

PERBANDINGAN ALGORITMA SUM OF SQUARED DIFFERENCE (SSD) DAN OPTIMISED SUM OF ABSOLUTE DIFFERENCE (OSAD) UNTUK PENGENALAN SIMBOL PADA CITRA EKSPRESI MATEMATIKA TERCETAK

Aditya Wikan Mahastama¹
mahas@staff.ukdw.ac.id

Abstract

The process of recognising printed mathematical expression consists of two general processes: symbol recognition and structural analysis which reads the expression structure and the relation between symbols within the structure, provide adequate labeling to preserve it, and finally be useful to reconstruct the whole expression.

This research focused on comparing the effectivity of two correlation-based pattern matching algorithms: the well-established Sum of Squared Difference (SSD) and the relatively new Optimized Sum of Absolute Difference (OSAD) which is based on average face rather than matching all trained symbols. Which one was better will be used on the next stage of a research on converting printed mathematical expression into LaTeX notation..

Results obtained from tests conducted shown that (1) the accuracy of OSAD is 85.6% and SSD is 70,6% on various test symbols, and (2) the usage of average face in OSAD which was proposed to reduce time needed for matching, proven to have no negative effects on its matching performance. Therefore OSAD is well ahead in processing time and accuracy than SSD.

Keywords— *Pattern recognition, symbol recognition, OSAD, SSD*

1. PENDAHULUAN

Penelitian mengenai pengenalan ekspresi matematika terus mengalami perkembangan, seiring berkembangnya peralatan interaksi dengan manusia dan bertambahnya minat dalam menyalin dokumen ilmiah ke dalam bentuk elektronik. Pengenalan ekspresi matematika tercetak merupakan bagian penting pada analisis citra dokumen ilmiah (Chan & Yeung, 2000). Pengenalan ekspresi matematika tercetak dapat dibagi ke dalam dua langkah utama: pengenalan simbol dan analisis struktural. Pengenalan simbol bekerja untuk melakukan segmentasi citra dan mengenali karakter matematis dengan benar. Analisis struktural bertujuan untuk menentukan hubungan antar simbol yang ditemukan, untuk merekonstruksi kembali ekspresi matematika secara lengkap. Kedua masalah ini berkaitan erat karena kesalahan pengenalan pada tahap pengenalan simbol dapat menyebabkan kekeliruan di tahap analisis. Sebaliknya, informasi struktural yang didapatkan dapat digunakan untuk membantu langkah pengenalan simbol (Alvaro, Sanchez, & Benedi, 2011).

Untuk pengenalan pola, pendekatan yang digunakan bervariasi misalnya klasifikasi seperti penggunaan klasifikasi k-means (Alvaro, Sanchez, & Benedi, 2011), klasifikasi k-means dan VSM (Ranger, Wang, & Cheng, 2012), dan klasifikasi *Bayesian Network* (Bluche, 2010). Pendekatan lain adalah menggunakan mesin *Optical Character Recognition* (OCR) (Prusa & Hlavac, 2007) (Eto & Suzuki, 2001). Terdapat pula pendekatan dengan metode template matching yaitu algoritma OCRChie yang berbasis *Sum of Squared Difference* (SSD) dengan fitur yang digunakan adalah fitur biner dari citra karakter tercetak (Mahastama & Tamatjita, 2008).

