

## **BAB VI**

### **PENUTUP**

#### **VI.1 Kesimpulan**

Berdasarkan dari pembahasan-pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan dari Tugas Akhir ini :

1. Sistem informasi yang mensimulasikan display data laporan keuangan neraca saldo, neraca bank, laporan laba-rugi dan laporan komitmen dan kontijensi pada setiap bank cabang atau pusat telah berhasil dibuat.
2. Konsep SOA telah diimplementasikan pada sistem informasi pada sistem bank pusat yang menangani pengambilan data lintas platform dengan *web service* pada setiap bank cabang untuk membuat laporan bank gabungan.

#### **VI.2 Saran**

Beberapa saran dan masukan yang dapat disampaikan penulis terhadap pembuatan SIP-IPAB ini dimasa yang akan datang adalah :

1. Sistem ini diharapkan dapat dikembangkan lebih jauh dengan memaksimalkan proses bisnis kegiatan perbankan, contohnya penambahan analisis kesehatan perbankan dengan data dari laporan keuangan perbankan atau penambahan laporan-laporan transaksi perbankan secara detail

2. Sistem ini diharapkan dapat dikembangkan lebih interaktif dengan memaksimalkan tools di Oracle JDeveloper yang menyediakan fitur graph yang dapat membantu pengguna mengamati perkembangan keuangan perbankan berdasarkan rentang waktu tertentu
3. Sistem ini diharapkan dapat dikembangkan lebih lanjut mensimulasikan pada platform machine berbeda dari *OS, IDE*, hingga *machine* komputer yang berbeda, sehingga konsep integrasi dengan SOA semakin tampak pada sistem ini.

## DAFTAR PUSTAKA

- Muller Benjamin, Viering Goetz, Legner Christine, Riempp Gerold. *Understanding the Economic Potential of Service-Oriented Architecture*, Journal of Management Information Systems / Spring 2010, Vol. 26, No. 4, pp. 145-180.
- Back Godmar, Bailey Annete. *Web Services and Widgets for Library Information Systems*, Information and Technology June 2010.
- Adam, John. *SOA's Now Par for The Course at Banks*, Bank Technology News (Forester Research Inc) January 2011.
- Hubert, Don. *SOA, Security, and Compliance*. A Supplement to National Mortgage News I (Published by SourceMedia's Custom Marketing Group)
- Bielski, Lauren. *SOA (Begins to Make) Inroads*, ABA Banking Journal-December 2006.
- Knorr, E. and O. Rist (2005). *10 steps to SOA*, InfoWorld, November 07, 2005, [http://infoworld.com/article/05/11/07/45FEsoaintro\\_1.html](http://infoworld.com/article/05/11/07/45FEsoaintro_1.html).
- Ren M., Lyytinen K. J. *Building Enterprise Architecture Agility and Sustenance with SOA*. The Communications of the Association for Information Systems 22, 4, 2008.

- Luftman, J. *Assessing business-IT alignment maturity*, Communications of the AIS (4:14) 2000, pp 1-50.
- Lammers, M. *Make, buy or share - Combining resource based view, transaction cost economics and production economies to a sourcing framework*, Wirtschaftsinformatik (46:3) 2004, pp 204-212.
- Homburg, Christian, Viviana V. Steiner, & Dirk Totzek, (2009). *Managing Dynamics in a Customer Portfolio*, Journal of Marketing, 73 (September), 70-89.
- Wang Chun, Ghenniwa Hamada, Shen Weiming. *Real time distributed shop floor scheduling using an agent-based service-oriented architecture*, International Journal of Production Research, Vol. 46, No. 9, 1 May 2008, 2433-2452.
- Lee Seok-Won, A. Gandhi Robin, J. Wagle Siddharth. *Ontology-Guided Service-Oriented Architecture Composition To Support*, International Journal of Software Engineering and Knowledge Engineering Vol. 19, No. 6 (2009) 791-821 c \_ World Scientific Publishing Company.
- Grzech Adam, Swiatek Pawel. *Modeling and Optimization of Complex Service in Service-Based Systems*, An International Journal, 40: 706-723(2009).
- J. Hannon Neal. *Leverage Business Reporting-Use Service-Oriented Architecture*, Strategic Finance November 2005.
- Sun Ping, Jiang Changjun, Zhou MengChu. *Interactive Web Service Composition Based on Petri net*,

- Transactions of the Institute of Measurement and Control 33, 1 (2011) pp. 116-132.
- Choi Jae, L, Nazareth Derek, K. Kain Hemant. *Implementing Service-Oriented Architecture in Organizations, Journal of Management Information Systems / Spring 2010, Vol. 26, No. 4, pp. 253-286.*
- Pakkala H, Christensen T, Martinez de Victoria I, Presser K, Kadvan A. Harmonised information exchange between decentralised food composition database systems, *European Journal of Clinical Nutrition* (2010) 64, S58-S63.
- Melrose J. Peter. *E-Health is The Way Via SOA, healthcare financial management* March 2007.
- Tejasvee Sanjay, Sarangdevot S.S. *E-Governance and Service Oriented Computing Architecture Model, CP1324, International Conference on Methods and Models in Science and Technology (ICM 2ST-10).*
- Wang Tsung-Li, Fang Lung-Ching, Wu Wen-Hsiung, Ho Chin-Fu. *Development of a Collaborative Product Development Framework Based on Centre-Satellite System and Service-Oriented Architecture, International Journal of Production Research* Vol. 47, No. 20, 15 October 2009, 5637-5656
- Robertson, Bruce. Sribar, Valentin. 2001. *The Adaptive Enterprise*. Intel Press, USA.dr

# SKPL

**SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK**  
**SISTEM INFORMASI PERBANKAN-INTEGRASI**  
**PELAPORAN KEUANGAN BANK**  
**(SIP-IPAB)**

Untuk :  
Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:  
Cosmas Dedy Kurniawan / 080705654

Program Studi Teknik Informatika  
Fakultas Teknologi Industri  
Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		<i>SKPL-SIP-IPAB</i>		1/22
		Revisi		

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	
<b>G</b>	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

### Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

## Daftar Isi

<b>1</b>	<b>Pendahuluan</b>	<b>170</b>
1.1	Tujuan	170
1.2	Lingkup Masalah	170
1.3	Definisi, Akronim dan Singkatan	170
<b>Server</b>		<b>171</b>
1.4	Referensi	171
1.5	Deskripsi umum (Overview)	171
<b>2</b>	<b>Deskripsi Kebutuhan</b>	<b>172</b>
2.1	Perspektif produk	172
2.2	Fungsi Produk	174
2.3	Karakteristik Pengguna	176
2.4	Batasan-batasan	176
2.5	Asumsi dan Ketergantungan	176
<b>3</b>	<b>Kebutuhan khusus</b>	<b>177</b>
3.1	Kebutuhan antarmuka eksternal	177
3.1.1	Antarmuka pemakai/pelanggan	177
3.1.2	Antarmuka perangkat keras	177
3.1.3	Antarmuka perangkat lunak	177
3.1.4	Antarmuka Komunikasi	179
3.2	Kebutuhan fungsionalitas Perangkat Lunak	179
3.2.1	Use Case Diagram	179
<b>4</b>	<b>Spesifikasi Rinci Kebutuhan</b>	<b>180</b>
4.1	Spesifikasi Kebutuhan Fungsionalitas	180
4.1.1	Use case Spesification : Login	41
4.1.2	Use case Spesification : Display Laporan Neraca Saldo	42
4.1.3	Use case Spesification : Display Laporan Lab-Rugi	43
4.1.4	Use case Spesification : Display Laporan Lab-Rugi	44
4.1.5	Use case Spesification : Display Laporan Komitmen dan Kontijensi	45
<b>5</b>	<b>Entity Relationship Diagram (ERD)</b>	<b>186</b>

## Daftar Gambar

Gambar 1. Arsitektur Perangkat Lunak SIP-IPAB	10
Gambar 2. Use Case Diagram	15
Gambar 3. Entity Relationship Diagram	22



## 1. Pendahuluan

### Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SIP-IPAB (Sistem Informasi Perbankan-Integrasi Pelaporan Keuangan Bank) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan pengguna), layanan-layanan yang diberikan dalam sistem, atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SIP-IPAB ini juga mendefinisikan batasan perancangan perangkat lunak.

### Lingkup Masalah

Perangkat Lunak SIP-IPAB dikembangkan dengan tujuan untuk :

3. Membantu pengguna dalam membuat laporan perbankan meliputi laporan neraca saldo, laporan laba-rugi, laporan kualitas aktiva, dan neraca bank pada bank cabang.
4. Membuat laporan keuangan gabungan dari bank-bank cabang dan pusat dengan platform berbeda (ASP .NET dan jsp).

### Definisi, Akronim dan Singkatan

Daftar definisi akronim dan Sigkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari

	perangkat lunak yang akan dikembangkan.
SKPL-SIP-IPAB-XXX	Kode yang merepresentasikan kebutuhan pada SIP-IPAB (Sistem Informasi Perbankan-Integrasi Pelaporan Keuangan Bank) dimana XXX merupakan nomor fungsi produk.
SIP-IPAB	Perangkat lunak yang digunakan untuk membuat laporan keuangan gabungan dari bank.
Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.

## Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

2. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented Sistem Analysis and Design USIPg UML*, McGraw-Hill Companies, 2002.
3. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.

## Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian pertama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak SIP-IPAB yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pelanggan, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SIP-IPAB tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SIP-IPAB yang akan dikembangkan.

### **Deskripsi Kebutuhan**

#### Perspektif produk

SIP-IPAB merupakan perangkat lunak berbasis web yang dikembangkan untuk membantu perusahaan, dalam kasus ini adalah bank untuk menampilkan laporan keuangan bulanan. Dengan sistem ini, diharapkan aliran pelaporan keuangan dari bank-bank cabang ke pusat dapat terintegrasi.

Sistem akan mengambil data dari transaksi bank yang masuk di neraca saldo untuk membuat laporan-laporan dasar perbankan. neraca saldo berisi transaksi harian, setiap transaksi mempunyai pos sendiri-sendiri, sehingga kemudian berdasarkan pos tersebut transaksi-transaksi itu akan diolah dan dimasukkan ke pos masing-masing. Adapun pos-pos tersebut adalah neraca saldo, laporan laba-rugi dan laporan komitmen dan kontijensi.

Proses pelaporan tersebut terjadi di tingkat bank cabang. Setiap bank cabang akan terjadi proses

yang sama. pada studi kasus, sistem informasi yang dibuat menggunakan platform IDE yang berbeda, yaitu menggunakan ASP .NET dan JDeveloper.

