# CHAPTER V

## IMPLEMENTATION AND TESTING

### 5. 1 Implementation

First of all this application need Mysql database is already installed. Thus, this application need to insert penjadwalan.sql into database named dbPenjadwalan inside phpmyadmin.

#### 5.1.1 Database

This application using Mysql as database. The database itself is written in penjadwalan.sql at folder data. The database host is localhost, username is root, and name is dbPenjadwalan. The database contain 10 tables to storing data. Connection to the database is saved in dbConfig.php which it also contain some function to fetch data from database, so when the application need to select data from database, the application will just do the query and call fetch function in dbConfig.php.

```
<?php
class dbconfig{
    private $host="
    private $username="
    private $password="";
    private $database="
    private $query;
    private $hubungkan="";

    // __construct merupakan fungsi yang dijalankan sebagai default fungsi ketika pembuatan objek
    public function __construct(){
        $this->hubungkan = mysqli_connect($this->host,$this->username,$this->password,$this->database);
        return $this->hubungkan; // fungsi __construct ini akan mengembalikan nilai true jika $this->hubungkan berhasil
    }

    // di bawah merupakan fungsi untuk melakukan query
    public function query($sql){
        $this->query = mysqli_query($this->hubungkan,$sql) or die (mysqli_error($this->hubungkan));
        return $this->query;
    }

    public function numrow($sql){
        $this->query = mysqli_num_rows(mysqli_query($this->hubungkan,$sql));
        return $this->query;
    }

    // hasil query di jadikan array pada fungsi di bawah ini
    public function fetch($hasil_sql){
        return mysqli_fetch_array($hasil_sql);
    }
```

*Figure 5.1.1 dbconfig*

### 5.1.2 Tree

As said above, this application has binary tree to storing data. And the contents of binary tree node is an array which redeclared into variables. There are about 21 variables in each node of binary tree. These variables contain a part of data each courses which is selected from database.

```php
<?php
CLASS BinaryNode
{
    PUBLIC $isi = array();    // contains the node item
    PUBLIC $index, $posisi, $velocity, $kelas, $matkul, $dosen, $fitness, $namaDosen, $namaMatkul, $sks;
    PUBLIC $senin, $selasa, $rabu, $kamis, $jumat, $sabtu, $tujuh, $sembilan, $sepuluh, $duabelas, $tigabelas, $limabelas;
    PUBLIC $ka, $ki;
```

*Figure 5.1.2 tree*

### 5.1.3 Initiation

This application using Particle Swarm Optimization as main algorithm, which this algorithm first step is random all course's position that no have fixed position. Well, in randoming course's position it must not have same place with the other course's position. So the previous position is saved into an array as flags for the next position so that would not get same with previous position.

```
96
97      PUBLIC FUNCTION randomPosisi($kelas, $sks){
98          //cari x
99          for($k=0; $k<$this->X; $k++){
100             if($this->harikelas[$k] == $kelas) break;
101         }
102
103         $this->yrand = rand(0, ($this->Y-1));
104
105         //cek posisi data apakah ada yg sama dengan posisi khusus dan umum
106         $this->cekPosisi($k, $this->yrand, $sks);
107
108         //sudah tidak ada posisi yang tumpukan
109         //hitung posisi dari x,y
110         $tmp = ($this->Y * $k)+$this->yrand;
111         $tmpsks = ($this->Y * $k)+$this->ysks;
112
113         $this->i = $this->i+1;
114         $this->post[$this->i] = $tmp;
115         $this->postsks[$this->i] = $tmpsks;
116
117         RETURN $this->post[$this->i];
118     }
```

