Muryan Awaludin

- SDN 09 Petarukan Pemalang (1997)
- SMP PGRI 5 Petarukan Pemalang (2000)
- SMK ISLAM Pemalang (2003)
- S.Kom di STIKOM CKI Jakarta (2010)
- M.Kom di STMIK ERESHA Jakarta (2014)

KONTAK

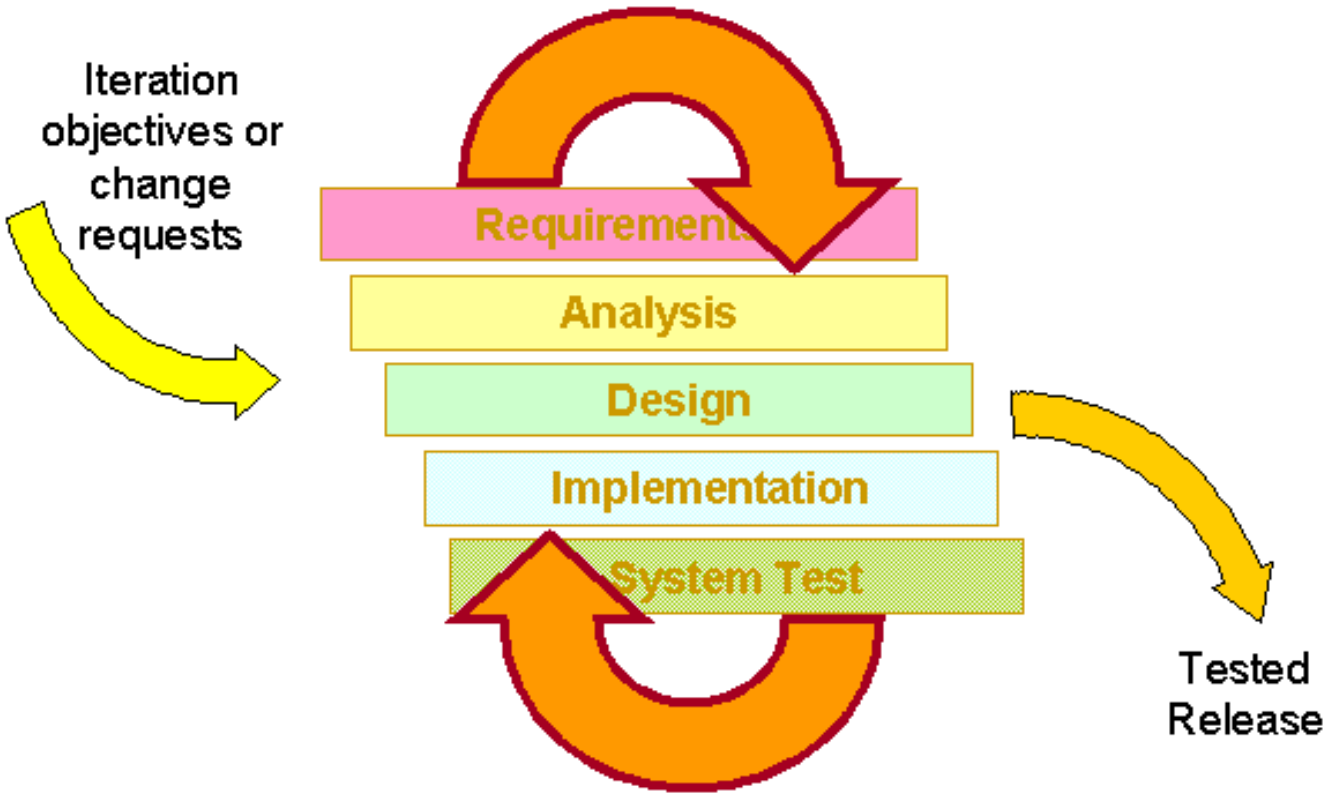
- Phone : 08562616116
- Email : muryan_awaludin@yahoo.co.id
- Blog : www.ilmudesaingrafis.blogspot.com
  : www.muryanawaludin.blogspot.com
- Twitter : @muryan_awaludin
- FB : muryan.awaludin
- Ym : muryan_awaludin

