

HASIL CEK2_Makalah 11

by Makalah 11 Bpk Suprihatin

Submission date: 11-May-2022 11:35AM (UTC+0700)

Submission ID: 1833565379

File name: makalah11.pdf (1.31M)

Word count: 4229

Character count: 26790

Sistem Informasi Presensi Menggunakan Sidik Jari (Study Kasus Presensi Perkuliahan Program Studi Sistem Informasi FMIPA UAD)

Suprihatin¹, Andi Nurhantara²

¹ Program Studi Sistem Informasi, Universitas Ahmad Dahlan, Yogyakarta
Email : pakprih@yahoo.co.id

ABSTRACT. Sistem perkuliahan dapat berjalan dengan baik, jika pencatatan kehadiran mahasiswa yang efektif dan efisien. Pencatatan kehadiran mahasiswa di Universitas Ahmad Dahlan Yogyakarta digunakan sebagai syarat untuk mengikuti UAS (Ujian Akhir Semester) yaitu minimal 75% mahasiswa harus menghadiri perkuliahan. Pencatatan kehadiran mahasiswa juga digunakan sebagai catatan bahwa mahasiswa mengikuti perkuliahan serta dapat digunakan sebagai bahan pembandingan prestasi mahasiswa. Sistem pencatatan kehadiran mahasiswa yang digunakan saat ini dengan diberi lembaran untuk ditandatangani setiap mahasiswa mengikuti perkuliahan kurang efektif dan efisien karena peluang mahasiswa untuk melakukan manipulasi data kehadiran tersebut terhitung masih terjadi. Sidik jari merupakan salah satu ciri dari seseorang (biometrika) sehingga setiap mahasiswa pasti mempunyai sidik jari yang berbeda dengan mahasiswa yang lainnya dan tidak mungkin ada sistem penitipan presensi pada mahasiswa lain.

Subyek penelitian dalam tugas akhir ini adalah suatu Aplikasi pencatatan kehadiran mahasiswa menggunakan sidik jari (Studi kasus presensi mahasiswa Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengantahuan Alam Universitas Ahmad Dahlan) menggunakan Microsoft Visual Basic 6.0 yang diharapkan dapat menjadikan suatu sistem presensi yang efektif dan efisien demi memunjangnya kemajuan teknologi dan prestasi mahasiswa.

Hasil yang diperoleh adalah suatu program Sistem Informasi Presensi mahasiswa dengan menggunakan Fingerprint Scanner yang dapat mengurangi kecurangan dalam melakukan presensi, meningkatkan keaktifan dan kedisiplinan mahasiswa serta mempermudah TU selaku admin untuk menentukan mahasiswa boleh ujian atau tidaknya.

Keywords: Sidik Jari, Presensi, Biometrika, Pencatatan kehadiran, Fingerprint.

1 Pendahuluan

Universitas Ahmad Dahlan (UAD) Yogyakarta adalah institusi pendidikan yang menerapkan sistem informasi. Sistem perkuliahan dapat berjalan dengan baik, jika pencatatan kehadiran mahasiswa yang efektif dan efisien. Pencatatan kehadiran mahasiswa digunakan sebagai syarat untuk mengikuti UAS (Ujian Akhir Semester) yaitu minimal 75% mahasiswa harus menghadiri perkuliahan. Pencatatan kehadiran mahasiswa juga digunakan sebagai catatan bahwa mahasiswa mengikuti perkuliahan serta dapat digunakan sebagai bahan pembandingan prestasi mahasiswa. Universitas Ahmad Dahlan saat ini masih menggunakan pencatatan kehadiran mahasiswa secara manual, yaitu mahasiswa diberi lembaran untuk ditandatangani setiap mahasiswa mengikuti perkuliahan. Sistem pencatatan kehadiran mahasiswa yang digunakan saat ini kurang efektif dan efisien karena peluang mahasiswa untuk melakukan manipulasi data kehadiran tersebut terhitung masih terjadi. Bidang Tata Usaha juga mengalami kesulitan dalam menentukan apakah mahasiswa dapat mengikuti ujian atau tidak

