

BAB II

KAJIAN LITERATUR

2.1 Hotel

Hotel merupakan bangunan yang dikelola secara komersial yang menyediakan jasa pelayanan penginapan makanan dan minuman serta jasa lainnya untuk seseorang yang melakukan perjalanan. Hotel biasanya menyediakan fasilitas yang dikelola secara profesional untuk mendapatkan keuntungan dan membuat tamu merasa nyaman, beberapa fasilitas yang disediakan hotel adalah fasilitas olahraga, restoran, *business center*, pelayanan yang ramah dan berbagai fasilitas lainnya. [5][6]

Fungsi hotel telah mengalami perubahan seiring dengan berubahnya gaya hidup, hotel tidak lagi sekedar tempat penginapan, tetapi juga menjadi tempat untuk rapat, pertemuan, resepsi pernikahan dan juga pertunjukkan.

Hotel dapat dibedakan berdasarkan lokasinya sebagai berikut: [7]

1. *City Hotel*, merupakan hotel yang didirikan di daerah perkotaan dan dilengkapi fasilitas bisnis seperti ruang rapat. Jenis hotel ini ditujukan untuk tamu yang menginap dalam waktu singkat.
2. *Residential Hotel*, merupakan hotel yang berlokasi di tepi kota yang memiliki akses dengan mudah untuk menuju tempat kegiatan bisnis dan memiliki fasilitas untuk keluarga. Jenis hotel ini biasanya ditujukan untuk tamu yang menginap dalam jangka waktu lebih lama.
3. *Resort Hotel*, merupakan hotel yang didirikan di daerah pegunungan atau di tepi pantai dan danau. Jenis hotel ini biasanya dijadikan tempat berlibur.
4. *Motor Hotel*, hotel yang juga disebut motel merupakan hotel yang didirikan di pinggiran sepanjang jalan raya yang menjadi penghubung antar kota. Jenis hotel ini biasanya terletak di dekat pintu gerbang perbatasan dua kota.

Hotel juga dapat dibedakan berdasarkan kelasnya menjadi kelas melati, bintang satu, dua, tiga, empat, dan lima. Sedangkan berdasarkan tarif kamar hotel dapat di bedakan menjadi *economy class hotel*, *first class*, dan *deluxe* ataupun *luxury hotel*. [5]

2.2 Hotel Agregator

Hotel Agregator adalah perusahaan yang menyewakan hotel dan kemudian mengambil alih fungsi operasi dan pemasaran hotel tersebut. Agregator hotel pada dasarnya adalah konsolidator hotel dimana pihak agregator dapat mengambil alih sebagian maupun seluruh

inventaris hotel dan dapat menjualnya dengan merek lain. Agregator hotel bahkan dapat mengambil kamar dan menjualnya secara terpisah untuk platform yang berbeda. Setelah menyetujui struktur perjanjian, agregator akan mengurus rebranding atau ketentuan perjanjian. Agregator dapat memengaruhi berbagai elemen hotel. Beberapa area untuk peningkatan dapat menerapkan sistem manajemen properti atau mesin pemesanan hotel untuk membantu mendorong pemesanan langsung. Area lain yang dapat ditangani agregator adalah peningkatan operasional, seperti merenovasi hotel atau meningkatkan keterampilan karyawan. [8]

Berikut dibawah ini beberapa manfaat yang dapat diperoleh dari hotel agregator adalah :[8]

1. Pihak hotel tidak perlu khawatir tentang pemasaran

Pemasaran dan kesadaran terhadap reputasi merek adalah dua perhatian utama para pelaku bisnis perhotelan, dan dengan model agregasi, aspek bisnis ini tidak lagi menjadi perhatian. Agregator melakukan ini untuk hotel, seringkali mendaftar di OTA atau portal mereka sendiri untuk membuat reservasi. Karena merupakan kepentingan agregator untuk menciptakan permintaan dan menghasilkan prospek serta bagian penting dari pekerjaan agregator adalah mengelola reputasi merek, yang merupakan berita bagus bagi para pelaku bisnis perhotelan karena pihak hotel tidak perlu lagi khawatir tentang pemasaran dan manajemen reputasi.

2. Meningkatkan pengalaman karyawan

Karyawan mungkin menemukan bahwa bekerja di lingkungan yang lebih kolaboratif juga menawarkan keuntungan tertentu. Karena tingkat layanan harus setara dengan semua hotel lain yang mereka kelola, agregator akan memastikan bahwa staf terlatih dengan baik dan memiliki kesempatan untuk belajar dan berkembang dalam bisnis, dan pada saat yang sama mendapatkan gaji yang lebih tinggi.

3. Standarisasi level layanan

Perusahaan agregat bekerja untuk menstandarkan tingkat layanan dan sekarang lebih banyak karyawan yang berkualitas akan memainkan peran kunci dalam memastikan kualitas tersebut. Protokol yang ketat akan sering diterapkan untuk memastikan kelancaran operasi. Ini sangat berkontribusi pada pengalaman tamu, yang secara positif akan memengaruhi pendapat mereka tentang hotel.

2.3 Online Travel Agent

Agen perjalanan *online* (OTA) merupakan halaman situs web yang berfungsi sebagai mesin pencari untuk aktivitas perjalanan. Agen perjalanan *online* menghubungkan penyedia layanan di industri perjalanan untuk memberikan kemudahan bagi wisatawan dalam merencanakan perjalanan mereka. Di situs web OTA, pelancong sering dapat mengakses penawaran paket akomodasi, tiket pesawat, kapal pesiar, penyewaan kendaraan, dan lainnya. [9]

Agen perjalanan *online* (OTA) telah berkembang menjadi wadah pemasaran yang canggih untuk properti dengan berbagai ukuran dan jenis. Dengan adanya OTA, konsumen dapat dengan mudah membandingkan opsi perjalanan secara berdampingan dengan harga, tanggal perjalanan, dan lokasi yang berbeda. Studi Cornell menemukan bahwa 93% konsumen beralih ke OTA sebelum membuat keputusan akhir. Oleh karena itu, penting bagi bisnis untuk bekerja di sebanyak mungkin saluran guna meningkatkan kesadaran merek dan mendorong pemesanan bahkan lebih banyak pemesanan langsung. Dengan semakin populernya OTA di seluruh dunia, para pelaku bisnis perhotelan kini dapat mengakses pasar yang sebelumnya tidak dapat diakses. [10]

Terdapat beberapa model bisnis OTA yaitu sebagai berikut : [9]

1. Model Merchant

Pada model ini, agen perjalanan *online* bertindak sebagai pedagang tercatat (*merchant of record*) dan memungut pembayaran dari tamu pada saat dilakukannya pemesanan. Setelah tamu melakukan *check out*, OTA membayar kepada pihak hotel. Terdapat perjanjian antara hotel dan OTA untuk menawarkan sejumlah kamar tertentu kepada OTA dengan tarif harga yang menguntungkan. OTA kemudian menerima keuntungan dari setiap kamar yang terjual, namun harus memenuhi ambang batas perjanjian tersebut.

2. Model Agensi (dikenakan biaya)

Dalam model ini, konsumen memesan melalui OTA tetapi membayar langsung di kasir hotel pada saat *check out*. Hotel kemudian membayar komisi OTA pada nilai total reservasi setelah *check out*. Tidak ada kontrak dan harga yang ditentukan oleh hotel.

3. Model Iklan

Popularitas hotel dapat semakin meningkat apabila hotel tersebut muncul di situs *metasearch* seperti Google Hotel Ads, Tripadvisor, Trivago, dan KAYAK. Mesin *metasearch* beroperasi terutama berdasarkan biaya per klik, yang memungkinkan hotel untuk mengiklankan kamar mereka dengan tautan ke mesin pemesanan langsung

mereka dan membayar berdasarkan jumlah klik yang diterima. Beberapa situs ini, seperti Google, juga menawarkan model biaya per akuisisi alias "bayar per masa inap" di mana properti membayar persentase dari total biaya reservasi di akhir masa inap.

2.4 Asset Light Hotel Chain

Asset Light Hotel Chain merupakan model bisnis dimana perusahaan hotel tidak memiliki kepemilikan atau kendali atas properti fisik seperti tanah bangunan tempat hotel beroperasi, tetapi menjalin kerjasama dengan pihak lain untuk memanfaatkan properti tersebut. Melalui model bisnis ini, perusahaan dapat lebih fokus pada pengembangan merek dan strategi pemasaran untuk meningkatkan pendapatan dan profitabilitas. [11]

Berikut manfaat dari penerapan model bisnis *asset light hotel chain*: [11]

1. Lebih efisien dalam pengelolaan modal.

Dalam model bisnis *Asset Light Hotel Chain*, perusahaan tidak perlu mengeluarkan modal besar untuk membeli atau membangun properti baru, karena properti yang digunakan dimiliki oleh pihak ketiga. Hal ini memungkinkan perusahaan untuk mengelola modal dengan lebih efisien dan memfokuskan sumber daya pada pengembangan merek dan operasional hotel.

2. Lebih fleksibel dalam operasional.

Dalam model bisnis *Asset Light Hotel Chain*, perusahaan dapat lebih fleksibel dalam mengelola operasional hotel dan merek, karena tidak terikat pada properti tertentu. Hal ini memungkinkan perusahaan untuk menyesuaikan diri dengan kondisi pasar yang berubah dan memperluas jangkauan merek mereka dengan lebih mudah.

3. Dapat memperluas jangkauan merek dengan lebih efisien.

Dalam model bisnis *Asset Light Hotel Chain*, perusahaan dapat memperluas jangkauan merek mereka dengan lebih efisien dan dengan biaya yang lebih rendah, karena tidak perlu mengeluarkan modal besar untuk investasi dalam properti baru. Hal ini memungkinkan perusahaan untuk mencapai pertumbuhan yang lebih cepat dan memperluas jangkauan merek mereka dengan lebih efektif.

2.5 Pemesanan Hotel

Pemesanan pada hotel adalah suatu permintaan untuk mendapatkan kamar yang dilakukan pada waktu sebelumnya melalui berbagai sumber dengan menggunakan berbagai cara pemesanan untuk memastikan tamu akan memperoleh kamar tersebut pada waktu kedatangannya. Beberapa informasi dari semua pemesanan kamar hotel harus dicatat ke dalam

formulir reservasi kamar. Keuntungan formulir reservasi kamar adalah sebagai gambaran dan memastikan semua informasi yang diperlukan sudah diperoleh, mengingatkan tamu tentang berbagai hal seperti harga kamar, pembayaran, waktu kedatangan dan mencegah hilangnya data tamu yang telah diperoleh.

Berikut adalah fungsi pemesanan yaitu:

1. Memberikan citra yang baik dan memotivasi tamu untuk datang ke hotel.
2. Calon tamu hotel dapat memperkirakan biaya yang akan dikeluarkan untuk akomodasi perjalanan.
3. Hotel mempunyai persiapan yang matang dalam menyajikan pelayanan kepada tamu, misalnya penjemputan atau penyiapan permintaan khusus dan lainnya.
4. Petugas hotel dapat menjual fasilitas hotel seperti restoran, *banquet* dan fasilitas lain ketika tamu memesan kamar. [12]

2.6 Pengadaan

Pengadaan merupakan sebuah kegiatan untuk mendapatkan barang dan jasa yang berhubungan dalam aktivitas menentukan jenis produk yang dibutuhkan, melakukan pembelian, menerima dan menyimpan pengiriman dan administrasi kontrak pembelian [13]. Pengadaan barang dan jasa di suatu perusahaan merupakan kegiatan rutin yang sering dilakukan untuk memenuhi kebutuhan barang dan jasa yang diperlukan guna keberlangsungan operasional perusahaan. [14]

Terdapat prinsip - prinsip yang harus dilakukan dalam pengadaan yaitu:[13]

1. Efisiensi

Prinsip ini memiliki arti dengan menggunakan sumber daya yang tersedia diperoleh barang dan jasa yang didapat dalam waktu yang optimal dan memiliki jumlah, kualitas yang di harapkan.

2. Efektif

Prinsip ini memiliki arti dengan sumber daya yang tersedia didapatkan barang dan jasa yang memiliki nilai manfaat yang tinggi.

3. Persaingan Sehat

Prinsip ini memiliki arti tidak ada praktik kolusi, korupsi dan nepotisme dalam persaingan di antara calon penyedia barang dan jasa.

4. Terbuka

Prinsip ini memiliki arti memberikan kesempatan kepada semua penyedia barang dan jasa untuk mengikuti pengadaan.

5. Transparansi

Prinsip ini memiliki arti memberikan informasi kepada semua calon penyedia barang dan jasa yang berminat, informasi yang diberikan merupakan informasi mengenai aturan pelaksanaan pengadaan barang dan jasa.

6. Tidak Diskriminatif

Pada prinsip ini calon penyedia barang dan jasa yang berminat mengikuti pengadaan barang dan jasa harus mendapatkan perlakuan yang sama.

