DFD (Data Flow Diagram) adalah diagram yang digunakan untuk menggambarkan aliran data dalam suatu sistem. DFD memperlihatkan bagaimana data berpindah dari satu proses ke proses lain dalam sistem, serta bagaimana data tersebut diproses, disimpan, dan berinteraksi dengan entitas luar.

### Elemen Utama DFD:

1. \*\*Proses\*\*: Menunjukkan transformasi data. Biasanya digambarkan sebagai lingkaran atau oval.

2. \*\*Aliran Data\*\*: Menunjukkan perpindahan data antara elemen-elemen dalam sistem. Digambarkan sebagai panah.

3. \*\*Penyimpanan Data (Data Store)\*\*: Tempat penyimpanan data dalam sistem. Digambarkan sebagai dua garis horizontal.

4. \*\*Entitas Eksternal\*\*: Entitas yang berada di luar sistem namun berinteraksi dengan sistem, seperti pengguna atau sistem lain. Biasanya digambarkan sebagai persegi panjang.

### Jenis-Jenis DFD:

1. \*\*DFD Level 0 (Context Diagram)\*\*: Menggambarkan sistem secara keseluruhan, menampilkan semua entitas eksternal yang berinteraksi dengan sistem serta aliran data masuk dan keluar dari sistem.

2. \*\*DFD Level 1\*\*: Merinci proses-proses utama yang terjadi dalam sistem, menggambarkan aliran data di antara proses dan penyimpanan data.

3. \*\*DFD Level 2 dan seterusnya\*\*: Memecah proses di level yang lebih tinggi menjadi sub-proses yang lebih rinci.

DFD banyak digunakan dalam analisis sistem karena memberikan gambaran jelas tentang bagaimana data diproses dalam sistem tanpa perlu menjelaskan detail teknis atau implementasi.

Menurut buku \*Systems Analysis and Design\* karya Tilley dan Scott, \*\*DFD (Data Flow Diagram)\*\* digunakan sebagai alat untuk memodelkan sistem dengan memfokuskan pada aliran data di dalam sistem. DFD memberikan pandangan fungsional tentang bagaimana data diproses, disimpan, dan dipindahkan antara berbagai bagian dari sistem dan entitas eksternal.

### Elemen DFD menurut Tilley dan Scott:

1. \*\*Proses\*\*:

- Proses adalah tempat transformasi data dari input menjadi output. Digambarkan sebagai lingkaran atau oval.

- Setiap proses memiliki nama yang menunjukkan apa yang dilakukan, misalnya “Memproses Pembayaran.”

2. \*\*Aliran Data (Data Flow)\*\*:

- Aliran data adalah perpindahan data dari satu elemen ke elemen lain. Digambarkan sebagai panah yang menghubungkan proses, penyimpanan data, atau entitas eksternal.

- Data yang mengalir harus diberi label yang mewakili isi data, seperti “Data Pesanan.”

3. \*\*Penyimpanan Data (Data Store)\*\*:

- Penyimpanan data menunjukkan tempat data disimpan. Digambarkan sebagai dua garis paralel atau persegi panjang terbuka.

- Nama penyimpanan data menggambarkan isinya, seperti “Database Pelanggan.”

4. \*\*Entitas Eksternal\*\*:

- Entitas eksternal adalah pihak di luar sistem yang berinteraksi dengan sistem, seperti pelanggan atau pemasok. Digambarkan sebagai persegi panjang.

- Setiap entitas eksternal diberi nama sesuai peran atau entitas sebenarnya, seperti “Pelanggan.”

### Pendekatan DFD oleh Tilley dan Scott:

* \*\*Context Diagram (DFD Level 0)\*\*: Menggambarkan keseluruhan sistem sebagai satu proses utama yang berinteraksi dengan entitas eksternal. Pada level ini, hanya aliran data masuk dan keluar yang ditampilkan tanpa detail tentang proses internal.

* \*\*DFD Level 1\*\*: Memecah proses utama menjadi beberapa sub-proses untuk menunjukkan bagaimana data mengalir di dalam sistem, dengan lebih banyak detail dibandingkan context diagram. Di sini, penyimpanan data dan sub-proses mulai muncul.
* \*\*DFD Level 2 dan Seterusnya\*\*: Setiap sub-proses dari DFD Level 1 dapat diuraikan lebih lanjut untuk memberikan rincian lebih spesifik tentang setiap langkah dalam sistem.

Menurut Tilley dan Scott, \*\*DFD\*\* digunakan secara luas dalam fase analisis sistem karena:

- \*\*Memudahkan komunikasi\*\* antara analis dan pengguna akhir.

- \*\*Menghilangkan detail teknis\*\* yang tidak perlu pada awalnya.

- Menyediakan gambaran yang \*\*visual dan mudah dipahami\*\* tentang bagaimana data mengalir di dalam sistem.

Buku ini menekankan bahwa DFD tidak hanya menggambarkan aliran data, tetapi juga \*\*membantu dalam memahami fungsi sistem secara menyeluruh\*\* sehingga bisa digunakan dalam pengembangan sistem yang lebih terstruktur.