

BAB III

LANDASAN TEORI

3.1 Ketentuan Kendaraan Dinas

Ketentuan Lampiran Pasal 17 angka Kendaraan Dinas Operasional/Kendaraan Dinas Jabatan dalam Peraturan Menteri Dalam Negeri Nomor 7 Tahun 2006 tentang Standarisasi Sarana dan Prasarana Kerja Pemerintahan Daerah, diubah sehingga berbunyi sebagai berikut:

Tabel 3.1 Kendaraan Dinas Operasional/Kendaraan Dinas Jabatan

No.	Jabatan	Jumlah	Jenis kendaraan	Kapasitas/isi silinder (maksimal)
1.	Ketua DPRD Provinsi	1 (satu) unit	Sedan atau Jeep	2.700 cc
2.	Wakil Ketua DPRD Provinsi	1 (satu) unit	Sedan atau Minibus	2.500 cc
3.	Ketua DPRD Kabupaten/Kota	1 (satu) unit	Sedan atau Minibus	2.500 cc
4.	Wakil Ketua DPRD Kabupaten/Kota	1 (satu) unit	Sedan atau Minibus	2.200 cc
5.	Pejabat Eselon I	1 (satu) unit	Sedan atau Jeep	2.700 cc
6.	Pejabat Eselon II	1 (satu) unit	Sedan atau Minibus(bensin) Minibus(solar)	2.000 cc 2.500 cc
7.	Pejabat Eselon IU	1 (satu) unit	Minibus(bensin) Minibus(solar)	1.600 cc 2.500 cc
8.	Pejabat Eselon N dan Eselon V	1 (satu) unit	Sepedah Motor	200 cc

Yang bersumber dari Peraturan Menteri Dalam Negeri Nomor 11 Tahun 2007, tentang perubahan atas peraturan menteri dalam negeri nomor 7 tahun 2006 tentang standarisasi sarana dan prasarana kerja pemerintahan daerah.

3.2 Pemeliharaan Barang atau Aset Daerah

3.2.1. Pemeliharaan dilakukan terhadap barang inventaris yang sedang dalam pemakaian maupun barang persediaan, berupa :

- Pemeliharaan ringan yaitu pemeliharaan yang mengakibatkan pembebanan anggaran merupakan kegiatan penggantian dari sebagian aset berupa rehabilitasi dan restorasi, dengan maksud meningkatkan umur / masa manfaat, mempertahankan kapasitas dan mutu produksi, sehingga tidak menambah nilai aset.
- Pemeliharaan berat yaitu pemeliharaan yang mengakibatkan pembebanan anggaran merupakan kegiatan penggantian aset berupa renovasi dengan maksud meningkatkan umur/masa manfaat, kapasitas, mutu produksi dan standar kinerja sehingga menambah nilai aset.

3.2.2. Untuk menghindari penurunan kemampuan produktifitas barang, agar dapat berfungsi sebagaimana mestinya, maka Kepala SKPD dapat mengalokasikan biaya pemeliharaan terhadap barang Inventaris yang sudah tercatat Daftar Barang Pengguna (DBP) dengan berdasarkan pada hasil inventarisasi yang telah dilaksanakan pada tahun sebelumnya, serta skala prioritas pelaksanaan pemeliharaan barang.

3.2.3. Biaya pemeliharaan barang inventaris yang menjadi obyek pinjam pakai ditanggung oleh SKPD yang bersangkutan atau pemakai sesuai dengan perjanjian.

3.2.4. Ketentuan dan Pelaksanaan Pemeliharaan Bangunan Rumah Golongan I, II, dan III yang dikelola oleh SKPD mengacu pada Peraturan Gubernur Jawa Timur tanggal 1 Juli 2005, Nomor 24 tahun 2005 tentang Pedoman dan Tata Cara Penggunaan Rumah Daerah Pemerintah Provinsi Jawa Timur.

3.2.5. Perbaikan/Penggantian Suku Cadang Kendaraan Dinas :

- Pengajuan perbaikan kendaraan dinas khusus untuk Sekretariat Daerah Provinsi Jawa Timur harus melalui Biro Umum Sekretariat Daerah Provinsi Jawa Timur.
- Untuk kendaraan Dinas Operasional yang berlokasi di Kabupaten/Kota dan keadaan fisiknya tidak mungkin dapat dijalankan ke Surabaya

untuk diperbaiki di Surabaya, dapat diperbaiki di Daerah dengan ketentuan :

- 1)Harus ada Surat permohonan/SPK Perbaikan dari Pejabat yang berwenang di Instansinya ;
- 2)Harus ada surat keterangan pemeriksaan dari UPT Dinas Perhubungan dan LLAJ Provinsi yang ada di Daerah ;
- 3)Setelah selesai perbaikan harus diperiksa kembali oleh UPT Dinas Perhubungan dan LLAJ Provinsi;
- 4)Bengkel yang menangani perbaikan harus memiliki Kualifikasi Teknis, Peralatan, Tenaga Teknis, dan syarat administratif antara lain :

- Surat Ijin Usaha
- NPWP (Nomor Pokok Wajib Pajak).

