



Teknik Industri
Fakultas Teknologi Industri
Universitas Islam Sultan Agung

MODUL PRA KTIKUM ANALISA PERANCANGAN SISTEM INFORMASI

**PENYUSUN :
BRAY DEVA BERNADHI, ST., MT.**



Format Laporan

Bab 1 Pendahuluan

- 1.1 Latar Belakang
- 1.2 Rumusan Masalah
- 1.3 Batasan Masalah
- 1.4 Tujuan Praktikum
- 1.5 Sistematika

Bab 2 Tinjauan Pustaka

Bab 3 Pengumpulan dan Pengolahan Data

- 3.1 Pengumpulan Data
 - 1. data sistem lama
 - 2. data sistem baru
- 3.2 Pengolahan Data
 - 1. Flowchart
 - 2. Algoritma
 - 3. Data Flow Diagram
 - 3. interface
 - 4. Verifikasi
 - 5. Coding/script/pseudocode

Bab 4 Analisa

- 4.1 Analisa Flowchart, Algoritma dan Data Flow Diagram
- 4.2 Analisa fungsi script
- 4.3 Kelemahan dan keunggulan sistem yang lama dan baru
- 4.4 Evaluasi atau rekomendasi

Bab 5 Kesimpulan dan Saran

- 5.1 Kesimpulan
- 5.2 Saran

BAB 1

SISTEM INFORMASI

1.1 Pengertian Sistem

“Sistem adalah kumpulan dari komponen-komponen yang saling berhubungan yang saling berinteraksi untuk melakukan suatu tugas untuk mencapai suatu tujuan”(Williams dan Sawyer, 2007 : 552). Sistem (O'Brian dan Marakas, 2009 : 24) adalah kumpulan komponen yang saling berhubungan dengan batasan yang jelas, dan bekerja sama untuk mencapai tujuan dengan menerima *input* dan menghasilkan *output* dalam suatu proses transformasi yang terorganisasi. Dalam sistem terdapat 3 komponen dasar yang terdapat didalamnya (O'Brian dan Marakas, 2009 : 24), seperti:

1. *Input*, memasukkan elemen-elemen (data mentah) yang akan diproses.
2. *Process*, proses transformasi *input* menjadi *output*.
3. *Output*, mengirimkan elemen-elemen (data mentah) yang telah diproses ke tujuannya.

Jadi, sistem adalah sekumpulan komponen yang saling terkait dan bekerja sama melakukan suatu tugas untuk mencapai suatu tujuan.

1.2 Pengertian Informasi

Informasi (Bali, 2009 : 5) adalah data yang telah disusun menjadi pola yang bermakna, sehingga data yang dimiliki dapat menghasilkan nilai yang relevan dan sesuai dengan tujuan yang diinginkan.

Informasi adalah data yang telah dirangkum atau dimanipulasi dalam bentuk lain dengan tujuan untuk mengambil keputusan, misalnya : jumlah suara untuk setiap kandidat yang digunakan dalam menentukan pemenang pemilu (Williams dan Sawyer, 2007 : 40).

Jadi, informasi adalah sekumpulan data yang dirangkum dan dimanipulasi sesuai dengan tujuan yang diinginkan untuk mengambil suatu keputusan bagi pemakai akhir.

1.3 Pengertian Sistem Informasi

Sistem Informasi (Satzinger, Jackson, dan Burd, 2010 : 7) adalah sekumpulan

komponen terpisah yang berfungsi untuk mengumpulkan, mengolah, menyimpan, dan menyediakan tugas-tugas dalam bisnis.

Sistem Informasi merupakan kombinasi teratur apapun dari orang-orang, *hardware*, *software*, jaringan komputer, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.

Jadi, sistem informasi adalah sekumpulan komponen yang saling terkait yang berfungsi untuk mengumpulkan, mengolah, menyimpan, dan menyediakan *output* untuk mencapai tujuan tertentu dalam suatu organisasi.

1.4 Pengertian DataBase

Pengertian Database,- Penggunaan teknologi dalam sebuah perusahaan, institusi ataupun organisasi mempunyai peranan penting guna mencapai tujuan. Suatu perusahaan dituntut untuk bekerja se-efisien mungkin supaya bisa bertahan di atas kerasnya persaingan. Salah satu teknologi yang harus dimiliki oleh sebuah perusahaan, institusi maupun organisasi adalah teknologi dalam memproses data sehingga menjadi informasi yang berguna, teknologi yang dimaksud adalah sistem pengolahan basis data atau database. Penggunaan database yang baik pada perusahaan retail misalnya, mampu membantu seorang kasir bekerja lebih cepat ketika mencari jumlah barang atau harga barang yang akan dijual. Begitupun dengan admin, database mempermudah ketika pencarian stok persediaan, barang paling laku dan banyak lagi yang lainnya.

Pengertian database adalah sekumpulan data yang sudah disusun sedemikian rupa dengan ketentuan atau aturan tertentu yang saling berelasi sehingga memudahkan pengguna dalam mengelolanya juga memudahkan memperoleh informasi. Selain itu adapula yang mendefinisikan database sebagai kumpulan file, tabel, atau arsip yang saling terhubung yang disimpan dalam media elektronik. Beberapa manfaat database yang bisa kita dapatkan antara lain:

1. Kecepatan dan Kemudahan

Database memiliki kemampuan dalam menyeleksi data sehingga menjadi suatu kelompok yang terurut dengan cepat. Hal inilah yang akhirnya dapat menghasilkan informasi

yang dibutuhkan secara cepat pula. Seberapa cepat pemrosesan data oleh database tergantung pula pada perancangan databasenya.

2. Pemakaian Bersama-sama

Suatu database bisa digunakan oleh siapa saja dalam suatu perusahaan. Sebagai contoh database mahasiswa dalam suatu perguruan tinggi dibutuhkan oleh beberapa bagian, seperti bagian admin, bagian keuangan, bagian akademik. Kesemua bidang tersebut membutuhkan database mahasiswa namun tidak perlu masing-masing bagian membuat databasenya sendiri, cukup database mahasiswa satu saja yang disimpan di server pusat. Nanti aplikasi dari masing-masing bagian bisa terhubung ke database mahasiswa tersebut.

3. Kontrol data terpusat

Masih berkaitan dengan point ke dua, meskipun pada suatu perusahaan memiliki banyak bagian atau divisi tapi database yang diperlukan tetap satu saja. Hal ini mempermudah pengontrolan data seperti ketika ingin mengupdate data mahasiswa, maka kita perlu mengupdate semua data di masing-masing bagian atau divisi, tetapi cukup di satu database saja yang ada di server pusat.

4. Menghemat biaya perangkat

Dengan memiliki database secara terpusat maka di masing-masing divisi tidak memerlukan perangkat untuk menyimpan database terhubung database yang dibutuhkan hanya satu yaitu yang disimpan di server pusat, ini tentunya memangkas biaya pembelian perangkat.

5. Keamanan Data

Hampir semua Aplikasi manajemen database sekarang memiliki fasilitas manajemen pengguna. Manajemen pengguna ini mampu membuat hak akses yang berbeda-beda disesuaikan dengan kepentingan maupun posisi pengguna. Selain itu data yang tersimpan di database diperlukan password untuk mengaksesnya.

6. Memudahkan dalam pembuatan Aplikasi baru

Dalam poin ini database yang dirancang dengan sangat baik, sehingga si perusahaan memerlukan aplikasi baru tidak perlu membuat database yang baru juga, atau tidak perlu mengubah kembali struktur database yang sudah ada. Sehingga Si pembuat aplikasi atau programmer hanya cukup membuat atau pengatur antarmuka aplikasinya saja.

Dengan segudang manfaat dan kegunaan yang dimiliki oleh database maka sudah seharusnya semua perusahaan baik itu perusahaan skala kecil apalagi perusahaan besar memiliki database yang dibangun dengan rancangan yang baik. Ditambah dengan pemanfaatan teknologi jaringan komputer maka manfaat database ini akan semakin besar. Penggunaan database sekaligus teknologi jaringan komputer telah banyak digunakan oleh berbagai macam perusahaan, contohnya saja perbankan yang memiliki cabang di setiap kotanya. Perusahaan Bank tersebut hanya memiliki satu database yang disimpan di server pusat, sedangkan cabang-cabangnya terhubung melalui jaringan komputer untuk mengakses database yang terletak di server pusat tersebut.

1.5 Komponen atau Elemen Penyusun Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (building blok), yang terdiri dari komponen input, komponen model, komponen output, komponen teknologi, komponen hardware, komponen software, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Komponen input

Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumendokumen dasar.

2. Komponen model

Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen output

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen teknologi

Teknologi merupakan "tool box" dalam sistem informasi, Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, neghasilkan

dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen hardware

Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi. Yang berfungsi sebagai tempat untuk menampung database atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen software

Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

7. Komponen basis data

Basis data (database) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (Database Management System).

8. Komponen kontrol

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, te,peratur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

ELEMEN SISTEM INFORMASI

Sistem informasi terdiri dari elemen-elemen yang terdiri dari orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data.

Semua elemen ini merupakan komponen fisik.

1. Orang

Orang atau personil yang di maksudkan yaitu operator komputer, analis sistem, programmer, personil data entry, dan manajer sistem informasi/EDP

2. Prosedur

Prosedur merupakan elemen fisik. Hal ini di sebabkan karena prosedur disediakan dalam bentuk fisik seperti buku panduan dan instruksi. Ada 3 jenis prosedur yang dibutuhkan, yaitu instruksi untuk pemakai, instruksi untuk penyiapan masukan, instruksi pengoperasian untuk karyawan pusat komputer.

3. Perangkat keras

Perangkat keras bagi suatu sistem informasi terdiri atas komputer (pusat pengolah, unit masukan/keluaran), peralatan penyiapan data, dan terminal masukan/keluaran.

4. Perangkat lunak

Perangkat lunak dapat dibagi dalam 3 jenis utama :

- a. Sistem perangkat lunak umum, seperti sistem pengoperasian dan sistem manajemen data yang memungkinkan pengoperasian sistem komputer.
- b. Aplikasi perangkat lunak umum, seperti model analisis dan keputusan.
- c. Aplikasi perangkat lunak yang terdiri atas program yang secara spesifik dibuat untuk setiap aplikasi.

5. Basis data

File yang berisi program dan data dibuktikan dengan adanya media penyimpanan secara fisik seperti diskette, hard disk, magnetic tape, dan sebagainya. File juga meliputi keluaran tercetak dan catatan lain diatas kertas, mikro film, an lain sebagainya.

6. Jaringan komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data.

7. Komunikasi data

Komunikasi data adalah merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer-komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu sistem informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain.

1.6 Pengertian Analisa dan Perancangan Sistem

Analisa sistem didefinisikan sebagai bagaimana memahami dan menspesifikasi dengan detail apa yang harus dilakukan oleh sistem. Sedangkan sistem desain diartikan sebagai menjelaskan dengan detail bagaimana bagian-bagian dari sistem informasi diimplementasikan. Sehingga Analisa dan desain sistem informasi (ANSI) bisa didefinisikan sebagai: Proses organisasional kompleks dimana sistem informasi berbasis komputer diimplementasikan. Sehingga bisa diringkas sebagai berikut:

- a. Analysis: mendefinisikan masalah**
– From requirements to specification
- b. Design: memecahkan masalah**
– From specification to implementation

Kenapa tahapan-tahapan ini penting?

1. Kesuksesan suatu sistem informasi tergantung pada analisa dan perancangan yang baik
2. ANSI telah digunakan secara luas di berbagai industri (teknologi yang telah teruji)
3. Bagian dari karir dalam dunia IT, menawarkan banyak kesenangan dan tantangan serta gaji yang tidak rendah
4. Kenaikan permintaan akan keahlian sistem analis

Secara singkat berdasarkan pendekatan ANSI, seorang sistem analis adalah orang yang bertugas:

1. Bagaimana membangun sistem informasi
2. Bagaimana menganalisa kebutuhan dari sistem informasi

3. Bagaimana merancang sebuah Sistem Informasi berbasis komputer
4. Bagaimana memecahkan masalah dalam organisasi melalui sistem informasi

1.7 Metodologi pengembangan Sistem

Proses-proses standard yang digunakan untuk membangun suatu sistem informasi meliputi langkah-langkah berikut ini:

1. Analisa
2. Desain
3. Implementasi
4. Maintenance

Pada perkembangannya, proses-proses standar tadi dituangkan dalam satu metode yang dikenal dengan nama Systems Development Life Cycle (SDLC) yang merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan dari usaha analisa dan desain. SDLC meliputi fase-fase sebagai berikut:

1. Identifikasi dan seleksi proyek
2. Inisiasi dan perencanaan proyek
3. Analisa
4. Desain
 - Desain logikal
 - Desain Fisikal
5. Implementasi
6. Maintenance

1.7.1 Identifikasi dan seleksi proyek

Langkah pertama dalam SDLC keseluruhan informasi yang dibutuhkan oleh sistem diidentifikasi, dianalisa, diprioritaskan dan disusun ulang. Dalam langkah ini dilakukan beberapa hal diantaranya:

- Mengidentifikasi proyek-proyek yang potensial
- Melakukan klasifikasi dan me-rangking proyek
- Memilih proyek untuk dikembangkan.

Adapun sumber daya yang terlibat adalah *user, sistem analis, manager yang mengkoordinasi proyek.*

Aktivitas yang dilakukan meliputi: *mewawancarai manajemen user, merangkum pengetahuan yang didapatkan, dan mengestimasi cakupan proyek dan mendokumentasikan hasilnya*

Output: Laporan kelayakan berisi definisi masalah dan rangkuman tujuan yang ingin dicapai

1.7.2 Inisiasi dan perencanaan proyek

Dalam tahapan ini Proyek SI yang potensial dijelaskan dan argumentasi untuk melanjutkan proyek dikemukakan. Rencana kerja yang matang juga disusun untuk menjalankan tahapan-tahapan lainnya. Hasil dari tahapan ini adalah :

Langkah-langkah detail - rencana kerja - high level system requirement penugasan untuk anggota tim.

1.7.3 Tahapan Analisa

Fase ketiga dalam SDLC dimana sistem yang sedang berjalan dipelajari dan sistem pengganti diusulkan. Dalam tahapan ini dideskripsikan sistem yang sedang berjalan, masalah dan kesempatan didefinisikan, dan rekomendasi umum untuk bagaimana memperbaiki, meningkatkan atau mengganti sistem yang sedang berjalan diusulkan. Tujuan utama dari fase analisis adalah untuk memahami dan mendokumentasikan kebutuhan bisnis (business need) dan persyaratan proses dari sistem baru.

Ada 6 aktifitas utama dalam fase ini:

- Pengumpulan informasi
- Mendefinisikan sistem requirement
- Membangun prototype untuk menemukan requirement
- Memprioritaskan requirement
- Menyusun dan mengevaluasi alternatif
- Mereview requiremen dengan pihak manajemen

1.7.4 Tahapan Desain

Pada tahapan ini deskripsi dari requirement yang telah direkomendasikan diubah ke dalam spesifikasi sistem physical dan logical.

1. Logical Design

Bagian dari fase desain dalam SDLC dimana semua fitur-fitur fungsional dari sistem dipilih dari tahapan analisis dideskripsikan terpisah dari platform komputer yang nanti digunakan. Hasil dari tahapan ini adalah :

- Deskripsi fungsional mengenai data dan proses yang ada dalam sistem baru
- Deskripsi yang detail dari spesifikasi sistem meliputi:
 - Input
 - Output
 - Process

2. Physical design

Pada bagian ini spesifikasi logical diubah ke dalam detail teknologi dimana pemrograman dan pengembangan sistem bisa diselesaikan. Adapun output dari sistem ini adalah :

- Deskripsi teknikal
- Deskripsi yang detail dari spesifikasi sistem meliputi:
 - programs,
 - files,
 - network,
 - system software

Pada tahapan desain ada beberapa aktifitas utama yang dilakukan yaitu:

- Merancang dan mengintegrasikan network
- Merancang Arsitektur aplikasi
- Mendesain user interface
- Mendesain sistem interface
- Mendesain dan mengintegrasikan database
- Memnuat prototype untuk detail dari desain
- Mendesain dan mengintegrasikan kendali sistem

1.7.5 Implementasi

Tahapan kelima pada SDLC, dimana pada tahapan ini dilakukan beberapa hal yaitu:

1. Coding
2. Testing
3. Insalasi

Output dari tahapan ini adalah : source code, prosedur pelatihan.

1.7.6 Maintenance

Langkah terakhir dari SDLC dimana pada tahapan ini sistem secara sistematis diperbaiki dan ditingkatkan. Hasil dari tahapan ini adalah Versi baru dari software yang telah dibuat.

1.7.7 Contoh Sistem Informasi

Berikut ini merupakan beberapa contoh informasi :

1. Sistem reservasi penerbangan, digunakan dalam biro perjalanan untuk melayani pemesanan/pembelian tiket.
2. Sistem POS (Point Of Sale) yang diterapkan di pasar swalayan dengan dukungan barcode reader untuk mempercepat pemasukan data.
3. Sistem layanan akademis yang memungkinkan mahasiswa memperoleh data akademis dan mendaftar mata kuliah yang diambil pada tiap semester.
4. Sistem penjualan secara kredit agar dapat memantau hutang pelanggan yang jatuh tempo.
5. Sistem smart card yang dapat digunakan tenaga medis untuk mengetahui riwayat penyakit pasien.

1.8 Web Programming

Web Programming merupakan bahasa program yang digunakan untuk membuat sebuah situs *web* (Wikipedia, 2013). Dan dibawah ini merupakan contoh bahasa pemrograman yang digunakan dalam membuat sebuah situs *web*, yaitu :

1. HTML (*HyperText Markup Language*)

Sebuah sistem yang digunakan untuk menandai atau menandakan sebuah dokumen, sehingga dapat dipublikasikan dalam *web*. Dimana, HTML mendefinisikan apa yang pada umumnya ditransmisikan di antara *node* dalam jaringan (Connolly dan Begg, 2010 : 1031). Sebuah file HTML ditulis menggunakan *tag-tag* dasar HTML yang terdiri dari `<html></html><head></head>`, dan `<body></body>` untuk membentuk sebuah halaman HTML

2. AJAX (*Asynchronous Java Script and XML*)

AJAX merupakan sekelompok teknik pengembangan *web* yang saling terkait, dimana digunakan pada *client* untuk menciptakan aplikasi *web* yang interaktif (Surhone, Tennoe, dan Henssonow, 2011: 3).

3. JavaScript

Bahasa pemrograman yang sangat sederhana yang memungkinkan halaman HTML untuk memasukkan fungsi dan skrip yang dapat mengenali dan menanggapi peristiwa pengguna, seperti : klik *mouse*, *input user*, dan navigasi halaman (Connolly dan Begg, 2010: 1041).

4. CSS (*Cascading Style Sheet*)

CSS adalah bahasa sederhana yang mendefinisikan konstruksi *style*, seperti : tulisan, warna, dan posisi, yang digunakan untuk menjelaskan bagaimana pada informasi pada halaman *web* yang diformat dan ditampilkan (Meloni, 2012 : 46).

