

BAB II TINJAUAN PUSTAKA

2.1. Dasar Teori

2.1.1. Bahasa Isyarat

Bahasa isyarat adalah bahasa non verbal yang umumnya digunakan oleh penyandang tunarungu untuk berkomunikasi. Umumnya bahasa ini menggabungkan antara bentuk, posisi, serta orientasi dari tangan, lengan, tubuh, bentuk bibir, atau ekspresi wajah untuk memberikan makna atau berkomunikasi kepada orang lain (Masood et al., 2018). Seperti dengan bahasa verbal umumnya, bahasa isyarat berbeda dan unik di setiap daerahnya bahkan untuk daerah dengan bahasa yang sama. Seperti Inggris dan Amerika Serikat yang menggunakan bahasa yang sama namun Inggris menggunakan *British Sign Language* sedangkan Amerika Serikat menggunakan *American Sign Language*.

2.1.2. Citra Digital

Citra digital adalah citra yang tersusun dari kumpulan matriks dua dimensi yang memiliki nilai terbatas yang disebut *pixel* (Shawal et al., 2014). Untuk mendapatkan citra digital diperlukan proses digitalisasi yang dimana proses digitalisasi ini akan mengubah gambar, teks, atau suara dari benda yang dapat dilihat menjadi data elektronik sehingga dapat disimpan dan digunakan lagi. Sebuah citra dapat didefinisikan sebagai fungsi 2 dimensi $i(x,y)$ dimana x dan y adalah titik koordinat pada bidang datar. Dalam sebuah komputer, citra digital dipetakan menjadi bentuk *grid* dan elemen *pixel* yang membentuk matriks 2 dimensi dimana setiap *pixel* merepresentasikan intensitas kecerahan sebuah titik serta memiliki nilai yang merepresentasikan posisi *pixel* dan nilai yang merepresentasikan komponen warna yang biasa disebut *channel*. Citra *grayscale* akan memiliki 1 channel warna sedangkan citra dengan format RGB akan memiliki 3 *channel* warna yang masing-masing merepresentasikan nilai intensitas (R) red, (G) Green, dan (B) blue untuk setiap *pixel*.

Sehingga citra digital dapat dijabarkan ke dalam matriks berikut :

$$I(x, y) = \begin{matrix} & P_{0,0}^c & P_{0,1}^c & \dots & P_{0,w-1}^c \\ P_{1,0}^c & P_{1,1}^c & \dots & P_{1,w-1}^c \\ \vdots & \vdots & \ddots & \vdots \\ P_{h-1,0}^c & P_{h-1,1}^c & \dots & P_{h-1,w-1}^c \end{matrix} \quad (2.1)$$

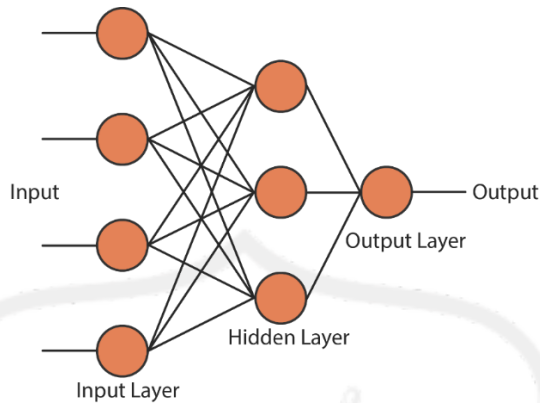
Nilai h , w dan c masing-masing melambangkan tinggi, lebar dan *channel* warna pada citra.

2.1.3. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) merupakan suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai Jaringan Syaraf Biologi (JSB) yang tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (O'Shea & Nash, 2015). Banyaknya masalah yang tidak dapat dirumuskan oleh algoritma biasa melatarbelakangi munculnya algoritma JST pertama dimana manusia ingin menggabungkan keunggulan otak manusia yang dapat belajar dan memberikan solusi serta kemampuan komputer dalam memproses dan menyimpan banyak data. Kemampuan otak manusia yang dapat melakukan kalkulasi harga yang sesuai serta penggunaan teknologi untuk meringankan beban manusia menjadikan JST banyak digunakan diberbagai bidang, baik pengenalan pola, prediksi ataupun klasifikasi.

