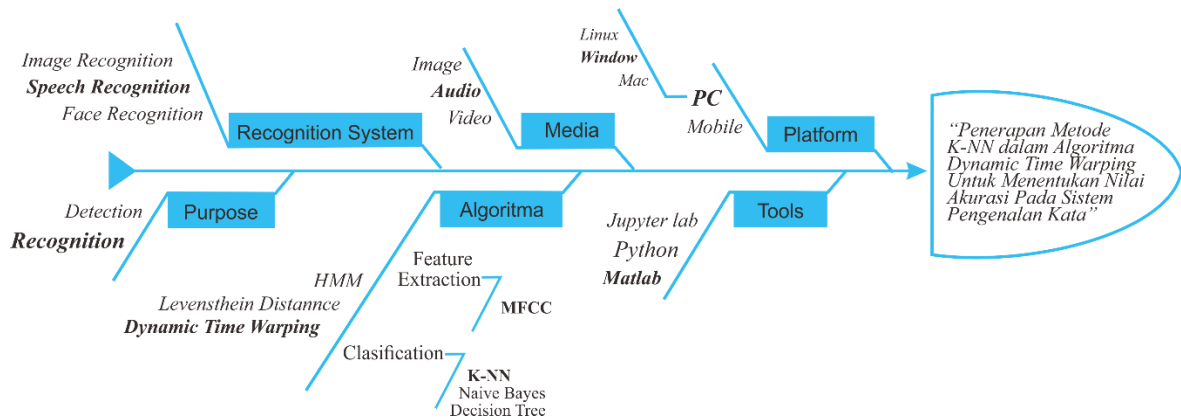


## BAB II

### LANDASAN TEORI

#### 2.1 Diagram Fishbone



**Gambar 2.1** Diagram Fishbone

Gambar 2.1 merupakan diagram fishbone yang menggambarkan hal-hal yang terkait dengan penelitian. Penelitian dibuat untuk melakukan pengenalan atau *recognition* yaitu *speech recognition* dengan menggunakan media *audio*. Algoritma yang digunakan yaitu *MFCC* untuk *feature extraction*, *DTW* untuk pencocokan ditambah dengan metode *K-NN* untuk proses akhir yaitu klasifikasi. *Platform* yang digunakan adalah *Personal Computer* dengan operasi sistem yang digunakan yaitu *Windows 10*, sedangkan *Tool* yang digunakan yaitu aplikasi *Matlab*.

#### 2.2 Kajian Penelitian Sebelumnya

Berikut merupakan kajian penelitian dari penelitian sebelumnya yang berhubungan dengan beberapa algoritma *speech recognition* untuk pengenalan kata atau ucapan suara ditunjukkan pada tabel 2.1:

Tabel 2.1 Literature Review

No	Penulis Penelitian	Judul Penelitian	Masalah Penelitian	Metode/Algoritma /Model Penelitian	Hasil Penelitian
1	Youllia Indrawaty, Andriana Zulkarnain, Reza Rianto. (2018)	<i>Pengembangan Pembelajaran Pengenalan Kata dalam Bhs Indonesia Menggunakan Multimedia Interaktif Dan Speech Recognition</i>	Media pembelajaran pada anak usia dini kurang menarik dan edukatif.	Algoritma: 1. <i>Algoritma Jaro Winkler Distance</i> 2. <i>Algoritma Levenshtein Distance</i>	<i>Algoritma Jaro-Winkler Distance</i> dan <i>Algoritma Levenshtein Distance</i> telah berhasil digunakan untuk mengukur akurasi pengucapan kata.
2	Aliv Faizal Muhammad, Akhmad Alimudin (2018)	<i>Penerapan Teknologi Speech Recognition untuk Latihan Pronunciation Bahasa Inggris Melalui Metode Dictation di Program Studi Multimedia BC</i>	Keterbatasan sumber belajar pronunciation bahasa Inggris dimana mahasiswa hanya belajar dari tatap muka di kelas	Metode: <i>Dictation</i>	Penggunaan <i>teknologi speech recognition</i> dengan <i>metode dictation</i> cenderung susah dilaksanakan namun dapat memotivasi belajar peserta didik.
3	Rahmawati Sitti Azizah, Dade Nurjanah, Ir.,M.T., Phd., Florita Diana Sari, SS., M.Pd. (2015)	<i>Automatic Speech Recognition System using MFCC and HMMs Method for Detecting English Language Pronunciation Mistake</i>	Adanya kesalahan pengucapan akibat tertukarnya satu kata dengan kata lain yang mirip dan sulit dideteksi oleh komputer.	Metode: 1. <i>Mel Frequency Cepstral Coefficient (MFCC)</i> 2. <i>Hidden Markov Models (HMM)</i>	Sistem <i>Speech Recognition</i> berhasil diimplementasi dengan <i>Algoritma MFCC</i> sebagai ekstraksi ciri dengan digabungkan dengan <i>Hidden Markov Model (HMM)</i> sebagai algoritma pencocokan. Algoritma ini mampu membedakan pasangan kata yang memiliki tingkat kemiripan dan cenderung tertukar pelafalannya.

Tabel 2.1 Literature Review (Lanjutan)

No	Penulis Penelitian	Judul Penelitian	Masalah Penelitian	Metode/Algoritma /Model Penelitian	Hasil Penelitian
4	I Kadek Suryadharma, Gelar Budiman, ST., MT., Budhi Irawan, Ssi.,MT. (2014)	<i>Design Application Speech To Text English To Balinese Language Using Pocketsphinx Base On Android</i>	Kendala perbedaan bahasa sering dijumpai terutama di negara dengan kota wisata yang menarik perhatian seperti Bali.	Metode: 1. <i>Hidden Markov Model (HMM)</i> 2. <i>Mel frequency cepstral coefficient</i> Library: 3. <i>Pocketsphinx</i>	Implementasi dari <i>pocketsphinx</i> ke dalam aplikasi <i>speech to text</i> dapat direalisasikan dengan hasil pengenalan yang baik. Ini terlihat dari tingkat akurasi yang dihasilkan di atas 80%.
5	M.Tri Satria Jaya, Diyah Puspitaningrum, Boko Susil. (2016)	<i>Penerapan Speech Recognition pada Permainan Teka-Teki Silang Menggunakan Metode Hidden Markov Model (HMM) Berbasis Desktop .</i>	Pengenalan suara atau <i>speech recognition</i> jarang dikembangkan dan digunakan dalam aplikasi game saat ini, sehingga game yang ada menjadi kurang menarik.	Metode: <i>Hidden Markov Model (HMM)</i>	Aplikasi penerapan <i>speech recognition</i> pada permainan Teka-teki silang menggunakan metode <i>Hidden Markov Model (HMM)</i> berbasis desktop ini, sistem pengenalannya memerlukan tempat pelatihan dan pengenalan yang jauh dari gangguan (noise).
6	Muhammad Anggia Muchtar, Raisha Ariani Sirait, Romi Fadillah Rahmat (2015)	Penerapan Perintah Suara Berbahasa Indonesia untuk Mengoperasikan Perintah Dasar di Windows	Penggunaan bahasa Indonesia jarang digunakan dalam inputan voice atau <i>speech recognition</i> .	Metode: 3. <i>Fast Fourier Transform (FFT)</i>	Penerapan <i>Speech Application Programming Interface (SAPI)</i> dan algoritma <i>Fast Fourier Transform</i> sesuai dengan kebutuhan aplikasi yang berhasil berjalan dengan tingkat keberhasilan 81,15 %.

