

**PENGEMBANGAN LANJUT SISTEM INFORMASI KESEHATAN  
“ANDAL” BERDASARKAN MODEL ORGANISASI DAN  
MANAJEMEN KESEHATAN PRIMER “ANDAL” DENGAN  
PENDEKATAN *INCREMENTAL PROCESS MODEL***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Fathir Izzuddin Qisthi  
NIM: 135150400111006



PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2017

## DAFTAR ISI

|   |      |
|---|------|
| DAFTAR ISI .....  | ii   |
| DAFTAR TABEL.....   | v    |
| DAFTAR GAMBAR.....  | vi   |
| DAFTAR LAMPIRAN .....   | viii |
| BAB 1 PENDAHULUAN.....  | 9    |
| 1.1 Latar Belakang.....   | 9    |
| 1.2 Rumusan Masalah.....  | 10   |
| 1.3 Tujuan .....  | 11   |
| 1.3.1 Tujuan Umum.....  | 11   |
| 1.3.2 Tujuan Khusus .....   | 11   |
| 1.4 Manfaat.....  | 11   |
| 1.5 Batasan Masalah .....   | 11   |
| 1.6 Sistematika Pembahasan .....                                  | 11   |
| BAB 2 LANDASAN KEPUSTAKAAN .....                                  | 13   |
| 2.1 Penelitian Terdahulu .....                                    | 13   |
| 2.2 Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” ..... | 13   |
| 2.2.1 Visi dan Misi .....   | 14   |
| 2.2.2 Tujuan.....   | 15   |
| 2.2.3 Strategi .....  | 15   |
| 2.2.4 Produk Pelayanan Kesehatan Primer.....                      | 16   |
| 2.3 Sistem Informasi Kesehatan “ANDAL” .....                      | 16   |
| 2.4 Generic Software Process .....                                | 17   |
| 2.5 Incremental Process Model .....                               | 18   |
| 2.5.1 Specification .....   | 20   |
| 2.5.2 Development.....  | 21   |
| 2.5.3 Validation .....  | 22   |
| 2.6 Unified Modeling Language .....                               | 23   |
| 2.6.1 Use-Case Diagram .....                                      | 24   |
| 2.6.2 Activity Diagram .....                                      | 26   |
| 2.6.3 Sequence Diagram .....                                      | 29   |

|  |           |
|--|-----------|
| 2.6.4 Class Diagram .....  | 30        |
| <b>BAB 3 METODOLOGI .....</b>  | <b>33</b> |
| 3.1 Alur Penelitian.....   | 33        |
| 3.1.1 Studi Pustaka .....  | 34        |
| 3.1.2 Analisis Kebutuhan.....  | 34        |
| 3.1.3 Pengumpulan Data.....  | 34        |
| 3.1.4 Incremental Process.....   | 35        |
| 3.1.5 Penarikan Kesimpulan.....  | 36        |
| <b>BAB 4 ANALISIS PERSYARATAN .....</b>  | <b>37</b> |
| 4.1 Analisis Persyaratan .....   | 37        |
| 4.1.1 Analisis Tipe Stakeholder .....  | 37        |
| 4.1.2 Analisis Masalah .....   | 38        |
| 4.1.3 Analisis Kebutuhan Stakeholder .....   | 39        |
| 4.1.4 Analisis Tipe Pengguna .....   | 42        |
| 4.1.5 Analisis Fitur .....   | 42        |
| 4.1.6 Persyaratan Fungsional .....   | 44        |
| 4.2 Use-Case Model .....   | 46        |
| 4.2.1 Diagram Use-Case .....   | 46        |
| 4.2.2 Deskripsi Aktor .....  | 48        |
| 4.2.3 Use-Case Description .....   | 48        |
| 4.3 Activity Model .....   | 54        |
| 4.3.1 Activity Diagram Mengelola Data Pengobatan Holistik .....                          | 55        |
| 4.3.2 Activity Diagram Membaca Panduan Cara Menghindari dan Mengatasi Penyakit.....      | 55        |
| 4.3.3 Activity Diagram Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien ..... | 55        |
| 4.3.4 Activity Diagram Mencatat Data Obat Keluar .....                                   | 58        |
| <b>BAB 5 PERANCANGAN DAN IMPELEMANTASI .....</b>   | <b>59</b> |
| 5.1 Perancangan Sequence Diagram .....   | 59        |
| 5.1.1 Sequence Diagram Mengelola Data Pengobatan Holistik .....                          | 59        |
| 5.1.2 Sequence Diagram Membaca Panduan Cara Menghindari dan Mengatasi Penyakit.....      | 63        |

|  |    |
|--|----|
| 5.1.3 Sequence Diagram Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien ..... | 65 |
| 5.2 Perancangan Class Diagram.....   | 67 |
| 5.2.1 Perancangan Analysis Class Diagram .....   | 68 |
| 5.2.2 Perancangan Controller Class Diagram.....  | 68 |
| 5.2.3 Perancangan Model Class Diagram.....   | 68 |
| DAFTAR PUSTAKA.....  | 73 |

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2.1 Notasi pada Use Case Diagram .....  | 24 |
| Tabel 2.2 Notasi Dalam Activity Diagram.....  | 27 |
| Tabel 2.3 Notasi dalam Sequence Diagram .....   | 29 |
| Tabel 2.4 Access Level pada Atribut dan Operasi Class .....                                       | 31 |
| Tabel 2.5 Notasi Hubungan Antar Class .....   | 31 |
| Tabel 4.1 Tipe Stakeholder.....   | 37 |
| Tabel 4.2 Problem Statement .....   | 38 |
| Tabel 4.3 Kebutuhan Stakeholder.....  | 39 |
| Tabel 4.4 Tipe Pengguna .....   | 42 |
| Tabel 4.5 Fitur Sistem Informasi Kesehatan .....  | 43 |
| Tabel 4.6 Persyaratan Fungsional .....  | 45 |
| Tabel 4.7 Hubungan Fitur dengan Use-Case.....   | 47 |
| Tabel 4.8 Deskripsi Aktor .....   | 48 |
| Tabel 4.9 Use-Case Description Mengelola Data Pengobatan Holistik .....                           | 48 |
| Tabel 4.10 Use-Case Description Membaca Panduan Cara Menghindari dan Mengatasi Penyakit.....      | 50 |
| Tabel 4.11 Use-Case Description Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien ..... | 52 |
| Tabel 4.12 Use-Case Description Mencatat Data Obat Keluar .....                                   | 53 |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 2.1 Incremental Process Model .....   | 19 |
| Gambar 2.2 Proses Requirement Engineering .....  | 21 |
| Gambar 2.3 Model Umum Proses Perancangan .....   | 23 |
| Gambar 2.4 Tahap-Tahap Pengujian .....   | 23 |
| Gambar 2.5 Contoh Use Case Diagram .....   | 26 |
| Gambar 2.6 Activity Diagram .....  | 29 |
| Gambar 2.7 Gambar Class dan Area yang Dimiliki .....   | 31 |
| Gambar 3.1 Alur Penelitian .....   | 33 |
| Gambar 4.1 Keterangan Kodifikasi Kebutuhan .....   | 39 |
| Gambar 4.2 Kodifikasi Fitur .....  | 43 |
| Gambar 4.3 Kodifikasi Persyaratan Fungsional .....   | 44 |
| Gambar 4.4 Use-Case Diagram Sistem Informasi Kesehatan “ANDAL” .....   | 47 |
| Gambar 4.5 Activity Diagram Mengelola Data Pengobatan Holistik .....   | 56 |
| Gambar 4.6 Activity Diagram Membaca Panduan Cara Menghindari dan Mengatasi Penyakit .....                      | 57 |
| Gambar 4.7 Activity Diagram Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien .....                  | 57 |
| Gambar 4.8 Activity Diagram Mencatat Data Obat Keluar .....  | 58 |
| Gambar 5.1 Sequence Diagram Mengelola Data Pengobatan Holistik .....   | 60 |
| Gambar 5.2 Sequence Diagram Memuat Data Modul .....  | 61 |
| Gambar 5.3 Sequence Diagram Memuat Detail Data Pengobatan Holistik .....                                       | 61 |
| Gambar 5.4 Sequence Diagram Memproses Penyimpanan Data Pengobatan Holistik .....                               | 62 |
| Gambar 5.5 Sequence Diagram Membaca Panduan Cara Menghindari dan Mengatasi Penyakit .....                      | 63 |
| Gambar 5.6 Sequence Diagram Memilih Modul Lain .....   | 64 |
| Gambar 5.7 Sequence Diagram Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien .....                  | 65 |
| Gambar 5.8 Sequence Diagram Menampilkan Detail Data Dasar Kesehatan Keluarga .....                             | 66 |
| Gambar 5.9 Sequence Diagram Melakukan Perhitungan Ulang Tingkat Risiko Penyakit dan Tingkat Stres Pasien ..... | 67 |

|   |    |
|---|----|
| Gambar 5.10 Controller Class Diagram..... | 68 |
| Gambar 5.11 Model Class Diagram.....      | 69 |
| Gambar 5.13 Analysis Class Diagram .....  | 70 |
| Gambar 5.14 Physical Data Model.....      | 71 |

## DAFTAR LAMPIRAN

No table of figures entries found.



## **BAB 1 PENDAHULUAN**

### **1.1 Latar Belakang**

Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” merupakan model yang berisi panduan strategis dan operasional pelayanan kesehatan primer bagi Fasilitas Kesehatan Tingkat Pertama (FKTP) yang terdiri dari puskesmas, praktik dokter, praktik dokter gigi, klinik pratama termasuk fasilitas kesehatan tingkat pertama milik TNI/POLRI, rumah sakit kelas D pratama untuk melaksanakan pengelolaan dalam pengendalian mutu dan biaya pelayanan. Model manajemen tersebut dirancang agar FKTP dapat memberikan pelayanan kesehatan primer kepada sasaran pelayanan (perorangan, keluarga, komunitas peserta BPJS) dengan tujuan memenuhi 5 tujuan dasar pelayanan kesehatan primer secara optimal di era Jaminan Kesehatan Nasional (JKN).

Saat ini Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” diterapkan di Poliklinik Pabrik Gula Kebon Agung, Kecamatan Pakisaji, Kabupaten Malang. Salah satu penerapan model tersebut ialah dengan melaksanakan 10 usaha pokok pada Poliklinik Pabrik Gula Kebon Agung. Berdasarkan hasil wawancara dengan dr. Jack Roebijoso, dalam penerapan Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” dibutuhkan dukungan teknologi yang selaras dalam proses pelaksanaannya. Dibutuhkan sistem informasi kesehatan yang mampu membantu mengelola data dasar kesehatan keluarga, mengelola data registrasi dan status pasien, menampilkan tingkat resiko kesehatan dan tingkat stres, mengelola hasil diagnosis pasien, mengelola data tindakan medis yang diberikan kepada pasien, mengelola data rujukan pasien, mengelola data pengecekan darah pasien, mengelola data kegiatan edukasi, advokasi, dan pengelolaan lingkungan tempat tinggal atau tempat kerja pasien. Selain itu, sistem informasi kesehatan diharapkan mampu menampilkan persentase penyakit yang tersebar mulai dari tingkat RT, RW, kelurahan, kecamatan, hingga kota/kabupaten, sehingga dapat membantu menentukan persediaan obat, tindakan medis, edukasi, advokasi, dan pengelolaan lingkungan tempat tinggal atau tempat kerja di tiap wilayah.

Sistem Informasi Kesehatan “ANDAL” saat ini telah dikembangkan dan mampu memberikan fitur pengelolaan data dasar kesehatan keluarga untuk memenuhi kebutuhan pengelolaan data dasar kesehatan keluarga dari masyarakat, fitur registrasi pasien untuk memenuhi kebutuhan pengelolaan data registrasi, fitur menampilkan tingkat resiko kesehatan dan tingkat stres, fitur pengelolaan hasil pemeriksaan paramedis yang meliputi pemeriksaan keluhan pasien, tekanan darah, denyut nadi, suhu badan untuk memenuhi kebutuhan pengelolaan status pasien. Sistem Informasi Kesehatan “ANDAL” yang dikembangkan saat ini merupakan purwarupa dari sistem informasi kesehatan yang dibutuhkan untuk mendukung penerapan Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” di Poliklinik Pabrik Gula Kebon Agung, Kabupaten Malang.

Masalah utama dalam sistem informasi kesehatan adalah kegagalan untuk memahami ekspektasi dan persyaratan pengguna (Aggarwal, 2002). Aggarwal menambahkan, kegagalan dalam memahami kebutuhan pengguna menyebabkan kegagalan implementasi dan menyebabkan fungsi sistem informasi kesehatan tidak efisien, sehingga menurut Aggarwal, analisis harus mampu menggali persyaratan pengguna dari pernyataan kebutuhan pengguna. Salah satu metode yang dapat digunakan untuk menganalisis masalah dan menentukan persyaratan sistem informasi adalah *Object-oriented Analysis* (OOA) (Aggarwal, 2002). Aggarwal menjelaskan bahwa OOA lebih cenderung menghasilkan spesifikasi persyaratan yang lengkap karena OOA memodelkan sistem seperti objek aslinya.

Menurut Aggarwal, aktivitas yang dapat dilakukan untuk mencapai tujuan OOA di antara lain adalah memodelkan fungsi sistem dan mengidentifikasi objek-objek sistem, memodelkan objek-objek dan mengidentifikasi hubungan antar objek, serta memodelkan *behavior* objek. Setiap aktivitas yang dilakukan akan menghasilkan representasi persyaratan sistem yang berbeda-beda. Pemodelan object-oriented mengkombinasikan tiga gambaran sistem dalam bentuk pemodelan objek, *dynamic*, dan fungsional (Krol & Reich, 1999). Representasi sistem tersebut dapat digambarkan melalui Unified Modeling Language (Aggarwal, 2002). Aggarwal menambahkan bahwa UML menjamin efektivitas komunikasi antara pengguna dengan analisis, sehingga dapat memudahkan analisis mendokumentasikan persyaratan pengguna.

Setelah analisis dan pemodelan dilakukan, proses implementasi sistem dengan konsep *object-oriented* dapat dimulai dengan menggunakan hasil pemodelan objek yang berisi atribut dan *behavior* objek. Menurut Jacobson, objek-objek yang telah diidentifikasi dan dimodelkan dapat diimplementasi menjadi *class* yang memiliki variabel sebagai representasi atribut objek, serta memiliki *operation* sebagai representasi *behavior* objek. Selain itu Jacobson menambahkan bahwa implementasi harus memperhatikan konsep-konsep enkapsulasi objek, inheritas antar kelas, *class* dan *instances*, serta *polymorphism*.

Berdasarkan hasil wawancara mengenai kebutuhan dukungan sistem informasi kesehatan terhadap penerapan Model Organisasi dan Manajemen Kesehatan Primer "ANDAL" di Poliklinik Pabrik Gula Kebon Agung, penelitian ini bertujuan untuk menghasilkan purwarupa pengembangan lanjut Sistem Informasi Kesehatan "ANDAL" dengan menyajikan hasil dokumentasi analisis persyaratan, perancangan, pengembangan lanjut, dan pengujian purwarupa Sistem Informasi Kesehatan "ANDAL" dengan menggunakan pendekatan *object-oriented*.

## 1.2 Rumusan Masalah

1. Dapatkan pengembangan lanjut Sistem Informasi Kesehatan Primer "ANDAL" dilakukan dengan menggunakan pendekatan berorientasi objek?

### **1.3 Tujuan**

#### **1.3.1 Tujuan Umum**

Melakukan pengembangan lanjut Sistem Informasi Kesehatan Primer “ANDAL” dengan menggunakan pendekatan berorientasi objek.

#### **1.3.2 Tujuan Khusus**

1. Mengidentifikasi persyaratan sistem dan merancang Sistem Informasi Kesehatan Primer “ANDAL” berdasarkan hasil perancangan sistem terdahulu dengan menggunakan pendekatan berorientasi objek.
2. Membangun Sistem Informasi Kesehatan Primer “ANDAL” berdasarkan hasil identifikasi persyaratan sistem dan perancangan sistem menggunakan pendekatan berorientasi objek.
3. Mengetahui hasil pengujian Sistem Informasi Kesehatan Primer “ANDAL”.

### **1.4 Manfaat**

Manfaat dari penelitian ini adalah terwujudnya purwarupa pengembangan lanjut sistem informasi kesehatan yang dapat menunjang penerapan Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” yang sesuai dengan kebutuhan bisnis pada Model Organisasi dan Manajemen Kesehatan Primer “ANDAL”. Hasil penelitian dapat digunakan sebagai usulan pengembangan sistem informasi kesehatan di masa yang akan datang, selain itu hasil penelitian dapat dijadikan sumber atau acuan pengembangan sistem informasi kesehatan yang lebih baik.

### **1.5 Batasan Masalah**

Pengembangan lanjut Sistem Informasi Kesehatan “ANDAL” dilakukan berdasarkan proses bisnis yang ada pada Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” yang diterapkan pada Poliklinik Pabrik Gula Kebon Agung, Kabupaten Malang, dengan langkah awal proses bisnis yang dimulai dari proses pengumpulan data dasar kesehatan keluarga, pengisian data registrasi dan status pasien, pelaksanaan diagnosa holistik kepada pasien. Hasil pengembangan lanjut Sistem Informasi Kesehatan “ANDAL” tidak untuk memenuhi kebutuhan dukungan sistem informasi terhadap proses pengelolaan data kegiatan edukasi, advokasi, dan pengelolaan lingkungan tempat tinggal atau tempat kerja pasien.

### **1.6 Sistematika Pembahasan**

Sistematika penulisan penelitian ini antara lain sebagai berikut:

#### **BAB I Pendahuluan**

Bab ini berisi latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika pembahasan skripsi.

#### **Bab II Landasan Kepustakaan**

Bab ini berisi tinjauan pustaka dari beberapa penelitian sebelumnya, serta berisi pembahasan tentang teori, model, metode, atau sistem berdasarkan literatur ilmiah yang dapat menunjang penulisan penelitian.

### **BAB III Metodologi**

Dalam bab ini menjelaskan langkah-langkah yang dilakukan untuk menyelesaikan penelitian.

### **BAB IV Analisis Kebutuhan**

Bab ini berisi pembahasan kebutuhan sistem yang sesuai dengan hasil identifikasi proses bisnis pada Model Organisasi dan Manajemen Kesehatan Primer “ANDAL”.

### **BAB V Perancangan dan Implementasi**

Bab ini menjelaskan hasil perancangan dengan menyertakan diagram-diagram yang dibutuhkan sebagai syarat perancangan. Selain itu, bab ini membahas detail implementasi dari fungsi-fungsi yang menjadi fokus utama.

### **BAB VI Pengujian**

Bab ini berisi proses dan hasil pengujian SIKES “ANDAL” dengan menggunakan metode pengujian *functional test* untuk mengetahui apakah fungsi-fungsi sudah berjalan dengan baik dan *user acceptance test* untuk mengetahui tingkat kesesuaian hasil pengembangan dengan kebutuhan bisnis dan sistem.

### **BAB VII Penutup**

Bab ini berisi tentang kesimpulan dan saran berdasarkan hasil pengujian yang dilakukan terhadap hasil pengembangan.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Penelitian Terdahulu

Penelitian ini dilakukan berdasarkan penelitian terdahulu yang membahas penggunaan pendekatan *Object-Oriented Analysis* (OOA), *Object-Oriented Design* (OOD), dan *Object-Oriented Programming* (OOP) dalam analisis, perancangan, dan implementasi perangkat lunak. Penelitian terdahulu juga menjadi landasan peneliti untuk menggunakan *Use-Case Modeling* sebagai bagian dari aktivitas OOA dan OOD untuk menganalisis kebutuhan dan persyaratan, serta kemudian memodelkannya dengan memanfaatkan kombinasi dari beberapa *Unified Modeling Language* (UML). Hasil pemodelan perancangan tersebut kemudian akan menjadi *input* untuk melakukan implementasi sistem informasi kesehatan yang akan dibangun dalam penelitian ini.

Vinod Aggarwal dalam penelitian yang berjudul "*The Application of the Unified Modeling Language in Object-Oriented Analysis of Healthcare Information Systems*" menjelaskan bahwa *Object-Oriented Analysis* merupakan teknik yang sesuai untuk mengembangkan sistem informasi kesehatan. Aktivitas OOA yang dilakukan dalam penelitian tersebut menghasilkan representasi persyaratan sistem informasi dari berbagai sudut pandang, kemudian representasi persyaratan tersebut dapat disajikan dengan Unified Modeling Language (UML). Penelitian tersebut juga menjelaskan bahwa UML memiliki manfaat dalam menentukan, memvisualisasi, membangun, mendokumentasikan, dan mengkomunikasikan hasil dari aktivitas OOA.

Marina Krol dan David L. Reich dalam penelitian yang berjudul "*Object-Oriented Analysis and Design of a Health Care Management Information System*" menjelaskan bahwa *Object-Oriented Analysis and Design* (OAAAD) memiliki keuntungan dalam pengembangan sistem komputer yang kompleks. OOAD juga terbukti diterima oleh komunitas informatika medis untuk mengembangkan sistem kesehatan, salah satunya informasi rumah sakit. Penelitian ini memiliki fokus untuk memodelkan objek, memodelkan urutan operasi masing-masing objek didalam sistem, dan memodelkan fungsi sistem dengan pendekatan OOAD.

Odeh, et al., dalam penelitiannya berjudul "*A Use-Case Driven Approach in Requirements Engineering: The MAMMOGRID Project*" menunjukkan penerapan teknik *Use-Case Modeling* untuk mengidentifikasi dan menentukan persyaratan pengguna dari proyek MAMMOGRID dalam pendekatan *incremental* dan *iterative*. Penelitian tersebut menunjukkan bahwa teknik *Use-Case Modeling* memperpendek jarak diantara setiap fase dalam proses Requirement Engineering, sehingga dapat mengurangi siklus proses *Requirement Engineering*.

### 2.2 Model Organisasi dan Manajemen Kesehatan Primer "ANDAL"

Model Organisasi dan Manajemen Kesehatan Primer "ANDAL" diusulkan berdasarkan pelayanan kesehatan primer di Indonesia masih berorientasi kuratif sehingga dapat menyebabkan pengeluaran biaya yang tinggi dalam pengobatan.

