Infotekmesin
Vol.14, No.01, Januari 2023
p-ISSN: 2087-1627, e-ISSN: 2685-9858
DOI: 10.35970/infotekmesin.v14i1.1636, pp.43-49

Cyclomatic Complexity dan *Graph Matrix* dalam Pengujian Sistem Informasi Manajemen Rumah Sakit

Cahya Vikasari[1*]
[1]Program Studi Teknik Informatika, Politeknik Negeri Cilacap
[1]Jln. Dr. Soetomo No.1 Karangcengis Sidakaya, Kabupaten Cilacap, 53212, Indonesia
E-mail: cahyavikasari@pnc.ac.id[1]

Info Naskah:
Naskah masuk: 16 November 2022
Direvisi: 29 Desember 2022
Diterima: 29 Desember 2022

Abstrak

Sistem informasi dapat membantu manajemen rumah sakit dalam meningkatkan pelayanan di sebuah rumah sakit. Metode pengembangan perangkat lunak yang dilakukan dalam pembuatan aplikasi manajemen rumah sakit yaitu menggunakan metode *waterfall*, salah satu tahap *waterfall* adalah pengujian. Masalah yang ada dalam pengembangan sistem yaitu proses pengujian yang tidak maksimal dan lebih fokus kepada pembangunan sistem sehingga bisa ditemukan kesalahan pada tahap implementasi. Pengujian sistem merupakan perizinan terhadap sebuah kesalahan di dalam sistem yang dikembangkan. Tujuan penelitian yaitu melaksanakan pengecekan detail desain dan menggunakan struktur control desain program secara procedural serta menjalankan program untuk menemukan *error*. Metode yang digunakan untuk melakukan tahap pengujian dalam sistem manajemen rumah sakit yaitu menggunakan metode whitebox dengan menggunakan pendekatan *cyclomatic complexity* dan *graph matriks*. Pengujian akan dilakukan pada proses pengelolaan data rawat inap di aplikasinya. Hasil penelitian dari pengujian sistem manajemen rumah sakit dengan pendekatan *cyclomatic complexity* yaitu dapat memberikan informasi jumlah pengujian minimal dan memastikan semua jalur stuktur program minimal digunakan satu kali. Selain itu, hasil pengujian *graph matriks* merupakan sisi probabilitas yang akan dilakukan.

Keywords:
whitebox;
graph matrix;
cyclomatic.

Abstract

Information systems can assist hospital management in improving services at a hospital. The software development method used in making hospital management applications is using the waterfall method which one of the stages is testing. The problem in system development is that the testing process is not optimal and focuses more on system development so that errors can be found at the implementation stage. System testing is a permit for an error in the system being developed. The aim of the research is to check the design details, use the program design control structure procedurally and run the program to find errors. The method applied to carry out the testing phase in the hospital management system is the Whitebox method using the cyclomatic complexity and graph matrix approaches. Testing will be carried out on the process of managing inpatient data in the application. The results of the research from testing the hospital management system using the cyclomatic complexity approach are that it can provide information on the minimum number of tests and ensure that all program structure paths are used at least once. In addition, the results of the graph matrix test are the probability side that will be carried out.

***Penulis korespondensi:**
Cahya Vikasari
E-mail: cahyavikasari@pnc.ac.id

1. Pendahuluan

Teknologi yang diterapkan dalam rumah sakit akan mendukung kegiatan rumah sakit tersebut antara lain meningkatkan produktivitas pegawai, meningkatkan relationship pihak rumah sakit dengan pelanggan yaitu pasien, serta dapat mengembangkan aplikasi strategi baru dalam hal pelayanan. Pelayanan di rumah sakit atau klinik kesehatan idealnya lebih bersahabat dengan pasien dan keluarganya dalam hal pelayanan medis yaitu lebih cepat dan akurat. Salah satu fokus untuk menjaga kualitas pelayanan sebuah rumah sakit adalah bagian jasa (*Service Quality*), Ketanggapan (*Responsiveness*) dalam memberikan informasi, Jaminan dan Kepastian (*Assurance*) dari data yang diinfokan[1], terutama terkait dengan informasi penggunaan kamar rawat inap oleh pasien. Teknologi dapat membantu meningkatkan kualitas pelayanan sebuah rumah sakit.

