

MAKALAH SEMINAR TUGAS AKHIR
**IMPLEMENTASI SISTEM INFORMASI RUMAH SAKIT
UNTUK SUBSISTEM LABORATORIUM**

Dwi Mardiatmo N H*, Agung B.P.**, Eko Handoyo**

Abstrak - Teknologi informasi merupakan salah satu teknologi yang sedang berkembang dengan pesat pada saat ini. Dengan kemajuan teknologi informasi, pengaksesan terhadap data atau informasi yang tersedia dapat berlangsung dengan cepat, efisien serta akurat. Penelitian ini bertujuan untuk memberikan gambaran sebuah model sistem informasi rumah sakit menggunakan layanan web melalui pembangunan sebuah aplikasi sistem informasi rumah sakit untuk subsistem laboratorium. Sistem Informasi Rumah Sakit ini sebenarnya merupakan gabungan dari beberapa subsistem, diantaranya subsistem pasien dan dokter, subsistem laboratorium dan subsistem farmasi yang digabungkan menjadi satu. Dengan aplikasi ini, pengguna dapat dengan mudah memperoleh pelayanan dan informasi seluruh kegiatan yang ada khususnya dalam hal manajemen laboratorium pada rumah sakit dimanapun dan kapanpun mereka berada secara on-line.

Aplikasi ini dibuat berbasis web dengan menggunakan framework Prado berbasis bahasa pemrograman PHP dan MySQL sebagai basis datanya. Dalam pembuatannya, aplikasi ini disesuaikan dengan kebutuhan rumah sakit secara umum. Sehingga fasilitas yang ada pada sistem ini benar-benar dapat mendukung semua kegiatan yang ada pada rumah sakit. Tentu saja pada awalnya dilakukan analisa kebutuhan untuk suatu sistem informasi rumah sakit agar penyediaan informasi dapat dilakukan dengan berbasis web. Analisis dan desain sistem dalam pembuatan sistem informasi rumah sakit ini menggunakan metode pendekatan berorientasi objek.

Aplikasi Sistem Informasi Rumah Sakit ini dapat digunakan sebagai sarana penyedia layanan dan informasi bagi penggunanya baik untuk dokter, staf dan karyawan, maupun pasien suatu rumah sakit dimanapun dan kapanpun mereka berada. Pengguna mendapatkan semua informasi yang akurat karena informasi yang tersedia senantiasa diperbaharui. Aplikasi ini akan lebih baik jika memiliki keamanan data yang tinggi dan penambahan modul lain.

Kata kunci: Sistem Informasi Rumah Sakit, Framework Prado

I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah mengubah manusia dalam menyelesaikan semua pekerjaannya, tidak hanya dalam pekerjaannya saja

tetapi dalam segala aspek kehidupan manusia, seperti pada saat pencarian informasi. Jika dahulu manusia mencari informasi sebatas pada buku, media cetak, maupun secara lisan, sekarang lebih banyak mencari informasi tersebut melalui internet. Secara tidak langsung dapat dikatakan semua serba terkomputerisasi.

Rumah sakit sebagai salah satu institusi pelayanan umum membutuhkan keberadaan suatu sistem informasi yang akurat dan andal, serta cukup memadai untuk meningkatkan pelayanannya kepada para pasien serta lingkungan yang terkait lainnya. Dengan lingkup pelayanan yang begitu luas, tentunya banyak sekali permasalahan kompleks yang terjadi dalam proses pelayanan di rumah sakit. Banyaknya variabel di rumah sakit turut menentukan kecepatan arus informasi yang dibutuhkan oleh pengguna dan lingkungan rumah sakit.

Pengelolaan data di rumah sakit merupakan salah satu komponen yang penting dalam mewujudkan suatu sistem informasi di rumah sakit. Pengelolaan data secara manual, mempunyai banyak kelemahan, selain membutuhkan waktu yang lama, keakuratannya juga kurang dapat diterima, karena kemungkinan kesalahan sangat besar. Dengan dukungan teknologi informasi yang ada sekarang ini, pekerjaan pengelolaan data dengan cara manual dapat digantikan dengan suatu sistem informasi dengan menggunakan komputer. Selain lebih cepat dan mudah, pengelolaan data juga menjadi lebih akurat.

Tugas akhir ini membahas perancangan dan pengembangan sistem informasi rumah sakit untuk subsistem laboratorium yang berbasis web. Aplikasi ini digunakan untuk membantu bagian laboratorium di rumah sakit untuk menyediakan pelayanan dan informasi kesehatan bagi pasien sehingga dapat mendukung kelancaran kegiatan manajemen di suatu rumah sakit sehari-hari.