CSS memiliki sifat dalam berbagai *style* yang digunakan untuk mengontrol tulisan, warna, keselarasan, dan *margin* (Meloni, 2012 : 52). Properti *style* dalam CSS secara umum dapat dikelompokkan dalam dua kategori utama, yaitu :

- a. *Layout properties*, yaitu terdiri dari beberapa sifat yang mempengaruhi posisi elemen pada halaman *web*, seperti : *margin*, *padding*, keselarasan dan sebagainya.

- b. *Formatting properties*, yaitu terdiri dari beberapa sifat yang mempengaruhi tampilan secara visual dari elemen dalam sebuah *web*, seperti: jenis tulisan, ukuran, warna, dan sebagainya.

1.8.1 Internet

“*Internet* adalah jaringan komputer di seluruh dunia yang menghubungkan ratusan bahkan ribuan jaringan yang lebih kecil” (Williams dan Sawyer, 2007 : 17). *Internet* adalah sebuah jaringan *web* tertutup yang dapat diakses oleh semua orang dan anggota organisasi untuk saling bertukar informasi (Connolly dan Begg, 2010 : 1026). Jadi, *Internet* adalah jaringan komputer yang menghubungkan berbagai jaringan nirkabel yang dapat diakses oleh semua orang untuk saling bertukar informasi dan sebagai teknologi komunikasi.

1.8.2 URL (*Uniform Resource Locator*)

URL (*Uniform Resource Locator*) adalah kumpulan karakter yang menunjuk pada sebuah potongan informasi khusus pada bagian *web* (Williams dan Sawyer, 2007 : 66). Sebuah URL terdiri dari protokol *web*, nama domain atau *web server*, direktori (*folder*) pada *server*, dan *file* yang terdapat pada direktori. Contoh URL untuk situs *web*, yaitu : <http://www.nps.gov/yose/home.htm>. Maka dapat dijelaskan karakter-karakter yang dimiliki oleh URL, yaitu :

1. Protokol : *http://* adalah sekumpulan aturan komunikasi untuk bertukar informasi. HTTP (*HyperText Transfer Protocol*) merupakan aturan komunikasi yang memungkinkan *browser* untuk tersambung ke *server web*.
2. Nama domain (nama *web server*) : www.nps.gov/ merupakan suatu domain yang menjelaskan suatu lokasi di *Internet*, yaitu di *web server* tertentu.
3. Nama direktori : *yose/* adalah nama direktori atau folder di *server* tempat *browser* mengambil *file*.
4. Nama *file* dan ekstensi : *home.htm*, *file* merupakan halaman atau dokumen tertentu yang *user* cari.

1.9 *Object Oriented Analysis and Design (OOAD)*

Object Oriented Analysis (OOA) mendefinisikan bahwa semua jenis objek yang

melakukan pekerjaan dalam suatu sistem dan menunjukkan interaksi dari pengguna yang dibutuhkan untuk menyelesaikan tugas (Satzinger, Jackson, dan Burd, 2010: 60). *Object Oriented Design* (OOD) didefinisikan bahwa semua tipe objek dari objek yang diperlukan untuk dapat melakukan komunikasi dengan orang dan perangkat yang ada di dalam sistem, untuk menggambarkan bagaimana objek dapat berinteraksi dalam menyelesaikan tugas serta menyesuaikan dan menyempurnakan masing-masing jenis objek sehingga dapat diimplementasikan dengan bahasa tertentu (Satzinger, Jackson, dan Burd, 2010: 60).

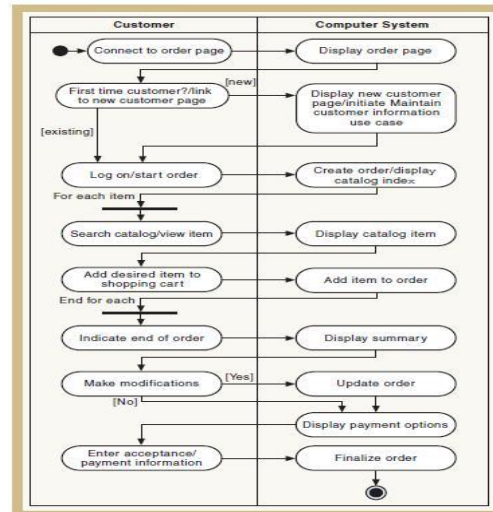
1.10 Unified Model Language (UML)

Unified Model Language (UML) merupakan serangkaian standar konstruksi model dan notasi yang secara khusus dirancang untuk pengembangan *object-oriented* (Satzinger, Jackson, dan Burd, 2010: 240). Model dari komponen sistem yang menggunakan UML, meliputi :

1. *Activity Diagram*
2. *Event Table*
3. *Use Case Diagram*
4. *Use Case Description*
5. *Sequence Diagram*
6. *Updated Class Diagram*
7. *Communication Diagram*
8. *User Interface*

1.10.1 Activity Diagram

Activity diagram adalah sebuah bentuk *workflow* diagram yang menjelaskan berbagai aktivitas *user* (sistem) dan *user* yang melakukan kegiatan (Satzinger, Jackson, dan Burd, 2010: 141). *Activity diagram* digunakan untuk menggambarkan proses pengembangan serta menggambarkan urutan dari proses bisnis dari sebuah organisasi (Bennet, McRobb, dan Farmer, 2010 : 113).



Gambar 1.1 Contoh Activity Diagram

Sumber : (Satzinger, Jackson, dan Burd, 2010 : 252)

1.10.2 Event Table

Event Table adalah sebuah katalog dari *use case* yang memuat kejadian dalam baris dan bagian penting dari informasi yang mengenai masing-masing kejadian dalam kolom (Satzinger, Jackson, dan Burd, 2010: 168). *Event table* terdiri dari *event*, *trigger*, *source*, *use case*, *response*, dan *destination*. Pada Gambar 3 yang terdapat pada halaman lampiran menjelaskan kegiatan yang terjadi pada *event table*, seperti : *Event* adalah sebuah kejadian pada suatu waktu dan tempat yang spesifik dan dapat digambarkan serta mudah untuk diingat. *Trigger* adalah sebuah tanda yang memberitahu sistem mengenai suatu kejadian yang terjadi pada saat data dibutuhkan. *Source* adalah *user* eksternal yang menyuplai data ke sistem. *Response* adalah sebuah *output* yang dihasilkan oleh sistem untuk mencapai sebuah tujuan. *Destination* adalah *user* eksternal yang menerima data dari sistem (Satzinger, Jackson, dan Burd, 2010: 169). *Event table* digambarkan dalam bentuk tabel yang berisi kolom dan baris, yang menjelaskan kejadian beserta dengan masing-masing detilnya.

Customer support system event table					
Event	Trigger	Source	Use case	Response	Destination
1. Customer wants to check item availability.	Item inquiry	Customer	Look up item availability	Item availability details	Customer
2. Customer places an order	New order	Customer	Create new order	Real-time link	Credit bureau
				Order confirmation	Customer
				Order details	Shipping
3. Customer changes or cancels order	Order change request	Customer	Update order	Transaction	Bank
				Change confirmation	Customer
				Order change details	Shipping
				Transaction	Bank

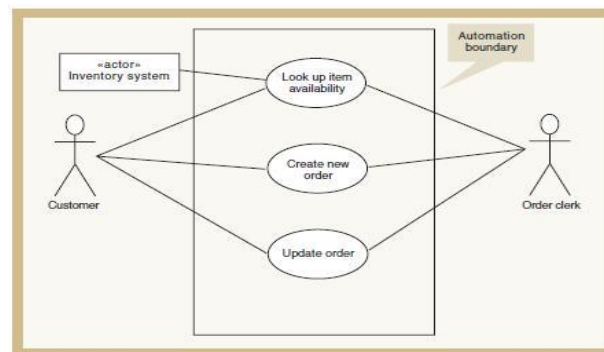
Gambar 1.2 Contoh *Event Table*

Sumber : (Satzinger, Jackson, dan Burd, 2010 : 170)

1.10.3 Use Case Diagram

Use Case Diagram adalah sebuah *diagram* yang menunjukkan berbagai peran *user* dan bagaimana cara *user* berinteraksi dengan sistem. Pada gambar 4 yang terdapat pada halaman lampiran menunjukkan *use case diagram* terdapat komponen-komponen sederhana yang menjelaskan bahwa yang melakukan interaksi dengan sistem dapat dikatakan sebagai *actor*, sedangkan untuk mengidentifikasi berbagai proses yang dilakukan oleh *user* dengan sistem baru merupakan *use case*, dan *actor* akan dihubungkan dengan *use case* menggunakan *connecting actor-use case*

(Satzinger, Jackson, dan Burd, 2010: 242).

**Gambar 1.3** Contoh *Use Case Diagram*

Sumber : (Satzinger, Jackson, dan Burd, 2010 : 244)

1.10.4 Fully Developed Description

Fully Developed Description merupakan metode formal yang digunakan untuk mendokumentasikan suatu *use case*. Metode yang dimaksud bertujuan untuk mendefinisikan alur dari kegiatan *use case*, dan memungkinkan *user* untuk memahami alur proses bisnis dan bagaimana sistem harus mendukung (Satzinger, Jackson, dan Burd, 2010:173).

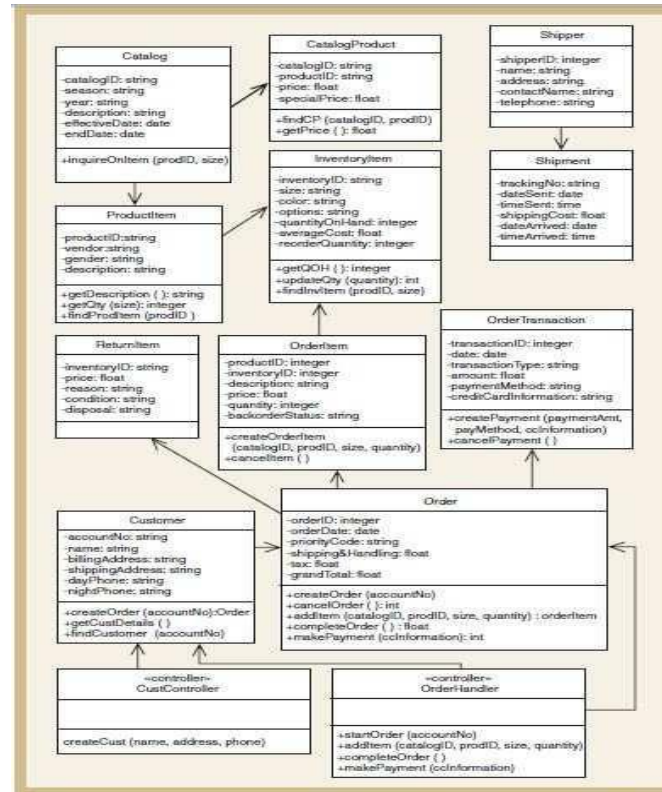
Use Case Name:	Create new order	
Scenario:	Create new telephone order	
Triggering Event:	Customer telephones RMO to purchase items from the catalog.	
Brief Description:	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.	
Actors:	Telephone sales clerk.	
Related Use Cases:	Includes: <i>Check item availability</i> .	
Stakeholders:	Sales department: to provide primary definition. Shipping department: to verify information content is adequate for fulfillment. Marketing department: to collect customer statistics for studies of buying patterns.	
Preconditions:	Customer must exist. Catalog, Products, and Inventory Items must exist for requested items.	
Postconditions:	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
Flow of Activities:	Actor	System
	<ol style="list-style-type: none"> 1. Sales clerk answers telephone and connects to a customer. 2. Clerk verifies customer information. 3. Clerk initiates the creation of a new order. 4. Customer requests an item be added to the order. 5. Clerk verifies the item (<i>Check item availability</i> use case). 6. Clerk adds item to the order. 7. Repeat steps 4, 5, and 6 until all items are added to the order. 8. Customer indicates end of order; clerk enters end of order. 9. Customer submits payment; clerk enters amount. 	<ol style="list-style-type: none"> 2.1 Display customer information. 3.1 Create a new order. 5.1 Display item information. 6.1 Add an order item. 8.1 Complete order. 8.2 Compute totals. 9.1 Verify payment. 9.2 Create order transaction. 9.3 Finalize order.
Exception Conditions:	<ol style="list-style-type: none"> 2.1 If customer does not exist, then the clerk pauses this use case and invokes <i>Maintain customer information</i> use case. 2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative. 4.1 If an item is not in stock, then customer can <ol style="list-style-type: none"> a. choose not to purchase item, or b. request item be added as a back-ordered item. 9.1 If customer payment is rejected due to bad-credit verification, then <ol style="list-style-type: none"> a. order is canceled, or b. order is put on hold until check is received. 	

Gambar 1.4 Contoh *Fully Developed Description*

Sumber : (Satzinger, Jackson, dan Burd, 2010 : 250)

1.10.5 Updated Design Class Diagram

Updated design class diagram merupakan pengembangan dari *first-cut class diagram*. Pada *updated design class diagram* ini, informasi metode dapat ditambahkan ke dalam *class* dan dilakukan pembaharuan arah panah navigasi yang didapat dari *sequence diagram* yang telah dibuat sebelumnya (Satzinger, Jackson, dan Burd, 2010: 457).



Gambar 1.5 Contoh Updated Class Diagram

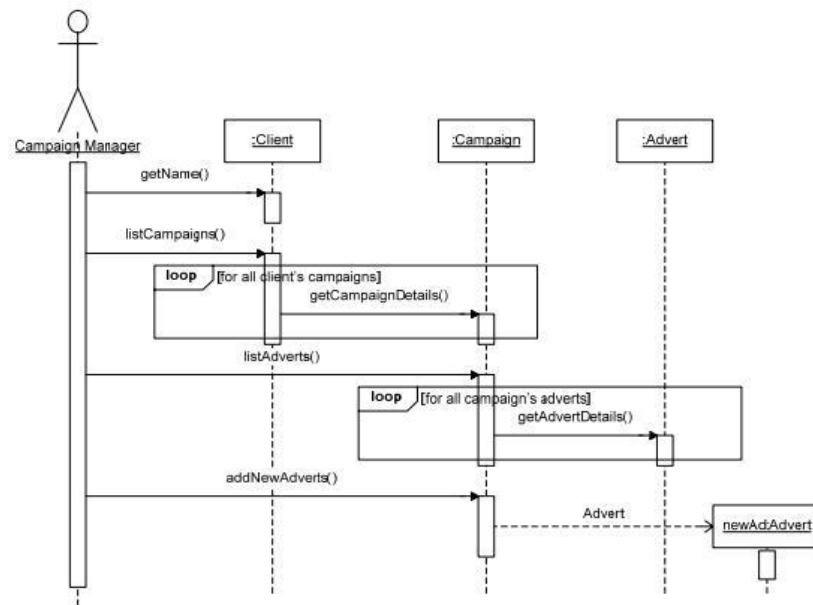
Sumber : (Satzinger, Jackson, dan Burd, 2010 : 458)

1.10.6 Sequence Diagram

Sequence diagram atau *interaction sequence diagram*, menampilkan sebuah interaksi antara objek-objek yang disusun dalam sebuah urutan waktu. *Sequence diagram* diambil pada tingkat detil yang berbeda dan memenuhi tujuan yang berbeda pada beberapa tahapan dalam siklus pengembangan (Bennet, McRobb, dan Farmer, 2006: 253).

Dimensi vertikal mempresentasikan waktu dan objek yang terlibat dalam interaksi secara horizontal yang melintasi *diagram*. Dimana, setiap objek dalam *sequence diagram* yang dipresentasikan dengan *lifeline*. *Lifeline* ini merupakan garis putus-putus vertikal dengan sebuah simbol objek yang terdapat di bagian atas. *Message* diperlihatkan dengan sebuah tanda panah horizontal padat dari satu *lifeline* ke *lifeline* lainnya dan diberikan label dengan nama *message*.

Dengan periode waktu selama berjalannya operasi dinamakan dengan blok persegi panjang yang diletakkan di sepanjang *lifeline* (Bennet, McRobb, dan Farmer, 2006: 253).



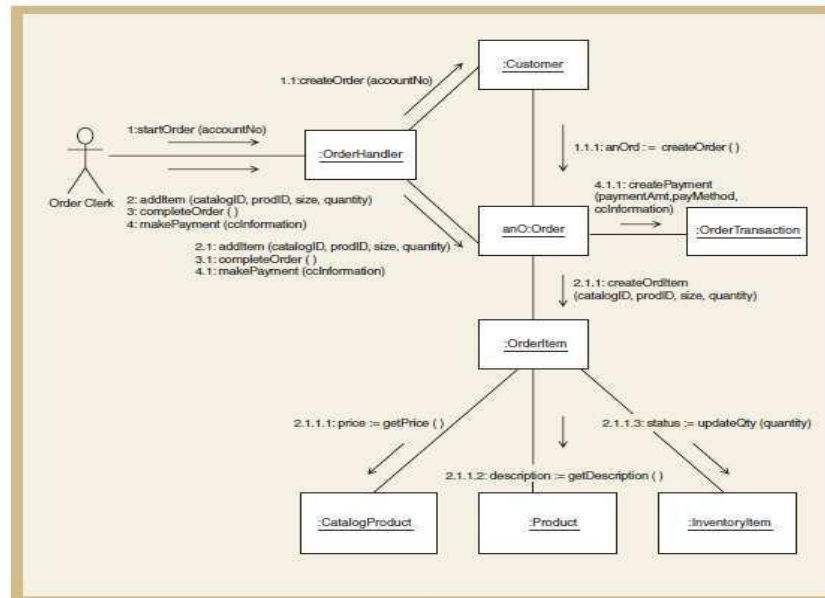
Gambar 1.6 Contoh *Sequence Diagram*

Sumber : (Bennet, McRobb, dan Farmer, 2006 : 254)

1.10.7 *Communication Diagram*

Communication diagram berguna untuk menunjukkan pandangan yang berbeda dari *use case*. *Communication diagram* juga lebih mudah digunakan untuk memberikan gambaran ide-ide dalam rapat, karena mudah untuk diubah (Satzinger, Jackson, dan Burd, 2010: 454).

Simbol yang digunakan pada *communication diagram* sama dengan simbol yang ada pada *sequence diagram*, seperti : aktor, objek, dan pesan. Tetapi, simbol *lifeline* dan *activation lifeline* yang terdapat pada *sequence diagram* tidak digunakan pada *communication diagram*.



Gambar 1.7 Contoh *Communication Diagram*

Sumber : (Satzinger, Jackson, dan Burd, 2010 : 456)

1.10.8 User Interface

User interface (Satzinger, Jackson, dan Burd, 2010:531) adalah sebuah sistem yang saling berhubungan secara langsung dengan *user*. Pada saat *User* menggunakan sistem tersebut, *user* memiliki tiga aspek *user interface*, yaitu:

1. Aspek Fisik *User Interface*

Sebuah *user interface* yang termasuk perangkat-perangkat yang dapat disentuh secara langsung oleh *user*, termasuk *keyboard*, *mouse*, *touch screen*, atau *keypad*. Dan beberapa bagian fisik lain dari sebuah *user interface* yaitu referensi dokumen yang tercetak, *form entry* data, dan lainnya yang dapat digunakan oleh *user* untuk menyelesaikan tugasnya.

