

BAB II TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi adalah suatu perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu. Istilah aplikasi ini sendiri diambil dari bahasa Inggris *application* yang dapat diartikan sebagai penerapan atau penggunaan (Pane, Zamzam and Fadillah, 2020). Pada penelitian ini jenis aplikasi yang digunakan adalah aplikasi *desktop* pada gate parkir untuk keluar masuk kendaraan karyawan. Dalam pengembangannya, aplikasi dapat dikategorikan dalam tiga kelompok, diantaranya :

- a. Aplikasi *desktop*, yaitu aplikasi yang hanya dijalankan diperangkat komputer atau laptop.
- b. Aplikasi *web*, yaitu aplikasi yang dijalankan menggunakan komputer dan koneksi internet.
- c. Aplikasi *mobile*, yaitu aplikasi yang dijalankan diperangkat *mobile*.

2.2 Citra Digital

Citra adalah suatu gambaran atau kemiripan dari suatu objek. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses oleh komputer secara langsung. Agar dapat diproses oleh komputer, cità analog harus dikonversi menjadi citra digital.

Citra digital adalah citra yang dapat diolah oleh komputer. Sedangkan citra yang dihasilkan dari peralatan digital (citra digital) langsung bisa diolah oleh komputer. Penyebabnya adalah di dalam peralatan digital terdapat sistem *sampling* dan *kuantisasi*. Sedangkan peralatan analog tidak dilengkapi kedua sistem tersebut. Sistem *sampling* adalah sistem yang mengubah citra kontinu menjadi citra digital dengan cara membagi citra analog menjadi M baris dan N kolom, sehingga menjadi citra diskrit. Semakin besar nilai M dan N, semakin

halus citra digital yang dihasilkan. Pertemuan antara baris dan kolom disebut piksel.

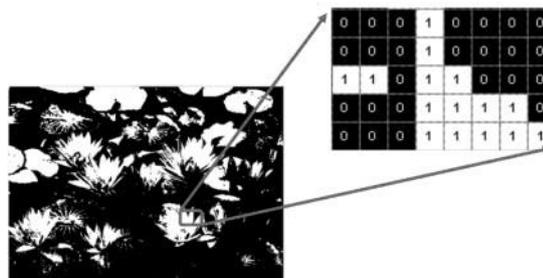
Sistem kuantisasi adalah sistem yang melakukan perubahan intensitas analog ke intensitas diskrit, sehingga dengan proses ini dimungkinkan untuk membuat gradasi warna sesuai dengan kebutuhan. Kedua sistem inilah yang bertugas untuk memotong-motong citra menjadi M baris dan N kolom (proses sampling) sekaligus menentukan besar intensitas yang terdapat di titik tersebut (proses kuantisasi), sehingga menghasilkan resolusi citra yang diinginkan (Andono, Sutojo and Muljono, 2017).

2.3 Tipe Citra

Beberapa tipe citra yang sering digunakan untuk penelitian adalah (Andono, Sutojo and Muljono, 2017):

a. Citra Biner

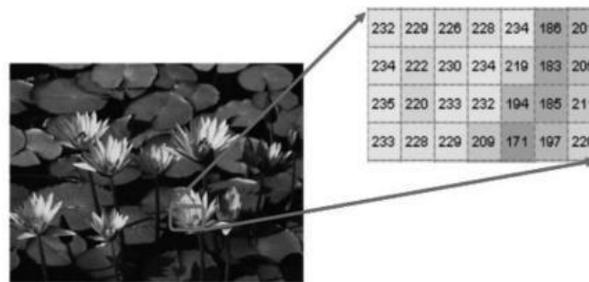
Pada citra biner, tiap-tiap piksel hanya membutuhkan 1 bit memori. Maka, dengan demikian, setiap piksel hanya mempunyai 2 (dua) buah kemungkinan nilai intensitas, yaitu 1 atau 0. Gambar 2.1 menunjukkan citra biner dilihat dari dekat dengan beberapa nilai intensitas piksel.



