

RANGKUMAN UAS ANSIS

PERTEMUAN 9 – KONSEP PEMODELAN

Pada pengembangan sistem informasi model biasanya digunakan untuk menggambarkan situasi yang kompleks atau frekuensi aktivitas sistem dengan pengguna. Model digunakan untuk menggambarkan sesuatu. Ada beberapa pemahaman yang berkaitan dengan model,

- Sebuah model harus cepat dan mudah untuk dibangun
- Sebuah model bisa digunakan untuk simulasi, mempelajari mengenai sesuatu yang akan direpresentasikan
- Sebuah model mampu mempelajari perkembangan dari suatu kegiatan atau masalah
- Kita bisa memilih secara rinci sebuah model
- Model bisa merepresentasikan sesuatu secara real atau tidak sebuah domain.

Diagram merupakan bentuk nyata yang digunakan untuk merepresentasikan sesuatu atau aksi dari dunia nyata. Fungsi diagram biasanya digunakan oleh analis dan designer untuk :

- Mengkomunikasikan ide-ide
- Mengenerate ide baru serta segala kemungkinan
- Melakukan tes terhadap ide serta membuat prediksi
- Mempelajari struktur dan hubungan suatu sistem

Sebuah diagram bisa digunakan sebagai ilustrasi atau dokumen untuk melihat beberapa aspek dari sistem. Sedangkan model menyediakan sudut pandang yang lebih lengkap untuk setiap tahapan tertentu.

Artifak dalam UML

didefinisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan software. Terdapat beberapa artifak utama dalam UML, yaitu :

a. Use Case Diagram

Diagram yang menggambarkan actor, use case dan relasinya

b. Class Diagram

Diagram untuk menggambarkan kelas dan relasi diantara kelas-kelas tersebut

c. Behaviour Diagram, yang terdiri dari :

i. Activity Diagram

Menggambarkan aktifitas-aktifitas, objek, state, transisi state dan event

ii. Collaboration Diagram

Menggambarkan objek dan relasinya, termasuk struktur perubahannya yang disebabkan oleh adanya suatu message

iii. Sequence Diagram

Menggambarkan objek dan relasinya termasuk kronologi (urutan) perubahan secara logis setelah menerima sebuah message

iv. Statechart Diagram

Menggambarkan state, transisi state dan event

d. Implementation Diagram, terdiri dari :

i. Component Diagram

Menggambarkan komponen dan relasi antara komponen tersebut

ii. Deployment Diagram

Menggambarkan komponen, titik awal dan relasi antara komponen tersebut

Beberapa notasi dalam UML :

1. Actor
2. Association
3. Class
4. Generalization
5. Use Case dan use case specification
6. Use Case Diagram
7. Realization
8. Sequence Diagram
9. Interaction
10. Class Diagram
11. Dependency
12. Package
13. Note
14. Interface, dll

PERTEMUAN 10 – DESAIN SISTEM

Tahap desain

Membuat keputusan bagaimana membangun sistem

Spesifikasi sistem

Hasil akhir dari tahap desain

Tahapan desain

1. Tentukan alternatif desain saat ini (buat, beli, atau kontrak/outsourcing)
2. Mengubah model proses dan model data logis menjadi model fisik
3. Desain arsitektur untuk sistem
4. Buat pilihan perangkat keras dan perangkat lunak
5. Desain input dan output sistem
6. Rancang cara data akan disimpan
7. Rancang program
8. Buat spesifikasi sistem

Strategi Desain

- Pengembangan kustom (dibuat dari awal) in-house

Pengembangan dilakukan sendiri oleh Tim di organisasi

Keuntungan:

- Memungkinkan fleksibilitas dan kreativitas
- Konsisten dengan teknologi dan standar yang ada
- Membangun keterampilan teknis dan pengetahuan fungsional organisasi

Kerugian:

- Membutuhkan waktu dan usaha yang signifikan
- Mungkin membutuhkan keterampilan yang sudah hilang
- Seringkali lebih mahal dan lebih lama
- Risiko kegagalan proyek

- Beli paket perangkat lunak (dan sesuaikan)

- Tersedia untuk banyak kebutuhan bisnis umum
- Teruji, terbukti, penghematan biaya dan waktu
- Jarang sekali cocok dengan kebutuhan bisnis
- Mungkin mengizinkan penyesuaian
- Manipulasi parameter sistem
- Mengubah cara kerja fitur
- Sinkronisasi dengan antarmuka aplikasi lain

- Pengembangan dikontrakkan ke pihak ketiga

Menyewa vendor eksternal, pengembang, atau penyedia layanan. Dapat mengurangi biaya atau menambah nilai

Ada Risiko:

- Kehilangan informasi rahasia
- Kehilangan kendali atas pengembangan di masa depan
- Kehilangan kesempatan belajar

Konversi Model Logika ke Fisik

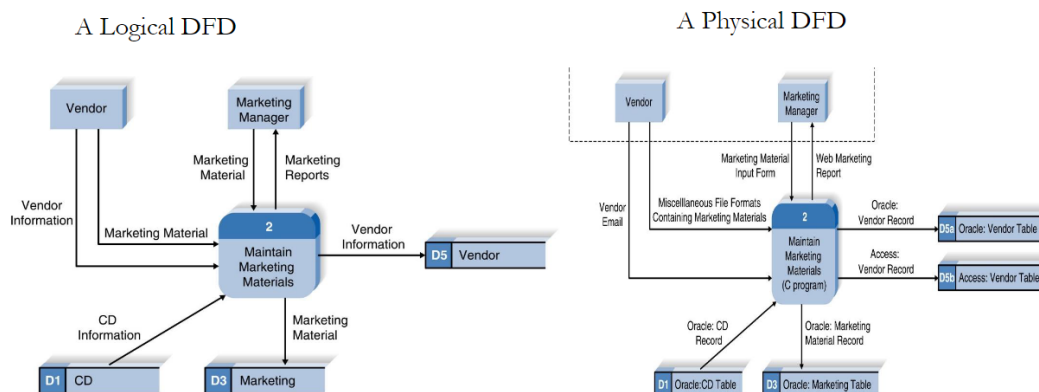
- Model proses fisik dan model data fisik
- menunjukkan detail implementasi dan menjelaskan bagaimana sistem akan bekerja, termasuk:
- Teknologi aktual dan spesifik
- Format informasi
- Interaksi manusia dengan sistem