- Khusus Dinas Pekerjaan Umum Bina Marga Provinsi Jawa Timur dengan berlakunya Peraturan Daerah Provinsi Jawa Timur Nomor 9 Tahun 2008 tentang Organisasi dan Tata Kerja Dinas Daerah Provinsi Jawa Timur, dimana Bidang Pemeliharaan Dinas Pekerjaan Umum Bina Marga Provinsi Jawa Timur mempunyai/membawahi Unit Bengkel yang memadai, maka perbaikan kendaraan dinas/penggantian suku cadang khusus untuk poin 5b, tidak diberlakukan bagi Dinas Pekerjaan Umum Bina Marga Provinsi Jawa Timur.
- Khusus Dinas Pekerjaan Umum Pengairan Provinsi Jawa Timur, sehubungan dengan terbentuknya Depo Peralatan pada Dinas Pekerjaan Umum Pengairan Provinsi Jawa Timur berdasarkan Peraturan Daerah Provinsi Jawa Timur Nomor 9 Tahun 1996 tentang Pembentukan Organisasi dan Tata Kerja Unit Pelaksana Teknis Lingkup Dinas Pekerjaan Umum Pengairan Provinsi Jawa Timur, maka perbaikan kendaraan dinas/penggantian suku cadang untuk poin 5b tidak diberlakukan pada Dinas Pekerjaan Umum Pengairan Provinsi Jawa Timur dan dilaksanakan oleh Depo Peralatan pada Dinas Pekerjaan Umum Pengairan Provinsi Jawa Timur.

3.2.6. Panitia/pejabat penerima hasil pekerjaan :

- Panitia/pejabat penerima hasil pekerjaan di lingkungan Sekretariat Daerah Provinsi Jawa Timur dibentuk oleh Gubernur Jawa Timur;
- Untuk Pemeriksaan Pemeliharaan/Perbaikan Barang pada Badan/Dinas/Lembaga yang menggunakan dana APBD dilaksanakan oleh 1 (satu) Panitia/pejabat penerima hasil pekerjaan yang dibentuk oleh Kepala SKPD yang bersangkutan;
- Susunan keanggotaan Panitia Pemeriksa Pemeliharaan/Perbaikan Barang terdiri dari unsur-unsur di dalam SKPD dan dapat melibatkan unsur teknis terkait sesuai kebutuhan.

3.2.7. Pelaksanaan Pemeriksaan Pemeliharaan/Perbaikan Barang :

- Seluruh hasil pelaksanaan Pemeliharaan/Perbaikan Barang harus diperiksa oleh Panitia Pemeriksa Pemeliharaan Barang, dengan cara membandingkan antara kondisi fisik barang serta jumlah barang dengan spesifikasi pekerjaan yang tercantum dalam dokumen kontrak yang telah ditandatangani oleh Pejabat Pembuat Komitmen dan penyedia jasa;
- Hasil pemeriksaan Pemeliharaan/perbaikan barang dituangkan dalam Berita Acara Pemeriksaan Hasil Pemeliharaan/Perbaikan Barang.

3.2.8. Penerimaan hasil pemeliharaan barang inventaris :

- Penerimaan hasil pemeliharaan barang inventaris termasuk cleaning service dan pemeliharaan taman yang bernilai sampai dengan Rp. 5.000.000,00 (lima juta rupiah), dilakukan oleh Pengurus Barang/Pembantu Pengurus Barang SKPD yang bersangkutan.
- Penerimaan hasil pemeliharaan barang inventaris termasuk cleaning service dan pemeliharaan taman yang bernilai di atas Rp.5.000.000,00 (lima juta rupiah) dilakukan oleh Pengurus Barang/Pembantu Pengurus Barang, setelah terlebih dahulu diperiksa oleh Panitia Pemeriksa Pemeliharaan dan Perbaikan Barang Inventaris pada SKPD yang bersangkutan dan hasil pemeriksaan pekerjaan tersebut dituangkan dalam Berita Acara Pemeriksaan Pemeliharaan dan Perbaikan Barang Inventaris.

- Dalam rangka efisiensi dan efektifitas pelaksanaan pemeliharaan/perbaikan barang pada UPTD, maka penerimaan hasil pemeliharaan/perbaikan barang inventaris termasuk cleaning service dan pemeliharaan taman dengan nilai antara Rp. 5.000.000,00 sampai dengan Rp. 15.000.000,00 dilakukan oleh Pembantu Pengurus Barang setelah terlebih dahulu diperiksa oleh Pejabat Pemeriksa Pemeliharaan dan Perbaikan Barang Inventaris pada UPTD yang telah ditunjuk oleh Kepala SKPD. Pejabat Pemeriksa dimaksud berasal dari salah satu pejabat struktural pada UPTD yang bersangkutan. Hasil Pemeriksaannya dituangkan dalam Berita Acara Pemeriksaan dan Pemeliharaan Barang Inventaris yang ditandatangani oleh Pejabat Pemeriksa dan Pihak Penyedia Jasa.

3.2.9. Setiap SKPD harus mencatat semua hasil pemeliharaan pada Kartu Pemeliharaan Barang Inventaris berdasarkan jenis barang. Dalam Kartu Pemeliharaan barang inventaris tersebut memuat data-data sebagai berikut:

- a) Nomor Urut;
- b) Nama barang Inventaris;
- c) Kode lokasi / Kode barang;
- d) Nomor registrasi;
- e) Nomor kendaraan;
- f) Jenis pemeliharaan/yang memelihara;
- g) Tanggal pemeliharaan;
- h) Biaya pemeliharaan;
- i) Bukti pemeliharaan;
- j) Keterangan.

