitian berisi uraian rencana tindaklanjut penelitian selanjutnya dengan melihat hasil penelitian yang telah diperoleh. Jika ada target yang belum diselesaikan pada akhir tahun pelaksanaan penelitian, pada bagian ini dapat dituliskan rencana penyelesaian target yang belum tercapai tersebut.

RENCANA TINDAK LANJUT PENELITIAN

Tahapan selanjutnya yaitu menyelesaikan pelatihan sampai 65.000 – 200.000 epoch, dan menguji kinerja dari model sintesis suara menggunakan MOS. Setelah didapatkan hasil pengujian yang baik, maka bisa melakukan finalisasi dari draft artikel untuk dipublikasikan di jurnal nasional.

Daftar Pustaka disusun dan ditulis **berdasarkan sistem nomor** sesuai dengan urutan pengutipan. **Hanya pustaka yang disitasi/diacu** pada laporan kemajuan saja yang dicantumkan dalam Daftar Pustaka.

DAFTAR PUSTAKA

[1] M. B. Hoy, “Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants,” *Med. Ref. Serv. Q.*, vol. 37, no. 1, pp. 81–88, Jan. 2018, doi: 10.1080/02763869.2018.1404391.

[2] A. van den Oord *et al.*, “WaveNet: A Generative Model for Raw Audio.” 2016.

[3] Y. Wang *et al.*, “Tacotron: Towards End-to-End Speech Synthesis.” 2017.

[4] Y. Gao, R. Singh, and B. Raj, “Voice Impersonation using Generative Adversarial Networks,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2506–2510, doi: 10.1109/ICASSP.2018.8462018.

[5] J. Shen *et al.*, “Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783, doi: 10.1109/ICASSP.2018.8461368.

[6] A. C. Kumar, Kundan and Kumar, Rithesh and de Boissiere, Thibault and Gestin, Lucas and Teoh, Wei Zhen and Sotelo, Jose and de Br\’{e}bisson, Alexandre and Bengio, Yoshua and Courville, “MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis,” in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

[7] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate.” 2016.

[8] D. Griffin and Jae Lim, “Signal estimation from modified short-time Fourier transform,” in *ICASSP ’83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Apr. 1983, vol. 8, pp. 804–807, doi: 10.1109/ICASSP.1983.1172092.

[9] T. Kaneko and H. Kameoka, “Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks.” 2017.

[10] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, “StarGAN-VC: non-parallel many-to-many Voice Conversion Using Star Generative Adversarial Networks,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 266–273, doi: 10.1109/SLT.2018.8639535.

[11] S. Ö. Arik *et al.*, “Deep Voice: Real-time Neural Text-to-Speech,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 195–204, [Online]. Available: http://proceedings.mlr.press/v70/arik17a.html.

[12] S. Arik *et al.*, “Deep Voice 2: Multi-Speaker Neural Text-to-Speech,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[13] W. Ping *et al.*, “Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning,” *arXiv e-prints*, p. arXiv:1710.07654, 2017.

[14] K. Ito and L. Johnson, “The LJ Speech Dataset.” 2017.

[15] A. M. V. Minh Nguyen Quan Anh, Erogol, Kuan Chen, Dawid Kobus, Takuya Ebata, Trinh Le Quang, Yunchao He, “Tensorflow-TTS,” 2021. https://github.com/TensorSpeech/TensorFlowTTS (accessed Nov. 10, 2021).

Lampiran-Lampiran

1. Bukti luaran wajib (masih draft karena hasil penelitian belum baik, perlu waktu tambahan untuk menyelesaikan pelatihan dengan epoch yang besar).
2. Bukti luaran tambahan (Jika ada)
3. Bukti dokumen realisasi kerjasama dengan mitra (Jika ada)

A Text-to-Speech Method Using Deep Learning for Bahasa Indonesia

Adhi Prahara [a,1,*], Murinto [b,2]

[a] Informatics Deparment, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] adhi.prahara@tif.uad.ac.id*; [2] murintokusno@tif.uad.ac.id
* corresponding author

ARTICLE INFO

Article history
Received
Revised
Accepted

Keywords
Keyword_1
Keyword_2
Keyword_3
Keyword_4
Keyword_5

ABSTRACT

Type your abstract here (10 pt).

