

BAB II LANDASAN TEORI

2.1. Penelitian Terdahulu

Pada era modern ini, perkembangan teknologi sangatlah pesat salah satunya adalah teknologi *self-driving* pada robot ataupun pada kendaraan. *Self-driving robot* merupakan robot yang mengendarai dirinya sendiri dengan pembelajaran menggunakan data salah satu pembelajarannya berupa *machine learning*.

Penelitian ini telah banyak dilakukan bahkan sudah ada produk mobil yang dapat bergerak sendiri tanpa pengemudi. Salah satu penelitian yang membahas tema serupa dilakukan oleh John M. Pierre [1]. Penelitian yang dilakukan adalah simulasi kendaraan menggunakan algoritma yang disebut sebagai *Correction-Based Incremental Learning* (CBIL) dimana menambahkan jumlah *training* secara strategis pada saat kendaraan melakukan kesalahan, dan menggulangi iterasi-iterasi dengan tujuan untuk memperbaiki secara drastis rata-rata waktu yang diperlukan. Hasil yang didapatkan berupa CBIL lebih efektif dibandingkan dengan *supervised learning* karena membutuhkan *training* yang lebih sedikit [1].

Terdapat juga penelitian yang dilakukan oleh Ahmad El Sallab et al. [2], mengenai pembelajaran menggunakan *reinforcement learning* untuk memenuhi 3 syarat pengembangan kendaraan otomatis yaitu : *Recognition, Prediction, and Planning*. *Recognition* untuk mengetahui lingkungan sekitar kemudian dilanjutkan dengan *prediction* untuk prediksi lingkungan yang akan dilalui dan yang terakhir *planning* untuk melakukan aksi mengemudi secara berurutan. Hasil yang didapatkan dari mengintegrasikan model perhatian ke dalam *framework*, menggunakan pemanfaatan jaringan sekilas dan aksi pengarahan langsung dari kernel CNN ke *input* yang berhubungan dengan proses pengemudi, mampu memperbaiki komputasi yang kompleks [2].

Penelitian mengenai pembelajaran mesin juga dilakukan oleh Fatih Ertam et al. [3], mengenai pembelajaran *deep learning* yang juga merupakan pembelajaran mesin dengan menggunakan tensorflow. Fungsi-fungsi yang digunakan untuk pembelajaran antara lain *Rectified Linear Unit* (ReLU), *Exponential Linear Unit* (eLu), *Hyperbolic Tangent* (tanH), sigmoid, softsign dan softplus. Pada pembelajarannya digunakan CNN dan SoftMax *classifier* untuk *Artificial Neural Network*. Hasil klasifikasi dengan akurasi tertinggi didapatkan menggunakan aktivasi fungsi ReLu [3].

Penelitian pengklasifikasi objek dengan kendaraan otomatis dilakukan oleh Hongbo Gao et al. [4], untuk mendeteksi penglihatan dan cahaya dalam sebuah lingkungan menggunakan metode CNN dan teori *image upsampling*. Data CNN didapatkan dengan menghubungkan RGB dengan *data upsampling* LIDAR yang dikonversikan menjadi kedalaman level piksel dalam sebuah *cloud*. Metode yang digunakan dipastikan menaikkan akurasi untuk mengklasifikasi objek dan tingkat *loss* yang minimal [4].

Penelitian untuk mengklasifikasi objek juga dilakukan oleh Nitin R. Gavai et al. [5], untuk mengklasifikasi bunga-bunga dimana pengklasifikasian bunga sangat penting dalam bidang tanaman. Tingkat kesulitan dalam pengklasifikasian bunga berupa species bunga, tekstur, warna dan ketidaksamaan antara species bunga yang sama. Eksperimen dilakukan pada *platform* tensorflow untuk melatih ulang pembelajaran mesin pada *dataset* kategori bunga, dan didapatkan waktu pelatihan dan penggunaan memori yang minimal dan tingkat akurasi yang tinggi [5].