2. Aspek Perseptual *User Interface*

Sebuah *user interface* yang dapat dilihat, didengar, maupun disentuh (tidak secara fisik) oleh *end user*. Yang dapat dilihat termasuk data dan instruksi yang ditampilkan pada layar, termasuk bentuk-bentuk, garis, nomor, dan kata-kata.

3. Aspek Konseptual *User Interface*

User interface yang diketahui oleh *end user* mengenai penggunaan sistem, termasuk dalam hal-hal yang terjadi di dalam *problem domain* pada sistem yang dimanipulasi

oleh *user*, operasi-operasi yang dapat dijalankan, dan prosedur-prosedur yang ditaati dalam menjalankan operasi.

2.10.8.1 Interaksi Manusia dan Komputer

Interaksi manusia dan komputer adalah suatu ilmu yang berhubungan dengan perancangan, evaluasi, dan implementasi sistem komputer interaktif yang bertujuan untuk digunakan oleh manusia. Tujuan dari interaksi manusia dan komputer adalah untuk membangun atau merancang suatu sistem yang aman, efektif dan efisien, serta dapat digunakan sesuai dengan kebutuhan (Shneiderman dan Plaisant, 2010 : 22). Interaksi manusia dan komputer dapat meningkatkan kualitas *user interface* dan menu dengan cara beradaptasi terhadap pola pikir manusia serta mempelajari pengaruh jangka panjang komputer terhadap manusia (Williams dan Sawyer, 2007 : 122).

2.10.8.2 Delapan aturan Emas Perancangan Antarmuka

Dalam perancangan suatu situs *web* yang baik harus sesuai dengan aturan yang disebut dengan *Eight Golden Rules of Interface Design* atau delapan aturan emas perancangan antarmuka (Shneiderman dan Plaisant, 2010 : 88), yaitu :

1. Berusaha keras untuk konsisten
Beberapa bentuk konsistensi yang harus diperhatikan adalah konsistensi dalam penggunaan bentuk dan ukuran *font*, pemberian warna dalam tulisan, *background*, pembuatan *layout*, dan penggunaan *terminology*.
2. Memungkinkan *user* menggunakan *shortcut*
Pada umumnya *user* yang sudah sering menggunakan aplikasi, akan menginginkan kecepatan dalam mengakses informasi yang diinginkan. Maka, tingkat interaksi yang dibutuhkan lebih pendek atau singkat dan dapat langsung menunjuk pada fungsi, sehingga diperlukan fasilitas *shortcut*.
3. Memberikan umpan balik yang informatif
Umpan balik harus diberikan untuk memberikan informasi kepada *user* yang sesuai dengan aksi yang dilakukan. *User* akan mengetahui aksi apa yang telah dan akan dilakukan dengan adanya umpan balik. Umpan balik pada umumnya berupa konfirmasi atau informasi dari suatu aksi.

4. Merancang dialog yang menghasilkan nilai akhir
Urutan tindakan harus diatur sesuai dengan bagian awal, tengah, dan akhir suatu grup. Dimana, umpan balik penyelesaian suatu aksi yang informatif menunjukkan aktivitas *user* yang telah selesai dan sukses.
5. Memberikan pencegahan dan penanganan kesalahan yang sederhana
Sistem dirancang dengan mudah dan sederhana sehingga dapat mencegah *user* dalam melakukan kesalahan yang serius. Apabila terjadi suatu kesalahan, maka sistem harus mampu mendeteksi kesalahan tersebut dengan memberikan mekanisme penanganan yang sederhana dan mudah dimengerti.
6. Mengizinkan pembalikan aksi yang mudah
Terkadang, *user* mungkin akan melakukan sebuah aksi yang tidak diinginkan tanpa disengaja. Oleh sebab itu, sebisa mungkin aksi harus dapat diulang sehingga dapat mengurangi kecemasan *user* dalam melakukan kesalahan. Dan *user* dapat mengizinkan pilihan-pilihan lain yang tidak dikenal.
7. Pengontrolan terletak pada pengguna sendiri (*internal focus of control*)
User yang berpengalaman sangat menginginkan pengontrolan yang kuat pada sistem yang digunakan, sehingga *user* dapat menguasai sistem tersebut.
8. Mengurangi beban ingatan jangka panjang
Ingatan yang dimiliki manusia terbatas untuk jangka panjang. Oleh karena itu, tampilan sistem harus sederhana. Sehingga *user* tidak perlu melakukan proses penyimpanan memori.

2.10.8.3 Empat pilar perancangan *Interface*

Untuk merancang sebuah *user interface* harus memperhatikan empat pilar utama (Shneiderman dan Plaisant, 2010 : 121), yaitu :

1. *User Interface Requirements*
Menentukan *user requirements* secara jelas merupakan kunci utama untuk sukses dalam segala aktivitas pengembangan. Segala kebutuhan mengenai *user interface* seperti *input* atau *output device*, *functionality*, *interfaces*, dan *rage of users* harus disampaikan secara spesifik dan disetujui bersama.

2. *Guidline Document and Proseses*

- a. Pada tahap proses perancangan, arsitektur *user-interface* perlu dibangun atau dibuat sebuah petunjuk kerja.
- b. Setiap proyek memerlukan kebutuhan yang berbeda, namun petunjuk kerja yang dibuat harus memperhatikan :
 - *Words, icon dan graphics (terminology, character set, icons, backgrounds).*
 - *Screen-layout issues (menu selection, form fill in, margins, data entry).*
 - *Alat input dan output (response time, keyboard, display).*
 - *Aksi sekuensial (direct-manipulation clicking, commond syntax, programmed function keys, recovery procedures).*
 - *Pelatihan (online help and tutorial, training and reference material).*

3. *User-Interface Software Tools*

Salah satu kesulitan dalam merancang sistem interaktif adalah *customer* dan *user* yang tidak memiliki ide yang jelas dalam bentuk sebuah sistem yang ingin dibuat. Masalah seperti ini dapat diatasi dengan membuat suatu *prototype* dari tampilan sistem yang akan dibuat.

4. *Expert Reviews and Usability Testing*

Setelah suatu sistem selesai dibuat, maka yang harus dilakukan berikutnya adalah mengandalkan *pilot testing* dari seluruh komponen sistem sebelum diberikan kepada *customer*.

1.11 Kunci Sukses Sistem

Kunci sukses sistem merupakan faktor penentu keberhasilan sebuah sistem yang telah dibuat dan memenuhi tujuannya.

1.11.1 Fakta Tentang Sistem Informasi

Pada tahun 1996 *Standingsh Group* melakukan survei hasilnya adalah 46% dari semua proyek Sistem Informasi (SI) perusahaan gagal atau ditinggalkan sebelum selesai. Sisanya selesai, tapi banyak yang terlambat. Ditahun yang sama *General Accounting Office* melakukan survei dengan hasil 53% dari semua SI pemerintah Amerika telah ditinggalkan.

1.11.2 Penyebab Masalah Sistem Informasi

Berikut ini merupakan penyebab masalah sistem informasi yang ada :

1. Analisis berusaha membuat sistem yang mengagumkan tanpa memahami organisasi
2. Analisis tidak menangkap kebutuhan pengguna dengan baik
3. Pengembangan cenderung fokus pada produk, bukan pengguna

1.11.3 Kunci Sukses

Tujuan utama pengembangan adalah bukan membuat sistem yang mengagumkan, namun sistem yang memberi nilai tambah bagi pelanggan atau organisasi

SUMBER :

<http://www.mandalamaya.com/pengertian-database/>

<http://arisuhari.blogspot.co.id/2011/10/komponen-komponen-sistem-informasi.html>

<http://www.definisi-pengertian.com/2015/03/definisi-dan-pengertian-informasi.html>

<http://annarizky83.blogspot.co.id/2015/01/basis-data-dalam-sistem-informasi.html>

<http://scdc.binus.ac.id/himsisfo/2016/07/pengertian-sistem-informasi/>

BAB II

BAHASA DAN ALGORITMA PEMROGRAMAN

2.1 Bahasa Pemrograman

Bahasa atau dalam bahasa Inggris *language* adalah suatu sistem untuk berkomunikasi. Bahasa tertulis menggunakan simbol (yaitu huruf) untuk membentuk kata. Dalam ilmu komputer, bahasa manusia disebut bahasa alamiah (natural language) dimana komputer tidak bisa memahaminya, sehingga diperlukan suatu bahasa komputer. Komputer mengerjakan transformasi data berdasarkan kumpulan perintah program - yang telah dibuat oleh pemrogram. Kumpulan perintah ini harus dimengerti oleh komputer, berstruktur tertentu (syntax) dan bermakna.

Bahasa pemrograman merupakan notasi untuk memberikan secara tepat program komputer. Berbeda dengan bahasa alamiah, mis. Bahasa Indonesia, Inggris dsb. yang merupakan bahasa alamiah (natural language), sintaks dan semantik bahasa pemrograman (komputer) ditentukan secara kaku, sehingga bahasa pemrograman juga disebut sebagai bahasa formal (formal language). Jadi, dalam bahasa pemrograman yang digunakan sebagai alat komunikasi untuk memberikan perintah kepada komputer tidak berlaku kebebasan berekspresi seperti lainnya dalam bahasa alamiah.

Pemrograman dalam pengertian luas meliputi seluruh kegiatan yang tercakup dalam pembuatan program, termasuk analisis kebutuhan (requirement's analysis) dan keseluruhan tahapan dalam perencanaan (planning) perancangan (design) dan penerapannya (implementation). Dalam pengertian yang lebih sempit, pemrograman merupakan pengkodean (coding atau program writing = penulisan program) dan pengujiannya (testing) berdasarkan rancangan tertentu.

Pemahaman yang lebih sempit ini sering digunakan dalam pembuatan program-program terapan komersial yang membedakan antara system analyst yang bertanggung jawab dalam menganalisa kebutuhan, perencanaan dan perancangan program dengan pemrogram (programmer) yang bertugas membuat kode program dan menguji kebenaran program.

Generasi bahasa pemrograman:

- Generasi I: machine language
- Generasi II: assembly language : Assembler
- Generasi III: high-level programming language: C, PASCAL, dan sebagainya.
- Generasi IV: 4 GL (fourth generation language): SQL

2.1.1 Bahasa Tingkat Rendah

Merupakan bahasa assembly atau bahasa mesin. Bahasa ini lebih dekat ke mesin (hardware), dimana pada high-level programming languages, suatu bahasa yang sudah dekat pada bahasa manusia. Pada bahasa ini akan lebih cepat dipahami oleh programmer karena sifatnya yang dekat dengan bahasa manusia.

2.1.2 Bahasa Mesin

Bahasa mesin merupakan representasi tertulis machine code (kode mesin), yaitu kode operasi suatu mesin tertentu. Bahasa ini bersifat khusus untuk mesin tertentu dan "dimengerti" langsung oleh mesin, sehingga pelaksanaan proses sangat cepat. Bahasa mesin kelompok komputer tertentu berlainan dengan bahasa mesin kelompok komputer yang lain. Abstraksi bahasa ini adalah kumpulan kombinasi kode biner "0" dan "1" yang sangat tidak alamiah bagi kebanyakan orang – kecuali insinyur pembuat mesin komputer. Karena tidak alamiah bagi kebanyakan orang, bahasa mesin juga disebut bahasa tingkat rendah.

2.1.3 Bahasa Assembly

Bahasa rakitan (assembly language) merupakan notasi untuk menyajikan bahasa mesin yang lebih mudah dibaca dan dipahami oleh manusia. Bahasa ini sudah menggunakan simbol alpabet yang bermakna (mnemonic). Contoh "MOV AX 1111", pindahkan ke register AX nilai 1111. Proses data oleh komputer berdasarkan perintah bahasa rakitan adalah cepat. Meski demikian masih merepotkan-bahkan bagi kebanyakan pemrogram, karena masih harus mengingat-ingat tempat penyimpanan data. Bahasa rakitan juga bersifat khusus untuk mesin tertentu. Contoh: Assembler

2.1.4 Bahasa Tingkat Tinggi

Adalah bahasa pemrograman yang dekat dengan bahasa manusia, kelebihan utama dari bahasa ini adalah mudah untuk di baca, tulis, maupun diperbaharui, sebelum bisa dijalankan program harus terlebih dahulu di-compile. Contoh Ada, Algol, BASIC, COBOL, C, C++,

FORTRAN, LISP, dan Pascal, dsb. Pada generasi bahasa pemrograman terakhir sekarang ini, kedua cara interpretasi dan kompilasi digabungkan dalam satu lingkungan pengembangan terpadu (IDE=integrated development environment).

2.1.5 Bahasa Generasi 3 & 4

Lebih dekat ke bahasa manusia high – level programming languages. Biasanya dipakai untuk mengakses database. Contoh perintah pada bahasa SQL: FIND ALL RECORDS WHERE NAME IS "JOHN".

2.2 Pengenalan PHP

PHP adalah Bahasa Pemrograman Berbasis Web Server yang ditemukan oleh Rasmus Lerdorf pada tahun 1994. Mulanya, Rasmus Lerdorf membuat sejumlah script Perl yang dapat mengamati siapa saja yang dapat melihat-lihat daftar riwayat hidupnya, pada awal tahun 1995 script-script ini selanjutnya dikemas menjadi tool dalam PHP 1 saat itu PHP merupakan singkatan dari Personal Home Page.

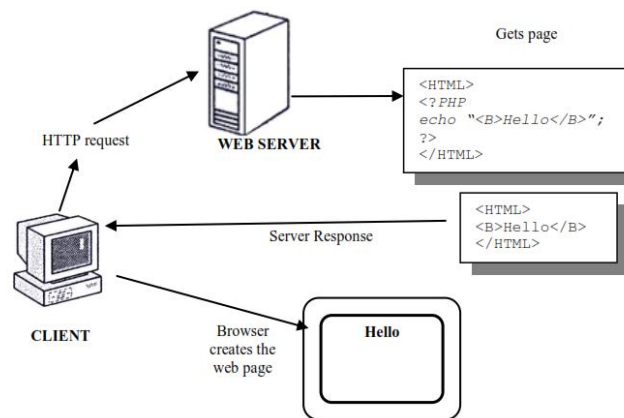
Pada Oktober 1995, Rasmus menciptakan PHP/FI Versi 2. Pada versi inilah pemrograman dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan database dan melakukan perhitungan-perhitungan yang kompleks. Pada saat ini PHP mulai populer sebagai piranti pemrograman web. Pada tahun 1998 lahirlah PHP3 yang kemudian kepanjangan PHP diubah menjadi Professional Home Page dan sekarang berubah menjadi PHP Hypertext Preprocessor. Pada dasarnya PHP dirancang untuk membuat aplikasi web dinamis. Yakni, PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misal, menampilkan isi database ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan script-script seperti ASP (Active Server Page), Cold Fusion, ataupun Perl. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara command line. Artinya, script PHP dapat dijalankan tanpa melibatkan web server maupun browser.

PHP merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah server-side embedded script language artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh

PHP pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di server.

Pada prinsipnya server akan bekerja apabila ada permintaan dari client. Dalam hal ini client menggunakan kode-kode PHP untuk mengirimkan permintaan ke server (dapat dilihat pada gambar dibawah). Ketika menggunakan PHP sebagai server-side embedded script language maka server akan melakukan hal-hal sebagai berikut :

- Membaca permintaan dari client/browser
- Mencari halaman/page
- Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page
- Mengirim kembali halaman tersebut kepada client melalui internet



2.2.1 TAG Dalam PHP

Banyak cara untuk menyisipkan PHP dalam script HTML, ada berbagai macam bentuk tag yang dapat digunakan, antara lain :

- Cara I

<?php → menandai awal tag

...

?> → menandai akhir tag

- Cara II

<? → menandai awal tag

...

?> → menandai akhir tag

- Cara III

<% → menandai awal tag
 ...
 %> → menandai akhir tag

Cara ini sama dengan tag pada ASP. Opsi ini bisa dilakukan jika nilai **asp_tags** pada **php.ini** bernilai **on**.

- Cara IV

<script language="php"> → menandai awal script
 ...
 </script> → menandai akhir script

Namun untuk lebih mudahnya, kita akan menggunakan cara II mulai sekarang dan seterusnya. Berikut ini merupakan contoh penulisan kode PHP di dalam HTML :

```
index.php
<!DOCTYPE html>
<html>
  <body>
    <h1>Program PHP Pertama</h1>
    <?php
      echo "Hello World!";
    ?>
  </body>
</html>
```

2.2.2 Komentar PHP

Komentar adalah bagian dari program yang berfungsi sebagai penjelas atau pemberi keterangan dalam program. Komentar ini tidak akan dieksekusi atau dikerjakan oleh interpreter. Pada PHP, ada 3 macam cara penulisannya :

1. /* komentar */

Tulisan apapun yang berada di antara `/*` dan `*/` akan dianggap sebagai komentar. Cara seperti ini sangat berguna dan efisien untuk pemberian komentar yang memakan banyak baris.

2. // komentar

Tulisan di baris yang sama setelah `//` akan dianggap sebagai komentar. Cara ini berguna untuk pemberian komentar singkat yang tak lebih dari 1 baris saja.

3. # komentar

Sama seperti '//', tulisan di baris yang sama setelah '#' akan dianggap sebagai komentar.

Cara ini berguna untuk pemberian komentar singkat yang tak lebih dari 1 baris saja.

Baris komentar tidak ditampilkan di halaman web browser karena komentar akan diabaikan oleh interpreter.

2.2.3 Variabel PHP

Variabel digunakan untuk menyimpan suatu nilai, seperti text, angka atau array. Ketika sebuah variabel dibuat, variabel tersebut dapat dipakai berulang-ulang. Dalam PHP, penulisan variabel diawali dengan karakter '\$', kemudian diikuti dengan huruf sebagai karakter pertama. Setelah itu, dapat dilanjutkan dengan kombinasi huruf dan angka. Variabel tidak boleh mengandung spasi maupun tanda baca di dalamnya, kecuali underscore ('_'). Variabel pada PHP bersifat "case sensitif", yang berarti Anda harus memperhatikan penulisan huruf besar dan huruf kecil. Variabel \$jumlah berbeda dengan \$Jumlah. Contoh penamaan variabel yang benar :

```
$nama_pemakai
$kota_3
$user1
```

Contoh penamaan variabel yang salah :

```
$nama pemakai
$3kota
$us\er1
```

2.2.4 Konstanta PHP

Konstanta adalah suatu tetapan nilai dalam program. Konstanta tidak dapat dirubah nilai sewaktu program dijalankan, kalau hal itu dilakukan akan menyebabkan error. Untuk mendefinisikan konstanta digunakan :

```
Define (NamaPengenal, nilai_konstanta)
```

Contoh :

```
<?php
//konstanta Judul="Hitung Luas Lingkaran"
Define ("Judul", "Hitung Luas Lingkaran")
//konstanta PHI=3.14
Define ("PHI", 3.14);
Echo Judul;
```

```

    $r=10
    Echo"<BR>Jari-jari : $r<BR>\n";
    $luas=PHI*$r*$r
    Echo "Luas Lingkaran = $luas";
?>

```

2.2.5 Echo dan Print

Echo dan print digunakan untuk menampilkan text atau variable. Contoh penggunaan echo dan print sebagai berikut:

```

echo "<h2>echo dengan tag html</h2>";
echo ("echo menggunakan tanda kurung (");

print "<h2>print dengan tag html</h2>";
print ("print menggunakan tanda kurung (");

```

Untuk melakukan print pada data array gunakan print_r() seperti berikut :

```

$data = array('nama'=>'Johan Budi', 'nim'=> '23456799');
print_r($data);

```

output:

```

{
  nama: "Johan Budi",
  nim: "23456799"
}

```

selain print_r() kita juga dapat menggunakan var_dump() untuk mengetahui tipe data pada variable-nya.

