**PEMBANGUNAN PERANGKAT LUNAK DATA *WAREHOUSE* DENGAN KONSEP *FUZZY DIMENSION* DI PT. SPACEMAN CLOTHING INDONESIA**

**Disusun untuk Menyelesaikan Mata Kuliah Skripsi**

**Semester Genap Tahun Akademik 2014/2015**

**OLEH:**

Ricky Azhari Pratama

10111260



**Program Studi Teknik Informatika**

**Fakultas Teknik Dan Ilmu Komputer**

**Universitas Komputer Indonesia**

**2015**

# DAFTAR ISI

[DAFTAR ISI i](#_Toc420751510)

[DAFTAR GAMBAR i](#_Toc420751511)

[DAFTAR TABEL i](#_Toc420751512)

[BAB I PENDAHULUAN 1](#_Toc420751513)

[I.1 Latar Belakang Masalah 1](#_Toc420751514)

[I.2 Perumusan Masalah 1](#_Toc420751515)

[I.3 Maksud dan Tujuan 1](#_Toc420751516)

[I.4 Batasan Masalah 1](#_Toc420751517)

[I.5 Metode Penelitian 1](#_Toc420751518)

[I.5.1 Metode Pengumpulan Data 1](#_Toc420751519)

[I.5.2 Metode Pembangunan Perangkat Lunak 1](#_Toc420751520)

[I.5.3 Metode Pembangunan Data *Warehouse* 1](#_Toc420751521)

[I.5.4 Metode Pemodelan *Fuzzy* Data *Warehouse* 1](#_Toc420751522)

[I.6 Sistematika Penulisan 1](#_Toc420751523)

[BAB II TINJAUAN PUSTAKA 1](#_Toc420751524)

[II.1 Profil Perusahaan 1](#_Toc420751525)

[II.1.1 Visi dan Misi Perusahaan 1](#_Toc420751526)

[II.1.2 Struktur Organisasi 1](#_Toc420751527)

[II.1.3 Deskripsi Kerja 1](#_Toc420751528)

[II.2 Landasan Teori 1](#_Toc420751529)

[II.2.1 Database 1](#_Toc420751530)

[II.2.2 Skema Diagram 1](#_Toc420751531)

[II.2.3 Data *Warehouse* 1](#_Toc420751532)

[II.2.4 Skema Data *Warehouse* 1](#_Toc420751533)

[II.2.5 *ETL ( Extract, Transform, Loading)* 1](#_Toc420751534)

[II.2.6 *OLAP (On-Line Analytical Processing)* 1](#_Toc420751535)

[II.2.7 *Fuzzy* Data *Warehouse* 1](#_Toc420751536)

[II.2.8 Konsep *Fuzzy* Data *Warehouse* 1](#_Toc420751537)

[II.2.9 Meta Model *Fuzzy* Data *Warehouse* 1](#_Toc420751538)

[II.2.10 Tipe Konsep *Fuzzy* 1](#_Toc420751539)

[II.2.11 *Object Oriented Analysis and Design* (*OOAD*) 1](#_Toc420751540)

[BAB III ANALISIS DAN PERANCANGAN DATA WAREHOUSE 1](#_Toc420751541)

[III.1 Analisis Data *Warehouse* 1](#_Toc420751542)

[III.1.1 Analisis Masalah 1](#_Toc420751543)

[III.1.2 Analisis Sumber Data 1](#_Toc420751544)

[III.1.3 Analisis Kebutuhan Informasi Strategis 1](#_Toc420751545)

[III.1.4 Analisis Dimensi dan Fakta Bisnis 1](#_Toc420751546)

[III.1.5 Analisis Data *Staging* 1](#_Toc420751547)

[III.2 Analisis Konsep *Fuzzy* 1](#_Toc420751548)

[III.2.1 Elemen Pengklasifikasian 1](#_Toc420751549)

[III.2.2 Pemodelan *Fuzzy Data Warehouse* 1](#_Toc420751550)

[III.3 Analisis OLAP 1](#_Toc420751551)

[III.4 Analisis Kebutuhan 1](#_Toc420751552)

[III.4.1 Analisis Kebutuhan Non Fungsional 1](#_Toc420751553)

[III.4.2 Analisis Kebutuhan Fungsional 1](#_Toc420751554)

[III.5 Perancangan Antarmuka 1](#_Toc420751555)

[BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM 1](#_Toc420751556)

[IV.1 Implementasi Data *Warehouse* 1](#_Toc420751557)

[IV.1.1 Implementasi Lingkungan 1](#_Toc420751558)

[IV.1.2 Implementasi Data 1](#_Toc420751559)

[IV.1.3 Implementasi Antarmuka 1](#_Toc420751560)

[DAFTAR PUSTAKA 1](#_Toc420751561)

# DAFTAR GAMBAR

[Gambar I‑1 *Waterfall model* [3] 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751562)

[Gambar I‑2 *Kimball Life Cycle Data Track Diagram* [4] 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751563)

[Gambar I‑3 *Method for modeling a fuzzy data warehouse* 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751564)

[Gambar II‑1 Struktur organisasi Spaceman Clothing Indonesia 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751565)

[Gambar II‑2 Komponen arsitektur dalam *three major areas* 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751566)

[Gambar II‑3 meta *fuzzy* data *warehouse* [5] 1](#_Toc420751567)

[Gambar II‑4 *Open end fuzzy concept* [5] 1](#_Toc420751568)

[Gambar II‑5 *Limited fuzzy concept* [5] 1](#_Toc420751569)

[Gambar II‑6 *Adaptive fuzzy concept* [5] 1](#_Toc420751570)

[Gambar III‑1 Skema diagram *OLTP* 1](#_Toc420751571)

[Gambar III‑2 Skema dimensional fakta belanja produksi 1](#_Toc420751572)

[Gambar III‑3 Skema dimensional fakta pembayaran konsumen 1](#_Toc420751573)

[Gambar III‑4 Skema dimensional fakta kepopuleran produk 1](#_Toc420751574)

[Gambar III‑5 Skema dimensional fakta biaya produksi 1](#_Toc420751575)

[Gambar III‑6 Skema dimensional fakta pemesanan produk 1](#_Toc420751576)

[Gambar III‑7 skema dimensional fakta pengiriman produk 1](#_Toc420751577)

[Gambar III‑8 Skema fakta kepopuleran *design* produk 1](#_Toc420751578)

[Gambar III‑9 Skema fakta *size* produk 1](#_Toc420751579)

[Gambar III‑10 *Control flow data staging* 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751580)

[Gambar III‑11 *Tools*  pembuatan *control flow* ETL 1](#_Toc420751581)

[Gambar III‑12 Data *flow* dimensi produk 1](#_Toc420751582)

[Gambar III‑13 Data *flow* dimensi waktu 1](#_Toc420751583)

[Gambar III‑14 Data *flow* dimensi konsumen 1](#_Toc420751584)

[Gambar III‑15 Data *flow* dimensi pengiriman 1](#_Toc420751585)

[Gambar III‑16 Data *flow* dimensi staff produksi 1](#_Toc420751586)

[Gambar III‑17 Data *flow* dimensi *job* 1](#_Toc420751587)

[Gambar III‑18 Data *flow* dimensi kredit 1](#_Toc420751588)

[Gambar III‑19 *Data flow* dimensi *design* 1](#_Toc420751589)

[Gambar III‑20 *Data flow* dimensi *size* 1](#_Toc420751590)

[Gambar III‑21 Data *flow* fakta pemesanan 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751591)

[Gambar III‑22 *Data flow* fakta pengiriman produk 1](#_Toc420751592)

[Gambar III‑23 Data *flow* fakta belanja 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751593)

[Gambar III‑24 Data *flow* fakta biaya produksi 1](#_Toc420751594)

[Gambar III‑25 Data *flow* fakta kepopuleran produk 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751595)

[Gambar III‑26 Data *flow* fakta kepopuleran design produk 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751596)

[Gambar III‑27 *Data flow* fakta pembayaran 1](#_Toc420751597)

[Gambar III‑28 *Data flow* fakta *size* produk 1](#_Toc420751598)

[Gambar III‑29 Kurva pemetaan himpunan fuzzy jumlah produk 1](#_Toc420751599)

[Gambar III‑30 Kurva pemetaan himpunan *fuzzy* pembayaran konsumen 1](#_Toc420751600)

[Gambar III‑31 *Data flow fuzzy* konsep populer desain 1](#_Toc420751601)

[Gambar III‑32 *Data flow fuzzy* konsep populer produk 1](#_Toc420751602)

[Gambar III‑33 *Data flow fuzzy* konsep rentang pembayaran 1](file:///D:\MK\VIII\Skripsi\Masterpieces\Dokument\Laporan%201.1.docx#_Toc420751603)

[Gambar III‑34 *Use case diagram data warehouse* pada perusahaan Spaceman 1](#_Toc420751604)

[Gambar III‑35 *Activity diagram* proses *extract data* 1](#_Toc420751605)

[Gambar III‑36 *Activity diagram conditioning* data 1](#_Toc420751606)

[Gambar III‑37 *Activity diagram merging* data 1](#_Toc420751607)

[Gambar III‑38 *Activity diagram scrubbing data* 1](#_Toc420751608)

[Gambar III‑39 *Activiy diagram validating* data 1](#_Toc420751609)

[Gambar III‑40 *Activity diagram loading* data 1](#_Toc420751610)

[Gambar III‑41 *Activity diagram refresh* data 1](#_Toc420751611)

[Gambar III‑42 *Activity diagram pivoting OLAP* 1](#_Toc420751612)

[Gambar III‑43 *Activity diagram OLAP drill down* 1](#_Toc420751613)

[Gambar III‑44 *Activity diagram OLAP roll up* 1](#_Toc420751614)

[Gambar III‑45 *Activity diagram filter* data 1](#_Toc420751615)

[Gambar III‑46 *Activity diagram print report* 1](#_Toc420751616)

[Gambar III‑47 *Activity diagram show dimension-fact* 1](#_Toc420751617)

[Gambar III‑48 Diagram kelas *ETL* 1](#_Toc420751618)

[Gambar III‑49 Diagram kelas dimensi dan fakta 1](#_Toc420751619)

[Gambar III‑50 Diagram kelas pada fungsional *Creating Report* 1](#_Toc420751620)

[Gambar III‑51 Diagram kelas keseluruhan 1](#_Toc420751621)

[Gambar III‑52 *Sequence diagram use case extract* data 1](#_Toc420751622)

[Gambar III‑53 *Sequence diagram use case conditioning*  data 1](#_Toc420751623)

[Gambar III‑54 *Sequence diagram use case merging* data 1](#_Toc420751624)

[Gambar III‑55 *Sequence diagram use case scrubbing* data 1](#_Toc420751625)

[Gambar III‑56 *Sequence diagram use case validating* data 1](#_Toc420751626)

[Gambar III‑57 *Sequence diagram use case loading* data 1](#_Toc420751627)

[Gambar III‑58 *Sequence diagram use case refresh* data 1](#_Toc420751628)

[Gambar III‑59 *Sequence diagram use case pivoting OLAP* 1](#_Toc420751629)

[Gambar III‑60 *Sequence diagram use case pivoting OLAP* 1](#_Toc420751630)

[Gambar III‑61 *Sequence diagram use case OLAP roll up* 1](#_Toc420751631)

[Gambar III‑62 *Sequence diagram use case filter* data 1](#_Toc420751632)

[Gambar III‑63 *Sequence diagram use case print report* 1](#_Toc420751633)

[Gambar III‑64 *Sequence diagram use case menampilkan dimensi- fakta* 1](#_Toc420751634)

[Gambar III‑65 Perancangan antarmuka beranda (T001) 1](#_Toc420751635)

[Gambar III‑66 Perancangan antarmuka proses ETL (T002) 1](#_Toc420751636)

[Gambar III‑67 Perancangan antarmuka menu utama (T003) 1](#_Toc420751637)

[Gambar III‑68 Perancangan antarmuka dimensi (T004) 1](#_Toc420751638)

[Gambar III‑69 Perancangan antarmuka fakta (T005) 1](#_Toc420751639)

# DAFTAR TABEL

[Tabel III‑1 Struktur data tabel kota 1](#_Toc420751640)

[Tabel III‑2 Struktur data tabel konsumen 1](#_Toc420751641)

[Tabel III‑3 Struktur data tabel *order\_service* 1](#_Toc420751642)

[Tabel III‑4 Struktur data tabel pemesanan 1](#_Toc420751643)

[Tabel III‑5 Struktur data tabel perencanaan 1](#_Toc420751644)

[Tabel III‑6 Struktur data tabel pengiriman 1](#_Toc420751645)

[Tabel III‑7 Struktur data tabel metode\_pengiriman 1](#_Toc420751646)

[Tabel III‑8 Struktur data tabel pembayaran 1](#_Toc420751647)

[Tabel III‑9 Struktur data tabel *order\_size* 1](#_Toc420751648)

[Tabel III‑10 Struktur data tabel *size* 1](#_Toc420751649)

[Tabel III‑11 Struktur data tabel *design\_order* 1](#_Toc420751650)

[Tabel III‑12 Struktur data tabel *design* 1](#_Toc420751651)

[Tabel III‑13 Struktur data tabel produksi 1](#_Toc420751652)

[Tabel III‑14 Struktur data tabel job 1](#_Toc420751653)

[Tabel III‑15 Struktur data staff\_produksi 1](#_Toc420751654)

[Tabel III‑16 Struktur data tabel bagian 1](#_Toc420751655)

[Tabel III‑17 Struktur data tabel satuan 1](#_Toc420751656)

[Tabel III‑18 Struktur data tabel detail\_belanja 1](#_Toc420751657)

[Tabel III‑19 Struktur data tabel belanja 1](#_Toc420751658)

[Tabel III‑20 Struktur data tabel kredit\_belanja 1](#_Toc420751659)

[Tabel III‑21 Struktur data tabel produk 1](#_Toc420751660)

[Tabel III‑22 Dimensi bisnis Perusahaan Spaaceman 1](#_Toc420751661)

[Tabel III‑23 Struktur data dim\_design 1](#_Toc420751662)

[Tabel III‑24 Hirarki design 1](#_Toc420751663)

[Tabel III‑25 Sturktur data dim\_job 1](#_Toc420751664)

[Tabel III‑26 Hirarki job 1](#_Toc420751665)

[Tabel III‑27 Struktur data dim\_konsumen 1](#_Toc420751666)

[Tabel III‑28 Hirarki kota 1](#_Toc420751667)

[Tabel III‑29 Hirarki konsumen 1](#_Toc420751668)

[Tabel III‑30 Struktur data dim\_kredit 1](#_Toc420751669)

[Tabel III‑31 Hirarki kredit 1](#_Toc420751670)

[Tabel III‑32 Struktur data dim\_pengiriman 1](#_Toc420751671)

[Tabel III‑33 Hirarki pengiriman 1](#_Toc420751672)

[Tabel III‑34 Sturktur data dim\_produk 1](#_Toc420751673)

[Tabel III‑35 Hirarki produk 1](#_Toc420751674)

[Tabel III‑36 Struktur data dim\_size 1](#_Toc420751675)

[Tabel III‑37 Hirarki size 1](#_Toc420751676)

[Tabel III‑38 Struktur data dim\_staff\_produksi 1](#_Toc420751677)

[Tabel III‑39 Hirarki bagian 1](#_Toc420751678)

[Tabel III‑40 Hirarki Staff 1](#_Toc420751679)

[Tabel III‑41 Struktur data dim\_waktu 1](#_Toc420751680)

[Tabel III‑42 Hirarki waktu 1](#_Toc420751681)

[Tabel III‑43 Struktur data fact\_belanja 1](#_Toc420751682)

[Tabel III‑44 Struktur data fact\_pembayaran 1](#_Toc420751683)

[Tabel III‑45 Struktud data fact\_pop\_produk 1](#_Toc420751684)

[Tabel III‑46 Struktur data fact\_produksi 1](#_Toc420751685)

[Tabel III‑47 Sturktur data fact\_pemesaan 1](#_Toc420751686)

[Tabel III‑48 Sturktur data fact\_pengiriman\_produk 1](#_Toc420751687)

[Tabel III‑49 Struktur data pada fact\_pop\_design 1](#_Toc420751688)

[Tabel III‑50 Struktur data fact\_size\_produk 1](#_Toc420751689)

[Tabel III‑51 Pemesanan produk oleh konsumen 1](#_Toc420751690)

[Tabel III‑52 Kepopuleran produk berdasarkan pemetaan fungsi himpunan 1](#_Toc420751691)

[Tabel III‑53 Struktur data Tabel konsep *fuzzy* kepopuleran produk 1](#_Toc420751692)

[Tabel III‑54 Struktur data tabel konsep *fuzzy* kepopuleran *design* 1](#_Toc420751693)

[Tabel III‑55 Struktur data tabel konsep *fuzzy* pembayaran konsumen 1](#_Toc420751694)

[Tabel III‑56 Spesifikasi Kebutuhan Fungsional 1](#_Toc420751695)

[Tabel III‑57 Spesifikasi Kebutuhan Non-Fungsional 1](#_Toc420751696)

[Tabel III‑58 Spesifikasi perangkat keras yang dibutuhkan 1](#_Toc420751697)

[Tabel III‑59 Spesifikasi perangkat keras yang ada saat ini 1](#_Toc420751698)

[Tabel III‑60 Spesifikasi perangkat lunak yang dibutuhkan 1](#_Toc420751699)

[Tabel III‑61 Spesifikasi perangkat lunak yang ada saat ini 1](#_Toc420751700)

[Tabel III‑62 Spesifikasi kebutuhan perangkat prikir 1](#_Toc420751701)

[Tabel III‑63 Spesifikasi perangkat pikir yang ada 1](#_Toc420751702)

[Tabel III‑64 *Use case scenario extract* data 1](#_Toc420751703)

[Tabel III‑65 *Use case scenario conditioning* data 1](#_Toc420751704)

[Tabel III‑66 *Use case scenario merging* data 1](#_Toc420751705)

[Tabel III‑67 *Use case scenario scrubbing* data 1](#_Toc420751706)

[Tabel III‑68 *Use case scenario validating* data 1](#_Toc420751707)

[Tabel III‑69 *Use case scenario loading* data 1](#_Toc420751708)

[Tabel III‑70 *use case scenario refresh* data 1](#_Toc420751709)

[Tabel III‑71 *use case scenario pivoting OLAP* 1](#_Toc420751710)

[Tabel III‑72 *use case scenario OLAP drill down* 1](#_Toc420751711)

[Tabel III‑73 *use case scenario OLAP roll up* 1](#_Toc420751712)

[Tabel III‑74 *use case scenario filter* data 1](#_Toc420751713)

[Tabel III‑75 *use case scenario print report* 1](#_Toc420751714)

[Tabel III‑76 *Use case scenario*  *show dimension-fact* 1](#_Toc420751715)

[Tabel III‑77 Jenis kelas yang digunakan di dalam sistem 1](#_Toc420751716)

[Tabel III‑78 Tanggung jawab kelas 1](#_Toc420751717)

[Tabel IV‑1 Implementasi antarmuka 1](#_Toc420751718)

# PENDAHULUAN

## Latar Belakang Masalah

Spaceman Clothing Indonesia merupakan salah satu perusahaan yang berkembang di Bandung yang bergerak di bidang jasa pengadaan barang hasil olahan konveksi, dimana produk yang dihasilkannya berupa pakaian seperti kemeja, kaos, jaket, dan berbagai seragam instansi ataupun seragam lainnya yang beralamat di jalan Cigadung Raya Timur No. 5 Kecamatan Cimeunyan Bandung. Perusahaan ini bertujuan untuk memberikan pelayanan kepada konsumen dengan memproduksi produk yang dipesan konsumen dengan baik dan terjaga kualitasnya. Konsumen yang telah menggunakan jasa perusahaan ini cukup beragam dan tersebar di wilayah Indonesia baik itu distro clothing, lembaga instansi, maupun sekolah. Perusahan ini telah berdiri selama enam tahun dan memiliki tiga bangunan utama, yaitu bangunan administrasi dan perencanaan, bangunan produksi dan pengendalian kualitas produk, dan bangunan desain produksi.

Sistem yang telah berjalan di dalam perusahaan ini dapat digolongkan sudah menggunakan teknologi dengan menerapkan penggunaan komputer dalam pendataan secara terpisah di setiap bagian dalam perusahaan ini dengan penginputan data secara manual. Kendala proses sistem yang berjalan di temukan seperti desain data yang kurang efektif untuk di analisis, Sulitnya menganalisis laporan dikarenakan laporan yang ada tidak dapat menampilkan informasi dari berbagai aspek penting yang disebut multidimensional yang diperlukan, seperti menganalisa laporan pembayaran konsumen yang memesan produk dalam periode waktu tertentu, pemesanan produk dalam periode waktu tertentu, ataupun biaya produksi dalam periode waktu tertentu. Kendala juga terjadi ketika informasi yang dihasilkan tidak dapat menyampaikan informasi yang semestinya. Seperti pada kendala pelunasan pembayaran produk yang dipesan konsumen, bila suatu konsumen mengalami telat dalam pelunasan maka akan menimbulkan rentang pembayaran terhadap tanggal semestinya yang akan digunakan dalam pembuatan laporan. Dalam data yang dihasilkan pada periode tertentu, rentang pembayaran yang seharusnya menginformasikan suatu fakta akan menjadi ambigu.

Sebagai solusi dari kendala dari permasalahan yang ada, maka perlu dibangunnya perangkat lunak data *warehouse* untuk memudahkan pimpinan dalam menganalisa data untuk mendapatkan informasi mengenai kepopuleran produk yang dipesan konsumen maupun pembayaran konsumen dalam pemesanan barang yang telah di lakukan oleh konsumen. Solusi kendala juga membutuhkan suatu metode agar mendapatkan informasi yang optimal. Dalam penelitian ini metode yang diterapkan dalam pembangunan data *warehouse* adalah menggunakan logika fuzzy. Penerapan logika fuzzy dalam penelitian ini dapat membantu pimpinan lebih mudah dalam mengartikan suatu informasi dan lebih intuitif dalam suatu penganalisisan [1].

## Perumusan Masalah

Berdasarkan latar belakang masalah maka dirumuskan masalah yaitu bagaimana membangun perangkat lunak *data warehouse* dengan menggunakan konsep *fuzzy dimension*.

## Maksud dan Tujuan

Adapun maksud yang ingin dicapai dalam menyelesaikan permasalahan adalah dengan membangun perangkat lunak data *warehouse* yang dapat mengelola data dari berbagai sumber secara valid dan akurat.

Adapun tujuan untuk dicapai dalam penyelesaian masalah yang telah dipaparkan adalah:

1. Membantu pimpinan perusahan dalam menganalisis suatu informasi dengan mengoptimalkan desain data dalam perusahan seperti :
   1. Kepopuleran produk yang dipesan oleh konsumen.
   2. Pemesanan produk yang dilakukan oleh konsumen
   3. Pembayaran terhadap pesanan yang dilakukan oleh konsumen.
   4. Pengiriman produk yang digunakan untuk mengirim produk ke konsumen.
   5. Pembelanjaan produksi terhadap pemesanan konsumen.
   6. Biaya produksi pada pesanan konsumen.
   7. Kepopuleran desain yang digunakan pada produk.
   8. *Size* yang telah dipesan per produk.

2. memodelkan data secara multidimensional yang dapat digunakan secara dinamis sesuai dengan informasi yang dibutuhkan.

3. Menerapkan metode logika *fuzzy* dalam pembangunan perangkat lunak.

## Batasan Masalah

Batasan masalah dalam pembangun perangkat lunak data *warehouse* ini adalah sebagai berikut:

1. *Data Base Management System*  yang digunakan adalah Microsoft SQL Server 2014.
2. Data yang digunakan adalah data yang ada di Perusahaan Spaceman.

3. Penganalisisan data *warehouse* menggunakan OLAP (*On-line Analytical Processing* ).

4. Analisis pembangunan perangkat lunak menggunakan *Object Oriented*.

## Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metode komparatif. Metode ini merupakan salah satu metode penelitian deskriptif yang bertujuan mencari jawaban atau fakta secara mendasar dengan menganalisis faktor-faktor penyebab terjadinya ataupun munculnya suatu fenomena [2]. Metode ini menggunakan teknik korelasi, yaitu meneliti ketergantungan dalam hubungan antar variabel [2].

### Metode Pengumpulan Data

1. Wawancara

Wawancara merupakan teknik pengumpulan data yang dilakukan dengan cara komunikasi langsung dengan responden, dimana komunikasi berupa pertanyaan dan jawaban yang bertujuan untuk menemukan permasalahan yang harus diteliti [2].

1. Kuesioner

Kuesioner merupakan teknik pengumpulan data dimana responden mengisi pertanyaan atau pernyataan yang disediakan dan dikembalikan kembali ke peneliti setelah terisi semua [2].

1. Observasi

Teknik perngumpulan data dengan cara mengamati langsung pola prilaku individu dalam situasi tertentu guna mendapatkan informasi tentang fenomena yang diinginkan [2].

1. Dokumen

Teknik pengumpulan data dengan cara mempelajari dan mengkaji catatan peristiwa yang sudah berlalu. Dalam penelitian ini dokumen yang di amati adalah catatan transaksi perusahaan baik itu berupa laporan, catatan keuangan maupun kepegawaian dan dokumen literatur yang membahas tentang penelitian ini [2].

### Metode Pembangunan Perangkat Lunak

Metode yang digunakan dalam pembangunan perangkat lunak pada penelitian ini menggunakan *waterfall model* atau *linear sequential model.* Model ini menyediakan pendekatan yang sistematis dan berurutan bagi pengembang perangkat lunak. Tahap – tahap pengembangan perangkat lunak dengan model ini adalah :

1. *Requirements Analysis and Definition*

Proses pengumpulan kebutuhan secara lengkap dalam tahap konsultasi dengan pengguna sistem yang kemudian digunakan sebagai bahan analisis dan pendefinisian spesifikasi sistem [3].

2. *System and Software Design*

Proses *software design* mengalokasikan segala persyaratan yang dibutuhkan baik dalam sisi perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan [3]. *Software design* merupakan proses yang melibatkan suatu identifikasi dan penggambaran abstrak dalam perwujudan dasar sistem dan relasinya [3].

3. *Implementation and Unit Testing*

Tahap ini merupakan tahap merealisasikan desain sistem sebagai serangkaian program atau unit program [3]. Sedangkan *unit testing* tahap untuk memverifikasi setiap unit terhadap spesifikasi kebutuhan sistem [3].

4. *Integration and System Testing*

Tahap pengintegrasian unit program menjadi satu program individu dan diuji sebagai sistem yang lengkap untuk memastikan bahwa persyaratan perangkat lunak telah terpenuhi [3]

5. *Operation and Maintenance*

Tahap sistem di-instalasi dan di pakai sebagai praktek untuk memenuhi kebutuhan yang seharusnya [3]. *Maintenance* merupakan proses mengoreksi kesalahan yang tidak ditemukan pada tahap siklus sebelumnya [3].

Requirements definition

System and software design

Implementation and unit testing

Integration and System Testing

Operation and Maintenance

**Gambar I‑1 *Waterfall model*** [3]

### Metode Pembangunan Data *Warehouse*

Metode yang digunakan dalam pembangunan data *warehouse* adalah metode *kimball* yang memiliki tahapan umum dalam pembangunan data warehouse. Tahap-tahap dalam metode *kimball* yang digunakan dalam pembangunan data *warehouse* adalah sebagai berikut :

1. *Business Requirements Definition*

Tahap ini merupakan tahap kolaborasi atau diskusi dengan pengguna bisnis untuk menentukan dan memahami kebutuhan yang di-inginkan dalam mencapai tujuan data *warehousing* [4]. Tahap ini hanya fokus pada pengumpulan kebutuhan-kebutuhan bisnis yang diperlukan [4].

1. *Dimensional Modelling*

Tahap ini merupakan tahap pemodelan dengan malakukan *review* terhadap kebutuhan bisnis pengguna, pemanfaatan alat pemodelan, pembangunan konvensi penamaan, koordinasi waktu yang dibutuhkan, dan perancangan pemodelan *dimensional* [4].

1. *Physical Design*

Tahap ini merupakan tahap pengembangan *dimensional*  model dan di dokumentasikan melalui sumber awal menuju ke targer pemetaan yang diterjemahkan ke dalam *phisical database* [4]. Tahap ini melakukan penavigasian pada model data dalam bentuk elemen tabel dan nama kolom yang mengandung arti dalam suatu bisnis, membangun pemodelan *database* yang bersifat fisik yang akan digunakan dan mendukung dalam pembangunan sistem *ETL*, mempersiapkan perencanaan inisalisasi *index* dalam tabel, dan merancang agregasi dalam suatu relasi pada perancangan fisik [4]. Hal ini bertujuan untuk untuk memberikan pemuatan data, pemeliharaan, dan kinerja *query* menjadi lebih efisien.

1. *ETL Design and Development*

Tahap ini merupakan tahap pemudelan dan pengembangan proses yang dibutuhkan dalam *warehousing* dengan pertimbangan terhadap data yang dibutuhkan dalam suatu bisnis, kualitas suatu data, kemanan data, integrasi data, *latency*  pada pengiriman sumber data yang harus dikirim ke sistem data *warehouse*, dan melakukan pengarsipan data [4]

1. *Deployment*

Tahap ini merupakan tahap penyajian penyebaran data yang dilakukan pada tahap sebelumnya. Dalam tahapan ini hal-hal yang bersifat kritis adalah melakukan pengujian sistem *end-to-end*, jaminan kualitas data, pengolahan operasi, kinerja sistem, dan pengujian kegunaan suatu sistem [4].

Gambar I‑2 *Kimball Life Cycle Data Track Diagram* [4]

Business Requirements Definition

Dimensional Modeling

Physical Design

ETL Design & Development

Deployment

### Metode Pemodelan *Fuzzy* Data *Warehouse*

Dalam pembangunan suatu *fuzzy* data *warehouse*, dibutuhkan suatu metode yang dapat mentransformasi suatu data *crisp* dalam data *warehose* menjadi  *fuzzy* data *warehouse* dimana nilai masukan awal pada metode ini adalah data *warehouse* klasik dan keluarannya merupakan  *fuzzy*  data  *warehouse* [5]*.* Adapun langkah pada metode ini adalah sebagai berikut:

1. *Defining Classification Elements*

Tahapan ini merupakan tahapan yang mendefensiskan kalsifikasi element yang akan digunakan pada tahapan selanjutnya dalam pembangunan suatu model *fuzzy* data *warehouse*. Tahapan ini melibatkan tiga langkah, yaitu:

1. *Identify Target Attribute*

Langkah ini merupakan langkah untuk mengidentifikasikan atribut target yang memiliki nilai-nilai yang akan diklasifikan secara *fuzzy* [5].

1. *Identify Linguistic Term*

Langkah ini merupakan langkah untuk mengidentifikasi *linguistik term* yang digunakan untuk mengklasifikasi target atribut dan menentukan nilai target atribut yang harus diklasifikan dalam suatu *lingusitic term* [5]*.*

1. *Define Membership Function*

Langkah ini merupakan langkah untuk mendefinisikan fungsi keanggotaan untuk setiap istilah linguistik yang telah ditetapkan [5]. Langkah ini dilakukan sedemikian rupa sehingga nilai dari atribut target dapat dipetakan melalui skala 0 sampai 1 [5].

1. *Building Fuzzy DWH Model*

Tahap ini merupakan tahap menggunakan elemen klasifikasi yang didapatkan pada tahap sebelumnya dalam membangun model *fuzzy* data *warehouse* [5]. Tahap ini melibatkan tiga langkah, yaitu:

1. *Model FCT*

Langkah ini merupakan langkah untuk memodelkan suatu tabel atau atribut klasifikasi pada setiap set dari *linguistic term* [5].

1. *Model FMT*

Langkah ini merupakan langkah untuk membuat tabel keanggotaan *fuzzy* [5]*.* Atribut yang ada di dalam tabel ini adalah atribut identifikasi (*primary key*), atribut identifikasi dari target atribut, atribut untuk klasifikasi *fuzzy*, dan atribut derajat keanggotaan dari target atribut [5]. Nilai dari atribut derajat keanggotaan dikalkulasikan berdasarkan fungsi keanggotaan yang telah ditetapkan [5].

1. *Relate FCT, FMT & TA*

Langkah ini merupakan langkah untuk merelasikan atau menghubungkan tabel keanggotaan fuzzy dan pengkalisifikasiaannya dengan tabel yang berisi atribut target [5]. Dalam perelasian ini, dibutuhkan penambahan kunci identifikasi dari target atribut yang direlasikan pada tabel *fuzzy* sebagai kunci tamu [5].

Gambar I‑3 *Method for modeling a fuzzy data warehouse*

*Defining Classification Elements*

*Identify Target Attribute*

*Identify Linguistic Term*

*Identify Membership Function*

*Building Fuzzy DWH Model*

*Model FCT*

*Model FMT*

*Relate FCT, FMT & TA*

## Sistematika Penulisan

Sistematika penulisan penelitian ini disusun agar mendapatkan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan penelitian ini adalah sebagai berikut :

**BAB I PENDAHULUAN**

Bab ini menguraikan tentang latar belakang permasalahan, mencoba merumuskan masalah, menentukan maksud dan tujuan, menentukan metodologi peneltian, serta sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Bab ini membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian-penelitian serupa yang telah pernah dilakukan sebelumnya termasuk sintesisnya. Membahas tentang konsep dasar serta teori-teori yang berkaitan dengan topik penelitian dan yang melandasi rancang bangun sistem.

