

Pengembangan Open E-health Gunadarm Information System

*Berikut Penggunaan Situs Pengelolaan Data
Rekam Medis Elektronik
<http://www.opeltgunasys.org/>*

A. B. Mutiara, A. Muslim, T. Oswari, R. A. Miharja

2012



Penerbit Gunadarma

Pengembangan Open E-health Gunadarma Information System

Serta Penggunaan Situs Pengelolaan Data Rekam Medis Elektronik

<http://www.opeltgunasys.org/>

A. B. Mutiara, A. Muslim, T. Oswari, R. A. Miharja



Penerbit Gunadarma

Pengembangan Open E-Health Gunadarma Information System

Penyusun : A. B. Mutiara, A. Muslim, T. Oswari, R. A. Miharja

Desain : R.A. Miharja

Desain dan Layout : A. Muslim dan R.A. Miharja

Diterbitkan pertama kali oleh Universitas Gunadarma
Hak Cipta dilindungi Undang-Undang
Depok, September 2012

ISBN 978-602-9438-09-3



KATA PENGANTAR

Puji syukur kepada Allah SWT yang telah memberikan rahmat, hidayah, pencerahan, dan kekuatan sehingga Buku Pengembangan berikut Petunjuk Penggunaan Situs Open E-health Gunadarma System Informasi ini dapat diselesaikan dengan lancar. Buku petunjuk penggunaan Situs Opelth Gunasys ini merupakan bagian dari pembangunan Situs Rekam Medis Elektronik Open E-health Gunadarma Information System.

Buku Pengembangan dan Petunjuk Penggunaan situs Opelth Gunasys ini berisikan i) model, metodologi pengembangan situs e-Health dan ii) semua panduan untuk menggunakan fasilitas-fasilitas yang tersedia dalam situs ini berdasarkan aktor-aktor yang berperan dalam pengolahan data rekam medis. Situs ini menyediakan fasilitas-fasilitas yang dapat mengoptimalkan pengolahan data rekam medis.

Akhirnya, kami menyadari bahwa Buku Pengembangan dan Petunjuk Penggunaan Situs Opelth Gunasys ini masih kurang sempurna dalam mengakomodir semua yang dapat dilakukan oleh situs itu sendiri dan keinginan pengguna. Oleh karenanya segala kritik yang membangun dan saran untuk perbaikan dan kebaikan sangat kami harapkan. Semoga buku pengembangan dan panduan ini dapat membantu pengguna dalam i) memahami prinsip pengembangan e-Health dan ii) mengoperasikan situs Open E-health Gunadarma Information System ini.

Depok, September 2012

A. B. Mutiara, A. Muslim, T. Oswari, R. A. Miharja

DAFTAR ISI

KATA PENGANTAR	- 3 -
DAFTAR ISI	- 5 -
BAB I. PENDAHULUAN	- 7 -
BAB II. DEFINISI, KONSEP DAN TOOLS	- 11 -
2.1. DEFINISI DAN FUNGSI RUMAH SAKIT	- 11 -
2.2. KESESUAIAN DAN KESELARASAN TEKNOLOGI DENGAN KEBUTUHAN	- 13 -
2.3. KONSEP PENGEMBANGAN HEALTH INFORMATION SYSTEM	- 13 -
2.4. REKAM KESEHATAN ELEKTRONIK	- 18 -
2.5. PROTÉGÉ	- 20 -
2.6. YII FRAMEWORK	- 20 -
2.6.1. MODEL VIEW CONTROLLER	- 21 -
2.6.2. FITUR <i>YII FRAMEWORK</i>	- 24 -
2.6.3. BEKERJA DENGAN <i>FORM</i>	- 25 -
2.6.4. BEKERJA DENGAN DATABASE	- 26 -
2.7. WAMP SERVER	- 26 -
2.7.1. APACHE	- 27 -
2.7.2. MYSQL	- 28 -
2.7.3. PHP	- 28 -
2.8. UML	- 28 -
2.8.1. USE CASE DIAGRAM	- 28 -
2.8.2. CLASS DIAGRAM	- 29 -
2.9. OPENEHR	- 29 -
2.9.1. MODEL INFORMASI STRUKTUR DATA (<i>DATA STRUCTURE INFORMATION MODEL</i>)	- 30 -
2.9.2. MODEL INFORMASI TIPE DATA (<i>DATA TYPE INFORMATION MODEL</i>)	- 31 -
2.9.3. TEMPLAT MODEL OBJEK (<i>TEMPLATE OBJECT MODEL</i>)	- 32 -
BAB III. METODE PENGEMBANGAN	- 35 -
3.1. ANALISA	- 36 -
3.2. DESAIN	- 36 -
3.3. CODING DAN TESTING	- 37 -

3.4. PENERAPAN	- 38 -
3.5. PEMELIHARAAN	- 38 -

BAB IV. PANDUAN APLIKASI OPELT GUNASYS	- 39 -
---	---------------

4.1. PEDOMAN UNTUK PUBLIK	- 39 -
4.1.1. HALAMAN HOME	- 39 -
4.1.2. HALAMAN LOGIN	- 39 -
4.1.3. HALAMAN USER GUIDE	- 40 -
4.1.4. HALAMAN FAQ	- 40 -
4.2. PEDOMAN UNTUK REGISTRASI PEMAKAI	- 41 -
4.2.1. HALAMAN USER	- 41 -
4.2.2. HALAMAN PASIEN	- 45 -
4.3. PEDOMAN UNTUK ADMINISTRATOR	- 65 -
4.3.1. CARA MEMBUAT PENGGUNA BARU	- 65 -
4.3.2. MENGELOLA REFERENSI	- 67 -
4.4. PEDOMAN UNTUK PETUGAS ADMINISTRASI	- 68 -
4.4.1. MEMBUAT DEMOGRAFI PASIEN	- 68 -
4.4.2. MEMBUAT KARTU PASIEN	- 69 -
4.4.3. PENGGUNAAN DASHBOARD STAFF	- 71 -
4.4.4. MEMBUAT TRANSAKSI PENANGANAN	- 71 -
4.4.5. MEMBUAT TRANSAKSI PELAYANAN	- 72 -
4.4.6. MEMBUAT INFO KLINIK	- 73 -
4.5. PEDOMAN UNTUK DOKTER	- 74 -
4.5.1. DOKTER MERUBAH ID STATUS KARTU PASIEN DIPERIKSA	- 75 -
4.5.2. DOKTER MENULISKAN HASIL PEMERIKSAAN	- 76 -
4.5.3. DOKTER MEMBUAT SURAT RUJUKAN DOKTER	- 78 -
4.5.4. DOKTER MEMBUAT SURAT RUJUKAN LABORAN	- 79 -
4.5.5. DOKTER MENGANALISIS HASIL LAB. DAN KEMBALI MENULIS ARCHETYPE	- 79 -
4.6. PEDOMAN UNTUK PENUNJANG MEDIS	- 80 -
4.6.1. LABORAN MEMBUAT HASIL LABORATORIUM	- 81 -
4.7. PEDOMAN UNTUK PASIEN	- 82 -
4.7.1. PASIEN MENGAKSES KARTU PASIEN	- 83 -
4.7.2. PASIEN MENGUBAH DEMOGRAFI	- 84 -

DAFTAR PUSTAKA	- 85 -
-----------------------	---------------

BAB I. PENDAHULUAN

Dalam program 100 hari Kabinet Indonesia Bersatu periode kedua, Presiden RI menetapkan 45 program penting yang akan dijalankan di seluruh tanah air berkaitan dengan pembangunan sektoral dan regional. Dari 45 program ini telah dipilih 15 program unggulan, dimana kesehatan masuk dalam program ke 12. Untuk mewujudkan masyarakat Indonesia yang sehat di masa depan melalui pembangunan kesehatan dirumuskan visi pembangunan kesehatan sebagai Indonesia Sehat 2015. Salah satu misi untuk mencapai visi Indonesia Sehat 2015 adalah memelihara dan meningkatkan pelayanan kesehatan yang bermutu, merata, dan terjangkau.

Visi dan misi Indonesia Sehat 2015 ini dapat tercapai dengan baik jika ditunjang oleh tenaga medis yang sangat kompeten, sarana prasarana yang lengkap dan baik, dan khususnya dukungan teknologi informasi dan komunikasi (TIK) serta basis data medis yang terpadu. Hingga saat ini, kebanyakan rumah sakit di Indonesia hanya mengembangkan sistem informasi medis bersifat lokal. Untuk mencapai misi Indonesia sehat 2015, keadaan seperti ini tidak dapat mendukung misi untuk memberikan pelayanan kesehatan yang bermutu karena pelayanan rumah sakit merupakan bagian dari pelayanan kesehatan.

Untuk menyediakan pelayanan kesehatan dengan lingkup yang luas dan komprehensif perlu dibuat sebuah aplikasi sistem informasi terpadu yaitu *Health Information System* (HIS). Teknologi yang disediakan oleh HIS sangat bermanfaat bagi peningkatan kualitas pelayanan yang dapat meningkatkan kinerja manajemen rumah sakit, seperti dokumentasi yang *auditable* dan *accountable*, data rekam medis yang terintegrasi sehingga dapat mempercepat akses dan proses pertukaran data rekam medis pasien antar rumah sakit, puskesmas dan klinik. Dengan demikian dapat dicapai visi dan misi untuk mewujudkan Indonesia Sehat 2015.

Untuk mewujudkan manfaat dari teknologi HIS tersebut dapat diimplementasikan dengan membuat suatu rekam medis elektronik (*Electronic Health Record*) yang memiliki kemampuan untuk menyimpan data-data medis pasien. Spesifikasi rekam medis yang diusulkan adalah spesifikasi terbuka yang terdapat pada *OpenEHR*. Spesifikasi *OpenEHR* memungkinkan pertukaran informasi antara berbagai format rekam medis, seperti HL7 dan CEN/ISO.

Berdasarkan latar belakang yang telah diuraikan sebelumnya, Penulis mencoba mewujudkan misi Indonesia Sehat 2015 dengan membuat suatu aplikasi HIS yaitu membuat aplikasi rekam medis elektronik berbasis *web* untuk mempercepat akses dan proses pertukaran data rekam medis pasien antar rumah sakit, puskesmas dan klinik dengan spesifikasi yang disediakan oleh *OpenEHR*.

Buku ini menyajikan bagaimana membuat aplikasi rekam medis elektronik untuk mempercepat akses dan proses pengolahan data rekam medis dengan spesifikasi yang disediakan oleh *OpenEHR*. Dengan kata lain, buku menyajikan bagaimana pembuatan aplikasi *web* rekam medis elektronik berbasis *OpenEHR*. Pembuatan aplikasi hanya difokuskan pada pembuatan aplikasi *web* berbasis *OpenEHR*, sedangkan untuk pertukaran data antara EHR satu dengan EHR yang lain tidak dijelaskan dalam buku ini.

Buku ini juga menyajikan hasil pengembangan Aplikasi *website* rekam medis elektronik dengan menggunakan spesifikasi *OpenEHR*. Spesifikasi ini digunakan dengan harapan dapat mempermudah akses, pengolahan dan penyimpanan data rekam medis oleh aktor-aktor yang berperan dalam pengolahan data rekam medis. Aplikasi yang dikembangkan ini diharapkan dapat bermanfaat bagi pengolahan data dan informasi kesehatan untuk peningkatan kualitas pelayanan yang akan berdampak meningkatnya kinerja manajemen rumah sakit sehingga menjadi pendukung diwujudkan visi dan misi Indonesia Sehat 2015

Dalam pembuatan aplikasi *web* rekam medis elektronik berbasis *OpenEHR* ini dibutuhkan suatu metode pengembangan yang merupakan prosedur pengumpulan data yang digunakan untuk memecahkan masalah yang dihadapi. Adapun metode ilmiah yang dilakukan pada pengembangan aplikasi ini yaitu dengan menggunakan metode pengembangan rekayasa perangkat lunak *Waterfall*. Metode ini memiliki tahap-tahap berikut :

1. Analisis.
2. Design.
3. Coding dan Testing.
4. Penerapan.
5. Pemeliharaan.

Untuk selengkapnya tahapan-tahapan di atas akan dijelaskan pada Bab III Metode Pengembangan.

BAB II. DEFINISI, KONSEP DAN TOOLS

2.1. Definisi dan Fungsi Rumah Sakit

Definisi dan fungsi rumah sakit telah diatur dalam Surat Keputusan Menteri Kesehatan Republik Indonesia Nomor 983.MENKES/SK/XI/1992, mengenai Organisasi Rumah Sakit dan Pedoman Rumah Sakit Umum, yang menyatakan bahwa Rumah Sakit Umum adalah rumah sakit yang memberikan pelayanan kesehatan yang bersifat mendasar, spesialisik dan pendidikan tenaga kesehatan dan pelatihan, beberapa tugas dan fungsi dari rumah sakit adalah melaksanakan pelayanan medis, pelayanan penunjang medis, melaksanakan pelayanan rawat jalan dan rawat darurat dan rawat tinggal, melaksanakan pelayanan rawat inap, melaksanakan pelayanan administratif dan membantu penelitian dan pengembangan kesehatan. (Surat Keputusan Menteri Kesehatan Nomor 983.MENKES/SK/ XI/1992)

Menurut Handoyo et., al. 2008; Singer et., al. 2009, tugas dan fungsi rumah sakit yang bersifat pelayanan mengharuskan setiap personal yang terlibat pada penyelenggaraan rumah sakit harus memenuhi standart dan kriteria minimum. Untuk meningkatkan mutu pelayanan rumah sakit pemerintah dalam hal ini Departemen Kesehatan berkerjasama dengan Departemen Pendidikan Nasional, meningkatkan jumlah tenaga kesehatan terdidik (dokter, bidan, spesialis, laboran, dan teknisi), selain itu pemerintah juga melakukan akreditasi terhadap tingkat pelayanan rumah sakit kepada pasien. Akreditasi meliputi pelayanan sampai Standart ISO 14000 dan ISO 2000, standart pelayanan yang harus dilakukan adalah pelayanan gawat darurat, pelayanan medik, pelayanan administrasi, pelayanan perawatan dan pelayanan rekam medik.

Werner et., al. 2008; Pittman et., al. 2009 menyatakan bahwa Rumah Sakit sebagai salah satu organisasi pelayanan di bidang kesehatan telah memiliki otonomi dan bersifat swadana, sehingga pihak rumah sakit dituntut untuk memberikan pelayanan yang sebaik-baiknya dengan manajemen yang seefektif mungkin. Sehingga rumah sakit harus bekerja keras agar dapat

memenuhi kebutuhan pembiayaan operasional. Hal ini disebabkan oleh setiap pengambilan keputusan yang tidak tepat akan berakibat pada inefisiensi dan penurunan kinerja rumah sakit, kendala akan terjadi jika kurang lengkapnya informasi yang tersedia untuk pengambilan keputusan. Kecanggihan teknologi yang dimiliki setiap rumah sakit bukan merupakan suatu jaminan akan terpenuhinya informasi, melainkan perlunya dukungan sistem yang terstruktur, handal dan mampu mengakomodasi seluruh informasi yang dibutuhkan dan harus dapat menjawab tantangan serta permasalahan yang dihadapi. Namun kenyataan yang dihadapi dilapangan menunjukkan lemahnya sistem informasi manajemen yang dimiliki oleh pihak rumah sakit yang berakibat pada terjadinya inefisiensi pengelolaan rumah sakit. Lemahnya sistem informasi manajemen membawa pengaruh secara langsung pada kinerja sistem pengendalian manajemen, yang akan berakibat pada melemahnya perencanaan dan sekaligus berkurangnya kontrol atas pelaksanaan operasional rumah sakit.

Menurut Ed Ort. 2005, prinsip arsitektural dalam merancang dan mendefinisikan *service* sebagai *interoperable services* adalah sebuah sistem yang menunjuk pada berbagai kemampuan *service* yang berbeda dan mengorganisasikannya sehingga dapat bekerja bersama (*inter-operate*). Prinsip tersebut seperti, a) *service encapsulation* sebagian besar *service* (*web service*) dikonsolidasikan untuk digunakan dengan berdasar pada SOA. b) *Service loose coupling* yang memelihara sebuah hubungan yang meminimalisasikan ketergantungan antar *service*. c) *Service contract* terpaku pada sebuah *communication agreement*, yang didefinisikan secara umum pada satu atau lebih *service description documents*. d) *Service abstraction* diluar apa yang dideskripsikan dalam *service contract*, setiap *service* menyembunyikan logikanya dari dunia luar. e) *Service reusability*, logika dibagi kedalam *service-service* yang berorientasi pada prinsip *reusability* (penggunaan kembali). f) *Service composability* merupakan kumpulan *service* dapat dikoordinasi dan dibentuk menjadi sebuah *service* gabungan. g) *Service autonomy* memegang control terhadap logika yang dia gunakan sendiri. h) *Service optimization* yang mengutamakan *high-quality service* dibanding dengan *low-quality service*. i)

Service discoverability dirancang sedemikian rupa sehingga *service* tersebut dapat ditemukan dan dinilai dengan mekanisme yang sudah ada. j) *Service relevance* dibuat sesuai dengan kebutuhan sehingga dikenali sebagai *service* yang memiliki nilai (*meaningful service*).

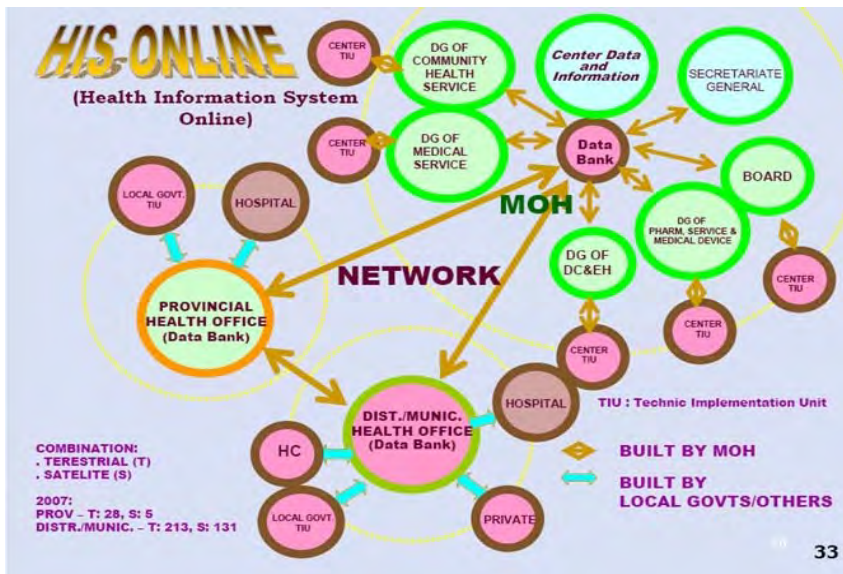
2.2. Kesesuaian dan Keselarasan Teknologi dengan Kebutuhan

Sudah menjadi kenyataan bahwa layanan kesehatan yang sangat baik pada suatu negara, akan menjamin peningkatan tingkat kecerdasan dan intelegensi masyarakat negara tersebut. Maka tidak heran bila kebijakan pemerintah saat ini mengarah pada penciptaan Negara Indonesia sehat. Untuk pencapaian tujuan tersebut, tidaklah cukup dengan meningkatkan jumlah dana untuk keperluan peningkatan jumlah tenaga medis, pengadaan peralatan medis dan obat-obatan yang bersifat jangka pendek, tetapi juga dibutuhkan dukungan IT terapan bidang medis. Teknologi ini dapat digunakan untuk jangka panjang dan menjadi:

1. Media peningkatan kemampuan bagi setiap tenaga medis melalui pertukaran data dan informasi.
2. Pengarsipan jenis penyakit, gejala-gejalanya, metode penanganannya, jenis obat yang dapat digunakan serta pencegahan dini yang dapat dilakukan.
3. Dapat dijadikan salah satu acuan konsultasi dalam menentukan jenis penyakit pasien dan penanganannya yang tepat.
4. Dapat meningkatkan efisiensi manajemen control.

2.3. Konsep Pengembangan Health Information System

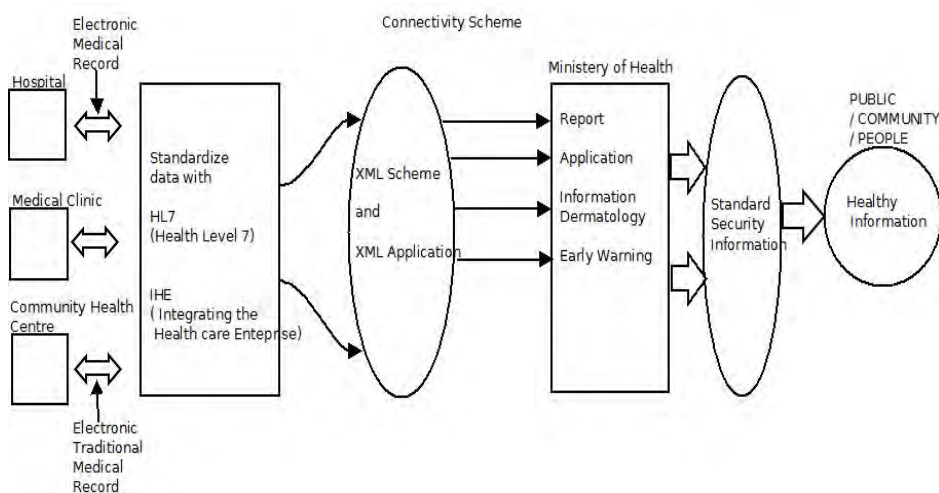
Pemerintah Indonesia melalui Departemen Kesehatan telah merilis suatu konsep pelayanan rumah sakit secara elektronik (HIS : *Health Information System Online*) dengan memanfaatkan teknologi komputasi yang lebih komprehensif, seperti yang diperlihatkan pada gambar berikut ini. Konsep ini menjadi landasan terwujud sistem HIS yang dapat membantu dokter, dan tenaga medis dalam melakukan diagnosa penyakit secara akurat dengan proses data rujukan, dalam pembuat kebijakan kesehatan, dalam membuat perencanaan dan penanggulangan masalah kesehatan.



