



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pada kesempatan kerja magang ini, peserta magang ditempatkan pada divisi IT dalam bagian *Application Delivery Support (ADS)* PT Bank DBS Indonesia sebagai *programmer*. Selama periode kerja, peserta dibimbing oleh Bapak Hermawan Petra dan Bapak David Irawan selaku *Analyst* pada perusahaan tersebut. Proyek pembangunan modul aplikasi ini dipimpin oleh Bapak Riza Kurniawan sebagai kepala divisi IT – ADS.

3.2 Tugas yang Dilakukan

Dalam periode magang ini, penulis sebagai peserta magang bertugas untuk membuat suatu modul aplikasi berbasis *web* sesuai dengan *user requirements* yang sudah disetujui yang nantinya aplikasi tersebut akan digabungkan dengan *web application* utama yang sudah berfungsi. Modul aplikasi yang dibangun adalah modul *Auto Reconciliation* untuk ATM VISA. *Auto reconciliation* adalah sebuah sistem pencocokan laporan yang berisikan informasi transaksi ATM harian terhadap laporan dari sumber lainnya, dimana pada modul ini informasi yang dicocokkan hanya terbatas pada penarikan tunai pada jaringan VISA. *Auto reconciliation* menghasilkan *output* berupa laporan yang berisikan deskripsi, total, serta nominal dari transaksi yang cocok maupun tidak cocok dari proses pencocokan antara laporan-laporan transaksi tersebut.

Secara terperinci, tugas yang dilakukan oleh peserta magang dalam pembangunan aplikasi *auto reconciliation* untuk ATM Visa yang menjadi subjek utama penulisan laporan ini meliputi :

- a. Melakukan analisa *Business Requirements Documents*
- b. Merancang fungsional aplikasi
- c. Merancang tabel-tabel untuk keperluan aplikasi
- d. Membangun aplikasi (tahap pemrograman / *coding*)
- e. Melakukan *unit testing*

3.3 Tahapan Pelaksanaan Kerja Magang

Penerapan kerja secara konkrit dalam pembangunan aplikasi dijabarkan sebagai berikut :

1. Analisa *Business Requirements Documents* (BRD)

Untuk mengajukan permintaan pembangunan aplikasi pada Bank DBS Indonesia, *user* sebagai klien harus membuat BRD terlebih dahulu dan disetujui oleh pihak klien serta IT-ADS. Setelah tahapan tersebut selesai, bagian Project Management Office (PMO) akan membuat jadwal pengerjaan pembangunan aplikasi tersebut. Bekerja sebagai *programmer* pada pembangunan aplikasi *ATM Auto Reconciliation* dalam kesempatan kerja magang ini, BRD yang diterima telah disetujui oleh kedua pihak, yaitu pada sisi IT-ADS dan sisi klien. Oleh karena itu, BRD sudah bersifat final dan tidak dapat diubah di kemudian hari. BRD yang diberikan berbentuk sebuah dokumen yang berisikan daftar data yang digunakan dalam proses *reconciliation* beserta penjelasan kategori dan kriteria data yang digunakan, langkah kerja fungsional *reconciliation*, serta contoh laporan dan memo yang menjadi *output* dari aplikasi ini.

2. Perancangan Fungsional Aplikasi

Tahap selanjutnya adalah perancangan aplikasi dari BRD yang telah disetujui. Dari setiap *user requirements* yang dituangkan dalam dokumen tersebut dibuat rancangannya untuk menjadi fungsional-fungsional aplikasi yang dapat diterapkan ke dalam

3. Perancangan Database dan Tabel

Untuk melakukan fungsional aplikasi, data-data yang dibutuhkan disimpan dalam sebuah database yang terdiri dari beberapa tabel. Tabel-tabel tersebut dirancang untuk dapat menampung data yang berasal dari beberapa sumber data mentah dalam bentuk teks. Data-data dalam tabel tersebut nantinya digunakan dalam proses kerja fungsional dalam aplikasi.

4. Pembangunan Aplikasi (*coding*)

Hasil dari perancangan-perancangan yang dilakukan sebelumnya dibuat ke dalam bentuk aplikasi. Pembangunan aplikasi berbasis web ini

menggunakan ASP.NET dengan Microsoft Visual Studio 2010 yang terkoneksi dengan database Microsoft SQL Server 2008 untuk penyimpanan data. Selama pembangunan aplikasi berlangsung, analisa terhadap *user requirements* masih dilakukan untuk memperjelas fungsional yang harus diterapkan.

5. *Unit Testing*

Selama proses pembangunan aplikasi dilakukan pengujian untuk memastikan kinerja fungsional yang diterapkan sudah bekerja dengan benar dan menghasilkan *output* yang tepat. Pengujian ini masih dilakukan secara pribadi oleh programmer, dalam hal ini adalah penulis sendiri sebagai peserta magang. Hasil dari *unit testing* dicocokkan dengan data dari user, jika tidak cocok tentunya akan dilakukan perbaikan untuk mencapai hasil yang dibutuhkan.

3.4. Uraian Pelaksanaan Kerja Magang

Berikut ini adalah uraian pelaksanaan kerja magang yang dilakukan oleh peserta magang dalam periode pembangunan aplikasi *auto reconciliation* untuk jaringan VISA pada PT Bank DBS Indonesia.

3.4.1 Kebutuhan Pengguna

Kebutuhan pengguna atau *user requirements* diperoleh dari *Business Requirements Documents* yang telah disetujui. Secara umum, kebutuhan pengguna dibagi menjadi tiga bagian besar, yaitu :

1. Memasukkan data

Data yang digunakan dalam proses *auto reconciliation* dimasukkan secara manual oleh *user* pada saat akan memulai proses tersebut. Data-data tersebut berupa data teks laporan yang berasal dari tiga saluran (*channel*) berbeda. Berikut adalah rincian kebutuhan *user* saat menjalan proses *input data*.

