

## Data Flow Diagram (DFD)

### 1. Definisi

**Data Flow Diagram (DFD)** adalah alat yang digunakan dalam pemodelan sistem untuk menggambarkan aliran data dalam suatu sistem. DFD membantu memvisualisasikan bagaimana data berpindah dari satu bagian sistem ke bagian lain, bagaimana data tersebut diproses, dan di mana data disimpan. Ini adalah alat yang sangat berguna dalam pengembangan sistem informasi, analisis sistem, dan perancangan perangkat lunak.

### 2. Komponen Utama DFD

- **External Entity (Entitas Eksternal)**

- ✓ Definisi: Entitas eksternal adalah sumber atau tujuan dari data yang berada di luar sistem yang dimodelkan. Mereka adalah bagian dari lingkungan sistem dan berinteraksi dengan sistem melalui pengiriman atau penerimaan data.
- ✓ Simbol: Entitas eksternal diwakili oleh kotak persegi panjang.
- ✓ Contoh: Pelanggan, pemasok, sistem eksternal lain.

- **Process (Proses)**

- ✓ Definisi: Proses adalah kegiatan atau operasi yang mengubah input menjadi output. Setiap proses menerima data dari entitas eksternal, data store, atau proses lain, kemudian mengolahnya, dan menghasilkan output yang baru.
- ✓ Simbol: Proses biasanya digambarkan dengan lingkaran atau oval.
- ✓ Contoh: Menghitung total penjualan, memproses pesanan, verifikasi pembayaran.

- **Data Flow (Aliran Data)**

- ✓ Definisi: Aliran data menunjukkan arah dan jenis data yang mengalir dari satu komponen ke komponen lain dalam sistem. Aliran data menghubungkan entitas eksternal, proses, dan penyimpanan data.
- ✓ Simbol: Aliran data digambarkan dengan garis panah, di mana arah panah menunjukkan ke mana data mengalir.
- ✓ Contoh: Data pesanan, informasi pelanggan, laporan penjualan.

- **Data Store (Penyimpanan Data)**

- ✓ Definisi: Penyimpanan data adalah tempat di mana data disimpan untuk digunakan kemudian. Ini dapat berupa basis data, file, atau bentuk lain dari penyimpanan informasi yang digunakan oleh proses dalam sistem.
- ✓ Simbol: Penyimpanan data digambarkan dengan dua garis paralel atau sebuah bentuk terbuka.
- ✓ Contoh: Basis data pelanggan, arsip transaksi, tabel produk.

### 3. Tingkatan DFD

DFD sering dibuat dalam beberapa tingkatan untuk memberikan pandangan yang lebih rinci tentang sistem:

- **Level 0 DFD (Context Diagram)**

- ✓ Deskripsi: DFD tingkat 0 adalah representasi paling sederhana dari sistem, menggambarkan keseluruhan sistem sebagai satu proses tunggal yang berinteraksi dengan entitas eksternal. Ini memberikan pandangan tingkat tinggi tentang aliran data ke dalam dan ke luar sistem tanpa rincian proses internal.
- ✓ Contoh: Sistem pemesanan online digambarkan sebagai satu proses yang menerima data pesanan dari pelanggan dan mengirimkan konfirmasi pesanan.

- **Level 1 DFD**

- ✓ Deskripsi: DFD tingkat 1 memperlihatkan rincian lebih lanjut dari proses yang ada di DFD tingkat 0. Proses utama dipecah menjadi sub-proses yang lebih rinci, menunjukkan bagaimana data mengalir di dalam sistem dan antara sub-proses
- ✓ Contoh: Sistem pemesanan online dapat dipecah menjadi proses-proses seperti memproses pesanan, mengelola inventaris, dan mengelola pembayaran.

- **Level 2 DFD dan Seterusnya**

- ✓ Deskripsi: Pada level ini, setiap sub-proses dari DFD level 1 dipecah lagi menjadi proses-proses yang lebih kecil, memberikan rincian lebih lanjut. Level 2 dan seterusnya digunakan jika diperlukan detail yang lebih mendalam untuk setiap bagian dari sistem.
- ✓ Contoh: Proses memproses pesanan dipecah lagi menjadi verifikasi ketersediaan barang, pencatatan pesanan, dan pengiriman pesanan.

### 4. Langkah-langkah Membuat DFD

- **Identifikasi Entitas Eksternal:**

- ✓ Tentukan siapa atau apa yang akan berinteraksi dengan sistem, baik sebagai sumber maupun tujuan data.

- **Identifikasi Proses Utama:**

- ✓ Gambarkan proses-proses utama yang mengubah data dari entitas eksternal menjadi output yang berbeda.

- **Tentukan Aliran Data:**

- ✓ Gambar aliran data antara entitas eksternal, proses, dan data store. Pastikan semua data yang masuk ke suatu proses memiliki output yang jelas.

- **Identifikasi Data Store:**

- ✓ Tentukan tempat-tempat di mana data akan disimpan dalam sistem dan bagaimana proses akan berinteraksi dengan data tersebut.

- **Validasi DFD:**

- ✓ Periksa DFD untuk memastikan konsistensi dan kelengkapan. Setiap aliran data harus masuk atau keluar dari suatu proses, dan semua proses harus memiliki input dan output yang jelas.

## **5. Manfaat Menggunakan DFD**

- **Pemahaman Sistem yang Lebih Baik:** DFD membantu tim pengembang dan pemangku kepentingan untuk memahami alur kerja dalam sistem dan interaksi antara berbagai komponen
- **Dokumentasi Sistem:** DFD adalah alat yang sangat baik untuk mendokumentasikan desain sistem secara grafis, yang membantu dalam komunikasi dan pengembangan lebih lanjut.
- **Identifikasi Kebutuhan Sistem:** DFD membantu dalam mengidentifikasi kebutuhan sistem dan potensi masalah sebelum pengembangan dimulai.

Dengan DFD, proses pengembangan sistem menjadi lebih terstruktur dan mudah dipahami, sehingga memudahkan dalam pembuatan desain sistem yang efektif dan efisien.