

## Analisis dan Implementasi Perancangan Metode *Rational Unified Process* pada layanan SDB dan Metode Pengujian *Product Metric* pada Bank Mandiri Cabang Palu Sam Ratulangi

### *Analysis and Implementation Rational Unified Process Method In SDB Service and Product Metric Testing Method In Bank Mandiri Branch Palu Sam Ratulangi*

**Muhammad Taufik<sup>1</sup>, Eko Darwiyanto, ST., MT.<sup>2</sup>, Shinta Yulia P, ST., MT.<sup>3</sup>**

<sup>1</sup>Program Studi S1 Teknik Informatika, School of Computing, Telkom University

<sup>2,3</sup>School of Computing, Telkom University

<sup>1,2,3</sup>Gedung E-F, Jalan Telekomunikasi 1, Terusan Buah Batu, Bandung 40257

<sup>1</sup>muhtaufik@students.telkomuniversity.ac.id, <sup>2</sup>ekodarwiyanto@telkomuniversity.ac.id,

<sup>3</sup>shintayulia@telkomuniversity.ac.id

#### **Abstrak**

Bank merupakan salah satu perusahaan yang memiliki aktivitas di dalamnya, tanpa adanya sebuah sistem yang memadai untuk menjalankan proses bisnis, tentunya akan mengurangi efektifitasnya. SDB (*save deposit box*) merupakan salah satu layanan yang di dalamnya terdapat beberapa proses bisnis, disamping itu SDB merupakan layanan tambahan yang diberikan kepada nasabah prioritas di bank. Salah satu bank yang menyediakan layanan SDB adalah Bank Mandiri, namun sayangnya layanan SDB ini masih dikerjakan secara manual. Oleh karena itu diperlukannya sebuah sistem yang terkomputerisasi yang dapat membantu proses bisnis. Dengan menggunakan metode RUP (*rational unified process*) dan metode pengujian *product metric* akan menghasilkan sebuah sistem yang berkualitas baik dan sesuai dengan kebutuhan konsumen. Dengan adanya sistem ini dapat meningkatkan efektifitas dari proses bisnis yang berjalan.

**Kata kunci :** SDB (*save deposit box*), RUP (*rational unified process*), *product metric*, *metric*

#### **Abstract**

Bank is one of the company which is have many activity, without a good sistem, surely the efectivity of the business process will decrease. SDB (*save deposit box*) is one of service which is have some business process, in other hand SDB is an additional service in bank that is provided to the priority customer. One of bank that has SDB service is Bank Mandiri, unfortunately this service still done it manually. Therefore it is necessary to have a computrized system that can help to do a business process. By using RUP (*rational unified process*) method and testing method *product metric* will produce a system with good quality and according to the company requirement. With this system will increase the efectivity of the bussiness process.

**Keywords :** *Rational unified process (RUP)*, *safe deposit box (SDB)*, *Product metric*, *metric*

## **1. Pendahuluan**

Bank Mandiri merupakan salah satu bank terbesar di Indoneiseia yang memiliki layanan SDB (*save deposit box*). SDB adalah layanan penyimpanan yang disediakan dari bank kepada pihak nasabah. Nasabah dapat menyimpan barang berharga mereka tanpa khawatir keamanannya. Berdasarkan hasil wawancara, pemilik SDB harus mendapatkan perhatian khusus karena yang mendapatkan layanan ini adalah nasabah prioritas saja. Pengelolaan SDB menjadi tolak ukur pertumbuhan dana nasabah. Namun tidak diimbangi dengan sistem yang mudah karena masih bersifat manual. Sistem manual ini memiliki banyak kendala sehingga perlu adanya sebuah sistem informasi yang membantu pengelolaan SD. Sistem informasi dipilih menggunakan metode RUP (*Rational Unified Process*). Dalam pengembangan RUP terdapat beberapa fase, seperti inception, elaboration, construction, dan transition<sup>[1]</sup>. Untuk menguji software yang telah dibuat akan menggunakan software quality metric bagian *product metric*. Software dikatakan berkualitas dapat dilihat dari kompleksitas dari produk rendah.

## **2. Dasar Teori**

### **2.1 Sistem Informasi**

Sistem informasi menurut John F. Nash adalah kombinasi dari manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian yang bermaksud menata jaringan komunikasi yang penting, proses atas transaksi-

transaksi tertentu dan rutin, membantu manajemen dan pemakai intern dan ekstern dan menyediakan dasar pengambilan keputusan yang tepat.

Menurut Henry Lucas, sistem informasi adalah suatu kegiatan dari prosedur-prosedur yang diorganisasikan, bilamana dieksekusi akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian didalam.

