Pertemuan 5

Guidelines, Principles, and Theories in Interaction System

(Penuntun, Prinsip-prinsip dan Teori – teori dalam Sistem Interaksi

Materi Pembahasan

1. Guidelines
2. Principles
3. Theoris
4. Object – Action Interface model

1. Guideline

Guidelines in Interaction /Pedoman dalam interaksi

- Menggunakan bermacam bahasa
- Penggunaan yang baik
- Kritik dalam berbagai cara seperti :

  - Terlalu fokus, Tidak serba ada, penerapan yang susah, dan kadang salah
- Pendukung

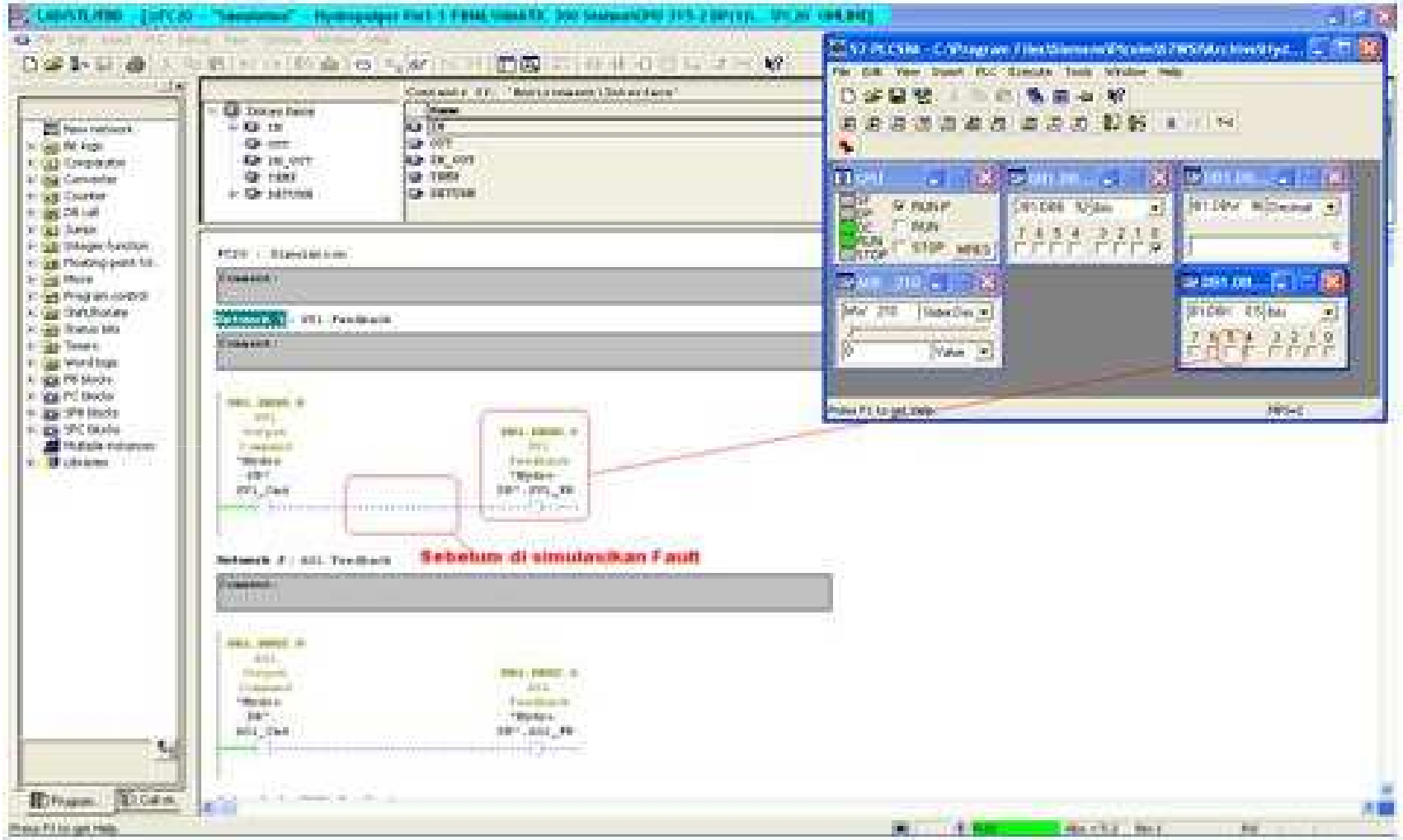
  - Merangkum semua pengalaman

A. Pedoman pengakesan

- Sediakan kesamaan untuk petunjuk dengan text dan non text
- Untuk setiap penyajian multimedia berikan alternatif yang sama
- Hindari bentuk text yang terlalu banyak warna
- Berikan judul pada setiap halaman untuk memudahkan identifikasi dan nagasi

