

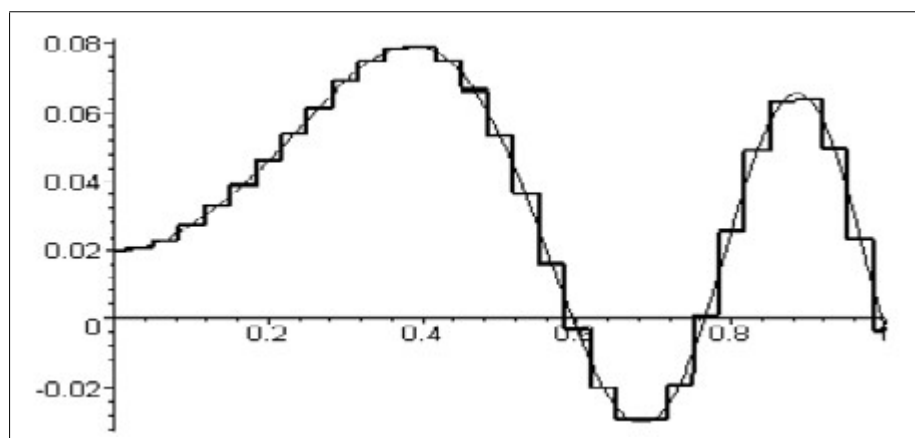
BAB II

TINJAUAN PUSTAKA

2.1 Teorema Penaksiran Universal

Teorema penaksiran universal atau *universal approximation theorem* menyatakan bahwa sebuah model jaringan *feed-forward* dapat membentuk fungsi apapun secara subjektif. Sebuah model jaringan saraf tiruan dibentuk dari serangkaian lapisan yang didalamnya terdapat deretan sel saraf atau *neuron* dengan kuantitas tertentu. Semakin panjang rangkaian lapisan yang tersedia, maka semakin banyak saraf yang tersedia sehingga dapat memetakan fungsi yang sulit. Model jaringan yang memiliki banyak saraf dapat mempelajari pola-pola yang ada dari satu domain ke domain lainnya [7].

Teorema penaksiran universal memiliki dua sifat yang dikategorikan berdasarkan pemanfaatannya dalam melakukan pemelajaran mesin. Sifat pertama adalah suatu model jaringan saraf tiruan dapat memperkirakan suatu fungsi dengan batasan-batasan tertentu sesuai dengan fungsi keluaran pada setiap lapisan yang terdapat dalam model khususnya lapisan yang terdapat pada bagian akhir. Sifat kedua adalah sebuah fungsi kontinu dengan jumlah variabel sembarang dapat ditiru sifatnya oleh sebuah jaringan saraf tiruan dengan jumlah parameter yang sembarang [13].

