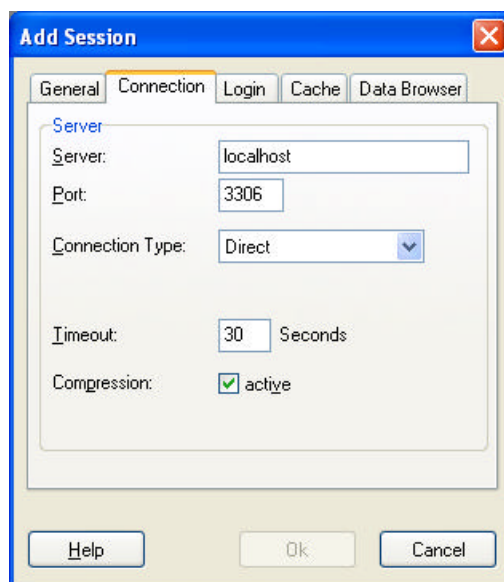


4. IMPLEMENTASI SISTEM

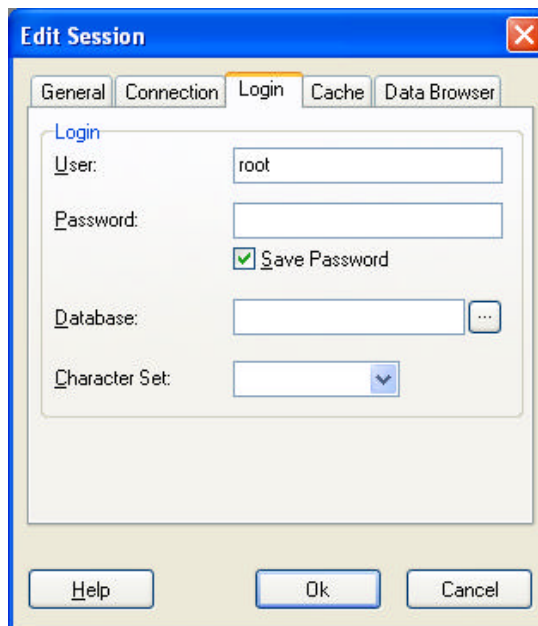
4.1. Setting Awal

Setting awal dilakukan dengan penginstalan Apache 2.05 sebagai *web server*, MySQL 4.1.9 sebagai *database server*, MySQL Front 3.2 sebagai GUI dari *database server*, PHP 4.3.10 untuk pemrograman *website* dan Macromedia Dreamweaver MX sebagai tempat kerja dan tempat untuk mendesain *website* yang akan dibuat.

Setelah menginstall *Apache*, *MySQL* dan *MySQL Front* maka akan dilakukan pembuatan *database* pada *MySQL* melalui *MySQL Front*. Pertama kali adalah menciptakan koneksi ke *database server* dengan cara memasukkan *IP address* dari *database server* pada *tab connection* yang dalam hal ini adalah 'localhost' kemudian memasukkan *user name* dan *password* pada *tab login* yang dalam hal ini adalah 'root' dan '', seperti dilihat pada Gambar 4.1 dan Gambar 4.2.

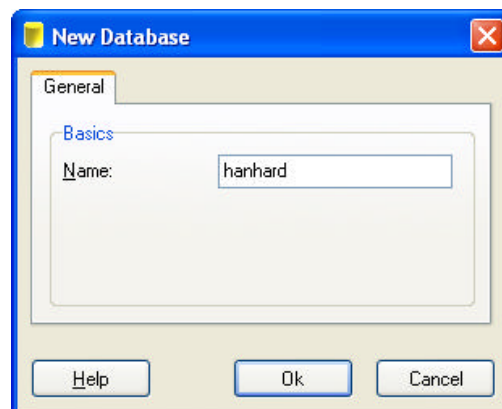


Gambar 4.1. Halaman *Connection MySQL*



Gambar 4.2. Halaman *Login MySQL*

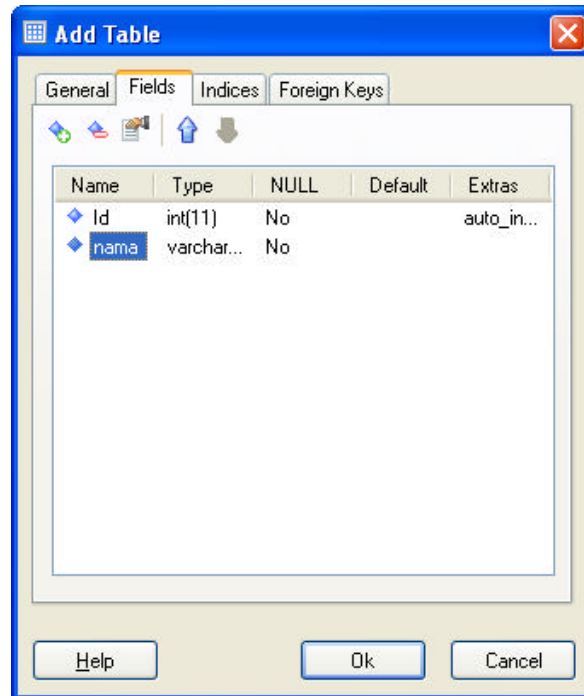
Langkah berikutnya adalah membuat *database* dengan cara klik kanan pada 'localhost' kemudian pilih *new > database*, kemudian masukkan nama *database* yang diinginkan yang dalam sistem ini adalah 'hanhard', seperti terlihat pada Gambar 4.3.



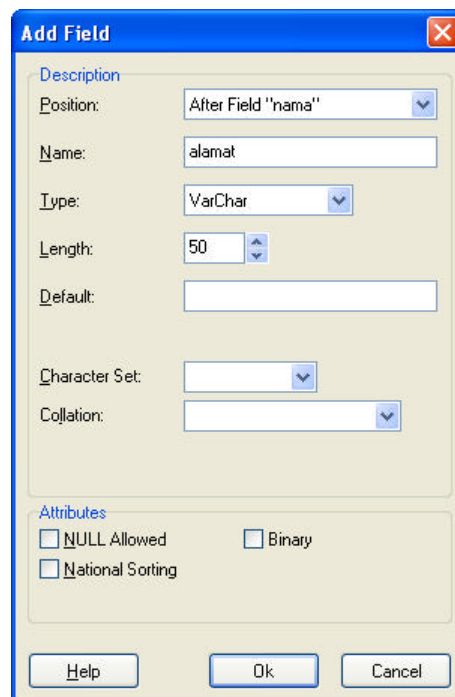
Gambar 4.3. Halaman *Create Database*

Langkah selanjutnya adalah dilakukan pembuatan tabel – tabel yang diperlukan untuk sistem, dengan cara klik kanan pada 'hanhard' kemudian pilih *new > table*. Kemudian masukkan nama tabel yang diinginkan dan masukkan juga

field yang dikehendaki beserta tipe datanya, Seperti terlihat pada gambar 4.4. dan Gambar 4.5.



Gambar 4.4. Halaman Pembuatan Tabel



Gambar 4.5. Halaman Penambahan *Field*

4.2. Koneksi Database Dari PHP

Koneksi ke *database* digunakan untuk menghubungkan aplikasi *website* penjualan *online* dengan *database* yang digunakan. *Script* yang digunakan untuk koneksi antara *database* dan *website* penjualan *online* adalah:

```
<?
    $DB_HOST = "localhost";
    $DB_DBNM = "hanhard";
    $DB_USER = "root";$DB_PASS = "";
    mysql_connect($DB_HOST, $DB_USER, $DB_PASS);
    mysql_select_db($DB_DBNM);
?>
```

Maksud dari perintah diatas adalah melakukan koneksi dari php ke *database server* di 'localhost' dengan user 'root' dan password ''. Kemudian memilih *database* 'hanhard' untuk diaktifkan, sehingga dapat dimanipulasi isinya.