1. Introduction

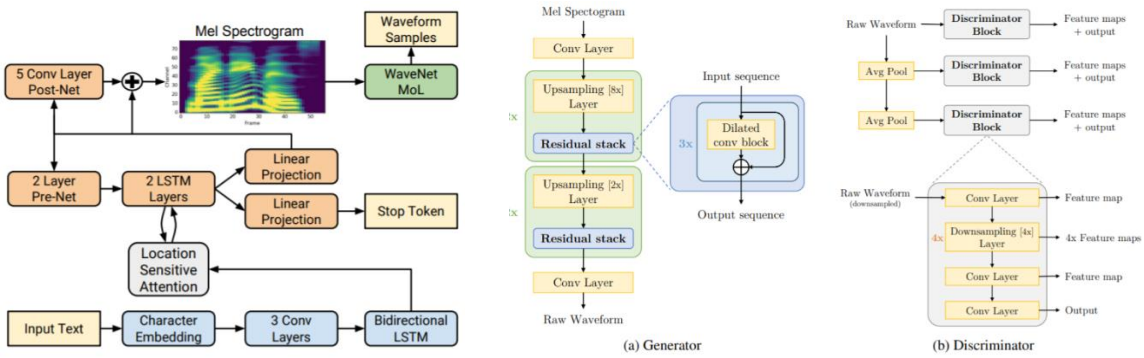
Model pengucapan terdiri dari model sintesis suara dan model konversi suara. Model sintesis suara menggunakan baseline dari model Tacotron [1], [2]. Tacotron merupakan model sintesis suara *end-to-end* yang dilatih menggunakan input dataset pasangan teks dan pengucapan. Dalam penelitian ini, model akan dilatih menggunakan dataset pengucapan bahasa Indonesia. Model Tacotron dibangun berdasarkan model sequence-to-sequence (seq2seq) yang menggunakan attention [3] sehingga tidak memerlukan penyelarasan tingkat fonem atau fitur linguistik seperti pada model WaveNet [4]. Model ini menggunakan rekonstruksi Griffin-Lim [5] untuk mensitesis pengucapannya. Hasil keluaran modelnya berupa spektrogram yang dapat digunakan untuk membuat sinyal suara dari pembicara tersebut berdasarkan masukan teks yang diberikan secara otomatis.

Model konversi suara menggunakan baseline model Generative Adversarial Network (GAN) [6]–[8] yang dilatih menggunakan inputan dari model Tacotron untuk dikonversi ke suara pembicara dari dataset yang diberikan. GAN merupakan model generatif yang biasanya digunakan untuk pembuatan citra dan transfer style dari satu citra/suara ke citra/suara yang lain. Model konversi suara juga menggunakan rekonstruksi Griffin-Lim untuk menghasilkan sinyal suara dari spektrogramnya.

2. Speech Synthesis Model

Pada penelitian ini dikembangkan model pengucapan Bahasa Indonesia bagi asisten virtual yang meliputi model sintesis suara dan model konversi suara. Model sintesis suara sudah banyak dikembangkan oleh peneliti terutama dari Google [1], [2], [4] dan Baidu [9]–

[11] dan mendapatkan hasil yang baik dari segi kealamian pengucapan untuk sintesis suara Bahasa Inggris dan Mandarin. Model konversi suara juga telah banyak dikembangkan dan mendapatkan hasil yang baik terutama menggunakan tipe-tipe model GAN [7], [8], [12] yang memang unggul untuk menangani kasus style transfer. Arsitektur model (a) Tacotron 2 dan (b) MelGAN yang digunakan pada penelitian ini ditunjukkan pada Gambar 1.

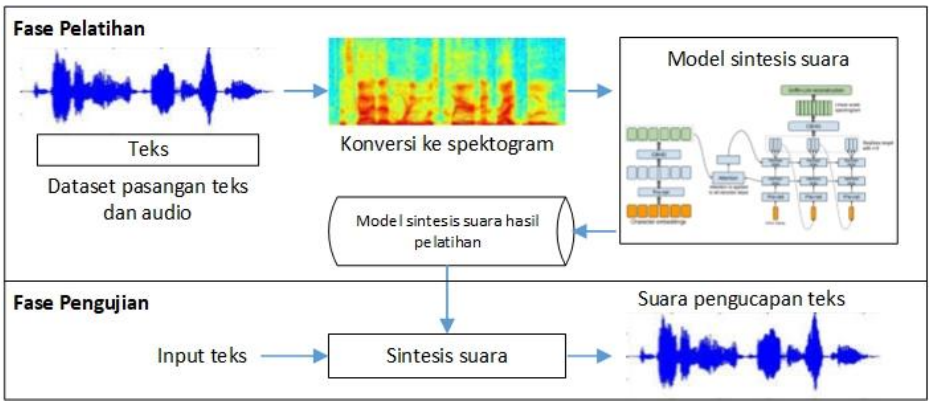


(a) Tacotron 2 [2] (b) MelGAN [6]

