

**RANCANG BANGUN SISTEM KLASIFIKASI MINERAL DAN BATUAN
MENGUNAKAN TENSORFLOW.JS**



TUGAS AKHIR

*Disusun dalam rangka memenuhi salah satu persyaratan
Untuk menyelesaikan program Strata-1 Departemen Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Makassar*

Disusun Oleh :

**YAKIP
D42114308**

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2020

HALAMAN PENGESAHAN

**RANCANG BANGUN SISTEM KLASIFIKASI MINERAL DAN BATUAN
MENGUNAKAN TENSORFLOW.JS**

Oleh :

YAKIP

D421 14 308

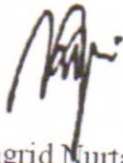
Skripsi ini telah di pertahankan pada Ujian Akhir Sarjana tanggal 30 November 2020. Diterima dan disahkan sebagai salah satu syarat memperoleh gelar Sarjana Teknik pada Program Studi S1 Teknik Informatika Departemen Teknik Informatika Fakultas Teknik Unviversitas Hasanuddin.

Makassar, 30 November 2020

Disetujui Oleh :

Pembimbing I,

Pembimbing II,



Dr. Ir. Ingrid Nurtanio M.T.
Nip. 19610813 198811 2 001



Dr. Eng. Zulkifli Tahir, S.T., M.Sc.,
Nip. 19840403 201012 1 004

Diterima dan disahkan oleh:

Ketua Departemen S1 Teknik Informatika



Dr. Amri Ahmad Ilham, S.T., M.IT
Nip. 19721010 199802 1 001

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : YAKIP

NIM : D421 14 308

Program Studi : S1 Teknik Informatika

Menyatakan dengan sebenar-benarnya bahwa skripsi yang berjudul :

RANCANG BANGUN SISTEM KLASIFIKASI MINERAL DAN BATUAN MENGUNAKAN TENSORFLOW.JS

Adalah karya ilmiah saya sendiri dan sepanjang pengetahuan saya di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah di ajukan/ditulis/diterbitkan sebelumnya. Kecuali yang secara tertulis dikutip dalam naskah ini dan di sebutkan dalam sumber kutipan dan daftar pustaka.

Apabila dikemudian hari ternyata di dalam naskah skripsi ini terdapat unsur-unsur jiplakan, saya bersedia menerima sanksi atas perbuatan tersebut dan dip roses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2000, pasal 25 ayat 2 dan pasal 70)

Makassar, 01 Desember 2020

Yang Membuat Pernyataan



YAKIP

ABSTRAK

Batuan adalah salah satu penyusun elemen kulit bumi yang menyediakan mineral-mineral anorganik melalui pelapukan yang selanjutnya menghasilkan tanah. Batuan mempunyai komposisi mineral, sifat-sifat fisik, dan umur yang beraneka ragam. Terdapat 2963 nama batuan dan 5639 spesies mineral. Dengan banyaknya batuan dan mineral tentu saja akan menyulitkan kita dalam mengidentifikasi semua batuan dan mineral yang tidak sedikit. TensorFlow.js merupakan *library machine learning* yang dikembangkan untuk mengidentifikasi citra, suara, pola dan masih banyak lagi. Dalam kasus ini tensorflow.js digunakan untuk mengenali dan mengklasifikasi citra batuan dan mineral. Sistem dibangun pada web browser di mana untuk pengenalan gambarnya menggunakan model yang datanya di *training* menggunakan *convolutional neural network*. Adapun hasil *training* dari 2125 citra batuan dan mineral didapatkan tingkat akurasi sebesar 98.96 % dan *validation* dari 375 citra batuan dan mineral menghasilkan akurasi yaitu 98.13 %. Kemudian penelitian ini menggunakan data *testing* untuk menguji model yang telah dibuat. Tingkat akurasi yang dihasilkan menggunakan data *testing* sebesar 87.2 % dari total 125 citra batuan dan mineral . Sehingga, performa dari model yang dibuat pada penelitian ini dapat dikatakan optimal dalam mengklasifikasikan citra batuan dan mineral.

Kata Kunci: *tensorflow.js, Klasifikasi citra, Batuan dan mineral, Convolution neural network;*

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Segala puji dan syukur kami panjatkan ke hadirat Allah S.W.T Tuhan Yang Maha Esa yang dengan limpahan rahmat dan hidayah-Nya sehingga tugas akhir dengan judul “Rancang Bangun system Klasifikasi Batuan dan Mineral Menggunakan TensorFlow.js” ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Dalam penyusunan penelitian ini disajikan hasil penelitian terkait judul yang telah diangkat dan telah melalui proses pencarian dari berbagai sumber baik jurnal penelitian, prosiding pada seminar-seminar nasional/internasional, buku maupun dari situs-situs di internet.

Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, mulai dari masa perkuliahan sampai dengan masa penyusunan tugas akhir, sangatlah sulit untuk menyelesaikan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih sedalam-dalamnya kepada:

- 1) Tuhan Yang Maha Esa atas semua berkat, karunia serta pertolongan-Nya yang tiada batas, yang telah diberikan kepada penulis disetiap langkah dalam pembuatan program hingga penulisan laporan skripsi ini.
- 2) Kedua orang tua penulis yang telah membimbing dan menjadi tempat berkeluh kesah serta keluarga yang senantiasa memberikan kekuatan,

motivasi, bimbingan moral, materi, kepercayaan dan kasih sayang yang tidak terbatas kepada penulis.

- 3) Ibu Dr. Ir. Ingrid Nurtanio M.T. selaku pembimbing I dan Pembimbing Akademik yang telah banyak memberikan bimbingan, motivasi dan masukan yang bermanfaat kepada penulis selama masa perkuliahan.
- 4) Bapak Dr. Eng. Zulkifli Tahir, S.T., M.Sc., selaku pembimbing II yang telah banyak memberikan bimbingan, motivasi dan masukan yang bermanfaat kepada penulis.
- 5) Ibu Elly Warni, S.T., M.T. dan Ibu Anugrahyani Bustamin, S.T., M.T. selaku dosen penguji yang telah memberikan saran sehingga laporan skripsi ini menjadi lebih baik
- 6) Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bantuan dan bimbingannya selama masa perkuliahan penulis.
- 7) Bapak Robert, Bapak Zainuddin, dan Ibu Santi serta segenap staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu kelancaran penyelesaian tugas akhir penulis.
- 8) Para sahabat dekat adiyaksa Squad Alamsyah Ibrahim, Hermawan Safrin, Syarif Hidayatullah, Abdillah satari Rahim, Rahmat Firman, Yusuf Ramadhan, Siswono Nasir, Ahmad Setiadi, Muhammad Ibnu, Arya Jaka Putra, dan Muh ardiansyah yang telah memberikan doa, bantuan dan dukungan serta bimbingan sejak masa perkuliahan sampai dengan penyelesaian tugas akhir ini.

- 9) Teman – teman Rectifier14 Al Riefqy Dasmito, Ulfah Rojiyyah, Rajab Muzakkar, Muh. Hanif Muzakkir, Fadel Pratama, Inka G Malissa, Cindy O lolo Bulan, Anastasya Yuki yang telah memberikan doa, nasihat, bantuan dan semangat selama proses penyelesaian tugas akhir ini.
- 10) Serta seluruh pihak yang tidak sempat disebutkan satu persatu yang telah banyak meluangkan tenaga, waktu dan pikiran selama penyusunan laporan tugas akhir ini.

Akhir kata, penulis berharap semoga Tuhan Yang Maha Esa berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Semoga tugas akhir ini dapat memberikan manfaat bagi pengembangan ilmu selanjutnya.

Amin.

Wassalamualaikum

Gowa, 21 April 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN SKRIPSI	Error! Bookmark not defined.
ABSTRAK	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian	6
1.4 Manfaat Penelitian	6
1.5 Batasan Masalah Penelitian.....	6
1.6 Sistematika Penulisan	7
BAB II	9
TINJAUAN PUSTAKA	9
2.1 Batuan dan Mineral	9
2.1.1 Identifikasi Batuan.....	11

2.1.2	Identifikasi Mineral	18
2.2	Artificial Intelligence	23
2.3	Machine Learning	25
2.2.1	Supervised Learning	26
2.2.2	<i>Unsupervised Learning</i>	29
2.4.3	<i>Reinforcement Learning</i>	31
2.4	Deep Learning	32
2.5	Artificial Neural Network	33
2.6	Convolutional Neural Network	37
2.6.1	<i>Convolution Layer</i>	38
2.6.2	<i>Pooling</i>	39
2.6.3	<i>Fully Connected Layer</i>	40
2.7	Website	41
2.8	Node.js	43
2.9	TensorFlow	44
BAB III		57
METODOLOGI PENELITIAN		57
3.1	Tahapan Penelitian	57
3.2	Lokasi Dan Waktu Penelitian	58
3.3	Instrumen Penelitian	58

3.4	Teknik Pengambilan Data	60
3.5	Perancangan dan Implentasi Sistem.....	61
3.5.1	Model/arsitektur	63
3.5.2	Pembuatan Model	63
3.5.3	Rancangan Tampilan Sistem	73
BAB IV		74
HASIL DAN PEMBAHASAN		74
4.1	Hasil Penelitian.....	74
4.1.1	Arsitektur Jaringan	74
4.1.2	Proses Konvolusi	78
4.1.3	Proses Pooling.....	81
4.1.4	Proses Fully Connected Layer	82
4.1.5	Hasil training model	83
4.1.6	Testing	83
4.2	Parameter Model	86
4.2.1	Parameter Jumlah Epochs	87
4.2.2	Parameter Jumlah Layer Konvolusi	88
4.2.3	Parameter Pooling Layer	88
4.2.4	Parameter Input Image	89
4.2.5	Parameter kernel size	90
BAB V		91
PENUTUP.....		91

5.1	KESIMPULAN	91
5.2	SARAN	92
	DAFTAR PUSTAKA	93
	LAMPIRAN	95

DAFTAR GAMBAR

Gambar 2. 1 The Rock Cycle.....	11
Gambar 2. 2 Ilustrasi Turing test	24
Gambar 2. 3 Subbagian machine learning	26
Gambar 2. 4 Classification.....	29
Gambar 2. 5 Clustering	31
Gambar 2. 6 Reinforcement learning	31
Gambar 2. 7 Arsitektur neuron neural network	33
Gambar 2. 8 Single layer artificial neural network.....	35
Gambar 2. 9 Multi layer artificial neural network	36
Gambar 2. 10 Recurrent neural network.....	37
Gambar 2. 11 Arsitektur convolutional neural network	38
Gambar 2. 12 Convolution layer.....	39
Gambar 2. 13 Max Pooling.....	40
Gambar 2. 14 Fully Connected Layer.....	41
Gambar 2. 15 Tensorflow Toolkit Hierarchy.....	45
Gambar 2. 16 Tensor “flow” melalui sejumlah lapisan, skenario umum di TensorFlow dan TensorFlow.js	53
Gambar 2. 17 Arsitektur TensorFlow.js	54
Gambar 3. 1 Diagram Tahapan Penelitian.....	57
Gambar 3. 2 Gambar batuan dan mineral	60
Gambar 3. 3 Gambaran Umum Proses Kerja Sistem.....	61
Gambar 3. 4 Flowchart gambaran sistem.....	62

Gambar 3. 5 Arsitektur CNN	63
Gambar 3. 6 Contoh properties gambar dengan bit depth 24	64
Gambar 3. 7 Contoh properties gambar dengan bit depth 32	64
Gambar 3. 8 Jumlah total gambar yang terdeteksi.....	67
Gambar 3. 9 model summary.....	69
Gambar 3. 10 Mengaktivasi folder tfjsvis.....	70
Gambar 3. 11 Menginstall Tensorflow.js.....	71
Gambar 3. 12 Model berhasil terdeteksi	71
Gambar 3. 13 Pilih No Compression (Higher Accuracy)	72
Gambar 3. 14 Output folder untuk menyimpan model tensorflowjs	72
Gambar 3. 15 Hasil Konversi model tensorflowjs.....	72
Gambar 3. 16 Rancangan Tampilan Sistem.....	73
Gambar 4. 1 Arsitektur Jaringan	74
Gambar 4. 2 Proses Konvolusi.....	78
Gambar 4. 3 Proses Perhitungan layer konvolusi	79
Gambar 4. 4 Pergeseran kernel pada konvolusi.....	80
Gambar 4. 5 Proses Maxpooling.....	82
Gambar 4. 6 Grafik hasil training dan validation.....	83
Gambar 4. 7 Hasil prediksi citra galena	84
Gambar 4. 8 Hasil prediksi citra azurite	84
Gambar 4. 9 Hasil Prediksi Citra Malachite	85
Gambar 4. 10 Hasil Prediksi Citra Jasper	85
Gambar 4. 11 Hasil Prediksi Citra Jasper	85

DAFTAR TABEL

Tabel 2. 1 Hierarki Toolkit Tensorflow	46
Tabel 3. 1 Pembagian masing-masing gambar pada folder training, dan validation	60
Tabel 4. 1 Model CNN.....	77
Tabel 4. 2 Confusion Matriks	86
Tabel 4. 3 Hasil Training menggunakan parameter jumlah epochs.....	87
Tabel 4. 4 Hasil akurasi dengan parameter jumlah konvolusi layer	88
Tabel 4. 5 Hasil akurasi Max Pooling dan Average Pooling	89
Tabel 4. 6 Hasil Training Input image	89
Tabel 4. 7 hasil akurasi menggunakan kernel size.....	90

BAB I

PENDAHULUAN

1.1 Latar Belakang

Secara geografis Indonesia berada di antara dua benua, yakni Benua Asia dan Benua Australia. Kemudian di samudera terletak di antara Samudera Hindia dan Samudera Pasifik. Sementara itu, letak geologis Indonesia berada di antara dua daratan pegunungan muda, seperti Sirkum Mediterania dan Pasifik. Pada bagian lempeng diapit oleh tiga Lempeng yakni Asia-Australia, Euresia, dan Pasifik. Berkat letak geologis Indonesia yang dilalui dua jalur pegunungan di dunia di deretan, tanah di Indonesia menjadi sangat subur. Letusan gunung api mempengaruhi tingkat kesuburan tanah. Kesuburan tanah memudahkan penduduk Indonesia untuk bercocok tanam. Selain itu, Indonesia juga dikenal akan kekayaan sumberdaya mineralnya sehingga banyak ditemukan galian tambang mineral dan batuan yang tersebar diseluruh Indonesia.

