

# SISTEM INVENTORY

---

## PENDAHULUAN

### I. Latar Belakang

Komputer adalah salah satu alat bantu utama yang dalam suatu kegiatan manusia, tidak hanya untuk bisnis saja, namun juga digunakan untuk keperluan kantor. Hal ini menunjukkan bahwa sebagaimana komputer telah menjadi bagian dari evolusi manusia terhadap teknologi [1]. Pada kemajuan teknologi yang sudah berkembang pada saat ini, komputer juga sangat membantu untuk meningkatkan kebutuhan suatu sistem pendukung menangani proses pemesanan barang melalui online shop dan juga sangat membantu dalam pemrosesan penerimaan data dengan cepat [2].

Dengan perkembangan internet masyarakat lebih mudah untuk melakukan segala aktivitas dengan mudah. Internet dengan perkembangannya juga menjadi sarana untuk mengatur urusan pergudangan logistic dengan mudah dan praktis. Begitu pula dengan diperlukannya aplikasi rancang bangun sebuah aplikasi online sistem *inventory* berbasis *web* pada PT. Karunia Alam Abadi. Rancang bangun sistem *inventory* ini dimaksudkan untuk mempermudah admin dalam mengelola data. Oleh karena itu rancang bangun aplikasi ini ditambahkan fitur-fitur lainnya agar bisa mempermudah dalam segala keperluan gudang lainnya. Penelitian yang dilakukan pada PT, Karunia Alam Abadi berorientasi pada pengelolaan data *inventory* dan penyajian informasi secara komputerisasi. Dalam artian system yang akan dirancang saat ini sangat membantu dalam menyelesaikan permasalahan yang dihadapi mulai dari belum bisanya pesan secara online dan waktu yang belum bisa efisien.

## **SISTEM INVENTORY**

---

Oleh sebab itu dengan adanya kesulitan seperti yang telah disebutkan di atas, maka akan di rancang nya suatu system rancang bangun aplikasi pemesanan online , diharapkan memudahkan masyarakat memesan produk PT. Karunia Alam Abadi secara online, sehingga kendala jarak bisa teratasi.

### **II. Rumusan Masalah**

Adapun rumusan masalah dalam mengerjakan proyek III ini diantaranya adalah sebagai berikut :

- a. Bagaimana cara masyarakat membeli produk PT. Karunia Alam Abadi secara online ?
- b. Bagaimana proses penerimaan pesanan online oleh admin di PT. Karunia Alam Abadi ?

### **III. Batasan Masalah**

Agar perancangan aplikasi tersebut terkesan sederhana dan mudah dimengerti tetapi tidak mengurangi tujuan penelitian, maka penulis membatasi perancangan aplikasi tersebut sebagai berikut :

1. Tidak ada proses penerimaan pembayaran secara online
2. Tidak menerima pembelian dalam jumlah sedikit

### **IV. Tujuan**

Tujuan dalam pembuatan aplikasi ini adalah supaya masyarakat mudah dalam membeli air mineral dalam jumlah cukup banyak dengan hanya memesan melalui di aplikasi tersebut dan dapat di proses secara mudah dan cepat.

### V. Sistematika Penulisan

Penyajian laporan proyek dibagi menjadi beberapa bab dengan tujuan untuk mempermudah pencarian data atau informasi yang dibutuhkan, serta menunjukkan penyelesaian pekerjaan yang sistematis. Pembagian bab tersebut dibagi seperti berikut:

Bab I Pendahuluan, membahas latar belakang yang berisi ulasan ringkas mengenai keadaan atau kondisi yang ada dan kekurangan dari sistem yang diamati sehingga muncul topik yang di ambil dan juga membahas profil lengkap dari perusahaan yang diteliti. Selain itu juga membahas identifikasi masalah yang berisi berbagai masalah yang sudah di kenali dan akan diberikan solusinya melalui fungsi dari aplikasi yang akan dibuat. Dalam bab ini juga dibahas tujuan untuk apa implementasi dan perancangan sistem *inventory* berbasis *web* tersebut dibuat, dan sistematika penulisan yang menjelaskan isi yang ada di laporan proyek ini.

Bab II Landasan Teori, membahas tentang konsep dasar yang menggunakan referensi metode yang ada dalam jurnal nasional dan internasional, dan juga diambil dari referensi beberapa buku, serta pengertian analisis dan perancangan yang mendukung terbentuknya proyek “***Rancang Bangun Aplikasi Pemesanan Online Terhadap Pelanggan PT. Karunia Alam Abadi***”.

## SISTEM INVENTORY

---

Bab III Analisis dan Perancangan, merupakan proses untuk menentukan bentuk dari kebutuhan aplikasi baik berupa kebutuhan pada saat merancang maupun pada saat pembuatan. Serta penjelasan perancangan aplikasi yang akan dibuat terdiri dari perancangan alur program (*Flow Map*) maupun perancangan *input/output* aplikasi.

Bab IV Kesimpulan dan Saran, berisi kesimpulan yang didapatkan dari analisis dan perancangan sebuah sistem mengenai keterkaitan dengan tujuan pembuatan sistem serta saran-saran yang berkaitan dengan perancangan sistem, penggunaan sistem, atau pengembangan sistem di masa yang akan datang yang diperlukan dalam mendapatkan hasil yang lebih baik.

# SISTEM INVENTORY

---

## LANDASAN TEORI

### 1.1 Sistem

Sistem adalah suatu kelompok dari komponen-komponen yang saling berhubungan dengan satu sama lain dengan tujuan yang sama untuk mencapai tujuan tertentu. Sedangkan informasi adalah berupa data yang diolah menjadi bentuk lebih berguna dan berarti bagi pemakainya.[4] Sehingga dapat disimpulkan bahwa system informasi adalah suatu komponen yang saling berhubungan yang bertujuan mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.



*Gambar Sistem*

Sumber : <http://www.naesys.com/>

## SISTEM INVENTORY

---

Sistem terdiri dari bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud (Budi Sutedjo, 2006:11 ). Sistem terdiri dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu suatu sistem mempunyai karakteristik atau sifat-sifat tertentu. Terdapat dua kelompok pendekatan dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya.

Suatu sistem adalah suatu jaringan kerja untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pengertian prosedur itu sendiri menurut Richard F. Neuschel, prosedur suatu urutan operasi klerikal (tulis menulis), biasanya melibatkan beberapa orang dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.

### **1.1.1 Karakteristik Sistem**

Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu (Sutabri, 2004:12): 153

#### **1. Komponen-komponen (*Components*)**

Suatu sistem terdiri dari sejumlah komponen yang sering disebut dengan subsistem yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai

## SISTEM INVENTORY

---

sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

### 2. Batas sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (scope) sistem itu sendiri.

### 3. Lingkungan luar sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

### 4. Penghubung sistem (*Interface*)

Penghubung merupakan media penghubung antara subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya.

### 5. Masukan sistem (*Input*)

Masukan yaitu energi yang dimasukkan ke dalam sistem, dimana dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). Masukan perawatan adalah energi yang di inputkan supaya sistem tersebut dapat beroperasi, sedang masukan sinyal adalah energi yang diproses untuk didapatkan keluaran.

## SISTEM INVENTORY

---

### 6. Keluaran sistem (*Output*)

Keluaran yaitu hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

### 7. Pengolah sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah *input* menjadi *output*.

### 8. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Apabila suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya.

## 1.1.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari berbagai sudut pandang, diantaranya adalah sebagai berikut (Sutabri, 2004:14):

### 1. Sistem abstrak dan sistem fisik.

Sistem abstrak adalah suatu sistem yang berupa ide-ide pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya teologi yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan.

Sedangkan sistem fisik adalah suatu sistem yang berupa pemikiran atau ide-ide yang nyata atau yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, sistem produksi dan lain sebagainya.

### 2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya: sistem perputaran



## SISTEM INVENTORY

---

bumi. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan human-machine system atau ada yang menyebut dengan man-machine system.

3. Sistem tertentu dan sistem tak tentu

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Contohnya: sistem komputer. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak berpengaruh dengan lingkungan luarnya. Sedangkan sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan.

Menurut Scott (1996) sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), keluaran (*output*) (Hanif Al Fatta 2007:4). Menurut McLeod (2004) sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan (Yakub, 2012:1).

Sedangkan menurut Jogiyanto (1999) terdapat dua kelompok pendekatan sistem didalam mendefinisikan sistem yaitu pendekatan pada prosedur dimana sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu, dan pendekatan pada komponen-komponen atau elemen-elemen, pendekatan pada komponen dianggap lebih mudah dalam mempelajari sistem untuk tujuan dan perancangan system (Yakub, 2012:1).

### 1.2 Sistem Informasi

Menurut Whitten, Bentley, Dittman “ Sistem informasi adalah pengaturan manusia, data, prosesproses, antarmuka yang saling berinteraksi untuk mendukung kebutuhan penyelesaian masalah dan pengambilan keputusan oleh manajemen dan para pengguna.” (1) “Sistem informasi adalah suatu sistem yang menggunakan teknologi informasi untuk mengambil, menyalurkan, menyimpan, mengambil kembali, memanipulasi, atau menampilkan informasi yang digunakan untuk proses bisnis” (2) Sedangkan menurut Wilkinson (Wilkinson, 1993:14), “Sistem Informasi adalah suatu kerangka kerja dengan mana sumberdaya (manusia,komputer) dikoordinasikan untuk mengubah masukan (data) menjadi keluaran (informasi), guna mencapai sasaran perusahaan”.

Dapat disimpulkan bahwa sistem informasi adalah kumpulan elemen yang saling berhubungan satu sama lain dengan memberikan masukan (data) untuk mendapatkan suatu keluaran (informasi) dengan fungsi untuk memproses, menyimpan, mendistribusikan informasi serta mendukung pengambilan keputusan.

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. Sedangkan menurut McLeod dalam informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti bagi penerimanya (Ladjamudin, 2005:5-6).

## SISTEM INVENTORY

---

### 1.2.1 Kualitas Informasi

Informasi yang baik adalah informasi yang berkualitas (Sutabri, 2004:25), informasi yang berkualitas ditentukan oleh beberapa hal, yaitu :

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan, informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*time lines*)

Informasi yang dihasilkan atau dibutuhkan tidak boleh terlambat, karena nantinya tidak mempunyai nilai yang baik, sehingga apabila dijadikan dasar dalam pengambilan keputusan akan berakibat fatal atau kesalahan pengambilan keputusan dan tindakan.

3. Relevan (*relevance*)

Berarti informasi tersebut mempunyai manfaat bagi pemakainya. Informasi merupakan *Output* dari data yang telah diolah sedemikian rupa hingga menjadi suatu bentuk yang berguna dan telah mempunyai arti. *Output* yang tidak memiliki tiga unsur diatas (Akurasi, Tepat Waktu, Relevan) dapat dikatakan sebagai informasi yang berguna, tetapi merupakan sampah.

Sistem informasi bukan merupakan hal yang baru. Yang baru adalah komputerisasinya. Sebelum ada komputer, Teknik penyaluran informasi yang memungkinkan manajer merencanakan serta

## SISTEM INVENTORY

---

mengendalikan operasi yang telah ada. Komputer menambahkan satu atau dua dimensi, seperti kecepatan, ketelitian dan penyediaan data dengan volume yang lebih besar yang memberikan bahan pertimbangan yang lebih banyak untuk mengambil keputusan.

Sistem informasi sistem yang didalamnya mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Sutabri, 2004:35).

Sistem Informasi Sistem adalah kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan dalam usaha mencapai suatu tujuan [5]. Sedangkan dalam buku yang berbeda [3], sistem adalah sekumpulan elemen atau subsistem yang saling bekerjasama atau yang dihubungkan dengan cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan.

Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem tersebut menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan yang dibutuhkan oleh orang untuk menambah pemahamannya terhadap fakta-fakta yang ada [5]. Sedangkan pendapat yang berbeda [3], informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara

## **SISTEM INVENTORY**

---

langsung saat itu juga atau secara tidak langsung pada saat mendatang.

Sistem informasi adalah kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi [5]. Dapat pula diartikan bahwa sistem informasi adalah suatu kumpulan dari komponen-komponen dalam perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi [3].

### **1.3 Integrasi Sistem**

Integrasi sistem didefinisikan sebagai suatu rekayasa dalam proses menyatukan subsistem komponen kedalam satu sistem (agregasi dari sub sistem yang bekerja sama sehingga sistem mampu memberikan fungsi menyeluruh) dan memastikan bahwa subsistem berfungsi bersama sebagai suatu sistem, dan dalam teknologi informasi sebagai proses menghubungkan bersama berbagai sistem yang terkomputasi dan aplikasi perangkat lunak secara fisik atau fungsional untuk bertindak sebagai keseluruhan yang terkoordinasi.

## SISTEM INVENTORY

---



*Gambar Integrasi Sistem*

Sumber : [coins.newbium.com](https://coins.newbium.com)

Sistem integrasi memanfaatkan berbagai Teknik dalam mengintegrasikan dua sistem seperti jaringan computer, integrase aplikasi *enterprise*, manajemen proses bisnis atau pengguna pemrograman.

Didunia modern yang terhubung oleh internet integrase sistem sangat memiliki peran penting. Semakin banyak sistem dirancang untuk terhubung, baik dalam sistem yang akan dibangun maupun ke sistem yang sudah digunakan.

## SISTEM INVENTORY

---

Integrasi Sistem Informasi sering disebut *Enterprise Information System*, yaitu sebuah platform teknologi yang memungkinkan organisasi mengintegrasikan dan mengkoordinasikan proses bisnis. Sedangkan Sandoe (2001) mengatakan bahwa integrasi Sistem Informasi bertujuan menggabungkan Sistem Informasi yang tadinya terpisah dengan tujuan sebuah sumber daya informasi yang lebih komplit dan menyeluruh bagi sebuah organisasi.[5]

### **1.4 Inventory/Persediaan**

Setiap perusahaan, apakah perusahaan itu perusahaan perdagangan ataupun perusahaan pabrik serta perusahaan jasa selalu mengadakan persediaan. Tanpa adanya persediaan, para pengusaha akan dihadapkan pada resiko bahwa perusahaannya pada suatu waktu tidak dapat memenuhi keinginan pelanggan yang memerlukan atau meminta barang/jasa. Persediaan diadakan apabila keuntungan yang diharapkan dari persediaan tersebut hendaknya lebih besar dari pada biaya-biaya yang ditimbulkannya.

Adapun menurut Sofyan Assauri dalam '*Inventory dan Manajemen Inventory*' (3) persediaan dapat didefinisikan sebagai berikut : "Persediaan adalah suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam suatu periode usaha yang normal".

Jadi, persediaan merupakan sejumlah barang yang disediakan untuk memenuhi permintaan dari pelanggan. Dalam perusahaan perdagangan pada dasarnya hanya ada satu golongan *inventory* (persediaan), yang mempunyai sifat perputaran yang sama yaitu yang disebut "*Merchandise Inventory*" (persediaan barang dagangan).

## SISTEM INVENTORY

---

Persediaan ini merupakan persediaan barang yang selalu dalam perputaran, yang selalu dibeli dan dijual, yang tidak mengalami proses lebih lanjut didalam perusahaan tersebut yang mengakibatkan perubahan bentuk dari barang yang bersangkutan. [3]

### ***1.5 Framework***

*Framework* adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi website, kiat harus mengikuti aturan dari *framework*.

Dengan *framework* kita tidak perlu memikirkan kode perintah/ fungsi dasar dari aplikasi website. Seperti bagaimana mengambil data dari database untuk ditampilkan. Kita hanya memikirkan apa kode SQLnya dan ditampilkan kemana? Hal penunjang lainnya seperti koneksi database, validasi form, GUI, dan keamanan telah disediakan oleh *framework* sehingga jumlah baris kode yang kita buat jauh lebih sedikit dibandingkan jika semua kode dari kita.

Jadi keuntungan yang dapat diperoleh dari penggunaan *framework* adalah:

1. Waktu pembuatan aplikasi website kita jauh lebih singkat.
2. Kode aplikasi website menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokonya. Detailnya adalah kode daari *framework* dan ini mungkin difikirkan, terjamin.



