BAB V

KESIMPULAN

5.1 Kesimpulan

1. Sarana e-Learning ini hanya sebagai alat Bantu perkuliahan, tidak untuk mengganti kuliah /kelas tatap muka.
2. Sistem yang dikembangkan adalah Moodle yang dimodifikasi berbasis web dengan dukungan PHP programming dan database MySQL.
3. Untuk saat ini, fasilitas *Quiz* yang ada di situs, hanya untuk uji coba kemampuan mahasiswa atau keaktifan mahasiswa dalam memanfaatkan situs e-learning Universitas Darma Persada, bukan untuk pengambilan nilai *Quiz* pada perkuliahan.
4. Beberapa alternatif paket software implementasi e-Learning yang dicoba untuk dikembangkan yaitu dari moodle merupakan paket software yang diproduksi untuk kegiatan belajar berbasis internet dan website atau tempat belajar dinamis dengan menggunakan model berorientasi objek.
5. Hasil dari pengembangan e-Learning Universitas Darma Persada yang telah dibuat memuat informasi kursus mengenai seputar aktivitas belajar-mengajar dan kegiatan kampus atau informasi umum lainnya yang dibutuhkan oleh user. Dari pihak Akademik maupun user dapat mendapatkan informasi ataupun memberikan informasi sehingga e-learning ini dapat menjadi solusi permasalahan dibidang informasi. Dengan telah dikembangkannya e-Learning ini diharapkan mahasiswa, dosen maupun tamu mendapat kemudahan dalam

memperoleh informasi dengan cepat dan mudah serta dapat dimanfaatkan dengan sebaik-baiknya. Sekalipun sarana ini masih memiliki kekurangan namun dapat dijadikan langkah awal yang baik untuk pengembangan selanjutnya.

5.2 Saran

Penyajian Informasi berupa aplikasi halaman web pengembangan e-Learning Universitas Darma Persada yang dibuat oleh penulis masih jauh dari sempurna, oleh karena itu banyak kemungkinan pengembangan yang dilakukan. Seperti halnya :

1. Sarana e-learning ini sangat positif sekali untuk dikembangkan dengan layanan yang lain seperti media komunikasi dalam format video atau voice.
2. Dengan adanya situs e-learning ini semoga dapat memberikan motivasi bagi mahasiswa agar lebih kreatif dalam hal pembelajaran atau perkuliahan,serta dapat mengembangkan e-learning yang sudah ada menjadi lebih baik lagi sesuai dengan kebutuhan mahasiswa maupun kampus sebagai lembaga pendidikan.

DAFTAR PUSTAKA

1. Fathansyah, *Basis Data*, Informatika, 2004
2. Jogiyanto Hartono,MBA,Ph.D., *Pengenalan komputer*. ANDi OFFSET, Yogyakarta, (2002).
3. Kadir Abdul, *Pemrograman Web Mencakup : HTML, CSS, JavaScript & PHP*, Penerbit Andi Yogyakarta, 2003
4. Kadir Abdul, *Pengenalan Sistem Informasi*. ANDI OFFSET, Yogyakarta, (2003).

Online Readings:

5. *Analisa & Perancangan Berorientasi obyek*, www. Ilmu Komputer.com, 2006 (*softcopy*)
6. Anonkuncoro@yahoo.com, *Pelatihan Penggunaan Aplikasi e-Learning Moodle 1*, 2006 (*softcopy*)
7. Harry B.Santoso, *e-Learning Belajar kapan saja, dimana saja*, www.Ilmu Komputer.com, 2006 (*softcopy*)
8. *Konsep dasar Berorientasi Obyek*, www.Ilmu Komputer.com, 2006 (*softcopy*)
9. Melfachrozi M, Penggunaan Aplikasi e-learning (moodle), www. Ilmu Komputer.com, 2006 (*softcopy*)
10. Romi Satria wahono, *Pengantar e-Learning dan Pengembangannya*, www.Ilmu Komputer.com ,2005 (*softcopy*)
11. http://www.moodle.org
12. http://www.e-learningcenter.co.uk

LAMPIRAN

Source code

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) {  //reloads the window if Nav4 resized
  if (init==true ) with (navigator) {if ((appName== "Netscape")&&(parseInt(app Version)==4)) {
    document.MM_pgW=inner Width; document.MM_pgH=inner Height;
onresiz e=MM_reloadPage; }}
  else if (inner Width!=document.MM_pgW | inner Height!=document.MM_pgH) location.reload();
}
MM_reloadPage(true);
//-->
</script>
<style type="text/css">
<!--
.style 1 {
        font-size: 36px;
        font-family: Georgia, "Times New Roman", Times, serif;
}
.style2 {color: #0000FF}
.style4 {color: #CC6666}
.style5 {color: #CC3366}
.style7 {color: #CC3399}
.style9 {color: #9900FF}
.style 10 {color: #00FF00}
.style12 {font-siz e: 18px}
.style 14 {color: #0099FF}
-->
</style>
</head>

<body>
<table width="100% %"  border="0" align="center">
  <tr>
    <td width="20%" height="312" rowspan="3" bgcolor="#FFFFFF"><div align="center">
      <p align="left"><img src="images/gerak_01.gif" width="113" height="540"> </p>
      </div>
      </td>
    <td width="60% %" bgcolor="#FFFFFF">      <div align="center">
      <p>
        <object classid= "clsid:166B 1BCA-3F9C-11CF-8075-444553540000"
codebase="http://download.macromedia.com/pub/shock wave/cabs/director/sw.cab#version=8,5,0,
0" width="600" height ="100">
          <param name="src" value="images/banner1.swf">
          <embed src="images/banner1.swf"
pluginspage="http://www.macromedia.com/shock wave/download/" width="600"
height="100"></embed>
        </object>
      </p>
```

```
  </div></td>
  <td width="20%" rowspan="3" bgcolor="#FFFFFF"><div align="right"><img
src="images/gerak_01.gif" width="113" height="540"></div></td>
 </tr>
 <tr>
  <td bgcolor="#FFFFFF"><img src="images/gdg.JPG" width="600" height="328"></td>
 </tr>
 <tr>
  <td bgcolor="#FFFFFF"><div align="center"><span class="style1"><span class="style2"><a
href="moodle/index.php">e</a></span><a href="moodle/index.php"><span
class="style10">L</span><span class="style4">i</span><span class="style5">s</span><span
class="style9">a</span><span class="style7">d</span><span
class="style14">a</span></a></span><br>
    <span class="style12">Komunitas e-learning Mahasiswa Universitas Darma Persada
</span></div></td>
 </tr>
 <tr bgcolor="#CCCCCC">
  <td height="21" colspan="3"><marquee>
   <strong>Copyright &copy; 2007 Devi Astianingsih </strong>
  </marquee></td>
 </tr>
</table>
<div align="center"></div>
</body>
</html>
```

INDEX.PHP PADA MOODLE

