BAB 2
TINJAUAN PUSTAKA

2.1 Profil Tempat Penelitian

PT Kadence International atau Kadence Indonesia adalah sebuah perusahaan survei konsumen yang bergerak di bidang penelitian pemasaran sejak tahun 2007. PT Kadence International mengumpulkan data yang dibutuhkan oleh klien dengan tujuan menyajikan laporan yang baik dengan cepat, akurat, dan hemat biaya. PT Kadence International bekerja di berbagai sektor bisnis penelitian pemasaran, yaitu *Retail & FMCG*, *Food & Beverage*, *Consumer Electronics*, *Media*, *Travel & Tourism*, *Automotive*, *Financial Services*, *Brand & Advertising*, *Strategic Planning*, *NPD*, *NPS & C-Sat*, *Mystery Shopping*, *Sensory Testing*, *Pricing*, *Market Sizing*, dan *Market Entry*.

Kadence Indonesia berada di lantai 15 di dalam gedung Wisma Bakrie 2 dengan alamat Jalan HR. Rasuna Said Kav. B2, Jakarta Selatan 12920, Indonesia memiliki visi yaitu "*To raise the impact of research – to ensure that we deliver insight worth sharing*". [5]

Adapun panduan pilar dari Kadence Indonesia adalah:

1. *Always Passionate*

   Memiliki semangat untuk melakukan sesuatu dan sikap pantang menyerah.

2. *Fundamentally Dynamic*

   Berkembang dan mudah beradaptasi pada tantangan yang berbeda.

3. *Simply, Straight Talking*

   Memungkinkan klien mengambil keputusan dengan percaya diri.

4. *Essentially Wordly*

   Mempunyai tim yang mendunia, berbagi ilmu dan pengalaman, dan membuat hubungan yang luas.

5. *Impact Driven*

   Memberikan dampak yang menginspirasi pada bidang penelitian.

6. *Method Neutral*

   Memadukan penelitian kualitatif dan kuantitatif serta dapat berbicara kepada audience yang sulit dijangkau.

7. *People Focused*

   Memiliki tim yang sesuai dengan kemampuan di bidangnya.

2.1.1 Logo

Logo dari PT Kadence International ditunjukkan pada Gambar 2.1.



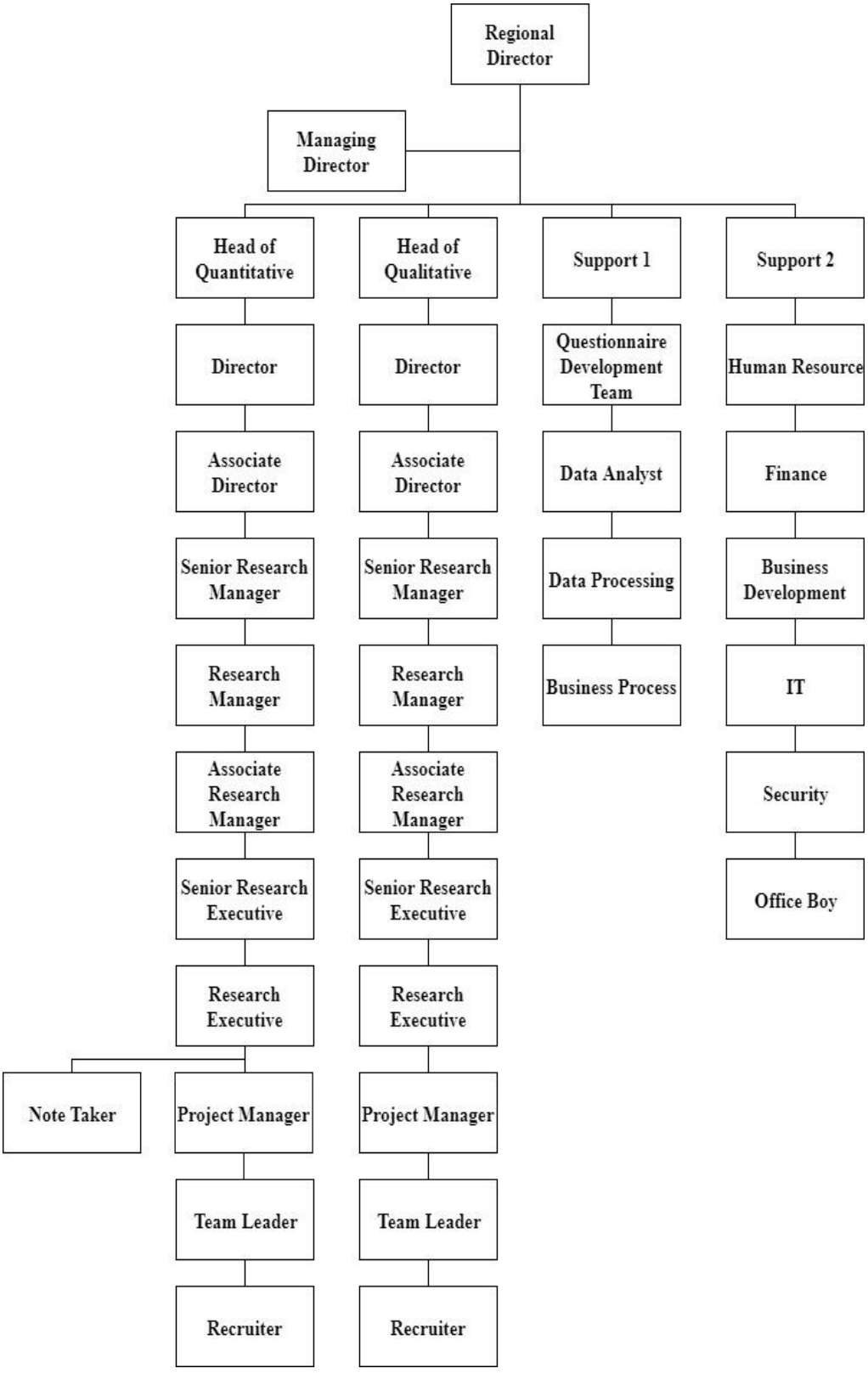
Gambar 2.1 Logo PT Kadence International

