

CHAPTER 5 IMPLEMENTATION AND TESTING

Implementation

Array is used as data structure to store employee data and weight value of variables used for performance calculation because it is easier to call on each method than other data structure method. Employee data used in this research is from January to June 2018 and inputted into SQL format.

First, the program will check the connection with SQL database. The program will send message to user whether the connection between program and database established or not.

Next, the program will fetch weight value from database and put it into arraylist. Variable of each weight then called to fill new variable as shown on following code

```
1.  int babsen = bbt.get(0);
    2.  int bjualcash = bbt.get(1);
    3.  int bjualkredit = bbt.get(2);
    4.  int blkerja = bbt.get(3);
5.  int bstatus = bbt.get(4);
7.  int sumweight = babsen + bjualcash + bjualkredit + blkerja
    + bstatus;
8.
9.  float bobotabsen = (float) babsen/sumweight;
10. float bobothjualcash = (float) bjualcash / sumweight;
11. float bobothjualkredit = (float) bjualkredit /
    sumweight;
12. float bobotlkerja = (float) blkerja / sumweight;
13. float bobotstatus = (float) bstatus / sumweight;
```

Each weight value then summed and normalized using formula in line 9-13. The results will be like following illustration

```

Bobot Absensi = 0.25
Bobot Hasil Jual Cash = 0.2
Bobot Hasil Jual Kredit = 0.25
Bobot Lama Kerja = 0.15
Bobot Status Pekerjaan = 0.15

```

Illustration 5.1: Normalized Weight Value

After finished fetching weight value data from database, the program will fetch employee performance data and put it into arraylist, then the arraylist converted into array using following code

```

1. String[] arrnama = (String[])nm.toArray(new String[nm.size()]);
2. Float[] arrabsen = (Float[])absen.toArray(new
Float[absen.size()]);
3. Integer[] arrhjualcash = (Integer[])hjualcash.toArray(new
Integer[hjualcash.size()]);
4. Integer[] arrhjualkredit = (Integer[])hjualkredit.toArray(new
Integer[hjualkredit.size()]);
5. Integer[] arrlkerja = (Integer[])lkerja.toArray(new
Integer[lkerja.size()]);
6. Integer[] arrstat = (Integer[])status.toArray(new
Integer[status.size()]);

```

The reason for arraylist conversion into array is to make data calling in looping process in Weighted Product Algorithm easier. Because variable value in Weighted Product Algorithm cannot have zero (0) value in it, while Cash and Credit Sales Results have zero value in it, it needs to be classified into specific value. The specific value is shown on table below

Table 5.1 Grouping of Sales Results Value

Sales Results (Cash and Credit)	Value
>20	5
10-20	4
5-9	3
2-4	2
<2	1

After the sales results classified, the calculation process initialized by program. First step is to find the Vector S of each employee data with formula below

```

1. W1 = (float) Math.pow(arrabsen[i], bobotabsen);
2. W2 = (float) Math.pow(arrhjualcash[i], bobothjualcash);

```

```

3. W3 = (float) Math.pow(arrhjualkredit[i], bobothjualkredit);
4. W4 = (float) Math.pow(arrlkerja[i], bobotlkerja);
5. W5 = (float) Math.pow(arrstat[i], bobotstatus); 6.
7. float NilaiVektorS = (W1*W2*W3*W4*W5);
8. VektorS[i] = (NilaiVektorS);

```

Vector S of each employee data stored into VektorS[] array, then used to count Vector V. To count Vector V, all Vector S stored in array must be summed and stored in SumVektorS variable. SumVektorS will be used as divider to search Vector V. After sum process of Vector S is done, the process to calculate Vector V will be executed. Vector V of employee needs to be separated based on job status, and it can be done using if-else function. First, array of VektorVTetap and VektorVMagang must be declared in order to contain separated Vector V data. After that, if function declared in looping process with formula for counting Vector V putted in it, with condition below

Table 5.2 Job Status

Job Status	Value
Tetap	2
Trainee/Magang	1

```

1. for(int k=0; k<VektorS.length; k++){
2.   if(arrstat[k]==1) {
3.     float NilaiVektorV = VektorS[k]/SumVektorS;
4.     VektorVTetap[k] = (NilaiVektorV); 5.   }
6. }
7.
1. for(int m=0; m<VektorS.length; m++){
2.   if(arrstat[m]==2) {
3.     float NilaiVektorV = VektorS[m]/SumVektorS;
4.     VektorVMagang[k] = (NilaiVektorV); 5.   }
6. }

```

If arrstat array contains "1" value, Vector V will be calculated and putted in VektorVTetap array, The same goes to arrstat array contains "2" value, Vector V will be calculated and putted in VektorVMagang array as shown on code above. The results will be like this following illustration

Vektor V Pegawai Tetap :

- Sunhaji = 0.091098286
- Budiono = 0.09780208
- M. Chabibbur Rohman = 0.09298841
- Noven Setiawan = 0.09174701
- Galuh Titis Randi Kartika = 0.07549183
- Nur Sinungan Rohmat = 0.06209288

