

# Selamat Datang

*Semoga Tuhan memberi berkah pada kelas ini.*

## • TUJUAN

Agar mahasiswa memahami Sistem Pakar

Agar mahasiswa dapat memahami  
aplikasi dan penerapan dari sistem pakar

## MATERI POKOK

Pertemuan ke-	Pokok Bahasan
1	Pengenalan Artificial Intelligent
2	Pengenalan Sistem Pakar (Expert System)
3	Model Sistem Pakar
4	Representasi Pengetahuan (Bagian 1)
5	Representasi Pengetahuan (Bagian 2)
6	Representasi Pengetahuan (Bagian 3) – Logika dan Pengetahuan
7	Review Materi / Kuis (Soal-Soal Essay)
8	UJIAN TENGAH SEMESTER
9	Metode Inferensi: Graph, Trees, Lattice
10	Metode Inferensi: Logika Deduktif dan Silogisme
11	Metode Inferensi: Argumen & logika Proporsional
12	Metode Inferensi: Rangkaian Forward & Backward
13	Ketidakpastian & Paradigma Soft Computing
14	Pengenalan CLIPS
15	Review Materi / Kuis (Soal-Soal Essay)
16	UJIAN AKHIR SEMESTER

## Sumber Referensi :

- H.S, Suryadi. Seri Diktat Kuliah: Pengantar Sistem Pakar. Penerbit Gunadarma. Jakarta. 1994.
- Arhami, Muhammad. Konsep Dasar Sistem Pakar. Andi. Yogyakarta. 2005.
- Kusrini. Sistem Pakar, Teori dan Aplikasi. Penerbit Andi. Yogyakarta. 2006.
- Fausett, Laurene V. (1993). Fundamentals of Neural Network: Architectures, Algorithms and Applications. New Jersey: Prentice Hall.

## CATATAN :

1. Mahasiswa wajib mempunyai buku referensi tersebut
2. Mahasiswa wajib memiliki slide dan LTM (Lembar Tugas Mahasiswa)
3. Mahasiswa bisa mencari jurnal yang berhubungan dengan Sistem pakar

# RENCANA PEMBELAJARAN

- Pertemuan 1 s.d 14 disampaikan dengan Metode Ceramah, Metode Diskusi dan Latihan Soal.
- Sistem Penilaian
  - 10 % Kehadiran
  - 30 % Nilai UTS
  - 40 % Nilai UAS
  - 20 % Tugas (LTM, Kuis)

# Deskripsi Singkat Tugas

Mahasiswa mengerjakan soal-soal Lembar Tugas  
Mahasiswa

## PENGENALAN

# ARTIFICIAL INTELLIGENCE



Pertemuan ke Satu

## Definisi *Artificial Intelligence (AI)* :

### Pendekatan Teknik (*An Engineering Approach*)

Merupakan proses di mana peralatan mekanik dapat melaksanakan kegiatan-kegiatan yang berdasarkan pada pemikiran atau kecerdasan buatan.

### Pendekatan Ilmiah (*A Scientific Approach*)

Merupakan ilmu yang mempelajari bagaimana sebuah komputer dapat melakukan kegiatan yang dapat dilakukan oleh manusia.

## Sejarah AI

- 1955, Pengembangan dalam bidang permainan atau game Catur oleh Shannon
- 1956, Istilah AI mulai dipopulerkan oleh John McCarthy sebagai suatu tema ilmiah di bidang komputer yang diadakan di Dartmouth College.
- 1956, Komputer berbasis AI pertama kali dikembangkan dengan nama Logic Theorist yang melakukan penalaran terbatas untuk teorema kalkulus. Perkembangan ini mendorong para peneliti untuk mengembangkan program lain yang disebut sebagai General Problem Solver (GPS).
- 1963, Checkers-playing program diperkenalkan oleh Samuel, program ini dapat digunakan untuk memainkan game dan punya kemampuan untuk menyimpan pengalaman untuk digunakan pada permainan berikutnya.

- 1972, Newell dan Simon memperkenalkan Teori Logika secara konseptual yang kemudian berkembang pesat dan menjadi acuan pengembangan sistem berbasis AI lainnya.
- 1978, Buchanan dan Feigenbaum mengembangkan bahasa pemrograman DENDRAL, bahasa pemrograman ini dibuat untuk Badan Antariksa AS (NASA) dan digunakan untuk penelitian kimia di planet Mars.

- Pada perkembangan selanjutnya studi pada AI difokuskan pada pemecahan masalah sehari-hari atau memberi pertimbangan yang masuk akal (**commonsense reasoning**) terhadap permasalahan yang dihadapi manusia.
- Hal ini mencakup pertimbangan mengenai suatu objek dan hubungannya dengan objek yang lain (mis : suatu objek mungkin berada dalam objek yang lain) dan pertimbangan tindakan apa yang dapat dilakukan terhadap objek dan konsekuensinya (mis : bila kita melepaskan objek dari genggaman objek tsb akan jatuh dan mungkin pecah).

## Beberapa bidang Tugas AI

### Bidang Umum

- Perception : Vision, Speech
- Natural Language : Understanding, Generation, Translation
- Commonsense Reasoning
- Robot control

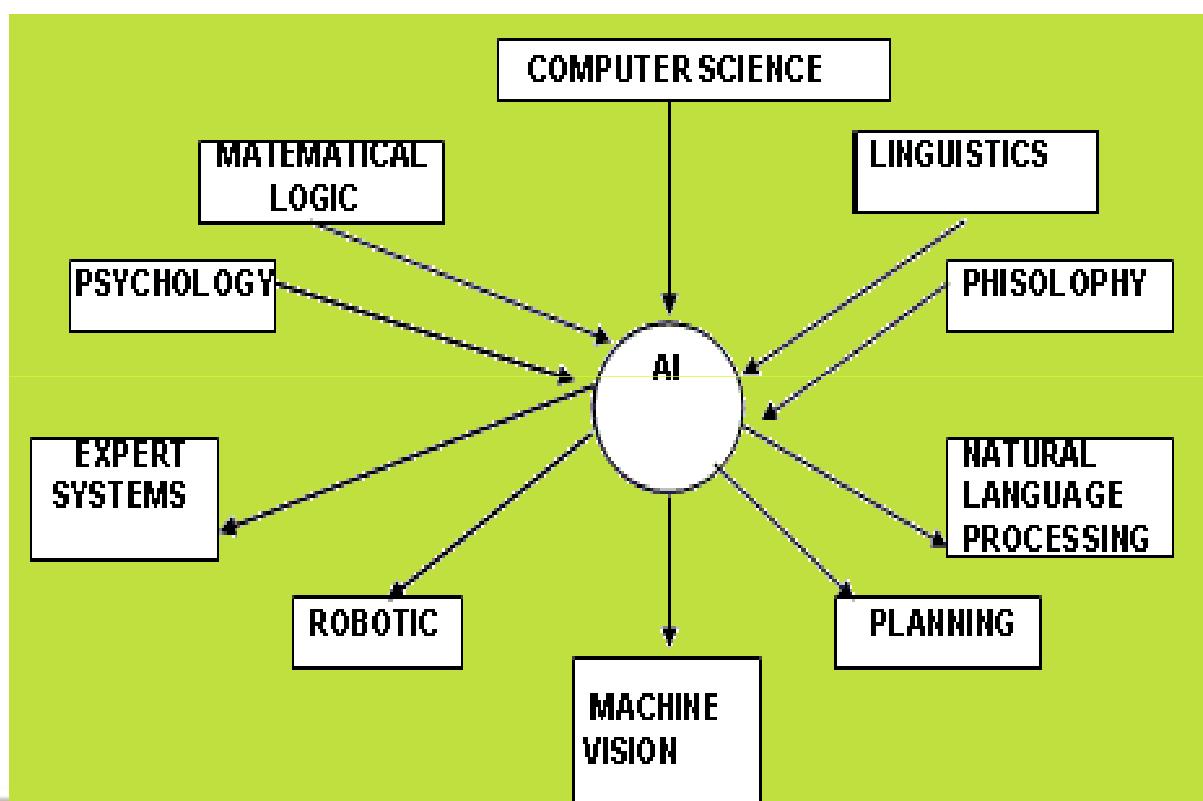
### Bidang Formal

- Games : Chess, Bakgammon, Checkers, Go
- Mathematics : Geometry, Logic, Integral Calculus

### Bidang Ahli

- Engineering : Design, Fault finding, Manufacturing planning
- Scientific analysis
- Medical diagnosis
- Financial analysis

## Bagian-Bagian AI



## Teknik-Teknik AI

Beberapa teknik AI yang penting diantaranya

### 1. **Search** (Pencarian)

menyediakan cara penyelesaian masalah untuk kasus dimana bila tidak ada lagi pendekatan langsung yang dapat digunakan maka pindahkan kerangka kerja kpd teknik langsung yang mungkin untuk dilekatkan.

### 2. **Use of Knowledge** (Penggunaan Pengetahuan)

menyediakan cara penyelesaian masalah yang lebih kompleks dengan mengeksploitasi struktur dari objek yang terkait dengan masalah tsb.

### 3. **Abstraction**

menyediakan cara untuk memilah/memisahkan keterangan dan variasi yang penting dari sekian banyak yang tidak penting dimana akan mempercepat penyelesaian masalah.

## Pertemuan 2

### Pengenalan Expert System (Sistem Pakar)

## Definisi Sistem Pakar

Sistem Pakar adalah suatu sistem yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian seorang pakar.

Yang diperlukan untuk membangun sistem pakar adalah sejumlah pengetahuan dan suatu mekanisme untuk mengakses pengetahuan itu secara efisien (mekanisme inferensi) untuk memecahkan masalah.

## Kelebihan Sistem Pakar

### 1. Increased Availability

Pengetahuan seorang pakar yang sudah diadaptasi kebentuk software dapat diperbanyak dan disebarluaskan dalam jumlah yang tidak terbatas

### 2. Reduced cost

biaya memperbanyak software lebih murah dibanding menghadirkan atau melatih seorang pakar

### 3. Reduced danger

dapat digunakan pada keadaan dan atau lokasi yang membahayakan manusia

### 4. Permanence

software sistem pakar dapat digunakan kapan saja tanpa ada batas waktu

## 5. **Multiple expertise**

sistem pakar dapat dibuat dan digunakan secara simultan dengan penggabungan beberapa pengetahuan dari banyak pakar

## 6. **Increased reliability**

Sifat komputer yang pasti dan selalu benar selama masukan dan algoritma yang diberikan benar shg keluaran sistem pakar dapat menjadi penunjang keputusan yang dapat diterima

## 7. **Explanation**

Sistem pakar dibuat dengan mekanisme inferensi yang menghasilkan suatu keluaran disertai penjelasan yang masuk akal, shg sistem pakar dapat juga digunakan untuk menjelaskan suatu teori atau keadaan tertentu

## 8. **Fast Response**

Kecepatan proses hardware komputer dapat terus ditingkatkan tanpa batas, shg kecepatan penyelesaian masalah dengan sistem pakar ikut meningkat

## 9. **Steady, unemotional, complete**

komputer bekerja dalam keadaan yang relatif tetap setiap saat, keadaan yang sulit sekali dicapai oleh manusia

## 10. **Intelligent tutor**

sistem pakar dapat dijadikan sumber belajar yang baik bagi siswa dengan menjalankan kasus-kasus

## 11. **Intelligent database**

komputer tidak memiliki sifat lupa, sistem pakar akan memberikan informasi yang sesuai selama tersimpan dalam database nya

## Kelemahan Pengembangan Sistem Pakar

Beberapa kelemahan Sistem pakar diantaranya :

- Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem
- Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan perangkat lunak konvensional.

## Perbandingan Perangkat Lunak Konvensional dengan Perangkat Lunak Sistem Pakar

Perangkat Lunak Konvensional	Perangkat Lunak Sistem Pakar
Fokus Pada Solusi	Fokus Pada Permasalahan
Pengembangan dapat dilakukan secara individu	Pengembangan dilakukan oleh tim kerja
Pengembangan secara sekuensial	Pengembangan secara iteratif

# Karakteristik Sistem Pakar

## 1. High Performance

Sistem pakar harus dapat bekerja pada level kompetensi yang sama atau lebih baik dari seorang pakar

## 2. Adequate response time

Sistem pakar harus dapat bekerja dan menghasilkan solusi atau kesimpulan lebih cepat dari pakar manusia

## 3. Good reliability

Sistem pakar harus dapat dipercaya dimana solusi yang dihasilkan tidak cenderung atau mengacu kepada satu hal atau keadaan tertentu saja

## 4. Understandable

Sistem pakar harus dapat menjelaskan setiap solusi yang dihasilkan sebagaimana juga seorang pakar dapat menjelaskan setiap kesimpulan yang dibuat, bukan black box system

## 5. Flexibility

Karena pengetahuan terus berkembang dan berubah, maka sebuah sistem pakar juga harus memiliki mekanisme untuk menambah, merubah dan memperbarui pengetahuan yang ada di dalamnya.

## 6. Hipotesis

Sistem pakar dapat melakukan serangkaian hipotesa yang dapat dibandingkan dan tidak bertentangan dengan hipotesa dari seorang pakar dalam masalah yang nyata

## Pembagian Sistem Pakar berdasar kelas

### 1. Konfigurasi

Merakit komponen sistem dengan cara yang benar

### 2. Diagnosa

Menarik kesimpulan terhadap masalah yang dihadapi berdasarkan bukti-bukti yang diobservasi

### 3. Instruksi

Metode pengajaran yang cerdas sehingga siswa sbg user dapat melakukan tanya jawab dengan sistem seolah berhadapan dengan pengajar atau seorang pakar

### 4. Interpretasi

Menjelaskan data-data yang diobservasi

### 5. Monitoring

Membandingkan data yang diamati dengan data acuan yang diharapkan untuk memberi penilaian terhadap prestasi, kinerja atau masalah tertentu

### 6. Perencanaan

Merancang tindakan atau menentukan langkah yang harus diambil untuk mendapatkan hasil yang dikehendaki

### 7. Prognosis

Membuat perkiraan hasil dari data yang ada atau keadaan yang sudah atau sedang terjadi

### 8. Perbaikan

Menjelaskan tindakan yang berlaku atau yang valid untuk suatu masalah atau keadaan tertentu

### 9. Kontrol

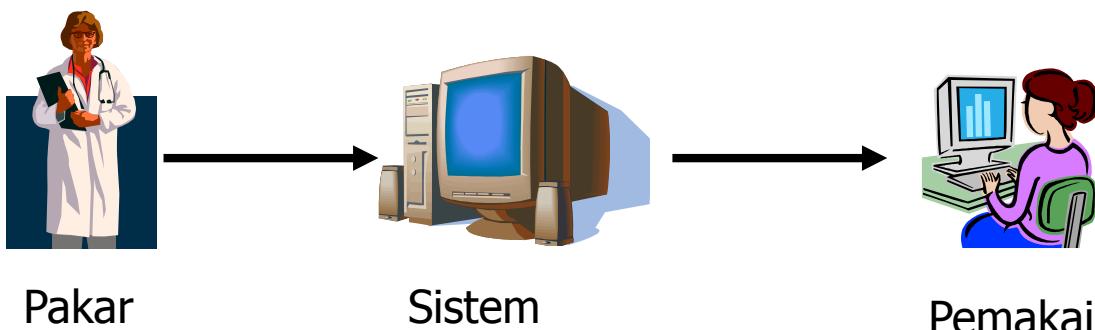
mengatur proses yang berasal dari penggabungan kelas sistem pakar yang lain

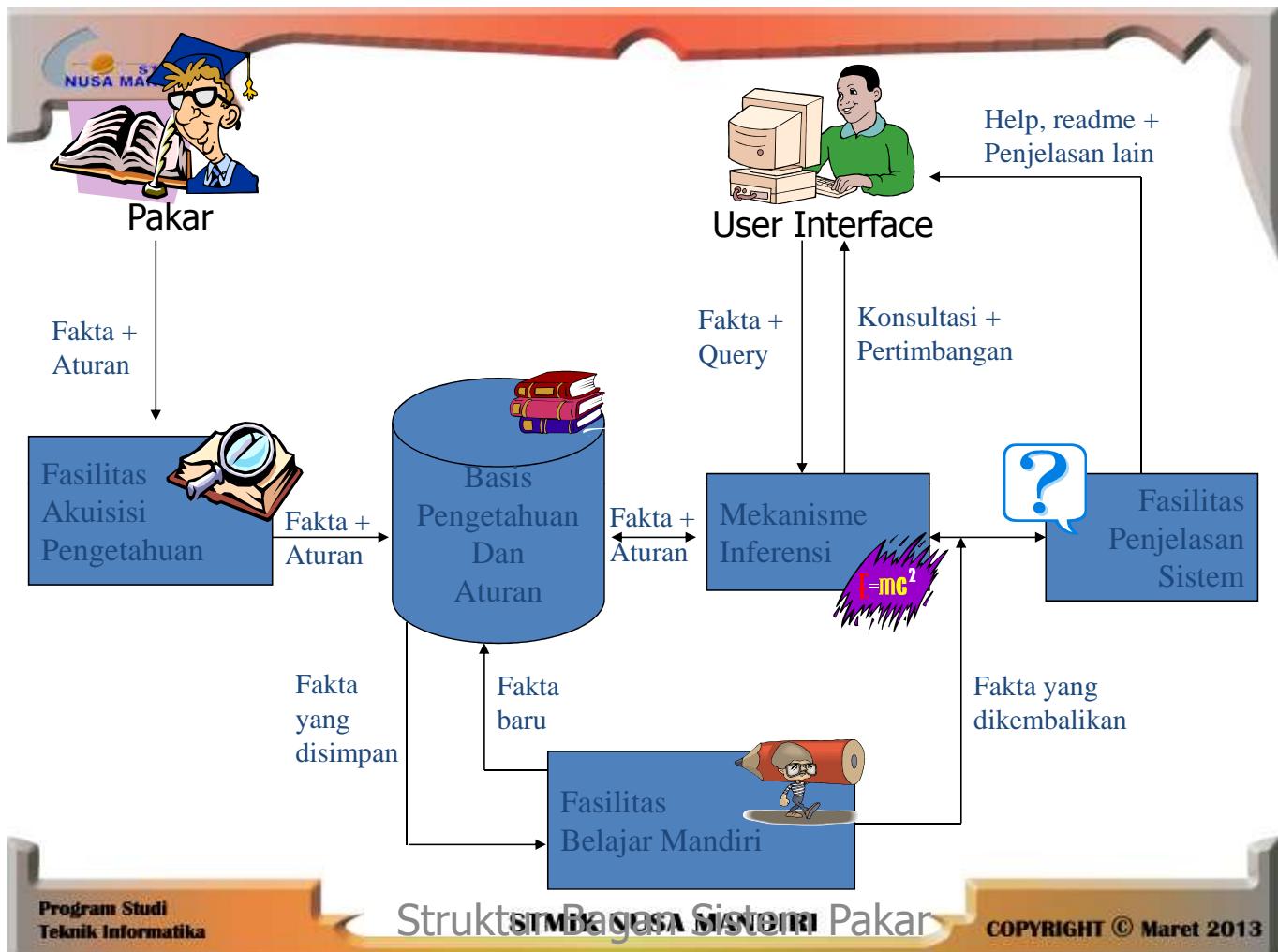
# Pertemuan 3

## Model Expert System

## Unsur Penting Pengembangan Sistem Pakar

Ada tiga unsur penting dalam pengembangan Sistem Pakar, yaitu :





## Komponen Sistem Pakar

Komponen Sistem Pakar terdiri dari :

1. Fasilitas Akuisisi pengetahuan
2. Basis Pengetahuan dan Basis Aturan
3. Mekanisme Inferensi
4. Fasilitas Belajar Mandiri
5. Fasilitas Penjelasan Sistem
6. Antarmuka Pemakai





## 1. Fasilitas Akuisisi Pengetahuan

Merupakan suatu proses untuk mengumpulkan data-data pengetahuan tentang suatu masalah dari pakar.

Bahan pengetahuan dapat diperoleh dengan berbagai cara, seperti dari buku, jurnal ilmiah, pakar di bidangnya, laporan, literatur dsb. Sumber pengetahuan tsb dijadikan dokumentasi untuk dipelajari, diolah dan diorganisir secara terstruktur menjadi basis pengetahuan.

## 2. Basis Pengetahuan dan Basis Aturan



Ada beberapa cara merepresentasikan data menjadi basis pengetahuan, seperti dalam bentuk atribut, aturan-aturan, jaringan semantik, frame dan logika. Semua bentuk representasi data tsb bertujuan untuk menyederhanakan data sehingga mudah dimengerti dan mengefektifkan proses pengembangan program.

Dalam pemrograman visual umumnya disediakan sarana untuk mengembangkan tabel-tabel penyimpanan data yang terangkum dalam sebuah database.



### 3. Mekanisme Inferensi

Adalah bagian sistem pakar yang melakukan penalaran dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu. Selama proses konsultasi antara sistem dan pemakai, mekanisme inferensi menguji aturan satu persatu sampai kondisi aturan itu benar.

Secara umum ada dua teknik utama yang digunakan dalam mekanisme inferensi untuk pengujian aturan, yaitu penalaran maju (*forward reasoning*) dan penalaran mundur (*reverse reasoning*)



### 4. Fasilitas Belajar Mandiri

Fasilitas ini memungkinkan sistem untuk mengembangkan dirinya sendiri dengan memilah atau mengelompokan kembali fakta yang sudah ada, memasukkan fakta-fakta baru kedalam basis pengetahuan yang merupakan hasil penurunan (iterasi) dari fakta-fakta sebelumnya dan dapat mengembalikan fakta ke pada mekanisme inferensi sehingga dapat dimintakan fakta lainnya dari pemakai melalui antarmuka pemakai



## 5. Fasilitas Penjelasan Sistem

Merupakan bagian komponen sistem pakar yang memberikan penjelasan tentang bagaimana program dijalankan, apa yang harus dijelaskan kepada pemakai tentang suatu masalah, memberikan rekomendasi kepada pemakai, mengakomodasi kesalahan pemakai dan menjelaskan bagaimana suatu masalah terjadi.

Dalam sistem pakar, fasilitas penjelasan sistem sebaiknya diintegrasikan ke dalam tabel basis pengetahuan dan basis aturan karena hal ini lebih memudahkan perancangan sistem



## 6. Antarmuka Pemakai

Komponen ini memberikan fasilitas komunikasi antara pemakai dan sistem, memberikan berbagai fasilitas informasi dan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi.

Syarat utama membangun antarmuka pemakai adalah kemudahan dalam menjalankan sistem, tampilan yang interaktif, komunikatif dan mudah bagi pemakai

# REPRESENTASI PENGETAHUAN

## (Bagian 1)

Pertemuan keempat

## Arti dari Pengetahuan

Pengetahuan merupakan salah satu kata dimana banyak orang mengetahui maknanya, tetapi sulit untuk mendefinisikannya.

Kata pengetahuan memiliki banyak arti, dan kata-kata lain seperti data, fakta dan informasi sering digunakan sebagai sinonim dari pengetahuan.

**Epistemology** adalah suatu studi tentang pengetahuan, studi ini dihubungkan dengan alam, struktur dan keaslian pengetahuan. Epistemology mengekspresikan pengetahuan dalam tiga bentuk, yaitu secara philosophy, priori dan posteriori

**Philosophy** digunakan untuk mengekspresikan kebenaran umum (general truth) yang terjadi di alam

Pengetahuan “*a priori*” datang sebelumnya dan bebas dari arti, secara universal benar dan tidak menimbulkan kontradiksi.

Contoh : “semua kejadian pasti ada sebabnya”

“jumlah sudut dalam segitiga 180 derajat”

Pengetahuan “*a posteriori*” adalah pengetahuan yang diperoleh dari arti. Kebenaran atau kesalahan pengetahuan posteriori dapat bervariasi, suatu pernyataan yang benar pada suatu saat, dapat disangkal dan menjadi salah pada saat yang lain.

## Klasifikasi lebih lanjut

Pengetahuan diklasifikasikan lebih lanjut kedalam *procedural knowledge*, *declarative knowledge* dan *tacit knowledge*.

Pengetahuan **prosedural** sering disebut sebagai pengetahuan tentang bagaimana melakukan sesuatu.

Pengetahuan **deklaratif** mengacu pada pengetahuan bahwa sesuatu itu benar atau salah, sehingga biasanya menghasilkan anjuran atau larangan.

Pengetahuan **tacit** kadang disebut sebagai “*unconscious knowledge*” karena tidak dapat diekspresikan dengan bahasa. Contoh : bagaimana cara mengangkat tangan, berjalan, mengendarai sepeda, dsb.

# Analogi Wirth

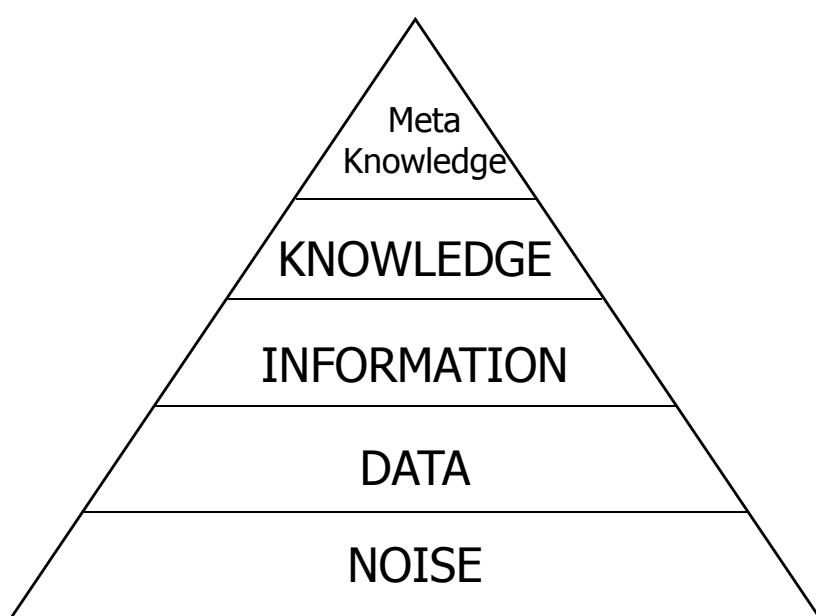
Pengetahuan merupakan hal penting dalam sistem pakar, seperti yang dianalogikan Nicklaus Wirth dalam ekspresi :

Algoritma + Struktur Data = Program

Dan

Pengetahuan + Inferensi = Sistem Pakar

# Hirarki Pengetahuan



## Bentuk Representasi Pengetahuan

1. Produksi
2. Jaringan Semantik
3. Schemata
4. Frame
5. Logika



## Produksi

Produksi atau sering juga disebut himpunan produksi, baris produksi atau baris, merupakan bentuk representasi pengetahuan yang menggunakan Backus-Nour Form (BNF) sebagai metalanguage untuk menentukan sintaks bahasa. Metalanguage adalah suatu bahasa yang digunakan untuk menjelaskan bahasa.

Suatu string atau deretan kata dapat dinyatakan sebagai kata atau kalimat yang valid jika dapat diturunkan atau diderivasi mulai dari simbol start sampai terbentuk kalimat dengan menggunakan baris produksi yang ditentukan.

Grammar merupakan set / rangkaian baris produksi lengkap yang menentukan suatu bahasa secara tidak ambigius.

Parse tree atau Derivation tree merupakan bentuk representasi grafis dari kalimat yang diuraikan kedalam simbol terminal dari seluruh simbol nonterminal yang digunakan untuk mendapatkan kalimat.

Compiler akan membuat parse tree pada saat mencoba menentukan apakah suatu pernyataan dalam program sesuai atau tidak dengan sintaks yang ditentukan.

Cara alternatif penggunaan produksi adalah untuk membuat kalimat dengan mengganti seluruh simbol terminal dengan simbol nonterminal sampai mencapai start (metode backward)



## Jaringan Semantik

Merupakan teknik representasi AI klasik yang digunakan untuk informasi proporsional, sehingga jaringan semantik sering disebut juga sebagai **jaringan proporsional**.

Proporsi merupakan kalimat, baik benar maupun salah.

Proporsi merupakan bentuk dari pengetahuan deklaratif karena proporsi menyatakan fakta. Proporsi selalu benar atau salah dan disebut sebagai atomic karena nilai kebenarannya tidak dapat dibagi lagi.

Jaringan semantik pertama kali dikembangkan untuk AI sebagai cara untuk menunjukkan memory manusia dan pemahaman bahasa. Jaringan semantik digunakan untuk menganalisa arti kata dalam kalimat, diterapkan juga pada banyak problem, termasuk representasi pengetahuan.

Struktur jaringan semantik digambarkan secara grafis dalam bentuk nodes dan arcs yang menghubungkannya. Nodes sering juga disebut sebagai objek dan arcs sering juga disebut sebagai links atau edges. Link digunakan untuk mengekspresikan suatu relasi, sedangkan node pada umumnya digunakan untuk menunjukkan objek fisik, konsep atau situasi.

Relasi didalam jaringan semantik sangatlah penting karena relasi tersebut menyediakan struktur pokok untuk pengorganisasian pengetahuan. Tanpa suatu relasi, maka pengetahuan hanya akan merupakan koleksi sederhana dari fakta yang tidak saling berhubungan. Dengan relasi, pengetahuan merupakan struktur kohesif tentang hubungan pengetahuan lain yang dapat disimpulkan.

## Tipe Relasi/Link dlm Jaringan Semantik

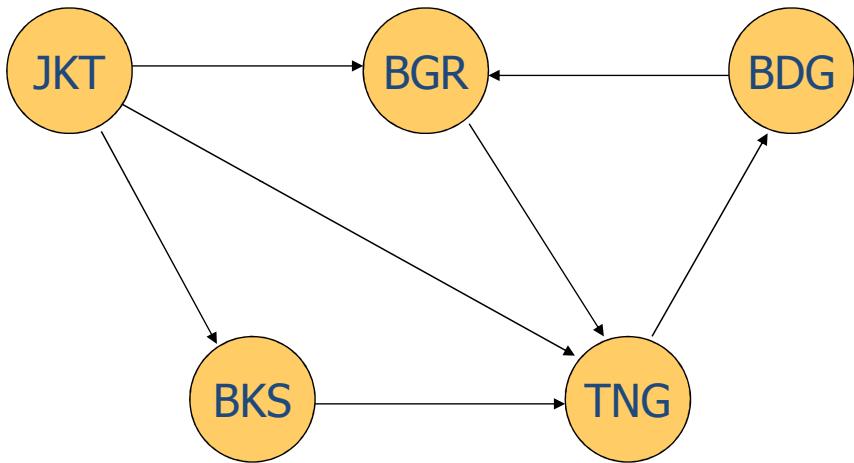
Dua tipe relasi atau link yang sering digunakan pada jaringan semantik adalah is-a (IS-A) dan a-kind-of (AKO).

Link IS-A biasa digunakan untuk menyatakan jarak antar node atau untuk menyatakan suatu objek merupakan anggota dari suatu kelompok objek atau kelas objek tertentu.

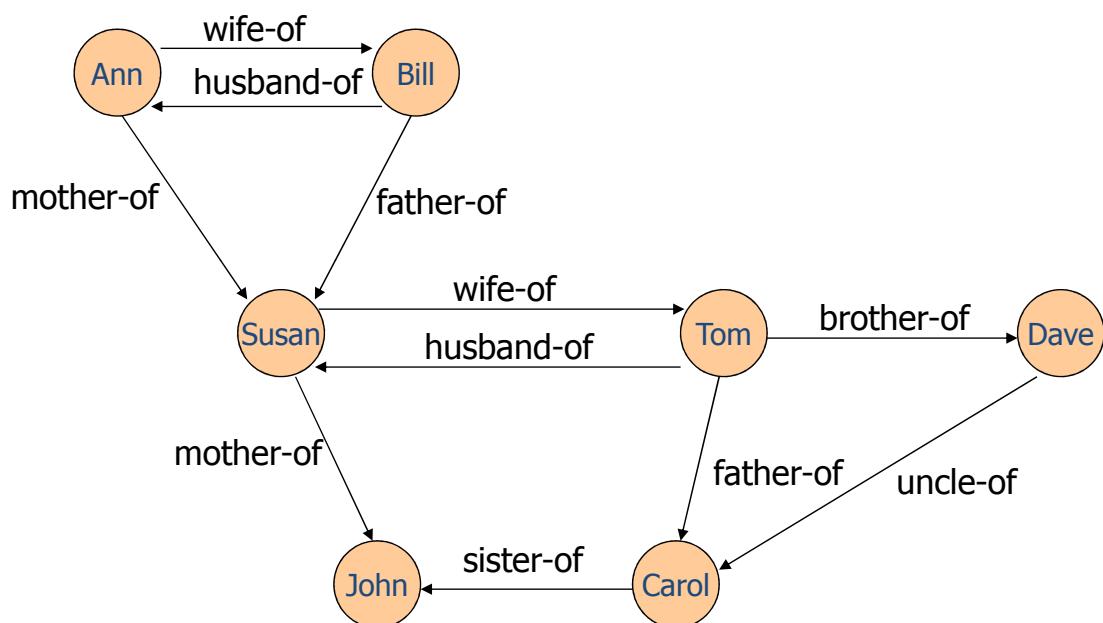
Link AKO digunakan untuk merelasikan satu jenis objek ke jenis objek lainnya. AKO juga akan menghubungkan jenis individual ke jenis induk dari jenis dimana individual merupakan anak dari jenis tersebut.

Objek didalam jenis/kelas memiliki satu atau lebih atribut secara umum. Setiap atribut memiliki nilai, gabungan atribut dan nilai disebut **properti**.

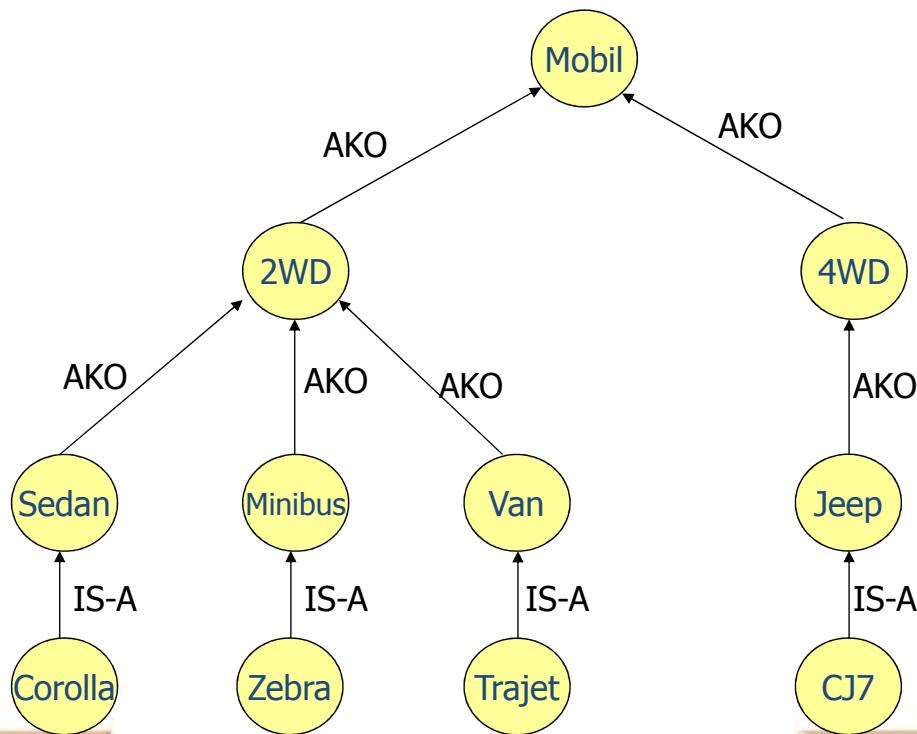
# Contoh Jaringan Umum



# Contoh Jaringan Semantik



## Contoh Jaringan Semantik dengan IS-A & AKO



## OAV dlm Jaringan Semantik

Object-attribute-value triple (OAV) atau triplet dapat digunakan untuk memberi karakter semua pengetahuan dalam jaringan semantik dan digunakan dalam sistem pakar MYCIN untuk diagnosa penyakit infeksi.

Representasi triple OAV sangat sesuai untuk pembuatan daftar pengetahuan dalam bentuk tabel dan menterjemah kan tabel ke dalam code komputer dengan induksi baris.

Objek	Attribute	Value
Apel	warna	Merah
Apel	kuantitas	100
Anggur	warna	Ungu
Anggur	tipe	Tanpa biji

## Keterbatasan Jaringan Semantik

1. Kesulitan untuk membuat standar nama link, sehingga berakibat sulit memahami suatu desain jaringan semantik, untuk apa dibuat dan bagaimana dibuat
2. Jaringan semantik, aslinya diusulkan sebagai memori gabungan manusia dimana satu node punya link ke node lainnya, namun ada sekitar  $10^{10}$  neuron dan  $10^{15}$  link dalam pikiran manusian, kalau semua dipetakan dengan jaringan semantik akan diperlukan waktu yang sangat lama untuk mendapat jawaban, terutama untuk pertanyaan negatif (yang sebenarnya tidak perlu dijawab)
3. Jaringan semantik secara logika tidak memadai karena tidak menentukan pengetahuan dengan cara yang dapat dilakukan oleh logika.
4. Jaringan semantik secara heuristik tidak memadai karena tidak ada cara untuk memasukkan informasi heuristik dalam jaringan untuk mengefisienkan kerja jaringan.

## REPRESENTASI PENGETAHUAN

### (bagian 2)

Pertemuan ke lima

# Schemata

Dalam AI, bentuk skema (schema, schemas, shematas) digunakan untuk menjelaskan struktur pengetahuan yang lebih kompleks dibandingkan dengan jaringan semantik.

Bentuk skema berasal dari psikologi dimana menunjukkan organisasi pengetahuan yang kontinyu atau merespond suatu stimuli. Yaitu seperti menciptakan pelajaran hubungan tidak formal antara sebab dan akibat atau cara mengatasinya, penyebab akan diulangi jika menyenangkan atau dihindari jika menyakitkan.

Seseorang tidak perlu memikirkan pengetahuan untuk mengetahui bagaimana atau mengapa melakukan sesuatu yang memang sulit untuk dijelaskan.

# Concept Schema

Concept schema merupakan tipe lain dari skema yang dengan skema jenis ini kita dapat menunjukkan konsep.

Jika tiap orang ditanya tentang konsep sesuatu, jawabannya bisa berbeda-beda, karena tiap orang memiliki stereotypes dalam konsep pikirannya masing-masing.

Skema konseptual adalah abstraksi dimana obyek khusus diklasifikasikan dengan properti umum.

Dengan memfokuskan pada property umum dari obyek, maka akan lebih mudah memberi alasan tentangnya tanpa menjadi membingungkan dengan detail yang tidak relevan.

## Perbedaan Schema & Semantik

Secara umum, schema memiliki struktur internal di dalam node nya, sedangkan jaringan semantik tidak.

Tabel jaringan semantik seluruhnya adalah pengetahuan tentang node. Jaringan semantik seperti struktur data dalam pengetahuan komputer dimana kunci penelitian juga merupakan data yang disimpan dalam node.

Skema adalah seperti struktur data diamana node berisi record, setiap record mungkin berisi data tunggal, record lain atau pointer ke node yang lainnya.



## Frames

Diajukan sebagai metode untuk melihat pemahaman bahasa natural dan bidang lain, frame menyediakan struktur yang cocok untuk menunjukkan objek yang tipikal pada situasi tertentu yang diberikan, seperti stereotype.

Jika jaringan semantik pada dasarnya adalah representasi dua dimensi dari pengetahuan, frame menambahkan dimensi ketiga dengan memungkinkan node untuk mempunyai struktur. Struktur ini dapat berupa nilai sederhana atau frame yang lainnya.

Karakteristik pokok dari frame adalah bahwa frame tersebut menunjukkan pengetahuan yang dihubungkan dengan subyek yang sempit yang mempunyai banyak pengetahuan default.

Frame merupakan kebalikan dari jaringan semantik yang pada umumnya digunakan untuk representasi pengetahuan yang luas

Frame dapat dianalogikan sebagai struktur record pada bahasa tingkat tinggi seperti Pascal.

Berhubungan dengan bidang dan nilai record, dalam frame terdapat slot dan filler yang menentukan stereotype dari suatu objek. Dalam bentuk OAV, slot berhubungan dengan atribut dan filler berhubungan dengan nilai.

Kelebihan frame adalah dalam sistem frame berlaku sifat hirarki dan pewarisan. Dengan menggunakan frame dalam slot dan filler serta pewarisan, maka kita dapat membuat suatu representasi pengetahuan yang kuat.

