

# Sistem Informasi Administrasi Pengujian Benih Tanaman Pangan dan Hortikultura (Studi Kasus Laboratorium BPSB Propinsi Jawa Timur)

Tawar\*, Arindra Saktiawan

Program Studi Sistem Informasi, Universitas Ahmad Dahlan,  
Jalan Prof. Dr. Soepomo, S.H., Janturan, Yogyakarta

E-mail: tawar@uad.ac.id

**ABSTRAK** *Pengelolaan data pengujian benih di Badan Pengawasan Sertifikasi Benih (BPSB) Propinsi Jawa Timur telah menggunakan sistem komputer. Layanan pengujian benih meliputi pengujian standard, pengujian khusus berat 1000 butir, pengujian khusus heterogenitas, dan pengujian khusus viabilita. Sistem komputer yang ada belum mencakup semua layanan pengujian benih. Selain itu, sistem yang ada belum menerapkan sistem terintegrasi antar cabang. Petugas di kantor pusat masih harus melakukan entry ulang terhadap data-data yang dikirim dari kantor cabang.*

*Penelitian ini dilakukan dalam beberapa tahapan yaitu mengidentifikasi permasalahan, pengumpulan data, analisis data, perancangan sistem, pengkodean, dan yang terakhir adalah pengujian sistem menggunakan metode pengujian Alpha test dan Beta test. Perancangan sistem dalam penelitian ini menggunakan framework PHP Codeigniter dengan beberapa tools pendukung seperti jQuery untuk mempercantik tampilan, penanganan event, dan Ajax.*

*Hasil penelitian ini adalah aplikasi Sistem Informasi Pengujian Benih Tanaman Pangan dan Hortikultura yang dapat membantu mengatasi masalah dalam mengelola data pengujian benih dan laporan bulanan UPT PSBTPH Propinsi Jawa Timur.*

**Keywords:** *Sistem Informasi, Pengujian benih, Framework, Codeigniter, jQuery, ajax.*

## 1 Pendahuluan

Dewasa ini perkembangan Teknologi Informasi mulai mendapat sambutan positif dari masyarakat. Salah satu contoh adalah Unit Pelaksana Teknis Pengawasan Sertifikasi Benih Tanaman Pangan dan Hortikultura (UPT PSBTPH) propinsi Jawa Timur yang lebih dikenal dengan sebutan BPSB (Badan Pengawasan Sertifikasi Benih). Instansi pemerintah yang bergerak di bidang sertifikasi benih ini mulai menyadari betapa pentingnya penerapan teknologi informasi dalam dunia kerja.

BPSB mempunyai tiga bagian struktur organisasi yang masing-masing mempunyai ketetapan tugas sendiri-sendiri, yaitu bagian Fungsional, Pemasaran, dan Laboratorium. Bagian Fungsional secara garis besar bertugas mengambil benih dari para konsumen yang selanjutnya benih tersebut dikirim ke Laboratorium untuk diuji sebelum kemudian dinyatakan lulus atau tidak. Bagian pemasaran mempunyai tugas untuk mengawasi peredaran benih yang telah beredar di pasaran. Dari tiga fungsi tersebut, bagian laboratoriumlah yang

memegang kunci terhadap hasil pengujian. Bagian laboratorium mempunyai tugas untuk melakukan ujicoba terhadap benih sampai dengan proses dikeluarkannya hasil pengujian dalam bentuk laporan terstruktur.

Mulai tahun 2009 untuk melakukan proses pengeluaran informasi hasil pengujian, Laboratorium BPSB propinsi Jawa Timur menggunakan sebuah Sistem Informasi yang dibangun menggunakan MS Access. Sistem Informasi yang digunakanpun tergolong sangat sederhana karena hanya dapat melakukan proses terhadap pengujian standard, sedangkan proses laboratorium yang lain belum menggunakan sistem komputer. Layanan pengujian yang belum menggunakan sistem komputer meliputi Pengujian Khusus Berat 1000 Butir, Uji Cepat Viabilitas, Uji Khusus Heterogenitas, dan Uji Khusus Kesehatan Benih.

