# BAB II
# STUDI LITERATUR

## 2.1 Tinjauan Pustaka

### 2.1.1 Citra

Citra adalah fungsi konstan menurut intensitas cahaya yang terdapat pada bidang dua dimensi yang dianggap menjadi gambaran menggunakan bentuk segi empat berformat horizontal dan vertikal yang mempunyai rona dan merupakan representasi digital. Sumber cahaya menyinari objek & memantulkannya balik sebagian menurut berkas cahaya tadi. Pemantulan cahaya ini ditangkap sang indra-indra optik, contohnya mata manusia, kamera, pemindai (*scanner*), dsb., sebagai akibatnya bayangan objek gambaran tadi terekam. Definisi citra merupakan suatu representasi, kemiripan, atau imitasi menurut suatu benda, secara harfiah, gambaran (*image*) merupakan gambar dalam bidang 2 dimensi. Citra menjadi keluaran suatu sistem perekaman, data bisa bersifat optik berupa foto, bersifat analog berupa frekuensi-frekuensi video misalnya gambar dalam monitor televisi, atau bersifat digital yang bisa eksklusif disimpan dalam suatu media penyimpanan[8], [9].

#### 2.1.1.1 Citra Digital

Citra digital adalah gambar 2 dimensi didapatkan berdasarkan gambar analog 2 dimensi yang berkelaluan sebagai gambar diskrit melalui proses sampling. Proses perubahan gambaran sebagai gambaran digital dinamakan digitasi. Digitasi adalah proses mengganti sebuah gambar, teks, atau bunyi berdasarkan benda yang bisa dilihat ke pada data elektro & bisa disimpan dan diproses buat keperluan lainnya. Dalam konteks lebih luas, pengolahan gambaran digital lebih mengacu dalam pemrosesan setiap 2 data dimensi. Citra digital dipetakan sebagai bentuk grid & elemen piksel berbentuk matriks dua dimensi. Setiap piksel tadi mempunyai data yang mempresentasikan saluran warna. Angka dalam setiap piksel disimpan secara berurutan dengan komputer & tak jarang dikurangi buat keperluan kompresi juga pengolahan tertentu. Sebuah matriks yang terdiri dari M kolom N baris dapat mewakili dari citra digital, di mana piksel (*pixel = picture element*) merupakan perpotongan antara kolom dan baris, yaitu elemen terkecil dari sebuah citra.

Koordinat dan warna atau intensitas merupakan parameter dari Piksel. Pada koordinat $(x, y)$ terdapat nilai adalah $f(x, y)$, yaitu nilai intensitas atau warna dari piksel di titik itu. Oleh karena itu, citra dapat dituliskan ke dalam sebuah Persamaan 2.1 matriks :

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,M-1) \\ \vdots & \vdots & \vdots \\ f(N-1,0 & \cdots & f(N-1,M-1) \end{bmatrix} \quad (2.1)$$

Berdasarkan rumus tersebut, suatu citra $f(x, y)$ dapat dituliskan ke dalam fungsi matematis seperti berikut ini :

$$0 \leq x \leq M-1$$

$$0 \leq x \leq N-1$$

$$0 \leq f(x,y) \leq G-1$$

Di mana :

M = jumlah piksel baris pada array citra

N = jumlah piksel kolom pada array citra

G = nilai skala keabuan (*grayscale*)

### 2.1.1.2 Pengolahan Citra Digital

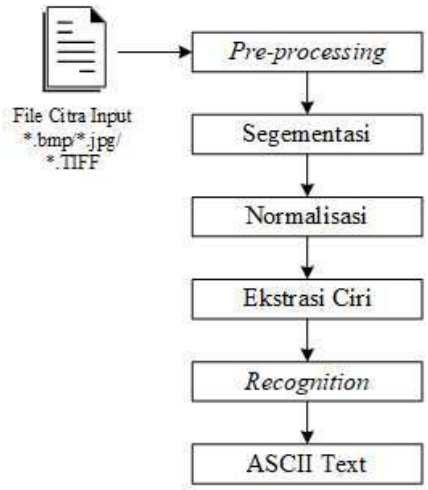
Pengolahan citra digital merupakan studi bidang ilmu tentang bagaimana teknik pengolahan sebuah citra. Dengan melalui pemrosesan citra, yaitu untuk mencapai tujuan untuk memperbaiki kualitas suatu citra sehingga dapat diinterpretasi dengan mudah oleh manusia atau sebuah mesin dalam hal ini komputer.

