Pengenalan Angka Tulisan Tangan dengan Menggunakan Local Binary Pattern Variance dan Klasifikasi K-Nearest Neighbour

Handwriting Digit Recognition with Use Local Binary Pattern Variance and K-Nearest Neighbour Classification

Nurul Ilmi[1]
[1]Fakultas Informatika, Universitas Telkom, Bandung (ilmynurull@gmail.com)

Abstrak - Dalam penelitian Tugas Akhir ini dijelaskan mengenai sistem pengenalan angka tulisan tangan dengan menggunakan metode ekstraksi ciri *Local Binary Pattern* Variance dan Klasifikasi *K-Nearest Neighbour*. Sistem ini diterapkan pada citra angka tulisan tangan pada formulir C1 Komisi Pemilihan Umum (KPU) untuk mempermudah petugas KPU dalam proses penginputan data hasil Pemilu ke sebuah database computer. Sistem ini terdiri dari beberapa tahap, yaitu pengumpulan data, *pre-processing*, ekstraksi ciri dan klasifikasi. Pengujian pada sistem pengenalan angka tulisan tangan ini menunjukkan bahwa metode Local Binary Pattern Variance dapat mengenali karakter tulisan tangan pada MNIST dengan akurasi 89,81% dengan menggunakan parameter terbaik radius 4, 256 dan 64 bin histogram, pembagian 9 region pada citra dan mengambil 10 tetangga terdekat pada tahap klasifikasi K-NN. Namun, pada saat melakukan pengenalan pada data uji dari formulir C1 akurasi yang dihasilkan menurun, yaitu sebesar 70,9091% dengan menggunakan data latih C1

Kata kunci : Pengenalan angka tulisan tangan, *Local Binary Pattern Variance* (LBPV), *K-Nearest Neighbour*, formulir C1 KPU.

I. Pendahuluan

Formulir C1 Komisi Pemilihan Umum (KPU) merupakan dokumen hasil perolehan suara Pemilu di setiap Tempat Pemungutan Suara (TPS). Mekanisme perhitungan manual yang dilakukan oleh petugas dengan memasukkan data pada sebuah formulir C1 dan melakukan tabulasi data memakan waktu yang cukup lama dan kurang akurat. Proses tabulasi data manual yang biasa dilakukan oleh petugas dapat dimudahkan dengan menggunakan sistem pengenalan angka tulisan tangan dengan mengenali angka hasil perhitungan suara yang ada pada formulir C1 KPU.

Optical Character Recognition (OCR) merupakan sarana yang digunakan untuk mengenali karakter optis untuk membaca informasi ke dalam sistem computer. Identifikasi karakter ini dilakukan dengan menggunakan alat pemindai terlebih dahulu agar gambar dapat diproses secara *offline*. sistem pengenalan karakter tulisan tangan ini dapat sangat bermanfaat untuk membantu pekerjaan manusia, misalnya saja pada kasus perhitungan suara Pemilihan Umum (Pemilu) ini.

Dalam tugas akhir ini, sistem pengenalan angka tulisan tangan dibuat dengan menggunakan *Local Binary Pattern Variance* (LBPV) untuk ekstraksi ciri dan *K-Nearest Neighbour* (K-NN) untuk klasifikasi. Pada penelitian [7] menunjukkan metode *Local Binary Pattern Variance* menunjukkan bahwa metode LBP *Variance* mampu mengenali gambar tekstur dengan akurasi yang tinggi. LBP *Variance* (LBPV) diusulkan untuk mengkarakterisasi informasi kontras lokal ke dalam satu LBP dimensi histogram. Untuk proses klasifikasi, klasifikasi *K–Nearest Nighbours* sangat cocok digunakan untuk pengklasifikasian data yang besar [11].

II. Citra Digital

Sebuah citra digital dapat mewakili sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel, yaitu elemen terkecil dari sebuah citra. Piksel mempunya dua parameter, yaitu koordinat dan intensitas. Nilai yang terdapat pada koordinat (x,y) adalah f(x,y), yaitu besar intensitas citra atau tingkat keabuan dari piksel di titik itu.