Persentase pengeluaran nasional sektor kesehatan tahun 2005 adalah sebesar 0,81% dan meningkat menjadi 1,09% pada tahun 2007 (Departemen Kesehatan RI, 2009). Begitu juga dengan anggaran kesehatan, jumlah APBN kesehatan tahun 2004 sebesar Rp 5,54 triliun, kemudian meningkat menjadi Rp 18,75 Triliun pada tahun 2007 (Departemen Kesehatan RI, 2009). Pengeluaran pemerintah untuk kesehatan terus meningkat, namun kontribusi pengeluaran pemerintah untuk kesehatan kecil, yaitu 38% dari total pengeluaran kesehatan (Departemen Kesehatan RI, 2009). Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” pernah diterapkan pada tahun 1998. Selama model manajemen tersebut diterapkan, profil rasio orang berobat dari karyawan menurun hingga 46% dan profil angka kejadian kasus rawat inap di RS menurun hingga 33% (Roebijoso, 2010).

Roebijoso (2010) menjelaskan bahwa Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” dapat memberikan panduan untuk menyusun organisasi pelayanan kesehatan primer dokter keluarga yang fungsional, sehingga dapat menjalankan upaya kesehatan primer bagi perorangan (UKP) yang berintegrasi optimal dengan upaya kesehatan bagi masyarakat (UKM). Selain itu, model manajemen tersebut dapat menjadi panduan untuk menyusun kompetensi, materi dan/atau modul, metode, dan evaluasi pendidikan dan/atau tenaga kesehatan klinik dokter keluarga. Model manajemen tersebut akan menyerap tenaga kerja pelayanan kesehatan primer sehingga secara fungsional, tugas pelayanan kesehatan primer dapat dilaksanakan secara optimal.

Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” merupakan model dengan manajemen yang integratif, strategis, efektif, efisien, berkualitas, merata, dan terjangkau yang disusun berdasarkan motivasi untuk memberikan kepuasan kepada pasien dan untuk memberikan layanan promotif, preventif, dan kuratif dengan mengusahakan kegiatan pelayanan promotif dan preventif lebih banyak dibandingkan pemberian pelayanan kuratif. Motivasi tersebut merupakan pendorong stakeholder dalam memberikan pelayanan kesehatan primer sehingga dinilai dapat menurunkan jumlah rawat inap dan rawat jalan pasien karena layanan pencegahan penyakit lebih banyak diberikan daripada layanan kuratif, sehingga penyakit dapat terobati dan dapat dicegah, yang menyebabkan angka pasien rawat inap dan rawat jalan semakin berkurang. Kepuasan pasien dapat mendorong pasien untuk terdaftar pada puskesmas, klinik, atau pada dokter praktek perorangan sehingga dapat meningkatkan kapitasi dan pendapatan puskesmas, klinik, atau pada dokter praktek perorangan (Nugraha, 2017).

### **2.2.1 Visi dan Misi**

Visi Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” adalah “Menjadi sentuhan pertama untuk dapat hidup sehat”.

Misi Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” adalah “Memberdayakan hidup sehat dan memberikan pelayanan kesehatan dasar pada sasaran”.

### 2.2.2 Tujuan

Menurut Roebijoso(2010) tujuan Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” adalah sebagai berikut:

1. Meningkatkan pemberdayaan hidup sehat dan derajat kesehatan (rasio usaha pokok pencegahan dengan pengobatan).
2. Menurunkan angka kesakitan, kematian, kecacatan, dan kecelakaan.
3. Menurunkan angka rujukan medis dan rawat inap di Rumah Sakit.
4. Menurunkan total pembiayaan kesehatan dan mengoptimalkan biaya kesehatan primer.
5. Meningkatkan kualitas, pemerataan, keterjangkauan pelayanan kesehatan primer serta dapat dievaluasi tiap tribulan, semester, tahun, sampai 5 tahun untuk menilai keberhasilan strategi.

### 2.2.3 Strategi

Roebijoso (2010) menjelaskan penyusunan alternatif strategi pada Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” dilakukan dengan memperhatikan hasil kajian model strategi pelayanan dokter keluarga dengan bukti penurunan jumlah kunjungan pasien berobat (*contact rate*) yang dapat digunakan sebagai indikator angka kesakitan, angkar rujukan medis dan rawat inap, serta total biaya kesehatan yang menurun 54% dalam satu tahun. Alternatif strategi pada Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” adalah sebagai berikut:

1. Menyusun atau modifikasi organisasi UKP yang fungsional dan mampu menjalankan usaha-usaha pokok pelayanan keseshatan dokter keluarga yang dapat diadopsi dari sebuah model manajemen strategis pelayanan kesehatan dokter keluarga (Roebijoso, 2010), yaitu berupa sepuluh usaha-usaha pokok pelayanan kesehatan dokter keluarga, yang terdiri dari 60% upaya promotif preventif (pemberdayaan kesehatan) dan 40% upaya kuratif dasar.
2. Menyusun *standart input*, proses, cakupan *output*, evaluasi *outcome*, dan *impact* dalam manajemen pelayanan kesehatan primer yang integratif, strategis, efektif, efisien, berkualitas, merata, terjangkau yang mengutamakan pencegahan penyakit. Standarisasi *input* meliputi fasilitas personalia terlatih, perangkat lunak, perangkat keras teknologi informasi kesehatan, pembiayaan, indikator keberhasilan, materi modul dan sub modul pemberdayaan, dan pelayanan kesehatan medis dasar yang ditujukan bagi perorangan, keluarga, komunitas sasaran yang sesuai dengan kondisi lokal. Standar proses meliputi upaya menentukan prioritas sasaran pelayanan dan *Standart Operating Procedure* (SOP) dari intervensi pelayanan kesehatan primer (medis, non medis) dalam sepuluh usaha-usaha pokok pelayanan kesehatan dokter keluarga dengan modul/sub modul yang baku sehingga kualitas dan konsistensi pelayanan kesehatan primer dapat selalu dijamin karena setiap anggota tim kesehatan akan menjalankan upaya yang dibakukan

di suatu wilayah kerja. Standar evaluasi keberhasilan meliputi alat ukur, waktu, cara mengumpulkan, menyusun dan menampilkan data indikator keberhasilan, melakukan analisa data pelayanan kesehatan, dan rencana intervensi pelayanan kesehatan pada sasaran yang dibakukan agar dapat mencapai tujuan.

#### **2.2.4 Produk Pelayanan Kesehatan Primer**

Beberapa usaha pokok pada Upaya Pelayanan Kesehatan bagi Perorangan (UKP) ditambahkan dan terbukti dapat mengurangi angka sakit dan dapat menurunkan biaya pengobatan (Roebijoso, 2010). Usaha pokok pada UKP diantaranya:

1. Diagnosa holistik/komprehensif di klinik.
2. Edukasi dan advokasi kesehatan perorangan di klinik (menggunakan modul).
3. Edukasi dan advokasi kesehatan keluarga di klinik.
4. Tindakan medis (*emergency, diagnostic*, kuratif, rehabilitasi dasar) di klinik.
5. Diagnosa holistik/komprehensif di rumah.
6. Edukasi dan advokasi kesehatan keluarga di rumah.
7. Tindakan medis (*emergency, diagnostic*, kuratif, rehabilitasi dasar) di rumah.
8. Mediasi dan advokasi kesehatan di rumah sakit rujukan.
9. Kunjungan tempat kerja (Identifikasi Resiko K3).
10. Edukasi dan advokasi kesehatan kelompok (komunitas) sesuai dengan masalah kesehatan yang ada.

#### **2.3 Sistem Informasi Kesehatan “ANDAL”**

Sistem Informasi Kesehatan “ANDAL” dikembangkan untuk memenuhi kebutuhan dukungan teknologi informasi bagi Model Organisasi dan Manajemen Kesehatan Primer “ANDAL”. Berdasarkan hasil wawancara dengan dr. Jack Roebijoso, Sistem Informasi Kesehatan “ANDAL” pada dasarnya digunakan untuk mengelola data dasar kesehatan keluarga, pengolahan data tersebut digunakan untuk membantu proses pengambilan keputusan tindakan medis bersifat promotif, preventif, dan kuratif, serta digunakan untuk mengevaluasi keberhasilan.

Data dasar kesehatan keluarga berisi identitas keluarga, riwayat penyakit atau kecelakaan yang pernah dialami anggota keluarga, dan data rekam medis pasien, data tersebut akan diperbarui setiap 3 bulan sekali. Data dasar kesehatan keluarga direkam melalui Sistem Informasi Kesehatan “ANDAL” oleh petugas administrasi, petugas rekam medis klinik, tenaga paramedis, atau tenaga medis. Pengolahan data dasar kesehatan keluarga akan menghasilkan tingkat resiko kesehatan dan tingkat stres pada masing-masing anggota keluarga. Sistem Informasi Kesehatan “ANDAL” juga digunakan untuk mengelola data registrasi dan status pasien ketika



pasien ingin mendapatkan pelayanan kesehatan di klinik. Data status pasien berisi hasil pemeriksaan paramedis yang meliputi pemeriksaan keluhan pasien, tekanan darah, denyut nadi, suhu badan, dan lain-lain. Berdasarkan data-data tersebut, tenaga medis dapat mendiagnosis pasien dan merekam hasil diagnosis melalui Sistem Informasi Kesehatan "ANDAL".

Tenaga medis dapat memberikan tindakan medis berdasarkan modul penyakit yang dapat diakses melalui Sistem Informasi Kesehatan "ANDAL" sesuai dengan hasil diagnosis yang telah direkam oleh tenaga medis. Selain itu, pemberian tindakan medis berdasarkan modul faktor pemicu dan faktor risiko penyakit dapat diberikan kepada pasien oleh tenaga medis setelah data faktor pemicu dan faktor risiko penyakit direkam melalui Sistem Informasi Kesehatan "ANDAL". Tenaga medis juga dapat mengelola kebutuhan rujukan (rujukan ke rumah sakit atau rujukan tindakan UGD) pasien, mengelola kebutuhan cek darah pasien, serta mengelola kebutuhan resep obat pasien.

Berdasarkan hasil pengolahan data dasar kesehatan keluarga, tenaga medis dapat menentukan kebutuhan kunjungan tempat tinggal dan tempat kerja pasien. Sistem Informasi Kesehatan "ANDAL" dapat membantu tenaga medis dan petugas kesehatan masyarakat untuk mengelola dokumen permintaan pengelolaan lingkungan tempat tinggal atau tempat kerja. Seluruh data hasil kegiatan edukasi kesehatan, advokasi kesehatan, pengelolaan lingkungan tempat tinggal atau tempat kerja akan direkam melalui Sistem Informasi Kesehatan "ANDAL", seluruh data yang telah direkam melalui Sistem Informasi Kesehatan "ANDAL" digunakan untuk membantu pengambilan keputusan atau kebijakan dalam pelayanan kesehatan primer dan evaluasi keberhasilan dari keputusan atau kebijakan pelayanan kesehatan primer yang diambil.

## **2.4 Generic Software Process**

*Generic software process* merupakan kumpulan aktivitas, aksi, dan tugas yang dilakukan ketika akan mengembangkan perangkat lunak. *Generic software process* tidak bersifat *rigid*, melainkan sebuah pendekatan yang dapat diadaptasi sehingga aktivitas, aksi, atau tugas yang akan dikerjakan pada sebuah pengembangan perangkat lunak dapat dipilih sesuai dengan kebutuhan. *Generic software process* memiliki tujuan agar perangkat lunak dapat di-*deliver* tepat waktu dan memiliki kualitas tinggi, sehingga kebutuhan calon pengguna dapat dengan baik. *Generic software process* meliputi lima aktivitas, yaitu:

1. *Communication*. Aktivitas ini bertujuan untuk memahami tujuan *stakeholder* pada sebuah proyek perangkat lunak dan untuk mengumpulkan kebutuhan yang dapat membantu pendefinisian fitur dan fungsi perangkat lunak.
2. *Planning*. Dalam aktivitas ini, pengembang mendeskripsikan tugas-tugas teknis yang akan dilakukan, resiko yang akan dihadapi dalam proses pengembangan perangkat lunak, sumber daya yang akan digunakan, penjelasan perangkat lunak yang akan dibuat, dan jadwal kegiatan selama pengembangan perangkat lunak berlangsung.

3. *Modeling*. Aktivitas ini bertujuan untuk lebih memahami kebutuhan perangkat lunak dan rancangan untuk dapat memenuhi kebutuhan tersebut.
4. *Construction*. Aktivitas didalamnya meliputi pembuatan kode perangkat lunak dan pengujian untuk mendeteksi kesalahan pada kode yang telah dibuat.
5. *Deployment*. Aktivitas ini bertujuan untuk mendapatkan tanggapan dan evaluasi dari calon pengguna dengan cara memberikan perangkat lunak yang telah dikerjakan kepada calon pengguna.

Dalam perkembangannya, *generic software process* diadaptasi menjadi berbagai macam model proses. Setiap model proses akan memiliki detail tugas-tugas yang berbeda, namun kerangka kerjanya secara umum tetap sama.

## 2.5 Incremental Process Model

Model *incremental* merupakan model yang akan memberikan beberapa seri rilis sebuah sistem yang secara progresif akan menyediakan fungsi-fungsi kepada calon pengguna. *Incremental Process Model* muncul berdasarkan ide untuk mengimplimentasi sistem awal, kemudian hasil implementasi tersebut dapat diserahkan kepada calon pengguna untuk mendapatkan penilaian apakah sistem awal tersebut sudah sesuai dengan sebagian kebutuhan pengguna, hasil dari penilaian tersebut dijadikan kebutuhan untuk mengimplementasi sistem secara progresif hingga sistem yang diserahkan kepada user dapat memenuhi seluruh kebutuhan yang sudah ditentukan sebelumnya.

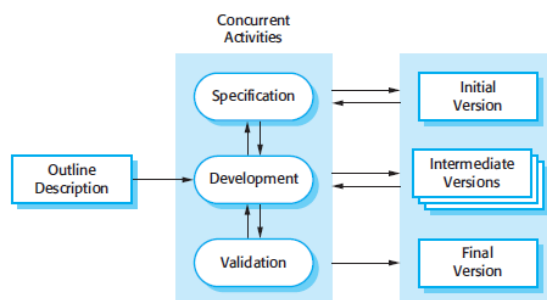
*Incremental Process Model* dapat dimanfaatkan pada kondisi ketika pengembang harus memberikan beberapa fungsi sistem yang memiliki prioritas dan urgensi tertinggi kepada calon pengguna dengan cepat, kemudian tahap pengembangan (*expansion*) fungsi-fungsi didalam sistem dapat diberikan secara progresif pada perilsan sistem di waktu yang akan datang. Sommerville (2011) menjelaskan bahwa *Incremental Process Model* memiliki beberapa keuntungan dibandingkan dengan model *Waterfall* yaitu:

1. Biaya untuk mengakomodasi perubahan kebutuhan calon pengguna dapat dikurangi. Biaya untuk melakukan analisa dan dokumentasi ulang secara signifikan dapat dikurangi dibandingkan pada model *Waterfall*.
2. Lebih mudah untuk mendapatkan tanggapan dan penilaian terhadap sistem dari calon pengguna. Calon pengguna dapat memberikan tanggapan dan melihat bagaimana hasil implementasi sistem pada saat demo sistem atau perangkat lunak.
3. Memungkinkan *delivery* dan *deployment* sistem kepada calon pengguna lebih cepat, meskipun belum semua fungsi dapat diimplementasikan kedalam sistem. Calon pengguna dapat dengan segera menggunakan dan mendapatkan nilai dari sistem yang di-*deliver* dibandingkan jika menerapkan model *Waterfall*.

Pada *Incremental Process Model*, *increment* pertama biasanya merupakan produk inti yang dibuat untuk memenuhi kebutuhan inti, namun fitur-fitur lain belum

diberikan kepada calon pengguna. Produk inti tersebut digunakan oleh pengguna untuk kemudian diberikan tanggapan sebagai evaluasi. Hasil evaluasi tersebut digunakan sebagai *input* untuk penyempurnaan fitur pada produk inti sehingga dapat memenuhi kebutuhan calon pengguna dengan baik. Selain itu, pada setiap *increment* fitur-fitur lain atau tambahan juga diimplementasi sehingga produk dapat diberikan kepada calon pengguna secara menyeluruh (Pressman, 2010).

*Incremental Process Model* dapat dilakukan dengan mengadopsi pendekatan *agile*, *plan-driven*, atau kombinasi dari keduanya. Dengan mengadopsi pendekatan *agile* kebutuhan awal sistem sudah teridentifikasi, namun pengembangan sistem (fitur-fitur yang akan diberikan) pada *increment* selanjutnya akan bergantung pada kondisi lingkungan sistem atau bergantung pada perkembangan kebutuhan calon pengguna di masa yang akan datang. Jika mengadopsi pendekatan *plan-driven* seluruh kebutuhan pengguna dan kebutuhan sistem sudah diidentifikasi terlebih dahulu. Kemudian seluruh kebutuhan tersebut dapat dimodelkan sebagai panduan bagi tim pengembang untuk melakukan pembuatan perangkat lunak. Pembuatan perangkat lunak dilakukan dengan mengambil sebagian kebutuhan berdasarkan prioritasnya, kemudian hasil pengembangan tersebut dapat di-*deliver* kepada calon pengguna untuk dievaluasi dan diberi tanggapan. Hasil evaluasi dan tanggapan tersebut menjadi *input* bagi pengembangan pada *increment* selanjutnya hingga perangkat lunak dapat memberikan seluruh fitur yang dapat mencakup seluruh kebutuhan calon pengguna. Dalam Gambar 2.1 menunjukkan kerangka kerja aktivitas pengembangan perangkat lunak pada *Incremental Process Model* terdiri dari *specification*, *development*, dan *validation*. *Specification*, *development*, dan *validation* tidak diselesaikan dengan saling terpisah, melainkan bersisipan dengan umpan balik (*feedback*) yang cepat diantara aktivitas tersebut (Sommerville, 2011).



**Gambar 2.1 Incremental Process Model**

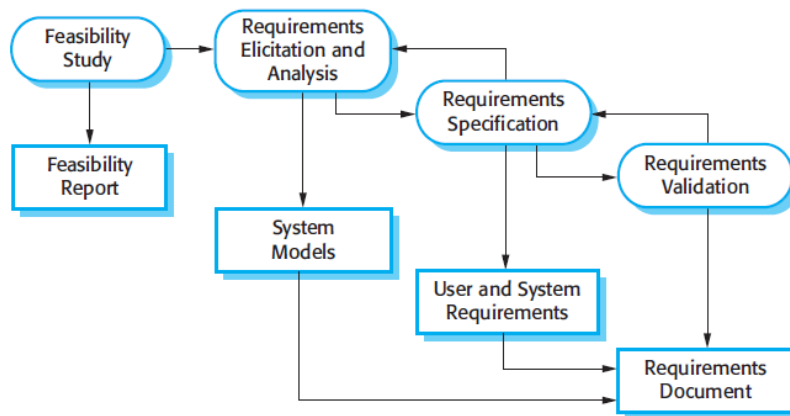
Sumber: Sommerville (2011)

### 2.5.1 Specification

Sommerville (2011) menjelaskan aktivitas *specification* atau yang dikenal dengan *Requirements Engineering* (RE) merupakan sebuah proses untuk memahami dan mendefinisikan layanan apa yang dibutuhkan dari sebuah sistem dan untuk mengidentifikasi batasan operasi pada sebuah sistem dan pengembangannya. Proses RE memiliki tujuan untuk menghasilkan dokumen kebutuhan yang disetujui yang berisi spesifikasi sistem yang dapat memenuhi kebutuhan *stakeholders*. Sesuai dengan Gambar 2.2 terdapat empat aktivitas utama dalam proses RE, yaitu:

1. *Feasibility study*. Aktivitas ini dilakukan untuk memperhitungkan sistem yang akan dibangun tidak akan memakan biaya yang tinggi dari sudut pandang bisnis dan untuk mempertimbangkan apakah sebuah sistem dapat dibangun dalam batas anggaran dana yang sudah ditentukan. Hasil dari *feasibility study* dapat mempengaruhi keputusan apakah pembuatan sebuah sistem akan diteruskan atau tidak.
2. *Requirements elicitation and analysis*. Aktivitas ini dilakukan untuk memperoleh kebutuhan sistem melalui observasi sistem yang sudah ada, diskusi dengan calon pengguna, analisis, dll. Aktivitas ini juga dapat melibatkan purwarupa sistem untuk membantu pengembang memahami sistem yang akan dibangun.
3. *Requirements specification*. Aktivitas ini dilakukan untuk mengubah informasi yang telah dikumpulkan melalui aktivitas analisa kedalam dokumen untuk mendefinisikan sekumpulan kebutuhan. Dua jenis kebutuhan biasanya akan termasuk dalam dokumen (*user requirements* dan *system requirements*).
4. *Requirements validation*. Aktivitas ini dilakukan untuk memeriksa realisme, konsistensi, dan kelengkapan kebutuhan. Pada aktivitas ini, kesalahan-kesalahan pada kebutuhan akan ditemukan, sehingga kebutuhan dapat dimodifikasi untuk memperbaiki kesalahan-kesalahan tersebut.

Ketika empat aktivitas diatas dilakukan, kebutuhan-kebutuhan dapat berubah dan bertambah, karena itu, analisa, pendefinisian, dan spesifikasi dilakukan saling bersisipan.



