

LAMPIRAN

LAMPIRAN 1

Source Code file Aplikasi Berbasis Web

Source code file Aplikasi.php

```
<style type="text/css">
<!--
    .style3 {
        font-size: 24px;
        font-family: Arial, Helvetica, sans-serif;
    }
    .style5 {color: #0000FF}
    .style6 {color: #0033FF}
--></style>
<?php require("Header.php"); ?>
<script type="text/JavaScript">
<!--
function MM_findObj(n, d) { //v4.01
    var p,i,x; if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
        d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
    if(!(x=d[n])&&d.all) x=d.all[n];
    for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
    for(i=0;!x&&d.layers&&i<d.layers.length;i++)
        x=MM_findObj(n,d.layers[i].document);
    if(!x && d.getElementById) x=d.getElementById(n); return x;
}
function MM_validateForm() { //v4.0
    var i,p,q,nm,test,num,min,max,errors="",args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=MM_findObj(args[i]);
    if (val) { nm=val.name; if ((val=val.value)!="") {
        if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
            if (p<1 || p==(val.length-1)) errors+='- '+nm+' must contain an e-mail
            address.\n';
        } else if (test!='R') { num = parseFloat(val);
            if (isNaN(val)) errors+='textbox jumlah spesies harus berupa bilangan
            positif\n';
        } if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
            min=test.substring(8,p); max=test.substring(p+1);

```

```

        if (max<num) errors+='Untuk jumlah spesies lebih besar dari 25,
            gunakan input berupa text file\n';
        if (min>num) errors+='Maaf jumlah spesies tidak boleh negatif\n';
    } } else if (test.charAt(0) == 'R') errors += ' - '+nm+' is required.\n'; }
} if (errors) alert('\n'+errors);
document.MM_returnValue = (errors == '');
}
//--></script>
<p>&nbsp;</p>
<table width="970" border="0" cellspacing="0" cellpadding="0">
  <tr><td width="200">&nbsp;</td>
    <td width="570"><div align="center">
      <p class="style3">Aplikasi untuk membangun &quot;Phylogenetic
        tree&quot; </p>
      <p class="style3">&nbsp;</p>
    </div></td>
  <td width="200">&nbsp;</td></tr>
  <tr>
    <td height="127"><div align="center"></div></td>
    <td rowspan="2" bgcolor="#95CEFD"><form id="form1" name="form1" method="post"
action="Input.php" ENCTYPE="multipart/form-data">
      <div align="center">
        <h3> <span class="style5">Masukan jumlah OTU pada kotak di bawah ini</span> </h3>
        <p><input name="jmlh_spcs" type="text" id="jmlh_spcs" value="1" size="4"
maxlength="1000" /></p>
        <h3 class="style6">Atau Anda dapat meng-upload file <em>distance matrix</em><br>
pada kotak di bawah ini</h3>
        <p><label><input name="file" type="file" size="40" maxlength="100" />
</label></p> <br>
        <p><input name="Submit" type="submit"
onclick="MM_validateForm('jmlh_spcs','NinRange1:25');return document.MM_returnValue"
value=" O K " /></p>
      </div>
    </form></td>
  <td><div align="center"></div></td>
</tr>
  <tr>
    <td height="129"><div align="center"></div></td>
    <td><div align="center"></div></td>
</tr>
</table>