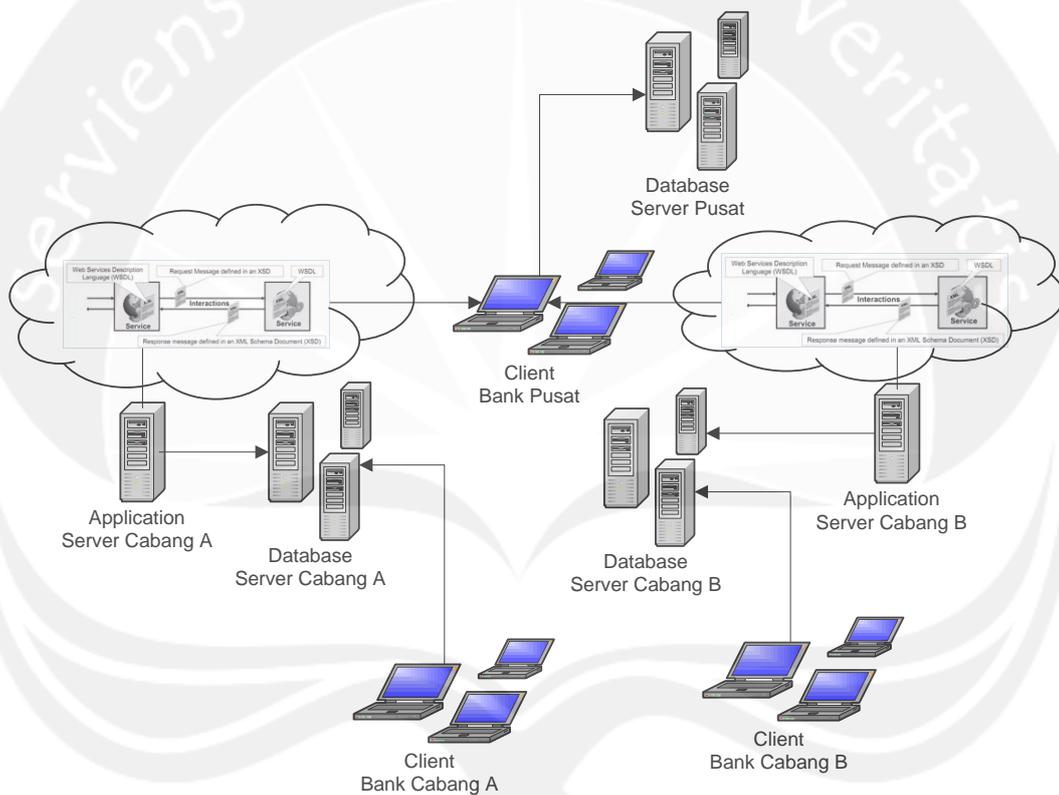
Untuk pembuatan laporan gabungan bank pusat, digunakan web service untuk mengakses data yang ada pada tiap bank cabang. Pada setiap sistem informasi di bank cabang, akan dibuat WSDL (*Web Service Definition Language*). Dengan menggunakan WSDL ini, walaupun platform yang digunakan berbeda, proses penggabungan laporan dapat dilakukan. Sistem informasi pusat dapat mengakses data dari setiap bank cabang dengan menggunakan WSDL.

penggabungan laporan menggunakan sistem ini akan mengambil data laporan neraca bank, laba-rugi, dan kualitas aktiva tiap bank cabang, yang kemudian digabungkan menjadi satu menjadi laporan keuangan bank.

Perangkat lunak SIP-IPAB ini berbasis web dan dalam pengembangannya menggunakan bahasa pemrograman C#, Java dan XML. Sedangkan, untuk lingkungan pemogramannya menggunakan Microsoft Visual Studio 2005 dan JDeveloper 11. Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*). Perangkat lunak yang digunakan adalah client-server di mana perangkat lunak dan database akan ditanam pada sistem informasi bank cabang, database cabang akan diakses oleh sistem informasi cabang.

Untuk seluruh data yang akan diolah disimpan di dalam database server. Sehingga jika ada pencarian data, maka data yang diinginkan akan dicari ke database server yang selanjutnya dikirimkan ke

client yang merequest melalui web server. Sedangkan untuk WSDL yang telah disediakan oleh sistem informasi bank cabang dapat diakses oleh bank pusat melalui jaringan secara on-line. Permintaan yang datang dari sistem informasi bank pusat akan dieksekusi di web server untuk kemudian dihasilkan response.



Gambar 3 Arsitektur Perangkat Lunak SIP-IPAB

## Fungsi Produk

Fungsi produk perangkat lunak SIP-IPAB adalah sebagai berikut :

1. **Fungsi Login (SKPL-SIP-IPAB-001).**

Merupakan fungsi yang digunakan oleh pengguna untuk dapat masuk dalam sistem yang akan digunakan sesuai dengan role yang dimiliki masing-masing pengguna. Fungsi ini mencakup :

**a. Fungsi *Input Data Login* (SKPL-SIP-IPAB-001-01)**

Merupakan fungsi yang digunakan user untuk menginputkan username serta password.

**b. Fungsi *Validasi Data Login* (SKPL-SIP-IPAB-001-02)**

Merupakan fungsi dimana sistem melakukan proses pengecekan apakah data login yang diinputkan valid atau tidak.

**2. Fungsi *Display Laporan Neraca Saldo* (SKPL-SIP-IPAB-002) .**

Merupakan fungsi yang digunakan untuk melihat laporan Neraca Saldo pada setiap bank cabang dan pusat

**3. Fungsi *Display Laporan Neraca Bank* (SKPL-SIP-IPAB-003) .**

Merupakan fungsi yang digunakan untuk melihat laporan Neraca Saldo pada setiap bank cabang dan pusat

**4. Fungsi *Display Laporan Laba-Rugi* (SKPL-SIP-IPAB-004) .**

Merupakan fungsi yang digunakan untuk membuat laporan Laba-Rugi dari setiap bank cabang dan pusat.

**5. Fungsi *Display Laporan Komitmen dan Kontijensi* (SKPL-SIP-IPAB-005) .**

Merupakan fungsi yang digunakan untuk membuat laporan Komitmen dan Kontijensi dari setiap bank cabang dan pusat.

## Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SIP-IPAB adalah sebagai berikut :

1. Memahami pengoperasian Personal Computer (PC).
2. Memahami pengoperasian fungsi-fungsi yang terdapat dalam SIP-IPAB.

## Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SIP-IPAB tersebut adalah :

1. Kebijakan Umum  
Berpedoman pada tujuan dari pengembangan perangkat lunak SIP-IPAB.
2. Keterbatasan perangkat keras  
Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

## Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada aplikasi web browser dan hanya bisa dijalankan pada PC dengan sistem operasi Windows XP, Windows Vista atau Windows 7.

## **Kebutuhan khusus**

### **Kebutuhan antarmuka eksternal**

Kebutuhan antar muka eksternal pada perangkat lunak SIP-IPAB meliputi kebutuhan antarmuka pemakai/pelanggan, antarmuka perangkat keras, antarmuka perangkat lunak, dan antarmuka komunikasi.

#### **Antarmuka pemakai/pelanggan**

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form-form. Pada proses pembuatan laporan, antarmuka utama ditampilkan dalam bentuk web dengan penampilan tabel

#### **Antarmuka perangkat keras**

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SIP-IPAB adalah:

1. Perangkat PC (sistem operasi Windows XP, Windows Vista, atau Windows 7).
2. CPU
3. Monitor
4. Keyboard
5. Mouse
6. Printer

#### **Antarmuka perangkat lunak**

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SIP-IPAB adalah sebagai berikut :

a. Perangkat Lunak di Bank Cabang B

5. Nama : Oracle Database 11g  
Sumber : Oracle  
Sebagai database management sistem (DBMS) yang digunakan untuk penyimpan data di sisi server.
6. Nama : Oracle JDeveloper 11g  
Sumber : Oracle  
Sebagai framework aplikasi.
7. Nama : Oracle Weblogic Server 11g  
Sumber : Oracle  
Sebagai web server.
8. Nama : Windows XP, Windows Vista, atau Windows 7  
Sumber : Microsoft  
Sebagai sistem operasi.

b. Perangkat Lunak Bank Cabang A dan Bank Pusat

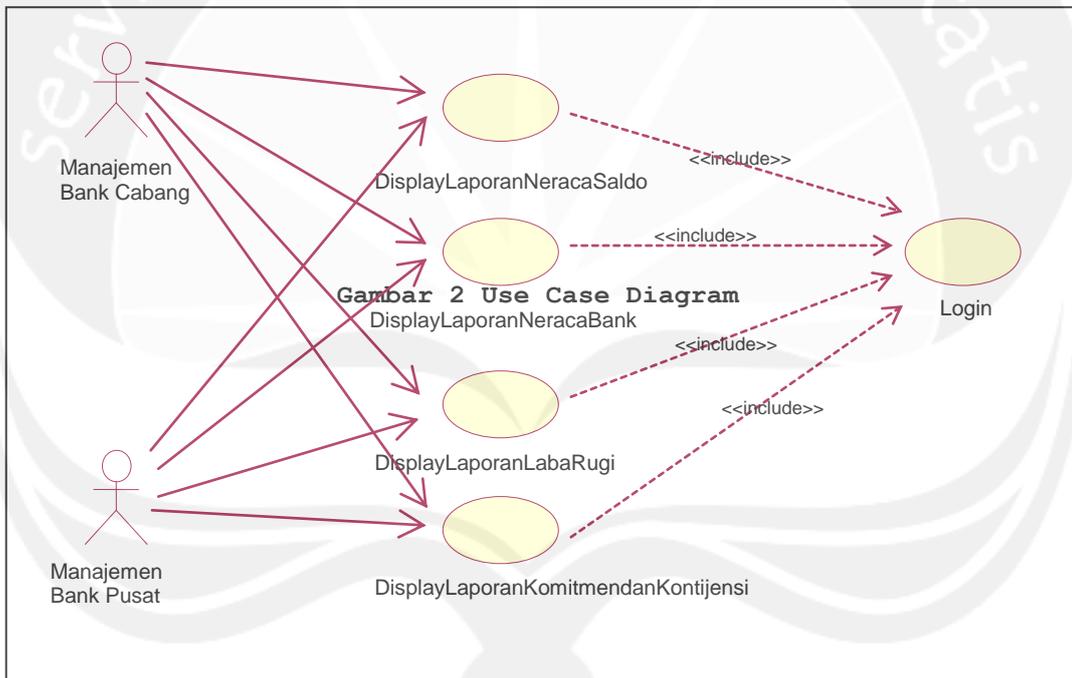
5. Nama : SQL Server Expres 2005  
Sumber : Microsoft  
Sebagai database management sistem (DBMS) yang digunakan untuk penyimpan data di sisi server.
6. Nama : .NET Framework  
Sumber : Microsoft  
Sebagai framework aplikasi.
7. Nama : IIS (Internet Information Server)  
Sumber : Microsoft  
Sebagai web server.
8. Nama : Windows XP, Windows Vista, atau Windows 7  
Sumber : Microsoft  
Sebagai sistem operasi.

## Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SIP-IPAB menggunakan protokol HTTP. Sedangkan untuk antarmuka komunikasi web service digunakan Web Service Description Language (WSDL).

## Kebutuhan fungsionalitas Perangkat Lunak

### Use Case Diagram



## **Spesifikasi Rinci Kebutuhan**

### Spesifikasi Kebutuhan Fungsionalitas

#### **Use case Spesification : Login**

##### **9. Brief Description**

Use Case ini digunakan oleh aktor untuk memperoleh akses ke sistem. Login didasarkan pada sebuah kombinasi unik yaitu username dari user dan Password yang berupa rangkaian karakter.

##### **10. Primary Actor**

1. Manajemen Bank Cabang
2. Manajemen Bank Pusat

##### **11. Supporting Actor**

none

##### **12. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melakukan login
2. Sistem menampilkan antarmuka untuk login
3. Aktor memasukkan username dan Password
4. Sistem memeriksa username dan password yang diinputkan aktor  
E-1 Password atau username user tidak sesuai
5. Sistem memberikan akses ke aktor
6. Use Case ini selesai

##### **13. Alternative Flow**

none

##### **14. Error Flow**

- E-1 Password atau username user tidak sesuai
1. Sistem menampilkan peringatan bahwa username user atau password tidak sesuai
  2. Kembali ke Basic Flow langkah ke 3

##### **15. PreConditions**

None

**16. PostConditions**

1. Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem.