## 2.2 Bank Mandiri

Bank Mandiri didirikan pada 2 Oktober 1998, sebagai bagian dari program restrukturisasi perbankan yang dilaksanakan oleh pemerintah Indonesia. Pada bulan Juli 1999, empat bank pemerintah -- yaitu Bank Bumi Daya, Bank Dagang Negara, Bank Ekspor Impor Indonesia dan Bank Pembangunan Indonesia -- dilebur menjadi Bank Mandiri, dimana masing-masing bank tersebut memiliki peran yang tak terpisahkan dalam pembangunan perekonomian Indonesia. Sampai dengan hari ini, Bank Mandiri meneruskan tradisi selama lebih dari 140 tahun memberikan kontribusi dalam dunia perbankan dan perekonomian Indonesia<sup>[3]</sup>.

Visi:

Menjadi Lembaga Keuangan Indonesia yang paling dikagumi dan selalu progresif

Misi:

- Berorientasi pada pemenuhan kebutuhan pasar
- Mengembangkan sumber daya manusia profesional
- Memberi keuntungan yang maksimal bagi stakeholder
- Melaksanakan manajemen terbuka
- Peduli terhadap kepentingan masyarakat dan lingkungan

## 2.5 Safe Deposit Box

*Safe Deposit Box* (SDB) merupakan jasa penyewaan kotak penyimpanan harta atau surat-surat berharga yang dirancang secara khusus dari bahan baja dan ditempatkan dalam ruang khasanah yang kokoh, tahan bongkar dan tahan api untuk memberikan rasa aman bagi penggunanya<sup>[4]</sup>. Berdasarkan hasil wawancara, Safe deposit box hanya ditujukan kepada nasabah-nasabah prioritas pemilik dana yang terus berkembang, sehingga tidak semua nasabah dapat menikmati layanan ini. Layanan SDB ini juga menjadi salah satu acuan bank untuk memberi perhatian khusus kepada nasabah-nasabah pemilik SDB. Berikut salah satu contoh gambar dari SDB yang diambil dari pencarian di situs google.com

## 2.6 Rational Unified Process

*Rational Unified Process* (RUP) merupakan salah satu proses rekayasa perangkat lunak yang menyediakan pendekatan untuk menentukan tugas dan tanggung jawab dalam pengembangan suatu organisasi, tujuannya adalah untuk memastikan produksi kualitas tinggi, *software* memenuhi dengan kebutuhan *user* sesuai dengan jadwal dan biaya yang telah dirancang<sup>[5]</sup>. Fase dan iterasi dari metode RUP ini adalah<sup>[6]</sup>

### Fase dan Iterasi RUP

Fase dan iterasi dari metode RUP adalah :

#### a. Inception.

Tahap ini membangun *business case* untuk sistem dan membatasi ruang lingkupnya, untuk melakukan hal ini diharuskan untuk mengidentifikasi semua entitas eksternal yang akan berinteraksi dengan sistem, dan mendefinisikan interaksi pada level tertentu. Ini juga termasuk mengidentifikasi semua *use cases* dan menjelaskan beberapa yang signifikan. *Business case* termasuk kriteria keberhasilan, perkiraan resiko, dan mengestimasi sumber daya yang dibutuhkan.

#### b. Elaboration

Tujuan dari fase *elaboration* adalah menganalisis domain masalah, membuat sebuah dasar arsitektur, membangun rencana proyek, dan mengeliminasi resiko terbesar dari proyek. Untuk menjalankan objek-objek tersebut diperlukan melihat lebih luas dan lebih dalam terhadap sistem. Pada tahap ini merupakan tahap paling sulit karena pada tahap ini memastikan bahwa arsitektur, kebutuhan, dan perencanaan cukup stabil sehingga waktu dan biaya tidak berubah.

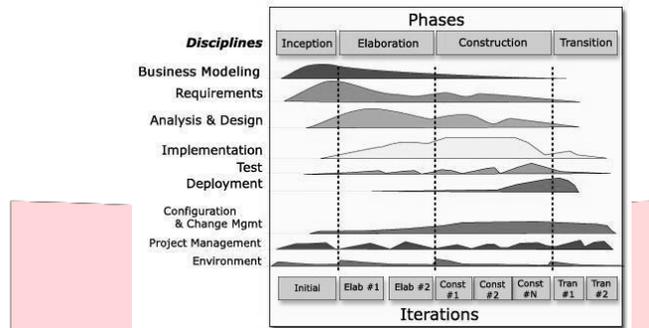
#### c. Construction

Dalam fase ini semua komponen dan fitur aplikasi yang dibuat dan di integrasikan kedalam *software*. Dalam fase ini juga dituntut untuk mengoptimalkan sumber daya, biaya, jadwal dan kualitas. Ini menjadi salah satu alasan mengapa pengembangan dari arsitektur ditekankan pada

fase *elaboration*. Keluaran dari fase *construction* ada sebuah *software* yang sudah siap diserahkan kepada *client*

d. Transition.

Transition adalah fase dimana *software* diserahkan kepada *client*, dalam fase ini juga dilakukan pengujian terhadap *software*, dan apabila *software* telah diserahkan kepada *client*, *developer* masih memantau bagaimana kinerja dari *software* tersebut.



