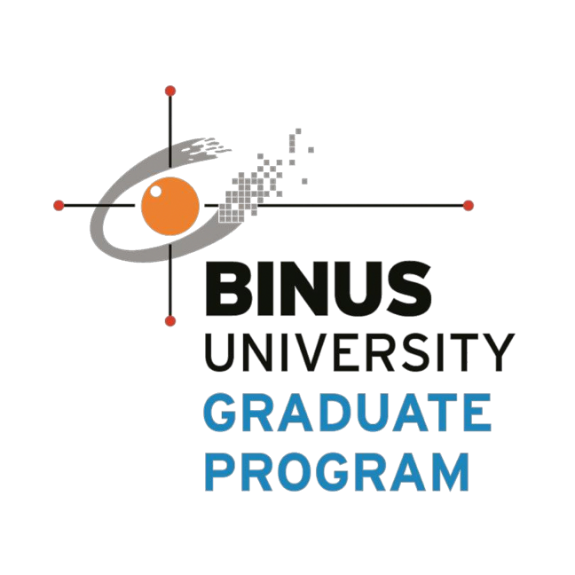
**PROPOSAL**

**Pendekatan *Deep learning attentive recurrent network* dengan *model compression* dan *quantization* untukmengurangi kebisingan agar meningkatkan kejelasan suara manusia**



**RESEARCH**

*SHOFYAN*

*2301980064*

**Program Pascasarjana Ilmu Komputer**

**PROGRAM STUDI TEKNIK INFORMATIKA JENJANG S2**

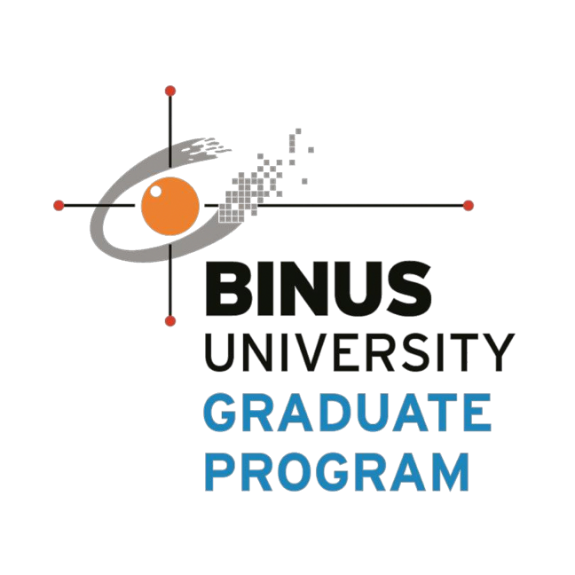
**UNIVERSITAS BINA NUSANTARA**

**JAKARTA**

**2022**

**PROPOSAL**

**Pendekatan *Deep learning attentive recurrent network* dengan *model compression* dan *quantization* untukmengurangi kebisingan agar meningkatkan kejelasan suara manusia**



**RESEARCH**

*SHOFYAN*

*2001847780*

Pembimbing:

**Iman Herwidiana Kartowisastro, Ph.D.**

08-11-2022

# Daftar Isi

[Daftar Isi iii](#_Toc111148365)

[Daftar Gambar v](#_Toc111148366)

[Daftar Persamaan vi](#_Toc111148367)

[Daftar Tabel viii](#_Toc111148368)

[BAB 1 PENDAHULUAN 1](#_Toc111148369)

[1.1 Latar Belakang 1](#_Toc111148370)

[1.1.1 Spectral subtractive algorithms 2](#_Toc111148371)

[1.1.2 Statistical model-based algorithms 4](#_Toc111148372)

[1.1.3 Subspace Algorithm 5](#_Toc111148373)

[1.1.4 Binary Mask 5](#_Toc111148374)

[1.1.5 Deep Learning algorithm 6](#_Toc111148375)

[1.2 Rumusan Masalah 8](#_Toc111148376)

[1.3 Tujuan dan Manfaat 8](#_Toc111148377)

[1.4 Ruang Lingkup 9](#_Toc111148378)

[BAB 2 LANDASAN TEORI 10](#_Toc111148379)

[2.1 Perkembangan *deep learning* 10](#_Toc111148380)

[*2.2* *Digital Signal Processing* 21](#_Toc111148381)

[*2.3* *Machine Learning* 22](#_Toc111148382)

[*2.4* *Deep Learning* 22](#_Toc111148383)

[2.5 Optimizer algorithm 23](#_Toc111148384)

[2.5.1 Gradient Descent 23](#_Toc111148385)

[2.5.2 Momentum 25](#_Toc111148386)

[2.5.3 Adam 26](#_Toc111148387)

[2.6 Signal Compression Signal Quantization 27](#_Toc111148388)

[2.6.1 Kompresi Sinyal 27](#_Toc111148389)

[2.6.2 Kuantisasi Sinyal 28](#_Toc111148390)

[BAB 3 METODOLOGI 33](#_Toc111148391)

[3.1 Metodologi Penelitian 33](#_Toc111148392)

[3.2 Pengumpulan dan Prapemrosesan Dataset 36](#_Toc111148393)

[3.3 Ekstraksi Fitur 37](#_Toc111148394)

[3.4 Pemisahan Data 39](#_Toc111148395)

[3.5 Permodelan *Attentive Recurrent Network* 39](#_Toc111148396)

[3.5.1 Quantization 40](#_Toc111148397)

[3.5.2 Compression 40](#_Toc111148398)

[3.5.3 Normalization Layer 40](#_Toc111148399)

[3.5.4 Recurrent Neural Network 41](#_Toc111148400)

[3.5.5 Self-attention Block 42](#_Toc111148401)

[3.5.6 Feedforward Block 43](#_Toc111148402)

[3.6 Evaluasi Performa 44](#_Toc111148403)

[3.6.1 Short-time objective intelligibility (STOI) 44](#_Toc111148404)

[3.6.2 Perceptual evaluation of speech quality (PESQ) 45](#_Toc111148405)

[DAFTAR PUSTAKA 46](#_Toc111148406)

# Daftar Gambar

[Gambar 2.1 Mekanisme Kompresi 27](#_Toc111148408)

[Gambar 2.2 Rangkaian Enkoder (Darnila & Safwandi, 2020) 28](#_Toc111148409)

[Gambar 2.3 Mekanisme Kompresi 29](#_Toc111148410)

[Gambar 2.4 Rangkaian Enkoder(Darnila & Safwandi, 2020) 30](#_Toc111148411)

[Gambar 3.1 Metode Penelitian 33](#_Toc111148412)

[Gambar 3.2 Diagram metode *speech enhancement* 35](#_Toc111148413)

[Gambar 3.3 Diagram ARN. 39](#_Toc111148414)

[Gambar 3.4 *Quantization* 40](#_Toc111148415)

[Gambar 3.5 Diagram ilustrasi dari LSTM 41](#_Toc111148416)

[Gambar 3.6 *Feedforward Block* 43](#_Toc111148417)

[Gambar 3.7 Struktur PESQ Model (Rix, 2001) 45](#_Toc111148418)

# Daftar Persamaan

[(1.1) 2](#_Toc110698514)

[(1.2) 3](#_Toc110698515)

[‎(1.3) 3](#_Toc110698516)

[(1.4) 3](#_Toc110698517)

[(1.5) 4](#_Toc110698518)

[(‎1.6) 4](#_Toc110698519)

[(2.1) 24](#_Toc110698520)

[(2.2) 24](#_Toc110698521)

[(‎2.3) 25](#_Toc110698522)

[(‎2.4) 25](#_Toc110698523)

[(2.5) 25](#_Toc110698524)

[(2.6) 26](#_Toc110698525)

[‎(2.7) 26](#_Toc110698526)

[(2.8) 26](#_Toc110698527)

[(‎2.9) 29](#_Toc110698528)

[(2.10) 29](#_Toc110698529)

[(‎2.11) 29](#_Toc110698530)

[(2.12) 31](#_Toc110698531)

[(2.13) 32](#_Toc110698532)

[(2.14) 32](#_Toc110698533)

[(‎3.1) 36](#_Toc110698534)

[(‎3.2) 36](#_Toc110698535)

[(‎3.3) 37](#_Toc110698536)

[(3.4) 37](#_Toc110698537)

[(‎3.5) 37](#_Toc110698538)

[(‎3.6) 39](#_Toc110698539)

[(3.7) 40](#_Toc110698540)

[(‎3.8) 40](#_Toc110698541)

[(3.9) 40](#_Toc110698542)

[(3.10) 41](#_Toc110698543)

[(3.11) 42](#_Toc110698544)

[(‎3.12) 43](#_Toc110698545)

# Daftar Tabel

[Tabel ‎2.1 Perkembangan model *deep learning speech enchancement* 21](#_Toc110699313)

[Tabel ‎2.2 Perbedaan Metode Trunkasi dan Pembulatan 29](#_Toc110699314)

[Tabel ‎2.3 Perbedaan Metode Trunkasi dan Pembulatan 32](#_Toc110699315)

# PENDAHULUAN

## Latar Belakang

Pandemi *Covid-19* pada akhirnya mengubah cara bekerja yang sebelumnya betatap muka langsung di kantor, kini harus bekerja dari rumah (*Work from Home*), komunikasi menggunakan media *video call* sudah menjadi bagian dari koordinasi dengan rekan kerja. Keadaan ruangan dirumah tidak seperti di kantor yang sudah diatur sedemikian rupa sehingga bebas dari gannguan suara seperti kendaraan yang lewat, anak kecil yang teriak, suara pedagang asongan menjajakan makanan, suara tetangga sedang memperbaiki rumah dan lain lain. Hal ini menimbulkan gangguan suara ketika berkomunikasi dengan rekan kerja. Pada kasus ini kita membutuhkan penghilang kebisingan suara ketika bekomunikasi sehingga pesan yang disampaikan dapat diterima dengan jelas.

*Speech enhancement* dapat didefinisikan sebagai suatu proses yang berfokus kepada pembersihan suatu sinyal suara yang telah dipengaruhi oleh sinyal noise tambahan. Dalam penerapannya, speech enhancement diperlukan untuk sinyal suara yang berasal dari tempat yang mengandung noise seperti pabrik yang menghasilkan kebisingan dari mesin. Komunikasi suara melalui telefon genggam yang dilakukan di tempat demikian akan terpengaruh oleh sinyal suara lain yang berada di tempat tersebut. Algoritma speech enhancement yang dapat digunakan untuk memproses suatu sinyal suara antara lain spectral subtractive algorithms, statistical model-based algorithms, subspace algorithms, dan binary mask algorithms (Loizou, 2013).

