

PENGEMBANGAN SISTEM INFORMASI ADMINISTRASI PEMERIKSAAN PASIEN DI INSTALASI RADIOLOGI RSUD KAJEN DENGAN UNIFIED PROCESS

Yadanur Alim, Suhartono, Priyo Sidik Sasongko

Program Studi Teknik Informatika, Fakultas MIPA, Universitas Diponegoro
yadanur.alim22@gmail.com

Abstract

Kajen Regional Hospital of Pekalongan Regency has applied information technology to improve the quality of service to the patients. Radiology installation is one of the installations in Kajen Regional Hospital which was being developed computer based information system to facilitate its operations. Earlier, the staff at the Radiology Installation of Kajen Regional Hospital used numbers processing application such as Microsoft Office Excel to process the data. The growth of patient and examination of patients data in the future was a consideration for the Radiology Installation of Kajen Regional Hospital of Pekalongan Regency to develop a Administration of Examination Patient Information System in Radiology Installation of Kajen Regional Hospital which was integrated with the database system. Administration of Examination Patient Information System in Radiology Installation of Kajen Regional Hospital was developed with the Unified Process. Unified Process is included in object-oriented software development paradigm. This system was implemented using the Visual Basic.Net programming language and MySQL 5 as its DBMS. This system was tested by using black box method, namely the testing methods that test the functionality of the software without having to know the internal structure of the program. This system can help the radiology staff in managing the administrative data and making the administrative reports.

Keywords: Information System, Administration, Unified Process, Object Oriented

1. Pendahuluan

Berdasarkan hasil *survey*, Rumah Sakit Umum Daerah (RSUD) Kajen Kabupaten Pekalongan telah menerapkan Teknologi Informasi untuk meningkatkan kualitas pelayanannya terhadap para pasien. Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan merupakan salah satu instalasi di RSUD Kajen Kabupaten Pekalongan yang tengah gencar dikembangkan sistem informasi berbasis komputer untuk mempermudah kegiatan operasionalnya.

Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan merupakan salah satu instalasi di RSUD Kajen Kabupaten Pekalongan yang melayani kegiatan pemeriksaan radiologi seperti *rontgen* dan Ultrasonografi (USG). Berdasarkan hasil *survey* dan wawancara dengan salah satu staf medis di Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan, sebelumnya para staf di Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan menggunakan aplikasi pengolah angka seperti Microsoft Office Excel untuk mengelola data pasien dan pemeriksaan pasien. Sebenarnya, aplikasi pengolah angka seperti

Microsoft Office Excel sudah memenuhi kebutuhan instalasi radiologi untuk saat ini, tetapi karena pertimbangan pertumbuhan data pasien dan pemeriksaan pasien di masa yang akan datang yang dikhawatirkan suatu saat tidak dapat ditangani oleh Microsoft Office Excel maka perlu dikembangkan suatu sistem informasi yang terintegrasi dengan sistem basis data.

Unified Process merupakan salah satu *software development process* yang menerapkan konsep berorientasi objek yang dikembangkan oleh Ivar Jacobson, Grady Booch, dan James Rumbaugh [6]. Dengan konsep berorientasi objek, perangkat lunak mempunyai kualitas yang tinggi dan dapat diuraikan serta diskalakan. Berdasarkan pertimbangan keunggulan yang dimiliki *Unified Process*, maka *Unified Process* digunakan dalam proses pengembangan sistem informasi ini, sehingga diharapkan dapat dihasilkan perangkat lunak yang berkualitas tinggi, *reusable*, dan mudah untuk dipelihara.

2. Tinjauan Pustaka

2.1 Sistem Informasi

Sistem informasi adalah kumpulan dari prosedur dan komponen yang saling berhubungan yang membentuk satu kesatuan untuk mencapai tujuan tertentu yakni mengolah data menjadi informasi sehingga lebih berguna bagi pengguna [7].

2.2 Administrasi

Menurut Kamus Besar Bahasa Indonesia susunan W.J.S Poerwadarminta administrasi diartikan sebagai tata usaha. Sedangkan tata usaha sendiri didefinisikan sebagai kegiatan penyelenggaraan urusan tulis menulis (keuangan dan sebagainya) dalam perusahaan (negara dan sebagainya).

2.3 Konsep Berorientasi Objek

Konsep berorientasi objek merupakan konsep yang membagi program menjadi objek-objek yang saling berinteraksi satu sama lain [4].

Beberapa istilah yang berkaitan dengan konsep berorientasi objek adalah [10]:

1) *Class* dan Objek,

Class dapat diartikan deskripsi secara umum atau cetak biru yang menggambarkan sekumpulan objek yang serupa. Sedangkan objek dapat diartikan sebagai bentuk instansiasi dari sebuah *class*.