Batuan adalah salah satu penyusun elemen kulit bumi yang menyediakan mineral-mineral anorganik melalui pelapukan yang selanjutnya menghasilkan tanah. Batuan mempunyai komposisi mineral, sifat-sifat fisik, dan umur yang beraneka ragam. Jarang sekali batuan yang terdiri dari satu mineral, namun umumnya merupakan gabungan dari dua mineral atau lebih. Mineral dapat kita definisikan sebagai bahan padat anorganik yang terdapat secara alamiah, yang terdiri dari unsur-unsur kimiawi dalam perbandingan tertentu, dimana atom-atom didalamnya tersusun mengikuti suatu pola yang sistimatis. Mineral dapat kita jumpai dimana - mana disekitar kita, dapat berwujud sebagai batuan, tanah, atau pasir yang diendapkan pada dasar sungai. Beberapa daripada mineral tersebut

dapat mempunyai nilai ekonomis karena didapatkan dalam jumlah yang besar, sehingga memungkinkan untuk ditambang seperti emas dan perak. Mineral, kecuali beberapa jenis, memiliki sifat, bentuk tertentu dalam keadaan padatnya, sebagai perwujudan dari susunan yang teratur didalamnya. Pengetahuan tentang “mineral” merupakan syarat mutlak untuk dapat mempelajari bagian yang padat dari Bumi ini, yang terdiri dari batuan. Bagian luar yang padat dari Bumi ini disebut litosfir, yang berarti selaput yang terdiri dari batuan, dengan mengambil “lithos” dari bahasa latin yang berarti batu, dan “sphere” yang berarti selaput. Tidak kurang dari 2000 jenis mineral yang kita ketahui sekarang. Terdapat dua cara untuk dapat mengenal suatu mineral, yang pertama adalah dengan cara mengenal sifat fisiknya. Yang termasuk dalam sifat fisik mineral adalah bentuk kristalnya, berat jenis, bidang belah, warna, kekerasan, goresan, dan kilap. Adapun cara yang kedua adalah melalui analisa kimiawi atau analisa difraksi sinar X, cara ini pada umumnya sangat mahal dan memakan waktu yang lama.

Pengamatan mineral dan batuan biasanya dilakukan pada singkapan - singkapan alami, seperti tempat-tempat yang tererosi, bagian terjal suatu pegunungan (gawir), pinggiran sungai dan pantai yang terjal. Selain itu pengamatan mineral dan batuan dapat dilakukan di tempat-tempat yang tersingkap karena kegiatan manusia, seperti tempat galian bahan tambang atau potongan di pinggir jalan. Pengamatan yang dilakukan mencakup jenis mineral dan batuan, struktur dan stratigrafi batuan, dan ketebalan masing-masing lapisan batuan. Perlapisan batuan dalam kerak bumi tidak selamanya horisontal, sehingga sudut dan arah kemiringan lapisan (dip dan strike) juga perlu diamati. Sudut ini diukur

dengan klinometer, sedangkan arah kemiringan diukur dengan menggunakan kompas. Kejadian-kejadian pada struktur lapisan batuan perlu diamati, antara lain lipatan (*fold*), patahan/sesar (*fault*) dan retakan (*joint*). Dengan banyaknya mineral dan batuan yang ada memungkinkan kita untuk sulit dalam mengidentifikasi setiap mineral dan batuan yang kita temukan. Berdasarkan data *mindat.org* terdapat 5640 mineral dan 1963 batuan. Untuk mengidentifikasi sebuah mineral dan batuan butuh langkah dan proses yang tidak mudah, mulai dari mengidentifikasi bentuk, pola, warna serta kandungan dalam mineral dan batuan tersebut. Pada penelitian ini diharapkan agar peneliti dapat mengidentifikasi mineral dan batuan berdasarkan fitur warna yang terdapat pada mineral dan batuan.

Seiring dengan kemajuan zaman, klasifikasi citra digital sangat dibutuhkan diberbagai macam bidang, seperti : informatika, kedokteran, kelautan, pertanian, dan bisnis. Beberapa penelitian yang telah dilakukan misalnya klasifikasi buku (Lukman, 2012) dan klasifikasi pada daging sapi (Budianita & Jasril, 2015). Pada penelitian ini peneliti akan berfokus pada klasifikasi mineral dan batuan yang bertujuan untuk memudahkan kita dalam mengidentifikasi mineral dan batuan. Tujuan dari klasifikasi citra adalah mengklasifikasikan masukkan citra kedalam beberapa kategori tertentu. Klasifikasi citra saat ini menjadi salah satu problem yang telah lama dicari solusinya dalam computer vision. Bagaimana menduplikasikan kemampuan manusia dalam memahami informasi citra digital, supaya komputer dapat mengenali objek pada citra selayaknya manusia. Proses *feature engineering* yang digunakan pada umumnya sangat terbatas dimana hanya

dapat berlaku pada dataset tertentu saja tanpa kemampuan generalisasi apapun. Hal ini dikarenakan berbagai perbedaan citra antara lain perbedaan sudut pandang, perbedaan skala, perbedaan kondisi pencahayaan, deformasi objek, dan sebagainya. Kalangan akademisi telah banyak bergelut dalam problem ini. Salah satu pendekatan yang berhasil digunakan dengan menggunakan Jaringan Syaraf Tiruan *Artificial Neural Network* (ANN). ANN adalah salah satu bentuk kecerdasan buatan yang mempunyai kemampuan untuk belajar dari data dan tidak membutuhkan waktu lama dalam pembuatan model (Setiawan dan Rudiyanto, 4 2004). Keuntungan dari penggunaan ANN adalah kemampuannya untuk mempelajari hubungan yang tidak diketahui yang sudah ada sebelumnya antara data input dan output dari setiap sistim. Selain itu pemodelan dengan ANN memiliki atribut yang diinginkan dan kemampuan belajar dari contoh-contoh tanpa memerlukan data fisik secara eksplisit. ANN merupakan bagian dari *Machine Learning* (ML). *Machine Learning* adalah ilmu (dan seni) pemrograman komputer sehingga mereka bisa belajar dari data (aurelien, 2017). *Machine Learning* merupakan cabang dari *Artificial Intelligence* dengan kemampuan mesin untuk mengakses data yang ada dengan perintah mereka sendiri. *Machine Learning* juga mampu mempelajari data yang ada dan melakukan tugas-tugas tertentu. *Machine Learning* mampu melakukan ini dengan metode mempelajari algoritma dan model statistik yang ada. Python dan C++ merupakan bahasa pemrograman yang populer untuk pengembangan *librabry machine learning*. Namun pada kenyataan pengembangan bahasa pemrograman javascript saat ini sangatlah populer. Hal ini dibuktikan dengan *pull request* pada github hingga

kuarter pertama tahun 2020 masih di dominasi bahasa pemrograman javascript. Dan berdasarkan survey yang dibuat oleh perusahaan Stack Overflow pada tahun 2019 dengan 87.354 responden yang tersebar diseluruh dunia mengakumulasi bahwa *Java Script* merupakan bahasa pemrograman terpopuler dengan perolehan 67.8%. TensorFlow.js Merupakan *librabry* yang dikembangkan dengan menggunakan bahasa pemrograman *javascript* dan dipublikasikan oleh google pada tahun 2018. Pada pengaplikasiannya TensorFlow.js digunakan untuk training dan penggunaan *machine learning* (ML) model di browser. TensorFlow.js dilakukan pada client browser, mulai dari aktivitas *building, train, task* hingga *run* dilakukan dalam satu instalasi saja. TensorFlow.js ini digunakan pada WebGL sehingga pada browser apapun, pada *device* apapun sangat memungkinkan bisa digunakan. sehingga proses ini nantinya akan mempermudah *client* untuk proses *real time* pada sensor-sensor yang dimuat oleh *device* tertentu untuk digunakan proses *data mining* maupun proses *training* tersebut. Kita tidak perlu memikirkan *cloud service* apa yang dimuat dengan model ini, karena sekali API dijalankan pada browser semua di proses sesuai dengan data yang kita masukkan.

Dari permaparan latar belakang diatas maka penulis mengangkat judul penelitian “**Rancang Bangun System Klasifikasi Mineral dan Batuan Menggunakan TensorFlow.js**”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian tugas akhir ini adalah:

1. Bagaimana membuat model mineral dan batuan pada tensorflow.js ?

2. Bagaimana mengintegrasikan TensorFlow.js pada aplikasi yang dibangun?
3. Bagaimana kinerja performa aplikasi yang dibuat untuk mengidentifikasi mineral dan batuan berdasarkan dataset yang ada?

1.3 Tujuan Penelitian

Merujuk pada rumusan masalah yang ada, maka tujuan yang ingin dicapai dalam penelitian ini adalah:

2. Membuat dataset mineral dan batuan dan dikonversi kedalam model TensorFlow.js
3. Mengintegrasikan aplikasi yang di bangun dengan dataset yang sudah dikonversi ke model TensorFlow.js sehingga dapat digunakan untuk mengklasifikasi mineral dan batuan.
4. Mengetahui akurasi dalam mengklasifikasi mineral dan batuan menggunakan gambar atau photo

1.4 Manfaat Penelitian

Ada pun manfaat dalam penilitan ini adalah:

1. Menerapkan ilmu-ilmu yang didapatkan penulis semasa perkuliahan
2. Memudahkan kita dalam mengklasifikasi gambar mineral dan bantuan.
3. Memberikan sudut pandang baru tentang klasifikasi gambar menggunakan TensorFlow.js

1.5 Batasan Masalah Penelitian

Mengingat banyaknya perkembangan yang bisa ditemukan dalam permasalahan ini, maka perlu adanya batasan-batasan masalah yang jelas

mengenai apa yang dibuat dan diselesaikan dalam program ini. Adapun batasan-batasan masalah pada penelitian ini sebagai berikut:

1. Pembuatan Website menggunakan *javascript* sebagai client-side dan *node.js* server-side.
2. Dataset mineral dan batuan yang digunakan yang terdiri dari azurite, gold, malachite, jasper dan galena .
3. Klasifikasi gambar hanya bisa dilakukan pada browser laptop/pc.
4. Menggunakan *library machine learning TensorFlow.js*.
5. Pengujian menggunakan data yang telah ditentukan oleh penulis dan tidak termasuk dalam dataset.

1.6 Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

BAB I PENDAHULUAN

Pada bab ini membahas latar belakang penelitian yang menjelaskan bagaimana latar belakang pembuatan aplikasi ini akan dibuat, ruang lingkup yang menjelaskan batasan-batasan penelitian dalam pembuatan Sistem lasifikasi gambar yang menggunakan TensorFlow.js sebagai komponen perancangan sistem, serta tujuan dan manfaat dari pembuatan aplikasi ini bagi para pengguna, metodologi penelitian, tinjauan pustaka dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan teori-teori yang menunjang percobaan penelitian seperti *Machine learning*, Javacript, TensorFlow.js dan sebagainya.

BAB III METODOLOGI PENELITIAN

Pada bab ini berisi tentang perancangan sistem, pembuatan sistem, skenario pengujian yang akan dilakukan dan skenario pengujian untuk melihat performa dari sistem yang dibuat.

BAB IV HASIL DAN PEMBAHASAN

Dalam bab ini menjelaskan tentang penyajian data hasil performa sistem, proses-proses yang dilakukan terhadap data-data yang telah diperoleh, dan pembahasan atau evaluasi hasil implementasi sistem secara keseluruhan.

BAB V PENUTUP

Bab ini berisi kesimpulan dan penelitian yang telah dibuat, serta saran-saran untuk dapat meningkatkan dan mengembangkan sistem pada masa yang akan datang.

BAB II

TINJAUAN PUSTAKA

2.1 Batuan dan Mineral

Mineral adalah zat anorganik alami, tersusun dari atom-atom dari salah satu unsur kimia tunggal atau sejumlah unsur yang berbeda. Ada lebih dari 4.000 mineral yang berbeda, dan masing-masing dibedakan oleh komposisi kimianya (rasio unsur-unsur kimianya) dan struktur kristalnya. Hampir semua mineral berbentuk kristal: atom-atom tersusun dalam pola teratur; ketika dibiarkan tumbuh bebas, mereka membentuk kristal simetris dengan permukaan datar.