GambarError! No text of specified style in document..1 Arsitektur RUP

**Iteration**

Setiap fase dalam metode RUP dapat mengalami kesalahan. oleh karenanya adanya perulangan pada tiap fase sehingga memungkinkan mendeteksi kesalahan sedini mungkin, dan menjadi lebih cepat untuk menjadi sebuah *software*. Beberapa keuntungan lain dengan adanya iteration ini adalah *developer* dapat terus belajar selama proses iterasi, level yang lebih tinggi dalam penggunaan kembali.

**2.7 Unified Modeling Language**

*UML (Unified Model Language)* merupakan alat industri standar yang memungkinkan kita untuk mengkomunikasikan dengan jelas tentang kebutuhan, arsitektur dan desain [6]. *UML* merupakan salah satu alat yang paling berguna dan paling populer di dalam dunia pengembangan sistem. Hal ini disebabkan karena *UML* adalah sebuah metode pemodelan visual yang memungkinkan kepada pengembang untuk membuat sebuah blueprints yang dapat di mengerti, sehingga hasil blueprint ini dapat dimengerti antara satu dengan yang lainnya [6].

**2.8 Software Metric**

Pengukuran merupakan suatu hal yang dasar dalam setiap disiplin ilmu teknik dan termasuk di dalamnya rekayasa perangkat lunak. Pengukuran memungkinkan kita untuk mendapatkan informasi dengan cara menyediakan mekanisme untuk mengevaluasi secara objektif [10]. Lord Kelvin berpendapat bahwa ketika anda bisa mengukur apa yang kita bicarakan dan dapat menggambarkan dengan angka berarti anda mengerti tentang itu, hal ini menjadi landasan mengapa pengukuran diperlukan di dalam pembangunan sebuah *software*. *Software Metric* merupakan ukuran kuantitatif dari atribut yang dimiliki suatu sistem atau perangkat lunak. *Software metric* biasa digunakan sebagai acuan dalam *quality assurance*, *performance*, *debugging*, *management*, dan *estimating cost*.

**2.9 Product Metric**

*Product metric* adalah sebuah *metric* yang berkaitan dengan perangkat lunak itu sendiri [11], adapun beberapa hal yang diukur didalam *product metric* seperti *complexity*, *size*, *design features*, *quality metric*, dan hal – hal yang berkaitan dengan perangkat lunak itu sendiri. Hal – hal lain yang dapat diukur dalam *product metric* ini adalah *Cyclomatic Complexity (CC)*, *Size*, *Comment Percentage*, *Weight Method per Class (WMC)*, *the defect density metric (LOC, Function points)*, dan *customer satisfaction metric*. Adapun rumus untuk menghitung *cyclomatic complexity* [12]

$$N = \text{Edges} - \text{Nodes} + 2$$

Nilai edges dan nodes didapatkan dengan cara menghitung setiap percabangan yang ada didalam program, semakin besar nilai *cyclomatic complexity* suatu *class* maka akan semakin kompleks *class* tersebut. Adapun rumus untuk *comment percentage* sebagai berikut

jumlah total koment

jumlah total baris



Semakin banyak jumlah *comment percentage* maka akan semakin besar nilai *understandability* nya, sehingga akan memudahkan untuk dapat dipahami oleh orang lain. Untuk perhitungan *weighted method per class* dapat dilakukan dengan cara menghitung jumlah kompleksitas dari semua method dalam suatu class.

2.10 Barcode[7]

Barcode adalah bentuk kode mesin yang dapat dibaca. Bentuknya berupa bar dan spasi berwarna hitam dan putih dalam jarak tertentu yang merepresentasikan karakter alphanumeric. Barcode digambarkan dalam bentuk baris hitam dan tipis yang disusun berderet sejajar secara horizontal. Pada produk-produk dipasaran barcode dilengkapi dengan kode angka yang dimaksud untuk memudahkan pembacaan. Namun kode angka tersebut tidak berpengaruh pada pola kode baris yang tercantum.

Bentuk Barcode ada dua jenis

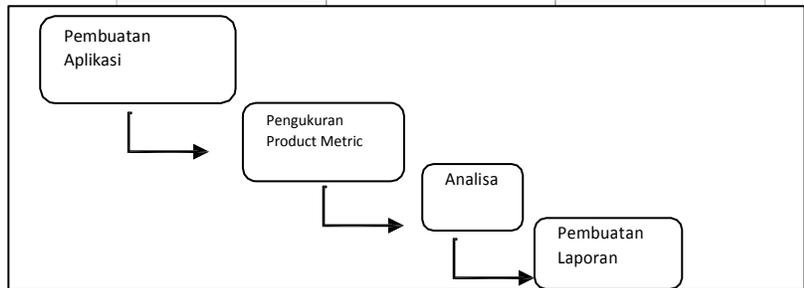
1. Barcode satu dimensi (linear bar berbentuk baris)
2. Barcode dua dimensi (space bar bentuk ruang)

