

Figure 4.2 Class Diagram

CHAPTER V

IMPLEMENTATION AND TESTING

5. 1 Implementation

First Step : we need to make sure that Mysql is already installed. After that, we need to open a browser and enter the localhost entry search that phpmyadmin afterwards go import and search in TA_Aakhir/data/db_jadwal.sql and go. After that enter localhost/TA_Aakhir and check krs, courses, the room, and dosen already exists. If all done there then click generate then all data will be processed automatically.

5.1.1 Step 1 - Early process

- create dbconfig.php to connect to the database using the create object in php
- In this coding I made a php object for processing the database that will be used is included into php in this coding. I created an object that is useful for calling existing data in phpmyadmin. That will come in handy to hold the data and process data.

- This is code dbconfig.php:

```

1 /**
2  * @author class untuk database
3  */
4 class Database{
5     private $host="127.0.0.1";
6     private $username="";
7     private $password="";
8     private $database="";
9     private $query;
10    private $result;
11
12    //__construct merupakan fungsi yang dijalankan sebagai default fungsi ketika pembuatan object
13    public function __construct(){
14        $this->connung = mysqli_connect($this->host,$this->username,$this->password,$this->database) or die ("tidak berhasil");
15        return $this->connung; // fungsi __construct ini akan memperbaiki nilai true pada $this->connung berarti
16    }
17
18    // di bawah merupakan fungsi untuk melaksanakan query
19    public function query($sql){
20        $this->query = mysqli_query($this->connung,$sql) or die mysqli_error($this->connung);
21        return $this->query;
22    }
23
24    // hasil query di jadikan array pada fungsi di bawah ini
25    public function fetch($hasil_query){
26        return mysqli_fetch_array($hasil_query);
27    }
28
29    // di bawah ini adalah fungsi untuk menutup koneksi ke database
30    public function close(){
31        return mysqli_close($this->connung);
32    }
33}

```

Figure 5.1 Binary Code dbconfig.php

5.1.2 Step 2 - Input and Display

- Display data Krs and input data for krs using php object :
- In this coding I made a php object (krsmahasiswa. php) that the content coding to display data that is taken from (dbconfig. php) so that it appears in the browser and can be seen.
- in this coding I use the include function to call (dbconfig. php) and create a new class so that the data can be stored and can be displayed.

- This is code `krsmahasiswa.php`:

Figure 5.2 Binary Code Krs

- In this coding I made 2 php content (`formmahasiswa.php`) and (`insertkrsmahasiswa.php`) that burguna to create forms and process the input data to the database.
 - In the process the insert data in input on the form will be sent to the (`insertkrsmahasiswa.php`) and in the process when it is successful then the data will be entered into the database and return to the krs.

- This insert the coding in the form of a php object:

```

1 <?php
2 //include class dbconfig
3 require_once("dbconfig.php");
4 $connect = new dbconfig(); // membuat objek dari class dbconfig
5
6 //data kirim dari formdosen.php
7 $nim=$_POST['nim'];
8 $toko=$_POST['toko'];
9 $alamat=$_POST['alamat'];
10 $idDosen=$_POST['idDosen'];
11
12 // seleksi query ke Database
13 $q = $connect->query("INSERT INTO dosen VALUES('$nim','$toko','$alamat','$idDosen')");
14
15 //kembali data display
16 header("location: formdosen.php");
17 ?>

```

Figure 5.3 Binary Code Input Krs

- in this coding I created the look of a lecturer (dosen. php), insert a lecturer (insertdosen. php), lecturer at the input form (formdosen. php)
- I created a php object is (insertdosen. php) and (dbconfig. php) because in the time data display must menagambil lecturer (dbconfig. php) which is useful for data and can be displayed
- *display data Dosen using php object :*

ID	NIM	Toko	Alamat
1	1234567890	ABC Toko	Jl. Pahlawan No. 123

Figure 5.4 Binary Code Dosen

- I created a php object is (*insertdosen. php*) and (*dbconfigr. php*) because in the time data display must menagambil lecturer (*dbconfig. php*) which is useful for data and can be displayed,
- *Then if you want to have to go through the process of filling in the form (*formdosen. php*) and it will be processed into (*insertdosen. php*) and data that will be on the insert would go into a database and can be directly displayed in (*Lecturer. Php*).*
- *display data Dosen input for dosen using php object :*

```

1  <?php
2  //include class dbconfig
3  require_once("dbconfig.php");
4  $connect = new dbconfig(); // membuat objek dari class Koneksi
5
6
7  $namaRuangan=$_POST['namaruangan'];
8  $dosen=$_POST['dosen'];
9
10 // melakukan query ke Database
11 $sq = $connect->query("INSERT INTO tableruang VALUES('','$namaRuangan','$dosen')");
12
13 //back to data display
14 header("Location: dosen.php");
15 ?>

```

Figure 5.5 Binary Code Input Dosen

- in this coding as above only displays the courses (*Matkul. php*) in the form of a php object so that it can display the data in the database by using the (*dbconfig. Php*)

- display data courses using php object :

```
<HTML>
<HEAD>
    <TITLE>
        Data Matakuliah
    </TITLE>
    <link rel="stylesheet" type="text/css" href="css/matakuliah.css" />
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/matakuliah.js"></script>
</HEAD>
<BODY>
<?php
//include template menu sidebar
include_once('header.php');

//include class database
require_once('connect.php');
$connect = new dbconfig(); // membuat object dari class dbconfig
// melakukan query ke database
$qd = $connect->query("SELECT * FROM Mahasiswa ORDER BY NamaMahasiswa ASC");
?>

<?php
echo "Data Mahasiswa Berhasil";
//tambah tabel
echo "<table border='1'>";
echo "<thead>";
echo "<tr> <th>ID</th> <th>Nama Mahasiswa</th> <th>Alamat</th> <th>Telepon</th> <th>Email</th> <th>Aksi</th> </tr> ";
echo "</thead>";
echo "<tbody>";
while($data = $qd->fetch_array(MYSQLI_ASSOC)){
    echo "<tr> <td>" . $data['id'] . "</td> <td>" . $data['namamahasiswa'] . "</td> <td>" . $data['alamat'] . "</td> <td>" . $data['telepon'] . "</td> <td>" . $data['email'] . "</td> <td> <a href='editmatakuliah.php?id=" . $data['id'] . "'>Edit</a> <a href='deletematakuliah.php?id=" . $data['id'] . "'>Delete</a> </td> </tr> ";
}
echo "</tbody>";
echo "</table>";
?>
```

Figure 5.6 Binary Code Courses

- in this coding as above only displays the room (ruangan.php) in the form of a php object so that it can display the data in the database by using the (dbconfig.php)
 - display data Rooms using php object :

```
<HEAD>
<HEAD>
<TITLE> Data Ruangan </TITLE>
<LINK rel="stylesheet" type="text/css" href="ruangan.css" />
<script type="text/javascript" src="ruangan.js" type="text/javascript"></script>
<script type="text/javascript" src="http://code.jquery.com/jquery-1.11.3.min.js" type="text/javascript"></script>
<HEAD>

<BODY>
<?php
//include compiled menu source
include_once("menu.php");
//include class database
require_once("database.php");
$connect = new database();
//inisiasi query ke Database
$qry = $connect->query("SELECT * FROM ruangan ORDER BY id_ruangan ASC");
?>

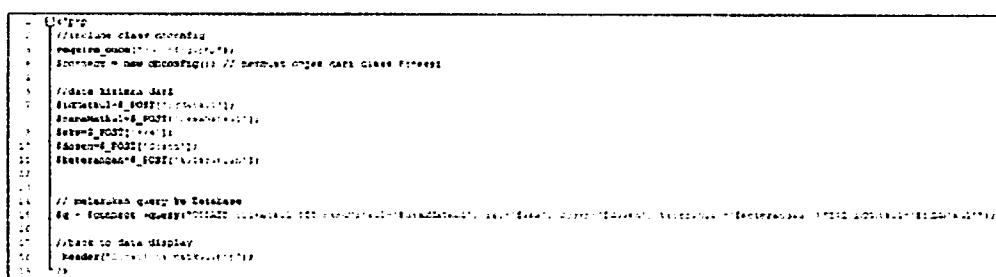
<?php
while($row = $qry->fetch_array())
{
    echo "<tr>";
    echo "<td>" . $row['id_ruangan'] . "</td>";
    echo "<td>" . $row['nama_ruangan'] . "</td>";
    echo "<td>" . $row['jumlah_kursi'] . "</td>";
    echo "<td>" . $row['status'] . "</td>";
    echo "<td>" . $row['ket'] . "</td>";
    echo "<td>";
```