**Gambar 2.2 Proses Requirement Engineering**

Sumber: Sommerville (2011)

### 2.5.2 Development

Pada pendekatan *Incremental Process Model*, tahap *development* berisi proses perancangan dan pembuatan sistem berdasarkan spesifikasi sistem yang telah ditentukan. Selain itu terdapat proses penyempurnaan atau perbaikan pada spesifikasi software. Perancangan sistem adalah deskripsi dari struktur sistem yang akan dibuat, model data, dan struktur-struktur lain yang akan digunakan oleh sistem, antarmuka diantara komponen sistem, dan terkadang algoritma yang akan digunakan. Kegiatan yang dilakukan pada proses perancangan bisa saja berbeda-beda, bergantung pada sistem yang akan dibuat, berikut merupakan empat aktivitas perancangan yang ada pada proses perancangan sistem informasi:

1. Perancangan arsitektur (*architectural design*), perancangan untuk mengidentifikasi struktur sistem secara umum, sub sistem atau modul-modul, hubungan antar modul, dan bagaimana pendistribusiannya.
2. Perancangan antarmuka (*interfaces design*), perancangan untuk mendefinisikan antarmuka bagi komponen-komponen sistem. Spesifikasi antarmuka harus tidak ambigu, karena jika spesifikasi antarmuka telah didefinisikan dengan baik, komponen-komponen sistem dapat dirancang dan dikembangkan dalam waktu yang hampir bersamaan (*concurrent*).
3. Perancangan komponen (*component design*), perancangan untuk menentukan bagaimana komponen didalam sistem akan beroperasi. Spesifikasi komponen bisa saja berbentuk sebuah kalimat sederhana yang menjelaskan tentang *expected requirement* yang akan diimplementasikan, sedangkan rancangan komponen yang lebih spesifik dapat diserahkan kepada *programmer*. Model perancangan biasanya dimanfaatkan untuk melakukan implementasi secara otomatis.

4. Preancangan basis data (*database design*), perancangan untuk menentukan struktur data dan untuk merepresentasikannya dalam sebuah basis data.

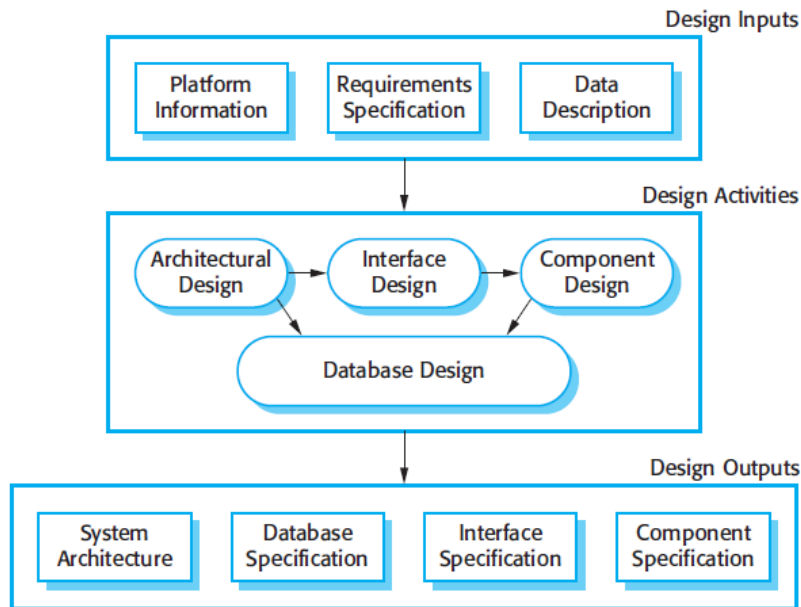
Setiap aktivitas diatas akan menghasilkan *output* perancangan. Jika pendekatan *model-driven* digunakan, *output* dari aktivitas diatas adalah diagram-diagram.

Pembuatan sebuah sistem dilakukan setelah proses perancangan sistem selesai. Kegiatan pemrograman merupakan kegiatan *personal*, tidak ada proses yang secara umum harus diikuti. Beberapa *programmer* biasanya akan memulai pemrograman pada komponen sistem yang diketahui, ataupun sebaliknya. Selain itu terdapat proses *debugging*, yaitu proses pengujian kode program untuk menemukan kecacatan atau kesalahan yang harus dihilangkan atau diperbaiki pada sistem. Pengujian kesalahan (*defect*) dengan debugging merupakan hal yang berbeda. Pengujian merupakan proses untuk menetapkan adanya kesalahan (*defect*), sedangkan *debugging* fokus pada pencarian dan perbaikan pada kesalahan. Gambar 2.3 menunjukkan proses-proses pada tahap perancangan.

### 2.5.3 Validation

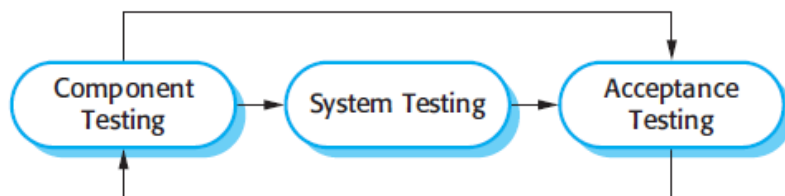
Validasi preangkat lunak atau biasa disebut *verification and validation* ditujukan untuk menunjukkan sebuah sistem sesuai dengan spesifikasi yang telah ditentukan dan sistem tersebut telah memenuhi ekspektasi konsumen atau *stakeholders* (Sommerville, 2011). Validasi melibatkan proses pengecekan, seperti inspeksi dan pengulasan pada setiap tahap SDLC. Gambar 2.4 menunjukkan 3 tahap proses pengujian yang terdiri dari pengujian komponen, pengujian sistem, dan *acceptance testing*. Tahap-tahap pada proses pengujian diantaranya adalah:

1. *Development testing*. Setiap komponen diuji secara independen. Komponen sistem bisa saja berupa fungsi, kelas-kelas, atau sekumpulan entitas didalam sistem yang saling berhubungan.
2. *System testing*. Pengujian ini ditujukan untuk menemukan kesalahan (*errors*) yang dihasilkan dari interaksi antar komponen sistem dan antarmuka komponen yang bermasalah. Selain itu, pengujian ini juga ditujukan untuk menunjukkan bahwa sistem sudah memenuhi kebutuhan fungsional dan non-fungsional, serta untuk menguji sigat sistem.
3. *Acceptance testing*. Dalam proses pengujian ini, sistem diuji dengan diberi data yang telah disiapkan oleh konsumen/*stakeholders*. Pengujian ini bisa jadi dapat memunculkan kesalahan dan kelalaian dalam pendefinisian kebutuhan sistem. *Acceptance testing* dapat juga memunculkan masalah kebutuhan yang disebabkan fasilitas sistem tidak sesuai dengan kebutuhan pengguna atau performa sistem tidak dapat diterima.



**Gambar 2.3 Model Umum Proses Perancangan**

Sumber: Sommerville (2011)



**Gambar 2.4 Tahap-Tahap Pengujian**

Sumber: Sommerville (2011)

## 2.6 Unified Modeling Language

*Unified Modeling Language* (UML) merupakan bahasa pemodelan visual yang digunakan untuk memvisualisasikan (*visualize*), menentukan (*specifying*), menyusun (*constructing*), dan mendokumentasikan (*documenting*) sistem perangkat lunak. UML merupakan bahasa grafis yang digunakan untuk memfasilitasi komunikasi, sehingga dapat meminimalisir kesulitan atau kesalahan

dalam memahami suatu konsep perangkat lunak (dalam kasus tertentu memahami konsep perangkat lunak yang kompleks). UML terdiri dari notasi yang yang dapat digunakan untuk membangun model yang tepat, tidak ambigu, dan lengkap. Pada dasarnya, UML dimaksudkan untuk digunakan pada pemodelan sistem yang melibatkan perangkat lunak pada berbagai lingkup sistem, salah satunya *medical electronic*. Namun, penggunaan UML tidak terbatas pada pemodelan sistem perangkat lunak, melainkan dapat digunakan juga untuk memodelkan sistem non-perangkat lunak.

**Commented [F1]:** (Booch, et al., 1998). Pada bagian [An Overview of the UML](#)


**Commented [F2]:** (Booch, et al., 1998) Pada bagian [Where Can the UML Be Used?](#)

### 2.6.1 Use-Case Diagram






*Use case diagram* merupakan diagram yang menggambarkan sekumpulan *use case* dan aktor, serta hubungan antara keduanya. *Use case diagram* menunjukkan *use case view* statis sebuah sistem. Diagram ini penting dalam mengorganisasi dan memodelkan sifat (*behavior*) sebuah sistem (Booch, et al., 1998). *Use case diagram* pada umumnya terdiri dari *use case*, aktor, dan notasi hubungan seperti dependensi, generalisasi, dan asosiasi. *Use case* merupakan bagian fungsional, yang menjelaskan interaksi dalam bentuk pesan yang berurutan antara sistem dan satu atau lebih aktor. Tujuan dari *use case* adalah untuk mendefinisikan sifat sistem tanpa harus menunjukkan struktur internalnya. Aktor dalam *use case diagram* meliputi manusia, serta sistem dan proses komputer. Aktor merupakan peran yang dimainkan oleh manusia, proses, atau sesuatu di luar sistem yang berinteraksi dengan sistem, subsistem, atau kelas.

*Use case diagram* dapat dimanfaatkan untuk memodelkan gambaran statis *use case* dari sebuah sistem. Penggambaran tersebut dapat menunjukkan sifat sebuah sistem, dengan kata lain layanan yang diberikan oleh sistem sesuai konteks di lingkungannya. Dalam Tabel 2.1 menunjukkan notasi pada *use case diagram*. Dalam Gambar 2.5 merupakan contoh diagram *use case* beserta dengan keterangan yang menunjukkan bagian-bagian dari diagram tersebut. Subjek diagram dibawah ini adalah *Telephone Catalog*, memiliki empat *use case* diantaranya *check status*, *place order*, *fill orders*, *establish credit*. Masing-masing aktor diantara lain *Customer* memiliki hubungan asosiasi dengan *use case check status*, *place order*, dan *establish credit*. Aktor *Salesperson* memiliki hubungan asosiasi dengan *use case check status*, *place order*. Aktor *Shipping Clerk* memiliki hubungan asosiasi dengan *use case fill orders*. Aktor *Supervisor* memiliki hubungan asosiasi dengan *use case establish credit*.

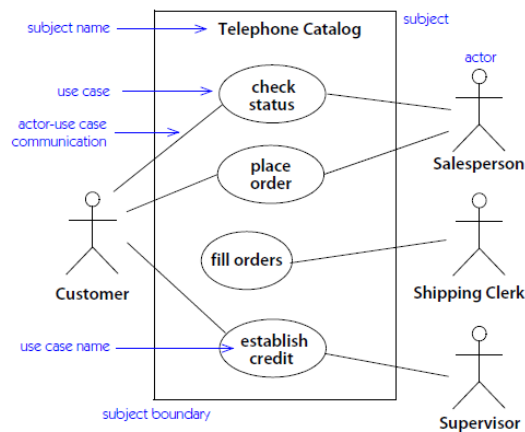
**Tabel 2.1 Notasi pada Use Case Diagram**

| Notasi  | Nama            | Deskripsi   |
|---|-----------------|---|
|  | <i>Use Case</i> | <i>Use case</i> menggambarkan bagaimana aktor menggunakan sistem untuk mencapai tujuannya. <i>Use case</i> menggambarkan bagaimana sistem dan |



| Notasi  | Nama                    | Deskripsi  |
|---|-------------------------|--|
|   |                         | aktor berkolaborasi untuk menghasilkan sebuah nilai (Bittner & Spence, 2002).  |
|    | Aktor                   | Aktor merupakan penggambaran dari sebuah peran yang diperankan oleh pengguna ketika berinteraksi dengan sistem. Pengguna bisa saja manusia atau sistem lain (Bittner & Spence, 2002).  |
|    | Association             | Menyatakan komunikasi antara aktor dengan <i>use case</i> .  |
|    | Extend                  | Penyisipan <i>behavior</i> tambahan kedalam sebuah <i>base use case</i> , dimana <i>use case</i> tambahan yang ditunjuk merupakan perluasan dari <i>base use case</i> .                |
|    | Include                 | Penyisipan <i>behavior</i> tambahan kedalam sebuah <i>base use case</i> , dimana sebuah <i>base use case</i> membutuhkan <i>use case</i> tambahan yang ditunjukkan.                    |
|  | Use Case Generalization | Hubungan antara <i>use case</i> umum dan <i>use case</i> yang lebih spesifik, dimana <i>use case</i> umum akan menurunkan/mewariskan fitur kepada <i>use case</i> yang lebih spesifik. |

Sumber: Diadaptasi dari Rumbaugh, Jacobson, & Booch (2004)



**Gambar 2.5 Contoh Use Case Diagram**

Sumber: Rumbaugh, Jacobson, & Booch (2004)

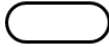







### 2.6.2 Activity Diagram

*Activity diagram* merupakan jenis diagram yang menunjukkan alur dari aktivitas ke aktivitas lainnya didalam sistem. *Activity diagram* menunjukkan *dynamic view* sebuah sistem. Diagram ini sangat penting khususnya dalam memodelkan fungsi sebuah sistem dan menekankan alur kontrol antar aktivitas-aktivitas (Booch, et al., 1998). *Activity diagram* melibatkan pemodelan langkah-langkah (*activities*) yang berurutan (dan mungkin langkah-langkah yang terjadi bersamaan) dalam proses komputasi. *Activities* pada akhirnya menghasilkan sebuah aksi yang menghasilkan komputasi yang dapat dieksekusi, yang akan mengubah *state* sebuah sistem atau nilai yang akan dikembalikan. *Activity diagram* pada umumnya berisi notasi *activity states* dan *action states*, *transition*, *object*. Tabel 2.2 menunjukkan notasi-notasi yang digunakan pada *activity diagram*. Pemodelan alur kerja dapat dilakukan dengan langkah sebagai berikut:

1. Membangun fokus pada alur kerja.
2. Memilih objek bisnis yang memiliki tanggung jawab untuk menjalankan alur kerja yang sudah ditentukan, didalam diagram, pisahkan masing-masing objek bisnis kedalam *swimlane*.
3. Identifikasi *precondition* dari sebuah rangkaian alur kerja dan *postcondition* yang dapat menyatakan kondisi akhir dari sebuah alur kerja.
4. Mulai dari kondisi awal sebuah rangkaian alur kerja, kemudian gambarkan masing-masing aktivitas secara berurutan ketika kondisi awal sudah terpenuhi kedalam notasi *activity diagram*.

5. Untuk setiap aksi atau aktivitas yang kompleks, atau sekumpulan aksi atau aktivitas yang dapat muncul beberapa kali, kelompokkan menjadi *activity states*. Kemudian jelaskan secara detil melalui *activity diagram* lain.
6. Gambarkan transisi yang saling menghubungkan aksi atau aktivitas. Mulai dari alur yang berurutan, kemudian gunakan *branch* jika diperlukan, serta gunakan *fork* atau *join* jika diperlukan.
7. Jika terdapat sebuah objek yang sangat penting, gambarkan objek tersebut didalam diagram, jika diperlukan gambarkan perubahan nilai dari objek tersebut.

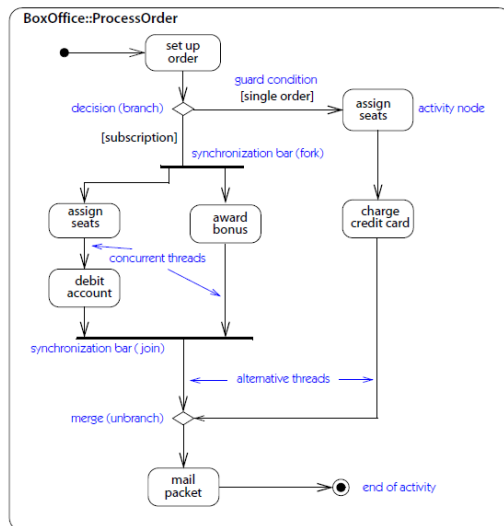
**Tabel 2.2 Notasi Dalam Activity Diagram**

| Notasi  | Nama                | Deskripsi  |
|---|---------------------|--|
|    | <i>Activity</i>     | Merepresentasikan aktivitas yang dikerjakan ketika sebuah sesuatu pada sebuah sistem dieksekusi.   |
|   | <i>Start</i>        | Merepresentasikan keadaan awal dari sebuah rangkaian aktivitas.  |
|  | <i>End</i>          | Merepresentasikan keadaan akhir dari sebuah rangkaian aktivitas.   |
|  | <i>Object flow</i>  | Merepresentasikan objek yang menjadi <i>input</i> atau <i>output</i> sebuah aktivitas.   |
|  | <i>Control flow</i> | Penghubung antar notasi aktivitas yang menggambarkan alur aktivitas yang dikerjakan.   |
|  | <i>Branch</i>       | Digunakan untuk menspesifikasikan alur alternatif yang akan ditempuh apabila sebuah kondisi dipenuhi.  |
|  | <i>Fork</i>         | Digunakan untuk menyatakan aktivitas yang dikerjakan secara <i>concurrent</i> . Masing-masing aktivitas pada setiap cabang bekerja secara independen.                        |
|  | <i>Join</i>         | Digunakan untuk menyatakan alur yang tersinkronisasi, seluruh aktivitas <i>concurrent</i> digabungkan dan aktivitas selanjutnya (dibawah notasi join) akan dilakukan apabila |

|  |        | seluruh aktivitas <i>concurrent</i> sudah mencapai <i>join</i> . |  |  |                 |  |
|--|--------|--|--|--|-----------------|--|
| <table><tr><th>Aktor</th><th>Sistem</th></tr><tr><td></td><td></td></tr></table> | Aktor  | Sistem   |  |  | <i>Swimlane</i> | Digunakan untuk mengorganisasi aktivitas-aktivitas berdasarkan organisasi bisnis yang bertanggung jawab untuk melaksanakan aktivitas tertentu. |
| Aktor  | Sistem |  |  |  |                 |  |
|  |        |  |  |  |                 |  |

Sumber: Diadaptasi dari Rumbaugh, Jacobson, & Booch (2004)

Dalam Gambar 2.6 menunjukkan sebuah rangkaian aktivitas didalam activity diagram. Ketika kondisi awal dari rangkaian aktivitas terpenuhi, maka aktivitas “*set up order*” akan dikerjakan, kemudian control akan melakukan pemeriksaan pada kondisi jenis order dari aktivitas “*set up order*”, jika order merupakan “*single order*” maka aktivitas “*assign seats*” akan dikerjakan. Selanjutnya, aktivitas “*charge credit card*” dikerjakan. Setelah selesai, kontrol akan melakukan unbranch dan selanjutnya aktivitas “*mail packet*” akan dikerjakan. Jika order merupakan “*subscription*”, kontrol akan melakukan *fork*, dimana aktivitas “*assign seats*” dan “*debit account*” akan dikerjakan secara *concurrent* dengan aktivitas “*award bonus*”. Setelah seluruh aktivitas *concurrent* selesai dan mencapai kondisi *join*, kontrol akan melakukan *unbranch*, yang kemudian dilanjutkan pengerjaan aktivitas “*mail packet*”. Setelah aktivitas “*mail packet*” selesai dikerjakan dan kondisi rangkaian aktivitas memenuhi kondisi akhir, maka rangkaian aktivitas akan dinyatakan selesai.



**Gambar 2.6 Activity Diagram**

Sumber: Rumbaugh, Jacobson, & Booch (2004)






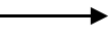
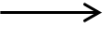
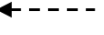
### 2.6.3 Sequence Diagram

*Sequence diagram* merupakan diagram yang menggambarkan interaksi dan menekankan urutan waktu dalam interaksi tersebut. Sequence diagram menunjukkan sekumpulan objek dan pesan yang dikirimkan dan diterima oleh objek-objek tersebut (Booch, et al., 1998). Sequence diagram menampilkan interaksi dalam skema dua dimensi. Dimensi vertical menunjukkan sumbu waktu, dimensi horizontal menunjukkan peran (*roles*) yang merepresentasikan objek secara individu yang saling berkolaborasi. Setiap peran (*role*) direpresentasikan dalam kolom vertical yang berisi symbol sisi paling atas, dan garis vertical (*lifeline*).

Secara umum sequence diagram hanya menunjukkan pesan yang berurutan dan tidak menunjukkan waktu interval pesan tersebut muncul secara tepat. Sebuah pesan digambarkan dalam bentuk anak panah dari *lifeline* dari sebuah objek yang menunjuk ke *lifeline* dari objek yang lain. Notasi untuk menggambarkan pesan dibagi menjadi 3, yaitu *synchronous message*, *asynchronous message*, dan *return message*. Notasi yang umum digunakan pada *activity diagram* ditunjukkan dalam Tabel 2.3.

**Tabel 2.3 Notasi dalam Sequence Diagram**

| Notasi | Nama | Deskripsi |
|--------|------|-----------|
|--------|------|-----------|

|   |                             |   |
|---|-----------------------------|---|
|    | Aktor                       | Merepresentasikan pengguna sistem.  |
|    | <i>Lifeline</i>             | Notasi yang digunakan untuk menunjukkan keberadaan sebuah objek, selain itu untuk menunjukkan bahwa sebuah objek aktif ketika mengirim atau menerima pesan. |
|    | <i>Boundary</i>             | Merepresentasikan antarmuka dan interaksi satu atau lebih aktor dengan sistem.  |
|    | <i>Control</i>              | Merepresentasikan objek yang mengatur perilaku dan alur suatu sistem.   |
|    | <i>Entity</i>               | Merepresentasikan objek yang bertanggung jawab menyimpan data.  |
|    | <i>Synchronous Message</i>  | Merepresentasikan relasi untuk memanggil operasi yang dimiliki objek, operasi yang dipanggil dikerjakan hingga selesai.                                     |
|  | <i>Asynchronous Message</i> | Merepresentasikan relasi untuk memanggil operasi yang dimiliki objek, dimana operasi yang dipanggil dapat dikerjakan secara paralel.                        |
|  | <i>Return Message</i>       | Merepresentasikan relasi yang digunakan untuk memberikan nilai kembali pada objek yang memanggil operasi pada objek yang lain.                              |

Sumber: Diadaptasi dari Rumbaugh, Jacobson, & Booch (2004)

#### 2.6.4 Class Diagram

*Class diagram* menunjukkan sekumpulan *class*, antarmuka, kolaborasi, dan hubungan antar *class*. *Class diagram* sering ditemukan pada pemodelan sistem berorientasi objek. *Class diagram* digunakan untuk mengilustrasikan tampilan perancangan statis sebuah sistem (Booch, et al., 1998). Sebuah *Class* terdiri dari atribut, operasi atau metode. Atribut merupakan variabel-variabel yang dimiliki sebuah *class*. Operasi atau metode merupakan fungsi yang dimiliki sebuah *class*. *Class diagram* memiliki tiga area, seperti yang ditunjukkan dalam Gambar 2.7.

| Name         |
|--------------|
| -attribute   |
| +operation() |

**Gambar 2.7 Gambar Class dan Area yang Dimiliki**

Sumber: Diadaptasi dari Booch, Rumbaugh, & Jacobson (1998)


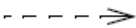
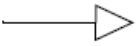

Dalam Gambar 2.7 menunjukkan area pada baris pertama merupakan nama *class*, area baris kedua berisi atribut yang dimiliki *class*, area baris ketiga berisi operasi atau fungsi yang dimiliki *class*. Atribut dan operasi memiliki *access level* untuk membatasi akses oleh *class* atau objek lain. *Access level* atribut dan operasi dijelaskan dalam Tabel 2.4. Sebuah *class* dapat berinteraksi dengan *class* yang lain, hubungan atau interaksi antar *class* ini diantaranya adalah *association*, *aggregation*, *composition*, *dependency*, dan *generalization*. Penjelasan notasi hubungan antar *class* dijelaskan dalam Tabel 2.5.


