CHAPTER 4
ANALYSIS AND DESIGN

4.1 Algorithm Explanation

This project uses 2 different algorithms, namely J48 and Naive Bayes. J48 algorithm or decision tree is an algorithm that is good to use for classification problems as well as regression problems. The goal of this algorithm is to create a model that can predict the value of the target variable. The formula is :

Entropy(S) = E pi log2(pi)

Naive Bayes algorithm is a classification algorithm based on Bayes Theorem. It has a Conditional probability that measures the probability of an event occurring based on the event that already happened. Naive Bayes models are easy to build and very useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even very sophisticated classification methods.The formula is :

P[A/B] = p[B/A] * P[A].

4.2 The Attributes

We have to determine first the attribute that will be used in this project. The attributes that will be used in this project are Genre(Action,Adventure,Romance,School) , Manga Title (Dragon ball, Naruto , Gundam) , Score, Popularity, Members, Favorites, Score Voted By.

For the decision tree, it is necessary to determine the weight value that will be used as a comparison. The value of the weight can be determined at any rate. However, it must be remembered that the total weight is 1. In this study, the attributes used are Score, Popularity, Member, Favorites, Score vote by then the weight value for each attribute is 0.1, 0.3, 0.3, 0.2, 0.1 for the following reasons: Score is the value given by readers who read manga, Popularity is the popularity of manga, Member is the number of people who have read it, Favorite is the person who has read the manga. read manga and like it, the selected score is people who have read the manga and leave positive or negative comments.

4.3 Data

Table 4.1 Data that will be used in this project.

Manga Title	Genres	Score	Ranked	Popularity	Members	Favorites	Score Voted By
Berserk	['Action', 'Adventure', 'Demons', 'D	9.33	1	4	296958	61992	139571
Jo Jo Imyou na t	['Action', 'Adventure', 'Mystery', 'H	9.21	2	50	97073	18083	54901
Death Note	['Mystery', 'Drama', 'Shounen', 'Su	8.73	36	8	247508	29035	149712
Real	['Drama', 'Sports', 'Psychological',	8.73	37	288	31030	1394	7176
WorldEnd: What	['Drama', 'Fantasy', 'Romance', 'S	8.65	60	657	15535	625	3139
BEASTAR	['Drama', 'Shounen', 'Slice of Life'	8.65	61	178	43936	4035	16838
Boku Dake Imai	['Mystery', 'Supernatural', 'Psycho	8.44	141	117	60674	2680	28323
Claymore	['Action', 'Adventure', 'Fantasy', 'H	8.34	221	34	114739	10819	56807
Crows	['Action', 'School', 'Shounen']	8.18	405	776	13563	593	4699
Pandora Hearts	['Action', 'Fantasy', 'Magic', 'Super	8.12	496	3555	3270	65	575

If the data is INT, Naive Bayes must first calculate the average of each genre. After that, add up all the values of each genre variable minus the average of each genre variable powered by 2.

SUM(score[‘genre’] - AVG(score[‘genre’])^2)/COUNT(genre) -1 .

The final sum result then divides the value of each genre variable by subtracting the average of each genre variable.

((Score[‘genre’]-AVG(score[‘genre’])/(SUM(score[‘genre’ - AVG(score[‘genre’)))^2).

The 10 best scores obtained from each genre are the results of recommendations according to Naive Bayes. While for the decision tree, generally use the formula:

Entropy(S) = E pi log2(pi)

But since all the data here are INT, then we have to use a different way. First find the average of each variable. Then give a weight to each variable, for example 0.3 for popularity, 0.3 for members, 0.2 for favorites, 0.1 for scores and selected scores. After that, each data will be multiplied by the weight of each variable and if the sum of all data is less than the average then it is not included in the recommendation.

