

Aplikasi Morse Code Translator Metode Klasifikasi Euclidean Distance dengan Algoritma Ocrchie Untuk Menerjemahkan Kode Morse

Alvin Wijaya¹⁾, Eric Kwok²⁾, Halim Agung³⁾

Teknik Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia
Jalan Lodan Raya No.2, Pademangan, Jakarta Utara, 14430

¹⁾Email: manofdebest@gmail.com,

²⁾Email: eric_artic@yahoo.com,

³⁾Email: hagung@bundamulia.ac.id

Abstract: Language is a tool a person uses to give birth to thoughts in feelings. Language also serves as a means of communication among members of the community as users of the language, so that each other to inform their ideas and feelings of the information. The information provided is a meaning that can be understood by members of the community, so it becomes social interaction. There are more than 6000 languages in the world, including sign languages and languages that use patterns in the form of lines and dots that we know with Morse code. But not all people know / know Morse code because the language is often used in the field of shipping. To facilitate the public in recognizing Morse code, the authors make an application of Morse code translator through images using OCRchie algorithm where this algorithm analyzes morse code in the image. It is expected that the application can translate morse code in the image into a character that can be understood by the user of the application.

Keywords: Euclidean Distance, Morse Code, OCRChie, Translator

Abstrak: Bahasa adalah alat yang digunakan seseorang untuk melahirkan pikiran-pikiran dalam perasaan. Bahasa juga berfungsi sebagai alat komunikasi antar anggota masyarakat sebagai pemakai bahasa, sehingga saling menginformasikan gagasan dan perasaannya dari informasi tersebut. Informasi yang diberikan tersebut adalah makna yang dapat dimengerti oleh anggota masyarakat, sehingga menjadi interaksi social. Terdapat lebih dari 6000 bahasa di dunia, termasuk bahasa isyarat dan bahasa yang menggunakan pola yang berbentuk garis dan titik yang kita kenal dengan kode morse. Tapi tidak semua masyarakat mengenal/mengetahui kode morse karena bahasa tersebut sering dipakai di bidang pelayaran. Untuk mempermudah masyarakat dalam mengenal kode morse, maka penulis membuat suatu aplikasi penerjemah kode morse (Morse Code Translator) melalui gambar menggunakan algoritma OCRchie dimana algoritma ini menerjemahkan kode morse yang ada di gambar. Diharapkan aplikasi dapat menerjemahkan kode morse yang ada di gambar menjadi suatu karakter yang dapat dimengerti oleh pengguna aplikasi.

Katak kunci: Euclidean Distance, Kode Morse, Penerjemah, OCRChi

I. PENDAHULUAN

Bahasa merupakan suatu kemampuan yang dimiliki manusia untuk berkomunikasi melalui tanda atau sebuah kata. Di dunia terdapat 6000 sampai 7000 bahasa baik itu tanda isyarat maupun lisan. Dan komunikasi merupakan hal penting bagi setiap orang untuk berinteraksi antar individu.

Diantara bahasa-bahasa tersebut, salah satunya adalah kode *morse*, yang merupakan kode yang biasanya diaplikasikan pada kapal laut untuk berkomunikasi. Namun, kode *morse* bukanlah bahasa yang mudah untuk dikuasai. Oleh karena itu, penting untuk memahami tentang kode *morse*. Berdasarkan

pada penjelasan yang dipaparkan di atas, peneliti ingin membuat sebuah aplikasi berbasis *desktop* yang dapat digunakan untuk menerjemahkan kode *morse* melalui gambar menggunakan algoritma *OCRchie*.

Pada penelitian terdahulu yang menggunakan metode Euclidean Distance yang berjudul Pengenalan Pola Tekstur *Brodatz* Dengan Metode Jarak *Euclidean* [1], peneliti menggunakan metode *Euclidean Distance* dalam mengetahui kemiripan pola tekstur pada gambar. *Euclidean Distance* dapat ditentukan ketika Ciri Statistik Orde Dua sudah dihitung dan datanya dijadikan *data training*. Lalu dengan metode *Euclidean Distance*, ditentukan klasifikasi gambar tersebut. Lalu pada penelitian yang berjudul

Analisis Kemiripan Pola Citra *Digital* Menggunakan Metode *Euclidean* [2], *peneliti* menggunakan metode *Euclidean Distance* dalam mengetahui kemiripan pola tekstur pada gambar. *Euclidean Distance* dapat ditentukan ketika Ciri Statistik Orde Dua sudah dihitung dan datanya dijadikan *data training*.