## SISTEM INVENTORY

---

3. Website kita menjadi lebih mudah diperbaiki, karena kita tidak perlu focus kesemua komponen kode website terutama kode sistem *framework*.
4. Kita tidak perlu lagi membuat kode penunjang aplikasi website seperti koneksi database, validasi form GUI dan keamanan.
5. Fikiran kita menjadi lebih terfokus ke kode alur permasalahan website. Apa yang ditampilkan dan layanan apa saja yang diberikan dari aplikasi website tersebut.
6. Jika dikerjakan teamwork maka akan lebih terarah karena sistem akan *framework*, mengharuskan adanya keteraturan peletakan kode seperti bagian pengambilan database terpisah dengan bagian pengaturan tampilan untuk pengunjung.

*Codeigniter* dilengkapi dengan berbagai pustaka siap pakai untuk berbagai kebutuhan misalnya saja koneksi database, email, *session*, dan *cokies*, keamanan dan manipulasi gambar dan banyak lagi sehingga mempermudah pekerjaan kita. Dan jika tidak tersedia kita dapat saja menambah pustaka yang ada dengan menciptakan pustaka sendiri yang di *includekan* kedalam *framework* tersebut.

### 1.6 CodeIgniter

*CodeIgniter* adalah kerangka kerja aplikasi web untuk membangun aplikasi web yang sifatnya *open source* yang digunakan untuk membangun aplikasi PHP yang dinamis. *CodeIgniter* memiliki Karakteristik yang sangat fleksibel dan ringan dalam memodifikasi dan mengintegrasikan suatu website. Dalam *CodeIgniter* dapat juga digunakan pola *Model View Controller* (MVC) sehingga struktur kode yang di hasilkan lebih terstruktur dan memiliki standar yang jelas.[8]

*CodeIgniter* adalah salah satu jawabannya bagaimana kita membuat website tidak dari nol. Dengan *codeigniter* kita tidak perlu pusing membuat website PHP karena *framewaork codeigniter* ini sudah menyediakan fasilitas untuk mempercepat pembuatan website. *Codeiginter* ini menyediakan fungsi-fungsi kode yang dapat digunakan kembali dalam pembuatan website dan ini mempercepat efisiensi dalam mengembangkan aplikasi web.

Ada banyak keuntungan penggunaan *codeigniter* antara lain penghematan waktu, *script website* menjadi lebih udah dibaca dan diupdate, mempermudah pembuatan sistem besar dan terstruktur, kode *script* kita menjadi lebih mudah dikelola dan karena adanya pengelompokan kode makan akan mempermudah pembagian kerja dalam tim jika kita membuat *website* secara gotong royong.

### 1.7 PHP (*Hypertext Preprocessor*)

PHP adalah singkatan dari *Hypertext Preprocessor* yang merupakan bahasa pemrograman yang bersifat *open source* yang berada didalam server yang diproses di server. PHP salah satu bahasa pemrograman yang memiliki script yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*).[9]

PHP merupakan bahasa pemrograman pelengkap HTML yang memungkinkan aplikasi web dinamis untuk pengolahan data, pemrosesan data dari *user via form*, membuat buku tamu, toko online, dan lain sebagainya. Dengan mudah PHP dapat melakukan koneksi ke database karen PHP memang dilengkapi fitur yang memungkinkan koneksi ke PHP dilakukan dengan mudah tanpa harus memusingkan (Tim EMS, 2016:55).

PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya akan dijalankan diserver tetapi disertakan pada dokumen HTML. Pembuatan web ini merupakan kombinasi antara php sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman web (Bimo sunarfrihantono, ST 2002:9).



*Gambar Hypertext Preprocessor*

Sumber : <https://www.webhozz.com/>

Menurut Andi Pramono dan M. Syafii (2004 : 2), PHP atau kependekan dari *Hypertext Preprocessor* adalah sebuah bahasa pemrograman berbasis *web* yang mempunyai banyak keunggulan dibandingkan dengan bahasa pemrograman berbasis *web* yang lain. PHP merupakan bahasa pemrograman yang bersumber dari Perl. Sedangkan Perl merupakan pengembangan dari bahasa C.

Oleh karenanya, struktur pemrograman yang ada di PHP sama dengan yang ada di bahasa C. Melihat bahwa PHP merupakan pengembangan dari bahasa C secara tidak langsung, maka PHP mempunyai banyak sekali fitur-fitur yang dapat digunakan. Misalnya, PHP dapat mengakses shell di Linux, mempunyai fungsi yang lengkap berhubungan dengan *networking*.

## SISTEM INVENTORY

---

Salah satu keunggulan PHP disbanding bahasa pemrograman lainnya adalah PHP dapat diperoleh secara gratis, meskipun bukan berarti karena gratis kemampuannya menjadi pas-pasan. PHP sangat powefull. Terbutki dengan banyaknya website yang dibangun menggunakan PHP.

PHP juga terkenal lebih aman daripada bahasa pemrograman website yang lain. PHP juga sudah mendukung OOP (*Object Oriented Programming*) sehingga *maintenance* kode menjadi jauh lebih muda dibandingkan procedural.

PHP bisa berinteraksi dengan database, file dan folder, PHP ini termaksud bahasa *cross-platform*, yang mana PHP ini bisa berjalan di sistem operasi yang berbeda-beda (Windows, Linux, ataupun MAC).[10]

Berdasarkan pernyataan diatas dapat disimpulkan bahwa PHP adalah bahasa adalah sebuah bahasa pemrograman yang sangat fleksibel untuk digunakan dalam membuat sebuah website, dan juga PHP dapat dijalankan di bawah sistem operasi LINUX dan Windows.

### 1.8 MySQL (*My Structure Query Language*)

MySQL adalah sebuah server database *open source* yang terkenal yang digunakan berbagai aplikasi terutama untuk server atau membuat web. MySQL juga adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS). Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja pengoptimasinya dalam melakukan proses perintah-perintah SQL yang dibuat oleh

## SISTEM INVENTORY

---

pengguna maupun program-program aplikasi yang memanfaatkannya.[10]

Berdasarkan dari pengertian diatas kesimpulan dari MySQL adalah sekumpulan sistem database yang banyak digunakan untuk pengembangan suatu aplikasi berbasis web dan bersifat open source.

MySQL merupakan software yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. Open Source menyatakan bahwa *software* ini dilengkapi dengan *source code*. MySQL pada awalnya dibuat oleh perusahaan konsultan bernama TcX yang berlokasi di Swedia. Saat ini pengembangan MySQL berada dibawah naungan perusahaan MySQL AB (Kadir, 2008:2).

Fitur yang terdapat pada MySQL:

1. *Multipatform*

MySQL tersedia pada beberapa platform (Windows, Linux, Unix dan lain-lain).

2. Andal, cepat dan mudah digunakan MySQL tergolong sebagai database server yang andal, dapat menangani database yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses database, dan sekaligus mudah untuk digunakan.

3. Jaminan keamanan akses

MySQL mendukung pengamanan database dengan berbagai kriteria pengaksesan.

#### 4. Dukungan SQL

MySQL mendukung perintah SQL (*Structurd Query Language*). Sebagai diketahui, SQL merupakan standar dalam pengaksesan database relasional.

### 2.8. 1 Perkembangan MySQL

Pada bulan Mei 1996, MySQL versi 1.0 berhasil dirilis secara terbatas hanya untuk 4 orang saja. Namun di bulan Oktober pada tahun yang sama versi 3.11.0 dilepas ke publik. Namun mula-mula kode ini tidak diberikan dibawah lisensi GPL (*General Public License*), melainkan lisensi khusus. Pada tahun 1998-1999, yaitu pada versi-versi akhir 3.22, MySQL menjadi semakin populer dan dilirik orang karena kestabilan dan kecepatan yang meningkat. Jika pada versi 3.22, MySQL mulai diadopsi oleh banyak orang. Berbeda halnya dengan versi 3.23 dan 4.0 yang telah terjadi banyak peningkatan dari sisi teknologi (Sukarno, 2006:5-7).

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas, MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

## SISTEM INVENTORY

---

3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *'Performance tuning'*, MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lainlain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.



## SISTEM INVENTORY

---

11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle

### 1.9 UML (*Unified Modeling Language*)

*The Unified Modeling Language* (UML) adalah standar de facto untuk analisis perangkat lunak berorientasi objek dan pemodelan desain.[11] UML membantu memodelkan berbagai aspek sistem melalui berbagai diagram yang di dukungnya. Setiap aspek dari suatu sistem disajikan dengan menggunakan jenis diagram UML tertentu dan satu set diagrama disebut sebagai model.[12]

UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Karena ini merupakan sebuah bahasa, UML memiliki sejumlah aturan untuk menggabungkan atau mengkombinasikan elemen-elemen tersebut.

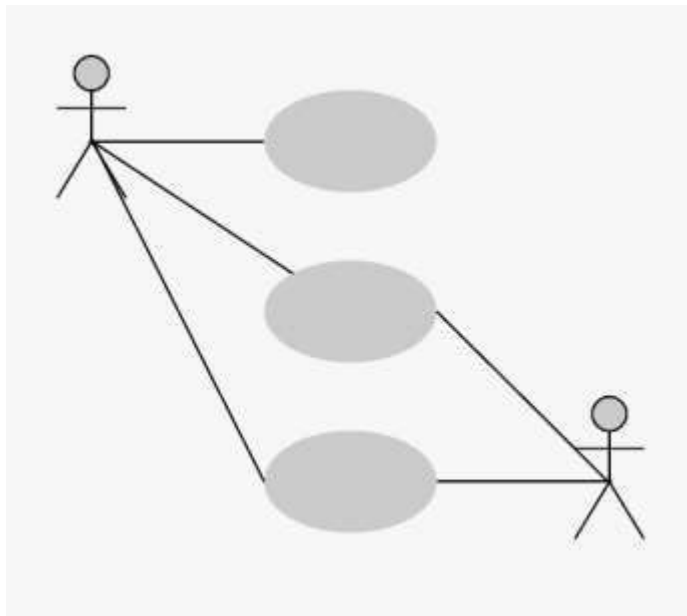
#### 1. *Use case diagram*

*Use case diagram* menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem. *Use case diagram* menjadi dokumen kesepakatan antara *customer*, *user* dan *developer*. *User*

## SISTEM INVENTORY

---

menggunakan dokumen *use case* diagram ini untuk memahami sistem dan mengevaluasi bahwa benar yang dilakukan sistem adalah untuk memecahkan masalah yang user ajukan. *Use case* diagram memberikan gambaran statis dari sistem yang sedang dibangun dan merupakan artifak dari proses analisis (Hermawan, 2004:23).



*Gambar Use Case Diagram*

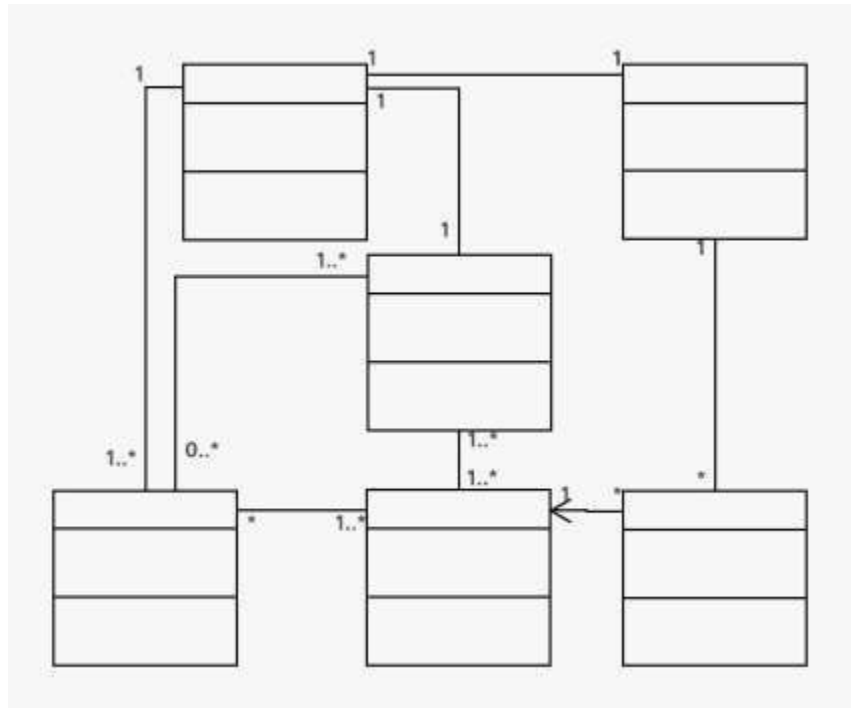
### 2. *Class Diagram*

*Class diagram* merupakan diagram yang selalu ada di pemodelan sistem berorientasi obyek. *Class diagram* menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai tujuan. *Class diagram* digunakan untuk

## SISTEM INVENTORY

---

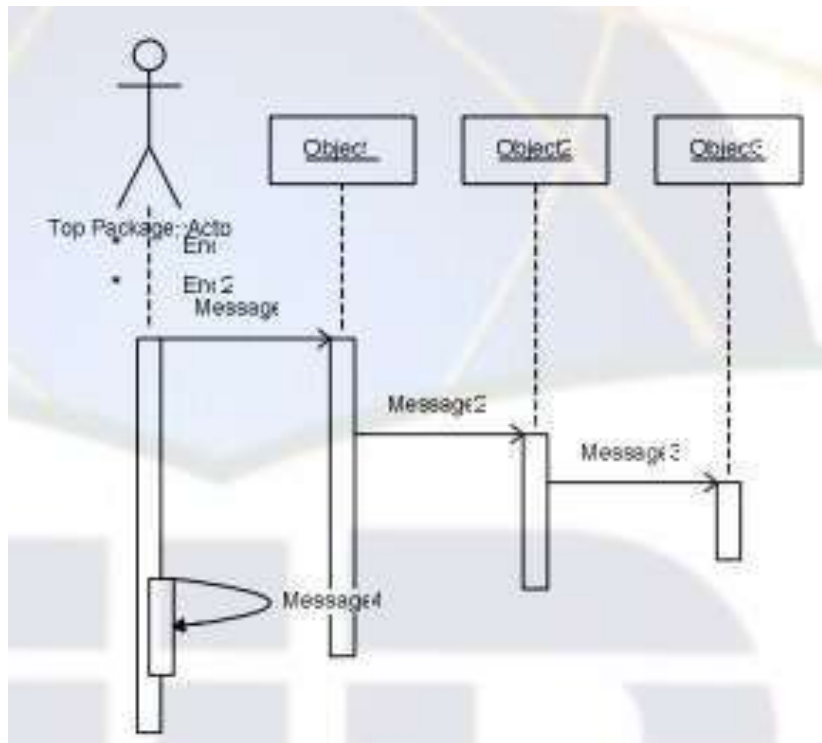
menggambarkan disain statis dari sistem yang sedang dibangun (Hermawan, 2004:27).



*Gambar Class Diagram*

### 3. *Sequence diagram*

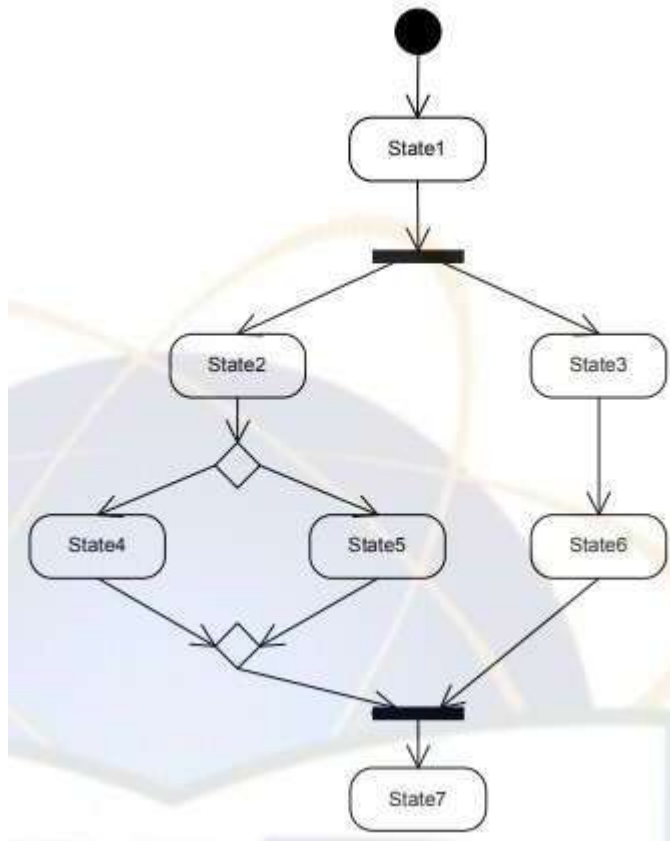
*Sequence diagram* menjelaskan secara detail uruta proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. *Sequence diagram* menjelaskan aspek dinamis dari sistem yang sedang dibangun (Hermawan, 2004:24).



