# BAB II TINJAUAN PUSTAKA

## Penelitian Terdahulu

Penelitian pertama dilakukan oleh (Taufiq *et al.*, 2019) dengan judul “*Perbandingan Performa Metode Klasifikasi SVM, Neural Network, Dan Naïve Bayes Untuk Mendeteksi Kualitas Pengajuan kredit Di Koperasi Simpan Pinjam*”. Dalam penelitian tersebut, peneliti menggunakan tiga metode klasifikasi yaitu *SVM*, *Neural Network*, dan *Naïve Bayes* untuk menemukan metode dengan performa yang paling baik dan optimal pada kasus pendeteksian kualitas kredit di koperasi simpan pinjam.

Proses yang dilakukan adalah dengan mengimplementasikan data hasil *preprocessing* menggunakan algoritma *SVM*, *Neural Network*, dan *Naïve Bayes* dengan proses evaluasi menggunakan *5-fold* *cross validation*. Hasil yang didapatkan adalah metode *Neural Network* menjadi metode dengan performa paling baik. Rerata tingkat akurasi yang dihasilkan sebesar 86,81%, rerata precision sebesar 0,8194, rerata recall sebesar 0,8236, dan rerata nilai *AUC* sebesar 0,9158.

Selanjutnya penelitian kedua dilakukan oleh (Desyanita *et al.*, 2020) dengan judul “*Pemodelan Sistem Prediksi Kelayakan Pengajuan kredit Kepemilikan Rumah Dengan Metode C4.5 Dan Naive bayes*”. Dalam penelitian tersebut, peneliti mendapatkan bahwa masalah yang sering muncul dari penyediaan kredit untuk pembayaran rumah adalah kesulitan dalam menentukan pelanggan yang memiliki prospek bagus untuk dapat menyelesaikan kredit dengan baik, juga masih banyaknya kredit yang masuk dalam kategori bermasalah. Selanjutnya peneliti menemukan bahwa salah satu penyebab kredit bermasalah adalah kesalahan dalam melakukan asesmen pada penentuan pemberian kredit.

*Data mining* adalah proses yang digunakan untuk menganalisis kasus untuk menemukan kinerja terbaik dari suatu algoritma diuji. Salah satu cara untuk mendapatkan informasi atau pola dari kumpulan data adalah dengan menggunakan teknik dalam *data mining*. Ada banyak metode klasifikasi yang dapat digunakan untuk menghasilkan nilai akurasi yang tepat. Dalam penelitian ini, dua klasifikasi Metode algoritma digunakan dalam mengklasifikasikan rumah mengkredit *dataset*, yaitu algoritma pohon keputusan *C4.5* dan algoritma *Naïve Bayes*. Perbandingan keduanya algoritma menghasilkan nilai akurasi untuk *Naïve Bayes* algoritma 36,36% dan algoritma *Decision Tree*  *C4.5* memiliki tingkat akurasi 59,54%.

Penelitian ketiga dilakukan oleh (Dewi Sri Mulyani *et al.*, 2020) dengan judul “*Klasifikasi Penentuan Kelayakan Pemberian kredit Menggunakan Metode Naive bayes Classifier”.* Dalam penelitian tersebut peneliti menggunakan metode *data mining* dan algoritma *naive bayes* *classifier*. Klasifikasi *data mining* dapat membantu para analis kredit dalam hal menentukan signifikansi dan kelayakan pemberian kredit pada nasabah. Sehingga dari proses klasifikasi ini didapatkan berupa atribut penentu berupa kategori kelayakan dalam hal penentuan pemberian kredit kepada salah satu nasabah yang bersangkutan.

Pengujian yang dilakukan yakni dengan menggunakan model *Confusion matrix* yang melibatkan data *training* yang berbeda dalam hal Jumlah atribut, dimana eksperimen pertama pengujian dilakukan terhadap 16 atribut data *training* kemudian eksperimen kedua pengujian dilakukan dengan melibatkan 9 atribut data *training.* Dan dari kedua eksperimen yang dilakukan akan diperoleh komparasi nilai akurasi, pengujian 16 atribut menghasilkan signifikansi akurasi 59,00% dan pengujian 9 atribut menghasilkan nilai akurasi sebesar 56,00 % sehingga dari kedua pengujian yang dilakukan dapat dipilih alternatif akurasi yang paling baik yakni pengujian dengan 16 atribut.