Kemajuan teknologi mengakibatkan masyarakat yang menguasai informasi juga semakin bertambah, dan berimbas masyarakat dapat memilih dan menuntut mendapatkan pelayanan kesehatan yang berkualitas dari sebuah rumah sakit[2]. Sistem dapat diimplementasikan pada sektor pelayanan terhadap pasien yaitu pada proses pengelolaan data rawat inap sehingga dapat meningkatkan pelayanan kebutuhan rawat inap di rumah sakit kesehatan untuk pasien dengan lebih cepat.

Aplikasi manajemen rumah sakit dalam pengelolaan rawat inap dibuat dengan metode pengembangan perangkat lunak waterfall, tahapannya yaitu analisis, desain, koding, testing dan *maintenance*[3]. Metode ini dilakukan dengan pendekatan secara sistematis dan terurut [4].Tahapan *coding* aplikasi yang telah dibuat untuk mendukung pelayanan rumah sakit dalam hal pengelolaan data rawat inap selanjutnya dilakukan pengujian sistem untuk menjaga mutu perangkat lunak dengan tujuan memastikan tidak terjadi *error* atau kesalahan maupun *bug*[5].

Masalah yang ada dalam pengembangan sistem yaitu proses pengujian yang tidak maksimal dan lebih fokus kepada pembangunan sistem sehingga bisa ditemukan kesalahan pada tahap implementasi. Pengujian sistem merupakan perizinan terhadap sebuah kesalahan di dalam sistem yang dikembangkan. Tahap pengujian merupakan elemen kritis atau sangat penting dalam menjamin kualitas dari sebuah sistem dan dapat memperlihatkan tidaknormalan pada tahap pengembangan sistem manajemen rumah sakit yang dibuat.

Selama tahap awal analisa kebutuhan dan tahap pembangunan, pengembang perangkat lunak dalam mengembangkan sistem informasi manajemen rumah sakit telah berusaha membangun aplikasi dimulai dari konsep abstrak sampai dengan implementasi. Namun sebelum dilakukan implementasi perlu dilakukan tahap pengujian terhadap perangkat lunak beserta implikasinya dimana nantinya hasil pengujian akan mengacu pada kualitas aplikasi manajemen rumah sakit. Tahap pembangunan sistem akan melibatkan rangkaian kegiatan produksi dimana bisa terjadi peluang adanya kesalahan yang dilakukan oleh manusia sangat besar juga dikarenakan adanya ketidakmampuan atau kurangnya manusia memiliki teknik komunikasi dengan baik yang dapat terjadi kesalahan[6]

Berdasarkan hal diatas struktur program yang dibuat dalam sistem manajemen rumah sakit bisa terdapat kesalahan sehingga diperlukan proses pengujian dalam memastikan sistem yang diproduksi memiliki kualitas yang baik. Pengujian sistem dapat dilaksanakan dengan metode blackbox *testing* dan whitebox *testing*[7]. Aplikasi yang mendukung pelayanan rumah sakit dilakukan pengujian dengan metode *whitebox testing* yaitu sebuah teknik dalam melakukan pengujian prosedural dan struktur untuk menghasilkan kasus uji sistem. W*hitebox* testing merupakan mekanisme yang akan melibatkan desain dan proses yang akan memverifikasi apakah kode program berfungsi seperti yang diharapkan sesuai dengan analisa sistem. Pengujian ini menyediakan struktur internal dari sistem atau komponen serta memberikan visibilitas penuh kepada kode dan logika program produk perangkat lunak. *Whitebox testing* digunakan untuk meningkatkan Keamanan sistem, melakukan pengecekan terhadap aliran masukan dan keluaran sistem, juga dapat digunakan untuk meningkatkan fungsi dan desain [8]. Teknik yang dilakukan dalam pengujian, seperti *Data Flow Testing*, *Basic Path* / *Path Testing*, *Loop Testing* dan *Control Flow Testing* [9].