mengikuti ujian. Sidik jari merupakan salah satu ciri dari seseorang sehingga setiap mahasiswa pasti mempunyai sidik jari yang berbeda dengan mahasiswa yang lainnya dan tidak mungkin ada sistem penitipan presensi pada mahasiswa lain. Aplikasi ini diharapkan dapat mengurangi bahkan menghilangkan manipulasi data kehadiran mahasiswa tersebut khususnya pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Ahmad Dahlan Yogyakarta

2 Landasan Teori

2.1 Penelitian Terdahulu

Penelitian ini merujuk pada penelitian yang dilakukan oleh Bramantyo Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada Yogyakarta 2008, yang berjudul “Design Pattern MVC (Model View Controller) Penerapan Dalam Membangun Framework Aplikasi web berbasis Java”. Dalam Penelitian ini study kasus diimplementasikan pada sebuah system informasi pengisian KRS online menggunakan aplikasi web berbasis Java.

Penelitian ini merujuk pada penelitian yang dilakukan oleh Haris Sri Wibowo Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Ahmad Dahlan 2006, yang berjudul “Pemodelan Analisis Dan Desain Sistem Berorientasi Objek Menggunakan UML”. Dalam Penelitian tersebut, UML digunakan sebagai alat bantu pemodelan analisis dan desain sistem berorientasi objek pada pengembangan sistem informasi penjadwalan kuliah di FMIPA-UAD.

Penelitian yang dilakukan di Laboratorium UPTPSBTPH Propinsi Jawa Timur ini memanfaatkan UML sebagai alat bantu analisis dan desain, sedangkan aplikasinya dibangun menggunakan PHP dengan *frame work* CodeIgniter yang berpola MVC.

2.2 *Frame work* PHP

Frame work 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).

MVC terdiri dari 3 bagian yaitu: Model, View dan Controller. Dalam website dinamis, terdiri dari 3 hal yang paling pokok, yaitu: basis data, logika aplikasi dan cara menampilkan halaman website. Ketiga hal tersebut direpresentasikan dengan MVC, yaitu: Model untuk basis data, View untuk cara menampilkan halaman website dan Controller untuk logika aplikasi.

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.

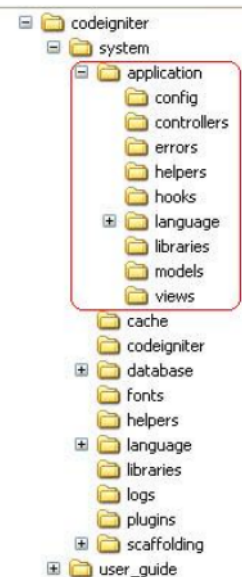
View merupakan informasi yang ditampilkan kepada pengunjung website. Sedapat mungkin view tidak berisi logika-logika kode, tetapi hanya berisi variable-variabel yang berisi data yang siap ditampilkan. View merupakan halaman website yang dibuat menggunakan HTML dengan bantuan Javascript atau CSS. View 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).

2.3 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 di atas dapat ditunjukkan bahwa aplikasi yang dikembangkan berpusat pada folder application. Wardana (2010) mendefinisikan fungsi dari masing-masing folder yang berada didalam folder system adalah sebagai berikut :



Gambar 1. Struktur direktori CodeIgniter

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

2.4 jQuery

John Resig menulis sebuah Javascript Library dan kini dikenal dengan nama jQuery. Sejak dirilis pertama kali pada tahun 2006, jQuery telah mencuri perhatian para developer web. jQuery telah banyak digunakan oleh website-website terkemuka didunia 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 (Query, 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.*”

2.5 Database

Database 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 (Disebut tabel), di mana 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 berjalanya waktu dan banyaknya peminat yang menggunakan datanase ini, MySQL merilis versi yang hampir dapat diinstall pada hampir semua platform, termasuk Windows.

