MODUL PRAKTIKUM

IMPLEMENTASI SISTEM INFORMASI

UNIVERSITAS ESA UNGGUL

JAKARTA

2017

SIKLUS HIDUP PENGEMBANGAN SISTEM

SDLC (System Development Life Cycle) --> Siklus Hidup Pengembangan Sistem

Fase Utama:

- Perencanaan: (Mengapa Mengembangkan Sistem ?)
- Analisis: (Siapa, apa, kapan dan dimana sistem ?)
- Perancangan: (Bagaimana kerja sistem?)
- Implementasi: (Bagaimana Sistem Dipasang/diinstal?)

Perencanaan:

- Mengidentifikasikan Nilai Bisnis
- Analisis Kelayakan
- Membuat Rencana Kerja
- Mengatur Staff
- Mengontrol dan Mengarahkan Projek

Analisis:

- **Analisis**
- Mencari informasi yang terkait dengan sistem
- Menentukan model proses
- Menentukan

model data

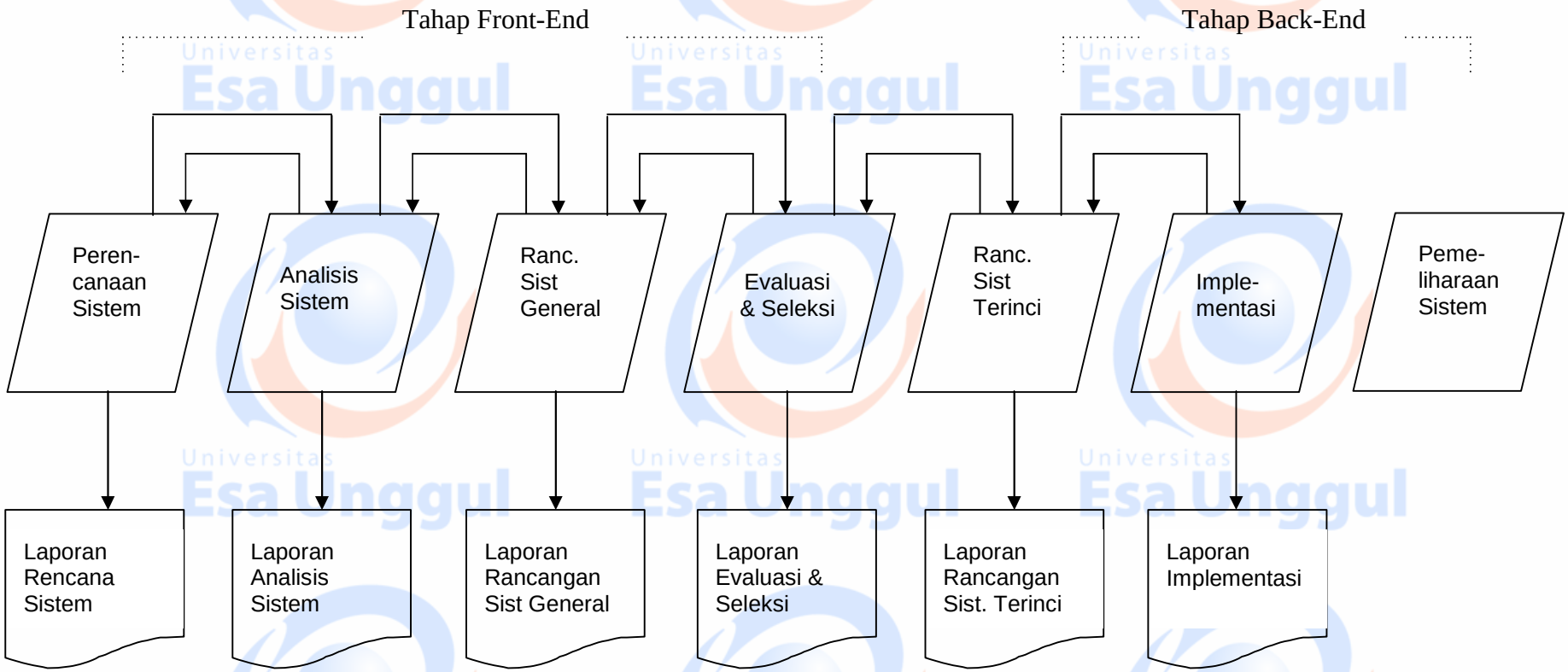
Perancangan

- **Perancangan Proses secara Fisik**
- Perancangan Arsitektur Sistem
- Perancangan Interface
- Perancangan Basis Data dan Berkas
- Perancangan Program

Implementasi:

- Construction
- Instalation

Gambar Siklus Hidup Pengembangan Sistem



- Proses yang direkayasa secara logik untuk mengembangkan sistem dari tahap perencanaan sampai penerapan
- 4 (empat) tahap pertama ---> Tahap FRONT -END
  - Digerakkan oleh pemakai
  - Untuk menyelidiki konsep sistem baru dan menentukan dengan tepat apa yang dibutuhkan para pemakai sebelum merancang sistem secara terinci
  - Dokumentasi Laporan yang dibuat ditujukan untuk para pemakai sistem
- 2 Tahap terakhir ---> Tahap BACK-END
  - Digerakkan oleh perancang dan teknokrat
- Proses dari pengembangan sistem yang terutama :
  - Analisis sistem
  - Desain sistem
  - Implementasi sistem

    + Proses kebijakan
    + Perencanaan sistem dalam tahapan pengembangan sistem (proses ini merupakan tahapan sebelum dilakukan pengembangan sistem → initiation of system project)

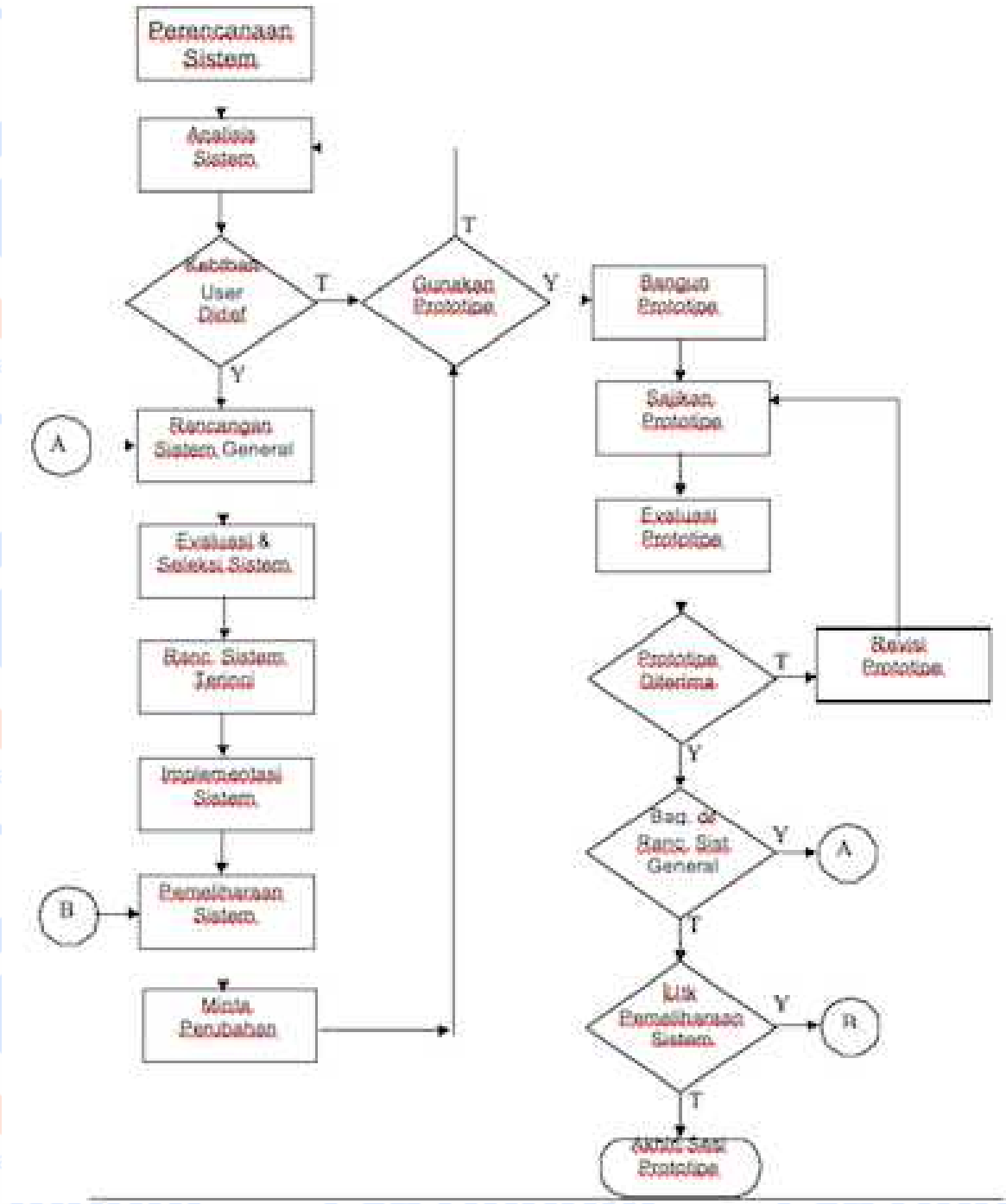
- Desain sistem dalam 2 tahapan :
  1. Desain sistem secara umum/ konsep/ makro/ logika/khusus
  2. Desain sistem secara rinci/fisik
- Setelah sistem baru dikembangkan dan diimplementasikan → Tahap Pemeliharaan (10 -20 tahun atau lebih)
- Jika sistem ini tidak lagi efisien dan efektif untuk tetap digunakan, maka tidak dilanjutkan dan sistem baru dikembangkan