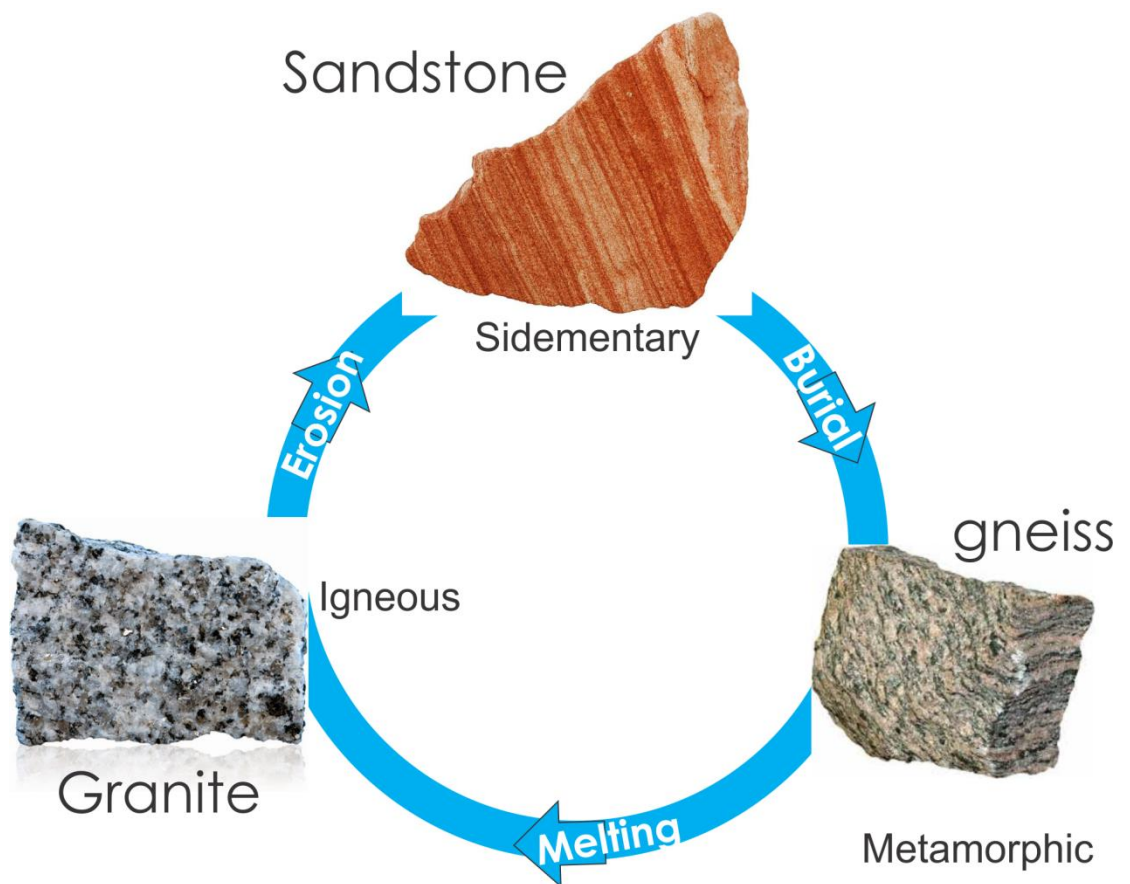
Mineral ada di semua batuan di Bumi. Mereka dapat ditemukan di mana pun batuan yang sudah diketahui, baik secara alami atau oleh manusia. Beberapa mineral kaya akan logam yang biasa digunakan dalam kehidupan sehari-hari, dan kita memanfaatkannya sebagai bijih.

Sebagian besar mineral yang membentuk sebagian besar batuan dan vena tidak bersifat logam atau terlalu berat, dan banyak yang tidak berwarna. Namun, ada beberapa pengecualian penting, dan beberapa yang paling kaya warna adalah mineral permata yang indah, tahan lama, dan langka

Bijih dan mineral sekundernya sering terdapat pada urat mineral, yang merupakan struktur seperti lembaran yang terjadi ketika mineral mengisi rekahan di dalam batuan yang ada. Banyak mineral bijih terlihat seperti logam, dan beberapa di antaranya terasa berat. Mineral sekunder dapat terbentuk ketika mineral bijih primer diubah oleh hujan dan air tanah. Mereka sering berwarna cerah, dan beberapa dari mereka sendiri mungkin memiliki nilai ekonomi.

Batuan adalah gabungan zat yang terjadi secara alami, yang terdiri dari mineral, potongan batuan lainnya, dan bahan fosil, seperti kerang atau tanaman. Batuan adalah hasil dari berbagai proses geologis yang terjadi secara alami di bawah permukaan bumi atau, dalam kasus lain seperti meteorit, di bagian lain dari Semesta. Batuan dapat dipelajari dan dibedakan antara dengan mengelompokkan jenis-jenis yang memiliki penampilan yang sama, komposisi yang sama, dan proses pembentukan yang sama.

Proses dinamis yang bekerja pada kerak bumi memungkinkan material batuan didaur ulang. Di permukaan bumi, pelapukan dan erosi memecah batuan yang sudah ada menjadi sedimen, yang membentuk batuan baru seperti batu pasir. Batuan ini dapat terkubur di bawah permukaan bumi; panas dan tekanan dari gerakan skala besar pada gilirannya menyebabkan fraktur, deformasi (perubahan yang disebabkan oleh tekanan), dan akhirnya meleleh. Misalnya, batu pasir diubah menjadi gneiss, dan gneiss yang meleleh membeku menjadi granit. Mengangkat bagian kerak bumi yang lebih dalam membawa bebatuan baru ini ke permukaan. Secara umum proses tersebut dapat dilihat pada gambar 2.1 (Monica, 2005).



Gambar 2. 1 *The Rock Cycle*

2.1.1 Identifikasi Batuan

Ada banyak fitur batuan yang dapat digunakan dalam identifikasi; ukuran dan bentuk biji-bijian, warna, dan penentuan mineral utama semuanya penting. Proses yang menghasilkan batuan juga menimbulkan tekstur dan struktur yang khas, misalnya, lava dapat menghasilkan batuan seperti kaca dengan struktur aliran.

1. Jenis Batuan

Ada tiga jenis utama batuan yaitu sedimen, beku, dan metamorf yang akan dijelaskan di bawah ini, tetapi beberapa jenis batuan lain adalah batuan deformasi, yang dihasilkan dari pergerakan Bumi;

meteorit; dan batuan tumbukan permukaan, yang diproduksi ketika meteorit menyerang Bumi.

- Batuan Sedimen

Batuan sedimen merupakan batuan yang terbentuk di permukaan bumi pada kondisi temperatur dan tekanan yang rendah. Batuan ini berasal dari batuan yang lebih dahulu terbentuk, yang mengalami pelapukan, erosi, dan kemudian lapukannya diangkut oleh air, udara yang selanjutnya diendapkan dan berakumulasi di dalam cekungan pengendapan, membentuk sedimen. Material-material sedimen itu kemudian terkompaksi, mengeras, mengalami litifikasi, dan terbentuklah batuan sedimen . Batuan sedimen terdiri dari berbagai macam jenis tergantung dari kandungan mineral yang terdapat di dalamnya (Devita, 2017).

- Batuan Beku

Batuan beku atau *igneous rock* (dari Bahasa Latin : ignis, “ api”) adalah jenis batuan yang terbentuk dari magma yang keluar dari perut bumi yang mendingin dan mengeras, dengan atau tanpa proses kristalisasi, baik di bawah permukaan sebagai batuan intrusif (plutonik) maupun di atas permukaan sebagai batuan

ekstusif (vulkanik). Pada saat magma mengalami penurunan suhu akibat perjalanan ke permukaan bumi, maka mineral-mineral akan terbentuk. Peristiwa tersebut dikenal dengan peristiwa penghabluran.

Berdasarkan penghabluran mineral-mineral silikat (magma), dimana komposisi magma berubah sifat dari basaltis → andesitik → rhyolitik oleh NL.Bowen (1887-1956) disusun suatu seri yang dikenal dengan *Bowen's Reaction Series*.

Berdasarkan teksturnya, batuan beku bisa dibedakan menjadi batuan beku plutonik dan vulkanik. Perbedaan antara keduanya bisa dilihat dari besar mineral penyusun batuan. Batuan beku plutonik umumnya terbentuk dari pembekuan magma yang relatif lebih lambat sehingga mineral mineral penyusunnya relatif besar. Contoh batuan ini seperti Gabro, Diorite, dan Granit. Sedangkan batuan beku vulkanik umumnya terbentuk dari pembekuan magma yang sangat cepat misalnya akibat letusan gunung api, sehingga mineral penyusunnya lebih kecil. Contoh batuan ini yaitu Basalt, Andesit, Dacite (Annisa, 2017).

- Batuan Metamorf

Batuan metamorposis atau metamorfosa atau metamorf

(*metamorphic rock*) membentuk sebagian besar kerak bumi dan membentuk 12% luas permukaan bumi.

Batuan ini diklasifikasikan berdasarkan tekstur, kandungan kimia dan

mineral. Batuan ini mungkin terbentuk berada jauh di bawah permukaan bumi, mengalami suhu tinggi dan

tekanan besar oleh lapisan batu di atasnya. Batuan ini

juga dapat terbentuk dari proses tektonik seperti benturan kontinental, yang menyebabkan tekanan

horisontal, gesekan dan distorsi. Batuan metamorf juga

terbentuk saat batuan dipanaskan oleh intrusi batuan cair panas yang disebut magma yang berasal dari

interior bumi. Studi tentang batuan metamorf yang

tersingkap / terpapar di permukaan bumi memberikan informasi tentang suhu dan tekanan yang terjadi pada

kedalaman yang dalam di dalam kerak bumi.

Metamorfosis adalah himpunan proses dimana batuan mengalami perubahan mineralogi, tekstur, atau

keduanya untuk mencapai ekuilibrium (keseimbangan)

dengan lingkungannya pada kondisi solid/padat. Istilah

metamorphism berasal dari bahasa Yunani "meta" yang berarti "berubah" dan "morph" yang berarti "bentuk". Dapat di simpulkan bahawa batuan metamorfosa adalah transisi satu batu ke yang lain oleh suhu dan atau tekanan dan membentuk batuan baru. Batuan metamorfik dihasilkan dari perubahan suhu pada proses metamorfosa (batuan induk): Batuan beku, Batuan sedimen atau Batu metamorf lainnya (repository.unri.ac.id/).

2. Kandungan Mineral

Beberapa mineral terbatas pada jenis batuan tertentu, dan menentukan kandungan mineral dapat membantu identifikasi batuan tersebut. Butir mineral yang jelas dapat memberikan petunjuk pada identitas batu, misalnya, garnet hanya muncul di batuan metamorf (Monica, 2005).

3. Ukuran Grain

Dalam batuan sedimen, ukuran butir (*grain*) tergantung pada sejauh mana butiran telah melakukan perjalanan; untuk batuan beku ditentukan oleh berapa lama batuan itu mengkristal; ukuran butir (*grain*) metamorf tergantung pada batuan yang sudah ada sebelumnya. Batas ukuran untuk butiran kasar, sedang, dan halus bervariasi sesuai dengan kelompok batuan.

4. Bentuk Grain

Bentuk butiran individu lebih mudah diamati menggunakan lensa tangan. Beberapa fitur yang perlu dipertimbangkan termasuk tingkat kebulatan, apakah semua butir memiliki bentuk yang serupa, dan seberapa baik kristal yang dikembangkan. Kristal yang berkembang dengan baik memiliki permukaan datar dan ujungnya lurus.

5. Tekstur

Tekstur mengacu pada penampilan permukaan batu atau hubungan antara butir individu dalam batu. Jika batu itu tidak memiliki butiran yang terlihat dan merupakan satu massa mineral atau gelas, ia dikenal sebagai masif. Banyak batuan beku tersusun dari butiran-butiran yang saling bertautan - suatu tekstur yang dikenal sebagai butiran.

6. Struktur

Struktur mengacu pada fitur-fitur yang dihasilkan oleh proses geologi. Beberapa struktur, seperti *bedding*, pelapisan beku, dan pita gneis, terbentuk bersamaan dengan batu itu sendiri, tetapi yang lain, seperti pelipatan dan pencukuran (peregangan butir mineral), terjadi kemudian. Banyak struktur ada dalam skala yang luas, misalnya lipatan dapat dilihat pada butiran kristal mikroskopis dan juga di seluruh permukaan tebing. Ahli geologi menggunakan

informasi struktural untuk membangun sejarah geologi dan menghasilkan peta geologi.

7. Warna

Warna adalah sifat yang jelas tetapi bermanfaat dalam identifikasi batuan. Ini dapat digunakan untuk membedakan berbagai jenis batuan, dan juga dapat memberikan informasi tentang komposisi batuan. Dalam banyak kasus, warna batu akan dipengaruhi oleh pelapukan, sehingga permukaan yang segar harus selalu terbuka.

8. Lingkungan Geologis

Jenis batuan jarang terjadi secara terpisah. Biasanya, beberapa jenis batuan yang berbeda tetapi saling berhubungan membentuk satu kesatuan dalam satu lingkungan geologis. Misalnya, bom basal, scoria, spilite, dan vulkanik ditemukan bersama di gunung berapi Hawaii. Mengetahui asosiasi ini dapat membantu mengidentifikasi batu yang terjadi bersama dan, juga, untuk mengenali lingkungan purba mereka - greensand dan kapur pada Zaman Kapur adalah bukti dari laut yang hangat dan tenang 100 juta tahun yang lalu.