*Gambar Sequence Diagram*

#### 4. Activity Diagram

*Activity diagram* adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* mendukung perilaku paralel sedangkan *flowchart* tidak bisa (Munawar, 2005:109).



*Gambar Activity Diagram*

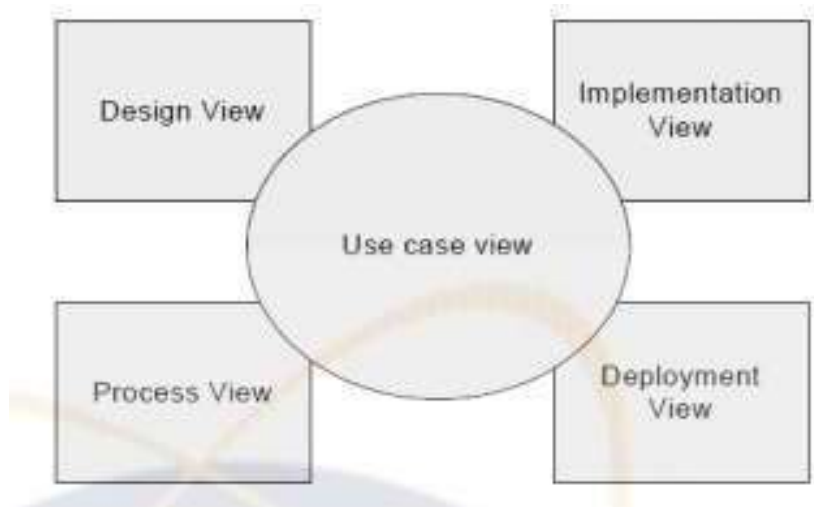
### 5. UML

UML dibangun atas model 4+1 view. Model ini didasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam 5 view dimana salah satu diantaranya *use case view*. *Use case view* ini memegang peran khusus diantaranya mengintegrasikan content ke *view* yang lain. Kelima view tersebut tidak berhubungan dengan diagram yang dideskripsikan di UML. Setiap *view* berhubungan dengan perspektif tertentu dimana sistem akan diuji. *View* yang

## SISTEM INVENTORY

---

berbeda akan menekankan pada aspek yang berbeda dari sistem yang mewakili ketertarikan sekelompok *stakeholder* tertentu.



*Gambar UML Diagram*

Penjelasan lengkap gambar UML Diagram tentang sistem bisa dibentuk dengan menggabungkan informasi-informasi yang ada pada *view* pada kelima *view* tersebut.

1. *Use case view* mendefinisikan perilaku eksternal sistem.
2. *Design view* mendefinisikan struktur logika yang mendukung fungsi-fungsi yang dibutuhkan di *use case*. Informasi yang terkandung di *view* ini menjadi perhatian para programmer karena menjelaskan secara detail bagaimana fungsionalitas sistem akan diimplementasikan.

## SISTEM INVENTORY

---

3. *Implementation view* menjelaskan komponen-komponen fisik dari sistem yang akan dibangun. Informasi yang ada di view ini relevan dengan aktifitas-aktifitas seperti manajemen konfigurasi dan integrasi sistem.
4. *Process sistem* berhubungan dengan hal-hal yang berkaitan dengan concurrency di dalam sistem.
5. *Deployment view* menjelaskan komponen-komponen fisik didistribusikan ke lingkungan fisik seperti jaringan komputer dimana sistem akan dijalankan.

### 1.10 Database



*Gambar Database*

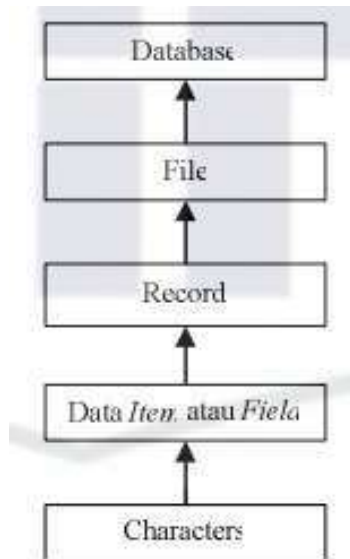
Sumber : robicomp.com

*Database* didefinisikan sebagai kumpulan data yang saling berkaitan secara teknis, database juga sebagai tempat media

## SISTEM INVENTORY

---

penyimpanan data kita dalam membuat sebuah program yang berisikan tabel, field, record, yang diselimuti dengan DBMS (*Database Management System*).[13]



Basis data (database) ini kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya (Jogiyanto, 2005:217). Jadi basis data merupakan suatu komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di basis data.

Sampai dengan membentuk suatu data base, data mempunyai jenjang yang dapat dilihat pada gambar (Jogiyanto, 1999:714-715):

a. *Characters*

*Characters* merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf atau pun karakter-karakter



## SISTEM INVENTORY

---

khusus (*special characters*) yang membentuk suatu item data atau field.

### b. *Field*

*Field* menggambarkan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari field membentuk suatu record

#### 1) Nama dari *field* (*field name*)

*Field* harus diberi nama untuk membedakan *field* yang satu dengan *field* yang lain.

#### 2) Representasi dari *field* (*field representation*)

Representasi dari *field* menunjukkan tipe dari *field* (*field type*) dapat berupa tipe *numeric*, karakter atau huruf, tanggal, dan memo, serta lebar dari field (*field width*) menunjukkan ruang maksimum dari field yang dapat diisi dengan karakter-karakter data.

#### 3) Nilai dari *field* (*field value*)

Nilai dari *field* menunjukkan isi dari *field* untuk masingmasing *record*.

### c. *Record*

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

## SISTEM INVENTORY

---

### *d. File*

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

### *e. Database*

Database merupakan kumpulan dari file membentuk suatu database. Tujuan basis data yang efektif termuat di bawah ini:

- Memastikan bahwa data dapat dipakai diantara pemakai untuk berbagai aplikasi.
- Memelihara data baik keakuratan maupun konsistensinya.
- Memastikan bahwa semua data yang diperlukan untuk aplikasi sekarang dan yang akan datang akan disediakan secara cepat.
- Membolehkan basis data untuk berkembang dan kebutuhan pemakai untuk berkembang.
- Membolehkan pemakai untuk mengembangkan pandangan personalnya tentang data tanpa memperhatikan cara data disimpan secara fisik.

*DataBase Management System* (DBMS atau DMS) adalah paket perangkat lunak yang kompleks digunakan untuk memanipulasi database (Jogiyanto, 1999:731). Lebih detail lagi dijelaskan oleh Hariyanto bahwa DBMS adalah perangkat lunak untuk mendefinisikan, menciptakan, mengelola dan mengendalikan pengaksesan basisdata. Semua operasi *input* dan *output* yang berhubungan dengan database harus menggunakan DBMS. Bila pemakai akan mengakses database, DBMS menyediakan penghubung

## SISTEM INVENTORY

---

(*interface*) antara pemakai dengan database (Jogiyanto, 1999:734).

Hubungan pemakai dengan database dapat dilakukan dengan cara:

- a. Secara interaktif menggunakan bahasa pertanyaan (*query language*).
- b. Dengan menggunakan program aplikasi.

Kegunaan Basis Data Penyusunan satu basis data digunakan untuk mengatasi masalah-masalah pada penyusunan data [2], yaitu:

- 1) Redudansi dan inkonsistensi data Jika file-file dan program aplikasi diciptakan oleh programmer yang berbeda pada waktu yang berselang cukup panjang, maka ada beberapa bagian data mengalami penggandaan pada file-file yang berbeda. Penyimpanan data yang berulang-ulang di beberapa file juga dapat mengakibatkan inkonsistensi (tidak konsisten).
- 2) Kesulitan Pengaksesan Data Suatu saat dibutuhkan untuk mencetak data siapa saja, padahal belum tersedia program yang telah tertulis untuk mengeluarkan data tersebut maka kesulitan tersebut timbul, dan penyelesaiannya untuk itu adalah kearah Sistem Manajemen Basis Data yang mengambil data secara langsung dengan bahasa yang familian dan mudah digunakan.
- 3) Isolasi data untuk standarisasi Jika data tersebar dalam beberapa file dalam bentuk format yang tidak sama, maka ini menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data, maka haruslah data dalam

## SISTEM INVENTORY

---

satu basis data dibuat satu format sehingga mudah membuat program aplikasinya.

- 4) Masalah keamanan atau Security Setiap pemakai sistem basis data tidak semuanya diperbolehkan untuk mengakses semua data. Misalnya data mengenai gaji pegawai hanya boleh dibuka oleh bagian keuangan dan personalia. Keamanan ini dapat diatur lewat program yang dibuat oleh pemrogram atau fasilitas keamanan dari operating sistem.
- 5) Masalah Integrasi (Kesatuan) Basis Data berisi file yang saling berkaitan, masalah utama adalah bagaimana kaitan antara file tersebut terjadi. Meskipun diketahui bahwa file A berkaitan dengan file B, namun secara teknis maka ada file kunci yang mengaitkan kedua file tersebut.
- 6) Masalah data independence (kebebasan data) Aplikasi yang dibuat dengan bahasa yang diciptakan dari Sistem Manajemen Basis Data, apapun yang terjadi pada struktur file, setiap kali hendak melihat data cukuplah dengan utility USE, hendak menambah data cukup dengan APPEND, ini berarti perintah-perintah dalam paket Sistem Manajemen Basis Data bebas terhadap basis data. Perubahan apapun dalam basis data, semua perintah akan mengalami kestabilan tanpa perlu ada yang diubah.

Berdasarkan dari uraian di atas dapat disimpulkan bahwa database adalah sekumpulan data yang beris tabel, field, dan record yang disimpan secara sistematis.

### 1.11 API (*Application Programing Interface*)

*Application Programming Interface* atau API adalah sebuah dokumentasi yang terdiri dari *interface*, kelas, fungsi, struktur dan sebagainya agar dapat membangun sebuah perangkat lunak. Dan API bisa dikatakan sebagai suatu kode pemrograman penghubung antara aplikasi atau web yang telah kita buat dengan fungsi yang dikerjakan [10]. Google+ api merupakan pemrograman yang menghubungkan ke Google+ [11].

Artinya dengan API, web PenA (*Penilaian Absensi*) akan terintegrasi dengan Rinfo+. Yang akan menghubungkan foto profile pada Rinfo+ dalam sistem PenA (*Penilaian Absensi*) . Dalam hal ini Google Api sebagai kode pemrograman yang disederhanakan yang dapat ditambahkan oleh sistem PenA (*Penilaian Absensi*) untuk mengakses dan terintegrasi dengan fitur Rinfo+.

*Application Programming Interface* (API) atau Antarmuka Pemrograman Aplikasi adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. [14]



## SISTEM INVENTORY

---

lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan programmer menggunakan sistem function. Proses ini dikelola melalui sistem operasi. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. [15]

*Application Programming Interface* (API) ini yang memfasilitasi pertukaran informasi atau data antara dua atau lebih aplikasi perangkat lunak. API adalah antarmuka virtual antara dua fungsi perangkat lunak yang saling bekerja sama, seperti antara sebuah word processor dan sebuah spreadsheet [7]. Sebuah API mendefinisikan bagaimana cara programmer memanfaatkan suatu fitur tertentu dari sebuah komputer. API tersedia untuk sistem windowing, sistem file, sistem database, serta sistem jaringan.

Sistem informasi berbasis API memungkinkan sebuah back-end dimanfaatkan dengan cara yang lebih luas karena logic pada sistem dengan logic pada antarmukanya terpisah. Sistem informasi berbasis API juga akan mempermudah sebuah sistem untuk berkolaborasi dengan sistem lain. Hal ini jadi sangat berguna jika pada waktu mendatang proses bisnis yang difasilitasi menuntut interaksi dengan sistem lain, atau minimal akan sangat berguna pada pengembangan modul-modul baru pada sistem informasi terkait.

Secara umum API merupakan ekspresi terfokus keseluruhan fungsional dalam suatu modul software yang dapat diakses oleh orang yang membutuhkan dengan cara yang telah ditentukan layanan.

## SISTEM INVENTORY

---

Representasi terfokus dari fungsi yang dideklarasikan dalam API dimaksudkan untuk menyediakan rangkaian layanan yang spesifik untuk target tertentu. Jika dalam satu modul memiliki API ganda, hal ini sudah menjadi hal yang umum karena setiap API dimaksudkan untuk penggunaan yang spesifik dari modul terkait (Rama dan Avinash, 2015).

Web API adalah antar muka program dari sistem yang dapat diakses melalui *method* dan *header* pada protokol HTTP yang standar. Web API dapat diakses dari berbagai macam HTTP client seperti Browser dan perangkat mobile. Web API juga memiliki keuntungan karena menggunakan infrastruktur yang juga digunakan oleh web terutama untuk penggunaan *caching* dan *concurrency* (Miller dkk, 2014).

*Windows Application Programming Interface* (API) terdapat dalam file-file DLL yang menyusun Sistem Operasi Windows[14]. Karena terdapat dalam file *library* (DLL) maka fungsi-fungsi dan konstanta-konstanta tersebut dapat digunakan oleh aplikasi atau library lainnya. Windows API biasanya disusun dalam bahasa pemrograman C/C++. Bahasa C dipilih karena kecepatan dan kedekatannya dengan hardware.

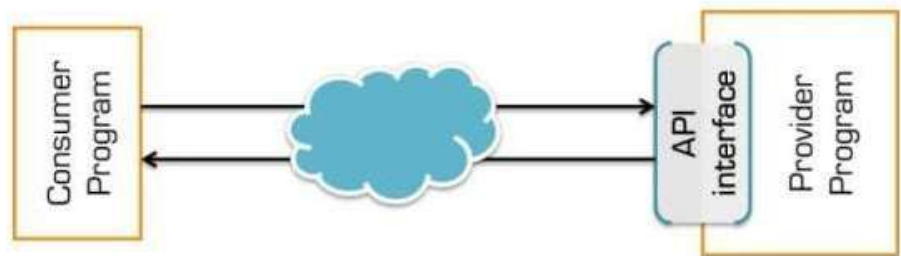
Windows menyediakan banyak sekali API yang memiliki fungsi berbedabeda. Berikut ini adalah beberapa API yang disediakan oleh Sistem Operasi Windows :



## SISTEM INVENTORY

---

1. Win32 API, API yang disediakan Windows untuk melakukan pemrograman Windows secara umum. Fungsi-fungsi yang terdapat dalam Win32 API ini sangat lengkap mulai dari menu, button, window, eksekusi program dan lain-lain.
2. *Telephony* API (TAPI), API yang digunakan untuk melakukan fungsifungsi telephony dengan menggunakan Sistem Operasi Windows.
3. *Messaging* API (MAPI), API yang digunakan untuk melakukan pengiriman pesan (termasuk pengiriman email) dengan menggunakan Sistem Operasi Windows.
4. *Speech* API (SAPI), API yang disediakan Windows untuk menambahkan fungsi speech ke dalam aplikasi yang berjalan di atas Sistem Operasi Windows.
5. *Cryptography* API, API untuk melakukan fungsi-fungsi enkripsi dan dekripsi. Banyak sekali metode enkripsi/dekripsi yang disediakan oleh cryptography API ini.
6. dan masih banyak API lainnya (termasuk yang tidak didokumentasikan oleh Microsoft). Seperti telah disinggung sebelumnya bahwa API sebenarnya hanyalah sebuah fungsi. Seperti halnya fungsi-fungsi lainnya, fungsi tersebut dapat memiliki parameter-parameter dan dapat mengembalikan nilai. Bedanya hanyalah fungsi-fungsi tersebut ditulis dalam bahasa C/C++ sehingga bagi yang belum familiar dengan bahasa C/C++ biasanya akan mengalami kesulitan.