BPSB Propinsi Jawa Timur mempunyai enam cabang termasuk yang berpusat di Surabaya. Selama ini proses integrasi data tidak dapat dilakukan karena keterbatasan sistem dan staf yang cukup mengerti terhadap cara kerja sistem yang telah ada. Hal ini berakibat pada proses pelaporan hasil pengujian yang telah diinput dari cabang ke pusat terkendala. Para staf yang ada di kantor pusat terpaksa harus melakukan penginputan ulang terhadap data hasil uji yang telah dikeluarkan oleh cabang. Hal tersebut tentu saja merupakan sesuatu yang sangat membebani bagi instansi pusat, belum lagi tuntutan dari Departemen Pertanian Propinsi Jawa Timur yang menuntut agar laporan bulanan berupa *hardcopy* dan *softcopy* berupa file \*.xls.

Hal seperti di atas tentu saja tidak akan terjadi, jika sistem yang digunakan berbasis Web, sehingga instansi cabang melakukan input data secara langsung ke pusat data. Pembuatan aplikasi berbasis web akan terasa lebih mudah jika menggunakan framework, karena di dalam framework telah terdapat kumpulan fungsi-fungsi standard yang dibutuhkan dalam pembuatan sebuah aplikasi. Salah satu framework berbasis bahasa pemrograman PHP Hyper Text Processor adalah CodeIgniter.

## 2 Landasan Teori

### 2.1 *Framework* PHP

Framework adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain, sehingga dalam pembuatan aplikasi programmer harus mengikuti aturan dari framework tersebut (Wardana, 2010). Framework memungkinkan programmer untuk mempersingkat waktu kerjanya tanpa harus membuat fungsi-fungsi dasar. Wardana (2010) mendefinisikan keuntungan yang dapat diperoleh dari penggunaan framework diantaranya:

- (a) Waktu pembuatan aplikasi jauh lebih singkat
- (b) Kode aplikasi lebih mudah dipahami karena sedikit dan bersifat pokok
- (c) Aplikasi lebih mudah diperbaiki
- (d) Fungsi-fungsi dasar yang telah tersedia dalam framework.

Salah satu framework PHP yang populer adalah CodeIgniter. CodeIgniter dibuat oleh Rick Ellis berbasis PHP 4, bersifat open source dan gratis. CodeIgniter adalah framework PHP yang dibuat berdasarkan kaidah Model View Controller (MVC).

Model merepresentasikan struktur data dari website, dapat berupa basis data maupun data lain, misalnya dalam bentuk file teks dan xml. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil dan melakukan update dan menghapus data website. Karena sebuah website biasanya menggunakan basis data dalam menyimpan data, maka bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL.

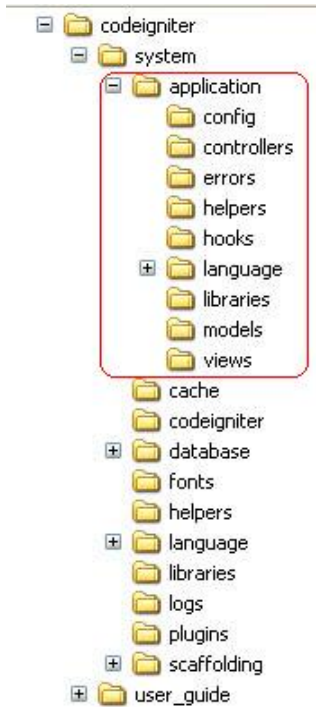
View merupakan informasi yang ditampilkan kepada pengunjung website. Sedapat mungkin view tidak berisi logika-logika kode, tetapi hanya berisi variable-variabel yang berisi data yang siap ditampilkan. View merupakan halaman website yang dibuat menggunakan HTML dengan bantuan Javascript atau CSS. View hanya dikhususkan untuk menampilkan data-data hasil dari model dan controller.

Controller merupakan penghubung antara Model dan View. Di dalam Controller inilah terdapat class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data. Tugas controller adalah menyediakan berbagai variable yang akan ditampilkan di View, memanggil Model untuk melakukan akses ke basis data, menangani error program, mengerjakan proses logika dari aplikasi serta melakukan validasi/cek terhadap input (Oktaviani, 2010).

## 2.2 Framework CodeIgniter

CodeIgniter adalah sebuah framework PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis PHP dibandingkan jika menulis semua code dari awal (Basuki, 2010). CodeIgniter mempunyai struktur direktori seperti pada Gambar 1.

