

BAB II

LANDASAN TEORI

II.1 *Organizational Agility*

Organization agility adalah organisasi atau perusahaan yang memiliki kemampuan untuk berespon dan beradaptasi dengan cepat terhadap keadaan yang berubah. Dengan menjadi *Organization agility*, perusahaan dapat menggabungkan kecepatan dan stabilitas dalam bekerja. Perusahaan yang termasuk AO juga akan bekerja dengan waktu yang efektif dan tetap konsisten, walaupun sedang berada dalam situasi yang tidak pasti. Selain itu, *agility* atau ketangkasan dapat membantu memperjelas peran, inovasi, dan disiplin operasional. Pada akhirnya, itu semua akan memberikan hasil yang positif untuk kesehatan dan kinerja organisasi. Budaya organisasi merupakan pondasi dasar untuk membangun agilitas organisasi. budaya organisasi mempengaruhi seluruh aspek organisasi, akan memberdayakan organisasi berdasarkan keyakinan dan nilai-nilai bersama, termasuk agilitas (Raharso 2018).

Organizational Agility adalah kemampuan perusahaan untuk cepat berubah atau beradaptasi dalam menanggapi perubahan di pasar. Tingkat ketangkasan organisasi yang tinggi dapat membantu perusahaan untuk bereaksi dengan sukses terhadap munculnya pesaing baru, pengembangan teknologi baru yang mengubah industri, atau perubahan mendadak dalam kondisi pasar secara keseluruhan. McKinsey mendefinisikan kelincahan sebagai “kemampuan organisasi untuk memperbarui dirinya sendiri, beradaptasi, berubah dengan cepat, dan berhasil dalam lingkungan yang berubah cepat, ambigu, bergolak”. Kelincahan tidak bertentangan dengan stabilitas justru sebaliknya. Kelincahan sangat dibutuhkan untuk stabilitas bagi sebagian besar

perusahaan. Dalam instansi yang sangat kecil, stabilitas biasanya diwujudkan dalam diri pendiri, dan beberapa orang di sekitar pendiri. Memulai dari garasi seseorang bisa cepat dan gesit tanpa perlu banyak stabilitas. Tapi begitu organisasi menjadi besar, tidak bisa gesit. Sehingga perusahaan yang sudah mapan sering berjuang untuk menjadi lebih dinamis dan harus dapat berkembang dalam lingkungan yang menuntut perubahan konstan.

Berdasarkan survei hanya 27% responden yang menganggap perusahaannya gesit. Dan budaya dianggap sebagai salah satu rintangan terbesar untuk beradaptasi dengan tingkat kelincahan yang lebih tinggi, dengan 50% responden menyebutnya sebagai tantangan. Budaya praktik dan proses perusahaan yang sudah mengakar dalam dapat mencegah karyawan dari menawarkan ide-ide baru dan menantang.

Definisi operasional merupakan suatu petunjuk bagaimana suatu variable diukur, untuk mengetahui baik buruknya pengukuran dari suatu penelitian. Menurut sugiyono (2018) adalah penentuan konstrak atau sifat yang di pelajari sehingga menjadi variable yang dapat diukur. adapun yang menjadi definisi operasional dalam penelitian ini meliputi :

Tabel II.1 Definisi *operational organizational agility*

No.	Variabel	Definisi	Indikator
1.	<i>Leadership</i>	<i>Leadership</i> merupakan sebagai kemampuan seseorang untuk mempengaruhi orang lain agar mencapai tujuan yang telah dicanangkan	1. Energi Tinggi 2. Percaya Diri 3. Integritas Stabilitas 4. Fleksibilitas 5. Kepekaan

			Terhadap Orang Lain
2	<i>Workforce Agility</i>	<i>Workforce agility</i> merupakan kemampuan karyawan untuk berkontribusi dalam lingkungan yang merubah dengan cepat	1. Berurusan 2. Pemecahan Masalah 3. Fleksibilitas Profesional 4. Mempelajari Tugas dan Prosedur Kerja 5. Daptabilitas Interpersonal 6. Mengatasi Stres Kerja

II.2 *Leadership* dan *workforce agility*

Di masa yang terus berkembang dari waktu ke waktu dengan cepat, sebagai karyawan pasti selalu dituntut untuk bisa beradaptasi dengan cepat dan bisa mengikuti perubahan yang terus terjadi. Ada berbagai pengertian tentang *agility*. Pada intinya *agility* bisa membuat karyawan lebih berkembang dan memiliki kemampuan pemecahan masalah yang lebih kompleks. Untuk menjadi individu atau karyawan yang agile menurut Sherehiy dan Karwowski (2014) karyawan atau individu harus bersifat proaktif, mampu beradaptasi, dan memiliki resiliensi tinggi. Employee *agility* bisa

