

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Optical Character Recognition (OCR)

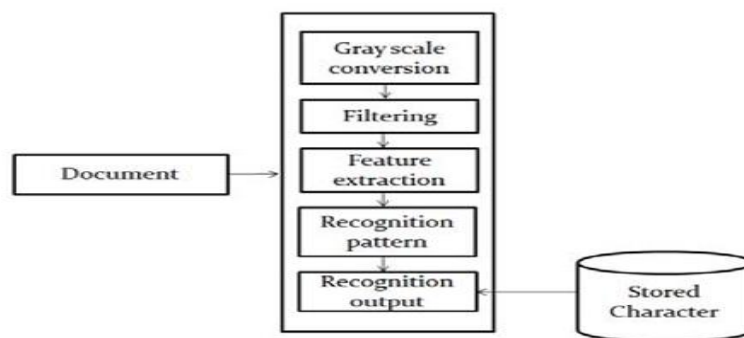
*Character Recognition* bertugas untuk mengenali tulisan didalam mengenali karakter tulisan dalam gambar dan merubahnya kedalam *American Standard Code for Information Interchange* (ASCII) atau bahasa mesin lainnya yang setara dan dapat diedit. Terdapat dua macam *Character Recognition*, antara lain: *Offline* dan *Online Character Recognition* (Rao, Sasrty, Chakracarthy, & Kalyanchakravarthi, 2016).

OCR memiliki beberapa tahapan utama yang terdiri dari:

- a. File Input berupa file citra digital dengan format \*.bmp atau \*.jpg.
- b. Preprocessing merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada gambar input untuk proses selanjutnya.
- c. Segmentasi adalah proses memisahkan area pengamatan (region) pada tiap karakter yang dideteksi.
- d. Normalisasi adalah proses merubah dimensi region tiap karakter dan ketebalan karakter.
- e. Ekstraksi ciri adalah proses untuk mengambil ciri-ciri tertentu dari karakter yang diamati.
- f. Recognition merupakan proses untuk mengenali karakter yang diamati dengan  
  
cara membandingkan ciri-ciri karakter yang diperoleh dengan ciri-ciri karakter yang ada pada basis data.

*Character recognition* juga diklasifikasikan kedalam dua tipe berdasarkan metode tulisannya, antara lain: *Optical Character Recognition* (OCR) dan *Handwritten Character Recognition* (HCR). Dimana akurasi pada HCR biasanya masih lebih rendah dikarenakan besarnya perbedaan bentuk dan tipe tulisan. Perbedaan karakter dalam Bahasa juga berpengaruh besar, contohnya: tulisan kanji mandarin, jepang, dan lainnya (Rao, Sasrty, Chakracarthy, & Kalyanchakravarthi, 2016).

Dalam proses OCR berikut ini gambaran sistem yang akan dilakukan (Mohammad, Anarase, Shingote, & Ghanwat, 2014):



**Gambar 2.1 System Block Diagram**

## 2.2 Implementasi dan Spesifikasi

### 2.2.1 Implementasi Optical Character Recognition

*Optical character recognition* (OCR) adalah proses konversi gambar huruf menjadi karakter *ASCII* yang dikenali oleh komputer. Gambar huruf yang dimaksud dapat berupa hasil scan dokumen, hasil print-screen halaman web, hasil foto, dan lain-lain.

Salah satu kebutuhan mengapa perlu konversi gambar huruf menjadi karakter *ASCII* adalah karakter *ASCII* memiliki kapasitas penyimpanan yang lebih kecil. Contohnya, suatu paragraf di-*printscreens* dan disimpan dalam format png. Paragraf tersebut juga di-*copy*. (Mohammad, Anarase, Shingote, & Ghanwat, 2014).

### 2.2.2 Spesifikasi Optical Character recognition

*OCR* adalah sistem yang sudah lama dikembangkan. Tahun 1914, Emanuel Goldberg telah mulai membuat sistem *OCR* yang berfungsi untuk telegram dan alat baca untuk orang tunanetra. Sistem *OCR* terus dikembangkan hingga kini sehingga dapat menghasilkan akurasi yang lebih baik bahkan dalam situasi-situasi yang dimana karakter sulit untuk dikenali. Pengaplikasian *OCR* sendiri memungkinkan komputer untuk melakukan proses lebih lanjut, contohnya translasi ke bahasa asing, pencarian, sistem baca otomatis untuk orang tunanetra, input data, pengenalan karakter seperti plat nomor, pengetesan *CAPTCHA*, atau masalah teks lainnya.