Khususnya expert system yang berdasarkan pada frame sangat berguna untuk menunjukkan pengetahuan tidak formal karena informasinya disusun berdasarkan pada sebab dan akibat

## Contoh Frame untuk obyek Mobil

Slot	Filler
Pabrik	Toyota Astra
Model	Corolla Altis
Tahun	2002
Transmisi	Automatic
Mesin	Bensin
Roda	4
Warna	Silver

## Procedural attachments pada Frame

Filler mungkin berupa nilai seperti properti dalam nama slot, atau rentang nilai dalam type slot. Slot mungkin berisi prosedur yang dihadapkan pada slot, atau disebut dengan procedural attachments, yang biasanya terdiri dari tiga type, yaitu :

1. If-needed, merupakan prosedur yang dibuat jika nilai filler yang diperlukan tidak ada atau nilai default tidak sesuai
2. If-added, merupakan prosedur yang dibuat jika akan menambahkan nilai ke slot
3. If-removal, akan digunakan jika suatu nilai dipindahkan atau dikeluarkan dari slot.

## Contoh Generic Frame untuk Property

Slot	Filler
Nama	Properti
Spesialisasi dari	Jenis Obyek
Tipe	(rumah, mobil, kapal) Bila ditambah Prosedur : Add_Property
Pemilik	Default : Pribadi Bila diperlukan Prosedur : Find_Pemilik
Lokasi	(rumah, kantor, berpindah)
Status	(bagus, jelek, hilang)
Dalam Garansi	(ya, tidak)

## Contoh Frame Mobil – Subframe Generic Properti

Slot	Filler
Nama	Mobil
Spesialisasi dari	Jenis properti
Tipe	(sedan, van, minibus)
Pabrik	(Toyota, Honda, Mitsubishi)
Lokasi	Berpindah
Roda	4
Transmisi	(manual, automatic)
Mesin	(bensin, diesel, gas, tenaga surya)

## Contoh Frame mobil

Slot	Filler
Nama	Mobil Heru
Spesialisasi dari	Adalah sedan
Pabrik	Toyota
Pemilik	Heru Sutimbul
Transmisi	Manual
Mesin	Bensin
Status	Bagus
Dalam Garansi	Ya

## Kalsifikasi frame berdasarkan aplikasinya

1. Situational frame, berisi pengetahuan tentang bagaimana terjadinya situasi yang diinginkan.
2. Action frame, berisi slot yang menentukan aksi yang akan dilakukan dalam situasi yang diberikan.
3. Casual knowledge frame, gabungan antara situational dan action frame yang digunakan untuk menjelaskan hubungan sebab dan akibat.

## Kelemahan Frame

- Pengetahuan dalam frame tidak dapat dipertimbangkan menjadi definisi dari suatu obyek.
- Slot pada frame dapat dikurangi, ditambah atau bahkan dirubah yang juga berarti merubah atribut dari suatu frame obyek, jika ini terjadi pada suatu frame generic yang memiliki sub frame maka akan menyebabkan sifat pewarisan atribut menjadi tidak valid atau dapat dibatalkan karena propertinya tidak lagi bersifat umum (primitif).

# Representasi Pengetahuan

## (Bagian 3)

# Logika dan Himpunan

Pertemuan ke enam

Program Studi  
Teknik Informatika

STMIK NUSA MANDIRI

COPYRIGHT © Maret 2013

## Syllogisme

Adalah logika formal pertama yang dikembangkan oleh filsuf Yunani, **Aristotle** pada abad ke-4 SM.

Syllogisme mempunyai dua *premises* dan satu *conclusion*.

Premise adalah proporsi atau pernyataan yang selalu bernilai benar atau salah karena berdasarkan fakta.

Conclusion adalah kesimpulan yang diturunkan dari dua pernyataan sebelumnya.

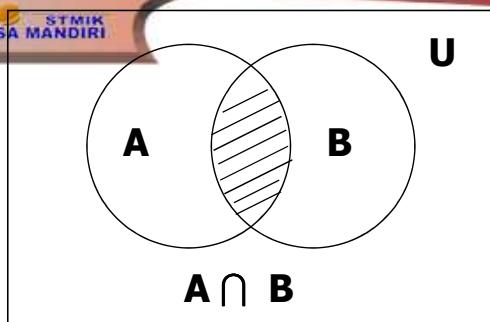
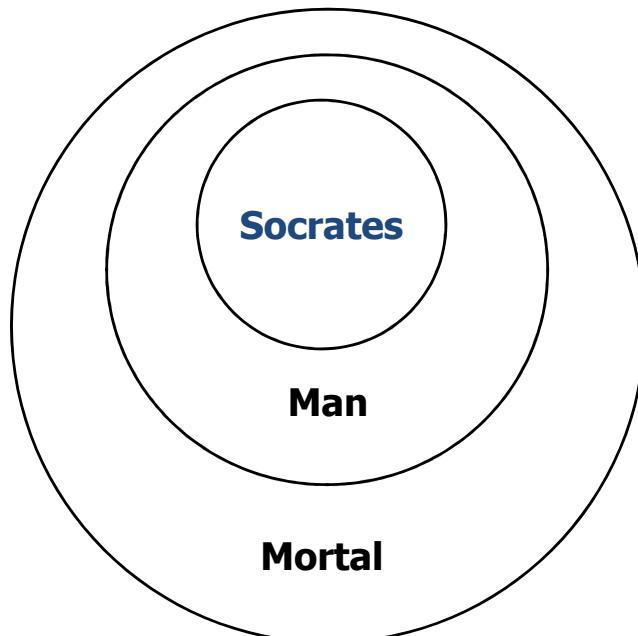
Contoh :

premise : All man are mortal

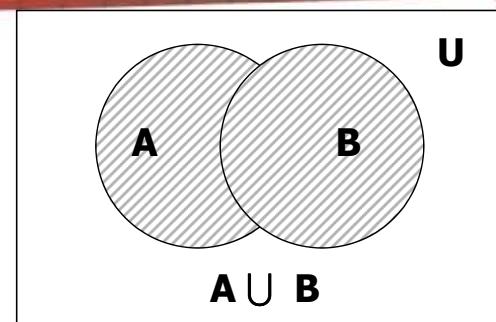
premise : Socrates is a man

Conclusion : Socrates is mortal

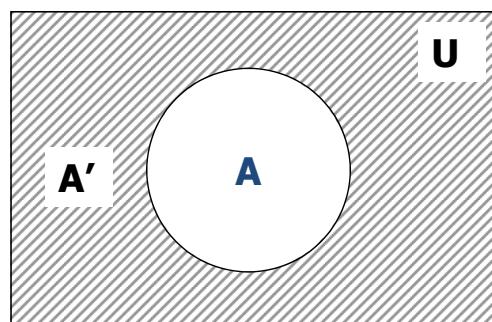
# Diagram Venn



Irisan / Intersection



Gabungan / Union



Selain / Complement

## Symbolic Logic

Diperkenalkan pertama kali oleh GW Leibnitz abad ke-17 dan disempurnakan oleh ahli matematika Inggris, George Boole yang menerbitkan bukunya tentang symbolic logic pada tahun 1897.

Konsep baru yang diperkenalkan Boole adalah memodifikasi pandangan Aristotle tentang subyek yang harus memiliki keberadaan (existensial import).

Dalam **pandangan modern** Boole dapat menyebutkan subyek yang tidak ada atau jenis yang tidak ada elemennya (himpunan kosong) sebagai premises.

Co/: All mermaids swim well

## Aksioma

Kontribusi Boole yang lain adalah serangkaian aksioma, yang berisi symbol untuk menunjukkan obyek dan jenis, dan operasi aljabar untuk memanipulasi symbol.

Aksioma merupakan definisi fundamental dari sistem logika seperti matematika dan logika itu sendiri. Dengan hanya menggunakan aksioma dapat menghasilkan teori.

Teori adalah pernyataan yang dapat dibuktikan dengan menunjukkan bagaimana teori tersebut diperoleh, yaitu dengan menggunakan aksioma.

## Logika Proporsional

Kadang disebut sebagai proportional calculus, merupakan logika simbol untuk memanipulasi proporsi, khususnya yang berhubungan dengan manipulasi variabel logika yang mewakili atau menunjukkan suatu proporsi.

Bentuk lain yang digunakan untuk logika proporsional adalah statement calculus atau sentential calculus, dimana statement/sentence atau kalimat pada umumnya dapat diklasifikasikan menjadi 4 type, yaitu :

1. Imperatif / perintah
2. Interogatif / pertanyaan
3. Kalimat seru
4. Deklaratif / pernyataan

Logika proporsional dihubungkan dengan kalimat-kalimat deklaratif yang dapat diklasifikasi sebagai pernyataan benar atau salah. Suatu kalimat deklaratif yang memiliki nilai benar atau salah yang pasti atau dapat ditentukan disebut dengan ***statement/pernyataan*** atau ***proposition/proposisi***. Suatu pernyataan disebut juga sebagai ***closed sentence*** (kalimat tertutup) karena kebenarannya tidak perlu dipertanyakan lagi.

Contoh : Bujursangkar memiliki empat sisi yang sama (pasti)  
Harimau berkaki empat (benar / salah)

Berikut contoh yg tidak termasuk proposisi :

Durian enak sekali (kebenarannya relatif)  
Orang itu tinggi (kalimat terbuka)

# Compound statement

Adalah pernyataan yang dibuat dengan cara menggabungkan atau menghubungkan beberapa pernyataan tunggal menggunakan konektor logika, spt :

Konektor	Arti
$\wedge$	AND; konjungsi
$\vee$	OR; disjungsi
$\sim$	NOT; negasi
$\rightarrow$	If . . . Then; kondisional
$\leftrightarrow$	If and only if; bikondisional

Tabel kebenaran logika compound statement

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

**Tautology**, adalah pernyataan gabungan yang selalu bernilai benar, dimana pernyataan individualnya benar atau salah, mis :  $p \vee \sim p$

**Contradiction**, adalah pernyataan gabungan yang selalu bernilai salah, dimana pernyataan individualnya benar atau salah, mis :  $p \wedge \sim p$

## Contoh pernyataan kondisional

$p \rightarrow q$  dapat diterjemahkan kebahasa natural menjadi :

- p menyatakan q
- jika p, maka q
- p, hanya jika q
- p cukup untuk q
- q jika p
- q dengan syarat p

p : anda berusia 18 tahun atau lebih

q : anda berhak memilih

Kondisional  $p \rightarrow q$  dapat berarti :

Anda berusia 18 tahun atau lebih menyatakan anda berhak memilih

Jika anda berusia 18 tahun atau lebih maka anda berhak memilih

Anda berusia 18 tahun atau lebih, hanya jika anda berhak memilih

Anda berusia 18 tahun atau lebih adalah cukup (memenuhi syarat) bagi anda untuk memilih

Anda berhak memilih jika anda berusia 18 tahun atau lebih

Anda berhak memilih dengan syarat anda berusia 18 tahun atau lebih

## Logika Predikat order pertama

Problem utama logika proporsional adalah tidak memiliki batasan dan hanya dapat dihubungkan dengan kalimat yang lengkap, yaitu tidak dapat menguji struktur internal suatu pernyataan. Logika proporsional tidak dapat menguji validitas sylogisme spt :

All humans are mortal

All man are humans

Therefor, all women are mortal

Untuk menganalisa kasus lebih luas, dikembangkan logika predikat, yang dihubungkan dengan penggunaan kata khusus yang disebut quantifiers, spt : *all*, *some* dan *no* yang secara eksplisit memberi kuantitas kata lain dan membuat suatu kalimat lebih nyata. Seluruh quantifier dihubungkan dengan how many shg penya cakupan lebih luas dari logika proportional.

## Quantifier Universal (All : V)

Digunakan untuk kalimat yang diberi kuantitas memiliki nilai kebenaran yang sama untuk semua pengganti / elemen dalam domain yang sama.

Mis :  $(V x) (x + x = 2x)$

jika pernyataan  $x + x = 2x$  diganti dengan p, jadinya

$(V x) (p)$

Jika p adalah kalimat "All triangles are polygon", ditulis

$(V x) (\text{if } x \text{ is triangle} \rightarrow x \text{ is polygon})$

dipersingkat dengan predikat function menjadi :

$(V x) (\text{triangel}(x) \rightarrow \text{polygon}(x))$

Kesamaan logika :

$(V x) P(x) \equiv P(x_1) \wedge P(x_2) \wedge P(x_3) \dots P(x_n)$

## Quantifier Eksistensi (some : 3)

Menjelaskan suatu pernyataan yang benar untuk minimal satu anggota domain.

Quantifier eksistensi dapat dibaca atau ditulis dalam bahasa natural menjadi :

there exist, at least one, for some, there is one, some

Untuk menyatakan "*some elephants has three-legged*" :

$$(\exists x) (\text{Elephant}(x) \wedge \text{three-legged}(x))$$

Kesamaan logika

$$(\exists x) P(x) \equiv P(x_1) \vee P(x_2) \vee P(x_3) \dots P(x_n)$$

## Quantifier dan Himpunan

Ekspresi Himpunan	Kesamaan Logika Quantifier
$A = B$	$\forall x (x \in A \leftrightarrow x \in B)$
$A \subset B$	$\forall x (x \in A \rightarrow x \in B)$
$A \cap B$	$\forall x (x \in A \wedge x \in B)$
$A \cup B$	$\forall x (x \in A \vee x \in B)$
$A'$	$\forall x (x \in \mathcal{V} \mid \sim(x \in A))$
$\mathcal{V}$ (Universe)	T (true)
$\emptyset$ (himpunan kosong)	F (false)

# Metode Inferensi

# GRAPH, TREES, LATTICES

Pertemuan ke sembilan

## Pengertian Umum

### Graph

Adalah suatu bentuk geometri yang menghubungkan titik-titik (node) dengan garis/tanda panah (arch).

### Tree

Bentuk Graph berarah, terbuka dimana selalu memiliki satu node sebagai root/akar atau start, node-node yang memiliki cabang dan node-node yang tidak memiliki cabang yang disebut leave/daun atau end. Root secara otomatis akan menjadi parent bagi node dibawahnya dan begitu juga untuk setiap node yang bukan daun akan menjadi parent untuk node dibawahnya.

## Lattice

Bentuk Graph berarah, bisa terbuka bisa tertutup, keunikan lattice dibanding Tree adalah bahwa Lattice dapat memiliki lebih dari satu node sebagai start, dan adanya kemungkinan diamana sebuah node memiliki lebih dari satu parent. Keunikan yang lain bahwa pada Lattice dapat terjadi hanya ada satu node sebagai leave/daun.

## Struktur Keputusan

Tree dan Lattice sangat berguna untuk mengklasifikasi obyek karena hirarki alamiahnya dari induk atas anak, contoh : family tree.

Aplikasi Tree dan Lattice yang lain adalah dalam membuat gambaran geometris proses pengambilan keputusan, yang dikenal dengan decision tree dan decision lattice, atau secara umum disebut sebagai decision structure / struktur keputusan.

Struktur keputusan adalah skema representasi pengetahuan dan metode pemberian alasan tentang pengetahuan itu.

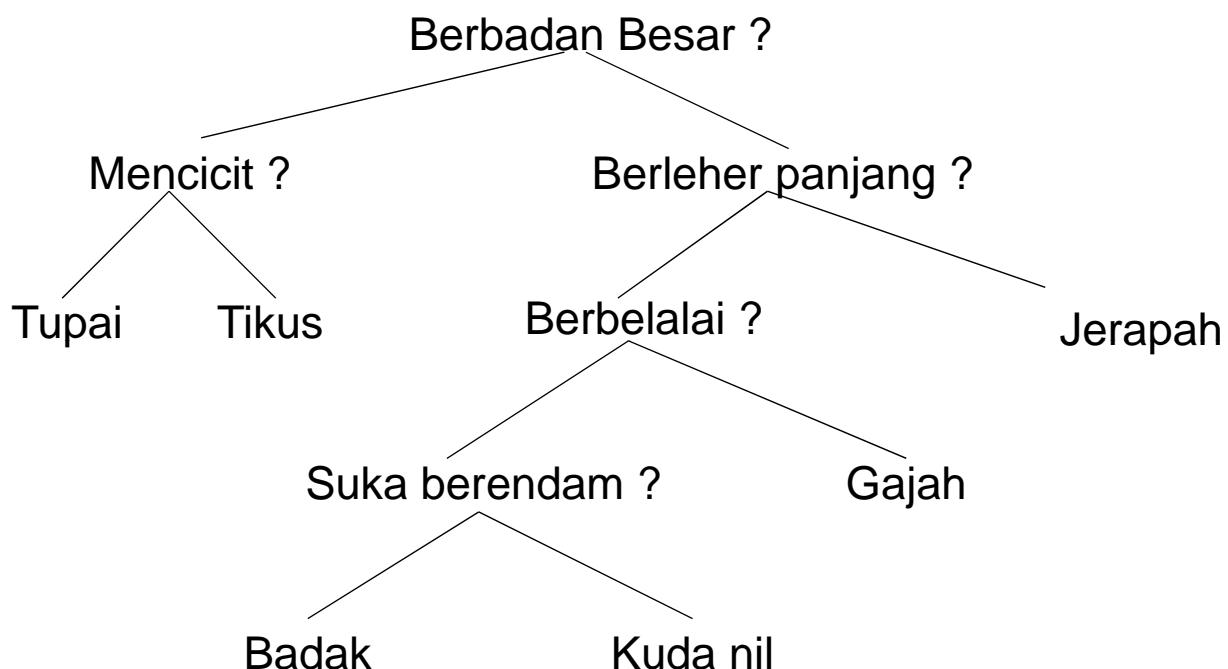
## Binary Tree

Jika suatu keputusan adalah binary (memiliki tepat 2 kemungkinan jawaban), maka tree keputusan binary akan mudah dibuat dan sangat efisien.

Dimulai dengan pertanyaan sebagai root, pilihan jawaban (ya atau tidak) sebagai arch/link, setiap pertanyaan berikutnya sebagai simpul bukan daun dan setiap kesimpulan atau jawaban akhir sebagai simpul bukan daun.

Dengan binary tree kita akan mendapatkan maksimum  $2^n$  kesimpulan untuk sebanyak  $n$  pertanyaan.

### Contoh Tree untuk Menebak Binatang



## Struktur dan Baris Produksi

Struktur keputusan dapat secara mekanis diterjemahkan ke dalam baris produksi, hal ini dapat dilakukan dengan mudah dengan mengamati struktur dan pembuatan baris IF . . . THEN pada setiap node nya.

Contoh dari tree diatas diterjemahkan kedalam baris sbb:

If Question = "Berbadan besar ?" and Response = "Tidak"  
then question := "Apakah mencicit?"

If Question = "Berbadan besar ?" and response = "Ya"  
then question := "Apakah berleher panjang ?"

dst

## Ruang Stata

Stata adalah koleksi karakteristik yang dapat digunakan untuk menentukan suatu keadaan atau menyatakan suatu obyek tertentu.

Ruang stata merupakan rangkaian pernyataan yang menunjukkan transisi antara pernyataan yang menyatakan bagaimana suatu keadaan dapat terjadi atau bagaimana suatu obyek dapat dicapai.

Untuk menggambarkan ruang stata pada umumnya akan didapatkan bentuk Lattice, atau sering juga disebut sebagai finite state machine (mesin stata hingga) karena menyatakan jumlah keadaan terbatas yang dapat dicapai dengan aturan produksi/mesin yang juga terbatas, atau menggunakan istilah yang lebih umum, diagram keadaan

## Contoh Ruang Stata

### Mesin Minuman Ringan

Diketahui harga minuman ringan 1200, mesin minuman hanya dapat menerima koin dengan pecahan 500 atau 200. Dapat digambarkan diagram keadaan dengan satu node sebagai simbol start dan satu node sebagai simbol end/penerima atau sukses yang digambar dengan lingkaran bergaris ganda dan node lain untuk keadaan yang bukan start atau sukses dengan lingkaran bergaris tunggal. Untuk arch/panah diberi keterangan dengan pecahan koin yang mungkin (untuk memudahkan gunakan simbol L untuk pecahan 500 dan D untuk pecahan 200).

\* Harap digambarkan diagramnya

## Contoh lain Ruang Stata

### Untuk problem Kera dan Pisang

Diketahui seekor kera berada dalam ruangan, dimana kera berusaha untuk mengambil pisang yang digantung dilangit-langit ruangan. Diruangan terdapat sebuah kursi dan sebuah tangga, pisang hanya akan dicapai oleh kera jika kera menggunakan tangan.

### Penyelesaian:

Kasus ini dimulai dalam keadaan yang bervariasi sehingga kita tidak menggambarkan simbol start untuk kasus seperti ini, hanya simbol end atau suksesnya saja yang dibuat selain node yang lain.

\*harap digambarkan diagram keadaanya

## Struktur AND-OR

Sutau type Tree dan Lattice yang berguna untuk menunjukkan masalah dengan rangkaian backward adalah struktur AND-OR.

Cara penggambaran metode backward dengan Struktur AND-OR adalah dengan menempatkan tujuan keadaan atau obyek yang akan dicapai sebagai root/akar dan kemudian menempatkan kemungkinan-kemungkinan untuk mencapai tujuan dibawahnya.

Untuk kondisi OR adalah sebuah node memiliki lebih dari satu cabang kebawah yang dapat dipilih (arch terpisah)

Untuk kondisi AND adalah sebuah node memiliki lebih dari satu cabang yang harus terpenuhi semuanya (arch digabungkan dengan garis lengkung)

## Contoh Struktur AND-OR

Tujuan : Menuju tempat kerja

Ketempat kerja dapat dicapai dengan jalan kaki, naik bus atau naik motor. Jalan kaki dapat dilakukan jika kaki cukup sehat.

Untuk naik bus bisa menuggu di halte atau jalan ke terminal.

Untuk naik motor harus dipastikan motor dalam kondisi baik dan ada bensinnya, untuk memastikan kondisi motor baik harus diperiksa sendiri atau orang lain yang memeriksa, untuk mendapatkan bensin membawa motor ke pom bensin atau membeli bensin tanpa motor, yang berarti diperlukan jerigen untuk membawa bensin.

# Metode Inferensi

## Logika Deduktif & Sylogisme

Pertemuan ke sepuluh

Umum

Salah satu dari banyak metode yang paling sering digunakan untuk menggambarkan inferensi adalah deduktive logic (logika deduktif), yang digunakan sejak awal untuk menentukan validitas dari *Argumen*.

Satu type argumen logika adalah Sylogisme, yang memiliki dua premises dan satu conclusion.

Dalam argumen, premises digunakan sebagai bukti untuk mendukung conclusion (kesimpulan). Premises disebut juga dengan *antecedent* dan kesimpulan disebut *consequent*.

Karakteristik pokok dari logika deduktif adalah bahwa kesimpulan benar harus mengikuti dari premises yang benar.

## Penulisan Argumen

Argumen harus ditulis dalam bentuk yang lebih singkat seperti :

Anyone who can program is intelligent

John can program

---

∴ Jhon is intelligent

Dimana simbol  $\therefore$  digunakan untuk menyatakan berarti/jadi (therefor) dan garis lurus digunakan untuk memisahkan conclusion dari premises

## Catagorical Syllogism

Pada umumnya, sylogisme merupakan argumen deduktif yang valid yang mempunyai dua premises dan satu conclusion. Sylogisme klasik merupakan type yang disebut sebagai catagorical syllogisme, dimana premises dan conclusion ditentukan sebagai pernyataan catagorical dari empat bentuk berikut :

Bentuk	Skema	Pengertian
A	All S is P	Universal affirmative
E	No S is P	Universal negative
I	Some S is P	Particular affirmative
O	Some S is not P	Particular negative

Dalam logika, skema kata digunakan untuk menunjukkan bentuk esensial dari argumen. Skema juga menunjukkan bentuk logika dari seluruh sylogisme seperti dalam bentuk berikut :

All M is P

All S is M

•• All S is P

Subject pada conclusion (S) disebut **minor term**, dan predikat pada conclusion (P) disebut **major term**.

Premis yang berisi minor term disebut **minor premise** yang berisi major term disebut **major premise**

### Contoh :

Major Premise : All M is P

Minor Premise : All S is M

Conclusion : All S is P

Adalah sylogisme yang disebut sebagai standard form dengan premise major dan minor yang ditentukan. Subject merupakan sesuatu yang digambarkan/ dijelaskan, sedangkan predicate menggambarkan beberapa property / sifat dari subject.

Contoh lain :

- All microcomputers are computers
- All microcomputers with 512 megabytes are computers with a lot of memory

## Middle Term

Adalah bentuk ketiga dalam sylogisme yang pada umumnya terdapat dikedua premises. Middle term memiliki kedudukan yang penting dalam sylogisme karena dalam sylogisme, kesimpulan tidak dapat di turunkan dari satu premis saja, sehingga middle term pada umumnya akan berfungsi sebagai penghubung dari dua premis

## Mood Sylogisme

Mood dari syllogisme ditentukan dengan tiga huruf yang memberikan bentuk major premise, minor premise dan kesimpulan secara respektif, berdasarkan tabel kategaori pernyataan.

Contoh :

All M is P

All S is M

• • All S is P

disebut memiliki mood AAA.

\* Beri contoh mood lain

## Type Sylogisme

Type dari sylogisme ditentukan berdasarkan mood dan figure dari penyusunan minor term (S), major term (P) dan middle term (M), dimana akan ada empat kemungkinan figure susunan S,P dan M, yaitu :

	<i>Figure 1</i>	<i>Figure 2</i>	<i>Figure 3</i>	<i>Figure 4</i>
Major premise	M P	P M	M P	P M
Minor Premise	S M	S M	M S	M S

Contoh :

Bentuk :

All M is P

All S is M

•• All S is P

Disebut bertipe AAA-1

Bentuk :

All M is P

No S is M

•• No S is P

Disebut bertipe AEE-1

dst

## Validitas Argumen

Untuk membuktikan validitas argumen sylogisme, digunakan dicision procedure, yaitu suatu metode mekanik umum atau algoritma yang dapat menentukan validitas secara otomatis.

Decision procedure untuk sylogisme dapat dikerjakan dengan menggunakan diagram Venn dengan tiga lingkaran yang saling beririsan dimana tiap lingkaran mewakili minor term (S), major term (P) dan middle term (M).

## Algoritma penggambaran diagram venn

- Beri arsiran untuk daerah yang tidak digunakan / diabaikan
- Pernyataan bentuk universal (A dan E) selalu dikerjakan lebih dulu dari pernyataan particular (O dan I)
- Jika kedua premise universal atau kedua premise particular maka kerjakan major premise dulu
- Untuk daerah yang diwakili dengan some beri tanda \*
- Jika ada dua daerah berdampingan yang harus diberi tanda \*, maka tanda \* diletakkan pada garis yang membatasi kedua daerah tsb
- Untuk daerah yang sudah diarsir tidak dapat lagi diberi tanda \*

## Contoh

Bentuk AEE-1

All M is P

No S is M

∴ No S is P

Bukan sylogisme valid, dengan contoh :

All microcomputers are computers

No mainframe is a microcomputer

∴ No mainframe is a computer

\* Buktikan dengan digram Venn

## Contoh

Bentuk EAE-1

No M is P

All S is M

∴ No S is P

Sylogisme valid, dengan contoh :

No microcomputers is mainframe

All laptops is microcomputers

∴ No laptops mainframe

\* Buktikan dengan digram Venn

## Contoh

Bentuk IAI-4

Some P are M

All M are S

∴ Some S are P

Sylogisme valid, dengan contoh :

Some computers are laptops

All laptops are transportable

∴ Some transportable are computers

\* Buktikan dengan digram Venn

## Metode Inferensi

# Argumen & Logika Proporsional

Adalah argumen yang berisi proporsi, sehingga dapat diekspresikan menjadi bentuk formal dalam logika proporsional.

Contoh :

If there is power, the computer will work

There is power

---

∴ The computer will work

Dapat diekspresikan dengan huruf (logika proporsional) :

$p = \text{There is power}$

$q = \text{The computer will work}$

Sehingga menjadi :

$p \rightarrow q$

$$\begin{array}{c} p \\ \hline \therefore q \end{array}$$

Skema inferensi untuk proporsional diatas disebut dengan berbagai istilah : *Direct Reasoning, modus ponens, law of detachment* atau *assuming the antecedent*

## Notasi modus ponens

Notasi lain untuk skema modus ponens :

$$p \rightarrow q, p; \therefore q$$

dimana koma digunakan untuk memisahkan premise dan titik koma untuk memisahkan conclusion dari premise.

Bentuk umumnya :

$$P_1, P_2, \dots, P_n; \therefore C$$

dimana P = premise dan C = Conclusion

## Kesamaan Logika Pada modus ponens

Perhatikan lagi bentuk :

$$p \rightarrow q, p; \therefore q$$

Dalam logika tanda koma (,) setara dengan notasi  $\wedge$  (AND) dan tanda titik koma (;) setara dengan notasi  $\rightarrow$  (then), sehingga skema diatas dapat ditulis menjadi :

$$(p \rightarrow q) \wedge p \rightarrow q$$

- \* Tanda kurung digunakan karena notasi  $\rightarrow$  (then) memiliki hirarki lebih rendah dari notasi  $\wedge$  (AND)

## Validitas modus ponens

Karena modus ponens adalah bentuk khusus dari syllogisme, maka suatu modus ponens dikatakan valid jika untuk kedua premise bernilai benar (T) maka nilai conclusion juga benar (T).

Tabel kebenaran untuk modus ponens diatas :

<b>p</b>	<b>q</b>	<b><math>p \rightarrow q</math></b>	<b><math>(p \rightarrow q) \wedge p</math></b>	<b><math>(p \rightarrow q) \wedge p \rightarrow q</math></b>
<b>T</b>	<b>T</b>	<b>T</b>	<b>T</b>	<b>T</b>
<b>T</b>	<b>F</b>	<b>F</b>	<b>F</b>	<b>T</b>
<b>F</b>	<b>T</b>	<b>T</b>	<b>F</b>	<b>T</b>
<b>F</b>	<b>F</b>	<b>T</b>	<b>F</b>	<b>T</b>

## Argumen Palsu

Perhatikan modus ponens berikut :

If there are no bugs, then program compiles

There are no bugs

---

∴ The Program compiles

Bandingkan dengan modus ponens berikut :

If there are no bugs, then the program compiles

The program compiles

---

∴ There are no bugs

Salah satunya palsu, dapat dibuktikan dengan tabel kebenaran logika

## Aturan Inferensi untuk logika yang benar

1. Hukum data skema :  $p \rightarrow q$   
 (modus ponendo ponens)  $\frac{p}{\therefore q}$

2. Hukum kontra positif :  $\frac{p \rightarrow q}{\therefore \sim q \sim p}$

3. Hukum modus tollen :  $p \rightarrow q$   
 (modus tollendo tollens)  $\frac{\sim q}{\therefore \sim p}$

4. Aturan Rangkaian :  $\frac{p \quad q}{\therefore p \quad r}$   
 (hukum sylogisme)  $\frac{q \quad r}{\therefore p \quad r}$

5. Hukum inferensi disjunktif :  $p \vee q$   
 (modus tollendo ponens)  $\frac{\sim p}{\therefore q} \quad \frac{\sim q}{\therefore p}$

6. Hukum negasi ganda :  $\frac{\sim(\sim p)}{\therefore p}$

7. Hukum penyederhanaan :  $\frac{p \wedge q}{\therefore p} \quad \frac{p \vee q}{\therefore q}$

### 8. Hukum Konjungsi :

$$\frac{p \\ q}{\therefore p \wedge q} \qquad \frac{p}{\therefore p \vee q}$$

### 9. Hukum De Morgan(1)

$$\frac{\sim(p \vee q)}{\therefore \sim p \vee \sim q} \qquad \frac{\sim(p \vee q)}{\therefore \sim p \wedge \sim q}$$

### 10. Hukum De Morgan (2) :

$$\frac{\sim(p \wedge q) \\ p}{\therefore \sim q} \qquad \frac{\sim(p \wedge q) \\ q}{\therefore \sim p}$$

## Kondisional dan variant-nya

Kondisional	$p \rightarrow q$
Konversi	$q \rightarrow p$
Inversi	$\sim p \rightarrow \sim q$
Kontra positif	$\sim q \rightarrow \sim p$

Inferensi pada argumen dgn lebih dari 2 premises

Perhatikan contoh berikut :

Harga chip naik hanya jika nilai Yen naik.

Nilai Yen naik hanya jika nilai dollar turun dan  
jika nilai dollar turun maka nilai yen naik.

Ketika harga chip naik,  
nilai dollar menurun.

Proporsi dapat ditentukan sbb :

C = harga chip naik

Y = nilai Yen naik

D = nilai dollar turun

Sehingga argumen diatas dapat ditulis :

$$C \rightarrow Y$$

$$(Y \rightarrow D) \wedge (D \rightarrow Y)$$

$$\frac{C}{\therefore D}$$

Disederhanakan menjadi :

$$C \rightarrow Y$$

$$Y = D$$

$$\frac{C}{\therefore D}$$

Disubstitusi menjadi :

$$C \rightarrow D$$

$$\frac{D}{\therefore D}$$

Memenuhi hukum modus ponens yang valid

## Batasan Logika Proporsional

Logika proporsional tidak dapat mengiferensi atau membuktikan kebenaran sylogisme klasik, seperti :

All men are mortal  
Socrates is a man  
therefore, Socrates is mortal

Sebelum dirubah menjadi sylogisme proporsional seperti :

If Socrates is a man, then Socrates is mortal  
Socrates is a man  
therefor, Socrates is mortal

## Metode Inferensi

# Rangkaian Forward & Backward

## Umum

### Rangkaian

merupakan salah satu metode inferensi yang berusaha menghubungkan masalah dengan solusinya.

### Rangkaian Forward

rangkaian yang menghubungkan keadaan/fakta yang dihadapi sebagai alasan kepada suatu kesimpulan.

### Rangkaian Backward

rangkaian yang menghubungkan suatu keadaan/fakta yang dihadapi sebagai suatu hipotesa kepada fakta-fakta lain yang dapat mendukung hipotesa tersebut

### Contoh kasus Forward

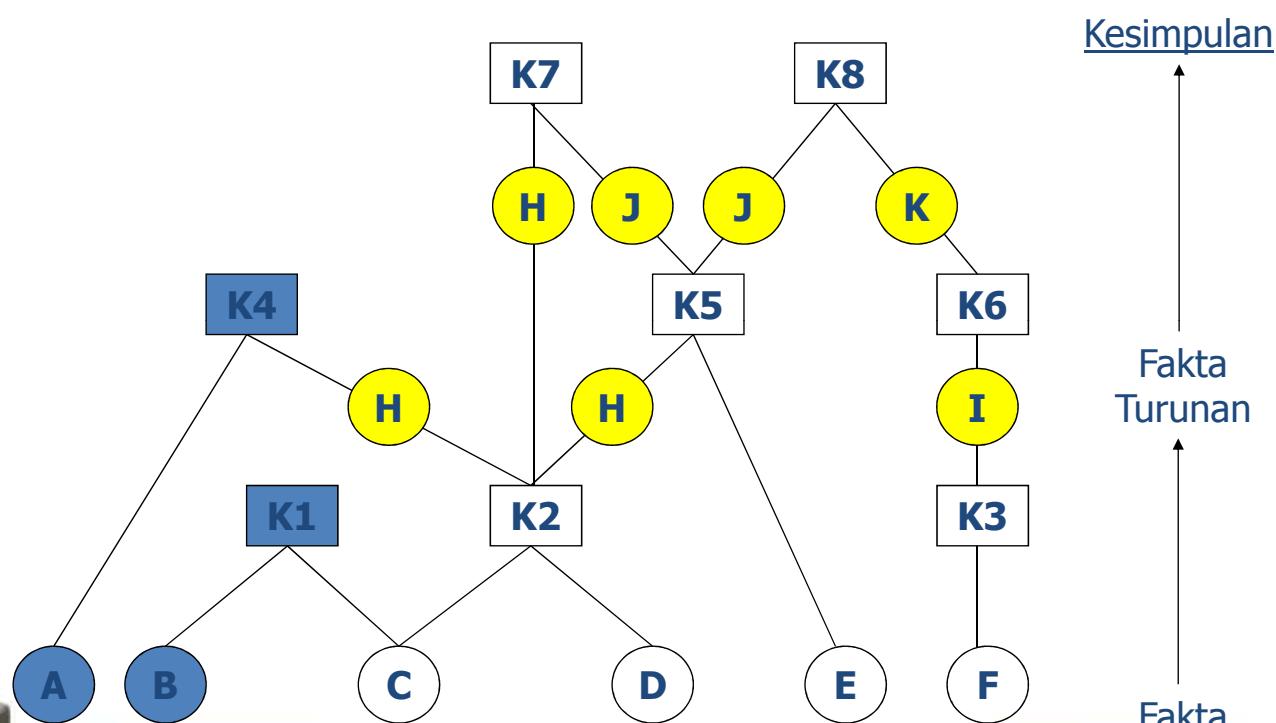
Pada suatu keadaan kita sedang mengendarai mobil, kemudian diikuti mobil polisi dengan serine dan lampu menyalा. Dengan rangkaian forward kita dapat berkesimpulan polisi dibelakang kita sedang berusaha mengejar atau menghentikan mobil lain. Bila kemudian polisi sudah tepat disamping mobil kita dan memberi isyarat dengan tangan (keadaan berubah/fakta baru) kita bisa berkesimpulan bahwa polisi meminta kita untuk berhenti. Setiap keadaan berubah atau ada fakta baru kita selalu memiliki alasan baru untuk membuat memilih kesimpulan yang baru atau tetap pada kesimpulan sebelumnya.

## Contoh kasus Backward

Untuk keadaan yang sama bila kita menggunakan rangkaian backward kita menjadikan keadaan sebagai hipotesa yang dianggap benar dan berfikir apa sebabnya (mengapa polisi ingin kita berhenti ?) lalu kita mengingat-ingat apa yang terjadi sebelumnya (mencari fakta pendukung), apakah kita salah jalur ?, atau melebihi batas kecepatan ?, atau lampu rem tidak menyala ? atau ada yang salah dengan plat nomer ? .

Jika salah satu atau beberapa pertanyaan dijawab dengan ya berarti kita mendapatkan hipotesa baru lagi yang membutuhkan fakta pendukung baru lagi.

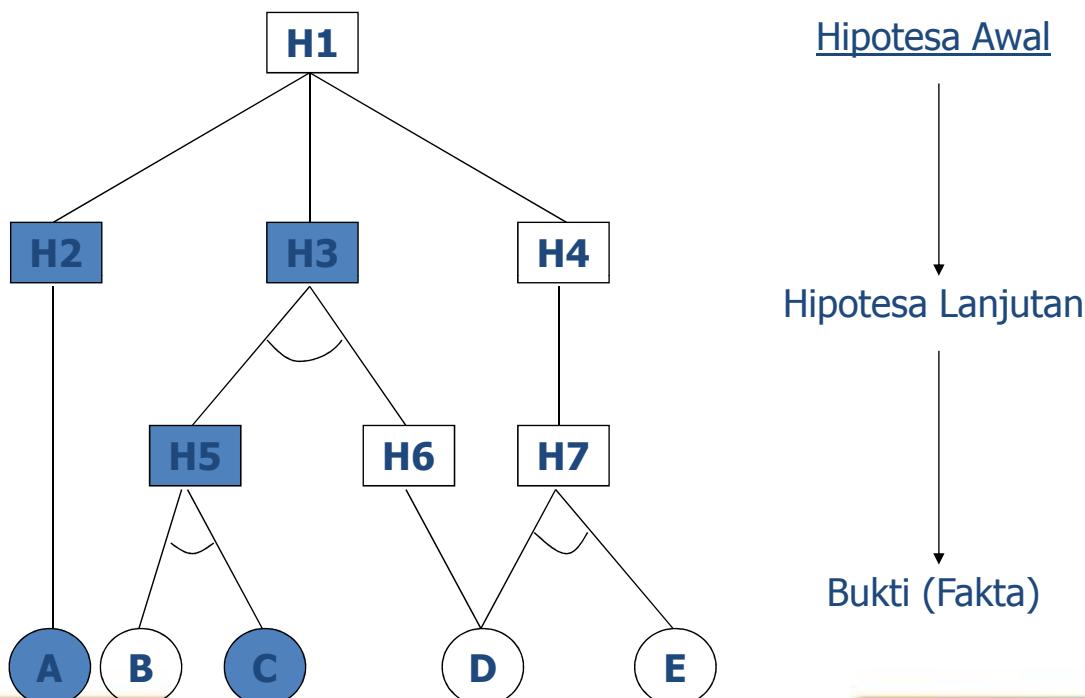
## Rangkaian Forward sebagai Bottom-up Reasoning



## Karakteristik Forward

- Kesimpulan yang benar hanya dapat diturunkan dari fakta-fakta yang benar
- Suatu fakta dapat membangkitkan / mendukung lebih dari satu kesimpulan
- Semakin banyak fakta yang didapat atau mungkin terjadi akan mengakibatkan semakin banyak pula kesimpulan yang dapat dibuat
- Kesimpulan akan menjadi lebih valid jika didukung oleh lebih banyak fakta
- Melakukan pencarian dengan metode breadth-first, yang berusaha mengumpulkan semua fakta yang mungkin pada setiap level

## Rangkaian Backward sebagai Top-Down Reasoning



## Karakteristik Backward

- Bukti adalah hal terpenting dalam backward
- Salah satu cara terpenting mendapatkan bukti adalah dengan mengajukan pertanyaan
- Pertanyaan yang benar (sesuai dengan hipotesa awal) akan mengarahkan kepada bukti yang diinginkan
- Pertanyaan juga berfungsi untuk mempersempit ruang hipotesa
- Melakukan pencarian dengan metode depth-first yang berusaha mendapatkan bukti lanjutan untuk mendukung bukti sebelumnya

## Perbandingan Forward & Backward

Forward	Backward
Perencanaan, Pemantauan, kontrol saat ini kemasa depan	Diagnosa keadaan sekarang ke masa lalu
Antecedent terhadap sebab, Bergerak dengan perubahan data	Akibat terhadap antecedent, Bergerak dgn perubahan tujuan
Bergerak maju untuk menemukan kesimpulan yang mengikuti fakta	Bergerak mundur untuk mendapatkan fakta yang mendukung hipotesa
Pencarian melebar (breadth-first)	Pencarian mendalam (depth-first)
Sebab menentukan pencarian	Akibat menentukan pencarian

# Beberapa Metode Lain Untuk Inferensi

## Analogi

Ide pokok dari pemberian alasan dengan analogi adalah mencoba dan menghubungkan situasi lama (yang pernah terjadi) sebagai penuntun ke situasi yang baru.

