

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Dalam penelitian ini Penulis menggunakan beberapa penelitian terdahulu yang membahas pengenalan wajah dengan metode Local Binary Pattern (LBP) atau Local Binary Pattern Histogram (LBPH). Untuk menambahkan gambaran penelitian penulis dibandingkan dengan penelitian yang sudah dilakukan, maka dibawah ini diuraikan tentang penelitian terdahulu yang relevan. Beberapa penelitian yang berkaitan dengan masalah penelitian ini adalah:

1. Jurnal yang ditulis oleh Qadriza Mutiara Detila dan Eri Prasetyo Wibowo yang berjudul: *“Perbandingan Metode Eigenface, Fisherface, dan LBPH pada Sistem Pengenalan Wajah”* Vol. 18, No. 4, Desember 2019 menjelaskan tentang perbandingan 3 metode untuk pengenalan wajah. Qadriza Mudiara Detila dkk. melakukan perbandingan dengan cara membandingkan gambar wajah dengan 4 kondisi yang berbeda yaitu tingkat kecerahan rendah, tingkat kecerahan normal, wajah tidak berekspresi, dan wajah tersenyum. Hasil dari jurnal ini adalah nilai persentase keberhasilan dari 4 kondisi yang ditentukan sebelumnya dengan rata-rata 46% untuk Eigenface, 54,25% untuk Fisherface dan 88,75% untuk LBPH. Dari hasil tersebut maka dapat

disimpulkan bahwa metode LBPH lebih unggul dibandingkan Eigenface dan Fisherface.

2. Jurnal yang ditulis oleh Fajar Setiawan dan Dewi Agushinta R yang berjudul: *“Sistem pengenalan wajah dengan metode local binary pattern histogram pada firebase berbasis opencv”* Vol. 4, No. 1, September 2020 membahas tentang bagaimana mencari wajah yang cocok berdasarkan *database* yang ada. Fajar Setiawan dkk. melakukan penelitian untuk mengukur tingkat keberhasilan metode LBPH dalam mengenali wajah. Sistem yang dibuat berbasis desktop dengan *firebase* sebagai *database*. Hasil yang diperoleh dari penelitian ini sistem pengenalan wajah yang memiliki tingkat keberhasilan 93,5%.
3. Jurnal yang ditulis oleh Li Wang and Ali Akbar Siddique yang berjudul: *“Facial recognition system using LBPH face recognizer for anti-theft and surveillance application based on drone technology”* Mei 2020 membahas tentang pengenalan wajah untuk pengawasan dan pencegahan pencurian berbasis *drone*. Sistem pengenalan wajah yang dibuat akan berjalan di *Rassberry PI* yang akan dipasang pada *drone*. Hasil dari penelitian tersebut adalah sistem pengenalan wajah yang memiliki tingkat akurasi sebesar 89,1%.
4. Jurnal yang ditulis oleh Aftab Ahmed, Jiandong Guo, Fayaz Ali, Farha Deebe, Awais Ahmed yang berjudul: *“LBPH Based Improved Face*

*Recognition at Low Resolution*” Mei 2018 membahas tentang mendeteksi gambar wajah dengan resolusi yang kecil (35px - 45px). Dalam penelitian yang ditulis oleh Aftab Ahmed dkk. Peneliti menggunakan *dataset* LR500 dan menguji dengan 2500 gambar yang diambil oleh webcam beresolusi 1080px. Dari 2500 gambar 98.8% berhasil dikenali. Untuk hasil akurasi ukuran wajah 45px sebesar 94% dan 35px sebesar 90%.

5. Jurnal yang ditulis oleh Sayeed Al-Aidid dan Daniel S. Pamungkas yang berjudul: “*Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram*” Vol.14, No. 1, April 2018. Sayeed Al-Aidid dan Daniel S. Pamungkas melakukan penelitian untuk mengetahui apakah mengenali wajah menggunakan *Haar cascade classifier* dan metode LBPH mampu mendeteksi wajah meskipun di sekitar wajah manusia terdapat gangguan objek lain. Dalam penelitian ini juga dibahas seberapa jauh sistem dapat mengenali wajah. Hasil dari penelitian Sayeed Al-Aidid dkk. adalah sistem pengenalan wajah dapat mendeteksi wajah meskipun terdapat gangguan objek lain dan dapat mendeteksi wajah paling jauh sekitar 150cm.

