

Game Data Analytics using Descriptive and Predictive Mining

1st Narendra Yogha Prathama
Department of Information and
Computer Engineering
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia
masterthiefwtf@it.student.pens.ac.id

2nd Rengga Asmara
Department of Information and
Computer Engineering
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia
rengga@pens.ac.id

3rd Ali Ridho Barakbah
Department of Information and
Computer Engineering
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia
ridho@pens.ac.id

Abstract— The game industry is an industry that includes game development, marketing, and monetization. However, to be able to enter the game industry is not easy. Game developers must know how the market is going to be able to reap huge profits. By knowing the market situation, game developers can also determine whether the games made are in accordance with market conditions. Getting this information is not easy, especially for small game studios. In this research, we made a new application to find knowledge about games that are and will be trending. We used data mining is used to obtain this information. Data mining uses data from the Steam API to do clustering using the Hierarchical K-Means method and predictive using the Multiple Linear Regression method. The use of the Hierarchical K-Means method produces 3 clusters for the game's popularity level. The use of the Multiple Linear Regression method produces predictions of the game's popularity in the future. This new system will be able to help indie game studios to be able to obtain information about the condition of the gaming market thereby increasing the benefits that can be obtained.

Keywords— *Game Data, Steam API, Data Mining, Clustering, Hierarchical K-Means, Predictive, Multiple Linear Regression*

I. INTRODUCTION

The game industry is an industry that has quite a lot of enthusiasts, but only a few want to explore it. In Indonesia, the gaming industry is still lacking. This is because people still think that the prospect of the gaming industry is not promising. In fact, the gaming industry is not inferior to the film industry and the software industry. However, over time the requirements for the game itself are increasing. Likewise with the genre and types.

To make a game, we need information about game trends at the time because it can affect the market of the game. For large companies, this is not a problem because in general they already have statistics and data about the trend. However, this is a problem for small companies. Game companies must know what kind of games can sell in the market. Starting from the genre, mechanics, gameplay, art style, the theme of the game needs to be known so that the game made can sell in the market. Often game companies do not know about game trends in the market, so games that have been made are less interesting, which affects the sales results of the game. However, some companies don't pay attention to gaming trends. Usually, companies like that are big companies that make AAA games. EA, Projekt Red CD, Ubisoft, Valve, and Bethesda are some examples of big companies that don't follow the game trends. That's because the company already has a franchise that is very popular so that if it does not follow the gaming trend then the sales will remain high.

II. RELATED WORKS

Hendrik Baier et.al[1] presented a method for player modeling using gameplay data and neural networks. V. Bonometti et.al[2] presented a theoretical framework rooted in engagement and behavioral science research for building a model able to estimate engagement-related behaviors employing only a minimal set of game-agnostic metrics. James A. Brown et.al[3] presented a machine learning system with unsupervised learning and supervised learning components to analyze big chess data. Chih-Wei et.al.[4] presented an analysis of game data using clustering and visualized using virtual reality technology. Lucas V. Fernandes et.al[5] presented a survey of articles published over the last fourteen years that apply Game Analytics techniques on Massive Multiplayer Online Games to outline the recent research conducted in this area: the type of MMOG that is the object of research; the most common game metrics adopted; the game telemetry used in the research. Lidson B et.al[6] presented a new approach for 2D game design analysis using image processing. Hwanhee Kim et.al[7] presented a new system to enable a game designer to procedurally create key content elements in the game level through simple association rule input. Jeppe Theiss Kristensen et.al[8] presented research about how to improve the current state-of-the-art in churn prediction by combining sequential and aggregate data using different neural network architectures. Bahar Kutun et.al[9] presented a new board game for learning by playing with racing cars. Eunjoon Lee et.al.[10] presented an experiment about players' churn prediction and survival analysis. Sang-Kwang Lee et.al.[11] presented a model to predict churn in mobile free-to-play games. Mark Russell Lewis et.al.[12] presented a program to procedurally generate gameplay using crowd-sourced data. Quan Li et.al.[13] presented a virtual analysis of game frame rate data. Andrew R Martin et.al[14] presented the applicability of some of these ideas to game development, and then outline a proposal for a live programming model suited to the unique technical challenges of game developments. Maxim Mozgovoy et.al.[15] presented an analysis of user behavior in a mobile tennis game for making a pseudo-multiplayer, an AI player that able to match its skills with players based on players' behavior. Tridib Mukherjee et.al[16] presented research about mining player intent for targeted gaming services based on a popular card game in India. Hyunsoo Park et.al[17] presented an analysis of self-learning AI from player skills on fighting games. Perez-Colado et.al[18] presented research about game learning analytics for serious games. Keattikorn Samarngeon et.al[19] presented a review and comparison of monetization models in free-to-play gaming. Elton Sarmanho Siqueira et.al.[20] presented a predictive analysis of players who will join and leave the World of Warcraft game. Jun Tao et.al[21] presented a practical and effective algorithm for an

optimization method based on the upper confidence bound tree search algorithm. Dulakshi Vihanga et.al[22] presented research about discovering patterns of weekly player population fluctuations in online games. Vivek R. Warriar et.al[23] presented a learning technique to model players' emotional preferences in an AR mobile game. Wanshan Yang et.al[24] presented research about churn prediction in free online games based on player in-game time spending. Wanshan Yang et.al.[25] presented a method to classify players into a high-level shopper(whales), mid-level shopper(dolphins), and low-level shopper(minnows) using clustering.

