

BAB IV

IMPLEMENTASI

Bab ini merupakan hasil implementasi sistem pendukung keputusan penentu jaminan mutu kemahasiswaan dan alumni dengan menggunakan metode *Analytic Network Process* (ANP) dalam bentuk program aplikasi dan penjelasan mengenai setiap fungsi halaman.

4.1. Lingkungan Implementasi

Lingkungan implementasi terdiri dari lingkungan perangkat keras dan perangkat lunak. Berikut ini penjelasan mengenai lingkungan perangkat keras dan perangkat lunak :

4.1.1. Lingkungan Perangkat Keras

Perangkat keras yang digunakan sebagai berikut:

1. Graphics AMD Radeon HD 6370M.
2. Processor Intel Core i5 2410M Speed 2.3 GHz L3 Cache 3MB.
3. Memori DDR3 4GB.
4. Hard disk 640GB.
5. Monitor 14".
6. Keyboard.
7. Mouse.

4.1.2. Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem pendukung keputusan penentu jaminan mutu kemahasiswaan dan alumni adalah sebagai berikut:

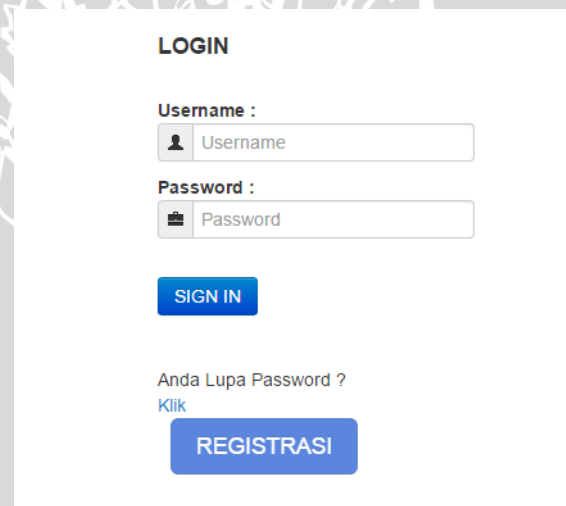
1. Sistem operasi windows 8 Ultimates 64bit.
2. XAMPP 1.8.1.
3. PHPDesigner8.
4. Google Chrome.

4.2. Implementasi Antar Muka

Setelah program aplikasi dibuat, tahap selanjutnya yaitu menjalankan aplikasi pada *browser*. *User* pada aplikasi sistem pendukung keputusan penentu jaminan mutu kemahasiswaan dan alumninya satu yaitu *user* yang dapat melakukan *login* adalah *user* yang sudah melakukan registrasi. Berikut ini adalah penjelasan mengenai aplikasi sistem pendukung keputusan penentu jaminan mutu kemahasiswaan dan alumni.

4.2.1. Halaman *Login*

Halaman *login* akan muncul ketika *user* berhasil mengakses aplikasi melalui *browser*. *User* melakukan *login* terlebih dahulu dengan mengisi *username* dan *password* untuk dapat mengakses sistem. Bila proses *login* gagal maka akan muncul pesan gagal dan bila proses *login* berhasil maka akan tampil menu *Home* dan deskripsi singkat mengenai sistem. Halaman *login* ditunjukkan pada gambar 4.1.



The image shows a login form with the following elements:

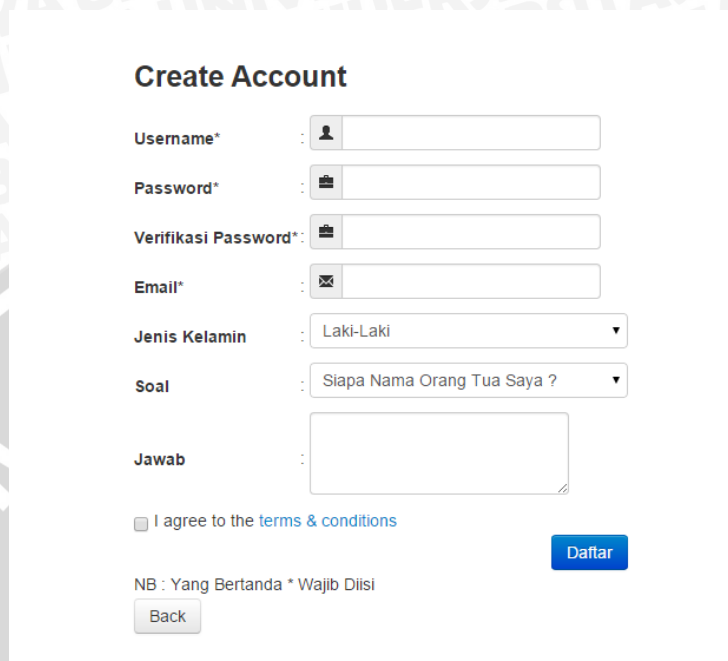
- LOGIN** (Section Header)
- Username :** Input field with a person icon and the text "Username".
- Password :** Input field with a key icon and the text "Password".
- SIGN IN** (Blue button)
- Anda Lupa Password ?** (Text)
- Klik** (Text)
- REGISTRASI** (Blue button)

Gambar 4.1 Halaman *Login*

4.2.2 Halaman Registrasi

Halaman registrasi digunakan bila *user* belum pernah mengakses aplikasi dan harus melakukan registrasi terlebih dahulu dengan cara menekan tombol registrasi pada halaman *login*. Pada halaman registrasi, data yang harus diisi yaitu *username*, *password*, verifikasi *password*, *email*, jenis kelamin, soal dan jawaban. Setelah seluruh data terisi maka tekan

tombol daftar untuk membuat akun baru. Lakukan *login* bila sudah memiliki *username* dan *password* pada halaman *login* seperti pada gambar 4.2 Halaman Registrasi.



