

## BAB 2

### LANDASAN TEORI

#### 2.1 Kajian Terhadap Penelitian Yang Terkait Sebelumnya

Berikut kajian dari penelitian sebelumnya yang menjadikan referensi dari melakukan penelitian ini: (Choudhary & Chauhan, 2023) dalam jurnalnya yang berjudul “*An Intelligent Chatbot Design And Implementation Model Using Long Short-Term Memory With Recurrent Neural Networks And Attention Mechanism*” dalam penelitian ini melakukan desain *chatbot* yang di implementasikan menggunakan model LSTM, dimana penelitian ini memiliki tujuan utama yaitu, untuk mensimulasikan percakapan mesin yang mirip dengan manusia. Penelitian ini juga menyatakan jika *chatbot* sebagai kunci untuk meningkatkan pengalaman pengguna dalam melakukan penanganan pelayanan pelanggan pada saat yang bersamaan dengan kecepatan yang konstan. Selain mendesain *chatbot* penelitian ini juga melakukan evaluasi untuk mengukur kualitas terjemahan mesin.

Berikut kajian dari penelitian sebelumnya yang menjadikan referensi dari melakukan penelitian ini: (Anki et al., 2021) dalam jurnal yang berjudul “*Intelligent Chatbot Adapted From Question And Answer System Using RNN-LSTM Model*” permasalahan dari penelitian ini dibuat untuk menjawab pertanyaan dari permasalahan konsumen dengan memilih *chatbot* sebagai solusinya dan memilih program *python* sebagai penyusunan struktur komponen tampilan *chatbot* sehingga pada akhirnya konsumen memperoleh jawaban dari pertanyaan yang diajukan ke *chatbot*.

Berikut kajian dari penelitian sebelumnya yang menjadikan referensi dari melakukan penelitian ini: (Katkade et al., 2020) dalam jurnal yang berjudul “*Implementation Of Artificial Intelligence Based Chatbot System With Long Term Memory*” jurnal ini mengeksplorasi kedalam metode tertentu untuk membangun sebuah agen percakapan. Sistem ini akan berkerja dengan kerangka urutan sebagai perbaikan dari teknologi sebelumnya. Ciri khas dari model penelitian ini yaitu dilatih secara *end to end* sehingga membutuhkan lebih sedikit aturan dengan

model sederhana ini bisa menghasilkan percakapan sederhana mengingat besar kumpulan data pelatihan percakapan.

Berikut kajian dari penelitian sebelumnya yang menjadikan referensi dari melakukan penelitian ini: (Supriyanto et al., 2018) dalam jurnal yang berjudul “Pusat Informasi Di Ruang Publik Terpadu Ramah Anak (RPTRA) Guna Meningkatkan Kecakapan Hidup (Life Skill) Remaja Urban” Universitas Pembangunan Jaya (UPJ) melakukan penelitian serta pengabdian masyarakat dengan tujuan untuk meningkatkan kegunaan pusat informasi yang tersedia di RPTRA Anggrek sebagai sumber rekomendasi dan pengetahuan untuk masyarakat. Hasil yang di dapat dari melakukan penelitian meliputi papan buletin, poster, artikel populer di media sosial dan artikel ilmiah yang diterbitkan dalam jurnal atau publikasi.

Berikut kajian dari penelitian sebelumnya yang menjadikan refrensi dari melakukan penelitian ini: “Aplikasi *Chatbot* MI3 Untuk Informasi Jurusan Teknik Informatika Berbasis Sistem Pakar Menggunakan Metode *Forward Chaining*” *Chatbot* dirancang untuk aplikasi yang mampu berkomunikasi dengan mesin. Komunikasi dalam *chatbot* memberikan informasi kepada user. Informasi yang diberikan berupa tentang akademik. *Chatbot* ini dibuat untuk informasi dalam lingkup jurusan Teknik Informatika UIN Maliki Malang (Nur Baiti & Nugroho, 2013).