Fase Pengembangan

1. Requirements
2. Desain Sistem Software
3. Pengembangan Software
4. Quality Assurance (QA)
5. Dokumentasi

FASE PENGEMBANGAN

1. REQUIREMENTS



2. DESAIN SISTEM DAN SOFTWARE

2.1 Spesifikasi Fungsional dan Teknis

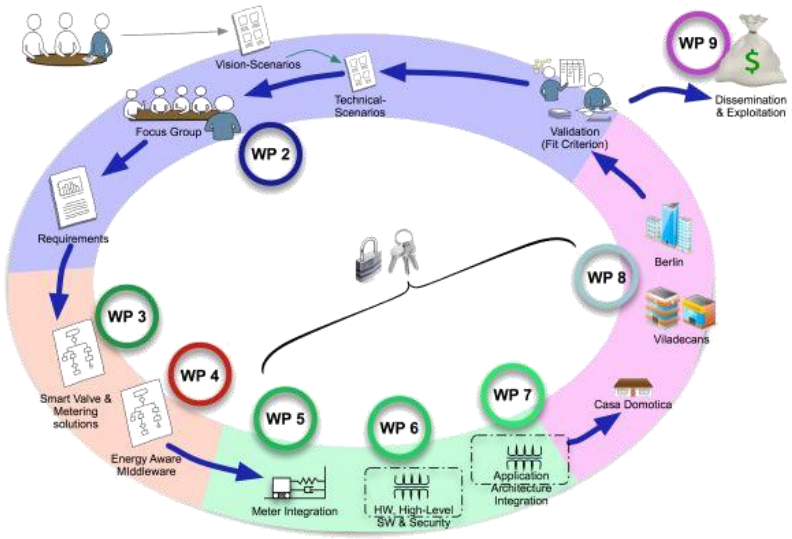
Spesifikasi fungsional bisa dideskripsikan melalui *Functional Specification Document* (FSD)

Dokumen ini menggambarkan bagaimana software akan berfungsi agar dapat memenuhi *requirements*

2. DESAIN SISTEM DAN SOFTWARE

2.1 Spesifikasi Fungsional dan Teknis

Spesifikasi teknis dalam desain sistem dan software dituangkan melalui *Software Technical Specification* (STS)



Reference image from www.sensingcontrol.com

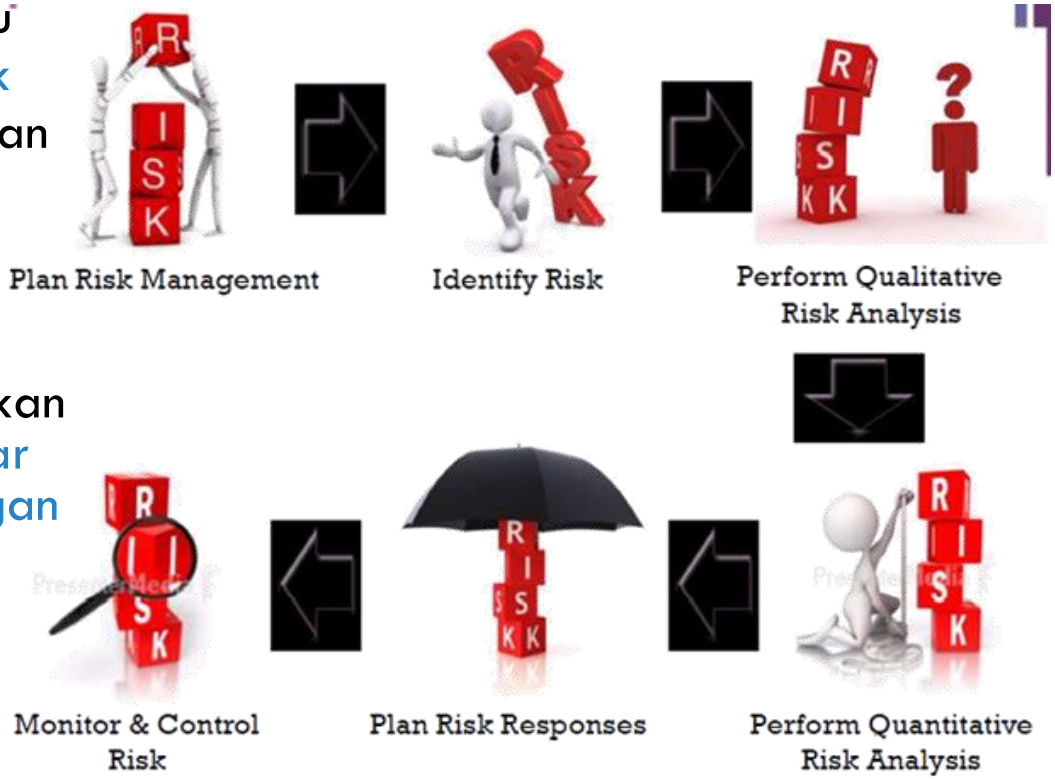
Dokumen ini menggambarkan aspek teknis dari software yang hendak didesain, seperti *platform* yang digunakan, bahasa pemograman, arsitektur, database dan struktur data

