

Pengembangan Sistem Pengelolaan Rekam Medis Pasien Berbasis Web Studi Kasus: Klinik Dr. Syarif

Helman Taufiqurrahman¹, Tri Astoto Kurniawan²

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹x4helman.smansa@gmail.com, ²triak@ub.ac.id

Abstrak

Klinik Dr. Syarif merupakan salah satu klinik yang berada di Kota Samarinda, Kalimantan Timur. Pengelolaan rekam medis pasien pada klinik tersebut masih dilakukan dengan menulis di kertas rekam medis dan disimpan di almari. Seiring waktu, hal tersebut dapat menyebabkan bertambahnya kebutuhan tempat untuk menyimpan rekam medis pasien. Duplikasi rekam medis dapat terjadi ketika petugas administrasi meletakkan rekam medis pada bagian almari yang salah, menyebabkan rekam medis tidak ditemukan dan petugas administrasi melakukan pembuatan rekam medis yang baru. Untuk mengatasi masalah tersebut, sistem pengelolaan rekam medis pasien berbasis web dikembangkan dengan tujuan untuk mempermudah proses pengelolaan rekam medis pasien. Sistem dikembangkan dengan menggunakan metode *Object Oriented Analysis & Design* (OOAD) dan metode *Object Oriented Programming* (OOP). Bahasa pemrograman yang digunakan untuk mengembangkan sistem tersebut adalah *Personal Home Page* (PHP) dan *framework* yang digunakan adalah *Codeigniter*. Pengujian unit dan pengujian integrasi dilakukan dengan menerapkan metode pengujian *white-box*, sedangkan pengujian validasi dilakukan dengan menerapkan metode pengujian *black-box*. Hasil dari pengujian unit dan juga pengujian integrasi menunjukkan bahwa sistem sanggup berjalan sesuai dengan rancangan yang telah dibuat. Hasil dari pengujian validasi menunjukkan bahwa seluruh kebutuhan fungsional telah terpenuhi oleh sistem. Dengan kata lain, hasil dari seluruh pengujian adalah 100% valid dan sistem layak untuk digunakan.

Kata kunci: *Sistem informasi, pengelolaan data, rekam medis, web, berorientasi objek*

Abstract

Dr. Syarif's Clinic is a clinic that is located in Samarinda City, East Borneo. The management of the patient's medical records in that clinic is still done by writing on a paper and stored inside the cupboard. As time goes on, it increases the place's needs for saving the medical records. Duplication of medical records could happen when the administration staff puts the medical records at the wrong side of the cupboard, which makes the medical records could not be found and the administration staff makes a new medical record. To solve these problems, a web-based management system of patient's medical records was developed for the intention of making the process of managing the patient's medical records easier. The system was developed using Object-Oriented Analysis & Design (OOAD) and Object-Oriented Programming (OOP) methods. The programming language that was used to develop the system is Personal Home Page (PHP) and the framework that was used is Codeigniter. The unit testing and the integration testing were done by using the white-box testing method, while the validation testing was done by using the black-box testing method. The result of unit testing and integration testing has shown that the system runs in accordance with the design that has been made. The result of the validation testing has shown that all the functional requirements have been fulfilled by the system. In other words, the result of all the testing is 100% valid and the system is worth to use.

Keywords: *Information System, management of data, medical record, web, object-oriented*

1. PENDAHULUAN

Klinik Dr. Syarif merupakan klinik yang berlokasi di Kota Samarinda, Kalimantan Timur. Sama seperti klinik pada umumnya, Klinik Dr. Syarif memiliki beberapa macam bidang pekerjaan, seperti dokter, petugas administrasi, dan apoteker. Dokter bertugas dalam melakukan pemeriksaan dan memberikan penanganan terhadap masalah yang dimiliki oleh pasien. Petugas administrasi bertugas dalam melayani pasien yang ingin mendaftar di klinik dan membuat rekam medis bagi pasien yang belum memilikinya. Apoteker bertugas dalam mengelola obat-obatan yang terdapat pada klinik.

Jumlah pasien yang semakin bertambah setiap tahunnya menyebabkan dibutuhkan tempat penyimpanan rekam medis yang lebih besar. Proses pencarian rekam medis pasien juga memakan waktu yang lama, dikarenakan banyaknya rekam medis yang dimiliki klinik. Duplikasi rekam medis juga dapat terjadi, jika pasien tidak ingat jika sebelumnya sudah berobat di klinik atau karena rekam medis tidak ditemukan saat dicari disebabkan oleh rekam medis terselip saat diletakkan pada tempat penyimpanan.