Berikut kajian dari penelitian sebelumnya yang menjadikan refrensi dari melakukan penelitian ini: “Aplikasi *Chatbot* Untuk *FAQ* Akademik Di IBI-K57 Dengan LSTM Dan Penyematan Kata” Penelitian ini membangun aplikasi *chatbot* untuk menjawab pertanyaan-pertanyaan umum akademik yang sering ditanyakan untuk mahasiswa di Institut Bisnis dan Informatika Kosgoro 1957 (IBI-K57). Dalam penelitian, model jaringan syaraf tiruan LSTM dilatih sendiri menggunakan *python* dan *jupyter notebook*. LSTM dipilih karena untuk klasifikasi maksud model ini cukup akurat jika dibandingkan dengan RNN dan *K-Nearest Neighbor*. Penelitian ini juga menambahkan penyematan kata agar mampu mengenali kata yang secara sintaks dan semantik mirip.

Perbandingan dari perbedaan implementasi yang dilakukan pada jurnal penelitian sebelumnya yaitu hanya membuat struktur program *chatbot* tanpa diterapkan ke sebuah aplikasi. Maka perbedaan dari aplikasi yang dibuat melakukan implementasi *chatbot* ke dalam *website* sebagai wadah untuk memudahkan pengguna untuk bisa menggunakannya.

## 2.2 *Chatbot*

Menurut Bryan Samperura, Muhammad Sapwan Suhadi (Samperura et al., 2023) *chatbot* merupakan jenis percakapan mesin dengan manusia menggunakan bahasa alami fungsinya sebagai program yang mampu merespon dari setiap pertanyaan pengguna sesuai ruang lingkup dari data yang dipelajari. *Chatbot* biasanya digunakan sebagai alat yang digunakan untuk membantu manusia dalam menjawab pertanyaan-pertanyaan yang biasanya digunakan pada sistem yang berhubungan dengan pelanggan seperti halnya *customer service* dan sistem bantuan. Dalam melakukan proses pembuatan *chatbot* dapat dilakukan beberapa model untuk melakukan pengembangan *chatbot* diantaranya yaitu *Rule-based*, *Static-based*, dan yang terakhir adalah *hybrid*.

*Rule-based* artinya pembuatan *chatbot* ini dibuat sesuai dengan *input* dan *output* dengan alur yang diharapkan. Sedangkan *static-based* artinya *input* di analisa atau dipelajari terlebih dahulu setiap kalimatnya lalu *chatbot* akan merespon sesuai dari inputan yang telah dianalisa secara ringkas *chatbot* ini dibuat secara otomatis untuk mempelajari dan menemukan data yang sesuai dari inputan yang diberikan. Terakhir adalah *hybrid* ini tidak jauh beda dari model sebelumnya karena *hybrid* memadukan kedua model tersebut. Menurut Sameera A. Abdul-Kader dan Dr. John Woods (Abdul-Kader & Woods, 2015) bahwa *chatbot* memiliki tiga tahap yaitu responses, klasifikasi, dan *graphmaster*, yang pertama tahap responses tahapan yang memberikan respon atau jawaban dari setiap inputan dari pengguna lalu setiap inputan pengguna akan melalui proses klasifikasi untuk mendeteksi setiap respon yang diberikan sehingga sistem *chatbot* dalam setiap *input* dan *outputnya* akan lebih terstruktur. Sedangkan klasifikasi melakukan proses *filtering* dan pendeteksian dari memahami setiap kalimat *input*

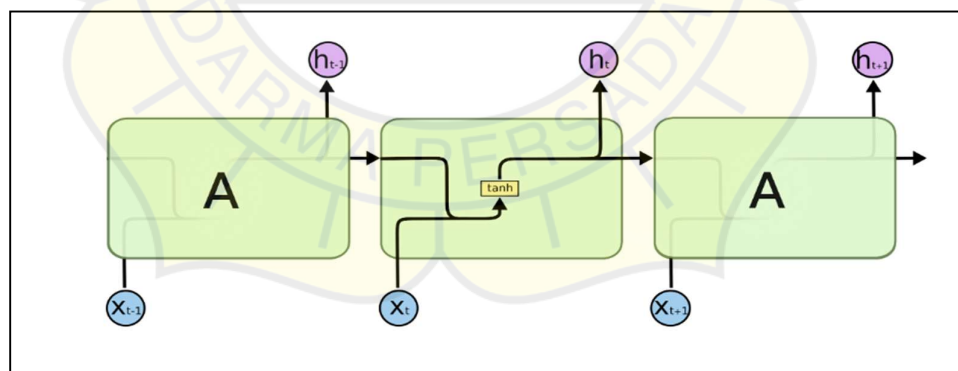
dari pengguna. *Graphmaster* itu melakukan kesamaan kata kunci terhadap masukan yang telah dibuat.

