

ANALISIS PERANCANGAN SISTEM

A. Pengantar

Sistem adalah kumpulan elemen yang masing-masing elemen tersebut memiliki fungsi masing-masing, namun secara bersama-sama bekerja untuk mencapai tujuan dari adanya sistem tersebut. Sebuah mobil dapat dikatakan sebuah sistem karena di dalamnya terdapat kumpulan elemen (seperti kemudi, rem, mesin, roda, kaca spion, lampu sen, dan sebagainya) yang masing-masing elemen tersebut memiliki fungsi masing-masing, namun secara bersama-sama bekerja untuk mencapai tujuan dibuatnya mobil tersebut yaitu sebagai alat transportasi.

Sistem yang kita bahas sebagai contoh sederhana di sini adalah sistem komputerisasi administrasi di suatu unit usaha. Misalkan unit usahanya adalah “Mini Market ‘Rachmana Tamaya’” yang akan kita buat “sistem penjualan barang”-nya.

B. Prosedur Singkat Penjualan Barang di “Rachmana Tamaya”

Mini market “Rachmana Tamaya” tidak melakukan pendataan pelanggan, sehingga pembeli manapun dapat melakukan transaksi pembelian. Namun demikian, untuk meningkatkan pelayanan kepada pembeli, di kasir (hanya ada 1 kasir), pembeli dapat menanyakan apakah barang yang dicari tersedia di sana, masih ada berapa unit, dan berapa harganya.

Setiap barang yang sudah dilakukan transaksi tidak dapat ditukar atau dikembalikan, dan setiap pembelian harus dilakukan secara tunai. Meskipun data pembeli tidak dicatat, namun di setiap struk belanja yang dicetak diberi kode. Kode tersebut terdiri dari masing-masing 2 digit tanggal, bulan, tahun, dan 3 digit nomor urut. Misalkan kode = “120903056” berarti, transaksi yang terjadi pada tanggal 12 bulan September tahun 2003 di nomor urut 056.

C. Analisis (Persiapan) Merancang Sistem

Ada beberapa hal yang harus dianalisis sebelum membuat perancangan sistem, yaitu : (1) ruang lingkup atau batasan sistem, (2) apa yang ingin dihasilkan oleh sistem (tujuan sistem/ output), (3) siapa saja yang terlibat di dalamnya, dan sebagainya.

Ruang lingkup sistem yang akan kita bahas adalah tentang penjualan barang di mini market “Rachmana Tamaya”, tidak membahas pembelian barang untuk keperluan stok di mini market tersebut, maupun bagaimana penanganan barang rusak, kadaluarsa, dan sebagainya. Masih dalam batasan sistem, pihak-pihak/ orang-orang yang berada di “lingkar luar” adalah pengunjung mini market yang selanjutnya disebut dengan “Pembeli” dan “Pemilik,” yaitu pemilik mini market yang harus diberi laporan hasil penjualan barang setiap hari (setelah toko tutup).

Tujuan pembuatan sistem ini adalah untuk mencatat transaksi penjualan barang di mini market tersebut (sehingga dapat digunakan untuk mengecek uang masuk, selanjutnya juga dapat dimanfaatkan untuk menghitung keuntungan unit usaha, dan sebagainya).

Pihak-pihak yang terlibat di dalamnya (pada proses penjualan barang) adalah kasir, dan beberapa penjaga toko yang merangkap pengontrol keberadaan barang di rak-rak pajang).

D. Penggambaran Perancangan Sistem

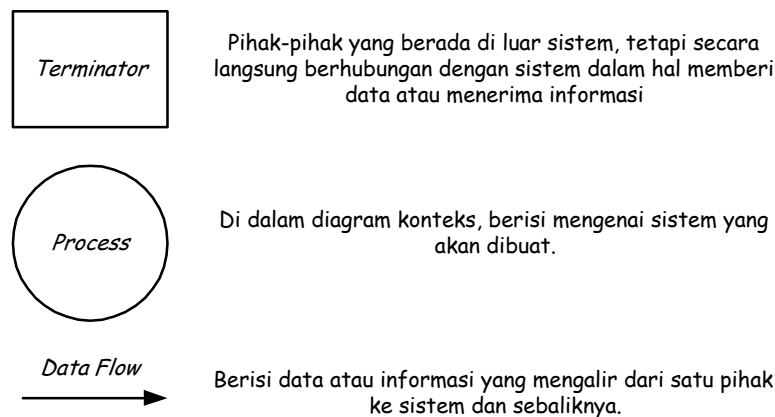
Di bahasan ini, penggambaran perancangan sistem yang digunakan adalah *Data Flow Diagram* (DFD), *Entity/ Relationship Diagram* (E/R Diagram), dan *Data Normalization*. DFD dibagi menjadi tiga strata (tingkatan), yaitu *Context Diagram*, *Zero Diagram*, dan *Detail Diagram*.

