

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Mustafid (2013), telah melakukan penelitian “Sistem *Business Intelligence* Untuk Mendukung Perguruan Tinggi yang Kompetitif”. Penelitian ini menunjukkan bahwa untuk pengambilan keputusan ditingkat perguruan tinggi, perlu diadakannya sistem *business intelligence*. Sistem ini berfungsi untuk mengukur, memonitor, dan mengelola kinerja perguruan tinggi secara lebih efektif. *Business intelligence* pada perguruan tinggi juga sebagai sarana evaluasi kinerja untuk pengambilan keputusan akurat yang digunakan untuk penyusunan perencanaan strategis perguruan tinggi di masa yang akan datang.

Mgs. Afriyan Firdaus, Apriansyah Putra, dan Dwi Rosa Indah (2013) dalam penelitiannya yang berjudul “Analisis *Business Intelligence* pada Pengelolaan Data Alumni: Upaya Mendukung Monitoring Kualitas Alumni di Perguruan Tinggi (Studi Kasus di Fakultas Ilmu Komputer Universitas Sriwijaya)” menganalisa tentang pentingnya penerapan sistem *business intelligence* dalam meningkatkan kualitas alumni perguruan tinggi. Hasil yang dicapai dalam penelitian ini adalah suatu usulan pembuatan *business intelligence* yang dapat memonitor tren alumni tiap tahunnya dengan adanya kesepakatan dari peneliti dan pihak perguruan tinggi.

Kurniawan Jatmika, Adhistya Erna P., dan Ari Cahyono (2015) dalam penelitiannya berjudul “Rancang Bangun Data Mart dan Purwarupa *Dashboard* untuk Visualisasi Performa Akademik” membangun sebuah *prototype dashboard* akademik di STIKOM Surabaya berbasis web dengan menggunakan *star scheme*. *Star scheme* dianggap mampu membantu dalam pembangunan sistem ini. Dalam penelitian ini juga digunakan indikator-indikator penentu sebagai acuan pembuatannya seperti, IPK, SKS, kelas, dan sebagainya.

Hendro Poerbo Prasetya dan Meme Susilowati (2016) telah melakukan penelitian “Visualisasi Informasi Data Perguruan Tinggi dengan *Data Warehouse*

dan *Dashboard System*” membuat sebuah sistem *dashboard* pada Visual Studio. Peneliti ini menggunakan adanya *Key Performance Indicator* (KPI) untuk menentukan indikator-indikator yang diperlukan dari presentasi yang dibuat. Indikator dibuat beserta nilai parameternya sehingga memudahkan dalam menentukan kualitas dari objek yang diteliti. Data dipresentasikan dalam bentuk grafik batang, grafik garis, dan *gauge bar*.

Fanny Fathya Nurul Fatimah, Murahartawaty, dan Adityas Widjajarto (2014) dalam penelitiannya “Penerapan *Business Intelligence* pada Aplikasi *Dashboard Monitoring* Performansi Mahasiswa dan Lulusan Berdasarkan Standar 3 BAN-PT Program Studi Sarjana Menggunakan Metode *Scrum*” membuat suatu sistem *dashboard* menggunakan salah satu metode pengembangan aplikasi, yaitu *Scrum*. Sistem ini bertujuan untuk memonitor performansi mahasiswa sehingga dapat menjadi acuan dalam pengambilan keputusan yang tepat dan akurat. Pada sistem ini juga disinkronisasikan dengan standar yang diterbitkan oleh BAN-PT terkait mahasiswa dan lulusan. Penggunaan metode *scrum* dianggap efektif dalam pembangunan sistem ini karena menerapkan *real-word progress* yang dianggap sesuai dengan sistem yang memiliki skala penjadwalan tertentu.

Josef Bernadi dan Suharjito (2016) telah melakukan penelitian “*Executive Information System Modelling to Monitor Indonesian Criminal Rate*”. Dalam penelitiannya, dibuat suatu rancangan sistem yang dapat memudahkan para eksekutif atau petinggi kepolisian untuk memantau tingkat kriminalitas di Indonesia. Sistem ini menggunakan *Executive Information System* (EIS) atau biasa disebut sistem *dashboard* yang nantinya akan menyajikan data-data dalam bentuk diagram untuk memudahkan proses pengambilan keputusan. Sistem ini menggunakan metode *Software Development Life Cycle* (SDLC) yaitu *prototype*. Dijelaskan dalam penelitian ini bahwa metode *prototype* merupakan salah satu dari metode *Rapid Application Development* (RAD).

Muhammad Yazid, Slamet Riyadi, dan Asroni (2015) dalam penelitiannya “Pembangunan *Data Mart* Lulusan Dengan Arsitektur *Normalized Data Store* dan

*Dimensional Data Store*” telah membangun suatu *data mart* berarsitektur *normalized data store* dan *dimensional data store* pada data lulusan di Universitas Muhammadiyah Yogyakarta. Pada sistem ini bertujuan menyajikan data lulusan di Universitas Muhammadiyah Yogyakarta dalam rentang waktu 5 tahun *reporting* yang berupa tabel berdasar dengan tahun ajaran yang telah ditentukan dengan menggunakan *Microsoft Excel*. Dalam sistem ini ditentukan juga *attribute* yang menjadi tolak ukur dalam menentukan jumlah lulusan dalam aktivitas tertentu.

Ricky Akbar, Azizi Soniawan, Rafel Dinur, Jovi Adrian, Rafki Azim, dan Afdhal Zikri (2017) telah melakukan penelitian “Implementasi *Business Intelligence* untuk Menganalisis Data Persalinan Anak di Klinik Ani Padang dengan Menggunakan Aplikasi *Tableau Public*” membuat sistem *dashboard* menggunakan aplikasi *Tableau Public* dengan menghasilkan beberapa grafik sebagai bahan analisis dalam data persalinan anak. Sistem ini menampilkan *tren* dari persalinan anak di Klinik Ani Padang sebagai pedoman dalam pengambilan keputusan untuk meningkatkan nilai bisnis di klinik tersebut. Sistem ini menggunakan indikator jumlah kelahiran berdasar jenis kelamin dan waktu kelahiran.