2.2.6 Tipe Data PHP

PHP mendukung tipe data berikut :

1. String
2. Integer
3. Float (floating point numbers – disebut juga double)
4. Boolean
5. Array
6. Object (contoh : Class instance)
7. NULL
8. Resource (contoh: Resource database)

Berikut contoh-contoh penggunaan tipe data:

1. *String*

Termasuk jenis data teks atau untaian karakter. String ditandai dengan tanda petik “ ” (double) atau ‘ ’ (single)

```
$x = "Hello world!";
echo $x;
```

2. *Integer*

Integer adalah tipe data angka bukan pecahan (decimal) yang bernilai antara - 2,147,483,648 dan 2,147,483,647. Berikut contohnya dalam PHP:

```
$x = 5985;
var_dump($x);
```

3. *Float* (dalam PHP disebut juga *double*)

Float adalah tipe data pecahan (decimal). Berikut contohnya :

```
$x = 10.365;
var_dump($x);
```

4. *Bolean*

Bolean adalah tipe data yang bernilai true atau false. Berikut contohnya :

```
$x = true;
$y = false;
```

5. *Array*

Array adalah tipe data yang dapat menyimpan berbagai tipe data dalam satu variable.

Contoh:

```
//array berisi string
$cars = array("Volvo", "BMW", "Toyota");
var_dump($cars);

//array dengan kunci dan berisi beberapa tipe data
$array_of_data = array("string"=>"String
Data", "integer"=>10, "float"=>50.45, "boolean"=>true);
var_dump($array_of_data);
```

2.2.7 Operator PHP

Operator digunakan untuk menentukan sebuah nilai dengan melakukan suatu prosedur, atau dengan suatu operasi dari beberapa nilai. Nilai yang digunakan dalam sebuah operasi disebut operand. Penjumlahan adalah sebuah contoh sederhana dari suatu operasi.

Sebagai contoh : 2 + 4

2 dan 4 adalah sebuah operand. Ekspresi ini akan menghasilkan 6.

Berikut ini merupakan aritmatika pada PHP :

Operator	Aksi	Contoh	Penejelasan
+	Penjumlahan	5 + 9	Menghasilkan nilai : 14
-	Pengurangan	9 - 4	Menghasilkan nilai : 5
*	Perkalian	5 * 6	Menghasilkan nilai : 30
/	Pembagian	20 / 4	Menghasilkan nilai : 5
%	Modulus	9 % 4	Menghasilkan nilai : 1

PHP mengabaikan karakter spasi dalam sebuah operasi. Ekspresi $\$x = 5 + 9$ dengan dengan $\$x=5+9$ akan menghasilkan nilai yang sama. Penggunaan spasi disesuaikan dengan keinginan dari para user.

2.2.8 Compiler dan Intepreter

Compiler Adalah suatu program yang menterjemahkan bahasa program (source code) ke dalam bahasa objek (object code). Compiler menggabungkan keseluruhan bahas program dikumpulkan kemudian disusun kembali. Berbeda dengan compiler, Interpreter menganalisis dan mengeksekusi setiap baris dari program tanpa melihat program secara keseluruhan. Keuntungan dari Interpreter adalah dalam eksekusi yang bisa dilakukan dengan segera. Tanpa melalui tahap kompilasi, untuk alasan ini interpreter digunakan pada saat pembuatan program berskala besar.

Tabel Perbedaan Compiler dan interpreter

Compiler	Interpreter
Menterjemahkan secara keseluruhan	Menterjemahkan Instruksi per instruksi
Bila terjadi kesalahan kompilasi maka source program harus diperbaiki dan dikompilasi ulang	Bila terjadi kesalahan interpretasi dapat Diperbaiki
Dihasilkan Object program	Tidak dihasilkan obyek program
Dihasilkan Executable program	Tidak dihasilkan Executable program
Proses pekerjaan program lebih cepat	Proses pekerjaan program lebih lambat
Source program tidak dipergunakan hanya bila untuk perbaikan saja	Source program terus dipergunakan
Keamanan dari program lebih terjamin	Keamanan dari program kurang terjamin

2.3 Tipe Pemrograman

Dalam Bahasa pemrograman komputer terdapat tipe pemrograman. Tipe pemrograman yang pertama adalah pemrograman terstruktur dan prosedural.

2.3.1 Pemrograman Terstruktur

Pemrograman terstruktur adalah cara pemrosesan data yang terstruktur. Terstruktur dalam: analisa, cara dan penulisan program. Prinsip pemrograman terstruktur adalah mempunyai ciri sebagai berikut:

- Menggunakan rancangan pendekatan dari atas ke bawah (top down design),
- Membagi program ke dalam modul-modul logika yang sejenis,
- Menggunakan sub-program untuk proses-proses sejenis yang sering digunakan,
- Menggunakan pengkodean terstruktur: IF THEN, DO WHILE dan lain-lainnya,
- Menghindari penggunaan perintah GO TO bila tidak diperlukan,
- Gunakan nama-nama bermakna (mnemonic names), dan
- Membuat dokumentasi yang akurat dan berarti.

Dalam perencanaan dan perancangan dari atas ke bawah, kategori dan penyelesaian masalah dimulai dari bagian yang utama kemudian dibagi menjadi bagian yang lebih kecil. Rancangan cara ini memudahkan penulisan, pengujian, koreksi dan dokumentasi program. Tahapan rancangan atas ke bawah dalam pemrograman:

- Tentukan keluaran (output) yang diminta, masukan (input) yang diperlukan dan proses-proses utama yang diperlukan untuk transformasi data.
- Membagi proses utama ke dalam modul-modul fungsional.
- Buat algoritma masing-masing modul, dari modul utama ke subsub modul.

Setiap modul dalam proses rancangan atas ke bawah biasanya dibatasi dalam isi maupun batasan-batasan berikut:

- Setiap modul hanya mempunyai satu masukan dan keluaran
- Setiap modul hanya mewakili satu fungsi program.

Rancangan (design) terstruktur:

- Membagi program menjadi subprogram
- Menekankan fungsionalitas.

- Cocok untuk sistem yang banyak mempunyai fungsi independen. Gaya penulisan program terstruktur: Menggunakan indentasi sehingga jelas struktur dan kontrol program. Memudahkan pembacaan, pemahaman, penelusuran kesalahan dan pembuatan koreksi.

2.3.2 Pemrograman procedural dan terstruktur

Bahasa pemrograman prosedural adalah bahasa pemrograman yang mendukung pembuatan program sebagai kumpulan prosedur. Prosedur - prosedur ini dapat saling memanggil dan dipanggil dari manapun dalam program dan dapat menggunakan parameter yang berbeda-beda untuk setiap pemanggilan. Prosedur adalah bagian dari program untuk melakukan operasi-operasi yang sudah ditentukan dengan menggunakan parameter tertentu. Bahasa pemrograman terstruktur adalah pemrograman yang mendukung abstraksi data, pengkodean terstruktur dan kontrol program terstruktur.

2.4 Algoritma

Pemrograman komputer dan algoritma pemrograman adalah dua hal yang tidak dapat dipisahkan karena pembuatan program komputer akan lebih sulit dan lama tanpa mengetahui dengan pasti bagaimana algoritma penyelesaian masalahnya. Sebelum mengetahui lebih lanjut apa yang dimaksud dengan algoritma pemrograman, kita bahas dahulu apa yang dimaksud dengan pemrograman komputer atau program komputer.

Definisi program komputer adalah sederetan perintah-perintah (instruksi) yang harus dikerjakan oleh komputer untuk menyelesaikan masalah. Deretan perintah-perintah tersebut tidak bisa kita tulis secara sembarangan atau semau kita tetapi harus teratur agar komputer dapat memahami dan memprosesnya dengan baik sehingga permasalahan yang ada dapat diselesaikan dengan baik pula.

Untuk itulah diperlukan algoritma karena definisi dari algoritma itu sendiri menurut Microsoft Press Computer and Internet Dictionary (1998) adalah urutan langkah logis tertentu untuk memecahkan suatu masalah. Yang ditekankan adalah urutan langkah logis, yang berarti algoritma harus mengikuti suatu urutan tertentu, tidak boleh melompat-lompat

dan disusun secara sistematis. Sedangkan yang dimaksud dengan langkah-langkah logis adalah kita harus dapat mengetahui dengan pasti setiap langkah yang kita buat.

Menurut Sjukani (2005), algoritma adalah alur pemikiran dalam menyelesaikan suatu pekerjaan yang dituangkan secara tertulis. Yang ditekankan pertama adalah alur pikiran, sehingga algoritma seseorang dapat berbeda dari algoritma orang lain. Sedangkan penekanan kedua adalah tertulis, yang artinya dapat berupa kalimat, gambar, atau tabel tertentu. Jadi dapat disimpulkan bahwa algoritma lebih merupakan alur pemikiran untuk menyelesaikan suatu pekerjaan atau suatu masalah daripada pembuatan program komputer. Algoritma inilah yang kemudian dijadikan landasan (pedoman) untuk membuat program komputer.

2.4.1 Ciri Algoritma

Oleh karena algoritma digunakan untuk memecahkan suatu permasalahan maka algoritma tersebut harus menghasilkan suatu jawaban atas permasalahan tersebut. Dengan kata lain algoritma harus memiliki paling tidak satu keluaran. Sedangkan masukan dari algoritma dapat nol (tidak ada) atau banyak masukan (data). Yang dimaksud dengan nol masukan adalah jika algoritma itu hanya untuk menampilkan suatu informasi saja. Misalnya output "Hello World" yang sering kita temukan pada tutorial tutorial saat kita baru belajar membuat program dari suatu bahasa pemrograman tertentu. Kedua hal diatas, memiliki paling sedikit satu keluaran dan dapat memiliki nol atau banyak masukan, merupakan dua dari beberapa ciri algoritma.

Tugas algoritma dikatakan selesai kalau algoritma tersebut sudah menghasilkan satu atau lebih jawaban atas permasalahan yang ada. Dengan demikian setelah mengerjakan langkah-langkah penyelesaian masalah, maka algoritma tersebut harus berhenti tidak melakukan proses apapun. Berhenti di sini artinya adalah jika diterjemahkan ke dalam bentuk program dan program dijalankan, maka setelah menghasilkan suatu output, program dapat langsung berhenti atau menunggu instruksi lebih lanjut dari pengguna program seperti mengulang perhitungan lagi, keluar program (menghentikan program), dan lain sebagainya. Dengan demikian ciri ketiga dari algoritma adalah setelah selesai mengerjakan langkah-langkah penyelesaian masalah, algoritma harus berhenti.

Ciri keempat dari algoritma adalah setiap langkah yang dibuat harus dibuat sesederhana mungkin tetapi efektif agar dapat dipahami oleh pemroses (manusia maupun komputer)

sehingga dapat diselesaikan dalam waktu yang singkat serta masuk akal. Sedangkan ciri yang terakhir adalah setiap langkah dalam algoritma harus didefinisikan dengan tepat dan jelas sehingga tidak berarti-dua (ambiguitas) sehingga menimbulkan kesalahan dalam penafsiran oleh pemroses.

2.4.2 Notasi Algoritma

Algoritma mempunyai aturan penulisan sendiri yang disebut dengan notasi algoritma. Notasi algoritma ini tidak tergantung dari spesifikasi bahasa pemrograman tertentu dan komputer yang mengeksekusinya. Hal ini dikarenakan notasi algoritma bukanlah notasi bahasa pemrograman. Notasi algoritma merupakan bahasa universal yang dapat diterima oleh semua bahasa pemrograman yang ada. Oleh sebab itu algoritma yang baik harus dapat diterjemahkan ke dalam bentuk source code dari semua bahasa pemrograman yang ada.

Untuk membuat algoritma dari suatu permasalahan, biasanya digunakan salah satu dari tiga buah notasi algoritma yang dikenal, yaitu uraian kalimat deskriptif, flow chart, atau pseudo code.

2.4.3 Deskriptif Algoritma

Algoritma dengan uraian kalimat deskriptif adalah notasi algoritma yang paling sederhana karena algoritma ini menggunakan bahasa sehari-hari. Untuk permasalahan yang sederhana penggunaan notasi ini sangatlah mudah, akan tetapi untuk permasalahan yang lebih kompleks dan rumit, penggunaan notasi ini akan lebih sulit dan sering kali terjadi ambigu dalam langkah-langkah penyelesaian masalah. Oleh karena itulah untuk kasus-kasus yang lebih kompleks, penggunaan notasi ini jarang sekali bahkan tidak digunakan.

2.5 Flow Chart

Notasi algoritma yang paling banyak digunakan adalah flow chart karena bentuknya yang sederhana dan mudah dipahami. Flow chart (diagram alir) adalah penggambaran secara grafik dari langkah-langkah pemecahan masalah yang harus diikuti oleh pemroses. Flow chart terdiri atas sekumpulan simbol, dimana masing-masing simbol menggambarkan suatu kegiatan tertentu. Flow chart diawali dengan penerimaan masukan (input), pemrosesan masukan, dan diakhiri dengan menampilkan hasilnya (output).

Adapun simbol-simbol yang sering digunakan untuk menyusun flow chart (dalam microsoft visio) adalah sebagai berikut :

1. Masukan

Masukan merupakan kegiatan penerimaan data yang disimbolkan dengan jajaran genjang. Kita dapat menuliskan masukan yang diperlukan pada suatu waktu secara satu per satu maupun secara keseluruhan, akan tetapi untuk alasan efisiensi ruang gambar biasanya masukan dituliskan bersamaan secara keseluruhan.



Gambar 2.1. Simbol masukan

2. Masukan manual

Untuk masukan secara manual yang dimasukkan melalui keyboard, atau perangkat input lainnya seperti barcode reader, kita dapat menggunakan simbol masukan secara manual. Sama dengan simbol masukan, pada simbol masukan manual ini untuk alasan efisiensi ruang gambar biasanya masukan juga dituliskan bersamaan secara keseluruhan.



Gambar 2.2. Simbol masukan manual

3. Proses

Data yang dimasukan kemudian diproses untuk menghasilkan jawaban atas persoalan yang ingin dipecahkan. Kegiatan memproses data ini disimbolkan dengan persegi panjang. Sama seperti simbol pada masukan, penulisan operasi-operasi pada data dapat dilakukan secara satu per satu maupun secara keseluruhan.



Gambar 2.3. Simbol proses

4. Keluaran

Keluaran adalah hasil dari pemrosesan data dan merupakan jawaban atas permasalahan yang ada. Keluaran ini harus ditampilkan pada layar monitor agar dapat dibaca oleh pengguna program. Sama seperti aturan pada simbol-simbol sebelumnya, penulisan hasil pemrosesan data dapat dilakukan secara satu per satu maupun secara keseluruhan.

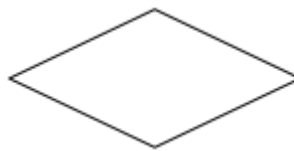


Gambar 2.4. Simbol keluaran

5. Percabangan

Yang dimaksud dengan percabangan disini adalah suatu kegiatan untuk mengecek atau memeriksa suatu keadaan apakah memenuhi suatu kondisi tertentu atau tidak. Jadi dalam

percabangan ini, kita harus menuliskan kondisi apa yang harus dipenuhi oleh suatu keadaan. Hasil dari pemeriksaan keadaan ini adalah YA atau TIDAK. Jika pemeriksaan keadaan menghasilkan kondisi yang benar, maka jalur yang dipilih adalah jalur yang berlabel YA, sedangkan jika pemeriksaan keadaan menghasilkan kondisi yang salah, maka jalur yang dipilih adalah jalur yang berlabel TIDAK. Berbeda dengan aturan pada simbol-simbol sebelumnya, penulisan kondisi harus dilakukan secara satu per satu (satu notasi percabangan untuk satu kondisi).



Gambar 2.5. Simbol percabangan

6. Sub rutin

Sub rutin adalah suatu bagian dalam program yang dapat melakukan (atau diberi) tugas tertentu. Jadi sub rutin merupakan “program kecil” yang menjadi bagian dari suatu program yang besar. Sub rutin ada dua macam, yaitu prosedur (procedure) dan fungsi (function). Perbedaan antara keduanya adalah setelah dipanggil prosedur tidak mengembalikan suatu nilai sedangkan fungsi selalu mengembalikan suatu nilai.

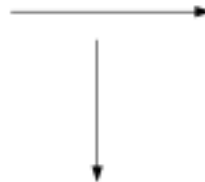
Sub rutin memiliki suatu flow chart yang berdiri sendiri diluar flow chart utama. Jadi dalam simbol sub rutin, kita cukup menuliskan nama sub rutusnya saja, sama seperti jika kita melakukan pemanggilan terhadap suatu sub rutin pada program utama (main program) yang akan anda pelajari pada bagian atau bab lain pada buku ini. Aturan penulisan simbol sub rutin sama dengan simbol percabangan, yaitu penulisan nama sub rutin dilakukan secara satu per satu.



Gambar 2.6. Simbol sub rutin

7. Arah aliran

Arah aliran merupakan jalur yang harus diikuti dan merupakan garis penghubung yang menghubungkan setiap langkah pemecahan masalah yang ada dalam flow chart. Arah aliran ini disimbolkan dengan anak panah.



Gambar 2.7. Simbol arah aliran

8. Terminator

Terminator berfungsi untuk menandai titik awal dan titik akhir dari suatu flow chart. Simbol terminator ini diberi label MULAI untuk menandai titik awal dari flow chart dan label SELESAI untuk menandai titik akhir dari flow chart. Jadi dalam sebuah flow chart harus ada dua simbol terminator, yaitu simbol terminator untuk MULAI dan SELESAI.

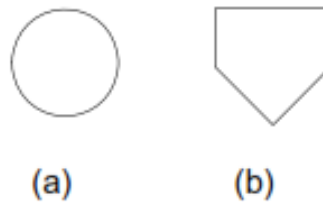


Gambar 2.8. Simbol terminator

9. Konektor

Konektor berfungsi untuk menghubungkan suatu langkah dengan langkah lain dalam sebuah flow chart dengan keadaan on page atau off page. Yang dimaksud dengan konektor

on page adalah konektor yang digunakan untuk menghubungkan suatu langkah dengan langkah lain dalam satu halaman. Sedangkan konektor off page adalah konektor untuk menghubungkan suatu langkah dengan langkah lain dalam halaman yang berbeda. Konektor ini digunakan apabila ruang gambar yang kita gunakan untuk menggambar flow chart tidak cukup luas untuk memuat flow chart secara utuh, jadi perlu dipisahkan atau digambar di halaman yang berbeda.