III. PROPOSED IDEA

In this paper, we present a new system to analyze and predicting game trends using descriptive and predictive mining. The system present information about game trends based on player behavior. This new system can help game developers to make a plan for the game they will make with the consideration of the game trends. Figure 1 shows the system architecture of the game data analytics we proposed.

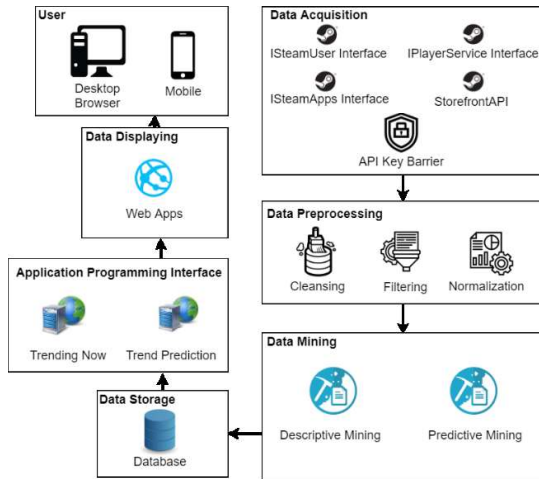


Fig. 1. System Architecture of The Game Data Analytics

Figure 1 shows the system architecture of the game data analytics. There are 7 main parts in this system; data acquisition, data preprocessing, data mining, data storage, application programming interface (API), data displaying, and users. Steam API is where we grab our data. Data mining is the process to analyze our data to get the information we want. We used web servers and web services to build our web app. And users are where the users can access our web app.

A. Data Acquisition

The first main part of this research starts with Steam API. In this section, data is retrieved from the Steam API using the get method. There are four components that are used from the Steam API. There are ISteamUser Interface, ISteamApps Interface, IPlayerService Interface, StorefrontAPI. ISteamUser Interface and IPlayerService interface is used to retrieve data about players, from player's names to player's gaming history in the last 2 weeks. ISteamApps Interface and StorefrontAPI are used to retrieve data about game details.

This research uses GetFriendList request to retrieve friends list and GetPlayerSummaries to retrieve user player data. That is because only user profile data is needed to be analyzed. Friends list is used to retrieve the id of a user who is

friends with a user to be able to retrieve his profile data. This is because there is no request to retrieve all Steam user ids.

The iPlayerService Interface is a part of the Steam API that can be used to access user information more deeply, unlike the ISteamUser Interface which only retrieves external information. Starting from GetRecentlyPlayedGames to retrieve information about games that have recently been played by users, GetOwnedGames to retrieve game list information that is owned by the user, IsPlayingSharedGame to retrieve information on whether the user is playing Shared Games, GetSteamLevel to retrieve user steam level information, to GetBadges to retrieve Badges information that is owned by the user. In this research, only the GetRecentlyPlayedGames request was used.

In ISteamApps Interface, the only request that is used is GetAppList to retrieve a list of games that are in Steam. Not only games that are in GetAppList, but there is also Downloadable Content (DLC) for a game. One game can have multiple DLCs, so most of the contents of the list are DLCs. Content filtering is needed to separate the game and DLC. However, GetAppList only contains a list of games, so it cannot be known which ones are DLCs. Content filtering can be done when retrieving game detail data on the Storefront API.

StorefrontAPI is part of the Steam API that can be used to retrieve information from the game store page on Steam. The game store page contains the name of the game, genre, short description, detailed description, about the game, publisher, developer, price, system requirements, game image, video trailer, supported language, age allowed to play the game, downloadable content(DLC) list, and so forth. Storefront API is used to retrieve game detail data based on the AppID game data that has been taken from GetAppList. In Storefront API content filtering can be done to separate the game and Downloadable Content, so that game data can be used to find game trends.

To be able to access information and data on the Steam API, an API Key is required. API Key is a collection of numbers and letters that functions as a key to open permission to access information and data in the Steam API. To get an API Key, we need a Steam account that has purchased and has an application or game with a minimum value of 5 USD. The Steam account will later be registered with the Steam API to get an API Key. If we don't have an API Key, data from the Steam API can't be accessed. Indirectly, the Steam requires someone who wants to retrieve data from the Steam API to create an account and make transactions on Steam.

B. Data Preprocessing

Before the data mining process is carried out, the data must go through a data preprocessing process. This process aims to prepare data so that it can be analyzed and produce accurate results. This process is done because the data coming from the Steam API is still dirty. Dirty is meant here is that there are still data that have invalid values and there are also data that have data that is less than it should be. There are 3 main processes in preprocessing, namely cleansing, filtering, and normalization.

In this research, the cleaning process is mainly used to delete data that cannot be used. An example is data users who haven't played the game in the last 2 weeks. Data is deleted because the data does not have the desired value, i.e. the game

played in the last 2 weeks along with the playing time. Therefore, data is deleted and not used for analysis. If used, the data will make the analysis results less accurate. After the data is cleared, the data will enter the filtering process.

The filtering process is used to find games that have more than 100 players in the last 2 weeks and group them into separate groups. This process is done because game data that has a number of players under 100 can reduce the accuracy of the analysis results. Data that has entered the filtering process will enter the normalization process.