D.1. Context Diagram (Diagram konteks)

Diagram konteks berisi gambaran umum (secara garis besar) sistem yang akan dibuat. Secara kalimat, dapat dikatakan bahwa diagram konteks ini berisi “siapa saja yang memberi data (dan data apa saja) ke sistem, serta kepada siapa saja informasi (dan informasi apa saja) yang harus dihasilkan sistem.”

Jadi, yang dibutuhkan adalah (1) **Siapa** saja pihak yang akan memberikan data ke **sistem**, (2) Data **apa** saja yang diberikannya ke sistem, (3) kepada **siapa** sistem harus memberi informasi atau laporan, dan (4) **apa** saja isi/ jenis laporan yang harus dihasilkan **sistem**.

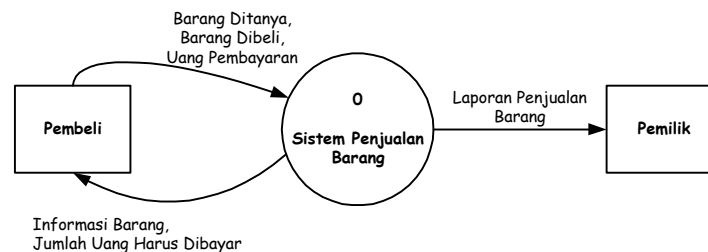
Kata “**Siapa**” di atas dilambangkan dengan kotak persegi (disebut dengan *terminator*), dan kata “**apa**” di atas dilambangkan dengan aliran data (disebut dengan *data flow*), dan kata “sistem” dilambangkan dengan lingkaran (disebut dengan *process*).



Gambar 1. Lambang-lambang yang akan Digunakan di Diagram Konteks

Beberapa kemungkinan (data) yang diberikan pembeli kepada kasir adalah : (1) barang yang ditanyakan, (2) barang yang akan dibeli, dan (3) Uang pembayaran. Sebaliknya, kemungkinan informasi yang diberikan kasir kepada pembeli adalah (1) keadaan barang yang ditanyakan, (2) jumlah uang yang harus dibayar.

Sedangkan informasi yang diberikan kasir kepada Pemilik adalah Laporan Jumlah Uang Masuk beserta Jumlah Barang yang Terjualnya. DFD Konteksnya :



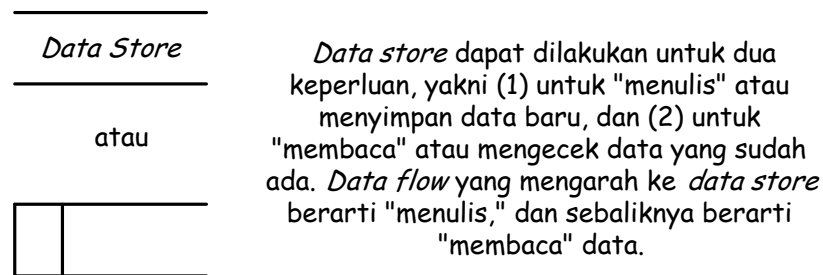
Gambar 2. Contoh Diagram Konteks

D.2. Zero Diagram (Diagram Nol).

Tujuan dari diagram nol adalah untuk “memerinci” sebuah sistem menjadi “proses-proses” yang harus dilakukan ‘orang dalam.’ Atau jika dibuat dalam kalimat adalah : “Apa saja proses yang harus dilakukan agar mencapai sistem tersebut ?.”

Jadi, diagram ini adalah kelanjutan dari diagram konteks, yang “memperbanyak lingkaran,” sedangkan untuk (jumlah dan isi) *terminator* serta (jumlah dan isi) *data flow* dari dan ke terminator tersebut **harus tetap**.

Pada diagram ini pula mulai ditampilkan *data store* (penyimpan data/ *file*) yang dibutuhkan.



Gambar 3. Lambang Penyimpan Data

File apa saja yang dibutuhkan di sistem ini ?. Jenis *file* data ada dua, yakni (1) *master file*, dan (2) *transaction file*. *Master file* adalah *file* berisi (mencatat) mengenai objek yang harus ada di sebuah unit usaha yang jika tidak ada objek tersebut maka unit usaha tersebut tidak akan berjalan secara sempurna.

