

## 2. TEORI PENUNJANG

### 2.1. Latar Belakang

Di era globalisasi ini, banyak perusahaan yang mulai melakukan system komputerisasi untuk segala aktivitas yang ada di perusahaan. Demikian juga untuk pengendalian dan perencanaan material atau sering disebut juga sistem MRP. Sistem MRP ini telah memiliki popularitas dalam bidang Industri yang memanfaatkan kemampuan komputer melaksanakan perencanaan dan pengendalian persediaan dengan memperhatikan hubungan antar item, sehingga pengelolannya dapat lebih efisien dalam menentukan kebutuhan material secara cepat dan tepat.

Jika suatu perusahaan membuat sedikit produk yang simple, masih memungkinkan untuk membuat MRP (*Material Requirement Planning*) secara manual, karena tujuan dari MRP ini adalah menjelaskan komponen-komponen yang diperlukan untuk memenuhi *Master Production Schedule* dan berbasis pada *lead time*, untuk mengkalkulasikan periode-periode kapan komponen-komponen itu harus tersedia, dan juga berapa banyak yang harus dipenuhi untuk memenuhi kebutuhan tersebut. Sebelum adanya komputerisasi atau ketika perusahaan-perusahaan masih memakai sistem manual, perusahaan cenderung untuk mempunyai banyak *inventory* yang mengakibatkan *lead time* yang panjang. Selain itu, juga diperlukan informasi yang *up-to-date*, dan mampu untuk melakukan kalkulasi secara cepat.

Setelah ada komputer yang luar biasa cepat, akurat, dan ideal, dan dengan kemampuan menyimpan dan manipulasi data serta memberi informasi dengan cepat, memungkinkan perusahaan mempunyai alat untuk menggunakan sistem pengendalian dan perencanaan *modern* sebagaimana mestinya. Komputerisasi MRP mula-mula dikembangkan dilingkungan APICS (*American Production and Inventory System Society*) dalam suatu pengembangan program yang profesional. Dengan sistem komputerisasi, kesulitan-kesulitan yang biasanya

terjadi dalam pelaksanaan manajemen persediaan tradisional telah teratasi dengan dikenalnya suatu pendekatan sistem persediaan yang terperinci dan lebih baik.

## 2.2. Pengertian dan Perkembangan MRP

*Material Requirement Planning* (MRP) dapat didefinisikan sebagai suatu teknik atau set prosedur yang sistematis untuk merencanakan penjadwalan kuantitas dan waktu dalam proses pengendalian kebutuhan bahan terhadap komponen-komponen permintaan yang saling bergantung. Apabila pencatatan dan perencanaan yang dilakukan akurat, maka data tersebut dapat digunakan sebagai dasar dari perencanaan semua hal yang berkaitan dengan sumber daya perusahaan.

Menurut kamus APICS, MRP didefinisikan sebagai satu kesatuan set atau teknik yang menggunakan *bill of material* dan data *inventory* untuk mengkalkulasi kebutuhan akan material. MRP memberikan rekomendasi atau saran mengenai kapan dan berapa pemesanan material yang harus dilakukan untuk memenuhi kebutuhan produksi.

Teknik penjadwalan biasanya dilakukan oleh program komputer, dengan mengolah tiga hal utama yang merupakan *input*, yaitu : *bill of material*, *inventory data*, dan *master production schedule*, menjadi dua *output* utama, yaitu rekomendasi untuk pemesanan barang, dan pemberitahuan untuk penjadwalan ulang.

Apabila proses yang terjadi pada perusahaan tersebut lebih luas lagi, proses tersebut disebut MRP II atau *Manufacturing Resource Planning*, karena MRP II menyediakan koordinasi antara penjualan dan produksi. *Manufacturing Resource Planning* adalah sistem perencanaan dengan tujuan pengiriman jumlah material yang tepat pada waktu yang tepat, berbasis pada kombinasi order dan perkiraan (*forecast*). Dalam kamus APICS didefinisikan sebagai metode perencanaan yang efektif untuk semua bahan baku bagi perusahaan manufaktur.