Euis Nina Saparina Yuliani, Heru Subawanto, dan Anggi Oktaviani (2017) dalam penelitiannya “*Business Intelligence Dashboard Implementation on a Travel Agency in Jakarta*” membuat *dashboard* untuk membantu pengambilan keputusan pada agen travel. Sistem ini menggunakan aplikasi *QlikView* dalam implementasinya. Sistem ini dapat memilih nama tur dan menampilkan informasi terkait berupa grafik batang pada satu halaman.

P. Ziuziański, M. Furmankiewicz, dan A. Sołtysik-Piorunkiewicz (2014) telah melakukan penelitian “*E-Health Artificial Intelligence System Implementation: Case Study of Knowledge Management Dashboard of Epidemiological Data in Poland*” membuat suatu sistem yang dapat memonitor perkembangan penyakit-penyakit yang menular di Polandia. Sistem ini dibuat dengan gabungan ilmu *e-health* dan *business intelligence* dengan menampilkan

data-data tersebut kedalam bentuk grafik sehingga memudahkan dalam melakukan proses pemantauan. Dalam pembuatannya menggunakan metode *dashboard lifecycle*. Sistem *dashboard* ini juga menerapkan adanya *key performance indicator* dalam pengembangannya.

Dari penelitian-penelitian di atas, dapat disimpulkan bahwa pentingnya sistem pembantu pengambilan keputusan dapat meningkatkan kualitas suatu instansi atau perusahaan dengan menggunakan *business intelligence*. Dalam sistem pembantu pengambilan keputusan diperlukan adanya visualisasi pada informasi terkait agar dapat memonitor dalam jangka waktu tertentu berupa grafik-grafik. Dalam tahapan pembuatannya dibutuhkan indikator kinerja kunci. Untuk implementasi pembangunan menggunakan berbagai macam aplikasi pendukung seperti, *QlikView*, *Tableau*, dan *Microsoft Excel*. Sedangkan metode yang digunakan dalam pembangunan sistem *dashboard* adalah SDLC, *dashboard lifecycle*, dan *scrum*.

Dalam penelitian ini menggunakan *business intelligence* sebagai dasar pembangunan sistem pendukung pengambilan keputusan di tingkat eksekutif dan memvisualisasikannya dalam bentuk grafik-grafik. Dalam implementasinya menggunakan aplikasi *Tableau*. Metode yang digunakan dalam penelitian ini adalah *dashboard lifecycle* dengan menggunakan *key performance indicator* sebagai indikator penentu dari *dashboard* pada penelitian ini. Selain membangun sistem *dashboard*, penelitian ini juga melakukan pembuatan *data warehouse* dengan arsitektur NDS + DDS.

## **2.2. Landasan Teori**

### **2.2.1 Database**

(Elmasri & Navathe, 2011) *Database* adalah kumpulan dari data-data yang memiliki keterkaitan. Data yang dimaksud adalah merupakan fakta-fakta yang diketahui yang dapat direkam dan memiliki maksud implisit. Sebagai contoh; nama, nomor telepon, dan alamat dari orang-orang yang kita kenal. Informasi-informasi

ini dapat direkam dalam buku alamat atau dapat disimpan dalam sebuah *hard drive*, menggunakan PC dan *software* seperti *Microsoft Access* atau *Excel*. Kumpulan dari data-data terkait dengan maksud implisit ini disebut *database*. *Database* memiliki sifat implisit sebagai berikut:

- a. *Database* mewakilkan beberapa aspek dari dunia nyata, terkadang disebut dunia kecil atau *universe of discourse* (UoD). Perubahan dalam dunia kecil digambarkan dalam *database*.
- b. *Database* adalah kumpulan data yang logis dan koheren dengan beberapa makna yang melekat. Berbagai macam data *random* tidak dapat disebut sebagai *database*.
- c. *Database* didesain, dibangun, dan diisi untuk tujuan tertentu. *Database* memiliki kelompok *user* yang dituju dan beberapa aplikasi yang membuat *user* tertarik.

### **2.2.2 Database Management System (DBMS)**

(Elmasri & Navathe, 2011) *Database management system* (DBMS) adalah kumpulan dari program-program yang mampu membuat *user* untuk membuat dan mengelola *database*. DBMS adalah sebuah sistem *software* yang memiliki fungsi umum yang memfasilitasi proses dari penggambaran, pembangunan, pemanipulasian, dan pembagian *database* diantara beberapa *user* dan aplikasi. Berikut adalah penjelasannya:

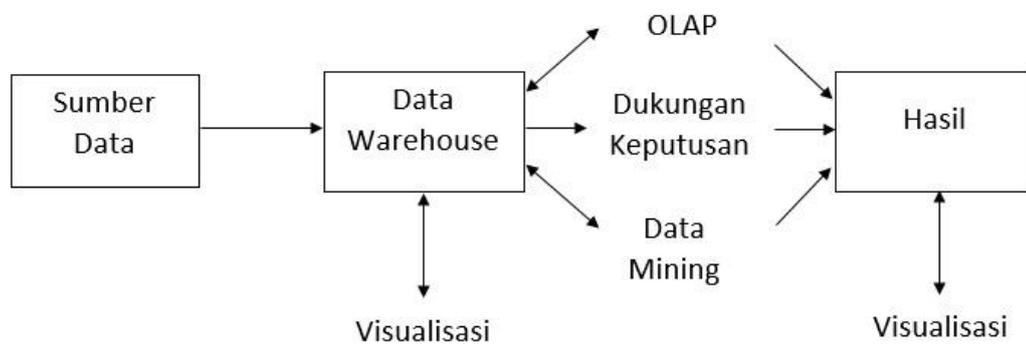
- a. Penggambaran *database* termasuk diantaranya adalah merincikan tipe data, struktur, dan batasan dari data yang tersimpan dalam *database*. Penggambaran *database* atau informasi deskriptif juga tersimpan dengan DBMS di dalam bentuk dari katalog *database* atau kamus yang biasa disebut *metadata*.
- b. Pembangunan *database* adalah proses dari penyimpanan data pada beberapa media penyimpanan yang dikendalikan oleh DBMS.

- c. Pemanipulasian *database* termasuk diantara fungsinya adalah pembuatan *query database* untuk mendapatkan kembali rincian data, pembaruan *database* untuk menggambarkan perubahan di dalam dunia kecil, dan membuat laporan dari data tersebut.
- d. Pembagian *database* mengizinkan untuk beberapa *user* dan program untuk mengakses *database* secara berkala.