**BAB III ANALISIS DAN PERANCANGAN SISTEM**

Pada Bab ini berisi tentang analisis data *warehouse*, analisis *OLAP* dan *reporting tool*, analisis kebutuhan fungsional dan *non* fungsional, dan perancangan sistem.

**BAB IV IMPLEMENTASI DAN PENGUJIAN**

Pada Bab ini berisi pembahasan implementasi serta penjelasan tentang teknik dan strategi pengujian sistem yang digunakan.

**BAB V KESIMPULAN DAN SARAN**

Pada Bab ini berisi tentang kesimpulan dan saran untuk pengembangan sistem selanjutnya.

# TINJAUAN PUSTAKA

## Profil Perusahaan

Spaceman Clothing Indonesia merupakan perusahaan yang bergerak dalam bidang jasa penyediaan barang hasil olahan konveksi seperti kaos, kemeja, polo, jaket, dan lain-lain. Perusahaan ini didirikan sejak tahun 2010 yang hingga saat ini terus berkembang dan tetap aktif melayani setiap pemesanan konsumen di seluruh wilayah Indonesia yang menggunakan jasa dari perusahaan ini. Perusahaan ini beralamat di jalan Cigadung Raya Timur No. 5 Kecamatan Cimeunyan Bandung.

### Visi dan Misi Perusahaan

Adapun visi dan misi dari perusahaan ini adalah sebagai berikut.

1. Visi

Menjadi perusahaan yang berskala nasional dan internasional yang mengedepankan kualitas, inovasi, dan profesionalisme dalam setiap kinerja team dalam upaya manjadi yang terdepan di bidangnya.

1. Misi
2. Mengangkat *prestise (image)* produk lokal sehingga dapat diperhitungkan dalam kancah industri global.
3. Menjadi perusahaan *clothing manufacture* yang menghasilkan produk berkualitas, inovatif, dan mengikuti perkembangan zaman yang dikonsumsi oleh pasar lokal regional, nasional, dan internasional.
4. Menjadi perusahaan yang dapat menyerap lebih banyak tenaga kerja profesional yang kompeten. Sehingga turut berpartisipasi dalam program pemerintah dalam penekanan tingkat jumlah pengangguran.

### Struktur Organisasi

Struktur Organasi merupakan pola hubungan antar bagian dari instansi atau menggambarkan dengan jelas pemisahan kegiatan pekerja antara bagian yang satu dengan bagian yang lain dalam suatu instnasi. Gambar II‑1menjelaskan struktur organisasi yang ada di Spaceman Clothing Indonesia.

Gambar II‑1 Struktur organisasi Spaceman Clothing Indonesia

Owner

Pengawas Produksi

Administrasi dan Perencanaan

Kepala Penjahit

Kepala Kualitas dan desain

Staff Lapangan

Staff Penjahit

Staff Sablon

Staff Cutting

### Deskripsi Kerja

a. Owner

Owner Memiliki fungsi sebagai pemimpin dan pengelola perusahaan dan bertugas untuk mewujudkan dan menjalankan visi dan misi, mengelola perkembangan perusahaan, keuangan, dan hal-hal yang berkaitan untuk perusahaan.

b. Pengawas Produksi

Pengawas Produksi memiliki fungsi sebagai mengawasi alur produksi terhadap suatu pesanan dan memiliki tanggung jawab mengontrol proses pesanan konsumen yang sedang dikerjakan dan mengontrol perpindahan bahan dari satu proses ke proses yang lain, dan mengontrol alat dan bahan yang dibutuhkan dalam suatu proses produksi terhadap suatu order.

c. Administrasi dan Perencanaan

Administrasi dan Perencanaan memiliki fungsi sebagai pengelola hal-hal bersifat dokumentasi dan estimasi terhadap suatu order dari konsumen dan memiliki tanggung jawab untuk membuat suat estimasi order, keperluan pembelanjaan bahan, pengecekan pembayaran, dan pembuatan laporan setiap order produksi.

d. Kepala Penjahit

Kepala penjahit memiliki fungsi sebagai pengelola distribusi tugas staff penjahit dan memiliki tanggung jawab untuk mengawas dan membantu staff penjahit dalam mengerjakan suatu produk yang di pesan oleh konsumen, memeriksa kelengkapan bahan yang akan dijahit, mengontrol kelengkapan peralatan yang digunakan selama menjahit, dan mengontrol kelayakan mesin untuk digunakan menjahit.

e. Kepala Kualitas dan Desain

Kepala kualitas dan design memiliki fungsi sebagai pengelola distribusi tugas staff sablon dan memiliki tanggung jawab untuk mengawas dan membantu staff sablon dalam mengerjakan suatu produk, membuat kalkir dari desain yang akan digunakan dalam proses sablon, mengontrol kelengkapan bahan yang digunakan, mengontrol kelayakan alat yang digunakan, dan mencocokkan warna yang akan digunakan.

f. Staff Penjahit

Staff penjahit memiliki fungsi sebagai penjahit yang menjaga kualitas dalam pengerjaannya dan memiliki tanggung jawab untuk mengambil tugas yang di berikan oleh kepala penjahit.

g. Staff sablon

Staff sablon memiliki fungsi sebagai tenaga keraja yang mengerjakan desain dari pakaian yang mempertahankan kualitas dalam pengerjaannya dan memiliki tanggung jawab untuk mengambil tugas yang di berikan oleh kepala kualitas dan desain.

h. Staff *Cuting*

Staff *cuting* memilki fungsi sebagai pemotong bahan yang digunakan dalam suatu pesanan sesuai dengan kebutuhan yang diperlukan dan memiliki tanggung jawab untuk membuat pola potong baru, memotong bahan yang digunakan se-optimal mungkin dengan penggunaan bahan, dan mengontrol ketersediaan bahan untuk dipotong.

i. Staff Lapangan

Staff lapangan memiliki fungsi sebagai tenaga kerja yang digunakan dalam proses pengadaan alat dan bahan yang di perlukan dalam suatu pesanan konsumen dan memiliki tanggung jawab untuk membeli alat dan bahan yang diperlukan, mengontrol biaya yang tersedia dalam pembelian alat dan bahan, dan memiliki alternatif untuk mengambil keputusan dalam pembelian alat dan bahan.

## Landasan Teori

### Database

*Database* merupakan sekumpulan data saling berhubungan yang mampu menyediakan informasi yang relevan untuk suatu organisasi atau *enterprise* [6]*.* Sedangkan alat untuk mengelola dan mengakses *database* disebut *DBMS ( Database Management System)*. Tujuan utama dari *DBMS* adalah menyediakan cara untuk menyimpan dan mengambil informasi dari *database* dengan baik, nyaman, dan efisien [6].

### Skema Diagram

Skema diagram merupakan gambaran dari skema *database* bersamaan dengan kunci primer dan depedensinya dengan kunci tamu yang setiap relasinya diwakili sebagai kotak yang dilengkapi dengan nama relasi dan atribut [6]. Atribut yang muncul sebagai atribut kunci primer akan digaris bawahi di dalam kotak dan atribut sebagai kunci tamu muncul dalam atribut di dalam kotak yang memiliki panah dari atribut kunci tamu ke kunci primer sebagai hubungan referensi [6].

### Data *Warehouse*

Data *warehouse* merupakan suatu sistem yang menyajikan suatu informasi, di dalam sistem tersebut terdapat integrasi dan transformasi terhadap data dari berbagai sistem operasional yang ada baik data yang bersumber dari internal maupun eksternal kedalam bentuk suatu informasi yang relevan untuk penunjang keputusan dalam starategi bisnis [7]. Dalam artian lain, data *warehouse* juga merupakan *repository* (arsip) informasi yang dikumpulkan dari berbagai sumber, disimpan di dalam skema yang *unified* di dalam aplikasi tunggal yang akan disimpan dalam waktu yang lama dan memungkinkan untuk mengakses data historis, sehingga menyediakan informasi untuk pendukung keputusan dari data *warehouse* [6]*.*

#### Karakteristik Data *Warehouse*

Karakteristik data didalam data *warehouse* adalah sebagai berikut:

1. Berorientasi subjek

Didalam data *warehouse*, data dibentuk berdasarkan subjek bisnis bukan berdasarkan fungsi pada sistem operasional pada aplikasi tertentu [7]. Sehingga data fokus terhadap suatu subjek yang dianggap kritikal untuk dijadikan informasi sebagai penunjang keputusan yang akan digunakan sebagai strategi bisnis yang diterapkan [7].

1. Terintegrasi

Data didalam data *warehouse* bersumber dari beberapa sistem dimana sumber datanya berada di *database*, *file*, atau segmentasi data yang berbeda [7]. Syarat integrasi terhadap sumber data dapat dipenuhi dengan cara memperbaiki data yang tidak konsisten, melakukan standarisasi terhadap elemen data, dan memastikan maksud dari nama-nama variabel yang digunakan di dalam data yang berasal dari setiap sumber yang digunakan [7].

1. *Time Variant*

Data di dalam data *warehouse* merupakan data yang bersumber dari data pada periode yang lalu dan sekarang, dimana setiap struktur data di dalam data *warehouse* mengandung elemen waktu dikarenakan data di dalam data *warehouse*  dimaksudkan untuk sebagai bahan analisis dan penunjang keputusan yang mengandung tidak hanya sekedar data di masa sekarang, tetapi mengandung data pada masa lampau juga [7].

1. *NonVolatile*

Data operasional dipindahkan ke dalam data *warehouse* dalam spesifikasi interval tertentu, tergantung dari kebutuhan bisnis yang diinginkan [7]. Di dalam data *warehouse* data diekstraksi sebagai *snapshot* dalam periode waktu, bisa dalam periode perminggu, perbulan, atupun pertahun [7]. Setiap data transaksional tidak dapat melakukan pembaharuan, penginputan, ataupun menghapus data secara *real-tme* di dalam data *warehouse* seperti yang dapat dilakukan di sistem operasional melainkan melakukan *refreshing* dari data operasional yang kemudian dimuat kembali ke dalam data *warehouse*.

#### Arsitektur Data *warehouse*

Arsitektur data *warehouse* merupakan struktur yang menyajikan semua komponen yang terlibat didalam data *warehouse* secara bersamaan [7]. Didalam data *warehouse*, arsitektur termasuk data yang terintegrasi sebagai satuan yang terpusat, semua kebutuhan untuk persiapan data dan penyimpanannya, dan arah penyajian informasi dari data *warehouse*  sehingga menghasilkan suatu aturan, prosedur, dan fungsional untuk memungkinkan data *warehouse* bekerja dan memenuhi kebutuhan bisnis [7]. Salah satu arsitektur yang dapat digunakan adalah arsitektur *three major areas* yang meliputi *data acquisition, data storage,*  dan *information delivery* [7]*.*

Gambar II‑2 Komponen arsitektur dalam *three major areas*

*Internal*

*Data Staging*

*Data Source*

*External*

*production*

*Archived*

***Data Acquisition***

*Data warehouse*

*DBMS*

*Data Marts*

*MDDB*

*Meta data*

***Data Storage***

*OLAP*

***Information Delivery***



*Data mining*

*Report / Query*

### Skema Data *Warehouse*

Dalam memodelkan suatu sistem data *warehouse* menggunakan teknik pemodelan *dimensional* atau juga disebut *dimensional model* [7]. Pemodelan ini berguna untuk menetapkan untuk membuat tabel dimensi, tabel fakta dan relasinya [7]. Hasil dari pemodelan ini akan membentuk skema yaitu:

1. *Star Schema*

Skema ini merupakan skema dasar yang terbentuk dari teknik perancangan dasar untuk data *warehouse*. Skema ini menggambarkan suatu tabel fakta yang ditampilkan di tengah-tengah diagram dan dikelilingi oleh dimensi-dimensi yang berelasi pada tabel fakta [7]. Relasional antar tabel dimensi dengan tabel fakta merupakan relasi satu ke banyak [7]. Skema ini sering digunakan dalam pembuatan sistem data *warehosue* dikarenakan lebih mudah dimengerti oleh pengguna, mengoptimalkna navigasi melalui suatu *database*, dan skema yang paling cocok dalam melakukan *qurey* *processing* [7]*.*

1. *Snowflake Schema*

*“Snowflaking”* merupakan metode normalisasi tabel dimensi dalam skema STAR dimana setelah semua tabel dimensi telah benar-benar dinormalisasikan, struktur yang dihasilkan menyerupai struktur salju dengan tabel fakta di tengah [7]. Prinsip dari *snowflaking* adalah menormalisasi tabel dengan menghapus atribut kardinalitas rendah dan membentuk tabel terpisah. Alasan melakukan *snowflaking* juga adalah untuk penghematan ruang penyimpanan dalam satu tabel dan penelusuran atribut lebih terperinci dalam suatu tabel dimensi [7].

### *ETL ( Extract, Transform, Loading)*

*ETL* merupakan himpunan fungsi yang dilakukan untuk mengubah dan membentuk kembali data ke dalam bentuk yang berbeda pada data di dalam sistem operasional yang disimpan di dalam data *warehouse* sebagai informasi yang relevan dan strategis [7]. Adapun kelompok himpunan *ETL* adalah ekstraksi data, transformasi, dan *loading* yang menjadi tahapan proses pengubahan dan pembentukan ulang data yang digunakan di dalam data *warehouse* [7]*.*

1. *Extraction*

Tahap extraction merupakan tahap untuk mengidentifikasi semua sumber data internal, menentukan proses komputasi dan sumber data mana yang akan di ekstrak, menentukan kompatibilitas struktur data jika dan hanya jika bila menggunakan sumber-sumber dari luar, dan mengindikasi metode untuk mengekstraksi data [7].

1. *Transform*

Tahap *transform* memiliki fungsi meliputi pemilihan input, pemisahan struktur input, normalisasi dan denormalisasi dari struktur data sumber, mengagregasi, mengkonversi, dan memecahkan nilai yang hilang [7].

1. *Loading*

Tahap *loading* merupakan tahap menginisialisasi *load* awal, menentukan seberapa sering suatu kelompok data harus tetap *up-to-date* dalam data *warehouse*, dan menentukan cara mengubah data yang akan dilaksanakan dalam periode waktu tertentu.

### *OLAP (On-Line Analytical Processing)*

*OLAP* merupakan salah satu katergori teknologi perangkat lunak yang memungkinkan analis, manajer, dan eksekutif untuk menggali suatu data secara cepat, konsisten, dan memiliki akses yang interaktif dalam luasnya penggalian suatu informasi yang di transformasikan dari suatu *raw*  data menjadi dimensi fakta yang dapat dimengerti oleh pengguna [7].

1. kriteria *OLAP*

Suatu sistem *OLAP* harus memiliki kriteria yang dapat menghasilkan suatu prioritas yang akan dituju. Adapun kriteria dari *OLAP* adalah sebagai berikut.

1. *Multidimensional Conceptual View*

Suatu model data *multidimensional* mangacu pada bagaimana peresepsi pengguna untuk memecahkan suatu masalah bisnis yang dapat menciptakan suatu analisis yang intuitif dan mudah digunakan [7].

1. *Transparency*

Teknologi, gudang data, arsitektur komputasi, dan data alami dari sumber ditransparasikan terhadap pengguna, dengan tujuan untuk dapat membantu meningkatkan ke-efisienan dan produktivitas pengguna [7].

1. *Accessibility*

Suatu akses yang dihasilkan untuk menampilkan data harus dapat menghasilkan hasil yang dapat di analis secara spesifik, dipresentasikan sebagai tunggal, konherensi, dan konsisten bagi pengguna yang melihatnya [7].

1. *Consistent Reporting Performance*

*OLAP* memastikan kekonsistenan penggunaan dimensi atau ukuran yang digunakan sehingga pengguna dapat melaporkan kinerja dengan efisien terhadap waktu baik itu waktu dijalankan, waktu respon, atau pemanfaat sistem setiap kali *query* digunakan.

1. *Client/Server Architecture*

Pembangunan *OLAP* sesuai dengan prinsip arsitektur klien atau server agar kinerja optimal, fleksibel, mampu beradaptasi, dan inter-operabilitas [7]. Sehingga pengguna tidak perlu menginputkan *query* lagi dalam pembentukan ulang [7].

1. *Generic Dimensionality*

*OLAP* memastikan terhadap setiap dimensi memiliki kesetaraan dalam struktur, kemampuan operasional, dan memiliki satu struktur yang logis [7].

1. *Multi User Support*

*OLAP* memberikan dukungan pada *end user* untuk bekerja secara bersamaan dengan model analisis yang sama atau untuk model yang berbeda pada data yang sama [7].

1. *Intuitive Data Manipulation*

*OLAP* memungkinkan berbagai macam jalur konsolidasi re-orientasi pada pemanipulasian informasi seperti *drill-down, roll-up,* dan manipulasi yang dilakukan secara intuitif dan langsung [7].

1. *Flexible Reporting*

*OLAP* memberikan kebebasan pengguna untuk mengatur dengan mudah suatu kolom, baris dan sel dengan fasilitas manipulasi yang mudah, analisis, dan sintesis informasi [7].

1. *Unlimited Dimensions and Agregation Levels.*

*OLAP* dapat mengakomodasi banyak dimensi data dalam suatu model umum analisis.

1. Karakteristik *OLAP*

Karakteristik dasar dari *OLAP* adalah sebagai berikut.

1. Memastikan pengguna memiliki suatu *view* yang *multidimensional* dan *logical* dalam suatu data warehouse [7].
2. Memfasilitasi *query* yang interaktif dan analisis yang kompleks untuk pengguna [7].
3. Memungkinkan user untuk melakukan *drill down* untuk mendapatkan detail yang lebih rinci atau melakukan *roll up* untuk proses agregasi dari suatu *metrics* dalam suatu dimensi tunggal ataupun dalam *multiple* dimensi [7].
4. Menghasilkan suatu *view* yang dapat menyajikan arti dari segala arah, termasuk *chart*  dan *graph* [7].

### *Fuzzy* Data *Warehouse*

*Fuzzy dimension* merupakan dimensi atau fakta yang memuat dan mengidentifikasi unsur-unsur *fuzzy* terhadap data yang digunakan untuk kepentingan analisis [1]. Data yang digunakan harus dipertimbangkan guna untuk memberikan suatu manfaat terhadap proses bisnis [1]. Informasi dasar yang dibutuhkan dalam mendefinisikan variabel *fuzzy*  adalah tabel sumber dan target atribut, asosiasi, jenis atribut, sifat-sifatnya, dan cara perhitungannya [1].

Dalam suatu sistem data *warehouse*, konsep *fuzzy* dapat diintegrasikan sebagai struktur *meta table* tanpa mempengaruhi inti dari suatu data *warehouse* [5]. Dengan pendekatan *fuzzy* *data warehousing* memungkinkan untuk mengekstraksi dan menganalisis data secara bersamaan dalam bentuk klasik yang jelas dan dengan cara *fuzzy* [5].

### Konsep *Fuzzy* Data *Warehouse*

Pengintegrasian konsep *fuzzy* menjadi data *warehouse* merupakan penganalisisan elemen-elemen dalam data *warehouse* yang bisa diklasifikasikan secara *fuzzy* dimana elemen yang dianalisis bisa saja suatu fakta di dalam tabel fakta ataupun suatu atribut dari suatu dimensi [5].

Target atribut (TA) merupakan suatu atribut dimensi atau fakta di tabel fakta yang diperlukan untuk diklasifikasi secara *fuzzy* [5]. Atribut domain merupakan suatu set kemungkinan nilai atau rentang kemungkinan nilai dari suatu atribut dimensi atau fakta yang bisa di petakan dalam suatu *lingustic term* yang akan dipetakan kedalam himpunan kelas dalam suatu konsep *fuzzy* [5] . Pemodelan *fuzzy* data *warehouse* merupakan suatu kombinasi dari empat elemen yaitu tabel dimensi, tabel fakta, *fuzzy membership*, dan *fuzzy classification* [5]. Ada tiga tipe dalam memodelkan *fuzzy data warehouse* yang dapat digunakan untuk merancang konsep *fuzzy* [5], yaitu:

1. *Distinct Fuzzy Classes / Linguistic Terms*

*Linguistic terms* tunggal yang digunakan tanpa adanya pengulangan yang dilakukan terhadap lainnya, yang artinya satu atribut target hanya diperuntukkan terhadap satu kelas dan derajat keanggotaan *fuzzy-*nya terhadap relasi yang diukur oleh fungsi keanggotannya [5].

1. *Different Membership Degrees for the Same Linguistic Terms*

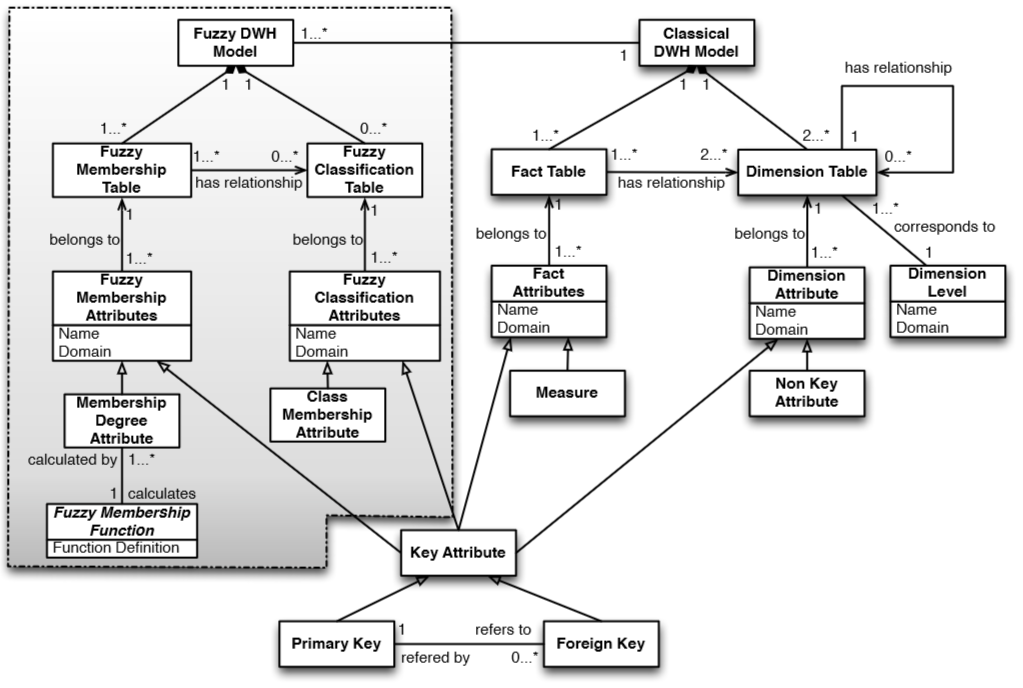
Suatu atribut target yang mungkin diperuntukkan pada satu *linguistic terms* tetapi juga memungkinkan untuk memiliki perbedaan kederajatan terhadap *linguistic terms*  yang ada [5].

1. *Different Linguistic Terms for a Target Attribut*

Suatu target atribut yang mungkin diperuntukkan pada *multiple linguistic terms*, sehingga dapat memiliki *multiple class*  dan memiliki *multiple* fungsi keanggotaannya [5].

### Meta Model *Fuzzy* Data *Warehouse*

Suatu meta model mendefinisikan elemen-elemen dari konseptualisasi dan hubungan mereka [5]. Dalam *fuzzy* data *warehouse*, meta model dalam data *warehouse* saling terintegrasi dengan konsep *fuzzy* sebagai struktur meta tabel. Model meta data *warehouse* mengacu pada skema data *warehouse* yang terdiri dari satu atau lebih tabel fakta yang terletak di pusat untuk menyajikan informasi proses bisnis dan dua atau lebih tabel dimensi [5]. Model meta *fuzzy* data *warehouse*  mengacu pada konsep *fuzzy*  yang terintegrasi dengan suatu data *warehouse* dimana untuk setiap target atribut yang diidentifikasikan dalam model *fuzzy* data *warehouse* bisa lebih dari satu dan begitu juga sebaliknya [5].



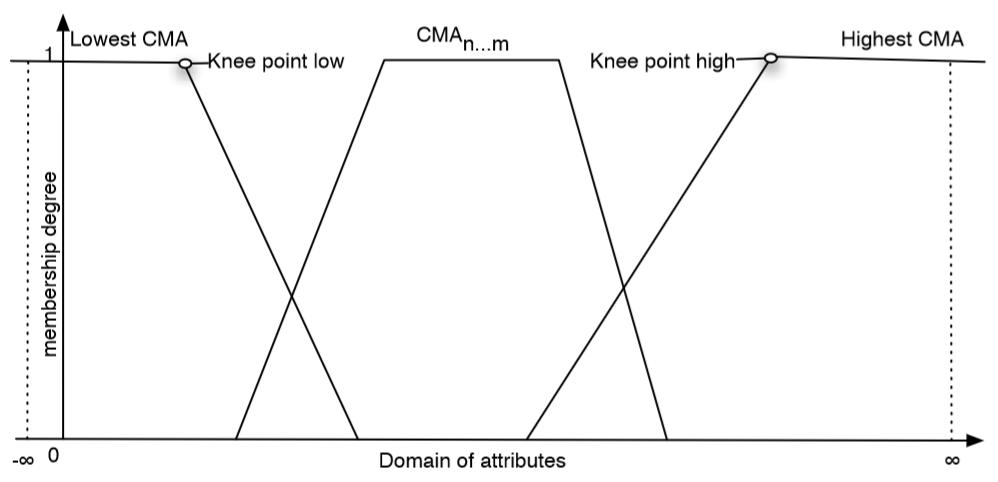
Gambar II‑3 meta *fuzzy* data *warehouse* [5]

### Tipe Konsep *Fuzzy*

Konsep *fuzzy* harus benar-bnar bisa menangani rentang nilai yang dimiliki oleh target atribut [5]. Berdasarkan dengan bagaimana suatu rentang nilai harus digunakan, maka pendefinisian konsep *fuzzy* bisa jadi berbeda [5]. Konsep *fuzzy* dapat dibedakan menjadi tiga jenis, yaitu:

1. *Open End Fuzzy Concept*

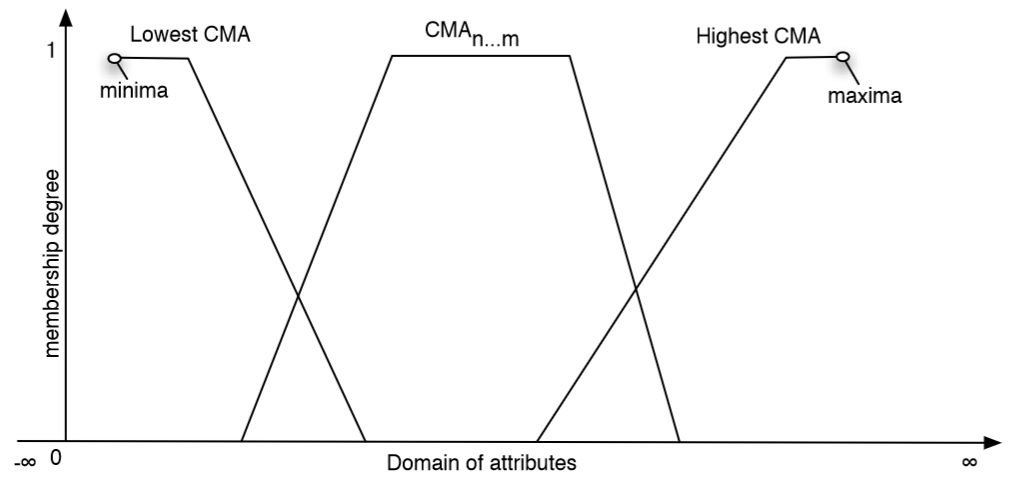
Konsep *fuzzy* yang memberikan keakuratan suatu klasifikasi antar titik yang mencakupi semua batasan dari nilai target atribut dan semua derajat keanggotaan [5]. Konsep ini membutuhkan banyak atribut derajat keanggotaan dalam tabel keanggotaan sebagai jumlah cartesian produk dari atribut kelas dan atribut target.



Gambar II‑4 *Open end fuzzy concept* [5]

1. *Limited Fuzzy Concept*

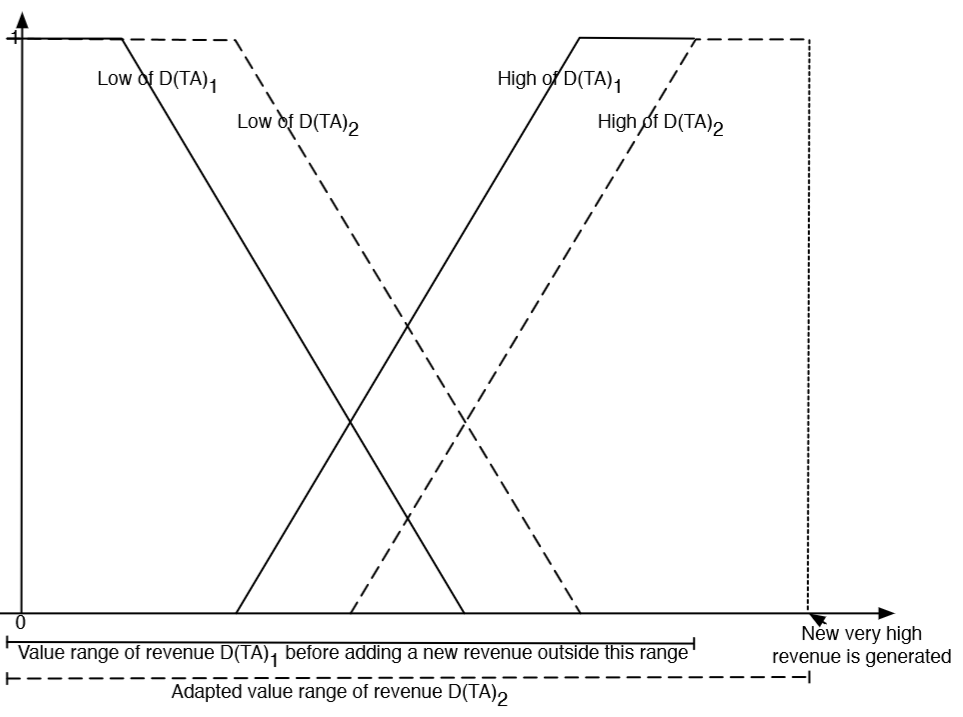
Konsep ini merupakan konsep yang terbatas dimana nilai target atribut yang didefinisikan di luar domain menjadi nilai yang tidak dianggap dan dibuang [5]. Nilai terendah dan nilai tertinggi pada nilai dari target atribut didefinisikan pada titik minima dan maksima [5].



Gambar II‑5 *Limited fuzzy concept* [5]

1. *Adaptive Fuzzy Concept*

Konsep ini merupkan konsep yang menggunakan atribut domain yang lengkap dimana konsep *fuzzy* tidak didefinisikan sebagai titik tetap atau sebagai titik minima dan maksima [5]. Atribut kelas didefinisikan secara relatif terhadap rentang nilai dan mencakup berbagai persentil yang dapat menyesuaikan diri dengan domain baru secara otomatis.



Gambar II‑6 *Adaptive fuzzy concept* [5]

### *Object Oriented Analysis and Design* (*OOAD*)

*OOAD* merupakan metode dan teknik pengembangan sistem yang bebasis objek [8]. Pendekatan *OOAD* menggabungkan data dan proses menjadi entitas tunggal yang disebut objek [8]. Objek biasanya terkorespondensi terhadap suatu sistem informasi seperti kostumer, suppliers, atau kontrak [8]. Tujuan dari *OOAD*  adalah untuk membuat elemen sistem lebih *reusable*, *improving system quality* dan produktivitas terhadap analisis dan desain sistem [8]. Alat bantu yang digunakan untuk membantu pemodelan dalam *OOAD* adalah *UML* ( *Unified Modeling Language )*. Berikut ini adalah diagram yang digunakan dalam *UML.*

1. *Use Case*

*Use Case* merupakan *case* atau situasi yang digunakan oleh sistem untuk memenuhi satu atau lebih dari kebutuhan pengguna, *use case* menangkap bagian-bagian dari fungsional yang disediakan oleh sistem [9].

1. *Activity Diagram*

*Activity* *Diagram* memungkinkan dalam menentukan bagaimana sistem mencapai tujuan dalam tindakan-tindakan yang dilakukan dalam suatu proses yang telah di gambarkan pada *use case* [9].

1. *Class Diagram*

*Class Diagram*  merupakan model hubungan dari *class* yang digunakan didalam sistem yang menggambarkan berbagai jenis objek yang ada pada sistem [9].

1. *Sequence Diagram*

*Sequence Diagram*  merupakan gambaran interaksi antar bagian-bagian yang ada di dalam sistem yang dipicu ketika suatu proses *use case* dijalankan dalam sistem [9].

# ANALISIS DAN PERANCANGAN DATA WAREHOUSE

## Analisis Data *Warehouse*

Analisis data *warehouse* merupakan suatu analisis yang dilakukan untuk mendapatkan kejelasan mengenai proses yang digunakan dalam membangun sistem data *warehouse* yang baik dan benar. Langkah dalam analisis ini yaitu analisis masalah, analisis sumber data, analisis kebutuhan informasi strategis, analisis data *staging,* dananalisis dimensi dan fakta.