DFD Fisik

- Berisi komponen yang sama dengan DFD logis
- Aturan yang sama terkait dengan keseimbangan dan dekomposisi juga berlaku
- Berisi perincian tambahan yang menjelaskan bagaimana sistem akan dibangun
- Gambarkan batas manusia-mesin
- Tambahkan penyimpanan data terkait sistem, aliran data, dan proses
- Perbarui elemen data dalam aliran data

DFD logis digunakan untuk memetakan proses bisnis dan menunjukkan data yang dibutuhkan dan dihasilkan. Bayangkan DFD logis sebagai "apa" dari suatu proses. DFD fisik menunjukkan proses yang sama dari perspektif sistem; ini menunjukkan "bagaimana" menyelesaikan pekerjaan.

Contoh DFD Logis vs Fisik



ERD Fisik

- Berisi komponen yang sama dengan ERD logis
- Aturan yang sama berkaitan dengan kardinalitas dan modalitas berlaku
- Berisi rincian tambahan yang menjelaskan bagaimana data akan disimpan, dalam file atau tabel database
- Diperlukan konten metadata tambahan

Tahapan Pembuatan ERD Fisik

- Ubah entitas menjadi tabel atau file
- Ubah atribut ke field
- Tambahkan primary key
- Tambahkan foreign key
- Tambahkan komponen terkait sistem

PERTEMUAN 11 – DESAIN ARSITEKTUR SISTEM

- Desain arsitektur

Merencanakan bagaimana sistem akan didistribusikan di seluruh komputer dan perangkat keras serta perangkat lunak apa yang akan digunakan untuk setiap komputer

- Spesifikasi hardware dan software

Menjelaskan komponen perangkat keras / perangkat lunak secara rinci untuk membantu mereka yang bertanggung jawab untuk membeli produk-produk tersebut.

Komponen Arsitektur Software

- Penyimpanan data (data storage)
- Algoritma akses data (data access logic)
 - Diperlukan pemrosesan untuk mengakses data yang disimpan
- Algoritma aplikasi (application logic)
 - Memproses logika aplikasi
- Algoritma presentasi (presentation logic)
 - Tampilan informasi dan pemrosesan perintah pengguna

Alternatif server

- Mainframe
- Minicomputer
- Microcomputer (personal computer)

Contoh Mainframe

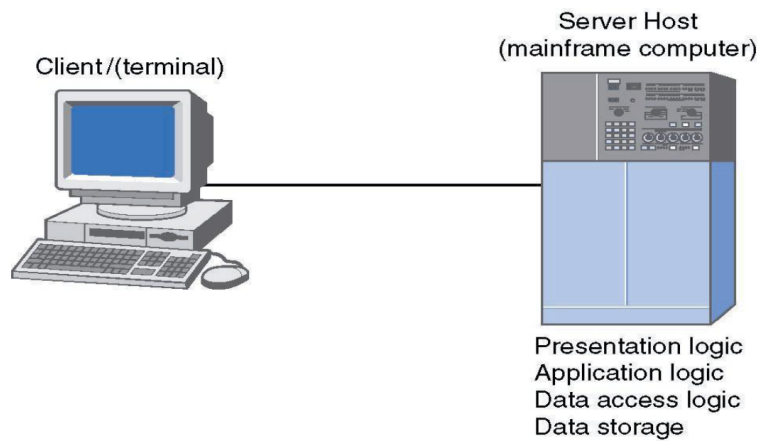
- Z Systems IBM
- AP10000-VOS3 Hitachi
- NonStop systems HP
- GS21 Fujitsu
- ACOS NEC

Alternatif Client

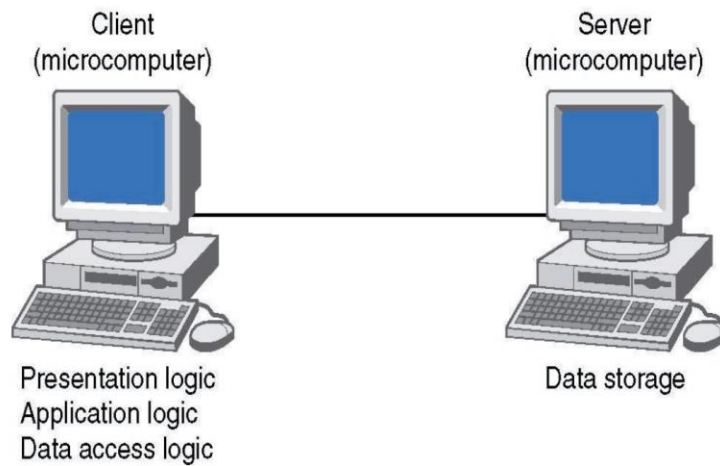
- Terminal
- Microcomputer (personal computer)
- Special purpose terminals (ATM, kiosk, Palm Pilot, dan yang lainnya)