Gambar 2.9. Simbol konektor on page (a) dan off page (b)

10. Dokumen

Dokumen merupakan tampilan data secara fisik yang dapat dibaca oleh manusia. Data ini biasanya merupakan hasil pemecahan masalah (informasi) yang telah dicetak (print out).



Gambar 2.10. Simbol dokumen

2.6 Pseudo Code

Pseudo code adalah algoritma yang bentuknya (strukturnya) sangat mirip dengan bahasa pemrograman khususnya bahasa pemrograman terstruktur seperti pascal. Kemiripan ini merupakan keuntungan dari pseudo code karena implementasi atau penerjemahan algoritma ke dalam source code suatu bahasa pemrograman sangatlah mudah meskipun penggunaannya tidak sepopuler flow chart.

Dalam penulisannya, pseudo code harus terdiri dari tiga bagian, yaitu :

1. Judul algoritma

Bagian yang terdiri atas nama algoritma dan penjelasan (spesifikasi) dari algoritma tersebut. Nama sebaiknya singkat dan menggambarkan apa yang dapat dilakukan oleh algoritma tersebut.

2. Deklarasi

Bagian untuk mendefinisikan semua nama yang digunakan di dalam program. Nama tersebut dapat berupa nama tetapan, peubah atau variabel, tipe, prosedur, dan fungsi.

3. Deskripsi

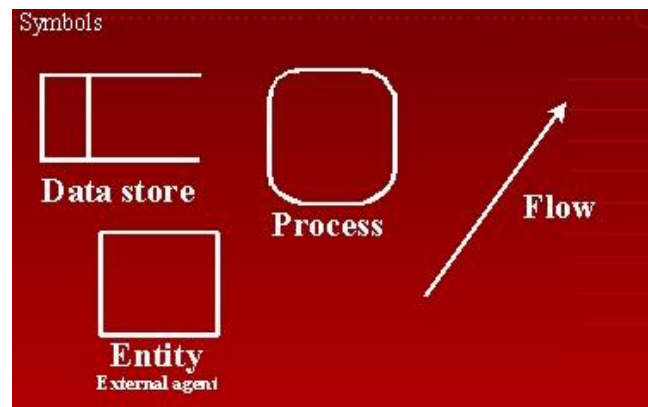
Bagian ini berisi uraian langkah - langkah penyelesaian masalah

DATA FLOW DIAGRAM (DFD)

Tentang DFD

- DFD bukan flowchart
- Proses dalam DFD bisa berjalan secara paralel
- DFD menggambarkan aliran data dalam sebuah sistem
- DFD adalah Data yang tersimpan dan proses dengan proses yang terhubung dengan data tersebut
- Tidak ada loop ataupun cabang dalam DFD
- DFD menggambarkan semua proses, meskipun proses tersebut terjadi dalam waktu yang berbeda.

Simbol DFD



Entity

- Digambarkan dengan simbol bujur sangkar.
- Merupakan sumber atau tujuan dari aliran data.
- Merupakan lingkungan luar dari sistem.
- Bisa menggambarkan secara fisik, seseorang atau sekelompok orang atau system lain.

Entity

- Kadang-kadang perlu untuk menduplikasinya untuk menghindari anak panah yang simpang siur.
- Ditandai dengan garis diagonal disudut kanan bawah yang menyatakan kalau entiti tersebut lebih dari satu.

Aliran Data

- Menggambarkan aliran data dari suatu proses ke proses lainnya.
- Merepresentasikan dengan menggunakan anak panah.
- Nama proses ditulis untuk menjelaskan arti dalam aliran tersebut dan ditulis untuk mengidentifikasi aliran tersebut.
- Aliran data dapat menyebar atau menyatu

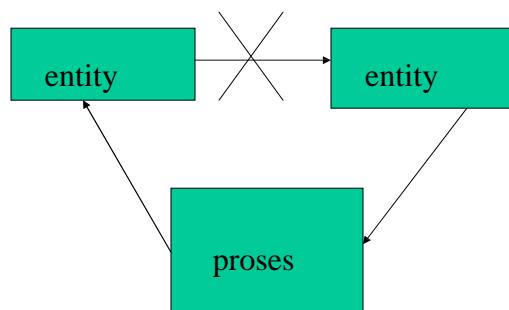
Proses

- Adalah fungsi yang mentransformasikan data secara umum.
- Karena proses adalah suatu pekerjaan, maka untuk menamai sebuah proses mulailah dengan kata kerja dan diikuti objek.

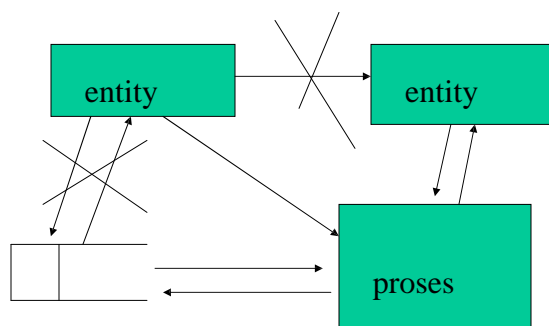
Storage/ Penyimpanan

- Komponen yang berfungsi untuk menyimpan data/ file adalah fungsi yang mentransformasikan data secara umum.

Peraturan penting dalam DFD



Peraturan penting dalam DFD



Peraturan penting dalam DFD

- Semua objek harus mempunyai nama
- Aliran data harus diawala dan diakhiri oleh proses
- Semua aliran data harus mempunyai tanda panah.

Level dalam DFD

- Model ini menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu dengan yang dengan aliran dan penyimpan data.
- Bisa terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus bisa merepresentasikan proses tersebut kedalam spesifikasi proses yang jelas.

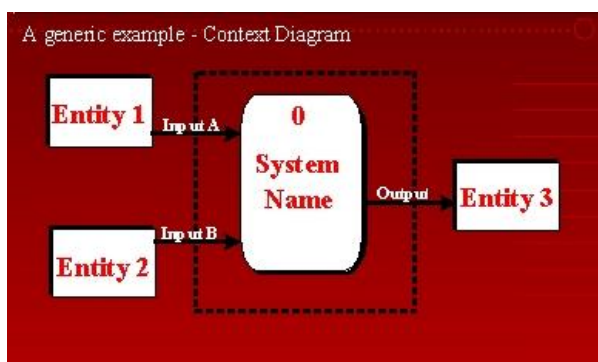
Level dalam DFD

- Dalam DFD level dimulai dari level 0 kemudian turun ke DFD level 1, 2 dan seterusnya sesuai dengan kebutuhan.
- Dalam penurunan tidak semua bagian sistem harus diturunkan dengan jumlah level yang sama.

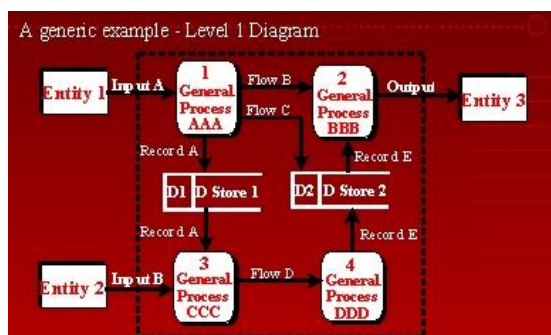
Level dalam DFD

- Aliran data dalam yang masuk dan keluar dalam level X harus harus berhubungan dengan aliran data yang masuk dan keluar pada level X +1 yang mendefinisikan proses pada level X tersebut.

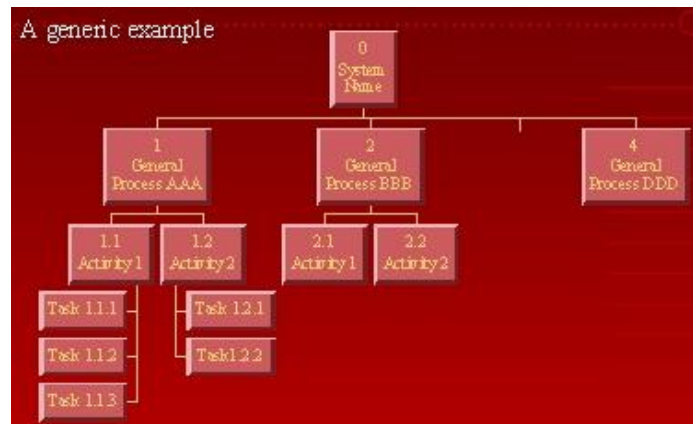
DFD Level 0



DFD-Level-1

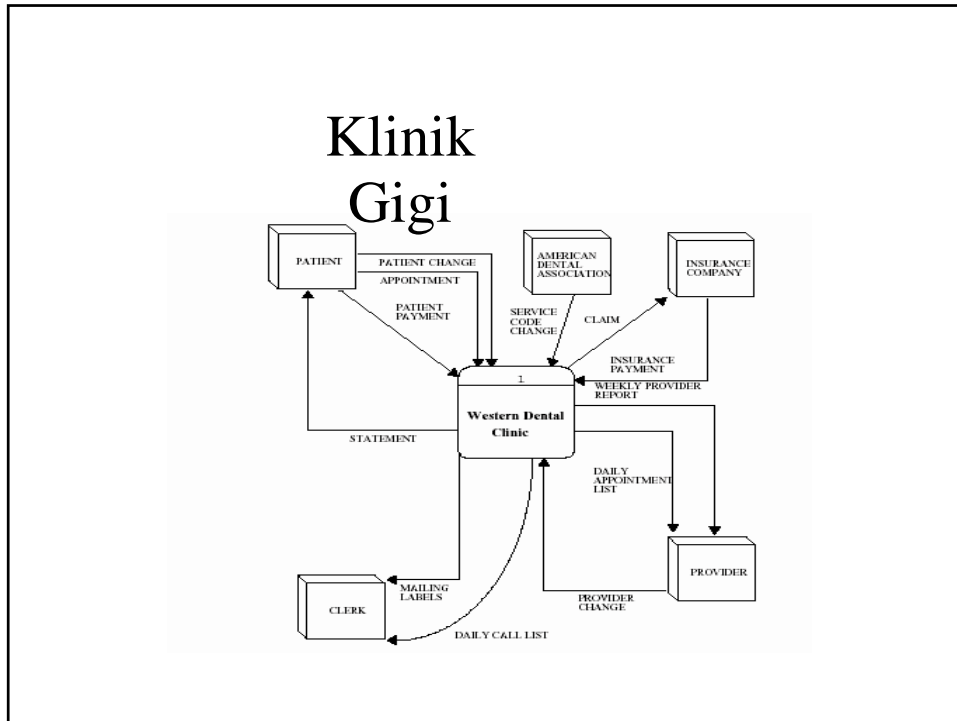


Decomposition Diagram



Logical Processes

- Cari
 - Function: Set of related and ongoing activities
 - Events :Logical units of work performed as whole
 - Elementary Processes : discrete, detail activities required to complete the response to an event



DAFTAR PUSTAKA

Nama, judul, penerbitan, tempat, tahun

1. Suprpto, Bahasa Pemrograman, Direktorat Pembinaan Sekolah Menengah Kejuruan, 2008
2. Modul Pemrograman Web berbasis PHP,
<http://wa2n.staff.uns.ac.id/files/2009/03/modulphp.pdf>, diakses tanggal 2 November 2017
3. Buku Bahasa Pemrograman
<http://staff.uny.ac.id/sites/default/files/penelitian/Drs.%20Totok%20Sukardiyono,%20M.T./Buku%20Bahasa%20Pemrograman%20Lengkap.pdf>, diakses tanggal 2 November 2017

BAB III SOFTWARE

3.1 Pengertian Dasar Pemrograman

Dasar Program adalah rangkaian instruksi-instruksi dalam bahasa komputer yang disusun secara logika dan sistematis. Pemrograman adalah suatu kumpulan urutan perintah ke komputer untuk mengerjakan sesuatu, dimana instruksi tersebut menggunakan bahasa yang dimengerti oleh komputer atau dikenal dengan bahasa pemrograman. Bahasa Pemrograman tingkat dasar seperti: Bahasa Mesin, Bahasa *Assembly* :

1. Bahasa Pemrograman tingkat tengah seperti: Bahasa *C*, Bahasa *FORTH*.
2. Bahasa Pemrograman tingkat tinggi seperti: Bahasa *Pascal*, *FORTRAN*, *COBOL*, dan lain-lain.

Bahasa pemrogram tingkat dasar, sebagai hirarki pertama, disebut juga bahasa generasi pertama, bahasa ini sangat tergantung pada jenis *CPU* yang dipakai oleh komputer itu. Bahasa ini sangat sulit dipelajari, karena sifatnya yang sangat tergantung pada mesinnya itu (*machine dependent*). Untuk menguasai bahasa ini, harus dipelajari dan dikuasai teknologi dan arsitektur komputer, matematika diskrit, elektronika, dan lain-lain. Bahasa ini masih menggunakan simbol-simbol yang bersifat *menemonic*.

3.2 Pengertian PHP

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*. Selain itu, *PHP* juga bisa digunakan sebagai bahasa pemrograman umum . *PHP* disebut bahasa pemrograman server side karena *PHP* diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti *Java Script* yang diproses pada *web browser* (*client*).

Pada awalnya *PHP* merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, *PHP* digunakan untuk membuat *website* pribadi. Dalam beberapa tahun

perkembangannya, *PHP* menjelma menjadi bahasa pemrograman *web* yang *powerfull* dan tidak hanya digunakan untuk membuat halaman *web* sederhana, tetapi juga *website* populer yang digunakan oleh jutaan orang seperti *wikipedia*, *wordpress*, *joomla* dan lain - lain.

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*. Selain itu, *PHP* juga bisa digunakan sebagai bahasa pemrograman umum (*wikipedia*). *PHP* dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The *PHP Group*. Situs resmi *PHP* beralamat di <http://www.php.net>. *PHP* disebut bahasa pemrograman *server side* karena *PHP* diproses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *Java Script* yang diproses pada *web browser (client)*. *PHP* dapat digunakan dengan gratis dan bersifat *Open Source*. *PHP* dirilis dalam lisensi *PHP License*, sedikit berbeda dengan lisensi *General Public License (GPL)* yang biasa digunakan untuk proyek *Open Source*.

Kemudahan dan kepopuleran *PHP* sudah menjadi standar bagi programmer *web* di seluruh dunia. Menurut *wikipedia* pada februari 2014, sekitar 82% dari *web server* di dunia menggunakan *PHP*. *PHP* juga menjadi dasar dari aplikasi *CMS (Content Management System)* populer seperti *Joomla*, *Drupal*, dan *WordPress*.

3.3 Pengertian MySQL

MySQL adalah *database server* yang dapat menyimpan sejumlah basis data yang dapat diakses oleh program *client (MySQL Comand Line Client)*, maupun program aplikasi), pengaksesan data bisa dilakukan dengan bahasa *SQL (Structure Query Language)*. Basis data bisa dibuat melalui *command prompt* atau melalui *shell* secara langsung setelah masuk pada *subdirektori bin*. Untuk membuat suatu basis data, user yang bersangkutan harus mempunyai hak akses untuk membuat basis data. Setelah dibuat suatu database bisa digunakan atau dihapus bila tidak diperlukan, yang selanjutnya dapat dibuat tabel basis datanya, dimana satu skema basis data bisa terdiri atas banyak subskema (tabel basis data). *Root* adalah *superuser* database server *MySQL*, *root* berkuasa dalam mengelola manajemen basis data.

3.4 Pengertian *NetBeans*

NetBeans mengacu pada kedua kerangka *platform* untuk aplikasi desktop *Java*, dan sebuah lingkungan pengembangan terpadu (*IDE*) untuk pengembangan dengan *Java*, *JavaScript*, *PHP*, *Python*, *Ruby*, *Groovy*, *C*, *C ++*, *Scala*, *Clojure*, dan lain-lain. *NetBeans* IDE ditulis dalam *Java* dan berjalan di mana-mana di mana *JVM* yang diinstal, termasuk *Windows*, *Mac OS*, *Linux*, dan *Solaris*. Sebuah *JDK* diperlukan untuk pengembangan fungsionalitas *Java*, tetapi tidak diperlukan untuk pembangunan di bahasa pemrograman lain. *Platform NetBeans* memungkinkan aplikasi untuk dikembangkan dari satu set komponen perangkat lunak modular yang disebut modul. Aplikasi berbasis *platform NetBeans* (termasuk *IDE NetBeans*) dapat diperpanjang oleh pengembang pihak ketiga. Merancang antarmuka merupakan bagian yang paling penting dari merancang sistem. Biasanya hal tersebut juga merupakan bagian yang paling sulit, karena dalam merancang antarmuka harus memenuhi tiga persyaratan: sebuah antarmuka harus sederhana, sebuah antarmuka harus lengkap, dan sebuah antar muka harus memiliki kinerja yang cepat. Elemen-Element dalam perancangan adalah :

1. Mendefinisikan konsep.
2. Mengumpulkan kebutuhan-kebutuhan pengguna dan mendefinisikan desain secara konseptual.
3. Mengevaluasi konsep.
4. Mengevaluasi konseptual desain tersebut.
5. Merancang.
6. Mengevaluasi *prototype*. Menandai dan memperbaiki masalah-masalah yang ditemukan.
7. Pengembangan

Melakukan pengujian secara berkala terhadap desain yang lebih dahulu dibuat dan desain yang paling terakhir dibuat. Menandai dan memperbaiki masalah-masalah yang ditemukan.

3.5 Pengertian *Java*

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, *Java* adalah nama untuk sekumpulan teknologi untuk

membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Kita lebih menyukai menyebut *Java* sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena *Java* lebih lengkap dibanding sebuah bahasa pemrograman konvensional. Teknologi *Java* memiliki tiga komponen penting, yaitu: *Programming-language specification*, *Application-programming interface*, *Virtual-machine specification*. *Java API* terdiri dari tiga bagian utama: *Java Standard Edition (SE)*, sebuah standar *API* untuk merancang aplikasi desktop dan applets dengan bahasa dasar yang mendukung grafis, *M/K*, keamanan, konektivitas basis data dan jaringan.

Java Enterprise Edition (EE), sebuah inisiatif *API* untuk merancang aplikasi server dengan mendukung untuk basis data. *Java Micro Edition (ME)*, sebuah *API* untuk merancang aplikasi yang jalan pada alat kecil seperti telepon genggam, komputer genggam dan pager. *Java Virtual Machine (JVM)* adalah sebuah spesifikasi untuk sebuah komputer abstrak. *JVM* terdiri dari sebuah kelas pemanggil dan sebuah interpreter *Java* yang mengeksekusi kode arsitektur netral. Kelas pemanggil memanggil file class dari kedua program *Java* dan *Java API* untuk dieksekusi oleh *Java Interpreter*. *Java* interpreter mungkin sebuah perangkat lunak *interpreter* yang menterjemahkan satu kode *byte* pada satu waktu, atau mungkin sebuah *just-in-time (JIT)* kompiler yang menurunkan *bytecode* arsitektur netral kedalam bahasa mesin untuk *host computer*. Sistem Operasi *Java* Sistem operasi biasanya ditulis dalam sebuah kombinasi dari kode bahasa *C* dan *assembly*, terutama disebabkan oleh kelebihan performa dari bahasa tersebut dan memudahkan komunikasi dengan perangkat keras.