**Tabel 2.4 Access Level pada Atribut dan Operasi Class**

| Notasi | Nama             | Deskripsi  |
|--------|------------------|--|
| -      | <i>Private</i>   | Atribut atau operasi tidak dapat diakses di luar <i>class</i> .  |
| #      | <i>Protected</i> | Atribut atau operasi hanya dapat diakses oleh sebuah <i>class</i> dan <i>child class</i> yang mewarisinya. |
| +      | <i>Public</i>    | Atribut atau operasi dapat digunakan oleh seluruh <i>class</i> dalam sistem.                               |

Sumber: Diadaptasi dari Rumbaugh, Jacobson, & Booch (2004)

**Tabel 2.5 Notasi Hubungan Antar Class**

| Notasi  | Nama                  | Deskripsi  |
|---|-----------------------|--|
|  | <i>Association</i>    | Merepresentasikan hubungan antar <i>instance</i> dari <i>class</i> .   |
|  | <i>Dependency</i>     | Merepresentasikan hubungan antara dua elemen model.  |
|  | <i>Generalization</i> | Merepresentasikan hubungan antara <i>class</i> yang bersifat spesifik dengan <i>class</i> yang bersifat umum. Digunakan untuk menggambarkan pewarisan antar <i>class</i> . |
|  | <i>Aggregation</i>    | Merepresentasikan hubungan yang menyatakan bagian ( <i>part-whole</i> ).   |

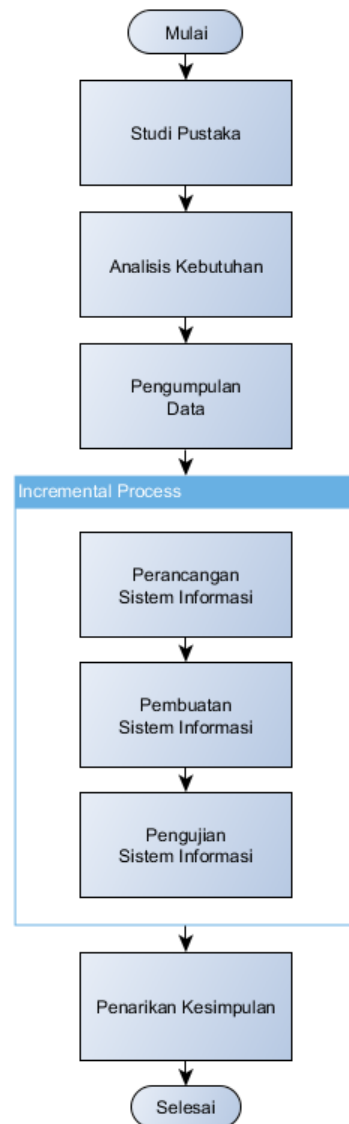
|   |                    |  |
|---|--------------------|--|
|  | <i>Composition</i> | Merepresentasikan hubungan antar <i>class</i> yang operasinya tidak dapat berdiri sendiri. |
|---|--------------------|--|

Sumber: Diadaptasi dari Rumbaugh, Jacobson, & Booch (2004)



## BAB 3 METODOLOGI

### 3.1 Alur Penelitian



**Gambar 3.1 Alur Penelitian**

### 3.1.1 Studi Pustaka

Dalam Gambar 3.1 penelitian dimulai dengan melakukan studi pustaka, yaitu dengan mengkaji penelitian terdahulu mengenai Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” yang menjadi landasan pengembangan lanjut SIKES “ANDAL”. Selain itu, pada tahap studi pustaka dilakukan pemahaman teori dan kerangka kerja yang berkaitan dan digunakan sebagai panduan untuk menyelesaikan penelitian, contohnya: teori tentang *Software Development Life Cycle* (SDLC), *Unified Modeling Language* (UML), dan perancangan arsitektur.

### 3.1.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan melakukan analisa terhadap proses bisnis yang ada pada Model Organisasi dan Manajemen Kesehatan Primer “ANDAL”, selain itu dilakukan juga dengan melakukan wawancara untuk mendapatkan informasi mengenai siapa saja *stakeholder*, aktor, dan tipe pengguna Sistem Informasi Kesehatan “ANDAL”. Wawancara juga digunakan untuk lebih memahami permasalahan dan kemungkinan solusi yang akan digunakan, selain itu wawancara juga digunakan untuk memastikan efektifitas wawancara itu sendiri, agar responden atau narasumber merasa nyaman dalam memberikan jawaban dalam kegiatan wawancara.

Analisa kebutuhan memiliki tujuan agar peneliti dapat memahami permasalahan dengan baik, menentukan batasan masalah yang akan diselesaikan, mengetahui lebih jauh siapa saja *stakeholder*, aktor, dan calon pengguna SIKES “ANDAL”, mengetahui lingkungan sistem secara teknis, mengetahui kebutuhan atau fungsi-fungsi yang harus diimplementasikan pada sistem, serta kemungkinan solusi dan kebutuhan yang akan digunakan dan dipenuhi untuk menyelesaikan masalah. Dalam penelitian ini wawancara dilakukan kepada *stakeholder* SIKES “ANDAL” yang meliputi dr. Jack Roebijoso dan peneliti yang menerbitkan Model Organisasi dan Manajemen Kesehatan Primer “ANDAL”.

Kegiatan yang dilakukan dalam pengumpulan kebutuhan mengadaptasi beberapa tugas dalam proses *Requirement Engineering* (RE) khususnya *inception*, *elicitation*, dan *negotiation*, seperti yang dijelaskan oleh Pressman (2010). Proses RE digunakan sebagai panduan untuk membantu peneliti memeriksa bagaimana SIKES “ANDAL” seharusnya berjalan, memahami kebutuhan secara spesifik, dan menentukan prioritas kebutuhan yang harus diimplementasikan. Hasil kegiatan analisa kebutuhan diantaranya daftar aktivitas-aktivitas berdasarkan proses bisnis pada Model Organisasi dan Manajemen Kesehatan Primer “ANDAL”, daftar fitur sistem informasi yang akan dibuat, serta kebutuhan fungsional dan non-fungsional sistem informasi yang akan dibuat. Hasil kegiatan analisa kebutuhan akan digunakan sebagai *input* pada tahap yang ada pada *incremental process*, yaitu perancangan sistem informasi.

### 3.1.3 Pengumpulan Data

Pengumpulan data dilakukan dengan mengumpulkan berkas-berkas data dasar yang berisi informasi identitas keluarga, riwayat penyakit atau kecelakaan yang

dialami anggota keluarga, serta perilaku kesehatan dan keselamatan melalui formulir data dasar yang diisi langsung oleh masyarakat pada daerah atau tempat kerja yang sudah ditentukan sebelumnya. Pengumpulan data akan dilakukan di Poliklinik Pabrik Gula Kebon Agung. Data yang sudah didapatkan akan disimpan kedalam basis data ketika tahap pembuatan sistem informasi.

#### **3.1.4 Incremental Process**

Kegiatan *Incremental process* dalam metodologi penelitian ini diadaptasi dari *Incremental Process Model* yang memiliki keuntungan, salah satunya, memungkinkan *delivery* dan *deployment* sistem kepada calon pengguna lebih cepat, meskipun belum semua fungsi dapat diimplementasikan kedalam sistem. Calon pengguna dapat dengan segera menggunakan dan mendapatkan nilai dari sistem yang di-*deliver* dibandingkan jika menerapkan model *Waterfall* (Sommerville, 2011). *Incremental process* pada penelitian ini mengadopsi pendekatan *plan-driven*, sehingga seluruh kebutuhan pengguna dan kebutuhan sistem sudah diidentifikasi terlebih dahulu pada tahap analisa kebutuhan.

Kemudian seluruh kebutuhan tersebut dapat dimodelkan sebagai panduan bagi peneliti untuk melakukan pembuatan perangkat lunak. Pembuatan perangkat lunak dilakukan dengan mengambil sebagian kebutuhan berdasarkan prioritasnya, kemudian hasil pengembangan tersebut dapat di-*deliver* kepada calon pengguna untuk dievaluasi dan diberi tanggapan. Hasil evaluasi dan tanggapan tersebut menjadi *input* bagi pengembangan pada *increment* selanjutnya hingga perangkat lunak dapat memberikan seluruh fitur yang dapat mencakup seluruh kebutuhan calon pengguna.

##### **3.1.4.1 Perancangan Sistem Informasi**

Perancangan sistem informasi meliputi pemodelan sistem kedalam notasi-notasi pada *Unified Modeling Language* (UML) berdasarkan hasil tahap analisa kebutuhan dan pengumpulan data. Seperti yang dijelaskan oleh Sommerville (2011) bahwa pemodelan digunakan untuk memperoleh kebutuhan sistem, menjelaskan kebutuhan sistem untuk memudahkan proses pengembangan sistem, serta untuk mendokumentasikan struktur dan operasi sistem. Hasil pemodelan sistem dapat membantu *stakeholder* memahami kebutuhan. Selain itu, bagi peneliti, hasil pemodelan digunakan sebagai dasar perancangan sistem dan sebagai dokumentasi dari sistem yang akan dibuat.

Pada penelitian ini, perancangan sistem informasi akan memanfaatkan *Use Case Diagram* untuk menunjukkan interaksi antara sistem dengan lingkungannya, *Activity Diagram* untuk menunjukkan aktivitas-aktivitas yang terlibat dalam sebuah proses dan pemrosesan data dalam sistem, *Sequence Diagram* untuk menunjukkan interaksi antara aktor dengan sistem dan antara aktor dengan komponen sistem, *Class Diagram* untuk menunjukkan kelas-kelas objek didalam sistem dan hubungan antar kelas tersebut.

#### **3.1.4.2 Pembuatan Sistem Informasi**

Pembuatan sistem informasi merupakan proses pengembangan SIKES “ANDAL” berdasarkan hasil perancangan pada tahap perancangan sistem informasi. SIKES “ANDAL” dibangun menggunakan bahasa pemrograman PHP, dengan framework Codeigniter. SIKES “ANDAL” dari penelitian ini merupakan purwarupa dengan fitur inti berdasarkan Model Organisasi dan Manajemen Kesehatan Primer “ANDAL”. Pembuatan sistem informasi dilakukan sebagai bagian dari *incremental process model*, sehingga pembuatan sistem informasi akan dilakukan secara berkala dengan mengambil beberapa bagian dari kebutuhan pengguna dan kebutuhan sistem untuk diimplementasikan pada setiap *increment*. Hasil dari setiap *increment* akan digunakan sebagai *input* pada tahap pengujian sistem informasi.

#### **3.1.4.3 Pengujian Sistem Informasi**

Pengujian merupakan sebuah elemen penting dari penjaminan kualitas perangkat lunak dan merepresentasikan *review* akhir dari spesifikasi, perancangan dan implementasi (Pressman, 2010). Pengujian SIKES “ANDAL” dilakukan dengan menggunakan pengujian *functional testing* untuk memastikan sistem dapat memberikan fungsi sesuai dengan yang telah ditentukan, serta untuk memastikan sistem tidak mengalami kegagalan pada saat digunakan.

Pengujian dilakukan terhadap hasil pembuatan sistem informasi pada setiap *increment*, jika pengujian dinyatakan tidak valid, maka pembuatan sistem pada *increment* selanjutnya digunakan untuk memperbaiki kesalahan yang telah dibuktikan melalui tahap pengujian. Jika pengujian dinyatakan valid dan masih terdapat kebutuhan sistem yang belum terimplementasi, maka *increment* selanjutnya digunakan untuk mengimplementasi kebutuhan-kebutuhan yang lain. Jika pengujian valid dan seluruh kebutuhan telah terimplementasi, maka penelitian dapat dianggap selesai. Hasil pengujian dari *increment* terakhir akan digunakan untuk tahap penarikan kesimpulan untuk menjawab rumusan masalah.

#### **3.1.5 Penarikan Kesimpulan**

Penarikan kesimpulan dilakukan berdasarkan penilaian terhadap hasil akhir akhir penelitian untuk menentukan apakah hasil penelitian dapat menjawab masalah yang telah dirumuskan.

## BAB 4 ANALISIS PERSYARATAN

Sistem Informasi Kesehatan “ANDAL” saat ini telah dikembangkan dan mampu memberikan fitur pengelolaan data dasar kesehatan keluarga, pengelolaan data registrasi dan status pasien, menampilkan tingkat resiko kesehatan dan tingkat stres, pengelolaan hasil diagnosis pasien, sehingga sistem informasi kesehatan saat ini dapat membantu petugas klinik mengelola data dasar kesehatan keluarga, mengelola data registrasi dan status pasien, menampilkan tingkat resiko kesehatan dan tingkat stres, dan mengelola hasil diagnosis pasien.

Berdasarkan hasil wawancara dengan dr. Jack Roebijoso, sistem informasi kesehatan diharapkan dapat dikembangkan untuk dapat membantu petugas klinik mengelola data tindakan medis yang diberikan kepada pasien, mengelola data rujukan pasien, mengelola data pengecekan darah pasien, mengelola data kegiatan edukasi, advokasi, dan pengelolaan lingkungan tempat tinggal atau tempat kerja pasien dan mampu menampilkan persentase penyakit yang tersebar mulai dari tingkat RT, RW, kelurahan, kecamatan, hingga kota/kabupaten.

Bab ini membahas analisis persyaratan yang terdiri dari hasil analisis kebutuhan, fitur-fitur yang akan dikembangkan, persyaratan fungsional dan non-fungsional, perancangan *use case diagram* beserta *use case specification*, dan *activity diagram*.

### 4.1 Analisis Persyaratan

#### 4.1.1 Analisis Tipe Stakeholder

Stakeholder adalah individu yang secara materi terpengaruh oleh hasil dari sebuah sistem atau sebuah proyek yang menghasilkan sistem (Bittner & Spence, 2002). Bagian ini menjelaskan klasifikasi tipe *stakeholders*, hubungan *stakeholder* terhadap sistem yang akan dibangun, dan menunjukkan organisasi atau individu yang termasuk kedalam masing-masing tipe *stakeholder*. Hasil identifikasi *stakeholder* dalam Tabel 4.1 didapatkan berdasarkan wawancara dengan dr. Jack Roebijoso selaku penggagas Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” sekaligus Kepala Poliklinik Pabrik Gula Kebon Agung.

**Tabel 4.1 Tipe Stakeholder**

| Tipe Stakeholder | Deskripsi   | Contoh Stakeholder   |
|------------------|---|--|
| Pengguna         | Individu yang berinteraksi dengan sistem informasi kesehatan secara langsung dengan memerankan peran yang didefinisikan oleh aktor. | Petugas Poliklinik Pabrik Gula Kebon Agung yang terdiri dari Kepala Poliklinik, Tenaga Medis, Tenaga Keperawatan, Tenaga Kefarmasian, dan Staf Administrasi. |
| Pengembang       | Organisasi atau individu yang melakukan pengembangan sistem informasi kesehatan.  | Peneliti   |

**Commented [F3]:** Sesuai dengan panduan pada (Bittner & Spence, 2002) Chapter 3 – Introducing Stakeholders and Users – Identifying Stakeholder Types, Identifying Stakeholder Representatives and Stakeholder Roles

**Tabel 4.1 Tipe Stakeholder (lanjutan)**

| <b>Tipe Stakeholder</b> | <b>Deskripsi</b>  | <b>Contoh Stakeholder</b>  |
|-------------------------|---|--|
| Pihak yang berwenang    | Organisasi atau individu yang terkait dengan masalah yang akan diselesaikan dengan bantuan sistem informasi kesehatan yang akan dikembangkan. | Poliklinik Pabrik Gula Kebon Agung.  |
| Pelanggan               | Organisasi atau individu yang akan mendapatkan manfaat dari sistem informasi kesehatan.   | Poliklinik Pabrik Gula Kebon Agung, pasien Poliklinik Pabrik Gula Kebon Agung. |

#### 4.1.2 Analisis Masalah

Analisa masalah dilakukan untuk memastikan seluruh fungsi yang disediakan oleh sistem dapat mengatasi masalah secara langsung dan kesuksesan sebuah produk (sistem) (Bittner & Spence, 2002). Dalam penelitian ini, analisis masalah dilakukan dengan melakukan wawancara dengan dr. Jack Roebijoso selaku penggagas Model Organisasi dan Manajemen Kesehatan Primer “ANDAL” dan Kepala Poliklinik Pabrik Gula Kebon Agung. Hasil analisis masalah disajikan dalam Tabel 4.2.

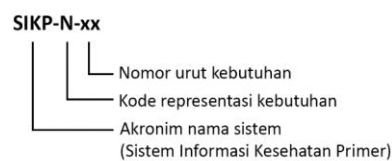
**Commented [F4]:** Contoh template bisa dilihat di (Krol & Kruchten, 2003) Bagian “Develop a Vision” lembar ke 267

**Tabel 4.2 Problem Statement**

|                                    |  |
|------------------------------------|--|
| <b>The problem of</b>              | <ul style="list-style-type: none"> <li>Pencatatan data kesehatan yang tidak konsisten yang disebabkan duplikasi dan hilangnya dokumen cetak data kesehatan.</li> <li>Penanganan kesehatan yang diberikan kepada pasien belum konsisten dan efektif karena tidak dilakukan berdasarkan data kesehatan yang konsisten.</li> <li>Penanganan kesehatan yang diberikan kepada pasien belum konsisten dan efektif karena tidak ada panduan standar pelayanan kesehatan.</li> </ul> |
| <b>Affects</b>                     | <ul style="list-style-type: none"> <li>Keterlambatan petugas poliklinik memberikan layanan kesehatan kepada pasien.</li> <li>Angka sakit yang disebabkan penyakit menular atau tidak menular meningkat.</li> </ul>   |
| <b>The impact of which is</b>      | Kualitas pelayanan kesehatan yang diberikan poliklinik kurang memuaskan.   |
| <b>A successful solution would</b> | Pelayanan kesehatan berupa layanan kesehatan yang bersifat promotif, preventif, dan kuratif yang diberikan kepada pasien semakin efektif berdasarkan kebutuhan pasien. Secara berangsur-angsur menurunkan angka sakit yang disebabkan oleh penyakit menular dan penyakit tidak menular pasien BPJS Poliklinik Pabrik Gula Kebon Agung.   |

#### 4.1.3 Analisis Kebutuhan Stakeholder

Bittner dan Spence menjelaskan bahwa pada dasarnya *stakeholder* memiliki perspektif yang berbeda-beda pada masalah dan memiliki kebutuhan yang berbeda, yang harus diselesaikan dengan solusi yang berbeda. Hal tersebut dapat diketahui dengan cara mengidentifikasi dan melakukan dokumentasi kebutuhan masing-masing tipe *stakeholder*. Identifikasi kebutuhan *stakeholder* dapat membantu memahami bagaimana dan apa saja aspek-aspek masalah yang mempengaruhi masing-masing tipe pengguna (Bittner & Spence, 2002).



**Gambar 4.1 Keterangan Kodifikasi Kebutuhan**

Sub bab ini menjelaskan kebutuhan dari masing-masing *stakeholder*, prioritas kebutuhan, penjelasan situasi saat ini dalam hal pemenuhan kebutuhan, serta solusi apa saja yang dapat memenuhi kebutuhan *stakeholder*. Kebutuhan *stakeholder* dalam penelitian ini didapatkan berdasarkan hasil wawancara dengan petugas Poliklinik Pabrik Gula Kebon Agung yang termasuk dalam tipe *stakeholder* pengguna. Hasil identifikasi kebutuhan *stakeholder* dijelaskan dalam Tabel 4.3. Setiap kebutuhan *stakeholder* diberi kode dengan aturan kode sesuai dalam Gambar 4.1.

