

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Menurut undang-undang Nomor 20 Tahun 2003 pasal 57 (ayat 1) untuk mengendalikan mutu pendidikan perlunya dilakukan evaluasi. Evaluasi ini perlu dilakukan karena memiliki peran untuk memantau proses, kemajuan, dan perbaikan hasil belajar peserta didik secara berkesinambungan seperti tercantum pada Undang-undang Nomor 20 Tahun 2003 pasal 58 (ayat 1) (P. R., 2003). Proses evaluasi peserta didik dapat dilakukan dengan cara mengadakan ulangan harian, Ulangan Tengah Semester, dan Ujian Akhir Semester.

Pada masa kini untuk mempermudah proses pemeriksaan lembar ujian dapat digunakan Lembar Jawaban Komputer (LJK). Menurut Wibowo (2013) LJK dibaca dan diproses menggunakan perangkat yang disebut Optical Mark Reader (OMR) yaitu pemindai khusus yang dapat mendeteksi tanda pada lembar jawab. Tanda pada lembar jawaban tersebut berupa bulatan hitam yang dibuat oleh pengisi jawaban menggunakan pensil, yang biasanya menggunakan jenis pensil 2B. OMR tersebut dapat mendeteksi letak tanda menggunakan inframerah lalu menyimpan informasinya dan diproses lebih lanjut, yaitu dapat berupa penghitungan jumlah jawaban tertentu atau dapat juga dilakukan pencocokan jawaban dengan kunci jawaban yang telah tersedia. OMR *scanner* ini telah banyak digunakan pada evaluasi lembar ujian, penilaian kehadiran otomatis, pemungutan suara, survei komunitas, dll. (Patel, Zaid, & Students, 2017).

Namun ternyata pada praktek penggunaannya menurut Wibowo (2013), teknologi OMR *scanner* masih memiliki beberapa kekurangan. Seperti pencetakan lembar jawab komputer serta pembelian *scanner* OMR yang membutuhkan biaya yang besar sehingga hanya beberapa pihak yang mampu menggunakannya. OMR *scanner* pun hanya mampu memeriksa lembar jawaban komputer dengan ketebalan tertentu. Selain itu apabila lembar ujian/LJK sobek (walaupun hanya sedikit) atau terlipat dapat menyebabkan lembar ujian/LJK tidak dapat diperiksa secara keseluruhan oleh OMR *scanner* (Patel & Zaid, 2017). Kemudian menurut Rahmat

dalam Wibowo (2013) walaupun dibutuhkan, sistem dengan teknologi *scanner* OMR tidak tepat untuk diterapkan pada institusi berskala kecil.

Walaupun LJK telah biasa digunakan oleh berbagai Universitas namun ternyata masih ada kasus kesalahan dalam pengisian LJK tersebut. Salah satu kasus kesalahan pengisian lembar jawaban ini terjadi di Universitas Terbuka menurut Aprijani & Sufandi (2011) berdasarkan data yang diperoleh dari Pusat Pengujian UT, pada masa registrasi 2010, terdapat 34.326 kasus (2,91%) kesalahan dalam menghitamkan bulatan pada LJK.

Mengingat dari beberapa kekurangan tersebut dibutuhkan aplikasi pemeriksa lembar ujian yang mampu menjadi alternatif murah dan mudah untuk digunakan pemeriksa dan pengisi lembar ujian. Beberapa masalah utama yang terjadi pada penggunaan OMR *scanner* untuk memeriksa LJK/lembar ujian adalah lembar ujian yang terlipat atau sobek yang tidak dapat diperiksa dan masih banyaknya peserta ujian yang mengisi lembar ujian dengan tidak tepat.

Untuk mengatasi permasalahan lembar jawaban yang tidak dapat diperiksa karena sobek atau terlipat salah satu cara yang dapat dilakukan adalah dengan melakukan pengenalan pola dari citra gambar lembar ujian tersebut. Pengenalan pola adalah proses pengenalan suatu objek dengan menggunakan berbagai metode dimana dalam proses pengenalanya memiliki tingkat akurasi yang tinggi. Memiliki tingkat akurasi yang tinggi mengandung pengertian bahwa suatu objek yang secara manual (oleh manusia) tidak dapat dikenali tetapi bila menggunakan salah satu metode pengenalan yang diaplikasikan pada komputer dapat dikenali (Wijaya & Kanata, 2004).

Kemudian sebagai alternatif untuk meminimalisir kesalahan pada proses pengisian lembar ujian dapat dilakukan penggantian proses pembulatan dengan tulisan tangan, karena model tulisan tangan sudah biasa digunakan pada keseharian (Sam'ani & Qamaruzzaman, 2017). Tulisan tangan ini dapat dikenali dengan pengenalan pola yang biasa disebut juga dengan pengenalan tulisan tangan. Pengenalan Tulisan Tangan adalah proses perubahan suatu bahasa yang dihadirkan dalam bentuk ruang melalui tulisan menjadi suatu representasi symbol (Plamondon & Srihari, 2004). Dalam kurun waktu yang cukup lama penelitian dalam bidang pengenalan tulisan tangan telah berkembang banyak. Karena semakin seringnya

model tulisan tangan dipergunakan dalam kehidupan sehari-hari penelitian tersebut dilakukan, seperti identifikasi dokumen-dokumen penting, bukti pengesahan dalam dunia perbankan, dan lain sebagainya. Permasalahan pada dunia pengenalan tulisan ini meliputi pengenalan karakter (huruf), pengenalan angka, pengenalan gestur, pengenalan tanda tangan, dan lain sebagainya. Dalam memecahkan permasalahan pada tulisan tangan tersebut akan berkaitan dengan pengenalan pola yang nantinya bertujuan untuk memilih dan menghasilkan pola-pola yang mampu digunakan pada proses identifikasi (Sam'ani & Qamaruzzaman, 2017).

Salah satu metode untuk mengenali tulisan tangan tersebut adalah metode jaringan syaraf tiruan *Convolutional Neural Network*. *Convolutional Neural Network* (CNN) merupakan perbaikan dari Jaringan Syaraf Umpan Maju Berlapis Banyak dalam menangani masukan yang memiliki dua dimensi. CNN ini terinspirasi dari sistem biologis visual manusia yang menggabungkan ekstraksi fitur dan tahap pengklasifikasian (Syulistyo dkk., 2016). Menurut Putra (2016) karena kedalaman jaringan yang tinggi, CNN termasuk dalam jenis *Deep Neural Network*. CNN ini banyak diaplikasikan pada data citra. Misalnya bila dibandingkan dengan *Multi Layer Perceptron* ia kurang sesuai untuk digunakan pada klasifikasi citra karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik. Kemudian hal lain mengenai MLP yaitu, jika jaringannya tidak memiliki hidden layer maka ia mampu memecahkan persamaan linear apapun, lalu jika jaringan dengan satu atau dua hidden layer ia mampu memecahkan sebagian besar persamaan pada data sederhana. Tetapi MLP memiliki keterbatasan pada data yang lebih kompleks. Pada permasalahan yang ada pada jaringan yang memiliki jumlah *hidden layer* dibawah tiga *layer*, terdapat solusi dengan melakukan pendekatan untuk menentukan jumlah neuron pada masing-masing *layer* untuk mendekati hasil optimal. Namun dalam penggunaan layer diatas dua pada umumnya tidak direkomendasikan pada MLP karena akan menyebabkan *overfitting* serta kekuatan *backpropagation* berkurang secara signifikan.

Dengan adanya perkembangan *deep learning*, ditemukan cara untuk mengatasi kekurangan MLP dalam menangani data yang kompleks, yaitu diperlukan fungsi untuk mentransformasi data input menjadi bentuk yang lebih

mudah dimengerti oleh MLP. Hal tersebut memicu perkembangan *deep learning* yang menjadikan dalam suatu model diberi beberapa *layer* untuk melakukan transformasi data sebelum data diolah menggunakan metode klasifikasi. Oleh karena itu model neural network berkembang dengan jumlah *layer* diatas tiga yaitu CNN. Tetapi karena fungsi *layer* awal bertugas sebagai metode ekstraksi fitur, maka jumlah *layer* dalam sebuah CNN tidak memiliki aturan universal sehingga jumlah layer berbeda-beda tergantung dengan dataset yang digunakan. Layer awal pada CNN ini terdiri dari layer konvolusi dan layer subsampling. Tujuan dari konvolusi pada data citra adalah untuk mengekstraksi fitur dari input citra tersebut. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN. Kemudian layer *subsampling* bertugas mereduksi ukuran data citra. Dalam pengolahan citra ini, *subsampling* pun bertujuan untuk meningkatkan invariansi posisi dari fitur citra tersebut (Putra, 2016). Dengan adanya layer awal ini memberikan keuntungan pada proses training karena fitur dari citra yang telah diekstraksi dan ukuran citra yang telah direduksi mampu meringankan dan mempercepat proses *training*, serta mampu meningkatkan invariansi posisi pada model.

Sebelumnya telah ada beberapa penelitian mengenai alternatif penggunaan OMR scanner ini. Seperti pada penelitian Karunanayake (2015) telah dilakukan pemeriksaan lembar jawaban dengan menggunakan *template matching* dengan tingkat keberhasilan mengenali isi lembar ujian sebesar 97,6%, namun lembar ujian yang digunakan masih lembar ujian yang dibulatkan. Selain itu ada penelitian Krisnadi, Mitra, Desanti, Ciputra, & Hery (2017) yang telah menggunakan lembar ujian isian tulisan tangan yang dalam proses pengenalannya menggunakan metode radial basis function neural network. Pada penelitian Krisnadi, dkk. (2017) memberikan hasil akurasi sebesar 82,28% pada karakter digit, 72,25% pada karakter huruf besar, dan tingkat keberhasilan mengenali isi lembar ujian sebesar 99.69%. Kemudian ada penelitian serupa lainnya mengenai pengenalan tulisan tangan pada lembar ujian yang telah dilakukan oleh Aprijani & Sufandi (2011) dengan menggunakan metode jaringan syarat tiruan propagasi balik untuk

mengenali tulisan tangan huruf A, B, C, dan D pada jawaban soal pilihan ganda. Hasil akurasi dengan menggunakan arsitektur jaringan yang paling optimal pada penelitian tersebut telah menghasilkan presisi sebesar 87,35% untuk data uji. Karena tingkat presisinya yang baru mencapai 87,35% maka perlu dilakukan percobaan lagi untuk mendapatkan nilai yang lebih tinggi karena idealnya untuk dapat digunakan mengolah data ujian presisi harus mencapai 100% menurut Aprijani & Sufandi (2011), sehingga peserta ujian tidak ada yang dirugikan.

