

BAB II

TINJAUAN PUSTAKA

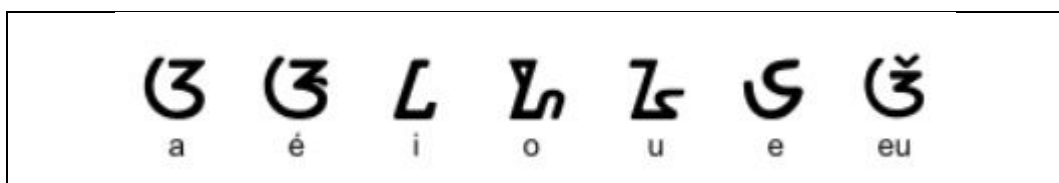
2.1 Landasan Teori

2.1.1 Aksara Sunda

Aksara Sunda merupakan huruf yang digunakan oleh suku Sunda untuk menuliskan kata-kata yang digunakan dalam bahasa Sunda. Aksara Sunda merupakan hasil tradisi ortografi atau sistem ejaan suatu bahasa melalui perjalanan sejarah mulai dari abad ke-5 hingga kini. Aksara Sunda mulai ditinggalkan masyarakat karena kondisi jaman kolonial hingga kemerdekaan, masyarakat diminta untuk meninggalkan penggunaan aksara Sunda sebagai salah satu identitas budaya Sunda. Akhirnya, aksara Sunda hampir tidak digunakan lagi oleh masyarakat saat ini (Amalia dkk. 2020). Aksara Sunda terdapat beberapa jenisnya, salah satunya adalah aksara swara atau vokal.

a. Aksara Sunda Swara

Aksara sunda swara merupakan aksara huruf terdiri dari 7 huruf vokal yaitu a, é, i, o, u, e, dan eu.



Gambar 2.1 Aksara Sunda swara (Amalia dkk., 2020)

2.1.2 Citra

Citra adalah fungsi dari intensitas cahaya yang digambarkan dalam bidang dua dimensi, citra merupakan suatu gambaran, kemiripan atau imitasi dari suatu objek. Citra dapat didefinisikan sebagai fungsi $f(x, y)$ berukuran M baris serta N kolom, x dan y merupakan koordinat *spatial* dan *amplitude* f di titik koordinat (x, y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut (Darma, 2010). Berdasarkan karya ilmiah dari (McAndrew, 2016), citra terbagi menjadi dua yaitu citra yang bersifat analog dan citra yang bersifat digital.

a. Citra analog

Citra analog merupakan citra yang dibentuk dari sinyal analog yang bersifat kontinu. Gambaran yang tertangkap oleh mata manusia dan foto atau filem yang tertangkap oleh kamera analog merupakan contoh dari citra analog. Citra tersebut memiliki tingkat kerincian yang sangat baik tetapi tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses di komputer secara langsung. Oleh sebab itu, agar dapat diproses di komputer harus dilakukan proses konversi dari analog ke digital.

b. Citra digital

Citra digital merupakan citra yang dapat diolah oleh komputer, citra tersebut terbentuk dari sinyal diskrit. Citra digital dapat diperoleh dengan mengambil gambar objek menggunakan kamera digital atau melakukan *scanning* gambar analog/fisik menggunakan *scanner*.

Berdasarkan kombinasi warna pada piksel, citra digolongkan menjadi 3 jenis yaitu citra biner, citra *grayscale* dan citra warna. Nilai suatu piksel memiliki rentang nilai tertentu, jangkauan yang digunakan berbeda-beda tergantung dari jenis warna (McAndrew, 2016).

a. Citra berwarna

Setiap piksel pada citra berwarna mewakili warna yang merupakan kombinasi dari tiga warna dasar yaitu RGB (*Red, Green dan Blue*). Masing- masing warna mempunyai 8bit dengan *range* nilai 0-255. Misalnya warna biru memiliki gradasi warna sebanyak 256 dari mulai hitam kebiruan sampai dengan biru keputih-putihan, hal tersebut berlaku untuk warna lainnya. Berarti setiap piksel memiliki kombinasi warna sebanyak $256 \times 256 \times 256 = 16,7$ juta.

b. Citra *grayscale*

Citra *grayscale* adalah citra yang nilai intensitas pikselnya didasarkan pada derajat keabuan. Pada citra *grayscale*, tingkatan warna hitam sampai dengan putih dibagi ke dalam 256 derajat keabuan dimana warna hitam sempurna direpresentasikan dengan nilai 0 dan putih sempurna dengan nilai 255. Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna. Semakin besar jumlah bit warna yang disediakan di memori, maka gradasi warna yang terbentuk akan semakin halus.

c. Citra biner

Citra biner merupakan citra yang memiliki dua kemungkinan nilai yaitu hitam dan putih. Citra ini dikenal juga dengan citra *Black and White* (B&W) atau

citra *monochrome*. Dibutuhkan 1 bit di memori untuk menyimpan citra biner. Citra biner seringkali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengembangan, morfologi, ataupun *dithering*.

