**PROYEKSI CURAH HUJAN DAERAH PADANG PARIAMAN  
MENGGUNAKAN DEEP LEARNING DENGAN  
METODE LONG SHORT-TERM MEMORY**

**SKRIPSI**

*Untuk Memenuhi Salah Satu Syarat*

*Memperoleh Gelar Sarjana Komputer*

**Program Studi : Teknik Informatika  
Jenjang Pendidikan : Strata 1 (S1)**



**OLEH :**

**EDO SULAIMAN**  
**NIM. 18101152630092**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG**

**2021**

# BAB II LANDASAN TEORI

## Rekayasa Perangkat Lunak

IEEE Computer Society mendefinisikan Rekayasa Perangkat Lunak (RPL) sebagai penerapan suatu pendekatan yang sistematis, disiplin dan terkuantifikasi atas pengembangan, penggunaan dan pemeliharaan perangkat lunak, serta studi atas pendekatan-pendekatan ini, yaitu penerapan pendekatan engineering atas perangkat lunak (Hasanah & Untari, 2020).

RPL sendiri adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, penggodaan, pengujian sampai pemeliharaan sistem setelah digunakan (Hasanah & Untari, 2020).

RPL lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan (customer). Adapun ilmu computer lebih fokus pada teori dan konsep dasar perangkat computer. Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria berikut (Hasanah & Untari, 2020) :

1. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (maintainability)
2. Dapat diandalkan dengan proses bisnis yang dijalankan perubahan yang terjadi (dependability robust)
3. Efisien dari segi sumber daya dan penggunaan
4. Kemampuan untuk dipakai sesuai dengan kebutuhan (usability)

### Proses Rekayasa Perangkat Lunak

Kerangka kerja proses membangun dasar bagi proses rekayasa perangkat lunak yang lengkap dengan cara mengidentifikasi sejumlah kecil aktivitas kerangka kerja yang cocok bagi semua proyek rekayasa perangkat lunak (Setiyani, 2018). Kerangka kerja proses pada rekayasa perangkat lunak terdiri atas lima aktivitas berikut (Setiyani, 2018) :

#### Komunikasi

Komunikasi, bertujuan untuk memahami tujuan-tujuan stakeholder atas proyek perangkat lunak yang sedang dikembangkan dan mengumpulkan kebutuhan-kebutuhan yang akan membantu mendefinisikan fitur-fitur perangkat lunak berikut dengan fungsi-fungsinya (Setiyani, 2018) :

#### Perencanaan

Kegiatan perencanaan menciptakan suatu peta yang dapat membantu membimbing tim perangkat lunak. Rencana proyek perangkat lunak menggambarkan risiko – risiko yang mungkin muncul, sumber daya yang akan dibutuhkan, produk – produk kerja yang harus dihasilkan dan schedule kerja (Setiyani, 2018) :

#### Pemodelan

Pemodelan, dilakukan bertujuan untuk membuat sketsa sehingga tim perangkat lunak dapat memahami gambaran besar produk yang akan di buat (Setiyani, 2018) :

#### Konstruksi

Konstruksi, sendiri adalah kegiatan yang menggabungkan penggodaan dan pengujian (Setiyani, 2018) :

#### Penyerahan

Penyerahan di sini merupakan Penyerahan perangkat lunak kepada user, penyajian perangkat lunak kepada user untuk di evaluasi (Setiyani, 2018) :

### Software Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) adalah metodologi klasik yang digunakan untuk mengembangkan, memelihara dan menggunakan sistem informasi. Siklus hidup sistem itu sendiri merupakan metodologi, tetapi polanya lebih dipengaruhi oleh kebutuhan untuk mengembangkan sistem yang lebih cepat. Pengembangan sistem yang lebih cepat dapat dicapai dengan peningkatan siklus hidup dan penggunaan peralatan pengembangan berbasis computer (Wahyudi, 2018).

#### Tahap-Tahap SDLC

Secara umum tahap-tahap dalam System Development Life Cycle (SDLC)terbagi dalam beberapa tahap (Wahyudi, 2018):

##### Tahap Perencanaan Sistem (system planning)

Tahap Planning Merupakan tahap awal dari pengembangan sistem, tahap ini bertujuan untuk mengidentifikasi dan memprioritaskan sistem informasi apa yang akan dikembangkan, sasaran-sasaran yang ingin dicapai, jangka waktu pelaksanaan serta mempertimbangkan dana yang tersedia dan siapa yang melaksanakan (Wahyudi, 2018).

##### Tahap Analisis Sistem (system analysis)

Tahap Analisis system adalah penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem baru atau memperbaharui sistem yang sudah ada (Wahyudi, 2018).

##### Tahap Perancangan/Desain Sistem (system design)

Tahap Rancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru. Jika sistem ini berbasis komputer, rancangan dapat menyerta kan spesifikasi jenis peralatan yang akan digunakan (Wahyudi, 2018).

##### Tahap Penerapan/Implementasi Sistem (system implementation)

Tahap Penerapan merupakan kegiatan memperoleh dan mengintegrasikan sumber daya fisik dan konseptual yang menghasilkan suatu sistem yang bekerja. Pada tahapan ini dilakukan beberapa hal yaitu: Coding, Testing, Instalasi. Dan Output dari tahapan ini adalah : source code, prosedur, pelatihan (Wahyudi, 2018).

##### Tahap Pemeliharaan/Perawatan Sistem

Tahap pemeliharaan/perawatan sistem merupakan tahap yang dilakukan setelah tahap implementasi yang meliputi penggunaan sistem, audit sistem, penjagaan sistem, perbaikan sistem dan peningkatan sistem (Wahyudi, 2018).

#### Model Waterfall

Model SDLC air terjun (waterfall) sering juga disebut model sekuensial linier (sequential linier) atau alur hidup klasik (classic life cycle). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuential atau terurut dimulai dari analisis kebutuhan perangkat lunak, desain, pembuatan kode program, pengujian, dan pemeliharaan (maintenance) (Tabrani & Pudjiarti, 2021). Berikut tahap tahapannya sebagai berikut (Tabrani & Pudjiarti, 2021):

##### Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan (Tabrani & Pudjiarti, 2021).

##### Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur penggodaan. Tahap ini mentrannslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan (Tabrani & Pudjiarti, 2021).

##### Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain (Tabrani & Pudjiarti, 2021).

##### Pengujian

Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan (Tabrani & Pudjiarti, 2021).

##### Pendukung atau Pemeliharaan (maintenance)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru (Tabrani & Pudjiarti, 2021).

## Unified Modelling Language *(UML)*

Pemodelan dalam suatu rekayasa perangkat lunak merupakan suatu hal yang dilakukan di tahapan awal. Pemodelan dalam perangkat lunak merupakan suatu yang harus dikerjakan di bagian awal dari rekayasa, dan pemodelan ini akan mempengaruhi perkerjaan-pekerjaan dalam rekayasa perangkat lunak tersebut. Salah satu perangkat pemodelan adalah Unified Modelling Language (UML). UML merupakan salah satu standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menypesifikasikan, menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak (Hasanah & Untari, 2020).



(Sumber : Hasanah & Untari, 2020)*.*

Gambar 2.1. Bagan UML

Berdasarkan **Gambar 2.1** berikut penjelasan singkat dari pembagian kategori tersebut (Hasanah & Untari, 2020) :

1. Behavior diagram, merupakan kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada suatu sistem.
2. Interaction diagram, merupakan kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar sub sistem pada suatu sistem.
3. Structure diagram, merupakan kumpulan diagram yang digunakan untuk menggambarkan struktur statis dari sistem yang dimodelkan.

### Behavior Diagrams

Behavior diagram adalah salah satu jenis diagram yang ada di dalam *Unified Modeling Language* (UML), dimana diagram ini digunakan untuk memberikan gambaran tingkah laku sebuah sistem informasi dan bagaimana sistem informasi tersebut melakukan tindakan terhadap kejadian atau perubahan (Sulistyo, Yudhana, & Sunardi, 2018).

#### Use Case Diagram

Use case diagram adalah teknik untuk merekam persyaratan fungsional sebuah system, menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Use case diagram menekankan kepada “apa” yang diperbuat oleh sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara actor dengan sistem. Seorang atau sebuah aktor adalah sebuah entitas dapat berupa manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu. (Hasanah & Untari, 2020).

Tabel 2.1. Notasi Use Case Diagram

|  |  |  |
| --- | --- | --- |
| No | Simbol | Keterangan |
| 1 | *Use case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. |
| 2 | *Actor*  *Description: Description: actor* | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem itu. |
| 3 | *Association* | use case yang memiliki interaksi dengan aktor. |
| 4 | *Extend* | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri. |
| 5 | *Generalization* | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan ini untuk menjalankan fungsinya. |

(Sumber : Hasanah & Untari, 2020)

#### Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram digunakan untuk menggambarkan langkahlangkah atau aktivitas pada suatu sistem (Hasanah & Untari, 2020).

Tabel 2.2. Notasi Activity Diagram

|  |  |  |
| --- | --- | --- |
| No | Simbol | Keterangan |
| 1 | *Initial State* | Sebuah diagram aktivitas memiliki sebuah status awal |
| 2 | *Activity* | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
| 3 | *Decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| 4 | *Join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
| 5 | *Final State* | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6 | *Swimlane*    or | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

(Sumber : Hasanah & Untari, 2020)

#### Statechart Diagram

Statechart diagram menelusuri individu-individu objek melalui keseluruhan daur hidupnya, menypesifikasikan semua urutan yang mungkin dari pesan-pesan yang akan diterima objek tersebut, bersama-sama dengan tanggapan atas pesan-pesan tersebut (Ilham & Fajri, 2020).

Tabel 2.3. Notasi Statechart Diagram

|  |  |  |
| --- | --- | --- |
| No | Simbol | Keterangan |
| 1 | *Initial State* | Kelas pada struktur sistem |
| 2 | *Final State* | Sama dengan konsep interface dalam beberapa pemrograman berorientasi objek. |
| 3 | *Event* | Relasi antarkelas dengan makna umum.  Asosiasi biasanya juga disertai dengan multiplicity |
| 4 | *State* | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |

(Sumber : Ilham & Fajri, 2020).

### Interaction Diagrams

Interaction diagram merupakan diagram yang digunakan untuk menggambarkan bagaimana sebuah objek berinteraksi baik aktor dan objek sistem (Ependi, Panjaitan, & Yulianingsih, 2018).

#### Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa messageyang digambarkan terhadap waktu. Sequence diagramterdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagrambiasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah eventuntuk menghasilkan outputtertentu. Diawali dari apa yang men-triggeraktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan outputapa yang dihasilkan (Rinaldi, 2019).

Tabel 2.4. Notasi State Machine Diagram

|  |  |  |
| --- | --- | --- |
| No | Simbol | Keterangan |
| 1 | *Actor*  *Description: Description: actor*  Or  Nama aktor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi biasanya dinyatakan menggunakan kata benda di awal frasa nama actor |
| 2 | *Lifeline* | Menyatakan kehidupan suatu objek |
| 3 | *Object*  objek : kelas | Menyatakan objek yang berinteraksi pesan |
| 4 | *Active Time* | Menyatakan objek dalam keadaan aktif berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, |
| 5 | *Message type C****reate***  <<*create*>> | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| 6 | *Message type* ***Call***  1:nama\_metode() | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri |
| 7 | *Message type* ***Send***  1: masukan | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim |
| 8 | *Message type* ***Return***  1: keluaran  ---------------------- | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian |
| 9 | *Message type* ***Destroy***  <<*destroy*>> | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, jika ada create maka ada destroy. |

(Sumber : Rinaldi, 2019)

#### Collaboration Diagram

Collaboration diagram adalah cara alternative untuk mengetahui tahap-tahap terjadinya suatu aktivitas. Perbedaan antara collaboration dan sequence diagram adalah collaboration diagram memperlihatkan bagaimana hubungan antara beberapa objek berdasarkan urutan dari pesan, sedangkan sequence diagram memperlihatkan bagaimana urutan kejadian berdasarkan waktu (Ilham & Fajri, 2020).

