

SKRIPSI

**SISTEM PENERJEMAH BAHASA ISYARAT INDONESIA
(BISINDO) MENGGUNAKAN METODE *GATED RECURRENT
UNIT* DAN *LONG SHORT TERM MEMORY* (STUDI KASUS
SLB-B YAYASAN PENDIDIKAN LUAR BIASA KOTA
MAKASSAR)**



Disusun dan diajukan oleh:

SITTI RAHMA

D121 19 1050

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2024

LEMBAR PENGESAHAN SKRIPSI**SISTEM PENERJEMAH BAHASA ISYARAT INDONESIA
(BISINDO) MENGGUNAKAN METODE *GATED RECURRENT
UNIT* DAN *LONG SHORT TERM MEMORY* (STUDI KASUS SLB-B
YAYASAN PENDIDIKAN LUAR BIASA KOTA MAKASSAR)**

Disusun dan diajukan oleh

**Sitti Rahma
D121 19 1050**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian
Studi Program Sarjana Program Studi Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Pada tanggal 8 Maret 2024
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,


Dr. Ir. Ingrid Nurtanio, M.T.
NIP 196108131988112001


Ir. Christoforus Yohannes, M.T.
NIP 196007161987021002

Ketua Program Studi,

Prof. Dr. Ir. Indrabayun, ST., MT., M.Bus.Sys., IPM, ASEAN. Eng.
NIP 197507162002121004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;
Nama : Sitti Rahma
NIM : D121 19 1050
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Sistem Penerjemah Bahasa Isyarat Indonesia (BISINDO) menggunakan Metode
Gated Recurrent Unit dan *Long Short Term Memory* (Studi Kasus SLB-B
Yayasan Pendidikan Luar Biasa Kota Makassar)

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 8 Maret 2024

Yang Menyatakan



Sitti Rahma

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT. Karena berkat rahmat, nikmat, dan pertolongan-Nya sehingga skripsi yang berjudul “**Sistem Deteksi Bahasa Isyarat Indonesia (BISINDO) menggunakan Metode *Gated Recurrent Unit* dan *Long Short Term Memory* (Studi Kasus SLB-B Yayasan Pembina Pendidikan Luar Biasa Kota Makassar)**” yang merupakan syarat penyelesaian studi Strata-1 dapat diselesaikan. Tidak lupa pula shalawat dan salam kepada Nabi Muhammad SAW. yang telah menyinari dunia ini dengan keindahan ilmu dan akhlak yang diajarkan kepada umatnya.

Dalam penelitian ini, penulis telah memberikan upaya yang sebaiknya untuk memaparkan hasil penelitian yang baik sesuai dengan metodologi yang telah penulis pelajari. Penulis berharap skripsi ini dapat menjadi landasan yang kokoh dalam pengembangan ilmu pengetahuan terlebih di bidang yang penulis teliti. Penulis menyadari selama pembuatan skripsi ini tidak lepas dari bantuan, bimbingan dan dukungan dari berbagai pihak. Untuk itu, dengan penuh kerendahan hati penulis menyampaikan terimakasih yang setulus-tulusnya dan setinggi-tingginya kepada:

1. Kedua orang tua penulis, Alm. Mandacing Dg. Pasolong dan Ibu Aminah Dg. Nialang yang selalu menjadi motivasi penulis untuk terus belajar dan menggapai cita-cita. Terimakasih untuk seluruh doa, dukungan dan pengertiannya yang telah diberikan kepada penulis.
2. Ibu Dr. Ir. Ingrid Nurtanio, S.T., M.T. selaku dosen pembimbing I dan Bapak Ir. Christoforus Yohannes, M.T selaku dosen pembimbing II penulis yang telah meluangkan waktu, tenaga pikiran dan dukungan yang luar biasa besar kepada penulis mulai dari penentuan ide sampai pada tahap penyelesaian tugas akhir ini.
3. Kak Ridwan dan Kak Fitrah, lulusan S2 dan S3 Informatika yang telah berbaik hati membagi ilmunya, menjadi tempat penulis bercerita kendala penelitian sampai pada akhirnya mendapatkan saran-saran yang membantu penulis untuk terus maju mengerjakan tugas akhir ini.

4. Kepala sekolah, guru dan staff SLB-B Yayasan Pendidikan Luar Biasa Kota Makassar yang dengan ramah menerima dan mengizinkan penulis untuk melakukan penelitian di SLB-B Kota Makassar. Adik Fadilah dan Aura yang menjadi sukarelawan penelitian, mengajari peneliti berbagai gerakan bahasa isyarat serta menjadi objek pembuatan dataset.
5. Kak Dirham Muhammad dan kak Mualif Nurindar yang mengambil peran sebagai pembimbing 3 dan pembimbing 4 penulis. Bersedia mendengarkan kendala yang penulis hadapi dan membantu menyelesaikan kendala tersebut terutama pada bagian implementasi aplikasi android.
6. Ahmad Fathanah M.Adil dan Sila Farsidia Putri yang telah membantu banyak dalam penyelesaian implementasi android. Arya Saputra adik angkatan 2020 yang sudah bersedia menjadi objek testing aplikasi.
7. Debi Rizky Ramadhana selaku teman kamar, teman kampus, sekaligus teman seperjuangan yang telah membantu banyak hal dalam pengerjaan tugas akhir peneliti.
8. Kakak-kakak anggota Laboratorium Animasi dan Multimedia yang menjadi tempat berbagi cerita suka dan duka pengerjaan tugas akhir.
9. Teman-teman Program Strata-1 Informatika Unhas angkatan 2019 yang telah memberikan dukungan, semangat, hiburan dan keceriaan-keceriaan selama pengerjaan tugas akhir ini.
10. Orang-orang baik lainnya yang penulis tidak sempat penulis tuliskan

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna dan dalam penyelesaiannya masih menemui banyak kesulitan dan hambatan sehingga memohon maaf dan tetap mengharapkan saran dan kritik untuk pengembangan lebih lanjut agar dapat memberikan manfaat yang banyak kepada seluruh pembaca.

Bone, 8 Maret 2024

Penulis

ABSTRAK

SITTI RAHMA. *Sistem Penerjemah Bahasa Isyarat Indonesia (BISINDO) menggunakan Metode Gated Recurrent Unit dan Long Short Term Memory (Studi Kasus SLB-B Yayasan Pendidikan Luar Biasa Kota Makassar)* (dibimbing oleh Ingrid Nurtanio dan Christoforus Yohannes)

Komunikasi memainkan peran penting bagi manusia untuk dapat berinteraksi dan berhubungan satu sama lainnya. Namun orang tunarungu sulit berkomunikasi menggunakan bahasa lisan. Keterbatasan yang dimiliki orang tunarungu dalam komunikasi secara lisan menjadikan bahasa isyarat sebagai alternatif bahasa sehari-hari. Namun tidak semua orang mengerti bahasa isyarat terutama orang normal. Inovasi untuk membantu komunikasi tunarungu dengan orang normal masih terbatas pada penerjemahan informasi dari orang normal untuk bisa dipahami oleh orang tunarungu, belum ada inovasi yang dikhususkan untuk menerjemahkan bahasa isyarat agar dipahami orang normal.

Penelitian ini dibuat sebagai inovasi penerjemah Bahasa Isyarat Indonesia yang menerjemahkan gerakan bahasa isyarat menjadi kata yang dapat dipahami oleh orang normal.

Penelitian ini menggunakan metode *Gated Recurrent Unit* (GRU) dan *Long Short Term Memory* (LSTM) yang dikenal mampu mengolah data *sequence* sebagaimana data gerakan bahasa isyarat yang merupakan kumpulan frame yang membentuk satu isyarat gerakan.

Penggunaan algoritma GRU menghasilkan tingkat akurasi 96,67% dalam mendeteksi bahasa isyarat, lebih besar dibandingkan algoritma LSTM yang memiliki akurasi 92,33% dengan penggunaan nilai parameter yang sama. Selain itu, GRU dapat mencapai nilai optimal pelatihan menggunakan epochs 120 sedangkan LSTM mencapai nilai optimal ketika menggunakan epochs 150 menunjukkan metode GRU lebih efisien dalam penggunaan sumber daya pelatihan dibandingkan dengan algoritma LSTM. Penggunaan dataset 30 video dan 60 video memberikan akurasi yang sama sebesar 98,33% dengan nilai loss dari 60 video lebih rendah sebesar 0,078 dibandingkan dataset 30 video sebesar 0,081. Penambahan *preprocessing* berupa ekstraksi fitur dan augmentasi citra menghasilkan nilai loss yang lebih rendah 0,047 dibandingkan tanpa penambahan *preprocessing*.

Kata Kunci: tunarungu, sistem penerjemah, Bahasa Isyarat Indonesia, GRU

ABSTRACT

SITTI RAHMA. *The Indonesian Sign Language Translator System (BISINDO) uses the Gated Recurrent Unit and Long Short Term Memory Method (Case Study of Special Education Foundation, Makassar City)* (supervised by Ingrid Nurtanio and Christoforus Yohannes)

Communication is important for humans to interact with and relate to one another. However, deaf people find it difficult to communicate using their spoken language. The limitations of deaf people in verbal communication make sign language an alternative to everyday language. However, only some people understand sign language. Innovations to help deaf people communicate with normal people are still limited to translating information from normal people so that deaf people can understand it, and there is no innovation specifically for translating sign language so that normal people can understand it.

This research was made as an innovation in the translation of Indonesian Sign Language that translates sign language gestures into words that can be understood by normal people.

This research uses Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) methods, which are known to manage sequence data as well as sign language movement data, which is a collection of frames that form one movement signal.

The use of the GRU algorithm produces an accuracy rate of 96.67% in detecting sign language, greater than the LSTM algorithm which has an accuracy of 92.33% using the same parameter values. In addition, GRU can achieve optimal training values using epochs 120 while LSTM achieve optimal values when using epochs 150 indicating the GRU method is more efficient in using training resources compared to the LSTM algorithm. The utilization of datasets comprising 30 and 60 videos yielded the same accuracy rate of 98.33%, with the 60 video dataset exhibiting a lower loss value of 0.078 compared with 0.081 for the 30 video dataset. The addition of preprocessing in the form of feature extraction and image augmentation produces a loss value of 0.047, which is lower than that obtained without additional preprocessing.

Keywords: deaf, translation system, Indonesian Sign Language, GRU

DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	Error! Bookmark not defined.
PERNYATAAN KEASLIAN.....	Error! Bookmark not defined.
KATA PENGANTAR	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
DAFTAR SINGKATAN DAN ARTI SIMBOL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian/Perancangan.....	5
1.4 Manfaat Penelitian/Perancangan.....	5
1.5 Ruang Lingkup/Asumsi Perancangan	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Bahasa Isyarat Indonesia (BISINDO).....	7
2.2 SLB-B Yayasan Pendidikan Luar Biasa Kota Makassar	8
2.3 Visi Komputer.....	10
2.4 Pemrosesan Gambar (<i>Image Processing</i>)	11
2.5 Mediapipe Holistic	16
2.6 <i>Recurrent Neural Network</i>	17
2.7 <i>Long Short Term Memory (LSTM)</i>	19
2.8 <i>Gated Recurrent Unit</i>	26
2.9 <i>Confusion Matrix</i>	34
2.10 Android	35
2.11 Black Box Testing.....	37
BAB III METODE PENELITIAN/PERANCANGAN	38
3.1 Tahapan Penelitian	38
3.2 Waktu dan Lokasi Penelitian	39
3.3 Instrumen Penelitian.....	40
3.4 Pengambilan Data	40
3.5 Perancangan Sistem	41
3.6 Validasi Model.....	47
3.7 Analisis Kinerja Model	50
3.8 Implementasi Model ke dalam Sistem Berbasis Android	51
3.9 Skenario Pelatihan dan Pengujian.....	52
BAB IV HASIL DAN PEMBAHASAN	56
Hasil	56
4.2 Pembahasan.....	77
BAB V PENUTUP.....	81
5.1 Kesimpulan	81
5.2 Saran.....	82
DAFTAR PUSTAKA	83