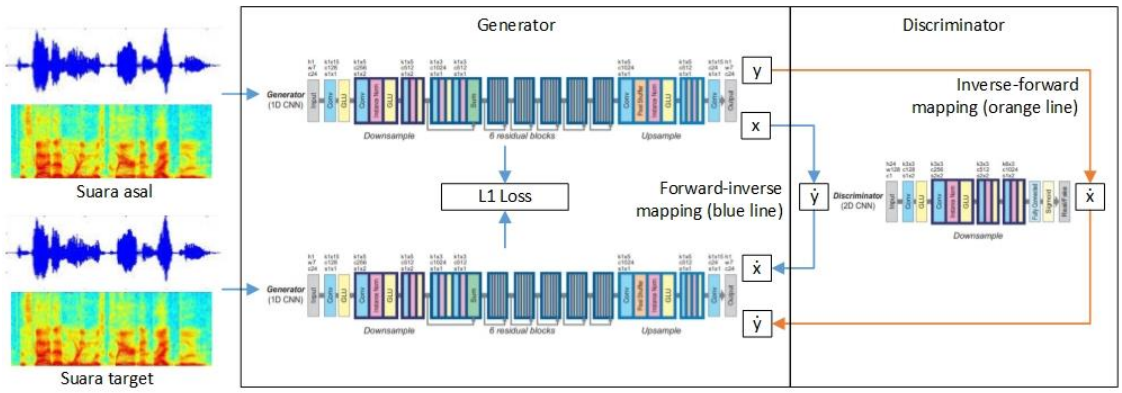
Gambar 1. Arsitektur Model (a) Tacotron 2 dan (b) MelGAN

3. Text-to-Speech Bahasa Indonesia

Pengembangan model pengucapan asisten virtual ditunjukkan pada Gambar 2. Prosedur dibagi menjadi dua bagian yaitu bagian model sintesis suara dan bagian model konversi suara untuk pengucapan Bahasa Indonesia. Pada bagian model sintesis suara, masukan untuk fase pelatihan adalah pasangan data teks dan audio pengucapan ahasa Indonesia dari beberapa pembicara. Data kemudian masuk ke bagian encoder, decoder dengan atensi dan menghasilkan spektogram yang bisa direkonstruksi menjadi sinyal suara menggunakan metode Griffin-Lim. Pada fase test, data teks yang diinputkan akan diproses oleh model yang sudah dilatih tersebut menjadi sinyal suara pengucapan pembicara tersebut. Pada bagian model konversi suara, masukan untuk fase pelatihan yaitu pasangan spektogram dari suara asal dan suara target. Data kemudian masuk ke bagian generator dan discriminator untuk dipelajari style dari suara target. Pada fase test, data suara asal akan dikonversi ke suara target menggunakan model yang sudah dilatih tersebut.



(b) Model sintesis suara



(b) Model konversi suara

Gambar 2. Prosedur dari (a) model sintesis suara dan (b) model konversi suara.

4. Results and Discussion

Dari hasil analisis kebutuhan perangkat untuk mengolah data dan membangun model sinstesis suara, model ini akan dijalankan pada Google Colab dengan Notebook berbahasa Python dan komputer dengan spesifikasi prosesor Intel Core i7, RAM 16 GB, dan kartu grafis Nvidia GeForce GTX1070. Untuk menerapkan arsitektur model digunakan tambahan library tensorflow-tts, soundfile, tensorflow, scikit-learn, panda, matplotlib dan tensorboard.