Pilihan Arsitektur

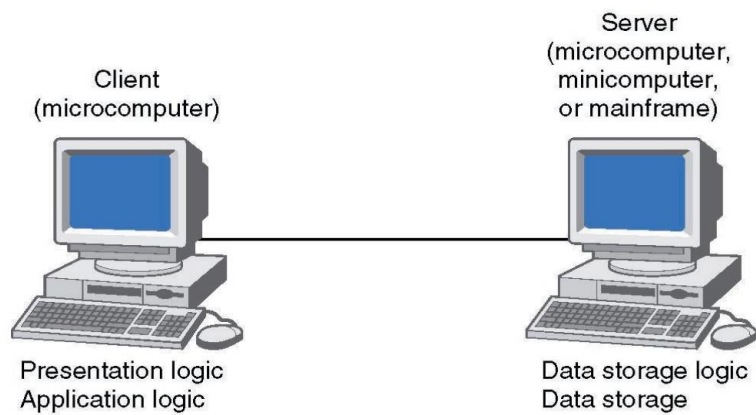
➤ Server-based Architecture



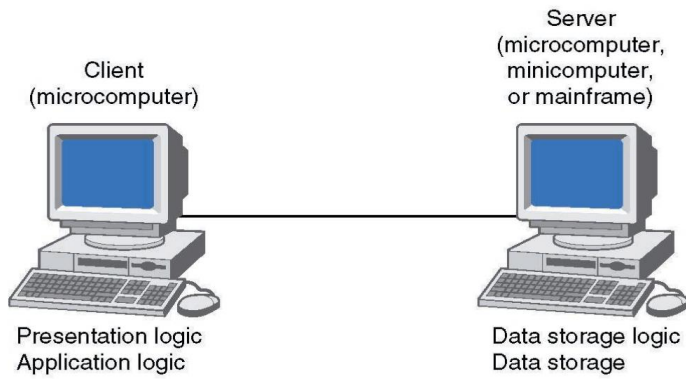
➤ Client-based Architecture



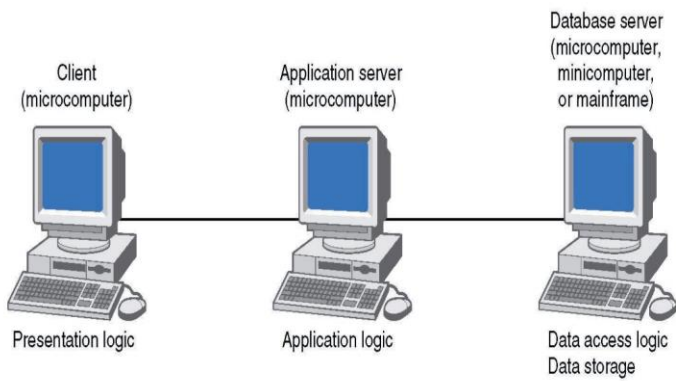
➤ Client-server based Architecture



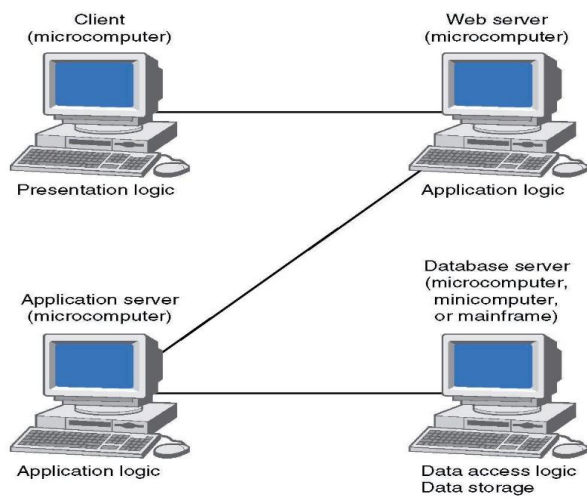
➤ Client Server-Based Architecture



Two Tiered



Three Tiered



FourTiered

N-Tiered versus 2-Tiered Architecture

- Keuntungan
 - Memisahkan pemrosesan untuk menyeimbangkan beban pada server yang berbeda
 - Lebih mudah ditingkatkan ukuran dan kemampuannya
- Kerugian
 - Beban lebih besar di jaringan
 - Lebih sulit diprogram dan diuji

Pemilihan Desain Arsitektur

- Faktor biaya lebih rendah sering digunakan untuk membenarkan pilihan client-server
- Proses seleksi yang disarankan:
 - Perluas perincian persyaratan non fungsional
 - Portability (mampu beroperasi di lingkungan berbeda)
 - Maintainability (kemudahan pemeliharaan)
 - Security (keamanan)
 - Integrasi Sistem
 - Speed (kinerja)
 - Reliability (Kehandalan)
 - Availability (Ketersediaan)
 - Pemilihan arsitektur didasarkan perincian persyaratan non fungsional

PERTEMUAN 12 – DESAIN INTERFACE USER

Antarmuka (Interface) Pengguna

- menentukan bagaimana sistem akan berinteraksi dengan entitas eksternal

Antarmuka sistem

- Menentukan bagaimana sistem bertukar informasi dengan sistem lain

Graphical User Interface (GUI)

- Antarmuka pengguna grafis adalah jenis antarmuka yang paling umum digunakan untuk mengembangkan sistem

Prinsip Desain Interface User

- Layout (tata letak)
- Content awareness (kesadaran konten)
- Aesthetics (estetika)
- User experience (pengalaman user)
- Consistency (konsisten)
- Minimal user effort (usaha minimal user)

Konsep Layout

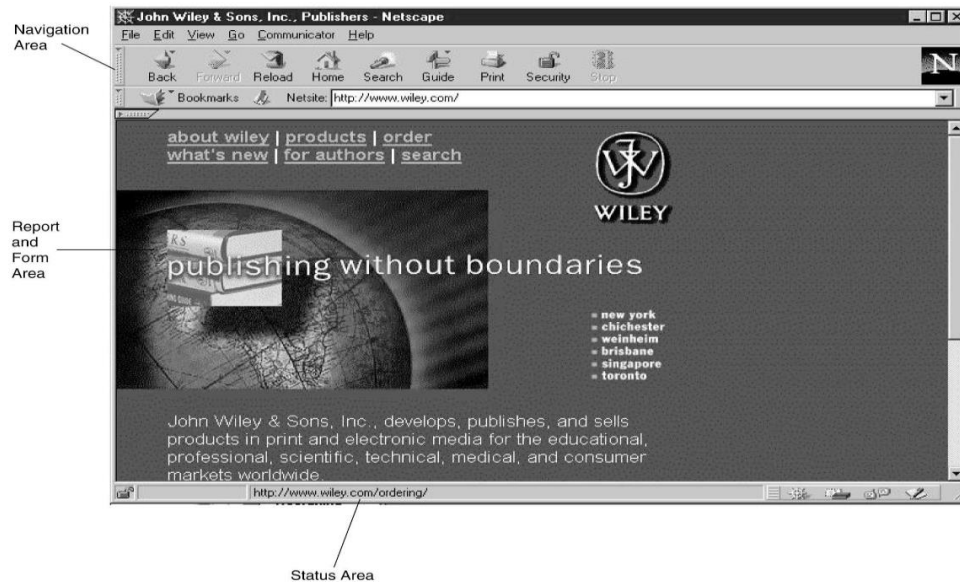
Layar sering dibagi menjadi tiga bagian

- Area navigasi (atas)
- Area kerja (tengah)
- Area status (bawah)