Analogi cukup baik diterapkan dalam kehidupan sehari-hari karena begitu banyaknya situasi baru yang kita dapat dalam keseharian kita.

Analogi tidak memiliki formula pembuktian, sehingga analogi dapat disebut sebagai pemberian alasan secara heuristic yang kadang dapat digunakan sebagai hipotesa awal sebelum melakukan rangkaian backward, atau mempersempit ruang pencarian pada rangkaian forward

Jika seorang pasien pergi ke dokter, dokter akan menanyakan gejala apa yang kita rasakan / sakit apa? (bukan apa penyebab kita sakit). Jika gejala yang kita alami sama dengan orang lain yang diketahui menderita sakit "X", maka dokter mungkin menyimpulkan dengan analogi bahwa kita juga menderita sakit "X".

Diagnosa ini mungkin salah karena pada dasarnya tiap orang berbeda (unik) atau gejala yang sama bisa berasal dari penyakit yang berbeda, sehingga dokter tidak menyimpulkan tapi menjadikan diagnosa tadi sebagai hipotesa awal. Dengan demikian sudah menghindari pengambilan kesimpulan yang salah namun sudah mempersempit kemungkinan dan menghemat waktu dan biaya dalam melakukan serangkaian test fisik.

## Generate and Test

Merupakan salah satu metode inferensi dalam AI klasik yang berusaha mendapatkan solusi dengan mengenerate atau membuat yang mirip solusi (diduga) melakukan test kemudian berhenti bila itu ternyata solusinya atau membuat lagi yang lain dan melakukan test lagi sampai ditemukan solusi.

Metode ini pertama kali digunakan dalam sistem pakar DENDRAL (1965) untuk melengkapi penentu struktur molekul organik, dengan membuat semua struktur organik yang potensial dan mengetestnya satu persatu.

Program lain yang menggunakan metode ini adalah AM (Artificial Mathematician) untuk menyimpulkan konsep matematika yang baru.

## Plan-Generate-Test

Variasi dari Generate-and-test dimana digunakan program perencanaan untuk mengurangi kemungkinan sehingga lebih sedikit yang harus dibuat dan ditest.

Contoh sistem pakar yang menggunakan metode ini antara lain adalah MYCIN untuk diagnosa matematika dan juga perencanaan perawatan/pengobatan pasien.

Perencana MYCIN pertama kali membuat daftar prioritas dari obat terapi untuk pasien yang sensitif terhadap obat. Generator kemudian akan mengambil daftar prioritas dari perencana dan membuat subdaftar dari satu atau dua macam obat, sub daftar obat ini kemudian di test terhadap pasien untuk melihat efek kesembuhan, infeksi atau alergi pasien

## Ketidakpastian

Ketidakpastian adalah sebutan untuk menyatakan kekurangan informasi yang memadai untuk mengambil suatu keputusan.

Ketidakpastian menjadi masalah karena menghalangi kita untuk membuat keputusan yang paling baik dan bahkan akan mengakibatkan kita membuat keputusan yang jelek. Dalam dunia kesehatan misalnya, ketidakpastian akan mengakibatkan perawatan yang tidak baik dan terapi yang salah. Dalam dunia bisnis, ketidakpastian dapat berarti hilangnya peluang untuk mendapat keuntungan atau bahkan mengakibatkan kerugian yang besar secara finansial.

Contoh sistem pakar klasik yang sukses dalam pengambilan keputusan dibawah keidakpastian adalah PROSPECTOR yang digunakan untuk membantu eksplorasi bahan tambang (mineral), dan MYCIN dalam diagnosa kesehatan

Dalam PROSPECTOR, kesimpulan datang ketika semua bukti yang diperlukan untuk mengambil kesimpulan tidak diketahui dengan pasti.

Meskipun mungkin untuk mencapai kesimpulan yang lebih baik dan dapat dipercaya dengan melakukan test, tapi akan memakan waktu dan biaya dalam melakukan test.

Dalam perawatan kesehatan juga akan memakan waktu dan biaya untuk melakukan test, dan juga selama test berlangsung, keadaan pasien bisa bertambah parah atau bahkan meninggal. Jadi lebih efektif kita ambil kesimpulan dengan keyakinan 95% daripada melakukan test dulu untuk mendapatkan keyakinan 98%.

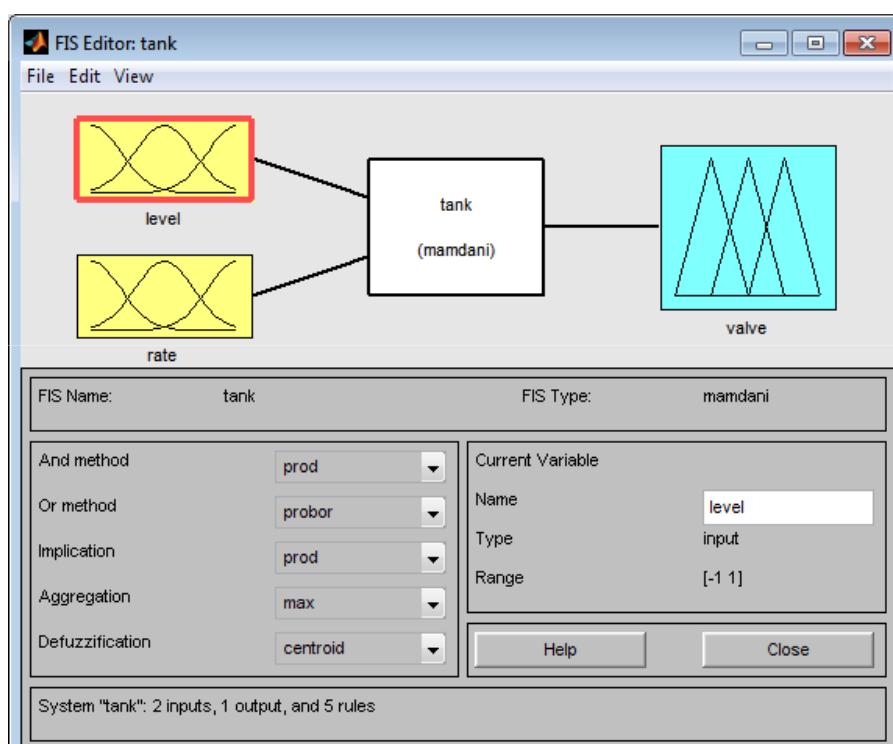
## Pengertian Soft Computing

- komputasi yang melibatkan data-data dengan ketidakpastian, ketidakakuratan maupun kebenaran yang parsial
- Teknik-teknik yang digunakan antara lain:
  - Fuzzy Inference Systems
  - ANFIS
  - Jaringan Syaraf Tiruan
  - Algoritma Genetik

# Fuzzy Inference System (FIS)

- Sistem inferensi yang mendasarkan logikanya dengan logika yang menyerupai bahasa sehari-hari.
- Dalam keseharian, terkadang kita menyukai informasi misalnya suhu 25 derajat selsius dinyatakan dengan suhu “Sedang”.
- Komponen FIS adalah Fungsi keanggotaan, Rule dan Defuzzifikasi.

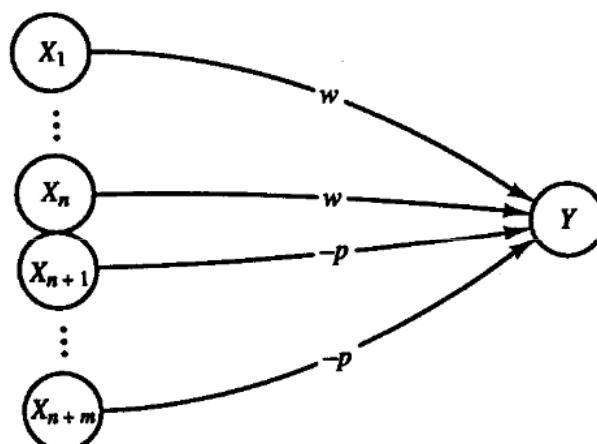
## Contoh FUZZY



# Jaringan Syaraf Tiruan (JST)

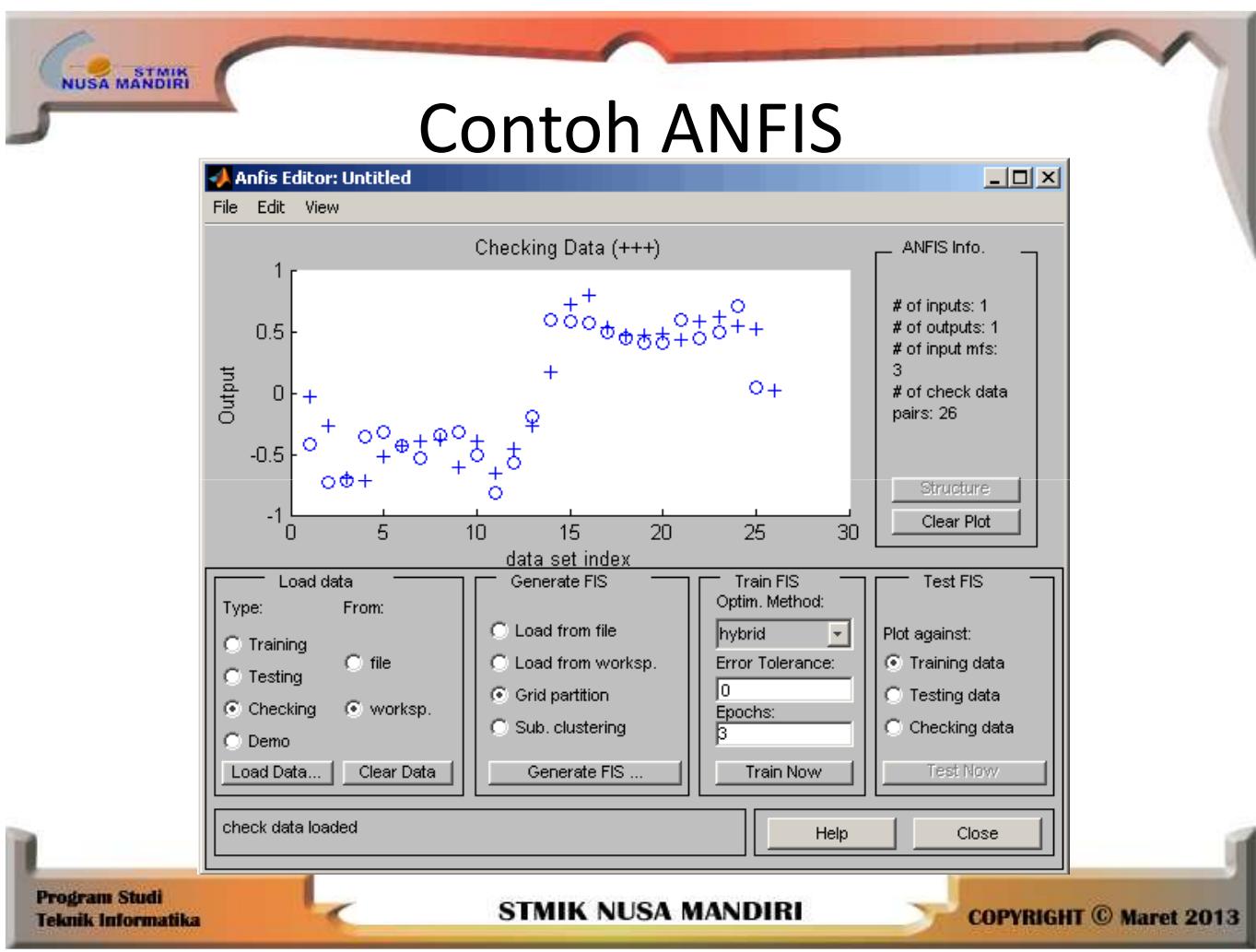
- Berusaha meniru mekanisme syaraf biologis (*Artificial Neural Network*).
- Terdiri dari susunan: Neuron dengan fungsi aktivasi antara lain: *linear* dan *sigmoid*.
- Pembelajaran diperlukan untuk merubah bobot (yg sesuai diperkuat yang tidak sesuai diperlemah).
- Terdiri dari satu lapis masukan, satu lapis keluaran dan beberapa lapis tersembunyi.

## Model JST Pertama



# **Adaptive Neuro Fuzzy Inference System (ANFIS)**

- Gabungan antara FIS dengan Jaringan Syaraf Tiruan (JST).
- Karena rumitnya membuat rule-based, dengan JST rule dibuat oleh sistem itu sendiri dengan mekanisme “*Learning*”.



# Genetic Algorithm

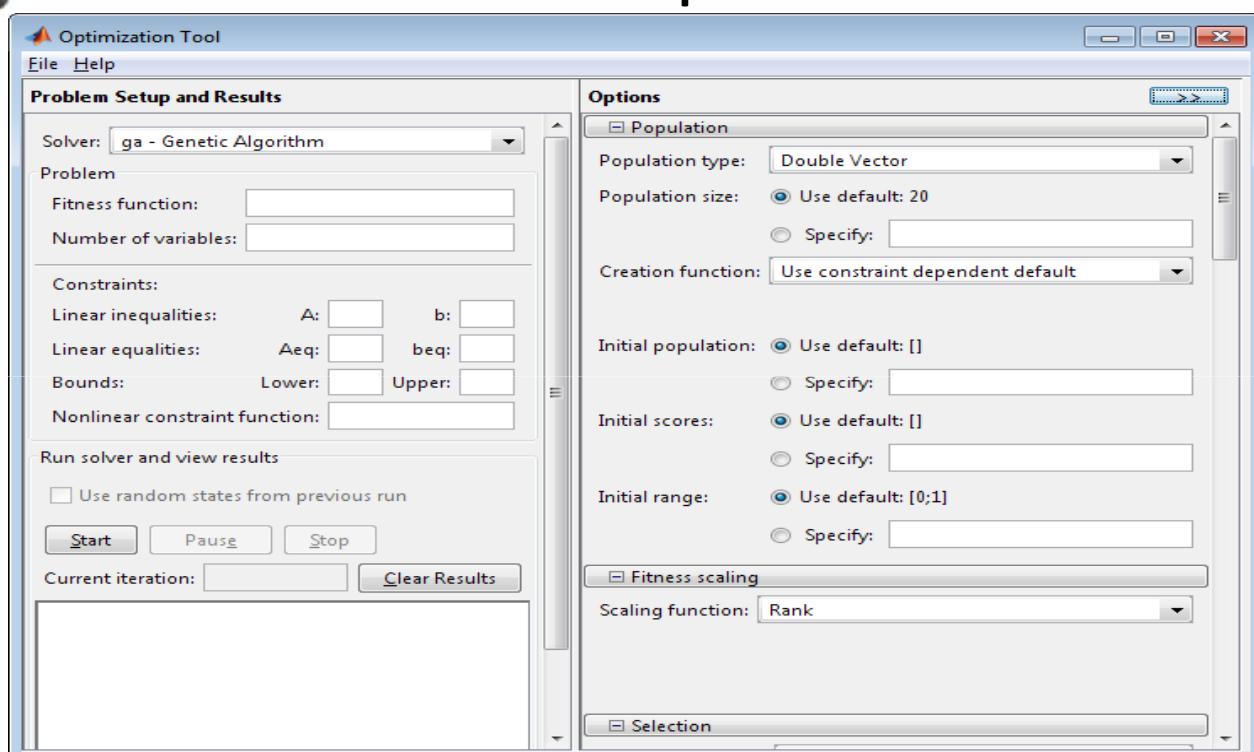
- Diperkenalkan pertama kali oleh Holland tahun 60-an.
- Berusaha meniru mekanisme evolusi makhluk hidup dalam menyelesaikan kasus tertentu.
- Kasus yang diselesaikan biasanya dalam optimasi dimana nilai minimum/maksimumnya banyak sehingga jika diselesaikan dengan aljabar biasa, sistem akan terjebak dalam local minimum/maksimum.

## Struktur Umum

- Populasi, istilah pada teknik pencarian yang dilakukan sekaligus atas sejumlah solusi yang mungkin
- Kromosom, individu yang terdapat dalam satu populasi dan merupakan suatu solusi yang masih berbentuk simbol.
- Generasi, populasi awal dibangun secara acak sedangkan populasi selanjutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi

- Fungsi Fitness, alat ukur yang digunakan untuk proses evaluasi kromosom. Nilai fitness dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut.
- Generasi berikutnya dikenal dengan anak (*offspring*) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilang (*crossover*).
- Mutasi, operator untuk memodi-fikasi kromosom.

## Toolbox GA pada MATLAB



# PENGENALAN CLIPS

Pertemuan ke empat belas

CLIPS adalah:

- ❖ Salah satu bahasa pemrograman untuk sistem pakar
- ❖ Dibuat dari Bahasa C
- ❖ Tipe inference yang dipakai rangkaian forward
- ❖ Dikembangkan oleh NASA

## Eleman Pokok CLIPS:

- Daftar Fakta (Fact List)
- Basis Pengetahuan (Knowledge Base)
- Mesin Inference (Inference Engine)

## FAKTA:

- Fakta dibuat dari field yang berupa kata, string, atau angka.
- Field pertama dari fakta secara normal digunakan untuk menunjukkan tipe informasi yang disimpan dalam fakta dan disebut dengan relasi. Template fakta dapat digunakan untuk dokumen tipe informasi yang disimpan dalam fakta.

## BARIS:

- Baris merupakan komponen kedua dari sistem CLIPS.
- Suatu baris dibagi menjadi LHS dan RHS.
- LHS dari baris dapat dianggap sebagai porsi IF dan RHS dapat dianggap sebagai porsi THEN.
- Baris dapat mempunyai pola multiple dan aksi.

## MESIN INFERENSI:

- Baris yang mempunyai pola yang dipenuhi dengan fakta akan membuat aktivasi yang ditampilkan pada agenda.
- Refraction akan mencegah baris dari pengaktifan secara konstant oleh fakta yang lama.

## NOTASI

( )	:	Diisi Fakta
[ ]	:	Pilihan
<>	:	Harus diisi angka
{ }	:	Menyatakan Himpunan Fakta
" "	:	Berarti String

## Mengisi Daftar Fakta

Sintaks:

(assert <<<fact>>>)

Contoh:

CLIPS> (assert (emergency fire)) enter

CLIPS>

*Note: Fakta selain diketik langsung, bisa juga diimport dari text editor lainnya.*

## Melihat Isi List Fakta

```
CLIPS> (facts) enter
f-1      (emergency fire)
CLIPS>
```

## Perintah-perintah Dasar CLIPS

- WATCH: digunakan untuk debugging program
- MATCHES: debugging pola baris.
- SET BREAK: Mengeset batas debugging

- Pernyataan dan penarikan kembali fakta, baris string dan aktivasi dapat dilakukan dengan menggunakan perintah WATCH.
- Perintah MATCHES akan menampilkan fakta yang telah menggabungkan pola baris termasuk daftar gabungan bagian untuk suatu baris.
- Perintah SET BREAK memungkinkan pembuatan/pelaksanaan untuk diakhiri sebelum baris di-fired.
- Perintah PRINTOUT dapat digunakan untuk output informasi dan RHS baris. Perintah CLEAR digunakan untuk memberi initial kembali suatu pernyataan dari lingkungan CLIPS.

# Selamat Datang

*Semoga Tuhan memberi berkah pada kelas ini.*

# Perencanaan pembelajaran

- **TUJUAN**

Setelah Mengikuti kuliah ini, mahasiswa diharapkan dapat merancang dan membangun sistem interaksi secara metodologis, berdasarkan prinsip dan arahan yang ada

Persyaratan untuk mengikuti matakulian sistem interaksi :

Telah mengikuti kuliah Teknik Perangkat Lunak (Software Engineering)

Implementas dari matakuliah sistem interaksi ini adalah :

1. Desain Web
2. Animasi Interaktif

## Materi pokok

Pertemuan ke-	Pokok Bahasan	Keterangan
1	Pengenalan Sistem Interaksi	
2	Konsep dan pengertian dari interaksi	
3	Understanding Users	
4	Usability from Interaction System	
5	Guidelines, Principles and Theories	
6	Interaction styles and Interaction device (1)	
7	Quis	
8	<b>Ujian Tengah Semester</b>	
9	Interaction styles and Interaction device (2)	
10	Metodologi perancangan sistem interaksi dan evaluasinya (1)	
11	Metodologi perancangan sistem interaksi dan evaluasinya (2)	
12	Design Prototyping and construction	
13	Teknik dan Perangkat bantu pengembangan sistem interaksi	
14	Evaluasi perancangan sistem interaksi	
15	<b>Ujian Akhir Semester</b>	

## Ringkasan Kontrak kuliah

Pada perkuliahan ini mengajarkan tentang teknik merancang sistem interaksi, dimana proses perancangan sistem interaksi melibatkan banyak faktor mulai dari lingkungan sistem yang akan diterapkan, karakteristik pemakai sampai dengan arsitektur dari sistem interaksi tersebut

Materi dari perkuliahan ini membahas seluruh faktor tersebut sampai dengan teknik dan metodologi pengembangannya

# Sumber Referensi

1. Ben Shneiderman, Designing the User Interface : Strategies for Effective Human – Computer Interaction 4<sup>th</sup> edition, Addison –Wesley 2005
2. John Wiley & Sons Interction Design beyond human computer interaction
3. Andrew Sears, Julie A Jacko, the Human Computer Interaction Handbook Fundamentals – evolving technologies and Emerging Application, 2<sup>nd</sup>, 2005

## RENCANA PEMBELAJARAN

- Pertemuan 1 s.d 6 disampaikan dengan Metode Ceramah, Metode Diskusi dan Latihan Soal.
- Pada Pertemuan 9 s.d 14 disampaikan dengan Metode Ceramah Metode diskusi dan latihan soal.
- Dosen pengampu matakuliah ini harus dapat mengarahkan mahasiswa agar dapat membuat desain interaksi baik dengan aplikasi animasi, aplikasi berbasis web dan desktop untuk persiapan penulisan skripsi

# Penilaian Nilai Tugas

- Tugas 1 : Bobot Nilai 25
  - Tugas 2 : Bobot Nilai 25
  - Tugas 3 : Bobot Nilai 25
  - Quiz UTS Essay : Bobot Nilai 25
- Jadi Total Nilai TUGAS : 100

# PERTEMUAN

1

## Beberapa pengertian Sistem Interaksi

***Interaction Design*** didefinisikan sebagai

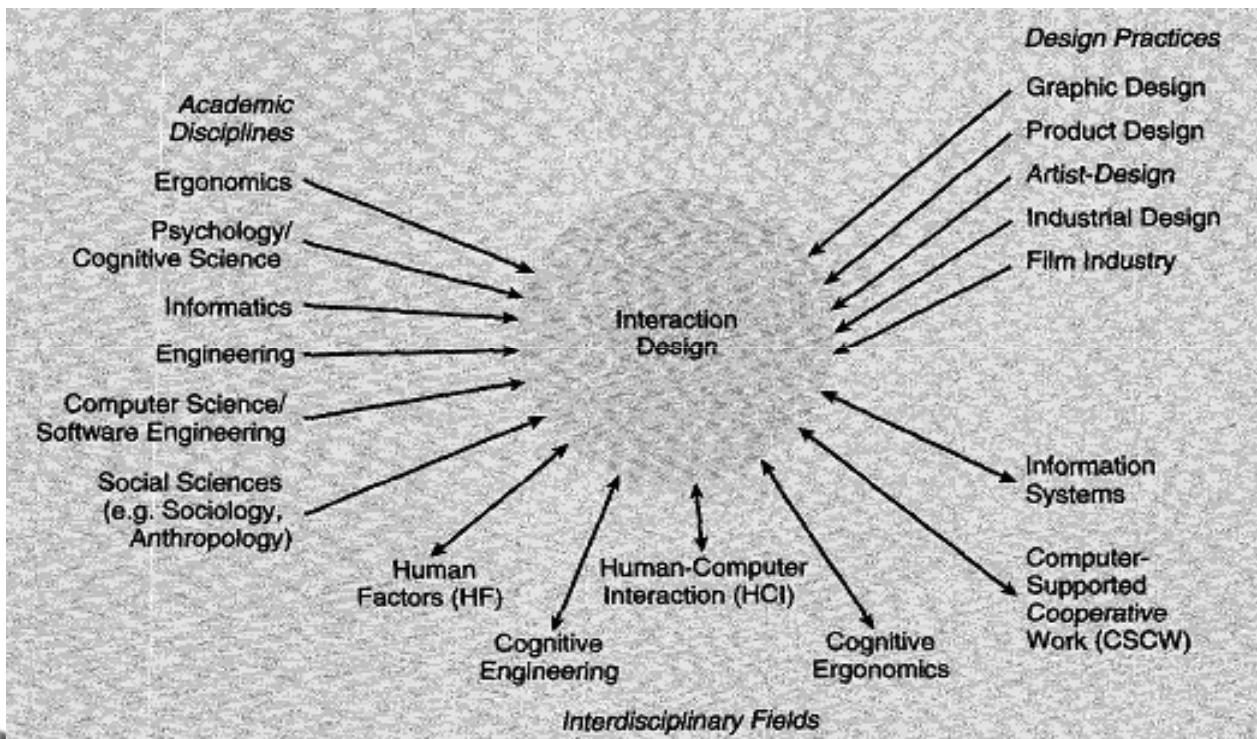
- Perancangan interaktif produk untuk mendukung cara orang berkomunikasi dan saling berhubungan satu dengan yang lain setiap hari
- Membuat pemakai meningkatkan pekerjaan dengan cara, berkomunikasi, dan interaksi
- Cara bagaimana dari kita sehari-hari berinteraksi dengan menggunakan komputer

- Merancang interaksi berarti membangun suatu produk yang dapat digunakan.
- Dapat digunakan artinya mudah untuk dipelajari, efektif digunakan, dan terasa nyaman bagi penggunanya

Untuk itu merancang interaksi. Maka hal yang perlu diperhatikan adalah :

1. Masukan, ide dan umpan balik dari pengguna berdasarkan yang mereka rasakan dan alami
2. Memperhatikan dan mempertimbangkan siapa yang akan menggunakan interaksi itu
3. Mengetahui cara mereka menggunakannya
4. Mengetahui aktifitas-aktifitas penggunanya

- UI designers engineers : orang yang membangun dan memodelkan cara penggunaan, metode analisis workflow dan prototype
- Information Architects : orang yang memberikan ide bagaimana merencanakan dan menyusun produk yang interaktif, khususnya website
- User Experience (UX) designer/architects / researchers : orang yang melakukan semua hal di atas tetapi juga melakukan studi lapangan untuk meneliti kebutuhan pengguna dan mengubahnya menjadi hasil yang nyata



## Proses – prses yang terlibat dalam perancangan interaksi :

Essentially, the process of interaction design involves four basic activities:

1. Identifying needs and establishing requirements.
2. Developing alternative designs that meet those requirements.
3. Building interactive versions of the designs so that they can be communicated and assessed.
4. Evaluating what is being built throughout the process.

## Characteristics of the interaction design process:

1. Users should be involved through the development of the project.
2. Specific usability and user experience goals should be identified, clearly documented, and agreed upon at the beginning of the project.
3. Iteration through the four activities is inevitable.

## Tujuan dari perancangan Interaksi

### A. Tujuan kegunaan

Berkaitan dengan produk interaktif yang mudah dipelajari, efektif, dan nyaman digunakan dari sudut pandang pengguna.

Hal-hal yang dipenuhi untuk tujuan kegunaan ini adalah:

1. Efektif : seberapa baik produk itu dapat bekerja
2. Efisien: cara produk mendukung pengguna melakukan tugasnya
3. Aman digunakan: melindungi pengguna dari situasi yang bahayakan
4. Memiliki utilitas yang baik: seberapa fungsi-fungsi tersedia bagi pengguna.
5. Mudah dipelajari : seberapa mudah produk digunakan
6. Mudah untuk diingat penggunaanya: seberapa mudah penggunaan diingat

## B. Tujuan Pengalaman Pengguna

Tujuan ini lebih berkaitan dengan rasa dan pengalaman yang dirasakan dan dialami oleh pengguna dengan produk yang digunakannya.

## C. Prinsip Merancang

- a. Visibility : fungsi jelas, bagian-bagian jelas, simbol jelas,
- b. Feedback : umpan balik yang diberikan sebagai hasil aksi yang diberikan
- c. Constraints : memberi batasan yang jelas untuk pengguna mengerti apa yang dikerjakan

- d. Consistency: penggunaan operasi, dan elemen pada produk untuk melakukan hal yang sama. Misal warna merah selalu digunakan untuk memberi pesan kesalahan, warna biru untuk konfirmasi
- e. Affordances: atribut dari suatu objek yang membuat orang tahu bagaimana menggunakannya. Misal button untuk ditekan, check box untuk dicentang.

Prinsip merancang lain adalah simplicity atau kesederhanaan yang biasanya diberlakukan pada website.

## Pertemuan 2

## Konsep dan pengertian dari interaksi

- Bayangkan anda diminta untuk mendesain satu aplikasi agar memungkinkan orang mengorganisir, penyimpanan, dan mengambil kembali surat elektronik (email) mereka dgn cepat, efisien dan menyenangkan.
- Apa yang akan anda lakukan?
- Bagaimana anda memulainya?

## Pengertian Ruang masalah

Pada proses pembuatan produk interaktif, hal ini dapat diibaratkan kita memilih dan memilah mana kacang mana baut, hal ini dimaksudkan bahwa pada level perancangan kita bekerja dengan memikirkan bagaimana merancang secara fisik dan gaya interaksi apa yang dapat digunakan oleh karena itu ruang masalah adalah bagaimana kita memahami perancangan interaksi secara fisik dan model interaksi yang dapat digunakan oleh user

## Model – model konsep

"The most important thing to design is the user's conceptual model. Everything else should be subordinated to making that model clear, obvious, and substantial. That is almost exactly the opposite of how most software is designed." (David Liddle, 1996, p. 17)

Understanding and conceptualizing interaction by a conceptual model is meant:

a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended.

## Model Konseptual Berbasis Pada Aktivitas-aktivitas

Jenis paling umum dari aktivitas-aktivitas dimana para pemakai akan berinteraksi dengan sistem:

1. Model instruksi/instructing
2. Model berbicara/conversing
3. Model memanipulasi dan melayari
4. Model menyelidiki dan menjelajah

## 1. Model Instruksi (instructing)

Model konseptual ini menggambarkan bagaimana para pemakai melaksanakan tugas mereka melalui intruksi (instructing) apa yang harus dikerjakan sistem

## 2. Model Berbicara (conversing)

Model konseptual yang ini menjadi dasar terhadap gagasan untuk seseorang berbicara dengan satu sistem, dimana sistem berlaku sebagai satu mitra temu-wicara

## 3. Memanipulasi dan melayari (manipulating and navigating)

Model konseptual yang ini menggambarkan aktivitas dari memanipulasi objek dan navigating melalui jarak virtual dengan cara memanfaatkan para pemakai' pengetahuan dari bagaimana mereka melakukan ini didunia fisik

## 4. Exploring and browsing

Model konseptual ini menjadi dasar terhadap gagasan untuk membiarkan orang untuk menyelidiki dan menjelajah informasi, memanfaatkan pengetahuan mereka dan bagaimana mereka melakukan ini dengan media yang sudah ada (contoh buku, majalah, TV, radio, perpustakaan, pamflet, brosur).

## Model Konseptual Berbasis Pada Objek

Kategori kedua model konseptual menjadi dasar terhadap satu objek , seperti

- Satu perangkat (tool), satu buku, atau satu wahana. Ini cenderung untuk yang lebih spesifik dibandingkan model konseptual
- Berbasis pada aktivitas-aktivitas, memusatkan pada cara satu objek tertentu adalah digunakan dalam satu tertentu konteks. Mereka sering berbasis pada satu analogi dengan sesuatu di dunia fisik.
- Satu contoh satu model konseptual yang benar-benar sukses berbasis pada satu objek adalah spreadsheet (Winograd, 1996).

## Metaphore Interface

- Jalan lain untuk menggambarkan model konseptual adalah dengan methaphore interface.
- Dalam hal ini adalah satu model konseptual berkembang untuk menjadi serupa dibeberapa cara pada aspek satu entitas fisik (atau entitas) tetapi itu juga mempunyai perilaku dan properti sendiri
- Beberapa model mungkin menjadi berbasis pada satu aktivitas atau satu objek atau keduanya

- Manfaat *Methapore Interface* telah terbukti benar-benar sukses, menyediakan para pemakai dengan suatu mengorientasikan alat yang dikenal dan membantu mereka memahami serta belajar bagaimana untuk menggunakan satu sistem.

## Paradigma Interaksi

- Di satu level yang lebih umum, sumber inspirasi lain untuk memberi tahu perancangan satu model konseptual adalah paradigma interaksi

Sejumlah paradigma interaksi alternatif diusulkan oleh peneliti ditujukan untuk pemandu masa depan desain interaksi dan pengembangan sistem

- Ubiquitous computing (teknologi menyatu di dalam lingkungan)
- Perpasive computing (integrasi sempurna dari teknologi)
- Attentive environment (komputer mengindahkan kebutuhan pemakai)
- Workaday World (aspek sosial dari menggunakan teknologi )

## Tugas

Gambarkan model konseptual yang mendasari perancangan:

- a. Satu kalender pocket-sized personal buku harian (satu minggu pada satu halaman)
- b. satu kalender dinding (satu bulan pada satu halaman, biasanya dengan satu foto gambar)
- c. Apakah jenis utama aktivitas dan objek mereka ?

# Pertemuan 3

## Memahami Pengguna

- Bayangkan usaha untuk mengemudi suatu mobil dengan menggunakan sekedar 'keyboard' komputer.
- Empat panah kunci adalah digunakan untuk mengemudikan, bar ruang/spasi untuk penggereman, dan kunci kembalinya untuk mempercepat.
- Untuk menunjukan meninggalkan anda perlu untuk menekan tombol F1 dan untuk menunjukan membetulkan tombol F2. Ke bunyi klakson anda perlu untuk menekan tombol F3. Untuk men-switch lampu besar anda perlu menggunakan kunci F4 dan, untuk men-switch wiper kunci F5.

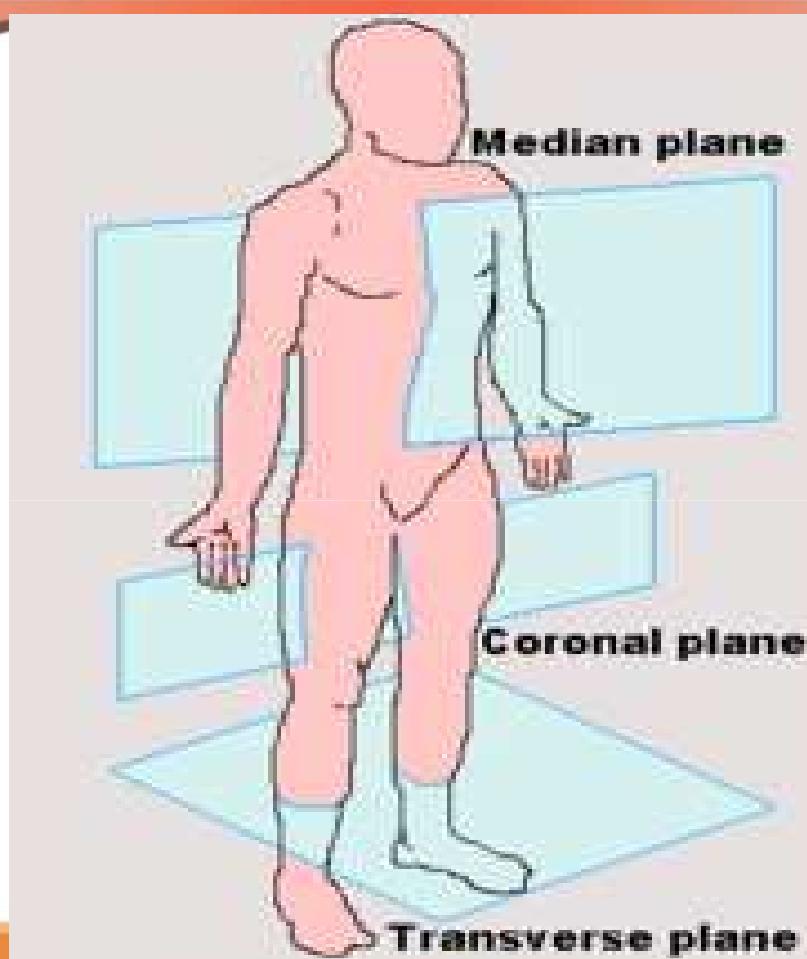
Dalam istilah kedokteran umum ,

1. Fisiologi adalah Ilmu yang mempelajari faal atau pekerjaan dari tiap – tiap jaringan tubuh atau bagian dari alat – alat tubuh dan sebaginya
2. Fisiologi mempelajari fungsi atau kerja tubuh manusia dalam keadaan normal

## ISTILAH LOKASI ANATOMI

Bidang anatomi adalah bidang yang melalui tubuh dalam posisi anatomi; Bidang ini dibagi menjadi :

- **Bidang median:** bidang yang membagi tepat tubuh menjadi bagian kanan dan kiri.
- **Bidang sagital:** bidang yang membagi tubuh menjadi dua bagian dari titik tertentu (tidak membagi tepat dua bagian). Bidang ini sejajar dengan bidang median.
- **Bidang horizontal:** bidang yang terletak melintang melalui tubuh (bidang X-Y). Bidang ini membagi tubuh menjadi bagian atas (superior) dan bawah (inferior).
- **Bidang koronal:** bidang vertikal yang melalui tubuh, letaknya tegak lurus terhadap bidang median atau sagital. membagi tubuh menjadi bagian depan (frontal) dan belakang (dorsal).



- 
- ### ARAH DAN BIDANG ANATOMI
- **Superior (=atas)** atau **kranial**: lebih dekat pada kepala.  
Contoh: Mulut terletak superior terhadap dagu.
  - **Inferior (=bawah)** atau **kaudal**: lebih dekat pada kaki.  
Contoh: Pusar terletak inferior terhadap payudara.
  - **Anterior (=depan)**: lebih dekat ke depan.  
Contoh: Lambung terletak anterior terhadap limpa.
  - **Posterior (=belakang)**: lebih dekat ke belakang.  
Contoh: Jantung terletak posterior terhadap tulang rusuk.
  - **Superfisial**: lebih dekat ke/di permukaan.  
Contoh: Otot kaki terletak superfisial dari tulangnya.
  - **Profunda**: lebih jauh dari permukaan.  
Contoh: Tulang hasta dan pengumpil terletak lebih profunda dari otot lengan bawah.
  - **Medial (=dalam)**: lebih dekat ke bidang median.  
Contoh: pangkal lengan terletak medial terhadap tubuh.
  - **Lateral (=luar)**: menjauhi bidang median.  
Contoh: Telinga terletak lateral terhadap mata.
  - **Proksimal (=dekat)**: lebih dekat dengan batang tubuh atau pangkal.  
Contoh: Siku terletak proksimal terhadap telapak tangan.
  - **Distal (=jauh)**: lebih jauh dari batang tubuh atau pangkal.  
Contoh: Pergelangan tangan terletak distal terhadap pangkal lengan.

Dalam ruang psikologi bahwa fisiologis manusia dibagi menjadi tiga hal utama :

1. Kognitif
2. Afektif
3. Psikomotorik

## Kawasan Kognitif

Kawasan ini merujuk pada potensi subyek yang meliputi kecerdasan atau intelektualitas, seperti pengetahuan yang dimiliki ataupun cara berfikir.

