SNESTIK

Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika

https://ejurnal.itats.ac.id/snestik dan https://snestik.itats.ac.id

Informasi Pelaksanaan :
SNESTIK I - Surabaya, 26 Juni 2021
Ruang Seminar Gedung A, Kampus Institut Teknologi Adhi Tama Surabaya

Informasi Artikel:
DOI : 10.31284/p.snestik.2021.1821

Prosiding ISSN 2775-5126

Fakultas Teknik Elektro dan Teknologi Informasi-Institut Teknologi Adhi Tama Surabaya
Gedung A-ITATS, Jl. Arief Rachman Hakim 100 Surabaya 60117 Telp. (031) 5945043
Email : snestik@itats.ac.id

Implementasi Model Incremental Pada Sistem Informasi Klinik Nurani Jaya Berbasis Desktop

Taufan Augusta[1], Moh. Ajiemas Pangestu[2], Diki Adita Mara[3], Tutuk Indriyani[4]
Jurusan Teknik Informatika, Institut Teknologi Adhi Tama Surabaya[1,2,3,4]
e-mail: taufan.augusta@protonmail.com

ABSTRACT

Clinics are public service agency serve in health services. Theerefore it is necessary to have an information system that can support the performance of health clinic services. There are still medical clinics that do medical records with conventional media. The use of paper in recording disease history is not only slow but also prone to human error and interferes with the smooth process of health services. This study aims to build a medical record information system with an incremental desktop-based model to improve the quality of health clinic services.

Keywords: *Incremental Model; Information System; Design Pattern; SNESTIK.*

ABSTRAK

Klinik merupakan sebuah instansi layanan publik yang bergerak di bidang jasa kesehatan. Oleh sebab itu perlu adanya suatu sistem informasi yang dapat menunjang kinerja pelayanan klinik kesehatan. Masih terdapat klinik medis yang melakukan rekam medis dengan media konvensional. Penggunaan kertas dalam mencatat riwayat penyakit tidak hanya lambat namun juga rawan dengan *human error* dan mengganggu kelancaran proses pelayanan kesehatan. Penelitian ini bertujuan untuk membangun sistem informasi rekam medis dengan incremental model berbasis desktop untuk meningkatkan mutu pelayanan klinik kesehatan.

Kata kunci: Incremental Model; Sistem Informasi; Design Pattern; Rekam Medis; SNESTIK.

PENDAHULUAN

Dalam UU nomor 25 tahun 2009 ayat 1 pasal 7 telah diatur bahwasanya standar pelayanan adalah tolok ukur yang dipergunakan sebagai pedoman penyelenggaraan pelayanan dan acuan penilaian kualitas pelayanan sebagai kewajiban dan janji penyelenggara kepada masyarakat dalam rangka pelayanan yang berkualitas, cepat, mudah, terjangkau, dan terukur [1]. Klinik merupakan sebuah instansi layanan publik yang bergerak di bidang jasa kesehatan. Di dalam suatu instansi pelayanan kesehatan, seperti halnya klinik, rekam medis merupakan sebuah standar yang harus memiliki ketepatan dan kecepatan di dalam proses pemeriksaan. Rekam medis setidaknya harus mencantumkan: tanggal dan waktu, hasil pemeriksaan, diagnosis, rencana pelaksanaan dan obat yang diberikan, seperti yang telah diatur pada Permenkes nomor 269 tahun 2008 tentang rekam medis [2].

Klinik Nurani Jaya yang berada di Kecamatan Kutisari, Kota Surabaya merupakan sebuah klinik yang berada di bawah naugan Badan Penyelengara Jaminan Sosial Kesehatan (BPJS). Klink tersebut menyediakan dua pelayanan, yaitu pemeriksaan kesehatan dan pembuatan obat berdasarkan hasil dari pemerikasaan medis. Berdasarkan hasil observasi yang dilakukan di Klinik Nurani Jaya, ditemukan beberapa permasalahan yang berkaitan dengan mutu pelayanan. Yaitu, proses pencatatan rekam medis masih menggunakan kartu pasien. Kartu ini disimpan di dalam lemari penyimpanan yang nantinya akan dibawa oleh pasien yang berkunjung, lalu diberikan kepada dokter dan pihak apoteker. Dalam pelayanannya, setidaknya untuk satu orang pasien membutuhkan waktu 10 hingga 15 menit mulai dari pemeriksaan medis hingga pengambilan obat di loket apotek. Tidak jarang kartu pasien ini terselip dan hilang. Dalam beberapa kasus apoteker bahwa harus harus keluar dari ruangannya itu menuju ke ruang dokter (yang mana juga mengganggu proses pemeriksaan pasien) untuk menyelesaikan proses rekomendasi pengobatan karena obat yang sebelumnya dirujuk oleh dokter sedang kosong. Kendala-kendala tersebut mengakibatkan proses pelayanan kesehatan Klinik Nurani Jaya menjadi tidak efektif, sering pula disebabkan banyak pasien yang berkunjung dalam satu hari, klinik ini harus melayani pasien melebihi jam operasionalnya.