Create Account

Username* :

Password* :

Verifikasi Password* :

Email* :

Jenis Kelamin : Laki-Laki ▼

Soal : Siapa Nama Orang Tua Saya ? ▼

Jawab :

I agree to the [terms & conditions](#)

NB : Yang Bertanda * Wajib Diisi

Gambar 4.2 Halaman Registrasi

4.2.3 Halaman Home

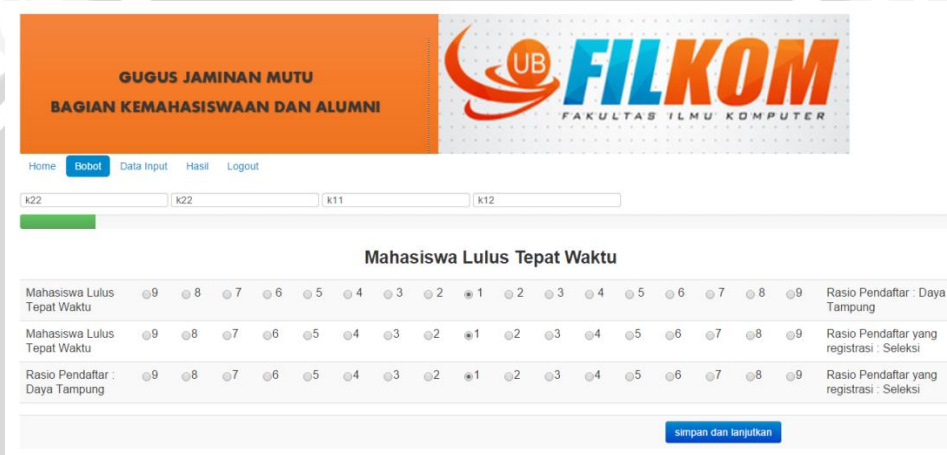
Halaman *Home* dapat dilihat bila *user* telah melakukan *login* dengan memasukkan *username* dan *password*. Halaman *Home* berisi deskripsi singkat mengenai penjaminan mutu kemahasiswaan dan alumni seperti pada gambar 4.3. Halaman *Home*.



Gambar 4.3 Halaman Home

4.2.4 Halaman Bobot

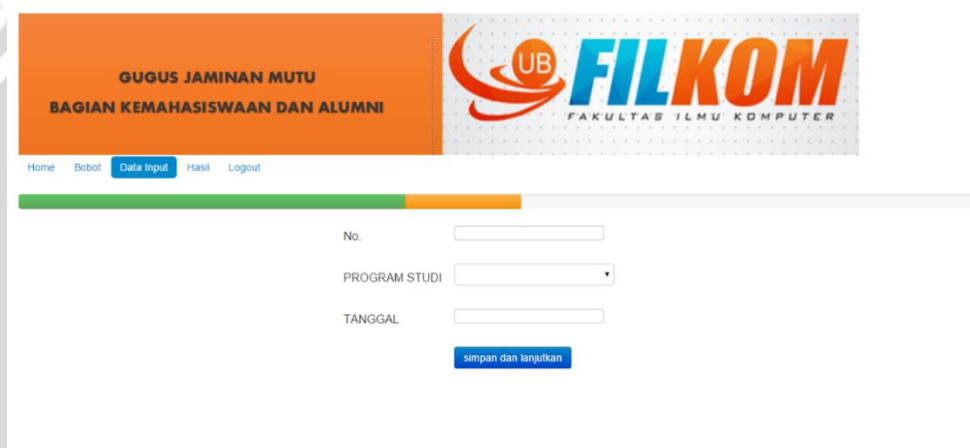
Halaman bobot berisi data perbandingan berpasangan setiap sub-kriteria yang saling berhubungan satu dengan yang lainnya. Ada sebanyak 20 halaman data perbandingan berpasangan yang harus diisi dengan nilai antara 1-9, setelah data-data diisi maka aplikasi akan melakukan prose perhitungan dengan menggunakan metode ANP untuk mendapatkan nilai bobot untuk setiap sub-kriteria seperti pada gambar 4.4. halaman bobot.



Gambar 4.4. Halaman Bobot

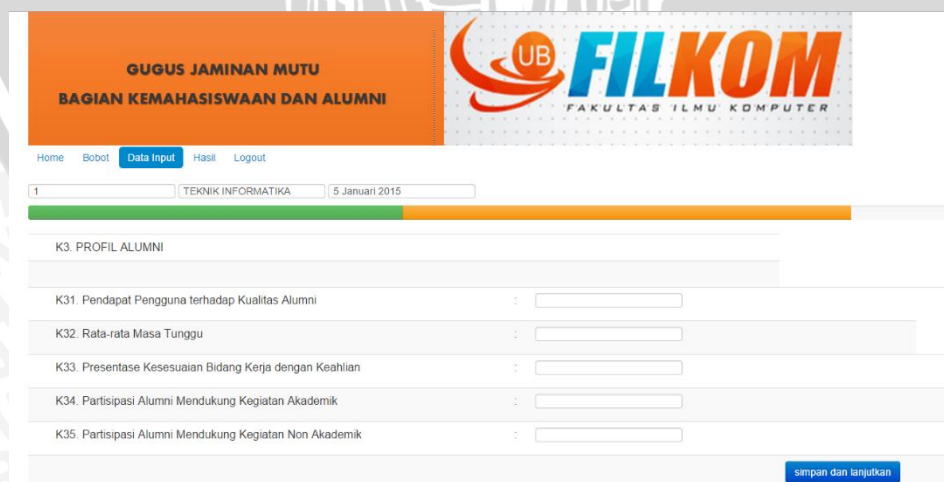
4.2.5 Halaman Data Input

Halaman data input merupakan halaman yang berisi nomor, program studi dan tanggal, hal ini dilakukan untuk mengetahui program studi mana dan tindak lanjut apakah yang sesuai dengan permasalahan terhadap penjaminan mutu kemahasiswaan dan alumni. Setelah itu *user* memasukkan data input yang sesuai dengan buku borang IIIB untuk setiap kriteria-kriteria yang telah tersedia pada aplikasi seperti pada gambar 4.5 Halaman Data Input dan gambar 4.6. Kriteria 3. Profil Alumni.



Gambar 4.5 Halaman Data Input

4.2.6 Halaman K3. Profil Alumni



Gambar 4.6 Halaman K3. Profil Alumni

4.2.7 Halaman Hasil

Halaman hasil merupakan halaman yang berisi mengenai hasil dari proses perhitungan dengan menggunakan ANP dimana data perbandingan berpasangan setiap sub-kriteria yang saling berhubungan satu dengan yang lainnya akan menghasilkan nilai bobot seluruh sub-kriteria dikali dengan data input yang sesuai dengan buku borang IIIB/data input baru mengenai kemahasiswaan dan alumni sehingga menghasilkan alternatif/tindak lanjut apa yang harus dilakukan untuk memperbaiki mutu kemahasiswaan dan alumni yang ada di program studi tersebut seperti pada gambar 4.7 Halaman Hasil.