DAFTAR GAMBAR

Gambar 1 Fitur dari mediapipe	16
Gambar 2 Loop RNN tertutup.....	18
Gambar 3 Loop RNN terbuka.....	18
Gambar 4 Cara kerja algoritma LSTM	20
Gambar 5 Skema gerbang <i>forget</i> LSTM.....	20
Gambar 6 Skema gerbang <i>input</i> LSTM	22
Gambar 7 Skema gerbang <i>output</i> LSTM	24
Gambar 8 Struktur dari model dasar GRU.....	27
Gambar 9 Cara kerja algoritma GRU.....	28
Gambar 10 Skema gerbang <i>update</i> GRU.....	29
Gambar 11 Skema gerbang reset GRU	31
Gambar 12 Tabel <i>confusion matrix</i>	34
Gambar 13 Diagram tahapan penelitian.....	38
Gambar 14 Relawan pengambilan dataset	41
Gambar 15 Diagram alir perancangan sistem	41
Gambar 16 Tampilan folder kata pada VSC dan file explorer.....	42
Gambar 17 Tampilan daftar video dalam kata “Apa”.....	43
Gambar 18 Contoh <i>frame</i> dari kata “Siapa”	44
Gambar 19 Deteksi <i>keypoints</i> pose dan tangan.....	45
Gambar 20 Hasil ekstraksi <i>keypoints</i>	45
Gambar 21 Nilai vektor frame hasil ekstraksi.....	46
Gambar 22 Validasi model mendeteksi kata “Guru”	47
Gambar 23 Validasi model mendeteksi kata “Ingat”	48
Gambar 24 <i>True positive</i> label “Bapak”	49
Gambar 25 <i>False positive</i> label “Bapak”	49
Gambar 26 <i>False Negative</i> label “Bapak”	50
Gambar 27 <i>True Negative</i> label “Bapak”.....	50
Gambar 28 Alur klasifikasi label di Android.....	52
Gambar 29 <i>Training</i> model menggunakan algoritma GRU.....	57
Gambar 30 <i>Training</i> model menggunakan algoritma LSTM	58
Gambar 31 <i>Training</i> model menggunakan dataset 60 video	60
Gambar 32 <i>Training</i> model dengan penambahan ekstraksi fitur	61
Gambar 33 <i>Homepage</i> Aplikasi Isyarat Kita	73
Gambar 34 Tutorial Rekam dan Tulis Aplikasi	74
Gambar 35 Fitur Rekam Aplikasi	75
Gambar 36 Fitur Tulis Aplikasi	76

DAFTAR TABEL

Tabel 1 Skenario pelatihan model <i>machine learning</i>	53
Tabel 2 Skenario validasi manual model <i>machine learning</i>	53
Tabel 3 Skenario pengujian <i>black box</i>	54
Tabel 4 Perbandingan proses <i>training</i> algoritma GRU.....	57
Tabel 5 Perbandingan proses <i>training</i> algoritma LSTM.....	58
Tabel 6 Perbandingan hasil <i>training</i> menggunakan metode GRU dan LSTM	59
Tabel 7 Hasil <i>training</i> menggunakan dataset 60 video	59
Tabel 8 Perbandingan hasil <i>training</i> model 60 video dan 30 video.....	60
Tabel 9 Hasil <i>training</i> dengan penambahan proses ekstraksi fitur	60
Tabel 10 Perbandingan hasil <i>training</i> model tanpa dan dengan tambahan <i>preprocessing</i>	62
Tabel 11 Hasil validasi langsung model 20 kata tanpa tambahan <i>preprocessing</i>	63
Tabel 12 Hasil perhitungan <i>confusion matrix</i> model 20 kata tanpa tambahan ekstraksi fitur dan augmentasi citra.....	65
Tabel 14 Hasil validasi langsung model 20 kata dengan tambahan <i>preprocessing</i>	66
Tabel 15 Hasil perhitungan <i>confusion matrix</i> model 20 kata dengan tambahan ekstraksi fitur dan augmentasi citra.....	68
Tabel 16 Metrik kinerja sistem tiap kategori model 20 kata dengan tambahan <i>preprocessing</i>	69
Tabel 17 Validasi model oleh siswa SLB-B Pendidikan Luar Biasa Makassar ...	70
Tabel 18 Metrik kinerja sistem tiap kategori untuk validasi menggunakan objek siswa SLB-B.....	71
Tabel 19 Rata-rata waktu pendeteksian tiap kata.....	72
Tabel 20 Hasil pengujian <i>block box</i> aplikasi.....	77
Tabel 21 Rangkuman hasil pengujian dan validasi sistem dari berbagai skenario	78
Tabel 22 Rata-rata hasil validasi model menggunakan data <i>realtime</i>	79

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
BGR	<i>Blue Green Red</i>
BISINDO	Bahasa Isyarat Indonesia
GERKATIN	Gerakan Kesejahteraan Tunarungun Indonesia
GRU	<i>Gated Recurrent Unit</i>
LSTM	<i>Long Short Term Memory</i>
RGB	<i>Red Green Blue</i>
RNN	<i>Recurrent Neural Network</i>
ROI	<i>Region of Interest</i>
SIBI	Sistem Bahasa Isyarat Indonesia
SLB	Sekolah Luar Biasa
SLR	<i>Single Lens Reflect</i>
σ	Fungsi Aktivasi Sigmoid
\odot	<i>Element-wise Multiplication</i>

BAB I

PENDAHULUAN

1.1 Latar Belakang

Komunikasi merupakan elemen dalam mewujudkan interaksi serta hubungan antar individu. Melalui proses komunikasi, individu dapat menyampaikan berbagai pemikiran mereka kepada orang lain, sehingga menciptakan interpretasi bersama dari pesan yang disampaikan. Proses komunikasi dapat terwujud ketika pelaku komunikasi memiliki pemahaman arti bahasa yang sama. Komunikasi dilakukan oleh siapa saja, dalam bentuk komunikasi verbal maupun komunikasi non verbal. Komunikasi verbal merujuk pada proses komunikasi yang melibatkan penggunaan bahasa secara lisan atau tulisan. Di sisi lain, komunikasi nonverbal melibatkan segala bentuk komunikasi yang terjadi tanpa menggunakan kata-kata secara langsung. Ini dapat mencakup penggunaan isyarat, ekspresi wajah, gerak tubuh, intonasi suara, dan gaya bahasa sebagai sarana ekspresi dan pemahaman pesan tanpa tergantung pada kata-kata yang diucapkan atau ditulis (Asiyah, 2018).

Umumnya bahasa yang digunakan merupakan bahasa verbal. Namun bagi penderita tunarungu atau yang lebih akrab disapa Teman Tuli, komunikasi verbal yang melibatkan bahasa lisan menjadi sulit dilakukan. Menurut Nofiarahmah, F. (2018) Teman Tuli adalah orang yang kehilangan kemampuan mendengar sehingga menghambat proses informasi bahasa melalui pendengarannya, baik memakai ataupun tidak memakai alat bantu dengar dimana batas pendengaran yang dimilikinya cukup memungkinkan keberhasilan proses informasi bahasa melalui pendengaran. Dalam menangani situasi ini, kehadiran bahasa isyarat berfungsi sebagai opsi yang mendukung komunikasi antara individu yang mengalami kesulitan pendengaran. Bahasa isyarat merupakan sistem komunikasi yang memanfaatkan bahasa tubuh, gerakan bibir, dan komunikasi manual. Bahasa isyarat terdiri dari kombinasi bentuk dan gerakan tangan, lengan, tubuh dan ekspresi wajah. Keberadaan bahasa isyarat menjadi alternatif untuk menyampaikan pesan yang jelas dan baik dalam interaksi antar

sesama Teman Tuli maupun komunikasi dengan individu yang tidak mengalami kesulitan pendengaran (Midyanti, D.M., 2020).

Dua bentuk bahasa isyarat yang digunakan oleh komunitas Teman Tuli di Indonesia yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). Dikutip dari pengantar Kamus SIBI, Sistem Isyarat Bahasa Indonesia (SIBI) merupakan salah satu media yang membantu komunikasi sesama Teman Tuli di dalam masyarakat yang lebih luas. Wujudnya berupa tatanan sistematis yang melibatkan isyarat jari, tangan dan berbagai gerakan yang mencerminkan suatu bentuk ekspresi terstruktur untuk menyampaikan kosakata dalam bahasa Indonesia. Sedangkan BISINDO merupakan bahasa yang didorong oleh Gerakan Kesejahteraan Tunarungu Indonesia (GERKATIN) dan mengalami pengembangan oleh masyarakat tunarungu itu sendiri. Akibatnya, variasi dalam bentuk “dialek” BISINDO muncul di berbagai daerah. BISINDO dikomunikasikan melalui gerakan dua tangan sebagai saran untuk menyampaikan pesan dan memperkaya komunikasi di antara individu tunarungu (Rohmat dan Bentar, 2018). Untuk mempelajari bahasa isyarat dapat dilakukan melalui beberapa media seperti buku atau video bahasa isyarat.

Kemajuan teknologi memungkinkan masyarakat yang sebelumnya tidak mengerti bahasa isyarat dapat belajar untuk mengenali bahasa isyarat dengan sebuah sistem yang dapat menerjemahkan bahasa isyarat. Saat ini telah bermunculan penelitian tentang penerjemah bahasa isyarat seperti penerjemah bahasa isyarat angka dan huruf alphabet. Akan tetapi masih kurang penelitian yang menerjemahkan bahasa isyarat berupa kata. Terlebih lagi untuk bahasa isyarat BISINDO yang memiliki gerakan berbeda di tiap daerah menjadi salah satu kendala dalam pengembangan sistem penerjemah bahasa isyarat. Salah satu sekolah SLB-B di Kota Makassar yaitu SLB-B Yayasan Pembina Pendidikan Luar Biasa Makassar menunjukkan keprihatinan melihat siswa kesulitan berkomunikasi dengan orang normal. Setelah melakukan wawancara bersama kepala sekolah SLB-B Yayasan Pembina Pendidikan Luar Biasa, beliau mengatakan “Siswa ketika di sekolah tidak mengalami kendala besar ketika berkomunikasi dengan guru karena guru telah dilatih menggunakan bahasa isyarat. Yang menjadi masalah ketika mereka keluar sekolah dan bertemu orang

normal. Mereka kesulitan untuk menyapa karena orang normal tidak paham bahasa isyarat. Bahkan orangtua siswa kadang tidak paham dengan bahasa mereka sehingga guru yang menjelaskan maksud siswa tersebut kepada orangtuanya”. Beliau juga menjelaskan, saat ini Telkom telah mensosialisasikan aplikasi I-Chat video bahasa isyarat. Namun hal ini masih jarang digunakan karena dinilai kurang efektif karena orang normal perlu menonton dan mempelajari semua video tersebut untuk mengetahui arti dari gerakan isyarat yang dilakukan oleh Teman Tuli.

Selain itu, Andi Kadri Unru selaku bagian dari GERKATIN (organisasi tunarungu nasional) menjelaskan bahwa saat ini aplikasi yang sering digunakan Teman Tuli adalah *Hear Me* dan *Live Transcribe*. Aplikasi *Hear Me* menerima masukan suara dan tulisan kemudian mengolah masukan tersebut menjadi video animasi 3D bahasa isyarat. Sedangkan aplikasi *Live Transcribe* menggunakan teknologi pengenalan suara otomatis dengan mentranskripsikan pembicaraan menjadi tulisan secara *real-time*. Hanya saja kedua aplikasi ini hanya menerima masukan dari orang normal kemudian mengubah masukan menjadi teks atau bahasa isyarat agar dipahami oleh Teman Tuli, tidak ada fitur yang bekerja sebaliknya yang menerjemahkan bahasa isyarat agar dipahami oleh orang normal.

Beberapa penelitian telah dilakukan untuk mendeteksi bahasa isyarat menjadi teks. Penelitian terkait deteksi bahasa isyarat dijadikan dasar pengembangan dalam penelitian ini. Diantara penelitian tersebut yakni Devina Yolanda yang berjudul “*Pengenalan Alfabet Bahasa Isyarat Tangan Secara Real-Time dengan menggunakan Metode Convolutional Neural Network*” (Yolanda, et. al., 2020) dimana penelitian ini membutuhkan waktu 2 menit 44 detik yang merupakan video dengan durasi 1 detik dengan 30 frame. Selanjutnya penelitian yang berjudul “*Pengenalan Bahasa Isyarat Indonesia (SIBI) Menggunakan Leap Motion Controller (LMC) dan Algoritma Data Mining Naïve Bayes*” (Ridwang, 2017), dan penelitian yang berjudul “*Implementasi Penerjemah Bahasa Isyarat Pada Bahasa Isyarat Indonesia (BISINDO) Dengan Metode Principal Component Analysis (PCA)*” (Rohmat, et. al., 2018) dengan keluaran penerjemah 26 isyarat huruf. Hanya saja ketiga penelitian tersebut hanya mendeteksi bahasa isyarat alphabet yang bersifat statis.

Penelitian terkait yang mendeteksi bahasa isyarat kata pernah dilakukan oleh (Pratamasunu et al., 2022) yang berjudul “*Deteksi Tangan Otomatis Pada Video Percakapan Bahasa Isyarat Indonesia Menggunakan Metode Deep Gated Recurrent Unit (GRU)*” untuk mengenali 3 kelas yaitu ucapan hallo, terimakasih, dan sama-sama yang merupakan gerakan dari Bahasa Isyarat Indonesia (BISINDO). Penelitian ini menghasilkan akurasi sebesar 88%. Namun sistem hanya mendeteksi 3 kata bahasa isyarat. Selain itu terdapat penelitian yang dilakukan oleh (Paisal, 2023) berjudul “*Pengembangan Sistem Komunikasi bagi Penyandang Tunarungu menggunakan Mediapipe dan Metode Gated Recurrent Unit*” dengan kata deteksi berupa kata SIBI dengan tingkat akurasi sistem sebesar 98%. Namun penelitian ini berfokus pada kata imbuhan yang ada di dalam kamus SIBI.