Informasi dapat disajikan dalam berbagai area. Area dan informasi harus meminimalkan pergerakan pengguna dari satu posisi ke posisi yang lain. Idealnya, area akan tetap konsisten dari sisi:

- Ukuran
- Bentuk
- Penempatan untuk memasukkan data
- Laporan yang menyajikan data

Contoh Layout



Content Awareness

- Semua antarmuka harus memiliki judul
- Menu harus menampilkan
 - dimana posisi saat ini
 - jalur/langkah untuk sampai ke sana
- Harus jelas informasi apa yang ada di dalam setiap area
- Filed dan label harus dipilih dengan cermat
- Gunakan tanggal dan nomor versi untuk membantu pengguna sistem

Estetika

- Antarmuka harus fungsional
- Hindari tampilan terlalu padat, terutama untuk pengguna pemula
- Desain teks dengan cermat
- Waspada font dan ukuran
- Hindari menggunakan huruf kapital semua
- Warna dan pola harus digunakan dengan hati-hati
 - Uji kualitas warna dengan mencoba antarmuka pada monitor hitam / putih
 - Gunakan warna untuk memisahkan atau mengelompokkan item

Pengalaman User

- Seberapa mudah program itu dipelajari?
- Seberapa mudah program ini digunakan untuk ahli?
- Pertimbangkan untuk menambahkan pintasan (shortcut) untuk pakar
- Ketika ada pergantian karyawan, beberapa pelatihan dapat mengurangi dampak dari antarmuka yang kurang tepat

Konsistensi

- Memungkinkan pengguna untuk memprediksi apa yang akan terjadi
- Mengurangi kurva belajar
- Mempertimbangkan item dalam suatu aplikasi dan lintas aplikasi
- Kontrol navigasi
- Terminologi
- Desain laporan dan formular

Minimal User Effort

- Aturan tiga kali klik
 - Pengguna harus dapat beralih dari menu awal atau utama sistem ke informasi atau tindakan yang mereka inginkan tidak lebih dari tiga klik mouse atau tiga penekanan tombol

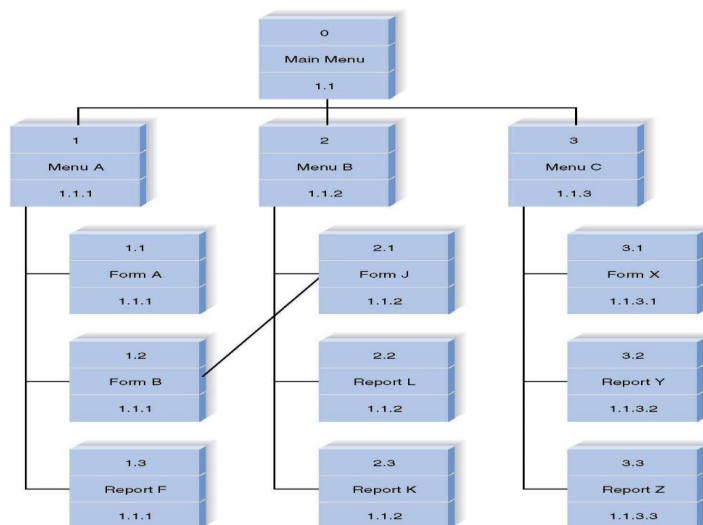
Use Scenario Development

- Garis besar langkah-langkah untuk melakukan pekerjaan
- Disajikan dalam narasi sederhana terkait dengan DFD terkait
- Dokumentasikan jalur yang paling umum melalui use case sehingga desain antarmuka akan mudah digunakan untuk situasi tersebut

Interface Structure Design

- Diagram yang menunjukkan bagaimana semua layar, formulir, dan laporan saling terkait
- Memperlihatkan cara pengguna berpindah dari satu ke yang lain
- Mirip dengan DFD dalam menggunakan kotak dan garis
- Kotak menunjukkan layar
- Garis menunjukkan gerakan dari satu ke yang lain
- Perbedaannya dengan DFD, disini tidak memiliki aturan atau format standar

Contoh



Interface Standard Design

- Elemen dasar yang umum di seluruh layar, formulir, dan laporan dalam aplikasi
- Antarmuka kiasan
 - Desktop, keranjang belanja
- Antarmuka Objek
- Antarmuka Tindakan (action)
- Antarmuka Ikon
- Antarmuka Template

Prototipe Interface Design

- Mock-up atau simulasi layar, formulir, atau laporan
- Metode umum yang sering digunakan
 - Kertas
 - Papan cerita (storyboard)
 - Prototipe HTML
 - Prototipe Bahasa

Interface Evalution

- Evaluasi heuristik
 - Bandingkan desain dengan daftar periksa (checklist)
- Evaluasi panduan (walkthrough)
 - Tim mensimulasikan gerakan melalui komponen
- Evaluasi interaktif
 - Pengguna mencoba sistem
- Pengujian kegunaan formal
 - Mahal
 - Penggunaan rinci pengujian laboratorium khusus

Desain Navigasi

- Asumsikan pengguna
 - Belum membaca manual
 - Belum mengikuti pelatihan
 - Tidak memiliki bantuan eksternal
- Semua kontrol harus jelas dan dapat dimengerti dan ditempatkan di lokasi yang intuitif di layar
- Cegah kesalahan
 - Batasi pilihan
 - Jangan pernah tampilkan perintah yang tidak dapat digunakan
 - Konfirmasikan tindakan yang sulit atau tidak mungkin dibatalkan
- Sederhanakan pemulihan dari kesalahan