3.2.10. Kepala SKPD bertanggung jawab untuk membuat Daftar Hasil Pemeliharaan Barang dalam lingkup pengelolaannya dan wajib melaporkan/menyampaikan daftar hasil pemeliharaan barang yang mengacu pada DKPBD (Daftar Kebutuhan Pemeliharaan Barang Daerah) kepada Gubernur melalui Kepala Biro Umum setiap 3 (tiga) bulan sesuai format terlampir, dengan jadwal sebagai berikut :

- a) Laporan daftar hasil pemeliharaan barang Triwulan I (1 Januari s/d 31 Maret) dan rekapitulasi per bidang barangnya disampaikan paling lambat tanggal 15 April tahun berjalan.
- b) Laporan daftar hasil pemeliharaan barang Triwulan II (1 April s/d 30 Juni) dan rekapitulasi per bidang barangnya disampaikan paling lambat tanggal 15 Juli tahun berjalan.
- c) Laporan hasil pemeliharaan barang Triwulan III (1 Juli s/d 30 September) dan rekapitulasi per bidang barangnya disampaikan paling lambat tanggal 15 Oktober tahun berjalan.
- d) Laporan daftar hasil pemeliharaan barang Triwulan IV (1 Oktober s/d 31 Desember) dan rekapitulasi per bidang barangnya disampaikan paling lambat tanggal 15 Januari tahun berikutnya, serta membuat rekapitulasi dalam 1 (satu) tahun anggaran.

Untuk UPTD laporan disampaikan pada Dinas masing-masing dengan jadwal sebagai berikut :

Triwulan I paling lambat tanggal 10 April 2011;

Triwulan II paling lambat tanggal 10 Juli 2011;

Triwulan III paling lambat tanggal 10 Oktober 2011;

Triwulan IV paling lambat tanggal 10 Januari 2012.

3.2.11. Hasil pemeliharaan dapat dikapitalisasi menambah nilai barang yang dipelihara apabila setelah dilakukan Pemeliharaan memenuhi kriteria sebagai berikut :

- a) Memperpanjang masa manfaat ;
- b) Meningkatkan kapasitas
- c) Meningkatkan mutu produksi, atau
- d) Meningkatkan standar kinerja.

3.3 Konsep Dasar Sistem Informasi

Menurut Jogianto Hartono (1990), terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu sistem yang menekankan pada prosedurnya dan menekankan pada prosedur mendefinisikan suatu sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran yang tertentu.

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*component*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*inputan*), keluaran (*output*), pengolah (*process*), dan sasaran (*objective*) atau tujuan (*goal*).

Komponen sistem merupakan bagian-bagian dari sistem yang saling berhubungan dan menjadi satu kesatuan. Komponen-komponen sistem atau sub-sistem ini memiliki karakteristik tersendiri dan menjalankan suatu fungsi tersendiri. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan *supra system* (Jogianto Hartono, 1990), suatu misal sebuah perusahaan dapat disebut sebagai sistem dan industri yang merupakan sistem yang lebih besar dapat disebut sebagai *supra system*.

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

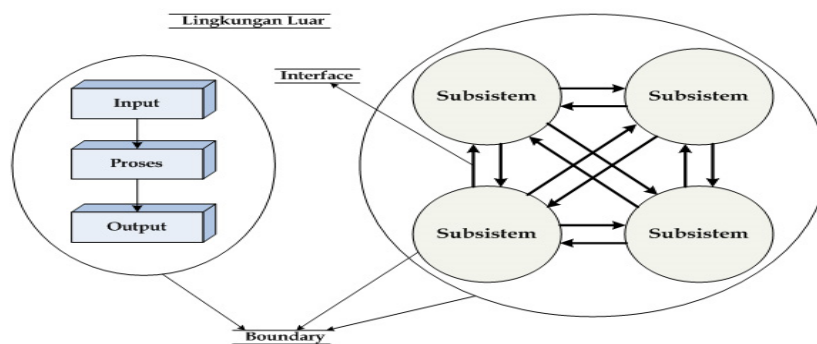
Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar sistem yang merugikan harus ditahan dan dikendalikan, agar tidak mengganggu kehidupan dari sistem itu sendiri (Jogianto Hartono, 1990).

Penghubung (*interface*) merupakan media penghubung antara satu sub-sistem dengan sub-sistem yang lainnya. Melalui penghubung ini memungkinkan sumber daya-sumber daya mengalir dari suatu sub-sistem ke sub-sistem yang lainnya. Keluaran (*output*) dari suatu sub-sistem akan menjadi masukan (*input*) untuk sub-sistem yang lainnya melalui penghubung (*interface*). Dengan penghubung (*interface*), suatu sub-sistem dapat berintegrasi dengan sub sistem lainnya untuk membentuk suatu kesatuan.

Masukan (*input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa sinyal atau berupa masukan perawatan. Masukan sinyal adalah energi yang dimasukkan yang nantinya akan diolah dan menghasilkan sesuatu. Sedangkan masukan perawatan adalah energi yang digunakan untuk melakukan suatu proses atau dengan kata lain energi yang menjamin suatu proses dapat berjalan. Keluaran sistem dapat dibedakan menjadi dua yaitu keluaran yang berguna dan sisa pembuangan. Keluaran dapat dijadikan sebagai masukan dari sub-sistem yang lainnya.

