

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Pengantar

Tinjauan pustaka berfungsi untuk memberikan dasar teori bagi sebuah penulisan karya ilmiah. Adapun dasar teori yang melandasi penulisan skripsi ini antara lain sebagai berikut:

1. **Sistem manajemen kualitas**, pembahasan pada sub bab ini antara lain mengenai definisi kualitas, kualitas produk, faktor penentu kualitas produk, dan pengendalian kualitas produk.
2. **Traceability**, pembahasan pada sub bab ini antara lain mengenai definisi, konsep & model *traceability*, *traceability* produk, langkah-langkah *traceability*, faktor-faktor *traceability*, *traceability database*, dan *traceability* pada proses produksi.
3. **Sistem produksi**, pembahasan pada sub bab ini antara lain mengenai konsep dasar sistem produksi, klasifikasi proses produksi, karakteristik aliran produk, dan diagram alir proses.
4. **Sistem identifikasi produk**, pembahasan pada sub bab ini antara lain mengenai klasifikasi produk, GS1 Indonesia, sistem EAN-UCC, identifikasi produk secara global, dan *barcode*.
5. **Database**, pembahasan pada sub bab ini antara lain mengenai sejarah *database*, hierarki data, tipe *file*, tujuan sistem *database*, komponen sistem *database*, *database* relasional, *Entity Relationship Diagram* (ERD), normalisasi, langkah-langkah perancangan *database*, sistem manajemen *database* (DBMS), *Structured Query Language* (SQL), *Microsoft Access*, dan *Microsoft Visual Basic* 6.0.

#### 2.2 Sistem Manajemen Kualitas

##### 2.2.1 Definisi Kualitas

Ada banyak sekali definisi kualitas, yang sebenarnya satu definisi hampir sama dengan definisi yang lain. Definisi kualitas menurut beberapa ahli yang sudah banyak dikenal antara lain:

1. **Menurut Juran (1962)**, kualitas adalah kecocokan penggunaan produk (*fitness for use*) dalam memenuhi kebutuhan dan kepuasan pelanggan.
2. **Menurut Crosby (1979)**, kualitas adalah *conformance to requirement*, yaitu sesuai dengan yang disyaratkan atau distandarkan. Standar kualitas meliputi bahan baku, proses produksi, dan produk jadi.
3. **Menurut Deming (1982)**, kualitas adalah kesesuaian dengan kebutuhan pasar atau konsumen. Perusahaan harus benar-benar memahami apa yang dibutuhkan konsumen atas suatu produk yang akan dihasilkan.
4. **Menurut Goetch dan Davis (1995)**, kualitas adalah suatu kondisi dinamis yang berkaitan dengan produk, pelayanan, orang, proses, dan lingkungan yang memenuhi atau melebihi apa yang diharapkan.
5. **Perbendaharaan istilah ISO 8402 dan dari Standar Nasional Indonesia (SNI 19-8402-1991)**, kualitas adalah keseluruhan ciri dan karakteristik produk atau jasa yang kemampuannya dapat memuaskan kebutuhan, baik yang dinyatakan secara tegas maupun tersamar.

### 2.2.2 Kualitas Produk

Kualitas suatu produk merupakan salah satu faktor penting dalam meningkatkan daya saing bisnis. Menciptakan produk yang berkualitas bukan merupakan hal yang serba kebetulan (*occur by accident*). Untuk mencapai produk yang berkualitas, perusahaan harus membuat perencanaan, melaksanakannya, dan mengawasinya secara total. Memandang kualitas suatu produk bisa ditinjau dari dua sisi, yaitu sisi sebagai konsumen dan sisi sebagai produsen. Ditinjau dari sisi sebagai konsumen, kualitas berkaitan dengan faktor-faktor sebagai berikut:

1. **Kualitas produk itu sendiri**, yaitu apakah karakteristik dari produk telah sesuai dengan keinginan konsumen.
2. **Harga**, yaitu apakah biaya yang dikeluarkan konsumen telah sesuai dengan karakteristik produk yang diinginkan.
3. **Ketersediaan**, yaitu apakah produk dapat diperoleh secara mudah baik waktu maupun lokasinya.
4. **Pelayanan**, yaitu adanya jaminan terhadap kualitas produk dan ada garansi jika produk tidak sesuai dengan spesifikasinya.

Sedangkan ditinjau dari sisi sebagai produsen, kualitas menyangkut beberapa faktor sebagai berikut:

1. Merancang (*to design*) produk sesuai dengan permintaan konsumen.
2. Membuat (*to produce*) produk secara baik sesuai dengan perencanaan.
3. Mengirimkan (*to deliver*) produk ke konsumen dalam kondisi baik.
4. Pelayanan (*service*) yang baik kepada konsumen.

### 2.2.3 Faktor Penentu Kualitas Produk

Menurut Suryadi Prawirosentono (2004:16), kualitas suatu produk dipengaruhi oleh faktor-faktor sebagai berikut:

1. Bentuk rancangan dari produk tersebut (*designing*).
2. Bahan baku yang digunakan (*raw material*).
3. Teknologi yang digunakan untuk membuat produk tersebut (*technology*).
4. Cara mengemasnya, cara mengirimnya, dan cara menjualnya kepada konsumen (*packaging, delivering, and selling*).

### 2.2.4 Pengendalian Kualitas Produk

Secara umum pengendalian kualitas dapat digambarkan sebagai suatu kegiatan inspeksi bertahap, mulai dari mengamati, mengumpulkan fakta, kemudian melaksanakan tindakan-tindakan yang perlu dilakukan, dan kegiatan tersebut dilakukan secara terus menerus (*continuous*). Hal itu dilakukan untuk mempertahankan dan meningkatkan kualitas produk sesuai dengan target-target yang telah ditetapkan. Menurut Richardus E. Indrajit (2005:185) bagian pengendalian kualitas terdiri dari:

1. **Research and Development (R&D)**, R&D bertugas melakukan pengembangan atau inovasi produk sebagai respon atas tuntutan pasar yang selalu menginginkan produk yang lebih baik dari waktu ke waktu.
2. **Quality Control (QC)**, QC merupakan sistem untuk menghindari atau mengurangi cacat (*defect*) pada produk melalui kegiatan inspeksi atau pengawasan. Lingkup kegiatan QC antara lain: inspeksi bahan baku, inspeksi pada proses, dan inspeksi pada produk. Kegiatan ini bertujuan agar kualitas produk dapat dikendalikan.

3. **Quality Assurance (QA)**, QA merupakan aktivitas yang mengupayakan agar semua sistem QC berjalan sebagaimana mestinya, sehingga produk yang sampai pada konsumen terjamin kualitasnya. Salah satu bagian penting dari aktifitas QA adalah **identifikasi & kemampuan penelusuran (*traceability*)**. Khusus di lingkup manufaktur, aktivitas penelusuran ini meliputi: penelusuran bahan baku, penelusuran proses produksi, dan penelusuran produk setengah jadi sampai produk jadi. Identifikasi dan kemampuan penelusuran (*traceability*) ini juga merupakan salah satu bagian dari 20 elemen sistem kualitas ISO 9001 yang merupakan sistem kualitas terlengkap dari ISO 9000.

### 2.3 Traceability

#### 2.3.1 Definisi, Konsep, dan Model Traceability

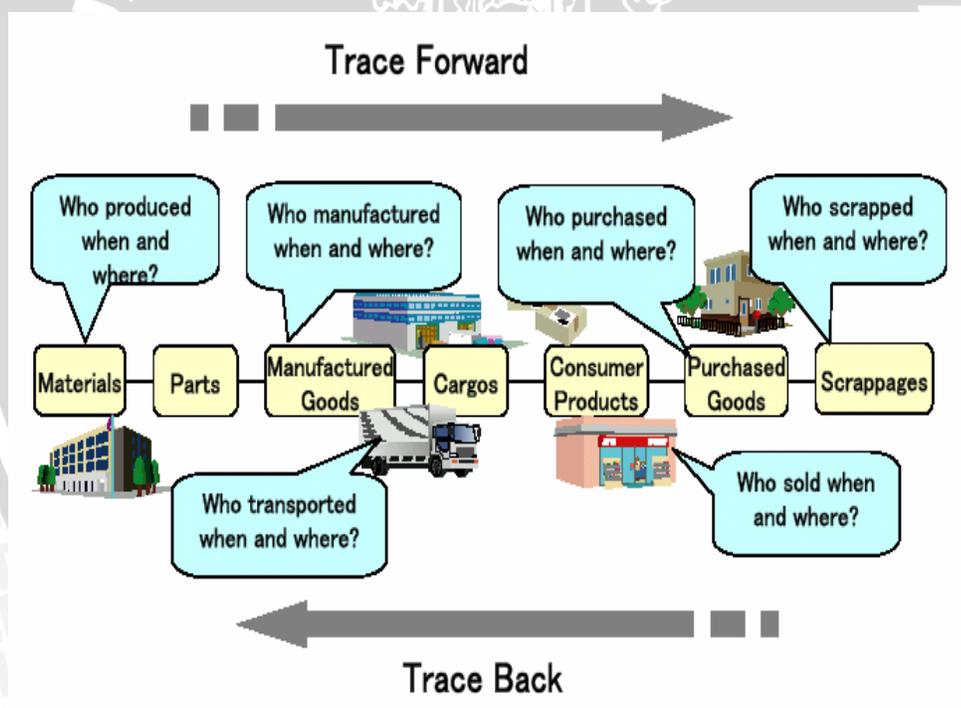
*Traceability* berasal dari kata "trace" yang berarti mengikuti jejak atau melacak dan "ability" yang berarti kemampuan. *Traceability* dijelaskan dalam *International Standardisation Organisation (ISO)* seri 9000:2000 tentang *Quality Management Systems* yang menyebutkan bahwa "Bila sesuai, organisasi perlu mengidentifikasi status produk atau layanan yang berhubungan dengan keperluan aktivitas verifikasi dan pengukuran, pengidentifikasian produk atau layanan itu dengan menggunakan cara atau alat yang sesuai pada seluruh proses, ini berlaku untuk semua komponen yang terlibat dalam produk atau layanan tersebut. Ketika *traceability* menjadi suatu kebutuhan, organisasi perlu mengendalikan dan merekam identitas yang unik dari produk atau layanan yang dihasilkan".

Ada berbagai macam definisi *traceability*, tetapi saat ini banyak perusahaan yang masih mengacu pada ISO seri 8402:1994 yang sebenarnya telah digantikan oleh ISO seri 9000:2000. Pada ISO seri 8402:1994 tentang *Quality Management and Quality Assurance*, definisi *traceability* adalah kemampuan dalam melacak sejarah berdasarkan keseluruhan informasi yang sudah direkam.

Dalam memahami definisi *traceability*, perlu mengetahui konsep dasar *traceability* yang dikenal dengan istilah *tracking* atau *trace forward* dan *tracing* atau *trace back*. *Tracking* atau pelacakan ke depan adalah kemampuan untuk melacak dengan mengidentifikasi suatu *batch* atau unit produk melalui jalur mata

rantai, dari bahan baku yang berasal dari *supplier* sampai pada produk jadi yang ada di pasar. Sedangkan *tracing* atau pelacakan ke belakang adalah kemampuan melacak dengan mengidentifikasi asal unit produk atau *batch* melalui jalur mata rantai, dari produk yang ada di pasar sampai pada bahan baku dari *supplier*.

Untuk menerapkan sistem *traceability* yang efektif, maka konsep *tracking* dan *tracing* perlu diterapkan sesuai dengan kebutuhan. Proses *tracking* dilakukan misalnya jika menemukan permasalahan pada produk atau terjadi komplain oleh konsumen, sedangkan *tracing* dilakukan misalnya jika ditemukan permasalahan pada material atau kesalahan pada *suppliernya*. Dalam industri manufaktur, sistem *traceability* merupakan metode untuk melacak dan memeriksa identitas produk, identitas proses, dan identitas material. Untuk sistem *traceability* di lingkup *manufaktur* ini, hal-hal yang perlu didokumentasikan adalah siapa, kapan, dan di mana suatu proses pembuatan produk dilakukan (Gambar 2.1).



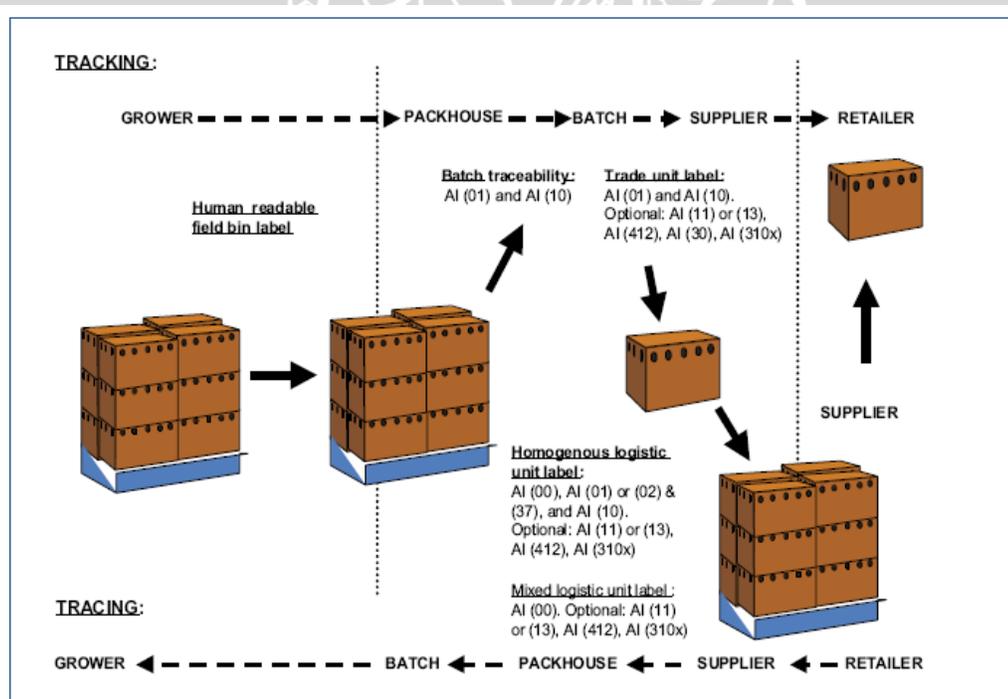
Gambar 2.1 Konsep Dasar *Traceability*

Sumber: Yoji Taniguchi, *IC-Tag Based Traceability*, HITACHI (2005:19)

Menurut *Food Standart Agency*, USA (2002), *traceability* dapat ditinjau dari empat konteks yang berbeda, dan tiap konteks mempunyai model dan aplikasi yang berbeda pula. Keempat konteks itu adalah:

1. Untuk produk, menciptakan suatu mata rantai antara material, proses, produk, dan distribusi dalam suatu jalur *supply chain*.
2. Untuk data, menghubungkan data dan kalkulasi untuk menghasilkan sistem laporan yang sistematis dan informatif.
3. Untuk kalibrasi, menyesuaikan alat ukur yang digunakan dengan standar nasional, internasional atau standar global.
4. Untuk IT dan pemrograman, menghubungkan desain dan proses implementasi program sesuai dengan kebutuhan sistem.

Model *traceability* untuk produk dapat dilihat pada Gambar 2.2. Gambar tersebut menjelaskan tentang aliran fisik barang dan aliran informasi yang menyertainya di sepanjang jalur *supply chain*. Tanda panah menunjukkan adanya transportasi dalam pendistribusian produk mulai dari produsen sampai pada pengecer. Pada setiap tahapan pendistribusian produk tersebut terdapat sistem identifikasi produk yang memiliki format yang berbeda-beda sesuai dengan penggunaannya. Sistem identifikasi ini membantu mempermudah dan memonitor aliran barang dan merupakan syarat utama untuk menjalankan sistem *traceability*.