Fokus pada sebuah MRP lebih ditujukan kepada:

- Bagaimana menghasilkan output untuk dapat memenuhi kebutuhan konsumen?

- Penyediaan input yang dibutuhkan untuk dapat memproduksi barang-barang kebutuhan konsumen tersebut
- Bagaimana proses untuk mengubah *input* menjadi *output*

### 2.3. Tujuan MRP

Suatu sistem MRP pada dasarnya bertujuan untuk merancang suatu sistem yang mampu menghasilkan informasi untuk mendukung aksi yang tepat baik berupa pembatalan pesanan, pesan ulang, atau penjadwalan ulang. Aksi ini sekaligus merupakan suatu pegangan untuk melakukan pembelian dan/ atau produksi.

Ada 4 macam yang menjadi ciri utama MRP, yaitu:

- a. Mampu menentukan kebutuhan pada saat yang tepat, kapan suatu pekerjaan akan selesai (material harus tersedia) untuk memenuhi permintaan produk yang dijadwalkan berdasarkan MPS yang direncanakan.
- b. Menentukan kebutuhan minimal setiap *item*, dengan menentukan secara tepat sistem penjadwalan.
- c. Menentukan pelaksanaan rencana pemesanan, dengan memberikan indikasi kapan pemesanan atau pembatalan suatu pesanan harus dilakukan.
- d. Menentukan penjadwalan ulang atau pembatalan atas suatu jadwal yang sudah direncanakan. Apabila kapasitas yang ada tidak mampu memenuhi pesanan yang dijadwalkan pada waktu yang dikehendaki, maka MRP dapat memberikan indikasi untuk melaksanakan rencana penjadwalan ulang (jika mungkin) dengan menentukan prioritas pesanan yang realistis. Jika penjadwalan ulang ini masih tidak memungkinkan untuk memenuhi pesanan, maka pembatalan terhadap suatu pesanan harus dilakukan.

Jadi, tujuan utama dari keseluruhan sistem MRP adalah untuk memenuhi tiga target utama suatu bisnis, yaitu: meningkatkan pelayanan terhadap konsumen, menciptakan sistem manufaktur perusahaan yang lebih efektif dan semaksimal mungkin menekan biaya inventori.

## 2.4. Input MRP

Ada 3 Input yang dibutuhkan dalam konsep MRP yaitu:

- a. Jadwal Induk Produksi (*Master Production Schedule*)
- b. Data *Inventory*
- c. *Bill of Material*

### 2.4.1. Jadwal Induk Produksi (*Master production schedule*)

Secara sederhana, *master production schedule* (MPS) adalah suatu set angka perencanaan yang mengidentifikasi jumlah dari *part number* tertentu yang dapat dan akan dibuat oleh perusahaan manufaktur. Ini adalah jadwal yang membentuk sistem MRP dan ini juga adalah salah satu dari tiga *input* utama bagi MRP *Logic*.

Ketika sudah ditemukan pola permintaan tertentu, maka harus diperhatikan juga permintaan untuk bagian tersebut dan mengimbangnya dengan *supply* bagi barang lainnya. Selain itu juga, *user* akan merencanakan jumlah tertentu yang akan diproduksi pada tanggal tertentu. Jadi, jadwal produksi adalah daftar dari sejumlah tanggal-tanggal tertentu beserta jumlah dan bagian yang akan diproduksi.

Cara untuk mengembangkan MPS adalah: pertama-tama harus memperhatikan *detail* dari MPS, kemudian melihat *detail* dari *capacity plan*. Kemudian akan tercipta jadwal yang dibagi-bagi berdasarkan fase-fase waktu yang dapat menunjukkan keseimbangan permintaan dan *supply* terhadap suatu bagian tertentu.

