

project report

COMPARING THE COLLABORATIVE FILTERING ALGORITHM WITH naïve BAYES ON THE FILM recommendation SYSTEM

liem, daniel aditya santoso

18.k1.0036

Faculty of Computer Science

Soegijapranata Catholic University

2020

APPROVAL AND RATIFICATION PAGE (Heading plain)

(gunakan style “Approval”)

COMPARING THE COLLABORATIVE FILTERING ALGORITHM WITH NAÏVE BAYES ON THE FILM RECOMMENDATION SYSTEM

by

Liem, Daniel Aditya Santoso – 18.K1.0036

This project report has been approved and ratified

By the Faculty of Computer Science on January, DD. YYYY

With approval,

|  |  |
| --- | --- |
|  | Supervisor  Dosen Pembimbing Project  NPP: |
| Examiners,  1.)  [NAMA PENGUJI 1]  NPP :  2.)  [NAMA PENGUJI 2]  NPP :  3.)  [NAMA PENGUJI 3]  NPP : |  |
|  | Dean of Faculty of Computer Science  Robertus Setiawan AJi Nugroho, Ph.D  NPP: 058.1.2004.264 |

DECLARATION OF AUTHORSHIP

I, the undersigned:

Name : LIEM, DANIEL ADITYA SANTOSO

ID : 18.K1.0036

declare that this work, titled "COMPARING THE COLLABORATIVE FILTERING ALGORITHM WITH NAÏVE BAYES ON THE FILM RECOMMENDATION SYSTEM", and the work presented in it is my own. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at Soegijapranata Catholic University
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
3. Where I have consulted the published work of others, this is always clearly attributed.
4. Where I have quoted from the work of others, the source is always given.
5. Except for such quotations, this work is entirely my own work.
6. I have acknowledged all main sources of help.
7. Where the work is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Semarang, MONTH, DD, YYYY

[sign here]

LIEM, DANIEL ADITYA SANTOSO

18.K1.036

ACKNOWLEDGMENT

Silahkan tuliskan anda ingin mengucapkan terima kasih atau ucapan persembahan ke siapapun yang anda rasa perlu ditulis disini

I have received a myriad of support, advice, and assistance throughout this document writing. I would like to thank my supervisors XXXXXX for formulating this topic. I would also like to thank my friend XXXXX for guiding with advice to finish this document.

I would like to thank my family and friends for giving me ceaseless love, support, and advices throughout my study in Soegijapranata Catholic University. You gave me great escape to rest my mind from my thesis.

ABSTRACT (Abstract Title)

Isi dari abstract menggunakan syle abstract content. Abstract ditulis dalam 3 paragraf. Semua ditulis dengan huruf italic dan 1 spasi Paragraf pertama berisi tentang permasalahan yang diselesaikan dalam project ini.

Paragraf kedua di sini, membahas tentang proses penyelesaian yang Anda tawarkan.

Sedangkan paragraf ketiga membahas tentang hasil akhir. Setelah itu di bagian paling bawah, sertakan keywords atau kata kunci 3-5 kata.

Keyword: kata\_kunci1, kata\_kunci2, kata\_kunci3, dst

\*Tambahkan informasi mengenai penelitian payung di sini apabila ada (konsultasikan dengan dosen pembimbing).

TABLE OF CONTENTS

COVER i

[APPROVAL AND RATIFICATION PAGE (Heading plain) ii](#_Toc77239046)

[DECLARATION OF AUTHORSHIP iii](#_Toc77239047)

[ACKNOWLEDGMENT iv](#_Toc77239048)

[ABSTRACT (Abstract Title) v](#_Toc77239049)

[TABLE OF CONTENTS vi](#_Toc77239050)

[LIST OF FIGURE viii](#_Toc77239051)

[LIST OF TABLE ix](#_Toc77239052)

[CHAPTER 1 PETUNJUK PENULISAN (Style: Heading 1) 1](#_Toc77239053)

[1.1. Petunjuk Penggunaan Template (Style: Heading 2) 1](#_Toc77239054)

[1.1.1. Bagian sub-subchapter 1](#_Toc77239055)