Dari penelitian di atas, dapat dilihat bahwa metode Gated Recurrent Unit memiliki tingkat akurasi pendeteksian yang tinggi pada data *image* video. Terlihat juga perbedaan dari kedua penelitian yang menggunakan GRU yaitu ketika diberikan penambahan *library mediapipe* meningkatkan akurasi deteksi sampai 10%. Adapun penelitian terakhir yang dilakukan oleh Paisal masih mendeteksi kata imbuhan, belum sampai pada pendeteksian bahasa isyarat berupa kalimat.

Berdasarkan uraian permasalahan di atas, penulis mengembangkan sistem penerjemah Bahasa Isyarat Indonesia (BISINDO) dalam bentuk kata yang sering digunakan oleh siswa SLB-B Yayasan Pembina Pendidikan Luar Biasa Kota Makassar menggunakan Metode *Gated Recurrent Unit* dan *Long Short Term Memory* dengan tambahan *library mediapipe* yang diimplementasikan ke bentuk aplikasi android dengan keluaran berupa teks kalimat hasil terjemahan. Oleh karena itu, penulis mengangkat judul “*Sistem Penerjemah Bahasa Isyarat Indonesia (BISINDO) menggunakan Metode Gated Recurrent Unit dan Long Short Term Memory (Studi Kasus SLB-B Yayasan Pembina Pendidikan Luar Biasa Kota Makassar)*”.

1.2 Rumusan Masalah

Adapun rumusan masalah yang menjadi dasar penelitian yaitu:

1. Bagaimana merancang sebuah sistem yang dapat mendeteksi kata Bahasa Isyarat Indonesia (BISINDO) menggunakan metode GRU dan LSTM?
2. Bagaimana kinerja sistem dalam menerjemahkan Bahasa Isyarat Indonesia (BISINDO) menggunakan metode GRU dan LSTM?

1.3 Tujuan Penelitian/Perancangan

Berdasarkan rumusan masalah yang diuraikan, maka tujuan penelitian ini, yaitu:

1. Untuk mengetahui rancangan sistem yang dapat mendeteksi kata Bahasa Isyarat Indonesia (BISINDO) menggunakan metode GRU dan LSTM
2. Untuk mengetahui kinerja sistem dalam menerjemahkan Bahasa Isyarat Indonesia (BISINDO) menggunakan metode GRU dan LSTM

1.4 Manfaat Penelitian/Perancangan

Manfaat penelitian ini yaitu:

1. Membantu memperbaiki teknologi pengenalan Bahasa Isyarat Indonesia (BISINDO) dengan penambahan referensi berupa model dengan metode GRU dan LSTM
2. Memberikan data terkait kinerja sistem ketika menggunakan metode GRU dan LSTM dalam mendeteksi pola bahasa isyarat yang dapat digunakan sebagai referensi penggunaan model dalam pengembangan berikutnya
3. Membantu masyarakat normal dan tunarungu untuk saling berkomunikasi dengan penyediaan media penerjemah bahasa isyarat

1.5 Ruang Lingkup/Asumsi Perancangan

1. Data latih yang digunakan berupa data gerakan dan pose yang diambil dengan menggunakan kamera *Webcam*.
2. Kalimat yang diterjemahkan adalah kalimat BISINDO.
3. Objek penelitian adalah siswa SLB-B Yayasan Pembina Pendidikan Luar Biasa Makassar.

4. Jarak pengambilan data latih dan data uji adalah 1 – 2 meter dari kamera untuk mendapatkan hasil yang akurat.
5. Bagian pose yang diambil adalah bagian perut ke atas.

BAB II TINJAUAN PUSTAKA

2.1 Bahasa Isyarat Indonesia (BISINDO)

Bahasa isyarat merupakan sistem komunikasi yang menggunakan bahasa tubuh, gerak bibir serta ekspresi manual. Individu yang tidak mengalami gangguan pendengaran mengalami kesulitan berkomunikasi dengan penderita tunarungu tanpa menggunakan bahasa isyarat (Midyanti, D.M., 2020). Perbedaan mendasar antara bahasa lisan dengan bahasa isyarat terletak pada modalitas atau sarana produksi dan persepsi yang digunakan. Bahasa isyarat diciptakan melalui gerakan tangan (*gesture*) dan dipersepsikan melalui visual, sementara bahasa lisan dihasilkan melalui alat lisan dan dipersepsikan melalui alat pendengaran. Salah satu metode komunikasi yang efektif adalah melalui penggunaan gestur tangan, ekspresi wajah, dan gerakan tubuh untuk merepresentasikan kata atau huruf dalam bahasa isyarat. Bahasa dan gerakan tubuh juga berfungsi sebagai identitas bagi pemakainya (Ridwang, 2022).

Ada dua jenis bahasa isyarat yang digunakan oleh komunitas tunarungu di Indonesia yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). Dikutip dari pengantar Kamus SIBI, Sistem Isyarat Bahasa Indonesia (SIBI) merupakan salah satu media yang membantu komunikasi sesama kaum tunarungu di dalam masyarakat yang lebih luas. Wujudnya adalah tataan yang sistematis tentang seperangkat isyarat jari, tangan, dan berbagai gerak yang melambangkan kosa kata bahasa Indonesia. Sementara itu, BISINDO (Bahasa Isyarat Indonesia) adalah bahasa yang dibuat oleh Gerakan Kesejahteraan Tunarungu Indonesia (GERKATIN) dan dipelihara dan dikembangkan oleh masyarakat tunarungu sendiri. Inilah sebabnya, BISINDO memiliki variasi “dialek” yang berbeda di berbagai daerah. BISINDO dikomunikasikan melalui gerakan kedua tangan (Borman & Priyopradono, 2018). Salah satu perbedaan yang paling mendasar antara BISINDO dan SIBI adalah penggunaan afiks atau imbuhan dalam isyarat SIBI, sedangkan BISINDO tidak melibatkan afiks/imbuhan. Aksara dalam SIBI mengikuti aturan penulisan dan ejaan bahasa Indonesia secara ketat, sementara aksara BISINDO tanda-tanda lingustiknya

dibentuk oleh komunitas tunarungu sendiri, sehingga tidak selalu mengikuti kaidah tata bahasa Indonesia yang resmi.

Selain itu perbedaan lain antara SIBI dan BISINDO terletak pada prosedur bahasanya. BISINDO yang digunakan oleh teman tunarungu dalam kehidupan sehari-hari mengandung kosakata simbolis. Dalam BISINDO kosakata tertulis yang sederhana sudah memadai selama maknanya sudah tersampaikan. Di sisi lain, SIBI didirikan dengan tujuan untuk mengajarkan sistem bahasa Indonesia kepada teman tunarungu. Aturan penggunaan kosakata simbolik dalam SIBI lebih kompleks dan mengikuti tahap-tahap tertentu, termasuk penggunaan awalan dan akhiran yang diadaptasi dari ejaan bahasa Indonesia untuk setiap kata. Hal ini membuat bahasa SIBI terasa lebih sulit untuk dipelajari sehingga banyak teman tuli lebih memilih menggunakan BISINDO dalam komunikasi sehari-hari.

2.2 SLB-B Yayasan Pendidikan Luar Biasa Kota Makassar

Sekolah Luar Biasa (SLB) B adalah lembaga pendidikan yang ditujukan khusus untuk anak-anak dengan gangguan pendengaran, yaitu anak-anak yang mengalami hambatan dalam indra pendengarannya. Pendidikan dianggap sebagai hak yang harus diberikan kepada setiap anak, termasuk anak-anak dengan kebutuhan khusus. Hak untuk menerima fasilitas pendidikan ini bertujuan untuk mengembangkan potensi yang dimiliki oleh setiap anak tanpa pengecualian. Pemerintah mewujudkan hak pendidikan bagi anak berkebutuhan khusus dengan mendirikan sekolah luar biasa dan memilih tenaga pendidik profesional dari pendidikan luar biasa. Penyelenggaraan pendidikan ini diatur oleh Peraturan Pemerintah Republik Indonesia Nomor 17 Tahun 2010, khususnya dalam Pasal 127 yang berkaitan dengan pengelolaan dan penyelenggaraan pendidikan.

Sebagaimana yang dijelaskan oleh Atmaja (2018), anak-anak berkebutuhan khusus yang mengalami gangguan atau kelainan, baik itu fisik, mental, intelektual, atau emosi, memiliki hak untuk menerima pendidikan dan pembelajaran yang disesuaikan dengan kebutuhan mereka. Salah satu kategori anak berkebutuhan khusus yang mengalami gangguan fisik adalah anak-anak tunarungu, yang mengalami kelainan pada indera pendengaran mereka

(Permatasari et al., 2019). Tunarungu adalah sebutan untuk anak yang mengalami hilangnya sebagian atau seluruh kemampuan pendengarannya. Secara fisik, anak tunarungu mungkin tidak terlihat berbeda dengan anak-anak lain pada umumnya. Namun, ketika berkomunikasi akan terlihat bahwa anak tersebut mengalami gangguan dalam fungsi pendengarannya. Gangguan ini menyebabkan mereka tidak dapat menangkap informasi secara optimal yang menghambat perkembangan kemampuan berbahasa mereka.

Sekolah Luar Biasa (SLB) Bagian B Yayasan Pembina Pendidikan Luar Biasa Kota Makassar telah berdiri sejak tahun 1966 dan berlokasi di Jalan Cendrawasih I No 226 A. Sekolah ini merupakan bagian dari Yayasan Pembina Pendidikan Luar Biasa (YPPLB) Kota Makassar dan terletak di Kecamatan Mariso, Kota Makassar. Fokus utama dari SLB Bagian B ini adalah menyelenggarakan pendidikan khusus bagi anak-anak yang memiliki kebutuhan khusus (Anak Berkebutuhan Khusus - ABK) terutama yang mengalami gangguan pendengaran, baik yang bersifat permanen maupun tidak permanen, yaitu tunarungu. SLB B ini adalah lembaga pendidikan swasta yang memiliki peran penting dalam mengembangkan bakat dan potensi yang dimiliki oleh anak-anak atau murid tunarungu. Hal ini mencakup aspek pembelajaran, kegiatan ekstrakurikuler, dan pengembangan interaksi sosial mereka. Dalam proses pembelajaran, SLB Bagian B mencoba untuk menciptakan lingkungan yang serupa dengan sekolah-sekolah pada umumnya. Mereka menyediakan berbagai bidang pembelajaran, baik yang bersifat umum maupun yang lebih spesifik sesuai dengan kebutuhan para muridnya.

Metode yang digunakan dalam proses belajar mengajar di SLB-B Yayasan Pembina Pendidikan Luar Biasa Kota Makassar untuk membina anak tunarungu adalah penggunaan bahasa isyarat dengan bantuan bahasa oral. Bahasa isyarat digunakan sebagai media utama komunikasi, tetapi jika ada situasi di mana penggunaan bahasa isyarat tidak dapat sepenuhnya dimengerti oleh siswa, maka bahasa oral digunakan sebagai suplemen. Penggunaan bahasa oral berguna untuk menjelaskan atau melengkapi informasi yang disampaikan dalam bahasa isyarat siswa (Zulfadhli, R., 2018). Proses pembinaan anak tunarungu di SLB-B Yayasan

Pembina Pendidikan Luar Biasa ini juga menghadapi beberapa hambatan, di antaranya:

1. Perubahan suasana hati siswa: Anak tunarungu sering mengalami perubahan suasana hati yang dapat memengaruhi keinginan mereka untuk belajar. Guru harus sensitif terhadap perubahan ini dan berusaha untuk memahami dan membujuk anak agar mau belajar tanpa memaksa.
2. Kurangnya minat belajar: Terkadang minat siswa terhadap pembelajaran dapat berubah-ubah. Hal ini dapat terjadi jika siswa lebih tertarik pada sesuatu yang lain selama pembelajaran. Pendidik perlu mencari cara untuk mempertahankan minat siswa dalam belajar.
3. Keterbatasan media dan alat peraga: Media bantu dan alat peraga memegang peran penting dalam pembelajaran anak tunarungu karena dapat membantu menjelaskan konsep-konsep dengan lebih jelas dan menarik perhatian siswa. Keterbatasan dalam hal ini dapat menghambat efektivitas proses belajar mengajar

2.3 Visi Komputer

Visi Komputer juga dikenal sebagai *computer vision*, merupakan sub-disiplin ilmu dalam bidang Kecerdasan Buatan yang berfokus pada studi tentang bagaimana mesin dapat mengenali objek yang terlihat atau diamati. Ilmu ini melibatkan pengembangan teori-teori dan algoritma untuk mengekstraksi dan menganalisis informasi yang berguna dari berbagai jenis citra atau data visual yang diperoleh melalui perangkat komputer. Secara prinsip, *computer vision* berupaya untuk meniru proses pengenalan objek yang dilakukan oleh manusia. Pada dasarnya, dalam proses penglihatan manusia, indra penglihatan (mata) digunakan untuk melihat objek, dan citra objek tersebut kemudian diteruskan ke otak untuk diinterpretasikan. Interpretasi ini memungkinkan manusia untuk memahami objek yang terlihat. Hasil dari interpretasi ini dapat digunakan untuk pengambilan keputusan. Dengan demikian, visi komputer bertujuan untuk mengajarkan komputer cara "melihat" dan "memahami" dunia visual seperti yang dilakukan oleh manusia. Ini melibatkan penggunaan teknik-teknik seperti pemrosesan gambar, pengenalan pola, dan pembelajaran mesin yang

memungkinkan komputer untuk mengenali objek, mengklasifikasikan citra, atau bahkan mengambil tindakan berdasarkan data visual yang diberikan (Munir & Yuliani, 2019).