Selain itu ada beberapa penelitian serupa mengenai pengenalan tulisan tangan yaitu seperti Pengenalan Angka Tulisan Tangan Menggunakan *Diagonal Feature Extraction* dan Klasifikasi Artificial Neural Network Multilayer *Perceptron* (ANN MLP) pada penelitian Firmasnsyah, Ramadhani, & Arifianto (2018) yang menghasilkan akurasi 92,30% pada dataset C1 dan 92,60% pada dataset MNIST (dataset yang berisi citra angka tulisan tangan). Kemudian pada penelitian Cireşan, Meier, Gambardella, & Schmidhuber (2011) dengan judul *Convolutional Neural Network Committees For Handwritten Character Classification*, ia mendapatkan akurasi untuk dataset MNIST sebesar 99,73% dengan menggunakan CNN. Kemudian penelitian yang serupa lainnya yang berjudul *Deep Learning Architectures for Hard Character Classification* oleh Bui & Chang (2016) yang menggunakan metode CNN dan menggunakan dataset notMNIST (isinya mirip dengan dataset MNIST) menghasilkan akurasi 98%. Lalu ada penelitian lain yang serupa yaitu penelitian Hara, Fitriawan, & Mulyani (2016) yang berjudul Penggunaan Deteksi Tepi (Canny) pada Sistem Pengenalan Tulisan Tangan Aksara Lampung Berbasis Jaringan Syaraf Tiruan. Jaringan syaraf tiruan pada penelitian ini menggunakan metode *Backpropagation*. Sistem aplikasi pengenalan aksara Lampung ini menghasilkan akurasi 78%.

Pada penelitian ini memilih metode CNN karena melihat beberapa kelebihan yang dimiliki CNN yaitu keberhasilan CNN pada beberapa penelitian sebelumnya, desainnya yang cocok untuk mengolah citra, kemampuannya untuk mempercepat proses training, kemampuannya untuk meningkatkan invariansi posisi pada model, serta kemampuan CNN dalam menghadapi persoalan yang kompleks. Oleh karena itu pada penelitian ini diharapkan dengan digunakannya metode CNN ini mampu

menghasilkan sistem yang memiliki akurasi tinggi dan mampu mengenali tulisan tangan pada lembar ujian dengan baik.

## 1.2 Rumusan Masalah

1. Bagaimana mengimplementasikan metode *Convolutional Neural Network* untuk mengenali tulisan tangan pada lembar ujian pilihan ganda?
2. Bagaimana tingkat akurasi dari sistem yang dibangun dalam menerapkan metode *Convolutional Neural Network* untuk mengenali tulisan tangan pada lembar ujian pilihan ganda?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini yaitu:

1. Mengimplementasikan metode *Convolutional Neural Network* untuk mengenali tulisan tangan pada lembar ujian pilihan ganda.
2. Didapatkannya hasil tingkat akurasi dari sistem yang dibangun dalam menerapkan metode *Convolutional Neural Network* untuk mengenali tulisan tangan pada lembar ujian pilihan ganda.

## 1.4 Batasan Masalah

Batasan dari masalah yang akan dibahas dalam penelitian ini adalah:

1. Tulisan tangan yang dikenali berupa huruf kapital dan angka.
2. Lembar ujian pilihan ganda yang digunakan adalah lembar ujian yang telah ditentukan formatnya.
3. Menggunakan alat tulis *ballpoint* atau spidol hitam.

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Mengenali tulisan tangan pada lembar ujian pilihan ganda.
2. Meminimalisir kesalahan dalam mengisi lembar ujian pilihan ganda.

## 1.6 Struktur Penulisan

Sistematika penulisan dalam penelitian ini adalah sebagai berikut:

### BAB I PENDAHULUAN

Bab ini mengandung latar belakang penelitian, identifikasi masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan dokumen proposal skripsi ini.

## BAB II TINJAUAN PUSTAKA

Bab ini berisi materi-materi hasil literatur, teori-teori tentang *Convolution Neural Network*.

## BAB III METODOLOGI PENELITIAN

Bab ini berisi penjelasan langkah-langkah yang akan dilakukan dalam penelitian.

## BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Bab ini berisi uraian tentang hasil penelitian dan pembahasan terhadap hasil penelitian yang dilakukan.

## BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari keseluruhan penelitian yang telah dilakukan, serta saran dari penulis untuk kegiatan penelitian selanjutnya terkait dengan topik yang sedang dibahas.

## BAB II

### KAJIAN PUSTAKA

#### 2.1 Penelitian Terkait

Ada beberapa penelitian serupa mengenai pengenalan tulisan tangan yaitu seperti pada Pengenalan Angka Tulisan Tangan Menggunakan *Diagonal Feature Extraction* dan Klasifikasi Artificial Neural Network Multilayer *Perceptron* (ANN MLP) pada penelitian (Firmansyah, dkk., 2018). Pada penelitian tersebut pada tahap ekstraksinya menggunakan *Diagonal Feature Extraction* yang mengekstraksi dengan cara merata-ratakan nilai piksel pada setiap diagonalnya kemudian dirata-ratakan sehingga didapatkan ciri pada area tersebut. Dan pada proses klasifikasinya menggunakan ANN MLP. Hasil akurasi cukup baik yaitu 92.30% pada dataset C1 dan 92.60% pada dataset MNIST (dataset yang berisi citra angka tulisan tangan). Menurut Firmansyah, dkk. (2018) ukuran area yang kecil yaitu 5X5 pada tahap ekstraksi serta dengan menggunakan ciri diagonal dan rata-rata diagonal yang membuat akurasi lebih tinggi.

Kemudian ada beberapa penelitian serupa yang menggunakan *Convolutional Neural Network* yaitu Ciresan, dkk. (2011) dengan judul *Convolutional Neural Network Committees For Handwritten Character Classification*, ia mendapatkan akurasi untuk dataset MNIST sebesar 99.73% dengan menggunakan CNN. Kemudian penelitian yang serupa lainnya yang berjudul *Deep Learning Architectures for Hard Character Classification* oleh Bui & Chang (2016) pada penelitian ini menggunakan metode CNN dan menggunakan dataset notMNIST yang isinya mirip dengan dataset MNIST namun citra dataset lebih nyata karena memiliki noise, pada penelitian ini digunakan data *training set* berisi 52900 citra dan *testing set* yang kurang lebih berisi 1870 citra pada setiap *classnya*. Penelitian ini menghasilkan akurasi 98%.

Lalu ada penelitian lain yang serupa namun tidak menggunakan metode CNN yaitu penelitian Hara, dkk. (2016) yang berjudul Penggunaan Deteksi Tepi (Canny) pada Sistem Pengenalan Tulisan Tangan Aksara Lampung Berbasis Jaringan Syaraf Tiruan. Jaringan syaraf tiruan pada penelitian ini menggunakan metode *Backpropagation*. Sistem aplikasi pengenalan aksara Lampung ini menghasilkan

akurasi 78% setelah 10 (sepuluh) kali pengujian. Namun menurut Hara, dkk. (2016) metode deteksi tepi (canny) yang digunakannya ini kurang maksimal dalam mengenali aksara Lampung, karena canny membaca bagian-bagian tepi objek dan hanya memberikan satu tanggapan biner, yang menyebabkan terpengaruhnya pengolahan citra selanjutnya dan ketika proses pengenalan karakter menjadi lebih sulit.

Penelitian serupa mengenai pengenalan tulisan tangan pada lembar ujian telah dilakukan seperti pada penelitian (Aprijani & Sufandi, 2011) yang menggunakan metode jaringan syarat tiruan propagasi balik untuk mengenali tulisan tangan huruf A, B, C, dan D pada jawaban soal pilihan ganda. Pada penelitian tersebut (Aprijani & Sufandi, 2011) menggunakan 8000 karakter huruf (A, B, C, dan D) sebagai sampel yang dibagi menjadi dua bagian untuk training dan testing. Jumlah sampel untuk data latih 6000 karakter dan 2000 karakter untuk data uji. Hasil dari penelitian (Aprijani & Sufandi, 2011) menunjukkan bahwa performa paling optimal pada arsitektur jaringannya adalah dengan menggunakan parameter 10 neuron dalam *hidden layer*, *learning rate* 0,1, dan iterasi sebanyak 3000 kali. Hasil akurasi dengan menggunakan arsitektur jaringan yang paling optimal tadi telah menghasilkan 90.28% untuk data latih dan 87.35% untuk data uji. Karena tingkat presisinya yang baru mencapai 87,35% maka perlu dilakukan percobaan lagi untuk mendapatkan nilai yang lebih tinggi karena idealnya untuk dapat digunakan mengolah data ujian presisi harus mencapai 100%, sehingga peserta ujian tidak ada yang dirugikan (Aprijani & Sufandi, 2011).

## 2.2 Tulisan Tangan

Menurut Colmas dalam Plamondon & Srihari (2000) tulisan tangan adalah keterampilan yang bersifat pribadi bagi individu. Menulis dianggap telah dimungkinkan banyak budaya dan peradaban. Setiap skrip memiliki satu set simbol, yang dikenal sebagai karakter atau huruf, yang memiliki bentuk dasar tertentu. Ada aturan untuk menggabungkan huruf untuk mewakili bentuk unit linguistik tingkat yang lebih tinggi.

Tulisan tangan dikembangkan sejak lama sebagai sarana untuk memperluas ingatan manusia dan untuk memfasilitasi komunikasi. Ada banyak cara untuk memperluas ingatan manusia serta memfasilitasi komunikasi sering dengan

berkembangnya jaman. Tulisan tangan telah sangat berubah dari waktu ke waktu dan, sejauh ini, dorongan teknologi telah berkontribusi pada perluasnya (Plamondon & Srirahari, 2000).

