

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1. Tulisan Tangan**

Tulisan tangan merupakan hasil seseorang dalam melakukan kegiatan menulis dengan menggunakan tangan. Pada setiap tulisan tangan pasti memiliki ciri khas atau keunikan tersendiri karena setiap orang memiliki karakter dan kepribadian yang berbeda-beda.

Seiring dengan perkembangan zaman, tulisan tangan yang memiliki perbedaan antara satu orang dengan yang lainnya dapat diolah menjadi citra digital dengan bantuan komputer. Mengubah tulisan tangan ke bentuk citra dapat dilakukan dengan scan atau memfoto dokumen kertas, foto dan bentuk lainnya [6].

#### **2.2. Citra Digital**

Perubahan dokumen cetak menjadi citra digital adalah proses digitalisasi. Proses digitalisasi dapat dilakukan pada dokumen cetak dengan cara memfoto atau scan dokumen kertas tersebut. Citra digital biasanya berbentuk persegi panjang dengan ukuran yang dinyatakan sebagai lebar x tinggi [7].

Citra digital biasanya disimpan dalam suatu berkas (file) dengan berbagai format tetapi dalam Microsoft windows berkas biasanya dinamai berkas bitmap (bmp). Format bmp memiliki kelebihan dan kekurangan tersendiri. Kelebihan format bmp untuk citra adalah mampu menampilkan citra ke layar monitor dengan menampilkan semua informasi yang ada, sedangkan untuk kekurangan format bmp adalah memiliki ukuran berkas yang relatif besar [7].

Citra dalam format bmp memiliki 3 macam jenis yaitu citra biner, citra berwarna dan citra hitam putih (*grayscale*). Dengan memiliki 1 bit saja citra biner dapat merepresentasikan nilai piksel karena citra biner hanya memiliki nilai keabuan dengan nilai 0 dan 1. Sedangkan untuk citra berwarna memiliki kombinasi 3 warna dasar yaitu warna RGB (Red, Green dan Blue). Salah satu contoh untuk citra warna adalah citra 24 bit. Setiap piksel dalam citra 24 bit dapat langsung

dikatakan kombinasi dari komponen warna RGB. Masing-masing komponen warna direpresentasikan menjadi 8 bit. Dengan demikian, bila 2 dipangkatkan 8 maka derajat keabuan memiliki jumlah 256 (0-255) untuk masing-masing komponen warna RGB [7].

### 2.3. Pengolahan Citra Digital

Istilah pengolahan citra digital adalah pemrosesan terhadap gambar 2 dimensi dengan menggunakan media komputer. Dalam konteks yang lebih luas, pengolahan citra digital merujuk pada pemrosesan setiap data atau gambar 2 dimensi. Citra digital merupakan sebuah larik (*array*) menggunakan bit tertentu yang berisi nilai-nilai real maupun kompleks [8]. Pengolahan citra digital merupakan proses pengolahan yang mempunyai ciri dari data input menjadi informasi output yang berbentuk citra. Sebuah informasi dalam citra memang sangat banyak tetapi banyak juga kekurangan dari citra tersebut. Kekurangan citra yang memiliki penurunan intensitas mutu bisa disebabkan karena, cacat, derau (*noise*), warna citra terlalu kontras ataupun citra yang diperoleh kabur sehingga informasi yang didapatkan akan berkurang [9].

Perbaikan citra yang memiliki banyak kekurangan dapat dilakukan dengan memanipulasi kualitas citra menjadi lebih baik. Hal-hal yang perlu dilakukan pada citra sebagai berikut [9]:

1. Perbaikan atau memodifikasi citra dilakukan untuk meningkatkan kualitas citra sehingga informasi yang didapatkan dalam citra bisa didapatkan lebih maksimal.
2. Elemen yang terdapat dalam citra dikelompokkan, dicocokkan dan diukur.
3. Penggabungan citra satu dengan citra lainnya.