Ada dua elemen dari permintaan terhadap bagian tertentu yaitu:

- Perencanaan atau perkiraan, umumnya datang dari bagian *marketing* yang memperkirakan jumlah barang yang akan dijual pada suatu periode tertentu.
- Pemesanan *customer* yang sesungguhnya.

Karena jadwal MPS adalah *master* dari semua jadwal dan menentukan MRP, mengakibatkan proses ini adalah yang paling utama dan menentukan

pembuatan keputusan pada keseluruhan sistem MRP. Akibatnya, seseorang harus mengatur MPS dengan menentukan jumlah yang akan di produksi dan waktu yang tepat kapan *stock-stock* tersebut akan selesai diproduksi. Juga sangat penting untuk menyamakan antara jumlah perencanaan produksi per bagian dengan jumlah total yang akan dibuat pada perencanaan produksi.

#### 2.4.2. Data Inventory

Ada dua macam tipe data inventory, yaitu:

- *Dynamic Inventory Data*, adalah semua tipe elemen yang memiliki perubahan yang konstan, seperti jumlah *stock* yang ada saat ini, *scheduled receipt*, dan semua elemen dari *gross to net logic*.
- *Static Inventory Data*, terdiri dari informasi yang berupa identitas seperti *part number* dan deskripsinya.

Proses pencatatan data, baik yang terkomputerisasi maupun yang manual, atau disebut juga sebagai *file*, dan daerah-daerah tertentu pada *record* tersebut yang dibagi berdasarkan kategorinya disebut juga *field*.

#### 2.4.3. Bill of Material

*Bill of Material* secara sederhana adalah daftar dari *item-item* yang dibutuhkan untuk pembuatan suatu barang.

##### **Part Number**

Ada beberapa cara yang berbeda untuk memberikan *part number*, antara lain:

- *Random*, yaitu kode angka yang diberikan hanya berfungsi sebagai pengenalan saja, tidak memberikan deskripsi apapun.
- *Significant*, memberikan informasi yang spesifik mengenai item tersebut seperti misalnya sumber/asal, material, bentuk dan deskripsi lainnya. Kelebihan dari sistem ini adalah dapat membantu seseorang untuk langsung mengenal dan mengetahui *item* tersebut cukup hanya dengan melihat *part number* tersebut. Sedangkan kekurangannya perubahan yang

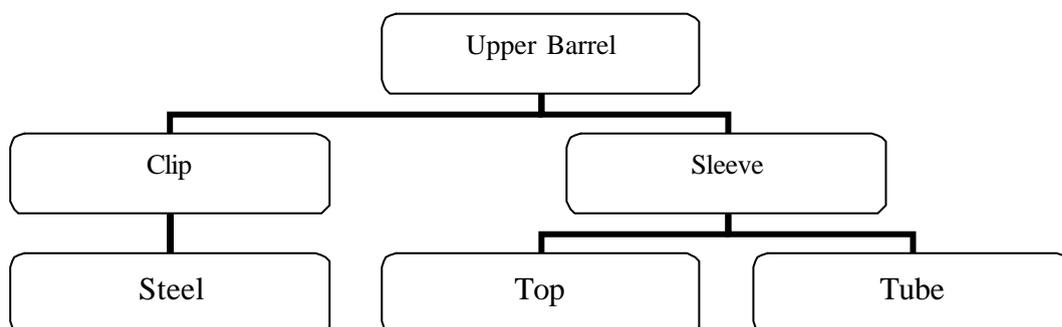
sedikit saja mengenai informasi karakteristik barang tersebut mengharuskan *user* mengubah kode secara keseluruhan.

- *Semisignificant* adalah alternatif lain, yaitu tetap menggunakan bagian pertama dari *part number* untuk memberi informasi barang apa, seperti yang ada pada metode *significant* dan juga menggunakan sistem *random* pada bagian lainnya.

*Part number* pada umumnya digunakan oleh toko untuk memindahkan barang dari proses manufaktur dan juga digunakan oleh komputer dalam pengorganisasian material.