Dari Gambar 1 dapat ditunjukkan bahwa aplikasi yang dikembangkan berpusat pada folder application. Wardana (2010) mendefinisikan fungsi dari masing-masing folder yang berada di dalam folder system adalah sebagai berikut:



**Gambar 1.** Struktur direktori *CodeIgniter*

- (a) Folder *application* merupakan tempat aplikasi yang akan dikembangkan. Folder *application* mempunyai sub folder *config*, *controller*, *errors*, *helpers*, *hooks*, *language*, *libraries*, *models*, dan *views*.
- (b) Folder *cache* tempat menyimpan semua cache yang dibuat *caching library*.
- (c) Folder *CodeIgniter* tempat menyimpan semua file internal CI.
- (d) Folder *database* tempat menyimpan semua driver database dan class yang akan digunakan.
- (e) Folder *font* tempat menyimpan semua fonts yang digunakan *image manipulation library*.
- (f) Folder *helpers* tempat menyimpan semua helper core CI
- (g) Folder *language* tempat menyimpan semua language core CI
- (h) Folder *libraries* tempat menyimpan semua file library atau pustaka core CI
- (i) Folder *logs* tempat menyimpan semua logs generated oleh CI
- (j) Folder *plugin* tempat menyimpan semua plugin core CI
- (k) Folder *scaffolding* tempat menyimpan semua file yang berfungsi sebagai scaffolding.

### 2.3 jQuery

John Resig merilis sebuah Javascript Library dan kini dikenal dengan nama jQuery. Semenjak dirilis pertama kali pada tahun 2006, jQuery telah mencuri perhatian para developer web. jQuery telah banyak digunakan oleh website-website terkemuka di dunia seperti Google, Microsoft, Intel dan masih banyak lagi.

jQuery adalah Javascript Library, kumpulan kode atau fungsi javascript siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode Javascript. jQuery menyederhanakan kode javascript (jQuery, 2011). Newton (2009) mendefinisikan jQuery adalah suatu library javascript yang menyederhanakan penggunaan dokumen HTML. Cara menggunakan jQuery cenderung sederhana seperti slogan yang terpampang di website jQuery “*jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript.*”

## 2.4 Database

Database secara sederhana, dapat disebut sebagai gudang data. secara teori, database adalah sekumpulan data atau informasi yang kompleks, data-data tersebut disusun menjadi beberapa kelompok dengan tipe data yang sejenis, setiap datanya dapat saling berhubungan satu sama lain atau dapat berdiri sendiri, sehingga mudah diakses (Nugroho, 2008). Berkaitan dengan penelitian ini database yang digunakan adalah MySQL. MySQL merupakan database server yang awal mulanya berjalan pada sistem UNIX atau Linux. Seiring berjalannya waktu dan banyaknya peminat yang menggunakan datanase ini, MySQL merilis versi yang hampir dapat diinstall pada hampir semua platform, termasuk Windows.

## 2.5 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sistem notasi yang sudah dibakukan di dunia pengembangan sistem, hasil kerjasama dari Grady Booch, James Rumbaugh dan Ivar Jacobson. UML yang terdiri dari serangkaian diagram memungkinkan bagi sistem analis untuk membuat cetak biru sistem yang komperhensif kepada klien, programmer dn tiap orang yang terlibat dalam proses pengembangan (Munawar, 2005). UML menawarkan sebuah standard untuk merancang model sebuah sistem. Dengan menggunakan UML model dapat dibuat untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

Seperti bahasa bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Abstraksi konsep dasar UML terdiri dari *structural classification, dynamic behavior dan model management*.

# 3 Pembahasan

## 3.1 User Requirement

Kebutuhan UPTPSBTPH terhadap sistem disajikan pada Tabel 1.

Tabel 1. Tabel kebutuhan user (*user requirement*)