Berdasarkan pemaparan diatas, pengolahan citra digital adalah suatu metode yang digunakan untuk mengolah suatu citra (gambar) 2 dimensi yang menghasilkan citra lainnya yang diolah sesuai dengan kebutuhan dengan menggunakan komputer. Suatu citra biasanya memiliki banyak kekurangan seperti derau (*noise*), kontras terlalu tinggi, kurang tajam dan sebagainya. Hal ini dikarenakan, citra sebagai fungsi kontinu dalam citra (gambar) 2 dimensi dari intensitas cahaya  $(x,y)$ .

Suatu citra dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Maka, citra digital adalah citra yang memiliki nilai x,y dan nilai amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit [8].

### 2.3.1. Grayscale

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki *grayscale* adalah hitam, keabuan dan putih [8]. Tingkat keabuan disini merupakan warna abu dengan berbagai tingkatan dari 0 sampai 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih [10]. Persamaan 2.1 adalah rumus konversi citra berwarna (RGB) menjadi *grayscale*.

$$\text{Gray} = (0.2989 * R) + (0.5870 * G) + (0.1141 * B) \quad (2.1)$$

Keterangan:

R = komponen nilai merah (*Red*) dari suatu titik pixel

G = komponen nilai hijau (*Green*) dari suatu titik pixel

B = komponen nilai biru (*Blue*) dari suatu titik pixel

Persamaan diatas merupakan salah satu rumus untuk mengkonversi citra berwarna (RGB) menjadi *grayscale*. Dipilihnya persamaan (2.1) dalam penelitian ini karena mata manusia secara alami lebih sensitif terhadap cahaya berwarna merah dan hijau. Maka dari itu, warna merah dan hijau diberi bobot lebih tinggi untuk memastikan bahwa keseimbangan intensitas relatif dalam citra *grayscale* yang dihasilkan mirip dengan warna RGB [10].

### 2.3.2. Thresholding

Citra biner adalah citra yang memiliki 2 nilai untuk setiap pikselnya. Nilai pikselnya yaitu 0 untuk citra hitam dan 1 untuk citra putih. Maka, untuk penyimpanan jenis citra yang dibutuhkan hanya 1-bit pada setiap pikselnya. Proses

*thresholding* digunakan untuk mengubah *foreground* yaitu tinta dari *background* kertas ke citra biner. Proses *thresholding* juga mengubah warna gambar menjadi citra biner yang menghasilkan nilai level dari *threshold*. Lalu, piksel yang memiliki nilai level dibawah level dari *threshold* diberi nilai 0 (warna hitam) dan piksel yang memiliki nilai level diatas level dari *threshold* diberi nilai 1 (warna putih) [11]. Persamaan 2.2 adalah algoritma biner yang digunakan:

$$g(x,y) = \begin{cases} 0 & \text{if } f(x,y) < T \\ 1 & \text{if } f(x,y) \geq T \end{cases} \quad (2.2)$$

Keterangan:

$g$  = *gray* yang merupakan nilai hasil *grayscale*

$T$  = Nilai *threshold* (128 untuk citra *grayscale*)

### 2.3.3. Segmentasi

Pada tahap segmentasi, metode yang digunakan adalah metode *Projection Profile*. Metode *projection profile* ini digunakan untuk menentukan suatu baris pada dokumen dimana spasi antar baris dikelompokkan dengan baik sehingga metode ini dapat digunakan untuk tulisan tangan. Metode *projection profile* dikelompokkan menjadi dua yaitu profil proyeksi secara horizontal dan profil proyeksi secara vertikal. Profil proyeksi horizontal digunakan untuk segmentasi baris sedangkan profil proyeksi vertikal digunakan untuk segmentasi kata [12].

Sistem kerja *projection profile* terhadap objek atau huruf ialah dengan menjadikan objek atau huruf tersebut kedalam bentuk garis-garis histogram horizontal dan vertikal. Suatu objek ketika citra diproyeksikan melalui sumbu  $x$  dan sumbu  $y$ , *projection profile* merupakan struktur data yang memiliki fungsi untuk menyimpan sejumlah piksel hitam [12]. Berikut contoh dari grafik profil proyeksi yang terdapat pada Gambar 2.1.