33

Gambar 2.1. Rancangan Arsitektur HIS Online Profil Kesehatan Indonesia 2008

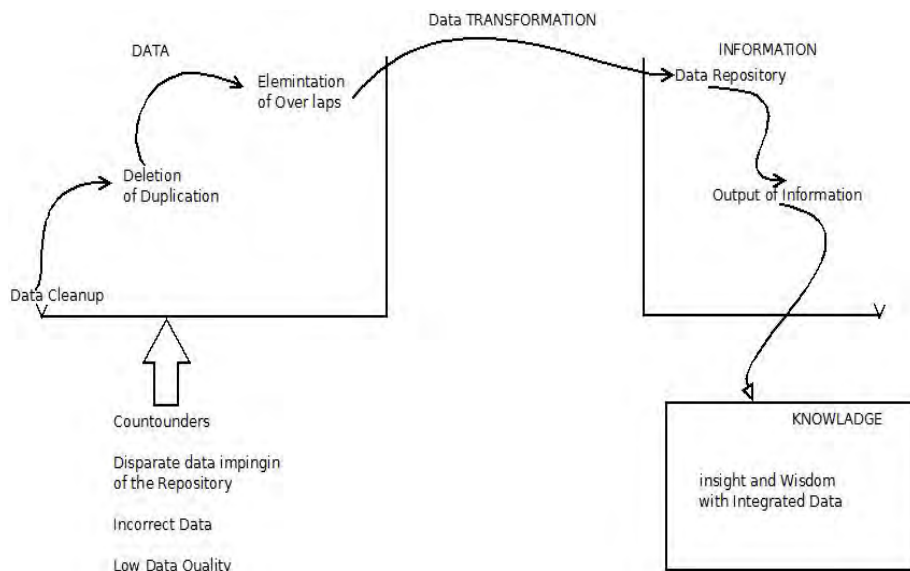
Untuk mengembangkan Sistem Informasi Rumah Sakit yang terintegrasi dengan *Health Information System Online*, diperlukan sebuah konsep disain arsitektur sistem yang dapat menjadi jembatan antara Sistem Informasi Rumah Sakit, Sistem Informasi Puskesmas dan Sistem Informasi Dokter Keluarga menjadi sebuah sistem terkoneksi dan terintegrasi dengan asas interoperabiliti. Koneksi dan integrasi sistem tersebut dapat digambarkan sebagai berikut:



Gambar 2.2. Skema Koneksitas Sistem

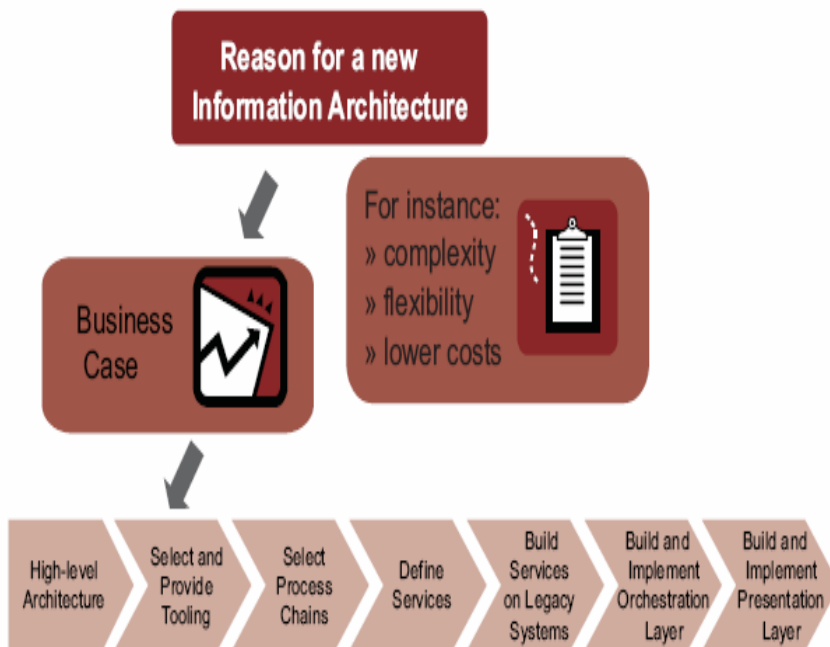
Ada 5 (lima) tahapan yang dibutuhkan untuk melakukan proses interoperability system HIS. Kelima tahapan tersebut dapat diuraikan sebagai berikut:

1. Inventarisasi aplikasi yang digunakan di oleh dokter keluarga/klinik, puskesmas dan rumah sakit untuk mengetahui karakteristik sistem yang digunakan.
2. Membuat standardisasi beberapa proses krusial pada aplikasi sistem yang mengacu pada HL7 atau Open HER (Standar internasional untuk aplikasi Sistem Informasi Rumah Sakit dan kesehatan yang telah diterapkan di Amerika dan Eropa).
3. Membuat disain koneksi berbasis *services* dengan XML *Scheme* dan XML *Application*.
4. Membuat Standar transformasi data untuk kebutuhan proses komunikasi data rujukan sebagai bagian dari *interoperability system*. Seperti pada tampak pada gambar 2.3.
5. Membuat simulasi aplikasi standar dan melakukan ujicoba pada tingkatan pelayanan kesehatan (Dokter keluarga/Klinik, Puskesmas, Rumah Sakit dan Dinas Kesehatan) baik pada tingkat lokal maupun pada tingkat nasional.



Gambar 2.3. Proses Transformasi Data

Sementara itu, terdapat tujuh langkah yang dikembangkan untuk mewujudkan HIS yang mendukung pelayanan prima (*excellence services*) dengan arsitektur berbasis SOA, tanpa mengganti aplikasi yang sudah ada dan digunakan saat ini, tapi dengan melakukan penyempurnaan penguraian pelaksanaan *service* yang lebih spesifik [IBM, SOA *standart*] seperti terlihat pada gambar 2.4. berikut ini:



Gambar 2.4. *Reuseable System* dengan SOA Method

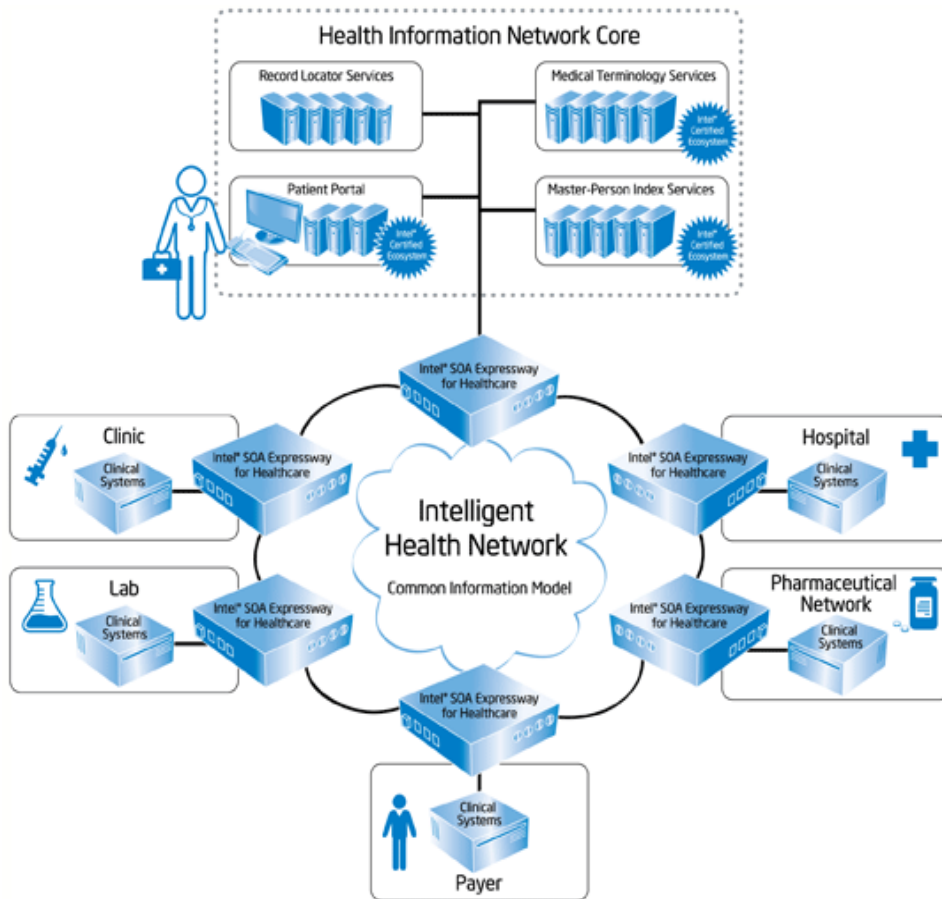
Berikut penjelasan tujuh langkah *reuseable system* dengan metode SOA:

1. *High Level Architecture*. Rancangan services mengacu pada kepentingan menyeluruh, Manajemen, Pelaksana Layanan (Dokter, Perawat, Teknisi, Laboran), Laboratorium, Apotek, Pengawas Kesehatan, Departemen Kesehatan, dan Masyarakat. Proses ini menjadi bagian teramat penting karena memerlukan interaksi yang intensif dengan pelaksana lapangan dan bersifat subjektif.
2. *Select and Provide Tooling*. Pemilihan dan penentuan alat bantu disesuaikan dengan model pelayanan yang akan diberikan dan mengacu pada sistem pelayanan yang optimal dan efektif.
3. *Select process Chains*. Menentukan urutan dan bagian proses yang harus dijalankan secara berurutan/sequential atau yang dapat dilakukan secara parallel.

Tidak memisahkan proses berdasarkan kepentingan dan pesanan bagian tertentu tapi merupakan sistem proses yang terintegrasi.

4. *Define Services*. Mendefinisikan setiap services dengan layak, tidak berlebihan dan mencakup semua yang menjadi kepentingan manajemen dan pelanggan. *Service* didefinisikan berdasarkan prosedur yang telah disepakati dan mengacu pada aturan legal yang dikeluarkan oleh instansi yang berwenang.
5. *Build Services on legacy system*. Melakukan *reusable* dan *restructures* pada sistem yang sudah terimplementasi, agar lebih menghemat dan memudahkan user melaksanakan aplikasi sistem yang telah diperbaharui.
6. *Build and implement Orchestration Layer*. Membangun aplikasi yang menggabungkan dan menghubungkan semua sistem layanan pada tingkat aplikasi, untuk tidak merubah secara mendasar dan menyeluruh aplikasi layanan yang sudah terimplementasi. Membangun sebuah aplikasi antara yang menjadi tempat pertukaran data dan informasi dari satu sistem layanan ke sistem layanan lainnya.
7. *Build and Implement Presentation Layer*. Membangun aplikasi yang lebih baik pada bagian layanan yang memerlukan perubahan tampilan untuk memberikan kemudahan baru bagi pelaksana dan penerima layanan.

Pengembangan yang dilakukan menggunakan semua tahapan di atas, dimulai dengan studi perencanaan, perancangan, implementasi hingga penerepan SOA (*Service Oriented Architecture*) untuk pengembangan HIS (*Health Information System*) di Indonesia. Jaringan computer yang digunakan untuk sistem diatas adalah jaringan komputer berbasis P2P. Gambar 2.5. memperlihatkan secara umum bagan pengembangan HIS yang didukung oleh SOA.



Gambar 2.5. Jaringan dan Koneksitas Sistem Kesehatan dengan Metoda SOA
(dikembangkan oleh Intel & IBM coop.)

2.4. Rekam Kesehatan Elektronik

Rekam Kesehatan Elektronik atau *Electronic Health Record* sering disingkat *EHR*. *EHR* merupakan kegiatan mengkomputerisasikan isi rekam jejak kesehatan dan proses yang berhubungan dengannya. Pada awalnya rekam kesehatan di Indonesia masih dikenal dengan istilah rekam medis yang sampai saat inipun sebagian rumah sakit di Indonesia masih menggunakan istilah yang sama. Rekam Medis adalah “Himpunan fakta tentang kehidupan seorang pasien dan riwayat penyakitnya, termasuk keadaan sakit, pengobatan saat ini dan lampau yang ditulis oleh para praktisi kesehatan dalam upaya mereka memberikan pelayanan kesehatan terhadap pasien” [Huffman, 1999].

Berdasarkan SK Menteri Kesehatan Nomor:269/Menkes/PER/III/2008 tentang rekam medis menjelaskan bahwa rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien (Bab I pasal 1).

Rekam medis yang memuat informasi evaluasi keadaan fisik dan riwayat penyakit pasien amat penting dalam perencanaan dan koordinasi pelayanan pasien, bagi evaluasi lanjut serta menjamin kontinuitas pelayanan yang diberikan. Oleh karena itu kelengkapan, keakuratan dan ketepatan waktu pengisian harus diupayakan dalam organisasi kesehatan karena amat penting bagi kelayakan tindakan pelayanan dan rujukan.

Electronic Health Record selanjutnya disebut *EHR* bukanlah sistem informasi yang dapat dibeli dan diinstall seperti paket *word-processing* atau sistem informasi pembayaran dan laboratorium yang secara langsung dapat dihubungkan dengan sistem informasi lain dan alat yang sesuai dalam lingkungan tertentu. *EHR* merupakan sistem informasi yang memiliki *framework* lebih luas dan memenuhi satu set fungsi, menurut Amatayakul Magret K dalam bukunya *Electronic Health Records: A Practical, Guide for Professionals and Organizations* harus memenuhi kriteria sebagai berikut:

1. Mengintegrasikan data dari berbagai sumber (*Integrated data from multiple source*).
2. Mengumpulkan data pada titik pelayanan (*Capture data at the point of care*).
3. Mendukung pemberi pelayanan dalam pengambilan keputusan (*Support caregiver decision making*).

Sedangkan Gemala Hatta menjelaskan bahwa *EHR* terdapat dalam sistem yang secara khusus dirancang untuk mendukung pengguna dengan berbagai kemudahan fasilitas untuk kelengkapan dan keakuratan data; memberi tanda waspada; peringatan; memiliki sistem untuk mendukung keputusan klinik dan menghubungkan data dengan pengetahuan medis serta alat bantu lainnya.

2.5. Protégé

Protégé adalah perangkat lunak bantu yang digunakan untuk pengembangan sistem *Knowledge Base System*. Aplikasi yang dikembangkan oleh Protégé digunakan dalam pemecahan masalah dan pembuatan keputusan dalam sebuah domain. Protégé dikembangkan oleh sebuah organisasi yang bernaung di bawah *Stanford University*, yang mengambil spesialisasi di bidang ontology.

Perangkat lunak Protégé menyediakan konsepsi dasar pengetahuan yang terintegrasi, serta mengubah tampilan visual lingkungan dengan memperluas arsitektur sistem untuk membuat pemodelan dasar pengetahuan secara lebih sederhana dan mudah. Protégé dapat juga digunakan dengan tujuan berikut; (1) membangun ontology, (2) memodelkan tampilan pengetahuan akuisisi dan (3) memasukan domain pengetahuan. Protégé memvisualisasikan hubungan *subclass* dalam *tree*, mendukung berbagai penurunan (*multiple inheritance*) dan *root* pada hirarki *class* yang terbentuk adalah *class "THING"*.

Untuk mendapatkan Protégé dapat dilakukan dengan cara mengunduh dari *web* penyedia tool di alamat <http://protege.stanford.edu/>. Ukuran file instalasi Protégé tergantung dari versi Protégé yang diinginkan dan termasuk atau tidaknya SDK java dalam program instalasi Protégé. Protégé dapat berjalan diberbagai platform sistem operasi, antara lain *Windows, Mac OS, Solaris, Linux, HP-UX, Unix*, dan *AIX*.

2.6. Yii Framework

Yii adalah *framework* (kerangka kerja) PHP berbasis-komponen, berkinerja tinggi untuk pengembangan aplikasi *Web* berskala-besar. Yii menyediakan *reusability* maksimum dalam pemrograman *Web* dan mampu meningkatkan kecepatan pengembangan aplikasi secara signifikan. Nama Yii (dieja sebagai /i:/) singkatan dari "Yes It Is!". Yii pertama kali dikembangkan oleh seorang master bernama Qiang Xue pada Januari 2008. Pada bulan Oktober 2008 setelah sepuluh bulan pembangunan, versi *alpha* Yii pertama dirilis. Pada tanggal 3 Desember 2008, Yii 1.0 secara resmi dirilis ke publik.

Untuk menjalankan aplikasi *Web* berbasis *Yii Framework*, pengguna memerlukan *server Web* yang mendukung PHP 5.1.0. Bagi para pengembang yang ingin menggunakan *Yii*, memahami pemrograman berorientasi-objek (OOP) akan sangat membantu, karena *Yii* merupakan *framework* OOP murni. *Yii* adalah *framework* pemrograman umum *Web* yang bisa dipakai untuk mengembangkan semua jenis aplikasi *Web*. Dikarenakan sangat ringan dan dilengkapi dengan mekanisme *caching* yang canggih, *Yii* sangat cocok untuk pengembangan aplikasi dengan lalu lintas-tinggi, seperti portal, forum, sistem manajemen konten (*CMS, Content Management System*), sistem *e-commerce*, dan lain-lain.

Seperti kebanyakan PHP *framework*, *Yii* adalah Model View Controller (MVC) *framework*. *Yii* melampaui *framework* PHP lain dalam hal efisiensi, kekayaan-fitur, dan kejelasan dokumentasi. *Yii* didesain dengan hati-hati dari awal agar sesuai untuk pengembangan aplikasi *Web* secara serius. *Yii* bukan berasal dari produk pada beberapa proyek maupun konglomerasi pekerjaan pihak-ketiga. *Yii* adalah hasil dari pengalaman kaya para pembuat pada pengembangan aplikasi *Web* dan investigasi *framework* pemrograman *Web* dan aplikasi yang paling populer.

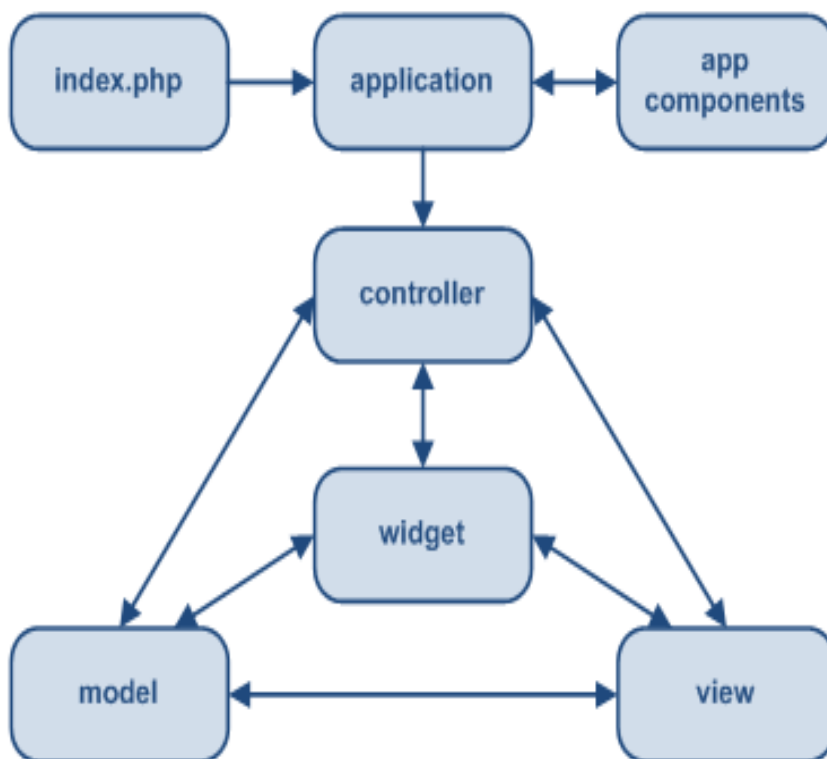
Performa kinerja *Yii Framework* jauh lebih cepat karena menggunakan teknik *lazy loading ekstensif*. Misalnya, tidak menyertakan file kelas sampai kelas digunakan untuk pertama kalinya, dan tidak menciptakan objek sampai objek tersebut diakses untuk pertama kalinya. *Framework* lainnya memiliki kinerja berat karena memungkinkan fungsi (misalnya koneksi database, sesi pengguna) terus digunakan walaupun tidak selama permintaan.

2.6.1. Model View Controller

Yii mengimplementasikan pola desain Model-View-Controller (MVC), yang diadopsi secara luas dalam pemrograman *Web*. MVC bertujuan untuk memisahkan logika bisnis dari pertimbangan antar muka pengguna agar para pengembang bisa lebih mudah mengubah setiap bagian tanpa mempengaruhi yang lain. Dalam MVC, model menggambarkan informasi (data) dan aturan

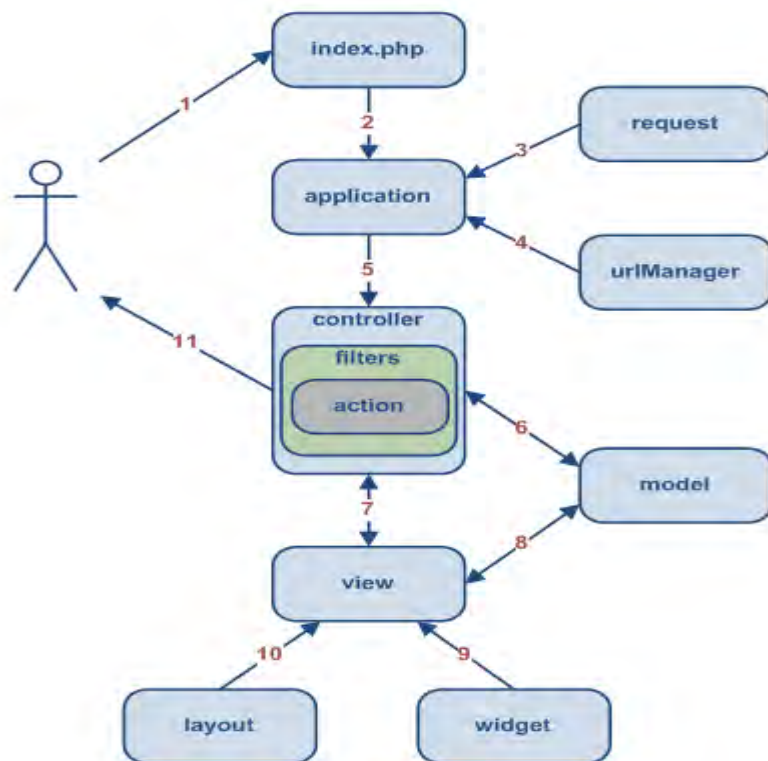
bisnis; *view*(tampilan) berisi elemen antar muka pengguna seperti teks, input *form*; sementara *controller* mengatur komunikasi antar model dan *view*.

Selain implementasi MVC, *Yii* juga memperkenalkan *front-controller*(kontroler-depan), yang disebut Aplikasi, yang mengenkapsulasi konteks eksekusi untuk memproses sebuah *request*. Aplikasi mengumpulkan beberapa informasi mengenai request pengguna dan kemudian mengirimnya ke *controller* yang sesuai untuk penanganan selanjutnya. Diagram berikut memperlihatkan struktur statis sebuah aplikasi *Yii*:



Gambar 2.6. Struktur Statis *Yii*

Diagram berikut memperlihatkan alur kerja umum sebuah aplikasi *Yii* saat menangani permintaan pengguna:



Gambar 2.7. Alur Kerja Umum Aplikasi Yii

1. Pengguna membuat permintaan dengan URL <http://www.example.com/index.php?r=post/show&id=1> dan *server Web* menangani permintaan dengan menjalankan skrip *bootstrapindex.php*.
2. Skrip *bootstrap* membuat sebuah instan Aplikasi dan menjalankannya.
3. Aplikasi mendapatkan rincian informasi permintaan pengguna dari komponen aplikasi bernama *request*.
4. Aplikasi menentukan *controller* dan aksi yang diminta dengan bantuan komponen aplikasi bernama *urlManager*. Dalam contoh ini, *controller* adalah *post* yang merujuk pada kelas *PostController*; dan aksi adalah *show* yang arti sebenarnya ditentukan oleh *controller*.
5. Aplikasi membuat *instance controller* yang diminta untuk selanjutnya menangani permintaan pengguna. *Controller* menentukan aksi *show* merujuk pada sebuah metode bernama *actionShow* dalam kelas *controller*. Kemudian membuat dan menjalankan filter (contoh kontrol

akses, pengukuran) terkait dengan aksi ini. Aksi dijalankan jika diijinkan oleh filter.