Keterangan logo:

1. Gambar 9 dot: menyimbolkan cara berpikir yang berbeda, tidak biasa, serta menggunakan perspektif yang baru dalam menggali informasi penelitian.
2. Tulisan warna hijau: menyimbolkan suasana yang santai, menyenangkan, dan menginspirasi.

2.1.2 Struktur Organisasi

Struktur organisasi dari Kadence Indonesia ditunjukkan pada Gambar 2.2.



Gambar 2.2 Struktur Organisasi Kadence Indonesia

2.2 Landasan Teori

Berikut adalah teori-teori yang digunakan mengenai sistem yang akan dibangun.

2.2.1 Wawancara

Wawancara adalah sebuah proses interaksi komunikasi yang dilakukan oleh pewawancara dan terwawancara atas dasar ketersediaan dengan mengacu kepada tujuan yang telah ditetapkan. Tujuan proses wawancara adalah memahami. Dalam proses memahami, diperlukan kemampuan agar wawancara dapat berjalan dengan baik. Kemampuan dalam proses memahami yaitu kemampuan memotivasi, kemampuan verbal, kemampuan mendengarkan, kemampuan nonverbal, kemampuan menarik kesimpulan, dan kemampuan menulis. Adapun persyaratan yang perlu dilakukan ketika hendak melakukan wawancara dalam konteks penelitian, antara lain: [6]

1. Peneliti perlu membuka jalan dan menemukan subjek yang akan diwawancarai.
2. Peneliti wajib menciptakan komunikasi yang kondusif terhadap subjek penelitian.
3. Peneliti perlu menjaga hubungan agar selalu stabil dan beriklim positif.
4. *Trust* adalah sesuatu yang mutlak dicapai.
5. Tetap menjaga jarak personal dan profesional dengan subjek penelitian.

2.2.1.1 Wawancara Terstruktur

Wawancara terstruktur digunakan dalam penelitian survei ataupun penelitian kuantitatif dan dapat dilakukan dengan penelitian kualitatif dalam beberapa situasi. Peneliti hanya mengajukan pertanyaan dan subjek penelitian hanya menjawab pertanyaan. Proses wawancara disesuaikan dengan perdoman wawancara (*guideline interview*) yang telah disiapkan. Contoh dari wawancara terstruktur adalah interogasi. [6]

2.2.1.2 Wawancara Semi Terstruktur

Wawancara semi terstruktur lebih tepat dilakukan pada penelitian kualitatif karena peneliti diberi kebebasan dalam bertanya dan dapat mengatur alur dan *setting* wawancara. Peneliti mengandalkan pedoman (*guideline*) wawancara untuk menggali data. Contoh dari wawancara semi terstruktur adalah *In-Depth Interview*. [6]

2.2.1.3 Wawancara Tidak Terstruktur

Wawancara tidak terstruktur tidak disarankan untuk sebuah penelitian kualitatif karena memiliki kelonggaran dalam banyak hal termasuk dalam hal pedoman wawancara. Contoh dari wawancara tidak terstruktur adalah seminar atau kuliah umum dan *talk-show.* [6]

2.2.2 Transkrip

Transkrip merupakan salinan atau catatan tiap kata dari suatu sumber untuk keperluan tertentu. Transkrip lebih dapat dipercaya dan dipertanggungjawabkan karena segala hal yang dibicarakan dalam wawancara sudah dialihkan menjadi bentuk skrip yang mudah untuk dianalisis. Bahasa skrip merupakan bahasa apa adanya tanpa adanya manipulasi atau penghalusan bahasa. Sumber didapatkan dari sebuah rekaman suara atau video dan isinya disalin dalam bentuk tulisan. [6]

