

BAB 6 IMPLEMENTASI SISTEM

Tahap implementasi yang dilakukan terdiri dari penjelasan tentang spesifikasi lingkungan pengembangan sistem, batasan-batasan implementasi, implementasi algoritmes, implementasi basis data, implementasi antarmuka pengguna dan implementasi komunikasi data.

6.1 Spesifikasi Lingkungan Implementasi

Pada sub bab ini menjelaskan mengenai spesifikasi dari perangkat keras dan perangkat lunak yang digunakan dalam membangun sistem.

6.1.1 Spesifikasi Perangkat Keras

Dalam penelitian ini penulis menggunakan sebuah komputer karena sistem yang dibangun tidak berkomunikasi dengan perangkat keras lain. Spesifikasi perangkat keras yang digunakan dalam pembuatan Sistem Informasi Rekam Medis Rumah Sakit dapat dilihat pada *Tabel 6.1*

Tabel 6.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>System model</i>	Acer Aspire 5733
<i>Processor</i>	Intel Core i3
<i>Memory</i>	500GB HDD
<i>Memory RAM</i>	4GB

6.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem informasi rekam medis rumah sakit pada klinik mutiara sehat dapat dilihat di *Tabel 6.2*

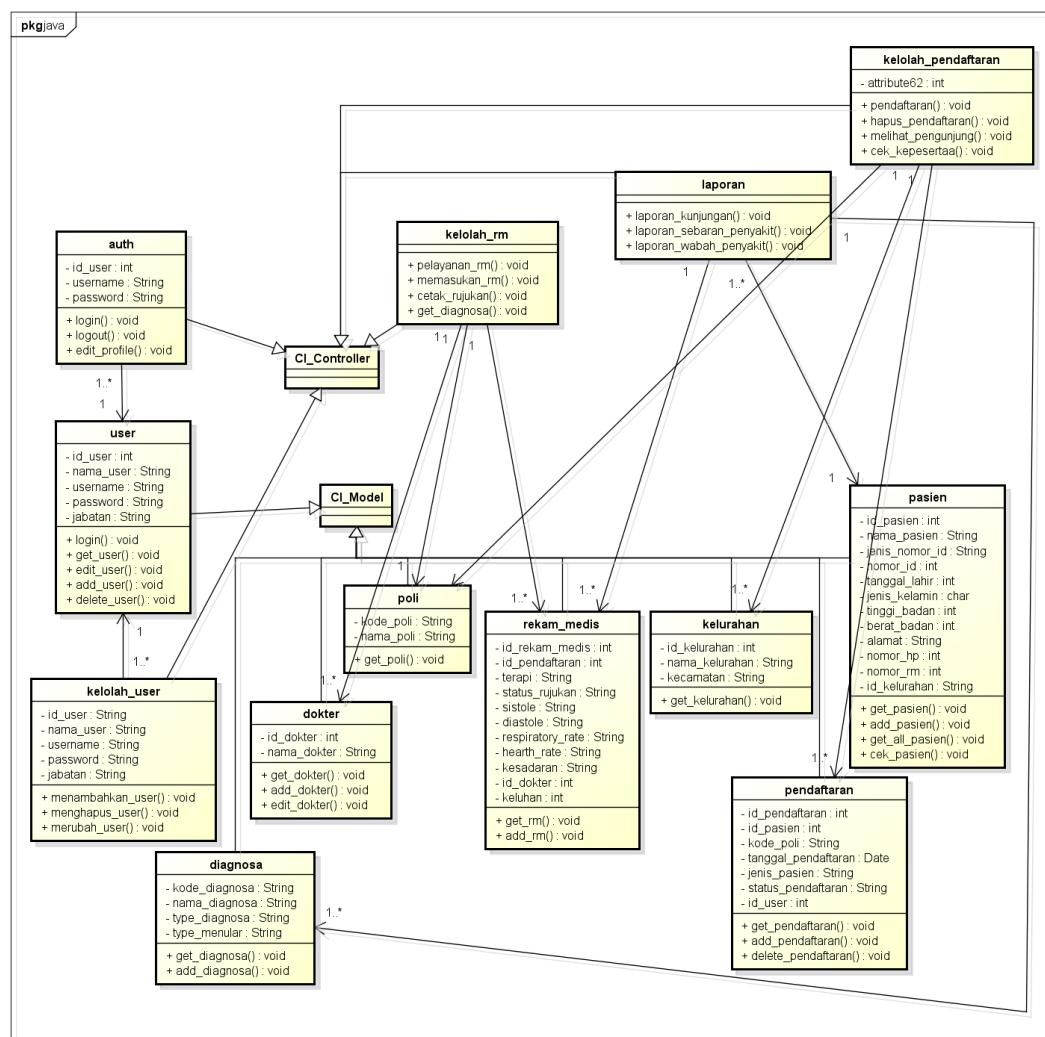
Tabel 6.2 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10 64-bit
<i>Pemrograman Language</i>	PHP
<i>Framework</i>	Codeigniter
<i>Text Editor</i>	Sublime Text
<i>Web server</i>	Xampp v.3.2.1