The normalization method used in this research is the Min-Max method. The method uses the minimum and maximum values of the data as the lower and upper limits. That way, the scale of values between one feature's data with other features becomes the same. For example, in this research, Min-Max is used to equalize the scale between the data on the number of players and the average amount of playing time. By doing the normalization process, it will facilitate analysis and make the analysis results more accurate. Data that has entered the normalization process will be used for the data mining process.

C. Data Mining

In data mining, the process of finding information about game trends is carried out. There are 2 data mining processes. The process is descriptive mining and predictive mining.

Descriptive mining aims to find knowledge about game trends in a country. Knowledge of game trends is based on the number of players in the past 2 weeks, average playing time, and user rating. By knowing the number of players, playing time, and user rating, knowledge of good and popular games can be known. The method used for clustering is Hierarchical K-Means. Clustering is done on the data on the number of players, average playing time, and user rating. Data that has been clustered is then divided into 3 clusters. All three clusters are less popular, fairly popular, and popular.

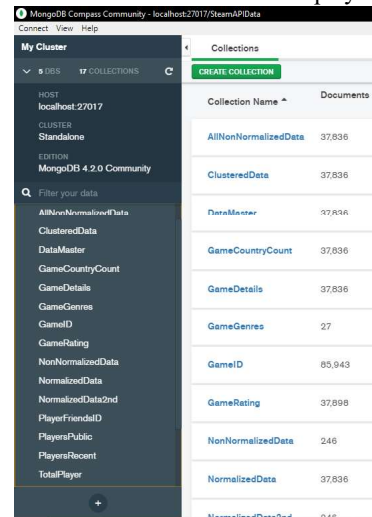
Predictive mining is done to get knowledge about the games that will be trending in the next 2 weeks. The method used to predict is the regression method. The type of regression used is multiple linear regression. This method is used because there are more than 2 features to be processed, so it cannot be done only with linear regression. To find the predicted value for each game, three regression processes are needed. The first is regression to predict the number of game players. Then the second is regression to predict the average gameplay. And the last regression is to predict the rating of the game.

D. Data Storage

In this study, a database is needed to store the data that has been obtained before and the data that has been processed. The database used in this study is MongoDB. We chose to use the MongoDB database because MongoDB is a non-relational database that supports data with the JSON format. Data taken from the Steam API has a JSON format, so it requires a database that can save in JSON format. This database is used to store data that has been taken from the Steam API and data mining results. Data from the mining process will be processed by the application programming interface (API) to be displayed on web applications.

Figure 2 shows an example of the table/collection used in this study. The AllNonNormalizedData table contains data that is not normalized. The ClusteredData table contains

clusters of each data and their labels. DataMaster contains all data that has been processed, starting from game details, the number of players per game in each country, to the normalized data. The GameCountryCount table contains the number of players per game in each country. The GameDetails table contains game details, from appid, name, genre, platform, to price. The GameGenres table contains all types of game genres. The GameID table contains all gameids for all games and DLC on Steam. GameRating contains rating data for each game. NonNormalizedData table contains data on filtering threshold that has not been normalized, starting from total player, average playtime, to rating. Normalized Data table contains all total player data, average playtime, and normalized ratings without filtering threshold first. Normalized Data2nd table contains data on the results of normalization of total players, average playtime, and ratings that have been done before the filtering threshold. The PlayerFriendsID table contains the userid player. The PlayersPublic table contains the userid of players who are public. The PlayersRecent table contains the game data played by players in the last 2 weeks along with the playing time. The TotalPlayer table contains the number of players per game.



| Collection Name | Documents |
|----------------------|-----------|
| AllNonNormalizedData | 37,836 |
| ClusteredData | 37,836 |
| DataMaster | 37,836 |
| GameCountryCount | 37,836 |
| GameDetails | 37,836 |
| GameGenres | 27 |
| GameID | 85,943 |
| GameRating | 37,898 |
| NonNormalizedData | 246 |
| NormalizedData | 37,836 |
| NormalizedData2nd | 946 |

Fig. 2. An Example of MongoDB Collections.

E. Application Programming Interface (API)

In this research, Application Programming Interface (API) is used as a bridge between a database and a web application. Data from the database will be processed and sent to the web application through the Application Programming Interface. In the Application Programming Interface, data organization is also made. So, when a Web application wants to retrieve data, it only needs to use the commands in the Application Programming Interface to get data without the need to query. There are two commands in the Application Programming Interface. The two commands are Trending Now and Trending Prediction.

The Trending Now request in the Application Programming Interface performs retrieval and organizing data about the current trending game in the database. The data is the result of descriptive mining data in the previous process. Data includes appid which is the id of a game, platform that contains the platform of a game, genre that is the genre of the game, total number of players that contains the number of players who played the game in the last 2 weeks, playtime which contains about the average amount of playing time games in the last 2 weeks, rating containing the ratings of the

games in the last 2 weeks, country count that contains the number of players who played games in a country in the last 2 weeks, clusters that contain clusters of games, labels that contain labels of clusters, and date that contains the time and date when the data was retrieved.