Sebelumnya telah dilakukan penelitian serupa, "Penerapan Konsep Model View Controller Pada Rancang Bangun Sistem Infromasi Klinik Kesehatan Berhasis Web" [3] dan "Pengembangan Sistem Informasi Klinik Kesehatan Ganesha Husada Menggunakan Metode SDLC" [4]. Di dalam dua peneltian tersebut hanya dibahas proses pengembangan berdasarkan kebutuhan *client* (dalam hal ini klinik) tanpa melihat segi kebutuhan pengembang dalam membuat aplikasi sistem informasi klinik.

Berdasarkan permasalahan yang dimiliki oleh Klinik Nurani Jaya dalam mutu pelayanan pencatatan rekam medis, maka penelitian bertujuan untuk mengembangkan sistem informasi pencatatan dan penyimpanan data rekam medis yang diharapkan akan meningkatkan kualitas pelayanan klinik tersebut. Serta menjamin kualitas "*back-end*" atau *code* dari aplikasi yang dibuat.

Dalam penelitian ini, proses pengembangan sistem informasi rekam medis akan menganut standar metode *Software Development Life Cycle (SDLC)* yaitu dengan menggunakan pemodelan *Incremental*. SDLC merupakan sebuah rangkaian proses yang ditujuan untuk membangun sebuah perangkat lunak. Tujuan dari SDLC selain untuk meningkatkan kualitas produk perangkat lunak adalah juga untuk mengetahui dan mengestimasi *scope* dan biaya yang dibutuhkan untuk membangun sebuah perangkat lunak [5]. Berikut yang termasuk dalam tahapan SDLC yaitu: *planning*, *requirement analysis*, *designing*, *building*, *testing*, *deployment*, dan *maintenace* [6]. Salah satu pemodelan dalam SDLC adalah model *Incremental* yang merupakan penyembangan dari model *waterfall* di mana dalam pengembangannya, sebuah perangkat lunak dibagi menjadi beberapa iterasi dengan langkah pengembangan: *requirement, design, programming, testing* dan *maintenance*. Model *Incremental* dipilih karena memiliki keunggulan yaitu: mudahnya proses pengujian dan *debugging*, kecilnya kemungkinan kegagalan proyek, dan dapat dengan cepat menghasilkan perangkat lunak sesuai dengan kebutuhan [7]. Pada awal pengembangan akan dilakukan proses analisis kebutuhan and berlanjut ket proses desain, pemrograman, dan pengujian.

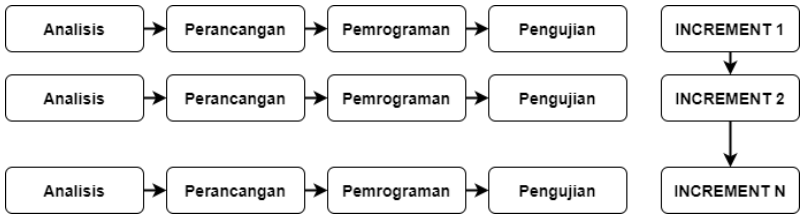
Tahap ini merupakan iterasi pertama, di tiap tahapannya akan didapatkan *feedback* dari *client* yang mana bisa terjadi perubahan terhadap desain yang telah dibuat dan proses pengembangan akan berlanjut ke iterasi berikutnya. Di sisi lain perlunya proses perencanan dan desain yang mendetail dan pemahaman yang mendalam dari keseluruahan sistem adalah kekurangan dari model *Incremental.* Sistem akan dirancang dengan *design pattern composite*, untuk menghindari bentuk *antipattern*. *Antipattern*, merupakan sebuah konsep di mana developer hanya memikirkan apakah program yang dibuat dapat berjalan tanpa memperhitungkan alur atau hirarki pemrograman. [8] Dengan menerapkan *design pattern (composite)*, SDLC ikut mempertimbangakan kebutuhan pengembang di mana akan memudahkan proses perawatan perangkat lunak. *Design pattern composite* merupakan pola arsitektur *object-oriented programming* di mana suatu *object* akan hanya memiliki satu tugas atau tanggung jawab dalam suatu system. [9]