Tabel 6.2 Spesifikasi Perangkat Lunak(lanjutan)

Nama Komponen	Spesifikasi
Web service	NuSoap
Database	MySQL
Browser	Google Chrome

6.2 Implementasi Class



Gambar 6.1 Implementasi class diagram

Pada Gambar 6.1 digambarkan class diagram yang sudah diimplementasikan dengan menggunakan *framework CodeIgneter*. Pada implementasi *class diagram* terdapat tambahan *class CI_controller* dan *class CI_model* karena pada tahapan implementasi sistem yang dibangun menggunakan

framework CodeIgniter. Semua class yang digambarkan pada perancangan sebagai *class controller* maka *extends* dari class *CI_controller* serta *class* pada perancangan sebagai domain model maka *extends* dari class *CI_model*.

6.3 Batasan Implementasi

Batasan implementasi pada pembangunan sistem ini sebagai berikut:

1. Web Server yang digunakan pada pengembangan sistem ini adalah Apache v 1.8.1
2. Database Management System yang digunakan pada sistem ini adalah mysql ver 5.6.14
3. Versi bahasa pemrograman PHP yang digunakan dalam membangun sistem ini adalah versi 7.0.8

6.4 Implementasi Algoritme

Didalam implementasi algoritme ini hanya mencantumkan beberapa proses yang telah dilakukan perancangan sebelumnya sehingga tidak semua algoritme dicantumkan. Berikut merupakan algoritme yang telah diimplementasikan antara lain adalah algoritma cek_kepesertaan, pendaftaran_pasien, dan masukan_rekam_medis.

6.4.1 Algoritme cek kepesertaan

Fungsi algoritme cek_kepesertaan terdapat pada controller pendaftaran yang digunakan untuk melakukan cek kepesertaan terhadap pasien yang mendaftar pemeriksaan pada klinik. Pada tabel 6.4 merupakan implementasi algoritme dari fungsi cek_kepesertaan() dalam controller pendaftaran.

Tabel 6.3 Implementasi Algoritme Fungsi cek kepesertaan

1	public function cek_kepesertaan()
2	{
3	\$session_data = \$this->session->userdata('logged_in');
4	\$id_user= \$session_data['id_user'];
5	\$jabatan= \$session_data['jabatan'];
6	\$nama_user= \$session_data['nama_user'];
7	\$data['jabatan']= \$jabatan;
8	\$data['id_user']= \$id_user;
9	\$data['nama_user']= \$nama_user;
10	\$client = new nusoap_client('http://localhost/service_bpjs/webservice.php');
11	\$data['login_result']= \$client->call('auth',
12	array('nomor_authentikasi'=>'91212',
13	'kode_authentikasi'=>'0b1e50e1fd71c96bac94144cc59cff40'));
14	if (\$id_user != '') {

Tabel 6.3 Implementasi Algoritme Fungsi cek kepesertaan(lanjutan)

