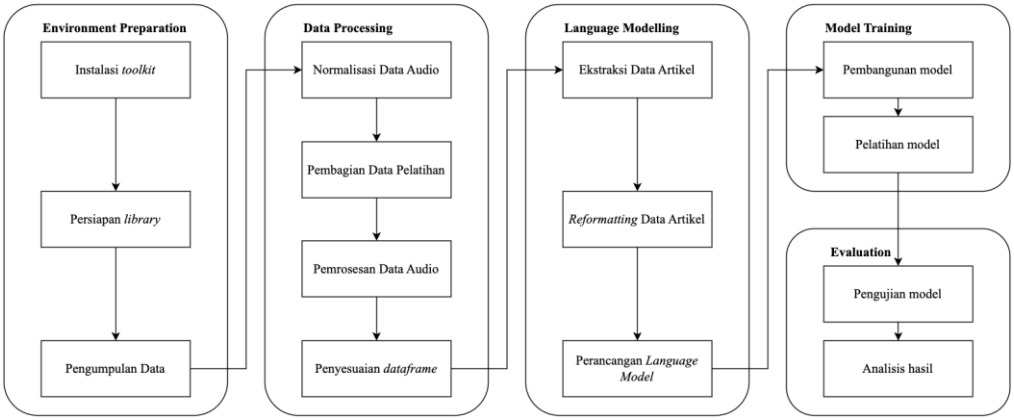
BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1. Metodologi Penelitian

Secara garis besar, penelitian dilakukan dalam lima fase, yaitu persiapan *environment*, pemrosesan data, perancangan *language model*, pelatihan *acoustic model*, dan evaluasi hasil pelatihan. Perincian tahap-tahap yang dilakukan pada masing-masing fase ditampilkan pada Gambar 3.1 sebagai berikut.



Gambar 3.1 Diagram Tahap Penelitian

3.1.1. Persiapan *Environment*

Sebuah *environment* dengan *library* dan data yang dibutuhkan disiapkan untuk memastikan fase-fase selanjutnya dapat dilakukan.

3.1.1.1. Instalasi *Toolkit*

Toolkit OpenSeq2Seq dari dibutuhkan pada perangkat sebagai prasyarat penelitian. Terdapat konfigurasi komponen-komponen untuk membangun model dengan arsitektur *QuartzNet* pada *OpenSeq2Seq* beserta *python script* untuk melakukan pelatihan dan evaluasi model. Perancangan *language model* dilakukan menggunakan *KenLM Language Model Toolkit* yang sudah tersedia pada *toolkit OpenSeq2Seq*.

3.1.1.2. Persiapan *Library*

Memuat *library* serta *dependency* dari setiap *library* yang akan digunakan dalam tahapan pemrosesan data, pelatihan model dan evaluasi hasil ke dalam *environment*.

3.1.1.3. Pengumpulan Data

Penelitian menggunakan *dataset Mozilla Common Voice Corpus* [14] versi 12.0 dalam bahasa Indonesia yang terdiri dari 62 jam data rekaman audio frasa ataupun kalimat pendek yang masing-masing memiliki durasi dibawah 10 detik dalam format *.mp3* dengan *sampling rate* 48kHz. Hanya bagian *dataset* yang tervalidasi digunakan dalam penelitian, yaitu *subset train, test* dan *dev*. Jumlah data yang terdapat di masing-masing *subset* ditampilkan pada Tabel 3.1 sebagai berikut.

	training	**test**	**validation**
Jumlah File Data Audio	5040	3647	3288

Tabel 3.1 Pembagian Data Training, Test dan Validasi

Penelitian juga menggunakan *database dump* Wikipedia Bahasa Indonesia sebagai data untuk merancang *language model* yang akan digunakan pada tahap evaluasi model.

