

# PERANCANGAN SISTEM INFORMASI REKAM MEDIS RAWAT INAP RUMAH SAKIT BERBASIS WEB

Nunik Srikandi Putri, Adian Fatchur Rochim, Ike Pertiwi Windasari  
Jurusan Sistem Komputer Fakultas Teknik Universitas Diponegoro  
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia.

## ABSTRAK

*Instalasi rekam medis merupakan tahapan proses pencatatan data pasien, pendaftaran pasien yang datang untuk kunjungan yang baru pertama kali atau berikutnya ke sebuah rumah sakit untuk mendapatkan nomor rekam medik ( medical record ). Pasien datang dengan mendaftarkan diri ke bagian rekam medis namun pasien yang datang tidak semua dalam keadaan sadar. Proses pencarian data pasien kadang memakan waktu yang relatif tidak sedikit akibat kartu member yang hilang menyebabkan pasien yang sama bisa mendapatkan nomor rekam medis yang lebih dari satu. Kejadian ini kerap terjadi apabila sistem yang berjalan di rumah sakit yang belum terkomputerisasi.*

*Dilatarbelakangi hal tersebut penulis mencoba menyusun Tugas Akhir Perancangan Sistem Informasi Rekam Medis Rawat Inap Berbasis Web. Perancangan aplikasi ini menggunakan UML, dengan database engine Mysql dan bahasa pemrograman PHP. Aplikasi ini juga didukung oleh javascript, jquery dan AJAX untuk membuat antarmuka menjadi lebih cepat terutama dalam penampilan data dan pencarian secara live search. Pengujian aplikasi ini dengan metode blackbox. Dengan aplikasi ini, pengguna dapat memperoleh pelayanan yaitu berupa data rekam medis dari seluruh pasien pada bagian rawat inap di rumah sakit tersebut yang dapat diakses dengan mudah. Hasil pengujian aplikasi ini meningkatkan pengelolaan data rekam medis menjadi lebih mudah dan efisien dilakukan, aman karena data tersimpan dalam database, tidak cepat rusak, mempermudah dalam pencarian data, serta lebih terstruktur dibandingkan dengan pengelolaan secara manual.*

**Kata Kunci :** Rekam Medis, PHP, MySql, sistem informasi

## 1) PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi komputer merupakan salah satu teknologi yang berkembang dengan pesat. Di zaman yang serba membutuhkan teknologi ini tidak pernah terlepas dari teknologi *software*. Perubahan lingkungan yang terjadi selama dekade terakhir ini, suatu rumah sakit juga tidak terlepas dari pengaruh yang ditimbulkan oleh dampak global sehingga menuntut adanya pertukaran informasi secara cepat, tepat dan akurat. Dengan kemajuan teknologi informasi, pengaksesan terhadap data atau informasi yang tersedia dapat berlangsung dengan cepat, efisien serta akurat. Rumah sakit sebagai salah satu institusi pelayanan kesehatan masyarakat melayani pasien dalam kesehariannya. Dalam hal tersebut petugas paramedik memerlukan informasi mengenai riwayat rekam medis pasien khususnya di instalasi rawat inap. Kegunaan sistem informasi rekam medis rawat inap rumah sakit yaitu seluruh pihak rumah sakit akan dapat dengan mudah, cepat, akurat dalam:

1. Mengetahui dan menyajikan data-data pasien
2. Melakukan pengolahan data secara terintegrasi dari setiap unit
3. Tetap *Up To Date* dengan berbagai kegiatan maupun perkembangan di rumah sakit dimana mereka berada dan kapanpun memerlukannya.

### 1.2 Tujuan Penelitian

Tujuan dari pengembangan sistem informasi ini adalah untuk mengembangkan suatu sistem informasi rekam medis rawat inap rumah sakit berbasis web.

### 1.1 Batasan Masalah

Dalam penulisan tugas akhir ini pembahasan masalah memiliki batasan pada permasalahan berikut :

- a. Bagaimana merancang sistem informasi rekam medis dengan menggunakan PHP dan MySQL.

- b. Sistem Informasi yang dibuat adalah sistem yang didasarkan pada aktifitas rekam medis rawat inap.

## 2) LANDASAN TEORI

### 2.1 Definisi Rekam Medis

- a. Definisi Rekam Medis Menurut Edna K Huffman  
Rekam Medis adalah berkas yang menyatakan siapa, apa, mengapa, dimana, kapan dan bagaimana pelayanan yang diperoleh seorang pasien selama dirawat atau menjalani pengobatan. [6]
- b. Definisi Rekam Medis Menurut Permenkes No. 749a/MenkesPer/XII/1989  
Rekam Medis adalah berkas yang berisi catatan dan dokumen mengenai identitas pasien, hasil pemeriksaan, pengobatan, tindakan dan pelayanan lainnya yang diterima pasien pada sarana kesehatan, baik rawat jalan maupun rawat inap. [5]
- c. Definisi Rekam Medis Menurut Gemala Hatta  
Rekam Medis merupakan kumpulan fakta tentang kehidupan seseorang dan riwayat penyakitnya, termasuk keadaan sakit, pengobatan saat ini dan saat lampau yang ditulis oleh para praktisi kesehatan dalam upaya mereka memberikan pelayanan kesehatan kepada pasien.[4]

### 2.2 PHP

PHP (PHP Hypertext Preprocessor) adalah bahasa berbentuk script yang ditempatkan di server dan diproses di server. Hasilnya akan dikirim ke klien, tempat pemakai menggunakan browser. PHP dirancang untuk dapat membangun sebuah web dinamis. Artinya, PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini. Kode PHP dapat disertakan sebagai tag pada kode HTML.

### 2.3 MySql

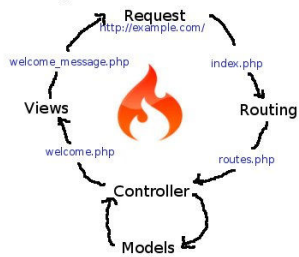
Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin, seseorang dapat membuat database, membuat tabel, mengisi

data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya.

PhpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada di komputer. Untuk membukanya, buka browser lalu ketikkan alamat <http://localhost/phpmyadmin>, maka akan muncul halaman phpMyAdmin. Di situ nantinya seseorang bisa membuat (*create*) basis data baru, dan mengelolanya.[1].

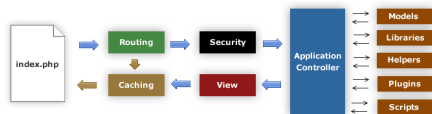
## 2.4 Code Igniter

CodeIgniter adalah aplikasi open source yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. CodeIgniter memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal.



**Gambar 2.1** Diagram Code Igniter seolah-olah sebuah struktur cangkir

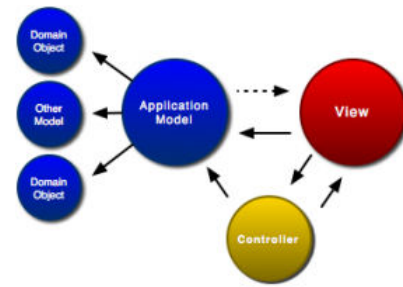
Code Igniter tergolong *framework* dengan ukuran kecil dan cukup mudah di kuasai. *Framework* itu sendiri adalah suatu kerangka kerja yang berupa sekumpulan folder yang memuat *file-file* php yang menyediakan *class libraries, helpers, plugins* dan lainnya.



