ABSTRAK

Aplikasi pencarian jodoh adalah suatu aplikasi untuk menfasilitasi para anggota mencari pasangan, teman dengan anggota lain. Program ini memiliki kemampuan untuk menampilkan hasil test kepribadian dari tiap anggota. Program ini juga mampu menentukan pasangan yang cocok terhadap anggota. Selain itu, dia juga mampu mencari pasangan sesuai dengan kata kunci yang dinginkan oleh anggota.

Pada *desktop*, aplikasi menyediakan fitur untuk menyimpan data *member*, data pasangan, data tipe kepribadian, data pertanyaan dari teori *Personality Plus* dan teori *Eneagram*, *edit* data *user*, melakukan registrasi, menyimpan dan mengolah data *test* kepribadian *Personality Plus*, menyimpan dan mengolah data *test Eneagram*, melihat hasil test, melihat profil pribadi dan pasangan, meng-upload foto, mencari pasangan, memberikan alert ke anggota jika ada anggota baru yang cocok, me-request anggota menjadi pasangan, dan mengirim pesan. Aplikasi ini juga menyediakan laporan tentang anggota atau pasangan yang memiliki tingkat kecocokan paling tinggi.

Aplikasi ini dikembangkan menggunakan bahasa pemrograman *XHTML* dan *PHP* untuk aplikasi dekstop dan nirkabelnya.

Kata kunci : *WAP*, *handphone*, teori *Personality Plus*, teori *Eneagram*, pencarian, *Desktop*, Openwave V7, *XHTML, PHP, MySQL.*

ABSTRACT

Searching for soulmate is an application to facilitated all members to find partner, friends with another member. This program has ability to show personality result from each member. This application also can decide a match partner to each member. Besides, it can find a partner based on the keyword that they want.

On the desktop, application has ability to save member data, partner data, personality type data, question from Personality Plus theory and Eneagram theory, to update member's data, registration, saving and processing personality test based on Personality theory, saving and processing personality test based on Eneagram theory, to view the test result, to view personal profile and partner, upload photo, searching partner, send alert to member if have a new match member, request member to be a soul mate and sending message. This application is providing report about member or partner who has the highest level of agreement.

This application developed by using XHTML and PHP programming language for the desktop and wireless application.

Keyword: WAP, Mobile phone, personality plus theory, eneagram theory, searching, desktop, openwave V7, XHTML, PHP, MySQL

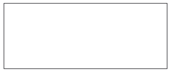
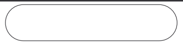
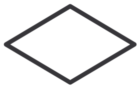
DAFTAR ISI

DAFTAR TABEL

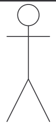
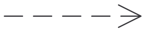
DAFTAR GAMBAR

DAFTAR SIMBOL

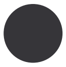
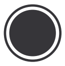
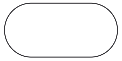
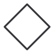
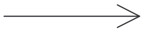
ERD

	Entitas
	Atribut
	Relasi

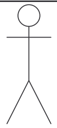
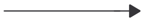
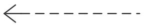
Use Case

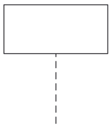
	Aktor
	Include / Extend

Aktivity Diagram

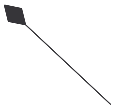
	Initial state
	Final state
	State
	Decision
	Transition

Sequence :

	Aktor
	Message call
	Message return

	Object

Class diagram

	Composition

LAMPIRAN

Berikut ini adalah seluruh *source code* yang digunakan untuk membangun aplikasi pencarian jodoh berdasarkan kesesuain sifat manusia menggunakan *Personality Plus* dan *Eneagram*.

Class SgnDB.php

```
<?php

class SgnDb {

        private $conn; // ADODB Connection handle
        private $rs;
        public $EOF;
        public $debug = false;

        public $Name;
        public $Id;

        private $current_record = array();
        private $_OnFirstRecord = false;



        function SgnDb($conn='')
        {
                global $sgn_adodb_conn;
                $this->conn = empty($conn) ? $sgn_adodb_conn : $conn;
        }


        private function WriteDebugLine($str) {
                echo "<hr>$str<hr>";
        }

        function MoveNext(){
                if ($this->rs) {
                        if (!$this->rs->EOF) {
                                if ($this->_OnFirstRecord) {
                                        $this->_OnFirstRecord = false;
                                } else {
                                        $this->rs->MoveNext();
                                }
                                $this->EOF = $this->rs->EOF;
                                return !$this->EOF;
                        }
                }
                return false;
        }

        public function UserDate($str, $fmt='') {
                return $this->rs->UserDate($str, $fmt);
        }
        public function DbDate($date) {
                return $this->rs->DbDate($date);
        }


        protected function fields($field_index){
                if ($this->rs) {
                        return $this->rs->fields[$field_index];
```

```
              } else {
                     return false;
              }
       }


       public function Execute($query) {

       if($this->debug)$this-
>WriteDebugLine("sgn_db:execute:query:".$query);
              $this->EOF = true;
              $this->rs = false;
              // execute query
              $rs = $this->conn->Execute($query);
              if (!$rs) {
                     if ($this->debug) $this->WriteDebugLine
                        ("sgn_db:execute:error:".$this->conn->ErrorMsg());
                     return false;
              } else {
                     $this->rs = $rs;
                     $this->EOF = false;
                     $this->_OnFirstRecord = true;
                     return true;
              }
       }

       private function ExecuteSelectLimit($query, $numrows, $offset) {
          if  ($this->debug)  $this->WriteDebugLine("Sgn_db:  SelectLimit:
query: ".$query);

              $this->EOF = true;
              $this->rs = false;

              // execute query
              $rs = $this->conn->SelectLimit($query, $numrows, $offset);
              if (!$rs) {
                     if ($this->debug) $this->WriteDebugLine("SgnDb:
                        SelectLimit: error: ".$this->conn->ErrorMsg());
                     return false;
              } else {
                     $this->rs = $rs;
                     $this->EOF = false;
                     $this->_OnFirstRecord = true;
                     return true;
              }
       }

       public function RecordCount() {
              return $this->rs->RecordCount();
       }

       public function Get($field_name) {
              return $this->fields($field_name);
       }


       public function Set($field_name='', $field_value='') {

                           if ( (!$field_name) && (!$field_value)
                     $this->current_record = Array();
                     return true;
              }

              if (is_array($field_name)) {
                     foreach ($field_name as $key=>$field_valueue) {
```

```
                $this->current_record[$key] = $field_valueue;
            }
            return true;
        }

        if ( (!is_array($field_name)) && ($field_name) ) {
            $this->current_record[$field_name] = $field_value;
            return true;
        }
        return false;

    }
    private function CreateFieldList($record) {
        $_return = "";
        if (is_array($record)) {
            $_separator = "";
            foreach($record as $key=>$value) {
                $_return .= $_separator . $key;
                $_separator = ",";
            }
        }
        return $_return;
    }


    private function GetValueList($record) {
        $_return = "";
        if (is_array($record)) {
            $_separator = "";
            foreach($record as $value) {
                $_return .= $_separator . "'" . $value ."'";
                $_separator = ",";
            }
        }
        return $_return;
    }

    public function Insert() {
        $record = $this->current_record;
        $_insert_field_list = $this->CreateFieldList($record);
        $_insert_value_list = $this->GetValueList($record);
        $sql = "INSERT INTO $this->Name ($_insert_field_list) VALUES
($_insert_value_list)";
        return $this->Execute($sql);
    }

    private function CreateUpdateList($record) {
        $_return = "";
        if (is_array($record)) {
            $_separator = "";
            foreach($record as $key=>$value) {
                $_return .= $_separator . $key . "='" . $value
."'";

                $_separator = ", ";
            }
        }
        return $_return;
    }

    public function Update($where='') {
        $record = $this->current_record;
        $_update_list = $this->CreateUpdateList($record);
        $_where_list = $this->CreateWhereList($where);
        $sql = "UPDATE $this->Name SET $_update_list $_where_list";
        return $this->Execute($sql);
        }
        public function UpdateById($value) {
                return $this->Update(array($this->Id=>$value));
        }

        public function Delete($where='') {
                $_where_list = $this->CreateWhereList($where);
                $sql = "DELETE FROM $this->Name $_where_list";
                return $this->Execute($sql);
        }
        public function DeleteById($value) {
                return $this->Delete(array($this->Id=>$value));
        }

        public    function    Select($column="",    $where="",    $order="",
$numrows=-1, $offset=-1) {
                $strSelect  = $this->CreateSelectList($column);
                $strWhere   = $this->CreateWhereList($where);
                $strOrderBy = $this->CreateOrderByList($order);
                if (($numrows>-1) && ($offset>-1)) {
                        return   $this->ExecuteSelectLimit("$strSelect   FROM
$this->Name $strWhere $strOrderBy", $numrows, $offset);
                } else {
                        return  $this->Execute("$strSelect  FROM  $this->Name
$strWhere $strOrderBy");
                }
        }
}

?>
```

SgnUser.php

```
<?php
class SgnUser {

        protected $username = "";
        protected $fullname = "";
        protected $roles = "";
        protected $not_authorized_page = "";

        function SgnUser()
        {
        }
        function Authenticate($username, $password) {
        }
                if (strstr($this->roles, $allowed_roles)) {
                        return true;
                } else {
                        header ("Location: ".$this-
>not_authorized_page);
                        exit;
                }
        }
        function IsMemberOf($role) {
                if (strstr($this->roles, $role)) {
                        return true;
                } else {
                        return false;
                }
        }
        function SetNotAuthorizedPage($page) {
            $this->not_authorized_page = $page;
      }
      static function SgnUserEncrypt($password, $encrypt_mode=1) {
            switch ($encrypt_mode) {
                  case 0: return $password; break;
                  case 1: return MD5($password); break;
                  default:  return MD5($password); break;
            }
      }


      function GetUsername() {
            return $this->username;
      }
      function GetFullname() {
            return $this->fullname;
      }
      function GetRoles() {
            return $this->roles;
      }

      function GetNotAuthorizedPage() {
            return $this->not_authorized_page;
      }
}
?>
```

```
sgnUpload.php
<?php

class sgnUpload {
      var $name = "";
      var $type = "";
      var $tmp_name = "";
      var $size = -1;

      var $files = array();

      var $max_file_size = -1;
      var $error_message = "";
      var $upload_dir = "";
      var $new_file_name = "";

      function sgnUpload($files){
            if ($files) {
                  $this->files = $files;
            }
      }

      function upload(){

            if (!$this->files) {
                  return false;
            }

            $source = $this->files['tmp_name'];
            if (empty($this->new_file_name)) {
            $destination = $this->upload_dir."/".$this-
>files['name'];
            } else {
            $destination = $this->upload_dir."/".$this-
>new_file_name;
            }

            if (move_uploaded_file($source, $destination)) {
                  return true;
            } else {
                  return false;
            }
      }

      function get_file_extension(){
            if ($this->files['name']) {
                  return get_file_extension($this->files['name']);
            } else {
                  return false;
            }
      }

      function set_upload_dir($upload_dir){
            $this->upload_dir = $upload_dir;
      }
      function set_max_file_size($max_file_size) {
            if ($max_file_size>0){
                  $this->max_file_size = $max_file_size;
            }
      }
      function set_new_file_name($new_file_name) {
            if (!empty($new_file_name)) {
                  $this->new_file_name = $new_file_name;
            }
      }
      function get_file_name() {
            if (empty($this->new_file_name)) {
                  return $this->files['name'];
            } else {
                  return $this->new_file_name;
            }
      }
}
```