Jenis citra berdasarkan nilai *level* bit dari masing – masing piksel diantaranya terdiri dari citra biner, *grayscale* dan citra warna (RGB). Citra biner disebut juga *monochrome*, merupakan citra digital yang setiap pikselnya hanya memiliki 2 kemungkinan derajat keabuan, yaitu 0 dan 1. Nilai 0 mewakili warna hitam dan 1 mewakili warna putih. Citra *grayscale* disebut juga citra aras keabuan yang memiliki warna abu – abu diantara warna minimum (hitam) dan warna maksimum (putih). Citra dengan skala keabuan 4-bit memiliki 2^4 = 16 kemungkinan warna, yaitu 0 (minimal) hingga 15 (maksimal). Sementara citra digital dengan skala keabuan 8-bit memiliki 2^8 = 256 kemungkinan warna, yaitu 0 (minimal) hingga 255 (maksimal). Pada citra warna, setiap piksel merupakan kombinasi dari tiga warna dasar merah, hijau, dan biru, sehingga warna ini disebut juga citra

RGB (*Red Green Blue*). Setiap komponen warna memiliki intensitas tersendiri dengan nilai minimum 0 dan nilai maksimum 255 (8-bit).[6]

III. Local Binary Pattern

Local Binary Pattern (LBP) adalah operator yang sederhana namun sangat efisien dimana label piksel dari suatu gambar dilakukan *thresholding* pada ketetanggaan masing-masing piksel dan menetapkan hasilnya sebagai bilangan biner. Properti yang paling penting dari operator LBP dalam aplikasi dunia nyata adalah *invariant* terhadap perubahan tingkat keabuan yang monoton, misalnya, dari variasi pencahayaan. Hal lain yang tidak kalah penting adalah kesederhanaan komputasi, yang memungkinkan untuk menganalisis gambar dalam perubahan pengaturan *real-time* [4].

LBP menggunakan 8 piksel dalam sebuah blok piksel 3x3, formulasi umum operator ini tidak memiliki batasan untuk ukuran ketetanggaannya atau jumlah sampel poin. Dari gambar monokrom I(x,y), g_C dilambangkan sebagai tingkat keabuan dari piksel acak (x,y), yaitu g_C = I (x,y). Selain itu, g_p dilambangkan sebagai nilai keabuan dari sampel poin dalam lingkungan ketetanggaan P *sample point* dan *radius* R di sekitar titik (x,y):

$$g_p = \mathrm{I}\,(x_P, y_p), \quad \mathrm{p} = 0, \ldots, \mathrm{P} - 1 \text{ dan} \tag{1}$$

$$x_P = \mathrm{x} + \mathrm{R}\cos(2\pi \mathrm{p}/\mathrm{P}) \tag{2}$$

$$y_P = \mathrm{y} + \mathrm{R}\cos(2\pi \mathrm{p}/\mathrm{P}) \tag{3}$$

Rumus Operator umum LBP yang telah ditetapkan [4] :

$$LBP_{P,R}(x,y) = \sum_{i=0}^{p-1} s(g_p - g_c)2^i \tag{4}$$

$$s(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{5}$$

Dengan s(x) adalah fungsi threshold dan i adalah titik ketetanggaan.

Vektor ciri hasil LBP umumnya direpresentasikan dengan histogram. Misalkan gambar tekstur NxM. Setelah mengidentifikasi pola LBP setiap piksel (i,j), gambar tekstur keseluruhan diwakili dengan membangun histogram:

$$H(k) = \sum_{i=1}^{N}\sum_{j=1}^{M} f\big(LBP_{P,R}(i,j), k\big), k \in [0, K] \tag{6}$$

$$f(x) = \begin{cases} 1, & x = y \\ 0, & yang\ lainnya \end{cases} \tag{7}$$

Histogtam ini merepresentasikan hasil dari *Local Binary Pattern* pada setiap piksel i,j. K adalah nilai maksimum dari pola LBP.