2.1.3 Pengolahan Citra

Menurut Efford (2000), pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Tujuan dari pengolahan citra pada awalnya adalah untuk memperbaiki kualitas citra, namun seiring dengan perkembangan dunia komputasi yang ditandai dengan meningkatnya kapasitas serta kecepatan proses komputer dan munculnya ilmu komputasi yang memungkinkan kita untuk mengambil informasi dari suatu citra, maka pengolahan citra tidak dapat dilepaskan dengan bidang computer vision. Bahkan dalam perkembangan lebih lanjut, pengolahan citra dan computer vision seolah-olah sebagai mata manusia dengan perangkat input image capture seperti kamera dan scanner sebagai mata dan komputer sebagai otak yang dapat mengelola informasi dan citra. Sehingga muncul beberapa bidang yang penting dalam computer vision diantaranya pengenalan pola, *biometric*, *content base image and video retrieval*, *video editing* dan lainnya (Learned-Miller, 2012).

Tujuan utama dari pengolahan citra sesuai dengan perkembangan computer vision adalah untuk memperbaiki kualitas suatu gambar sehingga dapat lebih mudah diinterpretasikan oleh mata manusia, juga untuk mengolah informasi yang terdapat pada suatu citra untuk keperluan pengenalan objek secara otomatis.

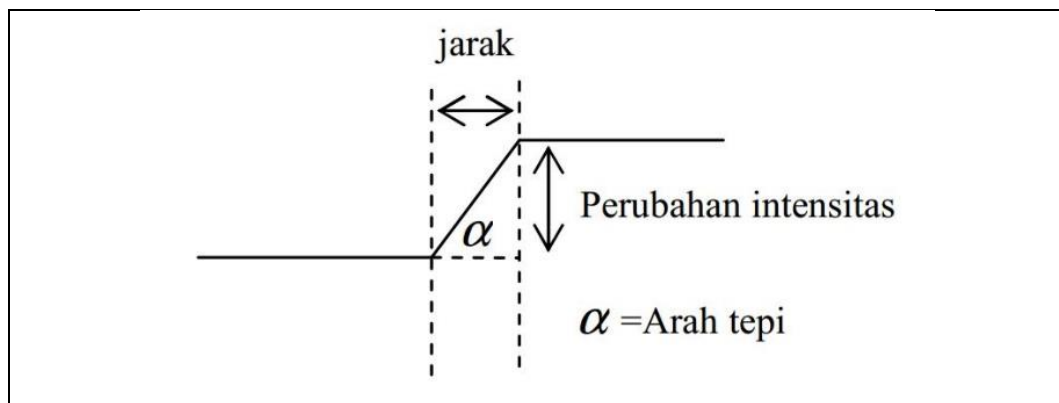
2.1.4 Pengenalan Pola

Pengenalan pola banyak digunakan di berbagai bidang sesuai dengan fungsinya. Secara umum pengenalan pola adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu objek (Faturrahman, dkk., 2018). Sistem pengenalan pola pada dasarnya meliputi tiga tahap berikut: (1) akuisisi data; (2) pra-pengolahan data; dan (3) pembuatan keputusan (Riansyah, dkk., 2017). Pola dapat dinyatakan dengan notasi vektor ataupun matriks.

Manusia dapat mengenali suatu objek yang dilihatnya karena otak manusia telah belajar untuk mengklasifikasikan objek yang ada di alam sehingga mampu membedakan objek satu dengan lainnya. Kemampuan sistem visual manusia inilah yang coba ditirukan oleh mesin untuk pengenalan suatu pola. Komputer menerima input berupa citra objek yang akan diidentifikasi, memproses citra, sehingga menghasilkan output berupa informasi dari citra tersebut. Pengenalan pola dapat diimplementasikan untuk mengenali huruf (Learned-Miller, 2012).

2.1.5 Deteksi Tepi

Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang mendadak besar dalam jarak yang dekat. Suatu titik (x, y) dikatakan sebagai tepi bila titik tersebut mempunyai perbedaan nilai piksel yang tinggi dengan nilai piksel tetangganya (Putra dan Ni, 2014). Gambar 2.2 menunjukkan salah satu model tepi untuk satu dimensi.



