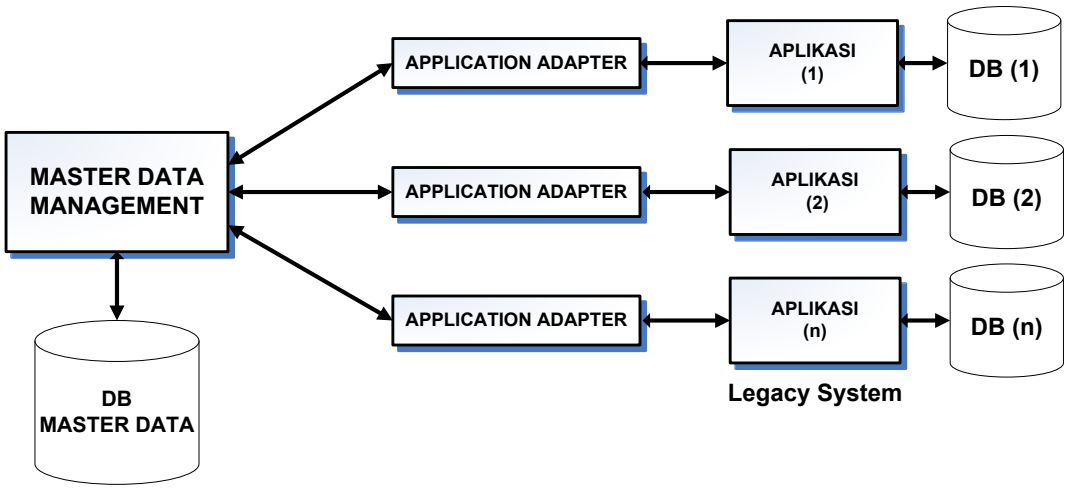
BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4. 1 Arsitektur Manajemen Master Data

4.1.1 Gambaran Umum Desain Arsitektur Manajemen Master Data

Desain Arsitektur Manajemen Master Data merupakan desain arsitektur yang mengintegrasikan aplikasi-aplikasi existing (*system legacy*) dengan master data menggunakan perantara komunikasi *application adapter* dalam interaksi *create*, *update* dan *delete* data master dari masing-masing aplikasi existing dan Master Data Management (MDM). Adapun bentuk Design Arsitektur Manajemen Master Data dapat dilihat pada Gambar 4.1 Design Arsitektur Manajemen Master Data.



Gambar 4.1 Design Arsitektur Manajemen Master Data

Berdasarkan gambar diatas, terdapat beberapa komponen utama pembentuk sistem kerja Arsitektur Manajemen Master Data, antara lain :

a. *Master Data Management (MDM)*

Master Data Management merupakan sebuah sistem yang mengatur dan menyimpan master data perusahaan yang melingkupi semua master data yang ada diseluruh aplikasi-aplikasi existing (*System Legacy*). MDM mengatur persebaran dan ketersediaan master data baik di MDM sendiri maupun di masing-masing aplikasi existing.

Secara prinsip setiap ada proses *create*, *updat*e dan *delete* data master di salah satu aplikasi existing akan melakukan penambahan data, perubahan data dan penghapusan data baik di MDM sendiri maupun di aplikasi existing lainnya.

b. *Database Master Data Management*

Database Master Data Management merupakan database repository yang menyimpan master data yang menjadi data master di semua aplikasi-aplikasi yang terhubung melalui perantara *application adapter*, data master di MDM harus dipastikan sama dengan data master yang ada di masing-masing aplikasi.

Prose *create*, *update* dan *delete* data master disalah satu database aplikasi existing akan dilakukan pula di database lain yang terhubung ke MDM.

c. *Application Adapter*

Application adapter adalah sebuah layer aplikasi yang mengkonversi data sehingga dapat diterima oleh aplikasi lain yang

terintegrasi kedalam sistem, dalam hal ini aplikasi-aplikasi existing terkoneksi kedalam *Master Data Management*.

d. System Legacy

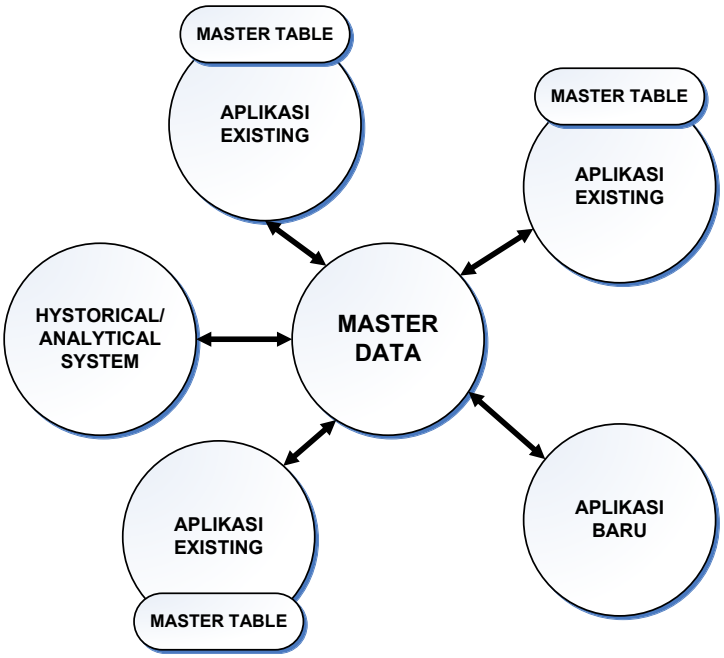
System Legacy adalah aplikasi-aplikasi existing yang memiliki database tersendiri, yang mana antara aplikasi satu dengan lainnya tidak saling berkomunikasi dalam hal integrasi data master aplikasi.

e. Database Aplikasi Existing

Database Aplikasi Existing adalah database dari masing-masing aplikasi existing yang hanya terhubung ke aplikasi yang bersangkutan. Secara content database aplikasi existing memiliki master data tersendiri dan ada kemungkinan data master yang sejenis dimiliki oleh database aplikasi existing lainnya, hal inilah yang melatarbelakangi perancangan arsitektur manajemen master data guna melakukan sinkronisasi data master dari masing-masing aplikasi existing.

Dari gambar 4.1 Design Arsitektur Manajemen Master Data, dapat dianalogikan Master Data Management menjadi poros komunikasi data master antar aplikasi-aplikasi existing yang terhubung ke MDM melalui *application adapter* pada masing-masing aplikasi. Dengan demikian, pola komunikasi data master antar aplikasi berpusat pada MDM yang menjadi perlintasan aplikasi-aplikasi yang terintegrasi untuk melakukan proses *create*, *update* dan *delete* data

master. Ilustrasi dari pola komunikasi antara data master MDM dengan data master masing-masing aplikasi dapat dilihat pada gambar 4.2 Pola Komunikasi Master Data dan Aplikasi Existing di bawah ini.



Gambar 4.2 Pola Komunikasi Master Data dan Aplikasi Existing

4.1.2 Use Case Diagram Arsitektur Manajemen Master Data

Use Case Diagram Arsitektur Manajemen Master Data memperlihatkan fungsionalitas yang diharapkan dari sebuah Sistem Manajemen Master Data, secara umum terdapat tiga fungsionalitas utama Arsitektur Manajemen Master Data yaitu *create master data*, *update master data* dan *delete master data*.

Dalam interaksi Manajemen Master Data, terdapat dua aktor utama yang berinteraksi menjalankan fungsionalitas utama dari Arsitektur Master Data Manajemen, dua aktor tersebut antara lain :

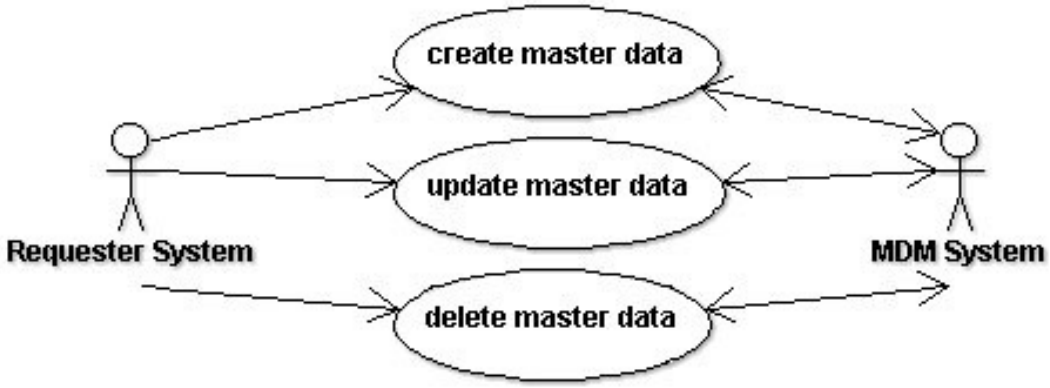
a. MDM System

MDM System merupakan aktor *use case* berupa sistem yang diperankan Master Data Management dalam mengelola master data utama serta melayani permintaan Requester System untuk proses *create master data*, *update master data* dan *delete master data*.

b. Requester System.

Requester System merupakan aktor use case berupa sistem yang diperankan aplikasi-aplikasi existing dengan bantuan *application adapter* untuk berintegrasi dengan Master Data Management. Requester System melakukan permintaan proses *create master data*, *update master data* dan *delete master data* terhadap MDM System, setiap permintaan *create master data*, *update master data* dan *delete master data* akan dilakukan pula kepada Requester System lain yang terhubung kedalam sistem Master Data Management.

Adapun gambaran dari Use Case Diagram Arsitektur Manajemen Master Data dapat dilihat pada Gambar 4.3 Use Case Diagram Arsitektur Manajemen Master Data.



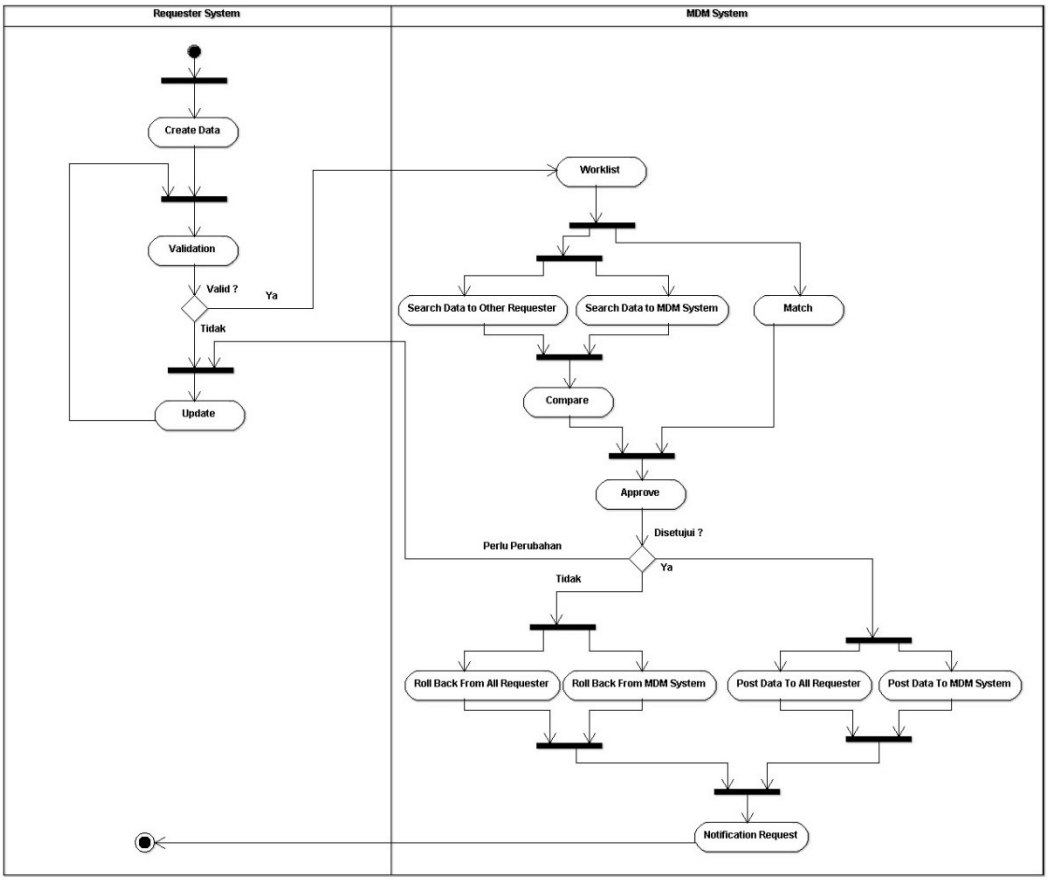
Gambar 4.3 Use Case Diagram Arsitektur Manajemen Master Data

4.1.3 Activity Diagram Arsitektur Manajemen Master Data

Activity Diagram Arsitektur Manajemen Master Data menunjukkan *workflow* sistem dengan penggambaran berbasis *flowchart*, secara umum Activity Diagram untuk Arsitektur Manajemen Master Data terdiri atas tiga aktifitas utama yaitu *create master data*, *update master data* dan *delete master data*.

a. *Activity Diagram Create Master Data*

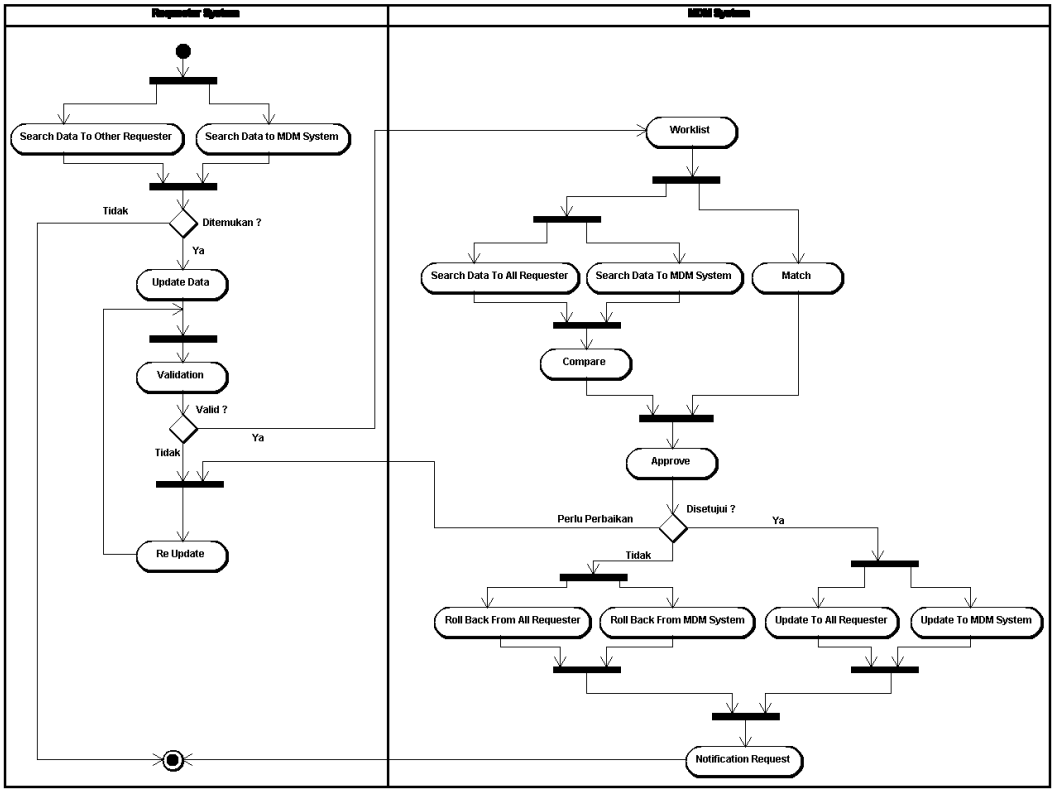
Activity Diagram Create Master Data merupakan *workflow* sistem yang menggambarkan permintaan *create master data* dari Requester System terhadap MDM System, seperti ditunjukkan pada gambar 4.4 Activity Diagram *Create Master Data*.