2) Atribut,

Atribut adalah sesuatu yang melekat pada objek yang mendeskripsikan sifat objek atau dengan kata lain atribut merupakan sifat karakteristik yang dimiliki oleh objek.

3) *Method* atau Operasi,

Method adalah prosedur atau fungsi yang dimiliki suatu objek. *Method* mempresentasikan perilaku dari suatu objek.

4) *Message*,

Suatu objek berinteraksi dengan objek lainnya melalui *message*. *Message* menstimulasi *method* yang ditemui pada objek yang menerima *message* tersebut untuk dieksekusi atau dijalankan.

Konsep berorientasi objek mempunyai tiga karakteristik, yaitu [10] :

1) Enkapsulasi,

Enkapsulasi merupakan pengemasan atribut dan *method* dalam objek. Enkapsulasi bertujuan sebagai upaya untuk menyembunyikan informasi atau *information*

hiding sehingga tidak dapat diakses secara bebas oleh objek lain.

2) Pewarisan,

Pewarisan merupakan bentuk sifat *reuseable* dalam konsep berorientasi objek, artinya suatu *class* dapat menurunkan atribut dan operasinya ke *class* yang lain.

3) Polimorfisme,

Polimorfisme merupakan suatu konsep yang memungkinkan sejumlah operasi yang memiliki perilaku berbeda untuk mempunyai nama yang sama.

2.4 Unified Process

Unified Process merupakan salah satu *software development proses* yang telah mendukung konsep berorientasi objek. *Unified Process* menggunakan *Unified Modeling Language* (UML) sebagai notasi dalam menggambarkan proses pengembangan perangkat lunak.

Unified Process mempunyai tiga karakteristik utama, yaitu [7] :

1) *Use Case Driven*,

Maksud dari *use case driven* adalah bahwa *use case* tidak hanya merupakan alat untuk menspesifikasi kebutuhan fungsional sistem, tetapi *use case* juga mengarahkan semua *workflow* dalam proses pengembangan *software*.

2) *Architecture Centric*,

Architecture Centric berarti bahwa suatu arsitektur sistem dipergunakan sebagai suatu *artifact* primer untuk *conceptualizing*, membangun, pengelolaan, dan meningkatkan sistem dalam pembangunannya.

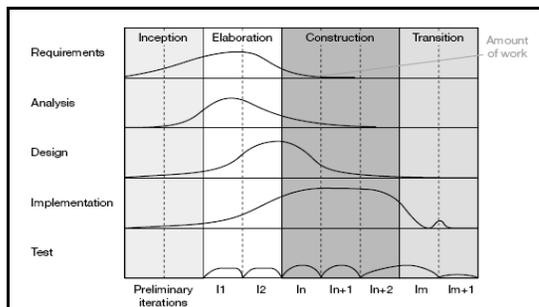
3) *Iterative & Incremental*,

Iterative berarti bahwa proses pengembangan perangkat lunak dilakukan berulang-ulang. Sedangkan *incremental* berarti bahwa setiap iterasi mengacu pada pertumbuhan produk.

Dalam *Unified Process*, pengembangan perangkat lunak dilakukan dalam sejumlah siklus yang berkelanjutan. Setiap siklus akan berakhir pada *release* sebuah produk. Setiap siklus tersebut terdiri atas empat fase yang berbeda, yaitu *inception*, *elaboration*, *construction*, dan *transition*. Setiap fase dapat terdiri atas sejumlah iterasi *workflow* (*requirement*, *analysis*, *design*, *implementation*, dan *test*). Selanjutnya, *workflow* menentukan

urutan aktivitas yang perlu dilakukan dan artifak apa yang harus dihasilkan. Saat pengembangan perangkat lunak beralih dari satu fase ke fase berikutnya, jumlah kerja yang dilakukan di tiap *workflow* juga berubah.

Pada fase *inception*, pekerjaan terbesar adalah *requirement* dan *analysis*. Fase *elaboration* menekankan pada *requirement*, *analysis*, dan sedikit *design*. Fase *construction* lebih menekankan pada *design* dan *implementation*, sedangkan fase *transition* penekanan beralih pada *implementation* dan *test* [2]. Gambar 1 menunjukkan hubungan antara fase dan *workflow* dalam *Unified Process*.



