

## **Analisa dan Perancangan Sistem Informasi Perawatan Pasien pada RSIA (Rumah Sakit Ibu dan Anak) dengan menggunakan pendekatan berorientasi aspek**

**Anofrizen**

Jurusan Sistem Informasi, Fakultas Sains dan Teknologi UIN SUSKA Riau  
Jl. HR. Soebrantas KM.15 Panam, Pekanbaru-Riau

### **ABSTRAK**

Penelitian ini difokuskan pada analisa dan perancangan Sistem Informasi Perawatan Pasien pada RSIA (Rumah Sakit Ibu dan Anak) dengan menggunakan pendekatan berorientasi aspek. Permasalahan yang terdapat pada rumah sakit ini yaitu proses perawatan pasien rawat jalan dan pasien rawat inap, tagihan pasien serta laporan pasien yang belum efektif dan efisien seperti membutuhkan waktu yang lama saat melakukan proses transaksi tersebut. Untuk mengatasi hal tersebut dirancanglah sistem informasi RSIA dengan menggunakan pendekatan berorientasi aspek. Metode penelitian yang digunakan dalam penelitian ini terdiri dari tahap analisa dan perancangan. Tahap analisa dan perancangan dilakukan dengan menggunakan bantuan UML (*Unified Modelling Language*). Pada tahap analisa dan perancangan diperlihatkan perbedaan pendekatan berorientasi objek dan pendekatan berorientasi aspek dan untuk kasus RSIA ini juga dihasilkan *pseudocode* untuk 2 modul yaitu perawatan pasien rawat inap dan perawatan pasien rawat jalan.

**Kata kunci :** Berorientasi aspek, *MySQL*, PHP, Sistem Informasi, UML

### **ABSTRACT**

*This study focused on the analysis and design of Patient Care Information System on RSIA (Mother and Child Hospital) by using aspect oriented approach. Issues contained in this hospital is the process of outpatient care and hospitalization of patients, patient billing and reports of patients who have not been effective and efficient as it takes along time to process the transaction. To overcome this design RSIA information systems using aspect-oriented approach. The research method used in this study consisted of analysis and design phases. Phase analysis and design is done by using the help of UML (Unified Modelling Language). At this stage of analysis and design of object-oriented approach have shown and the differences in aspect-oriented approach hand for this case also generated RSIA pseudocode for the two modules name inpatient care and out patient care.*

**Keywords:** *aspect-oriented, PHP, MySQL, Information Systems, UML.*

### **PENDAHULUAN**

RSIA (Rumah Sakit Ibu dan Anak) Labuh Baru yang bertempat di jalan Durian NO. 45 merupakan pelayanan medis yang dilengkapi dengan fasilitas untuk melahirkan, pemeriksaan kehamilan, pemeriksaan ibu dan anak serta berada dibawah pengawasan dokter dan bidan senior. Dalam menjalankan tugas-tugasnya, karyawan RSIA masih menemukan beberapa masalah yang mengakibatkan lamanya pelayanan yang diberikan seperti pencatatan data pasien yang juga membutuhkan waktu yang sangat lama serta pelaporan penyakit pasien yang sulit dilakukan karena penyimpanan seluruh data-data pasien masih dilakukan secara manual, yaitu masih disimpan dalam bentuk kertas folio yang dimasukkan kedalam map pasien.

Sehubungan dengan peningkatan pelayanan rumah sakit akan dilakukan secara terus menerus, maka perangkat lunak yang dibutuhkan adalah perangkat lunak yang mudah untuk dikembangkan pada masa yang akan datang.

Pemrograman berorientasi aspek merupakan paradigma pemrograman yang relatif baru, diperkenalkan sebagai hasil dari penelitian yang dilakukan oleh *Gregor Kiczales* di *Xerox's Palo Alto Research Center (PARC)*. Paradigma ini dikembangkan sebagai salah satu solusi untuk persoalan *separating crosscutting concerns* pada kode program. Dengan pendekatan pemrograman berorientasi aspek, persoalan didekomposisi menjadi kumpulan kelas (*class*) dan aspek (*aspect*). Kelas mewakili komponen-

komponen yang memiliki peran fungsional dalam domain persoalan sistem perangkat lunak yang akan dikembangkan. Sedangkan aspek adalah bagaimana memandang persoalan pemrograman dengan konsep fungsionalitas terhadap suatu kasus

### Sistem Informasi

Menurut Kristanto (2003). Sistem Informasi dan Aplikasinya, menyatakan bahwa: "Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu". Sistem juga merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan (*output*).