2. DESAIN SISTEM DAN SOFTWARE

2.2 Risiko dan Mitigasi

Manajemen risiko harus selalu dijadikan bagian dari proyek (seperti yang sudah dijabarkan pada fase inisialisasi)

Mitigasi dalam desain lebih ditunjukan untuk mempersiapkan solusi dan cara alternatif agar hambatan tahap pegembangan dapat dihindari

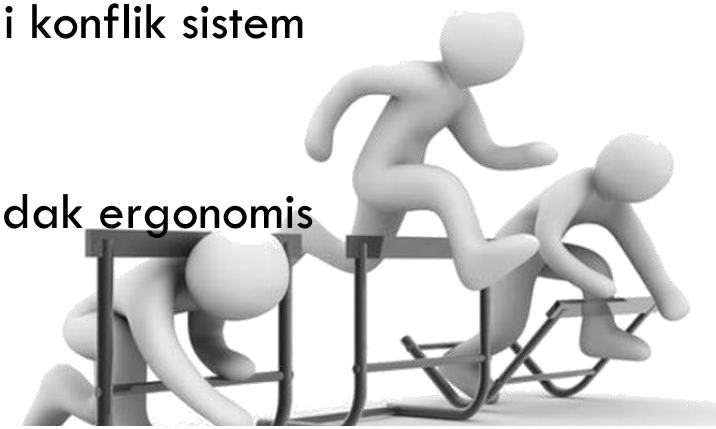


2. DESAIN SISTEM DAN SOFTWARE

2.2 Risiko dan Mitigasi

Hambatan untuk mewujudkan desain menjadi sebuah software:

1. Terlalu kompleks
2. Membutuhkan teknologi tertentu sehingga sulit diperoleh
3. Fungsi yang dibutuhkan tidak didukung oleh bahasa *tools* yang digunakan
4. Fungsi yang tidak *compatible* sehingga terjadi konflik sistem
5. Sistem tidak stabil karena desain ambigu
6. Menyulitkan pengguna akhir karena desain tidak ergonomis



2. DESAIN SISTEM DAN SOFTWARE

2.3 Desain Sistem

What is

System Design

?

2. DESAIN SISTEM DAN SOFTWARE

2.3 Desain Sistem

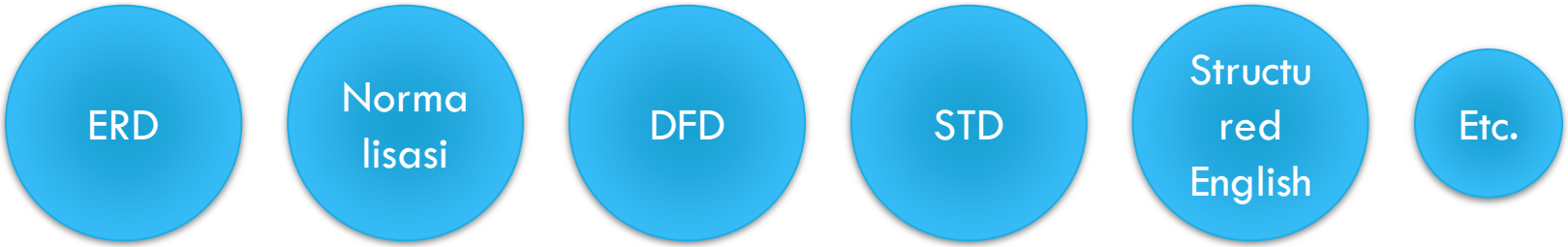
Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements (Bentley, Lonnie D., Kevin C. Dittman, and Jeffrey L. Whitten, 2004)