Gambar 2.2 Model tepi dalam satu dimensi (Putra dan Ni, 2014)

Deteksi tepi pada suatu citra menghasilkan tepi-tepi dari objek citra yang bertujuan untuk menandai bagian yang menjadi detail citra, juga memperbaiki detail citra yang kabur karena *error* atau adanya efek dari proses akuisisi citra.

2.1.5.1 Deteksi Tepi Canny

Metode deteksi tepi canny adalah metode deteksi tepi yang akan menghasilkan tampilan gambar yang berbeda dari semua metode karena menampilkan efek *relief* di dalamnya. Efek *relief* adalah seperti sebuah tampilan batu kasar yang diukir, yaitu garis-garis kasar yang membentuk sebuah penggambaran objek di dalamnya (Putra dan Ni, 2014).

Kelebihan metode Canny adalah kemampuan untuk mengurangi *noise* sebelum melakukan perhitungan tepi sehingga tepi yang dihasilkan akan lebih banyak. Algoritma ini memberikan tingkat kesalahan rendah, melokalisasi titik-titik tepi (jarak piksel-piksel tepi yang ditemukan deteksi dan tepi sesungguhnya sangat pendek), dan hanya memberikan satu tanggapan untuk satu tepi. Ada beberapa

kriteria pendeteksi tepian paling optimum yang dapat dipenuhi oleh operator Canny (Hermana dan Meikel, 2015),

- a. Mendeteksi dengan baik (kriteria deteksi).
- b. Melokalisasi dengan baik (kriteria lokalisasi).
- c. Respon yang jelas (kriteria respon).

Algoritma Canny bekerja melalui lima tahap berbeda yaitu diawali dari penghalusan, pencarian gradien, redaman non-maksimum, pengambangan ganda dan pelacakan tepi dengan histeresis. Langkah-langkah ini digunakan untuk menerapkan deteksi tepi yang kompleks (Maximillian dkk., 2023):

- a. Penghalusan (*Smoothing*)

Proses ini berjalan dengan mengaburkan citra untuk menghilangkan *noise* yang terdeteksi. Pemulusan citra dilakukan dengan formula berikut :

$$G(i, j) = \frac{1}{2\pi\sigma^2} \times e^{-\frac{(i-u)^2 + (j-v)^2}{2\sigma^2}}$$

Gambar 2.3 Formula proses penghalusan

Keterangan :

e = 2,71 (konstanta euler)

σ = simpangan baku (sigma)

π = 3,14 (pi)

b. Pencarian gradien (*Finding gradient*)

Setelah menghilangkan *noise* pada proses penghalusan, proses selanjutnya adalah menentukan resistansi tepi. Tepi harus ditandai pada gambar dengan gradien besar. Operator gradien digunakan dan pencarian dilakukan secara horizontal dan vertikal.

$$G_x \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} G_y \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix}$$

Gambar 2.4 Operator horizontal dan vertikal algoritma canny

Hasil dari kedua operator ini digabungkan untuk memberikan hasil gabungan dari sisi vertikal dan horizontal menggunakan formula berikut :

$$G = \sqrt{G_x^2 + G_y^2}$$

Gambar 2.5 Formula gabungan operator horizontal dan vertikal

Kemudian arah tepian yang ditentukan ditemukan dengan menggunakan persamaan berikut :

$$\theta = \arctan \left(\frac{G_y}{G_x} \right)$$

Gambar 2.6 Formula menentukan arah tepian

c. Non-maksimum

Penghapusan penghilangan yang kurang maksimal dilakukan di sepanjang tepi, menghilangkan piksel yang dibuang sebagai tepi gambar. Hanya nilai maksimum yang ditandai sebagai tepi. Ini membuat ujungnya lebih ramping.

d. Pengambangan ganda (*Double thresholding*)

Untuk membuat citra biner, bagi menjadi dua kondisi dimana nilai di bawah T1 diubah menjadi hitam dengan nilai 0, dan nilai di atas T2 diubah menjadi putih dengan nilai 255. Kedua kondisi ini menentukan ambang rendah (T1) dan ambang tinggi (Q2).

e. Pelacakan tepi dengan histeresis (*Edge Tracking by Hysteresis*)

Tepi terakhir ditentukan dengan menekan semua tepi yang tidak terhubung dengan tepi yang kuat. Piksel apa pun terhubung ke piksel putih dan memiliki nilai lebih besar dari T1 juga dianggap sebagai tepi.

