

TRANSLITERASI HURUF ARAB PEGON - LATIN MENGUNAKAN EIGENSPACE

Ahmat Nurwakit.*, Ema Utami*, Hanif Al Fatta*

* Magister Teknik Informatika, Universitas Amikom Yogyakarta

Correspondence Author: ahmat.nurwakit@gmail.com

Info Artikel :	ABSTRACT
<p>Sejarah Artikel : Menerima : 22 Februari 2020 Revisi : 27 Maret 2020 Diterima : 14 Juni 2020 Online : 26 Agustus 2020</p> <p>Keyword : <i>Transliteration, Arabic Font, Arab letters Pegon, Eigenspace, Eigenvalue</i></p>	<p><i>Pegon Arabic Transliteration - Latin Using Eigenspace is the translation of the Arabic letters Pegon into Latin letters using Eigenspace. This research focuses on several limitations, such as the raw dataset for pegon letters derived from handwritten letters C, G, P, each scanned pegon letter will be stored in an image document measuring 100 x 100 pixels, the eigenvalues of each image will be stored in a text file document. , test data using camera-captured images or handwritten scans. The purpose of this research is to collect a dataset for pegon letters that comes from direct handwriting, compile eigen values and eigen vectors from the collected dataset, classify pegon letters using eigenspace and measure the accuracy of using eigenspace in recognizing Arabic letters pegon. Handwritten object recognition steps, which are the image file is taken from the web cam, resize the image to 100x100 pixels, the resized image will be converted into a matrix, change the matrix into eigenvalues. After testing 10 times each of the letters C, G, P, it shows success on the characters that are fully recorded with the matrix position in accordance with the established pattern.</i></p>
	INTISARI
<p>Kata Kunci : <i>Tranliterasi, Huruf Arab, Huruf Arab Pegon, Eigenspace, Eigenvalue</i></p>	<p>Transliterasi Huruf Arab Pegon – Latin Menggunakan <i>Eigenspace</i> adalah penerjemahan huruf arab pegon kedalam huruf latin menggunakan <i>Eigenspace</i>. Penelitian ini terfokus pada beberapa batasan – batasan seperti Dataset mentah huruf pegon berasal dari tulisan tangan huruf C, G, P, tiap huruf pegon yang sudah terpindai akan disimpan dalam dokumen citra berukuran 100 x 100 pixel, <i>eigen values</i> setiap citra akan tersimpan dalam dokumen file teks, data uji menggunakan citra hasil tangkapan kamera atau pindaian dari tulisan tangan. Tujuan penelitian ini adalah mengumpulkan dataset untuk huruf pegon yang berasal dari tulisan tangan langsung, menyusun <i>eigen values</i> dan <i>eigen vectors</i> dari dataset yang sudah dikumpulkan, melakukan klasifikasi huruf pegon menggunakan <i>eigenspace</i> dan mengukur berapa tingkat akurasi penggunaan <i>eigenspace</i> dalam pengenalan huruf arab pegon. Langkah-langkah pengenalan objek tulisan tangan, yang pertama, file image diambil dari hasil pindai web cam, kedua, Mengubah ukuran image menjadi 100x100 pixel, ketiga, dari image yang sudah diresize dirubah menjadi matriks, keempat, mengubah matriks ke dalam eigen value, kelima, proses pencocokan nilai terdekat antar dataset. Setelah melakukan pengujian sebanyak masing-masing 10 kali uji pada huruf C, G, P menunjukkan keberhasilan pada karakter yang terekam secara penuh dengan posisi matriks yang sesuai dengan pola yang ditetapkan.</p>