**Gambar 2.2** Diagram kerja Code Igniter

CodeIgniter dirancang berdasarkan metode pengembangan *Model-View-Controller* (MVC). Metode ini memisahkan kode dalam 3 bagian, yaitu:

1. Model  
Mendefinisikan *database* dan *class* yang digunakan untuk membaca atau menyimpan informasi di *database*.
2. View  
Mendefinisikan tampilan halaman *website* (*template*), baik halaman web secara utuh atau potongan kode seperti *header, footer*. Dapat juga membuat halaman RSS atau 'halaman' dalam bentuk lainnya.
3. Controller  
Menangani *request* dari pengguna dan menghasilkan halaman web. Mengontrol alur proses antara *Model, View* dan *resource* lainnya.



**Gambar 2.3** Konsep MVC

## 2.5 Javascript

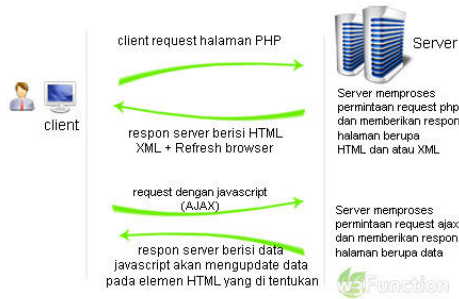
JavaScript adalah sebuah bahasa pemrograman yang cukup terstruktur. Salah satu ciri dari bahasa pemrograman terstruktur adalah mengizinkan pemrogram mengumpulkan beberapa perintah atau kode di dalam suatu wadah *function* atau *event*.

Berdasarkan definisinya, *function* adalah sekumpulan perintah yang digunakan untuk menjalankan proses tertentu, dimana proses tersebut boleh mengembalikan sebuah nilai kembalian (boleh juga tidak). Secara umum, struktur pendefinisian *function* setidaknya memiliki tiga elemen dasar, yaitu :

1. Nama fungsi harus dinyatakan secara jelas dengan menggunakan sebuah nama yang tidak termasuk di dalam kata-kata kunci yang sudah dipesan oleh JavaScript, jadi tidak boleh menggunakan nama *function* dengan nama "string", "array" atau kata-kata kunci JavaScript lainnya.
2. Parameter argumen, merupakan *variable* masukan jika menginginkan adanya nilai masukan sebelum menggunakan sebuah *function*. Kita boleh mendefinisikan parameter ini atau tidak mendefinisikan parameter. Jika kita memilih untuk menyatakan adanya parameter argumen, kita boleh menyatakan satu atau lebih parameter. Jika kita menyatakan lebih dari satu parameter, masing-masing parameter dipisahkan dengan tanda koma.
3. Perintah-perintah *function*, dinyatakan di dalam cakupan tanda kurung kurawal.

## 2.6 AJAX

AJAX memiliki kepanjangan Asynchronous Javascript And XML merupakan suatu teknik baru dalam dunia web tapi bukan merupakan bahasa pemrograman yang baru. Dengan adanya AJAX, akses data ke *server* yang dikirim melalui *client* via *web* dapat lebih cepat daripada mekanisme biasa. Hal ini dikarenakan AJAX tidak perlu melakukan proses *loading page* (*refresh page*) atau pindah ke *page* yang lain. AJAX dapat diintegrasikan dengan *server side programming* seperti PHP, ASP, JSP dll. (<http://www.w3function.com>).



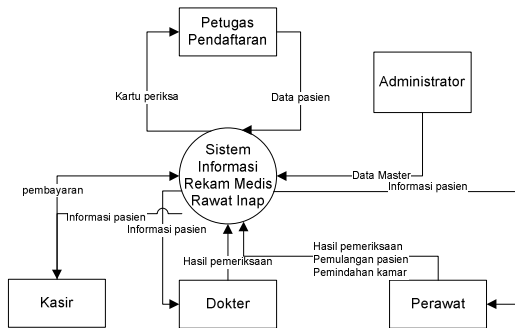
**Gambar 2.4** Mekanisme AJAX

Mekanisme di atas menjelaskan bahwa proses berawal dari *web client*. *Web client* me-request sebuah halaman PHP ke *server* melalui Javascript. Selanjutnya *server* akan merespon dan menjalankan *script* PHP. *Script* PHP akan menghasilkan respon dalam bentuk XML dan data XML tersebut akan dikirim kembali ke *web client* untuk diolah oleh Javascript. Hasil olahan javascript tersebut akhirnya akan ditampilkan di *web client* sebagai *output* tanpa harus me-refresh halaman *web*.

### 3) ANALISIS PERANCANGAN SISTEM

#### 3.1. Diagram Konteks

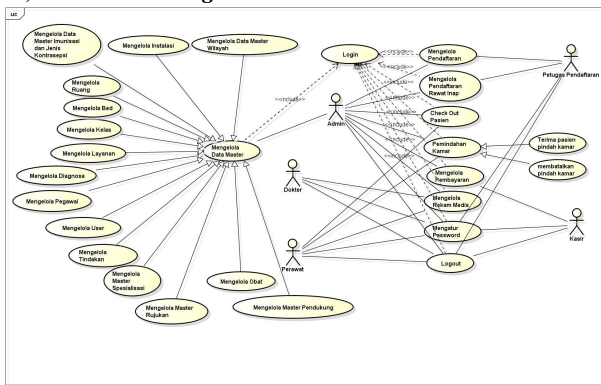
Desain proses akan menjelaskan gambaran sistem dan arus data yang digunakan. Aliran data dalam sistem informasi rekam medis Rumah Sakit ini memiliki entitas-entitas luar yang berinteraksi langsung dengan sistem, yaitu : petugas pendaftaran, perawat, petugas kasir, dokter, dan administrator. Aliran data sistem informasi rekam medis Rumah Sakit dapat diuraikan dengan diagram konteks pada gambar berikut,