Seperti halnya dengan otak manusia yang mampu belajar dan beradaptasi, jaringan syaraf tiruan juga menyelesaikan masalah dengan proses belajar menggunakan contoh kasus atau data yang diberikan pada proses pelatihan. Algoritma pada JST hanya dapat memproses data numerik karena beroperasi langsung dengan angka sehingga data yang tidak numerik harus diubah menjadi data numerik terlebih dahulu. Metode pelatihan yang sering digunakan adalah metode belajar terbimbing atau *supervised learning* dimana algoritma yang digunakan seakan akan dilatih untuk melakukan prediksi pada data latih yang diberikan sampai algoritma sudah mencapai atau mendekati target yang ditentukan. Pada proses belajar ini pola masukan disajikan bersama dengan pola keluaran yang

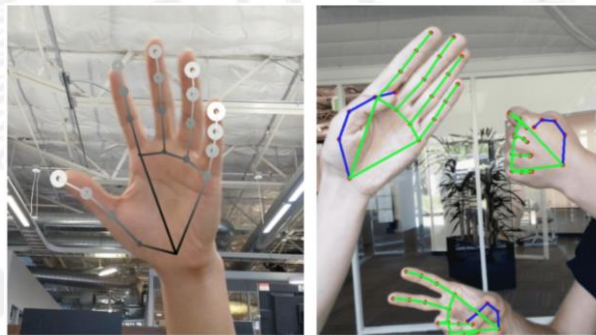
diinginkan. Jaringan akan menyesuaikan nilai bobot terhadap pola masukan serta sasaran yang diberikan.



Gambar 2. 1. Jaringan syaraf tiruan

2.1.4. MediaPipe: Hands

Mediapipe Hands adalah sebuah *library* pelacakan tangan dan jari secara *real time* dengan ketelitian tinggi menggunakan *Machine Learning* (ML) untuk menyimpulkan 21 landmark 3D tangan hanya dari satu frame gambar RGB.



Gambar 2. 2. Hasil prediksi landmark pada mediapipe hands

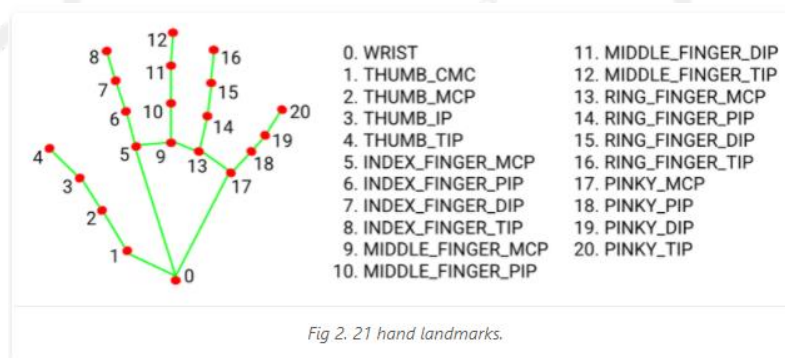
Mediapipe hands menggunakan *pipeline* ML yang terdiri dari dua model yang bekerja bersama:

1. BlazePalm Detector Model

Untuk mendeteksi lokasi awal tangan digunakan *single shot detector model* yang telah dioptimalkan pada penggunaan aplikasi *mobile*. Tangan yang dideteksi pada gambar input kemudian dikembalikan dengan kotak pembatas (*Ground Truth box*).

2. Hand Landmark Detector Model

Ground Truth box yang dihasilkan dari detektor telapak tangan digunakan untuk melokalisasi gambar tangan. Penggunaan gambar tangan yang dilokalisasi sesuai dengan kotak pembatas, secara drastis mengurangi kebutuhan untuk augmentasi data (misalnya rotasi, translasi, dan skala) sehingga memungkinkan jaringan untuk mengalokasikan sebagian besar kapasitas untuk menunjang akurasi prediksi *landmark* tangan. Model ini mempelajari representasi pose tangan internal yang konsisten dan kuat bahkan untuk tangan yang terlihat sebagian dan atau tertutupi oleh bagian tangan itu sendiri.