Penelitian pengenalan wajah umum dilakukan salah satunya diteliti oleh Xiao Ling Xia et al. [6]. Pengenalan wajah harus memerhatikan ekspresi *emotion* yang berpengaruh terhadap *background*, *brightness*, dan faktor lainnya. Penelitian ini berbasis pada model *Inception-v3* pada *platform* tensorflow, digunakan teknik *transfer learning* untuk melatih *dataset* ekspresi wajah. Hasil yang didapatkan mampu mempertahankan tingkat akurasi dan mengurangkan banyak waktu *training* [6].

CNN yang merupakan gabungan *Artificial Neural Network* (ANN) dengan strategi *deep learning* mampu digunakan untuk mengenali *pattern*, *speech*, wajah, teks, tulis tangan, dsb. Sebuah penelitian mengenai pengenalan tulisan tangan dilakukan oleh Fathma Siddique et al. [7], untuk mengobservasi tingkat akurasi dengan menggunakan variasi jumlah *hidden layers* dan *epoch*. Eksperimen dilakukan dengan menggunakan *dataset* MNIST (*Modified National Institute of Standards and Technology*), dan dilatih menggunakan *stochastic gradient descent* dan algoritma *backpropagation* [7].

Penelitian menggunakan tensorflow dengan CNN dilakukan oleh Arwa Mohammed Taqi et al. [8], untuk mengklasifikasi penyakit *Alzheimer*. *Network* yang dipakai terdiri atas 3 *layer* convulasi untuk mengekstrak fitur dari penyakit *Alzheimer*, sebuah *layer flattening* untuk mengurangi dimensi, dan dua *fully connected layer* untuk mengklasifikasi fitur yang terekstrak. Kegunaan tensorflow pada penelitian ini untuk membuat grafik *computasi*. Untuk meningkatkan performansi klasifikasi data, peneliti menggunakan *data augmentation* untuk mengurangi *overfitting* dan peneliti juga menggunakan *multi-optimizer*. Beberapa *optimizer* digunakan untuk mendapatkan tingkat akurasi pada TF-CNN antara lain Adam, Adagrad, ProximalAdagrad, dan RMSProp. Hasil yang didapatkan adalah *loss* pada Adam dan RMSProp lebih kecil dibandingkan dengan *loss* pada Adagrad dan ProximalAdagrad, dan akurasi yang didapatkan menggunakan Adam adalah 95.8% sedangkan akurasi pada RMSProp mencapai 100% [8].

Penelitian pengenalan wajah juga dilakukan oleh Liping Yuan et al. [9] dengan menggunakan *deep learning*. CNN yang berbasis pada tensorflow digunakan untuk mengenali wajah. Hasil yang didapatkan memiliki tingkat pengenalan yang lebih baik dan *robustness* yang lebih baik dilingkungan yang kompleks [9].

Penelitian terhadap TF-CNN juga diteliti oleh Zixian Zeng et al. [10] untuk mengenali gerakan atau *gesture*. Penelitian digunakan menggunakan *dataset* yang sudah dikenali dan *dataset* yang diambil sendiri. Hasil yang didapatkan memiliki pengenalan dengan akurasi yang tinggi, *komputasi* yang efisien, *robustness* yang kuat, mampu menyesuaikan diri ke struktur *network*, pengoptimal model, dan mengenali *gesture* lebih baik [10].

2.2. Machine Learning

Machine Learning merupakan pelatihan untuk membantu *software* melakukan sesuatu tanpa pemrograman atau aturan yang eksplisit. Dalam pemrograman tradisional, programmer harus menentukan aturan yang digunakan, jadi sangat berbeda dengan *Machine Learning*. *Machine Learning* ini sangat mengfokuskan terhadap analisis data, dimana pemrograman menyediakan serangkaian contoh dan komputer mempelajari pola dari data tersebut. Proses pembelajaran *Machine Learning* dimulai dengan mengobservasi data seperti contoh-contoh, pengalaman langsung, dan arahan hingga mengerti pola dari data supaya dapat membuat keputusan yang lebih baik di masa yang akan datang sesuai dengan data yang diberikan. *Machine Learning* dapat disebut sebagai pemrograman dengan data. Tujuan utama dari *Machine Learning* adalah komputer mampu belajar dengan sendiri tanpa bantuan dari manusia dan penyesuaian tindakan.