*Gambar 1 API*

API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi lain. API diimplementasikan dengan menulis fungsi panggilan atau sintaks dalam program, yang menyediakan sarana yang diperlukan untuk meminta layanan program.

Pada dasarnya, program API mendefinisikan cara yang tepat bagi pengembang untuk meminta layanan dari program itu. Sebagai contoh, Amazon.com merilis API sehingga pengembang situs web dapat lebih mudah mengakses informasi produk Amazon, menggunakan Amazon API.

Sebuah situs web pihak ke-tiga dapat memposting link langsung ke produk Amazon dengan harga yang terupdate dan pilihan untuk “beli sekarang”. (Prasetyo, 2014) Dalam API terdapat fungsi-fungsi/perintah-perintah untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh pengembang. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil *system calls*

## SISTEM INVENTORY

---

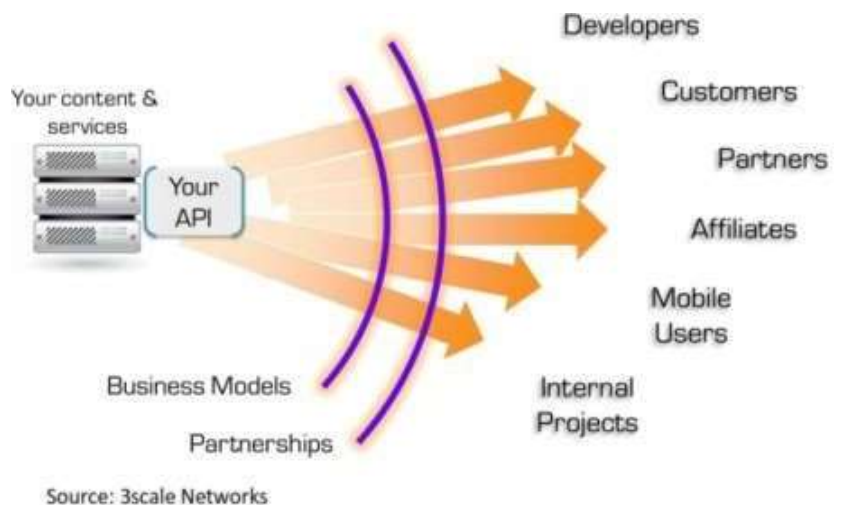
sesuai dengan sistem operasinya. Tidak tertutup kemungkinan nama dari systemcalls sama dengan nama di API.

Keuntungan memprogram dengan menggunakan API adalah:

1. Portabilitas

Pengembang yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah ter-install API tersebut. Sedangkan system call berbeda antar sistem operasi, dengan catatan dalam implementasinya mungkin saja berbeda (Gambar 1).

2. Lebih mudah dimengerti API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa system call. Hal ini sangat penting dalam hal editing dan pengembangan (Gambar 2).



*Gambar 2 API*

## SISTEM INVENTORY

---

*System call interface* ini berfungsi sebagai penghubung antara API dan system call yang dimengerti oleh sistem operasi. System call interface ini akan menerjemahkan perintah dalam API dan kemudian akan memanggil system calls yang diperlukan. Untuk membuka suatu file tersebut user menggunakan program yang telah dibuat dengan menggunakan bantuan API, maka perintah dari user tersebut diterjemahkan dulu oleh program menjadi perintah `open()`. Perintah `open()` ini merupakan perintah dari API dan bukan perintah yang langsung dimengerti oleh kernel sistem operasi.

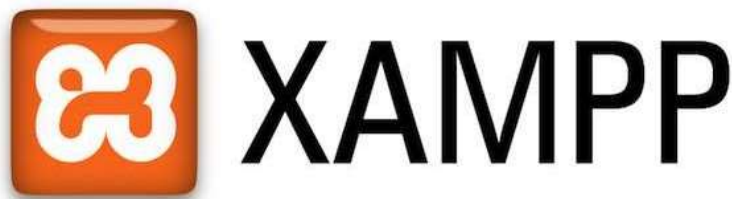
Oleh karena itu, agar keinginan user dapat dimengerti oleh sistem operasi, maka perintah `open()` tadi diterjemahkan ke dalam bentuk *system call* oleh *system call interface*. Implementasi perintah `open()` tadi bisa bermacam-macam tergantung dari sistem operasi yang kita gunakan (Setiawan, 2014).

### **1.12    *Web Application***

Sekarang ini *web application* dikenal sebagai aplikasi yang diakses *melalui web Browser* dan melalui jaringan seperti Internet atau intranet. Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstal perangkat lunak pada kemungkinan ribuan komputer klien merupakan keunggulan teknologi ini, selain juga untuk cross-platform compatibility. Termasuk aplikasi web common webmail, penjualan ritel online, online pelelangan, wiki dan banyak fungsi lainnya.

Pada jurnal yang ditulis oleh Xu, dkk, (2005), menitikberatkan pada efektifitas dan efisiensi sebuah testing terhadap aplikasi yang berbasis web application dengan membandingkan dua metode yaitu Semantic Label dan XML description technique. Lei Xu dan timnya mengembangkannya dengan melengkapi mekanisme feedback control pada pembangunan aplikasi agar lebih menyempurnakan kualitas sistem. Edinburgh (2005) membahas sebuah pendekatan pengujian pada web application. Dalam metode pendekatannya analisa aliran data akan dianggap sebagai *Function Level Testing*, *Function Cluster Level Testing*, *Object Level Testing* dan *Web Application Level Testing*, dari level terendah hingga level tertinggi.

### 1.13 XAMPP



*Gambar XAMPP*

Sumber : [commons.wikimedia.org](https://commons.wikimedia.org)

## SISTEM INVENTORY

---

XAMPP adalah sebuah *software web server apache* yang didalamnya sudah tersedia database server MySQL dan dapat mendukung pemrograman PHP. XAMPP merupakan software yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia Apache Web Server, MySQL Database Server, PHP Support (PHP 4 dan PHP 5) dan beberapa module lainnya.

Perbedaan versi untuk Windows operating system sudah dalam bentuk instalasi grafis dan yang Linux dalam bentuk file terkompresi tar.gz. Kelebihan lain yang berbeda dari versi untuk Windows adalah memiliki fitur untuk mengaktifkan sebuah server secara grafis, sedangkan Linux masih berupa perintah-perintah di dalam console.

### 1.14 CSC (*Cascading Style Sheets*)



*Gambar Logo CSS*  
Sumber : petanikode.com

## SISTEM INVENTORY

---

CSS (*Cascading Style Sheets*) Pada versi HTML yang terdahulu, web Browser mengontrol tampilan (rendering) dari setiap halaman web. Jika menggunakan elemen H1 pada (*large heading*) pada web dokumen, Browser akan merender elemen tersebut.

Dengan adanya CSS, *programmer* dapat mengontrol bagaimana *Browser me-render* halaman web. Mengaplikasikan CSS pada halaman web dapat memberikan tampilan yang lebih menarik dan spesifik sesuai dengan tema pada sebuah web site.

Teknologi CSS memberikan fasilitas untuk menentukan style (misal;*spacing, margins*) dari elemen halaman web terpisah dari struktur dokumen web (*section headers, body text, links* ). Pemisahan tersebut memberikan peningkatan yang lebih besar dalam pengaturan *web pages*, dan membuat perubahan – perubahan *style* dalam dokumen dapat dilakukan lebih cepat dan lebih mudah.

### 1.15 *Local Host*

*Localhost* adalah sebuah akses local yang didapat dari sebuah aplikasi untuk dapat mengakses local server yang dimaksudkan untuk mengetahui bagaimana kekurangan dan kelebihan aplikasi yang akan digunakan sebelum dilakukan hosting.

*Hosting* adalah sebuah tempat dimana kita bisa menyimpan data-data website kita sehingga dapat diakses lewat

internet. Alasan peneliti menginstall di *localhost* sebelum *hosting* dikarenakan selain mengetahui kekurangan dan kelebihan aplikasi peneliti juga dapat menghemat penggunaan internet jika ada kesalahan yang dialami sebelum di *upload* pada *hosting* yang akan digunakan.

Localhost adalah server lokal yang ada pada komputer. Localhost adalah hostname dari komputer itu sendiri. Localhost dapat diakses ketika ada web server yang berjalan.

*Local Host* Menurut Kasiman Peranginangin (2006), *Localhost* adalah sebuah *host server* yang ada pada komputer Anda sendiri. *Localhost* biasanya digunakan untuk kebutuhan intranet atau untuk mempraktikkan pembuatan *web* sebelum di-*upload* ke *web hosting*. Salah satu penyedia server local adalah Xampp.

Xampp sendiri merupakan kepanjangan dari X, Apache, MySQL, PHP, dan Perl. X di sini maksudnya Xampp, yaitu cross (X) platform yang dapat digunakan pada sistem operasi Windows, Linux, Mac, dan Solaris. Sedangkan Apache ialah server yang digunakan. MySQL berarti basis data yang digunakan untuk localhost. PHP dan Perl merupakan script/bahasa pemrogramannya.

### **1.16    *Flow Map***

*Flowmap* adalah campuran peta dan *flow chart*, yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan,



## SISTEM INVENTORY

---

atau jumlah. *Flowmap* menolong analisis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.[5]

*Flowmap* merupakan diagram alir dokumen yang digunakan untuk menggambarkan hubungan antara *entity* yang terlibat berupa aliran-aliran dokumen yang ada. Untuk menjalankan prosedur sistem, digunakan *flowmap* yang terbentuk dari analisis prosedur. Bagan alir dokumen tersebut juga disebut bagan alir formulir yang merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan – tembusannya.

### 1.17 HTML



Gambar Script HTML

Sumber : animamedia.com

## SISTEM INVENTORY

---

HTTP merupakan protokol yang digunakan untuk mentransfer data antara *web server* ke *web Browser*. Protokol ini mentransfer dokumen-dokumen web yang ditulis atau berformat HTML. Dikatakan markup karena HTML berfungsi untuk "mepercantik" file text biasa untuk ditampilkan pada program web Browser. Hal ini dilakukan dengan menambahkan tag-tag (perintah khusus) pada file teks biasa tersebut.


Tag HTML biasanya berupa tag-tag yang berpasangan dan ditandai dengan simbol `<` dan `>`. Pasangan dari sebuah tag ditandai dengan tanda `'/'`. misalnya awalnya perintah tagnya `<contoh>` maka diakhiri dengan `</contoh>`.

## TATA CARA INSTALASI DAN PENGGUNAAN *TOOLS* YANG DIGUNAKAN

### 2.1 Xampp

#### 2.1.1 Cara Install Xampp

1. *Download* aplikasi Xampp terbaru , link : <https://www.apachefriends.org/download.html>. Pilih salah satu sesuai dengan yang anda butuhkan :



The screenshot shows the XAMPP for Windows download page. It features a table with three rows representing different versions: 7.2.26 / PHP 7.2.26, 7.3.13 / PHP 7.3.13, and 7.4.1 / PHP 7.4.1. Each row includes a 'What's Included?' link, 'md5' and 'sha1' checksum links, a 'Download (64 bit)' button (highlighted with a blue box), and a 'Size' column. Below the table, there are links for 'Requirements', 'Add-ons', and 'More Downloads »'. A note at the bottom states: 'Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.'

Version	Checksum	Size
7.2.26 / PHP 7.2.26	What's Included? md5 sha1	145 Mb
7.3.13 / PHP 7.3.13	What's Included? md5 sha1	146 Mb
7.4.1 / PHP 7.4.1	What's Included? md5 sha1	148 Mb

Requirements Add-ons More Downloads »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

2. Klik dua kali pada file xampp yang baru saja anda *download*, jika muncul pesan *error* seperti gambar berikut abaikan saja dan pilih oke untuk melanjutkan instalasi

## SISTEM INVENTORY

---



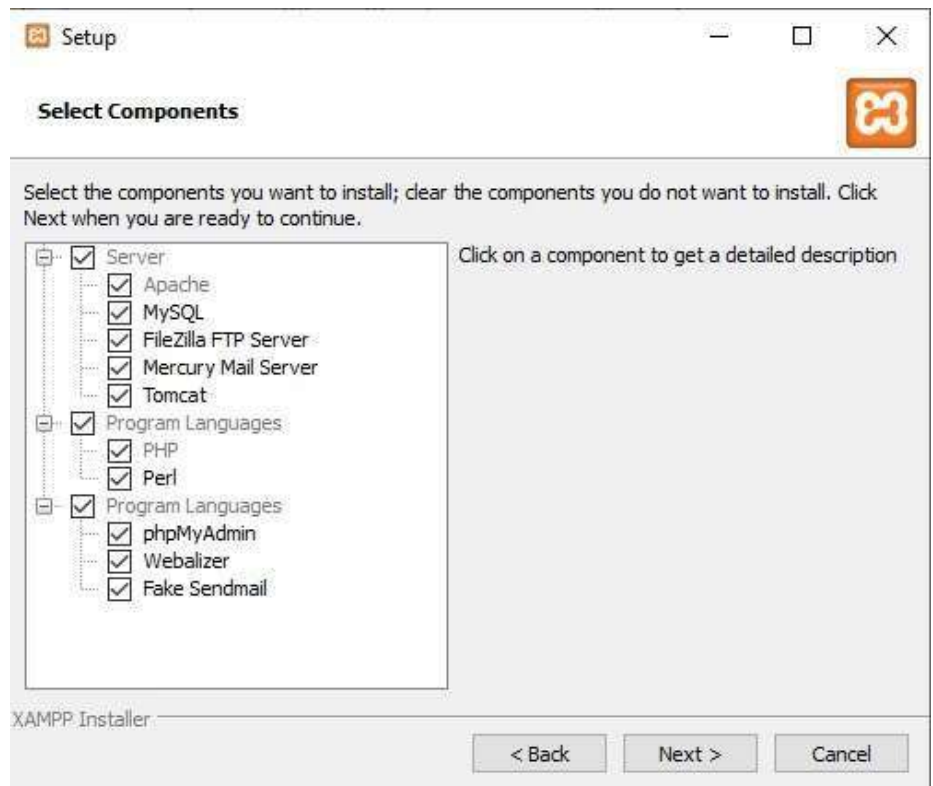
3. Selanjutnya anda akan dialihkan ke jendela yang isinya meminta anda untuk menginstall aplikasi Xampp



4. Klik Next untuk melanjutkan.
5. Selanjutnya anda akan diminta untuk memilih aplikasi yang mau di *install*. Centang saja semua pilihan dan klik tombol Next.