*Speech enhancement* dapat didefinisikan sebagai suatu proses yang berfokus kepada pembersihan suatu sinyal suara yang telah dipengaruhi oleh sinyal noise tambahan. Dalam penerapannya, speech enhancement diperlukan untuk sinyal suara yang berasal dari tempat yang mengandung noise seperti pabrik yang menghasilkan kebisingan dari mesin. Komunikasi suara melalui telefon genggam yang dilakukan di tempat demikian akan terpengaruh oleh sinyal suara lain yang berada di tempat tersebut. Algoritma speech enhancement yang dapat digunakan untuk memproses suatu sinyal suara antara lain spectral subtractive algorithms, statistical model-based algorithms, subspace algorithms, dan binary mask algorithms (Loizou, 2013).

### Spectral subtractive algorithms

Metode pertama yang dapat digunakan adalah spectral subtractive algorithm. Spectral subtractive algorithm dapat didefinisikan sebagai metode restorasi spektrum pada suatu sinyal dari suatu sinyal yang mengandung noise. Spektrum noise pada suatu sinyal dapat diprediksi dan diperbarui selama terdapat sinyal bersih dan noise pada suatu sinyal. Dalam penerapannya, sinyal dapat ditemukan dengan cara mengurangi noise pada suatu sinyal. Namun, pada penerapannya cukup sulit untuk menghilangkan noise yang bersifat acak, sehingga hal yang mungkin dilakukan adalah mengurangi pengaruh dari rata-rata noise pada suatu spektrum sinyal. Secara umum, sinyal yang mengandung noise dapat dinyatakan dalam bentuk (1.1) sebagai berikut:

|  |  |  |
| --- | --- | --- |
|  |  | (1.1) |

dengan adalah sinyal, adalah noise pada sinyal, dan adalah indeks waktu. Dalam domain frekuensi, sinyal dapat dinyatakan dalam bentuk (1.2) sebagai berikut:

|  |  |  |
| --- | --- | --- |
|  |  | (1.2) |

dengan adalah sinyal dalam domain frekuensi yang telah ditransformasi menggunakan transformasi fourier, adalah *noise* yang telah ditransformasi menggunakan transformasi fourier. Dalam metode *spectral subtractive algorithm* sinyal dalam domain waktu di-*buffered* dan dibagi dalam rentang N sampel dan tiap sampel di-*window* menggunakan window Hamming atau Hanning. Setelah itu, tiap segmen yang telah di-*window* ditransformasi menggunakan transformasi Fourier disktrit (DFT). *Window* digunakan dengan tujuan untuk meringankan pengaruh diskontinuitas pada pada sampel. Persamaan windowing suatu sinyal dapat dijabarkan dalam persamaan berikut:

|  |  |  |
| --- | --- | --- |
|  |  | (1.3) |

dalam domain frekuensi, persamaan windowing dapat dinyatakan sebagai:

|  |  |  |
| --- | --- | --- |
|  |  | (1.4) |

dengan \* adalah operasi konvolusi. Secara umum, metode spectral subtraction dapat dinyatakan dalam bentuk:

|  |  |  |
| --- | --- | --- |
|  |  | (1.5) |

|  |  |  |
| --- | --- | --- |
|  |  | (1.6) |

dengan b=1 untuk ekstraksi magnitude dan b=2 untuk ekstraksi daya. Parameter α merupakan parameter untuk menentukan jumlah sinyal noise yang akan diekstrak. Untuk pengurangan noise penuh α=1 sedangkan untuk pengurangan noise berlebih α>1 (Vaseghi, 2000).

Kekurangan dari metode *spectral subtractive* adalah metode ini menggunakan nilai parameter tetap, sehingga metode ini tidak dapat beradaptasi dengan tingkat sinyal *noise* yang ada. Optimisasi parameter pada metode ini tidak mudah jika sinyal noise yang diberikan tidak bersifat datar. Salah satu penerapan dari metode ini adalah pada ekstraksi pita ganda. Didapatkan sinyal *noise* sisa yang tidak dapat dilemahkan (Upadhyay & Karmakar, 2015).

### Statistical model-based algorithms

Metode kedua yang dapat digunakan adalah metode *Statistical model-based algorithms*. Metode statistical model-based algorithms dapat didefinisikan sebagai metode untuk mengestimasi sinyal asli dengan pendekatan model statistika. Tahap pertama yang harus dilakukan pada metode ini adalah memformulasi bentuk sinyal dan *noise*. Selanjutnya, hal yang harus dilakukan adalah mengkodekan sinyal yang hendak diproses. Kemudan, sinyal yang telah dikodekan akan diklasifikasikan antara sinyal dan noise. Untuk memeriksa performa model dalam mengklasifikasikan antara sinyal dan *noise*, dapat digunakan metrics evaluasi seperti *minimum mean squared error*. Salah satu model yang dapat digunakan pada metode ini adalah menggunakan jaringan syaraf tiruan. Jaringan syaraf tiruan dapat dilatih untuk mempelajari sinyal dan *noise* lalu memberikan nilai parameter yang terbaik pada *filter* yang digunakan (Ephraim, 1992).

### Subspace Algorithm

Metode ketiga yang dapat digunakan adalah metode *subspace algorithm*. Pada metode ini, digunakan dekomposisi *eigen value* dan transformasi Karhunen-Loeve untuk memproyeksikan suatu sinyal suara dan kemudian memisahkan antara sinyal dan noise pada suatu suara, lalu noise yang terpisahkan pada suatu sinyal suara dapat dihilangkan agar menghasilkan sinyal suara dengan kualitas yang lebih baik. Untuk meningkatkan kualitas dari sinyal yang diproses dengan menggunakan subspace algorithm, metode *perceptual feature* dapat ditambahkan untuk mengurangi distorsi pada sinyal. Berdasarkan penelitian yang telah dilakukan oleh (Surendran & Kumar, 2015) penggunaan *perceptual feature* pada metode *subspace algorithm* dapat meningkatkan nilai SNR (*Signal to Noise Ratio*) pada model.

### Binary Mask

Metode keempat yang dapat digunakan adalah metode *binary mask algorithm*. *Binary mask algorithm* dapat didefinisikan sebagai algoritma yang mengidentifikasi antara sinyal suara dan *noise* berdasarkan perbandingan antara parameter *signal to noise ratio* (SNR) pada setiap satuan TF terhadap titik ambang batas tertentu. Tahap pertama yang dilakukan pada metode ini adalah mengkonversi sinyal suara yang mengandung noise dalam bentuk vektor ke dalam bentuk matriks. Tahap ini bertujuan untuk mengubah suatu sinyal yang awal nya dalam bentuk deret menjadi bentuk matriks. Pada tahap ini, sinyal dapat dibagi menjadi matriks dengan ukuran 256 baris yang diambil setiap 32 sample dan setiap 32 sample dilakukan *windowing* dengan menggunakan window hanning. Tahap kedua yang dilakukan adalah menerapkan algoritma FFT terhadap sinyal yang telah diubah dalam bentuk matriks. Tahap ketiga yang dilakukan adalah menghitung nilai SNR pada sinyal yang telah ditransformasi menggunakan transformasi fourier. Tahap keempat adalah inisiasi batas ambang minimal dari nilai SNR untuk diteruskan menjadi nilai satu. Pada tahap ini, sinyal yang diteruskan menjadi nilai satu, dianggap menjadi sinyal suara jernih sedangkan sinyal yang diteruskan menjadi nol, dianggap menjadi sinyal *noise*. Tahap kelima adalah, menerapkan Inverse Fast Fourier Transform (IFFT), tahap ini bertujuan untuk mengubah sinyal dalam domain frekuensi ke dalam domain waktu. Tahap terakhir adalah mengubah sinyal dalam bentuk matriks menjadi vektor, tahap ini bertujuan untuk mengembalikan sinyal dalam bentuk matriks saat diproses menjadi bentuk sinyal kontinu (Patil & Mane, n.d.).

### Deep Learning algorithm

Dalam beberapa tahun terakhir, metode peningkatan kualitas suara manusia *supervised* berbasis *deep neural network* (DNN) telah menunjukkan kemajuan besar. Dalam metode peningkatan kualitas suara manusia yang *supervised*, desain target pelatihan memainkan peran penting. Banyak metode DNN telah diadopsi dari pemrosesan gambar, ada perbedaan penting antara domain pada audio. Sampel audio mentah membentuk sinyal deret waktu satu dimensi yang pada dasarnya berbeda dari gambar yang membentuk dua dimensi. Sinyal audio biasanya diubah menjadi representasi waktu-frekuensi dua dimensi untuk diproses, tetapi dua sumbu, waktu dan frekuensi, tidak homogen seperti sumbu horizontal dan vertikal dalam sebuah gambar. Gambar adalah *snapshot* instan dari target dan sering dianalisis secara keseluruhan atau sebagian dengan sedikit batasan urutan; namun sinyal audio harus dipelajari secara berurutan dalam urutan kronologis (Purwins, 2019). Properti ini memunculkan solusi khusus audio yang berdasarkan urutan waktu.

Pendekatan metode *convolutional recurrent neural network* (Tan, 2018) untuk memisahkan kebisingan membutuhkan banyak biaya baik dari sumber daya komputasi maupun penggunaan *memory* (Tan, 2021). Baru-baru ini, model yang disebut *temporal convolutional network* (TCN) (Shaojie, 2018), yang menggunakan pembelajaran residual yang digabungkan dengan unit konvolusi yang melebar, dapat menangkap ketergantungan kontekstual jangka panjang dan mengungguli penelitian , (Jitong, 2017) yang mengusulkan penggunaan *recurrent neural network* (RNN) dengan empat *hidden long short term memory* (LSTM). Di tahun 2022 diusulkan metode *self-attending recurrentneural network*, atau *attentive recurrent network* (ARN) Penggunaan metode ARN ini juga membutuhkan tenaga komputasi yang besar dan juga *memory*  yang banyak. Penelitian kali ini mengusulkan kompresi pada metode ARN tanpa mengurangi kualitas secara signifikan dengan mengkompresi dan kuantisasi ARN.

## Rumusan Masalah

Mengacu pada temuan masalah pada latar belakang di atas maka berikut adalah rumusan masalah dalam penelitian ini adalah:

1. Bagaimana membuat model deep learning attentive recurrent neural network yang lebih efisien pada penggunaan resource dengan model compression dan quantization dalam mengurangi kebisingan dan meningkatkan kualitas suara manusia?