Pengolah sistem (*process*) adalah bagian dari setiap sistem dan sub-sistem yang akan mengolah masukan sehingga menjadi keluaran (*output*), baik yang berguna maupun menjadi sisa. Suatu sistem pasti mempunyai tujuan (*goal*) ataupun sasaran (*objective*) yang ingin dicapai. Jika suatu sistem tidak memiliki sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran sistem sangat menentukan apa yang diperlukan serta keluaran apa yang dihasilkan. Suatu sistem dikatakan berhasil jika mengenai sasaran yang ingin dicapai.

Karakteristik dari suatu sistem dapat digambarkan dalam bagan sebagai berikut :



Gambar 3.1 Karakteristik suatu sistem

Informasi dapat diibaratkan sebagai darah dalam suatu makhluk hidup. Informasi memberi suatu semangat, motivasi, dan gairah dalam suatu organisasi. Tanpa adanya informasi, organisasi tersebut akan lesu, kerdil, dan akhirnya akan berhenti. Menurut Jogianto Hartono, informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi itu sendiri adalah data, yang merupakan jamak dari bentuk tunggal *datum*. Data adalah kenyataan yang menggambarkan suatu keadaan nyata. Data juga merupakan bentuk yang masih mentah dan belum dapat bercerita banyak, sehingga membutuhkan pengolahan yang lebih lanjut. Kualitas dari system informasi bergantung pada dua hal, yaitu:

1. Informasi harus akurat, dimana informasi itu harus bebas dari kesalahan.
2. Informasi tersebut harus relevan, supaya informasi tersebut dapat memberikan masukan bagi penerimanya.

3.4 Analisis dan Perancangan Sistem

Menurut Kendall dan Kendall (2003), Analisis sistem dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi.

Analisa dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.

Berikut ini adalah proses dalam analisis dan perancangan sistem:

1. Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi, dapat abstrak dan

nyata. Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. Relasi adalah hubungan antar *entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar *entity*.

2. Data Flow Diagram

Pada tahap ini, penggunaan notasi dapat membantu komunikasi dengan pemakai/user sistem untuk memahami sistem tersebut secara logika. Diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem ini dikenal dengan nama Diagram Arus Data (*Data Flow Diagram*). DFD berfungsi untuk menggambarkan proses aliran data yang terjadi di dalam sistem dari tingkat yang tertinggi sampai yang terendah, yang memungkinkan untuk melakukan dekomposisi, mempartisi atau membagi sistem kedalam bagian-bagian yang lebih kecil dan yang lebih sederhana.

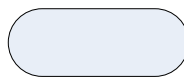
DFD fokus pada aliran data dari dan ke dalam sistem serta memproses data tersebut (Kendall, 2003).

3.5 Sistem Flow

System Flow adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem dimana bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem dan biasanya dalam membuat *system flow* sebaiknya ditentukan pula fungsi-fungsi yang melaksanakan atau bertanggung jawab terhadap sub-sistem yang aa (Jogianto, 1998:10).

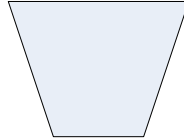
Terdapat berbagai bentuk symbol yang digunakan untuk merancang sebuah desain dari sistem, diantaranya adalah *terminator*, *manual operation*, *document*, *process*, *database*, *manual input*, *off-line storage*, *on-page reference*, dan *off-page reference*.

Terminator merupakan bentuk symbol yang digunakan sebagai tanda dimulainya jalan proses ataupun tanda akhir dari sebuah pengerjaan suatu sistem. Simbol dari *terminator* dapat dilihat pada gambar 3.2.



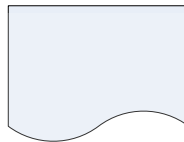
Gambar 3.2 Terminator

Manual operation digunakan untuk menggambarkan sebuah proses kerja yang digunakan tanpa menggunakan computer sebagai medianya (menggunakan proses manual). Simbol dari *manual operation* dapat dilihat pada gambar 3.3.



Gambar 3.3 *Manual Operation*

Document merupakan simbol dari dokumen yang berupa kertas laporan, surat-surat, memo, maupun arsip-arsip secara fisik. Simbol dari *document* dapat dilihat pada gambar 3.4.



Gambar 3.4 *Document*

Process adalah sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi. Simbol dari *process* dapat dilihat pada gambar 3.5.



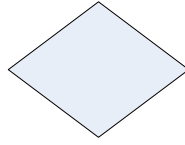
Gambar 3.5 *Process*

Database digunakan sebagai media penyimpanan data yang bersifat terkomputerisasi. Simbol dari *database* dapat dilihat pada gambar 3.6.

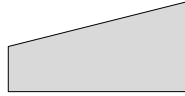


Gambar 3.6 *Database*

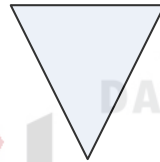
Decision merupakan operator logika yang digunakan sebagai penentu keputusan dari suatu permintaan atau proses dengan dua nilai, benar atau salah. Simbol dari *decision* dapat dilihat pada gambar 3.7.

Gambar 3.7 *Decision*

Manual input digunakan untuk melakukan proses *input* kedalam *database* melalui *keyboard*. Simbol dari *manual input* dapat dilihat pada gambar 3.8.