2.2.3 Speech Recognition

Speech recognition adalah sebuah proses dan teknologi yang berkaitan untuk mengubah sebuah percakapan atau *speech* ke dalam bentuk kata-kata yang sesuai atau entitas melalui algoritma tertentu yang diimplementasikan di dalam perangkat atau komputer. Aplikasi *speech recognition* sudah termasuk *voice dialing, call routing, interactive voice response, data entry and dictation, voice command and control, gaming, structured document, appliance control by voice, computer-aided language learning, content-based spoken audio search,* dan *robotics*. [7]

Dalam beberapa tahun terakhir, teknologi *speech recognition* telah merubah cara kerja manusia dalam berinteraksi dengan komputer. Dong Yu dan Li Deng

mengklasifikasikan aplikasi *speech recognition* untuk membantu manusia berinteraksi dengan komputer, yaitu: [8]

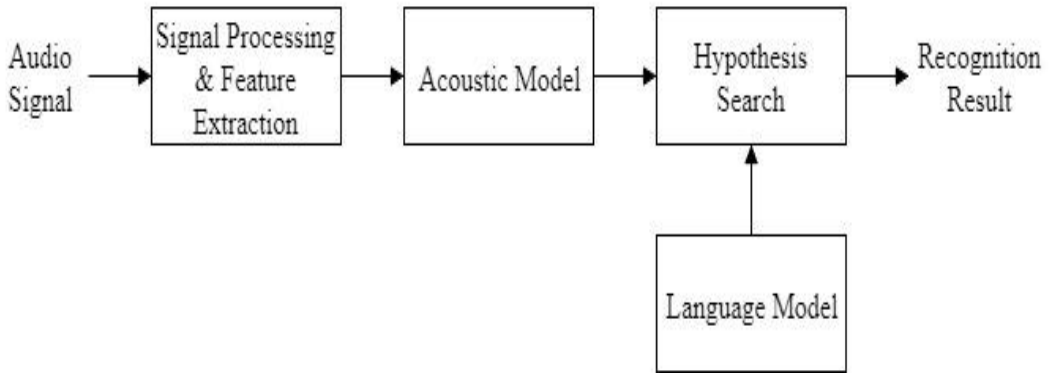
1. Human-Human Communication (HHC)

   Teknologi *speech* dapat menghapus pembatas dalam berinteraksi antara manusia dengan manusia. Contohnya orang yang tidak dapat berbahasa jepang akan sulit tinggal di Jepang, pembatas ini dapat dikurangi dengan menggunakan sistem penerjemah *speech-to-speech*. Teknologi ASR juga dapat digunakan untuk mendikte pesan singkat dan mengenali percakapan seorang guru agar murid mudah menemukan informasi yang dibutuhkan.

2. Human-Machine Communication (HMC)

   Teknologi *speech* berperan besar dalam perkembangan interaksi antara manusia dengan mesin. Aplikasi yang sering dikenali adalah *voice search*, *personal digital assistants* (PDA), *gaming*, dan *living room interaction systems*. *Voice search* mempermudah pengguna untuk mencari informasi menggunakan suara. *Personal digital assistance* (PDA) merupakan sistem yang dapat membaca informasi yang tersimpan di dalam perangkat, salah satunya adalah Siri dari Apple. Siri dapat menerima perintah yang dimasukkan melalui suara. Pengalaman dalam bermain *game* dapat bertambah ketika memakai teknologi *speech*. Teknologi *speech* juga dapat mengontrol perangkat yang ada di ruang tamu menggunakan suara.

Arsitektur *speech recognition* mempunyai empat komponen utama yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 Arsitektur *Speech Recognition*

(Sumber : *Automatic Speech Recognition; A Deep Learning Approach*, 2015)

Penjelasan dari tahapan proses dalam arsitektur *speech recognition* adalah sebagai berikut.

1. *Signal Processing & Feature Extraction*

   Komponen ini mengambil masukan sinyal suara lalu meningkatkan ucapan dengan menghilangkan gangguan dan distorsi, mengubah sinyalnya, dan kemudian mengekstraknya menjadi fitur yang dapat disesuaikan dengan model akustik pada tahap selanjutnya.

2. *Acoustic Model*

   Komponen ini mengintegrasikan pengetahuan, mengambilnya sebagai masukan fitur yang dihasilkan dari ekstraksi sebelumnya, dan kemudian menghasilkan skor AM untuk rangkaian fitur pada tahap selanjutnya.

3. *Hypothesis Search*

   Komponen ini memperkirakan probabilitas kemiripan data atau disebut skor LM dengan mempelajari korelasi tiap kata kunci.

4. *Language Model*

   Komponen ini menggabungkan skor AM dan skor LM dengan mengurutkan urutan kata sehingga menghasilkan urutan kata dengan skor tertinggi sebagai kata yang dikenali.

