# Kalkulator Visual Pada Tulisan Tangan Memanfaatkan Pengenalan Pola Berbasis Aturan Dengan Levenshtein Distance untuk Menghasilkan Informasi Ucapan

**Taufik Ridwan[1], Muhammad Indra NS[2]**
[1,2]Informatika, Sekolah Teknik Eleketro dan Informatika, ITB
[1]23515026@std.stei.itb.ac.id, [2]23515019@std.stei.itb.ac.id

***Abstrak*-Ketika diberikan tulisan dalam representasi citra, operasi dasar matematika sebenarnya mudah untuk dilakukan oleh manusia karena manusia telah memiliki pengetahuan untuk dapat melakukannya. Namun tidak begitu halnya dengan komputer. Diperlukan proses untuk menirukan proses pengenalan tulisan tangan yang dilakukan manusia.Praproses, Ekstraksi fitur, serta perhitungan jarak dengan Levenshtein adalah proses-proses yang didahului.**

***Kata kunci*- chaincode, citra, handwriting. lavenshtein**

## I. PENDAHULUAN

*Artificiall Intelligent* pada dasarnya merupakan usaha untuk menirukan perilaku manusia oleh komputer. Untuk dapat melakukan hal yang cerdas tersebut komputer harus memiliki knowledge dan kemampuan penalaran sehingga mampu untuk bertindak seperti manusia. Salah satu bagian dari Artificiall Intelligent adalah pengenalan pola. Beberapa metode yang sering digunakan diantaranya adalah berbasis statistik ataupun berbasis aturan.

Putra dan Supriana mengusulkan pendekatan berbasis aturan untuk memproses tulisan tangan tanpa langkah normalisasi dengan menghitung jarak menggunakan Levenshtein [1]

Penggunaan smartphone dapat dimanfaatkan untuk menunjang dan membantu kehidupan manusia dalam berbagai bidang. Salah satunya dapat digunakan untuk pembelajaran matematika dengan cara memproses citra (*image processing*)untuk mengenali operasi dasar matematika yang ditulis tangan (handwriting). Sebuah sistem yang diusulkan akan mampu untuk melakukan perhitungan matematika dari citra yang ambil kamera atau yang tersimpan di tempat penyimpanan. Sistem kemudian mengeluarkan output berupa teks dan hasil pengenalannya.

Penelitian ini bermaksud untuk merancang sebuah sistem yang dapat mengenali operasi dasar matematika pada tulisan tangan dengan keluaran berupa suara dengan memanfaatkan pattern recognition pada sistem operasi android..

## II. PENGENALAN POLA PADA CITRA TULISAN TANGAN

Secara harafiah, citra adalah gambar pada bidang dua dimensi[2]. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Salah satu bidang komputer yang memanfaatkan citra adalah pengenalan pola (*pattern recognition*). Pengenalan pola adalah proses untuk mengenali dan mengidentifikasikan simbol atau citra digital.

Untuk dapat melakukan proses pengenalan pola, maka perlu dilakukan beberapa langkah diantaranya

### A. PRAPROSES

1. Greyscale

   Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, artinya nilai dari Red = Green = Blue. Nilai-nilai tersebut digunakan untuk menunjukkan intensitas warna. Apabila suatu citra direpresentasikan dalam 8 bit maka berarti pada citra terdapat 28 atau 256 level grayscale, (biasanya bernilai antara 0 – 255), dimana nilai 0 menunjukkan level intensitas paling gelap dan nilai 255 menunjukkan intensitas paling terang.

2. Biner

   Citra menyimpan begitu banyak informasi, namun terkadang informasi yang terlalu banyak itu dapat dipilah-pilah untuk hanya menggunakan informasi mendasarnya saja. Salah satu pendekatan yang digunakan adalah melakukan binerisasi pada citra. Citra biner hanya mempunyai dua nilai derajat keabuan: hitam dan putih (1 dan 0). Berikut adalah pseudo code untuk merubah citra menjadi biner.

```
1   rgb = citra.getRGB(posisiX, posisiY);
2   r = (rgb & 0x00ff0000) >> 16; // 0xff
3   g = (rgb & 0x0000ff00) >> 8;
4   b = rgb & 0x000000ff;
5
6   int avg = (r + g + b) / 3;
7   if (avg < threshold)
8           return 1;
9   else
10  return 0;
```

Gambar 1. Contoh Pseudo code merubah citra ke biner

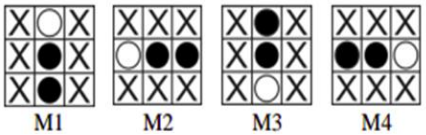
3. Thinning

   Penggerusan atau thinning terhadap image biner sehingga karakter yang diamati menjadi lebih ramping dan didapatkan bentuk struktur tulang (skeletonizing). Dari hasil yang berbentuk struktur tulang dari citra diharapkan pada proses berikutnya akan lebih mudah untuk menginterpretasikan dari sebuah karakter

dikarenakan struktur tulang dianggap dapat mewakili bentuk dasar dari sebuah karakter.