Gambar 2.1 Citra Biner dengan Nilai Piksel 0 atau 1

b. Citra *Grayscale*

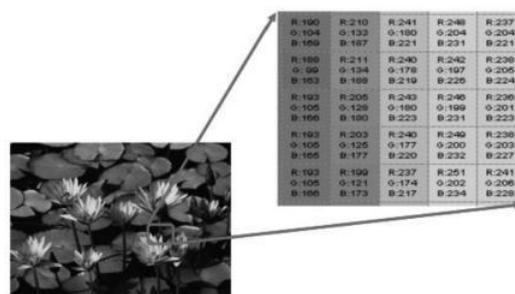
Citra *grayscale* adalah matriks data yang nilai-nilainya mewakili intensitas setiap piksel berkisar antara 0 sampai dengan 255. Setiap piksel membutuhkan 8 bit memori. Gambar 2.2 menunjukkan citra *grayscale* dilihat dari dekat dengan beberapa nilai intensitas piksel.



Gambar 2.2 Citra *Grayscale* dengan Nilai Pixel Antara 0 sampai 255

c. Citra Warna

Citra warna adalah citra yang masing-masing piksel mempunyai 3 (tiga) komponen warna yang spesifik, yaitu komponen merah (red), hijau (green) dan biru (blue). Warna setiap piksel ditentukan oleh kombinasi dari intensitas warna merah, hijau dan biru yang disimpan pada bidang warna di lokasi piksel. Format *file* grafis menyimpan citra warna sebagai citra 24 bit, yang berasal dari komponen merah, hijau dan biru masing-masing 8 bit. Hal ini menyebabkan citra warna mempunyai 24 juta kemungkinan warna. Gambar 2.3 menunjukkan citra warna dilihat dari dekat dengan beberapa nilai intensitas piksel.



Gambar 2.3 Citra Warna dengan KOMPonen Warna RGB

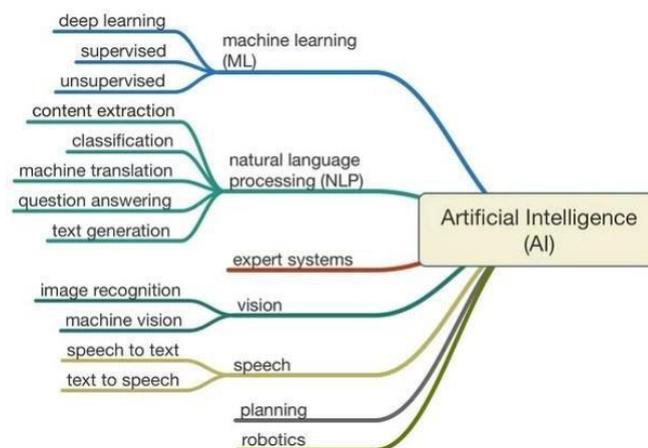
2.4 Machine Learning

Machine learning dapat didefinisikan sebagai aplikasi komputer dan algoritma matematika yang diadopsi dengan cara pembelajaran yang berasal dari data dan menghasilkan prediksi di masa yang akan datang. Adapun proses pembelajaran yang dimaksud adalah suatu usaha dalam memperoleh kecerdasan yang melalui

dua tahap antara lain latihan (*training*) dan pengujian (*testing*) (Roihan, Sunarya and Rafika, 2020).

Bahasa pemrograman yang digunakan untuk membuat *machine learning* adalah *Python* yang merupakan bahasa pemrograman tingkat tinggi yang berfokus pada keterbacaan kode. *Python* bersifat *open source* artinya dapat digunakan oleh siapa saja untuk mengembangkan program (Heryadi and Irwansyah, 2020).