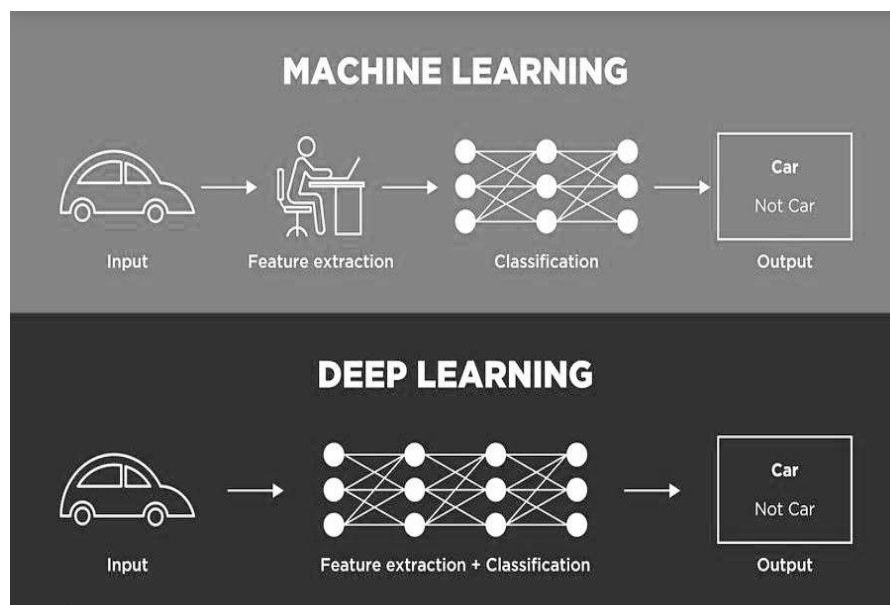
membuat perusahaan akan lebih sukses di masa depan karena sumber daya manusia yang berkualitas tinggi (Saptarini and Mustika 2023).

Keterlibatan karyawan dalam setiap proses yang terdapat di perusahaan sangatlah penting. Karyawan merupakan sumber daya manusia perusahaan. Kesuksesan perusahaan juga dinilai dari aspek sumber daya manusianya. Salah satunya adalah agility karyawan. Ketika karyawan mempunyai agility yang baik, itu akan membawa dampak yang besar bagi perusahaan. Perusahaan bisa menghemat biaya dan membuat perubahan yang efektif kedepannya. Karena dengan *agility* karyawan, ide-ide baru dan inovatif akan terus bermunculan. Di setiap masalah yang ada di perusahaan, harus dipertimbangkan dengan baik, maka dari itu perusahaan memerlukan karyawan yang mempunyai *agility* yang baik.

II.3 *Deep learning*

Deep Learning adalah salah satu fungsi kecerdasan buatan yang meniru cara kerja otak manusia dalam memproses data dan menciptakan pola untuk digunakan dalam pengambilan keputusan. *Deep Learning* merupakan bagian dari *Machine Learning*. *Deep Learning* terdiri dari banyak kelas atau jenis, beberapa yang paling sering digunakan adalah *Convolutional Neural Network* (CNN), *Multilayer Perceptrons* (MLP), dan *Recurrent Neural Network* (RNN). Keunggulan *deep learning* adalah memiliki performa terbaik dalam menyelesaikan masalah kompleks, mengurangi kebutuhan untuk rekayasa ciri, dan memiliki arsitektur yang mampu beradaptasi terhadap permasalahan baru dengan mudah. Sementara, kekurangan dari *deep learning* adalah membutuhkan banyak data, proses training yang memakan waktu, dan *overfitting*.

Deep Learning merupakan sub-bidang dari *Machine Learning* yang bekerja dengan melakukan representasi dari data pada lapisan pembelajaran yang berlapis-lapis hingga representasi menjadi lebih bermakna. *Deep* pada *Deep Learning* memiliki makna bahwa *Deep Learning* menggagas lapisan representasi yang berurutan. Sekarang ini, *Deep Learning* biasanya memiliki lapisan dengan jumlah puluhan hingga ratusan lapisan yang berurutan dan lapisan-lapisan tersebut dapat secara otomatis mempelajari data training yang diberikan. Pada *Deep Learning* lapisan representasi tersebut bernama *Neural Networks*. *Neural Networks* memiliki struktur yang bertumpuk yang berarti suatu lapisan akan berada diatas lapisan lain. *Neural Networks* mengambil konsep dari bidang *neurobiology* dan terinspirasi dari kemampuan dalam memahami sesuatu seperti yang dilakukan oleh otak manusia. Meskipun konsep utama *Deep Learning* dikembangkan atas inspirasi tersebut, tetapi model *Deep Learning* bukanlah model dari otak manusia. Hal tersebut dikarenakan belum ada bukti yang menunjukkan bahwa otak manusia bekerja layaknya model *deep learning* seperti sekarang ini.



Gambar II.1 Perbandingan *Deep Learning*

II.3.1 Jenis-Jenis Algoritma *Deep Learning*

Deep learning adalah sebuah teknologi yang mampu bekerja menggunakan beberapa algoritma tertentu. Sejatinya, tidak ada algoritma *deep learning* yang dianggap sempurna. Sebab, masing-masing jenis memiliki kapabilitas yang berbeda. Maka dari itu, *application developer* harus memilih jenis algoritma yang paling sesuai dengan kebutuhan mereka. Maka dari itu, untuk memilih algoritma yang tepat, ada baiknya kamu pahami masing-masing jenis algoritma *deep learning*.

1. *Convolutional neural networks (CNN)*
2. *Long short term memory network (LSTM)*
3. *Reccurent neural network (RNN)*
4. *Self organizing maps (SOM)*

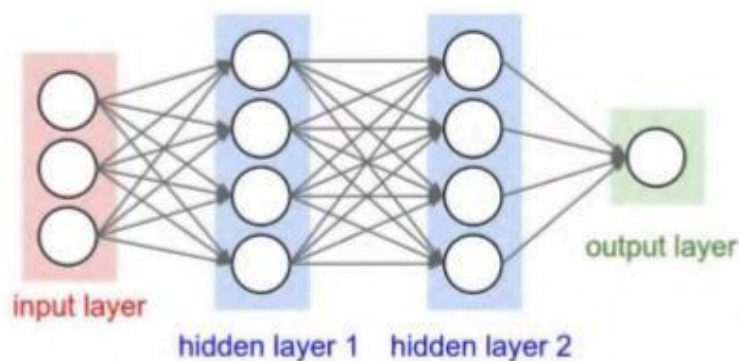
II.3.2 Manfaat *Deep Learning*

Deep learning adalah metode dalam kecerdasan buatan (AI) yang mengajarkan komputer untuk memproses data dengan cara yang terinspirasi otak manusia. Model *deep learning* dapat mengenali pola kompleks dalam gambar, teks, suara, dan data lain untuk menghasilkan wawasan dan prediksi yang akurat. Berikut adalah manfaat teknologi *deep learning* :