Penipisan pola merupakan proses yang iteratif yang menghilangkan pixel-pixel hitam (mengubahnya menjadi pixel putih) pada tepi-tepi pola. Jadi, algoritma penipisan mengelupas pixel-pixel pinggir objek, yaitu pixel-pixel yang terdapat pada peralihan 0→1.

Salah satu pendekatan yang digunakan untuk thinning adalah Stentiford. Algoritma Stentiford menggunakan template berukuran 3 x 3 untuk melakukan pencocokan dengan citra. Ketika kecocokan terjadi, titik tengah dari piksel 3 x 3 tersebut akan dihapus (diubah warnanya ke warna putih).




Gambar 2. Template thinning dengan Stentiford

Algoritma stentiford dapat dijabarkan sebagai berikut :

1. Cari lokasi piksel (i,j) dimana piksel yang ada di citra I, cocok dengan template M1.
2. Jika titik tengah piksel bukan merupakan endpoint (titik akhir) dan mempunyai indeks konektivitas = 1, maka tandai piksel ini untuk penghapusan.
3. Ulangi langkah 1 – 2 untuk semua lokasi piksel yang cocok dengan template M1.
4. Ulangi langkah 1 – 3 untuk sisa template yang belum dicocokkan : M2, M3, M4.
5. Jika ada piksel yang telah ditandai sebelumnya untuk proses penghapusan, maka hapuskan piksel tersebut dengan mengubahnya menjadi warna putih.
6. Jika masih ada piksel yang dapat dihapus pada langkah ke 5, maka ulangi semua proses mulai dari langkah ke – 1, jika tidak berhenti

Gambar 3. Contoh Proses *Thinning*

4. Segmentasi

Proses selanjutnya yang dilakukan adalah melakukan segmentasi pada objek. Proses segmentasi bertujuan mengelompokkan pixel-pixel objek menjadi wilayah (region) yang merepresentasikan objek. Ada dua pendekatan yang digunakan dalam segmentasi objek:

1. Segmentasi berdasarkan batas wilayah (tepi dari objek). Pixel-pixel tepi ditelusuri sehingga rangkaian pixel yang menjadi batas (boundary) antara objek dengan latar belakang dapat diketahui secara keseluruhan (algoritma boundary following).
2. Segmentasi ke bentuk-bentuk dasar (misalnya segmentasi huruf menjadi garis-garis vertikal dan horizontal, segmentasi objek menjadi bentuk lingkaran, elips, dan sebagainya)

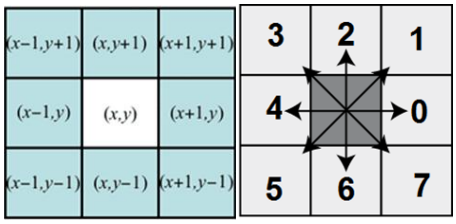
B. EKSTRASI FITUR

Ektraksi fitur adalah tahapan untuk mendapatkan ciri dari data yang akan diambil. salah satu langkah yang bisa dilakukan adalah dengan chain code (kode belok). Chain code merupakan suatu atribut unik yang dapat diekstraksi dari suatu objek di dalam gambar dengan cara menyusuri pixel batas objek tersebut berdasarkan arah-arah tertentu yang telah ditetapkan. Chain code pertama kali diusulkan oleh Freeman pada tahun 1961[3]. Keluaran dari pengambilan chain code adalah angka-angka yang menunjukkan arah yang merepresentasikan batas objek. Pencarian dilakukan dengan cara menyusuri batas objek, dimulai dari pixel batas objek yang pertama ditemukan sampai kembali lagi ke pixel tersebut

Dimisalkan *b* adalah *pixel* batas objek yang sedang dikunjungi, *w* adalah *pixel* tetangga, *dir* adalah variabel yang menyimpan arah dari *b* ke *w*, dan *C* adalah *array* yang akan menyimpan arah-arah penelusuran, maka penelusuran dilakukan sebagai berikut:

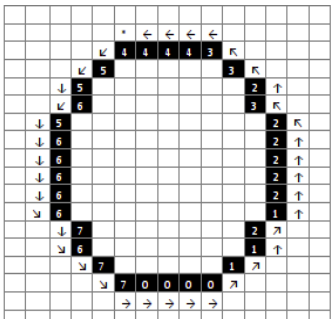
1. Set *b* sama dengan *pixel* batas objek yang sedang dikungjungi dan *w* sama dengan *pixel* nonobjek tepat di sebelah kiri *b*. Jika objek baru pertama kali ditemukan, maka *b* adalah *starting pixel*.
2. Tandai *starting pixel*.
3. Set *dir* sama dengan arah dari *b* ke *w* berdasarkan ketetanggaan (gambar 3).
4. Telusuri setiap tetangga *b* dimulai dari arah yang tercatat pada *dir*.
   a) Jika *pixel* pada arah *dir* merupakan *pixel* nonobjek, maka perbarui nilai *dir* ke *dir* + 1 mod 8 (arah selanjutnya), kemudian perbarui *w* sesuai dengan nilai *dir* yang baru.
   b) Jika pada arah *dir* ditemukan *pixel* yang merupakan elemen batas objek, maka masukkan *dir* ke dalam *C* dan perbarui *b* sebagai *pixel* batas objek yang akan selanjutnya dikunjungi.

   Langkah a dilakukan sampai akhirnya ditemukan *pixel* tetangga yang merupakan elemen batas. *Pixel* yang tercatat pada *w* tepat sebelum ditemukan elemen batas merupakan *pixel* yang akan dirunut balik dari *b* yang baru
5. Ulangi penelusuran dari langkah 2.
6. Penelusuran dihentikan apabila *starting pixel* dikunjungi untuk kedua kalinya.




Gambar 4. Ketetanggaan *pixel* menggunakan *8-connectivity*

Pada Gambar 4, pixel di tengah adalah b. Pencarian elemen batas dilakukan dengan mengitari tetangga b secara anti-clockwise berdasarkan urutan angka (a). Setiap arah tetangga ditunjukkan dengan nomer indeks yang diturunkan dari indeks b yang ditunjukkan dengan (x,y) (b). Ilustrasi keluaran algoritma *chain code* digambarkan pada gambar 5.



Gambar 5. Ilustrasi chain code huruf O

Pada Gambar 5 arah penelusuran berlawanan dengan arah jarum jam. Angka di dalam pixel hitam menunjukkan arah dari pixel sebelumnya. Panah menunjukkan arah penelusuran. Gambar tersebut akan menghasilkan chain code 5565666667677000011212222232334444

| | | k | i | t | t | e | n |
|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| s | 1 | 1 | 2 | 3 | 4 | 5 | 6 |
| i | 2 | 2 | 1 | 2 | 3 | 4 | 5 |
| t | 3 | 3 | 2 | 1 | 2 | 3 | 4 |
| t | 4 | 4 | 3 | 2 | 1 | 2 | 3 |
| i | 5 | 5 | 4 | 3 | 2 | 2 | 3 |
| n | 6 | 6 | 5 | 4 | 3 | 3 | 2 |
| g | 7 | 7 | 6 | 5 | 4 | 4 | 3 |

| | | S | a | t | u | r | d | a | y |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| S | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| u | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 5 | 6 |
| n | 3 | 2 | 2 | 2 | 3 | 3 | 4 | 5 | 6 |
| d | 4 | 3 | 3 | 3 | 3 | 4 | 3 | 4 | 5 |
| a | 5 | 4 | 3 | 4 | 4 | 4 | 4 | 3 | 4 |
| y | 6 | 5 | 4 | 4 | 5 | 5 | 5 | 4 | 3 |

Gambar 6. Contoh perbandingan string dengan Levenshtein

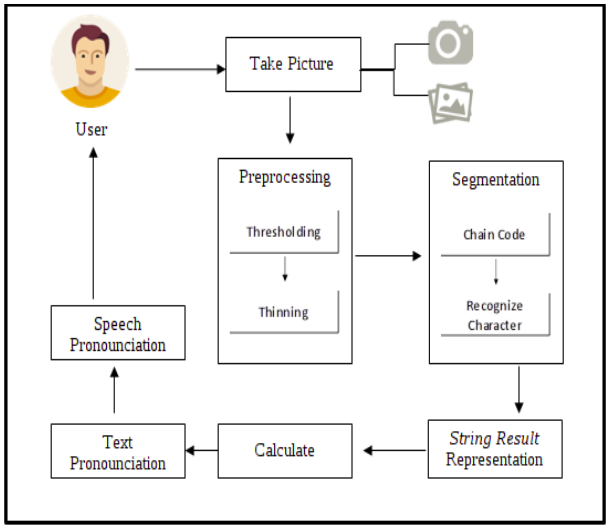
## C. PENGENALAN

Tahap pengenalan atau *recognize* merupaka tahapan yang digunakan untuk mengenali citra berdasarkan fitur yang dimilikinya. Ada beberapa pendekatan diantara nya berbasis statistik (*machine learning)* ataupun berbasis aturan (*rule based*).

Cara untuk Pengenalan dengan basis aturan salah satunya adalah dengan membandingkan jarak antara *template* dengan data yang diujikan. salah satunya adalah dengan algoritma Levenshtein distance. Levenshtein bekerja dengan cara mengukur jarak dua buah string dengan menggunakan prinsip dynamic programming. Contoh perbandingan string dengan algoritma Lavenstein dapat di lihat pada gambar 6 Operasi yang dilakukan meliputi insertion, substitution and deletion.

