BAB 3
PELAKSANAAN KERJA MAGANG

Pelaksanaan Kerja Magang ini di tujukan untuk membantu , menegmbangkan Fungsi Jacob Chatbot yang di awasi oleh Dosen pembimbing.

3.1 Kedudukan dan Organisasi

Kedudukan dalam proses magang ini adalah sebagai *Software Engineer* yang bertugas untuk membantu riset dan pengembangan modul sistem rekomendasi pada JacobChatbot. Secara struktural, proses kerja magang diposisikan di bidang pengelolaan teknologi informasi, komunikasi , di bawah pengawasan dosen pembimbing. Koordinasi dilakukan bersama dosen pembimbing yang memberikan informasi, arahan selama proses magang berlangsung. Komunikasi di lakukan lewat *Whatsapp* juga *Zoom* untuk melaporkan *Progress* yang ada setiap minggu.

3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama pelaksanaan kerja magang adalah perncangan sistem rekomendasi dan implementasi kedalam jacob chatbot.

3.3 Uraian Pelaksanaan Magang

Tugas yang dilakukan selama 40 hari periode magang adalah membuat modul sistem rekomendasi.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Diskusi dan memahami arah dan tujuan riset, Instalasi Environtment, aplikasi dan mengulas kode.
2	Mempelajari preseden terkait metode dan algoritma yang akan digunakan
3	Praktik membuat *Machine Learning Model* pada Jupyter Notebook
4	Mempelajari framework *Flask* dalam bahasa python.
5	Membuat *Machine Learning Model* dalam bentuk *Flask*
6	Integrasi, Test , Debugging

Pada minggu pertama, kegiatan yang dilakukan adalah berdiskusi dengan dosen pembimbing untuk memahami arah dan tujuan riset, juga instalasi Environtment aplikasi, juga mengulas ulang kode aplikasi.

Pada minggu kedua, kegiatan yang di lakukan adalah mempelajari preseden terkait metode dan algoritma yang akan digunakan untuk modul tersebut. Pada minggu ketiga, dimulailah praktik pembuatan *Machine Learning Model* dengan menggunakan Jupyter Notebook berdasarkan metode dan algoritma yang sudah ditentukan.

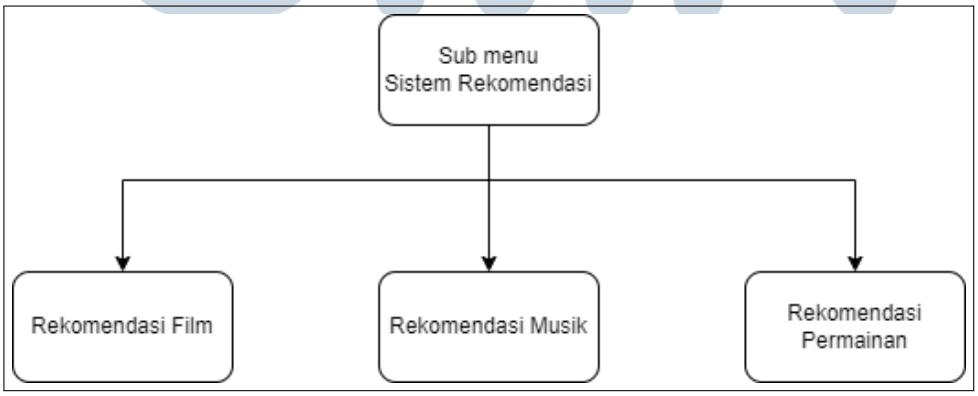
Pada minggu keempat, mempelajari lebih lanjut framework *Flask* dalam bahasa python guna integrasi yang akan di lakukan dengan aplikasi. Pada minggu kelima, dimulailah pembuatan *Machine Learning Model* dalam bentuk *Flask* agar dapat diintegrasikan ke dalam aplikasi. Pada minggu keenam, di mulailah Integrasi modul baru kepada aplikasi, serta menjalankan tes juga debugging.