2.6 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 standart 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.

6 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

46 Kebutuhan UPTPSBTPH terhadap distem disajikan pada tabel 1.

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

| Kebutuhan | Aktor | Use Case |
|----------------------------|---------------------------------|---------------------------------------|
| 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 standart. |
| 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 Standart gabungan |

Tabel 1. Tabel kebutuhan user (*lanjutan*)

| | | |
|---------------------|---------------------------------|--|
| Download Laporan | Operator Pusat, Operator Daerah | Laporan Bulanan uji standart komoditi Hortikultura |
| Download Laporan | Operator Pusat, Operator Daerah | Laporan Bulanan Uji Standart komoditi Pangan Daerah |
| Download Laporan | Operator Pusat, Operator Daerah | Membuat Laporan Bulanan Uji Standart komoditi pangan Negara |
| Download Laporan | Operator Pusat | Membuat rekapitulasi Laporan Bulanan Uji Standart gabungan seluruh satgas |
| Download Laporan | Operator Pusat | Membuat rekapitulasi Laporan bulanan uji standart komoditi Hortikultura gabungan seluruh satgas |
| Download Laporan | Operator Pusat | Membuat rekapitulasi Laporan bulanan uji standart komoditi Hortikultura gabungan seluruh satgas |
| Download Laporan | Operator Pusat | Membuat rekapitulasi Laporan Bulanan Uji Standart 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.

²⁴ (b) Administrasi Satgas Daerah

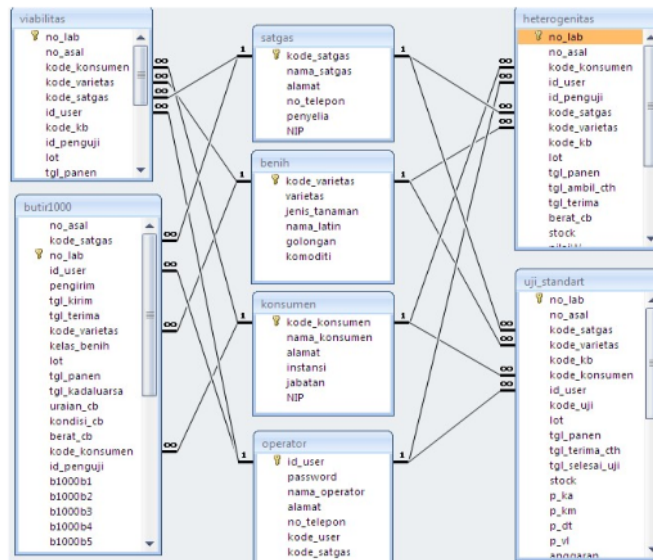
User administrasi daerah merupakan *user* yang berada di daerah atau *user* selain petugas administrasi laboratorium I Surabaya. *User* administrasi daerah terdiri dari seluruh petugas administrasi laboratorium daerah meliputi laboratorium II Madiun, laboratorium III Malang, laboratorium IV Kediri, laboratoium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

²⁴ (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.

²⁹ **3.2 Relasi Antar Tabel**

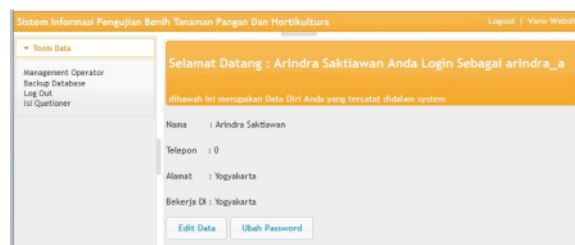
Relasi antar tabel-tabel di atas dapat dilihat pada gambar 2 berikut:



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, yang ⁴⁰ dalamnya terdapat satu menu dengan 4 sub menu. Tampilan untuk form utama untuk administrator ⁴⁰ dapat dilihat dalam gambar 3 di bawah ini.



