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

**SKRIPSI**

*Untuk Memenuhi Salah Satu Syarat*

*Memperoleh Gelar Sarajana Komputer*

**Program Studi : Tehnik 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 I PENDAHULUAN

## Latar Belakang

Dewasa ini data merupakan penunjang pengambilan keputusan secara cepat, atau dikenal dengan istilah *Data Driven Decision Making (DDDM)* dimana kemajuan teknologi berperan besar dalam memanfaatkan data dan informasi tersebut (Aditya, Mulyana, Eka, & Widianto, 2020). Dalam berbagai aspek termasuk melakukan prediksi mengenai informasi curah hujan yang akurat di mana pemodelan tersebut masih memiliki kekurangan seperti penggunaan jumlah parameter, asumsi matematis, dan rumusan persamaan yang cenderung rumit, untuk menghasilkan sebuah model prediksi yang mendekati keakuratan optimal harus memiliki banyak *paremeter* dan *variabel* input untuk memenuhi sebuah asumsi prediksi (Supriyadi, 2019).

Mengatasi perihal tersebut, dikembangkanlah sebuah sebuah *Kecerdasan Buatan (Artificial Inteligence)* yang memiliki kemampuan untuk melakukan pembelajaran untuk menganalisa berbagai macam asumsi dan aspek yang berpengaruh untuk menarik kesimpulan (Supriyadi, 2019)*. Artificial Intelligence (AI)* atau *kecerdasan buatan* didefinisikan secara berbeda dalam konteks yang berbeda pula dalam disiplin ilmu komputer AI adalah mempelajari cara mensimulasikan untuk melakukan tugas yang biasanya membutuhkan pemahaman seperti manusia (Knowledge@Wharton, 2018). AI juga suatu cabang ilmu komputer yang menggunakan lebih banyak simbol daripada angka, dan memproses informasi berdasarkan jumlah aturan dalam merepresentasikan pengetahuan (Swarnkar & Swarnkar, 2019).

*Machine Learning* adalah bagian dari AI di mana mesin digunakan untuk belajar dari pengalaman masa lalu (Aditya, Mulyana, Eka, & Widianto, 2020). Algoritma *Machine Learning* digunakan dalam berbagai aplikasi, seperti dalam kedokteran, pengenalan email, pengenalan suara, dan visi komputer, di mana sulit atau tidak mungkin untuk mengembangkan algoritma konvensional untuk melakukan tugas yang diperlukan (Hu, Niu, Carrasco, Lennox, & Arvin, 2020).

Beberapa implementasi *Machine Learning* menggunakan data dan *Neural Network (Jaringan Saraf)* dengan cara yang meniru kerja otak biologis manusia (Zhou, 2019). *Deep learning* dapat dipahami sebagai bentuk *Neural Network* layer berganda yang merupakan bagian dari *Machine Learning* yang dapat digunakan dalam tugas termasuk *computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection* dan *board game programs*, di mana mereka telah menghasilkan hasil yang sebanding dan dalam beberapa kasus melebihi kinerja para pakar (Hu, Niu, Carrasco, Lennox, & Arvin, 2020).

Misalnya, sebuah komputasi yang menggunakan *deep learning*, akan mampu memahami konsep seperti garis, bentuk, tekstur, dan juga pengaruhnya dengan melihat data-data citra tanpa bantuan tambahan dari manusia (Schneiderman & Kanade, 2002). *Machine Learning* senantiasa bekerja menggunakan 1 layer dimana *deep learning* bekerja lebih dari 1 *layer*. untuk batasan layer dari *deep learning* itu sendiri sebagai *Neural Network* biasanya memiliki 3 *layer* atau lebih, makin bayak *layer* yang digunakan akan memengaruhi lama / delay waktu yang terpakai untuk komputer mengkalkulasi (Hinton, et al., 2012).

*Layer* pada *deep learning* dapat di gambarkan seperti lapisan neuron pada otak manusia *layer* itu nantinya akan menggambarkan jarak atau vector menggunakan Fungsi matematika yaitu fungsi *sigmoid (σ)* (Putra J. G., 2020)*.* Alasan fungsi sigmoid digunakan karena dalam fungsi ini membutuhkan perhitungan yang relatif mudah dan cepat. Selain itu, fungsi sigmoid dapat diartikan sebagai nilai peluang karena nilainya antara 0 dan 1 (Putra J. G., 2020).

Salah satu pendekatan *deep learning* yang mampu secara otomatis mempelajari fitur yang dideskripsikan dalam bentuk *vektor* adalah *Recurrent Neural Networks (RNN)* (Puspaningrum, Bunga, & Iryanto, 2020). Pada *RNN* sendiri teknik learning bekerja dengan menyimpan layer dari output kembali sebagai input pada hidden layer berikutnya hingga memprediksi hasil akhir (Tarkus, Sompie, & Jacobus, 2020). Kelemahan *RNN* adalah tidak mampu lagi untuk belajar menghubungkan informasi ketika ada kesenjangan yang terus tumbuh, memori yang tersimpan akan semakin tidak relevan seiring waktu berjalan karena tertimpa dengan memori baru (Putra, Osmond, & Ansori, 2020), di karenakan kelemahan dari *RNN* sendiri tidak dapat mempelajari informasi yang terlalu jauh atau *Long-Term Dependencies*, yang cukup jauh pada masukannya (Wibisono & Khodra, 2018).

Penelitian *deep learning* terdahulu yang di lakukan Juanda, Jondri, & Rohmawati, tentang *Prediksi Harga Bitcoin Dengan Menggunakan Recurrent Neural Network* menurutnya, masalah prediksi *time series* adalah jenis pemodelan prediktif yang sulit, tidak seperti pemodelan predikftif regresi, *time series* juga menambah kompleksitas ketergantungan urutan antar variable input. *Recurrent Neural Network* terbukti berhasil digunakan untuk prediksi data *time series* karena *RNN* mampu menggunakan informasi yang telah direkam sebelumnya yang panjang urutannya atau sequence-nya beragam-ragam. Oleh karena itu, pembangunan sistem ini dibuat dengan metode *Recurrent Neural Network* dengan menggunakan algoritma *Backpropagation Through Time*. Hasil akhir Prediksi harga Bitcoin dapat dilakukan menggunakan Recurrent Neural Network. Akurasi rata-rata terbaik yang didapatkan adalah 98.76% pada data latih dan 97.46% pada data uji, dengan parameter jumlah pola input terbaik adalah 5, jumlah epoch 1000, nilai learning rate 0.001 dan jumlah hidden unit 50 (Juanda, Jondri, & Rohmawati, 2018).

