BAB II
LANDASAN TEORI

2.1 *Data Warehouse*

Data Warehouse (DW) merupakan basis data relasional yang didesain lebih kepada analisa dan *query* daripada proses transaksi, dan biasanya mengandung data historis dari proses transaksi dan bisa juga dari sumber lainnya untuk tujuan pengambilan keputusan. *Data Warehouse* dapat juga disebut sebagai tempat penyimpanan ringkasan dari data historis yang seringkali diambil dari basis data terpisah departemen, organisasi atau perusahaan (Kimball dan Caserta, 2004).

Menurut Inmon (2002) bahwa *data warehouse* merupakan koleksi data yang mempunyai sifat berorientasi subyek, terintegrasi, *time-variant*, dan bersifat tetap dari koleksi data dalam mendukung proses pengambilan keputusan manajemen.

Tujuan utama dari pembuatan *data warehouse* merupakan untuk menggabungkan data yang beragam ke dalam sebuah tempat penyimpanan dimana pengguna dapat dengan mudah menjalankan *query*, menghasilkan laporan, dan melakukan analisis. Meningkatkan efektifitas pembuatan keputusan merupakan salah satu keuntungan yang didapatkan dari *data warehouse.*

Dari definisi yang dijelaskan di atas, dapat disimpulkan bahwa *data warehouse* merupakan basisdata yang saling berinteraksi dan dapat digunakan untuk *query* dan analisis, bersifat orientasi subyek, terintegrasi, *time-variant*, tidak berubah (*adhoc*) yang nantinya digunakan dalam membantu pengambilan keputusan organisasi atau perusahaan oleh pihak pengambil keputusan.

2.1.1 Karakteristik Data Warehouse

Berikut ini merupakan karakteristik *data warehouse* (Inmon, 2002) yaitu:

1. Berorientasi Subyek (*Subject Oriented*) yaitu *data warehouse* dirancang untuk menganalisa data berdasarkan subyek tertentu dalam perusahaan atau organisasi, bukan pada proses atau fungsi aplikasi tertentu. Hal ini disebabkan karena kebutuhan dari *data warehouse* adalah untuk menyimpan data yang digunakan sebagai penunjang suatu keputusan. Secara garis besar perbedaan antara data operasional dan *data warehouse* yakni:

Tabel 2.1 Perbandingan Fungsi Data Operasional dan *Data Warehouse*

Data Operasional	**Data Warehouse**
Dirancang hanya berorientasi pada aplikasi dan fungsi tertentu	Dirancang berdasar pada subyek-subyek tertentu (utama)
Fokusnya pada desain basisdata dan proses	Fokusnya pada pemodelan data dan desain data
Berisi rincian atau detail data	Berisi data history yang akan dipakai dalam proses analisis
Relasi antar tabel berdasar aturan terkini (selalu mengikuti rule(aturan) terbaru)	Banyak aturan bisnis dapat tersaji antara tabel-tabel

Sumber : (Inmon, 2002).

2. Terintegrasi (*Integrated*) yaitu *data warehouse* dapat menyimpan data yang berasal dari sumber data yang berbeda kedalam suatu format yang konsisten dan saling terintegrasi satu dengan lainnya. Dengan demikian data tidak bisa dipecah-pecah karena data yang ada merupakan suatu kesatuan yang menunjang keseluruhan konsep *data warehouse* itu sendiri. Syarat integrasi sumber data dapat dipenuhi dengan berbagai cara seperti penamaan variabel yang konsisten, ukuran variabel yang konsisten, struktur pengkodean yang konsiten, dan atribut fisik dari data yang konsisten.
3. Rentang Waktu (*Time-Variant*) yaitu seluruh data pada *data warehouse* dapat dikatakan akurat atau valid pada rentang waktu tertentu. Untuk melihat interval waktu yang digunakan dalam mengukur keakuratan suatu *data warehouse*, dapat menggunakan cara antara lain yakni :
    - Cara yang paling sederhana merupakan menyajikan *data warehouse* pada rentang waktu tertentu, misalnya antara 5 sampai 10 tahun ke depan.
    - Cara yang kedua, dengan menggunakan variasi/perbedaan waktu yang disajikan dalam *data warehouse* baik implisit maupun secara eksplisit dengan unsur waktu dalam hari, minggu, bulan. Secara implisit misalnya pada saat data tersebut diduplikasi pada setiap akhir bulan, atau per tiga bulan. Unsur waktu akan tetap ada secara implisit didalam data tersebut.

