

---

## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Sebelumnya

Penelitian Meinarini Catur Utami dan Yoga Tri Hutomo pada tahun 2014 dengan judul “Penerapan *Waterfall* Dalam Analisis Dan Perancangan Sistem Informasi Manajemen Dokumen Surat Menyurat Pada Bank BJB Kantor Cabang BSD Tangerang”. Penelitian ini memiliki tujuan utama yaitu untuk memperbaiki pengelolaan surat masuk maupun keluar yang masih manual. Metode *waterfall strategy* digunakan karena mudah dalam penerapannya serta bisa menghasilkan sistem yang cukup baik karena memerlukan perencanaan yang matang serta langkah demi langkah yang harus dikerjakan dengan detail, dan juga menggunakan notasi *Unified Modelling Language* (UML). Berdasarkan penelitian tersebut, didapatkan hasil keteraturan dalam penyimpanan dokumen secara digital di bank BJB yang mempermudah pegawai BJB dalam pengelolaan dokumen. Meskipun demikian, Waterfall dinilai lambat dalam pembuatan sistem ini karena kesesuaian antar tahapan tidak fleksible sehingga membutuhkan waktu yang lebih lama dari metoda RAD<sup>[4]</sup>.

Penelitian Saidah pada tahun 2011 dengan judul “Perancangan Sistem Management Pengarsipan Berbasis Elektronik Pada Madrasah Tsanawiyah Negeri Gajah”, dibuat dengan permasalahan sulitnya mengelola arsip dalam hal penyimpanan serta pengaksesan. Hal itu menyebabkan tidak efisien sehingga menghabiskan waktu. Adapun metode yang digunakan adalah *Research and Development* (R&D) yang terbagi atas tiga tahap, yaitu: metode deskriptif, evaluatif, dan eksperimental. Metode deskriptif adalah fase untuk mencari informasi terkait dengan masalah yang dihadapi. Metode evaluatif adalah fase yang berkaitan dengan pencarian informasi dan data *empiris* melalui teori dan penelitian relevan terkait dengan produk yang akan dihasilkan. Metode eksperimental adalah percobaan untuk mendapatkan hasil dari pengidentifikasian yang telah dilakukan oleh peneliti terkait produk yang dikembangkan. Hasil yang di dapat adalah sebuah program aplikasi yang telah dirancang dan dapat

digunakan sebagai alternatif untuk memecahkan masalah yang ada di Madrasah Tsanawiyah Negeri Gajah. Penggunaan sistem perancangan berbasis *client-server* diharapkan dapat membantu pihak pengelola arsip untuk dapat mengelola arsip secara efektif dan efisien. Penggunaan media elektronik dalam pengelolaan arsip akan memperoleh manfaat kecepatan, kemudahan dan hemat. Maksud kecepatan di sini adalah melalui penggunaan media elektronik dalam pengelolaan arsip. R&D pada prinsipnya memerlukan waktu yang relatif panjang, karena prosedur yang harus di tempuh relatif kompleks sehingga waktu yang dibutuhkan sampai sistem informasi selesai cukup lama dibandingkan dengan RAD<sup>[5]</sup>.

Penelitian Lukman Hasan pada tahun 2014 dengan judul “Pengembangan Sistem Informasi Pengarsipan Tata Usaha Menggunakan Metode *Agile* di MTS Arryosyidin Secang Magelang”. Penelitian ini memiliki tujuan yaitu memperbaiki pengelolaan data sekolah yang masih manual dengan cara mendigitalisasikan data serta transaksi yang ada. Metode yang digunakan adalah *Agile Development*. Penelitian ini menggunakan *agile* karena tahapan-tahapan dalam *agile* tidak harus runtun dimulai dari analisis, desain, implementasi, pengujian dan evaluasi. Tetapi lebih menitikberatkan pada kebutuhan dari *stakeholder*-nya. Metode *agile* ini dapat membuat suatu sistem yang dibangun dalam waktu singkat dan mampu mengendalikan perubahan-perubahan kebutuhan yang mendadak dalam pengembangan sebuah sistem. Adapun hasil yang didapatkan adalah sistem informasi yang dibangun telah berhasil diimplementasikan dan dapat menunjang efektifitas kinerja, dan terbukti dari hasil pengujian pada quisioner responden. Akan tetapi, metode *agile* ini dinilai kurang efisien, karena peneliti harus selalu siap dengan perubahan sewaktu-waktu sehingga penggunaan sumber daya akan menjadi lebih besar dari metode RAD<sup>[6]</sup>.