9. Tempat ditemukan batuan

Batuan ditemukan di mana-mana, tetapi contoh terbaik ditemukan di mana efek dari pelapukan paling sedikit, seringkali dalam pengaturan yang diciptakan oleh aktivitas manusia, seperti

pemotongan jalan, pembuangan tambang, dan penggalian. Contoh-contoh yang baik juga ditemukan di mana alam baru-baru ini mengekspos batuan segar, seperti di tebing laut, pantai, gunung, dan medan vulkanik. Banyak batu yang lebih keras, lebih tahan, dan lebih menarik digunakan untuk membangun; ini dapat dilihat sebagai struktur pendukung, serta dekorasi interior dan eksterior. Peta geologis menunjukkan distribusi batuan di wilayah tertentu, dan dapat menjadi alat yang penting dan berguna untuk menemukan dan mengenali batuan.

2.1.2 Identifikasi Mineral

Fakta bahwa satu mineral dapat terjadi dalam berbagai bentuk kristal dan dalam berbagai warna dapat membingungkan. Namun, setiap mineral dapat diidentifikasi dengan menginvestigasi kombinasi sifat. Beberapa properti, seperti warna, terlihat oleh mata; yang lain, seperti kekerasan, perlu diukur menggunakan peralatan sederhana.

1. Komposisi Mineral

Mineral dibagi menjadi kelas-kelas berikut sesuai dengan komponen bermuatan negatifnya, misalnya karbonat (CO_3^{2-}). Banyak sifat suatu mineral hasil dari susunan kimianya, misalnya, warna, garis, dan magnetisme.

2. Kelompok Mineral

Beberapa mineral dikelompokkan bersama karena memiliki komposisi kimia dan struktur kristal yang saling berkaitan. Sebagai

akibatnya, mereka berbagi properti kunci tertentu dan mungkin sulit dibedakan. Banyak mineral silikat milik kelompok, yang meliputi feldspars, garnet, micas, amphiboles, dan pyrox.

3. Bentuk Kristal

Bentuk kristal adalah bentuk geometrisnya. Beberapa kristal adalah bentuk tunggal, seperti octahedron; yang lain lebih kompleks dan menggabungkan dua bentuk atau lebih, misalnya, kristal kuarsa menggabungkan prisma dengan dua piramida.

4. Kristal Kembar

Kristal kembar terbentuk ketika bagian-bagian berbeda dari satu kristal tumbuh sebagai gambar cermin satu sama lain. Ini dapat bergabung pada permukaan kristal, tepi, atau internal. Si kembar sederhana memiliki dua bagian; beberapa kembar memiliki tiga atau lebih. Kembaran memberikan mineral bentuk kristal yang berbeda, misalnya, aragonit tampak heksagonal.

5. Sistem Kristal

Bentuk kristal mineral menunjukkan sistem kristal mana yang dimiliki mineral tersebut. Sebagai contoh, kristal memanjang dengan penampang persegi, seperti yang vesuvianite, ditemukan dalam mineral yang termasuk dalam sistem tetragonal. Ada tujuh sistem kristal, dan masing-masing memiliki elemen simetri tertentu. Setiap bentuk mineral akan selalu memiliki elemen simetri dari sistem yang dimilikinya.

6. Kebiasaan Mineral

Kebiasaan mineral menggambarkan penampilan umum kristal atau kelompok kristal. Mineral dapat memiliki lebih dari satu kebiasaan. Sebagai contoh, aktinolit dapat berbilah, asikular, berserat, atau masif, tergantung pada kondisi di mana ia terbentuk. Kebiasaan besar menunjukkan bukan ukuran tetapi tidak adanya bentuk kristal yang terlihat. Akibatnya, sampel kecil mineral dapat menjadi sangat besar.

7. Cleavage and Fracture

Istilah-istilah ini menggambarkan cara mineral pecah. Pembelahan mengacu pada bidang datar di mana kristal cenderung membelah; itu dinilai dari miskin menjadi sempurna tergantung pada seberapa bersih dan mudahnya pecah. Fraktur menggambarkan permukaan di mana mineral rusak, tetapi di mana itu belum membelah. Perpisahan tampak seperti belahan gunung, tetapi memiliki penyebab lain.

8. Transparansi

Hampir semua mineral yang ditampilkan transparan atau tembus kecuali dinyatakan sebaliknya. Terkadang transparansi hanya dapat terlihat ketika melihat melalui lapisan mineral yang tipis.

9. Warna

Beberapa mineral selalu memiliki warna yang sama, sebagai konsekuensi dari komposisi dan struktur kimianya. Warna mereka adalah bantuan yang berguna untuk identifikasi. Mineral lain, seperti fluorit, bisa berwarna berbeda karena pengotor kimia dalam jumlah kecil, cacat struktural, atau karena kebiasaannya yang berbeda. Mineral tertentu bersifat fluoresen - warnanya berbeda di bawah sinar ultraviolet (UV)

10. Kilau

Kilau menjelaskan cara mineral memantulkan cahaya. Adamantine adalah kilau paling cemerlang yang ditunjukkan oleh mineral transparan atau tembus cahaya. Kilau seperti kaca kurang cemerlang. Pengkilap sedikit lebih rendah diawali 'sub', misalnya, kilau subadamantin sedikit kurang cerah dari adamantine. Kilau mineral mungkin berbeda pada wajah atau belahan tertentu, dan itu harus selalu diperiksa pada bagian mineral yang bersih dan tidak terkena cuaca.

11. Garis

Warna mineral bubuk halus disebut garis. Bahkan jika warna mineral bervariasi, goresan akan selalu sama untuk mineral. Misalnya, tidak peduli jenisnya, hematit memiliki garis coklat kemerahan. Garis diuji dengan menggambar mineral tersebut di

atas pelat coretan, yang merupakan bagian belakang putih ubin keramik tanpa glasir

12. Kekerasan

Seberapa keras suatu mineral tergantung pada bagaimana atom-atomnya diatur. Kekerasan diukur dengan mengacu pada sepuluh mineral standar pada skala Mohs, mineral paling lembut adalah talc, dan berlian paling keras. Pada skala Mohs, kristal yang menggores kalsit tetapi tergores oleh fluorit adalah 3,5 pada skala. Dimungkinkan untuk membedakan mineral yang tampak serupa dengan mengukur kekerasannya.

13. Kepadatan

Sebagian besar mineral memiliki kerapatan antara 2 dan 4, tetapi ada juga yang secara nyata lebih ringan atau lebih berat. Sebagai contoh, kristal baryte, di bawah, akan terasa sangat berat dibandingkan dengan gipsum. Kepadatan diukur dalam jumlah gram per sentimeter kubik.

14. Magnetisme

Beberapa mineral yang mengandung besi bersifat magnetis; ketika ditempatkan di samping kompas, mereka membelokkan jarum dari utara. Magnetit, magnet alami yang sangat kuat, menarik benda besi atau baja.

15. Tes Asam

Jika setetes kecil asam klorida encer (HCl) ditempatkan pada mineral karbonat tertentu, ia mendesis dan mengeluarkan gelembung gas karbon dioksida. Mineral tertentu bereaksi terhadap tes asam spot dengan cara khas lainnya, dengan mengubah warna atau melarutkan. Harap dicatat, semua asam bisa berbahaya: mereka harus selalu ditangani dengan sangat hati-hati.

16. Radioaktivitas

Mineral yang mengandung uranium atau thorium bersifat radioaktif. Radioaktivitas terdeteksi menggunakan penghitung Geiger. Semua mineral radioaktif membutuhkan penanganan dan penyimpanan khusus.

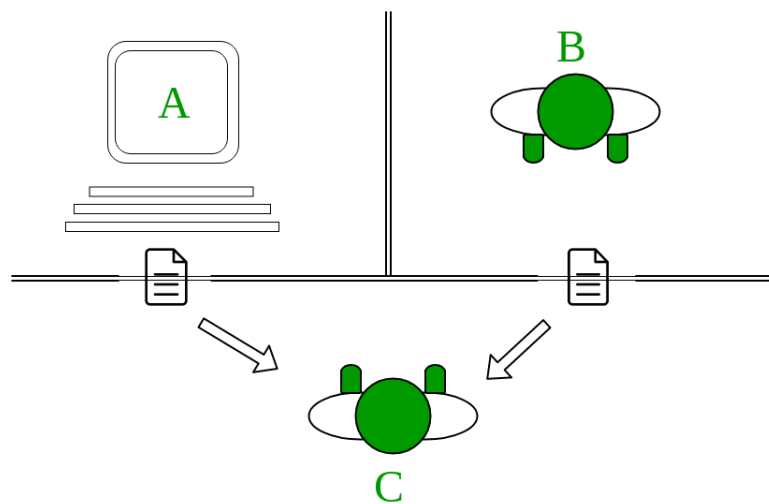
17. Asosiasi Geologi

Beberapa petunjuk paling penting untuk identitas mineral berasal dari matriks batuan di mana itu terjadi dan mineral lain yang terjadi dengannya. Beberapa mineral hanya terjadi pada jenis batuan tertentu; yang lain biasanya dikaitkan dengan mineral lain tertentu. Sebagai contoh, apophyllite sering terjadi dengan stilbite.

2.2 Artificial Intelligence

Artificial intelligence atau kecerdasan buatan seperti yang kita kenal sekarang dimulai pada tahun 1950-an. Hal itu dimulai ketika Alan Turing melakukan test kecerdasan buatan atau yang lebih dikenal dengan *turing test* yang bertujuan untuk mendefinisikan sebuah mesin cerdas. *Turing test* dilakukan

dengan cara meletakkan mesin pada ruang A , manusia pada ruang B dan seorang penguji pada ruang C. Penguji diminta berkomunikasi dengan mesin di ruang A dan manusia di ruang B, jika penguji di ruang C tidak dapat membedakan antara mesin di ruang A dan manusia di ruang B maka *turing test* ini dianggap berhasil, untuk ilustrasinya dapat dilihat pada gambar 2.2 .



Gambar 2. 2 Ilustrasi Turing test

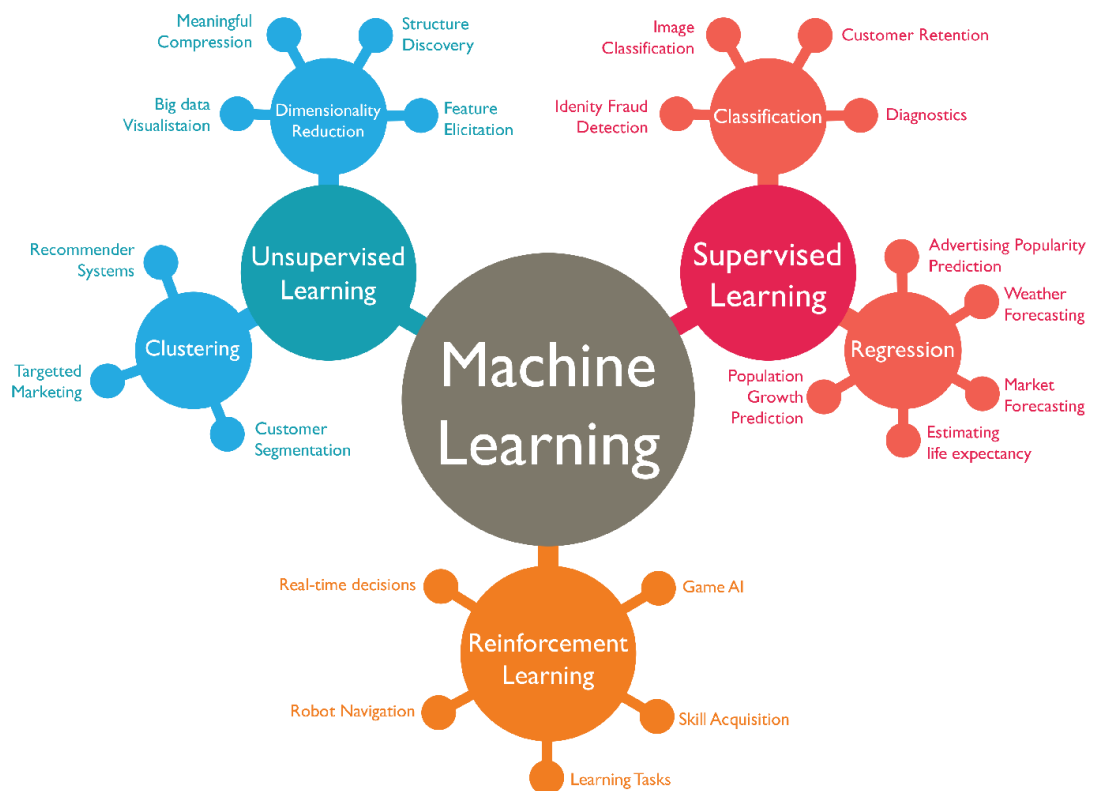
<https://www.geeksforgeeks.org/turing-test-artificial-intelligence/>

Kecerdasan buatan atau *artificial intelligence* secara umum dapat dijelaskan sebagai kecerdasan yang diberikan kepada mesin, hingga mesin dapat bertindak seperti manusia. Menurut John McCarthy (1956), kecerdasan buatan adalah suatu sistem komputer yang terbentuk untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Sehingga dalam kehidupan sehari-hari kita dapat mengartikan kecerdasan buatan adalah masa depan, kecerdasan buatan adalah fiksi ilmiah, dan sebagainya karena pada dasarnya kecerdasan buatan sudah menjadi bagian dari kehidupan kita sehari-hari.