PROTOTIPE

- Suatu teknik analisis dan rancangan yang memungkinkan pemakai ikut serta dalam menentukan kebutuan dan pembentukan sistem apa yang akan dikerjakan untuk memenuhi kebutuhan tersebut.
- Prototipe digunakan untuk mengembangkan kebutuhan pemakai yang sulit didefinisikan untuk memperlancar proses SDLC.
- Prototipe paling baik digunakan untuk mengembangkan sistem yang didefinisikan kurang baik dan cocok untuk penerapan sistem kecil yang unik.

Tabel berikut ini menunjukkan bagaimana prototipe digunakan dalam hubungan dengan SDLC

Karakteristik Sistem	Metodologi	
	Prototipe	**SDLC**
Kebutuhan Pemakai	Pemakai mempunyai kesulitan dalam mendefinisikan kebutuhan	Kebutuhan pemakai pada umumnya didefinisikan dengan baik
Masukan, Keluaran & Transaksi	Volume rendah	Volume tinggi
Database	Jumlah kecil catatan dan elemen-elemen dlm catatan	Jumlah besar catatan dan elemen-elemen dlm catatan
Kendali	Kendali editing dasar	Sistem kendali ekstensif, termasuk kendali keamanan canggih
Teknologi	Biasanya suatu komputer yg berdiri sendiri tanpa database "pribadi"	Biasanya suatu sistem komputer banyak pemakai yg besar, sering saling dikaitkan dgn suatu


Perancanaan Sistem
Analisis Sistem
Kebiasaan User Old.sf
T
Gunakan Prototipe
Y
T
Y
Bangun Prototipe
Susun Prototipe
Evaluasi Prototipe
Prototipe Diterima
T
Revisi Prototipe
Y
Bag. dr Ranc. Sist. General
Y
A
T
Utk Pemeliharaan Sistem
Y
B
T
Akhir Sist. Prototipe
A
Rancangan Sistem General
Evaluasi & Seleksi Sistem
Ranc. Sistem Terinci
Implemeotasi Sistem
B
Pemeliharaan Sistem
Minta Perubahan

PERANGKAT PEMODELAN

Perangkat pemodelan merupakan salah satu ciri pendekatan terstruktur. Perangkat pemodelan adalah suatu model yang digunakan untuk menguraikan sistem menjadi bagian-bagian yang dapat diatur dan mengkomunikasikan ciri konseptual dan fungsional kepada pengamat

Peran perangkat pemodelan :

1. Komunikasi
   Perangkat pemodelan dapat digunakan sebagai alat komunikasi antara pemakai dengan analis sistem dalam pengembangan sistem.
2. Eksperimentasi
   Pengembangan sistem bersifat trial and erroe
3. Prediksi
   Model meramalkan bagaimana suatu sistem akan bekerja

Jenis perangkat pemodelan antara lain :

1. Diagram Arus Data (DFD)
   Menunjukkan proses yang dijalankan data dalam sistem
2. Kamus Data
   Definisi elemen data dalam sistem
3. Entity Relationship
   Diagram (ERD) Model
   penyimpanan data
   dalam DFD
4. State Transition Diagram (STD)
   Menunjukkan keadaan tertentu dimana suatu sistem dapat ada dan transisi yang menghasilkan keadaan tertentu yang baru. STD digunakan untuk sistem yang real time.
5. Bagan Struktur
   Menggambarkan suatu hierarki modul program perangkat lunak termasuk dokumentasi interface antar modul
6. Diagram Alur Program Terstruktur (Structured
   Program Flowchart) Menggambarkan alur dan logika
   program
7. Alat Spesifikasi Proses
   Memberikan deskripsi yang lengkap tentang proses-proses yang ditemukan dalam diagram alur data tingkat dasar.
   Contoh :
   - Bahasa Inggris Terstruktur
   - Tabel Keputusan
   - Pohon Keputusan
   - Persamaan
8. Diagram Warnier-Orr (WOD)
   Menunjukkan penguraian hierarkhi proses atau data

9. Diagram Jackson

Membuat model struktur program perangkat lunak dari struktur data.

1. **Tahap Evaluasi dan Seleksi**
   - Alat dan Teknik yang digunakan :
     - Lembar kerja kelayakan TELOS
     - Lembar kerja faktor strategik PDM
     - Lembar kerja MURRE (Maintainability, Usability, Reusability, Realibility dan Extendability)
     - Analisis biaya dan keuntungan
   - Tujuan utama :
     Mendefinisikan hasil yang optimal dari setiap alternatif-alternatif rancangan secara umum
   - Hasil :
     Laporan Evaluasi dan seleksi

2. **Tahap Perancangan Rinci**
   - Alat dan Teknik yang digunakan :
     - Various layout grids
     - Various modeling tools
   - Tujuan utama :
     Membuat rancangan secara fungsional untuk : output, input, proses, control, database, dan platform teknologi
   - Hasil :
     Laporan rancangan rinci (blueprint untuk sistem baru)

3. **Tahap Implementasi**
   - Alat dan Teknik yang digunakan :
     - Software Metric
     - Struktur berbentuk grafik
     - Struktur program flowchart komputer
     - Struktur berbentuk Bhs Inggris pengembangan
     - Decision Table
     - Decision Tree
     - Equation
     - W/O diagram implementasi
     - JAD
     - ERD yg sudah dimodifikasi
     - Bahasa pemrograman
     - Perangkat lunak untuk
     - Walkthrough
     - Test Case
     - Training
     - Review sebelum
   - Tujuan utama :
     Membangun sistem baru dan mengoperasikan
   - Hasil :

Laporan implementasi sistem

PERANCANGAN SISTEM INFORMASI

2.1. PERANCANGAN INTERATIF, PENGUJIAN, DAN EVALUASI

Participatory Design dan Analisis Tugas

- Participatory design adalah perancangan yang melibatkan pemakai
- Dampak Keterlibatan pemakai dalam perancangan :
  - Menghasilkan lebih banyak informasi yang akurat tentang tugas
  - Memberi kesempatan untuk berargumen atas keputusan rancangan
  - Memberi rasa keikutsertaan yang membentuk investasi ego dalam implementasi yang sukses
  - Potensi untuk meningkatkan penerimaan pemakai atas Sistem Final
- Perancangan adalah hal yang kreatif dan tak dapat diduga. Carroll dan Rosson menyebutkan karakteristik perancangan sebagai berikut :
  - Perancangan adalah suatu *proses*; bukan merupakan keadaan dan tidak dapat direpresentasikan dengan memadai oleh statistik
  - Proses perancangan adalah *non-hierarkis*; tidak ketat *bottom-up* maupun *top-down*
  - Proses perancangan adalah *transformasional secara radikal* ; melibatkan pengembangan solusi sebagian dan sementara yang akhirnya mungkin tidak berperan dalam rancangan akhir
  - Perancangan secara intrinsik melibatkan *penemuan tujuan-tujuan baru*