### Analisis Masalah

Masalah yang ada pada perusahaan Spaceman sekarang adalah sebagai berikut:

1. Desain Data

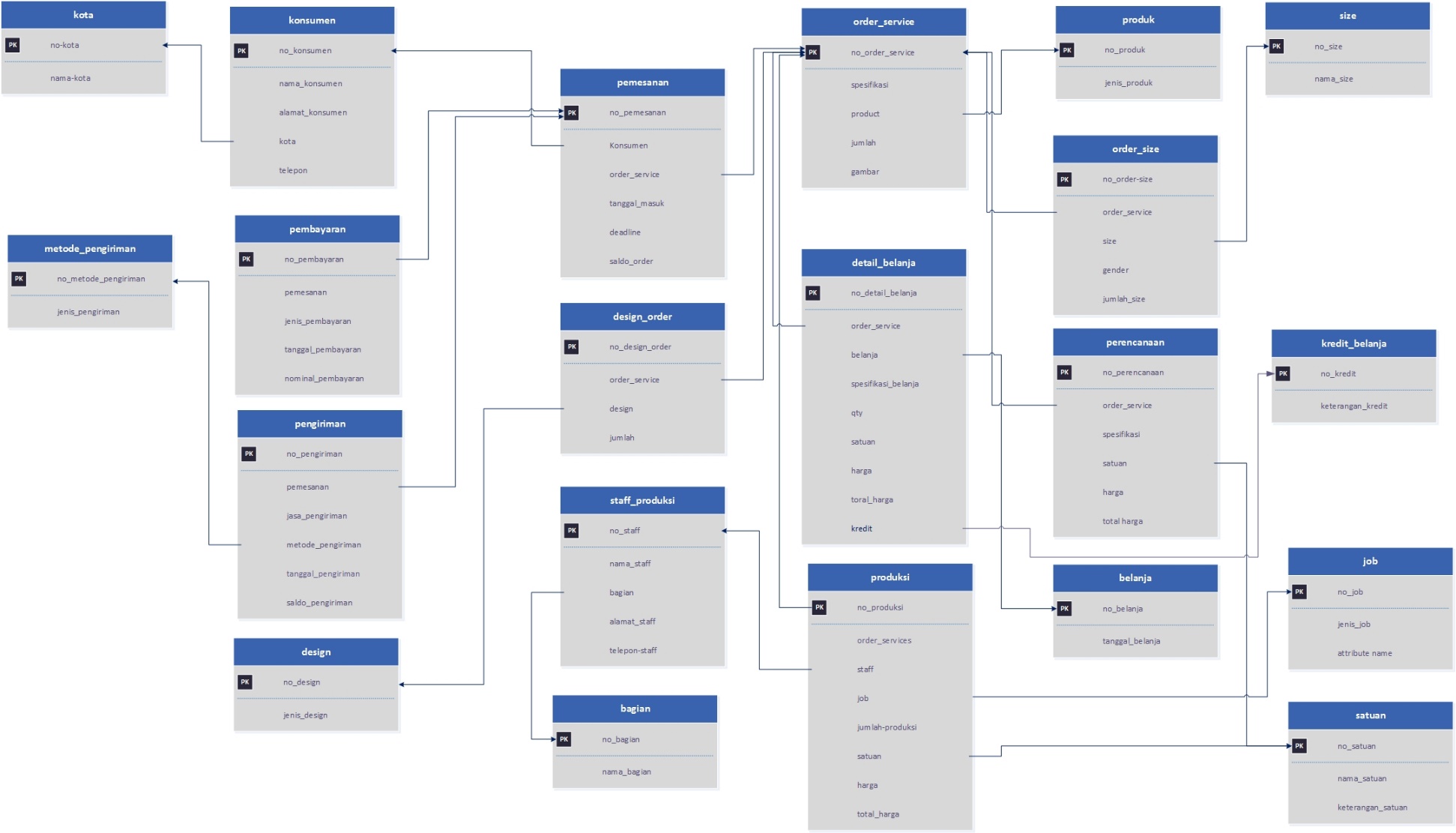
Desain data merupakan elemen penting dalam penunjang untuk mengambil keputusan. Desain data yang diterapkan di dalam perusahaan masih menggunakan konsep yang sederhana dan belum terintegrasi, sehingga sulit untuk melakukan suatu analisa pada informasi yang memerlukan desain yang dapat menampilkan banyak data dari data sekarang maupun data *historis*.

1. Pembuatan Laporan

Pembuatan laporan yang dilakukan dalam perusahaan dengan cara merekap data yang terlibat dalam transaksi suatu pemesanan konsumen yang kemudian dibuat laporan dengan bentuk laporan masing-masing. Masalah dalam pembuatan laporan ini adalah dengan merekap data yang terlibat sehingga harus melakukan pengecekan kembali dan hasil laporan yang tidak mendukung untuk dapat dilihat dari berbagai aspek yang terlibat dalam laporan tersebut.

### Analisis Sumber Data

Analisis sumber data pada perusahan Spaceman berdasarkan pada database yang sudah ada. Berikut skema OLTP yang dibutuhkan:



Gambar III‑1 Skema diagram *OLTP*

1. Tabel kota

Tabel kota merupakan tabel untuk menyimpan data kota konsumen yang melakukan pemesanan produk.

Tabel III‑1 Struktur data tabel kota

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_kota | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Nama\_kota | Varchar | 20 |  |  |

1. Tabel konsumen

Tabel konsumen merupakan tabel untuk menyimpan data profil konsumen yang melakukan pemesanan produk.

Tabel III‑2 Struktur data tabel konsumen

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_konsumen | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Nama\_konsumen | Varchar | 25 |  |  |
| 3 | Alamat | Varchar | 255 |  |  |
| 4 | Kota | Integer |  | *Foreign Key* | *Reference*(kota (no\_kota)) |
| 5 | Telepon | Varchar | 13 |  | Hanya bisa menginput numerikal |

1. Tabel order\_service

Tabel *order\_service* merupakan tabel yang menyimpan data produk yang akan diorder oleh konsumen.

Tabel III‑3 Struktur data tabel *order\_service*

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_order\_service | Varchar | 21 | *Primary Key* | Kode order bebentuk  Space/ord/bulan/urutan order/tahun |
| 2 | Spesifikasi | Varchar | 100 |  |  |
| 3 | Produk | Integer |  | *Foreign key* | *Reference*(produk (no\_produk)) |
| 4 | Jumlah | Integer |  |  |  |
| 5 | Gambar\_order | Varchar | 100 |  |  |

1. Tabel pemesanan

Tabel pemesanan merupakan tabel yang menyimpan data pemesanan konsumen yang memesan produk yang ada pada di tabel *order\_service*.

Tabel III‑4 Struktur data tabel pemesanan

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_pemesanan | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Service\_order | Varchar | 21 | *Foreign key* | *Reference*(*service\_order* (no\_*service\_order*)) |
| 3 | Konsumen | Integer |  | *Foreign key* | Reference(konsumen (no\_konsumen)) |
| 4 | Tanggal\_masuk | Date |  |  |  |
| 5 | Deadline | Date |  |  |  |
| 6 | Jumlah | Integer |  |  |  |
| 7 | Saldo\_order | Float |  |  |  |

1. Tabel perencanaan

Tabel perencanaan merupakan tabel yang menyimpan data estimasi pemesanan konsumen terhadap produk\_service yang dipesan.

Tabel III‑5 Struktur data tabel perencanaan

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_estimasi | Integer |  | *Primary key* | *Identity* (1,1) |
| 2 | Service\_order | Varchar | 21 | *Foreign key* | *Reference*(*service\_order* (no\_*service\_order*)) |
| 3 | Spesifikasi | Varchar | 100 |  |  |
| 4 | Jumlah | Float |  |  |  |
| 5 | Satuan | Integer |  | *Foriegn key* | *Reference*(satuan (no\_satuan)) |
| 6 | Harga | Float |  |  |  |
| 7 | Total\_harga | Float |  |  |  |

1. Tabel pengiriman

Tabel pengiriman merupakan tabel yang menyimpan data transaksi pengiriman produk ke konsumen.

Tabel III‑6 Struktur data tabel pengiriman

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_pengiriman | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Pemesanan | Integer |  | *Foreign key* | *Reference*(pemesanan (no\_pemesanan)) |
| 3 | Jasa\_pengiriman | Varchar | 100 |  |  |
| 4 | Metode\_pengiriman | Integer |  | *Foreign key* | *Reference*(metode\_pengiriman (no\_metode\_pengiriman)) |
| 5 | Nominal\_pengiriman | Float |  |  |  |

1. Tabel metode\_pengiriman

Tabel metode\_pengiriman merupakan tabel yang digunakan untuk menyimpan metode pengiriman produk ke konsumen.

Tabel III‑7 Struktur data tabel metode\_pengiriman

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_metode\_pengiriman | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Nama\_metode\_pengiriman | Varchar | 100 |  |  |

1. Tabel pembayaran

Tabel pembayaran merupakan tabel yang digunakan untuk menyimpan data transaksi pembayaran pemesanan terhadap konsumen.

Tabel III‑8 Struktur data tabel pembayaran

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_pembayaran | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Pemesanan | Integer |  | *Foreign key* | *Reference*(pemesanan (no\_pemesanan)) |
| 3 | Jenis\_pembayaran | Varchar | 100 |  |  |
| 4 | Nominal\_pembayaran | Float |  |  |  |
| 5 | Keterangan\_pembayaran | Enum | (“DP”,”Lunas”) |  |  |

1. Tabel *order\_size*

Tabel *order\_size* merupakan tabel yang digunakan untuk menyimpan data ukuran produk yang dipesan oleh konsumen.

Tabel III‑9 Struktur data tabel *order\_size*

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_order\_size | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Service\_order | Varchar | 21 | *Foreign key* | *Reference*(*order\_service* (no\_*order\_service*)) |
| 3 | Size | Integer |  | *Foreign key* | *Reference*(*size*(no\_*size*)) |
| 4 | Gender | Enum | “male”,”female” |  |  |
| 5 | Jumlah | Integer |  |  |  |

1. Tabel *size*

Tabel *size* merupakan tabel yang digunakan untuk menyimpan data jenis ukuran yang akan digunakan pada produk.

Tabel III‑10 Struktur data tabel *size*

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_size | Integer |  | *Primary Key* | *Identity(*1,1) |
| 2 | Nama\_size | Varchar | 7 |  |  |

1. Tabel *design\_order*

Tabel *design\_order* merupakan tabel yang digunakan untuk menyimpan data desain yang digunakan pada suatu produk yang dipesan oleh konsumen.

Tabel III‑11 Struktur data tabel *design\_order*

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_design\_order | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Service\_order | Varchar | 21 | *Foreign key* | *Reference*(pemesanan (no\_pemesanan)) |
| 3 | Design | Integer |  | *Foreign key* | *Reference*(*design* (no\_*design*)) |
| 4 | Jumlah | Integer |  |  |  |

1. Tabel *design*

Tabel *design* merupakan tabel yang digunakan untuk menyimpan data jenis desain yang akan digunakan pada produk.

Tabel III‑12 Struktur data tabel *design*

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_design | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Jenis\_design | Varchar | 200 |  |  |

1. Tabel produksi

Tabel produksi merupakan tabel yang digunakan untuk menyimpan data proses produksi yang dlakukan untuk pengadaan suatu produk.

Tabel III‑13 Struktur data tabel produksi

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_produksi | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Service\_order | Varchar | 21 | *Foreign key* | *Reference*(*order\_service* (no\_*order\_service*)) |
| 3 | Staff | Integer |  | *Foreign key* | *Reference*(staff\_produksi (no\_staff\_produksi)) |
| 4 | *Job* | Integer |  | *Foreign key* | *Reference*(*job* (no\_*job*)) |
| 5 | Jumlah | Float |  |  |  |
| 6 | Satuan | Integer |  | *Foreign key* | *Reference*(satuan (no\_satuan)) |
| 7 | Harga | Float |  |  |  |
| 8 | Total\_harga | Float |  |  |  |

1. Tabel *job*

Tabel *job* merupakan tabel yang digunakan untuk menyimpan jenis pekerjaan yang dilakukan terhadap proses produksi untuk pengadaan produk.

Tabel III‑14 Struktur data tabel job

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_job | Integer |  | *Primary Key* | *Identity* (1,1) |
| 2 | Jenis\_job | Varchar | 100 |  |  |

1. Tabel\_staff\_produksi

Tabel staff\_produksi merupakan tabel yang digunakan untuk menyimpan data profil pegawai yang bekerja di bagian produksi.

Tabel III‑15 Struktur data staff\_produksi

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_staff\_produksi | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Nama staff | Varchar | 30 |  |  |
| 3 | Bagian | Integer |  | *Foreign key* | *Reference*(bagian (no\_bagian)) |
| 4 | Alamat | Varchar | Max |  |  |
| 5 | Telepon | Varchar | 13 |  | Penginputan hanya boleh numerikal |

1. Tabel bagian

Tabel bagian merupakan tabel yang digunakan untuk menyimpan data bagian suatu pegawai di bagian produksi.

Tabel III‑16 Struktur data tabel bagian

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_bagian | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Nama\_bagian | Varchar | 20 |  |  |

1. Tabel satuan

Tabel satuan merupakan tabel yang digunakan untuk menyimpan data satuan yang digunakan terhadap alat atau bahan yang menggunakan satuan.

Tabel III‑17 Struktur data tabel satuan

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_satuan | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Nama satuan | Varchar | 5 |  |  |
| 3 | Keterangan | Varchar | 50 |  |  |

1. Tabel detail\_belanja

Tabel detail\_belanja merupaan tabel yang digunakan untuk menyimpan data pembelanjaan yang dibutuhkan untuk mengadakan suatu produk.

Tabel III‑18 Struktur data tabel detail\_belanja

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_detail\_belanja | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Service\_order | Varchar | 21 | *Foreign key* | *Reference*(*Order\_service* (no\_*order\_service*)) |
| 3 | Belanja | Integer |  | *Foreign key* | *Reference*(belanja (no\_belanja)) |
| 4 | Qty | Float |  |  |  |
| 5 | Satuan | Integer |  | *Foreign key* | *Reference*(satuan (no\_satuan)) |
| 6 | Harga | Float |  |  |  |
| 7 | Total\_harga | Float |  |  |  |
| 8 | Kredit | Integer |  | *Foreign Key* | *Reference*(kredit (no\_kredit)) |

1. Tabel belanja

Tabel belanja merupakan tabel yang digunakan untuk menyimpan data aktivitas belanja.

Tabel III‑19 Struktur data tabel belanja

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_belanja | Integer |  | *Primary Key* | *Identity*(1,1) |
| 2 | Tanggal\_belanja | Date |  |  |  |

1. Tabel kredit\_belanja

Tabel kredit\_belanja merupakan tabel yang digunakan untuk menyimpan jenis pembayaran dalam pembelian bahan dan alat produksi perusahaan.

Tabel III‑20 Struktur data tabel kredit\_belanja

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_kredit | Integer |  | *Primary Key* | *Identity*(1,1) |
| 3 | Keterangan | Varchar | 50 |  |  |

1. Tabel produk

Tabel produk merupakan tabel yang digunakan untuk menyimpan jenis produk yang dapat dipesan oleh konsumen.

Tabel III‑21 Struktur data tabel produk

| **No.** | **Nama Field** | **Type** | **Value** | **Status** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| 1 | No\_produk | Integer |  | *Primary Key* | *Iendtity*(1,1) |
| 2 | Nama\_produk | Varchar | 20 |  |  |

### Analisis Kebutuhan Informasi Strategis

Berdasarkan hasil diskusi dengan owner perusahaan, maka dapat diketahui bahwa informasi strategis yang dibutuhkan adalah sebagai berikut:

* 1. Informasi kepopuleran produk yang dipesan oleh konsumen per bulan dalam periode per tahun.
  2. Informasi pemesanan produk yang dilakukan oleh konsumen dalam periode per tahun.
  3. Informasi persentase dalam pelunasan pembayaran terhadap pesanan yang dilakukan oleh konsumen per golongan dalam periode per tahun.
  4. Informasi biaya pengiriman produk yang digunakan untuk mengirim produk ke konsumen per bulan dalam periode per tahun.
  5. Informasi belanja bahan terhadap pemesanan konsumen per produk dan per kredit dalam periode per tahun.
  6. Informasi biaya produksi produk pada pesanan konsumen per staff dan per job dalam periode per tahun.
  7. Informasi kepopuleran desain yang digunakan pada produk per bulan dalam periode per tahun.
  8. Informasi *size* yang telah dipesan per produk setiap bulan dalam periode per tahun.
  9. Informasi kepopuleran produk dalam periode per tahun.
  10. Informasi persentase produk yang kurang populer dalam periode per tahun.
  11. Informasi persentase produk yang populer dalam periode per tahun.
  12. Informasi persentase produk yang sangat populer dalam periode per tahun.
  13. Informasi jumlah suatu produk yang dipesan dengan golongan kurang populer dalam periode per tahun.
  14. Informasi jumlah suatu produk yang dipesan dengan golongan populer dalam periode per tahun.
  15. Informasi jumlah suatu produk yang dipesan dengan golongan sangat populer dalam periode pertahun.
  16. Informasi tiga jenis produk taratas berdasarkan masing-masing klasifikasi dalam periode per tahun.
  17. Informasi tiga konsumen teratas dalam pemesanan jumlah produk dalam periode per tahun.
  18. Informasi tiga konsumen teratas yang memesan produk dengan saldo terbanyak dalam periode per tahun.
  19. Informasi jumlah suatu produk yang dipesan oleh konsumen dalam periode per tahun.
  20. Informasi jumlah dan persentasi pelunasan pada konsumen yang melunasi tepat waktu dalam periode per tahun.
  21. Informasi jumlah dan persentasi pembayaran pemesanan konsumen yang sedikit terlambat dalam periode per tahun.
  22. Informasi jumlah dan persentasi pelunasan pemesanan konsumen yang terlambat dalam periode per tahun.
  23. Informasi jumlah dan persentase pelunasan pemesanan konsumen yang sangat terlambat dalam periode per tahun.
  24. Informasi biaya pengiriman produk dalam periode per jasa kirim per tahun.
  25. Informasi jasa pengiriman dengan biaya terbesar dalam periode per tahun.
  26. Informasi jumlah jasa pengiriman yang digunakan dalam periode per tahun.
  27. Informasi produk dengan belanja bahan terbanyak per kredit dalam periode per tahun.
  28. Informasi jumlah penggunaan kredit dalam belenja bahan per produk dalam periode per tahun.
  29. Informasi saldo kredit yang digunakan dalam belanja bahan per produk dalam periode per tahun.
  30. Informasi biaya produksi produk per job dalam periode per tahun.
  31. Informasi biaya staff per produk perbulan dalam periode per tahun.
  32. Informasi biaya staff per produk dalam periode per tahun.
  33. Informasi persantase desain pada produk yang sangat populer dalam periode per tahun.
  34. Informasi persentase desain pada produk yang populer dalam periode per tahun.
  35. Informasi persentase desain pada produk yang kurang populer dalam periode per tahun.
  36. Informasi jumlah suatu produk yang sangat populer dalam periode per tahun.
  37. Informasi jumlah suatu produk yang populer dalam periode per tahun.
  38. Informasi jumlah suatu produk yang kurang populer dalam periode per tahun.
  39. Informasi jumlah produk yang menggunakan size setiap bulan dalam periode per tahun.
  40. Informasi jumlah *size* yang digunakan pada produk dalam periode per tahun.
  41. Informasi jumlah produk yang menggunakan *size* dalam periode per tahun.

### Analisis Dimensi dan Fakta Bisnis

Berdasarkan pada *point* III.1.3 yang telah membahas kebutuhan informasi strategis, maka dapat dibuat model dimensi yang dibutuhkan dalam data *warehouse*. Tabel III‑22 menjelaskan dimensi bisnis yang dibutuhkan berdasarkan kebutuhan informasi strategisnya.

Tabel III‑22 Dimensi bisnis Perusahaan Spaaceman

|  |  |  |
| --- | --- | --- |
| **1** | **Kebutuhan Informasi Strategis** | Informasi kepopuleran produk yang dipesan oleh konsumen per bulan dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk |
| **2** | **Kebutuhan Informasi Strategis** | Informasi pemesanan produk yang dilakukan oleh konsumen dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pemesanan |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_konsumen 4. Dim\_size |
| **3** | **Kebutuhan Informasi Strategis** | Informasi persentase dalam pelunasan pembayaran terhadap pesanan yang dilakukan oleh konsumen per golongan dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pembayaran |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_konsumen |
| **4** | **Kebutuhan Informasi Strategis** | Informasi biaya pengiriman produk yang digunakan untuk mengirim produk ke konsumen per bulan dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pengiriman |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_jasa\_kirim |
| **5** | **Kebutuhan Informasi Strategis** | Informasi belanja bahan terhadap pemesanan konsumen per produk dan per kredit dalam periode per tahun. |
| **Tabel Fakta** | Fact\_belanja |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_kredit |
| **6** | **Kebutuhan Informasi Strategis** | Informasi biaya produksi produk pada pesanan konsumen per staff dan per job dalam periode per tahun. |
| **Tabel Fakta** | fact\_produksi |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_staff\_produksi 3. Dim\_produk 4. Dim\_job |
| **7** | **Kebutuhan Informasi Strategis** | Informasi kepopuleran desain yang digunakan pada produk per bulan dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_design |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_design 3. Dim\_produk |
| **8** | **Kebutuhan Informasi Strategis** | Informasi *size* yang telah dipesan per produk setiap bulan dalam periode per tahun. |
| **Tabel Fakta** | Fact\_size\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_size 3. Dim\_produk |
| **9** | **Kebutuhan Informasi Strategis** | Informasi kepopuleran produk dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk |
| **10** | **Kebutuhan Informasi Strategis** | Informasi persentase produk yang kurang populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk |
| **11** | **Kebutuhan Informasi Strategis** | Informasi persentase produk yang populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk |
| **12** | **Kebutuhan Informasi Strategis** | Informasi persentase produk yang sangat populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk |
| **13** | **Kebutuhan Informasi Strategis** | Informasi jumlah suatu produk yang dipesan dengan golongan kurang populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk |
| **14** | **Kebutuhan Informasi Strategis** | Informasi jumlah suatu produk yang dipesan dengan golongan populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk |
| **15** | **Kebutuhan Informasi Strategis** | Informasi jumlah suatu produk yang dipesan dengan golongan sangat populer dalam periode pertahun |
| **Tabel Fakta** | Fact\_pop\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk |
| **16** | **Kebutuhan Informasi Strategis** | Informasi tiga jenis produk taratas berdasarkan masing-masing klasifikasi dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk |
| **17** | **Kebutuhan Informasi Strategis** | Informasi tiga konsumen teratas dalam pemesanan produk dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pemesanan |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_konsumen |
| **18** | **Kebutuhan Informasi Strategis** | Informasi tiga konsumen teratas yang memesan produk dengan saldo terbanyak dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pemesanan |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_konsumen |
| **19** | **Kebutuhan Informasi Strategis** | Informasi jumlah suatu produk yang dipesan oleh konsumen dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pemesanan |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_konsumen |
| **20** | **Kebutuhan Informasi Strategis** | Informasi jumlah dan persentasi pelunasan pada konsumen yang melunasi tepat waktu dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pembayaran |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_konsumen |
| **21** | **Kebutuhan Informasi Strategis** | Informasi jumlah dan persentasi pembayaran pemesanan konsumen yang sedikit terlambat dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pembayaran |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_konsumen |
| **22** | **Kebutuhan Informasi Strategis** | Informasi jumlah dan persentasi pelunasan pemesanan konsumen yang terlambat dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pembayaran |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_konsumen |
| **23** | **Kebutuhan Informasi Strategis** | Informasi jumlah dan persentase pelunasan pemesanan konsumen yang sangat terlambat dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pembayaran |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_konsumen |
| **24** | **Kebutuhan Informasi Strategis** | Informasi biaya pengiriman produk dalam periode per jasa kirim per tahun. |
| **Tabel Fakta** | Fact\_pengiriman |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_jasa\_kirim |
| **25** | **Kebutuhan Informasi Strategis** | Informasi jasa pengiriman dengan biaya terbesar dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pengiriman |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_jasa\_kirim |
| **26** | **Kebutuhan Informasi Strategis** | Informasi jumlah jasa pengiriman yang digunakan dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pengiriman |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_jasa\_kirim |
| **27** | **Kebutuhan Informasi Strategis** | Informasi produk dengan belanja bahan terbanyak per kredit dalam periode per tahun. |
| **Tabel Fakta** | Fact\_belanja |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_kredit |
| **28** | **Kebutuhan Informasi Strategis** | Informasi jumlah penggunaan kredit dalam belanja bahan per produk dalam periode per tahun. |
| **Tabel Fakta** | Fact\_belanja |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_kredit |
| **29** | **Kebutuhan Informasi Strategis** | Informasi saldo kredit yang digunakan dalam belanja bahan per produk dalam periode per tahun. |
| **Tabel Fakta** | Fact\_belanja |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_kredit |
| **30** | **Kebutuhan Informasi Strategis** | Informasi biaya produksi produk per job dalam periode per tahun |
| **Tabel Fakta** | Fact\_produksi |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_job 4. Dim\_staff |
| **31** | **Kebutuhan Informasi Strategis** | Informasi biaya staff per produk perbulan dalam periode per tahun. |
| **Tabel Fakta** | Fact\_produksi |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_job 4. Dim\_staff |
| **32** | **Kebutuhan Informasi Strategis** | Informasi biaya staff per produk dalam periode per tahun. |
| **Tabel Fakta** | Fact\_produksi |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_job 4. Dim\_staff |
| **33** | **Kebutuhan Informasi Strategis** | Informasi persantase desain pada produk yang sangat populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_design |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_design |
| **34** | **Kebutuhan Informasi Strategis** | Informasi persentase desain pada produk yang populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_design |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_design |
| **35** | **Kebutuhan Informasi Strategis** | Informasi persentase desain pada produk yang kurang populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_design |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_design |
| **36** | **Kebutuhan Informasi Strategis** | Informasi jumlah suatu produk yang sangat populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_design |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_design |
| **37** | **Kebutuhan Informasi Strategis** | Informasi jumlah suatu produk yang populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_design |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_design |
| **38** | **Kebutuhan Informasi Strategis** | Informasi jumlah suatu produk yang kurang populer dalam periode per tahun. |
| **Tabel Fakta** | Fact\_pop\_design |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_design |
| **39** | **Kebutuhan Informasi Strategis** | Informasi jumlah produk yang menggunakan size setiap bulan dalam periode per tahun. |
| **Tabel Fakta** | Fact\_size\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_size |
| **40** | **Kebutuhan Informasi Strategis** | Informasi jumlah *size* yang digunakan pada produk dalam periode per tahun. |
| **Tabel Fakta** | Fact\_size\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_size |
| **41** | **Kebutuhan Informasi Strategis** | Informasi jumlah produk yang menggunakan *size* dalam periode per tahun. |
| **Tabel Fakta** | Fact\_size\_produk |
| **Tabel Dimensi** | 1. Dim\_waktu 2. Dim\_produk 3. Dim\_size |

Berdasarkan penjelasan yang ada pada Tabel III‑22, dapat diketahui tabel dimensi dan tabel fakta yang akan digunakan dalam sistem ini. Berikut penjelasan tabel dimensi dan tabel fakta yang digunakan.

#### Analisis Dimensi

Dimensi digunakan untuk menjelaskan entitas-entitas bisnis yang akan mendukung suatu fakta sehingga dapat di-analisis secara multidimensional. Berikut penjelasan dimensi yang digunakan dalam sistem:

1. Dim\_design

Dim\_design merupakan dimensi yang menjelaskan jenis design yang diterapkan pada produk. Tabel III‑23 menjelaskan struktur data pada dim\_design.

Tabel III‑23 Struktur data dim\_design

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_design | Integer |  | *Primary Key* |
| Nama\_design | Varchar | 100 | *Not null* |

Dim\_design memiliki hirarki yang dijelaskan pada Tabel III‑24.

Tabel III‑24 Hirarki design

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Nama\_design |
| 2 | No\_design |

1. Dim\_job

Dim\_job merupakan dimensi yang menjelaskan pekerjaan pada tahap produksi yang dilakukan oleh staff produksi. Tabel III‑25 menjelaskan struktur data pada dim\_job.

Tabel III‑25 Sturktur data dim\_job

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_job | Integer |  | *Primary Key* |
| keterangan\_job | Varchar | 50 | *Not null* |

Dim\_job memiliki hiararki yang dijelaskan pada Tabel III‑26**.**

Tabel III‑26 Hirarki job

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Keterangan\_job |
| 2 | No\_job |

1. Dim\_konsumen

Dim\_konsumen merupakan dimensi yang menjelaskan profil konsumen yang telah memesan produk. Tabel III‑27menjelaskan struktur data pada dim\_konusmen.

Tabel III‑27 Struktur data dim\_konsumen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_konsumen | Integer |  | *Primary Key* |
| Nama\_konsumen | Varchar | 50 | *Not null* |
| Nama\_kota | Varchar | 50 | *Not null* |

Dim\_konsumen memiliki hirarki yang dijelaskan pada Tabel III‑28dan Tabel III‑29.

Tabel III‑28 Hirarki kota

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Nama\_kota |
| 2 | Nama\_konsumen |

Tabel III‑29 Hirarki konsumen

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Nama\_konsumen |
| 2 | No\_konsumen |

1. Dim\_kredit

Dim\_kredit merupakan dimensi yang menjelaskan jenis pengeluaran terhadap pembelanjaan produksi. Tabel III‑30 menjelaskan struktur data pada dim\_kredit.

Tabel III‑30 Struktur data dim\_kredit

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_kredit | Integer |  | *Primary Key* |
| Jenis\_kredit | Varchar | 50 | *Not null* |

Dim\_kredit memiliki hirarki yang dijelaskan pada Tabel III‑31.

Tabel III‑31 Hirarki kredit

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Jenis\_kredit |
| 2 | No\_kredit |

1. Dim\_pengiriman

Dim\_pengiriman merupakan dimensi yang menjelaskan jenis pengiriman yang digunakan dalam pengiriman produk ke konsumen. Tabel III‑32 menjelaskan struktur data pada dim\_pengiriman.

Tabel III‑32 Struktur data dim\_pengiriman

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_pengiriman | Integer |  | *Primary Key* |
| Jenis\_pengiriman | Varchar | 100 | *Not null* |

Dim\_pengiriman memiliki hirarki yang dijelaskan pada Tabel III‑33.

Tabel III‑33 Hirarki pengiriman

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Jenis\_pengiriman |
| 2 | No\_pengiriman |

1. Dim\_produk

Dim\_produk merupakan dimensi yang menjelaskan produk yang dapat dipesan konsumen. Tabel III‑34 menjelaskan struktur data pada dim\_produk.

Tabel III‑34 Sturktur data dim\_produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_produk | Integer |  | *Primary Key* |
| Nama\_produk | Varchar | 50 | *Not null* |

Dim\_produk memiliki hirarki yang dijelaskan pada Tabel III‑35.

Tabel III‑35 Hirarki produk

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Nama\_produk |
| 2 | No\_produk |

1. Dim\_size

Dim\_size merupakan dimensi yang menjelaskan ukuran yang digunakan pada produk yang dipesan oleh konsumen berdasarkan jenis kelaminnya. Tabel III‑36menjelaskan struktur data pada dim\_size.

Tabel III‑36 Struktur data dim\_size

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_size | Integer |  | *Primary Key* |
| Nama\_size | Varchar | 10 | *Not null* |

Dim\_size memiliki hirarki yang dijelaskan pada Tabel III‑37.

Tabel III‑37 Hirarki size

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Nama\_size |
| 2 | No\_size |

1. Dim\_staff\_produksi

Dim\_staff\_produksi merupakan dimensi yang menjelaskan profil staff\_produksi yang mengerjkalan pembuatan produk. Tabel III‑38menjelaskan struktur data pada dim\_staff\_produksi.

Tabel III‑38 Struktur data dim\_staff\_produksi

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_staff | Integer |  | *Primary Key* |
| Nama\_staff | Varchar | 50 | *Not null* |
| Nama\_bagian | Varchar | 50 | *Not\_null* |

Dim\_staff\_produksi memiliki hirarki yang dijelaskan pada Tabel III‑39 dan Tabel III‑40.

Tabel III‑39 Hirarki bagian

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Nama\_bagian |
| 2 | Nama\_staff |

Tabel III‑40 Hirarki Staff

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Nama\_staff |
| 2 | No\_staff |

1. Dim\_waktu

Dim\_waktu merupakan dimensi yang menjelaskan waktu, baik itu dalam segi waktu transaksi, belanja, maupun pengiriman. Tabel III‑41 menjelaskan struktur data pada dim\_waktu.

Tabel III‑41 Struktur data dim\_waktu

| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| --- | --- | --- | --- |
| No\_waktu | Integer |  | *Primary Key* |
| Fulldate | Nvarchar | 50 | *Not null* |
| Hari | Nvarchar | 50 | *Not null* |
| Tanggal | Integer |  | *Not null* |
| Nama\_bulan | Nvarchar | 10 | *Not null* |
| Bulan | Integer |  | *Not null* |
| Tahun | Integer |  | *Not null* |

Dim\_waktu memiliki hirarki yang dijelaskan pada Tabel III‑42.