*Long Short-Term Memory (LSTM)* merupakan sebuah pengembangan metode dari arsitektur *Recurrent Neural Network (RNN)*, Banyak peneliti yang mengembangkan metode *LSTM* di berbagai bidang seperti dalam bidang prediksi deret waktu atau *forecasting* dikarenakan metode *LSTM* mampu mengatasi kekurangan tersebut karena metode ini dapat mengatur memori pada setiap masukannya dengan menggunakan memory cells dan gate units pada setiap neurons yang berfungsi sebagail pengatur memori (Putra, Osmond, & Ansori, 2020). Contoh penggunaan deep learning untuk data *time series* yang banyak dihasilkan dari pengamatan cuaca adalah *LSTM*, *LSTM* sendiri diciptakan oleh Hochreiter dan Schmidhuber pada tahun 1997 (Supriyadi, 2019).

Penelitian *LSTM* terdahulu yang yang di teliti oleh Poornima & Pushpalatha, mengenai *Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units* dalam penelitian tersebut menyajikan *Long Short-Term Memory (LSTM)* berbasis *Recurrent Neural Network (RNN)* 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 (RMSE), akurasi, jumlah epoch, loss*, dan *learning rate*. Menurutnya *LSTM* dapat menyimpan data besar ke dalam memorinya dan dapat menghindari gradient yang hilang lebih baik dari pada *RNN* dan menunjukan akurasi lebih baik di bandingkan *RNN*, *LSTM* pun juga mempertahankan akurasi di masa depan seiring dengan mempertimbangkan nilai *Root Mean Square Error (RMSE), akurasi, jumlah epoch, loss, dan learning rate* (Poornima & Pushpalatha, 2019)*.*

Penilituan *LSTM* terdahulu yang juga di lakukan oleh Supriyadi, mengenai metode *deep learning LSTM* untuk memprediksi parameter cuaca, seperti suhu udara, kelembaban, kecepatan angin, dan tekanan udara. Metode ini bekerja dengan memanfaatkan fungsi matematika seperti fungsi *tanh* dan *sigmoid* yang berada dalam layer *LSTM*. Adapun jumlah layer yang digunakan sebanyak 200 buah. 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. Diperoleh hasil prediksi suhu udara, kelembaban, kecepatan angin, dan tekanan udara 1 hari ke depan memiliki *RMSE* yang baik. Dari parameter cuaca tersebut hanya parameter suhu dan kelembaban udara yang mengalami pertambahan *RMSE* seiring bertambahnya waktu. Sedangkan parameter kecepatan angin dan tekanan udara mengalami penurunan di hari ketiga dan meningkat secara kontinu hingga 1 bulan ke depan (Supriyadi, 2019).

Berdasarkan rincian penjelasan sebelumnya, sangat dimungkinkan untuk menggunakan *deep learning* dengan metode *LSTM* dikarenakan mendukung kegiatan proyeksi curah hujan. Karena data pengamatan meteorologi umumnya berupa *vektor* dan *time* series hal tersebut dangat cocok di atasi dengan metode *LSTM*. Oleh karena itujudulpenelitian ini yakni “**PROYEKSI CURAH HUJAN DAERAH PADANG PARIAMAN MENGGUNAKAN DEEP LEARNING DENGAN METODE LONG SHORT-TERM MEMORY.**

## Perumusan Masalah

Berdasarkan uraian pada latar belakang masalah maka yang menjadi perumusan masalah pada skripsi ini adalah sebagai berikut :

1. Bagaimana melakukan proses prediksi curah hujan dengan mengunakan pendekatan Deep Learning untuk menghasilkan alternatif dalam pengambilan sebuah keputusan ?
2. Bagaimana konsep pendekatan Deep Learning dengan metode *Long Short-Term Memory* dapat melakukan prediksi curah hujan dengan parameter cuaca seperti suhu, kelembaban, kecepatan angin dan tekanan udara ?
3. Bagaimana Deep Learning dengan metode Long Short-Term Memory di implementasikan kedalam sebuah sistem yang dibangun untuk memprediksi curah hujan di daerah padang pariaman ?

## Hipotesis

Berdasarkan rumusan masalah yang telah ditentukan maka dapat diambil kesimpulan sebagai berikut :

1. Dengan melakukan proses prediksi curah hujan dengan mengunakan pendekatan Deep Learning diharapak dapat menghasilkan alternatif dalam pengambilan sebuah keputusan.
2. Dengan menggunakan konsep pendekatan Deep Learning metode *Long Short-Term Memory* diharapkan dapat melakukan prediksi curah hujan dengan parameter cuaca seperti suhu, kelembaban, kecepatan angin dan tekanan udara.
3. Penerapan Deep Learning dengan metode Long Short-Term Memory diharapkan dapat di implementasikan kedalam sebuah sistem yang dibangun untuk memprediksi curah hujan di daerah padang pariaman.

## Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut :

1. Data yang digunakan adalah data BMKG Klimatologi Padang Pariaman dari tahun 1985 sampai tahun 2021.
2. Metode yang digunakan dalam penelitian ini Long Short-Term Memory.
3. Menggunakan Bahasa Pemograman Python.
4. Fungsi Aktivasi yang digunakan adalah fungsi sigmoid (σ) dan tanh.
5. Untuk variabel yang digunakan dalam penelitian ini terdiri dari curah hujan, kecepatan angin, temperatur, dan kelembapan udara.
6. Data dibagi menjadi dua bagian, yaitu data training dan data test dengan rasio 9:1 di mana 9 untuk training dan 1 untuk test.

## Tujuan Penelitian

Adapun Tujuan dari penelitian ini adalah sebagai berikut :

1. Mendapatkan perkiraan pola curah hujan di daerah Padang Pariaman.
2. Mendapatkan hasil implementasi metode long short-term memory untuk prediksi curah hujan
3. Dengan mendapatkan hasil data pengujian yang baik maka dapat digunakan untuk memprediksi curah hujan masa depan. Dan dapat digunakan oleh BMKG Klimatologi Padang Pariaman untuk melakukan antisipasi dan perencanaan mendatang.