2.1.1. Sasaran Perancangan Sistem Informasi

- Menentukan secara tepat banyaknya Informasi yang dibutuhkan User
- Melakukan Standarisasi
- Pengembangan Sistem Pengendalian
- Mengurangi Fungsi-fungsi yang terduplikasi

2.1.2. Panduan Dasar Perancangan Sistem

- Apakah Sistem berjalan akan dikembangkan atau disederhanakan
- Apakah sejumlah langkah telah dijalankan
- Menghindari Fungsi yang mengalami Redundansi & Duplikasi

- Sistem baru **harus lebih** lengkap, bekerja lebih cepat, menyeluruh, dan lebih murah dibanding Sistem berjalan
- Laporan yang dikeluarkan Sistem Baru harus benar-benar memenuhi kebutuhan Informasi User dan Management
- Diperlukan Sarana yang mempermudah pengendalian terhadap Implementasi Sistem

2.1.3 Prinsip Pemgembangan Sistem

- Enduser harus dilibatkan secara aktif
- Pendekatan Fase pekerjaan perlu dilakukan
- Fase tersebut dapat saling tumpang tindak (parallel)
- Sistem yang dihasilkan merupakan investasi Perusahaan
- Management harus berani menyetop suatu pekerjaan pengembangan sistem yang dirasakan tidak layak
- Dokumentasi merupakan bagian yang tidak terpisahkan dari hasil akhir suatu proyek pengembangan sistem

2.1.4 Proses Pengembangan

Kembangkan spesifikasi dokumen

- Gunakan orientasi tugas
- Gunakan rancangan minimalis
- Tangani pembaca yang beraneka ragam
- Tentukan tujuan
- Organisasikan informasi dan kembangkan visualisasi
- Perhatikan tata letak dan warna

2. Buat prototipe
3. Buat draft (bentuk kasar)
4. Perbaiki
5. Kaji ulang
6. Uji di lapangan
7. Terbitkan
8. Lakukan ulasan pascaproyek
9. Pelihara

2.1.4.1. **Prototipe System**

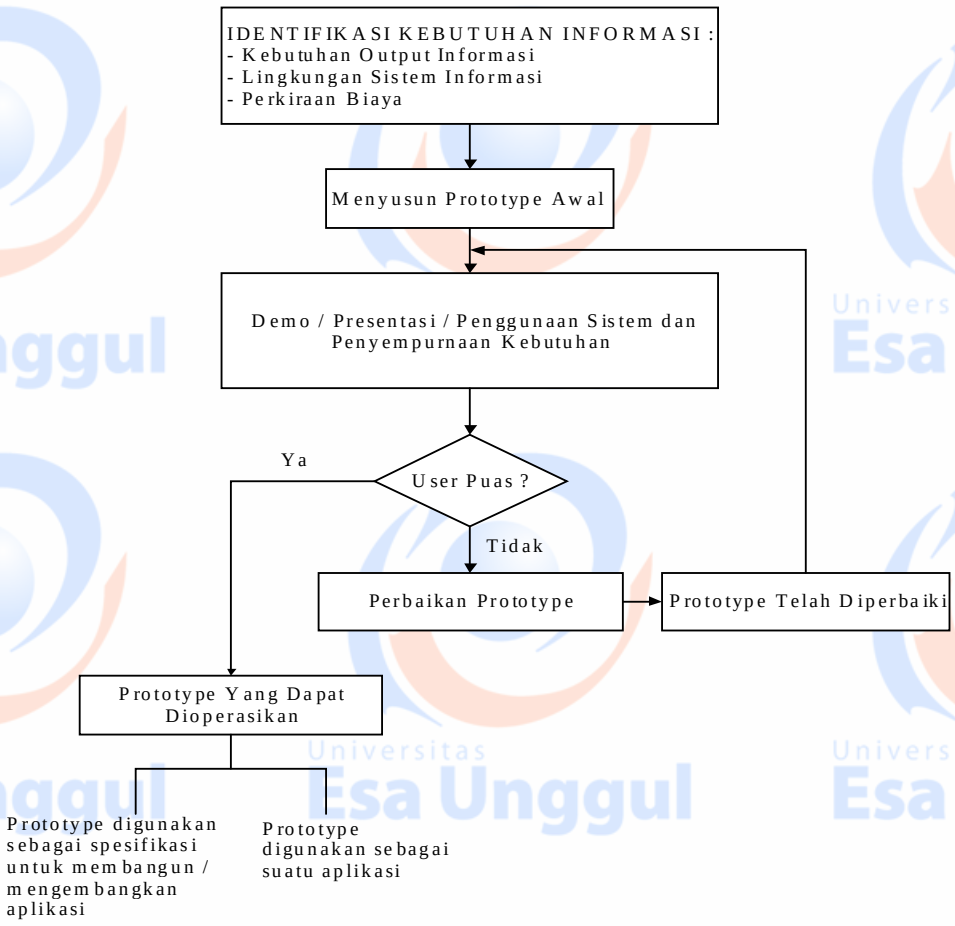
- User dapat lebih mudah untuk menilai sistem dalam bentuk model yang sudah ada, daripada dalam bentuk teori
- Langkah-langkah Pembuatan :

  - Identifikasi keperluan user dan feature-feature yang perlu
  - Buat / kembangkan prototype
  - Terapkan prototype perhatikan apa yang perlu diubah

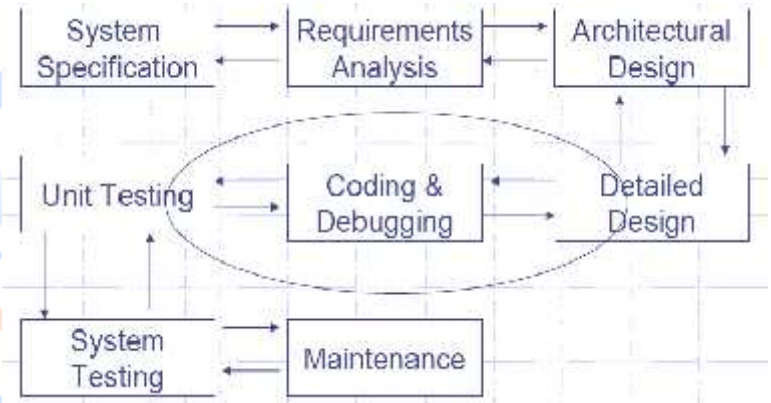
(penambahan dan pengurangan)

  - Perbaiki prototype berdasarkan informasi dari user
  - Ulangi langkah-langkah di atas (sesuai keperluan) untuk menghasilkan sistem yang memuaskan.

Bagan Prototipe



Pengembangan Perangkat Lunak



2.1.5. Terdapat 3 Pilar Perancangan sistem

1. Guidelines document, meliputi :
   -Tata letak layar
   -Piranti masukan dan keluaran
   -Urutan aksi
   - Pelatihan
2. User-Interface Software Tools (UIMS dan Rapid Prototyping Tools)
   -Pelanggan dan pemakai belum mempunyai gambaran yang jelas bagaimana sistem akhir akan terlihat
   - Kesulitan dihindari dengan pemberian kesan realistik tentang seperti apa bentuk sistem akhir
3. Usability Laboratories and Iterative Testing
   -Harus dilakukan uji pilot kecil dan besar dari komponen-komponen sistem sebelum dirilis ke pelanggan
   -Uji pilot membandingkan alternatif-alternatif rancangan, membedakan sistem baru dengan prosedur lama, atau mengevaluasi produk-produk kompetitif

2.1.6. Prinsip merancang tampilan Layar

- Organisasi Layar
- Caption dan Field Data Justification
- Headings
- Spacing
- Title dan Screen Identifier
- Warna

2.1.6.1.Organisasi Layar

1. Minimasi gerakan mata baik dari atas ke bawah, atau kiri ke kanan
1. Dua elemen kunci adalah Caption dan Field Data
2. Caption ditulis lengkap dengan huruf besar (uppercase) dan densitas normal. Huruf kecil (lowercase) dipergunakan untuk caption yang panjang
3. Field data tunggal, letakkan caption di kiri field data dan dipisahkan dengan simbol yang unik dan spasi.

   Contoh : Departemen : x Keuangan
5. Untuk field data yang tampil berulang, letakkan caption 1 baris di atas kolom field data

   Contoh : Biaya Produksi
   Bahan Langsung
   Direct Labor (Upah Langsung)
   Biaya Tidak Langsung
6. Untuk field data, teks sebaiknya left-justify dan karakter alfa numerik.