Hasil dari *OCR* bisa disimpan langsung dalam bentuk *ASCII*, namun untuk kasus tertentu, butuh disimpan layout-nya. Yang dimaksud dengan layout adalah posisi paragraf, margin, dan lainnya, sehingga sama persis dengan gambar yang diolah. Layout butuh disimpan contohnya dalam kasus konversi hasil *scan* buku ke dalam file .doc, tentunya posisi paragraf dan lainnya perlu disamakan.

### 2.3 Penelitian Terdahulu

**Tabel 2.1 Daftar Referensi Jurnal Nasional**

| No. | Nama Peneliti   | Judul Penelitian   | Metodologi Penelitian   | Hasil Penelitian   |
|-----|---|--|---|--|
| 1.  | Eka Ardianto, Veronica Lusiana dan Wiwien Hadikurniawati (2011) | Rancang Bangun Aplikasi Pengolah Gambar Digital untuk Segmentasi Otomatis Lokasi Objek Angka | Penelitian ini melakukan pensegmentasian lokasi angka meter listrik. Penggunaan algoritma dirancang untuk mendapatkan | Hasil akhir yang diperoleh adalah sebesar 47% dari sample yang diambil ternyata dapat menemukan lokasi angka |

|    |  |  |   |  |
|----|--|--|---|--|
|    |  | pada Meter Listrik.  | lokasi angka dengan menggunakan fungsi <code>regionprops</code> pada lingkungan MATLAB.   | meter dengan baik.   |
| 2. | Robert Gunawan, Sri Suwarno, Widi Hapsari (2014) | Penerapan <i>Optical Character Recognition</i> (OCR) untuk Pembacaan Meteran Listrik PLN | Pengenalan karakter menggunakan dua metode, yaitu: <i>smearing</i> untuk mendeteksi area angka pemakaian pelanggan, <i>Connected Component Labeling</i> untuk segmentasi karakter, dan <i>template matching</i> untuk pengenalan. | Pengenalan dengan dua langkah tersebut tidak terlalu baik dalam mengenali karakter yang ada pada meteran listrik. Hal ini terjadi karena adanya beberapa factor, seperti: masih adanya noise yang mengganggu pengenalan karakter, sudut pengambilan gambar, pengaruh pencahayaan, pengaruh batas <i>smearing</i> |
| 3. | Aldi Setiawan, Herry Sujaini,                    | <i>Implementasi Optical</i>  | Tujuan dari penelitian ini  | hasil Pengujian <i>black box</i> pada  |

|  |                                     |  |   |   |
|--|-------------------------------------|--|---|---|
|  | <p>Arif Bijaksana<br/>PN (2017)</p> | <p><i>Character Recognition</i> (OCR) pada Mesin Penerjemah Bahasa Indonesia ke Bahasa Inggris</p> | <p>adalah membuat <i>Optical Character Recognition</i> (OCR) pada mesin penerjemah bahasa Indonesia ke bahasa Inggris dengan menggunakan <i>platform Android</i> untuk penginputan kata penerjemahan. Sistem dibangun menggunakan <i>Optical Character Recognition</i> (OCR) dengan <i>library mobile vision</i> agar <i>user</i> tidak perlu lagi mengetikkan teks sebagai inputan kata yang akan diterjemahkan dengan material pengujian menggunakan gambar teks sebanyak lima gambar dan lembar teks sebanyak 30</p> | <p>kalimat aplikasi dapat menangkap kata dengan cukup baik pada satu kalimat dan pada saat proses penerjemahan aplikasi tidak dapat menerjemahkan kata dengan menggunakan huruf kapital dan tanda baca dan berdasarkan dari hasil perhitungan <i>recall</i> dan <i>precision</i> pada gambar teks kita dapatkan nilai <i>precision</i> sebesar 1.00, nilai <i>recall</i> sebesar 1.00 dan nilai <i>f-measure</i> sebesar 1.00. berdasarkan dari hasil perhitungan <i>recall</i> dan <i>precision</i> pada lebar teks didapatkan nilai</p> |
|--|-------------------------------------|--|---|---|