Menurut Bloom kawasan kognitif dibagi menjadi dua bagian yakni

1. Bagian pengetahuan yang meliputi : kemampuan terhadap pengertian atau definisi
2. Keterampilan intelektual yang meliputi pemahaman, aplikasi, analisa, sintesa dan evaluasi

Semakin meningkat kemampuan seseorang memperlihatkan akan memperlihatkan kecerdasannya yang meningkat

## Kawasan Afektif

Kawan ini mencakup kemampuan perasaan dan emosi , yang meliputi aspek – aspek :

1. Penerimaan terhadap lingkungan
2. Tanggapan atau respon terhadap lingkungan
3. Penghargaan dalam bentuk ekspresi

## Kawasan Psikomotorik

Kawasan ini mencakup kemampuan dan keterampilan fisik dalam mengerjakan atau menyelesaikan sesuatu

## Apa pengertian kognisi?

- Kognisi adalah apa yang terpikir oleh manusia ketika mereka ingin melaksanakan aktivitas-aktivitas setiap hari.

Kognisi juga menggambarkan secara spesifik berbagai jenis proses. meliputi:

- Perhatian
- Persepsi dan pengenalan
- Memori
- Belajar
- Membaca, pernyataan, dan mendengarkan
- Pemecahan masalah, perencanaan, penalaran, pengambilan-keputusan

Pengetahuan penerapan dari dunia fisik ke dunia digital. Demikian pula pemahaman berbagai proses kognitif dimana para pemakai terlibat ketika saling berinteraksi dengan sistem, Hal ini berguna untuk memahami cara orang mengatasi permintaan dari kehidupan sehari-hari

## Kerangka Konseptual untuk kognisi

Pendekatan lain harus menerapkan teori dan kerangka konseptual pada desain interaksi. Dalam hal ini bagian yang kita menguji tiga pendekatan ini, yang masing-masing mempunyai satu perspektif berbeda terhadap kognisi:

- Model mental
- Pemrosesan informasi
- Kognisi eksternal

### Model Mental

- Setelah mengembangkan satu model mental dari satu produk interaktif, orang akan menggunakan untuk membuat kesimpulan tentang bagaimana untuk melaksanakan tugas ketika menggunakan produk interaktif.
- Model Mental adalah juga digunakan untuk mengukur apa yang harus dikerjakan ketika sesuatu tak diduga terjadi dengan satu sistem dan ketika menemui sistem tidak familier. Terlebih lagi banyak orang belajar tentang satu sistem dan bagaimana sistem berfungsi, lebih mengembangkan model mental mereka .

Sebagai contoh, para insinyur TV mempunyai satu mental model tentang bagaimana cara TV bekerja yang memungkinkan mereka untuk mengembangkan dan membuat rencana bagaimana untuk melakukan perbaikan.

## Pemrosesan Informasi

Pendekatan lain menentukan konsep bagaimana pekerjaan pikiran untuk digunakan analogi dan kiasan

Model pemrosesan informasi menyediakan satu basis dimana untuk membuat ramalan tentang kinerja manusia. Hipotesis mungkin membuat seseorang akan lari dan memberikan reaksi terhadap satu stimulus (juga dikenal sebagai waktu reaksi) dan bottleneck apa yang akan terjadi jika seseorang dimuat berlebihan dengan terlalu banyak informasi.

## Kognisi Eksternal

Orang saling berhubungan dengan menciptakan informasi melalui menggunakan berbagai representasi eksternal, sebagai contoh buku, multimedia, surat kabar, halaman web, peta, diagram, catatan, gambar, saat perangkat (tools) berkembang di dalam kehidupan untuk membantu kognisi, seperti pena, kalkulator, dan teknologi berbasis-komputer.

Kognisi eksternal akan berkaitan dengan kognisi proses ketika kita saling berhubungan dengan representasi eksternal yang berbeda (Scaife dan Rogers, 1996)

## Informing Design: From Theory To Practice

Teori-teori, model, dan kerangka konseptual menyediakan abstrak untuk berpikir tentang fenomena. Khususnya, mereka memungkinkan penyamarataan untuk membuat tentang kognisi melintasi situasi berbeda.

Teori dalam bentuk murni, terkadang bisa sulit dicerna. Terminologi misterius dan jargon yang digunakan bisa cukup diletakkan untuk mereka yang tidak akrab dengannya. Hal ini juga memerlukan banyak waktu untuk membiasakan diri dengan itu.

Peneliti telah mencoba untuk membantu dengan membuat teori yang lebih mudah diakses dan praktis. Ini sudah termasuk menerjemahkannya ke dalam:

- \* prinsip-prinsip dan konsep desain
- \* aturan desain
- \* metode analitik
- \* desain dan metode evaluasi

## Tugas

Tujuan dari tugas ini adalah untuk memperoleh model mental dari orang-orang. Secara khusus, tujuannya adalah untuk memahami pengetahuan tentang produk interaktif tentang bagaimana menggunakannya dan bagaimana cara kerjanya. Pertama, menjabarkan model mental Anda sendiri. Tuliskan bagaimana Anda berpikir mesin kas (ATM) bekerja. Lalu menjawab pertanyaan-pertanyaan berikut Berapa banyak uang Anda yang diperbolehkan untuk mengambil?

Jika Anda telah mengambil uang dan kemudian pergi ke mesin ATM lain dan mencoba menarik jumlah yang sama, apa yang akan terjadi?

Apa yang terjadi dengan kartu Anda?

Bagaimana informasi tersebut digunakan?

Apa yang terjadi jika Anda memasukkan nomor yang salah?

Mengapa ada jeda antara langkah-langkah transaksi?

- \* Berapa lama jedanya? Apa yang terjadi jika Anda mengetik di depan selama jeda?
- \* Apa yang terjadi dengan kartu dalam mesin?
- \* Mengapa tertinggal di dalam mesin?
- \* Apakah Anda menghitung uang tersebut? Mengapa?

## Pertemuan 4

# Usability of Interactive Systems

## Disiplin ilmu yang terlibat dalam sistem interaksi

Disiplin Ilmu yang diperlukan adalah :

1. **Teknik Informatika** :Bidang ini merupakan bidang dasar dalam pembahasan interaksi manusia dengan komputer karena komputer sendiri merupakan subjek dalam interaksi tersebut.
2. **Psikologi** :Bidang ini memperhatikan sifat, kebiasaan, persepsi, pengolahan kognitif, dan ketrampilan motorik pengguna.
3. **Desain Grafis** :Sebuah gambar atau ikon dapat secara cepat memberikan informasi atau makna bagi user tanpa harus banyak berkata-kata. Bidang ini memperhatikan bagaimana representasi grafis (gambar) yang mewakili berbagai macam ragam kalimat.

4. **Ergonomic** :Bidang ini berkaitan dengan aspek fisik untuk mendapatkan lingkungan kerja yang nyaman.
5. **Antropologi** :ilmu pengetahuan yang mempelajari tentang manusia, dan memberi suatu pandangan tentang cara kerja berkelompok yang masing – masing anggotanya dapat memberikan kontribusi sesuai dengan bidangnya.

6. simbol dan suara diinterpretasikan sebagai **Lingistik** : Bahasa sebagai sarana komunikasi mendasar diperlukan dalam pembuatan desain interaksi antara manusia dengan komputer. Melalui bahasa, berbagai macam perintah yang akan dikerjakan oleh komputer.

## Usability

Dalam mendesain antarmuka komputer, perlu diperhatikan faktor tingkat kebergunaan/ usability.

Menurut Nielsen ada lima hal yang menentukan usability yaitu:

1. **Learnability**: User dapat segera memulai pekerjaanya semenjak dimulainya penggunaan sistem.
2. **Efficiency**: User dapat meningkatkan produktifitasnya setelah pertama kali belajar.
3. **Memorability**: User dapat menggunakan sistem kembali dengan cepat setelah lama tidak menggunakan aplikasi tersebut tanpa perlu belajar dari awal kembali

**4. Errors:** User harus mampu diarahkan untuk sekecil mungkin berbuat kesalahan. Apabila user melakukan kesalahan harus ada langkah penanganan yang dapat memulihkan kesalahan tersebut dengan segera.

**5. Satisfaction:** User harus merasa nyaman dengan sistem aplikasi yang digunakannya

## Usability requirements

- Kata lain untuk “user-friendly” di Microsoft Word 2002 yang mudah untuk digunakan, dapat diakses, mudah untuk dimengerti, memiliki keuntungan dan siap untuk digunakan
- Tapi “user-friendly” juga mencari bantuan dan diberi nilai. “user-friendly” tidak hanya dapat dipahami tetapi penuh pengertian, user-friendly dapat dipercaya dan tidak menyusahkan, user-friendly adalah kepercayaan untuk bersama
- Pengukuran ini yang masih bersifat subjektif dan samar oleh karena itu proses sinyal diperlukan untuk merancang sistem yang mudah untuk digunakan untuk pengguna spesifik di konteks yang spesifik

- Sebagai contoh dari kegunaan kebutuhan adalah U.S. Standar militer untuk Ukuran-ukuran Disain Rekayasa Manusia (1999) nyatakan penggunaan adalah :
  1. Pencapaian kinerja yang diperlukan sebagai operator, kontrol, dan personal pemeliharaan
  2. Perkecil keterampilan dan kebutuhan personalia dan waktu pelatihan
  3. Capai keandalan diperlukan dari alat-alat perlengkapan personalia / kombinasi perangkat lunak membantu perkembangan standardisasi desain diantara sistem

Harus meningkatkan kualitas hidup pengguna dan juga kualitas komunitas secara obyektif

Usability memerlukan manajemen proyek dan perhatian saksama pada fase analisa keperluan dan pengujian untuk mencapai tujuan yang obyektif

# Goals for requirements analysis

- Ascertain the user's needs
  - a. Tentukan tugas dan subtugas yang harus dibutuhkan
  - b. Tugas-tugas yang adakalanya yang hanya dilaksanakan, secara umum sangat mudah untuk dikenali
  - c. Fungsi – fungsi harus sesuai dengan kebutuhan pengguna

- **Ensure reliability** /pastikan keandalan
  - Actions must function as specified
  - Database data displayed must reflect the actual database
  - Appease the user's sense of mistrust
  - The system should be available as often as possible
  - The system must not introduce errors
  - Ensure the user's privacy and data security by protecting against unwarranted access, destruction of data, and malicious tampering

- **Promote standardization, integration, consistency, and portability**
  - *Standardization*: use pre-existing industry standards where they exist to aid learning and avoid errors (e.g. the W3C and ISO standards)
  - *Integration*: the product should be able to run across different software tools and packages (e.g. Unix)
  - *Consistency*:
    - compatibility across different product versions
    - compatibility with related paper and other non-computer based systems
    - use common action sequences, terms, units, colors, etc. within the program
  - *Portability*: allow for the user to convert data across multiple software and hardware environments

- *Complete projects on time and within budget* ( Lengkapi proyek – proyek pada waktunya dan dengan pembiayaan)  
*Late or over budget products can create serious pressure within a company and potentially mean dissatisfied customers and loss of business to competitors*

# Usability measures

- Define the target user community and class of tasks associated with the interface
- Communities evolve and change (e.g. the interface to information services for the U.S. Library of Congress)
- 5 human factors central to community evaluation:
  - *Time to learn*  
How long does it take for typical members of the community to learn relevant task?
  - *Speed of performance*  
How long does it take to perform relevant benchmarks?
  - *Rate of errors by users*  
How many and what kinds of errors are made during benchmark tasks?
  - *Retention over time*  
Frequency of use and ease of learning help make for better user retention
  - *Subjective satisfaction*  
Allow for user feedback via interviews, free-form comments and satisfaction scales

- Trade-offs in design options frequently occur. Changes to the interface in a new version may create consistency problems with the previous version, but the changes may improve the interface in other ways or introduce new needed functionality.
- Design alternatives can be evaluated by designers and users via mockups or high-fidelity prototypes. The basic tradeoff is getting feedback early and perhaps less expensively in the development process versus having a more authentic interface evaluated.

# Usability motivations

Many interfaces are poorly designed and this is true across domains:

- Life-critical systems
  - Air traffic control, nuclear reactors, power utilities, police & fire dispatch systems
  - High costs, reliability and effectiveness are expected
  - Length training periods are acceptable despite the financial cost to provide error-free performance and avoid the low frequency but high cost errors
  - Subject satisfaction is less an issue due to well motivated users

- Industrial and commercial uses
  - Banking, insurance, order entry, inventory management, reservation, billing, and point-of-sales systems
  - Ease of learning is important to reduce training costs
  - Speed and error rates are relative to cost
  - Speed of performance is important because of the number of transactions
  - Subjective satisfaction is fairly important to limit operator burnout

- **Office, home, and entertainment applications**
  - Word processing, electronic mail, computer conferencing, and video game systems, educational packages, search engines, mobile device, etc.
  - Ease of learning, low error rates, and subjective satisfaction are paramount due to use is often discretionary and competition fierce
  - Infrequent use of some applications means interfaces must be intuitive and easy to use online help is important
  - Choosing functionality is difficult because the population has a wide range of both novice and expert users
  - Competition cause the need for low cost

- **Exploratory, creative, and cooperative systems**
  - Web browsing, search engines, artist toolkits, architectural design, software development, music composition, and scientific modeling systems
  - Collaborative work
  - Benchmarks are hard to describe for exploratory tasks and device users
  - With these applications, the computer should "vanish" so that the user can be absorbed in their task domain

- Social-technical systems
  - Complex systems that involve many people over long time periods
  - Voting, health support, identity verification, crime reporting
  - Trust, privacy, responsibility, and security are issues
  - Verifiable sources and status feedback are important
  - Ease of learning for novices and feedback to build trust
  - Administrators need tools to detect unusual patterns of usage

## Universal Usability

- **Physical abilities and physical workplaces**
  - Basic data about human dimensions comes from research in *anthropometry*
  - There is no average user, either compromises must be made or multiple versions of a system must be created
  - Physical measurement of human dimensions are not enough, take into account dynamic measures such as reach, strength or speed

- Screen-brightness preferences vary substantially, designers customarily provide a knob to enable user control
- Account for variances of the user population's sense perception
- Vision: depth, contrast, color blindness, and motion sensitivity
- Touch: keyboard and touchscreen sensitivity
- Hearing: audio clues must be distinct
- Workplace design can both help and hinder work performance

- The draft standard *Human Factors Engineering of Computer Workstations* (2002) lists these concerns:
  - Work-surface and display-support height
  - Clearance under work surface for legs
  - Work-surface width and depth
  - Adjustability of heights and angles for chairs and work surfaces
  - Posture—seating depth and angle; back-rest height and lumbar support
  - Availability of armrests, footrests, and palmrests

- Cognitive and perceptual abilities

- The human ability to interpret sensory input rapidly and to initiate complex actions makes modern computer systems possible
- The journal *Ergonomics Abstracts* offers this classification of human cognitive processes:
  - Long-term and semantic memory
  - Short-term and working memory
  - Problem solving and reasoning
  - Decision making and risk assessment
  - Language communication and comprehension
  - Search, imagery, and sensory memory
  - Learning, skill development, knowledge acquisition and concept attainment

- They also suggest this set of factors affecting perceptual and motor performance:
  - Arousal and vigilance
  - Fatigue and sleep deprivation
  - Perceptual (mental) load
  - Knowledge of results and feedback
  - Monotony and boredom
  - Sensory deprivation
  - Nutrition and diet
  - Fear, anxiety, mood, and emotion
  - Drugs, smoking, and alcohol
  - Physiological rhythms
- But note, in any application, background experience and knowledge in the task domain and the interface domain play key roles in learning and performance

- **Personality differences**

- There is no set taxonomy for identifying user personality types
- Designers must be aware that populations are subdivided and that these subdivisions have various responses to different stimuli
- Myers-Briggs Type Indicator (MBTI)
  - extroversion versus introversion
  - sensing versus intuition
  - perceptive versus judging
  - feeling versus thinking

- **Cultural and international diversity**

- Characters, numerals, special characters, and diacriticals
- Left-to-right versus right-to-left versus vertical input and reading
- Date and time formats
- Numeric and currency formats
- Weights and measures
- Telephone numbers and addresses
- Names and titles (Mr., Ms., Mme.)
- Social-security, national identification, and passport numbers
- Capitalization and punctuation
- Sorting sequences
- Icons, buttons, colors
- Pluralization, grammar, spelling
- Etiquette, policies, tone, formality, metaphors

- **Users with disabilities**
  - Designers must plan early to accommodate users with disabilities
  - Early planning is more cost efficient than adding on later
  - Businesses must comply with the "Americans With Disabilities" Act for some applications
- **Elderly Users**
  - Including the elderly is fairly easy, designers should allow for variability within their applications via settings for sound, color, brightness, font sizes, etc.

- **Potential research topics**
  - Reducing anxiety and fear of computer usage
  - Graceful Evolution
  - Specification and implementation of interaction
  - Direct manipulation
  - Input devices
  - Online assistance
  - Information exploration

# Goals for our profession

- **Potential research topics**
  - Reducing anxiety and fear of computer usage
  - Graceful Evolution
  - Specification and implementation of interaction
  - Direct manipulation
  - Input devices
  - Online assistance
  - Information exploration

# Goals for our profession (cont.)

- **Providing tools, techniques, and knowledge for system implementers**
  - Rapid prototyping is easy when using contemporary tools
  - Use general or self-determined guideline documents written for specific audiences
  - To refine systems, use feedback from individual or groups of users
- **Raising the computer consciousness of the general public**
  - Many novice users are fearful due to experience with poor product design,
  - Good designs help novices through these fears by being clear, competent, and nonthreatening

## 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

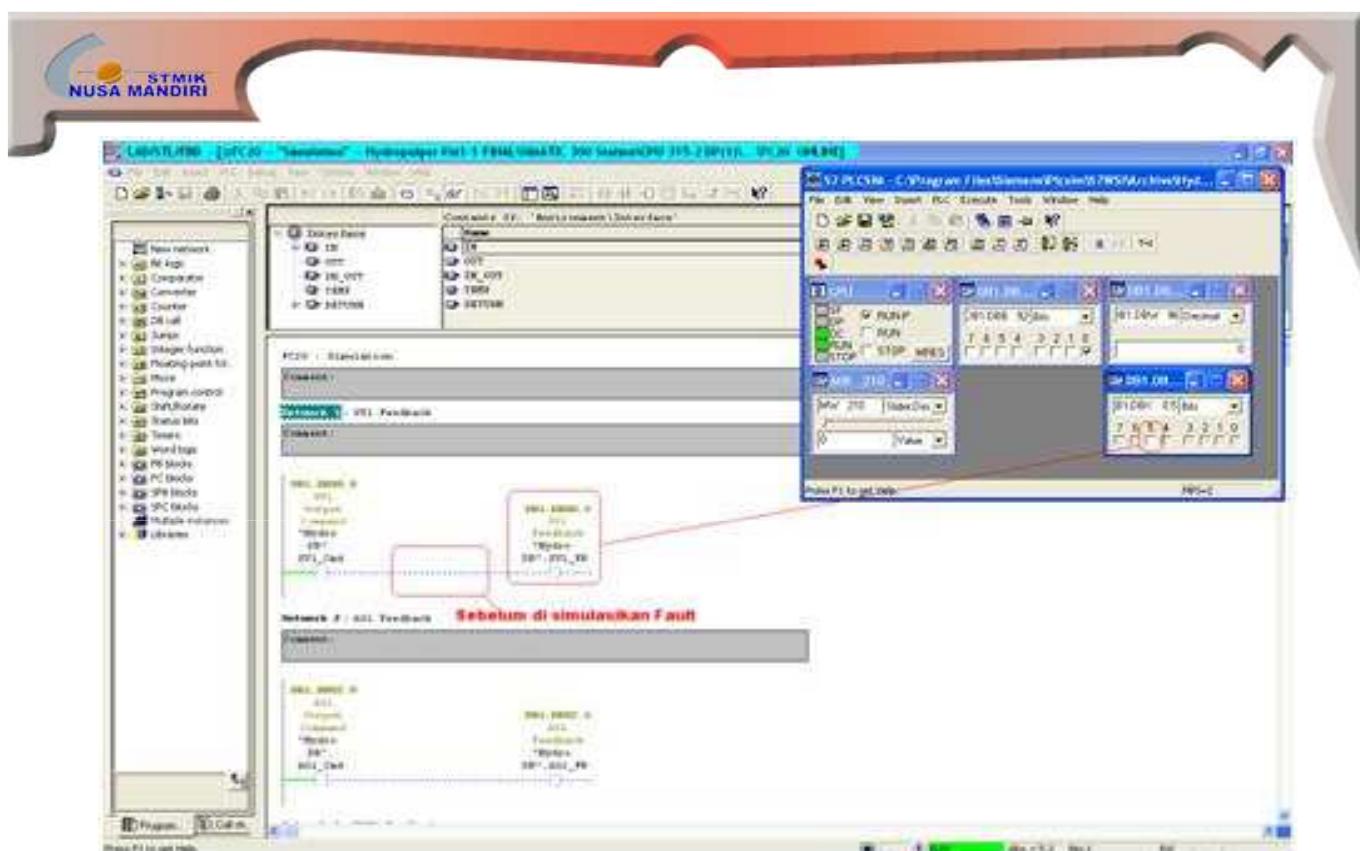
- Menggunakan bermacam bahasa
- Penggunaan yang baik
- Kritik dalam berbagai cara seperti :
  - Terlalu fokus, Tidak serba ada, penerapan yang susah, dan kadang salah
- Pendukung
  - Merangkum semua pengalaman

## 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 langkah Nagasi interface,
    - Buat urutan tugas2 yang baku
    - Memastikan semua hubungan dideskripsikan 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

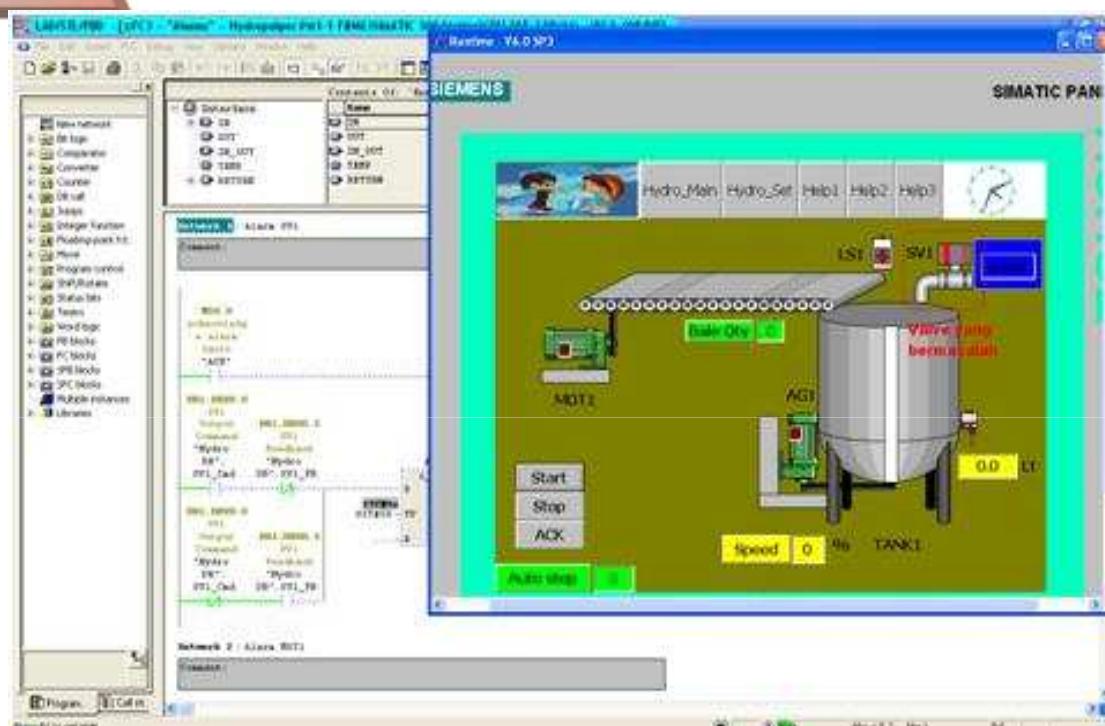


Contoh : Navigasi interface

## 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.

 STMIK  
NUSA MANDIRI

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.

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.

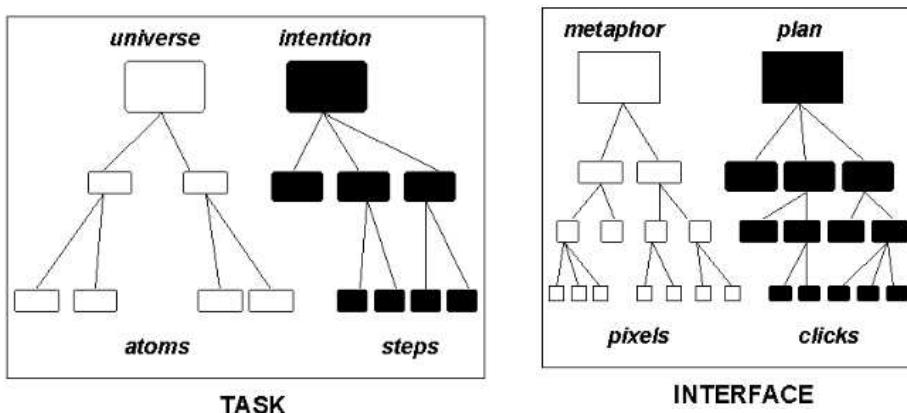


Figure from Designing the user interface by Ben Shneiderman

### 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.

# Pertemuan 6

Interaction styles and Interaction device (1)

## Materi yang diakan di bahas

- Bebagai style dasar ( sistem perintah, sistem menu, form)
- Teknik manipulasi
- Bahasa alami
- Obyek dan sumberdaya dialog (sistem window, dialog box, icon )
- Perangkat I/O

## 1. Interaction styles and Interaction device I

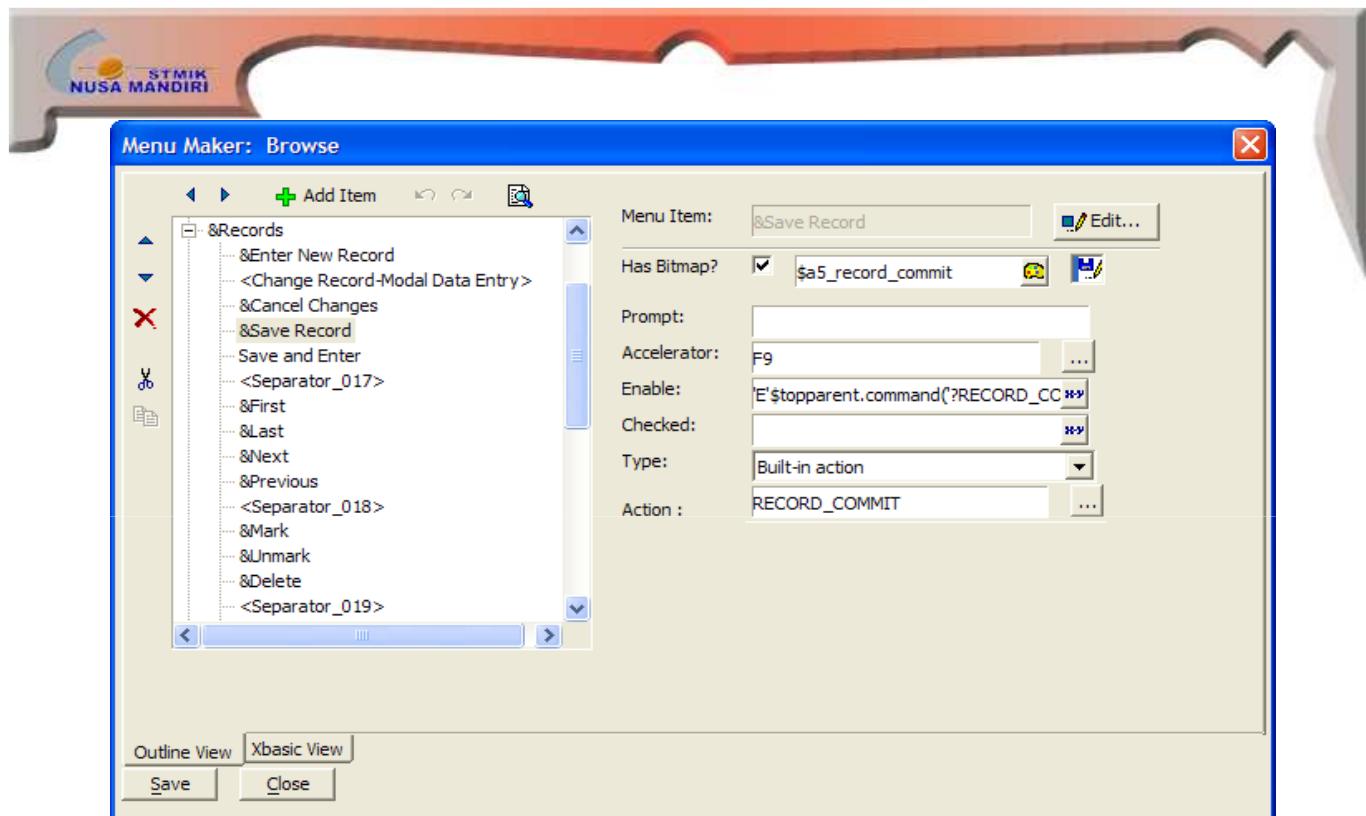
### Pendahuluan

Ketika para perancang tidak dapat membuat rancangan yang tepat, maka pilihan menu dan form isian adalah pilihan yang menarik dan efektif

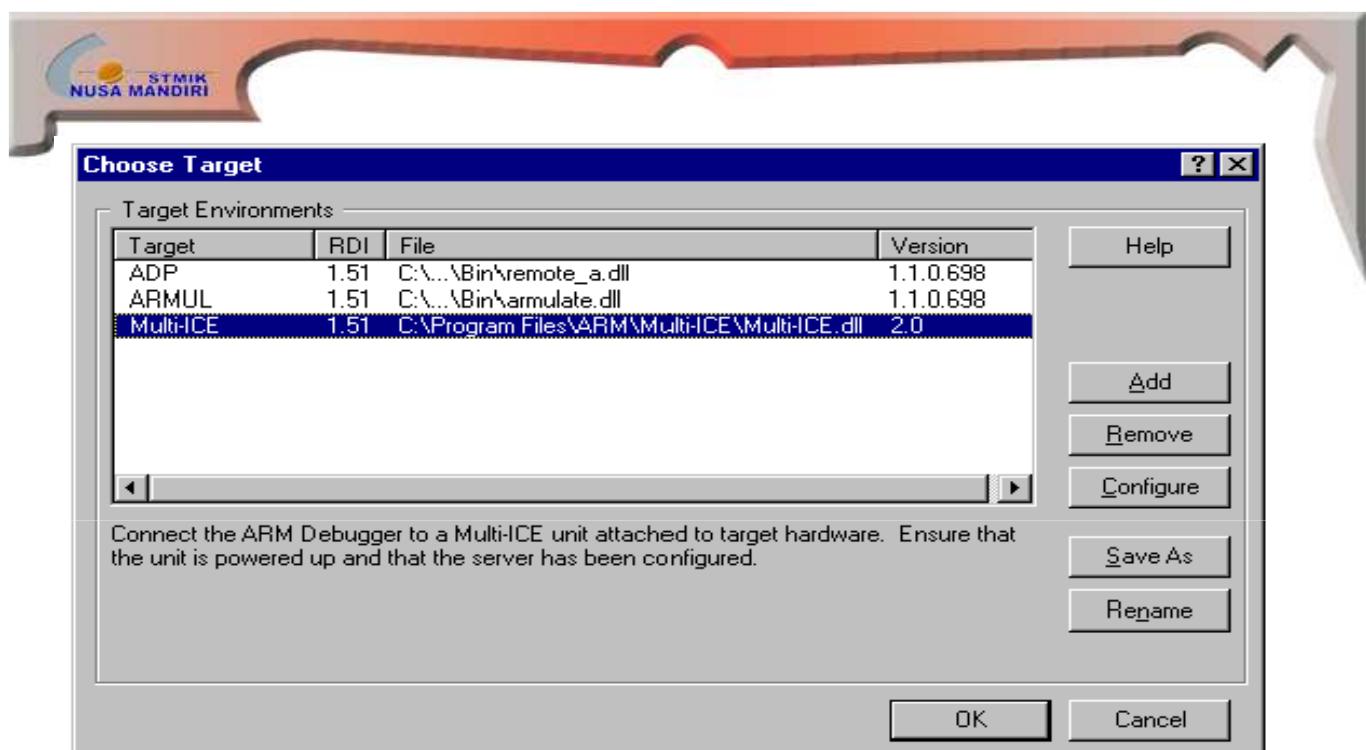
Pada saat bagian menu di tulis dengan istilah2 yang sudah dikenal pemakai yang dikelola dengan struktur yang mudah maka pemakai akan memilih item2 perintah dengan mudah

## Task Related Menu Organization

Tujuan utama dari pembuatan menu, form isi/sistem perintah dan dialog box adalah untuk menciptakan perancangan yang mudah dipahami, dapat dimengerti, mudah diingat, yang sesuai dengan tugas2 pemakai

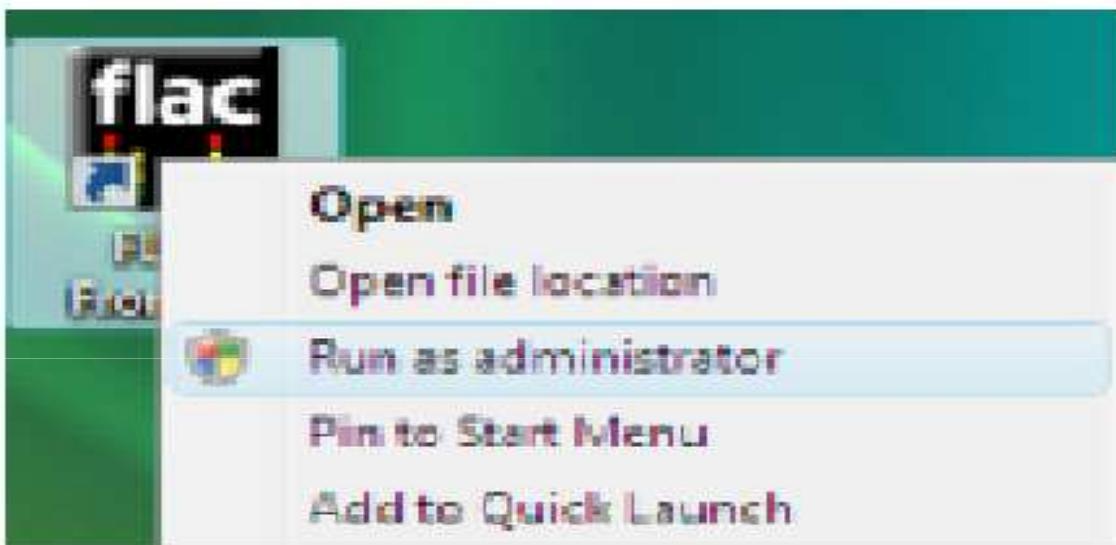


## Contoh Interaksi menu



## Contoh : dialogbox





## Pull-down, Pop-up and ribbon menus

Pull down menu adalah menu-menu yang dapat selalu diakses pengguna dengan cara memilih pada top menu bar, Item2 menu pada Pulldown seperti : File, Edit Format, View dan Help.



Gambar berbagai bentuk pull down

menu

STMIK NUSA MANDIRI

COPYRIGHT © Maret 2013



Ciri khusus dari bentuk dialog ini adalah penyajian langsung dari suatu aktifitas oleh sistem kepada user , dimana aktifitas itu akan dikerjakan oleh sistem komputer ketika user memberika instruksi lewat kenyataan maya. yang ada ditampilkan layar monitor

(virual reality) adalah lingkungan buatan yang didesain dengan hardware dan software dan dipresentasikan ke user dengan cara sedemikian sehingga lingkungan tersebut tampil dan terasa seperti lingkungan asli



Macam dialog ini adalah

## Simulator

Simulator adalah sistem miniatur yang menirukan kerja suatu sistem yang berskala sangat besar atau sangat kecil

Contoh :

1. Simulator penerbangan.

Didalam simulator penerbangan, seorang calon pilot seolah-olah sedang berada didalam sebuah pesawat yang menjadi tanggung jawab secara penuh. Meski dengan tampilan yang tidak selengkap papan kontrol pada pesawat, calon pilot dapat mempelajari hal-hal yang sangat mendasar agar ia dapat menerbangkan suatu pesawat terbang.





Program Studi  
Teknik Informatika

STMIK NUSA MANDIRI

**COPYRIGHT © Maret 2013**



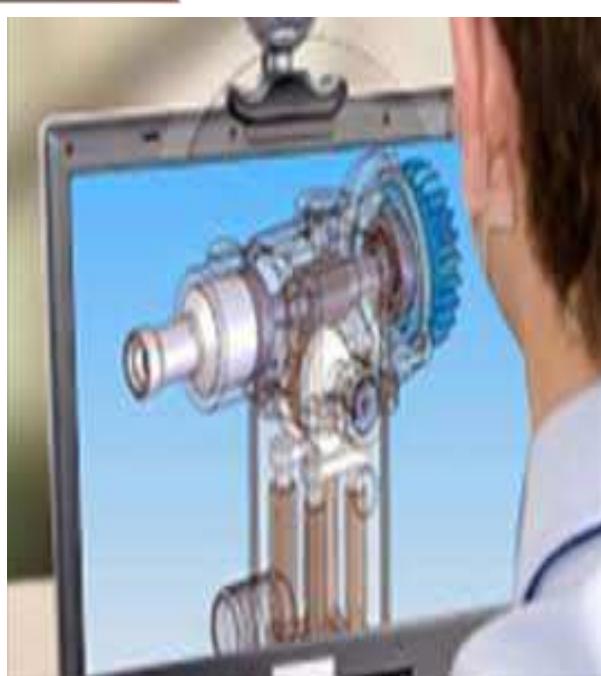
Program Studi  
Teknik Informatika

**STMIK NUSA MANDIRI**

**COPYRIGHT © Maret 2013**

## 2. Computer Aided Desain

**Computer Aided Design** adalah suatu program komputer untuk menggambar suatu produk atau bagian dari suatu produk. Produk yang ingin digambarkan bisa diwakili oleh garis-garis maupun simbol-simbol yang memiliki makna tertentu. CAD bisa berupa gambar 2 dimensi dan gambar 3 dimensi.



Gambar Computer Aided Desain

### 3. Bahasa Alamiah / Natural Language in computing

Sebelum ada teknologi komputer, manusia sudah memiliki mimpi tentang membuat mesin yang dapat memprosesa bahasa alamiah, hal ini sungguh mengagumkan yang ditandai dengan diciptakannya alat pemrosesan kata seperti word processor, audio recordes dan telepon yang memberikan kemajuan kepada manusia

Namun bahasa alamiah adalah hal yang tidak...., banyak kasus2 khusus, konteks yang rumit dan hubungan emosional yang kuat dan berakibat dalam komunikasi

Natural language Processing (NLP) telah membuat terobosan dan fokus baru pada on line system yang menggambarkan NLP toolkit yang sesuai untuk open source Seperti Python modul

Disini user memberikan instruksi dalam bahasa alami yang sifatnya lebih umum. User dapat memberikan perintah secara lebih bebas dan “manusiawi”. Sehingga komputer harus mempunyai kemampuan untuk mengolah bahasa alami yang bertujuan untuk mengetahui arti(semantik) dari instruksi yang diberikan. Sebuah sistem yang engimplementasikan dialog berbasis bahasa alami harus mempunyai sebuah sistem penterjemah (*interpreter*) yang dapat menterjemahkan kalimat pada dua arah.

```
NU STMIK
C:\Windows\system32\cmd.exe "C:\Users\Cendela\Desktop\Chatterbot Eliza 2.0\Debug\Eliza.exe"
HI USER, CAN YOU PLEASE TELL ME YOUR NAME?
>my name is Gonzales.
GLAD TO MEET YOU GONZALES.
>ok Eliza!
PLEASE BE MORE SPECIFIC.
>what do you mean by that?
SORRY I CAN'T PROVIDE MORE EXPLANATION ON THIS.
>why not?
WHY ARE YOU ASKING ME THIS?
>i dont know.
HOW COME YOU DONT KNOW?
>i'm just kidding.
WHAT MAKES YOU THINK THAT YOU ARE JUST KIDDING?
>well, because it is true.
SO YOU ARE TELLING ME THE REASON.
>yes, exactly.
YOU SOUND VERY OPTIMISTIC.
>thank you for the compliment!
DONT MENTION IT.
>ok.
PLEASE BE MORE SPECIFIC.
>i will try to be more specific next time.
DO YOU LIKE TALKING ABOUT YOURSELF?
>so, you think that i was talking about myself.
YOU ARE NOT REALLY TALKING ABOUT ME, ARE YOU?
>well, not really.
```

## Contoh Natural language

## 4. Obyek dan sumber daya dialog (sistem Windows, dialog Box dan Icon)

### SISTEM WINDOW

Sistem window adalah sistem *interface* yang memungkinkan *user* menampilkan berbagai informasi baik sendiri-sendiri maupun secara bersama, kedalam bagian layar yang tidak saling mempengaruhi. Sistem window dioperasikan, pada mode grafik. Informasi berbasis grafis memerlukan memori yang jauh lebih besar.