6. Aksi membaca *Post* model di mana ID adalah 1 dari database.
7. Aksi menyiapkan *view* (tampilan) bernama *show* dengan model *Post*.
8. *View* membaca dan menampilkan atribut model *Post*.
9. *View* menjalankan beberapa *widget*.
10. *View* menyiapkan hasil yang dipasangkan dalam *layout*(tata letak).
11. Aksi mengakhiri pembuatan *view* dan menampilkan hasil akhir kepada pengguna.

2.6.2. Fitur *Yii Framework*

Berikut fitur-fitur utama yang terdapat di *Yii Framework*:

1. Design pola *Model-View-Controller* (MVC). *Yii* mengadopsi arsitektur MVC, yang memungkinkan untuk pemisahan perhatian.
2. *Database Access Objects* (DAO), *Query Builder*, *Active Record*, *DB Migration*. *Yii* memungkinkan pengembang untuk melakukan model data database dalam hal objek dan menghindari kebosanan dan kompleksitas menulis pernyataan SQL berulang.
3. *Form input* dan validasi. *Yii* mengumpulkan *form input* sangat mudah dan aman. Muncul dengan satu *set validator* serta *helper methods* dan *widget* untuk menyederhanakan tugas untuk input bentuk dan validasi.
4. *AJAX-enabled widget*. Terintegrasi dengan *jQuery*, *Yii* datang dengan satu set *AJAX-enabled widget*, seperti *input field*, *treeview*, *data grid*, yang memungkinkan menulis antarmuka pengguna yang sangat efisien, serbaguna dan sangat sederhana.
5. Otentikasi dan otorisasi. *Yii* memiliki *built-in* dukungan otentikasi. Ini mendukung otorisasi melalui hirarkis *role-based* akses kontrol.
6. Layanan *Web*. *Yii* mendukung generasi otomatis layanan spesifikasi kompleks WSDL dan manajemen penanganan permintaan layanan *Web*.
7. Internasionalisasi (I18N) dan lokalisasi (L10N). *Yii* mendukung terjemahan pesan, tanggal dan waktu *format*, *format* angka, dan lokalisasi antarmuka.

8. *Layered caching* skema. *Yii* mendukung *caching* data, *cache* halaman, *cache* fragmen dan konten dinamis. Media penyimpanan *caching* dapat diubah dengan mudah tanpa menyentuh kode aplikasi.
9. Penanganan dan penebangan kesalahan. Kesalahan akan ditangani dan disajikan lebih baik, dan pesan log dapat dikategorikan, disaring dan disalurkan ke berbagai tujuan.
10. Keamanan. *Yii* dilengkapi dengan langkah-langkah keamanan untuk membantu mencegah aplikasi *Web* Anda dari serangan seperti *SQL injection*, *cross-site scripting* (XSS), *cross-site* pemalsuan permintaan (CSRF), dan *cookie* gangguan.
11. Unit dan pengujian fungsionalitas. *Yii* menyediakan dukungan untuk menulis dan menjalankan unit test serta tes fungsi, berdasarkan PHP Unit dan Selenium.
12. Otomatis menggenerasi kode. *Yii* menyediakan satu set alat kode intuitif dan *extensible code generation* yang dapat membantu pengguna lebih cepat menghasilkan kode yang dibutuhkan untuk fitur seperti *form* input, CRUD.
13. Kepatuhan ke XHTML. Kode HTML yang dihasilkan oleh komponen *Yii* dan generator kode sesuai dengan standar XHTML.
14. Murni berorientasi objek. *Yii framework* menggunakan paradigma OOP. *Yii* tidak mendefinisikan fungsi global atau variabel. Dan hirarki kelas yang didefinisikan memungkinkan usabilitas maksimum dan kustomisasi.
15. Ramah dengan kode pihak ketiga. *Yii* dirancang untuk bekerja dengan baik dengan kode pihak ketiga. Sebagai contoh, pengguna dapat menggunakan kode dari PEAR atau Zend *Framework* dalam aplikasi *Yii*.

2.6.3. Bekerja dengan *Form*

Pengumpulan data pengguna via *form* HTML adalah tugas utama dalam pengembangan aplikasi *Web*. Selain mendesain *form*, pengembang perlu mempopulasi *form* dengan data yang sudah ada atau nilai-nilai standar, memvalidasi input pengguna, menampilkan pesan kesalahan yang sesuai untuk input yang tidak benar, dan menyimpan input ke media penyimpan. *Yii*

sudah menyederhanakan alur kerja ini dengan arsitektur MVC. Langkah-langkah berikut umumnya diperlukan saat berhadapan dengan *form* dalam *Yii*:

1. Buat kelas model yang mewakili *field* data yang dikumpulkan.
2. Buat aksi *controller* dengan kode yang merespon pengiriman *form*.
3. Buat *form* dalam file skrip tampilan sesuai dengan aksi *controller*.

2.6.4. Bekerja dengan Database

Yii menyediakan dukungan berkemampuan untuk pemrograman database. Dibangun di atas extension PHP *Data Objects* (PDO), *Yii Data Access Objects* (DAO) memungkinkan pengaksesan ke sistem manajemen database (DBMS) yang berbeda dalam satu antar muka tunggal yang seragam. Aplikasi yang dikembangkan menggunakan *Yii* DAO dapat dialihkan dengan mudah ke DBMS berbeda tanpa perlu memodifikasi data pengaksesan code.

Yii Query Builder menyediakan sebuah metode berorientasi objek untuk membuat query SQL, yang bisa mengurangi resiko terserang *SQL injection*. Dan *Active Record Yii*, diimplementasikan sebagai pendekatan Pemetaan Relasional-Obyek / *Object-Relational Mapping* (ORM) yang diadopsi secara luas, mempermudah pemrograman database. Tabel direpresentasikan dalam bentuk kelas dan baris dalam bentuk *instance*, *Yii* AR mengeliminasi tugas berulang pada penulisan *SQL statement* terutama yang berkaitan dengan operasi CRUD (*create, read, update* dan *delete*).

Meskipun *Yii* menyertakan fitur-fitur database yang dapat menangani hampir semua tugas-tugas terkait-database, Anda masih bisa menggunakan pustaka database Anda sendiri dalam aplikasi *Yii* Anda. Bahkan, *Yii framework* didesain secara hati-hati agar bisa dipakai bersamaan dengan pustaka pihak ketiga lainnya.

2.7. WampServer

WampServer merupakan sebuah aplikasi yang dapat menjadikan komputer maupun laptop anda menjadi sebuah *server* atau bias dikatakan *server offline*. *WampServer* adalah singkatan dari *Windows, Apache, MySQL* dan *PHP*.

WampServer adalah aplikasi yang menggabungkan antara *Apache*, *MySQL*, dan *PHP*. Kegunaan dari *WampServer* untuk membuat jaringan local sendiri dalam artian pengguna dapat membuat *website* secara *offline*. *WampServer* hanya bisa digunakan untuk sistem operasi *Windows* saja, jadi untuk pemakai *linux* tidak bisa memakai aplikasi ini.

Sampai saat ini *WampServer* 5 merupakan versi yang terbaru. Terdapat beberapa kelebihan dari *WampServer* dibandingkan dengan *server offline* lainnya termasuk juga *Xampp* adalah sebagai berikut :

1. Proses instalasi yang cepat,
2. Sangat mudah untuk digunakan dan tidak membuat orang kebingungan,
3. Konfigurasi *PHP* dan *Apache* langsung melalui menu,
4. "*Online*" dan "*offline*" mode.

Jadi pengguna tidak perlu menghidupkan *Php* dan *MySQL* satu persatu. Hanya dengan klik icon *WampServer* 5 saja maka *PHP* dan *MySQL* secara otomatis sudah berjalan.

Aplikasi berbasis *web* selalu menggunakan *web server* untuk operasionalnya. Juga harus menjalankan *PHP* jika memang aplikasinya dibangun dari *PHP*. Dan yang hampir pasti, aplikasi masa kini selalu membutuhkan database untuk penyimpanan datanya, biasanya *PHP* selalu dipasangkan dengan *MySQL*.

2.7.1. Apache

Apache adalah sebuah nama *web server* yang bertanggung jawab pada *request-response HTTP*. Selain itu, *Apache* juga diartikan sebagai suatu *web server* yang kompak, modular, mengikuti standar protokol *HTTP*. *Apache* memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (*GUI*) yang memungkinkan penanganan *server* menjadi mudah. *Apache* merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

2.7.2. MySQL

MySQL adalah sebuah aplikasi database guna menyimpan data-data yang akan disimpan. *MySQL* merupakan aplikasi *database server*. *SQL* merupakan kepanjangan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang digunakan untuk mengolah database. *MySQL* dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan *MySQL* untuk mengelola database.

2.7.3. PHP

PHP adalah bahasa pemrograman yang paling banyak digunakan untuk membuat sebuah *website*. PHP atau *Hypertext Preprocessor* merupakan bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memprogram situs *web* dinamis. PHP dapat digunakan untuk membangun sebuah CMS. Beberapa kelebihan PHP dari bahasa pemrograman *web* yaitu PHP tidak melakukan kompilasi dalam pembangunannya dan *web server* yang mendukung PHP banyak ditemukan seperti *Apache*. Dalam sisi pengembangan lebih mudah karena banyak developer yang terlibat, dan dalam sisi pemahaman, PHP merupakan bahasa scripting yang paling mudah digunakan karena memiliki referensi yang banyak. Selain itu, PHP bersifat *open source* yang dapat digunakan di berbagai mesin dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.8. UML

UML atau *Unified Modelling Language* adalah sebuah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak.

2.8.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem dan

bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Berbagai lambang di *use case* diagram, diantaranya :

a. Aktor

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

b. *Case*

Menggambarkan diskripsi kegiatan yang melibatkan actor.

c. *Extend*

Relasi yang digunakan jika *use case* yang satu mirip dengan *use case* yang lain.

d. *Include*

Relasi jika terdapat perilaku yang mirip dengan beberapa *use case*.

2.8.2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan object serta hubungan satu sama lain seperti containment, pewarisan, asosiasi, lain-lain. *Class diagram* terdiri dari relasi beberapa *class*, dalam *class* itu sendiri terdiri dari *atributte* dan *operation* yang menggambarkan keadaan suatu sistem juga menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

2.9. OpenEHR

OpenEHR merupakan spesifikasi standar terbuka dalam informatika kesehatan yang menggambarkan manajemen dan penyimpanan, pencarian dan pertukaran data kesehatan dalam rekam medis elektronik. Dalam *OpenEHR*, semua data kesehatan untuk seseorang disimpan dalam satu riwayat hidup, vendor bersifat independen, dan berpusat pada satu orang. Spesifikasi

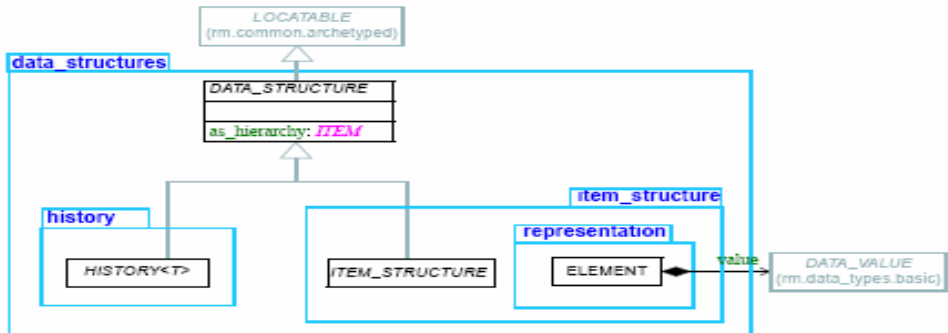
OpenEHR mencakup informasi dan model layanan untuk rekam medis elektronik, demografi, alur kerja klinis dan *archetype*.

2.9.1. Model Informasi Struktur Data (*Data Structure Information Model*)

Prinsip desain yang utama untuk model struktur data adalah kebutuhan memberikan spesifikasi yang jelas pada struktur logis dengan menggunakan representasi generik yang sama seperti hirarki. Struktur logis termasuk tabel, daftar, rangkai pohon, dan konsep sejarah. Representasi dari struktur logis itu dapat sebagai representasi murni atau sebuah arti semantik, hal ini menjadi penting, karena (1) Untuk proses interoperabilitas sistem, sebagai contoh struktur daftar tabel logis dan daftar sejarah dikodekan kedalam representasi generik yang sama oleh pengirim dan penerima informasi. (2) Memudahkan pengembang perangkat lunak secara eksplisit mengambil struktur logis sebagai antarmuka fungsional yang universal. (3) Memudahkan proses informasi oleh penerima menggunakan antarmuka fungsional dengan penyajian data yang sesuai tujuan dan data di layar komputer pemakai yang sama.

Salah satu motivasi mendefinisikan struktur data logis secara eksplisit adalah untuk menghilangkan ambiguitas dalam merekam struktur dan waktu, dengan mengacu pada standar spesifikasi *CEN 13606*, *GEHR*, *GEHR* Australia dan *HL v3*. Pendekatan standard yang menghapus semua ambiguitas tersebut akan meningkatkan kualitas data dan perangkat lunak [*OpenEHR Release 1.0.1, Data Structure Information Model*, 2006].

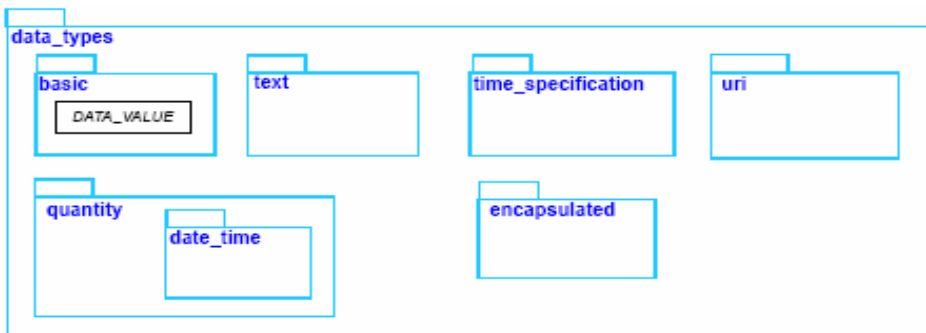
Struktur data dibentuk oleh jejak sejarah dan struktur item dimana struktur item merepresentasikan elemen-elemen yang berisi tipe data awal. Struktur data mengacu pada tabel local. Paket struktur data sendiri berisi kelas struktur data tunggal yang merupakan asal mula dari semua data struktur. Pada penelitian ini, struktur data tersebut menggunakan fitur *OpenEHR* sebagai satu fungsi hirarki yang diimplementasikan oleh masing-masing sub struktur data. Hal ini dikarenakan *OpenEHR* mendukung kompatibel sistem dengan sistem lainnya sebagai standar interoperabilitas yang mendukung standar *CEN EN 13606*.



Gambar 2.8. Paket Struktur Data *OpenEHR* [*OpenEHR Release 1.0.1, Data Structure Information Model, 2006*]

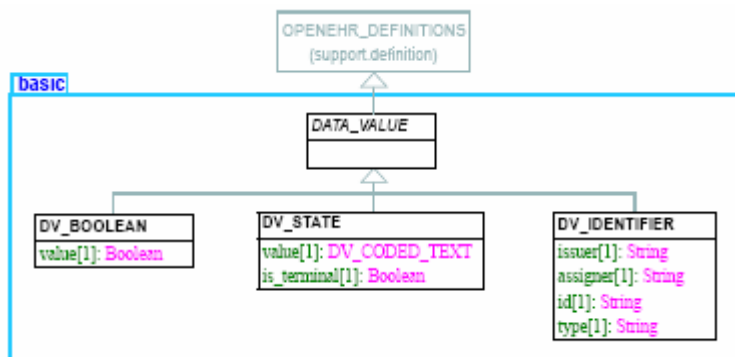
2.9.2. Model Informasi Tipe Data (*Data Type Information Model*)

Spesifikasi jenis data yang disajikan pada penelitian kami mendefinisikan jenis data klinis dan ilmiah seperti dalam model *OpenEHR* yang disesuaikan dengan kondisi nyata Indonesia. Harmonisasi tipe data antar model informasi yang digunakan oleh setiap instansi layanan kesehatan terkait, merupakan hal penting untuk mengurangi pekerjaan konversi dan potensi kesalahan penafsiran antar bagian atau instansi layanan kesehatan [*OpenEHR Release 1.0.1, Data Type Information Model, 2007*]. Tipe data yang kami bangun bersifat universal dan dapat juga dipergunakan oleh model informasi lainnya, seperti yang didefinisikan oleh *OpenEHR*, sebagai contoh model demografis dan terminologis. Jenis tipe data ini merupakan turunan dari tipe data model referensi yang digunakan oleh *GEHR*, *Sinapsis* dan *SynEX*, *CEN* dan *HL7v3*.



Gambar 2.9. Tipe Data Paket *Basic OpenEHR* [*OpenEHR Release 1.0.1, Data Type Information Model, 2007*]

Tipe data dikelompokkan menjadi enam yaitu; 1. *Basic*, 2. *Text*, 3. Spesifikasi waktu, 4. URL, 5. Kuantiti termasuk didalamnya data waktu dan 6. Kapsulasi data. Semua tipe data itu mengacu pada standar tipe data yang disajikan oleh *OpenEHR*.

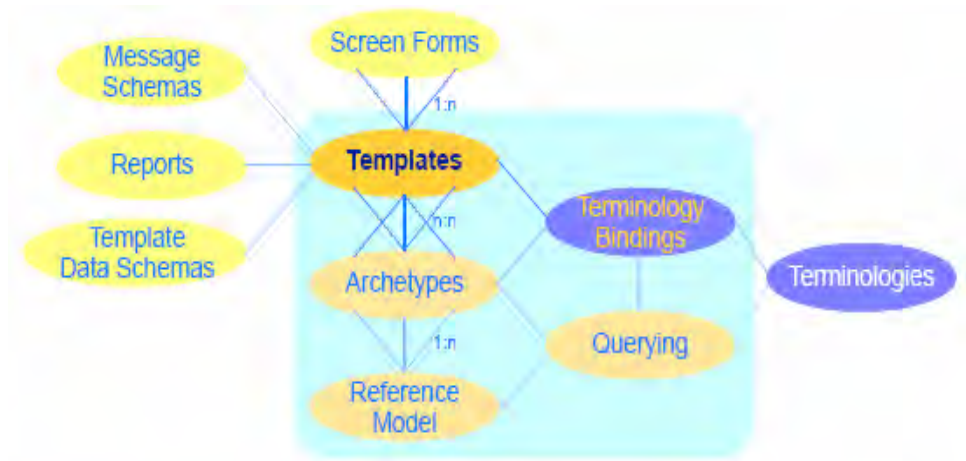


Gambar 2.10. Tipe Data *Basic OpenEHR* [*OpenEHR Release 1.0.1, Data Type Information Model, 2007*]

Gambar 2.10 memperlihatkan tipe data basic dimana isi atau *value* dari tipe data tersebut adalah, *Boolean*, *State* dan *Identifier*. Ketiga *data value* tersebut didefinisikan berdasarkan kondisi *real* sistem. Tipe data akan mewakili data master dari seorang pasien. Sebagai contoh penggunaan tipe data boolean pada pilihan jenis pembayaran [Y/T]= [1/0]. Untuk contoh data value identifier string seperti, Nama pasien: xxxx yyyy, Alamat pasien: xxxxx yyyyyy zzzzz. Pemilihan *data value string* untuk memudahkan pengisian dan pembatasan atas jumlah isian karakter. Untuk *data state*, sebagai contoh adalah tanggal lahir pasien dengan tipe yyyy/bb/tt.

2.9.3. Templet Model Objek (*Template Object Model*)

Templet Model Objek merupakan lapisan ketiga diatas *archetype* dan model referensi dalam arsitektur aplikasi *OpenEHR* seperti pada gambar 6. (Arsitektur Semantik *OpenEHR*). Arsitektur ini menyediakan sarana untuk pendefinisian kelompok *archetype* dan data untuk tujuan bisnis layanan kesehatan. Selain itu juga menyediakan terminologi khusus yang menghasilkan artefak, tipe, tampilan layar dan skema pesan.

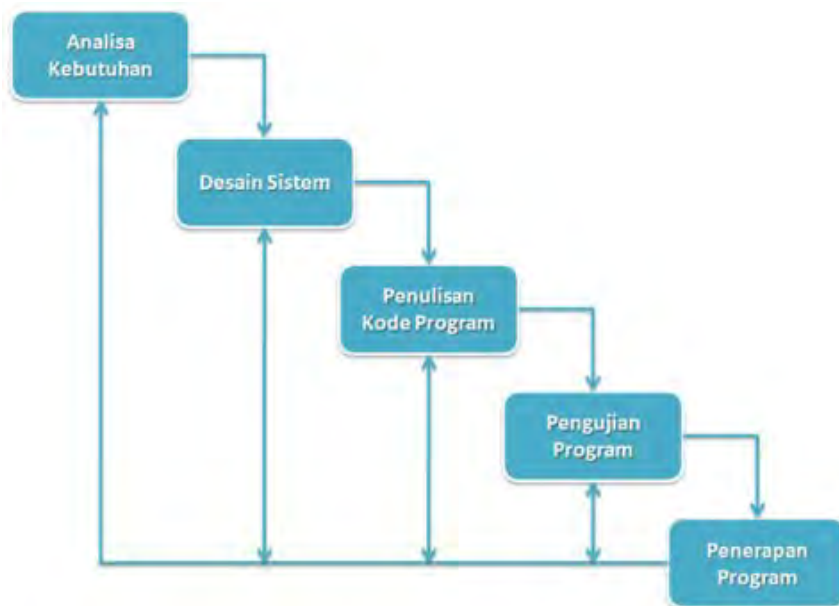


Gambar 2.10 Arsitektur Semantik *OpenEHR* [*OpenEHR Release 1.0.1, OpenEHR Template, 2012*]

Templet *OpenEHR* adalah merupakan artefak yang memungkinkan pendefinisian tertentu yang digunakan pada instansi penyelenggara layanan kesehatan. Contoh dari artefak itu adalah templet yang memungkinkan adanya interaksi antara pasien dan rumahsakit atau klinik, seorang pasien penderita diabetes mendapat terminologi 'EDdebit', sehingga memudahkan proses pencarian, dan standardisasi serta integrasi data. Templet ini juga merupakan precursor langsung ke bentuk lapisan penyaji pada layar perangkat lunak.