[1.1.2. Penulisan bilangan dan satuan 1](#_Toc77239056)

[1.1.3. Penomoran halaman 2](#_Toc77239057)

[1.1.4. Penomoran judul gambar dan judul tabel 2](#_Toc77239058)

[1.1.5. Penomoran formula dan fungsi matematika 2](#_Toc77239059)

[1.2. Isi Laporan Akhir 2](#_Toc77239060)

[CHAPTER 2 INTRODUCTION 3](#_Toc77239061)

[2.1. Background 3](#_Toc77239062)

[2.2. Problem Formulation 3](#_Toc77239063)

[2.3. Scope 3](#_Toc77239064)

[2.4. Objective 3](#_Toc77239065)

[CHAPTER 3 LITERATURE STUDY 4](#_Toc77239066)

[CHAPTER 4 RESEARCH METHODOLOGY 6](#_Toc77239067)

[CHAPTER 5 ANALYSIS AND DESIGN 7](#_Toc77239068)

[5.1. Analysis 7](#_Toc77239069)

[5.2. Penambahan Gambar, Judul Gambar, dan Penggunaan Gambar 7](#_Toc77239070)

[5.3. Tabel, Posisi, dan Isi tabel 9](#_Toc77239071)

[5.4. Desain 9](#_Toc77239072)

[5.5. Function 10](#_Toc77239073)

[CHAPTER 6 IMPLEMENTATION AND TESTING 12](#_Toc77239074)

[6.1. Implementation 12](#_Toc77239075)

[6.2. Testing 12](#_Toc77239076)

[CHAPTER 7 CONCLUSION 13](#_Toc77239077)

[REFERENCES 14](#_Toc77239078)

[APPENDIX a](#_Toc77239079)

LIST OF FIGURE

[**Figure 5.1** Diagram Analisis Masalah (Figure Caption) 7](#_Toc77239080)

[**Figure 5.2** (a) Model DCGAN [9] dan (b) Model VAE [10] 8](#_Toc77239081)

[**Figure 5.1** Contoh equation 10](#_Toc77239082)

[**Figure 5.2** Penambahan angka untuk equation 11](#_Toc77239083)

[**Figure 5.3** Equation setelah ditambahkan angka 11](#_Toc77239084)

LIST OF TABLE

[**Table 5.1.** Tabel Analisis Data (Table caption) 9](#_Toc77239085)

# INTRODUCTION

## Background

Movies have become our daily needs in terms of entertainment. It's no secret that several well-known platforms such as Netflix, Youtube, Disney+ Hotstar, and others have more or fewer millions and even tens of millions of films on each of these platforms. Therefore, the recommendation engine is at the heart of various movie provider platforms.

The number of these films is what causes a new problem, namely what film to watch next. Almost all platforms must collect data on movies watched by their users, the problem is if a platform still doesn't have data from that user or a new user. Then what are the movie recommendations that should be presented to the new user if a platform still doesn't have viewing data from that user or still has very little data from that user. Of course, users from each platform want to get movie recommendations that they think are most suitable for them, if not, then it's likely that these users will not use the platform anymore.

However, this problem has been solved with the recommendation engine. There are many algorithms used to create recommendation engines. Each algorithm has its advantages and disadvantages. Some algorithms recommend a film based on data between users, but there is also an algorithm that recommends also based on the relationship from one film to another. Then there is also a combined algorithm of the two, namely user-item, the user-item relation produces more maximum recommendations than relying on only one of the components above.

But of course, there are always shortcomings from the algorithms listed above, such as users who are new and don't have any viewing records. There is also an accuracy that is not optimal due to the low rating of each film. And in this study, the algorithm that I will use is item-based which uses data from ratings per film.

In this study, I tried to make 2 recommendation engines using 2 different algorithms, and then I will measure them concerning MSE and RMSE. And I will try this research with the Euclidean distance approach to find the proximity between 1 object to another. From this research, it is expected to see which of the two algorithms has the least error.

## Problem Formulation

1. Which of the Collaborative Filtering and Nave Bayes has the lowest MSE and RMSE values ?
2. What factors can be used as a reference to compare the two algorithms ?