MasterMember.php

```
require_once '../../web/utils/SgnDb.php';

class MasterMember extends SgnDb {

      function MasterMember() {
            $this->SgnDb();
            $this->Name = "ms_member"; // table name
            $this->Id = "id_member"; // primary key
      }
      function GetPasanganEneagram($id_member) {

            $this->Select("jk,result_eneagram",
                  array("id_member"=>$id_member));
            $member_jk = $this->Get("jk");
            $member_eneagram = $this->GetDominantEneagram
            ($id_member);
            $target_jk = ($member_jk=="L") ? "P" : "L";
            $_str = "(";
            foreach($member_eneagram as $persKey=>$persValue) {
                  $_str .= $_separator . " MPE.kepribadian1 =
                           '$persKey' ";
                  $_separator = " OR ";
            }
            $_str .= ")";
            $where_mpe_kepribadian1 = $_str;

            // select pasangan eneagram
$this->Execute("SELECT DISTINCT id_member,  nama,Photo,
nama_kepribadian2,keterangan,Provinsi,tggi_bdn,
brt_bdn,ket_pribadi,hobi,kecocokan,tgl_lhr
      FROM
         (
      SELECT MM.id_member, MM.nama, MM.Photo, MM.Provinsi,
MM.tggi_bdn, MM.brt_bdn, MM.ket_pribadi, MM.hobi, MM.result_eneagram,
MM.tgl_lhr, MPE.kepribadian1, MPE.kepribadian2, MPE.kecocokan,
 MK.nama nama_kepribadian2, MK.keterangan
      FROM
      ms_member MM
      JOIN ms_pasangan_eneagram MPE ON
MPE.kepribadian2=MM.result_eneagram JOIN ms_kepribadian MK ON
MPE.kepribadian2=MK.id_kepribadian
      WHERE
           $where_mpe_kepribadian1
           AND MPE.jk = '$member_jk'
           AND MM.jk = '$target_jk'
           ORDER BY
           MPE.kecocokan DESC
        ) as A
        LIMIT 0,3"
        );
}
      function GetTargetPersonality($personality) {
      if ($personality=="p1") { $target_personality = "p3"; } else
      if ($personality=="p2") { $target_personality = "p4"; } else
      if ($personality=="p3") { $target_personality = "p1"; } else
      if ($personality=="p4") { $target_personality = "p2"; }
          return $target_personality;
      }
      function GetTargetEneagram($eneagram) {
            if ($eneagram=="e1") { $target_eneagram = "e1"; } else
            if ($eneagram=="e2") { $target_eneagram = "e2"; } else
            if ($eneagram=="e3") { $target_eneagram = "e3"; } else
            if ($eneagram=="e4") { $target_eneagram = "e4"; } else
            if ($eneagram=="e5") { $target_eneagram = "e5"; } else
            if ($eneagram=="e6") { $target_eneagram = "e6"; } else
            if ($eneagram=="e7") { $target_eneagram = "e7"; } else
            if ($eneagram=="e8") { $target_eneagram = "e8"; } else
            if ($eneagram=="e9") { $target_eneagram = "e9"; }
            return $target_eneagram;
      }
      function GetPasanganRekomendasi2($id_member) {
            $this->Select("jk,result_eneagram,result_personality",
                   array("id_member"=>$id_member));
            $member_jk = $this->Get("jk");
            $member_personality = $this->Get("result_personality");
            $member_eneagram = $this->GetDominantEneagram($id_member);
            // set jenis kelamin target
            $target_jk = ($member_jk=="L") ? "P" : "L";

            $_str = "(";
            foreach($member_eneagram as $persKey=>$persValue) {
                  $_str .= $_separator . " MPE.kepribadian1 =
                  '$persKey' ";
                  $_separator = " OR ";
            }
            $_str .= ")";
            $where_mpe_kepribadian1 = $_str;
            $target_personality = $this->GetTargetPersonality
                                  ($member_personality);
$queryGetPasanganEneagram = " SELECT DISTINCT id_member,
nama,Photo,nama_kepribadian2,keterangan,Provinsi,tggi_bdn,
brt_bdn,ket_pribadi,hobi,kecocokan,tgl_lhr FROM
(
SELECT MM.id_member, MM.nama, MM.Photo, MM.Provinsi,
MM.tggi_bdn, MM.brt_bdn, MM.ket_pribadi, MM.hobi, MM.result_eneagram,
MM.tgl_lhr, MPE.kepribadian1, MPE.kepribadian2, MPE.kecocokan,
MK.nama nama_kepribadian2, MK.keterangan
            FROM
            ms_member MM
            JOIN ms_pasangan_eneagram MPE ON
            MPE.kepribadian2=MM.result_eneagram
            JOIN ms_kepribadian MK ON MPE.kepribadian2=
            MK.id_kepribadian
                   WHERE
                    $where_mpe_kepribadian1
                    AND MPE.jk = '$member_jk'
                    AND MM.jk = '$target_jk'
                   ORDER BY
                    MPE.kecocokan DESC
                   ) as A
                   LIMIT 0,3";

            $this->Execute($queryGetPasanganEneagram);
            $_separator = "";
            while ($this->MoveNext()) {
                  $strPasanganEneagramList .= $_separator . "'"
                   .$this->Get("id_member")."'";
                  $_separator = ",";
            }

            // select pasangan personality
$this->Execute("SELECT DISTINCT MM.id_member, MM.nama,MM.
Photo,MK.nama nama_kepribadian2,MK.keterangan,MM.Provinsi,MM.tggi_bdn,
MM.brt_bdn,MM.ket_pribadi,MM.hobi,MM.tgl_lhr
FROM ms_member MM JOIN ms_pasangan_eneagram MPE ON
MPE.kepribadian2=result_eneagram
JOIN ms_kepribadian MK ON MPE.kepribadian2=MK.id_kepribadian
                    WHERE
                    MM.jk = '$target_jk'
                    AND MM.result_personality = '$target_personality'
                    AND MM.id_member IN ($strPasanganEneagramList)");
      }

      function GetDominantPersonality($id_member) {

$this->Select("result_personality_p1, result_personality_p2,
result_personality_p3,result_personality_p4",array("id_member"=>$id_me
mber));
            $score = array();
            $score["p1"] = $this->Get("result_personality_p1");
            $score["p2"] = $this->Get("result_personality_p2");
            $score["p3"] = $this->Get("result_personality_p3");
            $score["p4"] = $this->Get("result_personality_p4");
            arsort($score);

            $result = array();
            $topScore = 101;
            $topScoreCount = 0;
            foreach($score as $key=>$value) {
      if ((($value<$topScore) && ($topScoreCount<2)) ||
($value==$topScore)) {
            $result[$key] = $value;
            if ($topScore != $value) {
                $topScoreCount++;
                $topScore = $value;
                        }
                  }

            }
            asort($result);
            return $result;

      }
function GetDominantEneagram($id_member) {

$this->Select("result_eneagram_e1,
result_eneagram_e2,result_eneagram_e3,
result_eneagram_e4,
result_eneagram_e5,result_eneagram_e6,result_eneagram_e7,
result_eneagram_e8,result_eneagram_e9",array("id_member"=>$id_member))
;
            $score = array();
            $score["e1"] = $this->Get("result_eneagram_e1");
            $score["e2"] = $this->Get("result_eneagram_e2");
            $score["e3"] = $this->Get("result_eneagram_e3");
            $score["e4"] = $this->Get("result_eneagram_e4");
            $score["e5"] = $this->Get("result_eneagram_e5");
            $score["e6"] = $this->Get("result_eneagram_e6");
            $score["e7"] = $this->Get("result_eneagram_e7");
            $score["e8"] = $this->Get("result_eneagram_e8");
            $score["e9"] = $this->Get("result_eneagram_e9");
            arsort($score);

            $result = array();
            $topScore = 101;
            $topScoreCount = 0;
            foreach($score as $key=>$value) {
      if ((($value<$topScore) && ($topScoreCount<2)) ||
($value==$topScore)) {
```

```
                        $result[$key] = $value;
                        if ($topScore != $value) {
                                    $topScoreCount++;
                              $topScore = $value;
      }
                  }
            }
            return $result;
      }
}
?>
```

MsKepribadian.php

```
<?php
require_once '../../web/utils/SgnDb.php';

class MsKepribadian extends SgnDb {
      function MsKepribadian() {
            $this->SgnDb();
            $this->Name = "ms_kepribadian"; // table name
            $this->Id = "id_kepribadian"; // primary key
      }
      function GetNamaById($id_kepribadian) {
$this->Execute("SELECT nama FROM ms_kepribadian WHERE
id_kepribadian='$id_kepribadian'");
            return $this->Get("nama");
      }
}
?>
```

MsMemberFoto.php

```
<?php
require_once '../../web/utils/SgnDb.php';

class MsMemberFoto extends SgnDb {

      function MsMemberFoto() {
            $this->SgnDb();
            $this->Name = "tr_member"; // table name
            $this->Id = "idn"; // primary key
      }
}
?>
```

MsPasanganEneagram.php

```
<?php
require_once '../../web/utils/SgnDb.php';

class MsPasanganEneagram extends SgnDb {
      function MsPasanganEneagram() {
            $this->SgnDb();
            $this->Name = "ms_pasangan_eneagram"; // table name
            $this->Id = "idn"; // primary key
      }
}
?>
```

MsPasanganPersonality.php

```
<?php
require_once '../../web/utils/SgnDb.php';

class MsPasanganPersonality extends SgnDb {

      function MsPasanganPersonality() {
            $this->SgnDb();
            $this->Name = "ms_pasangan_personality"; // table name
            $this->Id = "idn"; // primary key
      }
}
?>
```

MsPertanyaanEneagram.php

```
<?php
require_once '../../web/utils/SgnDb.php';

class MsPertanyaanEneagram extends SgnDb {

      function MsPertanyaanEneagram() {
            $this->SgnDb();
            $this->Name = "ms_pertanyaan_eneagram"; // table name
            $this->Id = "idn"; // primary key
      }

      function GetJumlahSoal() {
            $this->Execute("SELECT COUNT(idn) num FROM
                   ms_pertanyaan_eneagram");
            if ($this->MoveNext()) {
                  return $this->Get("num");
            }
      }

      function GetJumlahSoalByKepribadian($kepribadian) {
            $this->Execute("SELECT COUNT(idn) num FROM
ms_pertanyaan_eneagram WHERE id_kepribadian='$kepribadian'");
            if ($this->MoveNext()) {
                  return $this->Get("num");
            }
      }
}
?>
```

MsPertanyaanPersonality.php

```
?php
require_once '../../web/utils/SgnDb.php';

class MsPertanyaanPersonality extends SgnDb {

      function MsPertanyaanPersonality() {
            $this->SgnDb();
            $this->Name = "ms_pertanyaan_personality"; // table
name
            $this->Id = "idn"; // primary key
      }

      function GetJumlahSoal() {
            $this->Execute("SELECT COUNT(DISTINCT(no_soal)) num
FROM
                   ms_pertanyaan_personality");
            if ($this->MoveNext()) {
                  return $this->Get("num");
            }
      }
}
?>
```

SgnUserCJ.php

```
<?php
require_once '../../web/utils/SgnUser.php';
require_once "../../web/entities/MasterMember.php";


class SgnUserCJ extends SgnUser {

      protected $id_member = 0;

      function SgnUserSL()
      {
            $this->SgnUser();
      }
      function ChangePassword($new_password) {

            if ($this->roles=='member') {
                  $member = new MasterMember();

      $member->Select("id_member",array("username"=>$this-
>username));
      $id_member = $member->Get("id_member");

      $member->Set();
      $member->Set("password",$this-
>SgnUserEncrypt($new_password));
                  return $member->UpdateById($id_member);
            }
      }
      function Authenticate($username, $password, $encrypt_mode=1)
{
            $member = new MasterMember();
            $member->Select("", array("username"=>$username,
 password"=>$this->SgnUserEncrypt($password, $encrypt_mode)));
            if ($member->MoveNext()) {
                  $this->id_member = $member->Get("id_member");
                  $this->username = $member->Get("username");
                  $this->fullname = $member->Get("nama");
                  $this->roles = $member->Get("role");
                  return true;
            }
            return false;
      }

      function ValidatePageAccess($allowed_roles) {

      if ($this->roles && strstr($allowed_roles, $this->roles)) {
                  return true;
            } else {
                  header ("Location: ".$this-
>not_authorized_page);
                  exit;
            }
      }

      function GetIdMember() {
            return $this->id_member;
      }
}
?>
```

SgnUserSL.php

```
<?php

require_once '../../web/utils/SgnUser.php';
require_once "../../web/entities/MasterMember.php";

class SgnUserSL extends SgnUser {

      function SgnUserSL()
      {
            $this->SgnUser();
      }
      function ChangePassword($new_password) {

      if ($this->roles=='member') {
       member = new MasterMember();
      // get user id
      $member->Select("id_member",array("username"==$this-
>username));

      $member->Set();
      $member->Set("password",$this-
>SgnUserEncrypt($new_password));
                  return $member->UpdateById($id_member);
            }
      }
      function Authenticate($username, $password, $encrypt_mode=1)
{

            $member = new MasterMember();
            $member->Select("", array("username"=>$username,
"password"=>$this->SgnUserEncrypt($password, $encrypt_mode)));
            if ($member->MoveNext()) {
                  $this->username = $member->Get("username");
                  $this->fullname = $member->Get("nama");
                  $this->roles = "member";
                  return true;
            }
            return false;
      }
}
?>
```

TrJawabanEneagram.php

```
<?php
```

```
require_once '../../web/utils/SgnDb.php';

class TrJawabanEneagram extends SgnDb {

      function TrJawabanEneagram() {
            $this->SgnDb();
            $this->Name = "tr_jawaban_eneagram"; // table name
            $this->Id = "idn"; // primary key
      }

      function GetSoal($user_id,$page=1) {
            $limitStart = $page*4;
            $this->Execute(
            "SELECT J.no_urut, J.no_soal, P.no_urut_soal,
P.pertanyaan
            FROM tr_jawaban_personality J
            JOIN ms_pertanyaan_personality P ON
P.no_soal=J.no_soal
            WHERE J.user_id = ".$user_id."
            ORDER BY J.no_urut, P.no_urut_soal
            LIMIT $limitStart,4");
      }

      function CreateSummary($user_id) {

            $result_eneagram = array();
$this->Execute("SELECT id_kepribadian, COUNT(idn) AS score FROM
tr_jawaban_eneagram WHERE user_id='$user_id' AND jawaban='Y' GROUP
BY id_kepribadian");
while($this->MoveNext()) {
$result_eneagram[$this->Get("id_kepribadian")] = $this-
>Get("score");
            }
            require_once "MasterMember.php";
            $member = new MasterMember();
            $member->Select("", array("id_member"=>$user_id));
            $max_value = 0;
            foreach ($result_eneagram as $key=>$value) {
                  $member->Set("result_eneagram_".$key, $value);
                  if ($value>$max_value) {
                        $max_value = $value;
                        $member->Set("result_eneagram", $key);
                  }
            }
            $member->UpdateById($user_id);
      }
}
?>
```

TrJawabanPersonality.php

```
<?php
require_once '../../web/utils/SgnDb.php';

class TrJawabanPersonality extends SgnDb {

      function TrJawabanPersonality() {
            $this->SgnDb();
            $this->Name = "tr_jawaban_personality"; // table name
            $this->Id = "idn"; // primary key
      }

      function GetSoal($user_id,$page=1) {
            $limitStart = $page*4;
            $this->Execute(
"SELECT J.no_urut, J.no_soal, J.idn, P.no_urut_soal, P.pertanyaan
FROM tr_jawaban_personality J OIN ms_pertanyaan_personality P ON
P.no_soal=J.no_soal
            WHERE J.user_id = ".$user_id."
            ORDER BY J.no_urut, P.no_urut_soal
            LIMIT $limitStart,4");
      }
}
?>
```

TrMessage.php

```
<?php
require_once '../../web/utils/SgnDb.php';

class TrMessage extends SgnDb {

      function TrMessage() {
            $this->SgnDb();
            $this->Name = "tr_message"; // table name
            $this->Id = "idn"; // primary key
      }

}
?>
```