Tabel 2.5. Notasi Collaboration Diagram

|  |  |  |
| --- | --- | --- |
| No | Simbol | Keterangan |
| 1 | *Object*  objek : kelas | Objek yang melakukan interaksi pesan. |
| 2 | *Link* | Relasi antara objek yang menghubungkan objek satu dengan lainya atau dengan diri sendiri. |
| 4 | *State* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |

(Sumber : Ilham & Fajri, 2020)

### Structure Diagrams

Structure diagram merupakan diagram yang digunakan untuk menggambarkan struktur dari perangkat lunak atau aplikasi yang dikembangkan (Ependi, Panjaitan, & Yulianingsih, 2018).

#### Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class diagram menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain (Rinaldi, 2019).

Tabel 2.6. Notasi Class Diagram

|  |  |  |
| --- | --- | --- |
| No | Simbol | Keterangan |
| 1 | *Class*  **nama\_kelas**  + atribut  + operasi() | Kelas pada struktur sistem |
| 2 | *Interface* | Sama dengan konsep interface dalam beberapa pemrograman berorientasi objek. |
| 3 | *Association* | Relasi antar kelas dengan makna umum.  Asosiasi biasanya juga disertai dengan multiplicity |
| 4 | *Directed Association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |
| 5 | *Dependency* | Relasi antar kelas dengan ketergantungan antar kelas |
| 6 | *Aggregation* | Relasi antar kelas dengan makna semua-bagian (whole-part) |

(Sumber : Rinaldi, 2019)

#### Deployment Diagram

Menggambarkan secara lengkap bagaimana komponen di deployment dalam infrastruktur sistem, dimana komponen akan terletak, bagaimana kemampuan jaringan pada kondisi tertentu, spesifikasi server, dan hal-hal lain yang bersifat fisikal (Wijaya, Masriadi, & Ikhlas, 2020).

Tabel 2.7. Notasi Deployment Diagram

|  |  |  |
| --- | --- | --- |
| No | Simbol | Keterangan |
| 1 | *Package*  [Description: http://4.bp.blogspot.com/-sc_hs41py7c/VR5gbgH2yHI/AAAAAAAAAXU/xAgOjf7Nc-g/s1600/Snap%2B2015-04-03%2Bat%2B16.41.02.png](http://4.bp.blogspot.com/-sc_hs41py7c/VR5gbgH2yHI/AAAAAAAAAXU/xAgOjf7Nc-g/s1600/Snap+2015-04-03+at+16.41.02.png) | Package merupakan sebuah bungkusan dari satu atau lebih komponen. |
| 2 | *Node*  Nama\_node | Biasanya mengacu pada perangkat keras(hardware), perangkat lunak yang tidak dibuat sendiri (software), jika didalam node disertakan komponen untuk mengkonsistensikan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen. |
| 4 | *Dependency* | Ketergantungan antar komponen, arah panah mengarah pada komponen yang dipakai. |
| 5 | *Link* | Relasi antar komponen. |

(Sumber : Wijaya, Masriadi, & Ikhlas, 2020).

## Artificial Intelligence

*Andreas Kaplan* dan *Michael Haenlein* mendefinisikan kecerdasan buatan / *Artificial Intelligence (AI)* sebagai kemampuan system untuk menafsirkan data eksternal dengan benar, untuk belajar dari data tersebut, dan menggunakan pembelajaran tersebut guna mencapai tujuan dan tugas tertentu melalui adaptasi yang fleksibel. Sistem seperti ini umumnya dianggap sebagai computer (Siahaan, et al., 2020).

Lebih lanjut *Budiharto* menyatakan bahwa *Intelligence* merupakan istilah yang kompleks yang dapat didefinisikan dengan ungkapan yang berbeda seperti logika, pemahaman, self-awareness, pembelajaran, perencanaan, dan problem solving. Sedangkan “Artificial” adalah sesuatu yang tidak nyata, seperti tipuan karena merupakan hasil simulasi (Sihombing & Syaputra, 2020).

Savitri menguraikan bahwa Kecerdasan buatan / Artificial intelligence (AI) merupakan bidang ilmu komputer yang menekankan pada penciptaan mesin cerdas yang bekerja dan bereaksi seperti manusia yang perkembangannya terjadi sangat pesat di era revolusi industri keempat (Sihombing & Syaputra, 2020).

Kecerdasan diciptakan dan dimasukkan kedalam suatu mesin / komputer agar dapat melakukan pekerjaan seperti yang dapat dilakukan oleh manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (games), logika fuzzy, jaringan saraf tiruan dan robotika (Siahaan, et al., 2020).

Tujuan utama dari pembuatan AI adalah untuk membuat mesin memiliki fungsi yang memiliki kriteria-kriteria kecerdasan didalamnya sehingga mesin tersebut mampu melakukan pekerjaan manusia yang lebih kompleks, tergantung dari tingkat kecerdasan AI yang digunakan (Gunova, 2021).

Dalam penerapannya, terdapat 6 kemampuan utama yang dapat diklasifikasikan sebagai AI (Gunova, 2021), Kemampuan tersebut antara lain :

* Representasi Informasi/Pengetahuan (Knowledge Representation)
* Perencanaan (Planning)
* Persepsi (Perception)
* Machine Learning / Deep Learning
* Pemahaman Bahasa (Natural Language Processing)
* Robotics

Meskipun 6 kemampuan diatas memiliki fungsi yang berbeda-beda, kemampuan kemampuan tersebut saling berhubungan satu sama lain saat pengaplikasiaannya. Bahkan hubungan antar kemampuan utama diatas akan membentuk kemampuan baru (Gunova, 2021).

## Machine Learning

Machine Learning merupakan Disiplin ilmu yang menggunakan berbagai pendekatan untuk mengajarkan komputer untuk menyelesaikan tugas-tugas di mana tidak ada algoritma yang sepenuhnya memuaskan tersedia. Dalam kasus di mana terdapat sejumlah besar jawaban potensial, satu pendekatan adalah memberi label beberapa jawaban yang benar sebagai valid. Ini kemudian dapat digunakan sebagai data pelatihan bagi komputer untuk meningkatkan algoritme yang digunakannya untuk menentukan jawaban yang benar. Misalnya, untuk melatih sistem untuk tugas pengenalan karakter digital, dataset MNIST dari angka tulisan tangan sering digunakan (Alpaydin, 2020).

Program *Machine Learning* dapat melakukan tugas tanpa diprogram secara eksplisit untuk melakukannya. Ini melibatkan komputer belajar dari data yang disediakan sehingga mereka melakukan tugas-tugas tertentu. Untuk tugas-tugas sederhana yang ditugaskan ke komputer, dimungkinkan untuk memprogram algoritme yang memberi tahu mesin bagaimana menjalankan semua langkah yang diperlukan untuk memecahkan masalah yang dihadapi; di bagian komputer, tidak diperlukan pembelajaran. Untuk tugas yang lebih maju, mungkin sulit bagi manusia untuk membuat algoritme yang diperlukan secara manual. Dalam praktiknya, ternyata lebih efektif untuk membantu mesin mengembangkan algoritmenya sendiri, daripada meminta pemrogram manusia menentukan setiap langkah yang diperlukan (Alpaydin, 2020).

Algoritma *Machine Learning* digunakan dalam berbagai macam aplikasi, seperti dalam kedokteran, penyaringan email, pengenalan suara, dan visi komputer, di mana sulit atau tidak mungkin untuk mengembangkan algoritma konvensional untuk melakukan tugas-tugas yang diperlukan (Hu, Niu, Carrasco, Lennox, & Arvin, 2020).

Graphical user interface, text

Description automatically generated

(Sumber : Kumar, Amgoth, & Annavarapu, 2019)

Gambar 2.2. Struktur Machine learning

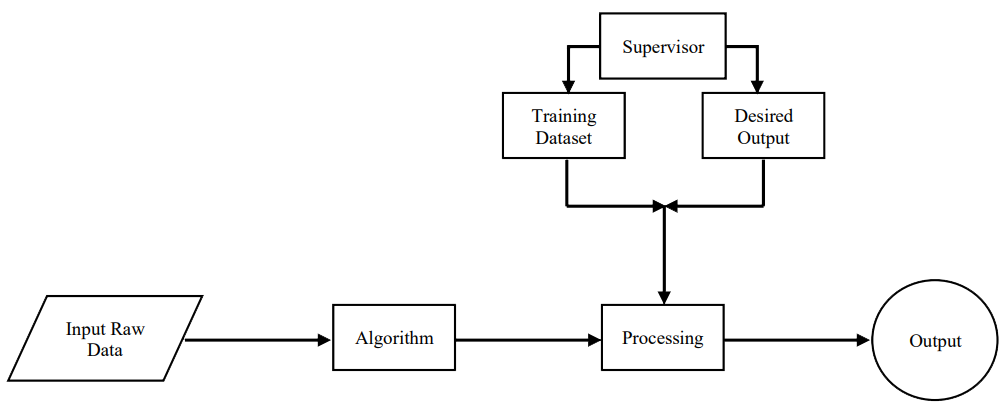
### Tipe-tipe Machine learning

Dalam Machine Learning, terdapat beberapa metode atau pendekatan yang dapat digunakan mesin untuk melakukan pembelajaran. Namun secara garis besar, terdapat 3 metode Machine Learning yang dipisahkan berdasarkan tipe input atau dataset dan cara pelatihannya (Gunova, 2021). Metode-metode tersebut antara lain :

#### Supervised learning

Supervised Learning (SL) merupakan sebuah metode ML dimana metode ini memberikan dataset yang berlabel dan data input kedalam mesin. Maksud dari dataset yang berlabel ini ialah untuk setiap tipe/bentuk dataset yang telah diberikan, outputnya telah ditentukan. Setelah menerima input, mesin kemudian akan memberikan output berdasarkan dataset yang dilabeli. Metode ini cocok digunakan untuk penyelesaian masalah classification dan regression (Gunova, 2021).

Algoritma yang termasuk kedalam teknik supervised learning diantaranya Decision Tree, K-Nearest Neighboor (KNN), Naive Bayes, Regresi, dan Super Vector Machine (Pamungkas, Prasetya, & Kharisudin, 2020).



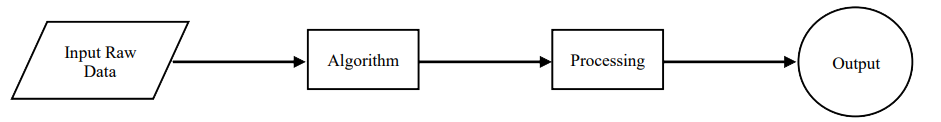
(Sumber : Gunova, 2021)

Gambar 2.3. Cara Kerja Supervised Learning (SL)

#### Unsupervised learning

Unsupervised Learning (UL) merupakan sebuah metode ML dimana metode ini hanya memberikan data input saja kedalam mesin. Setelah menerima input, mesin kemudian akan memberikan output berdasarkan pola data input yang diterima. Metode ini cocok digunakan untuk penyelesaian masalah pengelompokkan data, baik itu association maupun clustering (Gunova, 2021).

Beberapa Algoritma dalam unsupervised learning diantaranya DBSCAN, Fuzzy C-Means, K-Means, dan Self Organizing Map. DBSCAN pengelompokan berdasarkan kepadatan (density) data, konsep kepadatan menghasilkan status dari data yaitu core (inti), border (batas), dan noise (Ashari, Otniel, & Rianto, 2019).