## Tujuan dan Manfaat

Tujuan penelitian ini adalah:

1. Mengembangkan sistem pengurangan kebisingan dan peningkatan kualitas suara manusia dengan penggunaan *deep learning attentive recurrent network* dengan *model compression* dan *quantization* agar dapat digunakan pada kasus nyata di kehidupan sehari-hari

Dengan diadakannya penelitian ini, manfaat-manfaat yang didapatkan dijabarkan sebagai berikut:

1. Memberikan rancangan dan arsitektur model untuk pengurangan kebisingan dan peningkatan kualitas suara manusia yang dapat diimplementasikan dikehidupan sehari hari.
2. Menambah kejelasan suara manusia ketika pertemuan menggunakan *video conference* walaupun ada kebisingan suara.

## Ruang Lingkup

Ruang lingkup penelitian ini adalah:

1. Sumber suara berasar dari *monaural* (satu*-*mikropon)
2. Suara kebisingan dibuat dengan menggabungkan suara manusia dengan sumber suara kebisingan
3. Dataset yang digunakan diambil dari Librispeech (Panayotov, 2015)

# LANDASAN TEORI

## Perkembangan *deep learning*

Penggunaan metode deep learning untuk menghilangkan kebisingan dan meningkatkan kejelasan suara manusia menggunakan *Convolutional Recurrent Neural Network* (CRN) (Ke Tan, 2018) mengungguli dua dasar nilai dari metode long short-term memory (LSTM) untuk pembicara terlatih dan tidak terlatih dalam hal skor short-time objective intelligibility (STOI) dan perceptual evaluation of speech quality (PESQ).

Dilanjut dengan pendekatan *Minimum Mean-Square Error* (MMSE) (Aaron, 2019) yang memanfaatkan *deep learning* mampu menghasilkan ucapan yang disempurnakan yang mencapai skor kualitas dan kejelasan yang lebih tinggi daripada pendekatan pembelajaran mendalam berbasis masking untuk sumber kebisingan non-stasioner di dunia nyata.

Pada tahun 2020 (Hu, 2020) [Yanxin Hu](https://arxiv.org/search/eess?searchtype=author&query=Hu%2C+Y) mengembangkan *Deep complex convolution recurrent network*, (DCCRN). Model DCCRN menggunakan jaringan kompleks untuk pemodelan spektrum bernilai kompleks. Dengan batasan aturan perkalian yang kompleks, DCCRN dapat mencapai kinerja yang lebih baik daripada yang lain (LSTM, CRN) dalam hal PESQ dan  *Mean Opinion Score* (MOS).

Di tahun yang sama (Li, 2020) [Jingdong Li](https://arxiv.org/search/cs?searchtype=author&query=Li%2C+J) mengusulkan *Temporal Convolutional Recurrent Neural Networks* (TCRN) secara signifikan mengungguli baseline LSTM dengan margin yang besar. TCRN secara konsisten lebih baik daripada CRN. TCRN memiliki kemampuan generalisasi kebisingan yang lebih baik daripada *baseline*.

Tahun 2021 Qiquan Zhang (Zhang, 2021) mengusulkan *adaptive attention module* (FAA) untuk meningkatkan model *Temporal Convolutional Network* (TCN) untuk tugas peningkatan suara monaural. Modul FAA terintegrasi ke dalam TCN dan GaTCN untuk melakukan peningkatan suara. Hasil metrik objektif menunjukkan bahwa TCN-FAA dan GaTCN-FAA yang diusulkan secara konsisten mengungguli model dasarnya tanpa modul FAA.

Di tahun yang sama 2021 Ashutosh (Pandey, 2021) mengusulkan *Dense convolutional network* (DCN) yang dilatih pada korpus WSJ0 SI-84 dataset dan dievaluasi pada pembicara WSJ yang tidak terlatih. Peningkatan kemampuan bicara berbasis DNN gagal digeneralisasi menjadi tidak terlatih corpora, dan kinerja yang lebih baik pada corpus yang terlatih tidak tentu mengarah pada kinerja yang lebih baik pada corpora yang tidak terlatih.

Ashutosh (Pandey, 2021) juga mengusulkan *dual-path self-attention recurrent neural network* (DPSARNN)yang diusulkan menambah RNN di DP-RNN dengan perhatian. Menambahkan perhatian di DP-RNN mengarah pada peningkatan yang lebih baik dengan empat kali pergeseran bingkai, menghasilkan model latensi rendah. Sebagai hasilnya, mengembangkan versi real-time dari DP-SARNN yang tidak hanya lebih cepat dari tetapi juga mengungguli pendekatan yang ada.

Tahun 2022 Ruizhe Cao(Cao, 2022) mengusulkan *conformer-based metric generative adversarial network (CMGAN).*  Hasil eksperimen menunjukkan bahwa metode yang diusulkan mengungguli *state of art* saat ini pada set data Voice Bank+DEMAND dengan parameter yang relatif sedikit (1,83 M).

Ashutosh pada tahun 2022 (Pandey, 2022) mengusulkan *self-attending recurrent neural network,* atau *attentive recurrent network (ARN),* Hasil percobaan telah menunjukkan keunggulan ARN atas kompetitif algoritma, seperti RNN, DCCRN, DCN dan DPARN. Arah penelitian selanjutnya akan mengoptimalkan ARN untuk aplikasi dunia nyata dengan menggunakan teknik seperti kompresi model dan kuantisasi. Arah penelitian terkait adalah untuk mengeksplorasi arsitektur DNN yang memiliki jumlah parameter lebih sedikit tetapi memberikan generalisasi sinyal.

Rangkuman dari perkembangan model *deep learning* untuk masalah pengurangan kebisingan dan peningkatan suara manusia (*speech enhancement)* ditunjukan pada Tabel 2.1.

|  |  |  |  |
| --- | --- | --- | --- |
| Tahun | Penulis | Metode | Hasil |
| 2018 | Ke Tan, DeLiang Wang | *Convolutional Recurrent Neural Network* (CRN) | Hasil evaluasi menunjukkan bahwa *Convolutional Recurrent Neural Network* yang diusulkan secara konsisten mengungguli dua dasar nilai dari metode *long short-term memory* (LSTM) untuk pembicara terlatih dan tidak terlatih dalam hal skor STOI dan PESQ. |
| 2019 | Aaron Nicolson, Kuldip K. Paliwal | Metode *Deep learning* untuk *Minimum Mean-Square Error* (MMSE) | MMSE yang memanfaatkan *deep learning* mampu menghasilkan ucapan yang disempurnakan yang mencapai skor kualitas dan kejelasan yang lebih tinggi daripada pendekatan pembelajaran mendalam berbasis masking dan pemetaan baru-baru ini, untuk sumber kebisingan non-stasioner di dunia nyata. |
| 2020 | [Yanxin Hu](https://arxiv.org/search/eess?searchtype=author&query=Hu%2C+Y), [Yun Liu](https://arxiv.org/search/eess?searchtype=author&query=Liu%2C+Y), [Shubo Lv](https://arxiv.org/search/eess?searchtype=author&query=Lv%2C+S), [Mengtao Xing](https://arxiv.org/search/eess?searchtype=author&query=Xing%2C+M), [Shimin Zhang](https://arxiv.org/search/eess?searchtype=author&query=Zhang%2C+S), [Yihui Fu](https://arxiv.org/search/eess?searchtype=author&query=Fu%2C+Y), [Jian Wu](https://arxiv.org/search/eess?searchtype=author&query=Wu%2C+J), [Bihong Zhang](https://arxiv.org/search/eess?searchtype=author&query=Zhang%2C+B), [Lei Xie](https://arxiv.org/search/eess?searchtype=author&query=Xie%2C+L) | *Deep complex convolution recurrent network* (DCCRN) | Dalam penelitian ini, (DCCRN). Model DCCRN menggunakan jaringan kompleks untuk pemodelan spektrum bernilai kompleks. Dengan batasan aturan perkalian yang kompleks, DCCRN dapat mencapai kinerja yang lebih baik daripada yang lain dalam hal PESQ dan MOS dalam konfigurasi parameter model yang serupa. |
| 2020 | [Jingdong Li](https://arxiv.org/search/cs?searchtype=author&query=Li%2C+J), [Hui Zhang](https://arxiv.org/search/cs?searchtype=author&query=Zhang%2C+H), [Xueliang Zhang](https://arxiv.org/search/cs?searchtype=author&query=Zhang%2C+X), [Changliang Li](https://arxiv.org/search/cs?searchtype=author&query=Li%2C+C) | *Temporal Convolutional Recurrent Neural Networks* (TCRN) | TCRN yang diusulkan secara signifikan mengungguli baseline LSTM dengan margin yang besar. Dan TCRN secara konsisten lebih baik daripada CRN. TCRN memiliki kemampuan generalisasi noise yang lebih baik daripada baseline. |
| 2021 | Qiquan Zhang , Qi Song , Aaron Nicolson, Tian Lan, Haizhou Li | *Temporal Convolutional Network with Frequency Dimension Adaptive* | *adaptive at- tention module* (FAA) untuk meningkatkan model TCN untuk tugas peningkatan suara monaural. Modul FAA terintegrasi ke dalam TCN dan GaTCN untuk melakukan peningkatan suara. Hasil metrik objektif menunjukkan bahwa TCN-FAA dan GaTCN-FAA yang diusulkan secara konsisten mengungguli model dasarnya tanpa modul FAA. Lebih menjanjikan, itu juga menunjukkan kemampuan generalisasi yang sangat baik untuk lingkungan yang bising nyata. |
| 2021 | Ashutosh Pandey, DeLiang Wang | *Dense convolutional network (DCN) with self-attention for speech enhancement in the time domain.* | DCN dilatih pada korpus WSJ0 SI-84 dataset dan dievaluasi pada pembicara WSJ yang tidak terlatih. Peningkatan kemampuan bicara berbasis DNN gagal digeneralisasi menjadi tidak terlatih  corpora, dan kinerja yang lebih baik pada corpus yang terlatih tidak tentu mengarah pada kinerja yang lebih baik pada corpora yang tidak terlatih |
| 2021 | Ashutosh Pandey, DeLiang Wang | *dual-path self-attention recurrent neural network (DP-SARNN) for time-domain speech enhancement* | . DPSARNN yang diusulkan menambah RNN di DP-RNN dengan perhatian. Menambahkan perhatian di DP-RNN mengarah pada peningkatan yang lebih baik dengan empat kali pergeseran bingkai, menghasilkan model latensi rendah. Sebagai hasilnya, mengembangkan versi real-time dari DP-SARNN yang tidak hanya lebih cepat dari tetapi juga mengungguli pendekatan yang ada. |
| 2022 | Ruizhe Cao, Sherif Abdulatif , Bin Yang | *conformer-based metric generative adversarial network (CMGAN) for Speech Enhancement in the time-frequency (TF) domain* | Hasil eksperimen menunjukkan bahwa metode yang diusulkan mengungguli *state of art* saat ini pada set data Voice Bank+DEMAND dengan parameter yang relatif sedikit (1,83 M). |
| 2022 | Ashutosh Pandey; Deliang Wang | *self-attending recurrent neural network, or attentive recurrent network (ARN), for timedomain speech enhancement to improve cross-corpus generalization.* | Hasil percobaan telah menunjukkan keunggulan ARN atas kompetitif algoritma, seperti RNN, DCCRN, DCN dan DPARN. |

Tabel ‎2.1 Perkembangan model *deep learning speech enchancement*

Dari perkembangan deep learning sebagaimana terlampir pada Tabel 2.1 metode ARN perlu dikembangkan lagi karena metode ini tidak mempertimbangkan parameter efisiensi dan kompleksitas komputasi pada model. Maka dari itu, pada penelitian ini akan fokus pada arsitektur ARN untuk menurunkan kompleksitas komputasi dan parameter efisiensi.