<b>Kebutuhan</b>	<b>Aktor</b>	<b>Use Case</b>
Tambah, Ubah, Hapus	Operator Pusat, Operator Daerah	Mengolah Data Benih
Tambah, Ubah, Hapus	Operator Pusat, Operator Daerah	Mengelola data konsumen
Tambah, Ubah, Hapus	Operator Pusat, Operator Daerah	Mengelola data penguji
Tambah, Ubah, Hapus, Cetak	Operator Pusat, Operator Daerah	pengujian standard.
Tambah, Ubah, Hapus, Cetak	Operator Pusat, Operator Daerah	Pengujian Khusus 1000 Butir
Tambah, Ubah, Hapus, Cetak	Operator Pusat, Operator Daerah	Pengujian Khusus Heterogenitas
Tambah, Ubah, Hapus, Cetak	Operator Pusat, Operator Daerah	pengujian khusus Viabilitas
Download Laporan	Operator Pusat, Operator Daerah	Laporan Bulanan Uji Standard gabungan
Download Laporan	Operator Pusat, Operator Daerah	Laporan Bulanan uji standard komoditi Hortikultura
Download Laporan	Operator Pusat, Operator Daerah	Laporan Bulanan Uji Standard komoditi Pangan Daerah
Download Laporan	Operator Pusat, Operator Daerah	Membuat Laporan Bulanan Uji Standard komoditi pangan Negara
Download Laporan	Operator Pusat	Membuat rekapitulasi Laporan Bulanan Uji Standard gabungan seluruh satgas
Download Laporan	Operator Pusat	Membuat rekapitulasi Laporan bulanan uji standard komoditi Hortikultura gabungan seluruh satgas
Download Laporan	Operator Pusat	Membuat rekapitulasi Laporan bulanan uji standard komoditi Hortikultura gabungan seluruh satgas
Download Laporan	Operator Pusat	Membuat rekapitulasi Laporan Bulanan Uji Standard komoditi pangan Negara gabungan seluruh satgas
Tambah, Ubah, Hapus	Administrator	Kelola data petugas
Backup data	Administrator	Backup data

Berdasarkan Tabel 1, diperoleh aktor-aktor yang dimungkinkan menjadi *user* sistem yang akan dikembangkan, yaitu:

(a) Administrator

*User* administrator merupakan *user* yang mempunyai kewenangan untuk melakukan pengelolaan data penunjang sistem, seperti pengelolaan data petugas, pemberian hak akses kepada sistem dan backup data.

(b) Administrasi Satgas Daerah

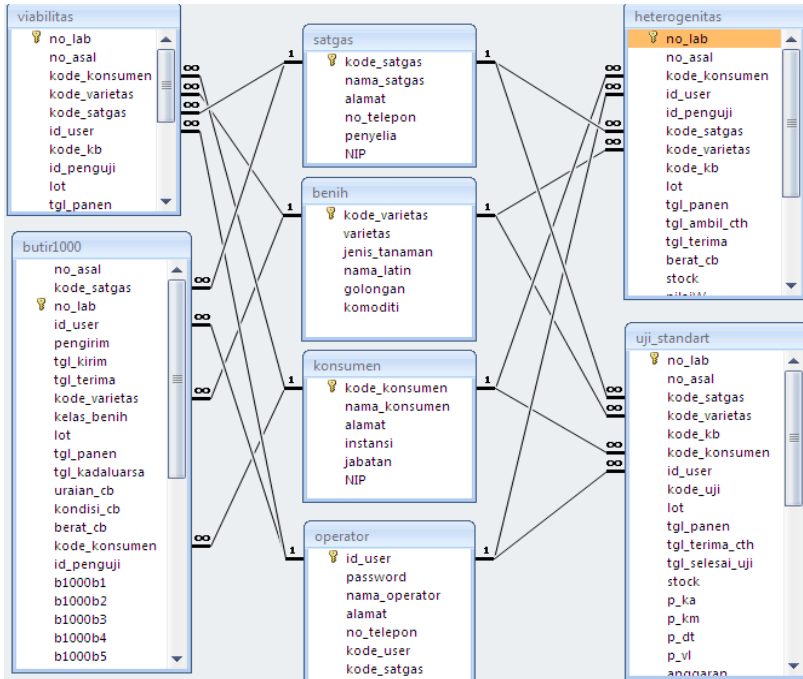
*User* administrasi daerah merupakan *user* yang berada di daerah atau *user* selain petugas administrasi laboratorium I Surabaya. *User* administrasi daerah terdiri dari seluruh petugas administrasi laboratorium daerah meliputi laboratorium II Madiun, laboratorium III Malang, laboratorium IV Kediri, laboratoium V Jember, dan laboratorium VI Banyuwangi.

### (c) Administrasi Satgas Pusat

User administrasi daerah merupakan user yang berada di pusat atau user petugas administrasi laboratorium I surabaya. User administrasi pusat terdiri dari seluruh petugas administrasi laboratorium I Surabaya.