Fungsi penting lainnya yang disediakan oleh DBMS termasuk diantaranya adalah menjaga *database* dan mengelolanya dalam jangka waktu yang lama. Penjagaan termasuk diantaranya penjagaan sistem dari kesalahan atau kerusakan *hardware* atau *software* dan penjagaan *security* dari akses yang tidak dikenal atau akses yang buruk. Tipikal *database* yang besar memungkinkan memiliki sebuah *life cycle* beberapa tahun, sehingga DBMS harus dapat mempertahankan sistem *database* dengan memungkinkan sistem berevolusi seiring dengan berubahnya persyaratan dari waktu ke waktu.

### **2.2.3 Business intelligence**

*Business intelligence* adalah kumpulan inovasi teknis dan proses pada *data warehouseing* dan ruang *business intelligence* itu sendiri. Dalam *business intelligence* kita dapat memperoleh informasi dan data bahkan suatu pengetahuan dari beberapa sumber data dan digunakan sebagai pengambilan keputusan (Turban, Aronson, & Liang, 2005). Aktivitas *business intelligence* dipandang sebagai sebuah strategi yang berkembang secara terus-menerus, sebuah visi dan arsitektur yang selalu disinkronisasi dengan tujuan bisnis strateginya. Pada *business intelligence*, data sumber yang tersedia akan diolah menjadi *data warehouse* dan menghasilkan suatu visualisasi data. Dari *data warehouse* yang ada dapat diolah menjadi 3 (tiga) sajian data; OLAP, dukungan keputusan (*decision support*), dan *data mining*. Kemudian, melalui olahan data tersebut data baru dapat divisualisasikan, atau dapat langsung divisualisasikan setelah adanya *data warehouse*. Berikut adalah gambaran dari aktivitas *business intelligence* pada Gambar 2.1.



**Gambar 2.1** Aktivitas *Business Intelligence*

(Sumber: Turban dkk., 2005)

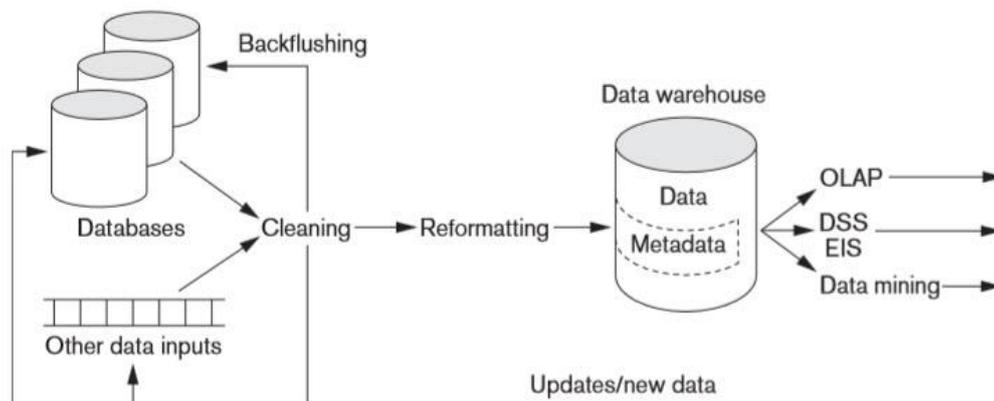
#### 2.2.4 *Data warehouse*

*Data warehouse* adalah suatu sistem yang mengambil dan menggabungkan data pada waktu tertentu dari data sumber kedalam suatu *normalized data store* atau *dimensional data store*. *Data warehouse* biasanya menyimpan data historis menahun dan digunakan untuk *business intelligence* dan aktivitas analisis lainnya. Pada *data warehouse* tidak setiap waktu pada suatu transaksi terjadi di data sumber, biasanya data di-*update* pada kumpulan-kumpulan (Reinardi, 2008). Sedangkan menurut M.H. Inmon, *data warehouse* memiliki karakteristik sebagai sekumpulan data yang berorientasi subyek, terintegrasi, tetap, dan memiliki variasi waktu dalam mendukung pengambilan keputusan (*decision support*). *Data warehouse* mendukung permintaan yang berkinerja tinggi pada data dan informasi sebuah organisasi. *Data warehouse* memiliki kemampuan analisis, memberikan informasi, dan pengambilan keputusan. Beberapa pengaplikasian yang didukung dari *data warehouse* ini adalah OLAP, DSS, dan *data mining*. (Elmasri & Navathe, 2011).

Jika dibandingkan dengan *database* transaksi, data pada *data warehouse* tidak mudah berubah. Hal ini menandakan bahwa perubahan informasi yang ada di dalam *data warehouse* jauh lebih sedikit dan dapat dianggap sebagai data yang memiliki waktu yang tidak sebenarnya dengan adanya pembaruan berkala. Dalam sistem transaksional, transaksi merupakan suatu unit dan agen perubahan ke *database*. Sedangkan pada informasi *data warehouse* cenderung lebih kasar dan

diperbarui sesuai dengan pilihan kebijakan perbaruan, dan biasanya merupakan tambahan *user*. Pembaruan *warehouse* diatur berdasar dengan perolehan komponen dari *warehouse* yang menyediakan semua kebutuhan sebelum proses berlangsung.

Dapat disimpulkan bahwa *data warehouse* merupakan teknologi pendukung pengambilan keputusan (*decision support*), ditujukan untuk memberi informasi kepada pekerja (eksekutif, manajer, analis) untuk membuat keputusan yang lebih baik dan lebih cepat. Pada Gambar 2.2 memberikan gambaran dari struktur konseptual dari *data warehouse*. Gambar tersebut menunjukkan seluruh proses pada *data warehouse* yang di dalamnya termasuk pembersihan dan pemformatan ulang dari data sebelum dimasukkan ke dalam *data warehouse*. Proses ini biasa juga disebut ETL (*extract, transform, load*) yang akan dijelaskan lebih lanjut dalam pembahasan selanjutnya. Pada akhir dari proses, OLAP, *data mining*, dan DSS dapat membuat informasi yang relevan dari data.