## **2.2 Teori Utama Penelitian**

### **2.2.1 Kecerdasan Buatan (*Artificial Intelligence*)**

Kecerdasan buatan adalah cabang ilmu komputer yang menjelaskan bagaimana komputer dapat meniru pemikiran manusia.

Kecerdasan buatan memungkinkan komputer untuk menarik kesimpulan dan mengidentifikasi masalah dengan cara manusia. Dalam buku yang ditulis oleh Dr. Hendra Jaya, S.Pd., M.T., dkk. Dengan judul “Kecerdasan Buatan” mengemukakan bahwa kecerdasan buatan merupakan salah satu bidang ilmu komputer yang membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik apa yang manusia lakukan [9].

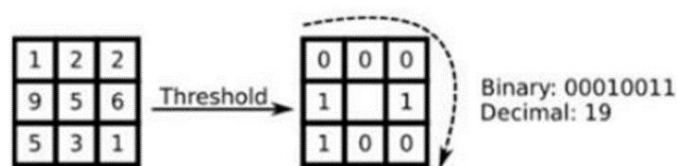
Menurut Dr. Hendra Jaya, S.Pd., M.T., dkk. Kecerdasan buatan dapat dipandang dalam 4 sudut pandang yaitu sudut pandang kecerdasan, penelitian, bisnis, dan pemrograman [9]. Berikut beberapa contoh penerapan kecerdasan buatan (*Artificial Intelligence*), diantaranya:

1. Sistem kemudi otomatis pada mobil, contohnya TESLA pabrikan mobil dari Amerika Serikat memproduksi mobil *autopilot*.
2. Asisten virtual seperti Siri pada Apple, Bixbi pada Samsung, dan Alexa pada Amazon.
3. Sistem rekomendasi, seperti rekomendasi produk di *e-commerce* dan rekomendasi iklan pada youtube.

### **2.2.2 Local Binary Pattern Histogram (LBPH)**

Untuk memahami apa itu Local Binary Pattern Histogram (LBPH) kita harus memahami dahulu apa itu Local Binary Pattern

(LBP) karena LBPH merupakan gabungan nilai yang dihasilkan LBP operator dari sebuah citra gambar. Operator LBP diterapkan untuk menggambarkan kontras informasi dari suatu piksel ke piksel tetangganya. Operator LBP asli didefinisikan di jendela 3\*3. Menggunakan nilai piksel median sebagai ambang jendela, dibandingkan dengan nilai abu-abu dari 8 piksel yang berdekatan. Jika nilai piksel tetangga lebih besar atau sama dengan nilai median piksel, nilai posisi piksel ditandai sebagai 1, sebaliknya ditandai sebagai (0) [9]. Fungsi didefinisikan seperti yang ditunjukkan pada persamaan 1. Dapat diilustrasikan seperti pada Gambar 2.1.



Gambar 2. 1 Operator LBP Asli  
Sumber: Ahmed et al (2018)

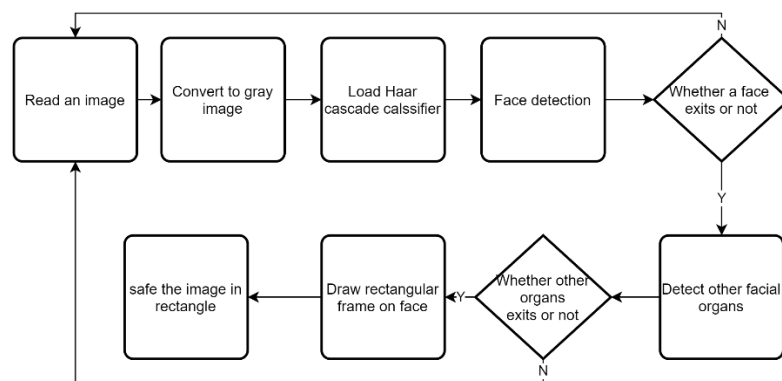
Setelah diubah menjadi nilai biner maka akan menghasilkan deret biner yang kemudian jika diubah menjadi nilai desimal akan menghasilkan nilai baru untuk nilai tengah piksel selanjutnya, deret biner tersebut yang disebut dengan kode LBP. Kode LBP yang dihasilkan tersebut direpresentasikan dengan histogram. Histogram akan menunjukkan frekuensi kejadian dari berbagai nilai LBP. Secara umum perhitungan LBP ini bisa dilakukan seperti persamaan 1.1 [10]:

$$LPB(x_c, y_c) = \sum_{n=0}^7 (i_n - i_c)^{2n} \quad (1.1)$$

Pada persamaan 1.1 menjelaskan operator LBP asli dengan  $i_c$  adalah nilai piksel dari titik pusat  $(x_c, y_c)$ , sedangkan  $i_n$  merupakan nilai piksel tetangga dari  $i_c$ . Berikut adalah alur pengenalan wajah menggunakan metode LBPH [11]:

### 1. Deteksi Wajah (*Face Detection*)

Untuk deteksi wajah, penulis menggunakan OpenCV dan *Haar cascade classifier*.



Gambar 2. 2 Flowchart deteksi wajah

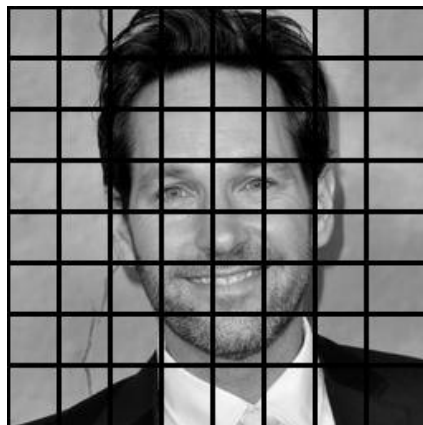
Sumber: A.Singh et al (2019)

Pada Gambar 2.2 menjelaskan tentang alur dari deteksi wajah. Langkah pertama yaitu membaca gambar kemudian dirubah ke gambar abu-abu, setelah itu lakukan deteksi wajah menggunakan *Haar cascade classifier* hingga wajah ditemukan dan disimpan menjadi gambar wajah.

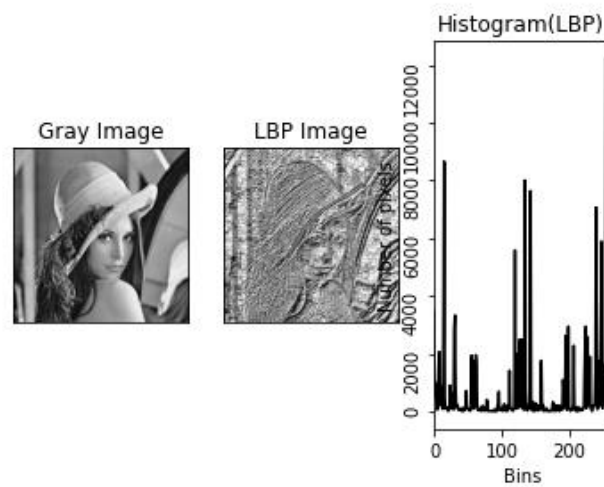
### 2. Ekstraksi Fitur (*Feature Extraction*)

Untuk mengekstrak fitur wajah dari citra digunakan operasi LBP yang membandingkan nilai intensitas setiap komponen dengan nilai 8 piksel tetangga terdekat. Jika nilai piksel tetangga lebih besar dari nilai piksel tengah, itu akan menetapkan 1 ke piksel tetangganya, jika tidak maka akan menetapkan 0. Untuk setiap piksel, tugas ini menyediakan string 8-bit. Nilai desimal dari string piksel 8-bit menentukan nilai LBP. Gambar 2.1 menunjukkan operasi ini.

Gambar input dibagi menjadi banyak sub-gambar kecil seperti pada Gambar 2.3. Setelah penerapan operator LBP dan histogram nilai LBP dari setiap sub-gambar diekstraksi. Kemudian semua histogram dihubungkan seperti pada Gambar 2.4 untuk membuat vektor fitur yang merepresentasikan citra dan digunakan untuk melatih pengklasifikasi pengenalan wajah.



Gambar 2. 3 Pembagian gambar



Gambar 2. 4 Histogram LBPH

### 3. Membuat Dataset

Penulis merancang sendiri dataset yang akan dibuat, dataset berisi 3 gambar untuk setiap wajah. Dataset akan dilatih setiap ada user yang mendaftarkan wajahnya ke sistem absen ini. Gambar 2.5 menunjukan contoh gambar dalam dataset.