- Cara yang ketiga, variasi waktu yang disajikan *data warehouse* melalui serangkaian *snapshot* yang panjang. *Snapshot* merupakan tampilan dari sebagian data tertentu sesuai keinginan pemakai dari keseluruhan data yang ada bersifat *read-only*.

4. *Non Volatile* yaitu data yang ada pada *data warehouse* tidak dapat diperbaharui atau di *update*, tetapi hanya dapat di *refresh* dari data operasional atau sumber data berdasarkan waktu yang telah ditentukan. Data yang baru selalu ditambahkan sebagai suplemen bagi *database* itu sendiri dari pada sebagai sebuah perubahan. *Database* tersebut secara kontinyu menyerap data baru ini, kemudian secara *incremental* disatukan dengan data sebelumnya.

2.1.2 Tugas Data Warehouse

Ada empat tugas yang bisa dilakukan oleh *data warehouse* (Kimball dan Caserta, 2004) yaitu :

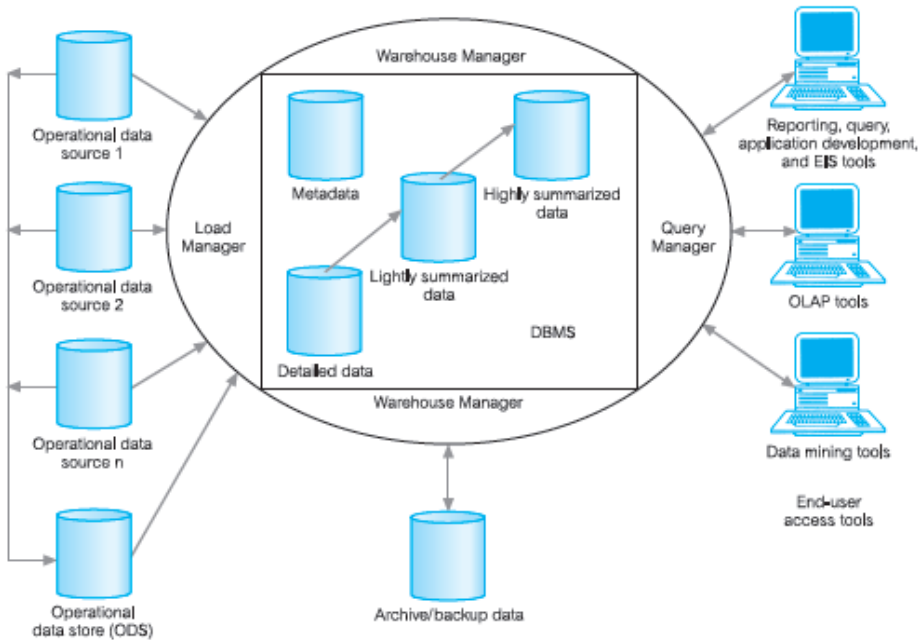
b. Pembuatan Laporan yakni proses pembuatan laporan merupakan salah satu kegunaan *data warehouse* yang paling umum dilakukan. Dengan menggunakan *query* sederhana didapatkan laporan perhari, perbulan, pertahun atau jangka waktu kapan pun yang diinginkan.

c. OLAP yakni dengan adanya *data warehouse*, semua informasi baik detail maupun hasil *summary* yang dibutuhkan dalam proses analisa mudah di dapat. OLAP mendayagunakan konsep multidimensional dan memungkinkan para pemakai menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah *query*.

d. *Data Mining* yakni merupakan proses untuk menggali pengetahuan dan informasi baru dari data yang berjumlah banyak pada *data warehouse*, dengan menggunakan kecerdasan buatan (*artificial intelligence*), statistik dan matematika.

e. Proses Informasi Eksekutif yakni *data warehouse* dapat membuat ringkasan informasi yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data. Dengan menggunakan *data warehouse* segala laporan telah diringkas dan dapat pula mengetahui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan.