Untuk perancangan sistem, peniliti akan menggunakan *Unified Modelling Language (UML)* sebagai bahasa standar untuk menspesifikasikan dan membangun perangkat lunak serta untuk mendokumentasikan proyek dan pemodelan *design pattern*. Sedangkan untuk tahap *testing* atau tahap uji akan menggunakan *Gray Box Testing*. Kedua metode pengujian akan dilakukan untuk menilai fungsionalias perangkat lunak berdasarkan tahap pengembangan [10].

. Sedangkan untuk tahap *testing* atau tahap uji akan menggunakan *Gray Box Testing*, *Gray Box Testing* dipilih karena dalam proses pengujian perangkat lunak terbagi menjadi dua, *White Box Testing* pada increment satu dan dua, dan *Black Box Testing* pada increment ketiga, tahap in befokus pada fungsionalitas suatu perangkat lunak tanpa *tester* perlu mengetahui teknikalitas di balik perangkat lunak yang diuji.

METODE

Proses membuatan sistem informasi rekam medis dengan metode SDLC *Incremental* model dimulai dengan tahan *requirement* atau analisis kebutuhan. Proses ini lakukan dua metode yaitu observasi dan wawancara di Klinik Nurani Jaya.



Gambar 1. Model Incremental

Tahap Wawancara

Setelah dilakukan wawancara selama 2 (dua) hari kepada penanggung jawab Klinik Nurani Jaya dan meninjau secara langsung proses pelayanannya. Dapat ditarik kesimpulan bahwa sistem informasi rekam medis masih menggunakan metode konvensional dengan menggunakan kartu pasien. Kartu pasien itu sering menyebabkan *human error* seperti terselip dan hilang, hingga terjadinya miskomunikasi antara dokter dan pihak apoteker setempat terkait ketersedian obat. Solusi yang peneliti tawarkan adalah dengan merancang sistem informasi rekam medis berbasis *desktop* yang terhubung dengan *Database*.

Tahap Analisis Kebutuhan

Pada tahap ini kebutuhan fungsional akan dibagi menjadi 2 (dua) yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Untuk kebutuhan fungsional, berdasarkan standar alur

pelayanan pada Klinik Nurani Jaya, tedapat tiga kebutuhan fungsionalitas untuk tiga jenis pegawai: resepsionis, dokter, dan apoteker. Adapun kebutuhan fungsional dan non-fungsional adalah sebagai beriktu:

Kebutuhan Fungsional

1. Resepsionis dapat mendaftarkan pasien baru.
2. Resepsionis dapat mencari pasien yang telah terdaftar.
3. Resepsionis dapat memasukan pasien ke dalam antrean.
4. Dokter dapat melihat daftar antrean pasien.
5. Dokter dapat melakukan *input* data diagnosis terhadap pasien.
6. Dokter dapat melihat riwayat penyakit pasien.
7. Dokter dapat melakukan *input* data resep obat.
8. Dokter dapat melihat data obat yang tersedia.
9. Apoteker dapat melihat daftar antrean pasien.
10. Apoteker dapat mengubah data obat.

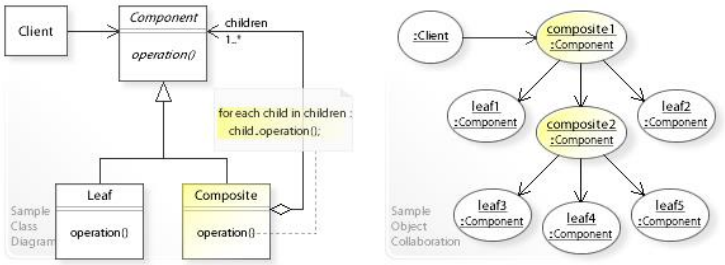
Kebutuhan Non-Fungsional

1. Sistem memiliki *Graphic User Interface* yang *User-Friendly.*
2. Aplikasi dapat digunakan diberbagai sistem operasi yang mendukung Java.

