

Segmentasi Baris Lontar Dengan Metode A * Path Planning

Anak Agung Istri Dewi Lestari^{a1}, I Gede Santi Astawa^{a2}, Ngurah Agus Sanjaya ER^{a3}, I Putu Gede Hendra Suputra^{a4}, Cokorda Rai Adi Pramatha^{a5}, I Wayan Supriana^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Badung, Bali, Indonesia

¹gungdewi59@gmail.com

²santi.astawa@unud.ac.id

³agus_sanjaya@unud.ac.id

⁴hendra.suputra@unud.ac.id

⁵cokorda@unud.ac.id

⁶wayan.supriana@unud.ac.id

Abstract

*Historical documents in the form of ancient manuscripts are one form of the Indonesian nation's cultural heritage that deserves to be important, one of which is Lontar. Currently not many people can read the writings in palm leaves, therefore, ancient manuscript collectors have made efforts to digitize ancient manuscripts. The digitization of ancient manuscripts aims to improve the image quality of ancient manuscripts with the help of computers. Digitization requires an image quality improvement process by performing noise reduction and edge detection and line segmentation on digital images of ancient manuscripts. In this study, the noise reduction process uses the Mean Filter method, edge detection uses the Sobel operator, and line segmentation in this study uses the A * Path Planning Algorithm. Based on research conducted on 24 lontar images, line segmentation process obtained an accuracy of 95%.*

Keywords: Image Processing, Lontar, Line Segmentation, A * Path Planning, Noise Reduction

1. Pendahuluan

Dokumen bersejarah berupa naskah kuno merupakan salah satu bentuk warisan budaya Bangsa Indonesia yang bernilai cukup penting, salah satunya yaitu Lontar. Seiring berjalannya waktu, penggunaan daun lontar semakin jarang karena telah tergantikan dengan penggunaan kertas. Daun lontar yang disimpan memiliki usia yang sudah tidak muda lagi sehingga lontar-lontar tua mulai mengalami kerusakan dan semakin sulit untuk dibaca secara manual karena lapisan epidermisnya yang terdiri dari sel-sel serat yang berupa kultikula (lapisan kuku) yang melindunginya sudah semakin terkikis dan akan menyebabkan naskah lontar lama-lama menjadi menipis. Sisa-sisa lepuhan dari lapisan epidermisnya akan kelihatan seperti debu halus yang melekat pada naskah lontar sehingga sulit untuk dibaca [1].

Saat ini tidak banyak orang yang dapat membaca tulisan dalam daun lontar, oleh karena itu, para kolektor naskah kuno telah melakukan upaya digitalisasi naskah kuno. Digitalisasi terhadap naskah kuno bertujuan untuk memperbaiki kualitas citra pada naskah kuno dengan bantuan komputer. Ketika naskah kuno telah diubah menjadi citra digital terdapat noise-noise pengganggu informasi, sehingga diperlukan perbaikan kualitas citra dengan melakukan reduksi noise pada citra digital naskah kuno. Pada saat proses reduksi noise tulisan-tulisan pada naskah kuno mengalami penipisan, sehingga setelah melakukan reduksi diperlukan proses penegasan karakter tulisan pada naskah kuno dengan deteksi tepi.

Setelah proses-proses tersebut terdapat beberapa pemrosesan citra lain, salah satunya adalah segmentasi. Segmentasi terdiri dari 2 tahap yaitu segmentasi baris dan juga segmentasi karakter. Segmentasi baris merupakan proses memisahkan baris satu dengan baris lainnya. Sedangkan segmentasi karakter merupakan proses memisahkan setiap karakter pada citra naskah kuno. Kedua proses segmentasi tersebut memiliki masalah-masalahnya tersendiri, misalnya, garis teks miring dan ketidakteraturan dalam garis seperti lebar baris, tinggi, dan jarak antar baris. Melihat karakteristik aksara Bali yang mengandung pengangge aksara berupa ulu dan gantungan, sering kali untuk