4.3. Implementasi Program Website Penjualan Online

Pada sub bab ini akan dibahas mengenai pemrograman yang terdapat dalam halaman *website*. Berikut ini daftar menu yang terdapat pada halaman *website* beserta dengan letak *script* pemrogramannya pada segmen program.

Tabel 4.1. Tabel Daftar Menu *Website* dan Segmen Program

No.	Menu	Segmen Program
1	Inisialisasi	4.1.
2	Konstanta	4.2.
3	Fungsi	4.3.
4	<i>Paging</i>	4.4.
5	<i>Login</i>	4.5.
6	Registrasi	4.6.
7	Keranjang Belanja	4.7.
8	<i>CheckOut</i>	4.8.
9	Sejarah Belanja	4.9.
10	Konfirmasi Pembayaran	4.10.
11	Ubah Password	4.11.
12	Ubah Data Pribadi	4.12.
13	Lupa Password	4.13.
14	Administrasi Barang	4.14.
15	<i>LogOut</i>	4.15.

Pada tiap halaman di bagian atasnya terdapat semua kesamaan pemrograman yaitu beberapa baris inisialisasi awal, seperti melakukan start *session*, mengambil semua variabel GET dan POST untuk mempermudah pengkodean, melakukan koneksi ke *database*, deklarasi fungsi – fungsi buatan sendiri, deklarasi *class paging* untuk membantu membuat *paging* yang dipakai hampir di keseluruhan halaman *website*.

Segmen Program 4.1. Inisialisasi

```
<?
    session_start();
    import_request_variables("gp","v_");
    import_request_variables("p","p_");
    include("includes/conndb.php");
    include("includes/const.php");
    include("includes/function.php");
    include("includes/c_paging.php");
?>
```

Inisialisasi konstanta adalah pendefinisian konstanta – konstanta yang akan digunakan pada hampir keseluruhan halaman, seperti gambar dari tombol – tombol, pesan *error*, dan lain sebagainya. Penggunaan konstanta ini ditujukan agar mempercepat proses pengkodean karena menulis menjadi singkat, yang kedua adalah supaya bila ada perubahan seperti gambar maka proses penggantian gambar tidak perlu dilakukan pada setiap halaman melainkan pada satu konstanta saja, mengingat halaman *website* yang banyak sekali hal ini akan sangat mempermudah pengkode. Beberapa contoh konstanta yang diinisialisasikan dapat dilihat pada Segmen Program 4.2.

Segmen Program 4.2. Konstanta

```
<?
define("WRONG_PASSWORD", "Password salah");
define("NO_SUCH_ID", "ID tidak ada");
define("NOT_SIGNED", "Silahkan login dahulu");
define("BRG_QTY_OVER", "Jumlah barang tidak mencukupi");
define("PLEASE_LOGIN", "Silahkan Login");
define("SEARCH_NOT_FOUND", "Tidak ada data yang sesuai");
define("REFRESH_BTN", "<img border='0' alt='refresh'
src='./images/refresh.gif'>");
define("DELETE_BTN", "<img border='0' alt='delete'
src='./images/delete2.gif'>");
define("UPDATE_BTN", "<img border='0' alt='update'
src='./images/update.gif'>");
define("ADD_BTN", "<img border='0' alt='add'
src='./images/add.gif'>");
define("ASC_BTN", "<img border='0' alt='ascending'
src='./images/ascending.gif'>");
define("DESC_BTN", "<img border='0' alt='descending'
src='./images/descending.gif'>");
?>
```

Pada awal inisialisasi juga dideklarasikan beberapa fungsi yang nantinya akan dipakai pada halaman – halaman *website*. Seperti fungsi untuk memarsing dan membuat *query string*, pengacakan *password* baru, penonaktifan inputan menggunakan *javascript*, dan beberapa lainnya seperti pada Segmen Program 4.3.

Segmen Program 4.3. Fungsi

```
<?
function new_pswd($count=8) {
    $tmpl = "ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
    $new = "";
    for ($i=0;$i<$count;$i++) $new .=
substr($tmpl,rand(0,strlen($tmpl)),1);
    return $new;
}
function parse_qs($qs,$decode=true) {
    $result = array();
    $temp = array();
    $temp = explode("&",$qs);
    foreach($temp as $value) {
        $result[substr($value,0,strpos($value,"="))] =
($decode === true) ?
urldecode(substr($value,strpos($value,"")+1)) :
substr($value,strpos($value,"")+1);
    }
    return $result;
}
```

Segmen Program 4.3. Fungsi (Lanjutan)

```

function make_qs($arr_qs,$encode=true) {
    $temp = array();
    foreach($arr_qs as $key=>$value) {
        $temp[] = ($encode === true) ?
            $key."=".urlencode($value) :
            "$key=$value";
    }
    return implode("&",$temp);
}
function jsDisable() {
    echo
        "<script language=\"javascript\">".
        "var i;".
        "function disable(obj,val) {".
            "if (obj.length > 1 && obj.tagName != 'SELECT') {".
                "for (i=0;i<obj.length;i++) {".
                    "obj[i].disabled = val;".
                    "if (obj[i].checked && !val)
                        obj[i].focus();".
                "}".
            "}" else {".
                "obj.disabled = val;".
                "if (obj.checked && !val) obj.focus();".
            "}".
        "}".
        "</script>";
}
?>

```

Pada hampir tiap halaman di *website* ini menggunakan fungsi *paging* yaitu suatu fungsi untuk membuat tampilan data yang banyak sekali menjadi beberapa halaman yang bisa dengan mudah dinavigasikan, seperti halaman produk yang mengandung 24 data, ditampilkan menjadi 5 halaman yang berisi 5 data pada setiap halamannya. Karena penggunaan yang bermacam – macam bentuk maka dibuatlah sebuah *class* agar dalam pemakaiannya dapat dimodifikasi dengan leluasa, seperti jumlah data per halamannya, *query string* yang menjadi *link*, *style* tulisan yang dipakai, tidak tampilnya navigasi jika jumlah halaman = 1.

Segmen Program 4.4. *Paging*

```

<?
define("PAGE_DEFAULT",1);
define("PAGESIZE_DEFAULT",5);
define("PAGESIZE_LOWER_LIMIT",1);
define("PAGESIZE_UPPER_LIMIT",50);
define("PHP_SELF",$_SERVER['PHP_SELF']);
define("QS",$_SERVER['QUERY_STRING']);

class pagingClass {
    var $page;
    var $pagesize;
    var $total_rows;
    var $last_page;
    var $qs;
    var $style;

    function pagingClass($total_rows,$qs=QS) {
        $this->page = PAGE_DEFAULT;
        $this->pagesize = PAGESIZE_DEFAULT;
        $this->total_rows = $total_rows;
        $this->last_page =
            ceil($this->total_rows/$this->pagesize);
        $this->qs = $qs;
        $this->style = "";
        if ($this->page < 1) $this->page = PAGE_DEFAULT;
        if ($this->last_page < 1)
            $this->last_page = PAGE_DEFAULT;
        if ($this->page > $this->last_page)
            $this->page = $this->last_page;
        if ($this->pagesize < PAGESIZE_LOWER_LIMIT)
            $this->pagesize = PAGESIZE_LOWER_LIMIT;
        if ($this->pagesize > PAGESIZE_UPPER_LIMIT)
            $this->pagesize = PAGESIZE_UPPER_LIMIT;
    }
    function setPage($page) {
        $this->page = $page;
        if ($this->page < 1) $this->page = PAGE_DEFAULT;
        if ($this->page > $this->last_page)
            $this->page = $this->last_page;
    }
    function setPageSize($pagesize) {
        $this->pagesize = $pagesize;
        if ($this->pagesize < PAGESIZE_LOWER_LIMIT)
            $this->pagesize = PAGESIZE_LOWER_LIMIT;
        if ($this->pagesize > PAGESIZE_UPPER_LIMIT)
            $this->pagesize = PAGESIZE_UPPER_LIMIT;
        $this->last_page =
            ceil($this->total_rows/$this->pagesize);
    }
    function getOffset() {
        return (($this->page-1) * $this->pagesize);
    }function getLimit() {
        return (" LIMIT ".$this->getOffset().",".
            $this->pagesize);
    }
}
?>