**Gambar 2.2** Karakteristik *data warehouse*

(Sumber: Elmasri & Navathe, 2011)

Dari penjelasan sebelumnya, dapat disimpulkan bahwa karakteristik khusus dari *data warehouse* adalah sebagai berikut:

- Berjalan konsep multidimensional
- Memiliki sifat dimensi yang generik (umum)
- Memiliki dimensi dan level agregasi yang tidak terbatas
- Memiliki operasi lintas dimensi yang tidak terbatas
- Memiliki penanganan matriks *sparse* yang dinamik

- Memiliki arsitektur klien-*server*
- Mendukung *multiuser* (jumlah *user* lebih dari satu)
- Aksesibilitas
- Transparansi
- Manipulasi data yang intuitif
- Memiliki kinerja pelaporan yang konsisten

### **2.2.5 *Decision Support System (DSS)***

(Turban dkk., 2005) *Decision Support System (DSS)* atau bisa disebut sistem pendukung pengambilan keputusan didefinisikan oleh Little (1970) sebagai “sekumpulan prosedur berbasis model untuk pemrosesan dan penilaian data guna membantu para manajer mengambil keputusan.” Little menyatakan bahwa untuk sukses, sistem tersebut haruslah sederhana, cepat, mudah dikontrol, adaptif, lengkap dengan isu-isu penting, dan mudah berkomunikasi. Keen (1980) mendefinisikan DSS sebagai suatu produk dari proses pengembangan di mana pengguna DSS, pembangun DSS, dan DSS itu sendiri mampu memengaruhi satu dengan yang lainnya, dan menghasilkan evolusi sistem dan pola-pola penggunaan.

Aplikasi DSS terdiri dari beberapa subsistem, salah satunya adalah subsistem manajemen data. Subsistem manajemen data memasukkan sebuah *database* yang berisi data yang relevan untuk situasi tertentu dan dikelola oleh suatu perangkat lunak yang disebut *Database Management System (DBMS)*. Subsistem manajemen data dapat diinterkoneksi dengan *data warehouse* perusahaan, suatu repositori untuk data perusahaan yang relevan untuk pengambilan keputusan.

### **2.2.6 *Key Performance Indicator (KPI)***

*Key Performance Indicator* atau KPI adalah sebuah indikator yang menyajikan performa dari proses yang dilakukan. KPI merupakan kumpulan ukuran kinerja yang paling kritis sebagai penentu keberhasilan suatu organisasi baik di waktu sekarang ataupun di waktu yang akan datang. KPI adalah suatu tolok ukur prediksi peluang keberhasilan atau kegagalan dari proses ataupun program yang

dilaksanakan organisasi, sehingga dapat menjadi indikasi peningkatan kinerja suatu organisasi (Prasetiya & Susilowati, 2016).

KPI juga dapat didefinisikan sebagai alat ukur yang menentukan sebaik apa proses yang berlangsung pada sistem untuk mencapai tujuan. KPI merupakan indikator utama yang mempresentasikan apakah tujuan tersebut dapat dicapai serta merupakan indikator dari kemampuan, penerapan, dan keterampilan. KPI mengukur tujuan dari aktivitas yang merupakan tindakan yang harus dilakukan oleh pemilik proses untuk mendapatkan proses kinerja yang efektif. (IT Governance Institute, 2007)

### **2.2.7 Borang akreditasi**

Borang atau biasa yang disebut formulir adalah sebuah kertas yang berisi beberapa pertanyaan formal yang harus diisi (Wikipedia, 2016). Adapun keuntungan dari penggunaan formulir adalah:

- Orang tidak perlu menulis sembarangan (sehingga mencetak hasilnya merupakan suatu yang standar dan bisa dilakukan secara otomatis)
- Orang akan mengerti apa yang perlu dijawab dan tidak perlu dijawab dari pertanyaan tersebut.

Salah satu kegunaan borang adalah untuk standar penilaian institusi. Pada penelitian ini menggunakan borang akreditasi perguruan tinggi oleh BAN-PT Akreditasi Perguruan Tinggi Buku III Pedoman Penyusunan Borang tahun 2011. Borang akreditasi ini diunduh pada *website* BAN-PT (BAN-PT, 2016). Terdapat tujuh standar akreditasi institusi perguruan tinggi, yaitu:

1. Standar 1: Visi, misi, tujuan dan sasaran, serta strategi pencapaian
2. Standar 2: Tata pamong, kepemimpinan, sistem pengelolaan, dan penjaminan mutu
3. Standar 3: Mahasiswa dan lulusan
4. Standar 4: Sumber daya manusia

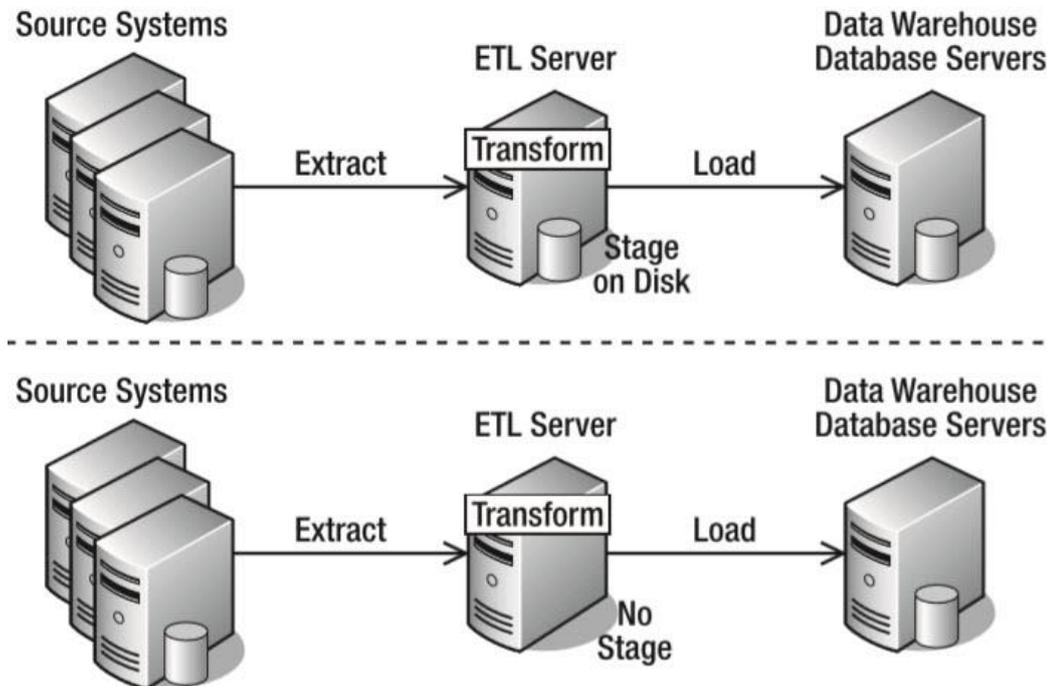
5. Standar 5: Kurikulum, pembelajaran, dan suasana akademik
6. Standar 6: Pembiayaan, sarana dan prasarana, serta sistem informasi
7. Standar 7: Penelitian, pelayanan/pengabdian kepada masyarakat, dan kerjasama