IV. Rotational Invariant Local Contrast

Rotasi kontras invarian lokal dapat dihitung dalam sebuah tetangga simetrik *circular* seperti halnya LBP [12] dengan rumus:

$$VAR_{P,R} = \frac{1}{P}\sum_{P=0}^{P-1}\big(g_p - \mu\big)^2 \tag{8}$$

$$\mu = \frac{1}{P}\sum_{P=0}^{P-1} g_p$$

Dimana,g_p adalah nilai pada titik ketetanggaan dari *sample point* dalam lingkungan ketetanggaan P *sample point* dan *radius* R di sekitar titik (x,y) pada matriks hasil LBP dan μ adalah nilai rata – rata titik ketetanggaan dari P *sample point* pada matriks hasil LBP.

$VAR_{P,R}$ menurut definisi, invarian terhadap pergesaran dalam *grayscale*. Karena kontras diukur secara lokal, pengukuran dapat dihitung secara lokal, perhitungan dapat bertahan asalkan variasi pencahayaan antar gambar selama perbedaan nilai mutlak *gray* tidak banyak terpengaruh. Deskripsi rotasi invarian tekstur dalam hal pola tekstur dan kekuatan mereka diperoleh dengan distribusi gabungan dari LBP dan varian lokal,

dinotasikan $LBP_{P1,R1}^{riu2}/VAR_{P2,R2}$. Biasanya, paremeter lingkungan dipilih, sehingga P1 = P2 dan R1 = R2, meskipun tidak ada yang mencegah satu dari memilih nilai yang berbeda.

V. Local Binary Pattern Variance

LBPV adalah sebuah metode yang sederhana tapi efisien mempelajari gabungan LBP dan metode distribusi kontras. Seperti yang dapat dilihat pada persamaan (2.18) perhitungan LBP histogram H tidak melibatkan informasi variansi $VAR_{P,R}$. Itu artinya, tidak peduli apa variansi LBP dari wilayah lokal, histogram perhitungan memberikan bobot yang sama, 1 untuk setiap pola LBP. Sebenarnya, variansi berkaitan dengan ciri tekstur. Biasanya daerah tekstur frekuensi tinggi akan memiliki varians yang lebih tinggi dan kontribusi mereka lebih kepada diskriminasi gambar tekstur. LBPV histogram dihitung dengan rumus [12]:

$$LBPV_{P,R}(k) = \sum_{i=1}^{N}\sum_{j=1}^{M} w\big(LBP_{P,R}(\mathrm{i,j}),\mathrm{k}\big), \mathrm{k} \in [0,\mathrm{k}] \tag{9}$$

$$w\big(LBP_{P,R}(\mathrm{i,j}),\mathrm{k}\big) = \begin{cases} \big(VAR_{P,R}(\mathrm{i,j})\big) & \big(LBP_{P,R}(\mathrm{i,j}) = k\big) \\ 0 & x \leq 0 \end{cases} \tag{10}$$

Jika kita melihat LBP dan VAR sebagai dua sumbu ortogonal dalam sistem koordinat, LBPV bisa dianggap sebagai integral proyeksi sepanjang sumbu VAR. Ini akan menjadi representasi yang disederhanakan dari distribusi 2D LBP/VAR. Karena $LBP_{P,R}^{riu2}$ adalah keterangan sederhana dari $LBP_{P,R}^{riu2}$ / $VAR_{P,R}$, ukuran fitur yang jauh lebih kecil daripada $LBP_{P,R}^{riu2}$ / $VAR_{P,R}$ dan sama seperti yang $LBP_{P,R}^{riu2}$.

VI. K – Nearest Neighbour

Metode K-Nearest Neighbour (K-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak. Masing – masing dimensi merepresentasikan ciri dari data. Ruang ini dibagi menjadi bagian – bagian berdasarkan klasifikasi data pembelajaran.