- a. Membaca *file* teks yang berasal dari laporan VISA langsung, di mana hanya data dengan *report ID* tertentu saja yang akan dibaca dan kemudian data-data dimasukkan ke dalam *database*.
- b. Membaca *file* teks yang berasal dari laporan sistem IST/Switch, di mana hanya data yang memiliki *Issuer Bank Identification Number* (*Issuer BIN*) dan atau *Acquirer Bank Identification Number* (*Acquirer BIN*) milik VISA saja yang akan dibaca dan dimasukkan ke dalam *database*.
- c. Membaca *file* teks yang berasal dari laporan sistem Finacle, di mana laporan VISA sudah menjadi *file* tersendiri yang akan dibaca dan data kemudian dimasukkan dalam *database*.
- d. Membaca *file General Ledger* yang didapatkan secara internal oleh *user*, di mana data yang dibaca hanyalah data yang memiliki ID tertentu dan kemudian dimasukkan dalam *database*.
- e. Untuk membaca *file*, *user* dapat memilih *file* yang diinginkan melalui window explorer.

2. Proses *auto reconciliation*

Proses rekonsiliasi merupakan kerja utama dari aplikasi yang dibangun. Proses ini akan mengakses seluruh data yang telah dimasukkan pada prosedur sebelumnya. Kebutuhan untuk menjalankan proses rekonsiliasi adalah sebagai berikut.

- a. Menentukan tanggal yang menjadi kriteria data yang akan diproses.
- b. Menentukan jumlah hari untuk mengikutsertakan data yang masih *pending* pada proses *reconciliation* yang akan dilaksanakan.

3. Pembuatan laporan dan *output* lainnya

Tujuan utama proses *auto reconciliation* adalah menghasilkan laporan dengan cepat dan tepat. Oleh karena itu, hasil proses *reconciliation* yang dilakukan dapat langsung menghasilkan laporan beserta beberapa *output* lainnya. Kebutuhan *user* dalam hal laporan dijelaskan sebagai berikut.

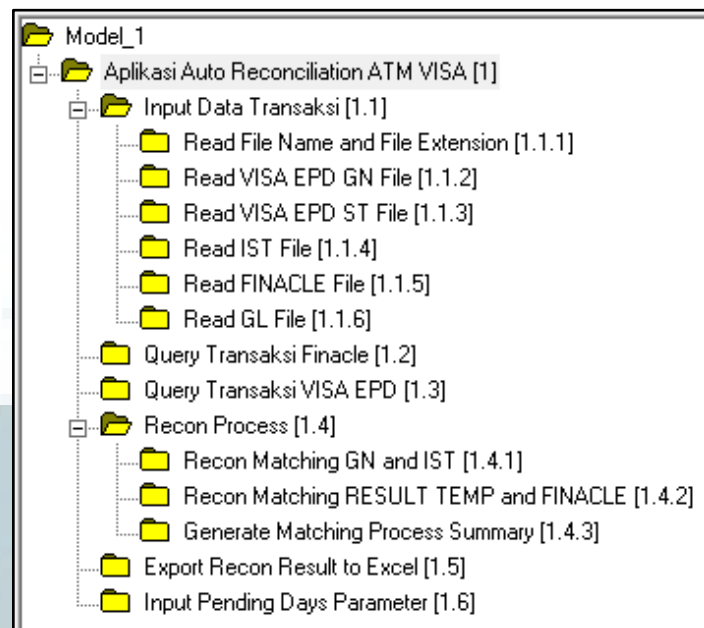
- a. Menghasilkan laporan tertulis hasil *reconciliation* untuk transaksi yang cocok atau memiliki status *matched*
- b. Menghasilkan laporan tertulis hasil *reconciliation* untuk transaksi yang tidak cocok atau memiliki status *pending*
- c. Menghasilkan laporan tertulis transaksi mana saja yang diikutsertakan pada proses *reconciliation* pada saat itu
- d. Menghasilkan memo tertulis internal untuk keperluan bank
- e. Menghasilkan laporan secara umum mengenai hasil *reconciliation* berupa tampilan pada halaman *web*.

Sedangkan untuk kebutuhan mengenai beberapa *output* lainnya yang tidak termasuk dalam laporan hasil proses *reconciliation* adalah sebagai berikut.

- a. *Review* data yang berasal dari *channel* VISA berdasarkan tanggal transaksi, ditampilkan pada halaman *web*
- b. *Review* data yang berasal dari *channel* IST/Switch berdasarkan tanggal transaksi, ditampilkan pada halaman *web*
- c. *Review* data yang berasal dari *channel* Finacle berdasarkan tanggal transaksi, ditampilkan pada halaman *web*

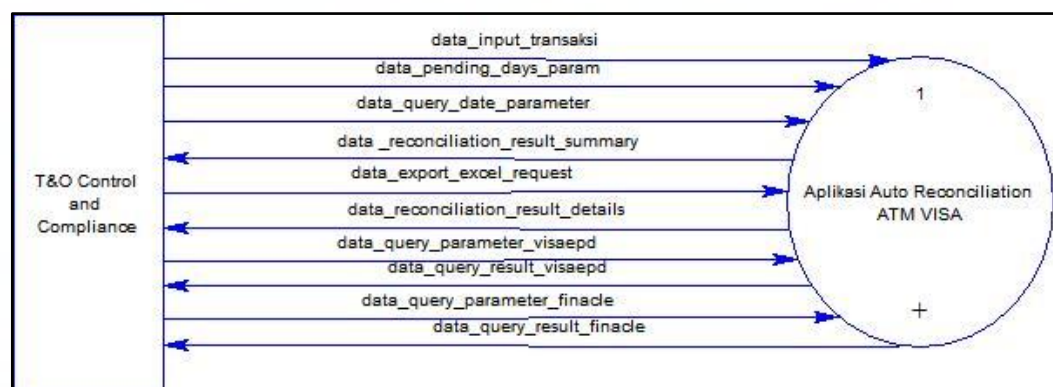
3.4.2 Data Flow Diagram (DFD)

Fungsional sistem aplikasi *auto reconciliation* yang dibangun melibatkan beberapa aliran data sepanjang proses tersebut berlangsung. Gambar 3.1 berikut ini adalah *process tree* dari aplikasi *auto reconciliation* untuk ATM VISA.