Kecerdasan buatan pada pengaplikasiannya secara garis besar terbagi tujuh cabang, yaitu *machine learning*, *natural language processing*, *expert system*, *vision*, *speech*, *planning*, dan *robotics*. Percabangan dari kecerdasan buatan tersebut dimaksudkan untuk mempersempit ruang lingkup saat pengembangan atau belajar AI, karena pada dasarnya kecerdasan buatan memiliki ruang lingkup yang sangat luas.



Gambar 2.4 Cabang Pengaplikasian AI

Pada penelitian ini, *machine learning* yang digunakan adalah jenis *deep learning* dengan algoritma CNN. *Deep learning* atau *deep structured learning/hierarchical learning* adalah bagian dari kecerdasan buatan dan *machine learning*, yang merupakan pengembangan dari *neural network multiple layer* untuk memberikan ketepatan tugas. Contohnya seperti deteksi objek, pengenalan suara, terjemahan bahasa, dan lain sebagainya.

Meski menjadi subbidang dari *machine learning*, *deep learning* ternyata memiliki teknik yang berbeda karena *deep learning* secara otomatis melakukan

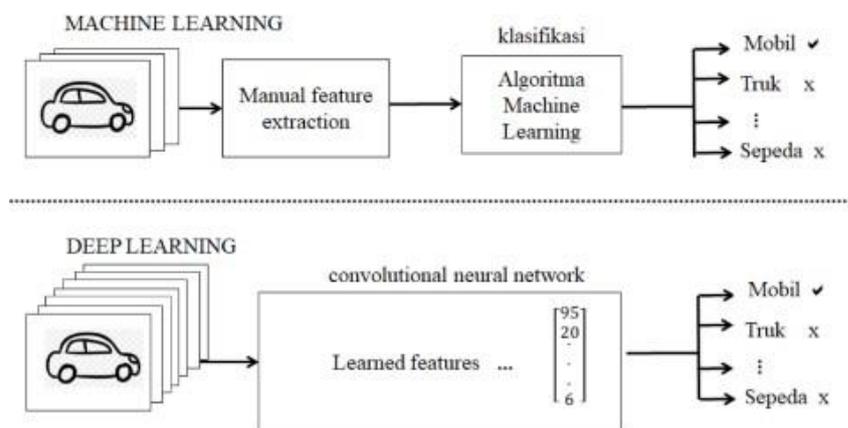
representasi dari data, seperti gambar, video, hingga teks tanpa memperkenalkan aturan kode atau pengetahuan domain manusia.

Setelah memahami pengertian dan contoh dari deep learning, penting juga untuk mengetahui apa saja jenis-jenis algoritma *deep learning*. Melihat dari beberapa sumber, jenis-jenis algoritma *deep learning* terdiri dari beberapa macam, seperti:

- Convolutional Neural Network* (CNN)
- Recurrent Neural Network* (RNN)
- Long Short Term Memory Network* (LSTM)
- Self Organizing Maps* (SOM)

2.5 Deep Learning

Deep learning merupakan bagian dari *machine learning* yang melakukan pembelajaran secara hirarki. Metode ini merupakan pengembangan dari *artificial neural network* pada *machine learning*. *Deep learning* melakukan pembelajaran yang terdiri dari banyak layer. Layer awal menghasilkan fitur sederhana hingga layer akhir menghasilkan fitur-fitur yang lebih kompleks. *Deep Learning* diaplikasikan pada teknik klasifikasi, clustering, segmentasi ataupun *recognition*. *Deep Learning* cocok untuk data yang tidak teratur seperti teks, suara dan citra.