2.1.5.2 Deteksi Tepi Sobel

Deteksi tepi Sobel adalah salah satu metode yang menghindari adanya perhitungan gradien di titik interpolasi. Metode Sobel menggunakan 2 matriks berukuran 3x3 yaitu matriks Gx dan Gy. Kedua matriks tersebut digunakan untuk menghitung perbedaan warna pada piksel yang sedang dihitung dengan piksel di sekitarnya secara horisontal dan vertikal. Matriks Gx digunakan untuk menghitung secara horisontal dan matriks Gy secara vertikal (Faturrahman dkk., 2018). Kelebihan dari metode Sobel ini adalah kemampuan untuk mengurangi noise

sebelum melakukan perhitungan deteksi tepi. Matriks metode Sobel dapat dilihat pada Gambar 2.7.

-1	0	+1
-2	0	+2
-1	0	+1
Gx		

+1	+2	+1
0	0	0
-1	-2	-1
Gy		

Gambar 2.7 Matriks metode deteksi tepi Sobel (Faturrahman dkk., 2018)

Adapun formula *operator* dari deteksi tepi Sobel dapat dilihat pada Gambar 2.8 di bawah. Berdasarkan penempatan piksel tetangga, besarnya gradien dihitung menggunakan *operator* berikut.

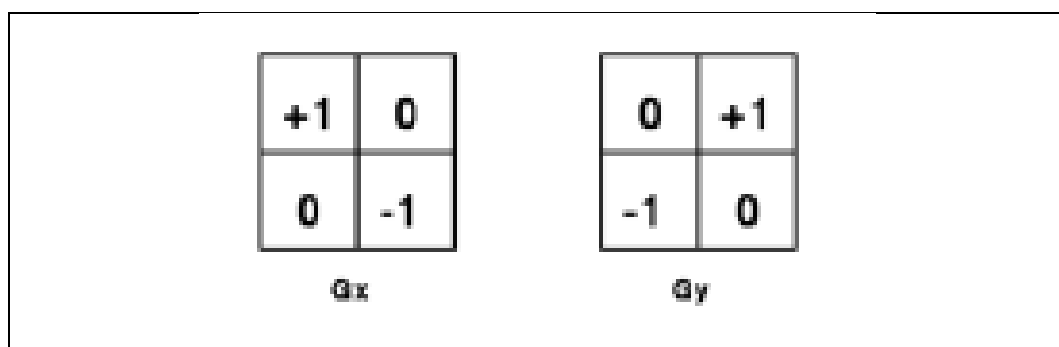
$$G = \sqrt{S_x^2 + S_y^2}$$

Gambar 2.8 Formula deteksi tepi Sobel (Maximillian dkk., 2023)

Merujuk dari rumus tersebut, diketahui bahwa G adalah besarnya gradien dari *operator* Sobel, S_x (atau G_x) adalah gradien Sobel mendatar, S_y (atau G_y) adalah gradien Sobel vertikal.

2.1.5.3 Deteksi Tepi Robert

Metode Robert adalah nama lain dari teknik diferensial yang dikembangkan, yaitu diferensial pada arah horizontal dan diferensial pada arah vertikal, dengan ditambahkan proses konversi biner setelah dilakukan diferensial. Metode Robert dikenal juga dengan istilah operator Robert Cross (diagonal) yang menggunakan kernel ukuran 2x2 piksel, sehingga tepi yang dihasilkan berada pada tepi atas atau tepi bawah (Krisna Putra dan Ayu Wirdiani, 2014). Matriks metode Robert dapat dilihat pada gambar 2.9 berikut.



Gambar 2.9 Matriks metode deteksi tepi Robert (Krisna dan Ayu, 2014)

2.1.5.4 Deteksi Tepi Prewitt

Metode Prewitt merupakan pengembangan metode Robert dengan menggunakan filter HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi *laplacian* yang dikenal sebagai fungsi untuk membangkitkan HPF (*High Pass Filter*).

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } H = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

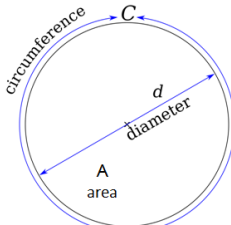
Gambar 2.10 Matriks metode deteksi tepi Prewitt (Krisna dan Ayu, 2014)

2.1.6 Ekstraksi Ciri (Morfologi)

Ekstraksi ciri citra adalah tahapan mengekstrak ciri atau informasi dari objek di dalam citra yang ingin dikenali ataupun dibedakan dengan objek lainnya. Untuk membedakan bentuk aksara sunda satu dengan lainnya, dapat menggunakan parameter *metric* dan *eccentricity* (Amalia dkk., 2020).

a. *Metric*

Metric memiliki rentang nilai antara 0 hingga 1. Objek yang berbentuk memanjang/mendekati bentuk garis lurus, nilai *metric*-nya mendekati angka 0, sedangkan objek yang berbentuk bulat/lingkaran, nilai metriknya mendekati angka 1. Penghitungan metrik diilustrasikan pada gambar berikut.