Gambar 4.4 Activity Diagram *Create Master Data*

b. *Activity Diagram Update Master Data*

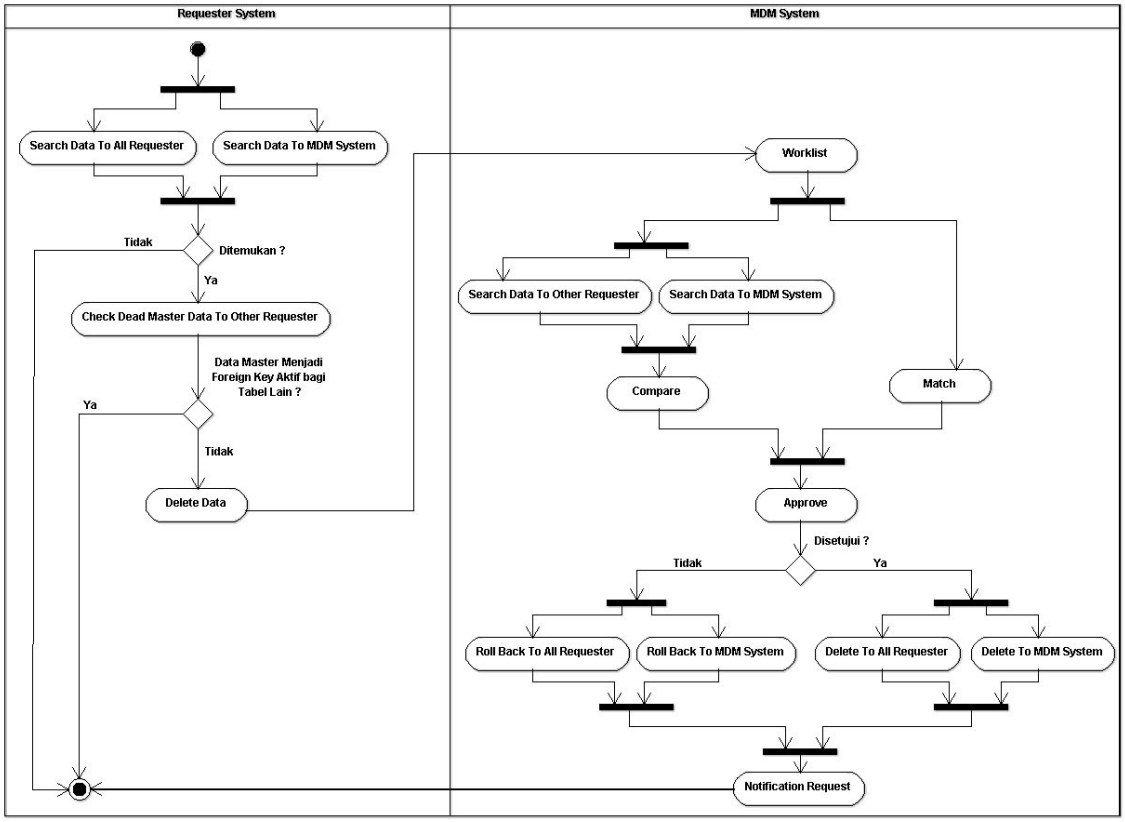
Activity Diagram Update Master Data merupakan *workflow* sistem yang menggambarkan permintaan *update master data* dari Requester System terhadap MDM System, seperti ditunjukkan pada gambar 4.5 Activity Diagram *Update Master Data*.



Gambar 4.5 Activity Diagram *Update Master Data*

c. *Activity Diagram Delete Master Data*

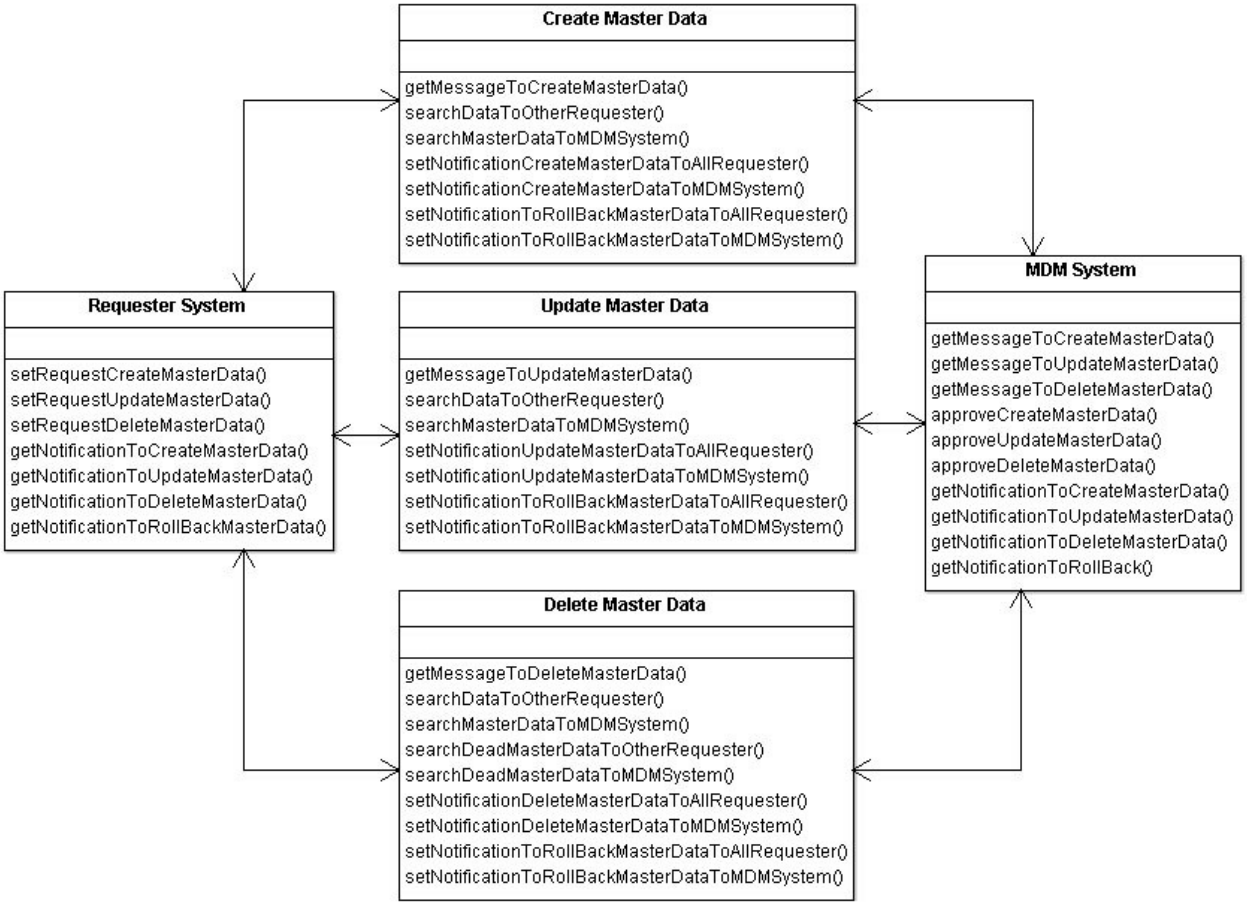
Activity Diagram Delete Master Data merupakan *workflow* sistem yang menggambarkan permintaan *delete master data* dari Requester System terhadap MDM System, seperti ditunjukkan pada gambar 4.6 Activity Diagram *Delete Master Data*.



Gambar 4.6 Activity Diagram *Delete Master Data*

4.1.4 Class Diagram Arsitektur Manajemen Master Data

Class Diagram Arsitektur Manajemen Master Data memperlihatkan hubungan antar Class dan Method yang dimiliki setiap Class dalam mendukung proses *Create Master Data*, *Update Master Data* dan *Delete Master Data*, seperti yang terlihat pada gambar 4.7 Class Diagram Arsitektur Manajemen Master Data.



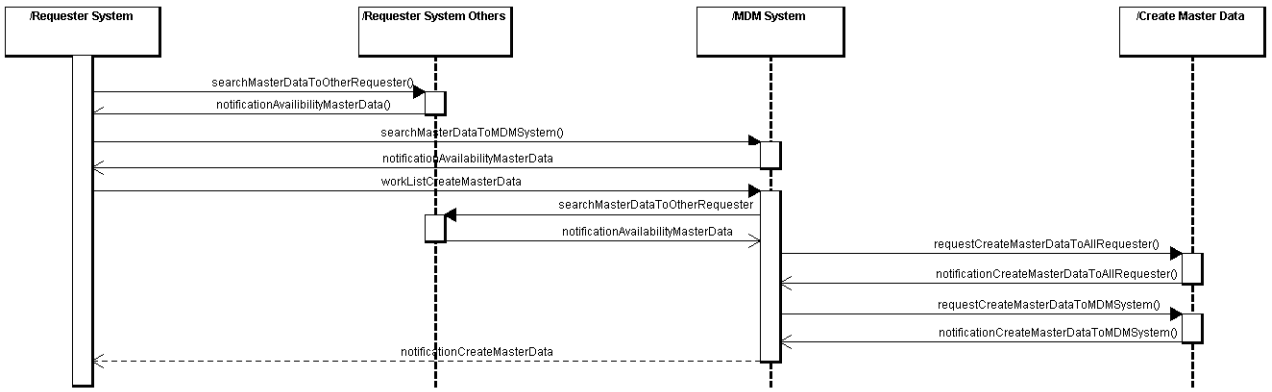
Gambar 4.7 Class Diagram Arsitektur Manajemen Master Data

4.1.5 Sequence Diagram Manajemen Master Data

Sequence diagram memperlihatkan interaksi yang terjadi berdasarkan rangkaian waktu dalam melakukan suatu proses, dalam hal ini terkait proses create master data, update master data dan delete master data. Berikut ini masing-masing sequence diagram untuk proses create master data, update master data dan delete master data.

a. Sequence Diagram Create Master Data

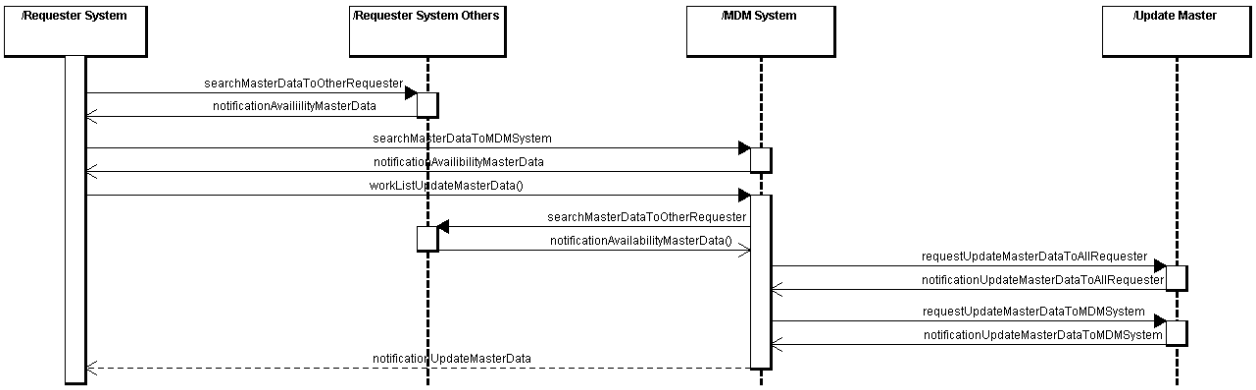
Sequence Diagram Create Master Data memperlihatkan interaksi yang terjadi berdasarkan rangkaian waktu untuk melakukan proses *create master data* seperti yang ditunjukkan pada gambar 4.8 *Sequence Diagram Create Master Data.*



Gambar 4.8 *Sequence Diagram Create Master Data*

b. Sequence Diagram Update Master Data

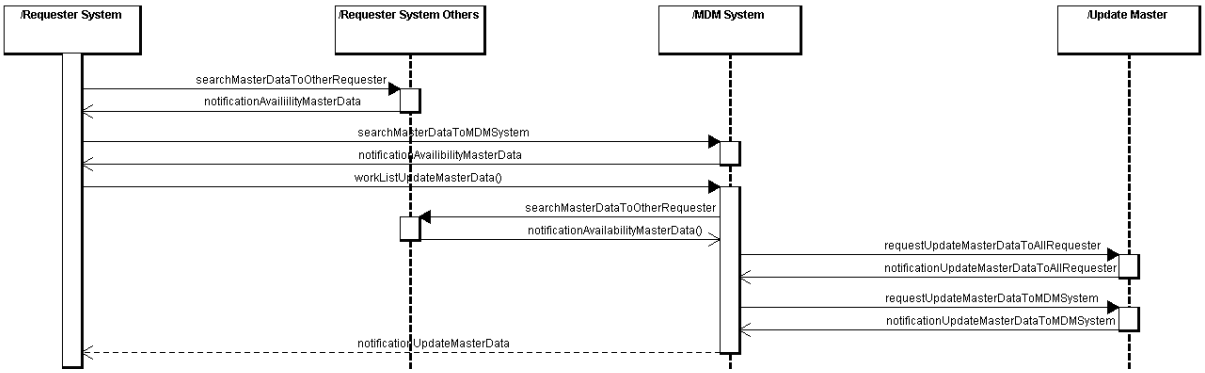
Sequence Diagram Update Master Data memperlihatkan interaksi yang terjadi berdasarkan rangkaian waktu untuk melakukan proses *update master data* seperti yang ditunjukkan pada gambar 4.9 *Sequence Diagram Update Master Data.*



Gambar 4.9 *Sequence Diagram Update Master Data*

c. Sequence Diagram Delete Master Data

Sequence Diagram Delete Master Data memperlihatkan interaksi yang terjadi berdasarkan rangkaian waktu untuk melakukan proses *delete master data* seperti yang ditunjukkan pada gambar 4.10 *Sequence Diagram Delete Master Data.*

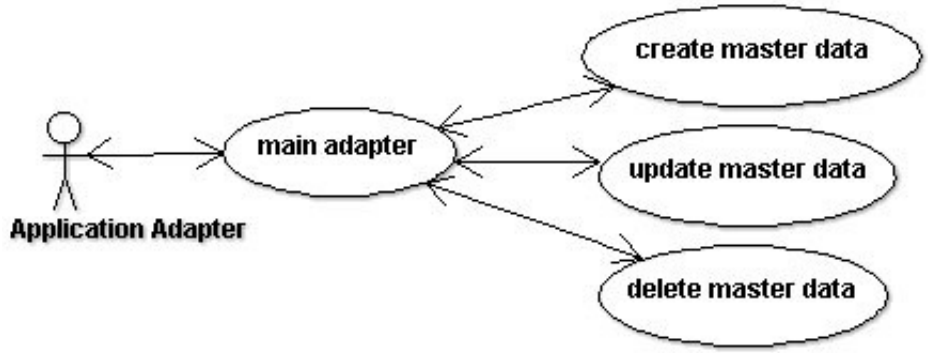


Gambar 4.10 *Sequence Diagram Delete Master Data*

4.1.6 Design Application Adapter

Design application adapter merupakan perancangan layer aplikasi yang mengkonversi data dari dan ke aplikasi lain yang terhubung dalam Master Data Management terkait permintaan *create master data*, *update master data* dan *delete master data* dari Requester System terhadap MDM System, *application adapter* seolah-olah menjadi perantara komunikasi yang menghubungkan keduanya.