|    |   |   |   |  |
|----|---|---|---|--|
|    |   |   | kalimat.  | <p><i>precision</i> sebesar 0,87, nilai <i>recall</i> sebesar 0,97 dan nilai <i>f-measure</i> sebesar 0,90, berdasarkan dari hasil perhitungan <i>recall</i> dan <i>precision</i> pada penerjemah didapatkan nilai <i>precision</i> sebesar 0,81, nilai <i>recall</i> sebesar 0,60 dan nilai <i>f-measure</i> sebesar 0,68. Hal ini menunjukkan bahwa aplikasi dapat menangkap hasil kalimat dengan cukup baik pada beberapa font dan ukuran yang berbeda.</p> |
| 4. | Yan Puspitarani, Yenie Syukriyah (2020) | <i>Pemanfaatan Optical Character Recognition Dan Text Feature</i> | penelitian ini dilakukan untuk membuat sistem yang memanfaatkan teknologi OCR | Hasil pengujian terhadap prototipe menunjukkan akurasi sebesar 66.7% terhadap  |

|    |                |   |   |   |
|----|----------------|---|---|---|
|    |                | <p><i>Extraction</i></p> <p>Untuk Membangun Basisdata Pengaduan Tenaga Kerja</p>  | <p>dan <i>text feature extraction</i> agar dapat melakukan input data pengaduan secara otomatis.</p> <p>Penelitian ini menghasilkan <i>prototype input</i> surat dan basisdata penyimpanan surat yang dapat dimanfaatkan lebih lanjut untuk <i>Data Mining</i> dan <i>Business Intelligent</i>.</p> | <p>surat hasil OCR dan 91.67% terhadap surat yang diketik manual.</p>   |
| 5. | Suciati (2022) | <p>Pembacaan Angka Meteran Listrik Pascabayar dengan Menggunakan <i>Algoritma Optical Character Recognition</i> (OCR)</p> | <p>Penelitian ini bertujuan untuk mengembangkan sebuah perangkat lunak yang dapat meningkatkan akurasi pembacaan meteran listrik pascabayar. Pada tahapan awal dilakukan akuisisi citra selanjutnya</p>   | <p>Berdasarkan hasil pengujian, sistem pembacaan angka meteran listrik pascabayar dengan menggunakan <i>algoritma Optical Character Recognition</i></p> |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  | proses segmentasi menggunakan bounding box dan dilanjutkan pencocokan. | (OCR) memiliki persentase keberhasilan atau tingkat akurasi pengenalan baik dengan akurasi pengenalan. |
|--|--|--|--|--|

**Tabel 2.2 Daftar Referensi Jurnal InterNasional**

| <b>No.</b> | <b>Nama Peneliti</b>  | <b>Judul Penelitian</b>  | <b>Metodologi Penelitian</b>   | <b>Hasil Penelitian</b>   |
|------------|---|--|--|---|
| 1.         | Ahmed Talat Sahlol, Mohamed Abd Elaziz, Mohammed A. A. Al Qaness, Sunghwan Kim (2020) | Karakter Optik Arab tulisan tangan Pendekatan Pengakuan Berdasarkan <i>Algoritma Optimasi Paus Hibrida</i> Dengan Perangkat Kasar Lingkungan | pendekatan pembelajaran mesin <i>hybrid</i> yang menggunakan perangkat kasar lingkungan dengan algoritme pengoptimalan paus biner untuk memilih yang paling banyak fitur yang sesuai untuk pengenalan huruf Arab tulisan tangan. Untuk memvalidasi pendekatan yang diusulkan, kami menggunakan kumpulan data | Hasilnya menunjukkan keuntungan yang jelas dari pendekatan yang diusulkan dalam hal: akurasi pengenalan, jejak memori, dan waktu prosesor daripada yang tidak memiliki fitur yang diusulkan metode. |