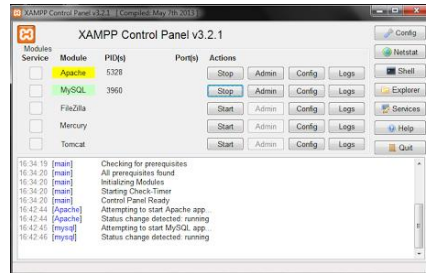
3.6 Pengertian XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari *X* (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari *web* resminya.

1. Cara menggunakan XAMPP

a. Klik aplikasi XAMPP

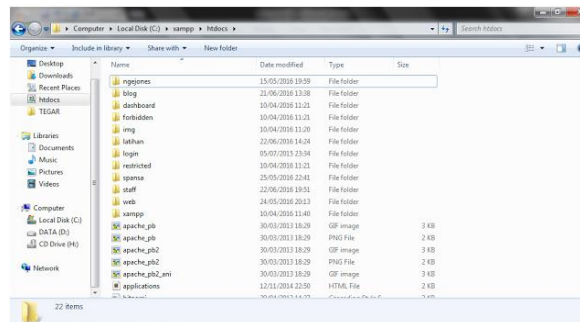
Akan muncul tampilan seperti dberikut, klik *start* pada *Action* bagian Apache dan Mysql



b. Mencari alamat *localhost* pada *web*



c. Penyimpanan pada XAMPP terdapat pada htdoc



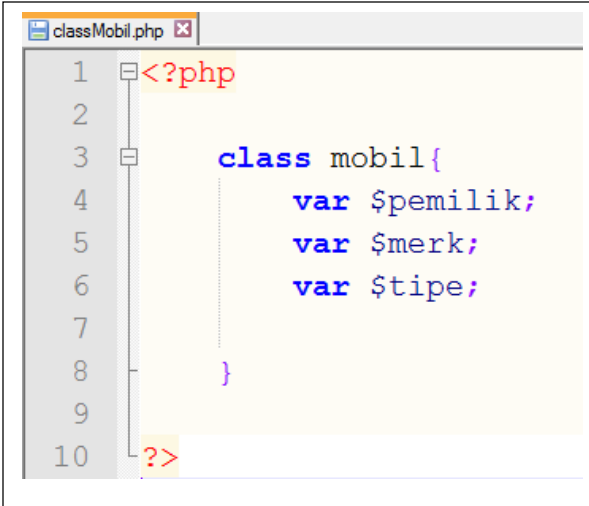
PHP dan BOOSTRAP

BAB I CLASS DAN FUNCTION

INTRODUCTION

merupakan suatu blueprint atau cetakan untuk menciptakan suatu dengan cepat dari object. class juga merupakan grup suatu object dengan kemiripan attributes/properties, behaviour dan relasi ke object lain.

Buatlah struktur class seperti berikut, isikan variable pemilik, merk dan tipe, simpan dan berinama file tersebut dengan nama classMobil.php



```
1 <?php
2
3 class mobil{
4     var $pemilik;
5     var $merk;
6     var $tipe;
7
8 }
9
10 ?>
```

Selanjutnya file tersebut kita selesaikan agar bias berfungsi dengan baik dan juga agar bias digunakan pada file yang lain, karena class akan di panggil kedalam file yang berhubungan.



```
1 <?php
2
3 class mobil{
4     var $pemilik;
5     var $merk;
6     var $tipe;
7
8     function nyalakan_mobil(){
9         return "Mobil menyala dengan bensin full";
10    }
11
12    function matikan_mobil(){
13        return "Mesin mobil mati";
14    }
15
16 }
17
18 ?>
```

Coding diatas belum bias dieksekusi, karena coding diatas hanya menyediakan class dengan fungsi namun belum kita gunakan fungsi dari class tersebut, cara menggunakan class tersebut adalah seperti berikut.

```

18      //buat objek dari class
19      $its = new mobil();
20
21      //property class
22      $its->pemilik   = 'Dedy';
23      $its->merk      = 'Toyota';
24      $its->tipe      = 'Jeep';
25
26      //tampilan dari class
27      echo $its->pemilik."<br>";
28      echo $its->merk."<br>";
29      echo $its->tipe."<br>";
30
31      ?>

```

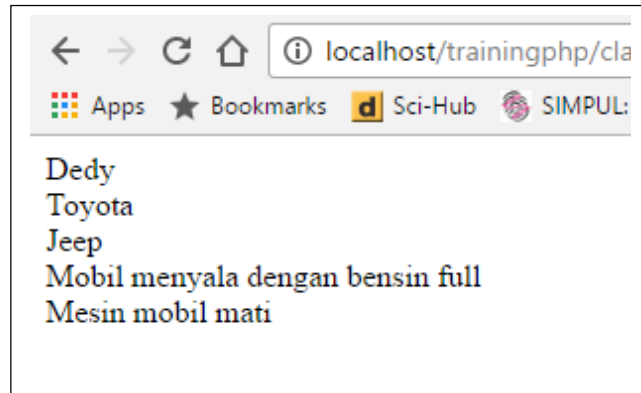
\$its merupakan property alias yang akan digunakan seterusnya untuk menggunakan fungsi dari class, karena \$its merupakan deklarasi dari class yang kita gunakan, kemudian kita akan menggunakan method yang ada didalam class yang tadi sudah kita buat, yaitu method function nyalakan dan matikan mobil.

```

26      //tampilan dari class
27      echo $its->pemilik."<br>";
28      echo $its->merk."<br>";
29      echo $its->tipe."<br>";
30
31      //penggunaan method
32      echo $its->nyalakan_mobil()."<br>";
33      echo $its->matikan_mobil()."<br>";
34
35      ?>

```

Hasil eksekusi dari file diatas adalah seperti berikut, dimana file tersebut mengenerate keseluruhan class dan function yang didalamnya ada method.



ENKAPSULASI OBJEK

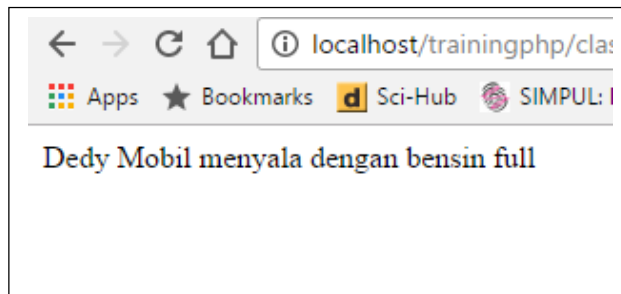
Untuk membatasi hak akses yang akan digunakan didalam property dan method yang ada dalam sebuah class disediakan 3 kata kunci yaitu public, protected dan private kata kunci ini diletakkan sebelum property dideklarasikan, class dibawah ini adalah class dengan **function public**.

```

1  <?php
2
3  class mobil{
4
5      //buat fungsi public
6      public $pemilik;
7
8      public function nyalakan_mobil(){
9          return "Mobil menyala dengan bensin full";
10     }
11
12 }
13
14 $its = new mobil();
15
16 $its->pemilik = 'Dedy ';
17
18 echo $its->pemilik.$its->nyalakan_mobil();
19

```

Class diatas bersifat public artinya class tersebut bias diakses dari mana saja asalkan class diikut sertakan dalam include hasil eksekusi dari class diatas adalah



Selanjutnya kita akan membuat class dengan **function protected** agar fungsi yang digunakan ada parentnya atau induknya yang akan menandakan bahwa fungsi tersebut terafiliasi dengan sebuah class.

```

1  <?php
2
3  class mobil{
4
5      //class protected
6      protected $pemilik;
7
8      //method class
9      protected function nyalakan_mobil(){
10         return "Mobil menyala dengan bensin full";
11     }
12
13 }
14
15 $its = new mobil;
16
17 echo $its->pemilik = 'Dedy';
18

```

Hasil eksekusi dari class diatas akan menghasilkan error, karena inisiasi pemilik tidak dibuatkan afiliasi class yang akan menampung method dari class tersebut.

Fatal error: Cannot access protected property mobil::\$pemilik

Kita harus membuat afiliasi dari class tersebut agar bias digunakan sebagaimana fungsinya, dan fungsi dari class tersebut harus mengandung class induknya yaitu class mobil class tersebut dinakan penurunan.

```
1 <?php
2
3 class mobil{
4
5     //class protected
6     protected $pemilik = 'Dedy ';
7
8     //method class
9     protected function nyalakan_mobil(){
10         return "Mobil manyala dengan bensin full";
11     }
12
13 }
14 //bagian adalah class yang akan digunakan,
15 //extends mobil adalah induk
16 class bagian extends mobil{
17     public function pemilik(){
18         return $this->pemilik;
19     }
20     public function nyala(){
21         return $this->nyalakan_mobil();
22     }
23 }
24
25 $its = new bagian;
26 echo $its->pemilik().$its->nyala();
27
28 ?>
```

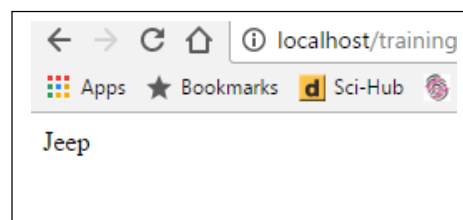
Class diatas menandakan bahwa class mobil sudah diturunkan dan menghasilkan class bagian yang mana class tersebut terafiliasi dengan class mobil, didalam class bagian dibuat fungsi yang bersifat public dan hanya class bagian saja yang bias mengaksesnya, tampilakn eksekusi file diatas adalah

Dedy Mobil manyala dengan bensin full

Selanjutnya adalah penggunaan dari **function private**, class dengan function private merupakan class yang hanya bisa diakses oleh class itu sendiri tidak bisa class private akan dibantu class public untuk mengakses file yang akan ditampilkan.

```
1 <?php
2
3 class mobil{
4
5     private $jenis = 'Jeep';
6
7     public function tampilkan_merk(){
8         return $this->jenis;
9     }
10
11 }
12
13 class bagian extends mobil{
14     public function tampilkan_merk(){
15         return $this->jenis;
16     }
17 }
18
19 $its = new mobil();
20
21 echo $its->tampilkan_merk();
22
23 ?>
```

Hasil eksekusi dari file class diatas adalah seperti berikut



VARIABEL OBJEK DALAM PROGRAM

Didalam pemrograman web penggunaan variable dalam obejk menggunakan \$this, string ini merupakan settingan default, cara penggunaan \$this adalah seperti berikut

```

1  <?php
2
3  class komputer{
4
5      public $pemilik;
6      public $merk;
7
8      //buat method
9      public function nyalakan_komputer(){
10         return "Nyalakan komputer ".$this->merk." punya ".$this->pemilik;
11     }
12
13     public function matikan_komputer(){
14         return "Matikan komputer ".$this->merk." punya ".$this->pemilik;
15     }
16
17     public function restart(){
18         $nyalakan = $this->nyalakan_komputer();
19         $matikan = $this->matikan_komputer();
20         $restart = $matikan."<br>".$nyalakan;
21         return $restart;
22     }
23 }
24
25
26 $komputer = new komputer();
27 $komputer->pemilik = 'Dedy';
28 $komputer->merk = 'Lenovo';
29
30 echo $komputer->nyalakan_komputer();
31
32 ?>

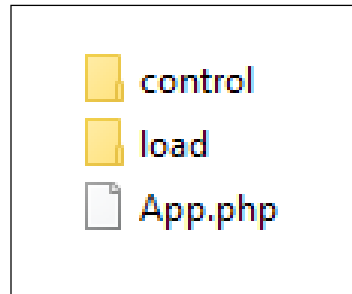
```

Lanjutkan coding diatas sehingga menampilkan tampilan seperti berikut

Apps ★ Bookmarks d Sci-Hub SIMPUL: IP DOMAIN serverbox Word to PDF Converter

Nyalakan komputer Lenovo punya Dedy
 Matikan komputer Lenovo punya Dedy
 Matikan komputer Lenovo punya Dedy kemudian Nyalakan komputer Lenovo punya Dedy

Load file dengan class dengan menggunakan base dir yang akan kita manipulasi sehingga bisa menampilkan apa yang kita inginkan pada halaman yang diload, buatlah dua folder seperti berikut



Kemudian langkah berikutnya adalah membuat file App.php yang berisikan class untuk memanipulasi dari loader yang akan meload file yang dipanggil

```
1 <?php
2
3 define("BASE_PATH", dirname(__FILE__));
4 $loader = (isset($_GET['act'])) ? $_GET['act'] : 'Loader';
5
6 class Loader{
7
8     public function index(){
9         @include (BASE_PATH.'/load/index.php');
10    }
11
12
13 }
14
15 if($loader !== 'Loader') {
16     include (BASE_PATH."/control/$loader.php");
17 }
18 $those = new $loader();
19 $those->index();
20 ?>
```

Selanjutnya membuat class turunan dinamis sesuai dengan kebutuhan dari system, class yang dibuat adalah class turunan yang mana induk dari class tersebut adalah class Loader, buatlah file index.php dan simpan di folder load sebagai penanda kalau defaultnya adalah ke file index.php di folder load, isikan dengan isian ini adalah file index, selanjutnya buat class contoh.php simpan di folder control.

```

1  <?php
2
3      class contoh extends Loader {
4          public function index() {
5              @include('load/contoh.php');
6          }
7      }
8
9  ?>

```

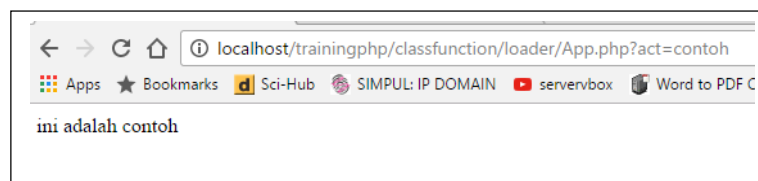
File class diatas adalah untuk mengontrol file yang akan di load didalam folder load, kemudian buatah file dengan nama contoh.php kemudian simpan didalam folder load.

```

1  ini adalah contoh dari folder load

```

Untuk mengakses file tersebut ketikan setelah App.php kemudian berilah tanda ? sehingga menjadi seperti berikut App.?act=contoh



TUGAS

1. Buatlah Class yang bersifat public dengan nama class informatika, dan buatlah 5 variabel yang diantaranya berisikan jurusan, dosen, lantai, ruangan, mata kuliah.
2. Buatlah class turunan dari class yang sudah dibuat yang bersifat protected dan public

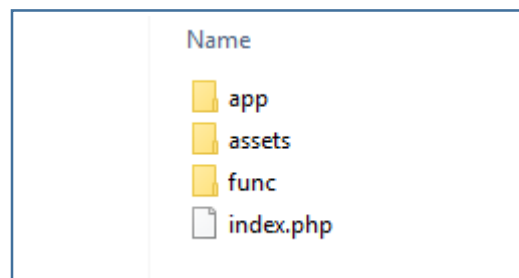
BAB II WEB TEMPLATING CONTROLLER DAN VIEW

SISTEM TEMPLATE

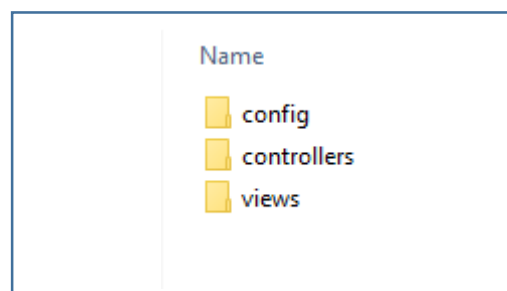
Ketika kita menggunakan aplikasi berbasis Web maka layout yang ada pada aplikasi tersebut harus di load dengan satu layout yang sama tujuannya adalah untuk memudahkan user, pada pemrograman web biasanya kita akan membagi beberapa fungsi untuk menyelesaikan alur kerja dengan tujuan tertentu baik itu input proses ataupun output, kemudian pada bagian layout biasanya akan dibagi menjadi beberapa kolom, umumnya adalah kolom header, konten dan footer, masing-masing bagian tersebut akan membutuhkan HTML dan CSS yang akan digunakan untuk mendesain tampilannya, untuk membentuk sebuah tema atau template.

MEMBUAT STRUKTUR TEMPLATE

Buatlah folder dengan nama pelatihan yang didalamnya berisikan beberapa folder yang nantinya akan kita gunakan sebagai komponen yang akan di load dalam web aplikasi.



Kemudian buatlah beberapa folder didalam folder app, folder config adalah untuk mengkonfigurasi koneksi dengan data base kemudian pada folder controllers adalah sebagai pengontrol file yang akan ditaruh pada folder views, selanjutnya folder views merupakan folder yang didalamnya diletakkan beberapa file utama yang akan ditampilkan.



MEMBUAT CLASS LOADER

Buatlah file index.php didalam folder webroot, folder index berguna untuk menload keseluruhan dari file yang sudah di dibuat secara dinamis tinggal memanggil alamat yang akan dituju, buat file index.php seperti berikut.

```

1 <html>
2   <title>Aplikasi PHP</title>
3   <head>
4   </head>
5   <h1>Header</h1><br>
6
7
8   <hr>
9   Footer
10 </html>

```

Kemudian tambahkan engine dan tambahkan tag body pada file index.php tersebut, gunanya adalah untuk mengetahui bahwa konten yang akan diload hanya yang berada pada tag body.

```

1 <html>
2   <title>Aplikasi PHP</title>
3   <head>
4   </head>
5   <h1>Header</h1><br>
6   <body>
7     <?php
8       define("BASE_PATH", dirname(__FILE__));
9       $loader = (isset($_GET['act'])) ? $_GET['act'] : 'Loader';
10      @include(BASE_PATH.'/app/App.php');
11      if($loader !== 'Loader'){
12        @include(BASE_PATH."/app/controllers/".$loader.".php");
13      }
14      $those = new $loader();
15      $those->index();
16    ?>
17   </body>
18
19
20   <hr>
21   Footer
22 </html>

```

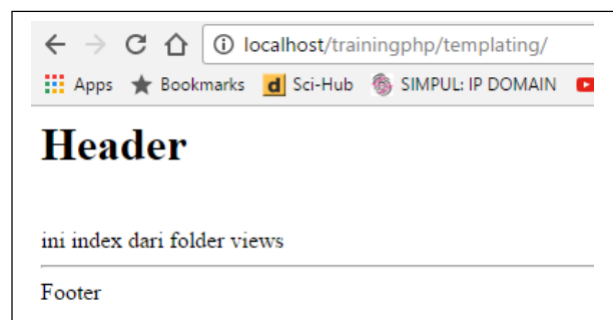
Kemudian buatlah class yang akan mengontrol semua file secara public kemudian simpan file tersebut dengan nama App.php file tersebut mengandung class yang nantinya akan diturunkan menjadi beberapa class yang ditaruh dalam folder controllers

```

1 <?php
2
3     class Loader {
4         public function index() {
5             @include (BASE_PATH."/views/index.php");
6         }
7     }
8
9 ?>

```

File tersebut adalah class utama dari system, secara default class tersebut akan meload file index.php yang disimpan dalam folder views hasil eksekusi dari coding diatas adalah seperti berikut.



semua folder yang akan ditampilkan ditaruh didalam folder views, selanjutnya buatlah file contoh.php untuk menguji system kita apakah sudah merespon dengan baik class yang baru saja kita buat dan simpanlah pada folder views.

```

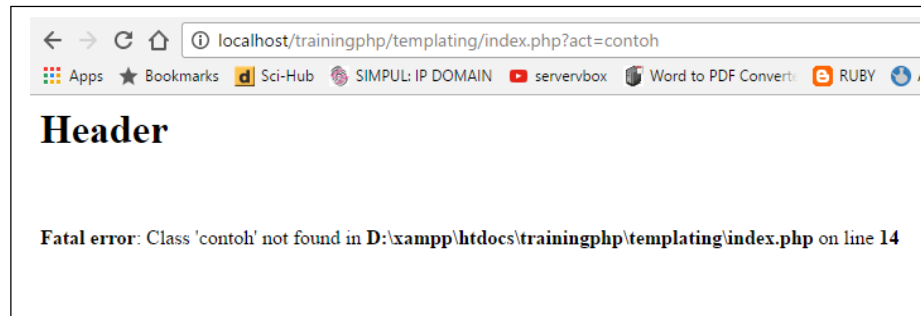
1 ini adalah file contoh yang di load
2 dari folder views

```

Isikan contoh tersebut seperti pada gambar diatas, kemudian panggil file tersebut melalui system dengan mengetikkan alamat pada address bar <http://localhost/templating/index.php?act=contoh>

PEMANGGILAN VARIABEL DI ADDRESS BAR

Hasil dari eksekusi file diatas akan menghasilkan error yaitu class not found, dari error tersebut langkah yang akan kita lakukan adalah membuat file contoh.php yang kemudian kita isikan class turunan dimana didalamnya berisikan public function.