BAB III. METODE PENGEMBANGAN

Dalam pembuatan aplikasi *website* rekam medis elektronik berbasis *OpenEHR* ini dibutuhkan suatu metode pengembangan yang merupakan prosedur pengumpulan data yang digunakan untuk memecahkan masalah yang dihadapi. Adapun metode pengembangan yang digunakan yaitu metode pengembangan rekayasa perangkat lunak *Waterfall*. Metode ini merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *Waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah 2, 3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.



Gambar 3.1 Gambar Langkah-langkah Metode *Waterfall*

Secara garis besar metode *Waterfall* yang dilakukan mempunyai langkah-langkah yang akan dijelaskan sebagai berikut:

3.1. Analisa

Langkah analisa merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini dilakukan dengan memberikan kuesioner ke beberapa rumah sakit dan poliklinik di Kota Depok untuk mengetahui sistem yang telah berjalan. Dalam hal ini responden yang mengisi adalah pengguna atau pelaksana sistem administrasi pada puskesmas, klinik dan rumah sakit yaitu petugas pendaftaran dan dokter/asisten dokter/perawat. Untuk lebih memahami bagaimana aliran data rekam medis, maka dilakukan wawancara dengan seorang yang pernah mempelajari rekam medis yaitu dengan salah satu perawat di Rumah Sakit UKI. Setelah itu dilakukan studi literatur untuk merancang sistem pengolahan rekam medis elektronik. Studi literatur dilakukan untuk mengetahui spesifikasi *OpenEHR* tentang bagaimana mengklasifikasikan data rekam medis. Selain itu studi literatur juga dilakukan untuk mencari, mempelajari dan merangkum berbagai macam pustaka yang berkaitan dengan rumusan masalah, teori-teori yang berkaitan dengan pembuatan *website* rekam medis elektronik seperti *website*, rekam medis elektronik, *Yii Framework*, *WampServer*, *uses case diagram*, *ontograf*, *class diagram* dan struktur navigasi. Tahapan analisis ini akan menjadi acuan untuk menterjemahkan ke dalam bahasa pemrograman.

3.2. Desain

Proses desain dapat digunakan untuk menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Pada tahap desain dilakukan perancangan dari sistem baru yang akan dibuat. Untuk mempermudah dalam penggambaran sistem yang akan dibuat, maka dirancang *uses case diagram*, *ontograf*, *class diagram* dan struktur navigasi. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor-aktor dengan sistem website rekam medis yaitu admin, pasien, dokter, staff, dan laboran. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dibuat untuk

melihat keterhubungan antar tabel satu dengan tabel lainnya di database. Dengan *class diagram* dapat dilihat tabel mana yang berdiri sendiri, dan tabel mana yang merupakan referensi dari tabel lain. Sedangkan *Ontograf* merupakan pemodelan *ontologi* untuk menggambarkan konsepsi dasar pengetahuan yang terintegrasi. Dengan *Ontograf* dapat dilihat hubungan sub *class* dalam *tree*, mendukung berbagai penurunan (*multiple inheritance*) dan *root* pada hirarki *class* yang terbentuk yaitu *class "THING"*. *Ontograf* dibuat untuk melihat turunan dari *class* pasien. *Class* turunan ini merupakan tabel referensi dari tabel pasien dan tabel-tabel yang berhubungan dengan pasien, seperti kartu pasien, hasil pemeriksaan dokter, hasil laboratorium, dan transaksi.

Selanjutnya dilakukan perancangan struktur navigasi dari sistem yang akan dibuat. Struktur navigasi dapat digunakan untuk menggambarkan alur yang digunakan dari aplikasi yang dibuat. Struktur navigasi yang dibuat adalah struktur navigasi campuran yaitu gabungan dari struktur navigasi linear, non linear, dan hirarki. Setelah itu, barulah dapat dilakukan perancangan halaman pada *website* yang akan dibangun. Arsitektur perangkat lunak yang digunakan dalam membangun *website* ini adalah :

- Sistem Operasi *Windows Seven Ultimate*
- *WampServer* 5 dengan spesifikasi *Apache* 2.2.4, *PHP* versi 5.2.2, dan *MySQL* versi 5.0.37
- *Yii Framework* versi 1.1.10

3.3. Coding dan Testing

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh user. Tahapan ini merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Pengkodean dilakukan dengan menggunakan *Yii Framework*. Pengkodean dilakukan berdasarkan perancangan yang telah dilakukan sebelumnya. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. *Testing*

dilakukan dengan menggunakan *website* server *WampServer* untuk menguji program yang dibuat. Program ini diuji di *localhost* terlebih dahulu. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

3.4. Penerapan

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh pengguna. Agar *website* ini dapat digunakan oleh pengguna, maka dilakukan *hosting* dari *website* yang sebelumnya telah dilakukan *testing* di *localhost*.

3.5. Pemeliharaan

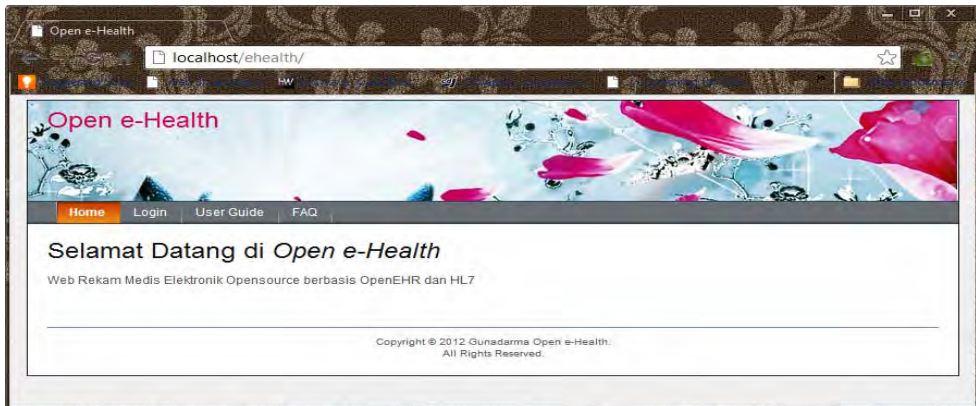
Website yang sudah di-*hosting* dan telah digunakan *user* dimasa mendatang pasti akan mengalami perubahan. Perubahan tersebut bisa terjadi karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan seperti hadirnya perangkat baru atau sistem operasi baru atau karena user membutuhkan perkembangan fungsional. Oleh karena itu, *website* yang sudah dibuat harus dilakukan pemeliharaan secara berkala.

BAB IV. PANDUAN APLIKASI OPELT GUNASYS

Bab berikut menyajikan panduan penggunaan Aplikasi Opelt Gunasys. Untuk pengembangan rinci dari aplikasi dapat dilihat pada [R.A. Miharja 2012]

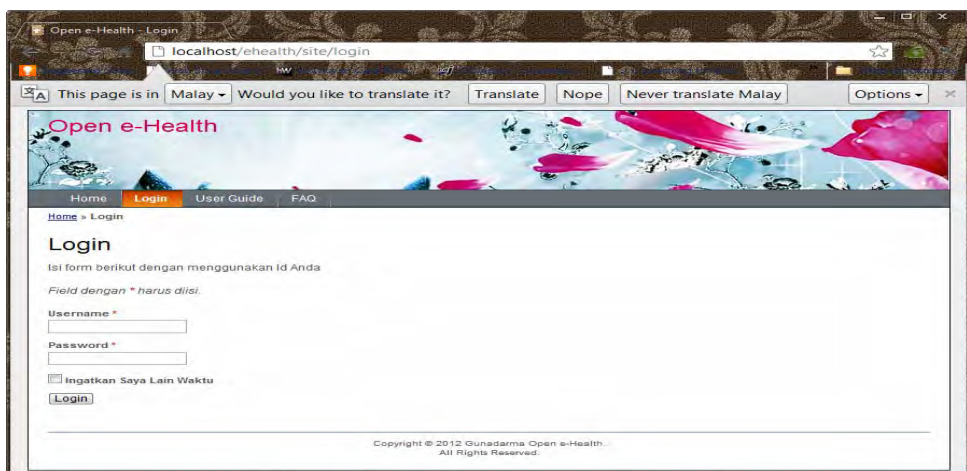
4.1. Pedoman Untuk Publik

4.1.1. Halaman Home



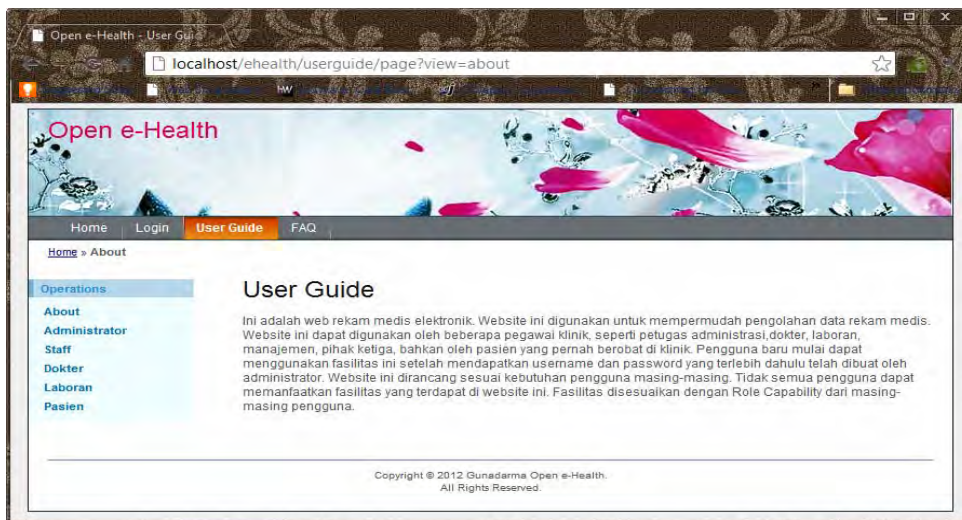
Halaman Home merupakan halaman awal yang akan muncul ketika pengguna pertama kali mengakses situs dan berisi sedikit penjelasan mengenai situs Open E-health.

4.1.2. Halaman Login



Halaman Login merupakan halaman yang dapat dimanfaatkan pengguna untuk melakukan login ke sistem. Ada beberapa pengguna yang dapat melakukan login ke situs ini, yaitu administrator, petugas administrasi (staff), dokter, penunjang medis (laboran), dan pasien. Pengguna dapat melakukan login ke situs ini dengan username dan password yang sebelumnya telah diberi oleh administrator untuk pengguna internal, dan oleh staff untuk pasien.

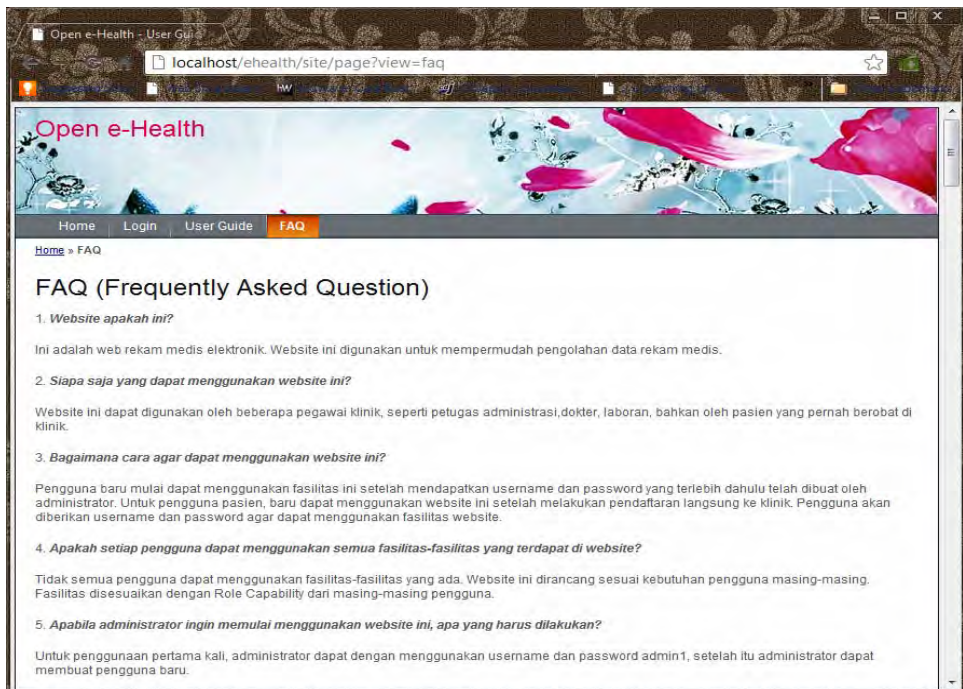
4.1.3. Halaman User Guide



Halaman User Guide merupakan halaman berisi panduan untuk menggunakan website dari pengguna website. Terdapat keterangan tentang situs ini dan disediakan panduan untuk menggunakan website berdasarkan peran pengguna situs, apakah sebagai administrator, staff, dokter, laboran, atau sebagai pasien.

4.1.4. Halaman FAQ

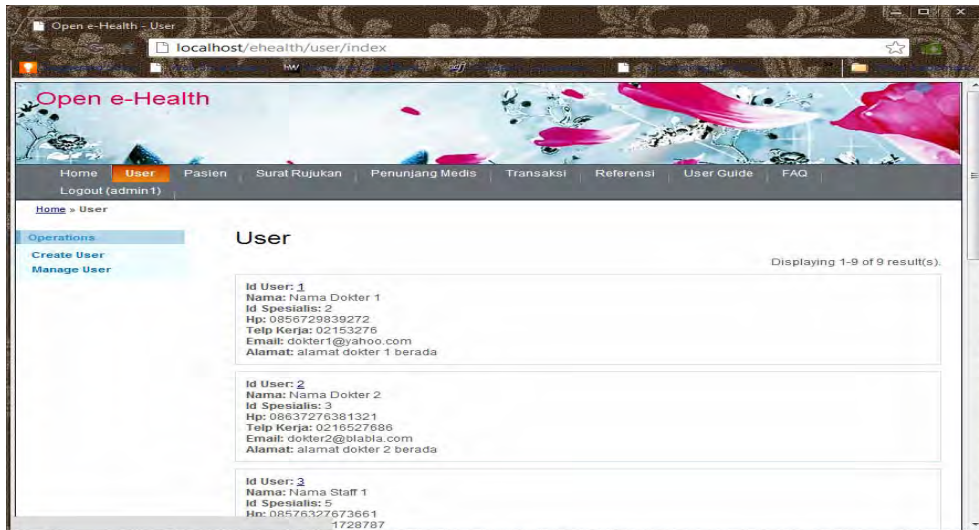
Halaman FAQ (Frequently Asked Question) merupakan halaman yang berisi pertanyaan-pertanyaan yang sering ditanyakan untuk mengoperasikan situs ini. Pertanyaan-pertanyaan yang disediakan seperti tentang apa situs ini, siapa saja yang dapat menggunakan situs, cara penggunaan situs untuk masing-masing pengguna, dan istilah yang digunakan di situs ini.



4.2. Pedoman Untuk Registrasi Pemakai

Ada beberapa pengguna yang dapat melakukan login ke situs Opelt Gunasys yaitu pengguna internal dan pasien. Pengguna internal merupakan pengguna yang berada di lingkungan medis dan berperan dalam pengelolaan data rekam medis. Yang termasuk pengguna internal yaitu staff, dokter, dan laboran, sedangkan administrator merupakan pengguna yang bertugas mengatur website. Untuk pengguna internal akan mendapatkan akun yang telah dibuat sebelumnya oleh admin, sedangkan pasien mendapat akun dari staff ketika pasien mendaftarkan diri di klinik. Terdapat beberapa halaman yang dapat dilihat oleh pengguna yang mempunyai akun untuk login ke situs ini. walaupun pengguna dapat melihat beberapa halaman, namun tidak semua pengguna dapat menggunakan fasilitas di halaman tersebut. Pengguna dibatasi hak aksesnya berdasarkan perannya dalam mengakses situs ini. Berikut beberapa halaman dan fasilitas-fasilitas yang dapat dilihat ketika pengguna melakukan login.

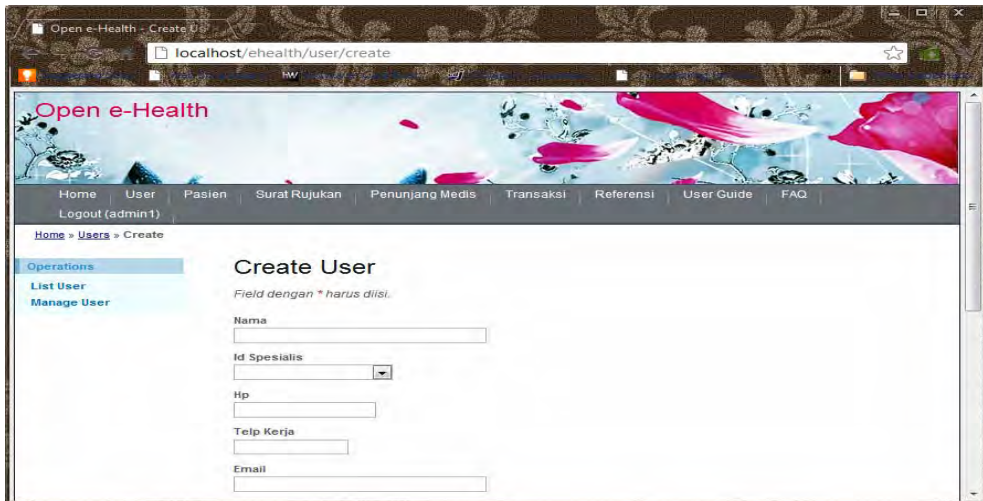
4.2.1. Halaman User



Halaman user merupakan halaman yang dapat dimanfaatkan pengguna untuk melihat daftar pengguna internal yang terdaftar di situs ini. Di halaman ini dapat dilakukan beberapa operasi seperti membuat pengguna internal baru, melihat daftar pengguna, mengubah data pengguna, dan menghapus data pengguna. Tidak semua pengguna dapat melakukan operasi-operasi tersebut.

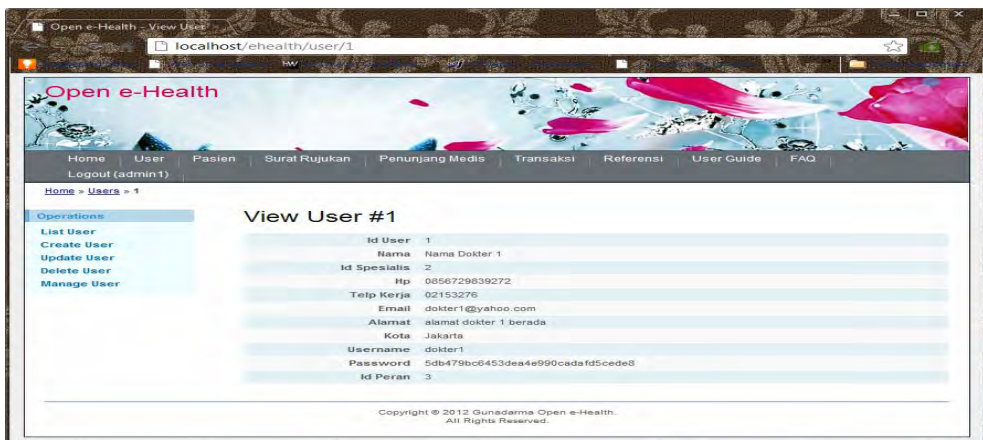
- Membuat Pengguna Baru

Untuk membuat pengguna baru, dapat diakses pilihan create user pada sidebar menu. Maka akan muncul tampilan form untuk mengisi data pengguna. Untuk id spesialis dapat dipilih sesuai spesialis dari pengguna, seperti pilihan dokter mata, dokter gigi, atau dokter umum untuk dokter. Daftar id spesialis ini dapat ditambahkan, diubah, ataupun dihapus pada halaman referensi yang akan dibahas selanjutnya. Untuk Id peran dapat dipilih sesuai peran dari pengguna yang akan diberi hak aksesnya apakah sebagai admin, staff, dokter, atau laboran. Pengguna yang dapat melakukan operasi tambah user hanya pengguna yang berperan sebagai administrator.



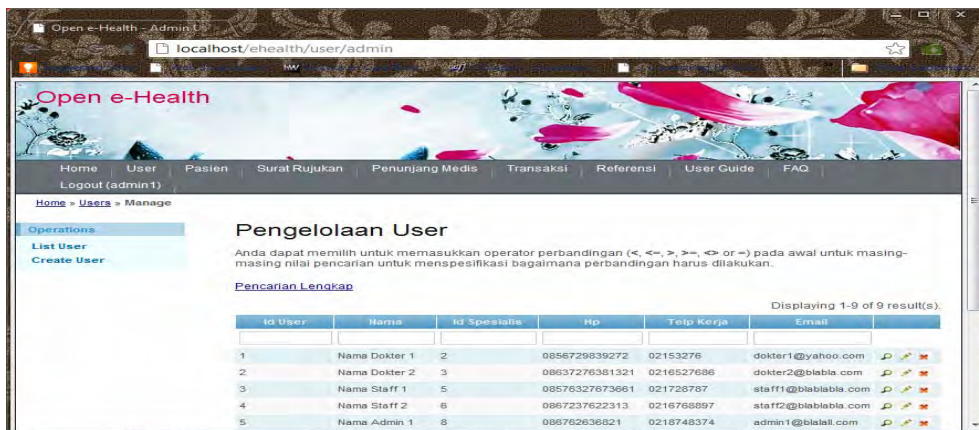
- Melihat Rincian Data Pasien

Untuk melihat data pengguna secara mendetail dapat dipilih id user yang dimaksud, maka akan muncul tampilan keterangan data pengguna secara lengkap.



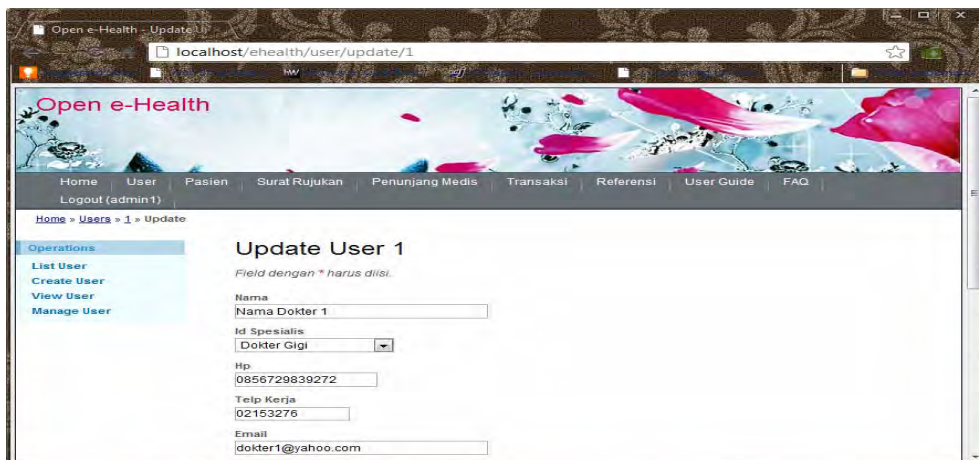
- Mengelola Data Pengguna

Untuk mengelola pengguna, dapat dipilih Manage User pada side bar menu. Dengan menggunakan menu ini dapat dilakukan pengelolaan terhadap pengguna seperti melihat daftar pengguna dalam tabel yang dapat dilakukan pencarian untuk setiap data dari pengguna. Selain itu dapat dilakukan operasi untuk menghapus, merubah, dan melihat detail data pengguna. Fasilitas ini hanya dapat digunakan oleh administrator.