### 2.1.2 *Optical Character Recognition*

*Optical Character Recognition (OCR)* dalam arti luas adalah cabang dari kecerdasan buatan dan visi komputer. OCR adalah aplikasi yang dapat digunakan untuk mengidentifikasi gambar huruf dan angka yang akan diubah menjadi file teks[3]. Sistem OCR ini dapat meningkatkan keluwesan atau kemampuan dan kecerdasan komputer, dengan kemampuan mengenal huruf akan sangat membantu dalam mendigitalkan informasi dan pengetahuan dari bentuk cetakan, misalnya dalam pembuatan koleksi perpustakaan digital, koleksi sastra, dll. Sistem OCR memerlukan kalibrasi untuk mengidentifikasi teks dengan membaca *font* yang spesifik. Pada tahap awal sistem harus diprogram dengan citra dari setiap karakter yang ada agar dapat mengenalinya dan menyimpannya pada *database*. Dengan

semakin banyak karakter yang dikenali melalui perbanyak citra karakter akan meningkatkan akurasi pada pengenalan karakter.

Secara umum proses OCR dapat dilihat pada Gambar 1 berikut :




**Gambar 2.1.** Skema Umum Proses OCR.

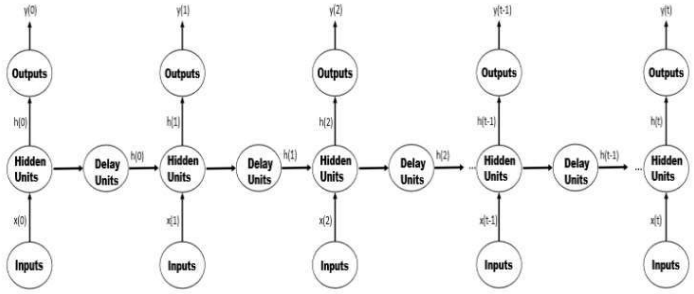
Skema umum dari proses OCR digambarkan pada Gambar 2.1, file input berupa file citra digital dengan format *.bmp atau *.jpg atau *.TIFF. Dilanjutkan dengan proses *Pre-processing,* Proses ini adalah proses buat menghilangkan bagian-bagian yang nir dibutuhkan dalam gambaran masukan buat proses selanjutnya yaitu segmentasi. Segmentasi merupakan proses menerima area atau objek yang diinginkan pada suatu gambaran menggunakan memisahkan area atau objek berdasarkan latar belakang. Sehingga bisa memisahkan wilayah pengamatan (region) dalam setiap karakter yangg terdeteksi. Normalisasi merupakan proses membarui dimensi area setiap karakter & ketebalan karakter. Tujuan berdasarkan normalisasi gambaran merupakan buat mengurangi resolusi gambaran yang nir bermanfaat selama proses sosialisasi gambaran & jua buat mempertinggi akurasi sosialisasi. Proses yang dipakai pada termin normalisasi ini merupakan proses penskalaan gambaran. Proses selanjutnya adalah ekstraksi ciri (*feature extraction).* Ekstraksi ciri adalah proses untuk mengambil ciri - ciri tertentu dari karakter yang diamati. Karakteristik ini digunakan dalam mendeskripsikan sebuah objek atau atribut yang ada. Pada proses terakhir yaitu *Recognition*, *Recognition* adalah proses

untuk mengenali karakter yang diamati dengan membandingkan karakteristik karakter yang ada di basis data.

Setelah proses-proses di atas selesai dilakukan, maka OCR akan menghasilkan keluaran atau hasil dari pengenalan karakter baik numerik maupun huruf. Saat ini sudah ada OCR dengan penambahan metode klasifikasi lain guna meningkatkan nilai akurasinya. Salah satunya adalah LSTM yang akan dibahas selanjutnya.

### 2.1.3 *Recurrent Neural Network*

Recurrent Neural Network (RNN) merupakan bagian dari jaringan syaraf tiruan untuk pemrosesan data sekuensial. Bentuk RNN mirip dengan bentuk yang dimiliki *feedforward* (ANN). Namun tidak seperti *feedforward* dimana aliran data bersifat satu arah, sedangkan aliran yang ada di RNN bisa berputar atau kembali ke *layer* sebelumnya.