Permasalahan tersebut menandakan bahwa Klinik Dr. Syarif membutuhkan sistem yang dapat mengelola data rekam medis pasien, mencari rekam medis, dan dapat melakukan proses penyimpanan data rekam medis dengan jumlah yang banyak. Maka dari itu, penelitian ini mengambil judul "Pengembangan Sistem Pengelolaan Rekam Medis Pasien Berbasis Web. Studi Kasus: Klinik Dr. Syarif". Sistem tersebut dibuat dengan harapan dapat mempermudah klinik dalam mengelola rekam medis pasien dan mengurangi kemungkinan terjadinya duplikasi rekam medis.

2. LANDASAN KEPUSTAKAAN

2.1 Pengelolaan Rekam Medis

Menurut Huffman (1994), yang dimaksud dengan rekam medis adalah salah satu bentuk dari berkas yang menyatakan siapa pasien yang dirawat, apa keluhan yang diutarakan oleh pasien, dan mengapa pasien bisa mendapatkan keluhan tersebut. Rekam medis juga berisi tentang dimana pemeriksaan kesehatan pasien dilakukan, kapan pemeriksaan tersebut dilaksanakan, dan bagaimana pelayanan yang diperoleh seorang pasien. Pengelolaan rekam

medis dapat diselenggarakan dengan mengacu pada tata kerja dan organisasi sarana pelayanan kesehatan (Permenkes RI No. 269, 2008).

2.1.1 Pendaftaran Pasien

Pendaftaran pasien pada klinik Dr. Syarif dimulai oleh pasien dengan menyebutkan nama dirinya dan nomor BPJS (jika ada) kepada petugas administrasi. Petugas administrasi kemudian bertanya kepada pasien jika pasien tersebut pernah berobat sebelumnya pada klinik Dr. Syarif tersebut. Apabila pasien tersebut menjawab pernah, petugas akan melakukan proses pencarian rekam medis di tempat penyimpanan yang terdapat pada klinik. Jika belum pernah, petugas akan memberikan formulir untuk keperluan pembuatan rekam medis pasien yang terdiri dari nomor telepon, jenis kelamin, alamat, tanggal lahir, tempat lahir, nomor BPJS jika mempunyai, dan nama pasien. Pasien mengisi formulir secara lengkap dan mengembalikannya kepada petugas administrasi. Petugas administrasi lalu membuat rekam medis pasien sesuai formulir tersebut. Petugas kemudian memberikan nomor antrian kepada pasien.

2.1.2 Pemeriksaan Pasien

Proses pemeriksaan pasien pada klinik Dr. Syarif dimulai ketika nomor antrian sudah menjadi giliran pasien. Asisten dokter membawa rekam medis pasien dan memasuki ruangan pemeriksaan bersama pasien. Asisten dokter kemudian memberikan rekam medis tersebut kepada dokter. Pasien kemudian memberitahukan keluhannya. Dokter memeriksa fisik pasien lalu menambahkan hasil pemeriksaan kedalam rekam medis. Hasil pemeriksaan tersebut terdiri dari tanggal pemeriksaan, keluhan pasien, hasil pemeriksaan fisik, diagnosa, obat yang dibutuhkan, dan catatan tambahan. Dokter kemudian membuat resep obat lalu memberikannya kepada pasien. Pasien keluar ruangan dan asisten dokter mengembalikan rekam medis kepada petugas administrasi.

3. METODE PENELITIAN

Model *waterfall* atau sekuensial linier merupakan model pengembangan sistem (*Software Development Life Cycle*) yang digunakan dalam penelitian ini. Tahapan-tahapan yang terdapat pada model *waterfall* adalah rekayasa kebutuhan, perancangan sistem,

implementasi sistem, dan pengujian sistem. Gambar 1 merupakan diagram alir dari metodologi penelitian ini.

Pada penelitian ini, studi literatur merupakan tahapan awal dalam melakukan penelitian. Teori-teori dari internet, jurnal, dan buku yang berhubungan dengan topik yang diteliti dipelajari pada tahap ini.