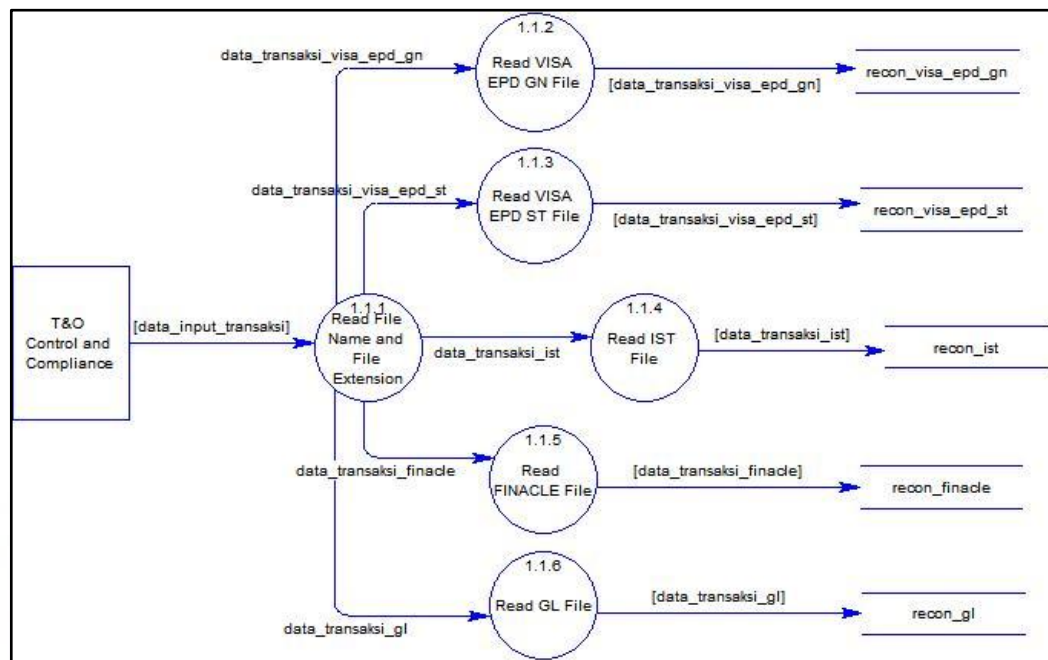


Gambar 3.1 Process Tree dari Aplikasi *Auto Reconciliation*

Berbagai data digunakan dalam proses yang dijalankan pada aplikasi *auto reconciliation* ATM VISA ini. Data-data digunakan mulai dari proses *input* data-data transaksi, proses pencocokan, hingga akhirnya menghasilkan informasi kesimpulan dan laporan. Gambar 3.2 berikut ini adalah *context diagram* dari aplikasi ini.



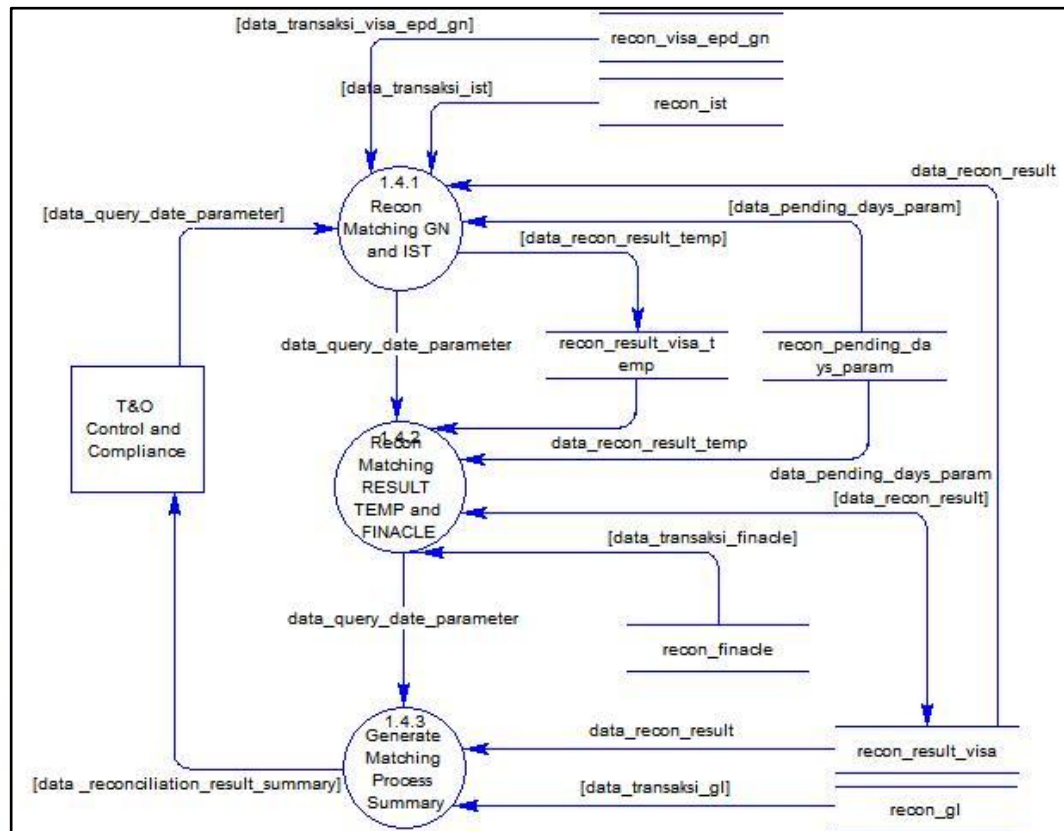
Gambar 3.2 *Context Diagram* aplikasi *Auto Reconciliation*



Gambar 3.4 DFD Level 2 *subprocess* Input Transaksi aplikasi *Auto Reconciliation*

Pada gambar 3.5 berikut ini dijelaskan perincian mengenai proses *reconciliation* yang ada pada aplikasi *Auto Reconciliation*.

U
M
M
N



Gambar 3.5 DFD Level 2 *subprocess Recon Process* aplikasi *Auto Reconciliation*

3.4.3 Struktur Tabel

Keseluruhan fungsional aplikasi *auto reconciliation* bergantung pada data-data yang disimpan pada beberapa tabel. Tabel-tabel yang ada menampung data yang dibutuhkan pada saat proses *reconciliation* itu sendiri berlangsung, dan juga tabel menjadi tempat untuk menampung hasil dari proses tersebut. Berikut adalah tabel-tabel yang digunakan oleh aplikasi *auto reconciliation* tersebut.

Tabel VISA GN

Tabel *recon_visa_epd_gn* berfungsi untuk menyimpan data transaksi harian VISA yang berasal dari *file* teks dengan kode GN, yang mana *file* ini berisikan data transaksi secara terperinci. Tabel 3.1 berikut ini adalah struktur tabel dari tabel *recon_visa_epd_gn*.