Gambar 2. 5 Contoh gambar dalam dataset

### 4. Pengenalan Wajah (*Face Recognition*)

Dalam langkah ini gambar akan diproses menggunakan operator LBP seperti pada Gambar 2.1. Setelah perhitungan menggunakan operator LBP selesai dan menghasilkan



Histogram, maka akan dibandingkan dengan dataset yang sudah ada.

### 2.2.3 OpenCV

OpenCV adalah kependekan dari *opensource computer vision*. OpenCV adalah sebuah kumpulan *tools*, *library*, dan *module* yang berkaitan dengan *computer vision*. Dalam OpenCV terdapat kumpulan algoritma yang dinamakan “*Haar Cascade Classifier*” untuk mendeteksi objek khususnya wajah manusia. Dengan menggunakan *Haar Cascade Classifier* penulis tidak kesulitan untuk mendeteksi wajah dalam sebuah gambar atau video.

### 2.2.4 Computer Vision

*Computer Vision* adalah bagian dari komputer yang menggambarkan bagaimana komputer terlihat seperti manusia, dan karena ini terkait erat dengan penglihatan, cahaya juga merupakan faktor penting dalam hal ini. [12].

### 2.2.5 Python

Python adalah bahasa pemrograman berorientasi objek yang ditafsirkan, interaktif. Ini menggabungkan modul, pengecualian, pengetikan dinamis, tipe data dinamis tingkat sangat tinggi, dan kelas. Ini mendukung beberapa paradigma pemrograman di luar pemrograman berorientasi objek, seperti pemrograman prosedural dan fungsional. Python menggabungkan

kekuatan yang luar biasa dengan sintaks yang sangat jelas. Ini memiliki antarmuka ke banyak panggilan sistem dan perpustakaan, serta berbagai sistem jendela, dan dapat dikembangkan dalam C atau C ++. Ini juga dapat digunakan sebagai bahasa ekstensi untuk aplikasi yang membutuhkan programmable antarmuka. Terakhir, Python bersifat portabel: ia berjalan di banyak varian Unix termasuk Linux dan macOS, dan di Windows.

#### 2.2.6 Amazon Web Service (AWS)

Ada banyak definisi dan interpretasi komputasi awan yang dapat ditemukan dari berbagai sumber. Istilah "komputasi awan" itu sendiri kemungkinan berasal dari diagram jaringan di mana *cloud* bentuk digunakan untuk menggambarkan jenis jaringan tertentu, baik Internet atau jaringan internal. Beberapa sumber menyebut komputasi awan sebagai sekumpulan aplikasi yang dikirimkan sebagai gabungan layanan dengan perangkat keras dan perangkat lunak pusat data yang memungkinkan aplikasi. Yang lain mengatakan awan itu komputasi adalah model bisnis daripada teknologi atau layanan tertentu [13].

Amazon Web Service (AWS) adalah penyedia layanan *cloud computing*. Menurut situs resminya yaitu <https://aws.amazon.com/>, AWS menyediakan lebih dari 200

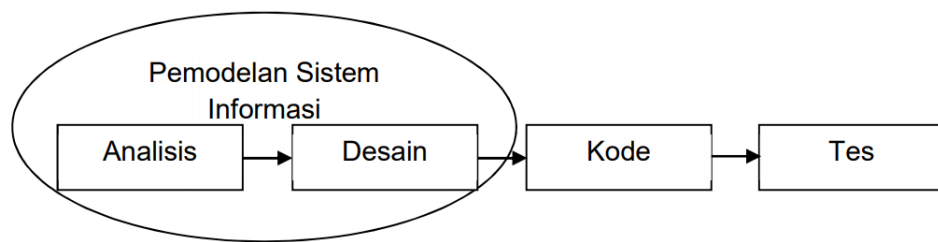
service seperti penyimpanan, *compute*, *internet of things*, *machine learning*, dan masih banyak lainnya.

## **2.3 Metode Pengembangan Perangkat Lunak**

### **2.3.1 Waterfall**

Pengembangan perangkat lunak meliputi metode, proses, dan alat yang memudahkan dalam proses pengembangan perangkat lunak dari tahap satu ke tahap lainnya agar menghasilkan perangkat lunak yang berkualitas [14].

