



Definisi Usability



- Derajat kemampuan sebuah perangkat lunak untuk membantu penggunaanya menyelesaikan sebuah tugas.
- Tingkat produk dapat digunakan yang ditetapkan oleh user untuk mencapai tujuan secara efektif dan tingkat kepuasan dalam menggunakannya.

Menurut ISO daya guna adalah tingkat produk dapat digunakan yang ditetapkan oleh user untuk mencapai tujuan secara efektif dan tingkat kepuasan dalam menggunakan (ISO,1998)



Menurut ISO 1998, daya guna atribut terdiri dari:

1. **Efektifitas:** ketelitian dan kelengkapan dimana user mencapai tujuan.
2. **Efisiensi:** sumber daya pembelajaran dalam hubungannya dengan ketelitian dan kelengkapan untuk user.
3. **Kepuasan:** bebas dari ketidak nyamanan dan sikap positif dalam menggunakan produk.

Penentu keberhasilan sistem



- Berguna (useful)
- Dapat digunakan (usable)
- Digunakan (used)

Komponen Penentu Daya Guna



Gambar 1. Komponen Penentu Daya Guna (Nielsen, 2003)

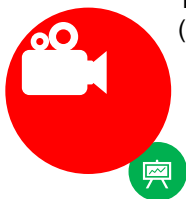
Tahapan pengukuran daya guna:



1. Pembelajaran (*learnbility*)
2. Keefisienan (*efficiency*)
3. Kemampuan mengingat (*memorability*)
4. Kadar kesalahan (*errors*)
5. Kepuasan (*saticfaction*)
6. Presentasi (*presentation*)
7. Susunan layar (*screen layouts*)
8. Istilah yang digunakan dan perintah yang disediakan oleh sistem
9. Kemampuan sistem (*system capabilitas*)



No	Kriteria untuk Metode Pengukuran Rekayasa Usabilitas
1	Waktu untuk menyelesaikan tugas
2	Berapa persen tugas bisa diselesaikan
3	Berapa persen tugas diselesaikan per unit waktu
4	Rasio keberhasilan dan kegagalan
5	Berapa waktu terjadi kesalahan
6	Berapa persen jumlah kesalahan
7	Berapa jumlah kompetitor dari produk yang sama
8	Jumlah perintah yang digunakan
9	Frekuensi help dan dokumentasi digunakan
10	Jumlah komentar dari user, yang baik maupun tidak
11	Jumlah perulangan perintah-perintah yang error
12	Jumlah run yang berhasil dan error
13	Berapa jumlah interface yang menyedatkan user
14	Jumlah fitur yang baik dan yang jelek yang digunakan
15	Jumlah perintah yang tidak pernah digunakan
16	Jumlah kelakuan sistem yang tidak diperlukan
17	Jumlah user yang menyukai sistem yang dibuat
18	Jumlah waktu yang digunakan untuk menyelesaikan suatu permasalahan
19	Jumlah waktu yang tidak efektif dalam menyelesaikan masalah
20	Berapa banyak user yang kehilangan kontrol terhadap sistem
21	Berapa jumlah user yang puas dan tidak dalam menggunakan sistem



Daya Guna / Usability Heuristik

Daya guna Heuristik merupakan prinsip atau panduan untuk merekayasa bentuk *user interface*.

Terdapat 10 prinsip sebagai daya guna heuristic (Pierre, 2015; Sudarmawan & Ariyus, 2007), diantaranya:

1. Dialog yang sederhana dan alami
2. Berbicara dengan bahasa pengguna
3. Mengurangi beban ingatan pengguna
4. Konsisten
5. Sistem timbal balik
6. Jalankeluar yang jelas
7. Jalan pintas
8. Pesan kesalahan yang baik
9. Mencegah Kesalahan
10. Bantuan dan dokumentasi