## Scope

The data used is the data that I downloaded from movielends.org. All the data that I use already have all the data records so there are no users who don't have data records at all.

## Objective

Comparing the algorithm between Collaborative Filtering and Naïve Bayes then calculating the RMSE value, and predicting the films that will be recommended to users while maintaining the genre.

Comparing two different algorithms against the film recommendation engine and finding out which of the two algorithms is the most effective, based on the smallest MSE and RMSE values. In collaborative filtering, we use the approach through KNN.

# LITERATURE STUDY

Khusna et al. [1] ⁠ perform an analysis on the product recommendation system on the Gadget Shield by using the collaborative filtering method based on user-to-user references. The data was obtained by the author by conducting several direct interviews with 17 respondents and managed to get 40 mobile phone product data and also 15 user data. Where the data in the training then make predictions to each user to test it. Then from the average data that has been obtained, it was tested using the RMSE test method and obtained an accuracy result of 0.496 or 90.08%. So that this research can be said to be quite good, but there are still some shortcomings that can be improved, such as the use of an algorithm that is more accurate than the Euclidean distance.

Wahyudi, Indah Survyana [2]⁠ trying to solve problems in the film industry, namely the film recommendation engine that has different search results or recommendation results based on the data or queries used. In this study, the author tries to overcome a common problem in the use of collaborative filtering algorithms for film recommendations, namely sparsity by using the best model. And get a validation result of 0.96 and a test result of 0.94 on a data amounting to 100 thousand, and a validation result of 0.86 and a test result of 0.96 on a data of 1 million. filtering by 62% and with the genre similarity method with 75% results.

N.Safir, S.Ziad [3] This research is intended to find out how the performance of an algorithm with data continues to grow. The algorithm used in this research is item-based collaborative filtering and also FunkSVD, Collaborative filtering itself here is still divided into 3 more, namely cosine-based similarity, correlation-based similarity, and also computing prediction. Different neighborhoods are then tested using MSE and RMSE. The results in testing datasets of 100 thousand and 1 million there is no significant difference, but when tested with continuously growing data FunkSVD gets a larger percentage compared to item-based collaborative filtering. From this research, we can see that in data that continues to grow, FunkSVD can show better performance than collaborative filtering.

N.Raval dan V.Khedar [4]⁠ In this study, researchers tested various recommendation algorithms with the aim of getting the algorithm that had the best performance. Various kinds of algorithms were tested such as collaborative filtering, ALS (Alternating Least Squares), rapid miner, and there is also a method of grouping data with a neural network. In this test, the researcher writes that the results of various tests still have their respective strengths and weaknesses. To be able to develop a better recommendation algorithm, the researcher gives suggestions to use the libraries that already exist in python to look for hidden factors that can be the key to the recommendation.

K.Goldberg et al. [5]⁠ researching to test a new algorithm, namely, eigenstate which is accurate and more efficient to users in 1 time online. Of course, the algorithm used is the Eigentaste algorithm, Eigentaste itself is collaborative filtering that uses universal queries to get data ratings based on rating data. Then this test is tested using the normalized mean absolute error with several other factors such as user data that is offline and also online by 20%. Then also in testing the Uniform and noise distribution model, the NMAE result is better, which is 33%.

A.Varma [6] In this study the researcher tries to understand various recommendation algorithms and also to compare their performance with the dataset from MovieLens. The algorithm used is collaborative filtering and also content-based recommendation. After passing several tests using a number of different datasets, several problems were obtained, such as the larger the dataset, the larger the RMSE test results, the smaller the level of accuracy. Therefore, the researchers hope to be able to add content-based collaborative filtering to add several criteria to be able to categorize several films, it can also be based on recommendations based on actors, directors, and writers.

L.Shuxian, F .Sen [7] In this study, the researcher tried to personalize the users to find similarities between one user to another. By using the nave Bayes method on the MovieLens dataset, researchers got better performance than item-based collaborative filtering or user-based collaborative filtering. For nave Bayes itself, the precision value is 0.642, and for collaborative filtering item-based and user-based, it is 0.1788 and 0.1984. Therefore, the use of nave Bayes in the recommendation engine can have a significant impact.