   Contoh :Nama : Amat Soleh
   Departemen : Keuangan
7. Data numerik menggunakan list yang right-justify.

   Contoh : Subtotal : $ 1973.40
   Handling : $ 200.00
   Tax : $ 47.20
   Total : $ 2220.60

8. Nomor dan bilangan ditampilkan berdampingan dalam kelompok 3, 4 atau 5 karakter.

   Contoh : 5416 7811 0895 1875

9. Daftar yang panjang, sisipkan baris diantara kelompok data tertentu.

   Contoh : Anvil ; Bracket ; Clasp ; Die

2.1.6.2. Heading

Untuk section heading, tempatkan section heading pada baris di atas field data terkait. Posisi caption minimum 5 spasi masuk kedalam terhadap posisi heading. Ditulis dalam uppercase.

Contoh:

Assets	
xxxxxCurrent:	$899,420.00
Fixed:	$1,566,748.12
Intangible:	$75,000.00

Untuk row heading, letakkan pada sebelah kiri atas field data yang bersangkutan. Pisahkan heading dengan simbol >> sebesar 1 spasi dan antara simbol >> dengan caption sebesar 3 spasi.

Contoh :

Cattle >>	xxxHereford:	5420xxxxx	Simmental:	470
	Angus:	6210	Brahman:	1800
Horsex>>	Quarter:	2214	Thoroughbred:	419

ATAU

Ranch Animals

xxxxxCattle >>	Hereford:	5420	Simmental:	470
	Angus:	6210	Brahman:	1800
Horse >>	Quarter:	2214	Thoroughbred:	419

2.1.6.3. Warna

Bila digunakan dengan tepat pemakaian warna dapat mengorganisasikan data, memfokuskan perhatian, menegaskan perbedaan, menambah daya tarik.

Untuk mendapatkan diskriminasi terbaik pilih 4 sampai dengan 6 warna dari spektrum warna, yaitu sbb: Merah, Orange, Kuning, Kuning-Hijau, Hijau, Biru-Hijau, Biru, Ungu.

Petunjuk dalam menggunakan warna :

- Untuk diskriminasi antara item, gunakan merah, kuning, hijau, biru dan putih.
- Gunakan warna cerah untuk penegasan, warna gelap untuk kurang ditegaskan. Gradasi kecerahan warna mulai dari terendah: Putih, Kuning, Hijau, Biru, Merah.
- Untuk menonjolkan perbedaan, pakai warna yang kontras, misalnya: Merah dengan hijau atau Biru dengan Kuning.
- Untuk menunjukkan persamaan, gunakan warna yang senada misalnya : Orange dengan Kuning, Biru dengan Ungu
- Gunakan tidak lebih dari 3 warna plus warna putih pada layar untuk satu saat yang sama
- Gunakan warna yang hangat untuk meminta perhatian, tindakan atau respon misalnya: Merah, Orange, Kuning
- Gunakan warna yang sejuk (Hijau, Biru-Hijau, Biru, Ungu) untuk menjelaskan status dan latar belakang informasi
- Untuk teks gunakan warna kuning atau hijau
- Untuk latar belakang layar gunakan merah, biru atau hitam. Warna Biru pilihan terbaik

Catatan

- Lakukan konfirmasi kepada user.
  Misal : - Merah : Hilang, berhenti atau berbahaya
  - Kuning : Hati-hati
  - Hijau : Lanjutkan, biasa saja
- Perubahan arti dari warna akan dapat menimbulkan kesukaran bahkan error karena
  salah arti

2.1.6.3.1 Pedoman penggunaan warna :

Gunakan warna secara konservatif

- Batasi jumlah warna
- Kenali kekuatan warna sebagai teknik pengkodean (coding)
- Pastikan bahwa color coding mendukung tugas
- Tampilkan color coding dengan usaha pemakai yang minimal
- Tempatkan color coding di bawah kendali pemakai
- Rancang untuk monokrom dulu
- Gunakan warna untuk membantu pemformatan
- Gunakan color coding yang konsisten
- Perhatikan ekspektasi umum tentang kode warna
- Perhatikan masalah pemasangan warna
- Gunakan perubahan warna untuk menunjukkan perubahan status
- Gunakan warna pada tampilan grafis untuk kerapatan informasi yang lebih tinggi
- Waspada atas kehilangan resolusi pada tampilan warna

Perpaduan Warna yang baik

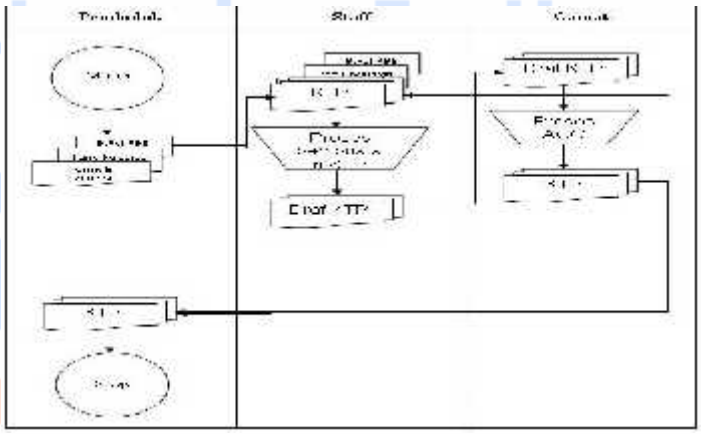
Latar Belakang	Garis dan Teks Tipis	Garis dan Teks Tebal
Putih	Biru, Hitam, Merah	Hitam, Biru, Merah
Hitam	Putih, Kuning	Kuning, Putih, Hijau
Merah	Kuning, Putih, Hitam	Hitam, Kuning, Putih, Sian
Hijau	Hitam, Biru, Merah	Hitam, Merah, Biru
Biru	Putih, Kuning, Sian	Kuning, Magenta, Hitam, Sian, Putih
Sian	Biru, Hitam, Merah	Merah, Biru, Hitam, Magenta
Magenta	Hitam, Putih, Biru	Biru, Hitam, Kuning
Kuning	Merah, Biru, Hitam	Merah, Biru, Hitam

Perpaduan Warna yang tidak baik

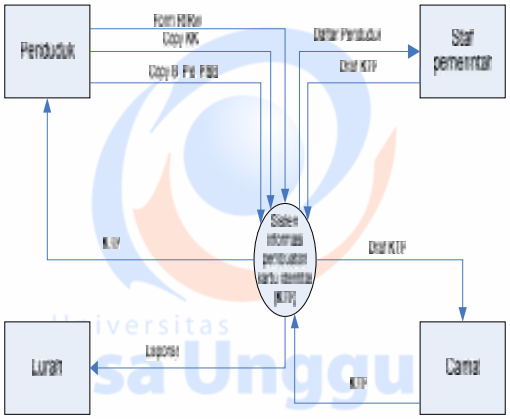
Latar Belakang	Garis dan Teks Tipis	Garis dan Teks Tebal
Putih	Kuning, Sian	Kuning, Sian
Hitam	Biru, Merah, Magenta	Biru, Magenta
Merah	Magenta, Biru, Hijau, Sian	Magenta, Biru, Hijau, Sian
Hijau	Sian, Magenta, Kuning	Sian, Magenta, Kuning
Biru	Hijau, Merah, Hitam	Hijau, Merah, Hitam
Sian	Hitam, Kuning, Putih	Kuning, Hijau, Putih
Magenta	Hijau, Merah, Sian	Sian, Hijau, Merah
Kuning	Putih, Sian	Putih, Sian, Hijau