Penelitian sebelumnya oleh Mubarak (2020) membahas mengenai teknik pengujian yang dilakukan untuk testing aplikasi yang telah diproduksi dan juga melakukan proses *Quality Assurance* terhadap sistem sebelum sistem tersebut diimplementasikan di tempat studi kasus[10]. Pengujian struktural perangkat lunak dapat dilakukan dengan teknik pengujian Kotak Hitam , Kotak Putih, dan Abu-abu. Penelitian yang dilakukan oleh Syaikhuddin menggunakan *Whitebox Testing* dalam menguji aplikasi yang sederhana. Pengujian dilakukan dengan teknik pengujian berdasarkan pengujian terhadap jalur yaitu pengembangan diagram alir. Oleh karena itu, proses pengujian menggunakan *Whitebox* Metode dengan teknik pengujian basis path dapat dijalankan [11].

Penelitian oleh Setiawan (2017) membahas tentang bagaimana mengetahui kualitas dari sebuah *software* maka yang perlu dilakukan yaitu dengan melakukan pengujianyang akan memvalidasi dan memverifikasi serta merupakan elemen yang sangat kritis dalam kegiatan pengembangan aplikasi. Hasil penelitian memperlihatkan metode whitebox yang digunakan dapat menggambarkan struktur kode program menyeluruh dan juga dapat memperlihatkan *error* kesalahan pada kode program[2].

Perbedaan dengan penelitian sebelumnya yaitu pengujian dilakukan dengan menggunakan penghitungan *cyclomatic complexity*, dan mengembangkan matriks grafik dan pengujian jalur independen dan memastikan hasil tiap metode sama sehingga dapat menjamin kualitas software dalam meningkatkan pelayanan rumah sakit.

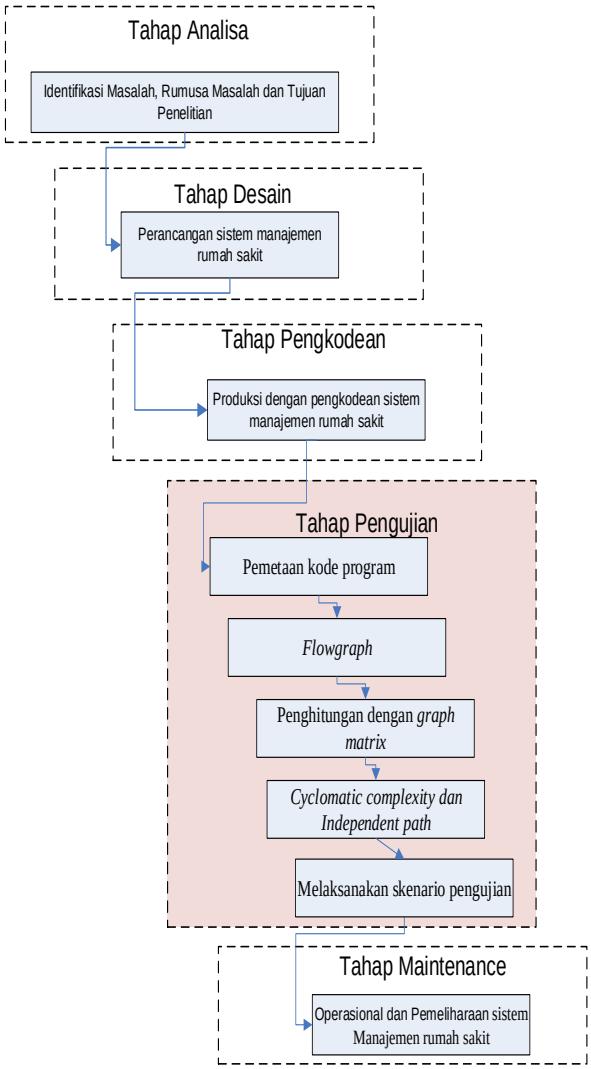
Tujuan penelitian yaitu melaksanakan pengujian terhadap aplikasi manajemen di sebuah rumah sakit yaitu pada form penggunaan kamar rawat inap dengan melaksanakan pengecekan detail desain dan menggunakan struktur control desain program secara procedural untuk mengetahui cara kerja dari aplikasi secara internal dan memastikan semua statemen kode program telah dijalankan paling tidak sekali pada saat dilakukan pengujian dan pada

saat pengujian juga memastikan bahwa semua kondisi logis sudah diujikan

2. Metode

Penelitian dilaksanakan untuk melakukan pengujian terhadap sitem informasi manajemen rumah sakit dengan menggunakan metode *whitebox* testing graph matrik dan *cyclomatic complexity* yang akan dilakukan terhadap sistem informasi manajemen rumah sakit dengan melibatkan kode, desain internal, struktur kode program.