Dari sisi *programmer*, penerapan berbagai teknik antarmuka grafis pada sebuah program aplikasi jelas mempersulit pekerjaan. Tetapi bagi *user*, kemudahan dan keramahan dalam menggunakan suatu program aplikasi merupakan satu aspek yang sangat penting untuk dipahami oleh perancang, antarmuka prorgam aplikasi.

Mengikuti penggunaan simbol-simbol dalam kehidupan sehari-hari. Dialog berbasis icon sebenarnya adalah variasi dari dialog berbasis menu.



Contoh menu icon

STMIK NUSA MANDIRI

COPYRIGHT © Maret 2013

## 6. Perangkat I/O dalam Sistem Interaksi

### Perangkat Masukan

Untuk penggunaan interaktif perlu digunakan text entry, drawing dan selection dari screen input yang lebih memfokuskan pada perekaman dan pemasukan data ke dalam sistem komputer dan memberikan command kepada komputer, atau dengan kata lain bahwa alat input harus dapat digunakan oleh pengguna dalam berinteraksi dengan sistem komputer dan harus dapat dimengerti

Perangkat masukan :

1. Text entry : Keyboard, speed and handwritting
2. Pointing : Mouse

### Perangkat Keluaran

Pada umumnya peralatan yang dapat menghasilkan keluaran dari hasil pengolahan system, perangkat keluaran ini bergantung dari hasil keluran yang dihasilkan oleh sistem inter aski

Pertemuan 9

## Interaction Styles and Interaction Device (2)

Materi yang akan di bahas

- Teknologi masukan dan keluaran
- Klasifikasi piranti masukan dan keluaran berdasarkan tugas pemakai
- Mengetahui berbagai jenis I/O dan karakteristiknya
- Menentukan berbagai I/O yang sesuai digunakan dalam suatu sistem interaksi berdasarkan kondisi dan spesifikasinya
- Mengetahui kondisi terakhir dan yang sedang berkembang dalam teknologi I/O

1. Teknologi masukan dan keluaran
  1. Keyboard dan keypads
  - a. Alat input langsung dapat digolongkan ke dalam beberapa golongan, antara lain yaitu :
    - Keyboard
    - Pointing device
    - Scanner
    - Cencor (Biometrik)
    - Voice recognizer.

## Prinsip kerja keyboard

- Computer keyboard sebagian merupakan perwujudan dari keyboard mesin ketik.
- Secara fisik computer keyboard berupa suatu susunan tombol-tombol persegi empat (disebut : key) yang memiliki karakter-karakter yang dituliskan di atas key dan masing-masing memiliki fungsi yang saling berhubungan.
- Dari keseluruhan tombol key, 50 % diantaranya menghasilkan letter (huruf), number (angka), sedangkan tombol-tombol key yang lainnya dapat menghasilkan “action”, contohnya pada tombol key  $\leftarrow\uparrow\downarrow\rightarrow$ , dan lain-lain.
- Pada computer modern, tombol-tombol key biasanya sudah ada dalam software, sehingga dalam penggunannya tidak tergantung lagi pada operator.

- Ada banyak macam susunan simbol-simbol yang terdapat pada keyboard. Perbedaan layout keyboard sebagian besar disebabkan oleh kebutuhan orang untuk mengakses simbol yang berbeda secara mudah. Secara khas, hal tersebut dikarenakan perbedaan penulisan bahasa, namun ada juga yang mengkhususkan desain keyboard untuk matematika, akuntansi, dan programming computer.
- Layout QWERTY sudah menjadi standar selama beberapa dekade sebelum adanya keyboard computer. Pada saat itu banyak digunakan untuk electronic keyboard, seperti Dvorak Simplified Keyboard, Colemak dan sebagainya yang sekarang sudah tidak dipergunakan lagi.
- Pada tahun 1990an, jumlah key bervariasi dari standar awal 101 hingga 104 ditambah dengan beberapa “additional key” (tombol tambahan), seperti key untuk membrowser web atau untuk email client.

## Computer Keyboard Keys

<b>Dead keys</b>	Compose
<b>Modifier keys</b>	Control · Shift · Alt/Option (Apple) · AltGr · Command/Meta (MIT keyboards) · Windows/Super · Fn (compact keyboards)
<b>Lock keys</b>	Scroll lock · Num lock · Caps lock
<b>Navigation</b>	Arrow · Page scrolling (Page up/Page down) · Home/End
<b>Editing</b>	Return/Enter · Backspace · Insert · Delete · Tab · Space bar
<b>Misc.</b>	SysRq/Print screen · Break/Pause · Escape · Menu · Numeric keypad · Function · Power management (Power, Sleep, Wake) · Language input · Any key

## Kombinasi Key

Key combination	Microsoft Windows/KDE	Emacs/Linux (command line and programs using readline)
Ctrl+A	Select all	Beginning of line
Ctrl+B	Bold	Backward one character
Ctrl+C	Copy (can also be used as an alternative to Ctrl+Break to terminate an application)	Compound command (emacs) or terminate application
Ctrl+D	Font Window (Word Processing)	Forward delete or end of input
Ctrl+E	Center Alignment (Word Processing)	End of line
Ctrl+F	Find (usually a small piece of text in a larger document)	Forward one character
Ctrl+H	Find and Replace	?
Ctrl+L	Create List	Vertically center current line in window/terminal
Dll.		



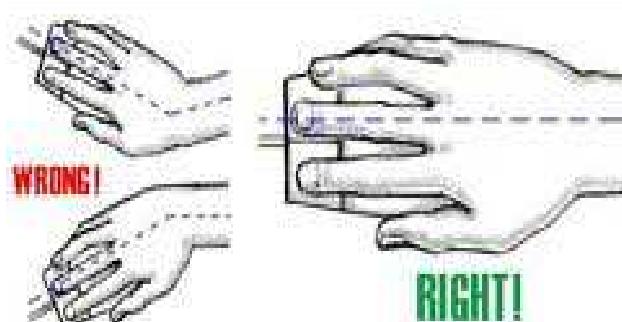
## 2. POINTING DEVICES

### Jenis Jenis Pointing Devices

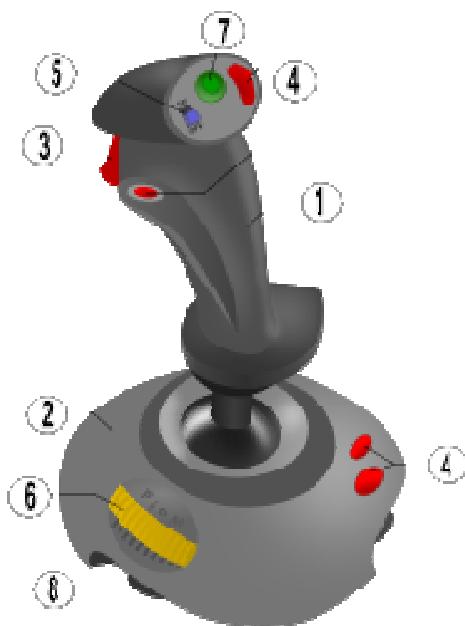
- Mouse
- Trackball
- touchpad
- touchscreen
- light pen
- light gun
- steering wheel
- graphics tablets
- DLL

## Mouse dan cara kerja

- Mouse bekerja dengan menangkap gerakan menggunakan bola yang menyentuh sebuah permukaan yang keras dan rata. Mouse yang lebih baru biasanya tidak mempunyai bola; menggunakan sinar optik untuk mendeteksi gerakan pemakai. Selain itu, ada pula yang sudah menggunakan teknologi nirkabel, baik itu yang berbasis radio, inframerah maupun bluetooth.
- Saat ini teknologi terbaru sudah memungkinkan mouse memakai laser sehingga resolusinya dapat mencapai 2000 dpi. Versi terbaru bahkan bisa mencapai 4800 dpi. Biasanya mouse semacam ini diperuntukkan untuk para pemain game.



### 3. Joystick



Joystick elements:

1. Stick
2. Base
3. Trigger
4. Extra buttons
5. Autofire switch
6. Throttle
7. Hat Switch (POV Hat)
8. Suction Cup

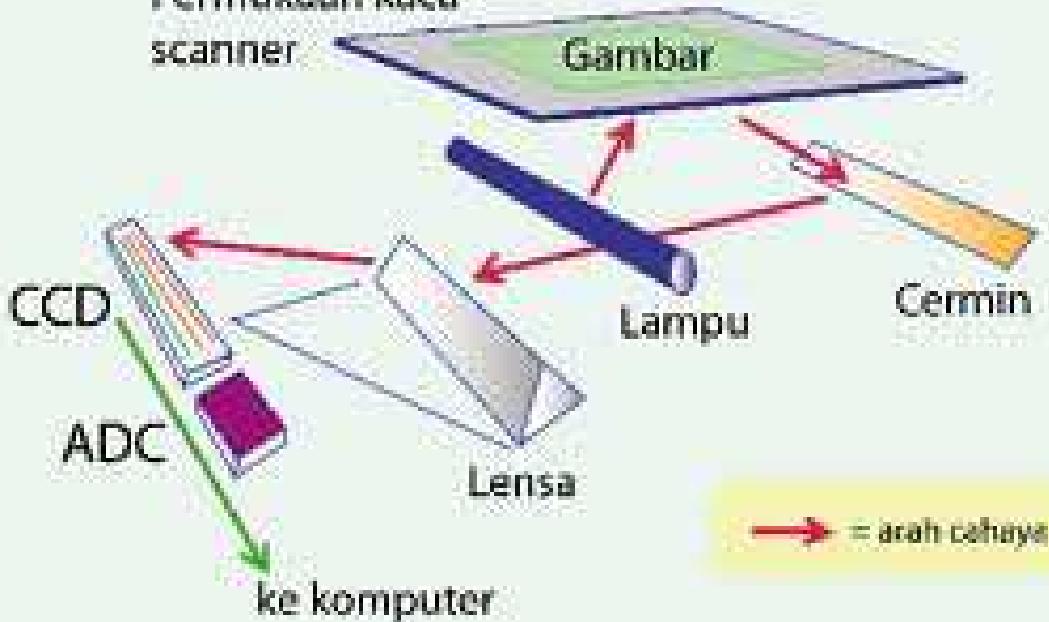
### 4. SCANNER(PEMINDAI)

- **Pemindai** atau **scanner** merupakan suatu alat yang digunakan untuk memindai suatu bentuk maupun sifat benda, seperti dokumen, foto, gelombang, suhu dan lain-lain. Hasil pemindaian itu pada umumnya akan ditransformasikan ke dalam komputer sebagai data digital.

#### Kegunaan Scanner

- Scanner berguna untuk mengcopy data (bentuk, text, image) dengan cepat, dan mengkonversikannya ke dalam data digital.
- Secara khusus kegunaan scanner tergantung pada jenisnya, misalnya: scanner logam untuk mendeteksi logam, scanner OMR dapat digunakan untuk memeriksa lembar jawaban, dan pemindai sinar x yang dikembangkan untuk kebutuhan dunia kedokteran.

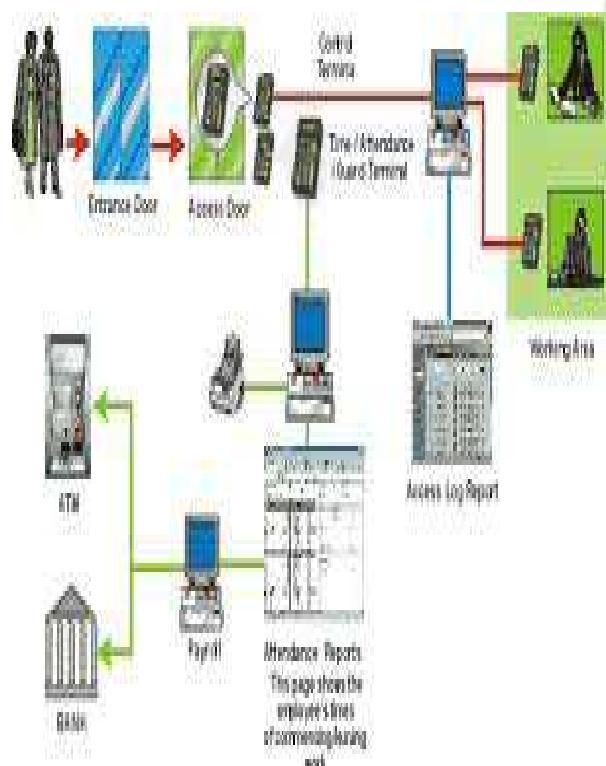
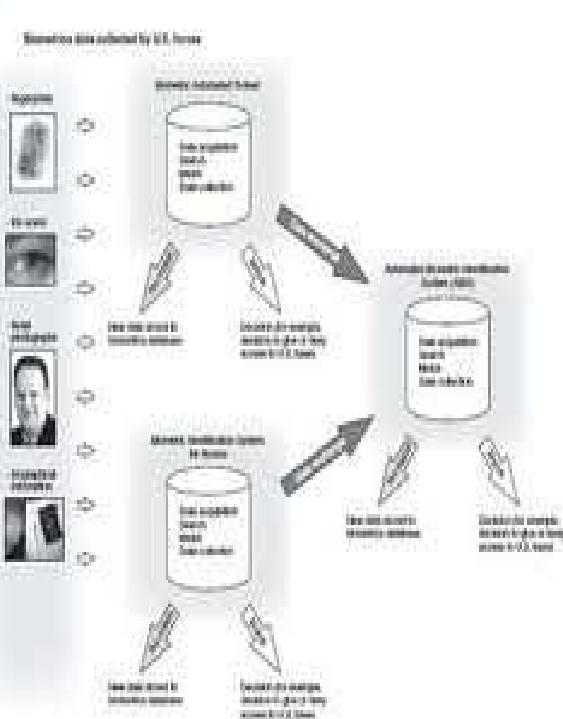
Permukaan kaca  
scanner

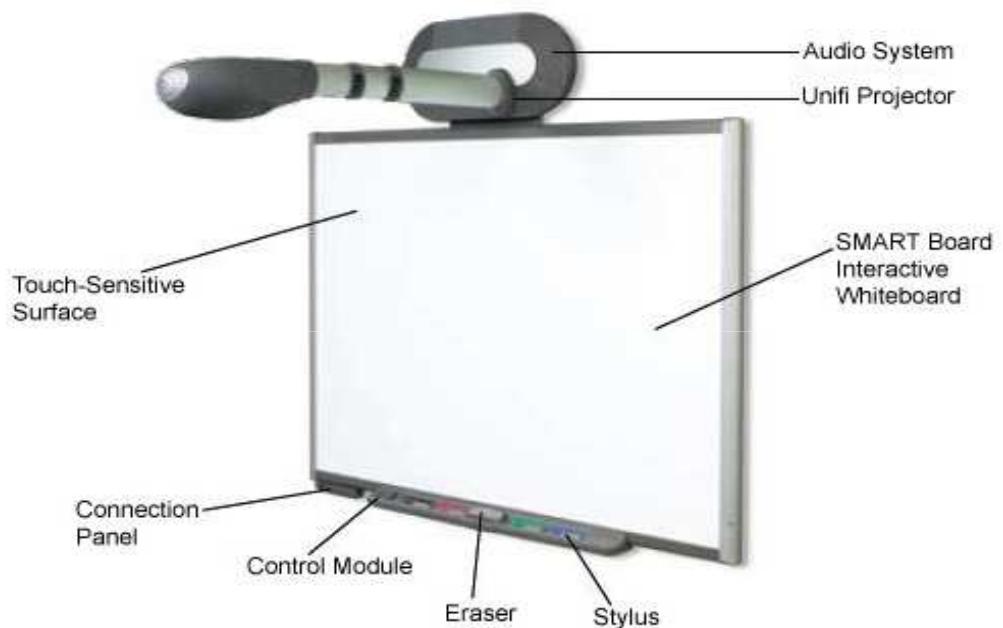


gambar prinsip kerja scanner

# Biometrik

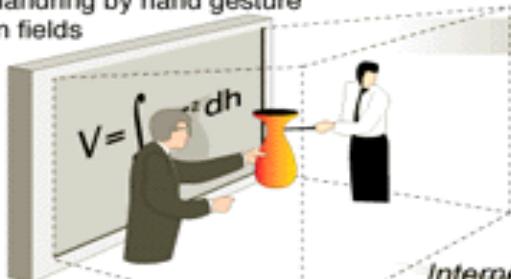
- **Biometrik** (berasal dari bahasa Yunani *bios* yang artinya hidup dan *metron* yang artinya mengukur)
- Adalah studi tentang metode otomatis untuk mengenali manusia berdasarkan satu atau lebih bagian tubuh manusia atau kelakuan dari manusia itu sendiri.
- Dalam dunia TI, biometrik relevan dengan teknologi yang digunakan untuk menganalisa fisik dan kelakuan manusia untuk autentifikasi. Contohnya dalam pengenalan fisik manusia yaitu dengan pengenalan :
  - Sidik jari,
  - Retina, iris,
  - Pola dari wajah (*facial patterns*),
  - Tanda tangan dan cara mengetik (*typing patterns*).
  - Suara, tanda tangan atau irama dalam menulis (perilaku)
- Yang dilakukan, sederhananya, adalah mencocokkan gambaran digital dari karakter-karakter yang diamati dengan dengan rekaman karakter di gudang data (database).
- Komputer memungkinkan pencocokkan itu berlangsung lebih cepat,





### Cooperative Work

- Object handing by hand gesture
- Coaction fields



### 3-D image synthesis

- Image rendering of human objects from coded data
- HyperClass scene composition



### Intelligent model-based coding

- Coding of 3-D human body and face motions
- Coding algorithm



### Image recognition

- Feature extraction from input images(Edge detection)
- Measurement of human face directions, Recognition of human body figures

4. Menentukan I/O (user interface) yang sesuai untuk digunakan dalam sistem interaksi

Menurut wikipedia Indonesia , antar muka pengguna ( user interface) adalah tampilan garfis yang menghubungkan user dengan sistem,

Tujuan dari desain antar muka ini adalah pengoperasian dan kontrol sistem yang efektif serta umpan balik dari sistem

Pertimbangan perancangan antar muka terjadi ketika merancang antarmuka pengguna yang berkaitan atau melibatkan berbagai disiplin ilmu seperti ergonomi dan psikologi , serta disesuaikan dengan kebutuhan pengguna agar sistem lebih efektif dan usability



dari keterangan diatas maka hal – hal yang dibutuhkan dalam menentukan perangkat I/O atau perancangan antar muka adalah :

1. Siapa pengguna ( User)
2. Faktor psikologis dan kognitif user
3. Bentuk sistem yang dirancang
4. Kehandalan dan ke efektifan dari perangkat
5. Usability

## 5. Kondisi perkembangan I/O



## Perangkat Masukan dan keluaran



## Contoh perkembangan alat input



Contoh peralatan output

## Pertemuan 10

# Metodologi perancangan sistem interaksi dan evaluasinya (1)

## Pokok Bahasan :

- 1.Tiga pilar perancangan
- 2.Metodologi pengembangan
- 3.Ethnographic Obsevation
- 4.Skenario Pengembangan
- 5.Expert Reciews

### 1. The Three Pillars of Design

- **Dokumen dan proses yang menjadi Pedoman**

Setiap proyek memiliki kebutuhan yang berbeda, oleh karena itu pedoman adalah hal yang harus dipertimbangkan untuk digunakan, :

### 1. Words, icons, and graphics

- Terminology (objects and actions), abbreviations, and capitalization
- Character set, fonts, font sizes, and styles (bold, italic, underline)
- Icons, graphics, line thickness, and
- Use of color, backgrounds, highlighting, and blinking

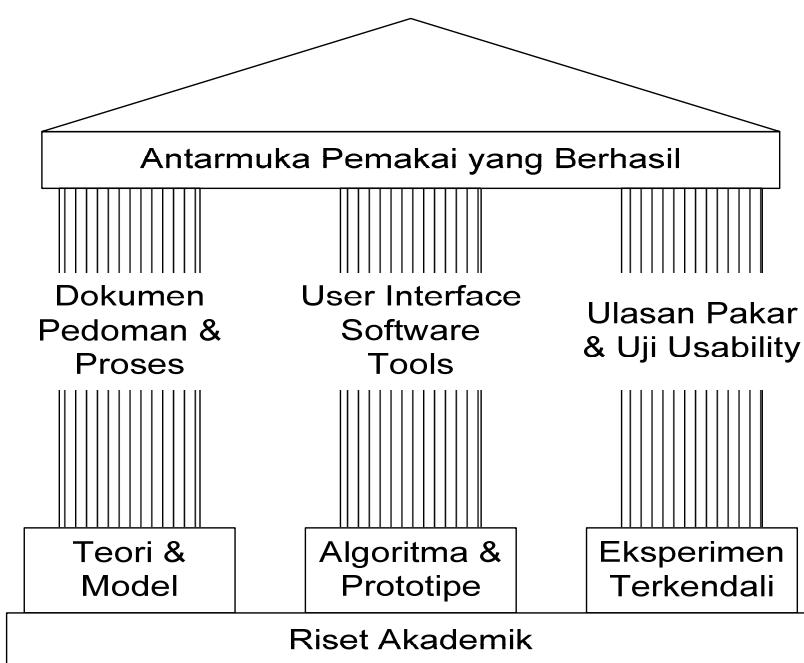
## 2. Tampilan keluaran

- Menu selection, form fill-in, and dialog-box formats
- Wording of prompts, feedback, and error messages
- Justification, white space, and margins
- Data entry and display formats for items and lists
- Use and contents of headers and footers

## 3. Input and output devices

- Keyboard, display, cursor control, and pointing devices
- Audible sounds, voice feedback, touch input, and other special devices
- Response time for a variety of tasks

## Tiga Pilar Perancangan



## 2. Metodologi pengembangan sistem interaksi

### The Logical User-Centered Interactive Design Methodology (LUCID) (Kreitzberg):

1. Kembangkan konsep produk
2. Riset dan analisis kebutuhan
3. Konsep perancangan dan prototipe layar kunci
4. Perancangan iteratif dan perbaikan
5. Implementasikan software
6. Dukungan rollout

### The Twelve areas of the LUCID Management Strategy

- |                       |   |
|-----------------------|---|
| 1. Definisi produk,   | 8. Prototipe,                             |
| 2. Business case,     | 9. Usability,                             |
| 3. Sumber daya,       | 10. Panduan perancangan,                  |
| 4. Lingkungan fisik,  | 11. Panduan isi,                          |
| 5. Lingkungan teknis, | 12. Dokumentasi (pelatihan dan petunjuk). |
| 6. Pemakai,           |   |
| 7. Fungsionalitas,    |   |

### **Role by phase:** process navigator

Role by phase matrix

Role/phase matrix	All Phases	Business Opportunity	Understanding Users	Initial Design	Development	Deployment	Life Cycle
All Roles							
User Experience Leadership	User Engineering Plan - Initial	User Engineering Plan - Final	Execution of the User Engineering Plan	Satisfaction of established metrics	Project Assessment	Satisfaction Survey	
Market Planning	Business and Market Requirements	Appropriate User Requirements	Draft Marketing Collateral	Detail Marketing Collateral	Final Marketing Collateral		
User Research		User Requirements	Appropriate Design				
User Experience Design		Design Direction	Conceptual Design, Low-Fidelity Prototypes	Detail Design, High-Fidelity Prototypes	Design Issue Resolution		
Visual & Industrial Design		Appearance Direction	Appearance Guidelines	Appearance Specification			
User Experience Evaluation		Competitive Evaluation	Conceptual Design Evaluation	Detail Design Evaluations	User Feedback and Benchmark	Usage Issue Report	

The matrix above depicts the key user engineering roles and the phases of a generic development process. At each role-phase intersection is a synopsis of the contribution that role provides during the phase. You can use this matrix to access more detailed information about each role, each development phase, and role-phase activities.

## 3. Ethnographic Observation

### • Preparation

- Pahami kebijakan dan budaya kerja organisasi.
- Kenali sistem dan sejarahnya.
- Tentukan tujuan awal dan siapkan pertanyaan.
- Minta akses dan izin untuk observasi dan wawancara.

### • Field Study

- Bangun hubungan dengan manajer dan pemakai.
- Amati atau wawancarai pemakai di tempat kerjanya.
- Kumpulkan data subjektif dan objektif, kuantitatif dan kualitatif.
- Ikuti semua petunjuk yang muncul dari kunjungan.
- Catat kunjungan

- **Analysis**
  - ❖ Gabungkan data yang dikumpulkan dalam database numeris, tekstual, dan multimedia.
  - ❖ Kuantifikasikan data dan gabungkan statistik.
  - ❖ Konsolidasikan dan interpretasikan data.
  - ❖ Perbaiki tujuan dan proses yang digunakan
- **Reporting**
  - ❖ Pertimbangkan peserta dan tujuan yang beraneka ragam.
  - ❖ Persiapkan laporan dan presentasi-kan hasil penelitian.

## 4. Scenario Pengembangan

### Day-in-the-life scenarios:

- characterize what happens when users perform typical tasks
- can be acted out as a form of walkthrough
- may be used as basis for videotape
- useful tools
  - table of user communities across top, tasks listed down the side
  - table of task sequences
  - flowchart or transition diagram

## Describe the new system and its benefits

- Convey the high level goals of the new system.
- Identify the stakeholders.
- Identify specific benefits

## 5. Expert Reciews

Ulasan pakar dapat dilakukan di awal atau di akhir fase perancangan, dan keluarannya berupa laporan formal dengan masalah yang ditemui atau rekomendasi perubahan.

Pakar yang berbeda cenderung menemukan masalah yang berbeda, maka **3-5 pakar** dapat sangat produktif sebagai uji *usability* pelengkap.

- Metode ulasan pakar:
  - Evaluasi heuristik
  - Ulasan kesesuaian dengan pedoman (*guidelines review*)
  - Pemeriksaan konsistensi
  - Penelusuran kognitif
  - Pemeriksaan *usability* formal

## Address concerns and potential barriers

- Anticipate changes in job functions and potential layoffs.
- Address security and privacy issues.
- Discuss accountability and responsibility for system misuse and failure.
- Avoid potential biases.
- Weigh individual rights vs. societal benefits.
- Assess trade-offs between centralization and decentralization.
- Preserve democratic principles.
- Ensure diverse access.
- promote simplicity and preserve what works.

## Outline the development process

- Present and estimated project schedule.
- Propose process for making decisions.
- Discuss expectations of how stakeholders will be involved.
- Recognize needs for more staff, training, and hardware.
- Propose plan for backups of data and equipment.
- Outline plan for migrating to the new system.

## Potential Controversies

- What material is eligible for copyright?
- Are copyrights or patents more appropriate for user interfaces?
- What constitutes copyright infringement?
- Should user interfaces be copyrighted?

## Pertemuan 11

# Metodologi Perancangan sistem Interaksi dan evaluasi (2)

Pokok bahasan :

- 1.Usability Testing
- 2.Instrument Testing
- 3.Acceptance Testing
- 4.Evaluasi

## 1. Usability Testing / Uji Usability

- Pengujian ekstensif dibutuhkan.
- Yang perlu diperhatikan dalam rencana evaluasi dan pengujian adalah:
  - **Tahapan perancangan** (awal, tengah, akhir).
  - **Tingkat kebaruan** proyek (terdefinisi atau bersifat eksplorasi).
  - **Jumlah pemakai** yang diperkirakan.

- **Tingkat kritis antarmuka**
- **Biaya** produk dan keuangan yang dialokasikan untuk pengujian.
- **Waktu** yang tersedia.
- **Pengalaman perancangan** dan tim evaluasi.

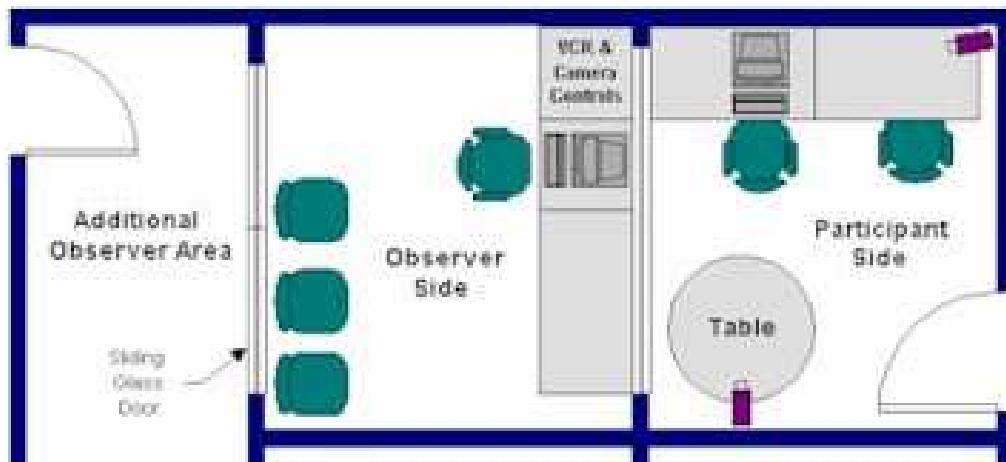
# Usability Testing / Uji Usability

- **Uji *usability* (*usability test*)** memberikan konfirmasi kemajuan untuk mendukung dan rekomendasi perubahan yang spesifik.
- Uji *usability* tidak hanya mempercepat proses, tetapi juga menghasilkan penghematan biaya
- Uji *usability* dilaksanakan oleh team work dan dilakukan didalam ruangan uji atau laboratorium

## Desain Laboratorium Usability Sederhana

- Dua ruangan 3x3 meter, dibatasi kaca satu arah.
- Satu untuk ruang kerja peserta.
- Satu untuk pengamat (perancang, manajer, pelanggan).

# Contoh Laboratorium Usability



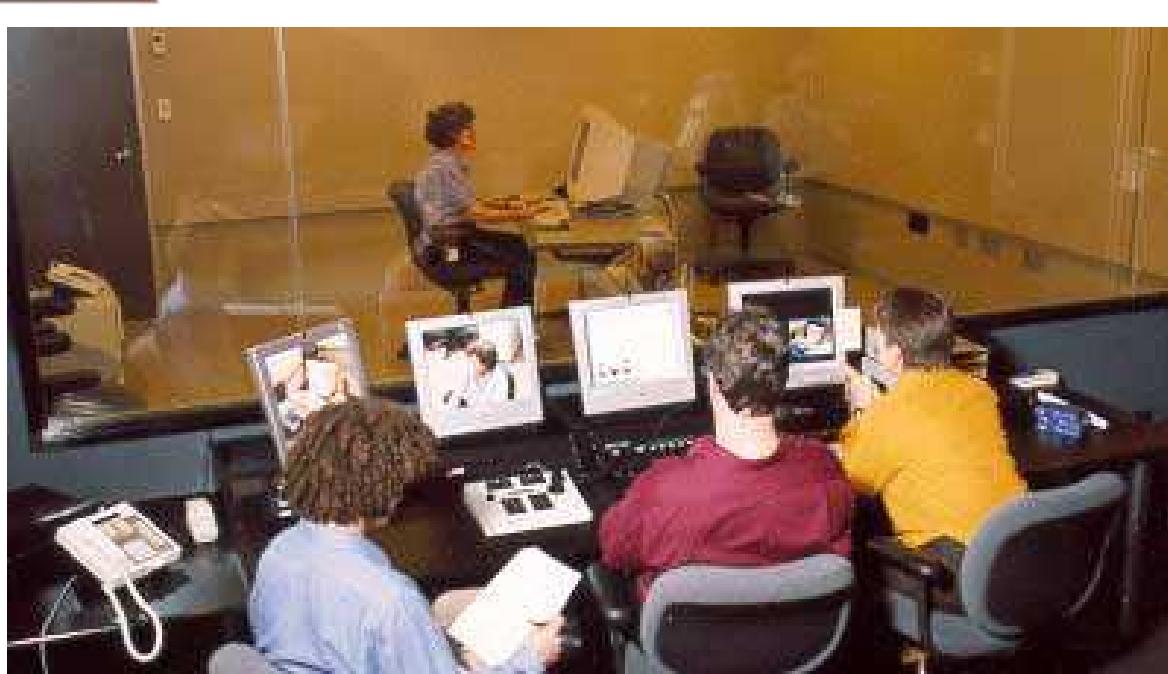
Tata letak Microsoft Usability Lab, Redmond

## Pada Kondisi Percobaan (Laboratory)

Pengujian system ini pada ruang percobaan (laboratory) harus memenuhi beberapa kondisi diantaranya:

1. Laboratorium memiliki komputer dengan perlengkapan yang memadai, serta memiliki fasilitas perekaman audio dan video yang baik.
2. Terbebas dari gangguan atau distorsi yang menghambat pekerjaan, dikarenakan pengujian dilakukan di ruang tetap
3. Sistem yang digunakan dapat ditempatkan pada lokasi yang sebenarnya baik tempat yang berbahaya ataupun tempat yang terpencil, jika sistem yang dibangun merupakan sistem simulator

4. Dapat memanipulasi situasi untuk memecahkan masalah dan melihat sedikit penggunaan prosedur atau membandingkan beberapa alternatif perancangan dengan situasi yang sebenarnya.
5. Situasi pada laboratorium tidak diasumsikan untuk menggambarkan situasi ruang kerja sebenarnya



## 2. Instrumen untuk uji usability

- Meminta pemakai **mengucapkan apa yang mereka pikirkan** dan akan kerjakan (*think aloud*).
- Menggunakan **dua peserta** bekerja bersama untuk mendukung bicara.
- **Memvideokan** kegiatan peserta untuk dilihat lagi kemudian.

### Memilih Peserta untuk Uji Usability

- Peserta dipilih mewakili komunitas pemakai dengan memperhatikan:
  - Pemahaman komputer
  - Pengalaman mengerjakan tugas
  - Motivasi dan pendidikan
  - Kemampuan bahasa alami yang digunakan dalam antarmuka.
- Peserta uji *usability* harus diberitahu bahwa **bukan mereka yang diuji**, tetapi software dan antarmuka pemakai.
- Keikutsertaan dalam uji *usability* adalah **sukarela**, dengan perjanjian terlebih dahulu.

- **Uji lapangan** berusaha menempatkan antarmuka pemakai dalam lingkungan realistik dalam periode waktu tertentu. Pencatatan (*logging*) software lebih membantu.

### 3. Uji Penerimaan (*Acceptance Tests*)

- Untuk proyek implementasi besar, klien biasanya menentukan tujuan **objektif dan terukur** untuk kinerja hardware dan software.
- Jika produk gagal memenuhi kriteria penerimaan, sistem harus diperbaiki sampai berhasil.
- Kriteria terukur dari antarmuka pemakai adalah **kelima faktor manusia terukur**.

- Setelah uji penerimaan berhasil, uji lapangan dapat meningkatkan:
  - Metode pelatihan
  - Materi tutorial
  - Prosedur bantuan melalui telefon
  - Metode pemasaran
  - Strategi publikasi

## 4. Evaluasi

1. Evaluasi dilakukan untuk melihat apakah hasil rancangan dengan proses uji coba system yang telah dibuat sesuai dengan permintaan pengguna (user). Hal ini bertujuan untuk melihat seberapa jauh sistem berfungsi dalam keadaan normal, agar menjadi pertimbangan untuk perencanaan sistem lain di kemudian hari.
2. Untuk melihat efek suatu interface ke pengguna, apakah pengguna mudah memahami tampilan sistem atau tidak.
3. Memahami problem yang terdapat pada system, kerusakan yang ditimbulkan akibat lingkungan atau karena pengguna yang ceroboh. Hal ini akan menjadi acuan dalam fase pemeliharaan sistem.
4. Evaluasi dapat dilakukan pada Laboratorium, lapangan pekerjaan, kerja sama dengan user dimana user akan memberikan feedback secara langsung maupun tidak langsung.

### Pokok Bahasan

1. Prototyping dan konstruksi interaksi
2. Konsep Perancangan
3. Desain Perencangan
4. Desain Phisik secara kongkrit
5. Tools Support

### 1. Prototyping dan Konstruksi interaksi

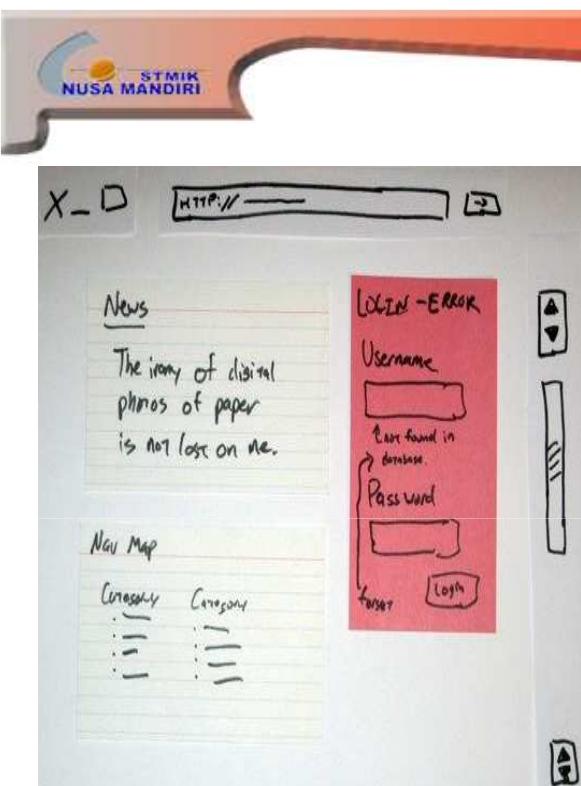
Prototyping adalah :

Model dari produk sebenarnya untuk mengkomunikasikan produk yang sedang dibangun kepada pengguna

## 1. Low-fidelity Prototyping ( prototyping dengan tingkat ketepatan yang rendah)

Beberapa karakteristik dari Low – fidelity prototype :

- Gambaran cepat dari sistem final
- Mempunyai fungsi atau interaksi yang terbatas
- Lebih menggambarkan konsep , perancangan, alternativ, dan layout layar dibanding model interaksi pengguna dengan sistem.
- Mendemonstrasikan secara umum ‘feel and look’ dari antarmuka pengguna.
- Tidak untuk memperlihatkan secara rinci bagaimana operasi sistem aplikasi.
- Digunakan pada awal siklus perancangan
- Memperlihatkan konsep pendekatan secara umum tanpa harus membuang banyak tenaga, biaya dan waktu.