2.2.4 Noise

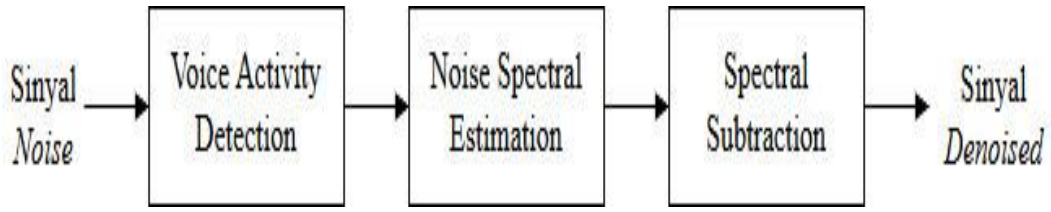
Noise dapat didefinisikan sebagai suara yang tidak diinginkan yang dapat mengganggu kenyamanan seseorang. *Noise* dapat berbentuk bisikan, suara kendaraan, ataupun suara di jalan. Tingkatan *noise* yang diproses menggunakan satuan desibel dengan jenis *noise* berupa *babble noise, car noise, white noise,* dan *street noise*. Tabel 2.1 menunjukkan tingkatan *noise* berdasarkan satuan desibel dari setiap jenis suara. [9]

Tabel 2.1 Tingkatan *Noise*

Desibel (dB)	Jenis Suara
0 – 10	Ambang batas pendengaran (1000 Hz)
10 – 20	Ruang belajar yang sunyi
20 – 30	Hembusan angin yang ringan
30 – 40	Bisikan seseorang dengan jarak sekitar 1.5 meter
40 – 50	Tempat tinggal di kota
50 – 60	Percakapan seseorang dengan jarak sekitar 1 meter
60 – 70	Percakapan di kelas
70 – 80	Kereta barang dengan jarak sekitar 30 meter
80 – 90	Mesin pabrik
90 – 100	Lokasi pembangunan
100 – 110	Konser music
110 – 120	Pengoperasian alat berat
120 – 130	Pesawat jet lepas landas dengan jarak sekitar 60 meter
130 – 140	Ambang batas rasa sakit di telinga

Noise dapat dikurangi dengan proses *noise reduction* ataupun *noise suppression*. *Noise suppression* merupakan proses pengambilan sinyal *noise* dan mengurangi *noise* sebanyak mungkin selama adanya suara yang berjalan. Penggunaan *noise suppression* dapat ditujukan pada aplikasi VoIP atau *video conference* dan aplikasi *speech recognition*. Implementasi pada s*peech recognition* dapat dilakukan secara *real-time* dengan cara menerima masukan suara melalui mikrofon dan kemudian menjalankan *noise suppression* sehingga mengeluarkan suara yang lebih mudah didengar dan dipahami (*denoised*). [10]

Konsep sederhana dari *noise suppression* ditunjukkan pada Gambar 2.4.



Gambar 2.4 Konsep Noise Suppression

(Sumber : *A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement*, 2017)

Penjelasan dari konsep *noise suppression* adalah sebagai berikut. [10]

1. Modul *Voice Activity Detection* (VAD) mendeteksi saat sinyal berisi suara dan jika hanya berisi *noise*.
2. Sinyal yang dideteksi VAD digunakan oleh modul *Noise Spectral Estimation* untuk mengetahui karakteristik dari *noise* yang ada seperti berapa banyak daya pada tiap frekuensi.
3. Jika modul *Voice Activity Detection* (VAD) tidak mendeteksi sinyal suara atau dalam keadaan diam, maka tahap estimasi pengenalan karakteristik *noise* menjadi opsional.
4. Dengan mengetahui bagaimana bentuk dari *noise* yang ada, modul *Spectral Subtraction* dapat mengurangi *noise* dari *input audio* untuk menghasilkan sinyal *denoised* atau sinyal suara yang memiliki *noise* rendah.

2.2.5 Web Speech API

Web Speech API merupakan spesifikasi *javascript web browser* yang mengizinkan pengembang untuk menggabungkan sistem *speech recognition* dan *speech synthesis* ke dalam sebuah *website*. Pengembang juga dapat menggunakan sekumpulan baris kode untuk merubah suara menjadi keluaran dalam bentuk teks (*speech-to-text*) dan menggunakan teknologi *speech recognition* sebagai masukan dari sebuah formulir, pendiktean, dan kontrol pengendalian melalui suara. Penggunaan Web Speech API harus dapat diizinkan oleh pengembang atau pengguna melalui *web browser* dimana b*rowser* memperlihatkan penggunaan API

dalam bentuk notifikasi. Saat ini Web Speech API mendukung 32 bahasa dimana salah satunya adalah Bahasa Indonesia.