### Aspect Oriented Programming (AOP)

AOP merupakan metode pemrograman yang menghadirkan konsep untuk memotong silang (*crosscut*) permasalahan *tangling* dan *scattering*. *Tangling* dalam artian bahwa terdapat hal-hal lain didalam suatu *object* yang sebenarnya tidak berhubungan langsung, namun hanya merupakan *technical concerns* semata.

AOP merupakan penyempurna dari metode OOP. Bila kita bandingkan dengan *Object-Oriented Programming* (OOP), *security* dijadikan sebagai sebuah *class*, dan tidak dapat digunakan *crosscutting concern*. Dengan semakin dikenalnya AOP, kemudian muncul Unified Modeling Language (UML) yang menyediakan sintaks grafik untuk membuat model berbasis objek

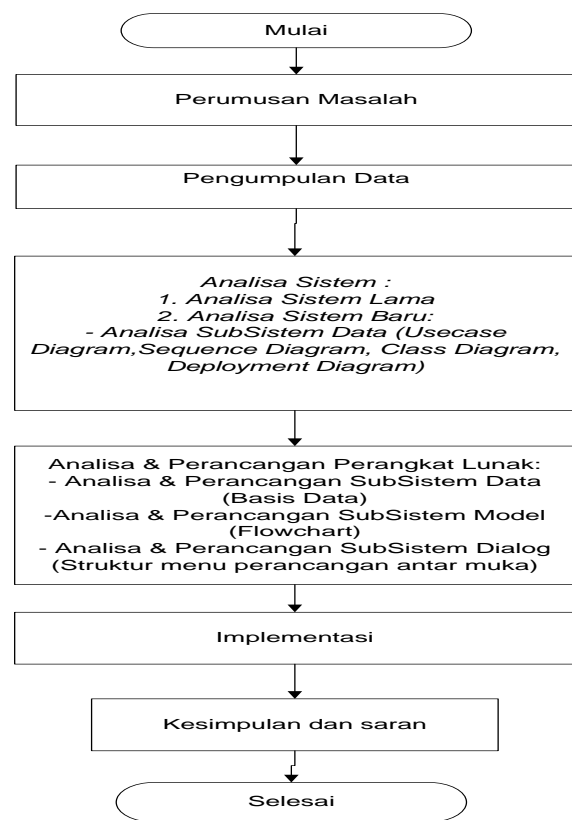
### Komponen Aspect Oriented Programming (AOP)

1. *Jointpoint* adalah suatu titik pengeksekusian pada aplikasi, misalnya pada saat pemanggilan method, inialisasi class, field assignment atau inialisasi object.
2. *Advice* adalah kode yang akan di eksekusi oleh jointpoint. Ada beberapa macam advice misal: before advice dieksekusi sebelum jointpoint, after advice di eksekusi setelah jointpoint.
3. *Poincut* adalah kumpulan beberapa jointpoint yang digunakan untuk

mendefinisikan kapan advice akan dijalankan.

4. *Aspect* adalah kombinasi antara advice dan poincut. Hasil kombinasi ini yang akan menghasilkan logic yang harus di eksekusi oleh aplikasi,
5. *Target* adalah object yang dimodifikasi oleh AOP. Misalkan ada method "setting" yang ada akan ditambahkan advice. Maka method "setting" disebut target.

## BAHAN DAN METODE



Gambar 2. Flowchart tahapan penelitian

### Pengumpulan Data

Data yang diambil merupakan data hasil wawancara dengan pihak rumah sakit serta data kuesioner dari keluarga pasien rawat inap yang ada RSIA (Rumah Sakit Ibu dan Anak) Labuh Baru.

Teknik pengambilan sampel yang digunakan adalah teknik pengambilan sampel random dengan menggunakan cara teknik