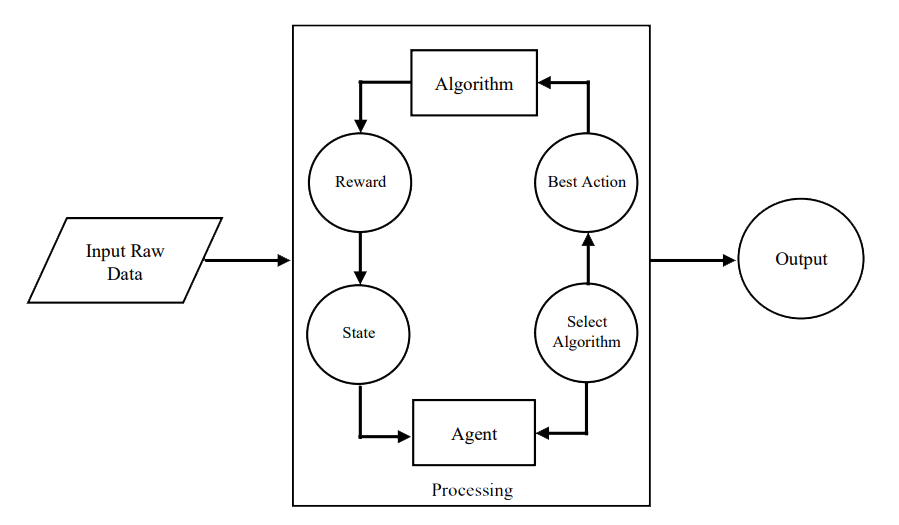


(Sumber : Gunova, 2021)

Gambar 2.4. Cara Kerja Unsupervised Learning (UL)

#### Reinforcement learning

Reinforcement machine learning adalah algoritma yang mempunyai kemampuan untuk berinteraksi dengan proses belajar yang dilakukan, algoritma ini akan memberikan poin (reward) saat model yang diberikan semakin baik atau mengurangi poin (error) saat model yang dihasilkan semakin buruk. Salah satu penerapan yang sering dijumpai yaitu pada mesin pencari (Fajarsari, 2020).

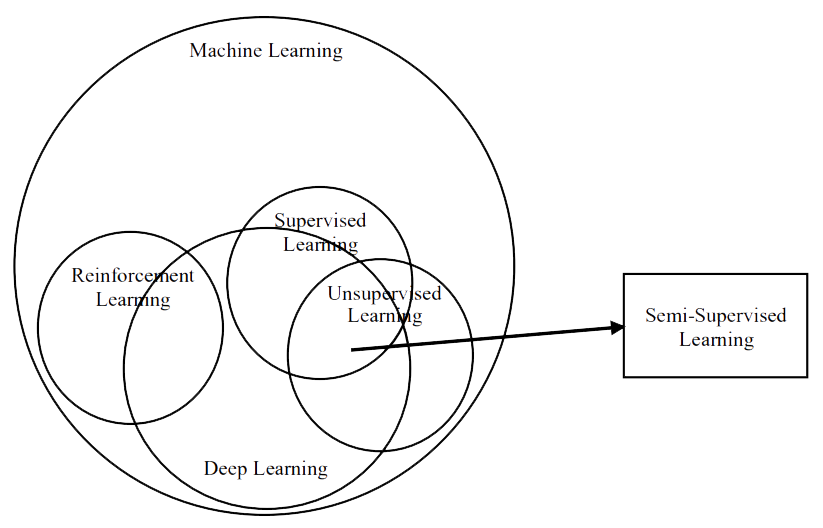


(Sumber : Gunova, 2021)

Gambar 2.5. Cara Kerja Reinforcement Learning (RL)

## Deep Learning

Sejak tahun 2006, *Deep Structured Learning* atau yang lebih dikenal dengan *Deep learning* atau *Hierarchical Learning* telah muncul sebagai area baru dalam penelitian *Machine Learning* yang berdasarkan pada suatu set algoritma yang mencoba untuk memodelkan abstraksi tingkat tinggi pada data dengan menggunakan graf yang mendalam dengan beberapa layer pengolahan, yang terdiri dari beberapa transformasi linier dan non-linier (François, 2018).



(Sumber : Gunova, 2021)

Gambar 2.6. Salah Satu Hubungan antar metode yang dalam ML (UL, SL, RL, dan DL)

*Deep Learning* sendiri adalah cabang ilmu *machine learning* berbasis *Neural Network (NN)* atau bisa dikatakan sebagai perkembangan dari *Neural Network* (Ilahiyah & Nilogiri, 2018). Metode pendekatan *deep learning* mengklasifikasi data dalam dua sesi yaitu sesi training dan testing. Pada sesi training mempelajari ekstrasi fitur dari setiap data supaya bisa membedakan suaru label dengan label yang lain. Pada sesi testing data-data yang diuji dapat dianalisa dari hasil sesi training (Azizah, Umayah, & Fajar, 2018).



(Sumber : Savalia & Emamian, 2018)

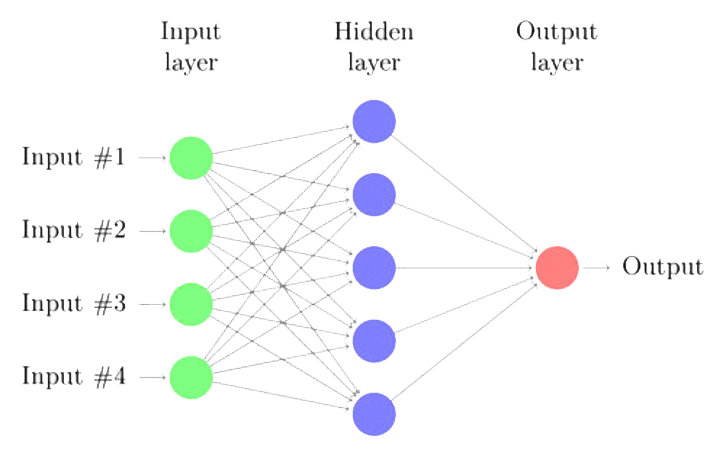
Gambar 2.7. Perbedaan Simple Neural Network dan Deep Learning

Dalam *deep learning*, sebuah komputer belajar mengklasifikasi secara langsung dari gambar atau suara. Metode *deep learning* menggunakan CPU dan RAM dalam proses komputasi, dan juga memanfaatkan GPU sehingga proses komputasi data yang besar dapat berlangsung lebih cepat (Ilahiyah & Nilogiri, 2018).

## Jaringan Syaraf Tiruan

*Jaringan saraf tiruan (JST) / Artificial Neural Network (ANN) / Neural Nenwork (NN)* adalah jaringan komputasi terinspirasi dari cara kerja otak manusia karena terbukti Otak manusia melakukan sesuatu hal yang sama persis seperti, hirarki pertama pada neuron menerima formasi pada visual cortex yang sensitif terhadap gambaran tepi dan gumpalan khusus (François, 2018).

Neural Network mempunyai lapisan masukan (input layer) dan lapisan keluaran (output layer). Pada setiap lapisan mempunyai satu atau beberapa unit neuron, dan mempunyai sebuah fungsi aktivasi dari unit tersebut untuk menentukan sebuah keluaran. Untuk meningkatkan kemampuan dari NN, dapat ditambahkan lapis tersembunyi atau hidden layer. Data training dapat digunakan untuk melatih NN, semakin banyak data training maka akan semakin baik unjuk kerja dari NN tersebut. Tetapi NN juga mempunyai keterbatasan pada jumlah lapisan, karena semakin banyak jumlah lapisan semakin banyaknya juga jumlah iterasi atau training yang dibutuhkan (Priyanto, Zarlis, Mawengkang, & Efendi, 2019).



(Sumber : Shekar, S´a, Ferreira, & Soares, 2018)

Gambar 2.8. Skema Jaringan Saraf Tiruan

Seperti yang terlihat pada **Gambar 2.8** Lapisan-lapisan penyusun JST tersebut dapat dibagi menjadi 3, yaitu (Satria, 2018) :

1. Lapisan Input, unit-unit di dalam lapisan input disebut unit-unit input. Unit-unit input tersebut menerima pola inputan data dari luar yang menggambarkan suatu permasalahan (Satria, 2018)
2. Lapisan tersembunyi, unit -unit di dalam lapisan tersembunyi disebut unit- unit tersembunyi (Satria, 2018)
3. Lapisan output, unit-unit didalam lapisan output disebut unit-unit output (Satria, 2018)

### Arsitektur Jaringan Syaraf Tiruan

Arsitektur atau struktur neural network adalah gambaran susunan komponen layer dan neuron pada input, hidden dan output yang terhubung dengan weight atau bobot, activation function dan learning function. Perceptron dan Multilayer Perceptron adalah dasar dari jaringan saraf tiruan. Sebuah perceptron adalah algoritma klasifikasi biner yang dimodelkan setelah berfungsinya otak manusia, hal ini dimaksudkan untuk meniru neuron. Meskipun perceptron memiliki struktur sederhana tetapi memiliki kemampuan untuk belajar dan menyelesaikan masalah yang sangat kompleks. Neural Netwrok yang paling populer adalah jaringan multi perceptron feed-forward yang dilatih melalui algoritma backpropagation (Ranjit, Shrestha, Subedi, & Shakya, 2018).

A picture containing shape

Description automatically generated

(Sumber : cllau, 2020)

**Gambar 2.9. Perceptron Input dan Output**

Menurut Revi, Solikhun, & Safii JST memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur JST tersebut, antara lain sebagai berikut (Revi, Solikhun, & Safii, 2018) :

#### Single Perceptron

ANN yang bentuknya paling kecil disebut single perceptron yang hanya terdiri dari sebuah neuron, seperti terlihat pada **Gambar 2.9** (Priyanto, Zarlis, Mawengkang, & Efendi, 2019).

Single perceptron hanya terdiri dari 1 lapisan input dan 1 lapisan output. Setiap neuron yang terdapat di dalam lapisan input selalu terhubung dengan setiap neuron yang terdapat pada lapisan output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi (Revi, Solikhun, & Safii, 2018).

A picture containing text, night sky

Description automatically generated

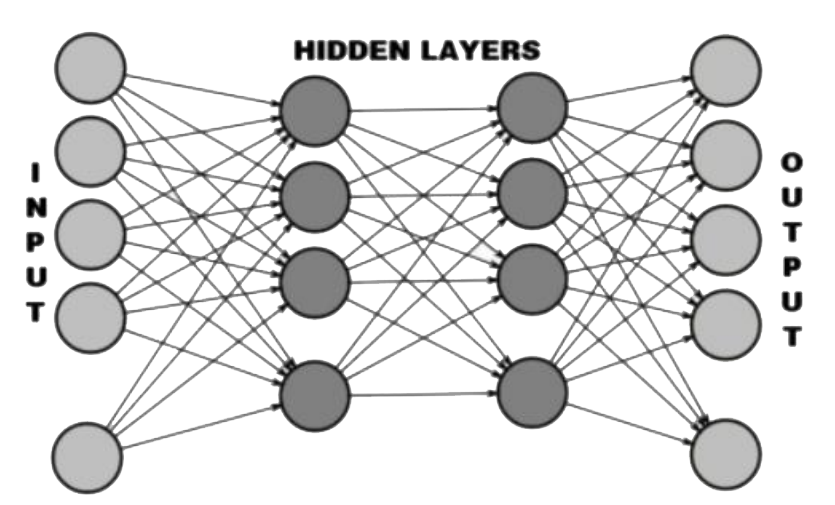
(Sumber : Putra J. G., 2020)

**Gambar 2.10. Single Perceptron**

Secara matematis pada **Gambar 2.10** terdapat feature vector *x* yang menjadi input bagi neuron tersebut. Feature vector merepresentasikan suatu data point, event atau instance. Neuron akan memproses input *x* melalui perhitungan jumlah perkalian antara nilai input dan *synapse weight*, yang dilewatkan pada fungsi non-linear. Pada training, yang dioptimasi adalah nilai *synapse weight* (learning parameter). Selain itu, terdapat juga bias *b* sebagai kontrol tambahan. Output dari neuron adalah hasil fungsi aktivasi dari perhitungan jumlah perkalian antara nilai *input* dan *synapse weight*. Ada beberapa macam fungsi aktivasi, misal *step function, sign function, rectifier dan sigmoid function* (Priyanto, Zarlis, Mawengkang, & Efendi, 2019).