Figure 5.7 Binary Code Rooms

- in this coding I created (updatedosen.php) in the form of a php object that is

useful for updating data when there are errors in the input. in the update process I created a function that takes data that is updated and I made the update data that can be updated only data other than the primary key. When the data have been updated then the data will be new and return to menu (dosen. Php)

- display Update Dosen using php object :



```
1 //<?php
2 //include class database
3 require_once('..\\..\\config.php');
4 $connect = new mysqli($host, $username, $password, $database);
5
6 //data listnya data
7 $nimkulid=$_POST['nimkulid'];
8 $namakulid=$_POST['namakulid'];
9 $tempat_lahir=$_POST['tempat_lahir'];
10 $tempat_kuliah=$_POST['tempat_kuliah'];
11 $statuskualid=$_POST['statuskualid'];
12
13 // melanjutkan query ke Database
14 $q = "UPDATE `tbl_kulid` SET `namakulid`='$namakulid', `tempat_kuliah`='$tempat_kuliah', `statuskualid`='$statuskualid' WHERE `nimkulid`='$nimkulid'";
15
16 //back to data display
17 header("Location: dosen.php");
18
```

Figure 5.8 Binary Code Update Dosen

- in this coding I created (updateruang. php) in the form of a php object that is useful for updating data when there are errors in the input. in the update process ,function that takes data that is updated and I made the update data that can be updated only data other than the primary key. When the data have been updated then the data will be new and return to menu (ruangan. Php)

- display Update room using php object :

```
1 //file.php
2 //include class dbconfig
3 require_once("dbconfig.php");
4 $connect = new Mysqlconfig(); // membuat objek dari class Koneksi
5
6 //data bilangan dari
7 $idkamarangkaian= $_POST['idkamarangkaian'];
8 $namakamarangkaian= $_POST['namakamarangkaian'];
9 $stukturangkaian= $_POST['stukturangkaian'];
10
11
12 // melakukan query ke Database
13 $q = $connect->query("update kamarangkaian set namakamarangkaian='".$namakamarangkaian."', stukturangkaian='".$stukturangkaian."' where idkamarangkaian='".$idkamarangkaian."'");
14
15 //tombol update
16 header("location: index.php");
17
```