Gambar 2.2 Model Traceability  
Sumber: *Fresh Produce Traceability Guidelines* (2004:8)

### 2.3.2 Traceability Produk

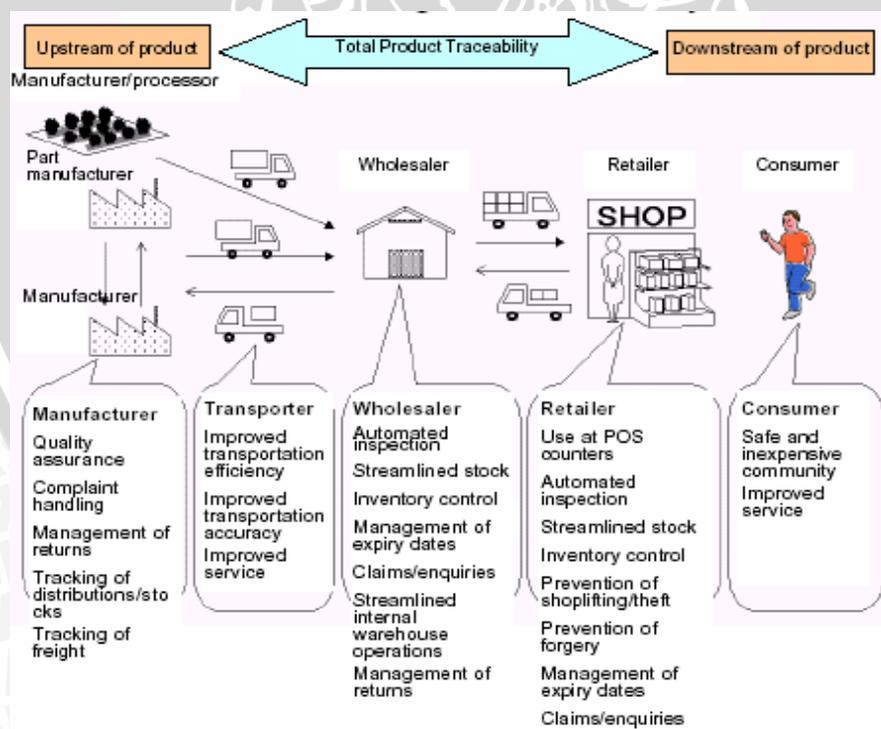
Istilah *traceability* yang berhubungan dengan produk atau pabrikasi telah mulai dikembangkan sejak awal dekade 1990an. Hal itu ditandai dengan adanya sertifikasi standar ISO seri 9000, yang salah satu rekomendasinya adalah tentang identifikasi produk dan *traceability*. Dalam dunia industri saat ini, *traceability* produk merupakan topik yang sedang hangat dibahas, karena *traceability* produk merupakan salah satu upaya perusahaan dalam melindungi bisnis dan merek mereka, yaitu dengan cara mengontrol semua produk yang dihasilkan, baik secara kualitas, jumlah, pendistribusian, dan sebagainya. Di berbagai negara, khususnya di Uni Eropa, implementasi *traceability* produk wajib dilakukan pada setiap industri makanan, obat-obatan, dan kosmetik. Bahkan untuk industri konstruksi, otomotif, elektronik, dan berbagai industri lain juga sudah mulai menerapkan sistem *traceability*.

Dalam tren industri saat ini kepuasan, keselamatan, dan keamanan konsumen merupakan hal yang sangat dikedepankan, hal ini diindikasikan dengan banyaknya regulasi baik nasional, regional, maupun global yang dibuat untuk menangani kasus sengketa konsumen. Banyak kasus sengketa konsumen sering terjadi, misalnya sebagai akibat kualitas produk yang jelek, produk sudah lewat tanggal kadaluarsa, konsumen menerima produk palsu, produk tidak aman atau berbahaya, dan sebagainya. Hal ini selain jelas merugikan konsumen, tentu juga banyak berpengaruh terhadap eksistensi perusahaan. Bagi perusahaan, di samping rugi materi secara langsung akibat membayar tuntutan konsumen, juga mengalami kerugian lain yang lebih besar yaitu jatuhnya nilai jual dari merek yang menjadi andalan, karena hal ini bisa mengancam kelangsungan bisnis perusahaan. Untuk itulah setiap perusahaan khususnya perusahaan *consumer goods* perlu menerapkan sistem *traceability* produk.

Meskipun TQM, Taguchi, Six sigma, SPC, dan program peningkatan mutu yang lain gencar dilakukan, berbagai hal di luar program tersebut dapat saja menyebabkan kesalahan dalam proses pembuatan suatu produk (Peter Green, 2001:1). Misalnya: kerja mesin tidak optimal, operator membuat kekeliruan, penggunaan material cacat, dan sebagainya. Pertanyaan yang muncul bukanlah apakah produk-produk cacat yang akan dikirimkan, tetapi pertanyaan yang riil

adalah bagaimana cara merespon dan menyikapi ketika masalah-masalah tersebut terjadi. Salah satu cara menganalisa dan mengevaluasinya adalah dengan adanya dokumentasi data sebagai alat informasi untuk menelusuri lokasi, waktu, jumlah, dan penyebab kesalahan dalam proses pembuatan suatu produk.

*Traceability* produk adalah kemampuan suatu perusahaan untuk melacak kembali prosedur-prosedur pembuatan produk dan juga kemampuan dalam melacak kembali produk-produk yang dikembalikan ke perusahaan (Quinn Garner, 2001:3). Sistem *traceability* umumnya dijalankan oleh Departemen *Quality Assurance* (QA) di suatu perusahaan. Departemen *Quality Assurance* menjalankan sistem *traceability* untuk memberikan dasar informasi kepada manajemen perusahaan tentang kondisi produk yang akan dipasarkan maupun produk yang sudah ada di pasaran. Penerapan *traceability* ini bisa mengacu pada aturan ISO atau prosedur-prosedur lain seperti *Good Manufacturing Practices* (GMP), *Good Distribution Practices* (GDP), *Good Retailing Practices* dan sebagainya.



Gambar 2.3 Manfaat *Traceability* Produk

Sumber: *Study Group on the Improvement of Product Traceability, Ministry of Economy, Trade and Industry, Japan* (2003:8)

Gambar 2.3 menjelaskan tentang manfaat yang diperoleh dari penerapan *traceability* produk. Manfaat itu bisa dirasakan oleh berbagai pihak baik *manufacturer*, *transporter*, grosir, agen, pengecer bahkan sampai pada konsumen. Khusus bagi perusahaan manufaktur, penerapan sistem *traceability* produk merupakan hal yang sangat penting. Dengan sistem *traceability* ini manajemen perusahaan dapat meningkatkan jaminan terhadap kualitas, khususnya dalam menangani komplain pelanggan (*complaint handling*). Adapun kegunaan utama dari penerapan *traceability* produk bagi suatu perusahaan antara lain:

1. Mematuhi perundang-undangan yang berlaku.
2. Melindungi reputasi merek.
3. Menentukan jumlah dan lokasi saat terjadi penarikan produk dari pasar.
4. Sebagai alat pembanding dengan produk-produk yang tidak standar, dalam hal ini termasuk produk-produk palsu.
5. Untuk mendiagnosa permasalahan pada setiap tahapan proses produksi sesuai dengan peta aliran prosesnya.

Kemampuan melacak proses pembuatan produk berdasarkan data sejarah produk yang telah didokumentasikan akan memberikan kemudahan manajemen perusahaan untuk menentukan kebijakan jika terjadi masalah dengan produk jadi yang masih ada di pabrik apakah perlu di *hold* atau di *reject* dan memberikan kepastian bahwa hanya produk yang bermutu tinggilah yang akan beredar di pasaran. Karena ketika terjadi keluhan pelanggan berkaitan dengan kualitas produk, perusahaan dapat merespon dan menyikapi berdasarkan data-data yang ada. Bila ternyata kualitas produk jauh dari standar mutu yang ditetapkan, perusahaan dapat menarik kembali produk-produk yang ada di pasar secara cepat dan tepat baik waktu, jumlah maupun lokasinya.

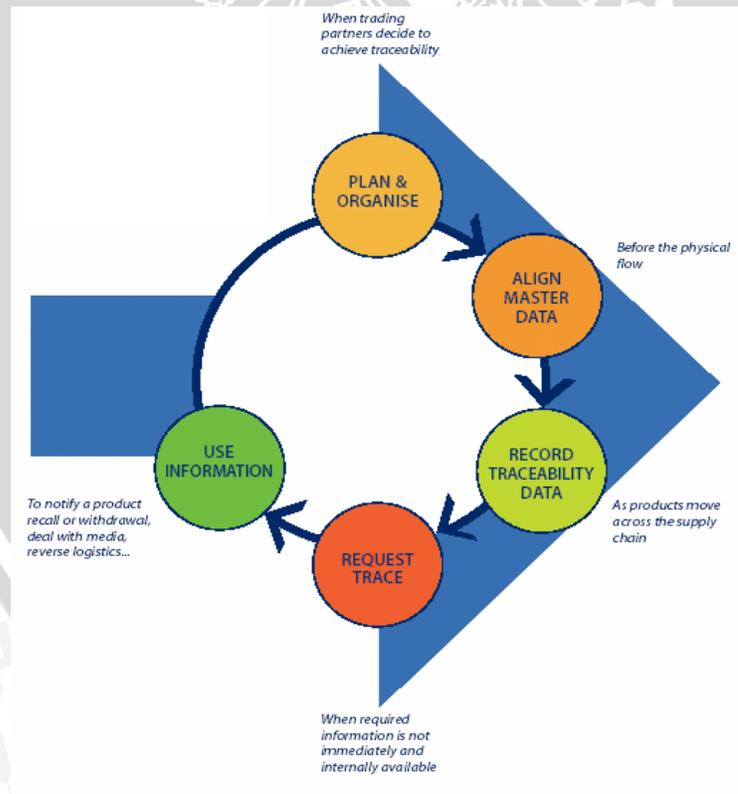
Sistem *traceability* produk digunakan untuk membantu memperbaiki kualitas secara keseluruhan. Jika ditemukan kerusakan atau produk cacat, dengan adanya sistem *traceability* akan memudahkan manajemen perusahaan untuk memprediksi dan menentukan penyebab kerusakan berdasarkan informasi yang sudah direkam. Beberapa contoh informasi itu misalnya: kapan produk dibuat, material apa saja yang dipakai, mesin-mesin dan peralatan apa saja yang digunakan dan sebagainya, kemudian data tersebut dibandingkan dengan

parameter proses yang telah distandarkan perusahaan. Hal tersebut bertujuan untuk mendapatkan jejak balik permasalahan dan mencegah terjadinya kesalahan lagi di lain waktu.

### 2.3.3 Langkah-langkah *Traceability*

Dalam industri manufaktur, penerapan sistem *traceability* harus didukung oleh semua pihak mulai dari manajemen puncak sampai pada operator yang menjalankan mesin di lingkungan produksi. Hal ini juga harus ditunjang dengan sistem komunikasi yang baik antar departemen, khususnya Departemen *Manufacturing, Quality Assurance, Planning and Logistics* dan *Marketing*.

*The Global Traceability Standard* pada situs [www.gsl.org](http://www.gsl.org) menjelaskan tentang aturan bisnis dan kebutuhan minimum yang harus dipenuhi untuk merancang dan menerapkan suatu sistem *traceability*. Hal itu digambarkan dalam suatu bagan yang digunakan sebagai dasar acuan dalam menjalankan peran dan tanggung jawab untuk masing-masing langkah proses *traceability* (Gambar 2.4).



Gambar 2.4 Langkah-langkah *Traceability*  
Sumber: *The Global Traceability Standard* (2006:11)

Gambar 2.4 adalah bagan dari langkah-langkah *traceability*. Langkah-langkah ini merupakan acuan atau syarat bagi Departemen *Quality Assurance* suatu perusahaan untuk menjalankan sistem *traceability*, yaitu menyesuaikan tugas-tugas yang harus dilaksanakan dengan mempertimbangkan jumlah dan kemampuan sumber daya manusia yang dimiliki. Secara rinci penjelasan dari bagian-bagian pada Gambar 2.4 adalah sebagai berikut:

**1. *The plan and organise***

Sub proses ini menentukan bagaimana cara mengumpulkan, menyimpan serta memelihara data *traceability*. Sub proses ini nantinya akan menentukan hubungan antara *input*, *internal process*, dan *output*.

**2. *The align master data***

Sub proses ini menentukan bagaimana cara mengidentifikasi produk dan material berdasarkan aspek-aspek yang relevan. Sistem identifikasi dalam suatu unit, *batch*, maupun *logistic* harus memiliki keunikan sehingga tidak menimbulkan keambiguan.

**3. *The record traceability data***

Sub proses ini menentukan bagaimana cara merekam identitas, mengumpulkan dan mengklasifikasikan data. Data harus dirawat dan dipelihara sebagai sumber informasi.

**4. *The request trace***

Sub proses ini menentukan bagaimana bereaksi terhadap permintaan kebutuhan *traceability*, misalnya ditemukannya *non conforming product* di *batch* "xxx" maka sub proses ini akan merespon dan menentukan langkah-langkah selanjutnya untuk menangani permasalahan tersebut.

**5. *The use information***

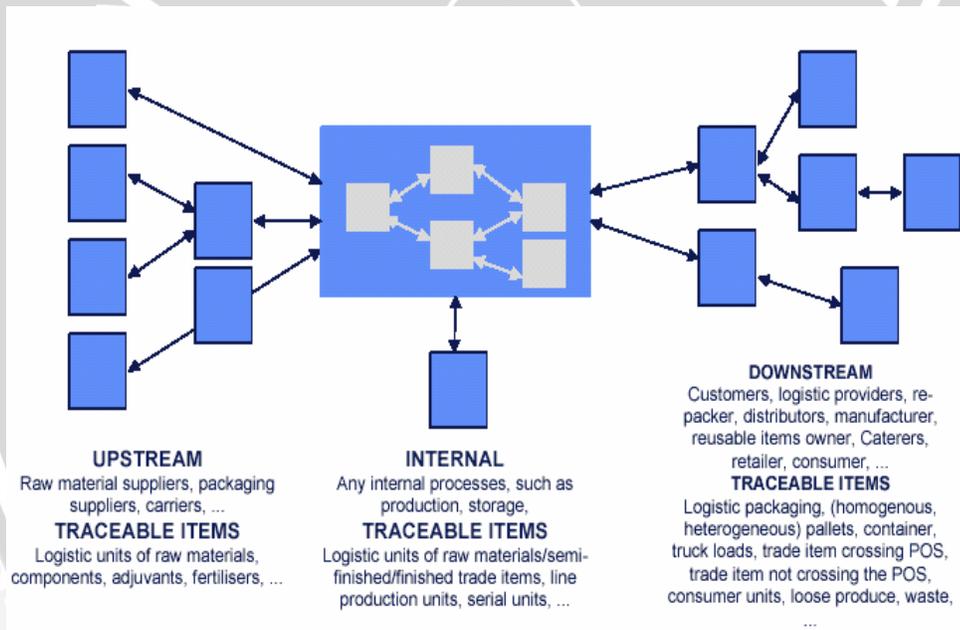
Sub proses ini bertugas menghasilkan informasi yang digunakan untuk pengambilan keputusan. Pada langkah inilah diperlukan suatu sistem *database* yang mengintegrasikan data dan menghasilkan informasi, dengan informasi tersebut manajemen perusahaan dapat mengambil keputusan secara cepat dan akurat. Pengambilan keputusan ini dilakukan berdasarkan sub proses sebelumnya serta aturan-aturan bisnis yang ada di organisasi atau perusahaan tersebut.