|    |   |  |  |  |
|----|---|--|--|--|
|    |   |  | <i>CENPARMI</i> , yang merupakan kumpulan data terkenal untuk eksperimen pembelajaran mesin yang melibatkan huruf arab tulisan tangan.   |  |
| 2. | Yulia S. Chernyshova, Alexander V. Sheshkus, Vladimir V. (2020) | Kerangka Kerja CNN Dua Langkah untuk Baris Teks Pengakuan dalam Gambar yang Diambil Kamera | menggunakan metode pengolahan citra untuk langkah <i>segmentasi</i> atau <i>JST end-to-end</i> . Untuk memenuhi batasan ketat pada ukuran memori yang dikenakan oleh sistem tertanam dan untuk menghindari <i>overfitting</i> , kami menggunakan <i>JST</i> dengan jumlah kecil parameter yang dapat dilatih. Tujuan utama kerangka kerja kami adalah untuk mengenali gambar berkualitas rendah dari dokumen identitas | menunjukkan bahwa solusi kami menunjukkan akurasi pengenalan yang tinggi pada kumpulan data alami bahkan dilatih pada murni sintetis data. |

|    |  |  |   |  |
|----|--|--|---|--|
|    |  |  | dengan latar belakang yang kompleks dan berbagai bahasa dan <i>font</i> .   |  |
| 3. | Jaewoo Park, Eunji Lee, Yoonsik Kim, Isaac Kang, Hyung Il Koo, Nam Ik Cho (2020) | Pengenalan Karakter Optik Multi-Bahasa Sistem Menggunakan Penguatan Pembelajaran Segmen Karakter | mengusulkan kerangka kerja baru menggunakan tiga blok saraf ( <i>segmenter</i> , <i>switcher</i> , dan beberapa pengenalan) dan pembelajaran penguatan segmenter: segmenter mempartisi gambar kata yang diberikan menjadi beberapa gambar karakter, pengalih menetapkan pengenalan untuk setiap sub-gambar, dan pengenalan melakukan pengenalan sub-gambar yang ditugaskan. Pelatihan pengenalan dan pengalih dapat | Hasil eksperimen telah menunjukkan bahwa metode yang diusulkan secara signifikan meningkatkan kinerja untuk skrip multi-bahasa dan bahasa set karakter besar tanpa menggunakan label batas karakter. |

|    |                |                           |  |                 |
|----|----------------|---------------------------|--|-----------------|
|    |                |                           | <p>dianggap tradisional tugas klasifikasi gambar dan kami melatihnya dengan metode pembelajaran terawasi. Namun, yang diawasi pembelajaran segmenter memiliki dua kelemahan kritis: Fungsi tujuannya kurang optimal dan pelatihannya membutuhkan sejumlah besar upaya anotasi. Jadi, dengan mengadopsi algoritma <i>REINFORCE</i>, kami melatih segmenter untuk mengoptimalkan kinerja keseluruhan, yaitu, kami meminimalkan jarak edit dari pengenalan akhir hasil.</p> |                 |
| 4. | Jamshed Memon, | Pengenalan Karakter Optik | menganalisis dokumen tulisan   | untuk merangkum |

|    |  |  |   |  |
|----|--|--|---|--|
|    | Maira Sami,<br>Rizwan<br>Ahmed<br>(2020)                                     | Tulisan Tangan<br>(OCR):<br>Sistematis yang<br>Menyeluruh<br>Tinjauan Pustaka<br>(SLR)   | tangan dan cetakan<br>untuk mengubahnya<br>menjadi format<br>elektronik.  | penelitian yang<br>telah dilakukan<br>tentang<br>pengenalan<br>karakter tulisan<br>tangan dokumen<br>dan untuk<br>memberikan<br>arahan<br>penelitian.  |
| 5. | Asghar Ali<br>Chandio, Md.<br>Asikuzzaman,<br>Mark R.<br>Pickering<br>(2020) | Pengenalan<br>Karakter Kursif<br>dalam<br>Pemandangan<br>Alam Gambar<br>Menggunakan<br><i>Konvolusi<br/>Multilevel Fusi<br/>Jaringan Saraf</i> | fitur <i>multi-skala<br/>agregasi (MSFA)</i><br>dan arsitektur<br>jaringan <i>multi-level<br/>feature fusion</i><br>(MLFF) untuk<br>mengenali yang<br>terisolasi Karakter<br>Urdu dalam gambar<br>alami. Jaringan<br>pertama-tama<br>menggabungkan<br>fitur multi-skala dari<br>konvolusi lapisan<br>dengan up-sampling<br>dan operasi<br>penambahan dan<br>kemudian<br>menggabungkannya<br>dengan fitur tingkat<br>tinggi. | Hasil percobaan<br>menunjukkan<br>bahwa agregasi<br>fitur multi-skala<br>dan multilevel<br>dan<br>perpaduannya<br>lebih efektif, dan<br>berkinerja lebih<br>baik metode lain<br>pada gambar<br>karakter Urdu<br>dan kumpulan<br>data Chars74K. |