Gambar 3. Form utama untuk administrator

Tampilan form utama ⁴¹ 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 standart**, **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. Contoh tampilan form disajikan pada gambar 4.

| No. | No. Lab. | No. Asal | Konsumen | Alamat | Jenis Tanaman | Varietas | Edit | Hapus | Cetak |
|-----|----------|----------|----------|---------------------------------------|---------------|-------------|------|-------|-------|
| 1. | P-01-N-C | P-01-N | Soyadi | Sertifikasi Tan. Pangan Surabaya | Padri | Cibarang | ✓ | 🗑️ | 🖨️ |
| 2. | P-02-N | P-01-N | Siamet S | F. Sertifikasi UPTPSETH Satgas Jember | Padri | Cianggarung | ✓ | 🗑️ | 🖨️ |
| 3. | P-01-N | P-01-N | Didik | Sertifikasi Hortikultura Surabaya | Padri | Cibogo | ✓ | 🗑️ | 🖨️ |

Gambar 4. Interface Pengujian Standart

Berdasarkan pada gambar 4 di atas, input pengujian standart dibagi menjadi dua bagian. Bagian pertama merupakan identitas pengujian. Identitas pengujian tersebut meliputi input tentang atribut dari benih dan atribut pengujian. Atribut benih terdiri input-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 standart adalah input mengenai hasil detail dari pengujian. Tampilan tahap ke dua pengujian standart dapat dilihat dalam gambar 5 di bawah ini :

| No. | No. Lab. | No. Asal | Konsumen | Alamat | Jenis Tanaman | Varietas | Edit | Hapus | Cetak |
|-----|----------|----------|----------|---------------------------------------|---------------|-------------|------|-------|-------|
| 1. | P-01-N-C | P-01-N | Soyadi | Sertifikasi Tan. Pangan Surabaya | Padri | Cibarang | ✓ | 🗑️ | 🖨️ |
| 2. | P-02-N | P-01-N | Siamet S | F. Sertifikasi UPTPSETH Satgas Jember | Padri | Cianggarung | ✓ | 🗑️ | 🖨️ |
| 3. | P-01-N | P-01-N | Didik | Sertifikasi Hortikultura Surabaya | Padri | Cibogo | ✓ | 🗑️ | 🖨️ |

Gambar 5. Tampilan tahap ke 2 pengujian standart

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 standart 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. *Interface* dari form input tanggal awal dan tanggal akhir dapat dilihat dalam gambar 6 di atas.

4. Pengkodean

4.1 Config codeigniter

Config codeigniter merupakan folder yang didalamnya terdapat pengaturan-pengaturan yang berkaitan dengan sistem yang akan di buat. 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 untuk autoload.php dapat dilihat dalam kode program 1.

Kode Program 1. autoload.php

```
1. $autoload['libraries'] = array('database',  
2. 'form_validation', 'table');  
3. $autoload['helper'] = array('form', 'url', 'json');  
4. $autoload['plugin'] = array();  
5. $autoload['config'] = array();  
6. $autoload['language'] = array();  
7. $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 untuk file database.php dapat dilihat dalam Kode program 2.

Kode Program 2. database.php

```
1. $db['default']['hostname'] = "localhost";  
2. $db['default']['username'] = "root";  
3. $db['default']['password'] = "12345678";  
4. $db['default']['database'] = "db_bpsb";  
5. $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 untuk pengaturan tersebut dapat dilihat dalam kode program 3.

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

Seperti yang telah dijelaskan di atas bahwa welcome *controller* merupakan *controller* yang pertama kali di panggil 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 untuk action index dapat dilihat dalam daftar Kode program 4.