**Use case Spesification :** Display Laporan Neraca Saldo

**7. Brief Description**

Use case ini digunakan oleh aktor untuk melihat laporan keuangan bank cabang berupa neraca saldo dari transaksi yang masuk ke basis data bank cabang atau pusat.

**8. Primary Aktor**

1. Manajemen Bank Cabang
2. Manajemen Bank Pusat

**9. Supporting Aktor**

none

**10. Basic Flow**

7. Use Case ini dimulai ketika aktor memilih untuk melihat laporan keuangan bank cabang atau Pusat
8. Sistem menampilkan menu utama dalam bentuk web
9. Aktor melihat data tabel laporan Neraca Saldo bank cabang
10. Sistem menampilkan data yang ada pada basis data sesuai permintaan aktor
11. Setiap beberapa waktu berkala sistem melakukan update basis data
12. Use case ini selesai

**11. Alternative Flow**

none

**12. Errow Flow**

None

**13. PreConditions**

3. Use Case Login telah dilakukan

4. Aktor telah memasuki sistem

**8. PostConditions**

2. Aktor melihat data laporan keuangan neraca saldo bank cabang atau pusat.

**Use case Spesification :** Display Laporan Lab-Rugi

**1. Brief Description**

Use case ini digunakan oleh aktor untuk melihat laporan neraca bank dari bank cabang atau pusat.

**2. Primary Aktor**

1. Manajemen Bank Cabang

1. Manajemen Bank Pusat

**3. Supporting Aktor**

none

**4. Basic Flow**

13. Use Case ini dimulai ketika aktor memilih untuk melihat laporan neraca bank cabang atau pusat

14. Sistem menampilkan menu utama dalam bentuk web

15. Aktor melihat data tabel laporan neraca bank cabang atau pusat

16. Sistem menampilkan data yang ada pada basis data sesuai permintaan aktor

17. Setiap beberapa waktu berkala sistem melakukan update basis data

18. Use case ini selesai

**5. Alternative Flow**

none

**6. Errow Flow**

None

**7. PreConditions**

5. Use Case Login telah dilakukan

6. Aktor telah memasuki sistem

**8. PostConditions**

3. Aktor melihat data laporan neraca bank

**Use case Spesification** : Display Laporan Lab-Rugi

**2. Brief Description**

Use case ini digunakan oleh aktor untuk melihat laporan laba-rugi dari bank cabang atau pusat.

**2. Primary Aktor**

1. Manajemen Bank Cabang

1. Manajemen Bank Pusat

**3. Supporting Aktor**

none

**4. Basic Flow**

19. Use Case ini dimulai ketika aktor memilih untuk melihat laporan laba-rugi bank cabang atau pusat

20. Sistem menampilkan menu utama dalam bentuk web

21. Aktor melihat data tabel laporan laba-rugi bank cabang atau pusat

22. Sistem menampilkan data yang ada pada basis data sesuai permintaan aktor

23. Setiap beberapa waktu berkala sistem melakukan update basis data

24. Use case ini selesai

**5. Alternative Flow**

none

**6. Errow Flow**

None

**7. PreConditions**

7. Use Case Login telah dilakukan

8. Aktor telah memasuki sistem

**8. PostConditions**

4. Aktor melihat data laporan laba-rugi

**Use case Spesification** : Display Laporan Komitmen dan Kontijensi

**2. Brief Description**

Use case ini digunakan oleh aktor untuk melihat hasil laporan komitmen dan kontijensi. Aktor dapat memilih hasil cetak berdasarkan *range* waktu tertentu.

**3. Primary Aktor**

1. Manajemen Bank Cabang

2. Manajemen Bank Pusat

**4. Supporting Aktor**

none

**5. Basic Flow**

1. Use Case ini dimulai ketika aktor memilih untuk melihat hasil laporan komitmen dan kontijensi bank

7. Sistem menampilkan menu utama dalam bentuk web

8. Aktor melihat data tabel laporan komitmen dan kontijensi bank cabang atau pusat
9. Sistem menampilkan data yang ada pada basis data sesuai permintaan aktor
10. Setiap beberapa waktu berkala sistem melakukan update basis data
11. Use case ini selesai

**6. Alternative Flow**

none

**7. Error Flow**

none

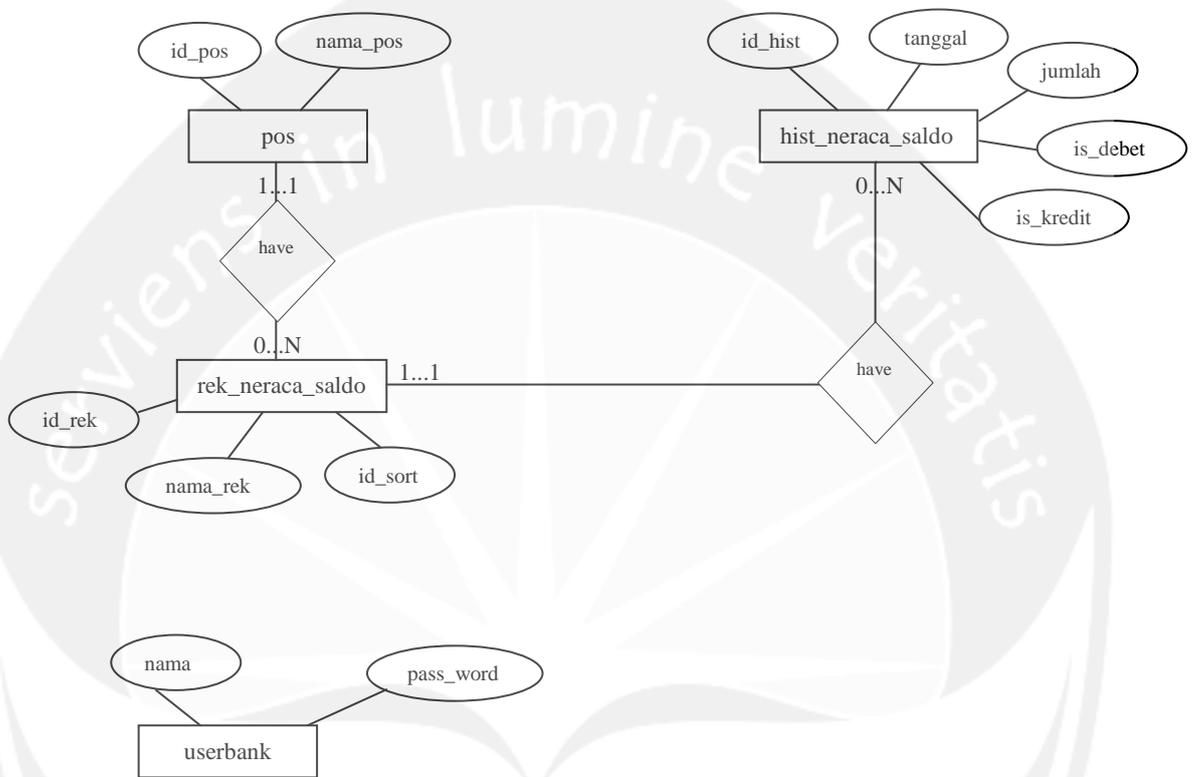
**7. PreConditions**

1. Use Case Login telah dilakukan
2. Aktor telah memasuki sistem

**8. PostConditions**

1. Aktor melihat data laporan Komitmen dan Kontijensi

### Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram

# DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SISTEM INFORMASI PERBANKAN-INTEGRASI  
PELAPORAN KEUANGAN BANK  
(SIP-IPAB)

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Cosmas Dedy Kurniawan / 08 07 05654

Program Studi Teknik Informatika - Fakultas  
Teknologi Industri  
Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		<i>DPPL-SIP-IPAB</i>		1/59
		Revisi		

## DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	



## Daftar Isi

<b>1. Pendahuluan</b> .....	<b>194</b>
1.1. Tujuan .....	194
1.2. Lingkup Masalah.....	194
1.3. Definisi, Akronim dan Singkatan.....	194
<b>Server</b> .....	<b>195</b>
1.4. Referensi .....	195
<b>1 Perancangan Sistem</b> .....	<b>196</b>
1.1 Perancangan Arsitektur.....	196
1.2 Perancangan Rinci.....	197
1.2.1 Sequence Diagram.....	197
2.2.1.1 Display Neraca Saldo.....	197
2.2.1.2 Display Neraca Bank.....	198
2.2.1.3 Display Laporan Laba-Rugi.....	199
2.2.1.4 Display Laporan Komitmen dan Kontijensi .....	200
1.2.2 Class Diagram.....	201
1.2.3 Spesifikasi Deskripsi Kelas Diagram.....	202
2.2.3.1 Specific Design Class LoginUI_A .....	202
2.2.3.2 Specific Design Class DisplayLaporanNeracaSaldoUI_A.....	202
2.2.3.3 Specific Design Class DisplayLaporanNeracaBankUI_A .....	202
2.2.3.4 Specific Design Class DisplayLaporanLabaRugiUI_A.....	202
2.2.3.5 Specific Design Class DisplayLaporanKomitmendankontijensiUI_A.....	203
2.2.3.6 Specific Design Class LoginUI_Pusat.....	203
2.2.3.7 Specific Design Class DisplayLaporanNeracaSaldoUI_Pusat .....	203
2.2.3.8 Specific Design Class DisplayLaporanNeracaBankUI_Pusat.....	203
2.2.3.9 Specific Design Class DisplayLaporanLabaRugiUI_Pusat.....	204
2.2.3.10 Specific Design Class DisplayLaporanKomitmendankontijensiUI_Pusat.....	204
2.2.3.11 Specific Design Class LoginUI_B .....	204
2.2.3.12 Specific Design Class DisplayLaporanNeracaSaldoUI_B.....	204
2.2.3.13 Specific Design Class DisplayLaporanNeracaBankUI_B .....	205
2.2.3.14 Specific Design Class DisplayLaporanLabaRugiUI_B.....	205