### 2.3 *Natural Language Processing (NLP)*

*Natural Language Processing (NLP)* termasuk ke dalam jenis *Artificial Intelligence (AI)* yang menerapkan bahasa alami antara komputer dengan manusia. NLP ini menggunakan model dengan sistem terkomputasi yang bertujuan untuk memberikan kemudahan dalam melakukan pencarian informasi (Mulyatun et al., 2021).

### 2.4 *Long Short Trem Memory (LSTM)*

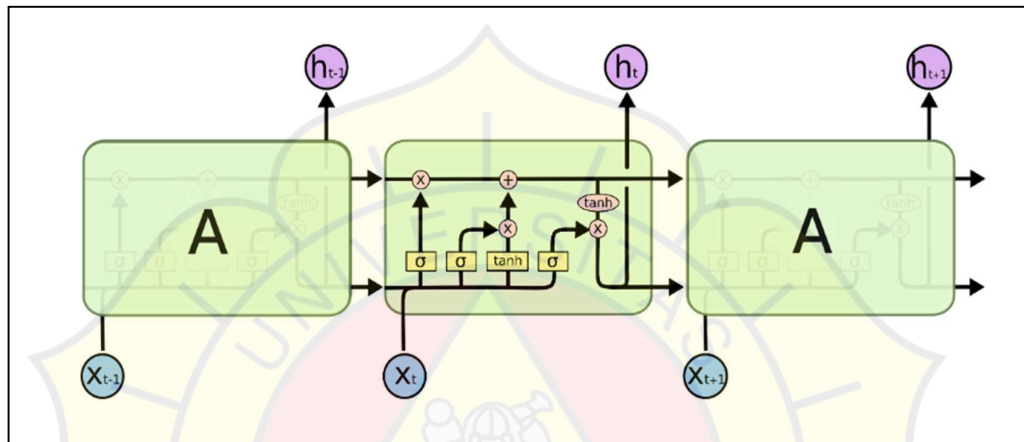
LSTM (*Long Short Term Memory*) masih termasuk ke dalam golongan RNN (*Recurrent Neural Network*) yang memiliki arsitektur untuk mencegah adanya permasalahan terkait ketergantungan dalam waktu yang lama. (Zurayyah et al., 2019). Pada umumnya RNN (*Recurrent Neural Network*) ini memiliki struktur yang lebih sederhana yaitu hanya memiliki satu lapisan yaitu *tanh*. Arsitektur RNN dapat ditunjukkan seperti pada gambar 2.1.



Gambar 2.1 Arsitektur RNN (Olah, 2015)

LSTM (*Long Short Term Memory*) diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997. LSTM (*Long Short Term Memory*) berbeda dengan RNN (*Recurrent Neural Network*) yang memiliki struktur lebih sederhana

dengan satu lapisan yaitu *tanh*. Sedangkan LSTM (*Long Short Term Memory*) memiliki empat lapisan yaitu *cell state* (memori sel), *input gate* (gerbang *input*), *forget gate* (gerbang *lupakan*), *output gate* (gerbang *output*) arsitektur LSTM (*Long Short Term Memory*) dapat ditunjukkan pada gambar 2.2 merupakan penjelasan dari arsitektur LSTM (Olah, 2015).