The Trending Prediction request in the Application Programming Interface performs retrieval and organizing data about the game predictions that will be popular in the future that has been stored in the database. The data is the result of data from predictive mining in the previous process. The data includes the appid which is the id of a game, the platform which contains the platform of the game, the genre that contains the genre of the game, the total players which contain the predicted results of the number of players who play the game for the future, playtime which contains the predicted results of the average amount of time playing the game in the future, the rating which contains the predicted results of the rating of a game in the future,

F. Web App

Web app is used as a medium to display data and knowledge stored in databases and has been organized by the Application Programming Interface. The web app will later display data from the database to users on a web based so that it can be accessed on various platforms.

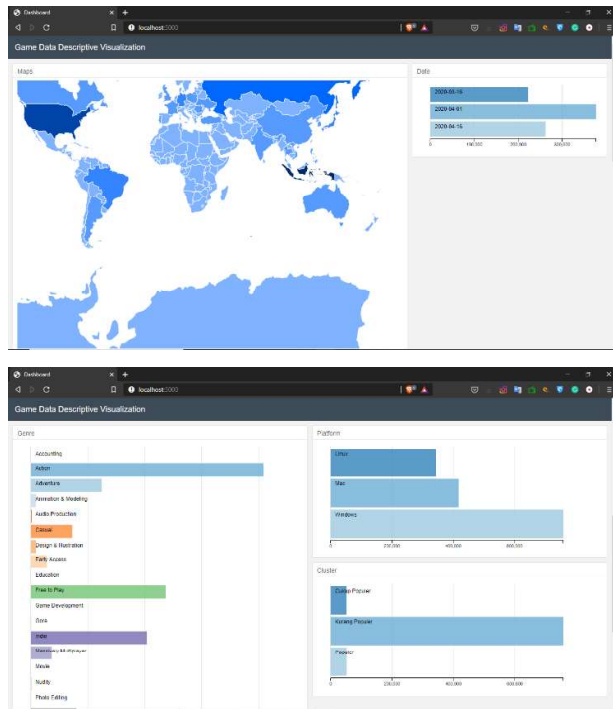


Fig. 3. Mockup of the Web App.

Figure 3 shows the initial design of the web app to be built. There are 5 sections displayed. The five sections are maps, date, platform, genre, and cluster. The maps section displays all countries in the world along with the number of gamers in each country. The date section shows when the data was retrieved. The platform section displays a game platform. The genre section displays game genres. And finally, the cluster section shows the clusters of the game. Every part of the web app is very responsive. So, when a country is selected, the data displayed in the genre, platform, and cluster section will change according to that country. Likewise, if other parts are

selected. With its responsive nature, it will be easier for users when using the web app.

G. User

Users can access our web app using a desktop browser or mobile browser. Various browsers can be used for desktop devices, ranging from the default browser device, Chrome, Opera, to the UC Browser. For mobile devices, various browsers can be used, from the default browser device, Chrome, Opera, to the UC Browser. The web app shows data from the database that have been queried on web service.

IV. EXPERIMENT & ANALYSIS

In this study, the experiment consisted of several stages. The first step is data acquisition to test how data is retrieved from the Steam API. Then the next step is to preprocess the data that has been obtained. The third step is the phase of testing the clustering of data that has been processed before. And the final step is the regression trial phase to predict future data.

A. Data Acquisition

In conducting Data Acquisition, we make a data collection program. Data acquisition starts from a request to the Steam API. Then check whether the appid used as a request parameter already exists or not in the database. If it already exists, the appid will be replaced with the next appid and checked again. However, if it is not in the database, the requested data is stored in the database. Then it will be checked whether the current number of iterations is equal to the total data. If it's not the same, the process will be repeated starting from the API request with the appid that has been replaced with the next appid. However, if the number of iterations equals the total data, then the program is complete. We use the four Steam API components to retrieve data. There is ISteamUser Interface, ISteamApps Interface, IPlayer Service Interface, StorefrontAPI. We use GetAppList from the ISteamApps Interface to retrieve data about AppId. AppID is an id for the game. Table 1 shows an example of data taken from GetAppList.

TABLE I. EXAMPLES OF DATA RETRIEVED FROM GETAPPLIST

| No | AppID |
|----|---------|
| 1 | 453480 |
| 2 | 311260 |
| 3 | 1118200 |

TABLE II. EXAMPLES OF DATA RETRIEVED FROM APPDETAILS

| N o | Steam AppID | Name | Platform | Price | Genre |
|-----|-------------|-------------|--------------|-------------|---------------|
| 1 | 453480 | Shadowverse | Windows, Mac | - | Strategy |
| 2 | 311260 | The Guild 3 | Windows | 19999 9 IDR | RPG, Strategy |

After retrieving AppID data, we retrieve game detail data using AppDetails requests from the Storefront API based on the AppID that was previously taken. There is a lot of data that can be retrieved from AppDetails. However, the data we

collect only about AppID, name, platform, price, and genre. Table II shows examples of data taken from AppDetails.

The AppID data that has been obtained is used to make AppReviews requests from the ISteamApps Service. The data taken from AppReviews requests is the number of positive reviews in the last 2 weeks and with the number of reviews in the last 2 weeks. Both of these data are used to calculate the rating of the game. The ranking formula can be seen in Equation (1).

$$rating = \frac{\text{number of positive reviews 2 weeks ago}}{\text{number of reviews 2 weeks ago}} \quad (1)$$

Example of rating data after calculation can be seen in Table III.