Fatal error on line 14 menandakan bahwa class tidak ditemukan, keterangan tersebut akan selalu muncul jika pada system kita tidak ditemukan class yang sesuai, buatlah file contoh.php dan simpanlah pada folder controllers.

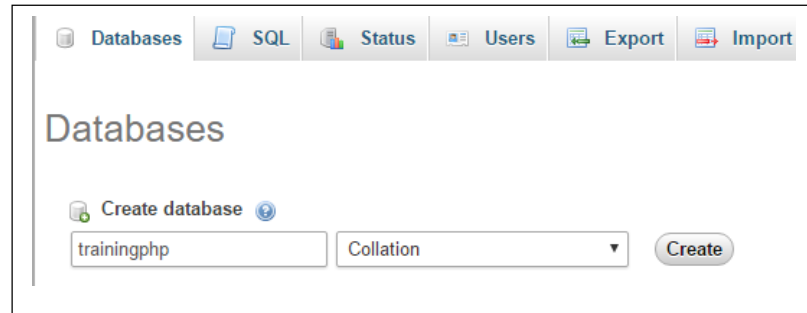
```

1  <?php
2
3      class contoh extends Loader {
4          public function index() {
5              @include(BASE_PATH."/app/views/contoh.php");
6          }
7      }
8
9  ?>

```

Kemudian simpanlah file tersebut dalam folder controllers file contoh.php tersebut merupakan file turunan dari class Loader yang mana class tersebut terafiliasi dengan class induk yang kita gunakan dalam system ini.

Setelah itu buatlah data base untuk menampung data didalam system yang akan digunakan untuk input dan output. Buatlah database dengan nama trainingphp.



Setelah itu ketikkan syntax sql berikut ini, syntax dibawah ini adalah untuk membuat table pengguna yang akan menggunakan system ini.

```

1 create table users(
2
3     id int(5)auto_increment,
4     nama varchar(55),
5     email varchar(55),
6     password varchar(55),
7     role int(2),
8
9     primary key(id)
10
11 )

```

Kemudian klik go atau kirim untuk mengeksekusi syntax tersebut, langkah selanjutnya adalah membuat file config yang akan kita hubungkan ke database yang sudah kita buat sebelumnya, Hasil eksekusi dari syntax diatas adalah seperti berikut

The screenshot shows the 'Structure' view of a table in a database management interface. The table has five columns: #, Name, Type, Collation, and Attributes. The columns are: 1 id (int(5)), 2 nama (varchar(55) latin1_swedish_ci), 3 email (varchar(55) latin1_swedish_ci), 4 password (varchar(55) latin1_swedish_ci), and 5 role (int(2)).

#	Name	Type	Collation	Attributes
1	id	int(5)		
2	nama	varchar(55)	latin1_swedish_ci	
3	email	varchar(55)	latin1_swedish_ci	
4	password	varchar(55)	latin1_swedish_ci	
5	role	int(2)		

File config untuk database yang akan menghubungkan aplikasi kita dengan database mysql adalah seperti berikut.

```

1 <?php
2
3     $db      = 'trainingphp';
4     $conn    = mysql_connect('localhost', 'root', '');
5     mysql_select_db($db, $con) or die(mysql_error());
6
7 >?

```

Setelah konsi jadi kemudian kita buat form login untuk validasi form, sebelum membuat form silahkan isikan table users dengan username dan nama anda sekalian, buatlah form login pada file index.php pada folder views.

```

1 <form method='post' action='?act=proseslogin'>
2 <div>
3     <table border=''>
4         <tr><td>Username</td><td>:</td><td><input type='text' name='user' required></td></tr>
5         <tr><td>Password</td><td>:</td><td><input type='password' name='pass' required></td></tr>
6         <tr><td colspan='3'><input type='submit' value='Login'></td></tr>
7     </table>
8 </div>
9 </form>

```

Hasil eksekusi dari coding diatas adalah menampilkan form login dimana isinya username dan password, form tersebut belum bisa berfungsi karena kita belum membuat class untuk form lofin tersebut.

Header

Username :	
Password :	
<input type="submit" value="Login"/>	

Footer

Kemudian buatlah class form login untuk menampung fungsi login dengan menghubungkan ke database untuk diketahui apakah benar atau salah.

```
1 <?php
2
3     $user = $_POST['user'];
4     $pass = $_POST['pass'];
5
6     $q = mysql_query("select * from users where email='".$_._POST['user']."");
7     $r = mysql_fetch_array($q);
8
9     if($pass == $r['password']){
10         echo "Boleh Masuk";
11     }else{
12         echo "Tidak Boleh Masuk";
13     }
14
15 ?>
```

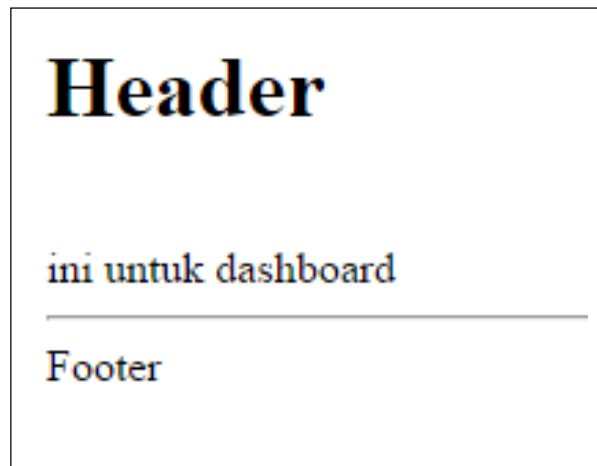
Hasil eksekusi file diatas adalah untuk memvalidasi apakah user yang login sudah sesuai dengan yang ada didalam dalatatabase atau belum, apabila sudah sesuai maka system akan mengenalinya, hasil eksekusi file diatas adalah seperti berikut.



Selanjutnya modifikasilah file tersebut sehingga ketika login benar maka akan menuju ke dashboard user yang mana berisi menu-menu yang digunakan untuk memanajemen system.

```
1 <?php
2
3     $user = $_POST['user'];
4     $pass = $_POST['pass'];
5
6     $q = mysql_query("select * from users where email='".$_._POST['user']."'");
7     $r = mysql_fetch_array($q);
8
9     if($pass == $r['password']){
10         header('location:?act=dashboard');
11     }else{
12         header('location:/trainingphp/templating/?oops=y');
13     }
14
15 ?>
```

Class tersebut akan mengarahkan kita ke dashboard, kemudian di dalam dashboard akan menampilkan menu yang mana dashboard tersebut adalah control panel dari system.



BAB III CREATE READ UPDATE DELETE

MEMBUAT CONTROLLER DAN VIEWS

Dalam sebuah aplikasi pasti ada file yang akan di load sebagai view yang akan digunakan oleh user dalam berkomunikasi dengan system, dan system akan memberikan respon sesuai dengan yang diperintahkan dalam syntax coding, didalam bab ini dibahas bagaimana cara membuat form inputan pemrosesan dan tampilan. Buatlah file formMahasiswa.php sebagai form untuk menginputkan data mahasiswa.

```

1 <form method='post' action='?act=prosesMahasiswa'>
2 <table border='1'>
3     <tr><td>Nama</td><td>:</td><td><input type='text' name='nmMHS'></td></tr>
4     <tr><td>Nim</td><td>:</td><td><input type='text' name='noMHS'></td></tr>
5     <tr><td>Jurusan</td><td>:</td><td><input type='text' name='jsMHS'></td></tr>
6     <tr><td colspan='3' align='right'><input type='submit' value='Simpan'></td></tr>
7 </table>
8 </form>

```

File diatas belum bisa dieksekusi karena controller untuk formMahasiswa.php belum dibuat, selanjutnya buat file controller formMahasiswa.php dan simpanlah di folder controller.

```

1 <?php
2
3     class formMahasiswa extends Loader {
4         public function index() {
5             @include(BASE_PATH."/app/views/formMahasiswa.php");
6         }
7     }
8
9 >

```

Hasil eksekusi dari file diatas adalah menampilkan form untuk inputan mahasiswa dan akan menampilkan nama, nim, jurusan, tampilan dari formMahasiswa.php adalah seperti berikut.

The screenshot shows a web browser window with the address bar containing 'localhost/trainingphp/crud/?act=formMahasiswa'. The page has a header section with the word 'Header' in a large, bold font. Below the header is a form with three input fields labeled 'Nama', 'Nim', and 'Jurusan', each followed by a colon and a text input box. At the bottom right of the form is a button labeled 'Simpan'. Below the form is a footer section with the word 'Footer'.

Langkah selanjutnya adalah membuat data base untuk menampung inputan dari formMahasiswa, bukalah phpmyadmin dan pilih database yang sebelumnya sudah dibuat, kemudian tambahkan tabel mahasiswa dengan syntax seperti berikut.

```

1 create table mahasiswa(
2     id int(5)auto_increment,
3     nama varchar(55),
4     nim int(12),
5     jurusan varchar(55),
6
7     primary key(id)
8 )

```

Hasil eksekusi dari syntax diatas adalah untuk menampung data mahasiswa, id dengan tipe data integer, nama dengan tipe data varchar, nim dengan tipe data integer dan jurusan dengan tipe data varchar setelah itu diberi primary key yaitu id sebagai kunci utamanya.

#	Name	Type	Collation
1	<u>id</u>	int(5)	
2	nama	varchar(55)	latin1_swedish_ci
3	nim	int(12)	
4	jurusan	varchar(55)	latin1_swedish_ci

Selanjutnya adalah membuat class untuk prosesMahasiswa.php dan views dari prosesMahasiswa.php kedua file tersebut adalah untuk mengontrol dan menginputkan data kedalam database, buatlah kedua file tersebut seperti dibawah ini.

```

1 <?php
2
3 class prosesMahasiswa extends Loader {
4     public function index(){
5         @include(BASE_PATH."/app/views/prosesMahasiswa.php");
6     }
7 }
8
9 ?>

```

Class diatas adalah untuk mengontrol views dari prosesMahasiswa.php selanjutnya adalah buat views dari class tersebut seperti dibawah ini.

```

1 <?php
2
3 $nama = $_POST['nmMHS'];
4 $nim = $_POST['noMHS'];
5 $jur = $_POST['jsMHS'];
6
7 $q = mysql_query("insert into mahasiswa(nama, nim, jurusan)
8     values('$nama', '$nim', '$jur')");
9 if($q){
10     header('location:?act=formMahasiswa&ins=y');
11 }
12
13 ?>

```

Kemudian tambahkan file notifikasi bahwa data sudah dimasukkan kedalam database dengan menambahkan beberapa listing coding kedalam class formMahasiswa.php sehingga bisa diidentifikasi apakah data yang diinputkan masuk atau tidak.

```

1 <?php
2     if(!isset($_GET['ins']) == 'y'){
3         echo "";
4     }else{
5         echo "Data Berhasil Diinputkan";
6     }
7 ?>
8 <form method='post' action='?act=prosesMahasiswa'>
9 <table border='1'>
10     <tr><td>Nama</td><td>:</td><td><input type='text' name='nmMHS'></td></tr>
11     <tr><td>Nim</td><td>:</td><td><input type='text' name='noMHS'></td></tr>
12     <tr><td>Jurusan</td><td>:</td><td><input type='text' name='jsMHS'></td></tr>
13     <tr><td colspan='3' align='right'><input type='submit' value='Simpan'></td></tr>
14 </table>
15 </form>

```

Hasil dari eksekusi file diatas dan tampilan yang akan ditampilkan setelah data dimasukkan kedalam database adalah seperti berikut.

	id	nama	nim	jurusan
<input type="checkbox"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	4	Andi	123456	Informatika

Kemudian tampilan dihalaman formMahasiswa.php setelah mendapat informasi bahwa yang diinputkan berhasil masuk kedalam database adalah seperti berikut.

Header

Data Berhasil Diinputkan

Nama	:	<input type="text"/>
Nim	:	<input type="text"/>
Jurusan	:	<input type="text"/>
		<input type="button" value="Simpan"/>

Footer

CLASS LIST MAHASISWA

Langkah selanjutnya adalah membuat list tabel dari daftar mahasiswa yang sudah diinputkan agar bisa diberikan fungsi edit dan delete, buatah class listMahasiswa.php dan views listMahasiswa.php

```

1 <?php
2
3     class listMahasiswa extends Loader {
4         public function index() {
5             @include(BASE_PATH."/app/views/listMahasiswa.php");
6         }
7     }
8
9 ?>

```

Kemudian buatlah views dari data mahasiswa yang diambil dari database yang sudah terdapat beberapa data.

```

<table border=1>
  <tr>
    <thead><th>Nama</th><th>Nim</th><th>Jurusan</th><th colspan=2>Aksi</th></thead>
  </tr>
  <?php
    $q = mysql_query("select * from mahasiswa");
    while($r = mysql_fetch_array($q)){
      echo '
        <tr>
          <td>'.$r['nama'].'</td><td>'.$r['nim'].'</td><td>'.$r['jurusan'].'</td>
          <td><a href="?act=formMahasiswaEdit&id='.$r['id'].'">Edit</a></td>
          <td><a href="?act=formMahasiswaHapus&id='.$r['id'].'">Hapus</td></td>
        </tr>
      ';
    }
  ?>
</table>

```

Hasil eksekusi file diatas adalah seperti berikut.

Header

Nama	Nim	Jurusan	Aksi	
Andi	123456	Informatika	Edit	Hapus
dedy	34567	Informatika	Edit	Hapus

Footer

Kemudian buatlah class untuk edit data dari data yang sudah ditampilkan, yaitu dengan membuat class untuk proses edit dan class views untuk proses edit.

```

1 <?php
2
3 class formMahasiswaEdit extends Loader {
4     public function index(){
5         @include (BASE_PATH."/app/views/formMahasiswaEdit.php");
6     }
7 }
8
9 ?>

```

Kemudian buatlah class untuk views dari class diatas

```

1 <?php
2 if(!isset($_GET['ins']) == 'y'){
3     echo "";
4 }else{
5     echo "Data Berhasil Diinputkan";
6 }
7 $id = $_GET['id'];
8 $q = mysql_query('select * from mahasiswa where id="'.$id.'');
9 $r = mysql_fetch_array($q);
10 <?>ini form mahasiswa edit
11 <form method='post' action='?act=prosesMahasiswa'>
12 <table border='1'>
13 <tr><td>Nama</td><td>:</td></tr>
14 <tr><td><input type='text' name='nmMHS' value='<?php echo $r['nama'];?>'></td></tr>
15 <tr><td>Nim</td><td>:</td></tr>
16 <tr><td><input type='text' name='noMHS' value='<?php echo $r['nim'];?>'></td></tr>
17 <tr><td>Jurusan</td><td>:</td></tr>
18 <tr><td><input type='text' name='jsMHS' value='<?php echo $r['jurusan'];?>'></td></tr>
19 <tr><td colspan='3' align='right'><input type='submit' value='Edit'></td></tr>
20 </table>
21 </form>

```

Langkah berikutnya adalah membuat proses agar proses edit bisa dilakukan dengan sukses.

```

1 <?php
2
3 $nama = $_POST['nmMHS'];
4 $nim = $_POST['noMHS'];
5 $jur = $_POST['jsMHS'];
6 $id = $_POST['id'];
7
8 $q = mysql_query("update mahasiswa set nama='$nama',
9                 nim='$nim', jurusan='$jur' where id='$id'");
10 if($q){
11     header('location:?act=formMahasiswaEdit&id='.$id.'&ins=y');
12 }
13
14 ?>

```

Hasil eksekusi dari class diatas adalah seperti berikut

Header

Data Berhasil Diedit

Nama :	dedy
Nim :	214748
Jurusan :	Informatika
<input type="button" value="Edit"/>	

Footer

Langkah berikutnya adalah membuat aksi delete untuk menghapus data yang sudah diinputkan, buatlah class proses delete dan class viewnya. Sebelumnya modifikasi listMahasiswa menjadi seperti berikut.

```

1 <table border=1>
2 <tr>
3 <thead><th>Nama</th><th>Nim</th><th>Jurusan</th><th colspan=2>Aksi</th></thead>
4 </tr>
5 <?php
6     $q = mysql_query("select * from mahasiswa");
7     while($r = mysql_fetch_array($q)) {
8
9         echo "
10        <tr>
11        <td>".$r['nama']. "</td><td>".$r['nim']. "</td><td>".$r['jurusan']. "</td>
12        <td><a href='?act=formMahasiswaEdit&id=".$r['id']."'>Edit</a></td>
13        <td><a href='?act=formMahasiswaHapus&id=".$r['id']."'
14        onclick='return confirm(\"anda Yakin?\")'>Delete</a></td>
15        </tr>
16        ";
17    }
18    ?>
19 </table>