```

```
<?php require("Footer.php");?>
```

Source code file Input.php

```
<style type="text/css">
<!--
.style2 {font-family: "Courier New", Courier, monospace}
.style4 {color: #0000FF; font-family: Geneva, Arial, Helvetica, sans-serif;}
-->
</style>
<?php require ("Header.php");
$uploaddir = 'Uploads';
if(is_uploaded_file($_FILES['file']['tmp_name'])){
move_uploaded_file($_FILES['file']['tmp_name'],$uploaddir.'/'.$_FILES['file']['name']);
$UploadFile = $uploaddir.'/'.$_FILES['file']['name'];
$fh = fopen($UploadFile, 'r');
$cond = true;
while (!feof($fh)) {
    $line[] =explode(" ",fgets($fh, 4096));}
fclose($fh);
$jml_spcs = sizeof($line[0]) - 1 ;
    for($i=0;$i<=sizeof($line)-1;$i++){
        if ($_FILES['file']['type']!="text/plain" or sizeof($line[$i])!=$jml_spcs + 1 or sizeof($line[$i])
== sizeof($line)){
            $cond = false;
            echo '<center></center>';
            if($_FILES['file']['type']!="text/plain")
                echo "<h3><center>File yang Anda upload harus berupa textfile </center></h3>
<br>";
            if(sizeof($line[$i]) == sizeof($line))
                echo "<h3><center>Anda Harus Menambahkan Nama Spesies Pada File Anda
</center></h3><br>"; break;}}
if($cond){
?><form action="Check.php" method="post" name="form1" id="form1" target =
"_blank"><p><center>
<? echo '';
echo "<br><br> ";
echo " <input name=\"Input2\" type=\"submit\" value=\" Lanjutkan \" /> " ;
for($c=1;$c<=$jml_spcs;$c++){
    echo "<input name=\"nama\".$c.\"\" type=\"hidden\" size=\"1\" maxlength=\"10\" value=
\"{$line[$c-1][0]}\">"; }
for($k=1; $k<=$jml_spcs; $k++){
    for($i=1; $i<=$jml_spcs; $i++){
```

```

        if ($k==$i){
            echo "<input name='distance'."$k.$i.'" type='hidden' size='1' readonly='true'
value='0'>";
        } else
            echo "<input name='distance'."$k.$i.'" type='hidden' size='1' maxlength='10'
value='\"{$line[$k-1][$i]}\">"; }
?></p><p>&nbsp;</p>
    <input name="jml_spcs" type="hidden" value="<?php echo "$jml_spcs"; ?>" />
</form>
<?php}&#123; else {
?>
<h1 align="center"><br />
<tt>Matriks Jarak Antar Spesies</tt></h1>
<h1 align="center" ><tt>(Distance Matrix)</tt></h1>
<form action="Check.php" method="post" name="form1" id="form1" target =
"_blank"><p><center>
<?php
$jml_spcs=$_POST["jmlh_spcs"];
echo '<input type="text" size="10" readonly="true" value="Nama OTU" > &nbsp;&#123;';
echo '<input type="text" size="2" readonly="true" value="Md" disabled/>';
for($j=1; $j<=$jml_spcs; $j++){
echo "<input type='text' size='1' readonly='true' value='x$j' disabled>";
echo '<br>';
for($k=1; $k<=$jml_spcs; $k++){
echo "<input name='nama'."$k.'" type='text' size='10' > &nbsp;&#123;";
echo "<input type='text' size='2' readonly='true' value='x$k' disabled >";
for($i=1; $i<=$jml_spcs; $i++){
    if ($k==$i){
        echo "<input name='distance'."$k.$i.'" type='text' size='1' readonly='true'
value='0'>"; } elseif ($k>=$i){
        echo "<input name='distance'."$k.$i.'" type='text' size='1' readonly='true'
value='***' disabled>"; } else
        echo "<input name='distance'."$k.$i.'" type='text' size='1' maxlength='10'>";
    }echo "<br>";}
?>
</p> <p>&nbsp;</p>
    <p><input name="jml_spcs" type="hidden" value="<?php echo "$jml_spcs"; ?>" />
    <input name="Input2" type="submit" value=" Lanjutkan " /></p>
</form>
<?php } require ("Footer.php");?>

```

Source code file Check.php

```
<style type="text/css">
<!--
.style2 {font-family: "Courier New", Courier, monospace}
.style4 {color: #0000FF; font-family: Geneva, Arial, Helvetica, sans-serif;}
-->
</style>
<html>
<head>
<title>Result</title>
</head><body bgcolor="#FFFFFF">
<h1 align="center"><br />
  <span class="style4"><tt> I N F O R M A S I </tt></span></h1><center>
<form action="Image.php" method="post" name="form1" id="form1"><p>
<? $N=$_POST['jml_spcs'];
for($k=1; $k<=$N; $k++){
  $nama[$k]=$_POST['nama$k'];
  for($i=1; $i<=$N; $i++){
    if ($k<=$i){
      $distance[$k][$i]=$_POST['distance$k$i'];
    } else
      $distance[$k][$i]=$distance[$i][$k];}
for($i=1; $i<=$N; $i++){
  for($j=$i+1; $j<=$N; $j++){
    for($k=$j+1; $k<=$N; $k++){
      for($l=$k+1; $l<=$N; $l++){
        $d = 0;
        $a = $distance[$i][$j]+$distance[$k][$l];
        $b = $distance[$i][$k]+$distance[$j][$l];
        $c = $distance[$i][$l]+$distance[$j][$k];
        if (($a==$b) && ($a >= $c) ){ $d = 1;}
        if (($a==$c) && ($a >= $b) ) { $d = 1;}
        if (($b==$c) && ($b >= $a) ) { $d = 1;}
        if ($d == 0){ $status = "False";break;}
      }}}
}
echo '<input type="text" size="2" readonly="true" value="Md" disabled/>';
for($j=1; $j<=$N; $j++){
echo "<input type='text' size='1' readonly='true' value='x$j' disabled>";
}
echo '<br>';
for($k=1; $k<=$N; $k++){
```

```

echo "<input type='text' size='2' readonly='true' value='x$k' disabled >";
for($i=1; $i<=$N; $i++){
    if ($k==$i){
        echo "<input name='distance'.$k.$i.'" type='text' size='1' readonly='true'
value='0'>";
    } elseif ($k>$i){
        echo "<input name='distance'.$k.$i.'" type='text' size='1' readonly='true'
value='***' disabled>";
    } else
        echo "<input name='distance'.$k.$i.'" type='text' size='1' maxlength='10' value =
'{$distance[$k][$i]}'>";
    }
    echo "<br>";
}
for($c=1;$c<=$N;$c++){
    echo "<input name='nama'.$c.'" type='hidden' size='1' maxlength='10' value=
'${nama[$c]}'>";
}
if ($status == "False"){
echo '<table width="500" height="20" border="0">' ;
echo '<tr>' ;
echo '<td width="500"><div align="center">';
echo '';
echo '<br>Distance Matrix diatas tidak memenuhi syarat additivity </div></td>';
echo '</tr>';
echo '<tr>' ;
echo '<td><div align="center">' ;
echo '<br>Klik Gambar untuk melanjutkan';
echo '</div></td>' ;
echo '</tr>' ;
echo '</table>' ;
} else {
echo '<table width="500" height="20" border="0">' ;
echo '<tr>' ;
echo '<td width="500"><div align="center">';
echo '';
echo '<br>Distance Matrix diatas memenuhi syarat additivity</div></td>';
echo '</tr>';
echo '<tr>' ;
echo '<td><div align="center">' ;
echo '<br>Klik Gambar untuk membangun Phylogenetic Tree';
echo '</div></td>' ;
echo '</tr>' ;
echo '</table>' ;
}
}