2.2.3.15	Specific Design Class	
	DisplayLaporanKomitmendankontijensiUI_B.....	205
2.2.3.16	Specific Design Class LoginControl_A.....	205
2.2.3.17	Specific Design Class NeracaSaldoControl_A.....	206
2.2.3.18	Specific Design Class NeracaBankControl_A.....	206
2.2.3.19	Specific Design Class LabaRugiControl_A.....	207
2.2.3.20	Specific Design Class Laporan	
	KomitmendankontijensiControl_A.....	209
2.2.3.21	Specific Design Class NET_wsdlControl.....	210
2.2.3.22	Specific Design Class LoginControl_Pusat.....	214
2.2.3.23	Specific Design Class NeracaSaldoControl_Pusat.....	215
2.2.3.24	Specific Design Class NeracaBankControl_Pusat.....	215
2.2.3.25	Specific Design Class LabaRugiControl.....	216
2.2.3.26	Specific Design Class Laporan	
	KomitmendankontijensiControl_Pusat.....	218
2.2.3.27	Specific Design Class NET_webReferencesControl.....	219
2.2.3.28	Specific Design Class	
	JDev_webReferencesControl.....	224
2.2.3.29	Specific Design Class LoginControl_B.....	228
2.2.3.30	Specific Design Class NeracaSaldoControl_B.....	228
2.2.3.31	Specific Design Class NeracaBankControl_B.....	229
2.2.3.32	Specific Design Class LabaRugiControl_B.....	230
2.2.3.33	Specific Design Class Laporan	
	KomitmendankontijensiControl_B.....	231
2.2.3.34	Specific Design Class JDev_wsdlControl.....	233
2.2.3.35	Specific Design Class userBank_A.....	237
2.2.3.36	Specific Design Class rekNeracaSaldo_A.....	237
2.2.3.37	Specific Design Class histNeracaSaldo_A.....	238
2.2.3.38	Specific Design Class pos_A.....	238
2.2.3.39	Specific Design Class userBank_Pusat.....	238
2.2.3.40	Specific Design Class rekNeracaSaldo_Pusat.....	238
2.2.3.41	Specific Design Class histNeracaSaldo_Pusat.....	239
2.2.3.42	Specific Design Class pos_Pusat.....	239
2.2.3.43	Specific Design Class userBank_B.....	239
2.2.3.44	Specific Design Class rekNeracaSaldo_B.....	239
2.2.3.45	Specific Design Class histNeracaSaldo_B.....	240
2.2.3.46	Specific Design Class pos_B.....	240
<b>2</b>	<b>Deskripsi Dekomposisi .....</b>	<b>240</b>
2.1	Dekomposisi Data.....	240
2.1.1	Deskripsi Entitas Data userbank.....	240
2.1.2	Deskripsi Entitas Data rek_neraca_saldo.....	240
2.1.3	Deskripsi Entitas Data hist_neraca_saldo.....	241
2.1.4	Deskripsi Entitas Data pos.....	241
2.2	Physical Data Model.....	241
<b>3</b>	<b>Deskripsi Perancangan AntarMuka.....</b>	<b>241</b>
3.1	Login.....	241
Us	.....	241
3.2	Halaman Utama.....	242

3.3	Halaman Display Neraca Saldo.....	243
3.4	Halaman Display Neraca Bank.....	243
3.5	Halaman Display Laporan Laba-Rugi.....	244
3.6	Halaman Display Laporan Komitmen dan Kontijensi.....	245



## Daftar Gambar

Gambar 1. Rancangan Arsitektur SIP-IPAB .....	10
Gambar 2. Sequence Diagram : Display Neraca Saldo.....	11
Gambar 3. Sequence Diagram : Display Neraca Bank.....	12
Gambar 4. Sequence Diagram : Display Laporan Laba-Rugi.....	13
Gambar 5. Sequence Diagram : Display Laporan Komitmen dan Kontijensi.....	14
Gambar 6. Class Diagram .....	15
Gambar 7. Physical Data Model.....	56
Gambar 8. Rancangan Antarmuka Login.....	56
Gambar 9. Rancangan Antarmuka Halaman Utama.....	57
Gambar 10. Rancangan Antarmuka Halaman Display Neraca Saldo...	57
Gambar 11. Rancangan Antarmuka Halaman Display Neraca Bank....	58
Gambar 12. Rancangan Antarmuka Halaman Display Laporan Laba- Rugi.....	58
Gambar 13. Rancangan Antarmuka Halaman Display Komitmen dan Kontijensi.....	59

59

## 2. Pendahuluan

### 1.1. Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

### 1.2. Lingkup Masalah

Perangkat Lunak SIP-IPAB dikembangkan dengan tujuan untuk :

5. Membantu pengguna dalam membuat laporan perbankan meliputi laporan neraca saldo, laporan laba-rugi, laporan komitmen dan kontijensi, dan neraca bank pada bank cabang.
6. Membuat laporan keuangan gabungan dari bank-bank cabang dan pusat dengan platform berbeda (ASP .NET dan jspx).

### 1.3. Definisi, Akronim dan Singkatan

Daftar definisi akronim dan Sigkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
SIP-IPAB	Perangkat lunak yang digunakan untuk membuat laporan keuangan gabungan dari bank.

Server	Komputer yang menyediakan sumber daya bagi klien yang terhubung melalui jaringan.
--------	---

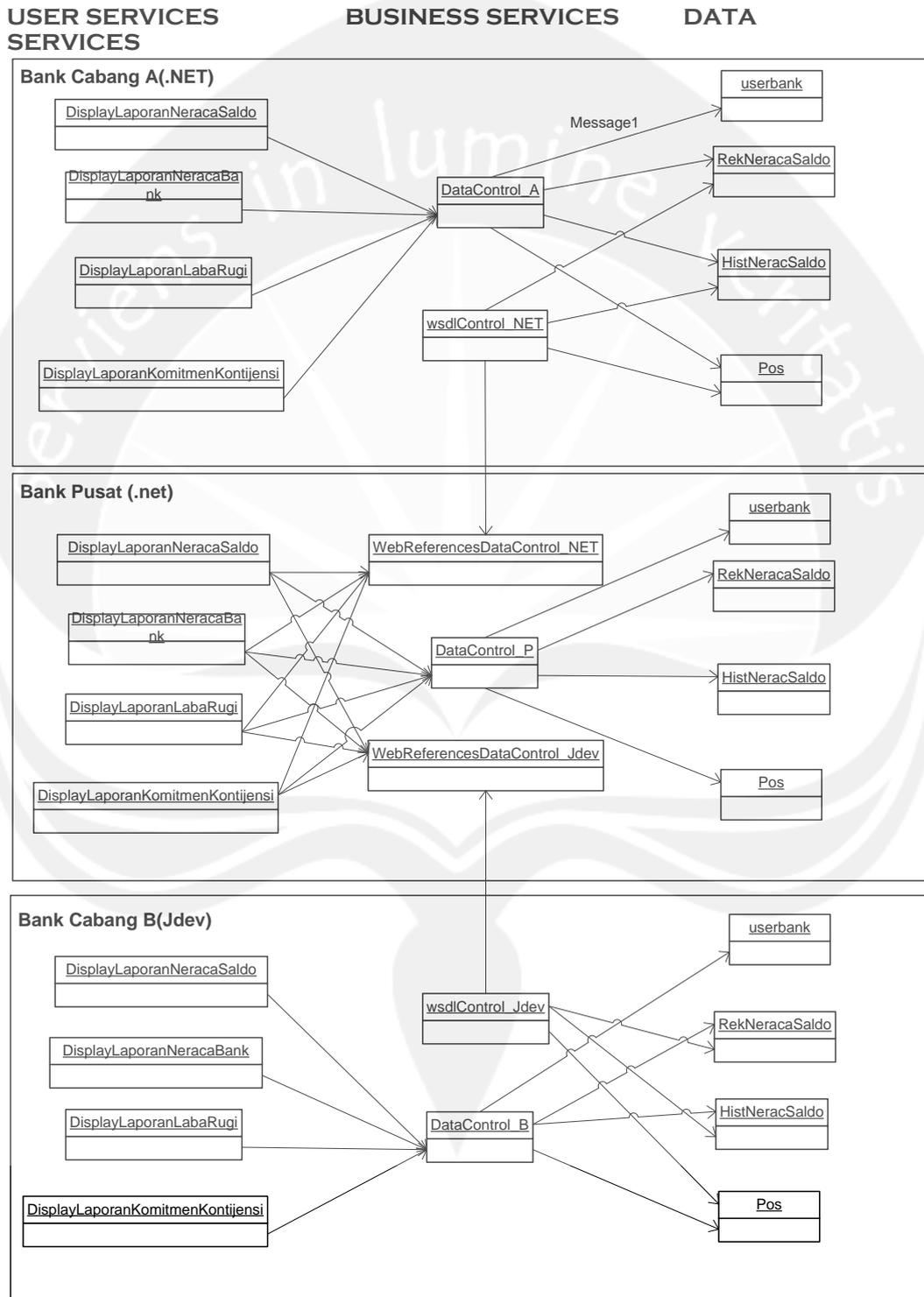
#### 1.4. Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

4. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented Sistem Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
5. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.