3.3.1 Perancangan

Perancangan *Machine Learning Model* terbagi dalam beberapa bagian sebagai berikut :

1. Sistem rekomendasi film
2. Sistem rekomendasi musik
3. Sistem rekomendasi permainan

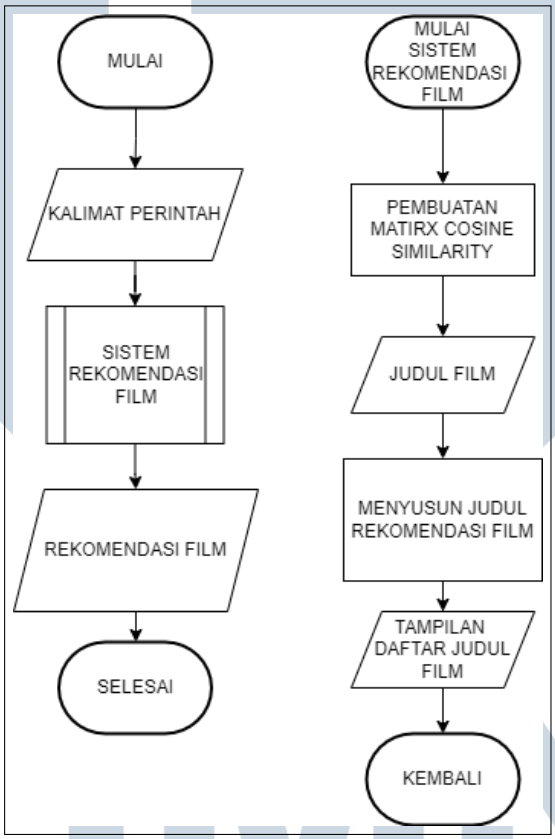
A. User Requirement



Gambar 3.1. Diagram User Requirement

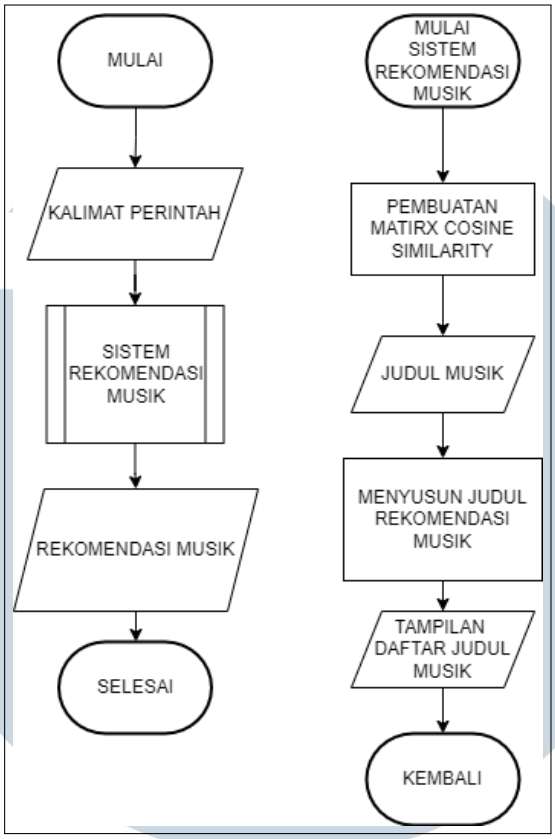
Gambar 3.1 merupakan Diagram *User Requirement* Modul sistem rekomendasi yang terbagi menjadi 3 bidang yaitu rekomendasi film, rekomendasi musik, dan rekomendasi permainan.