```
<?php
  if (!file_exists('./config.php')) {
    header('Location: install.php');
    die;
  }

  require_once('config.php');
  require_once($CFG->dirroot .'/course/lib.php');
  require_once($CFG->dirroot .'/lib/blocklib.php');

  if (!empty($THEME->customcorners)){
    require_once($CFG->dirroot.'/lib/custom_corners_lib.php');
  }

  if (empty($SITE)) {
    redirect($CFG->wwwroot .'/'. $CFG->admin .'/index.php');
  }

  // Bounds for block widths
  // more flexible for theme designers taken from theme config.php
  $lmin = (empty($THEME->block_l_min_width)) ? 100 : $THEME->block_l_min_width;
  $lmax = (empty($THEME->block_l_max_width)) ? 210 : $THEME->block_l_max_width;
  $rmin = (empty($THEME->block_r_min_width)) ? 100 : $THEME->block_r_min_width;
  $rmax = (empty($THEME->block_r_max_width)) ? 210 : $THEME->block_r_max_width;

  define('BLOCK_L_MIN_WIDTH', $lmin);
```

```
    define('BLOCK_L_MAX_WIDTH', $lmax);
    define('BLOCK_R_MIN_WIDTH', $rmin);
    define('BLOCK_R_MAX_WIDTH', $rmax);

    // check if major upgrade needed - also present in login/index.php
    if ((int)$CFG->version < 2006101100) { //1.7 or older
        @require_logout();
        redirect("$CFG->wwwroot/$CFG->admin/");
    }

    if ($CFG->forcelogin) {
        require_login();
    }

    if ($CFG->rolesactive) { // if already using roles system
        if (has_capability('moodle/site:config', get_context_instance(CONTEXT_SYSTEM))) {
            if (moodle_needs_upgrading()) {
                redirect($CFG->wwwroot .'/'. $CFG->admin .'/index.php');
            }
        } else if (!empty($CFG->mymoodleredirect)){    // Redirect logged-in users to My Moodle
overview if required
            if (isloggedin() && $USER->username != 'guest') {
                redirect($CFG->wwwroot .'/my/index.php');
            }
        }
    } else { // if upgrading from 1.6 or below
        if (isadmin() && moodle_needs_upgrading()){
            redirect($CFG->wwwroot .'/'. $CFG->admin .'/index.php');
        }
    }


    if (get_moodle_cookie()==''){
        set_moodle_cookie('nobody');   // To help search for cookies on login page
    }

    if (!empty($USER->id)) {
        add_to_log(SITEID, 'course', 'view', 'view.php?id='.SITEID, SITEID);
    }

    if (empty($CFG->langmenu)) {
        $langmenu = '';
    } else {
        $currlang = current_language();
        $langs = get_list_of_languages();
        $langlabel = '<span class="accesshide">'.get_string('language').':</span>';
        $langmenu = popup_form($CFG->wwwroot .'/index.php?lang=', $langs, 'chooselang',
$currlang, '', '', '', true, 'self', $langlabel);
    }

    $PAGE      = page_create_object(PAGE_COURSE_VIEW, SITEID);
    $pageblocks = blocks_setup($PAGE);
    $editing   = $PAGE->user_is_editing();
    $preferred_width_left  = bounded_number(BLOCK_L_MIN_WIDTH,
blocks_preferred_width($pageblocks[BLOCK_POS_LEFT]),
```

```
                                BLOCK_L_MAX_WIDTH);
    $preferred_width_right = bounded_number(BLOCK_R_MIN_WIDTH,
blocks_preferred_width($pageblocks[BLOCK_POS_RIGHT]),
                                BLOCK_R_MAX_WIDTH);

    print_header($SITE->fullname, $SITE->fullname, 'home', '',
            '<meta name="description" content="'. s(strip_tags($SITE->summary)).'" />',
            true, '', user_login_string($SITE).$langmenu);

?>


<table id="layout-table" summary="layout">
  <tr>
  <?php
    $lt = (empty($THEME->layouttable)) ? array('left', 'middle', 'right') : $THEME->layouttable;
    foreach ($lt as $column) {
        switch ($column) {
            case 'left':
    if (blocks_have_content($pageblocks, BLOCK_POS_LEFT) || $editing) {
        echo '<td style="width: '.$preferred_width_left.'px;" id="left-column">';
        if (!empty($THEME->customcorners)) print_custom_corners_start();
        blocks_print_group($PAGE, $pageblocks, BLOCK_POS_LEFT);
        if (!empty($THEME->customcorners)) print_custom_corners_end();
        echo '</td>';
    }
            break;
            case 'middle':
    echo '<td id="middle-column">';

    if (!empty($THEME->customcorners)) print_custom_corners_start();

/// Print Section
    if ($SITE->numsections > 0 ) {

        if (! $section = get_record('course_sections', 'course', $SITE->id, 'section', 1)) {
            delete_records('course_sections', 'course', $SITE->id, 'section', 1); // Just in case
            $section->course = $SITE->id;
            $section->section = 1;
            $section->summary = '';
            $section->sequence = '';
            $section->visible = 1;
            $section->id = insert_record('course_sections', $section);
        }

        if (!empty($section->sequence) or !empty($section->summary) or $editing) {
            print_box_start('generalbox sitetopic');

            /// If currently moving a file then show the current clipboard
            if (ismoving($SITE->id)) {
                $stractivityclipboard = strip_tags(get_string('activityclipboard', '', addslashes($USER-
>activitycopyname')));
                echo '<p><font size="2">';
```

```
            echo "$stractivityclipboard  (<a
href=\"course/mod.php?cancelcopy=true&amp;sesskey=$USER->sesskey\">". get_string('cancel')
.'</a>)';
            echo '</font></p>';
        }