Objek-objek yang harus ada di sebuah *mini market* adalah : (1) barang, (2) pengelola, (3) pembeli, dan (4) fasilitas. Jika salah satu objek tersebut tidak ada, maka *mini market* tidak akan berjalan sempurna. Dalam pencatatan penjualan barang, *master file* yang dibutuhkan hanya barang, pengelola (kasir), dan fasilitas (Rak, yang digunakan untuk memajang barang). *File* Pembeli tidak diperlukan karena data pembeli tidak dicatat.

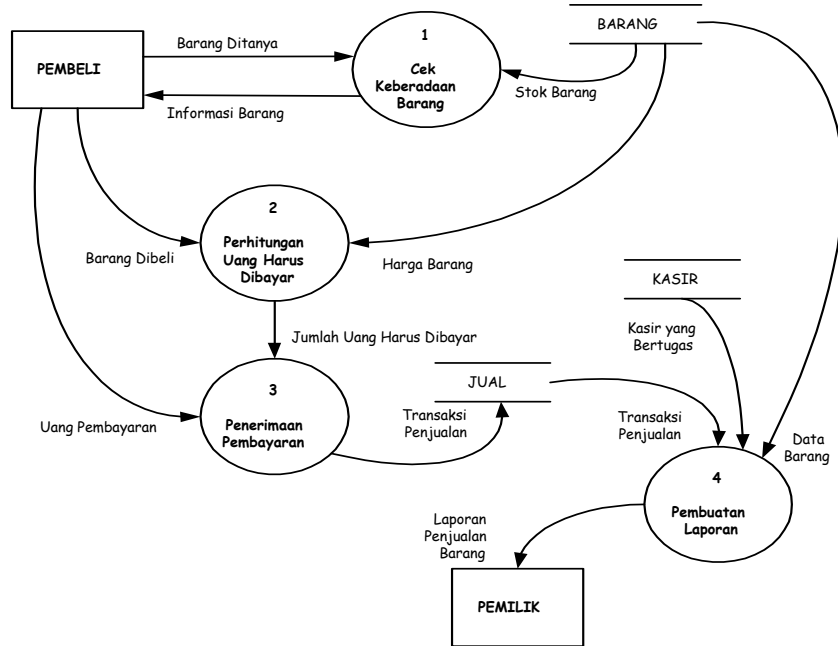
Sedangkan *transaction file* adalah *file* yang digunakan untuk mencatat transaksi yang terjadi di *mini market* tersebut. Transaksi adalah berelasinya (berhubungannya) dua *master file* (atau lebih). Jika kita lihat apa saja *master file* yang harus ada di atas, dan mana yang berhubungan sehingga terjadi transaksi, maka kemungkinan-kemungkinan transaksi yang terjadi di *mini market* tersebut adalah :

1. “Pembeli **membeli** barang”
2. “Kasir **menjual** barang”
3. “Pembeli **membeli** barang, dan Kasir **Menjual** Barang” (gabungan 1 dan 2)

Karena Pembeli tidak dicatat, maka kita menggunakan transaksi yang ke 2 saja, sehingga transaksi yang terjadi akan dicatat di *file* “JUAL”. (Nama *file* terserah perancang sistem)

Mari kita susun diagram nol-nya.

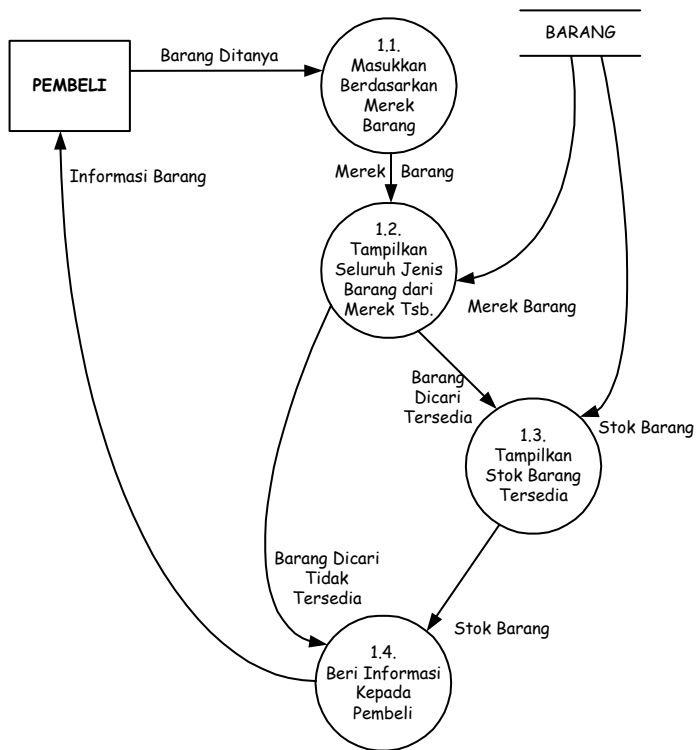
Ketika “Pembeli” datang, ada 2 kemungkinan yang akan dilakukannya, yaitu (1) bertanya keberadaan barang yang akan dibelinya, dan (2) ia mengambil barang-barang yang akan dibelinya dan menyerahkan kepada kasir untuk dihitung berapa yang harus dibayarnya. Apa aksi (proses yang harus dilakukan kasir) untuk kedua kemungkinan tersebut ?.