Figure 5.9 Binary Code Update Rooms

5.1.3 Step 3 - Algorithms GraphColoring And Welch-powell

- in the coding I made a php object for a useful combination for the formation of the graph so that the data is not broken then I started it by deleting the old data and replace it with the new data because I process automatically.

```
1 <?php
2     error_reporting(E_ALL ^ (E_NOTICE | E_WARNING));
3     function connect()
4     {
5         $conn=mysql_connect('localhost','root','');
6         mysql_select_db('db_jadwal',$conn);
7     }
8     connect();
9     $query7="DELETE FROM graph";
10    $action1=mysql_query($query7);
11    $query8="DELETE FROM graphcoloring";
12    $action1=mysql_query($query8);
13    $query9="DELETE FROM tbljadwal";
14    $action1=mysql_query($query9);
15    /*kombinasi class untuk membuat kombinasi data*/
16    class Combinations implements Iterator
17    {
18        protected $c = null;
19        protected $s = null;
20        protected $n = 0;
21        protected $k = 0;
22        protected $pos = 0;
23
24        function __construct($s, $k)
25        {
26            if(is_array($s))
27            {
28                $this->s = array_values($s);
29                $this->n = count($this->s);
30            }
31            else
32            {
33                $this->s = (string) $s;
34                $this->n = strlen($this->s);
35            }
36            $this->k = $k;
37            $this->rewind();
38        }
39
40        function rewind()
41        {
42            $this->pos = 0;
43        }
44
45        function current()
46        {
47            return $this->s[$this->pos];
48        }
49
50        function key()
51        {
52            return $this->pos;
53        }
54
55        function next()
56        {
57            $this->pos++;
58        }
59
60        function valid()
61        {
62            if($this->pos < $this->n)
63            {
64                return true;
65            }
66            else
67            {
68                return false;
69            }
70        }
71    }
72
73    $obj = new Combinations($s,$k);
74    $obj->rewind();
75    while($obj->valid())
76    {
77        $obj->next();
78        $s = $obj->current();
79        $k = $obj->key();
80        $query="INSERT INTO graph VALUES('$s','$k')";
81        $action=mysql_query($query);
82    }
83
84    $query="SELECT * FROM graph";
85    $action=mysql_query($query);
86    $data=mysql_fetch_array($action);
87    $s=$data['s'];
88    $k=$data['k'];
89
90    $query="SELECT * FROM graphcoloring";
91    $action=mysql_query($query);
92    $data=mysql_fetch_array($action);
93    $s=$data['s'];
94    $k=$data['k'];
95
96    $query="SELECT * FROM tbljadwal";
97    $action=mysql_query($query);
98    $data=mysql_fetch_array($action);
99    $s=$data['s'];
100   $k=$data['k'];
101
102   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
103   $action=mysql_query($query);
104   $data=mysql_fetch_array($action);
105   $s=$data['s'];
106   $k=$data['k'];
107
108   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
109   $action=mysql_query($query);
110   $data=mysql_fetch_array($action);
111   $s=$data['s'];
112   $k=$data['k'];
113
114   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
115   $action=mysql_query($query);
116   $data=mysql_fetch_array($action);
117   $s=$data['s'];
118   $k=$data['k'];
119
120   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
121   $action=mysql_query($query);
122   $data=mysql_fetch_array($action);
123   $s=$data['s'];
124   $k=$data['k'];
125
126   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
127   $action=mysql_query($query);
128   $data=mysql_fetch_array($action);
129   $s=$data['s'];
130   $k=$data['k'];
131
132   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
133   $action=mysql_query($query);
134   $data=mysql_fetch_array($action);
135   $s=$data['s'];
136   $k=$data['k'];
137
138   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
139   $action=mysql_query($query);
140   $data=mysql_fetch_array($action);
141   $s=$data['s'];
142   $k=$data['k'];
143
144   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
145   $action=mysql_query($query);
146   $data=mysql_fetch_array($action);
147   $s=$data['s'];
148   $k=$data['k'];
149
150   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
151   $action=mysql_query($query);
152   $data=mysql_fetch_array($action);
153   $s=$data['s'];
154   $k=$data['k'];
155
156   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
157   $action=mysql_query($query);
158   $data=mysql_fetch_array($action);
159   $s=$data['s'];
160   $k=$data['k'];
161
162   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
163   $action=mysql_query($query);
164   $data=mysql_fetch_array($action);
165   $s=$data['s'];
166   $k=$data['k'];
167
168   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
169   $action=mysql_query($query);
170   $data=mysql_fetch_array($action);
171   $s=$data['s'];
172   $k=$data['k'];
173
174   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
175   $action=mysql_query($query);
176   $data=mysql_fetch_array($action);
177   $s=$data['s'];
178   $k=$data['k'];
179
180   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
181   $action=mysql_query($query);
182   $data=mysql_fetch_array($action);
183   $s=$data['s'];
184   $k=$data['k'];
185
186   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
187   $action=mysql_query($query);
188   $data=mysql_fetch_array($action);
189   $s=$data['s'];
190   $k=$data['k'];
191
192   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
193   $action=mysql_query($query);
194   $data=mysql_fetch_array($action);
195   $s=$data['s'];
196   $k=$data['k'];
197
198   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
199   $action=mysql_query($query);
200   $data=mysql_fetch_array($action);
201   $s=$data['s'];
202   $k=$data['k'];
203
204   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
205   $action=mysql_query($query);
206   $data=mysql_fetch_array($action);
207   $s=$data['s'];
208   $k=$data['k'];
209
210   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
211   $action=mysql_query($query);
212   $data=mysql_fetch_array($action);
213   $s=$data['s'];
214   $k=$data['k'];
215
216   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
217   $action=mysql_query($query);
218   $data=mysql_fetch_array($action);
219   $s=$data['s'];
220   $k=$data['k'];
221
222   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
223   $action=mysql_query($query);
224   $data=mysql_fetch_array($action);
225   $s=$data['s'];
226   $k=$data['k'];
227
228   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
229   $action=mysql_query($query);
230   $data=mysql_fetch_array($action);
231   $s=$data['s'];
232   $k=$data['k'];
233
234   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
235   $action=mysql_query($query);
236   $data=mysql_fetch_array($action);
237   $s=$data['s'];
238   $k=$data['k'];
239
240   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
241   $action=mysql_query($query);
242   $data=mysql_fetch_array($action);
243   $s=$data['s'];
244   $k=$data['k'];
245
246   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
247   $action=mysql_query($query);
248   $data=mysql_fetch_array($action);
249   $s=$data['s'];
250   $k=$data['k'];
251
252   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
253   $action=mysql_query($query);
254   $data=mysql_fetch_array($action);
255   $s=$data['s'];
256   $k=$data['k'];
257
258   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
259   $action=mysql_query($query);
260   $data=mysql_fetch_array($action);
261   $s=$data['s'];
262   $k=$data['k'];
263
264   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
265   $action=mysql_query($query);
266   $data=mysql_fetch_array($action);
267   $s=$data['s'];
268   $k=$data['k'];
269
270   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
271   $action=mysql_query($query);
272   $data=mysql_fetch_array($action);
273   $s=$data['s'];
274   $k=$data['k'];
275
276   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
277   $action=mysql_query($query);
278   $data=mysql_fetch_array($action);
279   $s=$data['s'];
280   $k=$data['k'];
281
282   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
283   $action=mysql_query($query);
284   $data=mysql_fetch_array($action);
285   $s=$data['s'];
286   $k=$data['k'];
287
288   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
289   $action=mysql_query($query);
290   $data=mysql_fetch_array($action);
291   $s=$data['s'];
292   $k=$data['k'];
293
294   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
295   $action=mysql_query($query);
296   $data=mysql_fetch_array($action);
297   $s=$data['s'];
298   $k=$data['k'];
299
300   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
301   $action=mysql_query($query);
302   $data=mysql_fetch_array($action);
303   $s=$data['s'];
304   $k=$data['k'];
305
306   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
307   $action=mysql_query($query);
308   $data=mysql_fetch_array($action);
309   $s=$data['s'];
310   $k=$data['k'];
311
312   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
313   $action=mysql_query($query);
314   $data=mysql_fetch_array($action);
315   $s=$data['s'];
316   $k=$data['k'];
317
318   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
319   $action=mysql_query($query);
320   $data=mysql_fetch_array($action);
321   $s=$data['s'];
322   $k=$data['k'];
323
324   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
325   $action=mysql_query($query);
326   $data=mysql_fetch_array($action);
327   $s=$data['s'];
328   $k=$data['k'];
329
330   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
331   $action=mysql_query($query);
332   $data=mysql_fetch_array($action);
333   $s=$data['s'];
334   $k=$data['k'];
335
336   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
337   $action=mysql_query($query);
338   $data=mysql_fetch_array($action);
339   $s=$data['s'];
340   $k=$data['k'];
341
342   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
343   $action=mysql_query($query);
344   $data=mysql_fetch_array($action);
345   $s=$data['s'];
346   $k=$data['k'];
347
348   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
349   $action=mysql_query($query);
350   $data=mysql_fetch_array($action);
351   $s=$data['s'];
352   $k=$data['k'];
353
354   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
355   $action=mysql_query($query);
356   $data=mysql_fetch_array($action);
357   $s=$data['s'];
358   $k=$data['k'];
359
360   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
361   $action=mysql_query($query);
362   $data=mysql_fetch_array($action);
363   $s=$data['s'];
364   $k=$data['k'];
365
366   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
367   $action=mysql_query($query);
368   $data=mysql_fetch_array($action);
369   $s=$data['s'];
370   $k=$data['k'];
371
372   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
373   $action=mysql_query($query);
374   $data=mysql_fetch_array($action);
375   $s=$data['s'];
376   $k=$data['k'];
377
378   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
379   $action=mysql_query($query);
380   $data=mysql_fetch_array($action);
381   $s=$data['s'];
382   $k=$data['k'];
383
384   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
385   $action=mysql_query($query);
386   $data=mysql_fetch_array($action);
387   $s=$data['s'];
388   $k=$data['k'];
389
390   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
391   $action=mysql_query($query);
392   $data=mysql_fetch_array($action);
393   $s=$data['s'];
394   $k=$data['k'];
395
396   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
397   $action=mysql_query($query);
398   $data=mysql_fetch_array($action);
399   $s=$data['s'];
400   $k=$data['k'];
401
402   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
403   $action=mysql_query($query);
404   $data=mysql_fetch_array($action);
405   $s=$data['s'];
406   $k=$data['k'];
407
408   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
409   $action=mysql_query($query);
410   $data=mysql_fetch_array($action);
411   $s=$data['s'];
412   $k=$data['k'];
413
414   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
415   $action=mysql_query($query);
416   $data=mysql_fetch_array($action);
417   $s=$data['s'];
418   $k=$data['k'];
419
420   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
421   $action=mysql_query($query);
422   $data=mysql_fetch_array($action);
423   $s=$data['s'];
424   $k=$data['k'];
425
426   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
427   $action=mysql_query($query);
428   $data=mysql_fetch_array($action);
429   $s=$data['s'];
430   $k=$data['k'];
431
432   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
433   $action=mysql_query($query);
434   $data=mysql_fetch_array($action);
435   $s=$data['s'];
436   $k=$data['k'];
437
438   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
439   $action=mysql_query($query);
440   $data=mysql_fetch_array($action);
441   $s=$data['s'];
442   $k=$data['k'];
443
444   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
445   $action=mysql_query($query);
446   $data=mysql_fetch_array($action);
447   $s=$data['s'];
448   $k=$data['k'];
449
450   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
451   $action=mysql_query($query);
452   $data=mysql_fetch_array($action);
453   $s=$data['s'];
454   $k=$data['k'];
455
456   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
457   $action=mysql_query($query);
458   $data=mysql_fetch_array($action);
459   $s=$data['s'];
460   $k=$data['k'];
461
462   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
463   $action=mysql_query($query);
464   $data=mysql_fetch_array($action);
465   $s=$data['s'];
466   $k=$data['k'];
467
468   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
469   $action=mysql_query($query);
470   $data=mysql_fetch_array($action);
471   $s=$data['s'];
472   $k=$data['k'];
473
474   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
475   $action=mysql_query($query);
476   $data=mysql_fetch_array($action);
477   $s=$data['s'];
478   $k=$data['k'];
479
480   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
481   $action=mysql_query($query);
482   $data=mysql_fetch_array($action);
483   $s=$data['s'];
484   $k=$data['k'];
485
486   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
487   $action=mysql_query($query);
488   $data=mysql_fetch_array($action);
489   $s=$data['s'];
490   $k=$data['k'];
491
492   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
493   $action=mysql_query($query);
494   $data=mysql_fetch_array($action);
495   $s=$data['s'];
496   $k=$data['k'];
497
498   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
499   $action=mysql_query($query);
500   $data=mysql_fetch_array($action);
501   $s=$data['s'];
502   $k=$data['k'];
503
504   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
505   $action=mysql_query($query);
506   $data=mysql_fetch_array($action);
507   $s=$data['s'];
508   $k=$data['k'];
509
510   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
511   $action=mysql_query($query);
512   $data=mysql_fetch_array($action);
513   $s=$data['s'];
514   $k=$data['k'];
515
516   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
517   $action=mysql_query($query);
518   $data=mysql_fetch_array($action);
519   $s=$data['s'];
520   $k=$data['k'];
521
522   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
523   $action=mysql_query($query);
524   $data=mysql_fetch_array($action);
525   $s=$data['s'];
526   $k=$data['k'];
527
528   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
529   $action=mysql_query($query);
530   $data=mysql_fetch_array($action);
531   $s=$data['s'];
532   $k=$data['k'];
533
534   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
535   $action=mysql_query($query);
536   $data=mysql_fetch_array($action);
537   $s=$data['s'];
538   $k=$data['k'];
539
540   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
541   $action=mysql_query($query);
542   $data=mysql_fetch_array($action);
543   $s=$data['s'];
544   $k=$data['k'];
545
546   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
547   $action=mysql_query($query);
548   $data=mysql_fetch_array($action);
549   $s=$data['s'];
550   $k=$data['k'];
551
552   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
553   $action=mysql_query($query);
554   $data=mysql_fetch_array($action);
555   $s=$data['s'];
556   $k=$data['k'];
557
558   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
559   $action=mysql_query($query);
560   $data=mysql_fetch_array($action);
561   $s=$data['s'];
562   $k=$data['k'];
563
564   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
565   $action=mysql_query($query);
566   $data=mysql_fetch_array($action);
567   $s=$data['s'];
568   $k=$data['k'];
569
570   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
571   $action=mysql_query($query);
572   $data=mysql_fetch_array($action);
573   $s=$data['s'];
574   $k=$data['k'];
575
576   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
577   $action=mysql_query($query);
578   $data=mysql_fetch_array($action);
579   $s=$data['s'];
580   $k=$data['k'];
581
582   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
583   $action=mysql_query($query);
584   $data=mysql_fetch_array($action);
585   $s=$data['s'];
586   $k=$data['k'];
587
588   $query="SELECT * FROM graph WHERE s='$s' AND k='$k'";
589   $action=mysql_query($query);
590   $data=mysql_fetch_array($action);
591   $s=$data['s'];
592   $k=$data['k'];
593
594   $query="SELECT * FROM graphcoloring WHERE s='$s' AND k='$k'";
595   $action=mysql_query($query);
596   $data=mysql_fetch_array($action);
597   $s=$data['s'];
598   $k=$data['k'];
599
599 }
```