Tabel III‑42 Hirarki waktu

|  |  |
| --- | --- |
| **Level** | **Atribut** |
| 1 | Tahun |
| 2 | Bulan |
| 3 | Tanggal |

#### Analisis Fakta

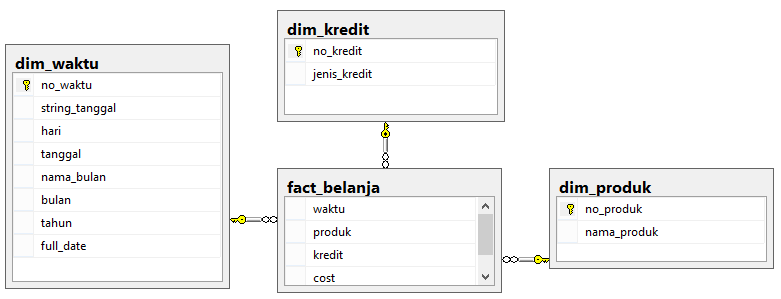
Fakta menjelaskan langsung nilai-nilai yang berkaitan dengan proses bisnis yang bersifat multidimensional agar lebih mudah penggunan untuk menganalisisnya. Berikut penjelasan dan skema fakta yang digunakan dalam sistem ini:

1. Fact\_belanja

Fact\_belanja menjelaskan fakta pengeluaran dalam biaya pemebelian bahan dan alat untuk kepentingan pembuatan produk. Tabel III‑43menjelaskan struktur data pada fact\_belanja dan Gambar III‑2menjelaskan skema fact\_belanja terhadap dimensi yang terkait.

Tabel III‑43 Struktur data fact\_belanja

| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| --- | --- | --- | --- |
| Waktu | Integer |  | *Foreign key* |
| Produk | Integer |  | *Foreign key* |
| Kredit | Integer |  | *Foreign key* |
| Cost | Float |  | *Not null* |



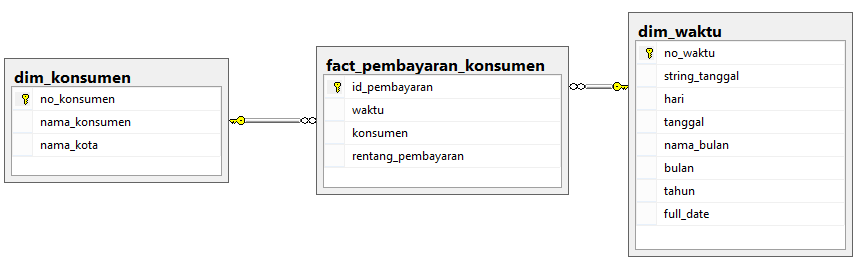
Gambar III‑2 Skema dimensional fakta belanja produksi

1. Fact\_pembayaran

Fact\_pembayaran\_konsumen menjelaskan pengelompokan rentang pembayaran konsumen terhadap tenggat waktu yang ditetapkan dalam skala ke-abuan. Tabel fakta ini diberikan *primary key* untuk mengidentifikasi target atribut yang akan digunakan dalam pemodelan *fuzzy*. Tabel III‑44menjelaskan struktur data pada fakta pembayaran konsumen dan Gambar III‑3 menjelaskan skema fakta terhadap dimensi yang terkait.

Tabel III‑44 Struktur data fact\_pembayaran

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| Id\_pembayaran | Integer |  | *Primary key* |
| Waktu | Integer |  | *Foreign key* |
| Konsumen | Integer |  | *Foreign key* |
| Rentang\_pembayaran | Integer |  |  |



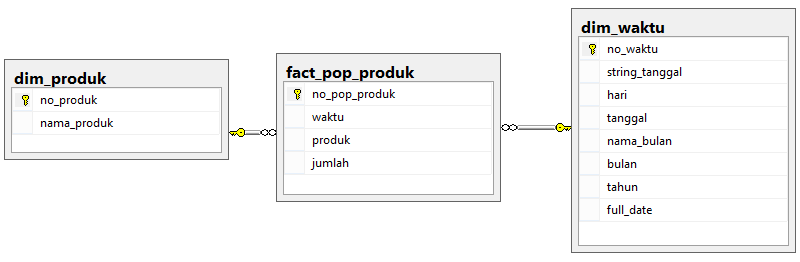
Gambar III‑3 Skema dimensional fakta pembayaran konsumen

1. Fact\_pop\_produk

Fact\_pop\_produk menjelaskan fakta kepopuler produk yang dipesan oleh konsumen. Tabel fakta ini diberikan *primary key* untuk mengidentifikasi target atribut yang akan digunakan dalam pemodelan *fuzzy*. Tabel III‑45menjelaskan struktur data pada fact\_pop\_produk dan Gambar III‑4menjelaskan skema fakta terhadap dimensi yang terkait.

Tabel III‑45 Struktur data fact\_pop\_produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_pop\_produk | integer |  | *Primary key* |
| Waktu | Integer |  | *Foreign key* |
| Produk | Integer |  | *Foreign key* |
| Jumlah | Integer |  | *Not null* |



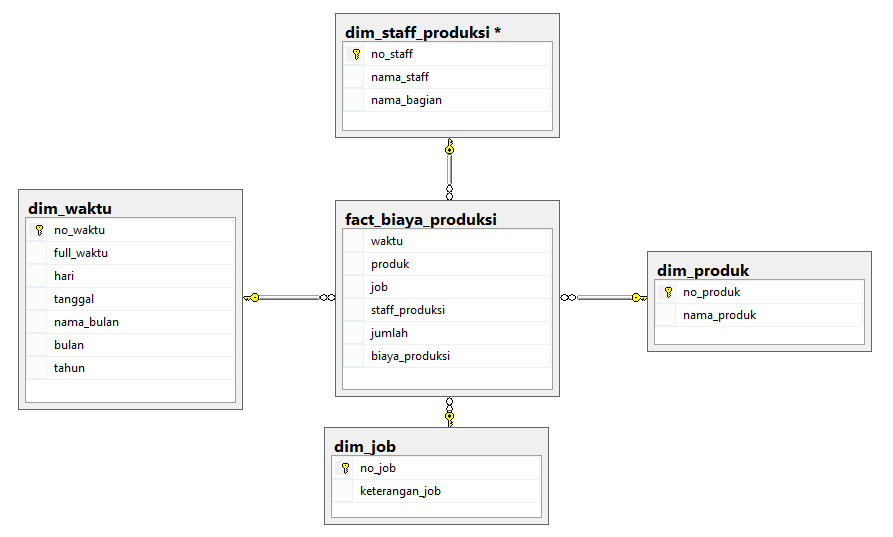
Gambar III‑4 Skema dimensional fakta kepopuleran produk

1. Fact\_produksi

Fact\_biaya\_produksi mejelaskan fakta biaya dalam proses pembuatan produk. Tabel III‑46menjelaskan struktur data pada fact\_biaya\_produksi dan Gambar III‑5menjelaskan skema fact\_biaya\_produksi terhadap dimensi yang terkait.

Tabel III‑46 Struktur data fact\_produksi

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| Waktu | Integer |  | *Foreign key* |
| Produk | Integer |  | *Foreign key* |
| Job | Integer |  | *Foreign key* |
| Staff\_produksi | Integer |  | *Foreign\_key* |
| Jumlah | Integer |  | *Not null* |
| Biaya\_produksi | Float |  | *Not null* |



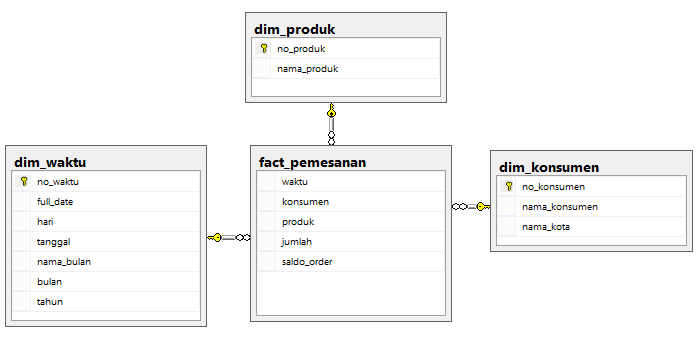
Gambar III‑5 Skema dimensional fakta biaya produksi

1. Fact\_pemesanan

Fact\_pemesanan menjelaskan fakta pemesanan produk yang dilakukan oleh konsumen. Tabel III‑47 menjelaskan struktur data pada fact\_pemesanan dan Gambar III‑6 menjelaskan skema fakta terhadap dimensi yang terkait.

Tabel III‑47 Sturktur data fact\_pemesaan

| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| --- | --- | --- | --- |
| Waktu | Integer |  | *Foreign key* |
| Konsumen | Integer |  | *Foreign key* |
| Produk | Integer |  | *Foreign key* |
| Jumlah | Integer |  | *Not null* |
| Saldo\_order | Float |  | *Not null* |



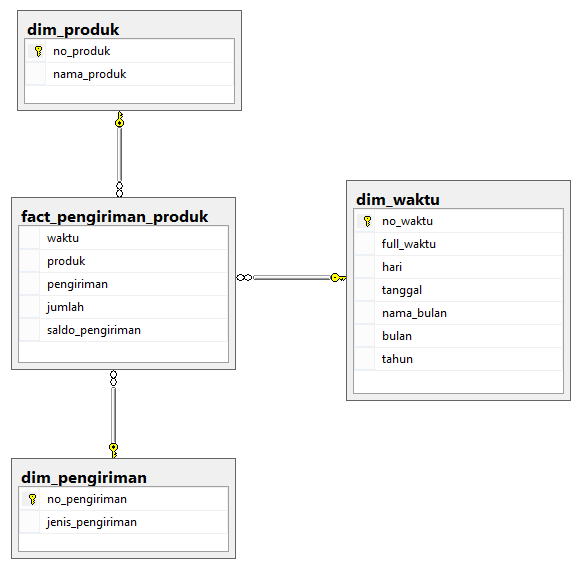
Gambar III‑6 Skema dimensional fakta pemesanan produk

1. Fact\_pengiriman\_produk

Fact\_pengiriman\_produk menjelaskan fakta biaya pengiriman suatu produk ke konsumen berdasarkan jenis pengiriman yang diterapkan. Tabel III‑48menjelaskan struktur data pada fact\_pengiriman\_produk dan Gambar III‑7 menjelaskan skema fakta terhadap dimensi yang tekait.

Tabel III‑48 Sturktur data fact\_pengiriman\_produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| Waktu | Integer |  | *Foreign key* |
| Produk | Integer |  | *Foreign key* |
| Pengiriman | Integer |  | *Foreign key* |
| Jumlah | Float |  | *Not null* |
| Saldo | Float |  | *Not null* |



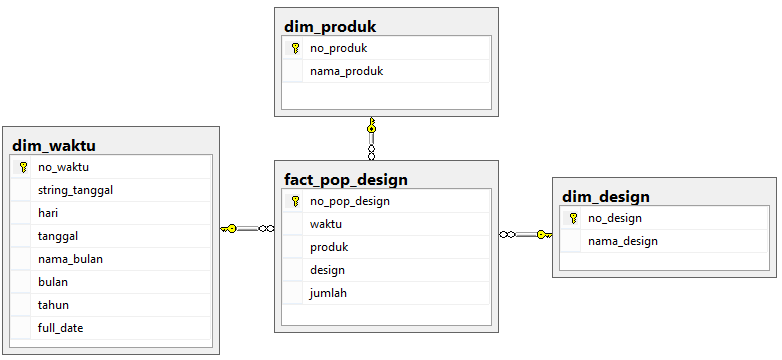
Gambar III‑7 skema dimensional fakta pengiriman produk

1. Fact\_pop\_design

Fact\_pop\_design menjelaskan kepopuleran suatu desain yang digunakan pada produk yang dipesan oleh konsumen yang digolong dalam derajat keanggotaannya. Tabel fakta ini ditambahkan *primary key* untuk mengidentifikasi target atribut yang akan digunakan dalam pemodelan *fuzzy*. Tabel III‑49 menjelaskan struktur data pada fact\_pop\_design dan Gambar III‑8 menjelaskan skema fakta terhadap dimensi yang terkait.

Tabel III‑49 Struktur data pada fact\_pop\_design

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_pop\_design | Integer |  | *Primary key* |
| Waktu | Integer |  | *Foreign key* |
| Produk | Integer |  | *Foreign key* |
| Design | Integer |  | *Foreign key* |
| Jumlah | Integer |  | *Not null* |



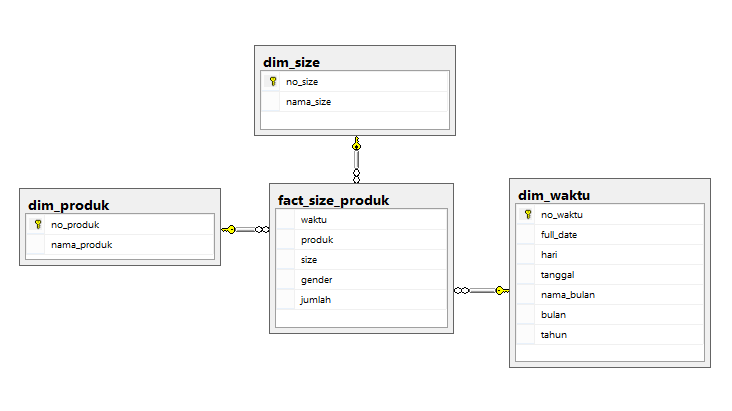
Gambar III‑8 Skema fakta kepopuleran *design* produk

1. Fact\_size\_produk

Fact\_size\_produk menjelaskan fakta ukuran yang digunakan produk pada pemesanan konsumen. Tabel III‑50menjelaskan struktur data pada fact\_size\_produk dan Gambar III‑9menjelaskan skema fakta pada terhadap dimensi yang terkait.

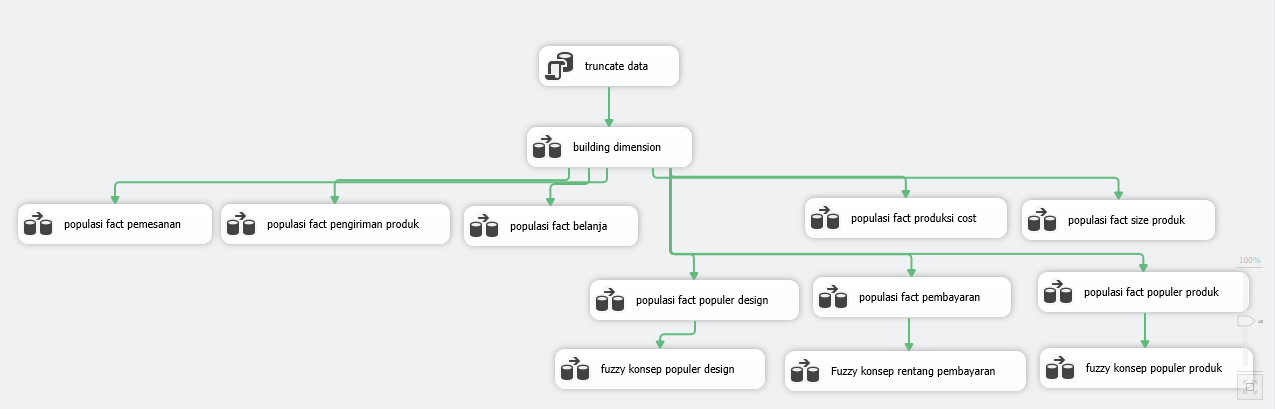
Tabel III‑50 Struktur data fact\_size\_produk

| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| --- | --- | --- | --- |
| Waktu | Integer |  | *Foreign key* |
| Produk | Integer |  | *Foreign key* |
| Size | Integer |  | *Foreign key* |
| Gender | Varchar(10) |  | *Not null* |
| Jumlah | Integer |  | *Not null* |



Gambar III‑9 Skema fakta *size* produk

### Analisis Data *Staging*

Dalam tahapan ini, data operasional (*OLTP*) akan dilakukan proses *data staging*  atau bisa juga disebut dengan proses ekstraksi, transformasi, dan *loading* (*ETL*) ke data *warehouse*. Gambar III‑10 merupakan proses ETL untuk setiap pembuatan tabel dimensi dan populasi tabel fakta dengan menggunakan *Microsoft Visual Studio 2013 : Business Intelligence Project* dengan menggunakan *template* *Business Intelligence Integration Services.*

Gambar III‑10 *Control flow data staging*

Pada Gambar III‑10 merupakan *control flow* dari proses data *staging* yang dilakukan untuk aliran data dari sumber ke tujuan, dan proses konsep *fuzzy* terhadap atribut target. Pada Gambar III‑11merupakan *toolbox* yang digunakan dalam pembuatan *control flow*.



Gambar III‑11 *Tools*  pembuatan *control flow* ETL

*Data flow task tools* merupakan alat untuk mendeskripsikan langkah aliran dari data. Sedangkan *execute SQL task* merupakan serangkaian perintah *SQL* yang dieksekusi. Berikut penjelasan dari setiap *task* yang dilakukan pada proses data *staging*.

1. *Truncate* *data*

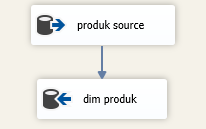
*Task* ini merupakan *execute SQL task* dimana dibangun perintah untuk membersihkan data dengan memperhatikan relasi antar tabel yang saling berhubungan di dalam *database data warehouse*.

1. *Building Dimension*

*Task* ini merupakan *data flow task* yang digunakan untuk mengalirkan data dari data sumber menuju ke tabel dimensi di data *warehouse.* Berikut aliran data masing-masing tabel dimensi dalam data *warehouse*.

1. Dimensi Produk

Dimensi produk dibentuk berdasarkan dari tabel produk yang ada di sumber. Gambar III‑12menggambarkan aliran data yang terjadi dalam dimensi Produk.

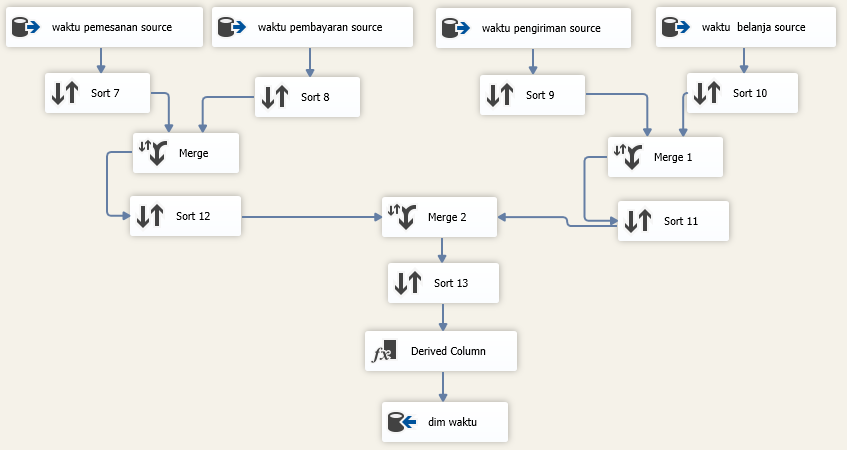


Gambar III‑12 Data *flow* dimensi produk

Adapun *step by step*  dari data *flow* dimensi produk ini adalah sebagai berikut:

1. Sumber data diambil dari tabel produk.
2. Data di-*extract* dengan memilih atribut yang akan digunakan didalam dimensi produk.
3. Hasil *extract* di petakan ke *destination* yaitu dimensi produk.
4. Dimensi Waktu

Dimensi waktu dibentuk sedemikian rupa berdasarkan dari tabel sumber yang memiliki atribut waktu. Masing-masing data di ekstraksi, transformasi, dan di muat ke dimensi waktu. Gambar III‑13menggambarkan aliran data yang terjadi pada dimensi waktu.

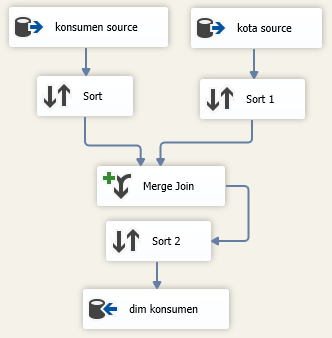


Gambar III‑13 Data *flow* dimensi waktu

Adapun *step by step* dalam aliran data menuju dimensi waktu adalah sebagai berikut.

1. Mengekstraksi data dari sumber yaitu, tabel pemesanan, tabel pembayaran, tabel pengiriman, dana tabel belanja.
2. Masing-masing data dari tabel digabung menjadi satu.
3. Mengecek data dan mensorting data, di tahapan ini data yang bersifat redudansi akan dihapus dan akan di ambil satu sebagai perwakilan data.
4. Melakukan transformasi data dengan pemecahan, *fulldate*, tanggal, bulan, dan tahun.
5. Hasil transformasi di muat ke dalam dimensi waktu.
6. Dimensi Konsumen

Dimensi konsumen dibentuk dari tabel konsumen dan kota sumber. Gambar III‑14 menggambarkan aliran data yang terjadi pada dimensi konsumen.

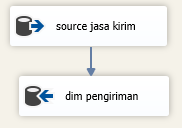


Gambar III‑14 Data *flow* dimensi konsumen

Adapun *step by step*  dalam aliran data menuju dimensi konsumen adalah sebagai berikut:

1. Mengktraksi data dari sumber yaitu tabel konsumen dan tabel kota
2. Melakukan penggabungan antara tabel konsumen dengan tabel kota berdasarkan identitas relasinya.
3. Memuat data ke dimensi konsumen.
4. Dimensi pengiriman

Dimensi pengiriman dibentuk dari tabel jasa\_kirim. Gambar III‑15 menggambarkan aliran data yang terjadi pada dimensi pengiriman.

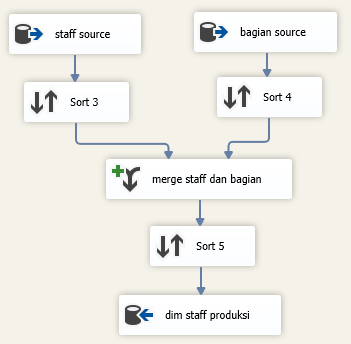


Gambar III‑15 Data *flow* dimensi pengiriman

Adapun *step by step* dari aliran menuju dimensi pengiriman ini adalah sebagai berikut:

1. Mengktraksi data dari sumber yaitu tabel metode pengiriman.
2. Data yang akan digunakan dimasukkan ke dalam dimensi pengiriman.
3. Dimensi Staff Produksi

Dimensi staff produksi dibentuk dari tabel staff produksi dan tabel bagian dari sumber. Gambar III‑16 menggambarkan aliran data yang terjadi pada dimensi staff produksi.

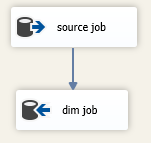


Gambar III‑16 Data *flow* dimensi staff produksi

Adapun *step by step*  aliran data menuju dimensi staff produksi adalah sebagai berikut:

1. Mengktraksi data staff produksi dan bagian dari sumber.
2. Menggabungkan data staff dan bagian berdasarkan relasinya.
3. Memuat data ke dimensi staff produksi.
4. Dimensi *Job*

Dimesi *job* dibentuk dari tabel job yang ada di sumber. Gambar III‑17menggambarkan aliran data yang terjadi pada dimensi job.

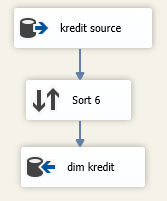


Gambar III‑17 Data *flow* dimensi *job*

Adapun *step by step* aliran data menuju dimensi *job* adalah sebagai berikut:

1. Mengkstrak data dari tabel job sumber.
2. Menentukan dan memuat atribut yang digunakan untuk dimuat ke dalam dimensi *job*.
3. Dimensi Kredit

Dimensi kredit dibentuk dari tabel kredit yang ada di sumber. Gambar III‑18 menggambarkan aliran data yang terjadi di dimensi kredit.

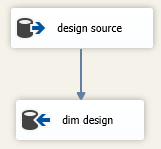


Gambar III‑18 Data *flow* dimensi kredit

Adapun *step by step* aliran data menuju dimensi kredit adalah sebagai berikut:

1. Mengkstrak data dari tabel kredit.
2. Membersihkan data dari redudansi.
3. Memuat data ke dimensi kredit.
4. Dimensi *Design*

Dimensi *design* dibentuk dari tabel design yang ada disumber. Gambar III‑19 menggambarkan aliran data yang terjadi di dimensi design.

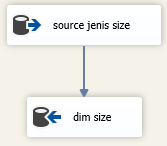


Gambar III‑19 *Data flow* dimensi *design*

Adapun *step by step* aliran data menuju dimensi *design* adalah sebagai berikut:

1. Mengkstrak data design dari sumber.
2. Menentukan dan memuat data ke dalam dimensi *design*.
3. Dimensi *Size*

Dimensi *size* dibentuk dari tabel size yang ada di sumber. Gambar III‑20menggambarkan aliran data yang terjadi pada dimensi size.



Gambar III‑20 *Data flow* dimensi *size*

Adapun *step by step* aliran data menuju dimensi *size* adalah sebagai berikut:

1. Mengkestrak data dari sumber.
2. Menentukan dan memuat data ke dimensi *size*.
3. Populasi Fakta

*Task* ini merupakan *data flow task* yang digunakan untuk mempopulasikan data dari data sumber menuju ke tabel fakta di *data warehouse.* Hal ini dilakukan untuk mendapatkan nilai bisnis yang dilakukan di tabel sumber untuk dijadikan nilai bisnis di tabel fakta. Selain itu, tabel fakta juga di bentuk berdasarkan hasil *lookup*  dari tabel dimensi yang memiliki relasi dengan tabel fakta agar mendapatkan kunci dai tabel dimensi yang akan dijadikan sebagai kunci tamu di tabel fakta. Berikut aliran data masing-masing tabel fakta dalam data *warehouse.*

1. Fakta Pemesanan Produk

Fakta pemesanan produk menunjukkan bisnis pemesanan produk yang terbentuk dari tabel pemesanan dan tabel *order service* yang ada di sumber dan hasil *lookup* dari tabel dimensi yang telah dibentuk. Gambar III‑21 menggambarkan aliran data yang terjadi di fakta pemesanan produk. Adapun *step by step* aliran data menuju fakta pemesanan produk adalah sebagai berikut:

1. Mengekstrak data dari tabel pemesanan dan tabel *service order*.
2. Menggabungkan kedua tabel berdasarkan relasinya
3. Melakukan *derived kolom* waktu menjadi data yang cocok untuk dilakukan *lookup*.
4. Melakukan *lookup* ke dimensi waktu, dimensi konsumen, dan dimensi produk.
5. Memuat data hasil *lookup*  ke dalam tabel fakta pemesanan produk.
6. Fakta Pengiriman Produk

Fakta pengiriman produk menunjukkan bisnis pengiriman suatu produk yang terbentuk dari tabel pengiriman, pemesanan, *order service,* dan metode pengiriman. Gambar III‑22 menggambarkan aliran data yang terjadi pada fakta pengiriman produk. Adapun *step by step*  aliran data menuju fakta pengiriman produk adalah sebagai berikut:

1. Mengekstrak data dari tabel pengiriman, pemesanan, *order service*, dan metode pengiriman.
2. Menggabungkan masing-masing tabel berdasarkan relasinya.
3. Melakukan *derive* terhadap tanggal pengiriman
4. Melakukan *lookup* data pada dimensi waktu, dan dimensi produk.
5. Memuat data hasil *lookup* ke fakta pengiriman produk.
6. Fakta Belanja

Fakta belanja menunjukkan bisnis pengeluaran perusahan dalam pembelian alat dan bahan yang terbentuk dari tabel detail belanja, belanja, kredit, dan *order service*. Gambar III‑23 menggambarkan aliran data yang terhadi pada fakta belanja. Adapun *step by step* aliran data menuju fakta belanja adalah sebagai berikut:

1. Mengekstrak data dari tabel detail belanja, belanja, kredit, dan *service order*.
2. Menggabungkan seluruh tabel berdasarkan dengan relasinya.
3. Melakukan *derive* waktu dan nominal.
4. Melakukan *lookup* data pada dimensi waktu, dimensi produk, dan dimensi kredit.
5. Memuat data hasil *lookup* ke fakta belanja.
6. Fakta Biaya Produksi

Fakta biaya produksi menunjukkan bisnis biaya sumber daya terhadap proses produksi yang terbantuk dari tabel produksi, *order service*, dan pemesanan. Gambar III‑24 menggambarkan aliran data yang terjadi pada fakta biaya produksi. Adapun *Step by step* aliran data menuju fakta biaya produksi adalah sebagai berikut:

1. Mengekstrak data dari tabel produksi, *order service*, dan pemesanan
2. Menggabungkan data dari masing-masing tabel berdasarkan relasinya.
3. Melakukan *derive* waktu.
4. Melakukan *lookup* data pada dimensi waktu, dimensi produk, dimensi job, dan dimensi staff produksi.
5. Memuat data hasil *lookup* ke fakta biaya produksi.
6. Fakta Populer Produk

Fakta populer produk menujukkan bisnis kepopuleran suatu produk terhadap konsumen yang terbentuk dari tabel *service order* dan tabel pemesanan. Fakta yang dihasilkan dari tabel fakta ini akan dijadikan sebagai atribut target dalam konsep *fuzzy.* Gambar III‑25 menunjukkan aliran data yang terjadi pada fakta populer produk. Adapun *step by step* aliran datanya adalah sebagai berikut:

1. Mengkstrak data dari tabel *order service*  dan pemesanan.
2. Menggabungkan data dari masing-masing tabel berdasarkan relasinya.
3. Melakukan *derive* waktu.
4. Melakukan *lookup* data pada dimensi waktu dan dimensi produk.
5. Memuat data hasil *lookup* ke fakta populer produk.
6. Fakta Populer Design

Fakta populer design menunjukkan bisnis kepopuleran suatu design yang di gambarkan pada produk terhadap konsumen yang terbentuk dari tabel *design order*,*service order,* dan pemesanan. Fakta yang dihasilkan pada tabel fakta ini akan dijadikan sebagai target atribut dalam penggunaan konsp *fuzzy*. Gambar III‑26 menggambarkan aliran data yang terjadi pada fakta kepopuleran produk. Adapun *step by step* aliran data fakta kepopuleran produk adalah sebagai berikut:

1. Mengekstrak data dari tabel *design order, order service,* dan pemesanan.
2. Menggabungkan data dari masing-masing tabel berdasarkan relasinya.
3. Melakukan *derive* waktu.
4. Melakukan *lookup* data pada dimensi waktu, dimensi produk, dimensi size.
5. Memuat data hasil *lookup* ke fakta kepopuleran produk.
6. Fakta Pembayaran

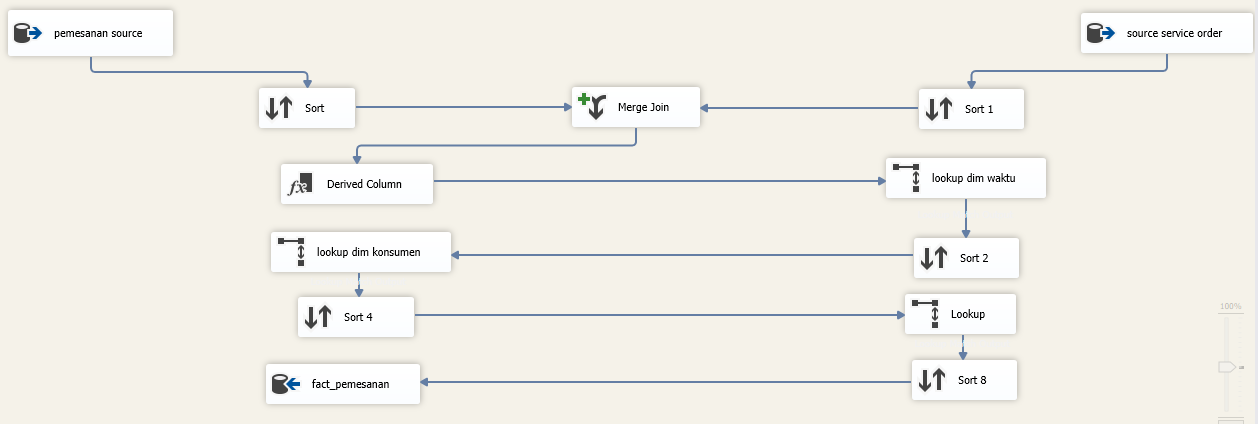
Fakta pembayaran menunjukkan bisnis pada rentang pemabayaran konsumen yang terbentuk dari tabel pemesanan, dan pembayaran. Gambar III‑27 menunjukkan aliran data yang terjadi pada fakta pembayaran. Adapun *stepby step* aliran datanya adalah sebagai berikut:

1. Mengkstrak data dari tabel pemesanan, dan pembayaran.
2. Menggabungkan data dari masing-masing tabel berdasarkan relasinya.
3. Melakukan *derive* waktu.
4. Melakukan *lookup* data pada dimensi waktu dan dimensi konsumen.
5. Melakukan *derive* rentang pembayaran.
6. Memuat hasil data ke fakta pembayaran.
7. Fakta size Produk

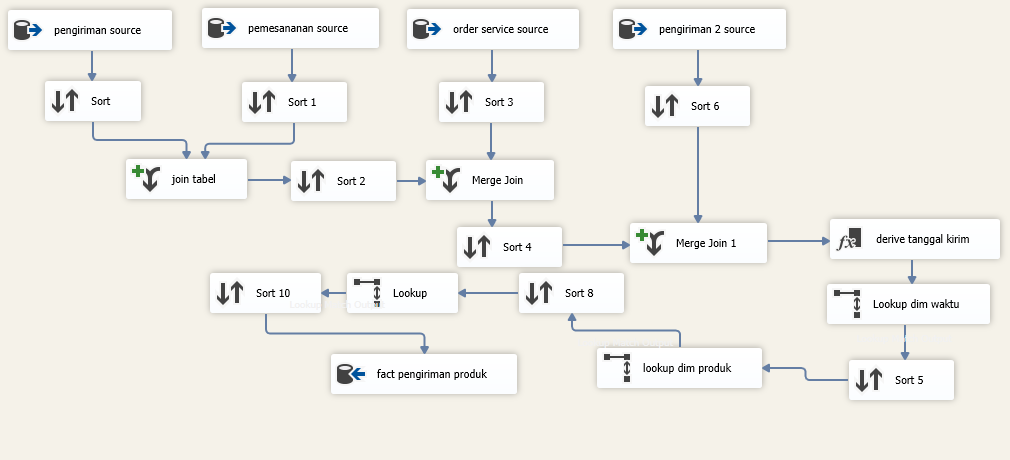
Fakta size produk menunjukkan bisnis ukuran yang digunakan produk yang tebentuk dari tabel *size order, order service,* dan pemesanan. Gambar III‑28 menggambarkan aliran data yang terjadi pada fakta size produk. Adapun *step by step* aliran data menuju fakta size produk adalah sebagai berikut:

1. Mengekstrak data dari tabel *size order, order service,* dan pemesanan.
2. Menggabungkan data pada masing-masing tabel berdasarkan relasinya.
3. Melakukan *derive*  waktu.
4. Melakukan *lookup* data pada dimensi waktu, dimensi produk, dan dimensi size.
5. Memuat data hasil *lookup* ke fakta size produk.