Data yang digunakan dalam penelitian ini yaitu data suara yang diambil dari dataset publik dan data suara yang di ambil dari internet. Dataset publik yang digunakan yaitu LJSpeech [13] sebagai benchmark dataset. Dataset suara Bahasa Indonesia untuk pelatihan awal model sintesis suara dicuplik dari suara artis yang diambilkan dari film pendek, vlog, dan talkshow yang diunduh dari internet. Dataset ini digunakan untuk mengembangkan arsitektur dan konfigurasi model sintesis suara yang disesuaikan dengan data suara Bahasa Indonesia. Setelah model teruji dengan baik, dataset suara Bahasa Indonesia dengan durasi waktu yang panjang dari beberapa pengisi suara akan diambil kemudian. Tabel 1 menunjukkan struktur dari dataset yang digunakan.

Tabel 1. Informasi dataset

No	Dataset	Jumlah Data	Format Data	Lama Waktu
1	LJSpeech (Bahasa Inggris)	13.100	WAV, each audio file is a single-channel 16-bit PCM WAV with a sample rate of 22050 Hz	Setiap data sekitar 1-10 detik dengan total durasi hampir 24 jam
2	Suara Artis (Bahasa Indonesia)	203	WAV, each audio file is a single-channel 16-bit PCM WAV with a sample rate of 22050 Hz	Setiap data sekitar 1-10 detik dengan total durasi hampir 60 menit

Dari hasil analisis, model sintesis suara kemudian diterapkan menggunakan bahasa pemrograman Python dan library tensorflow untuk diukur kinerjanya. Penelitian yang dilakukan hanya sampai pada model sintesis suara dan belum mencapai tahapan konversi suara. Hal ini dikarenakan hasil dari sintesis suara yang masih rendah sehingga belum memungkinkan untuk dilanjutkan ke tahapan selanjutnya.

Hasil Model Sintesis Suara

Dataset pada Tabel 1 digunakan sebagai input untuk pelatihan bagi baseline model Tacotron 2 [2] dan MelGAN [6] yang diambilkan dari library Tensorflow-TTS [14]. Prosesnya terdiri dari 3 tahapan yaitu 1) pra-proses data suara menjadi fitur mel spektogram, 2) pelatihan menggunakan model Tacotron 2 dan pelatihan menggunakan model MelGAN dan 3) hasil pelatihan digunakan untuk melakukan prediksi output spektogram oleh model Tacotron 2 yang kemudian disintesis menjadi suara oleh model model MelGAN. Detil dari tahapan tersebut dijelaskan sebagai berikut.

1. Pra-proses dataset

Pra-proes dataset akan mengekstrak beberapa fitur dari data suara yaitu ID data, mel spektogram, dan mel spektogram yang ternormalisasi ke rentang [-1,1]. Fitur tersebut dibagi menjadi data latih dan data validasi kemudian dihitung rerata dan standar deviasi untuk menstandarkan mel spektogramnya.

2. Pelatihan

Pelatihan terbagi menjadi dua tahapan yaitu pelatihan menggunakan model Tacotron 2 dan pelatihan menggunakan model MelGAN pada dataset nomor 2 yaitu suara artis (lihat Tabel 1). Pelatihan model Tacotron 2 menghasilkan model untuk menerima input teks yang diprediksi menjadi mel spektogram dari suara pembicara yang digunakan di data latih. Pelatihan model MelGAN menghasilkan model untuk menerima input mel spektogram menjadi waveform suara pembicara yang digunakan di data latih. Kami melakukan pelatihan dari awal dan tidak menggunakan pre-trained bobot dari model yang dilatih pada dataset LJSpeech karena struktur pengucapan Bahasa Indonesia dan Bahasa Inggris berbeda. Pengaturan hyperparameter menggunakan default dari library Tensorflow-TTS pada dataset LJSpeech seperti yang ditunjukkan pada Tabel 2. Pelatihan untuk Tacotron 2 menggunakan 100 epoch sedangkan pelatihan untuk MelGAN menggunakan 60 epoch karena konsumsi RAM yang cukup besar.