Figure 5.10 Binary Code Combination

- In this coding is about the launcher which will call a useful algorithm for the formation of this coding in the exam schedule data in the form of the letter that will be converted into numbers so that the data can

be made into a graphs.

- **Code Launcher using php object :**

```
1<?php
2 function connect()
3 {
4     $conn=mysql_connect("localhost","root","");
5     mysql_select_db("chaitanya",$conn);
6 }
7 connect();
8 require_once "GraphColoring.php";
9
10 function toAlpha($date)
11 {
12     $alphabet = array('A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z');
13     $alpha_flip = array_flip($alphabet);
14
15     if($date <= 12)
16         return $alphabet[$date];
17
18     elseif($date > 12)
19     {
20         $dividend = ($date + 1);
21         $alpha = '';
22         $modulo;
23
24         while ($dividend > 1)
25         {
26             $modulo = ($dividend - 1) % 13;
27             $alpha = $alphabet[$modulo] . $alpha;
28             $dividend = floor($dividend / 13);
29         }
30
31         return $alpha;
32     }
33 }
34
```

Figure 5.11 Binary Code Launcher

- In this coding functions graph coloring algorithm about here which will process the data from the combination that's been changed from numbers and will be processed further.

- **GraphColoring using php object:**

```
1<?php
2 class GraphColoring
3 {
4     private $graph;
5     private $source;
6     private $target;
7     private $sourceType;
8     private $targetType;
9
10     public function __construct()
11     {
12         $this->source = null;
13         $this->target = null;
14         $this->sourceType = null;
15         $this->targetType = null;
16     }
17
18     public function getSource()
19     {
20         return $this->source;
21     }
22
23     public function setSource($source)
24     {
25         $this->source = $source;
26     }
27
28     public function getTarget()
29     {
30         return $this->target;
31     }
32
33     public function setTarget($target)
34     {
35         $this->target = $target;
36     }
37
38     public function getSourceType()
39     {
40         return $this->sourceType;
41     }
42
43     public function setSourceType($sourceType)
44     {
45         $this->sourceType = $sourceType;
46     }
47
48     public function getTargetType()
49     {
50         return $this->targetType;
51     }
52
53     public function setTargetType($targetType)
54     {
55         $this->targetType = $targetType;
56     }
57 }
```

Figure 5.12 Binary Code GraphColoring And Welch-powell

- In part this is a way of coding to make the vertex that connects the graphics so it will be connected and able to check the number of relationships of any chart. in coding this using 2 dimensional array because to make the combination must be through 2 vertices.
 - *GraphColoring* using php object :

```

54
55     Function initColoring()
56     {
57
58         Sthis->vertexNo = Sthis->vertexTotal;
59         Sthis->dbug("No. of Vertices of the Input Graph is %d",Sthis->vertexNo);
60         Sthis->dbug("Graph Representation of the Input Graph is %d");
61         Sthis->dbug("Graph %d",Sthis->graph);
62         Sthis->dbug("Remaining Vertices %d",Sthis->vertexNo-Sthis->vertexNo);
63
64
65         while(Sthis->vertexNo)
66         {
67             Sthis->colorTheGraph();
68             Sthis->recoverTheGraph();
69         }
70
71
72     //EDFA
73
74
75     /**
76      * Colors the graph
77      * @param none
78      * @return none
79      */
80
81     function colorTheGraph()
82     {
83         Sthis->d++; //No. of colors
84
85         for(S1=1,S1<Sthis->vertexTotal;S1++)
86         {
87             for(S=1/S1,S<Sthis->vertexTotal;S++)
88             {
89                 if(Sthis->graph(S1)(S)) --();
90
91                 Sthis->vertexNo = Sthis->vertexNo - 1; //remaining vertices

```

Figure 5.12 Binary Code GraphColoring And Welch-powell :

- In this coding algorithm welch-powell who will process the end all processes so as to establish a schedule of exams based on the algorithm of welch-powell.

- The way this algorithm began changing numerical data into the letter. to be able to turn them into courses and the end result will bring up the exam schedule successfully made
 - *Welch-powell* using php object:

```

179
180     Function toAlpha(data)
181     {
182         var alphabet = "abcdefghijklmnopqrstuvwxyz";
183         var flipAlphabet = "zyxwvutsrqponmlkjihgfedcba";
184         if(data < 0)
185             return flipAlphabet(-data);
186
187         else if(data > 25)
188             return flipAlphabet(25 - data);
189
190         else
191             return alphabet(data);
192     }
193
194
195     Function displayColorResult()
196     {
197         Status->Debug("Displaying color result for %d", this);
198         Status->Debug("Count: %d", count);
199
200         for (var i in this.colors)
201         {
202             Status->Debug("Color: %s", this.colors[i]);
203
204             Status->Debug("  vertex: %d", this.colors[i].vertex);
205             Status->Debug("  face: %d", this.colors[i].face);
206             Status->Debug("  weight: %d", this.colors[i].weight);
207             Status->Debug("  distance: %d", this.colors[i].distance);
208             Status->Debug("  query: %s", this.colors[i].query);
209             Status->Debug("  totalQuery: %s", this.colors[i].totalQuery);
210         }
211     }

```

Figure 5.12 Binary Code GraphColoring And Welch-powell :

- In this process the graph to be marked so as not to occur collisions so that data can be successfully processed perfectly
- *Welch-powell* using php object:

```

function processGraph($j)
{
    for($i=0;$i<=$this->vertexTotal;$i++)
    {
        if($this->graph[$j][$i]==1)
        {
            $this->graph[$j][$i]=0;
            $this->subProcess($i);
        }
    }
}

//ECDn

/**
 * Disconnects a vertex(which is connected to a colored vertex) from its connected ones -- simplex disconnection
 * @param a vertex connected to the newly colored vertex
 * @return none
 */
function subProcess($m)
{
    for($i=0;$i<=$this->vertexTotal;$i++)
    {
        if($this->graph[$m][$i]==1)
        {
            $this->graph[$m][$i]=0;
        }
    }
}

//ECPn

function recoverTheGraph()
{
    for($i=0;$i<=$this->vertexTotal;$i++)
    {
}

```

Figure 5.12 Binary Code GraphColoring And Welch-powell :

- In this coding that takes all the data graphics that will be processed by the welch-powell algorithm so that it can be created and processed further.