1.2 Tujuan

Merancang dan membangun sebuah aplikasi sistem informasi rumah sakit untuk manajemen laboratorium berbasis web dengan menggunakan framework Prado yang menggunakan bahasa pemrograman PHP dengan MySQL sebagai basis datanya. Sistem informasi ini diharapkan dapat bermanfaat untuk membantu dalam mengolah data di laboratorium rumah sakit.

1.3 Batasan Masalah

- Dalam tugas akhir ini pembahasan dibatasi pada:
1. Membahas hanya pada bagian manajemen laboratorium sistem informasi rumah sakit.
 2. Sistem yang dibangun menggunakan tingkat keamanan standar yaitu menggunakan *session*.

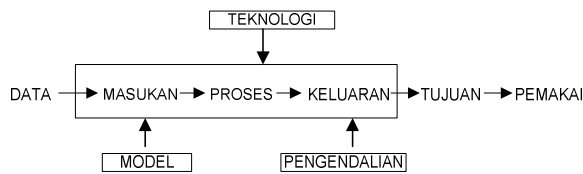
* Mahasiswa Teknik Elektro UNDIP

** Dosen Teknik Elektro UNDIP

II DASAR TEORI

3.1 Konsep Dasar Sistem Informasi

Sistem informasi adalah sekumpulan komponen pembentuk sistem yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu. Dalam sistem informasi diperlukannya klasifikasi alur informasi, hal ini disebabkan keanekaragaman kebutuhan akan suatu informasi oleh pengguna informasi. Kriteria dari sistem informasi antara lain, fleksibel, efektif dan efisien^[1]. Secara garis besar komponen yang terkait dengan suatu sistem informasi dapat dilihat pada Gambar 2.1 berikut.



Gambar 2.1 Komponen sistem informasi

3.2 Sistem Informasi Rumah Sakit

Sistem Informasi Rumah Sakit (SIRS) adalah suatu tatanan yang berurusan dengan pengumpulan data, pengelolaan data, penyajian informasi, analisis dan penyimpulan informasi serta penyampaian informasi yang dibutuhkan untuk kegiatan rumah sakit^[1].

Sebuah sistem informasi rumah sakit idealnya mencakup integrasi fungsi-fungsi klinikal (medis), keuangan, serta manajemen yang nantinya merupakan subsistem dari sebuah sistem informasi rumah sakit. Subsistem ini merupakan unsur dari sistem informasi rumah sakit yang tugasnya menyiapkan informasi berdasarkan fungsi-fungsi yang ada untuk menyederhanakan pelayanan pada suatu rumah sakit.

1. Subsistem Sistem Informasi Rumah Sakit

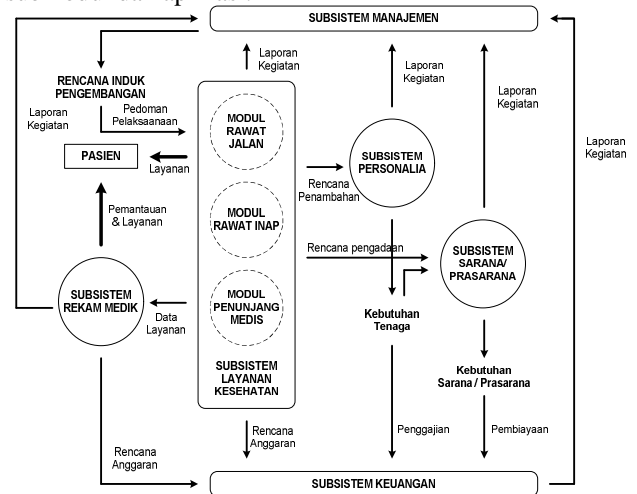
Fungsi utama dari rumah sakit yang pada umumnya adalah pelayanan kesehatan, serta pasien sebagai objek dari fungsi utama rumah sakit, dukungan operasional berupa tenaga kerja, keuangan, sarana dan prasarana, serta sistem manajemen yang dibutuhkan untuk mengelola suatu rumah sakit. Maka berdasarkan pertimbangan tersebut suatu sistem informasi rumah sakit terdiri dari beberapa subsistem sebagai berikut :

1. Subsistem Layanan Kesehatan. Subsistem Rekam Medis.
2. Subsistem Personalia.
3. Subsistem Keuangan.
4. Subsistem Sarana/Prasarana.
5. Subsistem Manajemen Rumah Sakit.