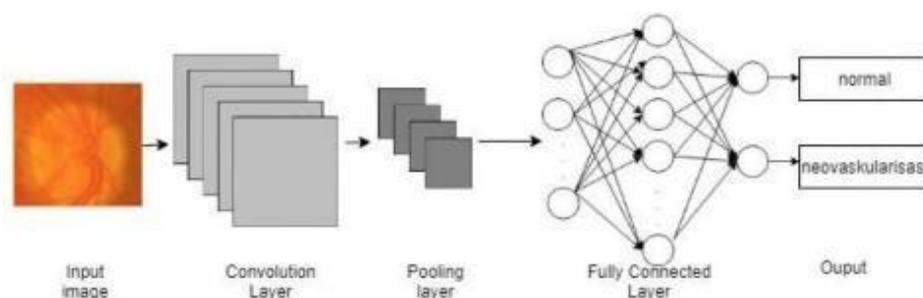
Gambar 2.5 *Machine Learning vs Deep Learning*

Gambar di atas menunjukkan perbandingan data yang digunakan, akurasi pengenalan yang dapat dicapai oleh manusia, *machine learning* konvensional, dan *deep learning*. *Deep learning* memiliki tingkat akurasi lebih tinggi dibandingkan lainnya. Pada *deep learning*, semakin banyak *data training* yang digunakan, semakin tinggi akurasi pengenalan yang didapatkan.

2.6 Algoritma CNN

Dalam bidang *deep learning*, *Convolutional Neural Network* (CNN) merupakan bagian dari *deep neural network*, yakni jenis jaringan saraf tiruan yang umumnya digunakan dalam pengenalan dan pemrosesan gambar. Algoritma ini dirancang khusus untuk memproses data piksel dan citra visual (Arrofiqoh and Harintaka, 2018). Hal ini yang membuat CNN cukup efektif digunakan dalam pemrosesan gambar dibanding algoritma *neural network* sejenis.

Kata *convolution* pada CNN yaitu sebuah operasi matematika pada dua buah fungsi yang kemudian menghasilkan fungsi ketiga. Fungsi ini menggabungkan 2 buah himpunan informasi dan menunjukkan bagaimana bentuk satu fungsi dimodifikasi oleh fungsi lainnya (Nur and Winarsih, 2018). CNN menggunakan *convolution* sebagai pengganti perkalian matriks umum. Lapisan pada algoritma CNN memiliki *neuron* yang diatur dalam 3 dimensi: *width*, *height*, dan *depth*. Dimensi *depth* mengacu pada dimensi ketiga dari fungsi aktivasi, bukan kedalaman *neural network* atau jumlah total layer dalam jaringan.

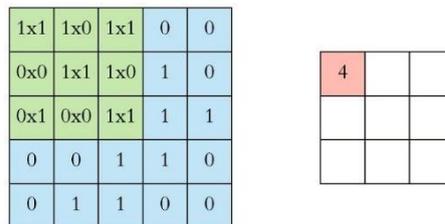


Gambar 2.6 Arsitektur CNN (*Convolutional Neural Network*)

2.6.1 Convolution Layer

Operasi *convolution* merupakan bagian penting dari CNN. Pada algoritma CNN, *convolution* dijalankan pada data *input* menggunakan sebuah filter atau kernel yang kemudian digunakan untuk memetakan fitur. Operasi *convolution* dilakukan dengan menggeser filter di atas *input*. Di setiap lokasi, perkalian matriks dilakukan dan menjumlahkan hasilnya ke dalam peta fitur (Setiawan, 2021).

Gambar di bawah menunjukkan cara kerja operasi *convolution*. Filter ditunjukkan dengan blok berwarna hijau bergerak di atas *input* (ditandai kotak berwarna biru) dan jumlah dari hasil operasi *convolution* masuk ke peta fitur (kotak merah).

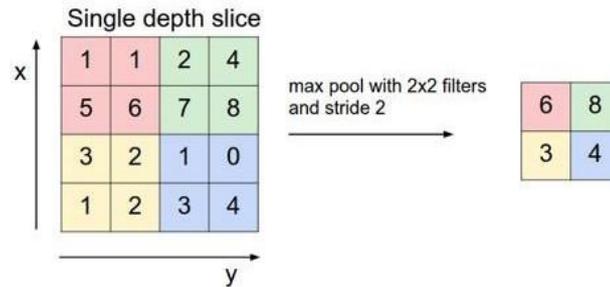


Gambar 2.7 Cara Kerja Convolution Layer

2.6.2 Pooling

Setelah *convolution layer*, biasanya ditambahkan lapisan penyatuan (*pooling layer*) di antara *layer* CNN. Fungsi *pooling* adalah untuk mengurangi dimensi secara terus menerus serta mengurangi jumlah parameter dan komputasi dalam jaringan. Hal ini akan mempersingkat waktu *training* dan mengontrol terjadinya *overfitting*.