Salah satu poin penilaian adalah pada standar 3 bagian Mahasiswa dan Lulusan yang menjadi acuan dalam penelitian ini. Penelitian ini terfokus pada poin 3.2 bagian lulusan sebagai alat untuk mengukur kualitas lulusan pada suatu instansi.

### **2.2.8 *Extract, Transform, Load (ETL)***

Proses *Extract, Transform, dan Load* atau ETL adalah suatu sistem yang memiliki kapabilitas untuk mengintegrasikan, menarik, mengubah, dan mengelola data sumber ke *data warehouse* (Reinardi, 2008). Penarikan data biasanya dilakukan secara periodis dalam jangka waktu tertentu. Data juga akan diubah dan terjadi penyamaan formatnya seperti, perubahan format bentuk nominal ke dalam bentuk tanggal atau perubahan bentuk nominal ke dalam bentuk teks.

Ada beberapa pendekatan dalam implementasi ETL. Pendekatan tradisional adalah dengan menarik data dari sumber data, meletakkannya pada *staging area*, dan kemudian mengubahnya dan memasukkannya ke dalam *data warehouse*, seperti yang ditunjukkan pada diagram yang atas dari Gambar 2.3. Cara lain, dibandingkan dengan meletakkan data pada *staging area*, terkadang *server ETL* melakukan transformasi ke dalam memori dan memperbarui *data warehouse* secara langsung (tanpa adanya *staging*), seperti yang ditunjukkan pada diagram yang bawah dari Gambar 2.3. *Staging area* adalah *database* fisik atau berupa *file*. Meletakkan data ke dalam *staging area* berarti memasukkannya ke dalam *database* atau menuliskannya ke dalam suatu *file*.

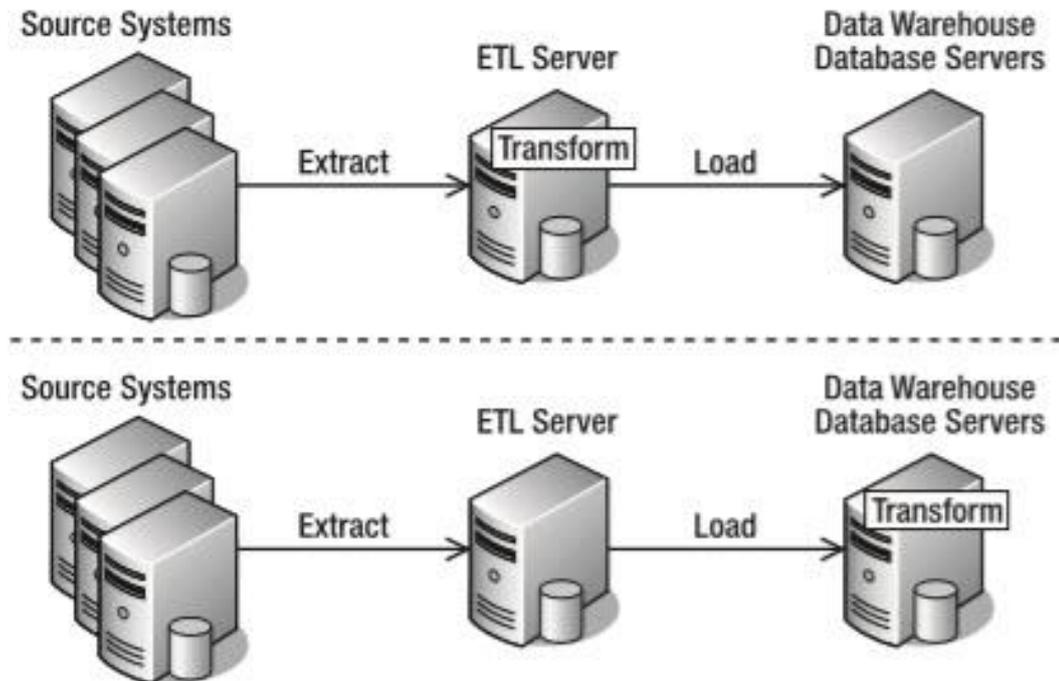


**Gambar 2.3** Diagram ETL

(Sumber: Reinardi, 2008)

Mentransform data dalam memori lebih cepat dibandingkan meletakkannya pada *disk* dulu. Jika data berukuran kecil, maka dapat mentransformnya dalam memori, tapi jika data berukuran besar, maka diperlukan untuk meletakkannya dalam *disk* terlebih dahulu. Seseorang dapat meletakkan data tersebut dalam memori sesuai memori yang ada pada *server*.

Pendekatan ETL yang lain disebut ELT (*Extract, Load, Transform*) seperti yang ditunjukkan pada diagram yang bawah pada Gambar 2.4. Pada pendekatan ELT, data ditarik dari sumber data, memasukkannya pada *data warehouse*, dan melakukan transformasi dengan memperbarui data pada *warehouse*. Pada pendekatan ELT, pada dasarnya menyalin data pada sumber data kedalam *data warehouse* dan melakukan transformasi didalamnya. Kebanyakan orang menggunakan pendekatan ETL jika memiliki *server* ETL yang kuat dan *software* yang kuat yang memiliki banyak jenis proses transformasi dan *data quality*.