## 2.4 Aplikasi

Menurut Hasan Abdurahman dan Asep Ririh Riswaya (2014), aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

## 2.5 Pemrograman

Program komputer atau sering kali disingkat sebagai program adalah serangkaian instruksi yang ditulis untuk melakukan suatu fungsi spesifik pada komputer. Komputer pada dasarnya membutuhkan keberadaan program agar bisa menjalankan fungsinya sebagai komputer, biasanya hal ini dilakukan dengan cara mengeksekusi serangkaian instruksi program tersebut pada *prosesor*.

Sebuah program biasanya memiliki suatu bentuk model pengeksesian tertentu agar dapat secara langsung dieksekusi oleh komputer. Program yang sama dalam format kode yang dapat dibaca oleh manusia disebut sebagai kode sumber, bentuk program yang memungkinkan programmer menganalisis serta melakukan penelaahan algoritma yang digunakan pada program tersebut.

Kode sumber tersebut pada akhirnya dikompilasi oleh utilitas bahasa pemrograman tertentu sehingga membentuk sebuah program. bentuk alternatif lain model pengeksesian sebuah program adalah dengan menggunakan bantuan *interpreter*, kode sumber tersebut langsung dijalankan oleh *utilitas interpreter* suatu bahasa pemrograman yang digunakan. Beberapa program komputer dapat dijalankan pada sebuah komputer pada saat bersamaan, kemampuan komputer untuk menjalankan beberapa program pada saat bersamaan disebut sebagai multitasking. Program komputer dapat dikategorikan menurut fungsinya; perangkat lunak sistem atau perangkat lunak aplikasi. (Saragih, 2016)

## 2.6 Kode Pemrograman

kode program (*source code*) adalah kumpulan deklarasi atau pernyataan dari bahasa pemrograman computer yang di tulis dan bisa dibaca oleh manusia. *Source code* merupakan pernyataan yang digunakan oleh *programmer* untuk berkomunikasi dengan computer menggunakan beberapa perintah yang terdefinisi. *Source code* ini biasanya dibuat dalam satu atau lebih file teks, terkadang disimpan didalam database yang tersimpan sebagai prosedur dan bisa juga muncul sebagai potongan kode yang tercetak di buku ataupun media lainnya. Banyaknya pernyataan atau deklarasi dari file *source code* bisa diatur dan diubah dalam bentuk *directory* pohon, atau biasa dikenal dengan *source tree*. (Belakang, 2015)

## 2.7 Jenis Format Citra Digital

Format file citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan dalam menyimpan citra dalam sebuah file. Setiap format memiliki karakteristik masing-masing. Berikut adalah penjelasan beberapa format umum digunakan saat ini. (tdy ok 2017)

**Tabel 2.3 File Format Citra Digital**

| Format Gambar       | Ekstensi    | Keterangan   |
|---------------------|-------------|--|
| Bitmap              | .bmp        | Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai pixel. |
| Tagged Image Format | .tif, .tiff | Format .tif merupakan format penyimpanan citra yang dapat digunakan untuk  |