Penelitian Sandi Kosasi pada tahun 2015 dengan judul “Penerapan *Rapid Application Development* Pada Sistem Penjualan Sepeda Online “, memiliki tujuan untuk menghasilkan sistem penjualan sepeda online melalui integrasi basis data dan membangun hubungan dengan pelanggan

secara lebih personal. Sasarannya lebih kearah memperluas jangkauan pemasaran dan meningkatkan penjualan melalui media yang lebih fleksibel dan lebih ekonomis. Metode yang digunakan pada penelitian kali ini adalah menggunakan *Rapid Application Development* (RAD). Metode RAD menekankan cakupan pemodelan bisnis (*bussiness modelling*), pemodelan data (*data modelling*), pemodelan proses (*process modelling*), pembuatan aplikasi (*application generation*) dan pengujian (*testing and turnover*). Penelitian tersebut mendapatkan hasil, melalui penerapan metode RAD (*Rapid Application Development*) dalam menghasilkan sistem penjualan sepeda online dapat memenuhi kebutuhan pengguna secara signifikan dan memberikan nilai tambah untuk pencapaian tujuan dan sasaran. Penyelesaian setiap modul perangkat lunak berdasarkan setiap tahapan dari metode RAD dapat berjalan dengan baik dan lancar. Sistem dapat memperkenalkan dan menjual produk sepeda kepada masyarakat luas. Hasil pengujian menampilkan sistem memiliki fitur navigasi dalam memberikan kemudahan bagi pengunjung, baik kepada konsumen ataupun pelanggan saat mengunjungi halaman website<sup>[7]</sup>.

**Tabel 2.1** Perbandingan Referensi

Nama peneliti	Judul Penelitian	Masalah	Metode	Hasil
Meinarini Catur Utami, Yoga Tri Hutomo (2014)	Penerapan <i>Waterfall</i> Dalam Analisis dan Perancangan Sistem Informasi Manajemen Dokumen Surat Menyurat Pada Bank BJB Kantor Cabang BSD Tangerang	Pengelolaan surat masuk dan keluar masih manual	<i>Waterfall</i>	Keteraturan dalam penyimpanan dokumen secara digital di bank BJB yang mempermudah pegawai BJB dalam pengelolaan dokumen

Tabel 2.2 Perbandingan Referensi (lanjutan)

Nama peneliti	Judul Penelitian	Masalah	Metode	Hasil
Saidah (2014)	Perancangan Sistem Management Pengarsipan Berbasis Elektronik Pada Madrasah Tsanawiyah Negeri Gajah	Sulitnya mengelola arsip dalam hal penyimpanan serta pengaksesan	<i>Research and Development (R&amp;D)</i>	Membantu pihak pengelola arsip untuk dapat mengelola arsip secara efektif dan efisien
Lukman Hasan (2014)	Pengembangan Sistem Informasi Pengarsipan Tata Usaha Menggunakan Metode <i>Agile</i> di MTS Arryosyidin Secang Magelang	Pengelolaan data sekolah masih manual	<i>Agile</i>	Berhasil diimplementasikan dan dapat menunjang efektifitas kinerja, dan terbukti dari hasil pengujian pada quisioner responden
Sandy Kosasi (2015)	Penerapan Rapid Application Development Pada Sistem Penjualan Sepeda Online	Penjualan sepeda online belum menyajikan informasi yang baik	<i>Rapid Application Development (RAD)</i>	Sistem penjualan sepeda online melalui integrasi basis data dan membangun hubungan dengan pelanggan secara lebih personal

Penelitian kali ini, peneliti menggunakan metode RAD untuk proses pembuatan sistem informasi manajemen dokumen. Sistem informasi ini dibuat dengan sumber daya dan studi kasus yang ada di Institut Teknologi Telkom Purwokerto Program Studi S1 Informatika.

