

## Summary Notes of Sesi 1.1 - Fondasi Pemrograman & Struktur Data – 01 Pengenalan SI dan Pemrograman.

Sebelum membahas tentang pemrograman ada baiknya kita mengetahui terlebih dahulu untuk apa pemrograman tersebut. Dalam bidang ilmu sistem informasi, pemrograman digunakan untuk mengambil data, mengolah data dan menghasilkan informasi. Mari kita lihat definisi-definisi terkait sistem informasi.

### Sistem

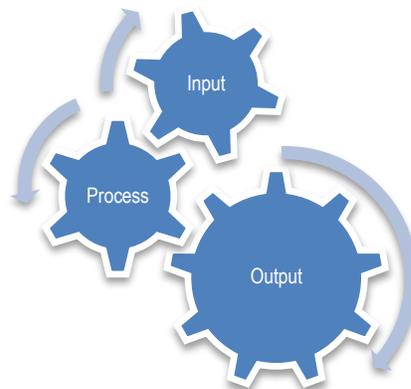
#### Definisi

Menurut **Gordon B. Davis**, Sistem adalah “**Kumpulan elemen yang saling berinteraksi dan bekerja sama untuk mencapai suatu tujuan**”.

Menurut **Raymond Mcleod**, Sistem adalah “Sistem adalah himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk satu kesatuan yang utuh dan terpadu”.

Menurut **Jerry FitzGerald**, Sistem adalah “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”.

Dari definisi yang dikemukakan para pakar tersebut, maka untuk bidang sistem informasi dapat pula kita definisikan sebagai berikut: “Kombinasi berbagai sumber daya yang bekerjasama dalam mengkonversikan masukan (input – input) menjadi keluaran (output – output)”. Berdasarkan definisi tersebut maka sistem dapat direpresentasikan dengan gambar 1.1.



Gambar 1.1 Sistem dalam representasi gambar.

Berdasarkan gambar 1.1 dapat kita lihat bahwa kinerja (effort) dari gear input yang bersentuhan dengan gear proses akan membawa dampak terhadap Bergeraknya gear proses, demikian

selanjutnya kinerja (effort) gear proses akan membawa dampak terhadap bergeraknya gear output.

Kinerja (effort) gear output sangat tergantung dari transformasi effort dari gear input melalui rasio atau besarnya gear proses. Cepat atau lambatnya dan kuat atau tidaknya torsi atau daya gear output sangat tergantung ukuran dari gear proses. Jika gear proses kecil maka perputaran gear output lebih lambat, namun torsi (power) gear output lebih kuat. Jika gear proses besar maka perputaran gear output lebih cepat, namun torsi (power) gear output lebih lemah. Dari penjelasan tersebut bisa dikatakan bahwa output sangat tergantung dari input dan proses.

### Model

Sistem secara umum terdapat dua model yaitu:

- **Sistem terbuka** → Sistem yang dalam operasionalnya masih terkait dengan lingkungan luar.

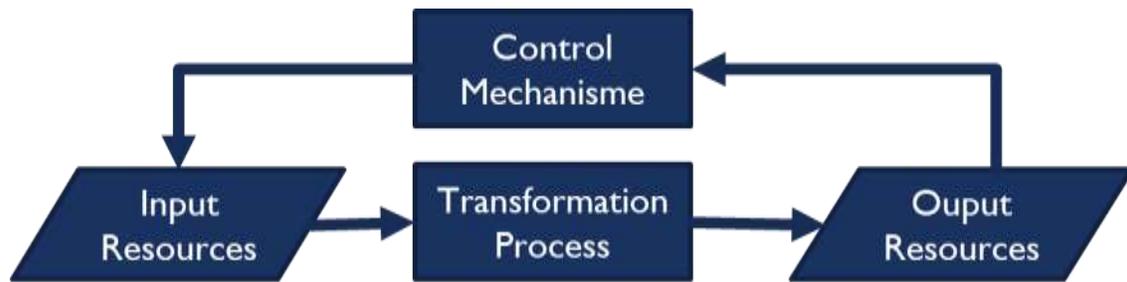


Gambar 1.2. Diagram alur sistem terbuka

Berdasarkan diagram alur di atas bisa dilihat bahwa elemen input akan menyuplai ke elemen proses dan elemen proses akan menyuplai ke elemen output. Untuk melakukan pengaturan agar mendapatkan output sesuai yang diharapkan sangat diperlukan intervensi dari lingkungan luar.