1. Memaksimalkan kinerja *unstructured* data dalam aplikasi atau situs web
2. Menghilangkan kebutuhan teknologi untuk rekayasa fitur
3. Memberikan tampilan *output* yang jauh lebih berkualitas
4. Mengurangi biaya *operasional development*
5. Teknik manipulasi data yang efektif

II.4 Convolutional Neural Network

CNN adalah variasi dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. *Convolutional Neural Network* adalah salah satu algoritma *Deep Learning* yang merupakan pengembangan dari *Multilayer Peceptron* (MLP) yang dirancang untuk mengolah data dalam bentuk dua dimensi, misalnya gambar atau suara (Ilahiyah and Nilogiri 2018). CNN juga merupakan jenis dari *deep neural network* yang didesain untuk mengolah data dua dimensi dengan kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. CNN pertama kali dikembangkan dengan nama NeoCognitron oleh Kuniyihiko Fukushima, seorang peneliti dari NHK *Broadcasting Science Research Laboratories*, Kinuta, Setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Yann LeChun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeChun pada penelitiannya mengenai pengenalan angka dan tulisan tangan.



Gambar II.2 Arsitektur Sederhana *Multilayer Perceptron*

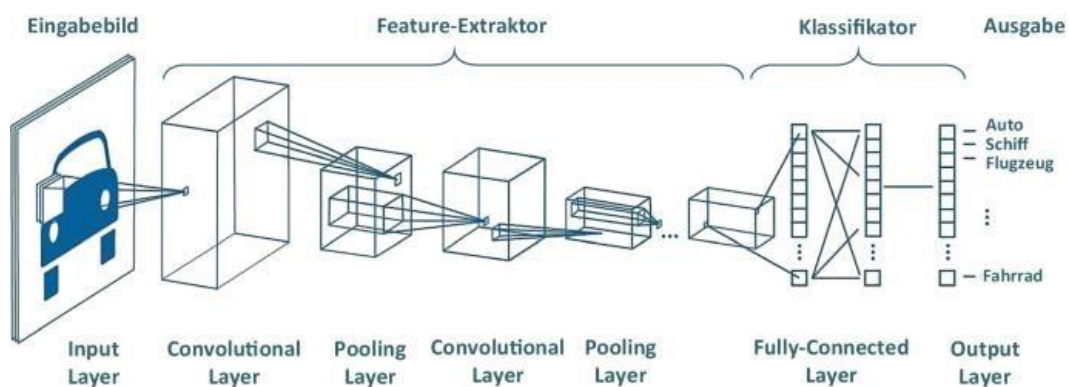
Sebuah MLP seperti pada Gambar II.1 memiliki i layer (kotak merah dan biru) dengan masing-masing layer berisi j neuron (lingkaran putih). MLP menerima input data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan output. Setiap hubungan antar neuron pada dua layer yang bersebelahan

memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Di setiap data input pada *layer* dilakukan operasi *linear* dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi *non-linear* yang disebut sebagai fungsi aktivasi. Pada CNN, data yang dipropagasikan oleh jaringan adalah data dua dimensi, sehingga operasi *linear* dan parameter bobot pada CNN berbeda. Pada CNN operasi *linear* menggunakan operasi konvolusi dan bobot berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi.

II.4.1 Arsitektur *Convolutional Neural Network*

Neural networks terdiri dari berbagai *layer* dan di setiap *layer* terdapat beberapa *neuron* (Stathakis, 2008). MLP memiliki keterbatasan untuk memetakan persamaan dengan data yang lebih kompleks. Pada permasalahan jumlah *hidden layer* dibawah tiga *layer*, terdapat pendekatan untuk menentukan jumlah *neuron* pada masing-masing *layer* untuk mendekati hasil optimal. Dengan berkembangnya *deep learning*, ditemukan bahwa untuk mengatasi kekurangan MLP dalam menangani data kompleks, diperlukan fungsi untuk mentransformasi data input menjadi bentuk yang lebih mudah dimengerti oleh MLP. Hal tersebut memicu berkembangnya model *neural network* dengan jumlah *layer* diatas tiga. CNN merupakan suatu *layer* yang memiliki susunan *neuron* 3D (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran *layer* sedangkan kedalaman mengacu pada jumlah *layer*. Dengan demikian, CNN merupakan metode untuk mentransformasikan gambar original *layer by layer* dari nilai *pixel* gambar kedalam nilai skoring kelas untuk klasifikasi. Setiap *layer* ada yang memiliki *hyperparameter* dan ada yang tidak (bobot dan bias pada *neuron*) (Zufar & Setiyono, 2016). LeNet-5 merupakan sebuah jaringan konvolusional 7-level yang awalnya dikembangkan oleh

LeCun pada tahun 1998 (Haridas, dkk., 2019). Arsitektur LeNet-5 digunakan untuk mengekstraksi fitur antarmuka BrainComputer Sakhavi, dkk., (2018) dengan mempertimbangkan *common spatial patterns* (CSPs) *algorithm*. Tetapi masalah utama yang terjadi ialah akurasi rata-rata yang diperoleh oleh jaringan yang dimodifikasi lebih rendah dari akurasi aslinya (Kurniawan and Mustikasari 2021).