**Gambar 3.1** Diagram Konteks Sistem Informasi

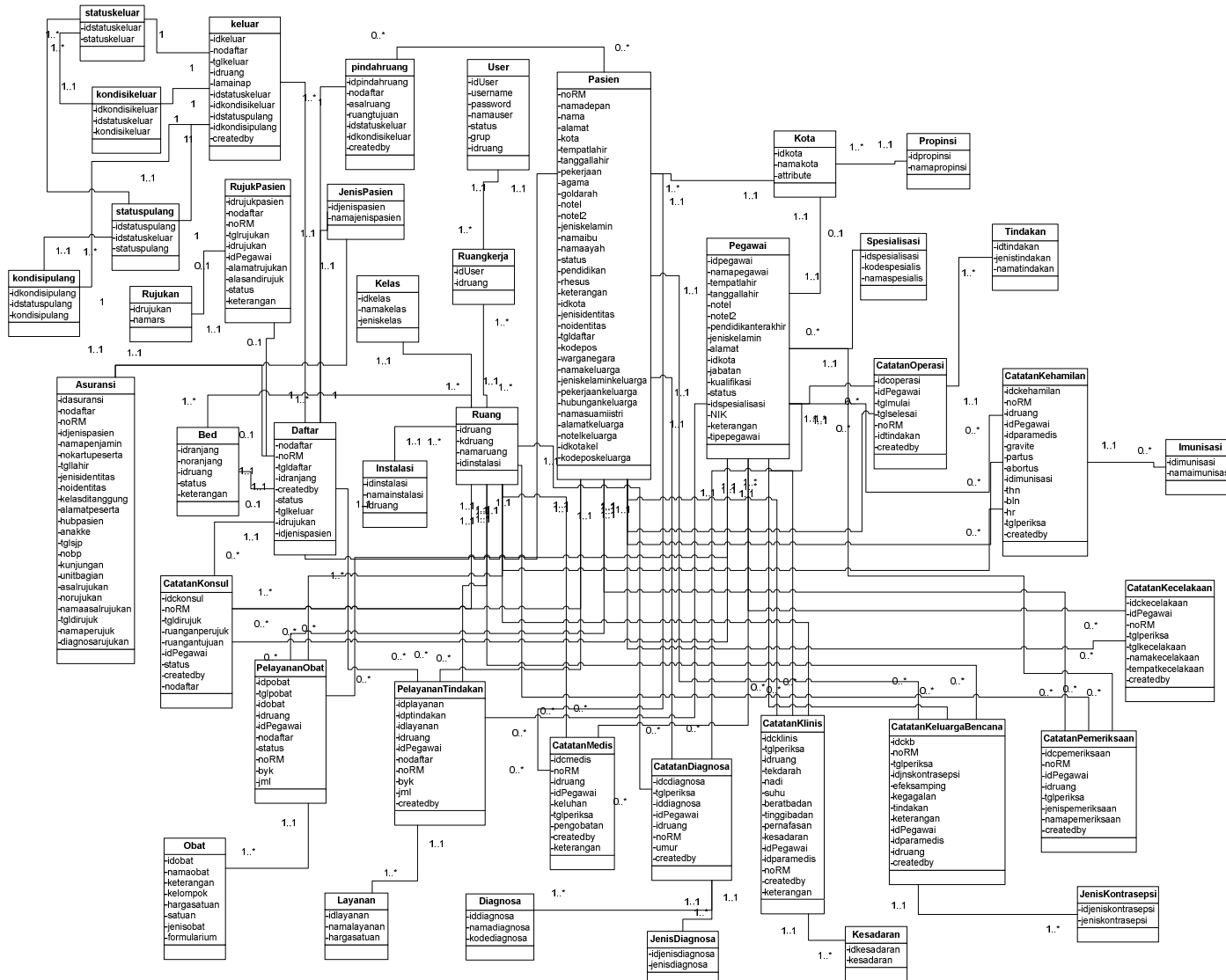
#### 3.2 Analisa Perilaku Sistem

##### 1) Use case Diagram



**Gambar 3.2** Diagram Use Case

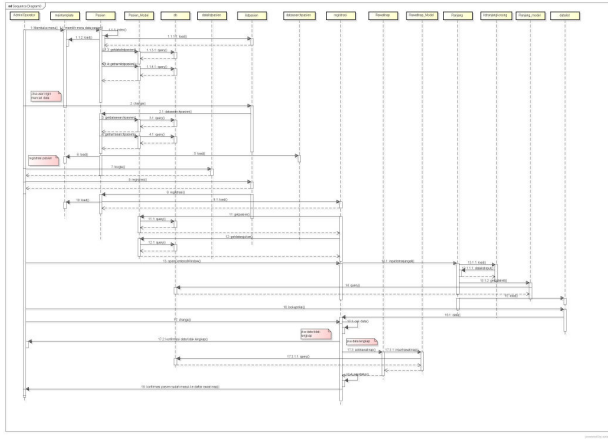
2) Entity Relationship Diagram



Gambar 3.3 Entity Relationship Diagram



#### 4) Sequence Diagram



Gambar 3.5 Sequence Diagram tambah pendaftaran rawat inap

#### 4. IMPLEMENTASI DAN PENGUJIAN

Sistem informasi rekam medis disimpan pada sebuah server yang sudah ter-install Apache HTTP Server 2.2.4 sebagai *web server*, MySQL sebagai DBMS, dan framework PHP Code Igniter. Sistem informasi diletakkan pada *document root web server* (htdocs). Sistem dibuat dengan sistem tersentralisasi, yaitu dengan meletakkan aplikasi dan basis data pada sebuah server. *Client* mengakses sistem informasi menggunakan *web browser* dengan mengakses URL misal <http://192.168.0.1/sirm>.

##### 4.1 Perangkat Pengembangan Dan Implementasi

Pembangunan dan pengimplementasian pembuatan sistem informasi ini membutuhkan spesifikasi perangkat keras, perangkat lunak dan operator (*user*). Adapun perangkat yang digunakan dalam pembangunan dan pengimplementasian perangkat lunak ini adalah sebagai berikut :

Pengelolaan perangkat lunak ini dibangun untuk diimplementasikan pada komputer personal yang menggunakan sistem operasi windows berbasis web.

Pembangunan aplikasi ini menggunakan perangkat keras yang minimal mempunyai spesifikasi sebagai berikut :

1. Personal komputer
2. Perangkat masukan yaitu keyboard, mouse.
3. Perangkat keluaran yaitu monitor dengan resolusi minimal 800 x 600 16 bit, printer dan media penyimpanan sekunder berupa hardisk yang ruang tersisanya 5 Gigabyte.
4. Sistem Operasi

Pengimplementasian aplikasi pengolahan perangkat lunak ini menggunakan perangkat keras dengan spesifikasi sebagai berikut :

- a. Komputer yang memiliki fungsi pemroses utama Core i3, dengan kapasitas memory utama 2 GB.
- b. Perangkat masukan yang dipakai yaitu *keyboard* dan *mouse*.
- c. Perangkat keluaran monitor,
- d. Sistem Operasi Windows 7

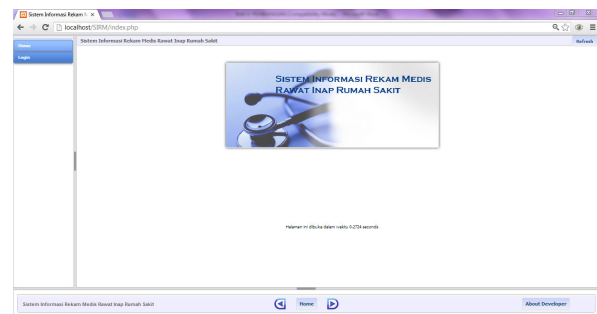
##### 4.2 Lingkungan Perangkat Lunak

Pengembangan dan pengimplementasian perangkat lunak ini membutuhkan suatu perangkat lunak yang memadai. Hal ini memerlukan perangkat lunak sebagai berikut :

1. HTML digunakan sebagai implementasi antar muka antara pengguna dengan *database*.
2. MySQL, sebagai *server* basis data yang digunakan untuk menyimpan data.
3. PHP, bahasa *scripting* pengontrol logika aplikasi web.
4. Ajax, *Asynchronous Javascript Xml* untuk pengiriman data melalui *backend* sehingga tidak memerlukan *refresh page*.
5. Jquery, untuk membuat animasi antarmuka dan antarmuka fungsional.
6. *Framework* Code Igniter, digunakan untuk memudahkan kerangka kerja dalam pembuatan *coding*.
7. SQL, merupakan bahasa *scripting* basis data.

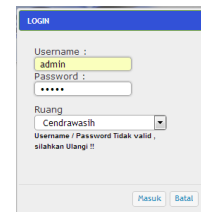
#### 4.3 Halaman awal

Halaman awal ini adalah halaman yang paling awal terbuka ketika membuka sistem informasi. *User* yang disini harus *login* terlebih dahulu dengan memasukan *username* dan *password* dan ruang. Hal ini dilakukan untuk menjaga keamanan data, yaitu agar aplikasi hanya dapat diakses oleh *user* yang berwenang di ruang kerja yang telah ditentukan, *user* di ruangan lain tidak bisa akses data di ruangan lain, kecuali admin. Ketika URL di *enter* browser kemudian *routing* akan menangani *default\_route* yang tersimpan di file *route* untuk dipanggil. *Default\_route* ini yang kemudian menjadi controller awal yang menangani proses antara model, view dan *resources* lainnya. Halaman login ini dapat dilihat pada gambar berikut.