Berikut gambaran tahapan dalam pengembangan sistem manajemen rumah sakit dengan metode waterfall namun fokus penelitian yang akan dilakukan yaitu pada tahap pengujian. yaitu melakukan pemetaan kode program, membuat *flowgraph*, penghitungan dengan *graph matrix*, *cyclomatic complexity*, *independent path* dan melaksanakan skenario pengujian. Tahapan pengembangan sistem yang akan dilakukan oleh penulis di gambarkan pada gambar 1.



Gambar 1. Tahapan pengembangan sistem

Graph matrix merupakan matrik segi empat sama sisi, mempunyai jumlah kolom dan baris yang sama dengan jumlah node yang didapatkan dalam pemetaan kode program, isi dari data merupakan keberadaan penghubung antar node. Beberapa properti ditambahkan dalam pengujian dengan graph matrix sebagai pembobotan pada hubungan antar node, sebagai berikut:

a. Jalur (*Edge*) yang pada saat dilakukan pengujian kemungkinan akan dieksekusi.
b. Waktu yang digunakan dalam memproses pengujian yang diharapkan ada pada *edge* selama proses pengujian yaitu saat transfer dilakukan.
c. Memori yang dibutuhkan selama proses transfer lakukan pada *edge*.
d. Resources yang dibutuhkan dalam proses transfer pada *edge*[12].

Cyclomatic complexity merupakan tahap untuk melakukan pengukuran terhadap aplikasi dengan memberikan hasil pengukuran secara kuantitatif kompleksitas terhadap logika dari program[9]. Nilai yang dihitung akan menentukan dari jumlah *edge-edge* secara independen dalam basis program yang diuji dengan memberikan informasi jumlah minimal tes yang nantinya dikerjakan untuk memastikan semua kode program telah dilewati dalam pengujian minimal sekali[12]. *Cyclomatic complexity* dipergunakan dalam menemukan jumlah jalur dalam *flowgraph* dengan menggunakan persamaan (1).

Jumlah yang dihasilkan dari region grafik alir disesuaikan dengan *cyclomatic complexity*. *Cyclomatix complexity* V(G) dari grafik alir dilakukan penghitungan dengan menggunakan rumus:

$$V(G)=E-N+2 \quad (1)$$

Keterangan :
E merupakan jumlah jalur (anak panah) yang ada pada grafik alir (*flowgraph*).
N merupakan jumlah node (titik) yang ada pada grafik alir (*flowgraph*).

Cyclomatix complexity V(G) juga dapat dilakukan penghitungan dengan persamaan (2).

$$V(G) = P + 1 \quad (2)$$

Keterangan :
P merupakan jumlah *predicate node* yang ada pada grafik alir.

Banyak fungsi dan modul dalam sistem yang diuji yaitu aplikasi manajemen rumah sakit dalam mendukung pelayanan yang dibuat disesuaikan dengan *requirement* Dalam penelitian yang dikerjakan yaitu fokus pada tahap pengujian sistem struktur program penggunaan kamar yang digunakan untuk menangani jika ada pasien yang akan menggunakan kamar untuk rawat inap di rumah sakit. Berikut tampilan dari form penggunaan kamar pada sistem pada Gambar 2.

Gambar 2. Tampilan Penggunaan kamar pada sistem.

3. Hasil dan Pembahasan

Penelitian dilaksanakan untuk melakukan pengujian terhadap sitem informasi manajemen rumah sakit pada fungsi atau form penggunaan kamar pada gambar 1. Langkah-langkah pengujian yang akan dilakukan adalah melakukan pemetaan kode program, membuat *flowgraph*, penghitungan dengan *graph matrix, cyclomatic complexity, independent path* dan melaksanakan skenario pengujian. Pengujian pertama menggunakan pengujian *whitebox* dengan metode graph matrik. Langkah pertama yaitu pemetaan kode program dengan mengubah listing program menjadi node sesuai dengan kode program yang ada di fungsi penggunaan kamar. Langkah berikutnya membuat graph matrik dengan menghubungkan node yang saling berhubungan sesuai dengan kode program.