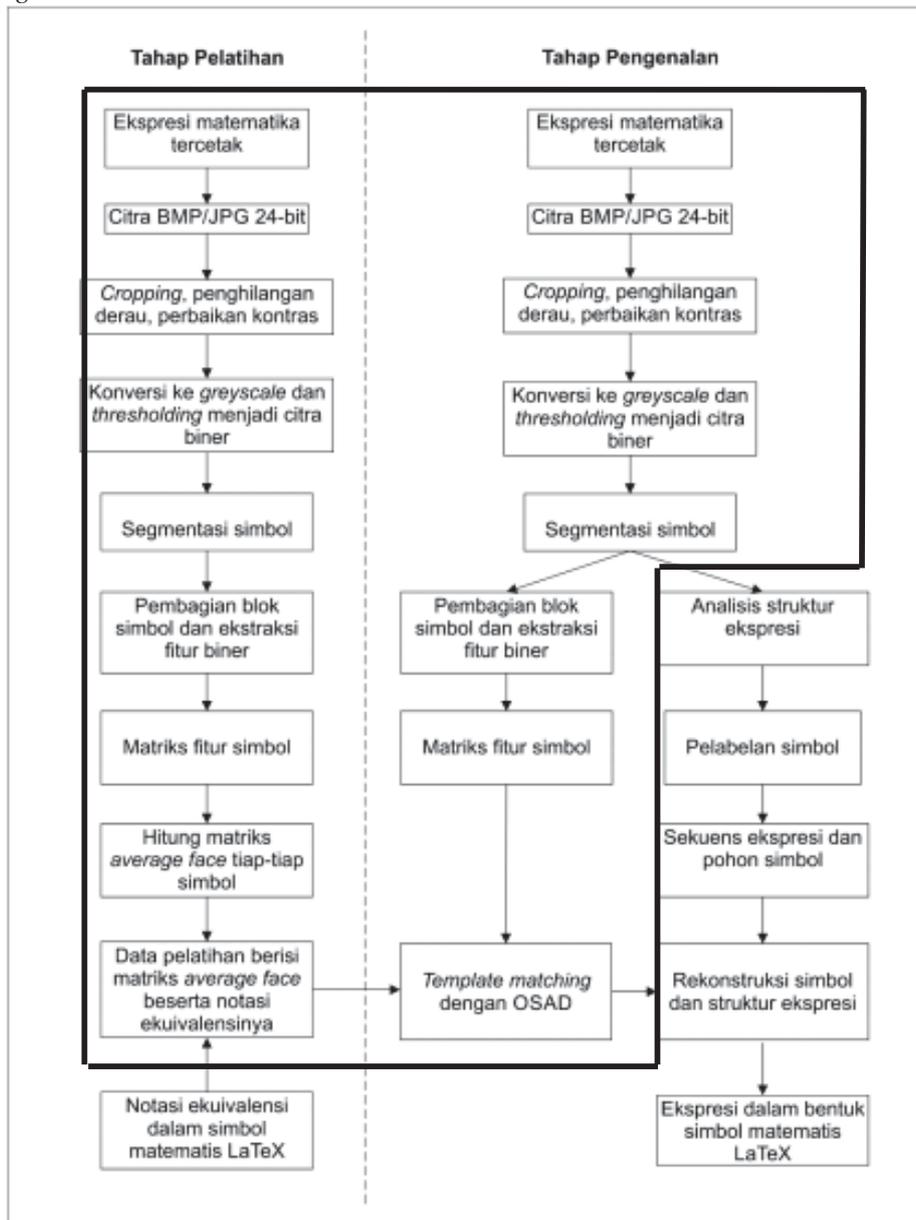
Penelitian ini mengkaji terutama pada pengenalan simbol pada ekspresi matematika dengan pendekatan *template matching* berbasis korelasi, menggunakan dua metode yang diperbandingkan yaitu *Sum of Squared Difference* (SSD) sebagai salah satu metode yang telah

¹ Program Studi Teknik Informatika, Fakultas Teknologi Informasi UKDW

banyak digunakan dalam *template matching*, serta algoritma *Optimised Sum of Absolute Difference* sebagai salah satu algoritma yang dikembangkan baru-baru ini dan menggunakan prinsip *average face* yaitu sebuah pola rata-rata yang mewakili seluruh pola yang merepresentasikan sebuah simbol, sehingga dapat menghemat waktu pencocokan. Di samping itu, OSAD memiliki performa pencocokan yang sangat baik terhadap basis data foto MIT (Dawoud, Samir, & Janier, 2011).