B. Flowchart



Gambar 3.2. *Flowchart* sistem rekomendasi film

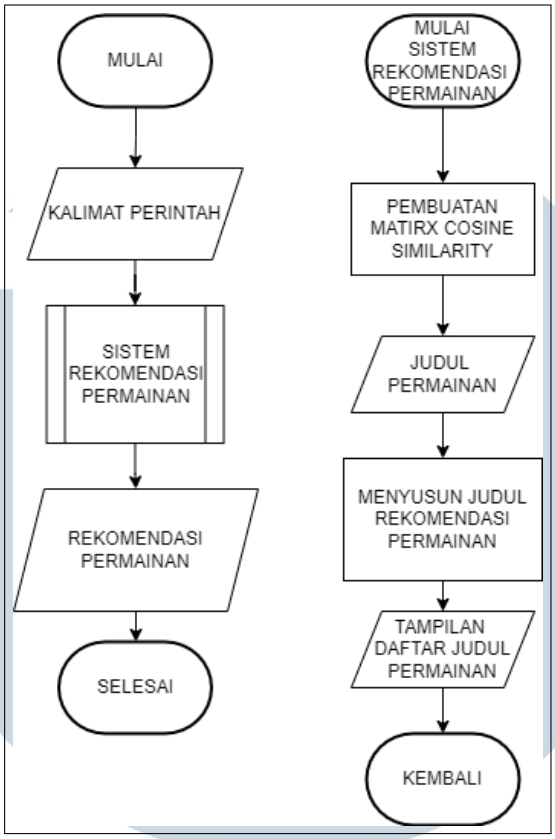
Gambar 3.2 merupakan *Flowchart* sistem rekomendasi film, Sistem akan di picu jika User memberikan pertanyaan atau perintah seperti ”Recommend me a Movie” atau ”Can you Recommend me a Movie ?” lalu sistem akan mengambil film kesukaan *user* dari *databse* sebagai *input* lalu di proses menjadi daftar sejumlah film yang mirip dengan *input user*.



Gambar 3.3. *Flowchart* sistem rekomendasi musik

Gambar 3.3 merupakan *Flowchart* sistem rekomendasi Musik, Sistem akan di picu jika User memberikan pertanyaan atau perintah seperti ”Recommend me a Music” atau ”Can you Recommend me a Music ?” lalu sistem akan mengambil musik kesukaan *user* dari *databse* sebagai *input* lalu di proses menjadi daftar sejumlah musik yang mirip dengan *input user*.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.4. *Flowchart* sistem rekomendasi permainan

Gambar 3.4 merupakan *Flowchart* sistem rekomendasi permainan, Sistem akan di picu jika User memberikan pertanyaan atau perintah seperti ”Recommend me a Game” atau ”Can you Recommend me a Game ?” lalu sistem akan mengambil permainan kesukaan *user* dari *databse* sebagai input lalu di proses menjadi daftar sejumlah permainan yang mirip dengan *input user*.

UNIVERSITAS MULTIMEDIA NUSANTARA

3.3.2 Implementasi

```
In [4]: def get_recommendations(title, cosine_sim=cosine_sim):
            idx = indices[title]
            sim_scores = list(enumerate(cosine_sim[idx]))
            sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
            sim_scores = sim_scores[1:11]
            movie_indices = [i[0] for i in sim_scores]
            tit = df2['title'].iloc[movie_indices]
            dat = df2['release_date'].iloc[movie_indices]
            rating = df2['vote_average'].iloc[movie_indices]
            moviedetails=df2['overview'].iloc[movie_indices]
            movietypes=df2['keywords'].iloc[movie_indices]
            movieid=df2['id'].iloc[movie_indices]

            return_df = pd.DataFrame(columns=['Title','Year'])
            return_df['Title'] = tit
            return_df['Year'] = dat
            return_df['Ratings'] = rating
            return_df['Overview']=moviedetails
            return_df['Types']=movietypes
            return_df['ID']=movieid
            return return_df

In [5]: get_recommendations('Home Alone')

Out[5]:
```