II. METODE PENELITIAN

Kode Morse [3] adalah metode mentransmisi informasi teks dalam urutan nada *on-off*, cahaya, atau klik yang bisa langsung dimengerti oleh pendengar atau pengamat berpengalaman tanpa alat khusus. *Morse code* dinamai berdasarkan penemu telegraf, *Samuel F. B. Morse*. *Morse code* internasional menyandi huruf latin dasar *ISO*, beberapa huruf latin spesial, angka *Arabic*, dan *prosigns* yang berupa standarisasi urutan *signal* pendek dan panjang yang disebut sebagai titik dan garis (*dots and dashes*). Setiap simbol *morse code* merepresentasikan sebuah teks karakter (huruf atau angka) atau *prosign*. Durasi lamanya *signal* panjang yang berupa *dashes*/garis adalah tiga kali lipatnya durasi *dots*/titik. Setiap garis/titik disela oleh suara sunyi singkat. Jarak antar garis/titik adalah satu garis.

Euclidean Distance [4] [5] adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. *Euclidean* ini biasanya diterapkan pada 2 dimensi dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. Rumus Metode *Euclidean Distance* adalah sebagai berikut:

$$d_e = \sqrt{\sum_{k=1}^m (f_{i,k} - k_j)^2}$$

OCRChie [6] [7] adalah algoritma pengenalan karakter yang diriset oleh oleh *Kathey Marsden* dibawah bimbingan Profesor *Richard J. Fateman* dari Departemen Teknik Universitas *Berkeley*, Amerika Serikat. Algoritma ini diterapkan pada citra hitam putih (1-bit), dan mempergunakan rasio piksel hitam dan piksel putih sebagai acuan untuk melakukan pengenalan sebuah karakter. Ciri dari algoritma *OCRChie* adalah sebagai berikut:

1. Learning Character

Algoritma *OCRChie* bergantung pada sekumpulan learning character atau sebuah kamus karakter yang akan digunakan dipergunakan sebagai acuan dalam mengenali karakter dalam citra input,

learning characters ini terangkum dalam sebuah learning set. Setelah *learning set* selesai dibaca dan dikenali, hasil pengenalan tersebut disimpan pada sebuah *file* terpisah dalam bentuk data numerik yang lebih ringkas tanpa harus mempertahankan citra asli yang terdapat dalam *learning set*. Bentuk penyimpanan seperti ini juga mengurangi waktu pembacaan kembali *learning set*.

2. Ekstraksi Karakter

Proses ekstraksi karakter meliputi beberapa tahap yaitu:

a. Pendeteksian garis pemisah dan teks

Proses ini membedakan garis pemisah (line break) dan karakter pada input, dengan tujuan menentukan posisi karakter pada baris-baris teks output dan mengambil satu baris teks untuk diproses pada langkah selanjutnya. Caranya adalah dengan menghitung jumlah piksel hitam yang terdapat pada sebaris piksel *horizontal*. Beberapa baris piksel yang berdekatan dan sama-sama memiliki jumlah piksel minimum akan dianggap sebuah garis pemisah. *Derau* mungkin akan mengacaukan proses ini, namun dapat diatasi dengan pemberian nilai toleransi jumlah piksel hitam *minimum*.

b. Pendeteksian dan pengisolasian komponen karakter

Secara prinsip, proses ini mendeteksi secara individual setiap karakter dalam sebuah baris teks, kemudian mengisolasi tiap-tiap karakter menjadi sebuah bagian yang berbentuk segi empat. Cara yang paling sederhana adalah menerapkan cara untuk mendeteksi garis pemisah, namun dilakukan secara vertikal, mulai dari ujung paling kiri sebuah baris teks. Setelah terbagi dalam tiap-tiap karakter, piksel putih yang *redundant* dihilangkan sehingga terbentuk isolasi berbentuk segi empat yang melekat pada ujung-ujung karakter.