7. Akuntabilitas

Prinsip akuntabilitas memiliki arti pertanggung jawaban pelaksanaan barang dan jasa kepada pihak yang terkait

2.7 Persediaan

Persediaan adalah sekumpulan barang yang dimiliki dan disimpan yang bertujuan untuk dijual atau digunakan dalam proses produksi atau digunakan untuk keperluan non produksi pada siklus kegiatan yang normal. Persediaan merupakan hal yang penting bagi perusahaan, di mana dari persediaan perusahaan dapat mengelola stok barang yang nantinya akan digunakan untuk keperluan operasional perusahaan.[6] Persediaan dianggap sebagai aset yang dapat dijual dan menghasilkan keuntungan, tetapi persediaan juga dapat dianggap beban atau *liability* karena merupakan pemborosan. Jika jumlah persediaan terlalu besar maka akan menyebabkan kerugian yang besar dan menimbulkan resiko kerusakan barang dan biaya penyimpanan. Namun jika persediaan terlalu sedikit dapat menyebabkan kekurangan persediaan yang menyebabkan terhentinya proses produksi dan tertundanya penjualan.[15] Oleh karena itu pengelolaan persediaan yang baik akan menghasilkan informasi akurat mengenai stok barang.[16]

Terdapat empat jenis persediaan yaitu:[17]

1. *Fluctuation stock*, ialah persediaan yang memiliki fungsi untuk menjaga agar tidak terjadinya fluktuasi permintaan yang tidak diperkirakan dan untuk mengatasi kesalahan dalam perkiraan penjualan, waktu produksi dan pengiriman barang.
2. *Anticipation stock*, merupakan persediaan yang berfungsi untuk memenuhi permintaan yang dapat diprediksi pada periode permintaan yang tinggi, tetapi kapasitas produksi tidak dapat memenuhi permintaan.
3. *Lost-size inventory*, merupakan persediaan yang diadakan pada jumlah yang lebih besar dari kebutuhan saat ini.

4. *Pipeline inventory*, adalah persediaan yang memiliki proses pengiriman dari tempat asal barang ketempat dimana barang tersebut akan digunakan.

2.8 Konsep Sistem Informasi

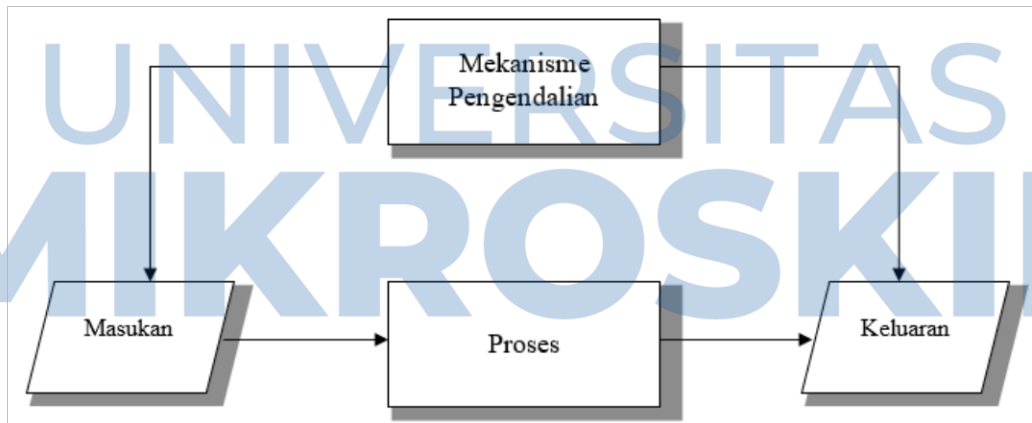
2.8.1 Sistem

Sistem adalah seperangkat elemen yang saling berhubungan dan memiliki bagian-bagian yang beroperasi secara bersamaan untuk mencapai tujuan. Sistem terdiri dari dua jenis yaitu dari sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang tidak mempunyai pengendalian mekanis, sasaran dan umpan balik, sistem terbuka juga merupakan sistem yang dihubungkan dengan lingkungannya dengan melalui arus sumber daya. [18]



Gambar 2.1 Gambar Sistem Terbuka

Sistem tertutup merupakan kebalikan dari sistem terbuka yaitu sistem yang mempunyai pengendalian mekanis, sasaran dan umpan balik. Sistem tertutup juga merupakan sistem yang tidak berinteraksi secara langsung dengan lingkungannya melalui arus sumber daya.



Gambar 2.2 Gambar Sistem Tertutup

2.8.2 Karakteristik Sistem

Sistem mempunyai karakteristik sebagai berikut:[19]

1. Komponen

Suatu sistem biasanya terdiri dari sejumlah komponen yang saling berinteraksi, artinya sistem tersebut bekerja sama dalam membentuk suatu kesatuan. Komponen-komponen sistem atau yang disebut elemen-elemen sistem dapat berupa subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari subsistem untuk menjalankan suatu fungsi dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem

Batas sistem adalah daerah yang membatasi sistem dengan sistem lain atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut, batas sistem juga memungkinkan sistem dipasang sebagai satu kesatuan.

3. Lingkungan Luar Sistem

Lingkungan luar sistem merupakan apapun yang di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat memberikan keuntungan dan kerugian pada sistem. Lingkungan luar sistem yang memberikan keuntungan merupakan energi dari sistem yang harus dijaga dan dipelihara. Sedangkan lingkungan luar sistem yang merugikan harus ditahan dan dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup sistem.

4. Penghubung Sistem

Media penghubung di antara subsistem dengan subsistem lain disebut penghubung sistem, melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi satu masukan untuk subsistem lain, dengan begitu satu subsistem dapat berintegrasi serta membentuk satu kesatuan dengan subsistem yang lainnya.

5. Masukan

Energi yang dimasukkan ke dalam sistem disebut masukan. Masukan dapat berupa *signal input* dan *maintenance input* yang merupakan energi yang diproses sehingga menghasilkan keluaran. Sebagai contoh pada sistem komputer di mana program adalah *maintenance input* yang berfungsi untuk mengoperasikan komputer sedangkan data merupakan *signal input* yang diolah untuk menjadi informasi.

6. Keluaran Sistem

Energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna disebut keluaran. Keluaran dapat menjadi masukan untuk subsistem lain atau kepada supra sistem.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran, contohnya seperti sistem produksi. Sistem ini akan mengolah masukan berupa bahan baku dan bahan lainnya yang menjadi keluaran berupa barang jadi.

8. Sasaran Sistem

Suatu sistem pasti mempunyai sasaran dan tujuan. Sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran sistem yang akan dihasilkan. Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak berguna.

2.8.3 Informasi

Informasi adalah hasil pengolahan data dalam bentuk yang lebih berguna lagi bagi penerimanya, yaitu menggambarkan kejadian yang sebenarnya dan dapat digunakan sebagai alat pengambilan keputusan.[20]

Informasi memiliki karakteristik sebagai berikut :[21]

1. Ketepatan Waktu

Ketepatan waktu berarti bahwa informasi harus sampai ke penerima dalam waktu yang ditentukan. Agar pengambilan keputusan menjadi efektif, informasi harus sampai ke pembuat keputusan pada waktu yang tepat, yang berarti penerima harus memiliki informasi ketika mereka membutuhkannya. Penundaan menghancurkan nilai dari suatu informasi.

2. Akurat

Informasi harus akurat, yang berarti bahwa informasi tersebut harus bebas dari kesalahan dan tidak ambigu. Keakuratan juga berarti bahwa informasi tersebut tidak bias. Informasi salah yang diberikan kepada penerima akan menyebabkan keputusan yang salah.

3. Relevan

Informasi dikatakan relevan jika terutama menjawab penerima apa, mengapa, di mana, kapan, siapa dan mengapa. Informasi yang disajikan kepada penerima sebagai laporan yang berguna akan membantu mereka membuat keputusan.

4. Kecukupan

Kecukupan berarti informasi harus dalam jumlah yang cukup, yaitu media yang menyajikan informasi harus menyediakan laporan yang memuat informasi yang

diperlukan untuk proses pengambilan keputusan. Laporan tidak boleh memberikan informasi yang tidak lengkap atau bahkan lebih, yang dapat menimbulkan situasi yang sulit bagi para pengambil keputusan. Sementara kurangnya informasi menyebabkan krisis, kelebihan informasi menyebabkan kekacauan.

5. Kelengkapan

Informasi yang diberikan kepada penerima harus lengkap dan memenuhi semua kebutuhannya. Informasi yang tidak lengkap dapat menyebabkan keputusan yang buruk dan dengan demikian dapat merugikan suatu pihak tertentu.

6. Ketegasan

Suatu laporan dikatakan berkualitas jika tidak memerlukan analisis lebih lanjut oleh penerima untuk mengambil keputusan.

Terdapat beberapa jenis informasi yang dapat dijelaskan sebagai berikut :[22]

1. *Philosophic Information*

Merupakan jenis informasi yang terkait dengan konsep menggabungkan informasi dengan pengetahuan dan kebijakan.

2. *Cultural Information*

Merupakan informasi yang menekankan aspek budaya.

3. *Objective Information*

Merupakan jenis informasi terkait dengan sifat logis dari sesuatu.

4. *Subjective Information*

Merupakan informasi yang bergantung pada cara penyajian atau cara orang yang menyampaikannya, karena informasi tersebut berkaitan dengan perasaan dan tentang manusia.

5. *Substitutional Information*

Merupakan konsep informasi yang digunakan sebagai komunikasi dalam hal pertukaran informasi.

6. *Absolute Information*

Merupakan inti dari suatu informasi yang dijamin kebenarannya dan tidak memerlukan penjelasan lebih lanjut.

2.8.4 Sistem Informasi

Sistem informasi merupakan kumpulan dari beberapa sistem pada suatu organisasi yang memiliki fungsi untuk mengumpulkan, memproses, menyimpan dan mendistribusikan informasi sebagai pendukung pengambilan keputusan dan pengendalian di dalam organisasi [23]. Sistem informasi memiliki hubungan antara data dan metode dan penggunaan perangkat

keras dan perangkat lunak untuk mengirimkan informasi yang berguna. Sistem informasi juga mencakup komponen-komponen yang saling berhubungan untuk mencapai suatu tujuan yang diharapkan [24].

Sistem informasi terdiri dari komponen-komponen yang disebut dengan blok bangunan:[25]

1. Blok masukan (*input block*).

Sebuah data yang masuk ke dalam sistem informasi disebut masukan (*input*), masukan yang dimaksud termasuk metode-metode dan media yang digunakan dalam menangkap data yang akan di *input* berupa dokumen dasar.

2. Blok model (*model block*).

Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*).

Blok keluaran menghasilkan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*).

Teknologi berfungsi untuk menerima input, menjalankan model, menyimpan dan mengakses data. Teknologi juga memiliki fungsi menghasilkan dan mengirimkan keluaran serta membantu pengendalian sistem secara keseluruhan. Teknologi terdiri dari unsur utama yaitu:

a. Teknisi (*humanware* atau *brainware*).

b. Perangkat lunak (*software*).

c. Perangkat keras (*hardware*).

5. Blok basis data (*database block*).

Blok basis data merupakan kumpulan dari data yang saling berhubungan satu sama lain yang tersimpan di perangkat keras komputer dan digunakan oleh perangkat lunak dimanipulasi.

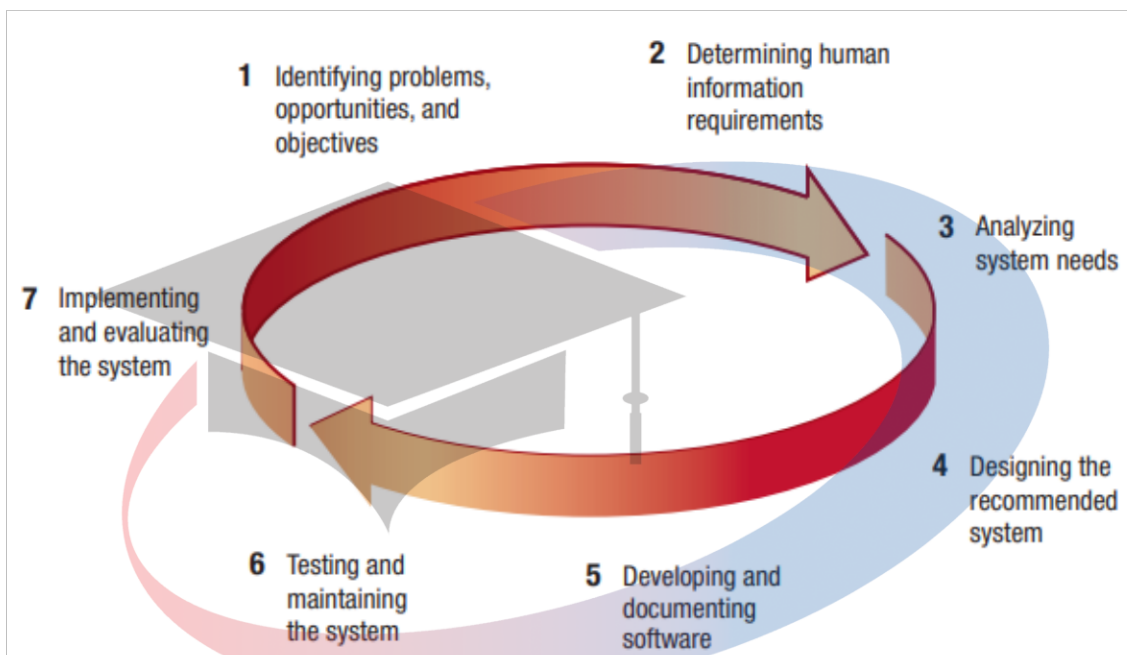
6. Blok kendali (*control block*)

Blok kendali merupakan pengendalian yang dirancang secara khusus untuk mencegah dan menanggulangi gangguan-gangguan terhadap sistem informasi, misalnya seperti bencana alam, api, temperature tinggi, air, debu, ketidakefisienan dan sabotase.