Tabel 2.1 Literature Review (Lanjutan)

No	Penulis Penelitian	Judul Penelitian	Masalah Penelitian	Metode/Algoritma /Model Penelitian	Hasil Penelitian
7	Andriana, Olly V, Riyanto S, Ganjar T, Zulkarnain (2016)	<i>Speech Recognition Sebagai Fungsi Mouse Untuk Membantu Pengguna Komputer Dengan Keterbatasan Khusus</i>	Keterbatasan tuna daksa /individu yang memiliki gangguan gerak menjadi masalah utama dalam aktivitas salah satunya menjalankan komputer karena keterbatasan menggunakan mouse.	Metode: <i>Easy Voice Recognition Modul</i>  Model: <i>4.Arduino Leonardo</i>	<i>Easy VRM</i> memiliki tingkat keberhasilan di atas 90 %, jika menerima perintah suara dari orang yang berbeda, kegagalan dalam pengolahan perintah suara disebabkan oleh intonasi kata yang diucapkan tidak sesuai dengan database yang ada.
8	Ammar Mohammed, Moh. Shahrizal Sunar, Md. Sah Hj Salam (2015)	<i>Quranic Verses Verification using Speech Recognition Techniques</i>	Larangan kesalahan dalam pembacaan ayat suci Al-quran dan tidak ada aplikasi otomatis yang dapat mengenali bacaan Al-Quran menggunakan teknik pengenalan ucapan	Metode: 1. <i>Hidden Markov Model (HMM)</i> 2. <i>Vector Quantization (VQ)</i> <i>Artificial Neural Network (ANN)</i> .	Metode <i>Hidden Markov Model (HMM)</i> , <i>Vector Quantization (VQ)</i> , <i>Artificial Neural Network (ANN)</i> yang digunakan sudah cukup baik dan terbukti dapat mengenali suara bahasa arab, namun perlu ditingkatkan untuk akurasi yang lebih dalam ayat-ayat Al-Quran terutama online.
9	M.Taofik Chulkamdi, Sulis Purnomo. (2016)	<i>Perancangan Dan Implementasi Game Interaktif Pengenalan Huruf Dan Angka Untuk Media Pembelajaran Di Paud Wachid Hasyim Ponggok Kabupaten Blitar</i>	Kurangnya waktu belajar pada anak, karena anak lebih suka bermain.	Metode: <i>Blinnd Searcng</i>  Algoritma: <i>4.Breadth-first search (BFS)</i> .	Penerapan <i>metode Blinnd Searching</i> dengan <i>algoritma BFS</i> cukup baik diterapkan dalam game interaktif pengenalan huruf dan angka pada anak.

Tabel 2.1 Literature Review (Lanjutan)

No	Penulis Penelitian	Judul Penelitian	Masalah Penelitian	Metode/Algoritma /Model Penelitian	Hasil Penelitian
10	Sefty Wijayanti, Asep Nurhuda, Reza Andrea. (2018)	<i>Edugame “Etam-tainment” Pembelajaran Bahasa Kutai dengan Shuffle Random dan Agen Cerdas</i>	Belum ada penelitian mengenai membangun edugame puzzle yang mengenalkan pelajaran muatan lokal tentang bahasa Kutai. Banyaknya permainan yang tidak dimainkan lagi karena gameplay yang membosankan.	Metode: <i>Finite State Machine</i>  Algoritma: <i>5.Shuffle random</i>	Penggabungan kecerdasan buatan (Artificial Intelligence/AI) dalam permainan edugame ini, membuat permainan belajar bahasa Kutai lebih interaktif di dalam proses belajar sambil bermain. Hasil persentase kelayakan 86%, sehingga edugame “Etam-tainment” ini dapat diterima dalam rating kelayakan sebuah aplikasi permainan.
11	Totok Chamidy (2016)	<i>Metode Mel Frequency Cepstral Coeffisients (MFCC) Pada klasifikasi Hidden Markov Model (HMM) Untuk Kata Arabic pada Penutur Indonesia</i>	Sinyal suara yang dalam pengucapannya mengucapkan teks yang berbeda-beda, menghasilkan pola ucapan yang berbeda-beda pula. Hal ini juga terjadi jika pengucapan teks suatu bahasa diucapkan oleh orang yang bukan merupakan bahasa ibu orang tersebut.	- <i>MFCC</i> - <i>HMM</i>	Sistem menghasilkan nilai akurasi rata-rata sebesar 83,1% untuk frekuensi sampling data uji sebesar 8000 Hz, 82,3% untuk frekuensi sampling data uji sebesar 22050 Hz, 82,2% untuk frekuensi sampling data uji sebesar 44100 Hz.

Tabel 2.1 Literature Review (Lanjutan)

No	Penulis Penelitian	Judul Penelitian	Masalah Penelitian	Metode/Algoritma /Model Penelitian	Hasil Penelitian
12	Dahlan Abdullah, Cut Ita Erliana. (2017)	<i>Aplikasi Pengenalan Ucapan Huruf Jepang Menggunakan Hidden Markov Model (HMM)</i>	kendala bahasa sering terjadi antara berbagai negara salah satunya kendala bahasa jepang, maka tidak sedikit yang belum bisa bahasa jepang padahal pemerintahan jepang banyak sekali membuka peluang dalam berbagai bidang salah satunya pendidikan.	<i>Metode: Hidden Markov Model (HMM). Mean dan Tranformasi Mellin.</i>	Dengan menggunakan metode <i>Hidden Markov Model</i> yang memiliki hasil yang cukup akurat dalam mengenali kata yang diucapkan. Hasil dari pengenalan ucapan menunjukkan keakuratan aplikasi dengan nilai kedekatan rata-rata adalah 0,83234179 atau 83%. Nilai kedekatan dalam pengenalan ucapan dalam aplikasi menunjukkan bahwa metode <i>Hidden markov Model</i> dapat digunakan sebagai salah satu pendekatan untuk pengenalan sinyal dan suara.
13	Jefrizal, Jaroji, Agus Tedyyana. (2017)	Aplikasi English Teacher Sebagai Alat Bantu Belajar English Conversation Berbasis Android dengan Menerapkan Voice Recognition	Belajar bahasa inggris secara manual atau dengan melakukan pengetikan rentan akan kesalahan dalam penulisan	Metode : <i>Automatic Speech Recognition</i>	Aplikasi dapat menerima masukan suara berdasarkan jarak antara perangkat - pengguna, aplikasi harus diterapkan ditempat yang tidak banyak suara. Semakin jauh jaraknya, semakin rendah tingkat respon penangkapan pengenalan suara.