B. Navigating the interface

- Contoh langakah Nagasi interface,
  - Buat urutan tugas2 yang baku
  - Memastikan semua hubungan dideskrpsikan dengan jelas
  - Use unique and descriptive headings
  - Use check boxes for binary choices
  - Develop pages that will print properly
  - Use thumbnail images to preview larger images

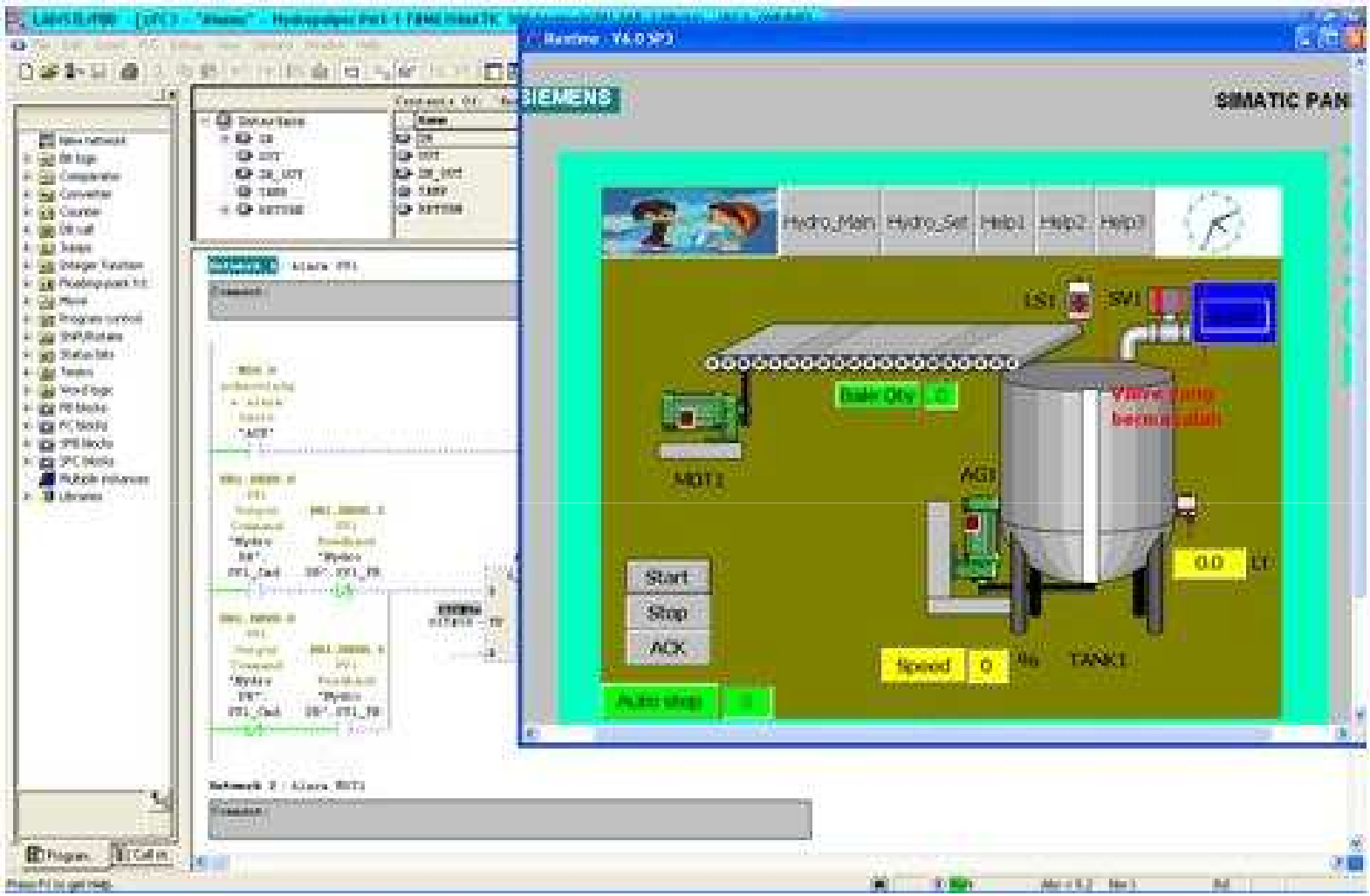


Contoh : Navigasi interface

C. Organizing the display

Smith and Mosier (1986) offer five high-level goals

- Konsistensi tampilan data
- Penyatuan informasi secara efisien oleh pengguna
- Sesuaikan data tampilan dengan data masukan
- Minimal memory load on the user
- Fleksibilitas untuk kontrol pengguna dari tampilan data



Contoh :organizing display

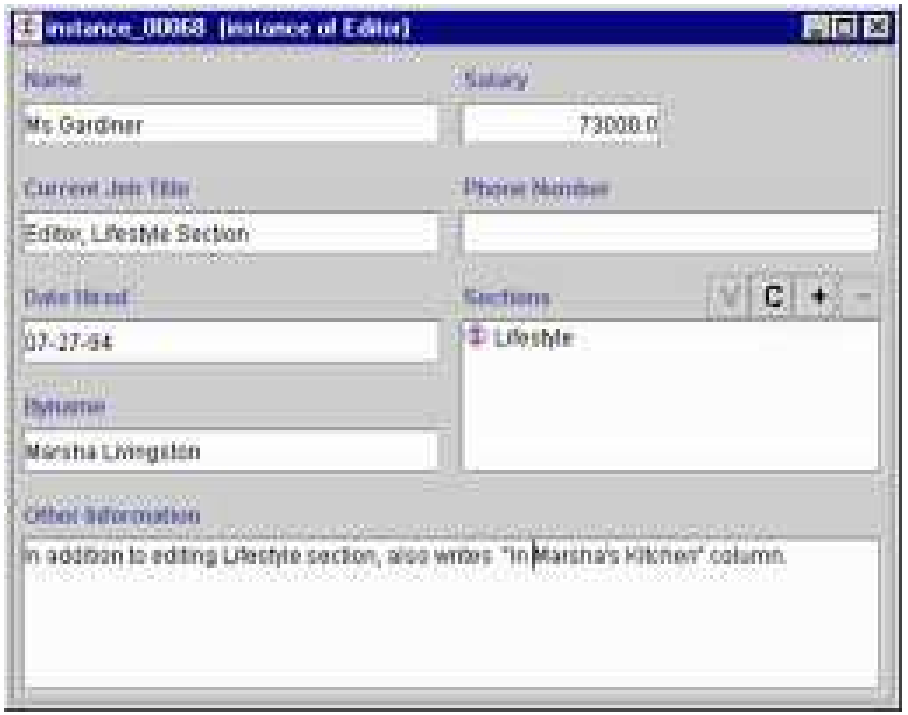
D. Getting the user's attention

- Intensity
- Ciri / Marking
- Size
- Choice of fonts
- Inverse video
- Blinking
- Color
- Audio

Contoh : visual user attention

E. Facilitating data entry

- Smith and Mosier (1986) offer five high-level objectives as part of their guidelines for data entry
  - Consistency of data-entry transactions
  - Minimal input actions by user
  - Minimal memory load on users
  - Compatibility of data entry with data display
  - Flexibility for user control of data entry



Contoh. Facilitating data entry

2. Principles

Prinsip prinsip interaksi

- More fundamental, widely applicable, and enduring than guidelines
- Need more clarification
- Fundamental principles
  - Determine user's skill levels
  - Identify the tasks
- Five primary interaction styles
- Eight golden rules of interface design
- Prevent errors
- Automation and human control

Syntactic knowledge:

By syntactic knowledge we mean the information necessary to be maintained and memorized by a user to be able to use a certain system efficiently. For example a user of a certain programming language can use it efficiently only when he knows a good deal of commands and syntax specific to this language. This kind of knowledge has the following drawbacks:

Places a considerable memory load on the user, which is hard to retain over time unless that knowledge is applied frequently. Recognition is usually preferred over recall. Most of the time, it is better to provide visual clues in the interface that make the user recall the underlying functionality

Hard to provide hierarchical or modular structure to cope with complexity of the system. For example, a user using a mail system can have several modes of termination : ENTER to terminate a paragraph, CTRL-D to terminate a letter, CTRL-C to cancel a letter, Q to quit the system, and so on . A novice user might be confused by those different modes of operations having a pretty similar syntactic structure.