Pada fase pelatihan, algoritma ini hanya melakukan penyimpanan vector – vector ciri dan klasifikasi data *sample* pelatihan. Pada klasifikasi, *feature* yang sama dihitung untuk pengujian data. Jarak antara data uji dan data latih dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data uji dengan semua titik yang merepresentasikan data latih dengan rumus *Euclidean Distance* sebagai berikut :

$$\begin{aligned} d &= \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \\ &= \sqrt{\textstyle\sum_{i=1}^{n}(a_i - b_i)^2} \end{aligned} \tag{11}$$

Dimana,d adalah jarak antara vector a_i dan b_i dengan a = $(a_1, a_2, \ldots, a_n)$ dan b = $(b_1, b_2, \ldots, b_n)$.

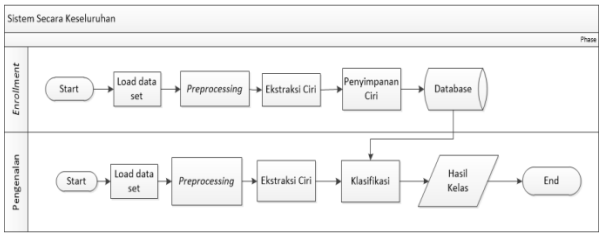
Ketepatan algoritma K-NN ini sangat dipengaruhi oleh ada atau tidaknya ciri yang tidak relevan, atau jika bobot ciri tersebut tidak setara dengan relevansinya terhadap klasifikasi.

K buah data latih terdekat akan melakukan perhitungan jumlah data latih untuk menentukan kelas mayoritas. Label data uji akan ditentukan berdasarkan label mayoritas dan jika ada lebih dari satu label mayoritas maka label data uji dapat ditentukan berdasarkan data latih yang paling dekat diantara mayoritas.

Keunggulan dari metode K-NN ini adalah relatif tidak terpengaruh dari error dari data dan juga dapat digunakan dengan kumpulan data dengan jumlah besar.

VII. Perancangan Sistem

Sistem pengenalan angka tulisan tangan ini dibangun dengan 2 proses utama, yaitu proses *enrollment* dan proses pengenalan. Gambaran sistem secara keseluruh dapat dilihat pada *flowchart* sistem dibawah.



Gambar 1 *Flowchart* Gambaran Sistem Keseluruhan

Pada tahap *enrollment* dilakukan proses penyimpanan ciri gambar angka tulisan tangan dari MNIST *database* dan formulir C1 yang dijadikan sebagai data model. Tujuan dilakukan *enrollment* adalah untuk menyimpan *vector* ciri data latih gambar angka tulisan tangan ke dalam *database* sistem. Setelah mendapatkan database ciri dilakukan proses pengenalan. Proses pengenalan merupakan proses pencocokan (*one to many)* gambar angka tulisan tangan terhadap ciri data model yang disimpan dalam database untuk menentukan kelas dari data uji.

1. *Preprocessing*

Tahap *preprocessing* dilakukan untuk menghasilkan citra uji yang sesuai dalam proses ekstraksi ciri dengan *Local Binary Pattern Variance.* Gambar yang sesuai adalah gambar *grayscale* yang ukurannya sama dengan gambar data latih, yaitu 28 x 28 piksel.

Pada MNIST *database* hanya dilakukan konversi data dengan membaca file CSV MNIST *Database* dan langsung menyimpannya ke dalam format JPEG. Proses ini tidak melakukan perubahan apapun terhadap file gambar karena gambar MNIST *Database* merupakan gambar yang telah berbentuk *grayscale* dengan ukuran yang telah dinormalisasi yaitu berukuran 28 x 28 piksel. Namun, gambar angka tulisan tangan dari gambar formulir C1 KPU harus dilakukan *preprocessing* agar mendapatkan gambar yang sesuai dengan gambar dari MNIST *database.* Proses *preprocessing* yang dilakukan pada gambar angka formulir C1 adalah penskalaan, binerisasi, penebalan, dan penyimpanan dalam *grayscale.* Adapun *diagram* konseptual dari tahapan *preprocessing:*

Gambar 2 Diagram Konseptual Preprocessing

2. Ekstraksi Ciri

Proses ekstraksi ciri dilakukan pada semua citra latih dan citra uji dengan menggunakan *Local Binary Pattern Variance.* Proses ekstraksi ciri ini dilakukan dengan beberapa tahap, yaitu segmentasi geometri, proses LBP, perhitungan VAR dan pembuatan histogram.