1.1.3 Arsitektur Data Warehouse



Gambar 2.1 Arsitektur *Data Warehouse*

Sumber : (Kimbbal dan Caserta,2004)

Komponen-komponen yang ada dalam arsitektur diatas antara lain sebagai berikut:

1. *Operational Data*

   Operational Data adalah data yang digunakan dalam proses operasional harian.

2. *Operational Data Store* (ODS)

   Operational Data Store (ODS) adalah sebuah tempat penyimpanan data saat ini dan data operasional terintegrasi yang digunakan untuk analisis. ODS menyediakan kemudahan untuk menggunakan *database* relasional ketika berada jauh dari fungsi *decision support* dari *data warehouse*. Dengan terlebih dahulu membangun ODS, pembuatan *data warehouse* menjadi lebih sederhana karena ODS dapat menyuplai data yang telah diekstrak dari sistem sumber.

3. *Load Manager*

   Load manager disebut juga komponen *front-end* melakukan semua operasi yang berhubungan dengan ekstraksi dan *loading* data ke dalam *data warehouse*.

4. *Warehouse Manager*

Warehouse manager melaksanakan semua operasi yang berhubungan dengan pengelolaan data dalam *data warehouse.* Operasi-operasi yang dilaksanakan oleh *warehouse manager* meliputi: Analisis data untuk memastikan konsistensinya; Transformasi dan penggabungan sumber data dari tempat penyimpanan sementara ke dalam tabel-tabel *data warehouse;* Pembuatan *index* dan *view* pada *base tables;* - Melakukan denormalisasi; Melakukan *aggregation; Backup* dan *archive data.*

5. *Query Manager*

   Query manager menampilkan semua operasi yang terkait dengan *query* pengguna. Operasi yang ditampilkan oleh komponen ini meliputi mengarahkan *query* pada tabel yang cocok dan menjadwalkan pelaksanaan *query.*

6. *Detailed Data*

   Dalam *data warehouse,* area ini menyimpan *detailed data* dalam skema *database.* Umumnya, *detailed data* tidak disimpan secara *online* tetapi dapat dibuat ada dengan mengagregasi data ke tingkatan yang lebih detail.

7. *Lightly and Highly Summarized Data*

   Area ini menyimpan semua data *lightly* dan *highly summarized* yang sudah terdefinisi sebelumnya yang dibuat oleh *warehouse manager.* Tujuan informasi yang terangkum ini untuk meningkatkan performa *query.*

8. *Archive* atau *backup data*

   Area ini menyimpan semua *detail* dan ringkasan data untuk tujuan *archiving* dan *backup.* Walaupun ringkasan data yang dihasilkan dari *data detail,* dengan itu akan memungkinkan *backup* ringkasan data yang dilakaukan secara *online.* Jika data ini disimpan melebihi periode penyimpanan untuk data yang rinci. Data akan ditransfer ke *storage archives* seperti *optical disk.*

9. *Metadata*

   Metadata merupakan data mengenai data yang mendeskripsikan *data warehouse. Metadata* digunakan untuk membangun, memelihara, mengatur, dan menggunakan *data warehouse. Metadaata* mengandung lokasi dan deskripsi dari komponen-komponen dalam *data warehouse.* Seperti nama, definisi, struktur, dan isi dari *data warehouse* dan *end user view. Metadata* juga mengidentifikasi sumber-sumber data yang terintegrasi dan bertransformasi dalam *data warehouse.*

10. *End User Access Tools*

Pembangunan data warehouse adalah untuk menyediakan data yang konsisten kepada *user* yang akan digunakan untuk menganalisis dan menyediakan informasi untuk mendukung pengambilan keputusan.

1.1.4 Perancangan Data Warehouse

Salah satu dari metodologi perancangan *data warehouse* adalah *Nine-Step Methodology* oleh Connoly dan Begg (2005:1187) yang memiliki 9 (sembilan) langkah sebagai berikut:

1. Memilih Proses (*Choosing The Process*)

   Proses (fungsi) mengacu pada subjek masalah dari *data mart* tertentu. *Data mart* yang pertama kali dibangun haruslah tepat waktu, sesuai dengan anggaran, dan dapat menjawab pertanyaan pertanyaan bisnis yang penting.