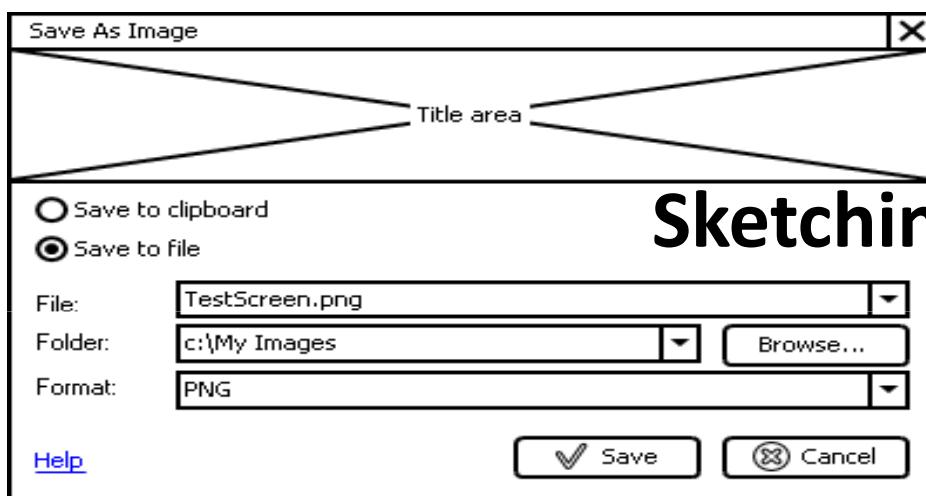


- ❖ Sederhana
- ❖ Murah
- ❖ Cepat
- ❖ Tidak mirip
- ❖ Mudah
- ❖ Eksploratif

Metode Prototyping low fidelity :

- Sketiching
- Storyboarding
- Prototyping with index card

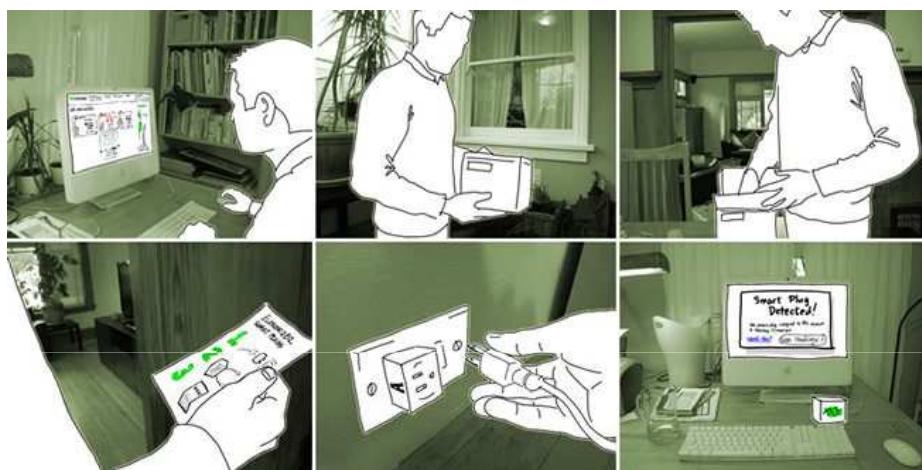
# a. Sketching



- o Gunakan ikon-ikon/simbol
- o Gambarkan elemen- elemen antarmuka
- o Sketsa dengan alat tulis atau dengan tools

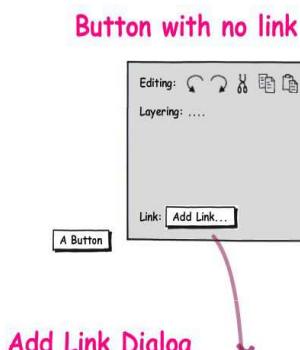


## b. storyboarding

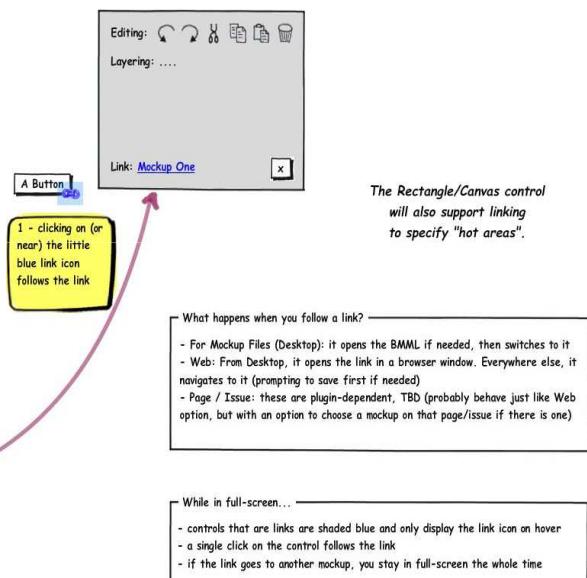


- Berkaitan dengan skenario
- Sketsa berseri
- Jelaskan rangkaian kegiatan pengguna dalam gunakan produk

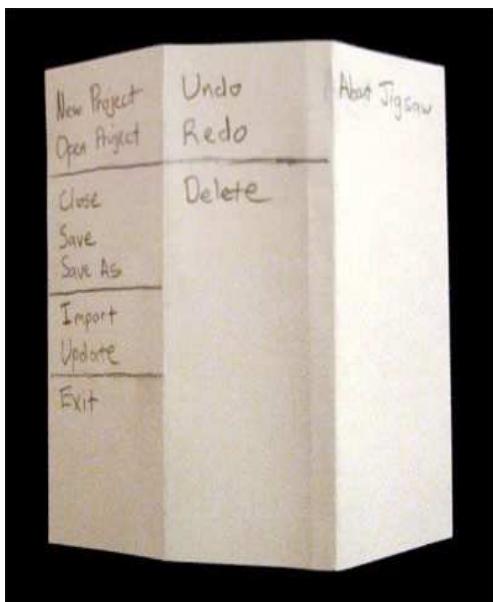
## storyboarding



**Button with Link**



## c. Prototyping with Index Cards



- Gunakan index card
- 1 kartu, 1 layar
- Desain website

Economy  
Environment + Energy  
Industry  
People

### index Cards

1	Balance of Payments	1-01
	Business Expectations	1-02
	Business Performance	1-03
	Economic Growth	1-04
	Employment	1-05
2	foreign Investment + Foreign Debt	1-06
	Foreign Trade	1-07
	Government Administration	1-08
3	Income + Expenditure (personal + household)	1-09
	Input-Output Tables	1-10
	National Income	1-11
	National Wealth + Saving	1-12
	Prices	1-13
	Productivity	1-14
	Financial Accounts	1-01-01
	Foreign Investment in Australia	1-01-02
	Imports + Exports	1-01-03
	Trade in Services	1-01-04

- ❖ Menggunakan material yang digunakan untuk membangun produk

Misal: produk adalah software akuntansi dengan visual basic, maka prototype dibangun dengan visual basic

- ❖ Masalah :

Perlu waktu lama

Perhatian evaluator pada permukaan bukan isi

Pembangun enggan ganti tool

Ekspektasi tinggi

Kesalahan kecil pada prototype sebabkan kegagalan uji

## Konsep Perencangan

- ❖ Mengubah Kebutuhan menjadi model konseptual
- ❖ Gambaran apa orang dapat lakukan dengan produk
  - Berkaitan dengan kebutuhan fungsional (sistem mampu lakukan apa)
- ❖ Konsep apa yang diperlukan untuk memahami
  - bagaimana berinteraksi dengan produk tersebut
  - Pengguna, jenis interaksi, jenis antarmuka,

## Tiga pandangan dalam perancangan model konseptual

### 1. Interface metaphors .

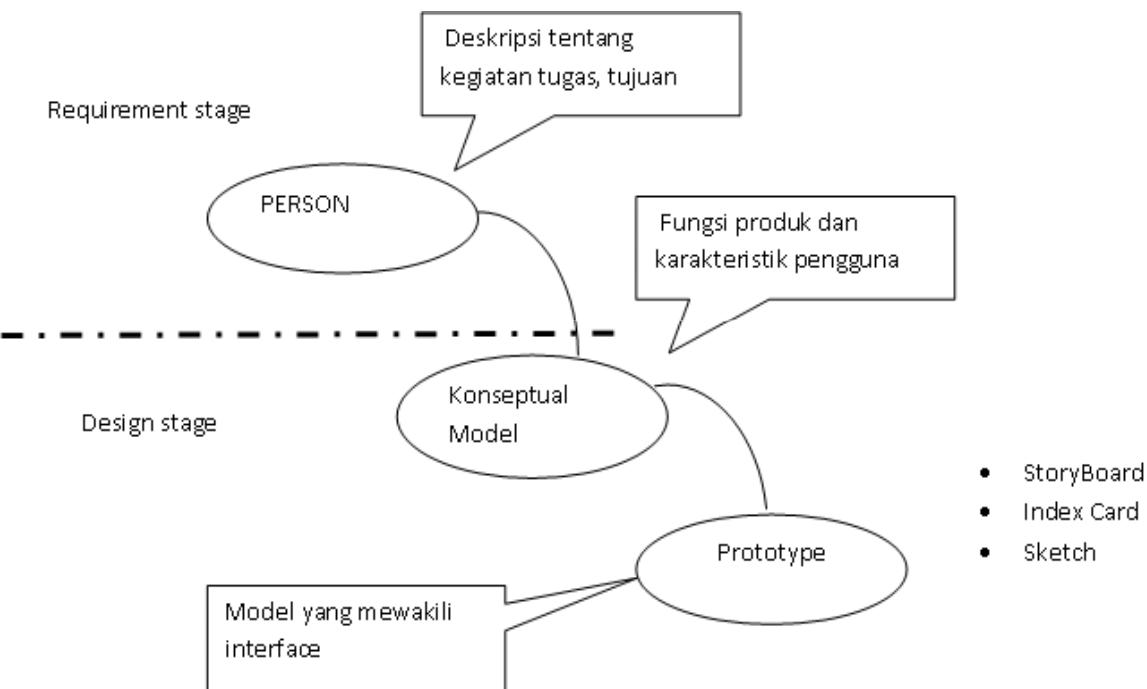
- Kombinasikan pengetahuan baru dan lama.
- Contoh: belajar matematika di kelas, diganti dengan belajar matematika gunakan Permainan

### 2. Interaction types

- Instructing: ketik perintah, menu, button, tekan
- Conversing: dialog dengan sistem
- Manipulating: buka, pindah, tahan, tutup objek
- Exploring: berada di lingkungan 3D virtual

### 3. Interface types

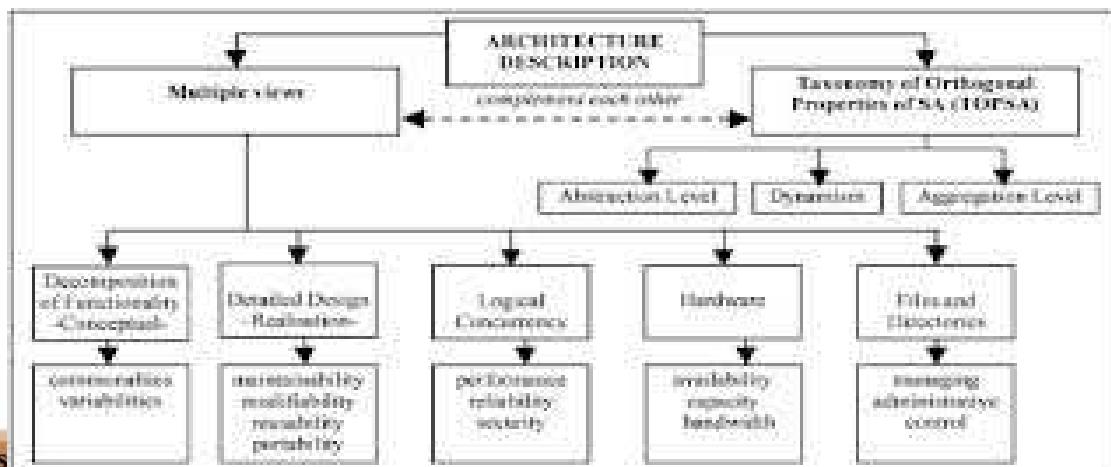
- GUI = antarmuka desktop, windows, icon, menu
- Tangible interface : sensor based interaction
- Shareable: digunakan bersamaan
- Advanced graphical: 3D, avatar



- What functions will the product perform?  
What will the product do and what will the human do (task allocation)?
- How are the functions related to each other?  
sequential or parallel?  
categorisations, e.g. all actions related to telephone memory storage
- What information needs to be available?  
What data is required to perform the task?  
How is this data to be transformed by the system?

## Penggunaan skenario dalam konsep desain

Skenario adalah suatu uraian interaksi manusia dengan mesin. Skenario disini membantu proses desain yang fokus pada keperluan user yang berbeda secara teknis. Skenario yang dihubungkan dengan kasus penggunaan yang dijelaskan dengan tingkat teknis interaksi.



## Guideline for physical desain

Ketidak lengkapan teori yang menjadi dasar perancangan berakibat sulitnya menetapkan standar yang spesifik dan autoritatif. Oleh karena itu mayoritas aturan perancangan bagi sistem interaktif bersifat pemberian saran ( suggestive) dan lebih bersifat umum

Guideline akan cocok pada tahap requirement specification, guideline juga menyediakan menyediakan mekanisme untuk menenterjemahjan spesifikasi detaied design menjadi implementasi aktual

### 3. Desain Phisik secara konkret

Deborah J. Mayhew, memperkenalkan *General Principles Of UI Design*, atau Prinsip Umum Desain User Interface. Ada beberapa hal yang perlu dipahami oleh para perancang sistem, terutama untuk mendapatkan hasil maksimal

#### 1. *User Compatibility*

User Compatibility artinya kesesuaian tampilan dengan typical pengguna. karena berbeda pengguna dapat menyebabkan kebutuhan tampilannya berbeda. misalnya, jika aplikasi diperuntukkan bagi anak-anak, maka jangan menggunakan istilah atau tampilan orang dewasa.

#### 2 . WYSIWYG

What You See Is What You Get, buat tampilan mirip seperti kehidupan nyata dari pengguna. dan pastikan semua fungsi - fungsi yang ada berjalan sesuai tujuan yang diinginkan

#### 3. *Ease of Learning*

aplikasi. mudah dipelajari.

#### 4. *Ease of use*

Aplikasi harus mudah digunakan

## 4. Tools Support

Brand Myers (1995) memberikan 9 saran dalam penyedian software interface tools

1. Tools dapat membantu perancang memberikan rincian dari tugas-tugas atau pekerjaan user
2. Memberikan bantuan perancangan implementasi antar muka secara rinci
3. Memberikan kemudahan untuk menggunakan antar muka
4. Mengijinkan perancang untuk mempercepat analisa desain yang berbeda
5. Mengijinkan bukan programmer untuk meracang dan mengimplementasikan antar muka pemakai
6. Secara otomatis dapat mengevaluasi dan mengusulkan perbaikan
7. Mengijinkan pemakai akhir untuk menyesuaikan antar muka
8. Menyediakan portabilitas
9. Mudah untuk digunakan

Dukungan peralatan dalam sistem interaksi memberikan gambaran (abstraksi) untuk memisahkan perangkat input dan output secara fisik, dimana alat bantu ini memberikan dukungan programmer untuk membuat bagaimana user menggunakannya, ada beberapa alat bantu antar muka pengguna grafis seperti :

1. Widget Toolkit
2. Scrollbars
3. Pull-down
4. Popup menu

## Widget Toolkit

Alat bantu antarmuka pengguna yang mengkombinasikan antar muka objek dan kelakukan manajemen dengan berbasis object oriented

□ Berdasarkan platform, toolkit terdiri dari :

1. X-Windows : X-Toolkit & X-Motif
2. Macintosh : Mac\_Toolbox/Carbon, MacApp, Cocoa
3. Windows : Microsoft Foundation Classes, Windows Forms
4. Java : Swings, abstract Window Toolkit (AWT)

## Berdasarkan Bahasa, Widget Toolkit

1. XML, AJAX, SVG : Jquery, scrips, aculo, us, Dojo Toolkit, Yahoo!, UI Library, Google Web Toolkit, XAML
2. Java : AWT, SWT, Awing Qt Jambi
3. C/C++ : FOX Toolkit, GTK, Qt, Wt Tk, WxWidgets, Xforms
4. Object Pascal : VCL, CLX IP Pascal, Lazarus, fb
5. Python : Pyjamas, PyQt, wxPython, PyGUI, PySide
6. Objective C : GNUStep

## Pertemuan 13

Teknik dan Perangkat bantu pengembangan sistem interaksi

### Pokok Bahasan

1. Spesifikasi metode
2. Interface Building tools
3. Evaluasi dan kritik terhadap tools

# 1. Spesifikasi Methode

## Spesifikasi Metode

- Multiparty Grammar
- Unified Modeling Language (UML)
- User Action Notation (UAN)

- Aset pertama dalam membuat perancangan adalah notasi yang baik untuk merekam dan mendiskusikan kemungkinan-kemungkinan.
  - **Spesifikasi dalam bahasa alami:** cenderung bertele-tele, samar-samar, dan membingungkan.
  - **Bahasa formal dan semiformal:** efektif untuk bahasa perintah.
  - **Menu-tree structures:** menunjukkan tata letak menu. Tidak menunjukkan seluruh aksi yang mungkin.

- **UML class diagram:** hubungan antar-elemen sistem. Lebih baik daripada menu-tree.
- **UML Statechart diagram:** sesuai untuk sistem interaktif.
- **User action notation (UAN):** notasi pendekatan untuk pengembangan sistem manipulasi langsung.

## 1. Multiparty Grammar

- Penggambaran interaksi dengan notasi seperti BNF (Backus-Naur Form).
- Contoh BNF:

```
<Telephone book entry> ::= <Name> <Telephone number>
<Name> ::= <Last name>, <First name>
<Last name> ::= <string>
<First name> ::= <string>
<string> ::= <character>|<character> <string>
<character> ::= A|B|C|...|W|X|Y|Z
<Telephone number> ::= (<area code><exchange> - <local number>)
<area code> ::= <digit><digit><digit>
<exchange> ::= <digit><digit><digit>
<local number> ::= <digit><digit><digit><digit>
<digit> ::= 0|1|2|3|4|5|6|7|8|9
```

- Pada *multiparty grammar* ada nonterminal yang diberi label untuk menyatakan pihak yang menghasilkan string (U: user; C: computer).
- Contoh: Proses log-in

```
<Session> ::= <U: Opening> <C: Responding>
<U: Opening> ::= LOGIN <U: Name>
<U: Name> ::= <U: string>
<C: Responding> ::= HELLO [<U: Name>]
```

- *Multiparty grammar* efektif untuk rentetan perintah berorientasi teks yang berulang-ulang dipertukarkan, seperti pada terminal bank.

## 2. UML

Menurut Jaob Nielsen (2000) dalam Sistem Interaksi ada delapan aturan yang dapat digunakan dalam perancangan antarmuka pemakai yaitu :

1. Berusaha untuk konsisten
2. Meningkatkan frequent user menggunakan shorcut
3. Memberikan feedback yang informatif
4. Merancang dialog penutup
5. Memberikan prementif dalam kesalahan dan penanganannya
6. Memberikan pembalikan aksi yang mudah
7. Internal focus of control
8. Mengurangi beban ingatan jangka pendek.

UML class diagram menggambarkan hubungan antar elemen – elemen sistem lebih baik dari pada menu –tree atau HIPO

UML Staechart/activity diagram di rancang sesuai dengan sistem interaktif

User action notation (UAN) adalah penggunaan notasi pendekatan untuk pengembangan sistem manipulasi langsung, digunakan untuk mengatasi keanekaragaman dunia manipulasi langsung

Simbol yang digunakan sebagai berikut :

~[icon]	: bergerak menuji icon	icon-!	: icon kembali normal
~[x,y]	: menuju koordinat (x,y)	icon!	: icon berkedip
Mv	: tomol mouse ditekan	icon>~	: icon bergerak mengikuti ursor
M^	: tombol mouse dilepas	*	: dapat berulang ( $\geq 0$ kali)
Icon!	: icon di highlight		

## 2. Interface-Building Tools

- Interface-building tools disebut juga:
  - Rapid Prototyper
  - User Interface Builder
  - User Interface Management System
  - User Interface Development Environment
  - Rapid Application Developer

- Fitur interface-building tools:
  - Kebebasan antarmuka pemakai
  - Metodologi dan notasi
  - Rapid prototyping
  - Dukungan perangkat lunak

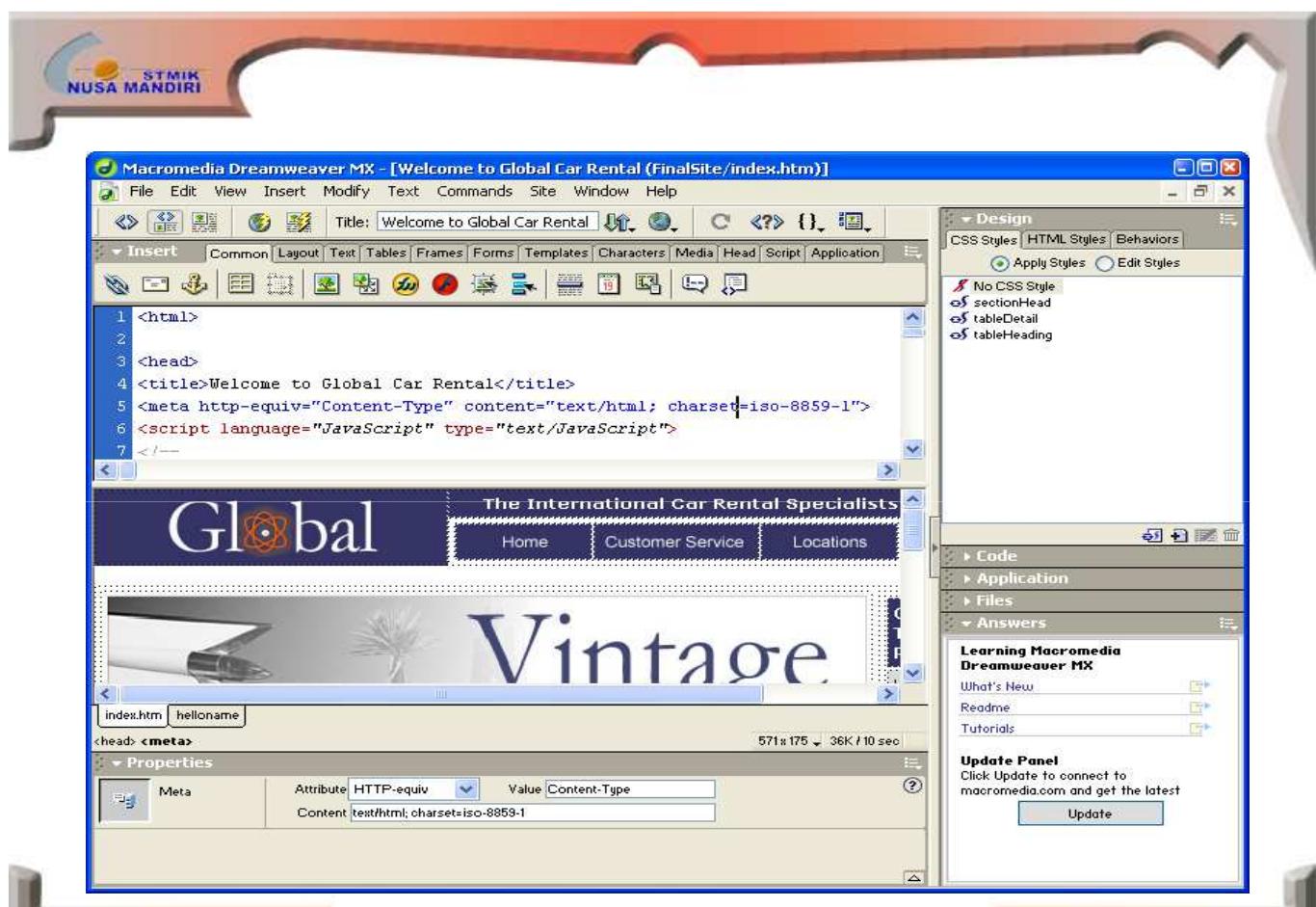
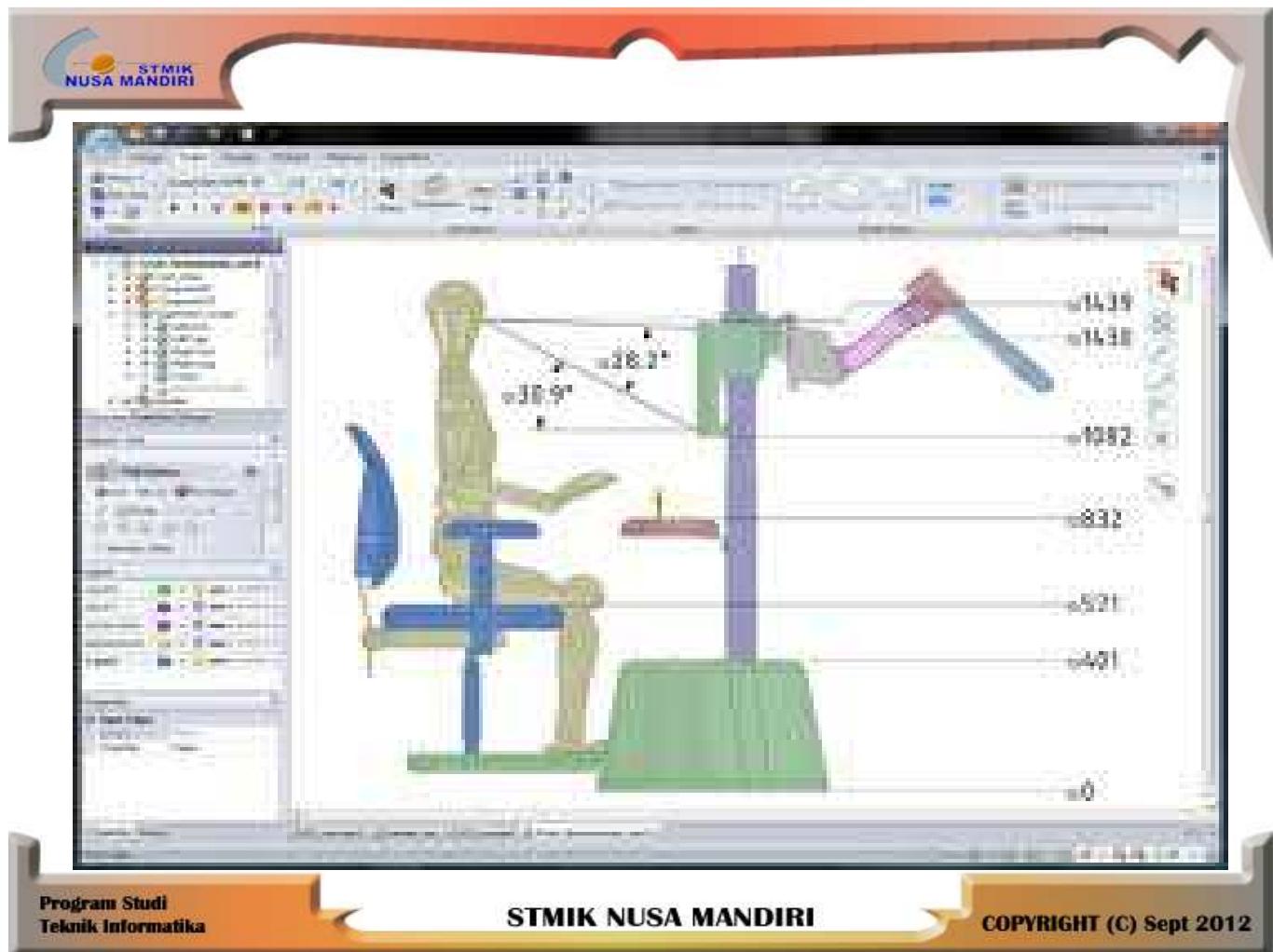
- **Kebebasan antarmuka pemakai**
  - Memisahkan perancangan antarmuka dari program internal.
  - Memungkinkan strategi *multiple user-interface*.
  - Memungkinkan dukungan *multi-platform*.
  - Memberi peranan arsitek antarmuka pemakai.
  - Menegakkan standar.

- **Metodologi dan notasi**
  - Mengembangkan prosedur perancangan.
  - Menemukan cara berbicara tentang perancangan.
  - Melakukan manajemen proyek.
- **Rapid prototyping**
  - Mencoba gagasan dengan sangat dini.
  - Uji, revisi, uji, revisi, ....
  - Mengikutsertakan end users, manajer, pelanggan.

- **Dukungan perangkat lunak**
  - Meningkatkan produktivitas.
  - Memberikan pemeriksaan kendala dan konsistensi.
  - Memfasilitasi pendekatan tim.
  - Memudahkan pemeliharaan.

- Design tools
- Software engineering tools
- Membuat gambaran awal dengan cepat penting di tahapan awal perancangan untuk:
  - Menjajaki berbagai alternatif;
  - Memungkinkan komunikasi dalam tim perancang;
  - Menyampaikan kepada klien seperti apa bentuk produk nantinya.

- Beberapa contoh design tools:
  - **CAI software:** Macromedia Authorware, IconAuthor, Quest.
  - **Multimedia construction tools:** HyperCard, Macromedia Director, Macromedia Flash.
  - **Slide presentation software:** Microsoft PowerPoint.
  - **Visual programming tools:** Microsoft Visual Basic (sekarang dalam Visual Studio .NET), Borland Delphi.
  - **Web design tools:** Macromedia Dreamweaver, Macromedia Fireworks.



# Software Tools

- **Java:**

```
class Test
{
    public static void main(String[] args)
    {
        for(int i = 0; i < args.length; i++)
            System.out.print(i == 0 ? args[i] :
                " " + args[i]);
        System.out.println();
    }
}
```

- **JavaScript**

```
<script language="JavaScript"
        type="text/javascript">
<!--
function square(i)
{
    return i * i;
}

document.write('The function returned:' +
    square(5) + '.');
//-->
</script>
```

## Pertemuan 14

# Evaluasi Perancangan Sistem Interaksi

### Pokok Bahasan

1. What, why and when to evaluation
2. Testing and modeling users

1. What, why and when to evaluation

Penting nya evaluasi :

Pemakai ingin sistem – sistem yang mudah untuk di pelajari dan digunakan dan juga efektif, efisien, aman dan menyenangkan

## A. Untuk apa evaluasi

Ada bermacam alasan yang melatar belakangi dilakukannya evaluasi :

1. Melihat seberapa jauh sistem berfungsi, perancang sistem harus dapat memudahkan pengguna dalam melakukan tugas
2. Melihat efek interface bagi pengguna yang mencakup aspek kemudahan sistem di pelajari, useability dan perilaku pengguna
3. Mengidentifikasi permasalahan yang terjadi pada perancangan, hal ini berhubungan dengan usability dan fungsionalitas dari perancangan
4. Untuk menyakinkan bahwa produk yang dibuat sudah sesuai dengan permintaan pengguna
5. Untuk melihat hasil rancangan yang sesuai dengan permintaan pengguna
6. Saat perancangan sudah menggunakan konteks yang diinginkan pengguna akantetapi hasilnya tidak sesuai
7. Terjadinya kekacauan diantara pengguna

## B. Mengapa di perlukan evaluasi

Ada tiga alasan yang mendasari mengapa proses evaluasi diperlukan untuk produk yang dirancang :

1. Melihat apakah produk yang dirancang sudah berfungsi secara maksimal, yang menungkinkan pengguna melakukan tugas dengan lebih mudah
2. Melihat efek dan interface yang di rancang untuk pengguna apakah mudah dipelajari mengukur tingkat usabiliy serta perilaku dari pengguna
3. Untuk mengidentifikasi probelем khusu yang terjadi pada sistem yang dirancang

## C. Kapan Evaluasi dilakukan

Kita melakukan evaluasi ketika :

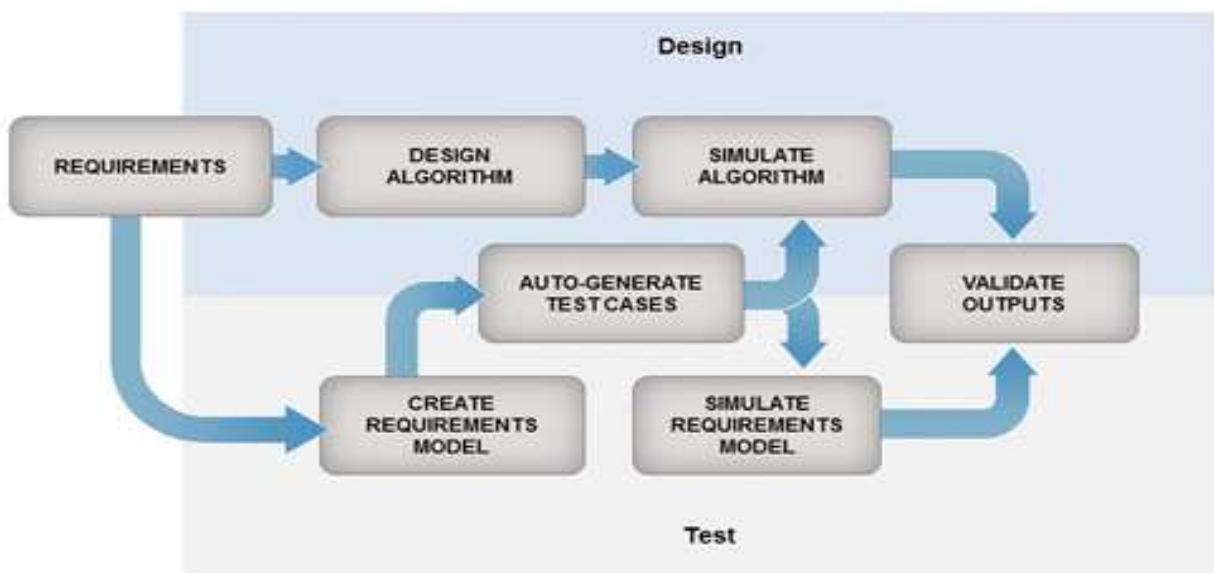
1. Membuat produk baru ( new product)
2. Perbaikan produk ( treatment produk)

## D. Bagaimana test dan evaluasi sistem dilakukan

Memilih metode evaluasi

Faktor yang membedakan teknik evaluasi :

- Tingkat siklus
- Jenis evaluasi
- Tingkat objektivitas dan subyektivitas
- Jenis ukuran yang tersedia
- Kesiapan dari suatu respon
- Tingkat gangguan yang tidak secara langsung
- Sumber yang tersedia



Generate tests from requirements using model-based testing.  
<http://www.mathworks.com/discovery/model-based-testing.html>

### Jenis – jenis Evaluasi :

- Dalam kondisi percobaan (Laboratory)
- Dalam kondisi lokal kerja sebenarnya
- Participation design

#### A. Pada kondisi Percobaan (Laboratory)

Penggunaan pengujian sistem ini dilakukan pada laboratorium dengan beberapa kondisi antara lain :

1. Laboratorium harus dilengkapi dengan fasilitas audio/visual, komputer yang baik dan lengkap
2. Operator bebas dari gangguan yang menghambat pekerjaan
3. Situasi pada laboratorium tidak dapat menggambarkan situasi ruang kerja sebenarnya

#### B. Pada kondisi lokasi kerja sebenarnya :

Penggunaan pengujian sistem dilakukan pada lokasi kerja yang sebenarnya yang memiliki kondisi sebagai berikut

- Tingkat gangguan yang melebihi batas
- Situasi yang terbuka antara pengguna dengan sistem
- Observasi pada lokasi kerja sebenarnya lebih baik dilakukan

#### C. Participatory Design

1. Adalah suatu pemikiran yang melibatkan keseluruhan alur perancangan dan tidak hanya proses evaluasi saja.
2. Perancangan ini dilakukan pada ruang kerja yang melibatkan pengguna yang tidak hanya digunakan sebagai subyek percobaan tetapi juga sebagai anggota yang aktif dalam team perancangan.

## 2. Testing and Modeling user

### Model GOMS

Istilah ini GOMS singkatan dari goals, operators, methods and selection rules

Model GOMS dikembangkan pada awal tahun '80-an Stu Card, Tom Moran dan Alan Newell. GOMS adalah upaya untuk model pengetahuan dan proses kognitif yang terlibat saat pengguna berinteraksi dengan sistem.

1. **Goals**, mengacu pada apa yang ingin dicapai pengguna (misalnya, menemukan sebuah situs web pada desain interaksi).

2. **Operators**, mengacu pada proses kognitif dan tindakan fisik yang perlu dilakukan untuk mencapai tujuan-tujuan (misalnya, pada mesin pencari yang menggunakan, kata kunci di mesin pencari). Perbedaan antara tujuan dan operator adalah bahwa tujuan diperoleh dan operator dijalankan.

3. **Methods**, prosedur untuk mencapai tujuan. yang terdiri dari urutan yang tepat dari langkah yang diperlukan (misalnya, tarik mouse atas lapangan masuk, jenis dalam kata kunci, tekan "Go" tombol).

4. **Selection rules** digunakan untuk menentukan metode yang dipilih bila tersedia lebih dari satu untuk tahap tugas tertentu. Sebagai contoh, sekali kata kunci dimasukkan ke dalam mesin, banyak mesin pencari yang menyediakan pengguna untuk melakukan tugas seperti tekan tombol Enter pada keyboard atau klik "Go" menggunakan tombol mouse. Aturan seleksi akan menentukan mana dari kedua metode dapat digunakan

### 3. Melihat ke masa depan

User Interface masa depan akan berkembang ke arah peningkatan unsur animasi objek, audio, unsur tiga dimensi dan user interface modern dalam realitas maya. Perkembangan tersebut berupa integrasi user interface ke arah object oriented daripada pengolahan instruksi dan fungsi.

Dari segi lainnya, yaitu dialog, sudah tidak lagi terbatas pada bentuk arah. Interaksi yang akan digunakan lebih ke arah natural language dan membenarkan input yang tidak berformat. User interface masa depan memungkinkan komputer melakukan analisis terhadap aktivitas user. Gaya interaksi pada user interfase masa depan ini sering ditampilkan dalam film-film fiksi ilmiah.

User interface masa depan memungkinkan komputer melakukan analisis terhadap aktivitas user. Gaya interaksi pada user interfase masa depan ini sering ditampilkan dalam film-film fiksi ilmiah.

## 5 dimensi interaksi untuk masa depan

1. **Embodied interaction**, yaitu interaksi yang menggunakan gerakan tubuh untuk memberikan input ke komputer, seperti gerakan tangan atau lainnya. Interaksi ini digambarkan dalam film Minority Report yang menggunakan gerakan tangan untuk memasukkan, melihat dan mengubah tampilan dan sama sekali tidak perlu menggunakan mouse atau keyboard.



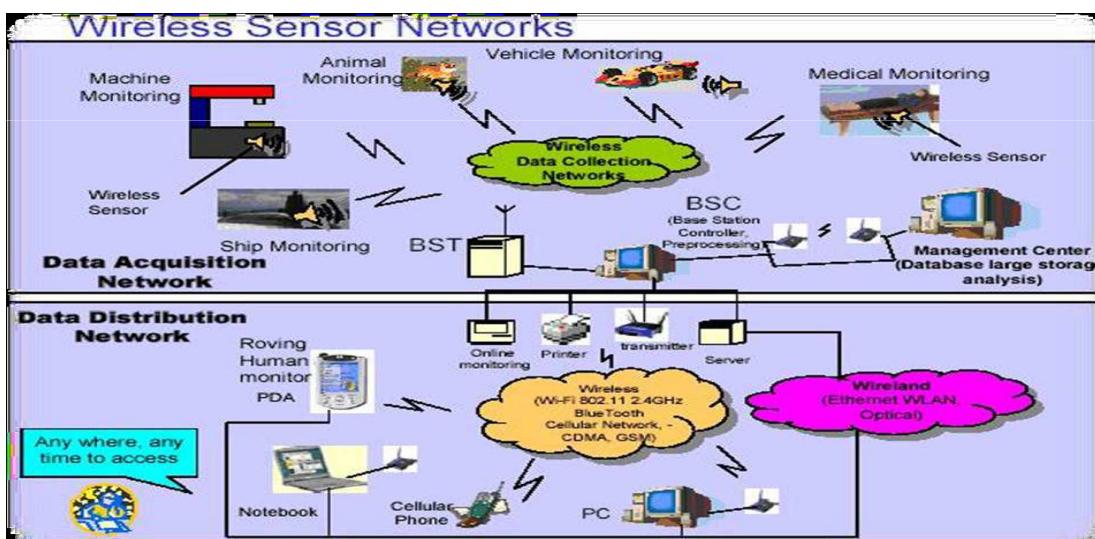
2. **3D virtual Reality**, yaitu interaksi yang memberikan suatu bentuk seolah-olah user ada di dalam komputer, atau perwujudan interaksi dunia nyata dibentuk ke dalam dunia maya. Bentuk interaksi ini telah digunakan pada game 3D, seperti game The Sim2.



3. **Sensing Affect**, yaitu perkembangan interaksi manusia dan mesin, dalam hal ini tidak harus komputer. Pada abad 20-an, komputer telah bisa mengenal dan merasakan pengaruh dari lingkungan. Meraskan suhu, mendengar suara lingkungan, melihat lingkungan dan sebagainya. Sistem sensing affect sangat membantu manusia dalam melakukan aktifitas sehari-hari. Sistem ini biasanya digunakan pada wearable computer dan aksesoris-aksesorisnya.



**4. Sensor Network**, suatu sensor yang ditempatkan pada tempat yang jauh dari komputer, bisa mendeteksi vision dan sound yang ada di lingkungan mana sensor itu ditempatkan. Ini merupakan aspek dari perkembangan teknologi jaringan komputer sistem wireless. Peralatan yang menggunakan sensor network sering digunakan untuk aktivitas mata-mata.



**5. Cyborg**, evolusi perkembangan interaksi manusia dengan komputer yang merupakan aspek dari artificial intelligent. Perkembangan interaksi manusia dengan komputer semakin meluas, di mana komputer mini yang dimasukkan ke dalam suatu robot (cyborg) dapat berfungsi sebagaimana mestinya



## PERTEMUAN 1

# Membuat Basis Data dan Normalisasi

### Konsep dasar Basis Data

Basis Data (*Database*) dapat dibayangkan sebagai sebuah lemari arsip. Jika kita memiliki sebuah lemari arsip dan berwenang untuk mengelolanya. Atau kumpulan informasi yang terorganisasi dan disajikan untuk tujuan khusus. Prinsip utama basis data adalah pengaturan data atau arsip. Sedangkan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan data atau arsip. Sedangkan sistem basis data merupakan perpaduan antara basis data dan sistem manajemen basis data (DBMS). DBMS (*Database Management System*) adalah *software* yang menangani semua akses ke basis data. Contoh dari DBMS yaitu Microsoft Access, MySQL, Oracle, Server 2000, Interbase, Paradox, dan Lain-Lain.

## 1. Entitas

Entitas digunakan untuk menerapkan integritas pada tingkat Entity (Tabel), agar setiap Instances (Record/Baris) pada suatu Entity bersifat *unique* yang disebut sebagai *Primary Key* sehingga dapat dibedakan antara yang satu dengan yang lain.

### Contoh :

- Semua pelanggan, atau pelanggan saja dengan entitas Adi, Ryan, Endah dan seterusnya.
- Semua Mobil atau mobil apa saja dengan entitas mobil Kijang, Starlett dan lain-lain.