Machine Learning memiliki beberapa metode, antara lain:

1. *Supervised machine learning algorithms* – mengaplikasikan apa yang telah dipelajari menggunakan contoh yang diberikan label. Metode ini mampu mendeteksi inputan baru setelah *training* yang cukup banyak.
2. *Unsupervised machine learning algorithms* – informasi untuk dipelajari tidak diklasifikasi ataupun diberikan label. Metode ini mempelajari fungsi yang tersembunyi dalam informasi yang diberikan. Metode ini tidak memberikan keluaran yang benar tetapi mengeksplorasi data dan membuat kesimpulan dari data.
3. *Reinforcement machine learning algorithms* – mempelajari dengan berinteraksi dengan lingkungan sekitar dengan aksi dan mencari *error* ataupun keberhasilan. *Trial and error* merupakan karakteristik utama metode ini. Metode ini memungkinkan mesin dan *software* untuk menentukan tindakan yang ideal supaya memaksimalkan kinerja dari sistem.

Salah satu library yang dapat digunakan untuk memprogram dengan metode *Machine Learning* adalah TensorFlow.

2.2.1 TensorFlow

TensorFlow adalah sebuah sistem *machine learning* yang mengoperasikan dalam skala besar dan dalam lingkungan yang *heterogeneous*. TensorFlow menggunakan grafik *dataflow* untuk merepresentasikan komputasi, pembagian keadaan, dan operasi pemutasian keadaan tersebut. Ini memetakan *node* dalam grafik aliran data pada banyak mesin dalam sebuah *cluster*, dan juga pada beberapa perangkat komputasi dari setiap mesin tersebut, termasuk *Central Processing Unit* (CPU) *multicore*, *Graphics Processing Unit* (GPU) *generalpurpose*, dan *Application-specific Integrated Circuit* (ASIC) dengan rancangan khusus yang dikenal sebagai *Tensor Processing Unit* (TPU). Arsitektur ini memberikan fleksibilitas terhadap *developer* untuk eksperimen optimasi baru dan algoritma *training*. TensorFlow mendukung variasi antar aplikasi, yang mengfokuskan dalam *training* dan interferensi *deep neural networks*. [11]