Gambar 2. 3. Landmark Pada Mediapipe hands

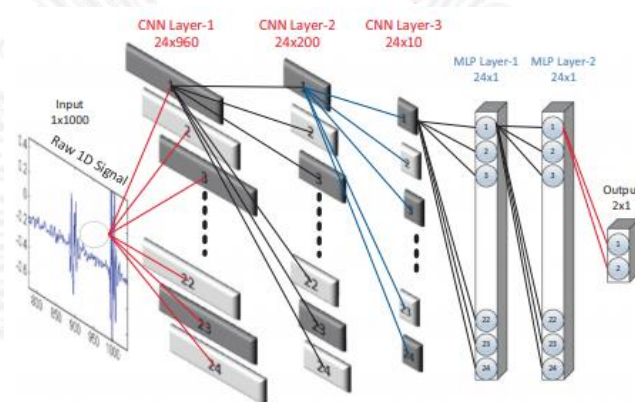
Model ini memiliki 3 output yang sebagai berikut :

1. *landmark* tangan yang terdiri dari x,y,dan kedalaman
2. *Hand presence* yang menunjukkan kehadiran tangan pada gambar input
3. Klasifikasi biner dari wewenang (seperti tangan kiri atau tangan kanan)

2.1.5. 1D Convolutional Neural Network

CNN adalah salah satu algoritma *machine learning* yang banyak digunakan saat ini karena terbukti dapat memberikan akurasi yang tinggi serta generalisasi model yang sangat baik. CNN konvensional dirancang untuk beroperasi secara eksklusif pada data 2D seperti gambar atau video sehingga dalam pemanfaatan CNN terhadap aplikasi pemrosesan sinyal 1D membutuhkan konversi 1D ke 2D yang tepat. Teknik yang umum digunakan adalah secara langsung membentuk kembali sinyal menjadi matriks $n \times m$ sehingga dapat dimasukkan ke dalam CNN konvensional. Namun, ada kelemahan dan keterbatasan tertentu dalam

menggunakan CNN sedalam itu. Terutama, diketahui bahwa CNN menimbulkan kompleksitas komputasi yang tinggi yang membutuhkan perangkat keras khusus terutama untuk pelatihan. Selain itu, pelatihan CNN mendalam yang tepat membutuhkan kumpulan data ukuran besar untuk pelatihan guna mencapai kemampuan generalisasi yang baik. Ini mungkin bukan pilihan yang baik untuk banyak aplikasi sinyal 1D. Untuk menggabungkan kelemahan tersebut (Kiranyaz et al., 2021) mengusulkan CNN 1D kompak dan adaptif. Dalam waktu yang relatif singkat, CNN 1D telah menjadi populer dengan kinerja yang canggih dalam berbagai aplikasi pemrosesan sinyal seperti deteksi dini aritmia dalam detak elektrokardiogram (EKG), pemantauan kesehatan struktural, dan kerusakan struktural.

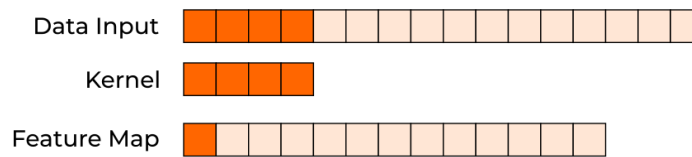


Gambar 2. 4. Arsitektur dengan 3 layer 1D CNN dan 2 MLP

2.1.5.1. *Convolutional Layer*

Pada proses pertama, citra masukan akan melalui *convolutional layer* dimana pada *layer* ini akan dilakukan operasi konvolusi antara matriks input dengan *kernel* pada matriks *filter*. konvolusi berarti pengaplikasian sebuah fungsi pada output fungsi lain yang dilakukan secara berulang. Dalam hal ini pengaplikasian sebuah fungsi yang dimaksud adalah *kernel* pada matriks *filter*. Lapisan konvolusi akan menentukan keluaran neuron-neuron yang terhubung ke daerah lokal input melalui perhitungan perkalian skalar antara bobotnya dan daerah yang terhubung dengan volume input (O'Shea & Nash, 2015). Hasil dari proses konvolusi pada algoritma CNN disebut juga sebagai *Feature map*. Pada CNN 1D kernel diletakkan pada data input dan menghasilkan *pixel* baru. Nilai *pixel* baru dihitung dengan

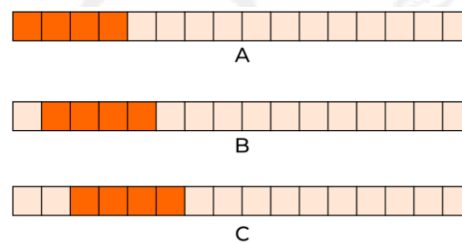
mengalikan setiap nilai *pixel* tetangga dengan bobot yang berhubungan pada *kernel* dan kemudian menjumlah hasil perkalian tersebut.