Penelitian keempat dilakukan oleh (Rahmawati *et al.*, 2021) dengan judul “*Pengembangan Model Persetujuan kredit Nasabah Bank Dengan Algoritma Klasifikasi Naïve Bayes, Decision Tree , Dan Artificial neural network*”. Dalam penelitian tersebut peneliti menggunakan tiga algoritma yaitu *Naïve Bayes*, *Decision Tree* , *Artificial neural network (ANN)* untuk mengklasifikasikan data profil nasabah yang memiliki peluang untuk mengajukan kredit pinjaman. Evaluasi hasil ketiga algoritma yang digunakan dilakukan dengan melihat nilai *gain ratio*, *confusion matrix*, dan kurva *Receiver Operating Characteristics* (*ROC*) dengan melihat nilai *Area Under Curve (AUC)* yang dihasilkan. Hasil yang didapatkan menunjukkan bahwa algoritma *ANN* menghasilkan nilai akurasi tertinggi dengan akurasi sebesar 99.61% dan *AUC* = 0.983. Disusul dengan *Decision Tree*  menghasilkan nilai akurasi sebesar 99.36% dan *AUC* = 0.999. Terakhir, algoritma *Naïve Bayes* dengan nilai akurasi = 90.79% dan *AUC* sebesar 0.935.

Karakteristik nasabah yang memiliki peluang untuk mengajukan kredit jika income rata-rata pada rentang 101-116 atau > 116 USD setiap bulannya, *age* = 23 - 43 tahun, *family* >4, memiliki *CD* *Account*, dan CC *Average* =3 – 9 USD. Terdapat tujuh faktor importance yang mempengaruhi hasil tiga algoritma tersebut yaitu *income, mortgage, family, CD account, CC average, education,* dan *age.*

Penelitian kelima dilakukan oleh (Hadianto *et al.*, 2019) dengan judul “*Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode Neural Network*”. Dalam penelitian tersebut peneliti menggunakan model *Neural Network*, yang bertujuan untuk melakukan klasifikasi terhadap nasabah yang layak untuk mendapat pinjaman, dengan memperhitungkan parameter yang ada seperti usia, Jumlah pendapatan, Jumlah keluarga, rata-rata pengeluaran per bulan, tingkat pendidikan dan lainnya. Dan kemudian untuk menilai akurasi pengolahan data menggunakan rapid miner kemudian dilanjutkan dengan pengukuran menggunakan *confusion matrix*, kurva *ROC*. Hasil algoritma *Neural Network* setelah melalui pengujian *confusion matrix*, kurva *ROC* menunjukkan nilai akurasi yang sangat tinggi, dan nilai dominan *AUC* dan algoritma. Nilai akurasi adalah 98,24% dengan *AUC* sebesar 0,979.

Penelitian keenam dilakukan oleh (Dewi, 2019) dengan judul “Komparasi Metode Algoritma *Data mining* pada Prediksi Uji Kelayakan *Credit Approval* pada Calon Nasabah kredit Perbankan”. Dalam penelitian ini, peneliti menggunakan algoritma klasifikasi yang sering digunakan untuk klasifikasi calon nasabah antara lain *Neural Network*, *Naive bayes* dan *KNN* dalam prediksi keberhasilan marketing dalam menentukan kelayakan dari nasabah peminjam kredit bank dari uji coba yang dilakukan maka algoritma *Neural Network* lah yang lebih akurat dengan akurasi 90,71% dengan nilai *AUC* 0.880,hal ini dapat menjadi perbandingan *data mining* klasifikasi Melihat nilai *AUC* dari ketiga metode tersebut yaitu *NN*, *Naive bayes* dan *KNN,* maka 3 algoritma tersebut termasuk kelompok klasifikasi baik karena nilai *AUC* -nya antara 0.80 -1.00.

Berdasarkan penelitian yang pernah dilakukan sebelumnya, terdapat beberapa perbedaan (*Research Gap*) dengan penelitian yang akan dilakukan. Diantaranya adalah penelitian sebelumnya belum ada yang berfokus untuk mengklasifikasikan calon nasabah untuk proses pemberian penawaran kredit*,* selanjutnya adalah model yang akan dibuat menggunakan algoritma *artificial neural network* yang berbeda dengan penelitian sebelumnya karena menggunakan data *training* dan juga arsitektur model yang berbeda, tentunya ini juga akan berpengaruh terhadap tingkat akurasi model dalam melakukan prediksi, dan yang terakhir penelitian ini akan memfokuskan agar model dapat diimplementasikan langsung ke dalam *website* perusahaan (BPR Amira).