Gambar 4.1 Halaman awal

Disini ada menu 'Home' dan 'Login'. Menu login memunculkan dialog agar *user* memasukkan *username* dan *password* dan memilih ruang. Jika *username* dan *password* yang dimasukan *valid* maka akan menuju halaman utama, jika tidak *valid* maka akan muncul peringatan sebagai berikut.



Gambar 4.2 Login

Jika semua valid maka akan muncul menu utama. Menu utama ini dibagi menjadi beberapa berdasarkan jenis *user* dan ruang. Tampilan admin akan berbeda dengan tampilan jenis *user* operator. Admin memiliki akses ke semua ruang sedangkan operator hanya bisa melihat beberapa menu

berdasarkan ruang tempat dia bekerja. Fungsi ini menggunakan html di view *template* yang di *load* oleh *controller* utama ketika *user* mengetikkan URL sebagai berikut,

```
<div id="login-dialog" title="LOGIN"
style=" height:600px;
display:none;font-family: Lucida
Grande, Verdana, Sans-serif;
font-size: 14px; ">
<br>
<form style="margin-left: 10px;"
name="form-login" id="form-login">
    Username :<br> <input type="text"
name="username" id="username"/> <br>
    Password :<br> <input
type="password" name="password"
id="password"/> <br></p>
<tr ><td align="right"
>Ruang</td><td><select
style="height:24px;" class="idruang"
id="idruang" >
    <?php
    $hasil = $ruang;

for($i=0;$i<count($hasil);$i++){
    >
<option value="<?php echo
$hasil[$i]['idruang']; ?>"><?php echo
$hasil[$i]['namaruang']; ?></option>
    <?php } ?>
</select> </td></tr>
<tr ><td align="right" ><label
id="strMsgLog"/ ></td></tr>
</form>
</div>
```

Tag `<div></div>` tersebut diberi css style `display:none` sehingga tidak ditampilkan ketika halaman login di *load browser*. Tag tersebut mempunyai pemrosesan php yang memunculkan list data ruangan yang akan diakses. \$ruang pada kode tersebut dihasilkan dari *controller* utama yang memanggil fungsi di model *Ruang\_model* yang berisi *query database* untuk mengambil semua data nama ruang dan idruang tersebut yang kemudian dihasilkan *array*. *Array* tersebut kemudian ditampilkan sebagai *dropdown* dalam tag `<select></select>`, data ruang dimasukkan dalam tiap `<option></option>` dengan value idruang masing-masing yang unik dan tidak sama sebagai parameter.

Pemanggilan html form dialog untuk login dilakukan dengan menggunakan ajax sehingga tidak perlu *refresh* halaman. Fungsi ini dipanggil dengan html tag `<input>` dengan tipe 'button' yang mempunyai atribut `onclick=showLoginForm()`; yang memanggil fungsi javascript untuk menampilkan dialog. Dialog ini kemudian memanggil fungsi validasi sehingga input *user* tidak ada yang kosong. Jika validasi berhasil maka dipanggil fungsi `login()`; yang memanggil fungsi di *controller* utama untuk proses login,

```
$.ajax({
    type: "POST",
    async : false,
    url:"<?php echo
base_url();?>index.php?/user/user/ProsesLogin/" +username+ "/" +password+ "/" +ruang,
```

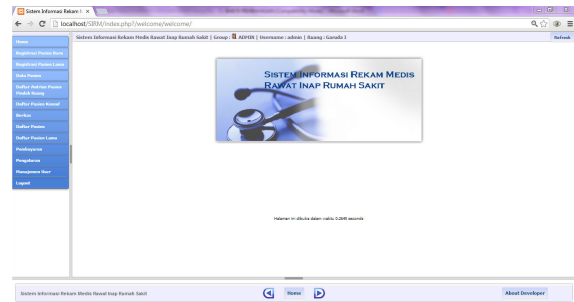
```
data:          $("#form-
login").serialize(),
    success: function(stts)
    {
        // kode jika sukses atau
        tidak //
    });
```

Fungsi ini menggunakan parameter *username*, *password* dan idruang yang didapat dari atribut value `<select>` dengan id="idruang". Fungsi ini akan menghasilkan status html yang akan ditampilkan di `<label id="strMsgLog"/>` dengan fungsi ajax `$("#strMsgLog").html(status);`

#### 4.4 Halaman Utama

Halaman utama ini merupakan halaman yang memuat menu-menu yang dapat di akses oleh *user* sesuai dengan status *user*nya. Admin memiliki menu terbanyak karena semua fungsi dapat dilakukan. Sedangkan untuk *user* operator biasa hanya di perbolehkan akses data sesuai dengan bidangnya dan ruangnya.

Halaman utama untuk admin memiliki beberapa menu seperti registrasi pasien baru, registrasi pasien lama, data pasien, daftar antrian pasien pindah ruang, daftar pasien konsul, berkas, daftar pasien, daftar pasien lama, pembayaran, pengaturan, manajemen *user* dan *logout*.



Gambar 4.3 Halaman Utama Admin

Bagian atas sistem informasi ini berupa *status bar* yang menunjukkan *user* yang sedang login, grup *user* tersebut dan ruangan tempat dia akses. Tampilan ini dihasilkan oleh *library* 'session' yang dipanggil di *controller* utama dengan syntax

```
$this->load->library('session');
Penggunaan library ini yaitu dengan
$this->session->set_userdata('username', $hasil[$i]['username'] );
$this->session->set_userdata('group', $hasil[$i]['grup'] );
$this->session->set_userdata('userid', $hasil[$i]['idUser']);
$this->session->set_userdata('ruang', $hasil[$i]['namaruang'] );
$this->session->set_userdata('idruang', $hasil[$i]['idruang'] );
```

*Array* \$hasil dihasilkan dari *query database* yang dipanggil di *controller user*

```
$valUserPass=$this->User->getUser($username, $password, '');
$valUserPass2=$this->User->getUser($username, $password, $ruang);
```

Fungsi-fungsi tersebut diletakkan di User\_Model yang digunakan untuk validasi apakah input data login tersebut menghasilkan nilai '!=0' yang berarti data tersebut ada di dalam *database*, jika menghasilkan nilai 0 maka data tersebut tidak ada yang cocok yang membuat *user* tidak bisa login.

Gambar 4.4 Status bar sistem informasi

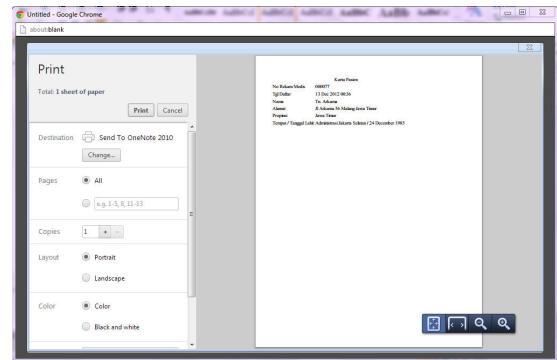
#### 4.5 Registrasi Pasien Baru

Menu registrasi pasien baru untuk mendaftarkan pasien baru yang belum pernah berkunjung sebelumnya, registrasi pasien lama dilakukan untuk pasien yang pernah berkunjung. Isi menu tersebut adalah sebagai berikut,

Gambar 4.5 Form input *profile* pasien

Form tersebut untuk memasukkan data pribadi pasien dengan data keluarga dan alamat keluarga pasien. Setelah data dimasukkan kemudian muncul tampilan data detail yang mempunyai tombol cetak kartu. Tampilan print cetak kartu tersebut sebagai berikut,

Gambar 4.6 Pasien baru ditambahkan



Gambar 4.7 Proses cetak kartu

Setelah pemrosesan cetak kartu selesai, pasien telah masuk ke daftar data pasien rumah sakit dan mendapatkan nomor rekam medis.