Model pengembangan perangkat lunak merupakan gambaran dari proses membuat suatu *software*. Menurut Pressman [14] Model Waterfall merupakan suatu model pengembangan secara sekuensial. Model Waterfall bersifat sistematis dan berurutan dalam membangun sebuah perangkat lunak. Proses pembuatannya mengikuti alur dari mulai analisis, desain, kode, pengujian dan pemeliharaan. Model pengembangan waterfall memiliki beberapa kelebihan, antara lain: dapat mudah dipahami dan dapat diterapkan dalam proses pengembangan perangkat lunak. Gambar 2.6. merupakan model pengembangan Waterfall.



Gambar 2. 6 Model Pengembangan Waterfall (Pressman 2002)

### 2.3.2 Tahapan-tahapan Waterfall

Tahapan-tahapan dalam metode waterfall adalah sebagai berikut [14]:

1. Analisis kebutuhan perangkat lunak.

Tahap ini merupakan tahap untuk mengumpulkan kebutuhan perangkat lunak. Tujuan dari tahap ini yaitu untuk mengetahui apa saja kebutuhan untuk membuat sebuah perangkat lunak atau *software*. *Analisis kebutuhan perangkat lunak ini perlu didokumentasikan* untuk dilihat kembali oleh pengguna apakah kebutuhan sudah sesuai atau tidak.

2. Desain

Desain merupakan tahapan yang berfokus pada perangkat lunak yang meliputi arsitektur, struktur data, antarmuka perangkat lunak, dan prosedur perngkodean. Ada beberapa pemodelan perangkat lunak yang umum digunakan, salah satu pemodelan perangkat lunak yang digunakan dalam tahapan ini

yaitu Unified Modeling Language (UML), yang merupakan gambaran mengenai perangkat lunak yang akan dibuat. UML dibuat untuk mempermudah pengembang dalam membuat suatu perangkat lunak.

### 3. Pembuatan Kode

Dalam tahap ini desain yang sebelumnya telah dibuat akan diterapkan kedalam sebuah kode. Penerjemahan desain ke dalam kode haruslah lengkap dan berjalan dengan baik, Hasil dari tahap ini adalah sebuah sistem informasi yang sesuai dengan desain yang telah dibuat.

### 4. Pengujian

Tahapan biasanya ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. pengujian melibatkan pembetulan kesalahan-kesalahan yang tidak ditemukan pada tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

## 2.4 Tools Perancangan

### 2.4.1 Flowchart

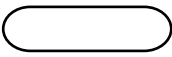


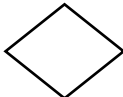



*Flowchart* menurut Drs. Suarga, M.Sc, Math, Ph.D, adalah untaian simbol atau gambar yang menunjukkan aliran dari data.

Dalam bukunya juga yang berjudul ALGORITMA (LOGIKA PEMROGRAMAN), Drs. Suarga, M.Sc, Math, Ph.D, membagi symbol *flowchart* menjadi 2 jenis [15], yaitu:

1. Simbol *flowchart* program

*Flowchart* Program merupakan simbol-simbol *flowchart* yang digunakan untuk menggambarkan logika dari pemrosesan terhadap data ditunjukkan pada Tabel 2.1.




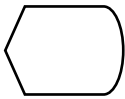

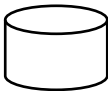



Tabel 2. 1 Simbol Flowchart Program

| Simbol  | Nama Simbol               | Keterangan                                    |
|---|---------------------------|---|
|  | terminator                | Menunjukkan awal/akhir sistem                 |
|  | proses                    | Menunjukkan operasi yang dilakukan sistem     |
|  | <i>Input/output</i>       | Menerima input atau menampilkan output        |
|  | pilihan                   | Memilih aliran berdasarkan syarat             |
|  | <i>Predefined process</i> | Menunjukkan fungsi atau <i>sub-program</i>    |
|  | penghubung                | Menunjukkan penghubung pada halaman yang sama |
|  | penghubung                | Menunjukkan penghubung pada halaman yang lain |

## 2. Simbol *flowchart* sistem

*Flowchart* sistem merupakan simbol-simbol peralatan sistem komputer yang digunakan untuk menyatakan proses pengolahan data ditunjukkan pada Tabel 2.2.