- *Welch-powell* using php object

```

function getGraphFromDb($sourceTable) [D:\TA akhir\TA_Aakhir-program\GraphColoring.class.php]
{
    //DB Connection
    //mysql_connect("localhost","root","");
    //mysql_select_db("db_jadwal");
    //include ("dbcconfig.php");

    $q = "SELECT * FROM $sourceTable";
    $res = mysql_query($q);

    // If data found
    if(mysql_num_rows($res))
    {
        while($row = mysql_fetch_array($res))
        {
            $i = trim($row['vertex']);
            $j = trim($row['connected_to']);

            // Initializing the graph - Initializing the connections
            $this->graph[$i][$j] = 1;
            $this->graph[$j][$i] = 1;
            $this->savedGraph[$i][$j] = 1;
            $this->savedGraph[$j][$i] = 1;
        }
    }
    else
    {
        die("Data Invalid OR Does Not Exist..");
    }

    $this->vertexTotal = count($this->graph); //Total vertices

    //Filling up the remaining connections with zero
    if(count($this->graph))
}

```

Figure 5.12 Binary Code GraphColoring And Welch-powell :

- This coding and end result that will process the entire graph is there so. it could be incorporated into the final exam schedule-making process in the course of taking all the vertices that have been processed.

- *Welch-powell* using php object :

```

    $this->vertexTotal = count($this->graph); //Total vertices
    //Filling up the remaining connections with zero
    if(count($this->graph))
    {
        for($i=1; $i <= $this->vertexTotal; $i++)
        {
            for($j=1; $j <= $this->vertexTotal; $j++)
            {
                if($this->graph[$i][$j] != 0)
                {
                    $this->graph[$i][$j] = 0;
                    $this->savedGraph[$i][$j] = 0;
                }
            }
        }
    }
    //EOFn
}

/**
 * Displays/dumps data
 * @param data to be dumped
 * @return none
 */
function dBug($dump)
{
    echo "<PRE>";
    print_r($dump);
    echo "</PRE>";
} //EOFn

//EO Class GraphColoring
?>

```

Figure 5.12 Binary Code GraphColoring And Welch-powell :

5.2 Interface and testing

5.2.1 Main Menu Indeks

After running lampp, open browser, enter localhost/TA_Akhir then if generate it will appear like this :

The following display when users click the button generate then the data will be processed and the exam schedule will appear to be made of data appearing to appear. upon matakuliah, krs taken by students, data room that meets the capacity data, the lecturer will keep testing all of the data that will appear in the table and are in the php home page (index.php)

The screenshot shows a web-based application titled "Exam Schedule Automatic". At the top, there is a navigation bar with links: "Exam Schedule", "MataKuliah", "Ruangan", "Krs Mahasiswa", "Input Krs", "Data Dosen", and "Input Dosen". Below the navigation bar, there is a "Generate" button. The main content area is titled "Exam Schedule" and contains a table with the following data:

No	Semester	Mata Kuliah	Ruangan	Krs Mahasiswa	Tgl	Waktu	Kapasitas	Dosen
287	Semester 2014-08-13	COMPUTER ORGANIZATION	17	08.30-09.30	5.2-B.2			Hirzunius Ledeng,S.Kom.,M.Kom
288	Semester 2014-08-13	APPLIED MATH	25	09.30-11.00	5.1			Aries Ade W.C
293	Semester 2014-08-13	SYSTEM ADMINISTRATION	13	12.30-14.00	5.1			Priyatno, M.Tech
294	Semester 2014-08-13	DATA STRUCTURE	10	13.30-15.00	5.2-A.2			Triyono, M.Tech
292	Semester 2014-08-13	THEORY OF OPERATING SYSTEMS	28	08.30-11.30	5.2-B.2			Eugenia EA, M.Si
295	Semester 2014-08-13	DATA MINING	12	13.30-15.00	5.1			
296	Semester 2014-08-13	COMPUTER INTERFACES	18	12.30-14.00	5.1			Krisk
297	Semester 2014-08-13	PROGRAMMING	10	13.30-15.00	5.1			Priyatno
298	Semester 2014-08-13	GUI PROGRAMMING	6	08.30-11.30	5.1			Priyatno

Figure 5.13 Index

In the display if the user click KrsMahasiswa will bring up the display of the table that contains the numbers, Nim, SKS, Semester, KodeMatakuliah as well as the existing data in the database so that we can see it.

	Eksam Schedule	MataKuliah	Ruangan	Krs Mahasiswa	Input Krs	Data Dosen	Input Dosen
DATA KRS MAHASISWA							
1	06.02.0001	12	2	C5101			
2	06.02.0002	12	2	C5102			
3	06.02.0003	12	2	C5103			
4	06.02.0001	12	2	C5102			
5	06.02.0002	12	2	C5102			
6	06.02.0003	12	2	C5103			
7	06.02.0001	12	2	C5103			
8	06.02.0002	12	2	C5103			
9	06.02.0003	12	2	C5102			
10	06.02.0004	12	2	C5101			
11	06.02.0004	12	2	C5101			
12	06.02.0033	12	2	C5102			
13	06.02.0034	12	2	C5102			
14	06.02.0033	12	2	C5103			
15	06.02.0034	12	2	C5103			
16	06.02.0039	8	2	C5103			
17	06.02.0010	8	2	C5103			
18	06.02.0009	8	2	C5112			
19	06.02.0019	8	2	C5112			
20	06.02.0010	8	2	C5104			
21	06.02.0012	8	2	C5104			
22	06.02.0011	8	2	C5117			
23	06.02.0012	8	2	C5117			
24	06.02.0013	8	3	C5106			
25	06.02.0014	8	2	C5106			
26	06.02.0013	8	2	C5112			
27	06.02.0014	8	2	C5112			
28	06.02.0023	8	2	C5101			
29	06.02.0024	8	2	C5101			

Figure 5.14 Display Krs

In this view contains data that will be used when the test containing.Table:Nomor,NamaRuangan,Kapasitas, Updates to update the data. it used to when creating the test schedule can register a room that is used and the capacity of the room.

	Eksam Schedule	MataKuliah	Ruangan	Krs Mahasiswa	Update Ruangan	Detail Ruangan	Input Ruangan
DATA RUANGAN							
1	0.0.0.0.0	12	2				
2	0.0.0.0.0	12	2				
3	0.0.0.0.0	12	2				

Figure 5.15 Display Rooms

In this view contains courses that have a table KodeMatakuliah, NamaMatakuliah, SKS, lecturer, description, Update to update the data in this table useful. for the making of Professor guards if there is room 5.2-5.3 used there should be teachers. and is also useful for showing data courses that will be used to schedule the exam.

Exam Schedule	Matakuliah	Ruang	Krs Mahasiswa	Input Krs	Data Dosen	Input Dosen
DATA Matakuliah						
	Kode Matakuliah	Nama Matakuliah	SKS	Guru	Courses	Update
CS101	COMPUTER ORGANIZATION	4	Hirwanus Leong,S.Kom.,M.Kom	wajib	Update	
CS102	COMPUTER APPLICATION	4	Hirwanus Leong,S.Kom.,M.Kom	wajib	Update	
CS103	FUNDAMENTAL PROGRAMMING	4	Rosita Herawati, ST.,MT	wajib	Update	
CS104	APPLIED MATH	4	Rosita Herawati, ST.,MT	wajib	Update	
CS105	THEORY OF OPERATING SYSTEM	4	Suryanto EA,Jr.,MSc	wajib	Update	
CS106	DATA STRUCTURES & ALGORITHM	4	Rosita Herawati, ST.,MT	wajib	Update	
CS107	FOUND. OF COMPUTER Sc. I	4	Yulianto Tejo P.,ST.,MT	wajib	Update	
CS108	COMPUTER INTERFACES	4	Rosita Herawati, ST.,MT	wajib	Update	
CS109	SYSTEM ADMINISTRATION	4	Suryanto EA,Jr.,MSc	wajib	Update	
CS110	GUI PROGRAMMING	4	Rosita Herawati, ST.,MT	wajib	Update	
CS111	CLIENT-SERVER COMPUTING	4	Suryanto EA,Jr.,MSc	wajib	Update	
CS112	RELATIONAL DATABASE MANAG	4	Hirwanus Leong,S.Kom.,M.Kom	wajib	Update	
CS113	EMBEDDED SYSTEM	4	Yulianto Tejo P.,ST.,MT	wajib	Update	
CS114	RUNTIME ENVIRONMENT	4	Siehrus Cipto KH.,S.Kom	wajib	Update	
CS115	OBJECT ORIENTED ANALYSIS	4	Rosita Herawati, ST.,MT	wajib	Update	
CS116	NETWORK PROTOCOL	4	Yulianto Tejo P.,ST.,MT	wajib	Update	
CS117	DATABASE ADMINISTRATION	4	Hirwanus Leong,S.Kom.,M.Kom	wajib	Update	
CS118	COMP. INTERFACES AND EMBE	4	Rosita Herawati, ST.,MT	wajib	Update	
CS119	PROJECT MANAGEMENT	4	Rosita Herawati, ST.,MT	wajib	Update	
CS120	WORKSHOP	4	Rosita Herawati, ST.,MT	wajib	Update	
CS121	PRA PROJECT	4	Suryanto EA,Jr.,MSc	wajib	Update	