Sahu, Satya Prakash et al. [8] researching comparing several algorithms aimed at finding the algorithm that has the best performance by conducting a comprehensive comparative analysis. And the results of the analysis are measured by one factor, namely the cold start problem where a user's data does not have any watch records or references. The methods used are content-based filtering, collaborative-based filtering, hybrid content collaborative-based filtering, k-mean clustering, and also nave Bayes. This test is tested with several dataset sizes such as 10 thousand, 50 thousand, and also 100 thousand datasets. In all these tests, the one who got the highest score was nave Bayes while the one who got the lowest score was content-based collaborative filtering. So that in handling the cold start problem with several sizes of the nave Bayes dataset, it is the one who has the highest predictive value among the other algorithms tested in this study.

Anchal Dubey dan Raju Ranjan [9] conduct research in solving problems such as cold start problems, sparsity, and also scalability which is usually found in content-based collaborative filtering and hybrid collaborative filtering. In solving this problem, the researcher tries to combine two algorithms, namely nave Bayes with collaborative filtering by predicting the rating and preference of each user. With several technical stacks such as XML, JAVA, Python, Android Studio, and also firebase. With all APIs built using nodeJs and also with database i.e. MongoDB. Android studio here plays a role in social media to bridge some users in sending some of the films they like and see the reaction of the user who receives the message whether he gives a negative or positive response. Can also be developed with a group between users to share. By conducting this research, the researcher got several reviews from several users for a film. With this review, it is hoped that in the future it can help other researchers in developing their recommendation engine.

Poonam Sharma dan Lokesh Yadav [10] conducted research using content-based collaborative filtering and also collaborative filtering in a film recommendation engine with the aim of accelerating the accuracy of recommendations based on ratings from users in each film. The results of this study are authentic data collection to be able to provide recommendations that are more accurate based on several factors such as likes, ratings, comments, password modification. With methods such as clustering, similarity, and classification that can reduce the MAE value which will increase the accuracy and precision of the recommendation engine. In the future, researchers can hope to use a hybrid recommender using clustering and similarity to get better performance.

The difference between my project and the projects from the journal above is the use of methods and also the use of analytical research. I will try to compare item-based collaborative filtering algorithms with nave Bayes in a movie recommendation engine. The dataset that I took myself was taken from MovieLen, which amounted to approximately 10,000 user history data who have rated a film. Then to compare the performance between the two algorithms I will calculate the MSE and also the RMSE which I will calculate with the two algorithms. In the journals above, they usually try to combine collaborative filtering with nave Bayes in the recommendation engine in order to get maximum results. In some journals there are also those who try to compare collaborative filtering with nave Bayes by using genre as its classification.

# RESEARCH METHODOLOGY

## Data Collection

The dataset used in this project is a downloadable dataset from MovieLens.org itself but I got this dataset from Kaggle. This dataset itself has 2 sheets, namely master movies and master ratings. In the master movies, there are several variables such as movieId, title, and genre. Then the master rating itself has userId, movieId, rating, and timestamp variables. With a rating value range from 0 to 5.

From the data above we can take several factors that can support this project. We can do collaborative filtering as well as nave Bayes by using ratings from item by item, then we can also use user's watch data to find similarities between one user and another.

## Algorithms

The algorithm that will be used in this research is item-based collaborative filtering and also naive Bayes. Both are used because both algorithms are quite widely used in making recommendation engines. Then the research at this time will be the main factor is the rating. Here we can predict a film that we will recommend to users by calculating the similarity of one item to another based on its rating using the naive Bayes method, and can also recommend films using the item-based collaborative filtering method.

## Design

The data that will be processed is data that comes from the master rating, with the movieId, we simply need to manage from the movieId first, then when we display the recommendation results we will take data from the master movie CSV which contains information on the title of the film and also the genre. The amount of the data will be calculated by comparing the amount of all data, then from the accumulated data, we will calculate the percentage as a benchmark for similarity. For collaborative filtering, we will use an algorithmic approach with the help of KNN to find the closest distance. Then for the nave bayes itself, we will try to calculate the distance based on the percentage of each film.