Gambar 2. 5. Proses Konvolusi

2.1.5.2. *Stride*

Stride adalah parameter yang mengatur jumlah pergeseran *filter* yang akan digunakan baik secara vertikal maupun horizontal. Pada CNN 1D pergeseran dilakukan terhadap 1 arah . (Yamashita et al., 2018) mendefinisikan *stride* sebagai jarak antara dua posisi kernel yang berurutan. Jika nilai *stride* ditetapkan 1 maka *filter* akan bergeser sebanyak 1.



Gambar 2. 6. Langkah Langkah Pada Stride

Bagian pada matriks citra yang terjadi proses konvolusi ditandai dengan area berwarna merah. Proses konvolusi akan bergeser sesuai nilai *stride* yang ditentukan. Untuk mendapatkan ekstraksi fitur yang maksimal, nilai *stride* yang digunakan harus disesuaikan dengan luas matriks atau menambahkan *padding* pada setiap sisi matriks. Dimensi *feature map* dapat digunakan untuk melihat kesesuaian *stride* yang digunakan. *Stride* dapat dikatakan sesuai apabila hasil output berupa bilangan *integer*.

$$output = \frac{W-N+2P}{s} + 1 \quad (2.2)$$

W = Panjang/ tinggi input

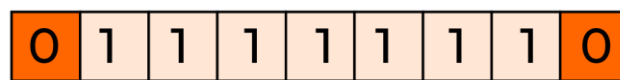
N= Panjang/ tinggi filter

P = Padding

S = Stride

2.1.5.3. *Padding*

Padding adalah parameter yang menentukan jumlah *pixel* di setiap sisi yang akan ditambahkan pada matriks input dan berisi nilai 0 sehingga tidak mengganggu nilai asli pada citra (Yamashita et al., 2018). *Padding* digunakan untuk memanipulasi *output* dari *feature map*. Karena dimensi *output* dari proses konvolusi selalu lebih kecil dari dimensi *input*, sementara *output* dari proses konvolusi akan digunakan lagi pada layer konvolusi berikutnya, sehingga informasi pada matriks banyak terbuang. Dengan menggunakan *padding*, matriks input akan dimanipulasi sehingga *filter* akan terfokus pada informasi di dalam *padding* dan mendapatkan ekstraksi fitur yang bisa meningkatkan performa *convolutional layer*.



Gambar 2. 7. Penambahan padding pada feature map

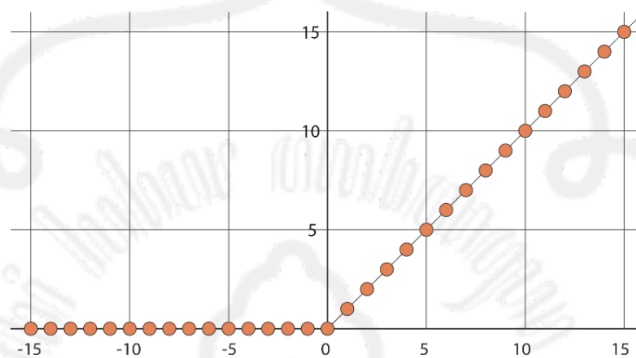
2.1.6. **Activation Function**

Activation Function atau fungsi aktivasi adalah fungsi yang digunakan dalam jaringan saraf untuk menghitung jumlah bobot input dan bias, yang digunakan untuk memutuskan apakah neuron dapat diaktifkan atau tidak (Nwankpa et al., 2018). Fungsi aktivasi memanipulasi data yang disajikan melalui beberapa pemrosesan gradien biasanya dan kemudian menghasilkan output untuk jaringan saraf yang berisi parameter dalam data. Fungsi aktivasi dapat berupa linier atau non-linier tergantung pada fungsi yang diwakilinya, dan digunakan untuk mengontrol output dari keluar jaringan saraf. Pilihan yang tepat dari fungsi aktivasi meningkatkan hasil dalam komputasi jaringan saraf (Nwankpa et al., 2018).