*Figure 5.1.3 Random Position*

After all courses get position then it's time to check whether the position is optimum with the terms of optimization. So it will select all data that stored in binary tree using in-order function. The default number of fitness is same with number of terms that used. The number of fitness will decrease by 1 for each terms of optimization that are not violated. So if all terms of optimization are not violated then the number of fitness would be 0 which means it's good optimization. But when the terms of optimization get violated, the number of fitness will increase by 1 for each violation.

```
//cek dosen ngajar 1x sehari
if($op3 <> 0){
$sekaliDosen = $cek->hariSekaliDosen($koor->getYcari(), $rootNode->dosen);
if($sekaliDosen == 0){
    $fitness=$fitness-1;
    $rootNode->fitness = $fitness;
} else {
    $fitness=$fitness+1;
    $rootNode->fitness = $fitness;
}
}

//1 hari max 6 sks per semester
if($op4 <> 0){
$sksSehari = $cek->sksHari($koor->getXcari(), $koor->getYcari(), $rootNode->sks);
if($sksSehari == 0){
    $fitness=$fitness-1;
    $rootNode->fitness = $fitness;
} else {
    $fitness=$fitness+1;
    $rootNode->fitness = $fitness;
}
}
```

*Figure 5.1.3 Number of Fitness*

While the course's position is not optimum, the course's position will be changed. That changed position is based on algorithm function. The course's position would be checked into some terms to made the course's position will not get cross days and cross another course's position; whether it courses with fixed position or not. However the course's position that stored in binary tree will be changed into new position and will be checked again with terms of optimization.

```
PUBLIC FUNCTION kecepatan($v, $x, $y, $sks, $index){

    $this->kecepatan = ($this->weight * $v) + (($this->C1 * $this->r1) * ($this->PBest - $y)) + (($this->C2 * $this->r2) * ($this->GBest - $y));
    $this->ybaru = round($this->kecepatan + $y);


    //jik posisi lebih dari matrix di loop sehingga kembali ke matrix
    WHILE($this->ybaru < 0){
        $this->ybaru = $this->ybaru + $this->maxY;
    }
    WHILE($this->ybaru > $this->maxY-1){
        $this->ybaru = $this->ybaru - $this->maxY;
    }

    $this->tidakSamaPindah($x, $this->ybaru, $sks);

    //jika ada posisi yang sama tidak sama antar data next tidak sama dengan data matkul khusus
    $this->i = $this->i+1;
    $this->temposisi[$this->i] = (($this->maxY * $x)+$this->ybaru);
    $this->temposisisks[$this->i] = ($this->maxY * $x)+$this->ybaru+($sks-1);
    //echo $this->i." : ".$this->temposisi[$this->i]." , ".$this->temposisisks[$this->i]."</br>";

    //RETURN $this->temposisi[$this->i];
    //set i jadi 0 sehingga temporary selanjutnya akan di tumpuk dengan data baru
    RETURN $this->temposisi[$this->i];
}
```

*Figure 5.1.3 Change position*

### 5.1.4 Text file

This application allowed to save the schedule optimization into a text file. This application also allowed to give a name for the text file. While save button is clicked, the application will download the text file. The schedule optimization cannot be seen in text file, because it's not in table. But if the schedule optimization really wants to loaded again, the schedule can be loaded again as timetable schedule in the application.

```
for($i=0; $i<$max; $i++){
    if(!isset($_POST[$i])) $_POST[$i] = " - ";
    $isi[$i] = $_POST[$i];
    $isi[$i] = $isi[$i]."|";
    if ($i % $y == 0 && $i <>0) $isi[$i-1] = $isi[$i-1]."\n";
}
//$isi2 = $_POST['32'];

$contents = "$contents\n".$isi[0];
for($i=1; $i<$max; $i++){
    $contents = "$contents".$isi[$i];
}


echo $contents;

header('Content-Type: application/gedit');
header('Content-Disposition: attachment;filename='.$filename);
header('Cache-Control: max-age=0');

?>
```

*Figure 5.1.4 Save text file*

## 5.2 Testing

This test is to make schedule optimization for second semester with few data lecturer, courses and 5 classes; in order of setting maximum looping 1000 and 4 terms of optimization is ON.