Gambar 1 Hubungan Fase dengan *Workflow* dalam *Unified Process*

Penjelasan tiap-tiap fase dalam *Unified Process* adalah sebagai berikut [6] :

- 1) *Inception*,
Inception mendefinisikan lingkup *project* dan mengembangkan *business case* untuk sistem. Fase ini terfokus pada *requirement* serta sedikit melakukan analisis dan *design*.
- 2) *Elaboration*,
Aktivitas yang dilakukan pada fase ini adalah melengkapi *requirement*, baik fungsional maupun nonfungsional serta melakukan analisis dan *design*.
- 3) *Construction*,
Construction merupakan fase pembangunan sistem. Sedikit analisis masih dapat dilakukan pada fase ini untuk melengkapi hasil analisis yang telah dihasilkan pada fase sebelumnya. Tiga *workflow* terakhir, yaitu *design*, implementasi, dan pengujian menjadi aktivitas utama dalam fase ini.
- 4) *Transition*,
Transition merupakan peralihan produk ke lingkungan *user* (*beta release*). *Beta release*

merupakan versi baru dari perangkat lunak yang belum diuji secara penuh dari kemungkinan adanya *defect* atau cacat program. Sejumlah *user* akan mencoba produk yang telah dihasilkan dan melaporkan *defect* yang ditemukan. *Defect* akan dibetulkan dan kemudian fase ini berakhir pada *release* produk resmi.

Unified Process terdiri atas lima *workflow*. Penjelasan tiap-tiap *workflow* tersebut adalah sebagai berikut [6] :

- 1) *Requirement*,
Tujuan dari *requirement* adalah untuk menemukan dan mencapai persetujuan mengenai apa yang harus dilakukan sistem yang diungkapkan dalam bahasa *user* [2]. Selanjutnya, *requirement* sistem tersebut akan digambarkan dalam istilah *use case*, sehingga output dari *requirement* ini berupa *use case model* yang terdiri atas *use case*, aktor, *GUI prototype*, dan *non-functional requirement* [5].
- 2) *Analysis*,
Analisis bertujuan untuk menerjemahkan *requirement* kedalam bahasa *developer* dan mengidentifikasi elemen atau entitas utama dari sistem yang diperlukan untuk memenuhi *user requirement* [5]. Artifak-artifak yang dihasilkan dalam tahap analisis antara lain *analysis class*, *use case realization analysis*, dan *analysis package*.
- 3) *Design*,
Design berusaha untuk menspesifikasikan secara penuh bagaimana fungsionalitas akan diimplementasikan dengan menggunakan model yang telah dihasilkan dari *workflow* analisis. Artifak yang dihasilkan dalam *workflow* ini meliputi *design subsystem*, *design classes*, dan *use case realization design* [2].
Database design yang akan digunakan sebagai tempat penyimpanan data juga dibuat pada tahap ini. *Database* yang digunakan dalam pengembangan sistem ini adalah *relational database*. *Relational database design* merupakan *mapping* kedalam *relational table* dari setiap *class* dalam *class diagram* yang bersifat *persistent*.

- 4) *Implementation*,
Implementation bertujuan untuk mentransformasikan *design model* ke dalam *executable code*. *Source code* program dibuat pada *workflow* ini.
- 5) *Test* (Pengujian)
 Tujuan dari pengujian adalah untuk memastikan bahwa sistem dapat menyediakan fungsionalitas yang diperlukan. Pengujian yang dilakukan hanya pengujian secara *black box*, yaitu menguji fungsionalitas dari perangkat lunak, tanpa harus mengetahui struktur internal program [10]. Artifacts yang dihasilkan pada *workflow* ini adalah *test model* [5].

2.5 Unified Modelling Language

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk menulis *blueprint* perangkat lunak. UML dapat digunakan untuk memvisualisasi, menspesifikasikan, membangun, dan mendokumentasikan artifacts dari sistem perangkat lunak.

UML terdiri atas tiga *building block*, yaitu [3]:

- 1) *Things*,
Things adalah *building block* berbasis objek yang utama dari UML. *Things* terdiri atas 4 macam, yaitu:
 - a) *Structural Things*,
Structural things adalah bagian model statis yang merepresentasikan elemen konseptual atau fisik. Jenis-jenis *structural things* adalah *class*, *interface*, *collaboration*, dan *use case*.
 - b) *Behaviorial Things*,
Behaviorial things adalah bagian dinamis dari model UML yang merepresentasikan *behavior*. Jenis *behaviorial things* adalah *interaction*, *state machine*, dan *activity*.
 - c) *Grouping Things*,
Grouping things adalah bagian dari model UML yang berfungsi untuk mengelompokkan elemen. Satu-satunya jenis *grouping things* adalah *package*.
 - d) *Annotational Things*,
Annotational things adalah bagian penjabar dari model UML. Jenis

annotational things adalah *note* yang digunakan untuk memberikan komentar.