Tabel 2. 2 Simbol Flowchart Program

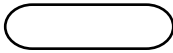

| Simbol  | Nama Simbol          |
|---|----------------------|
|    | <i>keyboard</i>      |
|   | <i>printer</i>       |
|  | <i>File/storage</i>  |
|  | <i>Display</i>       |
|  | <i>Magnetic tape</i> |
|  | <i>Magnetic disk</i> |
|  | <i>sorting</i>       |
|  | <i>extract</i>       |
|  | <i>merge</i>         |

### 2.4.2 Flowmap


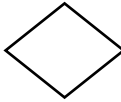



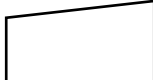
*Flowmap* adalah campuran peta dan *flowchart* yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain [16]. *Flowmap* sendiri sering disebut diagram alir atau aliran data yang berbentuk dokumen dalam sebuah aktifitas yang saling berkaitan antara kebutuhan data dan informasi. *Flowmap* dapat membantu programmer untuk memecahkan masalah yang besar ke dalam segmen atau bagian yang lebih kecil.

Fungsi flowmap adalah mendefinisikan hubungan antara bagian (pelaku proses), proses (manual/berbasis komputer) dan aliran data (dalam bentuk dokumen masukan dan keluaran) [16]. Simbol Flowmap dalam Tabel 2.3.

Tabel 2. 3 Simbol Flowmap

| Simbol  | Keterangan   |
|---|--|
|  | <i>Terminator Symbol</i><br>(Simbol yang menunjukkan awal/akhir dari sistem)               |
|  | <i>Process Symbol</i><br>(Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer) |



|   |  |
|---|--|
|    | <i>Simbol Document</i><br>(Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak kekertas) |
|    | <i>Decision Symbol</i><br>(Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau aksi)                 |
|    | <i>Predefined process Symbol</i> (Simbol untuk menjelaskan proses dalam <i>chart</i> tersendiri)                               |
|  | <i>Connector Symbol</i><br>(Simbol untuk keluar atau masuk prosedur atau proses dalam lembar atau halaman)                     |
|  | <i>Off-line Connector Symbol</i> (Simbol untuk keluar atau masuk proses pada halaman yang lain)                                |
|  | <i>Input/output Symbol</i><br>(Simbol yang dapat menerima input atau menampilkan output)                                       |

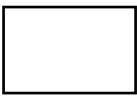
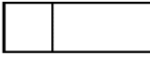



### 2.4.3 Data Flow Diagram (DFD)

Pada akhir 1970an diagram aliran data (DFD) diperkenalkan dan dipopulerkan untuk analisis dan desain terstruktur (Gane dan Sarson 1979). DFD menunjukkan aliran data dari entitas eksternal

ke dalam sistem, menunjukkan bagaimana data dipindahkan dari satu proses ke proses lainnya, serta penyimpanan logisnya [17].

Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Sistem dibatasi oleh boundary. Dalam diagram konteks hanya ada satu proses. Tidak boleh ada store dalam diagram konteks [16]. Diagram konteks terdiri dari beberapa simbol [16], yang ditunjukan pada Tabel 2.4

Tabel 2. 4 Simbol Data Flow Diagram (DFD)

| Simbol  | Keterangan   |
|---|--|
|  | <i>External entity</i> adalah sumber atau tujuan aliran data suatu sistem          |
|  | <i>Data store</i> adalah tempat penyimpanan data                                   |
|  | <i>Process</i> , menggambarkan bagaimana <i>input</i> diubah menjadi <i>output</i> |
|  | <i>Boundary</i> , merupakan Batasan dari sebuah sistem                             |
|  | Aliran data, menggambarkan aliran dari suatu proses ke proses lainnya              |

## 2.5 Tools Perangkat Lunak

### 2.5.1 Visual Studio Code

Visual Studio Code adalah *text editor* yang dikembangkan oleh Microsoft yang dapat berjalan di Windows, Linux dan MacOS. Ini termasuk dukungan untuk *debugging*, *GIT Control* yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema *editor*, *shortcut keyboard*, dan preferensi. Visual Studio Code gratis dan *opensource*, meskipun unduhan resmi berada di bawah lisensi MIT.