Kode Program 4 controller action index

```
1. function index(){
2.   if ($this->session->userdata('cpuser_auth') == TRUE){
3.     $pages['dataop']=$this->admin_model->get_cpuser($this->session->
   userdata('cpuser_username'));
4.     $pages['data'] = 'cpuser/home';
5.     $this->load->view('cpuser/dashboard', $pages);
6.   }
7.   else{
8.     $this->load->view('cpuser/login');
9.   }
}
```

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 untuk action *auth_cpuser* dapat dilihat dalam daftar kode program 5 .

Kode Program 5. action auth_cpuser

```
1. function auth_cpuser()
2. { $this->form_validation->set_rules('username', 'username', 'required');
3.   $this->form_validation->set_rules('password', 'password', 'required');
4.   if ($this->form_validation->run() == TRUE)
5.   { $username = $this->input->post('username');
6.     $password = $this->input->post('password');
7.     $data=$this->admin_model->get_cpuser($username, $password);
8.     foreach($data->result() as $row)
9.     { $kode_satgas=$row->kode_satgas;
10.      $previlage=$row->kode_user;
11.    }
12.    if ($this->admin_model->check_cpuser($username, $password) == TRUE)
13.    { $data = array(
```

```

14.         'cpuser_username'    => $username,
15.         'kode_satgas'        => $kode_satgas,
16.         'previlage'          => $previlage,
17.         'cpuser_auth'        => TRUE
18.     );
19.     $this->session->set_userdata($data);
20. }
21. Else
22. { $this->session->set_flashdata('message_auth', 'Maaf , username
    atau password yang anda inputkan salah, Silahkan masukkan data
    yang sesuai!!');
23. }
24. }
25. Else
26. { $this->session->set_flashdata('message_auth', 'Maaf , User Name dan
    Password Belum di Isi, Silahkan Mengisikan Username Dan Password
    yang sesuai!!');
27. }
28. redirect('welcome');
29. }

```

4.3 Proses Tambah

Proses tambah data adalah proses menambahkan data kedalam 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 untuk action index dapat dilihat dalam kode program 6.

Kode Program 6. Daftar Action Index Untuk Proses Tambah

```

34
1.  function index()
2.  {
3.      if ($this->session->userdata('cpuser_auth') == TRUE){
4.          $pages['output']=$this->m_satgas->tampil_all();
5.          $pages['data']='master/satgas';
6.          $this->load->view('cpuser/dashboard', $pages);
7.      }
8.      else{
9.          $this->load->view('cpuser/login');
10.     }
11. }

```

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*

```

1.  function tampil_all()
2.  {
3.      $getData = $this->db->query("Select * from satgas");
4.      return $getData;
5.  }

```

Kode program 7 di atas berfungsi ⁴⁵ 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 ⁴⁴ 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 ¹⁷ menambah data dan yang kedua adalah fungsi untuk ubah data. Kode program untuk daftar action crud dapat dilihat dalam Kode program 8 di bawah ini.

Kode Program 8. action crud pada *controller c_satgas*

```
1. function crud()
2. { $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' => $this->input->post('NIP') );
3. if($this->input->post('simpan'))
4. { $this->db->insert('satgas', $data);
5.   redirect('master/c_satgas');
6. }
7. elseif($this->input->post('edit'))
8. { $kode_satgas=$this->input->post('kode_satgas');
9.   $this->db->update('satgas', $data, array('kode_satgas' =>
    $kode_satgas) );
10.  redirect('master/c_satgas');
11. }
12. }
```

Kode program 8 di atas 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 ¹⁷ akan diteruskan kepada *controller* pada daftar action cetak. Kode program untuk action cetak dapat dilihat dalam kode program 9 di bawah ini.

Kode Program 9. Action cetak pada *controller c_viabilitas*.

```
1. function cetak($no_lab)
2. { $kode_satgas=$this->session->userdata('kode_satgas');
3.   $data['cetak']=$this->m_viabilitas->cetak( $kode_satgas, $no_lab);
4.   $this->load->view('pengujian/viabilitas/cetak_hasil', $data);
5. }
```

Daftar action meminta data yang dibutuhkan kepada *model* terkait, dan kemudian mengirimkan data yang didapat kedalam sebuah *variabel* array. Variabel array tersebut kemudian dikirim ke *form view* untuk di tampilkan 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. Adapun kode programnya [dapat dilihat pada kode program 10](#).