Tabel 2.1 Literature Review (Lanjutan)

No	Penulis Penelitian	Judul Penelitian	Masalah Penelitian	Metode/Algoritma /Model Penelitian	Hasil Penelitian
14	Candra Dinata, Diah Puspitaningrum, Ernawati. (2017)	<i>Implementasi Teknik Dynamic Time Warping (Dtw) Pada Aplikasi Speech To Text</i>	Implementasi speech recognition masih jarang digunakan, Waktu keselarasan ucapan yang berbeda adalah masalah inti untuk pengukuran jarak dalam pengenalan ucapan.	<i>Metode: mel-frequency cepstral coefficients (MFCC) Dynamic time warping (DTW)</i>	Sistem mampu mengenali kata dengan rata-rata akurasi sebesar 95.85% dan tingkat kesalahan sebesar 4,15 %.
15	Rahmawati Sitti Azizah, Dade Nurjanah, Ir., M.T., Phd., Florita Diana Sari, SS., M.Pd. (2015)	<i>Sistem Automatic Speech Recognition Menggunakan Metode MFCC dan HMMs Untuk Deteksi Kesalahan Pengucapan Kata Bahasa Inggris</i>	sering tertukarnya sebuah kata dengan kata yang lain menjadi masalah yang cukup sering dialami oleh seseorang yang mempelajari bahasa Inggris.	<i>Metode: Mel Frequency Cepstral Coefficient (MFCC) Hidden Markov Models (HMM)</i>	Pengujian dilakukan dengan menggunakan 10 pasangan kata dengan tingkat kemiripan yang tinggi dan sering tertukar jika dilafalkan secara terpisah. Dari hasil pengujian didapat tingkat akurasi rata-rata setiap pasangan kata sebesar 78,89% pada model HMM 3 state dan 78,33% pada model HMM 5 state

Berikut kelebihan dan kekurangan beberapa penelitian yang terdapat pada tabel literatur review:

- [1] Memiliki kelebihan dari segi tampilan aplikasi namun dengan kekurangan tidak adanya nilai akurasi atau ketepatan yang dihasilkan dari aplikasi yang dibuat.
- [2] Memiliki kelebihan dalam memotivasi belajar peserta didik, namun kekurangan penggunaan teknologi speech recognition dengan metode dictation cenderung susah dilaksanakan karena masalah hardware yang dimiliki anak berbeda.
- [3] Sistem ASR yang dibuat mampu membedakan pasangan kata yang memiliki tingkat kemiripan dan kecenderungan untuk tertukar pelafalannya cukup tinggi. Namun dari hasil pengujian tingkat akurasi tidak terlalu tinggi akurasi rata-rata setiap pasangan kata sebesar 78,89% pada model HMM 3 state dan 78,33% pada model HMM 5 state.
- [4] Kelebihan dari penelitian yang dilakukan yaitu bisa mengembangkan bahasa asli Indonesia dan mengenalkan kepada turis yang datang ke Bali, tapi kekurangannya dari segi format yang dimasukkan yaitu format .dic dimana format tersebut tidak familiar dikalangan masyarakat.
- [5] Permainan diselesaikan dengan baik namun kekurangannya dari segi perhitungan akurasi yang terlalu rumit tanpa ada rata-rata nilai akurasi keseluruhan.
- [6] Bahasan penelitian cukup menarik namun kekurangannya dari segi metode, karena metode SAPI sudah terlalu banyak digunakan.
- [7] Kelebihan dari penelitian yang dibuat yaitu dari segi inovasi penelitian sehingga benar-benar bisa diterapkan dalam kehidupan sehari-hari untuk



membantu orang berkebutuhan khusus dengan dengan tingkat akurasi 91%, namun kekurangannya dari segi hasil untuk inputan suara dengan berbeda intonasi yang sedikit susah dikenali.

- [8] Sistem yang dibuat cukup baik karena sudah bisa mengenali pengucapan dalam bahasa arab, kekurangannya perlu ditingkatkan untuk akurasi yang lebih dalam ayat Al-Quran karena untuk pembahasan ini perlu adanya sistem pakar untuk membantu prediksi dalam hal hukum-hukum bacaan yang ada.
- [9] Kelebihannya game yang dibuat cukup interaktif untuk anak paud, namun dalam penelitian perlu adanya hasil akurasi atau survei penggunaan game atau aplikasi yang dapat menunjukkan apakah game bisa diterima dengan baik.
- [10] Game dikemas dengan menarik mengenalkan pembelajaran muatan lokal tentang bahasa Kutai, namun tidak ada akurasi penerapan algoritma yang dilakukan dalam penelitian.
- [11] Kelebihan dari sistem yang dibuat yaitu sistem tetap berjalan meskipun pengucapan teks suatu bahasa diucapkan oleh orang yang bukan dengan dialek bahasa arab yang benar. Kekurangannya dari segi algoritma yang digunakan karena sudah terlalu umum memakai algoritma MFCC dan HMM untuk prosesnya.
- [12] Algoritma dari sistem bisa diterapkan dengan baik dengan hasil akurasi 83% namun dengan algoritma HMM dan algoritma tersebut sudah umum digunakan sehingga kurangnya keterbaruan dari penerlitan yang dibuat.
- [13] Aplikasi dapat menerima masukkan suara berdasarkan jarak antara perangkat dan pengguna, aplikasi harus diterapkan ditempat yang tidak banyak suara.

Semakin jauh jaraknya, semakin rendah tingkat respon penangkapan pengenalan suara.

[14] Kelebihan sistem yang dibuat mampu mengenali kata dari data kalimat, namun kurangnya jumlah data dalam *database* dan banyaknya suara yang noise sehingga mempengaruhi hasil akurasi sistem yang dibuat.

[15] Penelitian yang dilakukan cukup menarik namun jumlah data pengujian hanya 10 pasang kata dan itu terlalu sedikit sehingga menghasilkan akurasi yang tidak terlalu tinggi yaitu 73%.

Dari beberapa penelitian yang dilakuakn rata-rata kekurangan sistem yaitu dari segi akurasi yang kurang dan jumlah *database* atau *dataset* yang sedikit sehingga mempengaruhi hasil akurasi. Salah satunya yaitu penelitian dengan judul “*Implementasi Teknik Dynamic Time Warping (DTW) Pada Aplikasi Speech To Text*” Candra Dinata, Diyah Puspitaningrum, Ernawati. (2017), dengan Metode DTW yang digunakan dapat menghitung jarak atau selisih antara dua data yang dibandingkan. Sistem mampu mengenali kata dengan rata-rata akurasi sebesar 95.85% namun kurang efektif karena jumlah *dataset* yang terlalu sedikit dan suara noise dalam audio sehingga mempengaruhi hasil akurasi.