- 2) *Relationship*,
Relationship merupakan *building block* UML yang berfungsi sebagai penghubung antar *things*. Jenis-jenis *relationship* antara lain *dependency*, *Association*, *Generalization*, *Realization*.
- 3) *Diagram*,
Diagram adalah presentasi grafis yang merupakan *kombinasi* antara *things* dan *relationship*. *Diagram* dibuat untuk memvisualisasikan sistem dari sejumlah perspektif yang berbeda, sehingga *diagram* merupakan proyeksi terhadap sistem. UML mempunyai tiga belas *diagram*. Penjelasan dari tiap-tiap *diagram* tersebut ditunjukkan pada Tabel 1.

Tabel 1 Nama – Nama Diagram

Nama Diagram	Kegunaan
<i>Activity</i>	<i>Behavioral prosedural</i> dan paralel
<i>Class</i>	<i>Class</i> , fitur, dan hubungan – hubungannya
<i>Communication</i>	Interaksi antar objek, penekanan pada jalur
<i>Component</i>	Struktur dan koneksi komponen
<i>Composite structure</i>	Dekomposisi <i>runtime</i> sebuah <i>class</i>
<i>Deployment</i>	Pemindahan artifacts ke <i>node</i>
<i>Interaction overview</i>	Campuran <i>sequence</i> dan <i>activity diagram</i>
<i>Object</i>	Contoh konfigurasi dari contoh – contoh
<i>Package</i>	Struktur hirarki <i>compile-time</i>
<i>Sequence</i>	Interaksi antar objek, penekanan pada <i>sequence</i>
<i>State machine</i>	Bagaimana <i>event</i> mengubah objek selama aktif
<i>Timing</i>	Interaksi antar objek, penekanan pada <i>timing</i>
<i>Use Case</i>	Bagaimana pengguna berinteraksi

2.6 Object Relational Mapping (ORM)

Object Relational Mapping (ORM) adalah suatu metode untuk memetakan objek menjadi tabel dalam *relational database* [1]. Konsep *mapping* objek ke *relational database* yang digunakan yaitu konsep pemetaan dasar dan pemetaan hubungan objek [1].

2.7 Visual Basic .Net dan MySQL

Visual Basic.Net merupakan bahasa pemrograman untuk membuat aplikasi berbasis windows. Visual Basic .Net dibangun di atas fondasi Framework .Net. Visual Basic .Net juga sudah mendukung pemrograman berorientasi objek [8].

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*) dan bersifat *open source* [12].

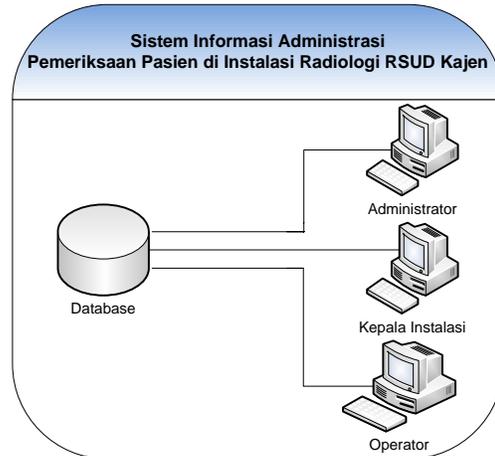
3. Definisi Kebutuhan, Analisis, dan Perancangan

3.1 Definisi Kebutuhan

Sistem Informasi Administrasi Pemeriksaan Pasien ini dikembangkan untuk diimplementasikan di Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan. Sistem ini memiliki tiga buah otoritas yaitu operator, kepala instalasi, dan administrator. Sistem ini dapat melayani pendaftaran pasien, menyusun rencana pemeriksaan pasien, menyimpan hasil diagnosa pemeriksaan pasien, menghitung tagihan pemeriksaan, dan menyusun laporan.

Sistem ini dibuat untuk mempermudah pekerjaan para staf di Instalasi Radiologi RSUD