Gambar 4. Diagram Nol

D.3. Detail Diagram (Diagram Detil).

Diagram detil adalah diagram yang memungkinkan proses yang ada di diagram nol lebih diperinci lagi. Misalkan untuk proses 1 di atas.



Gambar 5. Diagram Detil Proses 1.

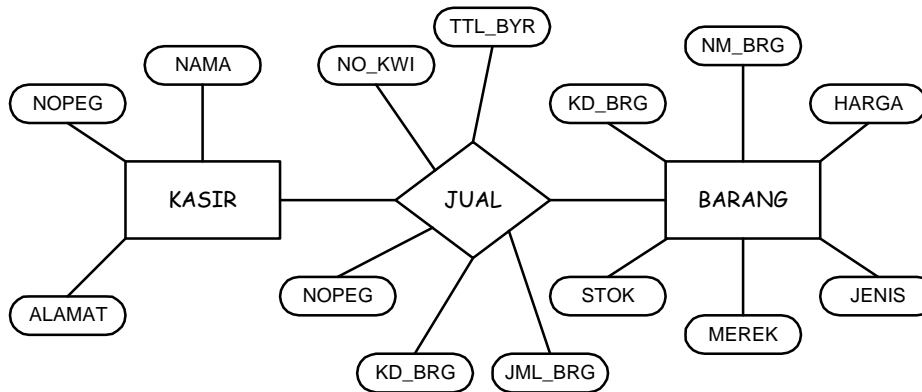
D.4. Entity/ Relationship Diagram (Diagram E/R)

Diagram E/R digunakan untuk memperlihatkan hubungan antarfile (*data store*) yang ada di DFD. Di sini, *file (data store)* disebut dengan *entity* (entitas). Bagian dari *file*, yaitu *fields*, di sini disebut dengan *attributes* (atribut-atribut). Berikut lambang-lambanganya.



Gambar 6. Lambang-lambang Diagram E/ R

Berikut contoh diagram E/R dalam kasus ini :



Gambar 7. Diagram E/R Kasus di Atas.

Berikut penjelasan atribut-atribut yang digunakan :

Nama File (Entitas)	Nama Field (Atribut)	Keterangan
KASIR	NOPEG	Nomor Pegawai
	NAMA	Nama Pegawai
	ALAMAT	Alamat Pegawai
BARANG	KD_BRG	Kode Barang
	NM_BRG	Nama Barang
	HARGA	Harga Satuan Barang
	STOK	Jumlah Barang Tersedia
	MEREK	Merek Barang
	JENIS	Jenis Barang

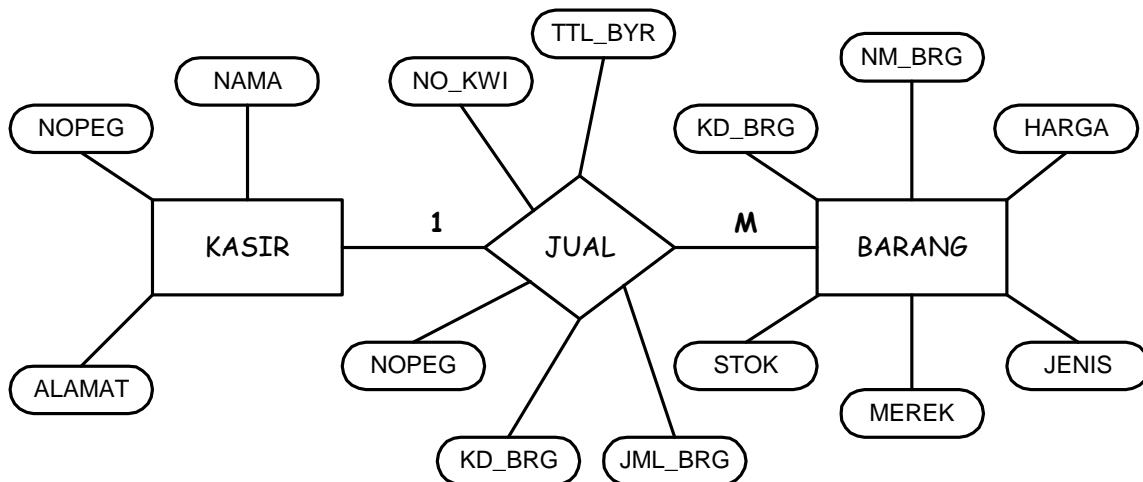
Nama <i>File</i> (Entitas)	Nama <i>Field</i> (Atribut)	Keterangan
JUAL	NOPEG	Nomor Pegawai
	KD_BRG	Kode Barang
	NO_KWI	Nomor Kwitansi
	JML_BRG	Jumlah barang yang dibeli per kode barang
	TTL_BYR	Jumlah uang yang dibayar