1. PENDAHULUAN

Computer vision salah satu bidang dari teknologi *Artificial Intelligence* adalah bidang *interdisiplin* yang mempelajari tentang bagaimana komputer dapat melakukan pemahaman terhadap citra digital dan video. *Computer vision* berfokus pada sistem cerdas yang dapat melakukan ekstraksi data dari citra digital ke dalam bentuk numerik. Sub domain dari *computer vision* meliputi *scene reconstruction*, *event detection*, *video tracking*, *object recognition*, *3D pose estimation*, *learning*, *indexing*, *motion estimation*, dan *image restoration*. *Handwriting character recognition* (pengenalan tulisan tangan) merupakan cabang dari *object recognition*, yaitu kemampuan komputer untuk menterjemahkan, menerima dan menafsirkan input tulisan tangan yang dapat dimengerti oleh komputer seperti foto, dokumen kertas, layar sentuh dan perangkat lainnya. Salah satu aksara yang dijadikan objek dalam pengenalan tulisan tangan adalah huruf arab pegon. Huruf pegon biasa digunakan untuk terjemah kitab kuning dalam versi bahasa daerah yang umumnya berlokasi di Jawa, Melayu, Sunda. Terjemah kitab-kitab kuning klasik di Indonesia banyak menggunakan 3 bahasa tersebut, sehingga orang-orang yang tidak menguasai dan memahami tulisan dari bahasa tersebut akan mendapatkan kesulitan untuk mendapatkan terjemah.

Berbeda dengan huruf arab baku, huruf pegon memiliki beberapa karakter yang merupakan rekayasa agar dapat dibaca menyesuaikan lidah bahasa daerah bersangkutan. Masalah yang muncul adalah huruf pegon tidak dapat dibaca oleh seseorang yang tidak memiliki kosa kata dalam bahasa bersangkutan sehingga membutuhkan proses penerjemahan. Proses penerjemahan sendiri hanya mungkin dilakukan jika kalimat tertulis dalam huruf latin. Penelitian ini memberi batasan masalah pengenalan huruf pegon C, G,P yang membedakan dengan kharakter tulisan arab standar.

Eigenspace adalah *subspace* dari *aljabar linier* yang terdiri dari sekumpulan *eigen vector*. Tiap *eigen vector* terbentuk dari banyak *eigen value*. Salah satu pemanfaatannya adalah *eigenface* yang digunakan dalam pengenalan wajah. Caranya adalah dengan mengubah citra digital ke dalam *eigen value* yang kemudian disusun menjadi *eigen vector*. Data uji kemudian akan dihitung jaraknya terhadap semua *vector* yang ada kemudian diambil yang nilainya paling dekat. Dari beberapa paparan di atas dapat diambil sebuah penelitian dengan judul **Transliterasi Huruf Arab Pegon Latin Menggunakan Eigenspace.**

2. METODOLOGI PENELITIAN

2.1 Metode Pengumpulan Data

Penelitian ini menggunakan tahap eksperimen sebagai metode pengumpulan data, namun selain itu peneliti juga mengumpulkan data awal sebagai bahan referensi melalui observasi, wawancara, dan studi pustaka kepada praktisi langsung yang terkait langsung dengan bidangnya. Tahap eksperimen dilakukan dengan melakukan scan huruf arab pegon kemudian diambil nilai *eigen* dan dikumpulkan kemudian diuji coba dan disimpulkan untuk mengukur tingkat akurasi. Proses pengambilan data pada objek yang akan diteliti, dalam kasus ini peneliti mengambil data tulisan tangan langsung huruf pegon yang akan digunakan sebagai dataset. Dataset yang diperoleh tersebut akan dijadikan model data *training* untuk melakukan identifikasi dengan menarik.

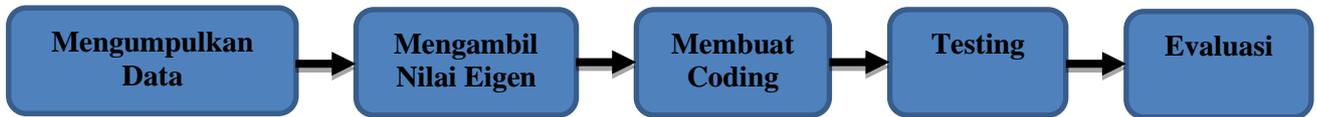
2.2 Metode Analisis Data

Metode analisis data pada penelitian ini adalah mengidentifikasi hasil perhitungan dengan melakukan validasi dengan ahlinya mulai dari awal sampai akhir. Identifikasi dimulai dari melakukan pengambilan foto atau scan tulisan tangan huruf pegon selanjutnya akan dilakukan transformasi data, dimana pada tahap ini akan dilakukan penghitungan untuk dijadikan menjadi data set berdasarkan label jenis huruf pegon. Setelah data berhasil dikelompokkan maka proses selanjutnya adalah melakukan proses pengambilan nilai *eigen* dan disimpan menjadi file teks.