Gambar 2.1 Contoh Grafik Profil Proyeksi

Profil proyeksi horizontal direpresentasikan dengan suatu vektor  $Ph$  dengan  $M$ . Profil proyeksi horizontal pada kolom ke- $j$ , yaitu  $Ph[j]$  didefinisikan sebagai berikut:

$$P_h[j] = \sum_{i=1}^N S[i,j] \quad (2.6)$$

Profil proyeksi vertikal direpresentasikan dengan suatu vektor  $Pv$  dengan  $N$ . Profil proyeksi horizontal pada kolom ke- $i$ , yaitu  $Ph[i]$  didefinisikan sebagai berikut:

$$P_v[i] = \sum_{j=1}^M S[i,j] \quad (2.7)$$

#### 2.3.4. *Resize*

*Resize* citra adalah mengubah besarnya suatu ukuran pada citra digital kedalam piksel. *Resize* digunakan untuk memperbesar atau memperkecil ukuran pada citra sesuai dengan ukuran yang diinginkan [8]. Proses pada tahap *resize* juga dapat mempengaruhi size citra agar citra memiliki kualitas lebih baik atau lebih buruk. Untuk melakukan *resize*, maka dilakukan metode Nearest Neighbour Interpolation yaitu metode sederhana untuk mengubah ukuran citra.

Untuk mendapatkan nilai piksel  $x$  dan  $y$  yang akan digunakan dalam *resize* maka dilakukan persamaan 2.8:

$$\begin{aligned} x1 &= \text{round} \left( \frac{x2 * w_i}{w_t} \right) \\ y1 &= \text{round} \left( \frac{y2 * h_i}{h_t} \right) \end{aligned} \quad (2.8)$$

Keterangan:

$x1, y1$  = titik koordinat citra

$x2, y2$  = titik koordinat citra *resize*

$w_i$  = lebar dari citra asal

$w_t$  = lebar dari citra *resize*

$h_i$  = tinggi dari citra asal

$h_t$  = tinggi dari citra *resize*

#### 2.4. *United Moment Invariant*

*United Moment Invariant* merupakan metode ekstraksi ciri yang diperkenalkan oleh Yinan pada tahun 2003. Metode UMI ini adalah sebuah metode yang dapat membedakan bentuk pada kondisi diskrit. Dalam pengolahan citra tentunya penting untuk membedakan bentuknya, karena keunikan dalam iluminasi, sudut pandang, posisi ruang, skala, sudut dan lokasi bentuk pada gambar tersebut harus diubah. Oleh karena itu, untuk mengubah suatu gambar diperlukan suatu nilai fitur atau ciri yang menjaga RTS (rotasi, terjemahan dan skala) dalam kondisi apapun [13].

Sebagai ekstraksi ciri, UMI digunakan untuk mengenali atau mengklasifikasikan objek dalam gambar. Sebelum melakukan pengklasifikasian untuk mendapatkan nilai akhir dilakukan beberapa praproses dan ekstraksi ciri. Proses ini dilakukan agar menghasilkan nilai-nilai ciri pada setiap citra. Ekstraksi ciri yang dilakukan UMI menggunakan delapan fitur. Berikut adalah persamaan-persamaan yang nantinya akan berpengaruh terhadap rumus UMI [14]:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (2.9)$$

$m_{pq}$  adalah momen dua dimensi dari fungsi  $f(x, y)$ . Indeks kolom piksel ( $x$ ) dan indeks baris piksel ( $y$ ) menunjukkan koordinat piksel gambar, sedangkan  $f(x, y)$  adalah intensitas (tingkat abu-abu) piksel pada titik ( $x, y$ ).  $dx$  adalah arah utama perubahan  $x$  dan  $dy$  merupakan perubahan arah utama dari  $y$  [14].