Dari fungsionalitas utama Application Adapter dapat kita gambarkan dalam bentuk Use Case Diagram Application Adapter yang mencakup tiga fungsionalitas utama yaitu *create master data*, *update master data* dan *delete master data*. Adapun gambaran dari Use Case Diagram Application Adapter dapat dilihat pada Gambar 4.11 *Use Case Diagram Application Adapter*.



Gambar 4.11 *Use Case Diagram Application Adapter*

Selain dalam bentuk Use Case Diagram, perancangan Application Adapter ini digambarkan dalam bentuk *Activity Diagram*, *Class Diagram*, *Sequence Diagram* serta Penjelasan Medologi

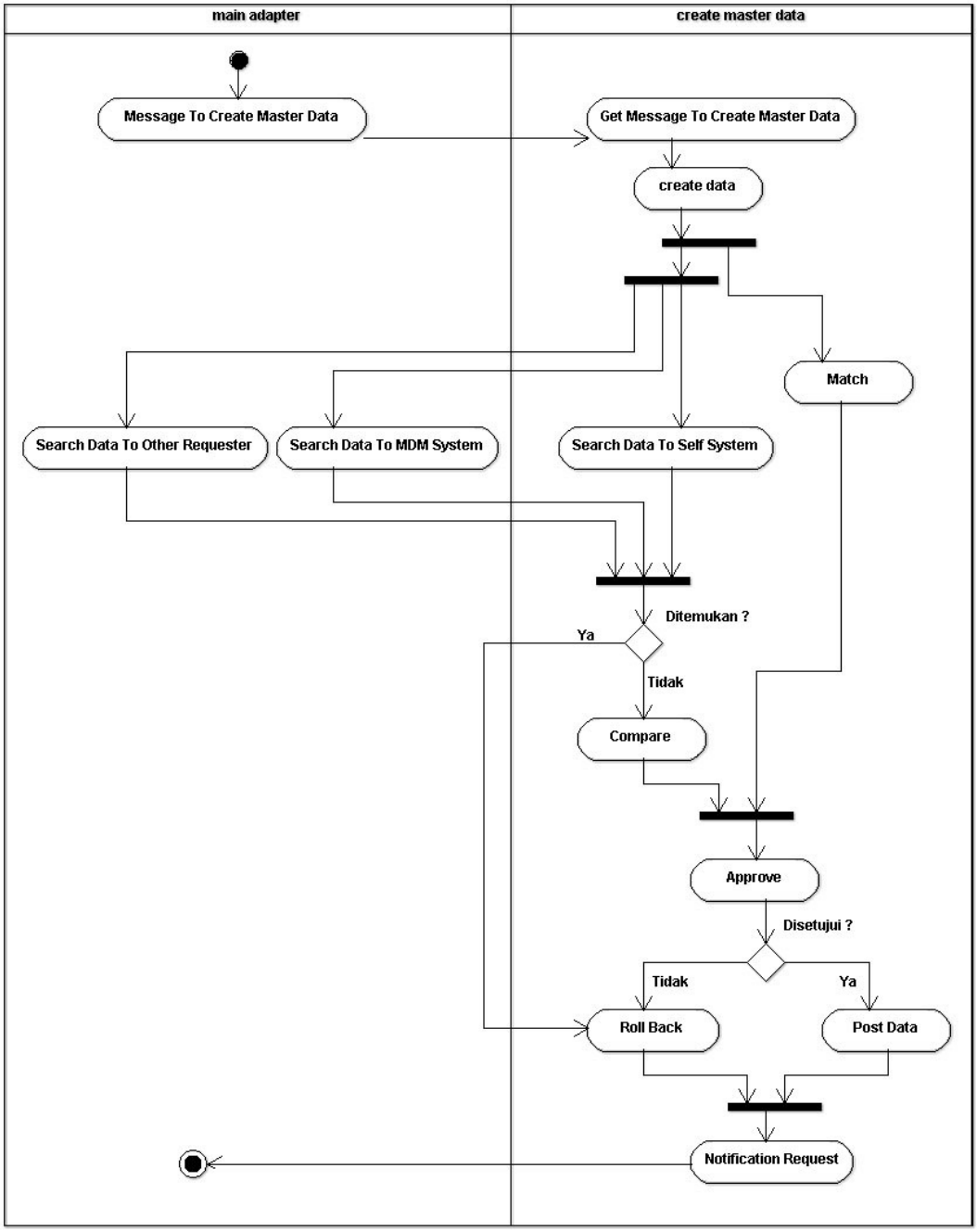
Manajemen Master Data dalam setiap proses yang dijalankan dalam suatu *Application Adapter*.

a. Activity Diagram Application Adapter

Activity Diagram Application Adapter menunjukkan *workflow* sistem dengan penggambaran berbasis *flowchart*, secara umum Activity Diagram untuk Application Adapter terdiri atas tiga aktifitas utama yaitu *create master data*, *update master data* dan *delete master data*.

1) Activity Diagram Create Master Data Application Adapter

Activity Diagram Create Master Data Application Adapter merupakan *workflow* sistem yang menggambarkan aktifitas *create master data* antara *main adapter* dan *create master data adapter*, seperti ditunjukkan pada gambar 4.12 *Activity Diagram Create Master Data Application Adapter*.



4.12 *Activity Diagram Create Master Data Application Adapter*

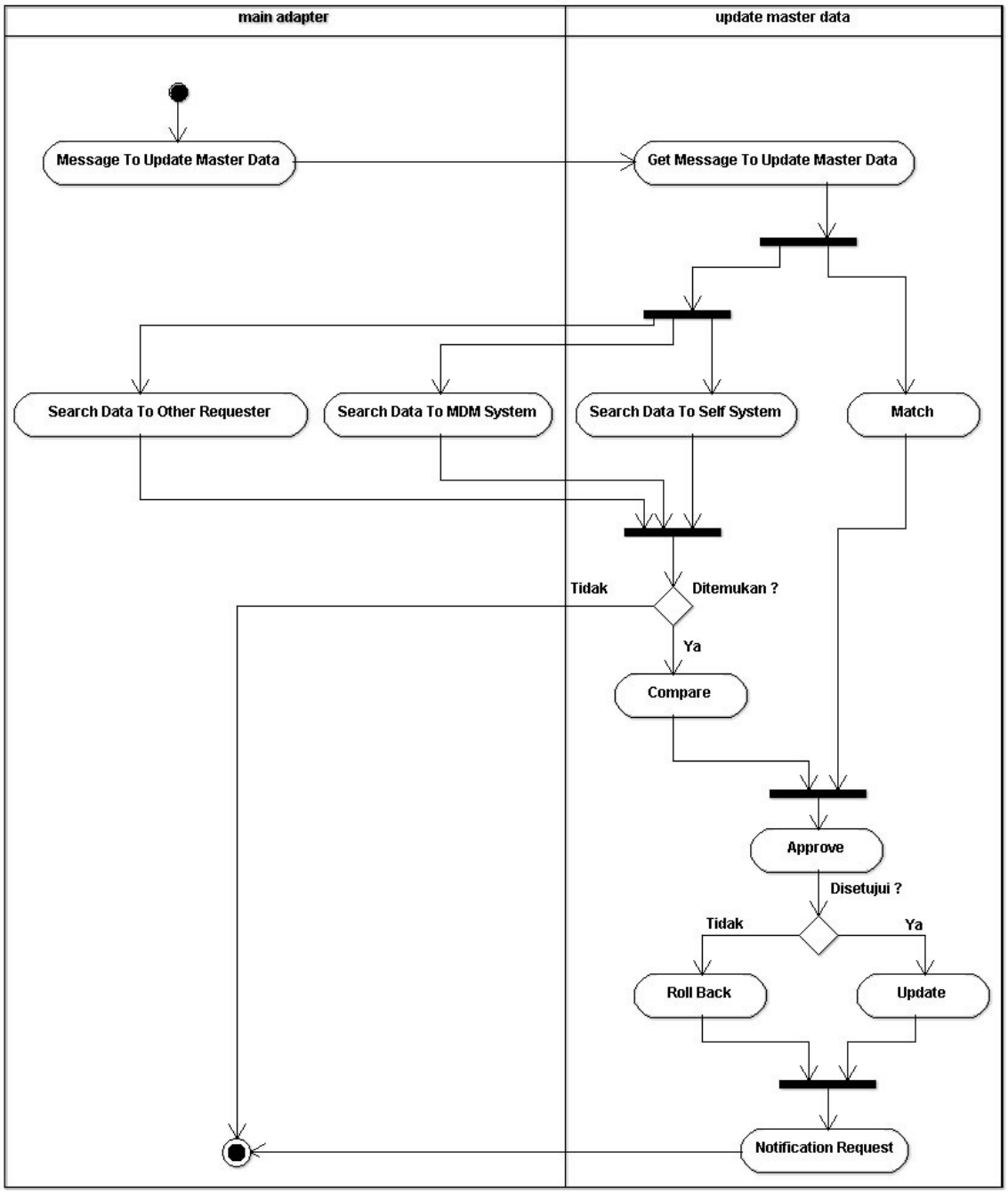
Dari proses diatas dapat dijelaskan bagaimana aktifitas yang terjadi saat *create master data* pada Application Adapter, dimulai dari pengiriman Message To Create Master Data dari

main adapter terhadap *create master data* yang memberikan respon Get Message To Create Master Data, untuk selanjutnya melakukan proses Create Data. Setelah Create Data dilakukan proses pencarian data ke dalam Aplikasi tempat Application Adapter berada (Search Data To Self System), Sistem MDM (Search Data To MDM System) serta Sistem lain yang terhubung ke MDM (Search Data To Other Requester).

Setelah tidak ditemukan baik di Sistem MDM maupun Sistem lain yang terhubung ke MDM dilakukan proses Compare dan Match setelah itu dilakukan proses Approve untuk melakukan proses approvisasi request create data apakah disetujui atau tidak. Jika tidak setujui dilakukan proses Roll Back, jika disetujui maka lakukan Post Data. Setelah itu diberikan Notification Request baik untuk Roll Back maupun Post Data terhadap *main adapter*.

2) Activity Diagram Update Master Data Application Adapter

Activity Diagram Update Master Data Application Adapter merupakan *workflow* sistem yang menggambarkan aktifitas *update master data* antara *main adapter* dan *update master dat adapter*, seperti ditunjukkan pada gambar 4.13 *Activity Diagram Update Master Data Application Adapter.*



4.13 *Activity Diagram Update Master Data Application Adapter*

Dari proses diatas dapat dijelaskan bagaimana aktifitas yang terjadi saat *update master data* pada Application Adapter, dimulai dari pengiriman Message To Update Master Data

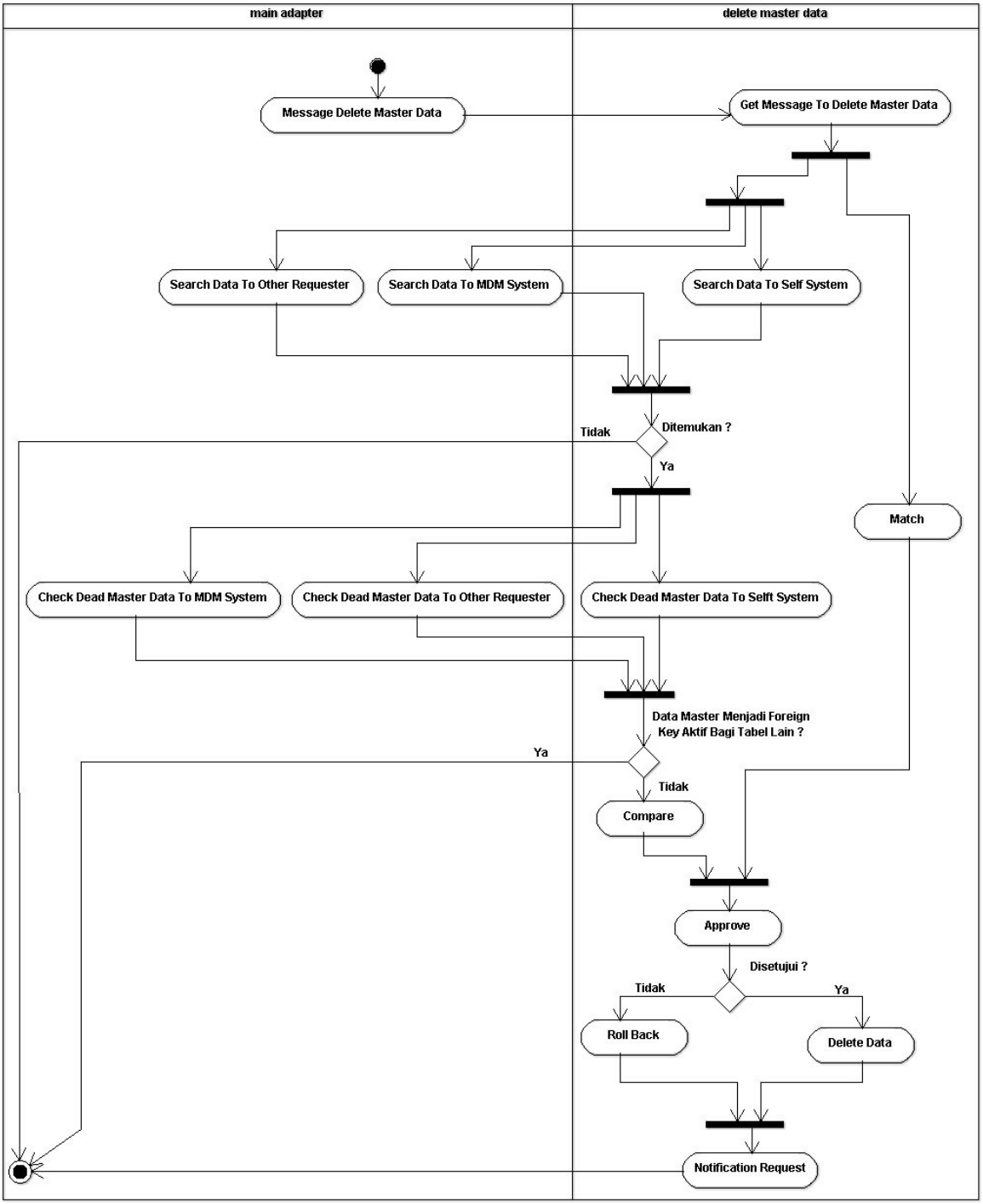
dari *main adapter* terhadap *update master data* yang memberikan respon Get Message To Update Master Data, untuk selanjutnya melakukan proses pencarian data ke dalam Aplikasi tempat Application Adapter berada (Search Data To Self System), Sistem MDM (Search Data To MDM System) serta Sistem lain yang terhubung ke MDM (Search Data To Other Requester).