3.1.2. Pemrosesan Data

Fase pemrosesan data dilakukan kepada *dataset* yang dikumpulkan untuk mendapatkan data *disordered speech* yang akan digunakan pada tahap pelatihan model. Hasil akhir dari pemrosesan data adalah sembilan *dataset* baru yang mengandung campuran antara data *normal speech* dan *disordered speech*. Pemrosesan juga dilakukan pada keseluruhan *subset test* dan *dev* untuk melakukan pengujian kinerja model dalam mengenal data *disordered speech*.

3.1.2.1. Normalisasi Data

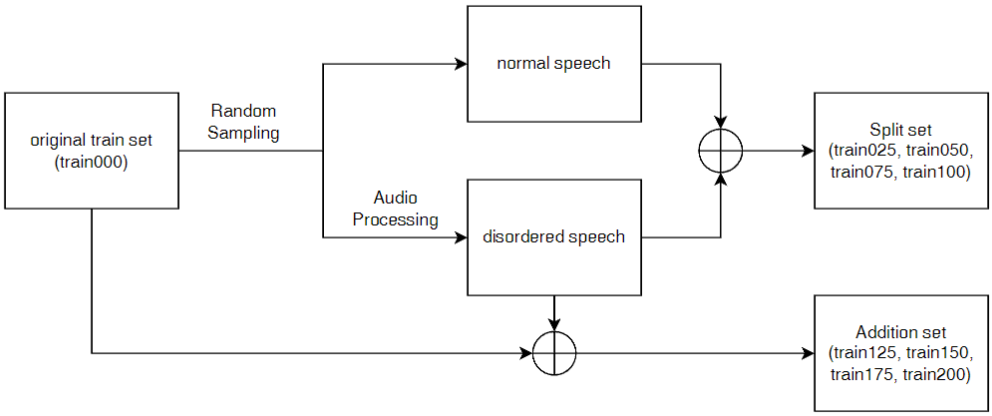
Tahap normalisasi data dilakukan untuk menyesuaikan data audio yang dikumpulkan dengan spesifikasi untuk input model *QuartzNet*, yaitu file audio dengan format *.wav* dan *sampling rate* 16kHz.

3.1.2.2. Pembagian Data Pelatihan

Sampling pada *subset train* dilakukan untuk memisahkan data yang akan diproses menjadi *disordered speech* sehingga rasio data yang sesuai dapat dicapai. Rasio data *disordered speech* yang diputuskan untuk digunakan dalam tahap pelatihan model adalah 0%, 25%, 50%, 75% dan 100%. Terdapat dua metode yang dilakukan untuk mendapatkan *dataset* mengandung data *disordered speech* dan data *normal speech* untuk pelatihan model.

Metode pertama dilakukan dengan memisahkan sejumlah data sesuai dengan persentase rasio data yang sudah ditetapkan untuk diproses menjadi data *disordered speech.* Data yang sudah diproses akan digabungkan kembali dengan data *normal speech* untuk melengkapi *dataset.*

Metode kedua dilakukan dengan melakukan penambahan data pada *subset train* menggunakan data *disordered speech* yang dihasilkan dari pemrosesan dalam metode pertama. Dataset yang ditambahkan data *disordered speech* akan memiliki rasio data *disordered speech* yang berbeda dari rasio yang sudah ditetapkan akibat jumlah total data yang berbeda. Gambar 3.2 menunjukkan diagram yang menggambarkan metode yang dijelaskan untuk mendapatkan masing-masing *dataset.*



Gambar 3.2 Flow Pembagian Data

Dua jenis metode pembagian data digunakan dengan tujuan untuk mengamati kemungkinan adanya pengaruh jumlah data total pada kinerja model.