## 3.2 Relasi Antar Tabel

Relasi antar tabel-tabel dapat dilihat pada Gambar 2 berikut:



Gambar 2. Relasi Antar Tabel

## 3.4 Tampilan Antarmuka Pengguna

CodeIgniter meletakkan semua *source code* tampilan dalam folder view. Semua permintaan user akan direkam dan disampaikan kepada daftar controller. *Controller* membawa user menuju ke halaman login. Setelah berhasil login, akan ditampilkan Form utama untuk administrator (Gambar 3).

The screenshot shows the main interface for an administrator. The page title is "Sistem Informasi Pengujian Benih Tanaman Pangan Dan Hortikultura". The user is logged in as "Arindra Saktiawan". The interface includes a sidebar with navigation options: Management Operator, Backup Database, Log Out, and Isi Questioner. The main content area displays a welcome message: "Selamat Datang : Arindra Saktiawan Anda Login Sebagai arindra\_a". Below the message, it states "dibawah ini merupakan Data Diri Anda yang tercatat didalam system". The user's details are listed: Nama : Arindra Saktiawan, Telepon : 0, Alamat : Yogyakarta, and Bekerja Di : Yogyakarta. There are two buttons: "Edit Data" and "Ubah Password".

Gambar 3. Form utama untuk administrator

Tampilan form utama terdiri dari tiga bagian. Ketiga bagian tersebut adalah bagian header, bagian menu dan bagian input-output. Bagian input-output merupakan bagian yang berfungsi untuk menampilkan segala bentuk request yang dilakukan oleh client berdasarkan pada menu yang dipilih. Terdapat empat macam menu pengujian yang dilayani yaitu Pengujian standard, Pengujian berat 1000 butir, Pengujian heterogenitas dan Pengujian viabilitas.

Keempat form layanan pengujian ini pada dasarnya mirip, dirancang terdiri dari dua bagian yaitu **Tahap1**, untuk entry identitas pengujian dan **Tahap2** digunakan untuk entry hasil pengujian (Gambar 4).

Gambar 4. Interface Pengujian Standard

Input pengujian standard dibagi menjadi dua bagian. Bagian pertama merupakan identitas pengujian. Identitas pengujian tersebut meliputi input tentang atribut benih dan attribut pengujian. Atribut benih terdiri input seperti nama varietas, no\_asal, kelas benih, tanggal panen, tanggal terima contoh benih, tgl selesai uji, stock, dan nomer kelompok. Atribut pengujian terdiri dari input seperti no\_lab, permintaan pengujian, jenis uji, dan anggaran.

Tahap kedua pengujian standard adalah input mengenai hasil detail dari pengujian (Gambar 5).

Gambar 5. Tampilan tahap ke 2 pengujian standard

Form output berbentuk tabel, Form output mempunyai tombol untuk menghapus dan mengubah data. Khusus untuk output pengujian terdapat tombol tambahan untuk mencetak form hasil akhir pengujian, dan form induk pengujian. Proses input pengujian akan dilaporkan setiap bulannya. Laporan



bulanan terdiri dari empat laporan untuk uji standard dan satu laporan untuk uji khusus yang meliputi uji berat 1000 butir, uji cepat viabilitas, dan uji heterogenitas.



Gambar 6. Form input tanggal awal dan tanggal akhir laporan persatgas

Untuk mendapatkan laporan bulanan dalam bentuk xls operator harus memasukkan tanggal awal dan tanggal akhir sebagai parameter rentang hari yang akan dilaporkan (Gambar 6).

## 4 Pengkodean

### 4.1 Config CodeIgniter

Config CodeIgniter merupakan folder yang didalamnya terdapat pengaturan-pengaturan yang berkaitan dengan sistem yang akan dibuat. Folder config terdiri dari beberapa file php diantaranya:

#### (a) Autoload.php

File autoload.php merupakan file yang berfungsi untuk mengatur class yang akan di panggil secara otomatis ketika sistem dimuat. File autoload.php mengatur pengaksesan file secara otomatis terhadap class yang berada di dalam folder *Libraries*, *Helper*, *Plugins* dan sebagainya

#### Kode Program 1. autoload.php

```
$autoload['libraries'] = array('database','form_validation','table');  
$autoload['helper'] = array('form', 'url', 'json');  
$autoload['plugin'] = array();  
$autoload['config'] = array();  
$autoload['language'] = array();  
$autoload['model'] = array();
```

Kode program di atas memperlihatkan kode program pada file autoload.php. Baris ke-1 dan ke-2 memperlihatkan bahwa *class library* yang otomatis di load ketika sistem dimuat yaitu *class database*, *class table*, dan *class form\_validation*. Baris ke-3 memperlihatkan class helper yang di load secara otomatis. Class yang di load secara otomatis diantaranya adalah class form, class url, dan class json.

