Pengenalan Captcha dengan Multivalued Image Decomposition dan Vector Space Image Recognition

Irpan Pardosi*[1], Pahala Sirait[2], Michael Oktando[3], Wilham[4]
STMIK Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4573767, Fax. (061) 4567789
[1,2,3,4]Jurusan Teknik Informatika, STMIK Mikroskil, Medan
[1]irpan@mikroskil.ac.id, [2]pahala@mikroskil.ac.id,
[3]111110271@students.mikroskil.ac.id, [4]111110289@students.mikroskil.ac.id

Abstrak

Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHA) merupakan program untuk meningkatkan keamanan web. Pengenalan CAPTCHA menggunakan aplikasi sering mengalami kegagalan karena posisi dari simbol yang terlalu rapat, juga karena sulitnya melatih simbol baru jika gagal dikenali. Metode Naive Pattern Recognition Algorithm salah satu metode yang belum memberikan hasil yang maksimal karena kesalahan pada proses pengenalan simbol tidak dapat dilatih kembali sehingga aplikasi tetap tidak akan mengenali simbol tersebut. Metode Multivalued Image Decomposition dan Vector Space Image Recognition dapat memberikan hasil yang lebih maksimal dengan menggunakan Training Set, dimana simbol yang tidak dikenali akan dilatih/training agar proses pengenalan simbol selanjutnya lebih akurat. Pengujian dilakukan terhadap CAPTCHA dengan berbagai warna background, CAPTCHA dengan simbol yang saling berdekatan (menyatu) dan kombinasi warna simbol dengan background yang berbeda. Untuk CAPTCHA dengan simbol berukuran berbeda dan saling terhubung, tidak dapat dikenali. Dengan threshold 0.90, hasil pengujian dengan training set yang dilakukan terhadap dengan algoritma ini menunjukkan akurasi tingkat keberhasilan sebesar 87%.

Kata kunci— *CAPTCHA, Multivalued Image Decomposition, Vector Space Image Recognition, training set.*

Abstract

Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHA) is a program to enhance the security of the web. The introduction of CAPTCHA using applications often fail because position of a symbol is too tight, as well as the difficulty to train a new symbol if it fails to recognize. Naive Method Pattern Recognition Algorithm is one of the methods that do not provide maximum results due to errors in the process of recognition symbols can not be retrained so that the system still will not recognize the symbol. Methods Multivalued Image Decomposition and Space Vector Image Recognition will give maximum results by using Training Set, where symbols are not recognized will be trained in order to further the process of introducing a more accurate symbol. Test conducted on a CAPTCHA with a variety of background colors, CAPTCHA with symbols that are close together (fused) and background color combinations with different symbols. CAPTCHA with symbols for different sized and interconnected, could not be recognized. With threshold 0.90 test results with the training set is done with this algorithm shows the accuracy of a success rate of 87%.

Keywords— *CAPTCHA, Multivalued Image Decomposition, Vector Space Image Recognition, training set.*

1. PENDAHULUAN

Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHA) merupakan program keamanan yang digunakan untuk membedakan antara manusia dan program komputer sebagai pengguna serta menghalangi autoscript yang tidak diinginkan. Namun program keamanan ini masih dapat dilewati, salah satunya dengan menggunakan metode Naive Pattern Recognition Algorithm [1]. Hasil dari penelitian tersebut masih perlu dikembangkan karena terjadi

kesalahan pada pengenalan simbol maka program tetap tidak akan mengenali simbol tersebut. Hal ini membuat kemampuan program untuk membaca CAPTCHA menjadi terbatas.

Pengenalan CAPTCHA juga dilakukan dengan berbagai algoritma seperti Optical Character Recognition (OCR) ataupun Vector Space Image Recognizer (VSIR) yang didapatkan bahwa algoritma VSIR lebih baik [2]. Ada juga yang menggunakan Convolutional Neural Networks (CNN)[3], namun pada algoritma-algoritma tersebut jika terjadi proses kegagalan saat pengenalan CAPTCHA maka harus kembali ke proses training set karena tidak dapat dilatih secara langsung.

Benjamin Boyter merekomendasikan metode Multivalued Image Decomposition untuk mengekstraksi simbol berupa angka atau huruf dan membuat histogram warna dari gambar CAPTCHA[4]. Algoritma Disjoint Sets of Pixels digunakan untuk menentukan letak setiap karakter yang ditentukan secara horizontal, kemudian pendekatan metode Vector Space Image Recognition digunakan untuk pengenalan pola simbol. Pendekatan ini didesain akan mampu untuk menambah atau menghapus simbol yang sulit dikenali, misalnya simbol O dan 0 (nol). Kelebihan dari metode yang diperkenalkan ini adalah metode ini memiliki tingkat keberhasilan pengenalan simbol yang sangat tinggi[4]. Pengujian yang dilakukan untuk mendapatkan akurasi pengenalan CAPTCHA serta kemudahan untuk melatih simbol yang tidak dikenali tanpa ke proses training set.