### **2.3 Evaluasi**

Evaluasi menurut Mardapi (2013) adalah kegiatan untuk mengetahui pencapaian suatu program. Evaluasi memberi informasi untuk kebijakan dengan dua cara. Pertama evaluasi memberi informasi untuk pencapaian belajar suatu grup tertentu atau pembuat kebijakan tentang keadaan pendidikan. Kemudian yang kedua, informasi dari evaluasi digunakan sebagai piranti administratif untuk menerapkan kebijakan.

Menurut Mardapi (2009) evaluasi memerlukan dua jenis data yaitu data kuantitatif dan kualitatif. Data-data tersebut diperoleh dari suatu pengukuran. Menurut Ebel pada Mardapi (2009) pengukuran adalah pemberian angka pada sesuatu objek atau seseorang yang bertujuan untuk membedakan tingkat objek atau orang tersebut mengenai hal yang diukur dengan cara sistimatis. Kemudian Nunnally pada Mardapi (2009) juga menjelaskan bahwa pengukuran terdiri dari beberapa aturan untuk memberikan angka kepada objek dengan cara yang sedemikian rupa sehingga didapatkan representasi secara kuantitatif dari sifat-sifat objek tersebut.

Menurut Mardapi (2009) pengukuran ini dapat dilakukan dengan melalui tes dan dapat pula tidak melalui tes. Tes itu sendiri menurut Cronbach pada Mardapi (2009) adalah prosedur sistematis yang digunakan untuk mengobservasi perilaku seseorang serta mendeskripsikan perilakunya dengan skala numerik atau sistem kategori.

Kemudian menurut Mardapi (2009), hasil pengukuran ini digunakan untuk melakukan penilaian dan/atau evaluasi. Terkait dengan penilaian dan evaluasi, menurut Mardapi (2009) ada beberapa ahli yang mengatakan evaluasi dan penilaian memiliki kesamaan namun ada pula ahli lainnya yang menganggap bahwa kedua hal itu berbeda. Menurut Nitko pada Mardapi (2009) penilaian adalah proses untuk memperoleh informasi yang bertujuan untuk pengambilan keputusan tentang kebijakan pendidikan, kurikulum, program pendidikan, dan kegiatan belajar siswa.

Sedangkan evaluasi adalah proses memperoleh informasi yang bertujuan untuk menimbang kebaikan kinerja siswa. Informasi yang didapatkannya ini digunakan untuk mengetahui apakah tujuan yang telah ditentukan tercapai atau tidak.

Menurut Worthen dan Sanders pada Mardapi (2009) menjelaskan bahwa evaluasi adalah penentuan nilai dari sesuatu, yang mencakup pengumpulan informasi yang akan digunakan pada penentuan nilai dari program, hasil, prosedur, atau suatu tujuan; atau kegunaan potensial dari pendekatan yang telah dirancang untuk mencapai suatu tujuan yang telah ditentukan. Kemudian ia menyatakan bahwa evaluasi merupakan proses pencarian, penggambaran, dan pemberian informasi yang amat bermanfaat bagi proses pengambil keputusan dalam menentukan alternatif keputusan.

Berdasarkan pengertian tadi, menurut Mardapi (2009) dalam kegiatan evaluasi terkandung beberapa makna yaitu adanya pengumpulan informasi, penggambaran, pencarian, dan penyajian informasi yang berguna untuk pengambilan keputusan tentang program yang sedang dilaksanakan. Menurut Valadez dan Bamberger pada Mardapi (2009) tujuan dari evaluasi adalah untuk menilai kesesuaian rencana suatu program dengan metode pelaksanaannya dalam mencapai tujuan, menilai hasil dari program, yang mana hasil yang diharapkan maupun yang tidak diharapkan, dan terakhir menilai beberapa faktor yang mempengaruhi keberhasilan dari suatu program.

#### **2.4 Lembar Jawaban Ujian**

Pada proses evaluasi biasanya digunakan lembar jawaban ujian. Lembar jawaban ujian adalah bagian dari modul yg berisi jawaban untuk digunakan para siswa atau guru dalam menilai jawaban yg telah dituliskan pada lembar uji. Pada masa kini untuk mempermudah proses pemeriksaan lembar jawaban ujian dapat digunakan Lembar Jawaban Komputer (LJK).

Menurut Johar, Andreswari, & Triyana (2014) Lembar Jawaban Komputer (LJK) merupakan lembar jawaban yang digunakan untuk melakukan ujian secara tertulis yang pada pemeriksaannya LJK harus menggunakan alat scanner khusus. Menurut Wibowo (2013) LJK dibaca dan diproses menggunakan perangkat yang disebut Optical Mark Reader (OMR) yaitu pemindai khusus yang dapat mendeteksi tanda pada lembar jawab. Tanda pada lembar jawaban tersebut berupa bulatan

hitam yang dibuat oleh pengisi jawaban menggunakan pensil, yang biasanya menggunakan jenis pensil 2B. OMR tersebut dapat mendeteksi letak tanda menggunakan inframerah lalu menyimpan informasinya dan diproses lebih lanjut, yaitu dapat berupa penghitungan jumlah jawaban tertentu atau dapat juga dilakukan pencocokan jawaban dengan kunci jawaban yang telah tersedia.

Kemudian menurut Wibowo (2013) ada beberapa kekurangan yang muncul dengan penggunaan teknologi OMR, yaitu diperlukan biaya yang mahal untuk pencetakan lembar jawab komputer serta pembelian scanner OMR sehingga hanya beberapa pihak yang dapat menggunakannya, diperlukan kertas dengan ketebalan khusus dalam pencetakan lembar jawab komputer, diperlukan alat tulis khusus yang digunakan untuk pengisian lembar jawab komputer sobek sedikit maka tidak dapat diperiksa tentu saja ini dapat merugikan pengisi jawaban, menurut Prabowo & Johar (2014) alasan LJK tidak boleh rusak adalah karena LJK rusak tidak dapat dimasukkan ke dalam scanner, karena dapat menyebabkan lembar tersebut tidak dapat terbaca oleh *scanner*, dan menurut Rahmat dalam Wibowo (2013) walaupun dibutuhkan, sistem dengan teknologi *scanner* OMR ini tidak tepat untuk diterapkan pada institusi berskala kecil.

Kesalahan pengisian lembar jawaban ternyata masih terjadi. Kesalahan inipun terjadi di Universitas Terbuka menurut Aprijani & Sufandi (2011) berdasarkan data yang diperoleh dari Pusat Pengujian UT, pada masa registrasi 2010, terdapat 34.326 kasus (2,91%) kesalahan menghitamkan bulatan pada LJK.

## 2.5 Kecerdasan Buatan

Kecerdasan buatan (*artificial intelligence*) adalah sebuah penelitian, aplikasi dan instruksi yang berkaitan dengan pemrograman komputer untuk melakukan sesuatu yang cerdas (menurut manusia) (Simon, 1995). Sebelum adanya komputer, satu-satunya contoh kecerdasan yang dapat diamati adalah pikiran dari organisme hidup, terutama manusia. Sekarang keluarga sistem cerdas telah bergabung dengan genus baru, program komputer cerdas (Simon, 1995).

Teknik yang digunakan dalam kecerdasan buatan memungkinkan pembuatan program yang setiap bagian berisi langkah pemecahan masalah dan dapat diidentifikasi dengan baik untuk memecahkan sejumlah masalah. Setiap bagian dari bagian program adalah seperti sepotong informasi dalam pikiran

manusia. Jika informasi diabaikan, itu dapat secara otomatis mengatur cara kerjanya untuk menyesuaikan dengan fakta atau informasi baru. Tidak perlu selalu mengingat setiap informasi yang telah dipelajari. Hanya informasi yang relevan dengan masalah yang dihadapi dan digunakan. Demikian pula dengan sistem kecerdasan buatan, setiap bagian dari program kecerdasan buatan dapat dimodifikasi tanpa mempengaruhi struktur keseluruhan programnya. Ini dapat mengarah pada program yang efisien dan mudah dimengerti (W. & Farqi, 2015). Menurut Kröse pada (W. & Farqi, 2015) ada 2 basis utama yang dibutuhkan untuk aplikasi kecerdasan buatan:

- 1) Basis pengetahuan (*knowledge base*): berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan yang lain.
- 2) Motor inferensi (*inference engine*): kemampuan menarik kesimpulan berdasarkan pengalaman.

## 2.6 Pengenalan Pola

Pengenalan pola adalah suatu proses untuk mengenal objek dengan menggunakan satu atau bermacam – macam metode yang mana hasil dari proses pengenalannya memiliki akurasi yang tinggi. Maksud dari memiliki akurasi yang tinggi ini adalah apabila suatu objek tidak dapat dikenali secara manual (oleh manusia) tetapi dapat dikenali oleh aplikasi komputer yang menggunakan suatu metode pengenalan (Wijaya & Kanata, 2004).

Kemudian menurut Putra pada Yodha & Kurniawan (2014) pengenalan pola adalah sebuah ilmu dalam kecerdasan buatan yang memiliki tujuan untuk mengklasifikasikan sesuatu berdasarkan pengukuran fiturnya. Pengenalan pola ini telah banyak digunakan dalam berbagai penelitian dan diterapkan pada berbagai bidang. Beberapa contoh penerapan pengenalan pola diantaranya adalah untuk pengenalan wajah, pengenalan tulisan tangan, finger print, finger knuckle, pola batik, nomor kendaraan dan wayang.

Menurut Yodha dan Kurniawan (2014) tujuan dari pengenalan pola adalah untuk menentukan kelompok atau kategori suatu pola yang berdasarkan ciri yang dimiliki oleh pola tersebut atau dengan kata lain, pengenalan pola bertujuan untuk membedakan suatu objek dengan objek yang lain.