Setelah ditemukan baik di Aplikasi tempat Application Adapter berada, Sistem MDM dan Sistem lain yang terhubung ke MDM dilakukan proses Compare dan Match untuk membandingkan data yang ditemukan, kemudian dilakukan proses Approve untuk melakukan proses approvisasi request update data apakah disetujui atau tidak. Jika tidak setujui dilakukan proses Roll Back, jika disetujui maka lakukan Update Data. Setelah itu diberikan Notification Request baik untuk Roll Back maupun Update Data terhadap *main adapter*.

3) Activity Diagram Delete Master Data Application Adapter

Activity Diagram Delete Master Data Application Adapter merupakan *workflow* sistem yang menggambarkan permintaan *delete master data* antara *main adapter* dan *update master dat adapter*, seperti ditunjukkan pada gambar

4.14 *Activity Diagram Delete Master Data Application Adapter*.



4.14 *Activity Diagram Delete Master Data Application Adapter*

Dari proses diatas dapat dijelaskan bagaimana aktifitas yang terjadi saat *delete master data* pada Application Adapter, dimulai dari pengiriman Message To Delete Master Data dari *main adapter* terhadap *delete master data* yang memberikan respon Get Message To Delete Master Data, untuk selanjutnya melakukan proses pencarian data ke dalam Aplikasi tempat Application Adapter berada (Search Data To Self System), Sistem MDM (Search Data To MDM System) serta Sistem lain yang terhubung ke MDM (Search Data To Other Requester).

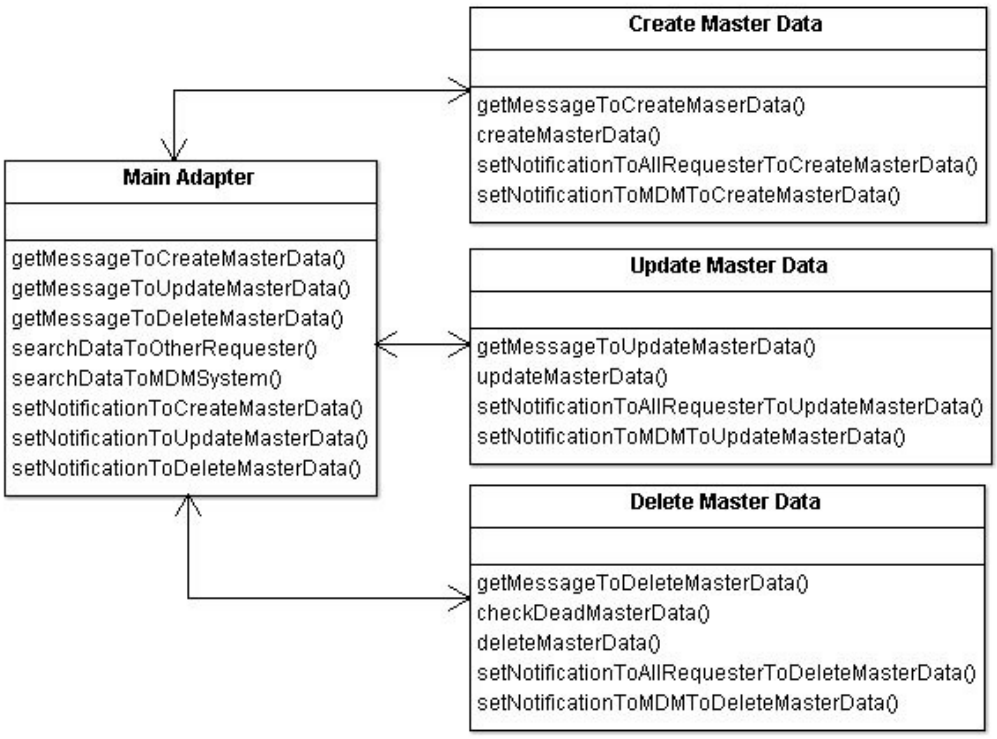
Setelah ditemukan baik di Aplikasi tempat Application Adapter berada, Sistem MDM dan Sistem lain yang terhubung ke MDM dilakukan proses Check Dead Master Data ke dalam Aplikasi tempat Application Adapter berada (Check Dead Master Data To Self System), Sistem MDM (Check Dead Master Data To MDM System) serta Sistem lain yang terhubung ke MDM (Check Dead Master Data To Other Requester).

Jika data yang di check bukanlah data yang menjadi Foreign Key bagi tabel lain baik di Sistem lain yang terhubung ke MDM, maka dilakukan Compare dan Match untuk membandingkan data yang ditemukan, kemudian dilakukan proses Approve untuk melakukan proses approvisasi request

delete data apakah disetujui atau tidak. Jika tidak setujui dilakukan proses Roll Back, jika disetujui maka lakukan Delete Data. Setelah itu diberikan Notification Request baik untuk Roll Back maupun Update Data terhadap *main adapter*.

b. Class Diagram Application Adapter

Class Diagram Application Adapter memperlihatkan hubungan antar Class dan Method yang dimiliki setiap Class dalam mendukung proses *Create Master Data*, *Update Master Data* dan *Delete Master Data* pada *Application Adapter*. Berikut ini *Class Diagram Application Adapter* yang ditunjukkan gambar 4.15 *Class Diagram Application Adapter*.



Gambar 4.15 *Class Diagram Application Adapter*

Dari gambaran Class Diagram Application Adapter diatas memperlihatkan class-class yang terbentuk dari suatu arsitektur *Application Adapter* yang mana terdiri dari class *Main Adapter*, *Create Master Data*, *Update Master Data* dan *Delete Master Data*.

Secara umum setiap proses request create, update ataupun delete akan melewati terlebih dahulu main adapter yang merespon setiap permintaan, untuk selanjutnya permintaan tersebut didistribusikan ke dalam class yang terkait, jika request yang diberikan berupa *create master data* maka request akan diteruskan ke dalam *class create master data*.

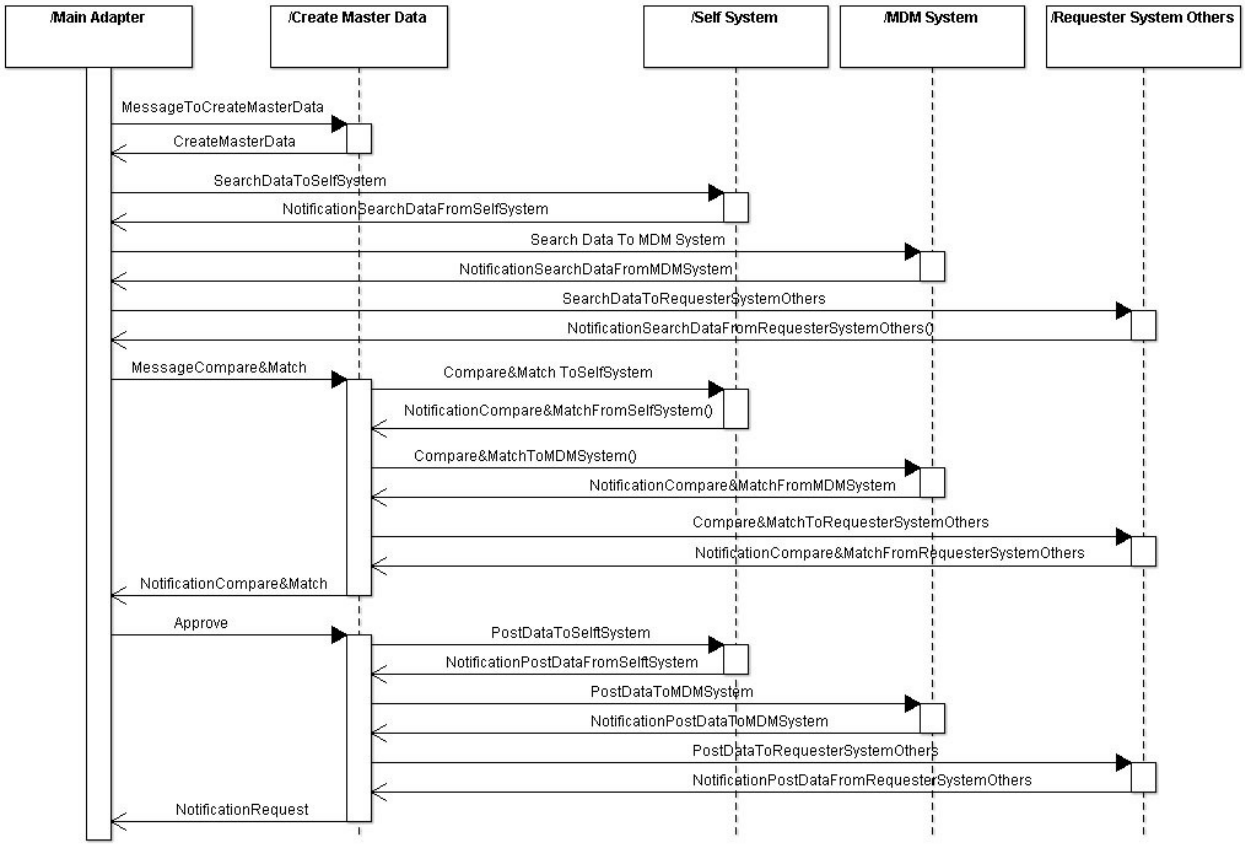
Jika request yang diberikan berupa *update master data* maka request akan diteruskan ke dalam *class update master data*. Dan jika request yang diberikan berupa *delete master data* maka request akan diteruskan delam *class delete master data*.

c. Sequnce Diagram Application Adapter

Sequence diagram memperlihatkan interaksi yang terjadi berdasarkan rangkaian waktu dalam melakukan suatu proses, dalam hal ini terkait proses *create master data*, update master data dan *delete master data* pada *Application Adapter*. Berikut ini masing-masing sequence diagram application adapter untuk proses *create master data*, *update master data* dan *delete master data*.

1) Sequence Diagram Create Master Data Application Adapter

Sequence Diagram Create Master Data Application Adapter memperlihatkan interaksi yang terjadi berdasarkan rangkaian waktu untuk melakukan proses *create master data* pada *Application Adapter* seperti yang ditunjukkan pada gambar 4.16 *Sequence Diagram Create Master Data Application Adapter.*



Gambar 4.16 *Sequence Diagram Create Master Data Application Adapter*

Dari gambar Sequence Diagram Create Master Data Application Adapter diatas dapat kita jelaskan logika

medologi *create master data* berupa algoritma kerja Application Adapter dalam melakukan proses create master data, antara lain :

[1] Pengiriman Pesan dari Main Adapter ke Create Master Data
Pesan untuk melakukan proses create master data pada Sistem tempat Application Adapter berada (misalnya create master data tabel x)

[2] Respon Pesan dari Create Master Data terhadap Main Adapter
- Respon pertama berupa penerimaan pesan create master data yaitu GetMessageCreateMasterData untuk create data pada tabel x.
- Respon kedua berupa aksi melakukan create master data tabel x dengan melakukan perintah:
  INSERT INTO table x (field-1,field-2,...,field-n) values ('value-1', 'value-2',...,'value-n')

[3] Respon Main Adapter setelah Proses CreateMasterData
Main Adapter melakukan pencarian data tabel x yang sama pada MDM System dan Requester System lain yang terhubung ke MDM System.
- Pencarian data pada MDM System :
  SELECT * FROM MDM.tabel x WHERE field-1='value-1'
  RETURN v_data
  // Keterangan :
  * Jika pencarian data ada, kita inisialisasikan v_data = 1
  * Jika pencarian data tidak ada kita inisialisasikan v_data=0
- Pencarian data pada Requester System lain :
  SELECT * FROM SISTEM1.tabel x WHERE field-1='value-1'
  RETURN v_data1
  // Keterangan :
  * Jika pencarian data ada, kita inisialisasikan v_data1 = 1
  * Jika pencarian data tidak ada kita inisialisasikan v_data1=0

SELECT * FROM SISTEM2.tabel x WHERE field-1='value-1'
RETURN v_data2
// Keterangan :
* Jika pencarian data ada, kita inisialisasikan v_data2 = 1
* Jika pencarian data tidak ada kita inisialisasikan v_data2=0

SELECT * FROM SISTEMn.tabel x WHERE field-1='value-1'
RETURN v_datan
// Keterangan :
* Jika pencarian data ada, kita inisialisasikan v_data n = 1
* Jika pencarian data tidak ada kita inisialisasikan v_data n=0

- Jika RETURN dari salah satu atau semua bernilai 1 maka proses create master data di tolak dan dilakukan Proses ROLL BACK
- Jika RETURN semua bernilai 0 maka proses dilanjutkan.

[4] Respon Create Master Data untuk melakukan Proses Compare & Match pada Create Master Data

Suatu respon melakukan penyesuaian isi data terhadap field-field pada tabel x, baik tabel x di sistem tempat Application Adapter berada, tabel x di MDM System maupun tabel x yang berada di sistem lain yang terhubung di MDM System.

[5] Proses Approvisasi Create Master Data

Suatu respon melakukan approvisasi create master data untuk tabel x, baik tabel x di sistem tempat Application Adapter berada, tabel x di MDM System maupun tabel x yang berada di sistem lain yang terhubung di MDM System.