Tabel 3.1 Struktur Tabel VISA GN

KOLOM	TIPE DATA	CONSTRAINT	KETERANGAN
cardNo	Varchar(20)	Not null	Nomor kartu yang digunakan untuk transaksi
refNumber	Varchar(20)	Not null	Nomor referensi unik dari transaksi yang dilakukan
date	Varchar(15)	Not null	Tanggal terjadinya transaksi
amount	Varchar(10)	Not null	Nominal transaksi
currency	Varchar(3)	Not null	Mata uang yang digunakan saat transaksi
typeOfVisa	Varchar(10)	Not null	Tipe transaksi visa (<i>issuer</i> atau <i>acquirer</i>)
transType	Varchar(2)	Not null	Tipe transaksi visa (debit atau kredit)

Tabel VISA ST

Tabel recon_visa_epd_st berfungsi untuk menyimpan data transaksi VISA yang berasal dari *file* teks dengan kode ST, yang mana *file* ini berisikan data transaksi yang sudah disimpulkan oleh jaringan tersebut. Tabel 3.2 berikut ini adalah struktur tabel dari tabel recon_visa_epd_st.

Tabel 3.2 Struktur Tabel VISA ST

KOLOM	TIPE DATA	CONSTRAINT	KETERANGAN
date	Varchar(10)	Not null	Tanggal transaksi
netSettAmount	Varchar(10)	Not null	Nominal <i>settlement</i> transaksi pada tanggal tersebut
currency	Varchar(3)	Not null	Mata uang <i>settlement</i> transaksi

Tabel IST

Tabel recon_ist berfungsi untuk menyimpan data transaksi terperinci yang berasal dari *report file* teks yang didapatkan dari *channel* IST/Switch. Tabel 3.3 berikut ini adalah struktur tabel dari tabel recon_ist.

Tabel 3.3 Struktur Tabel IST

KOLOM	TIPE DATA	CONSTRAINT	KETERANGAN
PAN	Varchar(20)	Not null	Nomor kartu yang digunakan untuk transaksi
refNumber	Varchar(20)	Not null	Nomor referensi unik dari transaksi yang dilakukan
transDate	Varchar(10)	Not null	Tanggal terjadinya transaksi
settAmount	Varchar(10)	Not null	Nominal transaksi
settCurrency	Varchar(3)	Not null	Kode mata uang yang digunakan saat transaksi
issuerBin	Varchar(15)	Not null	Kode <i>Bank Identification Number</i> yang menjadi <i>issuer</i> pada transaksi
acquirerBin	Varchar(15)	Not null	Kode <i>Bank Identification Number</i> yang menjadi <i>acquirer</i> pada transaksi

Tabel Finacle

Tabel recon_finacle berfungsi untuk menyimpan data transaksi secara rinci yang berasal dari *report file* teks yang didapatkan dari *channel* Finacle. Tabel 3.4 berikut ini adalah struktur tabel dari tabel recon_finacle.

Tabel 3.4 Struktur Tabel Finacle

KOLOM	TIPE DATA	CONSTRAINT	KETERANGAN
cardNumber	Varchar(20)	Not null	Nomor kartu yang digunakan untuk transaksi
refNumber	Varchar(20)	Not null	Nomor referensi unik dari transaksi yang dilakukan
transDate	Varchar(10)	Not null	Tanggal terjadinya transaksi
transAmount	Varchar(10)	Not null	Nominal transaksi
Currency	Varchar(3)	Not null	Mata uang yang digunakan saat transaksi
transParticular	Varchar(50)	Not null	Deskripsi data transaksi

Tabel GL

Tabel recon_gl berfungsi untuk menyimpan data dari laporan *General Ledger* yang dimiliki oleh internal bank yang dibutuhkan untuk keperluan *reconciliation*.

Tabel 3.5 ini adalah struktur tabel dari tabel recon_gl.

Tabel 3.5 Struktur Tabel GL

KOLOM	TIPE DATA	CONSTRAINT	KETERANGAN
asofDate	Varchar(10)	Not null	Tanggal dibuatnya data GL
glid	Varchar(10)	Not null	ID penunjuk tipe transaksi
balance	Varchar(15)	Not null	Nominal pada transaksi
currency	Varchar(3)	Not null	Informasi mata uang untuk transaksi
description	Varchar(50)	Not null	Deskripsi yang menjelaskan ID transaksi

Tabel Parameter Pending Days

Tabel recon_pending_days_param berfungsi untuk menyimpan total lamanya hari yang menjadi parameter bagi transaksi-transaksi yang sebelumnya masih berstatus *pending* untuk diikutsertakan kembali dalam proses *reconciliation* yang dilakukan selanjutnya. Tabel 3.6 berikut ini adalah struktur tabel dari tabel recon_pending_days_param.

Tabel 3.6 Struktur Tabel Parameter Pending Days

KOLOM	TIPE DATA	CONSTRAINT	KETERANGAN
editDate	Date	Not null	Tanggal yang menunjuk pada hari penggantian <i>parameter</i> oleh <i>user</i>
pendingDays	Integer	Not null	Jumlah hari yang menjadi parameter

Tabel Result Temporary VISA

Tabel *recon_result_visa_temp* berfungsi untuk menyimpan hasil proses *reconciliation* pertama. Tabel 3.7 berikut ini adalah struktur tabel dari tabel *recon_result_visa_temp*.

Tabel 3.7 Struktur Tabel Result Temporary VISA

KOLOM	TIPE DATA	CONSTRAINT	KETERANGAN
cardNumber	Varchar(20)	Not null	Nomor kartu yang digunakan untuk transaksi
transDate	Varchar(10)	Not null	Tanggal terjadinya transaksi
refNumber	Varchar(20)	Not null	Nomor referensi unik dari transaksi yang dilakukan
amount	Varchar(10)	Not null	Nominal transaksi
typeOfVisa	Varchar(10)	Not null	Tipe transaksi VISA (<i>issuer</i> atau <i>acquirer</i>)
transParticular	Varchar(50)	Not null	Deskripsi data transaksi
reconStatus	Varchar(10)	Not null	Status yang didapat setelah proses <i>reconciliation</i>
pendingStatus	Varchar(10)	Not null	Status yang menjelaskan di mana ketidakcocokan terjadi pada proses <i>reconciliation</i>

Tabel Result VISA

Tabel *recon_result_visa* berfungsi untuk menyimpan hasil proses *reconciliation* yang kedua, yang mana hasil proses *reconciliation* sudah bersifat *final*. Tabel 3.8 berikut ini adalah struktur tabel dari tabel *recon_result_visa*.