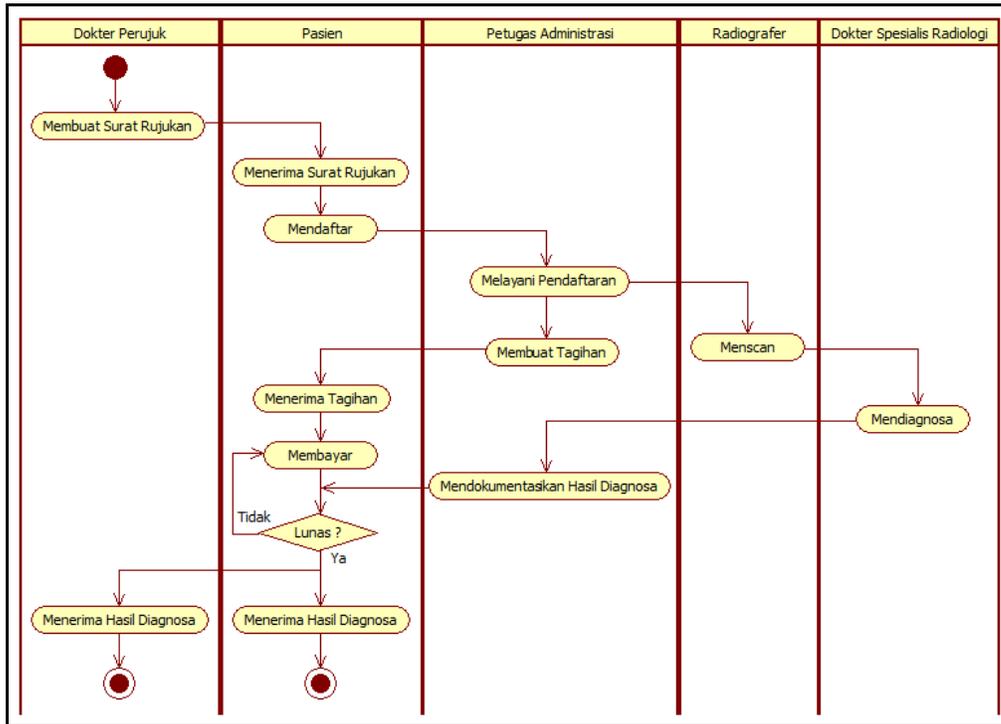
Kajen Kabupaten Pekalongan dalam mengelola data administrasi pasien, sehingga dapat meningkatkan kinerja pelayanan terhadap para pasien. Gambaran umum sistem yang dikembangkan dapat dilihat pada Gambar 2.



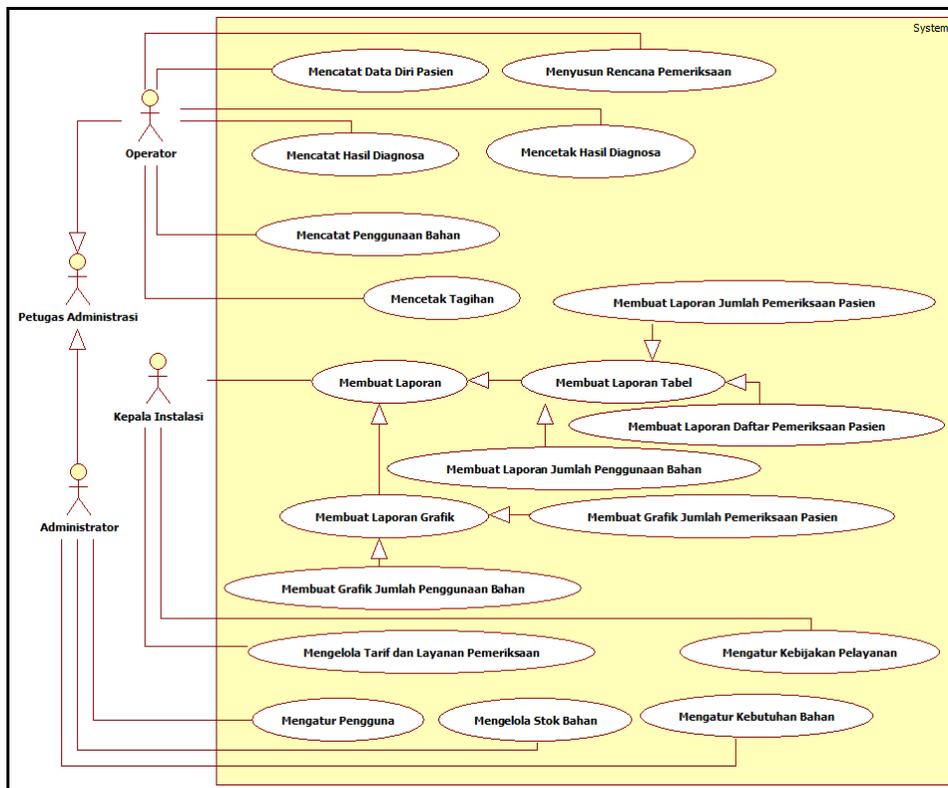
Gambar 2 Gambaran Umum Sistem

Prosedur atau alur kegiatan pemeriksaan pasien di Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan dapat dilihat pada diagram aktivitas Gambar 3.

Diagram *system use case* digunakan untuk menggambarkan sistem yang dikembangkan. Diagram *system use case* dari Sistem Informasi Administrasi Pemeriksaan Pasien di Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan dapat dilihat pada Gambar 4.