3.Pembahasan

Perancangan sistem ini akan berfokus pada pengelolaan *safe deposit box* (SDB). Proses penggunaan SDB masih belum memiliki sistem yang memadai, masih manual dalam pengelolaanya. Mengingat SDB merupakan salah satu bagian penting dari sebuah bank karena berkaitan langsung dengan nasabah prioritas, Dengan mengetahui keadaan inilah yang melandasi pembuatan sistem yang dapat mengelola proses pengolahan data dan informasi untuk penggunaan SDB. Proses pengolahan data dan informasi SDB dikelola oleh masing-masing kantor cabang sehingga data berpusat dikantor tersebut, sehingga pihak kantor cabang dapat mengawasi rekening nasabah yang menggunakan SDB, guna melihat potensi nasabah terhadap peningkatan profit bank. adapun data yang dikirim ke kantor pusat hanyalah data kepemilikan SDB dan masa tenggang sewanya.

Tahap Penelitian

Pada tahapan ini akan dilakukan beberapa aktivitas yaitu

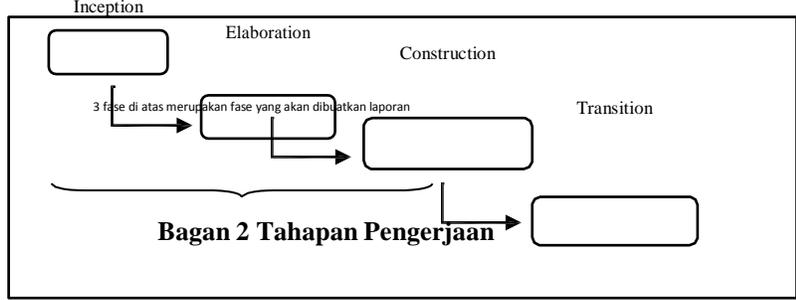


Bagan 1 Tahapan Penelitian

Pada tahapan penelitian di mulai dari pembuatan aplikasi, dengan menggunakan model proses RUP. Dan didalam RUP telah terdapat bagian atau tahapan untuk pembuatan aplikasi yaitu tahapan *inception, elaboration, construction, dan transition*. Setelah menyelesaikan pembuatan aplikasi, maka akan dilakukan pengukuran menggunakan *product metric*, setelah melakukan pengukuran akan dilakukan analisis terhadap hasil perhitungan, tahapan terakhir yaitu tahapan pembuatan laporan untuk semua proses yang telah dilakukan.

Tahapan Pengerjaan

Berikut tahapan pengerjaan pembuatan aplikasi berdasarkan metode RUP yang digunakan :



Bagan 2 Tahapan Pengerjaan

**Inception**

Pada tahap ini akan dilakukan beberapa aktivitas salah satunya adalah pengumpulan data, teknik yang digunakan dalam pengumpulan data yaitu menggunakan teknik

- a. Wawancara
- b. Observasi
- c. Review Dokumen

Setelah melakukan tiga proses diatas dilakukan proses pengecekan untuk mengecek perubahan. Dalam melakukan pengecekan terhadap hasil wawancara dibuatkan sebuah laporan berupa form dengan beberapa pertanyaan didalamnya. Setelah melakukan tahapan pengumpulan data, maka dilanjutkan ke aktivitas selanjutnya yaitu pendefinisian aplikasi yang akan dibuat. Beberapa hal yang perlu digambarkan terkait aplikasi yang akan dibuat adalah sebagai berikut

1. Aplikasi yang akan dibuat adalah aplikasi layanan SDB adapun aplikasi yang akan dibuat nantinya akan digunakan untuk menangani setiap transaksi SDB seperti pembukaan, peminjaman, dan kunjungan.
2. Aplikasi ini hanya digunakan untuk satu sisi, yaitu sisi petugas. Nasabah tidak akan bersentuhan secara langsung terhadap aplikasi

Setelah mendapatkan data dan telah mendefenisikan tentang aplikasi yang akan dibuat selanjutnya dibuatkan rancangan pengerjaan dari aplikasi, berikut rancangan pengerjaannya

- a. People

Adapun pihak-pihak yang terlibat dalam pembuatan aplikasi ini adalah developer dan pihak bank sebagai client.

- b. Cost

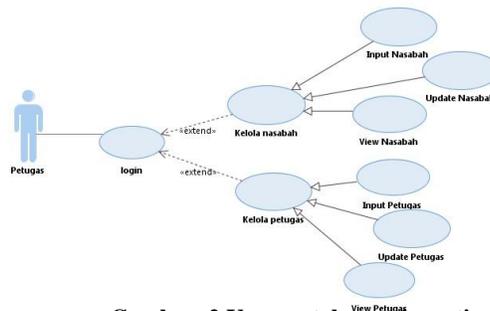
Pembuatan aplikasi ini tidak memiliki rancangan biaya.