        $options = NULL;
        $options->noclean = true;
        echo format_text($section->summary, FORMAT_HTML, $options);

        if ($editing) {
            $streditsummary = get_string('editsummary');
            echo "<a title=\"$streditsummary\"".
                " href=\"course/editsection.php?id=$section->id\"><img src=\"$CFG-
>pixpath/t/edit.gif\"".
                " class=\"iconsmall\" alt=\"$streditsummary\" /></a><br /><br />";
        }

        get_all_mods($SITE->id, $mods, $modnames, $modnamesplural, $modnamesused);
        print_section($SITE, $section, $mods, $modnamesused, true);

        if ($editing) {
            print_section_add_menus($SITE, $section->section, $modnames);
        }
        print_box_end();
    }
}

if (isloggedin() and !isguest() and isset($CFG->frontpageloggedin)) {
    $frontpagelayout = $CFG->frontpageloggedin;
} else {
    $frontpagelayout = $CFG->frontpage;
}

foreach (explode(',',$frontpagelayout) as $v) {
    switch ($v){     /// Display the main part of the front page.
        case strval(FRONTPAGENEWS):
            if ($SITE->newsitems) { // Print forums only when needed
                require_once($CFG->dirroot .'/mod/forum/lib.php');

                if (! $newsforum = forum_get_course_forum($SITE->id, 'news')) {
                    error('Could not find or create a main news forum for the site');
                }

                if (!empty($USER->id)) {
                    $SESSION->fromdiscussion = $CFG->wwwroot;
                    if (forum_is_subscribed($USER->id, $newsforum->id)) {
                        $subtext = get_string('unsubscribe', 'forum');
                    } else {
                        $subtext = get_string('subscribe', 'forum');
                    }
                    print_heading_block($newsforum->name);
                    echo '<div class="subscribelink"><a
href="mod/forum/subscribe.php?id='.$newsforum->id.'">'.$subtext.'</a></div>';
                } else {
```

```
                print_heading_block($newsforum->name);
            }

            forum_print_latest_discussions($SITE, $newsforum, $SITE->newsitems, 'plain',
'p.modified DESC');
        }
        break;

        case FRONTPAGECOURSELIST:

        if (isloggedin() and !has_capability('moodle/site:config',
get_context_instance(CONTEXT_SYSTEM)) and !isguest() and empty($CFG-
>disablemycourses)) {
            print_heading_block(get_string('mycourses'));
            print_my_moodle();
        } else if ((!has_capability('moodle/site:config',
get_context_instance(CONTEXT_SYSTEM)) and !isguest()) or (count_records('course') <=
FRONTPAGECOURSELIMIT)) {
            // admin should not see list of courses when there are too many of them
            print_heading_block(get_string('availablecourses'));
            print_courses(0, true);
        }
        break;

        case FRONTPAGECATEGORYNAMES:

            print_heading_block(get_string('categories'));
            print_box_start('generalbox categorybox');
            print_whole_category_list(NULL, NULL, NULL, -1, false);
            print_box_end();
            print_course_search('', false, 'short');
        break;

        case FRONTPAGECATEGORYCOMBO:

            print_heading_block(get_string('categories'));
            print_box_start('generalbox categorybox');
            print_whole_category_list(NULL, NULL, NULL, -1, true);
            print_box_end();
            print_course_search('', false, 'short');
        break;

        case FRONTPAGETOPICONLY:    // Do nothing!!  :-)
        break;

    }
    echo '<br />';
}

if (!empty($THEME->customcorners)) print_custom_corners_end();

echo '</td>';
    break;
    case 'right':
// The right column
```

```
    if (blocks_have_content($pageblocks, BLOCK_POS_RIGHT) || $editing ||
has_capability('moodle/course:update', get_context_instance(CONTEXT_COURSE, SITEID))) {
        echo '<td style="width: '.$preferred_width_right.'px;" id="right-column">';
        if (!empty($THEME->customcorners)) print_custom_corners_start();
        if (has_capability('moodle/course:update', get_context_instance(CONTEXT_COURSE,
SITEID))) {
            echo '<div style="text-align:center">'.update_course_icon($SITE->id).'</div>';
            echo '<br />';
        }
        blocks_print_group($PAGE, $pageblocks, BLOCK_POS_RIGHT);
        if (!empty($THEME->customcorners)) print_custom_corners_end();
        echo '</td>';
    }
        break;
    }
  }
?>

 </tr>
</table>

<?php
  print_footer('home');    // Please do not modify this line
?>
```

CONFIG.PHP

```
<?php  /// Moodle Configuration File

unset($CFG);

$CFG->dbtype    = 'mysql';
$CFG->dbhost    = 'localhost';
$CFG->dbname    = 'db';
$CFG->dbuser    = 'root';
$CFG->dbpass    = '';
$CFG->dbpersist =  false;
$CFG->prefix    = 'mdl_';

$CFG->wwwroot   = 'http://localhost/web/moodle';
$CFG->dirroot   = 'D:\xampp\htdocs\web\moodle';
$CFG->dataroot  = 'D:\xampp/moodledata';
$CFG->admin     = 'admin';

$CFG->directorypermissions = 00777;  // try 02777 on a server in Safe Mode

require_once("$CFG->dirroot/lib/setup.php");
// MAKE SURE WHEN YOU EDIT THIS FILE THAT THERE ARE NO SPACES, BLANK
LINES,
// RETURNS, OR ANYTHING ELSE AFTER THE TWO CHARACTERS ON THE NEXT
LINE.
?>
```

FILE.PHP

```
<?php // $Id: file.php,v 1.45 2007/04/02 16:13:41 skodak Exp $
    // This script fetches files from the dataroot directory
```

```
    // Syntax:      file.php/courseid/dir/dir/dir/filename.ext
    //              file.php/courseid/dir/dir/dir/filename.ext?forcedownload=1 (download instead of
inline)
    //              file.php/courseid/dir (returns index.html from dir)
    // Workaround:  file.php?file=/courseid/dir/dir/dir/filename.ext
    // Test:        file.php/testslasharguments


    //TODO: Blog attachments do not have access control implemented - anybody can read them!
    //      It might be better to move the code to separate file because the access
    //      control is quite complex - see blog/index.php

  require_once('config.php');
  require_once('lib/filelib.php');

  if (empty($CFG->filelifetime)) {
    $lifetime = 86400;     // Seconds for files to remain in caches
  } else {
    $lifetime = $CFG->filelifetime;
  }