Penarikan kesimpulan dilakukan berdasarkan hasil identifikasi berdasarkan dengan nilai *eigen*. Hasil identifikasi tersebut akan dijadikan acuan atau pedoman dalam menentukan hasil pada penelitian ini.

2.3 Alur Penelitian

Alur penelitian pada sistem Pengenalan tulisan tangan huruf arab pegon dimulai dengan mengumpulkan dataset berupa tulisan tangan huruf arap pegon C,G,P dengan cara dipindai dengan ukuran 100x100 pixels lalu disimpan dalam format file JPG. Langkah selanjutnya, mengambil nilai *eigen* dan membuat coding serta melakukan pengujian serta diakhiri dengan evaluasi. Adapun alur penelitian ditunjukkan sebagai berikut:



Gambar 1. Alur Penelitian

2.4 Pengumpulan dataset

Pada penelitian ini data yang digunakan untuk penelitian adalah data hasil pindai dari tulisan tangan huruf arab pegon dengan ukuran 100x100 pixel. Untuk penelitian saat ini diambil data sejumlah 9 data dengan 3 huruf arab pegon yaitu C, G, dan P.

Tabel 1. Foto Huruf Arab Pegon 1

		
Foto Huruf C1	Foto Huruf G1	Foto Huruf P1

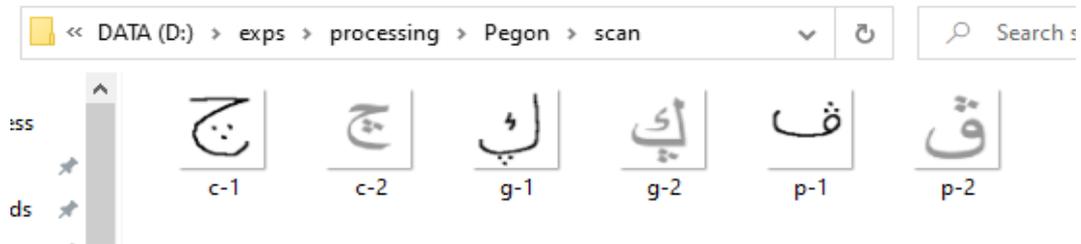
Tabel 2. Foto Huruf Arab Pegon 2

		
Foto Huruf C2	Foto Huruf G2	Foto Huruf P2

Tabel 3. Foto Huruf Arab Pegon 3

		
Foto Huruf C3	Foto Huruf G3	Foto Huruf P3

Huruf – huruf seperti yang telah diperlihatkan dalam Tabel 1, 2, dan 3 dipindai dan dilakukan proses *grayscale*, *cropping* dan *resizing* menggunakan perangkat lunak pengolahan *image*. Masing – masing huruf tersimpan dalam sebuah *file* bertipe .PNG. Gambar 4. berikut ini menunjukkan beberapa contoh file hasil pemindaian.



Gambar 4. File Hasil Pemindaian

Proses berikutnya adalah mengambil matriks dari tiap – tiap file huruf. Matriks merupakan representasi tiap *pixel* dalam file *image*. Proses pengambilan matriks menggunakan fungsi seperti ditunjukkan kode di bawah ini.

```
double[][] createMatrix(PImage src) {
    double[][] m = new double[100][100];

    for(int y = 0; y < 100; y++) {
        for(int x = 0; x < 100; x++) {
            color c = src.get(x, y);
            m[x][y] = brightness(c) < 50 ? 1 : 0;
        }
    }

    return m;
}
```

Fungsi ini akan membaca *pixel* demi *pixel* dan memeriksa nilai kecerahan (*brightness* atau *greyscale*). Bilai nilai kecerahan adalah 50 (dari skala 0 – 255) maka *pixel* akan diberi nilai 1, begitu sebaliknya adalah nilai 0. Nilai 0 dan 1 inilah representasi sebuah *image* dalam sebuah matriks.

Proses yang sama berlaku juga untuk *file* dataset yang lain. Selain untuk dataset, proses tersebut juga berlaku untuk data uji yang dimbil dari pemindaian *web cam* secara langsung. Perbedaannya adalah matriks untuk data uji tidak disimpan dalam *file* teks.