**Gambar 2.4** Perbedaan ETL dengan ELT

(Sumber: Reinardi, 2008)

Pada penelitian ini digunakan pendekatan ETL tradisional dengan menggunakan media *database* sebagai *staging area* dalam prosesnya. Penjelasan tentang *stage* akan dijelaskan dalam penjelasan selanjutnya.

### 2.2.9 Stage

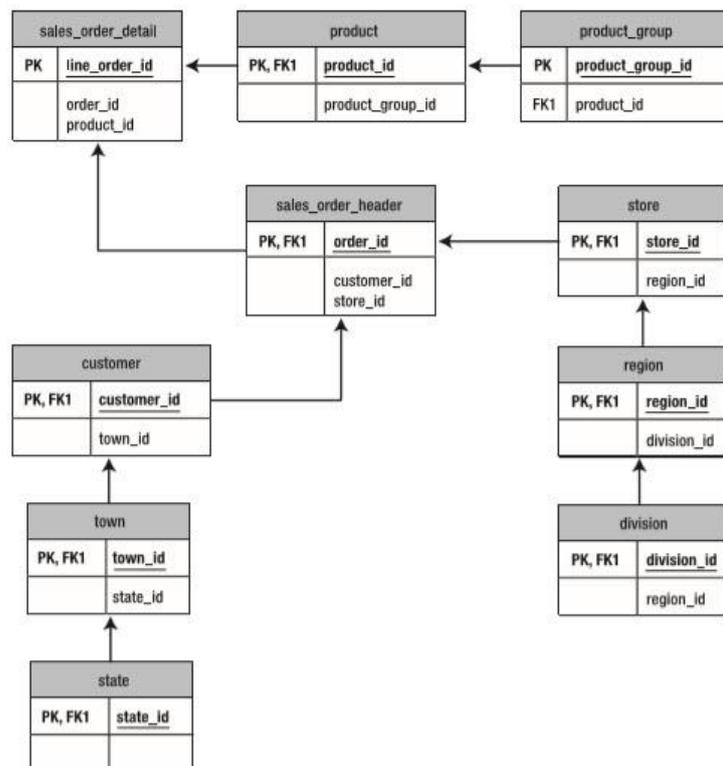
*Stage* adalah suatu wadah yang menyimpan data yang telah diekstrak dari data sumber sementara sebelum diproses lebih lanjut. *Stage* diperlukan ketika proses transformasi tersebut kompleks (dalam kata lain tidak dapat diselesaikan secara cepat dalam sekali proses dalam memori), ketika volume data tersebut besar (tidak cukup disimpan pada memori), atau ketika data dari beberapa data sumber diekstrak dalam sekali proses ETL. *Stage* juga diperlukan jika dibutuhkan waktu yang minim untuk mengekstrak data dari data sumber (Reinardi, 2008).

Proses ETL dari data sumber ke *stage* merupakan proses ETL yang tidak ada perubahan data berupa isi ataupun tipe data. Proses ini merupakan proses yang

hanya memindahkan atau mengekstrak data dari data sumber ke sistem lokal untuk selanjutnya dilakukan proses pengolahan *data warehouse* (Yazid dkk., 2015).

### 2.2.10 Normalized Data Store (NDS)

*Normalized data store* adalah satu atau lebih *database* yang berelasi dengan redundansi data yang sedikit atau tidak ada sama sekali. Dalam *data warehouse*, data akan mengalami normalisasi untuk mengurangi redundansi data. Reinardi (2008) menyatakan bahwa normalisasi adalah suatu proses penghilangan data redundansi dengan menggunakan aturan normalisasi. *Normalized data store* biasanya merupakan normalisasi ke tiga atau di atasnya. NDS merupakan salah satu dari *internal data store* yang datanya adalah data mentah dan belum dapat diolah atau di-*query*. Dapat dilihat pada Gambar 2.5.

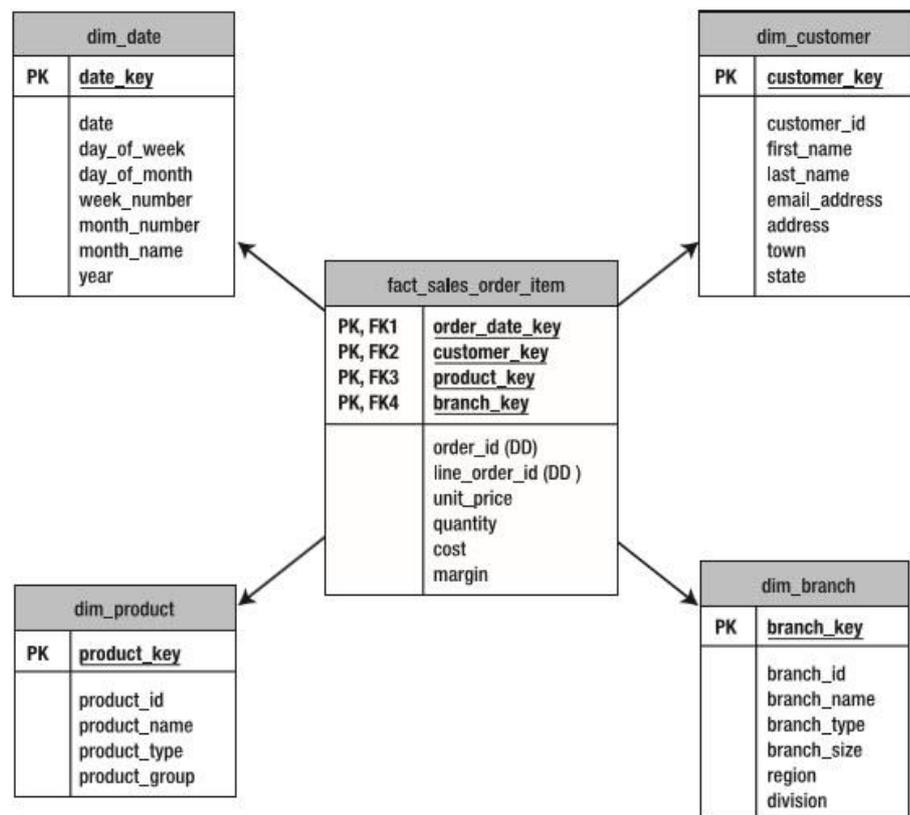


**Gambar 2.5** Normalized Data Store (NDS)

(Sumber: Reinardi, 2008)