## III. METODE

Arsitektur sistem pada penelitian ini dapat dilihat pada Gambar 7 di bawah ini,




Gambar 7. Arsitektur sistem

Proses yang dilalui untuk dapat menghasilkan sistem ini meliputi:

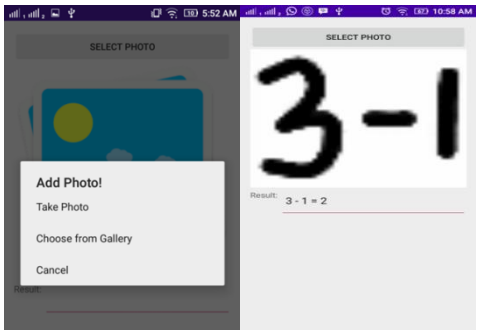
1. Pengambilan gambar yang dapat diambil baik dari file yang sudah ada di *smartphone* atau pun dengan cara membuka fitur camera
2. Preprocessing
   1. Thresholding
      Thresholding atau pengambangan merupakan tahapan untuk merubah citra sesuai ambang tertentu. Pada penelitian ini diusulkan thresholding dilakukan untuk membuat citra menjadi greyscale untuk selanjutnya diubah menjadi representasi biner.
   2. Thinning
      Penipisan atau *thinning* digunakan untuk mendapatkan citra tulisan tangan setebal satu pixel. Hal ini juga dimaksudkan untuk mendapatkan fitur yang lebih representatif dan waktu komputasi yang lebih baik. Metode stentiford dipilih pada proses thinning yang dilakukan.
3. Segmentasi untuk membagi citra menjadi per bagian yang akan dikenali. Proses yang dilalui adalah melakukan pemindaian citra secara vertikal yang bergerak horizontal(ke kanan). setiap segmen kemudian akan disimpan dalam sturktur data tertentu.
4. Chain Code digunakan untuk mendapatkan kode penelusuran dari citra. Kemudian untuk mendapatkan hasil yang lebih baik dilakukan proses tambahan yaitu menyeragamkan kode

yang sama menjadi satu karakter. Misal “11111” menjadi “1”. Hal ini digunakan untuk lebih memberikan hasil maksimal pada setiap karakter yang dikenali.

5. Pengenalan yang digunakan dengan metode levenshtein distance untuk mengukur kecocokan data uji dengan rule yang telah dibuat
6. Kalkulasi perhitungan karakter yang telah dikenali
7. Text to Sound digunakan untuk merubah teks pengenalan menjadi suara ucapan yang disimpan.

## IV. HASIL DAN PENGUJIAN

Hasil percobaan dengan menggunakan 8 template sebagai template training didapati hasil seperti dibawah ini dalam mengklasifikasikan citra.




Gambar 8. Hasil Pengujian

Dari pengujian yang telah dilakukan, Jika jumlah template diperbanyak justru tidak akan menjaminhasil pengenalan menjadi lebih baik. Hal ini diakibatkan beberapa jenis tulisan justru memiliki kemiripan dari segi *chain code* dengan tulisan yang bukan seharusnya.

## V. KESIMPULAN DAN SARAN

Sistem yang dirancang sudah dapat melakukan perhitungan dengan memanfaatkan informasi visual yang diberikan. Namun pada beberapa kasus akibat adanya noise hasil pengenalan menjadi kurang maksimal. Untuk itu diharapkan memberikaninput berupa citra yang tidak mengandung noise. Kemudian pada beberapa kasus pengenalan terdapat kesalahan baik karena tulisan tidak jelas ataupun karena masalah pada pengenalan.

Adapun saran untuk penelitian selanjutnya adalah:

1. Lakukan normalisasi data uji sesuai dengan data citra yang dijadikan template
2. Perbaiki chain code untuk lebih meningkatkan akurasi
3. Dicari metode lain untuk tahap pengenalan selin menggunakan Levenshtein distance
4. Bandingkan pendektan aturan dengan pendektan berbasis statistical (*machine learning*)
5. Pergunakan template yang lebih representatif.

## VI. REFERENSI

[1] Made Edwin Wira Putra, dan Iping Supriana SuwardiKadir, A. (2015). Structural off-line handwriting character recognition using approximate subgraph matching and levenshtein distance. *International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)*

[2] Rinaldi Munir. (2003). Diktat kuliah Pengolahan Citra,. Edisi Kedua. Departemen Teknik Informatika ITB.

[3] Annapurna, P., Kothuri, S., Lukka, S. (2013). Digit Recognition Using Freeman Chain Code. International *Journal of Application or Innovation in Engineering and Management, 2(8), 362*.