```

```

?>
</p>
</p>
<table width="285" border="1" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td width="209"><div align="left"><span class="style2">Background Color</span>
  </div></td>
    <td width="76">
      <div align="left">
        <select name="back">
          <option value="Black" >Black</option>
          <option value="Blue">Blue</option>
          <option value="Grey" selected >Grey</option>
          <option value="Green">Green</option>
          <option value="Purple">Purple</option>
          <option value="Red">Red</option>
          <option value="Soft">Soft Blue</option>
          <option value="Yellow">Yellow</option>
          <option value="White">White</option>
        </select>
      </div></td>
  </tr>
  <tr>
    <td><div align="left"><span class="style2">Node Color </span></div></td>
    <td>
      <div align="left">
        <select name="node">
          <option value="Black" selected >Black</option>
          <option value="Blue">Blue</option>
          <option value="Grey">Grey</option>
          <option value="Green">Green</option>
          <option value="Purple">Purple</option>
          <option value="Red">Red</option>
          <option value="Soft">Soft Blue</option>
          <option value="Yellow">Yellow</option>
          <option value="White">White</option>
        </select>
      </div></td>
  </tr>
  <tr>
    <td><div align="left"><span class="style2">Edge Color </span></div></td>
    <td>
      <div align="left">
        <select name="edge">
          <option value="Black" selected >Black</option>

```

```

        <option value="Blue">Blue</option>
        <option value="Grey">Grey</option>
        <option value="Green">Green</option>
        <option value="Purple">Purple</option>
        <option value="Red">Red</option>
        <option value="Soft">Soft Blue</option>
        <option value="Yellow">Yellow</option>
        <option value="White">White</option>
    </select>
</div></td>
</tr>
<tr>
<td><div align="left"><span class="style2">Text Color </span></div></td>
<td><div align="left">
    <select name="text">
        <option value="Black">Black</option>
        <option value="Blue">Blue</option>
        <option value="Grey">Grey</option>
        <option value="Green">Green</option>
        <option value="Purple">Purple</option>
        <option value="Red">Red</option>
        <option value="Soft">Soft Blue</option>
        <option value="Yellow">Yellow</option>
        <option value="White" selected >White</option>
    </select>
</div></td>
</tr>
</table>
<p><label></label>
</p>
<p><input name="jml_spcs" type="hidden" value="<?php echo "$N"; ?>" />
    <input name="Input2" type="submit" value=" Gambar " />
    <label></label>
</form>
</center>
</body>
</html>

```


Source code file Image.php

```
<?php
$N=$_POST['jml_spcs']; $jml_spcs=$N; $count_set=$N;
for($k=1; $k<=$N; $k++){
    $nama[$k]=$_POST["nama$k"];
    for($i=1; $i<=$N; $i++){
        if ($k<=$i){
            $distance[$k][$i]=$_POST["distance$k$i"];
        } else
            $distance[$k][$i]=$distance[$i][$k];
    }
}
function sum_row($distance,$N,$count_set){
    for ($i=1; $i<=$N; $i++){
        $jumlah=0;
        for ($j=1; $j<=$N ;$j++){ $jumlah=$jumlah+$distance[$i][$j];}
        $r[$i]=(1/($count_set-2))*$jumlah;} return $r;
}
function D($r,$distance,$N,$set){
    for ($j=2; $j<=$N; $j++){
        for ($i=1; $i<$j ;$i++){
            if(check($i,$set) || check($j,$set)){ $D[$i][$j] = 100;}
            else
                $D[$i][$j]=($distance[$i][$j])-( $r[$i]+$r[$j]);
        }
    } return $D;
}
function find_min($D,$N){
    $min=$D[1][2];
    $m["baris"]=1;
    $m["kolom"]=2;
    for ($j=2; $j<=$N; $j++){
        for ($i=1; $i<$j ;$i++){
            if($D[$i][$j]<$min){
                $min= $D[$i][$j]; $m["baris"]=$i; $m["kolom"]=$j;
            }
        }
    } return $m;
}
function update_distance($distance,$next_otu,$m,$set){
    for ($i=1; $i<=$next_otu; $i++){
        if (check($i,$set)){
            $distance[$next_otu][$i]=0;
        }
    }
}
```

```

    }else {
$distance[$next_otu][$i]=(1/2)*($distance[$m["baris"]][$i]+$distance[$m["kolom"]][$i]-
$distance[$m["baris"]][$m["kolom"]]);
    $distance[$i][$next_otu]=$distance[$next_otu][$i];
}
    $distance[$next_otu][$next_otu]=0;
    for($k=1; $k<=$next_otu; $k++){
        for($j=1; $j<=$next_otu; $j++){
            if(($k==$m["baris"] | $j==$m["kolom"])|($j==$m["baris"] | $k==$m["kolom"])){
                $distance[$k][$j]=0;
            }} return $distance;
    }
}
function check($i,$set){
    for ($j=0;$j<=sizeof($set);$j++){
        if ($i==$set[$j]){return true;}
    } return false;
}
}
function func_adj_node($mtrx_adj,$N){
    for($i=0; $i<=$N; $i++){
        for($j=0; $j<=$N; $j++){
            if(isset($mtrx_adj[$i][$j])){
                $adj_node[$i][]="$j";
            }} return $adj_node;
    }
}
function level_set($adj_node,$N){
    $level[-1]=array();
    $level[0]=array(0);
    for($i=1;$i<=$N;$i++){$set_node[]="$i";}
    $j=1;
    while (sizeof($set_node)> 0){
        $level[$j]=array();
        for($k=1; $k<=sizeof($level[$j-1]);$k++){
            $adj_node_diff=array_diff($adj_node[$level[$j-1][$k-1]],$level[$j-2]);
            $level[$j]=array_merge($level[$j],$adj_node_diff);}
        $set_node=array_diff($set_node,$level[$j]);
        $j++;
    } return $level;
}
}
function imagelinethick($image, $x1, $y1, $x2, $y2, $color, $thick = 1)
{
    if ($thick == 1) {
        return imageline($image, $x1, $y1, $x2, $y2, $color);}
    $t = $thick / 2 - 0.5;
    if ($x1 == $x2 || $y1 == $y2) {