Tahap kedua adalah melakukan tahapan kebutuhan. Tahap kebutuhan bertujuan untuk mengetahui macam-macam jenis dari kebutuhan yang harus terdapat pada sistem. Tahap ini juga menghasilkan *use case diagram* dan *use case scenario* yang dapat digunakan untuk mengetahui aktor yang terlibat di dalam penggunaan sistem. Selain itu, tindakan yang mampu dikerjakan oleh aktor beserta langkah-langkahnya juga dapat diketahui.

Tahap berikutnya ialah tahapan perancangan. Pada tahap ini, rancangan sistem dibuat agar kelak dapat diimplementasikan untuk membuat sistem secara utuh. Selanjutnya adalah tahap implementasi. Hasil dari tahap perancangan yang sebelumnya sudah dihasilkan kemudian dilakukan implementasi untuk mengembangkan rancangan menjadi sistem yang utuh.

Tahap berikutnya adalah tahapan pengujian. Pengujian sistem dilakukan untuk mendapatkan informasi jika sistem yang dibangun telah memiliki kesesuaian terhadap apa saja yang dibutuhkan pada tahap kebutuhan. Tahap pengujian juga menguji alur dari fungsi-fungsi sistem, apakah alur-alur tersebut sudah memiliki kesesuaian terhadap *usecase scenario* atau tidak. Macam-macam pengujian yang dilakukan dalam penelitian ini adalah pengujian unit, pengujian integrasi, dan pengujian validasi.

Tahap terakhir yang dilakukan adalah penutup. Tahap ini terdiri dari kesimpulan dan saran. Hasil dari tahap kebutuhan, perancangan, implementasi, dan pengujian disimpulkan pada tahap kesimpulan. Saran berguna untuk mengembangkan sistem agar menjadi lebih baik.

4. TAHAP KEBUTUHAN

Tahap kebutuhan menghasilkan beberapa aktor yang terlibat pada sistem. Selain itu, pada tahap ini juga menghasilkan kebutuhan fungsional dan skenario sistem. Teknik elisitasi kebutuhan yang digunakan adalah wawancara. Pihak yang diwawancarai adalah Dr. Syarif, selaku dokter yang bertugas pada Klinik Dr. Syarif. Aktor-aktor yang dapat menggunakan

sistem adalah pengguna, admin, petugas administrasi, dokter, dan apoteker. Tabel 1 merupakan identifikasi aktor. Kebutuhan fungsional yang dihasilkan dari proses penggalian kebutuhan adalah sebanyak 20 kebutuhan. Gambar 2 merupakan gambaran dari perilaku sistem dari tampak luar dan aktor apa saja yang dapat melakukan perilaku tersebut. Tabel 2 merupakan daftar dari sebagian kebutuhan yang terdapat pada sistem, aktor yang dapat melakukannya, dan nama pada *usecase diagram*.



Gambar 1. Diagram alir metodologi penelitian

Tabel 1. Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna merupakan aktor yang belum masuk ke dalam sistem.
Admin	Admin berperan dalam mengelola data pengguna sistem.
Petugas Administrasi	Petugas administrasi berperan dalam mengelola rekam medis pasien.
Dokter	Dokter berperan dalam melaksanakan proses pemeriksaan dan menambahkan hasil pemeriksaan terhadap pasien tersebut kepada rekam medis pasien.
Apoteker	Apoteker berperan dalam mengelola data obat.

Tabel 2. Daftar dari sebagian kebutuhan fungsional

No	Kebutuhan	Aktor	Use case
1	Sistem harus sanggup menyediakan fungsi login	Pengguna	Login
2	Sistem harus sanggup menyediakan fungsi menambah	Admin	Menambah Pengguna

	pengguna		
3	Sistem harus sanggup menyediakan fungsi menambah hasil pemeriksaan	Dokter	Menambah Hasil Pemeriksaan
4	Sistem harus sanggup menyediakan fungsi menambah pasien baru	Petugas Admin-istrasi	Menambah Pasien Baru
5	Sistem harus sanggup menyediakan fungsi menambah obat baru	Apoteker	Menambah Obat Baru