15	if (\$this->input->post()) {
16	\$data['jenis_identitas']= \$this->input->post('jenis_identitas');
17	\$data['nomor_identitas']= \$this->input->post('nomor_identitas');
18	\$jenis_identitas= \$this->input->post('jenis_identitas');
19	\$nomor_identitas= \$this->input->post('nomor_identitas');
20	\$data['result_bpjs']= \$client->call('cek_kepesertaan',
21	array('nomor_authentikasi'=>'91212',
22	'kode_authentikasi'=>'0ble50elfd71c96bac94144cc59cff40',
23	'jenis_identitas'=>\$jenis_identitas,
24	'nomor_identitas'=>\$nomor_identitas));
25	\$data_bpjs= \$data['result_bpjs'];
26	foreach (\$data_bpjs as \$key) {
27	\$nomor_bpjs= \$key['nomor_bpjs'];
28	}
29	if (\$nomor_bpjs == NULL) {
30	\$data['identitas_pasien']= \$this->M_pasien->get_pasien(\$jenis_identitas, \$nomor_identitas)->result();
31	}else{
32	\$data['identitas_pasien']= \$this->M_pasien->get_pasien('bpjs', \$nomor_bpjs)->result();
33	}
34	\$data['data_poli']= \$this->M_poli->get_poli()->result();
35	\$data['kelurahan']= \$this->M_kelurahan->get_kelurahan()->result();
36	\$this->load->view('head-nav-admin', \$data);
37	\$this->load->view('pendaftaran', \$data);
38	\$this->load->view('footer-js', \$data);
39	}else{
40	\$data['data_bpjs']= \$client->call('get_pendaftaran',
41	array('nomor_authentikasi'=>'91212',
42	'kode_authentikasi'=>'0ble50elfd71c96bac94144cc59cff40'));
43	\$data['data_pendaftaran']= \$this->M_pendaftaran->get_pendaftaran()->result();
44	\$this->load->view('head-nav-admin', \$data);
45	\$this->load->view('cek_kepesertaan', \$data);
46	\$this->load->view('footer-js', \$data);}
47	}else{

Tabel 6.3 Implementasi Algoritme Fungsi cek kepesertaan(lanjutan)

49	<code>redirect('auth');</code>
50	<code>}</code>
51	<code>}</code>

6.4.2 Algoritme Pendaftaran

Fungsi algoritme pendaftaran terdapat pada controller pendaftaran yang digunakan untuk melakukan pendaftaran pasien yang berobat ke klinik. Pada tabel 6.5 merupakan implementasi algoritme dari fungsi pendaftaran() dalam controller pendaftaran.

Tabel 6.4 Implementasi Algoritme Fungsi Pendaftaran

1	<code>public function pendaftaran(\$key= NULL) {</code>
2	<code> \$session_data = \$this->session->userdata('logged_in');</code>
3	<code> \$id_user= \$session_data['id_user'];</code>
4	<code> \$jabatan= \$session_data['jabatan'];</code>
5	<code> \$nama_user= \$session_data['nama_user'];</code>
6	<code> \$data['jabatan']= \$jabatan;</code>
7	<code> \$data['id_user']= \$id_user;</code>
8	<code> \$data['nama_user']= \$nama_user;</code>
9	<code> \$data['jenis_identitas']= NULL;</code>
10	<code> \$data['nomor_identitas']=NULL;</code>
11	<code> \$client = new</code> <code> nusoap_client('http://localhost/service_bpjs/webservice.php</code>
12	<code> ');</code> <code> if (\$id_user != '') {</code>
13	<code> if (\$this->input->post()) {</code>
14	<code> \$id_pasien= \$this->input->post('id_pasien');</code>
15	<code> \$tanggal_pendaftaran= \$this->input-></code>
16	<code> post('tanggal_pendaftaran');</code>
17	<code> \$status_peserta= \$this->input->post('status_peserta');</code>
18	<code> \$jenis_kunjungan= \$this->input->post('jenis_kunjungan');</code>
19	<code> \$nomor_identitas= \$this->input->post('nomor_identitas');</code>
20	<code> \$jenis_identitas= \$this->input->post('jenis_identitas');</code>
21	<code> \$id_kelurahan= \$this->input->post('kelurahan');</code>
22	<code> \$nama_pasien= \$this->input->post('nama_pasien');</code>
23	<code> \$tanggal_lahir= \$this->input->post('tanggal_lahir');</code>
24	<code> \$kelamin= \$this->input->post('kelamin');</code> <code> \$nomor_hp= \$this->input->post('nomor_hp');</code>

Tabel 6.4 Implementasi Algoritme Fungsi Pendaftaran(lanjutan)