2.9 Metode System Development Life Cycle

SDLC adalah pendekatan bertahap untuk analisis dan desain berdasarkan asumsi bahwa sistem yang dikembangkan dengan baik menggunakan siklus pengguna dan analisis tertentu. Metode SDLC disebut juga dengan metode air terjun (*waterfall*) karena proses analisis sistem menyelesaikan tahap pertama, kemudian berlanjut ke tahap berikutnya dan seterusnya. [26]

Adapun tahapan-tahapan dalam SDLC, antara lain :[26]



Gambar 2.3 Tahapan SDLC Menurut Kendall

1. Mengidentifikasi Masalah, Peluang, dan Tujuan

Pada fase pertama SDLC ini adalah kepentingan terbaik analisis untuk menentukan masalah, peluang, dan tujuan. Langkah ini sangat penting untuk keberhasilan proyek selanjutnya karena tidak ada yang mau membuang waktu untuk menyelesaikan masalah yang salah.

Tahap pertama mengharuskan analisis untuk melihat dengan jujur apa yang terjadi dalam bisnis. Kemudian, bersama anggota organisasi lainnya, analisis mengidentifikasi masalahnya. Orang sering menyebut beragam masalah merupakan alasan mengapa analisis dipanggil sejak awal. Peluang adalah situasi yang menurut analisis dapat ditingkatkan melalui penggunaan sistem informasi terkomputerisasi. Memanfaatkan peluang dapat memungkinkan perusahaan memperoleh keunggulan kompetitif atau menetapkan standar industri.

Menentukan tujuan juga merupakan bagian penting dari fase pertama. Analisis pertama-tama harus memahami apa yang coba dilakukan bisnis. Analisis kemudian akan dapat melihat apakah aspek tertentu dari adopsi sistem informasi dapat membantu perusahaan mencapai tujuannya.

Partisipan pada fase pertama adalah pengguna, analisis, dan koordinator proyek. Kegiatan dalam fase ini meliputi mewawancarai manajemen pengguna, meringkas pengetahuan yang diperoleh, memperkirakan ruang lingkup proyek, dan mencatat hasil. Keluaran dari fase ini adalah laporan kelayakan yang berisi definisi masalah dan ringkasan tujuan. Manajemen kemudian harus membuat keputusan apakah akan melanjutkan proyek yang diusulkan. Jika kelompok pengguna tidak memiliki cukup dana dalam anggaran mereka, atau jika mereka ingin menyelesaikan masalah yang tidak terkait, atau jika masalah tersebut tidak memerlukan sistem komputer, solusi lain dapat diusulkan dan sistem proyek tidak dituntut.

2. Menentukan Persyaratan Informasi

Pada tahap selanjutnya, analisis mengidentifikasi kebutuhan manusia dari pengguna yang relevan, menggunakan berbagai alat untuk memahami bagaimana pengguna berinteraksi dalam konteks bisnis dengan sistem informasi saat ini. Analisis menggunakan metode interaktif seperti wawancara, *sampling*, *review* data, dan penggunaan kuesioner.

Selama fase persyaratan informasi SDLC, analisis berusaha memahami informasi yang dibutuhkan pengguna untuk melakukan pekerjaan mereka. Pada tahap ini, analisis mempertimbangkan bagaimana membuat sistem berguna bagi orang-orang yang terlibat. Bagaimana sistem dapat lebih baik mendukung tugas individu yang perlu dilakukan, bagaimana sistem baru dapat memperluas kemampuan pengguna melebihi apa yang disediakan sistem lama, bagaimana analisis dapat membuat sistem yang berguna bagi pekerja.

Mereka yang terlibat dalam fase ini adalah analisis dan pengguna, pada umumnya adalah manajer operasional dan eksekutif. Analisis sistem harus tahu cara menganalisis fitur sistem saat ini: siapa (orang-orang yang terlibat), apa (aktivitas dalam bisnis), di mana (lingkungan bisnis berlangsung), kapan (waktu berlangsungnya bisnis) dan bagaimana (prosedur yang sedang berlangsung saat ini dan bagaimana hal itu dilakukan) dari perusahaan yang diteliti. Analisis kemudian harus bertanya mengapa perusahaan menggunakan sistem saat ini.

3. Menganalisis Kebutuhan Sistem

Tahap selanjutnya yang dilakukan analisis sistem adalah analisis kebutuhan sistem. Alat dan teknik khusus membantu analisis menentukan persyaratan. Alat-alat seperti *Data Flow Diagram* (DFD) yang menggambarkan *input*, proses, dan *output* dari fungsi bisnis, dan *Activity Diagram* atau *Sequence Diagram* yang menunjukkan urutan kejadian dan gambaran sistem dalam bentuk grafik terstruktur. Dari aliran data lain, urutan, atau diagram, kamus data dikembangkan yang mencantumkan semua data yang digunakan dalam sistem beserta spesifikasinya.

Pada fase SDLC ini, analisis sistem menyiapkan proposal sistem yang meringkas apa yang telah ditemukan tentang pengguna, penggunaan, dan kegunaan sistem saat ini; memberikan analisis biaya-manfaat alternatif dan rekomendasi tentang apa yang harus dilakukan (jika ada). Jika salah satu rekomendasi diterima oleh manajemen, analisis akan melaksanakan pekerjaan tersebut. Setiap masalah sistem unik dan tidak pernah hanya ada satu solusi yang benar. Cara di mana rekomendasi atau solusi dibuat tergantung pada kualitas individu dan pelatihan profesional dari setiap analisis dan interaksi analisis dengan pengguna dalam konteks lingkungan kerja mereka.

4. Merancang Sistem Yang Direkomendasikan

Selama fase desain SDLC, analisis sistem menggunakan informasi yang dikumpulkan sebelumnya untuk melengkapi desain logis dari sistem informasi. Analisis merancang prosedur bagi pengguna untuk membantu mereka memasukkan data dengan benar sehingga data yang dimasukkan ke dalam sistem informasi benar dan tepat. Selain itu, analisis memungkinkan pengguna untuk menyelesaikan entri sistem informasi secara efisien menggunakan teknik formulir dan desain web atau layar yang baik.

Bagian desain logis dari sistem informasi adalah desain *Human-Computer Interaction*. Antarmuka menghubungkan pengguna dengan sistem dan merupakan hal yang sangat penting. Antarmuka pengguna dirancang dengan mempertimbangkan *input* pengguna untuk memastikan bahwa sistem dapat didengar, mudah dibaca dan aman, serta menarik dan menyenangkan untuk digunakan. Contoh antarmuka pengguna fisik termasuk *keyboard* (untuk memasukkan pertanyaan dan jawaban), menu di layar (untuk menerima perintah pengguna), dan berbagai antarmuka pengguna grafis yang menggunakan mouse atau layar sentuh.

Fase desain juga mencakup perancangan *database* yang akan menyimpan banyak data yang dibutuhkan oleh para pengambil keputusan dalam organisasi. Pengguna mendapat manfaat dari *database* yang terorganisir dengan baik yang masuk akal bagi mereka dan selaras dengan cara mereka melihat pekerjaan mereka. Selama fase ini, analis juga bekerja dengan pengguna untuk merancang keluaran (pada layar atau cetakan) yang memenuhi kebutuhan informasi mereka.

5. Mengembangkan dan Mendokumentasikan Perangkat Lunak

Pada fase kelima SDLC, analis bekerja dengan pemrogram untuk mengembangkan perangkat lunak asli yang diperlukan. Selama fase ini, analis bekerja dengan pengguna untuk mengembangkan dokumentasi yang efektif untuk perangkat lunak.

Karena pengguna terlibat sejak awal, dokumentasi harus menjawab pertanyaan yang telah mereka ajukan dan sampaikan kepada analis. Dokumentasi menginstruksikan pengguna tentang cara menggunakan perangkat lunak dan apa yang harus dilakukan jika terjadi kegagalan perangkat lunak.

Pemrogram memiliki peran penting dalam fase ini karena mereka merancang, membuat kode, dan menghapus kesalahan sintaksis dari program komputer. Untuk memastikan kualitas, seorang pemrogram dapat melakukan penelitian desain atau kode, menjelaskan perangkat lunak yang kompleks kepada sekelompok pemrogram lain.

6. Menguji dan Mempertahankan Sistem

Sebelum sistem informasi dapat digunakan, maka harus dilakukan pengujian terlebih dulu untuk mendeteksi masalah sebelum daripada setelah sistem diserahkan kepada pengguna. Pemrogram melakukan beberapa pengujian sendiri dan beberapa pengujian dilakukan oleh analis.

Sangkaian pengujian pertama-tama dijalankan bersama-sama dengan data sampel dan terakhir dengan data nyata dari sistem saat ini. Rencana pengujian biasanya dibuat pada awal SDLC dan disempurnakan seiring kemajuan proyek. Pemeliharaan sistem dan dokumentasinya dimulai selama fase ini dan dilakukan secara teratur sepanjang umur sistem.

7. Mengimplementasikan dan Mengevaluasi Sistem

Pada tahap akhir pengembangan sistem, analis membantu mengimplementasikan sistem informasi. Fase ini termasuk melatih pengguna untuk

menangani sistem. *Vendor* menyediakan beberapa pelatihan, tetapi mengawasi pelatihan adalah tanggung jawab analis sistem. Selain itu, analis harus merencanakan transisi yang mulus dari sistem lama ke sistem baru.

Evaluasi yang ditunjukkan sebagai bagian dari tahap terakhir ini biasanya dimasukkan untuk pembahasan. Sebenarnya, evaluasi dilakukan di setiap tahapan. Kriteria utama yang harus dipenuhi adalah pengguna yang dituju benar-benar menggunakan sistem tersebut.

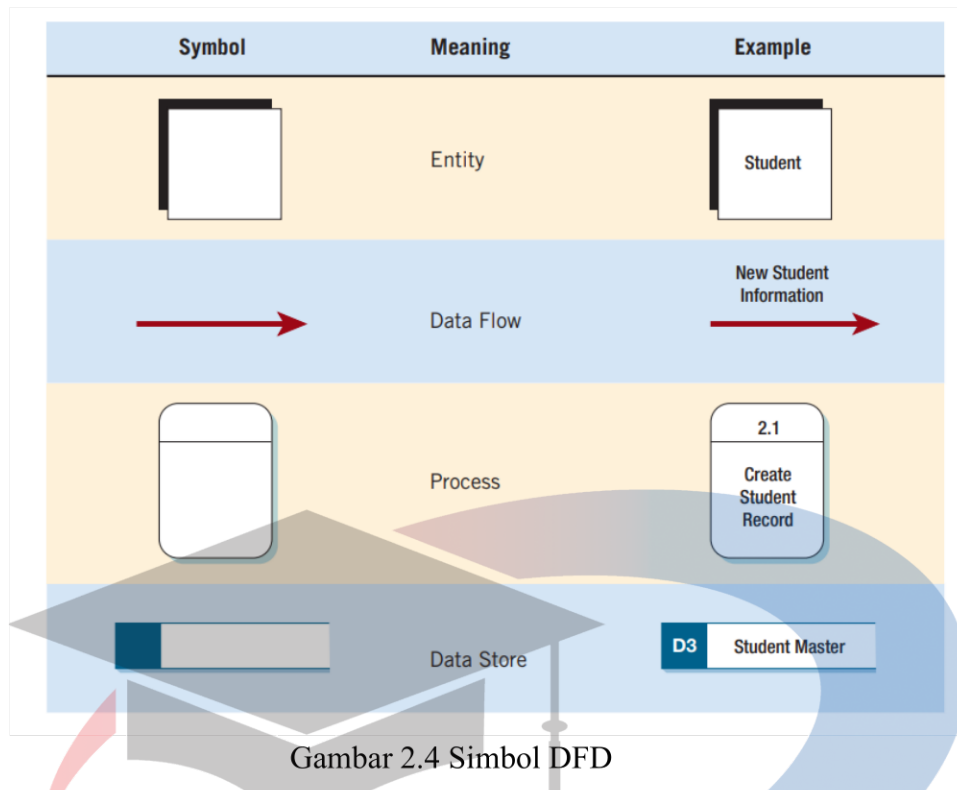
2.10 Alat Bantu dan Teknik Pengembangan Sistem

2.10.1 Data Flow Diagram

Selama fase analisis sistem pada SDLC, seorang analis sistem membuat model visual dari sistem informasi menggunakan serangkaian diagram aliran data. DFD hanya memvisualisasikan bagaimana data bergerak melalui sistem informasi tetapi tidak menjelaskan logika program atau langkah-langkah pemrosesan.[27] *Data Flow Diagram* (DFD) merupakan *tools* tradisional yang digunakan untuk memvisualisasikan arus informasi dalam sebuah sistem. DFD menunjukkan bagaimana informasi itu masuk dan keluar dari sistem, informasi apa yang berubah dan di mana informasi disimpan. Tujuan dari DFD adalah untuk menunjukkan ruang lingkup dan batas-batas dari suatu sistem.[28]

Sebuah DFD dapat dipecah menjadi beberapa level yang lebih detail untuk menggambarkan arus informasi atau fungsi yang lebih detail dan DFD juga menyediakan prosedur untuk pemodelan aliran informasi dan fungsional, oleh karena itu DFD tidak cocok digunakan dalam membangun pemodelan sistem perangkat lunak menggunakan pemrograman berorientasi objek tetapi DFD lebih sesuai digunakan dalam pemodelan perangkat lunak yang menggunakan pemrograman terstruktur karena pemrograman terstruktur memecah bagian-bagiannya dengan prosedur-prosedur dan fungsi. [28]