2.3 Machine Learning

Machine learning merupakan pendekatan dalam Kecerdasan buatan yang banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Istilah *machine Learning* pada dasarnya adalah proses komputer untuk belajar dari data (*Learn from data*). Menurut Arthur Samuel, *machine learning* adalah bidang studi yang memberikan komputer kemampuan untuk belajar tanpa diprogram secara eksplisit. Sebagai contohnya dapat kita lihat pada filter spam kita adalah program *machine learning* yang dapat belajar menandai *spam* dari contoh beberapa email spam dan email biasa. Contoh yang digunakan sistem untuk belajar disebut set pelatihan. Setiap contoh pelatihan disebut contoh pelatihan (atau sampel). Dalam hal ini, tugas T adalah menandai spam untuk email baru, pengalaman E adalah data pelatihan, dan ukuran kinerja P perlu didefinisikan; misalnya, Anda dapat menggunakan rasio email yang diklasifikasikan dengan benar. Ukuran kinerja khusus ini disebut akurasi dan sering digunakan dalam tugas klasifikasi (aurelien, 2017).

Machine learning tentu saja merupakan domain yang sangat luas. Alhasil, *machine learning* telah bercabang menjadi beberapa sub bidang yang berurusan dengan berbagai jenis tugas belajar. Secara garis besar *machine learning* dibagi menjadi 3 subbagian yaitu, *supervised learning*, *unsupervised learning*, dan *reinforcement learning* seperti yang tertera pada gambar 2.3



Gambar 2.3 Subbagian machine learning

2.3.1 Supervised Learning

Supervised learning adalah tipe *learning* di mana kita mempunyai variabel *input* dan variabel *output*, dan menggunakan satu algoritma atau lebih untuk mempelajari fungsi pemetaan dari *input* ke *output*. Tujuan dari metode ini adalah agar algoritma dapat belajar dengan membandingkan

output aktualnya dengan *output* yang diajarkan untuk menemukan kesalahan, dan memodifikasi model yang sesuai. Oleh karena itu *supervised learning* menggunakan pola untuk memprediksi nilai label pada data yang tidak berlabel tambahan.

Sebagai contoh dengan *supervised learning*, suatu algoritma dapat memasukkan data dengan gambar hiu yang dilabeli sebagai ikan dan gambar lautan yang berlabel air. Kemudian dilatih tentang data ini, algoritma *supervised learning* harus mampu mengidentifikasi gambar hiu yang tidak berlabel dan mengidentifikasinya sebagai ikan dan gambar laut yang tidak berlabel sebagai air. Proses dari sebuah algoritma belajar dari *training* dataset dapat diumpamakan sebagai seorang guru yang mengawasi (*supervising*) proses belajar. Kita tahu jawaban yang benar, dan algoritma secara iteratif membuat prediksi pada data latih (*training data*) dan dikoreksi oleh 'guru'-nya.

Learning berhenti ketika algoritma mencapai level performansi yang diterima. Permasalahan *Supervised Learning* dapat dikelompokkan menjadi masalah regresi (*regression problem*) dan masalah klasifikasi (*classification problems*).

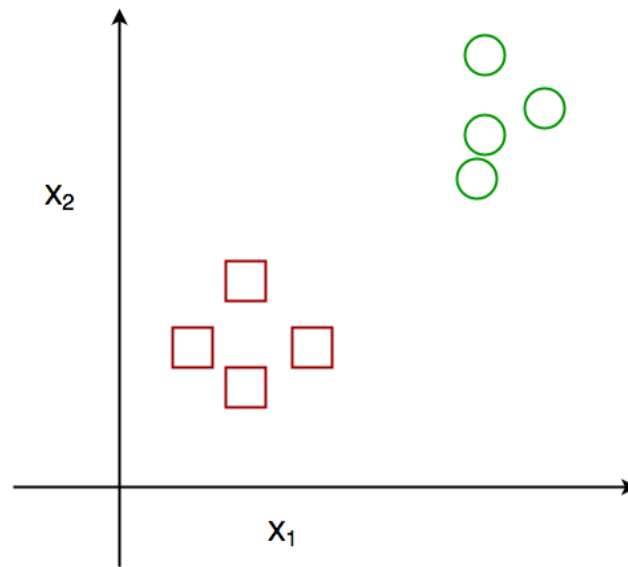
- ***Regression***

Regression adalah ketika variabel *output* adalah nilai nyata atau berkelanjutan, seperti "gaji" atau "berat". Banyak model yang berbeda dapat digunakan, yang paling sederhana adalah regresi linier. Sehingga dapat disimpulkan bahwa

regression adalah proses menemukan model atau fungsi untuk membedakan data menjadi nilai riil berkelanjutan daripada menggunakan kelas atau nilai diskrit. Sehingga dapat mengidentifikasi pergerakan distribusi tergantung pada data *historis*.

- ***Classification***

Classification adalah proses mencari atau menemukan model atau fungsi yang membantu dalam memisahkan data menjadi beberapa kelas kategori. Dalam *Classification*, data dikategorikan dalam label yang berbeda sesuai dengan beberapa parameter yang diberikan dalam *input* dan kemudian label diprediksi untuk data tersebut. *Output* yang dikategorikan dapat memiliki bentuk seperti “Merah” atau “Hitam” atau “Ya” atau “Tidak”. Secara matematis, *classification* adalah tugas mendekati fungsi pemetaan (f) dari variabel *input* (X) ke variabel *output* (Y). Ini pada dasarnya milik pembelajaran mesin yang diawasi di mana target juga disediakan bersama dengan set data *input*.



Gambar 2. 4 *Classification*

2.3.2 *Unsupervised Learning*

Unsupervised learning mempelajari bagaimana sistem dapat belajar untuk mewakili pola *input* tertentu dengan cara yang mencerminkan struktur statistik dari seluruh kumpulan pola yang diinput (peter dayan). Pendekatan *unsupervised learning* tidak menggunakan data latih atau data *training* untuk melakukan prediksi maupun klasifikasi. Berdasarkan model matematisnya, algoritma ini tidak memiliki target variabel. Salah satu tujuan dari algoritma ini adalah mengelompokkan objek yang hampir sama dalam suatu area tertentu. Hal ini tentu berbeda dengan *supervised learning* yang mana dari data *input* pasti ada *output* berupa “ya” atau “tidak” pada *unsupervised learning* lebih menjelaskan data atau menyimpulkan data tanpa adanya *output*. Sebagai contohnya *unsupervised learning* umumnya digunakan untuk data transaksional.

Dengan memiliki kumpulan data pelanggan yang besar dan pembelian *customer*, tetapi sebagai manusia kita mungkin tidak akan dapat memahami atribut apa yang dapat diambil dari profil pelanggan dan jenis pembelian mereka. Dengan data ini dimasukkan ke dalam algoritma pembelajaran tanpa pengawasan, dapat ditentukan bahwa wanita dari rentang usia tertentu yang membeli sabun tanpa pewangi kemungkinan akan hamil, dan oleh karena itu kampanye pemasaran yang berkaitan dengan kehamilan dan produk bayi dapat ditargetkan untuk customer ini dalam rangka untuk meningkatkan jumlah pembelian mereka.

Clustering adalah konsep penting dalam *unsupervised learning*. Terutama berkaitan dengan menemukan struktur atau pola dalam kumpulan data yang tidak dikategorikan. Algoritma *clustering* akan memproses data dan menemukan *cluster* (kelompok) secara alami jika ada dalam data tersebut. Sehingga kita dapat memodifikasi berapa banyak cluster yang harus diidentifikasi oleh algoritma yang digunakan.

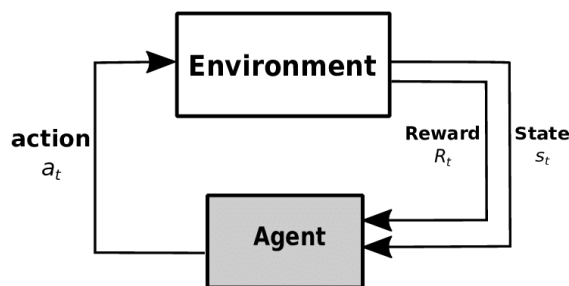
Tujuan dari *clustering* adalah untuk menentukan pengelompokan internal dalam satu set data yang tidak berlabel. Tetapi bagaimana cara memutuskan yang mana merupakan pengelompokan yang baik? Dapat ditunjukkan bahwa tidak ada kriteria absolut “terbaik” yang tidak tergantung pada tujuan akhir pengelompokan. Sehingga pengguna yang harus menentukan kriteria sendiri, sedemikian rupa sehingga hasil pengelompokan akan sesuai dengan kebutuhan pengguna.



Gambar 2. 5 Clustering

2.3.3 Reinforcement Learning

Reinforcement learning adalah salah satu bidang *machine learning*, yang mana mesin dapat mengambil tindakan yang sesuai untuk memaksimalkan keputusan yang diambil dalam kondisi tertentu. *Reinforcement learning* berbeda dengan *supervised learning* yang mana dalam *supervised learning* data pelatihan memiliki kunci jawaban dengan itu sehingga model dilatih dengan jawaban yang benar itu sendiri sedangkan dalam *reinforcement learning*, tidak ada jawaban tetapi mesin dalam hal yang memutuskan apa yang harus dilakukan untuk melakukan tugas yang diberikan. Dengan tidak adanya dataset pelatihan, ia pasti akan belajar dari pengalamannya.



Gambar 2. 6 Reinforcement learning

2.4 Deep Learning

Deep learning adalah bagian dari *machine learning* yang mengajarkan komputer untuk melakukan apa yang terjadi secara alami pada manusia: belajar dengan contoh. *Deep learning* adalah teknologi utama di balik mobil tanpa pengemudi, memungkinkan mengenali tanda berhenti, atau untuk membedakan pejalan kaki dari tiang lampu. Ini adalah kunci untuk kontrol suara di perangkat konsumen seperti ponsel, tablet, TV, dan *speaker hands-free*. Pada *deep learning*, model komputer belajar untuk melakukan tugas-tugas klasifikasi langsung dari gambar, teks, atau suara. Model dari *deep learning* dapat mencapai akurasi yang tinggi, terkadang melebihi kinerja manusia. Model dilatih dengan menggunakan data berlabel dalam jumlah yang besar dan *neural network* untuk menyelesaikan permasalahan.

sebagian besar metode *deep learning* menggunakan *neural network architecture*, itulah sebabnya mengapa model *deep learning* sering disebut sebagai *deep neural network*. Istilah "*deep*" biasanya mengacu pada jumlah lapisan tersembunyi di *neural network*. *Neural network* tradisional hanya mengandung 2-3 lapisan tersembunyi, sedangkan *deep neural network* dapat memiliki sebanyak 150 lapisan.

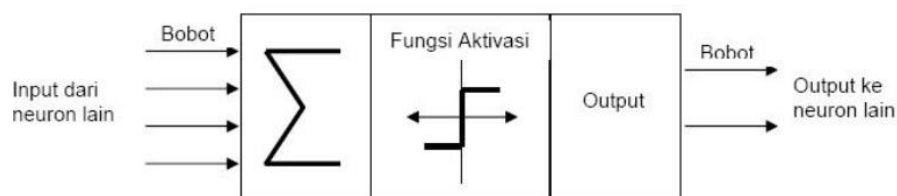
Feature Engineering adalah salah satu fitur utama dari *deep Learning* untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. *Feature Engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda

memerlukan pendekatan teknik yang berbeda juga. Algoritma yang digunakan pada *Feature Engineering* dapat menemukan pola umum yang penting untuk membedakan antara kelas dalam *deep learning*, metode CNN atau *Convolutional Neural Network* sangatlah cocok dalam menemukan fitur yang

baik pada citra untuk membentuk hipotesis nonlinier yang dapat meningkatkan kekompleksitasan sebuah model. Model yang kompleks tentunya akan membutuhkan waktu pelatihan yang lama sehingga di dunia *deep learning* penggunaan GPU sudah sangatlah umum (Danukusumo, 2017)

2.5 Artificial Neural Network

Artificial neural networks adalah bagian dari *deep learning*, *artificial neural network* merupakan implementasi kerja otak kita yang tersusun jaringan saraf yang disebut neuron. Seperti halnya sistem otak manusia, dalam *artificial neural network* terdiri dari beberapa neuron yang saling berhubungan. Masing-masing neuron akan menerima informasi dan mengakumulasikannya dari neuron yang tersambung sebelumnya. Hubungan ini disebut dengan sebutan bobot (*Weight*). Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tertentu (Farid, 2020).



Gambar 2. 7 Arsitektur neuron *neural network*

Pada *neural network*, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layers*). Biasanya

neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan *input* dan lapisan *output*). Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan *input* sampai ke lapisan *output* melalui lapisan yang lainnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*). Tergantung pada algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan secara mundur pada jaringan.

Artificial neural network adalah sekumpulan neuron yang terorganisir dalam lapisan-lapisan , diantaranya :

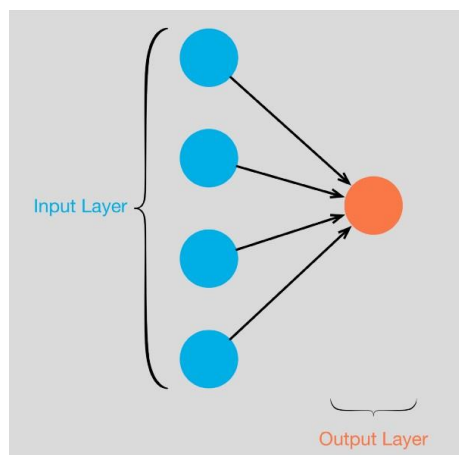
- *input layer* : lapisan yang membawa data masuk kedalam sistem untuk kemudian di proses pada layer selanjutnya.
- *hidden layer*: lapisan antara *input layer* dan *output layer*, dimana *artificial neuron* yang memiliki sekumpulan *input* pembobot ‘weight’ dan prosedur untuk menghasilkan output neuron melalui activation function.
- *output layer*: lapisan terakhir dari neuron yang menghasilkan *output* sistem.