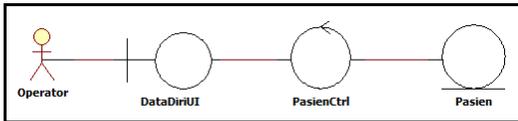
Gambar 3 Diagram Aktivitas Pemeriksaan Pasien



Gambar 4 Diagram System Use Case

3.2 Analisis

Setiap *use case* pada tahap analisis direalisasikan dengan menggunakan *analysis class diagram*. Realisasi *use case* “Mencatat Data Diri Pasien” ditunjukkan dengan *analysis class diagram* pada Gambar 5.



Gambar 5 *Analysis Class Diagram* Mencatat Data Diri Pasien

Class-class yang terkait pada *use case* “Mencatat Data Diri Pasien” antara lain :

- a) *Class Boundary*: DataDiriUI
- b) *Class Control*: PasienCtrl
- c) *Class Entity*: Pasien

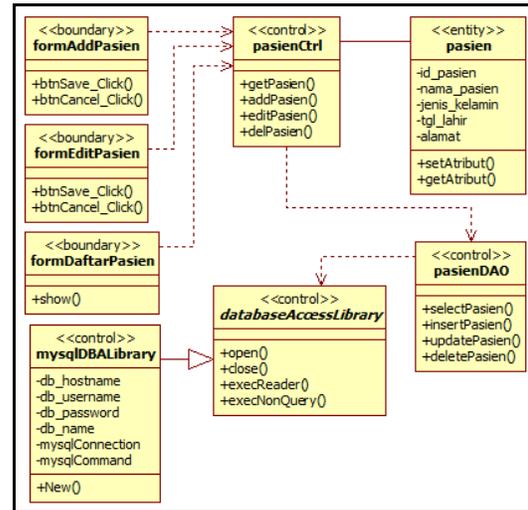
3.3 Perancangan

Realisasi *use case* tahap perancangan dapat digambarkan dengan *class diagram* dan *sequence diagram*. Tabel 2 menunjukkan hasil identifikasi *class* perancangan dari *class* analisis pada *use case* “Mencatat Data Diri Pasien”.

Tabel 2 Identifikasi *Class* Perancangan *Use Case* Mencatat Data Diri Pasien

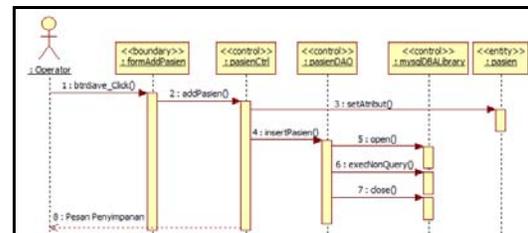
Stereotype	Class Analisis	Class Perancangan
boundary	DataDiriUI	formAddPasien
		formEditPasien
		formDaftarPasien
control	PasienCtrl	pasienCtrl
		pasienDAO
		databaseAccessLibrary
		mysqlDBALibrary
		mysqlDBALibrary
Entity	Pasien	pasien

Gambar 6 merupakan gambar *class diagram* yang menunjukkan hubungan antar *class* dari *use case* “Mencatat Data Diri Pasien”.