|                             |      |   |
|-----------------------------|------|---|
|                             |      | menyimpan citra bitmap hingga citra dengan warna palet terkompresi. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi.   |
| Portable Network Graphics   | .png | Format .png adalah format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra grayscale, citra dengan palet warna, dan juga citra fullcolor. Format .png juga mampu menyimpan informasi hingga kanal alpha dengan penyimpanan sebesar 1 hingga 16 bit per kanal. |
| JPEG                        | .jpg | jpg adalah format yang sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG.  |
| Graphics Interchange Format | .gif | Format ini dapat digunakan pada citra warna dengan palet 8 bit. Penggunaan umumnya pada aplikasi web. Kualitas yang rendah menyebabkan format ini tidak terlalu populer di kalangan peneliti pengolahan citra digital.  |
| RGB                         | .rgb | Format ini merupakan format penyimpanan citra yang dibuat oleh silicon graphics untuk menyimpan citra berwarna.   |

|                            |                 |   |
|----------------------------|-----------------|---|
| RAS                        | .ras            | Format .ras digunakan untuk menyimpan citra dengan format RGB tanpa kompresi.   |
| Postscript                 | .ps, .eps, epfs | Format ini diperkenalkan sebagai format untuk menyimpan citra buku elektronik. Dalam format ini, citra direpresentasikan ke dalam deret nilai desimal atau hexadesimal yang dikodekan ke dalam ASCII.   |
| Portable Image File Format |                 | Format ini memiliki beberapa bagian di antaranya adalah portable bitmap, portable graymap, portable pixmap, dan portable network map dengan format berturut-turut adalah .pbm, .pgm, .ppm dan .pnm. Format ini baik digunakan untuk menyimpan dan membaca kembali data citra.   |
| PPM                        |                 | PPM terdiri dari dua bagian umum yaitu bagian pendahuluan dan bagian data citra. Bagian pendahuluan memiliki tiga bagian kecil, yang pertama adalah pengenalan PPM yang dapat berupa p3 (untuk citra ASCII) dan p6 (untuk citra binari). Bagian pendahuluan yang kedua adalah ukuran panjang dan lebar citra. Bagian ketiga dari pendahuluan adalah nilai maksimum dari komponen warna. Keistimewaannya adalah dalam data citra dapat disimpan komentar dengan memberikan tanda '#' sebelum komentar. |
| PGM                        |                 | Format ini hampir mirip dengan format   |



|     |  |   |
|-----|--|---|
|     |  | PPM hanya saja format ini menyimpan informasi grayscale (satu nilai per pixel). Pengenal yang digunakan adalah p2 dan p5.   |
| PBM |  | digunakan untuk menyimpan citra biner. Hampir sama dengan PPM dan PGM, Format PBM ini memiliki pendahuluan, hanya saja pendahuluannya tidak memiliki bagian ketiga (penjelasan nilai maksimum pixel). Penggenap yang digunakan adalah p1. |

## 2.8 Pengertian Pengenalan Karakter

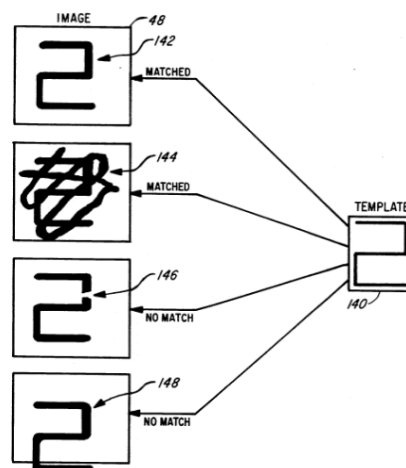
Pengenalan karakter yang umum disebut sebagai OCR (*Optical Character Recognition*) termasuk pada pengenalan pola, yang mana sistem ini khusus digunakan untuk mengenali tulisan teks cetak yang diketik dan dicetak menggunakan printer. Pengertian pengenalan pola sendiri merupakan suatu sistem yang mencoba untuk membaca / mengenali apakah citra masukan yang diterima cocok dengan salah satu citra yang telah ditentukan. Pengaplikasian pengenalan pola diantaranya seperti pendeteksi sidik jari, tulisan, tanda tangan, bahkan wajah seseorang. Tujuan dari pengenalan karakter adalah untuk mendukung perkembangan teknologi yang semakin pesat dalam bentuk digital. Sehingga bila terdapat suatu data fisik yang dikehendaki menjadi bentuk digital, maka sistem pengenalan karakter ini dapat dimanfaatkan. (Hayuning & Pramesti, n.d.)