- c. Time dan Product

**Tabel.1 Rancangan Pengerjaan**

No	Time	Product
1	Februari	Elaboration, membuat perancangan modul pinjam, petugas, box dan kuasa
2	Maret	Elaboration, membuat perancangan modul peminjaman, kunjungan, rekening, dan cetak laporan
3	April	Construction, pembuatan database
4	Mei	Construction, pembuatan program
5	Juni	Construction, pembuatan program
6	Juli	Pengukuran product metric
7	Agustus	Pengukuran product metric

Pada tahap ini juga digambarkan 10% – 20% diagram usecase dari palikasi yang akan dibuat, berikut gambaran diagram usecasenya



**Gambar. 2 Usecase tahapan inception**

Berikut skenario dari usecase yang dibuat

- Usecase : Login
- Aktor : Petugas
- Pre – Condition : Aktor masih berada diluar sistem dan ingin menggunakan sistem
- Post – Condition : Aktor melakukan login untuk dapat menggunakan sistem

Tabel 2 Skenario usecase

Aktor	Sistem
1. Menginputkan Username dan Password	
2. Mengklik tombol login	
	3. Validasi username dan password
	4. Jika benar akan menampilkan menu utama aplikasi
	5. Jika salah akan tetap berada di halaman login dan memulai proses nomor 1.

Keluaran dari fase ini adalah keputusan dari pihak client apakah mencapai kesepakatan dengan developer dengan rancangan pengerjaan yang telah dibuat.

**Elaboration**

Pada tahap ini merupakan tahap lanjutan dari inception, pada tahap ini akan dilakukan perancangan terhadap sistem yang akan dibuat berdasarkan hasil tahapan inception. Perancangan dalam tahapan ini meliputi pembuatan diagram UML, pembuatan rancangan database, pembuatan *mock-up* dari tampilan aplikasi, fase elaboration ini digambarkan menggunakan dokumen SKPL, untuk melakukan tahapan ini akan menggunakan sebuah dokumen yaitu spesifikasi kebutuhan perangkat lunak (SKPL).

**Gambaran Umum Sistem**

Aplikasi SDB sendiri nantinya akan digunakan secara khusus untuk melayani aktivitas SDB, sistem ini nantinya akan membantu petugas dalam menjalankan tugasnya. Berikut aktivitas yang akan ditangani oleh aplikasi SDB

1. Pendaftaran peminjaman box SDB, berikut didalamnya pendaftaran nasabah sebagai pemilik dan pendaftaran kuasa.
2. Pencatatan data kunjungan nasabah ataupun kuasa.

Adapun alur kerja sistem pendaftaran peminjaman box adalah sebagai berikut



Gambar .3 Alur Peminjaman

Alur kerja sistem kunjungan dapat digambarkan sebagai berikut



Gambar 4 Alur kunjungan

**Construction**

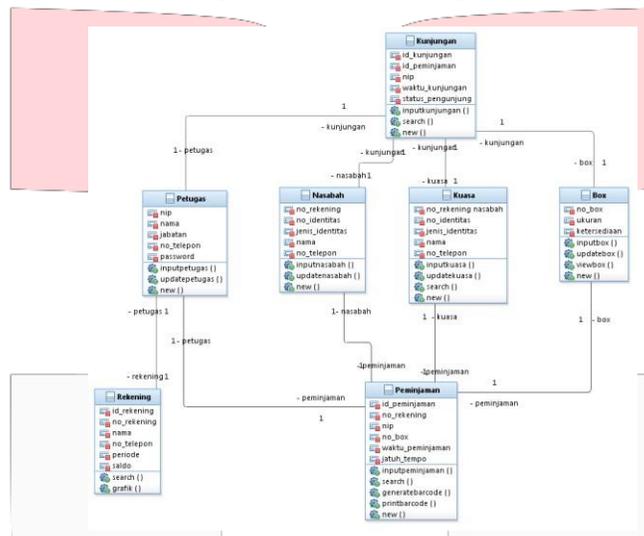
Tahapan construction adalah tahapan pembuatan aplikasi (coding) setelah tahapan elaboration selesai dilakukan. Pada tahap ini akan mengacu kepada dokumen SKPL yang telah selesai dan di setujui oleh pihak klien. Pembuatan aplikasi menggunakan bahasa pemrograman java dengan menggunakan framework hibernate dan menggunakan database MySql. Pemilihan database MySql dikarekan MySql

merupakan database *open source* sehingga memudahkan untuk menggunakannya, serta jumlah data yang dapat disimpan relatif besar sehingga database MySQL masih sanggup untuk menanganinya proses layanan SDB.