Implementasi Web Speech API dapat bekerja dari sisi *server* dan sisi *client*, contoh penggunaannya adalah sebagai berikut. [11]

1. *Voice Web Search*

   Pengguna dapat mengatakan kata kunci untuk mendapatkan informasi.

2. *Speech Command Interface*

   Sistem dapat mengenali dan menjalankan perintah yang dikatakan oleh pengguna, seperti “telpon [nomor]“ atau “cari untuk [kata kunci]”.

3. *Continuous Recognition of Open Dialog*

   Sistem mengumpulkan isian sebuah formulir dari suara pengguna seperti mendiktekan isi *email*.

4. *Voice Activity Detection*

   Sistem mendeteksi aktivitas *speech* untuk mengembangkan pengalaman pengguna seperti “Tekan untuk bicara” atau “Percakapan bebas”.

5. *Speech Translation*

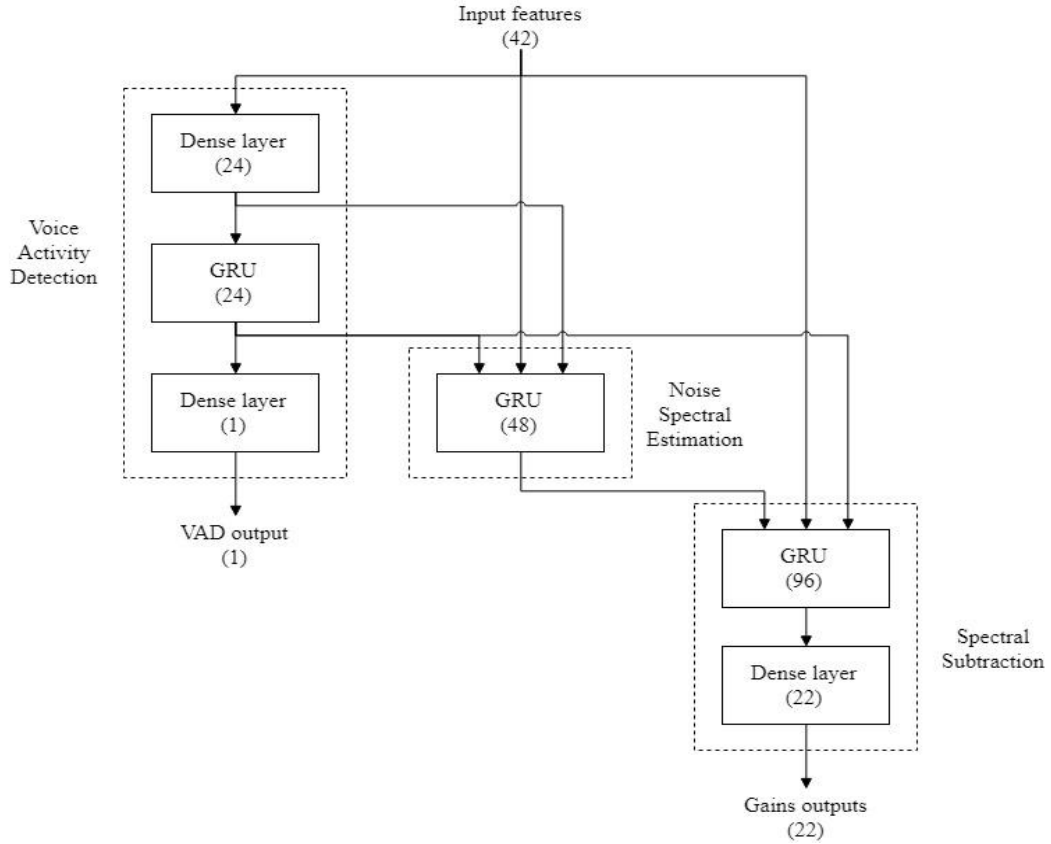
   Sistem dapat menerjemahkan antara dua bahasa dimana sistem mendengarkan masukan suara dalam bahasa pertama dan menghasilkan hasil terjemahan dalam bahasa kedua.

2.2.6 RNNoise

RNNoise merupakan suatu proyek yang mengimplementasikan *noise suppression* dengan menkombinasikan teknik DSP, *deep learning* dan *recurrent neural networks* (RNN) untuk mengurangi *noise*. Target utama dari proyek RNNoise adalah dalam bidang aplikasi *speech recognition* secara *real-time* seperti *video-conference* dengan kompleksitas yang rendah. Desain dan pelatihan data RNNoise dikembangkan dalam bahasa Python menggunakan *deep learning library*

Keras. Dengan tujuan membuat sistem yang *real-time*, RNNoise diimplementasikan pada bahasa C. RNNoise menyediakan *module javascript* yang dapat dijalankan melalui *browser* untuk dapat mengurangi *noise* suara yang direkam dari mikrofon secara *real-time*. [10]

Recurrent neural network (RNN) berperan penting di dalam RNNoise untuk memodelkan urutan waktu yang digunakan dalam proses *noise suppression* untuk memperkirakan *noise* dengan baik. RNNoise memiliki keterbatasan karena tidak dapat menyimpan informasi untuk waktu yang lama dan terjadi proses penurunan ketika proses *back-propagation* melewati waktu yang tidak efisien. RNNoise menggunakan *Gated Recurrent Unit* (GRU) yang dapat menyimpan informasi untuk jangka waktu yang lama. [10] Arsitektur dari RNNoise dapat dilihat pada Gambar 2.5.