2. DESAIN SISTEM DAN SOFTWARE

2.3 Desain Sistem

Ada beberapa metode untuk melakukan desain sistem:

(1) 1970-1990-an: ***Yourdon System Method* (YSM),** yang diklaim sebagai metode generasi ketiga (Erdward Yourdon). Metode ini menggunakan ***content diagram*** yang menggambarkan hasil analisis dalam bentuk sumber data, aliran dan batasan-batasan sistem



2. DESAIN SISTEM DAN SOFTWARE

2.3 Desain Sistem

(2) ***Object Oriented System Analysis and Design Method*** --> **UML** (*Unified Modelling Language*)

Ada 2 jenis desain dalam melakukan desain sistem:

1. Desain Logikal

Melakukan representasi abstrak sistem dengan diagram, gambar dan bagan, sehingga aliran data, input dan output mudah dipahami oleh *developer*

- **Fungsi:** apa yang dilakukan sistem
- **Waktu:** apa yang terjadi dan kapan
- **Informasi: informasi** apa yang digunakan oleh sistem

2. DESAIN SISTEM DAN SOFTWARE

2.3 Desain Sistem

2. Desain Fisikal

Desain Fisikal adalah desain yang berkaitan dengan bagaimana proses input dan output sebenarnya dalam sistem. Pertimbangannya adalah bagaimana data di-input dalam sistem, bagaimana melakukan verifikasi dan otentifikasi data, bagaimana data diproses dan output yang dihasilkan

2. DESAIN SISTEM DAN SOFTWARE

2.4 Pemodelan (Modeling)

What is Modeling ?

2. DESAIN SISTEM DAN SOFTWARE

2.4 Pemodelan (Modeling)

Representasi abstrak dari sistem yang akan dibangun

2. DESAIN SISTEM DAN SOFTWARE

2.4 Pemodelan (Modeling)

Mengapa membangun sistem informasi memerlukan pemodelan?

2. DESAIN SISTEM DAN SOFTWARE

2.4 Pemodelan (Modeling)

Dengan melakukan pemodelan maka *developer* dapat:

1. Memahami sistem yang membutuhkan contoh perwujudannya dengan mendefinisikan proses-proses bentuk model
2. Memahami bagaimana suatu proses bekerja dan asumsi-asumsi yang menjadi dasar bagaimana proses kerja tersebut
3. Memudahkan untuk mendefinisikan sumber input sistem
4. Menentukan representasi dari keterhubungan antara komponen sistem
5. *Review requirements*

2. DESAIN SISTEM DAN SOFTWARE

2.5 Desain Software

Desain Software?

2. DESAIN SISTEM DAN SOFTWARE

2.5 Desain Software

Bentuk lebih detail dari desain sistem, dimana sistem yang telah dibuat modelnya lebih terperinci dalam bentuk yang lebih dipahami oleh developer

2. DESAIN SISTEM DAN SOFTWARE

2.5 Desain Software

Desain software harus memenuhi beberapa konsep dasar:

1 Abstraksi	2 Coupling	3 Cohesion	4 Dekomposisi dan Modularisasi	5 Encapsulation	6 Pemisahan antarmuka dan implementasi
Proses atau hasil dari generalisasi suatu permasalahan	Tingkat relasi antara satu modul dengan modul lainnya, antara kompunen ataupun servis	Tingkat indeoendensi antar modul, komponen ataupun servis	Memecahkan satuan atau entitas software yang besar menjadi beberapa komponen yang independen	Menutupi detail internal dari abstraksi, sehingga detail tersebut tidak dapat diakses	Mendefiniskan komponen publik yang dapat diakses oleh pengguna dan yang tidak dapat diakses

2. DESAIN SISTEM DAN SOFTWARE

2.6 Unified Modeling Language (UML)

What is UML?