Fungsi aktivasi non linier memungkinkan penggunaan *multiple hidden layer* pada arsitektur *neural network*. Salah satu fungsi non linier yang sering digunakan adalah ReLU. Relu atau *Rectified Linear Unit* melakukan operasi *thresholding* nilai linear pada nol. Secara matematis dapat dituliskan sebagai berikut :

$$\sigma(x) = \max(0, x) \quad (2.3)$$

Walaupun dengan fungsi yang sederhana serta kemunculannya yang terhitung baru, fungsi ini menjadi populer serta menjadi sebuah standarisasi baru dalam implementasi CNN. (Krizhevsky et al., 2012) menjelaskan bahwa konvergensi proses pelatihan menggunakan fungsi ReLU lebih cepat hingga 6 kali dibandingkan dengan fungsi tanh. Meskipun begitu, fungsi ini memiliki kelemahan *saturating gradient* dikarenakan *range* nilai dibawah 0. Namun hal ini dapat dimitigasi dengan penggunaan *learning rate* yang sesuai pada proses pelatihan.



Gambar 2. 8. Distribusi fungsi ReLU

2.1.7. Subsampling

Subsampling berfungsi untuk mereduksi *input* secara spasial atau mengurangi jumlah parameter menggunakan operasi *down-sampling*. Berdasarkan arsitektur CNN, *layer pooling* terletak setelah *layer* konvolusi. *Layer pooling* terdiri dari sebuah *filter* dengan ukuran tertentu yang dioperasikan dengan *stride* tertentu terhadap *feature map* dari hasil *layer* konvolusi sebelumnya. Terdapat dua metode *pooling* yang biasa digunakan yaitu *average pooling* dan *max pooling* namun umumnya, metode *pooling* yang digunakan adalah *max pooling*. Output dari lapisan *max pooling* diberikan oleh aktivasi maksimum pada daerah persegi panjang yang tidak tumpang tindih. (Nagi et al., 2011). *Max pooling* membagi output dari *convolution layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi. Tujuan digunakan *pooling layer* pada CNN adalah untuk mereduksi ukuran citra sehingga dapat digantikan dengan sebuah *convolution layer* dengan *stride* yang sama dengan *pooling layer* yang bersangkutan (Springenberg et al., 2015).

2.1.8. Fully Connected Layer

Fully connected layer adalah jaringan *neural network* yang pada dasarnya sama dengan jaringan *neural network* biasa yaitu bisa dalam bentuk *single net* ataupun MLP. *Fully connected layer* menyatukan semua neuron untuk dilakukan proses klasifikasi menggunakan *neural network*. Dikarenakan *feature map* dari *feature learning* berupa *multidimensional array* sehingga perlu dilakukan *flatten* yang me-*reshape feature map* menjadi sebuah vektor sehingga bisa digunakan sebagai input untuk *fully connected layer*. Setelah proses *flatten*, semua bobot akan diklasifikasikan sesuai dengan banyaknya kelas yang sudah ditentukan. Pada *fully connected layer* tidak ada ketentuan penggunaan jaringan pasti sehingga jaringan *single net* atau MLP bisa digunakan pada *fully connected layer*. Namun beberapa penelitian menunjukkan penggunaan jaringan MLP dapat meningkatkan akurasi meskipun nilai peningkatan yang didapatkan tidak signifikan (Rosebrock, 2017).

2.2. Penelitian terkait

Penjelasan penelitian yang terkait pada penelitian ini disampaikan pada tabel 2. 1. serta dijelaskan sebagai berikut.

2.2.1. *Gesture Recognition of Sign Language Alphabet Using a Magnetic Positioning System*

Penelitian ini bertujuan mengembangkan metode dan algoritma untuk membangun sistem pengenalan bahasa isyarat yang mampu mengidentifikasi tanda-tanda yang sudah diberikan, menguraikan maknanya dan menghasilkan semacam keluaran tekstual, audio atau visual di tempat lain. Pada penelitian ini menggunakan sistem penentuan magnetik yang terdiri dari beberapa node transmisi yang dapat dikenakan, mengukur posisi 3D dan orientasi jari dalam volume operasi sekitar $30 \times 30 \times 30$ cm kemudian hasil dari sensor magnetik direkonstruksi menggunakan *mediapipe hands* dan *Support vector machine* (SVM) digunakan untuk mengenali gerakan statis yang terkait dengan alfabet bahasa isyarat yang telah dilatih sebelumnya. Dengan metode ini didapatkan nilai akurasi klasifikasi sekitar 97%.