	Title	Year	Ratings	Overview	Types	ID
2086	Home Alone 2: Lost in New York	1992-11-19	6.3	Instead of flying to Florida with his folks, K...	[{"id": 65, "name": "holiday"}, {"id": 242, "n...	772
4726	The Mighty	1998-10-23	7.1	This tells the story of a strong friendship be...	[{"id": 2494, "name": "ohio"}, {"id": 10683, "...	9821
2101	White Chicks	2004-06-23	6.3	Two FBI agent brothers, Marcus and Kevin Copel...	[{"id": 1568, "name": "undercover"}, {"id": 18...	12153
1050	Monster-in-Law	2005-05-13	5.6	Office temp Charlotte Cantilini thinks she's f...	[{"id": 2374, "name": "bad mother-in-law"}, {"...	4379
2826	Time Bandits	1981-07-13	6.6	Young history buff Kevin can scarcely believe ...	[{"id": 1454, "name": "treasure"}, {"id": 1969...	36819
4430	And Then Came Love	2007-01-01	5.0	Successful New York journalist and single mom ...	[]	29731
546	Minions	2015-06-17	6.4	Minions Stuart, Kevin and Bob are recruited by...	[{"id": 3487, "name": "assistant"}, {"id": 179...	211672
1616	What's the Worst That Could Happen?	2001-05-31	5.1	Thief Kevin Caffery attempts to rob from the h...	[{"id": 4480, "name": "business man"}, {"id": ...	14034
2420	Lottery Ticket	2010-08-20	5.6	Kevin Carson is a young man living in the proj...	[]	41382
4471	Kevin Hart: Laugh at My Pain	2011-09-09	7.7	Experience the show that quickly became a nati...	[{"id": 9716, "name": "stand-up comedy"}]	74510

Gambar 3.5. *Machine Learning Model* sistem rekomendasi film

Gambar 3.5 merupakan hasil machine learning model dari sistem rekomendai film. Pada bagian ini dengan 'Home Alone' Sebagai contoh input sistem dapat menghasilkan output dengan bentuk 10 judul film yang mirip dengan input.

```
snap mode 0                                                    VM12:790
                                                               VM12:438
▶ {_text: 'Recommend me a movie.', entities: {…}, WARNING: 'DEPRECATED', msg_id: '04ocg3
  ZEwTmOFeByK'}
                                                               VM12:494
▶ {answer: true, text: 'Here are some reccomendation for you The Dark Knight, and ,Batma
  n Forever, and ,Batman Returns', re_id: 0}
                                                               VM12:606
▶ {intent: 'get_movie', entity: {…}, question: 'Recommend me a movie.', explore: false,
  exploremore: false, …}
```

Gambar 3.6. Sistem rekomendasi film yang sudah diintegrasi

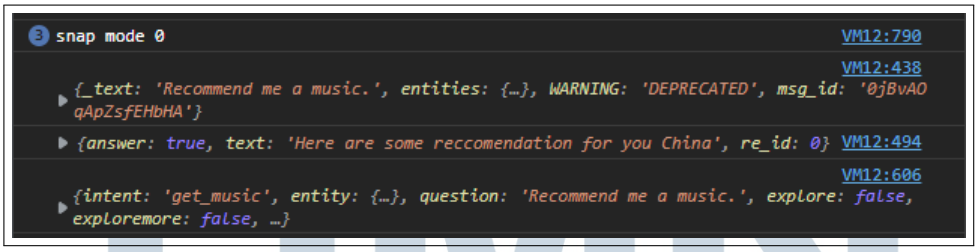
Gambar 3.6 merupakan hasil percobaan setelah diintegrasi,karena *input* dan *output* berupa suara maka dari itu contoh diberikan dalam bentuk potongan gambar dari *Browser Developer Tools*. Dengan kalimat '*Recommend me a movie*' sebagai pemicu , sistem mengambil '*The Batman*' sebagai contoh *input* dari *database* dan menghasilkan ouput '*Here are some Recommendation for you The Dark Knight, an, Batman Return, and batman Forever*'.

```
In [10]: def get_recommendations(TrackName, cosine_sim=cosine_sim):
             idx = indices[TrackName]
             sim_scores = list(enumerate(cosine_sim[idx]))
             sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
             sim_scores = sim_scores[1:11]
             music_indices = [i[0] for i in sim_scores]
             tname = df['Track.Name'].iloc[music_indices]
             aname = df['Artist.Name'].iloc[music_indices]
             genre = df['Genre'].iloc[music_indices]

             return_df = pd.DataFrame(columns=['Track.Name','Artist.Name'])
             return_df['Track.Name'] = tname
             return_df['Artist.Name'] = aname
             return_df['Genre'] = genre
             return return_df

In [11]: get_recommendations('Never Really Over')

Out[11]:
```

