

Kemudian juga terdapat implementasi CNN dalam klasifikasi pembalap MotoGP berbasis GPU yang dilakukan oleh Afandi Nur Aziz Thohari et al., [5], yang mana kelebihan dari implementasi ini yaitu mengenai arsitektur CNN yang digunakan adalah jaringan *Very Deep Convolutional Network* (VGGNet). Arsitektur ini hanya terdiri dari *3x3 layer* konvolusi, kemudian mengurangi dalam *volume size* dengan *max pooling* dan terakhir menggunakan *fully connected layer* pada akhir jaringan sebelum diklasifikasi oleh softmax. Saat ingin men-*training* data yang dikumpulkan, metode peneliti ini menjelaskan terdapat *2 training*, yaitu training dengan penggunaan CPU dan GPU dimana training yang dilakukan menggunakan GPU lebih jauh cepat 11 kali lipat dibandingkan dengan menggunakan CPU. Hal ini karena pemroses atau *core* yang dimiliki oleh GPU lebih banyak dibandingkan CPU. Penggunaan dari model CNN ini sebagai salah satu dasar untuk klasifikasi data dengan struktur dua dimensi seperti citra, sehingga akhir dari implementasinya yaitu akurasi data yang dicapai saat melakukan *testing* data yaitu sebesar 96.7% dan nilai dari sebuah akurasi dipengaruhi oleh *factor* jumlah iterasi (*epoch*) dan nilai *learning rate* yang diberikan.

Masuk dalam tahun 2019, terdapat klasifikasi jenis kendaraan dengan metode CNN yang dirancang oleh Nur Fadlia et al., [6], dimana tahap awal yang dilakukan yaitu pengumpulan data sebanyak 40 data dan kemudian masuk *preprocessing* dalam pengubahan ukuran data. Kelebihan dari penerapan CNN di dalam peneliti ini, hampir sama dengan penelitian sebelumnya yaitu untuk mengetahui tingkat akurasi dalam klasifikasi sebuah data. Sehingga dari hasil yang didapatkan dan diteliti, bahwa klasifikasi jenis kendaraan menggunakan metode CNN yaitu sebesar 73.33% yang mana diperolehnya nilai *loss* yang cukup besar yaitu 0.97 dan akurasi sebesar 0.73. Dari hasil yang diperoleh, maka diketahui mengenai kelemahan dalam peneliti ini dimana proses *training* perlu adanya data yang dikumpulkan lebih dari 40 data, sehingga saat proses *training* data dapat memperoleh akurasi yang lebih baik daripada sebelumnya.

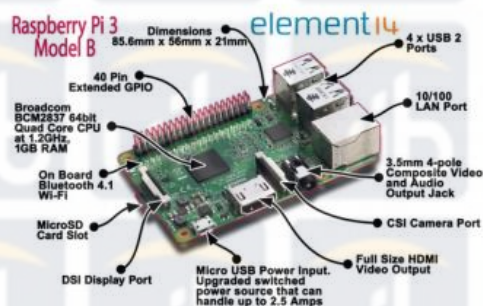
Dan pada tahun 2019, indentifikasi citra mengenai daun dilakukan kembali oleh Andi Asni B et al., [7], menggunakan *deep learning* CNN. Selain menggunakan model dari CNN, juga terdapat kelebihan dari peneliti ini yaitu mengenai penggunaan aplikasi GUI (*Graphical User Interface*) yang mana *design*

tampilan terdapat 5 *push button*, diantaranya *button load net*, *grayscale*, *reset*, indentifikasi dan *load citra*. Ke-5 *push button* mempunyai fungsi masing-masing, salah satunya indentifikasi berfungsi untuk menentukan hasil pada citra. Dalam pengujian yang dilakukan oleh peneliti terdapat 4 kelas yaitu kelas daun pepaya, singkong, yodium dan ganja. Dalam program GUI, juga terdapat kelemahan mengenai sistem yang akan tidak bekerja saat daun yang dimasukkan tidak terdaftar dalam pengujian kelas. Adapun hasil akhir dari penerapan CNN ini yaitu, aplikasi CNN hanya dapat mendeteksi label daun sesuai citra daun yang telah dikelompokkan pada kelas-kelas yang telah dilatih dan hasil akurasi dalam pengenalan mencapai 100%.

Dari beberapa penelitian diatas mengenai metode CNN yang telah diteliti dan dipahami oleh peneliti, maka selanjutnya peneliti akan merancang sebuah model CNN dengan nilai akurasi dari *training* data mencapai titik dekat dengan nilai akurasi sehingga *file* dari model akan diimportkan ke dalam *raspberry pi* untuk diolah lebih lanjut dalam klasifikasi data secara real time melalui *camera pi*.