Vektor V Pegawai Magang :

- Ayu Dian Rosa = 0.059384238
- Bayu Tirtarama = 0.04890417
- Muhammad Dzawinnuha = 0.056785733
- Indah Aprilia = 0.050435092
- Misbakhul Munir = 0.056176133
- Sudarlan = 0.049935993
- Manasik = 0.057934705
- Wahyu Bagus Jatmiko = 0.050435092
- Mei Janu Siswanto = 0.058788273

Illustration 5.2: Employee's Vector V Separated with If-Else Function

As shown above, Vector V are separated. The calculation using weighted product algorithm finished here. Program then determine which employee has the best performance and its ranking. Employee with best performance is decided by biggest Vector V value, so it needs additional algorithm to compare each data in array. It also needs another sorting algorithm for performance ranking from best to worst. Below is sample code to search best employee

```

1. for(int l=0; l<VektorVTetap.length; l++){
2.   if(arrstat[l] == 1 && PegawaiTetapTerbesar<VektorVTetap[l]){
3.     PegawaiTetapTerbesar = VektorVTetap[l];
4.     NamaPegawaiTerbaik = arrnama[l]; 5.   }
6. }

```

To search biggest Vector V value, declaration of temporary data container needs to be done for comparison with Vector V in array. Variable of data container contains Vector V from the first data in Vector V array. With that, comparison operation can be initialized. Value in data container will be replaced with Vector V in array if value in Vector V array is bigger. The process will run until there are no values in Vector V array available to be scanned. Results of this algorithm will be printed like illustration below

Pegawai Terbaik Bulan ini adalah :
 Budiono dengan performa 0.09780208
 Pemegang Terbaik Bulan ini adalah :
 Ayu Dian Rosa dengan performa 0.059384238

Illustration 5.3: Best Employee Results

After comparison process, the program will proceed into ranking process. First, data in Vector V array is put into container. The data in this container then compared with all data in array of Vector V, if it's smaller, it will be put on behind the compared data, and it will stop if there are no smaller values in it. The data container then fetch another value then execute comparison process again. The results of this program ranking process is shown in following illustration



Illustration 5.4: Employee Ranking after Sorting Process

In conclusion, the output of this program will show Normalized Weight Value, Vector S, Separated Vector V, Best Employee based on their Job Status and Vector V ranking from best to worst employee performance by their Vector V.

Testing

Manually calculated performance and calculated performance with this system is compared to analyze the difference between it. The manual performance calculation is using Sales Results as calculated variable, while the system are

using Absence Rate, Cash Sales Results, Credit Sales Results, Work Period and Job Status as calculated variables.

Table 5.3 Best Employee with Manual Calculation

Month	Best Employee
January	Chabibbur Rohman
February	Chabibbur Rohman
March	Chabibbur Rohman
April	Chabibbur Rohman
May	Chabibbur Rohman
June	Chabibbur Rohman

The results is shown on table above. Manual calculation does not separate employee based on their Job Status. It also using Sales Results (Cash and Credit summed into one) as their calculation variable. The results of this calculation is solely based on how many sales results employee produced in one month.

For results comparison, below is the table of performance calculation using this program

Table 5.4 Best Employee with Weighted Product - Jobholders

Month	Best Employee
January	Budiono
February	Chabibbur Rohman
March	Sunhaji
April	Chabibbur Rohman
May	Sunhaji
June	Sunhaji

Table 5.5 Best Employee with Weighted Product – Interns

Month	Best Employee
January	Ayu Dian Rosa
February	Manasik
March	Wahyu Bagus Jatmiko
April	Manasik
May	Sudarlan

The system separated and picked the best employee based on their job status. The program is separating Sales Results into Cash Sales Results and Credit Sales Results. It also used absence rate, job status and work period as additional variable to calculate employee, makes the results of performance calculation as shown in tables above more variative. Because of data separation process mentioned in implementation section, the system gave two best employee based on their job status as output.

Main difference that distinct the results of these two calculations are the amount of variables used for performance calculation. Absence rate, job status and work period variables in system calculation can make significant difference in determining best employees, while manual calculation focus solely on sales results of its employees, makes it more sales results-oriented.

From descriptions above, benefit and drawback between those two performance calculation are:

Manual Calculation

Table 5.6 Benefit and Drawback of Manual Calculation

Benefit	Drawback
+ Simpler to implement because it only search employee with biggest sales results	- Quite difficult to pick or rank the best employee if there are couple or more sales results between employees
+ The calculation can be done by either Excel or written in paper, makes it user-friendly	-

System Calculation with Weighted Product

Table 5.7 Benefit and Drawback of System Calculation

Benefit	Drawback
+ More efficient in calculating performance because the algorithms are already implemented in system	- Since its using SQL to operate and user must run MySQL software via terminal, this system is less user-friendly than Manual Calculation
+ Best employee in this system are picked separately based on two job status, gives more variative results for recommendation	-

+ Ranking process is quite easy to done since almost every employee have more variables to calculate, makes the calculated results variative between its employees