#### (b) Database.php

Database.php merupakan file yang digunakan untuk mengkonfigurasi *database*.

#### Kode Program 2. database.php

```
$db['default']['hostname'] = "localhost";
```

```

$db['default']['username'] = "root";
$db['default']['password'] = "12345678";
$db['default']['database'] = "db_bpsb";
$db['default']['dbdriver'] = "mysql";

```

Dari Kode program 4.2 baris ke-1 merupakan konfigurasi host dari website. Baris ke-2 merupakan konfigurasi *user* yang digunakan untuk koneksi database. Baris ke-3 merupakan konfigurasi password dari *user* yang digunakan. Baris ke-4 merupakan konfigurasi nama database yang digunakan. Baris ke-5 merupakan konfigurasi database driver yang digunakan.

#### (c) Routes.php

File routes.php merupakan file yang digunakan untuk mengatur default *controller* yang akan di load.

#### Kode Program 3. routes.php

---

```

$route['default_controller'] = "welcome";

```

Kode program 3 di atas menunjukkan bahwa *controller* yang pertama kali di load adalah *controller* welcome.

## 4.2 Welcome Controller

Welcome *controller* merupakan *controller* yang pertama kali dipanggil ketika sistem dihidupkan. Welcome *controller* mempunyai tiga daftar action.

#### (a) Action index

Action index digunakan oleh setiap *controller* sebagai action yang akan dijalankan secara default oleh *controller* ketika sebuah *controller* dipanggil.

#### Kode Program 4. controller action index

---

```

function index(){
    if ($this->session->userdata('cpuser_auth') == TRUE){
        $pages['dataop']=$this->admin_model->get_cpuser($this->session->
            userdata('cpuser_username'));
        $pages['data'] = 'cpuser/home';
        $this->load->view('cpuser/dashboard', $pages);
    }
    else {
        $this->load->view('cpuser/login');
    }
}

```

Baris ke dua dari kode program 4 di atas berfungsi sebagai seleksi kondisi. Seorang pengguna sistem akan di tujukan kepada sebuah halaman login yang terletak pada folder view/cpuser/login.php seandainya user belum melakukan login. Seorang pengguna sistem akan ditujukan kepada sebuah halaman utama ketika telah melakukan login. Proses autentikasi berlangsung ketika seorang *user* melakukan proses login yang akan mengarahkan *user* kepada daftar action *auth\_cpuser*.

#### (b) Action auth\_cpuser

Auth\_cpuser merupakan daftar action dalam *controller* welcome yang berfungsi sebagai proses autentikasi *user*. Action *auth\_cpuser* akan menuliskan beberapa data kepada session browser.

Kode Program 5. action *auth\_cpuser*

---

```
function auth_cpuser() {
    $this->form_validation->set_rules('username', 'username',
    'required');
    $this->form_validation->set_rules('password', 'password',
    'required');
    if ($this->form_validation->run() == TRUE) {
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $data=$this->admin_model->get_cpuser($username,$password);
        foreach($data->result() as $row) {
            $kode_satgas=$row->kode_satgas;
            $previlage=$row->kode_user;
        }
        if ($this->admin_model->check_cpuser($username,$password) == TRUE)
        { $data = array(
            'cpuser_username' => $username,
            'kode_satgas' => $kode_satgas,
            'previlage' => $previlage,
            'cpuser_auth' => TRUE
        );
            $this->session->set_userdata($data);
        }
        Else {
            $this->session->set_flashdata('message_auth', 'Maaf , username
            atau password yang anda inputkan salah, silahkan masukkan data
            yang sesuai!!');
        }
    }
    Else {
        $this->session->set_flashdata('message_auth', 'Maaf , User Name
        dan Password Belum di Isi, Silahkan Mengisikan Username Dan
        Password yang sesuai!!');
    }
    redirect('welcome');
}
}
```

### 4.3 Proses Tambah

Proses tambah data adalah proses menambahkan data ke dalam database. Kode program dibagi menjadi tiga bagian, yaitu kode *view* untuk tampilan, *controller* yang diwakilkan kepada daftar action, dan *model*. Penjelasan untuk proses tambah akan diambilkan sebuah contoh pada data master satgas. Master satgas mempunyai controller *c\_satgas*. Daftar action yang digunakan dalam proses penambahan data adalah sebagai berikut:

#### (a) Daftar Action Index

Daftar action index dalam kaitan proses penambahan data digunakan sebagai pemanggil *form view*. *Form view* yang dimuat terdiri dari form *input* dan *output*.