Tahap Perancangan Sistem

Terdapat tiga pengguna yaitu resepsionis, dokter, dan apoteker. Program akan dibagi menjadi tiga berdasarkan pengguna dan kebutuhan. Pada tahap increment pertama, peneliti perlu memecah komponen-komponen yang akan digunakan oleh tiap *object* pengguna. Yang mana tiap komponen dapat digunakan ulang (*reusable object*) oleh program user yang lain. Pada increment pertama. Komponen yang diperlukan di antarnya:

1. User Panel
   Merupakan *object* panel atau *user-interface* dimana *object* lain dapat ditampilkan kepada *user*.
2. Table Viewer dan Table Controller
   Tampilan (table abstract) untuk *object* pada user panel serta fungsionalistas lainnya seperti *select row* dan *refresh*.
3. Database Accessor dan Query Builder
   Object yang memiliki fungsi untuk mengakses database hanya dengan memberikan parameter secara abstrak.

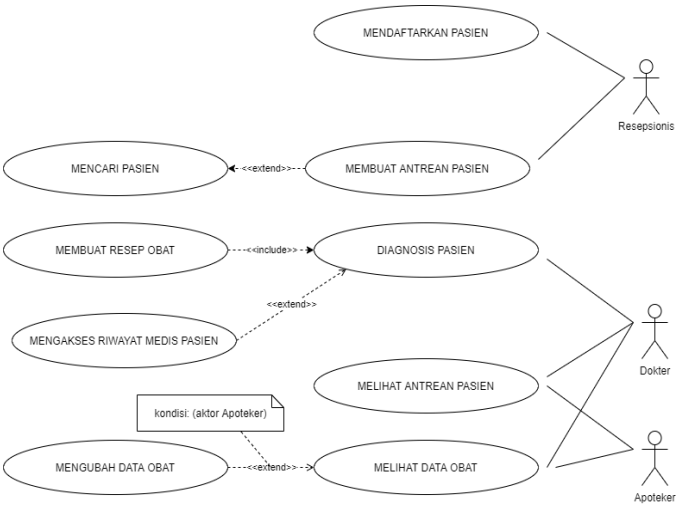


Gambar 2.Bentuk Composite Design Pattern

Pada *increment* kedua, perangkat lunak telah memiliki kerangka dasar. Proses berikutnya adalah menambahkan *object* yang dapat digunakan oleh *object* atau komponen yang telah dibuat pada increment pertama, seperti *object* pasien, obat, dan riwayat medis.

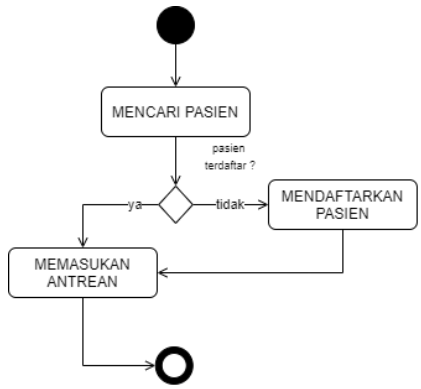
Pada *increment* ketiga, meerupakan tahap refaktorisasi untuk memastikan bahwa semua *object* dapat digunakan dengan baik saat dipanggil di ketiga program yang ada (resepsionis, dokter, dan apoteker).

Dilampirkan perancangan fungsionalitas sistem menggunakan *use case* diagram dan perancangan aktivitas *user* dengan menggunakan *activity* diagram.



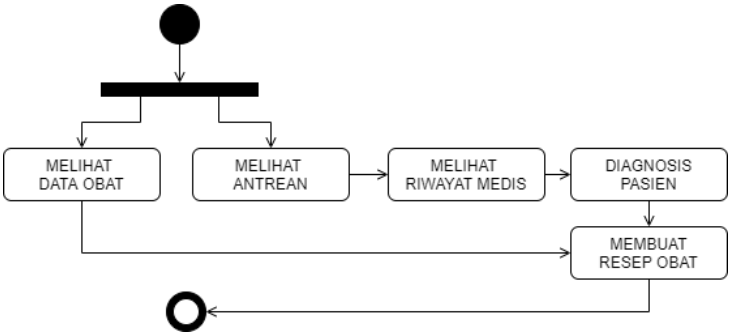
Gambar 3. Use Case Diagram Sistem Informasi Rekam Medis

Aplikasi desktop sistem informasi rekam medis Klinik Nurani Jaya memilki 3 (tiga) *user* yaitu, yaitu resepsionis, dokter, dan apoteker. Resepsionis merupakan pelayanan pertama di mana pasien yang datang berkunjung akan dimasukan di dalam antrean. Dokter and apoteker dapat melihat antrean yang tersedia. Dokter dapat melakukan diagnosis kepada pasien yang datanya akan terekam di dalam *database*, data rekam medis juga termasuk obat yang direkomendasikan dokter. Data tersebut akan langsung diterima oleh pihak apoteker ketika dokter men-*submit* diagnosis.