2. METODE PENELITIAN

Penelitian ini merupakan bagian dari seleksi metode yang akan digunakan pada penelitian mengenai konversi ekspresi matematika tercetak menjadi notasi LaTeX, oleh karena itu secara garis besar metode penelitian mengikuti alur pengenalan pola, seperti ditunjukkan oleh Gambar 1. Tahap-tahap yang dilakukan dalam penelitian ini ditunjukkan oleh bagian yang dibatasi oleh garis tebal, dengan metode SSD serta algoritma OSAD diujikan sebagai calon mesin pengenalan pola yang pada bagian terdapat pada bagian *template matching*.



Gambar 1. Rancangan Alur Penelitian

Tahapan-tahapan penting yang dilakukan dalam penelitian ini dijelaskan secara rinci dalam subbab berikut.

2.1. Thresholding

Citra pelatihan kemudian akan diubah menjadi citra biner melalui proses *thresholding*. Karena citra ekspresi berupa citra 24-bit, maka proses sebelum dilakukan *thresholding*, citra 24-bit terlebih dulu diubah ke citra keabuan (*greyscale*) menggunakan persamaan (1) (Finlayson, Qiu, & Qiu, 1999).

$$I = 0,333 \times R + 0,5 \times G + 0,166 \times B \quad (1)$$

I : intensitas keabuan
 R : intensitas warna pada kanal merah
 G : intensitas warna pada kanal hijau
 B : intensitas warna pada kanal biru

Selanjutnya citra keabuan tersebut diberi nilai ambang (*threshold*) di mana piksel yang memiliki intensitas keabuan di atas nilai ambang digolongkan sebagai piksel *background*, sementara piksel dengan intensitas keabuan sama dengan atau kurang dari nilai ambang digolongkan sebagai piksel *foreground*. Nilai ambang dimulai dari 180 (Mahastama & Tamatjita, 2008) kemudian disesuaikan bila perlu untuk mendapatkan hasil yang baik secara persepsi visual.

2.2. Segmentasi Simbol

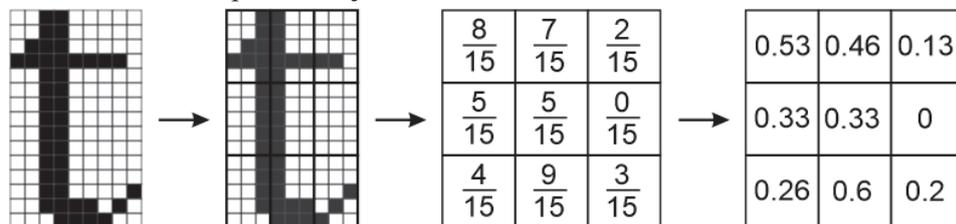
Citra pelatihan selanjutnya akan disegmentasi untuk mendapatkan simbol-simbol individual yang termuat dalam ekspresi yang ada pada citra. Segmentasi ini dilakukan menggunakan pendekatan *region growing* namun tidak memakai rekursi, melainkan perulangan dua dimensi saja.

Segmentasi ini perlu dilakukan dalam dua lapis yaitu deteksi segmen dan penyatuan segmen yang berimpit karena sifat dari arah perulangan yang digunakan, sehingga saat piksel diperiksa, kemungkinan terdapat kasus di mana sebuah simbol akan menghasilkan dua buah segmen.

2.3 Ekstraksi Fitur

Setelah didapatkan pasangan simbol dan ekuivalensi simbolnya, maka selanjutnya masing-masing simbol diekstrak fiturnya untuk mendapatkan matriks. Langkah-langkah ekstraksi fitur adalah sebagai berikut:

- Periksa koordinat ROI simbol, hitung lebar dan tinggi simbol, kemudian bagi seluruh area simbol menjadi 9 zona sama besar.
- Untuk setiap zona, hitung rasio piksel *foreground* terhadap seluruh piksel pada zona tersebut.
- Simpan hasil perhitungan rasio piksel seluruh zona ke dalam sebuah matriks fitur berukuran 3 x 3 seperti ditunjukkan oleh Gambar 2.