Gambar 3.8 *Manual Input*

Off-line storage merupakan bentuk media penyimpanan yang berbeda dengan *database*, dimana media penyimpanan ini menyimpan dokumen secara manual atau lebih dikenal dengan nama arsip. Simbol dari *off-line storage* dapat dilihat pada gambar 3.9.

Gambar 3.9 *Off-Line Storage*

On-page reference digunakan sebagai symbol untuk menghubungkan bagan desain sebuah sistem apabila hubungan arus data yang ada terlalu jauh dalam permasalahan letaknya. Simbol dari *on-page reference* dapat dilihat pada gambar 3.10.

Gambar 3.10 *On-Page Reference*

Off-page reference memiliki sifat yang sedikit berbeda dengan *on-page reference*, karena simbol ini hanya digunakan apabila arus data ada dilanjutkan ke halaman yang berbeda. Simbol dari *off-page reference* dapat dilihat pada gambar 3.11.

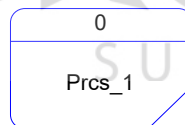
Gambar 3.11 *Off-Page Reference*

3.6 Data Flow Diagram (DFD)

Menurut (Kristanto, 2004:12), Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data tersebut disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut.

Data Flow Diagram merupakan suatu metode pengembangan sistem yang terstruktur (*structure analysis and design*). Penggunaan notasi dalam data flow diagram sangat membantu untuk memahami suatu sistem pada semua tingkat kompleksitas. Pada tahap analisis, penggunaan notasi ini dapat membantu dalam berkomunikasi dengan pemakai sistem untuk memahami sistem secara logika.

Di dalam data flow diagram terdapat empat simbol yang digunakan yaitu *process*, *external entity*, *data store*, dan *data flow*. Simbol *process* digunakan untuk melakukan suatu perubahan berdasarkan data yang diinputkan dan menghasilkan data dari perubahan tersebut. Simbol *process* dapat dilihat pada gambar 3.12.

Gambar 3.12. *Simbol Process*

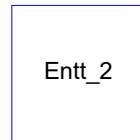
Pada bentuk gambar *process*, bagian atas berisi nomor untuk identitas proses. Suatu proses dengan nomor 0 (nol atau kosong) menandakan bahwa proses tersebut adalah sebuah *context diagram*. Diagram ini merupakan level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya. Pembuatan *context diagram* dapat dilakukan dengan terlebih dahulu menentukan nama sistemnya, menentukan batasan dari sistem, dan menentukan

terminator yang diterima atau diberikan daripada sistem untuk kemudian dilakukan penggambaran.

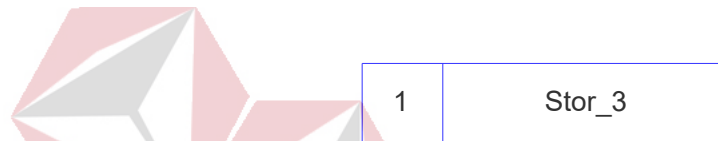
Nomor 1, 2, 3, dan seterusnya menandakan bahwa proses tersebut diartikan sebagai proses level-0 (nol) yang merupakan hasil turunan atau *decompose* dari proses *context diagram*. Proses level-0 membahas sistem secara lebih mendetil, baik dipandang dari segi kegiatan dari sebuah bagian, alur data yang ada, maupun *database* yang digunakan didalamnya. Pembuatannya dapat dilakukan dengan cara menentukan proses utama yang ada dalam sistem, menentukan alur data yang diterima dan diberikan masing-masing proses daripada sistem sambil memperhatikan konsep keseimbangan (alur data yang masuk atau keluar dari suatu level harus sama dengan alur data yang masuk dan keluar pada level berikutnya), memunculkan *data store* sebagai sumber maupun tujuan data (*optional*), menggambarkan diagram level-0, menghindari perpotongan arus data, dan melakukan pemberian nomor pada proses utama (nomor tidak menunjukkan urutan proses).

Nomor 1.1, 1.2, 2.1, 2.2, dan seterusnya merupakan sebuah proses turunan atau *decompose* dari proses level-0 yang disebut sebagai proses level-1 (satu). Proses level-1 menggambarkan detil kerja dari sebuah bagian dalam sebuah sistem. Penggambarannya dilakukan dengan cara menentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level-0, menentukan apa yang diterima atau diberikan masing-masing sub-proses daripada sistem dan tetap memperhatikan konsep keseimbangan, memunculkan *data store* sebagai sumber maupun tujuan alur data (*optional*), menggambar DFD level-1, dan berusaha untuk menghindari perpotongan arus data. Hasil turunan akhir disebut sebagai *the lowest level*, dimana hasil akhir ini tergantung dari kompleksitas sistem yang ada.

- a. *External entity* disimbolkan dengan bentuk persegi yang digunakan untuk menggambarkan pelaku-pelaku sistem yang terkait, dapat berupa orang-orang, organisasi maupun instansi. *External entity* dapat memberikan masukan kepada *process* dan mendapatkan keluaran dari *process*. *External entity* dapat dilihat pada gambar 3.13.

Gambar 3.13 *Simbol External Entity*

- b. *Data store* digunakan sebagai media penyimpanan suatu data yang dapat berupa *file* atau *database*, arsip atau catatan manual, lemari *file*, dan tabel-tabel dalam *database*. Penamaan *data store* harus sesuai dengan betuk data yang tersimpan pada *data store* tersebut, misalnya tabel pelamar, tabel pendidikan, tabel lulus seleksi, dan lain-lain. *Data store* dilihat pada gambar 3.14.