Atribut-atribut ini disesuaikan dengan kebutuhan oleh perancang sistemnya, misalkan, KASIR boleh saja memiliki atribut tempat dan tanggal lahir “TT_Lahir,” tetapi karena atribut itu tidak diperlukan maka tidak perlu dibuat/ ditulis. Tetapi harus dipatuhi bahwa setiap atribut yang ada di entitas adalah memang merupakan atribut (identitas) dari entitasnya. Jadi, jangan masukkan atribut “NM_BRG” di atribut KASIR, karena nama barang bukanlah atribut si KASIR.

Entitas “JUAL” adalah entitas yang merupakan *file* transaksi, jadi, (atribut-atribut) yang tercantum di sana adalah bagian-bagian transaksi yang harus dicatat. Adapun atribut NOPEG dan KD_BRG adalah atribut dari entitas lain yang menjadi “jembatan” untuk mengambil atribut-atribut dari *master file*-nya. (Penjelasan ini ada di bagian berikutnya).

Ada kekuatan hubungan di dalam Diagram E/R yang dinamakan dengan derajat kardinalitas (*cardinality degree*). Ada empat jenis derajat kardinalitas yaitu (1) *one to one* (dilambangkan dengan 1 : 1), (2) *one to many* (dilambangkan dengan 1 : M), (3) *many to one* (dilambangkan dengan M : 1), dan (4) *many to many* (yang dilambangkan dengan M : M). Untuk menetapkan derajat kardinalitas di atas, ikuti kalimat-kalimat berikut ini :

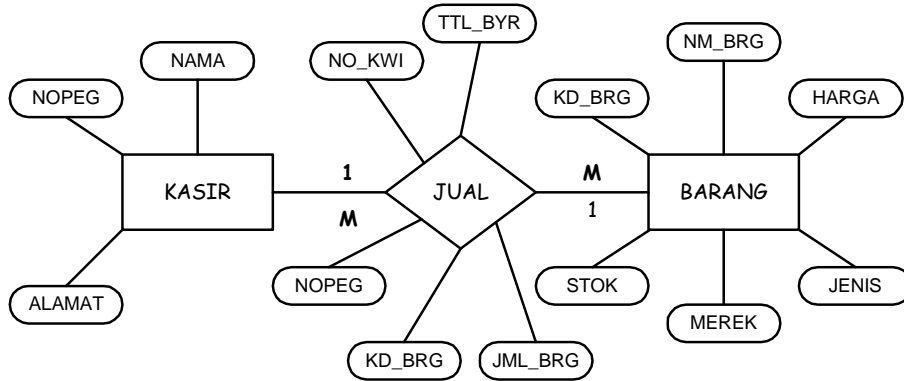
“**Satu** KASIR bisa men**JUAL satu atau lebih** BARANG.” Satu atau lebih = *Many*.



Gambar 8. Proses Pertama Penderajatan Kardinalitas Diagram E/R

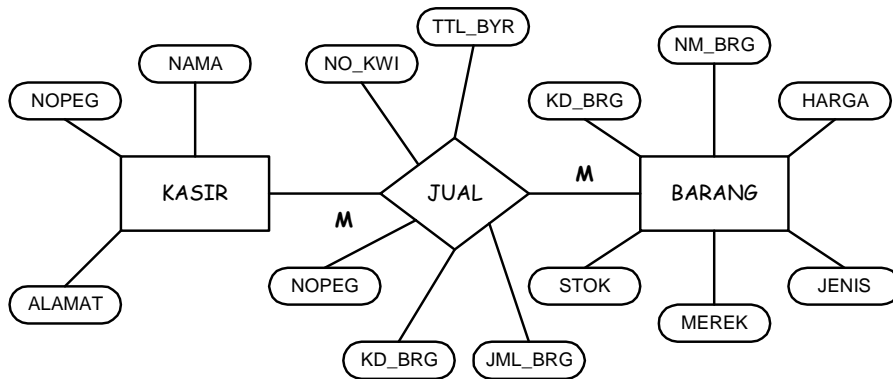
Selanjutnya, kalimat yang dibalik :