25	\$alamat= \$this->input->post('alamat');
26	\$poli_tujuan= \$this->input->post('poli_tujuan');
27	\$nomor_rm= \$this->input->post('nomor_rm');
28	\$id_pendaftaran_bpjs;
29	if (\$status_peserta== NULL \$status_peserta=='premi' \$status_peserta=='tidak aktif') {
30	\$jenis_pasien= 'UMUM';
31	\$this->session->set_flashdata('info', "Pasien Umum berhasil terdaftar");
32	}elseif(\$status_peserta=='aktif'){
33	\$jenis_pasien= 'BPJS';
34	\$pendaftaran_bpjs=\$client->call('set_pendaftaran',
35	array('nomor_authentikasi'=>'91212',
36	'kode_authentikasi'=>'0ble50elfd71c96bac94144cc59cff40', 'nom
37	or_bpjs'=> \$nomor_identitas, 'kode_poli'=>\$poli_tujuan,
38	'tanggal_pendaftaran'=>\$tanggal_pendaftaran,
39	'jenis_kunjungan'=> \$jenis_kunjungan));
40	\$this->session->set_flashdata('info', "Pasien BPJS berhasil terdaftar");
41	\$id_pendaftaran_bpjs= \$pendaftaran_bpjs['id_pendaftaran'];
42	} if (\$id_pasien== NULL) {
43	\$id_pasien = \$this->M_pasien->set_pasien(\$nomor_identitas,
44	\$jenis_identitas, \$nama_pasien, \$tanggal_lahir, \$kelamin,
45	\$nomor_hp, \$alamat, \$nomor_rm, \$id_kelurahan);
46	\$this->M_pendaftaran->set_pendaftaran(\$id_pasien,
47	\$jenis_pasien, \$poli_tujuan, \$id_user,
48	\$tanggal_pendaftaran, \$id_pendaftaran_bpjs);
49	}else{
50	\$this->M_pendaftaran->set_pendaftaran(\$id_pasien,
51	\$jenis_pasien, \$poli_tujuan, \$id_user,
52	\$tanggal_pendaftaran, \$id_pendaftaran_bpjs);
53	}
54	}else{
55	redirect('Kelolah_pendaftaran/cek_kepesertaan');
56	}else{
57	redirect('auth');

6.4.3 Algoritme Masukan rekam medis

Fungsi algoritme masukan_rekam_medis terdapat pada controller rekam_medis yang digunakan untuk melakukan memasukan data rekam medis pasien. Pada tabel 6.5 merupakan implementasi algoritme dari fungsi memasukan_rekam_medis() dalam controller rekam_medis_controller.

Tabel 6.5 Implementasi Algoritme Fungsi Masukan rekam medis

1	public function rekam_medis() {
2	\$session_data = \$this->session->userdata('logged_in');
3	\$id_user= \$session_data['id_user'];
4	\$jabatan= \$session_data['jabatan'];
5	\$nama_user= \$session_data['nama_user'];
6	\$data['jabatan']= \$jabatan;
7	\$data['id_user']= \$id_user;
8	\$data['nama_user']= \$nama_user;
9	\$client = new nusoap_client('http://localhost/service_bpjs/webservice.php');
10	\$data['login_result']= \$client->call('cek_auth',
11	array('nomor_authentikasi'=>'91212', 'kode_authentikasi'=>'0ble50elfd71c96bac94144cc59cff40'));
12	if (\$id_user != '') {
13	if (\$this->input->post()) {
15	\$id_pendaftaran= \$this->input->post('id_pendaftaran');
16	\$nomor_id= \$this->input->post('nomor_id');
17	\$id_pendaftaran_bpjs= \$this->input->post('id_pendaftaran_bpjs');
18	\$diagnosa= \$this->input->post('diagnosa[]');
19	\$kesadaran= \$this->input->post('kesadaran');
20	\$berat_badan= \$this->input->post('berat_badan');
21	\$tinggi_badan= \$this->input->post('tinggi_badan');
22	\$sistole= \$this->input->post('sistole');
23	\$diastole= \$this->input->post('diastole');
24	\$respiratory_rate= \$this->input->post('respiratory_rate');
25	\$hearth_rate= \$this->input->post('hearth_rate');
26	\$id_dokter= \$this->input->post('id_dokter');
27	\$keluhan= \$this->input->post('keluhan');
28	\$terapi= \$this->input->post('terapi');
	\$status_pulang= \$this->input->post('status_pulang');

Tabel 6.5 Implementasi Algoritme Fungsi Masukan rekam medis(lanjutan)