```

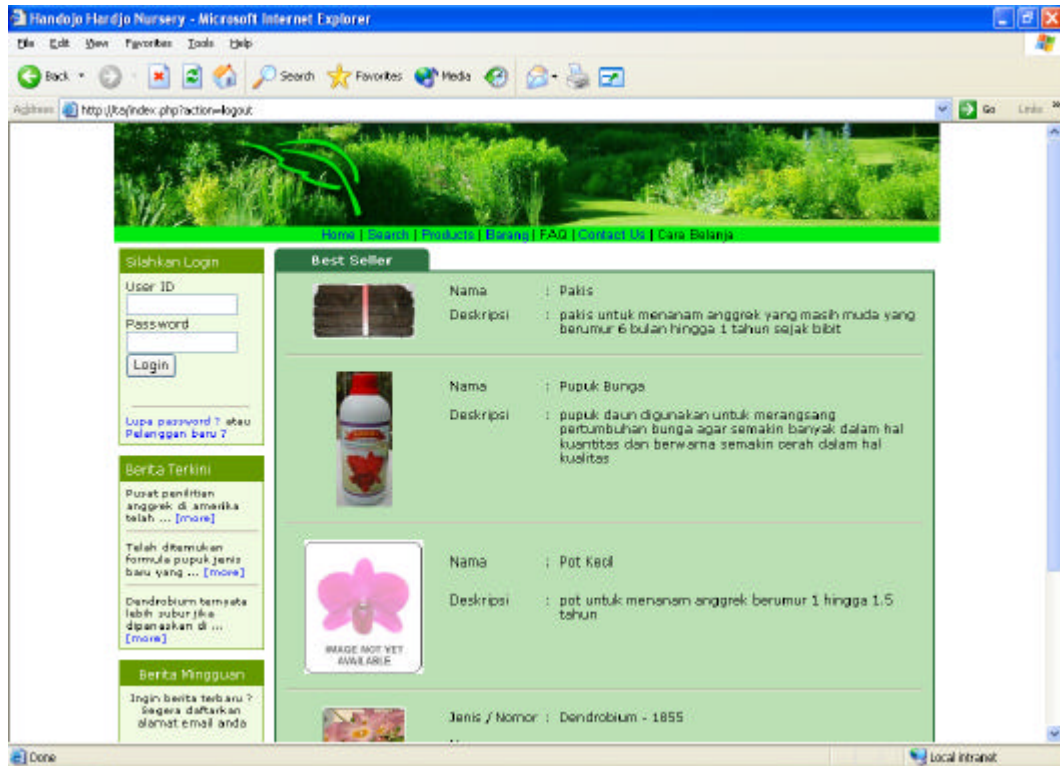
Segmen Program 4.4. *Paging* (Lanjutan)

```

<?
function getNavigation() {
    $this->qs = preg_replace("#\&page=\d+#", "", $this->qs);
    $this->qs =
        preg_replace("#\&pagesize=\d+#", "", $this->qs);
    $this->qs = preg_replace("#page=\d+\&?#", "", $this->qs);
    $this->qs =
        preg_replace("#pagesize=\d+\&?#", "", $this->qs);
    if (preg_match("#\D+=\w+#", $this->qs)) $this->qs .= "&";
    $this->qs .= "pagesize=".$this->pagesize."&";
    $pages_links = ($this->page > 6) ? "... " : "";
    $next_page_completer = ($this->page-5 < 1) ?
        abs($this->page-5)+1 : 0;
    $prev_page_completer =
        ($this->page+5 > $this->last_page) ?
        ($this->page+5) - $this->last_page : 0;
    for ($i=$this->page-5-$prev_page_completer;
        $i < $this->page; $i++) {
        if ($i < 1) continue;
        $pages_links .= " <a href=".PHP_SELF."?".
            $this->qs."page=$i>$i</a>";
    }
    $pages_links .= ($this->last_page != 1) ?
        " <b>$this->page</b>" : "";
    for ($i=$this->page+1;
        $i < ($this->page+6+$next_page_completer); $i++) {
        if ($i > $this->last_page) break;
        $pages_links .= " <a href=".PHP_SELF."?".
            $this->qs."page=$i>$i</a>";
    }
    $pages_links .= ($this->page+5 < $this->last_page) ?
        " ... " : "";
    return (((($this->last_page > 1) ?
        "<table align='center' border='0'><tr".
        ((!empty($this->style)) ?
        " class='$this->style' : "").">".
        "<td>".($this->page == 1 ? "First" :
        "<a href=".PHP_SELF."?".
        $this->qs."page=1>First</a>")."</td>".
        "<td>".($this->page == 1 ? "Prev" :
        "<a href=".PHP_SELF."?".$this->qs."page=" .
        ($this->page-1).">Prev</a>")."</td>".
        "<td align='middle'>$pages_links</td>".
        "<td align='right'>".
        ($this->page == $this->last_page ? "Next" :
        "<a href=".PHP_SELF."?".$this->qs."page=" .
        ($this->page+1).">Next</a>")."</td>".
        "<td align='right'>".($this->page ==
        $this->last_page ? "Last" :
        "<a href=".PHP_SELF."?".$this->qs."page=" .
        ($this->last_page).">Last</a>")."</td>".
        "</tr></table>" : ""))
    );
}
?>

```

Pada halaman pertama kali pelanggan mengunjungi *website* terdapat menu di kiri atas untuk *login*, seperti terlihat pada Gambar 4.6. Menu ini bertujuan agar membatasi hanya pelanggan yang merupakan *member* yang dapat menggunakan fasilitas – fasilitas tertentu dari *website* ini seperti keranjang belanja dan lain lain.