## Manfaat Penelitian

Hasil dari penelitian di harapkan dapat memberikan manfaat sebagai berikut :

1. Menambah pengetahuan penulis dalam memprediksi / memproyeksi data berbentuk *time series* yang di selesaikan dengan cara *deep learning* dengan menggunakan metode *long short-term memory*.
2. Membuktikan keakuratan metode *long short-term memory* dalam melakukan peramalan khususnya curah hujan.
3. Memberikan informasi tambahan mengenai peramalan curah hujan di daerah padang pariaman yang akan terjadi pada masa mendatang.

## Gambaran Umum Objek Penelitian

BMKG Klimatologi Padang Pariaman merupakan Stasiun Klimatologi BMKG Kelas II yang terletak di daerah *Jalan Raya Padang – Bukittinggi KM. 51 Kapalo Hilalang Sumatera Barat*. Berikut ini adalah gambaran umum tentang BMKG Klimatologi :

### Sejarah BMKG

Pengamatan meteorologi dan geofisika di Indonesia dimulai pertama kali pada tahun 1841 diawali dengan pengamatan yang dilakukan secara individual oleh Dr. Pieter Loth Onnen, Kepala Rumah Sakit di Bogor. Dari tahun ke tahun kegiatannya berkembang seiring dengan meningkatnya kebutuhan akan data hasil pengamatan cuaca dan geofisika. Pada tahun 1866, pemerintah Hindia Belanda meresmikan kegiatan pengamatan perorangan sebagai lembaga pemerintah dengan nama Observatorium Magnetisch en Meteorologisch atau Observatorium Magnetik dan Meteorologi yang dipimpin oleh Dr. Pieter Adrian Bergsma.

Pada tahun 1879, 74 jaringan alat pengukur hujan dibangun di Jawa. pengamatan medan magnet bumi dipindahkan dari Jakarta ke Bogor Pada tahun 1902. Pada tahun 1908 pemantauan gempa dimulai dengan pemasangan komponen horizontal seismograf Wiechert di Jakarta, sedangkan pemasangan komponen vertikal dilakukan pada tahun 1928. Pada tahun 1912, pengamatan meteorologi ditata ulang dengan menambahkan jaringan sekunder. Sedangkan pada tahun 1930 jasa meteorologi mulai digunakan untuk penerangan.

Pada masa pendudukan Jepang antara tahun 1942 dan 1945, nama badan meteorologi dan geofisika diubah menjadi Kisho Kauso Kusho atau Lembaga Meteorologi. Pada tahun 1945 Setelah proklamasi kemerdekaan Indonesia, badan tersebut dibagi menjadi dua: Di Yogyakarta dibentuk Badan Meteorologi yang berkedudukan di lingkungan Mabes TNI khusus untuk melayani kepentingan Angkatan Udara. Di bawah Kementerian Pekerjaan Umum dan Energi Badan Meteorologi dan Geofisika dibentuk Di Jakarta.

Pada tanggal 21 Juli 1947 Biro Meteorologi dan Geofisika diambil alih oleh Pemerintah Belanda dan namanya diubah menjadi Meteorologisch en Geofisiche Dienst. Badan Meteorologi dan Geofisika yang dikelola oleh Pemerintah Republik Indonesia berada di Jl. Gondangdia, Jakarta. Pada tahun 1949, setelah penyerahan kedaulatan Republik Indonesia dari Belanda, Meteorologisch en Geofisiche Dienst diubah menjadi Biro Meteorologi dan Geofisika di bawah Departemen Perhubungan dan Pekerjaan Umum. Selanjutnya pada tahun 1950 Indonesia resmi masuk sebagai anggota Organisasi Meteorologi Dunia dan Kepala Badan Meteorologi dan Geofisika menjadi Wakil Tetap Indonesia dengan WMO.

Pada tahun 1955 Biro Meteorologi dan Geofisika berubah nama menjadi Lembaga Meteorologi dan Geofisika di bawah Kementerian Perhubungan, dan pada tahun 1960 namanya dikembalikan menjadi Biro Meteorologi dan Geofisika di bawah Kementerian Perhubungan Udara.

Pada tahun 1965, namanya diubah menjadi Direktorat Meteorologi dan Geofisika, posisinya tetap di bawah Kementerian Perhubungan Udara. statusnya dinaikkan menjadi lembaga setingkat eselon I dengan nama Badan Meteorologi dan Geofisika, dengan jabatan tetap di bawah Kementerian Perhubungan. Badan Meteorologi dan Geofisika.

Melalui Peraturan Presiden Nomor 61 Tahun 2008, BMG berganti nama menjadi Badan Meteorologi, Klimatologi, dan Geofisika dengan status tetap sebagai Lembaga Pemerintah Non Departemen.

### Struktur Organisasi dan Nama-Nama Pejabatnya

Berikut adalah bentuk struktu organisasi BMKG Padang Pariaman :

(Sumber : Sta. Klim. Kelas II Padang Pariaman, 2021)

Gambar 1. Struktur BMKG Padang Pariaman

BMKG Klimatologi Padang Pariaman dipimpin oleh Kepala Stasiun Klimatologi (Kaslim) bertanggung jawab terhadap seluruh bidang yang ada pada instansi BMKG, Staf yang membantu Kaslim dalam menjalankan aktivitas di kantor adalah sebagai berikut :

1. Tata usaha yang bertanggung jawab terhadap administrasi kantor.
2. Bagian analisa yang bertanggung jawab terhadap pengolahan data dan analisis data-data yang dikirim ke Balai Wilayah I.
3. Tenaga teknis yang bertanggung jawab terhadap data-data klimatologi yang ada di BMKG Klimatologi Padang Pariaman.
4. Tim pengamat yang bertanggung jawab terhadap pengaturan jadwal pengamatan di BMKG Klimatologi Padang Pariaman.
5. Tim komunikasi dan peralatan yang bertanggung jawab terhadap pengiriman informasi kondisi peralatan yang ada di BMKG Klimatologi Padang Pariaman.

### Visi

Mewujudkan BMKG yang handal, tanggap dan mampu dalam rangka mendukung keselamatan masyarakat serta keberhasilan pembangunan nasional, dan berperan aktif di tingkat Internasional.