Kemampuan untuk mengenali objek atau pola pada citra dalam bidang *computer vision* adalah hasil dari kombinasi antara pengolahan citra yang melibatkan perbaikan atau penyempurnaan citra dengan pengenalan pola yang merupakan proses identifikasi objek atau pola dalam citra. Proses-proses dalam *computer vision* dapat dibagi menjadi (Basuki & Yuliani, 2019):

- a. Proses mengakuisisi citra digital (*image acquisition*)
- b. Proses pengolahan citra (*image processing*)
- c. Proses analisis data citra (*image analysis*)
- d. Proses pemahaman data citra (*image understanding*)

Menurut Boyle dan Thomas, *computer vision* tidak hanya mencakup pengenalan, tetapi juga melibatkan operasi “*Low Level Processing*” yang mencakup algoritma *image processing* yang murni. Selain itu, mereka mengklasifikasikan pengolahan citra sebagai bagian dari *computer vision*. Dengan kata lain, Computer Vision mencakup lebih dari sekedar pengenalan objek atau pola pada citra, termasuk melibatkan pengolahan citra dalam bentuk operasi tingkat rendah yang dapat meningkatkan pemahaman computer terhadap informasi visual. Ini menunjukkan bahwa *computer vision* adalah bidang yang lebih luas yang mencakup berbagai aspek dalam analisis dan interpretasi data visual (Boyle & Thomas, 1988).

2.4 Pemrosesan Gambar (*Image Processing*)

Oleh (Jumadi et al., 2021) mengemukakan bahwa citra adalah representasi (citra), kemiripan, atau tiruan dari suatu objek. Dilihat dari pembentukan gambarnya, terdiri dari dua jenis:

- a) Citra Analog

Gambar kontinu merupakan hasil dari kamera analog dan sistem optik yang menerima sinyal analog. Ini termasuk sinyal video, gambar di layar

televisi, sinar-X, foto di atas kertas fotografi, lukisan, dan gambar dalam format kaset.

b) Citra Digital atau Citra Diskrit

Citra yang berasal dari proses digitalisasi citra kontinu dapat berbentuk citra digital atau citra diskrit, yang dapat diolah oleh komputer. Contoh perangkat yang melakukan digitalisasi citra ini termasuk kamera digital dan pemindai. Citra digital adalah representasi numerik dari citra dua dimensi yang bisa diproses oleh komputer. Nilai numerik ini biasanya disimpan dalam format biner 8 bit dan diatur dalam elemen citra yang disebut piksel (D.Putra, 2010). Setiap citra digital direpresentasikan sebagai matriks dengan M baris dan N kolom, di mana perpotongan baris dan kolom ini disebut piksel. Piksel memiliki dua parameter koordinat serta nilai intensitas atau warna (Irwan & Sianipar, 2018). Nilai yang ditemukan pada koordinat (x,y) disebut sebagai $f(x,y)$ yang merepresentasikan intensitas atau warna piksel pada titik tersebut (Puspitasari, 2019).

2.4.1 Preprocessing

Preprocessing merupakan tahap awal dalam pengolahan citra yang bertujuan untuk menghilangkan elemen-elemen yang tidak diperlukan pada gambar masukan, sehingga dapat meningkatkan kualitas citra. Tujuan utama dari *preprocessing* adalah untuk mempermudah dan mempercepat kinerja sistem dalam mengenali pola citra (Miranda et al., 2020). Dalam upaya mengambil informasi yang relevan dari piksel yang terdapat pada gambar, setiap komponen dalam sistem pengenalan karakter dirancang untuk mengurangi jumlah data yang harus diproses.

2.4.2 Normalisasi Data

Normalisasi data adalah tahap yang sangat esensial dalam proses *preprocessing image* pada pengolahan citra atau pemrosesan gambar. Dalam analisis data, seringkali kita dihadapkan pada beberapa atribut yang memiliki skala yang berbeda. Akibatnya, atribut dengan nilai yang tinggi dapat memiliki pengaruh yang lebih signifikan dibandingkan dengan atribut yang memiliki nilai

yang lebih rendah. Oleh karena itu, normalisasi atribut diperlukan untuk menjadikan atribut-atribut ini memiliki skala yang seragam (Herlinda & Darwis, 2021). Berikut beberapa manfaat dari normalisasi data:

1. Meningkatkan Konsistensi

Normalisasi membantu memastikan bahwa seluruh data gambar memiliki rentang nilai yang seragam. Hal ini membantu mengurangi variabilitas yang tidak relevan dalam data.

2. Mempercepat Konvergensi

Normalisasi dapat mempercepat proses konvergensi algoritma karena data yang telah dinormalisasi memiliki skala yang serupa, yang membuat proses pembelajaran menjadi lebih efisien.

3. Mencegah Dominasi Fitur

Normalisasi juga mencegah fitur-fitur dengan skala yang lebih besar mendominasi proses pembelajaran. Sebagai contoh, dalam gambar berwarna, jika satu kanal (misalnya R, G, atau B) memiliki nilai yang jauh lebih tinggi daripada yang lain, maka algoritma cenderung lebih fokus pada kanal tersebut. Normalisasi memastikan bahwa semua kanal memiliki pengaruh yang seimbang pada model.

4. Persiapan Data untuk Algoritma Khusus

Beberapa algoritma, seperti yang digunakan dalam *deep learning*, memerlukan input yang sudah dinormalisasi ke dalam rentang tertentu (misalnya 0 hingga 1 atau -1 hingga 1). Oleh karena itu, normalisasi data dalam *preprocessing* gambar sangat penting untuk memastikan bahwa semua fitur yang relevan dapat diproses secara efisien dan adil oleh algoritma yang digunakan.

2.4.3 Flipping Image

Flipping image, atau yang dikenal sebagai pemutaran gambar, adalah teknik augmentasi data yang sering digunakan dalam pengolahan citra dan *deep learning*. Proses ini melibatkan tindakan membalikkan gambar secara

horizontal (*horizontal flipping*) atau vertikal (*vertical flipping*). Ketika melakukan *horizontal flipping*, gambar akan terbalik secara horizontal, seperti jika gambar tersebut dipantulkan secara horizontal. Sementara itu, *vertical flipping* menghasilkan gambar yang terbalik secara vertikal. Teknik *flipping image* digunakan untuk menciptakan variasi data yang lebih banyak dari satu gambar asli. Tujuannya adalah untuk melatih model agar lebih tangguh dan menghindari *overfitting*. Jika gambar berisi objek yang dapat muncul dalam berbagai orientasi, melakukan *flipping image* membantu model untuk belajar dengan cara yang tidak dipengaruhi oleh perubahan orientasi. *Flipping image* adalah salah satu bentuk augmentasi data yang sederhana namun sangat efektif, karena membantu meningkatkan jumlah data pelatihan tanpa perlu mengumpulkan lebih banyak data secara fisik.

2.4.4 Ekstraksi Fitur

Ekstraksi fitur adalah proses transformasi data awal atau data dengan dimensi tinggi menjadi bentuk representasi fitur yang lebih ringkas dan informatif. Tujuan utama dari ekstraksi fitur adalah untuk mengenali dan mengambil atribut-atribut yang relevan dari data agar lebih mudah dimengerti oleh algoritma pembelajaran mesin atau digunakan dalam tugas analisis data. Dalam konteks ekstraksi fitur, *Principal Component Analysis* (PCA) merupakan salah satu metode statistik utama yang diaplikasikan untuk mereduksi dimensi data dengan tetap mempertahankan atribut-atribut penting dari representasi aslinya (Hasym & Susilawati, 2021)

Principal Component Analysis (PCA) adalah metode statistik yang digunakan untuk mereduksi data berdimensi tinggi menjadi dimensi yang lebih rendah melalui ekstraksi fitur. PCA melakukan transformasi linier ortogonal ke sistem koordinat baru, di mana variabilitas terbesar ditemukan pada koordinat pertama (dikenal sebagai komponen utama pertama), dan variabilitas kedua terbesar pada koordinat kedua, begitu seterusnya. Proses ini melibatkan perhitungan standar deviasi, matriks kovariansi, nilai eigen, dan vektor eigen. PCA bisa berbasis metode kovariansi atau korelasi, dan seringkali data dinormalisasi terlebih dahulu untuk mendekati distribusi normal standar.

Metode ini umum digunakan untuk mengidentifikasi pola dalam data atau sampel berdimensi besar dan menyajikannya dalam cara yang memudahkan identifikasi kesamaan dan perbedaan dalam data (Unihehu & Suharjo, 2021).

PCA memerlukan data masukan yang memiliki karakteristik zero-mean pada setiap fiturnya. Karakteristik ini dapat dicapai dengan mengurangi setiap nilai dalam data dengan rata-rata fitur tersebut. Jika kita memiliki sebuah dataset X dengan ukuran $M \times N$, di mana M adalah jumlah baris dan N adalah jumlah kolom, maka dataset tersebut akan diolah sedemikian rupa.

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{M1} & \cdots & x_{MN} \end{bmatrix}$$

Untuk fitur ke- j , semua nilai pada kolom tersebut dikurangi dengan rata-ratanya, diformulasikan menggunakan persamaan 1.

$$x'_{ij} = x_{ij} - \bar{x}_j \quad (1)$$

dengan:

$$i = 1, 2, \dots, m$$

$$j = 1, 2, \dots, m$$

\bar{x}_j = nilai rata-raya kolom ke- j

Selanjutnya dilakukan proses untuk mendapatkan matriks kovarian dari matriks X' , yaitu C , dengan menggunakan formula pada persamaan 2.

$$C = \frac{1}{m} x' \cdot x'^t \quad (2)$$

x'^t adalah matriks transpose dari x' . Pada matriks C , elemen ke- ij adalah hasil kali elemen baris matriks x' dengan kolom matriks x'^t . Sifat-sifat yang dimiliki oleh matriks C adalah sebagai berikut:

1. C adalah matriks simetris bujur sangkar berukuran $m \times m$.
2. Bagian diagonal utama (dari kiri atas ke kanan bawah) adalah nilai varian masing-masing fitur sesuai dengan indeks kolomnya.
3. Bagian selain diagonal utama adalah kovarian di antara pasangan dua fitur yang berkesesuaian.

Jadi, matriks C merepresentasikan kovarian di antara semua pasangan yang mungkin dari fitur data set matriks x' . Nilai kovarian merefleksikan *noise*

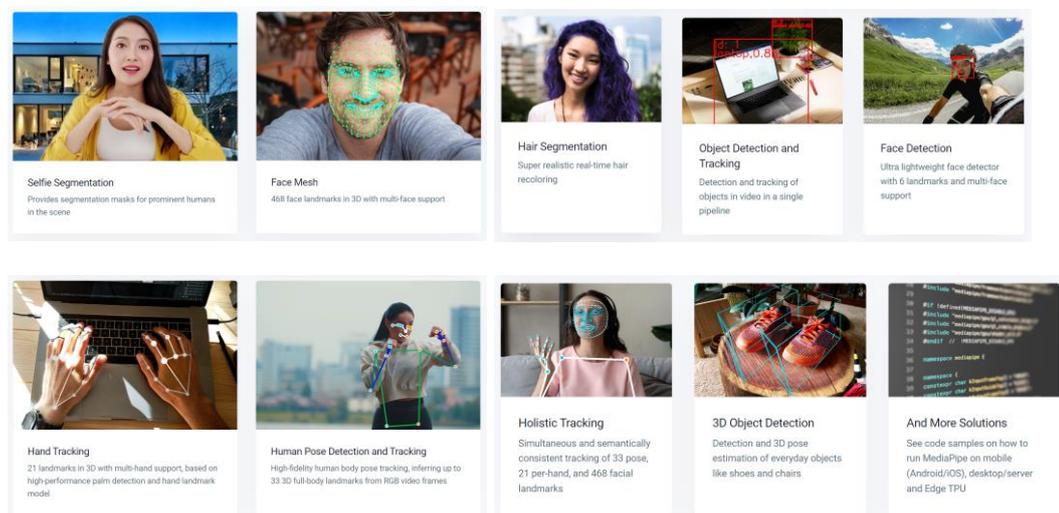
pada fitur. Nilai *eigen* dan vektor *eigen* dari matriks kovarian dihitung dengan menggunakan persamaan karakteristik berikut ini:

$$C - \lambda I = 0 \quad (3)$$

Dengan C adalah matriks kovarian, I adalah matriks identitas, λ adalah nilai *eigen* dan v adalah vektor *eigen*. Nilai *eigen* yang terbesar yang berkorespondensi terhadap nilai vektor *eigen* yang terbesar dipilih menjadi *principal component*.