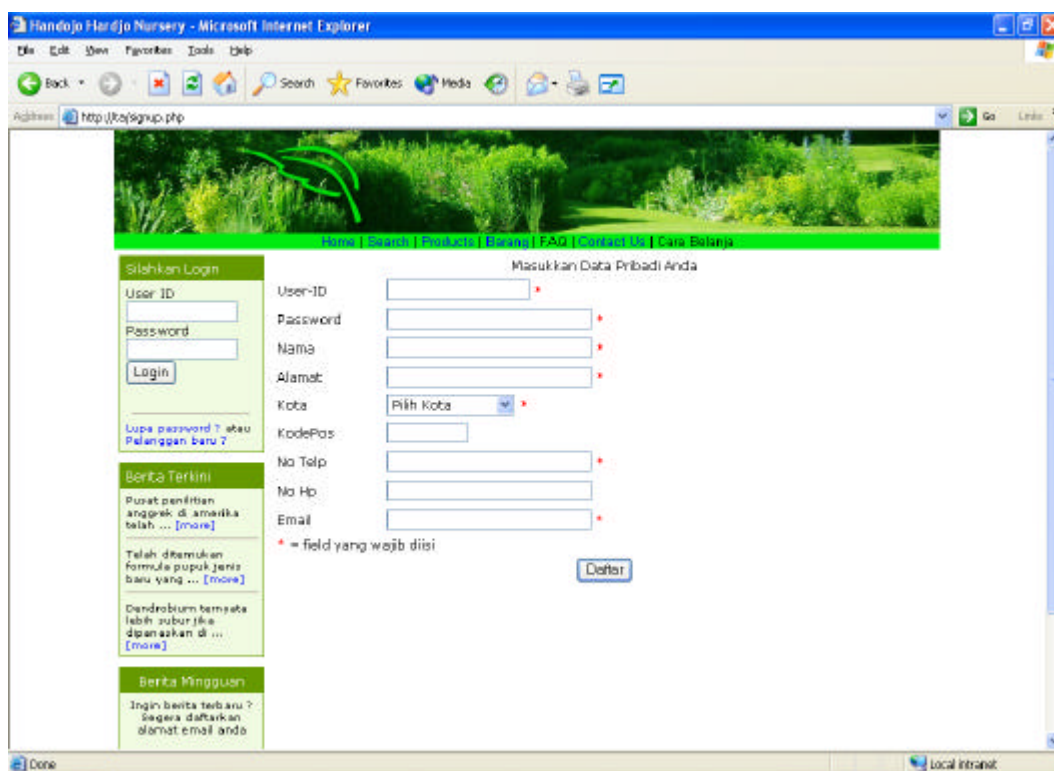


Gambar 4.6. Halaman *Interface Login*

Segmen Program 4.5. *Login*

```
<?
if (isset($p_user_id) && isset($p_password)) {
    if ($p_user_id == "admin" && $p_password == "112358") {
        $_SESSION["user_id"] = "admin";
        header("Location: admin.php");
    } else {
        $qry = "select cust_pswd from customer where cust_id =
        '$p_user_id'";
        $res = mysql_query($qry) or die($qry);
        if (mysql_num_rows($res)) {
            list($pswd) = mysql_fetch_row($res);
            if ($pswd == md5($p_password))
                $_SESSION["user_id"] = $p_user_id;
            else $err_login = WRONG_PASSWORD;
        } else $err_login = NO_SUCH_ID;
    }
}
?>
```

Bila pada menu *login* ada pelanggan yang hendak menggunakan fasilitas *website* tetapi belum terdaftar, maka dia dapat mendaftar dengan cara klik di 'Pelanggan baru ?' di bawah *form login* dan akan berpindah ke *form* pendaftaran / registrasi *user* baru, dengan ketentuan *user* tinggal di Indonesia dan memiliki *E-mail*. Contoh *form* pendaftaran seperti terlihat pada Gambar 4.7. dan Segmen Program 4.6. berikut ini.



The image shows a screenshot of a web browser window titled "Handoyo Hardjo Nursery - Microsoft Internet Explorer". The address bar shows "http://ke/signup.php". The page features a green header with a navigation menu: "Home | About | Products | Billing | FAQ | Contact Us | Cara Belanja". Below the header, there is a "Silahkan Login" section with fields for "User ID" and "Password", and a "Login" button. A link "Lupa password ? atau Pelanggan baru ?" is also present. To the right, a "Masukkan Data Pribadi Anda" section contains a registration form with fields for "User-ID", "Password", "Nama", "Alamat", "Kota" (with a "Pilih Kota" dropdown), "KodePos", "No Telp", "No Hp", and "Email". A "Daftar" button is at the bottom right of this section. A legend indicates "* = field yang wajib diisi". On the left side, there are three "Berita Terkini" (Latest News) sections with "more" links. The browser status bar at the bottom shows "Done" and "Local intranet".

Gambar 4.7. Halaman Registrasi

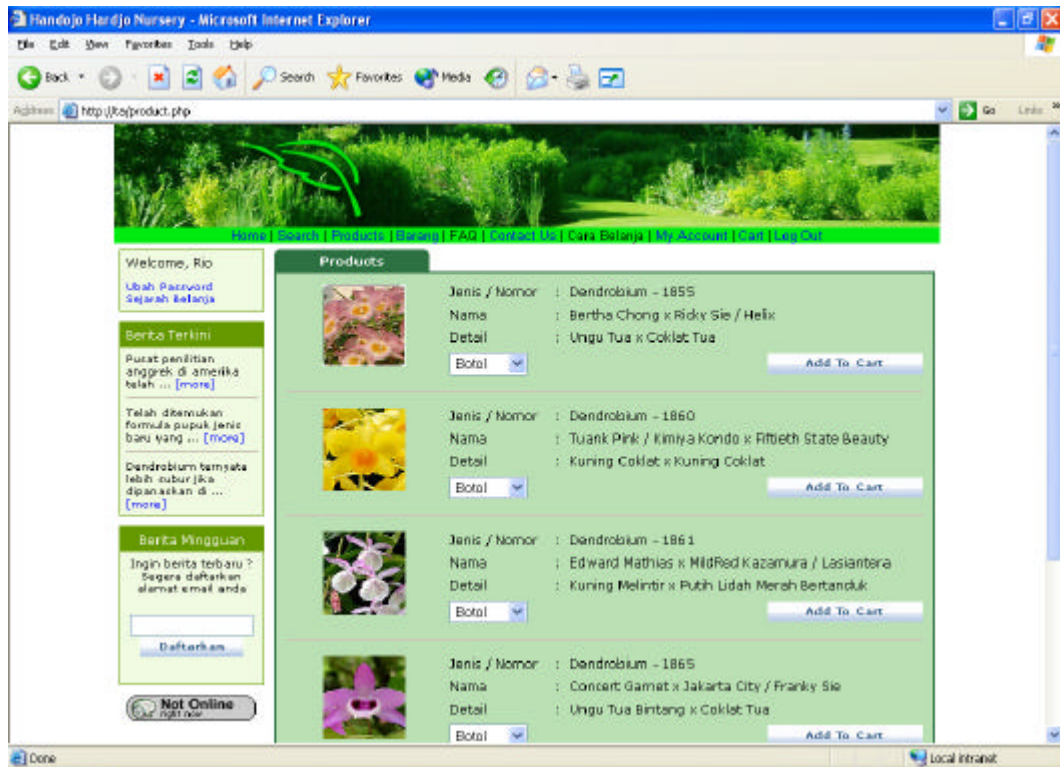
Segmen Program 4.6. Registrasi

```

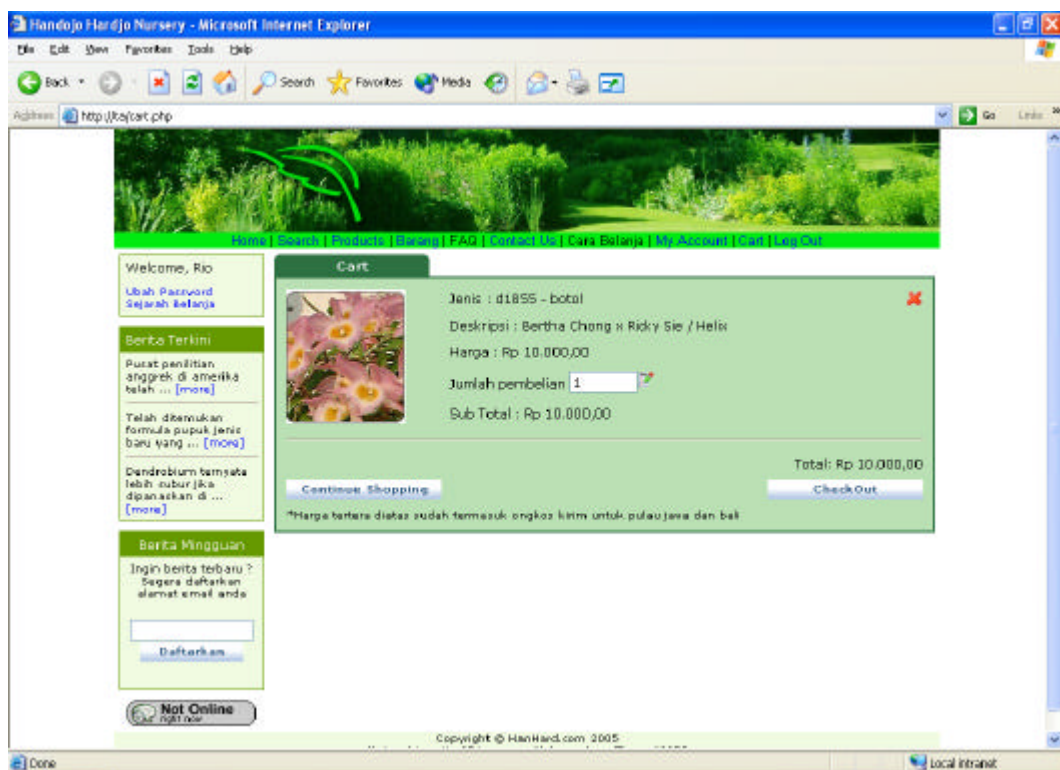
<?
  if (isset($p_Daftar)) {
    $qry = "SELECT COUNT(*),cust_id FROM customer WHERE
    (cust_id = '$p_cust_id') OR (email='$p_cust_email')
    group by cust_id";
    $res = mysql_query($qry);
    list($count,$cust_id) = mysql_fetch_row($res);
    if ($count == 0) {
      $qry = "INSERT INTO customer VALUES ('$p_cust_id',
      '".md5($p_cust_password)."', '$p_cust_nama',".
      "'$p_cust_alamat','$p_cust_kota','$p_cust_kodepos',
      '$p_cust_telp','$p_cust_hp','$p_cust_email',1)";
      $res = mysql_query($qry) or
        die($qry."<hr>".mysql_error());
      $msg = "Selamat $p_cust_id, Anda Telah Terdaftar
      Dengan E-mail $p_email";
    } else {
      if ($cust_id == $p_cust_id) {
        $msg = "ID Telah Dipakai";
        $cust["cust_id"] = "";
      } else {
        $msg = "E-mail Telah Terpakai";
        $cust["email"] = "";
      }
    }
  }
?>