## 2.7 Citra Digital

Menurut Kusumanto & Tompunu (2011) secara matematis, citra adalah fungsi kontinyu (terus-menerus) yang memiliki informasi intensitas cahaya secara dua dimensi. Citra dapat diolah oleh komputer dengan syarat, ia dipresentasikan dengan suatu nilai-nilai diskrit numerik. Reperesentasi dari fungsi terus menerus ini yang kemudian menjadi nilai-nilai diskrit yang disebut dengan digitalisasi citra. Gambar digital dapat diwakili oleh matriks dua dimensi (x, y) yang terdiri dari kolom M dan garis N seperti pada Gambar 2.1, di mana perpotongan antara kolom dan baris disebut sebagai piksel atau elemen gambar.

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

Suatu citra  $f(x,y)$  pada fungsi matematis dapat dituliskan sebagai berikut:

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

$$0 \leq y \leq N-1$$

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

Dimana :

M = jumlah piksel baris (*row*) pada array citra

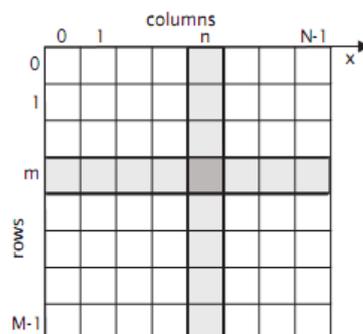
N = jumlah piksel kolom (*column*) pada array citra

G = nilai skala keabuan (*graylevel*)

Besarnya nilai  $M$ ,  $N$  dan  $G$  pada umumnya merupakan perpangkatan dari dua.

$$M = 2^m; N = 2^n; G = 2^k \quad (2)$$

Nilai  $m$ ,  $n$  dan  $k$  adalah bilangan bulat positif. Interval (0, G) disebut juga sebagai skala keabuan (*grayscale*). Besar G dipengaruhi oleh proses digitalisasinya.



Gambar 2.1 Matriks Citra (Kusumanto & Tompunu,2011)

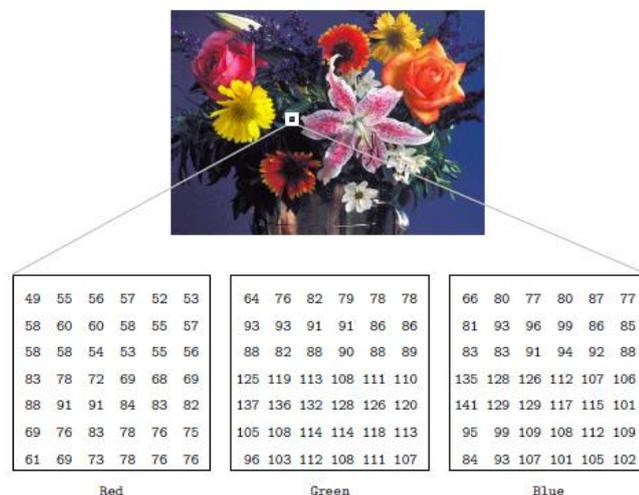
Tingkat keabuan 1 (satu) menyatakan intensitas putih dan 0 (nol) menyatakan intensitas hitam. Pada citra 8 bit, nilai G sama dengan  $2^8 = 256$  warna (derajat keabuan) (Kusumanto & Tomponu, 2011).

## 2.8 Jenis Citra Digital

Citra digital umumnya, memiliki 3 bagian yaitu, black and white image, color image, dan binary image (Kusumanto & Tomponu, 2011).

### 1) Color Image atau RGB (Red, Green, Blue)

Dalam gambar warna ini pada setiap pikselnya memiliki warna, warna yang dimaksud adalah merah (*Red*), hijau (*Green*) dan biru (*Blue*). Jika setiap warna memiliki kisaran nilai 0-255, maka totalnya adalah  $255^3 = 16,581,375$  (16 K) variasi warna yang berbeda-beda dalam gambar. Gambar juga disebut gambar bit-warna karena jumlah bit yang diperlukan untuk setiap pikselnya, *Color image* ini memiliki 3 buah matriks yang mewakili nilai merah, hijau, dan biru untuk setiap pikselnya. Dapat dilihat pada contoh gambar 2.2 diambil sampel matriks berukuran 6x6 (Kusumanto & Tomponu, 2011).



Gambar 2.2 Citra Berwarna (McAndrew pada Kusumanto & Tomponu, 2011).

### 2) Black and White

Citra digital *black and white* (grayscale) pada setiap piksel memiliki gradasi warna dari putih (bernilai 256) ke hitam (bernilai 0) sampel matriks nilai dari citra hitam putih berukuran 7x7 dapat dilihat pada gambar 2.3. Rentang ini dapat diwakili oleh 8 bit untuk setiap pikselnya, atau 1 byte. Rentang

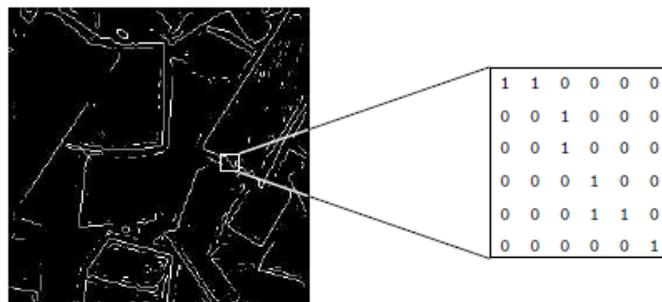
warna hitam dan putih dapat digunakan untuk pengolahan file gambar (Kusumanto & Tomponu, 2011).



Gambar 2.3 Citra Grayscale (McAndrew pada Kusumanto & Tomponu, 2011).

### 3) *Binary Image.*

Pada setiap piksel dari *binary image* hanya terdiri dari 2 warna yaitu hitam dan putih, karena setiap pikselnya hanya memiliki 2 warna, maka ia hanya perlu kapasitas 1 bit per piksel (0 dan 1) atau jika didalam 8 bit ( 0 dan 255) seperti terlihat pada gambar 2.4, oleh karena itu ini sangat efisien dalam hal penyimpanannya. Gambar yang direpresentasikan dengan biner sangat cocok untuk gambar tulisan tangan, sidik jari (finger print), atau gambar arsitektur (Kusumanto & Tomponu, 2011).



Gambar 2.4 Citra Binary (McAndrew ,pada Kusumanto & Tomponu,2011).

## 2.9 Pengolahan Citra Digital

Pengolahan citra (*Image Processing*) merupakan suatu sistem yang pada prosesnya melibatkan masukan berupa citra (*image*) yang kemudian menghasilkan citra (*image*) menurut Achmad, Basuki, Palandi, dkk pada Al Farqi (2015). Pada awalnya pengolahan gambar ini dilakukan untuk meningkatkan kualitas

gambar, tetapi dengan adanya perkembangan dunia komputasi yang memungkinkan manusia untuk mengambil informasi dari suatu gambar, pengolahan gambar menjadi tidak bisa dipisahkan dengan bidang computer vision menurut Achmad, Basuki, Palandi, dkk pada pada (W. & Farqi, 2015).

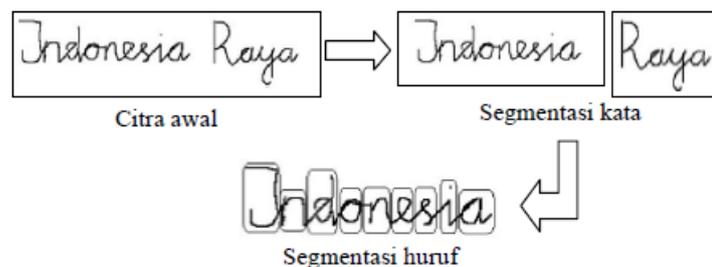
Sesuai dengan adanya perkembangan komputer itu sendiri, dua tujuan utama dari pengolahan citra, yaitu sebagai berikut:

- 1) Kualitas citra yang diperbaiki, sehingga gambar yang dihasilkan dapat memberikan informasi yang cukup jelas. Pada hal ini maka manusia berperan sebagai pengolah informasi (*human perception*).
- 2) Mengekstraksi ciri yang khusus pada citra untuk memberikan informasi yang khusus juga, yang mana hasilnya adalah informasi gambar yang terdiri dari ciri dari citra secara numerik atau dapat disebut juga komputer melakukan interpretasi terhadap informasi yang ada pada citra melalui data yang besar - besaran yang dapat dibedakan secara jelas (berupa besaran numerik).

Dalam pengembangan lebih lanjut, pemrosesan gambar dan *computer vision* digunakan sebagai pengganti mata manusia yang berperan menjadi gambar gambar input seperti kamera, *scanner*, dan komputer sebagai otak yang mengolah informasi. Beberapa orang menggunakan *computer vision*, pengenalan pola, pengenalan manusia berdasarkan karakteristik manusia, konten berbasis gambar dan retrieval video (mengambil gambar atau video) dengan informasi tertentu), penyuntingan video dan banyak lagi menurut Achmad, Basuki, Palandi, dkk pada pada (W. & Farqi, 2015).

Operasi pengolahan citra ada berbagai macam, segmentasi dan *scalling* merupakan sebagian kecil contoh pengolahan citra (Rohwana & Irawan, 2013).

- 1) Segmentasi memiliki tujuan untuk memisahkan bagian-bagian tertentu pada gambar dengan cara mencari piksel-piksel terluar, sehingga menjadikannya



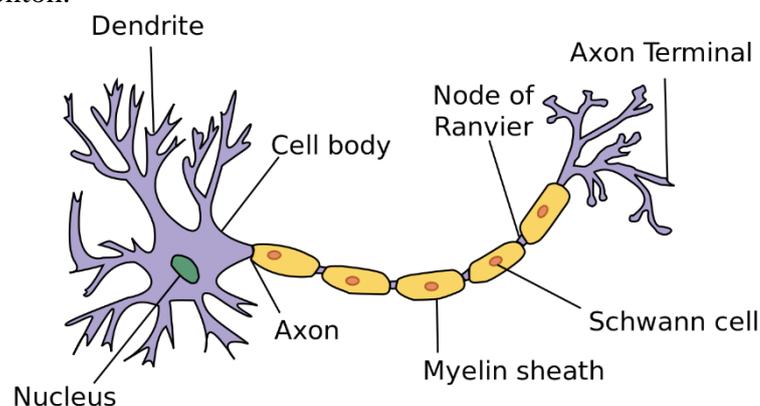
batas, yang nantinya didapatkan citra segiempat yang siap diolah seperti pada gambar 2.5 (Rohwana & Irawan, 2013).