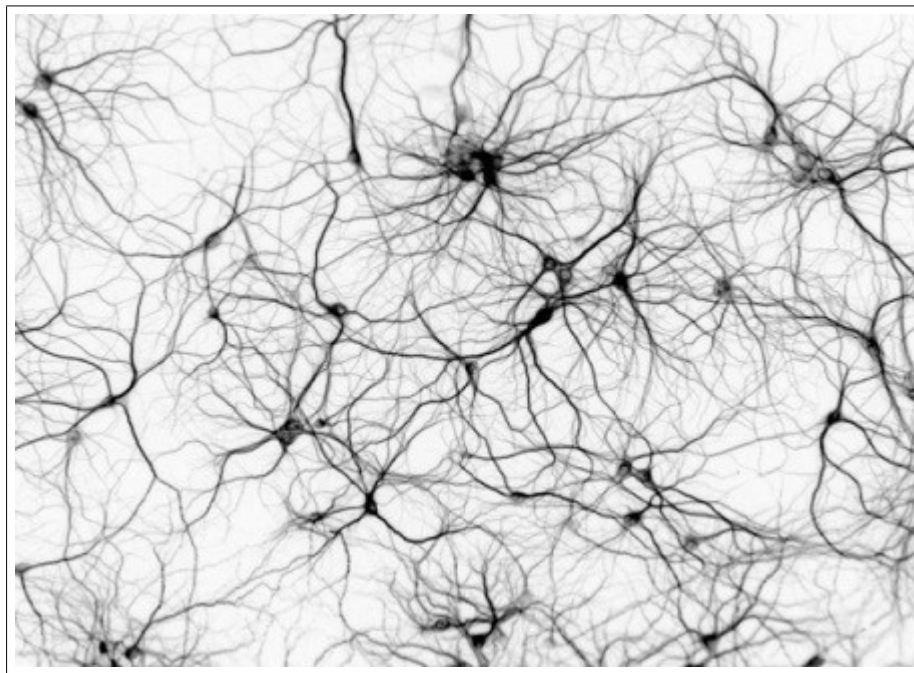


Gambar 2.1: Teorema Penaksiran Universal

Sumber: <https://encrypted-tbn0.gstatic.com/images?q=tbn>

2.2 Jaringan Saraf

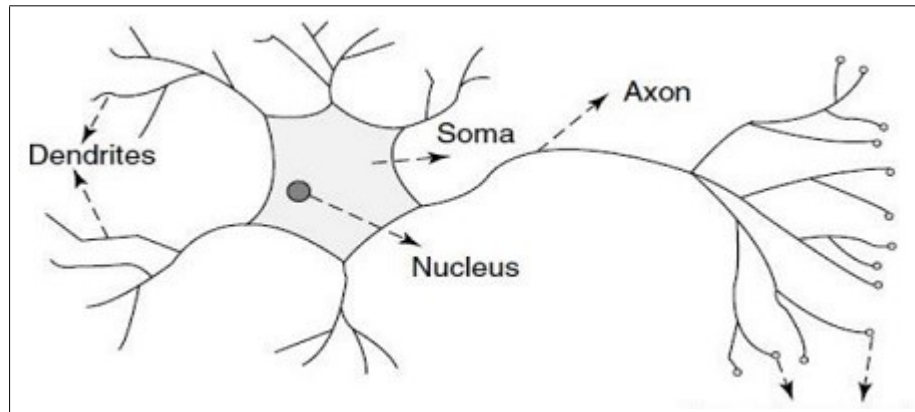
Otak manusia terdiri dari kumpulan sel saraf yang saling terkoneksi satu sama lain. Sebuah sel saraf adalah sel yang dapat memproses dan mengantarkan informasi apabila dirangsang dengan tegangan elektrokimia. Sel-sel saraf tidak membelah dirinya dan tidak digantikan apabila ada yang rusak. Jumlah sel saraf yang terdapat dalam otak manusia diperkirakan sebanyak satu miliar. Setiap sel saraf diperkirakan berkoneksi dengan sepuluh ribu sel saraf lainnya melalui sinapsis yang berarti otak manusia dewasa beroperasi seperti prosesor dengan kecepatan satu triliun bit per detik [9].



Gambar 2.2: Ilustrasi Jaringan Saraf Manusia

Sumber: <https://wccftech.com/scientists-artificial-neurons>

Bentuk sel saraf sangat bervariasi dengan berbagai ukuran, bentuk, dan sifat elektrokimianya. Sebuah sel saraf memiliki badan yang terdiri dari beberapa struktur penting meliputi *soma*, *dendrites*, *axon*, dan *synapses* seperti pada gambar 2.3. Sebuah sel saraf akan menerima beberapa masukan melalui *synapses*, memproses inputan tersebut melewati *dendrites*, kemudian diteruskan melalui *soma*, dan diberikan kepada sel saraf lainnya melalui *axon* [23].