```

Setelah pelanggan *login* pada halaman produk akan muncul pilihan untuk menambahkan barang belanjaan ke dalam keranjang belanja bila pelanggan merasa tertarik dengan barang yang ditampilkan, untuk kemudian melakukan transaksi pembelian. Dua fungsi keranjang belanja yang utama adalah mengubah jumlah barang dan menghapus barang dari keranjang belanja seperti terlihat dengan jelas dari Gambar 4.8. dan Gambar 4.9. Keranjang belanja ini menggunakan *session* sebagai media penyimpanannya agar bila pelanggan telah selesai keranjang belanja akan langsung dikosongkan dengan sendirinya.



Gambar 4.8. Gambar Halaman Produk



Gambar 4.9. Gambar Halaman Keranjang Belanja

Segmen Program 4.7. Keranjang Belanja

```

<?
    $cart = array();
    if (isset($_SESSION["ses_cart"]) &&
        !empty($_SESSION["ses_cart"])) {
        $cart = parse_qs($_SESSION["ses_cart"],false);
        foreach ($cart as $key=>$value) {
            $cart_part = explode("|",$value);
            $sum[] = $cart_part[0] * $cart_part[5];
        }
        $total = array_sum($sum);
        if (isset($v_action)) {
            switch ($v_action) {
                case "edit" :
                    $temp = explode("|",$cart[$v_ed_id]);
                    $temp[5] = $v_qty;
                    $cart[$v_ed_id] = implode("|",$temp);
                    $_SESSION["ses_cart"] = make_qs($cart,false);
                    break;
                case "delete" :
                    unset($cart[$v_del_id]);
                    $_SESSION["ses_cart"] = make_qs($cart,false);
                    break;
            }
            header("Location: cart.php");
        }
    }
    if (isset($v_id)) {
        if (substr($v_id,0,1) == "h") {
            $qry = "select harga_jual,deskripsi,path,nama,
                barang_id from barang where barang_id =
                ".substr($v_id,1);
            $res = mysql_query($qry)
                or die("$qry<hr>".mysql_error());
            if (mysql_num_rows($res))
                $row = mysql_fetch_row($res);
        } else {
            $qry =
                "select ad.harga_jual,concat(a.nama_induk_1,' x
                ',a.nama_induk_2),a.path,a.anggrek_id,
                ad.anggrek_id"." from anggrek as a,anggrek_detail
                as ad"." where (a.anggrek_id='".substr($v_id,0,
                -1)."' and (ad.media like '".substr($v_id,
                -1)."%')"." and (a.anggrek_id=ad.anggrek_id)";
            $res = mysql_query($qry)
                or die("$qry<hr>".mysql_error());
            if (mysql_num_rows($res))
                $row = mysql_fetch_row($res);
        }
        $row[] = 1;
        if (!isset($cart[$v_id]))
            $cart[$v_id] = implode("|",$row);
        $_SESSION["ses_cart"] = make_qs($cart,false);
        header("Location: cart.php");
    }
?>

```

Bila pelanggan telah puas dengan pilihan barang belanjanya, pelanggan dapat melakukan transaksi pembelian dengan cara meng-klik pada tombol 'CheckOut', Seperti pada Gambar 4.9. *CheckOut* disini akan memasukkan data belanjaan yang diambil dari *session* ke tabel penjualan dan detail_penjualan, kemudian menghapus isi dari *session* itu.

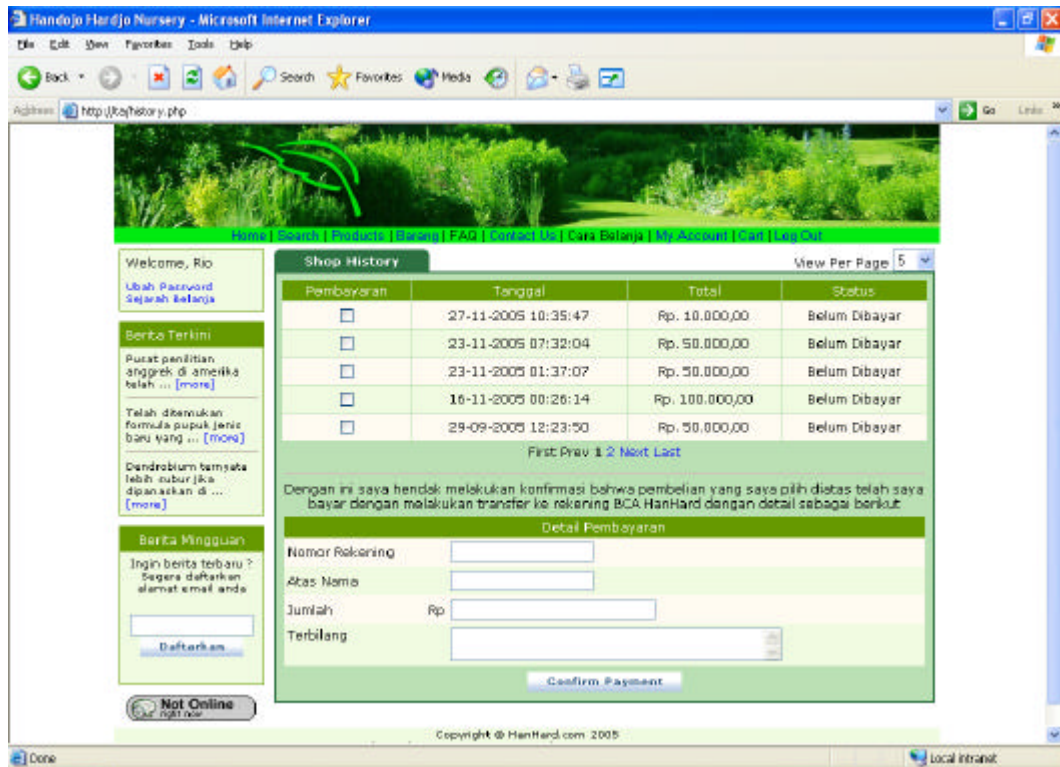
Segmen Program 4.8. *CheckOut*

```

<?
  if ($v_action == "checkout") {
    $cart = parse_qs($_SESSION["ses_cart"],false);
    $sum = array();
    foreach ($cart as $key=>$value) {
      $cart_part = explode("|",$value);
      $sum[] = $cart_part[0] * $cart_part[5];
    }
    $total = array_sum($sum);
    $qry = "insert into penjualan(cust_id,tgl,total)
    values('".$_SESSION["user_id"]."','".$.date("Y-m-d
    H:i:s")."', $total)";
    $res = mysql_query($qry) or die($qry);
    $qry = "select last_insert_id()";
    $res = mysql_query($qry) or die($qry);
    list($last_id) = mysql_fetch_row($res);
    foreach($cart as $key=>$value) {
      $cart_part = explode("|",$value);
      $qry = "insert into detail_penjualan
      (penj_id,brg_id,harga_jual,qty) values ($last_id,
      '$cart_part[4]', $cart_part[0], $cart_part[5])";
      $res = mysql_query($qry) or die($qry);
    }
  }
?>