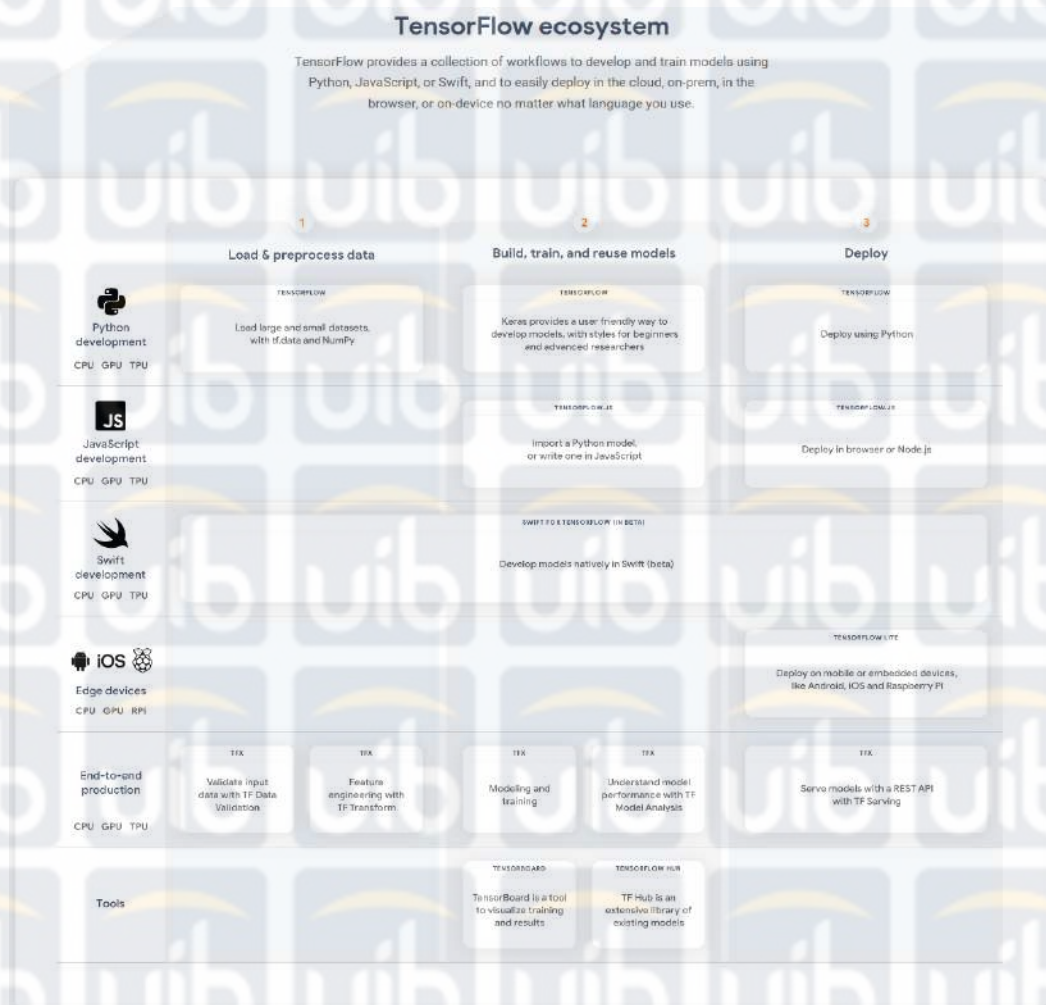
Gambar 2.1 TensorFlow Logo

(Source :

<https://upload.wikimedia.org/wikipedia/commons/1/11/TensorFlowLogo.svg>)

(17 Oktober 2019)

TensorFlow adalah library yang dapat diimplementasikan dalam beberapa jenis pemrograman yang dapat dilihat pada gambar berikut ini.



Gambar 2.2 TensorFlow Ecosystem

(Source : <https://www.tensorflow.org/learn>)

2.2.2 Keras

Keras merupakan *neural network* API level tinggi yang bersifat *open source* dan juga umumnya ditulis dalam kode program *python*. *Keras* membuat *prototyping* lebih mudah dan cepat seperti *user friendly*, *extensibility*, dan *modularity*. *Keras* bersifat *user friendly* karena mengikuti praktik terbaik dalam mengurangi beban kognitif dengan menggunakan API yang konsisten dan *simple*, mengurangi aksi *user* yang diperlukan, dan menyediakan *user error* yang jelas dan *feedback* yang dapat diberi aksi. *Keras*

juga bersifat *extensibility* dimana modul baru dapat dengan mudah ditambahkan, dengan mudahnya penambahan modul baru memberikan akses terhadap *total expressiveness* artinya *keras* sangat cocok untuk penelitian tingkat lanjut. *Keras* juga bersifat *modularity* karena modul-modul yang telah dikonfigurasi dapat digabung dengan restriksi yang sedikit, seperti *neural layers*, *optimizer*, *activation function* dapat digabung menjadi sebuah model baru.

Keras mendukung *convolutional neural network*, *recurrent neural network*, dan juga mendukung gabungan kedua *neural network* tersebut. *Keras* bekerja secara mulus pada CPU dan GPU. Jadi, *keras* sangat membantu dalam perancangan proyek pengenalan posisi objek yang menggunakan bantuan *platform tensorflow* dan metode *convolutional neural network*.