## Perancangan Sistem

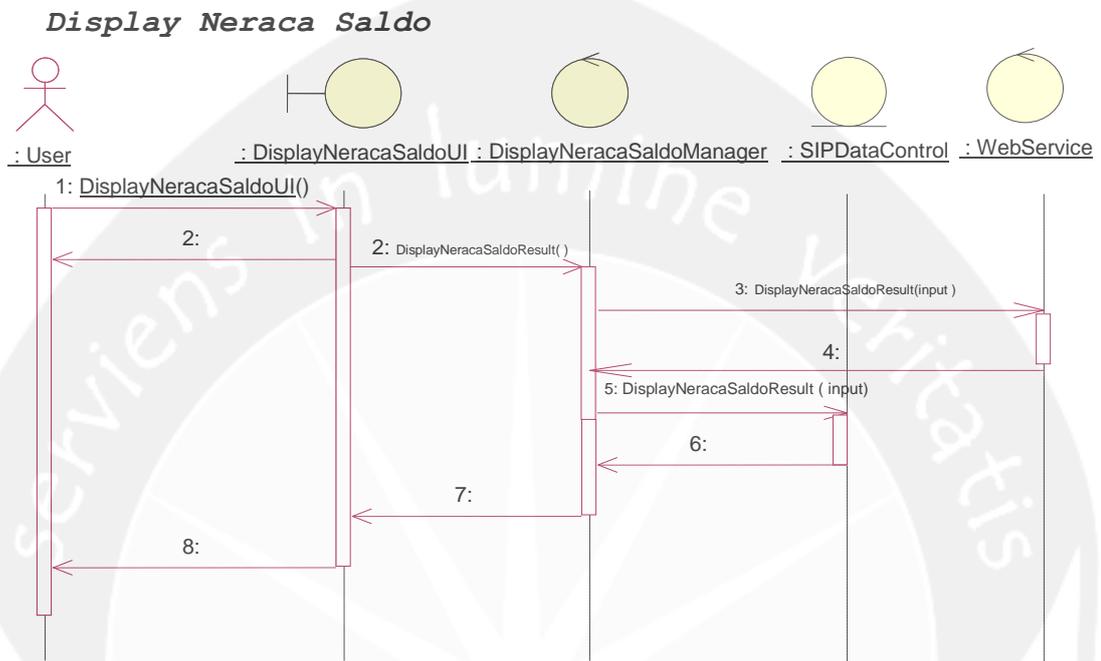
### Perancangan Arsitektur



Gambar 1 Rancangan Arsitektur SIP-IPAB

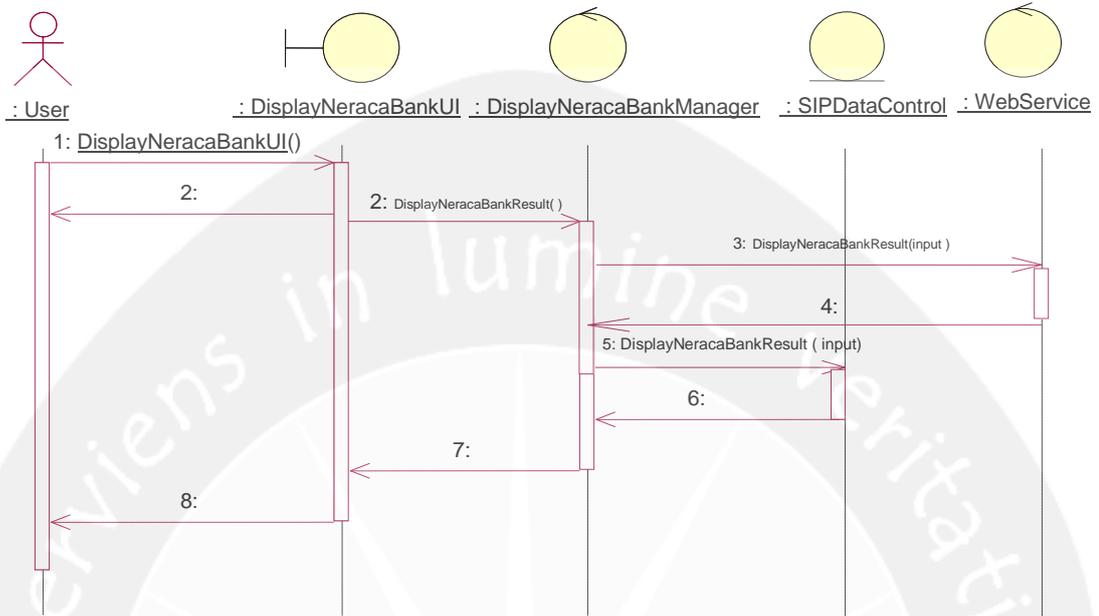
## Perancangan Rinci

### Sequence Diagram



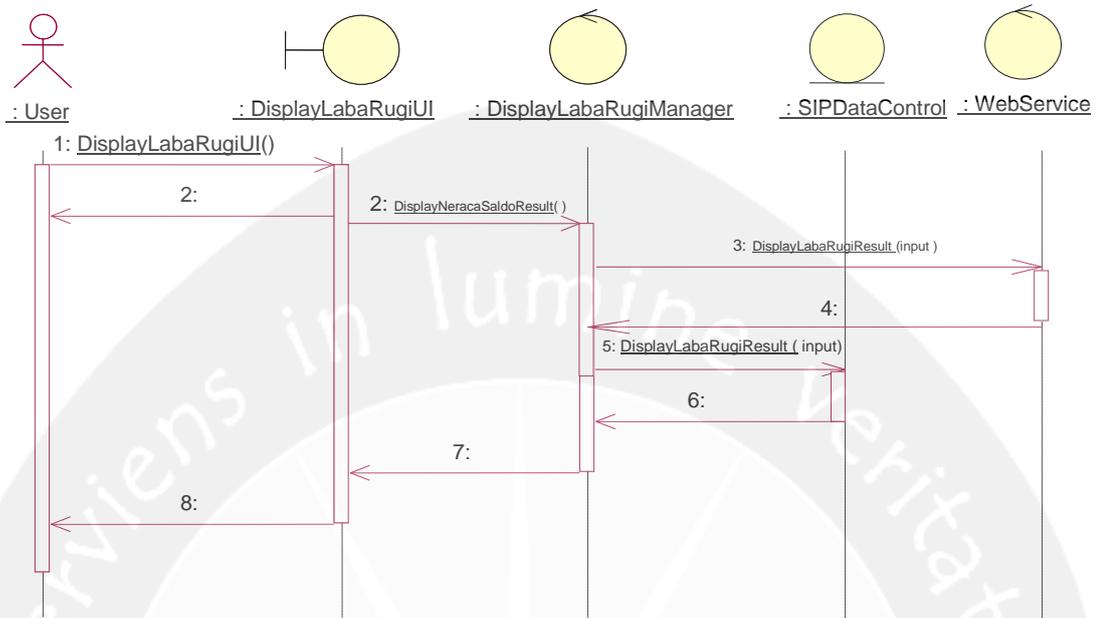
Gambar 2 Sequence Diagram : Display Neraca Saldo

### Display Neraca Bank



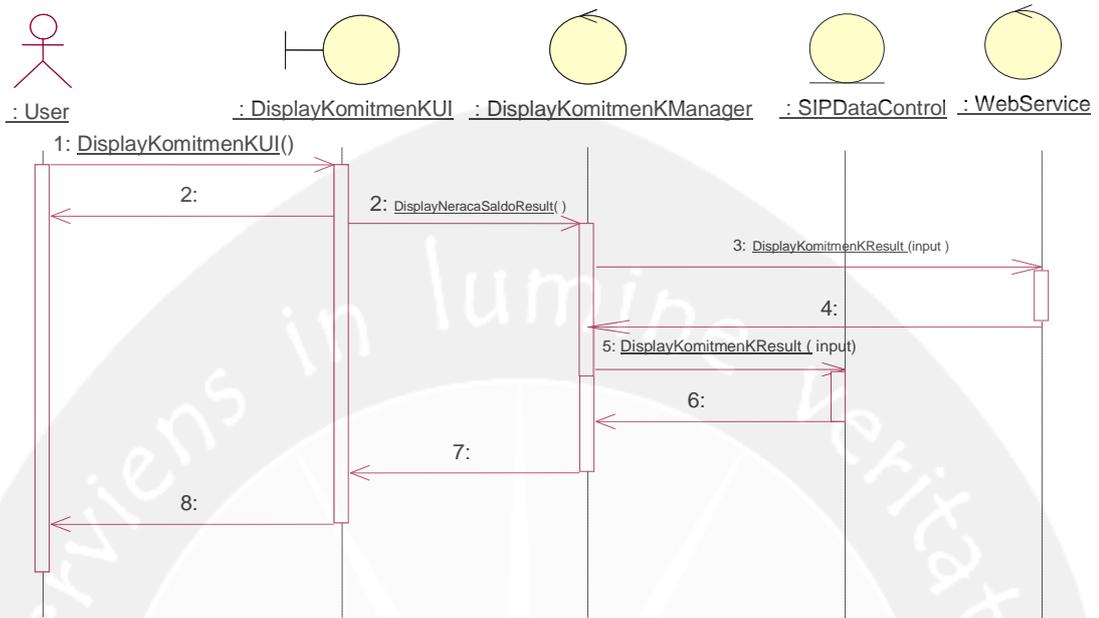
Gambar 3 Sequence Diagram : Display Neraca Bank

### Display Laporan Laba-Rugi



Gambar 4 Sequence Diagram : Display Laporan Laba-Rugi

### Display Laporan Komitmen dan Kontijensi



Gambar 5 Sequence Diagram : Display Laporan Komitmen dan Kontijensi



## Spesifikasi Deskripsi Kelas Diagram

### *Specific Design Class LoginUI\_A*

LoginUI	<<boundary>>
+loginUI() Merupakan default konstruktor, digunakan untuk inisialisasi semua atribut kelas ini.	
+inputLogin(string username, string password) Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu username dan password.	

### *Specific Design Class DisplayLaporanNeracaSaldoUI\_A*

DisplayLaporanNeracaSaldoUI	<<boundary>>
+ DisplayLaporanNeracaSaldo(string bulan, string tahun) Operasi ini digunakan untuk menampilkan laporan Neraca Saldo berdasarkan bulan dan tahun	

### *Specific Design Class DisplayLaporanNeracaBankUI\_A*

DisplayLaporanNeracaBankUI	<<boundary>>
+ DisplayLaporanNeracaBank(string bulan, string tahun) Operasi ini digunakan untuk menampilkan laporan Neraca Bank berdasarkan bulan dan tahun	

### *Specific Design Class DisplayLaporanLabaRugiUI\_A*

DisplayLaporanLabaRugiUI	<<boundary>>
+ DisplayLaporanLabaRugi(string bulan, string tahun) Operasi ini digunakan untuk menampilkan laporan Laba-Rugi berdasarkan bulan dan tahun	

**Specific Design Class**

**DisplayLaporanKomitmendankontijensiUI\_A**

DisplayLaporanKomitmendankontijensiUI	<<boundary>>
+ DisplayLaporanKomitmendankontijensi (string bulan, string tahun) Operasi ini digunakan untuk menampilkan laporan Komitmen dan Kontijensi berdasarkan bulan dan tahun	

**Specific Design Class LoginUI\_Pusat**

LoginUI	<<boundary>>
+loginUI() Merupakan default konstruktor, digunakan untuk inisialisasi semua atribut kelas ini. +inputLogin(string username, string password) Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu username dan password.	

**Specific Design Class DisplayLaporanNeracaSaldoUI\_Pusat**

DisplayLaporanNeracaSaldoUI	<<boundary>>
+ DisplayLaporanNeracaSaldo (string bulan, string tahun) Operasi ini digunakan untuk menampilkan laporan Neraca Saldo berdasarkan bulan dan tahun	

**Specific Design Class DisplayLaporanNeracaBankUI\_Pusat**

DisplayLaporanNeracaBankUI	<<boundary>>
+ DisplayLaporanNeracaBank (string bulan, string tahun) Operasi ini digunakan untuk menampilkan laporan Neraca Bank berdasarkan bulan dan tahun	

**Specific Design Class DisplayLaporanLabaRugiUI\_Pusat**

DisplayLaporanLabaRugiUI	<<boundary>>
+ DisplayLaporanLabaRugi(string bulan,string tahun) Operasi ini digunakan untuk menampilkan laporan Laba-Rugi berdasarkan bulan dan tahun	

**Specific Design Class**

**DisplayLaporanKomitmendanKontijensiUI\_Pusat**

DisplayLaporanKomitmendanKontijensiUI	<<boundary>>
+ DisplayLaporanKomitmendanKontijensi(string bulan,string tahun) Operasi ini digunakan untuk menampilkan laporan Komitmen dan Kontijensi berdasarkan bulan dan tahun	

**Specific Design Class LoginUI\_B**

LoginUI	<<boundary>>
+loginUI() Merupakan default konstruktor, digunakan untuk inisialisasi semua atribut kelas ini. +inputLogin(string username, string password) Operasi ini digunakan untuk mengambil data login yang diinputkan oleh user, yaitu username dan password.	

**Specific Design Class DisplayLaporanNeracaSaldoUI\_B**

DisplayLaporanNeracaSaldoUI	<<boundary>>
+ DisplayLaporanNeracaSaldo(string bulan,string tahun) Operasi ini digunakan untuk menampilkan laporan Neraca Saldo berdasarkan bulan dan tahun	

**Specific Design Class DisplayLaporanNeracaBankUI\_B**

DisplayLaporanNeracaBankUI	<<boundary>>
+ DisplayLaporanNeracaBank(string bulan,string tahun) Operasi ini digunakan untuk menampilkan laporan Neraca Bank berdasarkan bulan dan tahun	

**Specific Design Class DisplayLaporanLabaRugiUI\_B**

DisplayLaporanLabaRugiUI	<<boundary>>
+ DisplayLaporanLabaRugi(string bulan,string tahun) Operasi ini digunakan untuk menampilkan laporan Laba-Rugi berdasarkan bulan dan tahun	

**Specific Design Class**

**DisplayLaporanKomitmendankontijensiUI\_B**

DisplayLaporanKomitmendankontijensiUI	<<boundary>>
+ DisplayLaporanKomitmendankontijensi(string bulan,string tahun) Operasi ini digunakan untuk menampilkan laporan Komitmen dan Kontijensi berdasarkan bulan dan tahun	

**Specific Design Class LoginControl\_A**

LoginControl	<<Control>>
+LoginControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +validasi() Operasi ini digunakan untuk membandingkan account user yang dimasukkan dengan account user yang terdaftar dalam database.	