### ***Parent dan Component***

Untuk mengerti konsep dari *Bill of Material* maka perlu diketahui tentang konsep *parent* atau *component relationship*. Komponen adalah objek-objek yang dirakit bersama-sama untuk menciptakan.




---



---

Gambar 2.1. *Parent dan Component*

Ada beberapa elemen data yang penting untuk dapat mendeskripsikan hubungan *parent* atau *component relationship* ini. Yang pertama adalah *parent part number*; kedua *component part number*; dan ketiga adalah jumlah komponen yang dibutuhkan untuk membuat sebuah *parent*.

Salah satu dari banyak hal yang penting untuk diperhatikan ketika mengembangkan suatu sistem MRP adalah jumlah yang dibutuhkan untuk setiap komponen. Jumlah pada *Bill of Material* harus dapat menggambarkan jumlah dari

komponen yang diperlukan untuk melakukan perakitan. Proses MRP akan menggabungkan jumlah per unit ditambah faktor sisa *stock* agar dapat menghitung jumlah kotor kebutuhan.

## 2.5. Proses MRP

Proses MRP terjadi setelah ada permintaan dari konsumen akan sejumlah produk, pesanan atau permintaan ini atau disebut juga *master schedule*, yang selanjutnya akan dicek dengan jumlah barang yang masih tersedia di dalam *inventory*, dan jika perusahaan tidak memiliki barang yang diminta, maka akan dikonsultasikan dengan *bill of material* dan menentukan jumlah dari tiap komponen yang dibutuhkan untuk memproduksi barang tersebut. Keseluruhan langkah-langkah kerja di atas disebut juga *gross requirement* (*Gross requirement* adalah kebutuhan total bahan-bahan yang diperlukan untuk menghasilkan suatu produk).

Proses selanjutnya adalah *Gross requirements* akan mengurangi jumlah barang yang ada di *inventory* saat itu sekaligus juga menjadwalkan berapa jumlah barang yang harus ada di *inventory* pada periode mendatang.

*Net Requirements* adalah jumlah kekurangan *stock* yang harus disediakan untuk memenuhi kebutuhan konsumen, dengan cara memesan tambahan *stock* lagi. Sedangkan jumlah barang yang harus dipesan untuk memenuhi *net requirements* disebut juga *lot sizing*.

Waktu yang dibutuhkan mulai dari pemesanan barang, sampai kepada selesainya barang tersebut di produksi, disebut juga *lead time offset*. *Lead time offset* adalah waktu antara pesanan dilakukan sampai tibanya barang pesanan tersebut ke gudang. Salah satu elemen kunci yang ada pada MRP system adalah: setiap pemesanan yang direncanakan harus di perhitungkan waktunya untuk memenuhi *net requirement*.

Secara ringkas, proses MRP dapat dijelaskan secara sederhana, seperti berikut: dimulai dari *gross requirement*, dikurangi dengan *inventory* yang ada, menghasilkan *net requirement*. Dari *net requirement*, dihitung *lot size* dan *lead time offset*-nya untuk sampai ke *planned order release*.

## 2.6. Output MRP

Keluaran MRP sekaligus juga mencerminkan kemampuan dan ciri dari MRP, yaitu :

- a. *Planned Order Schedule* (Jadwal Pesanan Terencana) adalah penentuan jumlah kebutuhan material serta waktu pemesanannya untuk masa yang akan datang.
- b. *Reschedule Notice* adalah penyesuaian dan penjadwalan ulang untuk pemesanan barang-barang pada waktu tertentu.
- c. *Capacity Requirement Planning* adalah proses untuk menentukan berapa jumlah barang yang bisa diproduksi secara optimal pada suatu waktu tertentu dan berapa jumlah *input* yang dibutuhkan untuk melakukan proses produksi tersebut.