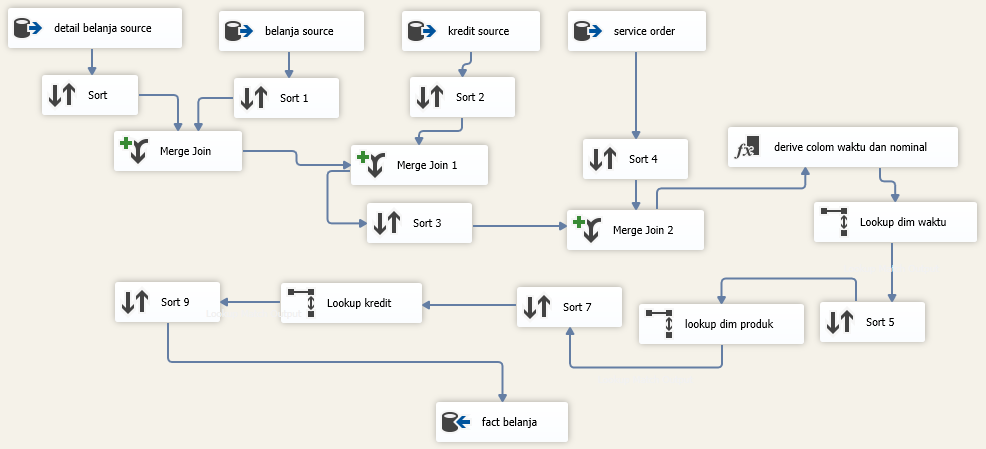
Pada proses *data staging* dalam suatu *data warehouse* akan berhenti disini, tetapi pada Gambar III‑10 setelah mendefinisikan aliran data pada tabel fakta, beberapa tabel fakta memiliki *data flow task*  yaitu konsep *fuzzy. Task* *fuzzy* konsep yang ada pada Gambar III‑10 akan dibahas pada *point*III.2.2.



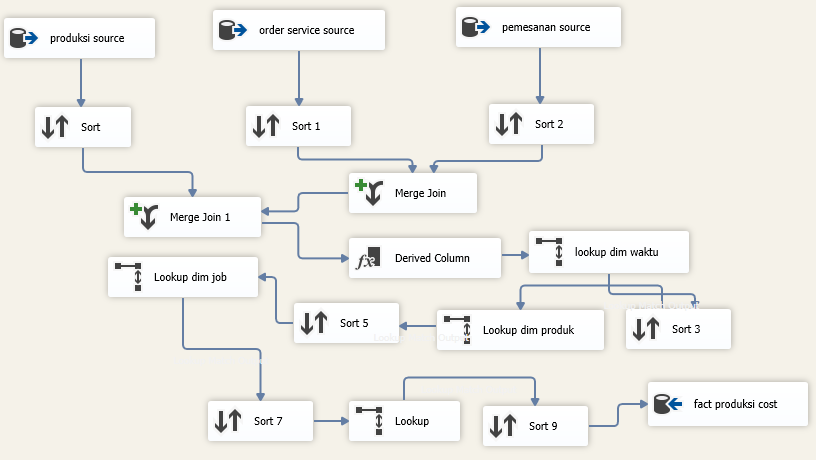
Gambar III‑21 Data *flow* fakta pemesanan



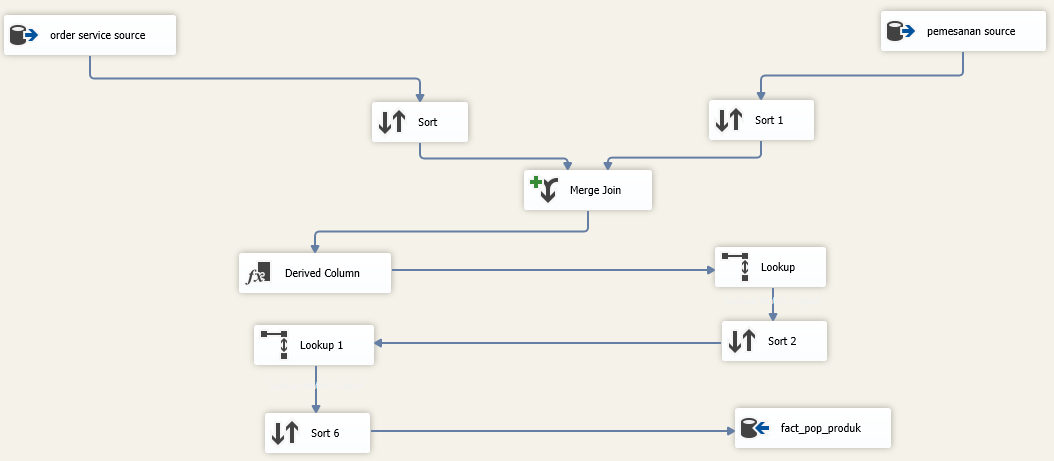
Gambar III‑22 *Data flow* fakta pengiriman produk



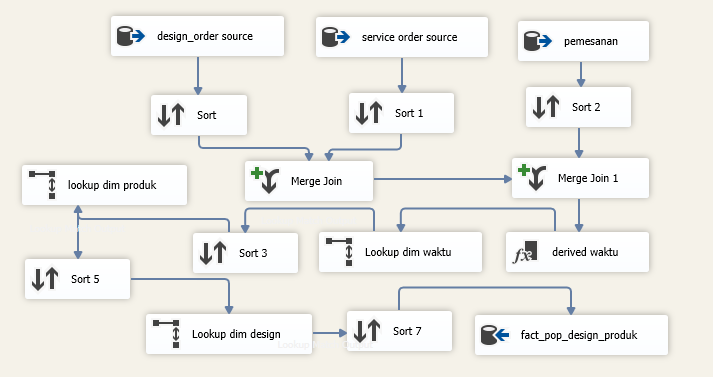
Gambar III‑23 Data *flow* fakta belanja



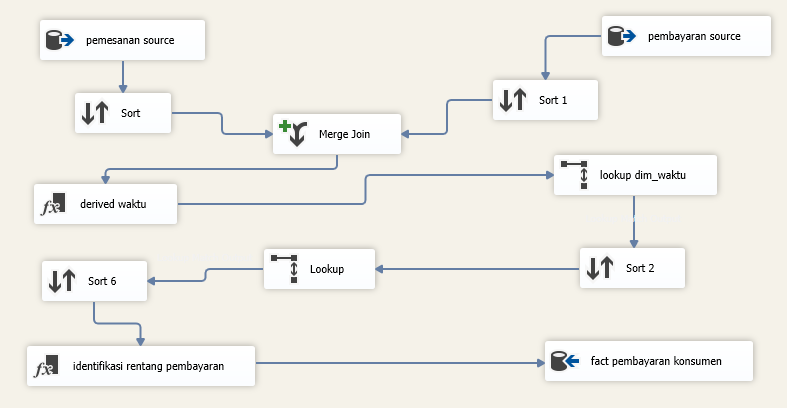
Gambar III‑24 Data *flow* fakta biaya produksi



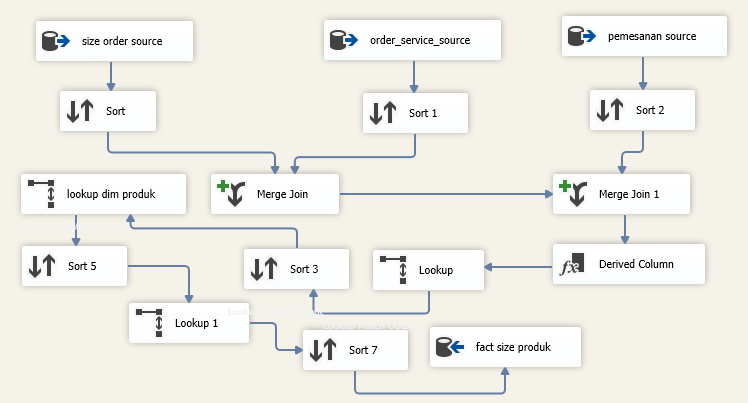
Gambar III‑25 Data *flow* fakta kepopuleran produk



Gambar III‑26 Data *flow* fakta kepopuleran design produk



Gambar III‑27 *Data flow* fakta pembayaran



Gambar III‑28 *Data flow* fakta *size* produk

## Analisis Konsep *Fuzzy*

Analisis konsep *fuzzy* digunakan untuk menjelaskan pemodelan konsep *fuzzy y*ang menghasilkan derajat keanggotaan berdasarkan nilai domain dari atribut target yang diklasifikasikan.

### Elemen Pengklasifikasian

Pada *point* III.1.4telah didefinisikan analisis yang menghasilkan tabel dimensi dan fakta. Adapun yang dijadikan target atribut yang dibutuhkan untk menganalisis konsep *fuzzy* ini adalah sebagai berikut.

1. Fakta jumlah produk dari tabel fact\_pop\_produk.
2. Fakta jumlah design dari tabel fact\_pop\_design.
3. Fakta rentang pembayaran dari tabel fact\_pembayaran.

Setelah menentukan target atribut, maka selanjutnya menentukan kelas yang digunakan untuk pngklasifikasian. Target atribut untuk fakta jumlah dari tabel fact\_pop\_produk pengklasifikasian yang digunakan terdiri dari kurang populer, populer, dan sangat populer. Bila jumlah suatu produk dipetakan dalam bentuk persentase terhadap jumlah masing-masing produk yang dipesan oleh konsumen dipetakan dalam *open end fuzzy concept*, maka dapat dibuat himpunannya seperti berikut:

1. Himpunan *fuzzy* kurang populer = 0 - 36
2. Himpunan *fuzzy* populer = 20 – 60
3. Himpunan *fuzzy* sangat populer = 44 – ∞

Setalah menentukan kelas yang digunakan, maka selanjutnya menentukan fungsi keanggotaan yang digunakan untuk memetakan derajat keanggotaannya. Adapun fungsi keanggotannya adalah sebagai berikut:

Keterangan :

*x =* Target atribut.

P = Pemetaan kurva menaik.

R = Pemetaan kurva menurun.

µ = Derajat keanggotan.

Berdasarkan fungsi keanggotaannya dapat dibangun kurva pemetaan kepopuleran suatu produk dalam periode waktu tertentu. Gambar III‑29 menggambarkan kurva pemetaan keanggotaan *fuzzy* kepopuleran produk.

Jumlah pemesanan produk dalam periode tertentu

µ

AA

B

C

Gambar III‑29 Kurva pemetaan himpunan fuzzy jumlah produk

Keterangan :

A = Himpunan *fuzzy* kurang populer

B = Himpunan *fuzzy* populer

C = Himpunan *fuzzy* sangat populer

Sebagai contoh, Tabel III‑51 memetakan pemesanan produk dalam *Order Service* pada suatu transaksi sebagai berikut:

Tabel III‑51 Pemesanan produk oleh konsumen

| **Produk** | **Jumlah** |
| --- | --- |
| Polo | 25 |
| T-shirt | 52 |
| Kemeja | 24 |
| Jacket Jumper | 12 |
| Jacket Parasit | 12 |
| Jacket Jeans | 0 |
| Total | 125 |

Jadi menurut fungsi keanggotaannya, maka setiap produk akan dipetakan sebagai berikut:

Tabel III‑52 Kepopuleran produk berdasarkan pemetaan fungsi himpunan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Produk** | **Jumlah** | **Kurang Populer** | **Cukup Populer** | **Sangat Populer** |
| Polo | 25 | 0.69 | 0.31 | 0 |
| T-shirt | 52 | 0 | 0.5 | 0.5 |
| Kemeja | 24 | 0.75 | 0.25 | 0 |
| Jacket Jumper | 12 | 0.75 | 0 | 0 |
| Jacket Parasit | 12 | 0.75 | 0 | 0 |
| Jacket Jeans | 0 | 0 | 0 | 0 |

Tabel III‑52merupakan penjelasan hasil penerapan konsep *fuzzy* sehingga dapat di-analisis berdasarkan kepopuleran suatu produk berdasarkan pemesanan konsumen. Pemetaan fungsi keanggotan berdasarkan klasifikasinya dapat juga diterapkan dalam pencarian fakta kepopuleran suatu design dalam produk pada tabel fact\_pop\_design dimana target atributnya adalah fakta jumlah design. Sedangkan untuk penerapan fuzzy dimensional pembayaran konsumen, perlu di buat himpunan keanggotaan yang lain berdasarkan rentang waktu dari tanggal pembayaran dengan *deadline* pemesanan, berikut penjelasan himpunan keanggotaan untuk persentasi pembayaran konsumen.

1. Tepat waktu = -∞ – 10
2. Sedikit terlambat = 7 - 20
3. Terlambat = 16 – 30
4. Sangat terlambat = 25 - ∞

Bedasarkan himpunan keangotaan , dapat dibuat fungsi keanggotaannya sebagai berikut:

Berdasarkan fungsi keanggotaannya dapat dibangun kurva pemetaan persentasi pembayaran konsumen terhadap pemesanan yang dilakukan. Berikut Gambar III‑30menggambarkan kurva pemetaan keanggotaan *fuzzy* pembayaran konsumen.

Rentang pembayaran kostumer

µ

a

b

c

d

Gambar III‑30 Kurva pemetaan himpunan *fuzzy* pembayaran konsumen

Keterangan :

a = Pembayaran konsumen tepat waktu

b = Pembayaran konsumen sedikit terlambat

c = Pembayaran konsumen terlambat

d = Pembayaran konsumen sangat terlambat

### Pemodelan *Fuzzy Data Warehouse*

Pada *point* III.2.1, telah di definiskan kelas, fungsi dan pemetaan domain terhadap fungsi himpunannya, sehingga dapat digunakan untuk menerapkan konsep *fuzzy*. Berdasarkan metode pemodelan *fuzzy data warehouse*, maka setiap tabel yang memiliki target atribut akan diberikan kunci utama untuk mengidentifikasi target atribut dan membangun tabel konsep *fuzzy* untuk memuat derajat keanggotaan dari hasil pengklasifikasian nilai domain target atribut. Dengan menggunakan tipe pemodelan *different membership degrees for the same linguistic terms,*  maka dapat diketahui target atribut dapat memiliki banyak kelas dalam pengklasifisian.Tabel III‑53**,** Tabel III‑54**,** dan Tabel III‑55 menjelaskan struktur data dari tabel konsep *fuzzy* terhadap masing-masing target atribut.

Tabel III‑53 Struktur data Tabel konsep *fuzzy* kepopuleran produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_md\_pop\_produk | Integer |  | *Primary key* |
| Ta\_pop\_produk | Integer |  | *Foreign key ( reference* dari tabel target atribut ) |
| Ca\_pop\_produk | Nvarchar | 15 | Hanya bisa diisi   * Kurang Populer * Populer * Sangat Populer |
| Md\_pop\_produk | Float |  | Nilai derajat keanggotaan |

Tabel III‑54 Struktur data tabel konsep *fuzzy* kepopuleran *design*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| No\_md\_pop\_design | Integer |  | *Primary key* |
| Ta\_pop\_design | Integer |  | *Foreign key ( reference* dari tabel target atribut ) |
| Ca\_pop\_design | Nvarchar | 15 | Hanya bisa diisi   * Kurang Populer * Populer * Sangat Populer |
| Md\_pop\_design | Float |  | Nilai derajat keanggotaan |

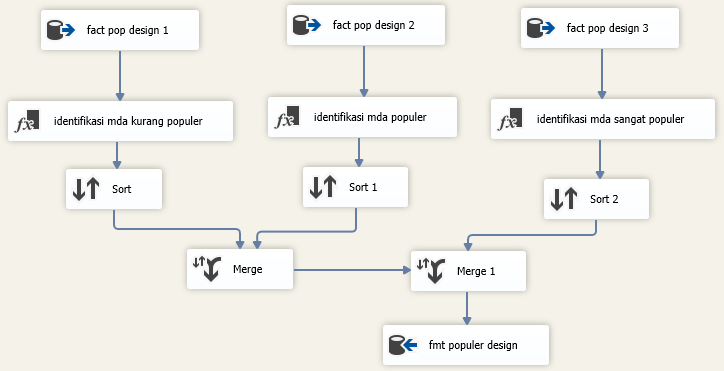
Tabel III‑55 Struktur data tabel konsep *fuzzy* pembayaran konsumen

| **Nama Field** | **Tipe Data** | **Panjang** | **Keterangan** |
| --- | --- | --- | --- |
| No\_md\_pembayaran | Integer |  | *Primary key* |
| Ta\_pembayaran | Integer |  | *Foreign key ( reference* dari tabel target atribut ) |
| Ca\_pembayaran | Nvarchar | 20 | Hanya bisa diisi   * Tepat Waktu * Sedikit Terlambat * Terlambat * Sangat Terlambat |
| Md\_pembayaran | Float |  | Nilai derajat keanggotaan |

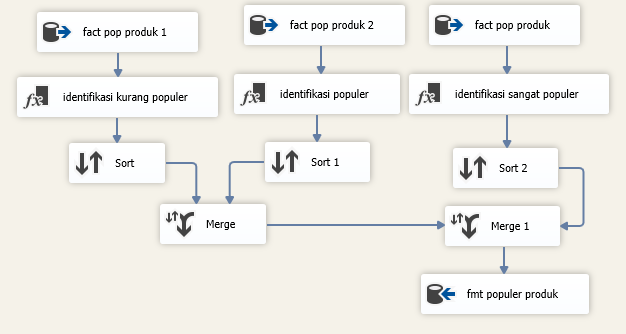
Berdasarkan pada *control flow*  yang terdapat pada Gambar III‑10yang menggambarkan aliran data dari target atribut menuju *fuzzy* konsep. Adapun *step by step* umumnya adalah sebagai berikut:

1. Mengekstrat target atribut.
2. Melakukan *derive* berdasarkan kelas dan fungsi keanggotaannya
3. Menggabungkan hasil *derive*.
4. Memuat data ke tujuan.

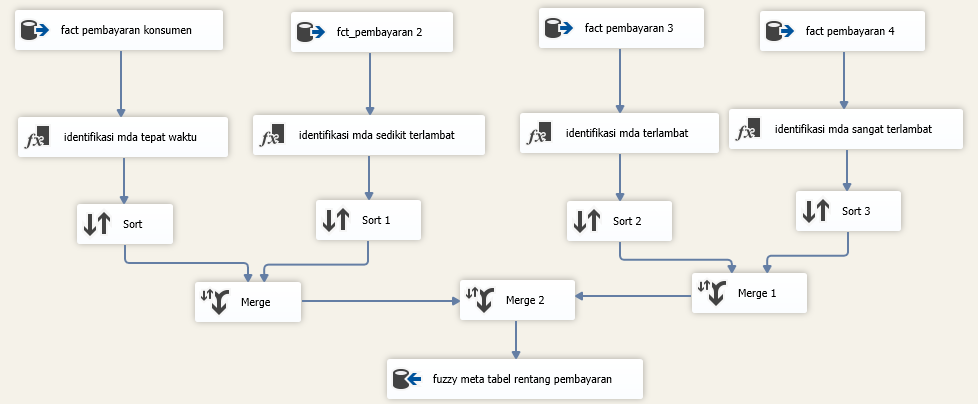
Berdasarkan Gambar III‑10**,**  maka penjelasan aliran data dan kalkulasi terhadap himpunan keanggotan sehingga mendapatkan derajat keanggotaannya pada masing-masing target atribut digambarkan pada Gambar III‑31**,** Gambar III‑32,dan Gambar III‑33.



Gambar III‑31 *Data flow fuzzy* konsep populer desain



Gambar III‑32 *Data flow fuzzy* konsep populer produk



Gambar III‑33 *Data flow fuzzy* konsep rentang pembayaran

## Analisis OLAP

Begitu selesai data staging dan konsep *fuzzy*, maka *fuzzy* data *warehouse* telah terbentuk dan siap di lakukan analisis *OLAP*. Analisis *OLAP* merupakan analisis yang dilakukan untuk menentukan bentuk *OLAP* yang sesuai dalam penunjang pengambilan keputusan. OLAP mempresentasikan banyak data dalam bentuk multidimensi agar menjadi lebih mudah untuk melakukan analisis terhadap suatu informasi. Dalam penelitian ini, Metode OLAP yang digunakan adalah metode *Pivoting* yang memungkinkan pengguna untuk dapat melihat suatu nilai dalam tata letak yang berbeda-beda demi kepentingan proses analisis dan penunjang pengambilan keputusan.

## Analisis Kebutuhan

Analisis kebutuhan digunakan untuk menjelaskan spesifikasi yang dibutuhkan dalam pembangunan perangkat lunak dan batasan-batasan dalam pengimplementasian sistem. Tabel III‑56menjelaskan spesifikasi kebutuhan fungsional pada sistem yang dibangun dan Tabel III‑57menjelaskan spesifikasi kebutuhan non-fungsional.

Tabel III‑56 Spesifikasi Kebutuhan Fungsional

|  |  |
| --- | --- |
| **Kode Kebutuhan** | **Keterangan** |
| SKPL-F-1 | User dapat melakukan proses ETL di dalam sistem. |
| SKPL-F-2 | User dapat membuat OLAP cube sesuai keinginan. |
| SKPL-F-3 | User dapat membuat laporan |
| SKPL-F-4 | User dapat melakukan pencetakan |
| SKPL-F-5 | User dapat menampilkan dimensi dan fakta |

Tabel III‑57 Spesifikasi Kebutuhan Non-Fungsional

| **Kode Kebutuhan** | **Keterangan** |
| --- | --- |
| SKPL-NF-1-1 | Proses Ekstraksi pada data dilakukan secara otomatis didalam sistem saat user melakukan proses *ETL*. |
| SKPL-NF-1-2 | Proses Transformasi pada data dilakukan secara otomatis didalam sistem saat user melakukan proses *ETL*. |
| SKPL-NF-1-3 | Menggunakan *threading* untuk menampilkan loading dalam proses *ETL*. |
| SKPL-NF-2-1 | Proses pembuatan *cube* *OLAP* dilakukan secara otomatis dalam penyesuaian kebutuhan *user*. |
| SKPL-NF-3-1 | Laporan yang di tampilkan oleh user sesuai dengan *cube OLAP* yang dilihat oleh *user*. |
| SKPL-NF-4-1 | Data yang diprint menggunakan *tamplate* yang telah ditentukan sesuai kebutuhan user. |
| SKPL-NF-5-1 | Batasan data yang ditampilkan sebanyak sepuluh baris |
| SKPL-NF-5-2 | Baris yang lebih dari sepulah akan di tampilkan dengan menggunakan *paging control*. |
| SKPL-NF-5-3 | Status *control paging* akan menampilkan indeks baris awal dana khir terhadap data yang ditampilkan |

### Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dibutuhkan untuk dapat menentukan fitur-fitur dari spesifikasi yang dibutuhkan oleh sistem. Spesifikasi ini meliputi elemen atau perangkat yang dibutuhkan untuk sistem yang dibangun sampai sistem siap diimplementasikan.

#### Analisis Perangkat Keras

Aplikasi berjalan dengan optimal apabila didukung dengan perangkat keras yang memadai. Spesifikasi perangkat keras dalam sistem yang dibangun adalah sebagai berikut:

Tabel III‑58 Spesifikasi perangkat keras yang dibutuhkan

| **No** | **Perangkat keras** | **Spesifikasi** |
| --- | --- | --- |
| 1 | Processor | 2.0 GHz |
| 2 | RAM | 2 GB |
| 3 | Hardisk | 128 GB |
| 4 | VGA | 128 bit kapasitas 512MB |
| 5 | Keyboard | Standard |
| 6 | Mouse | Standard |

Sedangkan spesifikasi perangkat keras yang ada saat ini adalah sebagai berikut:

Tabel III‑59 Spesifikasi perangkat keras yang ada saat ini

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat keras** | **Spesifikasi** |
| 1 | Processor | 3.2 GHz |
| 2 | RAM | 4 GB |
| 3 | Hardisk | 1024 GB |
| 4 | VGA | 128 bit kapasitas 2GB |
| 5 | Keyboard | Standard |
| 6 | Mouse | Standard |

Berdasarkan perbandingan analisis perangkat keras yang dibutuhkan dengan analisis perangkat keras yang ada saat ini, maka dapat disimpulkan bahwa perangkat keras yang telah ada saat ini telah memadai kebutuhan spesifikasi perangkat keras untuk menjalani sistem yang dibangun.

#### Analisis Perangkat Lunak

Analisis perangkat lunak digunakan untuk memenuhi spesifikasi perangkat lunak yang dibutuhkan untuk mendukung dalam pengimpelemntasian sistem yang dibangun. Adapun kebutuhn minimum perangkat lunak pendukung yang dibutuhkan untuk mendukung sistem adalah sebagai berikut:

Tabel III‑60 Spesifikasi perangkat lunak yang dibutuhkan

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat lunak** | **Spesifikasi** |
| 1 | Sistem Operasi | Windows 7 |
| 2 | Sql Server | Sql server 2014 |
| 3 | Runtime.net | Runtime.net 3.0,3.5,4.0,4.5 |

Sedangkan spesifikasi perangkat lunak yang ada saat ini adalah sebagai berikut.

Tabel III‑61 Spesifikasi perangkat lunak yang ada saat ini

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat lunak** | **Spesifikasi** |
| 1 | Sistem Operasi | Windows 8.1 |
| 2 | Sql Server | - |
| 3 | Runtime.net | Runtime.4.5 |

Berdasarkan perbandingan analisis dari perangkat lunak yang dibutuhkan dengan perangkat lunak yang ada saat ini, dibutuhkannya pembaharauan fitur perangkat lunak *framework* *runtime.net*  dari versi 3.0 hingga versi 4.5 dan aplikasi SQL *server 2014* untuk digunakan sebagai *database* sistem.

#### Analisis Perangkat Pikir

Analisis perangkat pikir digunakan untuk mempersiapkan perangkat pikir yang pakar dalam menggunakan sistem yang dibangun agar sistem dapat berjalan dengan optimal. Adapun spesifikasi kebutuhan perangkat pikir pada sistem ini dipetakan pada Tabel III‑62**.** Sedangkan perangkat pikir yang ada saat ini dipetakan pada Tabel III‑63.

Tabel III‑62 Spesifikasi kebutuhan perangkat prikir

|  |  |
| --- | --- |
| **Pengguna Sistem** | Owner |
| **Tingkat Keterampilan yang harus dimiliki** | 1. Mengoperasikan komputer 2. Memahami konsep database secara dasar. 3. Menganalisa data 4. Pengoperasian komputer |
| **Pengalam yang harus dimiliki** | 1. Menggunakan komputer 2. Menggunakan sistem dengan konsep database 3. Menggunakan sistem *warehousing* |

Tabel III‑63 Spesifikasi perangkat pikir yang ada

|  |  |
| --- | --- |
| **Pengguna Sistem** | Owner |
| **Tingkat Keterampilan yang harus dimiliki** | 1. Terbiasa melakukan pengoperasian pada komputer. 2. Memahami konsep database 3. Menganalisa data 4. Pengoperasian komputer |
| **Pengalam yang harus dimiliki** | 1. Menggunakan sistem dengan konsep database. 2. Menggunakan komputer |

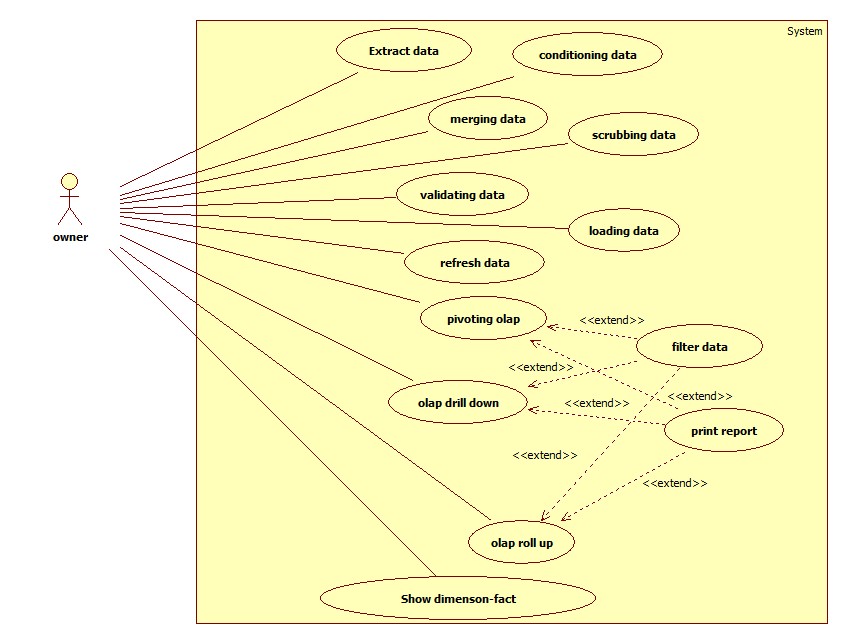
Berdasarkan perbandingan Tabel III‑62dan Tabel III‑63, dapat disimpulkan penggunan sistem harus melakukan pelatihan tentang konsep sistem warehousing agar dapat menggunakan sistem yang dibangun secara optimal.

### Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk memodelkan fungsi sistem yang dibangun. Dalam penelitian ini, pemodelan dalam pembangunan sistem menggunakan *OOAD( Object Oriented Analysis and Design)*.

#### *Use Case Diagram*

*Use Case Diagram* digunakan dalam pemodelan untuk menjelaskan terhadap apa yang akan dilakukan oleh pengguna di dalam sistem atau disebut dengan fungsional yang dimiliki oleh sistem. Gambar III‑34menggambarkan *use case diagram* pada fungsional sistem yang akan dibangun pada perusahaan Spaceman dimana fungsionalnya meliputi proses *ETL*, *OLAP*, dan menampilkan data dimensi atau fakta pada sistem data *warehouse* ini.



Gambar III‑34 *Use case diagram data warehouse* pada perusahaan Spaceman

#### *Use Case Scenario*

*Use case scenario* digunakan untuk menjelaskan setiap *use case*  yang akan dilakukan oleh *user*. *Use case scenario* juga menjelaskan tahapan tahapan terhadap proses yang dilakukan oleh *user*,kondisi awal sebelum melakukan proses, kondisi sukses atau gagalnya suatu proses, dan pemicu terhadap proses yang dijalankan. Berikut merupakan *use case scenario* dari sistem yang dibangun.

1. *Extract data*

*Use case scenario*  pada *use case* *extract data* menjelaskan proses, kondisi dan *user* pada saat melakukan proses *extrcat data* dan menjelaskan langkah-langkah proses *ETL* yang dideskripsikan pada Tabel III‑64.

Tabel III‑64 *Use case scenario extract* data

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Extract* data | |
| ***Goal In Context*** | Sistem mengekstrak data dari data *source*. | |
| ***Preconditions*** | Sistem memiliki data *source* dan data *destination* | |
| ***Successful End Condition*** | Sistem berhasil mengekstraksi data dari data *source* | |
| ***Failed End Condition*** | Sistem gagal memproses *ETL* | |
| ***Primary Actors*** | *User* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *User* meminta sistem melakukan proses *ETL* | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem membaca data *source*. |
| 2 | Sistem mengekstraksi data |
| 3 | Data berhasl di *extract* |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 1.1 | Sistem tidak menemukan data *source*. |
| 2.2 | Sistem membatalkan proses ETL |
| 2.3 | Menampilkan pesan status proses *failure*. |

2. *Conditioning* data

*Use case scenario*  pada *use case* *conditioning* datamenjelaskan proses, kondisi dan *user* pada saat melakukan proses *conditioning* datadan menjelaskan langkah-langkah proses *conditioning* datayang dideskripsikan pada Tabel III‑65.