Dalam *product metric* terdapat beberapa parameter yang diukur seperti *size, complexity, dan performance*. Product metric memiliki beberapa metric seperti *Cyclomatic Complexity (CC), Size, Comment Percentage, Weight Method per Class (WMC), the defect density metric (LOC, Function points), dan customer satisfaction metric*. Dalam penggunaan product metric akan digunakan *object oriented metric (OO Metric)* sebagai alat ukur. Beberapa hal yang diukur dalam *product metric* ini adalah *Complexity Cyclomatic (CC), Comment Percentage, dan Weighted Method per Class (WMC)*.

Untuk *weighted method per class (WMC)* semakin besar nilai WMC di dalam suatu kelas maka akan semakin spesifik penggunaan kelas tersebut, hal ini menyebabkan semakin sulit untuk dapat digunakan kembali.

Berikut hubungan antara rancangan *class diagram* dengan class – class yang akan dibuat



Gambar. 5 Class Diagram

Berdasarkan perbandingan antara class diagram dengan tabel diatas dapat dilihat bahwa terdapat perbedaan jumlah method dan penambahan class seperti rekening selama proses pengembangan. Hal ini dikarenakan adanya penambahan modul – modul baru dalam aplikasi yang akan dibuat.

**Transition**

Tahapan transition merupakan tahapan peralihan aplikasi dari sisi developer kepada klien dalam hal ini pihak bank. Tahapan ini tidak dilakukan, pembangunan aplikasi ini hanya dilakukan sampai pada tahap construction.

**4. Pengujian**

**Pengujian Sistem**

Untuk mengetahui kualitas dari aplikasi yang dibuat maka akan dilakukan pengujian menggunakan *product metric* adapun *metric* yang digunakan pada pengujian kali ini adalah *Complexity Cyclomatic (CC), CommentPercentage, Weighted Method per Class (WMC)*. Adapun tujuan dari pengukuran ini adalah untuk menganalisis hasil dari perhitungan product metric terhadap kualitas perangkat lunak

**Skenario Pengujian**

Skenario pengujian dibedakan berdasarkan tiap metric, untuk metric pertama yaitu complexity cyclomatic (CC) menggunakan rumus

$$N = \text{jumlah method} - \text{jumlah kelas} + 2$$

Untuk method yang akan dihitung adalah method pada kelas petugas. Sedangkan untuk method pada kelas lain akan dihitung, namun hanya akan menampilkan data akhirnya saja.

Skenario pengujian untuk metric comment percentage akan dilakukan menggunakan rumus

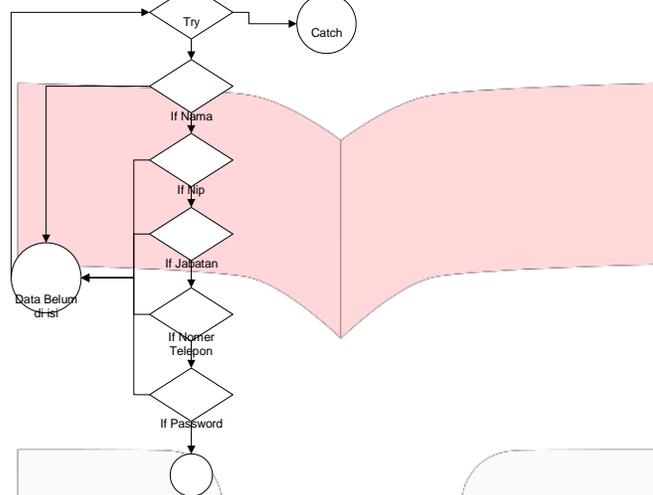
$$\frac{\text{jumlah total koment}}{\text{jumlah total baris}}$$

Dan skenario pengujian untuk metric *weighted method per class (WMC)* akan dilakukan perhitungan kepada setiap method yang ada pada class dan menampilkan ke dalam bentuk grafik.

**Hasil Pengujian**

**Complexity Cyclomatic (CC)**

Untuk melakukan perhitungan CC, cukup melihat percabangan yang terdapat didalam method tersebut. Setelah itu di gambarkan arah alur pengkondisian untuk melewati percabangan tersebut. Berikut penggambarannya



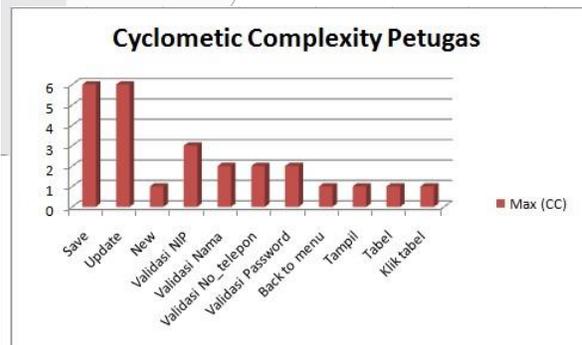
Berdasarkan perhitungan di atas maka didapatkan bahwa nodes berjumlah 9 dan edge berjumlah 13. Setelah mengetahui jumlah nodes dan edge maka nilai tersebut dimasukan ke dalam rumus. Dan berikut hasil perhitungannya

$$N - E + 2 = 13 - 9 + 2$$

Maka hasil akhirnya adalah 6. method save pada kelas petugas memiliki nilai cyclomatic complexity sebesar 6. Selanjutnya akan menghitung method lain yang terdapat pada class petugas