## 2. Atribut

Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik dari entitas tersebut. Penentuan atau pemilihan atribut-atribut yang relevan bagi sebuah entitas merupakan hal penting lainnya dalam pembentukan model data. Penetapan atribut bagi sebuah entitas umumnya memang didasarkan pada fakta yang ada. Istilah atribut sebenarnya identik dengan pemakaian kolom data.

### Contoh:

- Entitas pelanggan: kd\_pelanggan, nm\_pelanggan, alamat dan notelp

Normalisasi adalah proses penyusunan tabel-tabel yang redundan (doubel), yang dapat menyebabkan anomali pada saat terjadi operasi manipulasi data seperti tambah, ubah, dan hapus. Anomali yaitu proses basis data yang memberikan efek samping yang tidak diharapkan (misalnya menyebabkan ketidakkonsistenan data atau membuat sesuatu data menjadi hilang ketika data lain dihapus).

### Bentuk-bentuk Normalisasi :

#### 1. Bentuk Tidak Normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi, data dikumpulkan apa adanya sesuai keadaannya. Data didapat dari bentuk dokumen yang ada.

#### 2. Bentuk Normal Pertama (1NF)

Sebuah tabel disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut
- Masing-masing cell bernilai tunggal

#### 3. Bentuk Normal Kedua (2NF)

Bentuk Normal Kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

#### 4. Bentuk Normal Ketiga (3NF)

Bentuk Normal Ketiga (3NF) terpenuhi jika pada sebuah tabel, atribut bukan kunci tidak memiliki ketergantungan transitif terhadap kunci utama dan menghilangkan anomali-anomali hasil ketergantungan fungsional.

## Faktur Pembelian Barang

PT.XYZ

Jl.Bekasi Timur 24

Kode supplier :A01

tanggal:9 /09/2009

NamaSupplier :bbb

Nomor:002

Kode	Nama barang	Qty	Harga	jml
T01	TV 20 inchi	10	1200000	12.000.000
T02	TV 29 inchi	5	2400000	120.000.000

Total Faktur 132.000.000

Jatuh tempo 12/09/2009

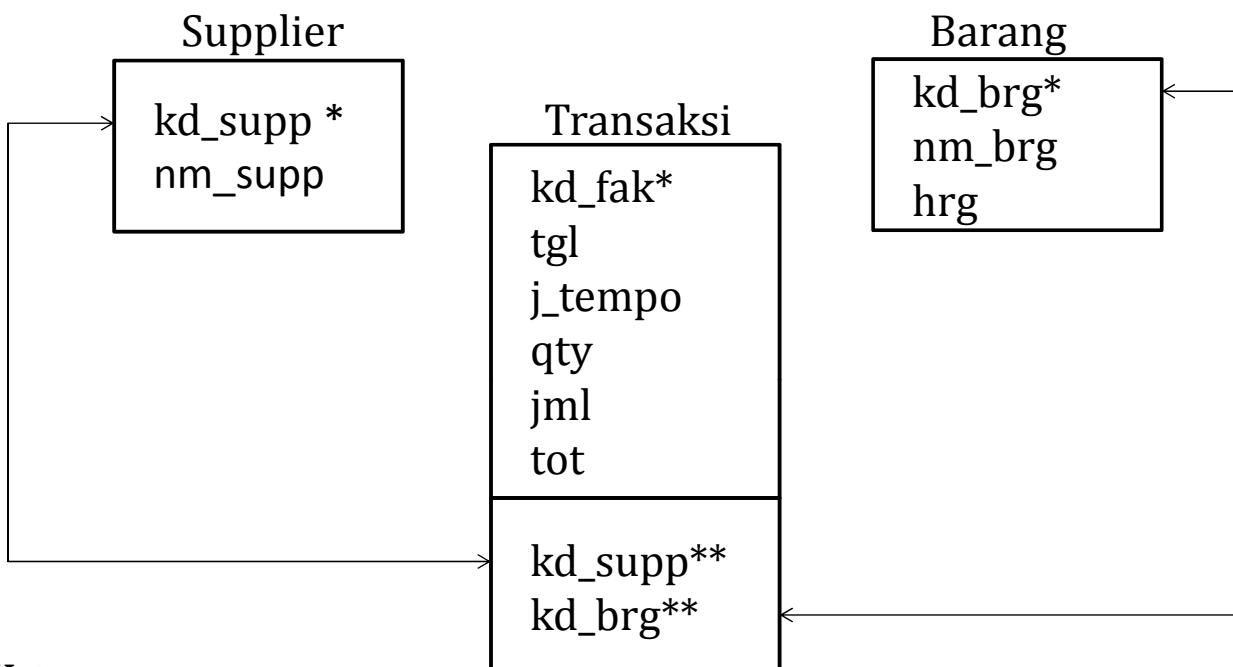
## Bentuk Tidak Normal (UNF) Bentuk Normal Pertama (1NF)

kd\_fak  
kd\_supp  
nm\_supp  
kd\_brg  
nm\_brg  
tgl  
j\_tempo  
qty  
hrg  
jml  
tot

kd\_fak\*  
tgl  
j\_tempo  
qty  
jml  
tot  
kd\_supp\*  
nm\_supp  
kd\_brg\*  
nm\_brg  
hrg

Ket : \* adalah Candidate Key

## Bentuk Normal Kedua (2NF)



**Keterangan:**

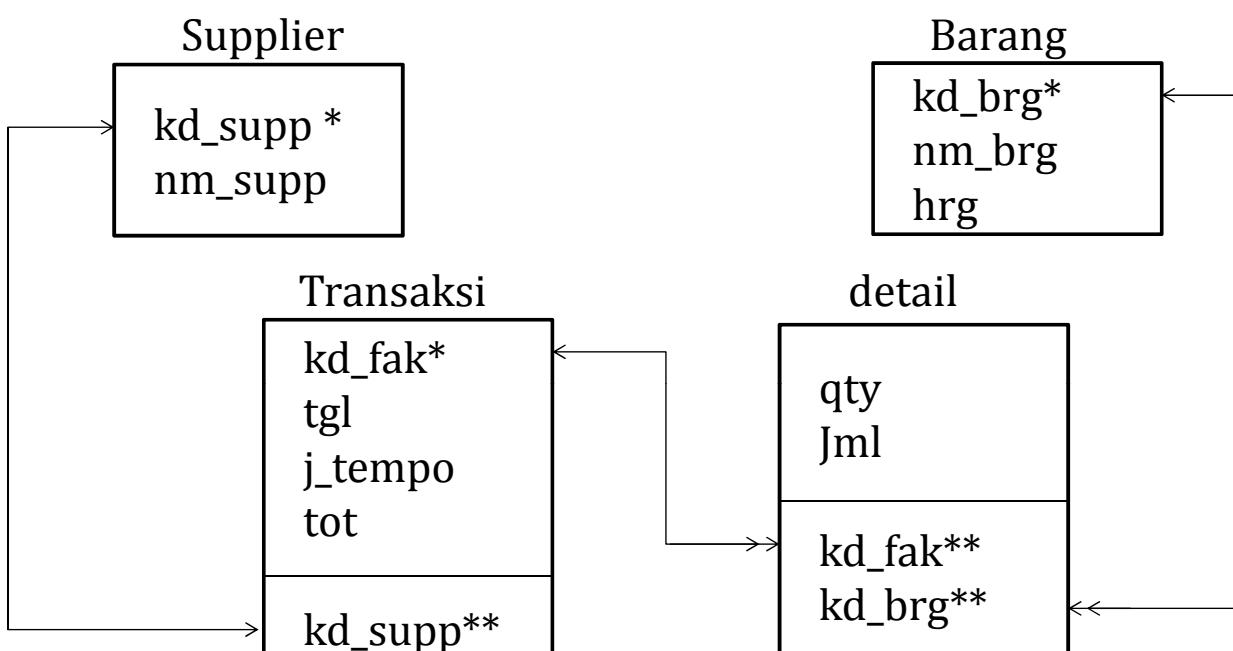
\* : Primary Key

←→ : One to One Relationships

\*\* : Foreign Key

←→→ : One to Many Relationships

## Bentuk Normal Ketiga (3NF)



**Keterangan:**

\* : Primary Key

←→ : One to One Relationships

\*\* : Foreign Key

←→→ : One to Many Relationships

## PERTEMUAN 2

# Database, Tabel dan Relasi



## Microsoft Access 2010

*Microsoft Access* adalah sebuah program aplikasi basis data dengan model relasional. Aplikasi ini menggunakan mesin basis data *Microsoft Jet Database Engine*, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi yang akan kita gunakan adalah *Microsoft Office Access 2010*.

*Microsoft Access* dapat menggunakan data yang disimpan di dalam format *Microsoft Access*, *Microsoft Jet Database Engine*, *Microsoft SQL Server*, *Oracle Database*, atau semua *container* basis data yang mendukung standar *ODBC*.

### Susunan / Hierarki pada Database

#### 1. Character

Bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus yang membentuk suatu item data (*field*).

#### 2. Field

Mempresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat, dan lain-lain.

### 3. Record

Kumpulan dari *field* membentuk *record*. *Record* menggambarkan suatu unit data individu yang tertentu. Kumpulan dari *record* membentuk suatu file. Misalnya *file personalia*, tiap-tiap *record* dapat mewakili data tiap-tiap karyawan.

### 4. File

File terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya *file* mata pelajaran, berisi data tentang semua mata pelajaran yang ada.

### 5. Database

Kumpulan dari *file/tabel* membentuk suatu database.

## Membuat Basis Data di Microsoft Access 2010

### 1. Mengaktifkan Ms. Access 2010

Klik menu *Start* → *All Program* → *Microsoft Office* → *Microsoft Access 2010*.

### 2. Membuat Database

Klik **Blank Database** maka akan tampil jendela *Blank Database* → Isi *File Name* dengan nama database **koperasi.accdb** →



Klik **icon folder** untuk memilih atau mengubah lokasi penyimpanan → lalu Klik **Create** untuk membuat database.

### 3. Membuat tabel

- Klik Menu **Create** → Pilih **Table Design**
- buatlah struktur tabel seperti dibawah ini:

<b>Field Name</b>	<b>Type</b>	<b>Size</b>	<b>Keterangan</b>
Kdbarang	<i>Text</i>	4	<i>Primary Key</i>
Nmbarang	<i>Text</i>	30	
Satuan	<i>Text</i>	6	<i>Combo Box</i>
Harga	<i>Currency</i>	-	

#### 1) Pembuatan *ComboBox*:

Klik field **Satuan** → Klik tab **Lookup** pada **Field Properties** → Lalu ubah settingan menjadi seperti dibawah ini:

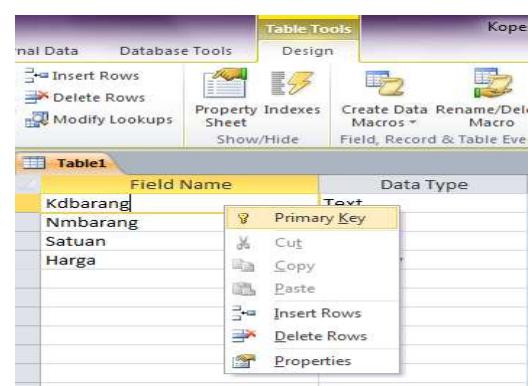
**Display Control** : *ComboBox*

**Row Source** : "Pack" ; "Sachet"

**Row Source Type** : *Value List*

#### 2) Membuat Primary Key

Klik field **Kdbarang** → Klik kanan → Pilih *Primary Key*



c. Menyimpan tabel

Klik Menu **File** → Klik **Save** → Ketikkan nama tabel yang akan disimpan → **Ok**

#### 4. Mengisi Record

*Double* klik nama tabel, lalu isikan data-data seperti dibawah ini:

Barang				
Kdbarang	Nmbarang	Satuan	Harga	Click to Add
B001	Nescafe Cream Isi 30	Pack	Rp20.000	
B002	Coffemix 50 gr	Sachet	Rp1.500	
B003	Kapal Api Spesial Isi 10	Pack	Rp7.500	
B004	Torabika Mocca 50 gr	Sachet	Rp1.200	
B005	Good Day Moccacino Isi 5	Pack	Rp45.000	
*				

### Latihan membuat tabel

Tambahkan 3 buah tabel pada database koperasi.accdb yang telah dibuat sebelumnya sesuai dengan struktur dibawah ini:

1. Nama tabel: **Karyawan** (diisi sebanyak 3 record)

Field Name	Data Type	Size	Keterangan
NIK	Text	9	Primary Key
Nmkaryawan	Text	30	
Bagian	Text	15	Combo Box → Administrasi, Kasir

2. Nama tabel: **Penjualan** (diisi sebanyak 3 record)

Field Name	Data Type	Size	Keterangan
Nofaktur	Text	5	Primary Key
Tglfaktur	Date/Time	8	
NIK	Text	9	Combo Box → Tabel Karyawan

Pembuatan *Combo Box*:

Klik tab *Lookup* pada *Field Properties* → Lalu ubah setingannya seperti dibawah ini:

- Display Control** : Combo Box  
**Row Source Type** : Table/Query  
**Row Source** : Karyawan

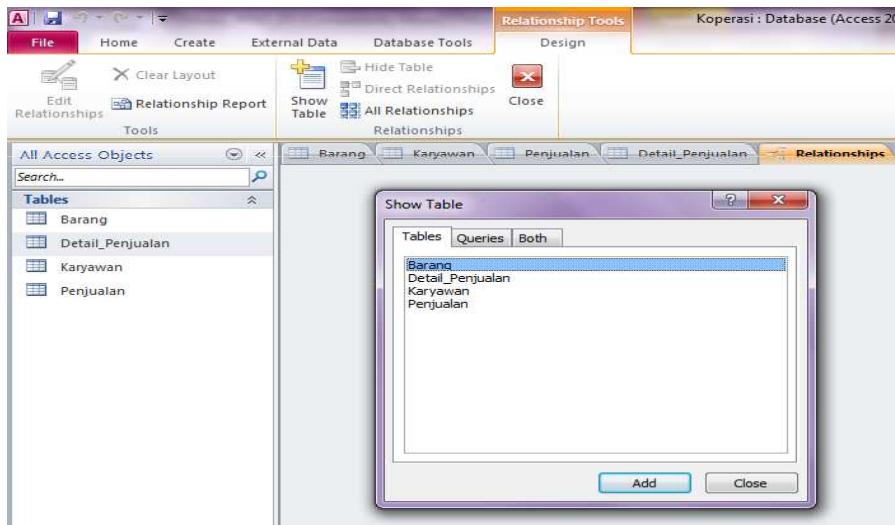
3. Nama tabel: **detail\_penjualan** (diisi sebanyak 3 record)

Field Name	Data Type	Size	Keterangan
Nofaktur	Text	5	Combo Box → Tabel Penjualan
Kdbarang	Text	4	Combo Box → Tabel Barang
Jumbel	Number		

☞ Selamat Mengerjakan ☞

## 5. Membuat Relationship

Untuk membuat *relationship*, buka salah satu tabel dalam keadaan *design* → lalu klik **icon Relationship**, maka akan tampil jendela **Show Table** →

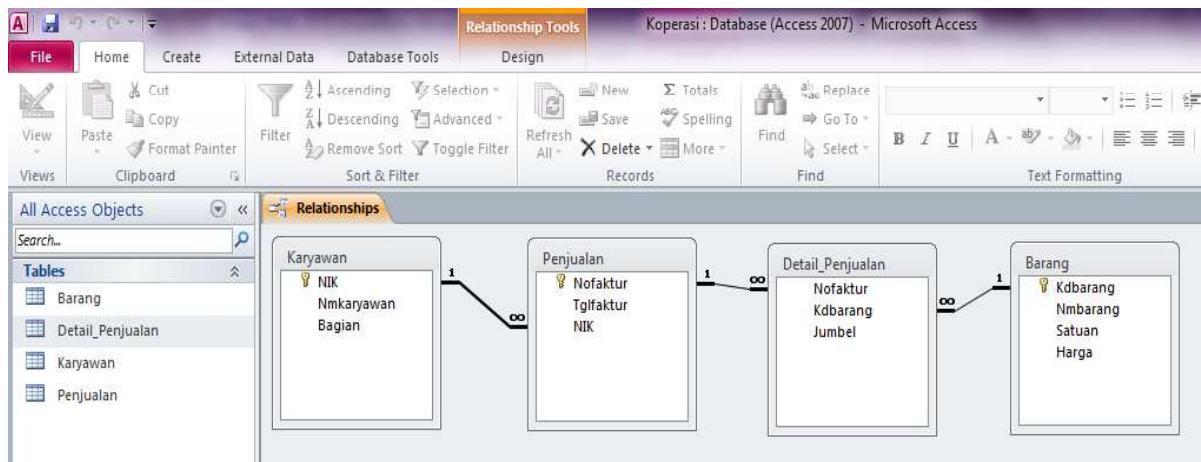


→ Pilih tabel satu persatu → kemudian Klik **Add** dan susun seperti berikut:



Untuk membuat relasi antar tabel, tarik garis diagonal dari *field* kdbarang(*primary key*) dari tabel **Barang** ke *field* kdbarang di tabel **Detail\_Penjualan**, hingga tampil jendela **Edit Relationship** seperti berikut ini:

Lakukan hal yang sama untuk field nik (*primary key*) pada tabel karyawan ke tabel **Penjualan**, dan field Nofaktur (*primary key*) pada tabel **Penjualan** ke tabel **Detail\_Penjualan**, hingga menjadi sebuah relasi seperti berikut:

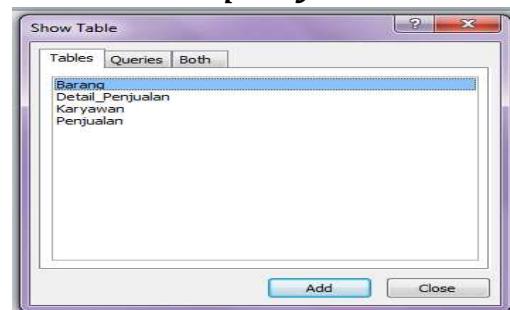


## **PERTEMUAN 3**

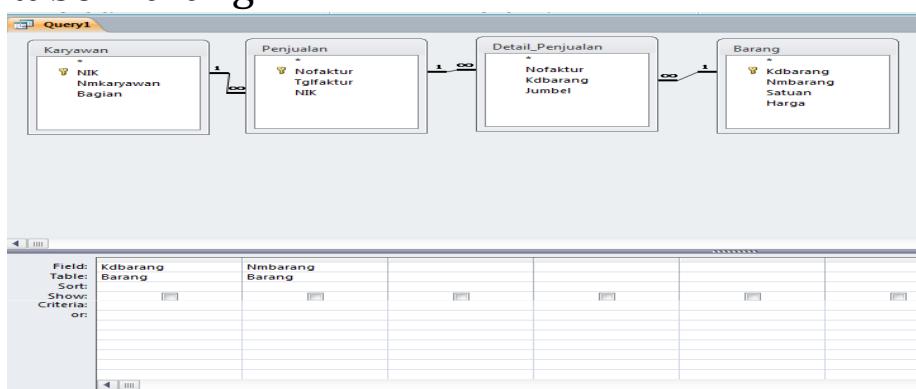
# **Data Definition Query**

Query adalah suatu objek database yang dapat digunakan untuk menampilkan, menyunting atau menganalisa suatu data dengan memberikan baris-baris perintah tertentu. Pada aplikasi ini sudah dilengkapi dengan *Structured Query Language (SQL)* yang menjadi standard bahasa dalam mengolah database. Dengan *SQL* kita dapat mempersingkat penulisan listing program yang panjang karena pada setiap perintahnya dapat mencakup banyak perintah sekaligus. Langkah-langkah membuat *query*:

1. Klik tab **Create**
2. Pilih Icon **Query Design**,  
maka akan muncul **Show Tables**



3. Pilih nama tabel kemudian Klik **Add**, contohnya kita pilih tabel Barang.



## Keterangan

- Field* : Nama *field* yang ingin ditampilkan
- Table* : Nama tabel dari *field* yang ditampilkan
- Sort* : Mengurutkan data hasil *query*
- Show* : Mengatur *field* ditampilkan atau tidak
- Criteria*: Syarat dari data yang ingin ditampilkan

- Jika ingin menggunakan semua *field* pada tabel maka kita hanya klik dan drop tanda \* pada tabel barang ke baris *field*. Jika tidak kita dapat memilih salah satu *field* dengan cara mendouble klik field yang diinginkan.

Terdapat 3 pilihan pada *toolbars View*, yaitu :

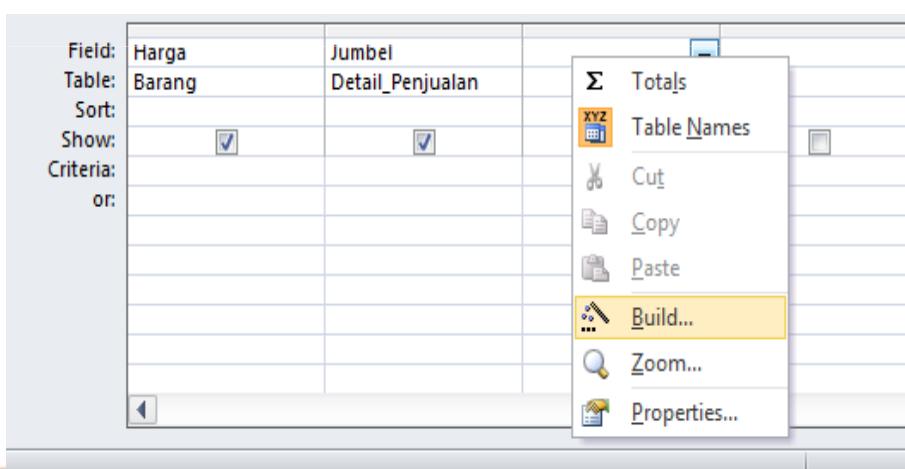
- Design View* : untuk menampilkan desain tabel query
- Datasheet View* : untuk menampilkan data hasil query
- SQL View* : untuk membuat query dengan script SQL

Pada bagian ini, jika kita sudah membuat desain *query* menggunakan *design view*, kita bisa melihat langsung script SQLnya pada *SQL View*.

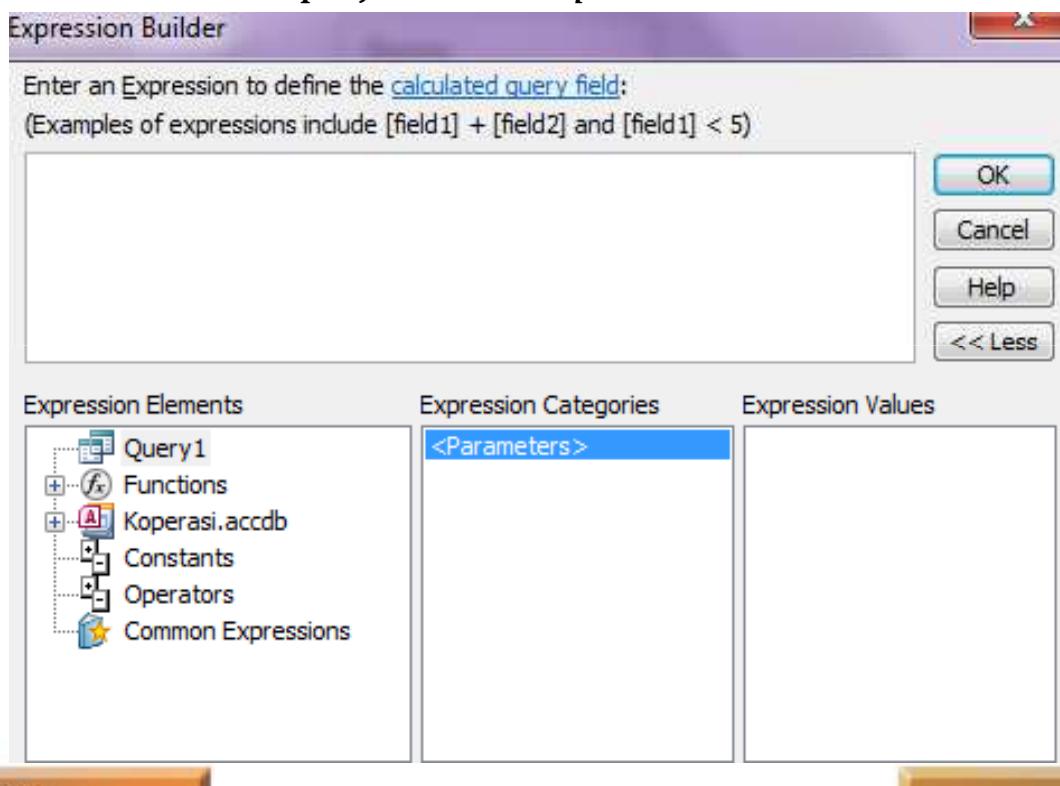
## Membuat Expresi pada query

Pembuatan ekspresi pada *query* dilakukan, apabila ada sebuah penambahan *field* yang bersifat sementara (tidak bergabung dengan tabel) yang berfungsi untuk membantu dalam pembuatan *query* yang lebih kompleks. Cara pembuatan query antara lain:

1. Klik kanan pada *field* → lalu pilih **Build**



## 2. Maka akan tampil jendela *Ekspression*



## 3. Ketikkan perhitungan yang akan dibuat Ekspresi → Ok



4. Setelah itu *running query* yang sudah dibuat dengan cara mengklik icon run !
5. Maka akan tampil sebuah *query* lengkap dengan isi *recordnya*

The screenshot shows the Microsoft Access 2007 interface. The title bar reads "Koperasi : Database (Access 2007) - Microsoft Access". The ribbon menu is visible with tabs like File, Home, Create, External Data, and Database Tools. The Home tab is selected. Below the ribbon, there are various toolbar icons for View, Paste, Cut, Copy, Format Painter, Filter, Sort & Filter, Refresh, Save, Totals, Spelling, Find, and Select. On the left, the navigation pane shows "All Access Objects" with sections for Tables (Barang, Detail\_Penjualan, Karyawan, Penjualan) and Queries (Query\_Penjualan). The main area displays a query named "Query\_Penjualan" with the following data:

Nmbarang	Satuan	Harga	Jumbel	Subtotal
Nescafe Cream Isi 30	Pack	Rp20.000	10	Rp200.000
Coffemix 50 gr	Sachet	Rp1.500	50	Rp75.000
Kapal Api Spesial Isi 10	Pack	Rp7.500	7	Rp52.500
*				

At the bottom, the footer includes "Program Studi Sistem Informasi", "STMIK NUSA MANDIRI", and "COPYRIGHT © Maret 2013".

## Latihan membuat Query

Berdasarkan database **Koperasi.accdb** yang telah dibuat, buatlah **Query\_Penjualan** berdasarkan struktur query dibawah ini:

Field	Asal Tabel
<b>Nofak</b>	Detail_penjualan
<b>Kdbarang</b>	Detail_penjualan
<b>Nmbarang</b>	Barang
<b>Satuan</b>	Barang
<b>Harga</b>	Barang
<b>Jumbel</b>	Detail_Penjualan
<b>Subtotal</b>	Ekspresi
<b>Diskon</b>	Ekspresi
<b>Bonus</b>	Ekspresi

**Keterangan**

- Subtotal = jumbel \* harga
- Diskon 15% , jika subtotal lebih dari Rp. 50.000 .  
Diskon 10%, jika subtotal lebih dari Rp. 30.000.
- Diskon 7%, jika subtotal lebih dari Rp. 20.000, selain itu diskon 5%
- Bonus 1 Pcs Kopi ABC Susu, jika kode barang yang dibeli adalah B002

At the bottom, the footer includes "Program Studi Sistem Informasi", "STMIK NUSA MANDIRI", and "COPYRIGHT © Maret 2013".

## PERTEMUAN 4

# Data Definition Language

### **Data Definition Language (DDL)**

Merupakan suatu sub bahasa SQL yang berfungsi mendefinisikan database dan tabel. Perintah yang digunakan diantaranya : **CREATE, ALTER, dan DROP**. Dengan menggunakan ketiga perintah tersebut kita sudah dapat membangun struktur data dengan lengkap.

1. Perintah **CREATE** : digunakan untuk membuat tabel baru

```
CREATE TABLE nama_tabel (field-1 tipe(panjang),  
field-2 tipe(panjang), field-n tipe(Panjang));
```

```
CREATE TABLE mahasiswa (nim text(8) not null  
primary key,nama text(30) not null, alamat text(50));
```

2. Perintah **ALTER** : digunakan untuk merubah, menambah(ADD), menghapus(DROP) struktur tabel.

**ALTER TABLE** nama\_tabel [spesifikasi perubahan];

**ALTER TABLE** mahasiswa **ADD** jenis\_kel text(1);

Beberapa perintah *alter* yang dapat digunakan antara lain:

- a. Menambahkan *Primary key* pada *field* NIM di tabel mahasiswa.

**ALTER TABLE** mahasiswa **ADD CONSTRAINT** NIM **Primary Key(NIM)**;

- b. Merubah ukuran dan tipe dari *field* NIM pada tabel mahasiswa.

**ALTER TABLE** mahasiswa **ALTER COLUMN** NIM Number;

- c. Menghapus *field* Jenis kelamin dari tabel mahasiswa.

**ALTER TABLE** mahasiswa **DROP COLUMN** jenis\_kel;

3. Perintah **DROP** : digunakan untuk menghapus tabel

**DROP TABLE** nama\_tabel;

**DROP TABLE** mahasiswa;

## Latihan Penggunaan DDL

Anda adalah seorang programmer yang diminta oleh sebuah perusahaan untuk membuat sebuah basis data dengan menggunakan perintah DDL. Berikut adalah struktur basis data yang harus anda kerjakan!!

1. Buatlah sebuah database dengan nama **db\_barang.accdb**.
2. Buatlah sebuah tabel dengan nama **tb\_barang** berdasarkan struktur tabel dibawah ini:

Field	Type	Size	Ket
Kode	Text	5	<i>Primary Key</i>
Nabar	Text	30	
Harga	Currency		
Satuan	Text		

3. Tambahkan sebuah *field* **stok** (number) pada tabel **tb\_barang** .
4. Ubahlah *type* data pada *field* **harga** menjadi **number**.
5. Hapuslah *field* **satuan** pada tabel **tb\_barang**.

☞ Selamat Mengerjakan ☞

## PERTEMUAN 5

# Data Manipulation Language

### Data Manipulation Language (DML)

Merupakan sub bahasa SQL yang berfungsi memanipulasi data yang ada di dalam basisdata atau database. Perintah yang termasuk didalam DML adalah **INSERT, SELECT, UPDATE** dan **DELETE**.

1. Perintah **INSERT** : berfungsi untuk menyisipkan, memasukkan, atau menyimpan data dari luar sistem kedalam database.

```
INSERT INTO nama_tabel VALUES (value-1, value-2,...,value-n);
```

```
INSERT INTO mahasiswa VALUES ("12120002","Risma",  
"28/04/1989","TANGERANG");
```

2. Perintah **SELECT** : berfungsi untuk membentuk tabel baru dengan cara mengcopy (*backup*) seluruh data dari tabel aktif.

```
SELECT      *      INTO      nama_tabel_baru      FROM
nama_tabel_aktif;
```

```
SELECT * INTO mahasiswa_baru FROM mahasiswa;
```

3. Perintah **UPDATE** : berfungsi untuk mengubah satu atau lebih data *field* yang terdapat pada satu atau lebih *record*.

```
UPDATE nama_tabel SET field=value WHERE kriteria;
```

```
UPDATE mahasiswa SET nama="Ahmad" WHERE
NIM="12120002";
```

4. Perintah **DELETE** : berfungsi untuk menghapus satu atau beberapa *record* dari suatu

```
DELETE * FROM nama_tabel WHERE kriteria;
```

```
DELETE * FROM mahasiswa WHERE nim="12120002";
```

Anda adalah seorang programmer yang diminta oleh sebuah perusahaan untuk membuat sebuah basis data dengan menggunakan perintah DDL. Berikut adalah struktur basis data yang harus anda kerjakan!!

1. Buatlah sebuah database dengan nama **db\_susu.accdb**.
2. Buatlah sebuah tabel dengan nama **tb\_susu** berdasarkan struktur tabel dibawah ini:

Field	Type	Size	Keterangan
kd_susu	Text	5	Primary Key
nm_susu	Text	30	
harga	Currency		
ukuran	Text	15	

3. Isilah **tb\_susu** dengan beberapa *record* data seperti dibawah ini:

kd_susu	nm_susu	harga	ukuran
S001	Dancow	35000	Besar
S002	Bendera	38000	Sedang
S003	Milo	23000	Kecil

4. Copy/Backup **tb\_susu** dengan nama **copy\_tb\_susu**.
5. Ubahlah isi *record* pada field **harga** menjadi Rp. 55.000,- untuk kode susu **S001**.
6. Hapuslah sebuah *record* dengan kode susu **S003**.

☞ Selamat Mengerjakan ☞

# PERTEMUAN 6

## Inner Join

### Inner Join

*Inner Join* merupakan operasi yang digunakan untuk mendapatkan data gabungan dari dua tabel atau lebih yang akan menghasilkan data yang berpasangan saja. *Inner Join* dipakai untuk memperoleh data-data yang lebih detail dari tabel-tabel yang saling memiliki hubungan. Operasi *inner join* ini biasanya digunakan dalam perintah **Select**. Contoh penggunaan sintak innerjoin dalam pembuatan query sebagai berikut:

#### Bentuk Dasar

```
SELECT * FROM nama_tabel;
```

## Keterangan:

- a. Tanda \* berarti menampilkan keseluruhan *field* yang berada pada suatu tabel.

**SELECT \* FROM barang;**

- b. Tanda \* dapat diganti dengan **nama\_tabel.nama\_field** : untuk menampilkan *field* tertentu yang berada pada tabel tertentu.

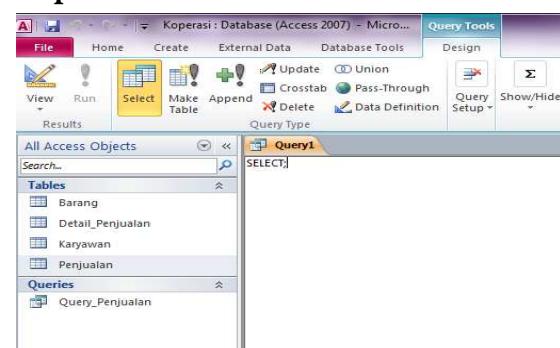
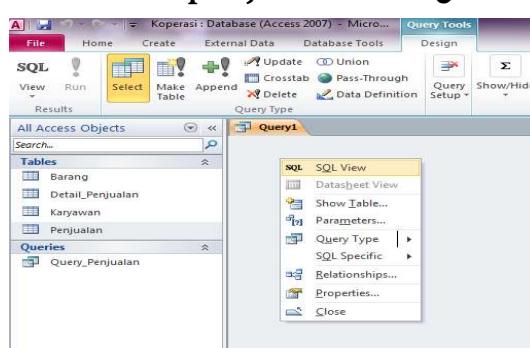
**SELECT Barang.Kdbarang, Barang.Nmbarang,  
Barang.Harga FROM barang;**

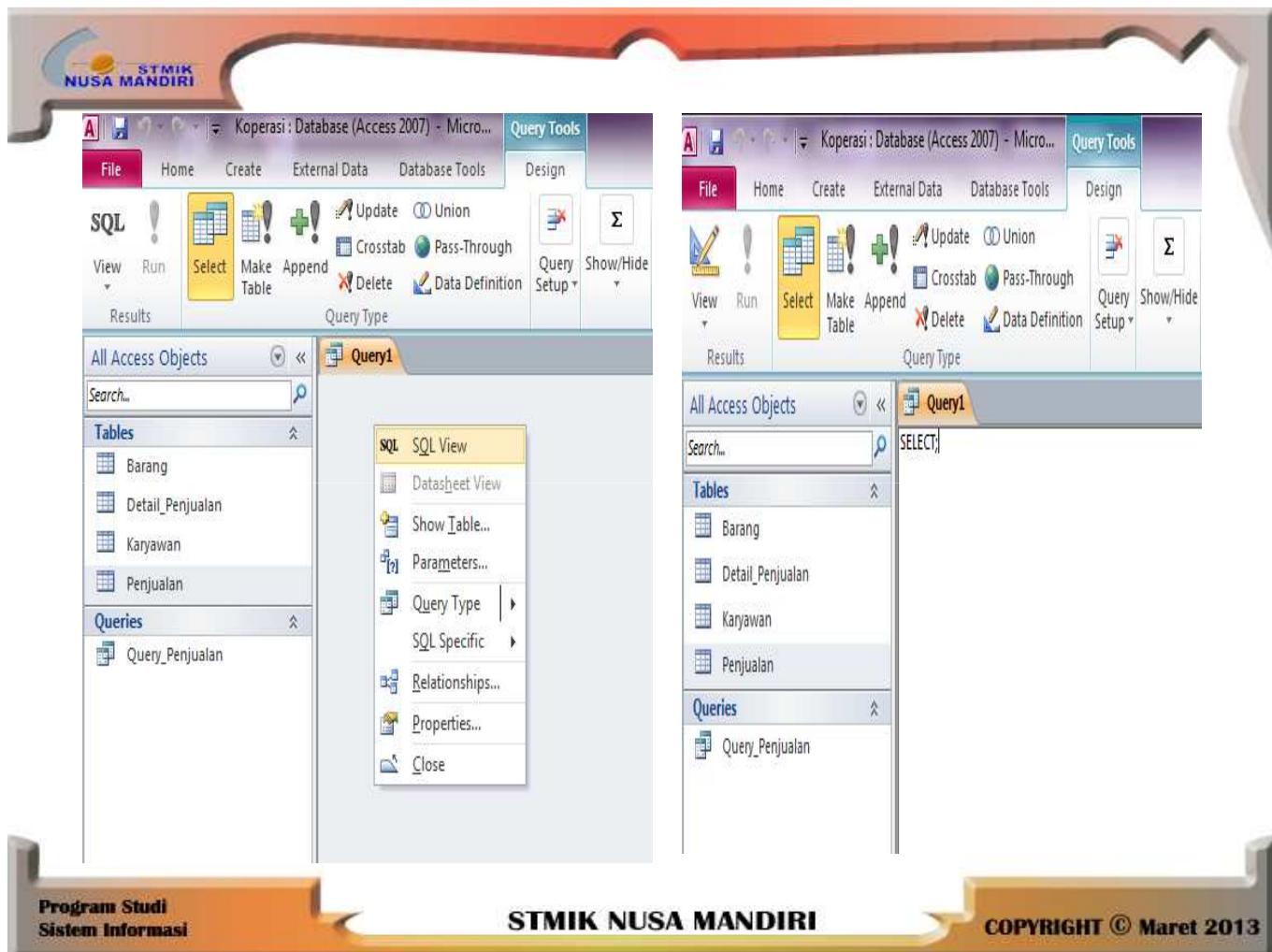
- c. Tanda \* dapat digantikan juga dengan **proses perhitungan as nama\_field** : untuk menampilkan *field* baru yang sebelumnya tidak berada pada tabel.

**SELECT Barang.Kdbarang, Barang.Nmbarang,  
Barang.Harga , 10 as jumbel, harga\*jumbel as subtotal  
FROM barang;**

Didalam Microsoft Access 2010, kita dapat mengaplikasikan *inner join* dengan menggunakan salah *toolbars view* yaitu **SQL View**. Buka kembali database **Koperasi.accdb**.