2. METODE PENELITIAN

Untuk mendapatkan hasil akurasi pengenalan CAPTCHA pada penelitian ini pengujian dilakukan hanya pada CAPTCHA visual based dua dimensi berformat .gif berukuran 60x20 hingga 120x40 sebanyak 100 gambar yang terdiri dari simbol-simbol dengan warna seragam serta tidak saling berdekatan (berkaitan), karena proses pengenalan CAPTCHA akan dilakukan secara horizontal.

2.1. Analisis Masalah

Proses pengenalan CAPTCHA sering mengalami kegagalan diakibatkan berbagai hal diantaranya belum adanya data training sesuai dengan CAPTCHA yang diuji, atau bentuk simbol yang berbeda dari yang biasa sehingga tidak dikenali. Pengenalan CAPTCHA secara umum diawali dengan preprocessing berupa training set terhadap CAPTCHA, namun dalam satu gambar CAPTCHA yang terdiri dari beberapa simbol bisa saja hanya 1 simbol yang tidak dikenali, biasanya solusi untuk hal ini maka dilakukan proses training set kembali sehingga akan menyulitkan jika seringa terjadi kegagalan. Algoritma yang diterapkan harus mampu menyelesaikan permasalah tersebut baik dari segi kemudahan melatih data dan keakuratannya.