Ada beberapa pemodelan pada *artificial neural network* , antara lain:

- ***Single Layer***

Dalam *artificial neural network*, neuron disusun dalam bentuk lapisan (layer). Pembentukan *artificial neural network* yang paling sederhana yaitu single layer. Cara kerja dari *single layer*, *input layer* yang berasal dari sumber node di proyeksikan langsung ke *output layer* dari neuron (node komputasi), tetapi tidak berlaku sebaliknya. Permodelan ini

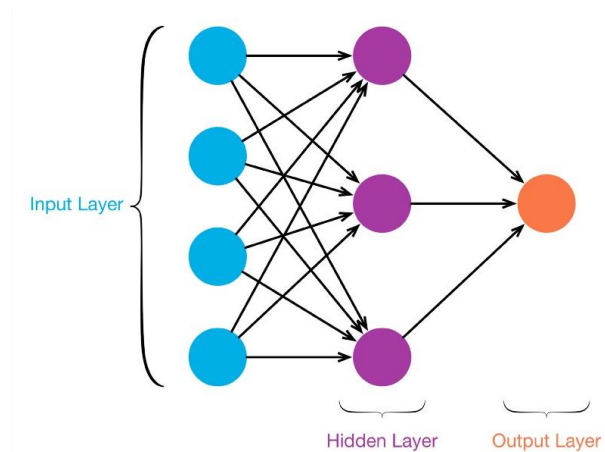
merupakan jenis jaringan *feedforward* yang dapat dilihat pada gambar 2.8. Pada gambar tersebut *input* dan output memiliki 4 node, namun yang dimaksud dengan *single layer* yaitu output dari jaringan, sedangkan *inputnya* tidak memiliki pengaruh karena pada saat melakukan *input* tidak terjadi proses komputasi (yaqutina, 2018).



Gambar 2. 8 *Single layer artificial neural network*

- **Multi Layer**

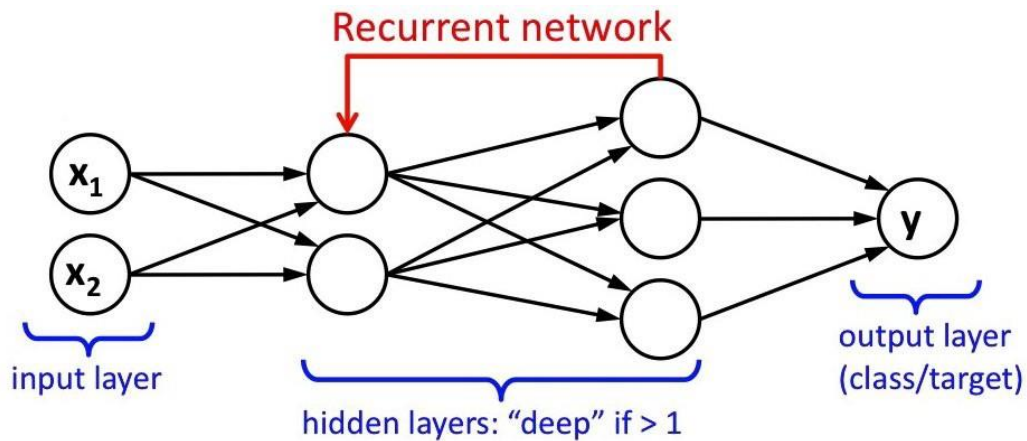
Pada *single layer* apabila terdapat tambahan satu atau dua *hidden layer* maka jaringan akan terganggu karena *input* dan output dari jaringan tidak dapat melihat *hidden layer* yang di masukkan. Sehingga memerlukan jaringan yang bisa menampung nya yaitu bernama *multi layer*. Cara kerja *multi layer* adalah *input layer* menyuplai *input* vektor pada jaringan, kemudian *input* yang dimasukkan melakukan komputasi pada *layer* yang kedua, lalu *output* dari *layer* yang kedua digunakan sebagai *input* dari *layer* yang ketiga dan seterusnya. Ilustrasi jaringan *multi layer* dapat di lihat pada gambar 2.9 (yaqutina, 2018).



Gambar 2. 9 *Multi layer artificial neural network*

- ***Recurrent Network***

Recurrent network terbentuk karena pada jaringan single layer dan multi layer harus memiliki feedback untuk dirinya sendiri pada setiap loop jaringan nya, pada *recurrent network* jaringan tidak memerlukan feedback untuk dirinya sendiri melainkan feedback dari input yang digunakan. Ilustrasi jaringan *recurrent network* dapat dilihat pada gambar 2.10 (yaqutina, 2018).



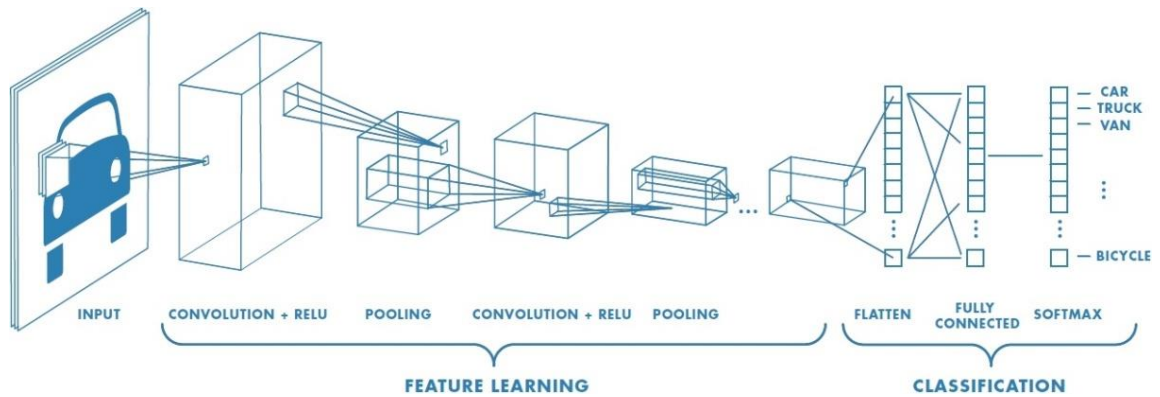
Gambar 2. 10 *Recurrent neural network*

2.6 Convolutional Neural Network

Convolutional neural network adalah salah satu algoritma paling populer dalam *deep learning*, bagian dari *machine learning* di mana model belajar untuk melakukan tugas-tugas klasifikasi langsung dari gambar, video, teks, atau suara. Secara garis besar *convolutional neural network* tidak jauh beda dengan *neural network* biasanya. *Convolutional neural network* biasa digunakan menemukan pola dalam gambar untuk mengenali objek, wajah, dan adegan. Model belajar langsung dari data gambar, menggunakan pola untuk mengklasifikasikan gambar dan menghilangkan kebutuhan untuk ekstraksi fitur manual.

Arsitektur *convolutional neural network* standarnya menumpuk beberapa lapisan konvolusional (masing-masing umumnya diikuti oleh layer ReLU), kemudian *pooling layer*, lalu beberapa lapisan konvolusional lainnya (+ ReLU), lalu *pooling layer* lain, dan seterusnya. Gambar semakin kecil dan semakin kecil seiring banyaknya jaringan, dan juga akan semakin dalam dan dalam dengan kata lain lebih banyak *layer* (dengan lebih banyak peta fitur) berkat lapisan konvolusional. Di bagian atas CNN, terdapat *feed forward neural network regular*

ditambahkan, terdiri dari beberapa *fully connected layer* (+ ReLUs), dan lapisan akhir menampilkan prediksi (lapisan softmax yang menghasilkan perkiraan probabilitas kelas) (Aurélien, 2017).

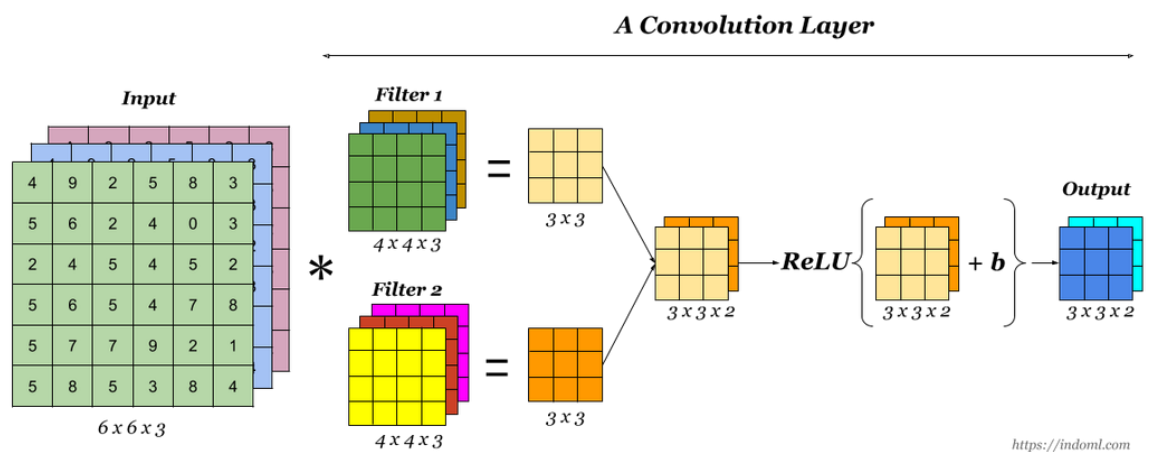


Gambar 2. 11 Arsitektur *convolutional nueral network*

2.6.1 *Convolution Layer*

Convolution layer adalah lapisan pertama yang mengekstraksi fitur dari gambar input. *Convolution layer* menjaga hubungan antara piksel dengan mempelajari fitur gambar menggunakan kotak kecil data input. Ini adalah operasi matematika yang mengambil dua *input* seperti matriks gambar dan filter atau kernel. *Convolution layer* sebenarnya menggunakan korelasi silang, yang sangat mirip dengan *konvolusi*. Pada *convolutional neural network*, neuron di lapisan konvolusional pertama tidak terhubung ke setiap piksel tunggal atau *full connected*, tetapi hanya piksel di bidang. Setiap neuron dalam lapisan konvolusional kedua terhubung hanya ke neuron yang terletak di dalam persegi panjang kecil di lapisan pertama. Arsitektur ini memungkinkan jaringan untuk lebih fokus

pada fitur tingkat rendah di *hidden layer* pertama, lalu membuatnya menjadi fitur tingkat tinggi di *hidden layer* berikutnya, dan seterusnya. Struktur hierarkis ini umum digunakan dalam gambar di dunia nyata, yang merupakan salah satu alasan mengapa *convolutional neural network* bekerja dengan baik untuk pengenalan gambar (Aurélien, 2017).

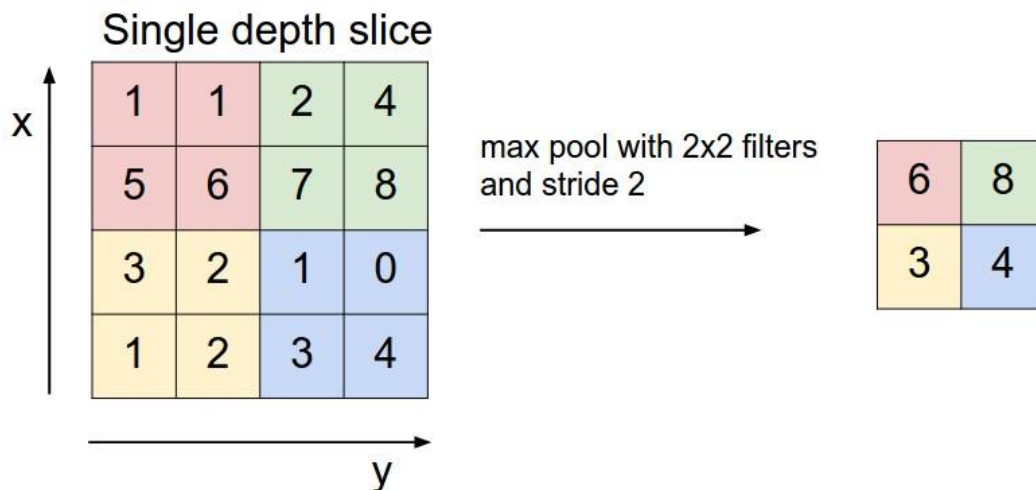


Gambar 2. 12 Convolution layer

2.6.2 Pooling

Bagian berikutnya dari *convolutional neural network* adalah *pooling layer*. Fungsi dari *pooling* ini adalah untuk mereduksi input secara spasial (mengurangi jumlah parameter) dengan operasi *down-sampling*. Umumnya, metode *pooling* yang digunakan adalah *max pooling* atau mengambil nilai terbesar dari bagian tersebut. Namun terdapat metode *pooling* lain yang dapat digunakan seperti *average pooling* atau *L2-norm pooling*.

Pooling layer beroperasi secara independen pada setiap irisan kedalaman input dan mengubah ukurannya secara spasial, menggunakan *max operation*. Bentuk paling umum adalah *pooling* layer dengan filter ukuran 2x2 diterapkan dengan langkah 2 *downsamples* setiap irisan kedalaman dalam input dengan 2 sepanjang lebar dan tinggi, membuang 75% dari aktivasi. Setiap *max operation* dalam hal ini akan mengambil max lebih dari 4 angka (wilayah 2x2 kecil dalam beberapa irisan mendalam). Dimensi kedalaman tetap tidak berubah. Lebih umum, lapisan penyatuan.