Gambar 2.3 Keras Logo

(Sumber : <https://pngio.com/PNG/a76009-keras-png.html>)

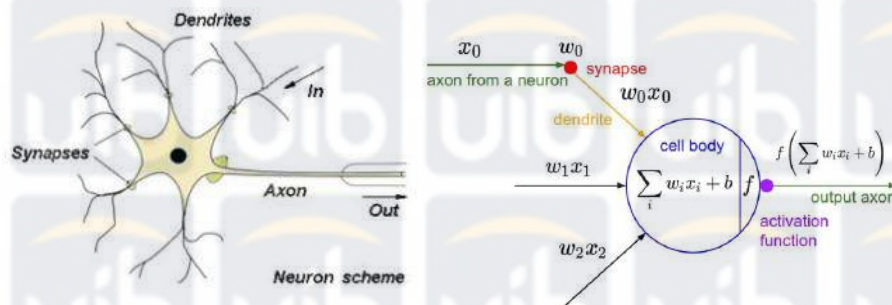
2.3. Neural Network

Neural Network (NN) merupakan komputasi intensif dan memori intensif, menjadikannya susah untuk diaplikasikan pada sistem yang sudah tertanam [12].

Neural network adalah pemodelan neuron dalam otak manusia beroperasi. *Neural network* ini juga sudah diaplikasikan dimana-mana berkisar pada *computer vision* [12]. Terdapat macam *neural network* salah satunya adalah ANN.

2.3.1 Artificial Neural Network

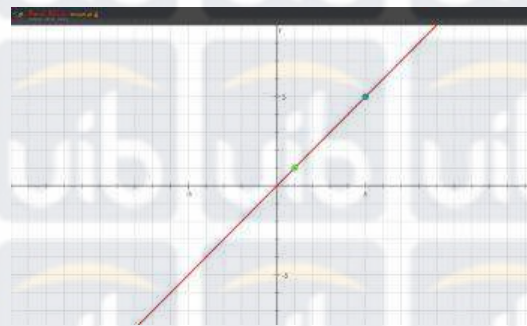
Artificial Neural Network adalah NN yang beroperasi seperti neuron pada manusia secara *artificial* atau buatan. Neuron pada manusia saling berhubungan antara satu dengan yang lainnya.



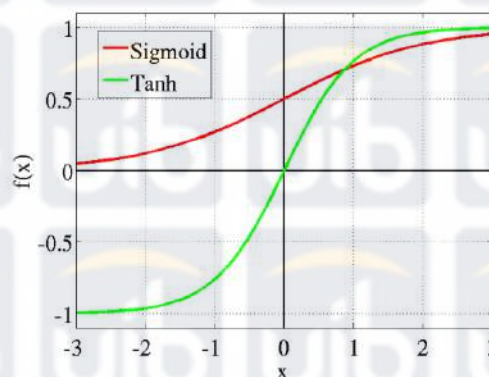
Gambar 2.4 Neuron dan Model Matematisnya

(Sumber : <https://i.stack.imgur.com/7mTvt.jpg>)

Pada gambar diatas penghubungan antara neuron adalah axon dengan dendrit neuron lainnya, inilah yang dijadikan inspirasi pembuatan *neural network*. Pada gambar diatas terdapat perhitungan antara axon dari neuron ke *output axon* yang merupakan fungsi aktivasi secara umum terbagi menjadi 2 macam yaitu linier dan non-linier. Fungsi aktivasi linier berupa garis lurus, dan yang non linier berupa sigmoid dan TanH, dan ReLU.