**Tabel 4.3 Kebutuhan Stakeholder**

| Kode Kebutuhan | Kebutuhan                                | Prioritas | Situasi Saat Ini  | Stakeholder   | Solusi yang Ditawarkan  |
|----------------|--|-----------|---|---|---|
| SIKP-N-01      | Mengelola data dasar kesehatan keluarga. | M         | Pengelolaan data saat ini meliputi kegiatan pencatatan data dalam format cetak, penyimpanan dilakukan di dalam folder poliklinik. | Kepala poliklinik, tenaga medis, tenaga keperawatan, staf administrasi. | Sistem menyediakan fitur untuk melihat, menambah, mengubah data dasar kesehatan keluarga. |
| SIKP-N-02      | Mengelola data status pasien.            | M         | Pengelolaan data saat ini meliputi kegiatan pencatatan data dalam format cetak dengan nama "Kartu Berobat", penyimpanan           | Staf administrasi.  | Sistem menyediakan fitur untuk melihat, menambah, mengubah data status pasien.            |

**Commented [F5]:** Tabulasi kebutuhan stakeholder sesuai dengan panduan pada (Bittner & Spence, 2002) Chapter 3 – Creating a Shared Vision - Understand the Key Stakeholder and User Needs – (poin-poin pad bullet numbering)

**Tabel 4.3 Kebutuhan Stakeholder (lanjutan)**

| Kode Kebutuhan | Kebutuhan   | Prioritas | Situasi Saat Ini   | Stakeholder                      | Solusi yang Ditawarkan   |
|----------------|---|-----------|--|----------------------------------|--|
|                |   |           | dilakukan di dalam folder poliklinik.  |                                  |  |
| SIKP-N-03      | Melihat tingkat risiko penyakit dan tingkat stres.  | M         | Tingkat risiko penyakit dan tingkat stres dapat diketahui dengan terlebih dahulu melakukan rekapitulasi data dasar kesehatan keluarga yang dilakukan dengan bantuan aplikasi Microsoft Excel.    | Kepala poliklinik, tenaga medis. | Sistem menyediakan fitur untuk mengelola data dasar kesehatan keluarga sehingga menghasilkan tingkat risiko penyakit dan tingkat stres pasien. |
| SIKP-N-04      | Mengelola data hasil diagnosis pasien yang terdiri dari data hasil diagnosis pasien, faktor pemicu penyakit, faktor risiko penyakit, dan tindakan medis yang diberikan kepada pasien. | M         | Pengelolaan data saat ini meliputi kegiatan pencatatan data dalam format cetak dalam kolom anamnese, diagnosis yang tersedia di Kartu Berobat, penyimpanan dilakukan di dalam folder poliklinik. | Tenaga medis.                    | Sistem menyediakan fitur untuk melihat, dan menambah data hasil diagnosis pasien.  |
| SIKP-N-05      | Mengelola data tindakan medis yang diberikan kepada pasien.   | M         | Pengelolaan data saat ini meliputi kegiatan pencatatan data dalam format cetak dalam kolom terapi yang tersedia di Kartu Berobat, penyimpanan dilakukan di                                       | Tenaga medis.                    | Sistem menyediakan fitur untuk melihat dan menambah data tindakan medis.   |



**Tabel 4.3 Kebutuhan Stakeholder (lanjutan)**

| Kode Kebutuhan | Kebutuhan   | Prioritas | Situasi Saat Ini  | Stakeholder        | Solusi yang Ditawarkan  |
|----------------|---|-----------|---|--------------------|---|
|                |   |           | dalam folder poliklinik.  |                    |   |
| SIKP-N-06      | Melihat modul yang berisi panduan untuk menghindari dan mengatasi penyakit.                 | M         | Modul disimpan dalam format dokumen Microsoft Word dan hanya dapat ditampilkan melalui aplikasi Microsoft Word.   | Tenaga medis       | Sistem menyediakan fitur yang memungkinkan tenaga medis membaca modul panduan untuk menghindari dan mengatasi penyakit. |
| SIKP-N-07      | Mengelola data rujukan pasien ke rumah sakit atau UGD, dan mengelola data cek darah pasien. | M         | Berkas rujukan pasien tersedia dalam bentuk cetak, berkas tersebut merupakan keluaran dari aplikasi PCare (aplikasi resmi BPJS). Penyimpanan berkas rujukan dilakukan di dalam folder poliklinik. | Tenaga medis.      | Sistem menyediakan fitur untuk melihat dan menambah data rujukan ke rumah sakit atau ke UGD, dan data cek darah pasien. |
| SIKP-N-08      | Mengelola pencatatan obat keluar.   | M         | Pencatatan obat keluar dilakukan pada formulir cetak, kemudian data formulir cetak disalin kedalam dokumen dengan format Microsoft Excel..  | Tenaga Kefarmasian | Sistem menyediakan fitur untuk merekam pencatatan obat keluar.  |
| SIKP-N-09      | Mengetahui hasil rekapitulasi pencatatan obat keluar setiap bulan.                          | M         | Rekapitulasi pencatatan obat dilakukan dengan bantuan aplikasi Microsoft Excel, kemudian hasil rekapitulasi dicetak untuk kebutuhan   | Tenaga Kefarmasian | Sistem menyediakan fitur untuk menampilkan laporan rekapitulasi pencatatan obat.  |

Tabel 4.3 Kebutuhan Stakeholder (lanjutan)

| Kode Kebutuhan | Kebutuhan | Prioritas | Situasi Saat Ini                   | Stakeholder | Solusi yang Ditawarkan |
|----------------|-----------|-----------|------------------------------------|-------------|------------------------|
|                |           |           | pengisian kembali persediaan obat. |             |                        |

#### 4.1.4 Analisis Tipe Pengguna

Tipe pengguna adalah klasifikasi dari sekumpulan pengguna yang memiliki peran dan tanggung jawab yang sama didalam lingkungan sistem (Bittner & Spence, 2002). Bittner dan Spence menjelaskan bahwa pengguna adalah tipe *stakeholder* yang menggunakan sistem secara langsung.

Tipe pengguna sistem didapatkan berdasarkan wawancara dengan dr. Jack Roebijoso selaku Kepala Poliklinik Pabrik Gula Kebon Agung dan pengamatan terhadap keberadaan petugas Poliklinik Pabrik Gula Kebon Agung. Tabel 4.4 menjelaskan tipe-tipe pengguna sistem informasi kesehatan yang akan dikembangkan dalam penelitian ini.

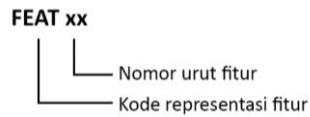
Tabel 4.4 Tipe Pengguna

| Tipe Stakeholder | Tipe Pengguna      | Deskripsi  |
|------------------|--------------------|--|
| Pengguna         | Kepala Poliklinik  | Petugas poliklinik yang berprofesi sebagai dokter dan memiliki kedudukan yang tertinggi pada struktur organisasi poliklinik. |
|                  | Tenaga Medis       | Petugas poliklinik yang berprofesi sebagai dokter.   |
|                  | Tenaga Keperawatan | Petugas poliklinik yang berprofesi sebagai perawat dan bidan.  |
|                  | Tenaga Kefarmasian | Petugas poliklinik yang bertugas mengelola ketersediaan obat dan memberikan obat kepada pasien sesuai dengan resep obat.     |
|                  | Staf Administrasi  | Petugas poliklinik khusus bidang administrasi.   |

**Commented [F6]:** Sesuai panduan pada (Bittner & Spence, 2002) Chapter 3 – Introducing Stakeholders and Users - Users: A Very Important Class of Stakeholder

#### 4.1.5 Analisis Fitur

Fitur adalah layanan dari sebuah sistem yang diperlukan untuk menghasilkan nilai atau manfaat kepada pengguna dan dapat memenuhi kebutuhan *stakeholder* dan pengguna (Bittner & Spence, 2002). Fitur yang akan disediakan oleh sistem dalam penelitian ini memiliki kode fitur dengan kodifikasi sesuai dalam Gambar 4.2.



**Gambar 4.2 Kodifikasi Fitur**

Fitur yang akan disediakan kepada pengguna melalui sistem informasi kesehatan merupakan solusi yang ditawarkan untuk memenuhi kebutuhan pengguna yang dijelaskan dalam Tabel 4.3. Prioritas fitur didapatkan berdasarkan wawancara dengan dr. Jack Roebijoso selaku Kepala Poliklinik Pabrik Gula Kebon Agung dan petugas lain di Poliklinik Pabrik Gula Kebon Agung yang termasuk dalam *stakeholder* yang menyatakan kebutuhan sesuai dalam Tabel 4.3. Penjelasan fitur ada dalam Tabel 4.5.

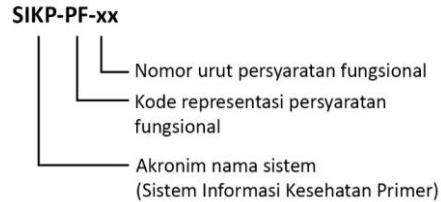
**Tabel 4.5 Fitur Sistem Informasi Kesehatan**

| Kode Fitur | Nama Fitur                               | Deskripsi   | Prioritas |
|------------|--|---|-----------|
| FEAT1      | Otentikasi Pengguna                      | Sistem dapat mengenali pengguna berdasarkan <i>username</i> dan <i>password</i> yang dimasukkan oleh pengguna. Setelah pengguna dikenali, sistem dapat menentukan data atau aksi apa saja yang berhak diakses oleh pengguna.  | M         |
| FEAT2      | Mengelola data dasar kesehatan keluarga. | Sistem dapat digunakan untuk mengelola data dasar kesehatan keluarga. Data dasar kesehatan keluarga terdiri dari data kepala keluarga, anggota keluarga, ekonomi keluarga, perilaku kesehatan anggota keluarga, dan riwayat kesehatan anggota keluarga. Pengelolaan data terdiri dari melihat, menambah, dan mengubah data. | M         |
| FEAT3      | Mengelola data status pasien.            | Sistem dapat digunakan untuk mengelola data status pasien poliklinik. Data status pasien terdiri dari keluhan penyakit, tekanan darah, denyut nadi, dan suhu badan pasien. Pengelolaan data terdiri dari melihat, menambah, dan mengubah data.  | M         |
| FEAT4      | Mengelola data pengobatan holistik       | Sistem dapat digunakan untuk mengelola data pengobatan holistik yang terdiri dari data hasil diagnosis pasien, faktor risiko penyakit, faktor pemicu penyakit, dan tindakan medis yang diberikan (termasuk rujukan dan tindakan cek darah pasien).  | M         |
| FEAT5      | Melihat modul penyakit                   | Sistem dapat digunakan untuk melihat isi modul penyakit yang berisi panduan tentang pencegahan dan penanganan penyakit.   | M         |
| FEAT6      | Melihat modul faktor risiko penyakit     | Sistem dapat digunakan untuk melihat isi modul faktor risiko penyakit yang berisi panduan tentang pencegahan dan penanganan penyakit sesuai dengan faktor risikonya.  | M         |

**Tabel 4.5 Fitur Sistem Informasi Kesehatan (lanjutan)**

| Kode Fitur | Nama Fitur  | Deskripsi  | Prioritas |
|------------|---|--|-----------|
| FEAT7      | Melihat modul faktor pemicu penyakit                                | Sistem dapat digunakan untuk melihat isi modul faktor pemicu penyakit yang berisi panduan tentang pencegahan dan penanganan penyakit sesuai dengan faktor pemicunya.   | M         |
| FEAT8      | Mengelola data pengantar cek darah                                  | Sistem dapat digunakan untuk mengelola data surat pengantar cek darah. Pengelolaan data terdiri dari melihat dan menambah data.  | M         |
| FEAT9      | Mengelola data obat   | Sistem dapat digunakan untuk mengelola data obat yang tersedia di apotek poliklinik. Sistem dapat digunakan untuk mencatat obat keluar termasuk didalamnya data mengenai dokter pemberi resep. Sistem dapat digunakan untuk mengecek status ketersediaan dan status pembiayaan obat. | M         |
| FEAT10     | Melihat hasil perhitungan tingkat risiko penyakit dan tingkat stres | Sistem dapat digunakan untuk melihat hasil perhitungan tingkat risiko penyakit dan tingkat stres pada masing-masing pasien berdasarkan data dasar kesehatan keluarga.  | M         |

#### 4.1.6 Persyaratan Fungsional



**Gambar 4.3 Kodifikasi Persyaratan Fungsional**

Bagian ini menjelaskan persyaratan fungsional sistem yang akan dikembangkan dalam penelitian ini. Persyaratan fungsional merupakan kondisi atau kemampuan yang harus dipenuhi oleh sistem agar fitur sistem dapat berjalan dan dapat memenuhi kebutuhan pengguna. Tabel 4.6 menunjukkan hubungan antara fitur dengan persyaratan fungsional pada sistem informasi kesehatan yang akan dikembangkan. Persyaratan fungsional memiliki kode sesuai dengan kodifikasi dalam Tabel 4.3.

**Tabel 4.6 Persyaratan Fungsional**

| Kode Fitur | Kode Persyaratan Fungsional | Deskripsi   |
|------------|-----------------------------|---|
| FEAT1      | SIKP-PF-01                  | Sistem dapat mengenali peran pengguna terhadap sistem berdasarkan username dan password yang dimasukkan oleh pengguna.                                      |
|            | SIKP-PF-02                  | Sistem dapat membatasi akses pengguna terhadap data dan fungsi sistem sesuai dengan peran pengguna.   |
| FEAT2      | SIKP-PF-03                  | Sistem dapat digunakan untuk melihat data dasar kesehatan keluarga yang tersimpan di sistem.  |
|            | SIKP-PF-04                  | Sistem dapat digunakan untuk menambah data dasar kesehatan keluarga.  |
|            | SIKP-PF-05                  | Sistem dapat digunakan untuk mengubah data dasar kesehatan keluarga yang tersimpan di sistem.   |
| FEAT3      | SIKP-PF-06                  | Sistem dapat digunakan untuk melihat data status pasien poliklinik yang tersimpan di sistem.  |
|            | SIKP-PF-07                  | Sistem dapat digunakan untuk menambah data status pasien poliklinik.  |
|            | SIKP-PF-08                  | Sistem dapat digunakan untuk mengubah data status pasien poliklinik yang tersimpan di sistem.   |
| FEAT4      | SIKP-PF-09                  | Sistem dapat digunakan untuk melihat data hasil diagnosis yang telah diberikan kepada pasien, yang tersimpan di sistem.                                     |
|            | SIKP-PF-10                  | Sistem dapat digunakan untuk menambah data hasil diagnosis yang telah diberikan kepada pasien.  |
|            | SIKP-PF-11                  | Sistem dapat digunakan untuk melihat data faktor risiko penyakit pasien yang tersimpan di sistem.   |
|            | SIKP-PF-12                  | Sistem dapat digunakan untuk menambah data faktor risiko penyakit pasien.   |
|            | SIKP-PF-13                  | Sistem dapat digunakan untuk melihat data faktor pemicu penyakit pasien yang tersimpan di sistem.   |
|            | SIKP-PF-14                  | Sistem dapat digunakan untuk menambah data faktor pemicu penyakit pasien.   |
|            | SIKP-PF-15                  | Sistem dapat digunakan untuk melihat data tindakan medis yang pernah diberikan kepada pasien.   |
|            | SIKP-PF-16                  | Sistem dapat digunakan untuk menambah data tindakan medis yang diberikan kepada pasien.   |
|            | SIKP-PF-17                  | Sistem dapat digunakan untuk melihat data rujukan yang pernah diberikan kepada pasien.  |
|            | SIKP-PF-18                  | Sistem dapat digunakan untuk menambah data rujukan yang diberikan kepada pasien.  |
|            | SIKP-PF-19                  | Sistem dapat digunakan untuk melihat data cek darah yang pernah diberikan kepada pasien.  |
|            | SIKP-PF-20                  | Sistem dapat digunakan untuk menambah data pengantar cek darah kepada pasien.   |
| FEAT5      | SIKP-PF-21                  | Sistem dapat digunakan untuk melihat isi modul penyakit yang berisi panduan cara mencegah dan mengatasi penyakit berdasarkan penyakit yang diderita pasien. |

**Tabel 4.6 Persyaratan Fungsional (lanjutan)**

| Kode Fitur | Kode Persyaratan Fungsional | Deskripsi   |
|------------|-----------------------------|---|
| FEAT6      | SIKP-PF-22                  | Sistem dapat digunakan untuk melihat isi modul faktor risiko penyakit yang berisi panduan cara mencegah dan mengatasi penyakit berdasarkan faktor risiko penyakit pasien. |
| FEAT7      | SIKP-PF-23                  | Sistem dapat digunakan untuk melihat isi modul faktor pemicu penyakit yang berisi panduan cara mencegah dan mengatasi penyakit berdasarkan faktor pemicu penyakit pasien. |
| FEAT8      | SIKP-PF-24                  | Sistem dapat digunakan untuk melihat data pengantar cek darah pasien.   |
|            | SIKP-PF-25                  | Sistem dapat digunakan untuk menambah data pengantar cek darah pasien.  |
| FEAT9      | SIKP-PF-26                  | Sistem dapat digunakan untuk merekam data obat keluar.  |
|            | SIKP-PF-27                  | Sistem dapat digunakan untuk memeriksa ketersediaan obat dan status pembiayaan obat.  |
| FEAT10     | SIKP-PF-28                  | Sistem dapat digunakan untuk menghitung tingkat risiko penyakit pasien berdasarkan data dasar kesehatan keluarga.   |
|            | SIKP-PF-29                  | Sistem dapat digunakan untuk menghitung tingkat stres pasien berdasarkan data dasar kesehatan keluarga.   |
|            | SIKP-PF-30                  | Sistem dapat menampilkan hasil perhitungan tingkat risiko penyakit pasien.  |
|            | SIKP-PF-31                  | Sistem dapat menampilkan hasil perhitungan tingkat stres pasien.  |

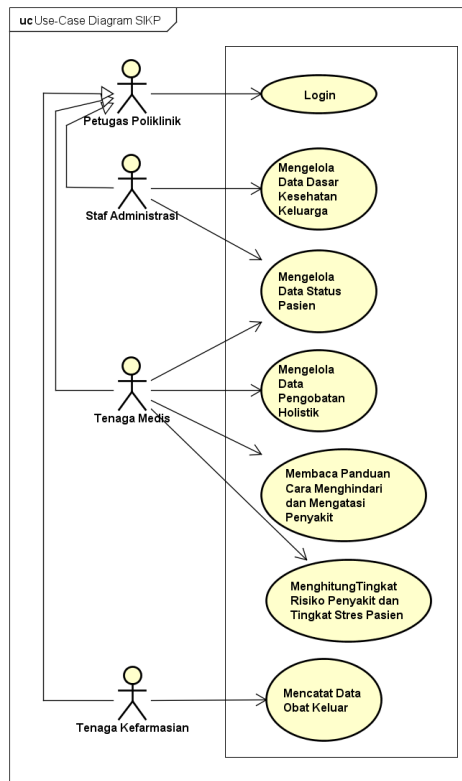
## 4.2 Use-Case Model

Sub bab ini berisi hasil pemodelan *use-case* yang terdiri dari diagram *use-case* yang menggambarkan hubungan antara *use-case* dengan aktor atau calon pengguna Sistem Informasi Kesehatan “ANDAL”, deskripsi mengenai aktor pada diagram *use-case*, dan *use-case description* yang berisi penjelasan lebih lanjut dari masing-masing *use-case* yang ada pada diagram *use-case*.

### 4.2.1 Diagram Use-Case

Gambar 4.4 menunjukkan diagram *use-case* dari Sistem Informasi Kesehatan “ANDAL”. Penamaan aktor berdasarkan hasil analisis tipe pengguna yang dijelaskan dalam Tabel 4.4, aktor merupakan peran yang akan diperankan oleh tipe pengguna. Penamaan *use-case* disesuaikan untuk merepresentasikan nilai atau tujuan yang ingin dicapai oleh aktor dalam menggunakan sistem informasi kesehatan yang akan dikembangkan.

**Commented [F7]:** represent the things of value that the system performs for its actors. Use cases are not functions or features, and they cannot be decomposed (Bittner & Spence, 2002). Bagian Actors and Use-Cases



**Gambar 4.4 Use-Case Diagram Sistem Informasi Kesehatan “ANDAL”**

Use-case dalam Gambar 4.4 memiliki hubungan dengan fitur yang sudah dijelaskan dalam Tabel 4.5. Hubungan fitur dengan use-case dapat dilihat dalam Tabel 4.7.

**Tabel 4.7 Hubungan Fitur dengan Use-Case**

| Fitur | Use-Case  |
|-------|---|
| FEAT1 | Login   |
| FEAT2 | Mengelola Data Dasar Kesehatan Keluarga                 |
| FEAT3 | Mengelola Data Status Pasien                            |
| FEAT4 | Mengelola Data Pengobatan Holistik                      |
| FEAT8 |   |
| FEAT5 | Membaca Panduan Cara Menghindari dan Mengatasi Penyakit |
| FEAT6 |   |
| FEAT7 |   |
|       |   |

**Commented [F8]:** Sesuai dengan panduan pada Chapter 4, sub bab Documenting Use-Cases, Trace the Use Cases to the Features and Constraints (Bittner & Spence, 2002)

**Tabel 4.7 Hubungan Fitur dengan *Use-Case* (lanjutan)**

| Fitur  | <i>Use-Case</i>   |
|--------|---|
| FEAT10 | Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien |
| FEAT9  | Mencatat Data Obat Keluar                                   |

#### 4.2.2 Deskripsi Aktor

Bagian ini berisi deskripsi aktor-aktor yang ada pada diagram *use-case*. Deskripsi aktor dapat dilihat dalam Tabel 4.8. Aktor pada penelitian ini merepresentasikan peran yang diperankan oleh individu yang akan menggunakan sistem secara langsung.

**Tabel 4.8 Deskripsi Aktor**

| Nama Aktor         | Deskripsi   |
|--------------------|---|
| Petugas Poliklinik | Aktor Petugas Poliklinik diperankan oleh seluruh petugas Poliklinik Pabrik Gula Kebon Agung yang menggunakan sistem secara langsung.                    |
| Tenaga Kefarmasian | Aktor Tenaga Kefarmasian diperankan oleh petugas Poliklinik Pabrik Gula Kebon Agung yang berprofesi sebagai apoteker.                                   |
| Staf Administrasi  | Aktor Staf Administrasi diperankan oleh petugas Poliklinik Pabrik Gula Kebon Agung yang berprofesi sebagai petugas administrasi dan tenaga keperawatan. |
| Tenaga Medis       | Aktor Tenaga Medis diperankan oleh petugas Poliklinik Pabrik Gula Kebon Agung yang berprofesi sebagai dokter, termasuk Kepala Poliklinik.               |

#### 4.2.3 *Use-Case Description*

Bagian ini berisi deskripsi dari beberapa *use-case* yang termasuk sebagai fungsi utama sistem informasi kesehatan yang akan dibangun dalam penelitian ini.

##### 4.2.3.1 *Use-Case Description Mengelola Data Pengobatan Holistik*

*Use-case* description pada Tabel 4.9 menjelaskan kondisi dan alur kegiatan ketika aktor Tenaga Medis menggunakan sistem untuk melihat atau menambah data hasil diagnosis yang diberikan kepada pasien, melihat atau menambah faktor risiko penyakit pasien, melihat atau menambah faktor pemicu penyakit pasien, melihat atau menambah data tindakan medis yang diberikan kepada pasien, melihat atau menambah data rujukan yang diberikan kepada pasien, melihat atau menambah data cek darah pasien.