2. Memilih Sumber (*Choosing The Grain*)

   Memilih *grain* berarti memutuskan secara pasti apa yang dinyatakan oleh *record* dari tabel fakta. Hanya dengan telah terpilihnya *grain* untuk table fakta maka kita dapat mengidentifikasi dimensi. Keputusan *grain* untuk tabel fakta juga menentukan *grain* untuk setiap dimensi pada tabel fakta.

3. Mengidentifikasi dan Penyesuaian Dimensi (*Identifying and Conforming The Dimensions*)

   Dimensi menentukan konteks untuk memberikan pertanyaan mengenai fakta-fakta dalam tabel fakta. Kumpulan dimensi yang dibangun dengan baik membuat *data mart* dapat dimengerti dan mudah untuk digunakan. Dimensi diidentifikasi dalam detail yang cukup untuk mendeskripsikan data seperti *client* dan *properties* dari *grain* yang tepat. Jika ada dimensi yang muncul dalam dua *data mart*, maka kedua *data mart* tersebut harus memiliki dimensi yang sama, atau salah satu *data mart* adalah subset matematis dari *data mart* yang lain. Hanya dengan cara ini dua *data mart*s berbagi satu atau lebih dimensi pada aplikasi yang sama. Ketika sebuah dimensi digunakan oleh lebih dari satu *data mart* dan dimensi tersebut tidak disinkronisasikan antar *data mart* maka seluruh *data warehouse* akan gagal, karena dua *data mart* tidak dapat digunakan pada saat yang bersamaan.

4. Memilih Fakta (Choosing The Fact)

Grain dari tabel fakta menentukan fakta mana yang dapat digunakan pada *data mart*. Semua fakta harus dinyatakan berdasarkan tingkatan yang tersirat oleh *grain*. Fakta tambahan dapat ditambahkan ke dalam tabel fakta pada setiap waktu dengan catatan fakta tersebut konsisten dengan *grain* dari tabel.

5. Menyimpan Perhitungan Awal dalam Tabel Fakta (*Storing Pre-Calculation in The Fact Table*)

   Setelah fakta dipilih, setiap fakta harus dikaji ulang untuk menentukan apakah ada kemungkinan untuk melakukan *pre-calculations*. *Pre-calculations* terjadi ketika fakta terdiri dari *statement* untung dan rugi.

6. Melihat Kembali Table Dimensi (*Rounding Out The Dimension Tables*)

   Pada langkah ini, kita kembali mengkaji tabel dimensi dan menambahkan sebanyak mungkin deskripsi teks ke dimensi. Teks deskripsi haruslah seintuitif mungkin dan dapat dimengerti oleh pengguna. Kegunaan dari *data mart* ditentukan oleh cakupan dan sifat atribut pada tabel dimensi.

7. Memilih Durasi *Database* (*Choosing The Duration of Database*)

   Durasi mengukur berapa lama tabel fakta dapat disimpan. Pada banyak perusahaan, ada ketentuan untuk melihat pada periode waktu yang sama satu atau dua tahun sebelumnya. Tabel fakta yang sangat besar akan mengakibatkan setidaknya dua masalah yang signifikan pada *data warehouse*. Pertama, bertambahnya kesulitan untuk menjadikan data lama yang semakin bertambah sebagai sumber. Semakin lama suatu data, semakin banyak masalah dalam membaca dan menginterpretasikan *file* lama tersebut. Kedua, kebutuhan dimensi menggunakan versi yang lama, bukan versi yang baru.

8. Menelusuri Perubahan dari Dimensi secara Perlahan (*Tracking Slowly Changing Dimension*)

   Pada tahap ini, *data warehouse* memperhatikan proses dimensi yang semakin tua seiring dengan berjalannya waktu. Untuk itu perlu dilakukan *update* agar *data warehouse* selalu konsisten. Terdapat tiga tipe dasar perubahan dimensi secara perlahan:

   a. Tipe 1: Perubahan data secara langsung atau *update* table dimensi.

   b. Tipe 2: Perubahan data membentuk *record* baru dengan *surrogate key* yang berbeda.

c. Tipe 3: Perubahan data akan membentuk atribut atau kolom baru pada tabel dimensi.

9. Memutuskan Prioritas *Query* dan Tipe *Query* (*Deciding The Query Priorities and The Query Models*)

Pada tahap ini dipertimbangkan masalah perancangan fisik (*physical design*). Masalah utama pada perancangan fisik yang mempengaruhi persepsi pengguna akhir dari *data mart* adalah urutan penyusunan tabel fakta pada *disk* dan adanya *pre-stored summaries* dan agregasi.