  // disable moodle specific debug messages
  disable_debugging();

  $relativepath = get_file_argument('file.php');
  $forcedownload = optional_param('forcedownload', 0, PARAM_BOOL);

  // relative path must start with '/', because of backup/restore!!!
  if (!$relativepath) {
    error('No valid arguments supplied or incorrect server configuration');
  } else if ($relativepath{0} != '/') {
    error('No valid arguments supplied, path does not start with slash!');
  }

  $pathname = $CFG->dataroot.$relativepath;

  // extract relative path components
  $args = explode('/', trim($relativepath, '/'));
  if (count($args) == 0) { // always at least courseid, may search for index.html in course root
    error('No valid arguments supplied');
  }

  // security: limit access to existing course subdirectories
  if (($args[0] != 'blog') and (!$course = get_record_sql("SELECT * FROM {$CFG-
>prefix}course WHERE id='".(int)$args[0]."'"))) {
    error('Invalid course ID');
  }

  // security: prevent access to "000" or "1 something" directories
  // hack for blogs, needs proper security check too
  if (($args[0] != 'blog') and ($args[0] != $course->id)) {
    error('Invalid course ID');
  }

  // security: login to course if necessary
```

```
if ($args[0] == 'blog') {
    if (empty($CFG->bloglevel)) {
        error('Blogging is disabled!');
    } else if ($CFG->bloglevel < BLOG_GLOBAL_LEVEL) {
        require_login();
    } else if ($CFG->forcelogin) {
        require_login();
    }
} else if ($course->id != SITEID) {
    require_login($course->id);
} else if ($CFG->forcelogin) {
    if(!empty($CFG->sitepolicy)
        and ($CFG->sitepolicy == $CFG->wwwroot.'/file.php'.$relativepath
            or $CFG->sitepolicy == $CFG->wwwroot.'/file.php?file='.$relativepath)) {
        //do not require login for policy file
    } else {
        require_login();
    }
}

// security: only editing teachers can access backups
if ((count($args) >= 2) and (strtolower($args[1]) == 'backupdata')) {
    if (!has_capability('moodle/site:backup', get_context_instance(CONTEXT_COURSE,
$course->id))) {
        error('Access not allowed');
    } else {
        $lifetime = 0; //disable browser caching for backups
    }
}

if (is_dir($pathname)) {
    if (file_exists($pathname.'/index.html')){
        $pathname = rtrim($pathname, '/').'/index.html';
        $args[] = 'index.html';
    } else if (file_exists($pathname.'/index.htm')) {
        $pathname = rtrim($pathname, '/').'/index.htm';
        $args[] = 'index.htm';
    } else if (file_exists($pathname.'/Default.htm')) {
        $pathname = rtrim($pathname, '/').'/Default.htm';
        $args[] = 'Default.htm';
    } else {
        // security: do not return directory node!
        not_found($course->id);
    }
}

// security: teachers can view all assignments, students only their own
if ((count($args) >= 3)
    and (strtolower($args[1]) == 'moddata')
    and (strtolower($args[2]) == 'assignment')) {

    $lifetime = 0;  // do not cache assignments, students may reupload them
    if (!has_capability('mod/assignment:grade', get_context_instance(CONTEXT_COURSE,
$course->id))
        and $args[4] != $USER->id) {
```

```
            error('Access not allowed');
        }
    }

    // security: force download of all attachments submitted by students
    if ((count($args)>= 3)
        and (strtolower($args[1]) == 'moddata')
        and ((strtolower($args[2]) == 'forum')
            or (strtolower($args[2]) == 'assignment')
            or (strtolower($args[2]) == 'data')
            or (strtolower($args[2]) == 'glossary')
            or (strtolower($args[2]) == 'wiki')
            or (strtolower($args[2])== 'exercise)
            or (strtolower($args[2]) == 'workshop')
            )) {
        $forcedownload = 1; // force download of all attachments
    }
    if ($args[0] == 'blog') {
        $forcedownload = 1; // force download of all attachments
    }

    // security: some protection of hidden resource files
    // warning: it may break backwards compatibility
    if (!empty($CFG->preventaccesstohiddenfiles)
        and (count($args)>= 2)
        and (!(strtolower($args[1]) == 'moddata' and strtolower($args[2])!= 'resource')) // do not
block files from other modules!
        and (!has_capability('moodle/course:viewhiddenactivities',
get_context_instance(CONTEXT_COURSE, $course->id)))){

        $rargs = $args;
        array_shift($rargs);
        $reference = implode('/', $rargs);

        $sql = "SELECT COUNT(r.id)" .
               "FROM {$CFG->prefix}resource r," .
                   "{$CFG->prefix}course_modules cm, " .
                   "{$CFG->prefix}modules m " .
               "WHERE r.course    = '{$course->id}'" .
                "AND m.name      = 'resource' " .
                "AND cm.module   = m.id" .
                "AND cm.instance = r.id" .
                "AND cm.visible  = 0 " .
                "AND r.type      = 'file' " .
                "AND r.reference = '{$reference}'";
        if (count_records_sql($sql)) {
            error('Access not allowed');
        }
    }

    // check that file exists
    if (!file_exists($pathname)) {
        not_found($course->id);
    }
```

```
    // ============================================================
    // finally send the file
    // ============================================================
    session_write_close(); // unlock session during fileserving
    $filename = $args[count($args)-1];
    send_file($pathname, $filename, $lifetime, $CFG->filteruploadedfiles, false, $forcedownload);

    function not_found($courseid) {
        global $CFG;
        header('HTTP/1.0 404 not found');
        error(get_string('filenotfound', 'error'), $CFG->wwwroot.'/course/view.php?id='.$courseid);
//this is not displayed on IIS??
    }
?>
```

INDEX.PHP ADMIN

```
<?php // $Id: index.php,v 1.278 2007/07/06 17:54:54 stronk7 Exp $

/// Check that config.php exists, if not then call the install script
    if (!file_exists('../config.php')) {
        header('Location: ../install.php');
        die;
    }

/// Check that PHP is of a sufficient version
/// Moved here because older versions do not allow while(@ob_end_clean());
    if (version_compare(phpversion(), "4.3.0")<0) {
        $phpversion = phpversion();
        echo "Sorry, Moodle requires PHP 4.3.0 or later (currently using version $phpversion)";
        die;
    }

/// Turn off time limits and try to flush everything all the time, sometimes upgrades can be slow.