Subsistem tersebut diatas kemudian dijabarkan lagi ke dalam modul-modul yang sifatnya lebih spesifik. Subsistem Layanan Kesehatan dapat dijabarkan lebih lanjut menjadi :

1. Modul Rawat Jalan.
2. Modul Rawat Inap.
3. Modul Layanan Penunjang Medis.

Skema rancang bangun SIRS secara global ini dapat dilihat pada Gambar 2.2. Pada gambar tersebut diberikan contoh hubungan antara satu subsistem dengan subsistem lainnya. Rancangan global SIRS berisi penjabaran SIRS menjadi subsistem, modul, submodul dan aplikasi.



Gambar 2.2 Rancangan global sistem informasi rumah sakit

3. Sistem Informasi Rumah Sakit Subsistem Laboratorium

Pelayanan Laboratorium merupakan salah satu pelayanan penunjang di lingkungan rumah sakit dalam menjalankan fungsinya untuk memberikan pelayanan kepada masyarakat.

Divisi Laboratorium bertanggung jawab terhadap pemberian jasa pelayanan yang berhubungan dengan pemeriksaan penunjang medis laboratorium baik pemeriksaan medis, patologi, dan mikrobiologi bagi seorang pasien. Selanjutnya saat ini Divisi Laboratorium juga diberikan tanggung jawab sebagai pengelola Bank Darah.

Pada masa yang akan datang beberapa konsep baru telah disepakati untuk digunakan di lingkungan Divisi Laboratorium. Hal pertama yang diperkenalkan dan akan dilaksanakan adalah order manajemen yaitu pemanfaatan pelayanan permintaan dan penyampaian hasil pemeriksaan dengan memanfaatkan fasilitas komputer secara *online*. Fasilitas ini dapat dimanfaatkan oleh semua unit pengguna. Status atau proses permintaan layanan termasuk hasil pemeriksaan dapat dipantau / dilihat langsung melalui fasilitas komputer.

3.3 Framework PRADO

Di dalam pengembangan perangkat lunak, suatu *framework* digambarkan sebagai suatu struktur pendukung dimana perancangan perangkat lunak yang lain dapat terorganisir dan dikembangkan^[2]. Suatu *framework* dapat meliputi program pendukung, kumpulan kode-kode program (*libraries*), suatu bahasa *scripting*, atau perangkat lunak lain untuk membantu mengembangkan dan menggabungkan komponen-komponen yang berbeda menjadi satu dari suatu perancangan perangkat lunak.

Prado adalah sebuah *framework* pemrograman berbasis komponen dan *event-driven* untuk pengembangan aplikasi web pada PHP 5. PRADO merupakan singkatan dari PHP *Rapid Application Development Object-oriented Framework* ini dibuat oleh Qiang Xue dan telah menjadi pemenang dalam Zend PHP 5 *Coding Contest*.

Teknik yang digunakan *framework* Prado sangatlah berbeda. Pembangunan aplikasi web menggunakan Prado melibatkan banyak komponen (yang telah dibuat oleh pengembangnya), *men-setting* properti, dan memberikan tugas pada komponen berupa *event*. Jika seorang programmer mempunyai pengalaman dengan pemrograman desktop menggunakan Visual Basic atau Delphi, maka pemrograman web dengan menggunakan *framework* Prado sangatlah serupa.

Sebuah komponen Prado adalah kombinasi file spesifikasi (ditulis dengan bahasa XML), *template* HTML, dan *page class*. Komponen-komponen Prado digabungkan untuk membangun komponen yang lebih besar atau halaman web yang utuh. Prado membutuhkan PHP 5 dengan Simple XML dan Simple PHP Library (SPL). Untuk web server dapat digunakan Apache HTTP Server maupun Windows IIS dan mendukung Sistem Operasi Windows Server 2003, Windows XP, Windows 2000, Mac OS X, FreeBSD, RedHat Linux, Fedora Linux, dan Gentoo Linux.

1. Keuntungan Menggunakan Framework Prado

Konsep Prado yang *component-based* dan *event-driven* memberikan banyak keuntungan bagi programmer web. Berikut keuntungan dengan menggunakan Prado:

- Reusability*, komponen-komponen dalam Prado dapat digunakan ulang.
- Ease of use*, komponen-komponen dalam Prado sangat mudah digunakan. Komponen juga dapat dibuat sendiri dengan menurunkan *class* yang sudah ada sesuai dengan kebutuhan.
- Robustness*, Prado membebaskan pengembang program (*developer*) dari kejenuhan dalam menulis kode-kode. Semua kode ditulis dalam format objek, method, dan properti. Tidak seperti pemrograman PHP yang biasa digunakan.
- Performance*, Prado menggunakan teknik *cache* untuk menjamin *performance* aplikasi. Dengan adanya *cache* ini, ia tidak perlu mem-*parser* ulang kode XML yang dibuat.
- Team Integration*, Prado memisahkan *business logic* dan *presentation logic*. Yang dimaksudkan adalah pembuatan layout tampilan (*template*) dengan kode program (*class*). Pembuatan keduanya dilakukan pada file yang terpisah. Dengan demikian, aplikasi berbasis Prado dapat dilakukan dalam sebuah tim dengan personal yang berbeda.