2.5 Mediapipe Holistic

Mediapipe adalah sebuah kerangka kerja sumber terbuka (*open-source*) yang berfungsi sebagai *platform hybrid* untuk membangun alur kerja (*pipeline*) guna memproses data perseptual seperti gambar, video, dan audio. Dengan pendekatan yang kuat dalam penggunaan *machine learning*, *mediapipe* fokus pada pelacakan tangan dan pengenalan gerakan secara *real-time*. Kerangka kerja ini memberikan beragam solusi yang canggih untuk pelacakan tangan dan jari dengan kemampuan mendeteksi gerakan tubuh secara akurat (Subramanian et al., 2022). Fitur yang ditawarkan oleh *mediapipe* ditunjukkan pada gambar 1.



Sumber: <https://mediapipe-studio.webapps.google.com/home>

Gambar 1 Fitur dari mediapipe

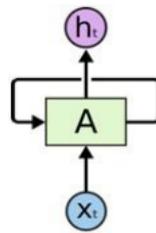
Secara spesifik, *mediapipe holistic* digunakan untuk mendapatkan *landmark* atau titik-titik penting pada wajah, tangan, dan postur tubuh. *Landmark* pose ini terdiri dari sekitar 33 landmark 3D, yang mencakup koordinat x , y , dan z

pada tubuh dalam gambar atau video yang diinputkan. Proses ini melibatkan penggunaan detektor *BlazePose* untuk mengidentifikasi orang atau postur tubuh dalam bingkai gambar dan menempatkan wilayah minat (ROI) pada orang atau pose tersebut. Setelah memotong ROI dari bingkai gambar, *mediapipe holistic* menggunakan *landmark* pose dan masker pemisahan di dalam ROI untuk mendeteksi pose secara berurutan. Ini memungkinkan untuk mengidentifikasi dengan akurat lebih banyak titik penting dan menyesuaikan dengan tampilan sudut tiga dimensi (SLR). Dengan demikian, *mediapipe holistic* memungkinkan pemrosesan yang canggih dalam pengenalan dan pelacakan pose tubuh, wajah, dan tangan dalam berbagai konteks aplikasi.

Mediapipe holistic memiliki sekitar 21 *landmark* atau penanda tangan tiga dimensi yang mencakup koordinat x, y, dan z dalam satu *frame* gambar. Untuk mencapai hasil ini, *mediapipe holistic* menggabungkan dua model, yaitu model deteksi telapak tangan dan model pelokalan titik kunci tangan. Awalnya, model ini bekerja dengan menggunakan detektor telapak tangan tunggal yang dikenal sebagai *Blaze Palm*. *Blaze Palm* membantu *mediapipe* dengan memproses gambar lebih efisien dengan fokus pada wilayah tangan, seperti telapak tangan dan jari-jari pertama. Hal ini mengurangi kerumitan deteksi tangan dengan memfokuskan perhatian pada objek yang relevan, yakni tangan, daripada menghabiskan waktu pada elemen yang tidak penting. Model ini menggunakan hasil dari deteksi telapak tangan untuk kemudian melokalisasi *landmark* atau titik kunci tangan.

2.6 Recurrent Neural Network

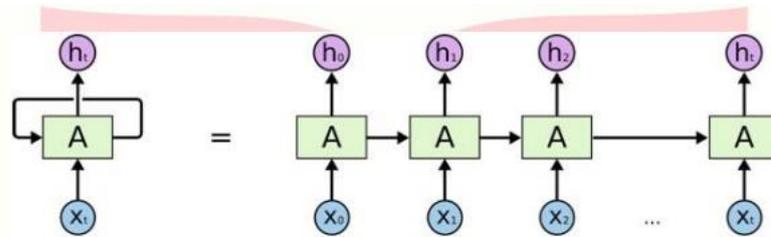
Manusia cenderung mengamati dan mempertimbangkan pengalaman masa lalu mereka saat membuat keputusan. Konsep berpikir ini menjadi dasar dari pengembangan *Recurrent Neural Network* (RNN). Secara sederhana, RNN merupakan bagian dari keluarga *Neural Network* yang dirancang untuk memproses data berkelanjutan atau berurutan (*sequential data*). RNN mencapai kemampuan untuk menyimpan informasi dari masa lalu dengan melakukan perulangan (*looping*) dalam arsitekturnya, sehingga informasi tersebut tetap tersedia untuk pengolahan selanjutnya secara otomatis (Lapian et al., 2018).



Sumber: Lapiam et al. (2018)

Gambar 2 Loop RNN tertutup

Gambar 2 adalah visualisasi contoh potongan dari sebuah RNN A. RNN tersebut mendapat *input* x_t dan menghasilkan *output* h_t . Dan alur *loop* tersebut memungkinkan informasi untuk dapat dilempar dari satu step menuju *step* selanjutnya.



Sumber: Lapiam et al. (2018)

Gambar 3 Loop RNN terbuka

Looping dalam RNN melibatkan pemrosesan *input* dari skala waktu awal (0) hingga waktu t . RNN memproses data *input* secara berurutan, satu per satu, dan mengalirkan informasi dari lapisan tersembunyi (*hidden layer*) ke lapisan tersembunyi pada waktu berikutnya, dan seterusnya secara berurutan (Lapiam et al., 2018). *Recurrent Neural Network* (RNN) adalah jenis *Neural Network* yang memiliki karakteristik utama berupa adanya koneksi berulang atau umpan balik (*feedback loop*), di mana sinyal kembali ke neuron itu sendiri. RNN memiliki setidaknya satu loop umpan balik dalam arsitekturnya, yang memungkinkannya untuk menyimpan informasi dalam struktur jaringannya. Kinerja RNN dalam tugas prediksi sangat dipengaruhi oleh konfigurasi bobot dan arsitektur yang digunakan. Selain itu, satu hal unik tentang RNN adalah adanya koneksi umpan balik yang memungkinkan informasi gangguan (*noise*) dari *input* sebelumnya memengaruhi *input* berikutnya.

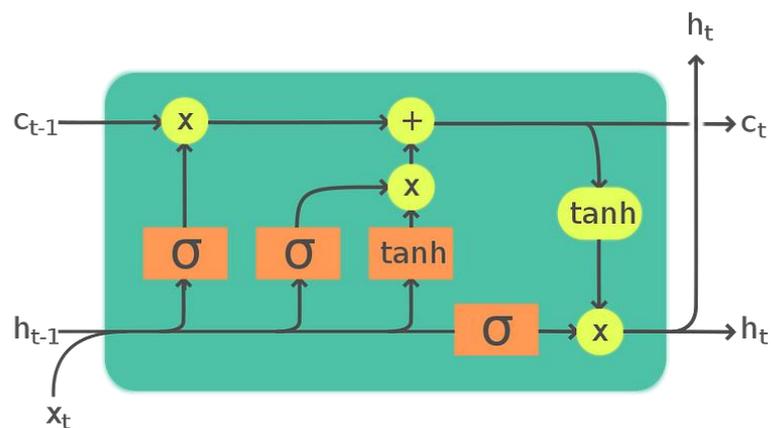
Recurrent Neural Networks (RNN) memang dikenal dengan kemampuannya untuk mempertahankan informasi sekuensial melalui loop umpan balik yang terintegrasi dalam arsitekturnya. Namun, ini juga menyebabkan salah satu tantangan terbesar dalam pelatihan RNN berupa masalah *vanishing gradient*. Saat gradien dari *output* jaringan (yang merupakan indikator dari perubahan yang perlu dibuat pada bobot selama back-propagation) diteruskan kembali melalui jaringan, nilai gradien ini cenderung menurun secara eksponensial dengan bertambahnya jumlah lapisan. Ini berakibat pada bobot di lapisan awal yang hampir tidak menerima update karena gradien yang sangat kecil, sehingga hampir tidak 'belajar' sama sekali.

Permasalahan ini mirip dengan efek berulangnya perkalian bilangan pecahan yang kecil; nilai yang dihasilkan menjadi semakin kecil dan pada akhirnya mendekati nol. Sebagai contoh, perkalian beruntun dari $1/4$, $1/3$, dan $1/5$ akan menghasilkan nilai yang sangat kecil, menunjukkan bagaimana gradien yang merambat ke belakang dalam RNN dapat berkurang drastis.

Untuk mengatasi *vanishing gradient*, berbagai metode telah dikembangkan, termasuk penggunaan arsitektur jaringan yang lebih canggih seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU), yang dirancang untuk mengatasi kendala tersebut dan memungkinkan informasi untuk mengalir melalui jaringan tanpa degradasi yang signifikan.

2.7 Long Short Term Memory (LSTM)

Sistem penyimpanan yang melibatkan RNN dan LSTM berperan penting dalam penyajian informasi dari data yang telah disimpan sebelumnya. RNN memiliki fungsionalitas ini, namun LSTM dibuat untuk menangani kelemahan RNN dalam menangani masalah *vanishing gradient* ketika memproses prediksi kata-kata dari informasi yang disimpan selama periode waktu yang panjang. LSTM dikembangkan dengan kemampuan untuk mempertahankan atau menghapus informasi seiring waktu, yang memungkinkannya lebih efektif dalam mengolah, memprediksi, dan mengklasifikasi data yang mengikuti urutan waktu tertentu (Ridwang, 2023).

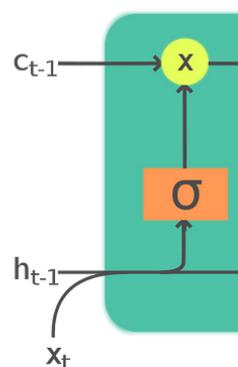


Sumber: Ridwang (2023)

Gambar 4 Cara kerja algoritma LSTM

Arsitektur LSTM yang standar mencakup komponen seperti sel memori yang bertugas menyimpan input, gerbang masuk yang menentukan jumlah informasi baru yang akan disimpan, gerbang keluar, dan gerbang lupa yang menentukan informasi mana yang akan dipertahankan atau dihapus dari sel memori.

a) *Forget Gate*



Sumber: Ridwang (2023)

Gambar 5 Skema gerbang *forget* LSTM

Dalam prosesnya, LSTM mengevaluasi dan memilah data yang masuk. Ini dilakukan melalui tahapan dimana fungsi sigmoid digunakan untuk menentukan informasi mana yang harus dipertahankan atau dibuang, tergantung pada relevansinya dengan situasi yang tengah dianalisis.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$

Variabel x_t mewakili vektor data yang masuk pada titik waktu t , dan h_{t-1} adalah representasi dari vektor keadaan tersembunyi dari waktu sebelumnya, atau waktu $t-1$. Melalui simulasi perhitungan, dapat digambarkan proses operasional dari gerbang forget LSTM. Misalnya, dengan parameter yang diberikan berikut:

Nilai input (x_t) = [0.55532569 0.28467178 -0.96421766]

Hidden state awal (h_{t-1}) = [0 0 0]

Bobot gerbang forget (W_f) =

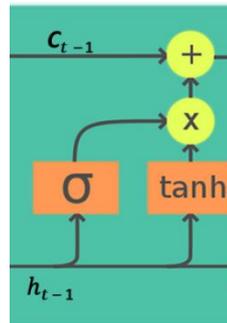
[1.76405235	0.40015721	0.97873798	2.2408932	1.86755799	-0.97727788]
[0.95008842	-0.15135721	-0.10321885	0.4105985	0.14404357	1.45427351]
[0.76103773	0.12167502	0.44386323	0.33367433	1.49407907	-0.20515826]

Bias gerbang *forget* (b_f) = [0.3130677 , -0.85409574, -2.55298982]

Dengan mengaplikasikan nilai-nilai yang diberikan ke dalam persamaan 4, didapatkan:

$$\begin{aligned}
 f_t &= \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \\
 &= \sigma (W_f \cdot [0, 0, 0, 0.55532569 \ 0.28467178 \ -0.96421766] + b_f) \\
 &= \sigma ([[1.76405235, 0.40015721, 0.97873798, 2.2408932, \\
 &\quad 1.86755799, -0.97727788], [0.95008842, -0.15135721, -0.10321885, \\
 &\quad 0.4105985, 0.14404357, 1.45427351], [0.76103773, 0.12167502, \\
 &\quad 0.44386323, 0.33367433, 1.49407907, -0.20515826]] \cdot [0, 0, 0, \\
 &\quad 0.55532569 \ 0.28467178 \ -0.96421766] + 0.3130677, -0.85409574, - \\
 &\quad 2.55298982]) \\
 &= \sigma [3.03144291, -1.9873109, -1.74455252] \\
 &= \frac{1}{1 + e^{-x}} ([3.03144291, -1.9873109, -1.74455252] \\
 &= \frac{1}{1 + e^{-[0.55532569 \ 0.28467178 \ -0.96421766]}} \\
 &\quad ([3.03144291, -1.9873109, -1.74455252] \\
 &= [0.95397457, 0.12054165, 0.1487356]
 \end{aligned}$$

b) *Input Gate*



Sumber: Ridwang (2023)