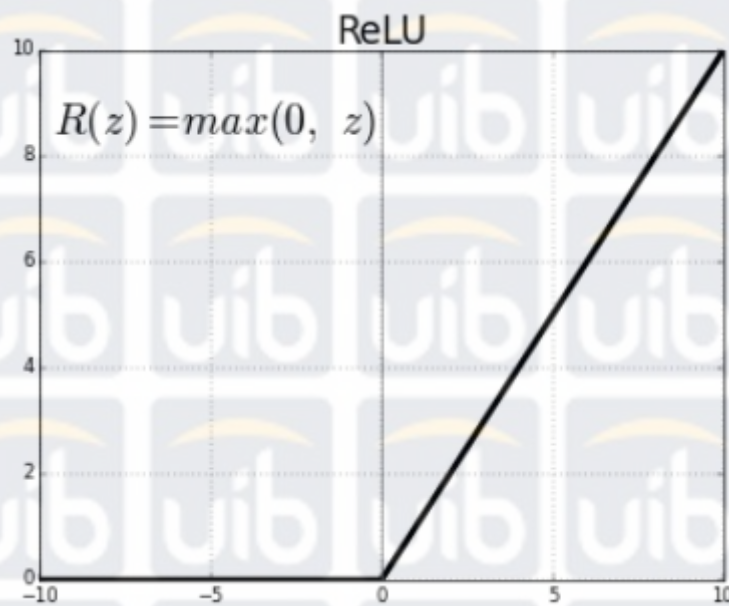


Gambar 2.5 Fungsi Linier



Gambar 2.6 Fungsi Sigmoid dan TanH

(Sumber : <https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>)

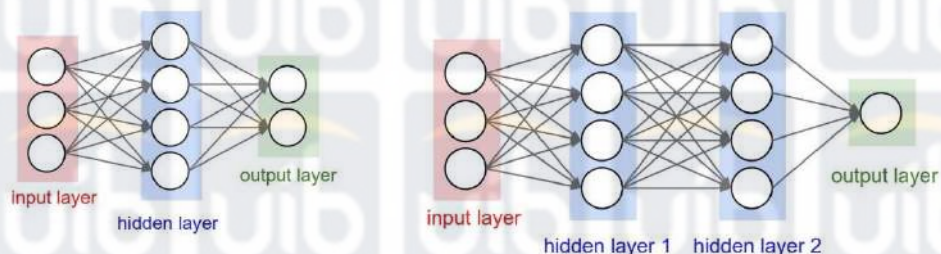


Gambar 2.7 Fungsi ReLU

(Sumber : <https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>)

Bentuk fungsi linier berupa garis lurus, fungsi sigmoid dan TanH berupa bentuk S dengan perbedaan rentang sigmoid antara 0 hingga 1 dan TanH berentang antara -1 hingga 1, dan fungsi ReLU memiliki bentuk yang memiliki *threshold* dari 0 hingga tidak terhingga. ReLU juga mampu menutupi kelemahan yang terdapat pada sigmoid dan TanH.

Arsitektur *Neural Network* dapat disebut sebagai *Multi Layer Perceptron* (MLP) atau *Fully Connected Layer*. Layer pada arsitektur biasanya terdapat *input* dan *output*, dan diantaranya terdapat layer yang tersembunyi (*hidden layer*).



Gambar 2.8 Arsitektur Neural Network

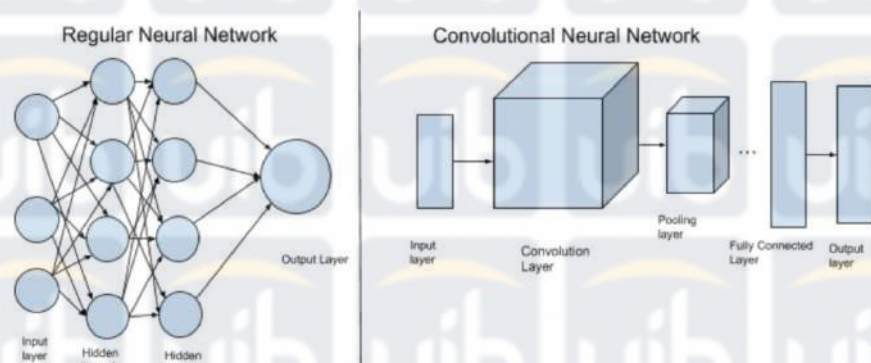
(Sumber : <https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>)

Bulat-bulat dalam layer disebut sebagai neuron, antara neuron satu dengan yang lainnya terhubung yang terdapat *weight* dan *bias* (dapat dilihat pada gambar 2.3 w dan x nya). Setiap koneksi antara neuron memiliki nilai *weight* yang berbeda-beda, sedangkan *bias* itu merupakan tambahan *input* pada *hidden layer* dan *output layer*.