## 2.7. Tipe MRP

Dalam manajemen material dikenal 2 tipe dasar dari sistem MRP, yaitu:

### a. *Regeneration MRP*

*Regeneration MRP* adalah proses mengambil jadwal *master production* dan memecahnya ke dalam *bill of material*. Dari proses ini akan dihasilkan jumlah kotor total kebutuhan, yang akan menghasilkan *planned order releases* dan mungkin juga peringatan untuk melakukan pemesanan kembali. Keuntungan dari sistem ini adalah penggunaan alat pemrosesan data akan lebih efisien, baik untuk digunakan pada suatu lingkungan yang stabil. Kerugiannya adalah tidak terlampau peka terhadap ketidakseimbangan antara kebutuhan dan kemampuan untuk memenuhi kebutuhan tersebut.

### b. *Net Change MRP*.

*Net Change MRP* pada umumnya berkaitan dengan *transaction-triggered MRP*. Perubahan pada permintaan, pemesanan, status *inventory*, *bill of material*, atau yang lain, yang mempengaruhi *item* akan menimbulkan perubahan

komponen-komponen dari *item* tersebut. Konsep ini pada dasarnya adalah merupakan proses penjabaran dari semua *requirement/* kebutuhan untuk memproduksi suatu produk. Proses ini hanya dilakukan apabila terjadi perubahan dalam MPS atau keadaan persediaan atau sistem persediaan untuk semua item. Keuntungan sistem ini adalah akan selalu memberikan catatan yang *up to date*.

## 2.8. Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan sesuatu kegiatan atau untuk menyelesaikan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya dan juga menggambarkan suatu kejadian yang nyata yang dapat juga digunakan untuk mengambil keputusan

Jadi Sistem Informasi adalah suatu kegiatan dari prosedur-prosedur yang diorganisasikan, bilamana dieksekusi akan menyediakan informasi untuk mendukung pengambil keputusan dan pengendalian di dalam organisasi.

## 2.9. Tingkat Kegunaan Sistem Informasi

Berdasarkan tingkat kegunaan sistem informasi dapat dibagi menjadi empat level, yaitu:

### a. Level 1

Sistem Informasi sebagai Sistem Pemrosesan Data (*Data Processing System*). Hal ini berguna untuk memproses sejumlah besar data untuk transaksi bisnis rutin. Terdapat juga data *entry* ke sistem dan laporan transaksi dengan sedikit membutuhkan keputusan.

b. Level 2

Sistem Informasi sebagai Sistem Informasi Manajemen (*Management Information System*), berguna untuk menyusun laporan secara periodik untuk kontrol, perencanaan, dan membuat keputusan. Merupakan interaksi antar orang, software, dan tidak dapat menggantikan sistem pemrosesan data karena merupakan bagiannya. Hal ini tidak sekedar memproses data namun termasuk analisis keputusan dan membuat keputusan. Laporan bersifat umum untuk semua bagian perusahaan.

c. Level 3

Sistem Informasi sebagai Sistem Pendukung Keputusan (*Decision Support System*). Mendukung pengambil keputusan dengan menyusun informasi pada kebutuhan tertentu. Sama dengan MIS keduanya tergantung pada database sebagai sumber data. Program ini dibuat khusus untuk orang tertentu atau grup tertentu pengambil keputusan.

d. Level 4

Sistem Informasi sebagai *Expert System* dan *Artificial Intelligent System*. Menangkap keahlian tiruan dari pembuat keputusan dalam menyelesaikan masalah. *Expert system* menggunakan pendekatan *Artificial Intelligent* yaitu membuat mesin seakan-akan punya kepandaian untuk menyelesaikan masalah secara beralasan sesuai dengan pemikiran manusia.

## 2.10. Teori Singkat DFD (*Data Flow Diagram*)

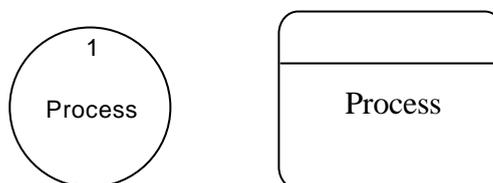
*Data Flow Diagram* atau yang sering disingkat dengan DFD adalah representasi dari sebuah sistem secara grafis yang digambarkan dengan sejumlah simbol tertentu untuk menunjukkan perpindahan data dalam proses-proses suatu sistem.