2.1.7. Perancangan Sistem

2.1.7.1 Prosedure sistem berjalan

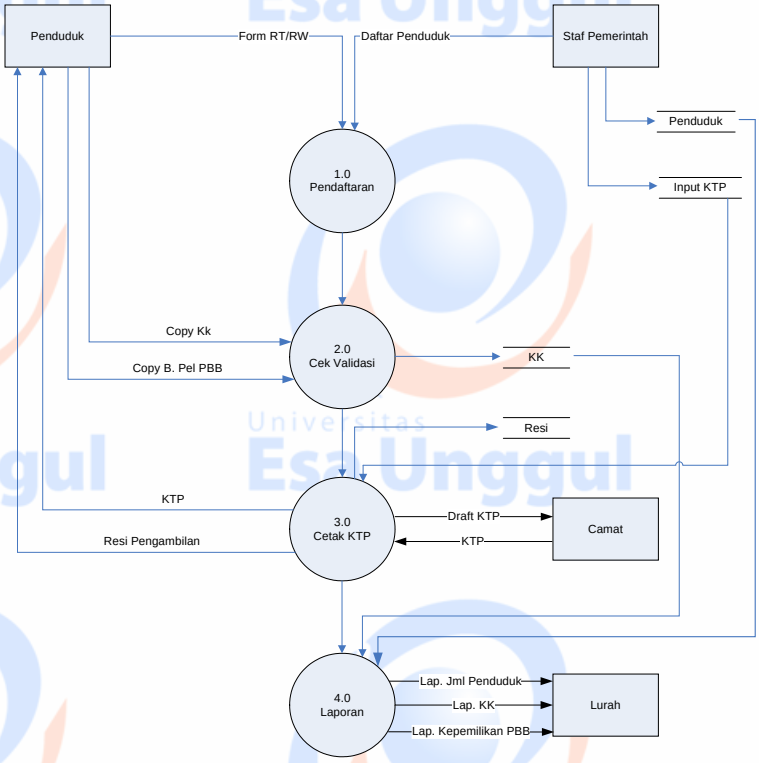


2.1.7.2. Menggunakan DFD (Data Flow Diagram)



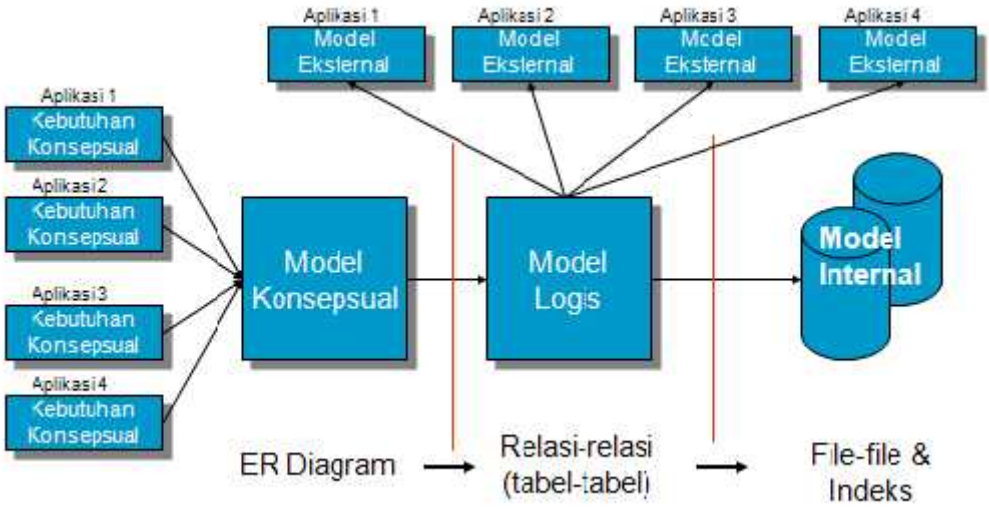
Digunakan untuk mengambarkan secara keseluruhan kebutuhan aliran data dari input-proses-output

2.1.7.4. Penggunaan DFD pada sistem yang akan dibangun



2.1.8. Perancangan Database

“Model Data merupakan bagian kecil dari spesifikasi sistem secara keseluruhan, tetapi mempunyai peranan penting dalam menentukan kualitas dan kelangsungan hidup sistem “(Coad dan Yourdon)



Mengapa Perancangan database sangat penting?

1.Mengekspresikan permintaan pemakai

2.Mengantisipasi anomali-anomali dalam Memanipulasi Data

3.Menyederhanakan pemrograman

4. Mengurangi biaya pengeluaran

5. Mempermudah pemahaman informasi

6. Meningkatkan performance

DAMPAK DARI KESALAHAN PERANCANGAN

1. Hilangnya Informasi

2.Sukarnya memodifikasi database

3.Memerlukan waktu yang lama dalam penulisan program

4.Memerlukan waktu yang lama dalam pengaksesan database

5.Menambah volume pekerjaan dalam sistem database

6.Pemborosan waktu bagi Administrator

2.1.8.2. Tujuan Perancangan Database

1. Menyajikan data dan hubungan antar data yang diperlukan oleh pemakai dan aplikasi

2. Mempermudah pemahaman informasi

3. Melengkapi model data yang mendukung transaksi-transaksi yang diperlukan

4. Mendukung proses permintaan

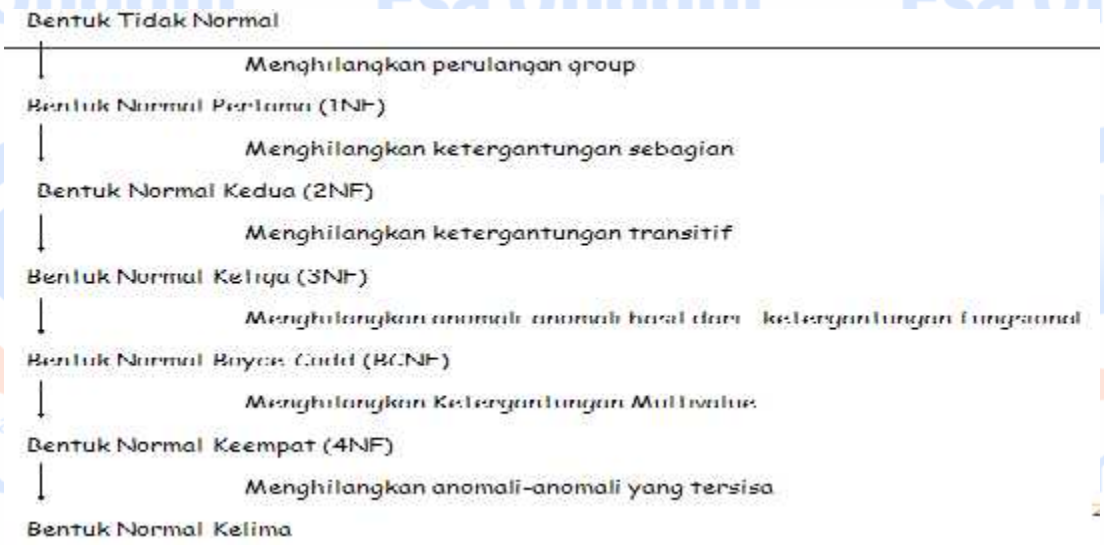
5. Meningkatkan performance

2.1.8.3 .Proses Perancangan Database Logika

2.1.8.3.1.Transformasi Data

- Transformasikan atribut-atribut dalam setiap Entity ke dalam suatu relasi
- Bila ada Atribut Multivalue buat relasi baru
- Bila Unary Degree , periksa cardinality rationya, tempatkan foreign key atau buat relasi baru
- Bila Binary Degree, periksa cardinality ratio dan participation constraintnya, tempatkan foreign keynya atau buat relasi baru
- Bila Ternary Degree buat relasi baru
- Bila Terdapat Weak Entity , Primary Keynya gabungan dari salah satu atribut dengan Primary Key dari Entity induknya

2.1.8.3.2. Normalisasi



Contoh:

PELANGGAN (NO-PLG, NM-PLG, ALAMAT, NO-TLP) 3NF

NO-PLG	NM-PLG
NO-PLG	ALAMAT
NO-PLG	NO-TLP

PESANAN (NO-PES, TGL-PES, NO-PLG) 3NF

NO-PES	TGL-PES
NO-PES	NO-PLG

BARANG (NO-BAR, NM-BAR, HARGA, JUMLAH) 3NF

NO-BAR	NM-BAR, HARGA, JUMLAH
NO-BAR	NM-BAR, HARGA, JUMLAH
NO-BAR	NM-BAR, HARGA, JUMLAH

JUMPES (NO-PES, NO-BAR, JUM-PES) 3NF

NO-PES, NO-BAR	JUM-PES

2.1.8.3.3. Penggabungan Relasi

User View 1

PEGAWAI1 (NOPEG, NAMA, ALAMAT, TELEPON)

User View 2

PEGAWAI2 (NOPEG, KOJAB)

Hasil penggabungan

PEGAWAI (NOPEG, NAMA, ALAMAT, TELEPON, KOJAB)

Kapan diperlukan Penggabungan Relasi

- Sinonim
- Homonim
- Ketergantungan Transitif
- Relasi Kelas/ subkelas

SINONIM

- Dua atau lebih atribut dapat mempunyai nama yang berbeda tetapi arti yang sama.
- SISWA1 (SISWA-ID, NAMA)
- SISWA2 (NO-SISWA, NAMA, ALAMAT)
- Penggabungan
- SISWA (NO-SISWA, NAMA, ALAMAT)

KETERGANTUNGAN TRANSITIF

- Dua relasi 3NF digabungkan menjadi satu relasi dapat menghasilkan relasi yang atributnya mempunyai ketergantugan transitif.