TrSoulmate.php

```
<?php
require_once '../../web/utils/SgnDb.php';

class TrSoulmate extends SgnDb {
      function TrSoulmate() {
            $this->SgnDb();
            $this->Name = "tr_soulmate"; // table name
            $this->Id = "idn"; // primary key
      }

      function GetStatus($status) {
            if ($status == "T") {
                  return "Diterima";
            } else if ($status == "F") {
                  return "Menunggu Konfirmasi";
            } else if ($status == "R") {
                  return "Ditolak";
            } else if ($status == "D") {
                  return "Putus";
            }
      }

      function GetDisplayStyle($status, $button) {
            if ($button=="approve" || $button=="reject") {
                  switch ($status) {
                        case 'F': return 'inline';break;
                        case 'T': return 'none';break;
                        case 'R': case 'D': return 'none'; break;
                  }
            } else if ($button=="delete") {
                  switch ($status) {
                        case 'F': return 'none';break;
                        case 'T': return 'inline';break;
                        case 'R': case 'D': return 'none'; break;
                  }
            }
      }

}
?>
```

Member.php

```
<?php

require_once "../../web/entities/MasterMember.php";
require_once('../../web/entities/SgnUserSL.php');
$user_auth = new SgnUserSL();
$user_auth-
>SetNotAuthorizedPage('../../modules/Member/NotAuthorized.php');

if ($_SESSION['User_username'] && $_SESSION['User_password']) {
      $_SESSION['User_password'] is encrypted password
      $username = $_SESSION['User_username'];
      $password = $_SESSION['User_password'];

      if (!$user_auth->Authenticate($username,$password,0)) {
            header("Location: ".$user_auth->GetNotAuthorizedPage);
      }
        } else {
      header("Location: ".$user_auth->GetNotAuthorizedPage);
}


$member = new MasterMember();

$thisyear = date("Y");
for($thn=$thisyear; $thn>($thisyear-100); $thn--) {
      $arrayTglLahirThn[$thn] = $thn;
}
$arrayTglLahirBln[1] = "Januari";
$arrayTglLahirBln[2] = "Pebruari";
$arrayTglLahirBln[3] = "Maret";
$arrayTglLahirBln[4] = "April";
$arrayTglLahirBln[5] = "Mei";
$arrayTglLahirBln[6] = "Juni";
$arrayTglLahirBln[7] = "Juli";
$arrayTglLahirBln[8] = "Agustus";
$arrayTglLahirBln[9] = "September";
$arrayTglLahirBln[10] ="Oktober";
$arrayTglLahirBln[11] ="November";
$arrayTglLahirBln[12] ="Desember";


$thisdate = date("D");
```

```
for($tgl=1; $tgl<=31; $tgl++) {
      $arrayTglLahirTgl[$tgl] = $tgl;
}

$arrayStatus['S'] = "Single";
$arrayStatus['R'] = "In Relationship";
$arrayStatus['M'] = "Married";
$arrayStatus['D'] = "Divorced";

if ($frm->getVar("action")=="submit_add") {

      $isFormValid = true;
      $a = $frm->getVar("nama");
      if (empty($a)) {
$tpl->assign("message", MSG_INSERT_FAILED.": Nama Pengguna harus
diisi", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $isFormValid = false;
      }

      $a = $frm->getVar("username");
      if (empty($a)) {
            $tpl->assign("message", MSG_INSERT_FAILED.": Username
harus diisi", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $isFormValid = false;
      }

      $member->Execute("SELECT COUNT(*) AS num FROM ms_member
WHERE username='".$frm->getVar("username")."'");
if ($member->Get("num")>0) {
$tpl->assign("message", "Nama Pengguna Telah Terpakai. Silakan isi
yang lain", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $isFormValid = false;
      }
      require_once "../../web/utils/sgnUpload.php";
      $upload = new sgnUpload($_FILES['photo']);

      $upload->set_upload_dir(UPLOAD_FOTO_PATH);
      $upload->set_max_file_size(BACKGROUND_MAX_FILE_SIZE);
            if (!$upload->upload()){
$tpl->assign("message", MSG_INSERT_FAILED.": Foto tidak dapat
diupload", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $upload_success = false;
      } else {
            $upload_success = true;
      }


      if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", true);
            $frm->setVar("action", "repost");
      } else {

            $member->Set();
            $member->Set("username", $frm->getVar("username"));
            $member->Set("password", SgnUser::SgnUserEncrypt($frm-
>getVar("password_new1")));
            $member->Set("nama", $frm->getVar("nama"));
            $member->Set("jk", $frm->getVar("jk"));
            $member->Set("tgl_lhr", $frm-
>getVar("tgl_lahir_thn")."-".$frm->getVar("tgl_lahir_bln")."-
".$frm->getVar("tgl_lahir_tgl"));
            $member->Set("alamat", $frm->getVar("alamat"));
            $member->Set("Provinsi", $frm->getVar("Provinsi"));
            $member->Set("Negara", $frm->getVar("Negara"));
            $member->Set("email", $frm->getVar("email"));
            $member->Set("hp", $frm->getVar("hp"));
            $member->Set("pkrjaan", $frm->getVar("pkrjaan"));
            $member->Set("status", $frm->getVar("status"));
            $member->Set("wrn_rmbut", $frm->getVar("wrn_rmbut"));
            $member->Set("tggi_bdn", $frm->getVar("tggi_bdn"));
            $member->Set("brt_bdn", $frm->getVar("brt_bdn"));
            $member->Set("agama", $frm->getVar("agama"));
            $member->Set("pendidikan", $frm-
>getVar("pendidikan"));
            $member->Set("ket_pribadi", $frm-
>getVar("ket_pribadi"));
            $member->Set("mkn_fav", $frm->getVar("mkn_fav"));
            $member->Set("film_fav", $frm->getVar("film_fav"));
            $member->Set("musik_fav", $frm->getVar("musik_fav"));
            $member->Set("hobi", $frm->getVar("hobi"));

            if ($upload_success) {
                  $member->Set("photo", $upload->get_file_name());

                  require_once
"../../web/entities/MsMemberFoto.php";
                  $foto = new MsMemberFoto();
                  $foto->Set("foto", $upload->get_file_name());
                  $foto->Set("id_member", $user->GetIdMember());
                  $foto->Insert();

            $tpl->set_page_template("browse"); // template overrie
if ($member->Insert()) {
            $tpl->assign("message", MSG_INSERT_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            $frm->setVar("action", "browse");
            } else {
            $tpl->assign("message", MSG_INSERT_FAILED, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
                  }
            }
      }
}

if ($frm->getVar("action")=="submit_update") {
      $isFormValid = true;

      $a = $frm->getVar("nama");
      if (empty($a)) {
            $tpl->assign("message", MSG_INSERT_FAILED.": Nama
Lengkap harus diisi", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $isFormValid = false; // flag
      }

      $a = $frm->getVar("username");
      if (empty($a)) {
      $tpl->assign("message", MSG_INSERT_FAILED.": Username harus
diisi", "Message");
      $tpl->assign("message_type_class", "MsgError", "Message");
      $isFormValid = false;
      }

$member->Execute("SELECT COUNT(*) AS num FROM ms_member WHERE
username='".$frm->getVar("username")."' AND id_member!='".$frm-
>getVar("id")."'");
if ($member->Get("num")>0) {
$tpl->assign("message", "Username is already exist. Please select
another", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $isFormValid = false;
      }

      if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", false);
            $frm->setVar("action", "repost");
      } else {
            $member->Set();
            $member->Set("username", $frm->getVar("username"));
            $member->Set("nama", $frm->getVar("nama"));
            $member->Set("jk", $frm->getVar("jk"));
            $member->Set("tgl_lhr", $frm-
>getVar("tgl_lahir_thn")."-".$frm->getVar("tgl_lahir_bln")."-
".$frm->getVar("tgl_lahir_tgl"));
            $member->Set("alamat", $frm->getVar("alamat"));
            $member->Set("Provinsi", $frm->getVar("Provinsi"));
            $member->Set("Negara", $frm->getVar("Negara"));
            $member->Set("email", $frm->getVar("email"));
            $member->Set("hp", $frm->getVar("hp"));
            $member->Set("pkrjaan", $frm->getVar("pkrjaan"));
            $member->Set("status", $frm->getVar("status"));
            $member->Set("wrn_rmbut", $frm->getVar("wrn_rmbut"));
            $member->Set("tggi_bdn", $frm->getVar("tggi_bdn"));
            $member->Set("brt_bdn", $frm->getVar("brt_bdn"));
            $member->Set("agama", $frm->getVar("agama"));
            $member->Set("pendidikan", $frm-
>getVar("pendidikan"));
            $member->Set("ket_pribadi", $frm-
>getVar("ket_pribadi"));
            $member->Set("mkn_fav", $frm->getVar("mkn_fav"));
            $member->Set("film_fav", $frm->getVar("film_fav"));
            $member->Set("musik_fav", $frm->getVar("musik_fav"));
            $member->Set("hobi", $frm->getVar("hobi"));

            $p = $frm->getVar("password_new1");
            if (!empty($p)) {
            $member->Set("password", SgnUser::SgnUserEncrypt($p));
            }

require_once "../../web/utils/sgnUpload.php";
$upload = new sgnUpload($_FILES['photo']);
$upload->set_upload_dir(UPLOAD_FOTO_PATH);
$upload->set_max_file_size(BACKGROUND_MAX_FILE_SIZE);

if (!$upload->upload()){
$tpl->assign("message", MSG_INSERT_FAILED.": Foto tidak dapat
diupload", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
                  $upload_success = false;
            } else {
                  $upload_success = true;
            }
                  if ($upload_success)
                  $member->Set("photo", $upload->get_file_name());

            if ($member->UpdateById($frm->getVar("id"))) {
                  $tpl->set_page_template("edit");
                  $tpl->assign("isAdd", false);
                  $frm->setVar("action", "edit");
                  $frm->setVar("id", $frm->getVar("id"));
            $tpl->assign("message", MSG_UPDATE_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            } else {
                  $tpl->assign("isAdd", false);
                  $tpl->set_page_template("edit");
            $tpl->assign("message", MSG_UPDATE_DELETE, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
      }

}

if ($frm->getVar("action")=="submit_delete") {
      if ($member->DeleteById($frm->getVar("id"))) {
      $frm->setVar("action", "browse");
      $tpl->set_page_template($frm->getVar("action"));
      $tpl->assign("message", MSG_DELETE_SUCCESS, "Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
      } else {
            $tpl->assign("message", MSG_DELETE_FAILED);
            $tpl->assign("message_type_class", "MsgError",
"Message");
      }

}
```

```
if ($frm->getVar("action")=="submit_delete_many") {

      $frm->setVar("action", "browse");

      $num_success = 0;
      $num_failed = 0;

      for ($i=0; $i<$frm->getVar("chkItemNum"); $i++) {
            $a = $frm->getVar("chk".$i);
            if (!empty($a)) {
                  $num_item++;
                  if ($member->DeleteById($a)) {
                        $num_success++;
                  } else {
                        $num_success++;
                  }
            }
      }

}
// Select, Fetch, Assign List
$member->Select("", $_where, $_order, $row_per_page, $paging-
>get("offset"));
      $line_count = 0;
      while ($member->MoveNext()) {
            $tpl->assign_list("list", array(
                  "line_count"=>$line_count++,
                  "id_member"=>$member->Get("id_member"),
                  "username"=>$member->Get("username"),
                  "nama"=>$member->Get("nama"),
                  "jk"=>$member->Get("jk"),
                  "tgl_lhr"=>$member->Get("tgl_lhr"),
                  "alamat"=>$member->Get("alamat"),
                  "email"=>$member->Get("email"),
                  "hp"=>$member->Get("hp"),
                  "agama"=>$member->Get("agama"),
                  "pkrjaan"=>$member->Get("pkrjaan")
                        )
                  );
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->register_list("list");
require_once "../../web/inc/end.php";

?>
```

MsKepribadian.php

```
<?php
require_once "../../web/inc/begin.php";
require_once "../../web/entities/MsKepribadian.php";

$kepribadian = new MsKepribadian();

// Add
if ($frm->getVar("submit_add")) {

      $isFormValid = true;
      $a = $frm->getVar("id_kepribadian");
      if (empty($a)) {
$tpl->assign("message", MSG_INSERT_FAILED.": ID Kepribadian harus
diisi", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $isFormValid = false;
      }
      $a = $frm->getVar("nama");
      if (empty($a)) {
            $tpl->assign("message", MSG_INSERT_FAILED.": Nama
Kepribadian harus diisi", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $isFormValid = false; // flag
      }
      $a = $frm->getVar("jenis");
      if (empty($a))    {
$tpl->assign("message",MSG_INSERT_FAILED.": Jenis Kepribadian
harus diisi", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $isFormValid = false;
            }
      $a = $frm->getVar("keterangan");
      if (empty($a))    {
$tpl->assign("message",MSG_INSERT_FAILED.": Keterangan harus
diisi", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            $isFormValid = false;
            }

      if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", true);
            $frm->setVar("action", "repost");
      } else {

$kepribadian->Set();
$kepribadian->Set("id_kepribadian", $frm-
>getVar("id_kepribadian"));
$kepribadian->Set("nama", $frm->getVar("nama"));
$kepribadian->Set("jenis", $frm->getVar("jenis"));
$kepribadian->Set("keterangan", $frm->getVar("keterangan"));

            $tpl->set_page_template("browse"); // template overrie
            if ($kepribadian->Insert()) {
            $tpl->assign("message", MSG_INSERT_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            } else {
            $tpl->assign("message", MSG_INSERT_FAILED, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
            $frm->setVar("action", "browse");
      }
```

```
}
//edit
if ($frm->getVar("action")=="submit_delete_many") {

      $frm->setVar("action", "browse");
      $num_success = 0;
      $num_failed = 0;

      for ($i=0; $i<$frm->getVar("chkItemNum"); $i++) {
            $a = $frm->getVar("chk".$i);
            if (!empty($a)) {
                  $num_item++;
                  if ($kepribadian->DeleteById($a)) {
                       $num_success++;
                  } else {
                        $num_success++;
                  }
            }
      }
if ($frm->getVar("action")=="browse") {

      $tpl->set_page_template($frm->getVar("action"));
      $_order = $sgn_poas->GetOrder();
      $_where = $sgn_poas->GetSearch(true);

      $total_record = $kepribadian->GetSelectCount($_where);
      $current_page = $frm->getVar("p");
      $row_per_page = BROWSE_PERPAGE;
$paging = new SgnPaging($total_record, $current_page,
$row_per_page);

      // Select, Fetch, Assign List
      $kepribadian->Select("", $_where, $_order, $row_per_page,
$paging->get("offset"));
      $line_count = 0;
      while ($kepribadian->MoveNext()) {
            $tpl->assign_list("list", array(
            "line_count"=>$line_count++,
            "id_kepribadian"=>$kepribadian->Get("id_kepribadian"),
            "nama"=>$kepribadian->Get("nama"),
            "jenis"=>$kepribadian->Get("jenis"),
            "keterangan"=>nl2br($kepribadian->Get("keterangan")),
                  ));
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->register_list("list");

require_once "../../web/inc/end.php";
?>
```

```
MsMemberFoto.php
```

```
<?php
require_once "../../web/inc/begin.php";

$user->ValidatePageAccess('member,admin');
```

```
require_once "../../web/entities/MsMemberFoto.php";
if ($frm->getVar("submit_add")) {

$isFormValid = true;
require_once "../../web/utils/sgnUpload.php";
$upload = new sgnUpload($_FILES['photo']);
$new_file_name = $user-
>GetIdMember().'_'.date("ymdHis").".".$upload-
>get_file_extension();

      $upload->set_upload_dir(UPLOAD_FOTO_PATH);
      $upload->set_max_file_size(BACKGROUND_MAX_FILE_SIZE);
      $upload->set_new_file_name($new_file_name);

      $tpl->set_page_template("browse");
      if ($foto->Insert()) {
            $tpl->assign("message", MSG_INSERT_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            $frm->setVar("action", "browse");
            } else {
            $tpl->assign("message", MSG_INSERT_FAILED, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
      }
// Set As Default

if ($frm->getVar("action")=="set_as_default") {
      $foto->SelectById($frm->getVar("action_value"));

      $member = new MasterMember();
      $member->Set('Photo', $foto->Get('foto'));
      $member->UpdateById($user->GetIdMember());
      $frm->setVar("action", "browse");
}
if ($frm->getVar("action")=="submit_delete") {
      if ($eneagram->DeleteById($frm->getVar("id"))) {
            $frm->setVar("action", "browse");
            $tpl->set_page_template($frm->getVar("action"));
      $tpl->assign("message", MSG_DELETE_SUCCESS, "Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
      } else {
            $tpl->assign("message", MSG_DELETE_FAILED);
            $tpl->assign("message_type_class", "MsgError",
"Message");
      }
}
require_once "../../web/inc/end.php";
?>
```