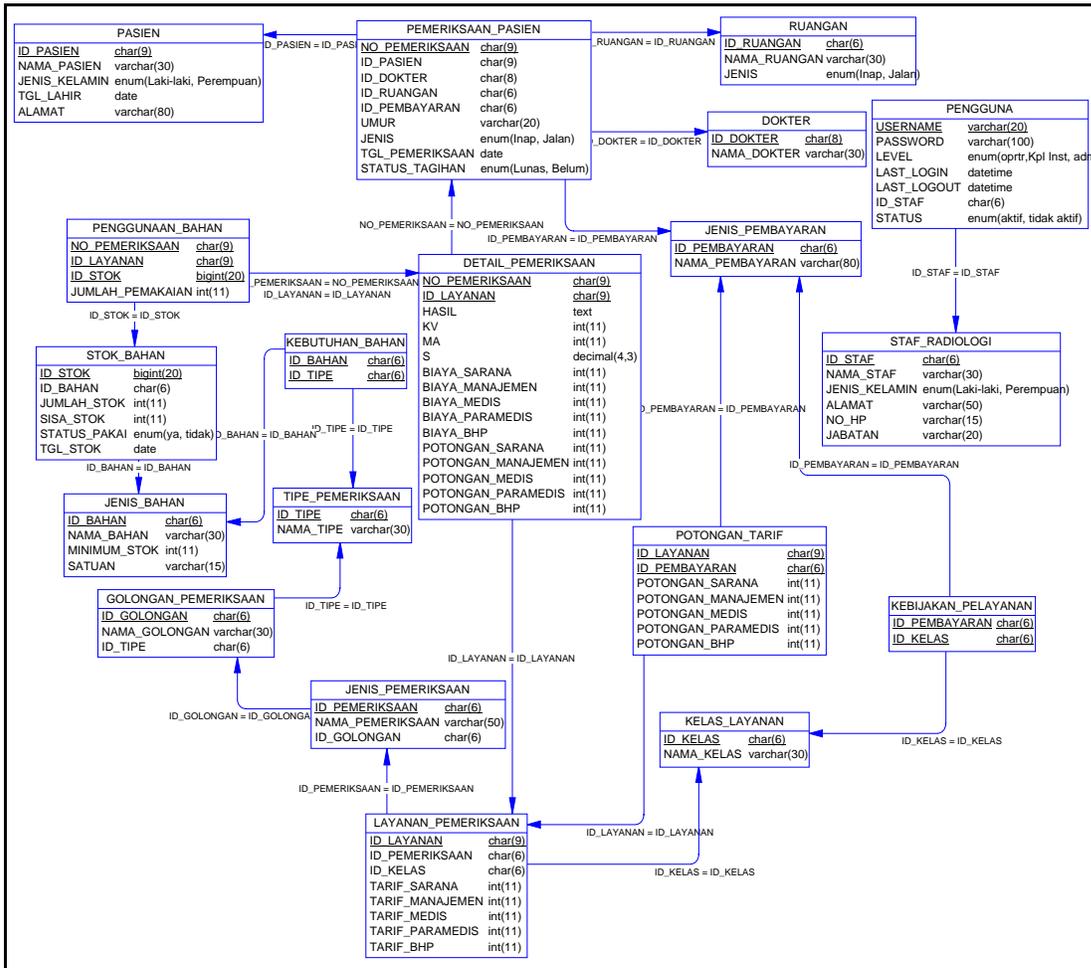
Gambar 6 *Class Diagram* untuk *Use Case* Mencatat Data Diri Pasien

Sedangkan salah satu contoh interaksi antar objek pada *use case* “Mencatat Data Diri Pasien” dapat dilihat pada *sequence diagram* yang ditunjukkan dengan Gambar 7.



Gambar 7 *Sequence Diagram* Mencatat Data Diri Pasien\

Rancangan *database* diperoleh dengan melakukan pemetaan *class-class entity* ke dalam tabel-tabel pada basis data relasional dengan menggunakan metode ORM (*Obeject Relational Mapping*). Skema rancangan *database* dari Sistem Informasi Administrasi Pemeriksaan Pasien di Instalasi Radiologi RSUD Kajan Kabupaten Pekalongan dapat dilihat pada Gambar 8.



Gambar 8 Skema Database

4. Implementasi dan Pengujian

4.1 Implementasi

Sistem Informasi Administrasi Pemeriksaan Pasien di Instalasi Radiologi RSUD Kajan Kabupaten Pekalongan diimplementasikan menggunakan bahasa pemrograman Visual Basic .Net dan MySQL 5 sebagai SMDBD-nya.

4.1.1. Implementasi Class

Implementasi *class* pada pengembangan Sistem Informasi Administrasi Pemeriksaan Pasien di Instalasi Radiologi RSUD Kajan Kabupaten Pekalongan ini direalisasikan kedalam *source code* dalam bahasa pemrograman Visual Basic .Net. *class* yang diimplementasikan sesuai dengan *stereotype* yang telah dirancang pada tahap perancangan.

4.1.2. Implementasi Antarmuka

Implementasi antarmuka untuk *use case* “Mencatat Data Diri Pasien” dapat dilihat pada

gambar 9. Operator mengisi data diri pasien kemudian menekan tombol ‘Simpan’ untuk menyimpan data diri pasien.

The screenshot shows a web form titled "FORM TAMBAH DATA PASIEN". It contains the following fields and controls:

- Nama**: A text input field.
- Jenis Kelamin**: Radio buttons for "Laki-laki" (selected) and "Perempuan".
- Tanggal Lahir**: A date picker showing "10 Mei 2011".
- Alamat**: A text input field.
- At the bottom right, there are two buttons: "Simpan" (Save) and "Batal" (Cancel).

Gambar 9 Tampilan Form Tambah Data Pasien

4.2 Pengujian

Pengujian Sistem Informasi Administrasi Pemeriksaan Pasien di Instalasi Radiologi RSUD

Kajen Kabupaten Pekalongan ini dilakukan dengan metode *black box*. Pengujian *black box* diterima jika fitur-fitur yang ada pada perangkat lunak memenuhi spesifikasi kebutuhan sistem atau *use case* yang telah diidentifikasi sebelumnya.