### 2.3.4 Faktor-faktor *Traceability*

Menurut EAN-UCC *Traceability Implementation* (2003:9), terdapat empat faktor utama yang harus dipenuhi untuk penerapan sistem *traceability*. Keempat faktor itu antara lain:

#### 1. Identifikasi

Dalam suatu manufaktur, sistem *traceability* melibatkan identifikasi dari semua aspek yang relevan, khususnya identifikasi pada material dan produk. Identifikasi dalam suatu unit material dan produk harus memiliki keunikan, karena saat diperlukan proses pelacakan akan membutuhkan data secara pasti berdasarkan identitas-identitas unik yang telah direkam. Maka dalam internal manufaktur perlu melakukan sistem identifikasi pada bahan baku, bahan setengah jadi, maupun pada produk jadi (Gambar 2.5).

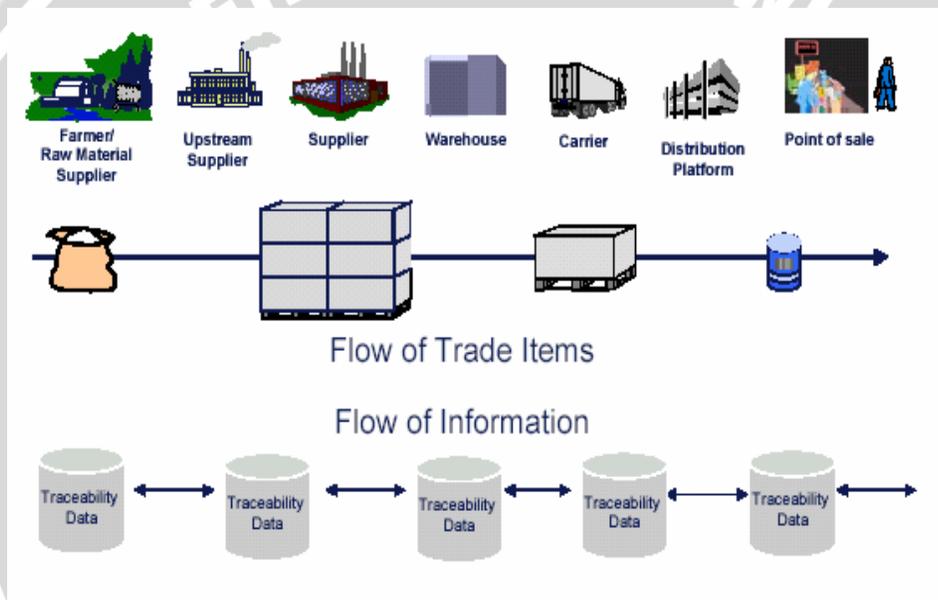


Gambar 2.5 Identifikasi dalam Sistem *Traceability*  
 Sumber: EAN-UCC *Traceability Implementation* (2003:10)

#### 2. Pendokumentasian data

Peranan sistem informasi sangat penting dalam penerapan sistem *traceability*, yaitu dengan adanya data yang sudah direkam dan didokumentasikan sebelumnya. Dalam pelaksanaan sistem *traceability* di lingkungan manufaktur, data tersebut meliputi data material atau komponen-komponen produk, proses perubahan bentuk pada lini produksi,

waktu pembuatan produk, kode produk dan lain-lain. Informasi ini dihubungkan berdasarkan identifikasi kode material, nama material, *supplier*, nomor mesin, kode produk, nomor *batch*, nama produk, tanggal pembuatan atau informasi lain yang mendukung. Dengan penggunaan sistem *database*, informasi yang ditangkap dapat direkam dan didokumentasikan secara sistematis. Pada sistem *traceability* yang kompleks, pendokumentasian data ini akan menghasilkan laporan-laporan sebagai sumber informasi dan mempunyai peranan penting sebagai penghubung antara pihak-pihak yang terlibat di jalur *Supply Chain* (Gambar 2.6).



Gambar 2.6 Aliran Informasi pada *Traceability Systems*  
Sumber: EAN-UCC *Traceability Implementation* (2003:11)

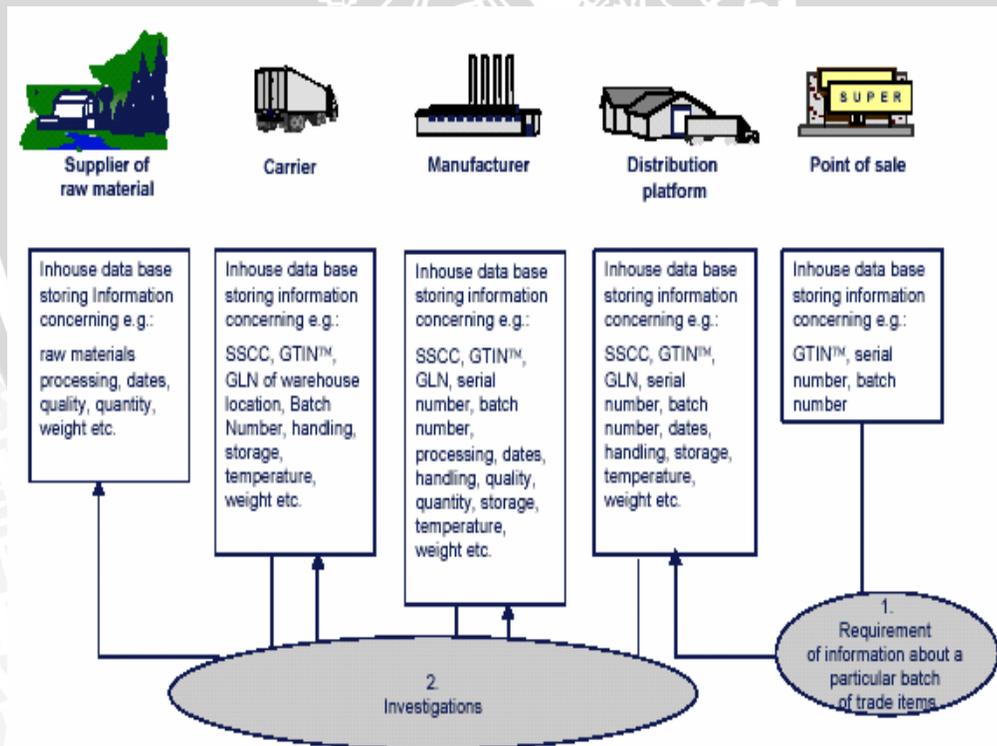
### 3. *Links Management*

Sistem *traceability* mewajibkan pengaturan mata rantai yang berurutan dalam memproduksi unit produk maupun *batch* pada keseluruhan proses. Di dalam manajemen perusahaan, sistem *traceability* merupakan kendali dari semua mata rantai, yang memungkinkan untuk membuat koneksi antara material apa yang telah diterima, apa yang diproduksi, dan produk apa yang telah dikirimkan atau sebaliknya. Jika salah satu bagian dari

rantai ini tidak berjalan maka bagian lain akan terkena dampaknya dan sering disebut dengan istilah "hilangnya *traceability*".

**4. Communication**

Sistem *traceability* melibatkan asosiasi aliran informasi tentang arus barang. Untuk memastikan mengalirnya informasi, masing-masing bagian perlu menyampaikan identitas kepada mitra berikutnya di dalam suatu jalur mata rantai. Mata rantai aliran informasi dan arus fisik barang harus sesuai dengan sistem identifikasi yang diterapkan. Jadi komunikasi *traceability* harus menyediakan penjelasan dan definisi secara jelas dan transparan untuk memberikan informasi secara sederhana, akurat, dan tepat. Penggunaan suatu bahasa umum dengan identifikasi yang unik dan mudah dipahami sangat penting bagi arus informasi yang menghubungkan aliran material dan produk. Karena dengan adanya alat atau bahasa komunikasi sepanjang jalur *supply chain* akan mempermudah investigasi terhadap suatu proses maupun barang (Gambar 2.7).

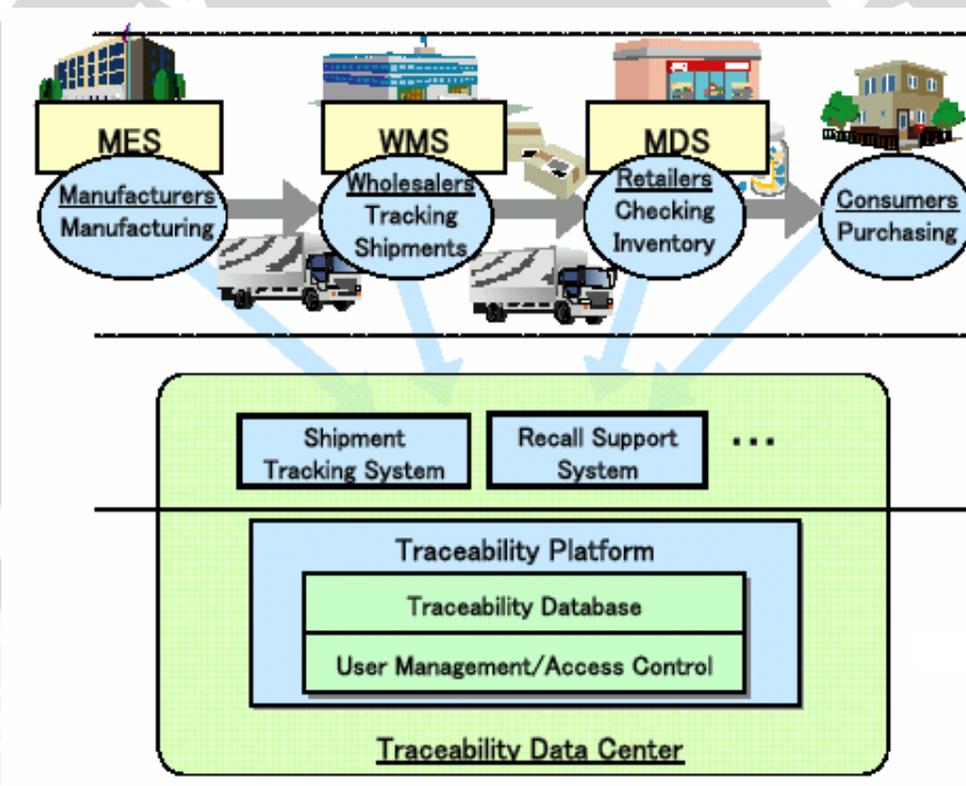


Gambar 2.7 Kebutuhan Sistem Identifikasi untuk Komunikasi  
 Sumber: EAN-UCC *Traceability Implementation* (2003:19)

### 2.3.5 Traceability Database

*Database* merupakan salah satu alat yang sesuai untuk membangun sistem *traceability* yang optimal. Dengan adanya sistem *database* akan meminimalisir kesalahan dalam *input* maupun *output* sistem. *Traceability database* mempunyai karakteristik sebagai berikut:

1. Data atau informasi aman dan mudah diperoleh.
2. Ketika terjadi permasalahan, konsep *tracking* dan *tracing* akan mudah untuk dilakukan.
3. Manajemen perusahaan dapat mengevaluasi kualitas material, keadaan produk, dan seluruh kelangsungan proses produksi dengan seksama.
4. Mengurangi biaya dalam sistem administrasi perusahaan.



Gambar 2.8 Traceability Database

Sumber: Yoji Taniguchi, *IC-Tag Based Traceability*, HITACHI (2005:21)

Gambar 2.8 merupakan implementasi *database* untuk sistem *traceability* yang kompleks. Suatu perusahaan yang menjalankan sistem tersebut perlu pendokumentasian data yang terintegrasi dengan sistematis, mulai data dari

manufaktur, grosir, pengecer, bahkan sampai pada konsumen. Data tersebut berfungsi sebagai sumber informasi dalam melakukan pelacakan saat terjadi permasalahan pada produk atau saat ada produk-produk yang terpaksa ditarik dari pasar. Dengan menggunakan *database*, manajemen perusahaan akan mendapatkan data-data yang akurat dan *up to date* untuk merespon serta menyikapi permasalahan-permasalahan yang ada.

### 2.3.6 Traceability pada Proses Produksi

Implementasi *traceability* pada proses produksi atau disebut *internal traceability*, merupakan kegiatan pendokumentasian data yang mengintegrasikan data material, data proses permesinan, dan data produk. Dengan adanya data yang terintegrasi tersebut, maka akan mempermudah manajemen perusahaan dalam mengidentifikasi akar permasalahan, khususnya berkaitan dengan produk-produk yang tidak memenuhi standar, apakah kesalahan berasal dari kinerja mesin, bahan baku yang dipakai, kelalaian operator, maupun yang lainnya.

Konsep *traceability* pada suatu proses produksi, secara umum dapat dijelaskan pada Gambar 2.9. Pada gambar tersebut mengindikasikan bahwa setiap material yang akan diproses di mesin maupun setiap produk yang dihasilkan oleh mesin harus mempunyai nomor identifikasi atau yang sering disebut dengan *barcode*.



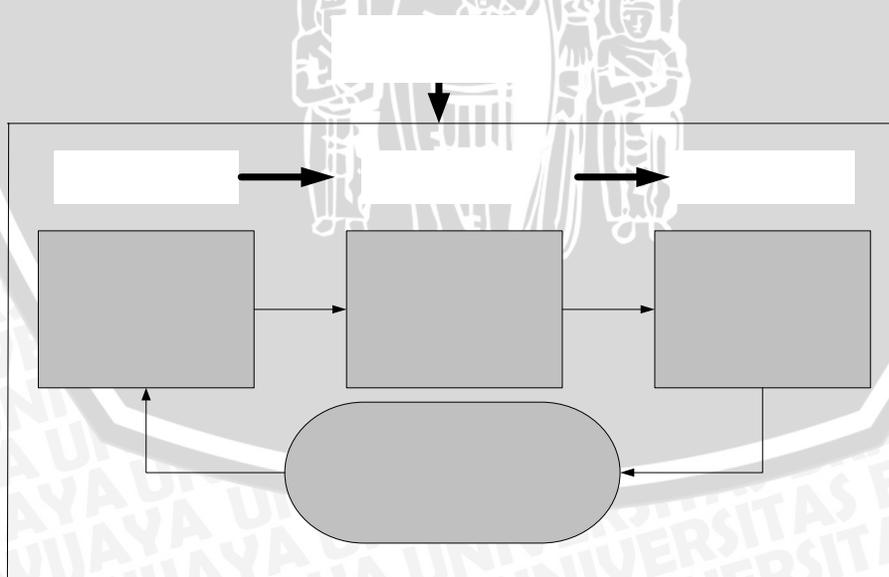
Gambar 2.9 Konsep *Traceability* pada Proses Produksi  
Sumber: *Shipment Traceability System*, 2006

## 2.4 Sistem Produksi

### 2.4.1 Konsep Dasar Sistem Produksi

Produksi merupakan bagian pokok dari organisasi pabrik, yang bertanggung jawab dalam mentransformasi nilai tambah dari *input* menjadi *output*. Sedangkan sistem produksi merupakan sistem integral yang mempunyai komponen struktural dan fungsional (Gaspersz, 2004:4). Komponen struktural yang membentuk sistem produksi terdiri dari: material, mesin, tenaga kerja, modal, energi, informasi, tanah, dan lain-lain. Sedangkan komponen fungsional terdiri dari: supervisi, perencanaan, koordinasi, dan kepemimpinan, yang kesemuanya berkaitan dengan suatu manajemen. Suatu sistem produksi selalu berada dalam lingkungan, sehingga aspek-aspek lingkungan, seperti: perkembangan teknologi, sosial dan ekonomi, serta kebijakan pemerintah akan selalu mempengaruhi keberadaan sistem produksi tersebut. Secara lengkap skema sistem produksi ini digambarkan pada Gambar 2.10.

Dari Gambar 2.10 tampak bahwa elemen-elemen utama dalam sistem produksi adalah *input*, proses, dan *output*, serta adanya suatu mekanisme umpan balik untuk mengendalikan sistem produksi tersebut. Dalam hal ini, proses produksi merupakan ujung tombak dalam mentransformasi nilai tambah dari *input* menjadi *output*.