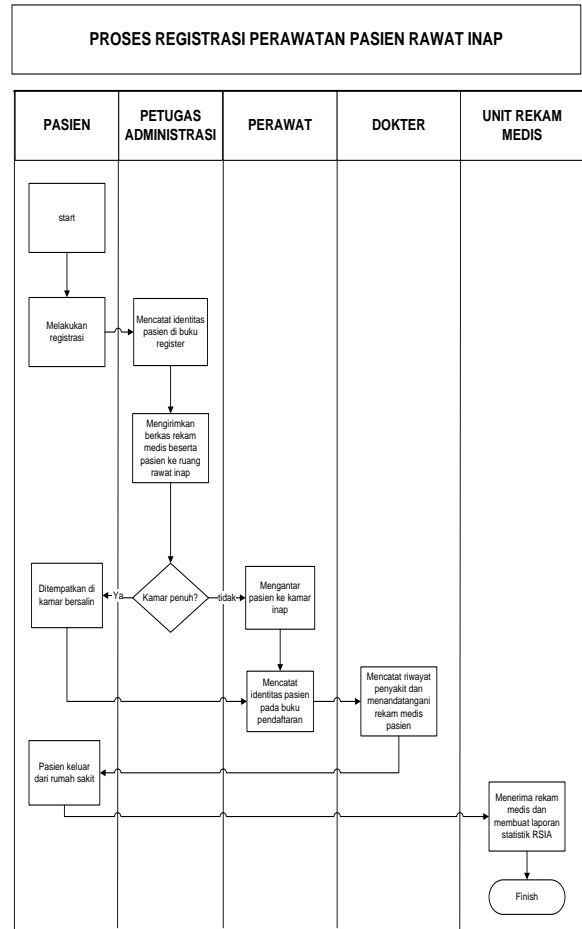
sampling kebetulan dimana teknik ini digunakan apabila pemilihan anggota sampelnya dilakukan terhadap orang atau benda yang kebetulan ada atau dijumpai.

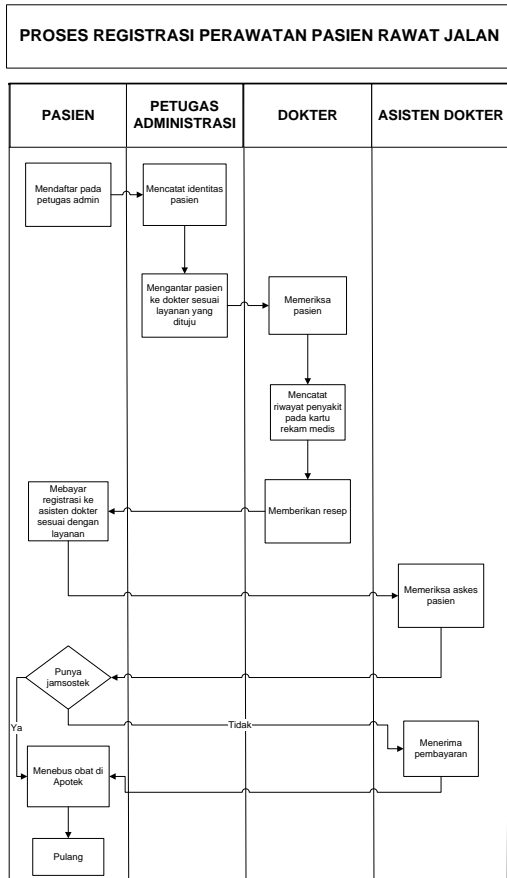
**HASIL DAN PEMBAHASAN**

**Analisa Sistem**

proses pencatatan data perawatan pasien rumah sakit ibu dan anak di Labuh Baru masih dilakukan dengan cara manual mulai dari pencatatan perawatan data pasien, data harga dan laporan data pasien maupun pegawai masih dicatat dalam suatu kertas folio dan disimpan dalam map pasien yang dilakukan oleh petugas administrasi. Hal ini menyebabkan tidak optimalnya pelayanan terhadap semua pihak yang mempunyai keterkaitan dengan RSIA tersebut, baik masyarakat yang berobat maupun instansi yang berkepentingan

Berikut akan diuraikan secara umum gambaran prosedur yang berkaitan dengan analisa dan perancangan sistem perawatan pasien rumah sakit ibu dan anak, secara umum, dibawah ini adalah gambar alur proses registrasi pasien rawat inap dan rawat jalan