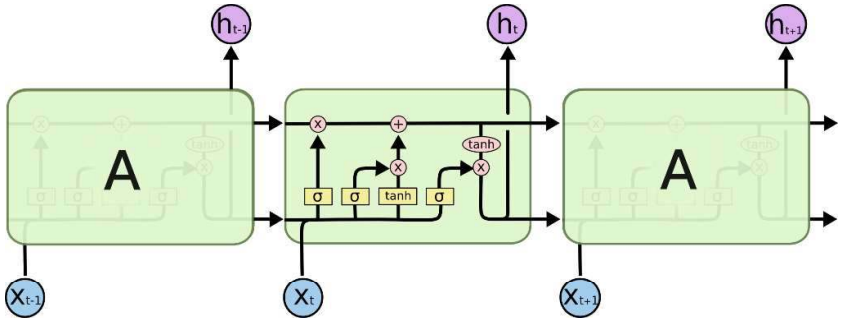

**Gambar 2.2.** Diagram pada RNN[10].

Bentuk RNN yang ditunjukkan pada Gambar 2.2 terdiri dari banyak *input*, *hidden state* dan *output*. Pada Satu *time step* mengandung satu *input*, satu *hidden state* dan satu *output*. Jumlah *time step* tidak dibatasi. Setiap *hidden state* terhubung dengan *hidden state* milik *time step* sebelumnya dan inilah yang menjadi sumber dari kelebihan yang dimiliki oleh RNN. Sambungan *hidden state* ini bisa mengalirkan informasi dari data sebelum-sebelumnya ke data selanjutnya sehingga pada setiap proses prediksi yang dilakukan, informasi mengenai masa lalu selalu dipertimbangkan.[10].

### 2.1.4 *Long Short-Term Memory*

*Long Short Term Memory networks* (LSTMs) merupakan sebuah algoritma evolusi dari arsitektur RNN, dimana pertama kali diperkenalkan oleh Hochreiter &

Schmidhuber pada tahun 1997[11]. Hingga penelitian ini dilakukan masih banyak para peneliti yang terus mengembangkan arsitektur LSTM di berbagai bidang seperti dalam bidang *speech recognition* dan *forecasting*. LSTM dapat dikatakan sebagai variasi dari RNN, dimana RNN merupakan bagian dari jaringan syaraf tiruan untuk pemrosesan data sekuensial. Pada RNN, memori yang tersimpan sebelumnya tidak berguna karena dengan seiringnya waktu, memori tersebut akan ditimpa dengan memori yang baru. Berbeda dengan RNN, LSTM melakukan pembelajaran jangka panjang. LSTM dapat bekerja dengan sangat baik pada berbagai macam masalah dan dapat mengingat informasi untuk jangka waktu yang lama macam masalah dan dapat mengingat informasi untuk jangka waktu yang lama.




**Gambar 2.3.** *Flowchart Memory Cell* pada LSTMs[12].

#### 2.1.4.1 *Memory Cells dan Gate Units*

LSTMs memiliki *memory cell* dan *gate units* pada setiap *neurons* nya yang berfungsi untuk mengatur memori dalam setiap *neurons.* Pada Gambar 2.3 menjelaskan bagaimana alur kerja *memory cells* pada setiap *neurons* LSTMs bekerja. Terdapat empat proses fungsi aktivasi pada setiap masukan pada *neurons* yang selanjutnya disebut sebagai *gates units*. *Gates units* tersebut ialah *forget gates*, *input gates*, *cell gates*, dan *output gates*.

Pada *forget gates* informasi pada setiap data masukan akan diolah dan dipilih data mana saja yang akan disimpan atau dibuang pada *memory cells*. Fungsi aktivasi yang digunakan pada *forget gates* ini adalah fungsi aktivasi sigmoid. Dimana hasil keluarannya antara 0 dan 1. Jika keluarannya adalah 1 maka semua data akan disimpan dan sebaliknya jika keluarannya 0 maka semua data akan dibuang. Dengan rumus pada persamaan 2.2 sebagai berikut :

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (2.2)$$

Pada *input gates* terdapat dua *gates* yang akan dilaksanakan, pertama akan diputuskan nilai mana yang akan diperbarui menggunakan fungsi aktivasi sigmoid. Selanjutnya fungsi aktivasi *tanh* akan membuat *vector* nilai baru yang akan disimpan pada *memory cell*. Dengan rumus pada persamaan 2.3 dan 2.4 sebagai berikut :