Tabel 3.8 Struktur Tabel Result VISA

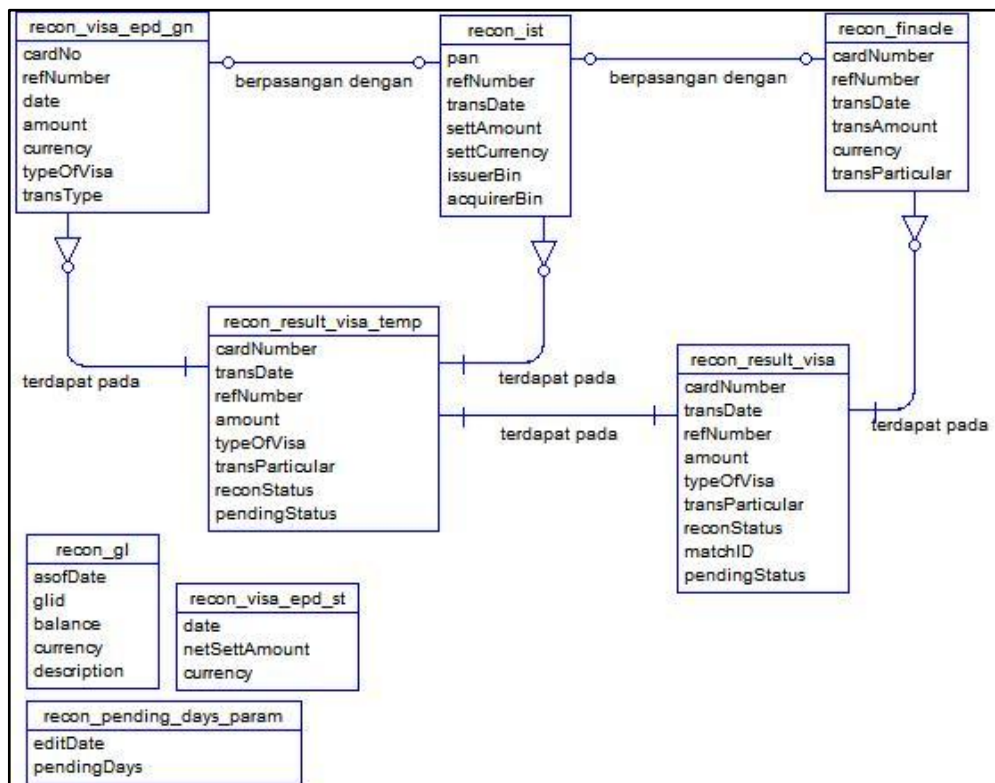
KOLOM	TIPE DATA	CONSTRAINT	KETERANGAN
cardNumber	Varchar(20)	Not null	Nomor kartu yang digunakan untuk transaksi
transDate	Varchar(10)	Not null	Tanggal terjadinya transaksi

refNumber	Varchar(20)	Not null	Nomor referensi unik dari transaksi yang dilakukan
amount	Varchar(10)	Not null	Nominal transaksi
typeOfVisa	Varchar(10)	Not null	Tipe transaksi VISA (<i>issuer</i> atau <i>acquirer</i>)
transParticular	Varchar(50)	Not null	Deskripsi data transaksi
reconStatus	Varchar(10)	Not null	Status yang didapat setelah proses <i>reconciliation</i>
matchID	Varchar(20)	Not null	ID yang didapatkan jika transaksi cocok saat proses <i>reconciliation</i>
pendingStatus	Varchar(10)	Not null	Status yang menjelaskan di mana ketidakcocokan terjadi pada proses <i>reconciliation</i>

3.4.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram yang digambarkan pada gambar 3.6 berikut ini hanya bersifat *logical* sesuai dengan penjelasan dari *user* serta keputusan dari *senior programmer* mengenai relasi yang komposit sehingga tidak perlu dibuatkan key yang menghubungkan antara data-data laporan tersebut. Relasi yang memperlihatkan hubungan setiap data hanya terlihat pada *query syntax* yang mengakses data pada tabel-tabel yang terdapat dalam database.

U
M
N



Gambar 3.6 *Entity Relationship Diagram* aplikasi *auto reconciliation*

3.4.5 Rancangan Tampilan Antarmuka (*Interface*)

Tampilan antarmuka pada sebuah aplikasi haruslah mengakomodasi kebutuhan user dengan tepat dan efisien. Berikut ini adalah gambar rancangan untuk tampilan antarmuka aplikasi. Halaman *login* dirancang untuk mengakomodasi user dalam memasukkan *user name* dan *password* serta memberikan otorisasi terhadap user untuk masuk ke dalam halaman utama dan menggunakan fitur-fitur yang disediakan berdasarkan hak aksesnya. Gambar 3.7 berikut ini adalah rancangan tampilan antarmuka untuk halaman *login*.

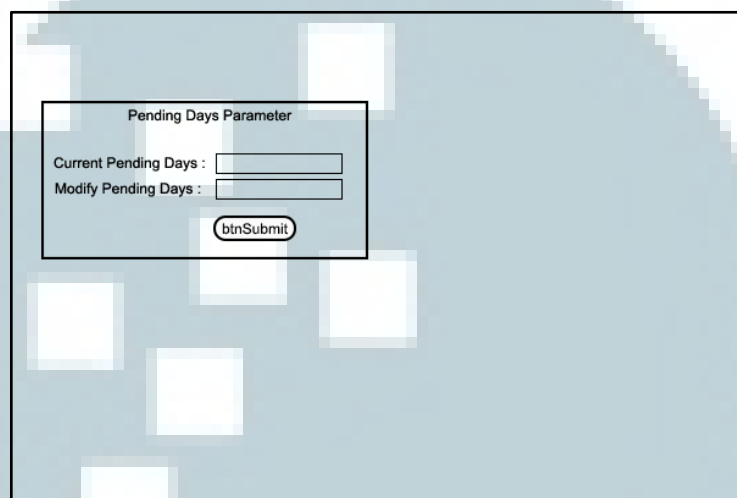
Gambar 3.7 Rancangan Halaman *login*

Halaman *reconciliation process* dirancang untuk menampung kebutuhan user dalam melakukan input data-data transaksi untuk kebutuhan proses *reconciliation* serta untuk memicu *user* menjalankan proses *reconciliation* dan membuat hasil proses *reconciliation* menjadi laporan. Gambar 3.8 berikut ini adalah rancangan tampilan antarmuka untuk halaman *reconciliation process*.