### 5.2.1 Inserting Data

Well, at first insert all data, there are: lecturers, courses, and classes. In lecturer's data there are 3 attributes: lecturer's nip, lecturer's name, and lecturer's address. Set the lecturer's days and hours when the lecturers have a time for teaching. In this case all lecturer's days and hours is ready or that means all lecturer's can teach at any days (Monday - Saturday) and at any hours (at 07:30 – 16:30). Then insert data courses, which have course's id, course's name, and number of credits. Next add connection between lecturer and course which means lecturer A taught Advanced Programming course. Therefore insert data classes which only have class id and class name.


*Figure 5.2.1 Insert data*

### 5.2.2 Adding connection

After inserting all data lecturers, courses, and classes, the complicated part of inserting data is adding connection between courses and classes. In this connection the data will be divided into two semester which means the courses is in first semester or in second semester. In this application there are two types of connection between courses and classes:

1. **Courses with fixed position:** this connection means, the course's position would not be changed by algorithm while scheduling. So this position required data course, data class, hours, and days. It will make course' position on day x and hour y in class z, and this position would not be changed. But the course's position still can be changed from main menu.
2. **Courses which can change position:** this connection all course's position to swap place. This connection is the place where particle swarm optimization algorithm working. The change of position is only change day and hour without the class. So while adding connection between courses and classes, it's only need data course and data class, it did not need for day and hour. The day and hour in the connection will change when it gets violation to terms of optimization.


*Figure 5.2.2 Connection course and class fixed position*

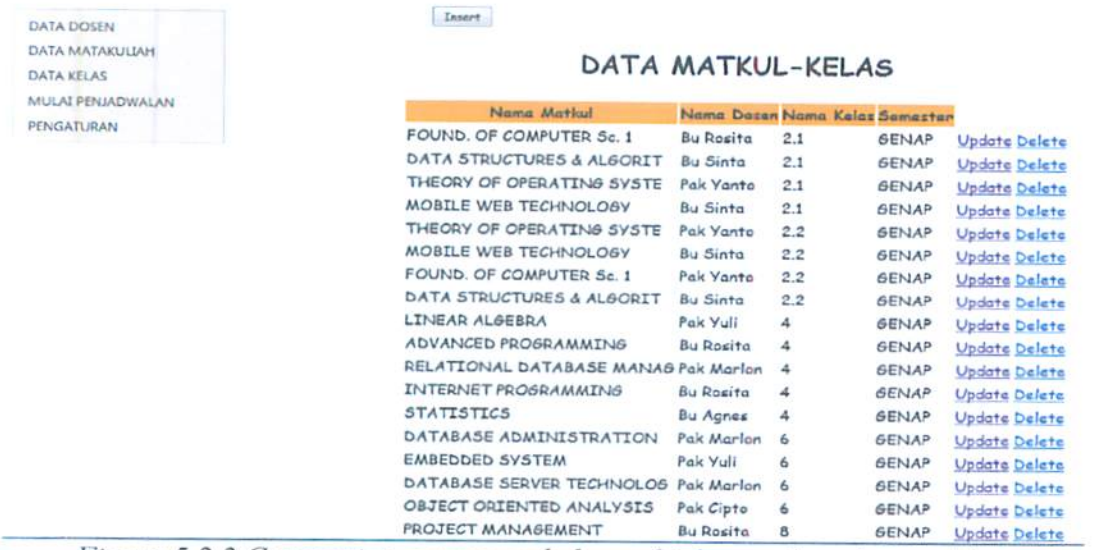


*Figure 5.2.2 Connection course and class which can changing position*

### 5.2.3 Scheduling

To start scheduling choose which semester that the schedule optimization want to be made. In this case choose second semester, thus it will show a schedule optimization with second semester data(lecturer, course, class). There are 6 days from Monday to Saturday, and 6 groups of hour from 07:30 until 16:30. In the appearance of scheduling optimization, there are loop which means maximum of loop, and number of fitness. This application can tell whether the schedule optimization is optimum or not by seeing the number of fitness. If the number is not 0, the schedule optimization has not been optimum yet. If the number of fitness is 0, so the schedule optimization has been optimum. When the schedule optimization need to be saved, click save button at top right corner. Before save the file, don't forget to give the name of the file in the text box. While saving the file, the schedule optimization will be directly downloaded as text file. If the saved schedule need to be loaded again, go to main menu→Start Scheduling, choose the text file by clicking browse button at the bottom and click process to execute.