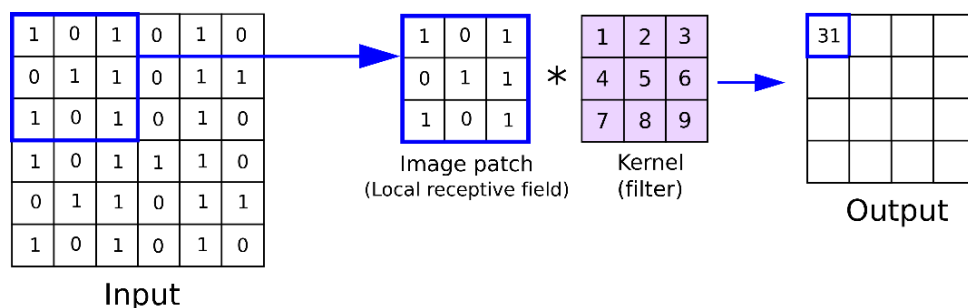
Gambar II.3 Arsitektur *Convolutional Neural Network*

Berdasarkan Gambar II.3, arsitektur CNN terbagi menjadi dua bagian besar yaitu: *convolutional layer* (terdiri dari *pooling layer*, *activation function* dan *hyperparameter*) dan *fully connected layer*.

a. *Convolution Layer*

Convolution layer merupakan bagian dari tahap pada arsitektur CNN. Tahap ini melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari jaringan arsitektur CNN. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada *output* fungsi lain secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi *output* sebagai *feature map* dari *input* citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen bernilai riil. *Convolutional Layer* terdiri dari neuron yang tersusun sedemikian rupa sehingga

membentuk sebuah filter dengan panjang dan tinggi (*pixels*). Sebagai contoh, *layer* pertama pada *feature extraction layer* biasanya adalah conv. *Layer* dengan ukuran 5x5x3. Panjang 5 *pixels*, tinggi 5 *pixels* dan tebal/jumlah 3 buah sesuai dengan channel dari *image* tersebut. Ketiga filter ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara *input* dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* atau biasa disebut sebagai activation map atau *feature map*. Perhatikan ilustrasi berikut :

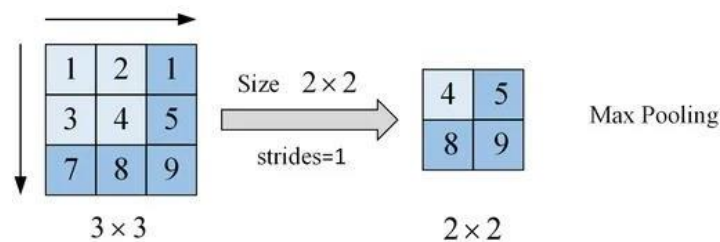


Gambar II.4 Ilustrasi *Feature Map*

b. Pooling

Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. *Pooling layer* biasanya berada setelah lapisan konvolusi. Pada dasarnya *pooling layer* terdiri dari sebuah filter dengan ukuran dan stride tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max-pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan *Pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume *output* pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di

jaringan, untuk mengendalikan *overfitting*. Lapisan *pooling* bekerja di setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan *pooling* umumnya dengan menggunakan filter dengan ukuran 2×2 yang diaplikasikan dengan langkah sebanyak dua dan beroperasi pada setiap irisan dari *inputnya*. Berikut ini adalah contoh gambar operasi *max-pooling* :

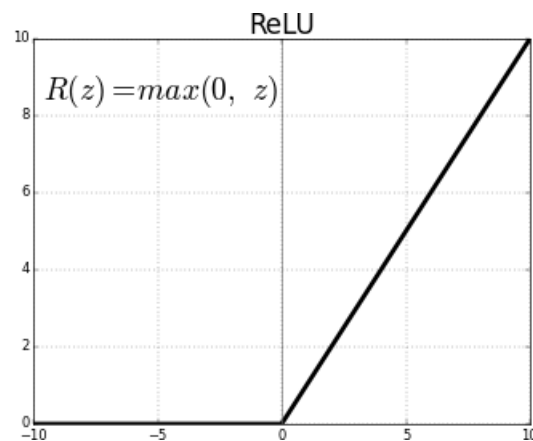


Gambar II.5 Contoh Operasi *Max Pooling*

Gambar II.5 menunjukkan proses dari *max-pooling*. *Output* dari proses pooling adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. Lapisan pooling diatas akan beroperasi pada setiap irisan kedalaman volume *input* secara bergantian. Jika dilihat dari gambar diatas operasi *max-pooling* dengan menggunakan ukuran filter 2×2 . Masukan pada proses tersebut berukuran 3×3 , dari masing-masing 3 angka pada *input* operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2×2 .

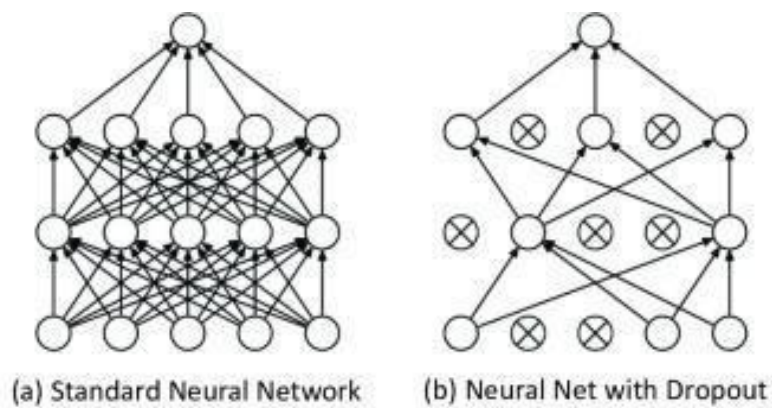
c. *Activation Function : ReLu (Rectified Linear Units)*

Rectified Linear Units (ReLU) layer ini mengaplikasikan fungsi $f(x) = \max(0, x)$ untuk meningkatkan sifat nonlinearitas fungsi keputusan dan jaringan secara keseluruhan tanpa mempengaruhi bidang-bidang reseptif pada *convolutional layer*. Contoh operasi ReLU seperti pada Gambar II.5.