```

Pelanggan yang telah melakukan pembelian dapat melihat sejarah belanja apa saja yang pernah dilakukannya dengan cara meng-klik pada *link* 'Sejarah Belanja', Seperti pada Gambar 4.9. kemudian pelanggan akan dipindahkan ke halaman sejarah belanja dimana pada halaman ini pelanggan dapat langsung melakukan konfirmasi pembayaran bila pelanggan telah melakukan pembayaran atas pembeliannya, Seperti terlihat pada gambar 4.10.



Gambar 4.10. Halaman Sejarah Belanja

Pada halaman sejarah belanja ini untuk memudahkan pelanggan dalam melakukan konfirmasi pembayaran digunakan fungsi *javascript* untuk membantu mengurangi kesalahan *input*, Seperti terlihat pada Segmen Program 4.9.

Segmen Program 4.9. Sejarah Belanja

```

<script language="javascript">
var total;
function set_harga(obj) {
  if (!obj.checked) total -= obj.value*1;
  if (obj.checked) total += obj.value*1;
  document.formhistory.jumlah.value = total;
  terbilang_rp(document.formhistory.jumlah);
}
function cek_harga(obj) {
  if (!cekRekeningBCA(obj.norek.value)) {
    alert('Nomor rekening tidak valid');return false;}
  if (!cekName(obj.an.value)) {
    alert('Nama tidak valid');return false;}
  if (total != 0)
  if (total == obj.jumlah.value) {
    obj.action.value = 'pembayaran';obj.submit();
  } else alert('Jumlah Setoran Tidak Sesuai');
}

```

Segmen Program 4.9. Sejarah Belanja (Lanjutan)

```

var old = '';
function terbilang_rp(obj) {
  if (/^\d/.test(obj.value)) {
    obj.value = old;
    return false;
  } else {
    old = obj.value;
  }
  digit = new Array('nol','satu','dua','tiga','empat','lima',
  'enam','tujuh','delapan','sembilan');
  order = new Array('','puluh','ratus','ribu',' ',' ','juta',
  ' ',' ','miliar',' ',' ','triliun',' ',' ','kuadriliun');
  var count; var i; var flag; var cont; var idx; var jum;
  var str_res; var s; var p; var r; var temp_str;
  while(/^\d/.test(obj.value)) {
    obj.value = obj.value.substr(1);
  }
  count = obj.value.length; flag = (count > 0) ? true : false;
  i=1;str_res = '';
  while (flag) {
    if (i*3 >= count) {
      idx = 0;
      jum = count % 3;
      flag = false;
      jum = (jum == 0) ? 3 : jum;
    } else {
      idx = count - i*3;
      jum = 3;
    }
    temp_str = obj.value.substr(idx,jum);
    while (/^\d/.test(temp_str)) {
      temp_str = temp_str.substr(1);
    }
    if (temp_str == '') temp_str=0;
    cont = parseInt(temp_str);
    s = cont % 10;
    p = ((cont-s)/10) % 10;
    r = ((cont-p*10-s)/100) % 10;
    if (s != 0 || p != 0 || r != 0) str_res = order[(i-
1)*3]+' '+str_res;
    if (p == 1 && s == 0) str_res = 'sepuluh'+ ' '+str_res
    else if (p == 1 && s == 1) str_res = 'sebelas'+
'+str_res
    else if (p == 1) str_res = digit[s]+' belas'+ ' '+str_res
    else str_res = ((p == 0) ? '' : digit[p]+' '+order[1]+'
')+
    ((i==2 && s == 1 && p == 0 && r == 0) ? 'se' : ((s!=0) ?
digit[s]+' ' : ''))+str_res;
    if (r == 1) str_res = 'se'+order[2]+' '+str_res
    else str_res = (r != 0) ? digit[r]+' '+order[2]+'
'+str_res : str_res;
    i++;
  }
  obj.form.terbilang.value = str_res;
}
</script>

```

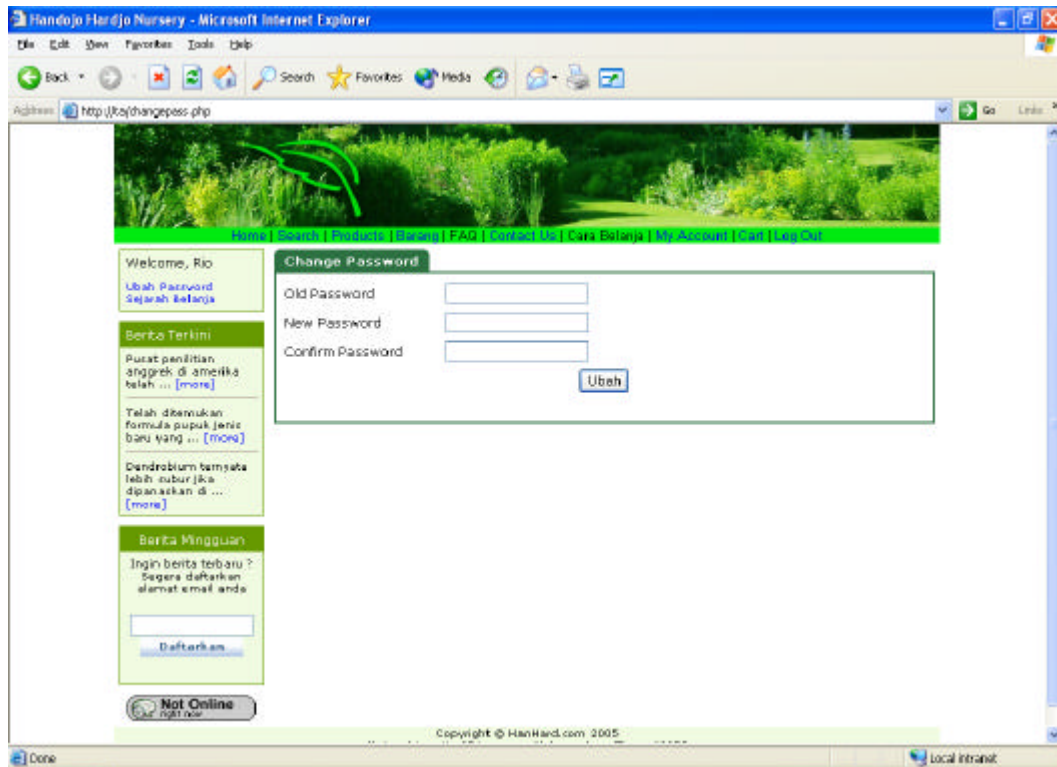
Segmen Program 4.10. Konfirmasi Pembayaran

```

<?
  if ($p_action == "pembayaran") {
    foreach($p_cb as $key=>$value) {
      $qry = "UPDATE penjualan SET pembayaran=' $p_norek\r\n
        $p_an\r\n$p_jumlah\r\n$p_terbilang\r\n".date("d-m-Y
        H:i:s")."' WHERE penj_id = $key";
      $res = mysql_query($qry);
    }
    $msg = "Terima kasih karena telah melakukan pembayaran
    sejumlah $p_terbilang dari nomor rekening $p_norek atas
    nama $p_an";
  } else {
    $qry = "SELECT penj_id,pembayaran,tgl,total FROM
    penjualan WHERE (cust_id = ' ".$_SESSION["user_id"]."')
    ORDER BY tgl";
    $res = mysql_query($qry);
    if (mysql_num_rows($res)) {
      $cont = array();
      while ($row = mysql_fetch_row($res)) {
        $cont[] = $row;
      }
    }
  }
?>