Gambar 3.14 *Simbol Data Store*

- c. *Data flow* merupakan penghubung antara *external entity* dengan *process* dan *process* dengan *data store*. *Data flow* menunjukkan aliran data dari satu titik ke titik lainnya dengan tanda anak panah mengarah ke tujuan data. Penamaan *data flow* harus menggunakan kata benda, karena di dalam *data flow* mengandung sekumpulan data. *Data flow* dilihat pada gambar 3.15.

Gambar 3.15 *Simbol Data Flow*

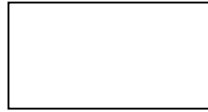
3.7 Entity Relationship Diagram (ERD)

Proses *reverse engineering* terhadap suatu basis data menjadi suatu kebutuhan bagi perancang basis data untuk mengetahui struktur dari sebuah basis data. Struktur tersebut biasanya dimodelkan dalam bentuk *Entity Relationship Diagram* (ERD).

ERD dibagi menjadi dua macam yaitu: *Conceptual Data Model* (CDM), dan *Physical Data Model* (PDM). Simbol-simbol yang sering digunakan adalah:

1. Entity

Entity merupakan sesuatu yang mudah diidentifikasi. Sebuah *entity* bisa berupa obyek, tempat, orang, konsep, atau aktivitas. *Entity* dinyatakan dalam simbol persegi panjang. Simbol *entity* pada gambar 3.16.



Gambar 3.16 Entity.

2. Atribut

Atribut merupakan penjelasan-penjelasan dari *entity* yang membedakan *entity* satu dengan yang lain. Sebuah atribut juga merupakan sifat-sifat dari sebuah *entity*. Atribut dinyatakan dalam simbol ellipsis. Simbol atribut pada gambar 3.17.



Gambar 3.17 Atribut.

3. Relationship

Relationship adalah penghubung antara suatu *entity* dengan *entity* yang lain dan merupakan bagian yang sangat penting di dalam mendesain *database*. Ada tiga tipe *relationship* yang dikenal yaitu :

a. One-to-One Relationship

Jenis hubungan antar tabel yang menggunakan secara bersama sebuah kolom *primary key*. Jenis hubungan ini tergolong jarang digunakan, kecuali untuk alasan keamanan atau kecepatan akses data. Seperti departemen hanya mengerjakan satu jenis pekerjaan saja dan satu pekerjaan hanya dikerjakan oleh satu departemen saja.

b. One-to-Many Relationship

Jenis hubungan antar tabel dimana satu *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain. Jenis hubungan ini yang paling sering

digunakan. Misalnya satu pekerjaan hanya dikerjakan oleh satu departemen saja, namun satu departemen dapat mengerjakan beberapa macam pekerjaan sekaligus.

c. *Many-to-Many Relationship*

Jenis hubungan antar tabel dimana beberapa *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain. Misalnya satu departemen mampu mengerjakan banyak pekerjaan, juga satu pekerjaan dapat ditangani oleh banyak departemen.

d. *Many-to-One Relationship*

Jenis hubungan antar tabel dimana beberapa *record* pada satu tabel terhubung dengan satu *record* pada tabel lain. Misalnya satu departemen mampu mengerjakan banyak pekerjaan, namun satu pekerjaan hanya dikerjakan oleh satu departemen saja.

Menurut Sutanta (2004), relasi antar entitas dapat digambarkan melalui salah satu gambar 3.18 dan gambar 3.19 dari pilihan berikut ini:

1. Pilihan 1

Jenis relasi

Simbol yang digunakan

1-ke-1 :



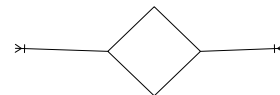
1-ke-n :



n-ke-1 :

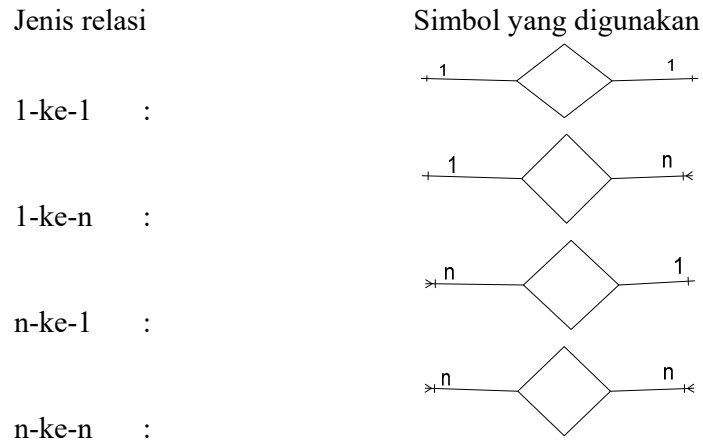


n-ke-n :



Gambar 3.18 Simbol relasi antar entitas (pilihan 1).

2. Pilihan 2



Gambar 3.19 Simbol relasi antar entitas (pilihan 2).

ERD dapat digambarkan menggunakan salah satu dari pilihan di atas, namun penggunaannya harus konsisten. Jika menggunakan simbol pilihan 1, maka untuk seluruh bagian ERD harus menggunakan simbol kelompok pilihan 1.