Adapun kelebihan RAD daripada metode yang sudah pernah digunakan adalah lebih efektif dalam menghasilkan sistem yang lebih cepat dari pada metode *waterfall*<sup>[8]</sup>. Selain itu waktu yang dibutuhkan dalam pembangunan suatu sistem tersebut penggunaan sumber daya lebih sedikit dari pada metode *agile*<sup>[7]</sup>. Alasan lain peneliti menggunakan RAD adalah karena melibatkan konsumen di dalam pembuatan sistem informasi, metode RAD mendapatkan suatu desain yang dapat diterima oleh konsumen. Selain mendapat desain yang dapat diterima, RAD dapat dikembangkan dengan mudah dan cepat yang membuatnya lebih baik dari *Research and Development* (R&D) karena R&D masih menjadi salah satu dari SDLC (*Software Development Lifecycle*)<sup>[3]</sup>.

## 2.2 Dasar Teori

### 2.2.1 Konsep Informasi

Informasi adalah sebuah istilah yang tidak tepat dalam pemakaiannya secara umum. Informasi dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi dan lain sebagainya. Informasi ibarat darah yang mengalir di dalam tubuh suatu organisasi, sehingga informasi ini sangat penting di dalam organisasi. Suatu sistem yang kurang mendapat informasi akan menjadi luruh, kerdil dan akhirnya mati. Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.<sup>[9]</sup>

### 2.2.2 Konsep Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Peneliti M . Ichsan Anugra mengambil kesimpulan bahwa suatu sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu sama lain dalam bekerja sama untuk mencapai tujuan tertentu.<sup>[10]</sup>

### 2.2.3 Dokumen dan Pengelolaan

Definisi dokumen adalah rekaman kegiatan atau peristiwa dalam berbagai bentuk dan media sesuai dengan perkembangan teknologi informasi dan komunikasi yang dibuat dan diterima oleh lembaga negara, pemerintahan daerah, lembaga pendidikan, perusahaan, organisasi politik, organisasi kemasyarakatan, dan perseorangan dalam pelaksanaan kehidupan bermasyarakat, berbangsa, dan bernegara<sup>[11]</sup>.

Pengelolaan adalah proses membantu merumuskan kebijakan dan tujuan organisasi atau proses yang memberikan pengawasan pada suatu hal yang terlibat dalam pelaksanaan kebijakan dan pencapaian tujuan. Pengelolaan sama dengan manajemen yaitu penggerakan, pengorganisasian dan pengarahan usaha manusia untuk memanfaatkan secara efektif material dan fasilitas untuk mencapai tujuan organisasi. Pengelolaan dokumen ialah urutan langkah untuk menyelesaikan suatu pekerjaan sejak permulaan sampai selesai. Pengelolaan dokumen terdiri dari prosedur permulaan dan prosedur penyimpanan. Prosedur permulaan untuk surat masuk meliputi kegiatan-kegiatan administrasi pencatatan, pendistribusian dan pengolahan<sup>[11]</sup>.

### 2.2.4 *Hypertext Preprocessor (PHP)*

PHP adalah bahasa pemrograman yang dijalankan di dalam halaman web, yang pada umumnya digunakan untuk mengelola informasi di internet. Sedangkan dalam pengertian lain PHP bahasa pemrograman *server-side* yang bersifat *open source* atau gratis. PHP merupakan *script* yang menyatu dengan HTML dan berada pada *server (Server-side HTML embedded scripting)*.<sup>[12]</sup>

### 2.2.5 MySql

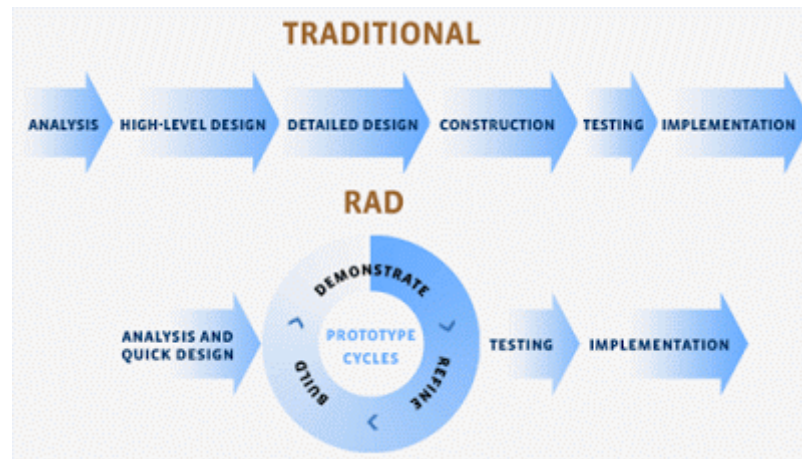
MySQL merupakan salah satu *database server* yang berkembang di lingkungan *open source* dan didistribusikan secara *free* (gratis) dibawah lisensi GPL. MySQL merupakan RDBMS (*Relational Database Management System*) server. RDBMS adalah program yang memungkinkan pengguna database untuk membuat, mengelola, dan menggunakan data pada suatu model *relational*.<sup>[12]</sup>