- Mengubah Data Pengguna

Untuk merubah data pengguna dapat dilakukan dipilihan manage user atau dengan melihat data lengkap pengguna kemudian memilih Update User di side bar menu. Akan muncul tampilan form yang berisi data pengguna yang dapat diubah nilainya. Yang dapat melakukan perubahan data pengguna untuk halaman user hanya administrator, sedangkan pengguna lainnya dapat merubah data dirinya melalui dashboard masing-masing pengguna.

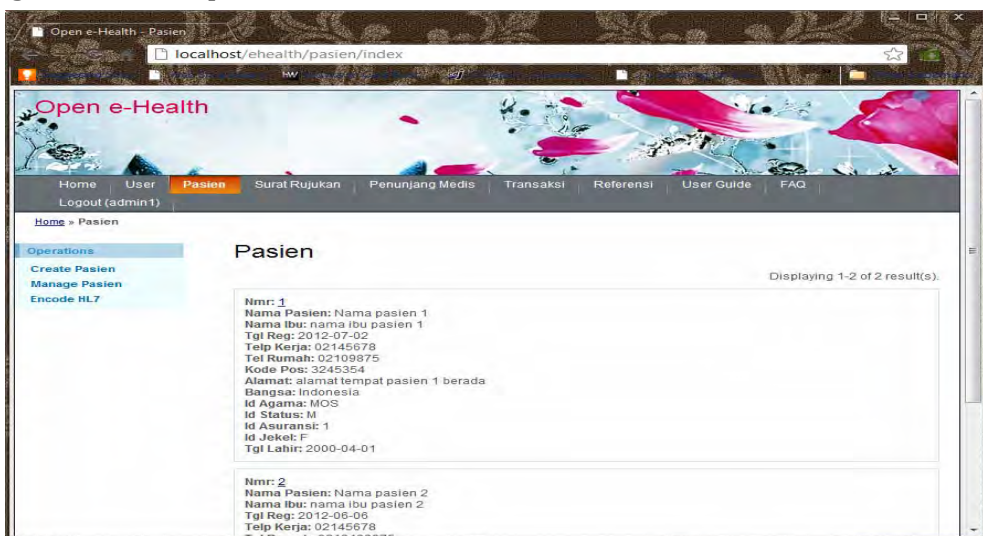


- Menghapus Pengguna

Untuk menghapus pengguna dapat dipilih Delete User pada side bar menu ketika melihat data lengkap pengguna, atau melalui menu Manage User. Untuk menghapus pengguna hanya dapat dilakukan oleh administrator.

4.2.2. Halaman Pasien

Halaman Pasien merupakan halaman yang berisi daftar pasien yang berkunjung di klinik. Pada halaman ini dapat dilakukan operasi untuk menambah, melihat, mengubah, dan menghapus data pasien. Pengguna yang dapat mengakses semua fasilitas halaman pasien adalah staff dan administrator, sedangkan pengguna lain hanya dapat melihat daftar dan rincian data pasien. Di halaman pasien ini juga terdapat kartu pasien yang merupakan representasi rekam medis. Berikut tampilan awal halaman Pasien yang berisi daftar pasien.



Kartu pasien menyimpan data-data yang berhubungan dengan pemeriksaan pasien untuk satu kali kunjungan. Kartu pasien dapat dibuat oleh staff. Untuk setiap kartu pasien, dokter dapat menuliskan hasil pemeriksaannya dan dapat melihat hasil laboratorium dari pasien yang dirujuk.

- Membuat Pasien Baru

Yang bertugas membuat pasien baru adalah staff administrasi. Untuk membuat pasien baru, dapat diakses pilihan create pasien pada sidebar menu. Maka akan muncul tampilan form untuk mengisi data pasien. Username dan password dapat diisi dengan username dan password default yang akan

diinformasikan kepada pasien. Kemudian dapat diisi data-data pasien sesuai dengan data pasien yang berkunjung

Open e-Health - Create Pasien

localhost/health/pasien/create

Open e-Health

Home | User | Pasien | Surat Rujukan | Penunjang Medis | Transaksi | Referensi | User Guide | FAQ

Logout (admin1)

Home » Pasien » Create

Operations

List Pasien

Manage Pasien

Create Pasien

Fields with * are required.

Username

Password

Nama Pasien

Nama Ibu

- Mengolah Data Pasien

Untuk mengelola pasien, dapat dipilih Manage Pasien pada side bar menu. Dengan menggunakan menu ini dapat dilakukan pengelolaan terhadap data pasien seperti melihat daftar pengguna dalam tabel yang dapat dilakukan pencarian untuk setiap data dari pasien. Selain itu dapat dilakukan operasi untuk menghapus, merubah, dan melihat detail data pengguna. Fasilitas ini hanya dapat digunakan oleh staff dan administrator.

Open e-Health - Admin

localhost/health/pasien/admin

Open e-Health

Home | User | Pasien | Surat Rujukan | Penunjang Medis | Transaksi | Referensi | User Guide | FAQ

Logout (admin1)

Home » Pasien » Manage

Operations

List Pasien

Create Pasien

Pengelolaan Pasien

Anda dapat memilih untuk memasukkan operator perbandingan (<, <=, >, >=, or =) pada awal untuk masing-masing nilai pencarian untuk menspesifikasi bagaimana perbandingan harus dilakukan.

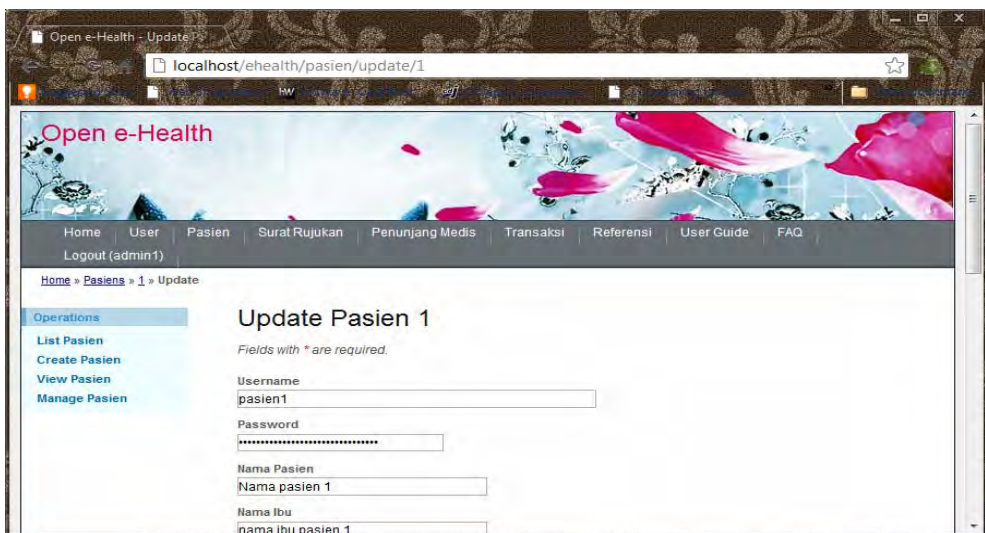
[Advanced Search](#)

Displaying 1-2 of 2 result(s).

Idmr	Nama Pasien	Tgl Lahir	Tgl Reg	Telp Kerja	Telp Rumah	Id Agama	Id Status	Id Asuransi	Id Jekel
1	Nama pasien 1	2000-04-01	2012-07-02	02145678	02109875	MOS	M	1	F
2	Nama pasien 2	2000-04-01	2012-06-06	02145678	0213489975	HIN	M	2	M

- Mengubah Data Pasien

Untuk merubah data pasien dapat dilakukan dipilihan Manage Pasien atau dengan melihat data lengkap pasien kemudian memilih Update Pasien di side bar menu. Akan muncul tampilan form yang berisi data pasien yang dapat diubah nilainya. Yang dapat melakukan perubahan data pasien untuk halaman pasien hanya staff dan administrator. Password di-encode dengan enkripsi md5 sehingga password tidak dapat terlihat.

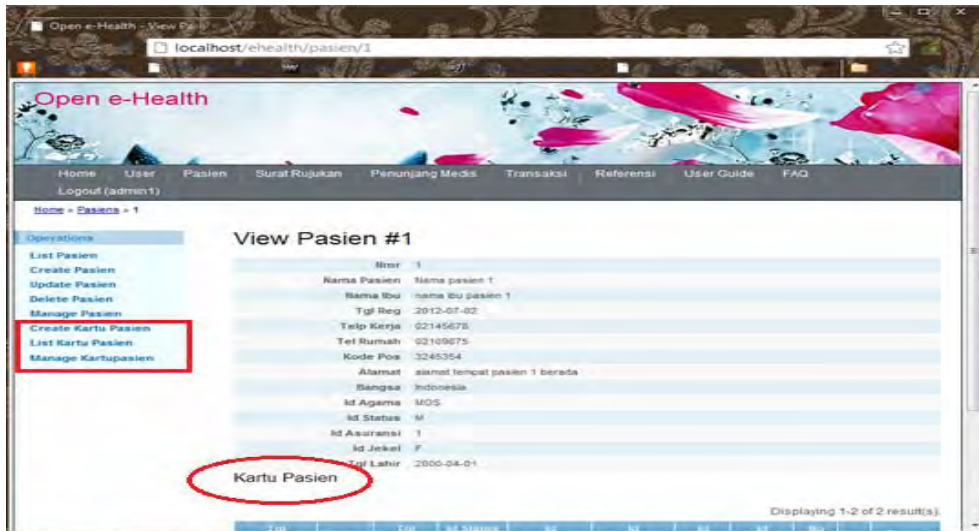


- Menghapus Pengguna

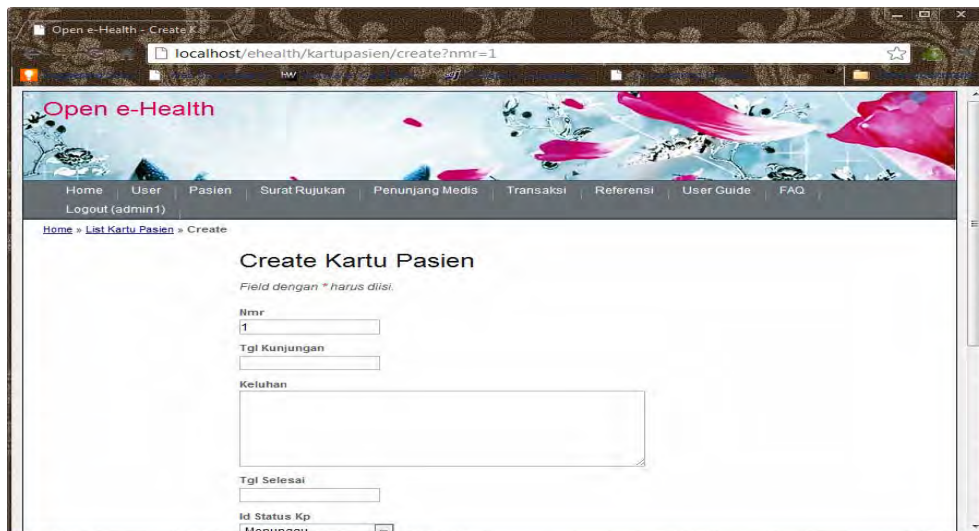
Untuk menghapus pengguna dapat dipilih Delete User pada side bar menu ketika melihat data lengkap pengguna, atau melalui menu Manage User. Untuk menghapus pengguna hanya dapat dilakukan oleh administrator.

- Melihat Rincian Data Pasien

Untuk melihat data pasien secara lengkap dapat dipilih id pasien yang dimaksud, maka akan muncul tampilan keterangan data pengguna secara lengkap. Dapat dilihat pula tabel kartu pasien kunjungan dari pasien yang dimaksud. Dari tampilan ini dapat dibuat pula kartu pasien, melihat daftar kartu pasien dari pasien yang dimaksud, dan mengelola data semua kartu pasien.



- Membuat Kartu Pasien



Setelah memilih Create Kartu Pasien, akan muncul form untuk mengisi data pada kartu pasien sesuai dengan Nomer Medikal Record (nmr) dari pasien yang akan dibuatkan kartu pasiennya. Yang dapat membuat kartu pasien adalah staff dan admin.

Untuk Id Status Kp (Kartu Pasien) dapat diisi sesuai dengan status kartu pasien. Apabila pasien masih baru mendaftar dan belum mendapatkan

penanganan maka status kartu pasien adalah menunggu. Apabila pasien telah diperiksa oleh dokter maka status kartu pasien adalah diperiksa dokter. Apabila pasien sedang mendapat pelayanan penunjang medis seperti laboratorium, status kartu pasien adalah diperiksa penunjang. Status kartu pasien berubah menjadi selesai apabila pasien sudah selesai berkunjung ke klinik pada hari berkunjung. Pada saat membuat kartu pasien, id status kartu pasien dapat dipilih menunggu.

Untuk Id penanganan, diisi sesuai dengan penanganan apa yang harus diberikan kepada pasien sesuai dengan keluhan pasien. Terdapat 3 penanganan, yaitu dokter mata, dokter umum, dan dokter gigi. Untuk menambahkan penanganan yang terdapat di klinik, dapat dilakukan di tab referensi, kemudian pilih penanganan.

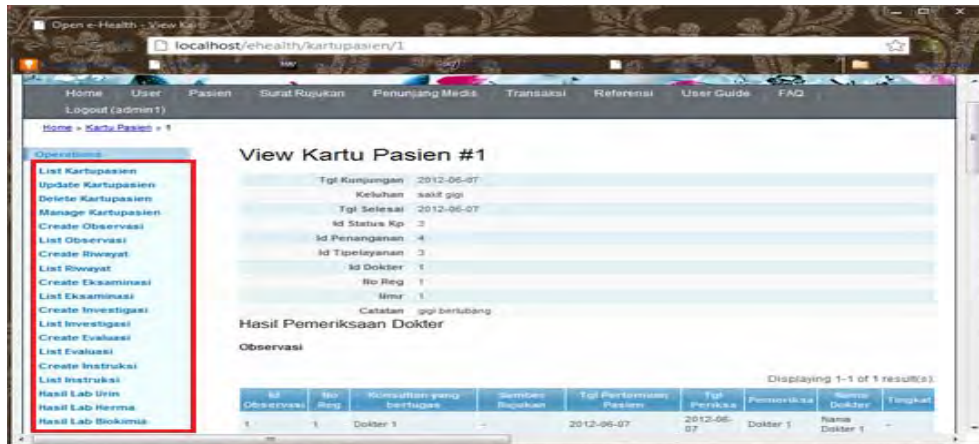
Id dokter diisi sesuai dengan nama dokter yang hendak memeriksa pasien. Nama dokter akan secara otomatis tertera pada list sesuai dengan dokter yang telah dibuat oleh administrator pada saat membuat User.

Id tipe layanan bersifat optional terutama untuk pasien yang baru dibuatkan kartu pasiennya. Id tipe layanan baru akan diisi apabila dokter merujuk pasien ke penunjang medis seperti laboratorium. Pada website ini tersedia 3 tipe pelayanan yaitu Laboratorium Urinalisis, Laboratorium Bio Kimia, dan Laboratorium Urinalisis. Catatan digunakan apabila hendak memberikan keterangan tambahan yang dapat dibaca oleh dokter.

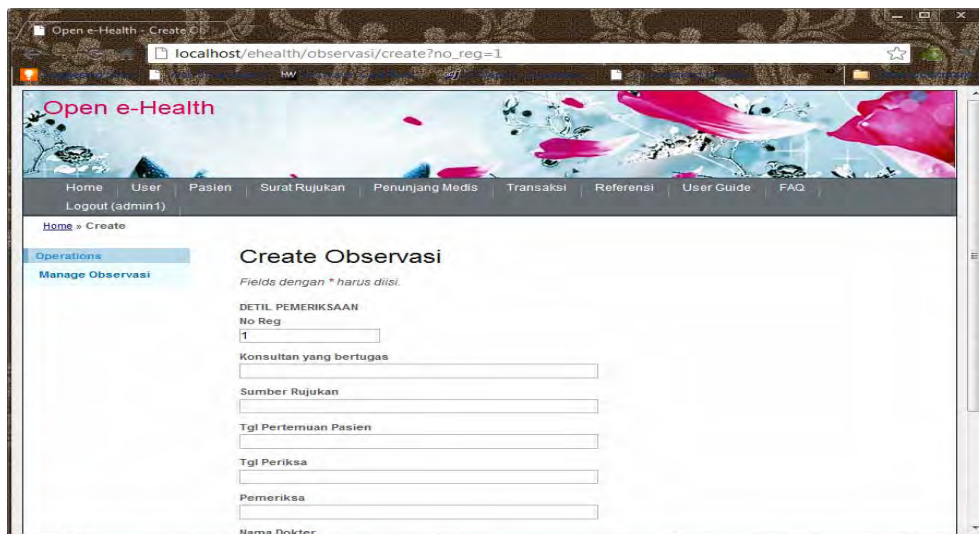
Pada kartu pasien juga dapat dilakukan operasi untuk melihat daftar, melihat keterangan lengkap, menghapus, mengubah, dan mengolah kartu pasien.

- Mengisi Hasil Pemeriksaan Dokter

Dokter dapat mengisi hasil pemeriksaannya melalui kartu pasien. Setelah dibuat kartu pasien, akan muncul tampilan keterangan lengkap kartu pasien. Pada side bar menu terdapat beberapa pilihan untuk mengisi pemeriksaan dokter sesuai kebutuhan. Terdapat pula hasil laboratorium dari kartu pasien yang dirujuk oleh dokter.



Ada beberapa kategori pemeriksaan dokter, yaitu observasi, riwayat, eksaminasi, investigasi, evaluasi, dan instruksi. Hasil pemeriksaan dokter ini dibuat berdasarkan no registrasi dari kartu pasien. Observasi digunakan untuk menuliskan keterangan-keterangan yang berhubungan dengan pemeriksaan.



Riwayat merupakan fasilitas yang dapat dimanfaatkan dokter untuk mengelola riwayat medis pasien, baik penyakit, pengobatan kini, maupun pengobatan yang telah lalu.

Open e-Health - Create Riwayat

localhost/ehealth/riwayat/create?no_reg=1

Open e-Health

Home | User | Pasien | Surat Rujukan | Penunjang Medis | Transaksi | Referensi | User Guide | FAQ

Logout (admin1)

Home » Create

Operations

Manage Riwayat

Create Riwayat

Field dengan * harus diisi.

RIWAYAT PASIEN

No Reg

1

Alasan Penerimaan

Riwayat Keluhan

RIWAYAT MEDIS

Riwayat Medis Umum

Penyakit Infeksi dan parasit

Pengobatan

Eksaminasi merupakan fasilitas yang dapat dimanfaatkan dokter untuk menuliskan hasil pemeriksaan langsung terhadap pasien. Pada eksaminasi ini berisi data-data yang bisa diukur dari pengamatan dokter.

Open e-Health - Create Eksaminasi

localhost/ehealth/eksaminasi/create?no_reg=1

Open e-Health

Home | User | Pasien | Surat Rujukan | Penunjang Medis | Transaksi | Referensi | User Guide | FAQ

Logout (admin1)

Home » Create

Operations

Manage Eksaminasi

Create Eksaminasi

Field dengan * harus diisi.

No Reg

1

OBJEKTIF EKSAMINASI

Penampakan Umum

TANDA VITAL

Tingkat Pernafasan

Denyut Jantung

Tekanan Darah Sistolik

Pengobatan

Investigasi merupakan fasilitas yang dapat dimanfaatkan dokter untuk menuliskan hasil analisisnya terhadap hasil laboratorium dari pasien yang dirujuknya. Setelah menganalisis hasil laboratorium, dokter dapat menuliskan resume terhadap hasil laboratorium di investigasi. Untuk menganalisis hasil

laboratorium dapat dipilih pada sidebar menu pada tampilan data kartu pasien.

The screenshot shows the 'Create Investigasi' form in the Open e-Health system. The browser address bar displays 'localhost/ehealth/investigasi/create?no_reg=1'. The system header includes the 'Open e-Health' logo and a navigation menu with links: Home, User, Pasien, Surat Rujukan, Penunjang Medis, Transaksi, Referensi, User Guide, and FAQ. A 'Logout (admin1)' link is also present. The left sidebar shows the 'Operations' menu with 'Manage Investigasi' selected. The main form area is titled 'Create Investigasi' and includes a note: 'Fields dengan * harus diisi.' The form contains the following fields: 'No Reg' (with value '1'), 'DETIL INVESTIGASI' (with a sub-field 'Oksimetri %'), 'Glukosa Darah', 'Urinisasi', 'PERMINTAAN DAN HASIL HERMATOLOGI' (with a sub-field 'Permintaan FBC'), and a large empty text area for additional details.

Evaluasi dapat digunakan dokter untuk menuliskan hasil evaluasi dari pengamatan terhadap observasi sebelumnya. Yang termasuk ke dalam observasi adalah observasi, riwayat, eksaminasi, dan investigasi. Evaluasi berisi penilaian, pendapat serta tujuan.

The screenshot shows the 'Create Evaluasi' form in the Open e-Health system. The browser address bar displays 'localhost/ehealth/evaluasi/create?no_reg=1'. The system header and navigation menu are identical to the previous screenshot. The left sidebar shows the 'Operations' menu with 'Manage Evaluasi' selected. The main form area is titled 'Create Evaluasi' and includes a note: 'Field dengan * harus diisi.' The form contains the following fields: 'No Reg' (with value '1'), 'HASIL EVALUASI' (with a sub-field 'Rangkuman dan Interpretasi Penemuan'), 'Perbedaan Diagnosis', and 'Faktor Resiko Relevan'. There is also a large empty text area for additional details.

Instruksi merupakan apa yang akan dilakukan dokter terhadap pasien di masa mendatang. Instruksi merupakan sesuatu yang harus ditugaskan kepada agen investigator untuk memberikan aksi kepada pasien.

Setiap hasil pemeriksaan dokter ini dapat dilakukan beberapa operasi seperti membuat, mengubah, melihat, menghapus, dan mengelola setiap hasil pemeriksaan dokter. Pada tampilan lengkap kartu pasien juga terdapat tampilan hasil pemeriksaan dokter dan hasil laboratorium dalam bentuk tabel.

1. Halaman Surat Rujukan

Halaman Surat Rujukan merupakan halaman berisi daftar kartu pasien yang akan dibuatkan surat rujukannya oleh dokter. Setelah memeriksa pasien dan memasukkan hasil pemeriksaan ke archetype, dokter dapat memutuskan apakah pasien perlu dirujuk atau tidak. Apabila pasien tidak perlu dirujuk, maka pasien sudah tidak memerlukan penanganan. Apabila perlu dirujuk, dokter dapat menentukan apakah pasien perlu dirujuk ke laboratorium atau ke dokter lain. Terdapat dua jenis surat rujukan, yaitu surat rujukan dokter dan surat rujukan laboran. Surat rujukan dokter dibuat apabila dokter memutuskan bahwa pasien memerlukan penanganan dokter, sedangkan surat rujukan laboran dibuat apabila pasien memerlukan pelayanan laboratorium.