MsPasanganEneagram.php

```
<?php

require_once "../../web/inc/begin.php";
require_once "../../web/entities/TrJawabanEneagram.php";
require_once "../../web/entities/MsPasanganEneagram.php";

$pasangan = new MsPasanganEneagram();

if ($frm->getVar("submit_add")) {
      $isFormValid = true;
      if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", true);
            $frm->setVar("action", "repost");
      } else {
            $pasangan->Set();
            $pasangan->Set("idn", $frm->getVar("idn"));
            $pasangan->Set("jk", $frm->getVar("jk"));
      $pasangan->Set("kepribadian1", $frm-
>getVar("kepribadian1"));
      $pasangan->Set("kepribadian2", $frm-
>getVar("kepribadian2"));
            $pasangan->Set("kecocokan", $frm-
>getVar("kecocokan"));
            $pasangan->Set("keterangan", $frm-
>getVar("keterangan"));

            // execution
      $tpl->set_page_template("browse");
      if ($pasangan->Insert()) {
            $tpl->assign("message", MSG_INSERT_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            } else {
            $tpl->assign("message", MSG_INSERT_FAILED, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
            $frm->setVar("action", "browse");
      }

}
```

MsPasanganPersonality.php

```
<?php
require_once "../../web/inc/begin.php";
require_once "../../web/entities/MsPasanganPersonality.php";
$pasangan = new MsPasanganPersonality();

if ($frm->getVar("submit_add")) {

      $isFormValid = true;
      if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", true);
            $frm->setVar("action", "repost");
      } else {
            $pasangan->Set();
            $pasangan->Set("idn", $frm->getVar("idn"));
            $pasangan->Set("jk", $frm->getVar("jk"));
      $pasangan->Set("kepribadian1", $frm-
>getVar("kepribadian1"));
      $pasangan->Set("kepribadian2", $frm-
>getVar("kepribadian2"));
            $pasangan->Set("kecocokan", $frm-
>getVar("kecocokan"));
            $pasangan->Set("keterangan", $frm-
>getVar("keterangan"));

            $tpl->set_page_template("browse");
            if ($pasangan->Insert()) {
            $tpl->assign("message", MSG_INSERT_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            } else {
            $tpl->assign("message", MSG_INSERT_FAILED, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
            $frm->setVar("action", "browse");
      }

}
//update
if ($frm->getVar("submit_update")) {
      $isFormValid = true;
      if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", false);
            $frm->setVar("action", "repost");
      } else {

            $pasangan->Set();
            $pasangan->Set("idn", $frm->getVar("idn"));
            $pasangan->Set("jk", $frm->getVar("jk"));
      $pasangan->Set("kepribadian1", $frm-
>getVar("kepribadian1"));
      $pasangan->Set("kepribadian2", $frm-
>getVar("kepribadian2"));
            $pasangan->Set("kecocokan", $frm-
>getVar("kecocokan"));
            $pasangan->Set("keterangan", $frm-
>getVar("keterangan"));

            if ($pasangan->UpdateById($frm->getVar("id"))) {
                  $tpl->set_page_template("edit");
                  $tpl->assign("isAdd", false);
                  $frm->setVar("action", "repost");
            $tpl->assign("message", MSG_UPDATE_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            } else {
                  $tpl->assign("isAdd", false);
                  $tpl->set_page_template("edit");
            $tpl->assign("message", MSG_UPDATE_DELETE, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
      }

}
// Delete Many

if ($frm->getVar("action")=="submit_delete_many") {

      $frm->setVar("action", "browse");
      $num_success = 0;
      $num_failed = 0;

      for ($i=0; $i<$frm->getVar("chkItemNum"); $i++) {
            $a = $frm->getVar("chk".$i);
            if (!empty($a)) {
                  $num_item++;
                  if ($pasangan->DeleteById($a)) {
                        $num_success++;
                  } else {
                        $num_success++;
                  }
            }
      }
            if ($num_success==$num_item) {
            $tpl->set_page_template($frm->getVar("action"));
            $tpl->assign("message", MSG_DELETE_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
      } else {
$tpl->assign("message", MSG_DELETE_FAILED." $num_failed dari
$num_item Data gagal dihapus.", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
      }

}
if ($frm->getVar("action")=="browse") {
$tpl->set_page_template($frm->getVar("action"));
      $_order = $sgn_poas->GetOrder();
      $_where = $sgn_poas->GetSearch(true);

      $total_record = $pasangan->GetSelectCount($_where);
      $current_page = $frm->getVar("p");
      $row_per_page = BROWSE_PERPAGE;
$paging = new SgnPaging($total_record, $current_page,
$row_per_page);

      $pasangan->Execute(
"SELECT MPP.*, MK1.nama nama_kepribadian1, MK2.nama
nama_kepribadian2
FROM ms_pasangan_personality MPP
JOIN ms_kepribadian MK1 ON MK1.id_kepribadian=MPP.kepribadian1
JOIN ms_kepribadian MK2 ON MK2.id_kepribadian=MPP.kepribadian2
LIMIT ".$paging->get("offset").",$row_per_page");

      $line_count = 0;
      while ($pasangan->MoveNext()) {
            $tpl->assign_list("list", array(
      "line_count"=>$line_count++,
      "idn"=>$pasangan->Get("idn"),
      "jk"=>($pasangan->Get("jk")=="L") ? "Laki-laki" :
"Perempuan",
                  "kepribadian1"=>$pasangan-
>Get("nama_kepribadian1"),
                  "kepribadian2"=>$pasangan-
>Get("nama_kepribadian2"),
                  "kecocokan"=>$pasangan->Get("kecocokan"),
                  "keterangan"=>nl2br($pasangan-
>Get("keterangan")),
                  ));
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->register_list("list");
}
require_once "../../web/inc/end.php";
?>
```

MsPertanyaanEneagram.php

```
<?php

require_once "../../web/inc/begin.php";
require_once "../../web/entities/MsKepribadian.php";
require_once "../../web/entities/MsPertanyaanEneagram.php";

$kepribadian = new MsKepribadian();
$eneagram = new MsPertanyaanEneagram();

// Vars
$tpl->set_page_template($frm->getVar("action"));
$arraySearchKeyList = array("idn"=>"ID
soal","id_kepribadian"=>"Nama Kepribadian","no_soal"=>"Nomor
soal");

// Get Kepribadian
$kepribadian->Execute("SELECT id_kepribadian, nama FROM
ms_kepribadian WHERE jenis='eneagram' ORDER BY id_kepribadian");
while($kepribadian->MoveNext()) {
$arrayKepribadianList[$kepribadian->Get("id_kepribadian")] =
$kepribadian->Get("nama");
}
if ($frm->getVar("submit_add")) {

      $isFormValid = true;
      if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", true);
            $frm->setVar("action", "repost");
      } else {
            $eneagram->Set();
            $eneagram->Set("idn", $frm->getVar("idn"));
$eneagram->Set("id_kepribadian", $frm->getVar("id_kepribadian"));
            $eneagram->Set("pertanyaan", $frm-
>getVar("pertanyaan"));
            $eneagram->Set("no_soal", $frm->getVar("no_soal"));
            $eneagram->Set("keterangan", $frm-
>getVar("keterangan"));
```

```
            $tpl->set_page_template("browse");
            if ($eneagram->Insert()) {
            $tpl->assign("message", MSG_INSERT_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            } else {
            $tpl->assign("message", MSG_INSERT_FAILED, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
            $frm->setVar("action", "browse");
      }
}
if ($frm->getVar("submit_update")) {

      $isFormValid = true;
      if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", false);
            $frm->setVar("action", "repost");
      } else {
            $eneagram->Set();
            $eneagram->Set("idn", $frm->getVar("idn"));
$eneagram->Set("id_kepribadian", $frm->getVar("id_kepribadian"));
            $eneagram->Set("pertanyaan", $frm-
>getVar("pertanyaan"));
            $eneagram->Set("no_soal", $frm->getVar("no_soal"));
            $eneagram->Set("keterangan", $frm-
>getVar("keterangan"));

            if ($eneagram->UpdateById($frm->getVar("id"))) {
                  $tpl->set_page_template("edit");
                  $tpl->assign("isAdd", false);
                  $frm->setVar("action", "repost");
            $tpl->assign("message", MSG_UPDATE_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            } else {
                  $tpl->assign("isAdd", false);
                  $tpl->set_page_template("edit");
            $tpl->assign("message", MSG_UPDATE_DELETE, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
      }

}
//delete
if ($frm->getVar("action")=="submit_delete") {
      if ($eneagram->DeleteById($frm->getVar("id"))) {
            $frm->setVar("action", "browse");
            $tpl->set_page_template($frm->getVar("action"));
      $tpl->assign("message", MSG_DELETE_SUCCESS, "Message");
      $tpl->assign("message_type_class", "MsgInfo", "Message");
      } else {
            $tpl->assign("message", MSG_DELETE_FAILED);
            $tpl->assign("message_type_class", "MsgError",
"Message");
      }
}

// Delete Many

if ($frm->getVar("action")=="submit_delete_many") {

      $frm->setVar("action", "browse");
      $num_success = 0;
      $num_failed = 0;

      for ($i=0; $i<$frm->getVar("chkItemNum"); $i++) {
            $a = $frm->getVar("chk".$i);
            if (!empty($a)) {
                  $num_item++;
                  if ($eneagram->DeleteById($a)) {
                        $num_success++;
                  } else {
                        $num_success++;               }
            }
      }

      if ($num_success==$num_item) {
      $tpl->set_page_template($frm->getVar("action"));
      $tpl->assign("message", MSG_DELETE_SUCCESS, "Message");
      $tpl->assign("message_type_class", "MsgInfo", "Message");
      } else {
      $tpl->assign("message", MSG_DELETE_FAILED." $num_failed dari
$num_item Data gagal dihapus.", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
      }

}

if ($frm->getVar("action")=="browse") {

      $tpl->set_page_template($frm->getVar("action"));
      $_order = $sgn_poas->GetOrder();
      $_where = $sgn_poas->GetSearch(true);

      $total_record = $eneagram->GetSelectCount($_where);
      $current_page = $frm->getVar("p");
      $row_per_page = BROWSE_PERPAGE;
$paging = new SgnPaging($total_record, $current_page,
$row_per_page);

$eneagram->Select("", $_where, $_order, $row_per_page, $paging-
>get("offset"));
      $line_count = 0;
      while ($eneagram->MoveNext()) {
            $tpl->assign_list("list", array(
                  "line_count"=>$line_count++,
                  "idn"=>$eneagram->Get("idn"),
                  "id_kepribadian"=>$kepribadian-
>GetNamaById($eneagram->Get("id_kepribadian")),
                  "pertanyaan"=>$eneagram->Get("pertanyaan"),
                  "no_soal"=>$eneagram->Get("no_soal"),
                  "keterangan"=>nl2br($eneagram-
>Get("keterangan")),

                  ));
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->register_list("list");
require_once "../../web/inc/end.php";

?>
```