Sistem terbuka ini bisa kita analogikan (misalkan) seperti halnya **sistem pompa air yang masih manual**. Elemen atau komponen dari sistem tersebut adalah sumber air, pompa dan tangki air, sedangkan orang (manusia sebagai elemen diluar sistem). Berikut ini adalah mekanismenya:

1. Jika ingin mengisi air ke tangki maka harus ada orang yang memeriksa apakah tangki kosong, jika tangki kosong maka orang tersebut akan membuka sumber air dan menyalakan pompa sehingga air akan terisi ke tangki.
  2. Sebelum mematikan pompa harus ada orang yang memeriksa tangki air apakah tangki sudah penuh terisi air atau belum, jika tangki sudah terisi air maka orang tersebut akan mematikan pompa air dan menutup sumber air.
- **Sistem tertutup** → Sistem yang dalam operasionalnya tidak terkait dengan lingkungan luar.



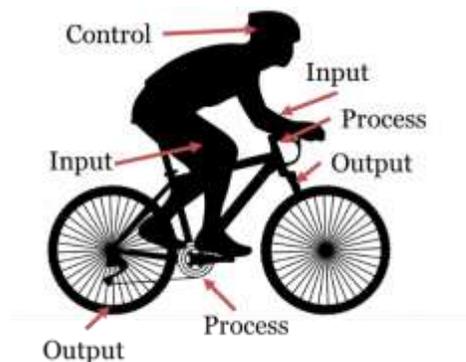
Gambar 1.3. Diagram alur sistem tertutup

Berdasarkan diagram alur di atas bisa dilihat bahwa elemen input akan menyuplai ke elemen proses, selanjutnya elemen proses akan menyuplai ke elemen output dan elemen output akan menyuplai ke elemen control, kemudian elemen control berdasarkan suplai dari elemen output akan menyuplai ke elemen input berupa kendali terhadap elemen input. Pengaturan agar mendapatkan output sesuai yang diharapkan dilakukan oleh elemen control tanpa intervensi dari lingkungan luar.

Sistem tertutup ini bisa kita analogikan (misalkan) seperti halnya **sistem pompa air yang otomatis**. Elemen atau komponen dari sistem tersebut adalah sumber air, pompa, tangki air dan sensor level air (sebagai komponen kontrol). Berikut ini adalah mekanismenya:

1. Jika level air pada tangki air kurang maka sensor sebagai komponen kontrol akan membuka katup sumber air dan mengakibatkan pompa aktif, selanjutnya air akan terisi ke tangki air
2. Jika level air pada tangki sudah tinggi (penuh) maka sensor sebagai komponen kontrol akan menutup katup sumber air dan mengakibatkan pompa tidak aktif, sehingga suplai air ke tangki air terhenti.

Selain analogi dengan sistem pompa air, berikut adalah analogi dengan sistem bersepeda:



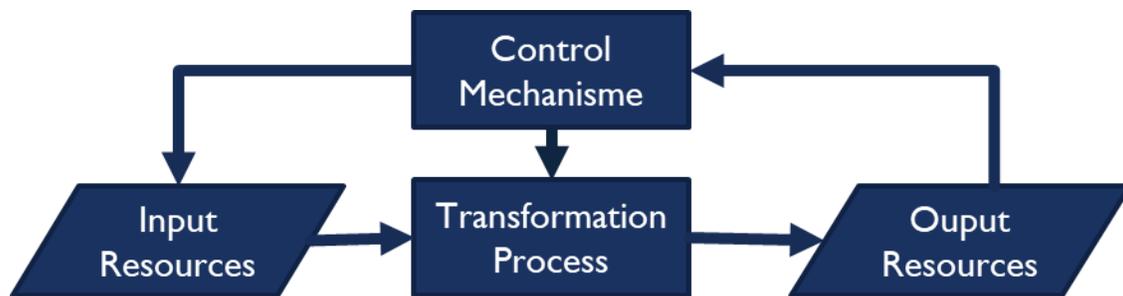
Gambar 1.4. Bersepeda sebagai analogi sistem

Pada sistem bersepeda terdapat dua sub sistem yaitu sistem untuk bergerak kedepan dan sistem untuk kendali (belok atau arah sepeda).

Elemen atau komponen dari sistem sub sistem untuk bergerak ke depan adalah Kaki (sebagai komponen input), pedal dan gear (sebagai komponen proses), roda (sebagai komponen output), dan otak atau pikiran pengendara (sebagai komponen control).

Elemen atau komponen dari sistem sub sistem untuk kendali (belok atau arah sepeda) adalah tangan (sebagai komponen input), setang (sebagai komponen proses), fork (sebagai komponen output), dan otak atau pikiran pengendara (sebagai komponen control).