Gambar 2.10 Skema Sistem Produksi  
Sumber: Gaspersz (2004:5)

#### 2.4.2 Klasifikasi Proses Produksi

Suatu industri manufaktur dapat memilih proses transformasi nilai tambah dari *input* menjadi *output* sesuai dengan jumlah kebutuhan pasar. Berdasarkan jumlah atau volume produksi yang dihasilkan, industri manufaktur dapat dikelompokkan menjadi tiga tipe (Sritomo W, 1993:53), yaitu:

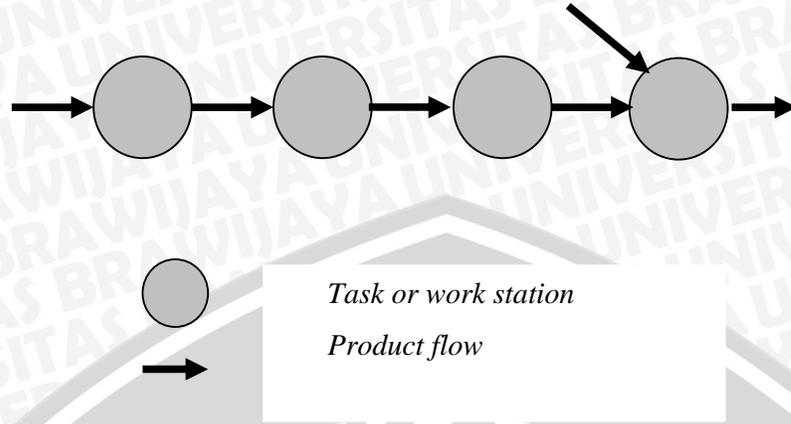
1. **Job shops production**, tipe industri ini sering pula disebut industri yang bekerja berdasarkan pesanan (*job order*). Volume yang dihasilkan seringkali rendah dan umumnya untuk memenuhi pesanan yang spesifik.
2. **Batch production**, industri kategori ini akan membuat produk dalam jumlah atau volume dengan skala *medium size*. Produk yang dihasilkan bisa dibuat hanya sekali atau bisa juga diproduksi pada interval waktu tertentu. Tujuan utama dari *batch production* adalah untuk memuaskan konsumen dalam menyediakan produk secara kontinyu.
3. **Mass Production**, tipe industri ini menghasilkan produk dalam jumlah besar tetapi relatif sejenis. Dalam proses produksinya menggunakan mesin dan peralatan yang bertipe *special purpose*.

#### 2.4.3 Karakteristik Aliran Produk

Menurut Schroeder (2000:55), aliran produk dalam industri manufaktur diklasifikasikan menjadi tiga tipe, yaitu: *line flow*, *batch flow*, dan *project flow*. Pada prakteknya aliran produk ini sama dengan aliran material, karena material yang dipakai akan dikonversikan menjadi produk melalui sebuah proses produksi. Berikut ini penjelasan dari masing-masing tipe aliran produk.

##### 1. **Line Flow**

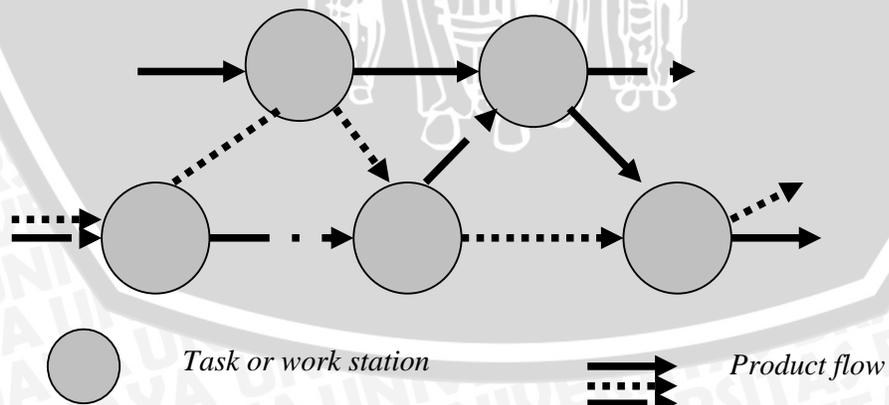
*Line flow process* menyusun stasiun-stasiun kerja (*work stations*) dalam sekuens operasi yang membuat produk, sehingga terkadang disebut sebagai *product flow*, karena produk mengalir mengikuti langkah-langkah sekuensial dalam proses produksi. Semua produk membutuhkan tugas-tugas yang sama, dan mengikuti pola aliran standar (*standard flow pattern*) seperti ditunjukkan pada Gambar 2.11. Salah satu contoh penerapan dari *line flow process* adalah *assembly line* dari suatu industri otomotif.



Gambar 2.11 Line Flow  
 Sumber: Schroeder, *Operations Management* (2000:55)

**2. Batch Flow**

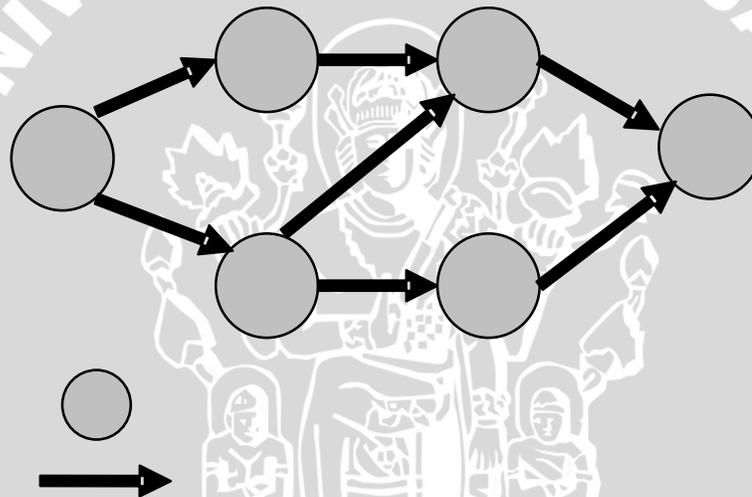
*Batch flow* merupakan karakteristik produksi yang produknya berada dalam *batches* atau *lots*. Setiap *batch* aliran produk dari satu operasi (*work center*) ke operasi (*work center*) lainnya merupakan suatu kumpulan dari mesin-mesin yang serupa, yang digunakan untuk membuat suatu produk. Gambar 2.12 menunjukkan sebuah *batch process* yang terdiri dari beberapa *work center* dan beberapa produk. Setiap produk mempunyai arah aliran yang berbeda dalam melewati *work center-work center* dari awal sampai akhir proses.



Gambar 2.12 Batch Flow  
 Sumber: Schroeder, *Operations Management* (2000:56)

### 3. *Project Flow (No Product Flow)*

Dalam suatu proyek tidak ada aliran produk, tetapi bagaimanapun juga suatu proyek tetap memiliki urutan-urutan atau sekuens operasi (Gambar 2.13). Bentuk proyek digunakan apabila terdapat suatu kebutuhan khusus untuk kreatifitas dan keunikan, serta memiliki batas waktu penyelesaiannya. Adalah sulit untuk mentransformasikan proses manufakturing pada proyek, karena pada umumnya proyek hanya dikerjakan sekali saja. Proyek cenderung memiliki biaya tinggi serta sulit untuk merencanakan dan mengendalikannya, sebab memiliki tingkat kesulitan dalam pendefinisian awal karena dapat saja membutuhkan tingkat perubahan dan inovasi yang tinggi.



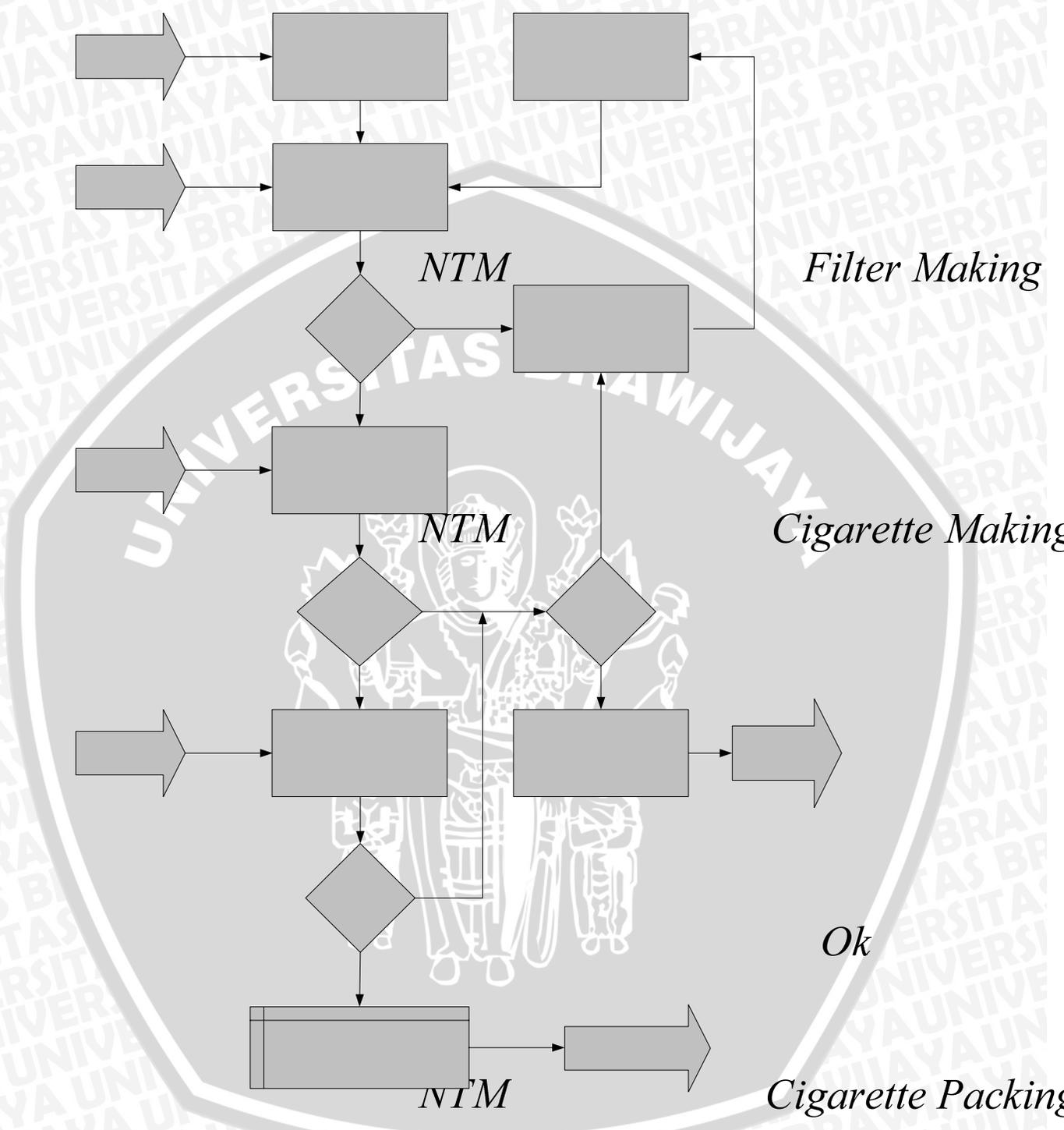
Gambar 2.13 *Project Flow*

Sumber: Schroeder, *Operations Management* (2000:57)

#### 2.4.4 Diagram Alir Proses

Salah satu cara yang umum dipakai untuk menggambarkan proses dari suatu sistem produksi adalah diagram alir proses (*process flow diagram*). Menurut Suryadi Prawirosentono (2001:63) diagram alir proses adalah diagram tentang urutan proses pengolahan suatu produk yang dibuat mulai dari bahan baku, barang setengah jadi, sampai dengan barang jadi. Tiap perusahaan mempunyai diagram alir proses yang berbeda-beda tergantung dari skala usaha dan jenis produk yang dihasilkan. Berikut ini adalah diagram alir proses produksi rokok di PT. Philip Morris Indonesia.

START



Gambar 2.14 Diagram Alir Proses Produksi PT. Philip Morris Indonesia.

Penjelasan masing-masing bagian dari gambar di atas adalah sebagai berikut:

- NTM (*Non Tobacco Material*), adalah semua material yang digunakan di dalam proses produksi selain tembakau.

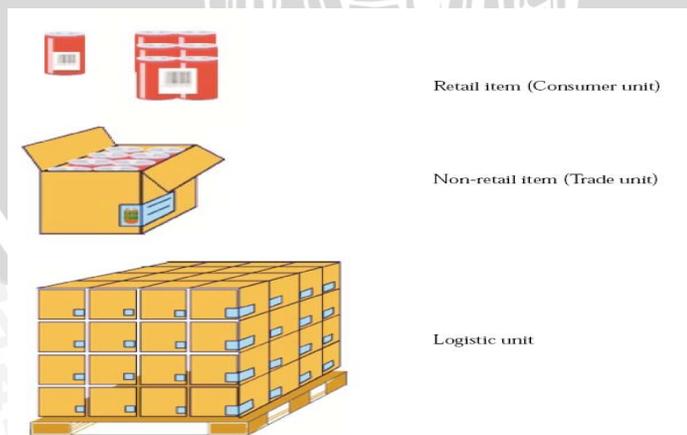
- *Cut filler*, adalah tembakau yang sudah mengalami proses pengolahan, perajangan, dan penambahan bahan-bahan lain, sehingga siap diproses untuk dijadikan rokok.
- *Filter making*, adalah proses pembuatan filter dengan menggunakan mesin filter. Bahan dasar utama dalam proses ini adalah tow dan pizet. Hasil dari proses ini adalah bahan setengah jadi, selanjutnya diproses dalam proses *cigarette making*.
- *Cigarette making*, adalah proses pembuatan batang rokok dengan menggunakan mesin maker. Bahan dasar pada proses ini adalah filter yang telah diproses pada *filter making*, *cut filler* yang disalurkan oleh *feeder machine* dan bahan-bahan lain yang siap langsung diproses seperti *Cigarette Tipping Paper (CTP)*, *Cigarette Paper (CP)*, *Adhesive*, dan *Ink*.
- *Cut Filler Feeding*, adalah suatu proses pendistribusian *cut filler* yang dimasukkan ke dalam mesin feeder dan kemudian didistribusikan ke masing-masing mesin *cigarette* maker melalui pipa-pipa secara otomatis.
- *Cigarette packing*, adalah proses pengepakan batang rokok dengan menggunakan mesin packer. Di dalam proses ini batang rokok diproses dalam kemasan *pack*, kemudian *pack* dikemas dalam *slof* dan akhirnya dikemas dalam *shipping case* yang berisi 10.000 batang rokok.
- *Shipping*, adalah proses pengemasan, pemberian identitas produk, penyimpanan dan produk siap untuk dipasarkan.
- *Ripping*, adalah proses pengolahan kembali batangan rokok yang di *reject* pada saat proses produksi, yaitu memisahkan tembakau dengan bahan-bahan yang lain. Kemudian tembakaunya diproses kembali ke dalam *cut filler feeding*.
- *Shreding* adalah proses penghancuran NTM yang bertuliskan merek produk.
- *Waste*, adalah proses pemilahan dan pembuangan sampah yang selanjutnya dikemas untuk dihancurkan atau dibakar.
- *Warehouse*, adalah tempat penyimpanan produk sebelum dikirim ke pasar.
- *Marketing*, adalah pemasaran dan pendistribusian produk sampai pada konsumen.