Kode Program 10. Action untuk mendownload laporan.

```
1. function cetak()
2. { if ($this->session->userdata('cpuser_auth') == TRUE)
3.   {
4.     $tgl_awal=$this->input->post('tgl_awal');
5.     $tgl_akhir=$this->input->post('tgl_akhir');
6.     $data['tgl_awal']=$tgl_awal;
7.     $data['tgl_akhir']=$tgl_akhir;
8.     $data['kode_satgas']=$this->session->userdata('kode_satgas');
9.     $data['nama_satgas']=$this->m_standart_g->get_satgas($this->
10.     session->userdata('kode_satgas'));
11.     $data['bulan']=$this->get_bulan($tgl_akhir);
12.     $this->load->view('laporan/download/khusus', $data);
13.   }
14.   Else
15.   { $this->load->view('cpuser/login');
16. }
17. }
18. function get_bulan($tgl_akhir)
19. { $bulan=substr($tgl_akhir,5,2);
20.   $array_bulan = array('01'=>' Januari ', '02'=>'Februari ',
21.     '03'=>'Maret ', '04'=>'April ', '05'=>' Mei ', '06'=>'Juni ',
22.     '07'=>'Juli ', '08'=>'Agustus ', '09'=>'September ',
23.     '10'=>'Oktober ', '11'=>'November ', '12'=>'Desember');
24.   return $array_bulan[$bulan];
25. }
```

Kode program 10 di atas berfungsi sebagai pemanggil file php yang diberi header xls. File tersebut mempunyai path didalam folder view/laporan/download. Function *get_bulan* pada baris ke-18 berfungsi untuk menkonversi format bulan kedalam format bulan dalam bahasa Indonesia. Laporan bulanan terdiri dari lima jenis laporan. Laporan bulanan standart 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 di bawah ini.