2. Konsep Dasar Pemrograman Dengan Framework Prado

Di dalam *framework* Prado terdapat susunan file-file standar sebagai berikut :

- File `.htaccess`
File ini berfungsi untuk mengatur hak akses suatu isi folder, dalam hal melihat isi folder dari suatu browser. File ini hanya berisi kode :
`deny from all`
- File `application.spec`
File ini berfungsi untuk menspesifikasi aplikasi, antara lain konfigurasi atribut *default*, menentukan lokasi kode utama *framework* Prado, namespace yang digunakan ,dan lain-lain.
- File `page display (index.php)`
File ini merupakan file yang akan diakses oleh user, dalam file ini berisi letak file utama Prado dan letak file aplikasi sistem informasi rumah sakit.
- File `page template (.tpl)`
File inilah yang bertanggung jawab terhadap tampilan yang dilihat user.
- File `page class (.php)`
File ini lebih banyak berhubungan dengan proses bisnis atau biasa disebut dengan *business logic*.

3. Koneksi Prado Dengan Basisdata

Untuk melakukan koneksi ke database, Prado memanfaatkan *database abstract layer*, ADOdb. ADOdb adalah class yang ditulis menggunakan bahasa PHP yang berfungsi sebagai *data tier*, dan akan membantu mengatasi perbedaan antara penggunaan database. Cukup dengan menuliskan sebuah kode, maka koneksi dapat dilakukan ke berbagai macam database seperti MySQL, SQLite, SQL Server, Oracle, DB2, Interbase, PostgreSQL, dan sebagainya.

III PERANCANGAN PERANGKAT LUNAK

Aplikasi yang dibangun bernama Sistem Informasi Rumah Sakit untuk Subsistem Laboratorium. Subsistem Laboratorium merupakan salah satu bagian dari Sistem Informasi Rumah Sakit secara keseluruhan, tujuan dari pembagian ini adalah untuk mempermudah pemahaman Sistem Informasi Rumah Sakit yang begitu kompleks, sehingga dengan adanya pembagian ini diharapkan penanganan pada subsistem laboratorium dapat lebih detail dan efektif disamping itu juga menghemat waktu dibanding menangani sistem secara keseluruhan.

Divisi Laboratorium bertanggung jawab terhadap pemberian jasa pelayanan yang berhubungan dengan pemeriksaan penunjang medis laboratorium baik pemeriksaan medis, patologi, mikrobiologi, dan sebagai pengelola bank darah bagi seorang pasien.

Aplikasi Sistem Informasi Rumah Sakit ini pada sisi server merupakan sebuah aplikasi berbasis web (*web-based application*) yang berfungsi untuk menerima masukan dari aplikasi sisi klien dan juga berfungsi untuk mengatur basis data. Perancangan aplikasi sisi server ini menggunakan analisis pemodelan berorientasi objek.

3.1 Perancangan Aplikasi Berorientasi Objek

1. Use Case.

Uraian berikut ini adalah use case dari Sistem Informasi Rumah Sakit pada Subsistem Laboratorium. Kata yang bercetak tebal menggambarkan calon objek sedangkan kata yang bercetak miring menggambarkan operasi.

Persiapan Sistem

Sistem ini mempunyai empat jenis **pengguna**, yaitu **administrator sistem**, **dokter**, **staf lab**, dan **user**. Setelah sistem ter-*install*, **administrator sistem** yang mempunyai hak akses tertinggi dalam sistem perlu *menyediakan informasi-informasi* yang nantinya akan dimasukkan ke dalam sistem. **Administrator sistem** juga memiliki kewenangan untuk *menelola informasi-informasi* tersebut. **ID pengguna** dan **kata kunci** yang digunakan oleh **administrator sistem** untuk memasuki sistem dapat *diubah* untuk menjamin keamanan akses. Sebelum mengakses sistem, **pengguna** harus melakukan **register** terlebih dahulu yang kemudian *diaktifasi* oleh **administrator**, setelah itu **pengguna** dapat mengakses ke dalam sistem informasi rumah sakit sesuai dengan **role**-nya.