Dalam hal ini, DFD menunjukkan perpindahan dan perubahan data dalam suatu sistem. Meskipun diberi nama *Data Flow Diagram*, namun

penekanan pada DFD lebih pada prosesnya, bahkan DFD merupakan salah satu alat pemodelan proses dari sistem yang paling sering digunakan.

Simbol-simbol yang digunakan dalam DFD terdiri dari 4 macam, yaitu : *process*, *data flow*, *data store*, dan *external entity*. Berikut uraian singkat mengenai 4 macam simbol di atas :

- a. *Process* adalah simbol yang mengubah suatu data dari suatu bentuk menjadi bentuk yang lain. Dengan kata lain, proses menerima masukan data dan mengeluarkan keluaran data lain yang telah diproses. Simbol dari *process* dalam DFD ada dua macam, yaitu :



Gambar 2.2. Simbol *Process*

- b. *Data flow* atau aliran data adalah aliran yang menunjukkan perpindahan data dari satu bagian ke bagian yang lain dalam suatu sistem. *Data flow* dalam DFD disimpulkan dengan tanda panah dan diberi keterangan di sampingnya yang menunjukkan data apa yang mengalir. Simbolnya sebagai berikut :



Data Flow

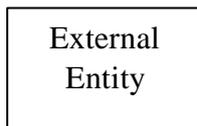
Gambar 2.3. Simbol *Data flow*

- c. *Data store* adalah tempat penyimpanan data dalam suatu sistem, baik secara manual maupun secara elektronik. Simpanan data digunakan jika suatu proses perlu menggunakan lagi data tersebut. Simbol dari *data store* ada dua macam yaitu :



Gambar 2.4. Simbol *Data Store*

- d. *External entity* adalah seseorang, sekelompok orang, sebuah departemen di dalam maupun di luar organisasi, atau sebuah sistem lain yang memberi masukan untuk sistem yang ada atau menerima keluaran dari suatu sistem. *External entity* disebut juga *terminator*, karena merupakan batas dari sebuah sistem. Dalam DFD, *external entity* disimbolkan dengan :



Gambar 2.5. Simbol *External Entity*

### 2.11. Teori Singkat ERD (*Entity Relationship Diagram*)

*Entity Relationship Diagram* (ERD) mendokumentasikan data perusahaan dengan mengidentifikasi tipe dari *data entities* dan *interrelationship* mereka. ERD dipersiapkan pada titik dalam sistem pengembangan proses yang merupakan gambaran dari data secara spesifik. Titik itu bisa berasal dari :

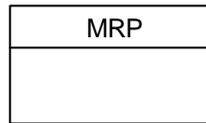
- Ketika eksekutif perusahaan ikut serta dalam *data modeling* untuk *enterprise*, mempertimbangkan kebutuhan data untuk seluruh perusahaan.
- Ketika eksekutif terlibat dalam data modeling untuk sebagian besar dari operasi perusahaan, seperti area bisnis.
- Ketika spesialis informasi dan pemakai terlibat dalam data modeling untuk area aplikasi.

#### 2.11.1. Tipe-tipe Entity

*Entity* bisa berupa :

- a. *Environmental element*
- b. *Resource*
- c. *Transaction*

*Entity* di ERD digambarkan dengan kotak segi empat seperti tampak pada gambar berikut :

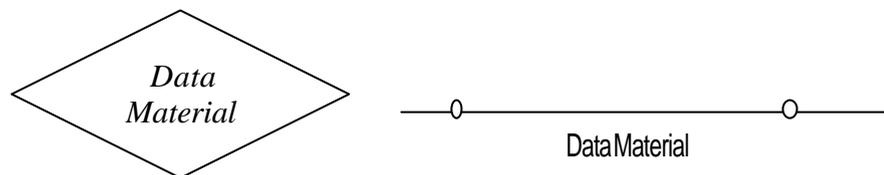


Gambar 2.6. Entity

Tiap kotak diberi label berupa kata benda.