Terminologi di dalam visi tersebut dapat dijelaskan sebagai berikut :

1. Pelayanan informasi meteorologi, klimatologi, kualitas udara, dan geofisika yang handal ialah pelayanan BMKG terhadap penyajian data, informasi pelayanan jasa meteorologi, klimatologi, kualitas udara, dan geofisika yang akurat, tepat sasaran, tepat guna, cepat, lengkap, dan dapat dipertanggungjawabkan.
2. Tanggap dan mampu dimaksudkan BMKG dapat menangkap dan merumuskan kebutuhan stakeholder akan data, informasi, dan jasa meteorologi, klimatologi, kualitas udara, dan geofisika serta mampu memberikan pelayanan sesuai dengan kebutuhan pengguna jasa.

### Misi

Dalam rangka mewujudkan Visi BMKG, maka diperlukan visi yang jelas yaitu berupa langkah-langkah BMKG untuk mewujudkan Misi yang telah ditetapkan yaitu :

1. Mengamati dan memahami fenomena meteorologi, klimatologi, kualitas udara dan geofisika.
2. Menyediakan data, informasi dan jasa meteorologi, klimatologi, kualitas udara dan geofisika yang handal dan terpercaya.
3. Mengkoordinasikan dan memfasilitasi kegiatan di bidang meteorologi, klimatologi , kualitas udara dan geofisika.
4. Berpartisipasi aktif dalam kegiatan internasional di Bidang meteorologi, klimatologi, kualitas udara dan geofisika.

# BAB II LANDASAN TEORI

## 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 variable 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.

## Time Series

Time series adalah rangkaian data yang berupa nilai pengamatan yang diukur selama kurun waktu tertentu, berdasarkan waktu dengan interval yang sama, dalam hal ini adalah data nilai indeks harga saham yang diperoleh dalam jangka waktu yang berurutan data ini dikumpulkan dari waktu ke waktu. Waktu yang digunakan dapat berupa hari, bulan, tahun (Pandji, Indwiarti, & Rohmawati, 2019).

Analysis *time series* dilakukan untuk memperoleh pola dari data tersebut dengan menggunakan historical data yang akan digunakan untuk memprediksi suatu nilai di masa yang akan datang. Oleh karena itu untuk memilih suatu metode time series yang tepat harus mempertimbangkan pola data, agar metode yang digunakan dengan pola tersebut dapat diuji (Pandji, Indwiarti, & Rohmawati, 2019).

## Perangkat Lunak *(Software)*

Software / Perangkat Lunak komputer adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer. Data elektronik yang disimpan oleh komputer itu dapat berupa program atau instruksi yang akan menjalankan suatu perintah. Melalui software atau perangkat lunak inilah suatu komputer dapat menjalankan suatu perintah (Wali, 2020).

### Jenis-Jenis Perankat Lunak

Software atau perangkat lunak komputer berdasarkan distribusinya dibedakan menjadi beberapa macam, yaitu software berbayar, software gratis atau free (freeware, free software, shareware, dan adware) (Wali, 2020).

#### Software Berbayar

Software berbayar merupakan perangkat lunak yang didistribusikan untuk tujuan komersial. Setiap pengguna yang ingin menggunakan atau mendapatkan software tersebut dengan cara membeli atau membayar pada pihak yang mendistribusikannya. Pengguna yang menggunakan software berbayar umumnya tidak diijinkan untuk menyebarluaskan software tersebut secara bebas tanpa izin dari penerbitnya. Contoh software berbayar ini misalnya adalah sistem Microsoft Windows, Microsoft Office, Adobe Photoshop, dan lain-lain (Wali, 2020).

#### Freeware

Freeware atau perangkat lunak gratis adalah perangkat lunak komputer berhak cipta yang gratis digunakan tanpa batasan waktu, berbeda dari shareware yang mewajibkan penggunanya membayar (misalnya setelah jangka waktu percobaan tertentu atau untuk memperoleh fungsi tambahan). Freeware juga didefinisikan sebagai program apa pun yang didistribusikan gratis, tanpa biaya tambahan. Sebuah contoh utama adalah Suite Browser, Mail Client, dan Mozilla News, juga didistribusikan di bawah GPL (Wali, 2020).

#### Free Software

Free software lebih mengarah kepada bebas penggunaan, tetapi tidak harus gratis. Pada kenyataannya, namanya adalah karena bebas untuk mencoba perangkat lunak sumber terbuka (open source) dan di sanalah letak inti dari kebebasan (Wali, 2020).

Perbedaan yang nyata antara free software dan freeware. Konflik muncul dalam arti kata free dalam Bahasa Inggris, yang berarti keduanya bebas dan gratis. Oleh karena itu, seperti yang disebutkan sebelumnya, free software tidak perlu bebas, sama seperti freeware tidak harus gratis (Wali, 2020).

#### Shareware

Shareware juga bebas, tetapi lebih dibatasi untuk waktu tertentu. Shareware adalah program terbatas didistribusikan baik sebagai demonstrasi atau versi evaluasi dengan fitur atau fungsi yang terbatas atau dengan menggunakan batas waktu yang ditetapkan (misalnya, 30 hari) (Wali, 2020).

#### Malware

Perangkat perusak (Bahasa Inggris: malware, berasal dari lakuran kata malicious dan software) adalah perangkat lunak yang diciptakan untuk menyusup atau merusak sistem komputer, peladen, atau jejaring komputer tanpa izin termaklum (informed consent) dari pemilik. Istilah ini adalah istilah umum yang dipakai oleh pakar komputer untuk mengartikan berbagai macam perangkat lunak atau kode perangkat lunak yang mengganggu atau mengusik. Istilah “virus computer” kadang-kadang dipakai sebagai frasa pemikat (catch phrase) untuk mencakup semua jenis perangkat perusak, termasuk virus murni (true virus) (Wali, 2020).

### 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, disain, pengkodean, 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.

#### Perencanaan

Kegiatan perencanaan menciptkan 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.

#### Pemodelan

Pemodelan, dilakukan bertujuan untuk membuat sketsa sehingga tim perangkat lunak dapat memahami gambaran besar produk yang akan di buat.

#### Konstruksi

Konstruksi, sediri adalah kegiatan yang menggabungkan pengkodean dan pengujian.

#### Penyerahan

Penyerahan di sini merupakan Penyerahan perangkat lunak kepada user, penyajian perangkat lunak kepada user untuk di evaluasi.