Jenis *pooling* yang paling sering digunakan pada CNN adalah *max pooling*. *Pooling* ini mengambil nilai maksimum di setiap *window*/blok. Ukuran *window*/blok ini perlu ditentukan sebelumnya. Operasi ini mengurangi ukuran peta fitur dan hanya akan menyimpan informasi penting dan signifikan dalam proses klasifikasi nantinya.



Gambar 2.8 Cara Kerja *Pooling Layer*

Ketika menggunakan CNN, ada 4 hyperparameter penting yang perlu kita tentukan, yakni:

- Ukuran kernel/filter, yaitu panjang dan lebar filter yang akan digunakan. Misalnya 3x3
- Jumlah filter, yaitu seberapa banyak filter yang akan kita gunakan
- Stride*, yaitu seberapa jauh step dari filter ketika digeser. Pada proses konvolusi, *stride* merupakan suatu parameter dimana bisa menentukan memutuskan berapa jumlah pergeseran filter. Jika *stride* nya sebanyak 2, jadi *convolutional filter* bergeser sejauh 2 *pixels* secara garis horizontal lalu secara garis vertikal. Semakin kecil *stride* maka akan semakin detail juga suatu informasi yang kita dapatkan dari masukan, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar (Harani and Hasanah, 2020).
- Padding*, adalah suatu parameter dimana dapat menentukan/memilih berapa banyak jumlah *pixels* yang (berisi nilai 0) yang akan ditambahkan pada setiap sisi dari masukan. Ini dipergunakan dengan tujuan untuk memanipulasi atau memalsukan dimensi keluaran dari **conv.layer** (*FeatureMap*). Tujuan menggunakan atau memanfaatkan *padding* adalah supaya dimensi *output* tetap sama dengan dimensi *input* atau tidak berkurang secara drastis. Karena dimensi *output convolutional layer* kerap lebih kecil dari pada hasil *input* nya dan hasil *output* ini yang akan kita gunakan kembali untuk masukan dari *convolutional layer* berikutnya, sehingga semakin banyak informasi yang terbuang.

2.6.3 Fully Connected

Setelah melalui *convolution layer* dan *pooling layer*, komponen selanjutnya adalah klasifikasi yang terdiri dari beberapa *fully-connected layer* (layer yang terhubung secara penuh). Layer ini hanya dapat menerima data berdimensi 1. Untuk mengonversi data 3 dimensi menjadi 1 dimensi, bisa menggunakan fungsi *flatten*. *Neuron* pada *fully-connected* memiliki koneksi menyeluruh ke semua aktivasi di layer sebelumnya.

2.6.4 Python

Code Python adalah bahasa pemrograman interpretatif yang bisa dipasang pada berbagai *platform*, khususnya *platform* yang berfokus pada keterbacaan kode. Kode *Python* bisa di-*embed* ke bahasa lain seperti C dan Java, atau sebaliknya, dari bahasa C atau Java ke *Python*.

Pemrograman *Python* itu merupakan salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multu guna secara langsung dengan metode *Object Oriented Programming* dan menggunakan sergantik dinamis untuk memberikan tingkat keterbacaan sintak. *Pemrograman Python* memiliki bahasa yang kemampuan, menggabungkan kapabilitas dan sintaksis kode yang jelas dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Pemrograman bahasa *Python* difokuskan untuk digunakan dalam menganalisis data, visualisasi data, membuat dan mengembangkan AI. Pemrograman *Python* adalah pemrograman yang paling mudah di pelajari dengan *code* yang pendek dan tidak susah. *Python* memiliki pustaka (*library*) yang luas dan dapat dikembangkan ke bidang-bidang lainnya. Beberapa *library python* yang populer dalam *Data Science* dan AI adalah *Scikit- Learn, TensorFlow, PyTorch*.