**Tabel 2.1** Tabel *Research Gap*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Judul, Peneliti, Tahun Terbit | Variabel | Metode Penelitian | Hasil | *Research Gap* |
| 1 | “*Perbandingan Performa Metode Klasifikasi SVM, Neural Network, Dan Naïve Bayes Untuk Mendeteksi Kualitas Pengajuan kredit Di Koperasi Simpan Pinjam*”, (Taufiq *et al.*, 2019) | Kreditkoperasi simpan pinjam | Kuantitatif | Penelitian tersebut menggunakan tiga metode klasifikasi *data mining* yakni *SVM*, *Neural Network* dan *Naive bayes* untuk mengklasifikasikan kualitas kredit di koperasi simpan pinjam. Dari hasil penelitian diperoleh metode *Neural Network* memiliki tingkat akurasi sebesar 86,81%. | 1. penelitian sebelumnya belum ada yang berfokus untuk mengklasifikasikan calon nasabah untuk proses pemberian penawaran kredit.  2. model yang akan dibuat menggunakan algoritma *artificial neural network* yang akan lebih dikembangkan dengan data *training* dan juga arsitektur model yang berbeda serta pemberian label yang disesuaikan dengan kebutuhan dan peraturan yang ada di BPR Amira.  3. penelitian ini akan memfokuskan agar model dapat diimplementasikan langsung ke dalam *website* perusahaan (BPR Amira) |
| 2 | “*Pemodelan Sistem Prediksi Kelayakan Pengajuan kredit Kepemilikan Rumah Dengan Metode C4.5 Dan Naive bayes*”,(Desyanita *et al.*, 2020) | Kreditkepemilikan rumah | Kuantitatif | Dalam penelitian tersebut peneliti mengklasifikasi pelanggan yang memiliki prospek bagus untuk dapat menyelesaikan kredit dengan baik, dengan menggunakan metode *C4.5* dan *Naive bayes*, dari hasil penelitian tersebut diperoleh bahwa nilai akurasi untuk algoritma *Naive bayes* sebesar 36,36% dan algoritma *Decision Tree*  *C4.5* memiliki akurasi 59.54%. |
| 3 | “*Klasifikasi Penentuan Kelayakan Pemberian kredit Menggunakan Metode Naive bayes Classifier”,* (Dewi Sri Mulyani *et al.*, 2020) | Kreditperbankan | Kuantitatif | Dalam penelitian tersebut peneliti menggunakan metode *data mining* dan algoritma *naive bayes* *classifier* untuk menentukan kelayakan pemberian kredit terhadap calon nasabah, dari hasil pengujian menggunakan *Confusion matrix* terhadap 16 atribut data *training* diperoleh akurasi sebesar 59%, dan terhadap 9 atribut data *training* diperoleh akurasi sebesar 56%. |
| 4 | “*Pengembangan Model Persetujuan kredit Nasabah Bank Dengan Algoritma Klasifikasi Naïve Bayes, Decision Tree , Dan Artificial neural network*”,(Rahmawati *et al.*, 2021) | KreditPerbankan | Kuantitatif | Dalam penelitian tersebut peneliti menggunakan tiga algoritma yaitu *Naïve Bayes*, *Decision Tree* , *Artificial neural network (ANN)* untuk mengklasifikasikan data profil nasabah yang memiliki peluang untuk mengajukan kredit pinjaman. Hasil yang didapatkan menunjukkan bahwa algoritma *ANN* menghasilkan nilai akurasi tertinggi dengan akurasi sebesar 99.61% dan *AUC* = 0.983. Disusul dengan *Decision Tree*  menghasilkan nilai akurasi sebesar 99.36% dan *AUC* = 0.999. Terakhir, algoritma *Naïve Bayes* dengan nilai akurasi = 90.79% dan *AUC* sebesar 0.935. |
| 5 | “*Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode Neural Network*”,(Hadianto *et al.*, 2019) | KreditPerbankan | Kuantitatif | Dalam penelitian tersebut peneliti menggunakan model *Neural Network*, yang bertujuan untuk melakukan klasifikasi terhadap nasabah yang layak untuk mendapat pinjaman, kemudian pengukuran hasil penelitian menggunakan *confusion matrix*, kurva *ROC*. Hasil algoritma *Neural Network* setelah melalui pengujian *confusion matrix*, kurva *ROC* menunjukkan nilai akurasi yang sangat tinggi, dan nilai dominan *AUC* dan algoritma. Nilai akurasi adalah 98,24% dengan *AUC* sebesar 0,979. |
| *6* | *“Komparasi Metode Algoritma Data mining pada Prediksi Uji Kelayakan Credit Approval pada Calon Nasabah kredit Perbankan”,*(Dewi, 2019) | KreditPerbankan | Kuantitatif | Dalam penelitian ini, peneliti menggunakan algoritma klasifikasi yang sering digunakan untuk klasifikasi calon nasabah antara lain *Neural Network*, *Naive bayes* dan *KNN* dalam prediksi keberhasilan marketing dalam menentukan kelayakan dari nasabah peminjam kredit bank dari uji coba yang dilakukan maka algoritma *Neural Network* lah yang lebih akurat dengan akurasi 90,71% dengan nilai *AUC* 0.880. |