Untuk membuat surat rujukan, dokter dapat memilih berdasarkan kartu pasien pada halaman surat rujukan. Surat rujukan dibuat berdasarkan kartu pasien.

The screenshot shows a web browser window with the URL `localhost/ehealth/suratrujukandokter/create/1`. The page title is "Open e-Health". The navigation menu includes: Home, User, Pasien, Surat Rujukan, Penunjang Medis, Transaksi, Referensi, User Guide, and FAQ. The breadcrumb trail is "Home > Kartupasien > Create". The main heading is "Create Surat Rujukan Dokter Kartu Pasien 1". Below the heading, it says "Fields dengan * harus diisi.". The form contains the following fields: "No Reg" with value "1", "Nmr" with value "1", "Tgl Kunjungan" with value "2012-06-07", and "Keluhan" with value "sakit gigi". There is a "Tgl Selesai" field at the bottom.

Dokter merubah id status kartu pasien menjadi menunggu apabila pasien dirujuk ke dokter lain. Untuk surat rujukan laboran, dokter dapat merubah id status kartu pasien menjadi diperiksa penunjang.

The screenshot shows a web browser window with the URL `localhost/ehealth/suratrujukanlaboran/create/1`. The page title is "Open e-Health". The navigation menu includes: Home, User, Pasien, Surat Rujukan, Penunjang Medis, Transaksi, Referensi, User Guide, and FAQ. The breadcrumb trail is "Home > Kartupasien > Create". The main heading is "Create Surat Rujukan Laboran Kartu Pasien 1". Below the heading, it says "Fields dengan * harus diisi.". The form contains the following fields: "No Reg" with value "1", "Nmr" with value "1", "Id Laboran" with a dropdown menu showing "Laboran Urinalisis 1", "Tgl Kunjungan" with value "2012-06-07", "Tgl Selesai" with value "2012-06-07", and "Id Tipelayanan" with a dropdown menu showing "Laboratorium Urinalisis".

2. Halaman Penunjang Medis

Halaman penunjang medis merupakan halaman yang dapat dimanfaatkan laboran untuk mengisi hasil pemeriksaan laboratorium pasien yang dirujuk oleh dokter ke laboratorium yang dimaksud. Halaman ini menampilkan daftar kartu pasien di halaman awal karena hasil laboratorium dibuat berdasarkan kartu pasien. Terdapat 3 jenis laboratorium pada sistem ini, yaitu Laboratorium Urinalisis, Laboratorium Dermatologi, dan Laboratorium Biokimia. Terdapat fasilitas untuk membuat dan mengakses daftar dari hasil laboratorium. Hasil laboratorium dapat diisi berdasarkan laboratorium masing-masing.

The screenshot shows the 'Open e-Health' web application interface. The browser address bar displays 'localhost/health/laburin/create?no_reg=1'. The navigation menu includes 'Home', 'User', 'Pasien', 'Surat Rujukan', 'Penunjang Medis', 'Transaksi', 'Referensi', 'User Guide', and 'FAQ'. The breadcrumb trail is 'Home > Lab Urinalisis > Create'. The left sidebar shows 'Operations' and 'Manage Hasil Lab Urin'. The main content area is titled 'Create Hasil Laboratorium Urinalisis' with a note 'Fields dengan * harus diisi.' Below this, there are input fields for 'No Reg' (containing '1'), 'Berat Jenis', 'Reaksi', 'Proteina', and 'Reduksi'.

The screenshot shows the 'Open e-Health' web application interface for dermatology. The browser address bar displays 'localhost/health/labherma/create?no_reg=1'. The navigation menu is the same as the previous screenshot. The breadcrumb trail is 'Home > Lab Dermatologi > Create'. The left sidebar shows 'Operations' and 'Manage Hasil Lab Herma'. The main content area is titled 'Create Hasil Laboratorium Dermatologi' with a note 'Field dengan * harus diisi.' Below this, there are input fields for 'No Reg' (containing '1'), 'Malaria', 'Laju Endapan Darah', 'Kadar Hemoglobin', and 'Eritrosit per MM³'.

The screenshot shows a web browser window with the URL `localhost/health/labbiokimia/create?no_reg=1`. The page header includes the 'Open e-Health' logo and a navigation menu with links: Home, User, Pasien, Surat Rujukan, Penunjang Medis, Transaksi, Referensi, User Guide, and FAQ. Below the menu is a 'Logout (admin1)' link. The main content area is titled 'Create Hasil Laboratorium Biokimia' and includes a note: 'Field dengan * harus diisi.' The form contains the following fields: 'No Reg' (with value '1'), 'GULA DARAH', 'Puasa /2jam PP', 'Sewaktu Jam N, 11, 16', and 'G.T.T'.

3. Halaman Transaksi

Halaman transaksi merupakan halaman yang dapat dimanfaatkan Staff untuk membuat transaksi dari penanganan dan pelayanan yang diterima pasien untuk satu kali berkunjung. Ada 2 tipe transaksi, yaitu transaksi penanganan yang merupakan transaksi yang dibuat apabila pasien mendapatkan penanganan dari dokter, serta transaksi pelayanan yang dibuat apabila pasien mendapatkan pelayanan dari laboratorium. Transaksi dibuat berdasarkan kartu pasien, sehingga perlu dipilih kartu pasien yang akan dibuat transaksinya.

The screenshot shows a web browser window with the URL `localhost/health/itempenanganan/create/1`. The page header is identical to the previous screenshot. The main content area is titled 'Create Item Penanganan' and includes a note: 'Field dengan * harus diisi.' The form contains the following fields: 'No Reg' (with value '0'), 'Id Tipe Penanganan' (a dropdown menu), and 'Harga Penanganan'. A 'Create' button is located at the bottom of the form.

Id Tipe Penanganan dapat dipilih berdasarkan jenis penanganan yang didapat pasien seperti penanganan dokter mata, dokter gigi, atau dokter umum.

Open e-Health - Create Item Layanan

localhost/ehealth/itemlayanan/create/1

Open e-Health

Home | User | Pasien | Surat Rujukan | Penunjang Medis | Transaksi | Referensi | User Guide | FAQ

Logout (admin1)

Home » Item Layanan » Create

Operations

Manage Item Layanan

Create Item Layanan

Field dengan * harus diisi.

No Reg
0

Id Tipelayanan
Tidak Ada Layanan

Harga Layanan

Create

Id Tipe Pelayanan dapat dipilih berdasarkan jenis pelayanan laboratorium yang didapat pasien seperti Laboratorium Dermatologi, Urinalisis, atau Biokimia. Dapat dilakukan operasi untuk menambahkan dan mengelola data transaksi yang ada baik transaksi penanganan maupun pelayanan. Untuk melihat daftar akumulasi dari transaksi-transaksi yang dilakukan untuk satu kali kunjungan, dapat dilihat pada tampilan lengkap setelah memilih kartu pasien pada daftar kartu pasien di halaman transaksi.

Open e-Health - View Transaksi

localhost/ehealth/transaksidashboard/1

View Pasien Berkunjung #1

No Reg: 1
Nmr: 1
Tgl Kunjungan: 2012-06-07
Keluhan: sakit gigi
Tgl Selesai: 2012-06-07
Id Status Kp: 3
Id Penanganan: 4
Id Dokter: 1
Catatan: gigi berlubang

Transaksi Penanganan

Displaying 1-1 of 1 result(s).

Id Itempenanganan	No Reg	Id Tipe Penanganan	Harga Penanganan
1	1	4	100000

Transaksi Layanan

Displaying 1-1 of 1 result(s).

Id Itemlayanan	No Reg	Id Tipelayanan	Harga Layanan
1	1	3	150000

4. Halaman Referensi

Halaman referensi merupakan yang dapat dimanfaatkan pengguna untuk melihat, menambah, mengubah, mengatur, dan melihat daftar referensi yang ada dalam setiap jenis referensi. Referensi berfungsi untuk memberikan pilihan pada beberapa pilihan agar pilihan bersifat fleksible yaitu dapat dilakukan operasi penambahan, perubahan, dan penghapusan pada item referensi. Terdapat 10 referensi yang tersedia yaitu agama, asuransi, jenis kelamin, status, peran, spesialis, status kartu pasien, tipe layanan, tipe penanganan, dan ICD 10.

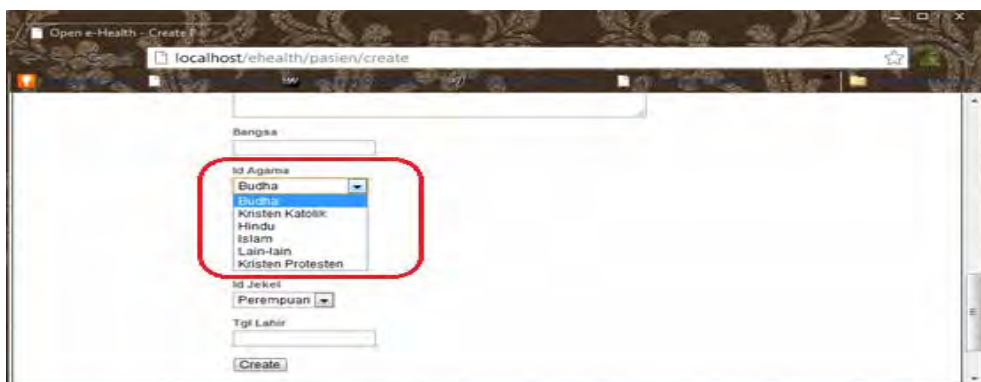
Halaman referensi ini hanya dapat dibuat, diubah, dan dikelola oleh admin, sedangkan pengguna lain hanya dapat melihat daftar dan keterangan dari referensi yang dimaksud. Halaman referensi juga diharapkan dapat membantu pengguna untuk lebih mengenal keterangan id dari referensi yang ada.

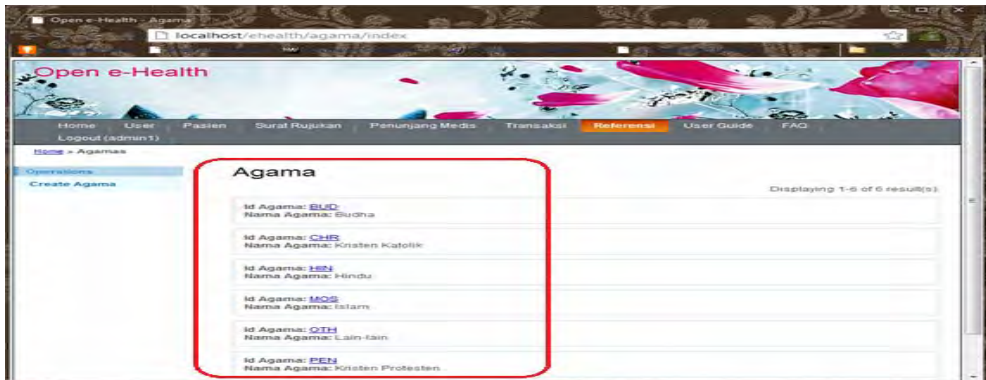
Untuk agama, jenis kelamin, status, dan ICD 10 hanya dapat dilihat daftarnya oleh pengguna karena mengandung kode-kode yang telah ditentukan sebelumnya. Berikut daftar dari referensi-referensi yang tersedia.

Berikut field yang menggunakan referensi sebagai pilihan.

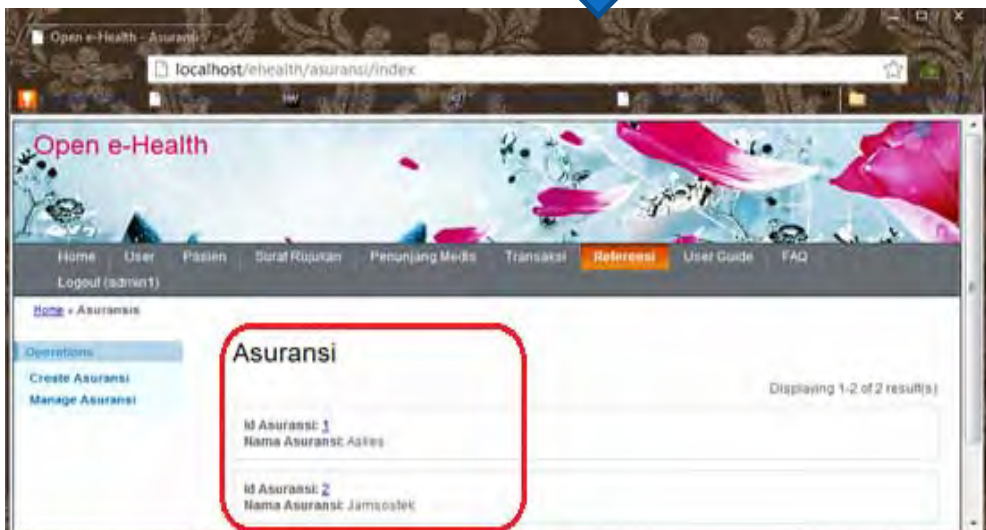
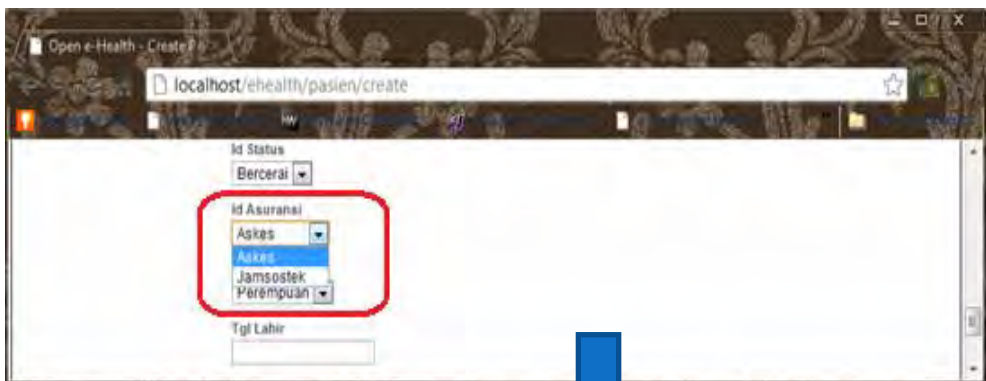
- Pada saat hendak membuat demografi pasien baru

Field id agama yang memberikan beberapa pilihan yang dapat dikelola di referensi.





Field Id Asuransi pada saat membuat demografi pasien baru



Field Id Status pada saat membuat demografi pasien baru

The screenshot shows a web browser window with the URL `localhost/health/pasien/create`. The form contains several fields: 'Id Status' (a dropdown menu with options 'Bercera', 'Bercera', 'Menikah', and 'Lajang'), 'Id Jekel' (a dropdown menu with 'Perempuan'), and 'Tgl Lahir' (a text input field). The 'Id Status' dropdown is highlighted with a red rectangle.



The screenshot shows a web browser window with the URL `localhost/health/status/index`. The page displays a table titled 'Status' with the following data:

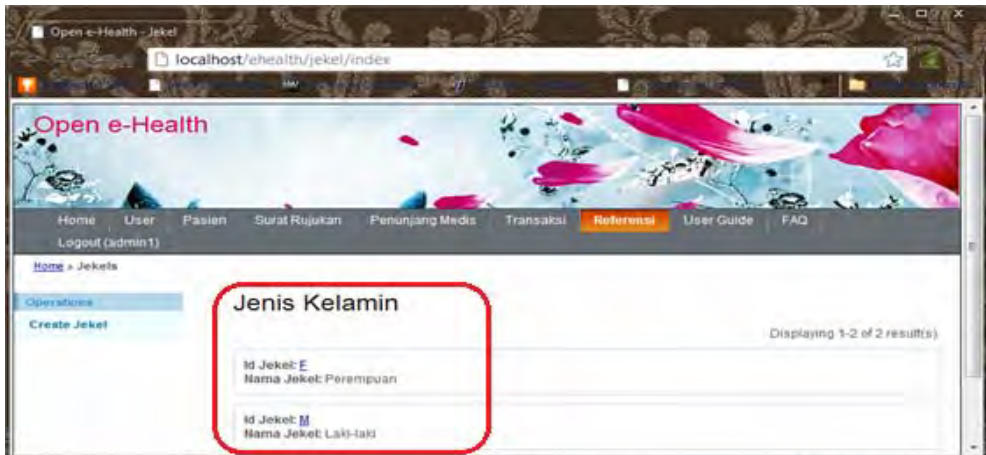
Id Status	Nama Status
B	Bercera
M	Menikah
L	Lajang

The table is highlighted with a red rectangle. The page also shows a navigation bar with links like 'Home', 'User', 'Pasien', 'Surat Rujukan', 'Penunjang Medis', 'Transaksi', 'Referensi', 'User Guide', and 'FAQ'. The 'Referensi' link is highlighted in orange.

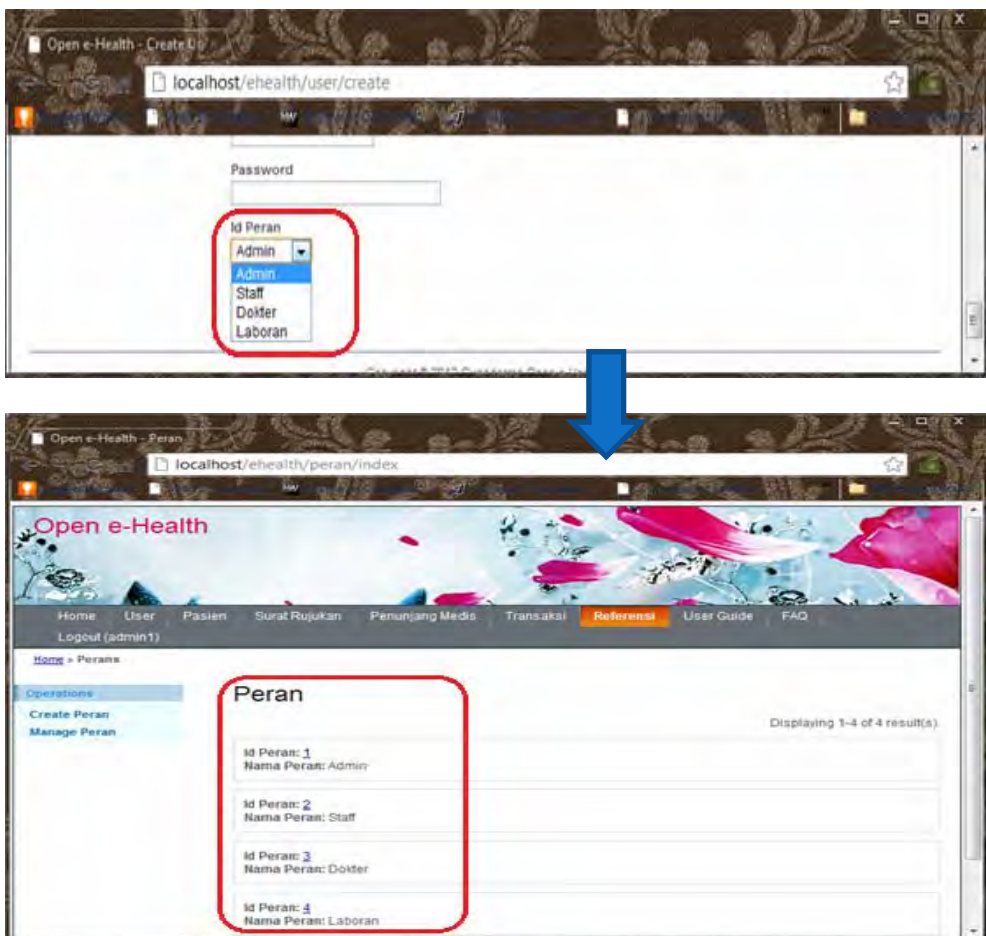
Id Jenis Kelamin pada saat membuat demografi pasien baru

The screenshot shows a web browser window with the URL `localhost/health/pasien/create`. The form contains several fields: 'Id Status' (a dropdown menu with 'Bercera'), 'Id Asuransi' (a dropdown menu with 'Askes'), and 'Id Jekel' (a dropdown menu with options 'Perempuan', 'Perempuan', and 'Laki-laki'). The 'Id Jekel' dropdown is highlighted with a red rectangle.