Gambar II.6 Ilustrasi *Feature Map*

d. *Dropout*

Dropout merupakan sebuah teknik regulasi jaringan syaraf dengan tujuan memilih beberapa *neuron* secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain *neuron-neuron* tersebut dibuang secara acak. Hal ini berarti bahwa kontribusi neuron yang

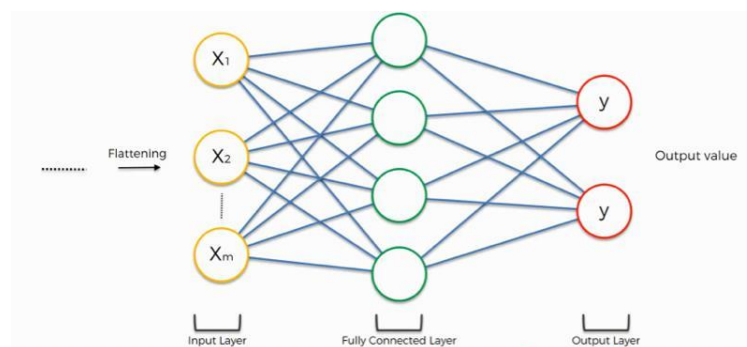
Gambar II.7 Ilustrasi *Dropout*

dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak diterapkan pada *neuron* pada saat melakukan *backpropagation*.

Gambar II.7 merupakan contoh proses dari *dropout*.

e. *Fully Connected Layer*

Fully-Connected Layer adalah sebuah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya sama seperti halnya dengan neural network biasa. Pada dasarnya lapisan ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linier. Perbedaan antara lapisan *fully-connected* dan lapisan konvolusi biasa adalah neuron di lapisan konvolusi terhubung hanya ke daerah tertentu pada *input*, sementara lapisan *fully-connected* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk dot, sehingga fungsinya tidak begitu berbeda. Gambar II.8 adalah contoh dari proses *fully-connected*.



Gambar II.8 Proses *Fully Connected Layer*

f. *Softmax Classifier*

Softmax Classifier adalah generalisasi dari fungsi logistik. *Output* dari softmax ini dapat digunakan untuk mewakili distribusi sebuah

kategori. *Softmax function* digunakan dalam berbagai macam metode klasifikasi.

contohnya *multinomial logistic regression*, *multiclass linear discriminant analysis*, *naive Bayes classifier*, dan *neural network*. Secara spesifiknya fungsi ini biasa digunakan pada metode klasifikasi *multinomial logistic regression* dan *multiclass linear discriminant analysis*.

Notasi f_j menunjukkan hasil fungsi untuk setiap elemen ke- j pada vektor keluaran kelas. Argumen z adalah hipotesis yang diberikan oleh model pelatihan agar dapat diklasifikasi oleh fungsi softmax. *Softmax* juga memberikan hasil yang lebih intuitif dan juga memiliki interpretasi probabilistik yang lebih baik dibanding algoritma klasifikasi lainnya. *Softmax* memungkinkan kita untuk menghitung probabilitas untuk semua label. Dari label yang ada akan diambil sebuah vektor nilai bernilai riil dan merubahnya menjadi vektor dengan nilai antara nol dan satu yang bila semua dijumlah akan bernilai satu.

II.4.2 Adaptive Kernel

Adaptive kernel merupakan sebuah modul yang memiliki kemampuan untuk secara otomatis melakukan penyesuaian terhadap kebutuhan. Tujuan utama dari modul ini adalah untuk memodelkan sistem secara dinamis. Permasalahan utama yang atasi adalah terkait variabel-variabel dengan sifat yang bervariasi, memiliki nilai berbeda-beda, dan tidak tetap. Modul ini memiliki dua fungsi, yaitu *utility function* berupa fungsi

untuk mengurutkan atau mendefinisikan ragam alternatif menurut kegunaannya, dan *learner function* yaitu melakukan pembelajaran dan pengenalan untuk memperoleh hasil yang paling optimal. Fungsi kernel didapatkan dengan menggunakan hasil dari konvolusi tiap input konvolusi $Q_{u,v}$ seperti pada persamaan (Zamora, dkk., 2019):

$$\sigma_{u,v} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} Q(u,v)_{i,j} x_{i,j} \quad (2)$$

Kernel baru $K_{u,v}$ dapat digunakan selanjutnya untuk melakukan konvolusi pada input gambar untuk menghasilkan S yang selanjutnya dijadikan sebagai kernel *output* seperti pada perhitungan (Zamora, dkk., 2019):