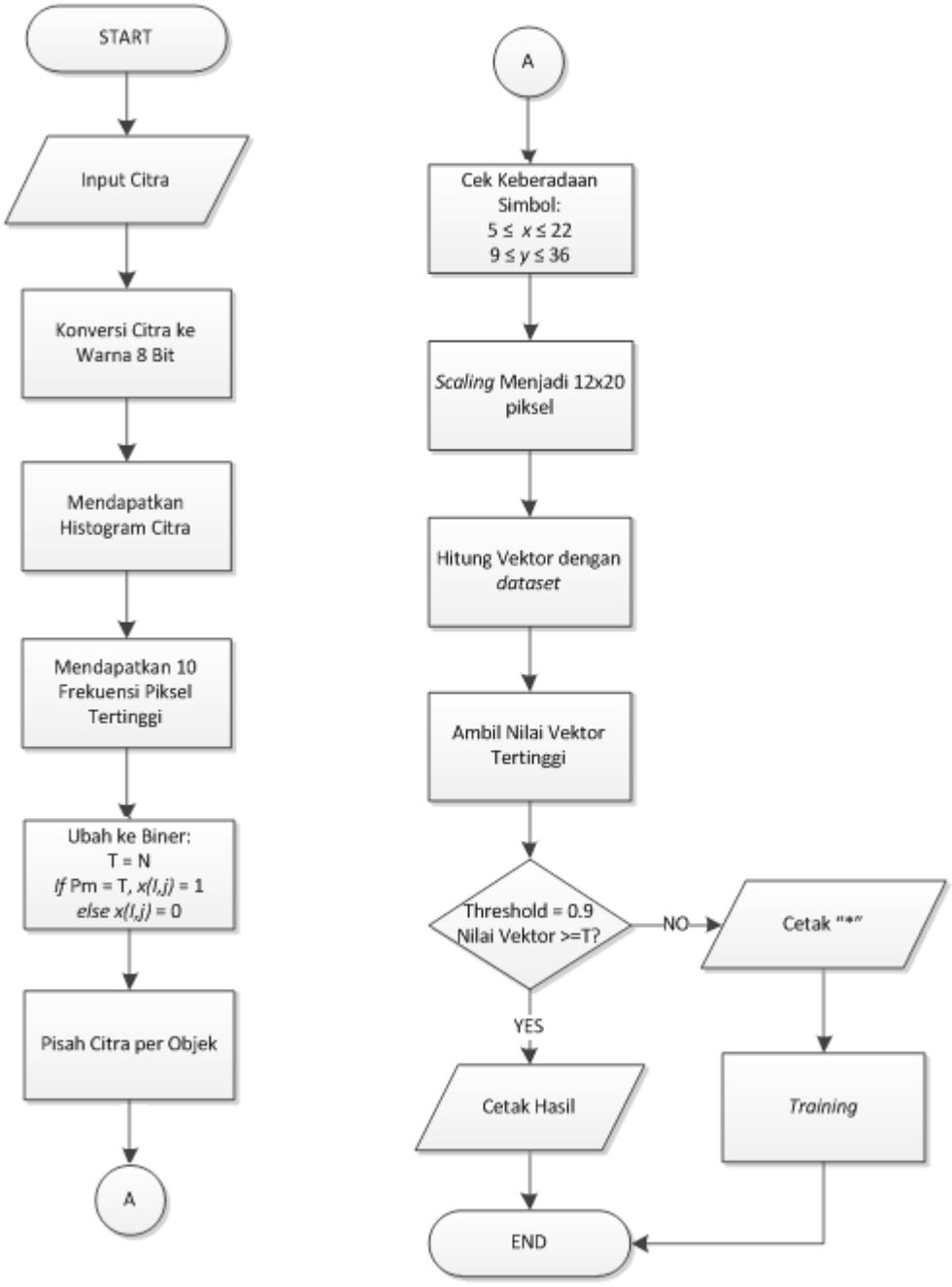
2.2. Analisis Proses

Pada bagian ini akan dijelaskan proses kerja dari sistem pembaca CAPTCHA mulai dari tahap awal hingga selesai dapat dilihat pada diagram alir Gambar 1.

2.3. CAPTCHA

CAPTCHA (*Completely Automated Public Turing test to tell Computers and Human Apart*) pada dasarnya adalah suatu program yang sebagian besar manusia dapat melewatinya, akan tetapi komputer tidak dapat melewatinya[5]. Aplikasi CAPTCHA banyak digunakan pada penyedia *web mail* contohnya Hotmail dan Yahoo!. CAPTCHA dikembangkan untuk mencegah program robot atau *bots* yang menciptakan ratusan *email account* untuk mengirimkannya ke *user*. *Bots* ini digunakan oleh *spammer* untuk melakukan penyerangan terhadap sistem dengan menggunakan *HTTP POST request submission*. Program robot akan mengambil nilai variabel yang terdapat pada *HTTP POST request* tersebut dari *form* yang akan disubmit sebelumnya dan mengirimkannya kembali secara berulang-ulang. Penyerang dapat dengan mudah melakukan hal tersebut dengan menulis *script* menggunakan bahasa *perl*. CAPTCHA terbagi dalam beberapa jenis, antara lain :

1. Berdasarkan Visual (*Visual Based*)



Gambar 1. Diagram Alir Proses Pembacaan CAPTCHA

Visual Based CAPTCHA memiliki beberapa variasi, yang paling umum digunakan saat ini adalah simbol yang dimiringkan dan dimunculkan pada sebuah gambar dan pengenalan bentuk[5]. CAPTCHA yang menggunakan simbol dimiringkan yang dimunculkan pada gambar disebut Gimpy, EZ Gimpy adalah varian dari Gimpy, Pessimal Print dan buffletext. Gimpy pertama kali dikembangkan oleh Luis Von Ahn dari Carnegie Mellon University yang mendesain versi paling sederhana dari Gimpy, disebut EZ-Gimpy. Ez-Gimpy yang sekarang ini digunakan oleh Yahoo! dan suatu versi serupa digunakan oleh Hotmail. Perbedaan yang mendasar antara Gimpy dan EZ-Gimpy adalah Gimpy memiliki tiga atau lebih kata yang dimiringkan yang dimunculkan pada suatu gambar, sedangkan EZ-Gimpy hanya memiliki satu kata yang dimiringkan pada suatu gambar.

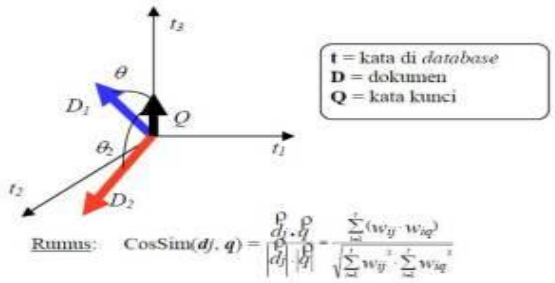
Gambar 2. Contoh Visual Based CAPTCHA[5]

2. Berdasarkan suara (*Sound Based*)
   Sound based CAPTCHA kebanyakan digunakan untuk membantu mereka yang buta atau mempunyai masalah dengan penglihatan[5]. Suatu contoh sound based CAPTCHA adalah bunyi yang sesuai. CAPTCHA ini digunakan pada Hotmail, Yahoo! dan Altavista sebagai tambahan terhadap CAPTCHA visual based ketika pendaftaran sebuah email account untuk masing-masing penyedia layanan email ini. Tes ini menjalankan klip audio yang berisi rekaman suatu urutan kata atau angka-angka yang dimiringkan dan jika kata atau angka-angka yang diduga tepat maka dapat melewati tes ini.