“**Satu** (kode) BARANG bisa di**JUAL oleh satu atau lebih** KASIR”



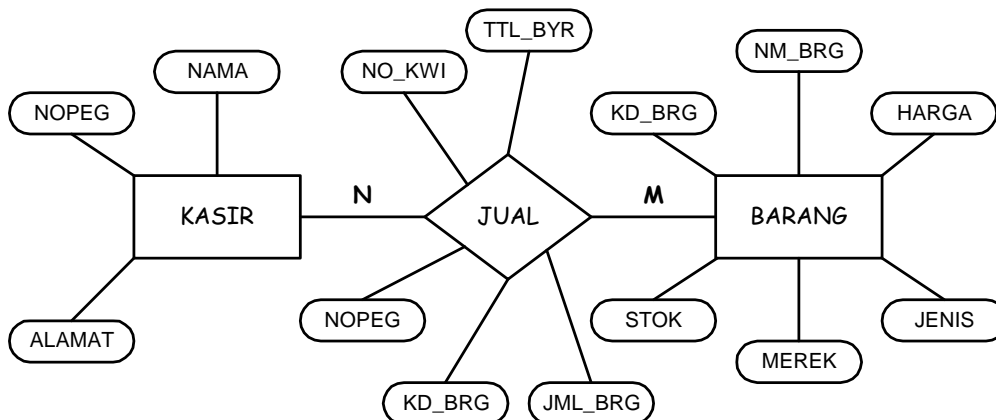
Gambar 9. Proses Kedua Penderajatan Kardinalitas Diagram E/R

Selanjutnya, pilih yang terbesar dari masing-masing sisi :



Gambar 10. Proses Ketiga Penderajatan Kardinalitas

Kini kita dapatkan “M” (*many*) di kedua sisinya yang berarti derajat kardinalitas relasi tersebut adalah “*many to many*.” Namun, karena dalam matematika, nilai M akan selalu sama dengan M, sedangkan belum tentu kalimat (jika nilai M di atas = 10), “Sepuluh KASIR akan selalu menJUAL sepuluh BARANG,” maka penulisan M di salah satu sisinya diganti dengan N, tetapi pembacaannya tetap *many*. Jadi, nilai M dan N bisa jadi sama, dan bisa jadi tidak sama ($M = N$ atau $M \neq N$).



Gambar 11. Proses Akhir Penderajatan Kardinalitas Diagram E/R

D.5. NORMALISASI DATA TINGKAT PERTAMA

Normalisasi data adalah salah satu cara membentuk sebuah *file* yang efektif dan efisien, sehingga dapat memanfaatkan *space* memori komputer seoptimal mungkin. Proses normalisasi data juga bertingkat-tingkat, dan di sini hanya akan dibahas mulai tingkat 1 hingga tingkat 3 saja.

Normalisasi data tingkat pertama (*First Normal Form/ 1NF*) adalah proses penganalisan setiap atribut yang ada di semua *file* yang terbentuk hingga Diagram E/R. Penganalisan ditujukan agar setiap atribut yang dibentuk dapat diolah untuk menghasilkan informasi yang dibutuhkan.

Syarat 1NF adalah “setiap atribut harus bersifat atomik,” artinya, setiap atribut merupakan unsur terkecil dari identitas entitas (tidak perlu dipecah-pecah lagi). Contoh : akan kita periksa atribut “NAMA,” apakah atribut tersebut sudah atomik ?. Misalkan salah satu isi atribut NAMA adalah “Rachmi Hidayat,” jika nama itu selamanya akan digunakan demikian, maka atribut tersebut sudah atomik. Tetapi, jika nama itu suatu saat harus dicetak menjadi “Hidayat, Rachmi,” maka atribut NAMA tersebut belum atomik. Bagaimana mencetak “Rachmi Hidayat” menjadi “Hidayat Rachmi” ?.

Jadi, jika di “negara barat,” biasanya atribut nama dipecah-pecah menjadi *first name* (FNAME), *middle name* (MNAME), dan *last name* (LNAME), karena memang nama mereka sering dibolak-balik.