**Tabel 4.9 *Use-Case Description* Mengelola Data Pengobatan Holistik**

| <i>Brief Description</i> | <i>Use-case</i> Mengelola Data Pengobatan Holistik menjelaskan bagaimana aktor Tenaga Medis menggunakan sistem untuk melihat atau menambah data hasil diagnosis yang diberikan kepada pasien, melihat atau menambah faktor risiko penyakit pasien, melihat atau menambah faktor pemicu penyakit pasien, melihat atau menambah data tindakan medis |
|--------------------------|---|
|--------------------------|---|

**Commented [F9]:** Referensi format penulisan mengacu pada (Bittner & Spence, 2002), bagian Appendix, *Use-Case Description—Withdraw Cash*



**Tabel 4.9 Use-Case Description Mengelola Data Pengobatan Holistik**

|                         |  |
|-------------------------|--|
|                         | yang diberikan kepada pasien, melihat atau menambah data rujukan yang diberikan kepada pasien, melihat atau menambah data cek darah pasien.  |
| <b>Actor</b>            | Tenaga Medis   |
| <b>Pre-condition</b>    | <ul style="list-style-type: none"> <li>• <i>Login</i> yang dilakukan aktor Tenaga Medis dinyatakan berhasil.</li> <li>• <i>Session login</i> aktor sebagai Tenaga Medis masih tersimpan dalam sistem.</li> <li>• Komputer yang digunakan untuk Mengelola Data Pengobatan Holistik memiliki koneksi internet.</li> </ul>  |
| <b>Basic Flow</b>       | <p><b>{use-case dimulai}</b></p> <ol style="list-style-type: none"> <li>1. <i>Use-case</i> dimulai pada saat aktor Tenaga Medis memilih untuk melihat data pengobatan holistik.</li> <li>2. Sistem memuat data pengobatan holistik.<br/><b>{memuat data}</b></li> <li>3. Sistem menampilkan data pengobatan holistik.</li> <li>4. Aktor Tenaga Medis memilih satu data pengobatan holistik untuk ditampilkan secara detail.</li> <li>5. Sistem memuat detail data pengobatan holistik.<br/><b>{memuat data}</b></li> <li>6. Sistem memuat data modul yang tersimpan di dalam sistem.<br/><b>{memuat modul}</b></li> <li>7. Sistem menampilkan detail data pengobatan holistik dan formulir pengobatan holistik.</li> <li>8. Aktor Tenaga Medis mengisi data diagnosis.</li> <li>9. Aktor Tenaga Medis mengisi faktor risiko penyakit.</li> <li>10. Aktor Tenaga Medis mengisi faktor pemicu penyakit.</li> <li>11. Aktor Tenaga Medis mengisi lokasi intervensi yang diberikan kepada pasien.</li> <li>12. Aktor Tenaga Medis mengisi data modul yang digunakan sebagai panduan pemberian tindakan medis.</li> <li>13. Aktor Tenaga Medis mengisi data tindakan medis yang diberikan kepada pasien.<br/><b>{memberi layanan kesehatan lain}</b><br/><b>{menyimpan data pengobatan holistik}</b></li> <li>14. Aktor Tenaga Medis menyimpan data pengobatan holistik.</li> <li>15. Sistem memproses penyimpanan data pengobatan holistik.<br/><b>{memproses penyimpanan data}</b></li> <li>16. Sistem menampilkan pesan yang mengindikasikan penyimpanan data berhasil.<br/><b>{use-case selesai}</b></li> <li>17. <i>Use-case</i> selesai.</li> </ol> |
| <b>Alternative Flow</b> | <p>A1. Menangani kegagalan ketika memuat data<br/>Pada <b>{memuat data}</b>, jika sistem mengalami kegagalan ketika memuat data, maka sistem menampilkan pesan bahwa sistem mengalami kegagalan dalam memuat data, <i>use-case</i> selesai.</p> <p>A2. Menangani kondisi modul yang tidak tersedia<br/>Pada <b>{memuat modul}</b>, jika modul tidak tersedia di dalam sistem, maka:</p> <ol style="list-style-type: none"> <li>1. Sistem menampilkan pesan bahwa modul belum tersedia.</li> <li>2. Sistem mencegah aktor Tenaga Medis untuk mengisi data modul.</li> </ol>   |

**Commented [F10]:** Termasuk data:

1. Tindakan medis yang diberikan
2. Anjuran pemberhentian, persetujuan, dan izin untuk melanjutkan pengobatan yang sedang dijalani

**Commented [F11]:** Alternative flow yang sederhana dapat dijelaskan dalam bentuk paragraf. Seperti pada panduan Chapter 9, Sub bab Writing Optional, Alternative, and Exception Flows, bagian Representing Alternative Flows in Separate Sections (lihat example kedua, pada A1) (Bittner & Spence, 2002)

**Tabel 4.9 Use-Case Description Mengelola Data Pengobatan Holistik**

|                       |   |
|-----------------------|---|
|                       | <p>3. <i>Use-case</i> dilanjutkan pada langkah selanjutnya.</p> <p>A3. Mengisi data rujukan<br/>Pada {memberi layanan kesehatan lain}, jika aktor Tenaga Medis memberikan rujukan kepada pasien, maka, aktor Tenaga Medis mengisi data rujukan, <i>use-case</i> dilanjutkan pada {menyimpan data pengobatan holistik}.</p> <p>A4. Mengisi data pengantar cek darah<br/>Pada {memberi layanan kesehatan lain}, jika aktor Tenaga Medis memberikan instruksi cek darah kepada pasien, maka aktor Tenaga Medis mengisi data pengantar cek darah, <i>use-case</i> dilanjutkan pada {menyimpan data pengobatan holistik}.</p> <p>A5. Menangani kegagalan penyimpanan data<br/>Pada {memproses penyimpanan data}, jika sistem mengalami kegagalan penyimpanan data, maka sistem menampilkan pesan bahwa sistem mengalami kegagalan dalam menyimpan data, <i>use-case</i> selesai.</p> |
| <b>Post-condition</b> | <ul style="list-style-type: none"> <li>Aktor Tenaga Medis dapat melihat data pengobatan holistik.</li> <li>Data pengobatan holistik yang berhasil ditambahkan akan disimpan di dalam sistem.</li> <li>Hasil penambahan data dapat diketahui oleh aktor berdasarkan pesan yang mengindikasikan penyimpanan data berhasil.</li> </ul>   |

#### 4.2.3.2 Use-Case Description Membaca Panduan Cara Menghindari dan Mengatasi Penyakit

*Use-case description* dalam Tabel 4.10 menjelaskan kondisi dan alur kegiatan ketika aktor Tenaga Medis menggunakan sistem untuk membaca panduan cara menghindari dan mengatasi penyakit. Panduan tersebut tersimpan didalam modul penyakit, modul faktor risiko penyakit, dan modul faktor pemicu penyakit yang dapat disimpan di dalam sistem.

**Tabel 4.10 Use-Case Description Membaca Panduan Cara Menghindari dan Mengatasi Penyakit**

|                          |  |
|--------------------------|--|
| <b>Brief description</b> | <i>Use-case</i> Membaca Panduan Cara Menghindari dan Mengatasi Penyakit menjelaskan bagaimana aktor Tenaga Medis menggunakan sistem untuk membaca isi modul penyakit, modul faktor risiko penyakit, modul faktor pemicu penyakit yang tersimpan di dalam sistem.   |
| <b>Actor</b>             | Tenaga Medis   |
| <b>Pre-condition</b>     | <ul style="list-style-type: none"> <li><i>Login</i> yang dilakukan aktor Tenaga Medis dinyatakan berhasil.</li> <li><i>Session login</i> aktor sebagai Tenaga Medis atau Kepala Poliklinik masih tersimpan dalam sistem.</li> <li>Komputer yang digunakan untuk Membaca Panduan Cara Menghindari dan Mengatasi Penyakit memiliki koneksi internet.</li> <li>Modul penyakit, modul faktor risiko penyakit, dan modul faktor pemicu penyakit tersimpan di dalam sistem.</li> </ul> |

**Tabel 4.10 Use-Case Description Membaca Panduan Cara Menghindari dan Mengatasi Penyakit**

|                         |   |
|-------------------------|---|
|                         | <ul style="list-style-type: none"> <li>Aktor Tenaga Medis telah mengetahui dan mengisi penyakit yang diderita pasien, faktor risiko penyakit pasien, dan faktor pemicu penyakit pasien pada data pengobatan holistik.</li> <li>Sistem menampilkan daftar modul yang dapat dipilih oleh aktor Tenaga Medis atau aktor Kepala Poliklinik.</li> </ul>  |
| <b>Basic Flow</b>       | <p><b>{use-case dimulai}</b></p> <ol style="list-style-type: none"> <li>1. Use-case dimulai pada saat aktor Tenaga Medis memilih modul yang digunakan sebagai panduan pemberian tindakan medis.</li> <li>2. Sistem menampilkan pilihan kepada aktor Tenaga Medis untuk dapat membaca isi modul yang dipilih oleh aktor Tenaga Medis atau aktor Kepala Poliklinik.</li> <li>3. Aktor Tenaga Medis memilih untuk membaca isi modul.</li> <li>4. Sistem memuat isi modul yang dipilih oleh aktor Tenaga Medis atau aktor Kepala Poliklinik.</li> </ol> <p><b>{memuat modul}</b></p> <ol style="list-style-type: none"> <li>5. Sistem menampilkan isi modul yang dipilih oleh aktor Tenaga Medis atau aktor Kepala Poliklinik.</li> <li>6. Aktor Tenaga Medis membaca isi modul yang ditampilkan.</li> </ol> <p><b>{memilih modul lain}</b></p> <p><b>{use-case selesai}</b></p> <ol style="list-style-type: none"> <li>7. Use-case selesai.</li> </ol> |
| <b>Alternative Flow</b> | <ol style="list-style-type: none"> <li>A1. Menangani kegagalan pemuatan modul<br/>Pada <b>{memuat modul}</b>, jika sistem mengalami kegagalan dalam memuat modul, maka sistem akan menampilkan pesan kegagalan memuat modul, use-case selesai.</li> <li>A2. Memilih modul lain ketika terdapat modul sedang ditampilkan<br/>Pada <b>{memilih modul lain}</b>, jika aktor Tenaga Medis ingin memilih modul lain ketika terdapat modul yang sedang ditampilkan, maka: <ol style="list-style-type: none"> <li>1. Aktor Tenaga Medis menutup modul yang sedang ditampilkan</li> <li>2. Sistem melakukan proses penutupan modul.</li> <li>3. Use-case dilanjutkan pada <b>{use-case dimulai}</b>.</li> </ol> </li> </ol>   |
| <b>Post-condition</b>   | <ul style="list-style-type: none"> <li>Sistem menampilkan isi modul yang dipilih oleh aktor Tenaga Medis atau aktor Kepala Poliklinik.</li> <li>Aktor Tenaga Medis dapat membaca isi modul yang berisi panduan cara menghindari dan mengatasi penyakit pasien.</li> </ul>   |

#### 4.2.3.3 Use-Case Description Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien

Use-case ini menjelaskan kondisi dan alur kegiatan ketika aktor Tenaga Medis menggunakan sistem untuk mengetahui hasil penghitungan tingkat risiko penyakit dan tingkat stres pasien berdasarkan data dasar kesehatan keluarga yang sudah diisi oleh pasien. Aktor Tenaga Medis dapat menggunakan sistem untuk menghitung ulang tingkat risiko penyakit dan tingkat stres pasien, alur kegiatan tersebut juga dijelaskan dalam Tabel 4.11.

**Tabel 4.11 Use-Case Description Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien**

|                          |  |
|--------------------------|--|
| <b>Brief description</b> | Use-case Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien menjelaskan bagaimana aktor Tenaga Medis menggunakan sistem untuk menghitung dan mengetahui hasil perhitungan tingkat risiko penyakit dan tingkat stres pasien. Tingkat risiko penyakit pasien dalam satu keluarga adalah sama, sedangkan tingkat stres pasien hanya berlaku bagi pasien yang bertindak sebagai kepala keluarga.  |
| <b>Actor</b>             | Tenaga Medis   |
| <b>Pre-condition</b>     | <ul style="list-style-type: none"> <li>• Login yang dilakukan aktor Tenaga Medis dinyatakan berhasil.</li> <li>• Session login aktor sebagai Tenaga Medis atau Kepala Poliklinik masih tersimpan dalam sistem.</li> <li>• Komputer yang digunakan untuk Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien memiliki koneksi internet.</li> <li>• Sistem telah menyimpan data dasar kesehatan keluarga yang telah diisi oleh pasien BPJS Poliklinik Pabrik Gula Kebon Agung.</li> </ul>  |
| <b>Basic Flow</b>        | <p><b>{use-case dimulai}</b></p> <ol style="list-style-type: none"> <li>1. Use-case dimulai pada saat aktor Tenaga Medis memilih untuk melihat data dasar kesehatan keluarga.</li> <li>2. Sistem memuat data dasar kesehatan keluarga.</li> </ol> <p><b>{memuat data}</b></p> <ol style="list-style-type: none"> <li>3. Sistem menampilkan data dasar kesehatan keluarga.</li> <li>4. Aktor Tenaga Medis memilih salah satu data dasar kesehatan keluarga untuk ditampilkan secara detail.</li> <li>5. Sistem memuat detail data dasar kesehatan keluarga.</li> </ol> <p><b>{memuat data}</b></p> <p><b>{menampilkan detail data dasar kesehatan keluarga}</b></p> <ol style="list-style-type: none"> <li>6. Sistem menampilkan detail data dasar kesehatan keluarga.</li> <li>7. Sistem menampilkan hasil perhitungan tingkat risiko penyakit dan tingkat stres pasien.</li> </ol> <p><b>{melakukan penghitungan ulang}</b></p> <p><b>{use-case selesai}</b></p> <ol style="list-style-type: none"> <li>8. Use-case selesai.</li> </ol> |
| <b>Alternative Flow</b>  | <p>A1. Menangani kegagalan memuat data<br/>Pada <b>{memuat data}</b>, jika sistem mengalami kegagalan ketika memuat data, maka sistem menampilkan pesan bahwa sistem mengalami kegagalan memuat data, use-case selesai.</p> <p>A2. Melakukan perhitungan ulang tingkat risiko penyakit dan tingkat stres pasien<br/>Pada <b>{melakukan penghitungan ulang}</b>, ketika aktor Tenaga Medis ingin mengetahui hasil penghitungan ulang tingkat risiko penyakit dan tingkat stres pasien, maka:</p> <ol style="list-style-type: none"> <li>1. Aktor Tenaga Medis meminta hasil penghitungan ulang tingkat risiko penyakit dan tingkat stres pasien.</li> <li>2. Sistem melakukan penghitungan ulang tingkat risiko penyakit dan tingkat stres pasien.</li> </ol> <p><b>{penghitungan gagal}</b></p>  |

|                       |  |
|-----------------------|--|
|                       | <p>3. <i>Use-case</i> dilanjutkan pada <b>{menampilkan hasil penghitungan}</b>.</p> <p>A3. Menangani kegagalan penghitungan ulang tingkat risiko penyakit dan tingkat stres pasien.<br/> Pada <b>{penghitungan gagal}</b>, jika sistem mengalami kegagalan penghitungan ulang tingkat risiko penyakit dan tingkat stres pasien, maka sistem menampilkan pesan bahwa sistem mengalami kegagalan penghitungan ulang, <i>use-case</i> dilanjutkan pada <b>{menampilkan detail data dasar kesehatan keluarga}</b>.</p> |
| <b>Post-condition</b> | <ul style="list-style-type: none"> <li>Sistem menampilkan hasil penghitungan tingkat risiko dan tingkat stres pasien.</li> <li>Aktor Tenaga Medis mengetahui hasil penghitungan tingkat risiko dan tingkat stres pasien.</li> </ul>  |

#### 4.2.3.4 Use-Case Description Mencatat Data Obat Keluar

**Tabel 4.12 Use-Case Description Mencatat Data Obat Keluar**

|                          |   |
|--------------------------|---|
| <b>Brief description</b> | <i>Use-case</i> Mencatat Data Obat Keluar menjelaskan bagaimana aktor Tenaga Kefarmasian menggunakan sistem untuk mencatat data obat keluar sesuai dengan resep obat yang diberikan dokter kepada pasien.   |
| <b>Actor</b>             | Tenaga Kefarmasian  |
| <b>Pre-condition</b>     | <ul style="list-style-type: none"> <li><i>Login</i> yang dilakukan aktor Tenaga Kefarmasian dinyatakan berhasil.</li> <li><i>Session login</i> aktor sebagai Tenaga Kefarmasian masih tersimpan di dalam sistem.</li> <li>Komputer yang digunakan untuk Mencatat Data Obat Keluar memiliki koneksi internet.</li> <li>Data obat tersimpan di dalam sistem.</li> <li>Aktor Tenaga Kefarmasian telah menerima resep obat dari pasien.</li> </ul>  |
| <b>Basic Flow</b>        | <p><b>{use-case dimulai}</b></p> <ol style="list-style-type: none"> <li><i>Use-case</i> dimulai pada saat aktor Tenaga Kefarmasian mencari informasi ketersediaan obat.</li> <li>Sistem mengecek ketersediaan obat dari data obat yang tersimpan di dalam sistem.</li> </ol> <p><b>{mengecek ketersediaan obat}</b></p> <ol style="list-style-type: none"> <li>Sistem mengecek status pembiayaan obat dari data obat yang tersimpan di dalam sistem.</li> </ol> <p><b>{mengecek status obat}</b></p> <ol style="list-style-type: none"> <li>Sistem menampilkan data obat dan status pembiayaan obat.</li> <li>Memilih obat yang akan diberikan kepada pasien.</li> <li>Mengisi dosis obat.</li> </ol> <p><b>{mengecek obat lain sesuai dengan resep}</b></p> <ol style="list-style-type: none"> <li>Memilih untuk menyimpan data obat.</li> <li>Sistem memproses penyimpanan data obat keluar.</li> </ol> <p><b>{menyimpan data obat}</b></p> <ol style="list-style-type: none"> <li>Sistem menampilkan pesan bahwa penyimpanan data obat keluar berhasil.</li> </ol> <p><b>{use-case selesai}</b></p> <ol style="list-style-type: none"> <li><i>Use-case</i> selesai.</li> </ol> |

**Tabel 4.12 Use-Case Description Mencatat Data Obat Keluar**

|                         |  |
|-------------------------|--|
| <b>Alternative Flow</b> | <p>A1. Menangani kegagalan pengecekan data obat<br/>Pada {mengecek ketersediaan obat} dan {mengecek status obat} dan {mengecek ketersediaan obat alternatif} ketika sistem mengalami kegagalan dalam proses pengecekan data obat, maka sistem menampilkan pesan bahwa proses pengecekan gagal, <i>use-case</i> selesai.</p> <p>A2. Menampilkan obat alternatif<br/>Pada {mengecek ketersediaan obat}, jika sistem menyatakan bahwa obat tidak tersedia, maka:</p> <ol style="list-style-type: none"> <li>1. Sistem mengecek ketersediaan obat alternatif. {mengecek ketersediaan obat alternatif}</li> <li>2. Sistem menampilkan data obat alternatif yang dapat diberikan kepada pasien.</li> <li>3. Use-case dilanjutkan pada {mengecek status obat}.</li> </ol> <p>A3. Menangani data obat alternatif yang tidak tersedia<br/>Pada {mengecek ketersediaan obat alternatif}, jika sistem menyatakan obat alternatif tidak tersedia, maka:</p> <ol style="list-style-type: none"> <li>1. Sistem menampilkan pesan bahwa obat alternatif tidak tersedia.</li> <li>2. Use-case dilanjutkan pada {mengecek obat lain sesuai dengan resep}.</li> </ol> <p>A4. Menangani kegagalan proses penyimpanan data obat<br/>Pada {menyimpan data obat}, jika sistem mengalami kegagalan dalam proses penyimpanan data obat, maka sistem menampilkan pesan bahwa proses penyimpanan gagal, <i>use-case</i> selesai.</p> <p>A5. Mengecek ketersediaan obat lain sesuai resep<br/>Pada {mengecek obat lain sesuai dengan resep}, jika aktor Tenaga Kefarmasian ingin mengecek obat lain sesuai dengan yang tertera pada resep obat, maka <i>use-case</i> dapat dimulai lagi pada {<i>use-case</i> dimulai}.</p> |
| <b>Post-condition</b>   | <ul style="list-style-type: none"> <li>• Aktor Tenaga Kefarmasian mengetahui ketersediaan obat yang akan diberikan kepada pasien.</li> <li>• Aktor Tenaga Kefarmasian mengetahui status pembiayaan obat yang akan diberikan kepada pasien.</li> <li>• Data obat keluar sesuai dengan resep dokter tersimpan di dalam sistem.</li> <li>• Aktor Tenaga Kefarmasian dapat memberikan obat sesuai resep obat kepada pasien.</li> </ul>   |

### 4.3 Activity Model

Salah satu cara memvisualisasi *flow of events* yang dimiliki oleh *use-case* adalah dengan menggunakan UML activity diagram (Bittner & Spence, 2002). Bittner dan Spence menambahkan, teknik visualisasi ini memberikan kemudahan kepada *stakeholder* untuk memahami *flow of events* (*basic flows* dan *alternative flows*), sehingga *stakeholder* dapat memahami cakupan *use-case*. Activity diagram

**Commented [F12]:** Sesuai dengan panduan pada Chapter 7, sub bab The Nature of the Flow of Events, bagian Visualizing the Flow of Events (Bittner & Spence, 2002)

dalam penelitian ini dibuat berdasarkan *flow of events* dari masing-masing *use-case* pada sub-bab sebelumnya.