Gambar 6 Skema gerbang *input* LSTM

Berikutnya, *input gate* (i_t) memproses informasi dengan menerapkan rumus tertentu, yang dikenal sebagai Persamaan 4. Melalui proses ini, informasi yang akan diperbaharui dipilih dengan cermat. Fungsi aktivasi sigmoid menentukan bagian mana dari status sel yang perlu diupdate, sementara fungsi aktivasi tanh membantu dalam pembentukan kandidat vektor yang akan digunakan untuk pembaruan. Kemudian, menggunakan formula yang dijelaskan dalam Persamaan 5, vektor kandidat ini diintegrasikan ke dalam *Cell State* \tilde{C}_t .

$$i_t = (W_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (6)$$

Kemudian perbarui nilai status sel lama c_{t-1} ke status sel baru c_t dengan persamaan 7

$$C_t = (f_t * C_{t-1} + i_t * \tilde{C}_t) \quad (7)$$

Melalui proses ini, informasi yang akan diperbaharui dipilih dengan cermat. Fungsi aktivasi sigmoid menentukan bagian mana dari status sel yang perlu diupdate, sementara fungsi aktivasi tanh membantu dalam pembentukan kandidat vektor yang akan digunakan untuk pembaruan. Kemudian, menggunakan formula yang dijelaskan dalam Persamaan 5, vektor kandidat ini diintegrasikan ke dalam *Cell State* \tilde{C}_t . Misalnya jika diberikan nilai bobot dan bias berikut:

Bobot gerbang input (W_i) =

[0.6536186 0.8644362 -0.74216502 2.26975462 -1.45436567 0.04575852]

[0.18718385 1.53277921 1.46935877 0.15494743 0.37816252 -0.88778575]
 [1.98079647 -0.34791215 0.15634897 1.23029068 1.20237985 -0.38732682]

Bobot kandidat cell state (W_c) =

[-1.70627019 1.9507754 -0.50965218 -0.4380743 -1.25279536 0.77749036]
 [1.61389785 -0.21274028 -0.89546656 0.3869025 -0.51080514 -1.18063218]
 [0.02818223 0.42833187 0.06651722 0.3024719 -0.63432209 -0.36274117]

Bias gerbang input (b_i) = [-0.30230275, -1.04855297, -1.42001794]

Bias kandidat cell state (b_c) = [-0.67246045, -0.35955316, -0.81314628]

Dengan mengaplikasikan nilai-nilai yang diberikan ke dalam persamaan 5, didapatkan:

$$\begin{aligned}
 i_t &= (W_i \cdot [h_{t-1}, x_t] + b_i) \\
 &= \sigma(W_f \cdot [0, 0, 0, 0.55532569 \ 0.28467178 \ -0.96421766] + b_i) \\
 &= \sigma([[0.6536186, 0.8644362, -0.74216502, 2.26975462, -1.45436567, \\
 &\quad 0.04575852], [-0.18718385, 1.53277921, 1.46935877, 0.15494743, \\
 &\quad 0.37816252, -0.88778575], [-1.98079647, -0.34791215, 0.15634897, \\
 &\quad 1.23029068, 1.20237985, -0.38732682]] \cdot [0, 0, 0, 0.55532569 \\
 &\quad 0.28467178 \ -0.96421766] + [-0.30230275, -1.04855297, -1.42001794]) \\
 &= \sigma [0.50001227, 0.00116421, -0.02105495] \\
 &= \frac{1}{1 + e^{-x}} ([0.50001227, 0.00116421, -0.02105495]) \\
 &= \frac{1}{1 + e^{-[0.55532569 \ 0.28467178 \ -0.96421766]}} ([0.50001227, 0.00116421, \\
 &\quad -0.02105495]) \\
 &= [0.62246221, 0.50029105, 0.49473646]
 \end{aligned}$$

Dengan mengaplikasikan nilai-nilai yang diberikan ke dalam persamaan 6, didapatkan:

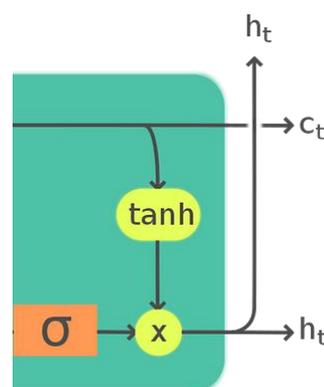
$$\begin{aligned}
 \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\
 &= \tanh(W_c \cdot [0, 0, 0, 0.55532569 \ 0.28467178 \ -0.96421766] + b_c) \\
 &= \tanh([[-1.70627019, 1.9507754, -0.50965218, -0.4380743, -1.25279536, \\
 &\quad 0.77749036], [-1.61389785, -0.21274028, -0.89546656, 0.3869025, - \\
 &\quad 0.51080514, -1.18063218], [-0.02818223, 0.42833187, 0.06651722,
 \end{aligned}$$

$$\begin{aligned}
& 0.3024719, -0.63432209, -0.36274117] \cdot [[0, 0, 0], [0.55532569 \\
& 0.28467178 -0.96421766]] + [-0.67246045, -0.35955316, -0.81314628]) \\
= & \tanh([-1.34957933, 1.20783149, 0.33715825] + [-0.67246045, - \\
& 0.35955316, -0.81314628]) \\
= & \tanh([-2.02203978, 0.84827833, -0.47598803]) \\
= & \frac{e^x - e^{-x}}{e^x + e^{-x}}([-2.02203978, 0.84827833, -0.47598803]) \\
= & \frac{e^{0.55532569 \cdot 0.28467178 - 0.96421766} - e^{-0.55532569 \cdot 0.28467178 - 0.96421766}}{e^{0.55532569 \cdot 0.28467178 - 0.96421766} + e^{-0.55532569 \cdot 0.28467178 - 0.96421766}}([-2.02203978, \\
& 0.84827833, -0.47598803]) \\
= & [-0.96555207, 0.69016896, -0.44302481]
\end{aligned}$$

Lalu dengan mengaplikasikan nilai-nilai yang diberikan ke dalam persamaan 7, didapatkan:

$$\begin{aligned}
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
&= [0.95397457, 0.12054165, 0.1487356] * [0, 0, 0] + [0.62246221, \\
& 0.50029105, 0.49473646] * [-0.96555207, 0.69016896, -0.44302481] \\
&= [0, 0, 0] + [-0.60101968, 0.34528536, -0.21918053] \\
&= [-0.60101968, 0.34528536, -0.21918053]
\end{aligned}$$

c) *Output Gate*



Sumber: Ridwang (2023)

Gambar 7 Skema gerbang *output* LSTM

Proses terakhir melibatkan *output gate*, di mana fungsi sigmoid dijalankan untuk mendapatkan nilai yang akan dijadikan *hidden state*, sementara itu, *cell state*

dimodifikasi dengan fungsi tanh. Nilai yang dihasilkan dari kedua fungsi aktivasi ini kemudian dikombinasikan. Langkah selanjutnya menghitung output berdasarkan kedua nilai tersebut, yang prosesnya diatur oleh Persamaan 8 dan 9.

$$O_t = (W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = O_t \tanh(C_t) \quad (9)$$

Langkah ini merupakan penentu akhir yang menggabungkan hasil fungsi aktivasi sigmoid dan tanh untuk memproduksi state tersembunyi selanjutnya dan output dari LSTM, siap untuk digunakan dalam pengambilan keputusan atau langkah pemrosesan selanjutnya. Misalnya jika diberikan nilai bobot dan bias berikut:

Bobot gerbang output (W_o) =

[1.76405235 0.40015721 0.97873798 2.2408932 1.86755799 -0.97727788]
 [0.95008842 -0.15135721 -0.10321885 0.4105985 0.14404357 1.45427351]
 [0.76103773 0.12167502 0.44386323 0.33367433 1.49407907 -0.20515826]

Bias gerbang output (b_o): [0.3130677, -0.85409574, -2.55298982]

Dengan mengaplikasikan nilai ini ke persamaan 10, diperoleh:

$$\begin{aligned} O_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ &= \sigma(W_o \cdot [0, 0, 0, 0.55532569, 0.28467178, -0.96421766] + b_o) \\ &= \sigma([1.76405235, 0.40015721, 0.97873798, 2.2408932, 1.86755799, \\ &\quad -0.97727788], [0.95008842, -0.15135721, -0.10321885, 0.4105985, \\ &\quad 0.14404357, 1.45427351], [0.76103773, 0.12167502, 0.44386323, \\ &\quad 0.33367433, 1.49407907, -0.20515826]) \cdot [0, 0, 0, 0.55532569 \\ &\quad 0.28467178, -0.96421766] + [0.3130677, -0.85409574, - \\ &\quad 2.55298982]) \\ &= \sigma[1.00210962, 0.35893593, -0.56981007] \\ &= \frac{1}{1 + e^{-x}}([1.00210962, 0.35893593, -0.56981007]) \\ &= \frac{1}{1 + e^{-[0.55532569, 0.28467178, -0.96421766]}}([1.00210962, 0.35893593, \\ &\quad -0.56981007]) \\ &= [0.73147315, 0.58878283, 0.36128065] \end{aligned}$$

Setelah menghitung *output gate* melalui persamaan 8, langkah berikutnya dalam LSTM adalah menghitung hidden state. Proses ini menggunakan output dari *output gate* dan kondisi cell state saat ini, yang diolah melalui Persamaan 9, untuk

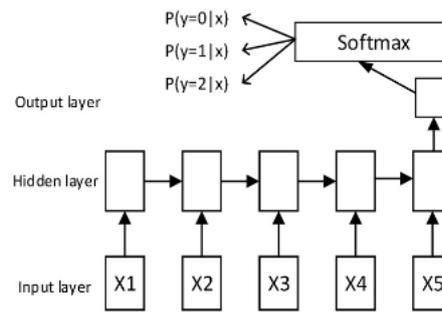
menghasilkan hidden state baru. Langkah ini penting karena hidden state menangkap dan menyimpan informasi penting dari urutan data yang telah diproses, memberikan dasar untuk prediksi atau keputusan yang akan diambil oleh LSTM selanjutnya.

$$\begin{aligned}
 h_t &= O_t * \tanh(C_t) \\
 &= [0.73147315, 0.58878283, 0.36128065] * \frac{e^x - e^{-x}}{e^x + e^{-x}} ([-0.60101968, \\
 &\quad 0.34528536, -0.21918053]) \\
 &= [0.73147315, 0.58878283, 0.36128065] * \\
 &\quad \frac{e^{0.55532569 \ 0.28467178 \ -0.96421766} - e^{-0.55532569 \ 0.28467178 \ -0.96421766}}{e^{0.55532569 \ 0.28467178 \ -0.96421766} + e^{-0.55532569 \ 0.28467178 \ -0.96421766}} \\
 &\quad ([-0.60101968, 0.34528536, -0.21918053]) \\
 &= [0.73147315, 0.58878283, 0.36128065] * [-0.53777475, 0.33218774, \\
 &\quad -0.21573687] \\
 &= [-0.39336779, 0.19558644, -0.07794156]
 \end{aligned}$$

2.8 Gated Recurrent Unit

Gated Recurrent Unit (GRU) adalah variasi dari LSTM yang dikembangkan sebagai respons terhadap kebutuhan untuk model yang lebih efisien secara komputasi. Meskipun LSTM sangat efektif dalam mengatasi masalah *vanishing gradient* dan memungkinkan pembelajaran ketergantungan jangka panjang, arsitekturnya relatif kompleks dan melibatkan sejumlah besar operasi dan parameter. GRU diusulkan untuk menyederhanakan struktur LSTM tanpa mengorbankan kemampuan untuk menangani ketergantungan jangka panjang.

GRU mengintegrasikan dan memodifikasi gerbang LSTM menjadi dua gerbang utama berupa gerbang pembaruan (*update gate*) dan gerbang setel ulang (*reset gate*) yang memungkinkan model untuk memutuskan kapan harus memperbarui atau mengabaikan informasi dengan biaya komputasi yang lebih rendah. GRU termasuk dalam kategori *gated RNN*, yang dirancang khusus untuk mengatasi masalah yang timbul pada RNN konvensional terkait dengan *gradient* yang menghilang saat proses pembelajaran dengan ketergantungan jarak jauh (Shen et al., 2018).