    @set_time_limit(0);
    @ob_implicit_flush(true);
    while(@ob_end_clean()); // ob_end_flush prevents sending of headers


    require_once('../config.php');
    require_once($CFG->libdir.'/adminlib.php');  // Contains various admin-only functions
    require_once($CFG->libdir.'/dmllib.php'); // Install/upgrade related db functions

    $id             = optional_param('id', '', PARAM_ALPHANUM);
    $confirmupgrade = optional_param('confirmupgrade', 0, PARAM_BOOL);
    $confirmrelease = optional_param('confirmrelease', 0, PARAM_BOOL);
    $agreelicense   = optional_param('agreelicense', 0, PARAM_BOOL);
    $autopilot      = optional_param('autopilot', 0, PARAM_BOOL);
    $ignoreupgradewarning = optional_param('ignoreupgradewarning', 0, PARAM_BOOL);

/// check upgrade status first
    if ($ignoreupgradewarning and !empty($_SESSION['upgraderunning'])) {
        $_SESSION['upgraderunning'] = 0;
    }
```

```
    upgrade_check_running("Upgrade already running in this session, please wait!<br />Click on
the exclamation marks to ignore this warning (<a
href=\"index.php?ignoreupgradewarning=1\">!!!</a>).", 10);

/// set install/upgrade autocontinue session flag
    if ($autopilot) {
        $_SESSION['installautopilot'] = $autopilot;
    }

/// Check some PHP server settings

    $documentationlink = '<a href="http://docs.moodle.org/en/Installation">Installation docs</a>';

    if (ini_get_bool('session.auto_start')) {
        error("The PHP server variable 'session.auto_start' should be Off - $documentationlink");
    }

    if (ini_get_bool('magic_quotes_runtime')) {
        error("The PHP server variable 'magic_quotes_runtime' should be Off -
$documentationlink");
    }

    if (!ini_get_bool('file_uploads')) {
        error("The PHP server variable 'file_uploads' is not turned On - $documentation link");
    }

    if (empty($CFG->prefix) && $CFG->dbfamily != 'mysql') { //Enforce prefixes for everybody
but mysql
        error('$CFG->prefix can\'t be empty for your target DB (' . $CFG->dbtype.')');
    }

    if ($CFG->dbfamily == 'oracle' && strlen($CFG->prefix) > 2) { //Max prefix length for Oracle
is 2cc
        error('$CFG->prefix maximum allowed length for Oracle DBs is 2cc.');
    }

/// Check that config.php has been edited

    if ($CFG->wwwroot == "http://example.com/moodle") {
        error( "Moodle has not been configured yet.  You need to edit config.php first.");
    }


/// Check settings in config.php

    $dirroot = dirname(realpath("./index.php"));
    if (!empty($dirroot) and $dirroot != $CFG->dirroot) {
        error("Please fix your settings in config.php:
            <p>You have:
            <p>\$CFG->dirroot = \"" .addslashes($CFG->dirroot)."\";
            <p>but it should be:
            <p>\$CFG->dirroot = \"".addslashes($dirroot)."\";",
            "./");
    }
```

```
/// Set some necessary variables during set-up to avoid PHP warnings later on this page
  if (!isset($CFG->framename)) {
    $CFG->framename ="_top";
  }
  if (!isset($CFG->release)) {
    $CFG->release = "";
  }
  if (!isset($CFG->version)) {
    $CFG->version = "";
  }

  if (is_readable("$CFG->dirroot/version.php")) {
    include_once("$CFG->dirroot/version.php");          # defines $version
  }

  if (! $version or !$release) {
    error('Main version.php was not readable or specified');# without version, stop
  }

/// Check if the main tables have been installed yet or not.

  if (! $tables = $db->Metatables()){    // No tables yet at all.
    $maintables = false;

  } else {                          //Check for missing main tables
    $maintables = true;
    $mtables = array("config", "course", "course_categories", "course_modules",
                "course_sections", "log", "log_display", "modules",
                "user");
    foreach ($mtables as $mtable) {
      if (!in_array($CFG->prefix.$mtable, $tables)) {
        $maintables = false;
        break;
      }
    }
  }
  if (! $maintables) {
  /// hide errors from headers in case debug enabled in config.php
    $origdebug = $CFG->debug;
    $CFG->debug = DEBUG_MINIMAL;
    error_reporting($CFG->debug);
    if (empty($agreelicense)) {
      $strlicense = get_string('license');
      print_header( $strlicense, $strlicense, $strlicense, "","", false, " ", " ");
      print_heading("<a href=\"http://moodle.org\">Moodle</a> - Modular Object-Oriented
Dynamic Learning Environment");
      print_heading(get_string('copyrightnotice'));
      print_box(text_to_html(get_string('gpl')), 'copyrightnotice');
      echo "<br />";
      notice_yesno(get_string('doyouagree'), "index.php?agreelicense=1",
                                "http://docs.moodle.org/en/License");
      print_footer('none');
      exit;
    }
    if (empty($confirmrelease)) {
```

```
        $strcurrentrelease = get_string("currentrelease");
        printheader($strcurrentrelease, $strcurrentrelease, $strcurrentrelease,"","", false,
" ", " ");
        print_heading("Moodle $release");
        print_box(get_string('releasenoteslink', 'admin', 'http://docs.moodle.org/en/Release_Notes'),
'generalbox boxaligncenter boxwidthwide');
        echo '<form action="index.php"><div>';
        echo '<input type="hidden" name="agreelicense" value="1" />';
        echo '<input type="hidden" name="confirmrelease" value="1" />';
        echo '</div>';
        echo '<div class="continuebutton"><input name="autopilot" id="autopilot"
type="checkbox" value="1" /><label for="autopilot">'.get_string('unattendedoperation',
'admin').'</label>';
        echo '<br /><br /><input type="submit" value="'.get_string('continue').'" /></div>';
        echo '</form>';
        printfooter('none');
        die;
    }

    $strdatabasesetup    = get_string("databasesetup");
    $strdatabasesuccess  = get_string("databasesuccess");
    print_header($strdatabasesetup, $strdatabasesetup, $strdatabasesetup,
                 "", upgrade_get_javascript(), false, " ", " ");
  /// return to original debugging level
    $CFG->debug = $origdebug;
    error_reporting($CFG->debug);
    upgrade_log_start();
    $db->debug = true;