SISWA1 (NO-SISWA, JURUSAN) 3NF

SISWA2 (NO-SISWA, PEMBIMBING) 3NF

Penggabungan

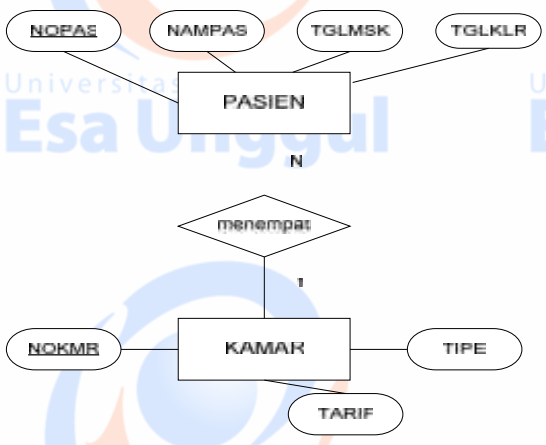
SISWA (NO-SISWA, JURUSAN, PEMBIMBING) 2NF

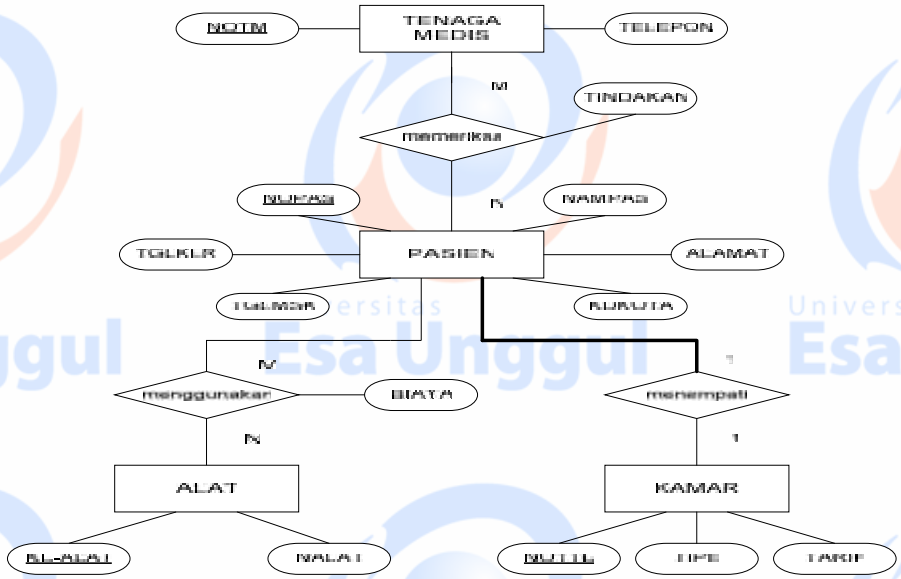
JURUSAN PEMBIMBING

SISWA (NO-SISWA, JURUSAN) 3NF

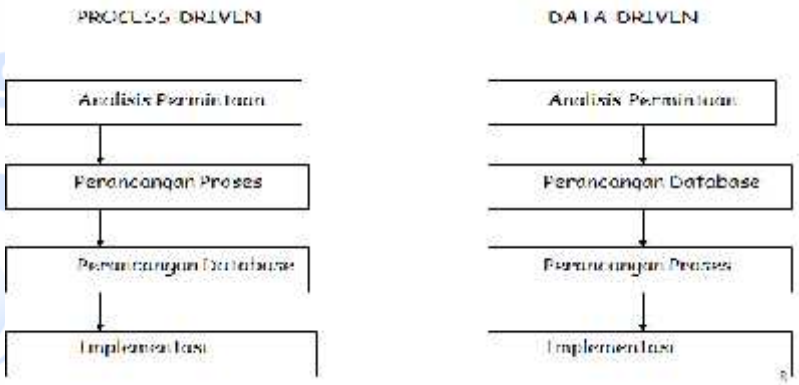
PEMBIMBING (JURUSAN, PEMBIMBING) 3NF

2.1.8.3.4. Penggunaan ERD





2.1.8.4 Perancangan Proses VS Perancangan Database.



PENGUJIAN DAN IMPLEMENTASI

3.1. Implementasi Sebuah Sistem

Dalam proses akhir dari pengembangan sebuah sistem yaitu melakukan implementasi dengan tahapan sebagai berikut:

1.Training Personal

2. Konversi Sistem

3. Review post-implementation

4. Dokumentasi

5. Dukungan lain

3.1. a. Tipe Training: disesuaikan dengan keadaan lingkungan implementasi sistem:

-Eksternal Training: kursus dr vendor pelatihan

- Internal Training: On the Job Training
- Belajar Sendiri

3.1. b. Memilih Staf yang akan dilatih

- Terdiri dari tiga kelompok User:

- Teknisi & Administrator yang akan bertugas merawat sistem
- Application user (user pd umumnya)
- General Manager

3.1.c. Konversi terdapat 4 macam:

- Konversi Langsung
- Konversi Paralel
- Konversi phase-in
- Konversi Pilot

3.1.d. Evaluasi Sistem

Bertujuan untuk mendapatkan cara meningkatkan efisiensi dan efektifitas sistem baru, serta memberikan informasi untuk pengembangan sistem mendatang.

Biasanya dilakukan setelah dua hingga enam bulan instalasi -> sudah terdapat periodik pelaporan serta isu sistem masih baru

3.1.2.a. Area Peninjauan Evaluasi Sistem (Faktor Sistem)

- Faktor kelayakan TELOS (Technical, Economical, Legality, Operation, Schedule)

Teknologi pendukung;

- Pendanaan utk biaya teknologi, operasional, pemeliharaan,
- Operasinal sistem yg tidak melanggar hukum, kesesuain jadwal.

-Keterampilan personal dlm Faktor strategik PDM (Productivity, Differentiation, Management)

- Sudah tercapainya produktivitas setelah implementasi sistem baru
- Kontribusi terhadap diferensiasi produk dan layanan
- Dukungan informasi utk peningkatan kualitas perencanaan, pengontrolan, dan pembuatan keputusan manajemen,

-Faktor rancangan MURRE (Maintenability, Usability, reusability, extenability)

- Dokumentasi sudah komprehensive, jelas dan uptodate
- Mendukung CMS (Change Management System)
- Module untuk user sudah terpenuhi
- Terpenuhinya dukungan ke user
- Modul perangkat lunak dapat digunakan kembali
- Terbebas dari fault/error
- Adaptif dan fleksibel

3.1.2.b. Komponen Rancangan Sistem:

Output

- Output harus sesuai, relevan, akurat dan dapat digunakan kembali
- Kemanan pengguna output bagi user yg tidak berhak
- Kemudahan akses
- Sesuai dengan kognitif user
- Ketepatan edit dan identifikasi laporan

Input

- Form memenuhi kaidah pedoman dan spesifikasi rancangan
- Verifikator data input
- Manual pengisian form

Proses

- Pengujian terhadap semua proses
- Peninjauan terhadap prosedur dan dokumentasi
- Prosedure pengoperasian
- Reliability sistem
- Platform Teknologi
- Peripheral, workstation, processor, dan jaringan,
- Membandingkan kinerja dengan rancangan

Membutuhkan komponen penunjang:

- Job accounting system
- Hardware monitoring
- Software monitoring
- Konfigurasi teknologi yang optimal
- Respon time yang acceptable
- Keakuratan Estimasi
- Waktu: menggunakan tool spt PERT, Gannt, atau lainnya.
- Biaya yg sebenarnya sesuai dg yg diestimasi

Manfaat

- Tingkat dukungan:
- Sumber daya tersedia
- Manajemen puncak
- Pelatihan (teknik pelatihan, user, tutor)

3.1.2.c. Dokumentasi Program

Dokumentasi Proses

Merekam proses pengembangan dan perawatan. Isi dokumentasi proses adalah: perencanaan, jadwal, kualitas proses, standard proses.