29	\$nama_dokter= \$this->M_dokter->get_nama_dokter(\$id_dokter);
30	\$rumah_sakit= \$this->input->post('rumah_sakit');
31	\$poli= \$this->input->post('poli');
32	\$kode_rujukan= NULL;
33	if (\$id_pendaftaran_bpjs != NULL) {
34	\$id_pendaftaran_bpjs;
35	\$result=\$client->call('insert_rm',
36	array('nomor_authentikasi'=>'91212',
37	'kode_authentikasi'=>'0ble50elfd71c96bac94144cc59cff40','id_
38	pendaftaran'=> \$id_pendaftaran_bpjs,'kesadaran'=>
39	\$kesadaran, 'sistole'=> \$sistole, 'diastole'=> \$diastole,
40	'respiratory_rate'=> \$respiratory_rate, 'hearth_rate'=>
41	\$hearth_rate, 'dokter'=> \$nama_dokter, 'keluhan'=>
42	\$keluhan,'terapi'=> \$terapi, 'status_pulang'=>
43	\$status_pulang, 'diagnosa'=>\$diagnosa, 'rumah_sakit'=>
44	\$rumah_sakit, 'poli'=> \$poli));
45	\$result_rujukan=\$client->call('get_rujukan',
46	array('nomor_authentikasi'=>'91212',
47	'kode_authentikasi'=>'0ble50elfd71c96bac94144cc59cff40','id_
48	pendaftaran'=> \$id_pendaftaran_bpjs));
49	foreach (\$result_rujukan as \$key) {
50	\$kode_rujukan= \$key['kode_rujukan'];
51	}
52	}
53	\$id_rm= \$this->M_rekam_medis->insert_rm(\$id_pendaftaran,
54	\$kesadaran, \$sistole, \$diastole, \$respiratory_rate,
55	\$hearth_rate, \$id_dokter, \$keluhan, \$terapi,
56	\$status_pulang);
57	\$this->M_rekam_medis->update_pelayanan(\$id_pendaftaran);
58	for (\$i=0; \$i < count(\$diagnosa); \$i++) {
59	\$this->M_rekam_medis->insert_diagnosa(\$id_rm,
60	\$diagnosa[\$i]);
61	}
62	if (\$status_pulang=='Rujuk Lanjut') {
63	\$this->M_rujukan->insert_rujukan(\$id_pendaftaran,
64	\$rumah_sakit, \$poli, \$kode_rujukan);
65	}
66	\$id_pasien= \$this->M_rekam_medis-
67	>update_pasien(\$id_pendaftaran, \$tinggi_badan,
68	\$berat_badan);
69	\$data['data_pendaftaran']= \$this->M_rekam_medis-
70	>get_pendaftaran()->result();
71	\$data['data_pasien']= NULL;
72	

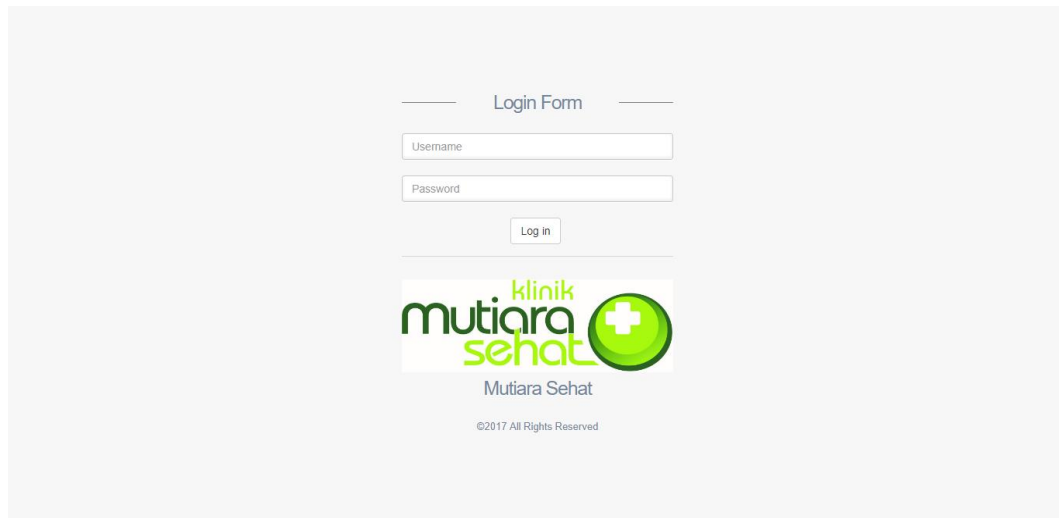
Tabel 6.5 Implementasi Algoritme Fungsi Masukan rekam medis(lanjutan)