## Unified Modelling Language *(UML)*

Pemodelan dalam suatu rekayasa perangkat lunak merupakan suatu hal yang dilakukan di tahapan awal. Pemodelan delam 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 standart bahasa yang banyak digunakan di dunia industry untuk mendefinisikan requirement, membuat analisis & desain, sert amenggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak (Hasanah & Untari, 2020).



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

Gambar 2. Bagan UML

Berdasarkan **Gambar 2** 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 subsistem pada suatu sistem.
3. Structure diagram, merupakan kumpulan diagram yang digunakan untuk menggambarkan strutur statis dari sistem yang dimodelkan.

### Behavior Diagrams

#### 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).

Table 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).

Table 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)

#### State Machine Diagram

State Machine Diagram menggambarkan perilaku objek tertentu dalam sistem perangkat lunak, menentukan urutan peristiwa dan tanggapan terhadap peristiwa objek selama masa berlakunya. Diagram mesin keadaan umum terdiri dari himpunan keadaan berhingga, himpunan keadaan semu berhingga, dan himpunan transisi berhingga (Damjan & Vatanawood, 2017).

Table 3. Notasi State Machine 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 : Damjan & Vatanawood, 2017)

### Interaction Diagrams

#### Sequence Diagram

Sequence diagrammenggambarkan 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).

Table 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 atau diagram kolaborasi merupakan suatu diagram yang memperlihatkan juga mengelompokkan pesan yang terdapat pada sekitar objek (Supriatna & Sherin, 2017).

Table 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 antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |

(Sumber : Supriatna & Sherin, 2017)

### Structure Diagrams

#### Class Diagram

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

Table 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 antarkelas dengan makna umum.  Asosiasi biasanya juga disertai dengan multiplicity |
| 4 | *Directed Association* | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |
| 5 | *Dependency* | Relasi antar kelas dengan kebergantungan antar kelas |
| 6 | *Aggregation* | Relasi antarkelas dengan makna semua-bagian (whole-part) |

(Sumber : Rinaldi, 2019)

#### Deployment Diagram

Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (run-time). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram deployment berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponenkomponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (distributed computing) (Dermawan & Hartini, 2017).

Table 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* | Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai. |
| 5 | *Link* | Relasi antar komponen. |

(Sumber : Dermawan & Hartini, 2017).

## Bahasa Pemograman Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar (Syahrudin & Kurniawan, 2018).

Python mendukung multi paradigma pemrograman, utamanya, namun tidak dibatasi pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa script meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa script. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi (Syahrudin & Kurniawan, 2018). Saat ini kode python dapat dijalankan di berbagai platform sistem operasi, beberapa diantaranya adalah (Syahrudin & Kurniawan, 2018) :

* Linux/Unix
* Windows
* Mac OS X
* Java Virtual Machine
* Amiga
* Palm
* Symbian (untuk produk-produk Nokia)

Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Namun pada prinsipnya Python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Lisensi Python tidak bertentangan baik menurut definisi Open Source maupun **General Public License (GPL)** (Syahrudin & Kurniawan, 2018)**.**

## Database

Database adalah sebuah system yang di buat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. Database terdiri dari kumplan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital. Database digital di manage menggunakan Database Management System (DBMS), yang menyimpan isi database, mengizinkan pembuatan dan maintenance data dan pencarian dan akses yang lain Beberapa Database yang ada saat ini adalah, Mysql, Sql Server, Ms.Access, Oracle, dan PostgreSql (Ramadhan & Mukhaiyar, 2020).

### Fungsi Database

Adapun beberapa fungsi dari database adalah (Ramadhan & Mukhaiyar, 2020) :

1. Mempermudah identifikasi data dengan cara pengelompokkan data, salah satu contoh nya dengan pembuatan beberapa tabel atau field yang berbeda-beda.
2. Meminimalisir suatu data ganda.
3. Mempermudah penggunaan user dalam berbagai hal, misalnya pada saat peng-inputan data baru.
4. Penyimpanan secara digital
5. Menjadi alternatif lain terkait masalah penyimpanan ruang dalam suatu aplikasi.

### Jenis Database

Adapun beberapa jenis database, diantaranya (Ramadhan & Mukhaiyar, 2020) :

1. Operational Database, Database jenis ini mengoperasikan penyimpanan data yang sangat rinci agar dapat dengan mudah digunakan. Database ini biasa digunakan untuk database pelanggan
2. Relational Database Pada database ini, user dapat mengakses atau mencari informasi dalam tabel yang berbeda –beda.
3. Distributed DatabaseUntuk database jenis ini dapat mendistribusikan data –data secara tersebar namun saling berhubungan serta dapat diakses secara bersama –sama.
4. External Database Database terakhir ini sering digunakan sebagai keperluan komersial karena kemudahan mengaksesnya yang memang dikhususkan untuk publik.

### MySQL

SQL (Structured Query Language) adalah sebuah bahasa scripting yang dipergunakan untuk mengolah database. Database besar seperti MySQL, PostgreSQL dan SQL Server sudah menggunakan SQL untuk mengolah database nya. SQL yangdi pergunakan software database tersebut adalah sama kecuali sedikit perbedaan dibeberapa tempat (Ramadhan & Mukhaiyar, 2020).

MySQL menggunakan bahasa SQL untuk mengakses database nya. Lisensi Mysql adalah FOSS License Exception dan ada juga yang versi komersial nya. Tag Mysql adalah “The World's most popular open source database”. MySQL tersedia untuk beberapa platform, di antara nya adalah untuk versi windows dan versi linux (Ramadhan & Mukhaiyar, 2020).