Tabel 2. Pengaturan hyperparameter model

Tacotron 2	MelGAN
Hop size: 256 Dataset batch_size: 32 remove_short_samples: true allow_cache: true mel_length_threshold: 32 is_shuffle: true use_fixed_shapes: true	Sampling rate: 22050 Hop size: 256 Dataset: batch_size: 16 batch_max_steps: 8192 batch_max_steps_valid: 81920 remove_short_samples: true allow_cache: true is_shuffle: true

Network hyperparameter:

```
embedding_hidden_size: 512
initializer_range: 0.02
embedding_dropout_prob: 0.1
n_speakers: 1
n_conv_encoder: 5
encoder_conv_filters: 512
encoder_conv_kernel_sizes: 5
encoder_conv_activation: 'relu'
encoder_conv_dropout_rate: 0.5
encoder_lstm_units: 256
n_prenet_layers: 2
prenet_units: 256
prenet_activation: 'relu'
prenet_dropout_rate: 0.5
n_lstm_decoder: 1
reduction_factor: 1
decoder_lstm_units: 1024
attention_dim: 128
attention_filters: 32
attention_kernel: 31
n_mels: 80
n_conv_postnet: 5
postnet_conv_filters: 512
postnet_conv_kernel_sizes: 5
postnet_dropout_rate: 0.1
attention_type: "lsa"
```

Optimizer:

```
initial_learning_rate: 0.001
end_learning_rate: 0.00001
decay_steps: 75
warmup_proportion: 0.02
weight_decay: 0.001
```

gradient_accumulation_steps: 1

Pelatihan:

```
train_max_steps: 100
save_interval_steps: 50
eval_interval_steps: 50
log_interval_steps: 20
start_schedule_teacher_forcing: 200001
start_ratio_value: 0.5
schedule_decay_steps: 50000
end_ratio_value: 0.0
```

Network hyperparameter – generator:

```
out_channels: 1
kernel_size: 7
filters: 512
upsample_scales: [8, 8, 2, 2]
stack_kernel_size: 3
stacks: 3
is_weight_norm: false
```

Network hyperparameter – discriminator:

```
out_channels: 1
scales: 3
downsample_pooling: "AveragePooling1D"
downsample_pooling_params:
    pool_size: 4
    strides: 2
kernel_sizes: [5, 3]
filters: 16
max_downsample_filters: 1024
downsample_scales: [4, 4, 4, 4]
nonlinear_activation: "LeakyReLU"
nonlinear_activation_params:
    alpha: 0.2
is_weight_norm: false
```

Adversarial loss:

lambda_feat_match: 10.0

Optimizer – generator dan discriminator:

```
lr: 0.0001
beta_1: 0.5
beta_2: 0.9
```

gradient_accumulation_steps: 1

Pelatihan:

```
train_max_steps: 60
save_interval_steps: 10
eval_interval_steps: 2
log_interval_steps: 10
discriminator_train_start_steps: 0
```

Hasil kinerja pelatihan yaitu loss ditunjukkan pada Tabel 3. Loss untuk pelatihan model Tacotron 2 dan MelGAN masih tinggi tapi terjadi penurunan disetiap epochnya. Dari benchmark dataset LJSpeech diketahui bahwa kinerja model akan baik setelah lebih

65.000-200.000 epoch [14]. Sampai saat ini kami masih melanjutkan pelatihan agar mencapai target tersebut dikarenakan terbatasnya resource Google Colab yang disediakan.

Tabel 3. Hasil kinerja pelatihan model.