### 2.2.6 Sistem Informasi Manajemen

Sistem informasi manajemen (SIM) adalah sebuah sistem informasi pada level manajemen yang berfungsi untuk membantu perencanaan, pengendalian, dan pengambilan keputusan dengan menyediakan resume rutin dan laporanlaporan tertentu.<sup>[13]</sup>

### 2.2.7 Rapid Application Development (RAD)

Metode *Rapid Application Development* (RAD) merupakan Sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan pada efektifitas waktu pembuatan sistem. Selain itu, RAD merupakan gabungan dari bermacam-macam teknik terstruktur dengan teknik *prototyping* dan teknik pengembangan *joint application* untuk mempercepat pengembangan sistem/aplikasi. RAD bisa menghasilkan suatu sistem dengan cepat karena sistem yang dikembangkan dapat memenuhi keinginan dari para pemakai sehingga dapat mengurangi waktu untuk pengembangan ulang setelah tahap implementasi<sup>[3]</sup>. Efektifitas RAD dapat dilihat pada Gambar 2.1:

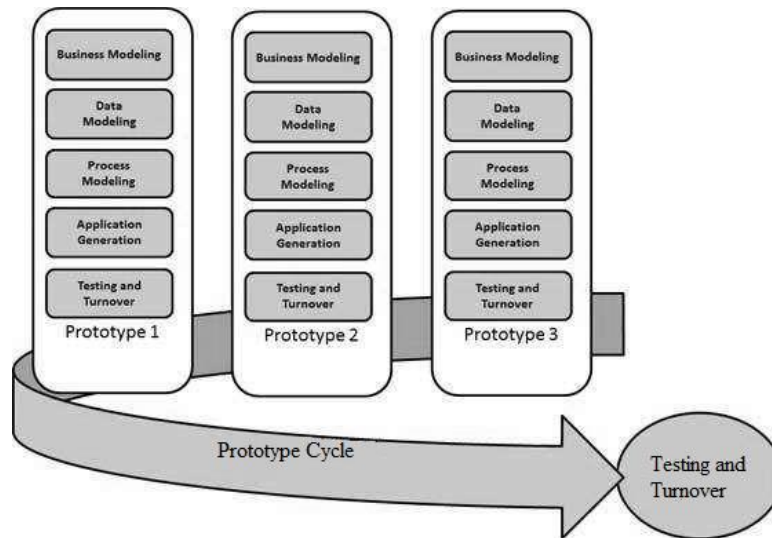


**Gambar 2.1** Metode Tradisional dan RAD<sup>[8]</sup>

Gambar 2.1 menggambarkan perbedaan dari metode tradisional dengan metode RAD. Dapat dilihat tahapan tradisional mengacu pada urutan *Software Development Life Cycle* (SDLC). SDLC adalah tahapan-tahapan pekerjaan yang dilakukan oleh analis sistem dan programmer dalam membangun sistem informasi. RAD tahap pertama langsung membuat analisis dan desain, dilanjutkan ke tahap siklus *prototyping* yaitu membangun, memperhalus, dan mendemonstrasikan. Hal itu akan mempercepat proses dalam pembuatan suatu project. RAD memang lebih cepat dari *waterfall*. Jika kebutuhan dan batasan proyek sudah diketahui dengan baik dan proyek memungkinkan untuk dijalankan<sup>[8]</sup>.

Pendekatan RAD melingkupi fase-fase sebagai berikut yang ditunjukkan pada Gambar 2.2.