	Track.Name	Artist.Name	Genre
22	No Me Conoce - Remix	Jhay Cortez	reggaeton flow
1	China	Anuel AA	reggaeton flow
23	Soltera - Remix	Lunay	latin
10	Callaita	Bad Bunny	reggaeton
16	LA CANCIÓN	J Balvin	latin
36	Otro Trago	Sech	panamanian pop
15	No Guidance (feat. Drake)	Chris Brown	dance pop
46	Te Robaré	Nicky Jam	latin
34	Never Really Over	Katy Perry	dance pop
29	QUE PRETENDES	J Balvin	latin

Gambar 3.7. *Machine Learning Model* sistem rekomendasi musik

Gambar 3.7 merupakan hasil *Machine Learning Model* dari sistem rekomendasi musik. Pada bagian ini dengan 'Never Really Over' Sebagai contoh *input* sistem dapat menghasilkan *output* dengan bentuk 10 judul musik yang mirip dengan *input*.



Gambar 3.8. Sistem rekomendasi musik yang sudah diintegrasi

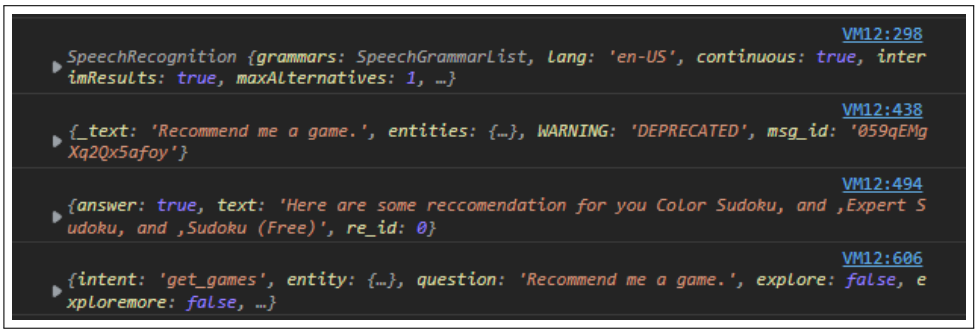
Gambar 3.8 merupakan hasil percobaan setelah diintegrasi,karena *input* dan *output* berupa suara maka dari itu contoh diberikan dalam bentuk potongan gambar dari *Browser Developer Tools*. dengan kalimat '*Recommend me a music*' sebagai pemicu , sistem mengambil '*Sucker*' sebagai contoh *input* dari *database* dan menghasilkan *output* '*Here are some Recommendation for you China*'.

```
In [11]: def get_recommendations(Name, cosine_sim=cosine_sim):
             idx = indices[Name]
             sim_scores = list(enumerate(cosine_sim[idx]))
             sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
             sim_scores = sim_scores[1:11]
             game_indices = [i[0] for i in sim_scores]
             name = df['Name'].iloc[game_indices]
             dev = df['Developer'].iloc[game_indices]
             price = df['Price'].iloc[game_indices]
             age = df['Age Rating'].iloc[game_indices]
             size = df['Size'].iloc[game_indices]
             genres = df['Genres'].iloc[game_indices]
             gid = df['ID'].iloc[game_indices]

             return_df = pd.DataFrame(columns=['Name','Developer'])
             return_df['Name'] = name
             return_df['Developer'] = dev
             return_df['Price'] = price
             return_df['Age Rating'] = age
             return_df['Size'] = size
             return_df['Genres'] = genres
             return_df['ID'] = gid
             return return_df

In [12]: get_recommendations('Reversi')

Out[12]:
```