3.8. Pengkodean Program

Pengkodean program adalah proses pembuatan program untuk melakukan testing. Kode program yang dibuat untuk testing terpisah dari kode untuk pengambilan dataset walaupun ada beberapa fungsi yang identik. Perbedaannya adalah di dalam program untuk testing ini ada fungsi – fungsi yang berhubungan dengan *capture image* dari *web cam*. Saat melakukan *capture* program harus bisa melakukan *cropping*, *resizing*, dan *grayscale* secara langsung.

Saat pertama kali dijalankan, program testing akan memuat dataset yang sudah tersimpan di *file* teks. Berikut ini adalah kode fungsi untuk memuat dataset dari file teks.

```

double[][] loadEigVect(String src) {
    double[][] m = new double[100][100];

    String[] lines = loadStrings(src);
    for(int i = 0; i < 100; i++) {
        String[] splt = split(lines[i], ' ');

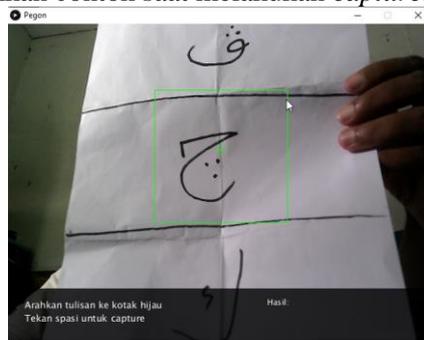
        for(int j = 0; j < 100; j++) {
            m[j][i] = Double.parseDouble(splt[j]);
        }
    }

    return m;
}

```

Eigen vector yang sudah dimuat tersebut untuk selanjutnya akan dibandingkan dengan *eigen vector* hasil *capture web cam* sebagai data tes.

Hasil *capture* dari *webcam* berukuran 640x480 *pixel*. Untuk mempermudah proses *capturing* di layar sudah dibuat *box* sebagai panduan penempatan huruf yang akan di-*capture*. Gambar 5 berikut ini menunjukkan contoh saat melakukan *capture*.



Gambar 5. Proses *Capturing*

Hasil *capture* akan di-*crop* sesuai ukuran *box*. Pemanggilan fungsi **cropImage()** dilakukan serentak dengan *resize* dan dilanjutkan dengan *generate eigen vector*. Proses *grayscale* dijadikan satu dengan proses saat *generate* matriks dan *eigen vector*. Proses – proses ini dibungkus dalam sebuah *class* yang bernama *EigenHelp* seperti ditunjukkan pada baris terakhir potongan kode di atas. Saat nilai *eigen vector* dari hasil *capture* sudah didapat, maka langkah berikutnya adalah melakukan kalkulasi bersama dengan dataset.

4. Hasil dan Pembahasan

Pengujian pada penelitian ini dilakukan dengan cara melakukan pemindaian sampel huruf C, G dan P yang sudah di tulis di kertas. Skenario yang diberlakukan untuk masing-masing huruf terdiri dari 10 jenis, yaitu dimulai dari menampilkan huruf secara utuh, perbesaran, pengecilan, dan rotasi. Hasil dari pengujian untuk huruf C ditampilkan pada Tabel 4, huruf G pada Tabel 5, dan huruf P pada Tabel 6.

Tabel 4. Pengujian Pegon untuk Huruf C

Pengujian	Keterangan	Cocok (Y/T)
C 1. 	-Huruf pada obyek masuk kotak pindai mendekati presisi.	Y
C 2. 	-Huruf pada obyek masuk kotak pindai -Kotak pada obyek tidak masuk kotak pindai.	Y
C 3. 	-Huruf pada obyek masuk kotak pindai -Kotak masuk bagian kiri pada kotak pindai.	Y
C 4. 	-Huruf masuk pada kotak pindai. -Kotak obyek masuk bagian atas pada kotak pindai.	Y
C 5. 	-Huruf masuk pada kotak pindai miring ke kanan -Kotak obyek tidak beraturan	Y
C 6. 	-Huruf masuk kotak pindai. -Kotak obyek posisi miring kanan tidak beraturan.	Y
C 7. 	-Huruf masuk kotak pindai. -Kotak obyek miring ke kiri	Y
C 8. 	-Huruf pada obyek hanya masuk sebagian pada kotak pindai. -Kotak obyek masuk sebagian pada kotak pindai	T
C 9. 	-Huruf obyek miring pada kotak pindai. -Kotak obyek miring pada kotak pindai	T
C 10. 	-Huruf pada obyek terbalik pada kotak pindai -Kotak juga terbalik pada kotak pindai.	T