2.2. Raspberry Pi B

Raspberry Pi merupakan salah satu komputer *mini* yang berukuran seperti kartu *ATM*. *Raspberry Pi* juga menggunakan SoC (*System on a chip*) *ARM* yang telah dikemas dan diintergrasikan diatas *PCB*.



Gambar 2.1 *Raspberry Pi* Model B

Sumber: element14.com/RaspberryPi (12 Desember 2019)

Pada gambar diatas merupakan salah satu generasi *Raspberry Pi 3 model B*, dimana terdapat RAM 1 GB dan memiliki 4 *USB port*, *pin GPIO*, *Full HDMI port*,

Port Ethernet, Combined 3.5 mm audio jack dan composite video, camera interface (CSI), display interface (DSI), slot kartu Micro SD, dan VideoCore IV 3D graphics core.

Kemudian dalam *Raspberry Pi 3*, terdapat GPIO yang dimana merupakan *pin* yang terdiri dari 40 dan masing-masing memiliki fungsi tersendiri. Dalam gambar 2.2 dapat diketahui penjelasan mengenai *pin* dari *Raspberry Pi 3* tersendiri.

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1, I2C)	DC Power 5v	04
05	GPIO03 (SCL1, I2C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO18	24
25	Ground	(SPI_CE1_N) GPIO17	26
27	ID_SD (I2C ID EEPROM)	(I2C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Gambar 2.2 GPIO *Raspberry Pi 3*

Sumber: element14.com/RaspberryPi (12 Desember 2019)

Dari penjelasan *pin* GPIO diatas, menurut Malik Abdillah Ibnul Hakim et al., [8], "*pin* yang berlabel SCL dan SDA dapat digunakan sebagai I²C". Kemudian untuk *pin* yang berlabel MOSI, MISO dan SCKL dapat berfungsi sebagai penghubung SPI kecepatan tinggi.

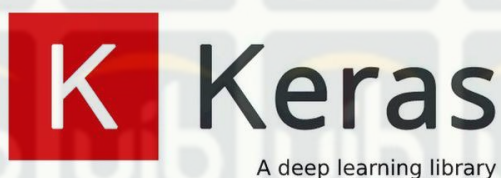
2.3. TensorFlow dan Keras



Gambar 2.3 TensorFlow

Sumber: <https://www.tensorflow.org/> (12 Desember 2019)

TensorFlow merupakan suatu multidimensi *array* yang dikenal sebagai tensor dan juga digunakan sebagai jaringan saraf dalam operasi yang berbeda. Selain dikenal sebagai suatu multidimensi *array*, *tensorflow* juga merupakan suatu *tool* perpustakaan perangkat lunak yang memiliki tugas pada pembelajaran dalam *machine learning* oleh jaringan saraf maupun pemrograman diferensial.



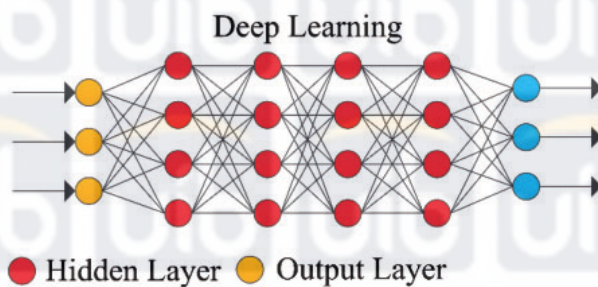
Gambar 2.4 Keras

Sumber: <https://keras.io> (12 Desember 2019)

Sedangkan *keras* merupakan suatu jaringan saraf API (*application program interface*) tingkat tinggi yang ditulis dengan bahasa *Python* dan dapat berjalan diatas *TensorFlow*. *Keras* juga mengkompilasi *model* dengan fungsi *loss* dan *optimizer* dalam proses pelatihan. Dalam *keras* terdapat salah satu istilah yang disebut sebagai “*Backend*”, dimana istilah tersebut dapat diketahui saat menjalankan program dari *keras* itu sendiri. Istilah tersebut merupakan perhitungan tingkat rendah dalam *tensor* dan konvolusi dengan bantuan *library* dari *tensorflow*.

2.4. Deep Learning

Deep learning merupakan salah satu model pembelajaran yang dirancang untuk menganalisa data dengan struktur logika. Dengan menggunakan *tool* dari *deep learning* terdapat struktur algoritma yang berlapis-lapis dimana disebut juga dengan jaringan saraf tiruan.