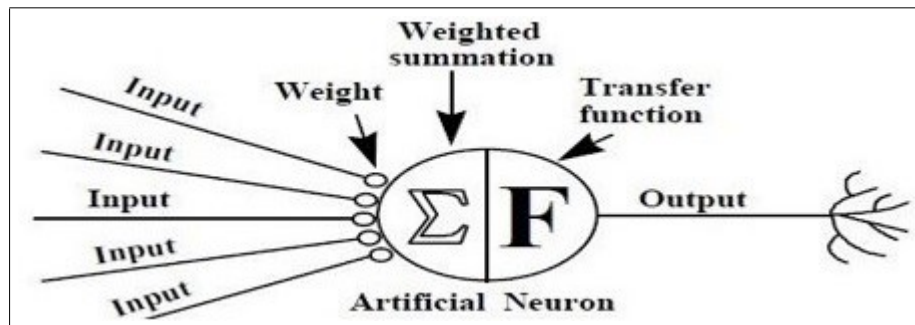
Gambar 2.3: Ilustrasi Sel Saraf Manusia

Sumber: <https://en.wikipedia.org/wiki/Neuron>

2.3 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan adalah sistem komputasi yang cara kerjanya menyerupai jaringan saraf pada otak makhluk hidup. Sebuah jaringan saraf tiruan dapat dengan mandiri memodelkan fungsi sembarang yang tingkat kesulitannya sesuai dengan jumlah koneksi yang tersedia. Jaringan ini menyerupai jaringan saraf asli dimana sebuah saraf tiruan menerima banyak masukan dari saraf lainnya kemudian dioperasikan dengan bobot yang terkandung pada sel tersebut dan akhirnya diteruskan ke sel berikutnya [20].

Sebuah sel saraf dapat dibagi menjadi empat bagian yang meliputi masukan, bobot, fungsi transfer atau aktivasi, dan keluaran. Jumlah masukan pada suatu sel saraf tiruan berjumlah sebanyak output dari sel-sel yang berada pada layer sebelumnya. Bobot sel adalah nilai numerik yang menjadi identitas dari sel yang merupakan hasil penyesuaian dari proses latihan. Fungsi aktivasi adalah sebuah fungsi menerima hasil operasi antara bobot dan masukan. Keluaran merupakan hasil dari sel yang diteruskan ke lapisan selanjutnya. Ilustrasi sebuah sel saraf tiruan dapat dilihat pada gambar 2.4 [20].



Gambar 2.4: Ilustrasi Sel Saraf Tiruan

Sumber: https://en.wikipedia.org/wiki/artificial_neural_network

2.4 Fungsi Aktivasi

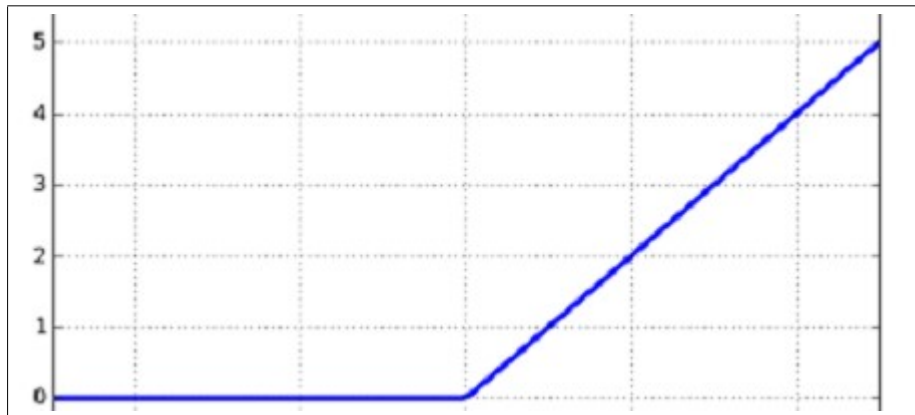
Fungsi aktivasi pada sel saraf tiruan berfungsi untuk mengkonversi hasil operasi matriks antara masukan dan bobot sebuah sel dari sistem linier menjadi sistem nonlinier. Konversi nilai ini dilakukan agar setiap sel mengambil perannya pada saat proses pelatihan model. Beberapa fungsi aktivasi yang dipakai pada umumnya adalah *Sigmoid*, *ReLU*, *Tanh*, dan *Softmax* [2].