TABLE III. EXAMPLES OF RATING DATA

| No | Steam AppID | Rating |
|----|-------------|--------|
| 1 | 453480 | 0.4 |
| 2 | 311260 | 0.21 |

Next is the player data retrieval. The player data is retrieved using the GetPlayerSummaries request from the ISteamUser Interface. On the GetPlayerSummaries request, we retrieve player detail data which includes SteamID and location. Table IV shows examples of data players taken.

TABLE IV. EXAMPLES OF DATA PLAYER

| No | SteamID | Loc Country Code |
|----|-------------------|------------------|
| 1 | 76561198138514697 | ID |
| 2 | 76541262231789243 | US |
| 3 | 76236240987619284 | JP |

Steamid data that has been obtained is used to make GetRecentlyPlayedGames request. Data taken from a GetRecentlyPlayedGames request is game the player played along with the playing time of the past 2 weeks. Table V shows an example of GetRecentlyPlayedGames data.

TABLE V. EXAMPLES OF RECENTLY PLAYED GAMES DATA

| No | SteamID | Played GameID | Playtime 2 Weeks |
|----|-------------------|---------------|------------------|
| 1 | 76561198138514697 | 453480 | 1652 |
| 2 | 76541262231789243 | 453480 | 84234 |
| 3 | 76236240987619284 | 453480 | 325 |

B. Data Preprocessing

In doing data preprocessing, we make a program for data normalization. The normalization method that we use is Min-Max, so we need to look for the minimum and maximum values of each value, starting from total player, average

playtime, to rating. Then the Min-Max method is used to find the normalized value and store the results in a database. Table VI shows the normalized data.

TABLE VI. NORMALIZED DATA

| No | Steam AppID | Total Players | Average Playtime | Rating |
|----|-------------|---------------|------------------|--------|
| 1 | 453480 | 0.2 | 0.5 | 0.4 |
| 2 | 311260 | 0.01 | 0.31 | 0.21 |
| 3 | 1118200 | 0.12 | 0.48 | 0 |

C. Data Clustering

In conducting clustering, we create a clustering program using the Hierarchical K-Means method. Firstly, we need to do clustering using the K-Means method 10 times. Then, use the centroid of each cluster in each iteration to do clustering using the Hierarchical method. The centroid of each cluster from the results of clustering using the Hierarchical method is used as a centroid for further clustering using the K-Means method. Figure 4 shows the clustering result.

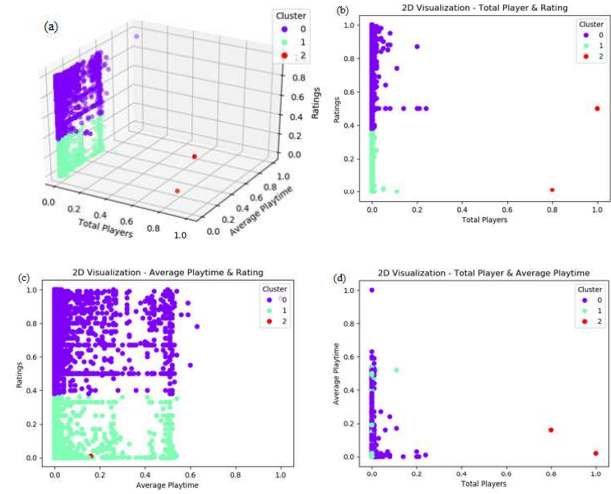


Fig. 4. 3D view (a), top view(b), front view(c), and side view of clustering result

Figure 4 can be seen several perspectives of visualization of the clustering result starting from the results of 3-dimensional visualization, top view, side view, and front view. It can be seen that most of the data has a total player value that is close to 0 when it has been normalized. This results in poor clustering results. What's more, 2 data have a very high total player value so that it becomes an outlier. The higher the total player value, the better. Next, we visualize to see the distribution of data based on total player values. This visualization can be seen in Figure 5.

Based on Figure 5, as many as 37,590 data from 37,836 data or 99.35% of data have a total player under 100, and only 246 or 0.65% of data have a total player above 100. This proves that almost all data has a small total player value, which will make the clustering results less good.

For this problem, we categorized the majority of data, which is data that has a total player value above 100 as Less Popular. Because data below the threshold has been categorized as Less Popular, clustering will be divided into 2

clusters, which are Fairly Popular and Popular. The results of clustering can be seen in Figure 6.

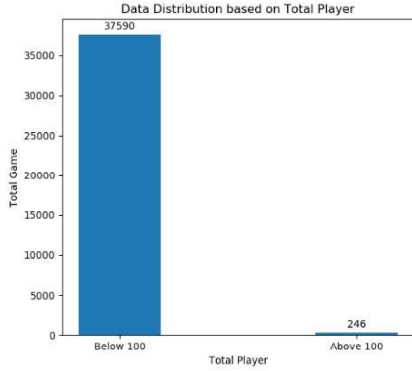


Fig. 5. Data distribution based on total player.