Secara lengkap flow sistem dapat digambarkan seperti berikut:



Gambar 1.5. Diagram alur sistem dengan kontrol lengkap

Berdasarkan diagram alur pada gambar 1.5 terlihat bahwa komponen control tidak hanya mengontrol komponen input melainkan juga mengontrol komponen proses.

## Informasi

### Definisi

Menurut **Gordon B. Davis**, informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Menurut **Jogiyanto HM.**, informasi adalah hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian – kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan.

Menurut **Jogiyanto HM.** (2004:8) dalam bukunya yang berjudul Analisis dan Desain Sistem Informasi, berpendapat bahwa informasi adalah data yang diolah menjadi bentuk yang lebih berguna bagi yang menerimanya.

Menurut **Raymond Mc.leod**, informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Dengan demikian informasi juga dapat didefinisikan sebagai:

*“Data yang sudah diolah melalui suatu proses menjadi bentuk yang berguna dan bermanfaat bagi Pemakai”*

## Sistem Informasi

### Definisi

Menurut **Raymond Mc.leod**, sistem informasi adalah sistem yang berkemampuan dapat mengumpulkan serta mengelompokkan informasi dari berbagai sumber dengan menggunakan berbagai media sehingga dapat menampilkan informasi.

Menurut **Tafri D. Muhyuzir**, sistem informasi adalah data-data yang telah dikumpulkan, dikelompokkan kemudian diolah menjadi sebuah informasi yang bermanfaat dan bernilai bagi penerimanya.

Menurut **Lani Sidharta**, sistem informasi adalah sistem yang dibuat oleh manusia yang berisi serangkaian komponen terpadu dan manual bagian komponen terkomputerisasi sehingga dapat mengumpulkan dan mengolah data lalu menghasilkan informasi bagi pengguna.

Berdasarkan definisi-definisi tentang sistem informasi, sistem dan informasi maka dapat dikatakan sistem informasi adalah **“Kumpulan komponen/elemen yang saling berinteraksi; berfungsi untuk mengumpulkan, memproses, menyimpan dan mendistribusikan informasi untuk mendukung kegiatan operasi suatu organisasi (sistem)”**



Gambar 1.6. Alur Sistem transformasi data menjadi informasi

### Elemen-elemen Sistem Informasi

- Perangkat keras (hardware)
- Program / perangkat lunak (Software)
- Data
- Prosedur
- Personil / manusia

Berdasarkan elemen-elemen sistem informasi di atas dapat kita lihat bahwa program/perangkat lunak (*software*) merupakan elemen sistem informasi yang berfungsi untuk mengambil data, mengolah data dan menghasilkan informasi. Untuk selanjutnya pembahasan akan lebih difokuskan kepada pemrograman.

## Summary Notes of Sesi 1.2 - Fondasi Pemrograman & Struktur Data – 02 Program & Langkah Pengembangan Program.

Sebelum membahas tentang cara pengembangan program (cara membuat program) ada baiknya kita mengetahui terlebih dahulu bagaimana ciri-ciri program yang baik.

### Ciri-ciri Program Baik

- Maintainability
- Correctness
- Reusability
- Reliability
- Portability
- Efficiency

#### Maintainability

Maintainability mencerminkan kemudahan dari suatu program untuk dilakukan pengembangan atau perubahan agar memenuhi kebutuhan/persyaratan baru atau untuk memperbaiki kekurangan. Dengan demikian maka program yang baik harus:

- Dirancang dengan baik agar mampu mengakomodasi perubahan yang diperlukan dimasa depan.
- Memiliki dokumentasi yang baik agar memenuhi karakteristik maintainability.

#### Correctness

Correctness mencerminkan suatu tingkat dimana program memenuhi kebutuhan/persyaratan yang ditentukan, atau dengan kata lain seberapa benar program memenuhi kebutuhan/persyaratan yang ditentukan. Dengan demikian maka program yang baik harus:

- Dirancang dengan baik agar Memenuhi semua kebutuhan/persyaratan yang diminta/ditetapkan, selain itu program harus berjalan sesuai yang diharapkan.
- Dilakukan analisis user requirement (kebutuhan pengguna) sebagai langkah awal dalam merancang program.
- Diuji untuk memastikan bahwa program berjalan sesuai dengan yang diharapkan.

#### Reusability

Reusability mencerminkan kemudahan dimana program dapat digunakan kembali dalam mengembangkan perangkat lunak lainnya. Dengan demikian maka program yang baik harus:

- Dirancang dengan baik agar dapat dimanfaatkan oleh program lain untuk digunakan ulang (*reuse*).

- Menerapkan konsep modular dan berbasis *object*, agar memenuhi karakteristik *reusability* dalam membuat program.