```

```

    return imagefilledrectangle($image, round(min($x1, $x2) - $t), round(min($y1, $y2) - $t),
round(max($x1, $x2) + $t), round(max($y1, $y2) + $t), $color);}
    $k = ($y2 - $y1) / ($x2 - $x1); //y = kx + q
    $a = $t / sqrt(1 + pow($k, 2));
    $points = array(
        round($x1 - (1+$k)*$a), round($y1 + (1-$k)*$a),
        round($x1 - (1-$k)*$a), round($y1 - (1+$k)*$a),
        round($x2 + (1+$k)*$a), round($y2 - (1-$k)*$a),
        round($x2 + (1-$k)*$a), round($y2 + (1+$k)*$a),
    );
    imagefilledpolygon($image, $points, 4, $color);
    return imagepolygon($image, $points, 4, $color);
}
$loop=1;
$n=4;
$count=1;
$set=array();
while ($loop<=$jml_spcs-3){
    $r=sum_row($distance,$N,$count_set);
    $D=D($r,$distance,$N,$set);
    $m=find_min($D,$N);
    $add_element=array($m["baris"],$m["kolom"]);
    $set=array_merge($set,$add_element);
    $next_otu=$jml_spcs+$count;
    $d[$next_otu][$m["baris"]]=(1/2)*($distance[$m["baris"]][$m["kolom"]]+$r[$m["baris"]]-
    $r[$m["kolom"]]);
    $d[$next_otu][$m["kolom"]]=(1/2)*($distance[$m["baris"]][$m["kolom"]]+$r[$m["kolom"]]-
    $r[$m["baris"]]);
    $mtrx_adj[$next_otu][$m["baris"]]=1;$mtrx_adj[$m["baris"]][$next_otu]=1;
    $mtrx_adj[$next_otu][$m["kolom"]]=1;$mtrx_adj[$m["kolom"]][$next_otu]=1;
    $count++;
    $n=$n+2;
    $distance=update_distance($distance,$next_otu,$m,$set);
    $N=$next_otu;
    $loop++;
    $count_set--;
}
}
if ($count_set==3){
    for ($i=1;$i<=$next_otu;$i++){
        if (check($i,$set)==false){
            $comp_set[]=$i;}
    }
    if ($jml_spcs==3){
        $comp_set[0]=1; $comp_set[1]=2; $comp_set[2]=3;
    }
}

```

```

}
$x=(1/2)*($distance[$comp_set[0]][$comp_set[1]]+$distance[$comp_set[0]][$comp_set[2]]-
$distance[$comp_set[1]][$comp_set[2]]);
$y=(1/2)*($distance[$comp_set[0]][$comp_set[1]]+$distance[$comp_set[1]][$comp_set[2]]-
$distance[$comp_set[0]][$comp_set[2]]);
$z=(1/2)*($distance[$comp_set[0]][$comp_set[2]]+$distance[$comp_set[1]][$comp_set[2]]-
$distance[$comp_set[0]][$comp_set[1]]);
  $mtrx_adj[0][$comp_set[0]]=1;$mtrx_adj[$comp_set[0]][0]=1; $d[0][$comp_set[0]]=$x;
  $mtrx_adj[0][$comp_set[1]]=1;$mtrx_adj[$comp_set[1]][0]=1; $d[0][$comp_set[1]]=$y;
  $mtrx_adj[0][$comp_set[2]]=1;$mtrx_adj[$comp_set[2]][0]=1; $d[0][$comp_set[2]]=$z;
  $adj_node=func_adj_node($mtrx_adj,$N);
  $level=level_set($adj_node,$N);
}
function max_level($level){
  $j=0;
  for($i=1;$i<=count($level);$i++){
    if ($j<sizeof($level[$i]))
      $j=sizeof($level[$i]);
  }return $j;
}
$back=$_POST["back"]; $node=$_POST["node"]; $edge=$_POST["edge"];
$text=$_POST["text"];
if ($N != 2){
header("Content-type: image/png");
$max_level = max_level($level);
$size_x= 100 + $max_level * 150;
$size_y=count($level)*150;
$image = imagecreate($size_x,$size_y);
switch($back){
  case "Black": imagecolorallocate($image, 0, 0, 0); break;
  case "Red": imagecolorallocate($image, 255, 0, 0); break;
  case "Green": imagecolorallocate($image, 0, 255, 0); break;
  case "Yellow": imagecolorallocate($image, 255, 255, 0); break;
  case "White": imagecolorallocate($image, 255, 255, 255); break;
  case "Grey": imagecolorallocate($image, 200, 200, 200); break;
  case "Soft": imagecolorallocate($image, 0, 255, 255); break;
  case "Purple": imagecolorallocate($image, 255, 0, 255); break;
  case "Blue": imagecolorallocate($image, 0, 0, 255); break;
}
switch($node){
  case "Black": $col_node=imagecolorallocate($image, 0, 0, 0); break;
  case "Red": $col_node=imagecolorallocate($image, 255, 0, 0); break;
  case "Green": $col_node=imagecolorallocate($image, 0, 255, 0); break;
  case "Yellow": $col_node=imagecolorallocate($image, 255, 255, 0); break;
  case "White": $col_node=imagecolorallocate($image,255,255,255); break;
}