Tabel 5. Pengujian Pegon untuk Huruf G

Pengujian	Keterangan	Cocok (Y/T)
G 1. 	-Huruf pada obyek masuk kotak pindai mendekati presisi.	Y
G 2. 	-Huruf pada obyek masuk kotak pindai -Kotak pada obyek tidak masuk kotak pindai.	Y
G 3. 	-Huruf pada obyek masuk kotak pindai -Kotak masuk bagian kanan pada kotak pindai.	Y
G 4. 	-Huruf masuk pada kotak pindai. -Kotak obyek masuk miring pada kotak pindai.	Y
G 5. 	-Huruf masuk pada kotak pindai miring ke kanan -Kotak obyek tidak beraturan	Y
G 6. 	-Huruf masuk kotak pindai. -Kotak obyek posisi miring kanan tidak beraturan.	Y
G 7. 	-Huruf masuk kotak pindai. -Kotak obyek miring ke kiri	Y
G 8. 	-Huruf pada obyek hanya masuk sebagian pada kotak pindai. -Kotak obyek masuk sebagian pada kotak pindai	T
G 9. 	-Huruf obyek masuk sebagian bawah pada kotak pindai. -Kotak obyek masuk sebagian bawah pada kotak pindai	T
G 10. 	-Huruf pada obyek masuk sebagian atas pada kotak pindai -Kotak juga masuk sebagian atas pada kotak pindai.	T

Tabel 6. Pengujian Pegon untuk Huruf G

Pengujian	Keterangan	Cocok (Y/T)
P 1. 	-Huruf pada obyek masuk kotak pindai mendekati presisi.	Y
P 2. 	-Huruf pada obyek masuk bagian kiri kotak pindai -Kotak pada obyek tidak masuk kotak pindai.	Y
P 3. 	-Huruf pada obyek masuk kotak pindai miring ke kanan -Kotak masuk miring bagian kanan pada kotak pindai.	Y
P 4. 	-Huruf masuk pada kotak pindai bagian kiri. -Kotak obyek masuk sebagian pada kotak	Y
P 5. 	-Huruf masuk pada kotak pindai miring ke kiri -Kotak obyek tidak beraturan	Y
P 6. 	-Huruf masuk kotak pindai miring ke kanan. -Kotak obyek posisi miring kanan tidak beraturan.	Y
P 7. 	-Huruf masuk kotak pindai. -Kotak obyek miring ke kiri	Y
P 8. 	-Huruf pada obyek hanya masuk sebagian bawah pada kotak pindai. -Kotak obyek masuk sebagian bawah pada kotak pindai	T
P 9. 	-Huruf obyek masuk sebagian atas pada kotak pindai. -Kotak obyek miring pada kotak pindai	T
P 10. 	-Huruf pada obyek sebagian kiri pada kotak pindai -Kotak juga masuk pojok bawah kanan pada kotak pindai.	T

4.1. Analisa Hasil

Tabel 4,5, dan 6 menunjukkan pengujian untuk huruf C, G, dan P. Masing - masing huruf diuji 10 kali dengan posisi yang berbeda - beda. Hasil dari proses pengujian menunjukkan pola sama, dimana tujuh pengujian pertama semuanya berhasil mengenali huruf, sedangkan tiga pengujian terakhir tidak berhasil mengenali huruf.

Sebagaimana telah dijelaskan sebelumnya bahwa program pengujian hanya memanfaatkan kotak sebagai panduan untuk meletakkan objek, sehingga program hanya akan melakukan pemrosesan objek yang ada di dalam kotak saja. Sehingga jika ada huruf yang tidak dipindai secara lengkap maka sistem tetap tidak bisa mengenali huruf tersebut. Hal ini dapat dilihat pada table hasil pengujian pada proses C8, C9, C10, G8, G9, G10, P8, P9, P10.