Pemrograman bahasa *Python* merupakan bahasa pemrograman yang tidak menggunakan *compiler*. Dengan sifat open-source yang dimilikinya, pengguna dapat mempelajari *Python* dengan mudah karena bahasa ini dapat dugunakan dalam membuat situs.mengembangkan situs, mengembangkan *video game*,

membangun *GUI Desktop*, dan mengembangkan perangkat lunak (Setiawan and Vania, 2022).

2.7 Kotlin

Kotlin adalah sebuah bahasa pemrograman dengan *Statically typed* (tipe statis) yang berjalan pada platform *Java Virtual Machine* (JVM). Kotlin menggunakan compiler LLVM yang artinya, dapat dikompilasi ke dalam kode *JavaScript*. Pengembang utamanya berasal dari tim programer *JetBrains* yang bermarkas di Rusia. Bahasa pemrograman yang satu ini banyak diminati oleh para *developer*. Adapun kelebihan Kotlin adalah (Aljundi and Akbar, 2018) :

- a. *Concise* : Kotlin mampu mengurangi *boilerplate of code* atau tingkat kerumitan dari kode yang biasa kita tulis, ketika menggunakan bahasa *Java*.
- b. *Safe* : Kotlin mampu menjamin bahwa setiap *syntax* yang kita tulis secara proses kompilasi dapat mencegah kemungkinan terjadinya *error*, misalnya mampu mencegah terjadinya *NullPointerException* ketika kita *coding* menggunakan bahasa *Java*.
- c. *Versatile* : *Kotlin* sejatinya sama seperti *Java*, karena memang *kotlin* itu sendiri di turunkan dari bahasa induknya, yaitu *Java*. Sehingga *kotlin* juga dapat di pakai dalam pengembangan aplikasi di *Web*, *Desktop*, maupun *Mobile*.
- d. *Interoperable* : *Kotlin* tidak sama seperti bahasa *Java* turunan lainnya (misal; *Scala* ataupun *Clojure*) yang tidak dapat dijalankan bersamaan dengan kode yang kita tulis menggunakan *Java*. *Kotlin* mampu membaca kode lama atau *library* yang kita gunakan atau kita tulis dengan bahasa *Java* dan begitupun sebaliknya.

2.8 Database

Database merupakan gabungan informasi yang memuat di dalam komputer secara terancang sehingga bisa dikontrol memakai suatu program komputer. Secara persepsi, *database* ialah gabungan dari data-data yang menata suatu arsip ataupun *file* yang saling berinteraksi dengan prosedur yang spesifik untuk memuat data baru maupun informasi (Subandi and Syahidi, 2018). Pengertian lain dari *database* yakni gabungan dari data yang saling berinteraksi antara satu dengan

yang lainnya yang penataan menurut rancangan ataupun sistem tertentu (Rosa and Shalahuddin, 2018). Adapun struktur *database* terdiri dari :

a. Data

Sekumpulan fakta mengenai objek tertentu, orang dan lain-lain yang dinyatakan dengan angka, huruf, gambar, film, suara dan sebagainya yang relevan dan belum mempunyai arti.

b. Informasi

Hasil pengolahan data yang konkrit dan sudah mempunyai arti untuk mencapai suatu tujuan tertentu.

c. Tabel

Merupakan hal yang paling mendasar dalam hal penyimpanan data yang terdiri dari *field* dan *record*.

d. *Field*

Merupakan elemen dari tabel yang berisikan informasi tertentu yang spesifik tentang sub judul tabel pada sebuah item data. Syarat-syarat pembentukan Field Name pada tabel adalah:

1. Harus Unik atau Spesifik
2. Boleh disingkat

Pemisah sebagai pengganti spasi dalam pembentuk *field* adalah tanda lambang “_”. Contoh: Kode Barang menjadi KdBarang, KodeBrg, Kd_Brg, Kd_Barang.