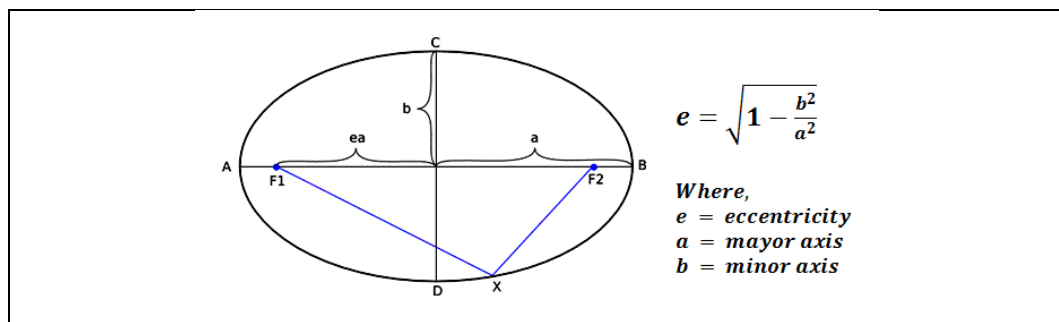
$$M = \frac{4\pi \times A}{C^2}$$

Where,
 M = Metric
 A = Area
 C = Circumference

Gambar 2.11 Perhitungan dalam morfologi metrik (Amalia dkk., 2020)

b. *Eccentricity*

Eccentricity merupakan nilai perbandingan antara jarak foci ellips minor dengan foci ellips mayor suatu objek. *Eccentricity* memiliki rentang nilai antara 0 hingga 1. Objek yang berbentuk memanjang/mendekati bentuk garis lurus, nilai *eccentricity*-nya mendekati angka 1, sedangkan objek yang berbentuk bulat atau lingkaran, nilai *eccentricity*-nya mendekati angka 0. Penghitungan *eccentricity* diilustrasikan pada gambar berikut.



Gambar 2.12 Perhitungan *eccentricity* (Amalia dkk., 2020)

2.1.7 Jaringan Saraf Tiruan (JST)

Jaringan saraf tiruan dibuat pertama kali pada tahun 1943 oleh *neurophysiologist* Warren McCulloch dan *logician* Walter Pitts. Jaringan saraf tiruan terinspirasi dari sistem sel saraf biologi, sama seperti otak manusia yang dapat memproses suatu informasi. Jaringan saraf tiruan ditentukan oleh 3 hal (Hermana dan Meikel, 2015),

- a. Pola hubungan antar neuron (disebut arsitektur jaringan).
- b. Metode untuk menentukan bobot penghubung (disebut metode *learning/training*).

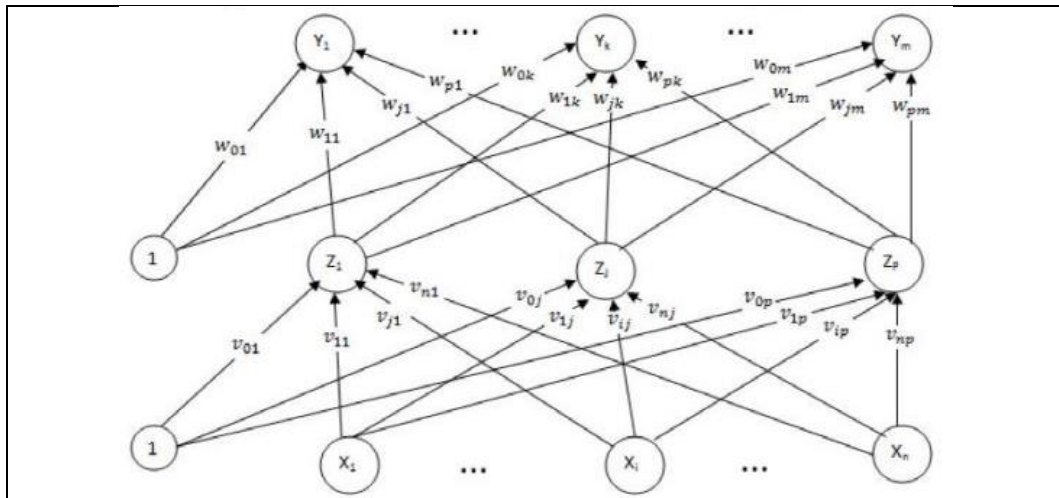
c. Fungsi aktivasi.