3.1 Pemetaan kode Program

Kode program didapatkan sesuai den form aplikasi yang akan diuji yaitu pada fungsi penggunaan kamar, setelah mendapatkan kode program lalu dilakukan pemetaan pada kode program. Berikut hasil pemetaan kode program pada form penggunaan kamar diperlihatkan pada tabel 1.

Tabel 1. Pemetaan kode program pada sistem.

Node	Kode Program

1	`private void BtnSimpan_Click(object sender, EventArgs e)`
	`Simpan();`
	`}`

2	`void Simpan()`
3	`{` `if (!(TXTIDpenggunaan.Text.Equals("") || TXTIDpasien.Text.Equals("") || TXTNmpasien.Text.Equals("") || TXTPenyakit.Text.Equals("") || TXTIDkamar.Text.Equals("") || TXTNmkamar.Text.Equals("")))`
5	`{`
6	`{` `if (!peng_kamar.isExist(TXTIDpenggunaan.Text.Trim()))`
7	`{` `if (MessageBox.Show("Yakin data tersebut akan disimpan ?", "KONFIRMASI", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)`

8	`{`
	`peng_kamar.ID_Penggunaan = TXTIDpenggunaan.Text.Trim();`
	`peng_kamar.ID_Kamar = TXTIDkamar.Text.Trim();`
	`peng_kamar.ID_Pasien = TXTIDpasien.Text.Trim();`
	`peng_kamar.Tgl_Masuk = Convert.ToDateTime(lbl_date.Text.Trim());`
	`peng_kamar.SimpanPengKamar();`

9	`kamar.UbahStatusKamar(TXTIDkamar.Text.Trim());`

10	`MessageBox.Show("Data Berhasil Disimpan ! ", "Simpan Data", MessageBoxButtons.OK, MessageBoxIcon.Information);`
	`}`
	`}`

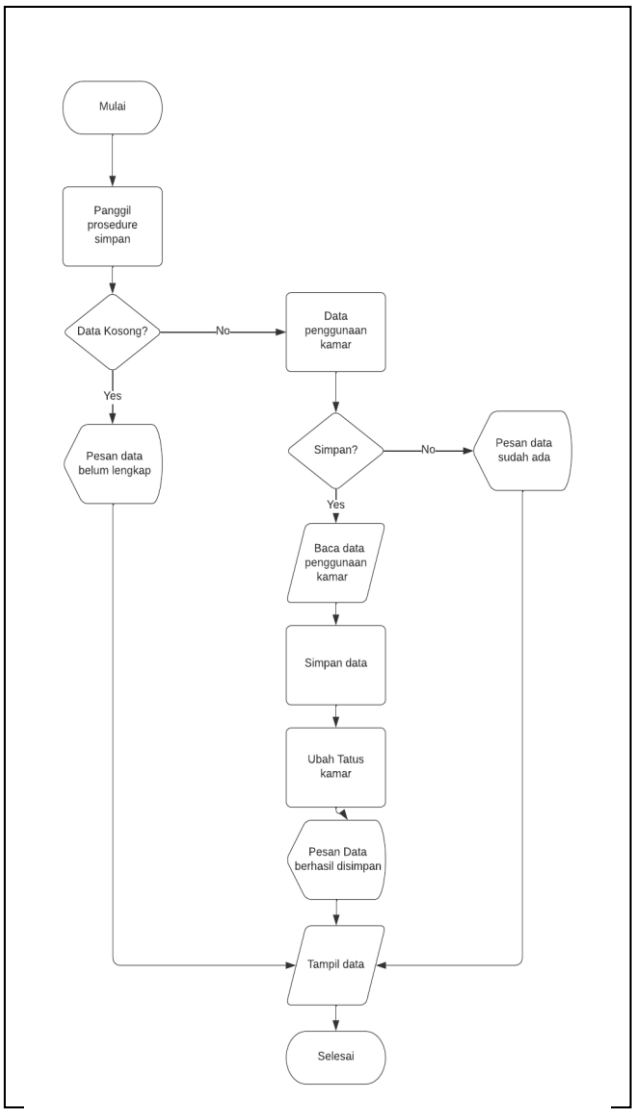
11	`Else`
	`{`
	`MessageBox.Show("Data Sudah Ada !", "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error);`
	`}`
	`}`