MsPertanyaanPersonality.php

```
<?php

require_once "../../web/inc/begin.php";
require_once "../../web/entities/MsKepribadian.php";
require_once "../../web/entities/MsPertanyaanPersonality.php";

$kepribadian = new MsKepribadian();
$personality = new MsPertanyaanPersonality();

// Add
if ($frm->getVar("submit_add")) {
$isFormValid = true;
if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", true);
            $frm->setVar("action", "repost");
      } else {

$personality->Set();
$personality->Set("idn", $frm->getVar("idn"));
$personality->Set("id_kepribadian", $frm-
>getVar("id_kepribadian"));
$personality->Set("pertanyaan", $frm->getVar("pertanyaan"));
$personality->Set("no_soal", $frm->getVar("no_soal"));
$personality->Set("no_urut_soal", $frm->getVar("no_urut_soal"));
$personality->Set("kelompok", $frm->getVar("kelompok"));
$personality->Set("keterangan", $frm->getVar("keterangan"));

$tpl->set_page_template("browse");
      if ($personality->Insert()) {
            $tpl->assign("message", MSG_INSERT_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            } else {
            $tpl->assign("message", MSG_INSERT_FAILED, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
            $frm->setVar("action", "browse");
      }
}
// Update
if ($frm->getVar("submit_update")) {
      $isFormValid = true;
      if (!$isFormValid) {
            $tpl->set_page_template("edit");
            $tpl->assign("isAdd", false);
            $frm->setVar("action", "repost");
      } else {
            $personality->Set();
            $personality->Set("idn", $frm->getVar("idn"));
$personality->Set("id_kepribadian", $frm-
>getVar("id_kepribadian"));
$personality->Set("pertanyaan", $frm->getVar("pertanyaan"));
$personality->Set("no_soal", $frm->getVar("no_soal"));
$personality->Set("no_urut_soal", $frm->getVar("no_urut_soal"));
$personality->Set("kelompok", $frm->getVar("kelompok"));
$personality->Set("keterangan", $frm->getVar("keterangan"));

            if ($personality->UpdateById($frm->getVar("id"))) {
                  $tpl->set_page_template("edit");
                  $tpl->assign("isAdd", false);
                  $frm->setVar("action", "repost");
            $tpl->assign("message", MSG_UPDATE_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            } else {
                  $tpl->assign("isAdd", false);
                  $tpl->set_page_template("edit");
            $tpl->assign("message", MSG_UPDATE_DELETE, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
      }

}
//add form

if ($frm->getVar("action")=="add") {

      // get no_soal max
$personality->Execute("SELECT MAX(no_soal) AS no_soal_max FROM
ms_pertanyaan_personality");
      $no_soal_max = $personality->Get('no_soal_max');

      // get no_urut_soal max
$personality->Execute("SELECT MAX(no_urut_soal) AS
no_urut_soal_max FROM ms_pertanyaan_personality WHERE
no_soal='$no_soal_max'");
$no_urut_soal_max = $personality->Get('no_urut_soal_max');

      // set new no_soal & no_urut_soal
if ($no_urut_soal_max<4) {
      $no_soal_new = $no_soal_max;
      $no_urut_soal_new = $no_urut_soal_max + 1;
      } else {
            $no_soal_new = $no_soal_max + 1;
            $no_urut_soal_new = 1;
      }
      $tpl->set_page_template("edit");
      $tpl->assign("isAdd", true);
if ($frm->getVar("action")=="browse") {

      // Vars
      $tpl->set_page_template($frm->getVar("action"));
      $_order = $sgn_poas->GetOrder();
      $_where = $sgn_poas->GetSearch(true);
      $total_record = $personality->GetSelectCount($_where);
      $current_page = $frm->getVar("p");
      $row_per_page = BROWSE_PERPAGE;
$paging = new SgnPaging($total_record, $current_page,
$row_per_page);

      // Select, Fetch, Assign List
      $personality->Select("", $_where, $_order, $row_per_page,
$paging->get("offset"));
      $line_count = 0;
      while ($personality->MoveNext()) {
      $tpl->assign_list("list", array(
            "line_count"=>$line_count++,
            "idn"=>$personality->Get("idn"),
"id_kepribadian"=>$kepribadian->GetNamaById($personality-
>Get("id_kepribadian")),
            "pertanyaan"=>$personality->Get("pertanyaan"),
            "no_soal"=>$personality->Get("no_soal"),
            "no_urut_soal"=>$personality->Get("no_urut_soal"),
            "kelompok"=>$personality->Get("kelompok"),
            "keterangan"=>nl2br($personality->Get("keterangan")),
                 ));
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->register_list("list");
}
require_once "../../web/inc/end.php";

?>
```

Register.php

```
<?php
require_once "../../web/inc/begin.php";
require_once "../../web/entities/MasterMember.php";


if ($frm->getVar("action")=="")
      $frm->setVar("action","add");

$member = new MasterMember();
$thisyear = date("Y");
for($thn=$thisyear; $thn>($thisyear-100); $thn--) {
      $arrayTglLahirThn[$thn] = $thn;
}
$arrayTglLahirBln[1] = "Januari";
$arrayTglLahirBln[2] = "Pebruari";
$arrayTglLahirBln[3] = "Maret";
```

```
$arrayTglLahirBln[4] = "April";
$arrayTglLahirBln[5] = "Mei";
$arrayTglLahirBln[6] = "Juni";
$arrayTglLahirBln[7] = "Juli";
$arrayTglLahirBln[8] = "Agustus";
$arrayTglLahirBln[9] = "September";
$arrayTglLahirBln[10] ="Oktober";
$arrayTglLahirBln[11] ="November";
$arrayTglLahirBln[12] ="Desember";


$thisdate = date("D");
for($tgl=1; $tgl<=31; $tgl++) {
      $arrayTglLahirTgl[$tgl] = $tgl;
}
$arrayStatus['S'] = "Single";
$arrayStatus['R'] = "In Relationship";
$arrayStatus['M'] = "Married";
$arrayStatus['D'] = "Divorced";

// Add
if ($frm->getVar("action")=="submit_add") {

      $isFormValid = true;
      $a = $frm->getVar("nama");
      if (empty($a)) {
$tpl->assign("message", MSG_INSERT_FAILED.": Nama Pengguna harus
diisi", "Message");
      $tpl->assign("message_type_class", "MsgError", "Message");
      $isFormValid = false; // flag
      }
      $a = $frm->getVar("username");
      if (empty($a)) {
      $tpl->assign("message", MSG_INSERT_FAILED.": Username harus
diisi", "Message");
      $tpl->assign("message_type_class", "MsgError", "Message");
      $isFormValid = false; // flag
      }

      // check user exist
$member->Execute("SELECT COUNT(*) AS num FROM ms_member WHERE
username='".$frm->getVar("username")."'");
      if ($member->Get("num")>0) {
$tpl->assign("message", "Nama Pengguna Telah Terpakai. Silakan isi
yang lain", "Message");
$tpl->assign("message_type_class", "MsgError", "Message");
      $isFormValid = false;
      }

require_once "../../web/utils/sgnUpload.php";
$upload = new sgnUpload($_FILES['photo']);
$upload->set_upload_dir(UPLOAD_FOTO_PATH);
$upload->set_max_file_size(BACKGROUND_MAX_FILE_SIZE);

if (!$upload->upload()){
$tpl->assign("message", MSG_INSERT_FAILED.": Foto tidak dapat
diupload", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
```

```
            $upload_success = false;
      } else {
            $upload_success = true;
      }
      if (!$isFormValid) {
            //$tpl->set_page_template("edit");
            $tpl->assign("isAdd", true);
            $frm->setVar("action", "repost");
      } else {

            $member->Set();
            $member->Set("role", "member");
            $member->Set("username", $frm->getVar("username"));
            $member->Set("password", SgnUser::SgnUserEncrypt($frm-
>getVar("password_new1")));
            $member->Set("nama", $frm->getVar("nama"));
            $member->Set("jk", $frm->getVar("jk"));
            $member->Set("tgl_lhr", $frm-
>getVar("tgl_lahir_thn")."-".$frm->getVar("tgl_lahir_bln")."-
".$frm->getVar("tgl_lahir_tgl"));
            $member->Set("alamat", $frm->getVar("alamat"));
            $member->Set("Provinsi", $frm->getVar("Provinsi"));
            $member->Set("Negara", $frm->getVar("Negara"));
            $member->Set("email", $frm->getVar("email"));
            $member->Set("hp", $frm->getVar("hp"));
            $member->Set("pkrjaan", $frm->getVar("pkrjaan"));
            $member->Set("status", $frm->getVar("status"));
            $member->Set("wrn_rmbut", $frm->getVar("wrn_rmbut"));
            $member->Set("tggi_bdn", $frm->getVar("tggi_bdn"));
            $member->Set("brt_bdn", $frm->getVar("brt_bdn"));
            $member->Set("agama", $frm->getVar("agama"));
            $member->Set("pendidikan", $frm-
>getVar("pendidikan"));
            $member->Set("ket_pribadi", $frm-
>getVar("ket_pribadi"));
            $member->Set("mkn_fav", $frm->getVar("mkn_fav"));
            $member->Set("film_fav", $frm->getVar("film_fav"));
            $member->Set("musik_fav", $frm->getVar("musik_fav"));
            $member->Set("hobi", $frm->getVar("hobi"));

      if ($upload_success)
            $member->Set("photo", $upload->get_file_name());
            if ($member->Insert()) {
            $tpl->assign("message", MSG_INSERT_SUCCESS,
"Message");
            $tpl->assign("message_type_class", "MsgInfo",
"Message");
            $frm->setVar("action", "browse");

            $_SESSION['User_username'] = $frm->getVar("username");
            $_SESSION['User_password'] = $user-
>SgnUserEncrypt($frm->getVar("password_new1"));
                  header("Location: ../Public/MenuUtama.php");
            } else {
            $tpl->assign("message", MSG_INSERT_FAILED, "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
            }
      }
```

```
}
require_once "../../web/inc/end.php";

?>
```

CariMember.php

```
<?php
require_once "../../web/inc/begin.php";
require_once "../../web/entities/MasterMember.php";
require_once('../../web/entities/SgnUserSL.php');
$user_auth = new SgnUserSL();
$user_auth>SetNotAuthorizedPage('../../modules/Member/NotAuthorize
d.php');

if ($_SESSION['User_username'] && $_SESSION['User_password']) {
      $_SESSION['User_password'] is encrypted password
      $username = $_SESSION['User_username'];
      $password = $_SESSION['User_password'];

      if (!$user_auth->Authenticate($username,$password,0)) {
            header("Location: ".$user_auth->GetNotAuthorizedPage);
      }
} else {
      header("Location: ".$user_auth->GetNotAuthorizedPage);
}
if ($frm->getVar("action")=="browse") {

      $tpl->set_page_template($frm->getVar("action"));
      $_order = $sgn_poas->GetOrder();

      // CJ custom where
      $_where = "";
      if ($frm->getVar("pekerjaan")) {
      $_where .= " AND pkrjaan LIKE '%".$frm-
>getVar("pekerjaan")."%'";
      }
      if ($frm->getVar("filter_nama")) {
      $_where .= " AND nama LIKE '%".$frm-
>getVar("filter_nama")."%'";
      }
      if ($frm->getVar("umur1") && $frm->getVar("umur2")) {
      $_where .= " AND YEAR(NOW())-YEAR(tgl_lhr)>=".$frm-
>getVar("umur1")." AND YEAR(NOW())-YEAR(tgl_lhr)<=".$frm-
>getVar("umur2");
      }
      if ($frm->getVar("agama")) {
            $_where .= " AND agama LIKE '%".$frm-
>getVar("agama")."%'";
      }
      if ($frm->getVar("hobi")) {
            $_where .= " AND hobi LIKE '%".$frm-
>getVar("hobi")."%'";
      }
      if (empty($_where)) {
            $_where = " username=''";
      } else {
            $_where = " username<>''" . $_where;
      }
      // CJ custom total_record
$member->Execute("SELECT COUNT(*) AS num FROM ms_member WHERE
$_where");
      $total_record = $member->Get("num");
      $current_page = $frm->getVar("p");
      $row_per_page = BROWSE_PERPAGE;
      $paging = new SgnPaging($total_record, $current_page,
$row_per_page);

      $member->Execute("SELECT * FROM ms_member WHERE $_where");
      $line_count = 0;
      while ($member->MoveNext()) {
            $tpl->assign_list("list", array(
                  "line_count"=>$line_count++,
                  "id_member"=>$member->Get("id_member"),
                  "username"=>$member->Get("username"),
                  "nama"=>$member->Get("nama"),
                  "jk"=>$member->Get("jk"),
                  "tgl_lhr"=>$member->Get("tgl_lhr"),
                  "alamat"=>$member->Get("alamat"),
                  "email"=>$member->Get("email"),
                  "hp"=>$member->Get("hp"),
                  "agama"=>$member->Get("agama"),
                  "pkrjaan"=>$member->Get("pkrjaan")
                        )
                  );
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->register_list("list");

require_once "../../web/inc/end.php";
?>
```

MsgInbox.php

```
<?php
require_once "../../web/inc/begin.php";
// read Inbox

$user->ValidatePageAccess('member,admin');
if ($frm->getVar("action")=="browse") {

      require_once "../../web/entities/TrMessage.php";
      $message = new TrMessage();
      $_order = $sgn_poas->GetOrder();
      $_where = array("msgto"=>$user->GetIdMember());

      // Paging Setup
      $total_record = $message->GetSelectCount($_where);
      $current_page = $frm->getVar("p");
      $row_per_page = BROWSE_PERPAGE;
$paging = new SgnPaging($total_record, $current_page,
$row_per_page);

      // Select, Fetch, Assign List
$message->Select("", $_where, $_order, $row_per_page, $paging-
>get("offset"));
      $line_count = 0;
      while ($message->MoveNext()) {

            $member1 = new MasterMember();
            $member1->SelectById($message->Get("msgfrom"));
            $tpl->assign_list("list", array(
                  "line_count"=>$line_count++,
                  "idn"=>$message->Get("idn"),
                  "status"=>$message->Get("flag"),
                  "waktu"=>$message->Get("waktu"),
                  "msgto"=>$message->Get("msgto"),
                  "msgfrom"=>$message->Get("msgfrom"),
                  "dari"=>$member1->Get("nama"),
                  "subject"=>$message->Get("subject"),
                  "message"=>$message->Get("message"),
                        )
                  );
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->register_list("list");
      require_once "../../web/inc/end.php";

?>
```