2. DESAIN SISTEM DAN SOFTWARE

2.6 Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system

(James Rumbaugh, Ivar Jacobson, Grady Booch, 2004)

2. DESAIN SISTEM DAN SOFTWARE

2.6 Unified Modeling Language (UML)

Ada 9 Jenis diagram dalam UML:

1. Diagram *Use case*: menunjukan sekelompok *use case* dan bagaimana pelaku (*aktor)* dapat menggunakanannya
2. Diagram *Class*: meggambarkan struktur sistem dan membagi dalam kelas dengan koneksi atau relasi yang berbeda
3. Diagram *Sequences*: menunjukjan interaksi antara sekelompok objek melalui pesan yang dapat dikirimkan antara objek tersebut
4. Diagram *State Chart*: *state machines* yang terdiri atas *states, transitions, events* dan aktivitas
5. Diagram Aktivitas: menggambarkan aliran yang melalui program, dari titik mulai yagn sudah didefinisikan sebelumnya, sampai ke titik akhir

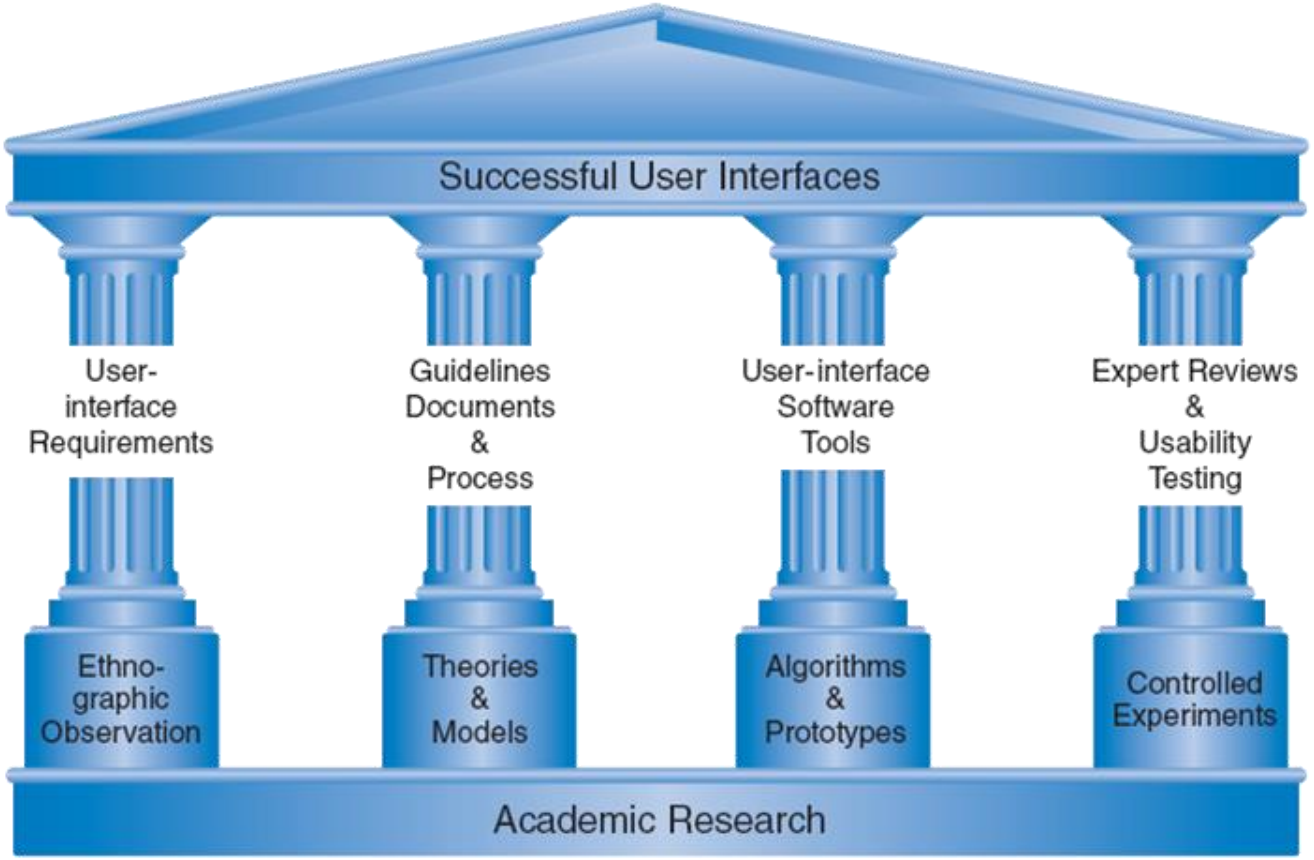
2. DESAIN SISTEM DAN SOFTWARE

2.6 Unified Modeling Language (UML)

6. Diagram Objek: sekelompok objek dan relasi-relasinya yang merupakan potongan kejadian yang menyangkut benda-benda yang ada dalam diagram *class*
7. Diagram Kolaborasi: menekankan pada urutan struktural dan objek-objek yang mengirim dan menerima pesan
8. Diagram Komponen: menunjukkan organisasi dan ketergantungan diantara sekelompok komponen
9. Diagram *Deployment*: menunjukan konfigurasi titik-titik pemrosesan saat sistem berjalan dan komponen yang berfungsi di dalamnya