## Normalisasi dan Denormalisasi

### Normalisasi

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

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 noermalisasi

: Hasil output

: Nilai minimal data actual keseluruhan

: Nilai maksimal data actual keseluruhan

## Fungsi Aktivasi

Fungsi aktivasi sangat berperan dalam mengaktifkan setiap neuron pada *jaringan saraf tiruan* serta menentukan keluaran dari suatu jaringan saraf tiruan (Susilawati & Muhathir, 2019). Fungsi aktivasi berfungsi untuk menghitung nilai keluaran berdasarkan nilai *input* dan bobot pada neuron, fungsi optimisasi berfungsi untuk meminimalisir error antara nilai keluaran dengan nilai yang dihasilkan oleh neuron. Minimalisasi nilai error dilakukan dengan cara mengubah nilai bobot di setiap neuron. Bobot diubah beberapa iterasi hingga menjadi konvergen, jumlah iterasi tersebut dinamakan dengan *epoch* (Wibawa, 2017). 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 3. 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, 2017)

Gambar 4. Ilustrasi Tanh

## 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).

*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)

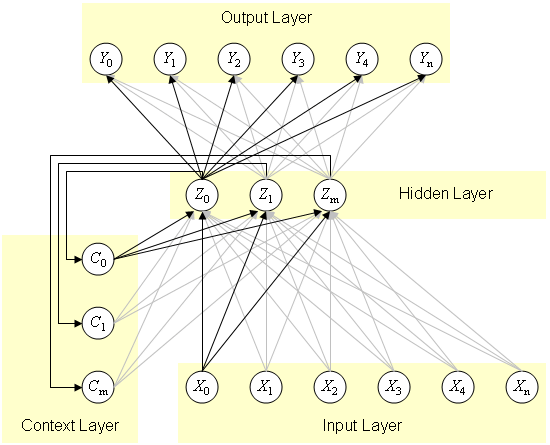
Gambar 5. Perbedaan Simple Neural Network dan Deep Learning

Pada *Neural Network* diilhami oleh pemrosesan informasi dan distribusi node komunikasi dalam sistem biologis dan 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).

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).

## Recurrent Neural Network

*Recurrent Neural Network (RNN)* adalah model yang efisien untuk analisis sentimen. *RNN* menggunakan sel memori yang mampu menangkap informasi tentang urutan panjang (Patel & Tiwari, 2019). Dalam melakukan prediksi, RNN menggunakan data input pada waktu saat ini dan juga input dari data sebelumnya. Hubungan antar input tersebut dapat berguna untuk memberikan informasi kepada seluruh hidden layer. Sehingga dapat dikatakan bahwa RNN memiliki memori yang berisikan hasil rekaman informasi yang dihasilkan sebelumnya (Gulli & Pal, 2017).



(Sumber : Juanda, Jondri, & Rohmawati, 2018)

Gambar 6. Arsitektur Recurrent Neural Network

*RNN* merupakan variasi dari *Artificial Neural Network*, perbedaan utama yang terdapat pada model ini adalah sinyal dapat mengalir secara *forward* dan *backward* secara berulang. Untuk bisa melakukan hal tersebut, maka ditambahkan sebuah layer baru yang disebut dengan *context layer*. Selain melewati input antar layer, output dari setiap layer juga menuju ke context layer untuk digunakan sebagai inputan pada *timestep* berikutnya. RNN menyimpan informasi di *context layer*, yang membuatnya dapat mempelajari urutan data dan menghasilkan output atau urutan lain. Jika ditarik kesimpulan maka dapat dikatakan bahwa RNN memiliki memori yang berisikan hasil rekaman informasi yang dihasilkan sebelumnya (Juanda, Jondri, & Rohmawati, 2018). Pada layer tersembunyi *(hidden layer)* bertindak sebagai penyimpanan internal untuk menyimpan informasi yang dikumpulkan selama tahap awal pemrosesan data sekuensial. Alasan *RNN* disebut "berulang" adalah karena model melakukan tugas yang sama untuk setiap elemen urutan saat memanfaatkan informasi itu (Ludwig, 2019).

Diagram, schematic

Description automatically generated

(Sumber : Juanda, Jondri, & Rohmawati, 2018)

Gambar 7. Proses Recurrent Neural Network

Pada **Gambar 7** menunjukkan RNN pada posisi yang tidak dibuka ke full network. Dengan membuka gulungan RNN maka simpelnya kita menuliskan seluruh jaringan dengan urutan (sequence) secara lengkap (Juanda, Jondri, & Rohmawati, 2018). Berikut ialah keterangan simbol yang ada pada **Gambar 7** :

1. ialah input pada setiap *time step* biasanya berupa input sebuah vektor.
2. adalah “memory” pada *RNN* akan dihitung berdasarkan *hidden state* sebelumnya dan input pada langkah *time step* sehingga metode ini disebut sekuensial pada setiap
3. adalah output untuk setiap step

Pemetaan satu simpul S𝑡 dan output 𝑂𝑡 dapat ditulis sebagai (Tian, Ma, Zhang, & Zhan, 2018) :

Dimana dikatakan memori jaringan pada waktu , , , dan adalah matriks bobot berbagai di setiap layer, dan mewakili *input* dan *output* pada waktu , dan fungsi dan fungsi mewakili fungsi non-linear (Tian, Ma, Zhang, & Zhan, 2018).

Berbeda dengan koneksi bobot yang dibuat antara layer dalam *ANN*, *RNN* dapat menggunakan internal state (memori) untuk memproses urutan input (Che, Purushotham, Cho, Sontag, & Liu, 2018). *Hidden state* menyimpan informasi pada proses sebelumnya, dan *outputnya* diturunkan dari proses sekarang dengan memory sebelumnya. Secara teori, *RNN* juga mampu menangani dengan ketergantungan jangka panjang. Namun, dalam beberapa hal, *RNN* tidak dapat menghafal informasi sebelumnya dengan baik ketika interval waktu sangat panjang, masalah ini biasanya di sebut dengan *gradient vanishing problem* (Tian, Ma, Zhang, & Zhan, 2018). Dikarenakan RNN hanya mengingat beberapa langkah awal dalam urutan data sehingga tidak cocok untuk mengingat urutan yang lebih panjang. Untuk mengatasi kekurangan ini dan meningkatkan kinerja *RNN*, maka dikembangkanlah jenis jaringan arsitektur *RNN* khusus berulang lainnya yaitu *LSTM (Long Short-Term Memory)* (Moghar & Hamiche, 2020; Tian, Ma, Zhang, & Zhan, 2018).

## 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 8. Perulangan modul dalam 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 9. Alur Informasi Cell State pada LSTM

Keterangan (Pardede & Ibrahim, 2020) :

: *Cell state*

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

Pada **Gambar 9**, 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, 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).

Graphical user interface, application

Description automatically generated

(Sumber : Manu, 2021)

Gambar 10. Layer sigmoid mengeluarkan angka antara nol dan satu.