**Tabel.3 Method Class Petugas**

	Method	Max (CC)
PETUGAS	Save	6
	Update	6
	New	1
	Validasi NIP	3
	Validasi Nama	2
	Validasi No Telepon	2
	Validasi Password	2
	Back To Menu	1
	Tampil	1
	Tabel	1
	Klik Tabel	1
	<b>Total</b>	<b>26</b>



**GambarError! No text of specified style in document..6 Grafik Cyclomatic**

**Comment Percentage**

Untuk perhitungan comment percentage dapat menggunakan rumus

$$Comment\ Percentage = \frac{\text{jumlah total koment}}{\text{jumlah total baris}}$$

Jumlah total baris yang dibagi tidak termasuk baris yang kosong. Comment yang dimaksud adalah comment yang menjelaskan tentang suatu baris code ataupun suatu method. Semakin besar persenan comment percentage-nya maka semakin besar understandability dan maintainability dari suatu software[2]. Berikut perhitungannya

**Tabel 4 Comment Percentage**

Nama Class	Jumlah Total Koment	Jumlah Total Baris	Comment Percentage
Petugas	7	272	2,57 %
Nasabah	6	259	2,31 %
Kuasa	7	345	2,02 %
Box	2	158	1,26 %
Peminjaman	10	461	2,16 %
Kunjungan	8	364	2,19 %
Login	3	132	2,27 %
Cetak Laporan	6	249	2,40 %
Rekening	8	179	4,46 %

Hasil perhitungan didapatkan dengan menggunakan rumus hasil akhirnya dikali dengan 100 % sehingga memudahkan dalam perhitungan.

**Weighted Method Per Class**

Perhitungan *weighted method per class* (WMC) ini didapatkan dari total jumlah kompleksitas pada tiap class

**Tabel 5 Weighted Method Per Class**

Class	Jumlah Total CC
Petugas	26
Nasabah	26
Kuasa	31
Box	20
Peminjaman	39
Kunjungan	27
Login	3
Cetak Laporan	12
Rekening	6

**Analisis**

Pada tahap ini akan dihasilkan analisis terhadap hasil pengujian yang udah dilakukan

1. Berdasarkan hasil pengujian *cyclometric complexity* didapatkan bahwa nilai yang didapatkan sebagai berikut

**Tabel 6 Total Cyclometric Complexity**

Class	Total CC
Petugas	26
Nasabah	26
Kuasa	31
Box	20

Peminjaman	39
Kunjungan	27
Login	3
Cetak laporan	12
Rekening	6

Peminjaman merupakan kelas yang memiliki kompleksitas terbesar, sehingga akan lebih sulit untuk menangani kelas peminjaman. Dan kelas login merupakan kelas dengan kompleksitas terkecil. Untuk kelas peminjaman memiliki total *cyclometric complexity* 39 dengan rincian sebagai berikut

Tabel. 7 Cyclometric Complexity Peminjaman

PEMINJAMAN	Method	Max (CC)
	Save	6
	Update	6
	New	1
	Validasi NIP	3
	Validasi Nama	2
	Validasi Jabatan	2
	Validasi No telepon	2
	Validasi Password	2
	generate barcode	1
	tampil barcode	1
	close barcode	1
	print barcode	2
	cari box	2
	cari rekening	2
	tampil tabel nasabah	1
	tabel nasabah	1
	klik tabel nasabah	1
	Tampil tabel peminjaman	1
	tabel peminjaman	1
	Klik tabel peminjaman	1
	<b>Total</b>	<b>39</b>

Berdasarkan McCabe Cyclometric Complexity dibuatkan suatu *threshold* yang dikeluarkan oleh Software Engineering Institute terkait pembagian range dari cyclometric complexity<sup>[13]</sup>, berikut tabelnya

Tabel Error! No text of specified style in document. 8 Threshold McCabe Cyclometric Complexity

Cyclometric Complexity	Risk Evaluation
1-10	A simple module Without too much risk
11-20	A more complex module with moderate risk
21-50	A complex module of high risk
51 and greater	An untestable program of very high risk

Berdasarkan perbandingan antara hasil yang didapatkan dengan tabel *threshold*, diketahui bahwa nilai *cyclometric* terbesar adalah 6 yaitu pada method *save* belum terhitung kompleks dan tanpa banyak resiko didalamnya, hal ini meningkatkan nilai understandability, maintainability nya.