**Specific Design Class NeracaSaldoControl\_A**

<b>NeracaSaldoControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<p>+get_itemNS1(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itemNS2(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_totalNS1(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan total jumlah debit dan kredit rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_totalNS2(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan total jumlah debit dan kredit rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p>	

**Specific Design Class NeracaBankControl\_A**

<b>NeracaBankControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<p>+get_itemaktiva (string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening aktiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itempasiva(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data</p>	

rekening pasiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalaktiva(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah aktiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalpasiva(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah aktiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

### ***Specific Design Class LabaRugiControl\_A***

<b>LabarugiControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<p>+get_itemJPB(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening pendapatan bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itemJBB(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itemJPO(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening pendapatan pendapatan operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itemJBO(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening beban operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itemPBNO(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening pendapatan beban operasional yang ada pada basis data</p>	

sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalJPB(string bulan,string tahun)  
Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalJBB(string bulan,string tahun)  
Operasi ini digunakan untuk mendapatkan total jumlah beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalPBB(string bulan,string tahun)  
Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalJPO(string bulan,string tahun)  
Operasi ini digunakan untuk mendapatkan total jumlah pendapatan operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalJBO(string bulan,string tahun)  
Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalLBO(string bulan,string tahun)  
Operasi ini digunakan untuk mendapatkan total jumlah laba-rugi operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalPBNO  
Operasi ini digunakan untuk mendapatkan total jumlah pendapatan dan beban non operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalLBSPD(string bulan,string tahun)  
Operasi ini digunakan untuk mendapatkan total jumlah laba-rugi sebelum pajak penghasilan yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

**KomitmendankontijensiControl\_A**

KomitmendankontijensiControl	<<Control>>
<pre> +get_itemJTK Operasi ini digunakan untuk mendapatkan data rekening tagihan komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_itemJKK Operasi ini digunakan untuk mendapatkan data rekening kewajiban komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_itemJTKj Operasi ini digunakan untuk mendapatkan data rekening tagihan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_itemJKKj Operasi ini digunakan untuk mendapatkan data rekening kewajiban kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_totalJTK Operasi ini digunakan untuk mendapatkan total jumlah tagihan komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_totalJKK Operasi ini digunakan untuk mendapatkan total jumlah kewajiban komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_totalJKB Operasi ini digunakan untuk mendapatkan total jumlah komitmen bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_totalJTKj </pre>	

Operasi ini digunakan untuk mendapatkan total jumlah tagihan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJKKj

Operasi ini digunakan untuk mendapatkan total jumlah kewajiban kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJKjB

Operasi ini digunakan untuk mendapatkan total jumlah kontijensi bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

***Specific Design Class NET\_wsdlControl***

<b>webReferencesControl</b>	<b>&lt;&lt;control&gt;&gt;</b>
<p>+wsdl_itemNS1(string bulan,string tahun)            Operasi ini digunakan untuk memberikan <i>web service</i> akses data rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+wsdl_itemNS2(string bulan,string tahun)            Operasi ini digunakan untuk memberikan <i>web service</i> akses data rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+wsdl_totalNS1(string bulan,string tahun)            Operasi ini digunakan untuk memberikan <i>web service</i> akses total jumlah debet dan kredit rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+wsdl_totalNS2(string bulan,string tahun)            Operasi ini digunakan untuk memberikan <i>web service</i> akses total jumlah debet dan kredit rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data</p>	

bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemaktiva (string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening aktiva yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itempasiva(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening pasiva yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalaktiva(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah aktiva yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalpasiva(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah aktiva yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJPB(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening pendapatan bunga yang ada pada basis data bank cabang sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJBB(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening beban bunga yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJPO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening pendapatan pendapatan operasional yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJBO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening beban operasional yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemPBNO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening pendapatan beban operasional yang ada pada basis data cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJPB(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah pendapatan bunga yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalPBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJPO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah pendapatan operasional yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJBO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah pendapatan bunga yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalLBO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses

total jumlah laba-rugi operasional yang ada pada basis data banak cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalPBNO

Operasi ini digunakan untuk memberikan *web service* akses total jumlah pendapatan dan beban non operasional yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalLBSPD(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah laba-rugi sebelum pajak penghasilan yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJTK

Operasi ini digunakan untuk memberikan *web service* akses data rekening tagihan komitmen yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJKK

Operasi ini digunakan untuk memberikan *web service* akses data rekening kewajiban komitmen yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJTKj

Operasi ini digunakan untuk memberikan *web service* akses data rekening tagihan kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJKKj

Operasi ini digunakan untuk memberikan *web service* akses data rekening kewajiban kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJTK

Operasi ini digunakan untuk memberikan *web service* akses

total jumlah tagihan komitmen yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKK

Operasi ini digunakan untuk memberikan *web service* akses total jumlah kewajiban komitmen yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKB

Operasi ini digunakan untuk memberikan *web service* akses total jumlah komitmen bersih yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJTKj

Operasi ini digunakan untuk memberikan *web service* akses total jumlah tagihan kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKKj

Operasi ini digunakan untuk memberikan *web service* akses total jumlah kewajiban kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKjB

Operasi ini digunakan untuk memberikan *web service* akses total jumlah kontijensi bersih yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

***Specific Design Class LoginControl\_Pusat***

<b>LoginControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
+LoginControl()	
Default konstruktor, digunakan untuk inisialisasi	

semua atribut dari kelas ini.  
+validasi()  
Operasi ini digunakan untuk membandingkan account user yang dimasukkan dengan account user yang terdaftar dalam database.

**Specific Design Class NeracaSaldoControl\_Pusat**

<b>NeracaSaldoControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<pre>+get_itemNS1(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</pre> <pre>+get_itemNS2(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</pre> <pre>+get_totalNS1(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan total jumlah debit dan kredit rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</pre> <pre>+get_totalNS2(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan total jumlah debit dan kredit rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</pre>	

**Specific Design Class NeracaBankControl\_Pusat**

<b>NeracaBankControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>

```
+get_itemaktiva (string bulan,string tahun)
Operasi ini digunakan untuk mendapatkan data rekening aktiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.
+get_itempasiva(string bulan,string tahun)
Operasi ini digunakan untuk mendapatkan data rekening pasiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.
+get_totalaktiva(string bulan,string tahun)
Operasi ini digunakan untuk mendapatkan total jumlah aktiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.
+get_totalpasiva(string bulan,string tahun)
Operasi ini digunakan untuk mendapatkan total jumlah aktiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.
```

***Specific Design Class LabaRugiControl***

<b>LabarugiControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<pre>+get_itemJPB(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening pendapatan bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter. +get_itemJBB(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter. +get_itemJPO(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening pendapatan pendapatan operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter. +get_itemJBO(string bulan,string tahun)</pre>	

Operasi ini digunakan untuk mendapatkan data rekening beban operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_itemPBNO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan data rekening pendapatan beban operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalJPB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalJBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalPBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalJPO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah pendapatan operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalJBO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalLBO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah laba-rugi operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalPBNO

Operasi ini digunakan untuk mendapatkan total jumlah pendapatan dan beban non operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada

parameter.  
+get\_totalLBSPD(string bulan,string tahun)  
Operasi ini digunakan untuk mendapatkan total jumlah laba-rugi sebelum pajak penghasilan yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

*Specific                      Design                      Class                      Laporan*  
**KomitmendankontijensiControl\_Pusat**

KomitmendankontijensiControl	<<Control>>
<pre> +get_itemJTK Operasi ini digunakan untuk mendapatkan data rekening tagihan komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_itemJKK Operasi ini digunakan untuk mendapatkan data rekening kewajiban komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_itemJTKj Operasi ini digunakan untuk mendapatkan data rekening tagihan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_itemJKKj Operasi ini digunakan untuk mendapatkan data rekening kewajiban kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_totalJTK Operasi ini digunakan untuk mendapatkan total jumlah tagihan komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter  +get_totalJKK Operasi ini digunakan untuk mendapatkan total jumlah </pre>	

kewajiban komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJKB

Operasi ini digunakan untuk mendapatkan total jumlah komitmen bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJTKj

Operasi ini digunakan untuk mendapatkan total jumlah tagihan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJKKj

Operasi ini digunakan untuk mendapatkan total jumlah kewajiban kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJKjB

Operasi ini digunakan untuk mendapatkan total jumlah kontijensi bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

***Specific Design Class NET\_webReferencesControl***

<b>webReferencesControl</b>	<b>&lt;&lt;control&gt;&gt;</b>
+wsdl_itemNS1(string bulan,string tahun)	
Operasi ini digunakan untuk mendapatkan wsdl akses data rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.	
+wsdl_itemNS2(string bulan,string tahun)	
Operasi ini digunakan untuk mendapatkan wsdl akses data rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.	
+wsdl_totalNS1(string bulan,string tahun)	
Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah debit dan kredit rekening neraca saldo bagian	

neraca bank dan laba-rugi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalNS2(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah debet dan kredit rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemaktiva (string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening aktiva yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itempasiva(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening pasiva yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalaktiva(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah aktiva yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalpasiva(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah aktiva yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJPB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening pendapatan bunga yang ada pada basis data bank cabang sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening beban bunga yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJPO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening pendapatan pendapatan operasional yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJBO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening beban operasional yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemPBNO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening pendapatan beban operasional yang ada pada basis data cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJPB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah pendapatan bunga yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalPBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJPO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah pendapatan operasional yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJBO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total

jumlah pendapatan bunga yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalLBO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah laba-rugi operasional yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalPBNO

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah pendapatan dan beban non operasional yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalLBSPD(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah laba-rugi sebelum pajak penghasilan yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJTK

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening tagihan komitmen yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJKK

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening kewajiban komitmen yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJTKj

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening tagihan kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJKKj

Operasi ini digunakan untuk mendapatkan wsdl akses data

rekening kewajiban kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJTK

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah tagihan komitmen yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKK

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah kewajiban komitmen yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKB

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah komitmen bersih yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJTKj

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah tagihan kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKKj

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah kewajiban kontijensi yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKjB

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah kontijensi bersih yang ada pada basis data bank cabang A sesuai dengan bulan dan tahun yang ada pada parameter

**Specific Design Class JDev\_webReferencesControl**

<b>webReferencesControl</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+wsdl_itemNS1(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan wsdl akses data rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter. +wsdl_itemNS2(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan wsdl akses data rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter. +wsdl_totalNS1(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah debet dan kredit rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter. +wsdl_totalNS2(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah debet dan kredit rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter. +wsdl_itemaktiva (string bulan,string tahun) Operasi ini digunakan untuk mendapatkan wsdl akses data rekening aktiva yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter. +wsdl_itempasiva(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan wsdl akses data rekening pasiva yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter. +wsdl_totalaktiva(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan wsdl akses total</pre>	

jumlah aktiva yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalpasiva(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah aktiva yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJPB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening pendapatan bunga yang ada pada basis data bank cabang sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening beban bunga yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJPO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening pendapatan pendapatan operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJBO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening beban operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemPBNO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening pendapatan beban operasional yang ada pada basis data cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJPB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah pendapatan bunga yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada

parameter.