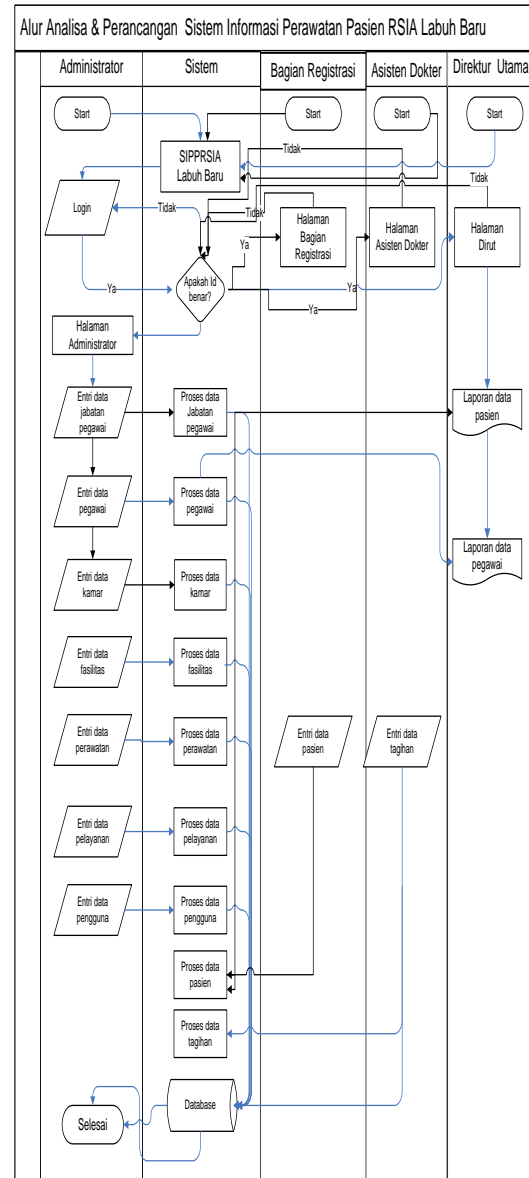
**Perancangan Sistem Baru**

Sistem diciptakan untuk membantu manusia memecahkan berbagai permasalahan, terutama masalah yang rumit dan jenis yang banyak. Analisa dan perancangan yang akan dibangun adalah analisa dan perancangan suatu sistem informasi yang memanfaatkan komputer sebagai perangkat utama pemrosesan. Manusia bertindak sebagai pengatur, pengoperasi, serta pengendali utama perangkat tersebut. Sistem berjalan setelah data masukan (*input*) diberikan. Data masukan (*input*) dalam sistem ini seperti Data Jabatan, Data Pegawai, Data Kamar, Data Fasilitas, Data Perawatan, Data Pelayanan, Data Pengguna, Data Tagihan dan Data ubah *password*. Setiap data pasien yang dimasukkan harus disimpan. Data pasien akan dimasukkan ke dalam sistem, maka sistem akan mengolah data pasien yang diinputkan, sehingga keluar data akhir berupa data laporan pasien secara keseluruhan. Data ini akan diarsipkan di dalam *database* dan jika dibutuhkan sewaktu-waktu bisa dicari kembali dengan cepat dan akurat. Hasil yang

diinginkan dari sistem ini adalah berupa informasi mengenai data laporan pasien rawat inap dan pasien rawat jalan RSIA Labuh Baru.

Tabel 2. Variabel Penelitian

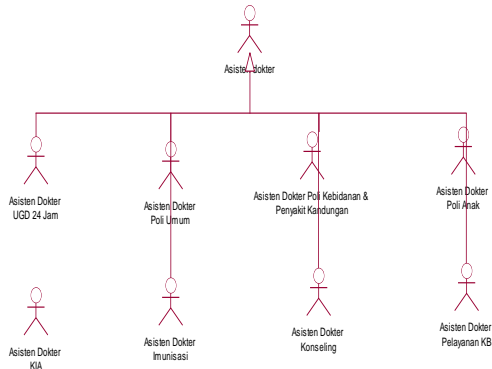
Untuk memperjelas proses yang terjadi pada SIPPRSIA ini, dapat digambarkan dengan menggunakan *flowchart* sebagai berikut:



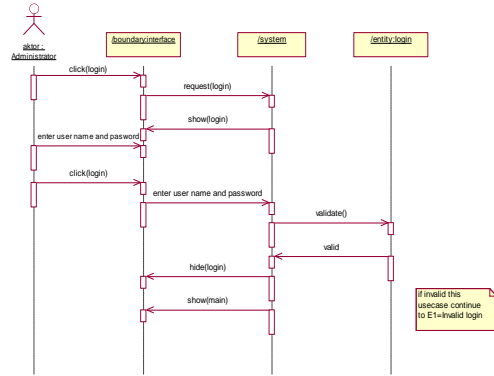
**Use Case Diagram**

*Use case diagram* terdiri dari *actor*, *use case* dan beserta hubungannya. Ada 4 *actor* yang terdapat dalam SIPPRSIA yaitu Administrator, Bagian Registrasi, Asisten Dokter dan Direktur Utama RSIA Labuh Baru. Karena dari masing-masing *actor* ini

mempunyai hak akses pengguna yang berbeda.