Menurut (Fausett, 1994), jaringan saraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan saraf biologi dengan asumsi sebagai berikut,

- a. Pemrosesan informasi terjadi pada banyak elemen sederhana (*neurons*).
- b. Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung.
- c. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.
- d. Untuk menentukan keluaran, setiap neuron menggunakan fungsi aktivasi yang dikenakan pada penjumlahan masukan yang diterima. Besarnya keluaran ini selanjutnya dibandingkan dengan suatu batas ambang.

2.1.8 *Backpropagation*

Jaringan saraf tiruan *backpropagation* merupakan algoritma pembelajaran terawasi yang terdiri dari beberapa layer (*multilayer*) yaitu *input layer*, *hidden layer* dan *output layer*. Jaringan saraf tiruan *backpropagation* mampu mengenali pola yang akan digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan (Faturrahman, dkk., 2018). Metode *backpropagation* menggunakan *error* keluaran untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Metode ini memiliki dasar matematis yang kuat, objektif dan algoritma ini mendapatkan bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah kuadrat galat *error* melalui model yang dikembangkan (*training set*).

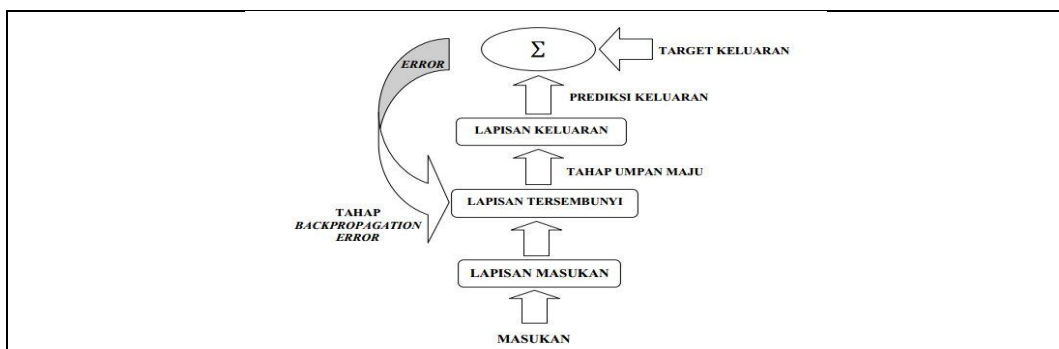


Gambar 2.13 Ilustrasi JST *backpropagation* (Faturrahman dkk., 2018)

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi. Gambar di atas ini merupakan arsitektur *backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah layer tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran (Faturrahman dkk., 2018).

a. Jalan kerja algoritma

Istilah *backpropagation* diambil dari cara kerja jaringan ini, berikut merupakan alur kerja jaringan *backpropagation*.



Gambar 2.14 Jalan kerja JST *backpropagation* (Faturrahman dkk., 2018)

Cara kerja jaringan *backpropagation*, pertama jaringan diinisialisasi dengan bobot yang diset dengan bilangan acak. Kemudian sampel pelatihan dimasukkan ke dalam jaringan. Sampel pelatihan terdiri dari pasangan vektor masukan dan vektor target keluaran. Keluaran dari jaringan berupa vektor prediksi keluaran. Selanjutnya vektor keluaran hasil jaringan atau prediksi keluaran dibandingkan dengan target keluaran, untuk mengetahui apakah jaringan keluaran sudah sesuai dengan yang diharapkan (prediksi keluaran sudah sama dengan target keluaran).

Error yang dihasilkan akibat adanya perbedaan antara prediksi keluaran dengan target keluaran tersebut kemudian dihitung untuk meng-*update* bobot-bobot koneksi yang relevan dengan jalan mempropagasikan kembali *error*. Setiap perubahan bobot yang terjadi diharapkan dapat mengurangi besarnya *error*. Proses akan terus berlanjut sampai kerja jaringan mencapai tingkat yang diinginkan atau sampai kondisi perhentian dipenuhi. Pada umumnya kondisi perhentian yang sering digunakan adalah jumlah iterasi atau *error*. Iterasi akan berhenti jika jumlah iterasi yang dilakukan jaringan telah melebihi jumlah iterasi yang ditentukan, atau jika nilai *error* yang didapat lebih kecil dari batas toleransi.

b. Sistem pelatihan

Backpropagation merupakan salah satu algoritma pelatihan dengan supervisi, berarti dalam proses pelatihan terdapat target yang akan dibandingkan dengan output yang akan dihasilkan. Menurut (Siang, 2005), pelatihan *backpropagation* meliputi 3 fase yaitu sebagai berikut :

1) Fase 1 (*Forward propagation*)

Pola masukan dihitung maju mulai dari lapisan masukan hingga lapisan keluaran menggunakan fungsi aktivasi yang ditentukan.