Gambar 3. Contoh Sound Based CAPTCHA[5]

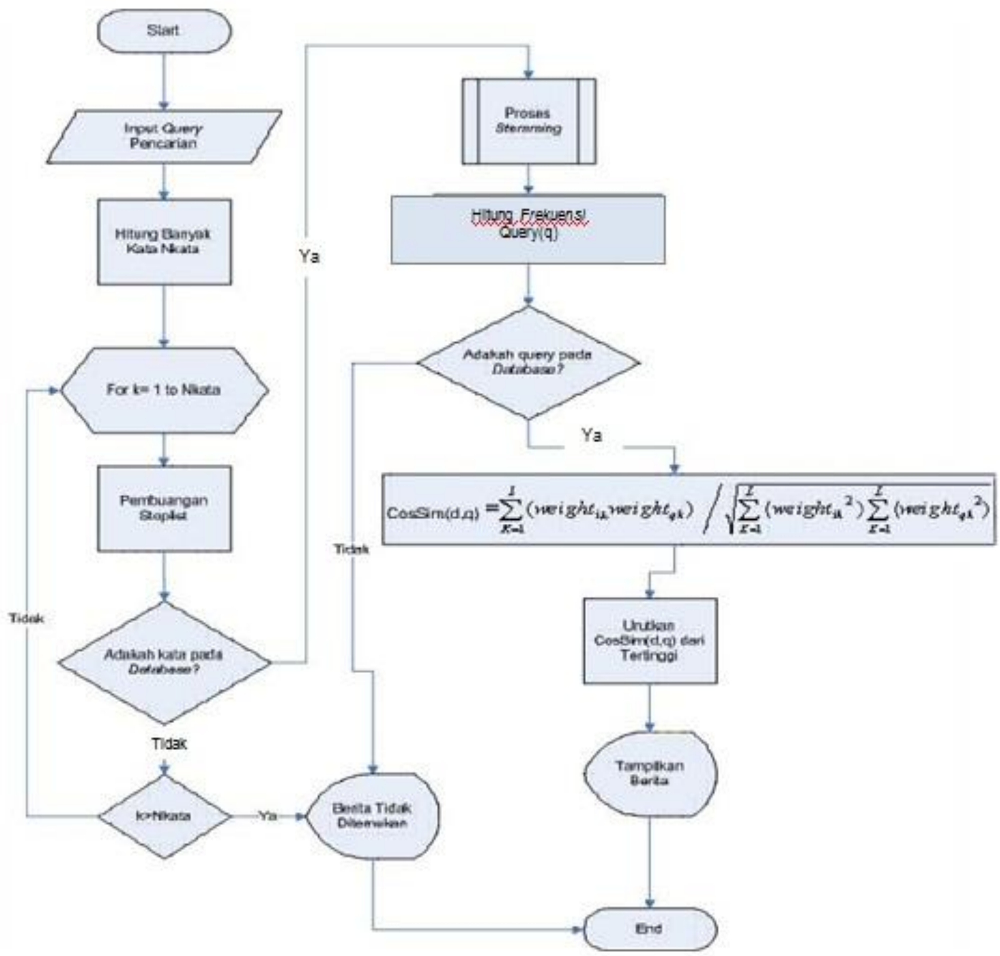
2.4. Vector Space Model

Vector Space Model merupakan suatu model yang digunakan untuk mengukur kemiripan antara data yang ada di database dengan query. Query dan data dianggap sebagai vektor-vektor pada ruang n-dimensi, dimana t adalah jumlah dari seluruh term yang ada dalam leksikon[6]. Leksikon adalah daftar semua term yang ada dalam indeks. Selanjutnya akan dihitung nilai cosinus sudut dari dua vektor, yaitu W dari tiap dokumen dan W dari kata kunci.



Gambar 4. Rumus Vector Space Model[6]

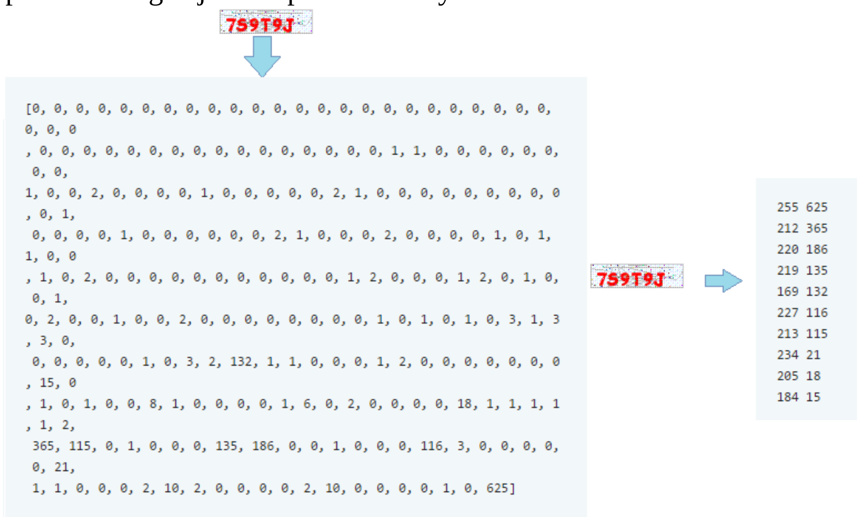
Vector Space Model solusi atas permasalah yang dihadapi jika menggunakan algoritma TF/IDF. Karena pada algoritma TF/IDF terdapat kemungkinan antar dokumen memiliki bobot yang sama, sehingga ambigu untuk diurutkan. Adapun Flowchart dari pencarian menggunakan algoritma *Vector Space Model* seperti pada Gambar 5.