1. Dialog yang sederhana dan alami (*simple and natural dialogue*)

User Interface harus sesederhana mungkin dan bersifat natural. Setiap dialog harus menghindari perintah-perintah yang tidak perlu dan tidak ada hubungannya dengan *interface*, karena untuk setiap ciri atau elemen baru yang ditambahkan berarti satu masalah baru yang harus dipelajari oleh pengguna

2. Berbicara dengan Bahasa user (*speak the user language*)



Dialog harus menggunakan bahasa yang dipahami oleh user. Perintah-perintah yang berorientasi mesin mestinya tidak digunakan sama sekali. Frase-frase yang digunakan harus mudah dipahami oleh kebanyakan user (bukan segelintir saja). Penggunaan singkatan dan bahasa yang tidak jelas harus dihindari, karena dapat disalah tafsirkan



3. Mengurangi beban ingatan user (*minimize user memory load*)

User seharusnya tidak dibebani untuk mengingat atau menghafal pada saat berinteraksi dengan system. Contoh dengan menu dapat mengurangi beban user dibandingkan dengan penggunaan baris perintah.

4. Konsisten (*consistency*)

Ciri-ciri yang konsisten dari suatu user interface dapat menghindarkan user dari rasa ragu-ragu atau was-was di saat menggunakan suatu perintah atau fungsi untuk pertama kali. Selain itu juga dapat mempercepat interaksi



5. Sistem timbal balik (*system feedback*)

Sistem seharusnya memberitahu pengguna segala aktivitas yang sedang berlaku atau status system. Status system menunggu input, memproses, menampilkan output dan sebagainya. Dimana proses ini juga akan memberitahu status system jika terjadi suatu kerusakan.



6. Jalan keluar yang jelas (*clearly mark exit*)

System seharusnya bisa memberikan penjelasan tentang kondisi dan solusi untuk menghindari user terperangkap dalam tampilan-tampilan yang tidak diinginkan. Apabila user melakukan kesalahan dalam memilih perintah maka ia dapat keluar dari kesalahan tanpa ada masalah.



7. Jalan pintas (*shortcut*)

Demi kemudahan dan kecepatan interaksi di dalam penggunaan suatu sistem maka tersedia shortcut.



8. Pesan-pesan kesalahan yang baik (*good error message*)

Menyediakan mekanisme pemberitahuan kesalahan dan menunjukkan situasi bahwa user berada dalam kondisi bermasalah serta membantu user untuk lebih memahami sistem.

Terhadap empat peraturan yang harus diikuti dalam penggunaan pesan kesalahan:

- Pesan kesalahan yang digunakan harus jelas dan mudah dipahami.
- Pesan yang disampaikan harus bersifat khusus, tidak bersifat umum.
- Pesan kesalahan yang disampaikan sebaiknya menyediakan cadangan penyelesaian atas kesalahan yang telah dilakukan.
- Penyampaian pesan kesalahan harus dilakukan secara sopan..





**Kampus
Merdeka**
INDONESIA JAYA

Contoh ...

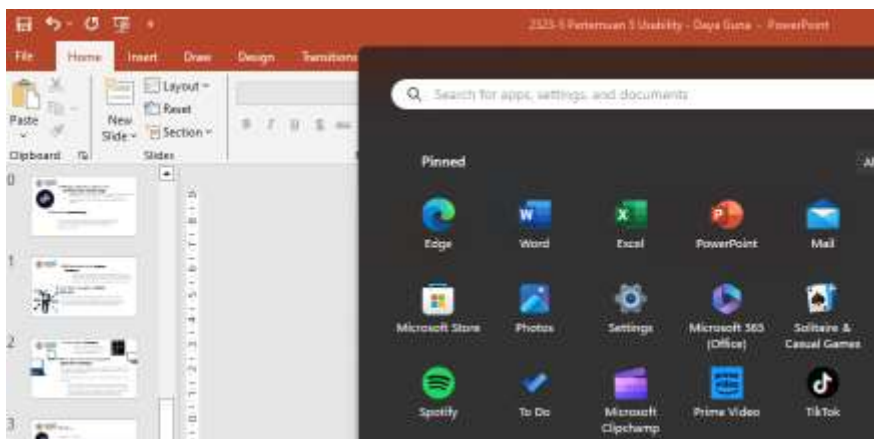


**Kampus
Merdeka**
INDONESIA JAYA

9. Mencegah Kesalahan (prevent errors)

Rekayasa interface yang baik seharusnya mampu membuat user menghindari kesalahan.