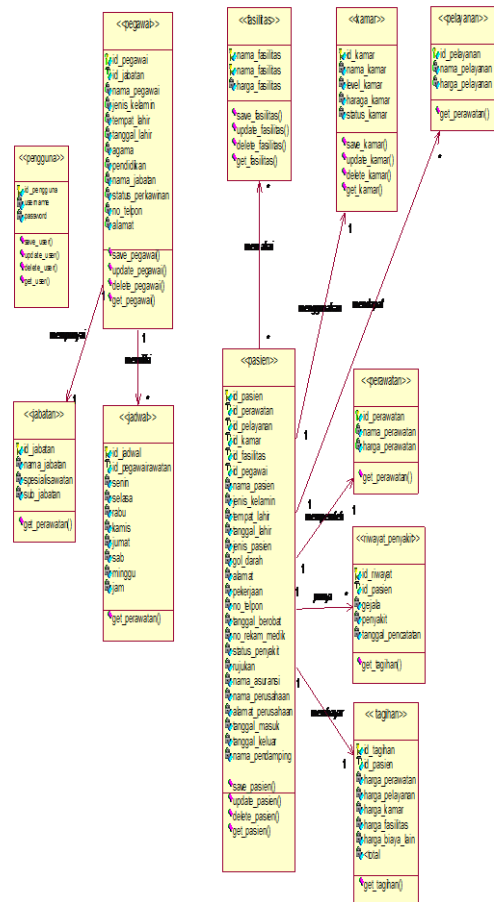
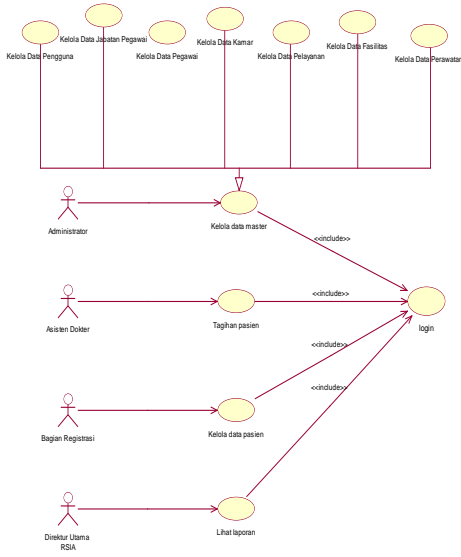


terkait).



**Class Diagram**

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek

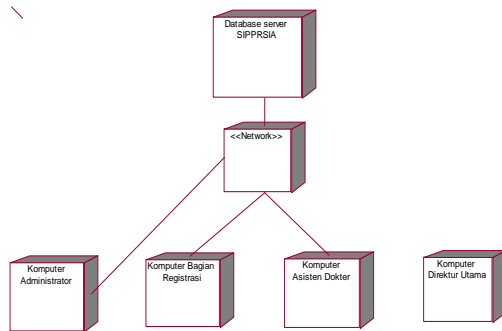


**Sequence Diagram**

Sequence diagram mendokumentasikan komunikasi/interaksi antar kelas-kelas. Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem termasuk user berupa pesanyang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang

## Deployment Diagram

*Deployment/physical diagram* menjelaskan hubungan antara *software* dan *hardware* yang ada serta menggambarkan detail bagaimana komponen di sebarkeandalam infrastruktur sistem, di mana komponen akan terletak pada mesin, server bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal.



## Penerapan Aspek

Aspek didenifisikan sebagai properti yang merentang pada sekelompok komponen fungsional. Sementara aspek-aspek tersebut dapat dianalisis secara relatif terpisah dari fungsionalitas dasar. Pada level implementasi aspek-aspek harus dikombinasikan bersama. *Aspect Oriented Programing (AOP)* dirancang untuk menangani masalah lintas sektoral. AOP tidak mengganti bahasa paradigma pemrograman yang ada melainkan bekerja dengan *Object Oriented Programming (OOP)* untuk meningkatkan kegunaan OOP. Aspek pengkapsulannya terpisah tetapi saling berhubungan satu dengan yang lainnya dan pada *class diagram* harus mempunyai *get save* di *method*. Sedangkan pada *Objek Oriented* malah sebaliknya.

Berikut adalah penerapan aspek yang terjadi pada perawatan pasien rawat inap dan rawat jalan pada RSIA Labuh Baru adalah :

	Pasien	Pelayanan	Total Tagihan
Registrasi Pasien	SavePasien() UpdatePasien()	SavePelayanan() UpdatePelayanan()	
Patients in Check	DeletePasien()	DeletePelayanan()	
Patients Out	GetPasien()	GetPelayanan()	GetTagihan()

Pasien	Pelayanan	Total Tagihan
*nama_pasien	nama_pelayanan	id_pasien
id_pasien	*id_pelayanan	id_pelayanan
id_pelayanan	id_fasilitas	id_pegawai
id_kamar	id_kamar	nama_tagihan
id_fasilitas	harga_pelayanan	harga_kamar
id_pegawai	level_kamar	harga_pelayanan
id_perawatan		harga_fasilitas
nama_pegawai		harga_perawatan
jenis_perawatan		*total_tagihan
no_rekam_medik		
status_penyakit		
gol_darah		
Alamat		
TTL		
jenis_kelamin		
Pekerjaan		
no_telp		
Rujukan		
tgl_masuk		
tgl_keluar		
ket_pasien		
harga_fasilitas		
harga_pelayanan		
harga_kamar		
nama_pendamping_pasien		
status_pendamping_pasien		
total_tagihan		

*Source Code* yang sederhana untuk pemeriksaan pasien rawat jalan menunjukkan daftar disederhanakan yang mengimplementasikan fungsi *Patients Out* sebagai aspek menggunakan AspectJ.