Gambar 2.5 Arsitektur RNNoise

(Sumber : *A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement*, 2017)

2.2.7 Google Chrome

Google Chrome merupakan perangkat lunak yang digunakan untuk mencari, menerima, dan menampilkan sebuah konten dari *world wide web* seperti halaman web, gambar, video, dan file lainnya yang dikembangkan oleh Google Inc. Chrome pertama kali dirilis pada bulan September tahun 2008 yang dapat digunakan pada sistem operasi Windows, Mac, Linux, iOS, dan perangkat Android. Sejak pertama rilis, keunggulan yang ditunjukkan oleh Chrome adalah kecepatan. Chrome dirancang untuk memenuhi kebutuhan *browsing* secara cepat, aman, dan mudah digunakan.

Chrome menawarkan beberapa fitur yang menjadi keunggulannnya sebagai *web browser*, yaitu: [12]

1. Kecepatan

   Chrome dapat dijalankan dari komputer dalam hitungan detik. Proses pencarian sangat cepat dengan menggunakan *Omnibox* yang dapat menyarankan kata kunci pencarian dan menjalankan fungsi *auto-completion* dalam kolom *address bar*. Chrome dilengkapi dengan mesin *WebKit* yang dapat membuka halaman *web* atau aplikasi *web* dengan cepat.

2. Kemudahan

   Chrome dirancang dengan simpel agar pengguna dapat fokus dengan apa yang dikerjakannya.

3. Keamanan

   Chrome memiliki fitur keamanan *safe browsing* dimana ketika halaman *web* yang dikunjungi berisi konten yang tidak aman, Chrome akan memberi pesan peringatan untuk menghindari hal-hal yang tidak diinginkan.

4. Privasi

   Chrome menjaga privasi pengguna untuk membantu melindungi informasi yang kita bagikan ketika sedang terhubung dengan jaringan *internet*.

5. Pengembangan

   Chrome menyediakan alat atau *plugins* yang dapat digunakan untuk mengontrol *web browser* secara lebih luas.

6. Integrasi

   Dengan menghubungkan akun Google dengan Chrome, pengguna dapat mengontrol aktivitas yang dilakukan saat menggunakan Chrome.

2.2.8 Laragon

Laragon adalah aplikasi yang dapat merubah komputer menjadi sebuah server maupun lokal atau dapat disebut sistem *web stack* untuk pengembangan *web*. Laragon berada di lingkungan yang terikat dengan sistem operasi Windows dan menawarkan program-program pendukung untuk pengembangan *web* modern seperti Ruby on Rails, Laravel, Django, Flask, MEAN, dan Spring Boot serta penggunaan MySQL, PostgreSQL, MongoDB, Memcached, Redis, PHP, Ruby, Python, Node.js, dan Java yang mudah. Laragon dapat bekerja dengan baik untuk mengembangkan sebuah *web* dan memiliki kecepatan yang luar biasa. Laragon memiliki beberapa fitur unggulan seperti mendukung SSL, memungkinkan pengembang mengubah versi program pendukung dengan mudah, mengatur *database*, membuat proyek aplikasi dengan cepat, dan tampilan yang *user friendly*. [13]

2.2.9 PHP

PHP adalah bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web. PHP dapat dipanggil dari *web browser* dimana program yang ditulis dengan PHP akan di-*parsing* di dalam *web server* oleh *interpreter* PHP dan kemudian diterjemahkan ke dalam dokumen HTML. Selanjutnya konten halaman

web akan ditampilkan kembali ke *web browser*. PHP atau kepanjangannya adalah *PHP: Hypertext Preprocessor* berfokus pada skrip yang berjalan di sisi *server* sehingga kode PHP tidak akan terlihat pada *web browser*. [14]

Cara kerja pemrograman PHP adalah sebagai berikut. [14]

1. Pengguna menulis *www.toko.com/produk.php* ke dalam alamat di *web browser*.
2. *Web browser* (Google Chrome) mengirimkan pesan ke server *www.toko.com* melalui jaringan internet dan meminta halaman *web* dari *produk.php*.
3. *Web server* (Apache dalam Laragon) akan menangkap pesan tersebut dan meminta *interpreter* PHP untuk mencari *file produk.php* dalam media penyimpanan server.
4. *Interpreter* PHP membaca *file produk.php* dari media penyimpanan server
5. *Interpreter* PHP akan menjalankan kode PHP yang ada di dalam *produk.php*.
6. *Interpreter* PHP mengirimkan halaman *web* dalam bentuk HTML ke *web server*.
7. Dalam jaringan internet, *web server* akan mengirimkan halaman yang diperoleh dari *interpreter* PHP ke komputer pengguna sebagai respon permintaan yang diberikan.
8. *Web browser* yang digunakan pengguna akan menampilkan halaman *web* yang dikirim oleh *web server*.