Figure 5.16 Display Courses

In this view contains data and room guards lecturer lecturers that will keep the exam. in this table contain the nomor, namaruangan, dosen, update to update data, delete to delete data. who useful this table for random data guard while keeping on professors of a room. every day so it could turn out to be a lecturer.

DATA Dosen Ruangan						
No	Nomor	Krs Mahasiswa	MataKuliah	Ruangan	Krs Mahasiswa	Aksi
1	5.1	Agnes Atis M.C			Update	Delete
2	5.1	Ferdinandus Hindarijo, S.Pd			Update	Delete
3	5.1	Berdi			Update	Delete
4	5.1	Pjgi ST			Update	Delete
5	5.4	Agus			Update	Delete
6	5.4	Badi			Update	Delete
7	5.4	Aji			Update	Delete
8	5.4	Frank			Update	Delete
9	8.1	Angges			Update	Delete
10	8.1	Anggun			Update	Delete
11	8.1	Minggo			Update	Delete

localhost/TA_Akhir/deleteDosen.php?iddosen=ruangan=4

Figure 5.17 Display Dosen

In this view is useful only for input lecturer guards. When the user click the input lecturer then go into (formdosen. php) that would be useful for input data that you want to input.

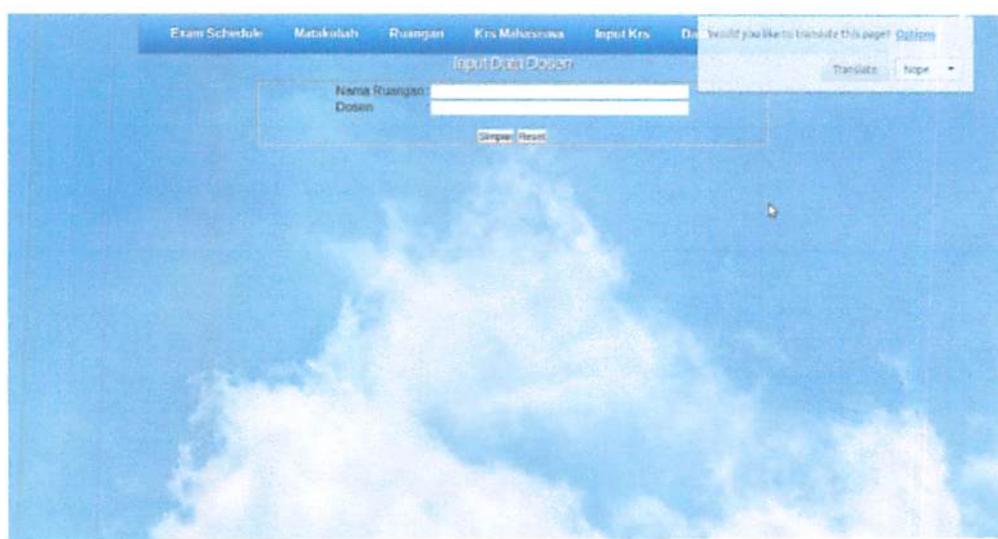


Figure 5.18 Display Dosen Input

5.2.2 Implementasi in human language

in human language is a form of graphic images that are taken in the manufacture of the test schedule that is processed from a krs. so it formed a useful graph to simplify the program makers in designing programs that will be made in the process of making the test schedule.

This is a form of graph based on data krs:

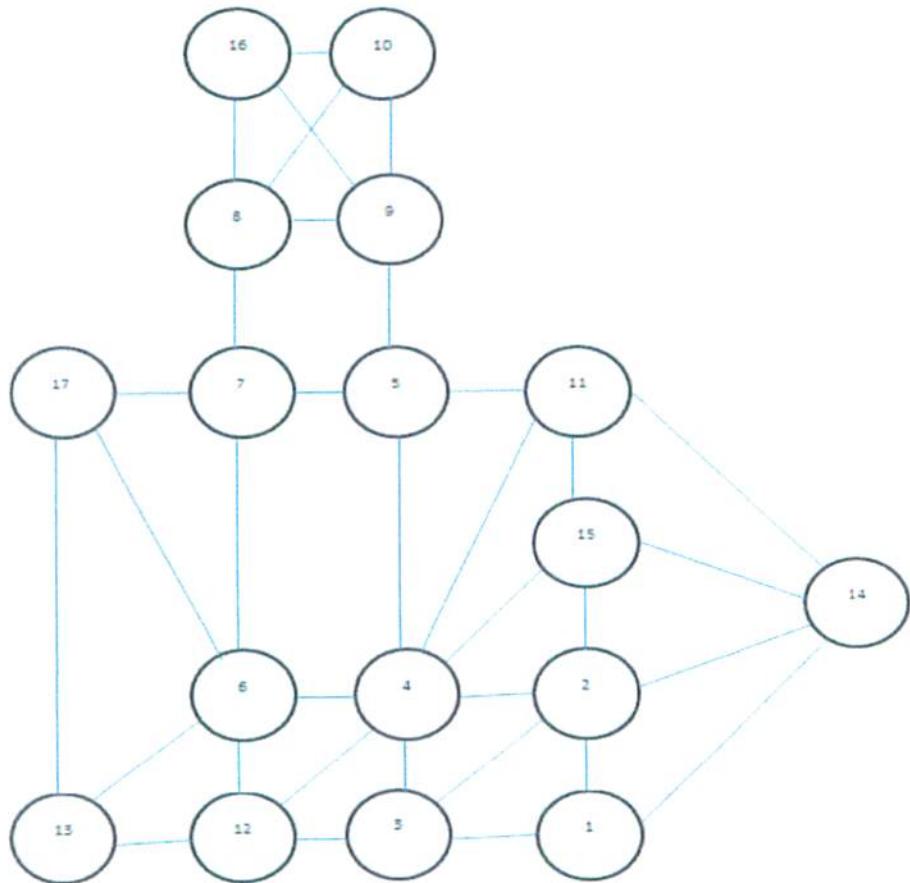


Figure 5.19 Display Graph in Human Language

This is the shape of the graph is formed in the human language when algorithm Welch-powell works on the run :

- Sort of G-node node in degrees downward. The sequence obtained may not be unique due to the number of nodes that have the same degree.
- Use a color to dye the first node that has the highest degree, and the other node-node (in sorted order) that is not the first neighbouring nodes.
- Start again by giving a second color on the next highest degree nodes in sorted list and have not been colored.
- Repeat the giving of different colors on the next-node until all nodes have been colored.