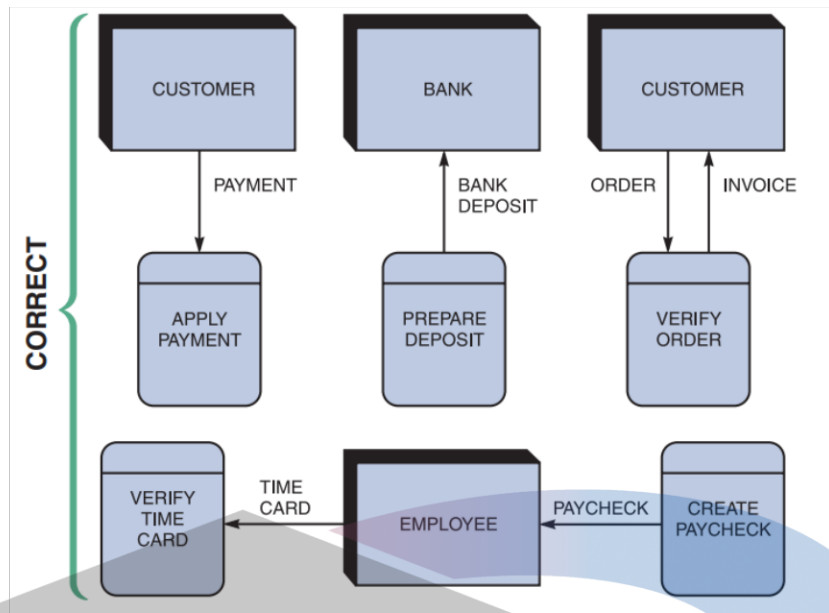
Ada beberapa versi dari simbol DFD yaitu simbol *Crish Gane* dan *Trish Sarson*, *Edward Yourdon* dan *Tom DeMarco*, kedua simbol tersebut memiliki tujuan yang sama, pada tugas akhir ini penulis menggunakan simbol *Crish Gane* dan *Trish Sarson*.[27] Terdapat empat simbol yang digunakan dalam menggambarkan DFD yaitu *Entity*, *Data Flow*, *Proses*, dan *Data Store*. [26]



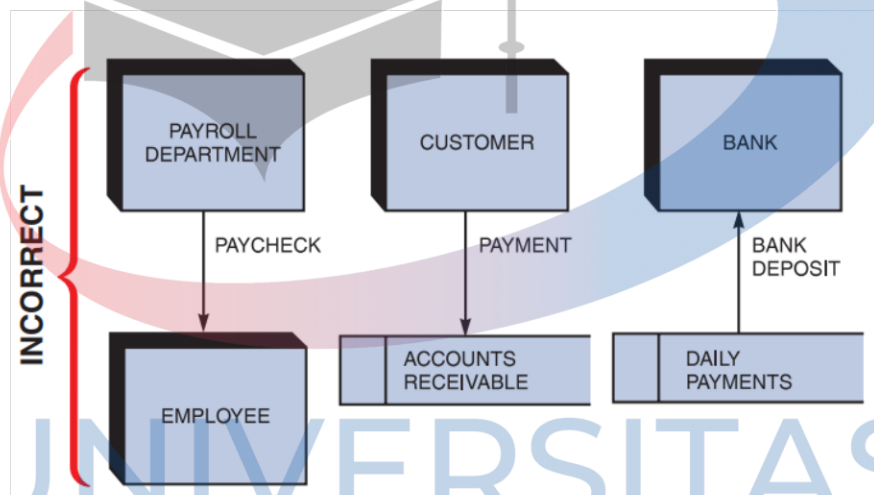
Gambar 2.4 Simbol DFD

a) Entity

Entitas digambarkan dalam bentuk persegi panjang yang diarsir sehingga terlihat tiga dimensi dan nama dari entitas dituliskan di dalam simbol entitas. Penamaan entitas biasanya menggunakan kata benda seperti departemen, organisasi luar, sistem informasi atau seseorang. Entitas DFD juga disebut *terminator* karena merupakan asal data atau tujuan akhir. DFD hanya menunjukkan entitas eksternal yang berfungsi untuk mengirim data atau menerima *output* dari sistem. Pada DFD entitas menunjukkan batas-batas sistem dan bagaimana sistem berinteraksi dengan dunia luar. Contohnya, entitas pelanggan mengirimkan data pesanan ke sistem pemrosesan pesanan. Entitas luar atau entitas eksternal dapat menjadi sumber(*source*) atau tujuan dari data yaitu yang menerima data dari sistem(*sink*) atau keduanya, tetapi setiap entitas harus terhubung ke suatu proses dengan aliran data. Pada gambar 2.5 dan 2.6 menunjukkan contoh yang benar dan salah dalam aturan penggambaran entitas DFD



Gambar 2.5 Contoh Penggambaran Entitas Yang Benar



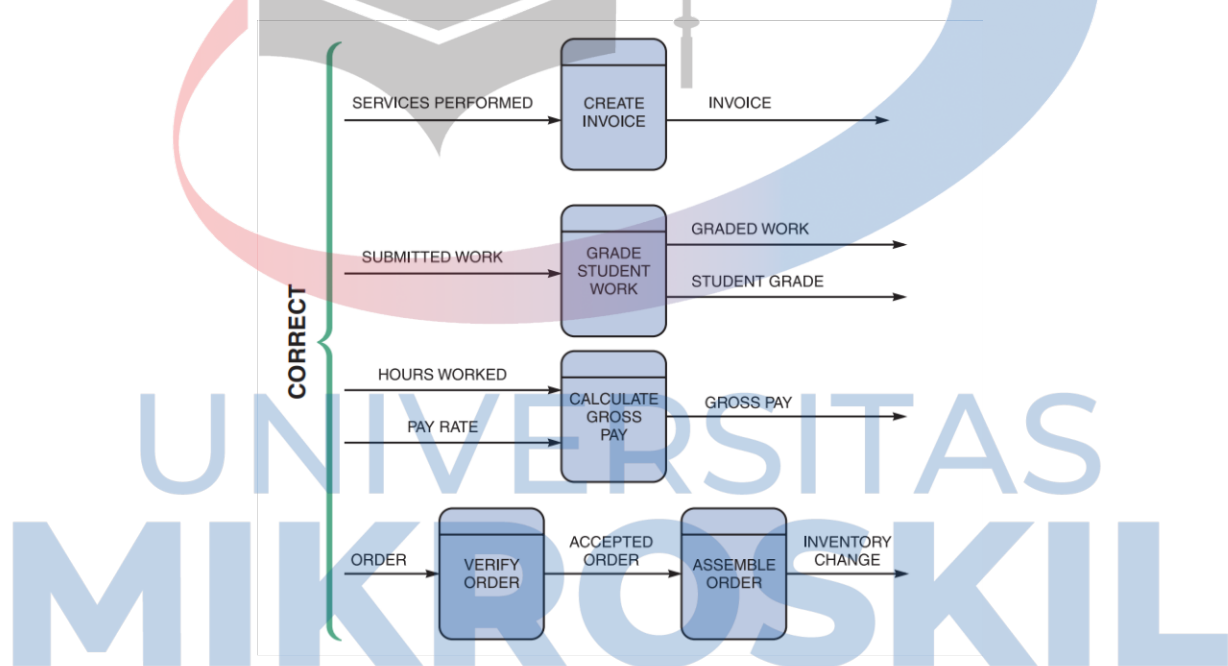
Gambar 2.6 Contoh Penggambaran Entitas Yang Salah

Contoh penggambaran entitas eksternal yang salah di atas pada DFD karena entitas eksternal harus dihubungkan melalui aliran data ke proses dan tidak langsung ke *data store* ataupun entitas eksternal lain. [27]

b) Data Flow

Data Flow menunjukkan alur perpindahan data dari bagian sistem informasi ke bagian lainnya. *Data flow* digambarkan dengan simbol anak panah tunggal ataupun ganda, nama *data flow* biasanya dituliskan di atas, di bawah, atau di samping garis anak panah dan penamaan *data flow* menggunakan kata benda tunggal atau kata sifat, jika diperlukan.

Data Flow pada DFD mewakili satu atau lebih item data. Contoh, *data flow* dapat terdiri dari item data seperti nomor ID siswa atau mencakup sekumpulan data seperti daftar kelas dengan nomor ID siswa, nama dan tanggal pendaftaran untuk kelas tertentu. Pada gambar 2.7 di bawah ini menunjukkan contoh *data flow* yang benar yang dihubungkan dengan proses karena menunjukkan sebuah proses mengubah isi atau *form* inputan data menjadi keluaran dengan *output* yang berbeda dari *inputnya*, *data flow* pada proses setidaknya memiliki satu aliran data masuk dan satu aliran data keluar. Contohnya pada proses *CREATE INVOICE*. Pada simbol proses dapat memiliki aliran data yang keluar lebih dari satu, seperti yang ditunjukkan pada proses *GRADE STUDENT WORK*, atau lebih dari satu aliran data masuk, seperti yang ditunjukkan dalam proses *CALCULATE GROSS PAY*. Proses juga dapat dihubungkan ke simbol lain, seperti proses lainnya dengan aliran data seperti yang ditunjukkan oleh hubungan di antara *VERIFY ORDER* dan *ASSEMBLE ORDER*.

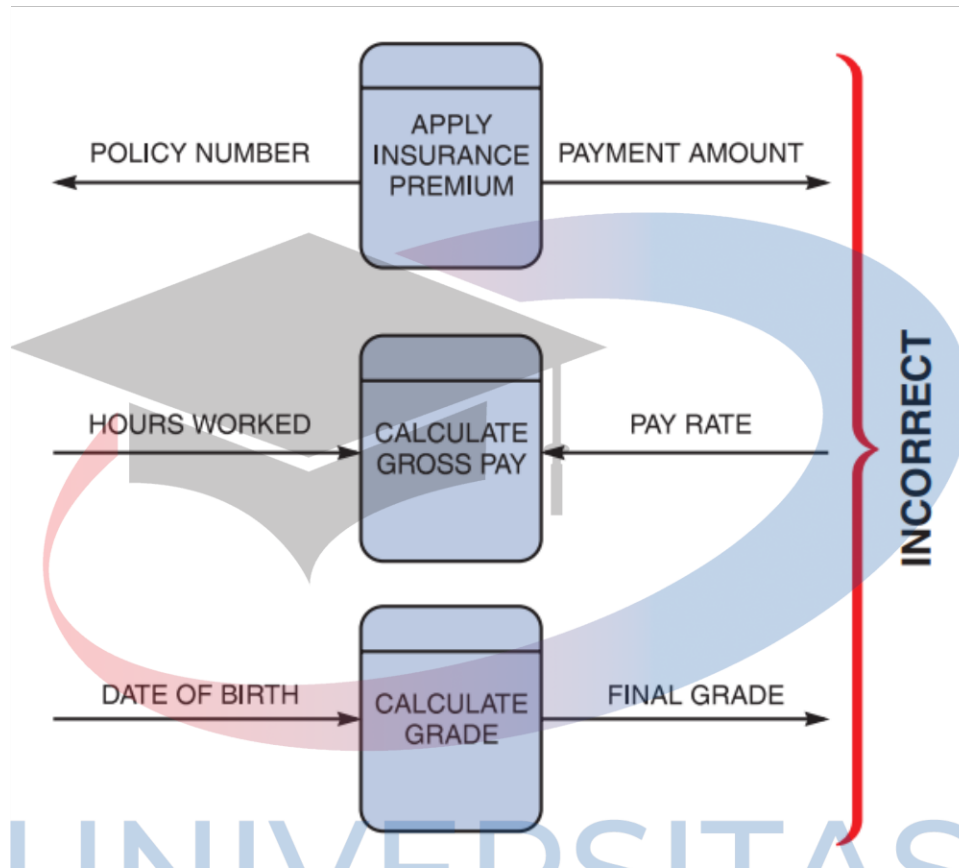


Gambar 2.7 Contoh Hubungan Yang Benar Dari Data Flow Dan Proses.

Pada gambar 2.8 menunjukkan contoh hubungan yang salah dalam penggambaran antara *data flow* dan proses dimana terdapat *Spontaneous generation*, *Black hole* dan *Gray hole*. *Spontaneous generation*. Pada proses *APPLY INSURANCE PREMIUM* menghasilkan *output* tetapi tidak memiliki aliran data *input*.

Black hole. Pada proses *CALCULATE GROSS PAY* disebut proses *black hole* karena proses tersebut memiliki *input* tetapi tidak memiliki *output*.

Gray hole. Pada proses *CALCULATE GRADE* inputan tanggal lahir tidak cukup menghasilkan *output* nilai akhir. Proses disebut *gray hole* karena memiliki satu *input* dan satu *output*, tetapi inputan tersebut jelas tidak cukup menghasilkan *output* yang ditampilkan.[27]



Gambar 2.8 Contoh Hubungan Yang Salah Dalam Penggambaran Antara *Data Flow* Dan Proses

c) *Process*

Proses dapat menerima data masukan dan menghasilkan keluaran yang memiliki bentuk, isi atau keduanya yang berbeda dari data masukannya. Misalnya, sebuah proses penghitungan gaji menggunakan dua masukan (tingkat gaji dan kerja) untuk menghasilkan satu keluaran (gaji total). Proses dapat mengandung logika bisnis, yang juga disebut aturan bisnis, yang mengubah data dan menghasilkan hasil yang diperlukan.

Proses digambarkan dengan simbol persegi panjang dengan sudut membulat, penamaan proses dinamai dengan kata kerja (dan kata sifat, jika perlu) diikuti oleh kata benda tunggal dan dituliskan di dalam persegi panjang. Nama proses mengidentifikasi fungsi

yang spesifik. Misalnya, PEMBAYARAN TAGIHAN, HITUNG KOMISI, HITUNG NILAI AKHIR, VERIFIKASI PESANAN, dan MASUKAN KERANJANG BELANJA. Detail pemrosesan tidak ditampilkan di DFD.