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (2.3)$$

$$\check{c}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \quad (2.4)$$

Pada *cell gates* akan mengganti nilai pada *memory cell* sebelumnya dengan nilai *memory cell* yang baru. Dimana nilai ini didapatkan dari menggabungkan nilai yang terdapat pada *forget gate* dan *input gate*. Dengan rumus pada persamaan 2.5 sebagai berikut :

$$c_t = f_t * c_{t-1} + i_t * \check{c}_t \quad (2.5)$$

Pada *output gates* terdapat dua *gates* yang akan dilaksanakan, pertama akan diputuskan nilai pada bagian *memory cell* mana yang akan dikeluarkan dengan menggunakan fungsi aktivasi sigmoid. Selanjutnya akan ditempatkan nilai pada *memory cell* dengan menggunakan fungsi aktivasi *tanh*. Terakhir kedua *gates* tersebut dikalikan sehingga menghasilkan nilai yang akan dikeluarkan. Dengan rumus pada persamaan 2.6 dan 2.7 sebagai berikut :

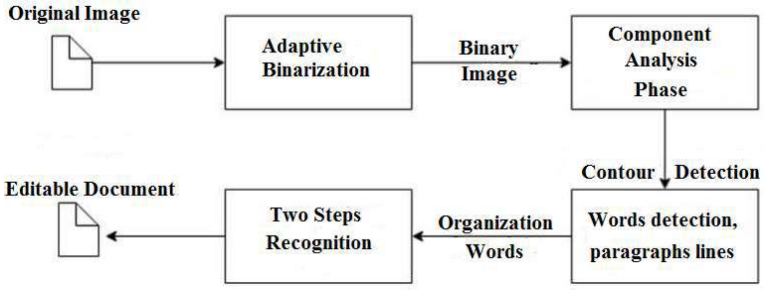
$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (2.6)$$

$$h_t = o_t tanh(c_t) \quad (2.7)$$

### 2.1.5 *Tesseract*

Tesseract adalah mesin pengenalan karakter optik berbasis open source. Tesseract dimulai sebagai proyek penelitian PhD di HP Labs, Bristol yang kemudian dikembangkan lebih lanjut di HP hingga tahun 1994, lalu dimodifikasi dan ditingkatkan pada tahun 1995 dengan akurasi yang lebih besar[13]. Pada akhir tahun 2005, HP merilis Tesseract untuk open source dan hingga sekarang tersedia.

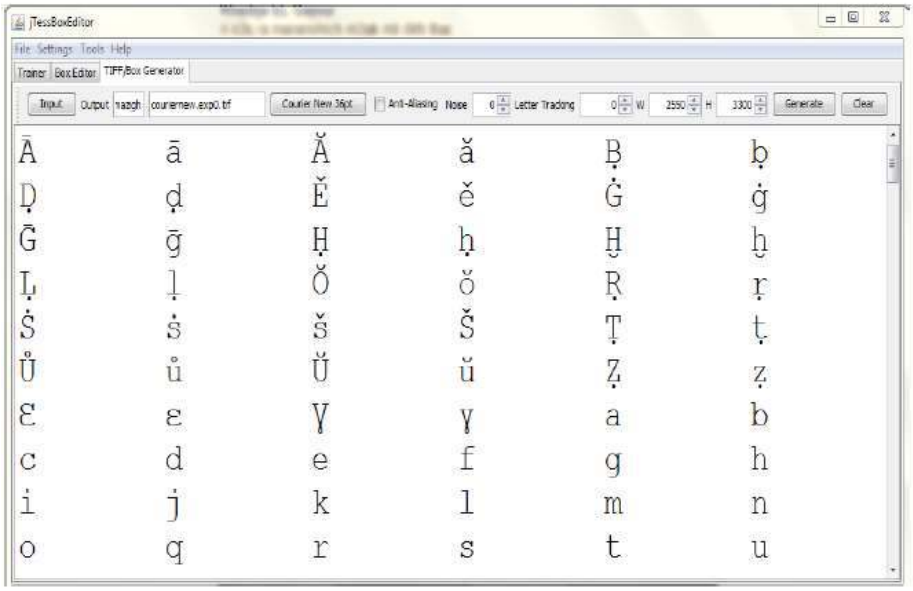
Arsitektur umum Tesseract ditunjukkan seperti pada Gambar 4. Terdapat beberapa langkah, yiatu *Adaptive Thresholding* yang mengubah gambar menjadi versi biner, *Component Analysis Phase* yang digunakan untuk mengekstrak blok teks dalam dokumen, *Word detection & paragraph lines* yaitu dari setiap baris dan teks terdeteksi dibagi menjadi kata-kata terpisah, selanjutnya kata dipisah menjadi karakter diekstraksi dari tahap sebelumnya. Pengenalan teks ini kemudian dimulai sebagai proses dua langkah. Pada tahap pertama, pengenalan kata dilakukan menggunakan *static classifier*. Setiap kata yang terdeteksi dengan baik akan diteruskan ke klasifikasi adaptif untuk dijadikan sebagai data pelatihan. Tahap kedua dijalankan di atas halaman, menggunakan penggolong adaptif yang baru dipelajari di mana kata-kata yang tidak cukup terdeteksi menjadi bisa terdeteksi lagi.