**Gambar 2.2** Proses Pengembangan RAD<sup>[8]</sup>

Proses Pengembangan dalam Model *Rapid Application Development* (RAD), yaitu :

a. *Business modeling*

Aliran informasi di antara fungsi – fungsi bisnis dimodelkan dengan suatu cara untuk mengetahui informasi yang mengendalikan proses bisnis, informasi yang akan dimunculkan serta pihak yang membuat informasi tersebut, arah informasi pergi, serta pihak yang memproses informasi tersebut.

b. *Data modeling*

*Data modeling* adalah tahap dimana hasil yang di peroleh pada fase *business modeling* disaring dan di bagi menjadi beberapa objek data yang dibutuhkan. Karakteristik masing-masing objek diinisialisasikan dan di indentifikasi hubungan antar objek-objek tersebut.

c. *Process modelling*

*Process Modelling* adalah tahap dimana objek-objek yang telah diidentifikasi dari fase *data modeling*, ditransformasikan kedalam suatu aliran informasi yang diperlukan untuk implementasi sebuah fungsi dalam organisasi.

d. *Application generation*

Tahap *application generation*, menjelaskan mengenai alat yang digunakan untuk mengkonversi model proses kedalam kode dan sistem yang akan di buat.

e. *Testing and turnover*

Tahap ini, sistem yang telah dibuat pada tahap *application generation* diuji *interfacenya* maupun komponen-komponen yang terkait.

RAD memiliki kelebihan sebagai berikut:

- a. RAD mengikuti tahapan pengembangan sistem seperti umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada (*reusable object*) sehingga pengembang tidak perlu membuat dari awal lagi<sup>[8]</sup>.
- b. RAD mempunyai kemampuan untuk menggunakan komponen yang sudah ada, menjadikam durasi perancangan sistem menjadi lebih singkat dan membuat biaya menjadi lebih rendah<sup>[8]</sup>.
- c. RAD dapat menghasilkan suatu desain yang dapat diterima oleh konsumen dan dapat dikembangkan dengan mudah<sup>[8]</sup>.

### 2.2.8 *Alpha Testing and Beta Testing*

a. *Alpha Testing*

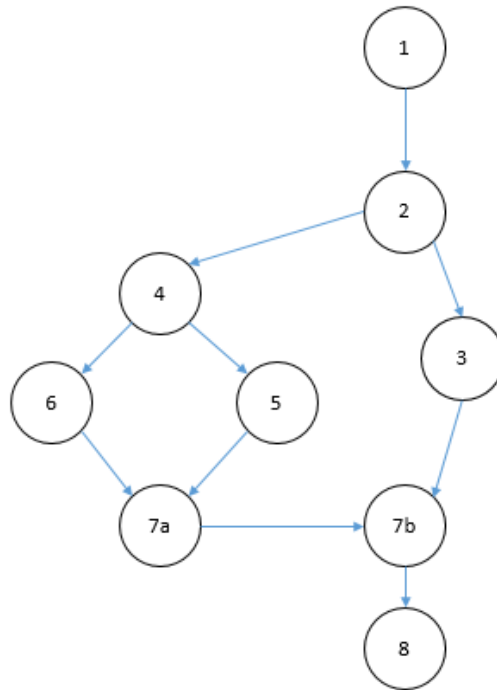
*Alpha testing* adalah pengujian yang dilakukan dengan tujuan mengidentifikasi dan menghilangkan sebanyak mungkin masalah sebelum akhirnya sampai ke *pengguna*. *Alpha testing* dilakukan setelah perangkat lunak jadi oleh orang-orang yang memang ahli di bidangnya<sup>[14]</sup>.

Pengujian kali ini peneliti menggunakan dua pendekatan yaitu *black-box* dan *white-box*. *Black-box* adalah ujicoba yang dilakukan pada *interface* perangkat lunak. Walaupun didesain untuk menemukan kesalahan, ujicoba *black-box* digunakan untuk mendemonstrasikan fungsi perangkat lunak yang

dioperasikan; apakah *input* diterima dengan benar, dan *ouput* yang dihasilkan benar; apakah integritas informasi eksternal terpelihara. Ujicoba *black-box* memeriksa beberapa aspek sistem, tetapi memeriksa sedikit mengenai struktur logikal internal perangkat lunak<sup>[15]</sup>.

Pengujian *white-box* merupakan metode desain uji kasus yang menggunakan struktur kontrol dari desain prosedural untuk menghasilkan kasus-kasus uji. Pengujian *white-box* didasarkan pada pemeriksaan detail prosedural. Alur logikal suatu perangkat lunak diujicoba dengan menyediakan kasus ujicoba yang melakukan sekumpulan kondisi dan/atau perulangan tertentu. Status dari program dapat diperiksa pada beberapa titik yang bervariasi untuk menentukan apakah status yang diharapkan atau ditegaskan sesuai dengan status sesungguhnya<sup>[15]</sup>.