## 2.5 Sistem Identifikasi Produk

Pemakaian teknologi untuk meningkatkan manajemen *supply chain* adalah dengan menggunakan standarisasi yang diakui secara global, sehingga memungkinkan perusahaan melakukan bisnis dengan berbagai macam mitra pada lingkungan bisnis yang terbuka. Dalam hal ini yang paling penting adalah penggunaan nomor identifikasi atau kode untuk memantau produk yang melewati jalur *supply chain*. Sistem identifikasi yang telah distandarkan sesuai dengan klasifikasinya, dapat meningkatkan ketepatan dan efisiensi dalam pergerakan fisik produk yang melalui jalur *supply chain*.

### 2.5.1 Klasifikasi Produk

Pada prinsipnya manufaktur, distributor, retailer, sampai dengan konsumen menginginkan untuk dapat mengetahui riwayat suatu produk dari sudut pandang yang berbeda-beda. Untuk menjembatani kepentingan tersebut maka diperlukan pengklasifikasian produk untuk mempermudah transaksi dan pertukaran informasi diantara pelaku bisnis. Dengan demikian para pelaku bisnis diharapkan akan mempunyai persepsi yang sama terhadap produk yang ditransaksikan. Pengklasifikasian produk yang umum dipakai dalam perdagangan global antara lain: *consumer unit*, *trade unit*, dan *logistics unit*. Di Indonesia pengaturan sistem identifikasi produk berdasarkan klasifikasi produk tersebut dilakukan oleh Global Standar Satu Indonesia (GS1 Indonesia). Untuk lebih jelasnya pengklasifikasian produk ini dapat dilihat pada Gambar 2.15.



Gambar 2.15 Pengklasifikasian Produk

Sumber: *Fresh Produce Traceability Guidelines* (2004:10)

### 2.5.2 GS1 Indonesia

GS1 Indonesia (baca: ji es wan) adalah suatu badan yang mengelola sistem identifikasi dan komunikasi untuk produk barang dan jasa, prasarana, satuan transport, dan lokasi (Sri Hartati, 2004). GS1 Indonesia mengembangkan serta menjaga standar kode bagi semua pemakai, bertujuan untuk kepentingan global dan standar multi-sektoral dengan maksud guna menyediakan persamaan bahasa bagi kepentingan penyelenggaraan perdagangan internasional. Sepanjang tahun 2005, sekitar 320 perusahaan di Indonesia mendaftarkan nomor identifikasi kode baru dari sekitar 1.000 *item* produk pada GS1 Indonesia.

Saat ini GS1 Indonesia sedang mengembangkan penggunaan identifikasi elektronik menggunakan teknologi *Radio Frequency Identification* (RFID) dan *electronic Product Code* (ePC) di pita frekuensi UHF 860-960 MHz. Dengan demikian, saat konsumen yang berbelanja hendak membayar di kasir tidak ada penghitungan produk belanja satu per satu. Teknologi informasi di dunia perdagangan itu sudah mulai digunakan secara global khususnya dalam jalur *supply chain* agar sistem *traceability* lebih efisien dan juga mengurangi faktor kesalahan oleh manusia. Label atau *chips* ePC yang ditempel pada suatu produk akan membuat produk tersebut menjadi pintar (*smart product*). Di saat kritikal, pabrikan dapat menarik kembali produknya yang bermasalah dalam waktu yang relatif singkat.

GS1 Indonesia merupakan badan yang dibentuk oleh dua organisasi sistem standar internasional yaitu: *European Article Numbering International* (EAN International) yang berpusat di Brussel, Belgia dan *Uniform Code Council* (UCC) yang berpusat di Ohio, Amerika Serikat. GS1 Indonesia menggunakan sistem EAN-UCC sebagai standarisasi penomoran global yang telah digunakan oleh lebih dari 1 juta organisasi di 101 negara. Sejak 2003, organisasi GS1 telah dipilih oleh MIT Auto-ID Center untuk menstandarisasi dan mengkomersilkan *electronic Product Code* (ePC), sebuah terobosan baru untuk teknologi *supply chain*. Menggunakan internet dan *radio frequency identification* (RFID), ePC secara strategis diseimbangkan untuk membantu perusahaan-perusahaan dalam meningkatkan pelaksanaan *supply chain*, meningkatkan kemampuan identifikasi produk, dan menyediakan kenyamanan dan keamanan konsumen.

### 2.5.3 Sistem EAN·UCC

Sistem EAN·UCC adalah sebuah sistem identifikasi dan komunikasi untuk produk, jasa, unit transport, dan lokasi yang diterima secara global di seluruh dunia (Sri Hartati, 2004). Sistem EAN·UCC mencakup hal-hal sebagai berikut :

- Standar global untuk identifikasi produk, transport, lokasi, dan lain-lain.
- Standar global untuk informasi tambahan seperti *batch number*, tanggal produksi, ukuran, dan lain-lain.
- Metode pengambilan data dengan menggunakan mesin pembaca (*scanner*) untuk menginput data secara cepat dan akurat ke dalam sistem komputer.
- Pesan standar yang digunakan untuk EDI dan e-commerce.

Sistem penomoran EAN·UCC untuk *trade item* merupakan elemen kunci sebagai dasar untuk meningkatkan komunikasi diantara organisasi bisnis dan juga untuk pengawasan yang lebih baik terhadap pengiriman barang. Penjabaran mengenai sebuah produk dengan menggunakan bahasa telah digantikan oleh sebuah nomor unik untuk mengidentifikasi produk. Sistem EAN·UCC mengatasi benturan yang ditimbulkan oleh adanya kode internal dengan menggunakan kemampuan *database* komputer untuk mengambil seluruh informasi yang dibutuhkan mengenai produk tersebut dan mengakses informasi tersebut melalui sebuah nomor unik.

### 2.5.4 Identifikasi Produk Secara Global

Sebuah standar sistem identifikasi bukan hanya memberikan fasilitas yang lebih baik untuk pertukaran informasi di antara pembeli dan penjual, tetapi juga memberikan jangkauan yang lebih luas bagi pertukaran informasi yang melewati seluruh jalur *supply chain*, khususnya dalam mendukung sistem *traceability*. Tanpa sebuah standarisasi, perusahaan terpaksa harus memperhatikan tabel-tabel konversi yang akan menyebabkan ketidak-efisiensi, kesalahan mengartikan, dan banyaknya biaya yang dikeluarkan untuk pengadministrasian.

Untuk mengatasi gesekan yang ditimbulkan oleh penggunaan kode internal (*in-house coding*) digunakan kemampuan *database* komputer untuk menampung semua informasi yang dibutuhkan mengenai produk dan mengakses informasi tersebut melalui sebuah nomor unik. Nomor tersebut tidak akan

menimbulkan kerancuan selama nomor produk yang digunakan bersifat unik (tidak ambigu) dan global.

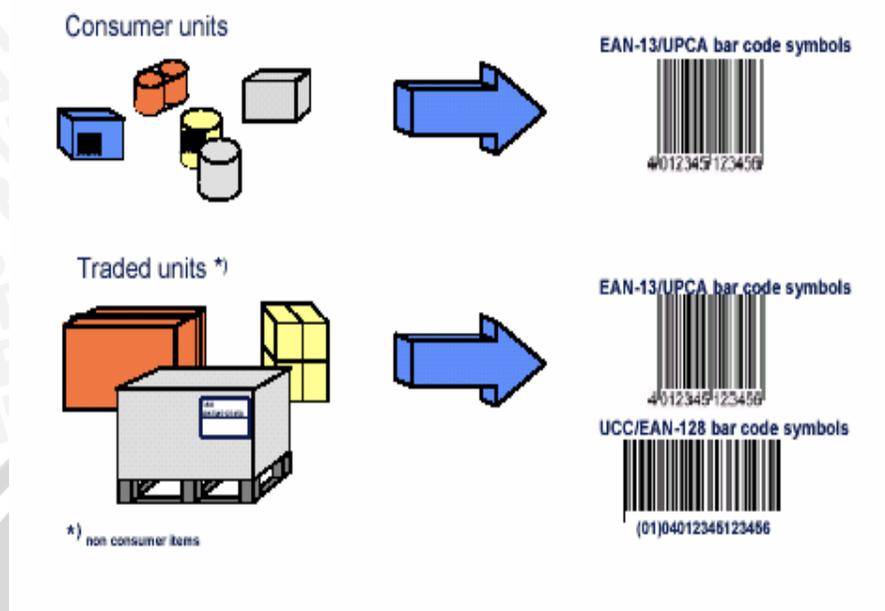
Fungsi sistem identifikasi produk bagi perusahaan manufaktur, pendistribusi produk, dan *retailer* antara lain sebagai berikut:

- Mengidentifikasi barang atau produk secara tepat.
- Mendukung penerapan sistem *traceability*.
- Meningkatkan kecepatan pengiriman barang.
- Mengurangi kesalahan dalam penanganan barang.
- Pengaturan barang persediaan lebih mudah dan akurat.
- Mempermudah inventori pada setiap tingkatan.
- Mengurangi order secara keseluruhan dan mempersingkat waktu perputaran barang.
- Membantu perencanaan yang lebih baik untuk jadwal produksi dalam rangka memenuhi permintaan pasar.
- Memberikan servis yang lebih baik kepada *customer*.

Dengan menggunakan penomoran sistem EAN·UCC, terbentuklah suatu data yang diwakili oleh garis-garis hitam putih atau dikenal sebagai *barcode*. *Barcode* bertindak sebagai pembawa data yang dapat dibaca dengan alat *scanner* untuk mendapatkan informasi detail tentang suatu produk. Informasi ini sangat dibutuhkan dalam mempermudah transaksi di jalur perdagangan.

### **2.5.5 Barcode**

*Barcode* pada dasarnya adalah susunan garis vertikal hitam dan putih dengan ketebalan yang berbeda, walaupun sangat sederhana tetapi sangat berguna, karena berfungsi untuk menyimpan data produk dengan mudah dan murah. Walaupun teknologi semacam itu terus berkembang dengan ditemukannya media magnetik, RFID, *electronics tags*, serial eeprom (seperti pada *smart card*), *barcode* terus bertahan dan masih memiliki kelebihan-kelebihan tertentu yaitu yang paling utama, murah, dan mudah. Sebab media yang digunakan adalah kertas dan tinta, sedangkan untuk alat pembacanya (*scanner*) banyak tersedia di pasaran dengan harga yang relatif murah dan memiliki berbagai tipe sesuai dengan kebutuhan.



Gambar 2.16 Contoh Aplikasi *Barcode* pada Identifikasi Produk  
 Sumber: EAN·UCC *Traceability Implementation* (2003:23)

Gambar 2.16 di atas adalah contoh penggolongan aplikasi *barcode* pada satuan kemasan, yang dikelompokkan sebagai berikut:

### 1. Satuan produk (*Single Item*)

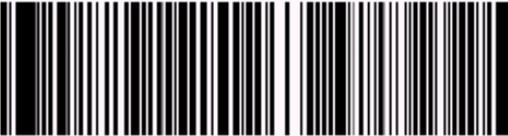
Adalah unit terkecil yang diperdagangkan melalui *Point of Sale* (POS). Setiap jenis produk memiliki *Global Trade Identify Number* (GTIN) yang berbeda. Untuk wilayah Indonesia dan di luar Amerika, GTIN ini berdasarkan simbologi *barcode* EAN-8 atau EAN-13. Sedangkan untuk wilayah Amerika Serikat, berdasarkan simbologi *barcode* UPC-A atau UPC-E.

### 2. Dus (*Grouping of Item*)

Adalah satuan unit yang terdiri dari beberapa atau kumpulan dari produk satuan. Apabila unit tersebut diperdagangkan melalui POS, maka dianggap sebagai produk satuan (*single item*). Namun apabila tidak, maka produk tersebut adalah unit logistik atau disebut sebagai *despacht unit*. Untuk unit yang melalui POS, menggunakan simbologi *barcode* EAN-13. Sistem identifikasi untuk unit logistik (*despacht unit*) diatur dalam *Global Location Number* (GLN), maka dapat menggunakan simbologi EAN-13, ITF-14 ataupun EAN·UCC 128. Tergantung dari kebutuhannya.

### 3. Pallet

Adalah satuan unit besar yang terdiri dari beberapa atau kumpulan dari unit logistik, yang merupakan satuan unit dalam pengiriman. Unit ini menggunakan label logistik (Gambar 2.17) yang memuat informasi mengenai produk yang ada di dalam kemasannya, dengan menggunakan simbologi EAN-UCC 128 yang biasa disebut sebagai *Serial Shipping Container Code* (SSCC). SSCC adalah identifikasi satuan unit logistik yang digunakan dalam pengiriman barang (*pallet*, *container*, dan lain lain). SSCC bermanfaat untuk *traceability* barang yang berada dalam pengiriman, sehingga dengan mudah dapat diketahui keberadaan dan status dari barang yang dikirim tersebut. Data yang terkandung dalam SSCC dapat menunjukkan berbagai macam informasi seperti: produk yang ada di dalamnya, tanggal pembuatan, nomor *batch*, dan sebagainya. Informasi tersebut diidentifikasi dengan kode awalan *Application Identifier* (AI).

| UCC/EAN LOGISTICS LABEL                                                              |                                                                                    |
|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| From<br>EAN International<br>rue Foyale 145<br>B-1000 Brussels                       | To<br>UNIFORM CODE COUNCIL<br>8136 Old Yankee Road<br>Dayton, Ohio 45429<br>U.S.A. |
| <b>SSCC</b><br><b>3 5412345 123456789 2</b>                                          |                                                                                    |
| SERIAL NUMBER<br>10321ABC3189                                                        | CTIN<br>5412345001232                                                              |
|  |                                                                                    |
| (01)05412345001232(21)10321ABC3189                                                   |                                                                                    |
|  |                                                                                    |
| (00)354123451234567892                                                               |                                                                                    |

Gambar 2.17 Contoh Label  
Sumber: EAN-UCC Traceability Implementation (2003:100)

## 2. 6 Database

### 2.6.1 Sejarah Database

Pada tahun-tahun awal penggunaan komputer, para pemakai terkendala dengan cara pencatatan data pada medium penyimpanan. Pada awal tahun 1960, Charles Bachman dari perusahaan General Electric mendesain generasi pertama DBMS yang disebut Penyimpanan Data Terintegrasi (*Integrated Data Store*).

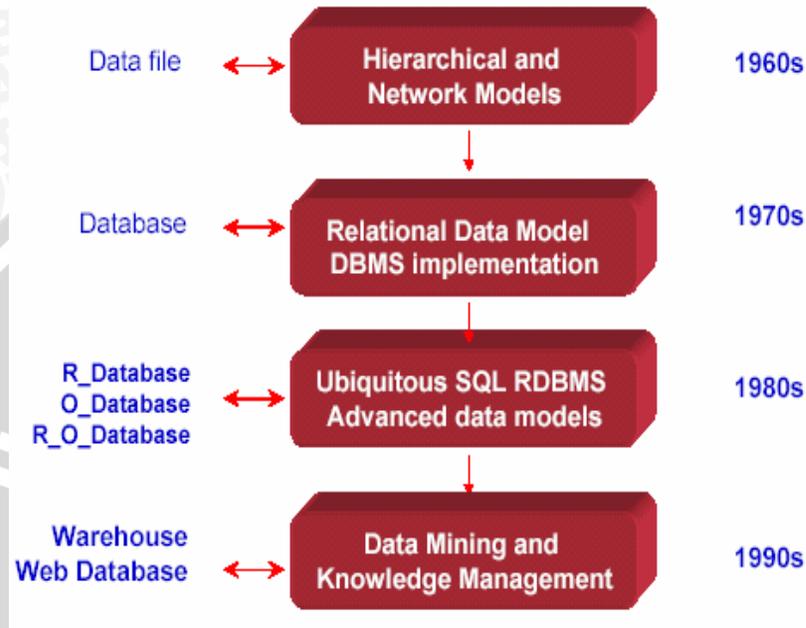
Pada akhir tahun 1960-an, IBM mengembangkan Sistem Manajemen Informasi (*Information Management Systems*) atau disingkat IMS. IMS dibentuk dari representasi data pada kerangka kerja yang disebut model data hierarki.