## SISTEM INVENTORY

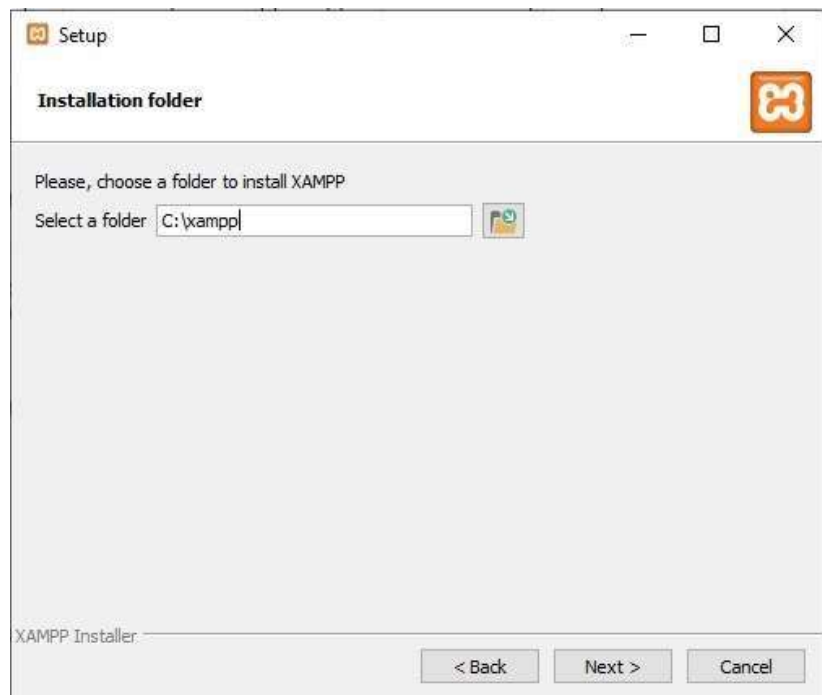
---



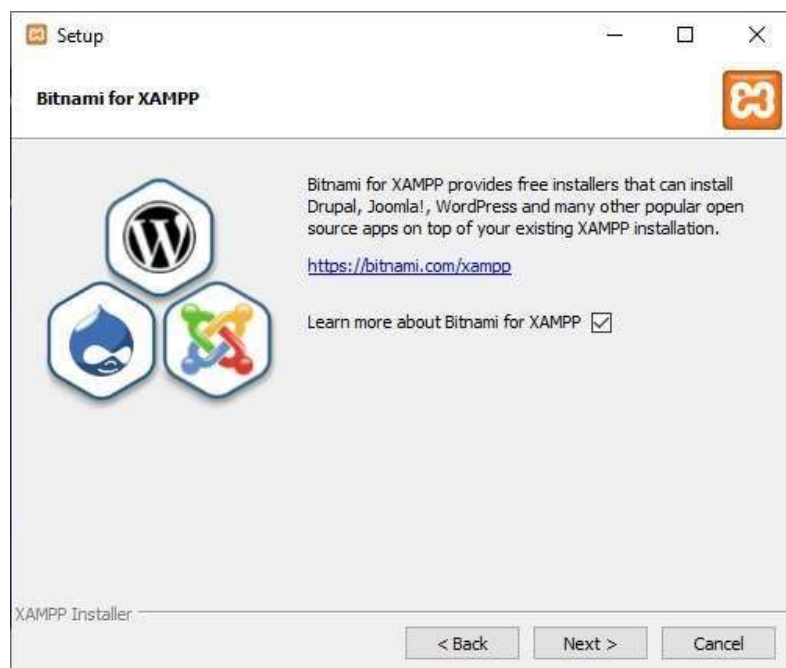
6. Kemudian anda akan diminta untuk menentukan lokasi folder penyimpanan file-file dan folder xampp. Secara *default* akan diarahkan ke lokasi C:\Xampp.
7. Tetapi jika anda ingin menyimpannya di folder lain maka anda bisa mengklik Next lalu pilih Browse kemudian menentukan secara manual folder yang anda ingin gunakan.
8. Jika sudah selesai, lanjutkan dan silahkan klik tombol *install*.

## SISTEM INVENTORY

---



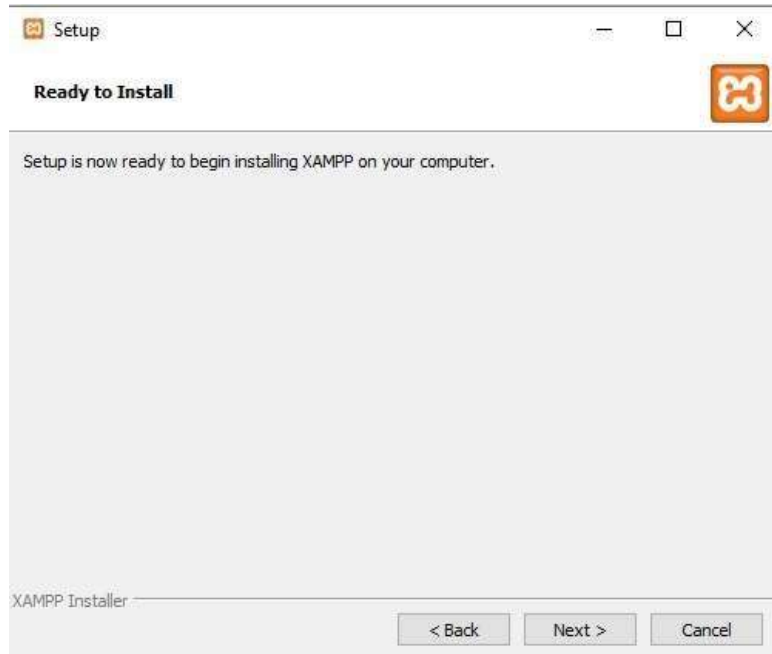
9. Selanjutnya akan muncul tampilan seperti gambit berikut, anda tinggal klik Next saja.



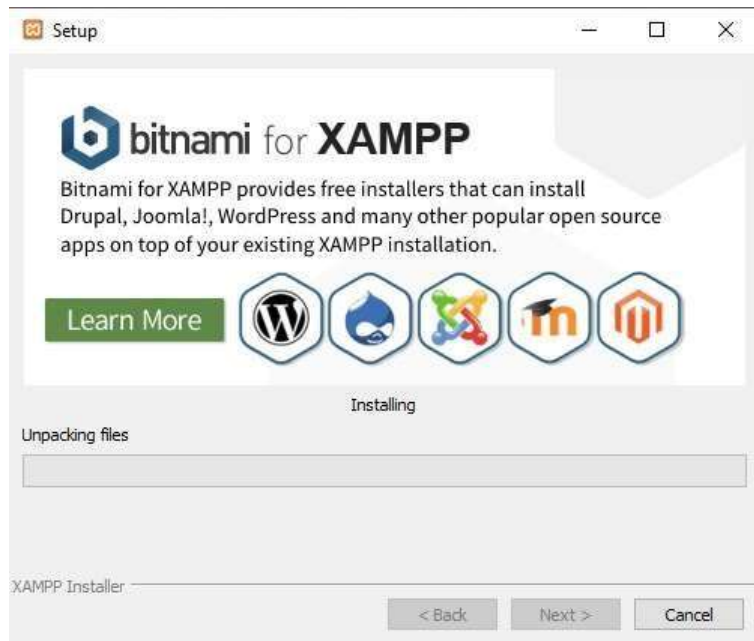
## SISTEM INVENTORY

---

10. Selanjutnya akan muncul tampilan bahwa xampp siap untuk di install. Anda tinggal klik Next saja.



11. Tunggu sampai proses instalasi selesai.

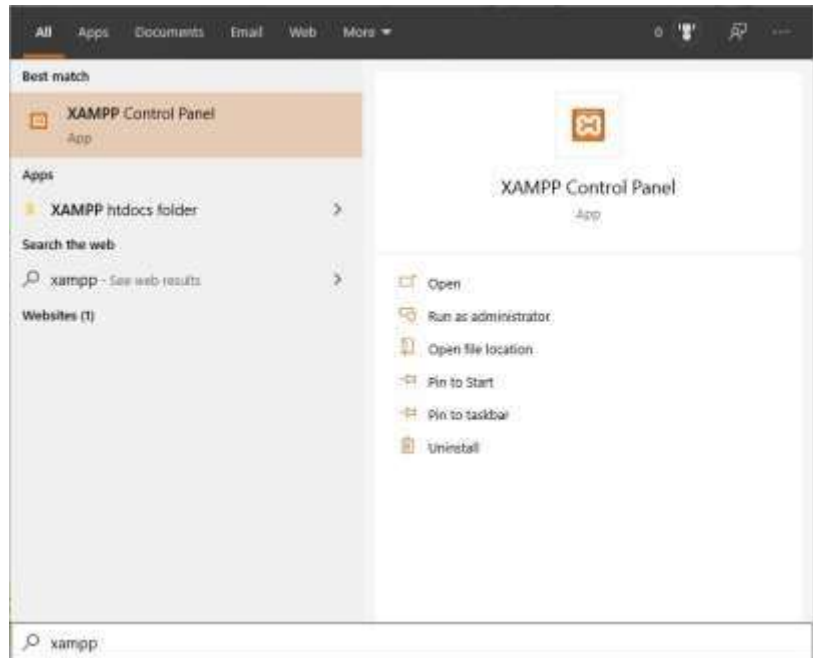


# SISTEM INVENTORY

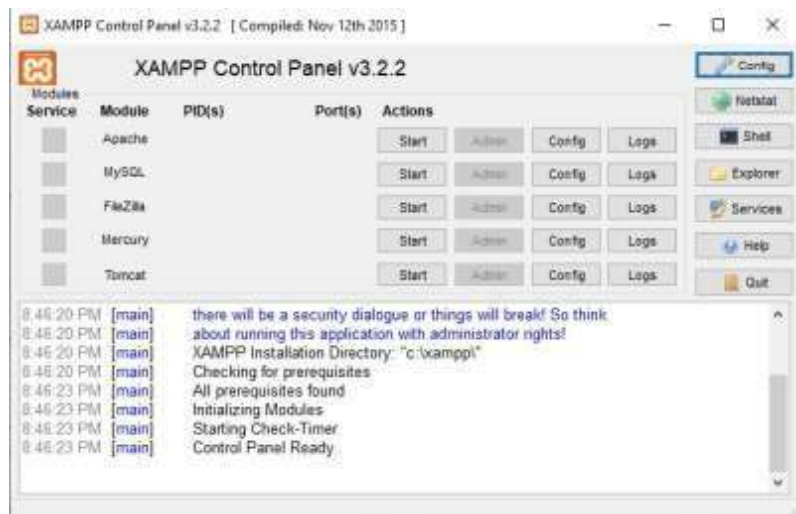
---

## Cara Menjalankan Aplikasi Xampp

1. Bulakah aplikasi Xampp bisa melalui pencarian di task bar atau di desktop



2. Klik dua kali pada xampp tersebut, maka akan muncul tampilan seperti berikut:

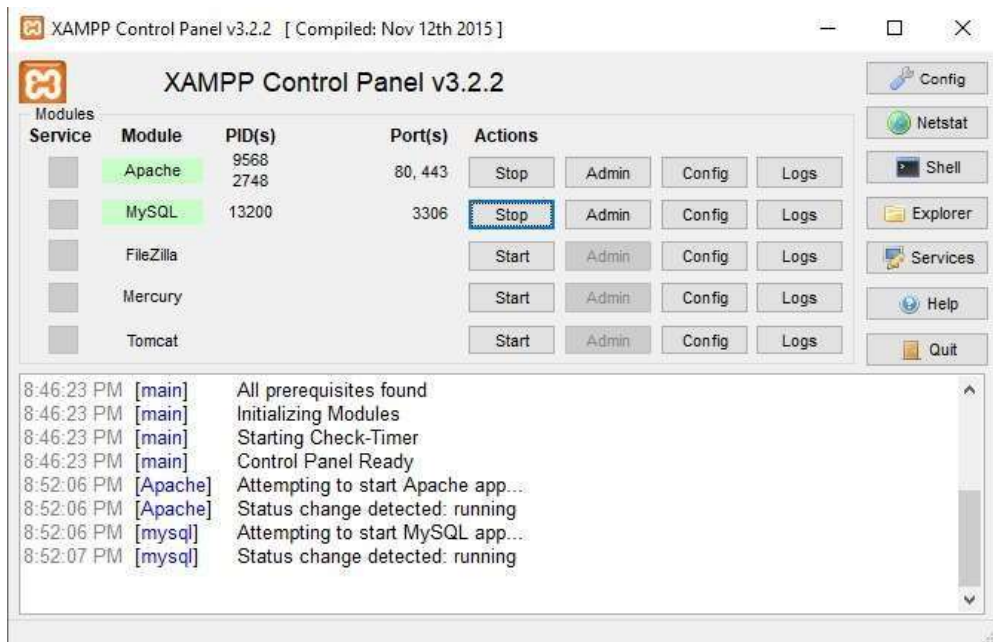




## SISTEM INVENTORY

---

3. Silahkan klik tombol *start* pada kolom *action* sehingga tombol tersebut berubah menjadi *stop*.
4. Dengan mengklik tombol tersebut artinya bahwa aplikasi tersebut telah dijalankan. Biasanya saya menggunakan xampp yang start hanyalah aplikasi Apache dan MySQL karena saya tidak memerlukan aplikasi lainnya.



5. Selanjutnya buka Browser anda dan cobq ketikkan <http://localhost/> di *address bar*, jika muncul tampilan seperti gambar dibawah ini berarti proses instalasi telah berhasil.

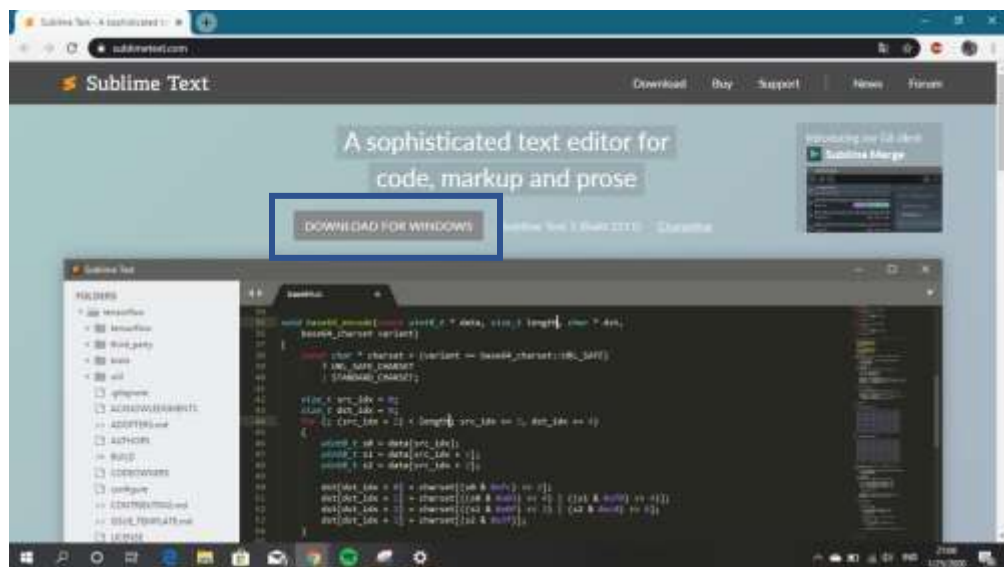
# SISTEM INVENTORY

---



## 3.1.2 Sublime

1. Download terlebih dahulu file sublime di situs resmi sublime,  
link : <https://www.sublimetext.com/>



## SISTEM INVENTORY

---

2. Setelah selesai di *download* klik dua kali pada file .exe



3. Selanjutnya akan muncul tampilan awal untuk proses instalasi.

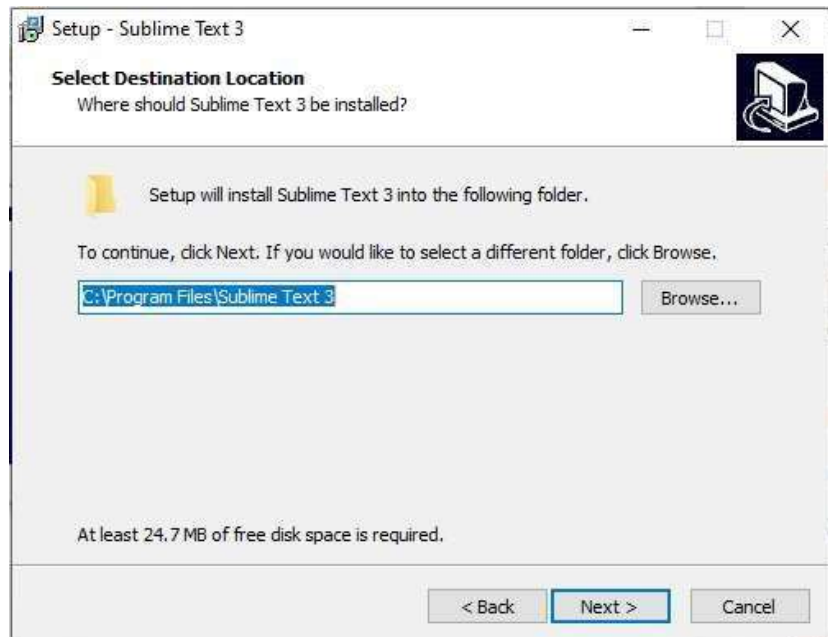


4. Klik Next untuk melanjutkan.
5. Setelah itu akan muncul tampilan library penyimpanan sublime. Secara default akan diarahkan ke lokasi C:\Program Files\Sublime Text 3

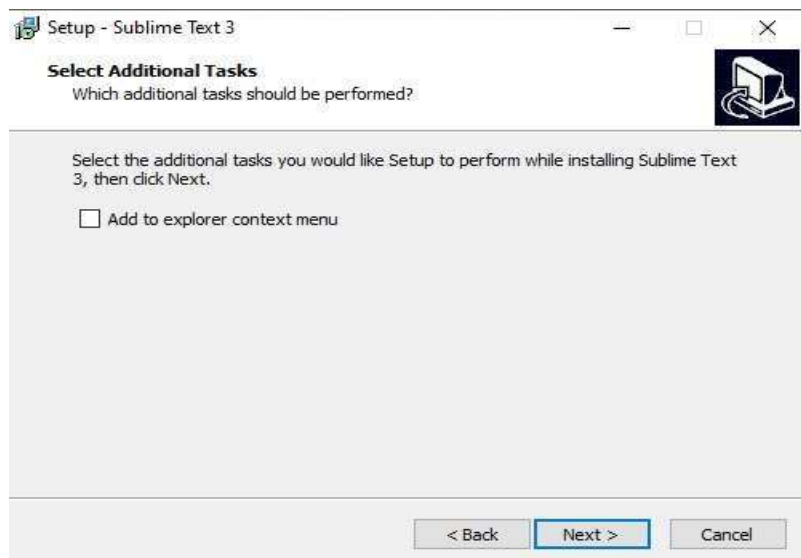
## SISTEM INVENTORY

---

6. Tetapi jika anda ingin menyimpannya di folder lain anda bisa mengklik *Browse* kemudian menentukan secara manual folder yang anda ingin gunakan.
7. Jika sudah selesai lanjutkan dan silahkan klik tombol *Next*.