Gambar 2. Matriks Fitur

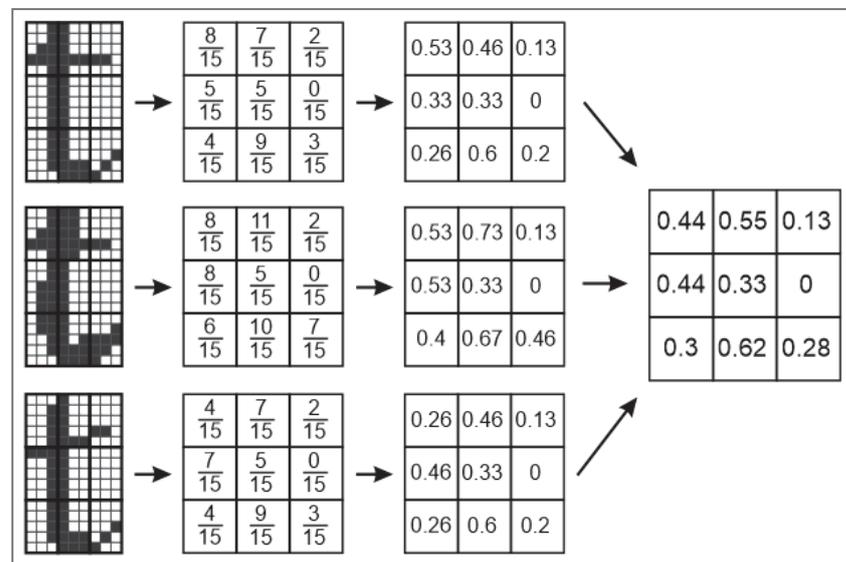
- Simpan pasangan matriks fitur dan ekuivalensi simbolnya sebagai sebuah *entry* dalam *data storage*.
- Lakukan langkah a hingga d untuk setiap simbol.

Hasil dari tahap ini adalah pasangan matriks fitur dengan ekuivalensi simbol untuk setiap simbol yang telah berhasil disegmentasi. Pasangan ini kemudian disimpan sebagai data pelatihan ke dalam tabel simbol di *data storage*.

2.4 Perhitungan Matriks Average Face (untuk OSAD saja)

Agar proses pengenalan simbol nantinya dapat dilakukan dengan lebih cepat, maka pada algoritma OSAD simbol uji yang dimasukkan tidak akan dicocokkan dengan setiap data pelatihan di tabel simbol, melainkan dengan sebuah data fitur rata-rata atau *average face* untuk setiap ekuivalensi simbol. Langkah-langkah penghitungan matriks *average face* adalah sebagai berikut (Dawoud, Samir, & Janier, 2011).

- Temukan data pelatihan yang memiliki ekuivalensi simbol yang sama pada tabel simbol di *data storage*.
- Baca matriks fitur setiap simbol, kemudian hitung nilai rata-rata untuk setiap elemen matriks.
- Simpan nilai rata-rata setiap elemen ke dalam matriks fitur *average face* yang berukuran sama dengan matriks fitur setiap simbol, sebagaimana ditunjukkan oleh Gambar 3.



Gambar 3. Matriks Fitur Average Face

- Simpan pasangan matriks fitur *average face* dan ekuivalensi simbol sebagai sebuah *entry* dalam tabel *average face* di *data storage*.

Matriks fitur *average face* akan selalu diperbaharui setiap kali pengguna memasukkan citra pelatihan yang baru, sehingga *average face* akan selalu dalam kondisi *up-to-date* setiap kali akan dilakukan uji pengenalan simbol. Untuk pengujian dengan SSD tidak melalui tahap ini.

2.5 Pengenalan Simbol

Matriks fitur yang didapatkan dari tahap sebelumnya, kemudian dicocokkan dengan *average face* yang telah dibangun berdasarkan data pelatihan. Proses pencocokan ini menggunakan dua metode yaitu metode Sum of Squared Difference (SSD) dan algoritma *template matching* Optimised Sum of Absolute Difference (OSAD).