Periksa juga atribut-atribut lainnya, seperti atribut ALAMAT, perlu tidak dipecah-pecah. Bayangkan saja, jika panjang atribut ALAMAT 100 karakter, kemudian kita diminta mengirim surat, apa jadinya ketika di amplop kita *print* alamat yang panjangnya 100 karakter ?.

D.6. NORMALISASI DATA TINGKAT KEDUA

Normalisasi data tingkat kedua (*second normal form/ 2NF*) bersyarat : (1) telah memenuhi 1NF, (2) setiap atribut *non key* harus tergantung secara fungsional dengan atribut *key*-nya. Apa itu atribut *key* dan *non key* ?.

Key Field (Kunci Atribut)

Kunci atribut adalah atribut yang dipilih untuk dapat mewakili atribut-atribut lain dalam sebuah *record* yang membedakan dengan *record-record* lainnya. Misalkan, untuk seorang mahasiswa, apa atribut yang bisa mewakili dirinya yang bisa membedakan dengan mahasiswa lainnya ?.

Jika dipilih nama, banyak nama mahasiswa lain yang sama. Jika dipilih tanggal lahir, kemungkinan akan ada yang sama, dan sebagainya. Bagaimana cara menentukannya ?.

Super Key (Kunci Super)

Kunci super adalah satu atau beberapa kombinasi atribut yang mungkin dapat dipilih menjadi kunci atribut. Misalkan, *File* “Mahasiswa” memiliki atribut-atribut : NPM, NAMA, KELAS, ALAMAT, dan TGL_LAHIR, maka atribut(-atribut) yang bisa dipilih menjadi kunci atribut adalah :

- (1) NPM
- (2) NAMA (dengan syarat tidak ada nama mahasiswa yang sama)
- (3) NPM + NAMA
- (4) NPM + TGL_LAHIR
- (5) NPM + NAMA + TGL_LAHIR
- (6) Dan seterusnya, berbagai kombinasi yang mungkin

Candidate Key (Kunci Calon)

Kunci calon adalah kunci atribut yang merupakan (kumpulan) kunci atribut yang jumlahnya paling sedikit di kunci super. Kita dapatkan dua buah atribut yang merupakan kunci atribut dengan jumlah atributnya tersedikit (1 atribut), yaitu :

- (1) NPM
- (2) NAMA (dengan syarat tidak ada nama mahasiswa yang sama)

Primary Key (Kunci Utama), yang sering disebut dengan *key field*.

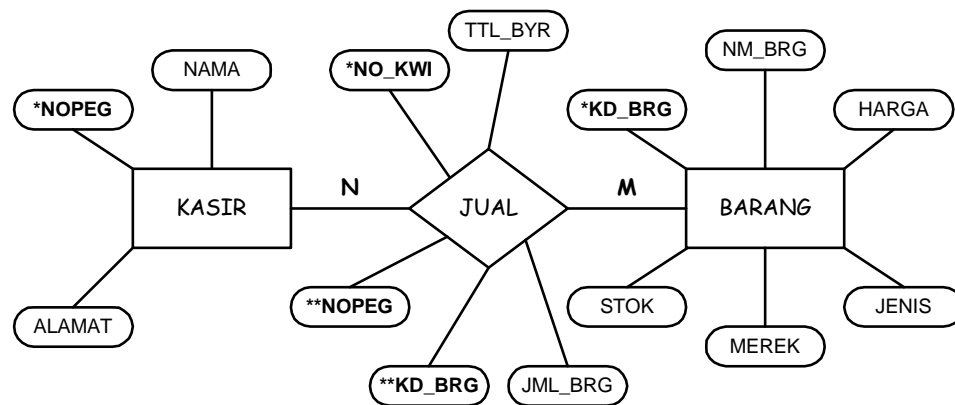
Adalah kunci kandidat yang dipilih untuk dijadikan *key field*. Pemilihan dilakukan dengan mempertimbangkan kemungkinan yang tidak akan pernah sama, maka kunci atribut yang dipilih adalah NPM.

Alternate Key (Kunci Alternatif)

Kunci alternatif adalah kunci kandidat yang tidak terpilih menjadi *primary key*, dalam hal ini, NAMA.