Gambar 2.5 Proses Segmentasi (Rohwana & Irawan, 2013)

- 2) Kemudian *scaling* bertujuan untuk menormalisasikan ukuran suatu gambar sehingga ukurannya dapat selalu sama.

## 2.10 Jaringan Syaraf Tiruan

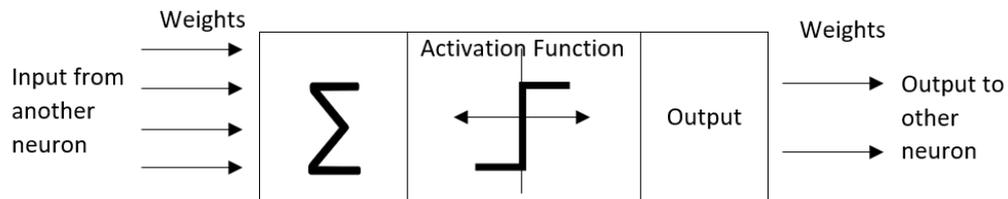
Menurut Maharani pada Sudarsono (2016) Jaringan Syaraf Tiruan (JST) adalah jaringan yang memiliki sistem pemrosesan informasi yang karakteristiknya menyerupai jaringan syaraf biologis (Gambar 2.6). Jaringan Syaraf Tiruan terbentuk dari pemahaman manusia (*human cognition*) yang digeneralisasi kedalam model matematis. Kemudian menurut Salim & Jauhari (2016) sistem dari pengolahan informasi pada jaringan syaraf tiruan terdiri dari banyak neuron yang saling berhubungan, yang bekerja secara bersamaan untuk menyelesaikan pekerjaannya. Cara kerja jaringan syaraf tiruan menyerupai cara kerja manusia yaitu belajar dari contoh.



Gambar 2.6 Syaraf Biologis

Sama halnya dengan otak manusia, jaringan syaraf tiruan terdiri dari beberapa *neuron*, dan memiliki hubungan antara *neuron-neuron* tersebut. Pada gambar 2.7, menunjukkan struktur *neuron* yang mentransformasikan informasi yang diterimanya menuju ke *neuron-neuron* yang lain. Pada jaringan syaraf tiruan hal ini lebih dikenal dengan nama bobot. Informasi tersebut tersimpan pada suatu nilai tertentu pada bobot tersebut. keduanya atau mungkin lebih untuk mendapatkan redundansi data. Hal ini akan diproses oleh suatu fungsi yang akan menjumlahkan semua nilai bobot yang akan datang. Hasil penjumlahan ini kemudian dibandingkan

dengan suatu Informasi yang disebut dengan masukan yang dikirim ke *neuron* dengan bobot kedatangan tertentu (Sudarsono, 2016).



Gambar 2.7 Struktur Neuron JST (Sudarsono, 2016)

Jaringan syaraf tiruan mempunyai kemampuan yang hebat dalam mendapatkan informasi dari data yang cukup rumit ataupun tidak tepat, mampu menyelesaikan permasalahan yang tidak terstruktur dan sulit didefinisikan, dapat belajar dari pengalaman, mampu mengakuisisi pengetahuan walaupun tidak ada kepastian, mampu melakukan generalisasi dan ekstraksi dari suatu pola data tertentu, dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar (*self organizing*), mampu memilih suatu input data ke dalam kategori tertentu yang sudah ditetapkan (klasifikasi), mampu menggambarkan suatu objek secara keseluruhan walaupun hanya diberikan sebagian data dari objek tersebut (asosiasi), mempunyai kemampuan mengolah data-data *input* tanpa harus mempunyai target (*self organizing*), dan mampu menemukan jawaban terbaik sehingga mampu meminimalisasi fungsi biaya (optimasi) (Salim & Jauhari, 2016).

Kelebihan-kelebihan yang diberikan oleh jaringan syaraf tiruan antara lain:

- 1) *Belajar Adaptive*: Kemampuan untuk mempelajari bagaimana melakukan suatu pekerjaan dengan hanya menggunakan data yang digunakan untuk pengalaman atau pelatihan.
- 2) *Self-Organisation*: Kemampuan untuk merepresentasikan atau membuat organisasi sendiri dari informasi yang dimiliki ketika belajar.
- 3) *Real Time Operation*: Karena perhitungan jaringan syaraf tiruan dapat dilakukan secara paralel maka perangkat keras yang dirancang secara khusus mendapatkan keuntungan dari kemampuan ini.

Namun jaringan syaraf tiruan juga memiliki kelemahan-kelemahan juga yaitu berikut ini (Salim & Jauhari, 2016).

- 1) Tidak efektif dalam proses operasi numerik yang membutuhkan presisi tinggi.
- 2) Tidak efisien bila digunakan untuk proses algoritma aritmatik, operasi logika, dan simbolis.
- 3) Waktu yang digunakan untuk proses pelatihan akan sangat lama bila jumlah datanya besar.

### **2.11 Deep Learning**

Menurut Lecun, Bengio, & Hinton (2015) teknik *machine learning* konvensional terbatas kemampuannya untuk memproses data alami dalam bentuk mentahnya. Selama beberapa dekade, membangun pengenalan pola atau sistem *machine learning* membutuhkan rekayasa yang cermat dan keahlian domain yang cukup untuk merancang fitur *extractor* yang mengubah data mentah (seperti nilai piksel dari suatu gambar) ke dalam representasi internal yang sesuai atau fitur vector dari subsistem pembelajaran, seringkali sebuah classifier, dapat mendeteksi atau mengklasifikasikan pola dalam input.

Representasi pembelajaran adalah seperangkat metode yang memungkinkan mesin untuk diberi masukan dengan data mentah dan secara otomatis menemukan representasi yang diperlukan untuk mendeteksi atau klasifikasi. Metode *deep-learning* adalah representasi- metode pembelajaran dengan berbagai tingkat representasi, yang diperoleh dengan menyusun modul sederhana namun non-linear yang masing-masing mengubah representasi pada satu level (dimulai dengan input mentah) menjadi representasi tingkat yang lebih tinggi, yang mana tingkatnya sedikit lebih abstrak. Dengan komposisi transformasi yang cukup, fungsi yang sangat kompleks mampu dipelajari. Untuk tugas-tugas klasifikasi, lapisan representasi yang lebih tinggi memperkuat aspek input yang penting untuk diskriminasi dan mampu menekan variasi yang tidak relevan. Sebuah gambar, misalnya, muncul di bentuk array nilai piksel, dan fitur yang dipelajari di yang pertama lapisan representasi biasanya mewakili ada atau tidak adanya tepi pada orientasi dan lokasi tertentu dalam gambar. Kedua lapisan biasanya mendeteksi motif dengan melihat pengaturan tertentu tepi, terlepas dari variasi kecil pada posisi

tepi. Ketiga, lapisan dapat merakit motif menjadi kombinasi yang lebih besar yang sesuai dengan bagian-bagian dari objek yang dikenal, dan lapisan berikutnya akan mendeteksi objek sebagai kombinasi dari bagian-bagian ini. Aspek kunci dari *deep learning* adalah lapisan-lapisan fitur ini tidak dirancang oleh insinyur manusia, lapisan-lapisan ini dipelajari dari data menggunakan *general-purpose learning procedure* (Lecun, dkk., 2015).

*Deep learning* telah membuat kemajuan pesat dalam memecahkan masalah yang telah menahan upaya terbaik komunitas *artificial intelligence* selama bertahun-tahun. *Deep learning* ini ternyata sangat bagus dalam menemukan struktur rumit dalam data dimensi tinggi dan karenanya dapat diterapkan ke banyak domain sains, bisnis dan pemerintahan (Lecun, dkk., 2015).

Menurut Lecun, dkk. (2015) *deep learning* akan memiliki lebih banyak keberhasilan dalam waktu dekat ini karena hanya membutuhkan sedikit teknik dengan tangan, oleh karena itu *deep learning* dapat dengan mudah mengambil keuntungan dari peningkatan jumlah perhitungan yang tersedia dan data. Algoritma dan arsitektur pembelajaran baru yang saat ini sedang dikembangkan *deep neural network* inipun akan mampu mempercepat kemajuan ini.

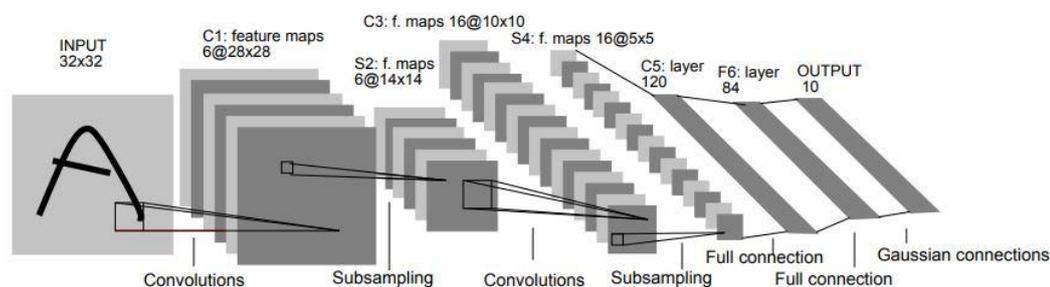
Selain itu menurut Ngiam, dkk. pada Zulfa & Winarko (2017) *Deep Learning* adalah suatu area baru dalam penelitian *Machine Learning*, yang bertujuan untuk menggerakkan *Machine Learning* lebih dekat dengan salah satu tujuan utamanya yaitu *Artificial Intelligence*. *Deep Learning* adalah tentang bagaimana belajar beberapa tingkat representasi dan abstraksi yang membantu untuk memahami data seperti gambar, suara, dan teks.

Kemudian *Deep Learning* menurut Bengio, Goodfellow, & Courville (2016) adalah suatu bentuk *machine learning* yang memungkinkan komputer untuk belajar dari pengalaman dan memahami dunia dalam hal hierarki konsep. Karena komputer mengumpulkan pengetahuan dari pengalaman, operator komputer manusia tidak perlu secara formal untuk menentukan semua pengetahuan yang dibutuhkan oleh komputer. Hirarki dari konsep ini adalah memungkinkan komputer untuk mempelajari konsep yang rumit dengan membangunnya dari sesuatu yang sederhana.