## Coding

Here the programming language that I will use is python 3.8 because python can process large amounts of data. In python, some libraries are quite adequate in the process of working on this research, but in the algorithm that I will be working on I will not use any libraries. The library that I will use is like pandas to read CSV files, then there is NumPy to perform the calculation process and also to change some forms of matrices, then I will use matplotlib to display some graphs as a form of visualization, and other libraries.

## Analysis

This project aims to measure the accuracy of the two algorithms, namely item-based collaborative filtering with nave Bayes in making a film recommendation engine. With the dataset obtained from MovieLens, there are 10,000 data ratings from users and about 149,000 movie title data. We will test the level of accuracy and efficiency. The results of this analysis can be seen using the MSE and RMSE methods as a benchmark for the comparison of these two algorithms. The one who gets the smallest value from the test results will have the best performance.

# ANALYSIS AND DESIGN

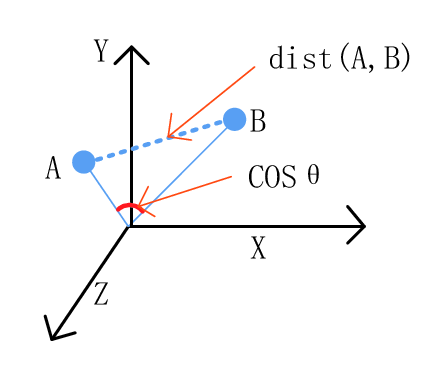
## Analysis

Pada penelitian ini saya menggunakan dua algoritma yaitu collaborative filtering item-based dengan pendekatan menggunakan knn dan juga naïve bayes dalam mesin rekomendasi film. Pada collaborative filtering sendiri saya menggunakan algoritma KNN sebagai pendekatannya dikarenakan knn memiliki kecocokan dalam menemukan jarak terdekat pada collaborative filtering item-based. Untuk mencari jarak terdekatnya juga saya akan menggunakan metode cosine similarity. Dimana memberikan rekomendasi film berdasarkan urutan jarak terdekat yang kita temukan melalui cosine similarity seperti gambar yang terlihat pada (Gambar 4.2.1). Dengan Menerapkan rumus seperti (Function 4.5.1).

Naïve Bayes adalah metode klasifikasi yang menggunakan probabilitas dan juga statistik untuk memprediksi kedepannya berdasarkan data masa lalu yang ada. Di metode naïve bayes sendiri saya akan menghitung probabilitas dari item item yang ada. Kemudian hasil rekomendasi yang akan diberikan adalah urutan teratas dari urutan persentase probabilitas terbesar. Naive bayes menggunakan rumus (Function 4.5.2).

Untuk mengukur perbandingan antara kedua algoritma tersebut saya akan mengukur dengan menggunakan metode analisis MSE dan juga RMSE. Dengan adanya validasi MSE dan juga RMSE sendiri dengan mudah melihat dimana algoritma yang memiliki nilai rendah. Dimana algoritma yang memiliki nilai terendah adalah algoritma yang memiliki tingkat akurasi lebih besar. Dengan menerapkan rumus yang ada pada (Function 4.5.3) dan juga (Function 4.5.4).

## Penambahan Gambar, Judul Gambar, dan Penggunaan Gambar



Gambar 4.2.1 Cosine similarity graphic

Cosine similarity adalah metode yang mengukur kemiripan antara teks yang dianggap sebagai vector, pada penelitian kali ini cosine similarity digunakan untuk menghitung kedekatan jarak antar satu item dengan item item lainnya. Jarak tersebut dihitung menggunakan rumus knn pada rumus 4.5.1 .

## Tabel, Posisi, dan Isi tabel

Pada penelitian kali ini kita mendapatkan data aslinya seperti yang ada pada table berikut.

