

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Analysis

This project will make program which can solve assignment problem by Hungarian algorithm. To solve assignment problem, the first thing to do is preparing workers, jobs, and costs data. Here is the example :

Table 4.1: Example Source Data

	Cleaning Bathroom (1)	Planting Flower (2)	Cleaning Floor (3)	Cleaning Window (4)
Andre (A)	80.000	40.000	50.000	46.000
Budi (B)	40.000	70.000	20.000	25.000
Corry (C)	30.000	10.000	20.000	30.000
Durian (D)	35.000	20.000	25.000	30.000

There are workers, jobs, and costs data in table 4.1. The workers are Andre, Budi, Corry, Durian. The jobs are cleaning bathroom, planting flower, cleaning floor, and cleaning window. Each worker has certain cost to do jobs. To do job cleaning bathroom, Andre's cost is Rp. 80.000, Budi's cost is Rp. 40.000, Corry's cost is Rp. 30.000, and Durian's cost is Rp. 35.000. To do job planting flower, Andre's cost is Rp. 40.000, Budi's cost is Rp. 70.000, Corry's cost is Rp. 10.000, and Durian's cost is Rp. 20.000. To do job cleaning floor, Andre's cost is Rp. 50.000, Budi's cost is Rp. 20.000, Corry's cost is Rp. 20.000, and Durian's cost is Rp. 25.000. To do job cleaning window, Andre's cost is Rp. 46.000, Budi's cost is Rp. 25.000, Corry's cost is Rp. 30.000, and Durian's cost is Rp. 30.000.

Then, datas above are inserted and mapped into table to be processed.

Table 4.2: Process 1

	1	2	3	4
A	80	40	50	46
B	40	70	20	35
C	30	10	20	30
D	35	20	25	30

After the datas inserted into table, the next step is normalizing row. First, it needs to find the most minimum cost each row. From the table 4.2, in row A the most minimum cost is 40, in row B the most minimum cost is 20, in row C the most minimum cost is 10, and in row D the most minimum cost is 20. Then each row will be subtracted by their most minimum cost. Costs in row A will be subtracted by 40, costs in row B will be subtracted by 20, costs in row C will be subtracted by 10, and costs in row D will be subtracted by 20.

Table 4.3: Result of Normalizing Row

	1	2	3	4
A	40	0	10	6
B	20	50	0	5
C	20	0	10	20
D	15	0	5	10

Then, the next step is normalizing column. First, it needs to find the most minimum cost each column. From the table 4.3, in column 1 the most minimum cost is 15, in column 2 the most minimum cost is 0, in column 3 the most minimum cost is 0, and in column 4 the most minimum cost is 5. Then each column will be subtracted by their most minimum cost. Costs in column 1 will be subtracted by 15, costs in column 2 will be subtracted by 0, costs in column 3 will be subtracted by 0, and costs in column 4 will be subtracted by 5.

Table 4.4: Result of Normalizing Column

	1	2	3	4
A	25	0	10	1
B	5	50	0	0
C	5	0	10	15
D	0	0	5	5

If 0 cost is obtained in each row and column after the assignment, the process is stop because the assignment is optimal. But if it is not, the next step is taking 0 costs only based on the table.

Table 4.5: Only Zero Cost

	1	2	3	4
A		0		
B			0	0
C		0		
D	0	0		

Now 0 cost can be seen in each row and column. Based on table 4.5, job 1 can be taken by worker D, job 2 can be taken by worker A, B, or C, job 3 can be taken by worker B, and job 4 can be take by worker B. Then the jobs are assigned to the worker. Job 1 is assigned by worker D. Because worker D has been assigned to job 1, so worker D can not take other job. Job 2 has two options workers to be assigned, they are A and C. Because worker A and C only can take job 2, so job 2 will be assigned to one of them. For this, job 2 is assigned to worker A. Job 3 is assigned to worker B. Because worker B has been assigned to job 3, so B can not take other job. So worker C has not job, and job 4 has not worker. This is not solution, so it needs to do the next step. The step is determining the minimum cost of lines (horizontal or vertical) which are required

to cover all 0 costs in the table. All 0 costs can be covered using three lines. See table 4.5.