Summary Information DBSS	Summary Information VISA-USD	Summary Information VISA-IDR
File VISA-GN Select <input type="text"/>	Start Date <input type="text"/> End Date <input type="text"/>	
File VISA-ST Select <input type="text"/>	<input type="button" value="btn Recon VISA"/> <input type="button" value="btn Recon DBSS"/>	
File IST Select <input type="text"/>	<input type="button" value="btn Export Excel"/>	
File FINACLE Select <input type="text"/>		
<input type="button" value="Upload"/>		

Gambar 3.8 Rancangan Halaman *reconciliation process*

Halaman *set parameter* dirancang untuk mengakomodasi *user* dalam melihat *pending days parameter* yang sedang berlaku dan juga mengubah *pending days parameter* itu sesuai kebutuhan. Gambar 3.9 berikut ini adalah rancangan tampilan antarmuka untuk halaman *set parameter*.



Pending Days Parameter

Current Pending Days :

Modify Pending Days :

Gambar 3.9 Rancangan Halaman *Set Parameter*

Halaman *VISA Network – VISA View* dirancang untuk mengakomodasi *user* dalam melihat data-data transaksi yang telah dimasukkan ke dalam *database*. Data-data transaksi yang ditampilkan pada halaman ini merupakan data transaksi yang berasal dari laporan transaksi dari saluran VISA. Gambar 3.10 berikut ini adalah rancangan tampilan antarmuka untuk halaman *VISA Network – VISA View*.

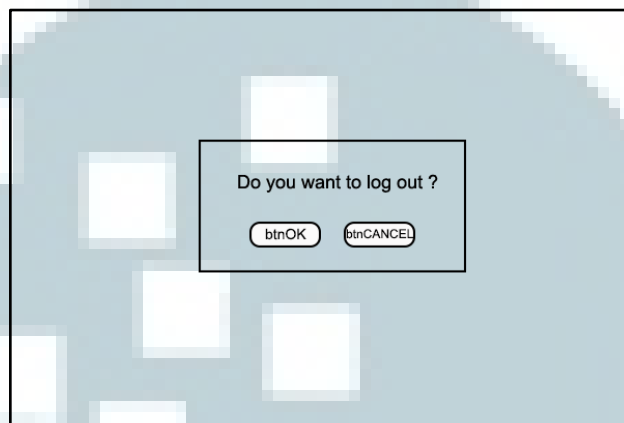
Gambar 3.10 Rancangan Halaman *VISA Network – VISA View*

Halaman *VISA Network – Finacle View* dirancang untuk mengakomodasi *user* dalam melihat data-data transaksi yang telah dimasukkan ke dalam *database*. Data-data transaksi yang ditampilkan pada halaman ini merupakan data transaksi yang berasal dari laporan transaksi dari saluran Finacle. Gambar 3.11 berikut ini adalah rancangan tampilan antarmuka untuk halaman *VISA Network – Finacle View*.

Gambar 3.11 Rancangan Halaman *VISA Network – Finacle View*

Halaman *logout* dirancang untuk melakukan verifikasi terhadap *user* yang akan melakukan *logout*. Apabila *user* menekan tombol ok maka *user* akan keluar

dari halaman kerja. Akan tetapi, apabila *user* menekan tombol *cancel* maka *user* akan dikembalikan ke halaman kerja. Gambar 3.12 berikut ini adalah rancangan tampilan antarmuka untuk halaman *logout*.



Gambar 3.12 Rancangan Halaman *Logout*

3.4.6 Hasil Rancangan Tampilan Antarmuka (*Interface*)

Aplikasi ini diakses oleh *user* melalui sebuah halaman *web*. Berikut adalah penjelasan tampilan antarmuka dari aplikasi *auto reconciliation* yang dibangun berdasarkan hasil perancangan.

- **Halaman *Login***



Gambar 3.13 Halaman *Login*

Halaman *login*, yang ditunjukkan pada gambar 3.13, digunakan oleh *user* yang memiliki hak akses untuk masuk dan menggunakan aplikasi ini. Halaman *login* untuk *auto reconciliation* sama untuk halaman *login* yang digunakan oleh modul-modul lainnya, karena modul-modul tersebut tergabung menjadi sebuah *engine* yang besar. Melalui validasi *login*, *user* akan diarahkan kepada modul yang diperbolehkan saja sesuai dengan hak akses yang dimilikinya.

- **Halaman *reconciliation process***

The screenshot displays the DBS Reconciliation Process interface. At the top, there are navigation tabs for 'ENGINE', 'DBSS', and 'VISA'. The main content area is titled 'RECON RESULT SUMMARY INFORMATION' and is divided into three columns: 'DBSS SGD', 'VISA USD', and 'VISA IDR'. Each column contains a table of reconciliation data, including 'Opening Balance', 'Matching', 'Pending', 'Settlement', and 'Ending Balance', all of which are currently set to 0. Below the summary table, there are two main sections: 'UPLOAD PROCESS' and 'RECON PROCESS'. The 'UPLOAD PROCESS' section includes a 'Please input data:' label and a file upload button labeled 'Add Files'. The 'RECON PROCESS' section includes a 'Please select recon date:' label, 'FROM' and 'TO' date pickers, and buttons for 'Recon Process', 'Export DBSS', and 'Export VISA'.