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

```
27 <head>
28   <? header("Content-type: application/octet-stream
29     header("Content-Disposition: attachment;
30     filename=LaporanBulananStandartGabungan.xls.
31     header("Pragma: no-cache");//Prevent Caching
32     header("Expires: 0
33   ?>
34 </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 standart, 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.

Sistem Informasi Administrasi Pengujian Benih Tanaman Pangan dan Hortikultura Laboratorium BPSB Propinsi Jawa Timur dapat menjadi salah satu solusi bagi permasalahan yang dihadapi oleh laboratorium UPT PSBTPH propinsi Jawa Timur.

Referensi

- Basuki, P.A. (2010), *Membangun Web Berbasis PHP dengan framework CodeIgniter*, Yogyakarta, Lokomedia .
- Bramantyo (2008), *Design Pattern MVC (Model View Controler) Penerapan Dalam Membangun Framework Aplikasi web berbasis Java*, Skripsi, Ilmu Komputer, Universitas Gajah Mada, Yogyakarta.
- 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. (2004), *Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL: Studi kasus, membuat sistem informasi Pengolahan Data Buku*, Yogyakarta, Gava Media.
- Nugroho,B., (2008), *Panduan Lengkap Menguasai Perintah SQL*, Media Kita, Jakarta.
- Oktaviani, D. (Desember 2010), *Konsep MVC (Model View Controller)*, ditemu kenali 09 maret 2011, Dari: <http://oktavianidewi.wordpress.com/2010/12/03/konsep-mvc-model-view-controller-bukan-pvc>
- Query, J. (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.
- Wibowo, S.H. (2006), *Pemodelan Analisis Dan Desain Sistem Berorientasi Objek Menggunakan UML*, Skripsi, Ilmu Komputer, Universitas Ahmad Dahlan, Yogyakarta

HASIL CEK2_Makalah 11

ORIGINALITY REPORT

25%

SIMILARITY INDEX

24%

INTERNET SOURCES

6%

PUBLICATIONS

11%

STUDENT PAPERS

PRIMARY SOURCES

| | | |
|---|---|----|
| 1 | repository.ipb.ac.id Internet Source | 1% |
| 2 | repository.ub.ac.id Internet Source | 1% |
| 3 | repository.amikom.ac.id Internet Source | 1% |
| 4 | es.scribd.com Internet Source | 1% |
| 5 | www.tokolokomedia.com Internet Source | 1% |
| 6 | godfr4.blogspot.com Internet Source | 1% |
| 7 | ijns.org Internet Source | 1% |
| 8 | ojs.widyakartika.ac.id Internet Source | 1% |
| 9 | bastianharry.blogspot.com Internet Source | 1% |

| | | |
|----|---|-----|
| 10 | jurnal.untan.ac.id Internet Source | 1 % |
| 11 | repository.uinjkt.ac.id Internet Source | 1 % |
| 12 | wordislam.blogspot.com Internet Source | 1 % |
| 13 | Submitted to Sriwijaya University Student Paper | 1 % |
| 14 | repository.uksw.edu Internet Source | 1 % |
| 15 | www.skripsiinformatika.com Internet Source | 1 % |
| 16 | www.termpaperwarehouse.com Internet Source | 1 % |
| 17 | journal.unilak.ac.id Internet Source | 1 % |
| 18 | teknikinformatikacatatan.blogspot.com Internet Source | 1 % |
| 19 | repository.its.ac.id Internet Source | 1 % |
| 20 | Submitted to Universitas Muria Kudus Student Paper | 1 % |
| 21 | core.ac.uk Internet Source | 1 % |

| | | |
|----|---|------|
| 22 | blog.unikom.ac.id Internet Source | 1 % |
| 23 | Ardian Dwi Praba. "Implementasi Model View Controller Dengan framework CodeIgniter Pada Perpustakaan", Indonesian Journal on Software Engineering (IJSE), 2018 Publication | 1 % |
| 24 | mypdfdownload.info Internet Source | 1 % |
| 25 | www.jurnalskripsi.net Internet Source | <1 % |
| 26 | library.binus.ac.id Internet Source | <1 % |
| 27 | abhimanyuajad.blogspot.com Internet Source | <1 % |
| 28 | blog.uad.ac.id Internet Source | <1 % |
| 29 | eprints.umg.ac.id Internet Source | <1 % |
| 30 | Submitted to KYUNG HEE UNIVERSITY Student Paper | <1 % |
| 31 | nanopdf.com Internet Source | <1 % |
| 32 | ejournal.kopertis10.or.id Internet Source | <1 % |

33 Submitted to Forum Perpustakaan Perguruan
Tinggi Indonesia Jawa Timur <1 %
Student Paper

34 Submitted to The University of Manchester <1 %
Student Paper

35 anitamegayanti.files.wordpress.com <1 %
Internet Source

36 canggih.web.id <1 %
Internet Source

37 jurnal.lppm.unsoed.ac.id <1 %
Internet Source

38 pdm-mipa.ugm.ac.id <1 %
Internet Source

39 repository.sar.ac.id <1 %
Internet Source

40 download.garuda.ristekdikti.go.id <1 %
Internet Source

41 belajarinternetmarketingpasuruan.blogspot.com <1 %
Internet Source

42 mayopu.wordpress.com <1 %
Internet Source

43 novitadwipermatasari.blogspot.com <1 %
Internet Source

44 avivadevi.blogspot.com
Internet Source

<1 %

45 fr.slideshare.net
Internet Source

<1 %

46 kumpulanmakalah2018.blogspot.com
Internet Source

<1 %

47 www.pustakaskripsi.com
Internet Source

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On