```

```

    case "Purple": $col_node=imagecolorallocate($image, 255, 0, 255); break;
    case "Grey": $col_node=imagecolorallocate($image,200,200,200); break;
    case "Soft": $col_node=imagecolorallocate($image, 0, 255, 255); break;
    case "Blue": $col_node=imagecolorallocate($image, 0, 0, 255); break;
}
switch($edge){
    case "Black": $col_edge=imagecolorallocate($image, 0, 0, 0); break;
    case "Red": $col_edge=imagecolorallocate($image, 255, 0, 0); break;
    case "Green": $col_edge=imagecolorallocate($image, 0, 255, 0); break;
    case "Yellow": $col_edge=imagecolorallocate($image, 255, 255, 0); break;
    case "White": $col_edge =imagecolorallocate($image,255,255,255);break;
    case "Purple": $col_edge=imagecolorallocate($image, 255, 0, 255); break;
    case "Grey": $col_edge=imagecolorallocate($image, 200, 200, 200);break;
    case "Soft": $col_edge=imagecolorallocate($image, 0, 255, 255); break;
    case "Blue": $col_edge=imagecolorallocate($image, 0, 0, 255); break;
}
switch($text){
    case "Black": $col_text =imagecolorallocate($image, 0, 0, 0); break;
    case "Red": $col_text =imagecolorallocate($image, 255, 0, 0); break;
    case "Green": $col_text =imagecolorallocate($image, 0, 255, 0); break;
    case "Yellow": $col_text =imagecolorallocate($image, 255, 255, 0); break;
    case "White": $col_text =imagecolorallocate($image,255,255,255); break;
    case "Purple": $col_text=imagecolorallocate($image, 255, 0, 255); break;
    case "Grey": $col_text =imagecolorallocate($image,200,200,200); break;
    case "Soft": $col_text=imagecolorallocate($image, 0, 255, 255); break;
    case "Blue": $col_text=imagecolorallocate($image, 0, 0, 255); break;
}
$node0=array(0 => array("x"=>$size_x/2,"y"=>100));
$node =array();
for ($a=1;$a<=count($level);$a++){
    $num=sizeof($level[$a]);
    for ($b=0;$b<$num;$b++){
        $temp=$node;
        $node=array($level[$a][$b] => array("x"=>($size_x/($num+1))*($b+1),"y"=>100+$a*150))
    ;
        $node=$node+$temp ;
        if ($a==1){
            imagelinethick
($image,$node0[0]["x"],$node0[0]["y"],$node[$level[$a][$b]]["x"],$node[$level[$a][$b]]["y"],$col_
edge,2);
            imagestring($image, 5,
($node[$level[$a][$b]]["x"]+$node0[0]["x"])/2,($node[$level[$a][$b]]["y"]+$node0[0]["y"])/2,round
($d[0][$level[$a][$b]],3), $col_text);
        }
    }
}
else{

```

```

    for ($c=0;$c<sizeof($level[$a-1]);$c++)
        if ($mtrx_adj[$level[$a][$b]][$level[$a-1][$c]]==1){
            imagelinethick
($image,$node[$level[$a][$b]]["x"],$node[$level[$a][$b]]["y"],$node[$level[$a-1][$c]]["x"],$node[$level[$a-1][$c]]["y"],$col_edge,2);
            imagestring($image, 5, ($node[$level[$a][$b]]["x"]+$node[$level[$a-1][$c]]["x"])/2,($node[$level[$a][$b]]["y"]+$node[$level[$a-1][$c]]["y"])/2, round($d[$level[$a-1][$c]][$level[$a][$b]],3), $col_text);
        }
    }
    if ($level[$a][$b]<=$jml_spcs){
        imagefilledellipse($image, $node[$level[$a][$b]]["x"],$node[$level[$a][$b]]["y"], 35, 35, $col_node);
        imagestring($image, 5, $node[$level[$a][$b]]["x"]-7,$node[$level[$a][$b]]["y"]-7, "x".$level[$a][$b], $col_text);
        imagestring($image, 5, $node[$level[$a][$b]]["x"]-(5*strlen($nama[$level[$a][$b]])),$node[$level[$a][$b]]["y"]+20, substr($nama[$level[$a][$b]],0,10), $col_text);
    }
}
}
ImageString($image,5,30,20,"Phylogenetic Tree of ".$jml_spcs." Species",$col_text);
ImageString($image,5,30,40,"Built On ".date("D, j M Y G:i:s T"),$col_text);
ImageString($image,4,10,$size_y-20,"http://".$_SERVER['HTTP_HOST'].".$_SERVER['REQUEST_URI'],$col_text);
imagepng($image);
imagedestroy($image);
}
if($jml_spcs==2){
$image=ImageCreate (200,200);
switch($back){
    case "Black": imagecolorallocate($image, 0, 0, 0); break;
    case "Red": imagecolorallocate($image, 255, 0, 0); break;
    case "Green": imagecolorallocate($image, 0, 255, 0); break;
    case "Yellow": imagecolorallocate($image, 255, 255, 0); break;
    case "White": imagecolorallocate($image, 255, 255, 255); break;
    case "Grey": imagecolorallocate($image, 200, 200, 200); break;
    case "Soft": imagecolorallocate($image, 0, 255, 255); break;
    case "Purple": imagecolorallocate($image, 255, 0, 255); break;
    case "Blue": imagecolorallocate($image, 0, 0, 255); break;
}
switch($node){
    case "Black": $col_node=imagecolorallocate($image, 0, 0, 0); break;
    case "Red": $col_node=imagecolorallocate($image, 255, 0, 0); break;
    case "Green": $col_node=imagecolorallocate($image, 0, 255, 0); break;
}
}