- Jika proses create master data disetujui berikan pesan untuk melakukan proses create master data di tabel x di MDM System maupun tabel x yang berada di sistem lain yang terhubung di MDM System.
  INSERT INTO MDM.table x (field-1,field-2,...,field-n) values ('value-1', 'value-2',...,'value-n')

INSERT INTO SISTEM1.table x (field-1,field-2,...,field-n) values ('value-1', 'value-2',...,'value-n')

INSERT INTO SISTEM2.table x (field-1,field-2,...,field-n) values ('value-1', 'value-2',...,'value-n')

INSERT INTO SISTEMn.table x (field-1,field-2,...,field-n) values ('value-1', 'value-2',...,'value-n')

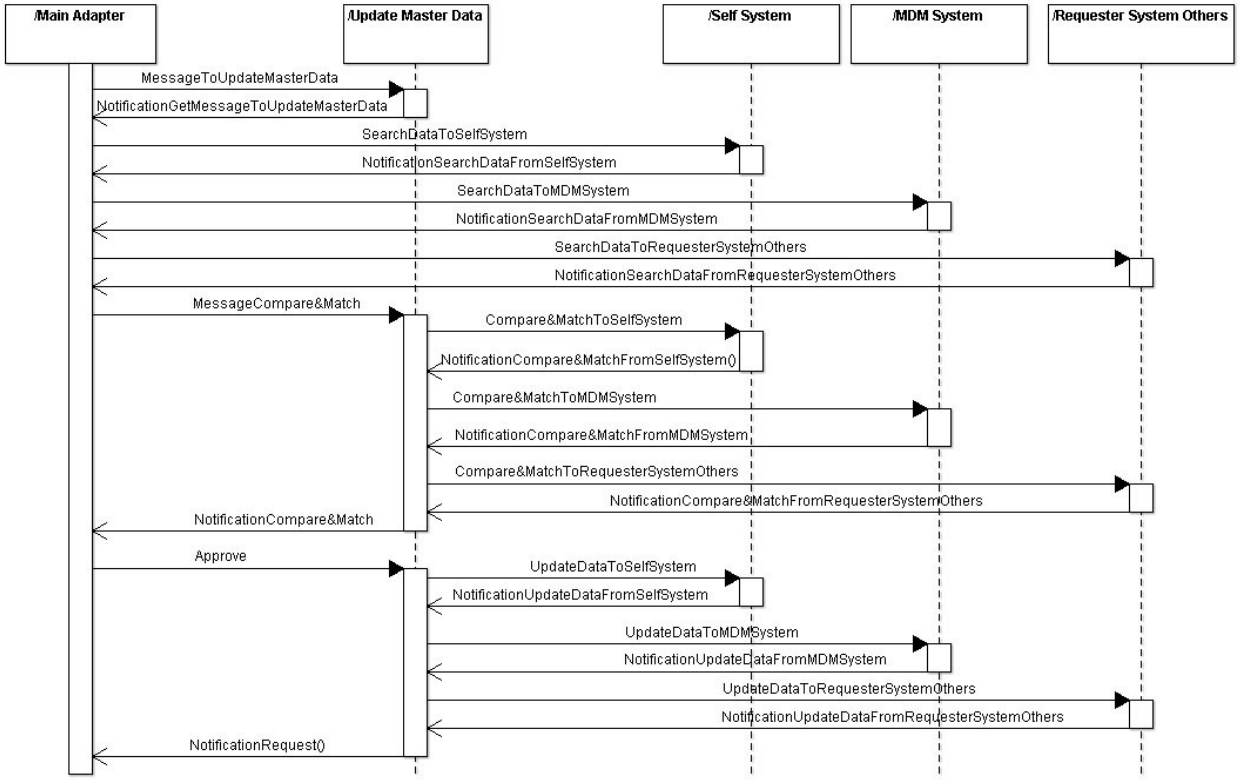
- Jika proses create master data tidak disetujui lakukan ROLL BACK data pada tabel x di sistem tempat Application Adapter berada.

[6] Create Master Data memberikan Pesan Notifikasi kepada Main Adapter.

- Jika Proses ROLL BACK yang dilakukan berarti proses Create Master Data gagal dilakukan.
- Jika Proses CREATE MASTER DATA yang dilakukan maka lakukan proses pengiriman pesan ke MDM System dan Sistem lain yang terhubung ke MDM System untuk melakukan proses create master data yang sama sebagai master data baru.

2) Sequence Diagram Update Master Data Application Adapter

Sequence Diagram Update Master Data Application Adapter memperlihatkan interaksi yang terjadi berdasarkan rangkaian waktu untuk melakukan proses *update master data* pada *Application Adapter* seperti yang ditunjukkan pada gambar 4.17 *Sequence Diagram Update Master Data Application Adapter.*



Gambar 4.17 *Sequence Diagram Update Master Data Application Adapter*

Dari gambar Sequence Diagram Update Master Data Application Adapter diatas dapat kita jelaskan logika medologi *update master data* berupa algoritma kerja Application Adapter dalam melakukan proses update master data, antara lain :

[1] Pengiriman Pesan dari Main Adapter ke Update Master Data
Pesan untuk melakukan proses update master data pada Sistem tempat Application Adapter berada (misalnya update master data tabel x)

[2] Respon Pesan dari Update Master Data terhadap Main Adapter

Respon pertama berupa penerimaan pesan update master data yaitu GetMessageUpdateMasterData untuk update data pada tabel x.

Respon kedua berupa aksi melakukan update master data tabel x dengan melakukan perintah:

UPDATE table x SET field-2='value-2',..., field-n='value-n' WHERE field-1='value-1'

[3] Respon Main Adapter setelah Proses UpdateMasterData

Main Adapter melakukan pencarian data tabel x yang sama pada MDM System dan Requester System lain yang terhubung ke MDM System.

- Pencarian data pada MDM System :

  SELECT * FROM MDM.tabel x WHERE field-1='value-1'
  RETURN v_data
  // Keterangan :
  * Jika pencarian data ada, kita inisialisasikan v_data = 1
  * Jika pencarian data tidak ada kita inisialisasikan v_data=0

- Pencarian data pada Requester System lain :

  SELECT * FROM SISTEM1.tabel x WHERE field-1='value-1'
  RETURN v_data1
  // Keterangan :
  * Jika pencarian data ada, kita inisialisasikan v_data1 = 1
  * Jika pencarian data tidak ada kita inisialisasikan v_data1=0

  SELECT * FROM SISTEM2.tabel x WHERE field-1='value-1'
  RETURN v_data2
  // Keterangan :
  * Jika pencarian data ada, kita inisialisasikan v_data2 = 1
  * Jika pencarian data tidak ada kita inisialisasikan v_data2=0

  SELECT * FROM SISTEMn.tabel x WHERE field-1='value-1'
  RETURN v_datan

// Keterangan :
* Jika pencarian data ada, kita inisialisasikan v_data n = 1
* Jika pencarian data tidak ada kita inisialisasikan v_data n=0

- Jika RETURN dari salah satu atau semua bernilai 0 maka proses update master data di tolak.
- Jika RETURN semua bernilai 1 maka proses dilanjutkan.

[4] Respon Update Master Data untuk melakukan Proses Compare & Match pada Update Master Data

Suatu respon melakukan penyesuaian isi data terhadap field-field pada tabel x, baik tabel x di sistem tempat Application Adapter berada, tabel x di MDM System maupun tabel x yang berada di sistem lain yang terhubung di MDM System.

[5] Proses Approvisasi Update Master Data

Suatu respon melakukan approvisasi update master data untuk tabel x, baik tabel x di sistem tempat Application Adapter berada, tabel x di MDM System maupun tabel x yang berada di sistem lain yang terhubung di MDM System.

- Jika proses update master data disetujui berikan pesan untuk melakukan proses update master data di tabel x di MDM System maupun tabel x yang berada di sistem lain yang terhubung di MDM System.

  UPDATE MDM.table x SET field-2='value-2',..., field-n='value-n' WHERE field-1='value-1'

  UPDATE SISTEM1.table x SET field-2='value-2',..., field-n='value-n' WHERE field-1='value-1'

  UPDATE SISTEM2.table x SET field-2='value-2',..., field-n='value-n' WHERE field-1='value-1'

  UPDATE SISTEMn.table x SET field-2='value-2',..., field-n='value-n' WHERE field-1='value-1'

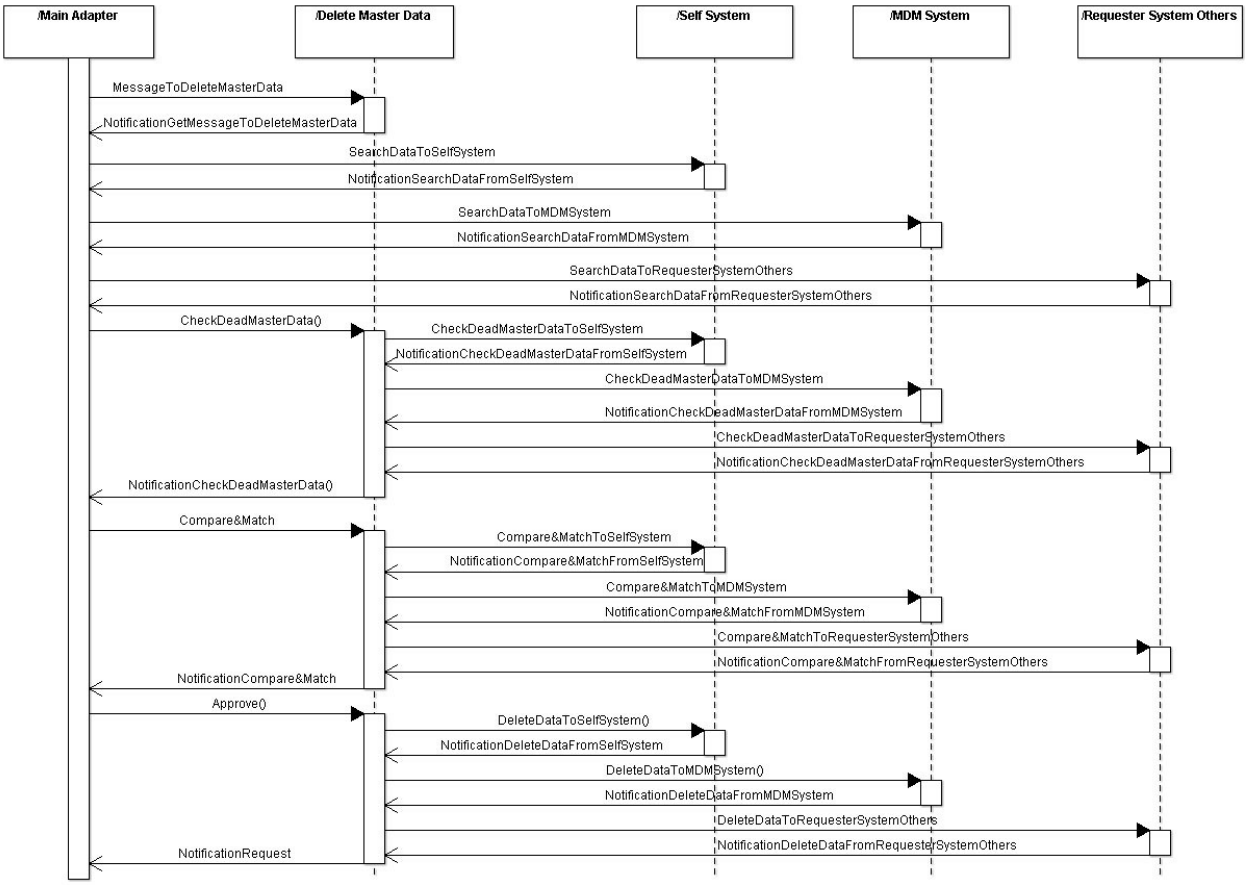
- Jika proses update master data tidak disetujui lakukan ROLL BACK data pada tabel x di sistem tempat Application Adapter berada.

[6] Update Master Data memberikan Pesan Notifikasi kepada Main Adapter

- Jika Proses ROLL BACK yang dilakukan berarti proses Update Master Data gagal dilakukan.
- Jika Proses UPDATE MASTER DATA yang dilakukan maka lakukan proses pengiriman pesan ke MDM System dan Sistem lain yang terhubung ke MDM System untuk melakukan proses update master data yang sama.

3) Sequence Diagram Delete Master Data Application Adapter

Sequence Diagram Delete Master Data Application Adapter memperlihatkan interaksi yang terjadi berdasarkan rangkaian waktu untuk melakukan proses *delete master data* pada *Application Adapter* seperti yang ditunjukkan pada gambar 4.18 *Sequence Diagram Delete Master Data Application Adapter.*



Gambar 4.18 *Sequence Diagram Delete Master Data Application Adapter*

Dari gambar Sequence Diagram Delete Master Data Application Adapter diatas dapat kita jelaskan logika medologi *delete master data* berupa algoritma kerja Application Adapter dalam melakukan proses delete master data, antara lain :

[1] Pengiriman Pesan dari Main Adapter ke Delete Master Data
Pesan untuk melakukan proses delete master data pada Sistem tempat Application Adapter berada (misalnya update master data tabel x)

[2] Respon Pesan dari Delete Master Data terhadap Main Adapter

Respon pertama berupa penerimaan pesan delete master data yaitu GetMessageDeleteMasterData untuk delete data pada tabel x.

Respon kedua berupa aksi melakukan delete master data tabel x dengan melakukan perintah:

DELETE FROM table x WHERE field-1='value-1'

[3] Respon Main Adapter setelah Proses DeleteMasterData (1)

Main Adapter melakukan pencarian data tabel x yang sama pada MDM System dan Requester System lain yang terhubung ke MDM System.

- Pencarian data pada MDM System :

  SELECT * FROM MDM.tabel x WHERE field-1='value-1'

  RETURN v_data

  // Keterangan :

  * Jika pencarian data ada, kita inisialisasikan v_data = 1

  * Jika pencarian data tidak ada kita inisialisasikan v_data=0

- Pencarian data pada Requester System lain :

  SELECT * FROM SISTEM1.tabel x WHERE field-1='value-1'

  RETURN v_data1

  // Keterangan :

  * Jika pencarian data ada, kita inisialisasikan v_data1 = 1

  * Jika pencarian data tidak ada kita inisialisasikan v_data1=0

  SELECT * FROM SISTEM2.tabel x WHERE field-1='value-1'

  RETURN v_data2

  // Keterangan :

  * Jika pencarian data ada, kita inisialisasikan v_data2 = 1

  * Jika pencarian data tidak ada kita inisialisasikan v_data2=0

  SELECT * FROM SISTEMn.tabel x WHERE field-1='value-1'

  RETURN v_datan

// Keterangan :
* Jika pencarian data ada, kita inisialisasikan v_data n = 1
* Jika pencarian data tidak ada kita inisialisasikan v_data n=0

- Jika RETURN dari salah satu atau semua bernilai 0 maka proses delete master data di tolak.
- Jika RETURN semua bernilai 1 maka proses dilanjutkan.