*Sourcode* untuk *Patients Out*:

```

1. public aspect Patients Out {
2.     ...
3.     public void Pasien.getPasien
4.     ()
5.     {
6.         // code to Patients Out
7.     }
8.     public void
9.     Pelayanan.getPelayanan ()
10.    {
11.        // code to getPelayanan
12.    }
13.    public void Total
14.    Tagihan.getTagihan ()
15.    {
16.        // code to generate an
17.        initial outstanding tagihan
18.    }
19.    ...
20. }
  
```

Baris 1 pada *Source Code Listing* untuk *Patients Out* menyatakan bahwafungsionalitas *Patients Out* sebagai aspek seperti ditunjukkan oleh kata kunci

aspek bahwa fungsional dilihat secara garis horizontal yang berisi serangkaian *intertype declarations* (kemungkinan untuk menulis fitur baru seperti atribut, *method*, dan hubungan kelas-kelas yang ada. Ada dua segmen dari sebuah *intertype declarations* yaitu segmen pertama adalah nama dari sebuah kelas yang ada dan segmen kedua adalah operasi yang sudah ada yang ingin ditambahkan ke dalam baris ke 3. Contoh misalnya:

```
Pasien.getPasien()
```

Menambahkan operasi ( ) *getPelayanan* keruang kelas Pasien. Perhatikan bahwa meskipun operasi ini adalah bagian dari kelas Pasien, didefinisikan di luar kelas pasien itu sendiri. Jadi orientasi aspek telah lolos dari pembatasan pada *method* kelas yang perlu didefinisikan dalam modularitas kelas tradisional

Analisa aspek yang mempengaruhi aktifitas rawat inap adalah:

	Pasien	Pelayanan	Kamar	Total Tagihan
Registrasi Patient	SavePasien()	SavePelayanan()	SaveKamar()	
Reserve Room	UpdatePasien()	UpdatePelayanan()	UpdateKamar()	
Check In Patient	DeletePasien()	DeletePelayanan()	DeleteKamar()	
Check Out Patient	GetPasien()	GetPelayanan()	GetKamar()	GetTagihan()

Pasien	Pelayanan	Kamar	Total Tagihan
*nama_pasien	nama_pelayanan	nama_kamar	id_pasien
id_pasien	*id_pelayanan	id_kamar	id_pelayanan
id_pelayanan	id_fasilitas	*level_kamar	id_pegawai
id_kamar	id_kamar	status_kamar	nama_tagihan
id_fasilitas	harga_pelayanan	harga_net	harga_kamar
id_pegawai	level_kamar		harga_pelayanan
id_perawatan			harga_fasilitas
nama_pegawai			harga_perawatan
jenis_perawatan			*total_tagihan
no_rekam_medik			
status_penyakit			
gol_darah			
Alamat			
TTL			
jenis_kelamin			
Pekerjaan			
no_telp			
Rujukan			
tgl_masuk			
tgl_keluar			
ket_pasien			
harga_fasilitas			
harga_pelayanan			
harga_kamar			
nama_pendamping_pasien			
status_pendamping_pasien			
total_tagihan			

*Sourcode* untuk *Check Out Patient*:

```
1. public aspect Check Out
Patient {
2.   ...
3.   public void Pasien.getPasien
   ()
4.   {
5.     // code to Check Out
Patient
6.   }
7.   public void
Pelayanan.getPelayanan ()
```

```
8.   {
9.     // code to getPelayanan
10.  }
11.  public void
Kamar.getKamar ()
12.  {
13.    // code to getKamar
14.  }
15.  Public void Total
Tagihan.getTagihan ()
16.  {
17.    // code to generate an
initial outstanding tagihan
18.  }
19.  ...
20. }
```