3. Kunci relasi

Kunci relasi atau *key* adalah suatu properti yang menentukan apakah suatu kolom pada table sangat penting atau tidak. Berdasarkan macamnya, kunci relasi terdiri dari:

a. Kunci kandidat

Yaitu satu atau atau gabungan minimal atribut yang bersifat unik yang dapat digunakan untuk mengidentifikasi setiap *record* dalam relasi.

b. Kunci primer

Yaitu bagian atau salah satu dari kunci kandidat yang digunakan sebagai kunci utama untuk membedakan setiap *record* dalam relasi. Kunci primer biasa disebut sebagai *primary key*.

c. Kunci alternatif

Yaitu bagian dari kunci kandidat yang tidak digunakan sebagai kunci utama.

d. Kunci penghubung

Kunci penghubung atau *foreign key* yaitu satu atau gabungan sembarang atribut yang menjadi kunci utama dalam relasi lain yang mempunyai hubungan secara logik. Kunci penghubung dan kunci utama harus memiliki tipe dan ukuran data yang sama.

3.8 Kebutuhan Aplikasi dan Bahasa Pemrograman.

3.8.1 Web Text Editor

Pertama dalam pembuatan web, memerlukan sebuah aplikasi *HTML*, *code authoring* yang dapat mempermudah kita dalam membuat maupun mendesain web. Dalam pembuatan web kali ini software yang digunakan adalah Macromedia Dreamweaver CS 4.

Macromedia Dreamweaver CS 4 adalah sebuah program aplikasi yang menyediakan segala kebutuhan untuk membuat sebuah web yang utuh dan mudah dengan adanya tool-tool yang sudah terintegrasi dalam program dreamweaver.

3.8.2 Web Server

Dalam pembuatan web secara offline dibutuhkan sebuah web server local (*localhost*). Hal ini berguna untuk mensimulasikan sebuah server sebenarnya (online). Pada web server akan diletakkan file beserta databasenya.

Web server yang digunakan kali ini adalah XAMPP, di dalamnya sudah berisi MySQL, Apache dan phpadmin yang merupakan komponen-komponen utama untuk sever local.

3.8.3 Web Browser

Web Browser merupakan software untuk membuka halaman web atau melihat web yang di buat. Web yang telah dibuat, sebelum di *publish* secara online atau dalam tahap pembuatan web, maka di cek terlebih dahulu dengan *view* melalui web browser akan terlihat hasil/ tampilan web yang dibuat.

Web browser yang dipakai kali ini adalah Mozilla Firefox versi terbaru untuk saat ini adalah versi 6.0.

3.8.4 Bahasa Pemrograman PHP

PHP adalah bahasa pemrograman yang memungkinkan para web *developer* untuk membuat aplikasi web yang dinamis dengan cepat dan mudah. PHP merupakan singkatan dari “PHP : Hypertext Preprocessor”. PHP dirintis dan diperkenalkan pertama kali sekitar tahun 1994 oleh Rasmus Lerdorf melalui

situsnya untuk mengetahui siapa saja yang telah mengakses ringkasan *online*-nya.

PHP adalah bahasa *server-side scripting* yang bisa menyatu dengan tag-tag HTML. *Server-side scripting* adalah sintaks dan perintah-perintah yang dijalankan pada server dan disertakan pada dokumen HTML pada suatu halaman web dinamis, PHP berfungsi sebagai bahasa pemrograman yang menjalankan suatu perintah tertentu, sedangkan HTML berfungsi sebagai struktur dari desain halaman web. Ketika sebuah halaman web dinamis dibuka pada web browser, pertama kali yang terjadi adalah server memproses semua perintah PHP yang ada kemudian menampilkan hasilnya pada format HTML ke web browser, sehingga yang ditampilkan ke web browser hanya tampilan desain dari HTML, sedangkan script PHP bekerja di belakang layar.

PHP merupakan software open source, jadi PHP bisa digunakan oleh siapa saja secara bebas tanpa harus membayar, dengan kata lain PHP adalah gratis. PHP juga merupakan software cross platform, jadi bisa berjalan dengan baik pada windows maupun (linux).

Kelebihan utama dari PHP adalah konektivitasnya dengan database PHP maupun mengolah data pada berbagai platform database misalnya MySQL yang merupakan database yang open source.

3.8.5 Database MySQL

MySQL adalah sistem untuk mengelola database atau manajemen data. Untuk menyimpan data dan informasi ke computer kita menggunakan data, contoh kita menyimpan data karyawan pada suatu perusahaan dan memasukkan suatu file. File data yang dikelompokkan inilah yang disebut database, dan MySQL bertugas mengatur dan mengelola data-data pada database. Dalam mengelola database MySQL menggunakan struktur atau kerangka yang berbentuk table. Dalam table-table itulah data di atur dan dikelompokkan.

Keunggulan MySQL adalah termasuk open source, dikenal sebagai sistem database yang efisien dan reliable, proses query cepat dan mudah. MySQL juga mudah dihubungkan dengan berbagai bahasa pemrograman dan aplikasi, sehingga tingkat kompatibilitasnya tinggi disamping itu kode-kode perintah MySQL sederhana sehingga dapat mudah dipahami.