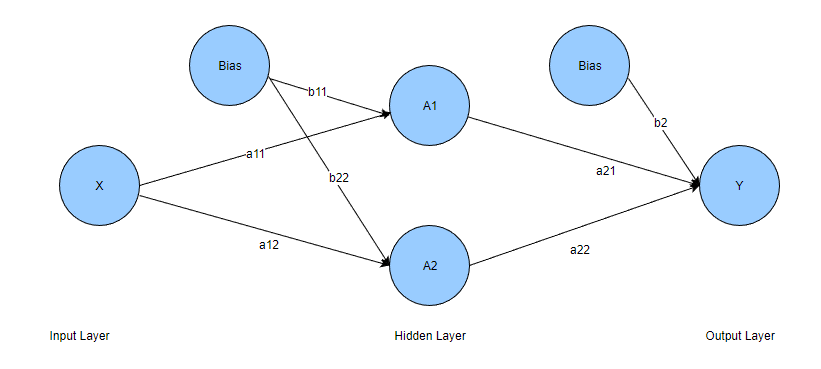
## Teori Terkait

* + 1. **Pengertian *Artificial neural network***

*Artificial neural network* adalah alat pemodelan data statistik dan non-statistik yang dapat memodelkan hubungan kompleks antara input dan output untuk menemukan pola dalam data.  (Erna Suryani *et al.*, 2021).

*Artificial neural network* meniru konsep sistem jaringan syaraf pada tubuh manusia, dimana terdapat bagian yang bernama *node* dan saling berhubungan satu dengan yang lainnya. Model *Artificial neural network* yang akan dibuat terdiri dari *input layer*, *hidden layer* dan *output layer* serta menggunakan fungsi aktivasi *ReLU* yang merupakan fungsi aktivasi default pada model *Artificial neural network* dan juga fungsi *Sigmoid* yang digunakan pada proses *backpropagation*.

Terdapat dua bagian utama dari proses pembelajaran *Artificial neural network* yakni *Forward Propagation* dan *Backward Propagation* yang mana pada *Forward Propagation* model akan mencoba menghasilkan output berupa nilai y, sedangkan pada *Backward Propagation* model akan mencoba untuk melakukan perbaikan dengan cara mengubah nilai bobot *w* yang diharapkan agar dapat memperoleh nilai output *y* yang lebih baik dari sebelumnya.



Gambar 2.1 Arsitektur *ANN*

* + 1. **Pengertian *Flowchart***

*Flowchart* adalah representasi sistematis dari proses dan logika operasi pemrosesan data, atau representasi grafis dari langkah-langkah program dan urutan operasi. *Flowchart* adalah diagram yang secara logis mewakili suatu proses /aliran dalam suatu program atau sistem. *Flowchart* terutama digunakan sebagai alat bantu komunikasi dan dokumentasi  (Rejeki dan Tarmuji, 2013).