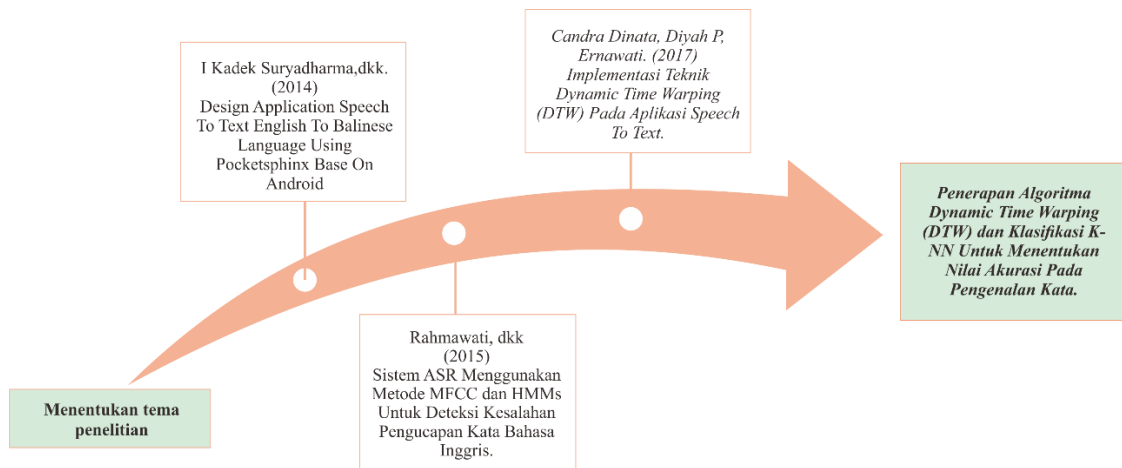
Berdasarkan beberapa penelitian yang telah disebutkan, maka diusulkan penelitian yang berjudul “*Penerapan Metode K-NN dalam Algoritma Dynamic Time Warping Untuk Menentukan Nilai Akurasi Pada Sistem Pengenalan Kata*” untuk meneruskan penelitian yang dilakukan oleh Candra Dinata (2017).



Tabel 2.3 Matriks Penelitian (Lanjutan)

No.	Penulis / Tahun	Judul Jurnal	Ruang Lingkup Penelitian											
			Algoritma										Tujuan	
			HMM	<i>Levenshtein Distance</i>	MFCC	Dictation	<i>FFT</i>	Naive Bayes	FSM	DTW	BFS	K-NN	Pengenalan	Deteksi
12	Dahlan A, ..	<i>(Aplikasi Pengenalan Ucapan ...)</i>	✓	-	-	-	-	-	-	-	-	-	✓	-
13	Jefrizal, Jaroji, ...	<i>(Aplikasi English Teacher ...)</i>	✓	-	-	-	-	-	-	-	-	-	✓	-
14	Candra Dinata, ...	<i>(Implementasi Teknik Dynamic ...)</i>	-	-	✓	-	-	-	-	✓	-	-	✓	-
15	Rahmawati Sitti, ...	<i>(Sistem Automatic Speech ...)</i>	✓	-	✓	-	-	-	-	-	-	-	-	✓
16	Rina Rosdiana (2020)	<i>“Penerapan Metode K-NN dalam Algoritma Dynamic Time Warping Untuk Menentukan Nilai Akurasi Pada Sistem Pengenalan Kata”</i>	-	-	✓	-	-	-	-	✓	-	✓	✓	-

## 2.4 Peta Penelitian



**Gambar 2.2** Peta Penelitian

Peta penelitian pada gambar 2.2 menjelaskan alur penelitian serta jalan untuk mengetahui arah penelitian yang dilakukan yang dimulai dari penentuan tema sampai mendapatkan judul untuk penelitian baru.

## 2.5 *State of The Art*

Berdasarkan beberapa penelitian yang telah dijelaskan sebelumnya, algoritma *Dynamic Time Warping (DTW)* memiliki tingkat akurasi yang lebih baik dibandingkan dengan algoritma *Hidden Markov Model (HMM)*, sehingga memungkinkan proses pengenalan ucapan kosa kata bahasa Inggris menjadi lebih akurat. Namun kelemahan DTW yaitu waktu pemrosesan data yang cukup lama sehingga mempengaruhi ketepatan prediksi, maka digunakanlah metode klasifikasi sebagai proses akhir atau validasi sistem. Pada penelitian ini algoritma klasifikasi yang digunakan adalah K-NN untuk mengklasifikasikan apakah suara yang diujikan benar atau tidak dengan data training yang tersedia.

Maka diusulkan penelitian berjudul “*Penerapan Metode K-NN dalam Algoritma Dynamic Time Warping Untuk Menentukan Nilai Akurasi Pada Sistem Pengenalan Kata*”. State of The Art diperoleh dari perbandingan antara penelitian ini dengan penelitian sebelumnya yang dilakukan oleh Candra Dinata tahun 2017 dengan objek penelitian yang hampir sama yaitu penerapan speech recognition.

## **2.6 Teori Pendukung Penelitian**

### **2.6.1 Recognition System**

Cabang dari AI yang saat ini banyak dipelajari yakni pembelajaran mendalam atau *deep learning*. Terobosan *deep learning* membuat AI makin populer karena implementasinya banyak digunakan hampir di semua lini industri. *Deep learning* mampu melatih komputer untuk mengerjakan pekerjaan manusia, seperti:

#### 1) ***Image Recognition***

*Image recognition* merupakan pengenalan yang berorientasi pada image atau gambar (Fauziah Dewi, 2019). Tekniknya sama dengan face recognition, namun bedanya image recognition ini mengenali langsung secara spesifik dari image statis (tidak hidup) seperti video sedangkan face recognition mengenali dari image yang ditangkap terlebih dahulu dari video baru dikenali.

#### 2) ***Face Recognition***

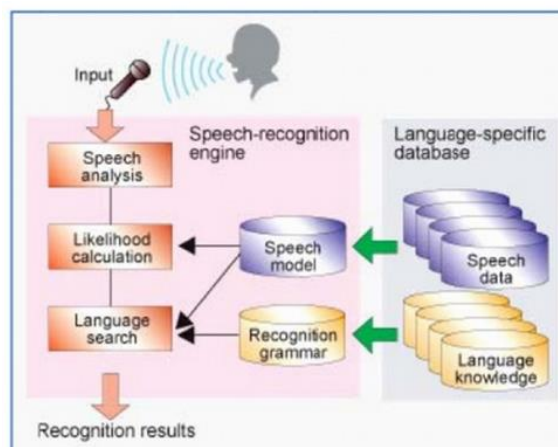
*Face recognition* adalah suatu metoda pengenalan yang berorientasi pada wajah yang merupakan salah satu sistem implementasi deep learning yang dapat mengenal wajah secara spesifik dari gambar digital atau *frame* video. (Luthfi Maslichul Kurniawan, 2014).