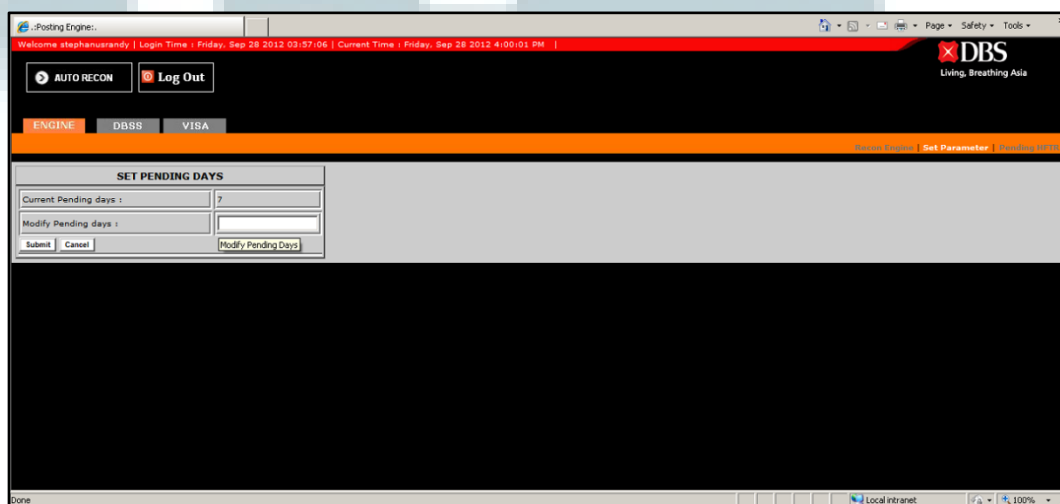
Gambar 3.14 Halaman *Reconciliation Process* Bagian Pertama

This screenshot shows the second part of the DBS Reconciliation Process interface. The 'UPLOAD PROCESS' section is expanded, showing three file upload areas: 'VISA', 'DBSS', and 'IST & Finacle & HFTR & GL', each with an 'Add Files' button. Below these is an 'Upload Files' button. The 'RECON PROCESS' section is also visible, showing the 'Recon DBSS Process' button. The interface includes the same navigation tabs and summary table as seen in the previous screenshot.

Gambar 3.15 Halaman *Reconciliation Process* Bagian Kedua

Halaman kerja *reconciliation process*, yang ditunjukkan pada gambar 3.14 dan gambar 3.15, terbagi menjadi dua bagian besar. Pertama, pada bagian atas halaman *web* terdapat *summary information* yang ditampilkan setelah sebuah proses *reconciliation* selesai dijalankan. Bagian tersebut akan menampilkan data-data hasil *reconciliation* yang dijalankan pada saat itu. Kedua, pada bagian bawah dari halaman *web* terdapat *interface* untuk melakukan *input* data-data yang akan digunakan untuk *reconciliation*, menentukan tanggal data yang akan diikutsertakan dalam *reconciliation*, serta *button* untuk menjalankan pembuatan laporan langsung menjadi *file* Microsoft Excel (.xls).

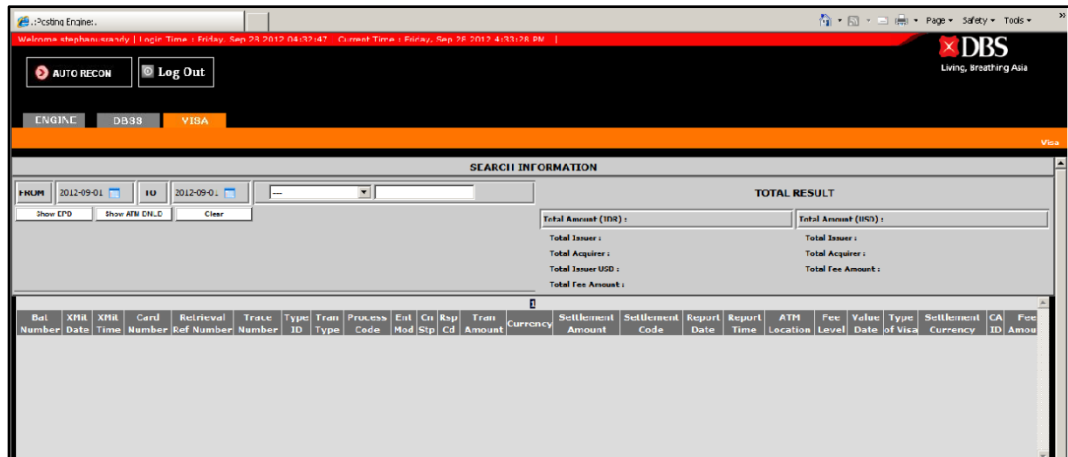
- **Halaman Set Parameter**



Gambar 3.16 Halaman *Set Parameter*

Halaman pada gambar 3.16 ini digunakan untuk mengatur jarak hari sebuah transaksi yang berstatus *pending* untuk diikutsertakan dalam proses *reconciliation* berikutnya.

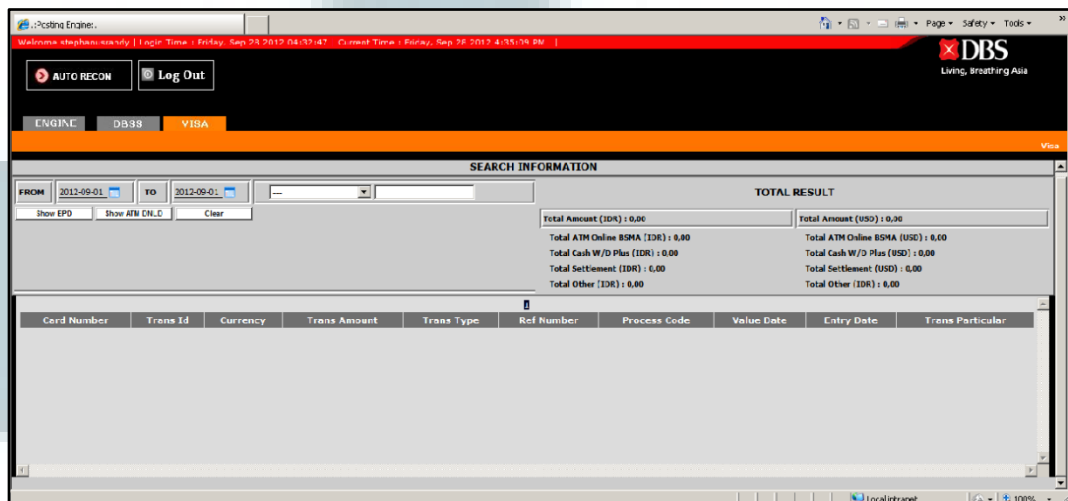
- Halaman *VISA Network – VISA View*



Gambar 3.17 Halaman *VISA Network – VISA View*

Halaman pada gambar 3.17 ini digunakan untuk melihat kembali data-data transaksi jaringan VISA yang berasal dari laporan VISA yang telah dimasukkan ke dalam *database*. *User* dapat memilih data transaksi yang ditampilkan berdasarkan tanggal serta kriteria-kriteria lainnya seperti *by Amount*, *by Card Number*, dan *by Reference Number*.