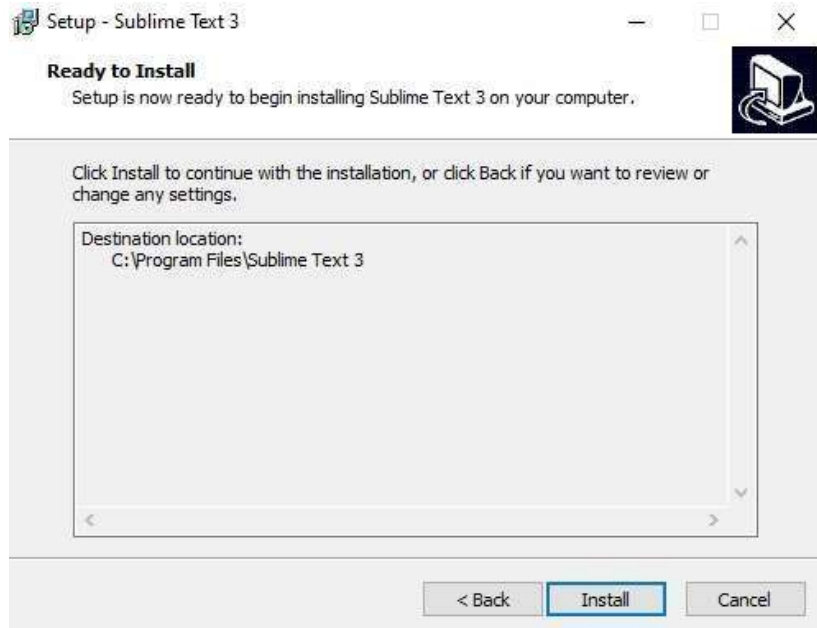
8. Selanjutnya akan muncul tampilan, abaikan dan klik *Next*.



## SISTEM INVENTORY

---

9. Setelah itu klik install untuk melanjutkan proses instalasi, tunggu sampai selesai.



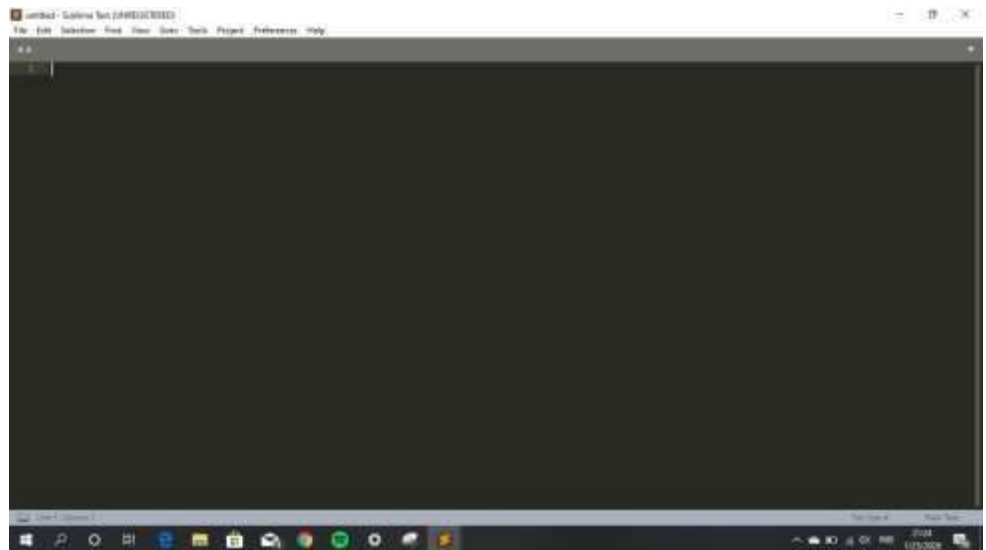
10. Setelah proses instalasi berhasil klik *finish*.



# SISTEM INVENTORY

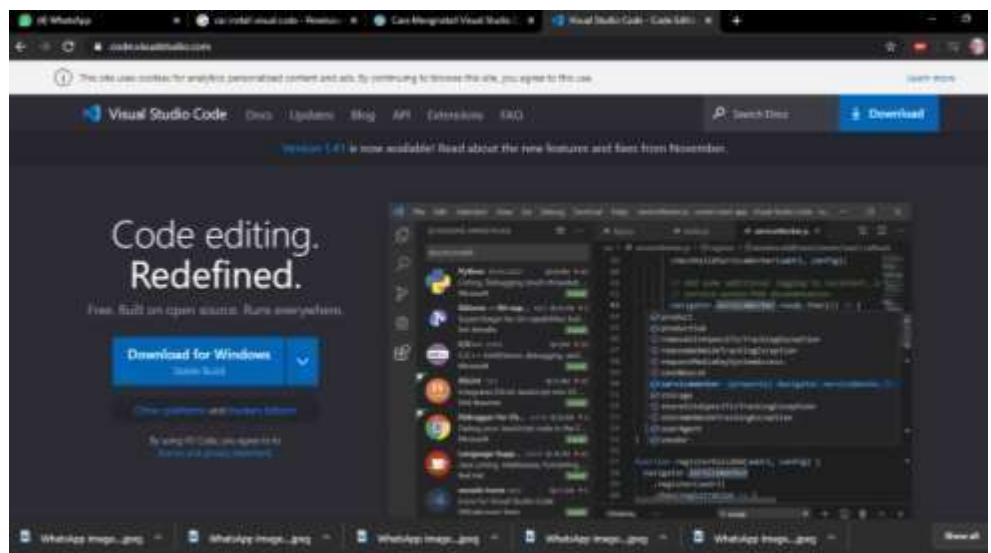
---

## 11. Tampilan awal sublime



### 3.1 Visual Code

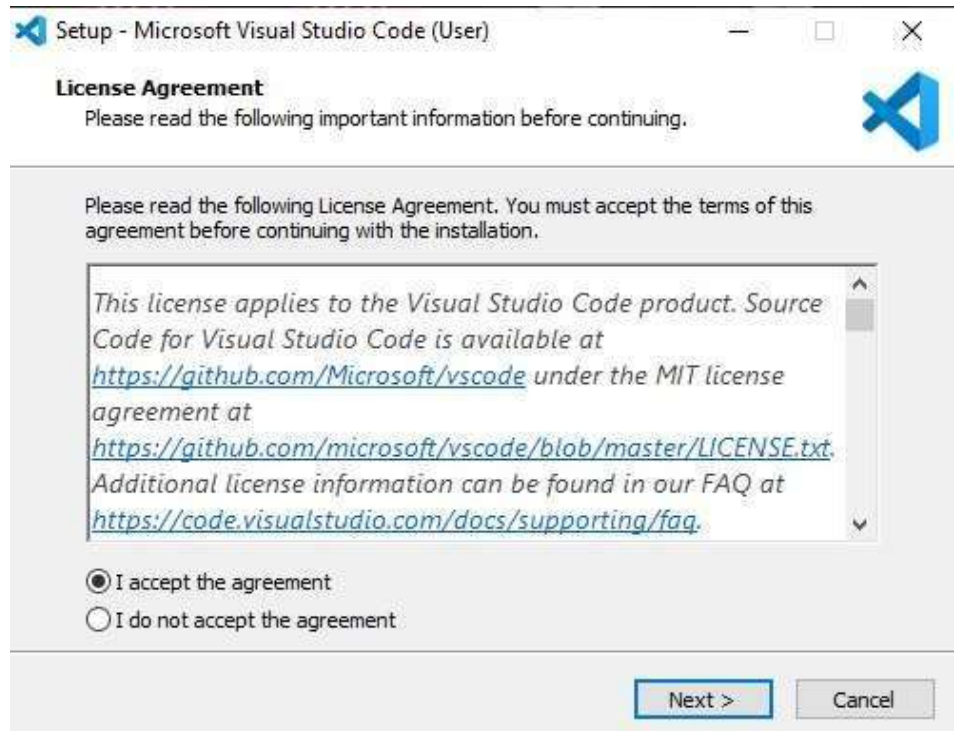
1. Pertama download aplikasi visual code, link <https://code.visualstudio.com/> .
2. Jika link sudah di buka klik Download di pojok kanan . seperti pada gambar dibawah ini.



## SISTEM INVENTORY

---

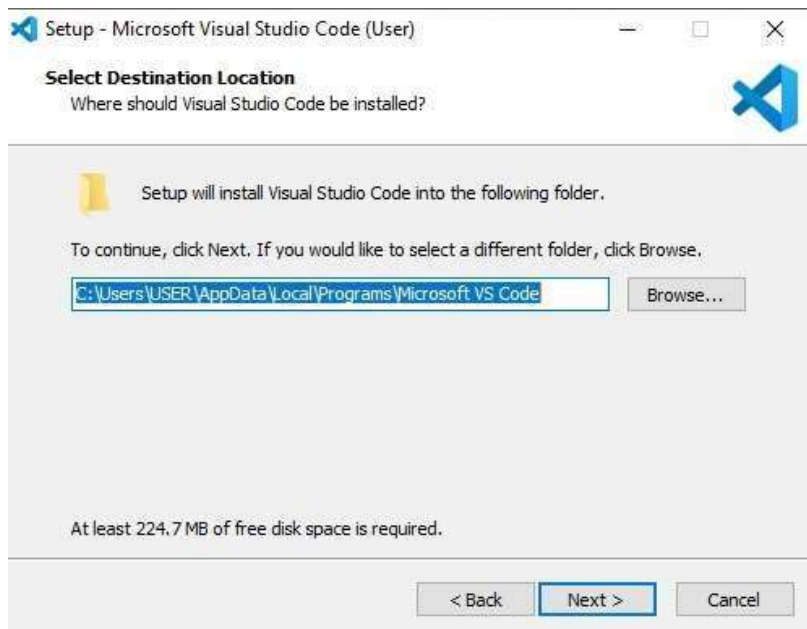
3. Setelah download selesai, buka aplikasi visual code yang telah di-*download* tadi, maka akan muncul *license agreement*.



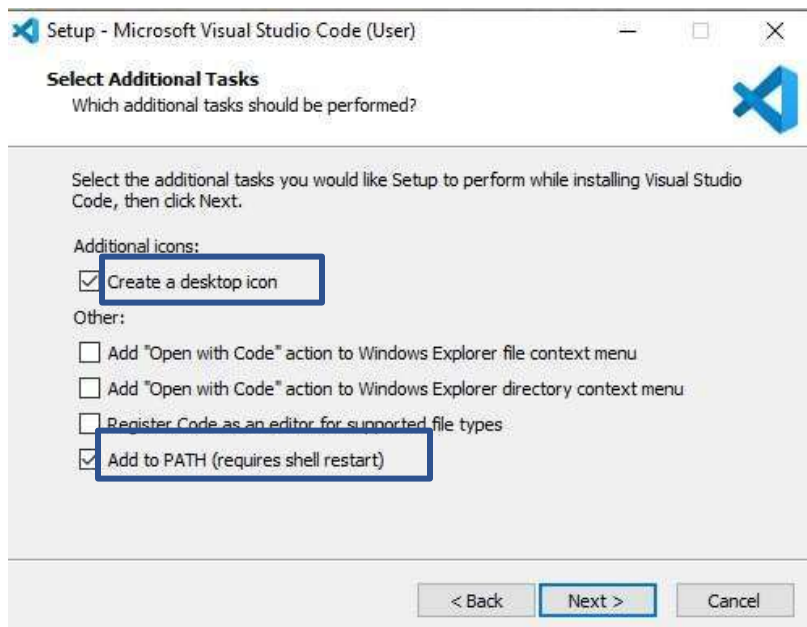
4. Pilih "*I accept the agreement*" untuk menyetujui kebijakan visual code, lalu untuk melanjutkan klik *Next*.
5. Setelah itu akan muncul tampilan library penyimpanan sublime. Secara default akan diarahkan ke lokasi C:\User\USER\AppData\Local\Programs\Microsoft VS Code
6. Tetapi jika anda ingin menyimpannya di folder lain anda bisa mengklik *Browse* kemudian menentukan secara manual folder yang anda ingin gunakan.
7. Jika sudah selesai lanjutkan dan silahkan klik tombol *Next*.

## SISTEM INVENTORY

---



8. Setelah itu akan tampilan seperti pada gambar di bawah ini, ceklis “*Create Dekstop Icon*” jika ingin membuat shortcut VSCode nya kemudian ceklis “*Add to PATH(require shell restart)*” jika sudah klik *Next*.