Gambar 5. Pencarian dengan Vector Space Model[6]

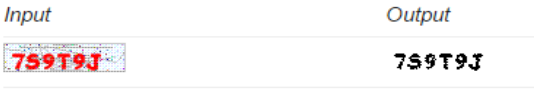
2.5. Multivalued Image Decomposition

Metode ini digunakan untuk mengekstraksi simbol pada CAPTCHA. Metode ini menggunakan ekstraksi warna yang ada pada gambar untuk mendapatkan simbol yang ada digambar[4]. Gambar akan dibuat ke histogram untuk mendapatkan nilai pikselnya. Kemudian dari histogram tersebut akan disusun menjadi 10 grup warna dengan jumlah piksel terbanyak.



Gambar 6. Histogram dan hasil 10 pixel terbanyak hasil histogram [4]

Dari 10 grup histogram tersebut akan dicari nilai piksel simbol dan dibuat menjadi sebuah gambar baru berdasarkan grup piksel tersebut. Hasil dari proses ini merupakan sebuah *binary image* yang berisi simbol dari CAPTCHA tersebut.

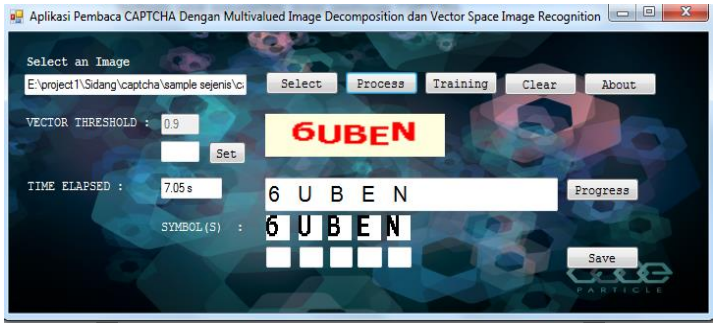


Gambar 7. Hasil Ekstraksi CAPTCHA [4]

3. HASIL DAN PEMBAHASAN

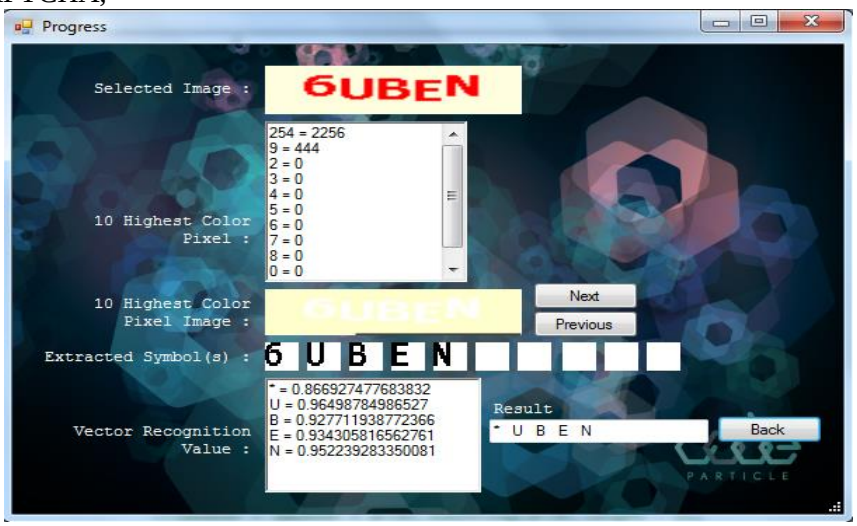
3.1. Hasil

Sebelum melakukan pembacaan CAPTCHA, pengguna harus memilih citra dengan format GIF, PNG atau BMP. Untuk mendapatkan hasil pembacaan yang lebih akurat, disarankan agar citra masukan memiliki ukuran diantara 60x20 piksel sampai dengan 120x40 piksel. Setelah pembacaan berhasil dilakukan, aplikasi menampilkan simbol yang terbaca (berupa huruf dan atau anngka) serta menampilkan "*" untuk simbol yang tidak terbaca. Pengguna dapat melakukan Training Set langsung pada form yang sama, baik untuk simbol yang tidak terbaca maupun simbol yang terbaca dengan menekan tombol Save.