Untuk citra digital, integral diganti dengan penjumlahan dan nilai  $m_{pq}$  menjadi:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (2.10)$$

Dimana  $p = 0, 1, 2, \dots$  dan  $q = 0, 1, 2, \dots$  yang bernilai integer. Momen pusat  $f(x, y)$  didefinisikan sebagai:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x + a)^p \cdot (y + b)^q f(x, y) \quad (2.11)$$

(2.12 dan 2.13) dengan mengganti  $a = -\bar{x}$  dan  $b = -\bar{y}$  seperti yang didefinisikan sebagai:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x + \bar{x})^p \cdot (y + \bar{y})^q f(x, y) \quad (2.12)$$

Dimana:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{dan} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (2.13)$$

Kemudian normalisasi central moment yang dinotasikan dengan  $\eta_{pq}$  dan didefinisikan pada persamaan 2.14 berikut:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (2.14)$$

Dimana,

$$\gamma = \frac{p+q}{2} + 1, \text{ untuk } p + q = 2, 3, 4, \dots$$

Selanjutnya, sebelum memasuki tahap UMI ada yang disebut dengan momen Hu. Momen Hu dikenal sebagai momen invarian karena sifatnya yang tidak terpengaruh oleh transformasi RTS bahkan pencerminan dengan menghitung tujuh besaran objek. Enam fitur pertama bersifat bebas dari RTS, sedangkan fitur ketujuh mempunyai tanda berlawanan untuk objek yang dicerminkan. Pada penelitian ini, fitur ketujuh momen Hu tidak dilakukan perhitungan karena tidak dibutuhkan pada saat perhitungan menggunakan metode UMI. Berikut adalah keenam fitur pada momen Hu dihitung dengan persamaan rumus 2.15 sampai 2.20 yang nantinya akan membantu dalam proses perhitungan metode UMI.

$$\phi_1 = \eta_{20} + \eta_{02} \quad (2.15)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + (2\eta_{02})^2 \quad (2.16)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \quad (2.17)$$

$$\phi_4 = (\eta_{30} + 3\eta_{12})^2 + (\eta_{03} + 3\eta_{21})^2 \quad (2.18)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{03} - 3\eta_{12})(\eta_{03} + \eta_{12}) [(\eta_{03} + \eta_{12})^2 - 3(\eta_{12} + \eta_{03})^2] \quad (2.19)$$

$$\phi_6 = (\eta_{20} - \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} - \eta_{12})(\eta_{30} + \eta_{21}) \quad (2.20)$$

Setelah dilakukan perhitungan terhadap rumus persamaan momen Hu, maka perhitungan akan dilanjutkan kepada delapan formula United Moment invariants (UMI), delapan rumus didefinisikan dalam Persamaan 2.21 sampai Persamaan 2.28.