### 2.11.2. Relationships

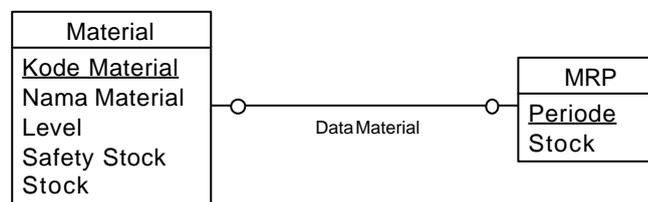
*Relationship* adalah hubungan yang terjadi antar dua *entity*. Hubungan ini digambarkan dengan belah ketupat (Gambar 2.7) dan diberi label berupa kata kerja.



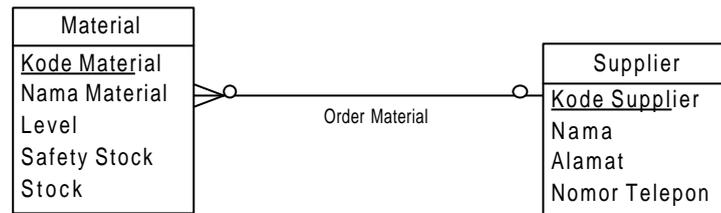
Gambar 2.7. Relationship

### 2.11.3. Konektivitas

Jumlah *relation* yang terjadi antar *entity* disebut konektivitas. Ada 3 macam konektivitas, yaitu *one-to-one*, *one-to-many*, dan *many-to-many*. Umumnya konektivitas menggunakan karakter 1 dan M. Contohnya dapat dilihat dalam gambar berikut :



Gambar 2.8. One to One



Gambar 2.9. One to Many atau Many to One



Gambar 2.10. Many to Many

#### 2.11.4. Identifikasi dan deskripsi dari *entity*

Identifikasi dan deskripsi dari *entity* dijelaskan oleh atribut-atributnya (karakteristik *entity*). Contoh atribut dari *customer* adalah *customer number*, *customer name*, dan *sales teritory*. Atribut yang mengidentifikasi *entity* disebut *identifier* (contoh nomor *customer* tidak boleh ada yang sama), sedangkan atribut lainnya yang mendeskripsikan *entity* disebut *descriptors*. *Identifier* biasanya ditunjukkan dengan garis bawah.



Gambar 2.11. Atribut

#### 2.11.5. Langkah-langkah dalam membuat ERD

##### a. Mengidentifikasi *entities*

Manajemen memutuskan *Environmental element*, *Resource*, *Transaction* yang digambarkan dengan data.

- b. Mengidentifikasi *relationships*  
Di bagian ini mulai digambarkan sketsa simbol dan membuat *relationshipnya* dari kiri ke kanan atau dari atas ke bawah.
- c. Membuat gambaran ERD yang akan dibuat  
Elemen data yang mengidentifikasi dan mendeskripsikan tiap *data entity* didaftar di samping *entity*.
- d. Memetakan elemen data pada *entities*  
Elemen data dipelajari untuk membuat struktur database yang efisien.
- e. Melakukan analisa data  
Elemen data dipelajari untuk membuat struktur database yang efisien. Proses melakukan analisa data disebut normalisasi dan digunakan untuk menyesuaikan data sehingga sesuai dengan urutan *normal forms*.
- f. Membuat ERD yang sudah dimodifikasi  
Hasil dari analisa data tergabung dalam ERD yang baru. Dalam cara ini, tipe *entity* dan *relationshipnya* disusun sedemikian rupa sehingga dapat menyediakan dasar yang paling efisien untuk mendesain *database*.
- g. Review ERD dengan user dan memperbaikinya  
Spesialis informasi me-review diagram dengan eksekutif, manajer, dan nonmanager dalam area user dan memperbaikinya jika perlu.