3. Ekstraksi Properti Karakter

Setelah proses *isolasi* dilakukan, maka dilakukan *property extraction* atau perhitungan nilai-nilai properti yang melekat pada sebuah karakter. Secara keseluruhan terdapat 29 sifat atau properti yang didapatkan dari setiap karakter. Langkah pertama, karakter dibagi menjadi dua puluh lima segmen yang sama besar, dengan penentuan posisi segmen:

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

Kemudian dihitung tiap-tiap properti dengan rincian sebagai berikut: Properti 0-24 adalah nilai rasio jumlah piksel hitam dan jumlah piksel putih pada tiap segmen karakter dalam skala keabuan (grayscale). Sebagai contoh segmen kiri atas (0) terdiri dari 4 piksel putih dan 5 piksel hitam, maka properti $0 = (255+255+255+255+0+0+0+0)/9 = 113,333 = 113$. Nilai pecahan yang didapat dibulatkan; Properti 25 adalah rasio jumlah piksel hitam dan jumlah piksel putih dalam skala keabuan, untuk separuh bagian atas karakter; Properti 26 adalah rasio jumlah piksel hitam dan jumlah piksel putih dalam skala keabuan, untuk separuh bagian bawah karakter; Properti 27 adalah rasio lebar dan tinggi karakter dalam skala 0-255. V; dan Properti 28 adalah indikator karakter yang terpisah secara vertikal seperti i dan j. Jika tidak terpisah, indikator bernilai 0. Jika terpisah bernilai 255. Rangkaian kedua puluh sembilan properti di atas disebut *property set*.

Perbandingan Karakter *Input* dan *Learning Characters*. Seperti telah dikemukakan di awal, sebelum menerima input, algoritma ini terlebih dahulu mengenali karakterkarakter yang terdapat *learning set*. Proses yang dijalankan untuk mengenali *learning characters* sama dengan pengenalan karakter input, sehingga akan dihasilkan pula sebuah *property set* untuk setiap *learning character*.

Dengan demikian pada tahap ini dilakukan perbandingan nilai masing-masing properti antara sebuah karakter input dengan setiap karakter yang tersimpan dalam *learning set*.

Perhitungan yang dipakai dalam perbandingan ini didefinisikan dalam *pseudo-code* yang terdapat pada Gambar 1.

```
Distance Component::distance(Component * comp)
{
    Property * a = fproperty;
    Property * b = comp->properties();
    Distance dist=0;
    int dif=0;

    for(int i= 0; i < numProperties; i++)
    {
        dif = a[i] - b[i];
        dist += dif * dif;
    }
    return dist;}

```

Gambar 1. Pseudocode OCRChie

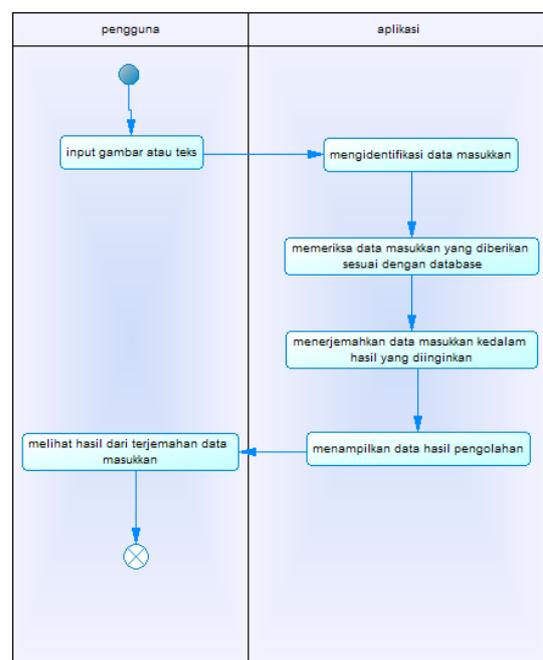
Untuk perancangan analisis ini digunakan model *waterfall* untuk menganalisis aplikasi yang akan dirancang. Adapun tahapan-tahapannya adalah: (1) *Requirements Definition*. Tahap ini, analisis kebutuhan perangkat lunak dan kebutuhan adalah data nilai penjualan perbulan, data bulan, data tahun, dan data produk; (2.) *System and Software Design*.

Tahapan *design* ini dimodelkan melalui flowchart untuk keseluruhan sistem. Untuk basis datanya menggunakan *Entity Relationship Diagram*; (3) *Implementation and Unit Testing*. Perancangan aplikasi menggunakan bahasa pemrograman Visual Basic dan basis datanya menggunakan My SQL; (4) *Integration and System Testing*. Pengujian aplikasi ini menggunakan metode *black box*; dan (5) *Operation and Maintenance*. Setelah aplikasi sudah dibuat, maka *maintenance* akan dilakukan jika aplikasi yang dibuat terdapat *bug-bug* yang akan mengganggu jalannya aplikasi.