## 2.9 Image

*Image* (Citra) adalah persepsi masyarakat terhadap perusahaan atau produk. *Image* dipengaruhi oleh banyak faktor yang diluar kontrol perusahaan. Pengertian *image* (citra) adalah kepercayaan, ide, dan impressi seseorang terhadap sesuatu. Citra citra merupakan kesan, impressi, perasaan atau persepsi yang ada pada publik mengenai perusahaan, suatu obyek, orang atau lembaga. (Kotler, 2009: 57)

## 2.10 Template Matching

*Template matching* adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi *template* (acuan). Teknik ini banyak digunakan dalam bidang industri sebagai bagian dari *quality control*. Metode *template matching* adalah salah satu metode terapan dari teknik konvolusi. Metode ini sering digunakan untuk mengidentifikasi citra karakter huruf, angka, sidik jari (*fingerprint*) dan aplikasi-aplikasi pencocokan citra lainnya. Secara umum teknik konvolusi didefinisikan sebagai suatu cara untuk mengkombinasikan dua buah deret angka yang menghasilkan deret angka ke tiga. (M.Chairi Solin, Garuda Ginting, Matias Julyus Fika Sirati, 2019)



Gambar 2.2 Template Matching

## 2.11 JavaScript

*JavaScript* adalah bahasa yang digunakan untuk membuat program yang digunakan agar dokument HTML yang ditampilkan pada sebuah Browser menjadi lebih interaktif, tidak sekedar indah saja. *JavaScript* memberikan beberapa fungsionalitas ke dalam halaman *web*, sehingga dapat menjadi sebuah program yang disajikan dengan menggunakan antar muka *web*. *JavaScript* merupakan bahasa *script*, yaitu bahasa yang tidak memerlukan kompiler untuk dapat

menjalankannya, tetapi cukup dengan Interpreter. Tidak perlu ada proses kompilasi terlebih dahulu agar program dapat dijalankan. (Yani et al., 2018)



**Gambar 2.3 Lambang Java Script**

## **2.12 Android Studio**

*Android studio* adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi *Android* dan bersifat open source atau gratis. Peluncuran *Android Studio* ini diumumkan oleh *Google* pada 16 mei 2013 pada event *Google I/O Conference* untuk tahun 2013. Sejak saat itu, *Android Studio* menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi *Android*. (Andi, 2015)



**Gambar 2.4 Android Studio**

*Android studio* sendiri dikembangkan berdasarkan *IntelliJ IDEA* yang mirip dengan *Eclipse* disertai dengan *ADT plugin (Android Development Tools)*. *Android studio* memiliki fitur :

- a. Projek berbasis pada *Gradle Build*

- b. *Refactory* dan pembenahan bug yang cepat
- c. *Tools* baru yang bernama "*Lint*" diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung *Proguard* dan *App-signing* untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah
- f. Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan.

### 2.13 SmartPhone

Sebuah perangkat yang memungkinkan untuk melakukan komunikasi (seperti telp atau sms) juga di dalamnya terdapat fungsi PDA (*Personal Digital Assistant*) dan berkemampuan seperti layaknya komputer. Kata "smartphone" didefinisikan sebagai "ponsel yang menggabungkan (*Personal Digital Assistant*) PDA" oleh Amerika Kamus *Oxford*. Dalam pengertian singkat *smartphone* adalah sebuah perangkat yang dapat digunakan untuk melakukan komunikasi seperti menelpon atau mengirim pesan singkat, juga di dalamnya terdapat fungsi *personal digital assistant* dan berkemampuan seperti layaknya komputer dan kemampuan mengolah pesan pada *smartphone*.(Hestyleta, 2009)



**Gambar 2.5 Smartphone**





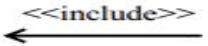

## 2.14 UML ( Unified Modeling Language)

UML adalah salah satu *tool/model* untuk merancang pengembangan *software* yang berbasis *object-oriented*. UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen yang diperlukan dalam sistem *software*. (Sonata, 2019)

## 2.15 Use Case Diagram

Merupakan diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* diagram terdiri dari sebuah aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, sistem lain, ataupun yang berinteraksi dengan sistem. (Kurniawan, T. Bayu, 2020)