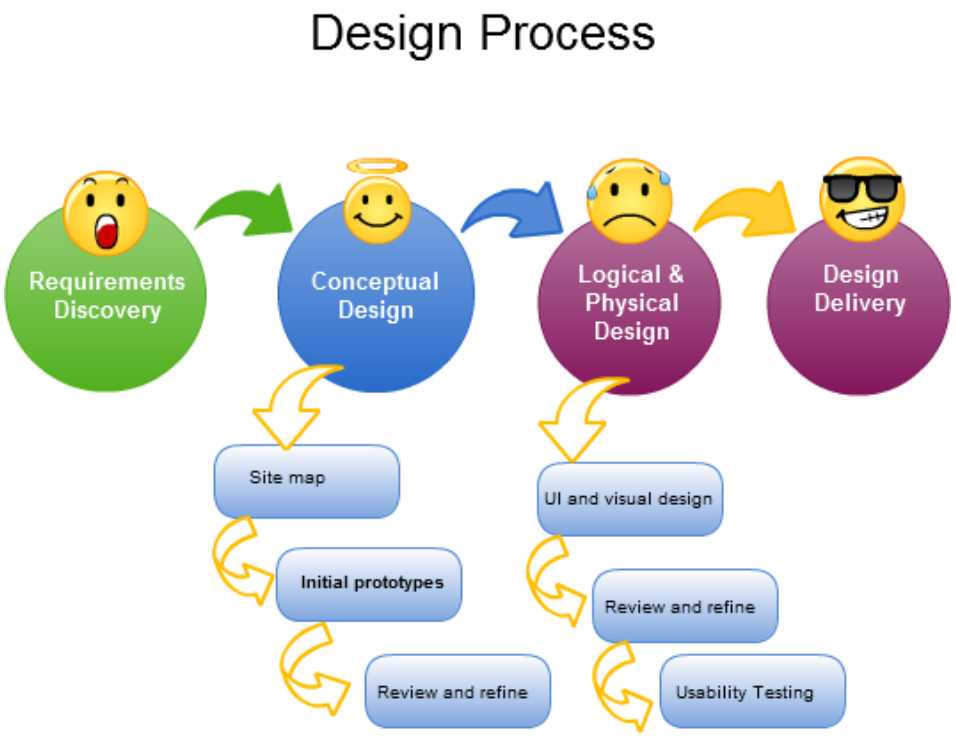
2. DESAIN SISTEM DAN SOFTWARE

2.7 Desain Antarmuka Pengguna



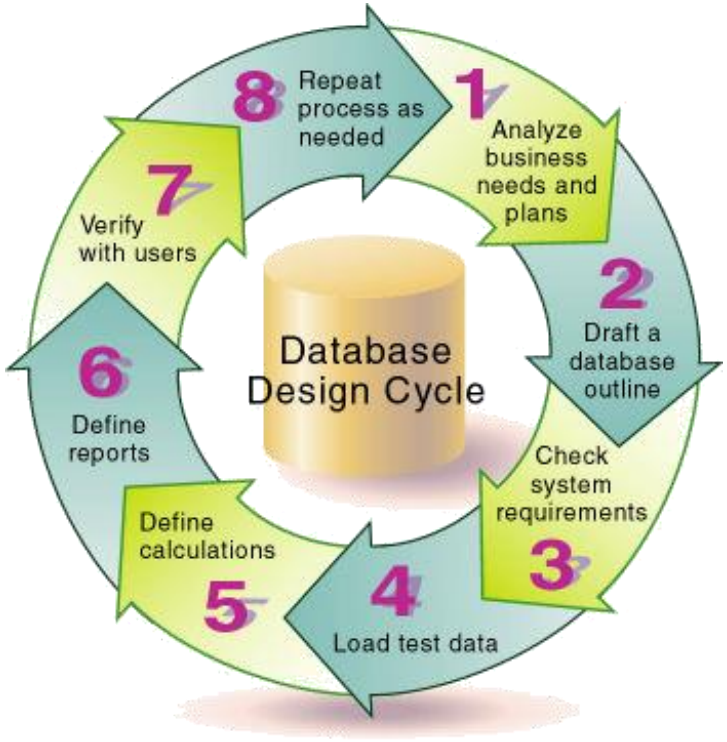
2. DESAIN SISTEM DAN SOFTWARE

2.7 Desain Antarmuka Pengguna



2. DESAIN SISTEM DAN SOFTWARE

2.8 Desain Database



Secara definisi, desain database adalah proses untuk menghasilkan model data dari database

Desain database yang baik akan dapat:

1. Membagi informasi menjadi tabel-tabel yang berdasarkan subyek untuk mengurangi redudansi data
2. Tabel-tabel kemudian dapat dihubungkan sesuai keperluan
3. Memberikan dukungan dan memastikan akurasi dan integritas informasi
4. Memiliki kemampuan untuk melakukan pemrosesan data dan keperluan laporan