#### 4.6 Registrasi Pasien Lama

Registrasi pasien lama isi menunya sebagai berikut,

Gambar 4.8 Registrasi pasien rawat inap lama

Ketika *user* tekan simpan kemudian ajax akan me *request* URI yang terdapat pada controller Pasien.

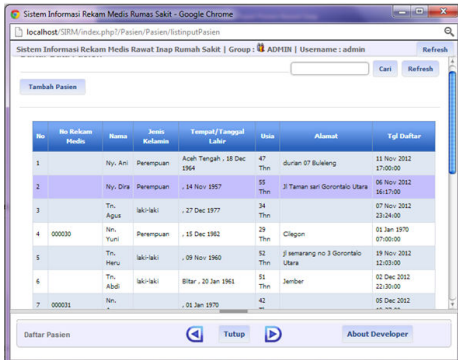
```
$.ajax({
    type: "POST",
    async : false,
    url: "=php echo base_url();?&gt;index.php?/Pasien/Pasien/addPasien/0/0",
    data:
    $('#formInputPasien').serialize(),
    success: function() {
        Jika berhasil maka akan muncul alert,
        statinput=document.getElementById("txtnamapasien").value;
        alert(" Pasien dengan nama : \n"+statinput+" telah disimpan! ");
    }
});</pre

```

```
Controller pasien menangani model-model yang digunakan sebagai pengontrol database dan class yang digunakan untuk membaca informasi di dalam database,
$this->load->model('Pasien_model', 'Pasien');
$this->load->model('Wilayah/Wilayah_model', 'Wilayah');
$this->load->model('Pendukung/Pendukung_model', 'Pendukung');
```

Pemasukan pasien dilakukan dengan tekan cari terlebih dahulu maka akan muncul window daftar pasien.





Gambar 4.9 Pencarian pasien

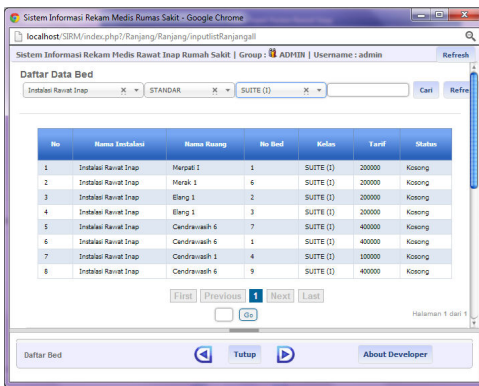
Pencarian pasien pada window tersebut dilakukan dengan ajax sehingga *user* tidak perlu *refresh window*. Kemudian dalam penginputan pasien tinggal *hover* baris sampai baris berwarna ungu muda dan menekannya. Pemilihan pasien selesai. Pemasukkan bed juga dilakukan hal yang sama. Window yang muncul adalah semua daftar bed yang kosong. Pemfilteran data bed dilakukan dengan pemilihan *dropdown* dan data akan dicari secara *live search*.

Fungsi *live search* merupakan kombinasi dari javascript dan ajax,

```

$("#slInstalasi").change(function() { //fungsi
    untuk mengenali dropdown dalam keadaan diubah
    $.ajax({
        type: "POST",
        url : "<?php echo
base_url();?>index.php?/Ranjang/Ranjang/datas
earchinput/1/0/", data:
$("#formcari").serialize(),
        success: function(msg) {
            $("#containerpag").ajaxComplete(function(event,
            request, settings) {
                $("#containerpag").html(msg);
            });
        }
    });
});

```



Gambar 4.10 Hasil pencarian secara *live search* dengan parameter-parameter

Fungsi *live search* dihasilkan dengan jquery

```

$(".cari").keyup(function()
{
    var Jeniscari = $(".slJeniscari").val();
    // variable yang dihasilkan dari input
    id=slJenis cari di form
    var searchbox = $(this).val(); //
    variable dari input yang diketik

```

Ada dua kemungkinan yaitu input dalam keadaan kosong dan keadaan sedang diketik, jika dalam keadaan kosong maka dipanggil fungsi `loadData(1, '', '')` ;

yang menghasilkan semua data tanpa ada filter pencarian, perbedaannya hanyalah di parameter pencariannya saja.

```

if(searchbox=='')
{
    loadData(1, '', ''); // memanggil fungsi
    loadData karena input null
}

```

Jika input dalam keadaan sedang diketik maka fungsi *keyup* akan memanggil ajax dengan URI yang terdapat pada *controller* pasien yang Melakukan pencarian dengan parameter yang didefinisikan di *controller* sebagai berikut,

```

$params cari=$this->input->post('cari');
Dimana 'cari' didapatkan dari form html. Kemudian data
ditampilkan di tag <div> dengan id= containerpag.
else
{
$.ajax
({
    type: "POST",
    url: "<?php echo
base_url();?>index.php?/Ranjang/Ranjang/datas
earchinput/1/0/",
    data: $("#formcari").serialize(),
    success: function(msg)
    {
        $("#containerpag").ajaxComplete(function(event,
        request, settings)
        {
            $("#containerpag").html(msg);
        });
    }
});
}return false;
});

```

Hasil pencarian bed selanjutnya untuk memilih data, *hover* baris data dan tekan maka akan terinput otomatis ke form registrasi. Ketika data dipilih maka javascript akan melakukan *passing variable* dari window daftar pasien ke window opener

```

opener.document.forms['formInputRawatInap'].t
xtranjang.value=Kd_ranjang;
opener.document.forms['formInputRawatInap'].H
iddenRanjang.value=id_ranjang;

```



Gambar 4.11 Form input registrasi pasien lama

*Dropdown* rujukan dihasilkan oleh controller yang memanggil fungsi di model rujukan,

```

$data['option_rujukan'] = $this->Rujukan-
>GetDatarujukan();