Bagan alir atau *flowchart* digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *Flowchart* sistem ditunjukkan pada gambar 2.



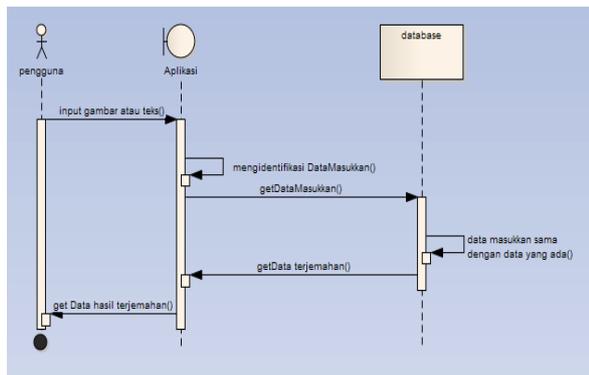
Gambar 2. Flowchart sistem



Gambar 3. Activity diagram aplikasi

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Activity Diagram digambarkan pada Gambar 3.

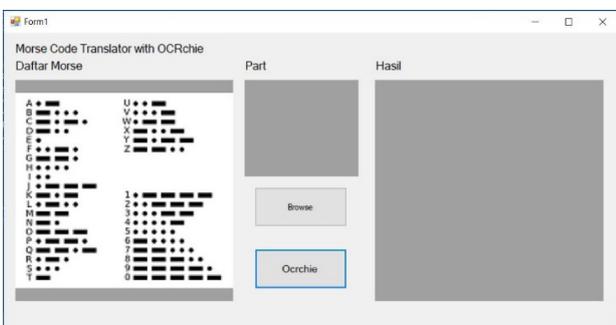
Cara kerja suatu waktu dari mulai saat pengguna mengirimkan atau memasukkan data hingga mendapatkan hasil terjemahannya dapat kita lihat pada Sequence Diagram berikut yang terdapat pada Gambar 4.



Gambar 4. Sequence diagram

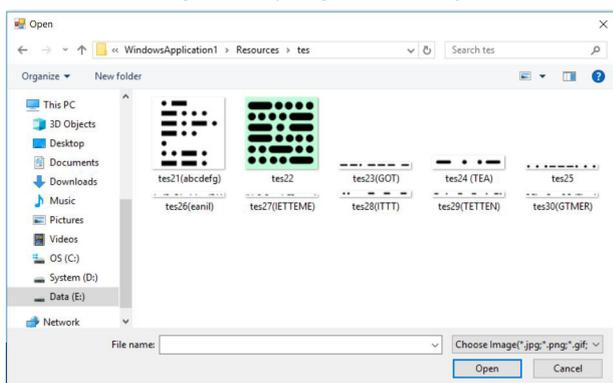
III. HASIL DAN PEMBAHASAN

Tampilan gambar 5 adalah tampilan halaman utama ketika aplikasi dijalankan. Tampak tombol untuk browse gambar dan *OCRchie*.



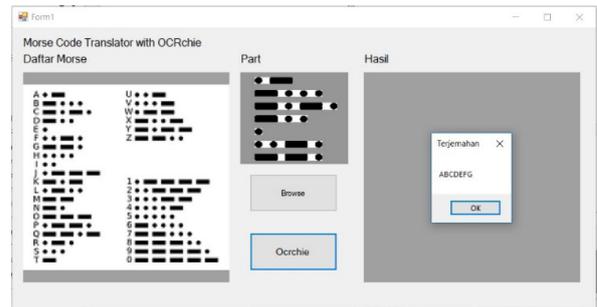
Gambar 5. Tampilan halaman utama

Tampilan gambar 6 merupakan tampilan apabila user memilih tombol *browse*, akan tampil window untuk memilih gambar yang akan diterjemah.



Gambar 6. Tampilan browse

Tampilan gambar 7 merupakan tampilan apabila user memilih tombol *OCRchie* ketika selesai memilih gambar yang akan diproses. Ketika proses selesai maka akan tampil hasil dari terjemahan.



Gambar 7. Tampilan hasil OCRchie

Hasil pengujian berupa penerjemahan gambar kode morse dengan *OCRchie* menggunakan metode *Euclidean Distance* dapat dilihat pada Tabel 1.