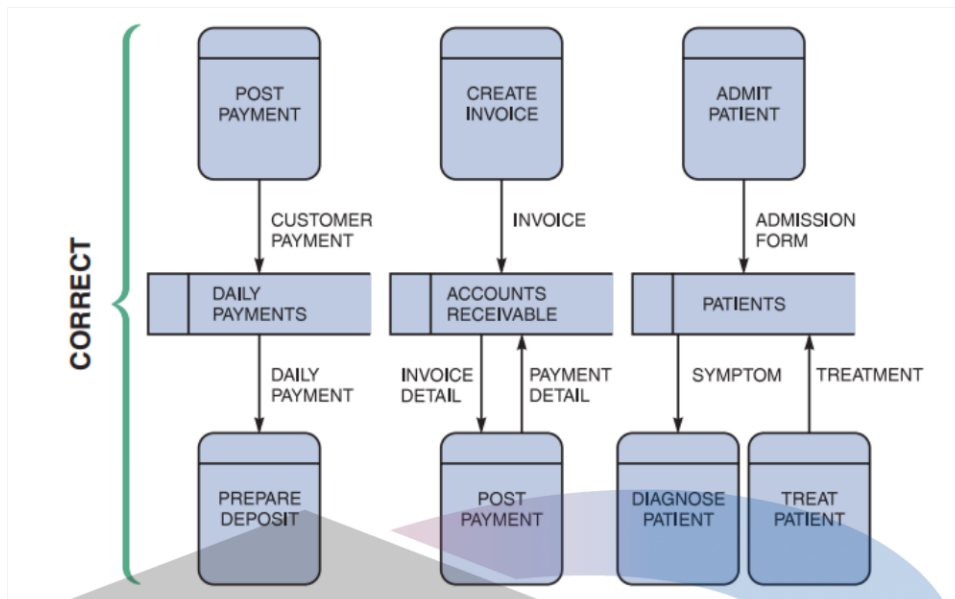
Pada DFD, simbol proses dapat disebut sebagai kotak hitam (*Black Box*), karena masukan, keluaran, dan fungsi umum proses diketahui, tetapi detail dan logika yang mendasari proses tidak diketahui. Dengan menunjukkan proses sebagai kotak hitam, seorang analis sistem dapat membuat DFD yang menunjukkan bagaimana sistem berfungsi tetapi menghindari detail dan kerumitan yang tidak perlu.[27]

d) *Data Store*

Pada DFD *data store* digunakan untuk merepresentasikan data yang disimpan sistem karena satu atau lebih proses perlu menggunakan kembali data tersebut nantinya. Contohnya, seorang guru perlu menyimpan nilai siswa hasil ujian dan tugas selama semester sehingga guru tersebut dapat memberikan nilai akhir di akhir semester. DFD tidak menunjukkan isi detail dari sebuah *data store*, struktur spesifik dan elemen data didefinisikan nantinya dengan kamus data.

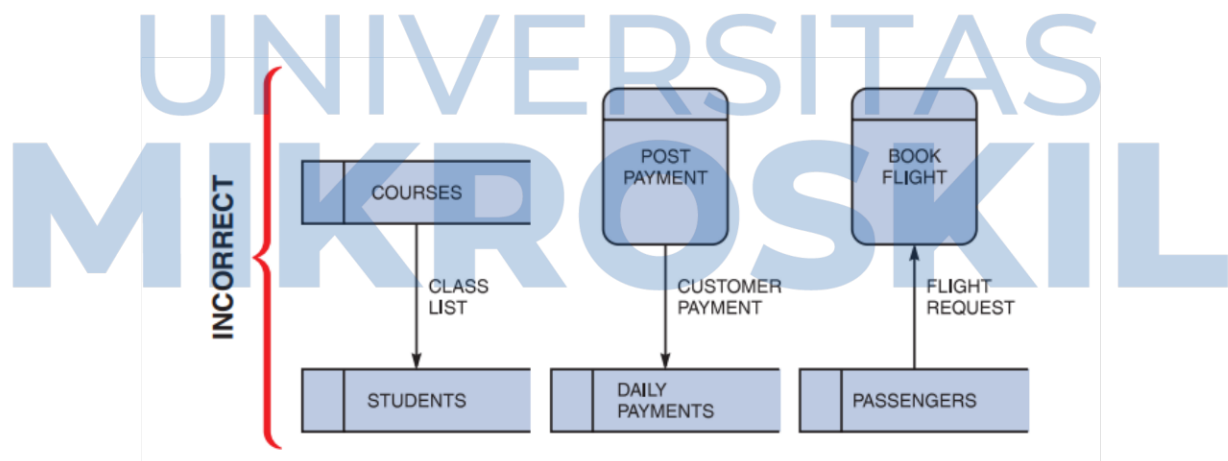
Di dalam DFD, simbol *Crish Gane* dan *Trish Sarson* digambarkan dengan persegi panjang data yang terbuka di sisi kanan dan tertutup di sisi kiri. Penamaan nama *data store* dituliskan di antara garis dan mengidentifikasi data yang disimpan serta dinamai dengan kata benda dan kata sifat jika diperlukan. Contohnya seperti, SISWA, PENERIMAAN PEMBAYARAN, PRODUK, PEMBAYARAN HARIAN, PESANAN PEMBELIAN. Setiap *data store* juga diberikan nomor referensi yang unik, seperti D1, D2, D3, dan seterusnya. [26]

Data store juga harus terhubung ke proses dengan aliran data. Pada gambar 2.9 menunjukkan contoh tipikal penggambaran *data store*. Dimana *data store* mewakili penyimpanan data untuk digunakan proses lain di masa depan, dalam setiap kasus *data store* memiliki setidaknya satu aliran data masuk dan keluar dan juga terhubung ke simbol proses dengan aliran data.



Gambar 2.9 Contoh Hubungan Yang Benar Antara Data Store Dan Proses

Pada gambar 2.10 menunjukkan aturan dalam penggambaran DFD yang salah, karena *data store* harus memiliki setidaknya satu aliran data masuk dan keluar. Pada contoh pertama, dua *data store* terhubung dengan tidak benar karena tidak ada proses diantar keduanya. Selain itu, proses *COURSES* tidak memiliki aliran data masuk dan *STUDENTS* tidak memiliki aliran data yang keluar. Contoh kedua dan ketiga, *data store* tidak mempunyai aliran data keluar atau masuk.



Gambar 2.10 Contoh Hubungan Yang Salah Antara Data Store Dan Proses.

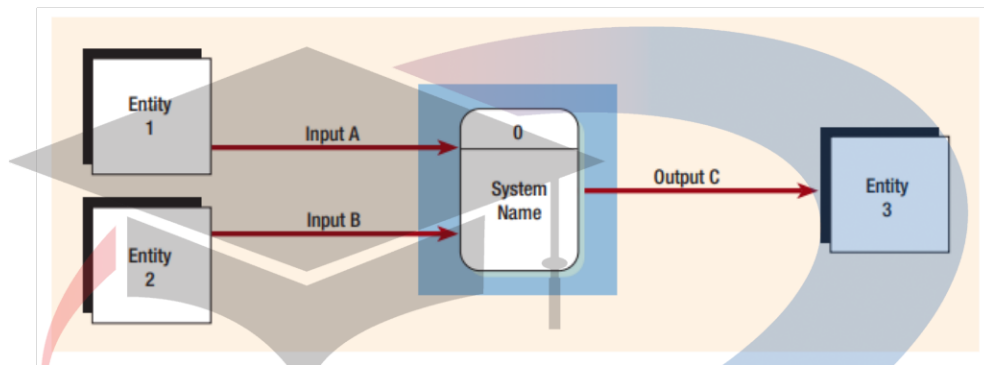
Terdapat pengecualian untuk persyaratan bahwa *data store* harus memiliki setidaknya satu aliran data masuk dan keluar. Tetapi pada beberapa situasi, *data store* tidak mempunyai aliran data masuk dikarenakan berisi data referensi tetap yang tidak diperbarui oleh sistem.[27]

DFD dikembangkan dengan cara berjenjang yang diawali dengan *context diagram* (CD) atau level 0, kemudian level 1, level 2, level 3 hingga selanjutnya sesuai dengan kompleksitas dari sistem yang dikembangkan. [29]

Berikut penjelasan tahapan-tahapan pada perancangan DFD:

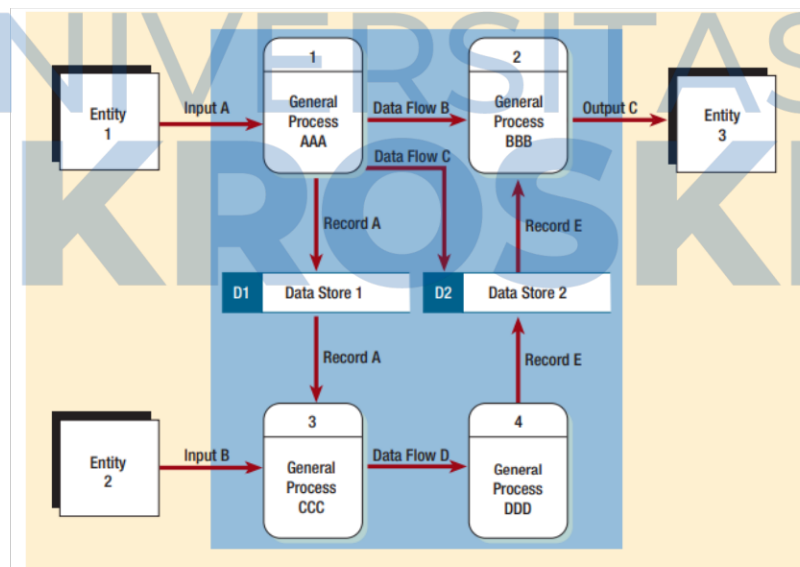
1. Membuat Diagram *Context* yang merupakan diagram level tertinggi pada DFD yang menggambarkan keseluruhan *input* ataupun *output* pada sistem. Diagram ini terdiri dari suatu proses yang menggambarkan ruang lingkup suatu sistem. [30]

Berikut contoh diagram *context*:



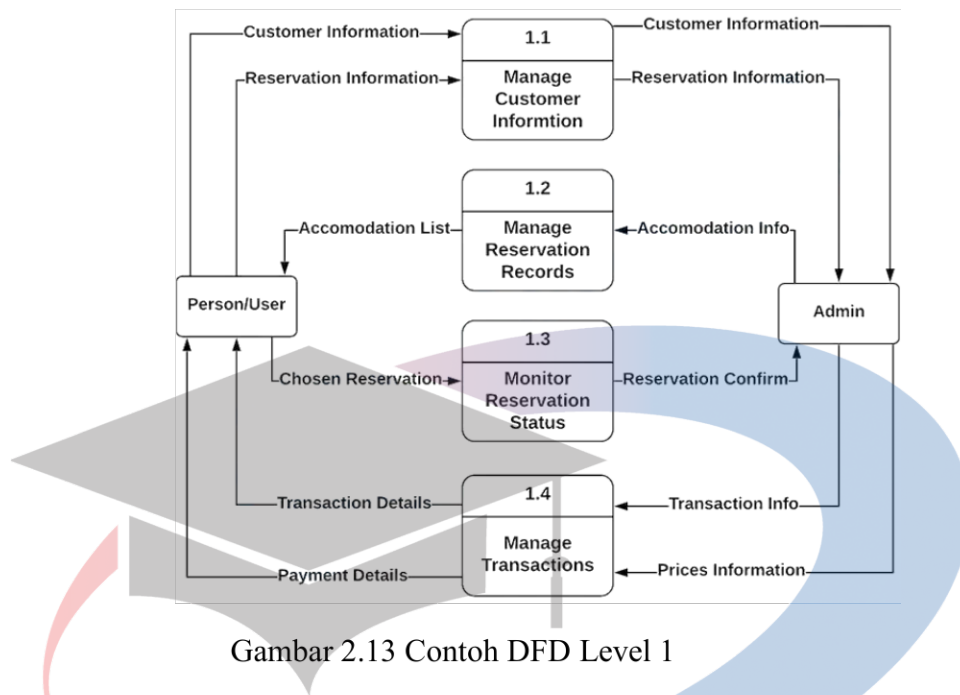
Gambar 2.11 Contoh Diagram *Context*

2. Membuat DFD level 0 di mana DFD level 0 digunakan untuk memvisualisasikan interaksi antara sistem yang akan dikembangkan dengan entitas luar.



Gambar 2.12 Contoh DFD Level 0

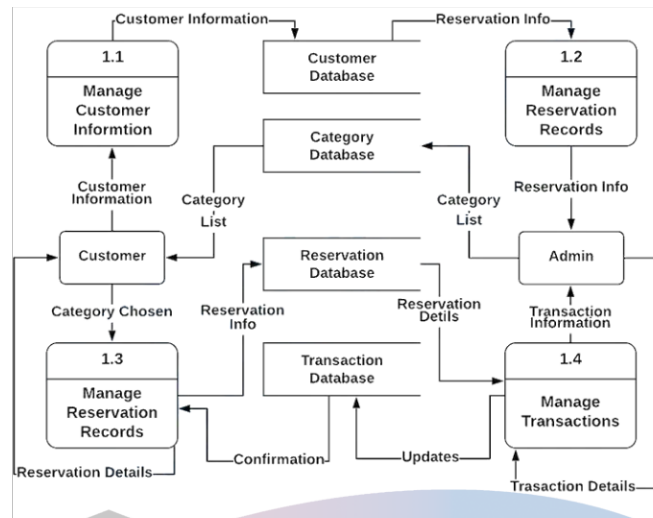
3. Membuat DFD level 1, DFD level 1 merupakan hasil *breakdown* dari DFD level 0 yang dibuat sebelumnya, DFD level 1 digunakan untuk memvisualisasikan modul-modul di dalam sistem yang akan dikembangkan.