  /// Both old .sql files and new install.xml are supported
  /// But we prioritise install.xml (XMLDB) if present

    change_db_encoding(); // first try to change db encoding to utf8
    if (!setup_is_unicodedb()) {
      // If could not convert successfully, throw error, and prevent installation
      print_error('unicoderequired', 'admin');
    }

    $status = false;
    if (file_exists("$CFG->libdir/db/install.xml")){
      $status = install_from_xmldb_file("$CFG->libdir/db/install.xml"); //New method
    } else if (file_exists("$CFG->libdir/db/$CFG->dbtype.sql")) {
      $status = modify_database("$CFG->libdir/db/$CFG->dbtype.sql"); //Old method
    } else {
      error("Error: Your database ($CFG->dbtype) is not yet fully supported by Moodle or
install.xml is not present.  See the lib/db directory.");
    }

    // all new installs are in unicode - keep for backwards compatibility and 1.8 upgrade checks
    set_config('unicodedb', 1);

  /// Continue with the instalation
    $db->debug = false;
    if ($status) {
      //ugly hack - install new groups: MDL-9217
```

```
        require_once("$CFG->dirroot/group/db/upgrade.php");
        install_group_db();

        // Install the roles system.
        moodle_install_roles();
        set_config('statsrolesupgraded',time());

        // install core event handlers
        events_update_definition();

        // Write default settings unconditionally (i.e. even if a setting is already set, overwrite it)
        // (this should only have any effect during initial install).
        $adminroot = admin_get_root();
        $adminroot->prune('backups'); // backup settings table not created yet
        apply_default_settings($adminroot);

        /// This is used to handle any settings that must exist in $CFG but which do not exist in
        /// admin_get_root()/$ADMIN as admin_setting objects (there are some exceptions).
        apply_default_exception_settings(array('alternateloginurl' =>'',
                                               'auth' => 'email',
                                               'auth_pop3mailbox' => 'INBOX',
                                               'changepassword' =>'',
                                               'enrol' => 'manual',
                                               'enrol_plugins_enabled' => 'manual',
                                               'guestloginbutton' => 1,
                                               'registerauth' => 'email',
                                               'style' => 'default',
                                               'template' => 'default',
                                               'theme' => 'standardwhite',
                                               'filter_multilang_converted' => 1));

        notify($strdatabasesuccess, "green");
        require_once $CFG->dirroot.'/mnet/lib.php';
    } else {
        error("Error: Main databases NOT set up successfully");
    }
    print_continue('index.php');
    print_footer('none');
    die;
}


/// Check version of Moodle code on disk compared with database
/// and upgrade if possible.

if (file_exists("$CFG->dirroot/lib/db/$CFG->dbtype.php")){
    include_once("$CFG->dirroot/lib/db/$CFG->dbtype.php");  # defines old upgrades
}
if (file_exists("$CFG->dirroot/lib/db/upgrade.php")) {
    include_once("$CFG->dirroot/lib/db/upgrade.php");  # defines new upgrades
}

$stradministration = get_string("administration");

if ($CFG->version) {
```

```
    if ($version > $CFG->version) {  // upgrade

    /// If the database is not already Unicode then we do not allow upgrading
    /// Instead, we print an error telling them to upgrade to 1.7 first.  MDL-6857
        if (empty($CFG->unicodedb)) {
            print_error('unicodeupgradeerror', 'error', '', $version);
        }

        $a->oldversion = "$CFG->release ($CFG->version)";
        $a->newversion = "$release ($version)";
        $strdatabasechecking = get_string("databasechecking","", $a);

        // hide errors from headers in case debug is enabled
        $origdebug = $CFG->debug;
        $CFG->debug = DEBUG_MINIMAL;
        error_reporting($CFG->debug);

        // logout in case we are upgrading from pre 1.7 version - prevention of weird session
problems
        if ($CFG->version < 2006050600) {
            require_logout();
        }

        if (empty($confirmupgrade)) {
            print_header($strdatabasechecking, $stradministration, $strdatabasechecking,
                 "","", false, " ", " ");

            notice_yesno(get_string('upgradesure', 'admin', $a->newversion),
'index.php?confirmupgrade=1', 'index.php');
            exit;

        } else if (empty($confirmrelease)){
            $strcurrentrelease = get_string("currentrelease");
            print_header($strcurrentrelease, $strcurrentrelease,"","", false,
" ", " ");
            print_heading("Moodle $release");
            print_box(get_string('releasenoteslink', 'admin',
'http://docs.moodle.org/en/Release_Notes'));

            require_once($CFG->libdir.'/environmentlib.php');
            print_heading(get_string('environment', 'admin'));
            if (!check_moodle_environment($release, $environment_results, true)) {
                notice_yesno(get_string('environmenterrorupgrade', 'admin'),
                        'index.php?confirmupgrade=1&confirmrelease=1', 'index.php');
            } else {
                notify(get_string('environmentok', 'admin'), 'notifysuccess');

                echo '<form action="index.php"><div>';
                echo '<input type="hidden" name="confirmupgrade" value="1" />';
                echo '<input type="hidden" name="confirmrelease" value="1" />';
                echo '</div>';
                echo '<div class="continuebutton"><input name="autopilot" id="autopilot"
type="checkbox" value="1" /><label for="autopilot">'.get_string('unattendedoperation',
'admin').'</label>';
                echo '<br /><br /><input type="submit" value="'.get_string('continue').'" /></div>';
```

```
            echo '</form>';
        }

        print_footer('none');
        die;
    } else {
        $strdatabasesuccess = get_string("databasesuccess");
        print_header($strdatabasechecking, $stradministration, $strdatabasechecking,
             "", upgrade_get_javascript(), false, " ", " ");

    /// return to original debugging level
        $CFG->debug = $origdebug;
        error_reporting($CFG->debug);
        upgrade_log_start();