+wsdl\_totalJBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalPBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJPO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah pendapatan operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJBO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah pendapatan bunga yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalLBO(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah laba-rugi operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalPBNO

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah pendapatan dan beban non operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalLBSPD(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah laba-rugi sebelum pajak penghasilan yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJTK

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening tagihan komitmen yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJKK

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening kewajiban komitmen yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJTKj

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening tagihan kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJKKj

Operasi ini digunakan untuk mendapatkan wsdl akses data rekening kewajiban kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJTK

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah tagihan komitmen yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKK

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah kewajiban komitmen yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKB

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah komitmen bersih yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJTKj

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah tagihan kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKKj

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah kewajiban kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKjB

Operasi ini digunakan untuk mendapatkan wsdl akses total jumlah kontijensi bersih yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

***Specific Design Class LoginControl\_B***

<b>LoginControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<pre>+LoginControl() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +validasi() Operasi ini digunakan untuk membandingkan account user yang dimasukkan dengan account user yang terdaftar dalam database.</pre>	

***Specific Design Class NeracaSaldoControl\_B***

<b>NeracaSaldoControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<pre>+get_itemNS1(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</pre>	

+get\_itemNS2(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan data rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalNS1(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah debit dan kredit rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+get\_totalNS2(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah debit dan kredit rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

#### ***Specific Design Class NeracaBankControl\_B***

<b>NeracaBankControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<p>+get_itemaktiva (string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening aktiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itempasiva(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening pasiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_totalaktiva(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan total jumlah aktiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_totalpasiva(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan total jumlah</p>	

aktiva yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

***Specific Design Class LabaRugiControl\_B***

<b>LabarugiControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<p>+get_itemJPB(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening pendapatan bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itemJBB(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itemJPO(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening pendapatan pendapatan operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itemJBO(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening beban operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_itemPBNO(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan data rekening pendapatan beban operasional yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_totalJPB(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+get_totalJBB(string bulan,string tahun) Operasi ini digunakan untuk mendapatkan total jumlah beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.</p>	

```

+get_totalPBB(string bulan,string tahun)
Operasi ini digunakan untuk mendapatkan total jumlah
pendapatan bunga bersih yang ada pada basis data sesuai
dengan bulan dan tahun yang ada pada parameter.
+get_totalJPO(string bulan,string tahun)
Operasi ini digunakan untuk mendapatkan total jumlah
pendapatan operasional yang ada pada basis data sesuai
dengan bulan dan tahun yang ada pada parameter.
+get_totalJBO(string bulan,string tahun)
Operasi ini digunakan untuk mendapatkan total jumlah
pendapatan bunga yang ada pada basis data sesuai dengan
bulan dan tahun yang ada pada parameter.
+get_totalLBO(string bulan,string tahun)
Operasi ini digunakan untuk mendapatkan total jumlah
laba-rugi operasional yang ada pada basis data sesuai
dengan bulan dan tahun yang ada pada parameter.
+get_totalPBNO
Operasi ini digunakan untuk mendapatkan total jumlah
pendapatan dan beban non operasional yang ada pada basis
data sesuai dengan bulan dan tahun yang ada pada
parameter.
+get_totalLBSPD(string bulan,string tahun)
Operasi ini digunakan untuk mendapatkan total jumlah
laba-rugi sebelum pajak penghasilan yang ada pada basis
data sesuai dengan bulan dan tahun yang ada pada
parameter.

```

*Specific                      Design                      Class                      Laporan*

**KomitmandanKontijensiControl\_B**

<b>KomitmandanKontijensiControl</b>	<b>&lt;&lt;Control&gt;&gt;</b>
<pre> +get_itemJTK Operasi ini digunakan untuk mendapatkan data rekening tagihan komitmen yang ada pada basis data sesuai dengan </pre>	

bulan dan tahun yang ada pada parameter

+get\_itemJKK

Operasi ini digunakan untuk mendapatkan data rekening kewajiban komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_itemJTKj

Operasi ini digunakan untuk mendapatkan data rekening tagihan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_itemJKKj

Operasi ini digunakan untuk mendapatkan data rekening kewajiban kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJTK

Operasi ini digunakan untuk mendapatkan total jumlah tagihan komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJKK

Operasi ini digunakan untuk mendapatkan total jumlah kewajiban komitmen yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJKB

Operasi ini digunakan untuk mendapatkan total jumlah komitmen bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJTKj

Operasi ini digunakan untuk mendapatkan total jumlah tagihan kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJKKj

Operasi ini digunakan untuk mendapatkan total jumlah kewajiban kontijensi yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

+get\_totalJKjB

Operasi ini digunakan untuk mendapatkan total jumlah

kontijensi bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter

*Specific Design Class JDev\_wsdlControl*

webReferencesControl	<<control>>
<p>+wsdl_itemNS1(string bulan,string tahun) Operasi ini digunakan untuk memberikan <i>web service</i> akses data rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+wsdl_itemNS2(string bulan,string tahun) Operasi ini digunakan untuk memberikan <i>web service</i> akses data rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+wsdl_totalNS1(string bulan,string tahun) Operasi ini digunakan untuk memberikan <i>web service</i> akses total jumlah debit dan kredit rekening neraca saldo bagian neraca bank dan laba-rugi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+wsdl_totalNS2(string bulan,string tahun) Operasi ini digunakan untuk memberikan <i>web service</i> akses total jumlah debit dan kredit rekening neraca saldo bagian komitmen dan kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+wsdl_itemaktiva (string bulan,string tahun) Operasi ini digunakan untuk memberikan <i>web service</i> akses data rekening aktiva yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.</p> <p>+wsdl_itempasiva(string bulan,string tahun) Operasi ini digunakan untuk memberikan <i>web service</i> akses</p>	

data rekening pasiva yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalaktiva(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah aktiva yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalpasiva(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah aktiva yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJPB(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening pendapatan bunga yang ada pada basis data bank cabang sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJBB(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening beban bunga yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJPO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening pendapatan pendapatan operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJBO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening beban operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemPBNO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses data rekening pendapatan beban operasional yang ada pada basis data cabang B sesuai dengan bulan dan tahun yang

ada pada parameter.

+wsdl\_totalJPB(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah pendapatan bunga yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah beban bunga yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalPBB(string bulan,string tahun)

Operasi ini digunakan untuk mendapatkan total jumlah pendapatan bunga bersih yang ada pada basis data sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJPO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah pendapatan operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalJBO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah pendapatan bunga yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalLBO(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah laba-rugi operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalPBNO

Operasi ini digunakan untuk memberikan *web service* akses total jumlah pendapatan dan beban non operasional yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_totalLBSPD(string bulan,string tahun)

Operasi ini digunakan untuk memberikan *web service* akses total jumlah laba-rugi sebelum pajak penghasilan yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter.

+wsdl\_itemJTK

Operasi ini digunakan untuk memberikan *web service* akses data rekening tagihan komitmen yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJKK

Operasi ini digunakan untuk memberikan *web service* akses data rekening kewajiban komitmen yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJTKj

Operasi ini digunakan untuk memberikan *web service* akses data rekening tagihan kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_itemJKKj

Operasi ini digunakan untuk memberikan *web service* akses data rekening kewajiban kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJTK

Operasi ini digunakan untuk memberikan *web service* akses total jumlah tagihan komitmen yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKK

Operasi ini digunakan untuk memberikan *web service* akses total jumlah kewajiban komitmen yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKB

Operasi ini digunakan untuk memberikan *web service* akses total jumlah komitmen bersih yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJTKj

Operasi ini digunakan untuk memberikan *web service* akses total jumlah tagihan kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKKj

Operasi ini digunakan untuk memberikan *web service* akses total jumlah kewajiban kontijensi yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

+wsdl\_totalJKjB

Operasi ini digunakan untuk memberikan *web service* akses total jumlah kontijensi bersih yang ada pada basis data bank cabang B sesuai dengan bulan dan tahun yang ada pada parameter

***Specific Design Class userBank\_A***

<b>userbank</b>	<b>&lt;&lt;entity&gt;&gt;</b>
+getuserbank()	
Operasi ini digunakan untuk mendapatkan data user dari tabel userbank yang ada pada basis data bank cabang A.	

***Specific Design Class rekNeracaSaldo\_A***

<b>rekNeracaSaldo</b>	<b>&lt;&lt;entity&gt;&gt;</b>
+ getrekNeracaSaldo()	
Operasi ini digunakan untuk mendapatkan data rekening	

dari tabel rek\_neraca\_saldo yang ada pada basis data bank cabang A.

*Specific Design Class histNeracaSaldo\_A*

<b>histNeracaSaldo</b>	<b>&lt;&lt;entity&gt;&gt;</b>
+ gethistNeracaSaldo() Operasi ini digunakan untuk mendapatkan data histori dari tabel hist_neraca_saldo yang ada pada basis data bank cabang A.	

*Specific Design Class pos\_A*

<b>pos</b>	<b>&lt;&lt;entity&gt;&gt;</b>
+ getPos() Operasi ini digunakan untuk mendapatkan data pos rekening dari tabel pos yang ada pada basis data bank cabang A.	

*Specific Design Class userBank\_Pusat*

<b>userbank</b>	<b>&lt;&lt;entity&gt;&gt;</b>
+getuserbank() Operasi ini digunakan untuk mendapatkan data user dari tabel userbank yang ada pada basis data bank Pusat.	

*Specific Design Class rekNeracaSaldo\_Pusat*

<b>rekNeracaSaldo</b>	<b>&lt;&lt;entity&gt;&gt;</b>
+ getrekNeracaSaldo()	

Operasi ini digunakan untuk mendapatkan data rekening dari tabel rek\_neraca\_saldo yang ada pada basis data bank Pusat.

***Specific Design Class histNeracaSaldo\_Pusat***

<b>histNeracaSaldo</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>+ gethistNeracaSaldo()          Operasi ini digunakan untuk mendapatkan data histori dari tabel hist_neraca_saldo yang ada pada basis data bank Pusat.</p>	

***Specific Design Class pos\_Pusat***

<b>pos</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>+ getPos()          Operasi ini digunakan untuk mendapatkan data pos rekening dari tabel pos yang ada pada basis data bank Pusat.</p>	

***Specific Design Class userBank\_B***

<b>userbank</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<p>+getuserbank()          Operasi ini digunakan untuk mendapatkan data user dari tabel userbank yang ada pada basis data bank cabang B.</p>	

***Specific Design Class rekNeracaSaldo\_B***

<b>rekNeracaSaldo</b>	<b>&lt;&lt;entity&gt;&gt;</b>

+ getrekNeracaSaldo()

Operasi ini digunakan untuk mendapatkan data rekening dari tabel rek\_neraca\_saldo yang ada pada basis data bank cabang B.

#### *Specific Design Class histNeracaSaldo\_B*

<b>histNeracaSaldo</b>	<b>&lt;&lt;entity&gt;&gt;</b>
+ gethistNeracaSaldo() Operasi ini digunakan untuk mendapatkan data histori dari tabel hist_neraca_saldo yang ada pada basis data bank cabang B.	

#### *Specific Design Class pos\_B*

<b>pos</b>	<b>&lt;&lt;entity&gt;&gt;</b>
+ getPos() Operasi ini digunakan untuk mendapatkan data pos rekening dari tabel pos yang ada pada basis data bank cabang B.	