Gambar 2.13 Contoh DFD Level 1

4. Membuat DFD level 2, DFD level 2 merupakan hasil *breakdown* dari DFD level 1 yang dibuat menjadi DFD level 2. Pada DFD level 2 modul-modul mana saja yang akan di *breakdown* lebih detail tergantung pada tingkat kedetailan modul. Jika modul tersebut sudah cukup detail dan rinci maka modul tidak perlu di *breakdown*. Jumlah modul pada DFD level 2 untuk sebuah sistem sama dengan jumlah modul yang di *breakdown* pada level 1.

UNIVERSITAS
MIKROSKIL



Gambar 2.14 Contoh DFD Level 2

5. Membuat DFD level 3 dan seterusnya, Pada DFD level 3,4 dan seterusnya juga merupakan *breakdown* dari modul DFD level di atasnya. Aturan *breakdown* pada level 3,4,5 dan seterusnya sama persis seperti pada DFD level 1 atau level 2.

2.10.2 Kamus Data

Kamus data atau repositori data adalah pusat penyimpanan informasi data dari suatu sistem. Analisis menggunakan kamus data untuk mengumpulkan, mendokumentasikan, dan mengatur fakta spesifik tentang suatu sistem, termasuk konten aliran data, penyimpanan data, entitas, dan proses. Kamus data juga mendefinisikan dan menjelaskan semua elemen data dan kombinasi elemennya yang memiliki arti.

Terdapat laporan yang berharga yang diperoleh dari kamus data yaitu:

Daftar berdasarkan abjad dari semua elemen data yang bisa terurut berdasarkan nama. Laporan yang mendeskripsikan setiap elemen data dan menunjukkan pengguna atau departemen yang bertanggung jawab untuk entri data, ubah atau penghapusan data. Laporan dari semua aliran *data store* yang menggunakan elemen data tertentu. Laporan yang detail dan menunjukkan semua karakteristik elemen data, catatan, aliran data, proses, atau item terpilih lainnya yang disimpan dalam kamus data.

Elemen data, juga dikenal sebagai item data atau bidang, adalah bagian dari data terkecil yang bermakna dalam sistem informasi. Contoh dari elemen data adalah nilai siswa, gaji, nomor jaminan sosial, saldo rekening, dan nama perusahaan. Elemen data digabungkan menjadi catatan, juga dikenal sebagai struktur data. *Record* merupakan kombinasi bermakna dari elemen data terkait yang disertakan dalam aliran data atau disimpan dalam penyimpanan data. Misalnya, catatan inventaris toko suku cadang mobil dapat mencakup nomor suku cadang,

deskripsi, kode pemasok, tingkat inventaris minimum dan maksimum, biaya, dan daftar harga. Hubungan yang bermakna ada antara elemen kamus data. Misalnya, aliran dan penyimpanan data didasarkan pada struktur data yang pada gilirannya terdiri dari elemen data. Aliran data terhubung ke penyimpanan data, entitas, dan proses.[27]

Penting untuk mendokumentasikan hubungan ini secara akurat sehingga kamus data konsisten dengan DFD. Semakin kompleks sistemnya, semakin sulit memelihara dokumentasi yang lengkap dan akurat. Untungnya, alat *Case* modern menyederhanakan tugas dengan secara otomatis mentransfer dokumentasi dari diagram pemodelan ke dalam repositori pusat serta informasi yang dimasukkan oleh pengguna.[27]

Kamus data mempunyai beberapa simbol yang berfungsi untuk menjelaskan informasi tambahan sebagai berikut: [28]

Tabel 2.1 Simbol Kamus Data

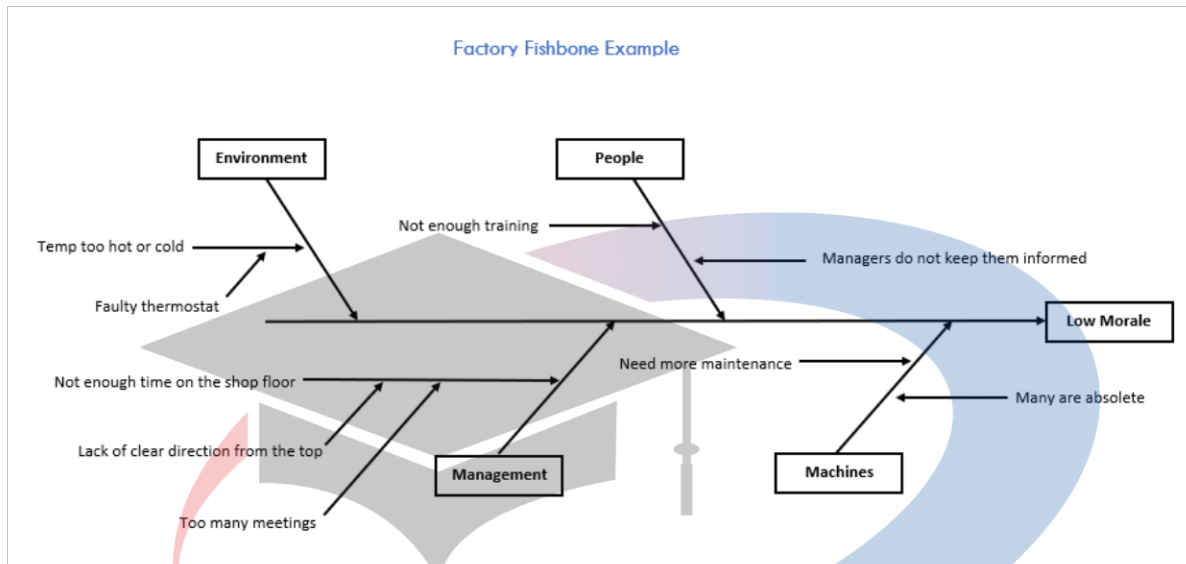
Simbol	Keterangan
=	Disusun atau terdiri dari
+	dan
[]	Baik... atau...
{ } ⁿ	n kali diulang atau bernilai banyak
()	Data opsional
* ... *	Batas komentar

Pada DFD kamus data harus bisa dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan dengan tabel hasil perancangan basis data berarti perancangan basis data dengan perancangan DFD belum sesuai, dan harus ada yang memperbaiki baik perancangan DFD, basis datanya ataupun keduanya.

2.10.3 Fishbone Diagram

Fishbone diagram disebut juga *ishikawa diagram* yang menunjukkan hubungan sebab akibat. Diagram ini terkait produktivitas dan manajemen, sehingga dapat digunakan untuk menetapkan faktor penyebab. *Fishbone diagram* membantu untuk menunjukkan faktor-faktor penting yang mempengaruhi kualitas dan memiliki dampak pada masalah yang terjadi serta dapat melihat detail aspek-aspek yang mempengaruhi dan berhubungan dengan faktor utama dalam masalah tersebut.[31]

Fishbone diagram adalah alat analitik yang secara grafis mewakili kemungkinan penyebab suatu masalah. Menggunakan *Fishbone diagram*, analis pertama-tama mengidentifikasi masalah dan menggambarkan tulang utama bersama dengan tulang kecil yang mewakili kemungkinan penyebab masalah. Sub-tulang mewakili kemungkinan penyebab dari masalah.



Gambar 2.15 Contoh Diagram *Fishbone*

Dalam contoh yang ditunjukkan dalam Gambar 2.15, masalahnya adalah moral yang rendah, dan analis telah mengidentifikasi empat area untuk diselidiki yaitu *Lingkungan*, *Manusia*, *Manajemen*, dan *Mesin*. Dalam setiap area, analis mengidentifikasi kemungkinan penyebab dan menggambarinya ke dalam sub-tulang horizontal. Misalnya, *suhu terlalu panas atau dingin* adalah kemungkinan penyebab di tulang *Lingkungan*. Untuk setiap penyebab, analis harus menggali lebih dalam dan mengajukan pertanyaan: Apa yang bisa menyebabkan gejala ini terjadi? misalnya, mengapa terlalu panas? jika jawabannya adalah *termostat yang rusak*, analis mengindikasikan ini sebagai sub-tulang untuk penyebab *temperatur terlalu panas atau dingin*. Dengan cara ini, analis menambahkan sub-tulang tambahan ke diagram, sampai analis mengungkap akar penyebab masalah, bukan hanya gejalanya.[26]

Berikut adalah sumber-sumber yang mempengaruhi masalah kualitas pada *fishbone* diagram berdasarkan prinsip 5M dan 1E yaitu: [32]

- a. *Man* (manusia), berkaitan dengan kekurangan pengetahuan, keterampilan dasar yang berhubungan dengan fisik, kelelahan, stres, ataupun ketidakpedulian.

- b. *Machine* (mesin), berkaitan dengan tidak adanya sistem pemeliharaan preventif pada mesin produksi termasuk pada fasilitas dan peralatan lainnya, terlalu rumit ataupun tidak sesuai dengan spesifikasi tugas.
- c. *Method* (metode kerja), berkaitan dengan bagaimana proses dilakukan dan memenuhi spesifikasi.
- d. *Materials* (bahan baku), berkaitan dengan spesifikasi mutu, penanganan bahan baku dan pengecekan bahan baku yang digunakan selaka proses produksi.
- e. *Measurement* (pengukuran), berkaitan dengan kegiatan menentukan angka untuk suatu objek secara sistematis.
- f. *Environment* (lingkungan), berkaitan dengan tempat dan waktu kerja, kesehatan, aspek kebersihan, keselamatan kerja dan lingkungan kerja yang kondusif.

2.10.4 Normalisasi

Normalisasi adalah proses pembuatan desain tabel dengan memberikan bidang atau atribut tertentu pada setiap tabel dalam *database*. Desain tabel menentukan bidang dan kunci utama dalam tabel atau file tertentu. Bekerja dengan kumpulan desain tabel, proses normalisasi digunakan untuk mengembangkan desain *database* keseluruhan yang sederhana, fleksibel, dan bebas redundansi data. Normalisasi melibatkan alat implementasi yang dapat membantu mengidentifikasi dan memperbaiki masalah dan kompleksitas yang melekat pada desain tabel.[27]

Normalisasi memberikan transformasi tampilan pengguna yang kompleks dan penyimpanan data menjadi kumpulan struktur data yang lebih kecil dan lebih stabil. Selain lebih sederhana dan lebih stabil, struktur data yang dinormalisasi lebih mudah dipelihara daripada struktur data lainnya.

Seorang analis menormalkan struktur data dalam tiga langkah. Setiap langkah melibatkan prosedur penting yang menyederhanakan struktur data.

Hubungan dari tampilan pengguna atau penyimpanan data sulit untuk dinormalkan. Langkah pertama dalam proses ini adalah menghapus semua grup berulang dan menentukan kunci utama. Untuk melakukan ini, hubungan harus dipecah menjadi dua atau lebih hubungan. Pada titik ini, hubungan tersebut mungkin sudah berada dalam bentuk normal ketiga, namun kemungkinan langkah lebih lanjut perlu diambil untuk mengubah hubungan tersebut menjadi bentuk normal ketiga.

Langkah kedua adalah memastikan bahwa semua properti bukan kunci bergantung sepenuhnya pada kunci utama. Semua ketergantungan parsial dihapus dan ditempatkan di hubungan lain. Langkah ketiga adalah menghapus semua dependensi transitif. Ketergantungan transitif adalah di mana atribut bukan kunci bergantung pada atribut bukan kunci lainnya.[26]

2.10.4.1 Format Notasi Standar

Desain tabel lebih mudah jika format notasi standar digunakan untuk menunjukkan struktur tabel, bidang, dan kunci utama. Format notasi standar pada contoh berikut diawali dengan nama tabel, diikuti dengan ekspresi dalam tanda kurung yang berisi nama bidang yang dipisahkan dengan koma. berisi nama bidang yang dipisahkan koma. Kunci utama digarisbawahi, seperti ini:

NAME (FIELD 1, FIELD 2, FIELD 3)

Saat mendesain data, analis harus dapat mengidentifikasi kelompok bidang berulang. Grup berulang adalah kumpulan dari satu atau beberapa bidang yang dapat muncul lebih dari satu kali dalam catatan, yang masing-masing dapat muncul beberapa kali dalam catatan, setiap kejadian memiliki nilai yang berbeda. Sebuah contoh khas dari kelompok berulang ditunjukkan pada Gambar 2.16. Jika sebuah perusahaan menggunakan dokumen tertulis asli untuk mencatat pesanan pembelian, mungkin terlihat seperti ini. Seperti yang ditunjukkan pada Gambar 2.16, kedua pesanan berisi banyak item, yang merupakan grup berulang dalam nomor pesanan yang sama. Perhatikan bahwa selain nomor dan tanggal pesanan, catatan beberapa produk berisi pengulangan *PRODUCT NUMBER*, *DESCRIPTION*, *NUMBER ORDERED*, *SUPPLIER NUMBER*, *SUPPLIER NAME*, dan *ISO*. Grup berulang dapat dianggap sebagai kumpulan catatan anak yang terkandung dalam catatan induk.

ORDER	DATE	PRODUCT NUMBER	DESCRIPTION	NUMBER ORDERED	SUPPLIER NUMBER	SUPPLIER NAME	ISO
86223	9-13-2019	304	Blue gadget	7	A-602	Acme	Yes
		633	Assembly	1	J-995	Jones	No
		684	Super gizmo	4	C-876	Cabot	Yes
86390	9-14-2019	128	Steel widget	12	A-602	Acme	Yes
		304	Blue gadget	3	A-602	Acme	Yes
86467	9-15-2019	304	Blue gadget	144	A-602	Acme	Yes

Gambar 2.16 Tabel *Order* Yang Belum Di Normalisasi

Diagram tabel yang berisi grup berulang disebut denormalisasi (*unnormalized*). Metode notasi standar untuk merepresentasikan desain non-standar adalah dengan menyertakan grup bidang berulang dalam tanda kurung kedua. Contoh tabel denormalisasi terlihat seperti ini:

NAME (FIELD 1, FIELD 2, FIELD 3, (REPEATING FIELD 1, REPEATING FIELD 2)).