Tidak hanya wajah yang dapat dikenali, namun juga usia, keadaan emosional, fitur-fitur wajah, dan sebagainya. Pengenalan ini dapat dibagi menjadi dua bagian yaitu : Dikenali atau tidak dikenali, setelah dilakukan perbandingan dengan pola yang sebelumnya disimpan di dalam database. Metoda ini juga harus mampu mengenali objek bukan wajah.

### 3) *Speech Recognition*

Pengenalan ucapan atau suara (speech recognition) adalah suatu teknik yang memungkinkan sistem komputer untuk menerima input berupa kata yang diucapkan. Kata-kata tersebut diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka lalu disesuaikan dengan kode- kode tertentu dan dicocokkan dengan suatu pola yang tersimpan dalam suatu perangkat. Speech recognition adalah proses identifikasi suara berdasarkan kata yang diucapkan. Parameter yang dibandingkan ialah tingkat penekanan suara yang kemudian akan dicocokkan dengan template database yang tersedia. (Wawan Kurniawan, 2016).

Terdapat 4 langkah utama dalam pengenalan suara :



**Gambar 2.3** Skema Speech Recognition (Wawan Kurniawan, 2016)

- a. Penerimaan data input
- b. Ekstraksi, yaitu penyimpanan data masukan sekaligus pembuatan database untuk template.
- c. Perbandingan/pencocokan, yaitu tahap pencocokan data baru dengan data suara (pencocokan tata bahasa) pada template.
- d. Validasi identitas pengguna.

Secara umum, speech recognizer memproses sinyal suara yang masuk dan menyimpannya dalam bentuk digital. Hasil proses digitalisasi tersebut kemudian dikonversi dalam bentuk spektrum suara yang akan dianalisa dengan membandingkannya dengan template suara pada database sistem.

### **2.6.2 Feature Extraction**

*Feature extraction* (ekstraksi ciri) merupakan suatu pengambilan ciri / feature dari suatu sinyal informasi yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya (Ulfa Miftakhul, 2016). Prinsip kerja ekstraksi ciri adalah dengan mengkonversi sinyal suara ke dalam beberapa parameter, dimana ada sebagian informasi tidak berguna yang dibuang tanpa menghilangkan arti sesungguhnya dari sinyal suara tersebut. Hasil keluaran dari ekstraksi ciri ini menjadi masukan pada proses pengenalan pola.

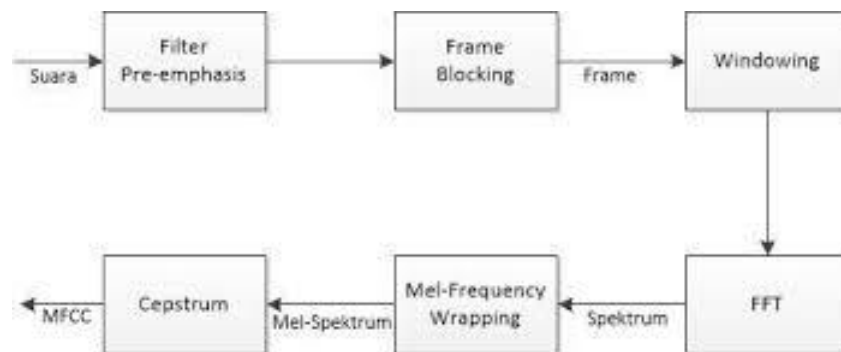
*MFCC (Mel Frequency Cepstrum Coefficient)* merupakan metode yang banyak digunakan dalam bidang speech technology, baik speaker recognition maupun speech recognition (Totok C, 2016). Metode ini digunakan untuk melakukan feature extraction, sebuah proses yang mengkonversikan sinyal suara menjadi beberapa parameter. Keunggulan dari metode ini adalah:



- a) Mampu untuk menangkap karakteristik suara yang sangat penting bagi pengenalan suara. Atau dengan kata lain, mampu menangkap informasi informasi penting yang terkandung dalam sinyal suara.
- b) Menghasilkan data seminimal mungkin tanpa menghilangkan informasi-informasi penting yang ada.

*MFCC (Mel Frequency Cepstrum Coefficients)* feature extraction sebenarnya merupakan adaptasi dari sistem pendengaran manusia dimana sinyal suara akan di-filter secara linear untuk frekuensi rendah (dibawah 1000 Hz) dan secara logaritmik untuk frekuensi tinggi.

Berikut ini alur diagram untuk proses yang dilakukan pada metode MFCC (Mel Frequency Cepstrum Coefficients)



**Gambar 2.4** Alur diagram metode MFCC (Anggoro Wicaksono, 2014)

Seperti yang terlihat pada gambar 2.4 terdapat 6 tahapan komputasi pada MFCC. Dimulai dengan masuknya inputan data suara yang direpresentasikan kedalam deretan amplitudo-amplitudo hingga membentuk koefisien yang merepresentasikan ciri dari sebuah kata. Setiap tahapan memiliki fungsi sebagai berikut:

### 2.6.2.1 Pre-Emphasis

Pada tahapan ini akan dilakukan filtering terhadap sinyal suara yang masuk dengan cara mengurangi nilai frekuensi sinyal tersebut sehingga nantinya hanya sinyal berfrekuensi tinggi saja yang dapat melewati filtering. Hal ini dilakukan untuk mengurangi noise dari sebuah suara sehingga hanya data sinyal suara yang sebenarnya saja yang dapat ditangkap oleh sistem. Secara matematis, dapat dirumuskan dalam persamaan (1).

$$sp(n) = s(n) - 0.97 s(n - 1) \dots\dots\dots(1)$$

Di mana  $sp(n)$  = sinyal yang ditekan,  $s(n)$  = sinyal terdigitasi. Koefisien nilai 0.97 menunjukkan sinyal yang diekstrak merupakan 97 % sinyal aslinya.

### 2.6.2.2 Framing

Pada tahapan ini sampel suara akan dipotong menjadi frame-frame berdurasi lebih pendek (framing) sebanyak  $M$  yang disimpan kedalam matriks  $Y$  berukuran  $M \times W$  dengan baris  $y_i$  menunjukkan nomor frame. Pada sampel suara sebenarnya akan ditemui sinyal suara yang tidak stabil sehingga akan susah mencari karakteristik dari sampel suara, oleh karena itu digunakanlah fungsi framing seperti pada persamaan (2).

$$Jumlah\ Frame = ((I-N) / M) + 1 \dots\dots\dots(2)$$

Penjelasan:  $I = Sampling\ Rate$

$N = Sample\ Point$

$M = Panjang\ Overlap$

Dengan memotong sampel kedalam frame-frame kecil maka sinyal suara yang ada akan lebih stabil sehingga kita mendapatkan karakteristik suara yang

lebih stabil. Frame yang dibuat akan overlapping dengan frame-frame lainnya untuk menghindari hilangnya informasi pada saat proses selanjutnya.