Syntactic knowledge is system dependent. It's hard to apply previous system knowledge to the new one. This goes from using different keyboard layout to the different syntax necessary to perform each task (The reason behind that is that the user build a mental model of a system while interacting with it. Shifting to a new system, supposedly similar in functionality, the user tried to apply the previous model to that system and faces frustration when the results he gets are not the ones he is expecting.)

Determine user's skill levels/Tentukan tingkat keahlian pengguna

- "Know the user" Hansen (1971)
- Age, gender, physical and cognitive abilities, education, cultural or ethnic background, training, motivation, goals and personality
- Design goals based on skill level
  - Novice or first-time users
  - Knowledgeable intermittent users
  - Expert frequent users
- Multi-layer designs

The 8 golden rules of interface design

1. Strive for consistency
2. Cater to universal usability
3. Offer informative feedback
4. Design dialogs to yield closure
5. Prevent errors
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short term memory

3.Theories

- Beyond the specifics of guidelines
- Principles are used to develop theories
- Descriptions/explanatory or predictive
- Motor task, perceptual, or cognitive

A. Explanatory and predictive theories

- Penjelasan Teori

- - Mengamati prilaku
- - Menggambarkan aktifitas
- - Bayangkan perancangan
- - Membuat 2 perbandingan konsep perancangan
- - Pelatihan

Predictive theories:

– Enable designers to compare proposed designs for execution time or error rates

B. Perceptual, Cognitive, & Motor tasks

- **Perceptual or Cognitive subtasks theories**
  - Predicting reading times for free text, lists, or formatted displays
- **Motor-task performance times theories:**
  - Predicting keystroking or pointing times

C. Conceptual, semantic, syntactic, and lexical model

- **Foley and van Dam four-level approach**
  - ***Conceptual level:***
    - **User's mental model of the interactive system**
  - ***Semantic level:***
    - **Describes the meanings conveyed (**dinyatakan) **by the user's command input and by the computer's output display**
  - ***Syntactic level:***
    - **Defines how the units (words) that convey semantics are assembled into a complete sentence that instructs the computer to perform a certain task**

- *Lexical level:*
  - Deals with device dependencies and with the precise mechanisms by which a user specifies the syntax

- Approach is convenient for designers
  - Top-down nature is easy to explain
  - Matches the software architecture
  - Allows for useful modularity during design

4. Object – Action Interface Model

There exists two basic interaction models for any given system :

1. Object-Action model : The user first selects an object and then selects the action to be performed on the selected object

2. Action-Object model : The user first selects an action to be performed and then selects the objects on which this action will be performed.

The OAI model

Designing an OAI model starts with examining and understanding the tasks to be performed by the system. The domain of tasks include the universe of objects within which the user works to accomplish a certain goal as well as the domain of all possible actions performed by the user. Once these tasks objects and actions are agreed upon, the designer starts by creating an isomorphic representation of the corresponding interface objects and actions.