Pada tahun 1970, Dr. Edgar Codd di laboratorium penelitian di San Jose mengusulkan suatu representasi data baru yang disebut model data relasional. Model ini menyajikan data dalam bentuk tabel yang terdiri atas baris dan kolom. Pada model *database* relasional ini, hubungan antar *file* direlasikan dengan kunci relasi (*relation key*), yang merupakan kunci utama dari masing-masing *file*.

Pada tahun 1980, model *database* relasional menjadi paradigma DBMS paling dominan. Bahasa *query* SQL dikembangkan untuk *database* relasional sebagai bagian proyek Sistem R dari IBM. SQL distandarisasi di akhir tahun 1980 dan SQL-92 diadopsi oleh *American National Standards Institute* (ANSI) dan *International Standards Organization* (ISO). Pada saat yang hampir bersamaan, sekitar tahun 1980, pemasok perangkat lunak mulai mengembangkan paket-paket DBMS berskala kecil pada pasar komputer mikro.

Pada permulaan tahun 1990, banyak bidang sistem *database* dikembangkan. Penelitian *database* meliputi bahasa *query* yang *powerfull*, model data yang lengkap, dan penekanan pada dukungan analisis data yang kompleks. Beberapa *vendor* memperluas sistemnya dengan kemampuan menyimpan tipe data baru misalnya *image* dan *text* serta kemampuan *query* yang kompleks. Suatu gerakan menarik yang sedang berlangsung dalam hal *database* adalah *Knowledge Discovery in Database* (KDD). KDD adalah istilah besar yang menjelaskan semua kegiatan pada data yang tersimpan dalam *database* yang besar dan rumit. KDD mencakup sejumlah istilah yang sekarang mendapat banyak perhatian dalam literatur komputer, yaitu *data warehousing* dan *data mining*.

Beberapa tahun-tahun terakhir ini, pengembangan DBMS juga berfokus pada pasar komputer mikro dan telah menerapkan struktur relasional. **Microsoft Access** adalah contoh sistem manajemen *database* relasional untuk komputer mikro. Perkembangan teknologi *database* dapat dilihat pada Gambar 2.18.

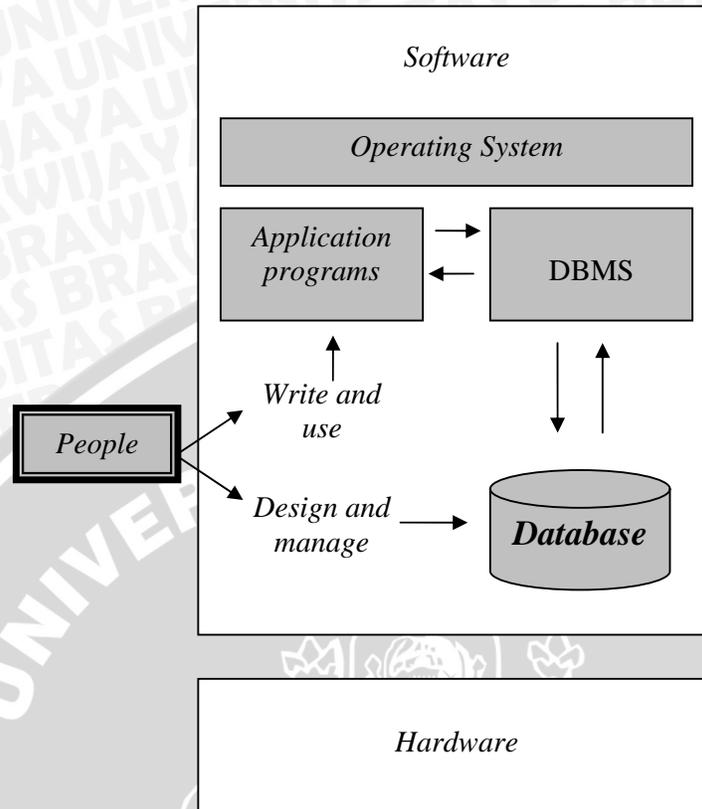


Gambar 2.18 Perkembangan Teknologi *Database*  
 Sumber: *National Institute of Standards and Technology (NIST)*  
 Boulder, CO, U.S.A. (2002:1)

### 2.6.2 Pengertian *Database*

Secara sederhana *database* dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini pengertian akses dapat mencakup cara memperoleh data maupun cara memanipulasi data seperti menambah, merubah, menyimpan, serta menghapus data.

Manajemen modern selalu mengikutsertakan informasi sebagai sumber daya penting yang setara dengan sumber daya manusia, modal, mesin, dan material. Menurut Gordon B. Davis (1997: 28) pengertian informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang. Dengan adanya *database* maka suatu organisasi akan lebih terpenuhi informasinya karena ketersediaan dan kemudahan dalam mengakses data.



Gambar 2.19 *Informations Systems*

Sumber: Peter Rob, *Databases-Design, Development, and Deployment* (2000:6)

Pada Gambar 2.19 menjelaskan tentang keutamaan *database* dalam suatu sistem informasi. *Database* merupakan sumber informasi yang berperan dalam suatu pengambilan keputusan.

Sebelum melihat lebih jauh mengenai *database*, maka perlu mengetahui beberapa istilah dalam *database*, yaitu:

1. **Enterprise**, adalah organisasi mandiri yang bergerak dalam bidang komersil, sains, teknik, dan lainnya. Contohnya: Pabrik, Bank, Universitas, dan lain-lain.
2. **Entity**, adalah obyek dalam *enterprise* yang akan disajikan di dalam *database*. Contoh: *supplier*, gudang, pegawai, *part*, dan lain-lain.
3. **Atribut atau elemen data**, adalah sifat atau karakteristik suatu entitas yang menjelaskan secara detail tentang entitas tersebut. Contohnya: kode pegawai, alamat, golongan, gaji, dan lain-lain.

4. **Relasi**, adalah suatu asosiasi hubungan antara dua atau lebih *entity*. Contohnya: mahasiswa mengambil mata kuliah, pasien dirawat dokter, dan lain-lain.

Setiap proyek dalam sistem *database* dapat dipresentasikan dengan menggunakan beberapa *entity* dan setiap *entity* dapat terdiri dari beberapa elemen data (atribut).

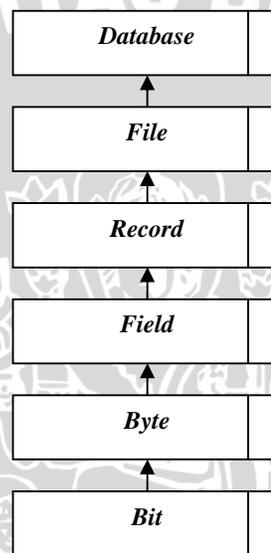
Menurut P. Budi Santoso (2005:2) *database* adalah kumpulan *file-file* yang saling berrelasi, relasi tersebut ditunjukkan dengan kunci dari tiap *file* yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi. Dalam satu *file* terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan *entity* yang beragam. Satu *record* terdiri dari *field-field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam suatu *record*.

### 2.6.3 Hierarki Data

Data harus disusun secara teratur agar pengolahannya dapat dilakukan dengan baik dan efisien. Menurut P. Budi Santoso (2005:1) pengorganisasian data dapat dibagi dalam enam tingkatan, yaitu:

1. **Bit**, adalah suatu sistem angka biner yang terdiri atas dua macam nilai saja, yaitu 0 dan 1. Sistem angka biner merupakan dasar-dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin (komputer) yang merupakan sekumpulan komponen elektronik dan hanya dapat membedakan dua keadaan saja.
2. **Byte**, adalah bagian terkecil yang dialamatkan dalam memori. *Byte* merupakan sekumpulan *bit* yang secara konvensional terdiri atas kombinasi delapan *bit*. Satu *byte* digunakan untuk mengkodekan satu buah karakter dalam memori.
3. **Field**, adalah unit terkecil yang disebut data. *Field* merupakan sekumpulan *byte* yang mempunyai makna.

4. **Record**, adalah kumpulan *item* yang secara logis saling berhubungan. Setiap *record* dapat dikenali oleh sesuatu yang mengenalinya, yaitu *field* kunci.
5. **File**, adalah kumpulan *record* yang sejenis dan secara logis berhubungan. Pembuatan dan pemeliharaan *file* adalah faktor yang sangat penting dalam sistem informasi manajemen yang memakai komputer.
6. **Database**, merupakan kumpulan *file-file* yang berhubungan secara logis dan digunakan secara rutin pada operasi-operasi sistem informasi manajemen.



Gambar 2.20 Hierarki Data  
Sumber: P. Budi Santoso (2005:2)

#### 2.6.4 Tipe File

*Database* terbentuk dari kumpulan *file-file*. *File* dalam pemrosesan aplikasi dikategorikan sebagai berikut:

##### 1. File Induk (Master File)

*File* induk merupakan *file* yang penting dalam sistem dan akan tetap ada selama siklus hidup dari sistem informasi itu berputar. *File* induk ini dibedakan menjadi 2 macam, yaitu:

- *File* induk acuan (*reference master file*), adalah *file* induk yang nilai *record*-nya relatif statis. Misalnya *file* daftar gaji dan *file* mata pelajaran.

- *File* induk dinamik (*dynamic master file*), adalah *file* induk yang nilai *record*nya sering berubah atau sering dimutakhirkan (*update*) sebagai hasil dari suatu transaksi. Misalnya *file* induk data barang, yang setiap saat harus di *update* bila terjadi transaksi.

## 2. *File Transaksi (Transaction File)*

*File* ini bisa disebut *file input*; digunakan untuk merekam data hasil dari transaksi yang terjadi. Misalnya *file* penjualan yang berisi data hasil transaksi penjualan.

## 3. *File Laporan (Report File)*

*File* ini bisa disebut *output file*, yaitu *file* yang berisi informasi yang akan ditampilkan. Biasanya struktur dari *file* laporan ada beberapa macam, sesuai dengan kepada siapa saja laporan tersebut akan didistribusikan.

## 4. *File Sejarah (History File)*

*File* ini bisa disebut *file arsip (archival file)*, merupakan *file* yang berisi data masa lalu yang sudah tidak aktif lagi, tetapi masih disimpan sebagai arsip.

## 5. *File Pelindung (Backup File)*

*File* ini merupakan salinan dari *file-file* yang masih aktif di dalam *database* pada suatu saat tertentu. *File* ini digunakan sebagai pelindung atau cadangan bila *file database* yang aktif mengalami kerusakan atau hilang.

## 6. *File kerja (Working File)*

*File* kerja dibuat oleh suatu proses program secara sementara karena memori komputer tidak mencukupi, atau untuk menghemat pemakaian memori dalam suatu proses, dan akan dihapus bila proses telah selesai.

### 2.6.5 Tujuan Sistem Database

Tujuan sistem *database* antara lain:

#### 1. **Isolasi data**

Menempatkan setiap data pada tempatnya masing-masing.

#### 2. **Multi User**

Saat perusahaan mengambil pendekatan berorientasi masalah, pertama masalah didefinisikan. Kemudian keputusan yang diperlukan untuk memecahkan masalah didefinisikan dan untuk tiap keputusan definisikan informasi yang diperlukan. Selanjutnya, pemrosesan yang diperlukan untuk menghasilkan informasi ditentukan dan akhirnya data yang diperlukan pemrosesan ditetapkan.

Tujuan dari manajemen *database* adalah sebagai berikut:

1. Menyediakan tempat penyimpanan masal untuk data yang relevan.
2. Memudahkan pemakai dalam mengakses data.
3. Memungkinkan respon yang segera atas permintaan data dari pemakai.
4. Melakukan modifikasi terakhir dengan segera pada *database*.
5. Memungkinkan secara serentak dan bersamaan beberapa pemakaian yang berarti juga meningkatkan kebebasan data sehingga berguna untuk beberapa program.
6. Meminimalisasi duplikasi dan redudansi dalam penyimpanan data.
7. Melindungi data dari gangguan kerusakan atau pemakaian oleh orang yang tidak terotorisasi.

Untuk mendapatkan data secara optimal, diperlukan kriteria-kriteria sebagai berikut:

**1. Kebenaran struktur (*consistency*)**

Konsistensi dengan jalur definisi dan informasi organisasi.

**2. Kemudahan**

Mudah untuk dimengerti oleh orang yang tidak mempunyai pendidikan khusus atau pemakai sistem yang bukan profesional komputer.

**3. Tidak ada *redudancy***

Tidak ada informasi yang berlebihan atau ditulis berkali-kali, dengan demikian untuk satu set informasi hanyalah disimpan dengan benar di satu tempat.

**4. Dapat dipakai bersama (*sharebility*)**

Kondisi data tidak spesifik dan hanya dapat dipakai oleh satu aplikasi, namun diharapkan dapat diakses oleh beberapa aplikasi, sesuai dengan kebutuhan.

### 5. Mudah dikembangkan (*extenbility*)

Mampu mengakomodasi kebutuhan baru dari suatu aplikasi dengan akibat yang ringan terhadap perubahan data dasarnya.

### 6. Kesatuan (*Integrity*)

Konsistensi dengan jalur bisnis yang digunakan dan pengelolaan informasi.

## 2.6.6 Komponen Sistem Database

Di dalam suatu sistem *database* terdapat empat komponen utama, yaitu:

### 1. Data

Data merupakan komponen dasar dari informasi yang akan diproses untuk menghasilkan informasi. Himpunan data harus memiliki sifat yang unik, yaitu:

- Saling keterkaitan (*Interrelated*), data tersebut saling berkaitan atau terintegrasi dan tersimpan secara terorganisir di dalam suatu media penyimpanan.
- Kebersamaan (*Shared*), data yang terintegrasi tersebut dapat diakses oleh berbagai pengguna, tetapi hanya satu yang dapat merubahnya, yaitu *Database Administrator* (DBA).

### 2. Hardware

*Hardware* merupakan piranti keras yang dibutuhkan manajemen *database*, meliputi:

- Peralatan penyimpanan data, contohnya *hard disk*.
- Peralatan input, contohnya *keyboard, mouse, scanner* dan lain-lain.
- Peralatan output, contohnya printer dan monitor.
- Peralatan komunikasi data, contohnya terminal, modem, dan lain-lain.

### 3. Software

*Software* merupakan kumpulan dari perintah atau fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu. Dalam sistem *database*, *software* harus menyediakan fasilitas sebagai berikut:

- Membuat *file*.
- Menyisipkan data.
- Menampilkan data.
- Menghapus data.
- Kontrol keamanan.
- Kontrol keterpaduan.

#### 4. Pemakai (*User*)

*User* merupakan orang yang terlibat dalam kegiatan sistem informasi, yaitu: *Database Administrator* (DBA), pemrogram aplikasi, dan pemakai akhir (*End-User*).

### 2.6.7 Database Relasional

Model *database* relasional ditemukan pertama kali oleh E.F. Cood (Bapak *Database Relasional*). Pada *database* relasional, *database* akan dipilah-pilah ke dalam berbagai tabel dua dimensi. Setiap tabel selalu terdiri atas lajur horisontal yang biasa disebut Baris Data (*Raw/Record*) dan lajur vertikal yang biasa disebut Kolom (*Column/Field*). Hubungan antar tabel ditentukan oleh integritas referensial (*referential integrity*). Integritas referensial merupakan fitur dalam *database* yang mengatur hubungan antara kunci utama (*primary key*) dan kunci tamu (*foreign key*) milik tabel-tabel yang berada dalam *database* relasional dengan tujuan untuk menjaga data (Abdul Kadir, 2003:99). Melalui integritas referensial, ada jaminan bahwa nilai dalam suatu tabel benar-benar menunjukkan ke suatu nilai yang memang ada. Jadi bisa dikatakan bahwa tabel dapat dihubungkan satu sama lain melalui nilai kolom yang disebut kunci (*key*).