Tabel III‑65 *Use case scenario conditioning* data

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Conditioning* data | |
| ***Goal In Context*** | Data *source* diubah menjadi tipe yang dibutuhkan. | |
| ***Preconditions*** | *Data flow task*  membutuhkan *deriving* pada data *source*. | |
| ***Successful End Condition*** | Sistem berhasil melakukan *conditioning* data. | |
| ***Failed End Condition*** | Melanjutkan data *flow task* selanjutnya. | |
| ***Primary Actors*** | *User* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *User* melakukan proses *ETL*. | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem membaca data hasil *extract*. |
| 2 | Cek data *flow task* untuk melakukan *conditioning* data. |
| 3 | Merubah tipe data menjadi tipe yang baru. |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 2.1 | Sistem tidak menemukan data yang akan di ubah tipenya. |
| 2.2 | Melanjutkan data *flow task* selanjutnya. |

3. *Merging* data

*Use case scenario*  pada *use case* *merging* datamenjelaskan proses, kondisi dan *user* pada saat melakukan proses *merging* datadan menjelaskan langkah-langkah proses *merging* datayang dideskripsikan pada Tabel III‑66.

Tabel III‑66 *Use case scenario merging* data

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Merging* data | |
| ***Goal In Context*** | Menggabung data hasil *extract*. | |
| ***Preconditions*** | *Data flow task*  membutuhkan penggabungan data dari data *source*. | |
| ***Successful End Condition*** | Sistem berhasil menggabungkan data. | |
| ***Failed End Condition*** | melanjutkan data *flow task*  selanjutnya. | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* melakukan proses *ETL*. | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem membaca data hasil *extract*. |
| 2 | Cek *data flow task*  untuk melakukan *merging* data |
| 3 | Sistem melakukan *merging* data. |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 2.1 | data *flow task* tidak memerlukan *merging* data. |
| 2.2 | melanjtukan data *flow task* selanjutnya. |

4. *Scrubbing* data

*Use case scenario*  pada *use case* *scrubbing* datamenjelaskan proses, kondisi dan *user* pada saat melakukan proses *scrubbing* data dan menjelaskan langkah-langkah proses *scrubbing* datayang dideskripsikan pada Tabel III‑67.

Tabel III‑67 *Use case scenario scrubbing* data

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Scrubbing* data | |
| ***Goal In Context*** | Membersihkan data hasil *extract*. | |
| ***Preconditions*** | *Data flow task* membutuhkan *scrubbing* data. | |
| ***Successful End Condition*** | Data berhasil dibersihkan. | |
| ***Failed End Condition*** | Melanjutkan data *flow task* selanjtunya. | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* melakukan proses *ETL* | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem membaca data hasil *extract*. |
| 2 | Cek *data flow task* untuk melakukan *scrubbing* data*.* |
| 3 | Data dibersihkan. |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 2.1 | Data *flow task* tidak memerlukan *scrubbing* data. |
| 2.2 | Melanjutkan data *flow task* selanjtunya. |

5. *Validating* data

*Use case scenario*  pada *use case* *validating* datamenjelaskan proses, kondisi dan *user* pada saat melakukan proses *validating* dan menjelaskan langkahlangkah proses *validating*  data yang dideskripsikan pada Tabel III‑68.

Tabel III‑68 *Use case scenario validating* data

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Validating* data | |
| ***Goal In Context*** | Pemeriksaan data terhadap hasil penggabungan. | |
| ***Preconditions*** | Data *flow task*  melakukan proses penggabungan data. | |
| ***Successful End Condition*** | Data berhasil divalidasi. | |
| ***Failed End Condition*** | Melanjutkan data *flow task*  selanjutnya. | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* melakukan proses *ETL* | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem membaca hasil penggabungan data. |
| 2 | Cek validasi data. |
| 3 | Melakukan pemerikasaan data. |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 2.1 | Data *flow task* tidak membutuhkan validasi data. |
| 1.2 | Melanjutkan data *flow task* selanjutnya. |

6*. Loading* data

*Use case scenario*  pada *use case* *loading*  datamenjelaskan proses, kondisi dan *user* pada saat melakukan proses *loading* data dan menjelaskan langkah-langkah proses *loading* datayang dideskripsikan pada Tabel III‑69.

Tabel III‑69 *Use case scenario loading* data

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Loading* data | |
| ***Goal In Context*** | Memuat data ke tabel *destination*. | |
| ***Preconditions*** | *Extracting*  dan *transformaing*  data telah dilakukan. | |
| ***Successful End Condition*** | Sistem berhasil memuat data. | |
| ***Failed End Condition*** | Sistem gagal memuat data. | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* melakukan proses *ETL*. | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem mengecek tabel *destination*. |
| 2 | Data dimuat ke dalam tabel *destination*. |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 1.1 | Terjadi kesalahan terhada tabel *destination*. |
| 2.2 | Menampilkan pesan  *failure*. |

7. *Refresh* data.

*Use case scenario*  pada *use case* *refresh*  datamenjelaskan proses, kondisi dan *user* pada saat melakukan proses *refresh* data dan menjelaskan langkah-langkah proses *refresh* datayang dideskripsikan pada Tabel III‑70.

Tabel III‑70 *use case scenario refresh* data

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Refresh* data | |
| ***Goal In Context*** | Merefresh data di *destination.* | |
| ***Preconditions*** | *Owner* telah melakukan proses *ETL* sebelumnya. | |
| ***Successful End Condition*** | Berahasil melakukan *refreshing* data | |
| ***Failed End Condition*** | Gagal melakukan *refresh* data. | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* melakukan proses *ETL*. | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem melakukan *extract* data dan *mentransforming*  data dari data *source* |
| 2 | Menampilkan pesan sukses *refresh*  data. |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 1.1 | Terjadi kesalahan terhadap *extract* data atau *transforming* data. |
| 1.2 | Menampilkan pesan  *failure*. |

8. *Pivoting OLAP*.

*U`se case scenario*  pada *use case* *pivoting OLAP* menjelaskan proses, kondisi dan *user* pada saat melakukan proses *pivoting OLAP* dan menjelaskan langkah-langkah proses *pivoting OLAP* yang dideskripsikan pada Tabel III‑71.

Tabel III‑71 *use case scenario pivoting OLAP*

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Pivoting OLAP* | |
| ***Goal In Context*** | *Refresh*  *cube*  data dengan metode *pivoting* | |
| ***Preconditions*** | Proses ETL telah sukses dilakukan | |
| ***Successful End Condition*** | Berhasil melakukan *pivoting OLAP* | |
| ***Failed End Condition*** | Gagal melakukan *pivoting OLAP*. | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* melakukan proses *ETL*. | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Melakukan *generate* *pivot OLAP* |
| 2 | Menampilkan pesan sukses |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 1.1 | Terjadi kesalahan terhadap *generate pivot* |
| 1.2 | Menampilkan pesan  *failure*. |

9. *OLAP drill down*.

*Use case scenario*  pada *use case* *OLAP drill down* menjelaskan proses, kondisi dan *user* pada saat melakukan proses *OLAP drill down* dan menjelaskan langkah-langkah proses *OLAP dril down* yang dideskripsikan pada Tabel III‑72.

Tabel III‑72 *use case scenario OLAP drill down*

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *OLAP drill down* | |
| ***Goal In Context*** | *Menampilkan child cube*  pada suatu dimensi | |
| ***Preconditions*** | *Owner* telah melakukan *pivoting OLAP* | |
| ***Successful End Condition*** | Menampilkan *child cube* | |
| ***Failed End Condition*** | Gagal menampilkan *child cube* | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* melakukan proses *drill down* | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem membaca *child cube* berdasrkan hirarki data. |
| 2 | Menampilkan data *child cube*. |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 1.1 | Sistem tidak menemukan *child cube* |
| 1.2 | Gagal menampilkan *child cube* |

10. O*LAP roll up*.

*U`se case scenario*  pada *use case*  *OLAP roll up* menjelaskan proses, kondisi dan *user* pada saat melakukan proses *OLAP roll up* dan menjelaskan langkah-langkah proses *OLAP roll up* yang dideskripsikan pada Tabel III‑73.

Tabel III‑73 *use case scenario OLAP roll up*

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *OLAP roll up* | |
| ***Goal In Context*** | Menyembunyikan *child cube* terhadap *parent* | |
| ***Preconditions*** | Melakukan proses *drill down*. | |
| ***Successful End Condition*** | Berhasil melakukan *pivoting OLAP* | |
| ***Failed End Condition*** | Gagal melakukan *pivoting OLAP*. | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* melakukan proses *roll up* | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem menyembunyikan *child cube*. |
| 2 | Menampilkan *parent cube*. |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 1.1 | Gagal menyembunyikan *child cube*. |
| 1.2 | Proses dibatalkan |

11. *filter* data.

*U`se case scenario*  pada *use case* *filter data* menjelaskan proses, kondisi dan *user* pada saat melakukan proses *filter* data dan menjelaskan langkah-langkah proses *filter* datayang dideskripsikan pada Tabel III‑74.

Tabel III‑74 *use case scenario filter* data

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Filter* data | |
| ***Goal In Context*** | Menampilkan data hasil *filter* | |
| ***Preconditions*** | Menampilkan *OLAP* | |
| ***Successful End Condition*** | Menampilkan olap hasil filtering | |
| ***Failed End Condition*** | Menampilkan *Olap* awal | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* memilih pilihan *filtering*. | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | User memilih *cube* yang ingin di filter. |
| 2 | Melakukan pivoting terhadap *flitering cube*. |
|  | 3 | Menampilkan *OLAP* hasil *filtering*. |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 2.1 | Gagal melakukan *filtering*. |
| 2.2 | Menampilkan *OLAP* awal. |

12. *Print report*.

*U`se case scenario*  pada *use case* *print report* menjelaskan proses, kondisi dan *user* pada saat melakukan proses *print report* dan menjelaskan langkah-langkah proses *print report* yang dideskripsikan pada Tabel III‑75.

Tabel III‑75 *use case scenario print report*

|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *print report* | |
| ***Goal In Context*** | Mencetak laporan | |
| ***Preconditions*** | Memiliki *template* | |
| ***Successful End Condition*** | Mencetak *template* | |
| ***Failed End Condition*** | Membatalkan proses | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | *Owner* mengklik pencetakan | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | *Generate template* dari *OLAP* |
| 2 | Mencetak laporan |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 2.1 | Gagal men-*generate* *template* |
| 2.2 | Membatalkan proses |

13. *Show dimension-fact*

*Use case scenario* pada *use case* *show dimension-fact* menjelaskan proses menampilkan data yang diolah dalam data *warehouse* dan menjelaskan langkah-langkah proses *show dimension-fact* yang dideskripsikan pada Tabel III‑76.

Tabel III‑76 *Use case scenario*  *show dimension-fact*

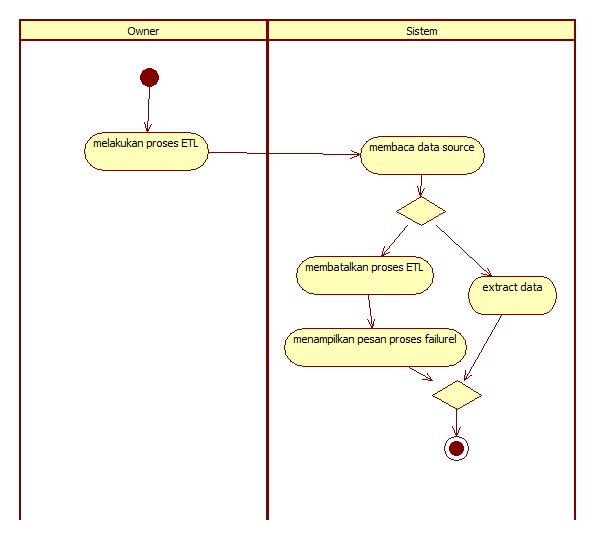
|  |  |  |
| --- | --- | --- |
| ***Use Case name*** | *Show dimension-fact* | |
| ***Goal In Context*** | Sistem dapat menampilkan data dimensi dan fakta. | |
| ***Preconditions*** | Sistem telah melakukan proses *ETL* | |
| ***Successful End Condition*** | Sistem menampilkan data dimensi | |
| ***Failed End Condition*** | Sistem menampilkan data fakta | |
| ***Primary Actors*** | *Owner* | |
| ***Secondary Actors*** | - | |
| ***Trigger*** | Sistem melakukan proses penampilan data | |
| ***Main Flow*** | ***Step*** | ***Action*** |
| 1 | Sistem memproses penampilan data oleh permintaan pengguna. |
| 2 | Menampilkan data dimensi |
| ***Extensions*** | ***Step*** | ***Branching Action*** |
| 1.1 | Menampilkan data fakta |

#### *Activity Diagram*

*Activity diagram* digunakan untuk memodelkan alur aktivitas dari suatu *use case.* Berikut merupakan *activity* *diagram*  yang merepresentasikan alur aktivitas *use case* pada sistem yang dibangun.

1. *Activity diagram extract* data

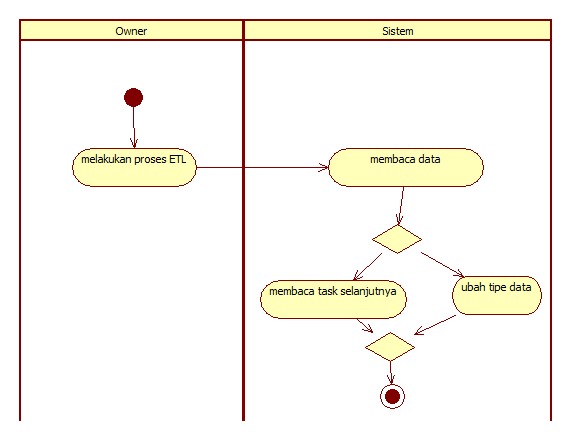
Gambar III‑35merupakan diagram alur aktivitas yang di jelaskan dalam *use case scenario extract* data.



Gambar III‑35 *Activity diagram* proses *extract data*

2*. Activity diagram conditioning* data

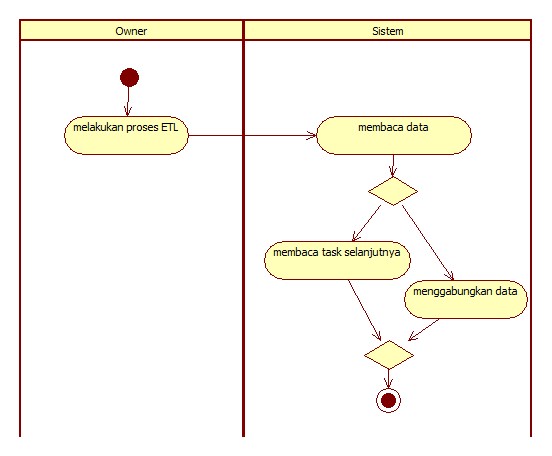
Gambar III‑36menggambarkan diagram alur aktivitas yang di jelaskan dalam *use case scenario conditioning* data.



Gambar III‑36 *Activity diagram conditioning* data

3. *Activity diagram merging* data

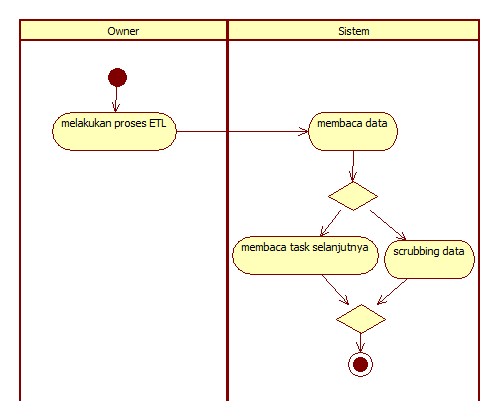
Gambar III‑37menggambarkan diagram alur aktivitas yang di jelaskan dalam *use case scenario merging* data*.*



Gambar III‑37 *Activity diagram merging* data

4. *Activity diagram scrubbing* data

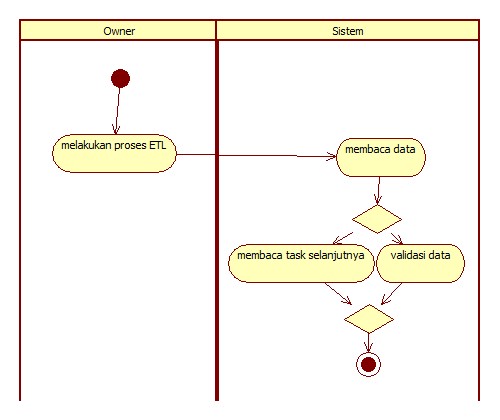
Gambar III‑38menggambarkan diagram alur aktivitas yang di jelaskan dalam *use case scenario scrubbing*  data*.*



Gambar III‑38 *Activity diagram scrubbing data*

5. *Activity diagram validating* data

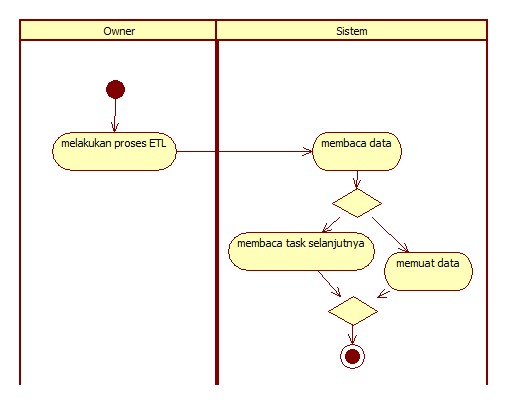
Gambar III‑39menggambarkan diagram alur aktivitas yang di jelaskan dalam *use case scenario validating* data*.*



Gambar III‑39 *Activiy diagram validating* data

6. *Activity diagram loading* data

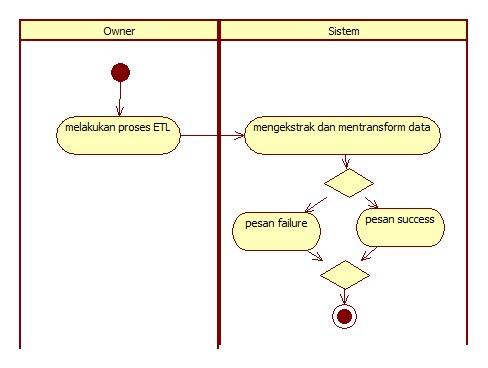
Gambar III‑40merupakan diagram alur aktivitas yang di jelaskan dalam *use case scenario loading* data*.*



Gambar III‑40 *Activity diagram loading* data

7. *Activity diagram refresh* data

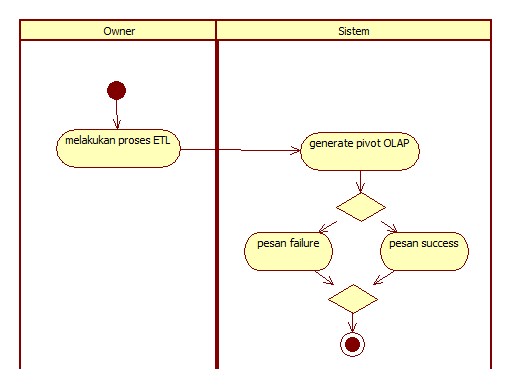
Gambar III‑41merupakan diagram alur aktivitas yang di jelaskan dalam *use case refresh* data*.*



Gambar III‑41 *Activity diagram refresh* data

8. *Activity diagram pivoting OLAP*

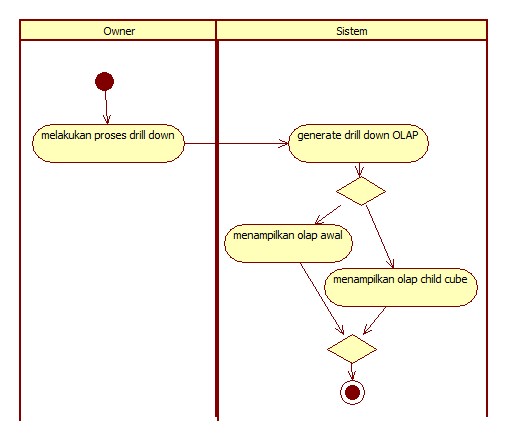
Gambar III‑42merupakan diagram alur aktivitas yang di jelaskan dalam *use case pivoting OLAP.*



Gambar III‑42 *Activity diagram pivoting OLAP*

1. *Activity diagram OLAP drill down*

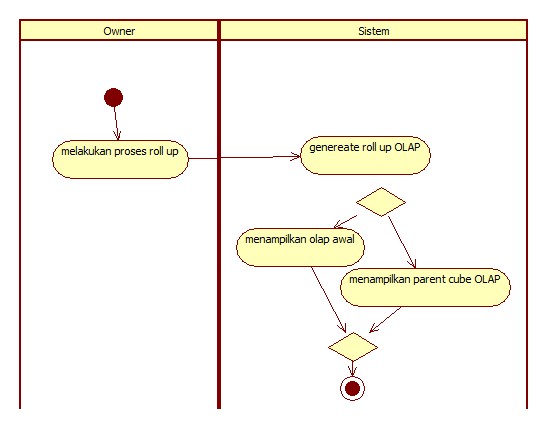
Gambar III‑43merupakan diagram alur aktivitas yang di jelaskan dalam *use case OLAP drill down.*



Gambar III‑43 *Activity diagram OLAP drill down*

1. *Activity diagram OLAP roll up*

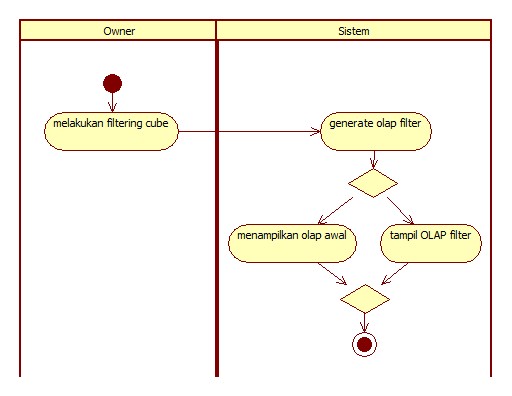
Gambar III‑44merupakan diagram alur aktivitas yang di jelaskan dalam *use case OLAP roll up.*

**

Gambar III‑44 *Activity diagram OLAP roll up*

1. *Activity diagram filter* data

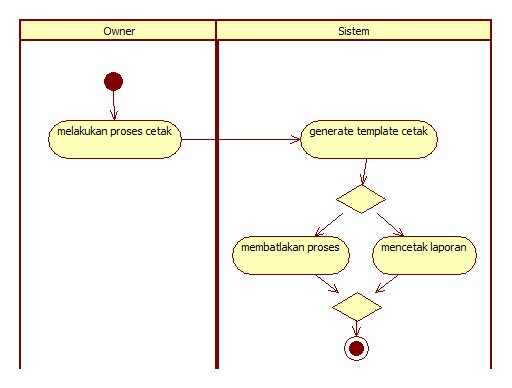
Gambar III‑45merupakan diagram alur aktivitas yang di jelaskan dalam *use case filter* data.

**

Gambar III‑45 *Activity diagram filter* data

1. *Activity diagram print report*

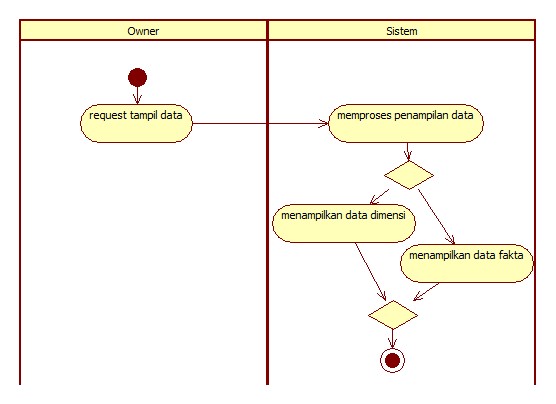
Gambar III‑46merupakan diagram alur aktivitas yang di jelaskan dalam *use case filter* data.



Gambar III‑46 *Activity diagram print report*

1. *Activity diagram show dimension-fact*

Gambar III‑47merupakan diagram alur aktivitas yang di jelaskan dalam *use case show dimension-fact.*



Gambar III‑47 *Activity diagram show dimension-fact*

#### Class Diagram

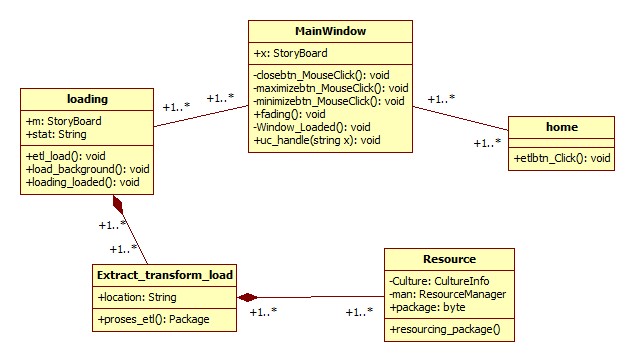
*Class diagram,* merupakan tahap pemodelan dalam merealisasikan *use case* yang ada pada **Gambar 43** kedalam kelas yang dibutuhkan untuk guna menjalankan proses yang ada didalam *use case*. Analisis ini juga menggambarkan keseluruhan kelas yang akan digunakan didalam sistem. Berikut diagram kelas berdasrkan proses dan *use case* yang menggunakannya.

1. *ETL*

*ETL* merupakan diagram kelas yang mendefinisikan kumpulan kelas yang digunakan pada fungsional sistem di *use case* yang digambarkan pada Gambar III‑34. Adapun fungsional pada *use case* yang menggunakan diagram kelas ini adalah:

* + - * 1. *Use case Extract* data.
        2. *Use case conditioning* data.
        3. *Use case merging* data.
        4. *Use case scrubbing* data.
        5. *Use case validating* data.
        6. *Use case loading* data.
        7. *Use case refresh* data.

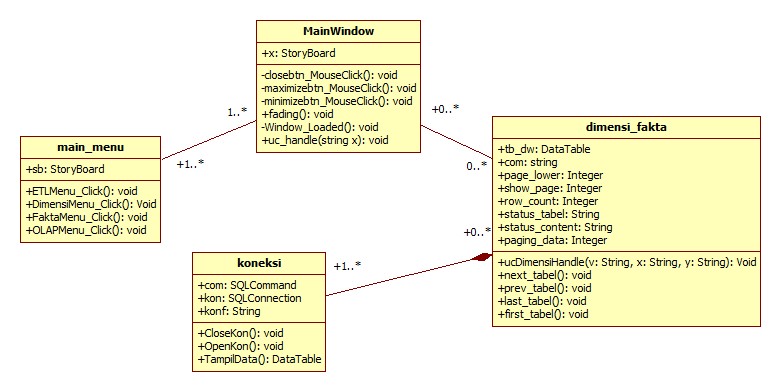
Gambar III‑48 merupakan diagram kelas yang digunakan untuk proses fungsional pada *use case*  yang telah didefinisikan.



Gambar III‑48 Diagram kelas *ETL*

2. Dimensi dan fakta

Dimensi dan fakta merupakan diagram kelas yang menggambarkan kumpulan kelas yang digunakan pada fungsional di *use case* yang digambarkan pada Gambar III‑34. Fungsional yang digambarkan pada diagram kelas ini adalah *use case show dimension-fact*. Gambar III‑49 merupakan diagram kelas yang digunakan untuk proses fungsional pada *use case*  yang telah didefinisikan.



Gambar III‑49 Diagram kelas dimensi dan fakta

3. *OLAP*

Dimensi dan fakta merupakan diagram kelas yang menggambarkan kumpulan kelas yang digunakan pada fungsional di *use case* yang digambarkan pada Gambar III‑34. Fungsional yang digambarkan pada diagram kelas ini adalah.

*Pivoting OLAP*

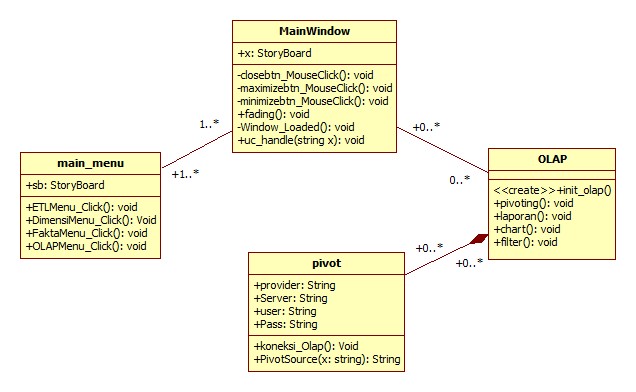
*OLAP drill down*

*OLAP roll up*

*Filter* data

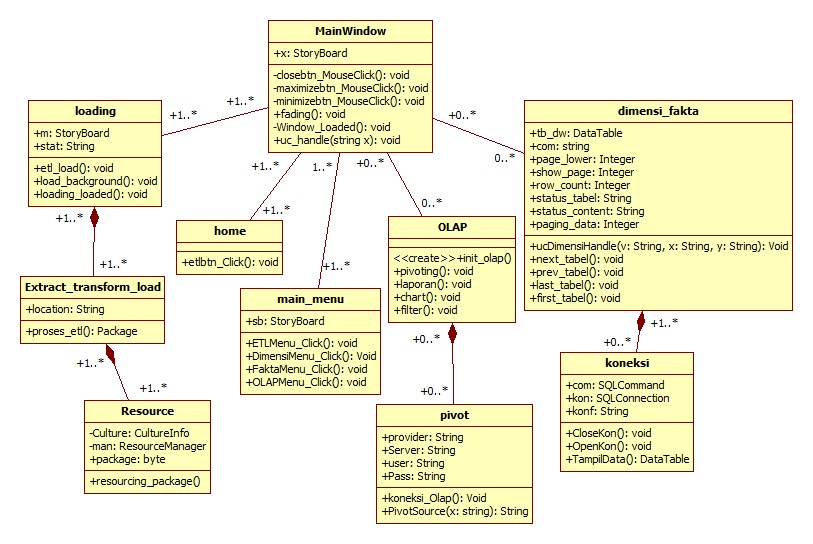
*Print report*

Gambar III‑50 merupakan diagram kelas yang digunakan untuk proses fungsional pada *use case*  yang telah didefinisikan



Gambar III‑50 Diagram kelas pada fungsional *Creating Report*

Gambar III‑51 merupakan diagram kelas keseluruhan yang digunakan di dalam sistem yang telah terealisasi oleh fungsional yang ada di dalam sistem.



Gambar III‑51 Diagram kelas keseluruhan

Analisis kelas digunakan untuk menjelaskan jenis-jenis kelas yang digunakan dan memaparkan tanggung jawab kelas terhadap fungsionalitasnya. Tabel III‑77 menjelaskan jenis-jenis kelas yang digunakan di dalam sistem.

Tabel III‑77 Jenis kelas yang digunakan di dalam sistem

| **No.** | **Nama Kelas** | **Jenis** |
| --- | --- | --- |
| 1 | *MainWindow* | *Boundary* |
| 2 | *Loading* | *Boundary* |
| 3 | *Home* | *Boundary* |
| 4 | *Main\_menu* | *Boundary* |
| 5 | *OLAP* | *Boundary* |
| 6 | Dimensi\_fakta | *Boundary* |
| 7 | *Extract\_transfrom\_load* | *Control* |
| 8 | Koneksi | *Control* |
| 9 | *Pivot* | *Control* |
| 10 | *Resource* | *Model* |

Tabel III‑78 menjelaskan tanggung jawab pada masing-masing kelas yang digunakan didalam sistem.