Gambar 2.2 Arsitektur LSTM (Saxena, 2023)

Pada arsitektur LSTM (*Long Short Term Memory*) terdapat empat komponen utama yang akan dijelaskan dibawah ini yaitu:

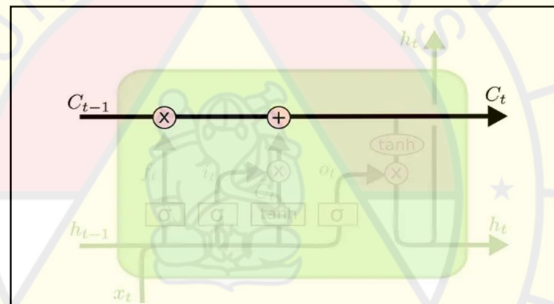
1. *Cell State* (memori sel)

*Cell state* merupakan lapisan pertama yang melewati seluruh unit LSTM (*Long Short Term Memory*), fungsinya untuk mengingat dan melupakan. *Remembering* (mengingat) artinya *cell state* dapat mengingat langkah waktu sebelumnya memungkinkan untuk menyimpan informasi yang relevan dalam jangka panjang. *Forgetting* (melupakan) artinya *cell state* juga sebagai kunci untuk mengontrol informasi lama yang tidak relevan akan dihapus atau dilupakan dari lapisan *cell state*. Operasi pertama yang dilalui oleh *cell state* yaitu (x) sebagai operasi pertama yang mengalikan *cell state* dengan array [-1, 0, 1]. Jika informasi yang didapatkan dikalikan 0 maka dilupakan oleh LSTM. Lalu oprasi selanjutnya adalah (+) yang bertugas untuk menambahkan beberapa informasi

baru ke memori. Maka persamaan dari *cell state* sendiri dapat ditulis sebagai berikut:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}(t)$$

Di sini,  $C_t$  adalah *cell state* pada langkah waktu  $t$ ,  $f_t$  adalah vektor *sigmoid* yang menentukan seberapa banyak informasi dari *cell state* sebelumnya  $C_{t-1}$  akan dipertahankan,  $i_t$  adalah vektor *sigmoid* yang menentukan seberapa banyak informasi baru akan ditambahkan ke *cell state*, dan  $\tilde{C}(t)$  adalah vektor nilai yang berpotensi untuk di *update* ke *cell state*. Gambar 2.3 merupakan penjelasan dari komponen *cell state*.



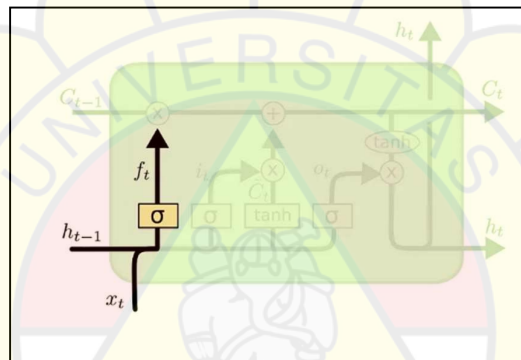
Gambar 2.3 Cell State LSTM (Saxena, 2023)

## 2. Forget Gate (Gerbang Lupakan)

*Forget gate* berfungsi sebagai gerbang LSTM (*Long Short Term Memory*) yang dapat memutuskan informasi mana yang bisa disimpan dan dilupakan. Keputusan yang diambil dari gerbang LSTM (*Long Short Term Memory*) menggunakan lapisan *sigmoid* yang biasa disebut dengan lapisan gerbang lupa. Operasi pertama yaitu melakukan perkalian  $h_{t-1}$  dan  $x_t$  lalu diteruskan dengan bantuan lapisan *sigmoid* ( $\sigma$ ), lapisan *sigmoid* menghasilkan angka antara 0 dan 1 untuk setiap angka yang masuk ke dalam *cell state*  $C_{t-1}$ . Hasil dari lapisan *sigmoid* ini menentukan *output* dari *forget gate* sebelum diteruskan ke *gate* berikutnya, jika hasil yang di dapatkan 1 maka *forget gate* akan menyimpannya jika sebaliknya yaitu 0 maka akan dilupakan sepenuhnya. Maka persamaan dari *forget gate* sendiri dapat ditulis sebagai berikut:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Di mana  $f_t$  merupakan *forget gate* pada langkah waktu  $t$ ,  $\sigma$  merupakan fungsi sigmoid yang menghasilkan nilai antara 0 dan 1,  $W_f$  sendiri merupakan matriks bobot yang digunakan untuk mengalikan input  $[h_{t-1}, x_t]$ ,  $[h_{t-1}, x_t]$  gabungan dari vektor output dari langkah waktu sebelumnya  $h_{t-1}$  dan vektor input saat ini  $x_t$ , terakhir adalah  $b_f$  yang merupakan vektor bias. Gambar 2.4 merupakan penjelasan dari komponen *forget gate*.