$$S = \sum_{u,v} x_{u,v} K \left(\sum_{i,j} Q_{u,v} x_{i,j} \right) \quad (3)$$

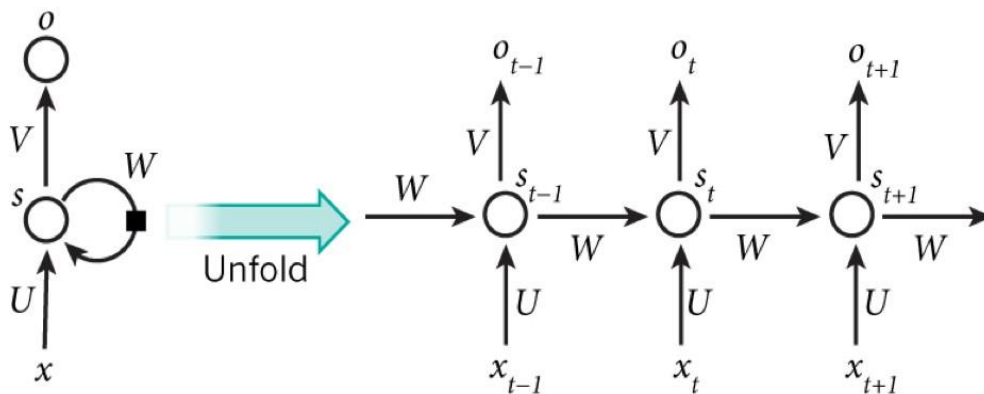
II.4.3 Adagrad

Adagrad adalah varian dari *Stochastic Gradient Descent* yang mengubah *learning rate* pada setiap parameter dalam proses pelatihan, oleh karena itu disebut *Adaptive Gradient Descent Algorithm* (Zhang, dkk., 2018). Pada tahun 2011, *Adagrad* diperkenalkan oleh (Duchi, dkk., 2011). *Adagrad* memodifikasi tingkat pembelajaran yang beradaptasi dengan arah penurunan menuju nilai optimum. Dengan kata lain, *Adagrad* mempertahankan tingkat pembelajaran yang rendah untuk fitur yang sering muncul dan tingkat pembelajaran yang tinggi untuk fitur yang jarang muncul. Kemudian, *Adagrad* menerapkan tingkat pembelajaran yang berbeda untuk setiap parameter pada setiap waktunya.

II.5 Recurrent Neural Network (RNN)

RNN merupakan salah satu metode *deep learning* yang merupakan salah satu bentuk arsitektur *Artificial Neural Networks* (ANN) yang dirancang khusus untuk memproses data yang bersambung/ berurutan (*sequential data*). RNN biasanya digunakan untuk menyelesaikan tugas yang terkait dengan data *time series*. *Recurrent Neural Network* (RNN) merupakan salah satu model arsitektur jaringan saraf tiruan yang dirancang khusus untuk memproses data yang berurutan (*sequential data*). RNN termasuk dalam pemrosesan Bahasa (Saptarini and Mustika 2023). Pengenalan suara, mesin terjemahan, pemodelan bahasa tingkat karakter, klasifikasi gambar, keterangan gambar, prediksi saham, dan rekayasa keuangan merupakan contoh aplikasi RNN. Operasi pada RNN dilakukan untuk setiap elemen dari suatu urutan, dengan hasilnya tergantung pada *input* saat ini dan operasi sebelumnya. RNN berfokus pada sifat data dimana hasil waktu sebelumnya atau saat ini mempengaruhi hasil pada waktu berikutnya (Saptarini and Mustika 2023).

Pada gambar sebelah kiri merupakan sebuah diagram sirkuit dimana kotak hitam menunjukkan penundaan waktu dari suatu *time step*. Gambar tersebut menunjukkan RNN pada posisi yang tidak dibuka (*unrolled/unfolded*) ke jaringan penuh (*full network*). Sedangkan gambar sebelah kanan menunjukkan RNN yang telah dibuka menjadi jaringan penuh (Cahyadi, Damayanti, and Aryadani 2020).



Gambar II.9 Proses RNN ketika melakukan perhitungan

t = Waktu saat sebelumnya atau saat ini (*time step*).

t+1 = Waktu berikutnya.

x_t = *Input* pada *time step*.

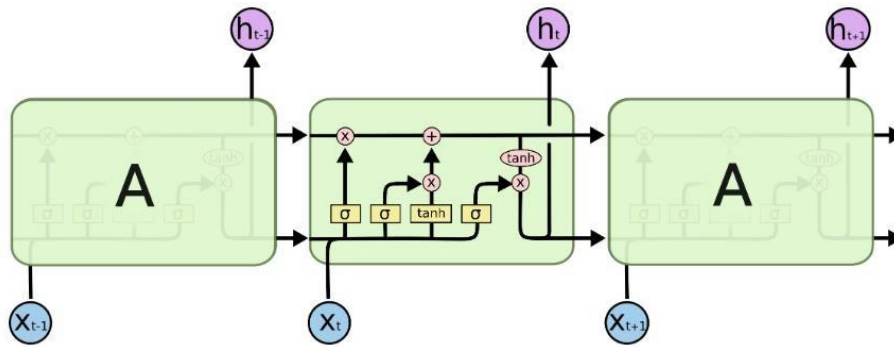
s_t = *Hidden state* pada setiap *time step*.

o_t = *Output* pada *step t*.

RNN memberikan manfaat untuk mengolah kata-kata yang *sekuens* dan memori jangka panjang sehingga menghasilkan kinerja yang sangat baik dan balasan yang akurat dan masuk akal pada *domain* terbuka.

II.6 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) adalah salah satu kelas pada RNN yang dapat mengatur memori pada setiap masukannya dengan menggunakan *memory cells* dan *gate units*. LSTM memiliki arsitektur mengingat dan melupakan hasil yang akan diproses kembali menjadi input (Zhang, 2016). Selain itu, LSTM dapat mempertahankan *error* yang terjadi ketika melakukan *back-propagation* sehingga tidak memungkinkan kesalahan meningkat.