## *Digital Signal Processing*

*Digital signal processing* adalah perhitungan, algoritma, dan teknik yang digunakan untuk memanipulasi sinyal yang didapatkan dari data sensor setelah diubah ke dalam bentuk digital (Smith, 1999). Sinyal yang didapatkan umumnya berasal dari data sensor pada kehidupan nyata, seperti getaran seismik, citra visual, gelombang suara, dan lainlain. Digital signal processing banyak digunakan pada berbagai bidang, salah satunya pada bidang *speech recognition.* Suara yang didapatkan melalui input analog akan disaring dan diubah menjadi *digital audio* pada proses *analog-to-digital conversion* (ADC). Suara yang telah diubah ke dalam bentuk digital akan diproses pada *digital signal processing* (DSP) dan dilanjutkan dengan mengubah suara bentuk digital tersebut ke dalam bentuk analog pada proses *digital-to-analog conversion* (DAC). Tidak semua implementasi *digital signal processing* melewati proses *analog-to-digital conversion* dan *digital-to-analog conversion. Speech recognition* tidak membutuhkan proses *digital-to-analog conversion* karena aplikasi cukup mengenali suara yang didapatkan dan mengubahnya menjadi teks*. Text-to-speech* membutuhkan proses *digital-to-analog conversion* untuk dapat mengucapkan kata-kata tersebut melalui alat output yang telah disediakan dan tidak membutuhkan *analog-to-digital conversion* untuk menerima input dari pengguna.

## *Machine Learning*

Dalam Kecerdasan Buatan, ada konsep di mana mesin mulai belajar dari data tanpa diprogram. Konsep ini disebut *Machine Learning.* *Machine Learning* dapat didefinisikan secara luas sebagai metode komputasi yang menggunakan pengalaman untuk meningkatkan kinerja atau membuat prediksi yang akurat (Mohri dkk., 2013). Ada banyak aplikasi pembelajaran mesin, seperti pengenalan gambar, pengenalan ucapan, pengenalan emosi, prediksi stok, dan banyak lagi. Pembelajaran mesin, berdasarkan teknik, dapat dibagi lagi menjadi dua kategori, yaitu pembelajaran *supervised* dan pembelajaran *unsupervised*.

Pembelajaran *supervised* membangun model berdasarkan data dan label yang ada dan menggunakannya untuk memprediksi label untuk input baru, sedangkan Pembelajaran *unsupervised* digunakan untuk menemukan pola tersembunyi dalam kumpulan data. Aplikasi *supervised* adalah klasifikasi dan regresi, di mana keduanya memprediksi sesuatu, sedangkan salah satu aplikasi *unsupervised* yang umum adalah pengelompokan, di mana kami menemukan pola/kelompok tertentu dalam kumpulan data.

## *Deep Learning*

*Deep Learning* memungkinkan model komputasi yang terdiri dari beberapa lapisan pemrosesan untuk mempelajari representasi data dengan berbagai tingkat abstraksi (Lecun dkk., 2015). Ada perbedaan antara Deep Learning dan Machine Learning, seperti jumlah lapisan dan keberadaan fitur. Machine Learning membuat model berdasarkan data yang diawasi sedangkan Deep Learning menyusun jaringan untuk meniru otak yang dapat mempelajari pola dan mengekstrak fitur dari sejumlah besar data untuk membuat model yang optimal.

Deep Learning menggabungkan kemajuan dalam daya komputasi dan jaringan saraf dengan banyak lapisan untuk mempelajari pola rumit dalam data dalam jumlah besar. Ini adalah perpanjangan dari jaringan saraf klasik dan menggunakan lebih banyak lapisan tersembunyi sehingga algoritma dapat menangani data yang kompleks dengan berbagai struktur (Muniasamy & Alasiry, 2020).

## Optimizer algorithm

Algoritma pengoptimal digunakan oleh jaringan saraf untuk mengontrol perubahan bobot dan kecepatan pembelajaran. Ada beberapa algoritma pengoptimal, seperti Gradient Descent, Momentum, Adam *(Adaptive Moment Estimation*), dll.

### Gradient Descent

Gradient Descent adalah algoritma optimasi yang paling dasar dan paling banyak digunakan. Algoritma Gradient Descent menghitung arah bobot berdasarkan error untuk mendapatkan fungsi biaya mencapai nilai maksimum/minimum. Gradient Descent memiliki beberapa variasi, seperti Batch Gradient Descent, Mini Batch Gradient Descent, dan Stochastic Gradient Descent.

#### Batch Gradient Descent

Batch Gradient Descent disebut vanilla gradient descent. Batch Gradient Descent memperbarui parameter model setelah semua data pelatihan dievaluasi. Batch Gradient Descent memiliki keunggulan dalam stable error gradient dan stable convergence. Kerugian dari Batch Gradient Descent adalah kemungkinan besar terjebak pada konvergensi yang bukan yang terbaik yang dapat dicapai model.

|  |  |  |
| --- | --- | --- |
|  | 𝑤𝑡+1 = 𝑤𝑡 −𝑎.∇𝑤 𝐽(𝑤) | (2.1) |

Pada (2.1), nilai baru bobot (w) diperoleh dengan mengurangi bobot terakhir dengan nilai gradien, yang diperoleh dengan nilai turunan bobot dikalikan dengan learning rate.

#### Stochastic Gradient Descent

Stochastic Gradient Descent memperbarui parameter model untuk setiap contoh pelatihan. Keuntungan Stochastic Gradient Descent adalah biaya waktu konvergensi yang lebih sedikit dan memori yang lebih sedikit karena tidak perlu menyimpan nilai fungsi kerugian. Kerugiannya adalah perubahan parameter memiliki varians yang tinggi dan biaya komputasi yang tinggi.

|  |  |  |
| --- | --- | --- |
|  | 𝑤𝑡+1=𝑤 −𝑎.∇𝑤𝐽(𝑥𝑖,𝑦𝑖;𝑤) | (2.2) |

Pada (2.2), nilai bobot baru diperoleh dengan mengurangi bobot terakhir dengan turunan bobot pada setiap data dikalikan dengan learning rate.

#### Mini Batch Gradient Descent

Mini Batch Gradient Descent merupakan kombinasi dari keunggulan Stochastic Gradient Descent dan Batch Gradient Descent. Ini membagi data pelatihan menjadi batch yang berbeda dan memperbarui parameter model setelah setiap batch selesai.

|  |  |  |
| --- | --- | --- |
|  | 𝑤𝑡+1 = 𝑤𝑡 − 𝑎. ∇𝑤 𝐽(𝑥{𝑖:𝑖+𝑏}, 𝑦{𝑖:𝑖+𝑏}; 𝑤) | (2.3) |

Dalam (2.3), penurunan gradien mini-batch memperoleh bobot baru dari bobot terakhir dikurangi turunan bobot yang dihitung dengan kumpulan data saat ini dikalikan dengan kecepatan pembelajaran.

### Momentum

Momentum diciptakan untuk memperbaiki kelemahan Stochastic Gradient Descent dalam varian tinggi dan melunakkan konvergensi. Dalam persamaannya, variabel tambahan 'γ' disebut momentum saat memperbarui bobot. Dalam menerapkan momentum, ada variabel yang disebut kecepatan (V) yang menentukan kecepatan pertumbuhan berat.

|  |  |  |
| --- | --- | --- |
|  | 𝑉t+1 =𝛾∗𝑉t + 𝑎.∇w 𝐽(𝑥𝑖,𝑦𝑖;𝑤) | (2.4) |

|  |  |  |
| --- | --- | --- |
|  | 𝑤 𝑡+1 = 𝑤 t −𝑉 𝑡+1 | (2.5) |

Pada (2.4), nilai kecepatan diperoleh dengan nilai variabel momentum dikalikan dengan arah dan ditambah dengan turunan bobot pada setiap data. Pada (2.5) menunjukkan bobot baru adalah hasil bobot terakhir dikurangi nilai kecepatan.

### Adam

Algoritma Adam Optimizer adalah algoritma yang diusulkan ditujukan untuk optimasi stokastik efisien yang hanya membutuhkan gradien orde pertama dengan sedikit kebutuhan memori (Kingma & Ba, 2015). Ide di balik Adam adalah untuk mengurangi kecepatan sehingga tidak melewati nilai minimum/maksimum. Algoritma Adam Optimizer menghitung 2 Momen M(t) dan V(t) yang masing-masing disebut mean dan varians tidak terpusat. (2.6) menunjukkan rumus untuk mean dan (2.7) menunjukkan rumus untuk varians tidak terpusat.