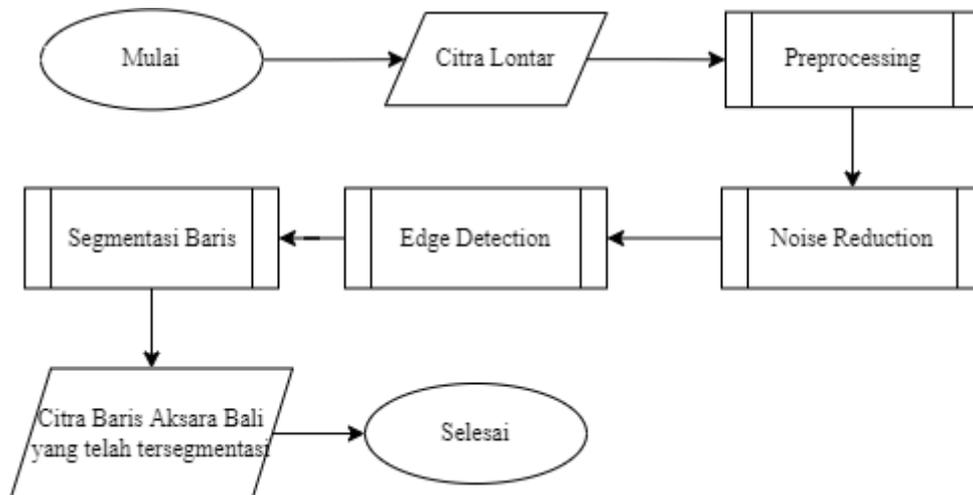
melakukan segmentasi baris pada tulisan di naskah kuno tersebut membutuhkan tantangan yang cocok untuk segmentasi baris.

Salah satu metode yang digunakan untuk proses segmentasi baris adalah metode A * Path Planning, adalah algoritma *path finding* untuk menghitung jalur terpendek dari titik awal ke tujuannya dengan menghindari rintangan sepanjang jalan. Penelitian mengenai penerapan metode A * Path Planning sudah pernah dilakukan sebelumnya, salah satunya adalah penelitian yang dilakukan oleh Pradnyawati [2]. pada penelitian ini dilakukan segmentasi karakter aksara Bali pada citra lontar, sebelum itu dilakukan segmentasi baris terlebih dahulu dengan akurasi sebesar 93%.

Pada penelitian yang dilakukan penulis ini, segmentasi baris dilakukan menggunakan metode A * Path Planning pada citra lontar Bali.

2. Metode Penelitian

Penelitian ini terbagi menjadi beberapa tahapan, diantaranya studi literatur, pengumpulan data, *preprocessing*, *noise reduction*, *edge detection*, segmentasi baris, dan testing. Pada tahap studi literatur, penulis melakukan pencarian, pengumpulan, serta pemahaman mengenai informasi dan literatur yang mendukung dalam penelitian ini. Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari arsip *The Balinese Digital Library*. Tahapan selanjutnya adalah *preprocessing*, dimana pada tahapan ini terdapat proses *grayscale* dan *binarization* untuk menghasilkan citra biner atau citra hitam putih. Kemudian dilanjutkan dengan denoising atau reduksi noise dengan metode *mean filter* dan deteksi tepi menggunakan operator *sobel* pada citra biner. Terakhir adalah tahapan segmentasi baris dengan A * Path Planning.



Gambar 1. Flowchart Metode Penelitian

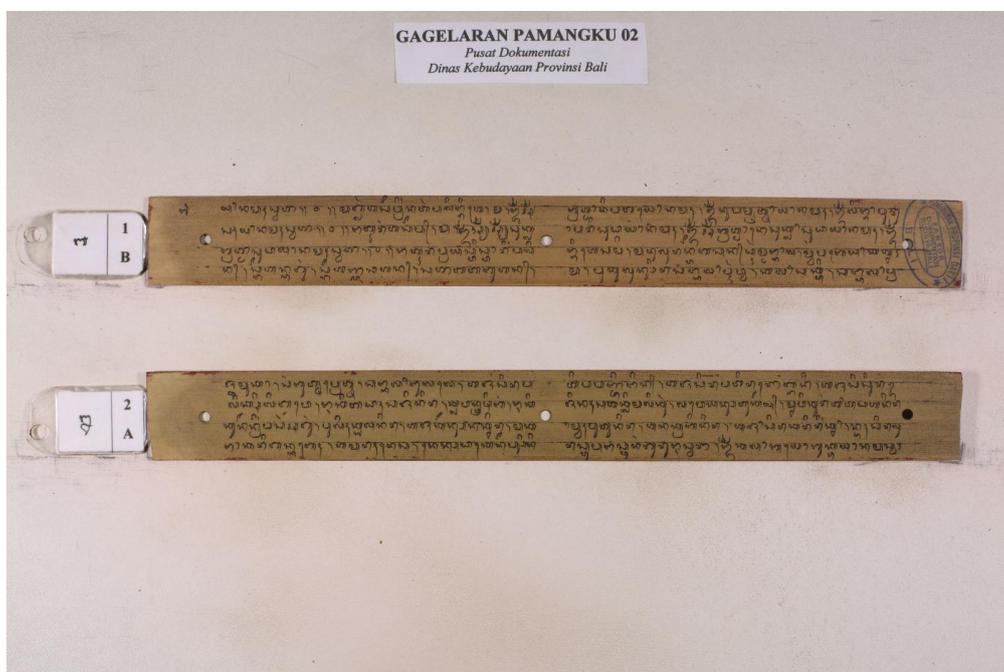
Gambar 1 merupakan desain penelitian segmentasi baris pada citra lontar Bali. Citra lontar sebagai input akan masuk pada proses *preprocessing*, dilanjutkan dengan denoising atau reduksi noise dengan metode *mean filter*, kemudian deteksi tepi menggunakan operator *sobel* pada citra biner, dilanjutkan dengan segmentasi baris menggunakan A * Path Planning sehingga menghasilkan citra baris aksara Bali yang telah tersegmentasi.