1.1.5 Pemodelan Dimensional

Menurut Connolly dan Begg (2005, p1183), pemodelan dimensional adalah sebuah teknik perancangan logical yang bertujuan untuk mempresentasikan data ke dalam sebuah standar, bentuk intuitif yang dapat diakses dengan performa yang tinggi. Setiap model dimensional terdiri dari sebuah tabel dengan sebuah *primary key* komposit yang disebut dengan tabel fakta, dan sekumpulan tabel yang lebih kecil yang disebut dengan tabel dimensi. Setiap tabel dimensi memiliki sebuah *primary key* (nonkomposit) sederhana yang berkorespondensi tepat dengan satu *key* komposit pada tabel fakta. Dengan kata lain, *primary key* dari tabel fakta terbuat dari dua atau lebih *foreign key*. Karakteristik dengan struktur yang seperti bintang ini disebut dengan *star schema* atau *star join*.

1. *Star Schema*

   Star schema adalah sebuah model data dimensional yang memiliki sebuah tabel fakta di pusatnya, dikelilingi oleh tabel-tabel dimensi yang berisi referensi data yang biasanya dapat didenormalisasi. Skema bintang mengeksploitasi karakteristik dari data faktual, seperti fakta dihasilkan dari *event-event* yang terjadi di masa lampau, dan tidak akan berubah terlepas dari bagaimana tabel fakta dianalisis. Fakta yang paling berguna dalam tabel fakta adalah perhitungan *numeric*, atau "fakta" yang terjadi pada setiap *record*. Sedangkan tabel dimensi mengandung informasi tekstual yang deskriptif. Atribut pada tabel dimensi digunakan sebagai batasan-batasan dalam *query data warehouse*. *Star schema* dapat digunakan untuk mempercepat kinerja *query* dengan mendenormalisasikan data referensi ke dalam sebuah tabel dimensi. Denormalisasi cocok dilakukan ketika ada beberapa entitas yang berhubungan dengan tabel dimensi sering diakses. Dengan denormalisasi maka penggabungan (*join*) tabel-tabel tambahan untuk mengakses atribut tersebut dapat dihindari.

2. *Snowflake Schema*

   Snowflake schema adalah model data dimensional yang memiliki sebuah tabel fakta sebagai pusatnya, dikelilingi tabel-tabel dimensi yang ternormalisasi. *Snowflake schema* adalah sebuah variasi dari *star schema* dimana tabel dimensinya boleh memiliki dimensi. Penggunaan tabel dimensi pada *snowflake schema* sangatlah mendasar, sedangkan pada *star schema* tidak. *Snowflake schema* dibuat berdasarkan OLTP sehingga semua data akan termuat detail dalam setiap tabel fakta dan tabel dimensi.

3. *Starflake Schema*

   Starflake schema adalah model data dimensional yang memiliki sebuah tabel fakta sebagai pusatnya, dikelilingi oleh tabel-tabel dimensi yang ternormalisasi dan terdenormalisasi. Beberapa pemodelan dimensional menggunakan campuran dari *star schema* yang terdenormalisasi dan *snowflake schema* yang ternormalisasi. Kombinasi dari *star schema* dan *snowflake schema* disebut *starflake schema*.

1.2 Business Intelligence

Menurut (Niu, 2009), *business intelligence* adalah proses mengekstrak, transformasi, mengelola, dan menganalisis data bisnis untuk mendukung pengambilan keputusan. Dalam proses ini pada umumnya melibatkan data set dalam jumlah besar yang tersimpan dalam *datawarehouse*. Proses b*usiness intelligence* meliputi lima tahapan :

1. Pengumpulan data.

   Sistem *business intelligence* dapat mengekstrak data dari beberapa sumber data yang berasal dari berbagai unit bisnis seperti pemasaran, produksi, sumber daya manusia, dan keuangan. Data yang sudah diekstrak harus dibersihkan, transformasi, dan terintegrasi untuk dapat dianalisis.