Rectified Linear Unit adalah fungsi aktivasi yang paling umum digunakan dalam aplikasi jaringan saraf tiruan. Fungsi aktivasi *ReLU* membatasi nilai masukannya dimana nilai yang kurang dari nol akan diubah menjadi nol [10, 16]. Fungsi aktivasi *Rectified Linear Unit* dapat dilihat pada persamaan 2.1 dengan grafik yang dapat dilihat pada gambar 2.5.

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \quad (2.1)$$

2.5 Residual Network

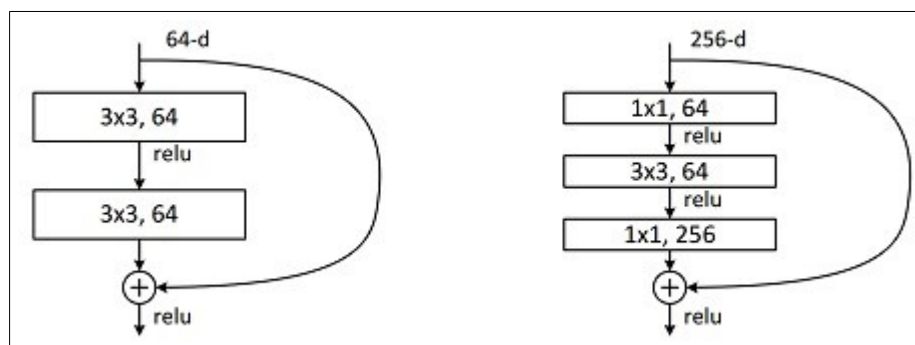
Residual Network atau *ResNet* merupakan model jaringan saraf tiruan yang menggunakan *residual block* sebagai dasar dari setiap lapisan. Sebuah *residual block* merupakan sebuah arsitektur jaringan saraf tiruan kecil yang terdiri dari beberapa lapisan. Setiap blok akan menjumlahkan masukan dan keluarannya sehingga layer di dalam suatu blok hanya menambahkan pola-pola yang dipelajari.



Gambar 2.5: Rectified Linear Unit

Sumber: https://static.packt-cdn.com/products/graphics/B05478_03_11.png

Hal ini memungkinkan *ResNet* untuk memiliki jumlah blok yang sangat banyak sehingga dapat memetakan suatu fungsi sembarang yang sulit sesuai dengan teorema penaksiran universal [8]. Jenis-jenis *residual block* dapat dilihat pada gambar 2.6.



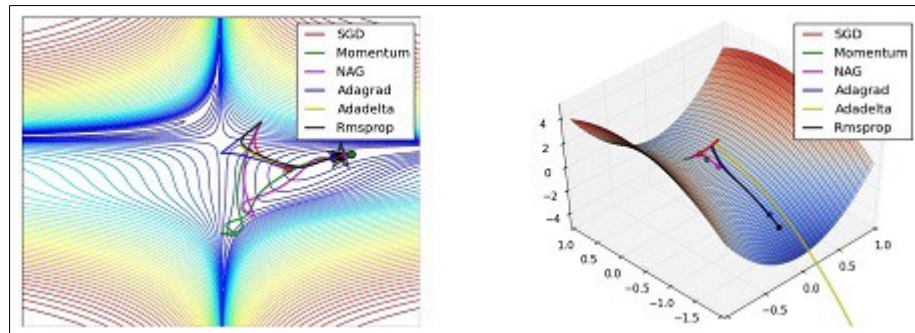
Gambar 2.6: Residual Block

Sumber: <https://arxiv.org/abs/1512.03385>

2.6 Optimisasi Model

Algoritma optimisasi model yang paling populer dalam melakukan pembelajaran model *deep learning* adalah *gradient descent*. *Gradient descent* meminimalisir selisih antara prediksi model dan target sebenarnya dengan merubah bobot-bobot yang terdapat dalam model. Nilai bobot yang ditambahkan berbanding terbalik dengan gradien hasil fungsi kesalahan terhadap masing-masing bobot. Proses optimisasi dilakukan dengan melakukan *backpropagation* yang