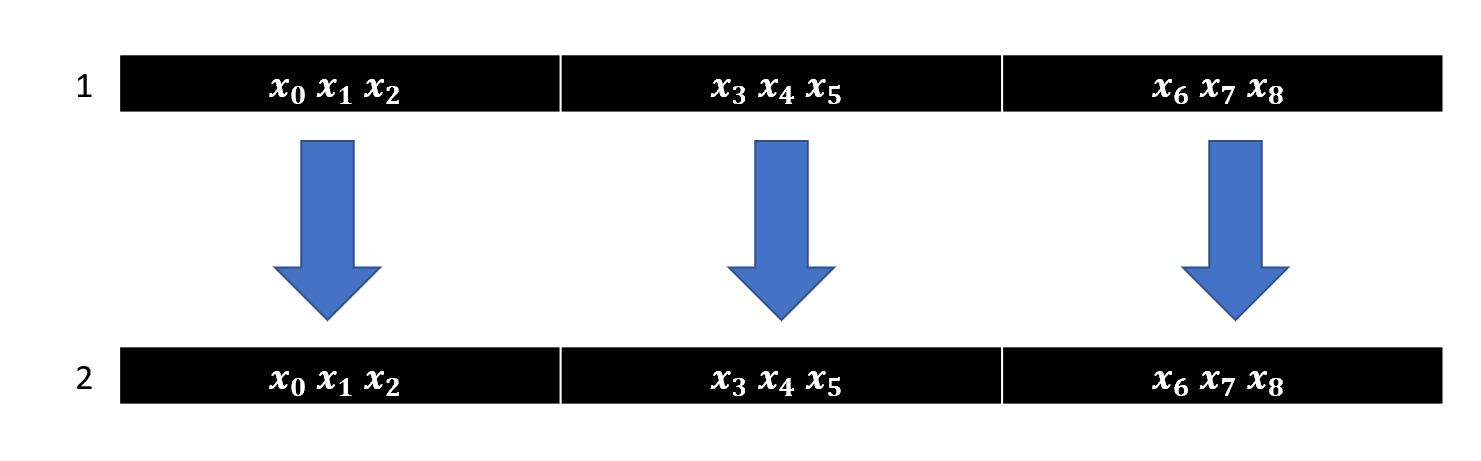
|  |  |  |
| --- | --- | --- |
|  | 𝑚𝑡 = 𝛽1 ∗ 𝑚𝑡−1 +(1−𝛽1)∗ ∇𝑤𝐽(𝑤) | (2.6) |
|  | 𝑣𝑡 = 𝛽2 ∗ 𝑣𝑡−1 +(1−𝛽2)∗ ∇𝑤𝐽(𝑤)2 | (2.7) |
|  | 𝑤𝑡=𝑤𝑡−1− 𝛼∗𝑚𝑡 √𝑣𝑡 + 𝜀 | (2.8) |

Dengan Adam, laju pembelajaran pengoptimal berubah selama pelatihan yang disebut peluruhan laju pembelajaran. Nilai peluruhan digunakan dalam mean dan varians tidak terpusat dengan simbol ''. Dengan fitur ini, menjadi banyak pilihan favorit selain SGD tradisional, yang mempertahankan tingkat pembelajaran tunggal untuk semua pembaruan bobot. Rumus untuk mengupdate bobot pada Adam dapat dilihat pada (2.8).

## **Signal Compression Signal Quantization**

### Kompresi Sinyal

Kompresi sinyal dapat didefinisikan sebagai proses untuk menyederhanakan suatu sinyal yang memiliki jumlah bit yang lebih banyak, menjadi sinyal dengan bit yang lebih sedikit. Proses kompresi suatu sinyal, dapat di ilustrasikan melalui Gambar ‎2.1.



Gambar 2.1 Mekanisme Kompresi

Kompresi sinyal pertama kali diterapkan pada saat terjadinya perubahan pada metode penyampaian informasi yang semula dalam bentuk analog menjadi digital. Selain itu, kompresi sinyal diterapkan untuk mengefisiensikan dan meningkatkan kualitas dari data yang dihasilkan dari piranti digital (Pearlman & Said, 2011).

Dalam rangkaian digital, proses kompresi dapat dianalogikan seperti rangkaian *encoder*. Rangkaian encoder dapat didefinisikan sebagai rangkaian yang memiliki jumlah masukan yang lebih banyak daripada keluaran. Rangkaian ini bertujuan untuk mengkonversi input dalam jumlah bit tertentu menjadi kode biner tertentu dengan jumlah yang lebih sedikit dibandingkan masukannya. Bentuk rangkaian dari rangkaian encoder dapat dijelaskan pada Gambar ‎2.2 (Darnila & Safwandi, 2020).

Diagram

Description automatically generated

Gambar 2.2 Rangkaian Enkoder (Darnila & Safwandi, 2020)

### Kuantisasi Sinyal

Kuantisasi sinyal dapat didefinisikan sebagai proses untuk mengubah sinyal kontinu x[n] menjadi sinyal diskrit xq[n] dengan menggunakan frekuensi pencuplikan tertentu. Nilai frekuensi pencuplikan harus lebih besar minimal 2 kali dari frekuensi sinyal yang hendak dicuplik untuk menghindari adanya *aliasing*. Kuantisasi sinyal dapat dilakukan dengan dua cara yaitu trunkasi dan pembulatan. Proses trunkasi merupakan proses kuantisasi dengan menghilangkan sejumlah angka tertentu dibelakang koma, sedangkan proses pembulatan merupakan proses kuantisasi dengan membulatkan sejumlah angka tertentu dibelakang koma yang lebih besar sama dengan 5. Proses kuantisasi dengan menggunakan pembulatan dan trunkasi dapat dijelaskan pada Tabel 2.2

|  |  |  |
| --- | --- | --- |
| Nilai Sinyal Analog | Metode Trunkasi | Metode Pembulatan |
| 0.37 | 0.30 | 0.40 |

Tabel ‎2.2 Perbedaan Metode Trunkasi dan Pembulatan

Proses kuantisasi dapat menghasilkan error dengan jumlah tertentu. Berikut adalah persamaan untuk menghitung error yang diakibatkan oleh kuantisasi pada sinyal.

|  |  |  |
| --- | --- | --- |
|  |  | (2.9) |

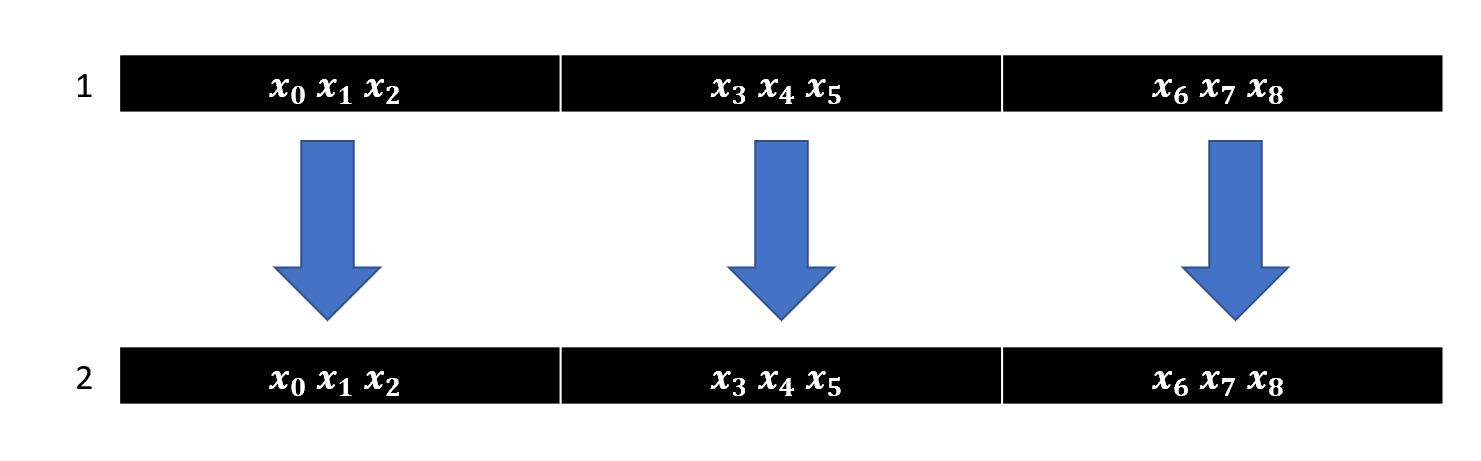
Untuk menghitung kualitas dari proses suatu kuantisasi sinyal, dapat digunakan parameter *signal to noise ratio*. Untuk menghitung parameter *signal to noise ratio*, tahap pertama yang harus dilakukan adalah mencari energi dari sinyal dan noise yang dapat dihitung dengan menggunakan (2.10) sebagai berikut.

|  |  |  |
| --- | --- | --- |
|  |  | (2.10) |

Tahap selanjutnya yang dilakukan adalah menghitung parameter signal to noise ratio dengan menggunakan (2.11) sebagai berikut.

|  |  |  |
| --- | --- | --- |
|  |  | (2.11) |

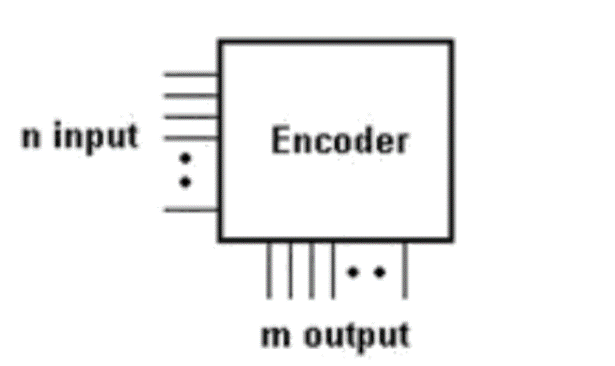
Kompresi sinyal dapat didefinisikan sebagai proses untuk menyederhanakan suatu sinyal yang memiliki jumlah bit yang lebih banyak, menjadi sinyal dengan bit yang lebih sedikit. Proses kompresi suatu sinyal, dapat di ilustrasikan melaluiGambar ‎2.3.



Gambar 2.3 Mekanisme Kompresi

Kompresi sinyal pertama kali diterapkan pada saat terjadinya perubahan pada metode penyampaian informasi yang semula dalam bentuk analog menjadi digital. Selain itu, kompresi sinyal diterapkan untuk mengefisiensikan dan meningkatkan kualitas dari data yang dihasilkan dari piranti digital (Pearlman & Said, 2011).

Dalam rangkaian digital, proses kompresi dapat dianalogikan seperti rangkaian *encoder*. Rangkaian encoder dapat didefinisikan sebagai rangkaian yang memiliki jumlah masukan yang lebih banyak daripada keluaran. Rangkaian ini bertujuan untuk mengkonversi input dalam jumlah bit tertentu menjadi kode biner tertentu dengan jumlah yang lebih sedikit dibandingkan masukannya. Bentuk rangkaian dari rangkaian encoder dapat dijelaskan pada Gambar ‎2.4 (Darnila & Safwandi, 2020).