## 2.12 Convolutional Neural Network (CNN)

CNN adalah jenis jaringan syaraf tiruan *feed forward* yang terinspirasi oleh struktur sistem visual manusia. CNN terdiri dari banyak neuron yang memiliki bobot dan bias, di mana setiap neuron menerima beberapa masukan. Dalam hal arsitekturnya, CNN terdiri dari satu atau lebih *convolutional layer* dengan *subsampling stage* dan satu atau lebih *fully connected layer* seperti pada jaringan syaraf tiruan pada umumnya (Syulistyo dkk., 2016). CNN merupakan pengembangan dari *Multi Layer Perceptron*. CNN didesain untuk mengolah data dua dimensi. CNN ini memiliki kedalaman yang tinggi oleh karena itu CNN termasuk *Deep Neural Network*. Pada kasus pengolahan citra MLP kurang baik untuk digunakan karena tidak menyimpan informasi spasial dari data suatu citra,serta menganggap setiap piksel merupakan fitur yang independen sehingga menghasilkan hasil yang kurang baik (Putra, dkk., 2016).

CNN umumnya terdiri dari tiga *layer* yaitu *convolutional layer*, *subsample/pooling layer*, dan *fully connected layer*. Gambar 2.8 menunjukkan contoh dari arsitektur dari CNN.



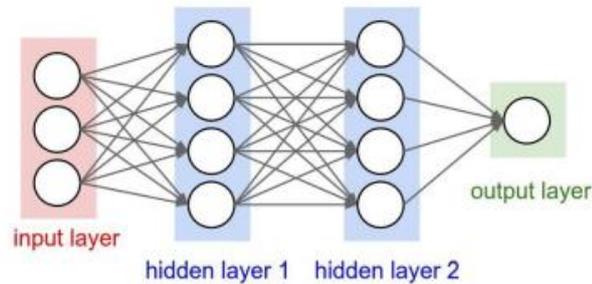
Gambar 2.8 Arsitektur CNN (Lecun dkk. Dalam Syulistyo dkk., 2016).

### 2.12.1 Konsep Convolutional Neural Network

Cara kerja dari CNN serupa dengan MLP, tetapi pada CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, sedangkan pada MLP setiap neuron hanya berukuran satu dimensi.

Suatu MLP seperti pada Gambar 2.9 memiliki  $i$  *layer* (kotak berwarna merah dan biru) dengan setiap *layer* yang berisi sejumlah  $j_i$  neuron (lingkaran berwarna putih). MLP menerima masukan suatu data satu dimensi kemudian mempropagasikan data tersebut pada jaringan hingga menghasilkan suatu *output*.

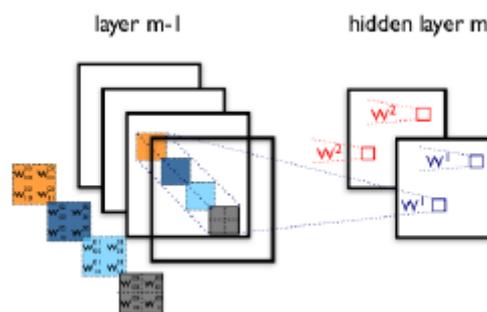
Setiap hubungan antar neuron pada dua *layer* yang bersebelahan memiliki parameter yaitu bobot satu dimensi yang akan menentukan kualitas dari model.



Gambar 2.9 Arsitektur Multi Layer Perceptron Sederhana (Putra, dkk., 2016)

Pada setiap data masukan pada *layer* dilakukan operasi *linear* dengan nilai bobot yang ada, kemudian hasil dari komputasi akan ditransformasikan menggunakan operasi *non linear* yang disebut juga sebagai fungsi aktivasi.

Dalam CNN, data yang dipropagasikan pada jaringan adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Operasi linear pada CNN menggunakan operasi konvolusi, sedangkan untuk bobotnya tidak lagi satu dimensi, tetapi berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi seperti pada Gambar 2.10. Karena sifat proses konvolusinya ini, maka CNN ini hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti pada data citra dan suara.



Gambar 2.10 Proses Konvolusi pada CNN (Putra, dkk., 2016)

### 2.12.2 Arsitektur Jaringan *Convolutional Neural Network*

Menurut Stathakis dalam Putra, dkk. (2016) JST terbangun dari bermacam-macam layer dan beberapa neuron pada setiap layernya. Kedua hal tersebut tidak dapat ditentukan karena tidak memiliki aturan yang pasti dan berlaku berbeda-beda pada tiap data yang berbeda.

Pada kasus MLP, jika jaringannya tidak memiliki hidden layer maka ia mampu memetakan persamaan linear apapun, sedangkan jaringan dengan satu atau dua hidden layer mampu memetakan sebagian besar persamaan pada data sederhana. Tetapi MLP memiliki keterbatasan pada data yang lebih kompleks. Pada permasalahan yang ada pada jaringan yang memiliki jumlah *hidden layer* dibawah tiga *layer*, terdapat solusi dengan melakukan pendekatan untuk menentukan jumlah neuron pada masing-masing *layer* untuk mendekati hasil optimal. Namun dalam penggunaan layer diatas dua pada umumnya tidak direkomendasikan pada MLP karena akan menyebabkan *overfitting* serta kekuatan *backpropagation* berkurang secara signifikan.

Dengan adanya perkembangan *deep learning*, ditemukan cara untuk mengatasi kekurangan MLP dalam menangani data yang kompleks, yaitu diperlukan fungsi untuk mentransformasi data input menjadi bentuk yang lebih mudah dimengerti oleh MLP. Hal tersebut memicu perkembangan *deep learning* yang menjadikan dalam suatu model diberi beberapa *layer* untuk melakukan transformasi data sebelum data diolah menggunakan metode klasifikasi. Oleh karena itu model neural network berkembang dengan jumlah *layer* diatas tiga yaitu CNN. Tetapi karena fungsi *layer* awal bertugas sebagai metode ekstraksi fitur, maka jumlah *layer* dalam sebuah CNN tidak memiliki aturan universal sehingga jumlah layer berbeda - beda tergantung dengan dataset yang digunakan.

Pada penelitian ini CNN hanya menerapkan 3 lapisan utama yaitu:

1) *Convolution Layer*

*Convolution Layer* menurut Putra, dkk. (2016) bertugas untuk melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Layer* ini adalah proses utama yang mendasari terlahirnya CNN. Menurut Putra, dkk. (2016) konvolusi adalah istilah matematis yang artinya mengaplikasikan suatu fungsi pada *output* fungsi lain secara berulang. Pada pengolahan citra, konvolusi memiliki arti mengaplikasikan sebuah *kernel/filter* pada citra disemua *offset* yang memungkinkan. Pada proses penggunaan filter ini ada istilah *stride*, *stride* adalah parameter yang menentukan berapa jumlah pergeseran filter. Misalnya apabila nilai *stride* adalah 1, maka conv. filter akan bergeser sebanyak 1 pixels secara horizontal lalu vertical.

Berikut ini salah satu contoh konvolusi, kotak biru pada Gambar 2.11 adalah contoh secara keseluruhan citra yang akan dikonvolusi. *Filter* akan bergerak dari sudut kiri atas hingga sudut kanan bawah. Bagian pada citra yang tertimpa oleh *kernel* (kotak berwarna merah pada Gambar 2.11), nilai citranya pada setiap piksel citra input tersebut akan dikalikan dengan nilai *kernel* (pada gambar 2.11 ditunjukkan dengan tanda x kecil yang diikuti

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

4
---

Gambar 2.11 Operasi Konvolusi

nilai kernelnya) kemudian dijumlahkan keseluruhan nilainya. Hasil dari proses konvolusi citra tersebut dapat dilihat pada Gambar 2.11 kotak berwarna hijau.

Berikut ini contoh penyelesaian proses konvolusi dengan satu *stride* pada gambar 2.11:

- Pertama-tama seperti yang ditunjukkan pada Gambar 2.11 *filter* (kotak berwarna merah) ditaruh dipojok kiri atas. Contoh perhitungan konvolusi pada gambar 2.11 pada bagian yang tertimpa kernel (kotak berwarna merah) adalah sebagai berikut:

$$(1 \times 1) + (1 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) + (1 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) = 4$$

Angka yang berwarna biru adalah nilai citra, angka yang berwarna merah adalah nilai filter, dan angka yang berwarna hijau adalah nilai citra setelah dikonvolusi.

Gambar 2.12 Filter Berada di Tengah Atas

- b. Kemudian filter bergeser satu pixel ke kanan seperti pada contoh gambar 2.12.

1	1 <sub>x1</sub>	1 <sub>x0</sub>	0 <sub>x1</sub>	0
0	1 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	0
0	0 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	1
0	0	1	1	0
0	1	1	0	0

4	3
---	---

Perhitungan konvolusi pada gambar 2.12 pada bagian yang tertimpa kernel (kotak berwarna merah) adalah sebagai berikut:

$$(1 \times 1) + (1 \times 0) + (0 \times 1) + (1 \times 0) + (1 \times 1) + (1 \times 0) + (0 \times 1) + (1 \times 0) + (1 \times 1) = 3$$

- c. Filter bergeser satu pixel lagi ke kanan seperti pada gambar 2.13.

1	1	1 <sub>x1</sub>	0 <sub>x0</sub>	0 <sub>x1</sub>
0	1	1 <sub>x0</sub>	1 <sub>x1</sub>	0 <sub>x0</sub>
0	0	1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>
0	0	1	1	0
0	1	1	0	0

4	3	4
---	---	---

Gambar 2.13 Filter Berada di Ujung Kanan Atas

Perhitungan konvolusi pada gambar 2.13 pada bagian yang tertimpa kernel (kotak berwarna merah) adalah sebagai berikut:

$$(1 \times 1) + (0 \times 0) + (0 \times 1) + (1 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) + (1 \times 0) + (1 \times 1) = 4$$

- d. Selanjutnya filter berpindah keposisi awal namun bergeser satu pixel ke bawah seperti pada gambar 2.14.