Table 4.3.1 Rating table

|  |  |  |  |
| --- | --- | --- | --- |
| userId | movieId | Rating | timestamp |
| 1 | 16 | 4 | 1.22E+09 |
| 1 | 24 | 1.5 | 1.22E+09 |
| 2 | 26 | 4 | 1.22E+09 |
| 3 | 34 | 2 | 1.22E+09 |
| 3 | 24 | 2.7 | 1.22E+09 |

Table 4.3.2 Movie table

|  |  |  |
| --- | --- | --- |
| movieId | title | Genres |
| 1 | Toy Story (1995) | Adventure|Animation|Children|Comedy|Fantasy |
| 2 | Jumanji (1995) | Adventure|Children|Fantasy |
| 3 | Grumpier Old Men (1995) | Comedy|Romance |
| 4 | Waiting to Exhale (1995) | Comedy|Drama|Romance |

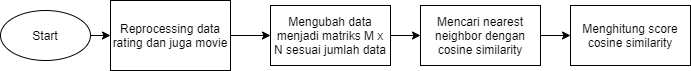
Yang akan digunakan pada penelitian ini pada table rating adalah kolom userId dan movieId lalu pada table movie kita hanya akan menggunakan movieId dan juga title. Kemudian kita akan melakukan prepocessing data pada table 4.3.3 yaitu adalah gabungan dari kedua table diatas dimana data dari table disana adalah data yang akan kita kelola dan hitung pada penelitian ini.

Table 4.3.3 Preprocessing data table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | … |
| Toy Story (1995) | 1 | 1 |  | 1 |  |
| Jumanji (1995) |  | 1 |  | 1 |  |
| Grumpier Old Men (1995) | 1 |  | 1 |  |  |
| Waiting to Exhale (1995) | 1 |  | 1 | 1 |  |
| … |  |  |  |  |  |

Pada prepocessing data table row teratas adalah daftar userId , lalu di bagian kolom terkiri adalah list dari movie movie nya. Lalu di tengah tengah nya kita berikan user 1 menonton film apa saja dengan memberikan angka satu , jika tidak berangka maka tandanya user tersebut belom menonton film tersebut.

## Desain



Flowchart 4.4.1 Collaborative filtering Flowchart

Pada collaborative filtering kita akan memulai nya dengan mengubah data dari table 4.3.1 dan 4.3.2 menjadi seperti table 4.3.3 . Lalu kita akan menghitung nearest neighbor dengan menggunakan rumus cosine similarity seperti yang tertera di rumus 4.5.1 lalu dengan demikian kita akan mendapatkan score jarak terdekat lalu kita akan merekomendasikannya.



Flowchart 4.4.2 Naïve Bayes Flowchart

Lalu untuk naïve bayes sendiri kita akan menggunakan proses yang sama seperti di collaborative filtering yaitu kita akan membuat data seperti pada table 4.3.3 . Lalu kita akan menghitung jumlah probabilitas keluarnya sebuah film atau jumlah user yang menonton dari masing masing kolom dan baris. Setelah kita mendapatkan perhitungan tersebut kita akan menggunakan rumus naïve bayes seperti di function 4.5.2 untuk menghitung berdasarkan input atau film yang akan kit acari. Lalu setelah kita mendaapatkan score nya maka kita akan merekomendasikan sesuai rank teratas nya.

## Function

Dalam collaborative filtering kita akan menggunakan pendekatan pada kkn dimana kita akan melakukan penghitungan seperti.

Function 4.5.1 Collaborative Filtering Function

Dimana x1 adalah sampel data nya atau data item lainnya, lalu x2 adalah data yang diuji atau film yang sedang dicari, i adalah variabel data , d adalah jarak antar data , lalu p adalah dimensi dari data tersebut.

Untuk Naïve bayes sendiri kita akan menggunakan rumus probabilitas yaitu

Function 4.5.2 Naïve Bayes Function

Dimana p(h) adalah probabilitas dari hipotesis h, lalu p(d) adalah probabilitas dari data yang diuji, p(h|d) adlaah probabilitas dari hipotesis h yang diuji kepada data d, dan p(d|h) adalah probabilitas dari data d yang diuji pada hipotesis h yang benar.

# IMPLEMENTATION AND RESULTS

## Implementation

Bab implementasi adalah bab tentang narasi pemanfaatan dari data structure dan algoritma dalam bentuk aplikasi terapan.

Bab implementasi menyertakan source code, namun tidak semua source code program disertakan dalam bab ini. Ambil lah penggalan nya saja yang penting dan menjadi inti dari program Anda. Gunakan style “Code” . Jika menyertakan gambar (capture), silahkan ditambahkan caption di gambar tersebut sebagaimana penjelasan pada Chapter 5.