Dalam sistem manajemen basisdata, terdapat tiga macam *field*:

1. Harus diisi (*required*)
2. Dapat diabaikan (*optional*)
3. Penghitungan dari field lainnya (*calculated*). Pengguna tidak dapat memasukkan data pada jenis *field* yang terakhir (*calculated*).

e. *Query*

Query adalah pertanyaan atau permintaan informasi tertentu dari sebuah basis data yang ditulis dalam format tertentu. Terdapat tiga metode utama untuk membuat *query*, yaitu:

1. *Query by example* (QBE) adalah metode query yang disediakan sistem dalam bentuk *record* kosong dan pengguna dapat menentukan *field* dan nilai tertentu yang akan digunakan dalam *query*.
2. Bahasa *query* (*query language*) adalah bahasa khusus yang digunakan untuk melakukan *query* pada sebuah basisdata. Metode ini paling rumit tetapi paling fleksibel.

f. Record

Sekumpulan data yang saling berkaitan tentang sebuah subjek tertentu, misalnya data seorang siswa akan disimpan dalam *record* yang terdiri dari beberapa kolom/*field*. Struktur *database* adalah cara data diorganisasi agar pemrosesan data menjadi lebih efisien. Sistem manajemen basis data (DBMS) adalah suatu aplikasi peranti lunak yang menyimpan struktur basis data-data itu sendiri, hubungan diantara data dalam basis data, dan nama-nama formulir, jenis-jenis data, angka dibelakang desimal, jumlah karakter, nilai-nilai *default* dan seluruh uraian *field* lainnya.

2.9 *Black Box Testing*

Black box testing merupakan pengujian untuk mengetahui fungsi perangkat lunak yang telah berjalan sesuai dengan kebutuhannya (Jurnal (ABDI) and Rahardja, 2020). Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan pengeluaran perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Rosa and Shalahuddin, 2018). *Black box testing* mencoba untuk menemukan kesalahan dalam kategori berikut :

- a. Fungsi tidak benar atau hilang
- b. Kesalahan *interface* atau antarmuka
- c. Kesalahan dalam struktur data atau akses *database* eksternal

2.10 Penelitian Terkait

Penelitian terdahulu untuk menunjang penelitian yang sedang dilakukan saat ini adalah sebagai berikut :

- a. Penelitian ini berusaha membangun sebuah sistem pengenalan candi sekitar Yogyakarta untuk membantu memudahkan masyarakat umum dalam

mengenali candi selain candi Borobudur menggunakan bantuan model *machine learning* berbasis *cloud*. Sebelum terbentuk model pengenalan candi, telah dilakukan pengujian terhadap dataset sebanyak 50 gambar dengan pembagian 80 banding 20. Dari hasil tersebut menghasilkan nilai *confidence* sebesar rata-rata 95 %. Dari hasil tersebut dapat membuktikan bahwa model pengenalan candi dengan *machine learning* berbasis *cloud* ini dapat digunakan untuk pengenalan candi dengan baik (Fatih and Miharja, 2022).

- b. *Machine learning* merupakan sub dari bidang keilmuan kecerdasan buatan (*Artificial Intelligence*) yang banyak diteliti dan digunakan untuk memecahkan berbagai masalah. Ulasan dari berbagai bidang disajikan dalam bentuk pemecahan masalah dan algoritmanya dan dibagi menjadi tiga kategori dalam *machine learning* antara lain *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Ulasan dibatasi hanya pada beberapa bidang dan hasilnya menunjukkan bidang yang paling dominan terkini adalah bidang kedokteran atau medis diantara beberapa bidang lain seperti industri, teknologi dan lalu lintas (Roihan, Sunarya and Rafika, 2020).