No.	PROGRAM STUDI	TANGGAL	HASIL
1	TEKNIK INFORMATIKA	5 Januari 2015	Promosi/Branding

Gambar 4.7 Halaman Hasil

4.3. Proses Implementasi sesuai pada Perancangan

Proses implementasi adalah proses-proses yang terkait dengan metode ANP di dalam aplikasi yang sesuai dengan perancangan yang telah dibuat. Proses-proses tersebut diantaranya :

4.3.1 Proses Matriks Perbandingan Berpasangan

Proses ini merupakan proses awal setelah user memasukkan nilai prioritas perbandingan berpasangan pada form kriteria. Nilai prioritas perbandingan berpasangan sesuai dengan tingkat kepentingan antar kriteria yang berhubungan terdapat pada Tabel 4.1.

Tabel 4.1 *Source Code* Proses Matriks Perbandingan Berpasangan

```

1 //perbandingan beban dosen : mahasiswa dengan rata-rata IPK
2 for($i=1;$i<=9;$i++){
3     if($nilai1=="kd$i"){
4         $bdosen1=10-$i;
5         $rtipk=1;
6     }
7 }
8 $j=1;
9 for($i=9;$i<=17;$i++){
10    if($nilai1=="kd$i"){
11        $bdosen1=1;
12        $rtipk=0+$j;
13    }
14    $j=$j+1;
15 }
16
17 <tr>
18 <td>Beban Dosen : Mahasiswa </td>
19 <td><?php echo $b1c1= $bdosen1/$bdosen1;?></td>
20 <td><?php echo $b1c2= $bdosen1/$rtipk;?></td>
21 <td><?php echo $b1c3= $bdosen2/$mhs lulus ?></td>
22 <td><?php echo $b1c4= $bdosen3/$pres akademik ?></td>
23 <td><?php echo $b1c5= $bdosen4/$pres non akademik ?></td>
24 </tr>
25
26

```

Pada baris pertama ditunjukkan untuk mengetahui bahwa proses yang akan dilakukan adalah proses perbandingan berpasangan antara kriteria beban dosen dan rata-rata mahasiswa. Baris ke-2 hingga ke-7 menjelaskan mengenai perulangan nilai yang dimasukkan yaitu berasal dari sebelah kiri / nilai 9 hingga nilai 1 maka jika *user* menyimpan nilai tersebut maka akan memberitahu inisialisasi nilai mana yang tersimpan yang nantinya berguna untuk proses matriks perbandingan kriteria beban dosen. Baris ke-8 hingga baris ke-14 sama seperti proses baris kedua hingga kelima namun hasil yang akan didapat nanti adalah nilai kebalikan. Baris-17 menjelaskan mengenai proses perbandingan berpasangan

pada kriteria beban dosen : mahasiswa dengan membagi nilai yang berada di baris dan kolom yang sama.

4.3.2. Proses Eigen Vectordan Nilai Consistency Ratio (CR)

Ketika *user* selesai melakukan pemilihan perbandingan berpasangan di setiap node maka inputan langsung di proses untuk mendapatkan *eigen vector*. Setelah itu di simpan dalam *database*. Sebelum mendapatkan *eigen Vector*, terlebih dahulu melakukan proses normalisasi untuk node beban dosen : mahasiswa dapat dilihat pada Tabel 4.2.

Tabel 4.2 Source Code proses normalisasi node beban dosen : mahasiswa

1	<table>
2	<tr>
3	<td>Beban Dosen : Mahasiswa</td>
4	<td><?php echo \$nb1c1= \$b1c1/\$b6c1;?></td>
5	<td><?php echo \$nb1c2=\$b1c2/\$b6c2;?></td>
6	<td><?php echo \$nb1c3=\$b1c3/\$b6c3;?></td>
7	<td><?php echo \$nb1c4=\$b1c4/\$b6c4;?></td>
8	<td><?php echo \$nb1c5=\$b1c5/\$b6c5; ?></td>
9	<td><?php echo
10	\$nb1c6=\$nb1c1+\$nb1c2+\$nb1c3+\$nb1c4+\$nb1c5?></td>
	</tr>

Pada baris ketiga menjelaskan bahwa perhitungan normalisasi yang akan dibuat adalah perhitungan dalam kolom beban dosen : mahasiswa. Baris keempat hingga kedelapan menjelaskan mengenai cara mendapatkan normalisasi yaitu pertama dengan menormalisasi seluruh nilai dimana kolom nilai beban dosen : mahasiswa terhadap beban dosen : mahasiswa yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom beban dosen : mahasiswa, kolom nilai rata-rata IPK terhadap beban dosen : mahasiswa yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom beban dosen : mahasiswa begitu pula dengan kolom mahasiswa lulus tepat waktu, prestasi bidang akademik, dan prestasi non akademik yang berada pada kolom prestasi bidang akademik. Hasilnya berupa nilai yang sudah di normalisasi. Baris kesembilan merupakan jumlah dari seluruh nilai normalisasi dalam satu kolom yaitu beban dosen : mahasiswa.



Proses Normalisasi untuk node rata-rata IPK dapat dilihat pada Tabel 4.3.

Tabel 4.3 Source Code proses normalisasi node rata-rata IPK

1	<table>
2	<tr>
3	<td>Rata-rata IPK</td>
4	<td><?php echo \$nb2c1= \$b2c1/\$b6c1;?></td>
5	<td><?php echo \$nb2c2=\$b2c2/\$b6c2;?></td>
6	<td><?php echo \$nb2c3=\$b2c3/\$b6c3;?></td>
7	<td><?php echo \$nb2c4=\$b2c4/\$b6c4;?></td>
8	<td><?php echo \$nb2c5=\$b2c5/\$b6c5; ?></td>
9	<td><?php echo\$nb2c6=\$nb2c1+\$nb2c2+\$nb2c3+\$nb2c4+\$nb2c5?></td>
10	</tr>

Pada baris ketiga menjelaskan bahwa perhitungan normalisasi yang akan dibuat adalah perhitungan dalam kolom rata-rata IPK. Baris keempat hingga kedelapan menjelaskan mengenai cara mendapatkan normalisasi yaitu pertama dengan menormalisasi seluruh nilai dimana kolom nilai beban dosen : mahasiswa terhadap rata-rata IPK yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom rata-rata IPK, kolom nilai rata-rata IPK terhadap rata-rata IPK yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom rata-rata IPK begitu pula dengan kolom mahasiswa lulus tepat waktu, prestasi bidang akademik, dan prestasi non akademik yang berada pada kolom prestasi bidang akademik. Hasilnya berupa nilai yang sudah di normalisasi. Baris kesembilan merupakan jumlah dari seluruh nilai normalisasi dalam satu kolom yaitu mahasiswa lulus tepat waktu.