```

Untuk proses perubahan *password* pelanggan diminta untuk memasukkan *password* lama yang akan diganti terlebih dahulu, baru setelah itu memasukkan *password* yang baru. Setelah itu sistem akan mengecek apakah *password* yang akan diganti tersebut cocok dengan konfirmasinya dan apakah *password* baru itu tidak sama dengan *password* lama. Jika benar maka dilakukan pengecekan sekali lagi apakah panjang *password* sudah cukup untuk dikatakan ‘aman’. Jika benar semua barulah sistem akan melakukan pengecekan dengan *database* apakah *password* lama cocok baru setelah itu melakukan penggantian *password*. Pengecekan yang tidak berhubungan dengan *database* dilakukan dengan bantuan *javascript*, supaya tidak ada *loading* halaman yang tidak diperlukan untuk menghemat *bandwidth*.



Gambar 4.11. Halaman Ubah *Password*

Segmen Program 4.11.1. Ubah *Password* (*JavaScript*)

```

<script language="javascript">
function validasi(obj) {
    var res; res = true; str = '';
    if (!cekPassword(obj.old.value)) {res = false;str+='old
password should be any keyboard character and 6 character
minimum\n';obj.old.value = '';}
    if (!cekPassword(obj.new_p.value)) {res = false;str+='new
password should be any keyboard character and 6 character
minimum\n';obj.new_p.value = '';}
    if (!cekPassword(obj.confirm_p.value)) {res = false;
str+='confirm password should be any keyboard character and
6 character minimum\n';obj.confirm_p.value = ''}
    if (obj.new_p.value != obj.confirm_p.value) {
        res = false;str+='confirm your new password\n';
        obj.new_p.value = '';obj.confirm_p.value = '';
    }
    if (obj.old.value == obj.new_p.value) {
        res = false;
        str+='password baru sama dengan password lama\n';
    }
    if (!res) {alert(str);} return res;
}
</script>

```

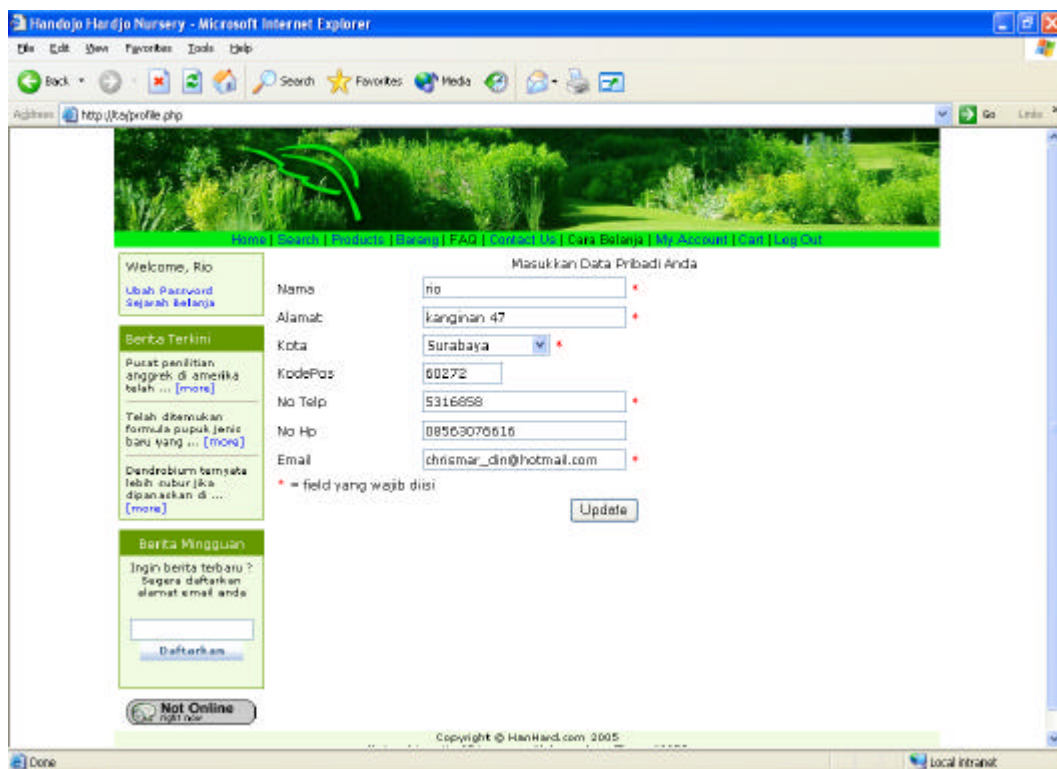
Segmen Program 4.11.2. Ubah *Password* (PHP)

```

<?
if (isset($p_Ubah)) {
    $qry = "SELECT cust_pswd FROM customer WHERE
cust_id='". $_SESSION["user_id"]. "'";
    $res = mysql_query($qry);
    if (mysql_num_rows($res)) {
        list($pswd) = mysql_fetch_row($res);
        if ($pswd == md5($p_old)) {
            $qry = "UPDATE customer SET
cust_pswd='". md5($p_new_p). "' WHERE
cust_id='". $_SESSION["user_id"]. "'";
            $res = mysql_query($qry);
            $msg = "Anda Telah Berhasil Mengganti
Password Anda";
        }
    }
}
?>

```

Pelanggan juga dapat mengubah data pribadi bila pelanggan telah berpindah alamat supaya barang yang dibeli tidak salah kirim.