melibatkan beberapa elemen seperti *learning rate*, dan *loss function*. Tujuannya adalah untuk mencapai titik optimal pada sebuah bidang berdasarkan hasil dari *loss function* [7]. Ilustrasi titik optimal digambarkan pada gambar 2.7.

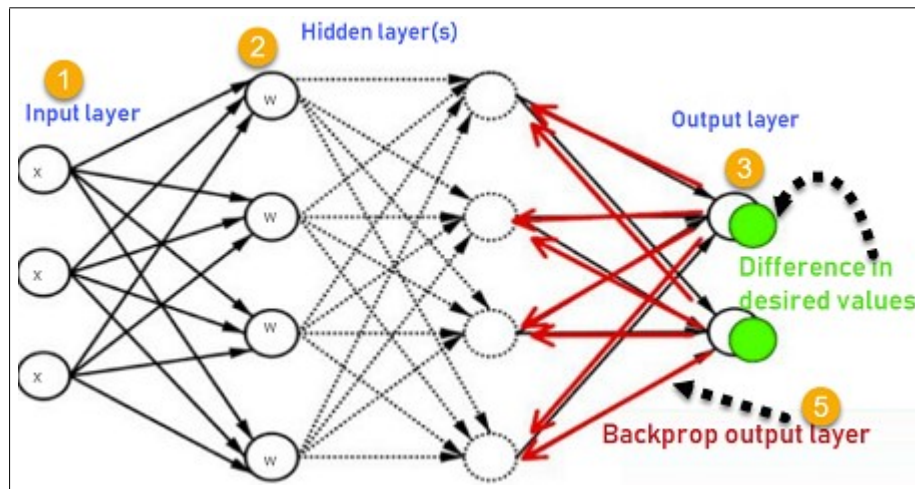


Gambar 2.7: Gradient Descent

Sumber: <https://arxiv.org/abs/1601.00013>

2.6.1 Backpropagation

Backpropagation adalah sebuah prosedur pembelajaran jaringan saraf tiruan yang secara berulang-ulang kali menyesuaikan bobot setiap sel hingga selisih antara keluaran dan target menjadi lebih kecil. Algoritma ini merupakan mengorganisasikan sebuah model jaringan untuk secara mandiri mencari titik optimal secara berkala. Optimisasi dilakukan dengan melakukan *forward-pass* pada satu atau sebagian atau semua data yang ada ke dalam model untuk mendapatkan hasil prediksi. Hasil tersebut kemudian diukur selisihnya dengan target yang sebenarnya menggunakan sebuah fungsi kesalahan. Turunan parsial masing-masing bobot terhadap selisih kesalahan ini adalah kuantitas negatif yang harus ditambahkan bobot yang berkaitan [19]. Skema algoritma *backpropagation* dapat dilihat pada gambar 2.8.