1	1	1	0	0
0 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	1	0
0 <sub>x0</sub>	0 <sub>x1</sub>	1 <sub>x0</sub>	1	1
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	0

4	3	4
2		

Gambar 2.14 Filter Berada di Ujung Kiri Tengah

Perhitungan konvolusi pada gambar 2.14 pada bagian yang tertimpa kernel (kotak berwarna merah) adalah sebagai berikut:

$$(0x1)+(1x0)+(1x1)+(0x0)+(0x1)+(1x0)+(0x1)+(0x0)+(1x1) = 2$$

- e. Selanjutnya filter berpindah keposisi awal namun bergeser satu pixel ke bawah seperti pada gambar 2.15.

1	1	1	0	0
0	1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0
0	0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1
0	0 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0
0	1	1	0	0

4	3	4
2	4	

Gambar 2.15 Filter Berada di Tengah

Perhitungan konvolusi pada gambar 2.15 pada bagian yang tertimpa kernel (kotak berwarna merah) adalah sebagai berikut:

$$(1x1)+(1x0)+(1x1)+(0x0)+(1x1)+(1x0)+(0x1)+(1x0)+(1x1) = 4$$

- f. Selanjutnya filter yang berada ditengah tadi bergeser satu pixel ke kanan seperti pada gambar 2.16.

1	1	1	0	0
0	1	1 <sub>x1</sub>	1 <sub>x0</sub>	0 <sub>x1</sub>
0	0	1 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>
0	0	1 <sub>x1</sub>	1 <sub>x0</sub>	0 <sub>x1</sub>
0	1	1	0	0

4	3	4
2	4	3

Gambar 2.16 Filter Berada di Ujung Kanan Tengah

Perhitungan konvolusi pada gambar 2.16 pada bagian yang tertimpa kernel (kotak berwarna merah) adalah sebagai berikut:

$$(1x1)+(1x0)+(0x1)+(1x0)+(1x1)+(1x0)+(1x1)+(1x0)+(0x1) = 3$$

- g. Kemudian filter bergeser ke ujung kiri lalu turun satu pixel kebawah seperti pada gambar 2.17.

1	1	1	0	0
0	1	1	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0 <sub>x0</sub>	0 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0

4	3	4
2	4	3
2		

Gambar 2.17 Filter Berada di Ujung Kiri Bawah

Perhitungan konvolusi pada gambar 2.17 pada bagian yang tertimpa kernel (kotak berwarna merah) adalah sebagai berikut:

$$(0x1)+(0x0)+(1x1)+(0x0)+(0x1)+(0x0)+(1x1)+(1x0)+(0x1) = 2$$

- h. Kemudian filter bergeser ke kanan seperti pada gambar 2.18.

1	1	1	0	0
0	1	1	1	0
0	0 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	1
0	0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	0
0	1 <sub>x1</sub>	1 <sub>x0</sub>	0 <sub>x1</sub>	0

4	3	4
2	4	3
2	3	

Gambar 2.18 Filter Berada di Tengah Bawah

Perhitungan konvolusi pada gambar 2.18 pada bagian yang tertimpa kernel (kotak berwarna merah) adalah sebagai berikut:

$$(0x1)+(1x0)+(1x1)+(0x0)+(1x1)+(0x0)+(1x1)+(1x0)+(0x1) = 3$$

- i. Terakhir filter bergeser ke kanan menuju ujung kanan seperti pada gambar 2.19.

1	1	1	0	0
0	1	1	1	0
0	0	1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>
0	0	1 <sub>x0</sub>	1 <sub>x1</sub>	0 <sub>x0</sub>
0	1	1 <sub>x1</sub>	0 <sub>x0</sub>	0 <sub>x1</sub>

4	3	4
2	4	3
2	3	4

Gambar 2.19 Filter Berada di Ujung Kanan Bawah

Perhitungan konvolusi pada gambar 2.19 pada bagian yang tertimpa kernel (kotak berwarna merah) adalah sebagai berikut:

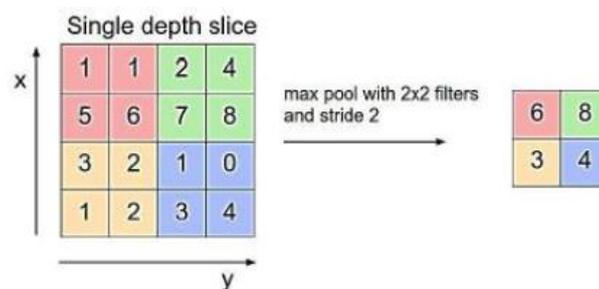
$$(1 \times 1) + (1 \times 0) + (1 \times 1) + (1 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (0 \times 1) = 4$$

Hasil konvolusi citra (kotak berwarna biru) secara keseluruhan dapat dilihat pada kotak berwarna hijau pada gambar 2.19. Kotak berwarna hijau tersebut akan diteruskan ke layer berikutnya.

Tujuan operasi konvolusi pada citra adalah untuk mengekstraksi fitur dari citra masukan. Konvolusi akan menghasilkan transformasi *linear* dari data *input* sesuai informasi spasial pada data. Bobot yang ada pada *layer* tersebut menspesifikasikan *kernel* konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan *input* pada CNN (Putra, dkk., 2016).

## 2) *Subsampling Layer*

*Subsampling* adalah proses untuk mereduksi ukuran dari sebuah data citra dan memiliki tujuan untuk meningkatkan invariansi posisi dari fitur citra. Pada sebagian besar CNN, metode *subsampling* yang digunakan adalah *max pooling*. *Max pooling* adalah proses membagi output dari *convolution layer* menjadi beberapa *grid* kecil kemudian mengambil nilai maksimal dari setiap *grid* untuk membangun matriks citra yang telah direduksi seperti ditunjukkan pada Gambar 2.20. Kotak – kotak yang berwarna merah, hijau, kuning dan biru adalah kelompok *grid* yang akan dipilih nilai maksimumnya. Hasil dari proses tersebut dapat dilihat pada kumpulan *grid* disebelah kanannya. Proses *max pooling* memastikan fitur yang didapatkan akan sama walaupun objek citra mengalami pergeseran (Putra, dkk., 2016)..



Gambar 2.20 Proses Max Pooling (Putra, dkk., 2016).

### 3) *Fully Connected Layer*

*Fully Connected Layer* adalah *layer* yang memiliki tujuan untuk mengklasifikasikan data secara linear dengan cara melakukan transformasi pada dimensi data. Agar data pada *convolution layer* dapat diproses oleh *fully connected layer*, *neuron* pada *convolution layer* perlu ditransformasi menjadi data satu dimensi sebelum dimasukkan ke dalam sebuah *fully connected layer*. Dengan adanya proses tersebut dapat menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, oleh karena itu *fully connected layer* hanya dapat diimplementasikan di akhir jaringan (Putra, dkk., 2016).

### 2.12.3 Fungsi Aktivasi

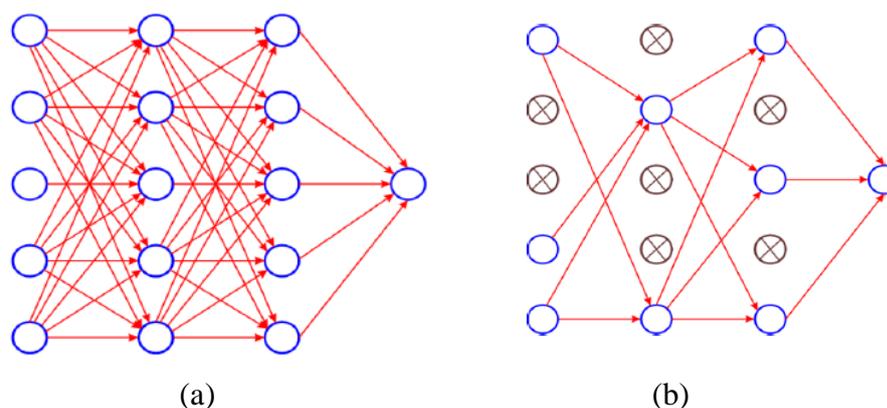
Menurut Zufar (2016) fungsi aktivasi atau fungsi transfer adalah fungsi non-linear yang memungkinkan sebuah jaringan untuk dapat menyelesaikan permasalahan non trivial. Dalam arsitektur CNN, fungsi aktivasi terletak sesudah proses perhitungan konvolusi atau pooling yang memiliki tujuan untuk menghasilkan suatu pola fitur. Ada beberapa fungsi aktivasi yang sering digunakan dalam penelitian yaitu fungsi sigmoid, tanh, Rectified Linear Unit (ReLU), Leaky ReLU(LReLU), Parametric ReLU, dan softmax.

Fungsi yang akan digunakan pada penelitian ini adalah ReLU dan *softmax*. ReLU atau Rectified Linear Unit menurut Dewi, Rusdinar, Pangaribuan, & Tiruan (2018) dikenal juga sebagai fungsi ramp dan analog dengan rektifikasi setengah gelombang dalam teknik elektro yang mana membuat pembatas pada bilangan nol, artinya apabila  $x \leq 0$  maka  $x = 0$  dan bila  $x > 0$  maka  $x$  adalah nilai input. Kemudian *softmax* menurut Bishop dan Christopher pada Dewi, dkk. (2018) adalah generalisasi fungsi logistik yang mencampurkan vektor K-dimensi vektor  $z$  dari nilai *real* secara acak ke K-dimensi vector  $\sigma(z)$  dari nilai real, yang mana keluarannya nanti ada pada rentang 0 sampai dengan 1.

### 2.12.4 Dropout

*Dropout* adalah salah satu cara untuk mencegah terjadinya *overfitting* dan juga bertujuan untuk mempercepat proses *learning* (Abhirawa, dkk. pada Santoso & Ariyanto, 2018). *Overfitting* adalah suatu kondisi yang mana hampir semua data yang telah melalui proses training mencapai persentase yang cukup baik, namun masih

terjadi ketidaksesuaian pada proses prediksi. *Dropout* bekerja untuk menghilangkan sementara suatu neuron yang berupa *Hidden Layer* maupun *Visible Layer* yang berada didalam jaringan secara acak (Santoso dan Ariyanto, 2018). Pada gambar 2.21 (a) dapat dilihat ketika jaringan belum melalui *dropout* dan pada gambar 2.21 (b) adalah hasil setelah *dropout*.