Gambar 2.4 *Forget Gate* LSTM (Saxena, 2023)

### 3. *Input Gate* (Gerbang Masuk)

*Input gate* digunakan untuk memberikan informasi baru ke LSTM (*Long Short Term Memory*) dan menentukan apakah informasi baru tersebut akan di simpan dalam status sel. *Input gate* memiliki tiga bagian yaitu *sigmoid*, *tanh*, dan penggabungan dari dua bagian tersebut yaitu *sigmoid* dan *tanh* yang perbarui status sel. Adapun proses *input gate* ini melalui 3 bagian yang yaitu:

- a) Lapisan *sigmoid* ( $\sigma$ ) akan menentukan nilai yang diperbarui, lapisan ini disebut lapisan gerbang masukan.
- b) Lapisan fungsi aktivasi *tanh* membuat vektor nilai kandidat baru  $\tilde{C}(t)$  yang bisa ditambahkan ke status.
- c) Selanjutnya dari lapisan *sigmoid* dan juga lapisan *tanh* ini digabungkan  $i_t \cdot \tilde{C}(t)$  lalu diperbarui ke status sel.

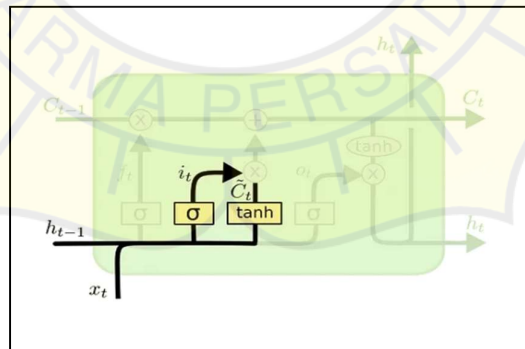
Maka persamaan dari *input gate* sendiri dapat ditulis sebagai berikut:

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

$$\tilde{C}(t) = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Di mana pada persamaan pertama yaitu  $i_t$  adalah input gate pada langkah waktu  $t$ ,  $\sigma$  adalah fungsi sigmoid yang menghasilkan nilai antara 0 dan 1,  $W_i$  adalah matriks bobot yang digunakan untuk mengalikan input  $[h_{t-1}, x_t]$ ,  $[h_{t-1}, x_t]$  adalah gabungan dari vektor output dari langkah waktu sebelumnya  $h_{t-1}$  dan vektor input saat ini  $x_t$ ,  $b_i$  adalah vektor bias.

Selanjutnya persamaan kedua yaitu  $\tilde{C}(t)$  atau *candidate value* adalah nilai kandidat yang akan di *update* ke dalam *cell state* pada langkah waktu  $t$ ,  $\tanh$  adalah fungsi *tangen hiperbolik* yang menghasilkan nilai antara -1 dan 1,  $W_c$  adalah matriks bobot yang digunakan untuk mengalikan input  $[h_{t-1}, x_t]$ ,  $[h_{t-1}, x_t]$  adalah gabungan dari vektor *output* dari langkah waktu sebelumnya  $h_{t-1}$  dan vektor *input* saat ini  $x_t$ ,  $b_c$  adalah vektor bias. Gambar 2.5 merupakan penjelasan dari *input gate*.