**Gambar 2.4.** Arsitektur Umum *Tesseract*

### 2.1.6 *jTessBoxEditor*

Eksplorasi dan akuisisi data dapat dilakukan dengan menggunakan *tools* ini. Langkah pertama adalah menghasilkan korpus yang terdiri dari karakter berbeda menggunakan *jTessBoxEditor*. *jTessBoxEditor* adalah editor dan pelatih untuk Tesseract OCR. Dengan menyediakan pengeditan data kotak untuk format Tesseract 2.0x dan 3.0x, dan penuh otomatisasi pelatihan Tesseract. Dapat membaca gambar format gambar umum, termasuk TIFF multi-halaman. Program ini membutuhkan *Java Runtime Environment* 7 atau yang lebih baru. Antarmuka ini (Gambar 5) memungkinkan penambahan file teks berisi karakter untuk dilatih, tentukan fontnya diinginkan.[14]




**Gambar 2.5.** *jTessBoxEditor*[14]

### 2.1.7 *Confusion Matrix*

*Confusion matrix* memiliki 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN).

**Tabel 2.1.** *Confusion Matrix*

| | | Sebenarnya | |
|---|---|---|---|
| | | Positif | Negatif |
| Prediksi | Positif | TP | FP |
| | Negatif | FN | TN |

Pada Tabel 2.1 di atas menunjukkan nilai *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif. Sementara itu, *True Positive* (TP) merupakan data positif yang terdeteksi benar. *False Negative* (FN) merupakan kebalikan dari *True Positive*, sehingga data positif, namun terdeteksi sebagai data negatif[15]. *Precision* adalah data yang diambil berdasarkan informasi yang kurang. Dalam klasifikasi biner, presisi dapat dibuat sama dengan nilai prediksi positif. Berikut ini adalah persamaan 2.8 *precision*.

$$precision = \frac{TP}{TP+FP} \times 100\% \qquad (2.8)$$

*Recall* adalah data penghapusan yang berhasil diambil dari data yang relevan dengan kueri. Dalam klasifikasi biner, *recall* dikenal sebagai sensitivitas. Munculnya data relevan yang diambil adalah menyetujui dengan query dapat dilihat dengan recall. Berikut ini adalah persamaan 2.9 *recall.*

$$recall = \frac{TP}{TP+FN} \times 100\% \quad (2.9)$$

Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai. Berikut ini adalah persamaan 2.10 akurasi.

$$akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2.10)$$

### 2.1.8 Android

Android merupakan sebuah sistem operasi berbasis Linux sebagai kernelnya yang dipergunakan untuk mengelola sumber daya perangkat keras baik untuk ponsel, ponsel pintar dan juga PC tablet. Secara umum Android adalah *platform* yang *open source* bagi para *programmer* untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai piranti bergerak. Oleh karena bersifat *open source*, sistem operasi *mobile* ini berkembang begitu pesat di era teknologi. Android juga merupakan sistem operasi seluler berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi pengembang untuk membuat aplikasi mereka. Sistem operasi dasar Android berada di bawah GNU General Public License Version 2 (GPLv2)[2].