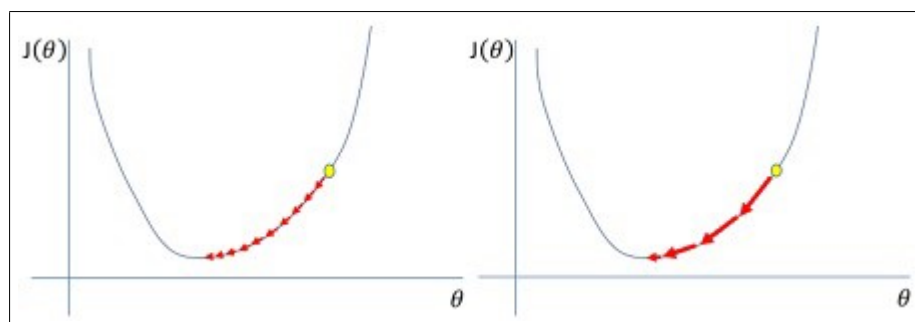


Gambar 2.8: Skema *Backpropagation*

Sumber: <https://www.nature.com/articles/323533a0>

2.6.2 *Learning Rate*

Learning rate adalah sebuah nilai skalar yang menentukan seberapa besar sebuah bobot pada jaringan saraf akan ditambahkan. Nilai *learning rate* umumnya bernilai kecil karena besar turunan parsial pada setiap iterasi akan berubah-ubah. Nilai *learning rate* yang besar akan merubah bobot dengan skala yang besar sehingga akurasi model cenderung tidak stabil. Nilai *learning rate* yang kecil akan menghasilkan model yang stabil tetapi memerlukan jumlah iterasi yang banyak seperti pada gambar 2.9 [22].



Gambar 2.9: Ilustrasi Perbedaan *Learning Rate*

Sumber: <https://arxiv.org/abs/1908.01878>

2.6.3 Mean Squared Error

Mean squared error adalah sebuah persamaan estimasi kesalahan dengan merata-ratakan hasil dari pangkat dua selisih antara dua vektor. Fungsi kesalahan ini dipakai untuk mengukur kesalahan model regresi dimana keluaran yang diukur bersifat kontinu. Persamaan ini akan selalu bernilai positif dan nilai yang mendekati nol menandakan bahwa selisih antara dua vektor semakin kecil seperti pada persamaan 2.2 [21].

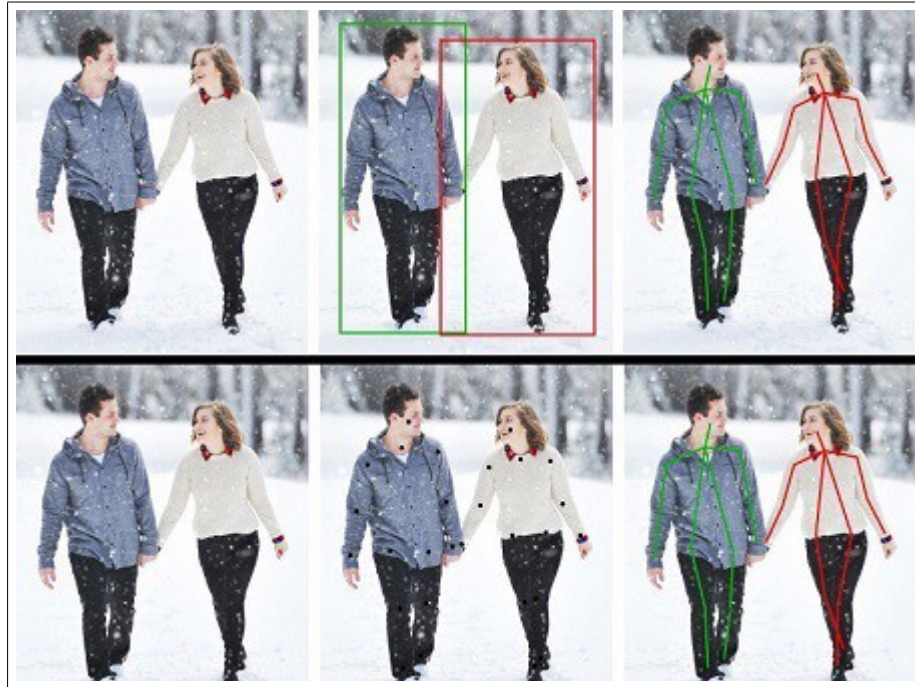
$$MSE = \frac{1}{n} \sum_{i=1}^n \left(\frac{d_i - f_i}{\sigma_i} \right)^2 \quad (2.2)$$