Gambar 2.4 Rangkaian Enkoder(Darnila & Safwandi, 2020)

1. Kuantisasi Sinyal

Kuantisasi sinyal dapat didefinisikan sebagai proses untuk mengubah sinyal kontinu x[n] menjadi sinyal diskrit xq[n] dengan menggunakan frekuensi pencuplikan tertentu. Nilai frekuensi pencuplikan harus lebih besar minimal 2 kali dari frekuensi sinyal yang hendak dicuplik untuk menghindari adanya *aliasing*. Kuantisasi sinyal dapat dilakukan dengan dua cara yaitu trunkasi dan pembulatan. Proses trunkasi merupakan proses kuantisasi dengan menghilangkan sejumlah angka tertentu dibelakang koma, sedangkan proses pembulatan merupakan proses kuantisasi dengan membulatkan sejumlah angka tertentu dibelakang koma yang lebih besar sama dengan 5. Proses kuantisasi dengan menggunakan pembulatan dan trunkasi dapat dijelaskan pada tabel 1.

|  |  |  |
| --- | --- | --- |
| Nilai Sinyal Analog | Metode Trunkasi | Metode Pembulatan |
| 0.37 | 0.30 | 0.40 |

Tabel ‎2.3 Perbedaan Metode Trunkasi dan Pembulatan

Proses kuantisasi dapat menghasilkan error dengan jumlah tertentu. Berikut adalah persamaan untuk menghitung error yang diakibatkan oleh kuantisasi pada sinyal.

|  |  |  |
| --- | --- | --- |
|  |  | (2.12) |

Untuk menghitung kualitas dari proses suatu kuantisasi sinyal, dapat digunakan parameter *signal to noise ratio*. Untuk menghitung parameter *signal to noise ratio*, tahap pertama yang harus dilakukan adalah mencari energi dari sinyal dan noise yang dapat dihitung dengan menggunakan persamaan 2 sebagai berikut.

|  |  |  |
| --- | --- | --- |
|  |  | (2.13) |

Tahap selanjutnya yang dilakukan adalah menghitung parameter signal to noise ratio dengan menggunakan persamaan 3 sebagai berikut.

|  |  |  |
| --- | --- | --- |
|  |  | (2.14) |

# METODOLOGI

## Metodologi Penelitian

Pengurangan kebisingan dan meningkatkan kejelasan suara merupakan aspek penting yang dapat digunakan di banyak bidang. Teknologi ini dapat diterapkan di berbagai bidang pekerjaan misalnya pada *video call meeting*. Penelitian tentang Pengurangan kebisingan dan meningkatkan kejelasan suara telah ada sejak lama. Ada banyak model yang ada untuk mendeteksi kebisingan suara. Penelitian ini bertujuan untuk menghasilkan model yang lebih baik dari beberapa model yang ada dari segi efektifitas dan penggunaan ongkos komputasi.

Model yang diusulkan menggunakan *Deep learning* dengan kombinasi ekstraksi tingkat fitur dari dataset yang berbeda karena metode yang diusulkan memiliki kemungkinan untuk mengungguli kinerja dari model yang diusulkan sebelumnya . Garis besar penelitian dapat dilihat pada Gambar ‎3.1.

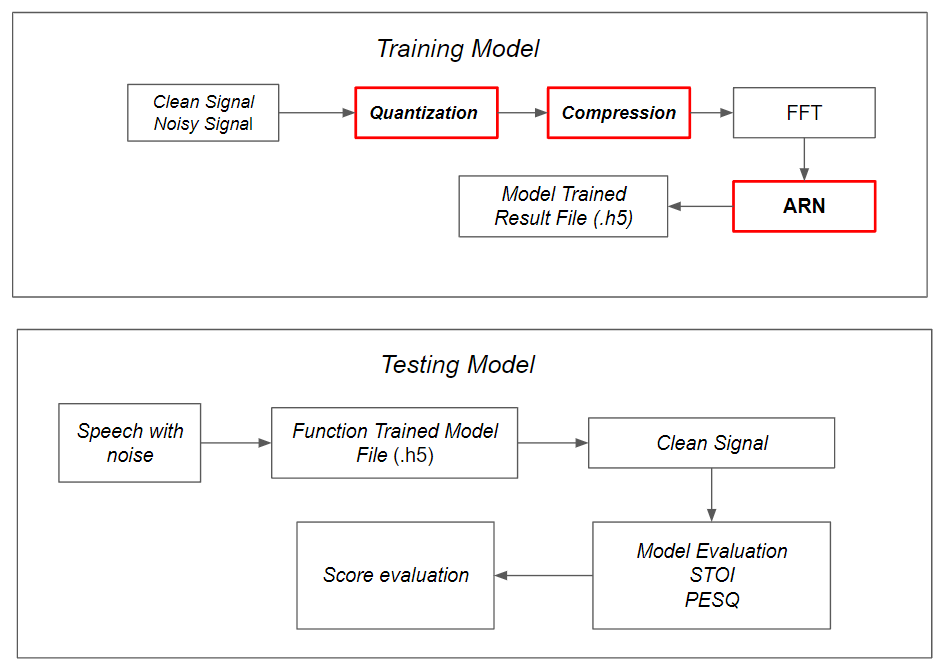
Diagram

Description automatically generated

Gambar 3.1 Metode Penelitian

Pada Gambar ‎3.1, penelitian diawali dengan dilakukannya identifikasi masalah, tujuan dari dilakukannya identifikasi masalah adalah mencari permasalahan yang akan dibahas dan diselesaikan pada penelitian ini. Setelah identifikasi masalah dilakukan, hal selanjutnya yang dilakukan adalah melakukan tinjauan literatur, tahap ini dilakukan untuk mencari literatur terkait untuk menyelesaikan masalah yang telah diidentifikasi pada identifikasi masalah. Setelah dilakukan tinjauan literatur, tahap selanjutnya yang dilakukan adalah pengumpulan data, tahap ini dilakukan dengan tujuan untuk mencari dataset yang digunakan untuk melatih model ARN untuk membedakan sinyal bersih dan sinyal *noise*. Setelah data dikumpulkan, tahap selanjutnya yang dilakukan adalah pra pemrosesan data, tahap ini dilakukan untuk memproses data yang telah dikumpulkan agar dapat digunakan untuk melatih model ARN. Setelah pra pemrosesan data dilakukan, dilakukan pemisahan data, tahap ini bertujuan untuk membagi data menjadi data latih dan data validasi saat pelatihan model dilakukan.

Setelah pemisahan data dilakukan, tahap selanjutnya adalah membuat model ARN, tahap ini bertujuan untuk membuat model yang akan memisahkan sinyal bersih dan sinyal noise. Pada proses pembuatan model, model akan dilatih dan dievaluasi akurasinya. Setelah model dibuat dan dilatih, model akan diuji untuk melakukan klasifikasi terhadap sinyal masukan yang berasal dari perangkat perekam suara, evaluasi dilakukan dengan menggunakan dua metrics evaluasi yaitu PESQ dan STOI. Setelah pengujian selesai, hal terakhir yang dilakukan adalah pembuatan laporan. Tahap ini dilakukan dengan tujuan untuk membahas hasil dari pembuatan model yang telah dilakukan.



Gambar 3.2 Diagram metode *speech enhancement*

Metode yang digunakan pada penelitian ini terlampir pada Gambar ‎3.2. Pada penelitian ini, terdapat dua tahap yang harus dilakukan, tahap pertama adalah tahap pelatihan model atau *training* dan tahap pengujian atau *testing.* Pada tahap pelatihan, hal pertama yang harus dilakukan adalah mengumpulkan dataset yang berisi sinyal pembicaraan bersih dan sinyal *noise*. Kemudian, dataset yang berisi sinyal akan dikuantisasi dengan tujuan untuk menurunkan resolusi dari sinyal yang digunakan untuk pelatihan agar proses pelatihan dapat berlangsung dengan lebih cepat. Setelah itu, sinyal yang telah dikuantisasi akan dikompresi dengan tujuan agar sinyal memiliki ukuran file yang lebih kecil. Lalu, sinyal yang telah dikompresi, ditransformasi menggunakan fast fourier transform, dengan tujuan untuk mengubah sinyal dalam domain waktu menjadi domain frekuensi. Setelah itu, dilakukan pelatihan terhadap model yang akan digunakan pada penelitian ini, tahap ini dilakukan dengan tujuan untuk melatih model agar dapat membedakan sinyal suara murni dan sinyal noise. Dan terakhir, hasil pelatihan model disimpan dalam bentuk dokumen dengan ekstensi .h5 agar dapat dijalankan pada perangkat keras lainnya.

Setelah model dilatih untuk membedakan sinyal suara dan noise, model akan diuji performanya untuk membedakan sinyal suara yang direkan dengan menggunakan perangkat microphone. Untuk menguji performa model, digunakan metode evaluasi yaitu STOI dan PESQ.

Penekanan pada penelitian ini, berada pada tiga tahap antara lain adalah *quantization*, *compresion*, dan arsitektur pada ARN.

## Pengumpulan dan Prapemrosesan Dataset

Kami mengevaluasi semua model dengan cara independen speaker, noise, dan corpus. Kami menggunakan semua ucapan dari set pelatihan korpus LibriSpeech (Panayotov, 2015) untuk menghasilkan campuran pelatihan. Ini terdiri dari sekitar 280 ribu ucapan pidato lebih dari 2000 pembicara. LibriSpeech telah terbukti menjadi korpus yang efektif untuk generalisasi lintas korpus karena direkam oleh banyak sukarelawan di seluruh dunia, dan karenanya terdiri dari ucapan-ucapan yang direkam dalam kondisi akustik yang berbeda. Ucapan pelatihan yang bising dihasilkan secara online selama pelatihan dengan cara berikut. Untuk setiap sampel dalam kelompok tertentu, kami secara acak mengambil sampel ucapan ucapan, mengekstrak potongan acak 4 detik darinya, dan menambahkan potongan acak kebisingan ke dalamnya dengan *signal to noise ratio* (SNR) acak dari {−5, 4, 3, 2, 1, 0} dB. Pidato sampel digunakan tidak berubah jika durasinya lebih kecil dari 4 detik. Satu set 224 suara non-suara manusia dari perpustakaan efek suara (docbox.etsi.org) digunakan sebagai suara pelatihan.