#### Multilayer Perceptron (MLP)

Multilayer merupakan bentuk lapisan perceptron yang digabungkan, dengan menambahkan lebih banyak layer dan neuron tiap layer. Multilayer Perceptron (MLP) sendiri merupakan arsitektur yang paling banyak digunakan untuk jaringan saraf (Sen, Sugiarto, & Rochman, 2020).



(Sumber : Sen, Sugiarto, & Rochman, 2020)

Gambar 2.11. Deep Neural Network Multilayer Perceptron

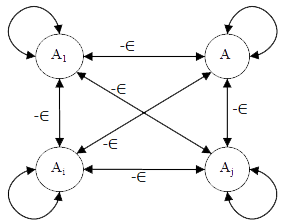
Multilayer Perceptron memiliki ciri khas tertentu yaitu memiliki 3 jenis lapisan yakni lapisan input, lapisan output, dan lapisan tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama (Revi, Solikhun, & Safii, 2018).

MLP dilatih dari data pelatihan melalui proses yang disebut backpropagation. Proses ini dapat digambarkan sebagai cara untuk memperbaiki kesalahan secara progresif segera setelah terdeteksi. Pada awalnya, semua bobot ditetapkan secara acak. Kemudian jaringan diaktifkan untuk setiap input dalam set pelatihan: nilai disebarkan ke depan (forward propagation) dari tahap input melalui tahap tersembunyi ke tahap output di mana prediksi dibuat (Sen, Sugiarto, & Rochman, 2020).

Karena nilai real yang diamati dalam set pelatihan diketahui, maka memungkinkan untuk menghitung kesalahan yang dibuat dalam prediksi. Proses kerja utama dalam backtracking adalah melakukan alur kembali dari output menuju input dengan menggunakan algoritma pengoptimal yang tepat, seperti gradient descent, untuk menyesuaikan bobot (weight) jaringan saraf dengan tujuan mengurangi kesalahan (Sen, Sugiarto, & Rochman, 2020).

#### Competitive Layer Net

Bentuk lapisan kompetitif merupakan jaringan saraf tiruan yang sangat besar. Interkoneksi antar neuron pada lapisan ini tidak ditunjukkan pada arsitektur seperti jaringan yang lain. Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif atau sering pula disebut dengan prinsip winner takes all atau yang menanglah yang mengambil semua bagiannya (Sadli, 2018).



(Sumber : Sadli, 2018)

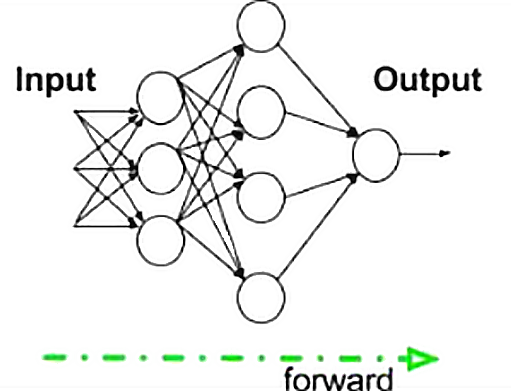
Gambar 2.12. Competitive Layer Net

### Teknik Training Jaringan Syaraf Tiruan

Proses Jaringan Syaraf Tiruan proses training terbagi menjadi dua yaitu Forward Pass dan Backpropagation (Winoto, 2020). Antara lain sebagai berikut :

#### Forward Propagation / Forward pass

Pemrosesan dari layer input ke hidden layer dan kemudian ke lapisan output adalah disebut forward propagation (Sen, Sugiarto, & Rochman, 2020). Forward propagation adalah proses mengolah sinyal input dengan bobot yang tersedia pada saat melewati Hidden Layer hingga sampai ke Output Layer. Setiap Layer memiliki fungsi aktivasi yang berfungsi untuk mengaktifkan atau tidak suatu sinyal (Winoto, 2020).



(Sumber : Sen, Sugiarto, & Rochman, 2020)

Gambar 2.13. Forward Propagation

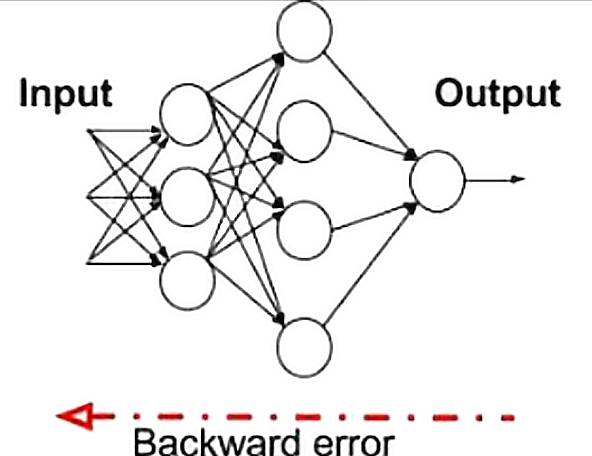
##### Algoritma Training Forward Propagation

Algoritma pembelajaran untuk JST forward propagation adalah sebagai berikut (Azise, Andono, & Pramunendar, 2019) :

1. Masing-masing input menerima sinyal , kemudian melanjutkan ke semua unit pada *hiden layer*
2. Pada masing-masing hiden layer, menjumlahkan bobot sinyal input dengan persamaan : Untuk menghitung sinyal output pada fungsi aktivasi menggunakan persamaan
3. Masing-masing unit output menjumlahkan bobot sinyal input dengan persamaan Untuk menghitung nilai outputdengan menerapkan fungsi aktivasi menggunakan persamaan

#### Backward propagation / Backward Pass

Backpropagation merupakan metode pelatihan dari Artificial Neural Network yang menggunakan arsitektur multilayer dengan algoritma pembelajaran suprvised. Metode ini bertujuan untuk melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola dalam pelatihan dan memberikan respon yang benar terhadap pola input yang hampir sama dengan pola yang dipakai selama pelatihan. Metode ini juga melakukan dua tahapan, yaitu feedforward atau perhitungan maju dan backward propagation atau perhitungan mundur (Pandji, Indwiarti, & Rohmawati, 2019).



(Sumber : Sen, Sugiarto, & Rochman, 2020)

Gambar 2.14. Backward Propagation

##### Tahap Training Backward Propagation

Secara garis besar, training jaringan dengan dengan metode backpropagation meliputi tiga tahap (Satria, 2018) :

###### Tahap maju (feedforward)

Tahap feedforward yang dimaksud adalah proses pengolahan input dari pola input training pada input layer sampai respon yang dihasilkan mencapai output layer (Satria, 2018).

###### Tahap perhitungan error propagasi balik (backpropagation of error)

Respon yang dihasilkan pada output layer akan dibandingkan dengan output target, kemudian dihitung errornya. Bila kriteria untuk kondisi berhenti (stopping condition) belum terpenuhi, maka dilanjutkan ke tahap ketiga (adjustment of the weights and biases). Namun jika kondisi berhenti sudah terpenuhi, maka proses perhitungan berhenti (Satria, 2018).

###### Tahap pembaharuan bobot dan bias

Kondisi ini terjadi jika output yang diharapkan tidak sesuai, maka jaringan akan bergerak mundur (backward) dari output layer menuju ke input layer dan akan melakukan update bobot dan bisa serta mengulangi proses dari tahap 1. Backpropagation merupakan algoritma pembelajaran yang biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya (Satria, 2018).

##### Algoritma Training Backward Propagation

Algoritma pembelajaran untuk JST Backpropagation adalah sebagai berikut (Muflih, Sunardi, & Yudhana, 2019) :

1. Inisiasi bobot (tetapkan dengan nilai acak kecil)
2. Selama syarat kondisi false kerjakan langkah 2-9
3. Untuk setiap pasangan yang akan dilakukan pembelajaran, kerjakan langkah 3-8
4. Setiap unit input menerima sinyal dan meneruskan sinyal ke semua unit pada lapisan tesembunyi
5. Setiap unit tersembunyi menjumlahkan bobot sinyal input dengan persamaan berikut: , Hitung sinyal output dengan fungsi aktivasinya Kirimkan sinyal ini ke semua unit pada lapisan output.
6. Setiap unit output menjumlahkan input terbobotnya Hitung sinyal output dengan fungsi aktivasinya
7. Setiap unit output menerima pola target sesuai dengan pola input pelatihan, kemudian hitung error seperti persamaan Hitung suku koreksi bobot (digunakan untuk perbaruan wjk) Hitung suku koreksi bias (digunakan untuk perbaruan wok) Kirimkan ke unit-unit dilapis bawahnya
8. Setiap unit tersembunyi menjumlahkan delta inputnya (dari unit-unit yang berada pada lapis diatasnya) Hitung suku informasi error Hitung suku koreksi bobot (untuk perbaruan vij) Hitung suku koreksi bias (untuk perbaruan voj) Perbaruan bobot dan bias
9. Setiap unit output perbarui bobot-bobot dan biasnya :
10. Setiap unit tersembunyi perbarui bobot-bobot dan biasnya :
11. Uji syarat berhenti, Fungsi aktivasi sigmoid

## Long Short-Term Memory

*Long Short-Term Memory (LSTM)* merupakan pengembangan dari *Recurrent Neural Network (RNN)* dengan mengatasi salah satu kekurangan *RNN* yaitu kemampuan pengelolaan informasi dalam periode yang lama yang mana dilakukan modifikasi pada Recurrent Neural Network (RNN) dengan memberi memory cell untuk dapat menyimpan informasi dalam waktu yang lama. Diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997, *LSTM* banyak dipilih untuk prediksi berbasis waktu atau *time series* karena dikenal lebih unggul dan handal dalam melakukan prediksi dalam waktu lama dibanding algoritma lain (Zahara, Sugianto, & Ilmiddafiq, 2019; Manaswi, 2018).

A screenshot of a video game

Description automatically generated with medium confidence  
Diagram

Description automatically generated

(Sumber : Manu, 2021)

Gambar 2.15. Arsitektur LSTM berisi empat layer yang saling berinteraksi.

Kunci LSTM adalah *cell state*, garis horizontal yang melewati bagian atas diagram keadaan sel seperti ban berjalan. Ini berjalan lurus ke bawah seluruh rantai, memiliki beberapa linier kecil interaksi (Ghosh, Bose, Maji, Debnath, & Sen, 2019). Untuk setiap sel memori memiliki tiga layer sigmoid dan satu layer tanh (Qiu, Wang, & Zhou, 2019).

A picture containing text, clock

Description automatically generated

(Sumber : Manu, 2021)

Gambar 2.16. Alur Informasi Cell State pada LSTM

Keterangan (Pardede & Ibrahim, 2020) :

: *Cell state*

: Nilai *Cell state* sebelum order ke *t*

Pada **Gambar 2.16**, garis horizontal yang melalui bagian atas diagram dikenal sebagai *cell state* (, ). Ini bertindak seperti ban berjalan yang berjalan di seluruh jaringan. Ini membawa informasi dari sel sebelumnya ke saat ini dan seterusnya (Hiransha, Gopalakrishnan, Menon, & Soman, 2018). Kemampuan untuk menambah atau menghapus informasi ke cell state dikendalikan oleh struktur yang disebut *gate*. *Gate* digunakan untuk secara opsional membiarkan informasi lewat. Informasi yang di saring melalui struktur *gate* yang akan mempertahankan dan memperbarui *cell state* memori (Ghosh, Bose, Maji, Debnath, & Sen, 2019; Qiu, Wang, & Zhou, 2019).