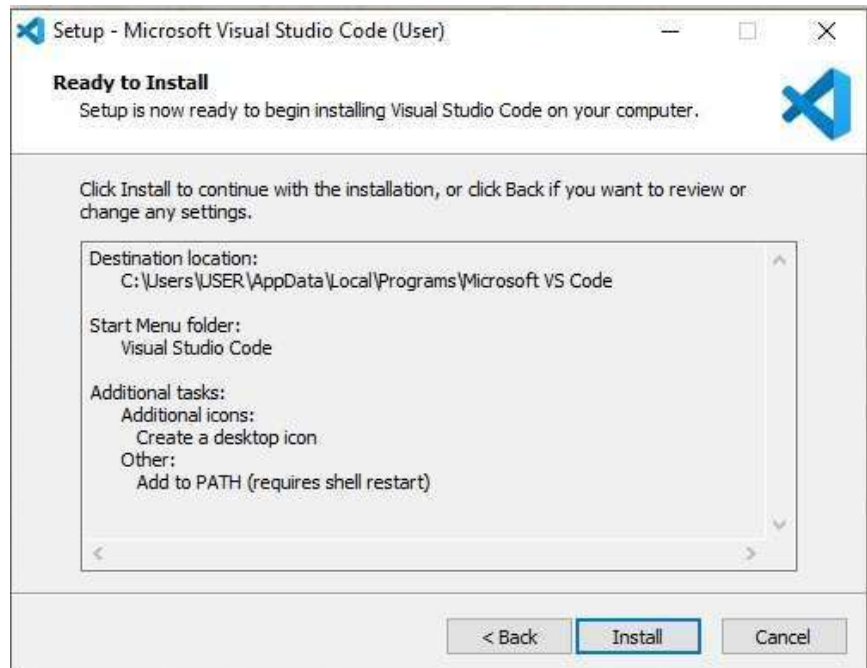




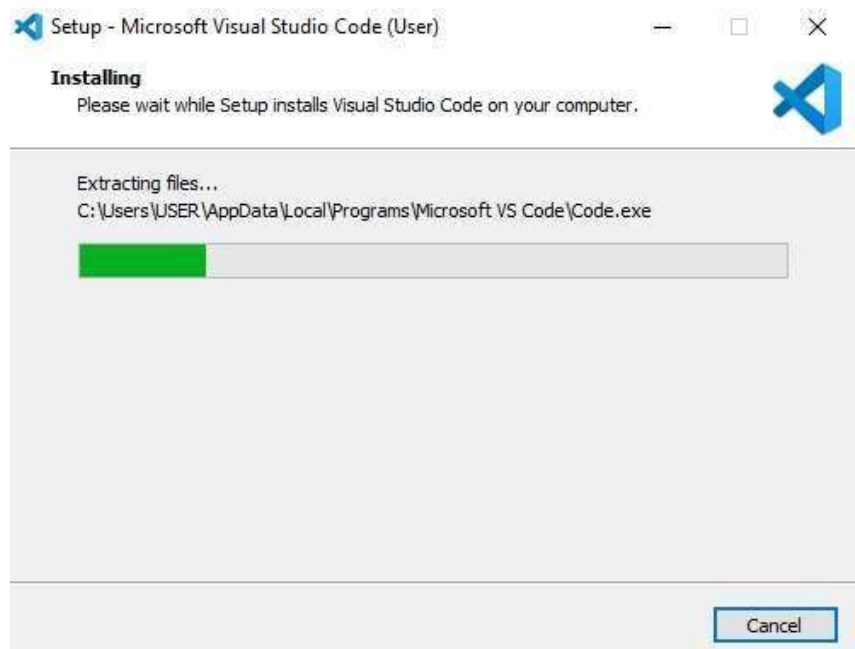
## SISTEM INVENTORY

---

9. Selanjutnya klik *install* untuk melakukan proses instalasi.



10. Proses instalasi dimulai, tunggu sampai proses selesai.



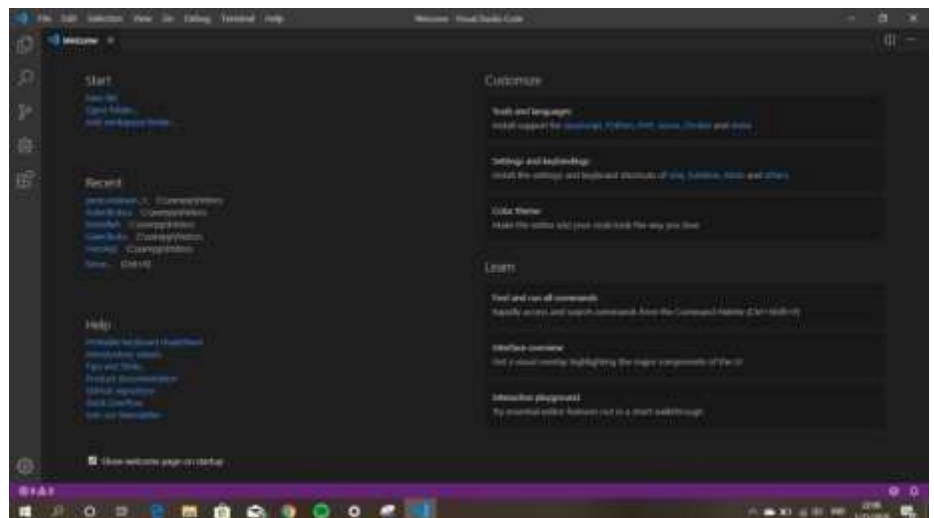
## SISTEM INVENTORY

---

11. Setelah muncul tampilan seperti gambar dibawah ini, artinya instalasi telah berhasil.



12. Berikut adalah tampilan awal dari visual code.



### PENJELASAN *FRAMEWORK* DAN BAHASA YANG DAPAT DIGUNAKAN

#### 3.1 *Framework CodeIgniter*

##### 3.1.1 *Pengertian Framework*

*Framework* adalah suatu kumpulan instruksi yang dikumpulkan dalam kelas dan *function-function* yang memiliki fungsi masing-masing untuk memudahkan *developer* dalam pemanggilan tanoa harus menuliskan *syntax* program yang sama berulang-ulang. Hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan *syntax* secara berulang-ulang agar terciptanya suatu *source code* yang bersih dan terstruktur.

Dalam *framework* sudah menyediakan fungsi *library* dan peralatan lainnya yang kita butuhkan. Salah satu *framework* yang cukup populer di Indonesia adalah *CodeIgniter*.

##### 3.1.2 *Pengertian CodeIgniter*

*CodeIgniter* adalah sebuah *framework* PHP yang bersifat *open source* dan menerapkan salah satu metode MVC (*Model View Controller*). Dalam *CodeIgniter* anda bisa menggunakannya secara gratis atau bersifat *free* dengan kata lain tidak berbayar. *Framework CodeIgniter* dibuat dengan tujuan yang sama seperti *framework* lainnya yaitu; memudahkan *developer* atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuat semuanya dari awal.

*CodeIgniter* dirilis pertama kali pada tanggal 28 Februari 2006. *CodeIgniter* cocok digunakan untuk membuat suatu aplikasi web seperti :

## SISTEM INVENTORY

---

- Portal berita,
- Sistem informasi,
- *Web start up*,
- *Profile company*,
- *E-Commerce*, dan
- *Blog*.

### **Kelebihan *CodeIgniter***

Ada beberapa kelebihan *codeigniter* dibandingkan dengan *framework-framework* yang lain yaitu :

- Performa cepat  
*Framework* ini merupakan *framework* yang paling cepat dibandingkan dengan yang lain, karena tidak menggunakan *template engine* dan ORM yang dapat memperlambat suatu proses.
- Konfigurasi yang minim (*Nearly zero configuration*)  
Untuk menyesuaikan dengan database dan keluasaan *routine* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file seperti *database.php* atau *autoload.php*, namun dalam menggunakan CI hanya menerapkan aturan standar, kita hanya perlu mengubah sedikit saja pada file di folder *config*.
- Memiliki banyak komunitas  
Di Indonesia komunitas CI lumayan ramai, selain itu tutorialnya pun mudah dicari dan kadang-kadang bisa dijadikan ajang diskusi dalam komunitas tersebut.

## SISTEM INVENTORY

---

- Dokumentasi yang lengkap

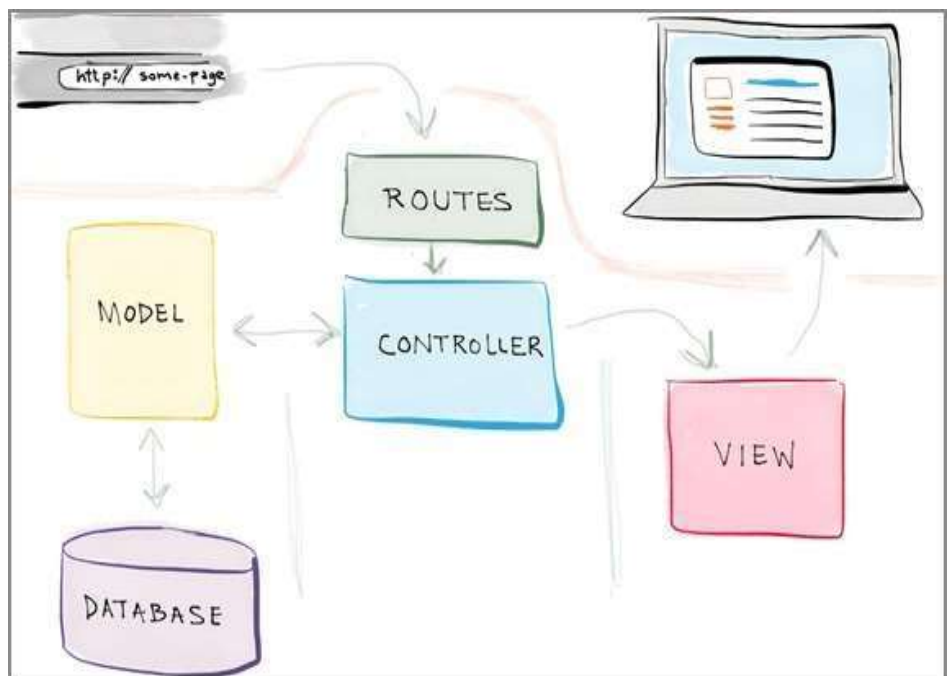
Ketika kita mendownload *CodeIgniter* didalam folder tersebut sudah disertai dengan *user\_guide* yang berisi dokumentasi yang lengkap dan bisa diakses secara *offline*.

- Mudah dipelajari pemula

Bagi pemula *codeigniter* sangat mudah dipelajari karena tidak perlu bergantung pada *tools* tambahan seperti composer ORM *template engine*.

- Metode *Model View Controller*

*Model view controller* atau MVC adalah suatu Teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu *model*, *view* dan *controller*.



- *Model*, *model* adalah suatu bagina penanganan yang berhubungan dengan pengolahan atau manipulasi database,

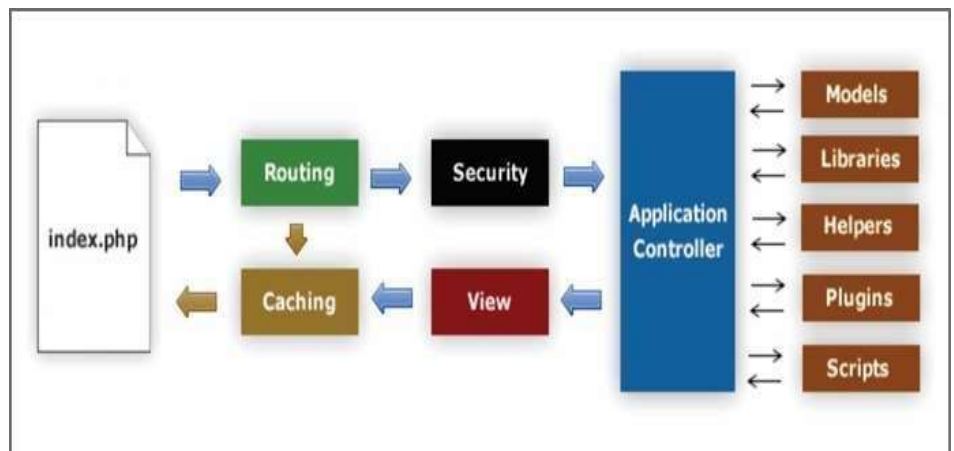
## SISTEM INVENTORY

---

misalnya mengambil data dari database, *meninput* dari pengolahan database lainnya, sehingga semua instruksi yang berhubungan dengan pengelolaan database diletakkan didalam model.

- *View*, *View* merupakan suatu bagian yang menangani halaman *user interface* atau halaman yang muncul pada *user*. Tampilan dari *user interface* dikumpulkan di *view* untuk memisah dengan *controller* dan *model* sehingga memudahkan *web designer* dalam melakukan pengembangan tampilan suatu halaman *website*.
- *Controller*, *controller* adalah suatu kumpulan instruksi atau aksi yang menghubungkan model dan *view* jadi *user* tidak akan berhubungan dengan model secara langsung begitu juga dengan *view* karena *controller* yang mengelolah instruksi.

Didalam *codeigniter* secara detail konsep MVC digambarkan sebagai berikut:



## SISTEM INVENTORY

---

Pada gambar diatas file *index.php* memiliki peran sebagai *controller* utama yang memanggil fungs-fungsi dasar yang digunakan untuk menjalankan *controller*. *Router* memeriksa HTTP *request* selanjutnya memuttusakn *controller* mana yang akan digunakan untuk menangani *request* tersebut.

Apabila file *cahce* tersedia alur aplikasi akan dilewati dan file *cahce* tersebut akan dikirimkan ke *Browser* pengguna. Sebelum *controller* dipanggil, HTTP *request* dan data yang dikirimkan pengguna akan di sortir terlebih dahulu dengan alasan keamanan. Kemudian *controller* memanggil file model, *library*, *helper* dan file lain yang dibutuhkan untuk menangani HTTP *request*. Hasil akhirnya akan ditampilkan oleh file *view* setelah itu dikirimkan ke *Browser* untuk di tampilkan. Jika modus *chacing* diaktifkan hasil *view* akan di *chace* terlebih dahulu. Sehingga jika ada *request* yang sama bisa langsung bisa digunakan.

- Mendukung PHP4 dan PHP5  
Saat ini PHP telah mencapai versi 5 bahkan versi tapi masih banyak juga orang-orang yang masih menggunakan PHP4, oleh akrena itu pengembang *framework codeigniter* memperhatikan betul setiap pengguna sehingga *framework codeigniter* dikembangkan agar mampu berjalan baik menggunakan PHP versi apapun
- *Codeigniter* mendukung banyak RDBMS (*Relational Database Management System*) seperti *MySQL*, *MySQLi*, *SQL Server*, *Oracle*, *Maria DB*, *PostgreSQL*, *SQLite*, dan lainnya.
- Mendukung SEO (*Search Engginie Optimization*) karena *codeigniter* pada dasarnya menganut *clean URL* sehingga

## SISTEM INVENTORY

---

setiap aplikasi yang dibangun menggunakan *codeigniter* lebih mudah di *index* oleh *search engine* populer seperti google, yahoo, msn dan lainnya.

- *Codeigniter* memungkinkan pengembangan web menggunakan *library* atau *helper* yang tidak disediakan oleh *codeigniter* seperti google map API, Facebook API, Fpdf dan lain sebagainya.
- *Codeigniter* bersifat tidak kaku sehingga memberika kebebasan kepada *developer web* untuk mengembangkan aplikasi berbasis *web* bahkan tanpa *framework*.
- Kompatibilitas dengan *Hosting*  
*Codeigniter* mampu berjalan pada hamper semua *platform hosting* karena *codeigniter* mendukung semua database yang paling umum atau yang sering digunakan.
- Sangat Mudah di Integrasikan  
*Codeigniter* sangat mengerti tentang pengembangan berbagai *library* saat ini. Maka dari itu *codeigniter* memberikan kemudahan untuk di intergasikan dengan *library-library* yang tersedia saat ini.
- *Small Footprint*  
*Small footprint* yaitu minim jejak, atau lebih sederhananya ketika *user* melihat tampilan website *codeigniter* maka akan sulit ditebak. Hal ini sangat berguna bagi *developer*, ketika ada musuk *attacker* lebih sulit melihat *framework* yang digunakan contohnya saja penggunaan wordpress yang dapat langsung dilihat hanya menggunakan meta datanya.



### **Kekurangan *CodeIgniter***

Dalam *codeigniter* tidak hanya memiliki kelebihan saja sama seperti *framework* lainnya *codeigniter* juga memiliki kekurangan yaitu diantaranya:

- *Codeigniter* tidak ditujukan atau disarankan untuk pembuatan web dengan skala yang besar.
- *Library* yang sangat terbatas. Hal ini dikarenakan sangat sulit mencari *pluggin* tambahan yang terverifikasi secara resmi, karena pada situsnya sendiri tidak menyediakan *plugin-plugin* tambahan untuk mendukung pengembangan aplikasi dengan CI.
- Belum memiliki editor khusus *codeigniter* sehingga dalam membuat suatu *project* dan modul-modulnya harus berpindah-pindah antara satu folder dengan folder yang lainnya.
- *Codeigniter* dikembangkan oleh Ellislab yang dimana bukanlah suatu komunitas sehingga menyebabkan *update core engine* nya tidak secepat *framework* yang lain.
- Memiliki banyak kelonggaran dalam hal *coding* contohnya bebas menambahkan file.
- *Codeigniter* tidak mencerminkan MVC yang sesungguhnya, contohnya penulisan *echo* masih dapat dilakukan pada file *controller*.