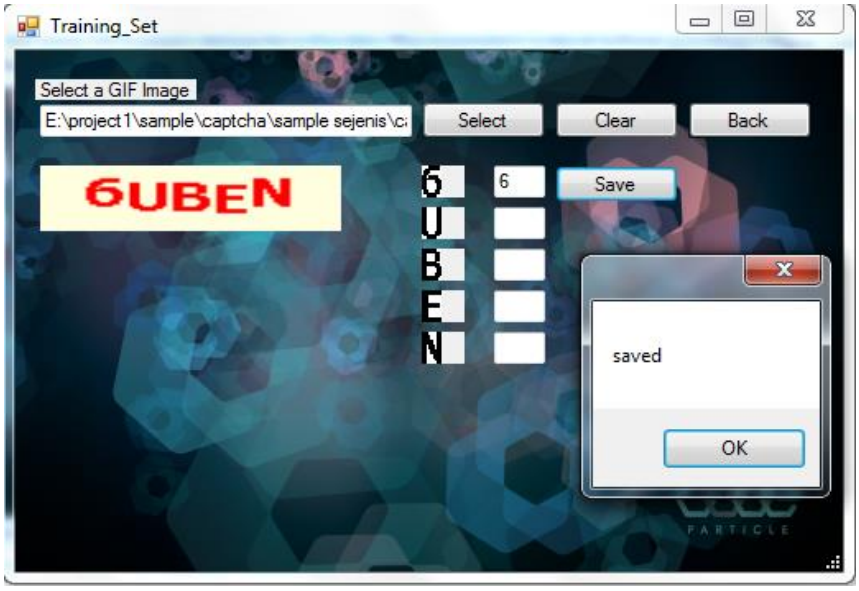
Gambar 8. Tampilan Akhir User Interface Form Home

Pada *form progress* ini pengguna dapat melihat hasil dari proses pembacaan CAPTCHA yang meliputi citra masukan, sepuluh frekuensi piksel tertinggi dari citra masukan, sepuluh citra dari frekuensi piksel tertinggi, hasil ekstraksi simbol, hasil perhitungan vektor (0 sampai dengan 1) dan hasil pembacaan CAPTCHA,



Gambar 9. Tampilan Akhir User Interface Form Progress

Sesuai dengan permasalah diawal agar hasil pembacaan CAPTCHA lebih akurat, aplikasi menyediakan fitur *Training Set* yang berfungsi untuk mengenalkan simbol yang sebelumnya tidak dikenali apliikasi ini.



Gambar 10. Tampilan Akhir User Interface Form Training Set

3.2. Pembahasan

Proses pengujian menggunakan 100 *sample* CAPTCHA sederhana dengan *threshold* 0.95, 0.90 dan 0.85 tanpa *Training* Set untuk menentukan *threshold* mana yang paling optimal. Kemudian dilanjutkan dengan pengujian terhadap 100 *sample* CAPTCHA yang sama dengan *threshold* yang paling optimal dan *Training Set*. Pengujian berikutnya adalah menggunakan CAPTCHA dengan beberapa variasi, yaitu warna, ukuran dan *margin* untuk mengetahui apakah varian ini berpengaruh terhadap kinerja algoritma. Berikut adalah proses pengujian terhadap 100 *sample* CAPTCHA sederhana dengan *threshold* yang berbeda, hasilnya dapat dilihat pada tabel 1 dibawah ini.

Tabel 1. Hasil Pengujian Terhadap 100 Sample CAPTCHA tanpa Training Set

JUMLAH CAPTCHA : 100	**T=0.95**	**T=0.9**	**T=0.85**
BERHASIL	0	1	40
GAGAL	100	99	60

JUMLAH SIMBOL : 639	**T=0.95**	**T=0.9**	**T=0.85**
BERHASIL	175	362	554
GAGAL	464	249	0
SALAH BACA	0	28	85

Kriteria “Berhasil” apabila seluruh simbol di dalam CAPTCHA dapat terbaca, jika terdapat 1 atau lebih simbol yang tidak terbaca atau salah terbaca, maka dianggap gagal. Dari Tabel 1 dapat dilihat bahwa pembacaan CAPTCHA dengan *threshold* 0.95 kurang akurat karena nilai *threshold* yang terlalu tinggi dibandingkan dengan *dataset* sedangkan hasil pembacaan dengan *threshold* 0.90 lebih baik daripada pengujian dengan *threshold* 0.95. Namun kesalahan baca terjadi pada fase ini, dimana huruf “O” dibaca sebagai angka 0 dan huruf “M” dibaca sebagai huruf “H”. Pada pengujian dengan *threshold* 0.85, kegagalan pengenalan objek sudah tidak ada. Tetapi tingkat kegagalan baca meningkat, yaitu