Vektor ciri dibuat berdasarkan tiga pembagian *region* yang berbeda, yaitu dengan 1 *region*, 4 *region* dan 9 *region*. Setiap *region* dilakukan proses ektraksi ciri yang menghasilkan 1 histogram. Maka, untuk 4 *region* terdapat 4 histogram ciri yang digabung menjadi 1 histogram. Begitu pun untuk 9 *region.*

Setiap piksel dari gambar angka tulisan tangan yang telah dilakukan *preprocessing* memiliki nilai hasil *grayscale*, yang kemudian dilakukan *thresholding* berpusat pada titik tengah. Piksel yang memiliki nilai sama tau lebih besar dari titik pusat piksel diberi nilai 1 selain itu diberi nilai 0. Kemudian nilai LBP didapat dari penjumlahan dua pangkat nilai angka yang bernilai satu. Berikut proses LBP pada blok 3x3 piksel.

4	3	2
7	4	1
9	8	7

Contoh

1	0	0
1		0
1	1	1

Threshold

1	2	4
128		8
64	32	16

Weight

Pattern = 11110001, LBP = 1 + 16 + 32 + 64 + 128 = 241

Operator pada LBP memiliki *label* yang ditandai dengan P dan R. P mewakili jumlah piksel tetangga yang digunakan dalam komputasi, sementara R adalah *radius* antara titik pusat dan piksel tetangga. Dalam hal ini penulis menggunakan 4 *radius*, yaitu R=1, R=2, R=3, dan R=4 dengan mengambil 8 ketetanggan (P). Dari banyaknya titik ketetanggan yang diambil, maka panjang histogram ciri yang dihasilkan adalah 2^8= 256.

Setelah menghitung hasil dari LBP setiap blok, hasil tersebut lalu dimasukkan ke dalam matriks baru. Perhitungan nilai LBP ini dimulai dengan menghitung LBP dari koordinat piksel gambar (2,2) hingga koordinat piksel gambar (27,27) dengan ukuran gambar 28x28 piksel. Maka, hasil dari proses LBP ini adalah matriks berukuran 26x26.

VAR adalah invarian terhadap pergesaran dalam *gray scale*, dalam hal ini menghitung rotasi kontras invarian lokal. Ada dua tahap dalam perhitungan VAR, yaitu menghitung rata – rata nilai dari titik tetangga sekitar titik pusat pada setiap blok perhitungan matriks LBP dan Melakukan pengurangan

setiap titik ketetanggan dengan nilai rata – rata, lalu dipangkat dua, dan kemudian dibagi dengan banyak tetangga yang digunakan .

Setelah mendapatkan 2 matrik hasil perhitungan, yaitu matriks hasil LBP dan matriks hasil VAR, maka tahap selanjutnya adalah membuat histogram dari kedua matriks tersebut. Nilai histogram ini diambil dari penjumlahan nilai VAR sesuai dengan indeks pada matriks LBP.

3. Klasifikasi

Sistem pengenalan angka tulisan tangan ini menggunakan klasifikasi K – Nearest Neighbour (K-NN). K-Nearest Neighbor adalah algoritma untuk menemukan pencocokan terdekat untuk kasus yang diberikan kepada kasus yang diketahui disimpan dalam memori. Proses klasifikasi K-NN memiliki beberapa tahap yaitu penentuan jumlah tetangga terdekat (K), perhitungan jarak, pengurutan hasil jarak, dan terakhir penentuan kelas.