### **Deskripsi Dekomposisi**

#### *Dekomposisi Data*

#### **Deskripsi Entitas Data userbank**

Nama	Tipe	Panjang	Keterangan
Nama	Character	50	Nama user, Primary Key
Pass_word	Character	50	Password

#### **Deskripsi Entitas Data rek\_neraca\_saldo**

Nama	Tipe	Panjang	Keterangan
Id_rek	Character	10	ID rekening, Primary Key
Nama_rek	Character	10	Nama rekening
Id_sort	Number	38	Urutan rekening

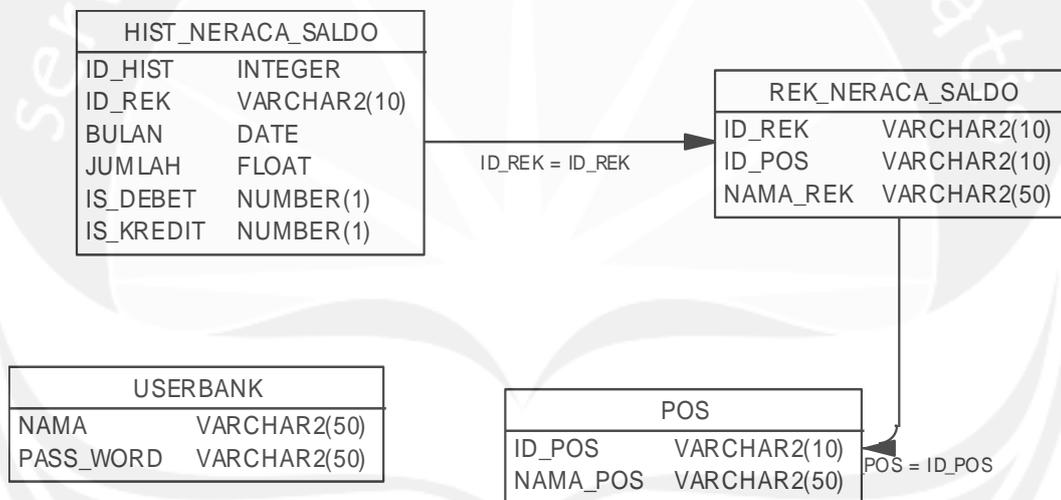
### Deskripsi Entitas Data hist\_neraca\_saldo

Nama	Tipe	Panjang	Keterangan
Id_hist	Number	38	ID histori, Primary Key
Tanggal	Date		Tanggal histori
Is_debet	Boolean	1	status untuk debet
Is_kredit	Boolean	1	status untuk debet

### Deskripsi Entitas Data pos

Nama	Tipe	Panjang	Keterangan
Id_pos	Character	10	ID pos, Primary Key
Nama_pos	Character	50	Nama pos rekening

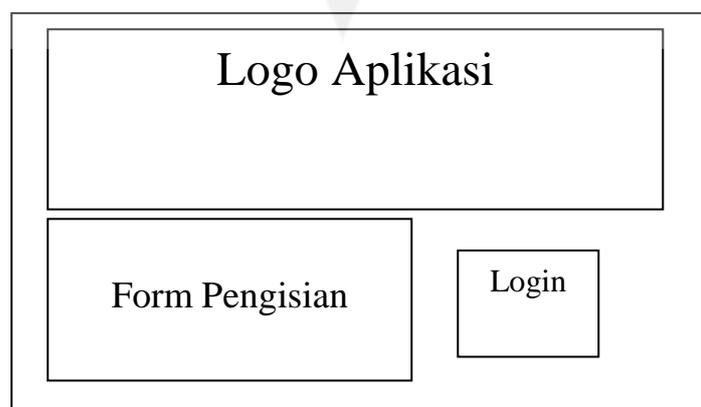
### Physical Data Model



Gambar 7 Physical Data Model

### Deskripsi Perancangan Antarmuka

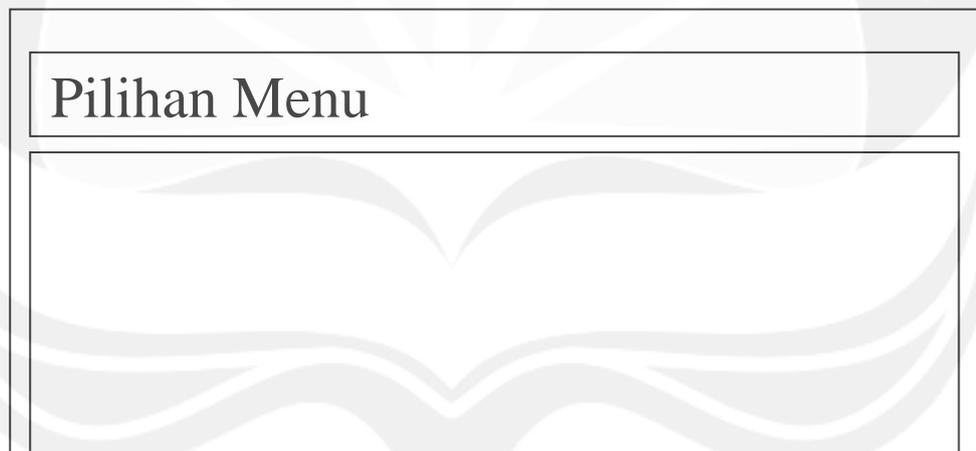
#### Login



*Gambar 8 Rancangan Antarmuka Login*

Antarmuka ini digunakan sebagai antarmuka untuk login. Pengguna (manajemen bank) akan membuka page ini pertama kali masuk ke dalam sistem. Yang perlu diinputkan adalah username dan password. Jika login berhasil, maka akan muncul akan masuk ke halaman index. Jika login tidak berhasil (username atau password yang diinputkan tidak sesuai dengan data yang ada pada basis data) akan muncul pesan kesalahan.

*Halaman Utama*



*Gambar 9 Rancangan Antarmuka Halaman Utama*

*Antarmuka ini ditampilkan ketika manajemen bank berhasil melakukan login. Antarmuka ini digunakan untuk halaman kerja dimana pengguna dapat memilih menu-menu yang tersedia.*

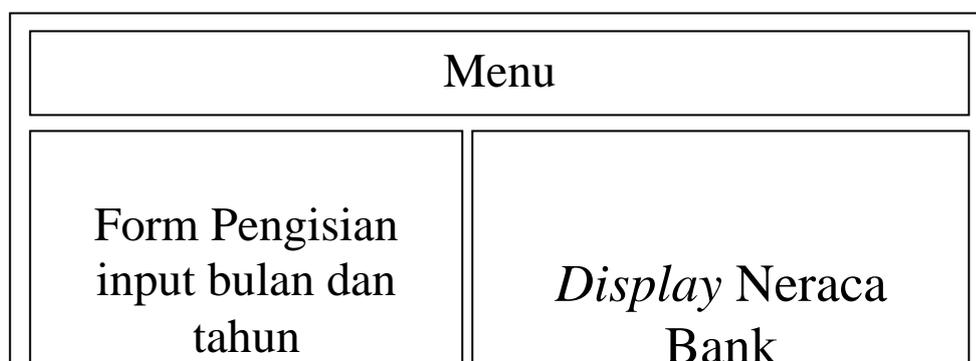
*Halaman Display Neraca Saldo*



*Gambar 10 Rancangan Antarmuka Display Neraca Saldo*

*Antarmuka ini ditampilkan ketika manajemen berhasil melakukan login dan memilih untuk melihat laporan neraca saldo. Antarmuka ini digunakan untuk melakukan memilih laporan neraca saldo berdasar bulan dan tahun*

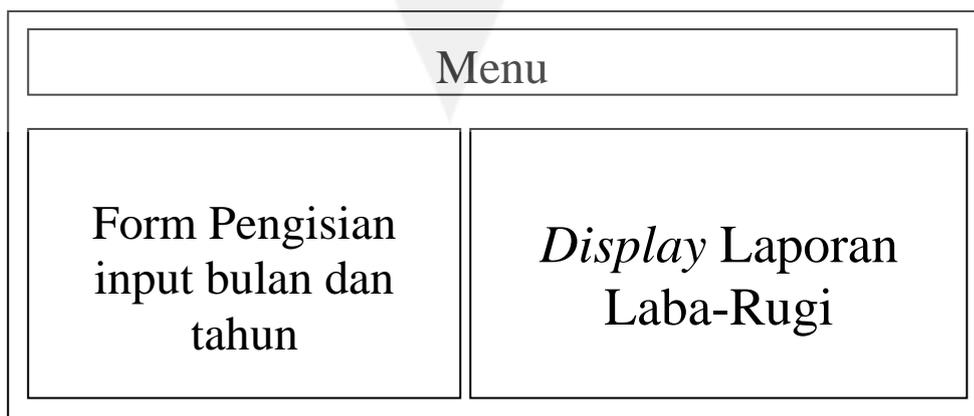
*Halaman Display Neraca Bank*



Gambar 11 Rancangan Antarmuka Display Neraca Bank

**Antarmuka ini ditampilkan ketika manajemen berhasil melakukan login dan memilih untuk melihat laporan neraca bank. Antarmuka ini digunakan untuk melakukan memilih laporan neraca bank berdasar bulan dan tahun**

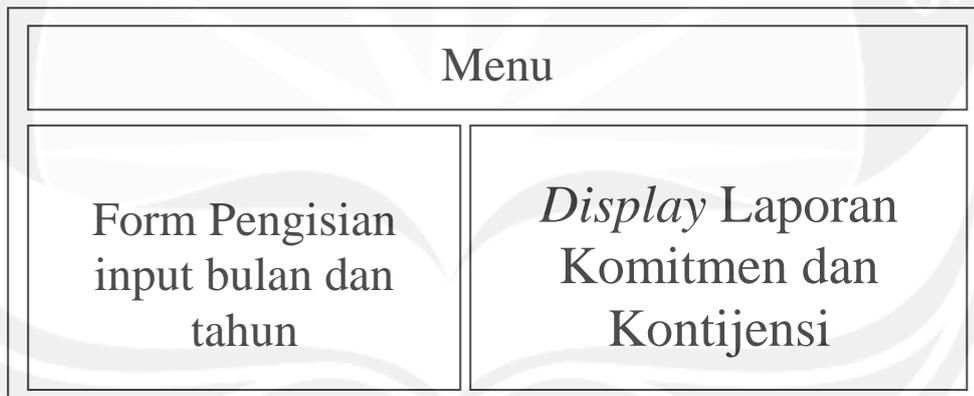
*Halaman Display Laporan Laba-Rugi*



Gambar 12 Rancangan Antarmuka Display Laporan Laba-Rugi

**Antarmuka ini ditampilkan ketika manajemen berhasil melakukan login dan memilih untuk melihat laporan laba-rugi. Antarmuka ini digunakan untuk melakukan memilih laporan laba-rugi berdasar bulan dan tahun**

Halaman Display Laporan Komitmen dan Kontijensi



Gambar 13 Rancangan Antarmuka Laporan Komitmen dan Kontijensi

**Antarmuka ini ditampilkan ketika manajemen berhasil melakukan login dan memilih untuk melihat laporan komitmen dan kontijensi. Antarmuka ini digunakan untuk melakukan memilih laporan komitmen dan kontijensi berdasar bulan dan tahun**