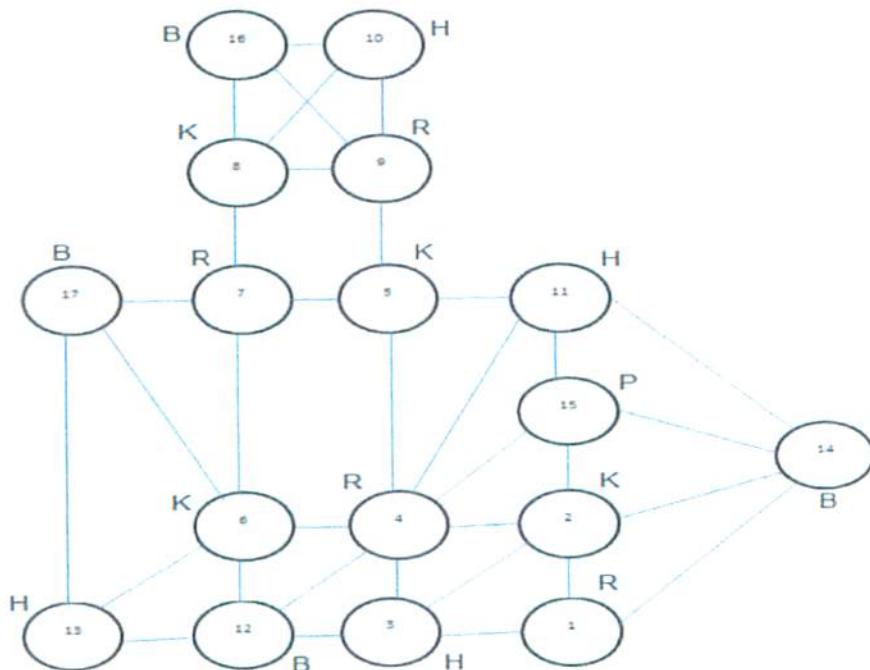


Figure 5.20 Display Welch-Powell And GraphColoring in Human Language

The first color in the given graph is red. It is formed based on courses that have the greatest degree. which contains the numbers: 1, 4, 7, 9

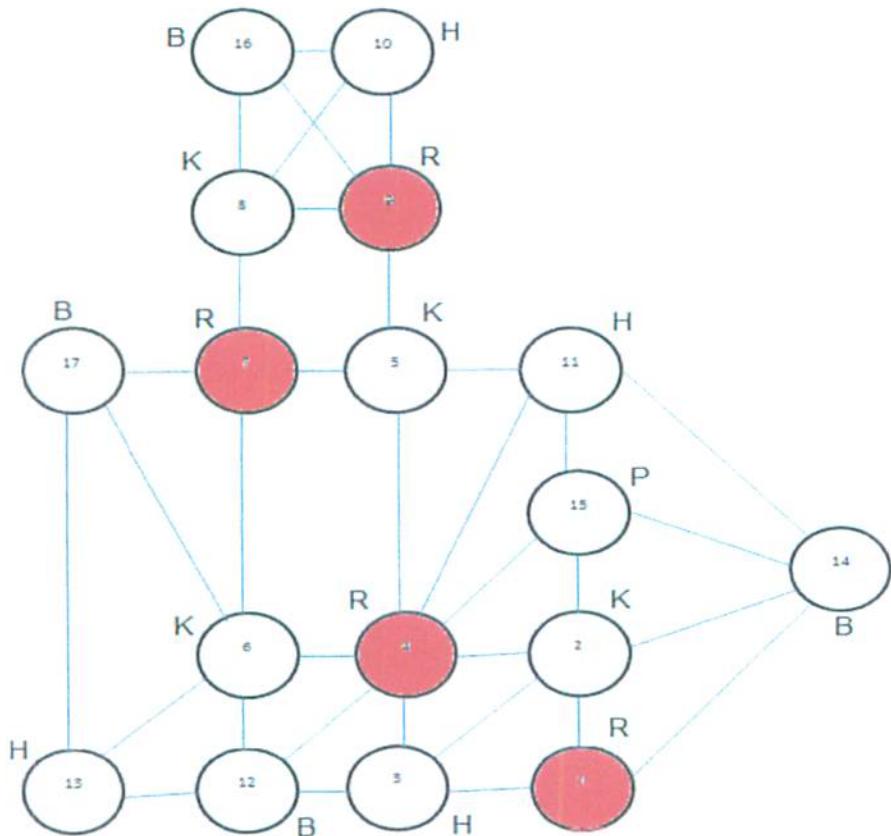


Figure 5.20 Display Welch-Powell And GraphColoring in Human Language

The next largest degree will be given color yellow that contains numbers: 2, 5, 6, 8 . Because the greatest degree of investigation by the number who have furtherdegrees.

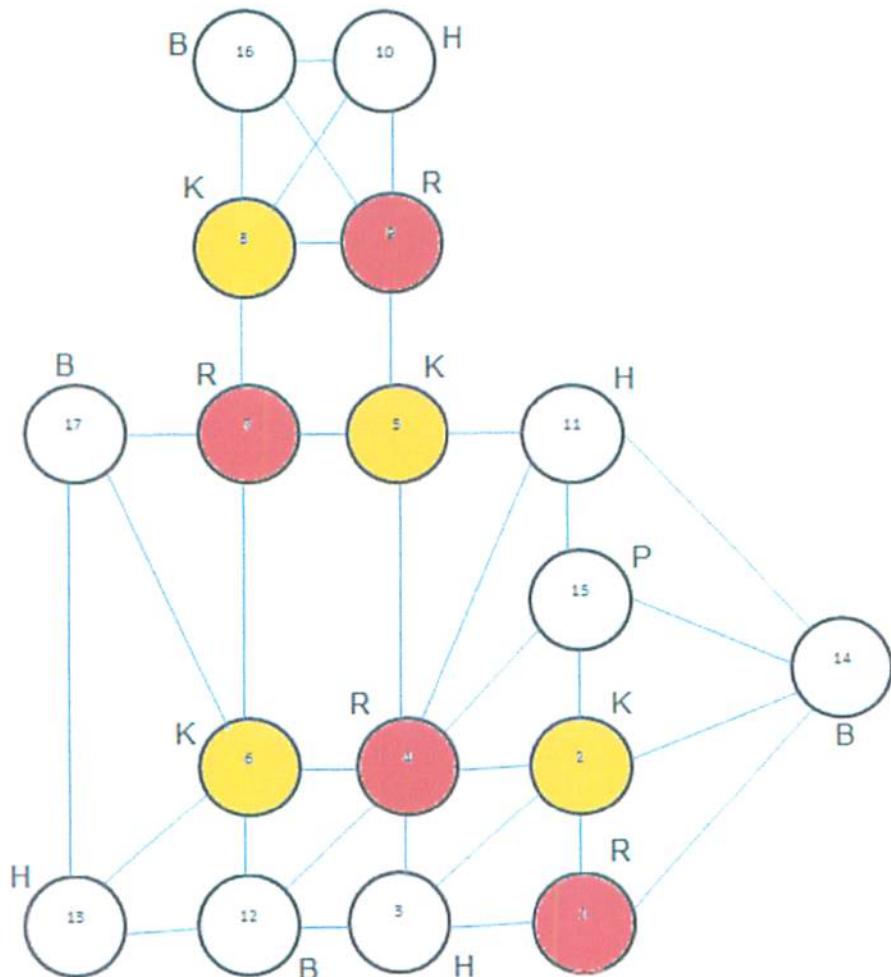


Figure 5.20 Display Welch-Powell And GraphColoring in Human Language

The next largest degree will be given the color green that contains number: 3,10,11,13 . Because the greatest degree of investigation by the number who have furtherdegrees.