```

```

    case "Yellow": $col_node=imagecolorallocate($image, 255, 255, 0); break;
    case "White": $col_node=imagecolorallocate($image,255,255,255); break;
    case "Purple": $col_node=imagecolorallocate($image, 255, 0, 255); break;
    case "Grey": $col_node=imagecolorallocate($image, 200,200, 200); break;
    case "Soft": $col_node=imagecolorallocate($image, 0, 255, 255); break;
    case "Blue": $col_node=imagecolorallocate($image, 0, 0, 255); break;
}
switch($edge){
    case "Black": $col_edge=imagecolorallocate($image, 0, 0, 0); break;
    case "Red": $col_edge=imagecolorallocate($image, 255, 0, 0); break;
    case "Green": $col_edge=imagecolorallocate($image, 0, 255, 0); break;
    case "Yellow": $col_edge=imagecolorallocate($image, 255, 255, 0); break;
    case "White":$col_edge =imagecolorallocate($image, 255,255,255);break;
    case "Purple": $col_edge=imagecolorallocate($image, 255, 0, 255); break;
    case "Grey": $col_edge=imagecolorallocate($image,200,200,200); break;
    case "Soft": $col_edge=imagecolorallocate($image, 0, 255, 255); break;
    case "Blue": $col_edge=imagecolorallocate($image, 0, 0, 255); break;
}
switch($text){
    case "Black": $col_text =imagecolorallocate($image, 0, 0, 0); break;
    case "Red":$col_text =imagecolorallocate($image, 255, 0, 0); break;
    case "Green": $col_text =imagecolorallocate($image, 0, 255, 0); break;
    case "Yellow":$col_text =imagecolorallocate($image, 255, 255, 0); break;
    case "White": $col_text =imagecolorallocate($image,255,255,255);break;
    case "Purple": $col_text=imagecolorallocate($image, 255, 0, 255); break;
    case "Grey": $col_text =imagecolorallocate($image, 200,200,200); break;
    case "Soft": $col_text=imagecolorallocate($image, 0, 255, 255); break;
    case "Blue": $col_text=imagecolorallocate($image, 0, 0, 255); break;
}
    imagelinethick ($image,50,100,150,100,$col_edge,2);
    imagefilledellipse($image, 50, 100, 35, 35, $col_node);
    imagefilledellipse($image, 150, 100, 35, 35, $col_node);
    imagestring($image, 5, 43,93, "x1", $col_text);
    imagestring($image, 5, 143,93, "x2", $col_text);
    imagestring($image, 5, 100,80, $distance[1][2], $col_text);
    header("Content-type: image/png");
    ImageString($image,1,10,10,"Phylogenetic Tree of ".$jml_spcs." Species",$col_text);
    ImageString($image,1,10,30,"Built On ".date("D, j M Y G:i:s"),$col_text);
    ImagePNG($image);
    ImageDestroy($image);}
?>

```

Lampiran 2

Data Koleksi Sekuens DNA
Milik Dr. Wellyzar Sjamsuridzal

| Nama Spesies / OTU | Sekuens DNA |
|--------------------------|--|
| Crypto/dimennae/AF075489 | taagcggagg aaaagaaact aacaaggatt cccttagtaa cggcgagtga accggaaga gctcaaattt gtaatctggc gtcctcggg cgtccgagtt gtaatctata gagtgtttt ccgcgccgga ccacgttaa gtctcctgga acggagtatc aaagagggtg acaatcccgt gcttgacgtg acgaccgtg ctatgtgata cacttcaac gactcgagtt gttgggaat gcagctctaa acgggtgta aattccatct aaggctaaat attggcgaga gaccgatagc gaacaagtac cgtgaggaa agatgaaaag cactttgaa agagagtaa acagcacgtg aaattgtaa aaggaaacg attgaagtca gtcgtgact tggattcag ccgtctcgg cgggtgattt gccttgctg ggtcaacatc agttctgacc ggcgaaaag ggctggaaga aggtggcacc ctgggtgtg ttatagcttc ctgctgata cgtcggttg gactgaggaa tgcagcttgc cgcaagccgg ggctcgccca cgttcaagct taggatgtg acataatggc tttaaacgac ccgtctgaa acacg |
| Cryp/curvatus/AF189834 | gctcaaattt gtaatctggc agtcttcgat tgtccgagtt gtaatctata gaagtgtttt ccgtgccgga ccatgtctaa gtccttggga atagggatc aaagagggtg acaatcccgt acttgatag accaccgtg ctctgtgata cactttctac gactcgagtt gttgggaat gcagctcaaa atgggtgta aattccatct aaagctaaat attggcgaga gaccgatagc gaacaagtac cgtgaggaa agatgaaaag cactttgaa agagagtaa acagtacgtg aaattgtga aaggaaacg attgaagtca gtcgtgttct ttagactcag ccggttcttc |