Gambar 4.12. Halaman Ubah Data Pribadi

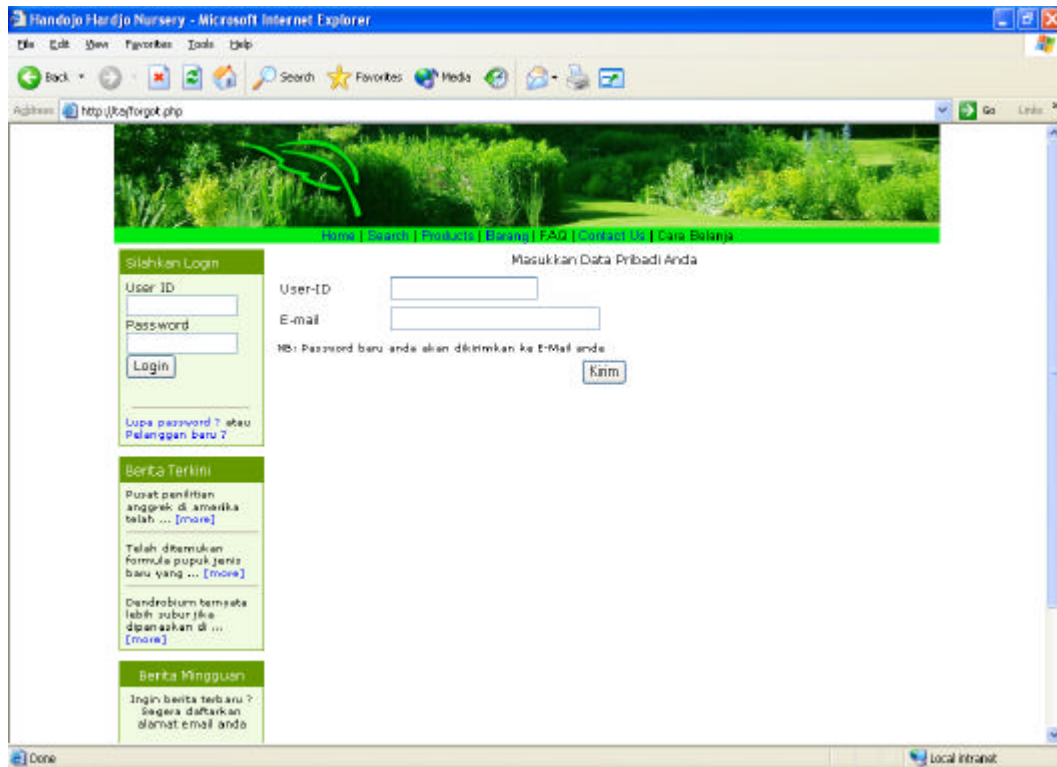
Segmen Program 4.12. Ubah Data Pribadi

```

<?
  if (isset($p_Update)) {
    $qry = "UPDATE customer SET nama='$p_cust_nama',
alamat='$p_cust_alamat' ".",kota='$p_cust_kota',kodepos=
'$p_cust_kodepos',telp='$p_cust_telp' ".",hp='
$p_cust_hp',email='$p_cust_email' WHERE
cust_id='$p_cust_id'";
    $res = mysql_query($qry)
      or die($qry."<hr>".mysql_error());
  } else {
    $qry = "select * from customer where cust_id =
'".$_SESSION["user_id"]."'";
    $res = mysql_query($qry)
      or die($qry."<hr>".mysql_error());
    $cust = mysql_fetch_assoc($res);
  }
?>

```

Pelanggan yang lupa akan *password*-nya dapat meminta *password* yang baru untuk dikirimkan ke alamat *E-mail*-nya, dengan cara klik pada *link* 'Lupa password ?' kemudian mengisi data pribadi seperti *user-id* dan alamat *E-mail* untuk konfirmasi bahwa *user-id* yang meminta *password* baru mempunyai alamat *E-mail* yang benar.



Gambar 4.13. Halaman Lupa *Password*

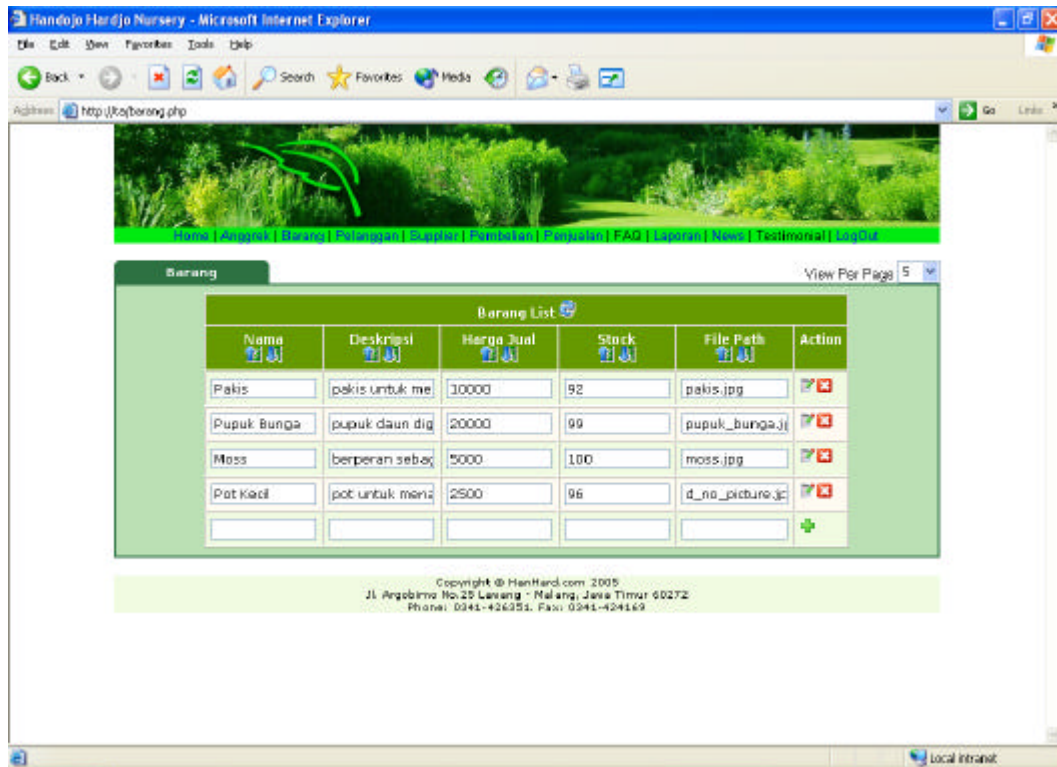
Segmen Program 4.13. Lupa *Password*

```

<?
    if (isset($p_Reset)) {
        $qry = "select * from customer where
        (cust_id='$p_forgot_id') and (email='$p_email)";
        $res = mysql_query($qry);
        if (mysql_num_rows($res)) {
            $new = new_pswd(8);
            $qry = "update customer set cust_pswd='".md5($new)." '
            where cust_id='$p_forgot_id'";
            $res = mysql_query($qry) or die(mysql_error());
            mail($p_email,"Password Baru HanHard","User-ID :
            $p_forgot_id\nPassword baru : $new",
            "From: webmaster@".$_SERVER['SERVER_NAME']);
        }
    }
??>

```

Administrasi Barang adalah sebuah halaman untuk melakukan administrasi dari tabel barang seperti memasukkan data baru ke tabel barang, mengubah 1 baris data lalu melakukan *update*, ataupun melakukan penghapusan 1 baris data.



Gambar 4.14. Halaman Administrasi Barang

Segmen Program 4.14. Administrasi Barang

```

<?
if ($p_action == "delete") {
    $qry = "DELETE FROM barang WHERE barang_id='$p_id'";
    $res = mysql_query($qry)
        or die("$qry<hr>".mysql_error());
} else if (isset($p_b) && is_array($p_b)) {
    foreach($p_b as $key1=>$value1) {
        $fields = ($key1 != 0) ? ",barang_id" : "";
        $values = ($key1 != 0) ? ",$key1" : "";
        foreach($value1 as $key2=>$value2) {
            $fields .= ",$key2"; $values .= ",'$value2'";
        }
        $fields = substr($fields,1);
        $values = substr($values,1);
        $qry = "REPLACE INTO barang($fields) VALUES
            ($values)";
        $res = mysql_query($qry)
            or die("$qry<hr>".mysql_error());
    }
}
?>

```

Setelah pelanggan atau *admin* selesai menggunakan *website* pelanggan atau *admin* dapat melakukan *logout* supaya tidak terjadi penyalah - gunaan *account* dan terjadi hal – hal yang tidak diinginkan, dengan cara klik pada bagian menu *logout* seperti terlihat pada Gambar 4.14. Kemudian proses *logout* akan segera berjalan.

Segmen Program 4.15. *LogOut*

```
<?
  if (isset($_SESSION["user_id"]) && $_v_action=="logout") {
    unset($_SESSION["user_id"]);
  }
?>
```