2) Fase 2 (*Backwards propagation*)

Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan yang terjadi itu dipropagasi mundur. Dimulai dari garis yang berhubungan langsung dengan unit-unit di lapisan keluaran.

3) Fase 3 (Perubahan bobot)

Modifikasi bobot untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi.

2.1.9 Recognition Rate

Dalam metode klasifikasi akan dilakukan evaluasi terutama pada bagian akurasi dari hasil klasifikasi. Perhitungan akurasi pada sebuah klasifikasi berpengaruh terhadap performa dari metode dalam klasifikasi yang digunakan. Akurasi dalam klasifikasi adalah presentasi ketepatan *record* data yang diklasifikasikan secara benar setelah dilakukan pengujian pada hasil klasifikasi. Banyak cara untuk menghitung akurasi dari klasifikasi, diantaranya menggunakan *recognition rate*. Rumus perhitungan *recognition rate* sebagai berikut (Widagdho, 2014),

$$\text{Recognition Rate} = \frac{\Sigma \text{Jumlah data benar}}{\Sigma \text{Jumlah data seluruhnya}} \times 100\% \dots\dots\dots$$

Gambar 2.15 Rumus perhitungan *recognition rate* (Widagdho, 2014)

2.2 *State-of-the-Art*

Berdasarkan topik permasalahan yang telah dirumuskan maka dibuatlah *literature review* dari jurnal penelitian sebelumnya yang berkaitan dengan penerapan metode *JST backpropagation* dan deteksi tepi untuk pengenalan aksara Sunda. Penelitian mengenai aksara sunda telah banyak dilakukan sebelumnya, diantaranya adalah,

- a. Pada penelitian (Putra dan Ni, 2014) beberapa metode deteksi tepi yaitu Sobel, Robert dan Canny telah dibandingkan dengan hasil metode Canny lebih baik dalam melakukan pengolahan kualitas citra untuk penghitungan deteksi tepi karena output dari metode Canny memiliki batas dan tepi yang lebih jelas.
- b. Penelitian (Zaitun, dkk., 2015) untuk pengenalan pola citra tanda tangan menggunakan metode *JST backpropagation* menghasilkan akurasi sebesar 70%. Kekurangan penelitian ini tidak menggunakan deteksi tepi sehingga hasil pengukuran tingkat akurasi yang dihasilkan terbilang rendah.
- c. Penelitian (Hermana dan Meikel, 2015) tentang pengenalan pola rumah adat dengan menggunakan *JST backpropagation* dan deteksi tepi Canny menghasilkan akurasi sebesar 97.6% untuk citra latih dan 50.67% untuk citra

uji. Kekurangan penelitian ini tidak ada perbandingan antara metode deteksi tepi.

- d. Penelitian (Hara, dkk., 2016) mengenai pengenalan tulisan tangan aksara Lampung dengan kombinasi JST dan deteksi tepi Canny, pada penelitian ini bertujuan untuk membuat prototipe aplikasi untuk pengenalan tulisan tangan aksara Lampung dengan metode jaringan saraf tiruan dan deteksi tepi Canny dan hasil menghasilkan tingkat akurasi rata-rata sebesar 78%. Kekurangan penelitian ini tidak ada perbandingan antara metode deteksi tepi.
- e. Penelitian dari (Vermala, dkk., 2016) pengenalan huruf Hijaiyah menggunakan metode *Fuzzy Feature Extraction* dan *JST backpropagation* menghasilkan akurasi sebesar 98.33%. kekurangan data uji dan data latih yang digunakan sedikit dan tidak ada perbandingan metode deteksi tepi.
- f. Penelitian (Farhan, dkk., 2017) dengan judul perancangan dan analisis pengenalan kata aksara Sunda menggunakan *Learning Vector Quantization*, pada penelitian ini bertujuan untuk mengukur tingkat hasil pengenalan aksara Sunda dengan metode *learning vector quantization* dan menghasilkan akurasi rata-rata sebesar 73.33%. Untuk kekurangan penelitian ini adalah metode JST lebih baik dalam pengenalan pola dibandingkan metode LQZ.
- g. Penelitian dari (Damayanti dan Pujiatus, 2018) mengenai identifikasi huruf Hijaiyah menggunakan metode *backpropagation* memperoleh akurasi sebesar 51.33%. Kekurangan penelitian ini tidak menggunakan deteksi tepi sehingga hasil pengukuran tingkat akurasi yang dihasilkan terbilang rendah.