Graphical user interface, application

Description automatically generated

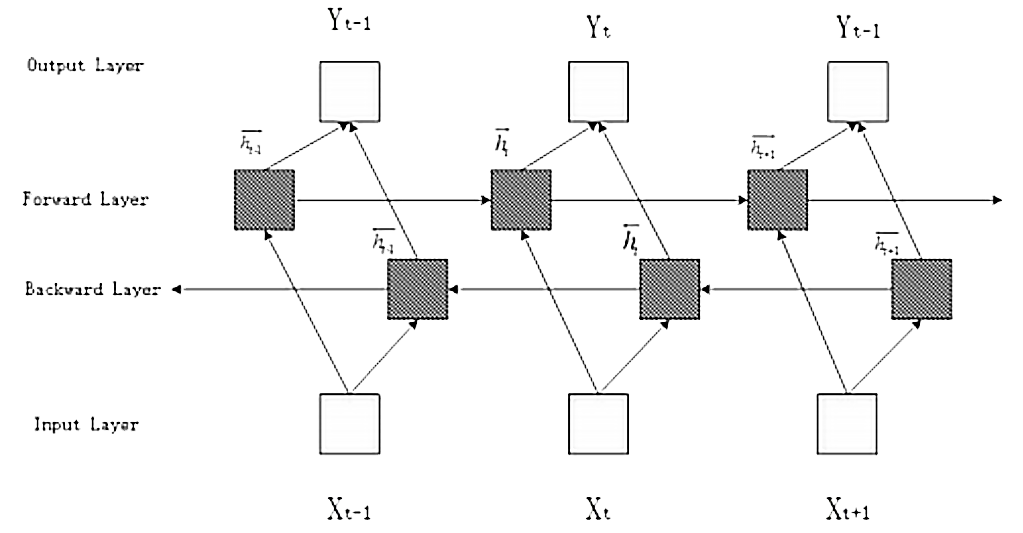
(Sumber : Manu, 2021)

Gambar 2.17. Layer sigmoid mengeluarkan angka antara nol dan satu.

Pada **Gambar 2.17** layer sigmoid mengeluarkan angka antara 0 dan 1, menggambarkan berapa banyak dari setiap komponen yang harus dilewati. Nilai 0 berarti “jangan biarkan apa pun lewat”, sedangkan nilai 1 berarti “biarkan semuanya lewat!” (Ghosh, Bose, Maji, Debnath, & Sen, 2019)*.*

### Proses Training dan Testing Pada LSTM

LSTM disebut juga sebagai jaringan saraf dengan arsitektur yang mudah beradaptasi, sehingga bentuknya dapat disesuaikan, tergantung pada aplikasinya. Long Short Term Memory merupakan turunan dari metode RNN (Recurrent Neural Network). Recurrent Neural Network merupakan jaringan saraf berulang yang didesain khusus untuk menghandle data berurutan (sequence data) (Wiranda & Sadikin, 2019).



(Sumber : Santosa, Bijaksana, & Romadhony, 2021)

Gambar 2.18. Ilustrasi Training Model LSTM dengan Training Backpropagation

#### Training Model LSTM

Beberapa tahapan dalam proses training model LSTM dengan backpropagation adalah (Arfan & ETP, 2019) :

1. Inisialisasi bobot awal
2. Input data training
3. Perhitungan LSTM pada setiap input yaitu dimulai dengan forget gates, fungsi inputgates,fungsi cell gates dan yang terakhir fungsi output gates.
4. Perhitungan standard deviasi RMSE untuk mendapatkan nilai seilisih antara nilai LSTM dengan target output.
5. Perhitungan gradient untuk menentukan nilai bobot supaya hasil loss mendekati 0 dengan menggunakan Backpropagation Through Time (BTTP).
6. Setelah mendapatkan nilai gradient, maka dilanjutkan dengan persamaan fungsi optimasi dan update bobot.
7. Kembali ke langkah dua sebanyak epoch yang telah ditentukan.

Bedasarkan Penjabaran di atas, Pembentukan model LSTM diawali dengan menginisialisasi paramater yang dibutuhkan yaitu hidden layer (lapisan tersembunyi), units (memori sel), epoch (putaran), dan batch size (jumlah sampel data). Setelah model dibentuk maka data akan dilatih dengan melewati mekanisme gates pada LSTM. Data akan dilatih terus hingga mencapai batas error yang diinginkan dengan penentuan serta pengubahan paramater yang digunakan (Agusta, Ernawati, & Muliawati, 2021).

Ketika data sudah mencapai target yang diinginkan, proses iterasi akan berhenti dan berikutnya model akan diuji dengan data pengujian atau dapat mengulang kembali proses pelatihan. Proses iterasi ini juga diolah dengan menggunakan fungsi optimasi dan dropout. Optimasi berguna untuk menentukan bobot optimal dan mengurangi kesalahan sehingga dapat memaksimalkan keakuratan model. Sedangkan dropout berguna untuk mencegah terjadinya overfitting pada model (Agusta, Ernawati, & Muliawati, 2021).

#### Testing Model LSTM

Pengujian / Testing ini dengan mengambil data testing kemudian dibandingkan dengan data yang dihasilkan dengan metode LSTM pada rentang waktu yang ditentukan dengan metode akurasi yang digunakan menggunakan standard deviasi RMSE (Arfan & ETP, 2019).

### Fungsi Aktivasi Pada LSTM

Fungsi aktivasi sangat berperan dalam mengaktifkan setiap neuron pada *jaringan saraf tiruan* serta menentukan keluaran dari suatu jaringan saraf tiruan (Susilawati & Muhathir, 2019). Berikut adalah beberapa fungsi aktivasi yang di gunakan dalam penelitian ini :

#### Sigmoid (σ)

Fungsi aktivasi *sigmoid* merupakan fungsi non-linear. *Input* untuk fungsi aktivasi ini berupa bilangan real dan *output* dari fungsi aktivasi ini memiliki range antara 0 sampai 1 (Suhermi, Suhartono, Dana, & Prastyo, 2018). Berikut ini perhitungan dari fungsi aktivasi sigmoid :

Dimana :

: data input

: konstanta matematika (2,718281828…)

Fungsi sigmoid mentransformasi range nilai dari *input* x menjadi antara 0 dan 1. Jika *inputnya* sangat negatif, maka keluaran yang didapatkan adalah 0, sedangkan jika *input* sangat positif maka nilai keluaran yang didapatkan adalah 1. Fungsi ini memiliki kekurangan yaitu dapat mematikan *gradient*, ketika aktivasi dari neuron mengeluarkan nilai yang berada pada range 0 atau 1, dimana gradient di wilayah ini hampir bernilai 0. Kemudian output dari sigmoid tidak zero-centered (Suhermi, Suhartono, Dana, & Prastyo, 2018).



(Sumber : Deng, Tong, Lan, & Huang, 2020)

Gambar 2.20. Illustrasi Sigmoid

#### Hyperbolic (Tanh)

Fungsi aktivasi *Tanh* merupakan fungsi non-linear. *Input* untuk fungsi aktivasi ini berupa bilangan real dan output dari fungsi tersebut memiliki range antara -1 sampai 1 (Suhermi, Suhartono, Dana, & Prastyo, 2018). Berikut ini perhitungan dari fungsi aktivasi *tanh* :

Dimana :

: data input

: konstanta matematika (2,718281828…)

Sama seperti fungsi sigmoid, fungsi ini memiliki kekurangan yaitu dapat mematikan *gradient*, akan tetapi fungsi ini juga memiliki kelebihan yaitu *output* yang dimiliki fungsi *Tanh* merupakan zero-centered. Dalam pengaplikasiannya fungsi *Tanh* lebih menjadi pilihan jika dibandingkan dengan fungsi sigmoid. Fungsi Perlu diketahui fungsi *tanh* merupakan pengembangan dari fungsi Sigmoid (Suhermi, Suhartono, Dana, & Prastyo, 2018).



(Sumber : Flywind, 2018)

Gambar 2.21. Ilustrasi Tanh

### Langkah Training Gates Pada LSTM

LSTM memiliki tiga di antaranya gerbang, untuk melindungi dan mengontrol cell state, Struktur gerbangnya mencakup *forget gate*, *input gate.* dan *output gate*. *Gate* terdiri dari layer jaring saraf sigmoid dan operasi perkalian pointwise (Ghosh, Bose, Maji, Debnath, & Sen, 2019; Qiu, Wang, & Zhou, 2019). Berikut ialah keterangan setiap gate yang ada pada **Gambar 2.15** :

#### Forget Gate

Pada *forget gate* informasi pada setiap data *input* yang akan diolah dan dipilih data mana saja yang akan disimpan atau dibuang pada memory cells. Fungsi aktivasi yang digunakan pada *forget gate* ini adalah fungsi aktivasi sigmoid. Dimana hasil keluarannya antara 0 dan 1. Jika keluarannya adalah 1 maka semua data akan disimpan dan sebaliknya jika keluarannya 0 maka semua data akan dibuang (Aldi, Jondri, & Aditsania, 2018). Dengan rumus seperti pada **Gambar 2.22** :

Diagram

Description automatically generated

(Sumber : Manu, 2021)

Gambar 2.22. Persamaan Forget Gate pada LSTM

Keterangan (Pardede & Ibrahim, 2020) :

: *Forget gate*

: Fungsi Aktivasi *Sigmoid*

: Nilai *Weight* untuk *Forget gate*

: Nilai *output* sebelum order ke *t*

: Nilai *input* pada order ke *t*

: Nilai bias pada *Forget gate*

Pada **Gambar 2.22** LSTM memutuskan informasi apa yang akan dibuang dari *cell state*. Keputusan ini dibuat oleh layer sigmoid yang disebut "layer forget gate." (). Terlihat pada dan dan nilai *output* antara angka 0 dan 1 untuk setiap angka dalam *cell state* pada **Gambar 2.16**. *Output* dari 1 mewakili 'sepenuhnya simpan ini' sementara 0 mewakili 'singkirkan ini sepenuhnya' (Boruah & Barman, 2018).

#### Input Gate

Pada *input gate* terdapat dua gates yang akan dilaksanakan, pertama akan diputuskan nilai mana yang akan diperbarui menggunakan fungsi aktivasi sigmoid. Selanjutnya fungsi aktivasi tanh akan membuat vector nilai baru yang akan disimpan pada memory cell (Aldi, Jondri, & Aditsania, 2018). Dengan rumus seperti pada **Gambar 2.23** :

Diagram

Description automatically generated

(Sumber : Manu, 2021)

Gambar 2.23. Persamaan yang melewati Input Gate pada LSTM

Keterangan (Pardede & Ibrahim, 2020) :

: *Input gate*

: Nilai baru yang dapat ditambahkan ke *cell state*

: Fungsi Aktivasi *Sigmoid*

: Fungsi Aktivasi *Tanh*

: Nilai *Weight* untuk *Input gate*

: Nilai *Weight* untuk *Cell state*

: Nilai *output* sebelum order ke *t*

: Nilai *input* pada order ke *t*

: Nilai bias pada *Input gate*

: Nilai bias pada *cell state*

Pada langkah berikutnya di **Gambar 2.23** LSTM memutuskan informasi apa yang akan disimpan dari *cell state*. Pertama layer *sigmoid* yang disebut "layer *input gate*" () memutuskan nilai mana yang akan diperbarui. Setelah itu, layer membuat vektor nilai kandidat baru, , yang dapat ditambahkan ke state (Ghosh, Bose, Maji, Debnath, & Sen, 2019).