	Name	Developer	Price	Age Rating	Size	Genres	ID
2823	Fresh Reversi	Alexander Deplov	1.99	4+	18739200.0	Games, Entertainment, Strategy, Board	732986215
16354	CraniumCrush: Reversi	Nitch Ventures LLC	0.00	4+	29253632.0	Games, Strategy, Board	1457753806
7803	Nip ~Circular Reversi~	Yuki Takeda	0.00	4+	25025536.0	Games, Board, Strategy, Entertainment	1080281831
9021	REVERSI VS	Takemi Fukuda	0.00	4+	114961408.0	Games, Strategy, Entertainment, Board	1129848966
2	Morocco	Bayou Games	0.00	4+	674816.0	Games, Board, Strategy	284946595
7566	Reversi \u03b1	Yuki Takeda	0.00	4+	26455040.0	Games, Strategy, Board, Entertainment	1070497195
13148	Othello Classic Plus	Mai Ha Ngoc Nhu	0.00	4+	22300672.0	Games, Trivia, Entertainment, Strategy	1315324978
3797	Green Othello	Emre Soyyigit	0.00	4+	37679104.0	Games, Education, Strategy, Board	873914888
14896	Chess Online\xb7	Solitaire Games Free	0.00	4+	147505152.0	Games, Board, Strategy	1424486657
3110	Dr. Reversi	SUD Inc.	0.00	4+	18947072.0	Games, Strategy, Board, Entertainment	787898884

Gambar 3.9. *Machine Learning Model* sistem rekomendasi permainan

Gambar 3.9 merupakan hasil *Machine Learning Model* dari sistem rekomendasi permainan. Pada bagian ini dengan 'Reversi' Sebagai contoh *input* sistem dapat menghasilkan *output* dengan bentuk 10 judul permainan yang mirip dengan *input*.



Gambar 3.10. Sistem rekomendasi permainan yang sudah diintegrasi

Gambar 3.10 merupakan hasil percobaan setelah diintegrasi, karena *input* dan *output* berupa suara maka dari itu contoh diberikan dalam bentuk potongan gambar dari *Browser Developer Tools*. dengan kalimat '*Recommend me a game*' sebagai pemicu , sistem mengambil '*Sudoku*' sebagai contoh *input* dari database dan menghasilkan *output* '*Here are some Recommendation for you Color sudoku and Expert sudoku and Sudoku free*'.

3.4 Kendala dan Solusi yang Ditemukan

Dalam pelaksanaan praktik kerja magang, terdapat beberapa kendala yang ditemukan, diantaranya:

1. Jumlah *Web Service*

   Pada awal perancangan dan integrasi, 3 *Machine Learning Model* tersebut (Film, Musik, dan Permainan) hendak dibangun dalam 1 *Web Service* tapi ternyata proses pembentukan matrix *Cosine Similarity* seperti pada gambar 3.2, terlalu banyak memakan *resource*, proses tersebut membuat sistem menajdi terlalu berat untuk dijalankan.

2. *Web Browser* yang digunakan terlalu berat

   Dengan banyaknya *Web Service* yang berjalan bersamaan, maka *resource* yang dibutuhkan cukup banyak. Penggunaan *Web Browser* seperti *Google Chrome* akan mengambil *resource* lebih banyak lagi.

Dari beberapa kendala selama proses pelaksanaan praktik kerja magang dilakukan langkah solusi yang dapat menjadi jalan keluar dari permasalahan yang dihadapi. Penjelasan terhadap solusi atas kendala yang ditemukan adalah sebagai berikut.

1. Membangun 3 *Web Service* yang berbeda

   Solusi dari permasalahan ini adalah dengan membangun 3 *Web Service* yang berbeda untuk tiap-tiap *Machine Learning Model* agar proses pembentukan matrix *Cosine Similarity* terbagi menjadi 3 bagian.

2. Mengganti *Web Browser* yang digunakan

   Solusi dari permasalahan ini adalah dengan menjalankan sistem di *Web Browser* yang lebih ringan seperti *Microsoft Edge*.

UNIVERSITAS MULTIMEDIA NUSANTARA