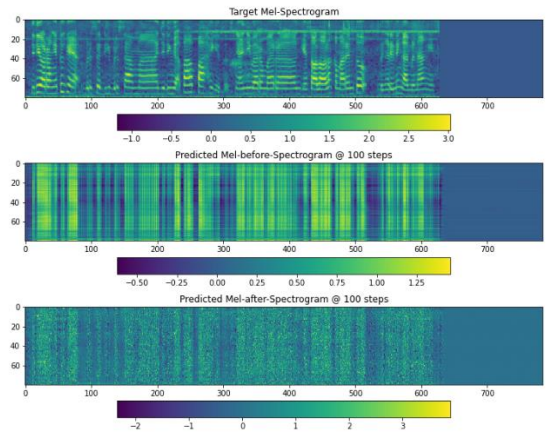
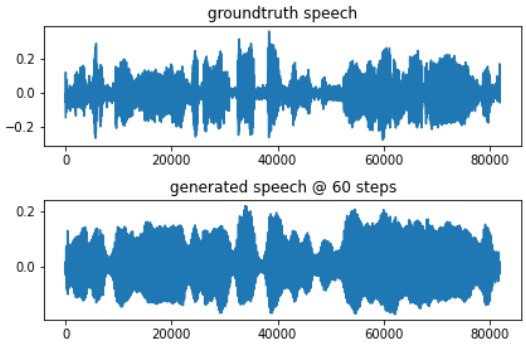
Tacotron 2		**MelGAN**	
Epoch-20 Training	Mel loss before: 0.4049 Mel loss after: 0.7906 Guided attention loss: 0.0146	Epoch-10 Training	fm_loss: 0.0045 Gen loss: 0.9779 Real loss: 0.9333 Fake loss: 0.0022 Dis loss: 0.9355 Mel spectrogram loss: 2.2060
Epoch-40 Training	Mel loss before: 0.2594 Mel loss after: 0.4246 Guided attention loss: 0.0122	Epoch-20 Training	fm_loss: 0.0041 Gen loss: 0.5738 Real loss: 0.5312 Fake loss: 0.0950 Dis loss: 0.6262 Mel spectrogram loss: 1.9270
Epoch-50 Evaluasi	Stop token loss: 0.0137 Mel loss before: 0.2176 Mel loss after: 0.3779 Guided attention loss: 0.0095	Epoch-30 Evaluasi	Adversarial loss: 0.2308 fm_loss: 0.0032 Gen loss: 0.2308 Real loss: 0.1897 Fake loss: 0.3152 Dis loss: 0.5049 Mel spectrogram loss: 2.6192
Epoch-60 Training	Mel loss before: 0.2211 Mel loss after: 0.3496 Guided attention loss: 0.0110	Epoch-40 Training	fm_loss: 0.0051 Gen loss: 0.3310 Real loss: 0.2724 Fake loss: 0.2402 Dis loss: 0.5127 Mel spectrogram loss: 1.8524
Epoch-80	Mel loss before: 0.2058	Epoch-50	fm_loss: 0.0052

Training	Mel loss after: 0.3233 Guided attention loss: 0.0106	Training	Gen loss: 0.3303 Real loss: 0.2754 Fake loss: 0.2392 Dis loss: 0.5146 Mel spectrogram loss: 1.9318
Epoch-100: Evaluasi	Stop token loss: 0.0130 Mel loss before: 0.2238 Mel loss after: 0.3259 Guided attention loss: 0.0087	Epoch-60 Evaluasi	Adversarial loss: 0.2366 fm_loss: 0.0040 Gen loss: 0.2366 Real loss: 0.1896 Fake loss: 0.3199 Dis loss: 0.5095 Mel spectrogram loss: 2.6832

3. Prediksi

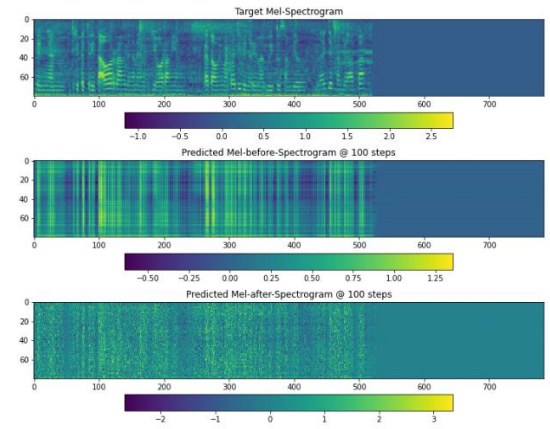
Hasil beberapa spektogram dari model Tacotron 2 pada epoch ke-100 dan hasil sinyal suara dari model MelGAN pada epoch ke-60 setelah pelatihan selesai ditunjukkan pada Tabel 4. Berdasarkan Tabel 4 dapat dilihat bahwa hasil dari pelatihan kurang baik karena lossnya masih tinggi. Untuk mengatasi hal itu diperlukan pelatihan lanjutan dengan epoch yang lebih besar yang membutuhkan waktu cukup lama dengan resource komputasi yang terbatas.

Tabel 4. Hasil prediksi pelatihan model.

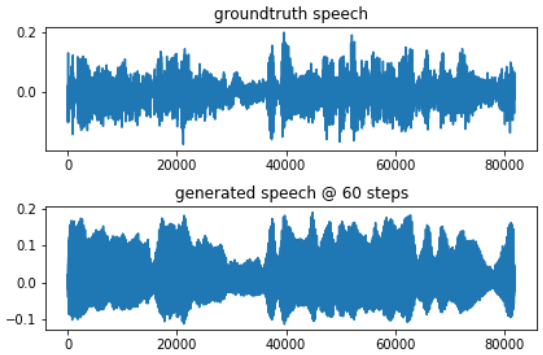
Nama file: RS001-0016 Teks: Jadi orang itu kayak bersedih bersama jadi ga apa-apa gitu, nyanyi bareng-bareng kayak seolah-olah kemarin tuh dengerinnya ga sambil nangis.	
	