5. TAHAP PERANCANGAN

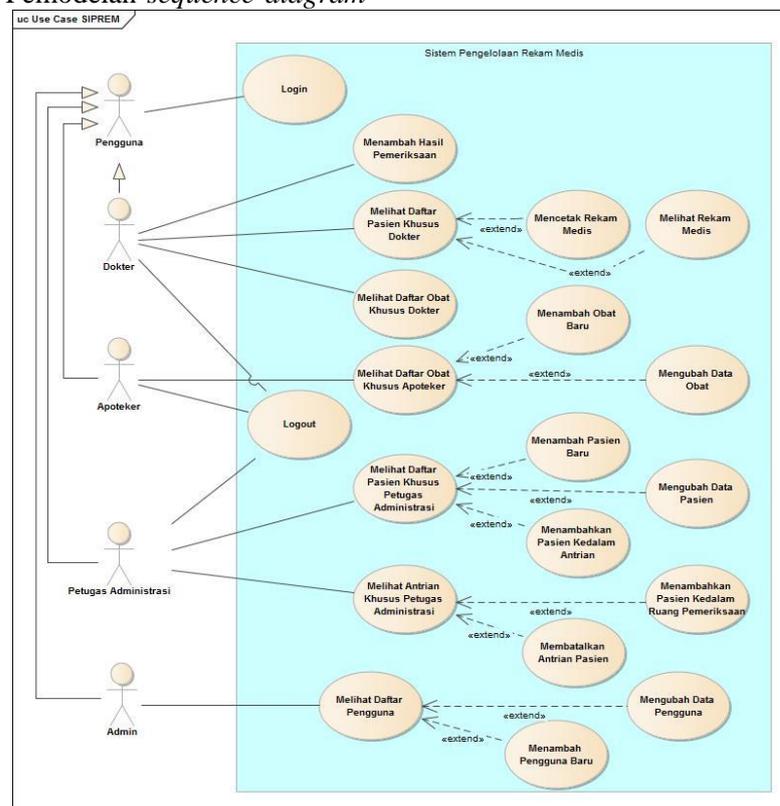
Perancangan arsitektural, perancangan data, perancangan antarmuka, dan perancangan komponen merupakan macam-macam perancangan yang dilakukan pada tahapan perancangan.

Pemodelan *sequence diagram* dan pemodelan diagram kelas atau *class diagram* dilakukan pada bagian perancangan arsitektural. Pemodelan *sequence diagram* menggambarkan pertukaran pesan yang dilakukan oleh aktor dan objek serta pertukaran pesan antara objek-objek di dalam sistem. Pemodelan *sequence diagram*

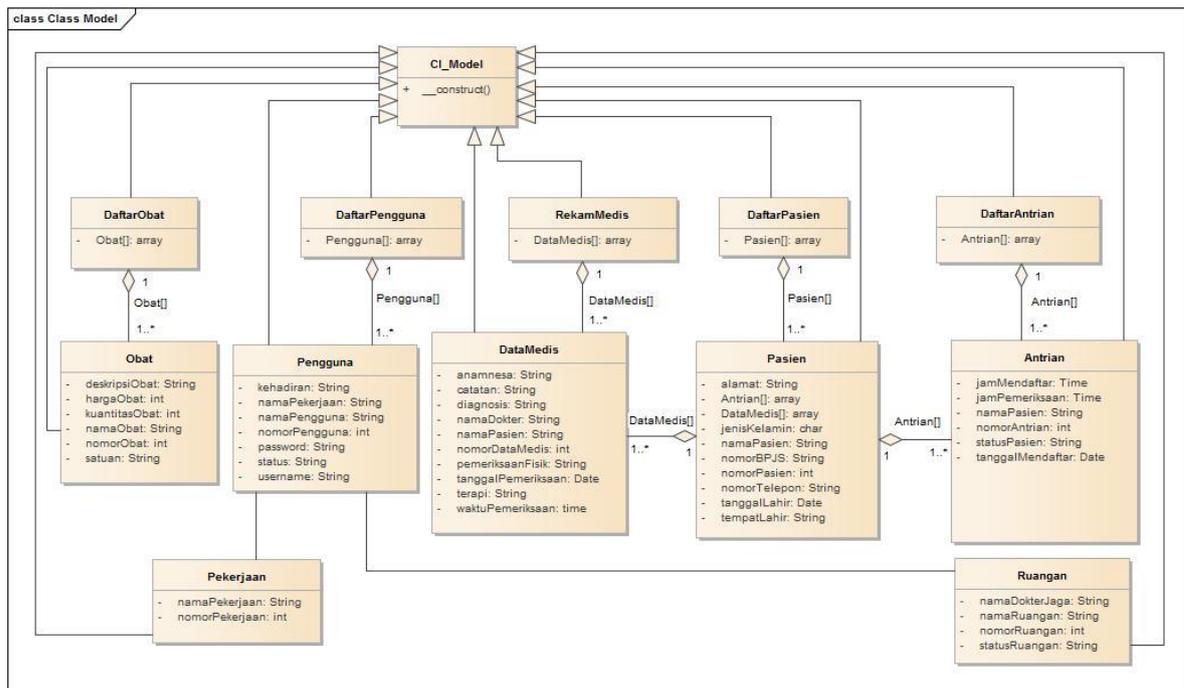
diperoleh berdasarkan *use case scenario* yang telah dibuat pada tahap kebutuhan. Pemodelan *class diagram* menggambarkan kelas-kelas beserta relasinya yang terdapat di dalam sistem. Kelas-kelas tersebut terdiri dari 17 kelas *boundary*, 6 kelas *controller*, dan 12 kelas *entity*. Kelas *boundary* berfungsi sebagai kelas yang langsung berinteraksi dengan aktor. Kelas *controller* berfungsi sebagai pengatur aliran informasi dari kelas *boundary* menuju kelas *entity*. Kelas *entity* berfungsi sebagai kelas yang merepresentasikan data sistem. Kelas-kelas *entity* tanpa *method* pada sistem digambarkan pada Gambar 3.