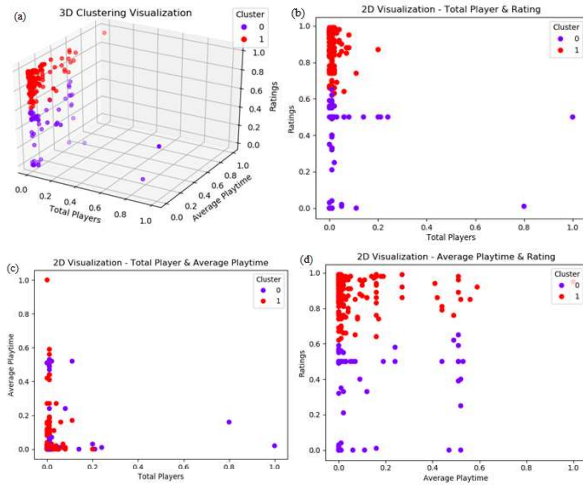


Fig. 6. 3D view (a), top view (b), front view (c), side view (d) of clustering result after threshold categorization.

Figure 6 can be seen several perspectives of visualization of the clustering result starting from the results of 3-dimensional visualization, top view, side view, and front view of the clustering result that has been categorized by a threshold. It can be seen that the results of clustering are better when compared to Figure 4. Because data that has a value of fewer than 100 players have been separated into Less Popular clusters, there are only 2 clusters left; Fairly Popular and Popular. In figure 6, red colors indicate a Popular cluster, while purple colors indicate a Fairly Popular cluster. Even though threshold categorization has been done, there are still some data that have a value of the number of players close to 0 after normalization. If the threshold is raised from 100 to 1000, fewer data can be analyzed by clustering. This causes clustering to be less accurate. So, even though the final clustering results are better, further trials need to be done to be able to obtain better results.

D. Data Prediction

In doing prediction, we create a prediction program using the Multiple Linear Regression method. The first thing to do is to group each game based on the time of data collection. Then, the prediction is done three times. The feature that we need to predict is the number of players, the average playing time, and the rating. After that, normalization for prediction data is carried out so that the data can produce good results during clustering. For more details, can be seen in Figure 7.

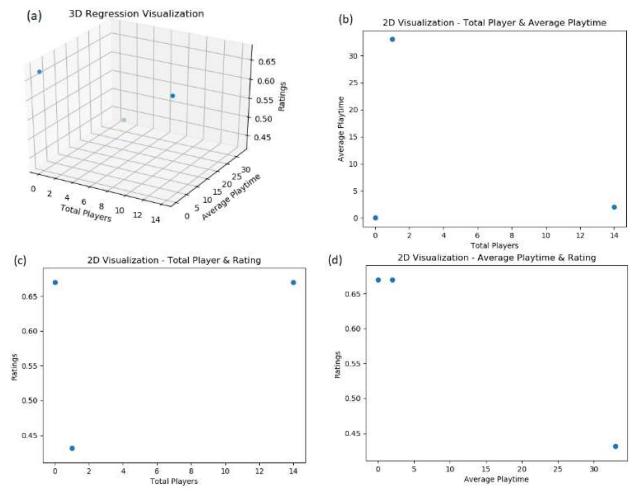


Fig. 7. 3D view (a), top view (b), front view (c), side view (d) of an example of regression result.

Figure 7 is an example of a graph of game data that has been predicted using regression. It can be seen in Figure 7 that the regression results are not good. This is due to the lack of data for each game, resulting in the results of the regression becomes less accurate. To overcome this, more data is needed.

The regression results are then normalized and used for clustering. After the normalization process is completed, the clustering process is carried out using the K-Means method ten times. Then, use the centroid of each cluster in each iteration to do clustering using the Hierarchical method. The centroid of each cluster from the results of clustering using the Hierarchical method is used as a centroid for clustering using the next K-Means method. For more details, can be seen in Figure 8.

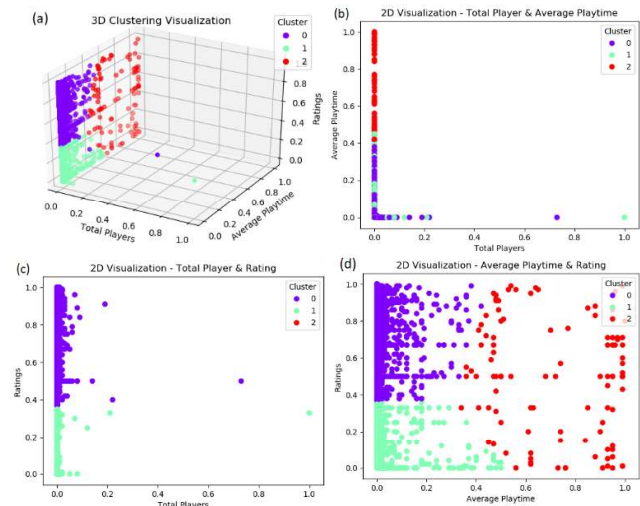


Fig. 8. 3D view (a), top view (b), and front view (c) of clustering result using regression data.

Figure 8 can be seen several perspectives of visualization of the clustering result starting from the results of 3-dimensional visualization, top view, and front view. It can be seen that most of the data from clustering results from regression data have a value of the number of players close to 0. This is the same problem as the problem in the previous clustering, so to overcome it can be done the same thing to make clustering results better. By making the number of

players 100 as the threshold, we can improve the accuracy of clustering. The results of clustering using threshold categorization data can be seen in Figure 9.