2.2.2. *Using Deep Convolutional Networks for Gesture Recognition in American Sign Language*

Penelitian ini bertujuan untuk membuat sistem pengenalan ejaan (ASL) yang dapat mempermudah proses ejaan nama, alamat, gelar, dan merk namun minimnya pengetahuan masyarakat tentang bahasa isyarat. Peneliti ini menggunakan *deep CNN* serta arsitektur CNN umum yang sering digunakan dalam mengklasifikasi gambar. Digunakan kumpulan 25 gambar dari 5 orang berbeda untuk masing-masing huruf ASL dan angka sebagai *dataset* pelatihan. Karena *dataset* yang dibuat tidak terkontrol sehingga rentan terhadap perbedaan cahaya, warna kulit, dan lingkungan sehingga digunakan juga *dataset* yang tersedia untuk dilakukan perbandingan terhadap *dataset* yang dibuat. Dengan metode yang digunakan, didapatkan nilai akurasi masing-masing 82,5% dan 97% untuk validasi huruf dan angka pada *dataset* yang tersedia (NZ ASL), serta nilai akurasi yang

sedikit lebih rendah pada *dataset* ASL yang dibuat sendiri yaitu 67% dan 70% untuk masing-masing akurasi pada validasi huruf dan angka.

2.2.3. A real-time american sign language recognition system using convolutional neural network for real datasets

Penelitian ini bertujuan untuk membuat sistem pengenalan bahasa isyarat dengan akurasi tinggi yang bisa diterapkan dan dijangkau oleh masyarakat luas. Penelitian ini menggunakan algoritma CNN dan arsitektur VGG net yang dilatih pada *dataset* ASL dimana *dataset* ini terdiri dari 26 kelas huruf (a-z) dan 2 kelas untuk spasi dan *delete* yang diambil menggunakan *webcam* dengan kualitas normal serta pencahayaan berbeda, warna kulit, latar belakang, dan berbagai macam situasi. Untuk *dataset* validasi yang digunakan sekitar 30% dari *dataset training*, sehingga total yang digunakan sebanyak 43.120 gambar untuk pelatihan, dan 18.480 gambar untuk *dataset* validasi. Dengan metode yang digunakan, didapatkan nilai akurasi yang tinggi untuk semua pelatihan termasuk data baru yang tidak disertakan dalam pengujian, dengan rata rata 98,53% pada *dataset* pelatihan dan 98,84% pada *dataset* pengujian.

2.2.4. MediaPie Hands: On-device Real-time Hand Tracking

Penelitian ini bertujuan untuk Membuat solusi baru yang tidak memerlukan perangkat keras tambahan dan bekerja secara *real time* di perangkat seluler. Penelitian ini menggunakan 2 model yang bekerja Bersama. Detektor telapak tangan yang akan mendeteksi telapak tangan dan memberikan *ground truth box* yang berisi gambar tangan dari gambar yang diberikan. Dan model yang kedua adalah *landmark model* yang beroperasi pada *ground truth* yang diberikan untuk mendeteksi 21 *landmark* tangan. Dengan menyediakan gambar tangan yang sudah dipangkas berdasarkan *ground truth box* yang dihasilkan oleh detektor telapak tangan, secara drastis mengurangi kebutuhan untuk augmentasi data (seperti rotasi, translasi, dan skala) dan memungkinkan jaringan untuk mendedikasikan sebagian besar kapasitasnya untuk akurasi lokalisasi *landmark*. dalam skenario *real time* digunakan *ground truth box* dari *frame* sebelumnya untuk melokalisasi gambar pada *frame* saat ini sehingga menghindari penerapan detektor tangan pada setiap

frame agar model bisa bekerja lebih ringan. Detektor telapak tangan akan dijalankan jika gambar tidak bisa memprediksi tangan berdasarkan *ground truth box* dari *frame* sebelumnya.