[4] Respon Main Adapter setelah Proses DeleteMasterData (2)

Main Adapter melakukan pencarian apakah data pada tabel x yang akan didelete berperan sebagai foreign key yang aktif pada tabel-tabel lain baik pada MDM System dan Requester System lain yang terhubung ke MDM System.

- Jika salah satu atau semua data yang akan didelete berperan sebagai FOREIGN KEY aktif maka proses delete master data di tolak.
- Jika semua data yang akan didelete berperan sebagai FOREIGN KEY tidak aktif maka proses dilanjutkan.

[5] Respon Delete Master Data untuk melakukan Proses Compare & Match pada Delete Master Data

Suatu respon melakukan penyesuaian isi data terhadap field-field pada tabel x, baik tabel x di sistem tempat Application Adapter berada, tabel x di MDM System maupun tabel x yang berada di sistem lain yang terhubung di MDM System.

[6] Proses Approvisasi Delete Master Data

Suatu respon melakukan approvisasi delete master data untuk tabel x, baik tabel x di sistem tempat Application Adapter berada, tabel x di MDM System maupun tabel x yang berada di sistem lain yang terhubung di MDM System.

- Jika proses delete master data disetujui berikan pesan untuk melakukan proses delete master data di tabel x di MDM System maupun tabel x yang berada di sistem lain yang terhubung di MDM System.

DELETE FROM MDM.table x WHERE field-1='value-1'

DELETE FROM SISTEM1.table x WHERE field-1='value-1'

DELETE FROM SISTEM2.table x WHERE field-1='value-1'

DELETE FROM SISTEMn.table x WHERE field-1='value-1'

- Jika proses delete master data tidak disetujui lakukan ROLL BACK data pada tabel x di sistem tempat Application Adapter berada.

[7] Delete Master Data memberikan Pesan Notifikasi kepada Main Adapter

- Jika Proses ROLL BACK yang dilakukan berarti proses Delete Master Data gagal dilakukan.
- Jika Proses DELETE MASTER DATA yang dilakukan maka lakukan proses pengiriman pesan ke MDM System dan Sistem lain yang terhubung ke MDM System untuk melakukan proses delete master data yang sama

4. 2 Studi Kasus Manajemen Master Data PT. Jayamandiri Gemasejati

4.2.1 Design Arsitektur Manajemen Master Data PT. Jayamandiri Sejati

Desain Arsitektur Manajemen Master Data PT. Jayamandiri Gemasejati merupakan desain arsitektur yang mengintegrasikan aplikasi-aplikasi existing yang terintegrasi kedalam sistem Master Data Management , adapun aplikasi existing tersebut antara lain :

a. Aplikasi Administrasi Perusahaan

   Aplikasi yang mengelola data karyawan, data aktifitas absensi, permohonan pengajuan, helpdesk, informasi internal perusahaan, dll.

b. Aplikasi Penjualan Motor (Sales)

   Aplikasi yang mengelola ketersediaan dan transaksi penjualan motor.

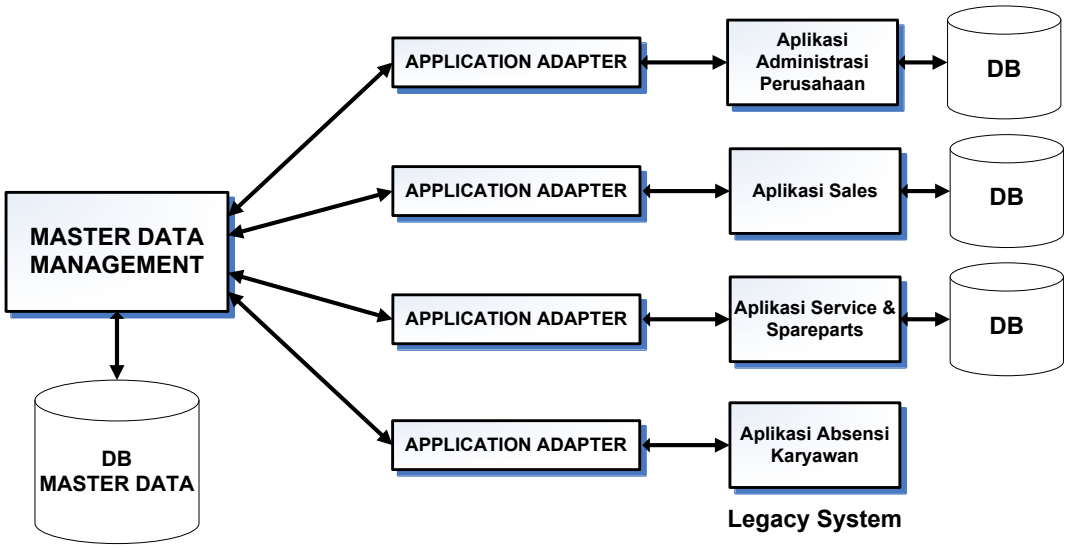
c. Aplikasi Service & Spareparts

   Aplikasi yang mengelola transaksi service, data dan transaksi spare parts.

d. Aplikasi Absensi Karyawan

   Aplikasi yang mengelola kegiatan absensi karyawan melalui mesin fingerprint yang tersebar ke seluruh cabang dan terkoneksi pada aplikasi absensi yang tersimpan di masing-masing cabang.

Bentuk Design Arsitektur Manajemen Master Data dapat dilihat pada Gambar 4.19 Design Arsitektur Manajemen Master Data PT. Jayamandiri Gemasejati.



Gambar 4.19 Design Arsitektur Manajemen Master Data PT. Jayamandiri Gemasejati

4.2.2 Identifikasi Master Data Perusahaan

Identifikasi master data perusahaan merupakan serangkaian proses dalam menghasilkan master data yang menjadi acuan untuk seluruh aplikasi yang ada berdasarkan mekanisme proses bisnis, entitas dan hubungan antar entitas yang ada dan mencakup seluruh proses bisnis yang dijalankan semua aplikasi existing, hal ini dilakukan guna mendukung perubahan kebutuhan bisnis dan bisnis yang berbeda.

Identifikasi master data merupakan bagian dari proses *Clean* and *Govern Master Data* yang melibatkan proses standarisasi,

pendefinisan atribut, *record* dan *business rule* tabel untuk memastikan data sesuai *requirement* dan aturan integritas data.

Dalam proses identifikasi master data perusahaan dihasilkan sebanyak 52 tabel master yang tersimpan pada database master data. Berikut ini tabel-tabel yang dihasilkan sebagai master data perusahaan, antara lain :

a. Unit

Tabel 4.1 unit

No.	Atribut	Type	Keterangan	Contoh
1.	unitId	Char(6)	Kode Unit Motor	UN0001
2.	unitNama	Varchar(150)	Nama Unit Motor	V-IXION
3.	cabangId	Char(6)	Kode Cabang	JG0001
4.	unitStatus	Enum('y','n')	Status Aktif Unit Motor	y

b. unitColor

Tabel 4.2 unitColor

No.	Atribut	Type	Keterangan	Contoh
1.	unitColorId	Char(15)	Kode Warna Unit Motor	JG0001UN0001 C01
2.	unitId	Char(6)	Kode Unit Motor	UN0001
3.	unitColorNama	Varchar(100)	Nama Warna Unit	Black Silver
4.	cabangId	Char(6)	Kode Cabang	JG0001
5.	unitColorStatus	Enum('y','n')	Status Aktif Warna Unit Motor	y

c. unitStock

Tabel 4.3 unitStock

No.	Atribut	Type	Keterangan	Contoh
1.	unitStockId	Char(17)	Kode Stock Unit Motor	JG0001UN0001S1207
2.	unitId	Char(6)	Kode Unit Motor	UN0001
3.	unitColorId	Char(15)	Kode Warna Unit Motor	JG0001UN0001C01
4.	cabangId	Char(6)	Kode Cabang	JG0001
5.	stock	Integer (5)	Stock Unit Motor	1000
6.	priceOffRoad	Decimal (15,2)	Harga Off Road Unit Motor	22.500.000,00
7.	priceOnRoad	Decimal (15,2)	Harga On Road Unit Motor	24.000.000,00
8.	discountCash	Decimal (15,2)	Diskon Cash Unit Motor	500.000,00
9.	discountCredit	Decimal (15,2)	Diskon Kredit Unit Motor	1.000.000,00
10.	unitStockStatus	Enum('y','n')	Status Aktif Stock Unit	y

d. unitStockDetail

Tabel 4.4 unitStockDetail

No.	Atribut	Type	Keterangan	Contoh
1.	unitStockId	Char(17)	Kode Stock Unit Motor	JG0001UN0001S1207
2.	unitStockDetailId	Char(20)	Kode Stock Detail Unit Motor	JG0001UN0001S1207001
3.	frameNo	Varchar(50)	Nomor Frame Unit Motor	MH35BP0068K114326
4.	machineNo	Varchar(50)	Nomor Mesin Unit Motor	5BP-114454
5.	stockDetailDate	Date	Tanggal Stock Detail Unit Motor	2012-07-07
6.	unitStockDetailStatus	Enum('y', 'n')	Status Aktif Stock Detail Unit	y

e. cabang

Tabel 4.5 cabang

No.	Atribut	Type	Keterangan	Contoh
1.	cabangId	Char(6)	Kode Cabang	JG0001
2.	cabangNama	Varchar(100)	Nama Cabang JG	JG BANDUNG
3.	Alamat	Varchar(150)	Alamat Cabang JG	Jalam BKR No. 5
4.	kelurahanId	Char(10)	Kode Kelurahan Cabang	1001001001
5.	kecamatanId	Char(7)	Kode Kecamatan Cabang	1001001
6.	kotaId	Char(4)	Kode Kota Cabang	1001
7.	propinsiId	Char(4)	Kode Propinsi Cabang	1000
8.	telpNo	Varchar(13)	Nomor Telepon Cabang	0227806029
9.	faxNo	Varchar(13)	Nomor Fax Cabang	0227806030
10.	rekeningNo	Varchar(75)	Nomor Rekening Cabang	BCA 434-306-1100
11.	cabangStatus	Enum('y','n')	Status Aktif Cabang	y

f. customer

Tabel 4.6 customer

No.	Atribut	Type	Keterangan	Contoh
1.	customerId	Char(17)	Kode Customer	JG0001CU120707001
2.	customerNama	Varchar(100)	Nama Customer	Asep Muhammad Indra Purnama
3.	customerPanggilan	Varchar(100)	Nama Panggilan Customer	Amip
4.	jenisKelamin	Enum('L','P')	Jenis Kelamin Customer	L = Laki-Laki P = Perempuan
5.	lahirTempat	Varchar(100)	Tempat Lahir Customer	Bandung
6.	lahirTanggal	Date	Tanggal Lahir Customer	1985-02-26
7.	Alamat	Varchar(150)	Alamat Customer	Jalan Pasir Malaka No. 26

Tabel 4.6 customer (lanjutan)

No.	Atribut	Type	Keterangan	Contoh
8.	kelurahanId	Char(10)	Kelurahan Customer	1001001001
9.	telpNo	Varchar(13)	Nomor Telepon Customer	0227806029
10.	hpNo	Varchar(15)	Nomor Handphone Customer	081809419562
11.	identitasId	Varchar(10)	Kode Identitas Customer	KTP
12.	identitasNo	Varchar(50)	Nomor Identitas Customer	3204072602850001
13.	pekerjaanId	Char(6)	Kode Pekerjaan Customer	KER001
14.	pendidikanId	Char(7)	Kode Pendidikan Customer	PEND001
15.	agama	Enum('ISLAM', 'KKALOTIK', 'KPROTESTAN', 'HINDU', 'BUDHA', 'KEPERCAYA-AN')	Agama Customer	ISLAM
16.	customerCabangId	Char(6)	Kode Cabang	JG0001
17.	customerType	Varchar(4)	Tipe Customer	CUUP
18.	customerStatus	Enum('y','n')	Status Aktif Customer	y

g. customerUnit

Tabel 4.7 customerUnit

No.	Atribut	Type	Keterangan	Contoh
1.	customerId	Char(17)	Kode Customer	JG0001CU120707001
2.	customerUnitId	Char(21)	Kode Customer Unit	JG0001CU120707001UN01
3.	frameNo	Varchar(50)	Nomor Frame Unit Motor	MH35BP0068K114326
4.	machineNo	Varchar(50)	Nomor Mesin Unit Motor	5BP-114454
5.	polisiNo	Varchar(11)	Nomor Polisi	D3908GT
6.	customerUnitStatus	Enum('y','n')	Status Aktif Customer Unit	y

h. customerServiceParts

Tabel 4.8 customerServiceParts

No.	Atribut	Type	Keterangan	Contoh
1.	customerId	Char(17)	Kode Customer	JG0001CU120707001
2.	customerServicePartsId	Char(21)	Kode Customer Unit	JG0001CU120707001SP01
3.	frameNo	Varchar(50)	Nomor Frame Unit Motor	MH35BP0068K114326
4.	machineNo	Varchar(50)	Nomor Mesin Unit Motor	5BP-114454
5.	polisiNo	Varchar(11)	Nomor Polisi	D3908GT
6.	tahunProduksi	Varchar(5)	Tahun Produksi Motor	2012
7.	customerServiceParts Status	Enum('y','n')	Status Aktif Customer Service Parts	y

i. identitas

Tabel 4.9 identitas

No.	Atribut	Type	Keterangan	Contoh
1.	identitasId	Varchar(10)	Kode Identitas	KTP
2.	identitasNama	Varchar(100)	Nama Identitas	Kartu Tanda Penduduk
3.	identitasSifat	Enum('I','G')	Sifat Identitas	I = Individu G = Group

j. propinsi

Tabel 4.10 propinsi

No.	Atribut	Type	Keterangan	Contoh
1.	propinsiId	Char(4)	Kode Propinsi	1000
2.	propinsiNama	Varchar(100)	Nama Propinsi	Jawa Barat
3.	propinsiIbuKota	Varchar(100)	Ibu Kota Propinsi	Bandung
4.	propinsiStatus	Enum('y','n')	Status Aktif Propinsi	y

k. kota

Tabel 4.11 kota

No.	Atribut	Type	Keterangan	Contoh
1.	kotaId	Char(4)	Kode Kota	1001
2.	kotaNama	Varchar(100)	Nama Kota	Kota Bandung
3.	ibuKota	Varchar(100)	Nama Ibu Kota	Bandung
4.	kotaType	Enum('KOTA','KAB')	Tipe Kota	KOTA
5.	propinsiId	Char(4)	Kode Propinsi	1000
6.	kotaStatus	Enum('y','n')	Status Aktif Kota	y