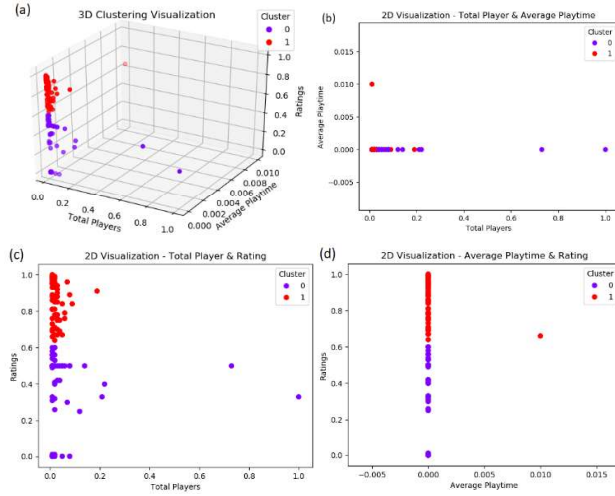


Fig. 9. 3D view (a), top view(b) of clustering result using regression data after threshold categorization.

In Figure 9, it can be seen that the results of clustering are better when compared to Figure 6. However, a new problem arises where most of the average playing time is 0. We can overcome this by only processing data that has an average playing time of more than 0. However, it will make the clustering results to be less good because it only produces little data. Therefore more data is needed to overcome this problem.

The next step is to calculate the accuracy value from the regression data. The accuracy value is calculated using the Mean Absolute Percentage Error (MAPE) method. In the MAPE method, the predicted data is reduced by the original data, then divided by the original data. After that, the data is multiplied by one hundred. For more details can be seen in Equation 2.

$$M = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (2)$$

In Equation 2, M is the accuracy value, A_t is the original data, and F_t is the regression data. The results of the calculation of the accuracy value can be seen in Table VII.

TABLE VII. EXAMPLE OF ACCURACY VALUE

| No | Feature | Original Data | Predicted Data | Accuracy Value (%) |
|----|------------------|---------------|----------------|--------------------|
| 1 | Total Player | 7 | 7 | 100 |
| 2 | Average Playtime | 448 | 448 | 100 |
| 3 | Rating | 0.95 | 0.95 | 100 |

In table 7, it can be seen the results of the calculation of the accuracy value. The accuracy value obtained is 100 percent. This is due to the lack of data so that the original data and predictive data have the same value. This problem can be overcome by adding more data to be predicted.

V. CONCLUSION

In this research, we have presented game data analytics to analyze and predicting game trends using descriptive and predictive mining. Our proposed system has 4 main features: (1) Data acquisition from Steam API, (2) Data preprocessing using Min-Max method, (3) Data Clustering using Hierarchical K-Means method, and (4) Data Prediction using Multiple Linear Regression method. For the applicability of our proposed system, we made a series of experimental studies using games and players data from March 2020 to April 2020 with 150000 numbers of data from Steam API. Our proposed application system provides information about current game trends and game trend predictions in the next 2 weeks. This new system can help game developers especially indie game developers to be able to plan games that will be made so that they can be sold in the market and reap a huge profit.

ACKNOWLEDGMENT (Heading 5)

We would like to thank Politeknik Elektronika Negeri Surabaya(PENS) for supporting this research.

REFERENCES

- [1] H. Baier, A. Sattaur, E. J. Powley, S. Devlin, J. Rollason, and P. I. Cowling, "Emulating Human Play in a Leading Mobile Card Game," *IEEE Trans. Games*, vol. 11, no. 4, pp. 386–395, 2018, doi: 10.1109/tg.2018.2835764.
- [2] V. Bonometti, C. Ringer, M. Hall, A. R. Wade, and A. Drachen, "Modelling early user-game interactions for joint estimation of survival time and churn probability," *IEEE Conf. Comput. Intell. Games, CIG*, vol. 2019-August, 2019, doi: 10.1109/CIG.2019.8848038.
- [3] J. A. Brown, A. Cuzzocrea, M. Kresta, K. D. L. Kristjansson, C. K. Leung, and T. W. Tebinka, "A machine learning tool for supporting advanced knowledge discovery from chess game data," *Proc. - 16th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2017*, vol. 2017-December, pp. 649–654, 2017, doi: 10.1109/ICMLA.2017.00-87.
- [4] C. W. Chen and T. Hsu, "Game development data analysis visualized with virtual reality," *Proc. 4th IEEE Int. Conf. Appl. Syst. Innov. 2018, ICASI 2018*, pp. 682–685, 2018, doi: 10.1109/ICASI.2018.8394349.
- [5] L. V. Fernandes, C. D. Castanho, and R. P. Jacobi, "A Survey on Game Analytics in Massive Multiplayer Online Games," *Brazilian Symp. Games Digit. Entertain. SBGAMES*, vol. 2018-November, pp. 21–30, 2019, doi: 10.1109/SBGAMES.2018.00012.
- [6] L. B. Jacob, T. C. Kohwalter, A. F. V. Machado, E. W. G. Clua, and D. De Oliveira, "A Non-intrusive Approach for 2D Platform Game Design Analysis Based on Provenance Data Extracted from Game Streaming," *Brazilian Symp. Games Digit. Entertain. SBGAMES*, vol. 2014-December, no. December, pp. 41–50, 2014, doi: 10.1109/SBGAMES.2014.33.
- [7] H. Kim, S. Lee, H. Lee, T. Hahn, and S. Kang, "Automatic generation of game content using a graph-based wave function collapse algorithm," *IEEE Conf. Comput. Intell. Games, CIG*, vol. 2019-August, pp. 1–4, 2019, doi: 10.1109/CIG.2019.8848019.
- [8] J. T. Kristensen and P. Burelli, "Combining sequential and aggregated data for churn prediction in casual freemium games," *IEEE Conf. Comput. Intell. Games, CIG*, vol. 2019-August, pp. 1–8, 2019, doi: 10.1109/CIG.2019.8848106.
- [9] B. Kutun and W. Schmidt, "Rallye game: Learning by playing with racing cars," *2018 10th Int. Conf. Virtual Worlds Games Serious Appl. VS-Games 2018 - Proc.*, pp. 1–2, 2018, doi: 10.1109/VS-Games.2018.8493440.
- [10] E. Lee et al., "Game Data Mining Competition on Churn Prediction and Survival Analysis Using Commercial Game Log Data," *IEEE Trans. Games*, vol. 11, no. 3, pp. 215–226, 2018, doi: 10.1109/tg.2018.2888863.
- [11] S. K. Lee, S. J. Hong, S. Il Yang, and H. Lee, "Predicting churn in mobile free-To-play games," *2016 Int. Conf. Inf. Commun. Technol. Converg. ICTC 2016*, pp. 1046–1048, 2016, doi: 10.1109/ICTC.2016.7763364.