2.1. Studi Literatur

Pada tahap studi literatur, penulis melakukan pengumpulan dan pencarian, serta pemahaman mengenai informasi, literatur dan teori-teori yang mendukung terkait dengan penelitian mencakup tahapan dalam pengolahan citra digital dan algoritma yang digunakan untuk segmentasi baris yaitu metode A * Path Planning.

2.2. Pengumpulan Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder. Data yang dipilih merupakan citra lontar Bali yang hanya mengandung objek aksara Bali dengan kondisi sedikit noise dan aksara yang tertulis pada lontar dapat dilihat dengan mata telanjang. Jumlah data lontar yang digunakan pada penelitian kali ini yaitu 24 citra lontar yang diperoleh dari arsip *The Balinese Digital Library* pada *website* <https://archive.org/>. Gambar 2 merupakan contoh data lontar yang akan digunakan yang mana gambar akan melalui proses *cropping* terlebih dahulu.



Gambar 2. Contoh Data Lontar

2.3. Grayscale

Citra digital dengan nilai dari *Red* = *Green* = *Blue* merupakan citra *grayscale* yang hanya memiliki satu nilai kanal pada setiap pikselnya [3]. Pada citra *grayscale* ditampilkan citra yang terdiri atas warna hitam, keabuan, dan putih. Tingkat keabuan di sini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih.

Untuk mengubah citra berwarna yang memiliki nilai matrik masing-masing *Red*, *Green*, dan *Blue* menjadi citra *grayscale*, dapat dilakukan dengan menggunakan metode *weighted* [4]. Metode ini pada elemen *green* memberikan kontribusi yang lebih besar dan pada elemen *red* dan *blue* dilakukan pengurangan kontribusi pada setiap pikselnya dengan pemberian nilai pada masing-masing elemen warna adalah sebesar 30% *red*, 59% *green*, dan 11% *blue*. Berikut merupakan persamaan dari metode *weighted*:

$$\text{Grayscale} = R * 0.299 + G * 0.587 + B * 0.114 \quad (1)$$

Keterangan:

R = intensitas warna merah/*red*

G = intensitas warna hijau/*green*

B = intensitas warna biru/*blue*

2.4. Binarization

Binarization adalah proses mengubah citra berderajat keabuan (*grayscale*) menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk objek dan *background* dari citra secara jelas. Citra biner merupakan citra yang hanya memiliki dua kemungkinan nilai piksel, yaitu hitam dan putih [3]. Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Pada proses *thresholding*, akan ditetapkan suatu nilai yang disebut *threshold* yang merupakan batas nilai dalam menentukan warna pada citra apakah berada di atas nilai ambang atau di bawah.