#### **4.3.1 Activity Diagram Mengelola Data Pengobatan Holistik**

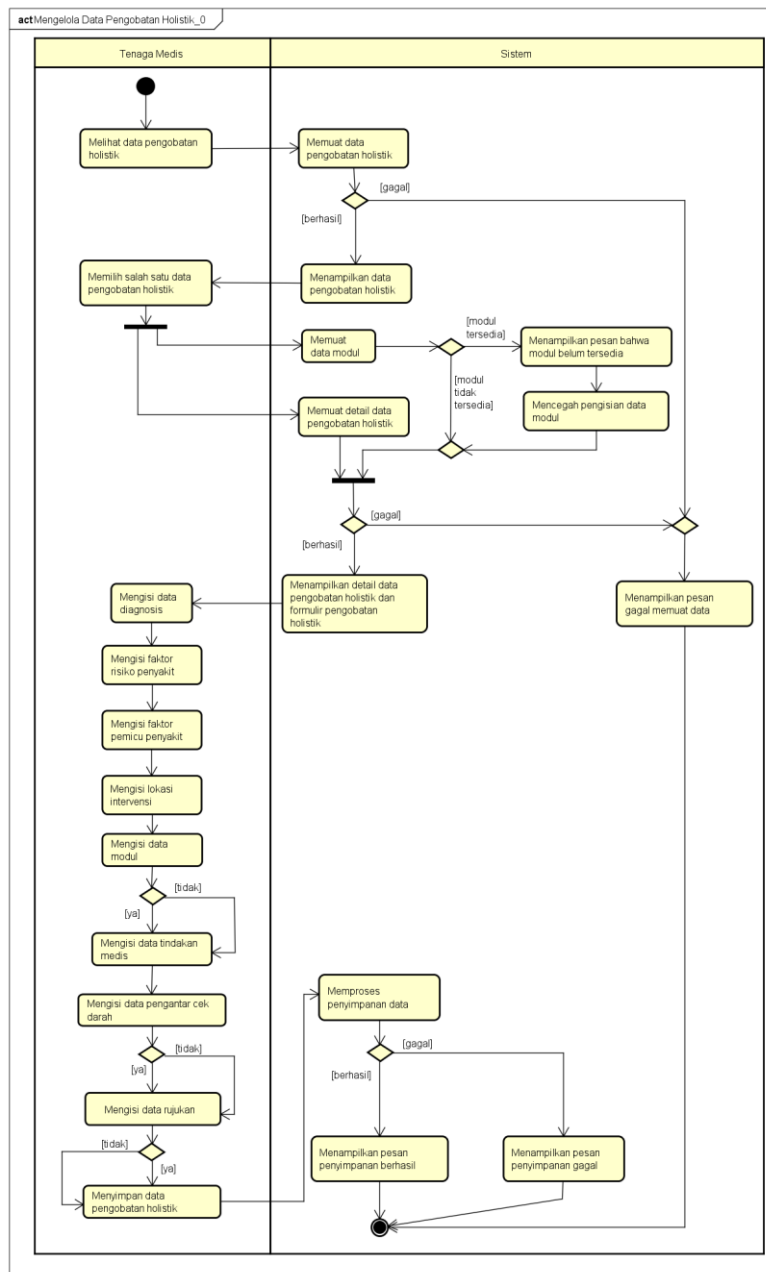
*Activity diagram* dalam gambar Gambar 4.5 menggambarkan *flow of events* dari *use-case* Mengelola Data Pengobatan Holistik yang dijelaskan dalam *use-case description* Tabel 4.9. Aktivitas dimulai pada saat aktor Tenaga Medis melihat data pengobatan holistik melalui sistem, kemudian sistem akan menampilkan seluruh data pengobatan holistik yang tersimpan kepada aktor Tenaga Medis. Aktor Tenaga Medis dapat melihat detail data pengobatan holistik, detail data tersebut, dan formulir pengobatan holistik akan ditampilkan oleh sistem. Kemudian aktivitas aktor Tenaga Medis dilanjutkan untuk mengisi data pada formulir pengobatan holistik dan memilih untuk menyimpan formulir pengobatan holistik yang telah diisi. Sistem akan memproses penyimpanan data dan akan menampilkan pesan kepada aktor Tenaga Medis apabila proses penyimpanan berhasil atau gagal.

#### **4.3.2 Activity Diagram Membaca Panduan Cara Menghindari dan Mengatasi Penyakit**

*Activity diagram* dalam Gambar 4.6 menggambarkan *flow of events* dari *use-case* Membaca Panduan Cara Menghindari dan Mengatasi Penyakit yang dijelaskan dalam *use-case description* pada Tabel 4.10. Aktor Tenaga Medis memulai aktivitas dengan memilih modul yang diinginkan, kemudian sistem menampilkan pilihan agar aktor Tenaga Medis dapat membaca isi modul. Setelah aktor Tenaga Medis memilih untuk membaca modul tertentu, sistem akan melakukan proses pemuatan isi modul dan akan menampilkan isi modul kepada aktor Tenaga Medis jika proses pemuatan berhasil. Selain itu, aktor Tenaga Medis juga dapat melakukan aktivitas untuk melihat isi modul lain yang tersimpan di dalam sistem.

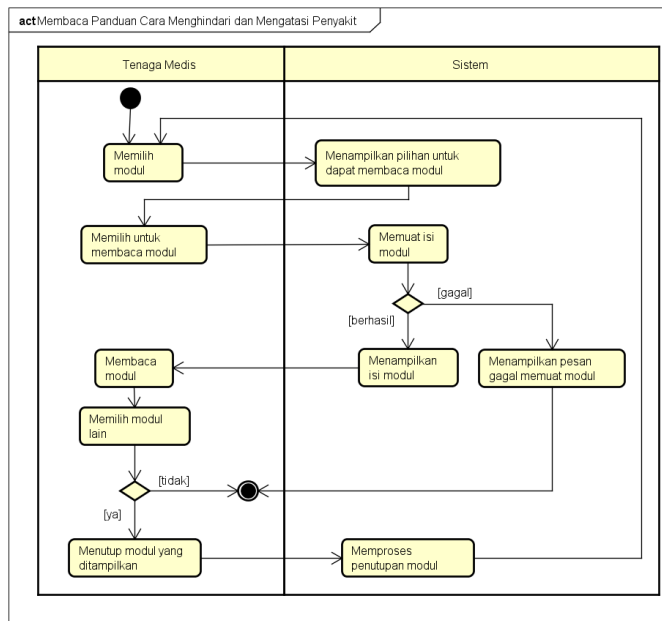
#### **4.3.3 Activity Diagram Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien**

*Activity diagram* dalam Gambar 4.7 menggambarkan *flow of events* dari *use-case* Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien yang dijelaskan dalam *use-case description* pada Tabel 4.11. Aktivitas menghitung tingkat risiko penyakit dan tingkat stres pasien dimulai ketika aktor Tenaga Medis melihat data dasar kesehatan keluarga, kemudian sistem akan menampilkan seluruh data dasar kesehatan keluarga yang tersimpan di dalam sistem. Aktor Tenaga Medis melakukan aktivitas untuk dapat melihat detail salah satu data dasar kesehatan keluarga. Sistem akan memproses pemuatan detail data dasar kesehatan keluarga dan menampilkan hasil perhitungan tingkat risiko penyakit dan tingkat stres pasien. Aktor Tenaga Medis juga dapat melakukan penghitungan ulang tingkat risiko penyakit dan tingkat stres pasien, kemudian sistem akan memproses penghitungan ulang tersebut dan menampilkannya kepada aktor Tenaga Medis.

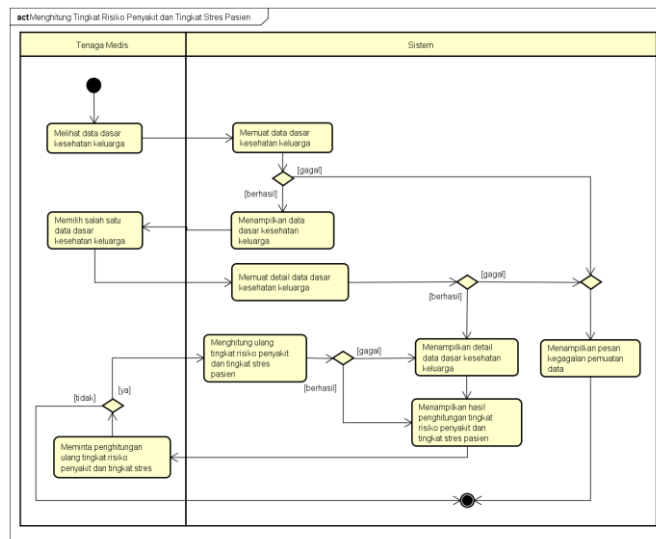


**Gambar 4.5 Activity Diagram Mengelola Data Pengobatan Holistik**





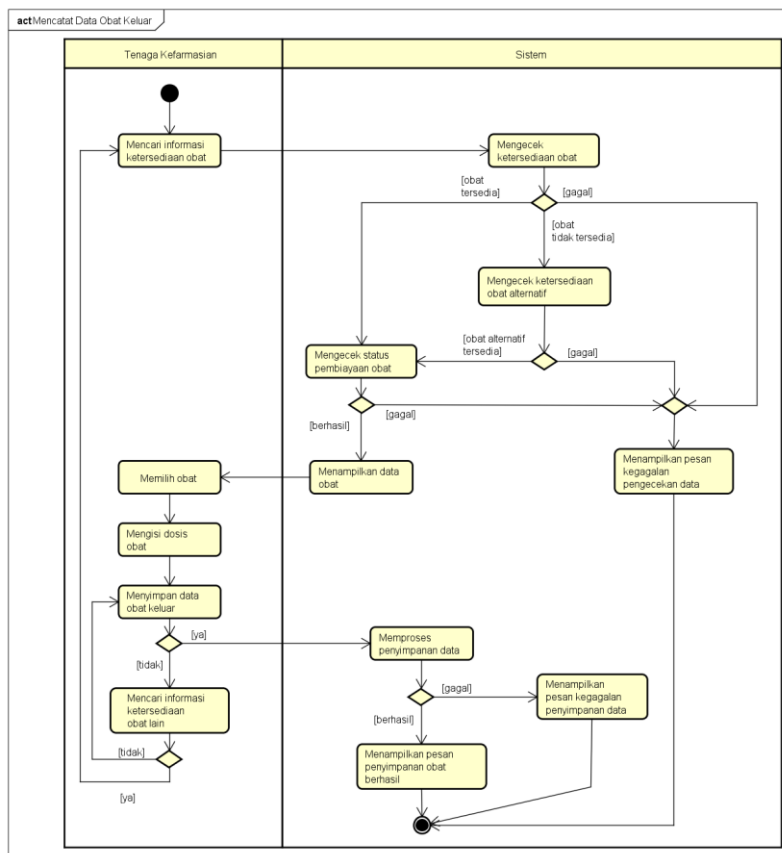
**Gambar 4.6 Activity Diagram Membaca Panduan Cara Menghindari dan Mengatasi Penyakit**



**Gambar 4.7 Activity Diagram Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien**

#### 4.3.4 Activity Diagram Mencatat Data Obat Keluar

Activity diagram dalam Gambar 4.8 menggambarkan *flow of events* dari *use-case* Mencatat Data Obat Keluar yang dijelaskan dalam *use-case description* pada Tabel 4.12. Aktor Tenaga Kefarmasian memulai aktivitas dengan mencari informasi ketersediaan obat. Kemudian, sistem akan mengecek ketersediaan obat, sistem juga mengecek ketersediaan obat alternatif, jika obat yang diinginkan aktor Tenaga Kefarmasian tidak tersedia. Selain itu, sistem juga mengecek status pembiayaan obat, setelah itu, sistem menampilkan hasil pengecekan ketersediaan dan status pembiayaan obat kepada aktor Tenaga Kefarmasian.



Gambar 4.8 Activity Diagram Mencatat Data Obat Keluar

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

### 5.1 Perancangan Sequence Diagram

Bagian ini menjelaskan interaksi antar objek, dimulai dari interaksi aktor dengan objek sistem. Selain itu, interaksi antar objek didalam lingkungan sistem juga digambarkan dalam *sequence diagram*. Seluruh interaksi yang tergambar dalam *sequence diagram* pada penelitian ini akan merepresentasikan *behavior* sistem yang akan dibangun penelitian ini.

**Commented [F13]:** Penggunaan "references" pada sequence diagram berdasarkan panduan pada (Rumbaugh, et al., 2004) Chapter 9 – Sequence Diagram – Structured Control Construct

#### 5.1.1 Sequence Diagram Mengelola Data Pengobatan Holistik

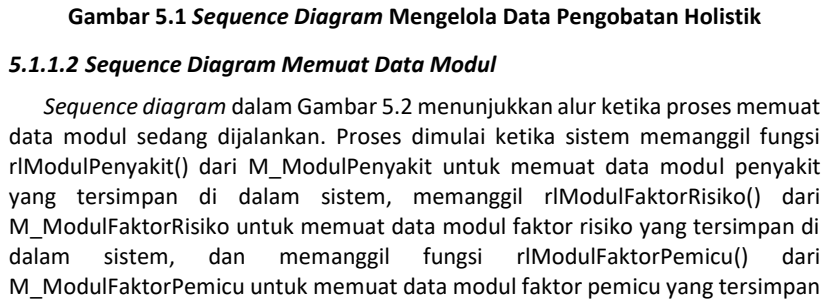
*Sequence diagram* pada bagian ini menjelaskan interaksi antar objek dalam kegiatan aktor untuk mengelola data pengobatan holistik. Sequence diagram dibagi dalam 4 gambar yang berbeda untuk menyederhanakan penggambaran kompleksitas interaksi objek dalam aktivitas mengelola data pengobatan holistik. Objek yang terlibat dalam interaksi mengelola data pengobatan holistik diantaranya adalah aktor Tenaga Medis, *boundary* V\_Menu.php, *boundary* Window, *boundary* V\_ListStatusPasien, *boundary* V\_DetailStatusPasien, *controller* C\_Menu, *controller* C\_ListStatusPasien, dan *model* M\_DiagnosisHolistik.

##### 5.1.1.1 Sequence Diagram Mengelola Data Pengobatan Holistik

Interaksi dimulai ketika aktor melihat data pengobatan holistik melalui sistem. Kemudian sistem akan memanggil fungsi *viPengobatanHolistik()* dari *C\_Menu* untuk menampilkan data pengobatan holistik yang tersimpan di dalam sistem. Kemudian sistem memanggil fungsi *rlStatusPasien()* dari *M\_DiagnosisHolistik* dengan menyertakan parameter *sort* dan *order* untuk memuat data pengobatan holistik. Jika fungsi *rlStatusPasien()* gagal dijalankan, maka sistem akan memanggil fungsi *alert()* dengan menyertakan parameter *message* yang memuat nilai pesan kegagalan pemuatan data, fungsi *alert()* digunakan untuk menampilkan pesan kegagalan tersebut kepada aktor Tenaga Medis.

Jika fungsi *rlStatusPasien()* berhasil dijalankan, maka sistem menjalankan fungsi *view()* dari *C\_Menu* untuk menampilkan data yang didapatkan dari fungsi *rlStatusPasien()* kepada aktor Tenaga Medis melalui *V\_ListStatusPasien.php*. Aktor dapat memilih salah satu data dan selanjutnya sistem menjalankan fungsi *rdStatusPasien()* dari *C\_ListStatusPasien*. Kemudian sistem akan menjalankan fungsi untuk memuat data modul dan memuat detail data pengobatan holistik. Setelah kedua fungsi tersebut dijalankan. Aktor Tenaga Medis dapat melakukan pengisian formulir pengobatan holistik.

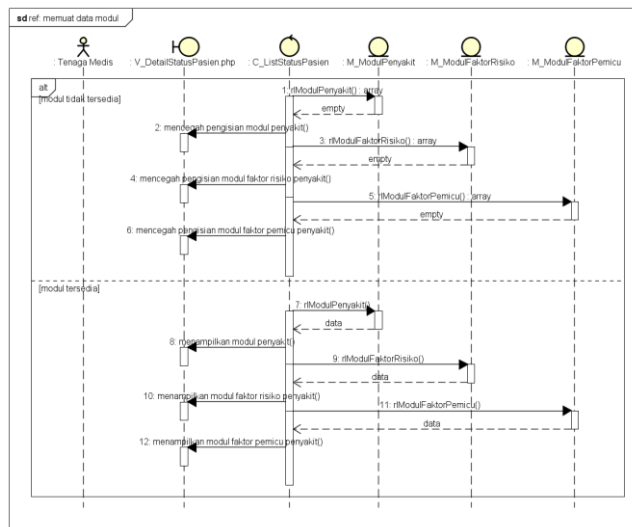
Aktor Tenaga Medis melanjutkan proses pengisian data modul melalui sistem, jika terdapat modul yang tersimpan di dalam sistem. Opsi lain yang dapat dilakukan oleh aktor Tenaga Medis diantaranya untuk melakukan proses mengisi data pengantar cek darah atau mengisi data rujukan atau mengisi data tindakan medis, kemudian dilanjutkan dengan memanggil fungsi untuk menyimpan data pengobatan holistik. Proses dilanjutkan dengan sistem menjalankan fungsi untuk



#### 5.1.1.2 Sequence Diagram Memuat Data Modul

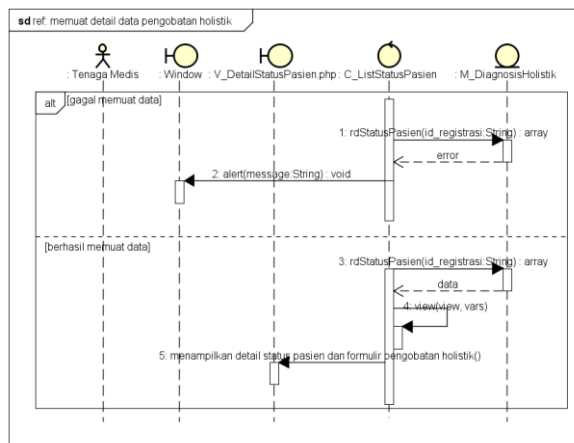
60

di dalam sistem. Jika masing-masing fungsi yang dipanggil tersebut gagal, maka sistem akan mencegah pengisian masing-masing modul melalui V\_DetailStatusPasien.php. Jika masing-masing fungsi yang dipanggil berhasil, maka sistem akan menampilkan masing-masing modul kepada aktor Tenaga Medis melalui V\_DetailStatusPasien.php.



**Gambar 5.2 Sequence Diagram Memuat Data Modul**

### 5.1.1.3 Sequence Diagram Memuat Detail Data Pengobatan Holistik

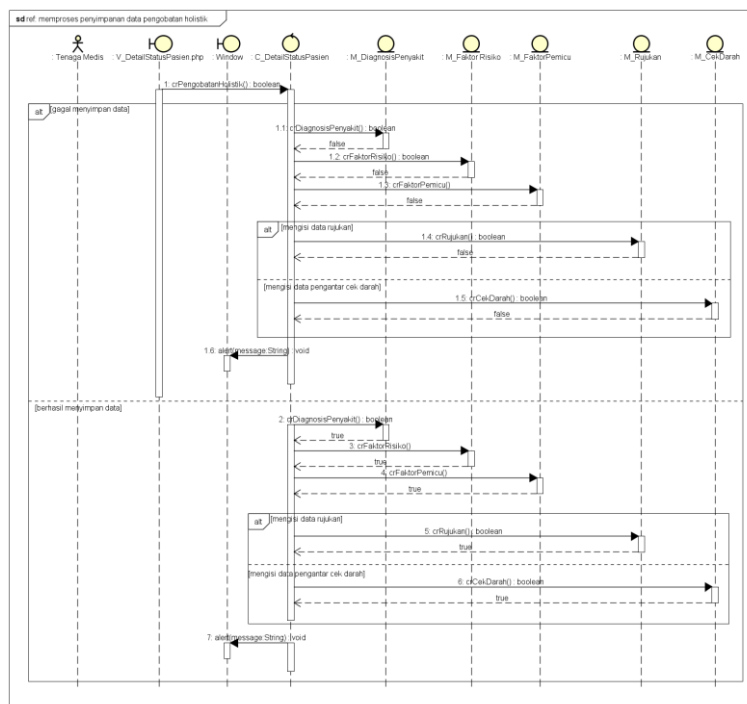


**Gambar 5.3 Sequence Diagram Memuat Detail Data Pengobatan Holistik**

Gambar 5.3 menunjukkan *sequence diagram* proses memuat detail data pengobatan holistik. Proses dimulai pada saat sistem memanggil fungsi

rdStatusPasien() dari M\_DiagnosisHolistik untuk memuat detail data diagnosis holistik. Jika fungsi rdStatusPasien() gagal dijalankan, maka sistem akan memanggil fungsi alert() dari Window dengan menyertakan parameter *message* untuk menampilkan pesan kegagalan proses memuat detail data diagnosis holistik. Jika fungsi rdStatusPasien() yang dipanggil berhasil dijalankan, maka sistem memanggil fungsi view() dari C\_ListStatusPasien untuk menampilkan detail status pasien dan formulir pengobatan holistik.

#### 5.1.1.4 Sequence Diagram Memproses Penyimpanan Data Pengobatan Holistik



**Gambar 5.4 Sequence Diagram Memproses Penyimpanan Data Pengobatan Holistik**

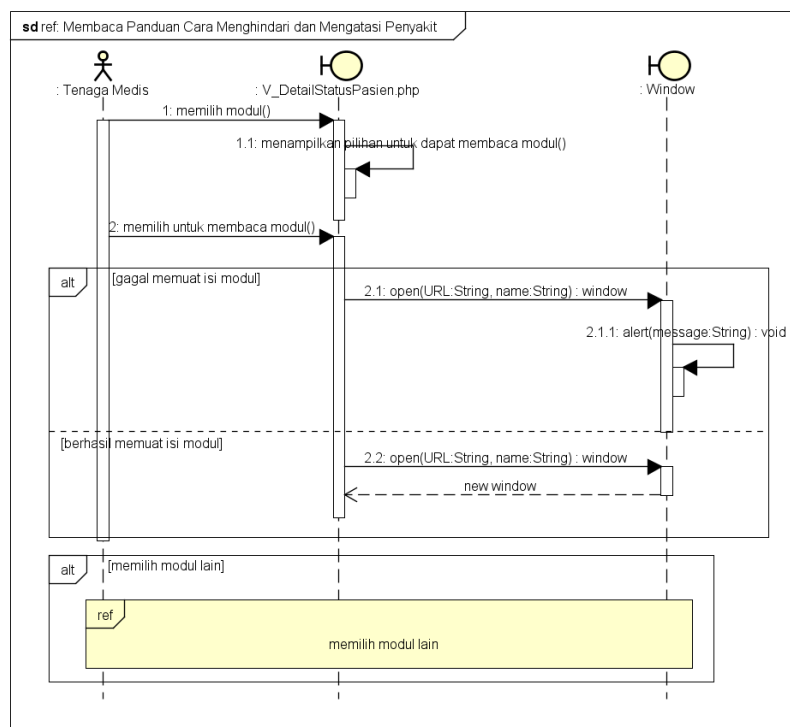
*Sequence diagram* dalam Gambar 5.4 menunjukkan proses ketika sistem memproses penyimpanan data pengobatan holistik. Proses dimulai ketika sistem memanggil fungsi `crPengobatanHolistik()` dari `C_DetailStatusPasien` untuk menyimpan data pengobatan holistik. Kemudian proses dilanjutkan dengan sistem yang memanggil fungsi `crDiagnosisPenyakit()` dari `M_DiagnosisHolistik`, `crFaktorRisiko()` dari `M_FaktorRisiko`, `crFaktorPemicu()` dari `M_FaktorPemicu` untuk menyimpan data pengobatan holistik.