Terdapat 3 area utama dimana skrip PHP digunakan, yaitu:

1. *Server-side scripting*

   Untuk menjalankan PHP pada sisi *server*, dibutuhkan PHP *parser*, *web server*, dan *web browser*. *Web server* yang sudah terhubung dengan PHP dijalankan dan hasil keluaran dapat dilihat pada *web browser*.

2. *Command line scripting*

   PHP dapat dijalankan tanpa *server* atau *browser*, hanya dengan menggunakan PHP *parser*. Skrip PHP pada *command line* biasanya digunakan untuk tugas pemrosesan teks yang sederhana.

3. *Writing desktop applications*

   PHP tidak berfokus pada pengembangan aplikasi di *desktop*, tetapi PHP memiliki fitur PHP-GTK untuk membuat program *client-side*.

2.2.10 BPMN (Business Process Model and Notation)

BPMN adalah suatu bagan yang menggambarkan urutan proses berdasarkan notasi untuk mendefinisikan proses bisnis. BPMN merupakan kesepakatan antara beberapa vendor pemodelan lainnya yang memiliki notasi mereka sendiri dengan menggunakan notasi tunggal untuk kepentingan pembelajaran dan pelatihan. BPMN menyediakan mekanisme untuk menghasilkan gambaran proses bisnis yang dapat dieksekusi dari notasi di tingkat bisnis. [15]

Terdapat tiga level dari proses *modeling* yang didukung oleh BPMN, antara lain: [15]

1. *Process Maps*

   Flow chart sederhana dari aktivitas.

2. *Process Descriptions*

   Flow chart yang menambahkan informasi tambahan.

3. *Process Models*

   Flow chart yang memiliki cukup informasi sehingga proses dapat dianalisis dan dieksekusi.

BPMN pada umumnya memiliki empat elemen diagram, yaitu: [15]

1. *Activities*

   Activity adalah pekerjaan yang dilakukan dalam proses bisnis, dapat dilakukan sekali atau berulang.

2. *Events*

   Event adalah sesuatu yang terjadi selama proses bisnis berlangsung. *Event* dapat mempengaruhi arus proses seperti memulai, menyela, atau mengakhiri arus proses.

3. *Gateways*

   Gateway adalah pemodelan elemen yang digunakan untuk mengontrol bagaimana urutan arus berinteraksi dengan proses.

4. *Connectors*

   Connector adalah penghubung antara *activity* atau entitas yang dilakukan di dalam proses bisnis.

2.2.11 DFD (Data Flow Diagram)

DFD merupakan pemodelan grafis sistem yang menunjukkan aktivitas pemrosesan atau fungsi sistem dan pertukaran data di antara fungsi-fungsi tersebut. DFD dimulai dengan serangkaian fungsi level teratas yang merepresentasikan sub fungsi yang dilakukan oleh fungsi tersebut menggunakan hierarki diagram. DFD didasarkan pada seperangkat konsep intuitif dan aturan yang sederhana agar mudah dipahami. Terdapat lima jenis simbol yang digunakan untuk membangun DFD, antara lain: [4]

1. *Process*

   Proses atau fungsi direpresentasikan menggunakan lingkaran untuk menggambarkan fungsi apa saja yang terdapat pada sistem.

2. *External Entity*

   Entitas eksternal direpresentasikan menggunakan kotak yang merupakan entitas fisik yang berada di luar sistem seperti pengguna sistem. Selain pengguna, simbol entitas eksternal dapat digunakan untuk mewakili perangkat keras dan perangkat lunak eksternal seperti perangkat lunak aplikasi lain yang akan berinteraksi dengan perangkat lunak yang dimodelkan.

3. *Data Flow*

   Arus data direpresentasikan menggunakan panah yang mewakili arus data yang terjadi antara dua proses atau antara entitas eksternal dan proses. Simbol arus data biasanya diberi anotasi dengan nama data yang sesuai.

4. *Data Store*

   Penyimpanan data direpresentasikan menggunakan dua garis paralel yang mewakili struktur data atau *file* fisik pada media penyimpanan. Setiap penyimpanan data terhubung ke sebuah proses dengan menggunakan simbol arus data. Panah yang mengalir masuk atau keluar dari penyimpanan data secara implisit mewakili keseluruhan data, oleh karena itu arus data yang terhubung ke penyimpanan data tidak perlu diberi anotasi dengan nama data yang sesuai.