4.4 Design

4.4.1 Flowchart

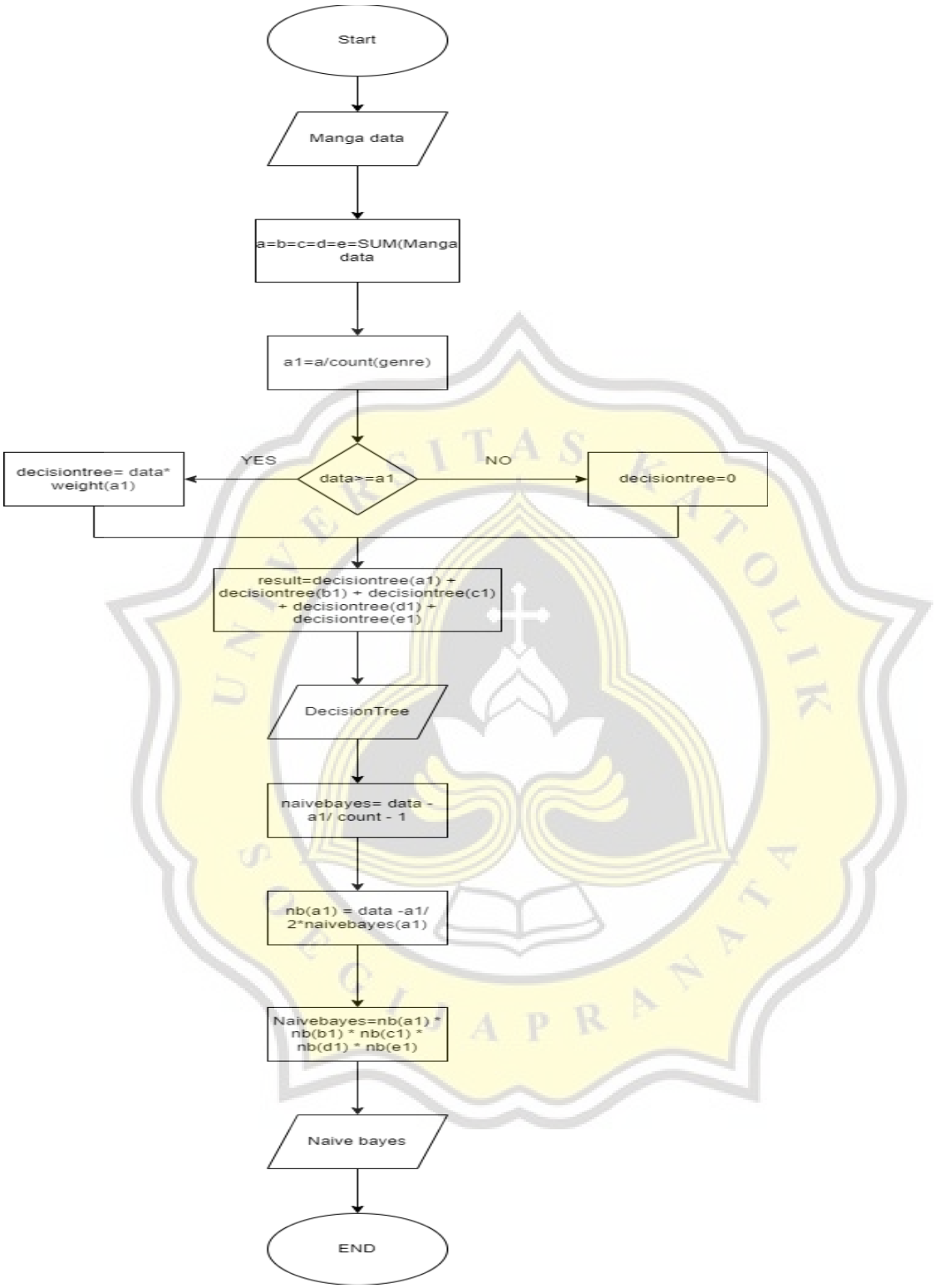

Figure 4.1 Flowchart of program

The figure 4.1 is about the flowchart of the program. Manga data that is input in this program is a manga dataset that is used in this project. First starting from the manga data input first. After input, the program will find the average of each data by genre. For example, if we are looking for an action genre, then for the part **a=b=c=d=e=SUM(data)** in flowchart, a= SUM(data['Score']) on condition that there must be an action genre in it. b is SUM(data['Popularity'], c is SUM(data['Member']). As for **a1=a/count,** The count used is a count of manga that has an action genre in it. for example, there are 500 manga data while the action genre is 162. Then the calculated value is 162.

After getting the results, the results are stored in the variables a1(Score), b1(Popularity), c1(Members), d1(Favorites), e1(Score Voted By). Then look for manga whose value is score > a1, Popularity value > b1, Member value > c1, and so on until the score is Voted By > e1. If it is greater than a1/b1/c1/d1/e1, then the decision tree value is the data value multiplied by a predetermined weight. For example, the weight for the score is 0.1, Popularity is 0.3, and the member is 0.2 based on self-assessment of the data used. If it is smaller than a1/b1/c1/d1/e1, then the decision tree value(a1/ b1/c1/d1/e1)=0.

After that the result and the decision tree is :

Decision Tree = SUM(decision tree from a1 to e1)

For naive bayes, variables a1 to e1 are still used. For example, we want to find a manga with the action genre. then naivebayes(a1) :

naive bayes(a1) = (data['Score']-a1)/count['Action']-1.

then the value of naivebayes(Score) is :

naive bayes(Score) = (data['Score'] - a1)/2*naive bayes(a1)

After that the final score for Naive bayes is :

Naive bayes = naivebayes(Score) * naive bayes(Popularity) * naive bayes(Member) * naive bayes(Favorite) * naive bayes(Score Voted By).