K digunakan untuk menandakan berapa banyak penilaian yang digunakan untuk pengambilan keputusan. Nilai K yang terbaik untuk klasifikasi K-NN ini tergantung pada data. Secara umum, nilai K yang tinggi akan mengurangi *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Maka, penentuan nilai K sebaiknya jangan terlalu kecil dan jangan terlalu besar. K yang digunakan tugas akhir ini adalah K = 10, K=50 dan K = 100. K = 10 ditetapkan penulis sebagai nilai K pertama untuk klasifikasi ini karena jumlah kelas pada klasifikasi ini ada 10 kelas.

Algoritma K-NN menentukan kelas dari sebuah data uji dengan cara mencari K jarak terdekat antara data uj dengan semua data latih yang ada. Perhitungan jarak pada sistem ini menggunakan rumus *Euclidean Distance* pada rumus 11. Setelah mengetahui masing – masing jarak antara data uji dengan semua data latih, selanjutnya dilakukan pengurutan dari nilai terkecil hingga terbesar.

Kelas data uji didapatkan dari melihat K buah data latih yang memiliki paling banyak hasil perhitungan jarak paling kecil (tetangga terdekat). Jika terdapat dua atau lebih kelas yang memiliki jumlah tetangga terdekat yang sama, maka akan terjadi kondisi seimbang. Maka, data uji masuk ke dalam kelas yang salah satu data latihnya paling dekat di antara dua kelas paling dekat yang jumlahnya sama.

VIII. Pengujian dan Analisis

Pengujian sistem dilakukan untuk menganalisis kelebihan dan kekurangan sistem pengenalan angka tulisan tangan yang dibangun menggunakan metode ekstraksi ciri *Local Binary Pattern Variance* dan Klasifikasi *K-Nearest Neighbour*. Implementasi dan pengujian sistem pengenalan angka tulisan tangan menggunakan aplikasi Matlab R2012b. Pengujian ini dilakukan dengan beberapa scenario pengujian.

1. Skenario Pengujian dengan Menggunakan Data Latih MNIST dan Data Uji MNIST.

Citra yang digunakan pada pengujian pertama adalah citra dari data set MNIST sebanyak 50.000 data latih dan 10.000 data uji. Untuk mengetahui parameter terbaik yang digunakan pada metode *Local Binary Pattern Variance* dan *K-Nearest Neighbour*, maka dilakukan pencarian 4 akurasi terbaik dengan beberapa scenario pengujian sebagai berikut.

- Skenario 1 : Penetapan radius pada *LBP Variance.*

  Pengujian ini dilakukan pada citra data latih dan data uji MNIST dengan 1 *region* dan 256 bin histogram. K yang dipakai pada klasifikasi *K-Nearest Neighbour* adalah K = 10. Radius yang diuji adalah radius satu sampai lima (1-5) dengan masing – masing mengambil 8 ketetanggaan. Berikut hasil pengujian akurasi perbedaan radius.

Tabel 1 Tabel Pengujian Parameter Radius dengan 8 Ketetanggan

Radius	Jumlah Bin	Jumlah *Region*	Nilai K	Akurasi (%)
1	256	1	10	41,15
2	256	1	10	54,4700
3	256	1	10	75,9000
4	256	1	10	86,8000
5	256	1	10	85,0000

Berdasarkan tabel di atas , pengukuran akurasi terhadap parameter radius yang diujikan pada 8 ketetanggan menghasilkan akurasi terbaik sebesar 86,8 % dengan radius 4. Pengujian parameter radius hanya sampai radius 5 karena setelah radius 4 akurasi yang dihasilkan lebih kecil. Hal ini dikarenakan semakin besar radius, ciri yang dapat dijangkau dari *sample point* semakin besar, namun terlalu besarnya jumlah radius pada gambar mengakibatkan ciri yang dihasilkan lebih sedikit.

- Skenario 2 : Penetapan jumlah bin histogram ciri.

Pengujian ini dilakukan dengan mengurangi jumlah bin histogram dengan proses kuantisasi. Jumlah bin histogram yang diuji adalah 256 bin, 64 bin dan 32 bin.