Gambar II.10 Flowchart Memory Cells LSTM

Pada Gambar II.10, terdapat alur kerja *memory cells* pada setiap neurons LSTM. Terdapat empat proses fungsi aktivasi pada setiap masukan pada *neurons* yang selanjutnya disebut sebagai *gates unit* yaitu *forget gates*, *input gates*, *cell gates*, dan *output gates*. [25]. Langkah-langkah yang dilakukan pada LSTM [26] adalah:

1. Menentukan informasi yang akan dibuang dari *cell state*. Setiap *input* akan diolah untuk dipilih mana saja yang akan disimpan atau dibuang pada *memory cells*. Keputusan ini dibuat oleh *sigmoid layer* yang bernama *forget gate layer*. Hasil keputusan akan berupa angka diantara 0 dan 1 dimana 1 menyatakan semua *input* akan disimpan dan 0 menyatakan semua *input* akan dibuang.

$$f_t = \sigma(\mathbf{w}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f)$$

Rumus 2.1. Forget Gate Layer

2. Menentukan informasi baru apa yang akan kita simpan pada *cell state*. Pada langkah ini terdapat 2 bagian, pertama *input gate layer* menentukan nilai mana yang akan diperbaharui. Selanjutnya, *tanh layer* akan membuat sebuah vector dari nilai kandidat baru.

$$i_t = \sigma(w_i.[h_{t-1}, x_t] + b_i)$$

Rumus 2.2. Input Gate Layer

$$\tilde{C}_t = \tanh(w_c.[h_{t-1}, x_t] + b_i)$$

Rumus 2.3. Input Gate Layer (tanh)

3. Memperbaharui *cell state* lama C_{t-1} menjadi *cell gate* baru C_t . Nilai ini didapatkan dari hasil gabungan nilai yang terdapat pada *forget gate* dan *input gate*.

$$\tilde{C}_t = f_t * c_{t-1} + i_t * \tilde{C}_t$$

Rumus 2.4. Cell Gate Layer

4. Menentukan hasil (*output*) yang akan dikeluarkan. *Output gates* akan berdasarkan *cell state* tetapi dalam versi yang telah di *filter* menggunakan aktivasi *sigmoid*. Pertama kita akan menjalankan *sigmoid layer* yang akan menentukan bagian dari *cell state* yang akan menjadi *output*. Kemudian, kita akan mengubah nilai *cell state* menjadi antar -1 dan 1 dengan *tanh* dan akhirnya kedua hasil tersebut akan kita kalikan untuk menghasilkan nilai yang akan dikeluarkan.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$$

Rumus 2.5. Output Gate Layer

$$h_t = o_t \tanh(c_t)$$

Rumus 2.6. Output Gate Layer

II.7 Penelitian yang Relevan

No	Penelitian	Metode	Tujuan	Kekurangan	Manfaat
1.	<i>Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network</i>	CNN	Dalam penelitian ini menyelidiki efek dalam kinerja CNN setelah menghapus <i>stopwords</i> dan kata-kata langka dari teks Urdu.	Penelitian ini menggunakan teks bahasa arab	Mempermudah Klasifikasi teks tingkat dokumen menggunakan Filter <i>Multisize</i> Lapisan Tunggal CNN
2.	Klasifikasi Teks Laporan Masyarakat Pada Situs Lapor! Menggunakan <i>Recurrent Neural Network</i>	RNN	Dapat mempercepat proses verifikasi data laporan masyarakat, mengetahui kinerja dari metode <i>Recurrent Neural Network</i> dalam klasifikasi	Penelitian ini banyak mengumpulkan tulisan dari laporan masyarakat untuk uji coba CNN	Mempermudah klasifikasi teks laporan masyarakat pada situs laporan

			teks laporan masyarakat serta memperoleh hasil akurasi yang baik.		
3.	<i>Recurrent Neural Network With Gate Recurrent Unit For Stock Price Prediction</i>	RNN, GRU	Melakukan evaluasi pada parameter-parameter arsitektur RNN-GRU yang mempengaruhi prediksi dengan tolak ukur MAE, RMSE, DA, dan MAPE	Nilai error pada penelitian ini (MAE, RMSE, MAPE) secara konstan naik seiring dengan bertambahnya rentang label.	Prediksi harga saham dengan RNN-GRU memiliki performa yang berbeda beda tergantung berapa jauh model memprediksi dan liquiditas perusahaan.
4.	<i>Recurrent Neural Network (RNN) Dengan Long Short Term Memory (LSTM)</i>	<i>Deep Learning,</i> RNN, LSTM	Tujuan dari penelitian ini adalah sebuah sistem yang dapat mengklasifikasi sentimen.	Kekurangan dalam proses analisis sentimen adalah data komentar yang	Komentar yang diidentifikasi apakah komentar memiliki sentimen positif, netral,

	Untuk Analisis Sentimen Data Instagram			jumlahnya tidak seimbang (<i>imbalanced dataset</i>) sehingga perlu dilakukan beberapa langkah tambahan dalam menyiapkan data untuk pelatihan model.	atau negatif. Penelitian ini menggunakan data sebanyak 1.473 data yang diperoleh dari hasil <i>crawling</i> .
5.	Prediksi Harga Saham menggunakan Metode <i>Recurrent Neural Network</i> (RNN)	<i>Recurrent Neural Network</i> (RNN)	Penggunaan <i>Recurrent Neural Network</i> (RNN) untuk melakukan prediksi terhadap nilai saham dari histori harga saham. Fitur	faktor-faktor yang berpengaruh terhadap pergerakan harga saham sebagai variasi data latih dan	Prediksi tujuh fitur variabel dengan RNN menghasilkan akurasi sebesar 94% untuk data latih dan 55% untuk data