Tipe Kontrol Navigasi

- Bahasa
 - Bahasa perintah

- Bahasa alami
- Menu
 - Umumnya mengarah pada menu yang luas
 - Pertimbangkan untuk menggunakan "hot keys"
- Manipulasi langsung
 - Digunakan dengan ikon untuk memulai program
 - Digunakan untuk membuat dan mengukur objek
 - Mungkin tidak intuitif untuk semua perintah

Tipe Menu

- Menu bar
- Drop-down menu
- Pop-up menu
- Tab menu
- Toolbar
- Image map

Tipe Pesan (Message)

- Error message
- Confirmation message
- Acknowledgment message
- Delay message
- Help message

Prinsip Dasar Desain Input

- Tujuannya adalah untuk secara sederhana dan mudah menangkap informasi yang akurat untuk sistem
- Mencerminkan sifat input
- Temukan cara untuk menyederhanakan cara mengkoleksi input

Tipe Input

- Data items yang terhubung ke fields
- Text
- Numbers
- Selection boxes
 - Check boxes
 - Radio buttons
 - On-screen list boxes
 - Drop-down list boxes
 - Combo boxes
 - Sliders

Type Form Input

Input Styles Example - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Guide Print Security Stop

Bookmarks Location: file:///D:/www/t.htm

Instant Message Internet Lookup New&Cool Netcaster

Sample Input Form

Name:

What is your major:
(Check one only)

☐ MIS
☐ Accounting
☐ Marketing
☐ Computer Science
☐ Management

What software do you feel comfortable using:
(Check all that apply)

☒ Word ☒ WordPerfect
☐ Excel ☐ Lotus 1-2-3 ☒ Access

Select where you were born:

Hair Color:

Interest Score

0 50 100

Document: Done

Type Validasi Input

- Completeness check
- Format check
- Range check
- Check digit check
- Consistency check
- Database checks

Prinsip Dasar Desain Output

- Memahami penggunaan laporan
 - Referensi atau esensial report?
 - Frekuensi?
 - Laporan realtime atau batch?
- Kelola pemuatan informasi
 - Hanya informasi yang dibutuhkan, tidak lebih
- Minimalkan bias

Type Report (Laporan)

- Detail reports
- Summary report
- Turnaround document
- Graphs

PERTEMUAN 13 – DESAIN DATA STORAGE (DATABASE)

Fungsi data storage (penyimpanan data)

- mengelola bagaimana data disimpan dan ditangani oleh program yang menjalankan sistem

Tujuan desain penyimpanan data

- Pengambilan data yang efisien (waktu respons yang cepat)
- Akses ke informasi yang dibutuhkan pengguna

Format Data Storage

- File: daftar data elektronik yang dioptimalkan untuk melakukan transaksi tertentu
- Database (Basis Data): kumpulan pengelompokan informasi yang saling berhubungan
- Database Management System (DBMS) adalah perangkat lunak yang membuat dan memanipulasi basis data

Atribut pada File

- File berisi informasi yang diformat untuk transaksi tertentu
- Biasanya diatur secara berurutan
- Pointer digunakan untuk mengaitkan satu record dengan record lainnya
- Linked List adalah file dengan record yang dihubungkan bersama menggunakan pointer

Tipe File

- File master - menyimpan data inti, informasi penting
- Look-up file - menyimpan nilai statis
- File transaksi - menyimpan informasi yang memperbarui file master
- File audit - merekam sebelum dan sesudah versi data
- File riwayat (arsip) - menyimpan informasi masa lalu

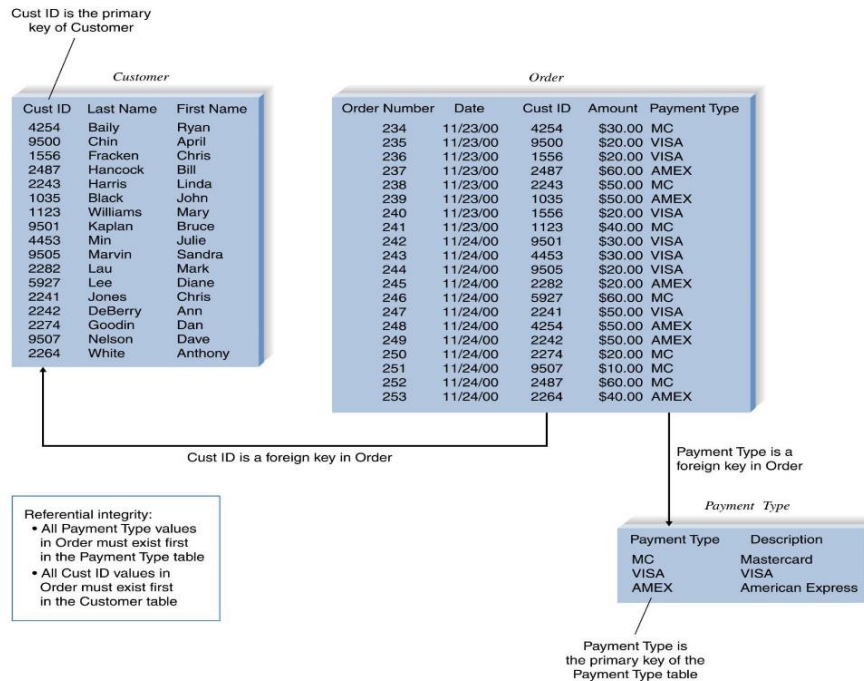
Tipe Database

- Legacy database (basis data lama)
 - Hierarchical (menggambarkan hubungan orangtua-anak menggunakan pohon terbalik)
 - Jaringan/network (menggambarkan asosiasi nonhierarkis menggunakan pointer)
- Database relasional
- Database objek
- Database multidimensi