2.7 Estimasi Pose Dua Dimensi

Estimasi pose dua dimensi melakukan lokalisasi titik kunci anatomi atau bagian tubuh manusia berdasarkan fungsi dari bagian tersebut pada sebuah gambar dua dimensi. Setiap gambar unik dapat berisi jumlah orang yang berbeda dan dapat muncul dengan posisi dan ukuran yang berbeda-beda. Interaksi antara manusia dengan benda disekitarnya dapat menyebabkan oklusi dan hilangnya titik kunci anatomi yang dicari. Kompleksitas permasalahan ini juga semakin meningkat berbanding lurus dengan jumlah orang yang ada dalam suatu gambar [5] [6].

Pendekatan yang umum dilakukan adalah dengan mendeteksi setiap orang dalam setiap gambar secara individu. Hal pertama yang dideteksi adalah lokalisasi kemunculan suatu benda yang dianggap sebagai objek manusia dan kemudian mencari titik kunci anatomi dari objek tersebut. Pendekatan *top-down* seperti ini cocok dipakai untuk mendapatkan *single-person* dengan mudah tetapi kompleksitas yang proporsional dengan jumlah orang. Sebaliknya dengan pendekatan *bottom-up* dimana deteksi dilakukan dengan mencari piksel pada gambar yang dianggap adalah titik kunci anatomi dan dilanjutkan dengan membangun pose manusia dari kandidat yang ada. Pendekatan *bottom-up* cocok untuk menyelesaikan masalah *multi-person* dikarenakan pencarian kandidat titik kunci dapat dilakukan terlebih dahulu tanpa harus mengetahui jumlah orang.

Perbedaan pendekatan *top-down* dan *bottom-up* dapat dilihat pada gambar 2.10 [3].



Gambar 2.10: Pendekatan *Top-Down* dan *Bottom-Up*

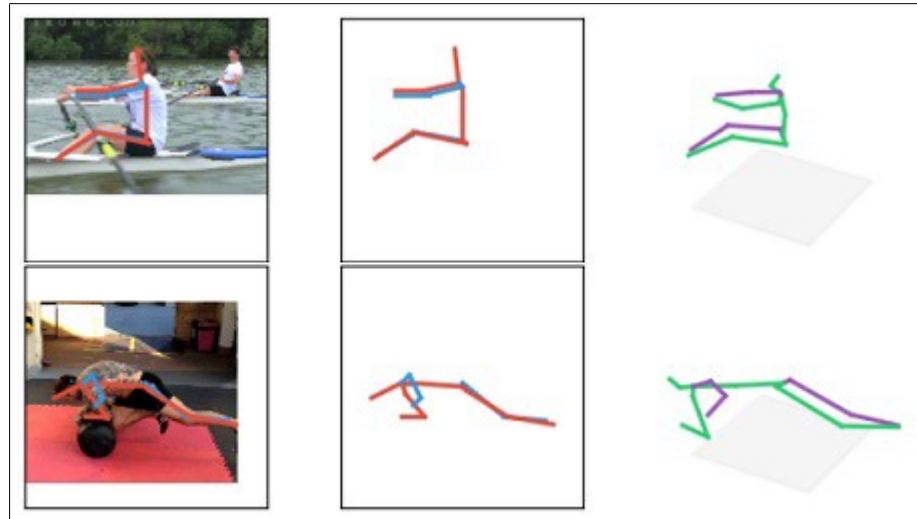
Sumber: <https://mc.ai/an-overview-of-human-pose-estimation-with-deep-learning>

2.8 Estimasi Pose Tiga Dimensi

Algoritma estimasi pose tiga dimensi merekonstruksi titik kunci tiga dimensi tubuh manusia dari sebuah gambar dua dimensi. Proses pencarian titik kunci dibagi menjadi dua golongan yaitu pencarian secara langsung menggunakan gambar sebagai masukan dan pencarian bertahap dengan mencari titik kunci anatomi dua dimensi terlebih dahulu kemudian dilanjutkan dengan pencarian titik kunci tiga dimensi. Titik kunci yang didapatkan dapat berada dalam koordinat lokal dan global. Pencarian secara langsung tidak menghasilkan akurasi yang tinggi dan cenderung buruk dalam mendeteksi orientasi yang benar [4].