```

#### 4.7 Data Pasien

Pasien yang sudah dimasukkan kemudian ditampilkan pada menu data pasien,

No	No Rekam Medis	Nama	Jenis Kelamin	Tanggal / Tanggal Lahir	Ura	Alamat	Tgl Daftar
1	00000	No. Yuni	Perempuan	Bireuen, 15 Dec 1982	29 Thn	3 Kalamas 234 Cilong	01 Jan 1970 07:00:00
2	00001	No. Anne	Perempuan	Administrasi Jakarta Pusat, 01 Jan 1970	42 Thn	3 Kepulauan 567 Banyumas	05 Dec 2012 18:27:00
3	00006	Trn. Kaka	laki-laki	Asah Utara, 05 Jan 1970	42 Thn	Administrasi Jakarta Selatan	11 Dec 2012 11:00:00
4	00007	Trn. Gusda	laki-laki	Bengkulu Utara, 17 Dec 1975	36 Thn	Jalan Kuluhanan 233 Lendak	12 Dec 2012 08:00:00
5	00007	Trn. Sudjo	laki-laki	Bitar, 12 Dec 1983	49 Thn	Cempalrang no 2 Administrasi Jakarta Selatan	13 Dec 2012 00:27:00
6	00007	Trn. Ariana	Perempuan	Asah Utara, 05 Jan 1970	42 Thn	3 Sekelu 567 Mangaka	13 Dec 2012 00:29:00
7	00076	Trn. Anjani	laki-laki	Asah Utara, 05 Dec 1982	30 Thn	3 Bengkulu Jaya 987 Padangbaru Sub	13 Dec 2012 00:00:00
8	00076	Trn. Dariana	laki-laki	Asah Besar, 05 Dec 1980	32 Thn	3 Jemberan 344 Mangrove	13 Dec 2012 00:30:00
9	00077	Trn. Ardana	laki-laki	Administrasi Jakarta Selatan, 14 Dec 1985	26 Thn	3 Ardana 55 Haling	13 Dec 2012 00:30:00
10	00078	Trn. Kana	laki-laki	Asah Tengah, 12 Dec 1985	27 Thn	3 Nenda Administrasi Jakarta Selatan	13 Dec 2012 00:40:00
11	00079	Ny. Keandra	Perempuan	Alor, 20 Dec 1961	50 Thn	3 Jakarta pembangunan 56 Administrasi Jakarta Selatan	13 Dec 2012 00:40:00
12	00001	Trn. Anif indayat	laki-laki	Asah Timur, 20 Apr 1980	32 Thn	Jalan Candravathi II No 3 Nunukan	09 Nov 2012 00:00:00
13	00002	Trn. Hafis nasrati	laki-laki	Barho Timur, 12 Jan 1978	34 Thn	Jalan Hajar II No 4 Kapukam Anambas	24 Oct 2012 00:00:00
14	00003	Trn. Wahyu Kumoro	laki-laki	Bolang Hongorobov Selatan, 11 Nov 1977	35 Thn	Jalan Kalamo No 3 Gunung Kidul	07 Jan 2012 00:00:00
15	00004	Nn. Yutha Pura	Perempuan	Bogomoro, 11 Nov 1989	23 Thn	Jalan Surten Raya No 11 Supori	06 Nov 2012 00:00:00

Gambar 4.12 Data Pasien

Halaman ini mempunyai tombol data detail untuk melihat data pribadi lengkap pasien, registrasi jika pasien tersebut akan dimasukkan ke rawat inap, dan riwayat pemeriksaan. Jika pasien setelah dicari tidak ada maka petugas tinggal menekan tambah pasien dan input pasien baru dari awal.

Gambar 4.13 Data Detail Pasien

Gambar 4.14 Registrasi pasien

Jika pasien dipilih dan ditekan riwayat pemeriksaan maka akan muncul data riwayat pemeriksaan pasien dari sejak awal sampai akhir pasien terakhir dirawat/meninggal.

Gambar 4.15 Riwayat pemeriksaan pasien

Menu ini menampilkan data-data yang dapat dilihat pada gambar.

#### 4.8 Manajemen Data Master

Pemasukan, pengeditan dan penghapusan pada semua manajemen data di aplikasi ini menggunakan ajax sehingga prosesnya cepat tanpa ada refresh semua bagian halaman page. Metode ini hampir sama dengan metode ajax pada pencarian data tetapi bedanya hanya pada fungsi yang dipanggil. Data-data master untuk proses pengolahan menjadi informasi ini dikelompokkan menjadi beberapa jenis.

1. Informasi Kamar
2. Data kelas
3. Data master instalasi
4. Data master ruang
5. Data master layanan
6. Data master pegawai
7. Data master obat dan alat kesehatan
8. Data master tindakan operasi
9. Data master diagnosa
10. Data master tempat rujukan
11. Data master spesialisasi
12. Data master pendukung
13. Data master wilayah
14. Data master imunisasi

Contoh data master adalah sebagai berikut. Data master diagnosa ini dibuat dengan standar internasional ICD 10.

No	Kode Diagnosa	Nama Diagnosa	Edit	Delete
1	A00	Kolera		
2	A01.0	Demam Tifoid		
3	A01.4	Demam paratifoid, tidak spesifik		
4	A02	Infeksi salmonella lainnya		
5	A03.9	Shigellosis, Disentri Basiler tidak Spesifik		
6	A04.9	Infeksi usus bakteri tidak spesifik		
7	A05.9	Keracunan makanan karena bakteri lainnya		
8	A06	Amubiasis, Disentri Amuba		
9	A06.5	Infeksi fusa dan Infeksi usus tertentu lainnya		
10	A09	Diare dan Gastroenteritis tidak A00 - B08		
11	A15-A1	TUBERKULOSIS		
12	A15.0	TB BTA (+) tanpa biakan		
13	A15.9	TB selain infeksi bakteriologi atau histologi		
14	A16.0	Tuberkulosis tanpa klinis		
15	A17.0	Meningitis Tuberkulosa		

Gambar 4.16 Data master diagnosa

#### 4.9 Manajemen pasien

Paginasi didesain untuk pengolahan data yang sangat banyak. Tombol-tombol tersebut akan menampilkan data per halaman dengan ajax.

##### 1. Daftar pasien

Gambar 4.17 Daftar pasien

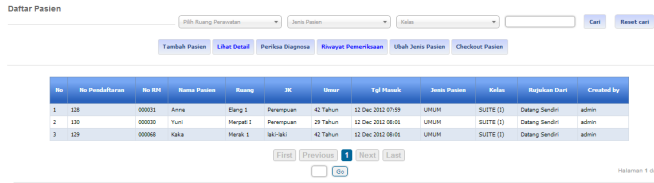
##### 2. Daftar pasien lama

Gambar 4.18 Daftar pasien lama

### 3. Daftar antrian pasien pindah ruang

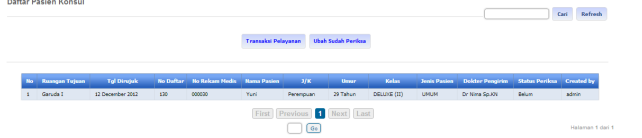


**Gambar 4.19** Daftar antrian pasien pindah ruang



**Gambar 4.20** Pasien telah pindah

### 4. Daftar pasien konsul



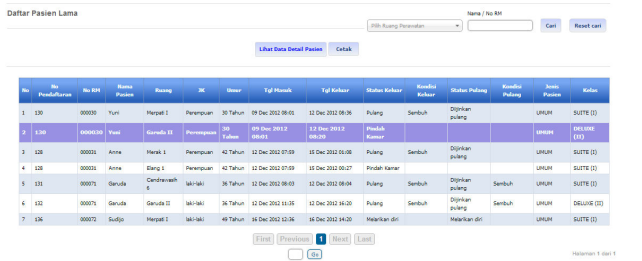
**Gambar 4.21** Daftar pasien konsul

**Form Input Tanggal Keluar**

Pasien: **Nn. Yuni**  
 Ruang: **Herpati I**  
 Bed: **1**  
 Tgl Masuk: **1 January 1970 07:00**  
 Tgl Keluar: **12-12-2012 8:35**

Status Keluar: **Pulang**  
 Kondisi Keluar: **Sembuh**  
 Status Pulang: **Dijinkan pulang**  
 Kondisi Pulang: **Simpan**