Gambar 2.5 *Input Gate* LSTM (Saxena, 2023)

#### 4. *Output Gate* (Gerbang Keluar)

*Output gate* ini sebagai gerbang keluaran dari unit LSTM (*Long Short Term Memory*) yang bergantung pada status sel baru. Tahapan *output gate* menghasilkan keluaran yang pertama, lapisan *sigmoid* menentukan bagian mana



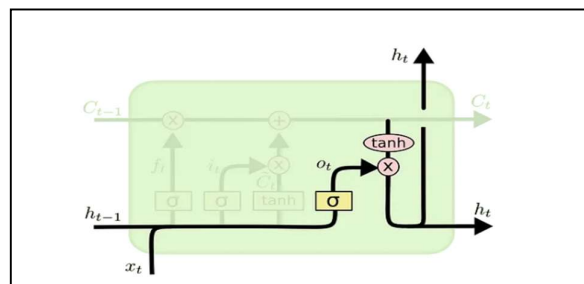
dari keadaan sel yang akan kita hasilkan. Kemudian, lapisan *tanh* pada status sel untuk menekan nilai antara minus satu dan satu, yang akhirnya dikalikan dengan keluaran gerbang *sigmoid*. Operasi pertama yang dilewati dari gerbang ini yaitu lapisan *sigmoid* yang memutuskan bagian mana dari keadaan sel yang akan dihasilkan. Lalu melewati lapisan *tanh* yang digunakan pada status sel untuk menekankan nilai antara -1 dan 1, yang akhirnya dikalikan dengan keluaran gerbang *sigmoid*. Maka persamaan dari *input gate* sendiri dapat ditulis sebagai berikut:

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

$$h_t = o_t \cdot \tanh(C_t)$$

Di mana, pada persamaan pertama untuk mencari hasil dari  $o_t$ ,  $o_t$  adalah *output gate* pada langkah waktu  $t$ ,  $\sigma$  adalah fungsi *sigmoid* yang menghasilkan nilai antara 0 dan 1,  $w_o$  adalah matriks bobot yang digunakan untuk mengalikan *input*  $[h_{t-1}, x_t]$ ,  $[h_{t-1}, x_t]$  adalah gabungan dari vektor *output* dari langkah waktu sebelumnya  $h_{t-1}$  dan vektor *input* saat ini  $x_t$ ,  $b_o$  adalah vektor bias.

Selanjutnya pada persamaan kedua yaitu  $h_t$ ,  $h_t$  adalah *output* dari LSTM pada langkah waktu  $t$ ,  $o_t$  adalah *output gate* pada langkah waktu  $t$ ,  $\tanh(C_t)$  adalah nilai dari *cell state* pada langkah waktu  $t$  yang diproses melalui fungsi *tangen hiperbolik*. Ini menunjukkan seberapa banyak informasi dalam *cell state* diekspos sebagai *output*. Gambar 2.6 merupakan penjelasan dari *output gate* (Saxena, 2023).



Gambar 2.6 *Output Gate* LSTM (Saxena, 2023)

## **2.5 Ruang Publik Terpadu Ramah Anak (RPTRA)**

RPTRA tempat yang diciptakan sebagai tempat layak anak maksudnya tempat ini dibangun yang diperuntukan terutama untuk anak, namun tak hanya anak orang dewasa juga bisa menikmati fasilitas yang ada. RPTRA memiliki tempat ruang publik dan ruang terbuka hijau yang digunakan sebagai tempat beraktivitas anak. RPTRA juga memiliki karakternya tersendiri seperti RPTRA dilengkapi tempat yang ramah anak yaitu berupa wahana permainan yang bisa dinikmati secara publik, lalu ruang terpadu yang bisa digunakan untuk kegiatan belajar selain itu juga sebagai kegiatan sosial bersama warga (Hernowo & Navastara, 2017).

## **2.6 PHP (*Hypertext Processor*)**

PHP (*Hypertext Processor*) merupakan bahasa pemrograman berbasis *web* untuk pengembangan *web*, yang pertama kali di kembangkan oleh Rasmus Lerdof. Pada tahun 1994 Rasmus Lerdof melakukan pengembangan software dan anggota tim Apache. Lalu Rasmus Lerdof melakukan pengembangan yang kedua dengan menambahkan form interpreter untuk penerjemahan perintah SQL. Sejak saat itu PHP menjadi bahasa pemrograman yang banyak diminati oleh pengguna. Sehingga tidak memungkinkan jika hanya dikelola satu orang saja. Maka dibentuklah sebuah tim untuk melakukan pengembangan yang mengorganisir kontributor dari seluruh dunia dengan model pengembangan proyek (Solichin, 2016).