Titik kunci tiga dimensi yang berada dalam koordinat lokal menjadikan posisi pinggang sebagai titik nol yang berarti lokasi pinggang akan selalu bernilai $\vec{hip} = [0.0, 0.0, 0.0]$. Posisi titik kunci lain relatif terhadap titik pinggang. Hal ini menyebabkan pose yang ditangkap hanya bersifat lokal yang berarti hanya

memiliki satu orientasi lokal. Pencarian pose tiga dimensi lokal dari gambar dua dimensi diilustrasikan pada gambar 2.11 [14].



Gambar 2.11: Pencarian Pose Tiga Dimensi Lokal

Sumber: <https://arxiv.org/abs/1705.03098>

Titik kunci tiga dimensi dapat juga berada dalam koordinat global yang berarti orientasi kamera saat mengambil gambar diperhitungkan sebagai sebuah observasi virtual. Kompleksitas yang dihadapi berkaitan dengan orientasi kamera yang berubah. Estimasi pose tiga dimensi global umumnya dilakukan dengan mendeteksi titik kunci anatomi tubuh manusia ke dalam suatu dunia virtual tiga dimensi sehingga dapat diamati dari berbagai sudut yang berkorelasi dengan kamera [24, 15]. Pencarian pose tiga dimensi global dari gambar dua dimensi diilustrasikan pada gambar 2.12.

2.9 PyTorch

PyTorch merupakan *framework* untuk melakukan *deep learning* dengan bahasa pemrograman python. Penggigit data umumnya menggunakan bahasa pemrograman ini dalam melakukan riset pemanfaatan dan pengolahan data. PyTorch menyediakan implementasi grafik jaringan saraf tiruan yang bersifat dinamis. Hal ini memungkinkan pengguna untuk mengubah arsitektur model secara cepat dalam setiap iterasi [17].



Gambar 2.12: Pencarian Pose Tiga Dimensi Global

Sumber: <http://gvv.mpi-inf.mpg.de/projects/SingleShotMultiPerson>



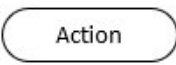

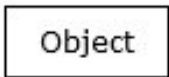
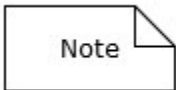
2.10 *Unified Modeling Language*

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang memiliki sintaks dan semantik. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. UML diaplikasikan dengan beberapa maksud seperti percangan perangkat lunak, sarana komunikasi antara perangkat lunak dengan proses bisnis, penjabaran sistem secara rinci dan pendokumentasian sistem yang ada.

Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para pengembang aplikasi untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem. UML merupakan Alat komunikasi yang konsisten dalam mendukung para pengembang saat ini. UML menyediakan sembilan jenis diagram yaitu : *Class Diagram*, *Packet Diagram*, *Use Case Diagram*, *Sequence Diagram*, *Communication Diagram*, *Statechart Diagram*, *Activity Diagram*, *Component Diagram* dan *Deployment Diagram*.

Activity Diagram merupakan penggabungan dari berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Tabel 2.1: Simbol-Simbol *Activity Diagram*

No	Nama	Gambar	Keterangan
1.		<i>Initial</i>	Bagaimana objek dibentuk atau diawali.
2.		<i>Final</i>	Bagaimana objek dihancurkan.
3.		<i>Action</i>	State eksekusi dari suatu aksi.
4.		<i>Fork</i>	Penggabungan atau pemisahan alur.
5.		<i>Object</i>	State hasil dari suatu aksi.
6.		<i>Note</i>	Keterangan tambahan suatu komponen.