- [12] M. R. Lewis et al., "Procedurally-Generate Gameplay within Mobile Games," 2018 10th Int. Conf. Virtual Worlds Games Serious Appl., pp. 1–4.
- [13] Q. Li, P. Xu, and H. Qu, "FPSSeer: Visual analysis of game frame rate data," 2015 IEEE Conf. Vis. Anal. Sci. Technol. VAST 2015 - Proc., pp. 73–80, 2015, doi: 10.1109/VAST.2015.7347633.
- [14] A. R. Martin and S. Colton, "Towards liveness in game development," IEEE Conf. Comput. Intell. Games, CIG, vol. 2019-August, pp. 1–4, 2019, doi: 10.1109/CIG.2019.8848092.
- [15] M. Mozgovoy, "Analyzing User Behavior Data in a Mobile Tennis Game," 2018 IEEE Games, Entertain. Media Conf. GEM 2018, pp. 449–452, 2018, doi: 10.1109/GEM.2018.8516512.
- [16] T. Mukherjee and S. Eswaran, "Towards mining of player intent for targeted gaming services," Proc. - 2018 IEEE World Congr. Serv. Serv. 2018, pp. 45–46, 2018, doi: 10.1109/SERVICES.2018.00041.
- [17] H. Park and K. J. Kim, "Learning to play fighting game using massive play data," IEEE Conf. Comput. Intell. Games, CIG, pp. 2–3, 2014, doi: 10.1109/CIG.2014.6932921.
- [18] I. J. Perez-Colado, D. C. Rotaru, M. Freire-Moran, I. Martinez-Ortiz, and B. Fernandez-Manjon, "Multi-level game learning analytics for serious games," 2018 10th Int. Conf. Virtual Worlds Games Serious Appl. VS-Games 2018 - Proc., pp. 1–4, 2018, doi: 10.1109/VS-Games.2018.8493435.
- [19] K. Samarngoon and A. Kunkhet, "An investigation of monetisation models in digital games," ECTI DAMT-NCON 2019 - 4th Int. Conf. Digit. Arts, Media Technol. 2nd ECTI North. Sect. Conf. Electr. Electron. Comput. Telecommun. Eng., pp. 64–68, 2019, doi: 10.1109/ECTI-NCON.2019.8692277.
- [20] E. S. Siqueira, C. D. Castanho, G. N. Rodrigues, and R. P. Jacobi, "A data analysis of player in world of warcraft using game data mining," Brazilian Symp. Games Digit. Entertain. SBGAMES, vol. 2017-November, pp. 1–9, 2018, doi: 10.1109/SBGAMES.2017.00009.
- [21] J. Tao and G. Wu, "Application and design of advanced algorithms in NoGo game of computer games," Proc. 28th Chinese Control Decis. Conf. CCDC 2016, pp. 4275–4278, 2016, doi: 10.1109/CCDC.2016.7531733.
- [22] D. Vihanga, M. Barlow, E. Lakshika, and K. Kasmarik, "Weekly seasonal player population patterns in online games: A time series clustering approach," IEEE Conf. Comput. Intell. Games, CIG, vol. 2019-August, pp. 1–8, 2019, doi: 10.1109/CIG.2019.8848108.
- [23] V. R. Warriar, J. R. Woodward, and L. Tokarchuk, "Modelling player preferences in AR mobile games," IEEE Conf. Comput. Intell. Games, CIG, vol. 2019-August, pp. 1–8, 2019, doi: 10.1109/CIG.2019.8848082.
- [24] W. Yang et al., "Mining player in-game time spending regularity for churn prediction in free online games," IEEE Conf. Comput. Intell. Games, CIG, vol. 2019-August, pp. 1–8, 2019, doi: 10.1109/CIG.2019.8848033.
- [25] W. Yang, G. Yang, T. Huang, L. Chen, and Y. E. Liu, "Whales, Dolphins, or Minnows? Towards the Player Clustering in Free Online Games Based on Purchasing Behavior via Data Mining Technique," Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018, pp. 4101–4108, 2019, doi: 10.1109/BigData.2018.8622067.