User

Setelah melakukan **registrasi**, **user** akan *mendapatkan ID pengguna* dan **kata kunci**. **Administrator** harus *mengaktifkan user* yang telah melakukan **registrasi** agar **user** tersebut dapat *mengakses* sistem. **User** yang sudah *diaktifkan* dapat *mengakses* sistem informasi rumah sakit dan dapat *ikut serta* dalam **aktifitas** di dalam sistem. **Kata kunci** yang digunakan oleh **user** juga dapat *diubah* untuk menjamin keamanan. **User** hanya dapat mencari dan melihat informasi **pasien** serta melihat **informasi antrian order** lab.

Staf Lab

Sama seperti **pengguna** lainnya, **Staf Lab** setelah *teraktifasi* kemudian *mendapatkan id* dan **kata kunci** yang digunakan untuk masuk ke dalam sistem.. Setelah **Staf Lab** melakukan **login** maka **Staf Lab** dapat *melakukan* berbagai macam **aktivitas** diantaranya melihat **informasi antrian order pasien**, *mengisikan* dan *mengedit permintaan test laboratorium pasien*, *mengisikan hasil test laboratorium medis*, *mengedit parameter test*, *mengisikan hasil test laboratorium patologi*, *mengisikan* dan *mengedit* permintaan pada **bank darah**, serta mencari dan melihat **informasi hasil test laboratorium** seorang **pasien**. **Staf Lab** juga dapat *mengubah kata kunci* milik mereka.

Dokter

Dokter juga *mendapatkan id* dan **kata kunci** sebagai hak akses ke dalam sistem setelah *melakukan registrasi* dan *teraktifasi*. Setelah **dokter** melakukan **login** maka **dokter** dapat *melakukan aktivitas* diantaranya melihat **data pasien**, melihat **informasi antrian order** lab dan melihat **informasi hasil test laboratorium** seorang **pasien**. **Dokter** juga dapat *mengubah kata kunci* milik mereka.

Administrator

Administrator memiliki **peranan** yang paling besar di dalam aktivitas sistem ini. **Administrator** memiliki **hak akses** yang paling lengkap dibandingkan user-user lainnya. **Administrator** mempunyai **kewajiban** dalam *mengatur administrasi user* diantaranya *mengaktifasi pengguna* yang sudah *melakukan registrasi* ke dalam sistem. **Administrator** juga berhak *menonaktifkan* seorang **user**. Untuk keperluan manajemen informasi **Administrator** dapat *menambahkan menu* baru atau *mengedit menu* dan **informasi** yang sudah ada sebelumnya. Selain itu **administrator** juga dapat *mengubah kata kunci* miliknya.

2. Identifikasi Objek.

Setelah proses dilanjutkan dengan indentifikasi calon objek kemudian diikuti oleh hasil seleksi calon objek. Berikut ini merupakan daftar spesifikasi atribut dari masing-masing objek atau kelas.

User = ID + password + role + status

Dokter = ID + password + role + status

Staf Lab = ID + password + role + status

Administrator = ID + password

Antrian order = tanggal order + nama pasien + divisi laboratorium + kode test + tanggal test + waktu test + nama dokter + nama petugas

Pemintaan test = tanggal order + nama pasien + divisi laboratorium + kode test + tanggal test + waktu test + nama dokter + nama petugas + diagnosa + catatan

Hasil Test = nama pasien + tanggal + waktu + parameter test

3. Pemodelan CRC

Setelah pengerjaan use case dan identifikasi objek, tahapan selanjutnya melakukan pemodelan kelas-tanggungjawab-kolaborator (CRC). Metode ini disusun berdasarkan identifikasi objek atau kelas. Objek merupakan instansiasi dari kelas, untuk mempermudah pada saat pembuatan program, tiap kelas diberi nama yang lebih sederhana.

4. Model hubungan Objek

Setelah melakukan pemodelan kelas-tanggungjawab-kolaborator maka dapat dibuat model hubungan antar objek dengan mengkaji kartu indeks model CRC, tanggung jawab dan kolaborator. Dari pembuatan kartu index CRC selanjutnya dapat dibuat diagram kelas. Diagram kelas memberikan gambaran tentang kelas-kelas apa saja yang perlu dibuat untuk membangun aplikasi, lengkap dengan atribut dan operasinya.