#### Cell State / Memory State

Pada *cell state* gates akan mengganti nilai pada memory cell sebelumnya dengan nilai memory cell yang baru. Dimana nilai ini didapatkan dari menggabungkan nilai yang terdapat pada forget gate dan input gate (Aldi, Jondri, & Aditsania, 2018). Dengan rumus seperti pada **Gambar 2.24** :

Diagram

Description automatically generated with medium confidence

(Sumber : Manu, 2021)

Gambar 2.24. Persamaan Memperbaharui Cell State pada LSTM

Keterangan (Pardede & Ibrahim, 2020) :

: *Cell state*

: *Forget gate*

: Nilai *Cell state* sebelum order ke *t*

: *Input gate*

: Kandidat konteks baru yang dapat ditambahkan ke *cell state*

Pada langkah selanjutnya di **Gambar 2.24**, kedua layer di gabungkan digabungkan untuk membuat pembaruan ke *cell state*. Pada Langkah inilah nilai *cell state* lama (), akan di perbaharui ke nilai dari *cell state* baru () dimana LSTM akan mengalikan *cell state* lama dengan () kemudian ditambahkan dengan () . Ini adalah nilai kandidat baru, yang diskalakan berdasarkan seberapa banyak kami memutuskan untuk memperbarui setiap nilai negara bagian. (Ghosh, Bose, Maji, Debnath, & Sen, 2019).

#### Output Gate

Pada *output gate* terdapat dua gate yang akan dilaksanakan, pertama akan diputuskan nilai pada bagian memory cell mana yang akan dikeluarkan dengan menggunakan fungsi aktivasi sigmoid. Selanjutnya akan ditempatkan nilai pada memory cell dengan menggunakan fungsi aktivasi tanh. Terakhir kedua gate tersebut di dikalikan sehingga menghasilkan nilai yang akan dikeluarkan (Aldi, Jondri, & Aditsania, 2018). Dengan rumus seperti pada **Gambar 2.25** :

A picture containing text, clock

Description automatically generated

(Sumber : Manu, 2021)

Gambar 2.25. Persamaan melewati Output Gate pada LSTM

Keterangan (Pardede & Ibrahim, 2020) :

: *Output gate*

: *Cell state*

: Fungsi Aktivasi *Sigmoid*

: Fungsi Aktivasi *Tanh*

: Nilai *Weight* untuk *Output gate*

: Nilai *output* sebelum order ke *t*

: Nilai *input* pada order ke *t*

: Nilai bias pada *Output gate*

: Nilai *output* pada order ke *t*

Terakhir pada **Gambar 2.25** adalah tahap dimana perlu memutuskan apa yang akan hasilkan. *Output* akan didasarkan pada *cell state*, tetapi akan menjadi versi yang difilter. Pertama, kita menjalankan layer sigmoid yang memutuskan bagian mana dari *cell state* yang akan kita hasilkan. Kemudian, *cell state* di tempatkan melalui *tanh* (untuk mendorong nilai menjadi antara -1 dan 1) dan mengalikannya dengan *output* *gate* layer *sigmoid*, sehingga hanya akan menampilkan bagian yang putuskan (Ghosh, Bose, Maji, Debnath, & Sen, 2019).

### Loss Function

Kinerja pembelajaran diukur dari optimalnya nilai suatu fungsi seperti minimalnya nilai loss dan error. Loss adalah ukuran seberapa dekat atau berbeda model yang dihasilkan dengan data asli, sedangkan error merupakan salah satu cara untuk menghitung loss. Nilai dari loss dan error tergantung dari parameter pembelajaran yang digunakannya. Kekurangan dari machine learning adalah membutuhkan data yang banyak untuk proses pembelajarannya (Putra J. G., 2020). Dalam Penelitian ini fungsi pengukuran nilai error yang akan di gunakan adalah Root Mean Squared Error (RMSE).

#### Root Mean Squared Error (RMSE)

Tingkat akurasi ditujukan selain secara visual dalam bentuk grafik juga dalam bentuk kuantitatif dengan mengukur nilai RMSE (Root Mean Square Error) RMSE berhubungan dengan variasi sebaran frekuensi (frequency distribution) dari besar kesalahan yang diperoleh, tapi tidak dengan variasi kesalahan. (Karno, Hastomo, Nisfiani, & Lukman, 2020). Berikut ini perhitungan dari RMSE :

Dimana (Aprian, Azhar, & Nastiti, 2020) :

: adalah jumlah data

: adalah data prediksi pada waktu ke i

: adalah data asli pada waktu i

Uji Validitas menggunakan Root Mean Squared Error (RMSE) dilakukan untuk mengukur hasil akurasi pengujian. RMSE merupakan salah satu contoh parameter yang biasa digunakan sebagai indikator untuk mengukur dan membandingkan kemiripan hasil prediksi dan data asli (Aprian, Azhar, & Nastiti, 2020).

### Batch dan Epochs

*Epoch* adalah ketika seluruh dataset sudah melalui proses training pada *Neural Network* sampai dikembalikan ke awal dalam satu putaran. Dalam *Neural Network* satu *epoch* itu terlalu besar dalam proses pelatihan karena seluruh data diikutkan kedalam proses training sehingga akan membutuhkan waktu cukup lama. Agar mempermudah dan mempercepat proses training, biasanya data rate dibagi per batch (Batch Size).

*Batch* size merupakan jumlah sampel data yang akan disebarkan dalam sebuah *neural network*. *Batch* size efisien secara komputasi ketika berhadapan dengan dataset yang besar. Penentuan nilai dari batch size biasanya tergantung peneliti dengan melihat banyak sampel (Thohari & Hertantyo, 2018).

### Optimasi Adaptive Moment Estimation (ADAM)

Adaptive Moment Estimation atau yang biasa disingkat Adam adalah sebuah metode optimasi. Adam melakukan penyimpanan nilai gradient-gradient di perhitungan sebelumnya untuk menghitung rata-rata nilai gradient pada momentum pertama maupun momentum kedua. Dari kedua rata-rata tersebut di adaptasi parameternya untuk digunakan sebagai pembaharuan nilai bobot (Winoto, 2020).

Adam menyimpan rata-rata gradien proses sebelumnya secara eksponensial. Nilai learning rate standar pada Adam adalah 0,001. Rumus perhitungan optimasi Adam ditunjukkan pada persamaan berikut (Miranda, Novamizanti, & Rizal, 2020) :

Dimana (Miranda, Novamizanti, & Rizal, 2020) :

: adalah parameter hasil pembaruan

: adalah parameter hasil pembaruan sebelumnya

: adalah *learning rate*

: adalah gradien kuadrat momen orde pertama

: adalah gradien kuadrat momen orde kedua

: merupakan scalar kecil untuk mencegah pembagian dengan 0

### Normalisasi dan Denormalisasi

Dalam rangka meminimalkan error perlu dilakukan normalisasi. Normalisasi berfungsi untuk menghindari terjadinya berbagai anomali data dan tidak konsistensinya data. Normalisasi ini juga bertujuan untuk merubah ukuran data menjadi lebih kecil tanpa harus merubah data asli. Teknik normalisasi yang digunakan adalah min-max scaling. Teknik ini digunakan untuk mengatasi perbedaan nilai yang cukup besar antar dataset. Cara kerjanya yakni dengan mengubah nilai pada data aktual menjadi nilai dengan skala (0,1) tanpa mengubah informasi yang ada. Teknik normalisasi dengan min-max scaling memiliki persamaan sebagai berikut (Aldi, Jondri, & Aditsania, 2018).

Dimana :

: Data hasil normalisasi

: Data asli

: Nilai minimum dari data *x*

: Nilai maximum dari data *x*

Denormalisasi adalah proses pengembalian data hasil normalisasi ke dalam data asli atau data sebenarnya. Hal tersebut dilakukan guna melihat hasil prediksi dengan cara membandingkan dengan data sebenarnya (Aldi, Jondri, & Aditsania, 2018).

Dimana :

: Nilai dari data normalisasi

: Hasil output

: Nilai minimal data actual keseluruhan

: Nilai maksimal data actual keseluruhan

## Prediksi / Forecasting

Menurut Sucipto & Syaharuddin Prediksi / Peramalan (forecasting) adalah kegiatan mengestimasi apa yang akan terjadi pada masa yang akan datang. Peramalan diperlukan karena adanya kesenjangan waktu (timelag) antara kesadaran dibutuhkannya suatu kebijakan baru dengan waktu pelaksanaan kebijakan tersebut (Sucipto & Syaharuddin, 2018).

Menurut Putro, Furqon, & Wijoyo prediksi merupakan suatu proses untuk meramalkan atau memperkirakan suatu variabel di masa yang akan datang. Dalam kasus prediksi biasanya data yang sering digunakan adalah data kuantitatif. Prediksi tidak harus menghasilkan suatu jawaban yang pasti kejadian, melainkan berusaha untuk mencari jawaban yang sedekat mungkin dengan kejadian yang akan terjadi (Putro, Furqon, & Wijoyo, 2018).

Secara umum, ada dua jenis prediksi yaitu kualitatif dan kuantitatif. Prediksi kualitatif merupakan prediksi yang bersifat subjektif, hal ini karena didasarkan pada pengalaman empiris, intuisi pengambilan keputusan dan emosi manusia.Sedangkan, prediksi kuantitatif merupakan prediksi yang bersifat objektif sebab didasarkan pada data aktual dan diolah menggunakan metode tertentu (Surtiningsih, Furqon, & Adinugroho, 2018).

Jenis sesuai yang di jelaskan oleh para ahli sebelumnya prediksi yang digunakan dalam penelitian adalah prediksi kuantitatif. Dikarenakan data yang digunakan adalah data dari masa lalu berupa angka dengan runtutan waktu.