```

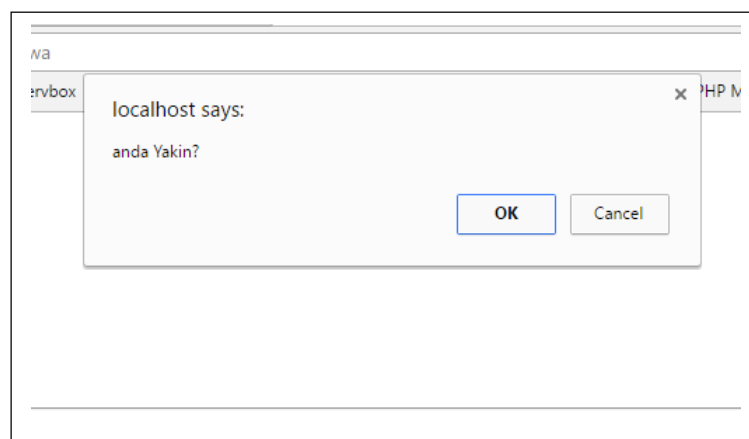
Kemudian buatlah class untuk controllers formMahasiswaHapus.php

```
1 <?php
2
3 class formMahasiswaHapus extends Loader {
4     public function index(){
5         @include(BASE_PATH."/app/views/formMahasiswaHapus.php");
6     }
7 }
8
9 ?>
```

Selanjutnya membuat class proses yang akan mengeksekusi file hapus.

```
1 <?php
2
3 $id = $_GET['id'];
4
5 $q = mysql_query("delete from mahasiswa where id='".$id."'");
6
7 if($q) {
8     header('location:?act=listMahasiswa&ins=y');
9 }
10
11 }
12
13 ?>
```

Kemudian eksekusi dari file tersebut adalah seperti berikut



Hasil eksekusinya adalah seperti berikut

Header			
Data Berhasil Dihapus			
Nama	Nim	Jurusan	Aksi
dedy	214748	Informatika	Edit Delete
Footer			

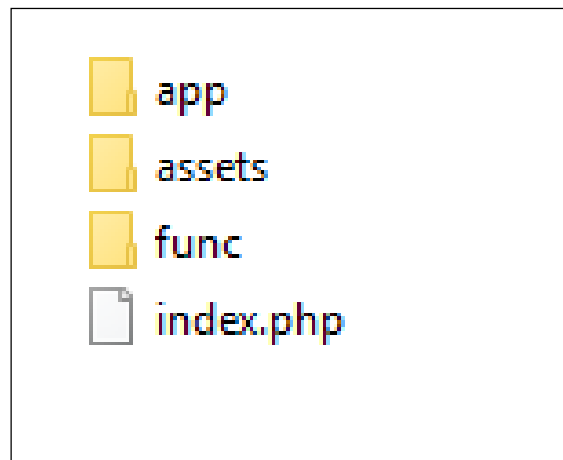
TUGAS

1. Tambahkan field alamat dan citacita pada database
2. Masukkan dan tampilkan data tambahan dari tugas nomor 1

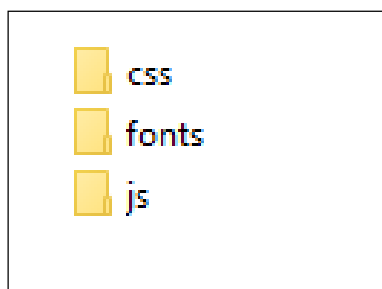
BAB IV BOOTSTRAP TEMPLATING DAN FORMING

BOOTSTRAP INSTALLING

Bootstrap merupakan framework css yang berguna untuk mendesain sebuah aplikasi, aplikasi seaneangih apapun jika tidak disertai dengan tampilan yang menarik pasti akan ditinggalkan oleh penggunanya, oleh karena itu dalam membuat aplikasi perlu yang nama disain interface yang cukup menarik sehingga lebih nyaman untuk dilihat dan bersahabat dipandang oleh user, langkah pertama adalah menambahkan folder assets kedalam aplikasi system kita.



Didalam folder assets ada folder bootstrap yang didalamnya terdapat tiga folder lainnya yang berisi framework bootstrap yaitu folder css, folder js dan folder font.



Langkah selanjutnya adalah mengintegrasikan framework bootstrap yang sudah kita setting sebelumnya agar bisa digunakan kedalam aplikasi system kita, buatlah function.php dan kemudian simpanlah function.php tersebut kedalam folder func.

```
1 <?php
2
3     function cssbootstrap() {
4         $css = "<link rel='stylesheet' href='assets/bootstrap/css/bootstrap.min.css'>";
5         echo $css;
6     }
7     function csscustom($name) {
8         $csc = "<link rel='stylesheet' href='assets/bootstrap/css/" . $name . ".css'>";
9         echo $csc;
10    }
11
12 >
```

Kemudian modifikasi file index yang sudah ada sebelumnya, sehingga menjadi seperti dibawah ini.

Sistem Informasi Akademik

Username:	admin
Password:	•••••
<input type="button" value="Login"/>	

© Footer

```
<?php include('app/config/database.config.php');
    include('func/function.php');
?>
<html><title>Aplikasi PHP</title>
    <head>
        <?php echo cssbootstrap();?>
    </head>
    <nav class='navbar'>
        <h1>Sistem Informasi Akademik</h1>
    </nav><hr>
    <body>
    <div class='container'>
        <?php
            define("BASE_PATH", dirname(__FILE__));
            $loader = (isset($_GET['act'])) ? $_GET['act'] : 'Loader';
            @include(BASE_PATH.'/app/App.php');
            if($loader !== 'Loader'){
                @include(BASE_PATH."/app/controllers/".$loader.".php");
            }
            $those = new $loader();
            $those->index();
        ?>
    </div>
    <footer>
    <div class='container'>
        <hr> &copy; Footer
    </div>
    </footer>
    </body>
</html>
```

FORM FUNCTION

Selanjutnya adalah membuat function form agar tampilan dari form input sesuai dengan tema framework yaitu bootstrap textinput.

```

12 function Form($label, $type, $name, $req){
13
14     $form = "
15         <div class='form-group'>
16             <span>".$label."</span>
17             <input type='".$type."' name='".$name."' class='form-control' ".$req.">
18         </div>
19
20     ";
21
22 }
23

```

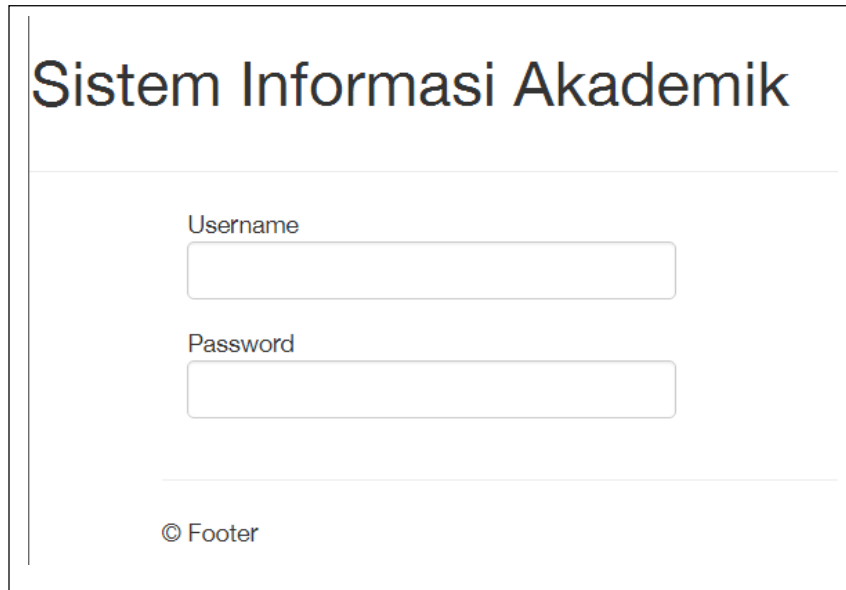
Langkah berikutnya adalah memodifikasi form login sehingga menjadi seperti berikut ini

```

1 <?php
2     if(!isset($_GET['oops']) == 'y'){
3         echo "";
4     }else{
5         echo "Login is Incorect";
6     }
7 ?>
8
9 <form method='post' action='?act=proseslogin'>
10 <div class='col-md-4'>
11     <?php
12         Form('Username', 'user', 'required');
13         Form('Password', 'pass', 'required');
14     ?>
15 </div>
16 </form>

```

Hasil dari modifikasi coding diatas adalah seperti berikut



Sistem Informasi Akademik

Username

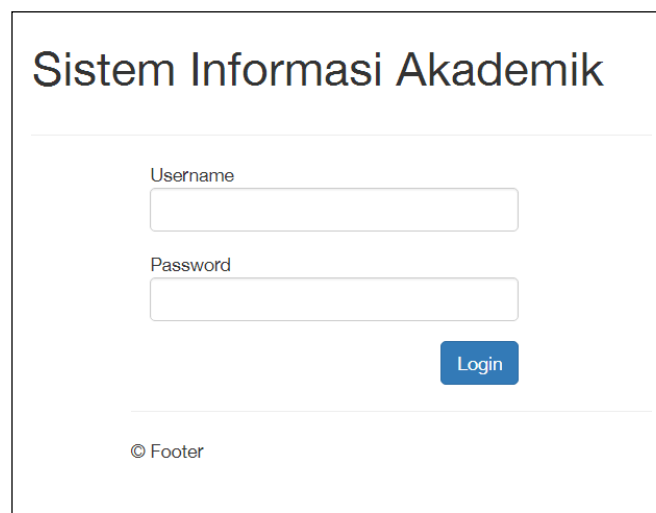
Password

© Footer

Kemudian tambahkan button untuk bisa dieksekusi, buatlah fungsi button ada file function.php

```
function button($label, $type){  
    $button = "  
        <input type='submit' style='float:right' value='".$label."' class='".$type."'>";  
    echo $button;  
}
```

Hasil eksekusi dari file diatas adalah seperti berikut.



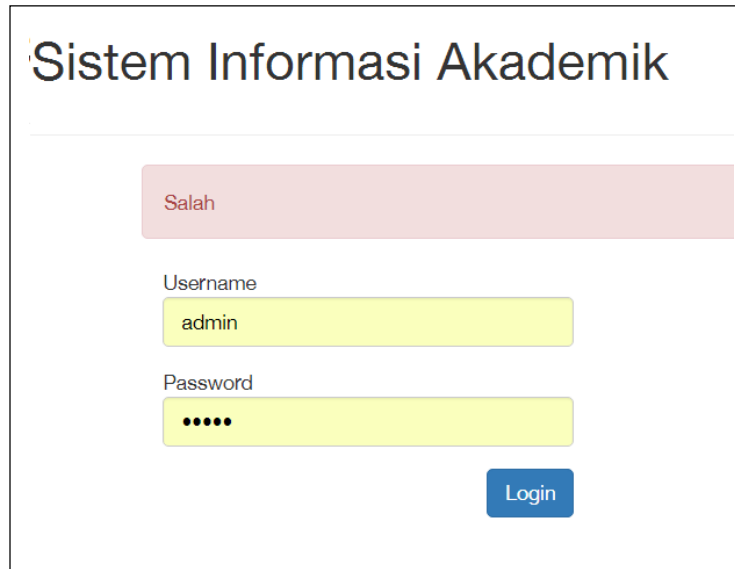
Sistem Informasi Akademik

Username

Password

© Footer

Jika login terdapat kesalahan maka akan memberikan info salah



Sistem Informasi Akademik

Salah

Username
admin

Password
••••

Login

Modifikasi pada file index.php yang terdapat pada folder views

```
1 <?php
2     if(!isset($_GET['oops']) == 'y'){
3         echo "";
4     }else{
5         echo "<div class='alert alert-danger'>Salah</div>";
6     }
7 ?>
8
9 <form method='post' action='?act=proseslogin'>
10 <div class='col-md-4'>
11     <?php
12         Form('Username', 'text', 'user', 'required');
13         Form('Password', 'password', 'pass', 'required');
14         button('Login', 'btn btn-primary');
15     ?>
16 </div>
17 </form>
```

Selanjutnya mengedit file dashboard seperti dibawah ini.

```

1 <a href='?act=formMahasiswa' class='btn btn-info' role='button'>Input Mahasiswa</a>
2 <a href='?act=listMahasiswa' class='btn btn-info' role='button'>Daftar Mahasiswa</a>

```

Sehingga tampilannya menjadi seperti berikut ini.



Langkah selanjutnya adalah mengedit table list mahasiswa sehingga menjadi tabel dengan class yang didapatkan dari bootstrap.

```

1 <?php
2     if(!isset($_GET['ins']) == 'y'){
3         echo "";
4     }else{
5         echo "Data Berhasil Dihapus";
6     }
7 >?>
8 <table class='table'>
9     <tr>
10        <thead><th>Nama</th><th>Nim</th><th>Jurusan</th><th colspan=2>Aksi</th></thead>
11    </tr>
12    <?php
13        $q = mysql_query("select * from mahasiswa");
14        while($r = mysql_fetch_array($q)) {
15
16            echo "
17            <tr>
18                <td>".$r['nama']. "</td><td>".$r['nim']. "</td><td>".$r['jurusan']. "</td>
19                <td><a href='?act=formMahasiswaEdit&id=".$r['id']."'>Edit</a></td>
20                <td><a href='?act=formMahasiswaHapus&id=".$r['id']."'
21                    onclick='return confirm(\"anda yakin?\")'>Delete</a></td>
22            </tr>
23            ";
24        }
25    >?>
26 </table>

```

Tampilan dari coding diatas adalah seperti berikut,

Nama	Nim	Jurusan	Aksi	
dedy	214748	Informatika	Edit	Delete
andi	123456	Informatika	Edit	Delete

© Footer

Langkah selanjutnya adalah membuat form inputan dan edit untuk memberi masukan kedalam database dan melakukan edit pada data yang sudah ada.

Nama	:	dedy
Nim	:	214748
Jurusan	:	Informatika
		<input type="button" value="Edit"/>

© Footer

Form semula adalah seperti pada gambar diatas, kemudian kita rubah agar menjadi form yang sesuai dengan desain yang ada pada framrework bootstrap, langkah yang harus dilakukan adalah membuat class form-control pada setiap form inputan dan harus dalam class form-group, kita buat dulu sebuah function yang mempunyai value untuk kita isikan value yang ada didalam database.

```

function FormVal($label, $type, $name, $val){
    $form = "
        <div class='form-group'>
        <span>". $label. "</span>
        <input type='". $type. "' name='". $name. "' class='form-control' value='". $val. "'>
        </div>
    ";
    echo $form;
}

```

Buatlah function diatas, tambahkan dibawah bari function Form buatlah FormVal untuk fungsi sebuah value edit yang nantinya digunakan sebagai form untuk mengedit setiap value yang ada di dalam database, setelah function tersebut dibuat lalu buatlah form value untuk mengedit data yang ada di dalam database.

```

1  <?php
2  if(!isset($_GET['ins']) == 'y'){
3      echo "";
4  }else{
5      echo "Data Berhasil Diedit";
6  }
7  $id = $_GET['id'];
8  $q = mysql_query('select * from mahasiswa where id="'. $id. "'");
9  $r = mysql_fetch_array($q);
10 ?>
11 <div class='col-md-4'>
12 <form method='post' action='?act=prosesMahasiswaEdit'>
13
14     <?php
15
16     FormVal('Nama', 'text', 'nmMhs', $r['nama']);
17     FormVal('Nama', 'text', 'noMhs', $r['nim']);
18     FormVal('Nama', 'text', 'jsMhs', $r['jurusan']);
19     button('Edit', 'btn btn-primary');
20
21     ?>
22 </form>
23 </div>

```

Selanjutnya modifasilah pada file formMahasiswaEdit.php menjadi seperti coding diatas, yang kemudian akan menghasilkan output sesuai dengan framework bootstrap seperti berikut ini.

Sistem Informasi Akademik

Nama

Nama

Nama

© Footer

Kemudian buatlah function untuk menandakan proses tersebut berhasil dengan menambahkan listing program seperti berikut, dan taruh dibagian paling bawah agar mudah diidentifikasi.

```
function success($tag) {  
    if(!isset($_GET['ins']) == 'y'){  
        echo "";  
    }else{  
        echo "<div class='alert alert-success'>".$tag."</div>";  
    }  
}
```

Hasil tampilan dari kombinasi file coding diatas adalah seperti berikut.

Data Berhasil Diedit

Nama

Nim

Jurusan

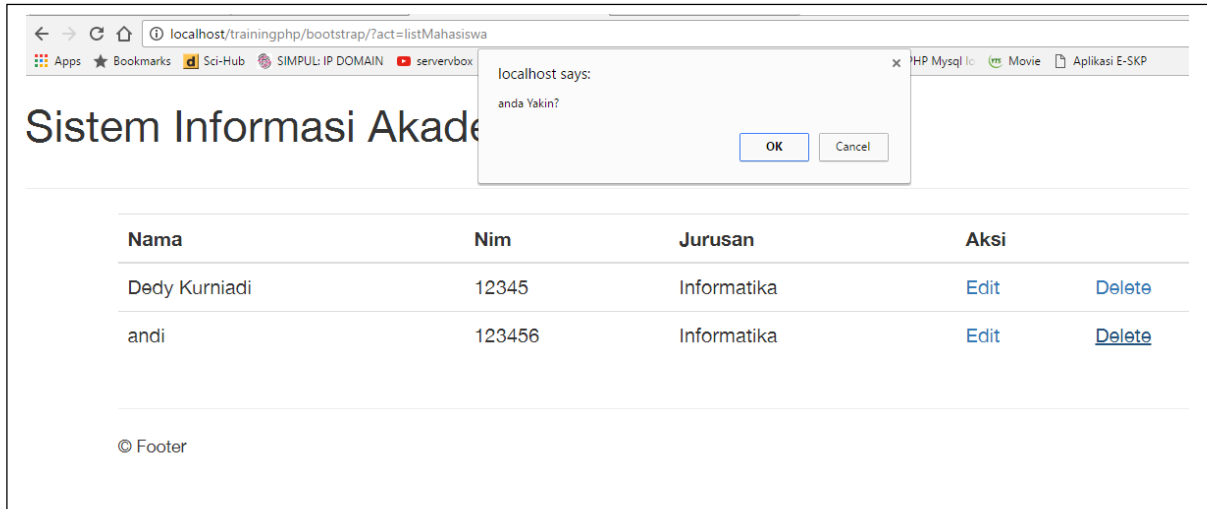
Selanjutnya implementasikan function success tersebut kedalam listMahasiswa agar proses delete bisa diketahui secara langsung didalam form tersebut.

```

1  <?php
2      success('Data Berhasil Dihapus');
3  ?>
4  <table class='table'>
5      <tr>
6          <thead><th>Nama</th><th>Nim</th><th>Jurusan</th><th colspan=2>Aksi</th></thead>
7          </tr>
8          <?php
9              $q = mysql_query("select * from mahasiswa");
10             while($r = mysql_fetch_array($q)) {
11                 echo "
12                 <tr>
13                 <td>".$r['nama']. "</td><td>".$r['nim']. "</td><td>".$r['jurusan']. "</td>
14                 <td><a href='?act=formMahasiswaEdit&id=".$r['id']."'>Edit</a></td>
15                 <td><a href='?act=formMahasiswaHapus&id=".$r['id']."'
16                 | onclick='return confirm(\"anda yakin?\")'>Delete</a></td>
17                 </tr>
18                 ";
19             }
20         }
21     ?>
22 </table>

```

Sehingga tampilan dari coding diatas adalah seperti berikut.



Kemudian setelah dieksekusi akan menjadi seperti berikut.