### 2.5.2 AWS Lambda

AWS Lambda atau biasa disebut Lambda merupakan layanan dari Amazon Web Service yang memungkinkan kita menjalankan kode program tanpa harus memikirkan infrastruktur dari server seperti tidak perlu memilih Sistem Operasi. Lambda berjalan oleh sebuah *trigger*, *trigger* tersebut dapat berupa *event* dari *endpoint*, *event* dari *S3 bucket*, maupun dari sumber lainnya.

### 2.5.3 Amazon Simple Cloud Storage (S3)

Amazon S3 atau biasa disebut S3 adalah kependekan dari *Simple Cloud Storage* yang merupakan layanan penyimpanan dari Amazon Web Service. Amazon S3 memiliki kelebihan yaitu memiliki ketahanan yang tinggi, sangat mudah digunakan dan memiliki penyimpanan dan *bandwidth* yang tidak terbatas [18].

Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja terdepan di industri. Ini berarti pelanggan dari semua ukuran dan industri dapat menggunakannya untuk menyimpan dan melindungi data dalam jumlah berapa pun untuk berbagai kasus penggunaan, seperti data lake, situs web, aplikasi seluler, pencadangan dan pemulihan, arsip, aplikasi perusahaan, perangkat IoT, dan besar analitik data. Amazon S3 menyediakan fitur manajemen yang mudah digunakan sehingga Anda dapat mengatur data dan mengonfigurasi kontrol akses yang disesuaikan untuk memenuhi persyaratan bisnis, organisasi, dan kepatuhan spesifik Anda. Amazon S3 dirancang untuk ketahanan 99,999999999% (11 9), dan menyimpan data untuk jutaan aplikasi untuk perusahaan di seluruh dunia.

#### **2.5.4 Amazon DynamoDB**

Amazon DynamoDB atau biasa disebut Dynamodb merupakan layanan penyimpanan data milik AWS. DynamoDB merupakan database NoSql yang memiliki ketahanan yang tinggi. Amazon DynamoDB adalah database nilai kunci dan dokumen yang memberikan kinerja milidetik satu digit pada skala apa pun. Ini adalah database yang terkelola sepenuhnya, multi-wilayah, multi-aktif, dan tahan lama dengan keamanan bawaan, pencadangan dan pemulihan, dan penyimpanan dalam memori untuk aplikasi skala

internet. DynamoDB dapat menangani lebih dari 10 triliun permintaan per hari dan dapat mendukung puncak lebih dari 20 juta permintaan per detik.

#### **2.5.5 Amazon Elastic Compute Cloud (EC2)**

Amazon EC2 atau biasa disebut EC2 adalah kependekan dari *Elastic Compute Cloud* yang merupakan layanan dari Amazon Web Service yang memungkinkan kita memiliki paket komputer lengkap dalam *cloud* dalam hitungan detik [18].

Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang aman dan dapat diubah ukurannya di cloud. Ini dirancang untuk membuat komputasi awan skala web lebih mudah bagi pengembang. Antarmuka layanan web sederhana Amazon EC2 memungkinkan Anda memperoleh dan mengonfigurasi kapasitas dengan gesekan minimal. Ini memberi Anda kendali penuh atas sumber daya komputasi Anda dan memungkinkan Anda berjalan di lingkungan komputasi Amazon yang telah terbukti.

#### **2.5.6 NodeJs**

Node.JS adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web. Node.JS dapat juga disebut sebagai runtime environment. Aplikasi ini ditulis dalam campuran Bahasa C++ dan juga JavaScript, mempunyai model *event driven* (basis

event) dan asynchronous I/O. Tidak seperti kebanyakan bahasa JavaScript yang dijalankan pada web browser, Node.JS dieksekusi sebagai aplikasi server. Node.JS dapat berjalan di server karena dukungan dari *V8 Engine* buatan Google dan beberapa modul bawaan yang terintegrasi seperti modul *http*, modul *filesystem*, modul *security* dan beberapa modul penting lainnya.

### **2.5.7 React Js**

React.js adalah salah satu web *framework* paling populer di dunia javascript, Dokumentasinya yang lengkap dan penggunaannya yang cukup mudah, dapat membuat kita mengembangkan berbagai produk seperti aplikasi web ataupun RESTful API dan dapat digunakan menjadi pijakan untuk membangun web *framework* yang lebih kompleks.