Proses normalisasi untuk node mahasiswa lulus tepat waktu dapat dilihat pada Tabel 4.4.

Tabel 4.4 Source Code proses normalisasi node mahasiswa lulus tepat waktu

1	<table>
2	<tr>
3	<td>Mahasiswa Lulus Tepat Waktu</td>
4	<td><?php echo \$nb3c1= \$b3c1/\$b6c1;?></td>
5	<td><?php echo \$nb3c2=\$b3c2/\$b6c2;?></td>
6	<td><?php echo \$nb3c3=\$b3c3/\$b6c3;?></td>
7	<td><?php echo \$nb3c4=\$b3c4/\$b6c4;?></td>
8	<td><?php echo \$nb3c5=\$b3c5/\$b6c5; ?></td>
9	<td><?php echo \$nb3c6=\$nb3c1+\$nb3c2+\$nb3c3+\$nb3c4+\$nb3c5?></td>
10	</tr>

Pada baris ketiga menjelaskan bahwa perhitungan normalisasi yang akan dibuat adalah perhitungan dalam kolom mahasiswa lulus tepat waktu. Baris keempat hingga kedelapan menjelaskan mengenai cara mendapatkan normalisasi yaitu pertama dengan menormalisasi seluruh nilai dimana kolom nilai beban dosen : mahasiswa terhadap mahasiswa lulus tepat waktu yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom mahasiswa lulus tepat waktu, kolom nilai rata-rata IPK terhadap mahasiswa lulus tepat waktu yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom rata-rata IPK begitu pula dengan kolom mahasiswa lulus tepat waktu, prestasi bidang akademik, dan prestasi non akademik yang berada pada kolom prestasi bidang akademik. Hasilnya berupa nilai yang sudah di normalisasi. Baris kesembilan merupakan jumlah dari seluruh nilai normalisasi dalam satu kolom yaitu mahasiswa lulus tepat waktu.

Proses normalisasi untuk node prestasi bidang akademik dapat dilihat pada Tabel 4.5.

Tabel 4.5 Source Code proses normalisasi node prestasi bidang akademik

1	<table>
2	<tr>
3	<td>Prestasi Bidang Akademik</td>
4	<td><?php echo \$nb4c1= \$b4c1/\$b6c1;?></td>
5	<td><?php echo \$nb4c2=\$b4c2/\$b6c2;?></td>
6	<td><?php echo \$nb4c3=\$b4c3/\$b6c3;?></td>
7	<td><?php echo \$nb4c4=\$b4c4/\$b6c4;?></td>
8	<td><?php echo \$nb4c5=\$b4c5/\$b6c5; ?></td>
9	<td><?php echo \$nb4c6=\$nb4c1+\$nb4c2+\$nb4c3+\$nb4c4+\$nb4c5?></td>
10	</tr>

Pada baris ketiga menjelaskan bahwa perhitungan normalisasi yang akan dibuat adalah perhitungan dalam kolom prestasi bidang akademik. Baris keempat hingga kedelapan menjelaskan mengenai cara mendapatkan normalisasi yaitu pertama dengan menormalisasi seluruh nilai dimana kolom nilai beban dosen : mahasiswa terhadap prestasi bidang akademik yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom beban dosen : mahasiswa, kolom nilai rata-rata IPK terhadap prestasi bidang akademik yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom rata-rata IPK begitu pula dengan kolom mahasiswa lulus tepat waktu, prestasi bidang akademik, dan prestasi non akademik yang berada pada kolom prestasi bidang akademik. Hasilnya berupa nilai yang sudah di normalisasi. Baris kesembilan merupakan jumlah dari seluruh nilai normalisasi dalam satu kolom yaitu prestasi bidang akademik.

Proses normalisasi untuk node prestasi bidang non akademik dapat dilihat pada Tabel 4.6.

Tabel 4.6 Source Code proses normalisasi node prestasi bidang non akademik

1	<table>
2	<tr>
3	<td>Prestasi Bidang Non Akademik</td>
4	<td><?php echo \$nb5c1=\$b5c1/\$b6c1;?></td>
5	<td><?php echo \$nb5c2=\$b5c2/\$b6c2;?></td>
6	<td><?php echo \$nb5c3=\$b5c3/\$b6c3;?></td>
7	<td><?php echo \$nb5c4=\$b5c4/\$b6c4;?></td>
8	<td><?php echo \$nb5c5=\$b5c5/\$b6c5; ?></td>
9	<td><?php echo \$nb5c6=\$nb5c1+\$nb5c2+\$nb5c3+\$nb5c4+\$nb5c5?></td>
10	</tr>

Pada baris ketiga menjelaskan bahwa perhitungan normalisasi yang akan dibuat adalah perhitungan dalam kolom prestasi bidang non akademik. Baris keempat hingga kedelapan menjelaskan mengenai cara mendapatkan normalisasi yaitu pertama dengan menormalisasi seluruh nilai dimana nilai kolom beban dosen : mahasiswa terhadap prestasi bidang non akademik yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom beban dosen : mahasiswa, kolom nilai rata-rata IPK terhadap prestasi bidang non akademik yang diinputkan dibagi dengan nilai jumlah keseluruhan kolom rata-rata IPK, begitu pula dengan kolom mahasiswa lulus tepat waktu, prestasi bidang non akademik, dan prestasi non akademik yang berada pada kolom prestasi bidang akademik. Hasilnya berupa nilai yang sudah di normalisasi. Baris kesembilan merupakan jumlah dari seluruh nilai normalisasi dalam satu kolom yaitu prestasi bidang non akademik.