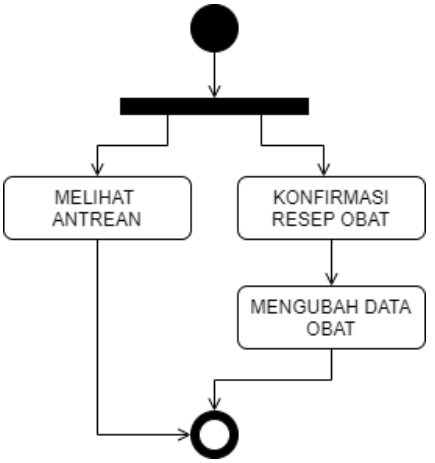
Gambar 4. Activity Diagram Resepsionis

Resepsionis merupakan pelayanan pertama di dalam klinik. Resepsionis bertugas untuk menambahkan pasien ke dalam antrean dan mendaftarkan pasien apabila belum terdaftar di dalam sistem.



Gambar 5. Activity Diagram Dokter

Dokter dapat melihat riwayat pasien yang sedang ditangani. Dengan men-*submit* hasil diagnosis data tersebut akan terekam sebagai riwayat medis. Dokter juga dapa melihat data obat yang tersedia saat membuat resep rekomendasi obat.



Gambar 6. Activity Diagram Apoteker

Proses terakhir dalam pelayanan klinik adalah apoteker. Setelah pasien diperiksa oleh dokter dan dibuatkan resep obat. Resep tersebut akan diterima langsung oleh pihak apoteker dan setelah obat itu diterima oleh pasien maka apoteker dapat melakukan konfimasi untuk mengubah data obat seperti ketersediaan obat.

Tahap Pemrograman

Pada tahap ini dilakukan pembuatan aplikasi *desktop* sistem informasi rekam medis Klinik Nurani Jaya dengan menggunakan bahasa pemrograman Java. Seperti berikut:

1. Increment 1 — Pemrograman Kerangka Aplikasi
   Tahap pembuatan sistem informasi dengan fungi-fungsi umum. Mengakses database dan menampilkan data.
2. Increment 2 — Pemrograman Object Berdasarkan Kebutuhan Sistem Informasi
   Tahap pemrograman di mana sistem informasi akan diubah untuk *spesific case* dan fungsionalitas berdasarkan tiga kebutuhan pengguna yang berbeda.
3. Increment 3 — Refaktorisasi
   Tahap refaktorisasi, memastikan semua object dan komponen program telah berjalan dengan baik.

Tahap Uji

Pada increment satu dan dua sistem akan diuji dengan metode *White Box Testing*. Pengembang menguji *object* dan komponen yang telah dibuat. Sedangkan pada increment 3, akan menggunakan metode pengujian *Black Box Testing*, akan dicari kemungkinan *error* atau *bug*. Metode pengujian bertujuan dilakukan untuk mengetahui apakah kebutuhan fungsional telah terpenuhi.

Tahap Perawatan

Tahap terakhir dalam pengembangan di mana setelah dilakukan pengujian di mana segala kebutuhan fungsional telah terpenuhi dan perangkat lunak dapat berjalan dengan baik. Tahap perawatan bertujuan untuk meningkatkan kinerja aplikasi dengan menjaga kestabilan aplikasi dan menambah fitur-fitur yang tidak termasuk di dalam analisis kebutuhan. Proses perawatan hanya perlu dilakukan pada *object* level (mengubah suatu bagian dari komponen) tanpa mengubah program secara keseluruhan.

HASIL DAN PEMBAHASAN

Tahap ini berisikan pembahasan dari tahap pengujian adalah sebagai berikut:

Proses Pengujian

Dalam pengujian *Gray Box Testing*, Tahap pengujian terbagi menjadi dua: *White Box Testing* untuk increment satu dan dua. Bertujuan untuk menguji fungsionalitas object dan komponen dari program. Dan *Black Box Testing*, menguji fungsionalitas dari segi *end-user* tanpa harus mengetahui rincian proses yang ada. Apabila suatu *requirement* telah memenuhi hasil yang diharapkan maka pengujian dinyatakan VALID.