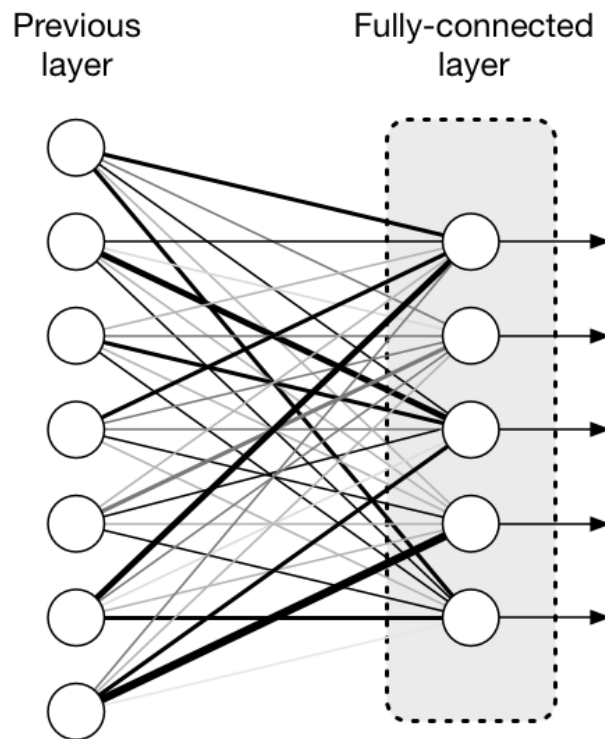


Gambar 2. 13 Max Pooling

2.6.3 Fully Connected Layer

Fully connected layer adalah komponen penting dari *convolutional neural network*, hal ini dibuktikan dengan hasil yang baik dalam mengenali dan mengklasifikasikan gambar untuk penglihatan komputer. Proses *convolutional neural network* dimulai dengan *convolution layer* dan *pooling*, memecah gambar menjadi fitur, dan menganalisisnya secara

independen. Hasil dari proses ini dimasukkan ke dalam struktur *neural network* yang terhubung dalam pengambilan keputusan klasifikasi akhir.



Gambar 2. 14 *Fully Connected Layer*

2.7 Website

Website adalah kumpulan informasi/kumpulan page yang biasa diakses lewat jalur internet. Setiap orang di berbagai tempat dan segala waktu bisa menggunakannya selama terhubung secara online di jaringan internet. Secara teknis, website adalah kumpulan dari page, yang tergabung kedalam suatu domain atau subdomain tertentu. Website-website yang ada berada di dalam *World Wide Web* (WWW) Internet.

Pengertian Website Menurut Para Ahli

- Website adalah fasilitas internet penghubung dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada website disebut dengan web

page sementara link dalam website memungkinkan pengguna bisa berpindah dari satu page ke page lain (*hyper text*), baik diantara page yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. Pages diakses dan dibaca lewat browser seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome dan aplikasi browser lainnya (pengertian website dari Hakim Lukmanul, 2004)

- Website adalah kumpulan halaman web yang saling terhubung dan file-filenya saling terkait satu sama lain. Web terdiri dari page/halaman, dan kumpulan halaman yang disebut *homepage*. *Homepage* berada pada posisi teratas, dengan halaman-halaman terkait yang berada di bawahnya. Biasanya setiap halaman di bawah *homepage* disebut child page, yang berisi *hyperlink* ke halaman lain dalam web (pengertian website dari Gregorius, 2000)
- Website adalah suatu metode untuk menampilkan informasi pada internet, baik berupa teks, gambar, suara atau video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*), dapat diakses melalui sebuah browser (pengertian website dari Yuhefizar, 1998)

Orang yang berjasa tersebut adalah Sir Tim Berners-Lee. Ia seorang ilmuwan komputer di Inggris. Berners dilahirkan di London, dan orang tuanya juga ilmuwan komputer di era-era awal komputasi.

Setelah Berners lulus dari Universitas Oxford, Berners-Lee akhirnya menjadi insinyur software di CERN. CERN adalah laboratorium fisika (partikel besar)

yang berada di dekat Jenewa, Swiss. Akhirnya, para ilmuwan yang hebat datang dari seluruh dunia untuk menggunakan akseleratornya, namun Berners menemukan bahwa mereka punya masalah/kesulitan soal bagaimana untuk berbagi informasi satu sama lain.

Kemudian pada bulan Oktober 1990, Berners menulis tiga teknologi dasar hingga kini menjadi dasar web dibuat (yang juga muncul pada bagian-bagian browser web Anda). Ketiganya yaitu:

- HTML: *HyperText Markup Language*. HTML adalah bahasa format buat web.
- URI: *Uniform Resource Identifier*. Semacam “alamat” yang unik. Fungsinya untuk mengidentifikasi ke setiap sumber daya yang ada di web. Kini lebih awam disebut URL.
- HTTP: *Hypertext Transfer Protocol*. Memungkinkan Anda untuk mengambil kembali sumber daya yang terhubung dari seluruh web.

Berners juga menulis halaman web *editor/browser* pertama (“*WorldWideWeb.app*”) serta *server web* pertama (“*httpd*”). Pada akhir tahun 1990, halaman web pertama akhirnya disediakan di internet secara terbuka. Kemudian pada tahun 1991, orang-orang di luar CERN termasuk orang awam juga diundang untuk bergabung dengan komunitas web baru ini (dewaweb,2018) .

2.8 Node.js

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang

berjalan di sisi *client / browser* saja, maka Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan *server web* tanpa menggunakan program *server web* seperti Apache atau Nginx (codepolitan, 2017).

2.9 TensorFlow

Tensorflow merupakan *open source framework* yang dapat digunakan untuk mengembangkan, melatih, dan menggunakan model deteksi objek. Sistem ini sudah banyak diterapkan pada berbagai produk google anatar lain pencarian image, deteksi wajah, dan plat nomor kendaraan pada *google street view*, *Google assistant*, *way mo* atau *self driving car*, dan lain-lain.

Tensorflow bekerja dengan computational untuk membuat model *machine learning*. Tensorflow menyediakan berbagai *toolkit* yang memungkinkan anda membuar model pada tingkat yang lebih rendah untuk membuat model dengan menentukan serangkaian matematis. Sebagai alternatif, anda dapat menggunakan API dengan tingkat yang lebih rendah untuk membuat model dengan menentukan API dengan tingkat yang lebih tinggi (seperti *tf.estimator*) untuk menentukan arsitektur yang telah ditetapkan, seperti regresi linear atau *neural network*.

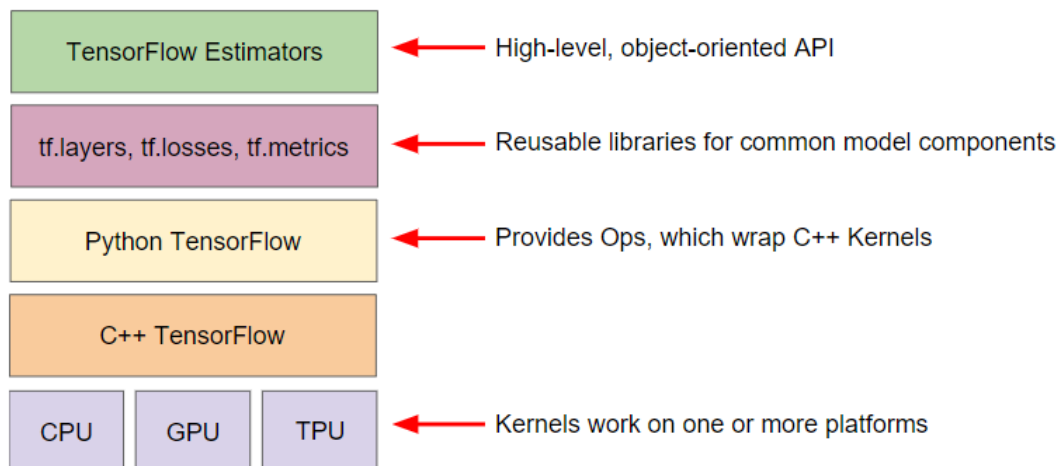
Framework tensorflow digunakan pada proses pembuatan sistem objek deteksi agar memudahkan implementasi algoritma dan penggunaan bahasa pemrograman kemudian terdapat GPU untuk mempercepat proses training.

Tensorflow sebagai kerangka *machine learning* yang dapat digunakan untuk mengolah banyak data atau ingin mempelajari kecedarasan buatan (*artificial Intelligence*) secara mendalam.

Tensorflow adalah *library open source* untuk perhitungan *numeric* berkinerja tinggi. Arsitekturnya yang fleksibel memungkinkan penyebaran komputasi dengan mudah di berbagai platform (CPU, GPU, TPU), dan dari desktop ke cluster server hingga perangkat seluler (Nisa Hanum Harani dan Miftahul Hasanah, 2020).

Awalnya dikembangkan oleh para peneliti dan insinyur dari tim google Brain dalam organisasi AI Google, ia hadir dengan dukungan kuat untuk machine learning dan deep learning dan inti perhitungan numerik yang fleksibel digunakan dibanyak domain ilmiah lainnya.

Gambar berikut menunjukkan hierarki toolkit tensorflow saat ini :



Gambar 2. 15 Tensorflow *Toolkit Hierarchy*

Tabel berikut berisi ringkasan tujuan dari berbagai lapisan :

Tabel 2.1 Hierarki *Toolkit* Tensorflow

Toolkit	Deskripsi
Estimator(<i>tf.estimator</i>)	Merupakan sebuah API tingkat tinggi yang berorientasi pada objek
<i>tf.layer/tf.losses/tf.metrics</i>	<i>Library</i> untuk komponen model umum
Tensorflow	API dengan tingkat lebih rendah

Menurut Lu Yifei (2017) terdapat 4 komponen *computational graph* terkait tensorflow yaitu:

- *Operations*

Dalam Tensorflow, node mewakili operasi. Node menggambarkan bagaimana data input mengalir melalui node dalam grafik yang diarahkan. Suatu operasi dapat memperoleh nol atau banyak input kemudian menghasilkan nol atau banyak output. Operasi semacam itu dapat berupa persamaan matematika, konstanta atau variabel. Konstanta diperoleh dengan operasi tanpa input dan menghasilkan output sama dengan konstanta yang sesuai. Demikian pula, variabel adalah operasi yang tidak mengambil input dan menghasilkan nilai saat ini dari variabel itu. Setiap operasi perlu diimplementasikan oleh kernelnya yang dapat dieksekusi pada perangkat keras seperti CPU atau GPU.

- *Tensors*

Dalam tensorflow, data diwakili oleh tensor yang mengalir antara node dalam grafik komputasi. Tensor adalah array multi dimensi dengan tipe statis

dan dimensi dinamis. Jumlah dimensi suatu tensor disebut pangkatnya. Bentuk tensor menggambarkan jumlah komponen di setiap dimensi. Dalam grafik komputasi, saat membuat operasi, tensor dikembalikan yang akan dikirim oleh tepi terarah sebagai input ke operasi yang terhubung.

- *Variables*

Di seluruh evaluasi pelatihan, sebagian besar tensor tidak bertahan sedangkan kondisi model seperti bobot dan bias perlu dipertahankan. Karenanya variabel ditambahkan ke grafik komputasi sebagai operasi khusus. Variabel menghemat tensor yang disimpan secara terus-menerus dalam buffer memori. Nilai variabel dapat dimuat saat melatih dan mengevaluasi model. Saat membuat variabel, perlu diberikan tensor sebagai nilai awal pada saat eksekusi. Bentuk dan tipe data dari tensor itu secara otomatis menjadi bentuk dan tipe variabel.

Inisialisasi variabel harus dieksekusi sebelum pelatihan. Ini dapat dilakukan dengan menambahkan operasi untuk menginisialisasi semua variabel dan menjalankannya sebelum melatih jaringan.

- *Sessions*

Eksekusi operasi dan evaluasi tensor dilakukan dalam konteks sesi. Sesi menggunakan Run rutin sebagai entri untuk mengeksekusi grafik komputasi. Dengan invocation run, input dimasukkan ke dalam proses komputasi seluruh grafik untuk mengembalikan output sesuai dengan definisi grafik. Grafik perhitungan akan dieksekusi berulang kali untuk melatih jaringan dengan menjalankan Run rutin. Sesi ini mendistribusikan operasi grafik ke perangkat

seperti CPU atau GPU pada mesin yang berbeda sesuai dengan algoritma penempatan tensorflow yang akan disajikan nanti. Selain itu, urutan eksekusi node didefinisikan secara eksplisit, yaitu dependensi kontrol. Mengevaluasi model memastikan bahwa dependensi kontrol dipertahankan.

Tensorflow memiliki kelebihan antara lain:

- Cepat

Performa merupakan faktor penting dalam mengembangkan dan menerapkan sistem machine learning. Karena itulah TensorFlow menggunakan XLA, pengkompilasi aljabar linear canggih yang membuat kode TensorFlow mampu berjalan secepat mungkin pada prosesor, CPU, GPU, TPU, dan platform hardware lain yang digunakan.

- Fleksibel

Tensorflow menyediakan API level tinggi yang memudahkan dalam mengembangkan dan melatih model, serta kontrol level rendah untuk mendapatkan fleksibilitas dan performa yang maksimal.

- Siap Produksi

Lingkup penggunaan tensorflow meliputi riset penyelidikan hingga produksi skala besar. Gunakan API tensorflow yang sama dan telah dipahami, baik untuk membuat jenis model baru maupun memproses jutaan permintaan dalam produksi.