Setelah didapatkan semua nilai normalisasi tiap kolom dan sudah dijumlahkan maka akan dilakukan penjumlahan tiap kolom sehingga akan menghasilkan nilai *eigen vector* yang dapat dilihat pada Tabel 4.7

Tabel 4.7 Source Code proses penjumlahan nilai normalisasi yang nantinya akan menghasilkan *eigen vector*

1	<tr>
2	<td>Jumlah</td>
3	<td><?php echo \$nb6c1=\$nb1c1+\$nb2c1+\$nb3c1+\$nb4c1+\$nb5c1?></td>
4	<td><?php echo \$nb6c2=\$nb1c2+\$nb2c2+\$nb3c2+\$nb4c2+\$nb5c2?></td>
5	<td><?php echo \$nb6c3=\$nb1c3+\$nb2c3+\$nb3c3+\$nb4c3+\$nb5c3?></td>
6	<td><?php echo \$nb6c4=\$nb1c4+\$nb2c4+\$nb3c4+\$nb4c4+\$nb5c4?></td>
7	<td><?php echo \$nb6c5=\$nb1c5+\$nb2c5+\$nb3c5+\$nb4c5+\$nb5c5?></td>
8	<td><?php echo \$nb6c6=\$nb1c6+\$nb2c6+\$nb3c6+\$nb4c6+\$nb5c6?></td>
9	</tr>
10	</table>
11	<?php
12	echo \$nb1c7=\$nb1c6/\$nb6c6." ";
13	echo \$nb2c7=\$nb2c6/\$nb6c6." ";
14	echo \$nb3c7=\$nb3c6/\$nb6c6." ";
15	echo \$nb4c7=\$nb4c6/\$nb6c6." ";
16	echo \$nb5c7=\$nb5c6/\$nb6c6." ";

Pada baris kedua menjelaskan proses yang ingin dilakukan yaitu dengan menjumlahkan nilai seluruh kolom untuk mendapat nilai *eigen* maksimum. Baris ketiga hingga baris ketujuh menjelaskan mengenai penjumlahan nilai-nilai normalisasi yang sudah didapat dari beban dosen : mahasiswa, rata-rata IPK, mahasiswa lulus tepat waktu, prestasi akademik dan prestasi non akademik yang selanjutnya dapat digunakan untuk mendapatkan nilai *eigen vector* dan nilai *eigen* maksimum pada baris kedelapan. Pada baris ke-12 hingga ke-16 menjelaskan mengenai jumlah keseluruhan nilai node yang berasal dari kolom-kolom yang sudah dinormalisasi dibagi dengan nilai *eigen* maksimum/nilai dari seluruh kolom yang sudah dinormalisasi.

Setelah selesai semua proses diatas maka dilakukan pengecekan *Consistency Ratio* (CR) yang nantinya apabila nilai lebih besardari 0,1 maka program akan mengarahkan ke halaman form matriks perbandingan berpasangan dan apabila kurang dari dari 0,1 maka program akan melakukan proses penyimpanan eigen yang didapatkan dan mengarahkan ke form berikutnya. Setelah didapatkan dilihat pada Tabel 4.8.

Tabel 4.8 Source Code proses Consistency Ratio (CR)

```

1 <?php
2 echo$nb1c7=$nb1c6/$nb6c6."<br />";
3 echo $nb2c7=$nb2c6/$nb6c6."<br />";
4 echo $nb3c7=$nb3c6/$nb6c6."<br />";
5 echo $nb4c7=$nb4c6/$nb6c6."<br />";
6 echo $nb5c7=$nb5c6/$nb6c6."<br />";
7
8 //menghitung cr note: cek rumus
9 $ci=($nb6c6-5)/(5-1);
10 $cr=$ci/1.12;
11
12 //jika lebih besar 0.1 maka akan kembali
13 ke form pembobotan
14 if($cr>0.1){
15
16 header("Location:form1.php?sukses=typ
17 ef"); //note: form pembobotan
18 }
19 else{
20 include("dbcon.php");
21
22 header("Location: form2.php");
23 }
24
25
26 ?>

```

Pada baris kedua hingga keenam merupakan perhitungan untuk mendapatkan *eigen vector*. Pada baris kesembilan merupakan rumus CI artinya bila nilai eigenmaksimum dikurangi banyaknya node lalu dibagi banyaknya node dikurangi 1.

Sedangkan untuk baris 10 merupakan baris CR dimana nilai CI yang sudah didapat pada baris Sembilan dibagi dengan nilai RI yang telah ditentukan berdasarkan banyaknya node yang digunakan. Pada baris 12 hingga 16 menjelaskan bahwa bila hasil $CR > 0.1$ maka proses perhitungan untuk mendapatkan eigen vector harus diulang atau kembali pada form awal. Sedangkan baris 18 hingga 21 bila nilai CR memenuhi syarat yaitu dibawah 0.1 maka proses dilanjutkan ke form berikutnya.

4.3.3. Proses *Unweighted Supermatrix*

Setelah selesai melakukan pemilihan terhadap perbandingan berpasangan dan telah mendapatkan *eigen vector* tahap selanjutnya mendapatkan matriks tidak berbobot. Proses dapat dilihat pada Tabel 4.9.

Tabel 4.9 *Source code* mengambil data *eigen vector* dari *database*

1	<?php
2	
3	include ("dbcon.php");
4	
5	\$c[]=0;
6	\$x=1;
7	\$query = mysql_query("SELECT `No`,`Node_Kolom`,`Node_baris`,`
8	Hasil` FROM `perkalian_supermatriks`");
9	
10	
11	
12	

Untuk mengambil nilai *eigen vector* dari *database* yang digunakan dalam proses *unweighted supermatrix* maka pada baris ketujuh melakukan pemanggilan pada isi dari tabel *perkalian_supermatriks* di *database*, isi tabel tersebut berupa *No*, *Node_Kolom*, *Node_baris* dan *Hasil*.

Selanjutnya mendapatkan matriks tidak berbobot. Proses dapat dilihat pada Tabel 4.10.