**Gambar 4.22** Checkout pasien



**Gambar 4.23** Pasien telah masuk ke daftar pasien lama

### 4.10 Manajemen Data Rekam Medis

Manajemen data rekam medis dikelompokkan menjadi beberapa jenis diantaranya,

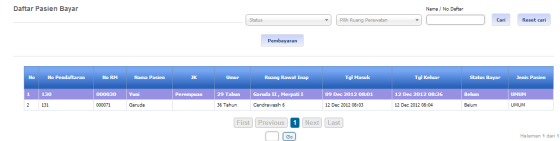
1. Data layanan
2. Data obat dan alat kesehatan
3. Data catatan klinis
4. Data catatan kehamilan
5. Data catatan keluarga bencana
6. Data riwayat diagnosa
7. Data catatan medis
8. Data riwayat operasi
9. Data riwayat konsul
10. Data riwayat kecelakaan
11. Data riwayat pemeriksaan

Contoh data rekam medis obat dan alat kesehatan seperti berikut ini,



**Gambar 4.24** Data rekam medis pelayanan obat

### 4.11 Pembayaran



**Gambar 4.25** Daftar pasien yang akan melakukan pembayaran

**Rincian Pembayaran**

Pasien: **Anne**  
 No RM: **67**  
 No Pendaftaran: **128**  
 Ranjang: **4**

**Daftar Rincian Pengeluaran Ruangan**

No	Ruang Pelayanan	Kabin	Tarif	Tgl Masuk	Tgl Keluar	Lama Inap	Jumlah
1	Makel I	SUITE (I)	200.000	13 Dec 2012 07:59	15 Dec 2012 09:08	2	Rp. 400.000
2	Elang I	SUITE (I)	200.000	12 Dec 2012 07:59	15 Dec 2012 09:07	2	Rp. 400.000
<b>Total</b>							<b>Rp. 800.000</b>

**Daftar Rincian Pelayanan Tindakan**

No	Tgl Pelayanan	Jenis Pelayanan	Nama Pelayanan	Ruang Pelayanan	Kabin	Bkyl	Tarif	Jumlah
1	13 December 2012	Pelayanan Ibu dan Anak	Tindakan Kacil	Gerunda I	DELUXE (II)	6	Rp. 36.000	Rp. 216.000
2	15 December 2012	Pelayanan Ibu dan Anak	Tindakan Khusus	Bank Darah	DELUXE (II)	5	Rp. 100.000	Rp. 500.000
3	15 December 2012	Pelayanan Ibu dan Anak	Tindakan Sedang	Bank Darah	DELUXE (II)	6	Rp. 60.000	Rp. 360.000
<b>Total</b>								<b>Rp. 1.076.000</b>

**Daftar Rincian Pemasukan Obat dan Alat Kesehatan**

No	Tgl Pelayanan	Detail Jenis Barang	Nama Barang	Ruang Pelayanan	Kabin	Tarif Satuan	Bkyl	Jumlah
1	15 December 2012	Generik	amoxiclin_500_mg	Bank Darah	DELUXE (II)	Rp. 3.000	1	Rp. 3.000
2	15 December 2012	Generik	alprazolm_100_mg	Bank Darah	DELUXE (II)	Rp. 80.000	6	Rp. 480.000
3	15 December 2012	generik	ctm_4_mg	Bank Darah	DELUXE (II)	Rp. 0	7	Rp. 0
<b>Total</b>								<b>Rp. 483.000</b>

**Lakukan Pembayaran**

Total Biaya Ruangan	Rp. 800.000
Total Biaya Pelayanan Tindakan	Rp. 1.076.000
Total Biaya Pelayanan Obat dan Alat Kesehatan	Rp. 483.000
<b>Total Biaya</b>	<b>Rp. 2.359.000</b>

**Gambar 4.26** Rincian pembayaran

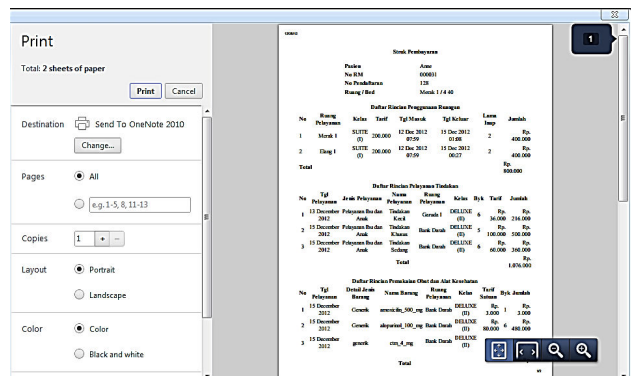
**Form Input Pembayaran**

Pasien: **Anne**  
 No RM: **67**  
 No Pendaftaran: **128**  
 Ruang / Bed: **Herpati I / 4 40**

Total Biaya Ruangan: **Rp. 800.000**  
 Total Biaya Pelayanan Tindakan: **Rp. 1.076.000**  
 Total Biaya Pelayanan Obat dan Alat Kesehatan: **Rp. 483.000**  
 Total Biaya: **Rp. 2.359.000**  
 Tgl Bayar: **26-12-2012 09:19**  
 Bayar: **Rp. 2400000**  
 Kembali: **Rp. 410000.00**

**Proses**

**Gambar 4.27** Proses pembayaran



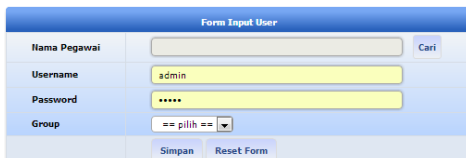
**Gambar 4.28** Print struk pembayaran

#### 4.12 Manajemen user

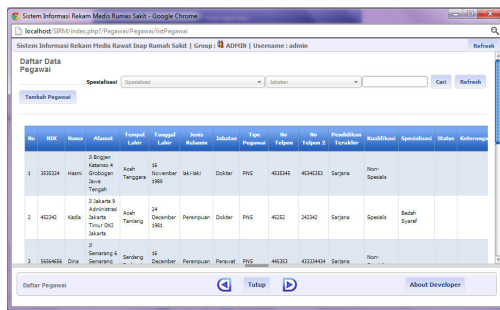


Gambar 4. 29 Submenu manajemen user

Admin bertugas untuk melakukan input pengguna yang bisa masuk ke sistem. Tugas ini terdapat dalam menu manajemen user yang terdiri dari input user yang dapat memasukkan pegawai ke dalam sistem dengan username dan password awalnya yaitu NIK masing-masing pegawai. Pemasukkan pengguna tersebut menggunakan daftar pegawai yang telah dimasukkan di berkas data pegawai. Daftar tersebut akan tampil tanpa merefresh page tersebut setelah tekan cari.

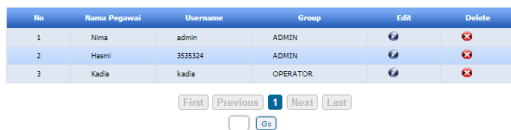


Gambar 4. 30 Form input pengguna sistem



Gambar 4. 31 Daftar pegawai yang ada di rumah sakit yang akan diinputkan ke dalam daftar user.

Setelah memasukkan data pegawai selesai maka username dan passwordnya akan otomatis terisi NIK dari pegawai tersebut. Namun, username dan passwordnya masih bisa diganti oleh admin. User tersebut ditentukan jenis usernya admin atau operator. Setelah proses memasukkan data selesai admin tinggal menekan tombol simpan. User baru tersebut akan ditampilkan di daftar user.



Gambar 4. 32 Manajemen daftar ruang kerja user

Pengguna sistem bisa bekerja di banyak ruang sehingga diperlukan daftar khusus yang berisi daftar pengguna dan ruang kerjanya.