**2.6.2.3 Windowing**

Setiap frame diberikan perkalian dengan fungsi window untuk meminimalisir diskontinuitas pada awal dan akhir frame yang diakibatkan oleh overlapping pada proses framing. Hamming Window digunakan untuk mengintegrasikan semua garis frekuensi terdekat. Menentukan Hamming Window bisa dilihat pada persamaan (3).

$$w(n) = 0,54 - 0,46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 < n < N - 1 \dots\dots\dots(3)$$

$$y1(n) = x1(n)w(n), \quad 0 < n < N - 1$$

Dimana: W(n) = Hamming window      y1(n) = Output signal

N = Jumlah sampel      x1(n) = Input signal

**2.6.2.4 Fast Fourier Transform (FFT)**

Sinyal suara yang ada masih dalam domain waktu sehingga perlu dilakukan konversi dari domain waktu ke domain frekuensi. Untuk mendapatkan sinyal dalam domain frekuensi dari sebuah sinyal diskrit, salah satu metode yang digunakan adalah Fast Fourier Transform (FFT). FFT dilakukan terhadap frame dari sinyal yang telah di windowing, menggunakan algoritma Discrete Fourier Transform (DCT) versi cepat yang dioperasikan pada sinyal diskrit yang terdiri dari N sampel seperti pada persamaan (4).

$$f(n) = \sum_{k=0}^{N-1} y_k e^{-2\pi jkn/N} \dots\dots\dots(4)$$

Dimana : f(n)= Output signal      n= indeks sampel

N = Jumlah sampel      y = sinyal hasil windowing

### 2.6.2.5 Mel-Frequency Wrapping

Mel-Filterbank sama saja dengan triangular filterbank biasa, hanya saja range frekuensi linier yang didapat dari FFT dikonversi kedalam skala Mel-Frequency untuk mendapatkan batas-batas filterbank berdasarkan skala Mel-Frequency. Skala Mel dapat diperoleh dengan persamaan (5) dan (6).

$$Mel(f) = 2595 * \log_{10} \left( 1 + \frac{f}{700} \right) \dots\dots\dots(5)$$

$$Freq(m) = 700 \left( 10^{\frac{m}{2595}} - 1 \right) \dots\dots\dots(6)$$

f = frekuensi linear(Hz)

Untuk membuat *Mel-Filterbank* pertama tentukan batas atas dan bawah dari filter artinya nilai yang berada diluar batas itu tidak akan masuk kedalam filter kedua batas tersebut dikonversi kedalam skala mel, kemudian kita bagi range kedua batas tersebut sesuai dengan jumlah filter yang ingin dibuat dari sana akan diketahui batas atas dan bawah dari setiap filterbank dalam skala mel. Semua batas dikonversi kembali kedalam skala frekuensi linier, karena range yang ada tidak dapat merepresentasikan di bin FFT mana sajakah nilai-nilai tersebut dilakukan konversi dari nilai batas frekuensi linier kedalam nilai bin fft terdekat dan dibuatlah filter triangular berdasarkan batas tersebut. Hasil dari FFT pada tahap sebelumnya dikalikan dengan Mel-Filterbank.

### 2.6.2.6 Discrete Cosine Transform

Untuk mendapatkan nilai koefisien dari hasil perkalian mel-filterbank yang masih berada pada domain frekuensi diperlukan pengkonversian kembali kedalam domain waktu karena kita akan mengacu kepada urutan waktu dalam menentukan ciri. Pada langkah ini, hasil log dari perkalian

sebelumnya domain waktu menggunakan Discrete Cosine Transform (DCT). Hasilnya disebut sebagai Mel-Frequency Cepstral Coefficient (MFCC). MFCC bisa didapat dari pendekatan persamaan (7).

$$C_s(n; m) = \sum_{k=0}^{N-1} a_k \cdot \log(fmel_k) \cos\left(\frac{\pi(2n+1)k}{2N}\right) \dots\dots\dots(7)$$

N = jumlah sampel , n = 0,1,2,3...N-1, fmel = frekuensi mel

Dimana ceps adalah hasil akumulasi dari kuadrat magnitude DFT yang dikalikan dengan Mel-Filter Bank. Setelah itu didapatkan MFCC.

### 2.6.3 Algoritma Recognition

#### 2.6.3.1 HMM

*Hidden Markov Model (HMM)* adalah metode yang menggunakan pendekatan statistik terhadap penyelesaian permasalahan (Totok C, 2016). HMM biasa disebut sebagai *Markov Chain* atau *Markov Process*, ditemukan oleh Andrey Markov dan menggunakan prinsip markov chain untuk menyelesaikan permasalahan yang kita tidak tahu dengan pasti kondisi apa saja yang bisa terjadi. Model ini merupakan bagian dari finite state atau *finite automaton*. *Finite automation* sendiri adalah kumpulan state yang transisi antar state-nya dilakukan berdasarkan masukan observasi. HMM ini dapat memodelkan persoalan – persoalan di dunia nyata yang sifatnya probabilistik. Sebuah HMM dapat direpresentasikan dengan notasi  $\lambda = (A,B,\pi)$  dimana A, B dan  $\pi$  berturut – turut menyatakan distribusi peluang transisi antar state, distribusi peluang emisi symbol observasi, dan distribusi peluang initial state. (Muhammad Eko, 2010).

### 2.6.3.2 *Levenshtein Distance*

*Levenshtein distance* merupakan metrik yang digunakan untuk perhitungan beda jarak antara dua string. Metode ini sering juga disebut dengan Edit Distance (Husain, 2013). Yang dimaksud dengan distance adalah jumlah modifikasi yang dibutuhkan untuk mengubah suatu bentuk string ke bentuk string yang lain. Algoritma ini menghitung jumlah operasi string paling sedikit yang diperlukan untuk mentransformasikan suatu string menjadi string yang lain (Adiwidya, 2009). Algoritma Levenshtein distance bekerja dengan menghitung jumlah minimum pentransformasian suatu string menjadi string lain. Setelah mendapatkan biaya edit distance maka untuk menghitung nilai Levenshtein distance atau perhitungan similarity menggunakan persamaan berikut:

$$d1 = \left( \frac{d}{\text{maxlength}(\text{str1}, \text{str2})} \right)$$

dimana:  $d$  = edit distance dan  $\text{maxLength}(\text{str1}, \text{str2})$  = jumlah karakter terbanyak antara string 1 dan string 2

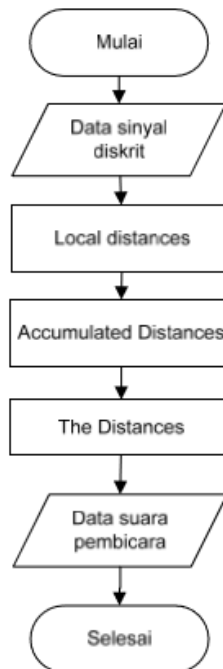
### 2.6.3.3 *Dynamic Time Warping (DTW)*

Dynamic Time Warping (DTW) adalah sebuah algoritma yang menghitung jalur pembengkokan optimal antara dua deretan waktu. Algoritma ini dimulai dengan perhitungan jarak lokal antara elemen dari dua urutan menggunakan berbagai jenis jarak (Candra Dinata, 2017).