Tabel 4.10 *Source code matriks tidak berbobot/ unweighted supermatrix*

```

1 while($row = mysql_fetch_array($query)){
2     $kolom[$x]= $row[1];
3     $baris[$x]= $row[2];
4     $nilai[$x]= $row[3];
5
6     $x=$x+1;
7 }
8
9 $a=array(
10 array($nilai[1],$nilai[2],$nilai[3],$nilai[4],$nilai[5],$nilai[6],$nilai[7],$nilai
11 [8],$nilai[9],$nilai[10],$nilai[11],$nilai[12],$nilai[13],$nilai[14],$nilai[15],$
12 nilai[16],$nilai[17],$nilai[18],$nilai[19],$nilai[20],$nilai[21]),
13
14 array($nilai[22],$nilai[23],$nilai[24],$nilai[25],$nilai[26],$nilai[27],$nilai[2
15 8],$nilai[29],$nilai[30],$nilai[31],$nilai[32],$nilai[33],$nilai[34],$nilai[35],
16 $nilai[36],$nilai[37],$nilai[38],$nilai[39],$nilai[40],$nilai[41],$nilai[42]),
17
18 array($nilai[43],$nilai[44],$nilai[45],$nilai[46],$nilai[47],$nilai[48],$nilai[4
19 9],$nilai[50],$nilai[51],$nilai[52],$nilai[53],$nilai[54],$nilai[55],$nilai[56],
20 $nilai[57],$nilai[58],$nilai[59],$nilai[60],$nilai[61],$nilai[62],$nilai[63]),
21
22 array($nilai[64],$nilai[65],$nilai[66],$nilai[67],$nilai[68],$nilai[69],$nilai[7
23 0],$nilai[71],$nilai[72],$nilai[73],$nilai[74],$nilai[75],$nilai[76],$nilai[77],
24 $nilai[78],$nilai[79],$nilai[80],$nilai[81],$nilai[82],$nilai[83],$nilai[84]),
25
26 array($nilai[85],$nilai[86],$nilai[87],$nilai[88],$nilai[89],$nilai[90],$nilai[9
27 1],$nilai[92],$nilai[93],$nilai[94],$nilai[95],$nilai[96],$nilai[97],$nilai[98],
28 $nilai[99],$nilai[100],$nilai[101],$nilai[102],$nilai[103],$nilai[104],$nilai[1
29 05]),
30
31 array($nilai[106],$nilai[107],$nilai[108],$nilai[109],$nilai[110],$nilai[111],
32 $nilai[112],$nilai[113],$nilai[114],$nilai[115],$nilai[116],$nilai[117],$nilai[
33 118],$nilai[119],$nilai[120],$nilai[121],$nilai[122],$nilai[123],$nilai[124],$
34 nilai[125],$nilai[126]),

```


35	
36	array(\$nilai[127],\$nilai[128],\$nilai[129],\$nilai[130],\$nilai[131],\$nilai[132],
37	\$nilai[133],\$nilai[134],\$nilai[135],\$nilai[136],\$nilai[137],\$nilai[138],\$nilai[
38	139],\$nilai[140],\$nilai[141],\$nilai[142],\$nilai[143],\$nilai[144],\$nilai[145],\$
39	nilai[146],\$nilai[147]),
40	
41	array(\$nilai[148],\$nilai[149],\$nilai[150],\$nilai[151],\$nilai[152],\$nilai[153],
42	\$nilai[154],\$nilai[155],\$nilai[156],\$nilai[157],\$nilai[158],\$nilai[159],\$nilai[
43	160],\$nilai[161],\$nilai[162],\$nilai[163],\$nilai[164],\$nilai[165],\$nilai[166],\$
44	nilai[167],\$nilai[168]),
45	
46	array(\$nilai[169],\$nilai[170],\$nilai[171],\$nilai[172],\$nilai[173],\$nilai[174],
47	\$nilai[175],\$nilai[176],\$nilai[177],\$nilai[178],\$nilai[179],\$nilai[180],\$nilai[
48	181],\$nilai[182],\$nilai[183],\$nilai[184],\$nilai[185],\$nilai[186],\$nilai[187],\$
49	nilai[188],\$nilai[189]),
50	
51	array(\$nilai[190],\$nilai[191],\$nilai[192],\$nilai[193],\$nilai[194],\$nilai[195],
52	\$nilai[196],\$nilai[197],\$nilai[198],\$nilai[199],\$nilai[200],\$nilai[201],\$nilai[
53	202],\$nilai[203],\$nilai[204],\$nilai[205],\$nilai[206],\$nilai[207],\$nilai[208],\$
54	nilai[209],\$nilai[210]),
55	
56	array(\$nilai[211],\$nilai[212],\$nilai[213],\$nilai[214],\$nilai[215],\$nilai[216],
57	\$nilai[217],\$nilai[218],\$nilai[219],\$nilai[220],\$nilai[221],\$nilai[222],\$nilai[
58	223],\$nilai[224],\$nilai[225],\$nilai[226],\$nilai[227],\$nilai[228],\$nilai[229],\$
59	nilai[230],\$nilai[231]),
60	
61	array(\$nilai[232],\$nilai[233],\$nilai[234],\$nilai[235],\$nilai[236],\$nilai[237],
62	\$nilai[238],\$nilai[239],\$nilai[240],\$nilai[241],\$nilai[242],\$nilai[243],\$nilai[
63	244],\$nilai[245],\$nilai[246],\$nilai[247],\$nilai[248],\$nilai[249],\$nilai[250],\$
64	nilai[251],\$nilai[252]),
65	
66	array(\$nilai[253],\$nilai[254],\$nilai[255],\$nilai[256],\$nilai[257],\$nilai[258],
67	\$nilai[259],\$nilai[260],\$nilai[261],\$nilai[262],\$nilai[263],\$nilai[264],\$nilai[
68	265],\$nilai[266],\$nilai[267],\$nilai[268],\$nilai[269],\$nilai[270],\$nilai[271],\$
69	nilai[272],\$nilai[273]),
70	
71	array(\$nilai[274],\$nilai[275],\$nilai[276],\$nilai[277],\$nilai[278],\$nilai[279],
72	\$nilai[280],\$nilai[281],\$nilai[282],\$nilai[283],\$nilai[284],\$nilai[285],\$nilai[
73	286],\$nilai[287],\$nilai[288],\$nilai[289],\$nilai[290],\$nilai[291],\$nilai[292],\$
74	nilai[293],\$nilai[294]),
75	
76	array(\$nilai[295],\$nilai[296],\$nilai[297],\$nilai[298],\$nilai[299],\$nilai[300],

```

77 $nilai[301],$nilai[302],$nilai[303],$nilai[304],$nilai[305],$nilai[306],$nilai[
78 307],$nilai[308],$nilai[309],$nilai[310],$nilai[311],$nilai[312],$nilai[313],$
79 nilai[314],$nilai[315]),
80
81 array($nilai[316],$nilai[317],$nilai[318],$nilai[319],$nilai[320],$nilai[321],
82 $nilai[322],$nilai[323],$nilai[324],$nilai[325],$nilai[326],$nilai[327],$nilai[
83 328],$nilai[329],$nilai[330],$nilai[331],$nilai[332],$nilai[333],$nilai[334],$
84 nilai[335],$nilai[336]),
85
86 array($nilai[337],$nilai[338],$nilai[339],$nilai[340],$nilai[341],$nilai[342],
87 $nilai[343],$nilai[344],$nilai[345],$nilai[346],$nilai[347],$nilai[348],$nilai[
88 349],$nilai[350],$nilai[351],$nilai[352],$nilai[353],$nilai[354],$nilai[355],$
89 nilai[356],$nilai[357]),
90
91 array($nilai[358],$nilai[359],$nilai[360],$nilai[361],$nilai[362],$nilai[363],
92 $nilai[364],$nilai[365],$nilai[366],$nilai[367],$nilai[368],$nilai[369],$nilai[
93 370],$nilai[371],$nilai[372],$nilai[373],$nilai[374],$nilai[375],$nilai[376],$
94 nilai[377],$nilai[378]),
95
96 array($nilai[379],$nilai[380],$nilai[381],$nilai[382],$nilai[383],$nilai[384],
97 $nilai[385],$nilai[386],$nilai[387],$nilai[388],$nilai[389],$nilai[390],$nilai[
98 391],$nilai[392],$nilai[393],$nilai[394],$nilai[395],$nilai[396],$nilai[397],$
99 nilai[398],$nilai[399]),
100
101 array($nilai[400],$nilai[401],$nilai[402],$nilai[403],$nilai[404],$nilai[405],
102 $nilai[406],$nilai[407],$nilai[408],$nilai[409],$nilai[410],$nilai[411],$nilai[
103 412],$nilai[413],$nilai[414],$nilai[415],$nilai[416],$nilai[417],$nilai[418],$
104 nilai[419],$nilai[420]),
105
106 array($nilai[421],$nilai[422],$nilai[423],$nilai[424],$nilai[425],$nilai[426],
107 $nilai[427],$nilai[428],$nilai[429],$nilai[430],$nilai[431],$nilai[432],$nilai[
108 433],$nilai[434],$nilai[435],$nilai[436],$nilai[437],$nilai[438],$nilai[439],$
109 nilai[440],$nilai[441])