			yang diidentifikasi yaitu harga terendah, harga tertinggi, harga buka, harga tutup, volume, rata-rata harga, dan pergerakan	memadukan antara analisis teknikal, fundamental sebagai atribut-atribut pelatihan dan pengujian.	uji. Akurasi diperoleh setelah pelatihan dengan menggunakan 1218 data.
6.	Klasifikasi Kalimat Ilmiah Menggunakan <i>Recurrent Neural Network</i>	<i>Recurrent Neural Network, Long Short Term Memory</i>	Pengklasifikasian hanya berbentuk satu kalimat ilmiah tunggal dan tidak terpengaruh oleh kalimat sebelum dan sesudah tetapi hanya berfokus pada satu kalimat ilmiah saja.	Dalam penelitian ini masih adanya fluktuasi dan belum asimtotik pada proses pembelajaran mesin sehingga terdapat peluang untuk	Penelitian ini telah berhasil membuat model komputasi klasifikasi kalimat menggunakan RNN, dengan fitur yang telah diekstraksi menggunakan fungsi Word2Vec untuk

				pengembangan pada penelitian yang akan datang.	menghasilkan satu set vektor.
7.	Penerapan Kecerdasan Buatan Dan Teknologi Informasi Pada Efisiensi Manajemen Pengetahuan	Teknologi Informasi, Kecerdasan Buatan	Penelitian ini secara sistematis menerapkan praktik manajemen pengetahuan saat ini yang mengandalkan mekanisme TI dan AI dan dampaknya terhadap perusahaan beserta tantangan dan keterbatasannya.	Pada penelitian ini penerapan AI tidak di spesifikasikan pada suatu metode tertentu	Dari hasil penelitian ini tehnik AI yang paling sering di gunakan adalah metode <i>Artificial Neural Network</i> dimana negara yang meneliti subjek terbanyak yaitu Arab Saudi dan UK, dengan studi manajemen pengetahuan yang terkait dengan AI meningkat

					dalam tiga tahun terakhir dari 2016 hingga 2019
8.	Pengaruh Kepemimpinan <i>Agility</i> , Budaya Organisasi Dan Motivasi Terhadap Agilitas Organisasi Pada Pascasarjana UMSU.	Kepemimpinan <i>Agility</i> , Agilitas Organisasi	Untuk mengetahui dan menganalisis pengaruh Kepemimpinan <i>Agility</i> terhadap Agilitas Organisasi Pada Pascasarjana UMSU.	Kekurangan penelitian ini adalah variable hanya berfokus Kepemimpinan <i>Agility</i> , Budaya Organisasi, Motivasi, Agilitas Organisasi	Untuk mengetahui dan menganalisis pengaruh Budaya Organisasi terhadap Agilitas Organisasi Pada Pascasarjana UMSU.
9.	<i>Application of Deep Learning Using Convolutional Neural</i>	<i>Deep Learning, Convolutional</i>	Penggunaan metode CNN dalam pengenalan objek otomatis dalam	Tidak adanya perbandingan metode CNN pada	metode <i>Convolutional Neural Network (CNN)</i> merupakan metode yang

	<i>Network (CNN) Method for Women's Skin Classification</i>	<i>Neural Network (CNN)</i>	membedakan jenis kulit sebagai bahan pertimbangan dalam menentukan objek citra.	penelitian ini	tepat untuk mengklasifikasikan jenis kulit wanita usia 20 – 30 tahun berdasarkan.
10.	<i>Review of deep learning: concepts, CNN architectures, challenges, applications, future directions</i>	<i>Convolution neural network (CNN), Deep neural network architectures.</i>	Melihat perkembangan penelitian menggunakan <i>deep learning, machine learning dengan metode convolutional neural networks (CNNs), Medical image analysis, Supervised learning, FPGA</i>	Kekurangan penelitian ini hanya memfokuskan pada <i>Medical image analysis</i> saja	Menggunakan <i>Deep learning, Machine learning, Convolution neural network (CNN), Deep neural network architectures</i> , untuk di terapkan pada <i>Medical image analysis</i> .
11.	Klasifikasi Citra	<i>Deep learning,</i>	Mengolah 390 citra pada	Pada penelitian ini	Melakukan klasifikasi

	Menggunakan <i>Convolutional Neural Network</i> (CNN) pada Caltech 101	<i>Convolution Neural Network</i>	data Caltech-101 yang didapat dari 150 citra adalah kategori Unggas dan sisanya adalah <i>Crocodile</i> , <i>Cougar</i> , dan <i>Face</i> masing-masing dari ketiga kategori tersebut 80 citra	hanya Klasifikasi Kategori <i>Cougar</i> , <i>Crocodile</i> dan <i>Face</i>	<i>Emu</i> , <i>Flamingo</i> , <i>Ibis</i> , <i>Pigeon</i> , dan <i>Rooster</i> yang terdiri dari 150 citra. Kategori lainnya, yaitu: <i>Cougar</i> (<i>Cougar Body</i> dan <i>Cougar Face</i>), <i>Crocodile</i> (<i>Crocodile</i> dan <i>Crocodile Head</i>) dan <i>Face</i> (<i>Face</i> dan <i>Face Easy</i>)
12.	Implementasi <i>Deep Learning</i> Menggunakan Metode CNN dan LSTM untuk Menentukan Berita	<i>Deep Learning</i> ; CNN; LSTM	Tujuan dari penelitian ini adalah menerapkan teknik deep learning untuk menentukan berita fakta	Data uji yang digunakan tahun 2016 sedangkan penelitian tahun 2020	untuk menentukan berita fakta dan berita palsu dalam bahasa Indonesia.

	Palsu dalam Bahasa Indonesia		dan berita palsu dalam bahasa Indonesia. Metode yang digunakan adalah <i>Convolutional Neural Network</i> (CNN) dan <i>Long Short Term Memory</i> (LSTM).		
--	------------------------------	--	---	--	--