Sumber: Shen et al. (2018)

Gambar 8 Struktur dari model dasar GRU

Seperti yang diilustrasikan pada Gambar 8, terdapat lapisan *input* yang terdiri dari beberapa neuron, jumlah neuron ditentukan oleh ukuran ruang fitur. Begitu juga jumlah neuron pada lapisan *output* sesuai dengan ruang keluaran. Lapisan tersembunyi yang berisi sel-sel memori meliputi fungsi utama dari jaringan GRU. Secara khusus, standar Arsitektur GRU didefinisikan dengan persamaan berikut:

$$z_t = \sigma(W_z x_t + W_z h_{t-1} + b_z), \quad (10a)$$

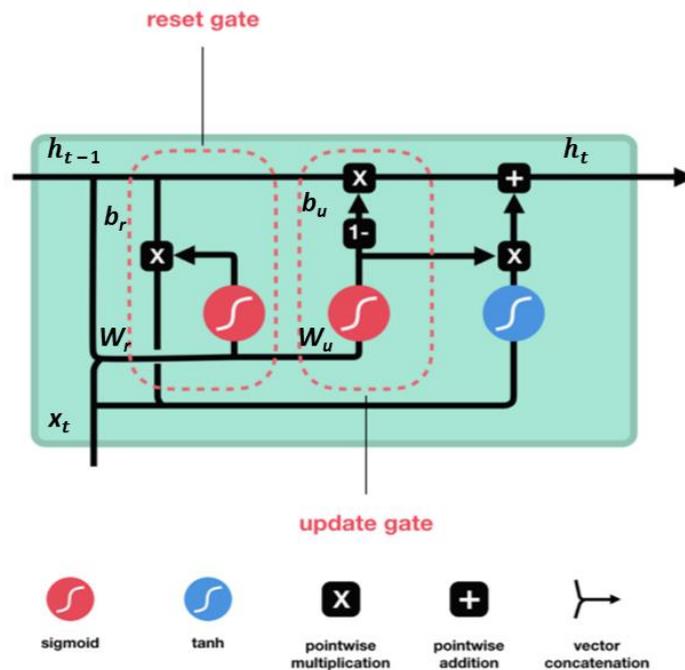
$$r_t = \sigma(W_r x_t + W_r h_{t-1} + b_r), \quad (10b)$$

$$h_t = \tanh(W_h x_t + W_h (h_{t-1} \odot r_t) + b_h), \quad (10c)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t \quad (10d)$$

Dalam konteks ini, terdapat vektor z_t dan r_t yang berkaitan dengan gerbang pembaruan (*update*) dan gerbang reset. Selain itu, terdapat juga vektor h_t yang menggambarkan keadaan pada waktu t . Perkalian elemen-wise ditunjukkan dengan tanda titik (\odot) dalam operasi. Aktivasi pada kedua gerbang ini dihitung menggunakan fungsi sigmoid logistik, yang memastikan bahwa nilai z_t dan r_t berada dalam rentang antara 0 dan 1. Vektor keadaan kandidat h_t diproses menggunakan fungsi tangen hiperbolik. Selain itu, terdapat parameter model yang terdiri dari matriks W_z , W_r , W_h (yang menggambarkan konektivitas *feed-forward* dan bobot rekuren). Arsitektur ini juga melibatkan vektor bias b_z , b_r , dan b_h yang dapat disesuaikan dan ditambahkan sebelum penerapan fungsi *non-linearitas* (Ravanelli et al., 2018).

Inti dari algoritma LSTM dan GRU adalah kemampuannya untuk menyimpan informasi yang penting dan relevan untuk keperluan prediksi, sementara data yang dianggap tidak relevan akan dilupakan. Proses ini terjadi dengan memanfaatkan nilai-nilai yang mendekati 0 untuk penghapusan informasi yang tidak relevan, dan nilai-nilai yang mendekati 1 untuk pembaruan informasi yang penting. Cara kerja algoritma GRU ditunjukkan pada gambar 5 (Paisal, 2023).



Sumber: Paisal (2023)

Gambar 9 Cara kerja algoritma GRU

Algoritma GRU beroperasi dengan memanfaatkan dua gerbang, yang dikenal sebagai gerbang pembaruan (*update*) dan gerbang reset. Pada prinsipnya, kedua gerbang ini merupakan dua vektor yang menentukan informasi mana yang harus disampaikan ke *output*. Secara spesifik, algoritma ini dapat dilatih untuk menyimpan informasi dari masa lalu tanpa menghapusnya dari waktu ke waktu atau mengeliminasi data yang tidak relevan dengan prediksi.

a) Gerbang *Update*

Untuk rumusnya sendiri adalah:

$$Z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (11)$$

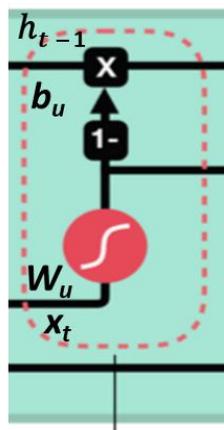
dimana,

x_t = vektor input pada waktu t

h_{t-1} = vektor *output* (*hidden state*) pada waktu sebelumnya

W_z = matriks dan vektor bobot yang sesuai untuk gerbang *update*

Ketika x_t diberikan ke-unit jaringan, itu dikalikan dengan bobotnya sendiri $W(z)$. Demikian pula untuk $h_{(t-1)}$ ia menyimpan informasi dari unit sebelumnya $t-1$ dan mengalikannya dengan bobotnya sendiri. Kedua hasil ditambahkan dan fungsi aktivasi sigmoidal diterapkan untuk membatasi hasil antara 0 dan 1.



Sumber: Paisal (2023)

Gambar 10 Skema gerbang *update* GRU

Setelah memahami mekanisme teoretis dari gerbang pembaruan dan reset dalam algoritma GRU, penting untuk melihat bagaimana prinsip-prinsip ini diterjemahkan ke dalam praktik. Melalui simulasi perhitungan, dapat digambarkan proses operasional dari gerbang update GRU. Misalnya, dengan parameter yang diberikan berikut:

Nilai input (x_t) = [0.55532569 0.28467178 -0.96421766]

Hidden state awal (h_{t-1}) = [0 0 0]

Bobot gerbang update (W_z) =

1.76405235	0.40015721	0.97873798	0.55532569	0.28467178	-0.96421766
-0.10321885	0.4105985	0.14404357	0.55532569	0.28467178	-0.96421766
1.49407907	-0.20515826	0.3130677	0.55532569	0.28467178	-0.96421766

Bias gerbang update (b_z) = [1.13940068 -1.23482582 0.40234164]

Dalam simulasi ini, vektor input pada waktu t digabungkan dengan *hidden state* dari waktu sebelumnya, kemudian mengaplikasikan bobot yang relevan. Kedua hasil ini ditambahkan bersama dan fungsi aktivasi sigmoid diterapkan untuk mendapatkan hasil gerbang *update*. Dengan mengaplikasikan nilai-nilai yang diberikan ke dalam persamaan 11, didapatkan:

$$\begin{aligned}
Z_t &= \sigma (W_z \cdot [h_{t-1}, x_t] + b_z) \\
&= \sigma (W_z \cdot [0, 0, 0, 0.55532569 \ 0.28467178 \ -0.96421766] + b_z) \\
&= \sigma ([[1.76405235, 0.40015721, 0.97873798, 0.55532569, 0.28467178, - \\
&\quad 0.96421766], [-0.10321885, 0.4105985, 0.14404357, 0.55532569, \\
&\quad 0.28467178, -0.96421766], [1.49407907, -0.20515826, 0.3130677, \\
&\quad 0.55532569, 0.28467178, -0.96421766]] \cdot [0, 0, 0, 0.55532569 \\
&\quad 0.28467178 \ -0.96421766] + [1.13940068, -1.23482582, 0.40234164]) \\
&= \sigma ([2.69019298, -0.70488329, 0.87495452]) \\
&= \frac{1}{1+e^{-x}} ([2.69019298, -0.70488329, 0.87495452]) \\
&= \frac{1}{1+e^{-[0.55532569 \ 0.28467178 \ -0.96421766]}} ([2.69019298, -0.70488329, \\
&\quad 0.87495452]) \\
Z_t &= [0.93644547, 0.33073043, 0.70577558]
\end{aligned}$$

Simulasi ini menunjukkan bagaimana nilai-nilai input dan state sebelumnya bersama dengan bobot dan bias mereka berinteraksi untuk menghasilkan output dari gerbang *update*. Proses ini merupakan bagian integral dari bagaimana GRU membuat keputusan tentang informasi mana dari masa lalu yang perlu dipertahankan, memberikan intuisi tentang cara model menangkap dependensi temporal dalam data.

b) Gerbang Reset

Pada dasarnya, gerbang ini digunakan dari model untuk memutuskan seberapa banyak informasi masa lalu yang harus dilupakan. Untuk menghitungnya, rumusnya sebagai berikut:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (12)$$

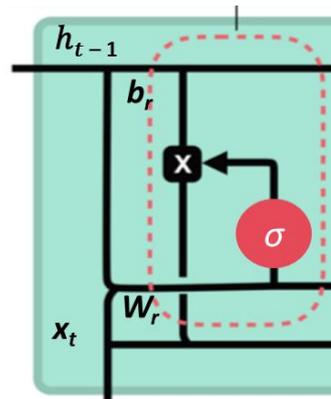
dimana,

X_t = vektor input pada waktu t

h_{t-1} = vektor *output* (*hidden state*) pada waktu sebelumnya

W_r, U_r, b_r = matriks dan vektor bobot yang sesuai untuk gerbang reset

Rumus ini sama dengan rumus gerbang pembaruan. Perbedaannya terletak pada berat dan metode penggunaan gerbang, skemanya terlihat pada gambar 11



Sumber: Paisal (2023)

Gambar 11 Skema gerbang reset GRU

Sebelumnya, h_{t-1} dan x_t dimasukkan, bobot yang sesuai diterapkan pada keduanya, hasilnya dijumlahkan, dan fungsi sigmoid diterapkan pada hasil gabungan tersebut (Ridwang, 2022). Gerbang reset (r_t) bekerja secara paralel dengan gerbang update untuk memungkinkan model memutuskan informasi mana dari masa lalu yang relevan untuk prediksi saat ini dan mana yang harus diabaikan. Misalnya jika diberikan nilai bobot dan bias berikut:

Bobot gerbang reset (W_r) =

$$\begin{bmatrix} 2.26975462 & -1.45436567 & 0.04575852 & 0.55532569 & 0.28467178 & -0.96421766 \\ -0.88778575 & -1.98079647 & -0.34791215 & 0.55532569 & 0.28467178 & -0.96421766 \\ -1.04855297 & -1.42001794 & -1.70627019 & 0.55532569 & 0.28467178 & -0.96421766 \end{bmatrix}$$

Bias gerbang reset (b_r) = [-0.68481009 -0.87079715 -0.57884966]

Dengan mengaplikasikan nilai-nilai yang diberikan ke dalam persamaan 12, didapatkan:

$$\begin{aligned} r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \\ &= \sigma(W_r \cdot [0, 0, 0, 0.55532569, 0.28467178, -0.96421766] + b_r) \\ &= \sigma([[2.26975462 \quad -1.45436567 \quad 0.04575852 \quad 0.55532569 \quad 0.28467178 \quad - \\ &\quad 0.96421766], [-0.88778575 \quad -1.98079647 \quad -0.34791215 \quad 0.55532569 \end{aligned}$$

$$\begin{aligned}
& 0.28467178 \quad -0.96421766], \quad [-1.04855297 \quad -1.42001794 \quad -1.70627019 \\
& 0.55532569 \quad 0.28467178 \quad -0.96421766]] \cdot [0 \quad 0 \quad 0 \quad 0.55532569 \quad 0.28467178 \\
& -0.96421766] + [-0.68481009 \quad -0.87079715 \quad -0.57884966]) \\
& = \sigma ([1.62713231, -1.71871278, -1.71252354]) \\
& = \frac{1}{1+e^{-x}} ([1.62713231, -1.71871278, -1.71252354]) \\
& = \frac{1}{1+e^{-[0.55532569 \quad 0.28467178 \quad -0.96421766]([1.62713231 \quad -1.71871278 \quad -1.71252354])}} \\
& = [0.83577641 \quad 0.15203704 \quad 0.15283669]
\end{aligned}$$

Dengan demikian, melalui simulasi perhitungan ini, kita dapat memvisualisasikan bagaimana gerbang reset mempengaruhi aliran informasi dalam jaringan GRU, mengontrol bagaimana informasi dari masa lalu dipertahankan atau dihilangkan dalam konteks tugas prediksi saat ini.