| | |
|-------------------------|--|
| | <p> cggctactt ctattgaacg ggtcaacatc agttttgtcc gatggataaa ggtagaagga atgtagcttc ctcggaagtg ttatagcctt ttattgcata cattgggtga gactgaggac tgcagctcgc ctttatggcc ggggttcgcc cacgttcgag cttaggatgt tgacataatg gctttaaacg acccgtctg aaacacgg </p> |
| Crypt-flavus/AF075497 | <p> taagcggagg aaaagaaact aacaaggatt cccctagtag cggcgagcga agcgggaaga gctcaaattt gtaatctggc gtcctcaggg cgtccgagtt gtaatctata gagacgtttt ccgtgccgga ccgtgtccaa gtcccttggga acagggatc aaagaggggtg acaatcccgt acttgacacg acgaccgggtg ctctgtgata cgttttctac gactcgagtt gtttgggaat gcagctcaaa atgggtggta aattccatct aaagctaaat ataggcgaga gaccgatagc gaacaagtac cgtgagggaa agatgaaaag cactttggaa agagagttaa acagtatgtg aaattgtga aagggaaacg attgaagtca gtcatgtcca ttggattcag ctggttctgc cagtgtattt cctttggacg ggtcaacatc agtttgagcc ggcggataat ggcagagggga atgtggcacc cccgggtgtg ttatagccct ttgtgcata cgtcggccca gactgaggaa tgcagctcgc ctttatggcc ggggttcgcc cacgtacgag cttaggatgt tggcataatg gctttaaacg acccgtctg aaacacg </p> |
| Crypt/luteolus/AF075482 | <p> taagcggagg aaaagaaact aacaaggatt cccctagtag cggcgagcga accgggaaga gctcaaattt aaaatctggc gtccttcggg cgtccgagtt gtaatctaca gaagtgtttt ccgtgccgga ccgtgtccaa gtcccttggga atagggatc aaagaggggtg acaatcccgt acttgacacg acaaccgggtg ctctgtgata cactttctac gactcgagtt gtttgggaat gcagctcaaa atgggtggta aattccatct aaagctaaat ataggcgaga gaccgatagc gaacaagtac cgtgagggaa agatgaaaag cactttggaa agagagttaa acagtacgtg aaattgtga aagggaaacg attgaagtca gtcgtgtctg ttggttcag ccggttctgc cgggttatta ccaacagacg ggtcaacatc agtttgagc ggtggaaaaa ggtagagggga </p> |

| | |
|----------------------------|--|
| | <p>acgtagctcc tccgggagtg ttatagccct ttattgata cactgctga gactgaggaa tgcagctcgc cttatggcc ggggttcgcc cacgttcgag cttaggatgt tgacataatg gctttaaacg acccgtcttg aaacacg</p> |
| Crypto/sp/AF075477 | <p>taagtggagg aaaagaaact aacaaggatt cccctagtaa cggcgagtg agcgggaaga gctcaaattt aaaatctggc aggctacgct tgtccgaatt gtaatctcga gaagtgtttt ccgctgtggc ctgtgtacaa gtccttggg acagggcgtc atagaggggtg agaatcccgt ccttgacaca gacaccaat gctttgtgat acacttcaa tgagtcgagt tgtttgggaa tgcagctcaa aatgggtggg aaattccatc taaagctaaa tactggcgag agaccgatag cgaacaagta ccgtgagga aagatgaaaa gcactttgga aagagagtca aacagtacgt gaaattgtg aaagggaaac gattgaagtc agtcgtcct gcctagtctc agcctttgg tgtactacta ggtcggcagg tcagcatcag ttgggagga ttaacaaggg agttaggaat gtggcaacct cggttgtgt atagcctagc ttcgattga tctgtctgga ctgaggaacg cagtgcgcc gcaaggggtg gtcttcggac acattgcac ttaggatgct gacataatgg ctttaaacga cccgtctga aaca</p> |
| Crypto/podzolicus/AF075481 | <p>taagcggagg aaaagaaact aacaaggatt cccctagtag cggcgagcga agcgggaaga gctcaaattt gaaagctggc gtccttcggg cgtccgcatt gtaatctata gaggcgtttt ctgtgtgga ccgtgtcaa gtccttggg acagggatc aaagaggggtg acaatcccgt acttgacacg acaaccagtg ctctgtgata cgtctctac gagtcgagtt gtttgggaat gcagctcaa atgggtggg aattccatct aaagctaaat atggcgaga gaccgatagc gaacaagtac cgtgagggaa agatgaaaag cactttgga agagagtaa acagtatgtg aaattgtga aagggaaacg atggaggta gtcgtgtctg tgggattcag ccgtctctg cgggtgactt cccacagacg ggtcaacatc agtttggtc ggcggataaa ggcaggagga aggtggcacc cccgggtgtg ttatagcctc ttgtgata cgctggccga gactgaggaa</p> |