Tabel III‑78 Tanggung jawab kelas

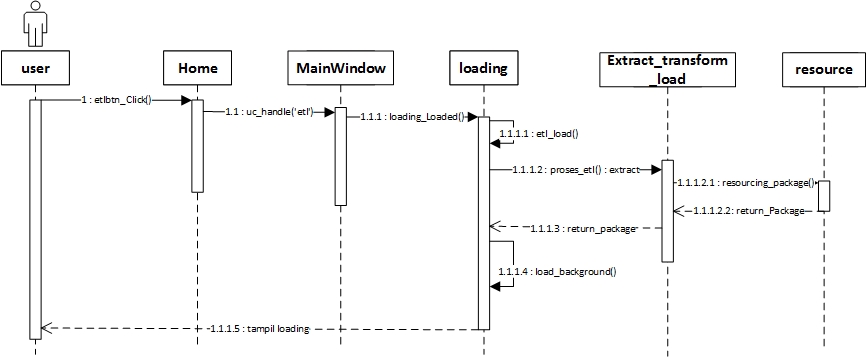
| **Nama Kelas** | **Tanggung Jawab** | **Attribut** |
| --- | --- | --- |
| *MainWindow* | 1. Mengatur *user control* 2. Mengatur alur antarmuka sistem. 3. Mengimplementasikan relasi kelas. | 1. x |
| *loading* | 1. mengatur status *data flow task* saat memproses *data staging*. 2. Menampilkan animasi saat menjalankan *data staging*. | 1. Stat 2. Core 3. M |
| *Home* | 1. Menampilkan antarmuka awal sistem. 2. Mengarahkan pengguna ke proses *ETL*. |  |
| *Main\_menu* | 1. Menampilkan pilihan yang dapat dipilih user. 2. Mengatur animasi menu. | 1. sb |
| *OLAP* | 1. Menampilkan fakta dalam bentuk *cube*. 2. Mencetak laporan. 3. Menampilkan *chart.* 4. Melakukan *filtering* data. |  |
| *Extract\_transform\_load* | 1. Mendefinisikan paket *data flow task* dari *resources*. 2. Mendefinisikan lokasi *path* paket. | 1. *location* |
| Koneksi | 1. Menghubungkan sistem ke data *source*. 2. Menghubungkan sistem ke data *destination*. 3. Mengolah proses ETL. 4. Memanggil data yang diperlukan dengan balikan berupa tabel. | 1. S\_com 2. Konf 3. conn |
| *Pivot* | 1. menghubungkan sistem ke *SQL Analysis service*. 2. Membuat *OLAP* metode *Pivot*. | 1. Provider 2. Server 3. User 4. pass |
| *Resource* | 1. Menampung file, gambar, atau media lainnya yang digunakan di dalam sistem. 2. Mengatur *behavior* *resource* sistem. | 1. Culture 2. Man 3. package |
| Dimensi\_fakta | 1. Menampung data dimensi yang digunakan. 2. Mengkostumasi tampilan dimensi. 3. Membuat list dimensi yang digunakan, 4. Menampilkan dimensi yang ada. | 1. Tb\_dw 2. Com 3. Page\_lower 4. Show\_page 5. Row\_count 6. Status\_tabel 7. Status\_content 8. Paging\_data |

#### Sequence Diagram

*Sequence diagram* digunakan dalam pemodelan perancangan yang bertujuan untuk menjelaskan relasi *Class diagram* yang telah dipaparkan pada *point* III.4.2.4.Tahapan ini merupakan perancangan relasi kelas yang digunakan terahadap realisasi alur proses dari *use case* yang ada pada Gambar III‑34.

1. *Extract* data

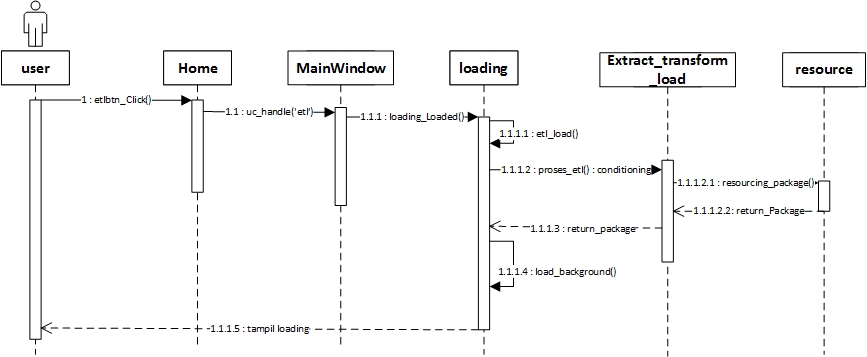
Gambar III‑52menggambarkan *sequence diagram* pada *use case extract* data*.*



Gambar III‑52 *Sequence diagram use case extract* data

2. *Conditioning* data

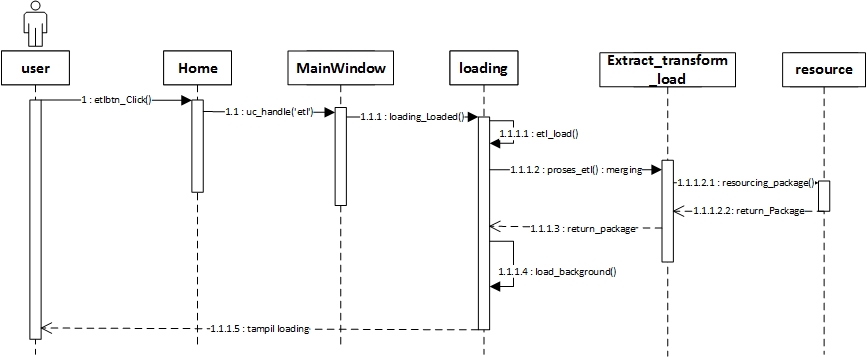
Gambar III‑53merupakan *sequence diagram* pada *use case conditioning* data*.*



Gambar III‑53 *Sequence diagram use case conditioning*  data

3. *Merging* data

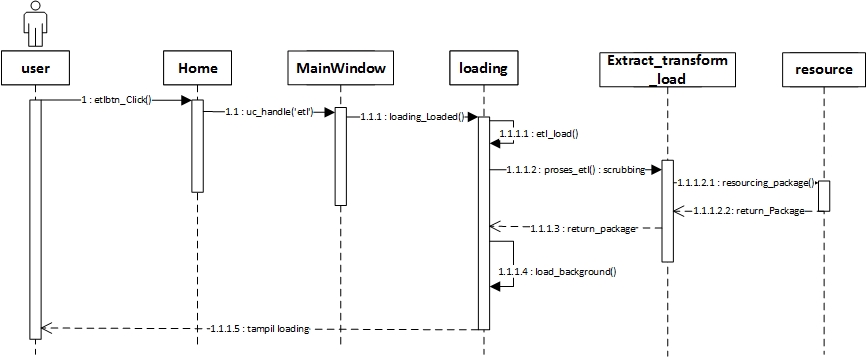
Gambar III‑54merupakan *sequence diagram* pada *use case merging* data*.*



Gambar III‑54 *Sequence diagram use case merging* data

4. *Scrubbing* data

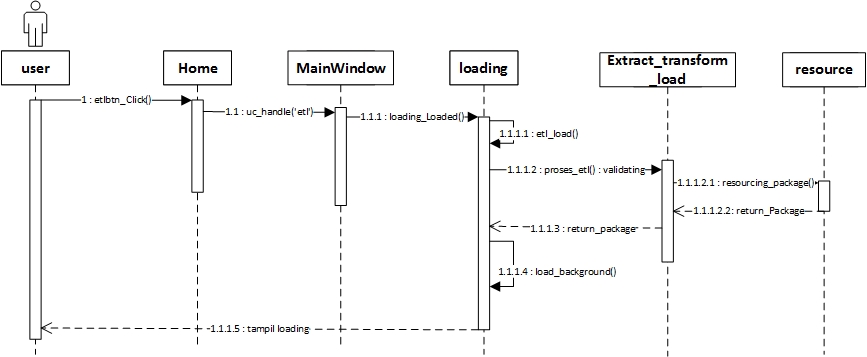
Gambar III‑55merupakan *sequence diagram* pada *use case scrubbing* data*.*



Gambar III‑55 *Sequence diagram use case scrubbing* data

5. *Validating* data

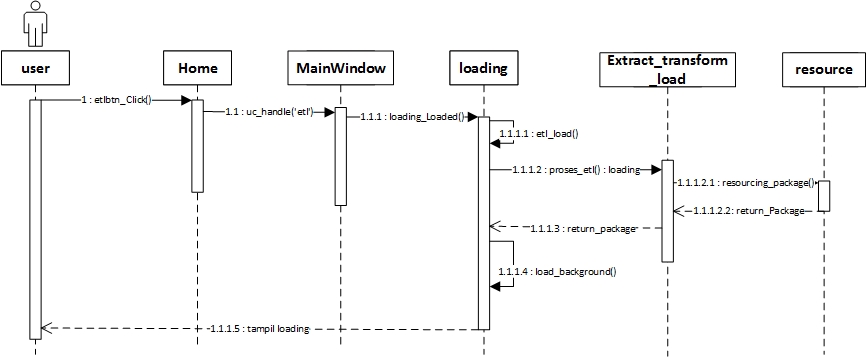
Gambar III‑56merupakan *sequence diagram* pada *use case validating* data*.*



Gambar III‑56 *Sequence diagram use case validating* data

6. *Loading* data

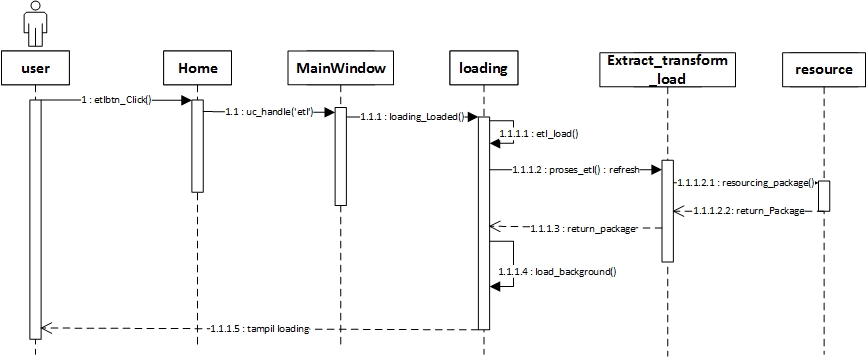
Gambar III‑57merupakan *sequence diagram* pada *use case loading* data*.*



Gambar III‑57 *Sequence diagram use case loading* data

7. *Refresh* data

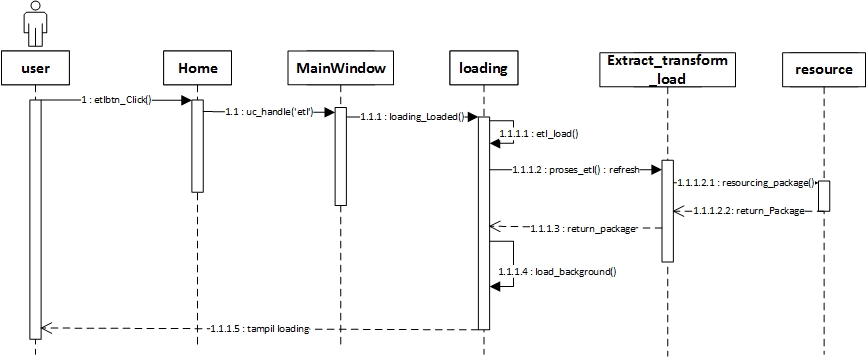
Gambar III‑58merupakan *sequence diagram* pada *use case refresh* data*.*



Gambar III‑58 *Sequence diagram use case refresh* data

8. *Pivoting OLAP*

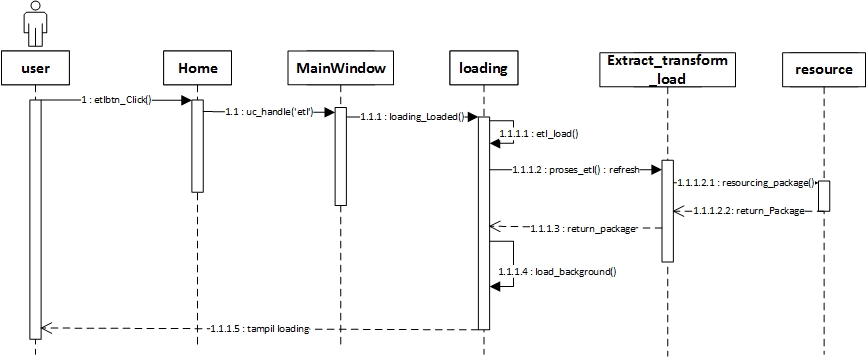
Gambar III‑59merupakan *sequence diagram* pada *use case refresh* data*.*



Gambar III‑59 *Sequence diagram use case pivoting OLAP*

9. *OLAP drill down*

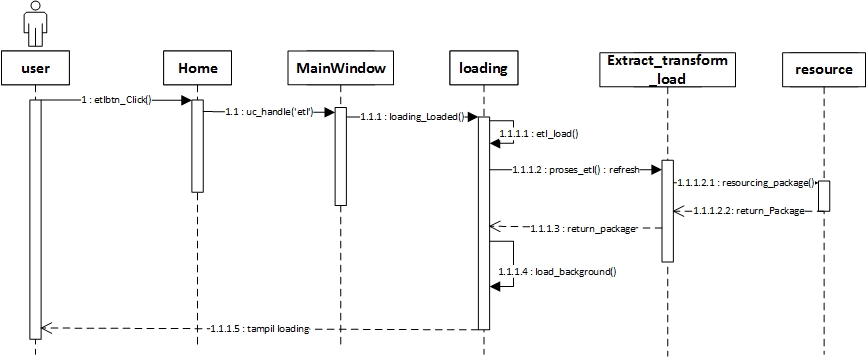
Gambar III‑60merupakan *sequence diagram* pada *use case OLAP drill down.*



Gambar III‑60 *Sequence diagram use case pivoting OLAP*

10. *OLAP roll up*

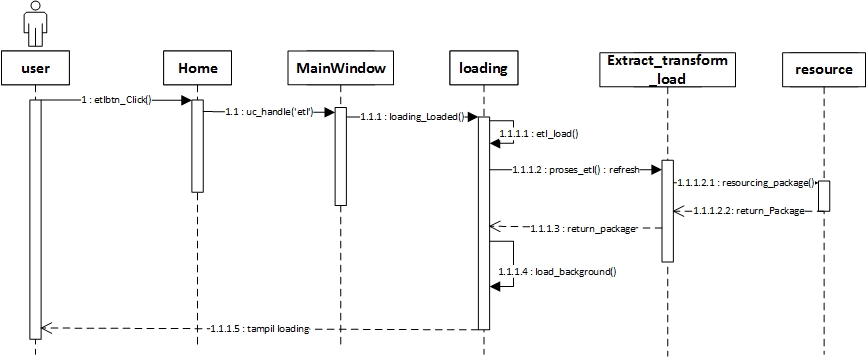
Gambar III‑61merupakan *sequence diagram* pada *use case OLAP roll up.*



Gambar III‑61 *Sequence diagram use case OLAP roll up*

11. *Filter* data

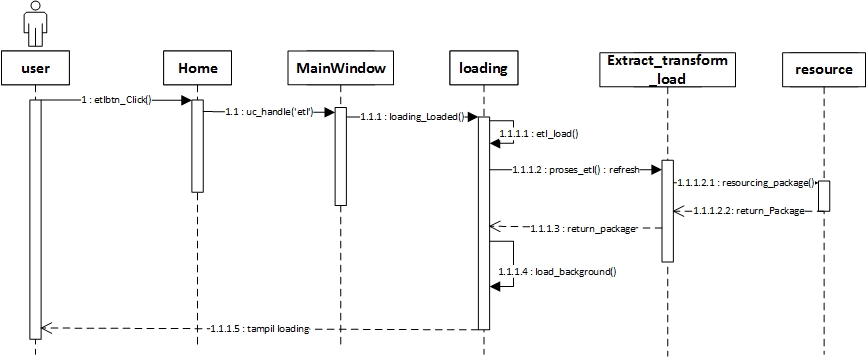
Gambar III‑63merupakan *sequence diagram* pada *use case OLAP filter* data*.*



Gambar III‑62 *Sequence diagram use case filter* data

12. *Print Report*

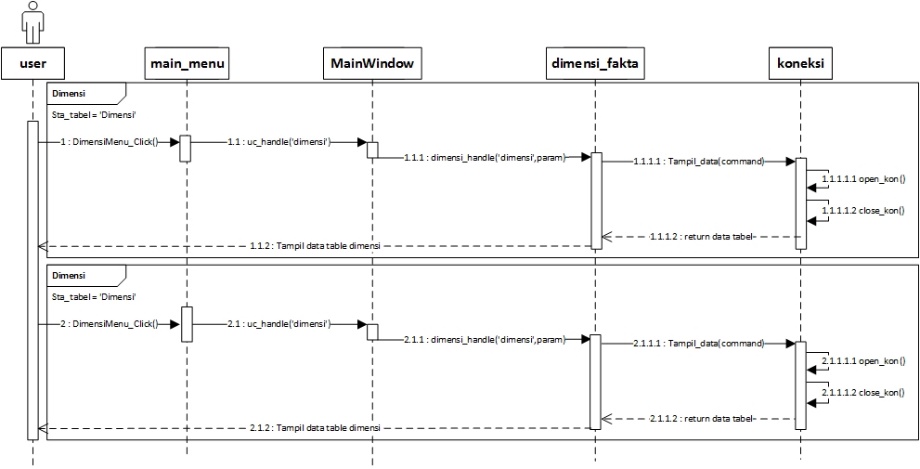
Gambar III‑63merupakan *sequence diagram* pada *use case OLAP print report..*



Gambar III‑63 *Sequence diagram use case print report*

13. *Show dimension -fact*

**Gambar 62** merupakan *sequence diagram* pada *use case* menampilkan dimensi- fakta.



Gambar III‑64 *Sequence diagram use case menampilkan dimensi- fakta*

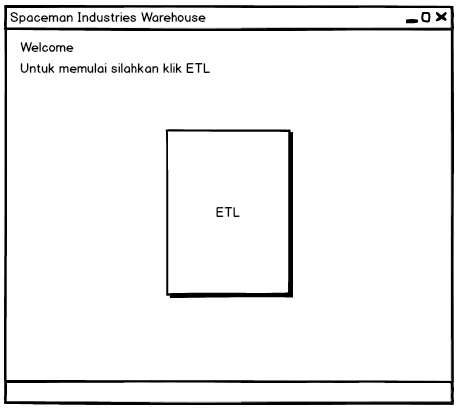
## Perancangan Antarmuka

Perancangan antarmuka bertujuan unutk memodelkan antarmuka yang akan diterapkan ke dalam sistem. Pemodela antarmuka yang baik dapat membantu *user* untuk memahami kerja sistem dan menjadikan panduan untuk menjalankan sistem yang telah dibangun. Berikut antarmuka yang ada di sistem ini.

1. T001-Antarmuka Beranda (*home*)

**Gambar 64** menggambarkan perancangan antarmuka beranda pada sistem yang dibangun. Adapun keterangan dari perancangan antarmuka beranda ini adalah sebagai berikut:

1. Klik tombol “ETL” untuk masuk ke T002

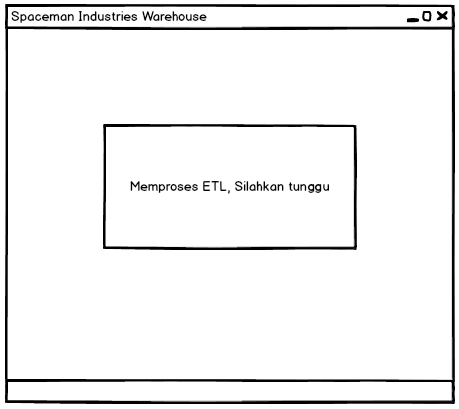


Gambar III‑65 Perancangan antarmuka beranda (T001)

1. T002-Antarmuka Proses ETL

**Gambar 65** menggambarkan perancangan antarmuka proses ETL. Adapun keterangannya adalah sebagai berikut:

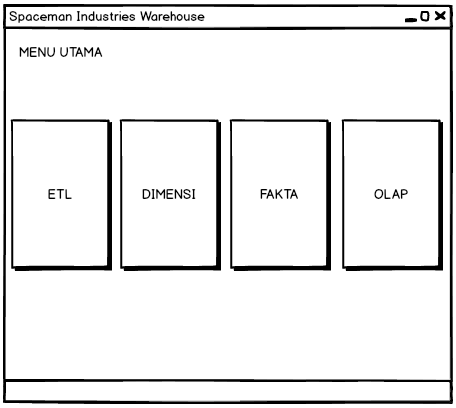
1. Masuk ke T003 bila proses ETL telah selesai.



Gambar III‑66 Perancangan antarmuka proses ETL (T002)

1. T003-Perancangan Antarmuka Menu Utama

**Gambar 66** menggambarkan perancangan antarmuka menu utama.

****

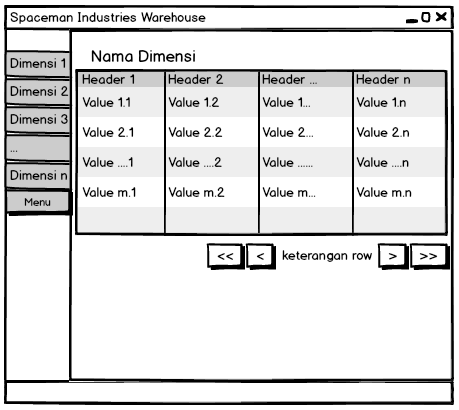
Gambar III‑67 Perancangan antarmuka menu utama (T003)

Adapun keterangan dari perancangan antarmuka menu utama adalah sebagai berikut:

1. Klik tombol “ETL” untuk masuk ke T002.
2. Klik tombol “DIMENSI” untuk masuk ke T004
3. Klik tombol “FAKTA” untuk masuk ke T005
4. T004-Perancangan Antarmuka Dimensi

**Gambar 67** menggambarkan perancangan antarmuka dimensi. Adapun keterangan dari perancangan antarmukan ini adalah sebagai berikut:

1. Klik tombol “Menu” untuk masuk ke T003.
2. Klik tombol “Dimensi” untuk masuk ke T004 dengan *datagrid* dimensi lainnya.
3. Klik tombol “<<” untuk menampilkan sepuluh *row datagrid* pertama.
4. Klik tombol “<” untuk menampilkan sepuluh *row datagrid* sebelumnya.
5. Klik tombol “>” untuk menampilkan sepuluh *row datagrid* selanjutnya.
6. Klik tombol”>” untuk menampilkan sepuluh *row datagrid* terakhir.

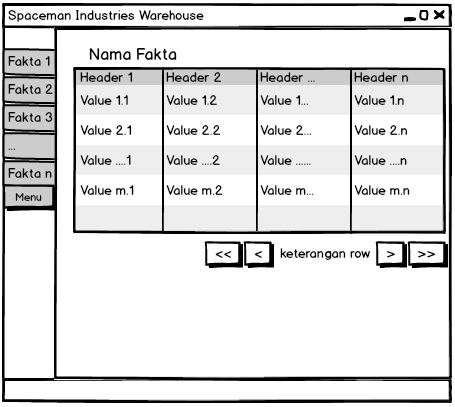
****

Gambar III‑68 Perancangan antarmuka dimensi (T004)

1. T005-Perancangan Antarmuka Fakta

**Gambar 68** menggambarkan perancangan antarmuka fakta. Adapun keterangan dari perancangan antarmukan ini adalah sebagai berikut:

1. Klik tombol “Menu” untuk masuk ke T003.
2. Klik tombol “Fakta” untuk masuk ke T004 dengan *datagrid* fakta lainnya.
3. Klik tombol “<<” untuk menampilkan sepuluh *row datagrid* pertama.
4. Klik tombol “<” untuk menampilkan sepuluh *row datagrid* sebelumnya.
5. Klik tombol “>” untuk menampilkan sepuluh *row datagrid* selanjutnya.
6. Klik tombol”>” untuk menampilkan sepuluh *row datagrid* terakhir.

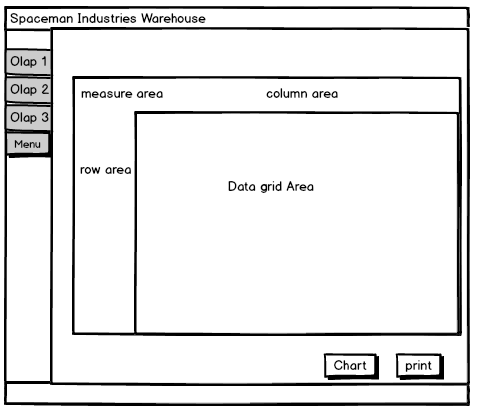


Gambar III‑69 Perancangan antarmuka fakta (T005)

1. T006-Perancangan antarmuka *OLAP*

**Gambar 68** menggambarkan perancangan antarmuka *OLAP*. Adapun keterangan dari perancangan antarmukan ini adalah sebagai berikut:

* + - * 1. Klik tombol menu untuk masuk ke T003
        2. Klik tombol “Olap” untuk masuk ke *OLAP pivot* yang lain.
        3. Klik tombol “ chart” untuk membuka *chart*.
        4. Klik tombol ”print” untuk mencetak *OLAP*.



# IMPLEMENTASI DAN PENGUJIAN SISTEM

## Implementasi Data *Warehouse*

Tahapan implementasi bertujuan untuk merealisasikan hasil analisis dan perancangan yang telah didefinisikan pada **BAB****III**. Tahapan implementasi yang dilakukan pada pembangunan data *warehouse* ini meliputi implementasi lingkungan, implementasi data, dan implementasi antarmuka.

### Implementasi Lingkungan

Implementasi lingkungan mengidentifikasikan pada lingkungan yang digunakan dalam pengimplementasian sistem. Implementasi lingkungan ini membahas tentang kebutuhan minimum yang harus ada untuk menjalankan sistem seperti *hardware*  dan *Software*  yang digunakan.

1. *Hardware*

Kebutuhan perangkat keras minimum yang harus ada dalam merealisasikan sistem adalah sebagai berikut :

1. 2 unit *Computer* (*Client* dan *server*)
2. Processor 2.0 GHZ
3. Hardisk 256 GB
4. RAM DDR3 2GB
5. Monitor Standar
6. Printer
7. Router
8. *Software*

Kebutuhan perangkat lunak minimum yang harus ada dalam merealisasikan sistem adalah sebagai berikut :

1. Sistem Operas *Windows* 7 atau *Windows*  8.
2. Runtime.net versi 4.5.
3. *MSSQL Server* 2014 *Standard Edition*.
4. *Microsoft Office* 2007

### Implementasi Data

Implementasi data menjelaskan perintah-perintah yang dibuat dan digunakan untuk membuat sistem *data warehouse*. Perintah-perintah yang dibuat menggunakan bahasa *query*  dan *derived* *integrity service* pada *tamplate Business Intelligent* pada pemrosesan *ETL*. Adapun implementasinya adalah sebagai berikut:

*Preparing Database*

Implementasi ini merupakan persiapan fisik *database* untuk data *warehouse*. Adapun perintahnya adalah sebagai berikut.