Hasil pengujian yang tidak berhasil lainnya adalah X, Y, Z. Berbeda dengan sebelumnya, dimana huruf hanya sebagian saja yang dipindai, untuk kasus huruf x,y,z huruf berhasil dipindai seluruhnya. Namun posisi huruf yang terotasi 90 dan 180 derajat membuat sistem tidak mampu mengenali huruf tersebut. Hal ini bisa dianalisa menggunakan pendekatan matriks yang membentuk *eigen factor*. Hasil pada huruf yang berotasi menunjukkan bahwa huruf tersebut telah membentuk pola matriks baru yang tidak sesuai dengan pola yang telah ditetapkan di sistem. Dengan demikian meskipun sistem bisa mengenali keseluruhan huruf, tetapi rotasi pada huruf menyebabkan pola matriks yang sama sekali baru dan tidak sesuai dengan rumus pada sistem. Sehingga pengujian huruf arab pegon C, G, P menunjukkan keberhasilan pada karakter yang terekam secara penuh dengan posisi matriks yang sesuai dengan pola yang ditetapkan.

Dari 10 kali pengujian untuk masing-masing huruf C, G, P tujuh diantaranya berhasil mengenali huruf dengan benar. Sedangkan tiga lainnya yang tidak dikenali dikarenakan *bug* dalam library Java Matrix Package.

5. KESIMPULAN

Kesimpulan yang dihasilkan pada uji coba dan evaluasi diatas diantaranya adalah :

1. Pengumpulan eigen value dan eigen vector dataset dengan cara memindai image, grayscale, cropping, dan resizing ke ukuran 100x100 pixel menggunakan perangkat lunak Paint. Kemudian dilanjutkan dengan pengambilan nilai grayscale dari tiap pixel menjadi matriks dan menyimpannya ke file teks.
2. Klasifikasi huruf data uji menggunakan program pengujian. Data uji akan dipindai menggunakan webcam dan dilakukan pengambilan eigen value dan eigen vector untuk kemudian dikalkulasi dengan dataset. Proses kalkulasi memanfaatkan library Java Matrix Package.
3. Dari 10 kali pengujian untuk masing-masing huruf C, G, P tujuh diantaranya berhasil mengenali huruf dengan benar. Sedangkan tiga lainnya yang tidak dikenali dikarenakan saat proses pemindaian posisi objek tidak masuk sempurna kedalam kotak pindai atau posisi objek terbalik.

DAFTAR PUSTAKA

- AlKhateeb, H., Jawad, 2015, *A Database for Arabic Handwritten Character Recognition*, Department Of Computer Science, College of Computer Science and Engineering, Taibah University, KSA
- El-Sawy, Ahmed, Loey, Mohamed and El-Bakry, Hazem, 2017, *Arabic Handwritten Characters Recognition using Convolutional Neural Network*, Benha University, Faculty of Computer & Informatics Computer Science Department Egypt
- Choudhary, Amit, Rishi, Rahul and Ahlawat, Savita, 2013, *A New Character Segmentation Approach for Off-Line Cursive Handwritten Words*, New Delhi, India
- Supriana, Iping, dan Nasution, Aldbr, 2013, *Arabic Character Recognition System Development*, Informatics Engineering, Institut Teknologi Bandung
- Mouhcinea, Rabi, Mustaphaa, Amrouch and Zouhirba, Mahani, 2017, *Recognition of cursive Arabic handwritten text using embedded training based on HMMs*, Morocco
- Boukharouba, Abdelhak, 2016, *A New Algorithm for Skew Correction and Baseline Detection Based on The Randomized Hough Transform*, Guelma, Algeria
- Sahu, Narendra dan Sonkusare, Manoj, 2017, *Stydy On Optical Character Recognition Techniques*, The International Journal of Computational Science, Information Technology and Control Engineering (IJCSITCE) Vol.4, No.1, Januari, 2017
- Lata, Y. Vijaya, Tungathurthy, B.K., Chandra, Rao, Mohan, R., Govardhan, and A., Reddy, L.P., 2009, *Facial Recognition Using Eigenfaces by PCA*, International Journal of Recent Trends in Engineering, Vol.1, No.1, May, 2009
- Turk, A., Matthew, Pentland, P., Alex, 1991, *Face Recognition Using Eigenfaces*, Vision and Modeling Group, The Media Laboratory Massachusetts Institute of technology, IEEE, 1991
- Solomon, Chris, and Breckon, Toby, 2011, *Fundamentals of Digital Image Processing*, UK : Willey-Blackwell, 2011
- Phangtriasi, R., Michael, 2017, *Optical Character Recognition (OCR)*.