    /// Upgrade current language pack if we can
        upgrade_language_pack();

        print_heading($strdatabasechecking);
        $db->debug=true;
    /// Launch the old main upgrade (if exists)
        $status = true;
        if (function_exists('main_upgrade')) {
            $status = main_upgrade($CFG->version);
        }
    /// If succesful and exists launch the new main upgrade (XMLDB), called
xmldb_main_upgrade
        if ($status && function_exists('xmldb_main_upgrade')){
            $status = xmldb_main_upgrade( $CFG->version);
        }
        $db->debug=false;
    /// If successful, continue upgrading roles and setting everything properly
        if ($status) {
            if (empty($CFG->rolesactive)) {
                //ugly hack - upgrade to new groups (from 1.6): MDL-9217
                require_once("$CFG->dirroot/group/db/upgrade.php");
                install_group_db();
                // Upgrade to the roles system.
                moodle_install_roles();
                set_config('rolesactive', 1);
            } else if (!update_capabilities()) {
                error('Had trouble upgrading the core capabilities for the Roles System');
            }
            // update core events
            events_update_definition();

            require_once($CFG->libdir.'/statslib.php');
            if (!stats_upgrade_for_roles_wrapper()) {
                notify('Couldn\'t upgrade the stats tables to use the new roles system');
            }
            if (set_config("version", $version)) {
                remove_dir($CFG->dataroot. '/cache', true); // flush cache
                notify($strdatabasesuccess, "green");
                print_continue("upgradesettings.php");
                print_footer('none');
```

```
                exit;
            } else {
                error('Upgrade failed!  (Could not update version in config table)');
            }
        /// Main upgrade not success
        } else {
            notify('Main Upgrade failed!  See lib/db/upgrade.php');
            print_continue('index.php?confirmupgrade=1&amp;confirmrelease=1');
            print_footer('none');
            die;
        }
        upgrade_log_finish();
    }
} else if ($version < $CFG->version) {
    upgrade_log_start();
    notify("WARNING!!!  The code you are using is OLDER than the version that made these
databases!");
    upgrade_log_finish();
}
} else {
    if (!set_config("version", $version)) {
        error("A problem occurred inserting current version into databases");
    }
}

/// Updated human-readable release version if necessary

if ($release <> $CFG->release) {  // Update the release version
    if (!set_config("release", $release)) {
        error("ERROR: Could not update release version in database!!");
    }
}

/// ugly hack - convert to new groups if upgrading from 1.7; must be reworked
require_once("$CFG->dirroot/group/db/upgrade.php");
upgrade_group_db("$CFG->wwwroot/$CFG->admin/index.php");  // Return here afterwards


/// Find and check all main modules and load them up or upgrade them if necessary
/// first old*.php update and then the new upgrade.php script
upgrade_activity_modules("$CFG->wwwroot/$CFG->admin/index.php");  // Return here
afterwards

/// Check all questiontype plugins and upgrade if necessary
/// first old *.php update and then the new upgrade.php script
/// It is important that this is done AFTER the quiz module has been upgraded
upgrade_plugins('qtype', 'question/type', "$CFG->wwwroot/$CFG->admin/index.php");  //
Return here afterwards

/// Upgrade backup/restore system if necessary
/// first old *.php update and then the new upgrade.php script
require_once("$CFG->dirroot/backup/lib.php");
upgrade_backup_db("$CFG->wwwroot/$CFG->admin/index.php");  // Return here afterwards

/// Upgrade blocks system if necessary
```

```
/// first old *.php update and then the new upgrade.php script
   require_once("$CFG->dirroot/lib/blocklib.php");
   upgrade_blocks_db("$CFG->wwwroot/$CFG->admin/index.php"); // Return here afterwards

/// Check all blocks and load (or upgrade them if necessary)
/// first old *.php update and then the new upgrade.php script
   upgrade_blocks_plugins("$CFG->wwwroot/$CFG->admin/index.php"); // Return here
afterwards

/// Check all enrolment plugins and upgrade if necessary
/// first old *.php update and then the new upgrade.php script
   upgrade_plugins('enrol', 'enrol', "$CFG->wwwroot/$CFG->admin/index.php"); // Return here
afterwards

/// Check all course formats and upgrade if necessary
   upgrade_plugins('format','course/format',"$CFG->wwwroot/$CFG->admin/index.php");

/// Check for local database customisations
/// first old *.php update and then the new upgrade.php script
   require_once( "$CFG->dirroot/lib/locallib.php");
   upgrade_local_db("$CFG->wwwroot/$CFG->admin/index.php"); // Return here afterwards

/// Check for changes to RPC functions
   require_once($CFG->dirroot.'/admin/mnet/adminlib.php');
   upgrade_RPC_functions("$CFG->wwwroot/$CFG->admin/index.php"); // Return here
afterwards

/// Upgrade all plugins for gradebook
   upgrade_plugins('gradeexport', 'grade/export', "$CFG->wwwroot/$CFG->admin/index.php");
   upgrade_plugins('gradeimport', 'grade/import', "$CFG->wwwroot/$CFG->admin/index.php");
   upgrade_plugins('gradereport', 'grade/report', "$CFG->wwwroot/$CFG->admin/index.php");

/// Check all message output plugins and upgrade if necessary
   upgrade_plugins('message','message/output',"$CFG->wwwroot/$CFG->admin/index.php");


/// just make sure upgrade logging is properly terminated
   upgrade_log_finish();

   unset($_SESSION['installautopilot']);

/// Set up the blank site - to be customized later at the end of install.
   if (! $site = get_site()) {
      // We are about to create the site "course"
      require_once($CFG->libdir.'/blocklib.php');

      $newsite = new Object();
      $newsite->fullname = "";
      $newsite->shortname = "";
      $newsite->summary = "";
      $newsite->newsitems = 3;
      $newsite->numsections = 0;
      $newsite->category = 1;
      $newsite->format = 'site'; // Only for this course
      $newsite->teacher = get_string("defaultcourseteacher");
```

```
        $newsite->teachers = get_string("defaultcourseteachers");
        $newsite->student = get_string("defaultcoursestudent");
        $newsite->students = get_string("defaultcoursestudents");
        $newsite->timemodified = time();

        if ($newid = insert_record('course', $newsite)) {
            // Site created, add blocks for it
            $page = page_create_object(PAGE_COURSE_VIEW, $newid);
            blocks_repopulate_page($page); // Return value not checked because you can always edit
later

            $cat = new Object();
            $cat->name = get_string('miscellaneous');
            if (insert_record('course_categories', $cat)) {
                redirect('index.php');
            } else {
                error("Serious Error! Could not set up a default course category!");
            }
        } else {
            error("Serious Error! Could not set up the site!");
        }
    }