5. Model Tingkah Laku Objek (*Object Behavior*)

Pemodelan objek yang dilakukan antara lain melalui CRC dan model hubungan antar objek berfungsi untuk mempresentasikan elemen statis dari model analisis berorientasi objek. Langkah selanjutnya adalah memodelkan elemen dinamis dari model analisis, hal ini dilakukan dengan memodelkan tingkah laku objek, antara lain dengan membangun diagram runtun (*sequence diagram*) dan *statechart diagram*. *Sequence diagram* merupakan diagram interaksi yang menekankan urutan waktu dari sebuah pesan. Diagram ini menunjukkan kumpulan objek dan pesan yang dibawa maupun diterima objek tersebut. *Statechart diagram* merupakan diagram yang menggambarkan sebuah sistem secara dinamis. Model tingkah laku objek menunjukkan bagaimana sistem akan merespon kejadian atau stimulus eksternal.

IV ANALISA DAN PEMBAHASAN

4.1 Pengguna SIRS Subsistem Laboratorium

Pengguna dari sistem informasi rumah sakit subsistem laboratorium terdiri dari 4 jenis, yaitu :

1. Administrator, merupakan pemegang hak akses paling tinggi dalam sistem.
2. Dokter
3. Staf Lab
4. User

4.2 Tampilan antarmuka sistem

1. Antarmuka sistem

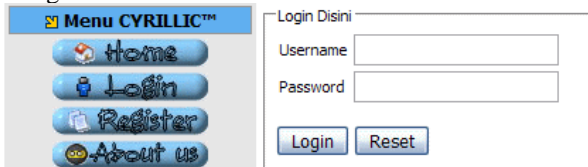
Aplikasi SIRS mempunyai halaman utama yang berisikan link ke form login yang digunakan untuk autentifikasi pengguna agar bisa mengakses sistem, link ke form registrasi yang digunakan untuk pendaftaran pengguna, serta link ke halaman about us yang memuat informasi pembuat aplikasi SIRS ini.



Gambar 4.1 Tampilan halaman muka

2. Antarmuka form login

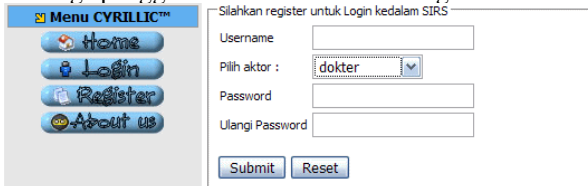
Antarmuka Login memuat form login yang dapat digunakan untuk tempat autentifikasi bagi pengguna. Setelah memasukkan username dan password yang benar, maka pengguna berhak untuk masuk dan mengakses SIRS.



Gambar 4.2 Tampilan halaman login

3. Antarmuka form registrasi

Antarmuka Registrasi memuat form registrasi yang dapat digunakan untuk tempat pendaftaran bagi seorang pengguna. Setelah melakukan registrasi maka seorang pengguna berhak untuk mengakses SIRS..



Gambar 4.3 Tampilan halaman registrasi

4. Informasi Antrian Order Laboratorium

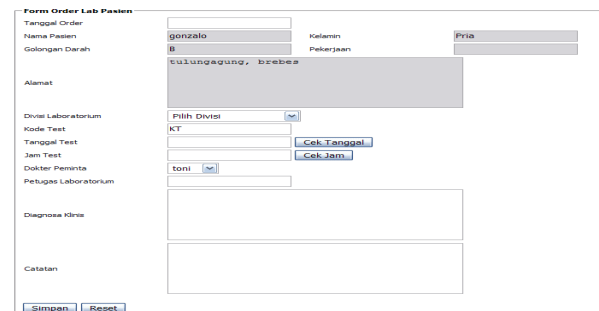
Antarmuka ini memuat informasi antrian permintaan tes pasien pada laboratorium tertentu yang dapat dilihat untuk semua tipe pengguna.



Gambar 4.4 Tampilan halaman informasi antrian order lab

5. Permintaan Test Laboratorium

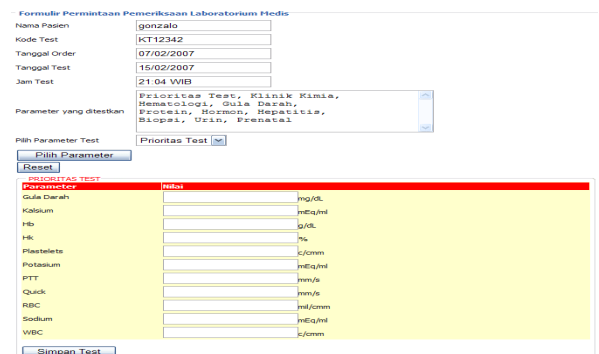
Antarmuka ini memuat form permintaan tes bagi seorang pasien.