5. *Output*

   Simbol keluaran digunakan ketika *file hard copy* dibuat.

Konsep penting yang terkait dengan DFD dibedakan menjadi dua, yaitu operasi *synchronous* and *asynchronous*. Jika dua proses saling terhubung langsung dengan panah arus data, maka bersifat *synchronous* karena beroperasi pada waktu yang berbarengan. Jika dua proses dihubungkan melalui penyimpanan data, maka operasi dilakukan independen atau *asynchronous*. Setiap model DFD pada sistem harus disertai dengan kamus data yang mencantumkan tujuan setiap data dan

definisinya. Model DFD dari sistem dibangun menggunakan hierarki DFD. Level teratas disebut level 0 DFD atau diagram konteks, untuk mengembangkan level teratas, dilakukan identifikasi proses dekomposisi menjadi sub proses. Tahap pertama pengembangan model aliran data dari suatu sistem yaitu merepresentasikan abstrak pada level teratas dari masalah yang harus dikerjakan, selanjutnya DFD tingkat bawah dikembangkan. Level 0 dan level 1 hanya terdiri dari satu DFD. Level 2 dapat berisi hingga tujuh DFD terpisah, dan level 3 sampai 49 DFD, dan seterusnya. Dalam pemodelan DFD hanya ada satu kamus data untuk keseluruhan model DFD, semua nama data yang muncul di semua DFD diisi pada kamus data dan kamus data berisi definisi semua data. [4]

2.2.11.1 Diagram Konteks

Diagram konteks adalah level teratas dari semua aliran data sistem yang direpresentasikan pada sebuah lingkaran besar. Lingkaran pada diagram konteks diberi anotasi dengan nama sistem perangkat lunak yang sedang dikembangkan dengan tujuan untuk menangkap konteks sistem daripada fungsinya. Pengembangan diagram konteks harus didasari dari analisis dokumen kebutuhan untuk mengidentifikasi berbagai jenis pengguna yang akan menggunakan sistem dan jenis data apa yang akan dimasukkan ke sistem serta data yang diterima oleh sistem. [4]

2.2.11.2 DFD Level 1

DFD Level 1 biasanya berisi tiga sampai tujuh proses yang merepresentasikan fungsi penting di setiap prosesnya. Pengembangan level 1 DFD diperiksa pada persyaratan fungsional tingkat teratas dalam dokumen kebutuhan. [4]

2.2.12 ERD (Entity Relationship Diagram)

ERD merupakan penggambaran komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari dunia nyata.

Notasi di dalam ERD yang dapat digunakan adalah sebagai berikut. [16]

1. Persegi panjang, menyatakan suatu entitas.
2. Lingkaran/Elip, menyatakan atribut.
3. Belah ketupat, menyatakan relasi.
4. Garis, menyatakan penghubung antara entitas dengan entitas dan entitas dengan atribut.

Kardinalitas relasi pada ERD menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi yang terjadi antara dua himpunan entitas meliputi: [16]

1. Satu ke Satu (*One to One*), yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B.
2. Satu ke Banyak (*One to Many*), yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.
3. Banyak ke Satu (*Many to One*), yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya.
4. Banyak ke Banyak (*Many to Many*), yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya.

2.2.13 MySQL

MySQL merupakan sistem manajemen basis data yaitu hubungan antar tabel yang berisi data-data dari suatu *database* terstruktur yang dirancang untuk mengelola suatu data. MySQL dikembangkan tahun 1994 oleh sebuah perusahaan perangkat lunak bernama MySQL-AB yang telah digunakan sebagian besar *web server* karena *database* bersifat *client server* di mana data disimpan pada *server* yang dapat diakses melalui komputer *client*. MySQL merupakan *database* yang dikembangkan dari bahasa SQL (*Structured Query Language*) yaitu bahasa

terstruktur yang digunakan untuk interaksi antara skrip program dengan *database* dalam hal pengolahan data. [16]

2.2.14 Black Box Testing

Pengujian *Black Box* merupakan pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dimana sekumpulan kondisi *input* didefinisikan oleh penguji dan melakukan pengetesan pada spesifikasi fungsional program. [17]

Black Box Testing bertujuan untuk menemukan hal-hal berikut: [17]

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka.
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi.
5. Kesalahan inisialisasi dan terminasi.

Adapun metode atau teknik untuk melakukan pengujian *Black Box*, antara lain: [17]

1. *Equivalence Partitioning*
2. *Boundary Value Analysis*/*Limit Testing*
3. *Comparison Testing*
4. *Sample Testing*
5. *Robustness Testing*
6. *Behavior Testing*
7. *Requirement Testing*
8. *Performance Testing*
9. Uji Ketahanan (*Endurance Testing*)
10. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*)