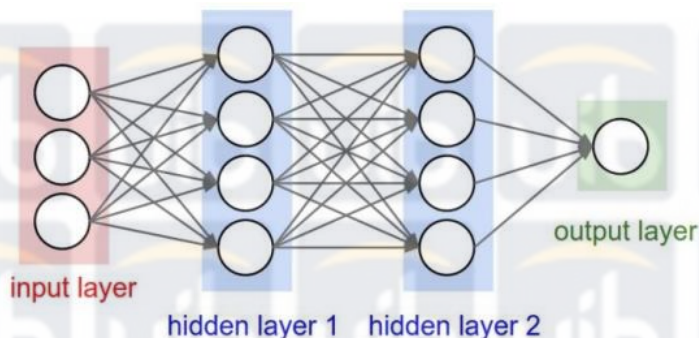
Gambar 2.5 *Deep Learning*

Sumber: researchgate.net/Deep-learning (12 Desember 2019)

Menurut Aditya Santoso et al., [9], *deep learning* merupakan salah satu “cabang dari *machine learning* yang terinspirasi dari korteks manusia dengan menerapkan jaringan syaraf bantuan yang memiliki banyak *hidden layer*”. *Deep learning* juga merupakan salah satu jaringan saraf tiruan yang sangat tinggi dimana dapat ditulis dengan bahasa *python* dan mampu mengoperasikan dalam *TensorFlow*.

2.5. Convolutional Neural Network (CNN)

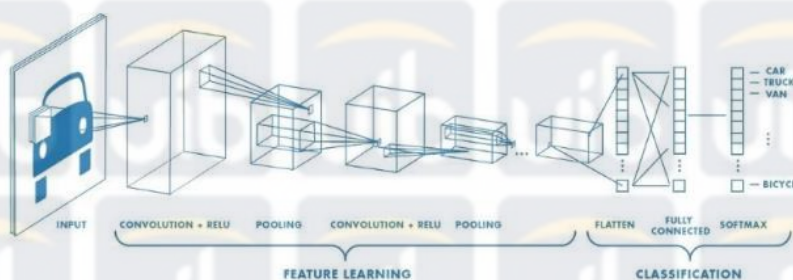
CNN merupakan suatu pengembangan dari *Multilayer Perceptron* (MP) yang bertujuan untuk mengolah suatu data yang berdimensi dua. CNN merupakan salah satu jenis dari *Deep Neural Network* karena merupakan salah satu jaringan saraf yang tinggi dan sering dioperasikan dalam data citra.



Gambar 2.6 CNN (*Convolutional Neural Network*)

Sumber: [researchgate.net/Convolutional-neural-network](https://www.researchgate.net/publication/332111111) (12 Desember 2019)

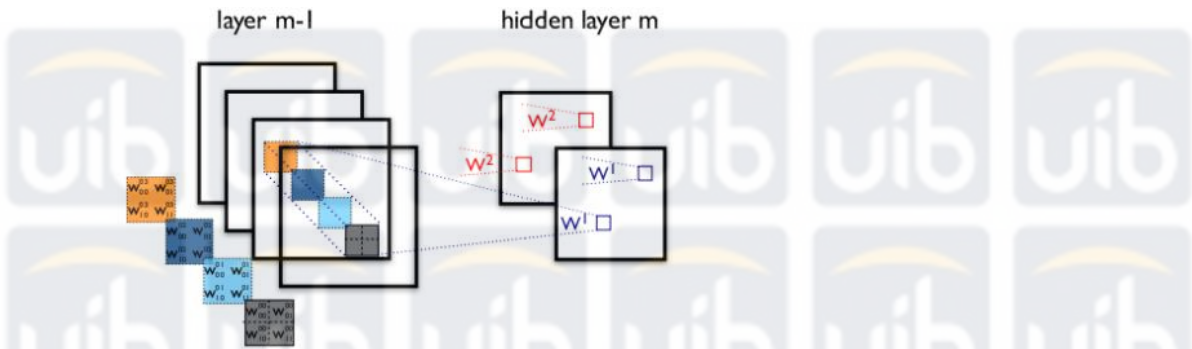
Dalam CNN terdapat beberapa *layer* yang mana tiap *layer* berfungsi sebagai *filter* dalam pemrosesan data. Pemrosesan yang dilakukan disebut juga pemrosesan *training* dimana terdapat 3 tahap yaitu *convolutional layer*, *pooling layer* dan *fully connected layer*.



Gambar 2.7 Proses CNN (*Convolutional Neural Network*)

Sumber: [mathworks.com/Convolutional-neural-network.html](https://www.mathworks.com/help/deeplearning/ug/convolutional-neural-network.html) (12 Desember 2019)

Menurut Eka Putra et al., [10], CNN “dalam operasi linear menggunakan operasi konvolusi, sedangkan bobot lainnya berbentuk empat dimensi yang merupakan kernel konvolusi”.



Gambar 2.8 Proses Konvolusi CNN (*Convolutional Neural Network*)

Sumber: mathworks.com/Convolutional-neural-network.html (12 Desember 2019)