1. Berdasarkan hasil pengujian *comment percentage* didapatkan hasil yang relatif sangat rendah hanya sekitar 2%. Nilai tersebut didapatkan karena hanya beberapa bagian pada program yang diberikan koment. Hanya baris yang memiliki perhitungan dan memiliki percabangan yang diberikan koment, serta baris awal mula penulisan suatu method untuk menjelaskan kegunaan dari method tersebut. Menurut sumber<sup>[14]</sup> bahwa jumlah total koment dibawah 20% sebaiknya ditambahkan komentnya, dan total koment diatas 40% tidak dibutuhkan karena akan mempengaruhi dalam pembacaan program. Berdasarkan data yang didapatkan dan sumber Jarchitect diketahui bahwa aplikasi yang telah dibuat perlu ditambahkan jumlah komentnya.
2. Berdasarkan hasil pengujian *weighted method per class* bahwa *class* dengan jumlah total kompleksitas terbesar adalah peminjaman sebesar 39. Menurut sumber[15] bahwa limit terendah dari sebuah WMC di dalam RefactorIT adalah 1 karena setidaknya class tersebut memiliki 1 fungsi dan batas teratas adalah 50, hal ini akan berpengaruh terhadap *understandability*, *maintenance*, dan *reusability*. Berdasarkan hasil yang didapatkan dengan sumber menunjukan bahwa class peminjaman cukup kompleks dengan angka 39 di antara rentang 1 – 50, hal ini menunjukan apabila terdapat penambahan modul atau fungsionalitas pada class peminjaman dapat dijadikan satu class baru, sehingga tidak melebihi batas angka ideal.

### 5.Kesimpulan

Berikut merupakan kesimpulan yang didapat berdasarkan hasil analisis yang telah dilakukan yaitu :

- a. Metode perancangan RUP telah diimplementasikan menjadi aplikasi layanan SDB dan telah sesuai kebutuhan dari perusahaan
- b. Hasil pengujian terhadap aplikasi menunjukan aplikasi berkualitas cukup baik dengan nilai *cyclometric complexity* dan *weighted method per class* yang relatif rendah, namun memiliki nilai *comment percentage* rendah.

### Saran

Berikut merupakan saran yang dapat disampaikan dalam Tugas Akhir ini yaitu :

- a. Penelitian ini dapat dikembangkan menggunakan model proses lainnya seperti agile method.
- b. Penelitian ini dapat dikembangkan menggunakan metode pengujian lainnya seperti metode McCall.

### Daftar Pustaka

- [1] <http://id.wikipedia.org/wiki/Sistem> (diakses pada 30 Juli 2015).
- [2] <https://id.wikipedia.org/wiki/Bank> (diakses pada 27 Juli 2015)
- [3] [http://www.bankmandiri.co.id/corporate01/about\\_profile.asp](http://www.bankmandiri.co.id/corporate01/about_profile.asp) (diakses pada 27 Juli 2015)
- [4] <http://www.bankmandiri.co.id/article/680323188111.asp> (diakses pada 27 Juli 2015)
- [5] company, R. t. (1998). What is the rational unified process. Dalam R. t. company, *Rational Unified Process best practices for software development teams* (hal. 1). United States of America: ibm company.
- [6] company, R. t. (1998). What is the rational unified process. Dalam R. t. company, *Rational Unified Process best practices for software development teams* (hal. 3-7). United States of America: ibm company.
- [7] Perancangan dan Implementasi Scanner Barcode dengan Interface Suara untuk Tuna Netra Berbasis Android. (2013) Anna Siwi Ramadhani. D3 Teknik Telekomunikasi. Institut Teknologi Telkom
- [8] [http://id.wikipedia.org/wiki/Sistem\\_informasi](http://id.wikipedia.org/wiki/Sistem_informasi) (diakses pada 30 Juli 2015)
- [9] yasin, Sanjaya. "Pengertian Informasi Menurut Para Ahli Definisi". 30 Oktober 2014 <http://www.sarjanaku.com/2012/11/pengertian-informasi-menurut-para-ahli.html>
- [10] Roger S. Pressman, P. (2001). *Software Engineering A Practitioner's Approach Fifth Edition*. New York: McGraw-Hill.
- [11] Kan, S. H. (2003). *Metrics and Models In Software Quality Engineering Second Edition*. UK: Addison-Wesley Professional.
- [12] Dr. Linda H. Rosenberg, L. E. (n.d.). *Software Quality Metrics for Object-Oriented Environments. Software Quality Metrics for Object-Oriented Environments* .
- [13] [http://docs.klocwork.com/Insight-10.0/McCabe\\_Cyclomatic\\_Complexity](http://docs.klocwork.com/Insight-10.0/McCabe_Cyclomatic_Complexity) (diakses pada 06 Agustus 2015)
- [14] <http://www.jarchitect.com/Metrics> (diakses pada 06 Agustus 2015)

[15] <http://staff.unak.is/andy/StaticAnalysis0809/metrics/wmc.html> (diakses pada 06 Agustus 2015)