Contoh interaksi melalui menu





**Kampus
Merdeka**
INDONESIA JAYA

10. Bantuan dan Dokumentasi (*help and documentation*)



Bantuan dan dokumentasi merupakan kemudahan yang diberikan dalam kebanyakan sistem, menjelaskan cara menggunakan sistem, ciri-ciri khusus sistem dan membolehkan user untuk mengendalikan sistem dengan lebih baik.

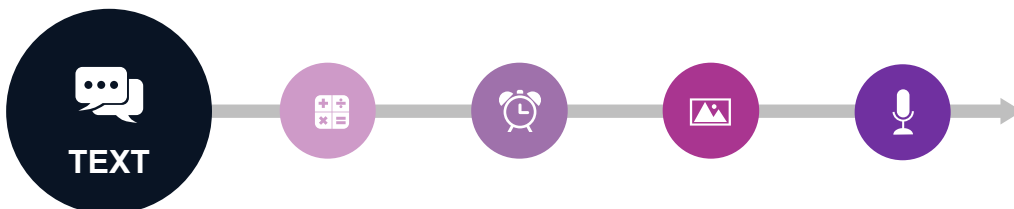


**Kampus
Merdeka**
INDONESIA JAYA

Teknik Daya Guna / Usability Siklus Hidup

Siklus hidup suatu daya guna, memiliki elemen antara lain :

- ✓ Know the user (Kenali Pengguna)
- ✓ Daya guna Banchmarking
- ✓ Goal-oriented interaction design
- ✓ Iterative design
- ✓ Follow up studies (studi Lanjutan)





Kenali Pengguna (*Know the User*)

Mengenal siapa user yang bertujuan untuk:

- ❖ mempelajari, mengenali dan memahami pengguna yang akan menggunakan system
- ❖ merangkum keperluan user,
- ❖ kepuasan,
- ❖ kemahiran komputer



Daya guna Banchmarking

Menganalisis perbandingan diantara beberapa produk agar bisa menghasilkan keputusan, panduan baru dan memberikan cadangan ide yang lebih baik terhadap rekayasa bentuk.

Produk-produk kompetitif atau produk yang sudah ada perlu dipelajari untuk memperbaiki system yang sedang dibangun, produk tersebut bisa dijadikan prototype terbaik untuk membangun sebuah produk

Menganalisis produk kompetitif dilakukan dengan jalan:

1. Menentukan kondisi dan memutuskan sejauh mana akan mengembangkan produk.
2. Meneliti perbedaan produk
3. Intelligent borrowing, ide dari sistem pesaing





Desain Interaksi Berorientasi Tujuan (Goal-oriented interaction design)

Desain yang dibangun selalu memiliki tujuan untuk berinteraksi.

Desain interface merupakan suatu interface antara kode dan user, desain interface mengacu ke fungsi, perilaku dan presentasi final, tujuannya sendiri

Cara kerja komputer tidak sama dengan manusia. Bagian perangkat lunak harus jelas, yang dituliskan pada instruksi-instruksi program dan bentuk dari interface harus bisa menyesuaikan dengan permintaan manusia



Iterative Design

Interface design bertujuan untuk desain tes dan re-desain, kemudian membangun prototipe interface dengan cara :

- . Menemukan masalah daya guna
- . Menetapkan masalah untuk interface yang baru
- . Men-capture dasar pemikiran desain, mengapa perubahan dibuat
- . Mengevaluasi interface



TERIMA KASIH

Pengantar Interaksi Manusia dan Komputer