Pada **Gambar 10** 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)*.* Berikut ialah keterangan setiap gate yang ada pada **Gambar 8** :

### *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 11** :

Diagram

Description automatically generated

(Sumber : Manu, 2021)

Gambar 11. Alur Informasi 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 11** 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 9**. *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 12** :

Diagram

Description automatically generated

(Sumber : Manu, 2021)

Gambar 12. Alur Informasi 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 12** 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 13** :

Diagram

Description automatically generated with medium confidence

(Sumber : Manu, 2021)

Gambar 13. Alur 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 13**, 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 14** :

A picture containing text, clock

Description automatically generated

(Sumber : Manu, 2021)

Gambar 14. Alur Informasi 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 14** 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).

## 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 parameter -nya 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

## 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 Size dan Epoch

*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).

# BAB III METODOLOGI PENELITIAN

## Kerangka Penelitian

Kerangka penelitian merupakan rangkaian kegiatan yang akan dilakukan dalam suatu penelitian berupa rangkaian grafik yang menggambarkan alur proses penelitian proyeksi curah hujan daerah padang pariaman menggunakan deep learning dengan metode long short-term memory, sehingga langkah-langkah yang dilakukan oleh penulis dalam perancangan ini tidak melenceng dari pokok bahasan dan lebih mudah dipahami, yang diilustrasikan seperti pada **Gambar 15** berikut.



Gambar 15. Kerangka Penelitian

## Tahapan Penelitian

Tahapan penelitian merupakan langkah-langkah yang akan dilakukan untuk mempermudah dalam melakukan penelitian. Tahapan dalam penelitian ini terdiri dari penelitian pendahuluan, pengumpulan data, analisa, perancangan, implementasi, pengujian dan kesimpulan yang akan dijelaskan sebagai berikut :

### Penelitian Pendahuluan

Penelitian pendahuluan merupakan proses melakukan pendekatan terhadap objek penelitian. Tujuan dari Penelitian pendahuluan tak lain dapat memberikan solusi terhadap masalah yang di identifikasi. Permasalahan pada proyeksi curah hujan khusus daerah padang pariaman masih terbatas dari masukan data pemodelan, di mana dalam pemodelan tersebut masih memiliki kekurangan seperti penggunaan jumlah parameter, asumsi-asumsi matematis, dan formulasi persamaan yang cenderung rumit. Oleh karena itu dilakukanlah sebuah penelitian yang nantinya dapat membantu BMKG Klimatologi Padang Pariaman dalam melakukan prediksi curah hujan di daerah padang pariaman. Penelitian ini nantinya akan memerikan alternatif lain dalam melakukan prediksi curah hujan di daerah padang pariaman dengam menggunakan metode Long Short-Term.

### Pengumpulan Data

Dalam Penelitian ini semua data bersumber dari Stasiun Klimatologi Kelas II Padang Pariaman dari tahun 1985-2021, dan beberapa pencarian referensi seperti buku-buku, karya-karya ilmiah maupun jurnal, baik yang ada diperpustakaan maupun yang ada di internet yang berhubungan dengan penelitian. Data juga didapat dari studi lapangan dengan melakukan wawancara secara langsung.

#### Waktu Penelitian

Penelitian dilakukan dengan merekap data-data pengamatan unsur-unsur cuaca yang dilakukan oleh Stasiun Klimatologi Kelas II Padang Pariaman secara umum yang di lakukan dari bulan November sampai dengan selesai.

Table 8. Jadwal Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Kegiatan** | **Bulan Ke** | | | | | | | | | | | | | | | | |
| **1** | | | | **2** | | | | **3** | | | | **4** | | | | |
| 1 | Penelitian Pendahuluan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Pengumpulan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Analisa |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Perancangan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Implementasi Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Pengujian Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Pembuatan Laporan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

#### Tempat Penelitian

Objek Penelitian di lakukan di Stasium Klimatologi Kelas II Padang Pariaman aygn beralamat di Jalan Raya Padang – Bukittinggi KM. 51 Kapalo Hilalang Sumatera Barat.

#### Metode Penelitian

Metode pada Penelitian ini menggunakan metode kuantitatif yang terdiri dari studi literatur, wawancara, interview, pengumpulan data. Dengan penjelasan yang di uraikan sebagai berikut :

##### Studi Literatur

Pada tahap ini peneliti melakukan studi literatur dengan membaca buku, jurnal, makalah dan laporan penelitian yang terkait dengan topik Penelitian. Tujuannya adalah untuk memperkuat permasalahan serta sebagai dasar teori dalam melakukan studi dan juga menjadi dasar untuk melakukan Penelitian terhadap prediksi curah hujan.

##### Wawancara

Pada tahap ini peneliti dan responden berhadapan langsung (face to face) untuk mendapatkan informasi secara lisan dengan tujuan mendapatkan data yang dapat menjelaskan permasalah yang berhubungan dengan penelitian. Wawancara dilakukan dengan dengan supervisor di Stasiun Klimatologi Kelas II Padang Pariaman terkait dengan permasalahan untuk proses menganalisis masalah, menentukan informasi yang di butuhkan oleh peneliti, seperti pengambilan data, dan lain-lain.

##### Identifikasi Data

Data yang diidentifikasi peneliti berupa mengumpulan data atau informasi-informasi yang didapat oleh peneliti dari sesuai wawancara yang di lakukan peneliti, data klimatologi yang di gunakan untuk Penelitian ini di peroleh dari Stasiun Klimatologi Kelas II Padang Pariaman dari tahun 1985-2021. Atribut yang digunakan yaitu curah hujan, kecepatan angin, temperatur, dan kelembapan udara. Data tersebut dijadikan acuan sebagai informasi dari masa lalu yang mana akan digunakan pada proses pelatihan data (data training) menggunakan *Deep Learning* dengan metode *Long Short-Term Memory (LSTM)*.

##### Preprocessing Data

Preprocessing Data merupakan proses pembersihan data dan persiapan data yang akan di gunakan sebagai pelatihan data melalui sebuah proses untuk menangani data yang hilang atau kosong dengan berbagai cara seperti mencari rata-rata suatu atribut untuk kelas yang sama, sekaligus mengidentifikasi dan membuang variabel data yang tidak di perlukan. Data akan melewati 4 proses yaitu subtotal pada persiapan data, deteksi outlier, difference dan scaling.