Setelah menetapkan nilai gerbang reset yang mengatur aliran informasi dari masa lalu, langkah selanjutnya dalam mekanisme GRU adalah pembentukan kandidat hidden state. Kandidat hidden state (\tilde{h}_t) merupakan hasil dari penggabungan informasi terkini dari input saat ini dengan informasi yang telah diubah oleh gerbang reset. Ini menentukan potensi state baru sebelum diintegrasikan dengan state sebelumnya yang telah diatur oleh gerbang update. Proses perhitungan kandidat hidden state dapat dijelaskan melalui persamaan:

$$\tilde{h}_t = \tanh (W \cdot [r_t * h_{t-1}, x_t] + b) \quad (13)$$

Dalam konteks simulasi yang sama, dengan menggunakan bobot dan bias berikut:

Bobot kandidat *hidden state* (W) =

$$\begin{aligned}
& [-1.61389785 \quad -0.21274028 \quad -0.89546656 \quad 0.55532569 \quad 0.28467178 \quad -0.96421766] \\
& [0.06651722 \quad 0.3024719 \quad -0.63432209 \quad 0.55532569 \quad 0.28467178 \quad -0.96421766] \\
& [0.17742614 \quad -0.40178094 \quad -1.63019835 \quad 0.55532569 \quad 0.28467178 \quad -0.96421766]
\end{aligned}$$

Bias kandidat *hidden state* (b_h) = [-0.31155253 0.05616534 -1.16514984]

Dengan mengaplikasikan nilai-nilai yang diberikan ke dalam persamaan 13, didapatkan:

$$\begin{aligned}
\tilde{h}_t &= \tanh (W \cdot [r_t * h_{t-1}, x_t] + b) \\
&= \tanh (W \cdot [0.83577641 \quad 0.15203704 \quad 0.1528366] * [0, 0, 0], [0.55532569 \\
&\quad 0.28467178 \quad -0.96421766]) + b)
\end{aligned}$$

$$\begin{aligned}
&= \tanh (W \cdot [[0 \ 0 \ 0], [0.55532569 \ 0.28467178 \ -0.96421766]] + b) \\
&= \tanh ([[-1.61389785 \ -0.21274028 \ -0.89546656 \ 0.55532569 \\
&\quad 0.28467178 \ -0.96421766], [0.06651722 \ 0.3024719 \ -0.63432209 \\
&\quad 0.55532569 \ 0.28467178 \ -0.96421766], [0.17742614 \ -0.40178094 \ - \\
&\quad 1.63019835 \ 0.55532569 \ 0.28467178 \ -0.96421766]] \cdot [[0 \ 0 \ 0], [\\
&\quad 0.55532569 \ 0.28467178 \ -0.96421766]] + [-0.31155253 \ 0.05616534 \ - \\
&\quad 1.16514984]) \\
&= \tanh ([0.230376 \ -1.70913249 \ 0.39468521]) \\
&= \frac{e^x - e^{-x}}{e^x + e^{-x}} ([0.230376 \ -1.70913249 \ 0.39468521]) \\
&= \frac{e^{0.55532569 \ 0.28467178 \ -0.96421766} - e^{-0.55532569 \ 0.28467178 \ -0.96421766}}{e^{0.55532569 \ 0.28467178 \ -0.96421766} + e^{-0.55532569 \ 0.28467178 \ -0.96421766}} ([0.230376 \\
&\quad -1.70913249 \ 0.39468521]) \\
&= [0.22638511 \ -0.93654102 \ 0.37539226]
\end{aligned}$$

Nilai kandidat *hidden state* yang didapatkan akan diintegrasikan lebih lanjut dengan output dari gerbang *update* untuk membentuk *hidden state* terakhir yang akan digunakan untuk prediksi selanjutnya dalam sekuens. Adapun perhitungan kandidat *hidden state* akhir sebagai berikut:

$$\begin{aligned}
h_t &= (1-z_t) * h_{t-1} + z_t * \tilde{h}_t \\
&= (1 - [0.93644547, 0.33073043, 0.70577558]) * [0, 0, 0] + [0.93644547, \\
&\quad 0.33073043, 0.70577558] * [0.22638511, -0.93654102, 0.37539226] \\
&= [0, 0, 0] + [0.21199731, -0.30974261, 0.26494269] * [0.21199731, \\
&\quad -0.30974261, 0.26494269] \\
&= [0.21199731, -0.30974261, 0.26494269] * \\
&\quad [0.21199731, -0.30974261, 0.26494269]
\end{aligned}$$

Perhitungan ini mencapai keseimbangan antara mempertahankan informasi berharga dari masa lalu dan mengadaptasi dengan *insight* baru yang didapat dari input saat ini. Nilai h_t ini kemudian menjadi *hidden state* yang akan dipakai untuk proses pada langkah waktu berikutnya, dan juga sebagai bagian dari output yang dihasilkan jika layer GRU tersebut diatur untuk mengembalikan output sekuensial.

2.9 Confusion Matrix

Evaluasi kinerja sistem klasifikasi adalah aspek penting dalam penilaian suatu sistem. Ini mencerminkan sejauh mana sistem mampu melakukan klasifikasi data secara efektif. Salah satu alat yang berguna untuk mengukur kinerja metode klasifikasi adalah *confusion matrix*. Pada dasarnya, *confusion matrix* berisi informasi yang membandingkan hasil klasifikasi yang diberikan oleh sistem dengan hasil yang seharusnya. Dalam pengukuran kinerja dengan *confusion matrix*, ada empat istilah yang digunakan untuk mewakili hasil klasifikasi, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *True Negative* (TN) menggambarkan jumlah data negatif yang benar-benar terklasifikasikan dengan benar, sementara *False Positive* (FP) mengacu pada data yang salah diklasifikasikan sebagai positif. Selanjutnya, *True Positive* (TP) mewakili data positif yang benar-benar terklasifikasikan dengan benar, sedangkan *False Negative* (FN) adalah situasi di mana data positif salah diklasifikasikan sebagai negatif. Ini adalah konsep yang umum digunakan dalam evaluasi sistem klasifikasi, terutama dalam konteks klasifikasi biner dengan dua kelas keluaran. Confusion matrix dapat disajikan seperti gambar 12.

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

Sumber: <https://ilmudatapy.com/apa-itu-confusion-matrix/> (2021)

Gambar 12 Tabel *confusion matrix*

Berdasarkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), dapat dihitung sejumlah metrik evaluasi yang memberikan wawasan tentang kinerja model klasifikasi. Akurasi (*Accuracy*) mengukur sejauh mana model mampu membuat prediksi yang benar secara keseluruhan. Tingkat akurasi yang tinggi menunjukkan bahwa model secara keseluruhan efektif dalam mengenali dan mengklasifikasikan data dengan tepat.

Dengan akurasi yang tinggi, model mampu memberikan prediksi yang akurat dan dapat diandalkan dalam membedakan berbagai kelas atau label.

Recall, yang juga dikenal sebagai sensitivitas, mengukur kemampuan model untuk mengidentifikasi semua contoh positif yang relevan dalam data yang sebenarnya. *Recall* yang tinggi menunjukkan bahwa model efektif dalam mengenali dan menangkap sebagian besar contoh positif. Dalam konteks klasifikasi, contoh positif adalah yang benar-benar termasuk dalam kelas atau label yang dituju. Oleh karena itu, *recall* yang tinggi menunjukkan bahwa model memiliki sedikit kesalahan dalam mengabaikan atau salah mengklasifikasikan contoh positif.

Presisi yang tinggi menunjukkan bahwa model cenderung memberikan sedikit *false positive* (prediksi positif yang sebenarnya negatif). Presisi yang tinggi memberikan keyakinan bahwa ketika model memberikan prediksi positif, prediksi tersebut cenderung benar.

F1-score adalah penggabungan antara presisi dan *recall*, yang memberikan gambaran yang lebih lengkap tentang kinerja model. *F1-score* mengukur keseimbangan antara presisi dan *recall*. *F1-score* yang tinggi menunjukkan bahwa model memiliki baik presisi maupun *recall* yang baik secara bersamaan. Dalam hal ini, model dapat memberikan prediksi yang akurat (presisi tinggi) dan menemukan sebagian besar contoh positif yang relevan (*recall* tinggi).

2.10 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware, dan aplikasi. Android merupakan sistem operasi Linux yang bersifat terbuka (*open source*) dan dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Untuk mengembangkan aplikasi pada platform android hanya perlu mendownload SDK (*Software Development Kit*) secara gratis dan dapat mulai menulis kode aplikasi untuk *handphone* android. Aplikasi-aplikasi android dapat di download dari Google Play, baik secara berbayar

maupun gratis (Khatima, 2022). Sistem operasi Android memiliki sejumlah keunggulan, termasuk (Pratama, 2022):

1. **Open Source:** Platform pengembangan yang berbasis Linux dan open source memungkinkan pembuat *handset* untuk mengeditnya tanpa membayar royalti. Kebebasan ini memberikan fleksibilitas kepada pengembang karena Android tidak terikat dengan vendor tertentu.
2. **Arsitektur Berbasis Komponen:** Android mengadopsi arsitektur berbasis komponen yang terinspirasi dari kebebasan dunia internet. Bagian dari sebuah aplikasi dapat digunakan kembali oleh pengembang dalam konteks yang berbeda, mendorong kreativitas baru di dunia mobile.
3. **Banyak Fitur Built-in:** Android menyertakan banyak fitur *built-in*, seperti layanan berbasis lokasi menggunakan GPS atau triangulasi menara sel, database SQL penuh tenaga, browser, dan tampilan peta yang dapat disematkan langsung dalam aplikasi. Fitur ini membantu meningkatkan fungsionalitas sambil mengurangi biaya pengembangan.
4. **Manajemen Siklus Hidup Aplikasi Otomatis:** Android memiliki manajemen siklus hidup aplikasi yang otomatis, dengan program yang terisolasi satu sama lain melalui lapisan keamanan, meningkatkan stabilitas sistem dan menghilangkan kekhawatiran pengguna terhadap aplikasi yang aktif atau nonaktif.
5. **Grafis dan Suara Berkualitas Tinggi:** Android menyajikan grafis vektor antialiasing 2D yang halus dan animasi terinspirasi oleh Flash, bersatu dengan grafis Open GL 3D yang dipercepat untuk mendukung jenis permainan dan aplikasi baru. Codec untuk audio dan video format standar industri seperti H.264 (AVC), MP3, dan AAC telah dibangun di dalamnya.
6. **Portabilitas di Berbagai Perangkat Hardware:** Android ditulis dalam Java dan dieksekusi oleh mesin virtual Android (Dalvik Android), membuat kode portabel di berbagai arsitektur perangkat keras seperti ARM dan x86. Dukungan untuk berbagai metode masukan seperti keyboard, sentuhan, dan trackball, serta kemampuan penyesuaian antarmuka pengguna untuk setiap resolusi layar dan orientasi.

2.11 Black Box Testing

Pengujian fungsional, yang lebih umum dikenal dengan istilah *black box testing*, adalah suatu metode pengujian perangkat lunak yang berorientasi pada persyaratan fungsional tanpa memperhatikan struktur internal kode atau program. Fokus pengujian terletak pada aspek rinci aplikasi, seperti antarmuka pengguna, fungsi-fungsi yang tersedia, dan kesesuaian alur fungsi dengan proses yang diinginkan oleh pengguna (Pratama, 2022). Ada beberapa teknik yang biasanya digunakan untuk menguji perangkat lunak. Berikut ini adalah teknik-tekniknya:

1. **All pair testing:** Teknik *all pair testing* ini dikenal juga dengan *pairwise testing*. Pengujian ini digunakan untuk menguji semua kemungkinan kombinasi dari seluruh pasangan berdasarkan input parameternya.
2. **Boundary value analysis:** Teknik ini berfokus pada pencarian error dari luar atau sisi dalam perangkat lunak.
3. **Cause-effect graph:** Berikutnya adalah teknik *cause-effect graph*. Teknik pengujian ini menggunakan grafik sebagai patokannya. Grafik ini menggambarkan relasi antara efek dan penyebab dari error.
4. **Equivalence partitioning:** Teknik ini bekerja dengan cara membagi data *input* dari beberapa perangkat lunak menjadi beberapa partisi data.
5. **Fuzzing:** *Fuzzing* merupakan teknik pencarian bug dalam perangkat lunak dengan memasukkan data yang tidak sempurna.
6. **Orthogonal array testing:** Teknik ini digunakan jika input berukuran kecil, akan tetapi cukup berat jika digunakan dalam skala yang besar.
7. **State transition:** Teknik ini berguna untuk melakukan pengujian terhadap mesin dan navigasi dari UI dalam bentuk grafik.

BAB III METODE PENELITIAN/PERANCANGAN

3.1 Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan sebagaimana ditunjukkan pada gambar 13 berikut



Gambar 13 Diagram tahapan penelitian

Berdasarkan diagram tahapan penelitian di atas, penelitian ini secara garis besar dapat dijabarkan sebagai berikut:

1. Studi Literatur

Pada tahap awal ini dilakukan pencarian informasi dan referensi- referensi penelitian sebelumnya yang terkait dengan informasi sistem penerjemah bahasa isyarat, metode pengolahan citra secara umum dan algoritma yang digunakan untuk klasifikasi.

2. Persiapan Instrumen Penelitian

Pada tahapan ini dibuat skenario pengambilan data dan persiapan peralatan atau instrumen yang digunakan dalam pengambilan data.