Konsep Database Relational

- Populer, mudah digunakan untuk pengembangan
- Primary dan foreign key digunakan untuk mengidentifikasi dan menautkan tabel
- Integritas referensial memastikan sinkronisasi tabel yang benar dan valid
- Structured Query Language (SQL) - bahasa standar untuk mengakses data

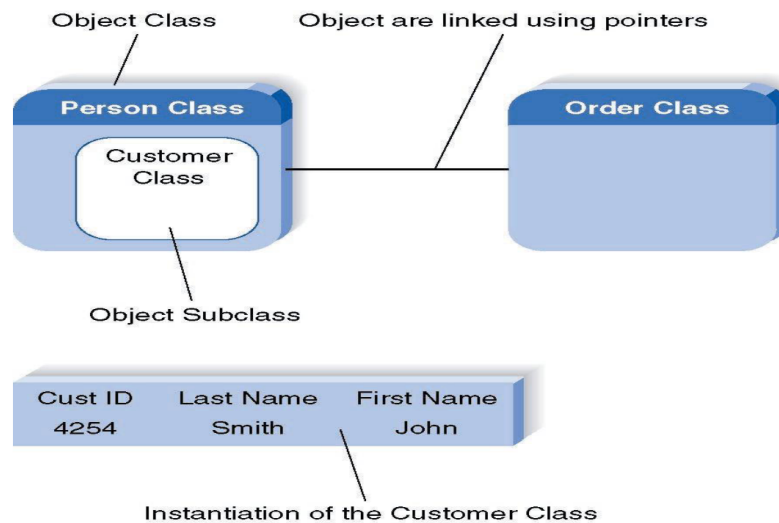
Contoh Database Relational



Konsep Database Berbasis Object

- Dibangun di sekitar objek yang terdiri dari data dan proses
- Objek dienkapsulasi (mandiri)
- Kelas objek - kategori objek utama
- OODBMS - digunakan terutama untuk aplikasi dengan data multimedia atau data kompleks
- OODBMS hibrid - fitur objek dan relasional

Contoh Database Object



Konsep Database Multidimensi

- Menyimpan data untuk kemudahan agregasi dan manipulasi di banyak dimensi
- Digunakan untuk gudang data (data warehouse)
- Data ringkasan sudah dihitung sebelumnya dan disimpan untuk kecepatan akses

Optimalisasi Data Storage

- 2 Tujuan (tetapi saling bertentangan):
 - Efisiensi penyimpanan (meminimalkan ruang penyimpanan)
 - Kecepatan akses (meminimalkan waktu untuk mengambil informasi yang diinginkan)
- Efisiensi penyimpanan
 - Minimalkan nilai null dan redundansi
 - Kurangi anomali pembaruan
 - Proses normalisasi mengoptimalkan desain penyimpanan data untuk efisiensi penyimpanan
- Optimalisasi kecepatan akses
 - Denormalisasi
 - Clustering (Intrafile dan Interfile)
 - Menempatkan record berdekatan secara fisik untuk data yang sering diakses sehingga mengurangi jumlah akses ke storage.
 - Pengelompokan Intrafile – record serupa dalam sebuah tabel disimpan bersama
 - Pengelompokan antarfile - menggabungkan record dari lebih dari satu tabel yang biasanya diambil bersama
 - Pengindeksan (indexing)

Tabel kecil yang berisi nilai dari satu atau beberapa field dalam tabel dan lokasi nilai dalam tabel. Mirip dengan indeks buku di perpustakaan

PERTEMUAN 14 – DESAIN PROGRAM

Desain program - membuat instruksi untuk programmer

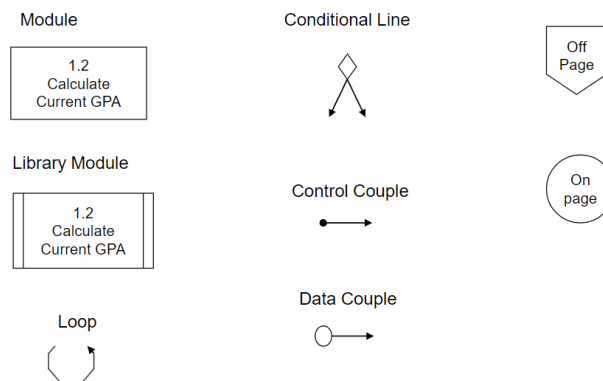
Pendekatan modular dari atas ke bawah – dimulai dengan "gambaran besar" dan secara bertahap menambahkan detail

Dokumen desain program - semua bagan struktur dan spesifikasi yang dibutuhkan oleh programmer untuk mengimplementasikan sistem

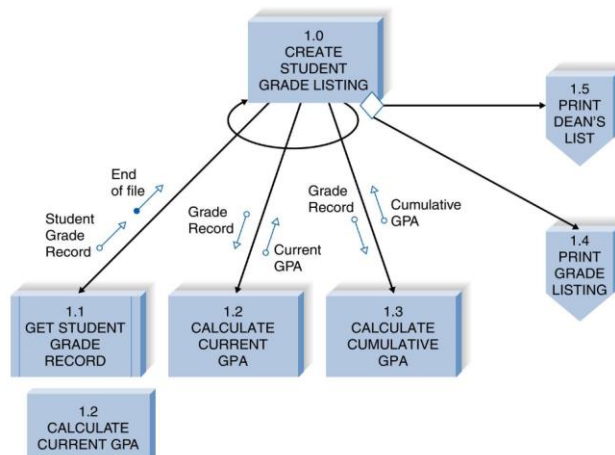
Structure Chart

- Structure Chart merupakan teknik desain program yang sering digunakan
- Menampilkan semua komponen kode dalam format hierarkis
 - Urutan (sequence)
 - Pilihan (branching)
 - Pengulangan (looping)

Elemen Structure Chart



Contoh Structure Chart



Pembuatan Structure Chart

- Satu proses dalam DFD cenderung mewakili satu modul pada structure chart
 - Proses afferent - memberikan input ke sistem
 - Proses central - melakukan operasi sistem
 - Proses efferent - menangani keluaran sistem