Kode Program 6. Daftar Action Index Untuk Proses Tambah

---

```
function index() {
    if ($this->session->userdata('cpuser_auth') == TRUE) {
        $pages['output']=$this->m_satgas->tampil_all();
        $pages['data']='master/satgas';
        $this->load->view('cpuser/dashboard',$pages);
    }
    else {
        $this->load->view('cpuser/login');
    }
}
}
```

Kode program 6 di atas menggambarkan proses permintaan data kepada *model* *m\_satgas*, seperti yang terlihat pada baris ke-4. Kode program untuk *model* *m\_satgas* dapat dilihat dalam Kode Program 7.

**Kode Program 7.** *model* untuk action index *c\_satgas*

---

```
function tampil_all() {
    $getData = $this->db->query("Select * from satgas");
    return $getData;
}
```

Kode program 7 berfungsi untuk menampilkan seluruh data yang ada di dalam tabel *satgas*. Data yang dikirim oleh *model* akan disimpan dalam tipe data array yang bernama *pages* dengan index array *output*. Baris ke-5 pada kode program 6 berfungsi sebagai pemanggilan form view yang kemudian akan ditampilkan melalui Kode program pada baris ke-6.

#### (b) Daftar Action crud

Daftar action crud merupakan daftar action yang menangani proses submit yang dilakukan oleh pengguna sistem. Daftar action crud akan menangkap data yang telah disubmit oleh *user* melalui variabel *post*. Daftar action crud mempunyai dua fungsi. Fungsi yang pertama adalah untuk menambah data dan yang kedua adalah fungsi untuk ubah data.

**Kode Program 8.** action crud pada *controller* *c\_satgas*

---

```
function crud() {
    $data = array(
        'kode_satgas' => $this->input->post('kode_satgas'),
        'nama_satgas' => $this->input->post('nama_satgas'),
        'alamat' => $this->input->post('alamat'),
        'no_telepon' => $this->input->post('no_telepon'),
        'penyelia' => $this->input->post('penyelia'),
        'NIP' => $this->input->post('NIP') );
    if($this->input->post('simpan')) {
        $this->db->insert('satgas', $data);
        redirect('master/c_satgas');
    }
    elseif($this->input->post('edit')) {
        $kode_satgas=$this->input->post('kode_satgas');
        $this->db->update('satgas', $data,array('kode_satgas' =>
            $kode_satgas) );
        redirect('master/c_satgas');
    }
}
```

Kode program 8 baris ke-3 sampai dengan baris ke-8 merupakan proses deklarasi tipe data array yang diisi dengan sejumlah data yang dikirim oleh *user* melalui proses submit. Kode program yang berfungsi sebagai insert data terdapat pada baris ke-10.

## 4.4 Proses Cetak

Proses cetak merupakan sebuah proses pencetakan data pengujian yang telah diinputkan, ke dalam bentuk format yang terstruktur. Proses cetak dilakukan untuk pembuatan form hasil pengujian maupun form induk pengujian. Proses diawali dengan menampilkan data yang akan dicetak. Setelah data tampil, *user*

akan meminta untuk melakukan cetak data. Permintaan tersebut akan diteruskan kepada *controller* pada daftar action cetak.

**Kode Program 9.** Action cetak pada *controller* *c\_viabilitas*.

```
function cetak($no_lab)
{
    $kode_satgas=$this->session->userdata('kode_satgas');
    $data['cetak']=$this->m_viabilitas->cetak( $kode_satgas,$no_lab);
    $this->load->view('pengujian/viabilitas/cetak_hasil',$data);
}
```

Daftar action meminta data yang dibutuhkan kepada *model* terkait, dan kemudian mengirimkan data yang didapat ke dalam sebuah *variabel* array. Variabel array tersebut kemudian dikirim ke *form view* untuk ditampilkan sesuai dengan format yang di kehendaki.