```

Pada baris pertama sebagai pemberitahuan bahwa akan menyimpan beberapa nilai pada beberapa row/baris. Baris kedua hingga keempat merupakan penjelasan bahwa untuk \$row[1]; untuk kolom node, untuk \$row[2]; untuk kolom baris node dan

\$row[3]; untuk kolom nilai dari *eigen vector* yang nantinya digunakan untuk proses *unweighted*, *weighted* dan *limiting supermatrix*. Baris keenam berfungsi bila terus melakukan penambahan data kolom, baris dan nilai dapat dilihat. Baris ke-9, \$a=array menjelaskan keseluruhan nilai array yang ada di \$a dimiliki oleh *unweighted supermatrix*. Baris ke-10 hingga baris ke-109 merupakan nilai keseluruhan node kolom maupun baris yang disimpan dalam bentuk array, mulai dari node ke-1 hingga node ke 21.

4.3.4. Proses *Weighted Supermatrix*

Proses *weighted supermatrix* setelah mendapatkan matriks tak berbobot atau *unweighted matrix* maka kita akan mendapatkan weight supermatrix dengan mengalikan *unweighted matrix* dengan *cluster matrix*. Proses *weighted Supermatrix* dapat di lihat pada Tabel 4.11.

Tabel 4.11 Source code mengambil data normalisasi *cluster matrix*

1	<?php \$x=1;
2	\$query = mysql_query("SELECT `rekrutmen`, `profil mahasiswa dan
3	lulusan`, `profil alumni` `layanan` FROM
4	`cluster`");
5	while(\$row = mysql_fetch_array(\$query)){
6	\$rekrutmen= \$row[0];
7	\$profmhs= \$row[1];
8	\$lulusan= \$row[2];
9	\$layanan=\$row[3];
10	\$x=\$x+1;
	}

Untuk mengambil nilai *cluster matrix* dari *database* yang digunakan dalam proses *weighted supermatrix* maka pada baris kedua melakukan pemanggilan pada isi dari tabel *cluster* di *database*, isi tabel tersebut berupa rekrutmen, profil mahasiswa dan lulusan, profil alumni, serta layanan.

Maka proses selanjutnya adalah mengalikan *cluster matrix* dengan *unweighted matrix*. Proses source code bisa dilihat pada Tabel 4.12.

Tabel 4.12 Source code mengalikan dengan *cluster matrix*

```

1 <table>
2 <tr >
3 <td >unweighed supermatix</td>
4 <td><?php echo
5 $e[0][0]=($b[0][0]*$rekrutmen)+($b[0][0]*$profmhs)+($b[0][0]*$lulusan)+
6 ($b[0][0]*$layan); ?></td>
7 <td><?php echo
8 $e[0][1]=($b[0][1]*$rekrutmen)+($b[0][1]*$profmhs)+($b[0][1]*$lulusan)+
9 ($b[0][1]*$layan); ?></td>
10 <td><?php echo
11 $e[0][2]=($b[0][2]*$rekrutmen)+($b[0][2]*$profmhs)+($b[0][2]*$lulusan)+
12 ($b[0][2]*$layan); ?></td>

```

```

13 <td><?php echo
14 $e[0][3]=($b[0][3]*$rekrutmen)+($b[0][3]*$profmhs)+($b[0][3]*$lulusan)+
15 ($b[0][3]*$layan); ?></td>
16 ; ?></td>
17 <td><?php echo
18 $e[0][4]=($b[0][4]*$rekrutmen)+($b[0][4]*$profmhs)+($b[0][4]*$lulusan)+
19 ($b[0][4]*$layan); ?></td>
20 ; ?></td>
21 <td><?php echo
22 $e[0][5]=($b[0][5]*$rekrutmen)+($b[0][5]*$profmhs)+($b[0][5]*$lulusan)+
23 ($b[0][5]*$layan); ?></td>
24 ; ?></td>
25 <td><?php echo
26 $e[0][6]=($b[0][6]*$rekrutmen)+($b[0][6]*$profmhs)+($b[0][6]*$lulusan)+
27 ($b[0][6]*$layan); ?></td>
28 ; ?></td>
29 <td><?php echo
30 $e[0][7]=($b[0][7]*$rekrutmen)+($b[0][7]*$profmhs)+($b[0][7]*$lulusan)+
31 ($b[0][7]*$layan); ?></td>
32 ; ?></td>
33 <td><?php echo
34 $e[0][8]=($b[0][8]*$rekrutmen)+($b[0][8]*$profmhs)+($b[0][8]*$lulusan)+