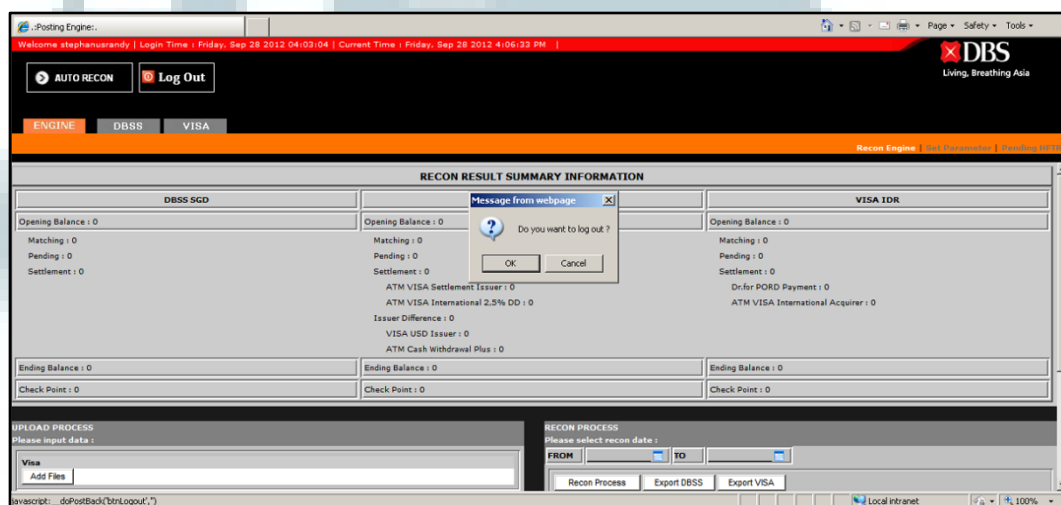
- Halaman *VISA Network – Finacle View*



Gambar 3.18 Halaman *VISA Network – Finacle View*

Halaman yang ditunjukkan pada gambar 3.18 ini digunakan oleh *user* untuk melihat kembali data-data transaksi jaringan VISA yang berasal dari *channel* Finacle yang telah dimasukkan ke dalam *database*. *User* dapat memilih data transaksi yang ditampilkan berdasarkan tanggal serta kriteria-kriteria lainnya seperti *by Amount*, *by Card Number*, dan *by Reference Number*.

- **Halaman Logout**



Gambar 3.19 Halaman *PopUp Logout*

Halaman aplikasi *web*, yang ditunjukkan oleh gambar 3.19, yang bersifat *popup* ini bertujuan untuk memastikan kembali apakah *user* ingin *logout* dari aplikasi ini. Jika *user* memilih *ok*, maka *user* akan keluar dari aplikasi *auto reconciliation* dan kembali ke tampilan *login*.

3.4.7 Sarana yang Digunakan

Dalam pembangunan aplikasi ini, beberapa sarana yang digunakan untuk mendukung proses pembangunan adalah sebagai berikut.

1. ASP digunakan untuk membuat rancangan tampilan antarmuka yang menjadi sarana komunikasi antara *user* dengan fungsional sistem dalam aplikasi ini.

2. VB.NET digunakan sebagai bahasa pemrograman utama untuk membuat fungsional sistem dalam aplikasi, mengerjakan proses-proses yang dikerjakan oleh sistem, seperti membaca *file*, mengakses *database*, melakukan *query* terhadap tabel pada *database*, serta menjalankan proses *reconciliation*.
3. .NET Framework digunakan sebagai *library* yang mendukung serta mempermudah pembangunan fungsionalitas di dalam aplikasi.
4. Microsoft Visual Studio 2010 sebagai *Integrated Development Environment* (IDE) utama yang berguna untuk membangun aplikasi dalam bentuk kode-kode dalam bahasa pemrograman.
5. SQL Server Management Studio yang terletak pada sisi *server* digunakan untuk mempermudah pembuatan dan pengaturan *database* beserta tabel-tabel dan deskripsinya masing-masing.
6. *Internet Information Services* (IIS) digunakan sebagai *web-server application* dalam tahap *unit testing* yang masih dilakukan secara lokal.

3.5 Kendala yang Ditemukan

Berbagai kendala yang ditemukan oleh peserta magang selama proses kerja magang membangun aplikasi untuk *auto reconciliation* ini adalah sebagai berikut.

1. *Business Requirements Document* (BRD) yang dibuat oleh pihak *user* bersifat *high-level*. Penjelasan secara teknis masih secara garis besar saja. Oleh karena itu, pembangunan aplikasi terhambat karena membutuhkan penjelasan lebih lanjut mengenai kinerja fungsional pada tingkat yang lebih dalam.
2. Revisi dan penjelasan lebih lanjut mengenai kinerja fungsional pada tingkat yang lebih dalam berlangsung pada saat peserta magang sudah menyelesaikan *coding* atas suatu fungsional dalam aplikasi berdasarkan BRD saja. Seringkali, revisi tersebut membuat bagian fungsional dalam aplikasi harus diubah lagi dan menyebabkan kemunduran pada pembangunan aplikasi.

3. Adanya *programmer* yang lebih dahulu tergabung dalam pembangunan ini yang mengundurkan diri dan tanggung jawab dilimpahkan kepada *programmer* lain, yakni peserta magang.
4. *User* juga memiliki kesibukannya tersendiri dan terkadang cukup sulit untuk meminta waktu penjelasan mengenai *user requirements* lebih lanjut.

3.6 Solusi atas Kendala yang Ditemukan

Solusi yang digunakan untuk mengatasi kendala-kendala yang menghambat proses pembangunan aplikasi selama periode kerja magang adalah sebagai berikut.

1. Melaporkan kepada pembimbing lapangan dan meminta waktu kepada *user* untuk memberikan penjelasan lebih lanjut secara intensif, serta membuat catatan mengenai teknis yang dijelaskan oleh *user*.
2. Mempelajari apa yang telah dilakukan oleh *programmer* sebelumnya terhadap aplikasi dan kemudian menentukan apa yang masih dapat digunakan serta bagian mana yang harus diganti untuk memenuhi kebutuhan pengguna aplikasi.
3. Meminta bantuan dari bagian lain yang masih memiliki keterkaitan dengan sistem *reconciliation* selain *user* untuk menjelaskan hal-hal praktis.

UMMN