Sekarang pertimbangkan desain tabel *ORDER* yang di denormalisasi yang ditunjukkan pada Gambar 2.16. Menurut pedoman penilaian, desain dapat digambarkan sebagai berikut:

ORDER (ORDER, DATE, (PRODUCT NUMBER, DESCRIPTION, NUMBER ORDERED, SUPPLIER NUMBER, SUPPLIER NAME, ISO))

Notasi menunjukkan bahwa desain tabel *ORDER* berisi delapan bidang, tercantum dalam tanda kurung luar. Bidang *ORDER* digarisbawahi untuk menunjukkan bahwa itu adalah kunci utama. *PRODUCT NUMBER*, *DESCRIPTION*, *NUMBER ORDERED*, *SUPPLIER NUMBER*, *SUPPLIER NAME*, dan *ISO*. *NUMBER ORDERED* diapit oleh tanda kurung untuk menunjukkan bahwa mereka adalah bidang dalam grup berulang. Perhatikan bahwa *PRODUCT NUMBER* juga digarisbawahi karena berfungsi sebagai kunci utama untuk grup berulang. Jika seorang pelanggan menempatkan tiga produk yang berbeda dalam suatu pesanan, keenam bidang tersebut harus diulangi untuk setiap produk, seperti yang ditunjukkan pada Gambar 2.16. [27]

2.10.4.2 Bentuk Normal Pertama

Suatu tabel berada dalam bentuk normal pertama (1NF) jika tidak berisi grup berulang. Untuk mengubah desain tidak beraturan menjadi 1NF, kunci utama tabel harus diperluas untuk menyertakan kunci utama grup berulang.

Misalnya, dalam tabel *ORDER* yang ditunjukkan pada Gambar 2.16, grup berulang terdiri dari enam bidang:

PRODUCT NUMBER, *DESCRIPTION*, *NUMBER ORDERED*, *SUPPLIER NUMBER*, *SUPPLIER NAME*, dan *ISO*. Dari ketiga bidang tersebut, *PRODUCT NUMBER* dapat menjadi kunci utama karena secara unik mengidentifikasi setiap contoh dari grup berulang. *DESCRIPTION* tidak dapat menjadi kunci utama karena mungkin unik atau tidak. Misalnya, sebuah perusahaan dapat menjual beberapa barang dengan nama deskriptif yang sama, seperti mesin cuci dan menggunakan suatu barang berkode numerik untuk mengidentifikasi setiap ukuran mesin cuci secara unik.

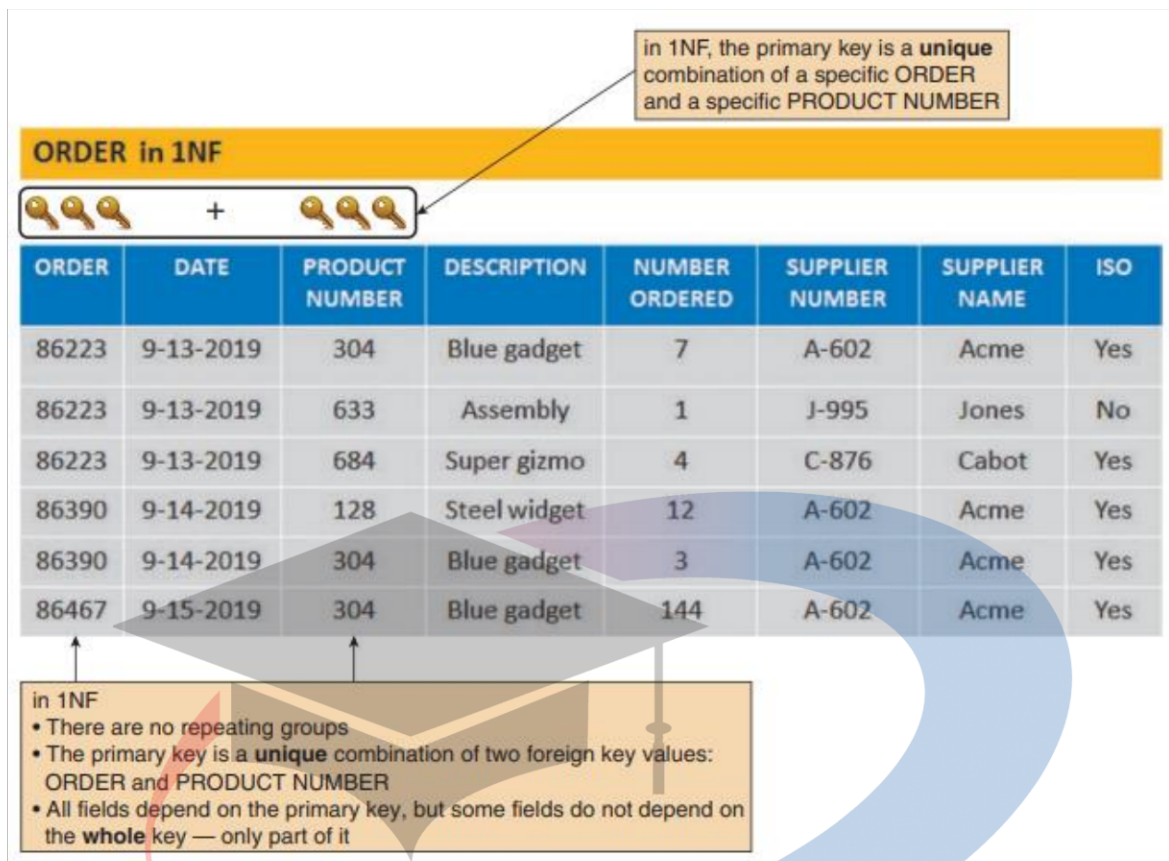
Saat kunci utama tabel *ORDER* diperluas untuk menyertakan *PRODUCT NUMBER*, grup berulang dihapus dan tabel *ORDER* sekarang berada di dalam 1NF, seperti yang ditunjukkan:

ORDER (ORDER, DATE, PRODUCT NUMBER, DESCRIPTION, NUMBER ORDERED, SUPPLIER NUMBER, SUPPLIER NAME, ISO)

Gambar 2.17 memperlihatkan tabel *ORDER* dalam 1NF. Perhatikan bahwa ketika grup berulang dihapus, catatan tambahan akan muncul satu untuk setiap kombinasi produk dan pesanan yang ditentukan. Hasilnya lebih banyak catatan tetapi desain disederhanakan. Dalam versi baru, grup pengulangan untuk nomor urut 86223 menjadi tiga catatan terpisah dan grup pengulangan untuk nomor urut 86390 menjadi dua catatan terpisah. Oleh karena itu, saat tabel berada di 1NF, setiap catatan menyimpan data tentang versi pesanan tertentu dan produk tertentu.

Perhatikan juga bahwa desain 1NF pada Gambar 2.17 memiliki kombinasi kunci. Kunci utama dalam desain 1NF tidak dapat hanya berada di bidang *ORDER* saja, karena nomor pesanan tidak secara unik mengidentifikasi setiap produk dalam pesanan *multi-item*.

Demikian pula, *PRODUCT NUMBER* tidak dapat menjadi kunci utama karena muncul lebih dari satu kali jika beberapa pesanan menyertakan pesanan yang sama. Karena setiap catatan harus mencerminkan produk tertentu di dalam kategori pesanan tertentu, dan bidang *ORDER* dan *PRODUCT NUMBER* diperlukan untuk mengidentifikasi catatan secara unik. Oleh karena itu, kunci utama adalah kombinasi dari dua bidang: *ORDER* dan *PRODUCT NUMBER*. [27]



Gambar 2.17 Tabel *Order* Normalisasi Pertama

2.10.4.3 Bentuk Normal Kedua

Untuk memahami bentuk normal kedua (2NF), konsep ketergantungan fungsional harus dipahami. Misalnya, bidang A bergantung secara fungsional pada bidang B. Seperti pada Gambar 2.17, nilai *DATE* secara fungsional bergantung pada *ORDER* karena untuk sejumlah pesanan tertentu, hanya ada satu tanggal. Deskripsi produk di sisi lain tidak tergantung pada nomor urut. Untuk pesanan tertentu, mungkin terdapat beberapa deskripsi produk, satu untuk setiap barang yang dipesan. Desain tabel berada dalam bentuk normal kedua (2NF) jika berada dalam 1NF dan semua bidang yang bukan bagian dari kunci utama secara fungsional bergantung pada seluruh kunci utama. Jika bidang dalam tabel 1NF hanya bergantung pada salah satu bidang dalam kombinasi kunci utama, tabel tersebut tidak dalam bentuk normal kedua (2NF).

Perhatikan bahwa jika desain 1NF memiliki kunci utama yang terdiri dari satu bidang, masalah ketergantungan parsial tidak akan muncul karena seluruh kunci utama adalah satu bidang. Oleh karena itu, tabel 1NF dengan satu bidang *primary key* secara otomatis adalah 2NF.

Sekarang tinjau desain 1NF untuk tabel *ORDER* seperti yang ditunjukkan pada Gambar 2.17:
ORDER (ORDER, DATE, PRODUCT NUMBER, DESCRIPTION, NUMBER ORDERED, SUPPLIER NUMBER, SUPPLIER NAME, ISO)

Ingatlah bahwa kunci utama adalah kombinasi dari nomor urut dan nomor produk. Bidang *NUMBER ORDERED* bergantung pada seluruh kunci utama karena *NUMBER ORDERED* mengacu pada nomor produk tertentu dan nomor pesanan tertentu. Di sisi lain, bidang *DATE* tergantung pada nomor urut dan itu hanya bagian dari kunci utama. Demikian pula, bidang *DESCRIPTION* bergantung pada nomor produk, yang juga hanya merupakan bagian dari kunci utama. Karena beberapa bidang tidak tergantung pada semua kunci utama, desainnya tidak berada dalam 2NF.

Ada prosedur standar untuk mengonversi tabel dari 1NF ke 2NF. Tujuannya adalah untuk membagi tabel asli menjadi dua atau lebih tabel baru dan menetapkan ulang bidang sehingga setiap bidang bukan kunci bergantung pada semua kunci utama dalam tabel itu.

Untuk mencapai ini, langkah-langkah berikut harus diikuti:

1. Buat dan berikan nama tabel terpisah untuk setiap bidang kunci primer yang ada. Misalnya, pada Gambar 2.17 *primary key* tabel *ORDER* memiliki dua *field* yaitu *ORDER* dan *PRODUCT NUMBER*, jadi dua tabel harus dibuat. Elipsis (. . .) menunjukkan bahwa bidang akan ditentukan nanti. Hasilnya adalah:

ORDER (ORDER, . . .)

PRODUCT (PRODUCT NUMBER, . . .)

2. Buat tabel baru untuk setiap kemungkinan kombinasi bidang kunci primer asli. Pada contoh Gambar 2.17, sebuah tabel baru akan dibuat dengan kombinasi *primary key* *ORDER* dan *PRODUCT NUMBER*. Tabel ini menjelaskan baris-baris individual dari sebuah perintah, karenanya dinamakan *COMMAND LINE*, seperti yang diilustrasikan:

ORDER LINE (ORDER, PRODUCT NUMBER)

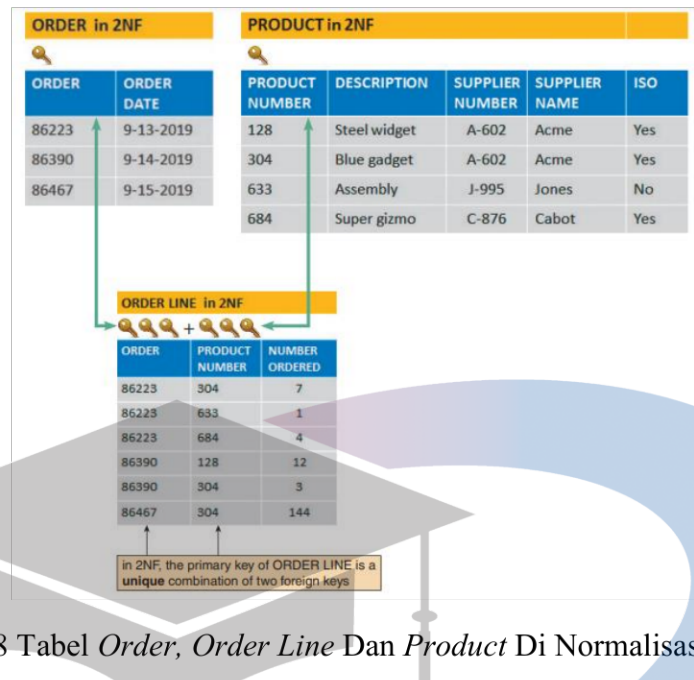
3. Lakukan pemeriksaan terhadap tiga tabel dan tetapkan masing-masing bidang kunci utama yang sesuai, yang merupakan kunci minimum tergantung pada fungsinya. Setelah semua bidang ditentukan, hapus semua tabel yang bidang tambahannya tidak ditentukan. Tabel yang tersisa adalah versi 2NF dari tabel asli. Ketiga tabel tersebut dapat direpresentasikan sebagai berikut:

ORDER (ORDER, DATE)

PRODUCT (PRODUCT NUMBER, DESCRIPTION, SUPPLIER NUMBER, SUPPLIER NAME, ISO)

ORDER LINE (ORDER, PRODUCT NUMBER)

Gambar 2.18 menunjukkan desain tabel 2NF. Dengan mengikuti langkah-langkah tersebut, tabel 1NF asli diganti dengan tiga tabel 2NF. [27]