63	\$this->session->set_flashdata('info', "Rekam medis berhasil di simpan");
64	\$this->load->view('head-nav-admin', \$data);
65	\$this->load->view('rekam_medis', \$data);
66	\$this->load->view('footer-js', \$data);
67	(\$this->input->get()) {
68	\$data['data_pendaftaran'] = \$this->M_rekam_medis->get_pendaftaran()->result();
69	\$id_pendaftaran = \$this->input->get('id');
70	\$data['data_pasien'] = \$this->M_rekam_medis->get_pasien_id(\$id_pendaftaran)->result();
71	\$data['daftar_diagnosa'] = \$this->M_diagnosa->get_diagnosa()->result();
72	\$data['data_dokter'] = \$this->M_dokter->get_dokter()->result();
73	\$this->load->view('head-nav-admin', \$data);
74	\$this->load->view('rekam_medis', \$data);
75	\$this->load->view('footer-js', \$data);
76	}else{
77	\$data['data_pasien'] = NULL;
78	\$data['data_pelayanan'] = \$this->M_rekam_medis->get_pelayanan()->result();
79	\$data['data_pendaftaran'] = \$this->M_rekam_medis->get_pendaftaran()->result();
80	\$this->load->view('head-nav-admin', \$data);
81	\$this->load->view('rekam_medis', \$data);
82	\$this->load->view('footer-js', \$data);
83	}
84	}else{
85	redirect('auth');
86	}
87	}
88	}

6.5 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna diimplementasikan atau dibuat berdasarkan perancangan sistem sebelumnya yang kemudian diterapkan ke dalam bahasa HTML5 untuk antarmuka pengguna. Berikut ini hasil implementasi antarmuka pengguna sistem informasi rekam medis

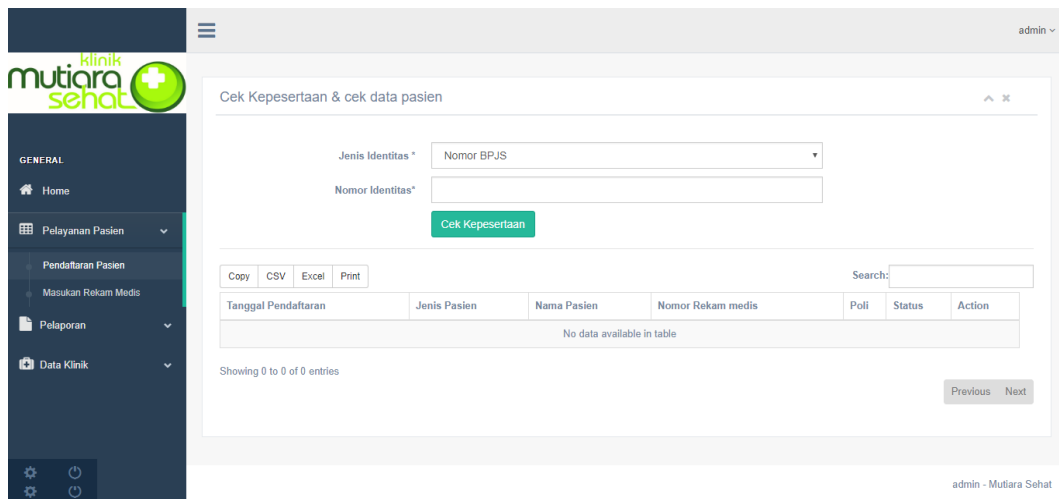
6.5.1. Implementasi Antarmuka halaman *log in*



Gambar 6.2 Implementasi Antarmuka halaman login

Gambar 6.3 menunjukkan hasil implementasi rancangan antarmuka halaman login. Halaman ini terdiri *form* berisi *username* dan *password* berserta tombol *log in* untuk masuk kedalam sistem rekam medis yang dilakukan oleh petugas admin klinik atau kordinator klinik.

6.5.2. Implementasi Antarmuka halaman cek kepesertaan



Gambar 6.3 Implementasi Antarmuka halaman cek kepesertaan

Gambar 6.3 menunjukkan hasil implementasi racangan antarmuka halaman cek kepesertaan pasien yang dimana terdapat *form* untuk memilih jenis identitas pasien beserta nomor identitas pasien kemudian tombol untuk melakukan cek kepesertaan pasien. Terdapat tabel pasien yang telah mendaftar sebelumnya yang ditampilkan dengan kolom tanggal pendaftaran pasien, jenis pasien, nama pasien, nomor rekam medis, poli, status, dan action berisikan tombol untuk membatalkan pendaftaran pasienn