Tabel 1. Hasil penerjemahan

No	Gambar	Manual	Aplikasi	Hasil (Sesuai/Tidak)
1			B	Tidak
2		Sister	sis5e5	Tidak (Hampir) 4/6
3		hijkl	xmw9l	Tidak
4		USD	159	Tidak
5		nopqrstuvwxyz	Lop7rs8u5jxh	Tidak (Hampir) 6/11
6		Tan fins	5h	Tidak
7		Ant	5	Tidak
8		10	55	Tidak
9		Duru	Duru	Sesuai
10		Iwi	X	Tidak
11		Sos	F	Tidak
12		Whw	55h	Tidak

Dari hasil pengujian maka dapat dihitung tingkat akurasi yang dapat dilihat pada Tabel 2.

Tabel 2. Tingkat akurasi

Data Uji		
Input	Sesuai	Tidak Sesuai
Gambar	3	27
Akurasi (%)	$= \frac{\text{jumlah citra uji yang sesuai}}{\text{total citra uji}} \times 100\%$ $= \frac{3}{30} \times 100\%$ $= 10\%$	

Dari Tabel 1 dapat disimpulkan bahwa tingkat akurasi algoritma *OCRchie* dengan menggunakan metode *Euclidean Distance* pada klasifikasi penerjemahan *morse code* sebesar 10% dikarenakan jarak dan ketebalan pada setiap gambar berbeda.

IV. SIMPULAN

Dari penelitian dan pengujian yang dilakukan maka dapat disimpulkan sebagai berikut: (1) Gambar *morse code* yang diterjemahkan, dapat diterjemahkan dengan baik jika gambar mempunyai ketebalan dan jarak yang sama berdasarkan aplikasi; (2) Dalam mengekstraksi nilai fitur setiap karakter, nilai tersebut sebagai nilai ambang (*threshold*) untuk ketepatan akurasi penerjemahan; dan (3) Dari hasil pengujian, dapat diketahui bahwa algoritma *OCRchie* dan *Euclidean Distance* bukanlah algoritma dan metode yang tepat untuk menerjemahkan *morse code* berdasarkan gambar dengan tingkat akurasi sebesar 10%. Namun persentase dapat berubah berdasarkan pengembangan algoritma yang dilakukan berikutnya.

V. DAFTAR RUJUKAN

[1] Z. Niswati, Pengenalan Pola Tekstur Brodatz dengan Metode Jarak Euclidean. Jurnal Faktor Exacta. Volume 5. Nomor 3. ISSN: 1979-276X. hlm 243-253, 2013.

[2] E. S. N. Aisyah, A. Hayat, P. Widanti, Prasetya, Y. Shinta, dan H. Iskandar. Analisis Kemiripan Pola Citra Digital Menggunakan Metode Euclidean. STMIK AMIKOM Yogyakarta, ISSN: 2302-3805. hlm 62, 2015.

[3] L. C. Peter. Morse Code: The Essential Language, radio amateur's library, issue 69, American Radio Relay League, ISBN: 0-87259-035-6. hlm 35, 1986.

[4] S. Fauziyah., U. Sunarya & R. D. Atmaja, Identifikasi Kode Jari Tangan Pada Sistem Operasi Android dengan Metode Eulidean Distance untuk Sistem Kunci Pintu. Jurnal Elektro dan Telekomunikasi Terapan 2014. Volume 1. Nomor 1. ISSN: 2442-4404. hlm 42, 2016.

[5] R. Wulanningrum & A. Rachmad. Pengenalan Rumput Laut Menggunakan Euclidean Distance Berbasis Eksktraksi Fitur. Seminar Nasional Aplikasi Teknologi Informasi, Yogyakarta, ISSN: 1907-5022. hlm 27, 2012.

[6] T. Juliatmojo & E. Aribowo. Pembelajaran Sandi Morse Dan Sandi Semaphore Dalam Bentuk Simulasi Berbasis Multimedia. Jurnal Sarjana Teknik Informatika, Vol.1, No.1, E-ISSN: 2338-5197. hlm 51, 2013.

[7] S. Xuan & Y. D. Zhong. Design of Morse-code Decoder with Filtering and Fault Correction Function Based on MATLAB. International Journal of Signal Processing, Image Processing and Pattern Recognition, Volume 8, Nomor 10, ISSN: 2005-4254. hlm 73, 2015.