Dua ucapan dari kata yang sama oleh pengguna yang sama dapat memiliki waktu yang berbeda. DTW menyelesaikan masalah ini dengan menyelaraskan kata-kata dengan benar dan menghitung jarak minimum

antara dua kata. Sebuah matriks jarak lokal dibentuk untuk semua segmen dalam kata sampel dan template kata. Dynamic Time Warping adalah metode yang efisien untuk memecahkan masalah keselarasan waktu.

Metode DTW dapat digambarkan dengan diagram berikut :



**Gambar 2.5** Flowchart *Dynamic Time Warping* (DTW)

Keterangan flowchart dynamic time warping di atas:

### 1) Local Distance

*Local distances* adalah jumlah jarak Euclidean, digunakan untuk mengukur jarak absolute antara nilai-nilai dari kedua elemen, distance adalah suatu ukuran yang dipakai untuk menentukan kemiripan antara dua buah data dan memberikan informasi tentang beberapa suara yang telah diketahui antara lain panjang frame dan nilai spektrum dari tiap frame.

```

Procedure local distance ( var c,d: long);
Kamus
  i,j,n: integer
  D (0,0)=0
algoritma
  For i = 1 to n do
    D (i, 0) = D (i-1,0) + 1
  EndFor
  For j = 1 to j do
    D (0, j) = D (0, j-1) + 1
  EndFor
  For i = 1 to i do
    For j = 1 to j do
      c1 = D (i-1, j-1) + 2d (i, j)
      c2 = D (i-1, j) + d(i,j)
      c3 = D (i, j-1) + d(i,j)
      D (i, j) = min (c1, c2, c3)
    EndFor
  EndFor

```

**Gambar 2.6** Algoritma prosedur *Local Distance*

## 2) Accumulated Distances

Setelah diketahui jalur optimal nama suara, panjang frame dan nilai spektrum dari tiap frame selanjutnya proses Accumulated Distances yaitu jarak perhitungan antara jarak lokal satu dengan jarak lokal sampingnya. Accumulated distances dapat dicari dengan mencari jarak minimal antara jarak lokal ditambah jarak akumulasi sampingnya.

```

Procedure accumulated distances;
kamus
  n ← X
  m ← Y
  dtw[] ← new [n x m]
  dtw(0, 0) ← 0
Algoritma
  for i ← 1 ; i ≤ n; j ++ do
    dtw(i, 1) ← dtw(i - 1, 1) + c(i,1)
  end for
  for j ← 1; j ≤ m; j ++ do
    dtw(1, j) ← dtw(1, j - 1) + c(1, j)
  end for
  for i ← 1; i ≤ n; j ++ do
    for j ← 1; j ≤ m; j ++ do
      dtw(i, j) ← c(i, j) + min {dtw(i - 1, j); dtw(i, j - 1); dtw(i - 1, j - 1)}
    end for
  end for
  return dtw

```

**Gambar 2.7** Algoritma prosedur *Accumulated Distance*



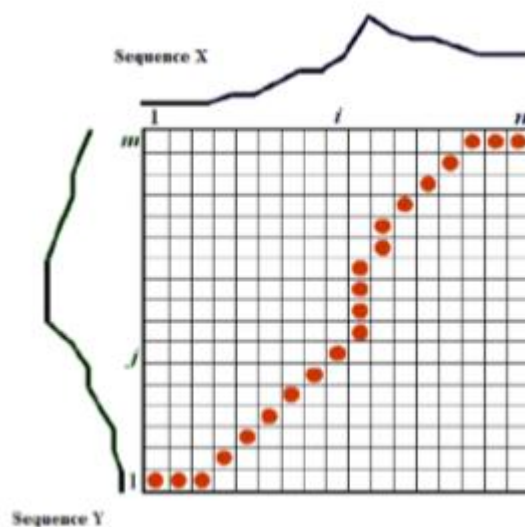
### 3) The Distances

The Distances adalah jarak akhir yang menunjukkan warping path jarak antara dua suara yang diinputkan dengan jarak suara yang ada di file suara yang telah tersimpan. Hasil dari proses the distances adalah deskripsi suara ucapan data pembicara.

Algoritma DTW ditujukan untuk menyelaraskan dua sekuen vector dengan membelokkan sumbu waktu berulang-ulang sampai kecocokan optimal antara dua sekuen ditemukan. algoritma ini melakukan sebagian pemetaan linear dari sumbu axis untuk menyelaraskan kedua sinyal. Anggap saja terdapat dua urutan vector di ruang n-dimensi :

$$x = [x_1, x_2, \dots, x_n] \text{ dan } y = [y_1, y_2, \dots, y_n] \dots\dots\dots(8)$$

Dua urutan tersebut selaras di sisi kotak, dengan satu di atas dan lainnya di sisi kiri. Kedua urutan mulai di bagian bawah kiri grid.



**Gambar 2.8** Contoh Jarak Grid Global

Dalam setiap sel, ukuran jarak ditempatkan dan dibandingkan sesuai dari dua sekuens. Jarak antara dua titik dihitung melalui jarak Euclidean .

$$Dist(x, y) = |x - y| = [ (x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2 ]^{1/2} \dots\dots\dots(9)$$

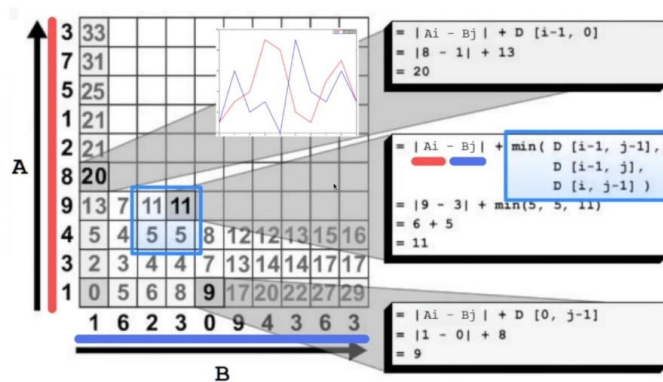
Pencocokan terbaik atau keselarasan antara dua sekuens ini adalah jalan melalui grid, yang meminimalkan total jarak antara mereka, yang disebut sebagai jarak global. Keseluruhan jarak (jarak Global) dihitung dengan menemukan dan pergi melalui semua rute yang mungkin melalui grid, masing-masing menghitung jarak keseluruhan. Jarak global minimum dari jumlah jarak (jarak Euclidean) antara unsur-unsur individual di jalan dibagi dengan jumlah dari fungsi pembobotan. Untuk setiap urutan cukup panjang jumlah kemungkinan jalan melalui grid akan sangat besar. mengukur jarak global diperoleh dengan menggunakan rumus rekursif .