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (2)$$

Keterangan :

$g(x,y)$: citra biner dari citra grayscale $f(x,y)$
T : nilai ambang

Ada berbagai macam metode *thresholding*, salah satunya adalah metode *Sauvola*. Metode *Sauvola* adalah algoritma *thresholding* yang termasuk ke dalam *local thresholding*. Adapun persamaan metode *Sauvola* adalah sebagai berikut [5]:

$$T_{sauvola} = m * (1 + k * (1 - \frac{s}{r})) \quad (3)$$

Keterangan:

m : mean dari piksel tetangga di bawah area jendela
r : rentang dinamis dari standar deviasi
k : konstanta dari 0-1
s : pixel yang akan diubah

2.5. Noise Reduction

Karakteristik derajat keabuan (*gray-level*) atau karakteristik Fungsi Probabilitas Kepadatan (*Probability Density Function* (PDF)) menyebabkan variable acak yang terjadi dapat menyebabkan noise pada sebuah citra. Untuk dapat mengekstrak fitur-fitur penting yang terkandung pada citra, maka perlu dilakukan reduksi noise. Salah satu teknik untuk mereduksi noise yaitu menggunakan *mean filter*. Menurut Hidayatullah, *Mean filter* adalah filter yang beroperasi dengan cara menginterasikan sebuah kernel dengan ukuran tertentu dari pojok kiri atas sampai pojok kanan bawah citra [6]. Setiap iterasi nilai piksel pada citra yang posisinya tepat di tengah-tengah kernel akan diganti dengan nilai rata-rata dari nilai piksel citra yang berada di dalam jendela tersebut. Luasan jumlah piksel tetangga ditentukan sebagai *masking/kernel/window* yang misalnya berukuran 2x2, 3x3, 4x4, dan seterusnya.

2.6 Edge Detection

Proses ini bertujuan untuk mendapatkan tepi dari area teks. Proses ini akan diterapkan dengan menggunakan operator pendeteksiian tepi yaitu operator *Sobel*. Operator sobel adalah operator yang menggunakan dua kernel 3x3 G_x dan G_y dan untuk magnitudo gradient (G) sebagai berikut:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad G = \sqrt{I * G_x^2 + I * G_y^2} \quad (4)$$

Penerapan operator *Sobel* dilakukan dengan mengkonvolusikan citra dengan kernel G_x dan G_y . Setelah dilakukan konvolusi dengan dua kernel yang berbeda, citra hasil konvolusi G_x dan G_y dikuadratkan kemudian dijumlahkan. Setelah dijumlahkan, hasil penjumlahan diambil akar kuadratnya. Akar kuadrat tersebut merupakan citra hasil penerapan operator *Sobel*.

2.7 A * Path Planning

Algoritma A * Path Planning menggunakan beberapa fungsi heuristik untuk mempercepat perhitungan solusi optimal untuk mencapai keadaan tujuan [7]. Tujuan dari algoritma ini adalah meminimalkan jumlah biaya pada path yang dilalui dari keadaan awal (s_1) menuju keadaan akhir (s_n). Misalkan terdapat serangkaian state $s_1^a, s_2^a, \dots, s_n^a$ yang dilalui oleh path p^a , maka untuk menghitung jalur optimal p^* dengan total biaya perjalanan terendah digunakan persamaan:

$$p^* = \arg \min_{p^a} \sum_{i=1}^{n^a-1} C(s_i^a, s_{i+1}^a) \quad (5)$$

Dimana $C(s_i, s_j)$ adalah *cost* untuk pergi dari posisi awal ke posisi akhir. Dalam menghitung $C(s_i, s_j)$ yaitu biaya perjalanan dari suatu keadaan sampai tercapainya keadaan tujuan digunakan lima fungsi biaya. Fungsi-fungsi biaya ini digabungkan untuk menghitung biaya perjalanan dari suatu keadaan sampai keadaan tujuan tercapai [7].

- a. Fungsi biaya jarak *foreground* $D(n)$ dan $D(n)^2$ mengontrol jalur agar tetap berada di celah antara piksel *foreground*. Fungsi $D(n)$ merupakan fungsi untuk melewati state n , dimana didefinisikan pada rumus di bawah:

$$D(n) = \frac{1}{1 + \min[d(n, n_{yu}), d(n, n_{yd})]} \quad (6)$$

Fungsi $D(n)^2$ merupakan atribut biaya yang jauh lebih tinggi untuk mendekati nilai *piksel* dibandingkan dengan menjau dari piksel hitam. $D(n)^2$ didefinisikan seperti di bawah :

$$D(n)^2 = \frac{1}{1 + \min[d(n, n_{yu}), d(n, n_{yd})]^2} \quad (7)$$

$d(n, n_{yu})$ dan $d(n, n_{yd})$ adalah jarak antara state n dan piksel *foreground* terdekat masing-masing dalam arah ke atas dan ke bawah [8].