6.5.3. Implementasi Antarmuka halaman pendaftaran pasien

The screenshot displays the 'Mutiara Sehat' application interface for patient registration. The left sidebar contains navigation links: 'GENERAL', 'Home', 'Pelayanan Pasien', 'Pelaporan', and 'Data Klinik'. The main content area is titled 'Cek Kepesertaan' and features two primary forms. The top form, 'Cek Kepesertaan', includes input fields for 'Jenis Identitas' and 'Nomor Identitas', followed by a green 'Cek Kepesertaan' button. The bottom form, 'Data dari BPJS kesehatan', contains fields for 'Tanggal Kunjungan', 'Poli yang dituju', 'Jenis Kunjungan' (with radio buttons for 'Kunjungan Sakit' and 'Kunjungan Sehat'), 'Nomor Identitas', 'Jenis Identitas' (with a radio button for 'BPJS'), 'Nama', 'Tanggal Lahir', 'Kelamin', 'Nomor Hp', 'Alamat', and 'Nomor RM'. A green 'Daftar' button is positioned at the bottom of this form. To the right of the registration form, there is a section titled 'Data dari BPJS kesehatan' with fields for 'No Kartu BPJS', 'Nama', 'Status Peserta', 'Jenis Peserta', 'Tanggal Lahir', 'Kelamin', and 'PPK Umum'. The interface is clean and modern, with a dark blue sidebar and a light gray main content area.

Gambar 6.4 Implementasi Antarmuka halaman pendaftaran

Gambar 6.4 menunjukkan hasil dari implementasi rancangan antarmuka halaman pendaftaran pasien untuk pasien berobat atau pemeriksaan. Pada halaman masih terdapat *form* cek kepesertaan pasien digunakan untuk melakukan cek kepesertaan ulang sedangkan di bawahnya terdapat 2 form yaitu form hasil cek kepesertaan pasien dari service BPJS sedangkan form selanjutnya hasil cek data pasien dari data klinik. Hasil dari cek kepsertaan pasien dari BPJS akan menentukan apakah pasien dikatagorikan pasien umum atau pasien BPJS.

6.5.4. Implementasi Antarmuka halaman pencatatan rekam medis pasien

The screenshot shows the 'Masukan data Rekam medis' (Enter patient medical record data) form. The form is organized into several sections:

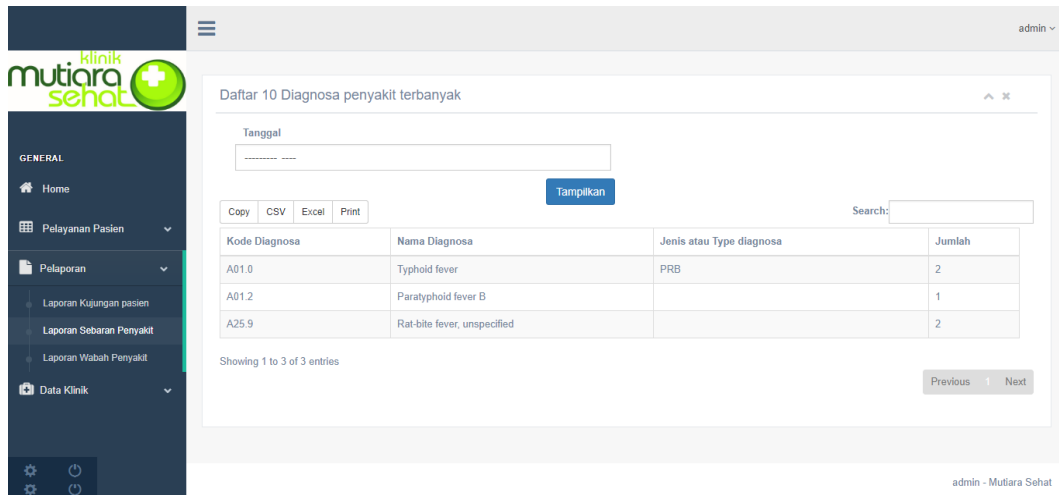
- Patient Information:**
 - Nama pendaftar*: [Dropdown menu]
 - Nama Pasien*: Masnur Anwar
 - Jenis Kelamin*: L
 - Tanggal Lahir*: 05/01/1995
 - Poli*: GIGI
- Kondisi Fisik (Physical Condition):**
 - Berat Badan*: 50 Kg
 - Tinggi Badan*: 144 Cm
- Pemeriksaan Fisik (Physical Examination):**
 - Kesadaran*: Compos Mentis
 - Sistole*: [Input field] mmHg
 - Diastole*: [Input field] mmHg
 - Respiratory Rate*: [Input field] per minute
 - Hearth Rate*: [Input field] bpm
 - Keluhan*: [Text area]
 - Diagnosa*: [Text area]
 - Terapi*: [Text area]
 - Tenaga Medis*: dr Putri
 - Status Pulang*: Berobat Jalan

A green 'Simpan' (Save) button is located at the bottom left of the form.