### 2.2.11 Dimensional Data Store (DDS)

Menurut Reinardi (2008), *dimension data store* adalah satu atau lebih database yang berisi kumpulan dari *dimensional data mart*. *Dimensional data mart* merupakan kumpulan dari relasi tabel faktual dan tabel dimensi. Di DDS, terdapat proses denormalisasi dan pembuatan tabel-tabel dimensi. Relasi antara tabel faktual dan tabel dimensi berbentuk *star schema* yang berpusat di tabel faktual. *Dimensional data store* merupakan format yang baik untuk menyimpan data pada *warehouse* yang bertujuan untuk *men-query* dan menganalisa data dibanding *normalized data store*. Hal ini disebabkan karena *dimensional data store* lebih sederhana (satu tingkatan lebih dalam di semua arah pada *star schema*) dan memberikan kinerja *query* yang lebih baik. Contoh dari *star schema* ditunjukkan pada Gambar 2.6.



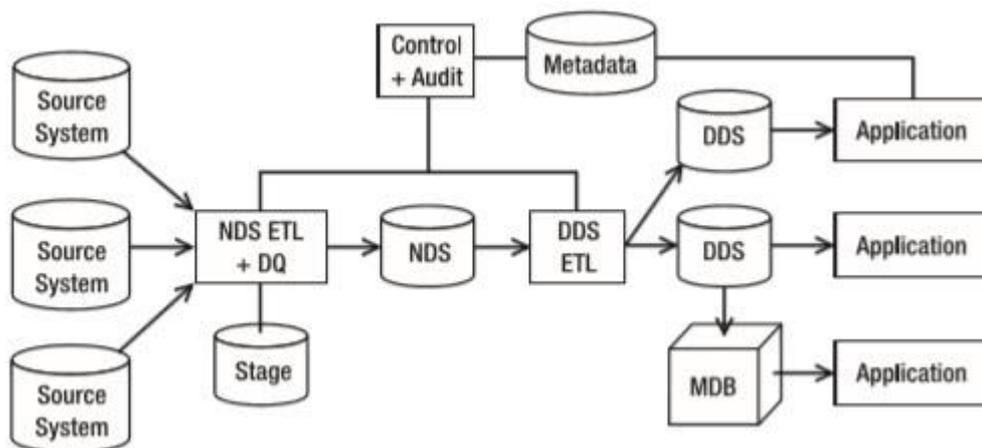
**Gambar 2.6** Star schema Dimensional Data Store (DDS)

(Sumber: Reinardi, 2008)

### 2.2.12 Arsitektur alur data NDS + DDS

(Reinardi, 2008) Pada arsitektur alur data NDS + DDS terdapat 3 (tiga) *data store*, yaitu *stage*, NDS dan DDS. Pada arsitektur ini, terdapat proses normalisasi pada *data store* sebelum pembuatan *dimensional data store* yang berisi tabel dimensi dan tabel faktual. Fungsi NDS pada arsitektur ini ada 2 (dua). Pertama NDS berfungsi untuk mengintegrasikan data dari beberapa data sumber. Kedua, NDS dapat memproses data ke dalam beberapa DDS. Dalam arsitektur NDS + DDS dapat memiliki beberapa DDS. Gambar 2.7 menunjukkan arsitektur alur data NDS + DDS.

Pada arsitektur NDS + DDS ini, NDS merupakan *master data store* yang memiliki arti bahwa NDS berisi beberapa kumpulan data yang lengkap termasuk seluruh data transaksi historis dan seluruh versi historis dari *master data*. Data transaksi historis adalah transaksi bisnis yang terjadi pada masa lampau. Data dari setiap tahun disimpan pada NDS. Sedangkan pada DDS bukan merupakan *master data store* yang berarti tidak berisi seluruh data transaksi tiap tahunnya. DDS tidak berisi tentang semua transaksi pada tiap tahunnya. Setelah data melalui proses DDS, data dapat langsung diolah ke dalam aplikasi lainnya.



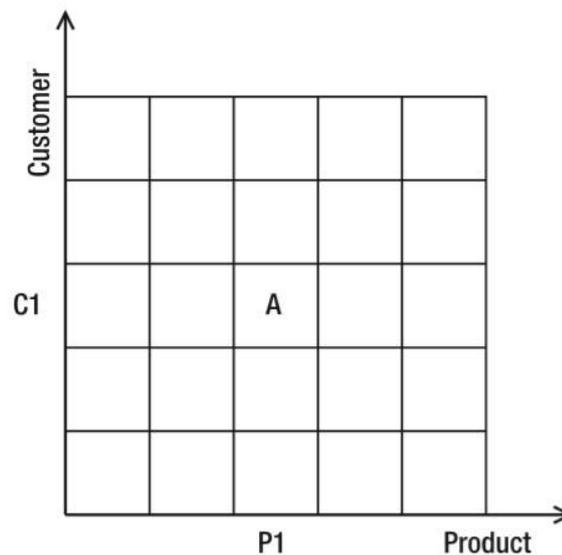
**Gambar 2.7** Arsitektur NDS + DDS

(Sumber: Reinardi, 2008)

Keunggulan utama dari arsitektur ini adalah *data warehouse* yang ada bisa diakomodasi, dan jangka waktu proses mengembangkan lebih pendek. Kekurangan utama dari arsitektur ini adalah dalam praktiknya sulit untuk membangun *warehouse* yang berkualitas baik dari berbagai standar yang ditemukan pada sumber *data mart* atau *data warehouse*.

### 2.2.13 Multidimensional database (MDB)

Reinardi (2008) menyatakan bahwa *multidimensional database* atau *cube* adalah suatu database yang tersimpan dalam suatu *cell* dan letak dari tiap-tiap *cell* ditentukan dari jumlah tingkatan atau biasa disebut dimensi. Setiap sel mewakili sebuah *business event*, dan suatu dimensi menunjukkan waktu dan tempat *event* tersebut terjadi. *Multidimensional database* biasa digunakan untuk analisis data dan data mining. Struktur penyimpanan pada *multidimensional database* merupakan hasil agregasi. Gambar 2.8 adalah ilustrasi dari dimensi suatu *cube*.