### Reliability

Mencerminkan kehandalan program terhadap kegagalan yang mungkin terjadi pada kondisi operasi normal dan suatu kondisi yang dimungkinkan terjadi. Dengan demikian maka program yang baik harus:

- Dirancang dengan baik dan sudah mengantisipasi kemungkinan-kemungkinan kegagalan yang dapat terjadi pada suatu kondisi tertentu.
- Mempertimbangkan beberapa kondisi (dalam membuat program) yang memungkinkan terjadinya kegagalan (spt: human error, dsb) agar memenuhi karakteristik reliability.

### Portability

Mencerminkan kemampuan program untuk digunakan pada berbagai konfigurasi komputer. Dengan demikian maka program yang baik harus:

- Dirancang dengan baik dengan sudah mempertimbangkan portabilitas (kemampuan program untuk digunakan pada komputer/perangkat lain dengan konfigurasi yang berbeda)
- Mengarah pada konsep multiplatform agar memenuhi karakteristik portability.

### Efficiency

Mencerminkan kemampuan program untuk memenuhi tujuannya tanpa pemborosan sumber daya. Dengan demikian maka program yang baik harus:

- Dirancang dengan baik dengan mempertimbangkan unsur efisien dalam penggunaan sumber daya (seperti: ram, bandwidth, dsb).
- Mempertimbangkan penggunaan sumber daya se-efisien mungkin (seperti; alokasi variable, metode input, dsb.) dalam pembuatan program.
- Untuk memenuhi karakteristik Efficiency ini maka dalam membuat program harus mempertimbangkan untuk menggunakan sumber daya se-efisien mungkin (seperti; alokasi variable, metode input, dsb).

### Langkah–langkah Pengembangan Program

Berikut adalah langkah-langkah yang harus dilakukan dalam pengembangan atau pembuatan program:

1. Definisikan masalah
2. Rancang outline pemecahan masalah
3. Buat algoritma berdasarkan outline pemecahan masalah
4. Test algoritma
5. Coding
6. Execute
7. Dokumentasi dan pemeliharaan

### Definisi Masalah

Dalam pembuatan program, langkah awal yang harus dilakukan adalah pendefinisian masalah. Pendefinisian masalah ini akan menentukan *output* (keluaran) apa sebenarnya yang diharapkan oleh pengguna, selanjutnya akan ditentukan *input* (masukan) apa yang harus diberikan ke aplikasi agar menghasilkan *output* sesuai yang diharapkan, baru kemudian proses apa saja harus yang dilakukan oleh program untuk mengolah *input* sehingga menghasilkan *output* sesuai yang diharapkan.

Berikut adalah langkah-langkah dalam melakukan pendefinisian masalah:

1. Mendefinisikan output (keluaran)
2. Mendefinisikan *input* (Masukan)
3. Mendefinisikan *Process* (Proses)

Dalam melakukan pendefinisian masalah sesuai langkah di atas, bisa digunakan tabel sebagai berikut:

*Tabel 1.1. Tabel definisi masalah*

Input	Process	Output

Contoh proses definisi masalah:

- Misal, diketahui kebutuhan untuk membuat program bagi perhitungan sebagai berikut;  $C=A+B$  dan selanjutnya dilakukan perhitungan  $Z=X*C$ . Berdasarkan wawancara dengan pengguna didapat alur prosesnya sebagai berikut; Program akan membaca nilai A dan nilai B, selanjutnya dilakukan perhitungan untuk mendapatkan nilai C dan menampilkan nilai C. Selanjutnya program membaca nilai X dan melakukan perhitungan untuk mendapatkan nilai Z dan menampilkan nilai Z.

Analisis:

Berdasarkan narasi di atas diketahui bahwa; Output: C, dan Z dan Input: A, B, dan X. Sedangkan Prosesnya sesuai dengan hasil wawancara dengan pengguna, maka untuk lebih memudahkan pemetaan dan sebagai alat bantu analisis lebih lanjut dibuat tabel definisi masalah sebagai berikut:

Tabel 1.2. Tabel definisi masalah untuk contoh kasus di atas

Input	Process	Output
A, B, X	Baca nilai A Baca nilai B $C = A + B$ Tampilkan nilai C Baca nilai X $Z = X * C$ Tampilkan nilai Z	C, Z

#### Rancangan Outline Pemecahan Masalah

Setelah dilakukan pendefinisian masalah maka akan didapat tiga hal, yaitu; output yang diharapkan, input yang harus diberikan dan proses untuk mengolah input menjadi output seperti yang diharapkan. Langkah selanjutnya adalah membuat outline pemecahan masalah.