Tipe Structure Chart

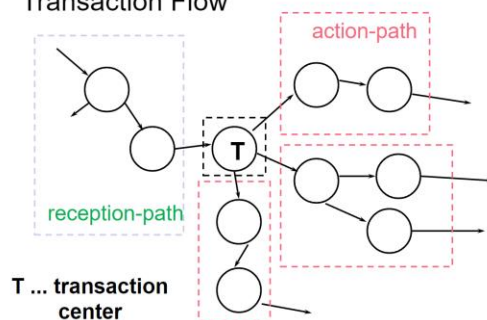
- Transaction structure - modul kontrol memanggil modul bawahan, yang masing-masing menangani transaksi tertentu
 - Banyak proses afferent
 - Sedikit proses efferent
 - Berada pada tingkat atas pada structure chart
 - Menggunakan input untuk membuat output baru
- Transform structure - modul kontrol memanggil beberapa modul bawahan secara berurutan
 - Setiap bawahan melakukan langkah dalam proses yang mengubah input menjadi output
 - Sedikit proses afferent
 - Banyak proses efferent
 - Berada pada tingkat yang lebih rendah pada structure chart

Tahapan Mapping DFD ke SC

- Tentukan aliran informasi pada DFD di sistem
- Identifikasi/Analisis pola struktural yang khas di DFD
- Gunakan metode heuristik untuk memetakan DFD ke dalam Structure Chart → Firstcut Structure Chart
- Periksa kembali dan perbaiki Structure Chart untuk mengoptimalkan hasil Structure Chart (dilakukan secara terus menerus) → Final Structure Chart

Dua Pola Mapping

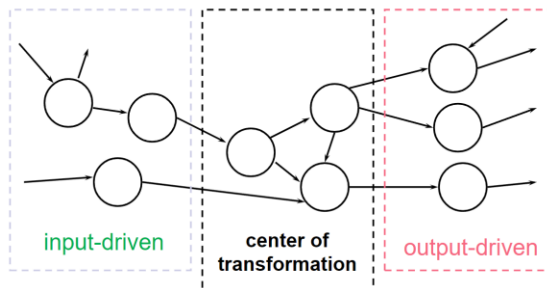
- Transaction Flow Analysis
Transaction Flow



Karakteristik Transaction Flow

- satu jalur proses penerimaan (reception path)
- transaksi: item data tunggal yang mencakup semua informasi yang diperlukan untuk eksekusi

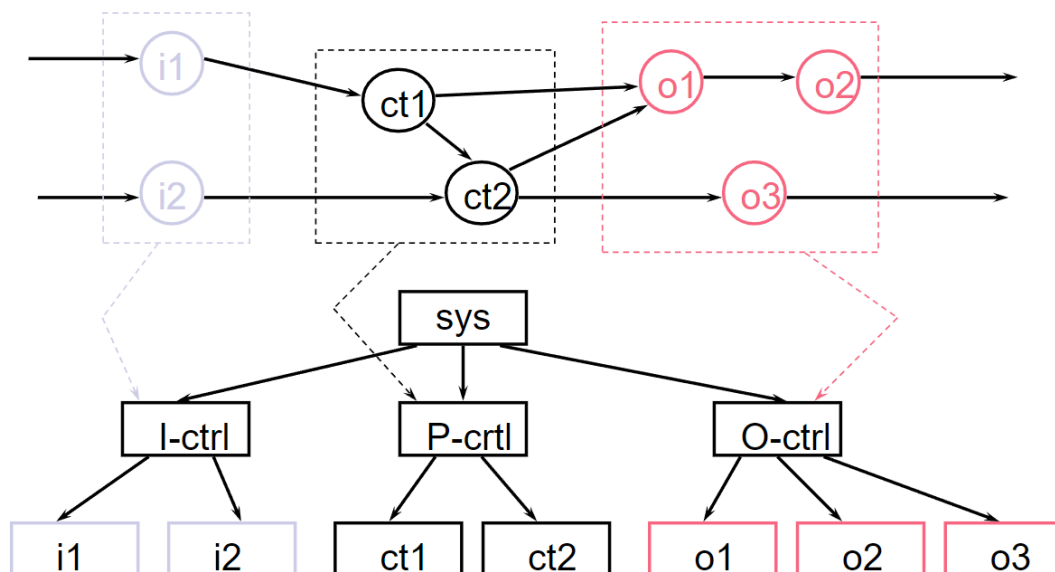
- pusat transaksi (transaction center) mengevaluasi transaksi & menginisialisasi jalur-aksi yang benar
- action-paths mengimplementasi berbagai jenis fungsionalitas sistem
- sebuah action-paths bisa merupakan sub-sistem lengkap dengan karakteristik transform flow
- Transform Flow Analysis
Transform Flow



Karakteristik Transform Flow

- sistem memiliki tujuan tunggal yang koheren
- pusat transformasi (center of transformation) mengeksekusi algoritma, transformasi data, manipulasi database, ...
- proses input-driven menyaring, memeriksa dan menerjemahkan aliran data eksternal
- proses output-driven memformat hasil untuk presentasi ke lingkungan (pengguna)
- beberapa jalur untuk mendapatkan input


Mapping Transform secara Heuristic




Optimalisasi Structure Chart

- Pembuatan structure chart dilakukan secara heuristic (perbaikan secara berkelanjutan) sampai didapatkan structure chart yang optimal
- Ciri structure chart berkualitas tinggi:
 - menghasilkan program yang modular
 - dapat digunakan kembali
 - mudah diimplementasikan
- Tindakan yang bisa dilakukan:
 - Meningkatkan kohesi dalam modul
 - Menurunkan kopling antar modul
 - Level fan-in dan fan-out yang sesuai