|  |
| --- |
| SE [master]  GO  /\*\*\*\*\*\* Object: Database [dw\_spaceman] Script Date: 30/05/2015 11:04:47 \*\*\*\*\*\*/  CREATE DATABASE [dw\_spaceman]  CONTAINMENT = NONE  ON PRIMARY  ( NAME = N'dw\_spaceman', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.SPACEMAN\MSSQL\DATA\dw\_spaceman.mdf' , SIZE = 20480KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )  LOG ON  ( NAME = N'dw\_spaceman\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.SPACEMAN\MSSQL\DATA\dw\_spaceman\_log.ldf' , SIZE = 1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)  GO  ALTER DATABASE [dw\_spaceman] SET COMPATIBILITY\_LEVEL = 120  GO  IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))  begin  EXEC [dw\_spaceman].[dbo].[sp\_fulltext\_database] @action = 'enable'  end  GO  ALTER DATABASE [dw\_spaceman] SET ANSI\_NULL\_DEFAULT OFF  GO  ALTER DATABASE [dw\_spaceman] SET ANSI\_NULLS OFF  GO  ALTER DATABASE [dw\_spaceman] SET ANSI\_PADDING OFF  GO  ALTER DATABASE [dw\_spaceman] SET ANSI\_WARNINGS OFF  GO  ALTER DATABASE [dw\_spaceman] SET ARITHABORT OFF  GO  ALTER DATABASE [dw\_spaceman] SET AUTO\_CLOSE OFF  GO  ALTER DATABASE [dw\_spaceman] SET AUTO\_SHRINK OFF  GO  ALTER DATABASE [dw\_spaceman] SET AUTO\_UPDATE\_STATISTICS ON  GO  ALTER DATABASE [dw\_spaceman] SET CURSOR\_CLOSE\_ON\_COMMIT OFF  GO  ALTER DATABASE [dw\_spaceman] SET CURSOR\_DEFAULT GLOBAL  GO  ALTER DATABASE [dw\_spaceman] SET CONCAT\_NULL\_YIELDS\_NULL OFF  GO  ALTER DATABASE [dw\_spaceman] SET NUMERIC\_ROUNDABORT OFF  GO  ALTER DATABASE [dw\_spaceman] SET QUOTED\_IDENTIFIER OFF  GO  ALTER DATABASE [dw\_spaceman] SET RECURSIVE\_TRIGGERS OFF  GO  ALTER DATABASE [dw\_spaceman] SET DISABLE\_BROKER  GO  ALTER DATABASE [dw\_spaceman] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF  GO  ALTER DATABASE [dw\_spaceman] SET DATE\_CORRELATION\_OPTIMIZATION OFF  GO  ALTER DATABASE [dw\_spaceman] SET TRUSTWORTHY OFF  GO  ALTER DATABASE [dw\_spaceman] SET ALLOW\_SNAPSHOT\_ISOLATION OFF  GO  ALTER DATABASE [dw\_spaceman] SET PARAMETERIZATION SIMPLE  GO  ALTER DATABASE [dw\_spaceman] SET READ\_COMMITTED\_SNAPSHOT OFF  GO  ALTER DATABASE [dw\_spaceman] SET HONOR\_BROKER\_PRIORITY OFF  GO  ALTER DATABASE [dw\_spaceman] SET RECOVERY FULL  GO  ALTER DATABASE [dw\_spaceman] SET MULTI\_USER  GO  ALTER DATABASE [dw\_spaceman] SET PAGE\_VERIFY CHECKSUM  GO  ALTER DATABASE [dw\_spaceman] SET DB\_CHAINING OFF  GO  ALTER DATABASE [dw\_spaceman] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )  GO  ALTER DATABASE [dw\_spaceman] SET TARGET\_RECOVERY\_TIME = 0 SECONDS  GO  ALTER DATABASE [dw\_spaceman] SET DELAYED\_DURABILITY = DISABLED  GO  ALTER DATABASE [dw\_spaceman] SET READ\_WRITE  GO  USE [dw\_spaceman]  GO  /\*\*\*\*\*\* Object: Table [dbo].[dim\_design] Script Date: 30/05/2015 11:04:34 \*\*\*\*\*\*/  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  SET ANSI\_PADDING ON  GO  CREATE TABLE [dbo].[dim\_design](  [no\_design] [int] NOT NULL,  [nama\_design] [varchar](200) NOT NULL,  CONSTRAINT [PK\_dim\_design] PRIMARY KEY CLUSTERED  (  [no\_design] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  SET ANSI\_PADDING OFF  GO  SET ANSI\_PADDING ON  GO  CREATE TABLE [dbo].[dim\_job](  [no\_job] [int] NOT NULL,  [keterangan\_job] [varchar](100) NOT NULL,  CONSTRAINT [PK\_dim\_job] PRIMARY KEY CLUSTERED  (  [no\_job] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  SET ANSI\_PADDING OFF  GO  SET ANSI\_PADDING ON  GO  CREATE TABLE [dbo].[dim\_konsumen](  [no\_konsumen] [int] NOT NULL,  [nama\_konsumen] [varchar](50) NOT NULL,  [nama\_kota] [varchar](50) NOT NULL,  CONSTRAINT [PK\_dim\_konsumen] PRIMARY KEY CLUSTERED  (  [no\_konsumen] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  SET ANSI\_PADDING OFF  GO  SET ANSI\_PADDING ON  GO  CREATE TABLE [dbo].[dim\_kredit](  [no\_kredit] [int] IDENTITY(1,1) NOT NULL,  [jenis\_kredit] [varchar](50) NOT NULL,  CONSTRAINT [PK\_dim\_kredit] PRIMARY KEY CLUSTERED  (  [no\_kredit] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  SET ANSI\_PADDING OFF  GO  SET ANSI\_PADDING ON  GO  CREATE TABLE [dbo].[dim\_pengiriman](  [no\_pengiriman] [int] NOT NULL,  [jenis\_pengiriman] [varchar](100) NOT NULL,  CONSTRAINT [PK\_dim\_pengiriman] PRIMARY KEY CLUSTERED  (  [no\_pengiriman] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  SET ANSI\_PADDING OFF  GO  SET ANSI\_PADDING ON  GO  CREATE TABLE [dbo].[dim\_produk](  [no\_produk] [int] NOT NULL,  [nama\_produk] [varchar](50) NOT NULL,  CONSTRAINT [PK\_dim\_produk] PRIMARY KEY CLUSTERED  (  [no\_produk] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  SET ANSI\_PADDING OFF  GO  SET ANSI\_PADDING ON  GO  CREATE TABLE [dbo].[dim\_size](  [no\_size] [int] NOT NULL,  [nama\_size] [varchar](10) NOT NULL,  CONSTRAINT [PK\_dim\_size] PRIMARY KEY CLUSTERED  (  [no\_size] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  SET ANSI\_PADDING OFF  GO  SET ANSI\_PADDING ON  GO  CREATE TABLE [dbo].[dim\_staff\_produksi](  [no\_staff] [int] NOT NULL,  [nama\_staff] [varchar](50) NOT NULL,  [nama\_bagian] [varchar](50) NOT NULL,  CONSTRAINT [PK\_dim\_staff\_produksi] PRIMARY KEY CLUSTERED  (  [no\_staff] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  SET ANSI\_PADDING OFF  GO  CREATE TABLE [dbo].[dim\_waktu](  [no\_waktu] [int] IDENTITY(1,1) NOT NULL,  [string\_tanggal] [nvarchar](50) NOT NULL,  [hari] [nvarchar](50) NOT NULL,  [tanggal] [int] NOT NULL,  [nama\_bulan] [nvarchar](50) NOT NULL,  [bulan] [int] NOT NULL,  [tahun] [int] NOT NULL,  [full\_date] [date] NULL,  CONSTRAINT [PK\_dim\_waktu] PRIMARY KEY CLUSTERED  (  [no\_waktu] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  CREATE TABLE [dbo].[fact\_belanja](  [waktu] [int] NOT NULL,  [produk] [int] NOT NULL,  [kredit] [int] NOT NULL,  [cost] [float] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[fact\_belanja] WITH CHECK ADD CONSTRAINT [FK\_fact\_belanja\_dim\_kredit] FOREIGN KEY([kredit])  REFERENCES [dbo].[dim\_kredit] ([no\_kredit])  GO  ALTER TABLE [dbo].[fact\_belanja] CHECK CONSTRAINT [FK\_fact\_belanja\_dim\_kredit]  GO  ALTER TABLE [dbo].[fact\_belanja] WITH CHECK ADD CONSTRAINT [FK\_fact\_belanja\_dim\_produk] FOREIGN KEY([produk])  REFERENCES [dbo].[dim\_produk] ([no\_produk])  GO  ALTER TABLE [dbo].[fact\_belanja] CHECK CONSTRAINT [FK\_fact\_belanja\_dim\_produk]  GO  ALTER TABLE [dbo].[fact\_belanja] WITH CHECK ADD CONSTRAINT [FK\_fact\_belanja\_dim\_waktu] FOREIGN KEY([waktu])  REFERENCES [dbo].[dim\_waktu] ([no\_waktu])  GO  ALTER TABLE [dbo].[fact\_belanja] CHECK CONSTRAINT [FK\_fact\_belanja\_dim\_waktu]  GO  CREATE TABLE [dbo].[fact\_biaya\_produksi](  [waktu] [int] NOT NULL,  [produk] [int] NOT NULL,  [job] [int] NOT NULL,  [staff] [int] NOT NULL,  [jumlah] [int] NOT NULL,  [biaya\_produksi] [float] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[fact\_biaya\_produksi] WITH CHECK ADD CONSTRAINT [FK\_fact\_biaya\_produksi\_dim\_job] FOREIGN KEY([job])  REFERENCES [dbo].[dim\_job] ([no\_job])  GO  ALTER TABLE [dbo].[fact\_biaya\_produksi] CHECK CONSTRAINT [FK\_fact\_biaya\_produksi\_dim\_job]  GO  ALTER TABLE [dbo].[fact\_biaya\_produksi] WITH CHECK ADD CONSTRAINT [FK\_fact\_biaya\_produksi\_dim\_produk] FOREIGN KEY([produk])  REFERENCES [dbo].[dim\_produk] ([no\_produk])  GO  ALTER TABLE [dbo].[fact\_biaya\_produksi] CHECK CONSTRAINT [FK\_fact\_biaya\_produksi\_dim\_produk]  GO  ALTER TABLE [dbo].[fact\_biaya\_produksi] WITH CHECK ADD CONSTRAINT [FK\_fact\_biaya\_produksi\_dim\_staff\_produksi] FOREIGN KEY([staff])  REFERENCES [dbo].[dim\_staff\_produksi] ([no\_staff])  GO  ALTER TABLE [dbo].[fact\_biaya\_produksi] CHECK CONSTRAINT [FK\_fact\_biaya\_produksi\_dim\_staff\_produksi]  GO  ALTER TABLE [dbo].[fact\_biaya\_produksi] WITH CHECK ADD CONSTRAINT [FK\_fact\_biaya\_produksi\_dim\_waktu] FOREIGN KEY([waktu])  REFERENCES [dbo].[dim\_waktu] ([no\_waktu])  GO  ALTER TABLE [dbo].[fact\_biaya\_produksi] CHECK CONSTRAINT [FK\_fact\_biaya\_produksi\_dim\_waktu]  GO  CREATE TABLE [dbo].[fact\_pembayaran\_konsumen](  [id\_pembayaran] [int] IDENTITY(1,1) NOT NULL,  [waktu] [int] NOT NULL,  [konsumen] [int] NOT NULL,  [rentang\_pembayaran] [int] NOT NULL,  CONSTRAINT [PK\_fact\_pembayaran\_konsumen] PRIMARY KEY CLUSTERED  (  [id\_pembayaran] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[fact\_pembayaran\_konsumen] WITH CHECK ADD CONSTRAINT [FK\_fact\_pembayaran\_konsumen\_dim\_konsumen] FOREIGN KEY([konsumen])  REFERENCES [dbo].[dim\_konsumen] ([no\_konsumen])  GO  ALTER TABLE [dbo].[fact\_pembayaran\_konsumen] CHECK CONSTRAINT [FK\_fact\_pembayaran\_konsumen\_dim\_konsumen]  GO  ALTER TABLE [dbo].[fact\_pembayaran\_konsumen] WITH CHECK ADD CONSTRAINT [FK\_fact\_pembayaran\_konsumen\_dim\_waktu] FOREIGN KEY([waktu])  REFERENCES [dbo].[dim\_waktu] ([no\_waktu])  GO  ALTER TABLE [dbo].[fact\_pembayaran\_konsumen] CHECK CONSTRAINT [FK\_fact\_pembayaran\_konsumen\_dim\_waktu]  GO  CREATE TABLE [dbo].[fact\_pemesanan](  [waktu] [int] NOT NULL,  [konsumen] [int] NOT NULL,  [produk] [int] NOT NULL,  [jumlah] [int] NOT NULL,  [saldo\_order] [float] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[fact\_pemesanan] WITH CHECK ADD CONSTRAINT [FK\_fact\_pemesanan\_dim\_konsumen] FOREIGN KEY([konsumen])  REFERENCES [dbo].[dim\_konsumen] ([no\_konsumen])  GO  ALTER TABLE [dbo].[fact\_pemesanan] CHECK CONSTRAINT [FK\_fact\_pemesanan\_dim\_konsumen]  GO  ALTER TABLE [dbo].[fact\_pemesanan] WITH CHECK ADD CONSTRAINT [FK\_fact\_pemesanan\_dim\_produk] FOREIGN KEY([produk])  REFERENCES [dbo].[dim\_produk] ([no\_produk])  GO  ALTER TABLE [dbo].[fact\_pemesanan] CHECK CONSTRAINT [FK\_fact\_pemesanan\_dim\_produk]  GO  ALTER TABLE [dbo].[fact\_pemesanan] WITH CHECK ADD CONSTRAINT [FK\_fact\_pemesanan\_dim\_waktu] FOREIGN KEY([waktu])  REFERENCES [dbo].[dim\_waktu] ([no\_waktu])  GO  ALTER TABLE [dbo].[fact\_pemesanan] CHECK CONSTRAINT [FK\_fact\_pemesanan\_dim\_waktu]  GO  CREATE TABLE [dbo].[fact\_pengiriman\_produk](  [waktu] [int] NOT NULL,  [produk] [int] NOT NULL,  [pengiriman] [int] NOT NULL,  [jumlah] [int] NOT NULL,  [saldo] [float] NOT NULL  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[fact\_pengiriman\_produk] WITH CHECK ADD CONSTRAINT [FK\_fact\_pengiriman\_produk\_dim\_pengiriman] FOREIGN KEY([pengiriman])  REFERENCES [dbo].[dim\_pengiriman] ([no\_pengiriman])  GO  ALTER TABLE [dbo].[fact\_pengiriman\_produk] CHECK CONSTRAINT [FK\_fact\_pengiriman\_produk\_dim\_pengiriman]  GO  ALTER TABLE [dbo].[fact\_pengiriman\_produk] WITH CHECK ADD CONSTRAINT [FK\_fact\_pengiriman\_produk\_dim\_produk] FOREIGN KEY([produk])  REFERENCES [dbo].[dim\_produk] ([no\_produk])  GO  ALTER TABLE [dbo].[fact\_pengiriman\_produk] CHECK CONSTRAINT [FK\_fact\_pengiriman\_produk\_dim\_produk]  GO  ALTER TABLE [dbo].[fact\_pengiriman\_produk] WITH CHECK ADD CONSTRAINT [FK\_fact\_pengiriman\_produk\_dim\_waktu] FOREIGN KEY([waktu])  REFERENCES [dbo].[dim\_waktu] ([no\_waktu])  GO  ALTER TABLE [dbo].[fact\_pengiriman\_produk] CHECK CONSTRAINT [FK\_fact\_pengiriman\_produk\_dim\_waktu]  GO  CREATE TABLE [dbo].[fact\_pop\_design](  [no\_pop\_design] [int] IDENTITY(1,1) NOT NULL,  [waktu] [int] NOT NULL,  [produk] [int] NOT NULL,  [design] [int] NOT NULL,  [jumlah] [int] NOT NULL,  CONSTRAINT [PK\_fact\_pop\_design] PRIMARY KEY CLUSTERED  (  [no\_pop\_design] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[fact\_pop\_design] WITH CHECK ADD CONSTRAINT [FK\_fact\_pop\_design\_dim\_design] FOREIGN KEY([design])  REFERENCES [dbo].[dim\_design] ([no\_design])  GO  ALTER TABLE [dbo].[fact\_pop\_design] CHECK CONSTRAINT [FK\_fact\_pop\_design\_dim\_design]  GO  CREATE TABLE [dbo].[fact\_pop\_produk](  [no\_pop\_produk] [int] IDENTITY(1,1) NOT NULL,  [waktu] [int] NOT NULL,  [produk] [int] NOT NULL,  [jumlah] [int] NOT NULL,  CONSTRAINT [PK\_fact\_pop\_produk] PRIMARY KEY CLUSTERED  (  [no\_pop\_produk] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  CREATE TABLE [dbo].[fact\_size\_produk](  [waktu] [int] NOT NULL,  [produk] [int] NOT NULL,  [size] [int] NOT NULL,  [gender] [varchar](10) NOT NULL,  [jumlah] [int] NOT NULL  ) ON [PRIMARY]  GO  SET ANSI\_PADDING OFF  GO  ALTER TABLE [dbo].[fact\_size\_produk] WITH CHECK ADD CONSTRAINT [FK\_fact\_size\_produk\_dim\_produk] FOREIGN KEY([produk])  REFERENCES [dbo].[dim\_produk] ([no\_produk])  GO  ALTER TABLE [dbo].[fact\_size\_produk] CHECK CONSTRAINT [FK\_fact\_size\_produk\_dim\_produk]  GO  ALTER TABLE [dbo].[fact\_size\_produk] WITH CHECK ADD CONSTRAINT [FK\_fact\_size\_produk\_dim\_size] FOREIGN KEY([size])  REFERENCES [dbo].[dim\_size] ([no\_size])  GO  ALTER TABLE [dbo].[fact\_size\_produk] CHECK CONSTRAINT [FK\_fact\_size\_produk\_dim\_size]  GO  ALTER TABLE [dbo].[fact\_size\_produk] WITH CHECK ADD CONSTRAINT [FK\_fact\_size\_produk\_dim\_waktu] FOREIGN KEY([waktu])  REFERENCES [dbo].[dim\_waktu] ([no\_waktu])  GO  CREATE TABLE [dbo].[fmt\_populer\_design](  [no\_fmt\_pop\_design] [int] IDENTITY(1,1) NOT NULL,  [pop\_design] [int] NOT NULL,  [ca\_pop\_design] [nvarchar](20) NOT NULL,  [mda\_pop\_design] [float] NOT NULL,  CONSTRAINT [PK\_fmt\_populer\_design] PRIMARY KEY CLUSTERED  (  [no\_fmt\_pop\_design] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[fmt\_populer\_design] WITH CHECK ADD CONSTRAINT [FK\_fmt\_populer\_design\_fact\_pop\_design] FOREIGN KEY([pop\_design])  REFERENCES [dbo].[fact\_pop\_design] ([no\_pop\_design])  GO  ALTER TABLE [dbo].[fmt\_populer\_design] CHECK CONSTRAINT [FK\_fmt\_populer\_design\_fact\_pop\_design]  GO  CREATE TABLE [dbo].[fmt\_populer\_produk](  [no\_fmt\_pop\_produk] [int] IDENTITY(1,1) NOT NULL,  [populer\_produk] [int] NOT NULL,  [ca\_populer\_produk] [nvarchar](20) NOT NULL,  [mda\_populer\_produk] [float] NOT NULL,  CONSTRAINT [PK\_fmt\_populer\_produk] PRIMARY KEY CLUSTERED  (  [no\_fmt\_pop\_produk] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO  ALTER TABLE [dbo].[fmt\_populer\_produk] WITH CHECK ADD CONSTRAINT [FK\_fmt\_populer\_produk\_fact\_pop\_produk] FOREIGN KEY([populer\_produk])  REFERENCES [dbo].[fact\_pop\_produk] ([no\_pop\_produk])  GO  ALTER TABLE [dbo].[fmt\_populer\_produk] CHECK CONSTRAINT [FK\_fmt\_populer\_produk\_fact\_pop\_produk]  GO  CREATE TABLE [dbo].[fmt\_rentang\_pembayaran](  [no\_fmt\_pembayaran] [int] IDENTITY(1,1) NOT NULL,  [pembayaran\_konsumen] [int] NOT NULL,  [ca\_pembayaran] [nvarchar](20) NOT NULL,  [mda\_pembayaran] [float] NOT NULL,  CONSTRAINT [PK\_fmt\_rentang\_pembayaran] PRIMARY KEY CLUSTERED  (  [no\_fmt\_pembayaran] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO |

*Truncate* Data

*Truncate* data merupakan implementasi yang diterapkan dalam proses data *staging*. Adapun pengimplementasiannya adalah sebagai berikut:

|  |
| --- |
| delete from fmt\_populer\_design;  DBCC checkident(fmt\_populer\_design,reseed,0);  delete from fmt\_populer\_produk;  DBCC checkident(fmt\_populer\_produk,reseed,0);  delete from fmt\_rentang\_pembayaran;  DBCC checkident(fmt\_rentang\_pembayaran,reseed,0);  delete from fact\_pemesanan;  delete from fact\_pengiriman\_produk;  delete from fact\_belanja;  delete from fact\_biaya\_produksi;  delete from fact\_pop\_produk;  DBCC checkident(fact\_pop\_produk,reseed,0);  delete from fact\_pop\_design;  DBCC checkident(fact\_pop\_design,reseed,0);  delete from fact\_pembayaran\_konsumen;  DBCC checkident(fact\_pembayaran\_konsumen,reseed,0);  delete from fact\_size\_produk;  delete from dim\_waktu;  DBCC checkident(dim\_waktu,reseed,0);  delete from dim\_produk;  delete from dim\_konsumen;  delete from dim\_pengiriman;  delete from dim\_staff\_produksi;  delete from dim\_job;  delete from dim\_kredit;  DBCC checkident(dim\_kredit,reseed,0);  delete from dim\_design;  delete from dim\_size; |

*Derived Waktu*

*Derived* waktu merupakan implementasi yang diterapkan dalam memproses data waktu yang ada pada database *OLTP* menuju ke dimensi waktu di data *warehouse*. Adapun implementasinya adalah sebagai berikut.

|  |
| --- |
| //tanggal  ISNULL(waktu) || TRIM(waktu) == "" ? (DT\_I4)0 : (DT\_I4)RIGHT(waktu,2)  //hari  (DT\_WSTR,50)((DT\_I4)DATEPART("dw",(DT\_DATE)(SUBSTRING(waktu,1,4) + "-" + SUBSTRING(waktu,6,2) + "-" + RIGHT(waktu,2))) == 1 ?"Senin": (DT\_I4)DATEPART("dw",(DT\_DATE)(SUBSTRING(waktu,1,4) + "-" + SUBSTRING(waktu,6,2) + "-" + RIGHT(waktu,2))) == 2 ? "Selasa" : (DT\_I4)DATEPART("dw",(DT\_DATE)(SUBSTRING(waktu,1,4) + "-" + SUBSTRING(waktu,6,2) + "-" + RIGHT(waktu,2))) == 3 ? "Rabu" : (DT\_I4)DATEPART("dw",(DT\_DATE)(SUBSTRING(waktu,1,4) + "-" + SUBSTRING(waktu,6,2) + "-" + RIGHT(waktu,2))) == 4 ? "Kamis" : (DT\_I4)DATEPART("dw",(DT\_DATE)(SUBSTRING(waktu,1,4) + "-" + SUBSTRING(waktu,6,2) + "-" + RIGHT(waktu,2))) == 5 ? "Jumat" : (DT\_I4)DATEPART("dw",(DT\_DATE)(SUBSTRING(waktu,1,4) + "-" + SUBSTRING(waktu,6,2) + "-" + RIGHT(waktu,2))) == 6 ? "Sabtu" : (DT\_I4)DATEPART("dw",(DT\_DATE)(SUBSTRING(waktu,1,4) + "-" + SUBSTRING(waktu,6,2) + "-" + RIGHT(waktu,2))) == 7 ? "Minggu" : "Null")  //bulan  ISNULL(waktu) || TRIM(waktu) == "" ? (DT\_I4)0 : (DT\_I4)SUBSTRING(waktu,6,2)  //nama\_bulan  (DT\_WSTR,50)(SUBSTRING(waktu,6,2) == "01" ? "Januari" : SUBSTRING(waktu,6,2) == "02" ? "Februari" : SUBSTRING(waktu,6,2) == "03" ? "Maret" : SUBSTRING(waktu,6,2) == "04" ? "April" : SUBSTRING(waktu,6,2) == "05" ? "Mei" : SUBSTRING(waktu,6,2) == "06" ? "Juni" : SUBSTRING(waktu,6,2) == "07" ? "Juli" : SUBSTRING(waktu,6,2) == "08" ? "Agustus" : SUBSTRING(waktu,6,2) == "09" ? "September" : SUBSTRING(waktu,6,2) == "10" ? "Oktober" : SUBSTRING(waktu,6,2) == "11" ? "November" : SUBSTRING(waktu,6,2) == "12" ? "Desember" : "NULL")  //tahun  ISNULL(waktu) || TRIM(waktu) == "" ? (DT\_I8)0 : (DT\_I4)SUBSTRING(waktu,1,4)  //*fulldate*  (DT\_DATE)waktu  // waktu *key*  (DT\_WSTR,8)REPLACE(waktu,"-","") |

Konsep *fuzzy* kepopuleran

Implementasi ini merupakan penerapan pemodelan konsep *fuzzy* yang digunakan untuk mengklasifikasi kepopuleran baik dari segi prooduk maupun desain. Adapun implementasinya adalah sebagai berikut.

|  |
| --- |
| //kurang populer  (DT\_R4)(jumlah >= 16 && jumlah <= 20 ? 1.00 : jumlah < 16 && jumlah > 0 ? (jumlah / 16.00) : jumlah > 20 && jumlah < 36 ? ((36 - jumlah) / 16.00) : 0.00)  //populer  (DT\_R4)(jumlah > 20 && jumlah < 36 ? ((jumlah - 20) / 16.00) : jumlah >= 36 && jumlah <= 44 ? 1.00 : jumlah > 44 && jumlah < 60 ? ((60 - jumlah) / 16.00) : 0.00)  //sangat populer  (DT\_R4)(jumlah > 44 && jumlah < 60 ? ((jumlah - 44) / 16.00) : jumlah >= 60 ? 1.00 : 0.00) |

Konsep *fuzzy* pembayaran

Implementasi ini merupakan penerapan pemodelan konsep *fuzzy* yang digunakan untuk mengklasifikasi rentang pembayaran konsumen. Adapun implementasinya adalah sebagai berikut.

|  |
| --- |
| //tepat waktu  (DT\_R4)((DT\_I4)rentang\_pembayaran == 0 ? 1.00 : (DT\_I4)rentang\_pembayaran > 0 && (DT\_I4)rentang\_pembayaran < 10 ? (10 - (DT\_I4)rentang\_pembayaran) / 10.00 : 0.00)  //sedikit terlambat  (DT\_R4)((DT\_I4)rentang\_pembayaran == 13 ? 1.00 : (DT\_I4)rentang\_pembayaran > 7 && (DT\_I4)rentang\_pembayaran < 13 ? (rentang\_pembayaran - 7) / 6.00 : (DT\_I4)rentang\_pembayaran > 13 && (DT\_I4)rentang\_pembayaran < 20 ? (20 - rentang\_pembayaran) / 7.00 : 0.00)  //terlambat  (DT\_R4)((DT\_I4)rentang\_pembayaran == 23 ? 1.00 : (DT\_I4)rentang\_pembayaran > 16 && (DT\_I4)rentang\_pembayaran < 23 ? ((DT\_I4)rentang\_pembayaran - 16) / 7.00 : (DT\_I4)rentang\_pembayaran > 23 && (DT\_I4)rentang\_pembayaran < 30 ? (30 - (DT\_I4)rentang\_pembayaran) / 7.00 : 0.00)  //sangat terlambat  (DT\_R4)((DT\_I4)rentang\_pembayaran >= 30 ? 1.00 : (DT\_I4)rentang\_pembayaran > 25 && (DT\_I4)rentang\_pembayaran < 30 ? ((DT\_I4)rentang\_pembayaran - 25) / 5.00 : 0.00) |

### Implementasi Antarmuka

Implementasi antarmuka diterapkan dengan menampilkan setiap tampilan sistem yang dibangun dan pengkodeannya dalam bentuk file program. Tampilan implementasi antarmuka dari sistem dapat dilihat di Lampiran A-1 dan implementasi antarmuka dalam merealisasikan perancangan antar mukaakan dipaparkan pada sebagai berikut:

Tabel IV‑1 Implementasi antarmuka

|  |  |  |
| --- | --- | --- |
| **Kode** | **Perancangan** | **Nama *File*** |
| T001 | Beranda | Home.xaml |
| T002 | Proses *ETL* | Etl\_load.xaml |
| T003 | Menu utama | 1. Main\_menu.xaml |
| T004 | Dimensi | Dimensi\_fakta.xaml  Dimensi\_menu.xaml |
| T005 | Fakta | Dimensi\_fakta.xaml  Fakta\_menu.xaml |
| T006 | *OLAP* | OLAP.xaml |

Pengujian Data *Warehouse*

Rencana Pengujian

Skenario Pengujian

Hasil Pengujian

Evaluasi Hasil Pengujian

KESIMPULAN DAN SARAN

Kesimpulan

Saran

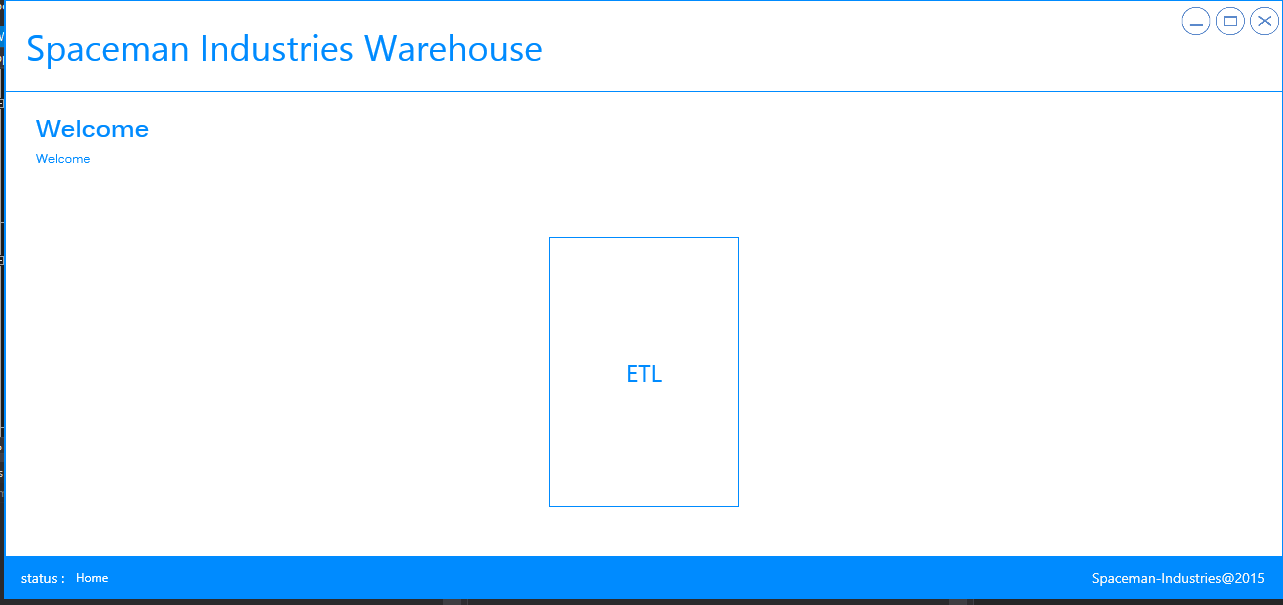
# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | L. Sapir, A. Schimilovici dan L. Rokach, “A Methodolgy for The Design of a Fuzzy Data Warehouse,” *Intelligent System,* vol. 1, no. 1, pp. 2.14 - 2.21, 2008. |
| [2] | P. D. Sugiyono, Metode Penelitian Kuantitatif, Kualtitatif dan Kombinasi, Bandung: Alfabeta, 2014. |
| [3] | I. Sommerville, Software Engineering Ninth Edition, New York: Pearson Education, 2011. |
| [4] | R. Kimball dan M. Ross, The Data Warehouse Toolkit, Indianapolis: John WIlley & Sons, 2013. |
| [5] | D. Fasel, Fuzzy Data Warehousing for Performance Measurement : Concept and Implementation, Switzerland: Springer, 2014. |
| [6] | A. Silberschatz, F. K. Henry dan S. Sudarshan, Database System Concepts, Sixth edition, New York: McGraw Hill Companies, 2011. |
| [7] | P. Ponniah, Data Warehousing Fundamental, New York: John Willey & Sons, INC., 2001. |
| [8] | H. A. Jefrey, G. F. Joey dan J. S. Valacich, Modern Systems Analysis and Design Sixth Edition, New York: Pearson Education, 2011. |
| [9] | K. Hamilton dan R. Miles, Learning UML 2.0, Sebastopol: O'Reilly Media Inc, 2006. |

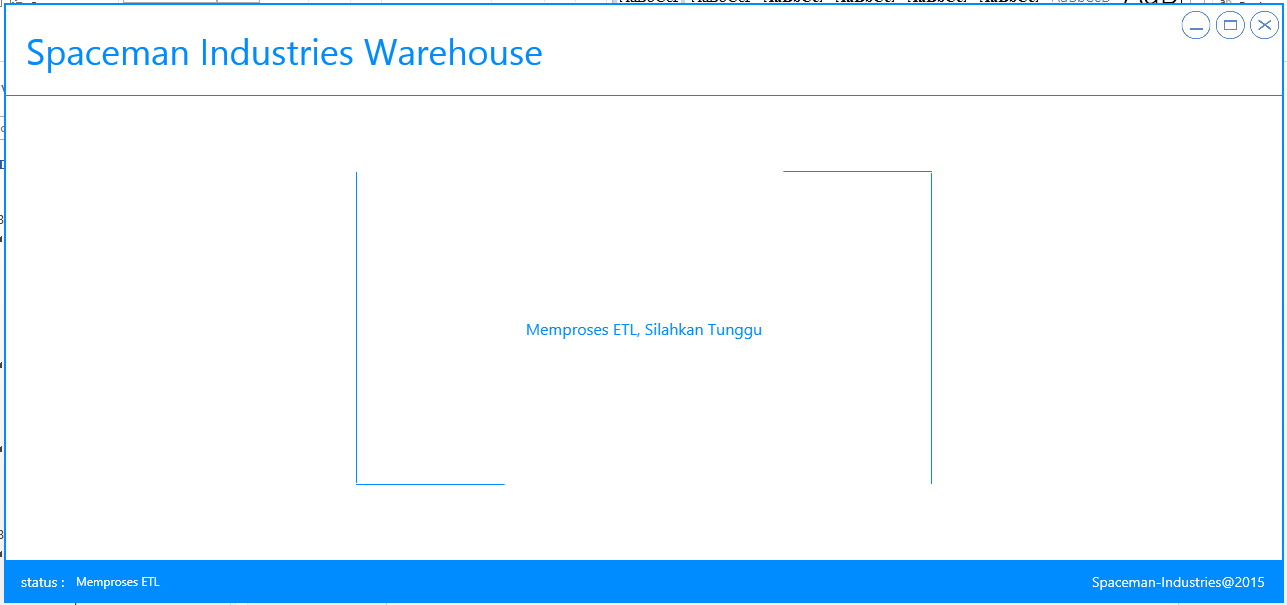
**LAMPIRAN A**

**IMPLEMENTASI ANTARMUKA**

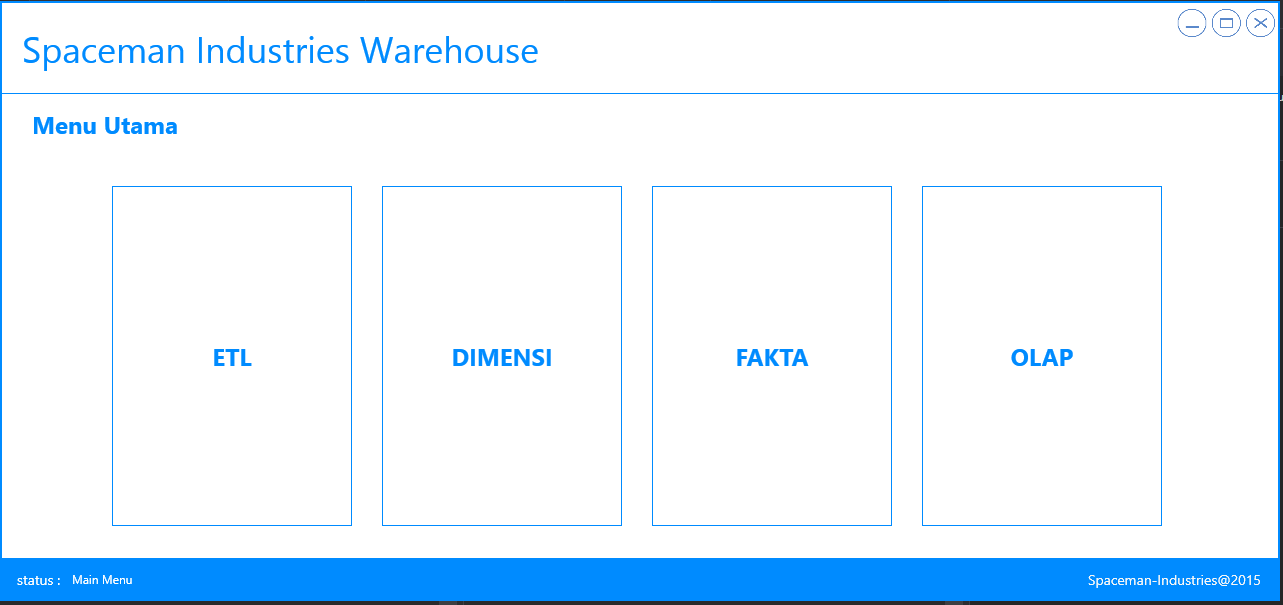
1. **Antarmuka beranda**

****

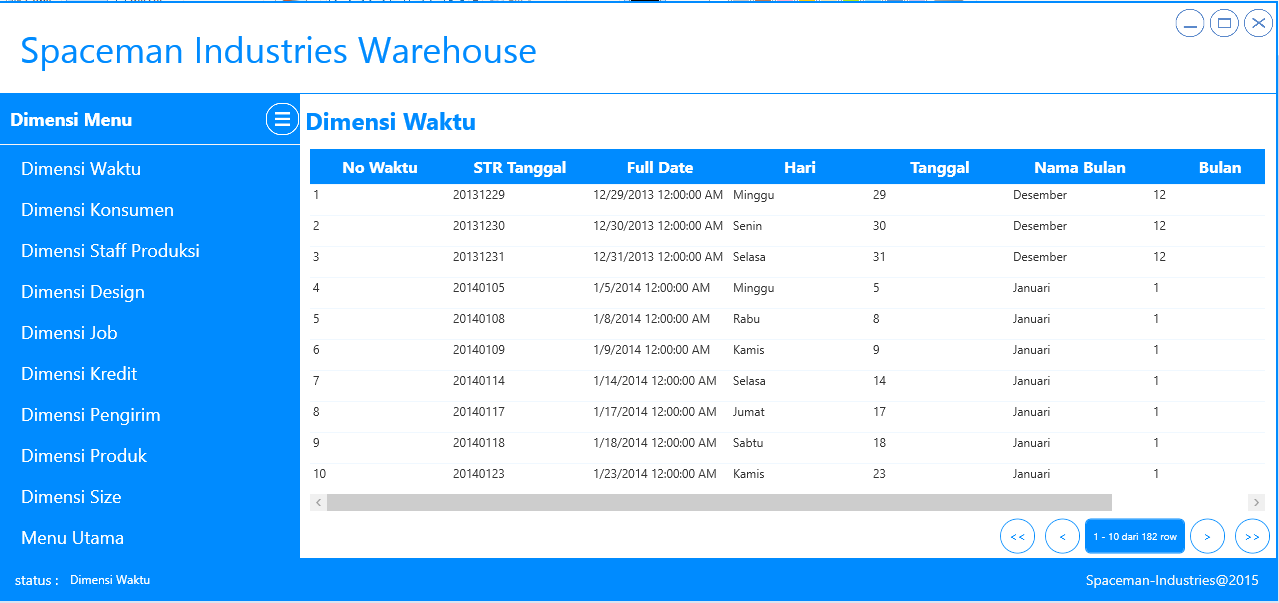
1. **Antarmuka proses ETL**

****

1. **Antarmuka menu utama**

****

1. **Antarmuka Dimensi**

****

1. **Antarmuka Fakta**

****