Ujicoba berbasis alur (*Base Flow Graph*) merupakan teknik ujicoba *white-box* pertama yang berhasil dijalankan. Metode berbasis alur memungkinkan perancang kasus uji untuk menghasilkan ukuran kompleksitas logikal dari desain prosedural dan menggunakan ukuran ini untuk mendefinisikan himpunan basis dari alur eksekusi. Kasus uji dihasilkan untuk melakukan sekumpulan basis yang dijamin untuk mengeksekusi setiap perintah dalam program, sedikitnya satu kali selama ujicoba<sup>[15]</sup>. Notasi sederhana untuk merepresentasikan alur kontrol disebut graf alur (flow graph), seperti Gambar 2.3.



**Gambar 2.3** *Flow Graph Notation*<sup>[15]</sup>

*Cyclomatic complexity* merupakan perangkat lunak *metric* yang menyediakan ukuran kuantitatif dari kompleksitas logikal suatu program. Ketika digunakan dalam konteks metode ujicoba berbasis alur, nilai yang dikomputasi untuk kompleksitas *cyclomatic* mendefinisikan jumlah jalur independen (*independent path*) dalam himpunan basis suatu program dan menyediakan batas atas untuk sejumlah ujicoba yang harus dilakukan untuk memastikan bahwa seluruh perintah telah dieksekusi sedikitnya satu kali. *Independent path* adalah alur manapun dalam program yang memperkenalkan sedikitnya satu kumpulan perintah pemrosesan atau kondisi baru. Contoh *independent path* dari *flow graph* pada gambar 2.3:

Path 1 : 1 – 11

Path 2 : 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Path 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Path 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Misalkan setiap *path* yang baru memunculkan *edge* yang baru, dengan *path* : 1 - 2 – 3 – 4 – 5 – 10 - 1 - 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

*Path* diatas tidak dianggap sebagai *independent path* karena kombinasi *path* diatas telah didefinisikan sebelumnya. Ketika ditetapkan dalam graf alur, maka *independent path* harus bergerak sedikitnya 1 *edge* yang belum pernah dilewati sebelumnya<sup>[15]</sup>. Kompleksitas *cyclomatic* dapat dicari dengan salah satu dari tiga cara berikut :

1. Jumlah region dari graf alur mengacu kepada kompleksitas *cyclomatic*
2. Kompleksitas *cyclomatic* untuk graf alur G didefinisikan :

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

Rumus 2.1 *Cyclomatic Complexity*

Dimana E = jumlah *edge*, dan N = jumlah *node*

3. Kompleksitas *cyclomatic* untuk graf alur G didefinisikan :

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

Rumus 2.2 *Cyclomatic Complexity*

Dimana P = jumlah *predicates nodes*

Berdasarkan flow graph gambar 2.3 diatas, maka kompleksitas *cyclomatic*-nya dapat di hitung sebagai berikut:

*Flow graph* diatas mempunyai empat *region*

$$V(G) = 11 \text{ edges} - 9 \text{ nodes} + 2 = 4$$

$$V(G) = 3 \text{ predicates nodes} + 1 = 4$$

Hasil kompleksitas *cyclomatic* menggambarkan banyaknya *path* dan batas atas sejumlah ujicoba yang harus

dirancang dan dieksekusi untuk seluruh perintah dalam program<sup>[15]</sup>.

b. *Beta Testing*

*Beta testing* adalah pengujian yang sepenuhnya dilakukan oleh *pengguna*. *Beta testing* dilakukan pada satu atau lebih calon pengguna akhir (*end-user*), pengembang maupun orang-orang yang ahli di bidang tersebut biasanya tidak ada pada pengujian ini.<sup>[14]</sup>

### 2.2.9 *Unified Modeling Language (UML)*

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat perangkat lunak berorientasi objek. UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma *object oriented*. UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen- komponen yang diperlukan dalam sistem perangkat lunak<sup>[16]</sup>.

UML sendiri terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. Penelitian kali ini, peneliti menggunakan dua diagram yaitu *Use case Diagram* dan *Sequence Diagram*<sup>[16]</sup>.

Diagram *Use case* menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. Diagram *Use case* dekat kaitannya dengan kejadian-

kejadian. Kejadian (scenario) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem<sup>[16]</sup>.

Diagram *sequence* merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut<sup>[16]</sup>.