Gambar 4.5 Tampilan halaman permintaan test lab

6. Input Hasil Test Laboratorium Medis

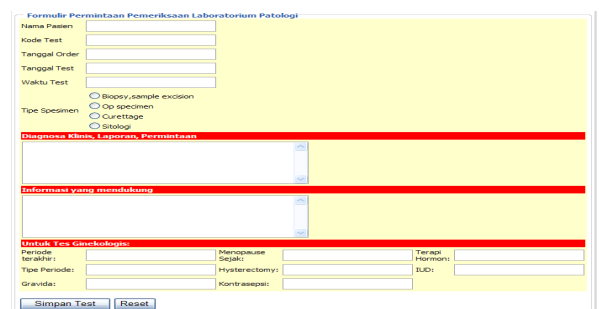
Antarmuka ini memuat form pengisian hasil tes pasien untuk laboratorium medis.



Gambar 4.6 Tampilan halaman input hasil test laboroairum medis

7. Input Hasil Test Laboratorium Patologi

Antarmuka ini memuat form pengisian hasil tes pasien untuk laboratorium patologi.



Gambar 4.7 Tampilan halaman input hasil test laboroairum patologi

8. Input Hasil Test Laboratorium Bakteriologi
Antamuka ini memuat form pengisian hasil tes pasien untuk laboratorium bakteriologi.

Gambar 4.8 Tampilan halaman input hasil test laboratoirum bakteriologi

9. Permintaan Darah
Antamuka ini memuat form permintaan darah bagi seorang pasien pada bank darah rumah sakit.

Gambar 4.9 Tampilan halaman permintaan darah

10. Informasi Hasil Test Laboratorium
Antamuka ini memuat form hasil tes laboratorium seorang pasien yang sudah melakukan tes pada salah satu laboratorium di rumah sakit.

Gambar 4.10 Tampilan halaman informasi hasil test laboratorium

4.3 Pengujian aplikasi

Aplikasi dinyatakan berhasil melewati pengujian apabila dalam percobaan-percobaan berikut tidak aplikasi tidak menjadi *error* atau melaksanakan perintah dengan salah. Pengujian aplikasi baik pada sisi klien maupun server menggunakan metode kotak hitam (*black box*).

Dalam metode pengujian *black box*, aplikasi diberikan berbagai macam kondisi masukan, kemudian keluaran yang dihasilkan sistem dibandingkan dengan keluaran yang diharapkan.

Prosedur pengujian aplikasi dilakukan seperti pada pengujian aplikasi sisi klien. Aplikasi dihadapkan pada berbagai kondisi yang memungkinkan untuk menimbulkan kesalahan, seperti misalnya pengisian *form* yang salah maupun tidak lengkap. Aplikasi harus mampu mengenali kondisi-kondisi tersebut dan menyiapkan fasilitas untuk mengatasinya, sehingga tidak menjadi *error* atau menjalankan fungsi yang salah.

Pengujian dilakukan antara lain pada form login. Adapun pengujiannya adalah sebagai berikut.

Tabel 4.1. Hasil pengujian black box proses login.

Input		Output	
Username	Password	Output diharapkan	Output Sistem
(-)	(-)	Pesan: Masukkan Username. Masukkan Password.	Pesan: Masukkan Username. Masukkan Password.
(-)	√	Pesan: Masukkan Username.	Pesan: Masukkan Username.
√	(-)	Pesan: Masukkan Password.	Pesan: Masukkan Password.
√(*)	√	Pesan: Login Gagal! Username atau Password anda salah.	Pesan: Login Gagal! Username atau Password anda salah.
√	√(**)	Pesan: Login Gagal! Username atau Password anda salah.	Pesan: Login Gagal! Username atau Password anda salah.
√	√	Tampilan Halaman index Pesan: Selamat datang	Tampilan Halaman index Pesan: Selamat datang

Keterangan :

- √ : Data diisi.
- √(*) : Username yang diinputkan tidak ada dalam basisdata.
- √(**) : Password yang diinputkan salah.
- (-) : Data kosong.

Pada pengujian proses login, ada tiga kondisi yang harus dipenuhi pada saat melakukan login. Ketiga proses itu yaitu yang pertama adalah melengkapi semua *field* login. Kedua adalah bahwa *username* terdapat dalam basis data. Ketiga adalah jika *password* yang diinputkan cocok dengan *password* yang tersimpan dalam basis data. Proses query hanya akan dijalankan apabila ketiga syarat tersebut dipenuhi. Apabila syarat tidak terpenuhi maka proses login tidak akan berhasil dan menampilkan pesan peringatan pada pengguna.