Perancangan data memiliki tujuan untuk mengubah model domain informasi yang diperoleh berdasarkan hasil dari analisis menjadi *data structure* yang akan digunakan pada tahap implementasi, yang digambarkan dalam dua bentuk. Bentuk pertama ialah PDM (*Physical Data Model*) dan bentuk yang kedua ialah CDM (*Conceptual Data Model*). CDM tanpa atribut yang terdapat pada sistem dapat dilihat pada Gambar 4. Gambar 5 merupakan entitas Pasien beratribut yang terdapat pada CDM.

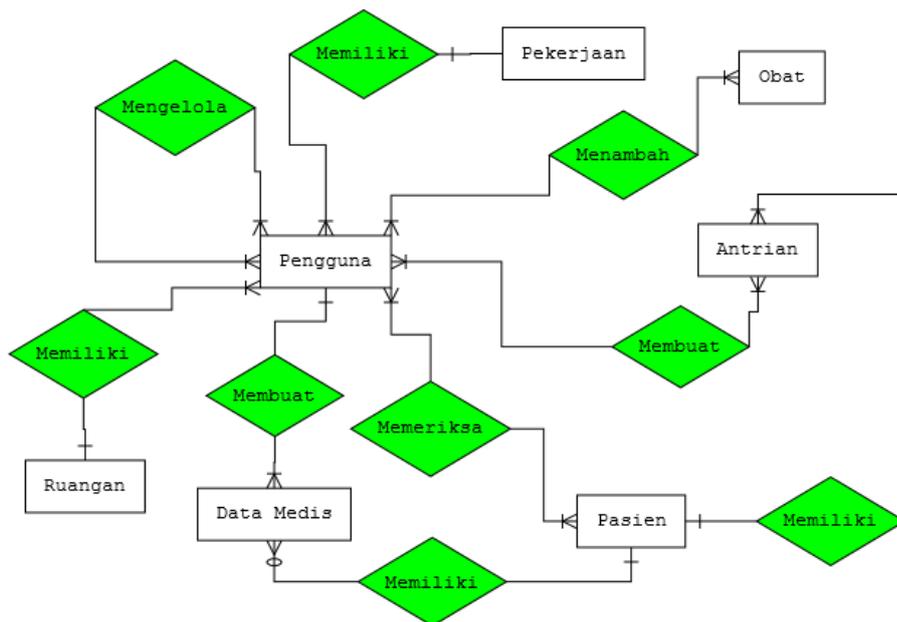
Proses yang akan terjadi pada sistem dirancang menjadi *pseudocode* pada tahap perancangan komponen. Perancangan antarmuka memiliki tujuan untuk menunjukkan rancangan awal dari antarmuka sistem.