l. kecamatan

Tabel 4.12 kecamatan

No.	Atribut	Type	Keterangan	Contoh
1.	kecamatanId	Char(7)	Kode Kecamatan	1001001
2.	kecamatanNama	Varchar(150)	Nama Kecamatan	Cibeunying Kidul
3.	kotaId	Char(4)	Kode Kota	1001
4.	kodePos	Varchar(7)	Kode Pos Kecamatan	40192
5.	kecamatanStatus	Enum('y','n')	Status Aktif Kecamatan	y

m. kelurahan

Tabel 4.13 kelurahan

No.	Atribut	Type	Keterangan	Contoh
1.	kelurahanId	Char(10)	Kode Kelurahan	1001001001
2.	kelurahanNama	Varchar(150)	Nama Kecamatan	Pasir Layung
3.	kecamatanId	Char(7)	Kode Kecamatan	1001001
4.	kelurahanStatus	Enum('y','n')	Status Aktif Kelurahan	y

n. salesGrade

Tabel 4.14 salesGrade

No.	Atribut	Type	Keterangan	Contoh
1.	salesGradeId	Char(6)	Kode Grade Sales	SLG001
2.	salesGradeNama	Varchar(100)	Nama Grade Sales	Sales Junior
3.	salesGradeTarget	Integer(3)	Target Sales	30
4.	salesGradeStatus	Enum('y','n')	Status Aktif Grade Sales	y

o. sales

Tabel 4.15 sales

No.	Atribut	Type	Keterangan	Contoh
1.	salesId	Char(17)	Kode Sales	JG0001SL120707002
2.	karyawanId	Char(13)	Kode Karyawan	KR20120707001
3.	salesParentId	Char(17)	Kode Sales	JG0001SL120707001
4.	salesGradeId	Char(6)	Kode Grade Sales	SLG001
5.	cabangId	Char(6)	Kode Cabang	JG0001
6.	salesStatus	Enum('y','n')	Status Aktif Grade Sales	y

p. pekerjaan

Tabel 4.16 pekerjaan

No.	Atribut	Type	Keterangan	Contoh
1.	pekerjaanId	Char(6)	Kode Pekerjaan	KER001
2.	pekerjaanNama	Varchar(100)	Nama Pekerjaan	Pegawai Swasta
3.	pekerjaanStatus	Enum('y','n')	Status Aktif Pekerjaan	y

q. pendidikan

Tabel 4.17 pendidikan

No.	Atribut	Type	Keterangan	Contoh
1.	pendidikanId	Char(7)	Kode Pendidikan	PEND001
2.	pendidikanTingkat	Varchar(100)	Tingkat Pendidikan	Strata 2

r. partsCategory

Tabel 4.18 partsCategory

No.	Atribut	Type	Keterangan	Contoh
1.	partsCategoryId	Char(6)	Kode Kategori Parts	CP0001
2.	partsCategoryNama	Varchar(50)	Nama Kategori Parts	Frame
3.	partsCategoryStatus	Enum('y','n')	Status Aktif Kategori Parts	y

s. partsType

Tabel 4.19 partsType

No.	Atribut	Type	Keterangan	Contoh
1.	partsTypeId	Char(6)	Kode Tipe Parts	TP0001
2.	partsTypeNama	Varchar(50)	Nama Tipe Parts	Spare Parts
3.	partsTypeStatus	Enum('y','n')	Status Aktif Tipe Parts	y

t. parts

Tabel 4.20 parts

No.	Atribut	Type	Keterangan	Contoh
1.	partsId	Char(16)	Kode Parts	CP0001TP00010001
2.	partsNama	Varchar(100)	Nama Parts	SPRING, TENSION (3B4G) ATV
3.	partsCategoryId	Char(6)	Kode Kategori Parts	CP0001
4.	partsTypeId	Char(6)	Kode Tipe Parts	TP0001
5.	cabangId	Char(6)	Kode Cabang	JG0001
6.	partsStatus	Enum('y','n')	Status Aktif Parts	y

u. partsStock

Tabel 4.21 partsStock

No.	Atribut	Type	Keterangan	Contoh
1.	partsStockId	Char(23)	Kode Stock Parts	JG0001ST201207000000001
2.	partsId	Char(16)	Kode Parts	CP0001TP00010001
3.	price	Decimal (15,2)	Harga Jual Spare Parts	7.000,00
4.	cabangId	Char(6)	Kode Cabang	JG0001
5.	stock	Integer(5)	Jumalh Stock Parts	100
6.	stockStatus	Enum('y','n')	Status Aktif Stock Parts	y

v. partsStockDetail

Tabel 4.22 partsStockDetail

No.	Atribut	Type	Keterangan	Contoh
1.	partsStockId	Char(23)	Kode Stock Parts	JG0001ST201207000000001
2.	partsStockDetailId	Integer(5)	Kode Detail Stock Parts	100
3.	partsStockDetailDate	date	Tanggal Stock Detail	2012-07-07
4.	partsStockDetailStatus	Enum('y','n')	Status Aktif Stock Detail Parts	y

w. serviceType

Tabel 4.23 serviceType

No.	Atribut	Type	Keterangan	Contoh
1.	serviceTypeId	Char(8)	Kode Tipe Service	SERTY001
2.	serviceTypeNama	Varchar(75)	Nama Tipe Service	Service Berat
3.	serviceTypeStatus	Enum('y','n')	Status Aktif Tipe Service	y

x. service

Tabel 4.24 service

No.	Atribut	Type	Keterangan	Contoh
1.	serviceId	Char(7)	Kode Service	SR00001
2.	serviceNama	Varchar(100)	Nama Service	Service Besar Matic
3.	serviceTypeId	Char(8)	Kode Tipe Service	SERTY001
4.	servicePrice	Decimal(15,2)	Harga Jasa Service	75.000,00
4.	cabangId	Char(6)	Kode Cabang	JG0001
5.	serviceStatus	Enum('y','n')	Status Aktif Service	y

y. mekanikGrade

Tabel 4.25 mekanikGrade

No.	Atribut	Type	Keterangan	Contoh
1.	mekanikGradeId	Char(6)	Kode Grade Mekanik	MCG001
2.	mekanikGradeNama	Varchar	Grade Mekanik	Mekanik Junior
3.	gradeServiceTarget	Decimal(15,2)	Target Service Mekanik	5.000.000,00
4.	gradePartsTarget	Decimal(15,2)	Target Parts Mekanik	9.000.000,00
5.	mekanikGradeStatus	Enum('y','n')	Status Aktif Grade Mekanik	y

z. mekanik

Tabel 4.26 mekanik

No.	Atribut	Type	Keterangan	Contoh
1.	mekanikId	Char(17)	Kode Mekanik	JG0001MC120707002
2.	karyawanId	Char(13)	Kode Karyawan	KR20120707007
3.	mekanikParentId	Char(17)	Kode Sales	JG0001MC120707001
4.	mekanikGradeId	Char(6)	Kode Grade Mekanik	MCG001
5.	cabangId	Char(6)	Kode Cabang	JG0001
6.	mekanikStatus	Enum('y','n')	Status Aktif Mekanik	y

aa. person

Tabel 4.27 person

No.	Atribut	Type	Keterangan	Contoh
1.	personId	Char(13)	Kode Person	L198502260001
2.	personNama	Varchar(150)	Nama Person	Asep Muhammad Indra Purnama
3.	personNamaPanggilan	Varchar(50)	Nama Panggilan	A M I P
4.	jenisKelamin	Enum('L','P')	Jenis Kelamin	L
5.	lahirTempat	Varchar(100)	Tempat Lahir	Bandung
6.	lahirTgl	Date	Tanggal Lahir	1985-02-26
7.	maritalStatus	Enum('lajang', 'menikah','duda', 'janda')	Status Pernikahan	lajang
8.	agama	Enum('ISLAM', 'KKALOTIK', 'KPROTESTAN', 'HINDU', 'BUDHA', 'KEPERCAYA-AN')	Agama Person	ISLAM
9.	sukuBangsaId	Char(7)	Kode Suku Bangsa Person	SB00001
10.	golonganDarah	Enum('A','B', 'A','AB","O')	Golongan Darah	O
11.	tinggiBadan	Integer(3)	Tinggi Badan	167
12.	beratBadan	Integer(3)	Berat Badan	50
13.	alamatSkrg	Varchar(150)	Alamat Person Sekarang	Jalan Pasir Malaka No. 26
14.	kelurahanSkrg	Char(10)	Kelurahan Person	1001001001

Tabel 4.27 person (lanjutan)

No.	Atribut	Type	Keterangan	Contoh
15.	kecamatanSkrgId	Char(7)	Kode Kecamatan	1001001
16.	kotaSkrgId	Char(4)	Kode Kota	1001
17.	propinsiSkrgId	Char(4)	Kode Propinsi	1000
18.	alamatTetap	Varchar(150)	Alamat Tetap Person	Jalan Pasir Malaka No. 26
19.	kelurahanTetapId	Char(10)	Kelurahan Tetap Person	1001001001
20.	kecamatanTetapId	Char(7)	Kode Kecamatan Tetap	1001001
21.	kotaTetapId	Char(4)	Kode Kota Tetap	1001
22.	propinsiTetapId	Char(4)	Kode Propinsi Tetap	1000
23.	email	Varchar(150)	Alamat Email	amip.khan@gmail.com
24.	telpNo	Varchar(13)	Nomor Telepon	0227806029
25.	hpNo	Varchar(15)	Nomor HP	081809419562
26.	kontakNama	Varchar(150)	Nama Kontak	Alessandro Del Piero
27.	hubunganKontak	Enum('Ayah', 'Ibu','Suami', Istri','Saudara', 'Anak','Lain')	Hubungan Person dengan Kontak	Saudara
28.	telpKontakNo	Varchar(13)	Nomor Telepon Kontak	0227809934
29.	hpKontakNo	Varchar(15)	Nomor HP Kontak	02292960072

bb. sukuBangsa

Tabel 4.28 sukuBangsa

No.	Atribut	Type	Keterangan	Contoh
1.	sukuBangsaId	Char(7)	Kode Suku Bangsa	SB00001
2.	sukuBangsaNama	Varchar(100)	Suku Bangsa	Sunda

cc. karyawan

Tabel 4.29 karyawan

No.	Atribut	Type	Keterangan	Contoh
1.	karyawanId	Char(13)	Kode Karyawan	KR20120707009
2.	personId	Char(13)	Kode Person	L198602260009
3.	nik	Char(9)	Nomor Induk Karyawan	201103099
4.	awal	Date	Tanggal Awal Kerja	2011-01-01
5.	akhir	Date	Tanggal Akhir Kerja	2012-09-30
6.	jabatanId	Char(6)	Kode Jabatan	JB0101
7.	cabangId	Char(6)	Kode Cabang	JG0001
8.	karyawanStatus	Enum('y','n')	Status Aktif Karyawan	y

dd. karyawanTypeHistory

Tabel 4.30 karyawanTypeHistory

No.	Atribut	Type	Keterangan	Contoh
1.	karyawanTypeHistoryId	Char(7)	Kode Tipe History Karyawan	KTH0001
2.	karyawanTypeHistoryNama	Varchar(100)	Tipe History Karyawan	Mutasi
3.	karyawanTypeHistoryStatus	Enum('y','n')	Status Aktif Tipe History Karyawan	y

ee. karyawanHistory

Tabel 4.31 karyawanHistory

No.	Atribut	Type	Keterangan	Contoh
1.	karyawanHistoryId	Char(6)	Kode History Karyawan	KH0001
2.	karyawanId	Char(13)	Kode Karyawan	KR20120707009
3.	karyawanTypeHistoryId	Char(7)	Kode Tipe History Karyawan	KTH0001
4.	keterangan	Varchar(150)	Keterangan History Karyawan	Mutasi ke Divisi IT

ff. jabatan

Tabel 4.32 jabatan

No.	Atribut	Type	Keterangan	Contoh
1.	jabatanId	Char(6)	Kode Jabatan	JB0101
2.	jabatanNama	Varchar(100)	Nama Jabatan	IT Manager
3.	jabatanStatus	Enum('y','n')	Status Aktif Jabatan	y

gg. trainingType

Tabel 4.33 trainingType

No.	Atribut	Type	Keterangan	Contoh
1.	trainingTypeId	Char(7)	Kode Tipe Training	TRT0001
2.	trainingTypeNama	Varchar(100)	Nama Tipe Training	My First Training
3.	trainingTypeStatus	Enum('y','n')	Status Aktif Tipe Training	y

hh. training

Tabel 4.34 training

No.	Atribut	Type	Keterangan	Contoh
1.	trainingId	Char(12)	Kode Training	TR2012070001
2.	trainingTypeId	Char(7)	Kode Tipe Training	TRT0001
3.	karyawanId	Char(13)	Kode Karyawan	KR20120707009
4.	trainingKeterangan	Varchar(100)	Keterangan Training	Training Karyawan Baru
5.	trainingStatus	Enum('y','n')	Status Aktif Training	y

ii. application

Tabel 4.35 application

No.	Atribut	Type	Keterangan	Contoh
1.	applicationId	Char(5)	Kode Aplikasi	AP001
2.	applicationName	Varchar(50)	Nama Aplikasi	01 Akses
3.	applicationFullName	Varchar(100)	Nama Lengkap Aplikasi	Menu Untuk Mengolah Akses Aplikasi
4.	baseFolder	Varchar(100)	Base Folder Aplikasi	mod=user
5.	applicationStatus	Enum('y','n')	Status Aktif Aplikasi	y
6.	applicationDateModified	Datetime	Waktu Terakhir Modifikasi	2012-07-07 12:51:01

jj. groupMenu

Tabel 4.36 groupMenu

No.	Atribut	Type	Keterangan	Contoh
1.	userGroupId	Char(6)	Kode Group User	UG0001
2.	menuId	Char(8)	Kode Menu	AP001002

kk. menu

Tabel 4.37 menu

No.	Atribut	Type	Keterangan	Contoh
1.	menuId	Char(8)	Kode Menu	AP001002
2.	level	Enum('root', 'leaf','function')	Tingkatan Menu	leaf
3.	url	Varchar(255)	Nama Link Menu	entryUser
4.	title	Varchar(100)	Judul Link	Entry User
5.	target	Varchar(50)	Target Aplikasi	right
6.	open	Enum('true','false')	Status Open Aplikasi	false
7.	menuParentId	Char(8)	Kode Menu Parent	AP001001
8.	menuIndex	Integer(2)	Index Menu	0
9.	menuStatus	Enum('y','n')	Status Aktif Menu	y
10.	applicationId	Char(5)	Kode Aplikasi	AP001
11.	menuKeterangan	Varchar(100)	Keterangan Menu	Menu Untuk Entry User Baru

ll. groupUser

Tabel 4.38 groupUser

No.	Atribut	Type	Keterangan	Contoh
1.	userGroupId	Char(6)	Kode Group User	UG0001
2.	userGroupName	Varchar(100)	Nama User Group	Admin
3.	userGroupStatus	Enum('y','n')	Status Aktif Group User	y
4.	userGroupModified	Datetime	Waktu Terakhir Modifikasi	2012-07-07 12:51:01

mm. user

Tabel 4.39 user

No.	Atribut	Type	Keterangan	Contoh
1.	userId	Char(12)	Kode User	US2011070001
2.	userName	Varchar(50)	User Name	user1
3.	karyawanId	Char(13)	Kode Karyawan	KR20120707009
4.	password	Varchar(255)	Password User	233546464648a
5.	userGroupId	Char(6)	Kode Group User	UG0001
6.	cabangId	Char(6)	Kode Cabang	JG0001
7.	tglInsert	Datetime	Tanggal Input User	2012-07-07 10:09:01
8.	tglLastLogin	Datetime	Tanggal Terakhir User Login	2012-07-08 08:31:01
9.	userInsert	Char(12)	Kode User	US2010070099
10.	userStatus	Enum('y','n')	Status Aktif User	y

nn. absensi

Tabel 4.40 absensi

No.	Atribut	Type	Keterangan	Contoh
1.	absensiId	Char(18)	Kode Absensi Karyawan	JG0001201207070001
2.	karyawanId	Char(13)	Kode Karyawan	KR20120707009
3.	absensiTipe	Enum ('N','I','S', 'A','L', 'LN', 'CT')	Tipe Absensi Karyawan	N = Normal I = Ijin S = Sakit A = Alpa L = Libur LN = Libur Nasional CT = Cuti
4.	absensiPeriode	Date	Tanggal Periode Absen	2012-07-01
5.	absensiTgl	Date	Tanggal Absen	2012-07-07
6.	jamKerjaMasuk	Datetime	Jam Masuk Standar	2012-07-07 08:30:00
7.	masuk	Datetime	Jam Masuk Karyawan	2012-07-07 08:15:31