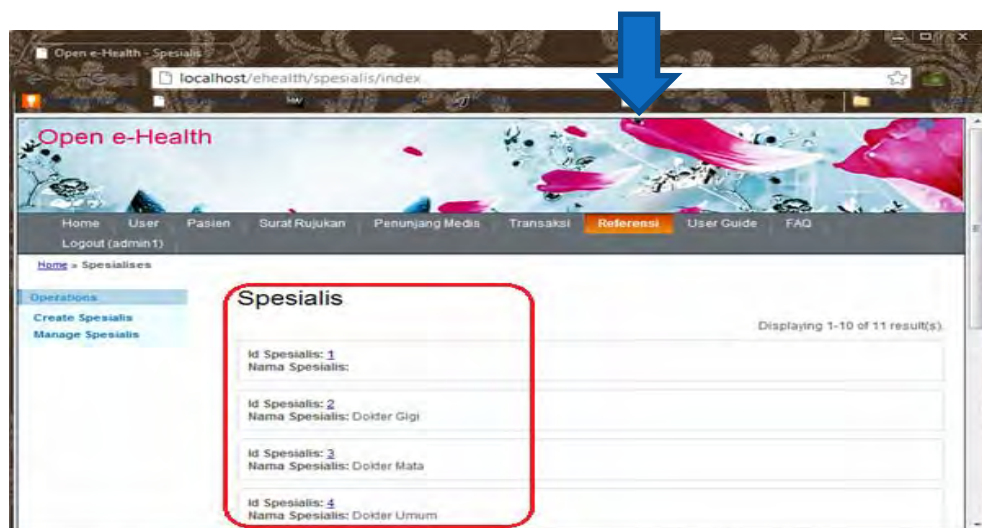
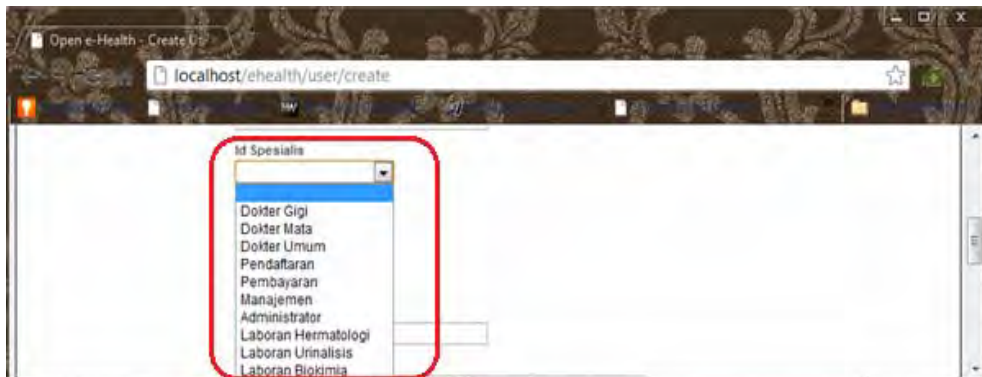




Field Id peran pada saat membuat pengguna baru

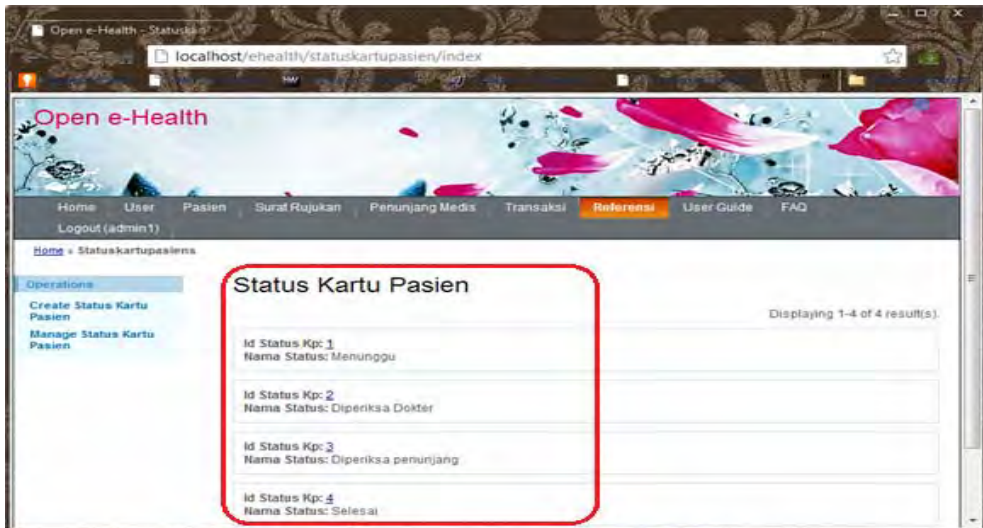


Id spesialis pada saat membuat pengguna baru

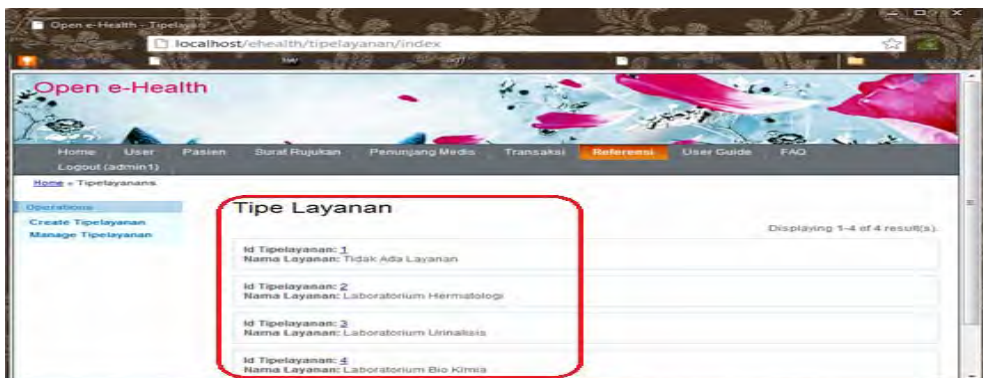
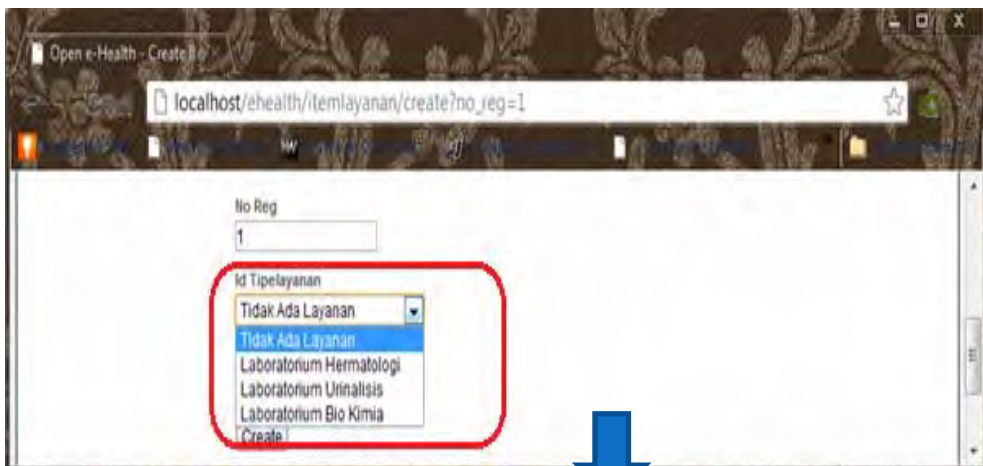


Id Status Kartu Pasien pada saat membuat kartu pasien baru

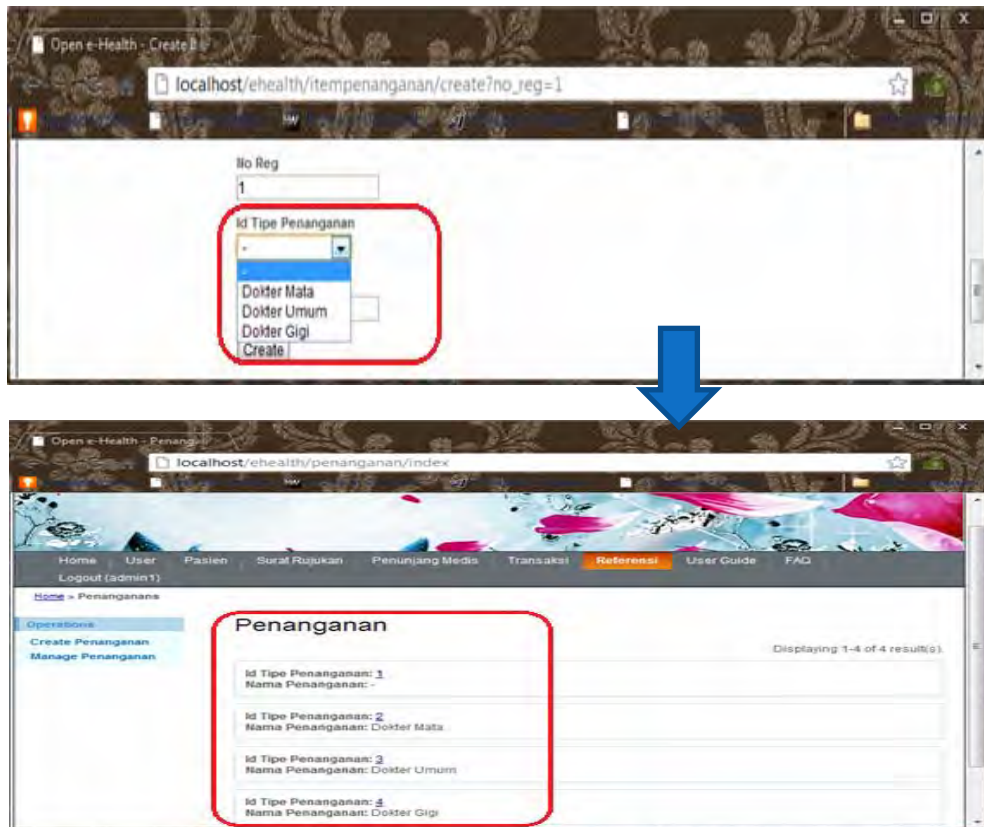




Id Tipe Layanan pada saat membuat transaksi pelayanan



Id Tipe Penanganan pada saat membuat transaksi penanganan



Riwayat medis umum pada saat membuat riwayat pada pemeriksaan dokter



Riwayat medis dihubungkan dengan referensi dari kode ICD 10 yang menklasifikasikan jenis penyakit.



4.3. Pedoman Untuk Administrator

Administrator merupakan pengguna yang bertugas sebagai pengatur dan pemelihara website. Administrator dapat membuat pengguna baru dan dapat mengakses semua fasilitas dari website. Untuk data di halaman user, admin yang bertugas membuat, merubah, menghapus, dan mengelola data pengguna, sedangkan pengguna lain hanya dapat melihat daftar dan keterangan lengkap dari halaman User. Pengguna lain dapat merubah datanya melalui dashboard masing-masing pengguna. Untuk penggunaan pertama kali, administrator dapat dengan menggunakan username dan password admin1, setelah itu administrator dapat membuat pengguna baru. Adminstrator juga bertugas mengelola referensi dari pilihan untuk beberapa field yang tersedia. Admin dapat membuat, merubah, menghapus dan mengelola data referensi, sedangkan pengguna lain hanya dapat melihat daftar dan keterangan lengkap referensi.

4.3.1. Cara Membuat Pengguna Baru

Fasilitas untuk membuat pengguna baru hanya dapat dilakukan oleh administrator. Untuk membuat pengguna baru, dapat dilakukan dengan:

1. Login sebagai administrator
2. Buka ke tab User
3. Pada Side Bar Operations yang terletak di sebelah kiri, pilih Create User.
4. Isi data-data pengguna baru.
 - Pilih id spesialis sesuai profesi dari pengguna baru
Untuk id spesialis dapat dipilih sesuai spesialis dari pengguna, seperti pilihan dokter mata, dokter gigi, atau dokter umum untuk dokter. Daftar id spesialis ini dapat ditambahkan, diubah, ataupun dihapus pada halaman referensi yang akan dibahas selanjutnya.
 - Isi username dan password default yang akan diberikan ke pengguna baru
 - Pilih Id peran pengguna baru sesuai dengan peran dari pengguna.

Untuk Id peran dapat dipilih sesuai peran dari pengguna yang akan diberi hak aksesnya apakah sebagai admin, staff, dokter, atau laboran. Khusus untuk pengguna laboran, nama diisi dengan nama laboratorium tempat laboran bekerja. Apabila laboran yang akan memakai sistem lebih dari satu, dapat diberikan angka dibelakang nama laboratorium, seperti Laboran Urinalisis 1, Laboran Urinalisis 2, dst.

5. Tekan Create.

Untuk melihat daftar User dapat dipilih List User di sidebar. Untuk manage User (melihat semua user pada tabel dan dapat diolah) dengan pilihan manage user. Untuk selengkapnya dapat dilihat pada bab sebelumnya. Admin dapat melakukan operasi create, update, delete dan mengakses list & manage, sedangkan user lain selain pasien hanya dapat melihat user keterangan pengguna dan melihat daftar pengguna. Untuk pasien tidak dapat mengakses menu User. Untuk mengetahui daftar id dapat dilihat pada Referensi.

4.3.2. Mengelola Referensi

Referensi dapat dikelola oleh administrator. Terdapat beberapa referensi. Untuk lebih memahami referensi yang ada dapat dilihat pada bab sebelumnya. Untuk membuat referensi baru, pada halaman referensi dapat dipilih, item mana yang akan dikelola referensinya. Terdapat beberapa referensi yang menggunakan kode-kode yang disesuaikan dengan kode HL7 dan ICD10 seperti referensi untuk agama, jenis kelamin, agama, status, dan ICD10 untuk kode penyakit pada riwayat medis umum. Untuk referensi yang mempunyai kode sebelumnya sudah disertakan pada situs ini. Dianjurkan untuk tidak merubah kode yang sudah ada. Untuk menambahkan referensi lain seperti asuransi, peran, spesialis, status kartu pasien, tipe layanan, dan tipe penanganan, dapat dipilih item yang dimaksud dan dipilih Create pada sidebar menu. Administrator juga diberi hak untuk mengelola semua fasilitas yang ada di situs ini.

4.4. Pedoman Untuk Petugas Administrasi

Petugas administrasi atau staff merupakan pengguna yang melakukan tugas di bagian pendaftaran dan pembayaran. Staff dapat membuat demografi pasien sekaligus membuat akun pasien yang sudah mendaftar. Selain membuat pasien baru, staff juga bertugas membuat kartu pasien dari pasien yang sudah mendaftar. Apabila pasien belum pernah mendaftar maka staff harus membuat demografi pasien tersebut terlebih dahulu. Apabila pasien sudah pernah mendaftar, staff hanya menambahkan kartu pasien di data pasien yang sudah ada sebelumnya.

4.4.1. Membuat Demografi Pasien

Untuk membuat pasien baru dapat dilakukan dengan cara sebagai berikut :

1. Login sebagai staff
2. Buka ke tab Pasien
3. Pada Side Bar Operations yang terletak di sebelah kiri, pilih Create Pasien.
4. Isi data-data pasien baru yang mendaftar.
 - Isi username dan password sementara yang harus diinformasikan ke pasien yang akan mendaftar. Pasien dapat merubah username dan passwordnya setelah login ke website ini nantinya.

- Isi data demografi pasien sesuai dengan keterangan pasien yang mendaftar.
- Format penanggalan yang digunakan adalah YYYY-MM-DD, contoh : 2012-12-30.

5. Tekan Create.

Untuk melihat daftar Pasien pilih List Pasien di sidebar. Untuk manage Pasien (melihat semua pasien pada tabel dan dapat diolah) dengan pilihan manage pasien. Staff dapat melakukan operasi create, update, delete dan mengakses list & manage, sedangkan user lain hanya dapat melakukan view dan mengakses list. Untuk pasien tidak dapat mengakses menu Pasien, namun dapat mengakses demografi pasien tersebut melalui dashboard pasien. Untuk pengelolaan data pasien lebih lengkapnya dapat dilihat pada bab sebelumnya.

4.4.2. Membuat Kartu Pasien

Selain membuat pasien baru, staff juga bertugas membuat kartu pasien dari pasien yang sudah mendaftar. Apabila pasien belum pernah mendaftar maka staff harus membuat demografi pasien tersebut terlebih dahulu. Apabila pasien sudah pernah mendaftar, staff hanya menambahkan kartu pasien di data pasien yang sudah ada sebelumnya.

Untuk membuat kartu pasien, staff terlebih dahulu harus berada di data pasien yang dimaksud. Staff dapat membuat kartu pasien dengan cara :

1. Buka tab Pasien
2. Pilih pasien yang akan dibuat kartu pasiennya dengan memilih nmr pada daftar pasien. Akan muncul tampilan demografi pasien.
3. Pada Side Bar Operations yang terletak di sebelah kiri, pilih Create Kartu Pasien.
4. Isi data-data kartu pasien sesuai field-field yang telah disediakan.
 - Id Status Kp diisi sesuai dengan status kartu pasien.Apabila pasien masih baru mendaftar dan belum mendapatkan penanganan maka status kartu pasien adalah menunggu. Apabila pasien telah diperiksa oleh dokter maka status kartu pasien adalah

diperiksa dokter. Apabila pasien sedang mendapat pelayanan penunjang medis seperti laboratorium, status kartu pasien adalah diperiksa penunjang. Status kartu pasien berubah menjadi selesai apabila pasien sudah selesai berkunjung ke klinik pada hari berkunjung.

- Id penanganan, diisi sesuai dengan penanganan apa yang harus diberikan kepada pasien sesuai dengan keluhan pasien.

Terdapat 3 penanganan, yaitu dokter mata, dokter umum, dan dokter gigi. Untuk menambahkan penanganan yang terdapat di klinik, dapat dilakukan di tab referensi, kemudian pilih penanganan.

- Id dokter diisi sesuai dengan nama dokter yang hendak memeriksa pasien.

Nama dokter akan secara otomatis tertera pada list sesuai dengan dokter yang telah dibuat oleh administrator pada saat membuat User.

- Id tipe layanan bersifat optional terutama untuk pasien yang baru dibuatkan kartu pasiennya.

Id tipe layanan baru akan diisi apabila dokter merujuk pasien ke penunjang medis seperti laboratorium. Pada website ini tersedia 3 tipe pelayanan yaitu Laboratorium Urinalisis, Laboratorium Bio Kimia, dan Laboratorium Urinalisis.

- Catatan digunakan apabila hendak memberikan keterangan tambahan yang dapat dibaca oleh dokter.

5. Pilih Create.

Untuk melihat daftar Kartu Pasien pilih List Kartu Pasien di sidebar. Untuk manage Kartu Pasien (melihat semua pasien pada tabel dan dapat diolah) dengan pilihan manage Kartu pasien. Staff dapat melakukan membuat, mengubah, menghapus, dan mengelola data kartu pasien, sedangkan user lain hanya dapat melameliht daftar dan keterangan lengkap kartu pasien saja. Untuk keterangan lengkap tentang pengelolaan data kartu pasien, dapat dilihat pada bab sebelumnya.

Kartu pasien merupakan representasi rekam medis. Semua kegiatan medis yang pernah diberikan ke seorang pasien dicatat di kartu pasien. Ketika muncul tampilan kartu pasien, akan terdapat menu untuk membuat pengamatan oleh dokter. Fasilitas ini hanya dapat dimanfaatkan oleh dokter, staff hanya dapat mengakses list dari pengamatan tersebut saja.

4.4.3. Penggunaan Dashboard Staff

Untuk mempermudah kegiatan staff, staff dapat melakukan kegiatannya dengan mengakses menu dashboard staff yang hanya akan muncul ketika staff yang melakukan login. Di dalam menu dashboard staff ini, staff dapat manage kartu pasien dari pasien yang berkunjung. Staff harus selalu memantau id status kartu pasien.

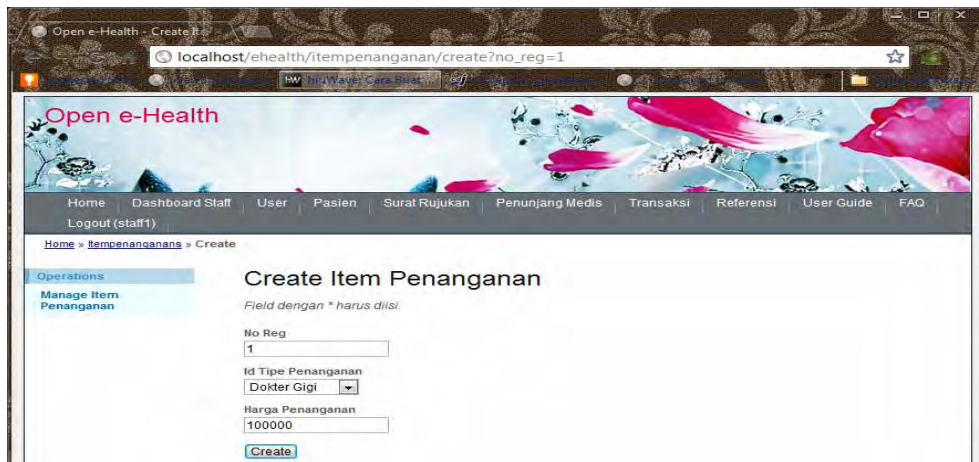
No Reg	Rm	Tgl Kunjungan	Keluhan	Tgl Selesai	Id Status	Id Penanganan	Id Tindakan	Id Dokter
1	1	2012-06-07	sakit gigi	2012-06-07	3	4	3	1
2	1	2012-06-07	sakit mata	2012-06-07	1	2	1	2
3	2	2012-06-07	Sakit Gigi	2012-06-07	1	4	1	1

Untuk lebih jelasnya berikut adalah id status kartu pasien dan status kartu pasien :

- 1 = Menunggu
- 2 = Diperiksa Dokter
- 3 = Diperiksa Penunjang
- 4 = Selesai

4.4.4. Membuat Transaksi Penanganan

Apabila status kartu pasien sudah berubah dari 1 ke 2 (hal ini terjadi setelah pasien mendapatkan penanganan dokter dan dokter merubah status kartu pasien) maka staff harus membuat transaksi penanganan sesuai dengan penanganan oleh dokter yaitu dengan cara :



1. Pilih icon view pada tabel sesuai dengan pasien yang dimaksud.
2. Pilih Create Transaksi Penanganan.
3. Isi Id tipe penanganan sesuai dengan id penanganan yang dapat dilihat di tabel manage kartu pasien.
4. Isi harga yang dikenakan pada user.
5. Pilih Create

Untuk mengetahui daftar id tipe penanganan dapat dilihat pada tipe penanganan pada halaman referensi. Berikut daftar id tipe penanganan dan nama penanganan:

- 1 = Tidak Ada Layanan
- 2 = Laboratorium Dermatologi
- 3 = Laboratorium Urinalisis
- 4 = Laboratorium Bio Kimia

4.4.5. Membuat Transaksi Pelayanan

Apabila status kartu pasien sudah berubah dari 2 ke 3 (hal ini terjadi setelah pasien mendapatkan penanganan dokter dan dokter merujuk pasien ke laboran sehingga dokter merubah status kartu pasien) maka staff harus

membuat transaksi layanan sesuai dengan layanan yang dirujuk oleh dokter yaitu dengan cara :

1. Pilih icon view pada tabel sesuai dengan pasien yang dimaksud.
2. Pilih Create Transaksi Layanan.
3. Isi Id tipe layanan sesuai dengan id tipe layanan yang dapat dilihat di tabel manage kartu pasien.
4. Isi harga yang dikenakan pada pasien.
5. Pilih Create

Untuk mengetahui daftar id tipe pelayanan dapat dilihat pada tipe pelayanan pada halaman referensi. Berikut daftar id tipe pelayanan dan nama penanganannya:

- 1 = -
- 2 = Dokter Mata
- 3 = Dokter Umum
- 4 = Dokter Gigi

Apabila status kartu pasien sudah berubah menjadi 4 (hal ini terjadi setelah dokter menyatakan pasien sudah selesai berobat) maka staff tidak harus berkonsentrasi terhadap pasien tersebut.

4.4.6. Membuat Info Klinik

Staff juga dapat membuat informasi klinik baik klinik tempat website digunakan, ataupun klinik lain yang mempunyai relasi dengan klinik. Untuk membuat info klinik dapat dilakukan sebagai berikut:

1. Login sebagai staff
2. Pada dashboard staff, pada side bar menu pilih Create Info Klinik
3. Info klinik dapat diisi sesuai dengan informasi klinik yang akan disimpan.

The screenshot shows a web browser window with the URL `localhost/ehealth/infoklinik/create`. The page title is "Create Info Klinik". On the left, there is a sidebar menu with "Operations", "List Infoklinik", and "Manage Infoklinik". The main content area contains a form with the following fields and values:

- Nama Klinik: rumah sakit a
- Jenis Klinik: rumah sakit
- Nama Kepala: nama kepala rumah sakit a
- Alamat: alamat tempat rumah sakit a berada
- Telepon: 021478238
- Fax: 021874783
- Kota: Depok
- Kode Pos: 269189

At the bottom of the form is a "Create" button.