## Ekstraksi Fitur

Sinyal suara bising x didefinisikan sebagai jumlah dari sinyal suara bersih s dan sinyal suara n, digambarkan dengan Persamaan 3.1

|  |  |  |
| --- | --- | --- |
|  | x=s+n | (3.1) |

{x,s,n} RM×1, dan M adalah jumlah sampel dalam sinyal suara. Algoritme peningkatan suara manusia bertujuan untuk mendapatkan perkiraan yang mendekati, s, dari s yang diberikan x. Tujuan dari algoritma peningkatan ucapan domain waktu adalah untuk menghitung s langsung dari x daripada menggunakan representasi T-F dari x. Peningkatan suara manusia domain waktu menggunakan DNN dapat dirumuskan sebagai Persamaan 3.2

|  |  |  |
| --- | --- | --- |
|  | =fθ(x) | (3.2) |

di mana fθ menunjukkan fungsi yang diwakili oleh DNN diparametrisasi oleh θ

Secara umum, algoritma peningkatan suara dirancang untuk memproses bingkai sinyal suara. Diberikan sinyal noise x, pertama-tama dipotong menjadi bingkai yang tumpang tindih yang kemudian diproses pada tingkat bingkai oleh model peningkatan suara. Biarkan X RT ×L menunjukkan matriks yang berisi bingkai sinyal x dan xt RL×1 bingkai ke-t. xt didefinisikan sebagai Persamaan 3.3

|  |  |  |
| --- | --- | --- |
|  | xt[k]=x[(t−1)·J+k], k=0,···,L−1 | (3.3) |

di mana T adalah jumlah bingkai, L adalah panjang bingkai, dan J adalah pergeseran bingkai. T diberikan oleh M , di mana menunjukkan J fungsi langit-langit. x diisi dengan nol jika M tidak habis dibagi J. Pemrosesan bingkai menggunakan DNN dapat didefinisikan sebagai Persamaan 3.4

|  |  |  |
| --- | --- | --- |
|  | t = fθ(xt−T1,··· ,xt−1,xt,xt+1,··· ,xt+T2) | (3.4) |

di mana t dihitung menggunakan xt, T1 frame masa lalu, dan T2 masa depan bingkai.

Algoritma peningkatan ucapan tingkat bingkai dianggap kausal jika estimasi dari s bingkai yang diberikan dihitung menggunakan bingkai yang bising pada waktu yang kurang dari atau sama dengan t. Untuk peningkatan bicara kausal dimodifikasi sebagaimana digambarkan pada (3.5):

|  |  |  |
| --- | --- | --- |
|  | t =fθ(xt−T1,···,xt−1,xt) | (3.5) |

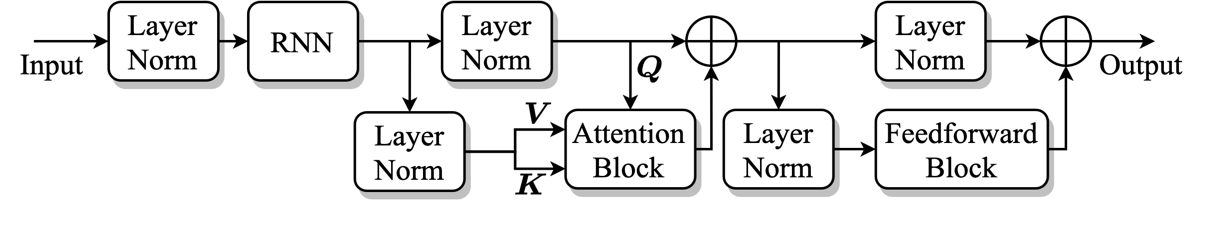
di mana t dihitung menggunakan xt dan T1 melewati frame. Kausalitas adalah persyaratan yang diperlukan untuk peningkatan kemampuan berbicara secara real-time. Lebih lanjut, kami mengamati bahwa algoritme kausal menunjukkan degradasi yang lebih besar pada korpora yang tidak terlatih dibandingkan dengan algoritme non-kausal yang sesuai. Oleh karena itu, kami juga mengembangkan dan membandingkan algoritma kausal.

## Pemisahan Data

Dataset yang digunakan oleh model dipisahkan menjadi set data splitting. Penelitian ini mengikuti metode tradisional dengan membagi dataset menjadi tujuan pelatihan dan pengujian. Langkah pertama dari pemisahan data adalah mengacak dataset untuk membuat data yang lebih merata. Setelah itu, kami membagi data latih dan menguji dengan rasio 8:2, 8 untuk data latih dan 2 untuk validasi untuk menghindari overfitting atau underfitting. Rasio 8:2 diterapkan untuk semua model yang diusulkan.

## Permodelan *Attentive Recurrent Network*

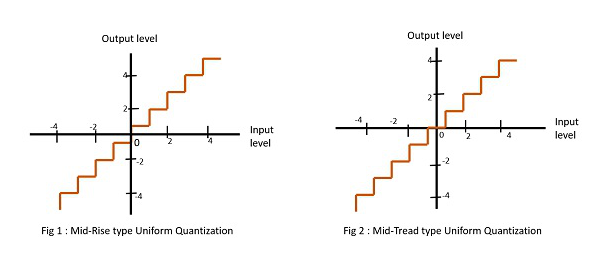
Metode yang diusulkan ialah penggunaan *quantization*  dan *compression* sebelum data masuk ke model *deep learning*. Model arsitektur *deep learning* yang ditunjukan pada Gambar ‎3.3 terdiri dari normalisasi *layer*, RNN, blok *self-attention*, dan blok *feedforward*. Selanjutnya, kami menjelaskan tiap tahapan satu per satu.



Gambar 3.3 Diagram ARN.

### *Quantization*

Penggunaan Kuantisasi diharapkan dapat mengurangi beban komputasi. Kuatisasi yang digunakan adalah kuantisasi *uniform.*  Jenis kuantisasi di mana tingkat kuantisasi diberi jarak yang seragam. Kuantisasi dapat ditunjukan pada Gambar 3.4



Gambar 3.4 *Quantization*

### *Compression*

Kompresi yang digunakan ialah *downsampling, i*denya adalah file audio terdiri dari data sampel secara periodik. Ukuran file audio meningkat karena jumlah sampel yang dikandungnya meningkat, jadi mengambil sedikit sampel akan mengubah ukuran file audio menjadi lebih kecil. Tipe kompresi audio yang akan menjadi eksperimen ialah mengubah .wav menjadi .mp3.

### *Normalization Layer*

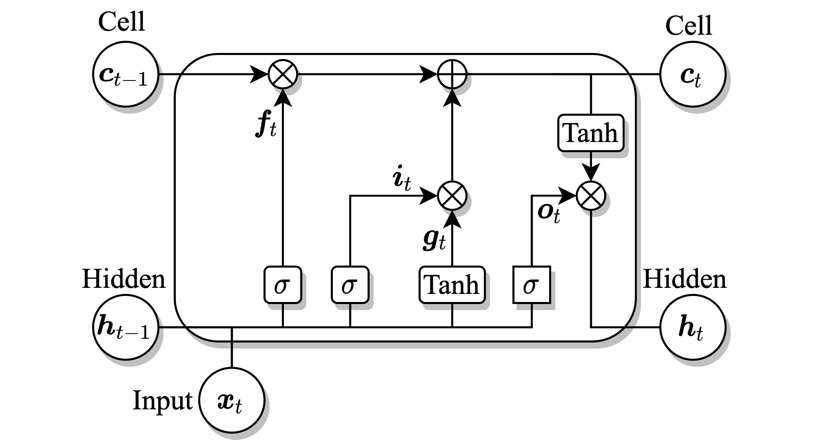
Penggunaan *Normalization Layer* digunakan dalam model untuk meningkatkan generalisasi dan memfasilitasi pelatihan yang lebih cepat (Ba, 2016). Hal ini diusulkan sebagai alternatif untuk normalisasi *batch* (Ioffe, 2015), yang ditemukan lebih sensitif terhadap pelatihan *batch* sebagaimana digambarkan di Persamaan 3.6

|  |  |  |
| --- | --- | --- |
|  | Diagram  Description automatically generated with medium confidence | (3.6) |

dimana xt dan , berturut-turut adalah mean dan varians dari . Simbol dan adalah parameter yang dapat dilatih dengan ukuran yang sama dengan xt, ⊙ menunjukkan perkalian elemen, dan adalah konstanta positif kecil yang digunakan untuk menghindari pembagian dengan nol. Menunjukkan perkalian elemen, dan adalah konstanta positif kecil yang digunakan untuk menghindari pembagian dengan nol.

### *Recurrent Neural Network*

Kami menggunakan *long short-term memory* (LSTM) RNN di ARN. Diagram ilustrasi dari LSTM ditunjukkan pada Gambar 3.5.



Gambar 3.5 Diagram ilustrasi dari LSTM

Diberikan urutan vektor input {x1,··· ,xt−1,xt,xt+1,··· ,xT}, keadaan tersembunyi pada waktu t, ht ,dihitung sebagai Persamaan 3.7

|  |  |  |
| --- | --- | --- |
|  | Text  Description automatically generated | (3.7) |

Di mana xt, gt, dan ct masing-masing mewakili input, input blok, dan status memori (sel) pada waktu t. Selain itu, ft, dan ot masing-masing adalah gerbang yang dikenal sebagai gerbang input, gerbang lupa dan gerbang keluaran. W dan b menunjukkan bobot dan bias yang dapat dilatih.

### *Self-attention Block*

Penggunaan Mekanisme *self attention* didefinisikan menggunakan tiga komponen: kunci K RT×R, nilai V RT×S, dan kueri Q RT×R. Pertama, skor korelasi antara pasangan baris dari Q dan K, {Qi,Kj}, dimana i,j {1,···,T}, dihitung dengan menggunakan Persamaan 3.8

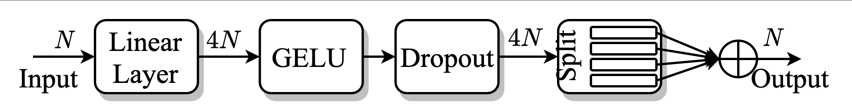
|  |  |  |
| --- | --- | --- |
|  | W = QKT | (3.8) |

Output akhir dari *self attention block* dihitung dengan Persamaan 3.9 dengan lapisan *softmax*  yang di representasikan dengan Persamaan 3.10

|  |  |  |
| --- | --- | --- |
|  | Text  Description automatically generated | (3.9) |
|  | A = Softmax(W ′)V | (3.10) |

### *Feedforward Block*

Blok feedforward di ARN ditunjukkan pada Gambar 3.6 Sebuah input yang diberikan dari ukuran N diproyeksikan ke ukuran 4N menggunakan lapisan linier, yang diikuti oleh *Gaussian Error Linear Units* (GELU) dan lapisan *dropout*. Akhirnya, keluaran berukuran 4N dibagi menjadi empat vektor berukuran N, yang dijumlahkan untuk mendapatkan keluaran akhir.