Pada *input layer* tidak terdapat fungsi aktivasi, sedangkan pada *hidden layer* dan *output layer* terdapat fungsi aktivasi yang bersesuaian dengan kebutuhan oleh data yang dilatih.

2.3.1.1 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu tipe dari ANN yang dirancang khusus untuk pengenalan gambar. CNN melakukan filter tambahan pada *input* gambar untuk mencari pola tertentu[13].



Gambar 2.9 Convolutional Neural Network

(Sumber : <https://www.topcoder.com/convolutional-neural-networks-in-pytorch/>)

CNN memiliki layer tambahan yaitu *Convolutional layer*, *Pooling layer*, *Flatten*, dan *dense* atau *fully connected layer*. Layer-layer tersebut merupakan proses umum CNN.

2.4. Python

Python merupakan bahasa pemrograman yang kuat dan mudah untuk dipelajari yang dibuat oleh Guido van Rossum dan dirilis pada tahun 1991. Python memiliki struktur data efisien yang berlevel tinggi dan pendekatan terhadap

pemrograman yang berorientasi objek secara sederhana dan efisien. Python merupakan bahasa yang ideal untuk melakukan *system scripting* dan pengembangan aplikasi dengan cepat dalam berbagai macam *platform*. [14] Selain dari itu, Python juga dapat digunakan untuk *web development*, *software development*, dan matematika.

Penerjemah Python mudah untuk diperluas dengan fungsi dan tipe data baru yang terimplementasi dalam bahasa C. Python juga dapat digunakan sebagai bahasa yang berkelanjutan pada aplikasi yang dapat disesuaikan. [14]



Gambar 2.10 Python Logo

(Source : <https://upload.wikimedia.org/wikipedia/commons/c/c3/Python-logo-notext.svg>) (7 November 2019)

2.5. Anaconda

Anaconda merupakan sebuah *platform* untuk memberdayakan *asset*, kolaborasi, dan meluncurkan proyek-proyek sains. *Anaconda Navigator* merupakan sebuah *graphical user interface* (GUI) yang dapat digunakan untuk menjalankan aplikasi dan mengelola *packages* untuk menggunakan *library* dalam kode program yang dibutuhkan untuk *data learning*. Dalam *Anaconda Navigator* terdapat beberapa aplikasi salah satunya adalah *Jupyter*.



Gambar 2.11 Anaconda Logo

(Sumber : <https://www.codepolitan.com/instal-python-dengan-anaconda-58a79fee367c0>)



Gambar 2.12 Anaconda Navigator

(Sumber : Data Primer)

2.5.1. Jupyter

Jupyter merupakan perangkat lunak yang bersifat *open source* dan servis dalam komputasi yang interaktif dalam berbagai macam bahasa pemrograman. *JupyterLab* adalah *interactive development environment* yang berbasis *web* untuk *jupyter notebooks* kode program, dan data. *JupyterLab* fleksibel dalam hal mendukung *workflow* untuk data sains, komputasi ilmiah, dan *machine learning*. *JupyterLab* juga bersifat ekstensibel dan modular.

Jupyter notebook adalah sebuah aplikasi *web* yang bersifat *open source* yang diperuntukan dalam *data cleaning* dan transformasinya, simulasi angka, visualisasi data, pemodelan statistic, *machine learning* dll.