3. PEMBANGUNAN SOFTWARE

3.1 Pemograman

Aktivitas pemograman utamanya melibatkan *programmer* yang menerima *deliverables* dari proses desain untuk diwujud-kan dalam bentuk software

3. PEMBANGUNAN SOFTWARE

3.1 Pemograman

3. PEMBANGUNAN SOFTWARE

3.2 Integrasi Sistem

3. PEMBANGUNAN SOFTWARE

3.3 Debugging

Istilah debugging digunakan untuk mendeskripsikan aktivitas yang dilakukan oleh *programmer* untuk menemukan masalah ataupun potensi masalah dalam program yang dibuatnya

3. PEMBANGUNAN SOFTWARE

3.3 Debugging

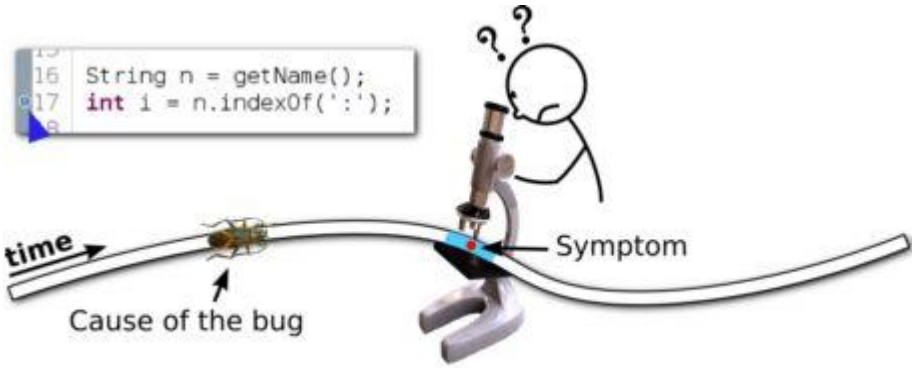
What is *bug*?

3. PEMBANGUNAN SOFTWARE

3.3 Debugging

Bug adalah kesalahan dalam program baik yang secara logika maupun secara teknis.