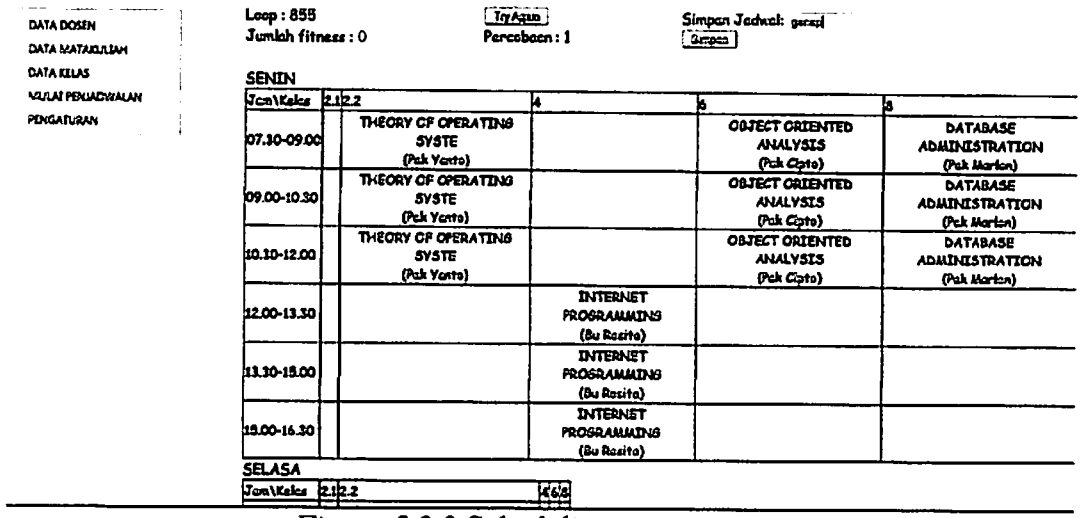


*Figure 5.2.3 Scheduling optimization*

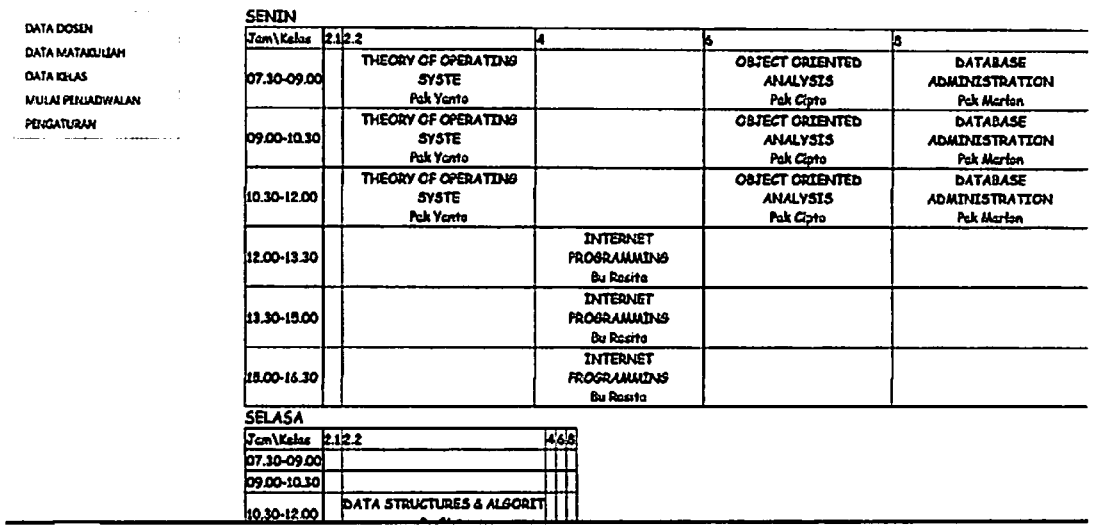


*Figure 5.2.3 Scheduling optimization loaded from text file*