Baris 1 pada *Sourcode Listing* untuk *Check Out Patient* menyatakan bahwa fungsionalitas *Check Out Patient* sebagai aspek seperti ditunjukkan oleh kata kunci aspek bahwa fungsional dilihat secara garis horizontal yang berisi serangkaian *intertype declarations* (kemungkinan untuk menulis fitur baru seperti atribut, *method*, dan hubungan kelas-kelas yang ada. Ada dua segmen dari sebuah *intertype declarations* yaitu segmen pertama adalah nama dari sebuah kelas yang ada dan segmen kedua adalah operasi yang sudah ada yang ingin ditambahkan ke dalam baris ke 3. Contoh misalnya:

```
Pasien.getKamar()
```

Menambahkan operasi ( ) *getKamar* keruang kelas Pasien. Perhatikan bahwa meskipun operasi ini adalah bagian dari kelas Pasien, didefinisikan di luar kelas pasien itu sendiri. Jadi orientasi aspek telah lolos dari pembatasan pada *method* kelas yang perlu didefinisikan dalam modularitas kelas tradisional.

### ***Pseudocode* Perawatan Pasien Rawat Inap dan Rawat Jalan Menggunakan Pendekatan Berorientasi Aspek Oriented**

*Pseudocode* untuk Analisa & Perancangan Sistem Informasi Perawatan Pasien RSIA Labuh Baru dalam aspek *oriented* pada

perawatan pasien rawat jalan dan rawat inap adalah:

*Pseudocode 1 (aspek oriented)*

```

1. a, n, RI, RJ, getPasien(), getPelayanan(), getKamar(), getTagihan()
2. a = n
3. input → RI
4.   if n ∈ RI then Check Out Patient
5.     if Check Out Patient ← true then
6.       input → getPasien(), getPelayanan(), getKamar(), getTagihan()
7.     and
8.       output → RI
9.     else if
10.      RI → false
11.    end if

```

```

12. RJ
13.   if Patients Out → false then
14.     input → getPasien(), getPelayanan(), getTagihan() and
15.     output → RJ
16.   end if
17.   RI → false
18.   output → RJ or
19.   output → RI
20. end.

```

*Pseudocode untuk Analisa & Perancangan Sistem Informasi Perawatan Pasien RSIA Labuh Baru dalam objek oriented pada perawatan pasien rawat jalan dan rawat inap adalah:*

```

1. a, n, RI, RJ, d-kam, d-par, d-pel, d-perwtn, d-pas
2. a = n
3. input → RI
4.   if d-kam ← true then
5.     input → d-pas
6.     input → d-pel
7.     input → d-perwtn
8.     input → d-kamar
9.     input → d-fas
10.    input → d-tagihan and
11.    output → RI
12.  else if
13.   RJ
14.   If d-kam → false then
15.     input → d-pas
16.     input → d-pel
17.     input → d-perwtn
18.     input → d-tagihan and
19.     output → RJ
20.   end if
21.   RI → false
22.   output → RJ or
23.   output → RI
24. end.

```

Ket:

RI: Rawat Inap

RJ: Rawat Jalan

d-kam: data kamar

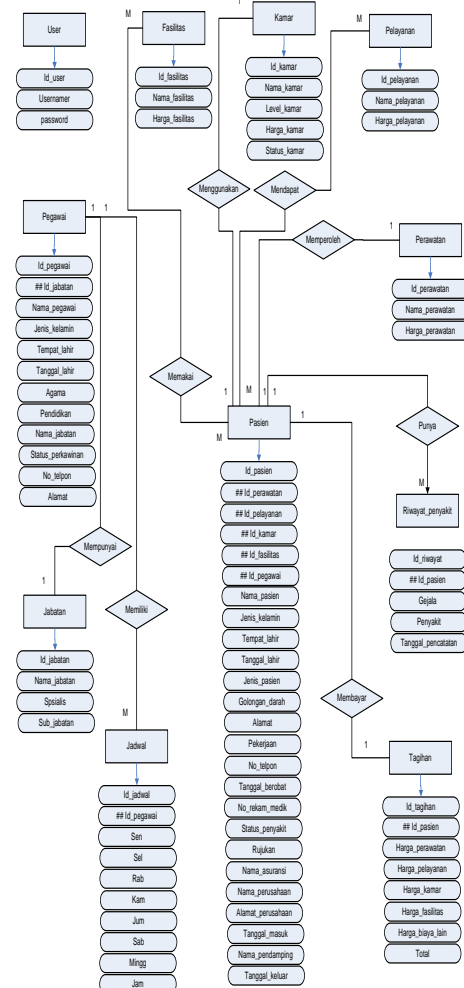
d-pel: data pelayanan

d-perwtn: data perawatan

d-fas: data fasilitas.