## 4.5 Proses Download

Proses download merupakan proses pengunduhan laporan bulanan berbentuk file \*.xls. Proses diawali ketika seorang *user* melakukan proses submit tanggal awal laporan dan tanggal akhir laporan. Proses submit akan ditangani oleh *controller* dan dilanjutkan kepada daftar action bersangkutan. Setiap laporan mempunyai *controller* dan daftar action.

**Kode Program 10.** Action untuk mendownload laporan.

```
function cetak() {
    if ($this->session->userdata('cpuser_auth') == TRUE) {
        $tgl_awal=$this->input->post('tgl_awal');
        $tgl_akhir=$this->input->post('tgl_akhir');
        $data['tgl_awal']=$tgl_awal;
        $data['tgl_akhir']=$tgl_akhir;
        $data['kode_satgas']=$this->session->userdata('kode_satgas');
        $data['nama_satgas']=$this->m_standart_g->get_satgas($this->
            session->userdata('kode_satgas'));
        $data['bulan']=$this->get_bulan($tgl_akhir);
        $this->load->view('laporan/download/khusus',$data);
    }
    Else {
        $this->load->view('cpuser/login');
    }
}

function get_bulan($tgl_akhir) {
    $bulan=substr($tgl_akhir,5,2);
    $array_bulan = array('01'=>' Januari', '02'=>'Februari',
        '03'=>'Maret', '04'=>'April', '05'=>'Mei', '06'=>'Juni',
        '07'=>'Juli', '08'=>'Agustus', '09'=>'September',
        '10'=>'Oktober', '11'=>'November', '12'=>'Desember');
    return $array_bulan[$bulan];
}
```

Kode program 10 di atas berfungsi sebagai pemanggil file php yang diberi header xls. File tersebut mempunyai path di dalam folder *view/laporan/download*. Function *get\_bulan* pada basis ke-18 berfungsi untuk menkonversi format bulan kedalam format bulan dalam bahasa Indonesia. Laporan bulanan terdiri dari lima jenis laporan. Laporan bulanan standard mempunyai format yang sama dan mempunyai kode program yang sama pula. Kode program mempunyai perbedaan pada sintak perintah sqlnya saja. Kode program **file.php** yang telah diberi header file xls dapat dilihat dalam daftar kode program 11.

Kode Program 11. header file xls dalam file.php.

---

```
<head>
<? header("Content-type: application/octet-stream
header("Content-Disposition: attachment;
filename=LaporanBulananStandardGabungan.xls.
header("Pragma: no-cache");//Prevent Caching
header("Expires: 0
?>
</head>
```

## 5 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan dapat disimpulkan bahwa telah dihasilkan aplikasi Sistem Informasi Administrasi Pengujian Benih Tanaman Pangan dan Hortikultura Laboratorium BPSB Propinsi Jawa Timur yang dapat membantu mengatasi masalah dalam mengelola data pengujian benih, yang meliputi pengujian benih standar, pengujian benih berat 1000 butir, pengujian benih heterogenitas, dan pengujian benih viabilitas. Aplikasi ini juga dapat menghasilkan laporan bulanan yang dibutuhkan oleh satgas baik pusat maupun daerah. Laporan bulanan yang dihasilkan berupa file berformat xls.

## Referensi

- Basuki, P.A. (2010), *Membangun Web Berbasis PHP dengan framework CodeIgniter*, Yogyakarta, Lokomedia .
- Munawar (2005), *Pemodelan Visual dengan UML*, Jakarta, Graha Ilmu.
- Newton, A. (Mei 2009), *jQuery vs MooTools*, ditemukanali 15 juni 2011, dari : <http://jqueryvsmootools.com/>
- Nugroho, B., (2008), *Panduan Lengkap Menguasai Perintah SQL*, Media Kita, Jakarta.
- Oktaviani, D. (Desember 2010), *Konsep MVC (Model View Controller)*, ditemukanali 09 maret 2011, Dari: <http://oktavianidewi.wordpress.com/2010/12/03/konsep-mvc-model-view-controller-bukan-pvc>
- jQuery, (15 Maret 2011). *Kenalan Sama jQuery yuk*, ditemukanali 15 Juni 2011, dari: <http://blog.unikom.ac.id/v/1G9/>
- Wardana (2010), *Menjadi Master PHP dengan Framework CodeIgniter*, Jakarta, PT Elex Media Komputindo.

\*\*\*