1. create Function sfHelloWorld (vNama varchar(30))
2. returns varchar(100)
3. begin
4. declare vHello varchar(255)
5. select concat(‘Hello ‘, vNama) into vHello;
6. return vHello;
7. End

Setiap source code diberikan nomor urut baris. Jelaskan baris perintah dan untuk apa perintah tersebut. Baris 1-2 kode program berisi perintah untuk membuat function dengan nama sfHelloWorld. Baris 3 dan 7 adalah blok baris untuk function khusus di dalamnya. Inti dari kode program ada pada baris 4 untuk deklarasi variabel vHello, baris 5 untuk menggabungkan karakter “Hello” dengan variabel vNama dan mengembalikan hasilnya pada baris ke 7.

## Results

Sub bab results berisi hasil dari uji coba algoritma dan struktur data yang diterapkan dalam bentuk aplikasi. Hasil disajikan dalam bentuk tabel, narasi atau gambar yang dapat memberikan penjelasan solusi masalah dengan bantuan program sehingga dapat ditarik kesimpulan dari penelitian anda.

# CONCLUSION

Bab ini membahas tentang kesimpulan akhir. Harus menjawab semua pertanyaan yang Anda ajukan sebagai permasalahan yang bab 1 bagian scope. Tidak sekedar menyimpulkan tapi sertakan argumentasi kuat terkait pengambilan kesimpulan tersebut.

Di bagian akhir, sertakan saran untuk penelitian lanjutan. Tidak perlu bertele-tele tapi fokuskan pada saran penelitian Anda saja, apa yang belum dilakukan disertakan di sini.

REFERENCES

[1] A. N. Khusna, K. P. Delasano, D. Chaerul, and E. Saputra, “Penerapan User-Based Collaborative Filtering Algorithm Studi Kasus Sistem Rekomendasi untuk Menentukan Gadget Shield Application of User-Based Collaborative Filtering Algorithm : Case Study of Recommendation System for Determining Gadget Shield,” vol. 20, no. 2, pp. 293–304, 2021, doi: 10.30812/matrik.v20i2.1124.

[2] I. S. Wahyudi, *Mesin Rekomendasi Film Menggunakan Metode Kemiripan Genre Berbasis Collaborative Filtering*. 2017.

[3] N. Safir and S. Ziad, “Evaluating Prediction Accuracy for Collaborative Filtering Algorithms in Recommender Systems,” *KTH R. Inst. Technol.*, 2016.

[4] N. Raval and V. Khedkar, “A review paper on collaborative filtering based moive recommedation system,” *Int. J. Sci. Technol. Res.*, vol. 8, no. 12, pp. 2507–2512, 2019.

[5] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, “Eigentaste: A Constant Time Collaborative Filtering Algorithm,” *Inf. Retr. Boston.*, vol. 4, no. 2, pp. 133–151, 2001, doi: 10.1023/A:1011419012209.

[6] A. N. Varma, “Movie Recommender System,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 1, pp. 833–837, 2021, doi: 10.22214/ijraset.2021.32936.

[7] L. Shuxian and F. Sen, “Design and Implementation of Movie Recommendation System Based on Naive Bayes,” *J. Phys. Conf. Ser.*, vol. 1345, no. 4, 2019, doi: 10.1088/1742-6596/1345/4/042042.

[8] S. P. Sahu, A. Nautiyal, and M. Prasad, “Machine Learning Algorithms for Recommender System - a comparative analysis,” *Int. J. Comput. Appl. Technol. Res.*, vol. 6, no. 2, pp. 97–100, 2017, doi: 10.7753/ijcatr0602.1005.

[9] A. Dubey and R. Ranjan, “Movie Recommendation System using Naive Bayes Algorithm with Collaborative Filtering,” vol. 9, no. 7, pp. 2018–2020, 2020, doi: 10.21275/SR20603193402.

[10] P. Sharma and L. Yadav, “Movie Recommendation System Using Item Based Collaborative Filtering,” *Int. J. Innov. Res. Comput. Sci. Technol.*, vol. 8, no. 4, pp. 8–12, 2020, doi: 10.21276/ijircst.2020.8.4.2.