Gambar 6.5 Antarmuka Halaman pencatatan rekam medis

Pada gambar 6.5 adalah hasil dari implementasi rancangan antarmuka halaman pencatatan rekam medis pasien. Pada halaman rekam medis pasien terdapat *form* yang berisikan data pasien dan *form* isian data pemeriksaan yang meliputi kesadaran, sistole, diastole, respiratory rate, hearth ratem keluhan, diagnosa, terapi, tenaga medis yang melakukan penanganan dan status pulang pasien.

6.5.5. Implementasi Antarmuka halaman laporan sebaran penyakit atau diagnosa pasien



Gambar 6.6 Implementasi Antarmuka sebaran penyakit

Gambar 6.6 menunjukkan hasil implementasi rancangan antarmuka halaman sebaran penyakit yang menunjukkan daftar 10 diagnosa penyakit terbanyak. Terdapat *form* tanggal untuk memilih bulan terhadap tampilan data sebaran penyakit yang akan ditampilkan.

4.4. Implementasi *web service*

Implementasi komunikasi data antara sistem informasi rekam medis dengan BPJS dibangun sebuah *web service* oleh diakses oleh sistem informasi rekam medis. Service service yang dibangun sesuai dengan perancangan dan kebutuhan data yang dibutuhkan oleh sistem informasi rekam medis. Implementasi dari *weservice* BPJS dibangun dengan menggunakan *library nuSoap*. Berikut pada gambar 6.7 merupakan hasil implementasi *webservice* yang telah dibangun.



Gambar 6.7 Implementasi WSDL *web service*

Pada gambar 6.7 tampilan *WSDL* ketika mengakses *link* atau alamat dari *webservice* yang menampilkan beberapa *service* yang disediakan. Berikut merupakan penjelasan dari *service service* yang digunakan untuk mendukung sistem informasi rekam medis. Berikut merupakan gambar serta penjelasan masing masing *service*

6.5.1 *Service cek kepesertaan*

Tabel 6.6 Penjelasan *service* cek kepesertaan

<i>Nama service</i>	Cek kepesertaan
<i>Parameter</i>	Nomor autentikasi, kode autentikasi, jenis identitas, dan nomor identitas
<i>Output</i>	<pre>{ "status": "true/fase", "data_peserta": { "nama_peserta": "Nama Peserta", "nik": "NIK peserta", "nomor_bpjs": "Nomor BPJS Peserta", "jenis_peserta": "BPJS Ketenaga Kerjaan", "jenis_kelamin": "L/P", "tanggal_lahir": "Tanggal Lahir", "status_peserta": "aktif/premi", "PPK_umum": "1 - Mutiara Sehat" } }</pre>

6.5.2 *Service insert pendaftaran*

Tabel 6.7 penjelasan *service* insert pendaftaran

<i>Nama service</i>	Set pendaftaran
<i>Parameter Request</i>	Nomor autentikasi, kode autentikasi, jenis identitas, nomor BPJS, kode poli, tanggal pendaftaran dan jenis kunjungan
<i>Output</i>	<pre>{ "status_pendaftaran": "true/fase", "id_pendaftaran": "id pendaftaran peserta" }</pre>

6.5.3 Service set rekam medis

Tabel 6.8 Penjelasan service set rekam medis

<i>Nama service</i>	Set rekam medis
<i>Parameter Request</i>	Nomor autentikasi, kode autentikasi, jenis identitas, nomor pendaftaran, kesadaran, sistole, diastole, respiratory rate, hearth rate, dokter yang menangani, keluhan, terapi, status pulang dan diagnosa
<i>Output</i>	<pre>{ "status ":"true/fase", "id_rujukan":"id rujukan" }</pre>

6.5.4 Service delete pendaftaran

Tabel 6.9 penjelasan service delete pendaftaran

<i>Nama service</i>	Delete pendaftaran
<i>Parameter Request</i>	Nomor autentikasi, kode autentikasi, id pendaftaran
<i>Response</i>	<pre>{ "status ":"true/fase" }</pre>

6.5.5 Service get diagnosa

Tabel 6.10 penjelasan service get diagnosa

<i>Nama service</i>	Get diagnosa
<i>Parameter Request</i>	Nomor autentikasi, kode autentikasi
<i>Output</i>	<pre>{ "status ":"true", "all_diagnosa": [{ "kode_diagnosa":"Kode diagnosa", "nama_diagnosa":"Nama diagnosa", "type_diagnosa":"type diagnosa" }] }</pre>