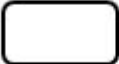




**Tabel 2.4 Simbol Use Case Diagram**

| Simbol  | Keterangan  |
|---|---|
|  | Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>                           |
|  | <i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor   |
|  | <i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>  |
|  | <i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>                          |
|  | Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya                        |
|  | Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi |

## 2.16 Activity Diagram

*Activity diagram* menggambarkan berbagai aliran aktivitas dalam sistem yang sedang di rancang, bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. (Kurniawan, T. Bayu, 2020)

**Tabel 2.5 Simbol Activity Diagram**

| Simbol  | Nama                   | Keterangan   |
|---|------------------------|--|
|    | Status awal            | Sebuah diagram aktivitas memiliki sebuah status awal.  |
|   | Aktivitas              | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.               |
|  | Percabangan / Decision | Percabangan dimana ada pilihan aktivitas yang lebih dari satu.                               |
|  | Penggabungan / Join    | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.          |
|  | Status Akhir           | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir    |
|  | Swimlane               | Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

## 2.17 Class Diagram

*Class diagram* merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam


sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. *class diagram* menggambarkan interaksi dan relasi antar kelas yang ada di dalam suatu sistem. Kelas memiliki atribut dan metode. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Atribut dan metode dapat memiliki salah satu sifat sebagai berikut: (Robinson, 1982)

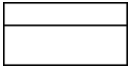





1. *Private*, tidak dapat dipanggil dari luar kelas yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh kelas yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.

*Class diagram* menggambarkan relasi atau hubungan antar kelas dari sebuah sistem. Berikut ini beberapa gambaran relasi yang ada dalam *class diagram*:


1. *Association* Hubungan antar class yang statis. *Class* yang mempunyai relasi asosiasi menggunakan class lain sebagai atribut pada dirinya.
2. *Aggregation* Relasi yang membuat *class* yang saling terikat satu sama lain namun tidak terlalu berkegantungan.
3. *Composition* Relasi agregasi dengan mengikat satu sama lain dengan ikatan yang sangat kuat dan saling berkegantungan.
4. *Dependency* Hubungan antar *class* dimana *class* yang memiliki relasi *dependency* menggunakan *class* lain sebagai atribut pada *method*.
5. *Realization* Hubungan antar *class* dimana sebuah *class* memiliki keharusan untuk mengikuti aturan yang ditetapkan *class* lainnya.

**Tabel 2.6 Simbol Diagram Class**

| No. | Simbol  | Nama                        | Keterangan   |
|-----|---|-----------------------------|--|
| 1.  |  | <i>Directed Association</i> | Simbol yang berbentuk tanda panah yang berfungsi untuk relasi antar <i>class</i> . |

|    |   |                     |  |
|----|---|---------------------|--|
| 2. |    | <i>Class</i>        | Simbol yang berbentuk kotak-kotak yang berbagi antara atribut serta operasi yang sama.   |
| 3. |    | <i>Interface</i>    | Simbol yang berbentuk <i>oval</i> garis putus-putus yang berfungsi untuk deskripsi dari urutan aksi-aksi yang akan ditampilkan pada sistem.                                    |
| 4. |    | <i>Generalisasi</i> | Simbol yang berbentuk garis panah putus-putus horizontal ke kiri yang berfungsi untuk <i>class</i> yang umum.  |
| 5. |  | <i>Dependency</i>   | Simbol yang berbentuk garis panah putus-putus horizontal ke kanan yang berfungsi untuk kebergantungan antar <i>class</i> .   |
| 6. |  | <i>Association</i>  | Simbol yang berbentuk garis horizontal yang berfungsi untuk asosiasi atau penggabungan setiap <i>class</i> yang berhubungan.   |
| 7. |  | <i>Agregation</i>   | Simbol yang berbentuk seperti belah ketupat yang berwarna hitam berfungsi untuk menyatakan bahwa suatu <i>class</i> menjadi bagian yang tidak wajib atas <i>class</i> lainnya. |



|    |   |                  |  |
|----|---|------------------|--|
| 8. |  | <i>Composite</i> | Simbol yang berbentuk seperti belah ketupat yang berwarna hitam berfungsi untuk menyatakan bahwa suatu <i>class</i> menjadi bagian yang wajib atas <i>class</i> lainnya. |
|----|---|------------------|--|