Dokumentasi Produk

Menggambarkan produk yang sedang dikembangkan dr sudut pandang engineer pengembang/perawatan. Yg termasuk dokumentasi produk adalah: user manual, help.

Dokumentasi Produk

Technical manual untuk hardware dan pengoperasion sistem dengan kualitas penulisan yang baik dan jelas. Fitur yang harus terdapat dalam user manual:

Bagian “Bagaimana Memulai” Indeks yang komprehensif Tutorial

Contoh-contoh

- Quick reference guide
- Referensi Ringkas yg menunjukkan fitur dan routin
- Ilustrasi
- Manual yg mudah diikuti, mudah dimengerti dan terurut secara logis
- User Manual

1.Pengantar

- Tujuan dari produk
- Lingkungan operasi
- Fungsi secara umum
- Fitur Khusus
- Keterbatasan
- Keterangan dan notasi dokumen serta terminologi

2. Instalasi

- Persyaratan minimal sistem yang dibutuhkan
- Menyalin dan memback-up
- Proses instalasi
- Konfigurasi/kustomisasi produk

3. Tutorial

Penjelasan langkah-demi langkah dengan contoh

- Penjelasan tiap contoh
- Pengembangan dari contoh dasar
- Penggunaan on-line Help

4. Instruksi detail

- Keluaran dari produk
- Masukan untuk produk
- Pengoperasian produk
- Penanganan error
- Fungsi khusus

5. Detail Teknis

- Prinsip dari operasi
- Fitur lanjutan
- Algoritma utama yang digunakan
- Struktur data utama
- Modifikasi produk

Cara memperoleh dukungan teknis dan informasi lanjutan

- User Manual Source Code
- Penamaan variable, constant, procedure. Function yang jelas dan konsisten
- Memberi keterangan pada header setiap procedure, yang berisi:
- Fungsi dari procedure
- Variable local masukan, dan keluaran
- Variable global yang digunakan dan yang dipengaruhi.
- Pada Header Program diberi:

  Nama penulis program

  Editor

  Compiler dan Library yang digunakan

  Versi dan upgrade history

  Tanggal pembuatan software

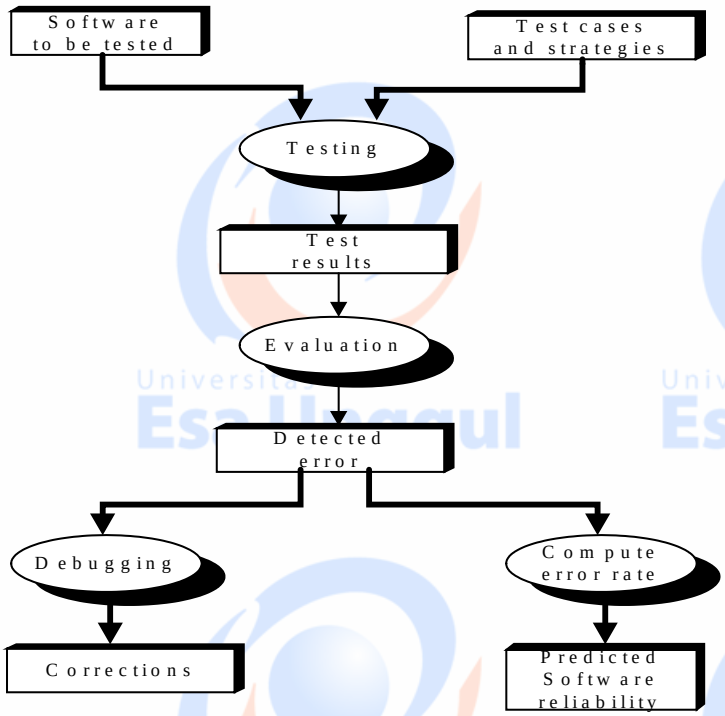
  Deskripsi singkat tentang software

3.1. MENGUJI SISTEM INFORMASI

3.1.1 Tujuan melakukan pengujian

- Pengujian perangkat lunak adalah proses yang harus mengikuti pola tertentu dan dirancang secara matang. Proses ini dilakukan oleh Team yang bertanggang jawab atas kuantitas dan keandalan yang tinggi dari perangkat lunak yang dibuat setelah dapat menemukan dan memperbaiki error
- Tujuannya adalah untuk menghasilkan perangkat lunak yang handal. Tetapi tidak berarti error hilang sama sekali. Selalu masih terdapat kemungkinan tidak terdeteksinya error meskipun telah dilakukan pengujian secara teliti dan seksama

3.1.2. Proses Pengujian



Area yang bisa dikenal oleh uji kasus adalah :

- Field
- Record
- File
- Data entry
- Control
- Aliran program

Jenis-jenis error dan kaitannya dengan keandalan

Fatal error :

- Crash
- Logic
- Hang

♦ Serious error:

- Menghasilkan keluaran yang tidak benar

♦ Minor error

- User tidak puas terhadap hasil program

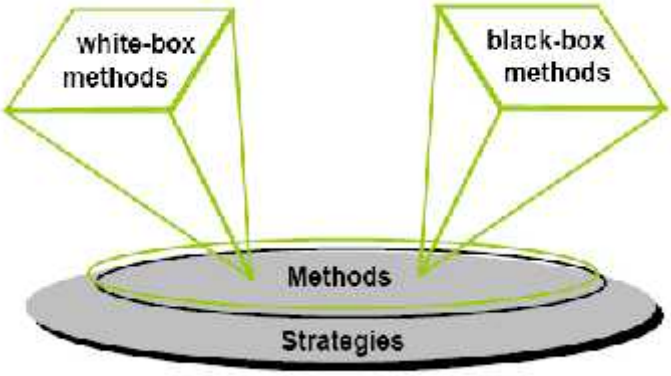
Penyebab error disebut Bug. Bila fatal error ditemukan maka keandalan program dipertanyakan, kemungkinan besar harus dirancang dan diprogram ulang, Bisa jadi dihancurkan sama sekali

- Minor error dapat dengan mudah diatasi, dengan kesimpulan :
  - Keandalan program cukup baik
  - Uji kasus tidak memadai sehingga tidak dapat mendeteksi fatal dan serious error

- **Debugging**

Adalah kegiatan menghilangkan Bug, yang memang harus dilakukan justru karena keberhasilan pengujian. Debugging mencoba untuk mencocokan antara simptom dengan problem, sehingga dapat dilakukan koreksi error secara menyeluruh

3.1.3. Jenis-jenis pengujian



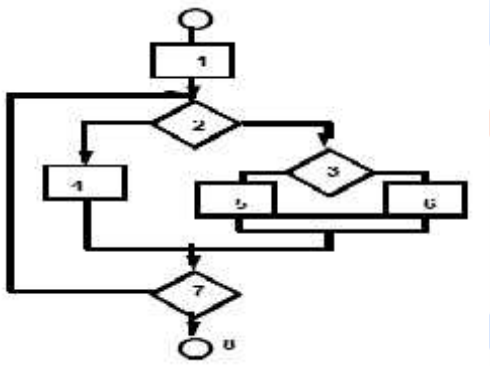
1.White Box Testing

- Mengetahui cara kerja dari program tersebut (terstruktur logik), sehingga dapat mengembangkan uji fungsional program yang efisien dan efektif.
- Lakukan pengujian pada sekelompok program yang berstruktur SEQUENCE, IF-THEN-ELSE, DO-WHILE dan DO-UNTIL
- Perlu diuji atas command dan percabangan berikut ini:
  - Select
  - Open/Close

- Copy replacing
- IF
- Perform until dan perform while
- Call

Konsep Pengujian Basis Path (1)

- Merupakan bagian dari pengujian White Box dalam hal pengujian prosedur-prosedure
- Mempergunakan notasi aliaran grapt (node,link unutk merepresentasikan sequence, if, while, until);
- Konsep kompleksitas cyclomatic antar lain cara perhitungan daerah tertutup pada graph planar dimana dapat menhubungkan batas atas jumlah pengujian yang harus direncanakan dan dieksekusi unutk menjamin pengujian seluruh statement program
- Memunculkan kasus-kasus yang akan diuji dengan membuat daftar lintasan kasus pengujian berdasarkan kompleksitas & cyclomatic yang didapat
- Membuat alat bantu matrik graph untuk membantu pengawasan pengujian.