Hasil Pengujian

- **White Box Testing**
  1. User Panel: VALID. Tiap user dapat menampilan layout panel berdasarkan jenis user.
  2. Table Viewer dan Table Controller: VALID. Panel yang memiliki table dapat menampilkan data.
  3. Database Accessor dan Query Builder: VALID. Object dapat instansiasi dan dapat berkerja secara abstrak dengan melempar suatu parameter pada *method object*.
  4. Object pasien: VALID: Object Pasien dapat dibuat dari database ke program dan sebaliknya.
  5. Object obat: VALID. Object Obat dapat dibuat dari database ke program dan sebaliknya.
  6. Object riwayat medis t: VALID. Object riwayat medis t dapat dibuat dari database ke program dan sebaliknya.
- **Black Box Testing**
  1. *User* Resepsionis: VALID. Resepsionis telah dapat memasukan pasien ke dalam antrien dan mendaftarkan pasien baru yang belum terdaftarkan di dalam sistem.
  2. *User* Dokter: VALID. Dokter dapat dengan mudah mengakses data obat dan data riwayat penyakit pasien. Serta mampu men-*submit* data diagnosis pasien.
  3. *User* Apoteker: VALID. Apoteker dapat secara *real-time* menerima data resep obat yang baru saja di-*submit* oleh dokter. Apotker juga dapat dengan mudah memantau jumlah ketersedian obat.

KESIMPULAN

Dari penelitian yang telah dilakukan, maka dapat disimpulkan sebagai berikut. Peneliti telah dapat mengembangkan aplikasi sistem informasi rekam medis berbasis *desktop* untuk Klinik Nurani Jaya. Dokter dapat dengan mudah mengakses riwayat penyakit pasien dan terjadi keselarasan antara pihak dokter dan apoteker perihal ketersedian obat saat dokter melakukan diagnosis. Dengan adanya aplikasi ini, diharapkan dapat meningkatkan mutu pelayan kesehatan klinik.

DAFTAR PUSTAKA

[1] Undang-Undang Republik Indonesia Nomor 25 Tahun 2009 Tentang Pelayanan Publik[online].Available: pn-kediri.go.id/files/Standar%20Pelayanan/UU_25_2009.pdf
[2] Menteri Kesehatan Republik Indonesia.Permenkes Nomor 269/MENKES/PER/III/2008[online].Available: ngada.org/menkes269-2008.html
[3] F. Devy, "Penerapan Konsep Model View Controller Pada Rancang Bangun Sistem Informasi Klinik Kesehatan Berbasis Web," vol. 18, no. 2, Mei 2018.
[4] B. S. Aryani, "Pengembangan Sistem Informasi Klinik Kesehatan Ganesha Husada Menggunakan Metode System Development Life Cycle," *e-Proceeding of Engineering*, vol. 5, Desember 2018.
[5] S. M. Salve, S. N. Samreen, and N. Khatri-Valmik, "A Comparative Study on Software Development Life Cycle Models," *International Research Journal of Engineering and Technology*, vol. 05, no. 02, p. 5, Feb. 2018
[6] S. Z. Iqbal and M. Idrees, "Z-SDLC Model: A New Model For Software Development Life Cycle (SDLC)," *IJEART*, vol. 3, no. 2, p. 9, Feb. 2017.
[7] M. A. Rather and Mr. V. Bhatnagar, "A COMPARATIVE STUDY OF SOFTWARE DEVELOPMENT LIFE CYCLE MODELS," *International Journal of Application or Innovation in Engineering & Management*, vol. 4, no. 10, p. 7, Oct. 2015.
[8] M. Abbes, F. Khomh, Y.-G. ̈el Gu ́eh ́eneuc, and G. Antoniol , "An Empirical Study of the Impact of TwoAntipatterns, Blob and Spaghetti Code,On Program Comprehension," *European Conference on Software Maintenance and Reengineering*, 2011.
[9] E. Gamma, R. Helm, R. Johnson, and J. Vlidssides, *Design Patterns: Elements of Reusable Object-Oriented Software.*
[10] S. Acharya and V. Pandya, "Bridge between Black Box and White Box – Gray Box Testing Technique," *International Journal of Electronics and Computer Science Engineering.*