Gambar 3.6 *Feedforward Block*

Dengan blok bangunan yang dijelaskan, kami sekarang menyajikan aliran pemrosesan ARN yang ditunjukkan pada Gambar 3.5 . Input ke ARN pertama-tama dinormalisasi dan kemudian diproses menggunakan RNN. Output dari RNN dinormalisasi menggunakan dua normalisasi lapisan paralel. Aliran pertama digunakan sebagai Q dan aliran kedua digunakan sebagai K dan V untuk blok perhatian berikut. Output dari blok perhatian ditambahkan ke Q untuk membentuk koneksi residual. Sekali lagi, output dinormalisasi menggunakan dua normalisasi lapisan paralel. Aliran pertama diproses menggunakan blok *feed forward* dan aliran kedua ditambahkan ke *output blok feed forward* untuk membentuk sambungan sisa.

Pada penelitian ini, model akan dibuat pada saat tahap pemodelan. Setelah model dibuat, model akan disimpan dalam bentuk file dengan ekstensi “.h5” dengan tujuan agar dapat dijalankan sebagaimana tertera pada gambar 3.2 pada tahap testing.

## Evaluasi Performa

Kami menggunakan *short-time objective intelligibility* (STOI) dan *perceptual evaluation of speech quality* (PESQ) sebagai metrik evaluasi untuk membandingkan model yang dilatih di Librispeech. STOI memiliki kisaran tipikal [0, 1], yang secara kasar mewakili persen benar. PESQ memiliki rentang [−0.5,4.5], di mana skor yang lebih tinggi menunjukkan kualitas ucapan yang lebih baik. Kedua metrik tersebut biasanya digunakan untuk mengevaluasi algoritma peningkatan bicara

### *Short-time objective intelligibility* (STOI)

STOI dapat didefinisikan sebagai metode pengukuran korelasi antara 1/3 pita oktaf dan 384 ms blok yang mengandung spektrum sinyal bersih dan sinyal yang telah diproses. Sebelum dilakukan perhitungan, sinyal yang hendak diproses dipaketkan dan dipotong dengan menggunakan persamaan pada Persamaan 3.11.

|  |  |  |
| --- | --- | --- |
|  | Text, logo  Description automatically generated | (3.11) |

x = Menunjukan *envelope vector* suara bersih (dihimpun pada 384 ms)

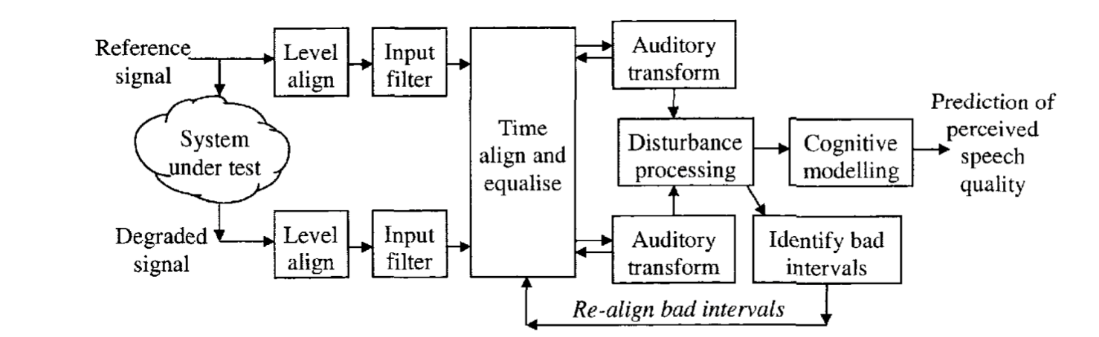
y = Menunjukan *envelope vector* yang sudah diproses

||.|| = Menunjukan 2 normalisasi dari *vector*

Pada penelitian ini, akan digunakan framework https://github.com/mpariente/pystoi untuk melakukan perhitungan STOI pada sinyal yang telah diproses (Mpariente 2022).

### *Perceptual evaluation of speech quality* (PESQ)

PESQ dapat didefinisikan sebagai model yang dikombinasikan sebagai audiomodel untuk mengevaluasi persepsi kualitas audio yang dijadikan rekomendasikan oleh ITU-R pada tahun 1999. Secara umum, PSEQ memiliki struktur sebagaimana tertera pada Gambar 3.7.



Gambar 3.7 Struktur PESQ Model (Rix, 2001)

Pada penelitian ini menggunakan framework python PESQ https://github.com/ludlows/PESQ untuk perhitungan PESQ pada evaluasi hasil sinyal yang bersih.

# DAFTAR PUSTAKA

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

Cao, R., Abdulatif, S., & Yang, B. (2022). CMGAN: Conformer-based Metric GAN for Speech Enhancement. arXiv preprint arXiv:2203.15149.

Darnila, E., & Safwandi, S. (2020). Dasar Digital (J. Simarmata, Ed.). Yayasan Kita Menulis.

Ephraim, Y., & Malah, D. (1984). Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *32*(6), 1109–1121. https://doi.org/10.1109/tassp.1984.1164453

Ephraim, Y., & Malah, D. (1985). Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 33(2), 443–445. https://doi.org/10.1109/tassp.1985.1164550*

Ephraim, Y. (1992). Statistical-model-based speech enhancement systems. Proceedings of the IEEE, 80(10), 1526-1555.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., & Pallett, D. S. (1993). DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon technical report n, 93, 27403.

J. Chen and D. Wang (2017). “Long short-term memory for speaker eneralization in supervised speech separation,” The Journal of the Acoustical Society of merica, vol. 141, no. 6, pp. 4705–4714.

Ke, Y., Li, A., Zheng, C., Peng, R., & Li, X. (2021). Low-complexity artificial noise suppression methods for deep learning-based speech enhancement algorithms. EURASIP Journal on Audio, Speech, and Music Processing, 2021(1), 1-15.

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436– 444. https://doi.org/10.1038/nature14539

Ludlows. (2022, July 8). *GitHub - ludlows/PESQ: PESQ (Perceptual Evaluation of Speech Quality) Wrapper for Python Users (narrow band and wide band)*. GitHub; github.com. https://github.com/ludlows/PESQ

Loizou, P. C. (2007). Speech enhancement: theory and practice. CRC press.

Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR.

Hu, Y., Liu, Y., Lv, S., Xing, M., Zhang, S., Fu, Y., ... & Xie, L. (2020). DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement. arXiv preprint arXiv:2008.00264.

Nicolson, A., & Paliwal, K. K. (2019). Deep learning for minimum mean-square error approaches to speech enhancement. Speech Communication, 111, 44–55. https://doi.org/10.1016/j.specom.2019.06.002

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2013). Foundations of Machine Learning second edition. Вестник Казнму, No3(1), с.30.

Mpariente. (2022, January 18). GitHub - mpariente/pystoi: Python implementation of the Short Term Objective Intelligibility measure. GitHub; github.com. https://github.com/mpariente/pystoi

Mustofa, A. (2018). Pengolahan Sinyal Digital. Universitas Brawijaya Press.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015, April). Librispeech: an asr corpus based on public domain audio books. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 5206-5210). IEEE.

Pandey, A., & Wang, D. (2020). Dual-path Self-Attention RNN for Real-Time Speech Enhancement. arXiv preprint arXiv:2010.12713.

Pandey, A., & Wang, D. (2021). Dense CNN with self-attention for time-domain speech enhancement. IEEE/ACM transactions on audio, speech, and language processing, 29, 1270-1279.

Pandey, A., & Wang, D. (2022). Self-attending RNN for speech enhancement to improve cross-corpus generalization. IEEE/ACM Transactions on Audio, Speech, and Language Processing.

Patil, P. V., & Mane, V. A. Speech Enhancement by using Ideal Binary Mask.

Paul, D. B., & Baker, J. (1992). The design for the Wall Street Journal-based CSR corpus. In Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992.

Purwins, H., Li, B., Virtanen, T., Schluter, J., Chang, S.-Y., & Sainath, T. (2019). Deep Learning for Audio Signal Processing. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 206–219. https://doi.org/10.1109/jstsp.2019.2908700

Pearlman, W. A., & Said, A. (2011). Digital Signal Compression: Principles and Practice. Cambridge University Press.

Scalart, P., & Filho, J. V. (n.d.). Speech enhancement based on a priori signal to noise estimation. *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. https://doi.org/10.1109/icassp.1996.543199

S. Bai, J. Z. Kolter, and V. Koltun.( 2018) “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” arXiv preprint arXiv:1803.01271.

Smith, S. W. (1999). Digital signal processing. California Technical Publishing. https://doi.org/10.1109/79.826412

Surendran, S., & Kumar, T. K. (2015). Perceptual subspace speech enhancement with variance normalization. Procedia Computer Science, 54, 818-828.

Rix, A. W., Beerends, J. G., Hollier, M. P., & Hekstra, A. P. (2001, May). Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In 2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221) (Vol. 2, pp. 749-752). IEEE.

Rothauser, E. H. (1969). IEEE recommended practice for speech quality measurements. IEEE Trans. on Audio and Electroacoustics, 17, 225-246.

Taal, C. H., Hendriks, R. C., Heusdens, R., & Jensen, J. (2010, March). A short-time objective intelligibility measure for time-frequency weighted noisy speech. In 2010 IEEE international conference on acoustics, speech and signal processing (pp. 4214-4217). IEEE.

Tan, K., & Wang, D. L. (2018). A convolutional recurrent neural network for real-time speech enhancement. *Interspeech 2018*. https://doi.org/10.21437/interspeech.2018-1405

Tan, K., & Wang, D. L. (2021). Compressing deep neural networks for efficient speech enhancement. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/icassp39728.2021.9413536

Upadhyay, N., & Karmakar, A. (2015). Speech enhancement using spectral subtraction-type algorithms: A comparison and simulation study. Procedia Computer Science, 54, 574-584.

Vaseghi, S. v. (2000). Advanced digital signal processing and noise reduction. John Wiley.

Zhang, Q., Wang, M., Lu, Y., Zhang, L., & Idrees, M. (2019). A novel fast nonstationary noise tracking approach based on MMSE spectral power estimator. *Digital Signal Processing*, *88*, 41–52. https://doi.org/10.1016/j.dsp.2019.01.019

Zhang, Q., Song, Q., Nicolson, A., Lan, T., & Li, H. (2021). Temporal Convolutional Network with Frequency Dimension Adaptive Attention for Speech Enhancement. Proc. Interspeech 2021, 166-170.