2.Black Box Testing

- Mendemontrasikan bahwa fungsi perangkat lunak berjalan sebagaimana mestinya, keluaran yang dihasilkan sesuai dengan masukannya, database diakses secara benar dan diremajakan.
- Kasus yang digunakan terdiri dari satu kumpulan kondisi masukan baik yang valid maupun yang invalid. Baik whitebox maupun black box berusaha mengungkap error selama pemrograman, tetapi khusus black box yang fokus pada pengungkapan error yang terjadi saat implementasi user requirement dan spesifikasi rancangan sistem.

- Karena itu black box testing sesuai untuk pengujian saat integrasi, sistem dan serah terima (acceptance) dibandingkan uji modul

3.1.3.1 UJI KLAS EKIVALEN

- Uji jenis ini merupakan kunci dari black box testing
- Contoh :

  - Suatu data hanya bisa diisi oleh angka 1 sampai dengan 50. maka uji dengan angka 1, 50 (batas dari klas), angka negatif, 0, dan angka > 50
  - Uji apakah control total telah dibuat dengan benar
  - Uji memproses transaksi yang peka tanpa otorisasi, apakah sistem akan menolak?
  - Lakukan uji numerik alpabet dan karakter khusus
  - Masukan nilai dengan tanda negatif, apakah sistem mampu menanganinya.
  - Bagi total atau jumlah dengan nol
  - Lakukan validitas untuk field data kunci
  - Lakukan pemeriksaan range dan reasonableness
  - Periksa urutan transaksi dengan benar
  - Masukan beberapa field dengan data yang tidak lengkap atau hilang
  - Coba untuk baca dan tulis pada file yang salah
  - Masukan nomor rekening dengan check digit yang telah ditentukan, periksa apakah sistem memproses dengan benar?

Tindakan yang harus dilakukan bila error telah dideteksi ?

- Laporan error dan Penjelasan
- Jenis Laporan :
  - Suggestion : Bukan berarti salah, tetapi ada yang bisa diperbaiki
  - Design Error : Biasanya berupa error pada antar muka dengan user
  - Coding Error : Kesalahan yang tidak sengaja dilakukan pemrograman
  - Documentation Error : Source code dan dokumentasi tidak sesuai
  - Querry : Program melakukan sesuatu yang pengujiannya tidak faham maksudnya, perlu klasifikasi lanjutan dengan pemrograman
  - Attachments : Berupa disk berisi test data, hasil cetakan, catatan atas pelaksanaan uji, dan sebagainya

3.1.3.2. Tahapan Pengujian

1. Module Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

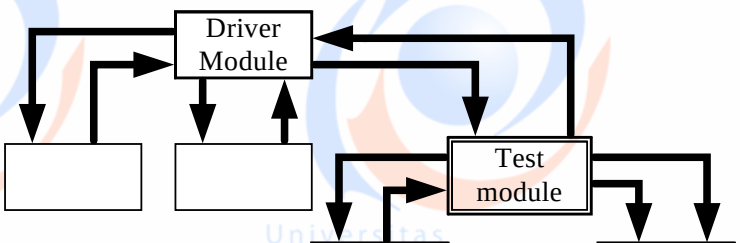
Module Testing

- Adalah proses pengujian unit terkecil dari seluruh program sebelum disatukan bersama menjadi program utuh. Merupakan bagian dari whitebox testing
- Tujuan module testing adalah:
    - Pelaksanaan setiap command pada sebuah modul.
    - Menelusuri setiap path logika modul. Hitung ulang semua command komputasi
    - Uji setiap modul dengan sekumpulan data input
- Karena tidak semua pengujian dapat dilakukan, maka konsentrasi ditujukan pada area yang memiliki resiko terbesar. Modul-modul di plot pada kisi yang bisa menggambarkan besarnya peluang error dan tingkatan pengaruhnya

Integration Testing

- Dilakukan dengan menggabungkan modul dengan modul lainnya fokusnya pada hirarki dari modul, khususnya antar-muka antar modul.
- Alasan perlunya pengujian jenis ini :
    - Data dapat hilang pada saat melewati antar muka
    - Fungsi tidak berjalan sebagaimana mestinya ketika digabungkan
    - Modul yang satu ternyata memberikan pengaruh yang kurang baik pada modul lainnya

- Karena modul bukan stand-alone program, maka diperlukan Stub dan Driver



Cara pengujian dengan Stub dan Driver

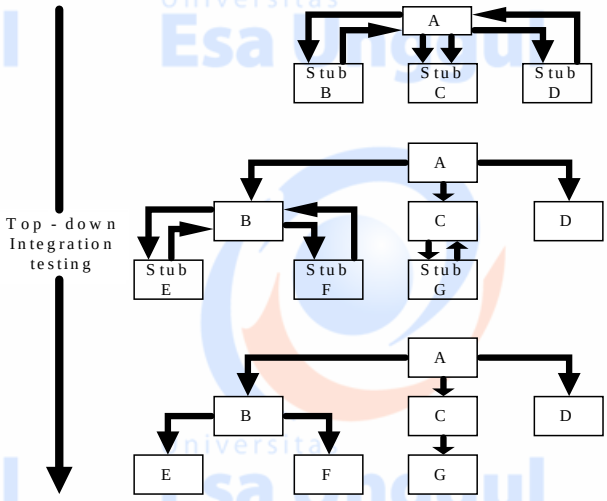
- Top Down

Penyerahan modul tingkat atas dan stub sebagai basis pengujian. Modul diintegrasikan dengan pergerakan ke bawah melalui kendali hirarki, dimulai melalui modul eksekutif utama. Setelah modul selesai ditulis untuk setiap tingkatannya, Sub diganti dengan modul sebenarnya. Demikian seterusnya hingga semua modul selesai ditulis. Program secara keseluruhan dapat diuji dengan baik.

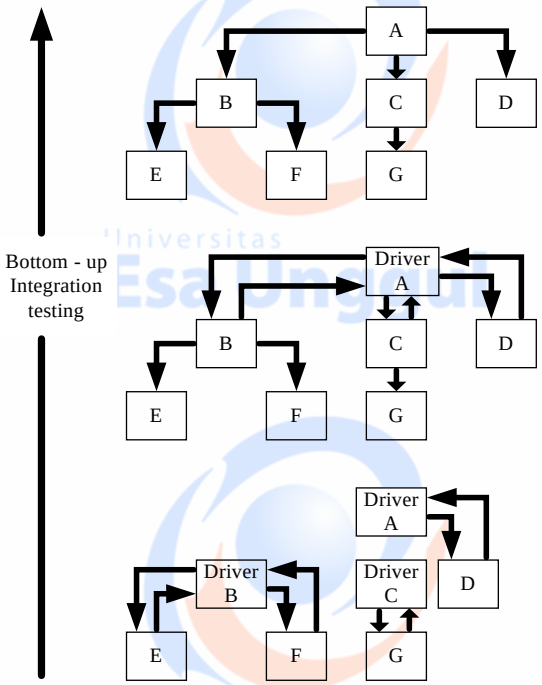
- Bottom Up

Modul tingkat paling bawah pada diagram terstruktur ditulis terlebih dulu dan diuji melalui driver yang memadai. Bersandar pada penyelesaian modul paling bawah dan driver sebagai basis pengujian integrasi. Demikian seterusnya hingga mencapai modul eksekutif utama

TOP-DOWN



BOTTOM DOWN



System Testing

- Adalah proses pengujian peng-gabungan perangkat lunak yang akan mendukung fungsi sistem secara keseluruhan
- System Testing meliputi :
  - Recovery Testing
  - Security Testing
  - Stress Testing

Acceptance Testing

- Evaluasi dari sistem untuk menguji apakah sudah dapat memenuhi user requirement sesuai dengan kondisi operasionalnya

Ada 2 jenis pengujian :

a. Alpha Testing

Dilakukan pada lingkungan operasional alami dengan diawasi oleh profesional sistem untuk dicatat error dan masalah-masalah yang muncul

Dua teknik yang dapat dipakai :

- Usability Labs
- Usability Factors Checklist

b. Beta Testing

Sama dengan alpha, tetapi tanpa pengawasan dari profesional user mencatat se-mua masalah (baik yang riil maupun imaginasi) dan melaporkan pada pemrograman sistem secara periodik Modifikasi dibuat agar sistem dapat siap dipakai pada implementasi secara penuh