**Gambar 2.8** *Multidimensional database*

(Sumber: Reinardi, 2008)

### 2.2.14 Star schema

Reinardi (2008) mengatakan bahwa *dimensional data store* dapat diimplementasikan ke dalam bentuk beberapa *schema*. Salah satu dari contoh

*schema dimensional data store* adalah *star schema* (lihat Gambar 2.6 pada pembahasan DDS). Dalam *star schema*, tiap dimensi tidak memiliki subdimensi atau cabang dari dimensi. Keuntungan dari menggunakan *star schema* adalah untuk membuatnya lebih sederhana dari *schema* yang lain, membuat proses ETL lebih mudah untuk data di-load kedalam *dimensional data store*.

### 2.2.15 Sistem *dashboard*

(Bernadi & Suharjito, 2016) Sistem *dashboard* atau biasa dikenal *Executive Information System* (EIS) adalah sebuah gambaran atau tampilan visual dari kumpulan informasi yang penting untuk mencapai suatu tujuan. EIS adalah suatu sistem yang memeberikan kebutuhan informasi dari para petinggi instansi atau perusahaan. EIS merupakan sistem yang *user-friendly*, berbasis grafis, dan dapat memberikan informasi secara lebih mendalam.

Hal yang terpenting dari suatu sistem *dashboard* adalah adanya tampilan yang informatif yang dapat mewakili informasi yang ada tetapi, tetap singkat, padat, serta jelas sehingga dapat dimonitor dalam sekali lihat. *Dashboard* pada umumnya berupa grafik dan teks tetapi, lebih ditonjolkan adanya grafik. Teks pada *dashboard* hanya sebagai alat bantu untuk memberi informasi tambahan pada grafik.

Integrasi antara *dashboard* atau *Executive Information System* (EIS) dengan DSS menghasilkan sebuah sistem pendukung eksekutif atau *Executive Support System* (ESS) (Turban dkk., 2005). Turban dkk. telah menyimpulkan bahwa EIS berbeda dengan DSS. Namun, terdapat manfaat dalam integrasi kedua sistem tersebut yang digunakan untuk pengambilan keputusan di perusahaan. DSS diberi input dari data awal yang sama dengan input yang diberikan ke EIS (misal *data warehouse*), tetapi tindakan DSS dipicu oleh EIS. Dapat disimpulkan bahwa untuk mendapatkan sistem pendukung pengambilan keputusan perlu adanya EIS.

### 2.2.16 Grafik Bisnis

Meilach (1986) mengatakan bahwa grafik dan bagan yang merupakan penggambaran dari sebuah performa merupakan dasar dari grafik bisnis. Kolom (*columns*), batang (*bars*), *pie*, peta (*maps*), dan garis yang bergelombang (*wiggly*

*line*) merupakan elemen yang digunakan untuk merepresentasikan data analitik, membandingkan angka, menampilkan dimana kejadian dan bagaimana data tersebut terorganisir, dan mengindikasikan relasi dari sistem-sistem yang berbeda.

Dalam pembuatan grafik analitis, ada beberapa hal yang perlu diperhatikan dalam pembuatannya, yaitu:

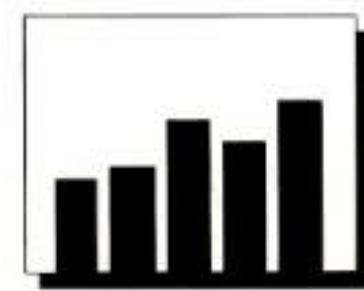
- a. Sederhana (*simplicity*)
- b. Adanya penekanan (*emphasis*)
- c. Merupakan satu kesatuan (*unity*)
- d. Seimbang (*balance*)
- e. Jarak (*spacing*)
- f. Skala (*scale*)
- g. Bercorak dan berwarna (*shade and color*)
- h. Bertekstur dan berpola (*texture and pattern*)
- i. Menggunakan garis *grid* (*grid line use*)
- j. Ketebalan garis (*line thickness*)
- k. Peletakan data dan tanda *tic* (*data and tic mark placement*)
- l. Peletakan angka dan label (*placement of numbers and labels*)

Adapun jenis dari grafik sendiri bermacam-macam sesuai dengan kebutuhan visualisasi data. Berikut adalah macam-macam grafik:

1. Grafik batang (*bar charts*)

Grafik batang memiliki dua jenis, yaitu vertikal dan horizontal. Grafik batang vertikal biasa disebut “grafik kolom.” Grafik ini digunakan pada data yang memerlukan adanya perbandingan dengan data yang lain

dalam bentuk yang sederhana, merupakan gabungan beberapa relasi, dan dengan skala yang sudah ditentukan. Lihat Gambar 2.9.

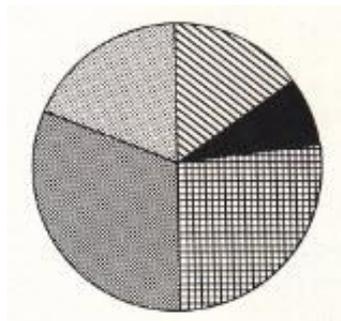


**Gambar 2.9** *Bar chart*

(Sumber: Meilach, 1986)

## 2. Grafik *pie* (*pie chart*)

Grafik *pie* menggambarkan presentase suatu bagian terhadap nilai keseluruhan. Tiap-tiap bagian dibedakan dengan warna atau takstur. Tiap-tiap bagian dari keseluruhan memiliki karakternya masing-masing yang ditunjukkan dari corak dan warna yang berbeda. Lihat Gambar 2.10.



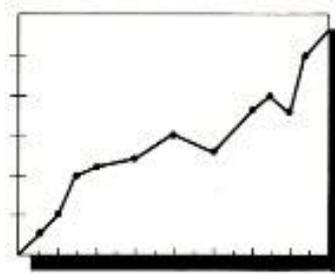
**Gambar 2.10** *Pie chart*

(Sumber: Meilach, 1986)

## 3. Grafik garis (*Line chart*)

Grafik garis menunjukkan tren, penekanan naik atau turunnya, dan arah perubahan pada suatu waktu. Grafik ini menghubungkan titik-titik hasil

relasi obyeknya menjadi satu garis lurus yang saling menghubungkan, seperti pada Gambar 2.11.



**Gambar 2.11** *Line charts*

(Sumber: Meilach, 1986)