Proses-proses tersebut digambarkan dengan simbol-simbol khusus yang saling terhubung. *Flowchart* akan sangat memudahkan dalam proses analisa pemecahan suatu masalah, juga berguna sebagai alat untuk berkomunikasi antara seluruh tim pengembang aplikasi dan *user*. Simbol-simbol yang digunakan dalam *flowchart* merupakan simbol standar yang dikeluarkan oleh *ANSI* dan *ISO*. Berikut ini merupakan beberapa simbol *flowchart* yang digunakan dalam menggambar suatu *flowchart*:

**Tabel 2.2** Simbol *Flowchart*

| **SIMBOL** | **NAMA** | **FUNGSI** |
| --- | --- | --- |
|  | *Terminal* | Menyatakan permulaan atau akhir suatu program |
|  | *Input /Output* | Menyatakan proses input atau output tanpa tergantung peralatannya |
|  | *PROCess* | Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer |
|  | *Decision* | Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya / tidak |
|  | *Connector* | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama |
|  | *Offline Connector* | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda |
|  | *Predefined PROCess* | Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal |
|  | *Punch Card* | Menyatakan input berasal dari kartu atau output ditulis ke kartu |
|  | *Punch Tape* | Digunakan ketika beberapa cabang input menyatu menjadi satu proses |
|  | *Document* | Mencetak keluaran dalam bentuk dokumen (melalui printer) |
|  | *Flow* | Menyatakan jalannya arus suatu proses |

* + 1. ***Unified Modeling Language* (*UML*)**

*Unified Modeling Language (UML)* adalah bahasa berbasis gambar untuk memvisualisasikan, mendefinisikan, membuat, dan mendokumentasikan sistem perangkat lunak berbasis objek. *Unified Modeling Language (UML)* bukanlah bahasa pemrograman, tetapi model yang dibuat berhubungan langsung dengan berbagai bahasa pemrograman, yang memungkinkan pemetaan langsung model *Unified Modeling Language (UML)* ke bahasa pemrograman. seperti *Java* (Syafitri, 2016).

Berikut ini adalah berbagai jenis *UML*:

* + - 1. ***Use case Diagram***

Diagram ini menunjukkan fungsionalitas sistem atau kelas dan bagaimana sistem itu berinteraksi dengan dunia luar, misalnya membuat daftar layanan kesehatan. *Use case* diagram dapat digunakan untuk menjelaskan kebutuhan sistem dan memahami bagaimana sistem seharusnya bekerja  (Syafitri, 2016).

*Use case Diagram* ini bertujuan untuk memetakan apa saja kebutuhan terhadap sistem, juga kebutuhan diluar sistem, dan juga untuk mempresentasikan interaksi yang terjadi antara pengguna terhadap sistem.

Dan berikut adalah simbol-simbol yang ada pada *Use case Diagram*:

**Tabel 2.3** Simbol *Use case Diagram*

| **Simbol** | **Nama** | **Aktor** |
| --- | --- | --- |
| Actor | *Actor* | Merupakan peran  orang,  sistem yang lain, atau alat ketika berhubungan  dengan *use case*. |
| Use Case | *Use case* | Abstraksi dari penghubung  antara aktor dengan  *use case*. |
|  | *Association* | Abstraksi dari penghubung  antara aktor dengan *use case*. |
| Generalisasi | *Include Relationship* | *Include*, merupakan  di dalam *use case*  lain (*required*) atau  pemanggilan *use*  *case* oleh *use case*  lain, contohnya  adalah pemanggilan  sebuah fungsi  program. |
| C:\Users\BISMILLAH NAWAITU\Pictures\image.jpg | *Extend Relationship* | *Extend*, merupakan  *Perl*uasan dari *use*  *case* lain jika kondisi  atau syarat terpenuhi. |

* + - 1. ***Activity Diagram***

*Activity Diagram*s menggambarkan aliran aktivitas yang berbeda dari sistem yang direncanakan, bagaimana setiap aliran dimulai, keputusan apa yang dapat dibuat, dan bagaimana aktivitas berakhir. *Activity Diagram*s juga dapat mewakili proses paralel yang dapat terjadi dalam beberapa aktivitas. Fungsi dapat diimplementasikan dengan satu atau lebih *use case*. Aktivitas menjelaskan proses yang sedang berlangsung, sedangkan *use case* menjelaskan bagaimana aktor menggunakan sistem untuk melakukan aktivitas  (Syafitri, 2016).