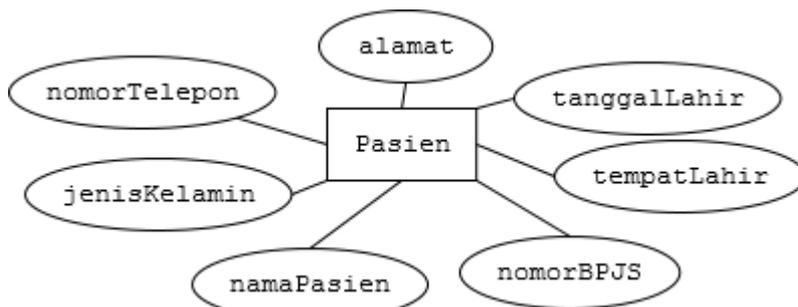
Gambar 2. Use Case Diagram



Gambar 3. Pemodelan Class Diagram Entity



Gambar 4. Conceptual Data Model tanpa atribut



Gambar 5. Detail Entitas Pasien

6. IMPLEMENTASI SISTEM

Tahap ini dijalankan dengan beracuan kepada tahapan sebelumnya yang telah dilakukan, yaitu perancangan. Tahap implementasi terbagi menjadi beberapa bagian, yaitu implementasi data, implementasi antarmuka, dan yang terakhir adalah tahap implementasi komponen. Gambar 6 merupakan contoh dari hasil implementasi antarmuka dari Sistem Pengelolaan Rekam Medis Pasien Berbasis Web. Implementasi sistem menerapkan pendekatan pemrograman memiliki orientasi berupa objek atau biasa disebut *object-oriented programming*. Selain itu, beberapa macam bahasa pemrograman yang digunakan penelitian ini ialah HTML, PHP, CSS, dan JS.

7. PENGUJIAN SISTEM

Pengujian sistem memiliki tujuan mengetahui apabila sistem yang dikembangkan dapat bekerja dengan memiliki kesesuaian terhadap hasil dari perancangan yang telah dibuat. Pengujian sistem memiliki tujuan lain yaitu untuk mendapatkan informasi apakah sistem yang dikembangkan dapat melengkapi keseluruhan dari kebutuhan yang terdapat pada

tahap kebutuhan. Pengujian sistem pada penelitian ini terbagi menjadi tiga jenis. Pengujian pertama ialah pengujian unit, pengujian berikutnya yang dilakukan ialah pengujian integrasi, dan pengujian terakhir yang dilakukan dalam penelitian ini ialah pengujian validasi. Teknik pengujian *white box* diterapkan pada pengujian unit dan pengujian integrasi. Teknik pengujian *black box* diterapkan pada pengujian validasi.

Pengujian unit dikerjakan dengan melakukan penerapan metode pengujian jalur dasar atau biasa dikenal dengan *basis path testing*. Unit yang diuji adalah klas *entity* bernama Pasien dengan *method* tambahPasien(), klas Pengguna dengan *method* tambahPengguna(), dan klas Obat dengan *method* tambahObat(). Ketiga *method* tersebut menghasilkan tiga buah *test case*. Ketiga *test case* tersebut bernilai valid. Contoh pengujian unit klas Pasien dapat dilihat pada *pseudocode method* tambahPasien() pada Tabel 3 dan *flowgraph method* tambahPasien() pada Gambar 7. Tabel 4 merupakan hasil pengujian unit pada *method* tambahPasien() klas Pasien.

Gambar 6. Implementasi Antarmuka Halaman Menambah Pasien Baru

Tabel 3. *Pseudocode Method* tambahPasien() kelas Pasien

<i>Pseudocode</i>	
START	
	data = array(
	'Nama_Pasien' = namaPasien,
	'Nomor_BPJS' = nomorBPJS,
	'Tempat_Lahir' = tempatLahir,
	'Alamat' = alamat,
	'Jenis_Kelamin' = jenisKelamin,
	'Tanggal_Lahir' = tanggalLahir,
	'Nomor_Telepon' = nomorTelepon)
	Query insert
END	



Gambar 7. Flow Graph tambahPasien()

▪ *Cyclomatic Complexity*

$$V(G) = \text{Jumlah Region} = 1$$

$$V(G) = \text{Edge} - \text{Node} + 2 = 0 - 1 + 2 = 1$$

$$V(G) = \text{Predicate Node} + 1 = 0 + 1 = 1$$

▪ *Independent Path*

Jalur 1 : 1

Tabel 4. Hasil Uji kelas Pasien *method* tambahPasien()

No. Jalur	1
Prosedur Uji	<i>Method</i> tambahPasien dari kelas Pasien dipanggil oleh kelas Driver dengan Nama pasien = "Fikri Arroisi", Nomor BPJS = "0000345763465", Jenis Kelamin = "L", Tempat Lahir = "Malang", Tanggal Lahir = 1996-09-11, Alamat = "Jalan Singosari", Nomor Telepon = "081273645382"
<i>Expected Result</i>	Memanggil <i>method</i> tambahPasien dan return value true
<i>Result</i>	Memanggil <i>method</i> tambahPasien dan return value true
Status	Valid