*Key* adalah elemen *record* yang dipakai untuk menemukan *record* tersebut pada waktu akses, atau bisa juga digunakan untuk mengidentifikasi setiap *entity/ record/ baris* (Al Bahra, 2005:138). Jenis *key* yang biasa digunakan adalah:

#### 1. *Primary Key* (Kunci primer)

Kriteria *primary key* adalah sebagai berikut:

- *Key* tersebut lebih natural untuk digunakan sebagai acuan.
- *Key* tersebut lebih sederhana.
- *Key* tersebut terjamin keunikannya.

## 2. **Foreign Key (Kunci tamu)**

*Foreign key* merupakan sembarang atribut yang menunjuk kepada *primary key* pada tabel lain. *Foreign key* akan terjadi pada suatu relasi yang memiliki kardinalitas *one to many* (satu ke banyak) atau *many to many* (banyak ke banyak). *Foreign key* biasanya selalu diletakkan pada tabel atau relasi yang mengarah ke banyak.

Keuntungan model *database* relasional antara lain sebagai berikut:

1. Data dapat diakses secara cepat.
2. Struktur *database* mudah diubah.
3. Data disajikan secara logis sehingga pengguna tidak perlu mengetahui bagaimana data disimpan.
4. Pengguna mudah membuat *query* yang kompleks untuk mengambil data.
5. Pengguna mudah menerapkan integritas data.
6. Data sering lebih akurat.
7. Pengguna mudah membuat dan memodifikasi program aplikasi.
8. Bahasa standar (SQL) sudah dibuat.

### 2.6.8 **Entity Relationship Diagram (ERD)**

*Entity Relationship Diagram* atau disingkat ERD adalah alat pemodelan data utama dan akan membantu mengorganisasikan data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur *database* yang baik sehingga data dapat disimpan dan diambil secara efisien ([www.infocom.cqu.edu.au](http://www.infocom.cqu.edu.au)). *Entity Relationship Diagram* juga dikenal dengan sebutan *E-R diagram*.

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada 3 macam simbol yang digunakan yaitu :

#### 1. **Entitas (Entity)**

Pada *E-R diagram*, entitas digambarkan dengan sebuah bentuk persegi panjang. Entitas adalah sesuatu apa saja yang ada di dalam sistem nyata atau abstrak di mana data tersimpan. Entitas diberi nama dengan kata

benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, dan kejadian (terdapat unsur waktu di dalamnya).

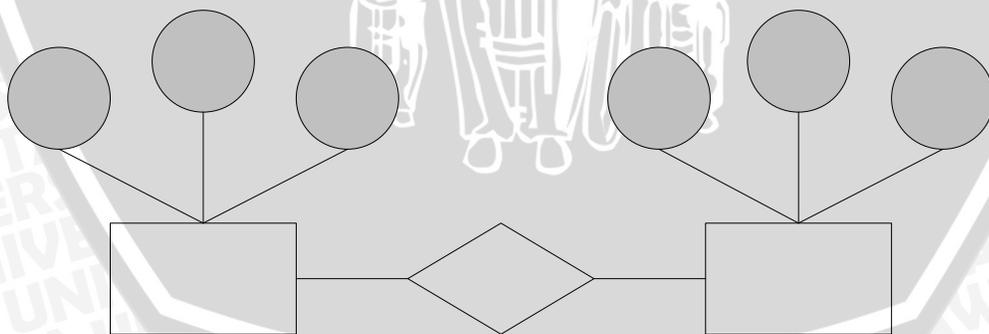
## 2. Atribut (*Attribute*)

Pada E-R *diagram*, atribut digambarkan dengan sebuah bentuk *ellips*. Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun relasi. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun relasi.

## 3. Relasi (*Relationship*)

Pada E-R *diagram*, relasi dapat digambarkan dengan sebuah bentuk belah ketupat. Relasi adalah hubungan alamiah yang terjadi antar entitas. Pada umumnya penghubung diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya. Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk persegi panjang.

Sebagai contoh adalah bagian sistem *database* universitas yang terdiri atas mahasiswa dan mata kuliah. Gambar 2.21 menunjukkan diagram ER dari contoh tersebut. Diagram tersebut menunjukkan bahwa ada dua kumpulan entitas, yaitu mahasiswa dan mata kuliah, relasi mahasiswa mengambil mata kuliah, dan atribut NIM, Nama, dan Alamat pada entitas Mahasiswa serta atribut Kode MTK, Nama MTK, dan SKS pada entitas Mata kuliah.

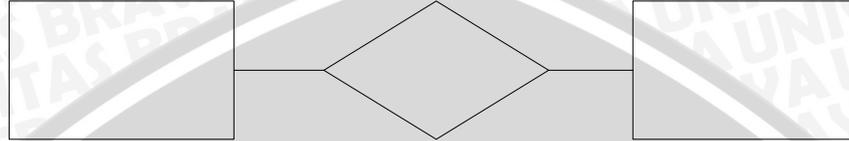


Gambar 2.21 Diagram E-R  
Sumber: Janner, Basis Data (2006:60)

Ada tiga tipe relasi, yaitu:

**1. Satu ke satu (*One-to-one*)**

Misalnya dalam suatu perusahaan mempunyai aturan satu sopir hanya boleh menangani satu kendaraan karena alasan tertentu.



Gambar 2.22 Hubungan Satu ke Satu  
Sumber: F Irmansyah (2003:9)

**2. Satu ke banyak (*One-to-many*)**

Misalnya suatu sekolah selalu mempunyai asumsi bahwa satu kelas terdiri dari banyak siswa tetapi tidak sebaliknya, yaitu satu siswa tidak dapat belajar pada kelas yang berbeda.

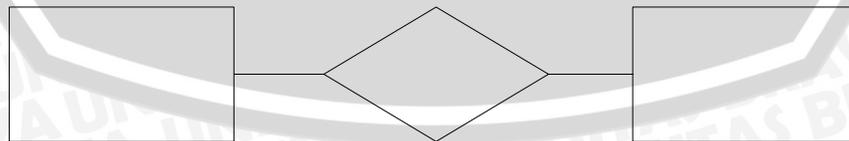


1

Gambar 2.23 Hubungan Satu ke Banyak  
Sumber: F Irmansyah (2003:10)

**3. Banyak ke banyak (*Many-to-many*)**

Misalnya pada saat pengambilan mata kuliah, dimungkinkan satu mahasiswa mengambil beberapa mata kuliah, tetapi semua mata kuliah yang dia ambil juga diambil oleh mahasiswa-mahasiswa lain.



Gambar 2.24 Hubungan Banyak ke Banyak  
Sumber: F Irmansyah (2003:10)

1

### 2.6.9 Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang *database* relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang, lalu menghilangkan data terduplikasi dari tabel relasional ([www.utecas.edu](http://www.utecas.edu)).

Pada proses normalisasi selalu diuji pada beberapa kondisi. Apakah ada kesulitan saat menambah (*insert*), menghapus (*delete*), mengubah (*update*), dan membaca (*retrieve*) pada suatu *database*. Bila ada kesulitan pada pengujian tersebut maka relasi tersebut dipecahkan pada beberapa tabel lagi atau dengan kata lain perancangan belumlah mendapat *database* yang optimal. Tujuan normalisasi adalah membuat kumpulan tabel relasional yang bebas dari pengulangan informasi dan memudahkan identifikasi *entity* atau obyek. Tanpa normalisasi, sistem *database* menjadi tidak akurat, lambat, tidak efisien, serta tidak memberikan data yang diharapkan ([www.microsoft-accesssolutions.co.uk](http://www.microsoft-accesssolutions.co.uk)).

Bentuk-bentuk normalisasi yaitu:

#### 1. Bentuk Tidak Normal (*Unformalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

#### 2. Bentuk Normal Kesatu (1NF/ *First Normal Form*)

Bentuk normal kesatu mempunyai ciri yaitu setiap data dibentuk dalam *flat file* (*file* datar atau rata), data dibentuk dalam satu *record* demi satu *record* dan nilai dari *field-field* berupa "*atomic value*". Tidak ada set atribut yang berulang-ulang atau atribut bernilai ganda (*multi value*). Tiap *field* hanya satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti mendua, hanya satu arti saja dan bukanlah pecahan kata-kata sehingga artinya lain. Suatu relasi memenuhi 1NF jika dan hanya jika setiap atribut dari relasi tersebut hanya memiliki nilai tunggal dalam satu baris atau *record*.

### 3. Bentuk Normal Kedua (2NF/ *Second Normal Form*)

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama (*primary key*). Sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci *field*. Kunci *field* haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

Suatu relasi memenuhi 2NF jika dan hanya jika:

- a. Memenuhi 1NF.
- b. Setiap atribut yang bukan kunci utama tergantung secara fungsional terhadap semua atribut kunci dan bukan hanya sebagian atribut.

### 4. Bentuk Normal Ketiga (3NF/ *Third Normal Form*)

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key*.

Suatu relasi memenuhi bentuk 3NF jika dan hanya jika:

- a. Relasi tersebut memenuhi 2NF.
- b. Setiap atribut bukan kunci tidak tergantung secara fungsional kepada atribut bukan kunci yang lain dalam relasi tersebut.

#### 2.6.10 Langkah-langkah Perancangan *Database*

Tahapan perancangan (*design*) memiliki tujuan untuk membuat sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dengan pemilihan sistem yang terbaik. Adapun tahap-tahap perancangan *database* adalah sebagai berikut:

##### 1. *Conceptual design*

*Conceptual design* adalah model lengkap yang merangkap keseluruhan struktur organisasi data, yang tidak mungkin terlepas dari sistem manajemen *database* atau pertimbangan implementasi lainnya. Konsep data yang digunakan adalah model data relasional. Menurut Mc Fadden (1994) konsep data yang termasuk di dalamnya adalah:

a. *Entity identification*

Pada tahap ini dilakukan identifikasi *entity* yang terlibat. Setiap *entity* tersebut merupakan calon dari tabel yang akan dibuat.

b. *Functional decomposition*

Suatu proses untuk menguraikan *entity* yang terlibat tersebut menjadi lebih detail. Selanjutnya hasil dari *functional decomposition* disebut sebagai atribut.

c. *Entity relationship diagram*

Pada tahap ini dilakukan penentuan hubungan atau relasi yang terjadi antar *entity*.

**2. Logical design**

Konsep model data yang telah terbentuk pada tahap *conceptual design* dibawa ke bentuk yang logis, yaitu:

a. Membuat model relasional

*Entity-entity* yang telah teridentifikasi menjadi sebuah tabel dengan *field* berasal dari atribut pada setiap *entity*.

b. Normalisasi

Semua struktur tabel yang telah terbentuk dinormalisasikan dengan menggunakan tahapan dan aturan yang telah dijelaskan pada sub bab sebelumnya.

**3. Physical design**

Model data yang telah terbentuk pada tahap *logical design* dibawa ke bentuk suatu penyimpanan data yang nyata dengan menggunakan sebuah *software*. Oleh karena itu pada tahap ini diawali dengan pemilihan *software* yang akan dipakai. Setelah itu baru dirancanglah struktur tabel sesuai dengan kebutuhan kita untuk melakukan penyimpanan data.

**4. Implementations**

Pada tahap ini dilakukan pengembangan *database* yaitu merancang hierarki *menu*, *form* dan *report* beserta program (*procedure* atau *function*) yang menghubungkan ketiganya. Tahap ini diakhiri dengan memasukkan data ke dalam sistem *database*.

### 2.6.11 Sistem Manajemen *Database* (DBMS)

Sistem Manajemen *Database* (*Database Management System*)-DBMS adalah perangkat lunak atau program komputer yang dirancang secara khusus untuk memudahkan pengelolaan *database*. DBMS memungkinkan untuk menciptakan *database* dalam penyimpanan data langsung pada komputer, memelihara isinya, dan menyediakan isi tersebut bagi pemakai tanpa pemrograman khusus yang mahal.

Ketika perusahaan atau individu memutuskan apakah akan menggunakan suatu DBMS, keuntungan dan kerugiannya harus dipertimbangkan. Menurut *McLeod* (2001:270) keuntungan DBMS antara lain:

1. **Mengurangi pengulangan data.** Jumlah total *file* dikurangi dengan menghapus *file-file* duplikat. Juga hanya terdapat sedikit data yang sama di beberapa *file*.
2. **Mencapai independensi data.** Spesifikasi data disimpan dalam skema pada tiap program aplikasi. Perubahan dapat dibuat pada struktur data tanpa mempengaruhi program yang mengakses data.
3. **Mengintegrasikan data beberapa *file*.** Ketika *file* dibentuk sehingga menyediakan kaitan logis, organisasi fisik tidak lagi menjadi kendala.
4. **Mengambil data dan informasi secara cepat.** Hubungan-hubungan logis, bahasa manipulasi data, serta bahasa *query* memungkinkan pengguna mengambil data dalam hitungan detik atau menit saja.
5. **Meningkatkan keamanan.** DBMS maupun komputer mikro dapat menyertakan beberapa lapis keamanan seperti kata sandi (*password*), *directory* pemakai, dan bahasa sandi (*encryption*). Sehingga data yang dikelola DBMS akan lebih aman dari pada data lainnya.

Sedangkan kerugian DBMS antara lain:

1. **Memperoleh perangkat lunak yang mahal.** DBMS *mainframe* masih sangat mahal. Walaupun harga DBMS berbasis komputer mikro lebih murah, tetapi tetap merupakan pengeluaran yang besar bagi suatu organisasi kecil.

2. **Memperoleh konfigurasi perangkat keras yang besar.** DBMS sering memerlukan kapasitas penyimpanan dan memori yang besar daripada program aplikasi lain.

3. **Mempekerjakan dan mempertahankan staf DBA.** DBMS memerlukan pengetahuan khusus agar dapat dimanfaatkan kemampuannya secara penuh. Pengetahuan khusus ini disediakan paling baik oleh para pengelola *database (Database Administrator)*-DBA.

Salah satu macam DBMS yang populer pada dewasa ini berupa RDBMS (*Relational Database Management System*), yang menggunakan model basis data relasional atau dalam bentuk tabel-tabel yang saling berhubungan. RDBMS merupakan suatu paket perangkat lunak yang kompleks digunakan untuk memanipulasi *database*.

Ada tiga prinsip dalam RDBMS, yaitu:

1. **Data definition**

Mendefinisikan jenis data yang akan dibuat (dapat berupa angka maupun huruf), cara relasi data, validasi dan lain-lain.

2. **Data manipulations**

Data yang telah dibuat dan didefinisikan tersebut akan dilakukan beberapa pengerjaan, seperti menyaring data, melakukan proses *query* dan sebagainya.

3. **Data control**

Bagian ini berkenaan dengan cara mengendalikan data seperti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak *user*, dan sebagainya.

### 2.6.12 SQL

SQL (*Structured Query Language*) adalah bahasa komputer standar ANSI (*American National Standard Institute*) untuk mengakses dan memanipulasi sistem *database*. Pernyataan SQL digunakan untuk mengambil dan meng-*update* data dalam *database*. SQL bekerja dengan program *database* seperti Oracle, Sybase, **Microsoft Access**, Microsoft SQL Server, dan sebagainya.

Perintah dalam SQL dibedakan menjadi tiga bagian, yaitu:

### 1. DDL (*Data Definition Language*)

DDL adalah kelompok perintah dalam SQL yang digunakan untuk membuat dan mendefinisikan struktur *database*, yakni membuat *database*, membuat tabel, relasi antar tabel, membuat *view* data, menghapus *database*, menghapus tabel, dan tipe data dari setiap kolom data yang ada dalam tabel.