angka 6 dibaca sebagai angka 5, huruf "T" dibaca sebagai huruf "I" dan huruf "Q" dibaca sebagai huruf "O" dan didapatkan *threshold* yang paling optimal, yaitu T = 0.90. Selanjutnya akan dilakukan pengujian terhadap 100 *sample* CAPTCHA yang sama dengan *Training Set* dengan *threshold* = 0.90 dan didapatkan hasil sebagai berikut:

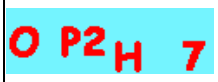
Tabel 2. Hasil Pengujian Terhadap 100 Sample CAPTCHA dengan Training Set

JUMLAH CAPTCHA : 100	**T=0.9**
BERHASIL	87
GAGAL	13

JUMLAH SIMBOL : 639	**T=0.9**
BERHASIL	624
GAGAL	13
SALAH BACA	2

Dari hasil pengujian pada Tabel 2. diatas, didapatkan 87 CAPTCHA yang berhasil dibaca, 13 CAPTCHA yang gagal (87%) dan dari 639 simbol yang ada terdapat 624 simbol yang berhasil dibaca, 13 simbol gagal terbaca dan 2 simbol yang salah terbaca (97.65%). Dari hasil pengujian yang dilakukan terhadap CAPTCHA dengan *margin* yang bervariasi dapat dilihat pada tabel 3 bahwa perangkat lunak mampu mengenali simbol. Namun jika simbol berdampingan satu sama lain, maka simbol tersebut akan dianggap sebagai satu simbol sehingga hasil pembacaan menjadi gagal.

Tabel 3. Hasil Pengujian Terhadap CAPTCHA dengan margin yang Beragam

No	CAPTCHA	CAPTCHA	HASIL	UKURAN	KESIMPULAN
1	C7WA26			120X40	BERHASIL
2	0P2H7			120X40	BERHASIL
3	1TS79			120X40	BERHASIL
4	C3LQP			120X40	BERHASIL
5	9R20Z			120X40	GAGAL KARENA HURUF BERDAMPINGAN

Dari hasil pengujian tabel 4 di bawah yang dilakukan terhadap CAPTCHA dengan ukuran yang bervariasi dapat dilihat bahwa ukuran CAPTCHA berpengaruh terhadap hasil perhitungan vektor dengan *dataset*. Jika ukuran citra terlalu kecil atau terlalu besar, maka ada kemungkinan dimana simbol tidak dapat terbaca pada saat pengecekan keberadaan simbol atau simbol tidak dikenali oleh program.

Tabel 4. Hasil Pengujian Terhadap CAPTCHA dengan Ukuran yang Beragam

CAPTCHA	3TS79	3TS79	3TS79	3TS79	3TS79
GAMBAR CAPTCHA					
UKURAN CITRA	40X16	70X23	80X22	98X26	180X60
HASIL	-	* T * * *	* T * * *	3 T * 7 9	* * * *
KESIMPULAN	Gagal melewati cek simbol	3= 0.79530 S= 0.82548 7=0.79598 9=0.89417	3=0.78825 S= 0.81388 7=0.78428 9=0.86522	S=0.89221	Gagal dibaca

Berdasarkan hasil dari pengujian pada tabel 5 yang dilakukan terhadap CAPTCHA dengan warna yang beragam, perangkat lunak mampu untuk membaca simbol. Simbol berhasil diekstrak dengan menggunakan algoritma *Multivalued Image Decomposition* yang dapat memisahkan simbol dari *background*.

Tabel 4. Hasil Pengujian Terhadap CAPTCHA dengan Warna yang Beragam

No	CAPTCHA	GAMBAR CAPTCHA	HASIL	UKURAN CITRA	KESIMPULAN
1	8KU0O			120x40	BERHASIL
2	DXS15			120x40	BERHASIL
3	YI1SKTB6			120x40	BERHASIL
4	A5WFXH			120x40	BERHASIL
5	C7WA26			120x40	BERHASIL

