

## Prinsip Good Design

- ▀ User Compatibility
- ▀ Product Compatibility
- ▀ Task Compatibility
- ▀ Work Flow Compatibility
- ▀ Consistency
- ▀ Familiarity
- ▀ Simplicity
- ▀ Direct Manipulation
- ▀ Control
- ▀ Flexibility
- ▀ Responsiveness
- ▀ Invisible Technology
- ▀ Robustness
- ▀ Protection
- ▀ Ease of Learning
- ▀ Ease of Use

## User Compatibility

Seorang perancang sistem harus benar-benar paham tentang pengetahuan, cara berpikir dan cara menerima informasi dari user sehingga sistem yang nantinya akan digunakan oleh user dapat membuat user lebih produktif. Dan yang harus diperhatikan juga adalah bahwa perancang (*designer*) atau *developer* tidak sama dengan user.



## Product Compatibility

- ▀ Selalu memperhatikan dan mempertahankan kompatibilitas antar produk, misalnya mampu mengorbankan *User Interface* yang memungkinkan sistem lebih kompatibel.



## Task Compatibility

- ▀ Rancanglah *interface* sistem sesuai dengan tugas dari user, jangan sampai user kesulitan untuk menggunakannya, karena hal ini dapat menyebabkan aplikasi yang kita buat tidak akan terpakai dan akhirnya tidak dapat membantu pekerjaan/tugas user.



## Work Flow Compatibility

- ▀ Selalu mengorganisasikan setiap fungsinya sesuai dengan kategori fungsinya sehingga dapat memfasilitasi segala perubahan tugas user.



## Consistency

- ▀ Prinsip ini sudah jelas, bahwa sistem harus konsisten terhadap fungsionalitas/kegunaan dari sistem tersebut. **Contoh** sederhananya adalah ketika user menekan tombol "save" maka proses yang terjadi adalah penyimpanan bukan hapus data.

## Familiarity

- Gunakanlah konsep, terminologi, dan pengaturannya yang mudah dipahami oleh user. Seperti ikon atau gambar "Recycle Bin" pada Sistem Operasi Windows, ini membuktikan bahwa focus user terhadap gambar tersebut adalah file-file yang sudah dihapus sebelumnya.

## Simplicity

- Kompleksitas suatu aplikasi akan menimbulkan frustrasi pada user itu sendiri, maka dari itu gunakan *system default* pada aplikasi yang dirancang. Maksudnya adalah sediakan dan utamakanlah fungsi-fungsi yang benar-benar sesuai dengan tugas dari user. Usahakan agar tidak menampilkan semua fungsionalitasnya.



## Direct Manipulation

- Maksud dari prinsip ini adalah user dapat langsung menyaksikan aksi sistem pada suatu objek. Contoh sederhana, pada saat kita menekan mengetikkan huruf "A" maka di layar akan langsung muncul huruf "A".



## Control

- Sistem yang digunakan oleh user jangan sampai membuat user merasa frustrasi dan dikontrol oleh user. Seperti memberikan komentar pada saat user melakukan kesalahan dengan bahasa yang tidak membuat user merasa dikontrol oleh sistem tersebut.

## WYSIWYG

- WYSIWYG (*What You See Is What You Get*), artinya adanya korespondensi satu ke satu antara informasi di layar dengan informasi di *printed-output* atau file. Contoh, pada saat kita membuat laporan menggunakan [Microsoft Office](#) lalu mencetaknya (*print out*) laporan tersebut, maka hasil *print out* harus sama dengan yang ada pada lembar kerja Microsoft Office

## Flexibility

- Prinsip ini merupakan prinsip yang sangat penting bagi user dengan keterbatasan fisik. Ini berarti mengijinkan banyak kontrol dari user yang mendukung untuk menggunakan aplikasi yang kita rancang dan mampu mengakomodir kemampuan user yang lain. Seperti aplikasi yang dapat didukung oleh perangkat lain (*mouse, keyboard, joystick, trackball*).

## Responsiveness

- Sistem harus selalu merespon dengan cepat apa yang di *inputkan* oleh user. Seperti menampilkan *Progress Bar*.

## Invisible Technology

- Menyembunyikan detail teknis dari suatu sistem merupakan hal yang sangat direkomendasikan dalam membuat *User Interface*. Sehingga user tidak memiliki rasa khawatir dan ketakutan untuk menggunakan aplikasinya.

## Robustness

- Sistem harus mampu mentolerir kesalahan manusia baik disengaja maupun tidak disengaja dan yang umumnya tidak dapat dihindari. Menyediakan *Recovery System* merupakan hal yang baik untuk mengimplementasikan prinsip ini.

## Protection

- Prinsip ini berbeda dengan prinsip *Robustness*, karena pada prinsip ini sistem seharusnya memproteksi kesalahan-kesalahan umum manusia. Seperti pada saat kita menutup lembar kerja Microsoft Office yang belum kita simpan sebelumnya, maka Office akan secara otomatis memberikan konfirmasi untuk menyimpannya atau tidak.

## Ease of Learning

- Buatlah sistem yang mudah dipelajari bagi user *novice* (awam). Hal ini akan memberikan motivasi kepada user tersebut untuk menggunakannya.

## Ease of Use

- Buatlah sistem yang mudah digunakan untuk *expert user*. Sehingga sistem yang kita bangun tidak hanya dipakai untuk *novice user* tetapi bisa juga dipakai untuk user yang sudah ahli (berpengalaman).