4	`else` `{`

Node	Kode Program
	MessageBox.Show("Data Belum Lengkap ! " + " Silahkan Lengkapi Data Terlebih Dahulu !", "Gagal", MessageBoxButtons.OK, MessageBoxIcon.Error); }
12	LoadData();
13	}

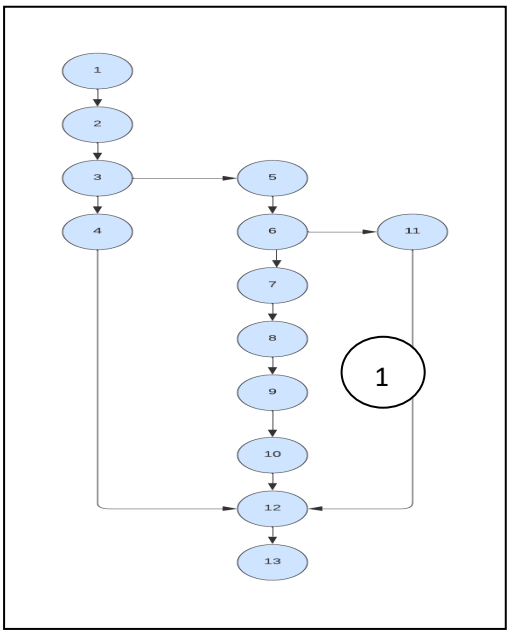
3.2 Flowchart

Flowchart penggambaran secara grafik dilakukan dengan menggunakan flowchart dari tahapan dan urutan dari prosedur suatu program, flowchart yang digunakan dalam penelitian ini yaitu program flowchart[13]. Program flowchart diperlihatkan pada gambar 3.



3.3 *Flowgraph*

Langkah berikutnya yaitu membuat *flowgraph* dari pemetaan kode program yang ada pada fungsi penggunaan kamar pasien dengan menghubungkan node-node yang saling terhubung. *Flowgraph* digambarkan pada gambar 4.



Gambar 4. Undirect graph

3.4 Penghitungan Graph Matrik

Node-node yang saling berkomunikasi sesuai dengan kode program diberikan simbol 1, dan kemudian dilakukan penjumlahan untuk setiap node *edge* nya dengan rumus n(E)-1, dan dijumlahkan hasilnya. Hasil dari perhitungan yaitu sebesar 3, proses penghitungan diperlihatkan pada tabel 2.

Tabel 2. Proses penghitungan graph matrik

Node	Connected Node													N(E)-1
	1	2	3	4	5	6	7	8	9	10	11	12	13	
1		1												1-1=0
2			1											1-1=0
3				1	1									2-1=1
4												1		1-1=0
5						1								1-1=0
6							1				1			2-1=1
7								1						1-1=0
8									1					1-1=0
9										1				1-1=0
10												1		1-1=0
11														
12													1	1-1=0
13														
	Jumlah+1													2+1=3

Graph dibuat dengan konsep undirected graph, sesuai dengan kode program node yang terbentuk sebanyak 13 buah dan dihubungkan sesuai pogram. Setelah *graph* terbentuk kemudian dibuat *degree* untuk melihat kemungkinan jalur (*Edge*) akan dilalui / dieksekusi. Hasil yang diperoleh yaitu sebanyak 3 jalur yang akan dilalui.