$$\theta_1 = \frac{\sqrt{\phi_2}}{\phi_1} \quad (2.21)$$

$$\theta_2 = \frac{\phi_6}{\phi_1\phi_4} \quad (2.22)$$

$$\theta_3 = \frac{\sqrt{\phi_5}}{\phi_4} \quad (2.23)$$

$$\theta_4 = \frac{\phi_5}{\phi_3\phi_4} \quad (2.24)$$

$$\theta_5 = \frac{\phi_1\phi_6}{\phi_2\phi_3} \quad (2.25)$$

$$\theta_6 = \frac{(\phi_1 + \sqrt{\phi_2})\phi_3}{\phi_6} \quad (2.26)$$

$$\theta_7 = \frac{\phi_1\phi_5}{\phi_3\phi_6} \quad (2.27)$$

$$\theta_8 = \frac{(\phi_3 + \phi_4)}{\sqrt{\phi_5}} \quad (2.28)$$

## 2.5. Algoritma K-Nearest Neighbor

Metode k-Nearest Neighbor sangat sederhana dan mudah untuk diimplementasikan. Mirip dengan klustering, pengelompokan data baru berdasarkan jarak data baru itu ke beberapa data/tetangga (*neighbor*) terdekat. Dalam hal ini jumlah data/tetangga terdekat ditentukan oleh user yang dinyatakan dengan  $k$ . Misalkan untuk menentukan nilai  $k=5$ , maka setiap data testing dihitung jaraknya dari data *training* dan pilih 5 data *training* yang jaraknya paling dekat ke data *testing*. Setelah 5 data diketahui, periksa *output* atau label masing-masing data itu. Lalu tentukan *output* mana yang frekuensinya paling banyak [15]. Untuk



menghitung jarak antara titik x dan y bisa menggunakan jarak *Euclidean* sebagai berikut:

$$d(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

(2.29)

Keterangan:

$d(x,y)$  = jarak antara x dan y

$x_i$  = data latih ke-i

$y_i$  = data uji ke-i

$i$  = bilangan cacah  $I = (1,2,3,\dots,n)$

## 2.6. Penelitian Terkait

Pada bagian ini berisi penelitian yang pernah dilakukan dan literatur terkait dengan penelitian yang dilakukan:

*Tabel 2.1 Tabel Penelitian Sebelumnya*

No.	Penulis	Tahun	Judul	Isi
1.	Aditya Surya Wijaya, Nurul Chamidah, Mayanda Mega Santoni	2019	Pengenalan Karakter Tulisan Tangan Dengan K- Support Vector Nearest Neighbor	Metode K-SVNN adalah metode modifikasi dari metode KKN untuk mengatasi masalah memori dan waktu. Pada penelitian ini digunakan suatu metode untuk mereduksi data latih, sehingga penggunaan memori dapat berkurang dan proses klasifikasi pengujian cepat tanpa mengabaikan hasil akurasi. Penelitian ini juga memanfaatkan metode Zona (ICZ dan ZCZ) untuk ekstraksi ciri dan K-Nearest Neighbor

				untuk klasifikasi tulisan tangan.
2.	Ryan Arief Misnadin, S. Adi. S. Mola, Adriana Fanggidae	2014	Pengenalan Pola Tulisan Tangan Dengan Metode K- Nearest Neighbor	Penelitian ini menggunakan metode <i>K-Nearest Neighbor</i> (K-NN). Pada pengujiannya menggunakan 3 data set, dataset pertama dengan 15X36 data latih dan 5X36 data uji, dataset kedua dengan 35X36 data latih dan 15X36 data uji serta data set ketiga dengan 100X36 data latih dan 25X36 data uji selain itu digunakan 3 nilai K yaitu 1,3 dan 5. Persentase akurasi pengenalan secara keseluruhan terhadap seluruh karakter data set mencapai 40,38%.

3.	Intan Widyaningrum	2018	Pengenalan Tulisan Arab Menggunakan United Moment Invariant dan Modified Quadratic Discriminant Function	United Moment Invariant merupakan metode ekstraksi ciri yang memiliki keunggulan dapat menjaga agar inventarisasi terhadap rotasi, terjemahan dan skala (RTS) tetap berjalan dengan baik di berbagai macam keadaan. Dalam penelitian pengenalan tulisan tangan dengan metode Invariant Moment yang dilakukan oleh Ramteke pada tahun 2006, berhasil mendapatkan persentase kesuksesan sebesar 92%. Namun penelitian tersebut hanya dilakukan pada tulisan tangan Devanagari yang merupakan sebuah tulisan yang berisi huruf-huruf yang berasal dari India Utara. Sedangkan penelitian tulisan tangan berbahasa Inggris menggunakan metode <i>Invariant Moment</i> masih sedikit ataupun belum banyak dilakukan. Dengan melihat keberhasilan penelitian Ramteke, maka pada penelitian ini akan
----	-----------------------	------	---	---