4.5. Pedoman Untuk Dokter

Dokter merupakan pengguna yang memeriksa pasien. Dokter mengisi setiap hasil pemeriksaannya melalui kartu pasien. Dokter memeriksa pasien yang telah ditentukan sebelumnya oleh staff. Setelah itu dokter harus melaporkan hasil pemeriksaannya yaitu dengan mengisi archetype di kartu pasien. Dokter dapat membuat surat rujukan untuk merujuk pasien ke dokter yang lain atau ke suatu laboratorium. Untuk mempermudah kinerja dokter, dokter dapat melakukan aktivitasnya dengan menggunakan dashboard dokter yang hanya akan ada apabila dokter melakukan login. Di dalam menu dashboard dokter ini dokter hanya akan melihat pasien yang perlu diperiksa oleh dokter yang login. Dokter dapat mengelola pasien yang harus diperiksa.

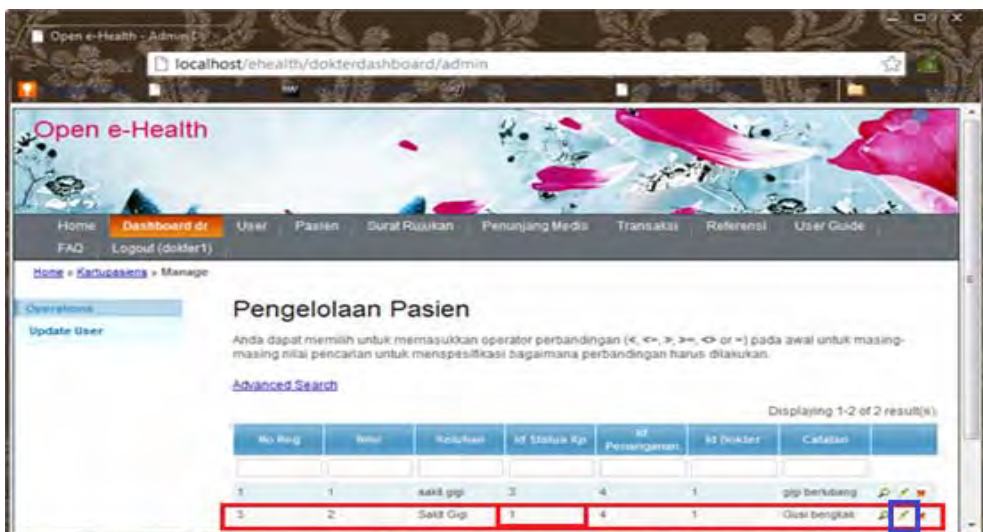
Dokter harus memantau id status kartu pasien. Untuk lebih jelasnya berikut adalah id status kartu pasien dan status kartu pasien :

- 1 = Menunggu
- 2 = Diperiksa Dokter
- 3 = Diperiksa Penunjang
- 4 = Selesai

4.5.1. Dokter Merubah Id Status Kartu Pasien diperiksa

Apabila status kartu pasien 1, dokter dapat memulai memeriksa pasien tersebut yaitu dengan merubah id status kartu pasien menjadi 2. Untuk merubah id status kartu pasien dapat dilakukan dengan cara sebagai berikut :

1. Pilih ikon edit pada tabel di dashboard dokter sesuai dengan pasien yang dimaksud.



2. Pada field id status kartu pasien, ubah dari menunggu menjadi diperiksa dokter.

3. Pilih Update.

Setelah melakukan update, dokter dapat melihat tampilan kartu pasien. Di tampilan ini, dokter dapat melihat demografi dan rekam medis, membuat surat rujukan ke dokter dan surat rujukan ke Laboratorium.

4.5.2. Dokter Menuliskan Hasil Pemeriksaan

Setelah mengubah id status kartu pasien, dokter dapat mulai memeriksa pasien. Setelah memeriksa pasien, dokter dapat menuliskan hasil pemeriksaannya di archetype. Apabila dokter ingin menuliskan hasil pemeriksaannya, dokter dapat memilih Rekam Medis pada sidebar. Setelah itu dokter dapat mulai mengisi archetype. Archetype terdiri dari 3 yaitu :

1. Observasi

Observasi mempunyai 4 bagian yaitu observasi, riwayat, eksaminasi dan investigasi. Secara umum, observasi merupakan hasil pengamatan yang menghasilkan data-data yang dapat diukur dengan angka. Observasi merupakan hasil pemeriksaan langsung ke pasien. Observasi terdiri dari:

- Observasi

Observasi digunakan untuk menuliskan keterangan-keterangan yang berhubungan dengan pemeriksaan, seperti nama konsultan,

tanggal kunjungan pasien, nama dokter, dll. Diisi apabila dokter melakukan pemeriksaan terhadap pasien.

- Riwayat

Riwayat merupakan fasilitas yang dapat dimanfaatkan dokter untuk mengelola riwayat medis pasien, baik penyakit, pengobatan kini, maupun pengobatan yang telah lalu. Diisi sesuai riwayat pasien dan digunakan sebagai referensi untuk mengambil keputusan.

- Eksaminasi

Eksaminasi merupakan fasilitas yang dapat dimanfaatkan dokter untuk menuliskan hasil pemeriksaan langsung terhadap pasien. Pada eksaminasi ini berisi data-data yang bisa diukur dari pengamatan dokter. Diisi ketika dokter melakukan pemeriksaan langsung kepada pasien.

- Investigasi.

Investigasi merupakan fasilitas yang dapat dimanfaatkan dokter untuk menuliskan hasil analisisnya terhadap hasil laboratorium dari pasien yang dirujuknya. Setelah menganalisis hasil laboratorium, dokter dapat menuliskan resume terhadap hasil laboratorium di investigasi. Untuk menganalisis hasil laboratorium dapat dipilih pada sidebar menu pada tampilan data kartu pasien.

2. Evaluasi

Evaluasi merupakan hasil pengamatan yang tidak dapat diukur seperti diagnosa. Evaluasi dilakukan oleh investigator yang menganalisis hasil observasi berdasarkan bukti dan pengetahuan pribadi. Evaluasi berisi penilaian, pendapat serta tujuan. Dibuat ketika dokter/petugas medis mengevaluasi observasi sebelumnya.

3. Instruksi

Instruksi merupakan apa yang akan dilakukan dokter terhadap pasien di masa mendatang. Instruksi merupakan sesuatu yang harus ditugaskan kepada agen investigator untuk memberikan aksi kepada

pasien. Dibuat ketika pasien memerlukan keputusan untuk dilakukan tindakan terhadap pasien.

Untuk keterangan lengkap tentang archetype dapat di lihat di Subbab 4.2. pedoman untuk Registered User.

Di dalam tampilan rekam medis, Dokter juga dapat melihat hasil laboratorium dari pasien yang dirujuk oleh dokter. Dari tampilan rekam medis ini, dokter dapat melihat archetype dan hasil laboratorium pasien dalam 1 kali berkunjung (per kartu pasien).

4.5.3. Dokter Membuat Surat Rujukan Dokter

Setelah memeriksa pasien dan memasukkan hasil pemeriksaan ke archetype, dokter dapat memutuskan apakah pasien perlu dirujuk atau tidak. Surat rujukan dibuat berdasarkan no registrasi kartu pasien. Apabila dokter hendak merujuk pasien ke dokter yang lain, dokter harus membuat surat rujukan dokter yaitu dengan :

The screenshot displays a web browser window with the title 'Open e-Health - Create S...'. The address bar shows 'localhost/ehealth/suratrujukandokter/create/1'. The page header features the 'Open e-Health' logo and a navigation menu with links: Home, User, Pasien, Surat Rujukan, Penunjang Medis, Transaksi, Referensi, User Guide, and FAQ. Below the menu is a 'Logout (admin1)' link. The main content area is titled 'Create Surat Rujukan Dokter Kartu Pasien 1' and includes a note 'Fields dengan * harus diisi.'. The form contains the following fields: 'No Reg' with value '1', 'Nmr' with value '1', 'Tgl Kunjungan' with value '2012-06-07', and 'Keluhan' with value 'sakit gigi'. A 'Tgl Selesai' field is visible at the bottom of the form.

1. Pilih icon view pada tabel di dashboard dokter sesuai dengan pasien yang akan dibuat surat rujukan.

2. Pilih Create Surat Rujukan Dokter pada sidebar menu.
3. Ubah Id Status Kartu pasien menjadi menunggu. Kemudian ubah id penanganan dan id dokter sesuai dengan dokter yang akan dirujuk.
4. Pilih Save.

4.5.4. Dokter Membuat Surat Rujukan Laboran

Apabila dokter memutuskan bahwa pasien memerlukan pelayanan dari laboratorium, maka dokter dapat membuat surat rujukan laboran yaitu dengan :

The screenshot displays a web browser window with the URL `localhost/health/suratrujukanlaboran/create/1`. The page title is "Open e-Health". The navigation menu includes Home, User, Pasien, Surat Rujukan, Penunjang Medis, Transaksi, Referensi, User Guide, and FAQ. The main content area is titled "Create Surat Rujukan Laboran Kartu Pasien 1" and contains a form with the following fields:

- No Reg: 1
- Rmr: 1
- Id Laboran: Laboran Urinalisis 1 (dropdown menu)
- Tgl Kunjungan: 2012-06-07
- Tgl Selesai: 2012-06-07
- Id Tindakan: Laboratorium Urinalisis (dropdown menu)

1. Pilih ikon view pada tabel di dashboard dokter sesuai dengan pasien yang akan dibuat surat rujukan.
2. Pilih Create Surat Rujukan Laboran pada sidebar menu.
3. Ubah Id Status Kartu pasien menjadi diperiksa Laboran. Kemudian ubah id laboran dan id tipe layanan sesuai dengan pelayanan laboratorium yang akan diberikan kepada pasien.
4. Pilih Save.

4.5.5. Dokter Menganalisis Hasil Lab. dan Kembali Menulis Archetype

Setelah dirujuk ke laboratorium, dokter dapat melihat hasil laboratorium dengan memilih Rekam Medis pada sidebar menu tampilan kartu pasien. Setelah melihat hasil laboratorium, dokter harus menganalisis dan menuliskan resume hasil laboratorium dengan membuat investigasi di archetype. Pada

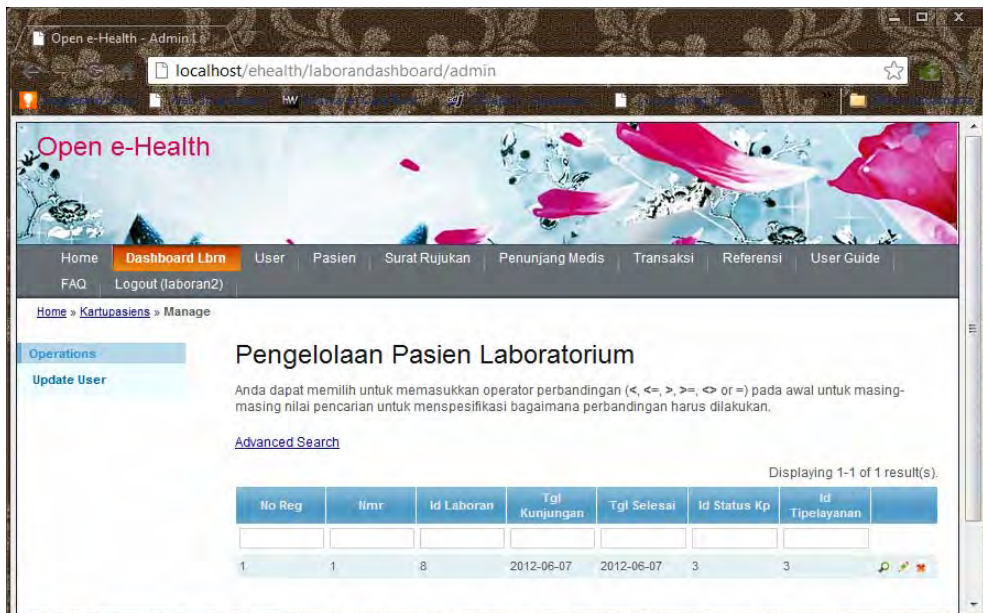
saat hendak membuat investigasi, disediakan field-field dari laboratorium yang tersedia. Dokter dapat mengisi sesuai laboratorium yang dirujuk. Untuk lebih jelasnya dapat dilakukan dengan :

1. Pilih ikon view pada tabel di dashboard dokter sesuai dengan pasien yang akan dibuat investigasi dari hasil laboratorium.
2. Pilih Rekam Medis pada sidebar menu.
3. Pilih Create Investigasi. Isikan sesuai hasil laboratorium yang dirujuk.
4. Pilih Create.

Apabila tidak ada lagi penanganan yang perlu diberikan ke pasien, dokter dapat merubah id status pasien menjadi selesai (4). Untuk mengetahui daftar id dapat dilihat pada halaman Referensi.

4.6. Pedoman Untuk Penunjang Medis

Penunjang medis atau laboran merupakan pengguna yang bertugas di laboratorium. Laboran hanya akan bertugas ketika dokter merujuk pasien ke Laboratorium tempat laboran bekerja. Laboran bertugas memasukkan hasil laboratorium ke dalam sistem yang nantinya akan dianalisis oleh dokter. Untuk mempermudah kinerja laboran, laboran dapat melakukan aktivitasnya dengan menggunakan dashboard laboran yang hanya akan ada apabila laboran login.



Di dalam menu dashboard laboran ini laboran hanya akan melihat pasien yang perlu diperiksa oleh laboran yang login. Laboran dapat manage pasien yang harus diperiksa di laboratorium. Laboran dapat menuliskan hasil laboratorium dari pasien yang diperiksa berdasarkan id tipe layanan. Berikut daftar id tipe layanan beserta nama pelayanan laboratorium yang ada :

- 1 = Tidak Ada layanan
- 2 = Laboratorium Dermatologi
- 3 = Laboratorium Urinalisis
- 4 = Laboratorium Bio Kimia

4.6.1. Laboran Membuat Hasil Laboratorium

Setelah dokter menetapkan pasien yang harus diperiksa laboran, laboran dapat mulai memeriksa pasien dan harus memasukkan hasil laboratorium pasien sesuai dengan laboratorium tempat laboran bekerja. Untuk membuat hasil laboratorium dapat dilakukan dengan cara sebagai berikut :

1. Login sebagai laboran masing-masing laboratorium.
2. Pilih ikon view pada tabel di dashboard laboran sesuai dengan pasien yang dimaksud.

3. Pilih Create Hasil Lab pada sidebar. Hasil lab dapat disesuaikan dengan id pelayanan.
4. Isi field-field berdasarkan hasil laboratorium.
5. Pilih create.

Tugas laboran hanya memasukkan hasil laboratorium saja. Laboran tidak diperkenankan mengubah id status kartu pasien ataupun data kartu pasien. Laboran juga tidak diberikan hak akses untuk menghapus kartu pasien.

Hasil laboratorium dibuat berdasarkan no registrasi kartu pasien. Untuk mengetahui daftar id dapat dilihat pada Referensi.

4.7. Pedoman Untuk Pasien

Pasien merupakan pengguna yang berobat di klinik. Pasien hanya dapat menggunakan website setelah melakukan pendaftaran langsung di petugas pendaftaran. Petugas akan memberikan username dan password default untuk pasien agar pasien dapat melakukan login ke sistem. Pasien dapat mengganti username dan passwordnya setelah melakukan login. Meskipun pasien dapat login, namun pasien tidak dapat menggunakan fasilitas-fasilitas yang terdapat dalam situs ini. Pasien hanya dapat mengakses dashboard pasien yang hanya akan muncul apabila pasien melakukan login.

The screenshot shows the 'View Kartu Pasien #1' page. It includes a sidebar with 'Operations' and 'List Kartupasien'. The main content area displays patient information, transaction details, and a list of services.

View Kartu Pasien #1

Tgl Kunjungan	2012-06-07
Keluhan	sakit gigi
Tgl Selesai	2012-06-07
Id Status Kp	3
Id Penanganan	4
Id Tipelayanan	3
Id Dokter	1
Id Laboran	8
No Reg	1
Nmr	1
Catatan	gigi berlubang

Transaksi

Transaksi Penanganan

Displaying 1-1 of 1 result(s).

Id Itempenanganan	No Reg	Id Tipe Penanganan	Harga Penanganan
1	1	4	100000

Transaksi Layanan

Displaying 1-1 of 1 result(s).

Id Itemlayanan	No Reg	Id Tipelayanan	Harga Layanan
1	1	3	150000

Hasil Pemeriksaan Dokter

Pemeriksaan

Displaying 1-1 of 1 result(s).

Id Observasi	No Reg	Konsultan yang bertugas	Sumber Rujukan	Tgl Pertemuan Pasien	Tgl Periksa	Pemeriksa	Nama Dokter	Tingkat
1	1	Dokter 1	-	2012-06-07	2012-06-07	Dokter 1	Nama Dokter 1	-

Investigasi

Displaying 1-1 of 1 result(s).

Id Investigasi	No Reg	Oksimetri %	Glukosa Darah	Urinisasi	Permintaan FBC	Permintaan Pembekuan	Banyak HB	Banyak WCC	Banyak Trombosit	Hasil INR	Hasil Hematol Lain
1	1	23	23	baik							

Hasil Laboratorium

Hasil Lab Urinisasi

Displaying 1-1 of 1 result(s).

No Reg	Id Laburin	Berat Jenis	Reaksi	Proteina	Reduksi	Urobilin	Bilirubin	Lokosit	Eritrosit	Silinder Hialin	Silinder Berbutir	Sel Epitel	Kristal
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Di dashboard pasien terdapat demografi pasien yang login dan daftar kartu pasien dari kunjungan yang pernah dilakukan oleh pasien. Pasien dapat melakukan update terhadap demografinya. Pasien juga dapat melihat daftar kartu pasien secara lebih lengkap, seperti transaksi, hasil pemeriksaan dokter, dan hasil laboratorium.

4.7.1. Pasien Mengakses Kartu Pasien

Kartu pasien merupakan representasi rekam medis dari pasien. Untuk mengakses kartu pasien, dapat dilakukan dengan cara :

1. Login sebagai pasien.
2. Buka dashboard pasien.

3. Pilih List Kartu Pasien yang berada di side bar menu. Akan muncul daftar kartu pasien dari pasien yang melakukan login.
4. Pilih No Reg dari kartu pasien yang akan dilihat. Akan muncul data lengkap tentang kartu pasien, transaksi, hasil pemeriksaan dokter, dan hasil laboratorium.

4.7.2. Pasien Mengubah Demografi

Melalui dashboard pasien, pasien dapat mengubah data demografi pasien. Untuk mengubah demografi dapat dilakukan dengan :

1. Login sebagai pasien.
2. Buka dashboard pasien.
3. Pada side bar menu, pilih update pasien.

The screenshot displays the 'Open e-Health' web application interface. The browser address bar shows 'localhost/ehealth/pd/update/1'. The application has a top navigation bar with links: Home, Dashboard Ps, User, Pasien, Surat Rujukan, Penunjang Medis, Transaksi, Referensi, and User Guide. Below this is a sub-navigation bar with Home, Pasien, and 1 Update. The main content area is titled 'Update Pasien 1' and includes a note: 'Field dengan * harus diisi.' The form contains the following fields: Username (filled with 'pasien1'), Password (masked with dots), Nama Pasien (filled with 'Nama pasien 1'), Nama Ibu (filled with 'nama ibu pasien 1'), Tgl Reg (filled with '2012-07-02'), and Telp Kerja (empty).

DAFTAR PUSTAKA

- Greenes, R. A., Shortliffe E. H. 1990. Medical informatics: An Eemerging Aacademic Discipline and Institutional Priority. JAMA. 263:1115-1120.
- Handoyo, E., Agung B. P., Fuad N. S. 2008. Aplikasi Sistem Informasi Rumah Sakit Berbasis Web Pada Sub-Sistem Farmasi Menggunakan Framework Prado. Jurnal Teknologi Elektro. 13 Vol. 7 No. 1. Januari - Juni 2008.
- Hill, J.W. dan Powel, P. 2009. The National healthcare crisis: is eHealth a key Solution? Kelly School of Business, Indiana University Published by Elsevier Inc.
- Henderson, M., Behlen, F. M., dan Parisot, C. 2001. Integrating the Healthcare Enterprise: Aprimer, Part 4 The Role of Existing Standrads in IHE.
- Miharja, A.R. 2012. Skripsi: Jurusan Sistem Informasi Gunadarma.
- Pittman P. dan Holve E.. 2009. The Health Services Researcher of 2020: A Summit to Assess the Field's Workforce Needs. Health Research and Educational Trust. 44:6. December 2009.
- Shortliffe E. H, Perreault, L. E, Wiederhold G. dan Fagan, L. M. 1990. Medical Informatics: Computer Application in Health Care. Reading, MA: Addison-Wesley.
- Shortliffe, E. H. 1995. Medical Meet Medical Informatics Education. (<http://www-camis.stanford.edu/projects/smi-web/academics/jama-pulse.html> URL)
- Singer, Sara, S. Lin, A. Falwell, D. Gaba and L. Baker. 2009. Relationship of Safety Climate and Safety Performance in Hospitals. Health Research and Educational Trust. 44:2, Part I. April 2009.
- Surat Keputusan Menteri Kesehatan Nomor 983.MENKES/SK/XI/1992. Organisasi Rumah Sakit dan Pedoman Rumah Sakit Umum. Departemen Kesehatan R.I. Jakarta. 1992.
- Surat Keputusan Menteri Kesehatan Nomor 1202/MENKES/SK/VIII/2003. Indokator Indonesia Sehat 2010 dan Pedoman Penetapan Indikator Provinsi Sehat dan Kabupaten Kota/Sehat. Departemen Kesehatan R.I. Jakarta. 2003.
- Tang, P.C., dan Hammond, W.E. 1997. The Computer-based Patient Record: An Essential Technology for Healthcare, 2nd ed. Washington DC, National Academy Press, pp 1-20.
- URL: <http://www.yiiframework.com/doc/guide/1.1/id>, 28 Juni 2012
- URL: [http:// inamc.or.id/download/Manual%20Rekam%20Medis.pdf](http://inamc.or.id/download/Manual%20Rekam%20Medis.pdf), 30 Juni 2012

URL: <http://en.wikipedia.org/wiki/OpenEHR>, 30 Juni 2012

URL: <http://www.santolead.com/tentang-yii-framework.html>, 28 Juni 2012

URL: <http://www.javacreativity.com/2011/05/wamp-server-5-aplikasi-untuk-membuat.html>, 30 Juni 2012

URL: <http://blog.rajawebhost.com/pengertian-website-menurut-para-ahli.html>, 28 Juni 2012

URL: <http://nurichsan.blog.unsoed.ac.id/2010/11/19/metode-pengembangan-waterfall-prototyping/>, 18 Juli 2012

URL: <http://sabitlabscode.files.wordpress.com/menjelahiyiiiframework>, 30 Maret 2011

Veli Bicer, Gokce B. Laleci, Asuman Dogac, Yildiray Kabak. 2006. Artemis Message Exchange Framework: Semantic Interoperability of Exchanged Messages in the Healthcare Domain, Software Research and Development Centre Middle East Technical University (METU) 06531.

Werner, R.M., Bradlow, E.T., dan Asch, D.A. 2008. Does Hospital Performance on Process Measures Directly Measure High Quality Care or Is It a Marker of Unmeasured Care?. Health Research and Educational Trust. 43:5, Part I. October 2008.