Tulislah semua daftar referensi, buku jurnal, dll yang Anda gunakan dalam project ini termasuk jurnal yang ada di literature study juga masuk di sini. Ditulis dengan style IEEE. Ditulis paling sedikit 10 referensi harus dipergunakan untuk memperkuat teori dan penelitian yang anda lakukan. Jika anda menggunakan software referensi seperti Mendeley atau Zotero, pembuatan referensi akan sangat mudah. Namun, jika anda memutuskan tidak menggunakan software semacam, maka pastikan semua sudah tertulis dengan rapi dan urut berdasarkan nomor in-text-citations yang anda buat di dalam laporan.

[1]

[2] A. Shibly, “Referencing and Citation,” May 2016.

[3] R. Bremananth, B. Nithya, and S. R, “Wood Species Recognition System,” *International Journal of Computer and Information Engineering*, vol. 3, no. 4, 2009.

[4] M. Francisco-Fernández, J. Tarrío-Saavedra, S. Naya, J. López-Beceiro, and R. Artiaga, “Classification of wood using differential thermogravimetric analysis,” *Journal of Thermal Analysis and Calorimetry*, vol. 120, no. 1, pp. 541–551, Apr. 2015, doi: 10.1007/s10973-014-4260-y.

[5] D. P. Kingma and M. Welling, “An Introduction to Variational Autoencoders,” *FNT in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019, doi: 10.1561/2200000056.

[6] U. Latifah and E. Sugiharti, “PENERAPAN ALGORITMA PRIM DAN KRUSKAL PADA JARINGAN DISTRIBUSI AIR PDAM TIRTA MOEDAL CABANG SEMARANG UTARA,” *Unnes Journal of Mathematics*, vol. 4, no. 1, Art. no. 1, May 2015, doi: 10.15294/ujm.v4i1.7418.

[7] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, “Tampering with the Delivery of Blocks and Transactions in Bitcoin,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2015, pp. 692–705, doi: 10.1145/2810103.2813655.

[8] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, “Toward Controlled Generation of Text,” *arXiv:1703.00955 [cs, stat]*, Sep. 2018, Accessed: Feb. 11, 2021. [Online]. Available: http://arxiv.org/abs/1703.00955.

[9] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *arXiv:1511.06434 [cs]*, Jan. 2016, Accessed: Feb. 16, 2021. [Online]. Available: http://arxiv.org/abs/1511.06434.

[10] J. Rocca, “Understanding Variational Autoencoders (VAEs),” *Medium*, Jul. 07, 2020. https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73 (accessed Feb. 16, 2021).

[11] I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” *arXiv:1406.2661 [cs, stat]*, Jun. 2014, Accessed: Feb. 16, 2021. [Online]. Available: http://arxiv.org/abs/1406.2661.

[12] G. L. Squires, *Practical Physics*, 4th ed. Cambridge: Cambridge University Press, 2001.

APPENDIX

Jika Anda punya lampiran dari project, silahkan dilampirkan di bagian ini. Yang wajib Anda lampirkan adalah kode program (coding) lengkap dan diberikan keterangan terlebih dahulu pada bagian atas dari coding tersebut, koding ditulis dengan format font yang berbeda. Contoh:

**CODING PEHITUNGAN**

1. SELECT @a := 5;
2. SELECT @b := 5;
3. SELECT hasil:= @a \* @b;

**PROCEDURE HITUNG PERKALIAN**

1. CREATE PROCEDURE spMaksimal ()
2. BEGIN
3. DECLARE a INT;
4. DECLARE b INT;
5. DECLARE hasil INT;
6. SELECT a \* b INTO hasil;
7. SELECT hasil
8. END

Selain coding, yang dapat dijadikan lampiran adalah: hasil hitungan yang panjang dan tidak mungkin dimuat dalam laporan utama, gambar atau ilustrasi diagram yang cukup panjang namun hanya sebagai penjelasan dari diagram utama yang ada di dalam laporan, foto-foto penunjang, dan dokumen lain yang sifatnya menunjang namun dianggap penting.