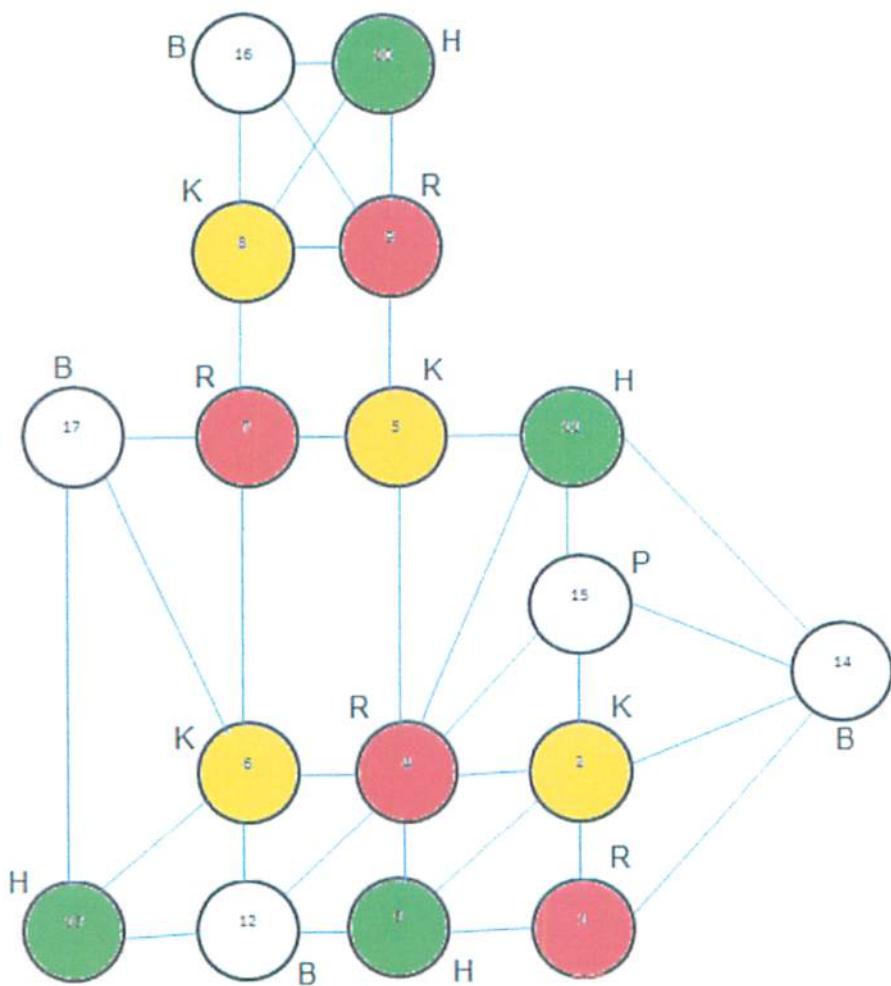


Figure 5.20 Display Welch-Powell And GraphColoring in Human Language

The next largest degree are given the color blue which contains number: 12,14,16,17. Because the greatest degree of investigation by the number who have furtherdegrees.

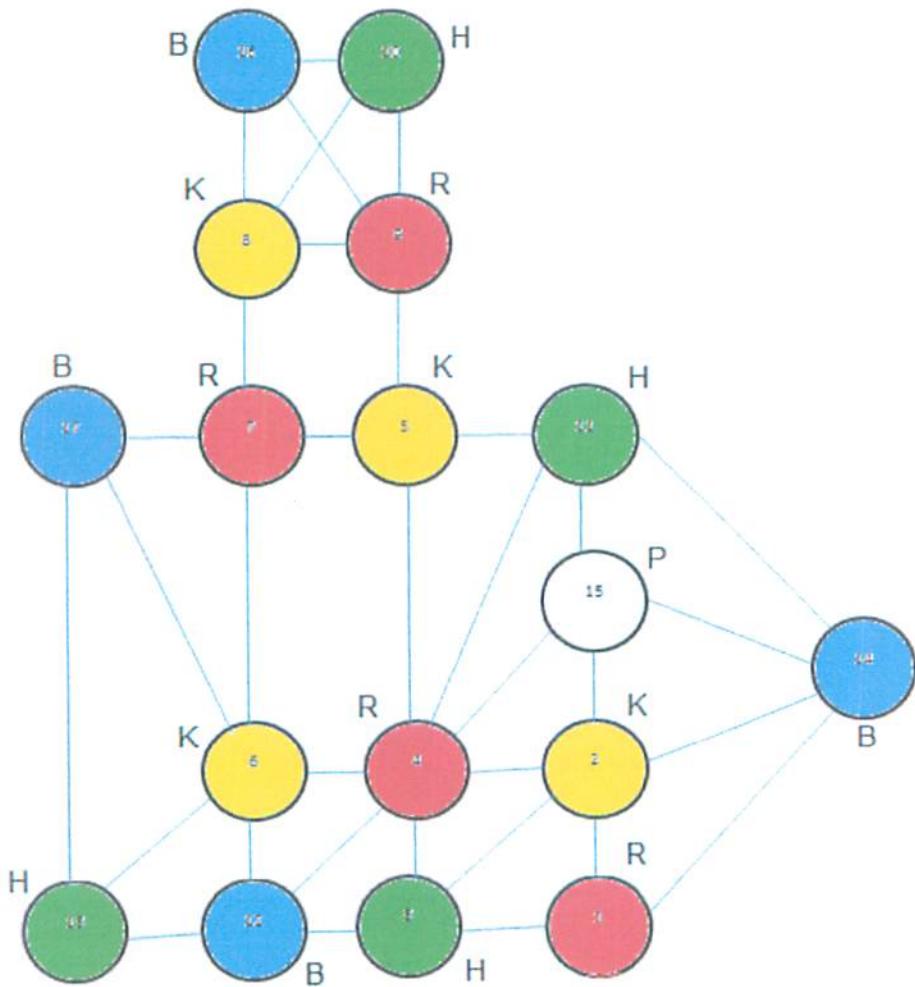


Figure 5.20 Display Welch-Powell And GraphColoring in Human Language

And Last Color is Purple which contains number: 15 so all the graph already exists in colour and there are no similar colors which are connected.

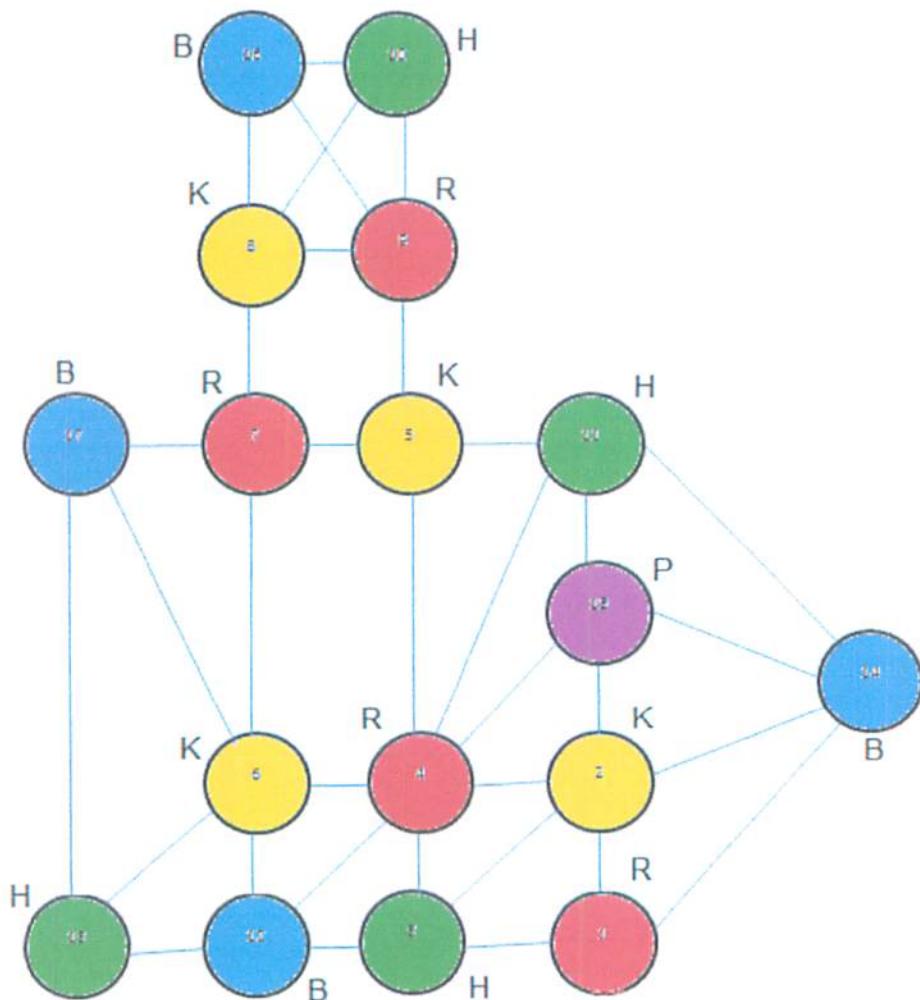


Figure 5.20 Display Welch-Powell And GraphColoring in Human Language