3.5 Cyclomatic Complexity

Pengujian Selanjutnya yaitu menggunakan teknik cyclomatic complexity menggunakan persamaan (1) diperoleh :

V(G) = E – N + 2P
= 14-13 +2.1 = 3

Menggunakan persamaan (2) diperoleh :
V(G) = P + 1
= 2 + 1 = 3

3.5.1 Independent Path

Independent path sesuai dengan penghitungan *cyclomatic complexity* yaitu :

1. Jalur 1 : 1 – 2 – 3 – 4 – 12 – 13
   (Skenario data belum lengkap)
2. Jalur 2 : 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 -10 – 12 – 13
   (Skenario data tersimpan)
3. Jalur 3 : 1 – 2 – 3 – 5 – 6 – 11 – 12 – 13
   (Skenario data sudah ada)

Kegiatan pengujian pada form penggunaan kamar pada aplikasi manajemen rumah sakit dengan serangkaian kegiatan yang dilakukan untuk memastikan semua statemen yang dibuat telah dijalankan paling tidak sekali selama dilakukan pengujiandan juga memastikan bahwa semua kondisi logis telah diuji dapat dilihat pada jalur path yang telah dihasilkan yaitu sebayak 3 jalur path dan semua node minimal satu kali terlalui dalam pengujian.

Program yang memiliki hasil cyclomatic complexity yang lebih besar dari 10 maka program dalam sistem memiliki angka probabilitas yang lebih tinggi terhadap error program. Jika hasil yang didapat dari pengujian *cyclomatic complexity* sebuah sistem semakin tinggi maka akan meningkatkan kesalahan atau *error* dan pemeliharaannya[14]. Hasil dari penghitungan *cyclomatic complexity* sistem informasi manajemen rumah sakit pada proses pengolahan rawat inap yaitu sebanyak 3 dapat disimpulkan bahwa sistem bebas dari resiko sesuai dengan *software engineering institute* yang mengklasifikasikan hasil pengujian dalam evaluasi dari resiko yang disesuaikan dengan hasil pengujian *cyclomatic complexity* dapat dilihat pada tabel 3.

Tabel 3. Evaluasi *cyclomatic complexity*

Hasil Cyclomatic Complexity	Evaluasi Resiko
Nilai 1-10	Bebas dari risiko
Nilai 11-20	Resiko sedang
Nilai 21-50	Resiko sedang
Nilai >51	Sangat beresiko

3.5 Skenario Pengujian

Seorang pengembang perangkat lunak dapat membuat rancangan skenario pengujian yang dapat melakukan verifikasi validitas struktur data internal, melatih kondisi logis yaitu benar dan salahnya, mencoba jalur independen yang dapat dilalui, menjalankan *loop* pada batasan operasionalnya[15]. Pengujian dengan metode *whitebox* pada sistem informasi manajemen rumah sakit pada fungsi simpan penggunaan kamar pasien yaitu sebanyak 3 path, sesuai dengan pengujian *graph matrix* dan *cyclomatic complexity*.berikut skenario yang akan dilakukan :

1. Memasukan data yang belum lengkap sesuai dengan isian yang ada pada tampilan penggunaan kamar dan mengeklik tombol simpan.
2. Memasukan data yang sudah ada atau sudah pernah disimpan sebelumnya.
3. Mengisikan data dengan lengkap dan belum pernah diisi sebelumnya.

4. Kesimpulan

Pengujian terhadap aplikasi yang mendukung manajemen rumah sakit pada form penggunaan kamar pasien telah dilakukan dengan menggunakan metode *whitebox* dengan melaksanakan beberapa tahap yaitu melakukan pemetaan kode program, membuat *flowgraph*, penghitungan dengan *graph matrix, cyclomatic complexity, independent path* dan melaksanakan skenario pengujian. Hasil dari pengujian *graph matrix dan complexity* yaitu sama sebesar 3 begitu juga dengan independent path yang terbentuk. Evaluasi dari *cyclomatic complexity* yaitu sistem informasi manajemen rumah sakit yang diuji bebas dari resiko. Sehingga setelah proses testing dapat dilanjutkan untuk proses implementasi dan memastikan bahwa pengujian telah dilakukan untuk mengecek tingkat *error* sistem dan memastikan bahwa kode program telah dieksekusi minimal satu kali serta memastikan semua kondisi logis telah diuji. Berdasarkan hasil pengujian menggunakan *graph matrix, cyclomatic complexity* dan *independen path* menghasilkan jumlah yang sama yaitu 3, maka dapat ditarik kesimpulan bahwa sistem manajemen rumah sakit pada form penggunaan kamar sudah bebar kesalahan dan sesuai dengan logika yang ada pada kode program.

Ucapan Terimakasih

Penulis mengucapkan terima kasih kepada Politeknik Negeri Cilacap sesuai dengan Kontrak Penelitian Tahun Anggaran 2022.