MsgRead.php

```
<?php
require_once "../../web/inc/begin.php";
$user->ValidatePageAccess('member,admin');

require_once "../../web/entities/TrMessage.php";
$msg = new TrMessage();
$msg->SelectById($frm->getVar("id"));
if ($user->GetIdMember()==$msg->Get("msgto")) {

      // nama pengirim
      require_once "../../web/entities/MasterMember.php";
      $member = new MasterMember();
      $member->SelectById($msg->Get("msgfrom"));

      // display data
      $tpl->assign("id_member", $msg->Get("msgfrom"));
      $tpl->assign("waktu", $msg->Get("waktu"));
      $tpl->assign("judul", $msg->Get("subject"));
      $tpl->assign("message", $msg->Get("message"));
      $tpl->assign("dari", $member->Get("nama"));

      // update status
      require_once "../../web/entities/TrMessage.php";
      $msg = new TrMessage();
      $msg->Set("flag", "R");
      $msg->UpdateById($frm->getVar("id"));
}
require_once "../../web/inc/end.php";
?>
```

MsgWrite.php

```php
<?php
require_once "../../web/inc/begin.php";

$user->ValidatePageAccess('member,admin');
$tpl->set_page_template("Utama");

// Write Message
if ($frm->getVar("send_message")) {
      require_once "../../web/entities/MasterMember.php";
      $member = new MasterMember();
      $member->SelectById($frm->getVar("msgto"));
      $tpl->assign("nama", $member->Get("nama"));
      $tpl->assign("msgto", $frm->getVar("msgto"));
}

// Send Message
if ($frm->getVar("action")=="Kirim") {
      require_once "../../web/entities/TrMessage.php";
      $message = new TrMessage();
      $message->Set();
      $message->Set("waktu", date("Y-m-d H:i"));
      $message->Set("msgfrom", $user->GetIdMember());
      $message->Set("msgto", $frm->getVar("msgto"));
      $message->Set("subject", $frm->getVar("subject"));
      $message->Set("message", $frm->getVar("message"));
      $message->Set("flag", "N");
      $message->Insert();

      if (strstr($_SERVER["HTTP_REFERER"],"Profile")) {
            header("Location: Profile.php?id=".$frm-
>getVar("msgto"));
      } else {
            header("Location: MsgInbox.php");
      }
}
require_once "../../web/inc/end.php";
?>
```

MySoulmate.php

```php
<?php
require_once "../../web/inc/begin.php";

$user->ValidatePageAccess('member,admin');
$user->GetIdMember();
$tpl->set_page_template("Utama");

//approve
if ($frm->getVar("action")=="approve_soulmate") {

      require_once "../../web/entities/TrSoulmate.php";
      $soulmate = new TrSoulmate();

      $frm->setVar("action", "browse");

      // update status
      $soulmate->Set();
      $soulmate->Set('is_aprv','T');
      $result = $soulmate->UpdateById($frm->getVar('idn'));
      if ($result) {
      $tpl->assign("message", "Status sudah diupdate.",
"Message");
      $tpl->assign("message_type_class", "MsgInfo", "Message");
      } else {
      $tpl->assign("message", "Status gagal diupdate", "Message");
      $tpl->assign("message_type_class", "MsgError", "Message");
      }
}
if ($frm->getVar("action")=="reject_soulmate") {

      require_once "../../web/entities/TrSoulmate.php";
      $soulmate = new TrSoulmate();
      $frm->setVar("action", "browse");

      // update status
      $soulmate->Set();
      $soulmate->Set('is_aprv','R');
      $result = $soulmate->UpdateById($frm->getVar('idn'));

      if ($result) {
$tpl->assign("message", "Status sudah diupdate.", "Message");
$tpl->assign("message_type_class", "MsgInfo", "Message");
      } else {
      $tpl->assign("message", "Status gagal diupdate", "Message");
            $tpl->assign("message_type_class", "MsgError",
"Message");
      }

}
//reject
if ($frm->getVar("action")=="reject_soulmate") {

      require_once "../../web/entities/TrSoulmate.php";
      $soulmate = new TrSoulmate();

      $frm->setVar("action", "browse");

      // update status
      $soulmate->Set();
      $soulmate->Set('is_aprv','R');
      $result = $soulmate->UpdateById($frm->getVar('idn'));

      if ($result) {

      $tpl->assign("message", "Status sudah diupdate.",
"Message");
      $tpl->assign("message_type_class", "MsgInfo", "Message");
      } else {
      $tpl->assign("message", "Status gagal diupdate", "Message");
      $tpl->assign("message_type_class", "MsgError", "Message");
      }

}

// Delete
if ($frm->getVar("action")=="delete_soulmate") {
      require_once "../../web/entities/TrSoulmate.php";
      $soulmate = new TrSoulmate();

      $frm->setVar("action", "browse");

      $soulmate->Set();
      $soulmate->Set('is_aprv','D');
      $result = $soulmate->UpdateById($frm->getVar('idn'));

      if ($result) {
      $tpl->assign("message", "Status sudah diupdate.",
"Message");
      $tpl->assign("message_type_class", "MsgInfo", "Message");
      } else {
      $tpl->assign("message", "Status gagal diupdate", "Message");
      $tpl->assign("message_type_class", "MsgError", "Message");
      }

}

// Read Soulmate

if ($frm->getVar("action")=="browse") {
      // get my gender
      require_once "../../web/entities/MasterMember.php";
      $member = new MasterMember();
      $member->SelectById($user->GetIdMember());
      $my_gender = $member->Get('jk');

      require_once "../../web/entities/TrSoulmate.php";
      $soulmate = new TrSoulmate();
      $_order = $sgn_poas->GetOrder();
      if ($my_gender=="L") {
            $_where = array("co"=>$user->GetIdMember());
      } else {
            $_where = array("ce"=>$user->GetIdMember());
      }

      // Paging Setup
      $total_record = $soulmate->GetSelectCount($_where);
      $current_page = $frm->getVar("p");
      $row_per_page = BROWSE_PERPAGE;
$paging = new SgnPaging($total_record, $current_page,
$row_per_page);

      // Select, Fetch, Assign List
$soulmate->Select("", $_where, $_order, $row_per_page, $paging-
>get("offset"));
      $line_count = 0;
      while ($soulmate->MoveNext()) {

            $member1 = new MasterMember();
            if ($my_gender=='L') {
                  $member1->SelectById($soulmate->Get("ce"));
            } else {
                  $member1->SelectById($soulmate->Get("co"));
            }
            $tpl->assign_list("list", array(
                  "line_count"=>$line_count++,
                  "idn"=>$soulmate->Get("idn"),
                  "pasangan"=>$member1->Get('nama'),
                  "pasanganid"=>$member1->Get('id_member'),
                  "reqname"=>$member1->Get('nama'),
      "status"=> $soulmate->GetStatus($soulmate->Get("is_aprv")),
      "display_approve" => ($user->GetIdMember()==$soulmate-
>Get("request_by")) ? 'none' : $soulmate-
>GetDisplayStyle($soulmate->Get("is_aprv"), "approve"),
"display_reject" => ($user->GetIdMember()==$soulmate-
>Get("request_by")) ? 'none' : $soulmate-
>GetDisplayStyle($soulmate->Get("is_aprv"), "reject"),
"display_delete" => ($user->GetIdMember()==$soulmate-
>Get("request_by")) ? 'none' : $soulmate-
>GetDisplayStyle($soulmate->Get("is_aprv"), "delete"),
                  )
            );
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->register_list("list");
require_once "../../web/inc/end.php";

?>
```

ReqSoulmate.php

```
<?php

require_once "../../web/inc/begin.php";
$user->ValidatePageAccess('member,admin');

$tpl->set_page_template("Utama"); // template

if ($frm->getVar("request_soulmate")) {
      require_once "../../web/entities/TrSoulmate.php";
      $soulmate = new TrSoulmate();
      $soulmate->Set();

      if ($frm->getVar("gender")=="L") {
            $soulmate->Set("co",$frm->getVar("reqfrom"));
            $soulmate->Set("ce",$frm->getVar("reqto"));
      } else {
            $soulmate->Set("co",$frm->getVar("reqto"));
            $soulmate->Set("ce",$frm->getVar("reqfrom"));
      }
      $soulmate->Set("is_aprv","F");
      $soulmate->Set("request_by",$frm->getVar("reqfrom"));
      $soulmate->Insert();

}
require_once "../../web/inc/end.php";
?>
```

Profile.php

```
<?php

require_once "../../web/inc/begin.php";
$user->ValidatePageAccess('member,admin');
```

```
$tpl->set_page_template("Utama"); // template

$userid = $frm->getVar("id"); // baca dari query string

require_once "../../web/entities/MasterMember.php";
$member = new MasterMember();

// get my gender
$member->Select("", array("id_member"=>$user->GetIdMember()));
$my_gender = $member->Get('jk');

// get target profile
$member->Select("", array("id_member"=>$userid));

// calculate  umur
$time_now = time();
$time_birth = strtotime($member->Get("tgl_lhr"));
$time_diff = $time_now - $time_birth;
$umur = date("Y",$time_diff)-1970;

// request soulmate
require_once "../../web/entities/TrSoulmate.php";
$soulmate = new TrSoulmate();

if ($my_gender != $member->Get("jk")) {
      if ($my_gender == "L") {
            $soulmate->Select("is_aprv", array("co"=>$user-
>GetIdMember(),"ce"=>$userid));
      } else {
$soulmate->Select("is_aprv", array("ce"=>$user-
>GetIdMember(),"co"=>$userid));
      }

      $showRequestSoulmate = true;
      while ($soulmate->MoveNext()) {
            if ($soulmate->Get('is_aprv')=='T') {
                  $showRequestSoulmate = false;
            }
      }
}

require_once "../../web/entities/MsMemberFoto.php";
$foto = new MsMemberFoto();
$foto->Execute("SELECT * FROM tr_member WHERE
id_member='".$member->Get("id_member")."'");
while ($foto->MoveNext()) {
      $tpl->assign_list("list", array(
            "img"=>UPLOAD_FOTO_PATH."/".$foto->Get("foto"),
            ));
}

$tpl->assign("reqto", $userid);
$tpl->assign("reqfrom", $user->GetIdMember());
$tpl->assign("gender", $my_gender);
$tpl->assign("ShowRequestSoulmate", $showRequestSoulmate);

require_once "../../web/inc/end.php";
?>
```