### 2. DML (*Data Manipulation Language*)

DML adalah kelompok perintah dalam SQL yang digunakan untuk melakukan manipulasi data dan membangun *query* untuk mengambil data (SELECT) beberapa tabel, menyisipkan (INSERT) data baru, memperbaiki (UPDATE) data, dan menghapus (DELETE) data.

### 3. DCL (*Data Control Language*)

DCL adalah kelompok perintah dalam SQL yang digunakan untuk mengatur pemakai *database*, siapa saja yang dapat melakukan pengolahan data, siapa saja yang boleh mengakses data dalam tabel, hak akses apa saja yang diperbolehkan, melihat, menambah, memperbaiki, dan menghapus data.

#### 2.6.13 *Microsoft Access*

*Microsoft Access* merupakan salah satu contoh produk RDBMS yang sangat populer di lingkungan *Windows*. *Microsoft Access* adalah program aplikasi pengolah *database* yang merupakan bagian dari *Microsoft Office*, di mana di dalamnya terdapat beberapa *software* lain seperti: *Microsoft Word*, *Microsoft Excel*, *Microsoft Power point* dan sebagainya. *Microsoft Access* memiliki kinerja utama yaitu: membuat dan memodifikasi tabel, melakukan *query*, membuat formulir *entri* data, dan membuat laporan. Kemampuan memodifikasi keempat hal tersebut juga didukung oleh keterkaitannya dengan bahasa pemrograman lain, salah satunya yaitu *Visual Basic*. Pada *Microsoft Access*, istilah kolom yang biasa dipakai pada *database* relasional disebut *field*, sedangkan istilah baris biasa disebut *record*. Setiap *field* yang didefinisikan harus memiliki tipe data. Tipe data pada *Microsoft Access* dijelaskan pada tabel berikut:

Tabel 2.1 Daftar Tipe Data

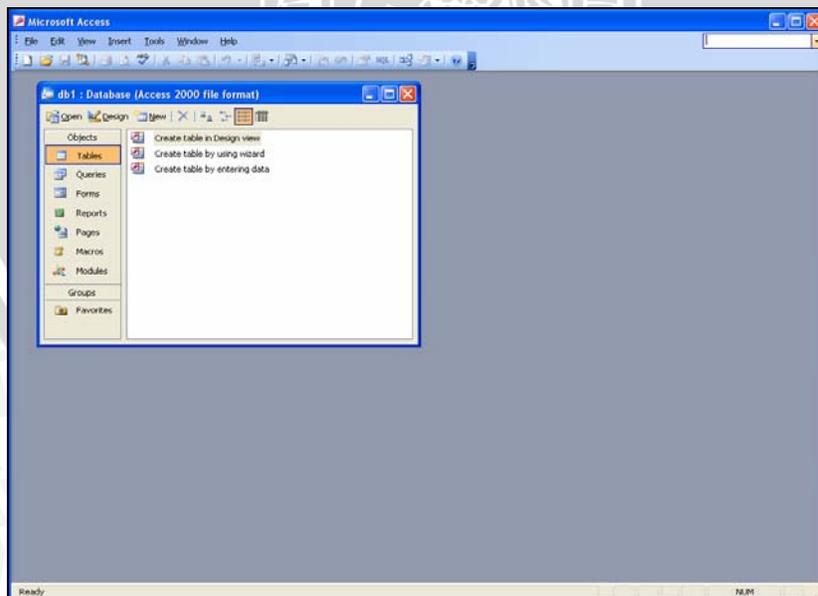
| TIPE DATA            | KETERANGAN                                                                                                                                                                                                                                                                                                                                                                                                | UKURAN                                                 |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| Teks                 | Merupakan tipe data bawaan yang akan dipilihkan oleh Microsoft Access ketika suatu <i>field</i> baru diciptakan. Tipe ini digunakan untuk menyatakan teks atau data yang bisa mengandung huruf, angka, dan karakter-karakter lain seperti tanda * dan &. Contoh pemakaiannya antara lain untuk nama orang, alamat, dan bahkan untuk angka yang tidak dimaksudkan untuk dihitung (misalnya nomor telepon). | Maksimum terdiri atas 255 karakter                     |
| <i>Memo</i>          | Merupakan tipe data teks yang berukuran besar.                                                                                                                                                                                                                                                                                                                                                            | Dapat mencapai 65.535 karakter                         |
| <i>Number</i>        | Merupakan tipe data untuk suatu nilai bilangan yang bisa dihitung.                                                                                                                                                                                                                                                                                                                                        | 1,2,4,8, atau 16 <i>byte</i> tergantung tipe detailnya |
| <i>Date/time</i>     | Nilai jam dan tanggal dimulai dari tahun 100.                                                                                                                                                                                                                                                                                                                                                             | 8 <i>byte</i>                                          |
| <i>Currency</i>      | Merupakan tipe data untuk nilai uang. Keakuratan sampai 15 digit di sebelah kiri tanda pecahan dan 4 digit di sebelah kanan tanda pecahan.                                                                                                                                                                                                                                                                | 8 <i>byte</i>                                          |
| <i>AutoNumber</i>    | Menyatakan nilai yang urut. Nilai dibangkitkan oleh Microsoft Access dan tidak dapat diubah oleh pemakai.                                                                                                                                                                                                                                                                                                 | 4 <i>byte</i>                                          |
| <i>Yes/No</i>        | Menyatakan data yang hanya memiliki dua kemungkinan nilai saja. Contohnya benar atau salah, pria atau wanita.                                                                                                                                                                                                                                                                                             | 1 <i>bit</i>                                           |
| <i>OLE Object</i>    | Menyatakan data obyek seperti lembar kerja Excel, dokumen Word yang dihubungkan atau diletakkan ke Microsoft Access.                                                                                                                                                                                                                                                                                      | Sampai 1 <i>gigabyte</i>                               |
| <i>Hyperlink</i>     | Menyatakan data alamat <i>hyperlink</i> (misalnya digunakan sebagai Web atau untuk lompatan ke Word).                                                                                                                                                                                                                                                                                                     | Sampai 2.048 karakter                                  |
| <i>Lookup Wizard</i> | Memungkinkan berhubungan dengan nilai pada tabel lain melalui fasilitas kotak daftar (list box) atau kotak kombo ( <i>combo box</i> ).                                                                                                                                                                                                                                                                    | Biasanya 4 <i>byte</i>                                 |

Sumber: Abdul Kadir, 2002:23

### 2.6.13.1 Tampilan Utama *Microsoft Access*

Tampilan utama dari jendela *Microsoft Access* ditunjukkan pada Gambar 2.25. Tampilan utama tersebut terdiri dari beberapa bagian, antara lain:

1. **Baris Judul**, berisi nama program *Access* serta nama *database* yang sedang dibuka.
2. **Baris Menu**, terdiri dari sejumlah nama operasi menu, seperti: *File*, *Edit*, *View*, *Insert*, *Tools*, *Window*, dan *Help*. Pada setiap menu terdiri dari kumpulan sub menu yang memiliki fungsi dan kegunaan masing-masing.
3. **Baris *Toolbar***, *toolbar* memiliki fungsi yang sama dengan menu. Perbedaannya adalah pada *toolbar*, menu ditampilkan dalam bentuk *icon-icon* untuk mempercepat pengaksesan menu, seperti: *New*, *Open*, *Save*, *Research*, *Print*, dan lain-lain.
4. **Baris Status**, menampilkan keterangan dari *item* atau proses yang sedang diaktifkan. Pada tampilan awal *Access*, baris status memunculkan tulisan *READY*.
5. ***Database Container***, merupakan kotak atau jendela *database* yang digunakan untuk aktivitas utama dalam pembuatan program.



Gambar 2.25 Tampilan Utama *Microsoft Access* 2003

### 2.6.13.2 Obyek *Microsoft Access*

Ada tujuh obyek yang dimiliki *Microsoft Access*, antara lain:

1. **Tabel**, merupakan struktur dasar tempat data disimpan di dalam *database* sekaligus obyek pertama yang harus dibuat. Tabel terdiri dari baris (*record*) dan kolom (*field*). Tabel yang dibuat dapat berjumlah satu atau lebih disesuaikan dengan kebutuhan.
2. **Form**, merupakan suatu piranti yang memungkinkan untuk melihat dan mengedit data di dalam tabel. *Form* dapat dibuat lebih menarik dan mudah digunakan dari pada tabel.
3. **Query**, bagian ini digunakan untuk mengatur data mana saja dari suatu tabel yang perlu ditampilkan. Dalam *query* ini juga dapat diatur kriteria atau syarat penampilan suatu data serta bagaimana data tersebut diurutkan.
4. **Report**, bagian ini digunakan untuk membuat laporan dari suatu data yang telah diolah menjadi informasi. Dari *report* dapat langsung dibuat *print out* informasi yang telah diolah.
5. **Pages**, bagian ini berfungsi untuk menciptakan halaman Web berupa data *Access Page*.
6. **Macros**, bagian ini berfungsi untuk mengotomasikan perintah-perintah yang dikehendaki dalam mengolah data.
7. **Modules**, bagian ini digunakan untuk pembuatan dan pengolahan modul di dalam *Visual Basic*.

### 2.6.14 *Microsoft Visual Basic 6.0*

*Microsoft Visual Basic 6.0* adalah salah satu *development tools* yang merupakan alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi *Windows*. Dalam pengembangan aplikasi, *Visual Basic 6.0* menggunakan pendekatan visual untuk merancang *user interface* dalam bentuk *form*, sedangkan untuk *coding* programnya menggunakan dialek bahasa *basic*.

Pada pemrograman visual, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian mengatur properti dari obyek-obyek yang

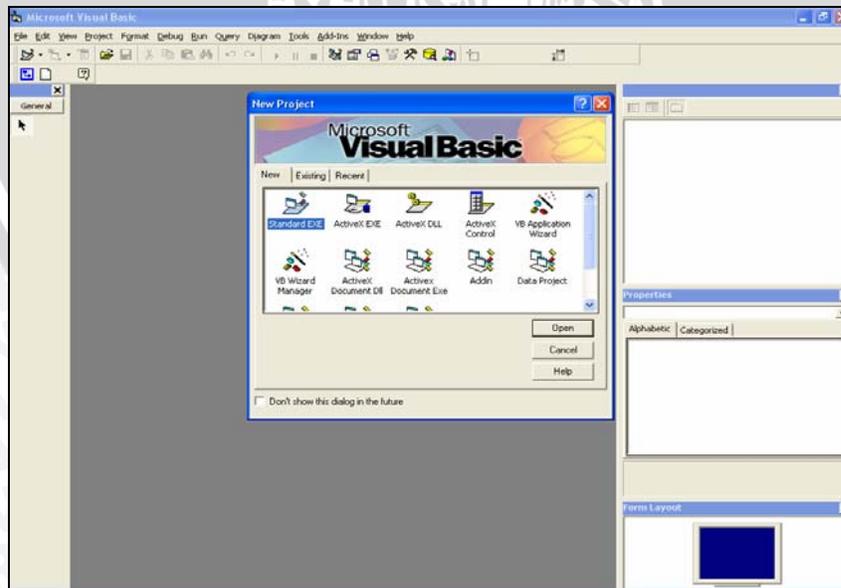
digunakan dalam *user interface*. Setelah tahap tersebut kemudian dilakukan penulisan kode program untuk menangani kejadian-kejadian (*event*).

#### 2.6.14.1 Tampilan Awal *Visual Basic 6.0*

Untuk memulai menjalankan *Visual Basic 6.0* dapat menggunakan langkah-langkah sebagai berikut: ***Start program – Microsoft Visual Studio 6.0 – Microsoft Visual Basic 6.0***, atau dengan menunjuk ikon *shortcut Visual Basic 6.0* pada *dekstopnya*. *Visual Basic 6.0* akan menampilkan kotak dialog *new project* secara otomatis saat pertama kali program ini dijalankan. Pada kotak dialog tersebut terdapat tiga pilihan tabulasi dengan keterangan sebagai berikut:

1. ***New***, pilihan ini digunakan untuk membuat *project* baru dengan berbagai macam pilihan.
2. ***Existing***, pilihan ini digunakan untuk membuka *project* yang pernah dibuat sebelumnya dengan menentukan *folder* sekaligus nama *file*.
3. ***Recent***, pilihan ini digunakan untuk membuka *project* yang telah dibuat dan terakhir kali dibuka.

Gambar 2.26 menunjukkan ilustrasi gambar tabulasi *New* dengan beberapa pilihan *project* yang dapat digunakan untuk membuat aplikasi. Untuk membuat *project* baru maka dipilih *Standard EXE* dan siap memasuki jendela *Visual Basic 6.0* dengan berbagai fasilitas-fasilitas yang dimilikinya.

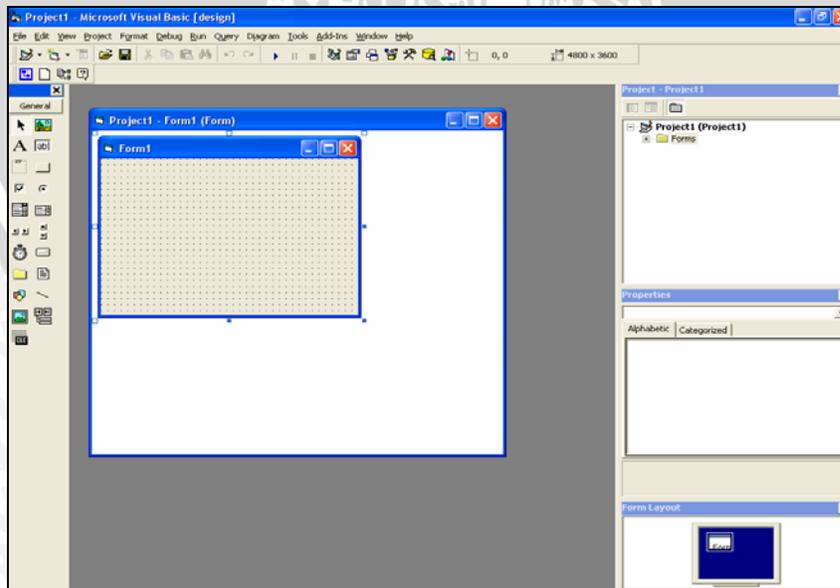


Gambar 2.26 Tabulasi *New* dengan Beberapa Pilihan *Project*

### 2.6.14.2 Komponen *Visual Basic* 6.0

Tampilan *New Project* dari *Visual Basic* 6.0 ditunjukkan pada Gambar 2.27. Tampilan tersebut terdiri dari komponen-komponen sebagai berikut:

1. **Title Bar**, merupakan judul dari program *Visual Basic* 6.0 yang terletak pada bagian paling atas. *Title bar* menunjukkan nama dari *file project* yang sedang dikerjakan.
2. **Menu Bar**, terletak di bawah *Title Bar* dan berfungsi untuk menampilkan pilihan menu atau perintah seperti: *File*, *Edit*, *View*, *Project*, *Format*, dan lain-lain.
3. **Toolbar**, terletak di bawah *Menu Bar* dan berfungsi untuk mengoperasikan program *Visual Basic*.
4. **Toolbox**, merupakan kotak yang berisi kumpulan tombol obyek atau kontrol untuk mengatur desain dari aplikasi yang akan dibuat.
5. **Project**, merupakan kumpulan *module* atau program aplikasi itu sendiri. Dalam *Visual Basic* 6.0, *file project* disimpan dengan nama *file* berakhiran *.VBP* (*Visual Basic Project*), *file* ini berfungsi untuk menyimpan seluruh komponen program.
6. **Properties Window**, merupakan jendela yang digunakan untuk menampung nama properti dari kontrol yang terpilih. Ukuran jendela *properties* ini dapat diatur sesuai kebutuhan.



Gambar 2.27 Tampilan Utama *Visual Basic* 6.0