$$GD_{xy} = LD_{xy} + \min (GD_{x-1 y-1}, GD_{x-1 y}, GD_{x y-1}) \dots\dots\dots(10)$$

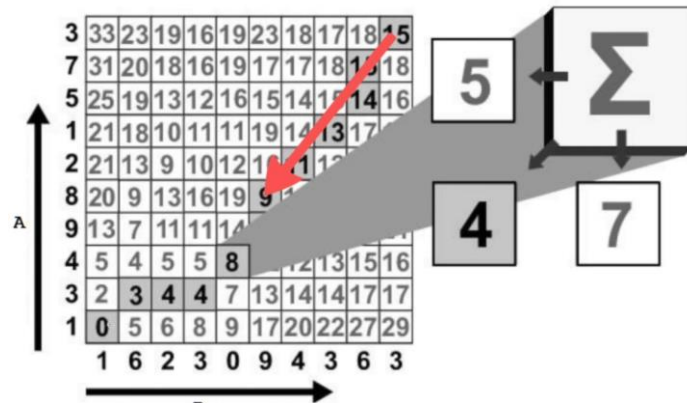
Dimana, GD = Global Distance (overall distance)

LD = Local Distance (Euclidean distance)

Gambar 2.5 dan 2.6 adalah ilustrasi dari algoritma DTW untuk menentukan jarak lokal dan optimal path:



Gambar 2.9 Ilustrasi perhitungan local distance



**Gambar 2.10** Ilustrasi backtracking untuk optimal path

#### 2.6.4 Algoritma Klasifikasi

Klasifikasi merupakan cara pengelompokan data sesuai dengan ciri-ciri atau karakteristik data tersebut atau pengelompokan fitur ke dalam kelas yang sesuai (Aji Prasetya, 2018). Dalam klasifikasi, terdapat dua fungsi utama yang dilakukan, yaitu pembangunan model sebagai prototype untuk disimpan sebagai memori dan penggunaan model tersebut untuk melakukan pengenalan / klasifikasi / prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya (Naziah Amalia, 2015).

##### 2.6.4.1 Naive Bayes

*Naive Bayes* merupakan sebuah model klasifikasi statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas, didasarkan pada teorema bayes yang memiliki kemampuan klasifikasi serupa dengan decision tree dan neural network. Teknik ini salah satu bentuk kecil dari Bayesian yang jaringan untuk klasifikasi. Sebuah jaringan Bayes dapat dilihat sebagai diarahkan sebagai tabel dengan distribusi probabilitas gabungan lebih dari satu set diskrit dan variabel stokastik (Liao, 2007).

#### 2.6.4.2 *Decision Tree*

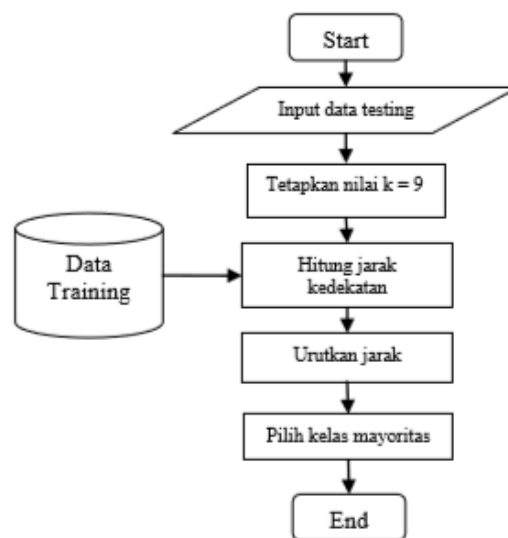
*Decision tree* merupakan metode klasifikasi dan prediksi yang sangat kuat dan banyak di minati (Wu, 2009) . Dalam *decision tree* ini data yang berupa fakta dirubah menjadi sebuah pohon keputusan yang berisi aturan dan tentunya dapat lebih mudah dipahami dengan bahasa alami. Model pohon keputusan banyak digunakan pada kasus data dengan output yang bernilai diskrit . Walaupun tidak menutup kemungkinan dapat juga digunakan untuk kasus data dengan atribut numeric.

#### 2.6.4.3 *K-Nearest Neighbor (K-NN)*

Metode *K-Nearest Neighbor* adalah salah satu metode *clasification* yang sangat sederhana dan bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan KNN-nya. Selain itu, metode ini sangat tangguh terhadap *training data* yang memiliki banyak *noise* sehingga cocok diterapkan dalam sistem *speech recognition* untuk dapat mengenali suara. Sebagaimana yang telah diketahui bahwa lebih banyak data yang diakses akan lebih efektif hasil akurasi (Indrayanti, 2017).

Tujuan algoritma KNN adalah mengklasifikasikan obyek baru berdasarkan atribut dan *training sample*. Clasifier tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik *query*, akan ditemukan sejumlah *k* obyek atau (titik *training*) yang paling dekat dengan titik *query*. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari *k* obyek. Algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari *query instance* yang baru.

Nilai  $k$  yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai  $k$  yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Menurut Indrayanti, dkk dalam penelitiannya yang berjudul “*Optimasi Parameter K Pada Algoritma K-Nearest Neighbour Untuk Klasifikasi Penyakit Diabetes Mellitus*” mengungkapkan bahwa jumlah  $K=1$  akan membuat hasil klasifikasi terasa kaku karena hanya memperhitungkan satu tetangga terdekat atau satu record karakteristik data terdekat. Sedangkan jumlah  $K$  yang terlalu banyak akan menghasilkan klasifikasi yang samar.



**Gambar 2.9** Alur Algoritma K-NN

1. Tentukan parameter  $K$
2. Hitung jarak antara data yang akan dievaluasi dengan semua pelatihan
3. Urutkan jarak yang terbentuk (dari yang terkecil)
4. Tentukan jarak terdekat sampai urutan  $K$
5. Pasangkan kelas yang bersesuaian

6. Cari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi

$$\text{Rumus KNN: } d_t = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2}$$

Keterangan:  $x_1$  = Sampel Data                       $x_2$  = Data Uji / Testing

$i$  = Variabel Data                                   $d$  = Jarak

$p$  = Dimensi Data