1. Pilih menu **Create** → Pilih **icon Query Design** → Maka akan muncul jendela **Show tables** → Klik tombol **Close**, maka akan tampil jendela **Query**.
2. Klik kanan pada jendela **query** → Pilih **SQL View**, maka akan tampil jendela **SQL View** seperti dibawah ini:





3. Ketikkan sintaks *inner join* didalam jendela query.

Sumber: Latihan Modul PPBD halaman 14

## **SELECT**

Barang.Nmbarang, Barang.Satuan,  
Barang.Harga, Detail\_Penjualan.Jumbel,

[harga]\*[jumlah] AS Subtotal,

`IIf([Subtotal]>50000,15/100*[subtotal],IIf([Subtotal]>30000,10/100*[subtotal],IIf([Subtotal]>20000,7/100*[subtotal]*5/100*[Subtotal])))) AS Diskon,`

IIf([Detail\_Penjualan].[Kdbarang]="B002","1 Pcs Kopi ABC Susu","Tidak dapat") AS Bonus

**FROM** (Karyawan INNER JOIN Penjualan ON Karyawan.NIK = Penjualan.NIK) INNER JOIN (Barang INNER JOIN Detail\_Penjualan ON Barang.Kdbarang = Detail\_Penjualan.Kdbarang) ON Penjualan.Nofaktur = Detail\_Penjualan.Nofaktur;

4. Klik **icon run**  apabila ingin melihat hasilnya.
5. Klik **File → Save**, apabila ingin menyimpan *query*.

## PERTEMUAN 7

# Latihan Praktikum Perancangan Basis Data

## Inner Join

Sebuah Toko Alat Musik di Jakarta dalam melakukan transaksi penjualan alat musiknya, ingin merubah sistemnya ke komputerisasi. Anda sebagai seorang programer diminta merancang database menggunakan Microsoft Access 2010. Pembuatan tabel dan pengisian recordnya dapat anda kerjakan secara wizard yang tersimpan dalam sebuah database bernama **studiomusik.accdb**. Buatlah tabel seperti struktur dibawah ini:

- a. Nama tabel: **alatmusik** (disi sebanyak 3 record)

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Keterangan</i>
kdalat	Text	3	<i>Primary Key</i>
nmalat	Text	15	
harga	Currency	-	

- b. Nama tabel: **kasir** (disi sebanyak 3 record)

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Keterangan</i>
kdakasir	Text	3	<i>Primary Key</i>
nmkasir	Text	20	
jabatan	Text	10	

- c. Nama tabel: **penjualan** (disi sebanyak 3 record)

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>	<i>Keterangan</i>
nonota	Text	5	Primary Key
tgl	Date	-	
kdkasir	Text	3	
kdalat	Text	3	
jumbel	Number	-	

Field	Asal Tabel
nonota	penjualan
tgl	penjualan
kdkasir	penjualan
nmkasir	kasir
kdalat	penjualan
nmalat	alat
harga	alat
jumbel	penjualan
subtotal	ekspresi
ppn	ekspresi
diskon	ekspresi
total	ekspresi

Buatlah **QueryPenjualanAlat** dengan menggunakan **SQL** dengan struktur *query* sebagai berikut :

### Ketentuan Soal

- Subtotal = harga x jumbel
- PPn = 10% dari harga
- Diskon diberikan 10 % jika jumlah beli lebih dari 5 buah, 7% jika jumlah beli lebih dari 3 buah, selain itu tidak dapat diskon.
- Total = subtotal + PPn - diskon

# PERTEMUAN 9

# MySQL

Mysql adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (Structure Query Language).

MySQL merupakan hasil ciptaan dari **Michael "Monty" Widenius, David Axmark, dan Allan Larson**. Pada tahun 1995 mereka lalu mendirikan perusahaan bernama MySQL AB di Swedia. Tujuan awal diciptakannya MySQL yaitu untuk mengembangkan aplikasi web yang akan digunakan oleh salah satu klien MySQL AB.

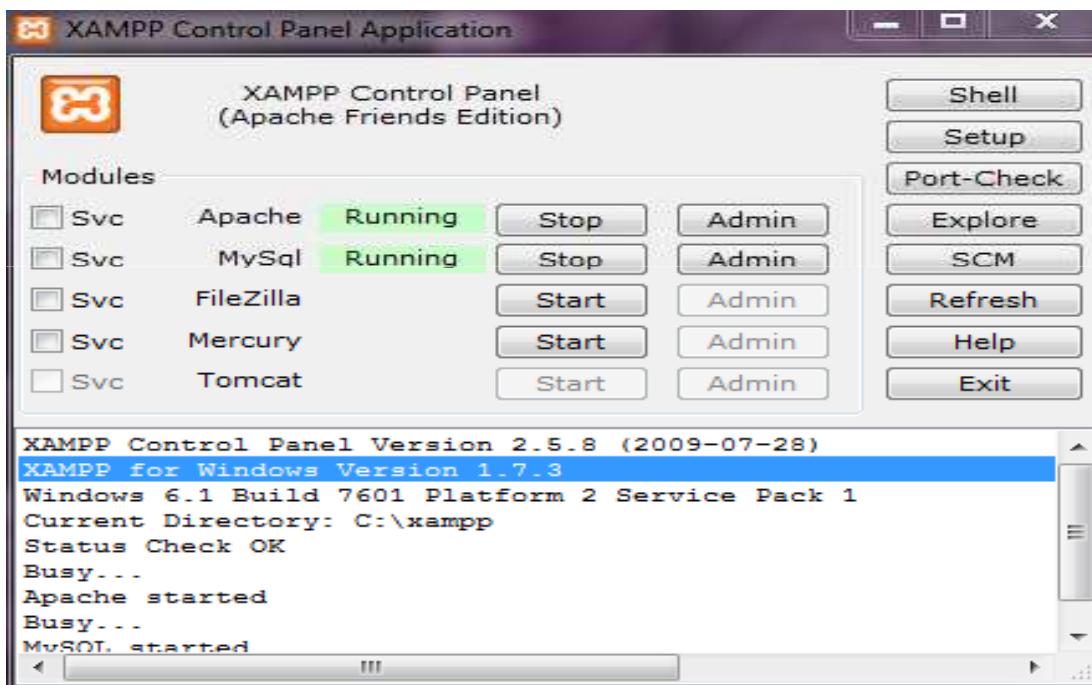
Database MySQL merupakan perangkat lunak database yang berbentuk database relasional atau dalam bahasa basis data sering disebut dengan Relation Database Management System (RDBMS) yang menggunakan suatu bahasa permintaan bernama SQL.

MySQL adalah sebuah database *server* dapat juga berperan sebagai *client* sehingga disebut database *client / server* yang *open source* dapat berjalan pada OS manapun dengan *platform* Windows maupun Linux.

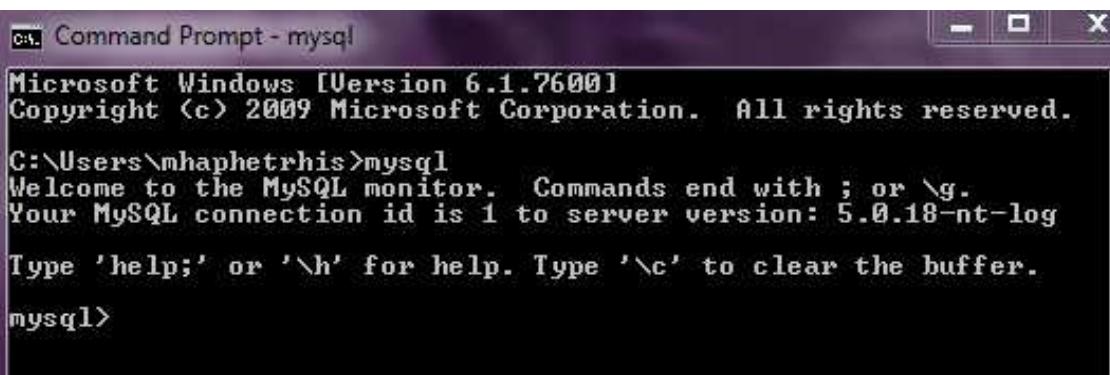
## Penggunaan MySQL di *Command Prompt*

Dalam pembahasan ini kita akan menggunakan paket program XAMPP versi 1.7.3 dengan sistem operasi Windows XP. Paket XAMPP biasanya di-instal pada direktori atau folder C:\ XAMPP. Sedangkan untuk databasenya akan tersimpan di direktori C:\xampp\mysql\data.

Silahkan melakukan penginstalan XAMPP sebelum melanjutkan ke pembahasan selanjutnya.



1. Pilih menu ***Start*** → ***All Programs*** → ***Accessories***. Kemudian Pilih ***Command Prompt***. Selanjutnya akan tampil jendela *command prompt*.
2. Kemudian ketikkan ***mysql*** pada command prompt tersebut untuk menjalankan program mysql: **C:\Users\mhaphetrhis>mysql**. Bila perintah tersebut berhasil, maka pada layar akan tampil ucapan selamat datang seperti tampilan dibawah ini:



```
Command Prompt - mysql
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\mhaphetrhis>mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.18-nt-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

MySQL merupakan bahasa pemrograman database dimana penulisan sintaknya tidak serumit bahasa pemrograman lainnya seperti Java, C++, dan sebagainya. Satu hal yang perlu diingat bahwa setiap penulisan script MySQL di Dos-Prompt harus selalu diakhiri dengan tanda titik koma (;

1. Sintak untuk menampilkan semua nama database.

**Mysql>show databases;**

2. Sintak untuk bisa masuk kedalam salah satu sistem database

**Mysql>use nama\_database;**

3. Sintak untuk menampilkan seluruh tabel yang telah ada di dalam database mysql

**Mysql>show tables;**

4. Sintak untuk menampilkan seluruh isi record dari suatu tabel

**Mysql>select \* from nama\_tabel;**

## PERTEMUAN 10

# Data Definition Language



### **Data Definition Language (DDL)**

Skema basis data dispesifikasi oleh sekumpulan definisi dengan sebuah bahasa khusus yang disebut *data-definition language* (DDL). Hasil kompilasi DDL berupa tabel-tabel yang disimpan dalam sebuah file, disebut data dictionary (Kamus Data).

#### **Penggunaan Data Definition Language (DDL)**

##### 1. Pembuatan Database

Database adalah sebuah media utama yang harus dibuat dalam membangun sebuah basis data agar nantinya dapat diletakkan beberapa tabel dengan field-fieldnya. Bila kita lihat dalam mysql database hanyalah semacam direktori saja, berbeda jika dibandingkan dengan program database yang berbentuk visual seperti access ataupun paradox.

Pada mysql database akan disimpan dan dikumpulkan pada sebuah direktori khusus yang bernama data. Dalam direktori **C:\Apache2triad\mysql\data**, semua sub direktori yang ada didalamnya disebut database.

- Sintak untuk menciptakan database baru

```
Mysql>create nama_database;
```

```
Mysql>create db_jual;
```

- Sintak untuk menghapus database

```
Mysql>drop database nama_database;
```

```
Mysql>drop database db_jual;
```

## 2. Pembuatan Tabel

Tabel adalah obyek utama yang harus ada pada sebuah basis data karena didalamnya semua data akan disimpan. Tabel terletak pada sebuah database, sehingga pembuatan table dilakukan setelah database dibuat. Untuk membuat tabel kita harus terlebih dahulu **mengaktifkan** database, perintah yang digunakan adalah **use**. Syntak penulisannya adalah : **Mysql>use namadatabase;** Setelah masuk kedalam database, anda telah diijinkan melakukan operasi-operasi menyangkut tabel dan isinya.

- Sintak untuk membuat sebuah tabel.

```
Mysql>create table nama_tabel(field-1  
type(length), field-2 type(length),..... field-n  
type(length),(.....));
```

**Mysql>create table tb\_pel (kd\_pel char(5), nm\_pel varchar(20), alamat varchar(50), telp varchar(15));**

- b. Sintak untuk melihat suatu struktur/field tabel

**Mysql>desc nama\_tabel;**

**Mysql>desc tb\_pel;**

- c. Sintak untuk membuat *primary key*

**Mysql >alter table nama\_table add primary key (nama\_field\_primary);**

**Mysql >alter table tb\_pel add primary key (kd\_pel);**

- d. Sintak untuk menghapus *Primary Key*

**Mysql>alter table nama\_table drop primary key;**

**Mysql>alter table tb\_pel drop primary key;**

- e. Sintak untuk menghapus *Primary Key*

**Mysql>drop table nama\_tabel;**

**Mysql>drop table tb\_pel;**

Keterangan	Perintah	Contoh
Mengganti nama tabel	<b>Rename</b> table tabel_lama <b>to</b> tabel_baru;	<b>Rename</b> table pelanggan <b>to</b> pelanggan_baru;
Perubahan field table	<b>Alter table</b> nama_table alter spesifikasi kondisi;	<b>Alter table</b> pelanggan <b>add</b> status varchar(10);  <b>Alter table</b> pelanggan <b>add</b> status varchar(10) <b>after</b> nama;
Menambah field unik	<b>Alter table</b> nama_table <b>add</b> <b>index</b> (nama_field);	<b>Alter table</b> pelanggan <b>add</b> <b>index</b> (nm_pel);
Menghapus field	<b>Alter table</b> nama_table <b>drop</b> nama_field;	<b>Alter table</b> pelanggan <b>drop</b> status;

Keterangan	Perintah	Contoh
Menghapus field	<b>Alter table</b> nama_table <b>drop</b> nama_field;	<b>Alter table</b> pelanggan <b>drop</b> status;
Menghapus index	<b>Alter table</b> nama_table <b>drop</b> <b>index</b> nama_field;	<b>Alter table</b> pelanggan <b>drop</b> <b>index</b> nm_pel;
Membuat Validasi	<b>Create table</b> nama_tabel(field-1 type(length), field-2 type(length), ... field-n enum("pilihan1","pilihan2")); status	<b>Create table</b> pelanggan(kd_pel char(5), nm_pel varchar(20), alamat varchar(50), telp varchar(15), enum("member","nonmember" ));

## PERTEMUAN 11

# Data Manipulation Languange

### ***Data Manipulation Language (DML)***

DML atau *Data Manipulation Language* merupakan perintah-perintah yang berfungsi untuk melakukan manipulasi data ataupun objek-objek yang ada didalam tabel.

Beberapa manfaat atau kegunaan dari DLL diantaranya adalah sebagai berikut :

1. Pengambilan informasi yang disimpan dalam basis data.
2. Penyisipan informasi baru ke basis data.
3. Penghapusan informasi dari basis data.
4. Modifikasi informasi yang disimpan dalam basis data.
5. Query adalah perintah yang ditulis untuk mengambil informasi. Bagian dari DML yang menangani pengambilan informasi ini disebut bahasa query.

### 1. Perintah Insert

Perintah INSERT digunakan untuk menambah data baru kedalam tabel.

```
mysql>insert into nama_tabel (daftar_field) values  
(daftar_nilai);
```

```
mysql>insert into tb_pel (kd_pel, nm_pel, alamat, telp)  
values ("P001", "Risma", "Ciledug", "08561234567");
```

### 2. Perintah Select

Perintah SELECT digunakan untuk menampilkan data yang terdapat didalam database.

```
Mysql>select * from nama_tabel;
```

```
Mysql>select * from tb_pel;
```

### 3. Perintah Update

Perintah UPDATE digunakan untuk melakukan perubahan terhadap sejumlah data yang ada didalam tabel yang telah dibuat.

```
Mysql>update nama_tabel set field-2=nilai_field-2,  
field-n=nilai_field-n where field-1=nilai_field-1;
```

```
Mysql>update tb_pel set nm_pel="Normah",  
alamat="Kebayoran" where kd_pel="P001";
```

### 4. Perintah Delete

Perintah DELETE digunakan untuk menghapus sejumlah data yang ada didalam tabel yang telah dibuat.

```
Mysql> delete from nama_tabel where field=nilai_field;
```

```
Mysql> delete from tb_pel where kd_pel="P001";
```

1. Buat database dengan nama **db\_barang**.
2. Buat tabel dengan nama **tb\_barang**.

Field	Type	Size	Keterangan
Kobar	Char	5	<i>Primary key</i>
Nabar	Varchar	25	
Hrg	Int		
Stok	Int		

3. Isilah *record* pada tabel tb\_barang seperti dibawah ini:

Kobar	Nabar	Hrg	Stok
B001	Pulpen	3500	20
B002	Pensil	2000	30
B003	Penghapus	1000	25

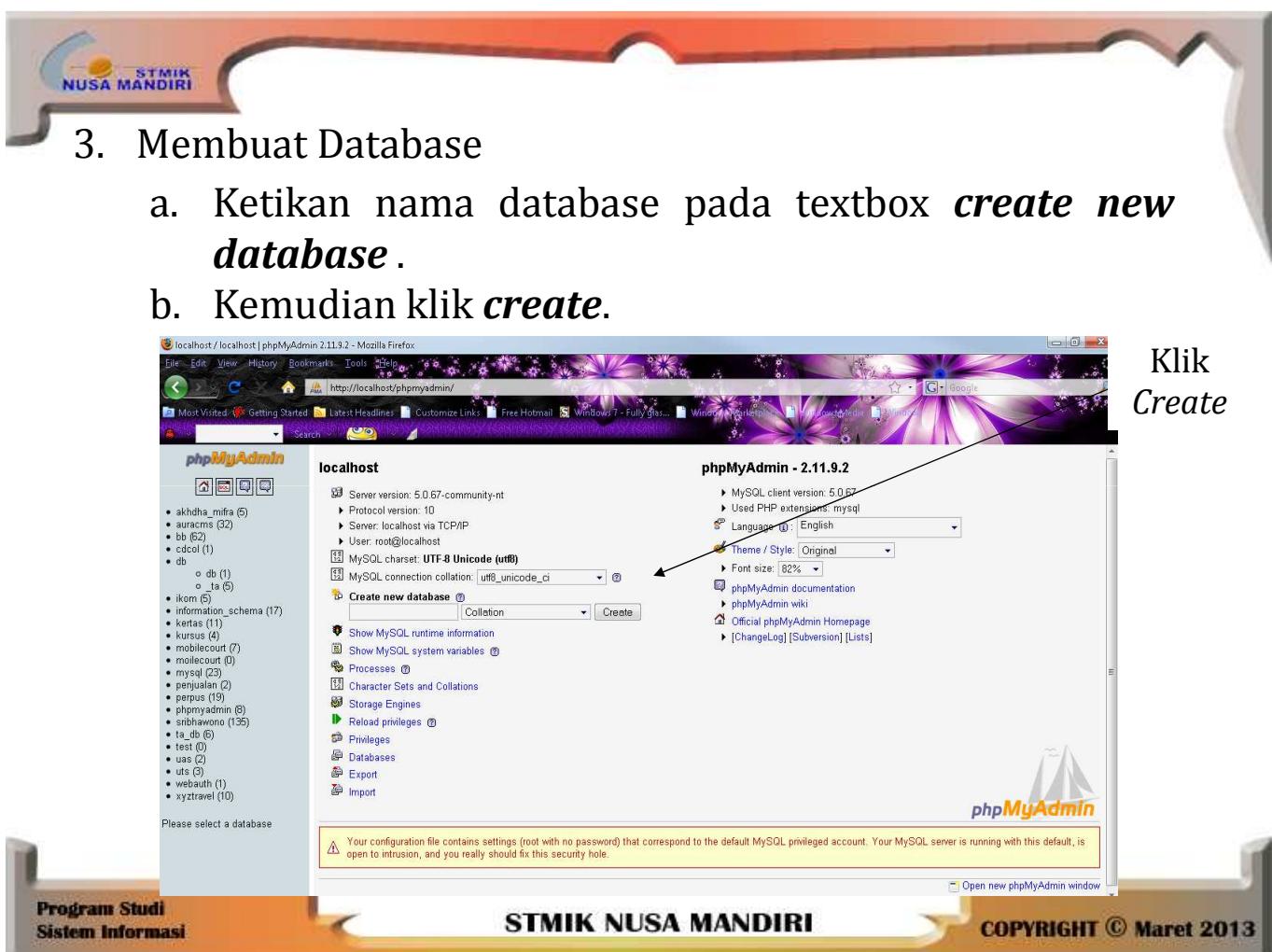
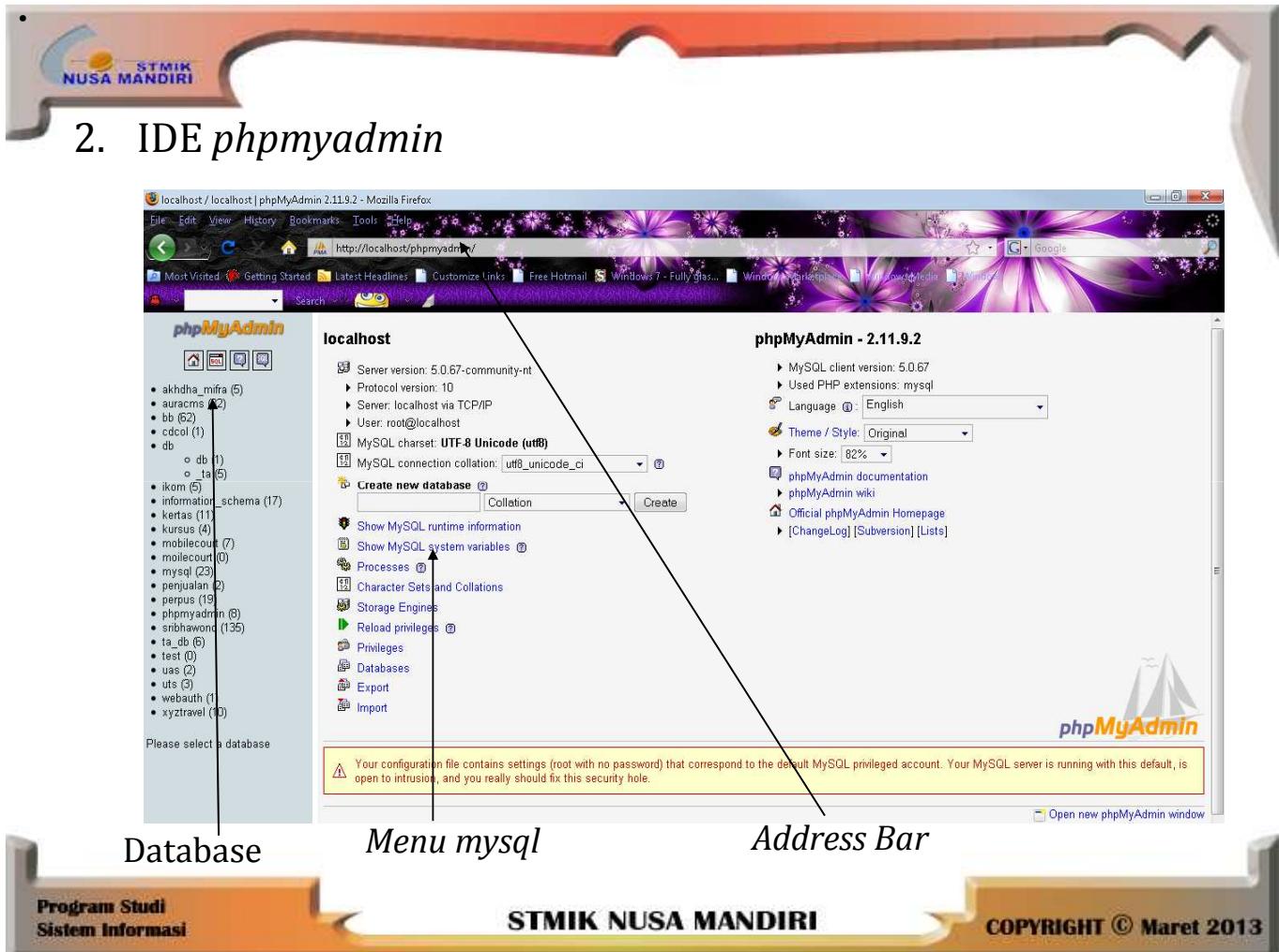
4. Tampilkanlah isi *record* dari tabel **barang**.
5. Gantilah **nabar** pada kobar **B001** menjadi **penggaris**.
6. Hapuslah **record** pada kobar **B003**.
7. Ganti nama *field hrg* menjadi **harga**.
8. Ganti stok= **20** menjadi stok =**10** untuk *record* dengan kobar "**B001**".
9. Ganti nama tabel **tb\_barang** menjadi **tb\_brg**.

## PERTEMUAN 12

# PhpMyadmin

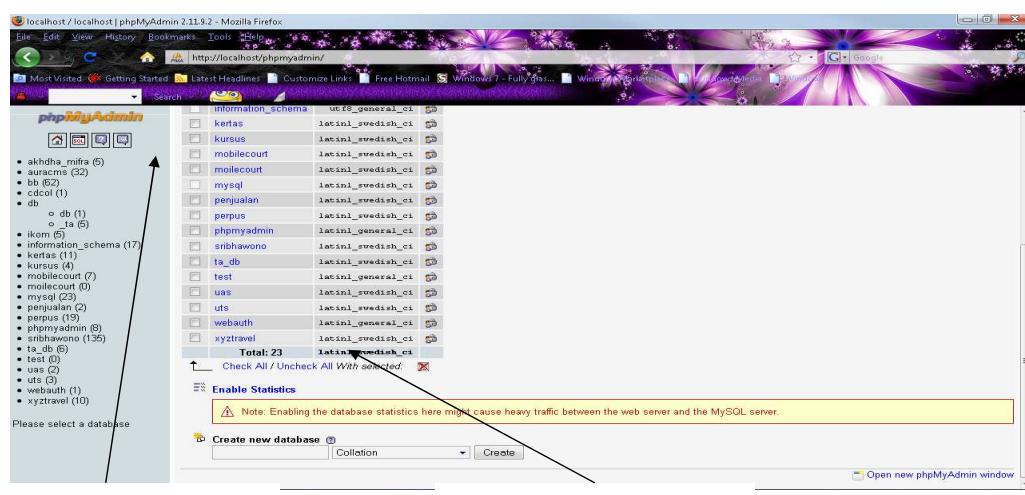
Phpmyadmin adalah alat yang ditulis dalam PHP yang ditujukan untuk menangani administrasi MySQL melalui web. Membuat dan drop database. Membuat/drop/merubah table, menghapus/mengedit/menambahkan kolom, jalankan SQL pernyataan apapun, mengatur kunci pada field, mengatur hak, ekspor data kedalam berbagai format dan tersedia dalam 50 bahasa.

1. Mengaktifkan PHPMyadmin
  - a. Buka *browser* (IE/Mozilla Firefox/Opera)
  - b. Kemudian ketikan **localhost/phpmyadmin** pada **address bar**.
  - c. Masukkan username : **root** dan passwordnya : **password**, kemudian akan tampil seperti di bawah ini :



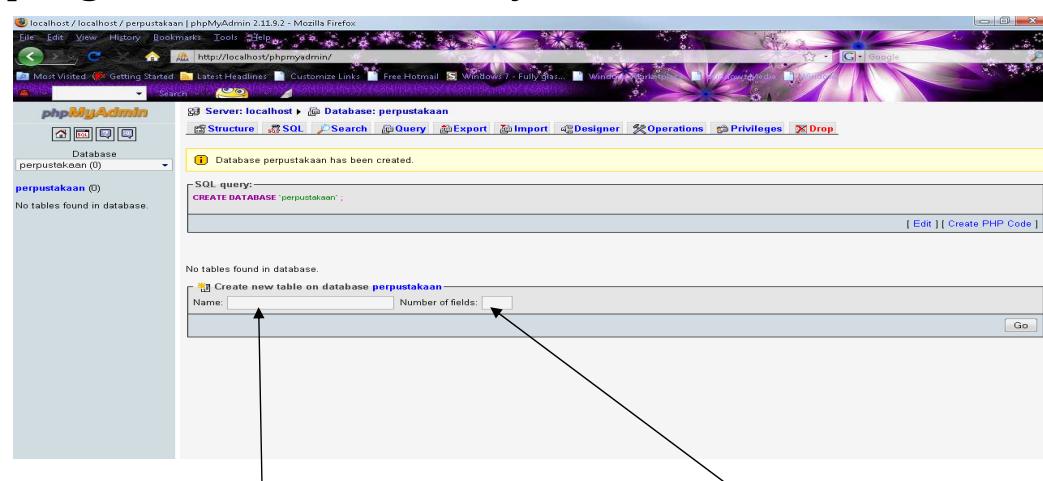
#### 4. Menghapus Database

- Pilih menu database pada tampilan *Home* Phpmyadmin.
- Checklist* pada database yang akan dihapus kemudian pilih **drop**.



#### 3. Membuat Database

- Ketikkan nama tabel pada *textbox name*, kemudian ketikkan jumlah *field* yang akan dibuat pada *number of field* tersebut setelah itu klik **Go** untuk memulai pengetikan struktur datanya.



- b. Maka tampil layar seperti gambar di bawah ini:

Field	Type	Length/Values <sup>1</sup>	Collation	Attributes	Null	Default <sup>2</sup>	Extra
	VARCHAR				not null		
	VARCHAR				not null		
	VARCHAR				not null		
	VARCHAR				not null		
	VARCHAR				not null		

- c. Kemudian ketikkan struktur tabel seperti contoh dibawah ini:

Nama Field	Type	Size	Keterangan
Id_anggota	Varchar	4	Primary Key
Nama	Varchar	20	
Alamat	Varchar	30	
Notelp	Varchar	13	
Tgl_lahir	Date		

- d. Setelah kita masukkan *field-field* yang dibutuhkan kemudian klik **save**.

Klik Save

e. Apabila telah sukses maka akan tampil tampilan seperti dibawah ini

SQL query:

```
CREATE TABLE `perpustakaan`.`anggota` (
  `id_anggota` VARCHAR(4) NOT NULL,
  `nama` VARCHAR(20) NOT NULL,
  `alamat` VARCHAR(30) NOT NULL,
  `notelp` VARCHAR(13) NOT NULL,
  `tgl_lahir` DATE NOT NULL,
  PRIMARY KEY (`id_anggota`)
) ENGINE = MYISAM
```

[ Edit ]

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	<b>id_anggota</b>	varchar(4)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>nama</b>	varchar(20)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>alamat</b>	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>notelp</b>	varchar(13)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>tgl_lahir</b>	date			No			

↑ Check All / Uncheck All With selected:

Print view Relation view Propose table structure Add 1 field(s)  At End of Table  At Beginning of Table  After **id\_anggota** Go

## PERTEMUAN 13

# Mengelola Tabel di PhpMyadmin

## 1. Menambah Field

- Buka tabel yang ingin diubah fieldnya.
- Klik menu tab **structure**.
- Masukkan jumlah *field* yang ingin ditambah.
- Pilih posisi *field at end of table/at beginning of table/after dari field* yang sudah ada.
- Setelah itu klik **Go**.

SQL query:

```
CREATE TABLE `perpustakaan`.`anggota` (
`id_anggota` varchar(4) NOT NULL ,
`nama` varchar(20) NOT NULL ,
`alamat` varchar(30) NOT NULL ,
`notelp` varchar(13) NOT NULL ,
`tempat_lahir` date
)
PRIMARY KEY (`id_anggota`)
) ENGINE = MYISAM
```

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> <b>id_anggota</b>	varchar(4)	latin1_swedish_ci		No			
<input type="checkbox"/> <b>nama</b>	varchar(20)	latin1_swedish_ci		No			
<input type="checkbox"/> <b>alamat</b>	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/> <b>notelp</b>	varchar(13)	latin1_swedish_ci		No			
<input type="checkbox"/> <b>tgl_lahir</b>	date			No			

Check All /  Uncheck All With selected:   
 Add field(s)  At End of Table  At Beginning of Table  After **id\_anggota**

Jumlah  
field

Posisi *field*

STMIK NUSA MANDIRI

COPYRIGHT © Maret 2013

- Masukan nama *field* yang ditambahkan → kemudian klik **save**.

localhost / localhost / perpustakaan / anggota | phpMyAdmin 2.11.9.2 - Mozilla Firefox

File Edit View Bookmarks Tools Help https://localhost/phpmyadmin/

Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows 7 - Fully ga... Windows Help Search

phpMyAdmin Server: localhost Database: perpustakaan Table: anggota

Browse Structure SQL Search Insert Export Import Operations Empty Drop

Field **tempat\_lahir**  
Type **VARCHAR**  
Length/Values<sup>1</sup> **20**  
Collation  
Attributes  
Null **not null**  
Default  
Extra  
Comments  
MIME type  
Browser transformation  
Transformation options<sup>3</sup>

**Klik Save**

1 If field type is "enum" or "set", please enter the values using this format: 'a','b','c'...

- g. Maka apabila telah di klik **save** akan tampil seperti dibawah ini:

Server: localhost > Database: perpustakaan > Table: anggota

SQL query:

```
ALTER TABLE `anggota` ADD `tempat_lahir` VARCHAR(20) NOT NULL;
```

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	<b>id_anggota</b>	varchar(4)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>nama</b>	varchar(20)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>alamat</b>	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>notelp</b>	varchar(13)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>tgl_lahir</b>	date			No			
<input type="checkbox"/>	<b>tempat_lahir</b>	varchar(20)	latin1_swedish_ci		No			

Check All / Uncheck All With selected:

## 2. Menghapus Field

- a. Checklist *field* yang ingin dihapus kemudian klik **drop**.

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	<b>id_anggota</b>	varchar(4)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>nama</b>	varchar(20)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>alamat</b>	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>notelp</b>	varchar(13)	latin1_swedish_ci		No			
<input type="checkbox"/>	<b>tgl_lahir</b>	date			No			
<input checked="" type="checkbox"/>	<b>tempat_lahir</b>	varchar(20)	latin1_swedish_ci		No			

Check All / Uncheck All With selected:

Checklist

Klik Drop

- b. Setelah klik **drop** maka akan tampil pertanyaan, pilih **yes**. Kemudian akan tampil pesan sukses.

### 3. Merubah Field

- a. Checklist field yang ingin dirubah, kemudian klik **change**

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input checked="" type="checkbox"/>	<b>id_anggota</b>	varchar(4)	latin1_swedish_ci		No			
<input type="checkbox"/>	nama	varchar(20)	latin1_swedish_ci		No			
<input type="checkbox"/>	alamat	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/>	notelp	varchar(13)	latin1_swedish_ci		No			
<input type="checkbox"/>	tgl_lahir	date			No			

Checklist

Klik Change

- b. Ubahlah nama field **id\_anggota** menjadi **kode**, setelah itu klik **save**.

### 4. Menambah Isi record tabel

- a. Klik database perpustakaan, kemudian checklist tabel yang ingin kita isikan data setelah itu klik **insert**.

Table	Action	Records	Type	Collation	Size	Overhead
<input checked="" type="checkbox"/> anggota		1	MyISAM	latin1_swedish_ci	2.0 KB	-
1 table(s)	Sum					

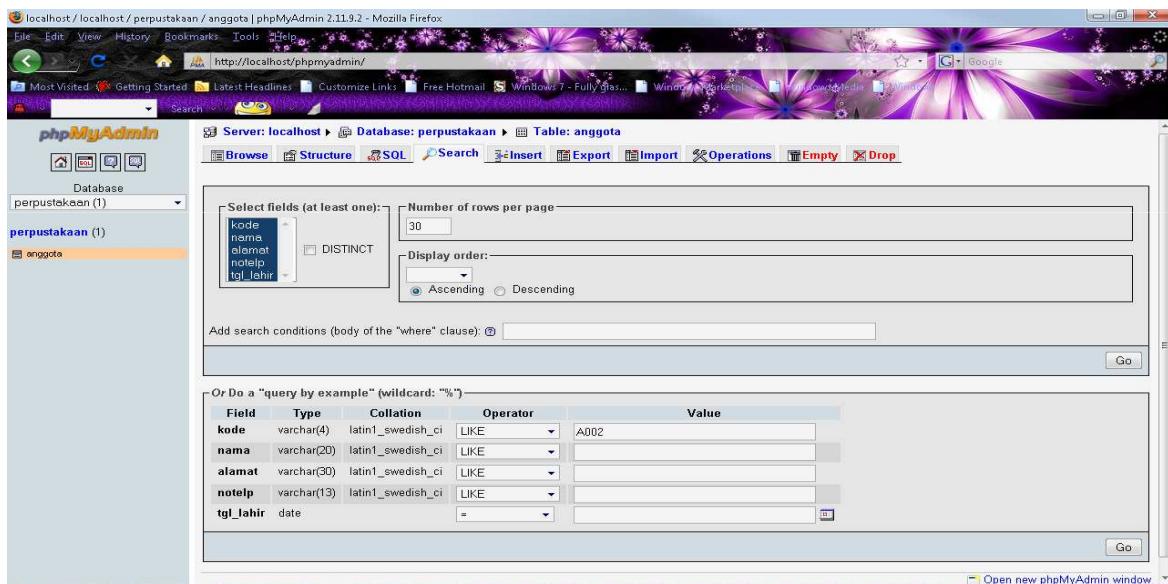
Checklist

Klik Insert

- b. Kemudian isikan sebanyak 2 record di **value**, setelah itu klik **Go**

## 5. Mencari Data

- Klik **Search** pada menu tab, kemudian masukkan kriteria yang ingin dicari di **value**, kemudian klik **Go**



The screenshot shows the phpMyAdmin interface for the 'anggota' table. In the 'Select fields (at least one)' section, 'kode' is selected. The 'Number of rows per page' is set to 30. Under 'Display order:', 'Ascending' is selected. Below these settings, there is a search condition input field. At the bottom, there is a table titled 'Or Do a "query by example" (wildcard: "%")' with columns: Field, Type, Collation, Operator, and Value. The table contains the following data:

Field	Type	Collation	Operator	Value
kode	varchar(4)	latin1_swedish_ci	LIKE	A002
nama	varchar(20)	latin1_swedish_ci	LIKE	
alamat	varchar(30)	latin1_swedish_ci	LIKE	
notelp	varchar(13)	latin1_swedish_ci	LIKE	
tgl_lahir	date		=	

# PERTEMUAN 14

# Export-Import Database

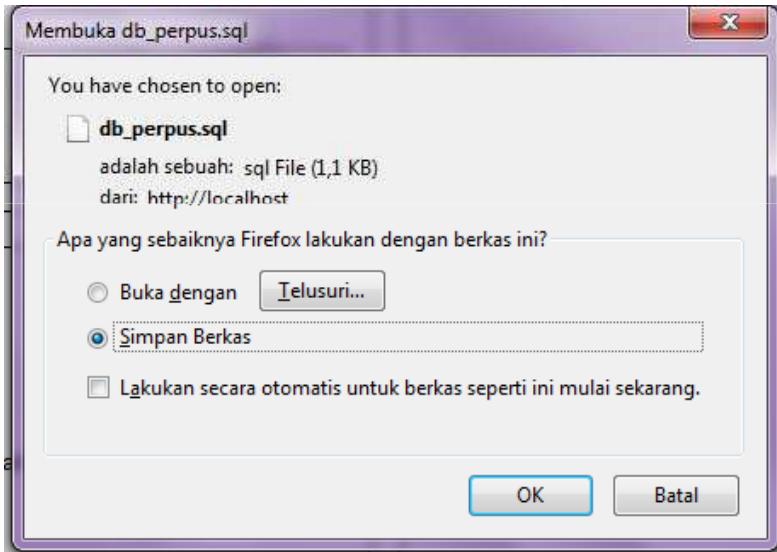
PhpMyadmin memiliki fungsi-fungsi untuk proses pengelolaan database, dimana dengan PhpMyadmin kita bisa menghapus tabel, import database maupun export database.

### 1. Export Database

- Masuk kembali ke halaman PhpMyadmin.
- Pilih database yang hendak di-*Export*.
- Pada tab menu database pilih *Export*.

- Setelah itu akan tampil halaman *Export*, lalu pada opsi *Export* pilih *SQL* dan pada opsi **Save as File** ketikkan nama filenya. Kemudian klik tombol *Go*.

- f. Simpan dan download database tersebut. Lama proses download databse tergantung besar atau tidaknya database yang tersimpan.



## 1. *Import* Database

- a. Buat database baru.
- b. Masuk kembali ke PhpMyadmin.
- c. Pilih database yang hendak di ***import*** (dengan asumsi belum ada tabel didalamnya).
- d. Klik pada tab ***Import***, lalu klik tombol **Browse**

localhost › perpustakaan

Structure SQL Search Query Export Import Operations Privileges Drop

Import

File to import

Location of the text file  Browse... (Max: 2,048KB)

Character set of the file: utf8

Imported file compression will be automatically detected from: None, gzip, bzip2, zip

Partial import

Allow interrupt of import in case script detects it is close to time limit. This might be good way to import large files, however it can break transactions.

Number of records(queries) to skip from start: 0

Format of imported file This format has no options

SQL

Go

- e. Pilih database yang hendak di *import*, lalu klik **Open**.
- f. Lalu klik tombol **Go**.
- g. Tunggu lah beberapa saat, proses *import* database sedang dilakukan.
- h. Jika *import* database sudah dilakukan, maka akan ditampilkan tabel-tabelnya.

# PERTEMUAN 15

## Latihan PPBD UAS

1. Kerjakanlah soal dibawah ini dengan menggunakan perintah SQL di *Command Prompt*.
  - a. Buatlah sebuah database dengan nama “db\_perpustakaan”.
  - b. Buatlah sebuah tabel dengan nama “tb\_anggota” dengan struktur dibawah ini:

Field Name	Type	Size	Keterangan
id_agt	Char	5	Primary Key
nama_agt	Varchar	25	
alamat	Varchar	50	
telp	Varchar	15	

- e. Isilah tabel “**tb\_anggota**” sebanyak 5 record.
  - f. Gantilah nama *field telp* menjadi **telepon**.
  - g. Gantolah alamat “**Kalibata**” menjadi alamat “**Kaliabang**” untuk record dengan id\_agt “**AGT02**”.
  - h. Gantilah nama “**tb\_anggota**” menjadi “**tb\_agt**”.
2. Kerjakalah soal dibawah secara wizard ini di *PhpMyadmin*.
- a. Buatlah sebuah tabel dengan nama “**tb\_buku**” didalam database “**db\_perpustakaan**” dengan struktur sebagai berikut:

<i>Field Name</i>	<i>Type</i>	<i>Size</i>	<i>Keterangan</i>
kd_buku	<i>Char</i>	5	<i>Primary Key</i>
Judul	<i>Varchar</i>	30	
Pengarang	<i>Varchar</i>	20	
penerbit	<i>Varchar</i>	30	

- b. Isilah tabel “**tb\_buku**” sebanyak 10 record.
- c. Export database “**db\_perpustakaan**” dan simpan di C:/LatihanUAS/NIM/db\_perpustakaan.sql.