    // initialise default blocks on admin and site page if needed
    if (empty($CFG->adminblocks_initialised)) {
        require_once("$CFG->dirroot/$CFG->admin/pagelib.php");
        require_once($CFG->libdir.'/blocklib.php');
        page_map_class(PAGE_ADMIN, 'page_admin');
        $page = page_create_object(PAGE_ADMIN, 0); // there must be some id number
        blocks_repopulate_page($page);

        //add admin_tree block to site if not already present
        if ($admintree = get_record('block', 'name', 'admin_tree')) {
            $page = page_create_object(PAGE_COURSE_VIEW, SITEID);
            blocks_execute_action($page, blocks_get_by_page($page), 'add', (int)$admintree->id,
false, false);
            if ($admintreeinstance = get_record('block_instance', 'pagetype', $page->type, 'pageid',
SITEID, 'blockid', $admintree->id)) {
                blocks_execute_action($page, blocks_get_by_page($page), 'moveleft',
$admintreeinstance, false, false);
            }
        }

        set_config('adminblocks_initialised', 1);
    }

/// Define the unique site ID code if it isn't already
    if (empty($CFG->siteidentifier)){   // Unique site identification code
        set_config('siteidentifier', random_string(32).$_SERVER['HTTP_HOST']);
    }

/// Check if the guest user exists.  If not, create one.
    if (! record_exists("user", "username", "guest")) {
        if (! $guest = create_guest_record()) {
            notify("Could not create guest user record !!!");
```

```
    }
  }

/// Set up the admin user
  if (empty($CFG->rolesactive)) {
    create_admin_user();
  }

/// Check for valid admin user
  require_login();

  $context = get_context_instance(CONTEXT_SYSTEM, SITEID);

  require_capability('moodle/site:config', $context);

/// check that site is properly customized
  if (empty($site->shortname) or empty($site->shortname)) {
    redirect('settings.php?section=frontpagesettings&amp;return=site');
  }

/// Check if we are returning from moodle.org registration and if so, we mark that fact to remove
reminders

  if (!empty($id)) {
    if ($id == $CFG->siteidentifier) {
      set_config('registered', time());
    }
  }

  $adminroot = admin_get_root();

/// Check if there are any new admin settings which have still yet to be set
  if( any_new_admin_settings( $adminroot )){
    redirect('upgradesettings.php');
  }

/// Everything should now be set up, and the user is an admin

/// Print default admin page with notifications.

  admin_externalpage_setup('adminnotifications');
  admin_externalpage_print_header();

/// Deprecated database! Warning!!
  if (!empty($CFG->migrated_to_new_db)) {
    print_box(print_string('dbmigrationdeprecateddb', 'admin'), 'generalbox adminwarning');
  }

/// Check for any special upgrades that might need to be run
  if (!empty($CFG->upgrade)) {
    print_box(get_string("upgrade$CFG->upgrade", "admin", "$CFG->wwwroot/$CFG-
>admin/upgrade$CFG->upgrade.php"));
  }
```

```
    if (ini_get_bool('register_globals') && !ini_get_bool('magic_quotes_gpc')) {
        print_box(get_string('globalsquoteswarning', 'admin'), 'generalbox adminwarning');
    }

    if (is_dataroot_insecure()) {
        print_box(get_string('datarootsecuritywarning', 'admin', $CFG->dataroot), 'generalbox
adminwarning');
    }

/// If no recently cron run
    $lastcron = get_field_sql('SELECT max(lastcron) FROM'. $CFG->prefix. 'modules');
    if (time() - $lastcron > 3600 * 24){
        $strinstallation = get_string('installation', 'install');
        $helpbutton = helpbutton('install', $strinstallation, 'moodle', true, false, '', true);
        print_box(get_string('cronwarning', 'admin')." ".$helpbutton, 'generalbox
adminwarning');
    }

// Print multilang upgrade notice if needed
    if (empty($CFG->filter_multilang_converted)) {
        print_box(get_string('multilangupgradenotice', 'admin'), 'generalbox adminwarning');
    }

/// Alert if we are currently in maintenance mode
    if (file_exists($CFG->dataroot.'/1/maintenance.html')) {
        print_box(get_string('sitemaintenancewarning', 'admin'), 'generalbox adminwarning');
    }


/// Print slightly annoying registration button every six months  ;-)
/// You can set the "registered" variable to something far in the future
/// if you really want to prevent this.  eg  9999999999
    if (!isset($CFG->registered) || $CFG->registered < (time() - 3600*24*30*6)){
        $options = array();
        $options['sesskey'] = $USER->sesskey;
        print_box_start('generalbox adminwarning');
        print_string('pleaseregister', 'admin');
        print_single_button('register.php', $options, get_string('registration'));
        print_box_end();
        $registrationbuttonsshown = true;
    }

    ////////////////////////////////////////////////////////////////////////////
    /// IT IS ILLEGAL AND A VIOLATION OF THE GPL TO HIDE, REMOVE OR MODIFY
THIS COPYRIGHT NOTICE //
    $copyrighttext = '<a href="http://moodle.org/">Moodle</a>'.
                     '<a href="http://docs.moodle.org/en/Release">'.$CFG->release.'</a> ('.$CFG-
>version.')<br />'.
                     'Copyright &copy; 1999 onwards, Martin Dougiamas<br />'.
                     'and <a href= "http://docs.moodle.org/en/Credits">many other contributors</a>.<br
/>'.
                     '<a href="http://docs.moodle.org/en/License">GNU Public License</a>';
    print_box( $copyrighttext, 'copyright');
    ////////////////////////////////////////////////////////////////////////////////
```

RIWAYAT HIDUP

Data Pribadi

Nama	: Devi Astianingsih
Tempat,Tanggal Lahir	: Jakarta, 2 Desember 1984
Jenis Kelamin	: Perempuan
Agama	: Islam
Kewarganegaraan	: Indonesia
Alamat Rumah	:Jln.Prambanan 3 No.2 Blok H 5 Bekasi Timur
Telepon dan Email	: 021-8823166/081319779882
	: devi.astianingsih@gmail.com

Pendidikan Formal

2002 –2007	Teknik Informatika Universitas Darma Persada
1999 –2002	SMU PGRI Bekasi
1996 –1999	SMP PGRI Bekasi
1990 – 1996	SDN Duren XIII Bekasi

Agustus 2007,

Devi Astianingsih