##### Implementasi Long Short Term Memory (LSTM)

Pada penelitian ini LSTM digunakan karena data yang akan diolah adalah data rentang waktu / time series dan LSTM sangat cocok untuk memprediksi rentang waktu ketika ada langkah-langkah waktu dengan ukuran yang bebas. Salah satu keunggulan penting LSTM adalah kemampuan mengingat dari sekuens long term (ukuran data) yang sulit dicapai dengan teknik fitur tradisional. Pada penelitan ini menggunakan 4 layer LSTM yaitu 1 input layer, 2 hidden layer, dan 1 output layer. Proses dari implementasi model LSTM yaitu data akan masuk pada input layer pada model, kemudian terjadi proses perhitungan dalam sel LSTM.

##### Uji Validitas Root Mean Squared Error (RMSE)

Pengujian dilakukan menggunakan data dari Stasiun Klimatologi Kelas II Padang Pariaman untuk melihat prediksi curah hujan kota Malang dan untuk melihat nilai kesalahan dari peramalan digunakan rumus RMSE (Root Mean Squarred Error). *RMSE* merupakan salah satu contoh parameter yang biasa digunakan sebagai indikator untuk mengukur dan membandingkan kemiripan hasil prediksi dan data asli. *RMSE* merupakan nilai rata-rata dari jumlah kuadrat kesalahan / error.

### Analisa

Analisa dalam penelitian adalah penjabaran dari suatu masalah dari objek yang di teliti yang akhirnya menghasilkan suatu kesimpulan, dalam penelitian ini dimaksudkan untuk mengidentifikasi masalah dalam memprediksi curah hujan berdasarkan data harian klimatologi Dalam proses analisa terdapat tiga tahapan analisa yang harus dilakukan yakni tahapan analisa data, analisa proses dan analisa sistem dengan penjelasan sebagai berikut :

#### Analisa Data

Analisis data adalah sebuah proses untuk memeriksa, membersihkan, mengubah, dan membuat pemodelan data dengan maksud untuk menemukan informasi yang bermanfaat sehingga dapat memberikan petunjuk bagi peneliti untuk mengambil keputusan terhadap pertanyaan-pertanyaan penelitian. Data dalam Penelitian ini di dapat langsung dari Stasiun Klimatologi Kelas II Padang Pariaman. Data yang di dapat berupa :

* Temperatur minimum
* Temperatur maksimum
* Temperatur rata-rata
* Kelembapan rata-rata
* Curah hujan
* Lamanya penginaran matahari
* Kecepatan agnin maksimum
* Arah angin saat kecepatan maksimum
* Kecepatan angin rata-rata
* Dan arah angin terbanyak

Tidak semua data variabel di atas akan masuk proses pengujian nantinya, hanya beberapa variabel data yang di butuhkan dalam Penelitian ini seperti *Temperatur rata-rata, Kelembapan rata-rata, Curah hujan, Kecepatan angin rata-rata.*

#### Analisa Proses

Pada Analisa Proses peneliti akan menentukan nilai epoch dan batch untuk metode Long Short Term Memory pada data yang sudah di pre-processing yang kemudian output/hasil dari metode tersebut.

#### Analisa Sistem

Selanjutnya pada tahap analisis sistem ini merupakan tahapan yang sangat kritis dan sangat penting, di karenakan nilai output pada proses sebelumnya perlu di lakukan perbandingan dengan realita, dilakukan proses Analisis Sistem yang mana data yang telah diproses menggunakan metode Long Short Term Memory akan di lakukan uji validitas dengan melihat nilai error pada standard diviasi agar peneliti dapat melihat nilai ketepatan prediksi yang di lakukan.

### Perancangan

#### Perancangan Model

Tahapan perancangan ini, peneliti mengunakan *Unified Modeling Language (UML)* sebagai tools dalam menjelaskan alur analisa yang akan dibuat. Tahapan perancangan bertujuan untuk membuat penelitian dirancang sesuai dengan tujuannya, sehingga tidak melenceng dari tujuan Penelitian. Perancangan akan menggunakan UML sebagai model rancangan agar terorganisasi dan terstruktur dengan rancangan.

##### Usercase Diagram

*Usecase Diagram* memiliki duaaktoryaitu, admin yang bertugas mengelolah sistemdan manajer yang bertugas mengelola data karyawan dan melakukan penilaian untuk pemilihan kenaikan gaji.

##### Class Diagram

*Class Diagram* menggambarkan bagaimana strukturdari aplikasi yang akan dibentuk dalam merancang aplikasi data mining.

##### Sequence Diagram

Sequence Diagram menjelaskan urutan-urutan kejadian yang akan terjadi di dalam aplikasi pada saat manajer menggunakan aplikasi Data Mining yang telah dirancang, serta untuk menggambarkan interaksi antara keseluruhan objek di dalam aplikasi nantinya. Sequence Diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu dalam bentuk gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan output sesuai dengan Use Case diagram. Sequence Diagram juga digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan oleh manajer nantinya dalam melakukan pemilihan kenaikan gaji dari aplikasi untuk menghasilkan output atau keputusan.

##### State Machine Diagram

State Machine Diagram atau lebih dikenal Statechart Diagramdigunakan untuk menggambarkan perubahan status yang akan terjadi dari aplikasi Data Minig dan objeknya selama aplikasi Data Mining dijalankan sampai dengan akhir.

##### Activity Diagram

ActivityDiagram terbagi menjadi dua yaitu ActivityDiagram admin yang menggambarkan segala aktivitas yang dilakukan oleh admin terhadap sistem yang berupa aktivitas mengelola databasedan aplikasi Data Mining, dan ActivityDiagrammanajer yang menggambarkan segala aktivitas yang biasa dilakukan oleh manajer terhadap sistem. Diagram ini berhubungan dengan Statechart Diagram. Jika Statechart Diagram berfokus pada objek yang dalam suatu proses (atau proses menjadi suatu obyek), ActivityDiagram berfokus pada aktifitas- aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas- aktifitas yang dilakukan oleh admin dan manajer.

##### Deployment Diagram

Deployment Diagram menunjukan konfigurasi komponen didalam proses eksekusi aplikasi. Sistem atau aplikasi Data Mining akan berhubungan dengan web server untuk melakukan proses pemanggilan database sehingga manajer akan mendapatkan keputusan sesuai dengan pengolahan yang telah dilakukan oleh manajer berhubungan pendistribusian barang.