Model ini dilatih menggunakan *dataset real word* dan *dataset* sintetis. *Dataset real word* berisi kumpulan data dengan keragaman geografis serta mencakup berbagai pose tangan yang memungkinkan sementara dataset sintesis berisi pose tangan dengan pemetaan 3D sehingga kedua dataset saling melengkapi. Didapatkan hasil yang dapat memprediksi *landmark* tangan secara *real time* serta dapat dengan mudah diterapkan diberbagai jenis perangkat.

2.2.5. A 1D-CNN Based Deep Learning Technique for Sleep Apnea Detection in IoT Sensors

Penelitian ini bertujuan untuk pengembangan metode *deteksi sleep apnea non intrusif* secara otomatis karena secara tradisional gangguan yang berhubungan dengan tidur didiagnosis menggunakan polisomnografi semalam dibawah pengawasan seorang dokter. Merekam polisomnogram untuk evaluasi mahal dan tidak nyaman bagi pasien. Penelitian ini Mendeteksi *sleep apnea* pada basis detik dengan cara mempertimbangkan satu *frame* data berisi 11 detik sinyal elektrokardiogram. Data pasien diklasifikasikan sebagai apnea jika pasien tidak bernafas setidaknya selama 10 detik sehingga akan tumpang tindih dengan *frame* data 11 detik sebagai inputan. 1D CNN digunakan untuk klasifikasi serta tidak ada penyaringan atau pemrosesan sinyal yang dilakukan sebelum tahap 1D-CNN. 3 model dianalisis untuk mengurangi ukuran model agar sesuai untuk lingkungan dengan sumber daya terbatas dan membandingkan kinerja diantaranya. Metode yang diusulkan mencapai akurasi 99,56%, model yang dipangkas mencapai akurasi 97,34%, model biner menunjukkan akurasi 75,59%, dan model khusus pasien mencapai akurasi rata-rata 97,79%.

Tabel 2. 1. Rangkuman penelitian terkait

NO	JUDUL JURNAL, SITASI	MASALAH	TUJUAN	METODE	HASIL	KETERKAITAN
1	<i>Gesture Recognition of Sign Language Alphabet Using a Magnetic Positioning System</i> (Rinalduzzi et al., 2021).	Pembangunan pengenalan bahasa isyarat yang tidak terlalu bergantung terhadap keluaran langsung dari sensor	mengembangkan metode dan algoritma untuk membangun sistem pengenalan bahasa isyarat yang mampu mengidentifikasi tanda-tanda yang sudah diberikan, menguraikan maknanya dan menghasilkan semacam keluaran tekstual, audio atau visual di tempat lain.	Menggunakan sistem penentuan magnetik yang terdiri dari beberapa node transmisi yang dapat dikenakan, mengukur posisi 3D dan orientasi jari dalam volume operasi sekitar $30 \times 30 \times 30$ cm kemudian direkonstruksi menggunakan <i>mediapipe hands</i> dan <i>Support vector machine</i> (SVM) untuk mengenali gerakan statis yang terkait dengan alfabet bahasa isyarat	Metode yang diusulkan memperoleh akurasi klasifikasi sekitar 97%	Memberikan pemahaman tentang penggunaan koordinat tangan yang dihasilkan dari sistem penentuan posisi magnetik sebagai input untuk <i>machine learning</i>
2	<i>Using Deep Convolutional</i>	Penggunaan pengejaan jari yang	Membuat pengenalan huruf dan angka pada	Menggunakan <i>deep CNN</i> dengan arsitektur umum	Mendapatkan tingkat akurasi 82,5% dan	Memberikan penjelasan tentang