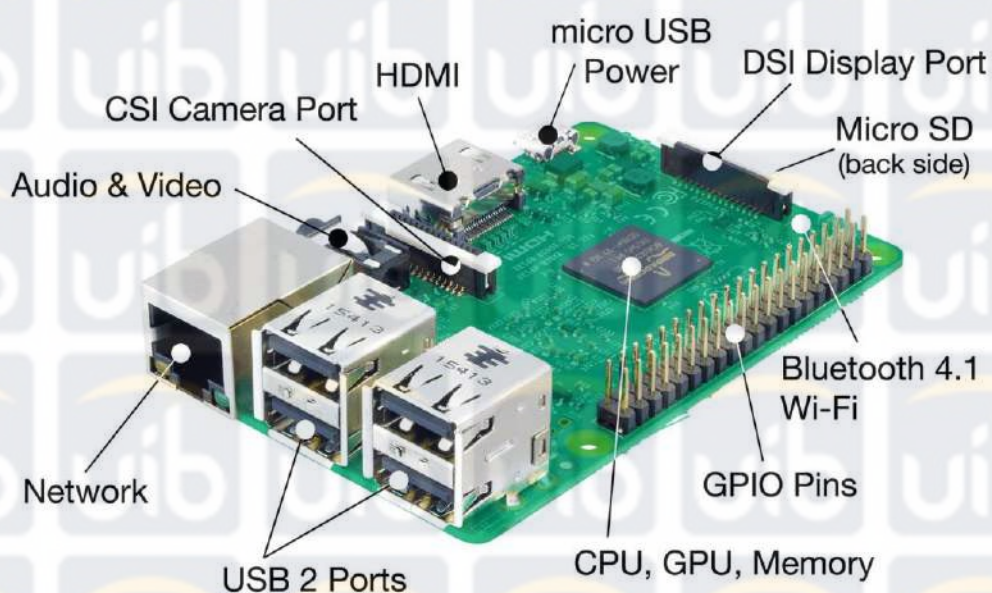
Gambar 2.13 Jupyter logo

(Sumber : <https://technopremium.com/blog/using-jupyter-notebook/>)

2.6. Raspberry Pi

Raspberry pi merupakan *singleboard computer* yang berukuran seperti kartu kredit maka dapat disebut sebagai *minicomputer*. Raspberry pi memiliki banyak model dan seluruh model tersebut memiliki sebuah *Broadcom system* pada setiap *chip*, dimana terdapat *Central Processing Unit* (CPU) yang cocok dengan ARM dan sebuah chip *Graphics Processing Unit* (GPU). Kecepatan CPU raspberry pi 3 berkisar antara 700MHz hingga 1.2MHz, dengan RAM antara 256MB hingga 1GB. Terdapat juga slot *Secure Digital* (SD) *cards* untuk penyimpanan *Operating System* (OS) dan memori program. Raspberry pi juga terdapat *port* USB, HDMI, DSI, 40 pin GPIO, *Audio Jack*, Bluetooth, Wifi dsb [15].

Raspberry pi memiliki OS tersendiri seperti Raspbian, Ubuntu, Pidora, Linutop, dan bahasa pemrograman yang didukung untuk pemrograman antara lain C++, Python, SQL. Ada 4 jenis mode power pada *minicomputer* ini: *Run*, *Standby*, *Shutdown*, dan *Dormant* [15].



Gambar 2.14 Raspberry Pi Board

(Source : <https://www.silverlineelectronics.in/raspberry-pi-3-model-b-armv8-with-1gb-ram.html>)

2.7. Pi Camera 5 MP

Pi Camera merupakan kamera yang kompatibel dengan raspberry pi dengan ukuran yang kecil 25mm x 20mm x 9mm dengan berat yang relatif ringan. *Pi camera* dapat digunakan untuk mengambil gambar dalam bentuk foto dengan piksel 2592x1944 dan juga dapat menyajikan untuk mengambil video 480 piksel pada 90 fps, 720 piksel pada 60 fps, dan yang terakhir 1080 piksel pada 30 fps [13]. Pengambil video semakin besar piksel semakin kecil tingkat *frame rate* maka pengambilan video akan semakin lambat.

Pi camera terdiri atas 2 macam, yang satunya memiliki *infrared* (IR) dan yang satunya lagi tidak memiliki *infrared* atau dapat disebut non-infrared (NoIR). Pengambilan gambar oleh kedua macam kamera ini memiliki hasil yang berbeda warna dan kamera NoIR memiliki keunggulan pada lingkungan yang rendah cahaya[13]. Penelitian ini menggunakan kamera IR, karena lingkungan yang dipakai untuk pelatihan tidaklah gelap.



Gambar 2.15 Pi Camera

(Sumber : <https://www.jakartanotebook.com/raspberry-pi-camera-board-500w-pixels#>)