```

```
35 ($b[0][8]*$layanan); ?></td>
36 ; ?></td>
37 <td><?php echo
38 $e[0][9]=($b[0][9]*$rekrutmen)+($b[0][9]*$profmhs)+($b[0][9]*$lulusan)+
39 ($b[0][9]*$layanan); ?></td>
40 ; ?></td>
41 <td><?php echo
42 $e[0][10]=($b[0][10]*$rekrutmen)+($b[0][10]*$profmhs)+($b[0][10]*$lulusa
43 n)+($b[0][10]*$layanan); ?></td>
44 ; ?></td>
45 <td><?php echo
46 $e[0][11]=($b[0][11]*$rekrutmen)+($b[0][11]*$profmhs)+($b[0][11]*$lulusa
47 n)+($b[0][11]*$layanan); ?></td>
48 ; ?></td>
49 <td><?php echo
50 $e[0][12]=($b[0][12]*$rekrutmen)+($b[0][12]*$profmhs)+($b[0][12]*$lulusa
51 n)+($b[0][12]*$layanan); ?></td>
52 ; ?></td>
53 <td><?php echo
54 $e[0][13]=($b[0][13]*$rekrutmen)+($b[0][13]*$profmhs)+($b[0][13]*$lulusa
55 n)+($b[0][13]*$layanan); ?></td>
56 ; ?></td>
57 <td><?php echo
58 $e[0][14]=($b[0][14]*$rekrutmen)+($b[0][14]*$profmhs)+($b[0][14]*$lulusa
59 n)+($b[0][14]*$layanan); ?></td>
60 ; ?></td>
61 <td><?php echo
62 $e[0][15]=($b[0][15]*$rekrutmen)+($b[0][15]*$profmhs)+($b[0][15]*$lulusa
63 n)+($b[0][15]*$layanan); ?></td>
64 ; ?></td>
65 <td><?php echo
66 $e[0][16]=($b[0][16]*$rekrutmen)+($b[0][16]*$profmhs)+($b[0][16]*$lulusa
67 n)+($b[0][16]*$layanan); ?></td>
68 ; ?></td>
69 <td><?php echo
70 $e[0][17]=($b[0][17]*$rekrutmen)+($b[0][17]*$profmhs)+($b[0][17]*$lulusa
71 n)+($b[0][17]*$layanan); ?></td>
72 ; ?></td>
73 <td><?php echo
74 $e[0][18]=($b[0][18]*$rekrutmen)+($b[0][18]*$profmhs)+($b[0][18]*$lulusa
75 n)+($b[0][18]*$layanan); ?></td>
76 ; ?></td>
```

```

77 <td><?php echo
78 $e[0][19]=($b[0][19]*$rekrutmen)+($b[0][19]*$profmhs)+($b[0][19]*$lulusan
79 n)+($b[0][19]*$layan); ?></td>
80 ; ?></td>
81 <td><?php echo
82 $e[0][20]=($b[0][20]*$rekrutmen)+($b[0][20]*$profmhs)+($b[0][20]*$lulusan
83 n)+($b[0][20]*$layan); ?></td>
84 ; ?></td>
85 $e[0][21]=($b[0][21]*$rekrutmen)+($b[0][21]*$profmhs)+($b[0][21]*$lulusan
86 n)+($b[0][21]*$layan); ?></td>
87 ; ?></td>
88
89 </tr>
90 </table>

```

Pada baris keempat hingga keenam merupakan proses *unweighted supermatrix* dikali dengan *cluster matrix* yang menghasilkan *weighted supermatrix* untuk node ke-1. *Cluster matrix* yaitu rekrutmen, profil mahasiswa dan lulusan, profil alumni, serta layanan. Proses tersebut berlaku untuk seluruh node yaitu node ke-1 hingga node ke-21 dimulai pada baris ke-4 hingga baris ke-86.

4.3.5. Proses *Limiting Supermatrix*

Source code ini merupakan proses perhitungan terakhir dari metode *Analytic Network Process (ANP)*. Perhitungan *Limitingsupermatrix* merupakan hasil perkalian *weighted supermatrix* dengan dirinya hingga didapatkan nilai yang sama besar, dapat dilihat pada Tabel 4.13.

Tabel 4.13 *Source code Limiting supermatrix*

```

1  for($i=0;$i<21;$i++){
2      for($j=0;$j<21;$j++){
3          $sum=0;
4          for($k=0;$k<21;$k++){
5              $sum=$sum+($a[$i][$k]*$b[$k][$j]);
6          }
7          $c[$i][$j]=$sum;
8      echo "data i = ".$i." dan j = ".$j." hasilnya : ".$c[$i][$j]."<br />";
9  }
10 }
11 for($i=0;$i<21;$i++){
12     for($j=0;$j<21;$j++){
13         $sum=0;
14         for($k=0;$k<21;$k++){
15             $sum=$sum+($c[$i][$k]*$a[$k][$j]);
16         }
17         $d[$i][$j]=$sum;
18     echo "data i = ".$i." dan j = ".$j." hasilnya : ".$d[$i][$j]."<br />";
19 }
20 }
```

Pada baris ke-1 merupakan penjelasan mengenai perulangan nilai baris dari node ke-1 hingga ke-21 dan akan terus bertambah, pada baris ke-2 merupakan penjelasan mengenai perulangan nilai kolom dari node ke-1 hingga ke-21 dan akan terus bertambah. Pada baris ke-4 merupakan penjelasan mengenai perulangan nilai baris dan kolom dari node ke-1 hingga ke-21 yang dimiliki oleh *weighted supermatrix* untuk proses *limiting supermatrix*. Pada baris ke-5 merupakan perkalian antara *weighted supermatrix* dengan dirinya sendiriselingga mendapatkan *limiting supermatrix* pangkat 2/iterasi kedua. Baris ke-7 menjelaskan ketika sudah didapatkan

dari melalui perkalian supermatriks sehingga baris ke-8 menjelaskan hasil dimana limiting supermatrix pangkat 2 disimpan pada array $\$c[\$i][\$j]$. Baris ke-11 hingga ke-18 menjelaskan hal yang sama seputar limiting supermatrix pangkat 3, prosesnya sama seperti baris-baris sebelumnya yang membedakan adalah Pada baris ke-15 merupakan perkalian antara *limiting supermatrix* pangkat 2/iterasi kedua dengan *weighted supermatrix* sehingga mendapatkan *limiting supermatrix* pangkat 3/iterasi ketiga yang artinya $\$c[\$i][\$k]*\$a[\$k][\$j]$. Baris ke-17 menjelaskan ketika sudah didapatkan dari melalui perkalian supermatriks sehingga baris ke-18 menjelaskan hasil dimana limiting supermatrix pangkat 3 disimpan pada array $\$d[\$i][\$j]$.