StartTest.php

```
<?php
require_once "../../web/inc/begin.php";
$user->ValidatePageAccess('member,admin');

require_once "../../web/entities/MsKepribadian.php";
require_once "../../web/entities/MsPertanyaanPersonality.php";
require_once "../../web/entities/TrJawabanPersonality.php";
require_once "../../web/entities/MsPertanyaanEneagram.php";
require_once "../../web/entities/TrJawabanEneagram.php";

// Start Test personality
if ($frm->getVar("TestPersonality")) {

      $jawaban_personality = new TrJawabanPersonality();
      $pertanyaan_personality = new MsPertanyaanPersonality();
      $jawaban_personality->Execute("SELECT COUNT(*) AS num FROM
tr_jawaban_personality WHERE user_id=".$user->GetIdMember());
      if ($jawaban_personality->Get("num")>0) {
            // cari no soal yg blm dijawab
            $jawaban_personality->Execute("SELECT no_urut FROM
tr_jawaban_personality WHERE jawaban=0 AND user_id=".$user-
>GetIdMember()." ORDER BY no_urut ASC LIMIT 0,1");
            $next_question = $jawaban_personality->Get("no_urut")
- 1;

            // back to first question
            if ($next_question==-1) $next_question = 0;
            header("Location:
TestPersonality.php?p=$next_question");
            exit;
      }
      $num_soal = $pertanyaan_personality->GetJumlahSoal();
      $random_soal_order = array();

      for ($i=1; $i<=$num_soal; $i++) {
            $random_soal = rand(1,$num_soal);
            while (array_search($random_soal, $random_soal_order))
{
                  $random_soal = rand(1,$num_soal);
            }
            $random_soal_order[$i] = $random_soal;
      }
      for ($i=1; $i<=$num_soal; $i++) {
            $jawaban_personality->Set();
      $jawaban_personality->Set("user_id", $user->GetIdMember());
            $jawaban_personality->Set("no_urut", $i);
      $jawaban_personality->Set("no_soal",
$random_soal_order[$i]);
            $jawaban_personality->Set("jawaban", 0);
            $jawaban_personality->Insert();
      }

      header("Location: TestPersonality.php");
}

// Start Test Eneagram
```

```
if ($frm->getVar("TestEneagram")) {

      $jawaban_eneagram = new TrJawabanEneagram();
      $pertanyaan_eneagram = new MsPertanyaanEneagram();
      $kepribadian = new MsKepribadian();

      // cek apakah user ini sudah punya set jawaban
      $jawaban_eneagram->Execute("SELECT COUNT(*) AS num FROM
tr_jawaban_eneagram WHERE user_id=".$user->GetIdMember());
      if ($jawaban_eneagram->Get("num")>0) {
            // get soal yg belum dijawab
            $jawaban_eneagram = new TrJawabanEneagram();
            $jawaban_eneagram->Execute("SELECT COUNT(*) AS num
FROM tr_jawaban_eneagram WHERE user_id=".$user->GetIdMember()."
AND (jawaban='Y' OR jawaban='T')");
            $terjawab = $jawaban_eneagram->Get("num");
            $nextpage = ($terjawab / 10) + 1;
      if ($terjawab>=$jawaban_eneagram->Get("num")) {
      header("Location: TestEneagram.php?p=1");
                  exit;
            } else {
      header("Location: TestEneagram.php?p=$nextpage");
                  exit;
            }
      }


      $num_soal = $pertanyaan_eneagram->GetJumlahSoal();
      // random soal
      $random_soal_order = array();
      for ($i=1; $i<=$num_soal; $i++) {
            $random_soal = rand(1,$num_soal);
            while (array_search($random_soal, $random_soal_order))
{
                  $random_soal = rand(1,$num_soal);
            }
            $random_soal_order[$i] = $random_soal;
      }
// generate jawaban
      for ($i=1; $i<=$num_soal; $i++) {
            $pertanyaan_eneagram-
>SelectById($random_soal_order[$i]);
            $jawaban_eneagram->Set();
            $jawaban_eneagram->Set("user_id", $user-
>GetIdMember());
            $jawaban_eneagram->Set("id_kepribadian",
$pertanyaan_eneagram->Get('id_kepribadian'));
            $jawaban_eneagram->Set("no_urut", $i);
            $jawaban_eneagram->Set("no_soal",
$pertanyaan_eneagram->Get("id_kepribadian").$pertanyaan_eneagram-
>Get("no_soal"));
            $jawaban_eneagram->Set("jawaban", "X");
            $jawaban_eneagram->Insert();
      }
      header("Location: TestEneagram.php");
}
require_once "../../web/inc/end.php";
```

```
?>
```

```
TestPersonality.php
```

```
<?php

require_once "../../web/inc/begin.php";
$user->ValidatePageAccess('member,admin');

require_once "../../web/entities/MsKepribadian.php";
require_once "../../web/entities/TrJawabanPersonality.php";

$kepribadian = new MsKepribadian();

// Next Question
if ($frm->getVar("test_next")) {
      // paging
      $p = $frm->getVar("p");
      $frm->setVar("p",$p+1);

      // get answer
      $jawaban = $frm->getVar("pertanyaan");
      $idn = $frm->getVar("idn");
      $jawaban_personality = new TrJawabanPersonality();
      $jawaban_personality->Set("jawaban", $jawaban);
      $jawaban_personality->Update(array("idn"=>$idn));

      $frm->setVar("action","browse");
      // last question

      $jawaban_personality->Execute("SELECT COUNT(*) num FROM
tr_jawaban_personality WHERE user_id='".$user->GetIdMember()."'
GROUP BY user_id");
      $last_question = $jawaban_personality->Get("num");
      if (($p+1) == $last_question) {
            // calculate result
            $id_member = $user->GetIdMember();

require_once "../../web/entities/TrJawabanPersonality.php";
            $personality = new TrJawabanPersonality();
            $max_personality = 0;
            $personality->Execute( "select P.kelompok,
P.id_kepribadian, J.jawaban, count(J.idn) as num
from tr_jawaban_personality J join ms_pertanyaan_personality P ON
P.no_soal = J.no_soal and P.id_kepribadian=concat('p',J.jawaban)
where J.user_id=$id_member and J.jawaban>0
group by P.kelompok, J.jawaban");
while($personality->MoveNext()) {
$tpl->assign($personality->Get("kelompok")."_".$personality-
>Get("id_kepribadian"), $personality->Get("num"));
${"total_p".substr($personality->Get("id_kepribadian"), -1)} +=
$personality->Get("num");
if (${"total_p".$max_personality}<${"total_p".substr($personality-
>Get("id_kepribadian"), -1)}) {
$max_personality = substr($personality->Get("id_kepribadian"), -
1);
                  }
            }
// update member's personality
require_once "../../web/entities/MasterMember.php";
            $member = new MasterMember();
            $member->Set();
            $member->Set("result_personality_p1", $total_p1);
            $member->Set("result_personality_p2", $total_p2);
            $member->Set("result_personality_p3", $total_p3);
            $member->Set("result_personality_p4", $total_p4);
            $member->Set("result_personality",
"p".$max_personality);
            $member->UpdateById($id_member);
header("Location:
../Report/TestPersonalityRecomendation.php?action=send");
      }
}

// Browse

if ($frm->getVar("action")=="browse") {

      // Select, Fetch, Assign List
      $jawaban_personality = new TrJawabanPersonality();
$jawaban_personality->GetSoal($user->GetIdMember(),$frm-
>getVar("p"));
      $line_count = 0;
      while ($jawaban_personality->MoveNext()) {
            $tpl->assign_list("list", array(
                  "line_count"=>$line_count++,
                  "no_soal"=>$jawaban_personality->Get("no_soal"),
            "no_urut_soal"=>$jawaban_personality-
>Get("no_urut_soal"),
            "pertanyaan"=>$jawaban_personality->Get("pertanyaan"),
                 ));
            $tpl->assign("DataNotEmpty", true);
            $no_urut = $jawaban_personality->Get("no_urut");
            $idn = $jawaban_personality->Get("idn");
      }
      $tpl->assign("line_total", $line_count);
      $tpl->assign("p", $frm->getVar("p"));
      $tpl->assign("no_urut", $no_urut);
      $tpl->assign("idn", $idn);
      $tpl->register_list("list");
}
require_once "../../web/inc/end.php";
?>
```

TestEneagram.php

```
<?php
require_once "../../web/inc/begin.php";
$user->ValidatePageAccess('member,admin');

require_once "../../web/entities/MsKepribadian.php";
require_once "../../web/entities/TrJawabanEneagram.php";

$kepribadian = new MsKepribadian();

// Next Question
if ($frm->getVar("test_next")) {
      // pindah halaman
      $p = $frm->getVar("p");
      $pt = $frm->getVar("pt");
      // simpan jawaban
      $jawaban = $frm->getVars();
      foreach($jawaban as $key=>$value) {
            if (substr($key,0,3)=="tny") {
                  $idn = $value;
                  $soal_id = substr($key,3,6);
                  $jawaban_eneagram = new TrJawabanEneagram();
$jawaban_eneagram->Set("jawaban",
(strtolower($jawaban["jwb".$soal_id])=="on") ? "Y" : "T");
                  $jawaban_eneagram->Update(array("idn"=>$idn));
            }
      }
      // last page.. create summary
      if ($p==$pt) {
            $jawaban_eneagram->CreateSummary($user-
>GetIdMember());
      header("Location: ../Report/TestEneagramRecomendation.php");
      }
      // pindah halaman
      if ($p<$pt) $frm->setVar("p",$p+1);

}
if ($frm->getVar("action")=="browse") {

      $jawaban_eneagram = new TrJawabanEneagram();
      $kepribadian = new MsKepribadian();


      $tpl->set_page_template($frm->getVar("action"));
      $_order = $sgn_poas->GetOrder();
      $_where = array("user_id"=>$user->GetIdMember());

      $total_record = $jawaban_eneagram->GetSelectCount($_where);
      $current_page = $frm->getVar("p");
      $row_per_page = 10;
      $paging = new SgnPaging($total_record, $current_page,
$row_per_page);

      // Select, Fetch, Assign List
$jawaban_eneagram-
>SelectFrom("J.idn,J.user_id,J.id_kepribadian,J.no_urut,
J.no_soal,J.jawaban,S.pertanyaan,S.keterangan",
"tr_jawaban_eneagram J LEFT JOIN ms_pertanyaan_eneagram S ON
S.id_kepribadian=substr(J.no_soal,1,2) AND
S.no_soal=substr(J.no_soal,3,2)", $_where, $_order, $row_per_page,
$paging->get("offset"));
      $line_count = 0;
      while ($jawaban_eneagram->MoveNext()) {
            $tpl->assign_list("list", array(
                  "line_count"=>$line_count++,
                  "idn"=>$jawaban_eneagram->Get("idn"),
"id_kepribadian"=>$jawaban_eneagram->Get("id_kepribadian"),
"id_kepribadian_nama"=>$kepribadian-
>GetNamaById($jawaban_eneagram->Get("id_kepribadian")),
                  "pertanyaan"=>$jawaban_eneagram-
>Get("pertanyaan"),
                  "no_soal"=>$jawaban_eneagram->Get("no_soal"),
            "keterangan"=>nl2br($jawaban_eneagram-
>Get("keterangan")),
                  ));
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->assign("p", $frm->getVar("p"));
      $tpl->assign("pt", $paging->get_total_pages());
      $tpl->register_list("list");
}
require_once "../../web/inc/end.php";
?>
```

MyPersonality.php

```
<?php
require_once "../../web/inc/begin.php";
$tpl->set_page_template("Utama");
require_once "../../web/entities/MsKepribadian.php";
require_once "../../web/entities/TrJawabanPersonality.php";

$tpl->assign("K1_p1", "0");
$tpl->assign("K1_p2", "0");
$tpl->assign("K1_p3", "0");
$tpl->assign("K1_p4", "0");
$tpl->assign("K2_p1", "0");
$tpl->assign("K2_p2", "0");
$tpl->assign("K2_p3", "0");
$tpl->assign("K2_p4", "0");

$total_p1 = 0;
$total_p2 = 0;
$total_p3 = 0;
$total_p4 = 0;

$id_member = $user->GetIdMember();
$personality = new TrJawabanPersonality();
$max_personality = 0;
$personality->Execute(
"select P.kelompok, P.id_kepribadian, J.jawaban, count(J.idn) as
num
from tr_jawaban_personality J join ms_pertanyaan_personality P ON
P.no_soal = J.no_soal and P.id_kepribadian=concat('p',J.jawaban)
where J.user_id=$id_member and J.jawaban>0
group by P.kelompok, J.jawaban");
while($personality->MoveNext()) {
      $tpl->assign($personality->Get("kelompok")."_".$personality-
>Get("id_kepribadian"), $personality->Get("num"));
      ${"total_p".substr($personality->Get("id_kepribadian"), -1)}
+= $personality->Get("num");
if (${"total_p".$max_personality}<${"total_p".substr($personality-
>Get("id_kepribadian"), -1)}) {
max_personality = substr($personality->Get("id_kepribadian"), -1);
      }
}
$tpl->assign("total_p1", $total_p1);
$tpl->assign("total_p2", $total_p2);
$tpl->assign("total_p3", $total_p3);
```

```
$tpl->assign("total_p4", $total_p4);
$tpl->assign("persen_p1", sprintf("%01.0f",
$total_p1/40*100)."%");
$tpl->assign("persen_p2", sprintf("%01.0f",
$total_p2/40*100)."%");
$tpl->assign("persen_p3", sprintf("%01.0f",
$total_p3/40*100)."%");
$tpl->assign("persen_p4", sprintf("%01.0f",
$total_p4/40*100)."%");

// Get Dominant Personality
$member1 = new MasterMember();
$member1->Select("result_personality");

$dominantPersonality = $member1->GetDominantPersonality($user-
>GetIdMember());

$targetPersonality = array();
foreach ($dominantPersonality as $pers) {
      $targetPersonality[] = $member1-
>GetTargetPersonality($pers);
}

require_once "../../web/entities/MsKepribadian.php";
$kepribadian = new MsKepribadian();

foreach($dominantPersonality as $pers=>$score) {
      $kepribadian->SelectById($pers);

      $tpl->assign_list("list", array(
            "personality_name" => $kepribadian->Get("nama"),
            "personality_description" => $kepribadian-
>Get("keterangan")
      ));
}
require_once "../../web/inc/end.php";
?>
```

MyEneagram.php

```
<?php
require_once "../../web/inc/begin.php";
$tpl->set_page_template("Utama");

require_once "../../web/entities/MasterMember.php";
require_once "../../web/entities/TrJawabanEneagram.php";

$member = new MasterMember();
$member->Select("", array("id_member"=>$user->GetIdMember()));
$tpl->assign("score_e1", $member->Get("result_eneagram_e1") / 20 *
100);
$tpl->assign("score_e2", $member->Get("result_eneagram_e2") * 5);
$tpl->assign("score_e3", $member->Get("result_eneagram_e3") * 5);
$tpl->assign("score_e4", $member->Get("result_eneagram_e4") * 5);
$tpl->assign("score_e5", $member->Get("result_eneagram_e5") * 5);
$tpl->assign("score_e6", $member->Get("result_eneagram_e6") * 5);
$tpl->assign("score_e7", $member->Get("result_eneagram_e7") * 5);
$tpl->assign("score_e8", $member->Get("result_eneagram_e8") * 5);
$tpl->assign("score_e9", $member->Get("result_eneagram_e9") * 5);
```

```
$member1 = new MasterMember();
$member1->Select("result_eneagram");

$dominantEneagram = $member1->GetDominantEneagram($user-
>GetIdMember());

$targetEneagram = array();
foreach ($dominantEneagram as $pers) {
      $targetEneagram[] = $member1->GetTargetEneagram($pers);
}

require_once "../../web/entities/MsKepribadian.php";
$kepribadian = new MsKepribadian();

foreach($dominantEneagram as $pers=>$score) {
      $kepribadian->SelectById($pers);

      $tpl->assign_list("list", array(
            "eneagram_name" => $kepribadian->Get("nama"),
            "eneagram_description" => $kepribadian-
>Get("keterangan")
      ));
}
require_once "../../web/inc/end.php";
?>
```