Pengujian integrasi dikerjakan pada tiga sampel pengujian. Pengujian integrasi yang dilakukan pertama kali ialah menguji operasi tambahPasien() pada kelas PasienController yang memanggil operasi tambahPasien() pada kelas Pasien. Pengujian kedua yaitu operasi tambahPegguna() pada kelas PenggunaController memanggil operasi tambahPegguna() pada kelas Pengguna. Pengujian ketiga yaitu operasi tambahObat() pada kelas ObatController memanggil operasi tambahObat() pada kelas Obat. Ketiga sampel pengujian integrasi tersebut menghasilkan 7 kasus uji dengan hasil pengujian 100% valid.

Pengujian validasi atau *validation testing* dilakukan dengan cara melakukan pengujian terhadap seluruh kebutuhan fungsional yang terdapat di sistem, dimana pengujian tersebut dilakukan dengan membuat 37 *test case* berdasarkan 20 kebutuhan fungsional. Hasil dari pengujian validasi adalah seluruh keluaran dari kebutuhan fungsional sistem sesuai dengan harapan, yang menandakan bahwa seluruh pengujian validasi memberikan hasil berupa nilai *valid* sebesar 100%.

8. KESIMPULAN

Tahap kebutuhan menghasilkan lima aktor yang terdiri dari pengguna, admin, petugas administrasi, dokter, dan apoteker. Tahap kebutuhan juga memberikan hasil 20 kebutuhan fungsional, yang dimana kebutuhan fungsional tersebut harus terdapat pada sistem, seperti sistem harus dapat menyediakan layanan menambahkan *user*, sistem harus dapat meleakakukan penyediaan sarana menambahkan hasil pemeriksaan, dan sistem harus sanggup melakukan penyediaan sarana menampilkan data diri yang dimiliki oleh pasien.

Tahap perancangan menghasilkan perancangan antarmuka, perancangan komponen, perancangan arsitektural, dan perancangan data. Perancangan arsitektural terbagi menjadi dua bagian, yaitu pemodelan diagram sekuensial atau biasa disebut *sequence diagram* dan pemodelan diagram kelas atau biasa disebut *class diagram*. Pemodelan diagram kelas memberikan gambaran berupa *classes* atau kelas-kelas yang berkaitan dengan sistem, seperti kelas Pengguna, Pasien, DataMedis, dan Obat. Perancangan komponen menghasilkan *pseudocode* berdasarkan proses yang terdapat pada komponen sistem. CDM dan PDM merupakan hasil dari perancangan data. Perancangan antarmuka menghasilkan desain awal dari antarmuka sistem. Pendekatan pemrograman berorientasi objek diterapkan pada tahap implementasi sistem. Implementasi sistem juga menggunakan *codeigniter* sebagai *framework*. Selain itu, basis data yang digunakan adalah MySQL.

Tahap pengujian sistem terdiri dari pengujian unit, pengujian integrasi, dan yang terakhir adalah pengujian validasi. Pengujian unit dilakukan kepada tiga *method* yang berbeda pada tiga kelas yang berbeda, dengan nilai *cyclomatic complexity* 1, 1, dan 1. Dari ketiga *test case* pengujian unit, seluruhnya memberikan

hasil berupa nilai sebesar 100% *valid*. Pengujian integrasi dilakukan kepada klas PasienController dengan klas Pasien, PenggunaController dengan klas Pengguna, dan ObatController dengan klas Obat. Pengujian integrasi memberikan hasil *valid* sebesar 100%. Pengujian validasi memuat 37 *test case* yang didasarkan dari 20 kebutuhan fungsional yang terdapat pada sistem. Seluruh *test case* tersebut menghasilkan keluaran yang diharapkan, yang menandakan bahwa pengujian validasi memberikan hasil berupa nilai *valid* sebesar 100%.

9. DAFTAR REFRENSI

- Huffman, E. K., 1994. *Health Information Management*. Brewyn: Physicians Record Company.
- Peraturan Menteri Kesehatan Republik Indonesia nomor 269 tahun 2008 Tentang Rekam Medis. Jakarta: Kementrian Kesehatan Republik Indonesia.