Hasil pengujian menggunakan metode diatas, aplikasi tidak menjadi *error* atau menampilkan keluaran yang salah. Semua kondisi penggunaan telah diujikan dan berhasil, dengan demikian aplikasi telah berhasil melewati pengujian.

V PENUTUP

5.1 Kesimpulan

Dari Tugas Akhir pembuatan aplikasi Sistem Informasi Rumah Sakit Subsistem Laboratorium ini dapat diambil beberapa kesimpulan antara lain :

1. Berdasarkan hasil pengujian dengan metode kotak hitam (*black box*), aplikasi berbasis web yang dibangun yaitu SIRS Subsistem Laboratorium telah sesuai dengan yang diharapkan dan dapat berfungsi dengan baik.
2. Aplikasi SIRS Subsistem Laboratorium ini berfungsi sebagai pendukung dalam kegiatan pelayanan laboratorium di rumah sakit.
3. Dengan konsep *framework* Prado yang berbasis komponen dan event driven, Prado memberikan keuntungan yang banyak dalam pengembangan aplikasi berbasis web.
4. Dari hasil pengujian dapat disimpulkan bahwa aplikasi SIRS Subsistem Laboratorium ini telah sesuai dengan fungsi-fungsi yang dimiliki oleh laboratorium di suatu rumah sakit pada umumnya.

5.2 Saran

1. Aplikasi SIRS Subsistem Laboratorium masih jauh dari sempurna, maka perlu dikembangkan lebih lanjut dengan penambahan modul atau fasilitas penunjang lain misalnya fasilitas akuntansi (*billing*) rumah sakit, fasilitas pelayanan bedah pasien, fasilitas pemeliharaan sistem seperti *backup* dan *restore* basisdata, pembuatan laporan, dan lain sebagainya sesuai dengan kebutuhan umum suatu rumah sakit.

DAFTAR PUSTAKA

- [1] Sabarguna, MARS, Dr. dr. H. Boy S., *Sistem Informasi Rumah Sakit*, Penerbit Konsorsium Rumah Sakit Jateng - DIY, 2005
- [2] Siswoutomo, Wiwit, *PHP Enterprise Kiat Jitu Membangun Web Skala Besar*, Penerbit PT. Elex Media Komputindo Jakarta, 2005
- [3] Kadir, A., *Konsep dan Tuntunan Praktis Basis Data*, Penerbit Andi Yogyakarta, 1999
- [4] Fowler, Martin, *UML Distilled Edisi 3 Panduan Singkat Tentang Bahasa Pemodelan Objek Standar*, Penerbit Andi Yogyakarta, 2005
- [5] Suhendar, A, S.Si, Hariman Gunadi S.Si., MT., *Visual Modeling Menggunakan UML dan Rational Rose*, Penerbit Informatika Bandung, 2002
- [6] Prasetyo, D. D., *Kolaborasi PHP dan MySQL untuk Membuat Web Database yang Interaktif*, PT. Elex Media Komputindo Jakarta, 2003
- [7] Siswoutomo, Wiwit, *Membangun Web Service Open Source Menggunakan PHP*, Penerbit PT. Elex Media Komputindo Jakarta, 2005
- [8] Azis, M.Kom, Ir. M. Farid, *Object Oriented Programming Dengan PHP5*, Penerbit PT. Elex Media Komputindo Jakarta, 2005

- [9] Siswoutomo, Wiwit, *PHP Undercover Mengungkap Rahasia Pemrograman PHP*, Penerbit PT. Elex Media Komputindo Jakarta, 2005
- [10] Siswoutomo, Wiwit, *Membuat Aplikasi Database Berbasis Web*, Penerbit PT. Elex Media Komputindo Jakarta, 2005
- [11] Azis, M.Kom, Ir. M. Farid, *Pemrograman PHP4 Bagi Web Programmer*, Penerbit PT. Elex Media Komputindo Jakarta, 2001



Dwi Mardiatmo N H

(L2F001589) lahir di Semarang, 2 Maret 1983. Menempuh pendidikan dasar di SDN Cakra Madya Dwipa lulus tahun 1995, kemudian melanjutkan ke SLTPN 21 Semarang lulus tahun 1998, dilanjutkan lagi di SMUN 3 Semarang lulus pada tahun 2001, dan sampai saat ini

masih menyelesaikan studi S1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang Konsentrasi Informatika dan Komputer.

Menyetujui dan Mengesahkan,

Pembimbing I,

Agung Budi P., S.T., MIT.

NIP. 132 137 932

Tanggal

Pembimbing II,

Eko Handoyo, S.T., M.T.

NIP. 132 309 142

Tanggal