Foreign Key (Kunci Tamu)

Adalah kunci utama dari *file (master)* lain yang digunakan di *file (transaksi)*. Kunci atribut tersebut digunakan sebagai “jembatan” untuk mengambil nilai data dari atribut-atribut lain. Perhatikan Diagram E/R berikut ini (dari kasus sebelumnya) :



Gambar 12. Penulisan Kunci Atribut di Diagram E/R

Di *master file (entity)* KASIR, kunci utamanya : NOPEG (nomor pegawai)

Di *master file (entity)* BARANG, kunci utamanya : KD_BRG (kode barang)

Di *transaction file (relationship)* JUAL, kunci utamanya : NO_KWI (nomor kwitansi)
kunci tamunya : NOPEG dan KD_BRG

NOPEG sebagai kunci tamu di JUAL digunakan untuk menganbil nilai data atribut NAMA dan ALAMAT di *file* KASIR. KD_BRG sebagai kunci tamu di JUAL digunakan untuk mengambil nilai data NM_BRG, HARGA, dan berbagai atribut di *file* BARANG, sehingga bisa diketahui barang tertentu dijual oleh kasir yang mana.

Kembali ke 2NF, kata “tergantungan secara fungsional” menurut pengertiannya adalah (untuk contoh kasus *file* KASIR yang kunci atributnya NOPEG) : “jika NOPEG berubah,

maka harus pasti orangnya (NAMA, dan ALAMAT si Kasir) berubah pula.” Meski demikian, bisa saja alamatnya tidak berubah karena dua orang kasir tinggal di alamat yang sama, itu tidak masalah.

Kasus di atas sudah memenuhi syarat 2NF. Contoh jika belum memenuhi syarat 2 NF adalah jika pada *file* BARANG terdapat atribut NAMA (kasir), atau TGL_BYR (tanggal bayar di JUAL), dan berbagi contoh lainnya yang pada satu *file* terdapat atribut yang tidak semestinya dapat dijadikan atribut *file* tersebut.

D.6. NORMALISASI DATA TINGKAT KETIGA

Normalisasi data tingkat ketiga (*third normal form/ 3NF*) bersyarat : (1) telah memenuhi 2NF, (2) setiap atribut *non key* tidak boleh tergantung dengan atribut *non key* lainnya (tidak boleh terjadi ketergantungan transitif).

Contoh transitif : Jika $A \rightarrow B$, dan $B \rightarrow C$, maka sudah pasti $A \rightarrow C$.

Kebetulan, di kasus di atas tidak ada atribut yang bersifat transitif.

Contoh atribut yang bersifat transitif. Jika $A = \text{NPM}$, $B = \text{KODE_POS}$, dan $C = \text{KOTA}$ yang misalnya ada di *file* MAHASISWA, maka *file* tersebut harus dipecah menjadi, *file* 1 (MAHASISWA) berisi atribut NPM, dan KODE_POS, dan *file* 2 (KODEPOS) berisi atribut KODE_POS dan KOTA.

E. PEMROGRAMAN

Setelah perancangan sistem dibuat (oleh *system analys*), maka kini giliran *programmer* membuat programnya. *File data* yang harus dibuat sesuai dengan *data store* yang tercipta di DFD, atau entitas di ERD, dan atribut-atributnya mengikuti hasil final proses normalisasi data.

Program yang dibuat mengikuti langkah-langkah proses yang ada di DFD *zero* dan detil, selain itu ditambah lagi dengan program umum untuk *filing*, yaitu Input (memasukkan data, menyisipkan data, dan menambah data), Proses (menghapus, mengganti, mengolah, dsb.), Output (membuat laporan/ sesuai DFD, menampilkan data, dsb.).

Selain perancangan sistem di atas, *programmer* perlu memanfaatkan alat-alat bantu lain, seperti kamus data (*data dictionary*) yang berfungsi untuk mendefinisikan setiap elemen data, sehingga dapat mencegah data yang salah yang dimasukkan ke komputer. Misalkan, program harus mengecek apakah isian tentang NPM sudah benar, jangan sampai *user* mengetik misalkan NPM dengan awalan karakter ‘9’ karena NPM di Gunadarma maksimal karakter ‘5’, dan sebagainya.

Perlu juga HIPO *chart*, *Flowchart*, dan sebagainya untuk menyusun algoritma dan logika pemrograman. Perlu juga melakukan rancangan bentuk input (*input design*), dan output (*output design*) agar tampilan yang dihasilkan program tampak indah, menarik, dan komunikatif.

F. KESIMPULAN

Merancang suatu sistem administrasi komputer adalah pekerjaan yang tidak ringan yang biasanya dilakukan secara berkelompok, sehingga tidak boleh main-main dan diperlukan kedisiplinan waktu dan konsentrasi. Banyak *software house* kini mengambil bisnis ini, dan bagi mereka yang belum berpengalaman, kesulitan utama yang muncul adalah (1) koordinasi anggota, (2) estimasi waktu pengerjaan proyek, dan (3) estimasi biaya.