## 2.12. Teori Singkat *Database*

Manajemen adalah bagian dari IRM (*Information Resource management*) dan menjamin sumber data dari perusahaan mencerminkan sistem fisik yang direpresentasikan.

Sumber data disimpan di *secondary storage* :

- *Sequential access : magnetic tape*
- *Direct access : magnetic disk, optical disk*

Konsep *database* dimulai dari indeks dan *link* untuk memperoleh hubungan logika antara *file*.

### 2.12.3. Konsep *Database*

*Database* adalah kumpulan dari komputer data yang diintegrasikan, diorganisir, dan disimpan dengan cara yang mudah untuk diakses. Untuk ini

digunakan *Direct Access Storage*. Integrasi secara logika dari *record* di beberapa *file* disebut konsep *database*.

Dua tujuan utama konsep *database* :

- Meminimalkan *redundancy*
- Memperoleh data independen

Kemampuan untuk mengubah struktur data tanpa mengubah program untuk memproses data. Hal ini dapat dicapai dengan menempatkan data *specification* pada suatu tabel yang secara fisik terpisah dari programnya. Program akan mereferensi tabel untuk mengakses data, perubahan struktur data hanya dilakukan sekali pada tabel.

### 2.12.2. Relationship Dalam *Database*

Untuk menghubungkan antara *entity* yang satu dengan yang lain dalam *database*, dibutuhkan *entity key*, yaitu satu atau beberapa *atribut* tertentu yang bersifat unik sehingga dapat digunakan untuk membedakan anggota *entity* yang satu dengan yang lainnya pada *entity* yang sama. Pada diagram *entity* juga dibutuhkan *relationship key*, yaitu setiap hubungan yang diperlukan untuk menyatakan hubungan antara *entity key* yang satu dengan yang lain.

Ada beberapa macam hubungan yang terjadi antara dua *entity* :

#### a. *One to One Relationship*

Hubungan antara satu *entity* dengan *entity* yang lain. Hubungan ini dibedakan menjadi dua macam, yaitu *obligatory* dan *non-obligatory*.

*Obligatory* bila semua anggota dari suatu *entity* harus berpartisipasi atau mempunyai hubungan dengan *entity* yang lain, dan *non-obligatory* bila semua anggota *entity* tidak harus mempunyai hubungan dengan anggota *entity* yang lain.

#### b. *One to Many Relationship*

Hubungan dimana satu anggota *entity* mempunyai hubungan dengan beberapa anggota *entity* lain yang berbeda.

#### c. *Many to Many Relationship*

Hubungan dimana beberapa *entity* mempunyai hubungan dengan beberapa anggota *entity* yang lain pada *entity* yang berbeda. Jadi, kedua pihak bisa

memiliki hubungan dengan lebih dari satu anggota *entity* yang lain. Hubungan ini dapat dibedakan menjadi hubungan *obligatory* dan *non-obligatory*.

Dengan menggunakan metode hubungan *entity* dalam perencanaan *database*, dapat dilakukan beberapa langkah untuk membedakan hubungan antara *file* yang ada :

- Menentukan *entity* yang ada dalam *database* dengan masing-masing atribut
- Menentukan semua hubungan yang dapat terjadi antar *entity* yang ada
- Menentukan hubungan yang ada, termasuk *One to One*, *One to Many*, atau *Many to Many* juga menentukan apakah hubungan tersebut *obligatory* atau *non-obligatory*.
- Dari jenis hubungan yang telah ditentukan tersebut, maka ditentukan jumlah relasi yang diperlukan.

Setelah tahap ini selesai dilaksanakan, maka telah tersedia suatu *database* yang telah didesain dengan baik dan siap digunakan dalam kondisi yang seefisien mungkin, dan dapat memberikan informasi secara tepat dan benar.