Table 4.6: Covered Zero Cost

	1	2	3	4
A	25	0	10	1
B	5	50	0	0
C	5	0	10	15
D	0	0	5	5

After covering 0 costs in lines, the next step is finding the most minimum cost in uncovered costs. It is 1. Then the uncovered costs are subtracted by 1, while the costs in the cross line (row B with column 2, and row D with column 2) are added by 1.

Table 4.7: Result of Subtracted and Added by Smallest Cost

	1	2	3	4
A	24	0	9	0
B	5	51	0	0
C	4	0	9	14
D	0	1	5	5

Then the next step is taking 0 costs only based on the table 4.7.

Table 4.8: Only Zero Cost

	1	2	3	4
A		0		0
B			0	0
C		0		
D	0			

From table 4.8, the result is job 1 is assigned to worker D, job 2 is assigned to worker C, job 3 is assigned to worker B, and job 4 is assigned to worker A. Because all jobs have been assigned to worker, so the process is done. The conclusion is :

Cleaning bathroom is assigned to worker Durian with cost Rp. 35.000.

Planting flower is assigned to worker Corry with cost Rp. 10.000.

Cleaning floor is assigned to worker Budi with cost Rp. 20.000.

Cleaning window is assigned to worker Andre with cost Rp. 46.000.

4.2 Desain

Design Input

There is textfield in GUI to user inputs number of data. The data will be generated by program automatically in stored procedure spGenerateAngka(). The function of stored procedure spGenerateAngka() is generating random numbers based on the user input. Then, the random numbers are inserted into table. These random numbers are as the costs data.

Design Process

The data will be mapped into table process in stored procedure spInputIntoTable().

Table 4.9: Sample of Table Process

	Job 1	Job 2	Job 3	Job n
Worker 1	cost	cost	cost	cost
Worker 2	cost	cost	cost	cost
Worker 3	cost	cost	cost	cost
Worker n	cost	cost	cost	cost

The computing processes are :

- The process to find minimum cost in each row is in stored procedure spFindMinimBaris(). The function of stored procedure spFindMinimBaris() is finding the minimum cost in each row and inserting them into table.
- The process to normalized row is in stored procedure spNorRow(). The function of stored procedure spNorRow() is normalizing the rows by subtracting costs in each row with its minimum cost which has been inserted into table.
- The process to find minimum cost in each column is in stored procedure spFindMinimKolom(). The function of stored procedure spFindMinimKolom() is finding the minimum cost in each column and inserting them into table.
- The process to normalized column is in stored procedure spNorCol(). The function of stored procedure spNorCol() is normalizing the columns by subtracting costs in each column with its minimum cost which has been inserted into table.
- The process to find the 0 cost only is in stored procedure spFindNol(). The function of stored procedure spFindNol() is finding the workers and jobs which have 0 cost and inserting them into table.
- The process to sort the result of 0 cost data only based on the most jobs can be assigned by one worker is in stored procedure spSortTableTemp(). The function of stored procedure spSortTableTemp() is sorting data in table result of stored procedure spFindNol() based on the most jobs can be taken by one worker, and input them into table.

- The process to store the result is in `spInputIntoTableHasil()`. The function of stored procedure `spInputIntoTableHasil()` is inserting data result of matching process into table.
- The process to check if the result is perfect matching or not is in `spCheckMatching()`. The function of stored procedure `spCheckMatching()` is checking the final result that it is perfect matching or not. It works by matching the number of data from the final result with the source data.
- The process to find the smallest cost in uncovered line is in `spFindSmallest()`. The function of stored procedure `spFindSmallest()` is finding the smallest cost among costs in uncovered line. Because the workers and jobs in covered line have been saved before, so this stored procedure works by finding the smallest cost with workers and jobs which have not been saved.
- The process to normalize result which is not optimal is in `spNorMatch()`. The function of stored procedure `spNorMatch()` is normalizing the result which is not optimum by subtracting the uncovered costs with the smallest cost, and adding the cross line cost with the smallest cost.