CariPasanganPersonality.php

```
require_once "../../web/inc/begin.php";
$tpl->set_page_template("Utama");
require_once "../../web/entities/MsKepribadian.php";
require_once "../../web/entities/TrJawabanPersonality.php";

$arrayPersonality['p1'] = "Sanguinis";
$arrayPersonality['p2'] = "Koleris";
$arrayPersonality['p3'] = "Melankolis";
$arrayPersonality['p4'] = "Plegmatis";

// Cari Pasangan
$tpl->assign("target_personality",
$option>generate_input_option("target_personality",$arrayPersonali
ty,$frm->getVar("target_personality")));

$target_personality = $frm->getVar("target_personality");
if ($target_personality) {

      require_once "../../web/entities/MasterMember.php";
      $member = new MasterMember();

      // get user's gender
      $member->Select("jk", array("id_member"=>$user-
>GetIdMember()));
      $my_gender = $member->Get("jk");
      $partner_gender = ($my_gender=="L") ? "P" : "L";

      // set filter untuk cari pasangan: personality & gender
$sgn_poas->SetSearch("result_personality", $frm-
>getVar("target_personality"));
```

```
$sgn_poas->SetSearch("jk", $partner_gender);
      //$sgn_poas->SetSearch(array("jk"=>$partner_gender,
"result_personality"=>$frm->getVar("target_personality")));
      $_order = $sgn_poas->GetOrder();
      $_where = $sgn_poas->GetSearch(true);

      // Paging Setup
      $total_record = $member->GetSelectCount($_where);
      $current_page = $frm->getVar("p");
      $row_per_page = BROWSE_PERPAGE;
$paging = new SgnPaging($total_record, $current_page,
$row_per_page);
$member->Select("", $_where, $_order, $row_per_page, $paging-
>get("offset"));
      $line_count = 0;
      while ($member->MoveNext()) {
      // calculate  umur
      $time_now = time();
      $time_birth = strtotime($member->Get("tgl_lhr"));
      $time_diff = $time_now - $time_birth;
      $umur = date("Y",$time_diff)-1970;
            $tpl->assign_list("list", array(
                  "line_count"=>$line_count++,
                  "id_member"=>$member->Get("id_member"),
                  "username"=>$member->Get("username"),
                  "nama"=>$member->Get("nama"),
                  "jk"=>$member->Get("jk"),
                  "umur1"=>$umur,
                  "tggi_bdn"=>$member->Get("tggi_bdn"),
                  "Provinsi"=>$member->Get("Provinsi"),
                  "ket_pribadi"=>$member->Get("ket_pribadi"),
                  "brt_bdn"=>$member->Get("brt_bdn"),
                  "hobi"=>$member->Get("hobi"),
                  "foto"=>$member->Get("Photo")
                        )
                  );
            $tpl->assign("DataNotEmpty", true);
      }
      $tpl->assign("line_total", $line_count);
      $tpl->register_list("list");
// keterangan pasangan
require_once "../../web/entities/MsPasanganPersonality.php";
$pasanganPersonality = new MsPasanganPersonality();

      $member1 = new MasterMember();
      $member1->Select("result_personality",
array("id_member"=>$user->GetIdMember()));
      $kepribadian1 = $member1->Get("result_personality");
      $kepribadian2 = $target_personality;

      $pasanganPersonality->Select("keterangan",
array("kepribadian1"=>$kepribadian1,
"kepribadian2"=>$kepribadian2));
      $tpl->assign("keterangan_pasangan", $pasanganPersonality-
>Get("keterangan"));

      // nama kepribadian
require_once "../../web/entities/MsKepribadian.php";
$kepribadian = new MsKepribadian();
```

```
$kepribadian->Select("nama",
array("id_kepribadian"=>$kepribadian1));
$tpl->assign("kepribadian1", $kepribadian->Get("nama"));
$kepribadian->Select("nama",
array("id_kepribadian"=>$kepribadian2));
      $tpl->assign("kepribadian2", $kepribadian->Get("nama"));
require_once "../../web/inc/end.php";
?>
```

TestPersonalityRecomendation.php

```
<?php
require_once "../../web/inc/begin.php";

$tpl->set_page_template("Utama"); // template

require_once "../../web/entities/MsKepribadian.php";
require_once "../../web/entities/TrJawabanPersonality.php";

$arrayPersonality['p1'] = "Sanguinis";
$arrayPersonality['p2'] = "Koleris";
$arrayPersonality['p3'] = "Melankolis";
$arrayPersonality['p4'] = "Plegmatis";

// Get Dominant Personality
$member1 = new MasterMember();
$member1->Select("result_personality");
$dominantPersonality = $member1->GetDominantPersonality($user-
>GetIdMember());
$targetPersonality = array();
foreach ($dominantPersonality as $pers=>$score) {
      $targetPersonality[] = $member1-
>GetTargetPersonality($pers);
}
// Cari Pasangan

$tpl->assign("target_personality", $option-
>generate_input_option("target_personality",$arrayPersonality,$frm
->getVar("target_personality")));

if (true) {

      require_once "../../web/entities/MasterMember.php";
      $member = new MasterMember();

      // get user's gender & personality
$member->Select("jk,result_personality", array("id_member"=>$user-
>GetIdMember()));
      $my_gender = $member->Get("jk");
      $target_gender = ($my_gender=="L") ? "P" : "L";
      $target_personality = $member->GetTargetPersonality($member-

      >Get("result_personality"));

      // Select, Fetch, Assign List
      $member->Execute("SELECT * FROM ms_member WHERE
jk='$target_gender' AND
result_personality='$target_personality'");
      $line_count = 0;
      while ($member->MoveNext()) {

            // send message: jika action=='send'
            if ($frm->getVar("action")=='send') {
                  require_once "../../web/entities/TrMessage.php";
                  $message = new TrMessage();
$str_message = "Lihat profile <a
href=\"../Member/Profile.php?id=".$user-
>GetIdMember()."\">".$user->GetUsername()."</a>";
                  $message->Set();
                  $message->Set("waktu", date("Y-m-d H:i"));
                  $message->Set("msgfrom", 1);
                  $message->Set("msgto", $member-
>Get("id_member"));
      $message->Set("subject", "Pasangan yang cocok untuk kamu");
                  $message->Set("message", $str_message);
                  $message->Set("flag", "N");
                  $message->Insert();
            }

      $time_now = time();
      $time_birth = strtotime($member->Get("tgl_lhr"));
      $time_diff = $time_now - $time_birth;
      $umur = date("Y",$time_diff)-1970;
      $tpl->assign_list("list", array(
            "id_member"=>$member->Get("id_member"),
            "nama"=>$member->Get("nama"),
            "umur1"=>$umur,
            "foto"=>$member->Get("Photo"),
            "tggi_bdn"=>$member->Get("tggi_bdn"),
            "Provinsi"=>$member->Get("Provinsi"),
            "hobi"=>$member->Get("hobi"),
            "brt_bdn"=>$member->Get("brt_bdn"),
            "ket_pribadi"=>$member->Get("ket_pribadi")
            ));
      $tpl->assign("DataNotEmpty", true);
}
$tpl->assign("line_total", $line_count);
$tpl->register_list("list");

      if ($frm->getVar("action")=='send') {
      header("Location:
../Report/TestPersonalityRecomendation.php");
            exit;
      }

require_once "../../web/entities/MsKepribadian.php";
$kepribadian = new MsKepribadian();

      foreach($dominantPersonality as $pers=>$score) {
      $kepribadian->SelectById($pers);

      $tpl->assign_list("list1", array(
            "kepribadian" => $kepribadian->Get("nama"),
            "keterangan" => $kepribadian->Get("keterangan")
      ));
}

      // keterangan pasangan
      require_once "../../web/entities/MsPasanganPersonality.php";
      $pasanganPersonality = new MsPasanganPersonality();

      $member1 = new MasterMember();
$member1->Select("result_personality", array("id_member"=>$user-
>GetIdMember()));
      $kepribadian1 = $member1->Get("result_personality");
      $kepribadian2 = $target_personality;

      $pasanganPersonality->Select("keterangan",
array("kepribadian1"=>$kepribadian1,
"kepribadian2"=>$kepribadian2));
      $tpl->assign("keterangan_pasangan", $pasanganPersonality-
>Get("keterangan"));

      // nama kepribadian
      require_once "../../web/entities/MsKepribadian.php";
      $kepribadian = new MsKepribadian();
$kepribadian->Select("nama",
array("id_kepribadian"=>$kepribadian1));
      $tpl->assign("kepribadian1", $kepribadian->Get("nama"));
$kepribadian->Select("nama",
array("id_kepribadian"=>$kepribadian2));
      $tpl->assign("kepribadian2", $kepribadian->Get("nama"));
            }else {
      $tpl->register_list("list");
}

$str_dominant_personality = "";
$separator = "";
foreach ($dominantPersonality as $persId=>$persScore) {
      $str_dominant_personality .= $separator . $kepribadian-
>GetNamaById($persId) . " ($persScore)";
      $separator = " dan ";
}
$tpl->assign("dominant_personality", $str_dominant_personality);

$str_target_personality = "";
$separator = "";

foreach ($targetPersonality as $persId) {
      $str_target_personality .= $separator . $kepribadian-
>GetNamaById($persId);
      $separator = " dan ";
}
$tpl->assign("target_personality", $str_target_personality);
require_once "../../web/inc/end.php";
?>
```

TestEneagramRecomendation.php

```
<?php
require_once "../../web/inc/begin.php";
$tpl->set_page_template("Utama"); // template

require_once "../../web/entities/MsKepribadian.php";
require_once "../../web/entities/TrJawabanEneagram.php";

$arrayEneagram['e1'] = "Perfeksionis";
```

```
$arrayEneagram['e2'] = "Penolong";
$arrayEneagram['e3'] = "Pengejar Prestasi";
$arrayEneagram['e4'] = "Romantis";
$arrayEneagram['e5'] = "Pengamat";
$arrayEneagram['e6'] = "Pencemas";
$arrayEneagram['e7'] = "Petualang";
$arrayEneagram['e8'] = "Pejuang";
$arrayEneagram['e9'] = "Pendamai";

// Get Dominant Eneagram
$member1 = new MasterMember();
$member1->Select("result_eneagram");

$dominantEneagram = $member1->GetDominantEneagram($user-
>GetIdMember());

$targetEneagram = array();
foreach ($dominantEneagram as $pers=>$score) {
      $targetEnagram[] = $member1->GetTargetEneagram($pers);
}

require_once "../../web/entities/MsKepribadian.php";
$kepribadian = new MsKepribadian();

foreach($dominantEneagram as $pers=>$score) {
      $kepribadian->SelectById($pers);

      $tpl->assign_list("list1", array(
            "eneagram_name" => $kepribadian->Get("nama"),
            "eneagram_description" => $kepribadian-
>Get("keterangan")
      ));
}

// Cari Pasangan
$tpl->assign("target_eneagram",$option-
>generate_input_option("target_eneagram",$arrayE,$frm-
>getVar("target_eneagram")));

$member = new MasterMember();
$member->GetPasanganEneagram($user->GetIdMember());
while($member->MoveNext()) {

// send message: jika action=='send'
      if ($frm->getVar("action")=='send') {
            require_once "../../web/entities/TrMessage.php";
            $message = new TrMessage();
            $str_message = "Lihat profile <a
href=\"../Member/Profile.php?id=".$user-
>GetIdMember()."\">".$user->GetUsername()."</a>";
            $message->Set();
            $message->Set("waktu", date("Y-m-d H:i"));
            $message->Set("msgfrom", 1);
            $message->Set("msgto", $member->Get("id_member"));
      $message->Set("subject", "Pasangan yang cocok untuk kamu");
            $message->Set("message", $str_message);
            $message->Set("flag", "N");
            $message->Insert();
      }
// calculate  umur
      $time_now = time();
      $time_birth = strtotime($member->Get("tgl_lhr"));
      $time_diff = $time_now - $time_birth;
      $umur = date("Y",$time_diff)-1970;
      $tpl->assign_list("list", array(
            "id_member"=>$member->Get("id_member"),
            "nama"=>$member->Get("nama"),
            "umur1"=>$umur,
            "foto"=>$member->Get("Photo"),
            "tggi_bdn"=>$member->Get("tggi_bdn"),
            "Provinsi"=>$member->Get("Provinsi"),
            "hobi"=>$member->Get("hobi"),
            "brt_bdn"=>$member->Get("brt_bdn"),
            "ket_pribadi"=>$member->Get("ket_pribadi")
            ));
      $tpl->assign("DataNotEmpty", true);
}
$tpl->register_list("list");

// redirect untuk menampilkan halaman ini tanpa send
if ($frm->getVar("action")=='send') {
header("Location:
../Test/TestEneagramRecomendation.php?action=send");
      exit;
}

require_once "../../web/entities/MsKepribadian.php";
$kepribadian = new MsKepribadian();

$str_dominant_eneagram = "";
$separator = "";
foreach ($dominantEneagram as $persId=>$persScore) {
      $str_dominant_eneagram .= $separator . $kepribadian-
>GetNamaById($persId) . " ($persScore)";
      $separator = " dan ";
}
$tpl->assign("dominant_eneagram", $str_dominant_eneagram);

$str_target_eneagram = "";
$separator = "";
foreach ($targetEneagram as $persId) {
      $str_target_eneagram .= $separator . $kepribadian-
>GetNamaById($persId);
      $separator = " dan ";
}
$tpl->assign("target_eneagram", $str_target_eneagram);
require_once "../../web/inc/end.php";
?>
```

Login.php

```
<?php
require_once "../../web/inc/begin.php";

if ($frm->getVar("Logout")==1) {
      $_SESSION['User_username'] = "";
      $_SESSION['User_password'] = "";

      header("Location: Login.php");
      exit;
}

if ($frm->getVar("submit_login")) {

      $username = $frm->getVar("username");
      $password = $frm->getVar("password");

      $login_user = new SgnUserCJ();
      if ($login_user->Authenticate($username,$password)) {
$tpl->assign("message", "Username atau Password benar.",
"Message");
$tpl->assign("message_type_class", "MsgError", "Message");

      $_SESSION['User_username'] = $username;
      $_SESSION['User_password'] =
SgnUser::SgnUserEncrypt($password);
            header("Location: ../Public/MenuUtama.php");
            exit;
      } else {
$tpl->assign("message", "Username atau Password salah.",
"Message");
$tpl->assign("message_type_class", "MsgError", "Message");
      }

}

$tpl->assign("username", "");
$tpl->assign("password", "");
$tpl->assign("@UserBox", "", $tpl->_master_template);

require_once "../../web/inc/end.php";

?>
```

ChangePassword.php

```
<?php
require_once "../../web/inc/begin.php";


if ($frm->getVar("action")=='submit_chpwd') {

      $password_old = $frm->getVar("password_old");
      $password_new1 = $frm->getVar("password_new1");

      if
(SgnUser::SgnUserEncrypt($password_old)!=$_SESSION['User_password'
]) {
      $tpl->assign("message", "Password lama tidak cocok",
"Message");
      $tpl->assign("message_type_class", "MsgError", "Message");

      } else {
            if ($user->ChangePassword($password_new1)) {
      $tpl->assign("message", "Password sudah diubah", "Message");
      $tpl->assign("message_type_class", "MsgInfo", "Message");

$_SESSION['User_password'] =
SgnUser::SgnUserEncrypt($password_new1);
            } else {
$tpl->assign("message", "Password tidak dapat diubah.",
"Message");
      $tpl->assign("message_type_class", "MsgError", "Message");
            }

      }

}

$tpl->assign("username", "");
$tpl->assign("password", "");
require_once "../../web/inc/end.php";

?>
```