3.1.2.3. Pemrosesan Audio

Data *disordered speech* dibuat dari data *normal speech* dengan metode *speed perturbation* menggunakan *Rubber Band Audio Time Stretcher Library. Speed perturbation* merupakan metode pemrosesan yang mengubah kecepatan tanpa mengubah *framerate* dari sinyal dengan cara *resampling* sinyal tersebut dengan sebuah nilai faktor. Untuk meningkatkan variasi pada data *disordered speech*, pemrosesan menggunakan empat nilai faktor berbeda berdasarkan faktor yang disediakan oleh penelitian Vacchani *et al.* [7], yaitu 1.2, 1.4, 1.8 dan 2.0. Validasi yang dilakukan oleh penelitian untuk memastikan metode pemrosesan *speed perturbation* dengan nilai-nilai faktor yang disebutkan sebelumnya dapat digunakan untuk membuat data *disordered speech* dari data *normal speech* adalah dengan cara melakukan *inference* dengan data *disordered speech* pada model yang dilatih menggunakan data *normal speech* yang diproses dengan metode *speed perturbation. Word error rate* yang rendah dari hasil *inference* menandakan bahwa data yang diproses menggunakan metode *speed*

perturbation memiliki karakteristik yang mirip dengan data *disordered speech* yang sesungguhnya sehingga metode dapat digunakan untuk mengubah data *normal speech* menjadi data *disordered speech. Speed perturbation* juga digunakan untuk memproses keseluruhan *dataset test* menjadi data *disordered speech*.

3.1.2.4. Penyesuaian *Dataframe*

Data pada *dataframe* disesuaikan sehingga menunjuk kepada data yang benar saat melakukan pelatihan dan pengujian model. Proses yang dilakukan merupakan mengubah *filepath* pada *dataframe* untuk menunjuk kepada lokasi masing-masing *dataset*, mengubah format data pada nama *file* menjadi *.wav*, serta menghapus tanda baca dan huruf kapital pada transkrip.

3.1.3. Perancangan *Language Model*

Language model yang dirancang merupakan sebuah 3-gram *language model* menggunakan hasil ekstraksi *dump* dari artikel-artikel *wikipedia* berbahasa Indonesia dengan menggunakan *library* wikiextractor. Data yang diekstraksi merupakan *body* dari setiap artikel yang kemudian diformat ulang sehingga tidak mengandung tanda baca maupun *special character* untuk digunakan pada perancangan *language model. Language model* dirancang menggunakan *KenLM Language Model Toolkit.*

3.1.4. Pelatihan Model

Model dilatih dengan masing-masing *dataset* yang didapatkan dari fase pemrosesan data. *Subset dev* digunakan untuk melakukan validasi berkala pada proses pelatihan.

3.1.4.1. Arsitektur Model

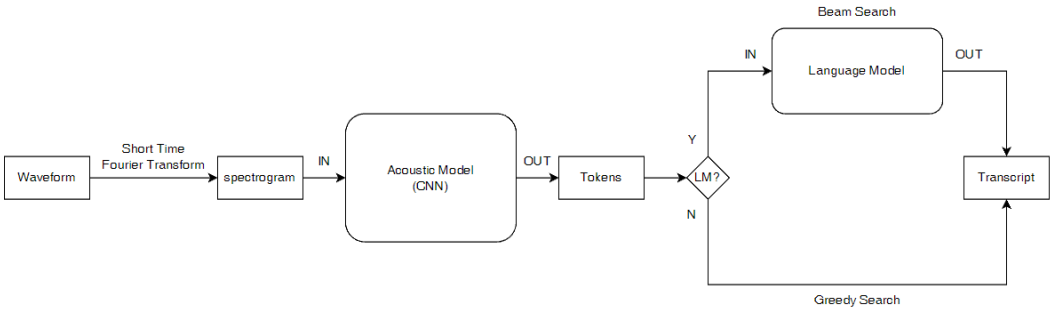
Model yang dibangun untuk penelitian merupakan arsitektur *QuartzNet* 5x5 [10]. Penelitian menggunakan model dengan jumlah

parameter yang rendah untuk mencegah terjadi *overfitting* terlalu awal pada saat pelatihan akibat jumlah data pelatihan terbatas. Perincian masing-masing parameter *layer* pada model yang dibangun ditampilkan pada Tabel 3.2 sebagai berikut.