Dan berikut ini adalah simbol atau komponen dari *Activity Diagram*:

**Tabel 2.4** Simbol *Activity Diagram*

| **Simbol** | **Nama** | **Keterangan** |
| --- | --- | --- |
| https://2.bp.blogspot.com/-8iXGpasv5sw/XoAiSJCySPI/AAAAAAAABAk/o5LrrfHszX08EvQpZo9leNGDbRfaNU3FwCLcBGAsYHQ/s1600/28.JPG | *Status awal* | Sebuah diagram  Aktivitas memiliki sebuah status awal. |
| https://2.bp.blogspot.com/-U60e55A9_wE/XoAiWjWmqQI/AAAAAAAABAo/qr_m_Iqz22ADwVXJD9Lq9MnvGXQXPAASwCLcBGAsYHQ/s1600/29.JPG | *Aktivitas* | Aktivitas yang  dilakukan sistem,aktivitas  biasanya diawali  dengan kata kerja. |
| https://2.bp.blogspot.com/-ienDNaQT5sU/XoAibfP9KUI/AAAAAAAABAs/_Z4g9leMjxoaECeCyjo82rM4d0oHnJeXQCLcBGAsYHQ/s1600/30.JPG | *Percabangan / Decision* | Percabangan  dimana ada pilihan aktivitas  yang lebih dari  satu. |
| https://2.bp.blogspot.com/-tiFv0q8EpH8/XoAiqyNM_BI/AAAAAAAABA0/T6HN29bvqSEdWS-41pw7Dic6V7lEEXlhQCLcBGAsYHQ/s1600/31.JPG | *Penggabungan / Join* | Penggabungan  dimana yang  mana lebih dari  satu aktivitas lalu digabungkan  jadi satu. |
| https://2.bp.blogspot.com/-eL2fJ5XBsig/XoAiw8SN1aI/AAAAAAAABA8/qClHv83BpKAZPDNHT_FkS-OMzKSdLt4TQCLcBGAsYHQ/s1600/32.JPG | *Status Akhir* | Status akhir yang dilakukan  sistem, sebuah  diagram aktivitas  memiliki sebuah  status akhir |
| https://2.bp.blogspot.com/-q8yoa2_yczE/XoAi4VDRYOI/AAAAAAAABBA/fNspEKEDYIIPBV7P-31vonrEquPsgmnGQCLcBGAsYHQ/s1600/33.JPG | *Swimlane* | Swimlane  memisahkan organisasi bisnis  yang bertanggung jawab terhadap aktivitas yang  terjadi |

* + - 1. ***Sequence Diagram***

*Sequence diagram* menggambarkan interaksi antar objek dalam bentuk pesan (*message*) yang digambarkan dalam waktu. Sebuah *Sequence Diagram* terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (item terkait). Pesan direpresentasikan sebagai garis panah dari satu objek ke objek lainnya. Pada fase desain berikutnya, pesan dipetakan ke fungsi/metode kelas  (Syafitri, 2016).

Activation bar menunjukkan lamanya eksekusi sebuah proses Dan berikut ini adalah simbol atau komponen dari *Sequence Diagram*:

**Tabel 2.5** Simbol *Sequence Diagram*

| **Simbol** | **Nama** | **Keterangan** |
| --- | --- | --- |
| Activation box | *Activation box* | Digunakan untuk mempresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugas tertentu |
| Actors | *Actors* | Digunakan untuk mewakili seorang pengguna |
| Object | *Object* | Mendemonstrasikan bagaimana sebuah objek akan berperilaku dalam sebuah sistem. |
| Lifelines | *Lifelines* | Digunakan untuk menunjukkan kejadian berurutan yang terjadi. |
| Option loop | *Option loop* | Digunakan untuk melakukan skenario pemodelan untuk keadan pada kondisi tertentu. |
| Synchronous message | *Synchronous message* | Digunakan sebagai isyarat bila pengirim harus menunggu respon pesan sebelum melanjutkan. |
| Reply message | *Reply message* | Merupakan balasan untuk sebuah panggilan tertentu. |
| Asynchronous Message | *Asynchronous message* | Merupakan pesan yang tidak memerlukan respons sebelum pengirim melanjutkan aktivitas. |
| Message return | *Message return* | Merupakan penggambaran hasil dari pengiriman pesan dan ditunjukkan dengan arah simbol dari kanan ke kiri |
| General | *General* | Komponen ini merepresentasikan entitas tunggal dalam sebuah *Sequence Diagram* yang berupa stereotype atau class (instance). |
| Boundary | *Boundary* | Komponen ini dapat berupa *User* *Interface* atau alat yang digunakan untuk berinteraksi antar sistem. |
| Control | *Control* | Bertugas untuk mengatur arus informasi dalam sebuah skenario sistem. Elemen ini secara umum dapat mengatur perilaku bisnis dari suatu sistem teknis. |
| Entity | *Entity* | Merupakan elemen yang memiliki tugas menyimpan data atau informasi dan umumnya berupa object model atau beans. |