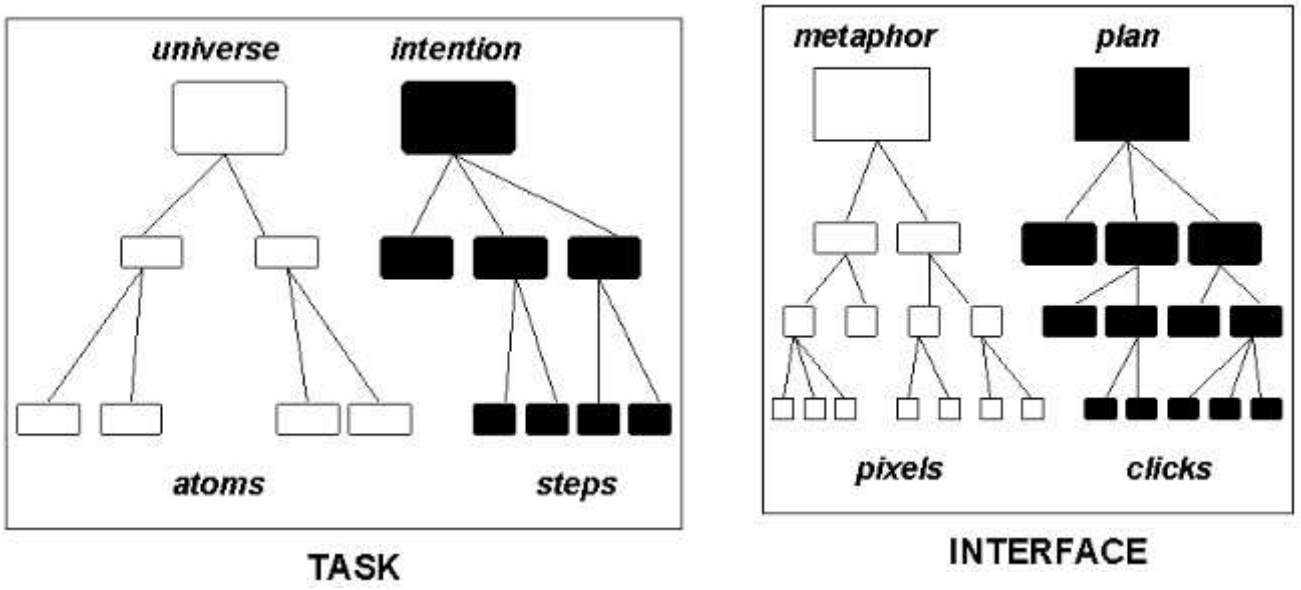


Figure from Designing the user interface by Ben Shneiderman

Task hierarchies of objects and actions:

Tasks include hierarchies of objects and actions at different high and low levels. For a certain user, these hierarchies might not be perfect, but since they are comprehensible, they provide a great deal of usefulness.

For the user:

Hierarchical decomposition of a complex task into several simpler tasks has been a successful way in problem solving specially when dealing with large complex problems. Most real world entities and objects have this property of being built of hierarchical simple structures. Similarly, intentions can be decomposed into a plan of small action steps. People learn tasks actions and objects through their course of life while interacting with these objects and learning about their properties, functionalities and limitations. It is easy to conclude that people learn these issues independently from the underlying implementation on a certain system. We note that a user has to be first proficient in the task domain before using an interface to accomplish those real-world tasks.

For the designer:

The following steps are recommended (Shneiderman) in order to build correct tasks hierarchies by designers for a system:

1. Know about the users and their tasks (interviewing users, reading workbooks and taking training sessions)
2. Generate hierarchies of tasks and objects to model the users' tasks
3. Design interface objects and actions that metaphorically map to the real world universe

Interface hierarchies of objects and actions:

Similar to the task domain, the interface domain contains hierarchies of objects and tasks at different levels.

Interface Objects:

Users interacting with a computer get to understand some high level concepts relevant to that system. As an example, they learn that computer stores information, that these information are stored in files contained within a hierarchy of directories, and that each file has its own attributes like name, size, date, etc ...

Interface Actions:

These are also hierarchies of lower levels actions. A high level plan is to create a text file might involve mid-level actions such as creating a file, inserting text and saving that file. The mid-level action of saving a file the file can be decomposed into lower level actions such as storing the file with a backup copy and may be applying the access control rights. Further lower level actions might involve choosing the name of the file, the location folder to be saved in, dealing with errors such as space shortage, and so on.

For the user:

There are several ways users learn interface objects and actions such as demonstrations, sessions, or trial and error sessions. When these objects and actions have logical structure that can be related to other familiar task objects and actions, this knowledge becomes stable in the user's memory.

For the designer:

The OAI model helps a designer to understand the complex processes that a user has to perform in order to successfully use an interface to perform a certain task. Designers model the interface actions and objects based on familiar example and then fine tune these models to fit the task and the user.