- h. Penelitian (Faturrahman, dkk., 2018) mengenai pengenalan huruf Hijaiyah dengan spesifik khat menggunakan metode JST *backpropagation* dan deteksi tepi Sobel menghasilkan akurasi terbaik sebesar 100% yang didapat pada learning rate 0.01 dan epoch 10000.
- i. Penelitian (Nisa Amalia, dkk., 2020) dengan judul pengenalan aksara Sunda menggunakan metode JST *backpropagation* dan deteksi tepi Canny, pada penelitian ini bertujuan untuk membuat aplikasi prototipe pengenalan pola aksara Sunda dengan metode JST *backpropagation* dan deteksi tepi Canny dengan menggunakan MATLAB serta mengukur hasil nilai akurasi yang dihasilkan. Untuk hasil penelitian aplikasi memiliki tingkat akurasi rata-rata 76.19% dan untuk kekurangan penelitian ini tidak ada perbandingan antara metode deteksi tepi.
- j. Penelitian (Andhika Rahma Putra, dkk., 2022) dengan judul pengenalan pola huruf Hijaiyah dengan penerapan metode JST dan deteksi tepi, pada penelitian ini bertujuan untuk membuat aplikasi prototipe pengenalan pola huruf Hijaiyah dengan metode JST *backpropagation* dan deteksi tepi dengan aplikasi MATLAB serta mengukur hasil nilai dan menghasilkan tingkat akurasi rata-rata sebesar 88,3%.

2.3 Matriks Penelitian

Berdasarkan SOTA yang telah dirangkum sebelumnya, dapat disimpulkan dalam bentuk matriks penelitian seperti yang ditunjukkan dalam tabel berikut.

Tabel 2.1 Matriks Penelitian

No	Peneliti	Ruang lingkup															
		Objek yang dideteksi						Program			Metode Ekstraksi Fitur				Metode Klasifikasi		
		Aksara Hijaiyah	Tulisan Tangan	Aksara Sunda	Batik	Rumah Adat	Aksara Lampung	MATLAB	Java-based	Delphi-based	Canny	Sobel	Roberts	Prewitt	JST Backpropagation	Learning Vector Quantization	Template Matching
1.	(Putra dan Ni, 2014)	✓						✓			✓	✓	✓		✓		
2.	(Zaitun dkk., 2015)		✓						✓						✓		

3.	(Hermana dan Meikel, 2015)					✓		✓			✓				✓		
4.	(Hara dkk., 2016)		✓				✓		✓		✓				✓		
5.	(Vermala dkk., 2016)	✓	✓						✓						✓		
6.	(Farhan dkk., 2017)			✓												✓	
7.	(Damayanti dkk., 2018)	✓						✓							✓		
8.	(Faturrahman dkk., 2018)	✓							✓			✓			✓		
9.	(Nisa Amalia dkk., 2020)			✓				✓			✓				✓		
10.	(Andhika Rahma Putra dkk., 2022)	✓						✓			✓	✓	✓		✓		
11.	Rizki Abdul Rahman	✓	✓	✓				✓			✓	✓	✓	✓	✓		

2.4 Penelitian Terkait

Pada tabel matriks penelitian adalah beberapa jurnal yang terkait dengan penelitian yang akan dilakukan dan diantara penelitian-penelitian tersebut ada 2 penelitian yang paling dekat dengan penelitian ini, diantaranya penelitian (Nisa Amalia dkk., 2020) dengan judul pengenalan aksara Sunda menggunakan metode JST *backpropagation* dan deteksi tepi Canny, pada penelitian ini bertujuan untuk membuat aplikasi prototipe pengenalan pola aksara sunda dengan metode JST *backpropagation* dan deteksi tepi Canny dengan menggunakan MATLAB serta mengukur hasil nilai akurasi yang dihasilkan.

Penelitian (Farhan dkk., 2017) dengan judul perancangan dan analisis pengenalan kata aksara Sunda menggunakan *learning vector quantization*, pada penelitian ini bertujuan untuk mengukur tingkat hasil pengenalan aksara Sunda dengan metode *learning vector quantization*.

Dari kedua penelitian sebelumnya ada beberapa perbedaan dalam metode pengenalan pengenalan pola dan metode deteksi tepi yang digunakan. Untuk metode yang digunakan pada penelitian ini menggunakan metode JST dan metode deteksi tepi Canny, Sobel dan Robert. Metode pengenalan pola JST memiliki tingkat akurasi lebih tinggi jika dibandingkan dengan metode LQZ dan digabungkan dengan 3 metode deteksi tepi sebagai pembanding untuk menghasilkan tingkat akurasi yang lebih tinggi.