Tipe Kohesi

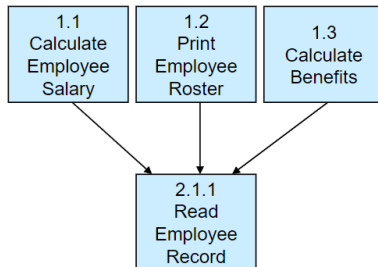
- | | |
|--|-------|
| ■ Functional: dikelompokkan karena menjalankan satu fungsi | Baik |
| ■ Sequential: dikelompokkan karena dijalankan secara berurutan | |
| ■ Communicational: dikelompokkan karena mengelola data yang sama | |
| ■ Procedural: output dari satu bagian menjadi input di bagian lainnya | |
| ■ Temporal: karena kesamaan waktu proses | |
| ■ Logical: dikelompokkan karena secara logika melakukan hal yang sama | |
| ■ Coincidental: setiap bagian tidak mempunyai hubungan yang penting | Buruk |
- 

Tipe Kopling

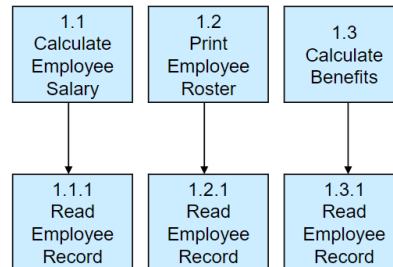
- | | |
|---|-------|
| ■ Data: antar modul berbagi data, misalnya melalui parameter | Baik |
| ■ Stamp: modul berbagi struktur data komposit dan hanya menggunakan bagian-bagiannya | |
| ■ Control: satu modul mengendalikan aliran modul lainnya | |
| ■ Common: beberapa modul memiliki akses ke data global (common) yang sama | |
| ■ Content: satu modul menggunakan kode modul lain | Buruk |
- 

Fan-in dan Fan-out

- **Fan-in:** jumlah modul yang masuk ke suatu modul
- **Fan-out:** jumlah modul yang keluar dari suatu modul
- Fan-in yang tinggi lebih baik dibanding Fan-in yang rendah



Fan-in tinggi,
penggunaan kembali modul
sebelumnya tinggi



Fan-in rendah,
kurang baik

PERTEMUAN 15 – IMPLEMENTASI SISTEM

Transisi ke sistem baru melibatkan pengelolaan

perubahan norma dan kebiasaan yang sudah ada sebelumnya

Manajemen perubahan melibatkan:

- Unfreezing - melonggarkan kebiasaan dan norma orang
- Moving - transisi dari sistem lama ke sistem baru
- Refreezing - melembagakan dan membuat efisien cara baru dalam melakukan sesuatu

Perencanaan Migrasi

Kegiatan apa yang akan dilakukan kapan dan oleh siapa

- Aspek teknik
 - menginstal perangkat keras dan perangkat lunak
 - mengonversi data
- Aspek organisasi
 - melatih pengguna tentang sistem
 - memotivasi karyawan untuk menggunakan sistem baru untuk membantu pekerjaan mereka

Tipe Konversi

- Konversi langsung (Direct conversion)
 - sistem baru langsung menggantikan yang lama
- Konversi paralel (Parallel conversion)
 - untuk sementara waktu sistem lama dan baru digunakan. Yang lama ditinggalkan ketika yang baru terbukti sepenuhnya mampu

Lokasi Konversi

- Konversi percontohan (Pilot conversion)
 - Satu atau lebih lokasi dikonversi sebelum memperluas ke Lokasi lain
- Konversi bertahap (Phased conversion)
 - Lokasi dikonversi secara bertahap
- Konversi simultan (Simultaneous conversion)
 - Semua lokasi dikonversi pada saat yang sama

Konversi Modul

- Konversi seluruh sistem (Whole system conversion)
 - Semua modul dikonversi dalam satu langkah
- Konversi modular (Modular conversion)
 - Ketika hubungan antar modul longgar, konversi dapat dilakukan satu per satu

Faktor kunci Pemilihan Strategi Konversi

- Risiko
 - Keseriusan konsekuensi dari bug yang tersisa
- Biaya
 - Paralel membutuhkan pembiayaan untuk dua sistem untuk jangka waktu tertentu
 - Secara bersamaan membutuhkan lebih banyak staf untuk mendukung semua lokasi
- Waktu
 - Paralel, bertahap, dan modular membutuhkan lebih banyak waktu

Manajemen Perubahan

Tahapan dalam manajemen perubahan:

1. Merevisi kebijakan manajemen
2. Menilai model biaya dan manfaat dari bagian/pengguna pengadopsi potensial
3. Memotivasi adopsi
4. Memungkinkan orang untuk mengadopsi

Resistensi Terhadap Perubahan

- Perubahan yang menguntungkan organisasi tidak selalu menguntungkan setiap individu
- Beradaptasi dengan proses kerja baru membutuhkan upaya, yang mungkin tidak ada kompensasi tambahan

Revisi Kebijakan Manajemen

- Tidak ada sistem komputer yang akan berhasil diadopsi kecuali kebijakan manajemen mendukung pengadopsiannya
- Alat manajemen untuk mendukung adopsi
 - Standart Operasional Prosedur (SOP)
 - Pengukuran dan penghargaan
 - Alokasi sumber daya

Pelatihan (Training)

- Setiap sistem baru membutuhkan keterampilan baru
- Keterampilan baru mungkin melibatkan penggunaan teknologi itu sendiri
- Keterampilan baru mungkin diperlukan untuk menangani proses bisnis yang berubah

Tipe Pelatihan

- One-to-one (individual)
- Classroom (ruang kelas)
- Computer based (berbasis computer)

Aktivitas Pasca Implementasi

- Penyediaan dukungan (support)

- Bantuan dalam menggunakan sistem
- Perbaikan/perawatan
 - Perbaiki atau perbaiki bug atau kesalahan yang ditemukan
 - Tambahkan peningkatan kecil untuk memberikan nilai tambah
- Pengkajian proyek
 - Analisis apa yang dilakukan dengan baik
 - Temukan kegiatan apa yang perlu ditingkatkan di masa depan

Tipe Dukungan (Support)

- Pelatihan sesuai permintaan pada saat dibutuhkan pengguna
- Dukungan online
 - Pertanyaan yang sering diajukan (FAQ)
- Meja bantuan (Helpdesk)
 - Layanan telepon untuk masalah yang diketahui