Fungsi `crRujukan()` dari `M_Rujukan` akan dijalankan jika terdapat proses pengisian data rujukan, selain itu fungsi `crCekDarah()` dari `M_CekDarah` akan

dijalankan jika terdapat proses pengisian data pengantar cek darah. Jika masing-masing fungsi yang dijalankan tersebut gagal, maka sistem akan menjalankan fungsi alert() dengan parameter *message* yang berisi pesan kegagalan dalam menjalankan fungsi-fungsi yang telah dipanggil kepada aktor Tenaga Medis melalui objek Window. Jika fungsi-fungsi yang dijalankan berhasil, maka sistem akan menjalankan fungsi alert() dengan parameter *message* yang berisi pesan bahwa fungsi-fungsi berhasil dijalankan.

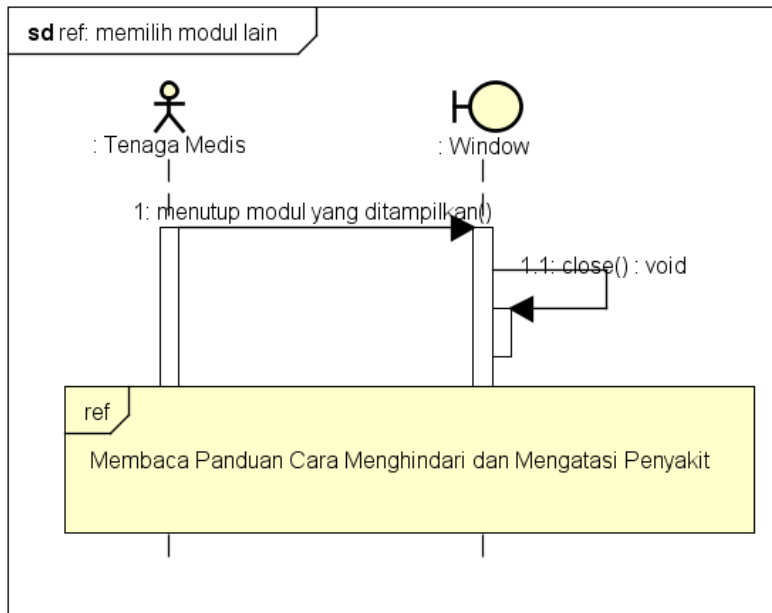
### 5.1.2 Sequence Diagram Membaca Panduan Cara Menghindari dan Mengatasi Penyakit

#### 5.1.2.1 Sequence Diagram Membaca Panduan Cara Menghindari dan Mengatasi Penyakit



Gambar 5.5 Sequence Diagram Membaca Panduan Cara Menghindari dan Mengatasi Penyakit

### 5.1.2.2 Sequence Diagram Memilih Modul Lain

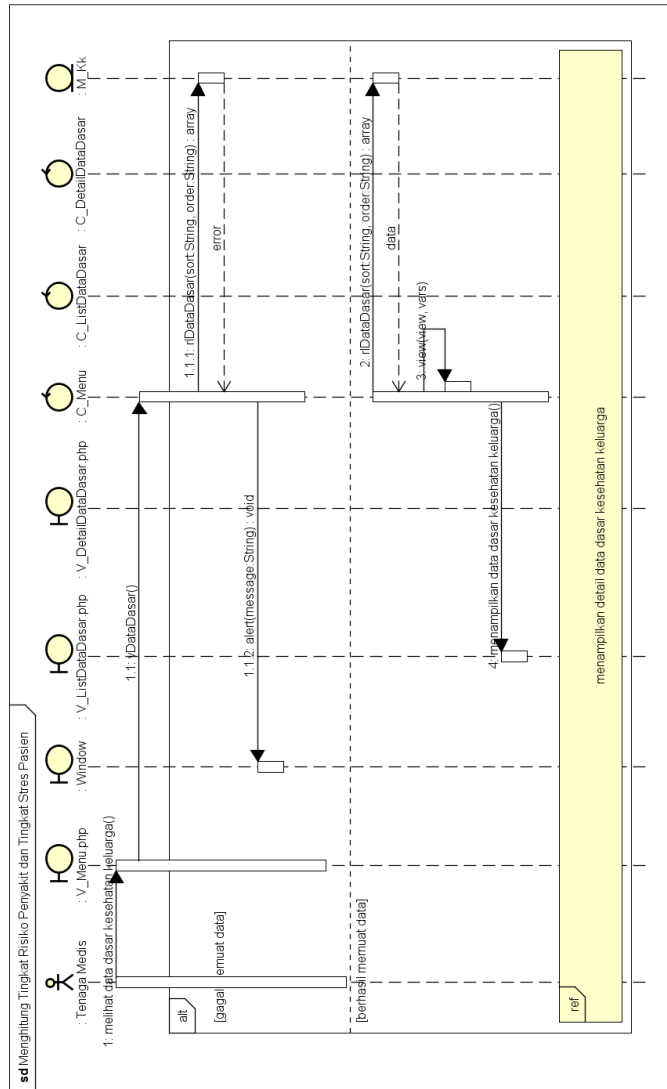


Gambar 5.6 Sequence Diagram Memilih Modul Lain



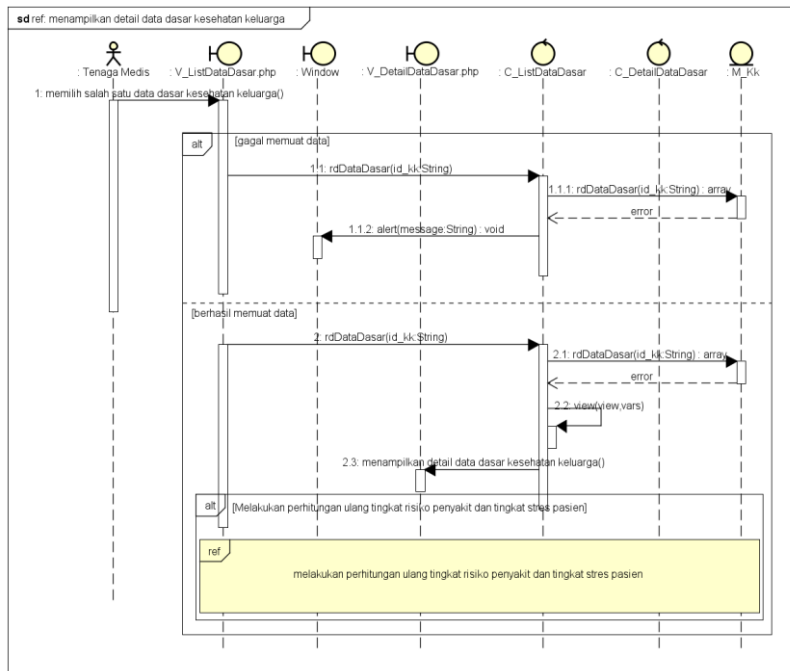
### 5.1.3 Sequence Diagram Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien

#### 5.1.3.1 Sequence Diagram Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien



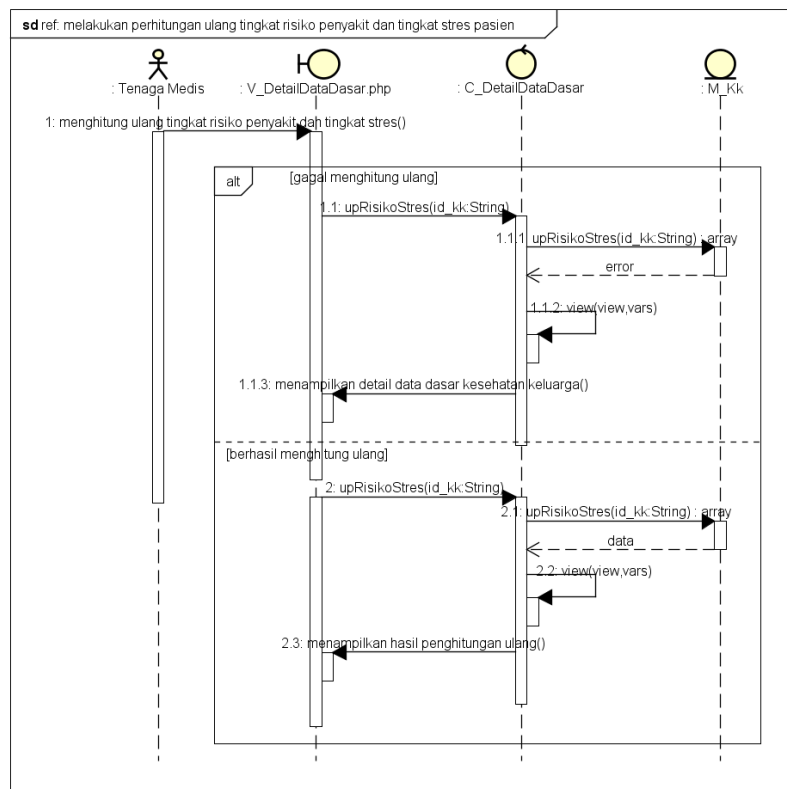
**Gambar 5.7** Sequence Diagram Menghitung Tingkat Risiko Penyakit dan Tingkat Stres Pasien

### 5.1.3.2 Sequence Diagram Menampilkan Detail Data Dasar Kesehatan Keluarga



Gambar 5.8 Sequence Diagram Menampilkan Detail Data Dasar Kesehatan Keluarga

### 5.1.3.3 Sequence Diagram Melakukan Perhitungan Ulang Tingkat Risiko Penyakit dan Tingkat Stres Pasien



Gambar 5.9 Sequence Diagram Melakukan Perhitungan Ulang Tingkat Risiko Penyakit dan Tingkat Stres Pasien

## 5.2 Perancangan Class Diagram

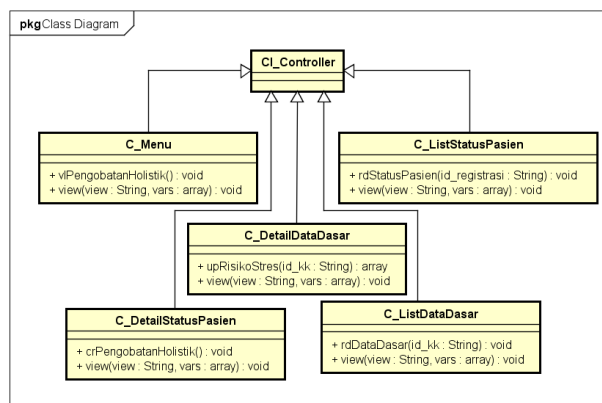
Bagian ini menjelaskan pemodelan hasil identifikasi obyek-obyek yang ada di dalam sistem kedalam bentuk *class diagram*. *Class diagram* akan menampilkan atribut dan operasi yang dimiliki oleh masing-masing obyek, selain itu *class diagram* juga menunjukkan hubungan antar obyek di dalam sistem. Obyek di dalam *class diagram* mencakup obyek-obyek didalam sistem yang saling berinteraksi yang tergambar dalam beberapa *sequence diagram* pada sub-bab sebelumnya.

### 5.2.1 Perancangan Analysis Class Diagram

*Analysis class diagram* dalam Gambar 5.12 menunjukkan pemodelan obyek-obyek yang ada di dalam sistem, *analysis class diagram* akan mendasari pemodelan data dalam penelitian ini.

### 5.2.2 Perancangan Controller Class Diagram

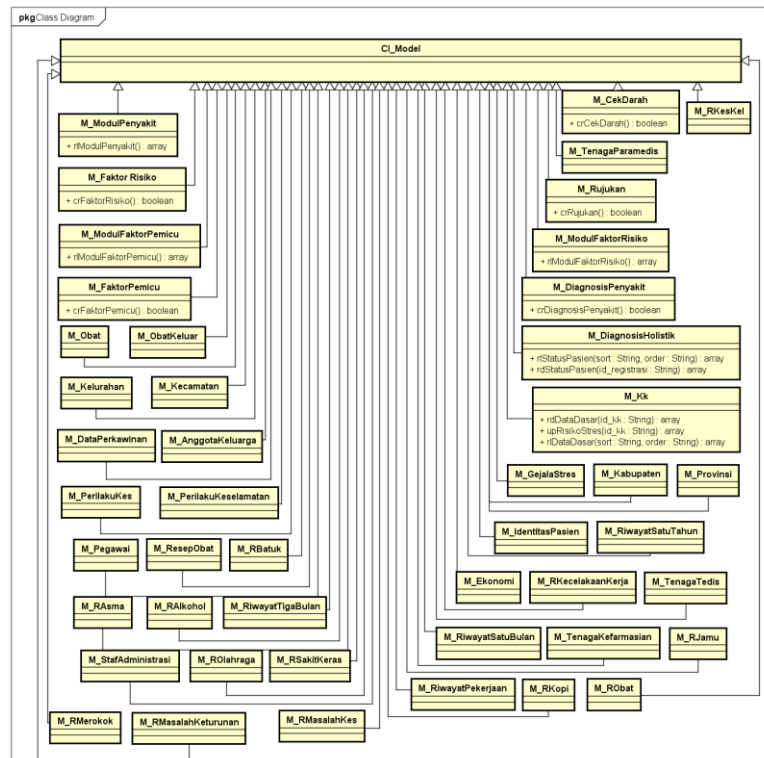
Bagian ini menjelaskan hasil pemodelan obyek dalam bentuk *class diagram* yang disesuaikan dengan penggunaan *framework* MVC CodeIgniter. Dalam Gambar 5.10 menunjukkan bahwa kelas C\_Menu, C\_ListStatusPasien, C\_DetailDataDasar, C\_DetailStatusPasien, dan C\_ListDataDasar merupakan generalisasi dari kelas CI\_Controller.



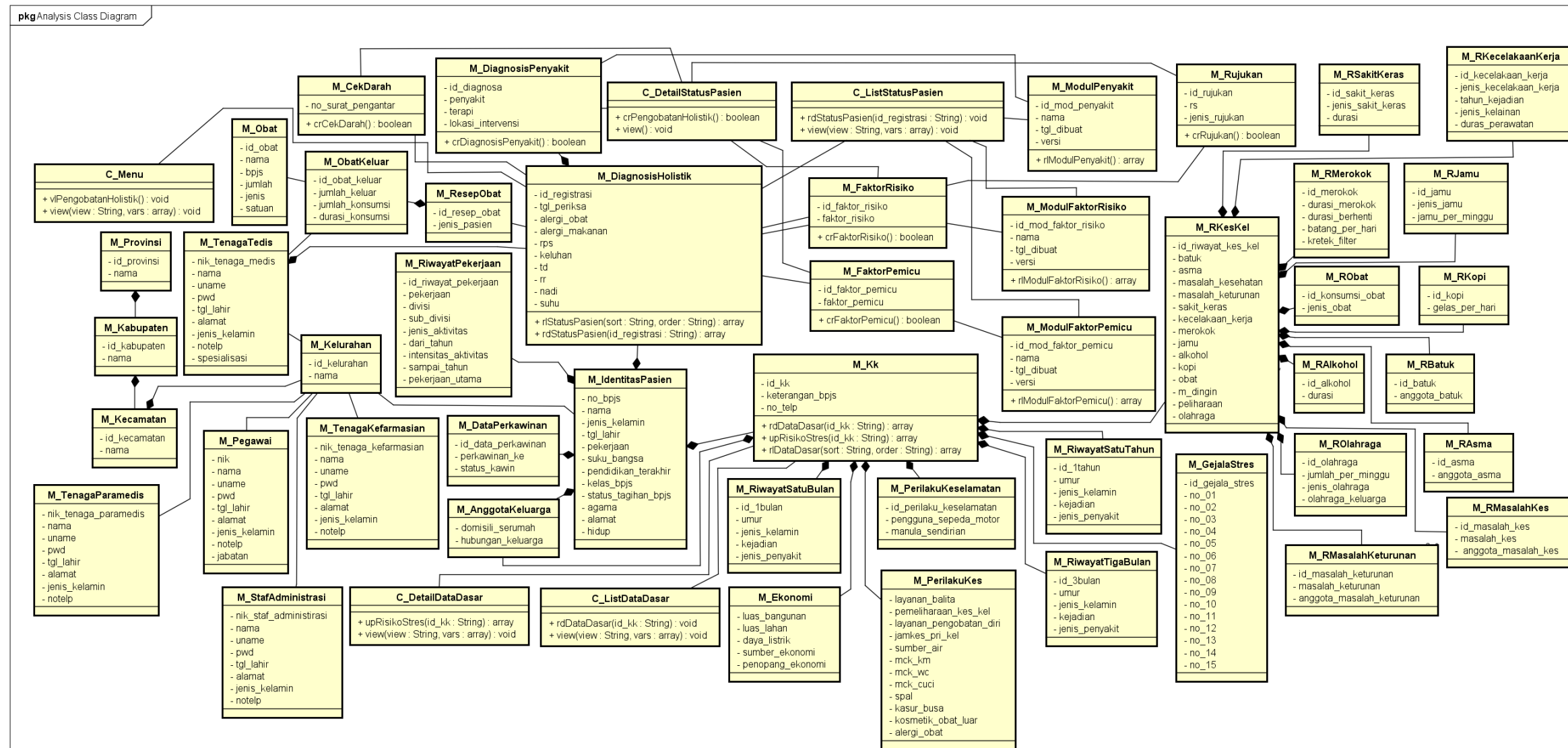
Gambar 5.10 Controller Class Diagram

### 5.2.3 Perancangan Model Class Diagram

Gambar 5.11 menunjukkan hasil pemodelan obyek dalam bentuk class diagram yang disesuaikan dengan penggunaan *framework* MVC CodeIgniter. Gambar Class diagram menunjukkan bahwa kelas M\_AnggotaKeluarga, M\_CekDarah, M\_DataPerkawinan, M\_DiagnosisHolistik, M\_DiagnosisPenyakit, M\_Ekonomi, M\_Faktor Risiko, M\_FaktorPemicu, M\_GejalaStres, M\_IdentitasPasien, M\_Kabupaten, M\_Kecamatan, M\_Kelurahan, M\_ModulFaktorPemicu, M\_ModulFaktorRisiko, M\_ModulPenyakit, M\_Obat, M\_ObatKeluar, M\_Pegawai, M\_PerilakuKes, M\_PerilakuKeselamatan, M\_Provinsi, M\_RAlkohol, M\_RAsma, M\_RBatuk, M\_ResepObat, M\_RiwayatPekerjaan, M\_RiwayatSatuBulan, M\_RiwayatSatuTahun, M\_RiwayatTigaBulan, M\_RJamu, M\_RKecelakaanKerja, M\_RKesKel, M\_RKopi, M\_RMasalahKes, M\_RMasalahKeturunan, M\_RMerokok, M\_RObat, M\_ROlahraga, M\_RSakitKeras, M\_Rujukan, M\_StafAdministrasi, M\_TenagaKefarmasian, M\_TenagaParamedis, M\_TenagaTedis merupakan generalisasi dari kelas CI\_Model.



Gambar 5.11 Model Class Diagram



### Gambar 5.12 Analysis Class Diagram







## DAFTAR PUSTAKA

- Aggarwal, V., 2002. The Application of the Unified Modeling Language in Object-Oriented Analysis of Healthcare Information Systems. *Journal of Medical Systems*, 26(5), pp. 383-397.
- Bittner, K. & Spence, I., 2002. *Use Case Modeling*. Boston: Addison Wesley.
- Booch, G., Rumbaugh, J. & Jacobson, I., 1998. *The Unified Modeling Language User Guide*. 1st ed. Canada: Addison Wesley.
- Departemen Kesehatan RI, 2009. *Departemen Kesehatan Republik Indonesia*. [Online]  
Available at: [http://pppl.depkes.go.id/asset/regulasi/KEPMENKES\\_374-2009\\_TTG\\_SKN-2009.pdf](http://pppl.depkes.go.id/asset/regulasi/KEPMENKES_374-2009_TTG_SKN-2009.pdf)  
[Accessed 3 4 2017].
- Jacobson, I., 1992. *Object-Oriented Software Engineering: A Use Case Driven Approach*. New York: Addison-Wesley.
- Krol, M. & Reich, D. L., 1999. Object-Oriented Analysis and Design of a Health Care Management Information System. *Journal of Medical Systems*, 24(2), pp. 145-158.
- Krol, P. & Kruchten, P., 2003. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Boston: Addison Wesley.
- Nugraha, D. C. A., 2017. *A Business Architecture Modeling Methodology to Support the Integration of Primary Health Care: Implementation of Primary Health Care in Indonesia*. s.l.:s.n.
- Odeh, M., Hauer, T., McClatchey, R. & Solomonides, T., 2004. A Use-Case Driven Approach in Requirements Engineering: The MAMMOGRID Project.
- Pressman, R. S., 2010. *Software Engineering: A Practitioner's Approach*. 7th ed. New York: McGraw-Hill.
- Roebijoso, J., 2010. Model Manajemen Strategik dan Operasional, Pelayanan Kesehatan Dokter Keluarga yang Berwawasan Mutu pada Karyawan dan Keluarga.
- Rumbaugh, J., Jacobson, I. & Booch, G., 2004. *The Unified Modeling Language Reference Manual*. 2nd ed. Boston: Addison Wesley.
- Sommerville, I., 2011. *Software Engineering*. 9th ed. Boston: Addison-Wesley.