## 2.2 Tinjauan Studi

Pengolahan citra digital khususnya pada sistem OCR (*Optical Character Recognition*) terus dikembangkan dalam pemrosesan citra tulisan dengan berbagai metode klasifikasi karakter. Beberapa metode dan teknologi yang dapat digunakan pada aplikasi OCR antara lain SVM (*Support Vector Mechine*), CNN (*Convolutional Neural Network)*, MLP (*Multilayer Perceptron*). Telah banyak dilakukan penelitian sebelumnya tentang OCR oleh para *researcher* baik terhadap metode atau berbagai macam *object* yang digunakan. Terdapat penelitian dengan mengombinasikan *Artificial Neural Network* dengan metode *Feature Vectors* yang menghasilkan keakuratan 85.83% yaitu 209 karakter dari 360 benar dikenali,

dengan 10 *simple font style* dan 75% yaitu 540 dari 720 benar dikenali ketika tinggakatan *font style* yang dinaikkan. Dengan ANN dapat menghasilkan akurasi cukup baik meskipun tidak ada pada *data training* seperti pada *Simple Font Style*. Namun memiliki kekurangan, seperti sering terjadi *false recognition* pada karakter I dan J, K dan R, O dan Q, B dan 8. Menggunakan *Walsh-Hadamard Transforms* (WHT) yang kemudian sebagai *input* NN. WHT yaitu teknik transformasi ortogonal non-sinusoidal yang menguraikan sinyal menjadi serangkaian fungsi dasar. Fungsi-fungsi dasar ini adalah fungsi *Walsh*, yang merupakan gelombang persegi atau persegi dengan nilai +1 atau –1 [16].

Ekstrasi informasi yang terdapat pada *various unstructured documents* memiliki kendala yang dihadapi pada pendeteksian karakter dapat diselesaikan dengan penerapan *Artificial Neural Network* pada *Optical Character Recognition* dengan mengunakan MLP (*Multilayer Perceptron)* dan menghasilkan tingkat akurasi cukup tinggi yang diklaim oleh peneliti[3].

Penelitian dengan metodologi yang mencakup penerapan metode segmentasi yang memberi label huruf pada gambar dengan sangat efisien, dan kemudian rata-rata koordinat setiap huruf dalam gambar digunakan setelah penerapan metode segmentasi untuk mengurutkan huruf dengan benar, Selain itu, algoritma ini meliputi pengenalan metode deteksi garis baru berdasarkan rata-rata label, dan menggunakan *Neural Network* sebagai *autoencoder*-nya. Dengan hasil akhir ditemukan kekurangan yaitu tidak dapat mendeteksi tanda baca dan spasi namun dapat mengenali garis baru[17].

Metode *Artificial Neural Network* dengan *Static Image Properties* terdapat dua sub-metode didalamnya yaitu *Online* dan *Offline*. Metode *Offline* merupakan Ekstraksi Fitur, *Clustering* dan Pencocokan Pola. Sedangkan *Online* KNN *Classifier* dan *direction-based algorithm.* Dengan *Nerual Network* maka akan memiliki kecepatan yang tinggi dan mampu belajar (*learning)* lebih cepat terhadap perubahan tetapi memiliki kekurangan yaitu tidak terlalu efektif pada kasus teks/*font* kotor, kemiringan tulisan, perbedaan *font*. Kelebihan pada penelitian ini, terdapat *Skewness*, yaitu *Rotate* pada gambar yang dipetakan pada kertas yang dipindai untuk sistem pengenalan karakter[18].

Sebuah penelitian *Character Recognition* pada dokumen yang rusak akibat banjir dengan menerapkan *Artificial Neural Network* dengan metode *Perceptron*. *Image* dikonversi ke Biner (0 dan 1). Setiap karakter di *training* dan menghasilkan data pada Daftar Karakter, yang nantinya akan digunakan sebagai *matching base* ketika *testing* berlangsung. Sistem yang dibuat berjalan dengan syarat karakter yang akan dikenali sudah dalam bentuk digital/*image*. Kekurangan dari penelitian ini adalah tidak terdapat metode pengambilan data gambar dari hardcopy. Tidak ada tabel hasil *testing* sehingga tidak dapat mengukur tingkat ke akurasiannya[19].

Dalam menghadapi permasalahan seperti *Low Quality Image*, *Unclear Words*, *Typical Font*, *Image have a lot of colour*, *Blurred picture*. Metode MSER (*Maximally Stable External Region*) yang diterapkan pada *Convolutional Neural Network* untuk menanggulangi masalah-masalah tersebut dan menghasilkan 92.31% akurasi[20].