## Perancangan Basisdata

model basisdata dari Sistem Informasi Perawatan Pasien Rumah Sakit Ibud Anak Labuh Baru

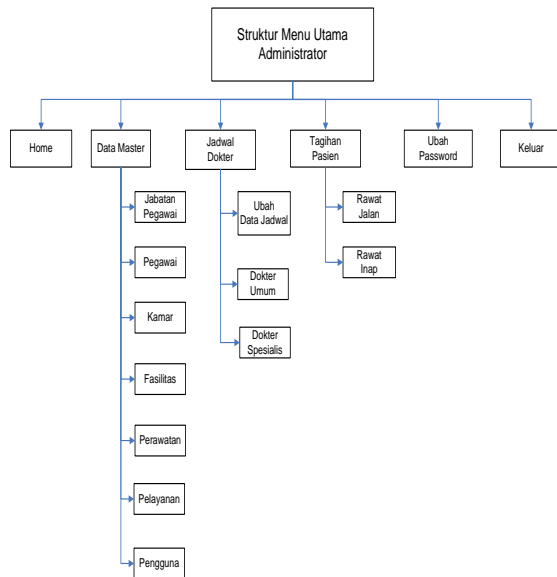


Gambar rancangan Basis Data

## Struktur Menu

Berikut ini merupakan gambar struktur menu SIPPRSIA. Sistem ini terdiri 4 level hak akses pengguna yang terdiri dari administrator, bagian registrasi, asisten dokter dan direktur utama RSIA





**Antarmuka**

Pada subbab ini akan ditampilkan antar muka sistem informasi perawatan pasien RSIA Labuh Baru



**6.1. Kesimpulan**

Dengan adanya sistem baru ini, dapat diambil sebuah kesimpulan yaitu sebagai berikut :

1. Analisa dan perancangan system informasi perawatan pasien RSIA ini bias diterapkan menggunakan konsep

metodologi berorientasi aspek dengan pemodelan menggunakan *Unified Modelling Language (UML)*.

2. Analisis dan perancangan sistem informasi perawatan pasien RSIA dengan menggunakan metodologi berorientasi aspek memerlukan waktu cukup lama hal ini disebabkan melakukan abstraksi terhadap masalah-masalah yang ada.
3. Analisis dan perancangan sistem informasi perawatan pasien RSIA ini dapat memberikan informasi dengan cepat dan akurat dalam penyajian pelayanan an registrasi pasien, seperti dapat memberikan kemudahan dalam melakukan registrasi pasien, baik itu pasien rawat jalan dan pasien rawat inap.

**DAFTAR PUSTAKA**

Adhari, "Pengantar Pemrograman Berorientasi Aspek", [Online] Available URL: <http://www.ilmukomputer.com.aop@groups.or.id>.

Ari Yanuar Ridwan, "memperbaiki modularitas program dengan pemrograman berorientasi aspek", Manajemen Informatika Politeknik Pos Indonesia, 2006.

Brown, Jeff. "Developing Aspects with Spring AOP". 2008.

Coad, Peter dan Yourdon. *Object-Oriented Analysis*. New York: Yourdon Press Prentice Hall Building, 1991.

Filman, R. "Aspect-oriented programming is quantification and obliviousness". Proceeding of OOPSLA2000 workshop on Advanced Separation of Concerns, 2000.

Kristanto, Andi. "Sistem Informasi dan Aplikasinya". Yogyakarta : Gava Media Yogyakarta, 2003.

Jacobson Ivan dan Pan-Wei Ng. "Aspect-Oriented Software Development with Use Cases", 2004.

- Jogiyanto, H.M. *“Analisa dan Desain Sistem Informasi”*. Yogyakarta : Andi Offset, 1999.
- Kristanto, Andri. *“Perancangan Sistem Informasi dan Aplikasinya”*. Jakarta: Penerbit Gava Media, 2003.
- Machacek, Jan., Vukotic, Aleksa., Chakraborty, Anirvan., & Ditt, Jessica.Pro Spring 2.5, United States of America: Appress, 2008.
- Nugroho, Adi. *“Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek”*. Bandung : Informatika, 2005.
- Suhendar, A. *“Visual Modelling menggunakan UML dan Relational Rose”*. Bandung: Penerbit Informatika Bandung, 2002.
- Sutopo, Ariesto Hadi. *“Analisis dan Desain Berorientasi Objek”*. Jakarta: Penerbit J & J Learning, 2002.

[http://iaprma.staff.gunadarma.ac.id/Downloads/files/5456/Bahasan7\\_UML\\_bagian1](http://iaprma.staff.gunadarma.ac.id/Downloads/files/5456/Bahasan7_UML_bagian1)

<http://www.digituck.com/konsep-oop-object-oriented-programming.html>.