Sebelumnya, citra yang dibagi menjadi sejumlah blok segi empat, yang kemudian diambil satu ciri atau fitur untuk setiap blok. Jumlah blok harus sama untuk citra acuan maupun citra uji. Terhadap seluruh citra acuan yang mewakili sebuah kelompok, dibuat citra *template* atau citra rata-rata (*average image*) yang merupakan nilai rata-rata dari setiap blok citra acuan sebagai perwakilan vektor fitur dari kelompok citra acuan tersebut. Citra *template* akan dicocokkan dengan SSD maupun OSAD terhadap citra uji, dengan semakin kecil nilai SSD maupun OSAD menunjukkan bahwa citra semakin mirip dengan citra pelatihan. Oleh karena itu sebagai hasil pencocokan akan diambil ekuivalensi simbol dengan nilai SSD

maupun OSAD terkecil. Seandainya nilai sama dengan nol, maka kedua citra tersebut sama persis (Dawoud, Samir, & Janier, 2011).

Karena sama-sama berbasis korelasi, maka antara *Sum of Squared Difference* (SSD) dan *Optimised Sum of Absolute Difference* (OSAD) bekerja dengan cara yang hampir sama. Berikut cara kerja metode SSD (Martin & Crowley, 1995):

1. Misal terdapat dua buah matriks fitur citra, yaitu A adalah matriks fitur citra uji dan B adalah matriks fitur simbol hasil pelatihan.
2. SSD dari matriks selisih dihitung berdasarkan persamaan (2) berikut.

$$d(A, B) = \sum_i \sum_j (A(i, j) - B(i, j))^2 \quad (2)$$

$d(A, B)$: selisih jarak fitur matriks A dan B

$A(i, j)$: komponen fitur matriks A pada kolom i dan baris j

$B(i, j)$: komponen fitur matriks B pada kolom i dan baris j

OSAD merupakan pengembangan dari *Sum of Absolute Difference* (SAD) yang bekerja mencocokkan citra dengan citra *template* sebagai berikut (Dawoud, Samir, & Janier, 2011):

1. Misal terdapat dua buah matriks fitur citra, yaitu A adalah matriks fitur citra uji dan B adalah matriks fitur *template*.
2. SAD dari matriks selisih dihitung berdasarkan persamaan (3) berikut.

$$d(A, B) = \sum_i \sum_j |A(i, j) - B(i, j)| \quad (3)$$

$d(A, B)$: selisih jarak fitur matriks A dan B

$A(i, j)$: komponen fitur matriks A pada kolom i dan baris j

$B(i, j)$: komponen fitur matriks B pada kolom i dan baris j

3. Persamaan SAD kemudian dinormalisasi. Persamaan tersebut ditunjukkan oleh persamaan (4) dan disebut sebagai OSAD.

$$d(A, B) = \sum_i \sum_j \frac{|A(i, j) - B(i, j)|}{\max(A(i, j) - B(i, j))} \quad (4)$$

$d(A, B)$: selisih jarak fitur matriks A dan B

$A(i, j)$: komponen fitur matriks A pada kolom i dan baris j

$B(i, j)$: komponen fitur matriks B pada kolom i dan baris j

$\max(A(i, j), B(i, j))$: nilai maksimum antara komponen A(i, j) dengan B(i, j)

Pada penelitian ini, matriks fitur simbol yang didapatkan dari tahap sebelumnya akan dicocokkan menggunakan OSAD terhadap kumpulan matriks fitur *average face* yang telah dimiliki sistem.

Perbedaan dengan SSD adalah pencocokan pada SSD dilakukan terhadap seluruh matriks fitur yang telah didapatkan dari proses pelatihan, maka seandainya terdapat 1000 matriks fitur hasil pelatihan yang merepresentasikan 10 simbol saja, SSD akan melakukan proses pencocokan terhadap 1000 buah matriks fitur tersebut, sedangkan pada OSAD hanya dilakukan terhadap 10 buah matriks *average face*, di mana satu matriks *average face* mewakili representasi satu buah simbol.