Bug logika bisa berasal dari desain maupun saat pemograman dilakukan



3. PEMBANGUNAN SOFTWARE

3.3 Debugging

Beberapa hal yang bisa menyebabkan kesalahan teknis:

1. *Programmer* menggunakan *command* yang salah
2. *Programmer* menggunakan *command* yang sudah tidak didukung oleh versi bahasa progam yang digunakannya
3. *Programmer* menggunakan tipe data yang tidak sesuai sehingga data yang digunakan dalam proses menjadi tidak valid
4. *Programmer* menggunakan seleksi ataupun iterasi yang terlalu kompleks sehingga berpotensi menyebabkan kesalahan, seperti misalnya *nested selection* yang terlalu dalam, bahkan melebihi batas kemampuan *compiler*

3. PEMBANGUNAN SOFTWARE

3.4 Programmer

Programmer harus mempu menerjemahkan *requirements* yang telah dituangkan dalam desain menjadi kode-kode program yang jika dikompilasi akan menjadi software

3. PEMBANGUNAN SOFTWARE

3.4 Programmer

4. QUALITY ASSURANCE (QA)

Quality Assurance atau yang lebih dikenal dengan sebutan QA merupakan **pengujian atau *testing*** terhadap suatu produk sistem atau aplikasi **untuk memastikan bahwa sistem yang dikembangkan sesuai dengan kebutuhan yang telah ditentukan sebelumnya** sehingga menghasilkan sistem yang terjamin kualitasnya. Pengujian terhadap aplikasi biasa dijalankan oleh QA *Engineer.*

4. QUALITY ASSURANCE (QA)

Dalam melakukan proses *testing*, seorang QA *Engineer* harus memiliki beberapa kriteria kemampuan tertentu. Kemampuan tersebut diantaranya yaitu:

1. *Mindset* Pengujian
2. Analisa dan Pengujian Fungsional
3. Perbaikan Proses
4. Pengujian Keamanan
5. Pengujian Performa
6. *User Acceptance Testing*

5. DOKUMENTASI

Dokumentasi Selama Fase Project

Dokumen Konsep proyek (Project Concept Document) – ikhtisar dariapa yang diucapkan client pada meeting pendahuluan (preliminary meetings)

Dokumen Kebutuhan Proyek (Project Requirement Document) – hasil dari analisa kebutuhan.

Dokumen Persetujuan / Validasi (Project Charter) – dokumen yang berisi pengesahan manajemen dari client (acknowledges),bahwa proyek diizinkan untuk mengalokasikan sumberdaya.

Dokumen lingkup proyek(Project Scope Document) – kandungan proyek (yang berhubungan dengan proyek seperti : project members, project sponsor, dsb).

Dokumen Perencanaan Proyek (Project Plan) – detil yang menunjukkan strategi untuk dapat menyelesaikan proyek. Outline-nya bisa berupa tahapan-tahapan fase dan langkah demi langkah kerja.

Dokumen sosialisasi (Closing Document) – metode sosialisasi, training, serah terima dengan stakeholders dan komitmen akhir seperti garansi dsb.

5. DOKUMENTASI

Semua proyek pastinya memiliki dokumentasi dalam prosesnya. Banyak proyek yang juga menggunakan dokumentasi untuk penyelesaian tugasnya.

Pengumpulan dokumentasi dalam bentuk apapun ke dalam satu tempat atau *database*, dan mengkategorikannya ke dalam bentuk yang mudah di cari dapat membuat pemantauan dan pembuatan rencana ke depan lebih mudah.

Untuk ukuran proyek kecil, dampak dari komponen ini memang tidak terlalu kelihatan, namun pada proyek besar, dampaknya akan sangat kelihatan.

Karena dengan mengumpulkan semua dokumentasi ke dalam satu tempat, *project manager* dan manajemen dapat memantau perkembangan proyek dengan lebih mudah dan efisien.

5. DOKUMENTASI

Semua proyek pastinya memiliki dokumentasi dalam prosesnya. Banyak proyek yang juga menggunakan dokumentasi untuk penyelesaian tugasnya.

Pengumpulan dokumentasi dalam bentuk apapun ke dalam satu tempat atau *database*, dan mengkategorikannya ke dalam bentuk yang mudah di cari dapat membuat pemantauan dan pembuatan rencana ke depan lebih mudah.

REFERENSI

Agan, DJ, Debuging: The 9 Idispensable Rules for Finding Even the Most Eluisive Software and Hardware Problems, 2002

Cooney B, Separiting Analiysis from design, IRM Trainning 2007

Poolet, M.A., Databae Design anda Management "Getting It Right the First Time, Mount Vernon Data System, 229

Anonymous, Guidance for Preparing Standard Operating Procedures (SOPs), 2007

THANK YOU

Dilanjutkan pada pertemuan berikutnya..!!!!!!!!!!!