Tabel 2 Tabel Pengujian Jumlah Bin Histogram Pada Ekstraksi Ciri LBP*Variance*

Jumlah Bin	Radius	Jumlah *Region*	K	Akurasi (%)	Waktu Komputasi (detik)
256	4	1	10	86,8000	230,5034
64	4	1	10	81,84	56,8954
32	4	1	10	73,87	32,7571

Berdasarkan table di atas, dapat dilihat akurasi terbaik dimiliki oleh ciri dengan 256 bin histogram, tapi waktu komputasi yang paling rendah, dalam hal ini tercepat, adalah ciri dengan 64 bin. Hal ini karena dengan 256 bin histogram, histogram ciri yang dihasilkan lebih detail tapi vektor ciri lebih panjang, sehingga waktu komputasinya jauh lebih lama sedangkan 64 bin ciri yang dihasilkan menjadi kurang detail namun vektor ciri lebih pendek sehingga lebih cepat.

- Skenario 3 : Pembagian *region* pada citra

Dari scenario 1 dan 2 telah didapatkan nilai *radius* terbaik dan jumlah bin histogram terbaik. Berikutnya adalah mencari akurasi terbaik dari pembagian region pada citra untuk mendapatkan jumlah *region* yang optimal dalam membagi citra menjadi bagian – bagian yang lebih kecil.

Tabel 3 Tabel Pengujian Jumlah Pembagian Region pada Citra

Jumlah *Region*	Jumlah Bin	Radius	K	Akurasi (%)	Waktu (detik)
1	256	4	10	86,80	123,4723
4	256	4	10	86,96	507,3551
9	256	4	10	89,81	1131,706
16	256	4	10	86,31	1878,657
1	64	4	10	81,84	29,76993
4	64	4	10	84,98	110.2027
9	64	4	10	89,65	239.806
16	64	4	10	85,54	458,6465

Berdasarkan tabel pengujian di atas, dapat dilihat perbedaan akurasi yang dihasilkan pada pembagian jumlah *region*, diketahui bahwa akurasi terbaik diperoleh pada besar akurasi 89,81% untuk 9 *region* dengan jumlah 256 bin histogram dan 89,65% pada 64 bin histogram. Semakin banyak jumlah regionnya, maka semakin tinggi akurasinya. Hal ini dikarenakan semain besarnya jumlah region, ekstraksi ciri yang dilakukan menjadi lebih detail.

- Skenario 4 : Penetapan nilai K pada klasifikasi *K-Nearest Neighbour.*

Pada tahap klasifikasi, digunakan *K-Nearest Neighbour* yang mencari K buah tetangga terdekat untuk menentukan kelas dari suatu data uji.

Tabel 4 Tabel Pengujian Nilai K pada Klasifikasi K-NN

K	Radius	Jumlah Bin	Jumlah Region	Akurasi (%)
10	4	256	9	89,8100
50	4	256	9	87,3100
100	4	256	9	85,8000
10	4	64	9	89,6500
50	4	64	9	87,6000
100	4	64	9	86,3300

Pada tabel di atas, akurasi terbaik diperoleh dari pengujian klasifikasi dengan K = 10 dengan akurasi 89,81 %. Semakin besar nilai K maka akurasi sistem semakin menurun. Hal ini dikarenakan semakin besar nilai K yang ditetapkan maka semakin banyak kemungkinan kelas data latih yang akan menjadi kelas data uji sehingga batasan antara setiap klasifkasi menjadi lebih kabur.

2. Skenario Pengujian dengan menggunakan data latih MNIST dan data uji gambar dari Formulir C1.

Citra yang digunakan pada pengujian ini menggunakan 50.000 data MNIST dan 550 data dari formulir C1 yang telah dilakukan *preprocessing*.

Tabel 5 Tabel Pengujian Data Latih MNIST dan Data Uji Formulir C1

Radius	Bin	Region	K	Akurasi (%)
4	256	9	10	46,0000
4	64	9	10	44,3636

Berdasarkan tabel di atas, dapat dilihat pengujian yang dilakukan menghasilkan akurasi yang rendah. Hal ini dapat terjadi karena perbedaan kualitas citra antara data MNIST dan data formulir C1.