2. Analisis data.

   Pada tahapan ini, data dikonversi menjadi informasi atau pengetahuan melalui berbagai macam teknik analisis seperti laporan, visualisasi, dan *data mining*. Hasil dari proses analisis dapat membantu pihak manajemen untuk memahami situasi dan mengambil keputusan yang lebih baik.

3. Kesadaran situasi.

   Kesadaran terhadap situasi dapat memberikan pemahaman yang lebih mendalam terhadap keadaan keputusan saat ini berdasarkan hasil analisis data.

4. Penilaian resiko.

   Kesadaran terhadap situasi yang cukup bervariasi dapat membantu manajer untuk memprediksi masa depan, identifikasi ancaman dan peluang, dan merespon sesuai dengan kebutuhan. Saat ini bisnis beroperasi dalam kondisi lingkungan yang kompleks. Pengambilan keputusan bisnis lebih mungkin disertai resiko yang berasal dari lingkungan eksternal dan internal. Sehingga dapat disimpulkan bahwa penilaian resiko merupakan fungsi penting pada sistem *business intelligence.*

5. Dukungan pengambilan keputusan.

   Tujuan utama dari *business intelligence* adalah membantu manajer mengambil keputusan dengan bijaksana berdasarkan data bisnis saat ini.

2.2.1 Arsitektur Sistem Business Intelligence

Menurut (Niu, 2009), pada umumnya sistem *business intelligence* terdiri dari empat level komponen dan modul manajemen *metadata.* Arsitektur general dari sistem *business intelligence* terlampir pada gambar 1. Komponen-komponen saling berinteraksi untuk memfasilitasi fungsi dasar *business intelligence:* mengekstrak data dari sistem operasional perusahaan, menyimpan data yang sudah diekstrak kedalam *datawarehouse*, dan menarik data yang disimpan untuk berbagai aplikasi analisis bisnis.

a. Level sistem operasional. : Sebagai sumber data dari sistem *business intelligence*, sistem operasional bisnis pada umumnya menggunakan sistem *online transaction processing (OLTP)* untuk mendukung kegiatan bisnis sehari-hari. Pada umumnya sistem OLTP adalah sistem penerimaan order pelanggan, sistem keuangan, dan sistem sumber daya manusia.

b. Level akuisisi data : Pada level ini adalah komponen pra proses terdiri dari 3 tahapan yaitu : ekstraksi, transformasi, dan memasukkan (ETL). Sebuah perusahaan memiliki beberapa sistem OLTP yang menghasilkan jumlah data yang sangat besar. Data tersebut

pertama kali diekstrak dari sistem OLTP oleh proses ETL dan kemudian ditransformasi sesuai dengan aturan transformasi. Apabila data yang sudah ditransformasi maka data tersebut dimasukkan ke *data warehouse.* ETL merupakan komponen dasar dari sistem *business intelligence* karena kualitas data dari komponen lain tergantung pada proses ETL. Dalam perancangan dan pengembangan ETL, kualitas data, fleksibilitas sistem dan kecepatan proses adalah perhatian utama.

c. Level penyimpanan data : Data yang telah diproses oleh komponen ETL disimpan dalam *data warehouse* dimana biasanya diimplementasikan dengan menggunakan tradisional sistem manajemen *database (RDMS). RDMS* didesain untuk mendukung proses transaksi, sangat bertolak belakang dengan *data warehouse* berfokus kepada subyek, varian waktu dan disimpan secara terintegrasi. Skema *star* dan *snowflake* merupakan skema *data warehouse* yang paling populer. Apapun skema yang dipakai, tipe tabel pada *data warehouse* adalah *fact tables* dan *dimension tables.*
d. Level analitis : Berdasarkan *data warehouse*, berbagai macam aplikasi analitikal telah dikembangkan. Sistem *business intelligence* mendukung 2 tipe dasar dalam fungsi analitikal: pelaporan dan *online analytical processing (OLAP)*. Fungsi pelaporan menyediakan manajer berbagai jenis laporan bisnis seperti laporan penjualan, laporan produk, dan laporan sumber daya manusia. Laporan dihasilkan dari menjalankan *queries* kedalam *data warehouse.Data warehouse queries* pada umumnya sudah didefinisikan oleh pengembang *data warehouse.* Laporan yang dihasilkan oleh sistem *business intelligence* biasanya memiliki format yang statis dan berisi tipe data yang pasti.