Gambar 4. 33 Data ruang kerja user

Fungsi untuk menambah atau mengurangi ruang kerja user tinggal menekan edit maka akan tampil daftar ruang kerja user tersebut.



Gambar 4. 34 Manajemen data ruang kerja tiap user

Ruang kerja tersebut bisa dihapus dan proses ini menggunakan ajax sehingga cepat dalam antarmukanya. Selain itu penambahan ruang kerja juga javascript untuk memunculkan form tambah ruang kerja dengan menekan tombol tambah ruang kerja.

Penambahan ruang kerja juga bisa menggunakan menu input ruang kerja user yang menampilkan form berikut. Instalasi dan ruang merupakan chain dropdown yang menghasilkan masing-masing ruang yang ada di database sesuai instalasi terkait.

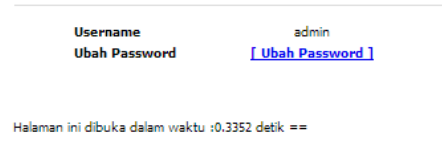


Gambar 4. 35 Form input ruang kerja user

#### 4.13 Pengaturan

Menu pengaturan digunakan untuk mengubah password yang awalnya berupa NIK yang diberi oleh admin. Password ini harus diubah demi keamanan sistem.

##### Pengaturan



Gambar 4. 36 Isi menu pengaturan

Parameter dalam fungsi ini menggunakan password lama yang kemudian dienkripsi md5 dan dicocokkan dengan database. Jika cocok maka password bisa diganti.



Gambar 4. 37 Form ubah password

### 5. PENUTUP

#### 5.1 Kesimpulan

1. Sistem informasi ini melakukan manajemen pasien di rumah sakit mulai dari proses registrasi, pindah ruang, konsul, input data rekam medis sampai pasien keluar dan melakukan pembayaran. Sistem informasi ini juga merekam data medis pasien dari awal pasien masuk ke rumah sakit sampai meninggal.

2. Sistem informasi ini merupakan suatu aplikasi yang digunakan untuk membantu pengelolaan data rekam medis. Adapun data diagnosa pada aplikasi ini telah mengikuti standar internasional ICD 10. Penggunaan aplikasi akan meningkatkan pengelolaan data rekam medis menjadi lebih mudah dan efisien dilakukan, aman karena data tersimpan dalam *database*, tidak cepat rusak, mempermudah dalam pencarian data, serta lebih terstruktur dibandingkan dengan pengelolaan secara manual.
3. Fungsi aplikasi ini adalah sebagai berikut,
  - a) Sebagai alat bantu pendaftaran perawatan rekam medis dengan menyediakan *database* perawatan pasien secara lengkap sesuai dengan standar jenis pertolongan medik yang dibutuhkan.
  - b) Sebagai alat bantu pendaftaran perawatan dengan melakukan pengelompokan pasien berdasarkan penjamin pembayaran, kelas dan ruangan dimana pencarian dengan fitur *live search* berdasarkan parameter kelompok tersebut dan kata pencarian.
  - c) Sebagai alat bantu dalam proses pembayaran perawatan rekam medis.
  - d) Sebagai alat bantu untuk menyimpan hasil penanganan rekam medis dengan menyediakan *database* hasil pemeriksaan pasien secara lengkap sesuai standar jenis pertolongan medik yang dibutuhkan.
  - e) Sebagai alat bantu jika menginginkan konsultasi dokter spesialis.
  - f) Sebagai alat bantu untuk konsul ke laboratorium dan ruangan konsul lain.

## 5.2 Saran

1. Sistem informasi ini dapat dikembangkan menjadi sistem yang lebih besar diintegrasikan dengan modul-modul yang lain.
2. Sistem informasi ini juga dapat dikembangkan lebih jauh lagi agar dapat menghasilkan informasi-informasi sehingga aplikasi ini layak disebut sistem informasi ditinjau dari segi fungsionalitas keluaran dan keamanannya.

Dari hasil tugas akhir ini, penulis menyarankan sebaiknya masing-masing ruang disediakan satu unit komputer untuk menginput data pemeriksaan pasien, sehingga proses manajemen data lebih efisien.

## DAFTAR PUSTAKA

- [ 1 ] Ariasari, Fany. 2010. *Panduan Praktis Bikin Blog dengan Wordpress*. Jakarta : Media Kita.
- [ 2 ] Craig , R. D. dan S. P. Jaskiel. 2002 . *Systematic Software Testing*. Norwood MA: Artech House Publis.
- [ 3 ] Gordon, David. B.. 1985. *Sistem Informasi Manajemen*, Jakarta. PT. Pustaka Binaman Pressindo, Jilid I
- [ 4 ] Hatta , Gemala R., 2008, *Petunjuk Teknis Penyelenggaraan Rekam Medis Rumah Sakit*. Penerbit Universitas Indonesia
- [ 5 ] [http://dinkes-sulsul.go.id/new/images/pdf/pdf\\_rs/permenkes%20no](http://dinkes-sulsul.go.id/new/images/pdf/pdf_rs/permenkes%20no)

mor%20269%20tahun%202008%20-rekam%20medik.pdf, diakses tanggal 10 Agustus 2012

- [ 6 ] Huffman, Edna K, 1994 . *Health Information Management*.
- [ 7 ] Jogiyanto, H.M. 2005. *Sistem Teknologi Informasi Edisi 2*. Yogyakarta : Andi.
- [ 8 ] Kadir, Abdul, *Mudah Mempelajari Database MySQL*, Penerbit Andi, Yogyakarta, 2010.
- [ 9 ] Kristanto, Andri. 2004. *Rekayasa Perangkat Lunak (KonsepDasar)*. Bandung; Penerbit Gava Media.
- [ 10 ] Ladjamudin B, Al Bahra Bin. 2006. *Rekayasa Perangkat Lunak*. Yogyakarta : Graha Ilmu.
- [ 11 ] Lee Babin *Programmer, 2007, Beginning Ajax With PHP from Novice to Profesional*.
- [ 12 ] Pressman , Roger S. , Ph.D., 1997. *software Engineering : A Practioner's approach*. Mcgraw-Hill Book Co. Terjemahan bahasa indoneis oleh LN Hananingrum,2002.Yogyakarta;Penerbit Andi.
- [ 13 ] Pribadi Basuki, Awan. 2011. *Membangun Web Berbasis PHP Dengan Framework CodeIgniter*. Yogyakarta : Lokomedia.
- [ 14 ] Ramadhan, Arief. 2006. *SGS : Pemr.Web Database PHP&MySQL*. Jakarta : Elex Media Komputindo.
- [ 15 ] Sutanta, Edhy.2005. *Pengantar Teknologi Informasi*. Yogyakarta; Graha Ilmu.
- [ 16 ] T. Saputro, Wahyu. *MySQL Untuk Pemula*. Pena Media. Yogyakarta, 2005

## PENULIS



Nunik Srikandi Putri, lahir di Bandung, 3 November 1987. Menjalani pendidikannya di SDN Pangalengan III, SDN Majalengka Kulon VI, SMPN I Majalengka, SMAN 3 Bandung. Dan sekarang tengah menyelesaikan pendidikan Strata Satu di Jurusan Teknik Sistem Komputer, Universitas Diponegoro, Semarang, Indonesia Angkatan Tahun 2008.

Mengetahui/Mengesahkan  
Dosen Pembimbing I

**Adian Fathur Rochim ,ST,MT**  
**NIP 197302261998021001**

Dosen Pembimbing II

**Ike Pertiwi Windasari, ST,MT**  
**NIP 198412062010122008**