Gambar 2.18 Tabel *Order*, *Order Line* Dan *Product* Di Normalisasi Kedua

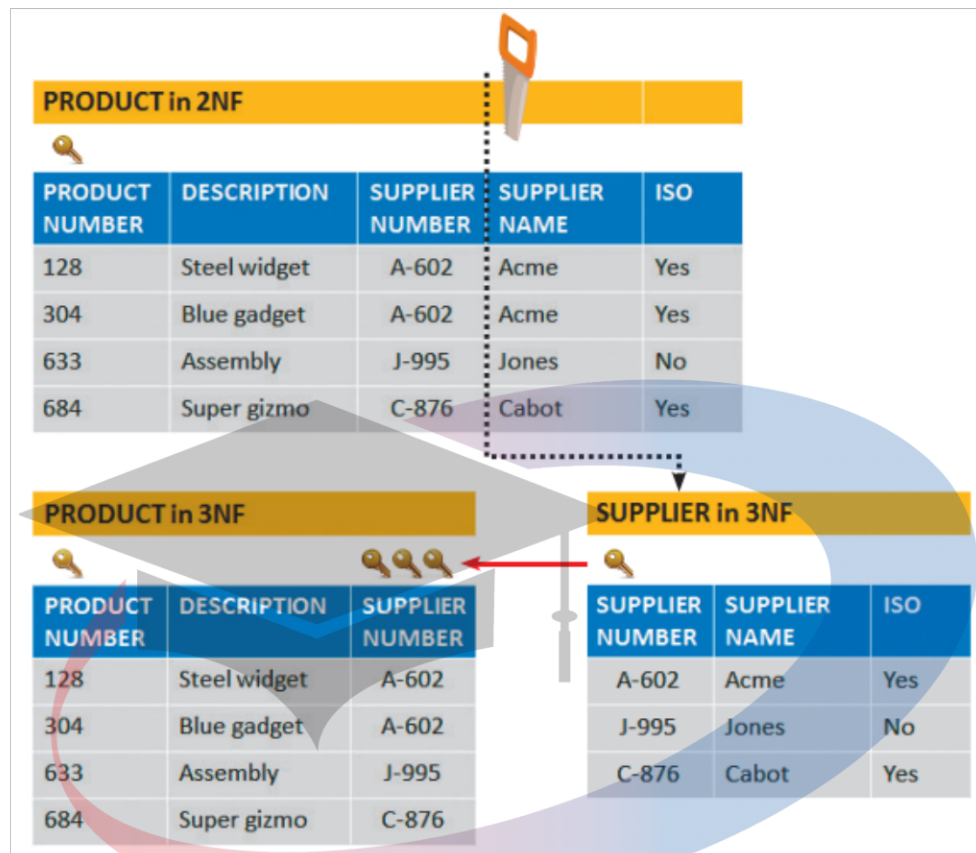
2.10.4.4 Bentuk Normal Ketiga

Aturan umum suatu desain dalam 3NF adalah jika setiap bidang non-kunci bergantung pada kunci, seluruh kunci, dan tidak lain yaitu kuncinya. Desain 3NF menghindari redundansi dan masalah integritas data masih bisa ada dalam desain 2NF.

Melanjutkan contoh *ORDER*, sekarang tinjau desain tabel *PRODUCT* di Gambar 2.19 :

UNIVERSITAS
MIKROSKIL

PRODUCT (PRODUCT NUMBER, DESCRIPTION, SUPPLIER NUMBER, SUPPLIER NAME, ISO)



Gambar 2.19 Tabel *Product* Dan *Supplier* Di Normalisasi Ketiga

Tabel *PRODUCTS* berada di dalam 1NF karena tidak memiliki grup berulang. Tabel tersebut juga ada di dalam 2NF karena kunci utamanya merupakan bidang tunggal. Namun tabel tersebut masih memiliki empat potensi masalah:

1. Untuk mengubah *SUPPLIER NAME*, semua catatan dengan nama tersebut harus diubah. Dengan ratusan atau bahkan ribuan catatan, pemrosesan akan menjadi lambat dan kesalahan *input* dapat terjadi.
2. Desain 2NF memungkinkan *SUPPLIER* memiliki nama atau status *ISO* dalam catatan yang berbeda.
3. Karena *SUPPLIER NAME* disertakan dalam tabel *ORDER*, catatan *ORDER* harus dibuat untuk menambahkan pemasok baru yang belum menerima pesanan apa pun.
4. Jika semua pesanan pemasok dihapus, nomor dan nama pemasok akan hilang. Masalah-masalah potensial tersebut disebabkan karena desainnya tidak berada dalam 3NF. Tabel berada dalam bentuk normal ketiga (3NF) jika berada dalam 2NF dan tidak ada bidang

nonkunci yang bergantung pada bidang nonkunci lainnya. Bidang nonkunci adalah bidang yang bukan merupakan kunci kandidat untuk kunci utama.

Tabel *PRODUCT* pada Gambar 2.19 bukan 3NF karena terdapat dua bidang nonkunci yaitu *SUPPLIER NAME* dan *ISO*, keduanya bergantung pada bidang nonkunci lainnya, *SUPPLIER NUMBER*.

Untuk mengonversi tabel menjadi 3NF, semua bidang dalam tabel 2NF yang bergantung pada bidang nonkunci lainnya harus dihapus dan ditempatkan di tabel baru dengan menggunakan bidang nonkunci sebagai kunci utama. Pada contoh *PRODUCT*, *SUPPLIER NAME* dan *ISO* harus dihapus dan ditempatkan pada tabel baru dengan menggunakan *SUPPLIER NUMBER* sebagai *primary key*.

Seperti yang ditunjukkan pada Gambar 2.19, 3NF membagi versi 2NF menjadi dua tabel 3NF yang terpisah: [27]

PRODUCT (PRODUCT NUMBER, DESCRIPTION, SUPPLIER NUMBER)
SUPPLIER (SUPPLIER NUMBER, SUPPLIER NAME, ISO)

2.10.5 Kerangka PIECES

Kerangka PIECES merupakan kerangka yang dipakai untuk mengklasifikasikan suatu permasalahan, peluang dan arahan yang terdapat pada *scope definition* analisis dan perancangan sistem. Kerangka PIECES dapat menghasilkan hal-hal baru yang dapat dijadikan sebagai pertimbangan dalam mengembangkan sistem.

Terdapat enam komponen utama yang digunakan untuk menganalisis sistem informasi yaitu:

1. *Performance* (keandalan)

Performance berperan penting dalam proses pengamatan untuk mengetahui kinerja sistem informasi dalam mengolah data untuk menghasilkan informasi dan mencapai tujuan yang diharapkan. Terdapat dua komponen utama yang dibuat sebagai acuan dalam proses evaluasi kinerja sistem yaitu:

- a) Apakah suatu sistem mampu mengerjakan perintah dalam waktu yang telah ditentukan tanpa ada hambatan.
- b) Seberapa cepat kemampuan sistem dalam merespon perintah atau permintaan dari suatu transaksi.[33]

2. *Information and Data* (Data dan informasi)

Penyajian data dan informasi merupakan hal yang penting demi kemajuan suatu perusahaan. [33]

Dalam sebuah temuan data pasti akan menghasilkan sebuah informasi yang akan ditampilkan, Variabel ini digunakan untuk menganalisis seberapa banyak dan jelas informasi yang didapat dari sebuah pencarian.[34] Adapun komponen yang perlu diperhatikan dalam proses evaluasi sistem yang berkaitan dengan data dan informasi yaitu:

- a) Keluaran (*Output*), adalah hasil yang didapat dari sistem yang menyajikan informasi yang dibutuhkan oleh perusahaan.
- b) Masukan (*Input*), merupakan kinerja sistem dalam memasukkan data sehingga kemudian di olah untuk menjadi informasi yang berguna.
- c) Data yang disimpan (*Stored Data*), adalah kehandalan dari sistem dalam proses penyimpanan data dan proses pengaksesan data.[33]

3. *Economics* (Nilai Ekonomi)

Variabel *economics* digunakan dalam menganalisis sistem untuk mengetahui apakah sistem tersebut dapat diterapkan pada perusahaan dilihat dari segi finansial dan biaya yang dikeluarkan. Ini merupakan hal yang penting karena sistem akan dipengaruhi oleh besarnya biaya yang dikeluarkan perusahaan.[34] Terdapat dua komponen yang perlu diperhatikan pada proses evaluasi sistem dari variabel *economics* yaitu:

- a) Biaya, evaluasi dari biaya yang digunakan dalam penerapan sistem informasi.
- b) Keuntungan, evaluasi yang dilakukan terhadap keuntungan yang didapat dalam penerapan sistem informasi.[33]

4. *Control and Security* (Pengamanan dan Pengendalian)

Variabel *Control and Security* merupakan proses pengamanan dan control yang dilakukan agar sistem tersebut terhindar dari gangguan yang tidak di inginkan dan dapat berjalan dengan baik. Terdapat hal yang perlu diperhatikan dalam pengendalian dan pengamanan sistem informasi, yaitu:

- a) Pengamanan dan pengendalian kepada sistem yang tidak kuat.
- b) Pengamanan dan pengendalian kepada sistem yang terlalu kompleks.[33]

5. *Efficiency* (Efisiensi)

Efisiensi dan efektifitas dari sistem perlu dipertanyakan pada kinerja dan alasan mengapa sistem tersebut dibuat. Sebuah sistem harus dapat menjawab secara efisien serta membantu suatu permasalahan yang ada khususnya pada hal otomasi. Analisis ini

dilakukan untuk mengetahui apakah sebuah sistem tersebut efisien, dengan memiliki *input* yang sedikit bias tetapi menghasilkan *output* yang memuaskan. [34]

6. *Service* (Pelayanan)

Pelayanan konsumen merupakan hal yang penting untuk selalu diperhatikan. Maksud dari konsumen disini ialah pengguna sistem informasi di mana pengguna sistem informasi harus diberikan pelayanan yang baik sehingga pengguna tersebut puas dan tidak beralih ke pesaing bisnis.[33] Bukan hanya itu sistem yang diterapkan dapat berjalan dengan baik dan seimbang jika memiliki dengan pelayanan yang baik juga. Analisis ini digunakan untuk mengetahui permasalahan yang terkait tentang pelayanan dan mengetahui bagaimana pelayanan yang dilakukan.[34] Terdapat hal yang perlu diperhatikan dalam proses mempertahankan konsumen yaitu:

- a) Sistem harus dapat menghasilkan informasi yang sesuai dengan kebutuhan pengguna dan juga akurat.
- b) Hasil dari sebuah sistem harus konsisten.
- c) Penerapan sistem harus mudah dipelajari, mudah di mengerti dan mudah digunakan.
- d) Sistem harus kompatibel dan fleksibel.[33]

2.10.6 Database

Sistem *database* merupakan sistem yang terkomputerisasi yang memiliki tujuan untuk memelihara data yang sudah dikelola atau informasi dan membuat sebuah informasi tersedia saat dibutuhkan. [28]

Database menyediakan kerangka kerja yang secara umum yang menghindari redundansi data dan mendukung lingkungan yang dinamis dan *real-time*. [27] Sistem informasi tidak akan lepas dengan kebutuhan *database* dalam bentuk apapun baik dalam bentuk *file teks* ataupun *database management system* (DBMS). [28]

Sistem manajemen basis data (DBMS) adalah kumpulan alat, fungsi, dan antarmuka yang memungkinkan pengguna untuk menambah, memperbaiki, mengelola, mengakses, dan menganalisis data. Dari perspektif pengguna, keuntungan utama DBMS adalah menyediakan akses data yang tepat waktu, interaktif, dan fleksibel.

Adapun keuntungan dari DBMS meliputi :

1. *Scalability*

Skalabilitas berarti bahwa sistem dapat dengan mudah diperluas, dimodifikasi, atau disederhanakan untuk memenuhi kebutuhan organisasi bisnis yang berubah dengan cepat. Misalnya, jika perusahaan memutuskan untuk menambahkan informasi tentang pemasok sekunder bahan yang digunakannya, tabel baru dapat ditambahkan ke database relasional dan ditautkan ke *database* relasional dengan bidang umum.

2. *Economy of scale*

Desain basis data memungkinkan penggunaan perangkat keras yang lebih baik. Jika sebuah perusahaan mengelola basis data seluruh perusahaan, pemrosesan dengan bantuan server dan jaringan komunikasi yang kuat akan lebih murah. Efisiensi yang melekat dalam memproses volume besar pada komputer yang lebih besar disebut skala ekonomi.

3. *Enterprise-wide application*

Sistem DBMS biasanya dikelola oleh seseorang yang dikenal sebagai administrator basis data (DBA). DBA mengevaluasi persyaratan umum dan memelihara *database* untuk kepentingan seluruh organisasi untuk kepentingan satu departemen atau pengguna. Sistem basis data dapat mendukung aplikasi di seluruh perusahaan secara lebih efisien daripada sistem manajemen *file*.

4. *Stronger standards*

Manajemen basis data yang efektif membantu memastikan standar penamaan, pemformatan, dan dokumentasi data diikuti secara konsisten di seluruh organisasi.

5. *Better security*

DBA dapat menetapkan prosedur otorisasi untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses *database* dan dapat memberikan tingkat akses yang berbeda kepada pengguna yang berbeda. Sebagian besar sistem DBMS menawarkan dukungan keamanan data yang diperluas.

6. *Data independence*

Sistem yang berinteraksi dengan DBMS relatif tidak bergantung pada bagaimana data fisik dikelola. Desain ini memberi DBA fleksibilitas untuk mengubah struktur data tanpa mengubah sistem informasi yang menggunakan data tersebut.[27]