* + 1. ***Python***

*Python* adalah bahasa pemrograman universal yang dirancang khusus untuk membuat kode sumber lebih mudah. *Python* juga memiliki pustaka lengkap yang memungkinkan pengembang membuat aplikasi canggih dengan kode sumber yang tampak sederhana.  (Syarifah, 2018).

*Python* pertama kali muncul pada tahun 1992. *Python* dibuat dengan cara yang relatif intuitif untuk ditulis dan dipahami. Dengan demikian, *python* merupakan bahasa *Coding* yang ideal bagi mereka yang menginginkan perkembangan pesat.

* + 1. ***Tensorflow***

*Tensorflow* adalah pustaka perangkat lunak yang dikembangkan oleh Tim *Google Brain*, organisasi penelitian mesin cerdas *Google*, untuk melakukan pembelajaran mesin dan penelitian jaringan saraf yang dalam. *Tensorflow* menyertakan teknik pengoptimalan terjemahan aljabar komputer yang membuatnya mudah untuk menghitung banyak ekspresi matematika ketika waktu yang diperlukan untuk menyelesaikan perhitungan menjadi masalah.  (Syarifah, 2018).

*Tensorflow* dibuat oleh tim *Google Brain*, di mana *framework* ini merupakan *library* open *source* yang digunakan untuk pemrosesan numerik dan proyek *machine learning* skala besar. *Tensorflow* berisi banyak model dan algoritma untuk pembelajaran mesin, termasuk *deep learning* (jaringan saraf). *Framework* ini dibangun menggunakan *Python* *front-end API* untuk membangun aplikasi untuk menggunakan dan menggunakan *C++* yang menawarkan kinerja eksekusi terbaik.

*Tensorflow* dapat melatih dan menjalankan jaringan saraf untuk mengklasifikasikan tulisan tangan, mengenali gambar/objek, dan menggabungkan kata. Selanjutnya adalah *Recurrent Neural Network*, *natural language processing* (NLP), *PDE* (*Partial Differential Equation*) berbasis simulasi. Yang terpenting, *Tensorflow* dapat digunakan secara luas untuk produksi dengan model proses yang sama dengan data pelatihan.

* + 1. ***Mysql***

*Mysql* merupakan salah satu jenis *DBMS* (*Database* *Management* *system*) yang menggunakan perintah *SQL* (*structured query language*) dan banyak digunakan diseluruh dunia.

*MySQL* dikembangkan oleh *Oracle Corporation* dan merupakan salah satu DBMS *open source* paling populer. Keunggulan *MySQL* adalah kemampuannya menangani database besar dengan performa tinggi dan stabilitas yang baik.

*MySQL* memiliki keunggulan kecepatan dan skalabilitas, yang membuatnya cocok untuk aplikasi yang membutuhkan pemrosesan data yang besar dan cepat. Selain itu, *MySQL* juga mendukung berbagai *platform* seperti *Windows*, *Linux*, *macOS*, sehingga dapat diintegrasikan dengan berbagai sistem operasi populer.

*MySQL* adalah database open source yang populer di seluruh dunia. Ini banyak digunakan sebagai database untuk bahasa pemrograman umum seperti *PHP* dan *Java*. Untuk mempermudah penggunaan *MySQL*, terdapat *software open source* berbasis *GUI* bernama *phpmyadmin* yang dapat diunduh secara gratis dari *www.phpmyadmin.net*. *Phpmyadmin* juga tersedia secara default di *Xampp* (*www.apachefriends.org*), sebuah perangkat lunak yang menggabungkan *Apache, PHP, Mysql dan Perl* dengan modul tambahan  (Sofwan,2011).