Block	Repeat	Kernel	Channel Output
C_1	1	33	256
B_1	5	33	256
B_2	5	39	256
B_3	5	51	512
B_4	5	63	512
B_5	5	75	512
C_2	1	87	512
C_3	1	1	1024
C_4	1	1	||*label*||

Tabel 3.2 Konfigurasi Parameter Layer Model QuartzNet 5x5

Model *QuartzNet* menerima input berupa *mel-spectrogram* dari sinyal audio dan mengeluarkan output *token* berupa karakter. Karakter-karakter yang digunakan sebagai *token* pada output model dalam penelitian adalah `[a-z]` dan spasi. Output dari model dapat diproses lebih lanjut menggunakan algoritma *beam search* dengan *language model* untuk meningkatkan akurasi pada hasil transkripsi. Gambar 3.3 menunjukkan *pipeline* sistem *speech recognition* yang dirancang pada penelitian dimulai dari input berupa sinyal audio menjadi output berupa hasil transkripsi teks.



Gambar 3.3 Pipeline Sistem Speech Recognition

3.1.4.2. Tahap Pelatihan

OpenSeq2Seq tidak memiliki fungsi *early stopping* sehingga untuk mendapatkan nilai *epoch* yang sesuai untuk digunakan pada pelatihan model, dilakukan pelatihan awal pada satu model sebanyak 50 *epoch* dengan validasi berkala. Masing-masing model dilatih menggunakan nilai *epoch* dengan *validation loss* terendah yang tersimpan dari tahap pelatihan awal.

3.1.5. Evaluasi Hasil Pelatihan

3.1.5.1. Pengujian Model

Kinerja model diuji dengan melakukan *inference* menggunakan *subset test*. Hasil *inference* berupa hasil transkripsi dalam format teks oleh model terhadap data audio yang ada pada *subset test*. *Inference* dilakukan pada masing-masing model sebanyak dua kali untuk mendapatkan transkripsi menggunakan algoritma *greedy search* tanpa *language model* dan transkripsi menggunakan algoritma *beam search* dengan 3-gram *language model*. Transkripsi dilakukan menggunakan dua jenis algoritma sebagai pertimbangan untuk merancang sistem *speech recognition* dengan kinerja terbaik.

3.1.5.2. Analisis Hasil

Analisis kinerja model dengan masing-masing data pelatihan dilakukan dengan memperhitungkan *error rate* pada hasil *inference. Error rate* merupakan sebuah persentase terjadinya kesalahan pada hasil transkripsi saat dibandingkan dengan transkrip sebenarnya dari input. Terdapat tiga jenis kesalahan yang dapat diidentifikasi dari segi kata maupun karakter, yaitu *substitution, insertion* dan *deletion.* Contoh dari masing-masing kesalahan transkripsi ditampilkan pada Tabel 3.3 sebagai berikut.

	Skala Error	
Jenis Error	**Karakter**	**Kata**
-	PERGI	BESOK HARI SENIN
Substitution	PEPGI	BESOK DARI SENIN
Insertion	PERGGI	BESOK HARI DAN SENIN
Deletion	PEGI	BESOK SENIN

Tabel 3.3 Jenis Error dalam Proses Transkripsi

Nilai *error rate* yang rendah merupakan indikasi bahwa model dapat melakukan transkripsi audio secara akurat. *Error rate* yang diperhitungkan dalam analisis berupa *character error rate* (CER) untuk mengevaluasi kinerja model dalam skala karakter, dan *word error rate* (WER) untuk mengevaluasi kinerja model dalam skala kata. Evaluasi *error rate* dilakukan dalam skala karakter untuk mengukur kinerja pada *acoustic model* dalam mengenal masing-masing fonem pada input dan evaluasi *error rate* dalam skala kata dilakukan untuk mengukur kinerja sistem *speech recognition* secara keseluruhan. Penilaian WER minimum untuk model *speech recognition* yang layak digunakan adalah dibawah 20%.