	<i>Networks for Gesture Recognition in American Sign Language</i> (Bheda & Radpour, 2017).	dapat membantu untuk mengeja nama, alamat, gelar, dan merk namun minimnya pemahaman tentang bahasa isyarat	<i>American sign language</i> menggunakan jaringan konvolusi	yang digunakan dan melakukan <i>training</i> mulai dari awal serta melakukan perbandingan dengan dua dataset ASL yaitu dataset ASL yang dibuat sendiri dan dataset ASL yang sudah tersedia (NZ ASL)	97% masing-masing untuk validasi huruf dan angka pada dataset NZ ASL, 67% dan 70% untuk alfabet dan angka pada dataset yang dibuat sendiri	pengenalan bahasa isyarat yang di <i>train</i> dari awal menggunakan arsitektur jaringan CNN yang dalam
3	<i>A real-time american sign language recognition system using convolutional neural network for real datasets</i> (Kadhim & Khamees, 2020).	Sistem pengenalan bahasa isyarat dengan tingkat akurasi tinggi masih sulit dijangkau dan diaplikasikan pada masyarakat luas karena penggunaan perangkat pendukung	Pembuatan sistem pengenalan bahasa isyarat yang menggunakan kamera PC sebagai inputnya sehingga mudah dijangkau dan diterapkan	Menggunakan algoritma CNN dan arsitektur VGG yang latih pada dataset ASL yang terdiri dari 28 kelas (26 huruf (A-Z) ditambah spasi dan delete) yang diambil menggunakan <i>webcam</i> dengan kualitas normal serta pencahayaan dan posisi yang berbeda	Didapatkan nilai akurasi yang tinggi untuk semua pelatihan termasuk data baru yang tidak disertakan dalam pengujian, dengan rata rata 98,53% pada dataset pelatihan dan 98,84% pada dataset pengujian	Memberikan pemahaman tentang klasifikasi terutama pada data pengujian baru yang tidak disertakan pada pelatihan

4	<i>MediaPipe Hands: On-device Real-time Hand Tracking</i> (Zhang et al., 2020).	Sebagian besar pekerjaan sebelumnya memerlukan perangkat keras khusus, seperti. sensor kedalaman. Solusi lain tidak cukup ringan untuk dijalankan secara <i>real time</i> dan dengan demikian terbatas pada <i>platform</i> yang dilengkapi dengan prosesor yang kuat.	Membuat solusi baru yang tidak memerlukan perangkat keras tambahan dan bekerja secara <i>real time</i> di perangkat seluler	Membuat solusi pelacakan tangan dengan 2 model yang bekerja sama yaitu detektor telapak tangan yang akan mendeteksi telapak tangan dan memberikan <i>ground truth box</i> yang berisi gambar tangan dari gambar yang diberikan. Dan model yang kedua adalah <i>landmark</i> model yang beroperasi pada <i>ground truth</i> yang diberikan untuk mendeteksi 21 <i>landmark</i> tangan serta model dilatih dengan dataset <i>real word</i> dan dataset sintetis	Didapatkan model yang dapat memprediksi <i>landmark</i> tangan secara <i>real time</i> serta dapat dengan mudah diterapkan di berbagai perangkat	Memberikan model alternatif untuk mengekstrak <i>features</i> gambar tangan (dalam hal ini 21 <i>landmark</i> tangan) tanpa menggunakan perangkat tambahan khusus
5	<i>A 1D-CNN Based Deep Learning</i>	Secara tradisional gangguan yang berhubungan	pengembangan metode deteksi <i>sleep</i>	Mendeteksi <i>sleep apnea</i> pada basis detik dengan cara mempertimbangkan satu	3 model dianalisis untuk mengurangi ukuran model agar	Memberikan pemahaman Tentang

	<p><i>Technique for Sleep Apnea Detection in IoT Sensors</i> (John et al., 2021).</p>	<p>dengan tidur didiagnosis menggunakan polisomnografi semalam di bawah pengawasan seorang dokter. Merekam polisomnogram untuk evaluasi mahal dan tidak nyaman bagi pasien.</p>	<p><i>apnea non intrusif</i> secara otomatis</p>	<p><i>frame</i> data berisi 11 detik sinyal elektrokardiogram. Data pasien diklasifikasikan sebagai apnea jika pasien tidak bernafas setidaknya selama 10 detik sehingga akan tumpang tindih dengan <i>frame</i> data 11 detik sebagai inputan. 1D CNN digunakan untuk klasifikasi serta Tidak ada penyaringan atau pemrosesan sinyal yang dilakukan sebelum tahap 1D-CNN</p>	<p>sesuai untuk lingkungan dengan sumber daya terbatas dan membandingkan kinerja di antaranya. Metode yang diusulkan mencapai akurasi 99,56%, model yang dipangkas mencapai akurasi 97,34%, model biner menunjukkan akurasi 75,59%, dan model khusus pasien mencapai akurasi rata-rata 97,79%.</p>	<p>implementasi 1D CNN serta penggunaan model 1D CNN yang dangkal untuk melakukan klasifikasi pada lingkungan dengan sumber daya yang terbatas.</p>
--	---	---	--	---	--	---