3. Skenario Pengujian dengan menggunakan data latih dan data uji dari gambar dari Formulir C1

Pengujian terakhir menggunakan data latih dan data uji dari data set formulir C1 dengan menggunakan parameter terbaik. Citra yang dilakukan adalah sebanyak 2100 data latih dan 550 data uji.

Tabel 6 Tabel Pengujian Data Latih C1 dan Data Uji Formulir C1

Radius	Bin	Region	K	Akurasi
4	256	9	10	67,4545
4	64	9	10	70,9091

Berdasarkan tabel di atas, dapat dilihat pengujian yang dilakukan menghasilkan akurasi yang lebih tinggi daripada pengujian 2 yaitu meningkat 34,6 %. Hal ini membuktikan bawa data latih dari formulir C1 yang berjumlah 2100 sampel dapat mengenali data uji formulir C1 lebih baik daripada menggunakan data latih MNIST yang berjumlah 50.000.

IX. Kesimpulan

Dari beberapa pengujian dan analisis yang dilakukan, dapat diambil kesimpulan bahwa Sistem pengenalan angka tulisan tangan dengan menggunakan *Local Binary Pattern Variance* dan Klasifiksi *K-Nearest Neighbour* mampu mengenali citra karakter angka tulisan tangan pada data set MNIST dengan akurasi tertinggi mencapai 89,81% dan 89,65% dengan parameter terbaik, yaitu radius 4, 256 dan 64 bin histogram, pembagian 9 region dan K pada K-NN adalah 10. Namun, akurasi sistem menurun ketika sistem diterapkan pada data uji

formulir C1 menjadi 70,9091% dengan menggunakan data latih MNIST dan 46% dengan menggunakan data latih formulir C1. Hal ini karena perbedaan kualitas citra pada data set MNIST dan formulir C1.

X. Daftar Pustaka

[1] *Matlab*. (2015). Retrieved April 2015, from Mathworks Documentation: www.mathworls.com/help/

[2] Asmoro. (n.d.). Perbandingan Antara Metode Kohonen Neural Network dan Vector Quantization pada Sistem Pengenalan Tulisan Tangan secara Real Time. 7.

[3] Biglari, M., Mirzaei, F., & Neycharan, J. G. (2014). Persian/Arabic Handwritten Digit Recognition Using Local Binary Pattern. *International Journal of Digital Information and Wireless Communication*, 7.

[4] Cheriet, M., Kharma, N., Liu, C.-L., & Y.Suen, C. (2007). *Character Recognition Sistem : A Guide for Students and Practioner.* Canada: Wiley Interscience.

[5] Fauzi, M. R., Agus, N., & Ajulian, A. (n.d.). Mengubah Tulisan Tangan menjadi Text Digital OCR (Optical Character Recognition) dengan Menggunakan Metode Segmentasi dan Korelasi. 8.

[6] Gonzalez, R. C., & Woods, R. E. (2001). *Digital Image Processing.* New Jersey: Prentice Hall.

[7] Gui, Z., Zhang, L., & Zhang, D. (2009). Rotational Invariant Texture Classification Using LBP Variance (LBPV) with Global Matching. *Pattern Recognition*, 14.

[8] Jain, G., & Ko, J. (2008, 11 21). *Handwritten Digits Recognition.*

[9] Mohamed, A., & Yampolskiy, R. (n.d.). An Improve LBP Algorithm for Avatar Face Recognition.

[10] Mohamed, A., & Yampolskiy, R. (n.d.). *Face Recognition Based on Wavelet Transform and Adaptive Local Binary Pattern.*

[11] Perez-Cortes, C. J., Llobet, R., & Arlandis, J. (n.d.). Fast and Accurate Handwitten Character Recognition using Approximate Nearest Neighbours Search on Large Databases.

[12] Pierikaiinen, M., Hadid, A., Zhao, G., & Ahonen, T. (2011). *Computer Vision Using Local Binary Patterns.* Netherland: Springer.