Tactoron 2 - Spektogram	MelGAN - Suara

Nama file: RS001-0019

Teks: Dengan cerita-cerita orang dengan energi orang yang kata om Iwan tadi dengerin lagu itu sambil belajar sambil apa



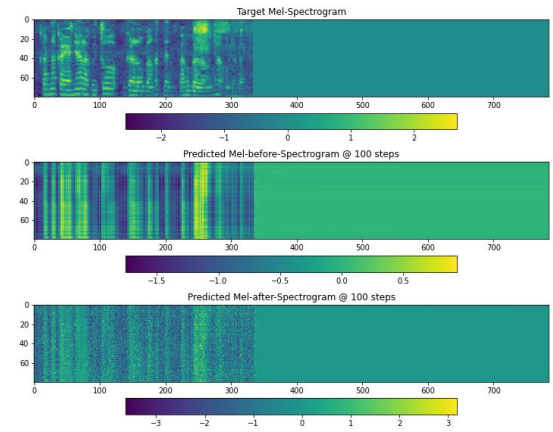
Tactoron 2 - Spektogram



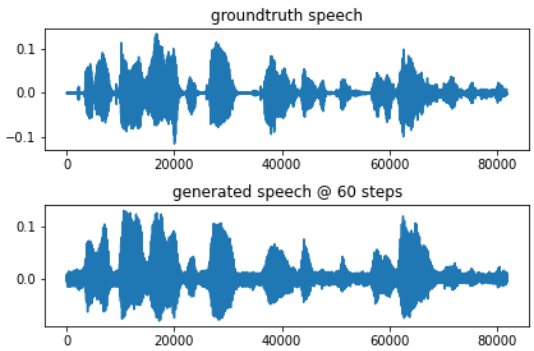
MelGAN - Suara

Nama file: RS005-0100

Teks: Karena kayaknya lagu itu galau banget dan lagu galaunya udah banyak



Tactoron 2 - Spektogram



MelGAN - Suara

5. Conclusion

References

[1] Y. Wang *et al.*, "Tacotron: Towards End-to-End Speech Synthesis." 2017.

[2] J. Shen *et al.*, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783, doi: 10.1109/ICASSP.2018.8461368.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to

Align and Translate." 2016.

[4] A. van den Oord *et al.*, "WaveNet: A Generative Model for Raw Audio." 2016.

[5] D. Griffin and Jae Lim, "Signal estimation from modified short-time Fourier transform," in *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Apr. 1983, vol. 8, pp. 804–807, doi: 10.1109/ICASSP.1983.1172092.

[6] A. C. Kumar, Kundan and Kumar, Rithesh and de Boissiere, Thibault and Gestin, Lucas and Teoh, Wei Zhen and Sotelo, Jose and de Br\'{e}bisson, Alexandre and Bengio, Yoshua and Courville, "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis," in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

[7] T. Kaneko and H. Kameoka, "Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks." 2017.

[8] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: non-parallel many-to-many Voice Conversion Using Star Generative Adversarial Networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 266–273, doi: 10.1109/SLT.2018.8639535.

[9] S. Ö. Arik *et al.*, "Deep Voice: Real-time Neural Text-to-Speech," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 195–204, [Online]. Available: http://proceedings.mlr.press/v70/arik17a.html.

[10] S. Arik *et al.*, "Deep Voice 2: Multi-Speaker Neural Text-to-Speech," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[11] W. Ping *et al.*, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning," *arXiv e-prints*, p. arXiv:1710.07654, 2017.

[12] Y. Gao, R. Singh, and B. Raj, "Voice Impersonation using Generative Adversarial Networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2506–2510, doi: 10.1109/ICASSP.2018.8462018.

[13] K. Ito and L. Johnson, "The LJ Speech Dataset." 2017.

[14] A. M. V. Minh Nguyen Quan Anh, Erogol, Kuan Chen, Dawid Kobus, Takuya Ebata, Trinh Le Quang, Yunchao He, "Tensorflow-TTS," 2021. https://github.com/TensorSpeech/TensorFlowTTS (accessed Nov. 10, 2021).