3. HASIL DAN PEMBAHASAN

Bahan yang digunakan dalam penelitian ini adalah citra *bitmap* berisi citra ekspresi matematika yang didapatkan dari hasil pemindaian halaman buku. Dari pemindaian didapatkan 120 buah ekspresi matematika dengan 1733 buah simbol. Setelah praproses dan segmentasi, didapatkan 1671 buah simbol yang layak digunakan sebagai bahan penelitian.

Simbol yang layak ini masih dipilah lagi menjadi 1540 simbol sebagai bahan, yang dibagi menjadi dua bagian mengikuti metode *holdout* yaitu dari 1000 simbol untuk pelatihan dan 640 simbol untuk pengujian.

Dari persentasi data pelatihan dan data uji, didapatkan irisan sebanyak 200 data yang ditujukan untuk mengevaluasi apakah simbol yang sudah pernah dilatihkan akan dapat dikenali dengan lebih baik ketika diujikan kembali. Secara detail 1000 simbol sebagai data pelatihan terdiri dari 800 simbol sebagai data pelatihan murni dan 200 sebagai data irisan, dan 640 simbol sebagai data uji terdiri dari 500 data uji murni, 200 data irisan dan 40 data yang merupakan *outlier*, dalam hal ini simbol yang menggunakan font yang tidak pernah dilatihkan sebelumnya.

Dari hasil pengujian menggunakan metode SSD maupun algoritma OSAD terhadap simbol-simbol tersebut didapatkan bahwa:

1. Simbol Uji yang bukan merupakan sampel pelatihan yang berhasil dikenali dengan tepat, dari 500 data uji, dengan SSD didapatkan 442 (88,4%) dan dengan OSAD didapatkan 469 (93,8%) data.
Untuk OSAD, 26 simbol salah dikenali sebagai salah satu diantara tanda minus, pecahan mendatar, bar, dan dot. Sisanya 5 simbol adalah kesalahan pengenalan seperti misalnya “o” dengan simbol derajat.
Untuk SSD, 41 simbol salah dikenali sebagai salah satu diantara tanda minus, pecahan mendatar, bar, dan dot. Sisanya 17 simbol adalah salah pengenalan.
2. Simbol Uji yang juga merupakan sampel pelatihan yang berhasil dikenali dengan tepat, dari 200 data uji, dengan SSD didapatkan 192 (96%) dan dengan OSAD didapatkan 186 (93%) data.
Untuk OSAD, 12 simbol salah dikenali sebagai salah satu diantara tanda minus, pecahan mendatar, bar, dan dot. Dua kesalahan simbol “o” dianggap sebagai derajat
Untuk SSD, 8 simbol salah dikenali sebagai salah satu diantara tanda minus, pecahan mendatar, bar, dan dot.
3. Simbol Uji dengan font berbeda yang dapat dikenali dengan tepat, dari 40 data uji, dengan SSD didapatkan 11 (27,5%) dan dengan OSAD didapatkan 28 (70%) data.

3.1 Temuan Pada Proses Pengenalan Simbol

Temuan yang dicatat sebagai kelompok ini adalah berbagai temuan yang mempengaruhi hasil eksperimen.

1. Kesalahan pengenalan paling banyak terjadi antara simbol tanda minus, pecahan mendatar, bar dan dot karena memiliki matriks fitur yang mirip. Seandainya ini diabaikan, maka tingkat pengenalan untuk SSD dengan data yang seluruhnya bukan data pelatihan, misalnya, dapat mencapai 96%, dan OSAD 99%.
2. Untuk data uji yang juga digunakan sebagai data pelatihan, SSD dapat memberikan hasil yang lebih baik karena pencocokan langsung terhadap matriks fitur masing-masing simbol dan dapat menemukan jarak minimum = 0 dibandingkan dengan jarak terhadap *average face*.
3. Untuk data uji dengan font yang berbeda dengan data pelatihan, OSAD memberikan hasil yang lebih baik karena penggunaan *average face* dapat mewakili simbol yang mirip meskipun secara *typeface* tidak sama persis, sedangkan untuk SSD karena dibandingkan langsung dengan matriks fitur masing-masing simbol pelatihan, jarak yang didapatkan lebih besar.