Gambar 2.21 (a) Sebelum *Dropout* (Santoso dan Ariyanto, 2018), dan (b) Sesudah *Dropout* (Santoso dan Ariyanto, 2018)

### 2.13 *Projection Profile-Based Histogram*

Pada umumnya, metode *Projection Profile* digunakan untuk menentukan baris pada dokumen yang mana spasi antar baris terkelompokkan dengan jelas. Namun, metode ini pun dapat digunakan pada tulisan tangan. Metode *projection profile*/profil proyeksi ini dapat dikelompokkan menjadi dua yaitu profil proyeksi secara vertikal atau horizontal. Profil proyeksi vertikal biasanya digunakan untuk segmentasi kata. Sedangkan profil proyeksi horizontal banyak digunakan untuk segmentasi baris. Namun tentu saja ini tergantung pada cara penulisan, misalnya kalimat mandarin disusun dari atas ke bawah (vertikal) (Paulus, Suryani, Hadi, & Hidayat, 2017).

Menurut Rodrigues (dalam Sunarya, Kesiman, & Purnami, 2017) cara kerja dari metode *Projection Profile-Based Histogram* adalah dengan menjadikan huruf atau objek kedalam bentuk garis-garis histogram vertikal dan horisontal. Metode *Projection Profile Histogram* ini berdasarkan pada penggunaan profil histogram yang telah tercipta dan diproporsikan oleh suatu citra. Profil proyeksi merupakan sebuah struktur data yang digunakan untuk menyimpan sejumlah piksel hitam yang

merupakan objek ketika suatu citra diproyeksikan melalui sumbu x maupun sumbu y. Perhitungan dilakukan dengan cara menghitung jumlah piksel hitam pada suatu citra secara mendatar (horisontal) dan juga secara menurun (vertikal).

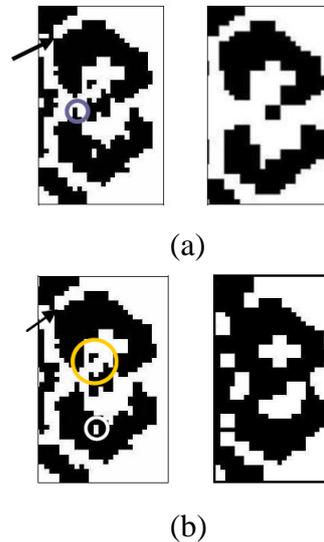
## 2.14 Matematika Morfologi

Menurut Dougherty dan Geoff pada Masfran & Nugroho (2012) matematika Morfologi (Mathematics Morphology) adalah sebuah metode yang mana citra diasumsikan tersusun dari himpunan piksel, metode ini digunakan untuk menganalisa citra yang didasarkan pada teori dasar matematika yaitu teori himpunan. Input operasi morfologi menggunakan dua himpunan yaitu citra (umumnya citra biner) dan sebuah kernel. Pada morfologi, kernel memiliki istilah lain yaitu *structuring elements* (elemen pembentuk struktur). SE adalah matrik dan pada umumnya berukuran kecil. 1, 0, don't care adalah beberapa nilai elemen yang dapat terkandung pada SE. Pada umumnya nilai don't care biasanya ditandai dengan nilai elemen yang kosong atau diberi tanda silang. Ada dua operasi dasar morfologi yaitu dilasi dan erosi. Dilasi digunakan untuk memperbesar batas dan menutup lubang, sedangkan erosi sebaliknya (Jamil, Sembok, & Bakar, 2008). Kedua operasi dasar tersebut menjadi dasar untuk membuat berbagai operasi morfologi yang sangat berguna untuk pengolahan citra digital, seperti *opening*, *closing*, dan lain – lain (Putra & Dama pada Masfran & Nugroho, 2012).

Menurut Jamil, dkk (2008) erosi dan dilasi dapat dilakukan berulang kali pada suatu gambar sampai output yang diinginkan dihasilkan. Namun, urutan operasi dapat membuat perbedaan pada gambar yang diproses. Dua operasi umum yang menggabungkan erosi dan dilasi ini dikenal sebagai *opening* dan *closing*.

*Opening operation* melibatkan erosi yang kemudian diikuti oleh dilasi, sedangkan operasi *closing* dimulai dengan dilasi yang kemudian diikuti oleh erosi. *Opening* menghaluskan kontur benda, menghilangkan sambungan yang tipis/kecil (ditunjukkan panah pada Gambar 2.22 (a)) dan menghilangkan tonjolan yang tipis (terlihat pada Gambar 2.22(a) yang ditandai dengan lingkaran). Lalu *closing* menghasilkan penghalusan bagian kontur tetapi menyambungkan patahan yang sempit (ditunjukkan oleh panah pada Gambar 2.22 (b)), mengisi celah dalam kontur (ditunjukkan oleh lingkaran kuning pada Gambar 2.22 (b)) dan menghilangkan lubang kecil (ditunjukkan oleh lingkaran putih pada Gambar 2.22 (b)). Kemudian menurut

Seul, dkk pada Jamil, dkk (2008) operasi *opening* digunakan ketika gambar memiliki banyak daerah noise kecil. Di sisi lain *closing*, digunakan untuk mengembalikan konektivitas antara objek yang memiliki jarak dekat



Gambar 2.22 (a) *After Opening Process* (Jamil, dkk, 2008), dan (b) *After Closing Process* (Jamil, dkk, 2008)

### 2.15 Presisi, *Recall*, dan *F-measure*

Menurut Powers & W. (2011) presisi atau *confidence* (seperti yang disebut pada *Data Mining*) adalah proporsi kasus dari *predicted positive* yang merupakan *real positive* (benar – benar positif). Berikut ini adalah persamaan presisi yang mana TP adalah *true positive* dan PP adalah *predicted positive* (semua data yang diprediksi positif):

$$\text{Presisi} = \text{TP} / \text{PP} \quad (3)$$

Lalu menurut Powers (2011) *recall* atau *sensitivity* (biasa disebut dalam psikologi) adalah proporsi kasus *real positive* yang diprediksi *positive*. Berikut ini adalah persamaan Recall yang mana TP adalah *true positive* dan RP adalah *real positive*:

$$\text{Recall} = \text{TP} / \text{RP} \quad (4)$$

Selain itu ada juga *F1-Score* atau biasa disebut juga dengan *F-Measure* adalah rata-rata harmonik dari Presisi dan *Recall*. *F-Measure* memberikan gambaran seberapa presisi dan seberapa baik model dalam memberikan label

klasifikasi (J. Santoso et al., 2018). Menurut Rijsbergen pada Hamzah (2017) persamaan *F-Measure* adalah sebagai berikut:

(5)

Dengan  $\beta$  sebagai parameter kepentingan relative aspek Presisi dan *Recall*. Jika Recall (R) dan Presisi (P) memiliki bobot yang sama pentingnya, maka  $\beta=1$  sehingga

parameter F-measure

$$F\text{-measure} = \frac{(\beta^2 + 1) PR}{\beta^2 P + R}$$

menjadi:

$$F\text{-measure} = \frac{2 PR}{P + R}$$

(6)

Contoh kasus misalnya ada suatu sistem untuk mendeteksi sepatu bermerek. Sistem ini akan diuji coba dengan 100 sepatu bermerek dan 20 sepatu tidak bermerek. Pada proses uji coba sistem memprediksi ada 80 sepatu bermerek. Dari 80 sepatu yang diprediksi ternyata yang benar-benar sepatu bermerek ada 60 buah dan sisanya bukan. Sistem ini akan dinilai dengan melihat nilai presisi, *recall*, dan *f-measure*nya (kepentingan recall dan presisi sama) pada sepatu bermerek. Berikut ini adalah acara menghitung nilai-nilai tersebut:

$$TP=60$$

$$PP=80$$

$$RP=100$$

$$\text{Presisi} = 60/80 = 0.75$$

$$\text{Recall} = 60/100 = 0.6$$

$$F\text{-measure} = (2 \times 0.75 \times 0.6) / (0.75 + 0.6) = 0.67$$

Jadi nilai presisi sistem pada sepatu bermerek adalah 0.75, *recall* 0.6, dan *f-measure* 0.67.

## 2.16 Bahasa Pemrograman *Python*

Menurut Winarno, Dirgantara, & Yulisusianto (2017) python adalah bahasa pemrograman interpretatif multiguna yang mana filosofi perancangannya berfokus pada tingkat keterbacaan kode. Python ini diklaim sebagai bahasa yang mampu menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas

serta dilengkapi dengan fungsionalitas pustaka standar yang besar dan komprehensif. Python mendukung multi paradigma pemrograman, namun tidak dibatasi pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Python memiliki banyak fitur salah satunya adalah fitur python sebagai bahasa pemrograman yang dinamis serta dilengkapi dengan manajemen memori otomatis. Python pada umumnya digunakan sebagai bahasa skrip seperti halnya pada bahasa pemrograman dinamis lainnya, walaupun pada praktiknya penggunaan bahasa ini lebih luas cakupannya, python pemanfaatannya mampu mencakup hal yang umumnya tidak dilakukan dengan menggunakan bahasa skrip juga. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak. Berikut ini beberapa platform sistem operasi yang dapat menjalankan python di antaranya adalah:

- 1) Linux/Unix.
- 2) Windows.
- 3) Mac OS X.
- 4) Java Virtual Machine.
- 5) OS/2.
- 6) Amiga.
- 7) Symbian (untuk produk-produk Nokia).

Kemudian menurut Winarno, dkk. (2018) kelebihan dari bahasa pemrograman Python antara lain adalah bahasanya mudah untuk digunakan dalam mengembangkan sebuah produk, baik itu web, software, aplikasi web, maupun video game, dan juga mendukung ekosistem Internet of Things dengan sangat baik. Internet of Things adalah sebuah teknologi yang mampu menghubungkan benda-benda di sekitar kita ke dalam sebuah jaring-jaring yang menghubungkan.