3.9 Interaksi Manusia dan Komputer

Interaksi Manusia dan Komputer dideskripsikan sebagai sebuah disiplin ilmu yang mempelajari desain, evaluasi, implementasi dari system computer interaktif untuk dipakai oleh manusia, beserta studi tentang factor-faktor utama dalam lingkungan (Rizky, 2007). Deskripsi IMK menurut Rizky (2007) adalah suatu ilmu yang mempelajari perencanaan dan desaintentang cara manusia dan computer saling bekerjasama sehinggal manusia merasa puas dengan cara yang paling efektif.

Menurut Rizky (2007), komponen-komponen penting dalam IMK yaitu interaksi, manusia, dan Komputer. Interaksi adalah komunikasi yang terjadi antara manusia dan komponen. Jenis-jenis komunikasi tersebut antara lain *command entry, menus and navigation, forms and spreadsheets, question and answer dialogue, natural language dialogue, windows icon menu pointer*, dan *direct manipulation*. Komponen selanjutnya yaitu manusia yang dalam hal ini adalah pengguna yang sangat dapat berupa seseorang ataupun sekelompok pengguna yang bekerja dalam sebuah tim atau organisasi dan saling berkeitan dalam mengerjakan tugas tertentu. Manusia dalam konteks IMK merupakan faktor utama yang perlu diperhatikan dalam konteks psikologi yang disebut *cognitive psychology*. Komponen terakhir dalam IMK yang juga harus diperhatikan adalah komputer. Komputer diartikan sebagai perangkat keras ataupun perangkat lunak dari berbagai macam jenis yang nantinya akan berinteraksi dengan unsur manusia.

Rizky (2007) menjelaskan bahwa sebelum memulai sebuah proses desain interface, terdapat beberapa tip desain yang harus diperhatikan, antara lain :

a) Memenuhi kaidah estetika

Sebuah desain dapat disebut baik secara estetika jika (1) didalamnya terdapat perbedaan yang jelas dan kontras antara elemen dalam sebuah tampilan, misalnya tampilan tombol yang berbeda warna dengan tampilan textbox, (2) terdiri dari beberapa kelompok yang jelas antara inouran dan tombol proses, (3) antara elemen dan kelompok tampilan dipisah dengan *alignment* yang rapi, (4) sederhana dan tidak terlalu banyak aksesoris(Gambar, animasi, *icon*) yang terkesan sia-sia.

b) Dapat dimengerti

Sebuah desain harus dapat dimengerti dengan cepat dari segi tampilan secara visual, fungsi yang akan ditonjolkan, penggunaan kata-kata yang singkat dan jelas baik dalam tampilan maupun dalam perintah. Penggunaan metafora atau pemisalan yang berlebihan dalam sebuah fungsi harus dihindari.

c) Kompatibilitas

Sebuah desain *interface* harus dapat memenuhi kompatibilitas dari berbagai segi antara lain (1) kompatibilitas pengguna yaitu dapat digunakan oleh pengguna dari kalangan yang lebih luas, baik berdasarkan strata pendidikan maupun berdasarkan usia, (2) kompatibilitas penggunaan yaitu dapat memenuhi fungsi dan tujuan yang ingin dicapai dari perancangan sebuah perangkat lunak dan perangkat keras yang digunakan, (3) kompatibilitas produk yaitu agar perangkat lunak dapat berjalan dengan baik di berbagai perangkat keras yang ada dan sistem operasi yang menjadi target aplikasi.

d) Komprehensif

Sebuah system yang baik akan membimbing penggunanya agar dapat dan lebih mudah memahami apa yang harus diperhatikan, bagaimana cara melakukan sesuatu, kapan dan di mana melakukan sesuatu, dan mengapa harus melakukan sesuatu.

e) Kongfigurabilitas

Sebuah sistem juga harus dapat dikonfirmasi ulang jika pengguna mengingatkan sesuatu berdasarkan fungsi tertentu.

f) Konsistensi

Pengguna dapat melakukan kontrol jika suatu saat terjadi kesalahan dalam proses serta pemilihan fungsi tambahan dari sebuah sistem. Hindari desain yang nantinya akan membatasi pengguna dalam memilih tampilan tertentu.

g) Kontrol pengguna

Pengguna dapat melakukan control jika suatu saat terjadi kesalahan dalam proses serta pemilihan fungsi tambahan dari sebuah sistem. Hindari desain yang nantinya akan membatasi pengguna dalam memilih tampilan tertentu.

h) Efisien

Desain dibuat seefisien mungkin, terutama dalam penempatan komponen, misalnya penempatan tombol dalam sebuah panel yang dapat menarik perhatian pengguna.

i) Mudah dikenali

Gunakan antar muka yang sudah dikenal oleh pengguna, misalnya penempatan *icon Cut, Copy, Paste* secara standar dalam sebuah *toolbar*.

j) Toleransi

Tidak ada sebuah system yang sempurna, karenanya terdapat beberapa toleransi untuk kesalahan yang mungkin terjadi. Usahakan agar terjadi sebuah pesan yang dapat membimbing pengguna untuk keluar dari kesalahan yang terjadi.

k) Sederhana

Lima cara untuk membuat desain sederhana dan tetap sesuai dengan keinginan pengguna, yaitu (1) sembunyikan komponen visual jika tidak diperlukan, (2) sediakan pilihan standar atau *default*, (3) minimalkan penggunaan berbagai macam *alignment*, (4) usahakan agar fungsi yang sering digunakan terlihat, (5) perhatikan konsep konsistensi.