## **2.7 JavaScript**

*JavaScript* merupakan bahasa pemrograman *web* dimana developer bisa menerapkan fitur yang kompleks pada halaman *web*. *JavaScript* sendiri ditemukan oleh Brendan Eich tahun 1995 yang dikembangkan pertama kali oleh *Netscape 2* dan menjadi standar ECMA-262 tahun 1997 (Hendra, 2022). *JavaScript* digunakan untuk menyempurnakan tampilan dan sistem pada halaman *web-based application* yang dikembangkan (Mariko, 2019).

## 2.8 Python

*Python* adalah bahasa pemrograman tingkat tinggi yang serba guna, dinamis, dan berorientasi objek. Dikembangkan pertama kali oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991. *Python* dirancang dengan 15 fokus pada keterbacaan kode, sehingga memudahkan pengembang untuk menulis kode yang lebih bersih dan lebih mudah dipahami (Kadir, 2019).

## 2.9 PhpMyadmin

Nama PhpMyadmin tercampur dari nama PHP yang artinya sebagai bahasa yang digunakan dan MySQL sebagai database yang dikelolanya. PhpMyadmin itu aplikasi *open source* yang fungsinya untuk memudahkan manajemen MySQL. PhpMyadmin ini mendukung operasi MySQL diantaranya untuk membuat database, membuat tabel, menambahkan, menghapus dan mengupdate data (Sofwan, 2007). Perbedaan yang dimiliki dari PhpMyadmin dengan MySQL adalah ada pada fungsinya. Jika PhpMyadmin adalah sebagai alatnya untuk memudahkan dalam mengoperasikan MySQL, sedangkan MySQL sebagai tempatnya untuk menyimpan datanya (Nur Hartiwati, 2022).

## 2.10 UML

UML yang memiliki kepanjangan *Unified Modeling Language* yang artinya sebagai sekumpulan diagram yang dilakukan untuk melakukan penggambaran dari sebuah sistem atau perangkat lunak berbasis objek. Ada beberapa jenis diagram UML yang digunakan untuk memodelkan sistem perangkat lunak berikut adalah beberapa di antaranya:

### 2.10.1 Diagram Use Case

Diagram *use case* salah satu jenis diagram UML yang digunakan untuk menampilkan suatu interaksi antara *use case* dan *actor*. Selain itu juga *use case* digunakan untuk menggambarkan fungsionalitas sistem atau persyaratan yang harus dipenuhi sistem dari pandangan pemakai (Setiawan & Khairuzzaman, 2017). Adapun komponen utama untuk membuat diagram yang dimiliki oleh *use case* terdiri dari 3 jenis yaitu, sistem, *actor*, dan *use case*. Relasi yang digunakan

untuk membuat diagram *use case* yaitu, *association*, *dependency*, dan *generalization* (Adani, 2021).

### **2.10.2 Diagram Activity**

*Activity* diagram adalah jenis UML yang digunakan untuk menggambarkan aliran aktivitas dalam suatu proses sistem berjalan yang sedang dirancang, bagaimana masing-masing dari aliran aktivasi ini berjalan berawal dari, *decision* yang mungkin terjadi, lalu menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi dan bagaimana mereka berakhir (Hendini, 2016).

### **2.10.3 Diagram Sequence**

*Sequence* diagram digunakan sebagai aktivitas atau tindakan yang terjadi dalam suatu proses atau aliran kerja, serta ketergantungan dan urutan aktivitas tersebut. *Activity* diagram berguna untuk merancang, menganalisis, dan mendokumentasikan logika proses bisnis atau fungsi sistem. Untuk membuat diagram urutan, perlu dipahami objek-objek yang terlibat dalam suatu skenario *use case*, bersama dengan metode-metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek itu sendiri. Diagram urutan memerlukan adanya skenario yang terdapat dalam *use case* tersebut agar dapat disajikan secara visual (Setiawan & Khairuzzaman, 2017).



**TEKNOLOGI INFORMASI**  
**UNIVERSITAS DARMA PERSADA**