Daftar Pustaka

[1] Abdurahman, Junaidi, and Aminuyati, “Analisis Kualitas Pelayanan Jasa Kesehatan (Pada Pasien Rawat Inap Rumah Sakit Pendidikan Universitas Tanjungpura Pontianak),” *J. Pendidik. dan Pembelajaran Khatulistiwa*, vol. 6, no. 2, pp. 1–22, 2017, [Online]. Available: https://www.semanticscholar.org/paper/ANALISIS-KUALITAS-PELAYANAN-JASA-KESEHATAN-(PADA-Abdurahman-Junaidi/cbc3defcab2e923da685fcefe9b00ad8fa9ae1cf

[2] R. Setiawan, “Pengujian Perangkat Lunak,” *Semin. Nas. Inform. dan Apl.*, no. September, pp. 37–39, 2017, [Online].

Available: https://journal.stekom.ac.id/
[3] D. S. Purnia, A. Rifai, and S. Rahmatullah, "Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android," *Semin. Nas. Sains dan Teknol. 2019*, pp. 1–7, 2019.
[4] Novianita, "Analisa Sistem Informasi Geografis Pemetaan Rumah Sakit Di Kota Samarinda Berbasis Web," *Jitac*, vol. 1, no. 1, p. 2020, 2020.
[5] N. W. Rahadi and C. Vikasari, "Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions," *Infotekmesin*, vol. 11, no. 1, pp. 57–61, 2020, doi: 10.35970/infotekmesin.v11i1.124.
[6] R. Ama, *E-Book Implementasi dan Pengujian Sistem*. Indragiri: Sistem Informasi UNISI, 2020.
[7] J. B. L. Sie, I. A. Musdar, S. Bahri, and T. Informatika, "PENGUJIAN WHITE BOX TESTING TERHADAP WEBSITE ROOM MENGGUNAKAN TEKNIK BASIS PATH," *J. Kharisma Tech*, vol. 17, no. 02, pp. 45–57, 2022.
[8] D. Y. Annisa Octaviana Nurshanty, Aldo Saputra, Farhan Rifanto Hardjanto, Muhammad Bio Franklyn, "Teknik Dalam White-box dan Black-box Testing," *Binur Univercity*, 2020. https://socs.binus.ac.id/2020/07/02/teknik-dalam-white-box-dan-black-box-testing/ (accessed Dec. 01, 2022).
[9] M. F. Londjo, "Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Login," *J. Siliwaangi*, vol. 7, no. 2, pp. 35–40, 2021.
[10] R. Mubarak, "Implementasi Metode White Box Testing Pada Proses Quality Assurance Perangkat Lunak Berbasis Web Dan Mobile Collection System," *J. Teknol. Inf. ESIT*, vol. XV, no. 10, pp. 57–63, 2020.
[11] M. M. Syaikhuddin, C. Anam, A. R. Rinaldi, and M. E. B. Conoras, "Conventional Software Testing Using White Box Method," *Kinetik*, vol. 3, no. 1, p. 67, 2018, doi: 10.22219/kinetik.v3i1.231.
[12] dosenpendidikan, "White Box Testing," 2020.
[13] I. Budiman, S. Saori, R. N. Anwar, Fitriani, and M. Y. Pangestu, "ANALISIS PENGENDALIAN MUTU DI BIDANG INDUSTRI MAKANAN (Studi Kasus: UMKM Mochi Kaswari Lampion Kota Sukabumi)," *J. Inov. Penelit.*, vol. 1, no. 10, pp. 1–15, 2021.
[14] F. N. Pranata, F. Pradana, and T. A. Kurniawan, "Pengembangan Sistem Perhitungan Komplesitas Kode Sumber Berdasarkan Metrik Halstead dan Cyclomatic Complexity," *Pros. Semin. Nas. Teknol. dan rekayasa Inf.*, pp. 27–35, 2016.
[15] S. Patel, V. Pratap, and Katiyar, "Original Research Paper Computer Science WHITE-BOX TESTING TECHNIQUE FOR FINDING DEFECTS," *Glob. J. Res. Anal.*, vol. 8, no. 7, pp. 83–85, 2019.