				<p>ditambahkan dengan metode <i>United Moment Invariant</i> (UMI) untuk dapat meningkatkan tingkat akurasi yang lebih baik.</p>
--	--	--	--	---

## 2.8. Python

Python adalah Bahasa pemrograman interpretatif multiguna [16]. Bahasa pemrograman tingkat tinggi yang dapat melakukan berbagai eksekusi dari instruksi multiguna secara langsung dengan metode yang berorientasi objek dan menggunakan semantik dinamis untuk pembacaan suatu syntax merupakan penjelasan dari bahasa pemrograman python. Dengan mengendalikan manajemen memori secara otomatis (pointer) maka python dapat dipelajari dengan mudah dalam penggunaannya. Python juga dianggap sebagai bahasa yang mampu menggabungkan kemampuan, kapabilitas dengan sintaksis kode yang sangat jelas karena dilengkapi dengan fungsionalitas pustaka standar yang besar dan komprehensif.

Bahasa tingkat tinggi adalah salah satu bahasa mesin yang dapat mengeksekusi suatu program pada komputer. Dalam mengeksekusi sebuah program tentu memerlukan waktu sebelum jalannya program tersebut. Hal ini menjadi satu kelemahan pemrograman tingkat tinggi karena saat cukup lama dalam hal mengeksekusi suatu program. Hal itu tidak perlu dikhawatirkan, karena kelebihan dari bahasa tingkat tinggi yang mudah diubah portable untuk disesuaikan

dengan mesin yang dijelankannya membuat banyaknya aplikasi-aplikasi diluaran sana ditulis menggunakan bahasa tingkat tinggi salah satunya adalah phyton.

Dalam bahasa pemrograman phyton memiliki banyak library yang dapat digunakan untuk mempermudah pada saat menulis suatu program sesuai dengan kebutuh, berikut adalah beberapa library yang dapat digunakan:

### **2.8.1. Tensorflow**

Tensorflow merupakan sebuah perpustakaan software yang digunakan machine learning dalam berbagai macam tugas pemahaman persepsi dan bahasa. Tensorflow dikembangkan oleh Brain Google Team untuk penelitian dan produksi google, kemudian dirilis dibawah lisensi Apache 2.0 pada November 2015. Tensorflow merupakan API generasi kedua yang menggantikan DistBelief yang digunakan oleh 50 tim yang berbeda untuk penelitian dan pengembangan puluhan produk komersial milik google.

Tensorflow juga merupakan library yang ditulis dengan bahasa C++ dan biasanya digunakan dengan bahasa pemrograman Python. Dengan adanya tensorflow, sekarang sudah bisa menggunakan beberapa fitur tensorflow di sisi web browser tanpa harus dibebani oleh instalasi yang cukup ‘menantang’. Dengan tensorflow kita tinggal melakukan instalasi dengan `npm install @tensorflow/tfjs` ataupun juga dengan menggunakan CDN. TensorFlow kemudian menggabungkan aljabar komputasi teknik pengoptimalan kompilasi, mempermudah penghitungan banyak ekspresi matematis dimana masalahnya adalah waktu yang dibutuhkan untuk melakukan perhitungan. Fitur utamanya meliputi:

1. Mendefinisikan, mengoptimalkan, dan menghitung secara efisien ekspresi matematis yang melibatkan array multi dimensi (tensors).
2. Pemrograman pendukung jaringan syaraf dalam dan teknik pembelajaran mesin
3. Penggunaan GPU yang transparan, mengotomatisasi manajemen dan optimalisasi memori yang sama dan data yang digunakan. Tensorflow bisa menulis kode yang sama dan menjelankannya baik di CPU atau GPU. Lebih khusus lagi, TensorFlow akan mengetahui bagian perhitungan mana yang

harus dipindahkan ke GPU. Skalabilitas komputasi yang tinggi di seluruh mesin dan kumpulan data yang besar.