Adapun cara tensorflow bekerja sebagai berikut :

- *Import or generate datasets*

Data merupakan hal yang paling penting dalam *machine learning* dengan adanya data makanya *machine learning* akan berjalan dengan baik. pada penelitian ini data yang digunakan diambil sendiri oleh peneliti.

- *Transform and normalize data*

Biasanya, dataset yang di input tidak memiliki bentuk yang dapat dibaca TensorFlow, jadi kita perlu mengubah dataset yang di input agar dapat dibaca dengan TensorFlow. *Transform and normalize data* berguna agar data input dapat langsung dibaca oleh tensorflow.

- *Partition datasets into train, and validation sets*

Untuk menguji data yang dimiliki kita perlu membagi data menjadi beberapa bagian, dalam penelitian peneliti membagi data menjadi dua bagian yaitu *training* dan *validation*. Data *training* merupakan data yang akan digunakan untuk melakukan proses *training*. Data *validation* digunakan untuk mengvalidasi hasil data yang sudah di *training*.

- *Set algorithm parameters (hyperparameters)*

Algoritma yang digunakan dalam *training* dan *validation* data adalah *convolutional neural network*. Penentuan parameter sangat penting demi memberikan hasil akurasi yang baik. Parameter yang maksud antara lain, epochs, jumlah layer konvolusi dan masih banyak lagi.

- *Initialize variables and placeholders*

Kita perlu menginisialisasi variabel dan *placeholder* ini dengan ukuran dan jenis yang sesuai, sehingga TensorFlow tahu apa yang diharapkan. TensorFlow juga perlu mengetahui jenis data yang diharapkan, sehingga dapat bekerja sesuai dengan apa yang kita harapkan.

- *Define the model structure*

Setelah kita memiliki data, dan menginisialisasi variabel dan *placeholder* kita, kita harus mendefinisikan modelnya. Ini dilakukan dengan membuat grafik komputasi. TensorFlow memilih operasi dan nilai apa yang harus menjadi variabel dan placeholder untuk sampai pada hasil model kita.

- *Declare the loss functions*

Setelah menentukan model, kita harus bisa mengevaluasi keluarannya. Di sinilah kami mendeklarasikan *loss function*. *Loss function* sangat penting karena memberi tahu kita seberapa jauh prediksi kita dari nilai sebenarnya.

- *Initialize and train the model*

Sekarang setelah kita memiliki segalanya, kita perlu membuat contoh grafik kita, memasukkan data melalui *placeholder*, dan membiarkan TensorFlow mengubah variabel untuk memprediksi data pelatihan kita dengan lebih baik.

- *Evaluate the model*

Setelah membuat dan melatih model, langkah selanjutnya adalah mengevaluasi model dengan melihat seberapa baik performanya dengan data baru melalui beberapa kriteria yang ditentukan.

- *Tune hyperparameters*

Sering kali, kami ingin kembali dan mengubah beberapa parameter sehingga kinerja model dapat sesuai dengan kita harapkan. Kami kemudian mengulangi langkah sebelumnya dengan parameter berbeda dan mengevaluasi model pada set validasi.

- *Deploy/predict new outcomes*

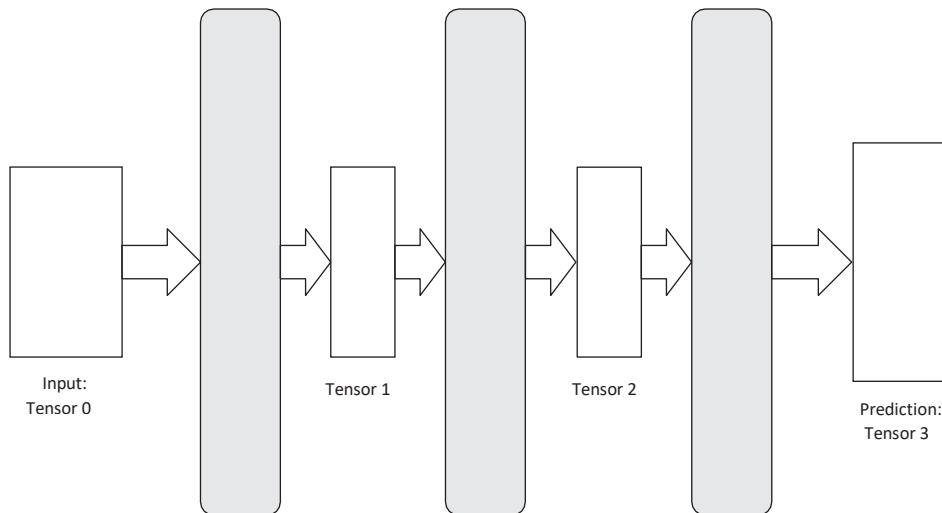
Penting juga untuk mencoba pada data baru sehingga menghasilkan data prediksi yang baru.

TensorFlow dibuat *open source* pada november 2015 oleh tim insinyur yang sedang mengerjakan *deep learning* di Google. TensorFlow telah mendapatkan popularitas luar biasa hingga saat ini sudah banyak digunakan untuk berbagai aplikasi industri dan proyek penelitian baik di Google dan di komunitas teknis yang lebih besar. Nama "TensorFlow" diciptakan untuk mencerminkan apa yang terjadi pada tipikal program yang ditulis dengan *framework*: representasi data yang disebut aliran tensor melalui lapisan dan node pemrosesan data lainnya, yang memungkinkan inferensi dan pelatihan terjadi pada model *machine learning*.

Dalam *neural network* dan *deep learning*, setiap bagian data dan setiap hasil perhitungan direpresentasikan sebagai tensor. Tensor sendiri di artikan oleh beberapa ilmuwan sains computer sebagai "array multidemensi". Misalnya, gambar skala abu-abu dapat direpresentasikan array angka 2D sebagai tensor 2D; gambar warna biasanya diwakili sebagai tensor 3D, dengan dimensi ekstra menjadi saluran warna. Suara, video, teks, dan tipe data lainnya semuanya dapat direpresentasikan sebagai tensor. Setiap tensor memiliki dua properti dasar: tipe

data (seperti float32 atau int32) dan bentuk. Bentuk menggambarkan ukuran tensor sepanjang semua dimensinya. Sebagai contoh, tensor 2D mungkin memiliki bentuk [128, 256], dan tensor 3D mungkin memiliki bentuk [10, 20, 128]. Setelah data berubah menjadi tensor dari tipe dan bentuk data yang diberikan, itu dapat dimasukkan ke dalam semua jenis lapisan yang menerima tipe dan bentuk data itu, terlepas dari makna asli data tersebut. Oleh karena itu, tensor adalah lingua franca dari model *deep learning*. Tensor adalah wadah yang mengatur data ke dalam struktur yang dapat diproses secara efisien secara paralel. Ketika kita menambahkan tensor A dengan bentuk [128, 128] ke tensor B dengan bentuk [128, 128], sangat jelas bahwa ada $128 * 128$ penambahan independen yang perlu dilakukan.

TensorFlow, mengalir melalui grafik struktur data yang terdiri dari operasi matematika yang saling berhubungan yang disebut node. Seperti yang ditunjukkan pada Gambar 2.15, node dapat menjadi layer yang berhasil dalam jaringan saraf. Setiap node mengambil tensor sebagai input dan menghasilkan tensors sebagai output. "*Fluor tensor*" ditransformasikan menjadi berbagai bentuk dan

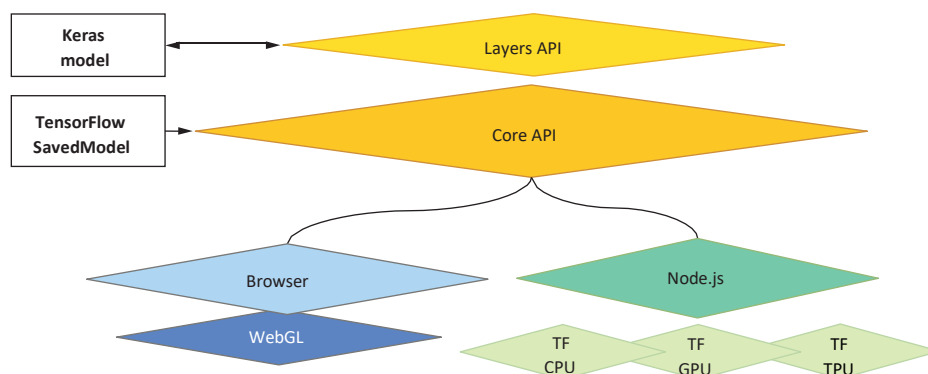


Gambar 2. 16 Tensor “flow” melalui sejumlah lapisan, skenario umum di TensorFlow dan TensorFlow.js

Dalam dunia TensorFlow, API tingkat tinggi yang disebut Keras. Keras menyediakan satu set jenis lapisan *neural network* yang paling sering digunakan, masing-masing dengan parameter yang dapat dikonfigurasi. Ini juga memungkinkan pengguna untuk menghubungkan lapisan bersama untuk membentuk *neural network*. Selain itu, Keras juga dilengkapi dengan API untuk :

- Menentukan bagaimana *neural network* akan dilatih (kehilangan fungsi, metrik, dan pengoptimal)
- diberikan data untuk dilatih atau mengevaluasi *neural network* atau menggunakan model untuk inferensi.
- Memantau proses pelatihan yang sedang berlangsung (*callbacks*)
- Menyimpan dan memuat model
- Mencetak atau memplot arsitektur model

Gambar 2.16 menyajikan gambaran umum arsitektur TensorFlow.js. Level terendah bertanggung jawab untuk komputasi paralel untuk operasi matematika cepat. Meskipun lapisan ini tidak terlihat oleh sebagian besar pengguna, sangat penting untuk memiliki kinerja tinggi sehingga pelatihan model dan inferensi di tingkat yang lebih tinggi dari API dapat secepat mungkin. Di browser, ia memanfaatkan WebGL untuk mencapai akselerasi GPU. Di Node.js, keduanya mengikat langsung ke paralelisasi CPU multicore dan akselerasi CUDA GPU. Node.js adalah *backend* matematika yang sama yang digunakan oleh TensorFlow dan Keras di Python. Dibangun di atas level matematika terendah adalah Ops API, yang memiliki paritas yang baik dengan API tingkat rendah dari TensorFlow dan mendukung pemuatan *SavedModels* dari TensorFlow. Pada tingkat tertinggi adalah API Layers Keras. Layers API adalah pilihan API yang tepat untuk sebagian besar programmer menggunakan TensorFlow.js. Layers API juga mendukung model dua arah yang mengimpor / mengekspor dengan Keras.



Gambar 2. 17 Arsitektur TensorFlow.js

Core API merupakan API yang langsung berhubungan dengan *low-level code* dan membuat dengan sedetail mungkin. Seperti membuat tensor dan scalar

secara manual, dan biasanya di gunakan ketika kita menginginkan untuk membuat model sendiri dengan kebutuhan spesial dan belum ada di dalam library layer API. Layer API merupakan library API yang yang di buat menggunakan Core Api dalam TensorFlow.js yang merupakan *high-level code*. Layer API di buat menjadi sederhana agar pemula dapat menggunakan *machine learning* dengan mudah. Layer API pada TensorFlow adalah KERAS dan biasanya di gunakan untuk hal-hal yang sudah biasa di lakukan di dalam *machine learning* seperti klasifikasi, regresi dan lain-lain.

TensorFlow.js adalah *library* yang memungkinkan pengguna melakukan *deep learning* dalam JavaScript. Seperti namanya, TensorFlow.js dirancang agar konsisten dan kompatibel dengan Tensor-Flow, framework python untuk *deep learning*. Untuk memahami TensorFlow.js, kita perlu memeriksa secara singkat sejarah TensorFlow.

TensorFlow.js bukan satu-satunya *library* Javascript untuk *deep learning*, dan juga bukan yang pertama kali muncul sebelumnya sudah ada brain.js dan ConvNetJS. Alasan mengapa TensorFlow.js sering digunakan dibandingkan dengan *library* yang lain adalah kelengkapannya. TensorFlow.js adalah satu-satunya perpustakaan yang tersedia saat ini yang mendukung semua bagian penting dari alur kerja dalam *deep learning*.

- Mendukung inferensi dan pelatihan
- Mendukung browser web dan Node.js
- Memanfaatkan akselerasi GPU (WebGL di browser dan kernel CUDA di Node.js)

- Mendukung definisi arsitektur model neural network dalam Javascript.
- Mendukung serialisasi dan deserialisasi model
- Mendukung konversi ke dan dari kerangka kerja *deep learning* python
- Kompatibel dalam API dengan kerangka kerja *deep learning* Python
- Dilengkapi dengan dukungan bawaan untuk konsumsi data dan dengan API untuk visualisasi.

Alasan kedua adalah ekosistem. Sebagian besar *library deep learning* JavaScript menentukan API unik mereka sendiri, sedangkan TensorFlow.js terintegrasi erat dengan TensorFlow dan Keras. Ketika memiliki model terlatih dari Python TensorFlow atau Keras dan pengguna dapat menggunakannya di browser. Anda telah membuat model TensorFlow.js di browser dan ingin membawanya ke Keras untuk akses ke akselerator yang lebih cepat seperti Google TPU maka itu bisa dilakukan. Integrasi yang ketat dengan kerangka kerja non-JavaScript tidak hanya meningkatkan interoperabilitas tetapi juga memudahkan pengembang untuk bermigrasi antara dunia bahasa pemrograman dan tumpukan infrastruktur(Shanqing, 2020).