Berdasarkan pengujian yang dilakukan terhadap aplikasi ini, dapat dilihat bahwa *threshold* vektor yang paling optimal adalah 0.90. Pada *threshold* vektor 0.95 banyak terjadi kegagalan pengenalan simbol yang disebabkan oleh terlalu tingginya nilai *threshold*, sedangkan pada *threshold* vektor 0.85 banyak terjadi kesalahan pengenalan simbol, seperti huruf "T" yang dibaca sebagai huruf "I" dan angka 6 dibaca sebagai angka 5. *Threshold* memiliki pengaruh terhadap hasil pembacaan CAPTCHA dan simbol sehingga *threshold* menentukan tingkat akurasi pembacaan dari perangkat lunak. *Margin* dan warna simbol tidak berpengaruh pada hasil perhitungan vektor aplikasi. Faktor yang berpengaruh pada hasil perhitungan vektor adalah ukuran simbol. Apabila simbol pada CAPTCHA terlalu kecil atau terlalu besar, maka perangkat lunak akan gagal mengenali simbol tersebut.

4. KESIMPULAN

Berdasarkan hasil pengujian, didapatkan beberapa kesimpulan sebagai berikut:

1. Algoritma *Multivalued Image Decomposition dan Vector Space Image Recognition* mampu untuk membaca kembali CAPTCHA yang sebelumnya tidak terbaca dengan menggunakan Training Set dari aplikasi.
2. Nilai *Threshold* sangat berpengaruh pada tingkat akurasi pembacaan CAPTCHA, Nilai *threshold* 0.90 merupakan ukuran yang paling optimal untuk dataset yang digunakan.
3. Ukuran CAPTCHA mempengaruhi hasil perhitungan vektor. Ukuran yang terlalu kecil atau besar menyebabkan simbol tidak terdeteksi dan akan salah dikenali, Namun warna dan margin tidak memberikan pengaruh.
4. Jika simbol menyatu (*join*) akan dianggap menjadi 1 simbol dan menyebabkan kesalahan saat pembacaan.

5. SARAN

Adapun beberapa saran yang dapat disampaikan oleh penulis dalam penelitian ini diantaranya sebagai berikut:

1. Penerapan dapat dilakukan terhadap CAPTCHA tiga dimensi dan CAPTCHA audio.
2. Proses pengenalan CAPTCHA dapat juga dilakukan terhadap CAPTCHA yang memiliki margin yang terlalu dekat / terhubung.
3. Penentuan nilai threshold dapat dilakukan secara otomatis sesuai kriteria CAPTCHA.

DAFTAR PUSTAKA

[1] J. Yan and a. S. El Ahmad, “Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorithms,” Twenty-Third Annu. Comput. Secur. Appl. Conf. (ACSAC 2007), pp. 279–291, 2007.

[2] M. Korakakis, E. Magkos, and P. Mylonas, “Automated CAPTCHA solving: An empirical comparison of selected techniques,” Proc. - 9th Int. Work. Semant. Soc. Media Adapt. Pers. SMAP 2014, pp. 44–47, 2014.

[3] F. Stark, R. Triebel, and D. Cremers, “CAPTCHA Recognition with Active Deep Learning.”

[4] Boyter, B., 2010, Decoding CAPTCHA's, tersedia pada : http://www.boyter.org/ decoding-captchas/, tanggal akses : 03 November 2016.

[5] Riadi, I., 2008, Optimalisasi Keamanan Website Menggunakan CAPTCHA, Seminar Nasional Informatika 2008, UPN Veteran Yogyakarta, tersedia : http://jurnal.upnyk.ac.id/index.php/ semnasif/article/view/740, tanggal akses : 25 Oktober 2016

[6] Harjono, K. D. 2005, Perluasan Vektor Pada Metode Search Vector Space. Integral Vol. 10 No.2, Jurusan Ilmu Komputer, Universitas Katolik Parahyangan, Bandung

[7] Agustian, I. M., 2013, Definisi Citra, tersedia pada : http://te.unib.ac.id/lecturer/indraagustian/ 2013/06/defnisi-citra/, tanggal akses : 18 Agustus 2016.

[8] Fauji, S. A., 2012, Citra Digital dan citra analog, tersedia pada: http://shofwanalifauji.blogspot.com/2012/03/citra-digital-dan-citra-analog.html, tanggal akses: 05 September 2016.

[9] Kusumaningsih, I., 2009, Ekstraksi Ciri Warna, Bentuk dan Tekstur Untuk Temu Kembali Citra Hewan, tersedia pada : http://repository.ipb.ac.id/bitstream/handle/123456789/13031/ G09iku.pdf;jsessionid=AD9F8F165294643A992E453B116ECC18?sequence=11, tanggal akses : 02 Nopember 2016.

[10] Munir, R., 2004, Pengolahan Citra, tersedia pada : http://informatika.stei.itb.ac.id/~rinaldi.munir/ Buku/Pengolahan%20Citra%20Digital/E-book.htm, tanggal akses : 05 Oktober 2016.