### **2.8.2. Numpy**

Numpy adalah salah satu library yang terdapat pada bahasa pemrograman python yang digunakan untuk komputasi numerik. Numpy menghadirkan kekuatan komputasi bahasa seperti C dan Fortran ke python, bahasa yang jauh lebih mudah dipelajari dan digunakan. Numpy membentuk dasar pustaka pembelajaran mesin yang handal seperti scikit-learn dan SciPy [17].

### **2.8.3. OpenCV**

OpenCV (Open Source Computer Vision) adalah library yang dibangun untuk computer vision sebagai sumber dan pustaka pembelajaran perangkat lunak. openCV ini memiliki lebih dari 2.500 library yang dapat digunakan. Library itu dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, dan banyak hal lain juga. openCV tidak hanya dapat digunakan dalam bahasa pemrograman python saja melainkan bisa digunakan pada aplikasi MATLAB yang didukung oleh Windows, Linux, Android dan Mac OS [18].

## **2.9. Pengujian Sistem**

Pengujian sistem adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan-perbedaan antara hasil yang diterapkan dengan hasil yang terjadi [19]. Pengujian seharusnya meliputi tiga konsep berikut.

1. Demonstrasi validitas perangkat lunak pada masing-masing tahap di siklus pengembangan sistem.
2. Penentuan validitas sistem akhir dikaitkan dengan kebutuhan pemakai.
3. Pemeriksa perilaku sistem dengan mengeksekusi sistem pada data sampel pengujian.

Pada dasarnya pengujian diartikan sebagai aktivitas yang dapat atau hanya dilakukan setelah pengkodean (kode program selesai).

Namun, pengujian seharusnya dilakukan dalam skala lebih luas. Pengujian dapat dilakukan begitu spesifikasi kebutuhan telah dapat didefinisikan. Evaluasi terhadap spesifikasi dan perancangan juga merupakan Teknik pengujian. Kategori pengujian dapat dikategorikan menjadi dua [19], yaitu:

1. Berdasarkan ketersediaan logic sistem, terdiri dari *Black box testing* dan *White box testing*.
2. Berdasarkan arah pengujian, terdiri dari Pengujian *top down* dan Pengujian *bottom up*.

### **2.9.1. Pengujian Black Box**

Konsep *Black box* digunakan untuk merepresentasikan sistem yang cara kerja didalamnya tidak tersedia untuk diinspeksi. Di dalam *black box*, item-item yang diuji dianggap “gelap” karena logikanya tidak diketahui, yang diketahui hanya apa yang masuk dan apa yang keluar dari *black box* [19].

Pada pengujian *black box*, kasus-kasus pengujian berdasarkan pada spesifikasi sistem. Rencana pengujian dapat dimulai sedini mungkin di proses pengembangan perangkat lunak. Teknik pengujian konvensional yang termasuk pengujian “*black box*” adalah sebagai berikut [19]:

1. Graph-based testing
2. Equivalence partitioning
3. Comparison testing
4. Orthogonal array testing

Pada pengujian *black box*, kita mencoba berbagai masukan dan memeriksa keluaran yang dihasilkan. Kita dapat mempelajari apa yang dilakukan kotak, tapi tidak mengetahui sama sekali mengenai cara konversi dilakukan. Teknik pengujian *black box* juga dapat digunakan untuk pengujian berbasis skenario, dimana isi dalam sistem mungkin tidak tersedia di inspeksi tapi masukan dan keluaran yang didefinisikan dengan *use case* dan informasi analisis yang lain.

### **2.9.2. Pengujian Akurasi**

Perhitungan akurasi yang digunakan pada penelitian untuk mengetahui tingkat akurasi pengenalan tulisan tangan [20]. Untuk mengetahui tingkat

akurasi dari *United Moment Invariant* dan *K-Nearest Neighbor* dalam pengenalan karakter tulisan tangan dilakukan perhitungan rumus sebagai berikut:

$$\text{Akurasi} = \frac{\text{Jumlah yang benar}}{\text{jumlah data}} \times 100\% \quad (2.11)$$