Tabel 4.40 absensi (lanjutan)

No.	Atribut	Type	Keterangan	Contoh
8.	cabangIdMasuk	Char(6)	Kode Cabang Masuk	JG0001
9.	jamKerjaKeluar	Datetime	Jam Keluar Standar	2012-07-07 17:30:00
10.	keluar	Datetime	Jam Keluar	
11.	cabangIdKeluar	Char(6)	Kode Cabang Keluar	JG0001
12.	lemburPoint	Integer(2)	Point Lembur	10
13.	lemburKeterangan	Varchar(150)	Alasan Lembur	Setting aplikasi Absensi Cabang
14.	terlambatKeterangan	Varchar(150)	Alasan Keterlambatan	Ijin ke dokter terlebih dahulu

oo. finger

Tabel 4.41 finger

No.	Atribut	Type	Keterangan	Contoh
1.	<u>fingerId</u>	Char(11)	Kode Finger Karyawan	F2011030990
2.	karyawanId	Char(13)	Kode Karyawan	KR20120707009
3.	fingerIndex	Char(1)	Nomor Index Finger Karyawan	0
4.	fingerData	Text	Data Finger Karyawan	567ffg88f8f8f8f8f88ff Fjfjfjfkif890000ijd7f9

pp. payroll

Tabel 4.42 payroll

No.	Atribut	Type	Keterangan	Contoh
1.	payrollId	Char(6)	Kode Payroll	PY0001
2.	jabatanId	Char(6)	Kode Jabatan	JB0101
3.	payrollPokok	Decimal(15,2)	Gaji Pokok	5.000.000,00
4.	payrollTransport	Decimal(15,2)	Tunjangan Transport	1.500.000,00
5.	payrollMakan	Decimal(15,2)	Tunjangan Makan	1.000.000,00
6.	payrollStatus	Enum('y','n')	Status Aktif Payroll	y

qq. customerComplainType

Tabel 4.43 customerComplainType

No.	Atribut	Type	Keterangan	Contoh
1.	complainTypeId	Char(7)	Kode Tipe Complain Customer	CCT0001
2.	complainTypeNama	Varchar(100)	Tipe Complain Customer	Complain Transaksi Unit Motor
3.	complainTypeStatus	Enum('y','n')	Status Aktif Tipe Complain Customer	y

rr. customerComplain

Tabel 4.44 customerComplain

No.	Atribut	Type	Keterangan	Contoh
1.	complainId	Char(19)	Kode Complain Customer	JG0001CC20120707001
2.	uraian	Text	Uraian Complain Customer	Pengiriman Unit Terlambat dan Motor SedikitPudar Warna Stripingnya
3.	customerId	Char(17)	Kode Customer	JG0001CU120707001
4.	cabangId	Char(6)	Kode Cabang Complain	JG0001
5.	complainTypeId	Char(7)	Kode Tipe Complain Customer	CCT0001

Tabel 4.44 customerComplain (lanjutan)

No.	Atribut	Type	Keterangan	Contoh
6.	complainStatus	Enum('y','n')	Status Aktif Complain Customer	y

ss. marketingToolsType

Tabel 4.45 marketingToolsType

No.	Atribut	Type	Keterangan	Contoh
1.	marketingToolsTypeId	Char(7)	Kode Tipe Marketing Tools	MTT0001
2.	marketingToolsTypeNama	Varchar(150)	Tipe Marketing Tools	Marketing Tools Event External
3.	marketingToolsTypeStatus	Enum('y','n')	Status Aktif Tipe Marketing Tools	y

tt. marketingTools

Tabel 4.46 marketingTools

No.	Atribut	Type	Keterangan	Contoh
1.	marketingToolsId	Char(7)	Kode Marketing Tools	MT00001
2.	marketingToolsNama	Varchar(100)	Marketing Tools	Umbul-Umbul
3.	marketingToolsTypeId	Char(7)	Kode Tipe Marketing Tools	MTT0001
4.	jumlah	Integer(4)	Jumlah Marketing Tools	1000
5.	marketingToolsStatus	Enum('y','n')	Status Aktif Marketing Tools	y

uu. merchant

Tabel 4.47 merchant

No.	Atribut	Type	Keterangan	Contoh
1.	merchantId	Char(11)	Kode Merchant	JG0001MC001
2.	merchantNama	Varchar(100)	Nama Merchant	Eqlips
3.	cabangId	Char(6)	Kode Cabang Merchant	JG0001
4.	merchantStatus	Enum('y','n')	Status Aktif Merchant	y

vv. ajuan

Tabel 4.48 ajuan

No.	Atribut	Type	Keterangan	Contoh
1.	ajuanId	Char(18)	Kode Ajuan	JG0001AJ2012070001
2.	karyawanId	Char(13)	Kode Karyawan	KR20120707009
3.	cabangId	Char(6)	Kode Cabang Ajuan	JG0001
4.	infoSingkat	Varchar(150)	Informasi Singkat Ajuan	Pengajuan Switch
5.	uraian	Text	Informasi Lengkap Ajuan	Pengajuan 5 buah Switch Untuk Cabang BKR
6.	anggaran	Decimal(15,2)	Anggaran Ajuan	1.5000.000,00
7.	tujuanId	Char(13)	Kode Tujuan	KR20000107009
8.	tujuanCatatan	Text	Catatan dari Tujuan	Efektifkan penggunaan switch
9.	tujuanTanggal	Datetime	Tanggal Approvisasi Tujuan	2012-07-07 14:09:08
10.	tujuanApprovisasi	Enum('y','n')	Approvisasi oleh Tujuan	y
11.	picId	Char(13)	Kode PIC	KR20120707011
12.	assetId	Char(9)	Kode Asset	AST000001
13.	assetTypeId	Char(8)	Kode Tipe Asset	ASTT0001

Tabel 4.48 ajuan (lanjutan 1)

No.	Atribut	Type	Keterangan	Contoh
14.	pemeriksa1	Char(13)	Kode Pemeriksa 1	KR20010707011
15.	pemeriksa1Tgl	Datetime	Tanggal Pemeriksa 1	2012-07-07 15:09:08
16.	pemeriksa1Catatan	Text	Catatan dari Pemeriksa 1	Efektifkan penggunaan switch
17.	pemeriksa1Approvisasi	Enum('y','n')	Approvisasi oleh Pemeriksa 1	y
18.	pemeriksa2	Char(13)	Kode Pemeriksa 2	KR20010707014
19.	pemeriksa2Tgl	Datetime	Tanggal Pemeriksa 2	2012-07-07 15:11:04
20.	pemeriksa2Catatan	Text	Catatan dari Pemeriksa 2	Efektifkan penggunaan switch
21.	pemeriksa2Approvisasi	Enum('y','n')	Approvisasi oleh Pemeriksa 2	y
22.	pemeriksa3	Char(13)	Kode Pemeriksa 3	KR20010707016
23.	pemeriksa3Tgl	Datetime	Tanggal Pemeriksa 3	2012-07-07 15:13:04
24.	pemeriksa3Catatan	Text	Catatan dari Pemeriksa 3	Efektifkan penggunaan switch
25.	pemeriksa3Approvisasi	Enum('y','n')	Approvisasi oleh Pemeriksa 4	y
26.	pemeriksa4	Char(13)	Kode Pemeriksa 4	KR20010707017
27.	pemeriksa4Tgl	Datetime	Tanggal Pemeriksa 4	2012-07-07 15:14:04
28.	pemeriksa4Catatan	Text	Catatan dari Pemeriksa 4	Efektifkan penggunaan switch
29.	pemeriksa4Approvisasi	Enum('y','n')	Approvisasi oleh Pemeriksa 4	y
30.	pemeriksa5	Char(13)	Kode Pemeriksa 5	KR20010707018
31.	pemeriksa5Tgl	Datetime	Tanggal Pemeriksa 5	2012-07-07 15:16:04
32.	pemeriksa5Catatan	Text	Catatan dari Pemeriksa 5	Efektifkan penggunaan switch

Tabel 4.48 ajuan (lanjutan 2)

No.	Atribut	Type	Keterangan	Contoh
33.	pemeriksa5Approvisasi	Enum('y','n')	Approvisasi oleh Pemeriksa 5	y
34.	pemeriksa6	Char(13)	Kode Pemeriksa 6	KR20010707019
35.	pemeriksa6Tgl	Datetime	Tanggal Pemeriksa 6	2012-07-07 15:17:04
36.	pemeriksa6Catatan	Text	Catatan dari Pemeriksa 6	Efektifkan penggunaan switch
37.	pemeriksa6Approvisasi	Enum('y','n')	Approvisasi oleh Pemeriksa 6	y
38.	pemeriksa7	Char(13)	Kode Pemeriksa 7	KR20010707020
39.	pemeriksa7Tgl	Datetime	Tanggal Pemeriksa 7	2012-07-07 15:18:04
40.	pemeriksa7Catatan	Text	Catatan dari Pemeriksa 7	Efektifkan penggunaan switch
41.	pemeriksa7Approvisasi	Enum('y','n')	Approvisasi oleh Pemeriksa 7	y
42.	pemeriksa8	Char(13)	Kode Pemeriksa 8	KR20010707021
43.	pemeriksa8Tgl	Datetime	Tanggal Pemeriksa 8	2012-07-07 15:19:04
44.	pemeriksa8Catatan	Text	Catatan dari Pemeriksa 8	Efektifkan penggunaan switch
45.	pemeriksa8Approvisasi	Enum('y','n')	Approvisasi oleh Pemeriksa 8	y
46.	otorisasiIndex	Char(1)	Nomor Index Otorisasi	0
47.	otorisasiId	Char(13)	Kode PIC	KR20120707011
48.	tokenNo	Char(16)	Nomor Token Ajuan	201109UUUGPMGUCH
49.	ajuanStatus	Enum('y','n')	Status Aktif Ajuan	y

ww. assetType

Tabel 4.49 assetType

No.	Atribut	Type	Keterangan	Contoh
1.	assetTypeId	Char(8)	Kode Tipe Asset	ASTT0001
2.	assetTypeNama	Varchar(100)	Nama Tipe Asset	Asset IT
3.	assetTypeStatus	Enum('y','n')	Status Tipe Asset	y

xx. asset

Tabel 4.50 asset

No.	Atribut	Type	Keterangan	Contoh
1.	assetId	Char(9)	Kode Asset	AST000001
2.	assetNama	Varchar(100)	Nama Asset	Switch
3.	assetTypeId	Char(8)	Kode Tipe Asset	ASTT0001
4.	assetStatus	Enum('y','n')	Status Asset	y

yy. complainUserType

Tabel 4.51 complainUserType

No.	Atribut	Type	Keterangan	Contoh
1.	complainUserTypeId	Char(6)	Kode Tipe Complain User	CU0001
2.	complainUserTypeNama	Varchar(100)	Tipe Complain	Aplikasi
3.	complainUserStatus	Enum('y','n')	Status Tipe Complain	y

zz. complainUser

Tabel 4.52 complainUser

No.	Atribut	Type	Keterangan	Contoh
1.	complainId	Char(18)	Kode Complain User	JG0001CU2012070001
2.	karyawanId	Char(13)	Kode Karyawan	KR20120707009

Tabel 4.52 complainUser (lanjutan)

No.	Atribut	Type	Keterangan	Contoh
3.	cabangId	Char(6)	Kode Cabang Ajuan	JG0001
4.	complainUserTypeId	Char(6)	Kode Tipe Complain User	CU0001
5.	uraian	Text	Uraian Complain User	Kami tidak bisa akses ke Aplikasi Operasional
6.	complainTgl	Datetime	Tanggal Complain	2012-07-07 10:01:09
7.	complainTglApprovisasiUser	Datetime	Tanggal Complain	2012-07-07 15:01:09
8.	complainStatus	Enum('y' ,'n')	Status Complain	y
9.	complainPic	Char(13)	Kode Karyawan	KR20110707009
10.	complainTglEksekusi	Datetime	Tanggal Eksekusi Complain	2012-07-07 14:01:09
11.	complainTglApprovisasiPic	Datetime	Tanggal Approvisasi Complain by PIC	2012-07-07 14:07:09
12.	batalUser	Text	Keterangan Pembatalan Complain	Complain tidak jadi karena aplikasi sudah normal kembali
13.	batalPic	Char(13)	Kode Karyawan	KR20110707009
14.	batalTgl	Datetime	Tanggal Batal Complain	2012-07-07 14:07:09
15.	tokenNo	Char(16)	Nomor Token Complain	201109UUUGPMGUCH
16.	complainUserStatus	Enum('y' ,'n')	Status Complain User	y

4.2.3 Kebutuhan Perubahan Aplikasi Existing

Dalam perancangan Arsitektur Manajemen Master Data terdapat kebutuhan perubahan aplikasi existing guna mendukung integrasi kedalam Sistem Master Data Management, kebutuhan perubahan tersebut antara lain :

a. Integrasi dan harmonisasi *business process* dari masing-masing aplikasi.
b. Penyesuaian *business logic* aplikasi terkait dengan aliran program, validasi, pemprosesan data, security dan lain-lain.
c. Pemodelan *application adapter* masing-masing aplikasi terutama untuk *design application adapter* yang bersifat khusus dan spesifik terkait *business logic* masing-masing aplikasi.