## Penelitian Terdahulu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Pengarang, Tahun | Metode | Data | Hasil |
| 1 | (Moghar & Hamiche, 2020) | Long Short Term Memory | Dokumentasi saham NYSE  (GOOGL dan NKE) dari website yahoo finance | Setelah melatih NN, hasil pengujian peneliti menunjukkan hasil yang berbeda, jumlah epoch serta panjang data memiliki pengaruh yang signifikan terhadap hasil pengujian. Setelah mengamati data peneliti, peneliti dapat melihat bahwa pada awalnya data kurang stabil dan memiliki nilai yang lebih rendah, setelah NKE mulai mengintip nilai yang lebih besar, aset menjadi lebih fluktuatif, maka sifat aset ini berubah. Model peneliti telah kehilangan jejak harga pembukaan sekitar 600 hingga 700 hari pengujian yang sesuai dengan perubahan sifat data. |
| 2 | (Karno, Hastomo, Nisfiani, & Lukman, 2020) | Long Short Term Memory berbasis *Gated Recurrent Units* | Dokumentasi saham bank di Indonesia dari website yahoo finance | Dengan menggunakan paket library seborn di python, hasil perhitungann korelasi dapat ditunjukkan di matrik heatmap dalam bentuk numerik dan tingkatan warna. Hasil numerik korelasi berupa bilangan dengan rentang -1 dan 1, dimana nilai 1 menunjukan hubungan yang kuat, dan nilai 0 menunjukan hubungan yang rendah antara dua data. Jadi dapat dilihat bahwa RMSE mampu meredam perubahan kesalahan yang besar, sebaliknya MSE mampu melihat perubahan kesalahan yang kecil. GRU berupa sel yang berisi hanya 2 gate dengan rangkaian yang lebih sederhana dibandingkan dengan LSTM. |
| 3 | (Boruah & Barman, 2018) | Long Short Term Memory | Identifikasi kalimat pada Novel *Bhanumati* ditulis oleh Assamese  scholar *Padmanath Gohainbaruah* | Epoch dengan akurasi maksimum untuk model yang berbeda diberikan. Akurasi rata-rata yang terlihat setelah setiap seribu epoch menampilkan kerugian rata-rata setelah setiap seribu epoch dari Uji 1, yang memiliki akurasi maksimum di antara model berbeda yang memiliki konfigurasi berbeda. Tetapi untuk bahasa Assam yang ditranskripsi secara fonetis pada peneliti dapat melihat bahwa akurasi rata-rata meningkat ketika peneliti meningkatkan jumlah neuron dari 128 menjadi 256 di setiap lapisan. Tetapi peningkatan lebih lanjut dari neuron menjadi 512 di setiap lapisan menurunkan akurasi. |
| 4 | (Supriyadi, 2019) | Long Short Term Memory | Observasi sinoptik Stasiun  Meteorologi Maritim Tanjung Priok | Metode deep learning LSTM digunakan untuk memprediksi parameter cuaca, seperti suhu udara, kelembaban, kecepatan angin, dan tekanan udara. Sedangkan jumlah datanya dibagi dua menjadi training data dan test data dengan rasio 9:1.pada bulan Januari 2019. Diperoleh RMSE parameter suhu udara, kelembaban, kecepatan angin, dan tekanan udara nilainya semakin baik ketika menggunakan deep learning LSTM dengan update dibandingkan LSTM tanpa update. Dari parameter cuaca tersebut hanya parameter suhu dan kelembaban udara yang mengalami pertambahan RMSE seiring bertambahnya waktu. |
| 5 | (Poornima & Pushpalatha, 2019) | LSTM dengan *Weighted Linear Units* | Dokumentasi kumpulan data *China Meteorological Administration* | Dalam penelitian tersebut menyajikan Metode Long Short-Term Memory dengan Weighted Linear Units untuk memprediksi curah hujan. Neural Network dilatih dan diuji menggunakan dataset standar curah hujan. Jaringan yang dilatih akan menghasilkan atribut prediksi curah hujan. Parameter yang dipertimbangkan untuk evaluasi kinerja dan efisiensi model prediksi curah hujan yang diusulkan adalah Root Mean Square Error, akurasi, jumlah epoch, loss, dan learning rate. |
| 6 | (Zahara, Sugianto, & Ilmiddafiq, 2019) | Long Short Term Memory | Dokumentasi secara online dari website *Dinas Perdagangan dan Perindustrian Provinsi Jawa Timur* | Pembangunan model prediksi dilakukan di lingkungan cloud computing Amazon Web Service tipe EC2. IHK prediksi pada bulan Desember 2016 yaitu 133.98, mempunyai nilai prediksi yang paling mendekati nilai IHK aktual yaitu 133.98 Dari hasil pengujian nilai RMSE tiap algoritma optimasi LSTM yang masih tergolong besar, terlihat bahwa metode LSTM belum bisa disebut metode yang maksimal dalam melakukan prediksi IHK. |
| 7 | (Aldi, Jondri, & Aditsania, 2018) | Long Short Term Memory | Dokumentasi dari website *blockchain.info* | Pada penelitian tersebut dibangun model LSTM untuk memprediksi harga Bitcoin dengan pengujian parameter komposisi data, jumlah pola time series, jumlah hidden neuron dan max epoch. Pada pengujian tersebut didapatkan hasil yang terbaik yaitu dengan komposisi data latih 70% dan data uji 30%, parameter 1 pola time series, jumlah  25 neuron hidden, dan max epoch adalah 100 dengan akurasi rata-rata pada data latih 95.36% dan data testing 93.5%. |
| 8 | (Putra, Osmond, & Ansori, 2020) | Long Short Term Memory | Dokumentasi data mentah dari Dinas Perindustrian dan Perdagangan Provinsi Jawa Barat | Ada 4 parameter yang akan diuji untuk setiap komoditas, yang terdiri dari Nilai Epoch, Nilai Lookback, Jumlah Hidden Layer, dan Nilai Train Batch. Jumlah hidden layer adalah banyaknya layer LSTM pada model, terdapat Single-Layer dan Multi-Layer dimana Multi-Layer sendiri terdiri dari 2 dan 3 hidden layer. Tujuan pengujian prediksi disini yaitu untuk menguji tingkat akurasi dan kecocokan model LSTM dalam memprediksi harga masa depan dengan menggunakan konfigurasi dari hasil pada pengujian model. |
| 9 | (Ghosh, Bose, Maji, Debnath, & Sen, 2019) | Long Short Term Memory | Dokumentasi dari website resmi *Bombay Stock Exchange* | Dalam penelitian tersebut, peneliti menganalisis pertumbuhan perusahaan dari berbagai sektor dan mencoba mencari tahu yang merupakan rentang waktu terbaik untuk memprediksi harga saham di masa depan. Prediksi bisa lebih akurat jika model akan berlatih dengan jumlah data yang lebih banyak mengatur. Kerangka kerja ini secara luas membantu dalam analisis pasar dan prediksi pertumbuhan perusahaan yang berbeda dalam rentang waktu yang berbeda. |
| 10 | (Rizki, Basuki, & Azhar, 2020) | Long Short Term Memory | Dokumentasi dari website BMKG | Aplikasi berhasil memproses prediksi curah hujan kota Malang dengan parameter curah hujan. Jumlah neuron hidden layer dengan hasil paling optimal yaitu dengan 256 neuron hidden layer. Jumlah epoch dengan hasil paling optimal yaitu 150 epoch. Komposisi Data Train dan Data Test dengan hasil yang paling optimal yaitu dengan komposisi data train 50% dan data test 50%. Hal ini karena komposisi data train 50% dan data test 50% memiliki tingkat error yang paling rendah yaitu pada data train sebesar 12.079 dan pada data test sebesar 11.288. |

# daftar pustaka

Agusta, A., Ernawati, I., & Muliawati, A. (2021, August). Prediksi Pergerakan Harga Saham Pada Sektor Farmasi Menggunakan Algoritma Long Short-Term Memory. *JURNAL INFORMATIK, 17*(2).

Aldi, M. W., Jondri, & Aditsania, A. (2018, August). Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. *e-Proceeding of Engineering, 5*, 3548-3555. Diambil kembali dari https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/6739

Alpaydin, E. (2020). *Introduction to Machine Learning, fourth edition.* United States: MIT Press. Diambil kembali dari https://books.google.co.id/books?id=tZnSDwAAQBAJ

Aprian, B. A., Azhar, Y., & Nastiti, V. R. (2020, November 2). Prediksi Pendapatan Kargo Menggunakan Arsitektur Long Short Term Memory. *Jurnal Politeknik Caltex Riau, 6*, 148-157. Diambil kembali dari https://jurnal.pcr.ac.id/index.php/jkt/

Arfan, A., & ETP, L. (2019, August 22). Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term Memory. *Seminar Nasional Teknologi Informasi dan Komunikasi, 3*(1), 225-230.

Ashari, B. S., Otniel, S. C., & Rianto. (2019). PERBANDINGAN KINERJA K-MEANS DENGAN DBSCAN UNTUK METODE CLUSTERING DATA PENJUALAN ONLINE RETAIL. *JURNAL SILIWANGI, 5*(2), 64-67.

Azise, N., Andono, P. N., & Pramunendar, R. A. (2019, July). Prediksi Pendapatan Penjualan Obat Menggunakan Metode Backpropagation Neural Networkdengan Algoritma Genetika Sebagai Seleksi Fitur. *Jurnal Teknologi Informasi-CyberKU,, 15*(2), 142-154. Diambil kembali dari http://research.pps.dinus.ac.id/index.php/Cyberku/article/view/91/82

Azizah, L. M., Umayah, S. F., & Fajar, F. (2018, November). Deteksi Kecacatan Permukaan Buah Manggis Menggunakan Metode Deep Learning dengan Konvolusi Multilayer. *SEMESTA TEKNIKA, 21*, 230-236. doi:10.18196/st.212229

Boruah, A., & Barman, P. P. (2018). A RNN based Approach for next word prediction in Assamese Phonetic Transcription. *Procedia Computer Science, 143*, 117-123. doi:10.1016/j.procs.2018.10.359

cllau, j. (2020, June 20). *Perceptron*. Diambil kembali dari Medium: https://medium.com/@cllaujhohan/perceptron-de822b401b58

Deng, X., Tong, Z., Lan, Y., & Huang, Z. (2020, May 15). Detection and Location of Dead Trees with Pine Wilt Disease Based on Deep Learning and UAV Remote Sensing. *AgriEngineering, 2*, 294-307. doi:10.3390/agriengineering2020019

Ependi, U., Panjaitan, F., & Yulianingsih, E. (2018, October 19). Pemodelan Aplikasi Mobile Sebagai Penunjang Perjalanan Wisata Menggunakan UML Diagram. *Seminar Nasional TeknologiInformasi DanKomunikasi*, 325-330.

Fajarsari, E. J. (2020). Studi Komparatif Prediksi Kestabilan Lereng Dengan Menggunakan Machine Learning (ML). *no. Ml*.

Flywind. (2018, June 23). *Graphs of the hyperbolic sine, cosine, and tangent.* Diambil kembali dari Wikipedia: https://en.wikipedia.org/wiki/File:Sinh\_cosh\_tanh\_graphs\_JCB.jpg

François, C. (2018). *Deep Learning with Python.* Manning. Diambil kembali dari https://books.google.co.id/books?id=Yo3CAQAACAAJ

Ghosh, A., Bose, S., Maji, G., Debnath, N. C., & Sen, S. (2019, September). Stock Price Prediction Using LSTM on Indian. *Proceedings of 32nd international conference, 63*, 101-110. doi:10.29007/qgcz

Gunova, V. (2021, Juni). IMPLEMENTASI DEEP LEARNING PADA SIMULASI AUTONOMOUS DRIVE MENGGUNAKAN AIRSIM. *Journal of Software Engineering, Information and Communication Technology, 2*, 83-92. Diambil kembali dari https://ejournal.upi.edu/index.php/SEICT/article/view/34674

H. M., G. E., Menon, V. K., & S. K. (2018). NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science, 132*, 1351-1362. doi:10.1016/j.procs.2018.05.050

Hasanah, F. N., & Untari, R. S. (2020). *BUKU AJAR REKAYASA PERANGKAT LUNAK.* (M. Suryawinata, M. Nashrullah, & A. Y. Prajati, Penyunt.) Sidoarjo, Jawa Timur, Indonesia: UMSIDA PRESS.

Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. (2020, October 29). Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology, 69*(12), 14413-14423. doi:10.1109/TVT.2020.3034800

Ilahiyah, S., & Nilogiri, A. (2018, August 2). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *Jurnal Sistem & Teknologi Informasi Indonesia, 3*, 49-56. doi:10.32528/justindo.v3i2.2254

Ilham, W., & Fajri, N. (2020, May). PENENTUAN JUMLAH PRODUKSI TAHU DENGAN MENGGUNAKAN METODE FUZZY TSUKAMOTO PADA UKM ABADI BERBASIS WEB. *JURNAL DIGIT, 10*(1), 71~82. doi:http://jurnaldigit.org/index.php/DIGIT/article/viewFile/158/116

Karno, A. S., Hastomo, W., Nisfiani, E., & Lukman, S. (2020, August 6). Optimais Deep Learning untuk Prediksi Data Saham Di Era Pandemi Covid-19. *Seminar Nasional Teknik Elektro dan Informasi*, 43-54. doi:10.26905/santei.v1i1.3098

Kumar, D. P., Amgoth, T., & Annavarapu, C. S. (2019). Machine learning algorithms for wireless sensor networks: A survey. *Information Fusion, 49*, 1-25. doi:10.1016/j.inffus.2018.09.013

Manaswi, N. K. (2018). *Deep Learning with Applications Using Python.* Bangalore, Karnataka, India. doi:10.1007/978-1-4842-3516-4

Manu. (2021, January 30). *A simple overview of RNN, LSTM and Attention Mechanism*. Diambil kembali dari Start it up: https://medium.com/swlh/a-simple-overview-of-rnn-lstm-and-attention-mechanism-9e844763d07b

Miranda, N. D., Novamizanti, L., & Rizal, S. (2020, August 3). CONVOLUTIONAL NEURAL NETWORKPADA KLASIFIKASI SIDIK JARI MENGGUNAKAN RESNET-50. *Jurnal Teknik Informatika (JUTIF)*, 61-68. doi:10.20884/1.jutif.2020.1.2.18

Moghar, A., & Hamiche, M. (2020, April). Stock Market Prediction Using LSTM Recurrent Neural Network. *International Workshop on Statistical Methods and Artificial, 170*, 1168-1173. doi:10.1016/j.procs.2020.03.049

Muflih, G. Z., Sunardi, & Yudhana, A. (2019, July). JARINGAN SARAF TIRUAN BACKPROPAGATION UNTUK PREDIKSI CURAH HUJAN DI WILAYAH KABUPATEN WONOSOBO. *MUST: Journal of Mathematics Education, Science and Technology, 4*(1), 45-46.

Pamungkas, F. S., Prasetya, B. D., & Kharisudin, I. (2020). Perbandingan Metode Klasifikasi Supervised Learning pada Data Bank Customers Menggunakan Python. *PRISMA, Prosiding Seminar Nasional Matematika*, 689-694. Diambil kembali dari https://journal.unnes.ac.id/sju/index.php/prisma/

Pandji, B. Y., Indwiarti, & Rohmawati, A. A. (2019, September). PERBANDINGAN PREDIKSI HARGA SAHAM DENGAN MODEL ARIMA DAN ARTIFICIAL NEURAL NETWORK. *Indonesia Journal on Computing (Indo-JC), 4*(2), 189-198. doi:10.34818/INDOJC.2019.4.2.344

Pardede, J., & Ibrahim, R. G. (2020, December 31). Implementasi Long Short-Term Memory Untuk Identifikasi Berita Hoax Berbahasa Inggris Pada Media Sosial. *J-COSINE, 4*, 179-187. doi:10.29303/jcosine.v4i2.361

Poornima, S., & Pushpalatha, M. (2019, October 31). Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units. *Atmosphere, 10*. doi:10.3390/atmos10110668

Priyanto, D., Zarlis, M., Mawengkang, H., & Efendi, S. (2019, December). Studi Literatur Konsep Dasar Machine Learning Dan Neural Network. *Seminar Nasional Matematika dan Terapan 2019, 1*, 160-166.

Putra, J. G. (2020). *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning.* Tokyo, Jepang. Diambil kembali dari https://wiragotama.github.io/resources/ebook/intro-to-ml-secured.pdf

Putra, M. R., Osmond, A. B., & Ansori, A. S. (2020, April 1). ESTIMASI HARGA KEBUTUHAN POKOK DI KOTA BANDUNG DAN PROVINSI JAWA BARAT MENGGUNAKAN METODE LSTM. *e-Proceeding of Engineering, 7*, 1455-1459.

Putro, B., Furqon, M. T., & Wijoyo, S. H. (2018, November). Prediksi Jumlah Kebutuhan Pemakaian Air Menggunakan Metode Exponential Smoothing (Studi Kasus : PDAM Kota Malang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2*, 4679-4686. Diambil kembali dari http://j-ptiik.ub.ac.id

Qiu, J., Wang, B., & Zhou, C. (2019, December 13). Forecasting stock prices with long-short term memory neural network based on attention mechanism. (T. Song, Penyunt.) *PLoS ONE*, 1-15. doi:10.1371/journal.pone.0227222

Ranjit, S., Shrestha, S., Subedi, S., & Shakya, S. (2018, July 01). Foreign Rate Exchange Prediction Using Neural Network and Sentiment Analysis. *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, 1173–1177. doi:10.1109/ICACCCN.2018.8748819

Revi, A., Solikhun, & Safii, M. (2018, October). JARINGAN SYARAF TIRUAN DALAM MEMPREDIKSI JUMLAH PRODUKSI DAGING SAPI BERDASARKAN PROVINSI. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), 2*(1), 297-304. Diambil kembali dari http://ejurnal.stmik-budidarma.ac.id/index.php/

Rinaldi, R. (2019, February 26). PENERAPAN UNIFIED MODELLING LANGUAGE (UML) DALAM ANALISIS DAN PERANCANGAN APLIKASI E-LEARNING. *Jurnal SIMTIKA, 2*, 43–50. Diambil kembali dari http://ejournal.undhari.ac.id/index.php/simtika/article/view/15

Rizki, M., Basuki, S., & Azhar, Y. (2020, March). Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory Untuk Prediksi Curah Hujan Kota Malang. *REPOSITOR, 2*(3), 331-338.

Sadli, A. (2018, April). SIMULASI PENGENALAN KARAKTER MENGGUNAKAN NEURAL NETWORK PADA MATLAB. *JURNAL SISTEM INFORMASI DAN TEKNOLOGI INFORMASI, 7*(1), 89-97.

Santosa, R. D., Bijaksana, M. A., & Romadhony, A. (2021, Februari). Implementasi Algoritma Long Short-Term Memory (LSTM) untuk Mendeteksi Penggunaan Kalimat Abusive Pada Teks Bahasa Indonesia. *Jurnal Tugas Akhir Fakultas Informatika, 8*(1), 691-702.

Satria, B. (2018, December 15). Prediksi Volume Penggunaan Air PDAM Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation. *JURNAL RESTI, 2*(3), 674-684.

Savalia, S., & Emamian, V. (2018, April 28). Cardiac Arrhythmia Classification by Multi-Layer Perceptron and Convolution Neural Networks. *Bioengineering, 5*, 2-10. doi:10.3390/bioengineering5020035

Sen, S., Sugiarto, D., & Rochman, A. (2020). Komparasi Metode Multilayer Perceptron (MLP) dan Long Short Term Memory (LSTM) dalam Peramalan Harga Beras. *target, 4*(1).

Setiyani, L. (2018). *Rekayasa Perangkat Lunak [Software Engineering].* (L. Setiyani, Penyunt.) Karawang, Jawa Barat, Indonesia: Jatayu Catra Internusa.

Shekar, A. K., S´a, C. R., Ferreira, H., & Soares, C. (2018, April). Building robust prediction models for defective sensor data using Artificial Neural Networks.

Siahaan, M., Jasa, C. H., Anderson, K., Rosiana, M. V., Lim, S., & Yudianto, W. (2020, November). Penerapan Artificial Intelligence (AI)Terhadap Seorang Penyandang Disabilitas Tunanetra. *Journal of Information System and Technology, 1*, 186-193. Diambil kembali dari https://journal.uib.ac.id/index.php/joint/article/view/4322/1122

Sihombing, E. N., & Syaputra, M. Y. (2020, September 23). IMPLEMENTASI PENGGUNAAN KECERDASAN BUATAN DALAM PEMBENTUKAN PERATURAN DAERAH. *JURNAL ILMIAH KEBIJAKAN HUKUM, 4*, 419-434. doi:10.30641/kebijakan.2020.V14.419-434

Sucipto, L., & Syaharuddin, S. (2018, September 2). Konstruksi Forecasting System Multi-Model untuk pemodelan matematika pada peramalan Indeks Pembangunan Manusia Provinsi Nusa Tenggara Barat. *Jurnal Ilmiah Teknologi Sistem Informasi, 4*, 114-123. doi:10.26594/register.v4i2.1263

Suhermi, N., Suhartono, Dana, I. G., & Prastyo, D. D. (2018, November). Pemilihan Arsitektur Terbaik pada Model Deep Learning Melalui Pendekatan Desain Eksperimen untuk Peramalan Deret Waktu Nonlinier. *Departemen Statistika, 18*, 153 –159. doi:10.29313/jstat.v18i2.4545

Sulistyo, A., Yudhana, A., & Sunardi, S. (2018). PERANCANGAN SISTEM INFORMASI BREEDING PLACE TERHADAP KEJADIAN DEMAM BERDARAH DENGUE (DBD) BERBASIS MOBILE TECHNOLOGY. *Prosiding SNST Fakultas Teknik, 1*(1).

Supriyadi, E. (2019, Oktober 16). PREDIKSI PARAMETER CUACA MENGGUNAKAN DEEP LEARNING LONG-SHORT TERM MEMORY (LSTM). *JURNAL METEOROLOGI DAN GEOFISIKA, 21*, 55-67. Diambil kembali dari https://pdfs.semanticscholar.org/f17a/634eb087b317c0e9eaf632c0ec4350725eb3.pdf

Surtiningsih, L., Furqon, M. T., & Adinugroho, S. (2018, August 8). Prediksi Jumlah Kunjungan Wisatawan Mancanegara Ke Bali Menggunakan Support Vector Regressiondengan Algoritma Genetika. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2*, 2578-2586. Diambil kembali dari https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/1857

Susilawati, & Muhathir. (2019, January 2). Analisis Pengaruh Fungsi Aktivasi, Learning Rate Dan Momentum Dalam Menentukan Mean Square Error (MSE) Pada Jaringan Saraf Restricted Boltzmann Machines (RBM). *JITE (Journal of Informatics and Telecommunication Engineering), 2*, 77-91.

Tabrani, M., & Pudjiarti, E. (2021, August 2). Pembangunan Perangkat Lunak E-Learning dalam Kegiatan Belajar Mengajar Sekolah Menengah Kejuruan Angkasa Husein. *Jurnal Sains Manajemen Informatika dan Komputer, 20*(2), 130~138. Diambil kembali dari https://ojs.trigunadharma.ac.id/

Thohari, A. N., & Hertantyo, G. B. (2018, August 11). Implementasi Convolutional Neural Network untuk Klasifikasi Pembalap MotoGP Berbasis GPU. *Conference on Electrical Engineering, Telematics, Industrial Technology, and Creative Media (CENTIVE)*, 50-55. Diambil kembali dari http://conferences.ittelkom-pwt.ac.id/index.php/centive/article/view/9

Wahyudi, A. (2018). Perancangan sistem menggunakan metode sdlc. *J. Din. Inform, 4*(2), 1-7.

Wijaya, R., Masriadi, & Ikhlas, M. (2020, July). IMPLEMENTASI MODEL WATERFALL PADA PENGEMBANGAN SISTEM INFORMASI PERHITUNGAN NILAI MATA PELAJARAN BERBASIS WEB PADA SEKOLAH DASAR AL-AZHAR SYIFA BUDI JATIBENING. *INFORMATION SYSTEM DEVELOPMENT, 5*(2).

Winoto, A. R. (2020, September). RANCANG BANGUN APLIKASI PRESENSI DENGAN MEDIA SUARA MENGGUNAKAN MFCC DAN ANN BERBASIS ANDROID. *Jurnal Ilmiah Sainsbertek Jurnal Ilmiah Sains & Teknologi, 1*, 46--57.

Wiranda, L., & Sadikin, M. (2019, December). PENERAPAN LONG SHORT TERM MEMORY PADA DATA TIME SERIES UNTUK MEMPREDIKSI PENJUALAN PRODUK PT. METISKA FARMA. *Jurnal Nasional Pendidikan Teknik Informatika, 8*(3), 184-196.

Zahara, S., Sugianto, & Ilmiddafiq, M. B. (2019, December 2). Prediksi Indeks Harga Konsumen Menggunakan Metode Long Short Term Memory(LSTM) Berbasis Cloud Computing. *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi), 3*, 357–363. doi:10.29207/resti.v3i3.1086