Berikut adalah langkah-langkah dalam melakukan perancangan outline pemecahan masalah:

1. Buat langkah – langkah proses.
2. Buat rincian/detail proses.
3. Tentukan Variable dan/atau record.
4. Tentukan struktur kontrol (pengulangan, kondisi, dsb).
5. Buat logika ‘Mainline’ (logika utama).

Untuk merancang *outline* pemecahan masalah kita memerlukan proses yang sudah didapat langkah pendefinisian masalah. Selanjutnya proses tersebut dibuat lebih rinci/detil dan dengan menambahkan deklarasi variabel sebagai penampung nilai. **Variabel adalah penampung suatu**

nilai yang memiliki nama dan nilai penampung tersebut dapat berubah sesuai nilai yang diisikan ke penampung tersebut. Selain dengan menambahkan deklarasi variable, proses juga dibuat rinci dengan menentukan struktur kontrol dan pembuatan logika mainline atau logika utama (kedua hal tersebut akan dibahas lebih dalam pada bagian atau sesi 2 dan 3.

Contoh proses rancangan outline pemecahan masalah:

- Kita ambil kasus sebelumnya, yaitu; membuat program suatu perhitungan. Berdasarkan hasil definisi masalah didapatkan tabel definisi masalah sebagai berikut:

Tabel 1.3. Tabel definisi masalah untuk contoh kasus suatu perhitungan

Input	Process	Output
A, B, X	Baca nilai A Baca nilai B $C = A + C$ Tampilkan nilai C Baca nilai X $Z = X * C$ Tampilkan nilai Z	C, Z

Berdasarkan tabel definisi masalah di atas maka diketahui prosesnya adalah sebagai berikut;

```
Baca nilai A
Baca nilai B
C = A + C
Tampilkan nilai C
Baca nilai X
Z = X * C
Tampilkan nilai Z
```

Proses tersebut kemudian kita buat **detil dengan menambahkan deklarasi variabel**. Untuk menentukan deklarasi variabel apa saja yang perlu dibuat bisa kita lihat melalui tabel definisi masalah yaitu dengan melihat pada kolom *input* dan kolom *output* sebagai variabel untuk menampung nilai *input* dan variable penampung nilai *output*. Untuk itu maka kita perlu menambahkan deklarasi untuk variabel A, B, X, C dan Z. Dengan demikian maka proses sekarang akan menjadi seperti berikut:

```

Deklarasi variable A
Deklarasi variable B
Deklarasi variable X
Deklarasi variable C
Deklarasi variable Z
Baca nilai A
Baca nilai B
C = A + C
Tampilkan nilai C
Baca nilai X
Z = X * C
Tampilkan nilai Z

```

Untuk menentukan deklarasi variabel apa yang perlu ditambahkan selain dengan melihat pada kolom *input* dan *output*, juga dilakukan dengan melihat kolom *process*. Pada kolom *process* kita lihat apakah ada penampung nilai selain yang dituliskan pada kolom *input* dan *output*. Jika ada penampung nilai selain yang tertulis di kolom *input* dan *output*, maka penampung tersebut juga harus kita deklarasikan sebagai variabel pada proses detil.

Proses tersebut kemudian kita buat **detil dengan menambahkan struktur kontrol** yang diperlukan. Karena pada proses tersebut tidak terdapat untuk perulangan dan unsur kondisi dan syarat maka struktur kontrol perulangan dan kondisi tidak perlu ditambahkan. Berdasarkan analisis terhadap proses kita lihat bahwa ada *statement* untuk baca suatu nilai, namun pengguna tidak diberitahu nilai apa yang harus dimasukkan oleh pengguna, untuk itu maka kita harus menambahkan struktur kontrol *statement* tampilan yang akan memberitahu pengguna bahwa pengguna harus memasukkan suatu nilai untuk variabel tertentu. Dengan demikian maka proses sekarang akan menjadi seperti berikut:

```

Deklarasi variable A
Deklarasi variable B
Deklarasi variable X
Deklarasi variable C
Deklarasi variable Z
Tampilkan tampilan untuk memasukkan nilai A
Baca nilai A
Tampilkan tampilan untuk memasukkan nilai A
Baca nilai B
C = A + C
Tampilkan nilai C
Tampilkan tampilan untuk memasukkan nilai X
Baca nilai X
Z = X * C
Tampilkan nilai Z

```

Selanjutnya karena contoh kasus masih sederhana maka tidak diperlukan untuk penambahan logika mainline pada detil proses. Demikianlah contoh proses pembuatan rancangan outline pemecahan masalah.