- b. *Map-obstacle cost function* $M(n)$ yang memberi tanda apabila path harus melewati *piksel foreground*. $M(n)$ akan bernilai 1 jika state n bertepatan dengan *foreground piksel*. $M(n)$ akan bernilai 0 untuk kondisi sebaliknya.
- c. *Vertical cost function* $V(n)$ mencegah jalan menyimpang dari posisi y dari keadaan awal dan keadaan tujuan.

$$V(n) = \text{abs}(n_y - n_y^{\text{start}}) \quad (8)$$

n_y dan n_y^{start} merupakan posisi y pada *state* saat ini dan *state* awal.

- d. *Neighbor cost function* $N(s_i, s_j)$ yaitu untuk mengatur jalur terpendek antara keadaan awal dan tujuan. Ketika berpindah ke state baru, nilai $N(s_i, s_j)$ adalah 14 untuk arah diagonal, dan bernilai 10 untuk langkah vertikal dan horizontal.

Maka rumus untuk algoritma A* Path Planning yaitu menggunakan fungsi biaya gabungan $C(s_i, s_j)$:

$$C(s_i, s_j) = c_d D(s_i) + c_{d2} D(s_i)^2 + c_m M(s_i) + c_v V(s_i) + c_n N(s_i, s_j) \quad (9)$$

$c_d, c_{d2}, c_m, c_v, c_n$ disesuaikan dengan menggunakan beberapa gambar selama percobaan pendahulu.

2.8 Pengujian

Pada tahap ini akan dilakukan pengujian dengan menggunakan *confusion matrix* atau disebut juga matriks klasifikasi adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep data mining [9]. Format dari *confusion matrix* dapat dilihat pada Gambar 3.

		Observed	
		TRUE	FALSE
Predicted	TRUE	TRUE POSITIVE (TP)	FALSE POSITIVE (FP)
	FALSE	FALSE NEGATIVE (FN)	TRUE POSITIVE (TP)

Gambar 3. Format *Confusion Matrix*

Definisi dari ketentuan yang digunakan yaitu :

- True Positive* (TP) : Pengamatan positif, tetapi diperkirakan negatif.
- True Negative* (TN) : Pengamatan negatif, dan diperkirakan negatif.
- False Positive* (FP) : Pengamatan negatif, tetapi diperkirakan positif.
- False Negative* (FN) : Pengamatan positif, tetapi diperkirakan negatif.

Hasil klasifikasi mempunyai kemungkinan yang benar dalam kelasnya (TP dan TN) atau salah yang akan masuk ke kelas lainnya (FP dan FN). Hasil pengukuran dapat diperoleh dengan melihat pada format confusion matriks di atas pada Gambar 3. Akurasi dari pengklasifikasian yang merupakan jumlah record data yang diklasifikasikan (prediksi) secara benar oleh algoritma diperoleh dari formula:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

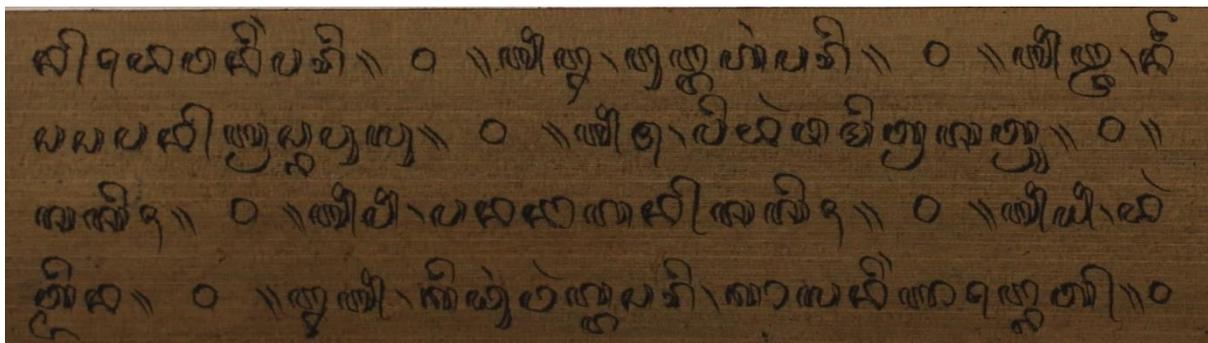
Presentase jumlah record data yang diklasifikasikan prediksi secara salah oleh algoritma disebut *Misclassification (Error) Rate*, yang dihitung dengan rumus:

$$Misclassification\ Rate = \frac{FP+FN}{TP+TN+FP+FN} \quad (11)$$

3 Hasil dan Pembahasan

3.1 Implementasi Metode

Penelitian mengenai segmentasi baris ini diimplementasikan menggunakan bahasa pemrograman *Python*. Data yang digunakan dalam penelitian ini adalah 24 citra lontar Bali dengan format *JPG*. Pada tahap pertama akan dilakukan proses *cropping* citra secara manual. *Cropping* dilakukan agar seluruh data citra yang digunakan sebagai citra masukan memiliki ukuran piksel yang digunakan sama. Gambar 4 menunjukkan citra lontar yang sudah melalui proses *cropping*.



Gambar 4. Lontar yang sudah melalui proses *cropping*

3	Lontar12	Baris 1	41	0	1	0.024	0.976	0.969
		Baris 2	42	0	0	0.000	1.000	
		Baris 3	41	2	0	0.047	0.953	
		Baris 4	37	0	2	0.054	0.946	

Berdasarkan pada pengujian 24 buah citra lontar yang telah melalui proses *cropping* terlebih dahulu maka diperoleh hasil akurasi segmentasi baris sebesar 95%. Masing-masing baris aksara yang tersegmentasi memiliki tingkat akurasi yang beragam, namun tidak ada satupun hasil pengujian yang menghasilkan akurasi dibawah 90%, dimana akurasi citra memberikan hasil pengujian antara 90% - 98%.

4 Kesimpulan

Dari penelitian yang dilakukan dapat disimpulkan bahwa metode *A * Path Planning* dapat digunakan untuk segmentasi baris pada citra lontar Bali. Berdasarkan hasil pengujian menggunakan *confusion matrix* diperoleh tingkat akurasi sebesar 95%. Hal ini menunjukkan bahwa metode *A * Path Planning* pada proses segmentasi baris sudah berjalan dengan baik. Dalam penelitian ini, terdapat beberapa hal yang bisa dilakukan di penelitian selanjutnya yaitu pemisahan latar belakang dan tulisan pada lontar ketika latar belakang tidak seperti latar belakang lontar pada umumnya. Selain itu, perlu adanya penambahan proses agar citra yang dihasilkan dapat terbaca dengan jelas dan sedikit *noise*.

Referensi

- [1] I. W. Suardiana, Teknik Konservasi Lontar, Denpasar: Dinas Kebudayaan Provinsi Bali, 2021.
- [2] A. Wedianto, H. L. Sari, & H. Y. Suzantri, "Analisa Perbandingan Metode Filter Gaussian, Mean Dan Median Terhadap Reduksi Noise" *Jurnal Media Infotama*, vol. 12, no. 1, p. 21–30. 2016. <https://doi.org/https://doi.org/10.37676/jmi.v12i1.269>
- [3] D. Salomon, "The Computer Graphics Manual", 2011. <https://doi.org/10.1007/978-0-85729-886-7>
- [4] D. Putra, Pengolahan citra digital, Penerbit Andi, 2010
- [5] M. P. J. Sauvola*, & K. Inen, "Adaptive degraded document image binarization". *Pattern Recognition*, vol. 39, no. 3, p. 317–327, 2006. <https://doi.org/10.1016/j.patcog.2005.09.010>
- [6] P. Hidayatullah, Pengolahan Citra Digital: Teori dan Aplikasi Nyata, Informatika, 2017
- [7] O. Surinta, M. Holtkamp, F. Karabaa, J. P. Van Oosten, L. Schomaker, & M. Wiering, "A Path Planning for Line Segmentation of Handwritten Documents". *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR, 2014-December*, p. 175–180, 2014. <https://doi.org/10.1109/ICFHR.2014.37>
- [8] Kesiman, M. W. A., Vally, D., Burie, J.-C., Paulus, E., Sunarya, I. M. G., Hadi, S., ... Ogier, J.-M. (2016). Southeast Asian palm leaf manuscript images: a review of handwritten text line segmentation methods and new challenges. *Journal of Electronic Imaging*, 26(1), 011011. <https://doi.org/10.1117/1.jei.26.1.011011>
- [9] Rahman, M. F., Alamsah, D., Darmawidjadja, M. I., & Nurma, I. (2017). *Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN)*. *Jurnal Informatika*, 11(1), 36. <https://doi.org/10.26555/jifo.v11i1.a5452>

This page is intentionally left blank.