4. KESIMPULAN

Tingkat akurasi pengenalan simbol menggunakan algoritma *template matching* OSAD secara umum menghasilkan performa rata-rata yang lebih baik (85,6%) dibandingkan dengan metode SSD (70,6%), meskipun khusus untuk uji yang menggunakan data irisan yang juga digunakan pada tahap pelatihan, SSD mampu memberikan hasil yang lebih baik karena pencocokan tidak dilakukan terhadap *average face* tetapi langsung terhadap matriks fitur simbol sehingga memungkinkan adanya nilai SSD = 0 atau sama persis.

Perbandingan Algoritma Sum of Squared Difference (SSD) dan Optimised Sum of Absolute Difference (OSAD) Untuk Pengenalan Simbol Pada Citra Ekspresi Matematika Tercetak

Meskipun demikian, *average face* justru menjadi keunggulan algoritma OSAD jika digunakan untuk mengenali data uji yang belum pernah dilatihkan sebelumnya. Penggunaan *average face* tidak memiliki pengaruh negatif terhadap akurasi pengenalan pola, justru dapat membantu asosiasi simbol baru dengan ekuivalensi simbol pada basis data yang sudah disusun dengan baik, sekaligus mengurangi waktu yang dibutuhkan untuk proses pengenalan secara keseluruhan.

5. SARAN

Untuk pengembangan penelitian di masa depan, mungkin dapat dilakukan penelitian untuk memperoleh jumlah *zoning* atau pembagian area simbol yang akan diekstraksi sebagai fitur dengan lebih baik, atau cara *zoning* yang lebih ideal untuk meningkatkan akurasi metode *template matching* dengan pendekatan korelasi secara keseluruhan, sehingga hasilnya dapat dimanfaatkan untuk memperbaiki hasil pengenalan yang diberikan oleh SSD maupun OSAD.

DAFTAR PUSTAKA

- Alvaro, F., Sanchez, J-A., & Benedi, J-M. (2011). Recognition of Printed Mathematical Expressions Using Two-dimensional Stochastic Context-Free Grammars. *2011 International Conference on Document Analysis and Recognition*.
- Bluche, T. (2010). Mathematical Formula Recognition using Machine Learning Techniques. *Tesis*, University of Oxford.
- Chan, K., & Yeung, D. (2000). Mathematical expression recognition: a survey. *International Journal on Document Analysis and Recognition*, 3, 3-15.
- Dawoud, N. N., Samir, B. B., & Janier, J. (2011). Fast Template Matching Method Based Optimized Sum of Absolute Difference Algorithm for Face Localization, *International Journal of Computer Applications (0975-8887)*, Vol. 18-No.8.
- Eto, Y., & Suzuki, M. (2001). Mathematical formula recognition using virtual link network. *Washington, D.C.: International Conference on Document Analysis and Recognition*, 762-767.
- Finlayson, G. D., Qiu, G., & Qiu, M. (1999). *Contrast Maximizing and Brightness Preserving Color to Grayscale Image Conversion*.
- Mahastama, A.W., & Tamatjita, E. N. (2008). Optical Character Recognition dengan Algoritma OCRCHIE. *Konferensi Nasional Sistem Informasi (KNSI)*, Yogyakarta.
- Martin, J., & Crowley, J.L. (1995). Comparison of Correlation Techniques, *Conference on Intelligent Autonomous Systems (IAS) 1995*, Karlsruhe.
- Prusa, D., & Hlavac, V. (2007). Mathematical formulae recognition using 2d grammars, *International Conference on Document Analysis and Recognition*, 2, 849-853.
- Ranger, J., Wang, F., & Cheng, H. (2012). Optical Character Recognition of Printed Mathematical Symbols using A Hierarchical Classifier, *The 2012 International Conference on Image Processing, Computer Vision, and Pattern Recognition*.