Penerapan *Convolutional Neural Network* dalam pengembangan dari sisi arsitektur dan banyaknya lapisan yang digunakan pada jaringan. Pembuatan arsitektur yang baik sangat berpengaruh pada klasifikasi citra untuk semua kategori. Tahun 2012 penerapan *Deep Learning* dengan metode CNN dipopulerkan dengan arsitektur AlexNet yang diuji dengan *dataset* ImageNet. Penelitian ini menggunakan *dataset* ImageNet LSVRC-2010 ke dalam 1000 classes. Arsitektur yang dibuat oleh Alex Krizhevsky menunjukkan hasil yang sangat signifikan pada *testing test* dengan *test error* sebesar 17%. Hasil ini sudah dapat dinilai sangat baik karena citra yang digunakan pada *dataset* sangatlah banyak[21].

Salah satu metode klasifikasi yang terus dikembangkan dalam proses pengenalan karakter yakni *Long Short Term Memory Networks* (LSTMs). Metode ini memiliki pendekatan algoritma yang sederhana dengan akurasi baik, sehingga memungkinkan untuk diimplementasikan dan merupakan pengembangan dari *Recurent Neural Network* (RNN) yang baik dibidang pengenalan karakter.

Menurut penelitian-penelitian yang telah diuraikan mengenai *image classification* metode CNN memiliki akurasi yang lebih baik dari SVM dan MLP namun dalam pemanfaatannya masih terdapat beberapa kekurangan pada CNN yaitu dalam menangani citra dengan banyak teks. CNN memerlukan banyak sumber daya yang

diperlukan untuk mengolah citra sedangkan pada LSTMs menunjukkan bahwa metode tersebut lebih baik dalam menghemat sumber daya maupun kecepatannya mengenali banyak karakter, jenis RNN yang dimaksud adalah LSTMs. Sehingga penelitian ini menerapkan metode LSTMs. Berikut merupakan rangkuman dari penelitian-penelitian sebelumnya yang dapat dilihat pada Tabel 2.2.

**Tabel 2.2.** Rangkuman penelitian terkait.

| No | Penulis | Judul | Hasil |
|---|---|---|---|
| 1. | Vivek Shrivastava, dkk. (2012) | *Artificial Neural Network Based Optical Character Recognition* | ANN dilengkapi *feature vector* menghasilkan keakuratan 85.83% pada 10 *simple font style* dan 75% pada *font style* yang ditingkatkan. |
| 2. | Anamika Bhaduri, dkk. (2016) | *Optical Character Recognition Using Artificial Neural Network* | ANN pada OCR dengan menggunakan MLP (*Multilayer Perceptron).* Tidak memiliki pengukuran akurasi selama penelitian. |
| 3. | Ammar A. Radhi (2017) | *Text Recognition using Image Segmentation and Neural Network* | *Neural Network* sebagai *autoencoder,* dengan hasil akhir ditemukan kekurangan yaitu tidak dapat mendeteksi tanda baca dan spasi namun dapat mengenali garis baru. Tidak memiliki pengukuran akurasi selama penelitian. |
| 4. | Sunil Kumar, dkk. (2016) | *Scene Text Recognition using Artificial Neural Network: A Survey* | ANN menghasilkan kecepatan *learning* tidak efektif pada kasus tulisan/*font* kotor, kemiringan tulisan, perbedaan *font.* Tidak memiliki pengukuran akurasi selama penelitian. |
| 5. | Arif Setiawan, dkk. (2012) | Analisa Sistem Pengenalan Karakter Menggunakan Jaringan Syaraf Tiruan Untuk Pembacaan | Sistem dapat berjalan dengan syarat karakter yang akan dikenali sudah dalam bentuk digital/*image,* sehingga tidak ada tahap |

| No | Penulis | Judul | Hasil |
|---|---|---|---|
| | | Dokumen Yang Rusak Karena Banjir | akuisisi citra secara langsung. Tidak memiliki pengukuran akurasi selama penelitian. |
| 6. | C.P. Chaithanya (2019) | *Automatic Text Detection and Classification in Natural Images* | CNN menghasilkan akurasi 92.31%. |
| 7. | Alex Krizhevsky, dkk. (2012) | *ImageNet Classification with Deep Convolutional Neural Networks* | CNN dilengkapi arsitektur AlexNet menghasilkan *error rate* 17%. |
| 8. | Andi Ariyandi (2021) | *Image to Text* dengan *Inp*ut Kamera *Mobile* dengan Metode *Recurrent Neural Network* | Pada penelitian ini akan dilakukan penerapan aplikasi OCR berbasis *mobile* menggunakan salah satu jenis RNN yaitu LSTMs guna mengonversi teks fisik ke teks digital serta dilakukan pengujian tingkat keakuratan dari metode ini. |