| | |
|-------------------------------|--|
| | <p> tgcagctcgc cttatggcc ggggttcgcc cacgttcgag cttaggatgt tgacataatg gccttcaacg acccgtcttg aaacacgg </p> |
| Cysto/bisporidii/AF189832 | <p> taagtggagg aaaagaaact aacaaggatt cccctagtaa cggcgagtga agcgggaaga gctcaaattt aaaatctggc agtctacgat tgtccgaatt gtaatctcga gaagtgtttt ccgcgttggc ctgtgtacaa gtcccttggg acagggcgtc atagaggggtg agaatcccgt ccttgacaca gacaccaat gctttgtgat acacttcaa tgagtcgagt tgtttgggaa tgcagctcaa aatgggtggt aaattccatc taaagctaaa tactggcgag agaccgatag cgaacaagta ccgtgagggg aagatgaaaa gcactttgga aagagagtca aacagtacgt gaaattgttg aaagggaaac gattgaagtc agtcgtgcct gcctagattc agccttctgg tgtatttcta ggtcggcagg tcagcatcag ttggggggg ttaacaaggg agttaggaat gtagcaacct cgggtgtgtt atagcctagc ttcgattga tctcgctgga ctgaggaacg cagtgcgcc gcaaggttg tcttcggaca cattcgcact taggatgctg gcataatg </p> |
| Cystofi-infirmitatis/AF075505 | <p> taagtggagg aaaagaaact aacaaggatt cccctagtaa cggcgagtga agcgggaaga gctcaaattt aaaatctggc aggtacgct tgtccgaatt gtaatctcga gaagtgtttt ccgcgttggc ctgtgtacaa gtcccttggg acagggcgtc atagaggggtg agaatcccgt ccttgacaca gacaccaat gctttgtgat acacttcaa tgagtcgagt tgtttgggaa tgcagctcaa aatgggtggt aaattccatc taaagctaaa tactggcgag agaccgatag cgaacaagta ccgtgagggg aagatgaaaa gcactttgga aagagagtca aacagtacgt gaaattgttg aaagggaaac gattgaagtc agtcgtgcct gcctagattc agccttctgg tgtatttcta ggtcggcagg tcagcatcag ttggggggg ttaacaaggg agctgggaat gtggcaacct tcgggtgtgt atagcccagt ttcgattga tctcgctgga ctgaggaacg cagtgcgcc gcaaggttg gtcttcggac acattgcac ttaggatgct ggcataatg ctttaaacga cccgtcttga aacac </p> |

| | |
|-----------------------------------|--|
| <p>crypt-heveanensis/af406890</p> | <p>ctaacaagga ttccttagt aacggcgagc gaaccgggaa gagctcaaat ttgaaatctg gcgctctcag ggcgtccgag ttgtaatcta tagaggcgtt tccgtgccg gaccgtgtcc aagtcccttg gaacagggta tcaaagaggg tgacaatccc gtacttgaca cgactaccgg tgctctgtga tacgtcttct acgagtcgag ttgtttggga atgcagctca aaacgggtgg taaactccat ctaaagctaa atattggtgg gagaccgata gcgaacaagt accgtgaggg aaagatgaaa agcactttgg aaagagagtt aaacagtatg tgaaattgtt gaaagggaaa cgattgaagt cagtcgtgtc cagaggactc agccggttct gccgggtctac ttctctgga cgggtcaaca tcggttctgg acggcggata aggacgggag gaaggtggca cccctcgtgg tgtgtatag cctcctgtcg catacgtcga ccgggaccga ggaacgcagc tcgctttat ggccgggggt cgcccacgta cgagcttagg atgttgacgt aatggcttta aacgac</p> |
| <p>crypt/laurentiaj555467</p> | <p>tttttgtt gacctcaaat caggtagggc taccgctga acttaagcat atcaataagc ggaggaaaag aaactaaca ggattcccct agtaacggcg agtgaaccgg gaagagctca aattgaaat ctggcgtgct cagtcgtcc gagttgtaat ctatagagtc gtttccgtg ccggactgtg tccaagtccc ttggaacagg gtatcaaaga gggtgataat cccgtacttg acacaatgac cggtgctctg tgatacgtct tctacgagtc gagttgtttg ggaatgcagc tcaaatggg tggtagttc catctaaagc taaataatgg cgagagaccg atagcgaaca agtaccgtga gggaaagatg aaaagcactt tggaagaga gtaaacagt acgtgaaatt gttaaaggga aaacgattga agtcagtcgt gactgagagg ctacgccggt tctgccggtg tattcccctc agtcgggtca acatcagttt tgttcggtgg ataagggcag ttggaagggtg gcaccctcgg gtgtgttata gccagctgtc gcatacatcg gatgagactg aggaatgcag ctcgccttta tggccgggggt tcgcccacgt tcgagcttag gatgttgaca taatggcttt aaacgaccgg tcttgaaaca cggaccaagg agtctaacat atctgcgagt gtttgggtgg</p> |