Dari hasil pengujian menunjukkan bahwa Sistem Informasi Administrasi Pemeriksaan Pasien di Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan ini telah memenuhi untuk :

- 1) Mencatat data diri pasien yang akan melakukan pemeriksaan radiologi.
- 2) Menyusun rencana pemeriksaan yang hendak dilakukan pasien.
- 3) Mencatat penggunaan bahan pemeriksaan.
- 4) Mencatat hasil diagnosa dokter spesialis radiologi terhadap hasil *scan* pasien.
- 5) Mencetak hasil diagnosa dokter melalui media pencetakan yaitu printer.
- 6) Menghitung tagihan dan mencetaknya melalui media pencetakan yaitu printer.
- 7) Membuat laporan daftar pemeriksaan pasien.
- 8) Membuat laporan jumlah pemeriksaan pasien.
- 9) Membuat laporan jumlah penggunaan bahan pemeriksaan.
- 10) Membuat grafik jumlah pemeriksaan pasien.
- 11) Membuat grafik jumlah penggunaan bahan pemeriksaan.
- 12) Mengelola data layanan pemeriksaan serta mengatur tarif dan potongan tarifnya.
- 13) Mengatur kebijakan pelayanan pemeriksaan radiologi.
- 14) Mengatur kebutuhan bahan dari tiap tipe pemeriksaan.
- 15) Mengelola stok dari bahan pemeriksaan.
- 16) Mengelola dan mengatur pengguna sistem.

5. Kesimpulan dan Saran

Kesimpulan yang dapat diambil dalam pembuatan tugas akhir ini adalah dihasilkan sebuah Sistem Informasi Administrasi Pemeriksaan Pasien di Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan yang dikembangkan dengan *Unified Process*. Dengan *Unified Process*, sistem ini dapat dikembangkan lebih lanjut untuk diintegrasikan dengan sistem

lain seperti sistem administrasi rumah sakit, sistem kasir rumah sakit, dan sistem-sistem lain yang berada di instalasi lain di RSUD Kajen Kabupaten Pekalongan. Sistem ini dapat membantu pekerjaan para staf radiologi dalam mengelola data pasien beserta pemeriksaannya dan membantu dalam menghitung tagihan pemeriksaan pasien serta menyusun laporan. Dengan adanya sistem ini kinerja para staf radiologi dapat meningkat, sehingga pelayanan terhadap pasien pun semakin optimal.

Sistem Informasi Administrasi Pemeriksaan Pasien di Instalasi Radiologi RSUD Kajen Kabupaten Pekalongan ini sebaiknya dikembangkan lebih lanjut untuk diintegrasikan dengan sistem yang lain seperti sistem administrasi rumah sakit, sistem kasir rumah sakit, dan sistem-sistem lain yang berada di instalasi lain di RSUD Kajen Kabupaten Pekalongan, sehingga sistem ini dapat saling berbagi data dengan sistem-sistem lain di RSUD Kajen Kabupaten Pekalongan.

Referensi

- [1] Ambler S. W., 2002, "*Mapping Objects To Relational Databases*".
- [2] Arlow J. dan Neustadt I., 2002, *UML and The Unified Process Practical Object-Oriented Analisis & Design*, Addison Wesley.
- [3] Booch G., Rumbaugh J., Jacobson I., 2005. *The Unified Modeling Language User Guide Second Edition*. Addison Wesley Professional.
- [4] Hermawan B., 2004, *Menguasai Java 2 & Object Oriented Programming*, Penerbit Andi, Yogyakarta.
- [5] Hunt J., 2003, *Guide to the Unified Process featuring UML, Java and Design Patterns*, Springer, London.
- [6] Jacobson I., Booch G., dan Rumbaugh J., 1999, *The Unified Software Development Process*, Addison Wesley.
- [7] Jogiyanto, 2003, *Sistem Teknologi Informasi*, Penerbit Andi, Yogyakarta.
- [8] Kusumo A. S., 2006, *Pemrograman Visual Basic 2005*, Elex Media Komputindo, Jakarta.

- [9] Poerwadarminta W. J. S., 1984, *Kamus Besar Bahasa Indonesia*, PN Balai Pustaka, Jakarta.
- [10] Pressman R. S., 2001, *Software Engineering : A Practitioner's Approach Fifth Edition*, McGraw – Hill, New York.
- [11] Sholiq, 2006, *Pemodelan Sistem Informasi Berorientasi Objek dengan UML*, Graha Ilmu, Yogyakarta.
- [12] Valade J., 2010, *PHP and MySQL For Dummies, 4th Edition*, Wiley Publishing, Inc., Indiana.