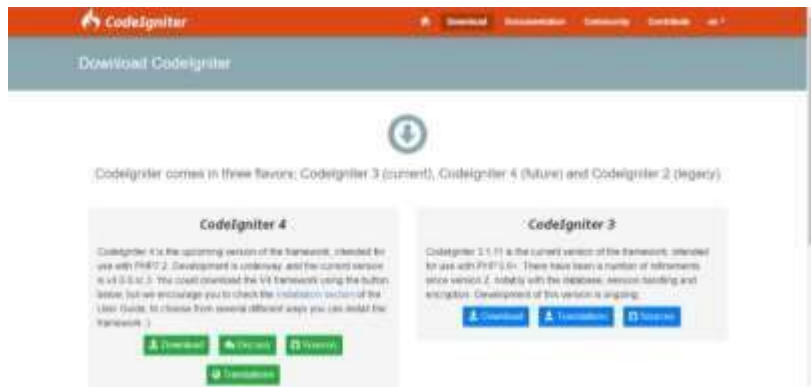
Dari penjelasan MVC diatas dapat di simpulkan bahwa *controller* sebagai penghubung *view* dan model. Contohnya pada aplikasi yang menampilkan data dengan menggunakan metode MVC, *controller* memanggil instruksi pada model yang mengambil

data pada database, kemudian *controller* yang meneruskannya pada *view* untuk ditampilkan.

Jadi sudah jelas bahwa dalam pengembangan aplikasi dengan menggunakan konsep MVC ini sangat mudah dan membantu *web designer* atau *frontend developer* tidak perlu lagi berhubungan dengan *view* untuk mendesign tampilam aplikasi, sedangkan bagian *backand developer* mereka yang menangani bagian *controller* dan modelnya. Jadi pembagian tugas pun menjadi pun dan pengembangan aplikasi dapat dilakukan dengan cepat.

### 3.1.3 Cara Menggunakan *CodeIgniter*

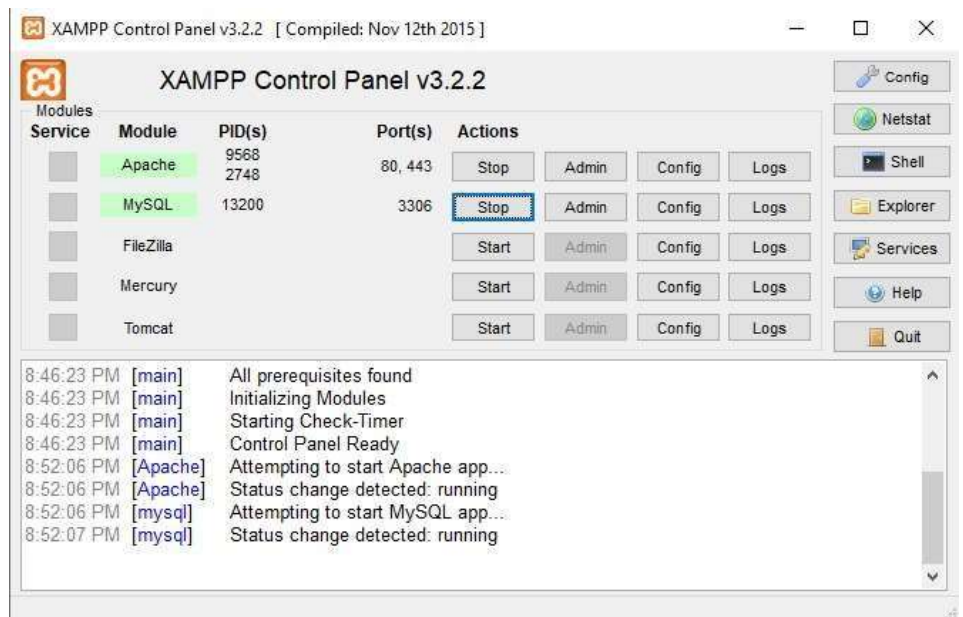
1. Download terlebih dahulu *codeigniter* di link <https://codeigniter.com/download>
2. setelah link tersebut di buka , pilih sesuai dengan kebutuhan masing masing



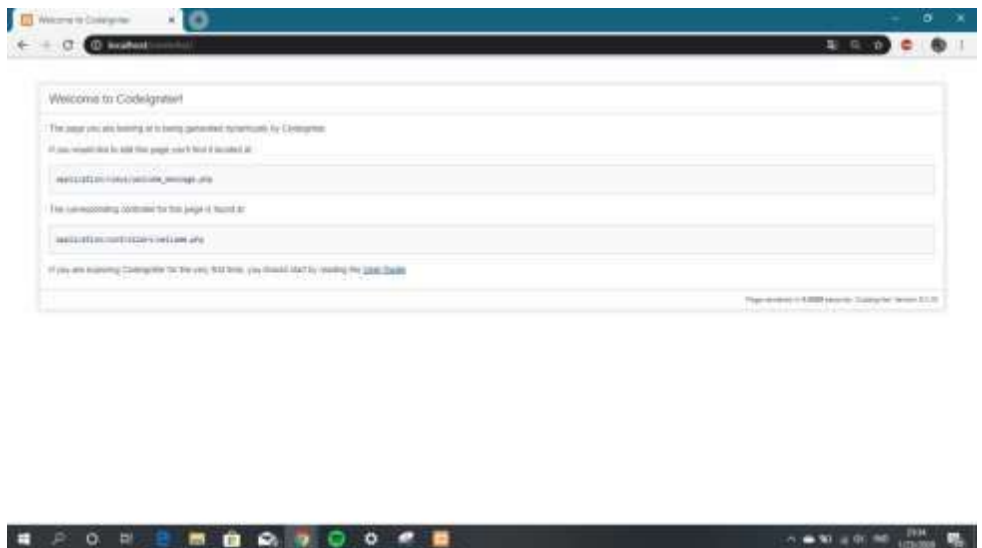
3. kemudian *extract* file yang telah berhasil di *download*.



# SISTEM INVENTORY



7. Selanjutnya buka *Browser* dan ketikkan <http://localhost/contohhci/>



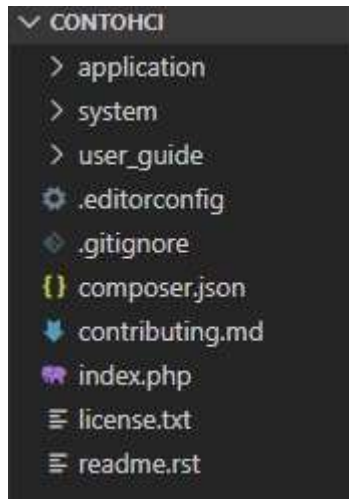
8. Selamat *codeigniter* berhasil di install.

## SISTEM INVENTORY

---

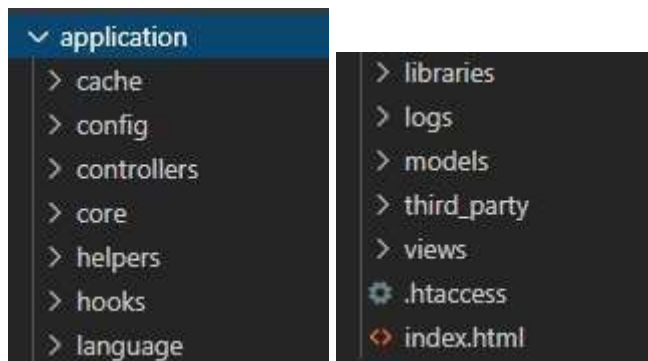
9. Setelah itu buka VisualCode yang telah di install tadi, hal ini dilakukan untuk mengenali struktur direktori pada *codeigniter*.

Berikut adalah struktur direktori *codeigniter*



Terdapat dua direktori penting didalam CI yaitu folder *application* dan folder *System* . selain juga terdapat direktori folder *user\_guide* dan beberapa file. Berikut adalah beberapa penjelasan yang terdapat dalam CI:

- Folder *Application*

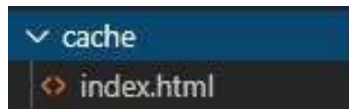


## SISTEM INVENTORY

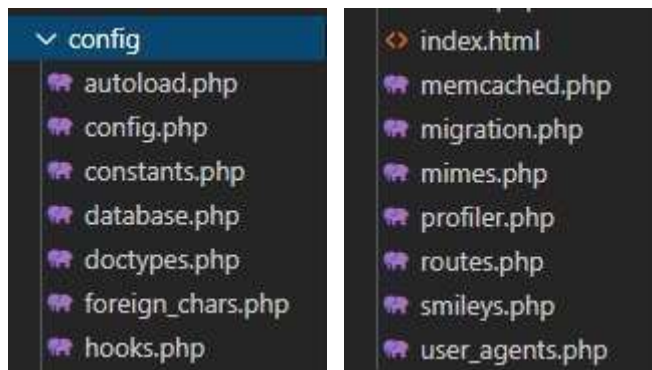
---

Folder *application* berisi code aplikasi. Disinilah kita akan menulis semua baris kode aplikasi yang akan kita bangun. Dalam folder *application* terdapat beberapa direktori yaitu:

- Folder *chace*, adalah suatu folder yang berisi *chace* dari aplikasi.



- Folder *config*, adalah suatu folder yang berisi konfigurasi aplikasi.



- Auotoload.php*, merupakan suatu tempat yang mendefinisikan *autoload*.
- Config.php*, adalah suatu file untuk mengkonfigurasi aplikasi.
- Constants.php*, merupakan suatu file yang berisi konstanta.
- Database.php*, merupakan suatu file yang berisi konfigurasi database aplikasi.
- Doctypes.php*, adalah suatu file yang berisi definisi untuk *doctype html*.

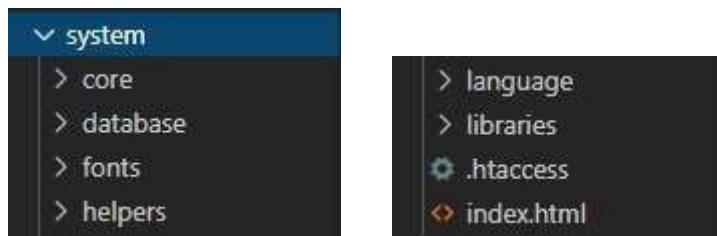
## SISTEM INVENTORY

---

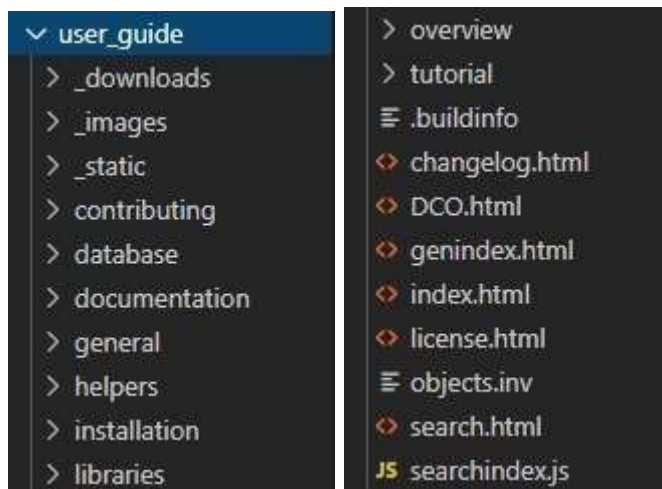
- f. *Foreign\_chars.php*, merupakan suatu file yang berisi *character* dan *symbol*.
- g. *Hooks.php*, merupakan suatu file yang berisi konfigurasi *hooks*.
- h. *Index.html*, merupakan suatu file untuk mencegah *direct access*.
- i. *Memchaced.php* , merupakan suatu file yang berisi suatu konfigurasi untuk *menchace*.
- j. *Migration.php*, merupakan suatu file yang berisi untuk migrasi.
- k. *Mimes.php*, merupakan suatu file yang berisi definisi *type file*
- l. *Profiler.php*, merupakan suatu file yang berisi konfigurasi untuk *profiler*.
- m. *Routes.php*, merupakan suatu file yang digunakan sebagai tempat untuk menulis *route* aplikasi.
- n. *Smileys.php*, berisi kode untuk emoji.
- o. *User\_agents.php*, merupakan suatu file yang berisi definisi untuk *user\_agents*.
- Folder *System*, Folder ini berisi kode-kode inti dari *codeigniter*. Perlu diingat bahwa janganlah anda mengubah apapun didalam direktori ini, jika anda ingin *menuprage* versi ke yang baru anda cukup *mereplace* direktori tersebut dengan yang baru.

## SISTEM INVENTORY

---



- Folder *User-guide*, Folder ini berisi dokumentasi lengkap dari *codeigniter*, dengan folder ini kita bisa mengakses dokumentasi CI secara *offline*. Anda bisa menghapus direktori ini saat *web* yang anda buat telah jadi.



- File *.editorconfig*, File ini berisi konfigurasi untuk *text* editor.



- File *.gitignore*, File ini berisi daftar file dan folder yang akan diabaikan oleh git.

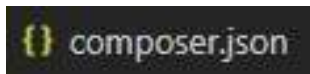


- File *Composer .json*, File ini adalah suatu yang berisi keterangan project dan keterangan *library* yang digunakan, selain itu file ini dibutuhkan oleh composer

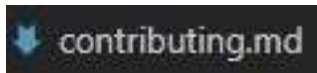


## SISTEM INVENTORY

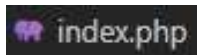
---

A dark rectangular box with a yellow curly brace icon on the left and the text "composer.json" in white.

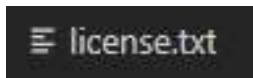
- File *Contributing.md*, file ini adalah suatu file yang berisi penjelasan tentang cara berkontribusi diproyek CI. File ini dapat dihapus apabila *web* sudah jadi.

A dark rectangular box with a blue arrow icon pointing down on the left and the text "contributing.md" in white.

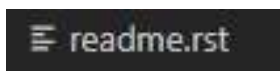
- File *Index.php*, file ini adalah suatu file utama dari *codeigniter* yang merupakan file yang akan dibuka pertama kali saat kita mengakses *web*.

A dark rectangular box with a purple elephant icon on the left and the text "index.php" in white.

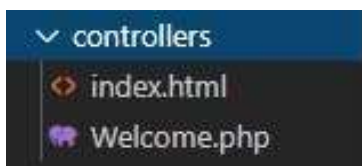
- File *license.txt*, file ini berisi keterangan lisensi dari *codeigniter*

A dark rectangular box with a white icon of three horizontal lines on the left and the text "license.txt" in white.

- File *Readme.rst*, File ini sama seperti dengan file *contributing.md*. file ini berisi penjelasan dan informasi tentang proyek *codeigniter*. File ini dapat dihapus saat *web* sudah selesai.

A dark rectangular box with a white icon of three horizontal lines on the left and the text "readme.rst" in white.

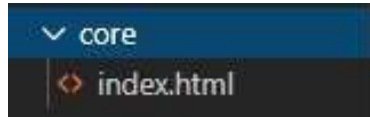
- Folder *Controller*, merupakan suatu folder yang berisi kode-kode *controller*.

A dark rectangular box with a blue header containing a white downward arrow and the text "controllers". Below the header, there are two items: "index.html" with an orange icon and "Welcome.php" with a purple elephant icon.

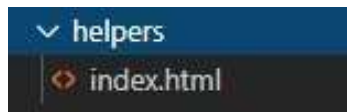
## SISTEM INVENTORY

---

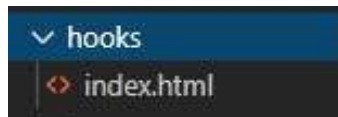
- Folder *Core*, merupakan suatu folder yang berisi kode-kode untuk *coustome core*.



- Folder *Helpers*, merupakan suatu folder adalah fungsi-fungsi helper.



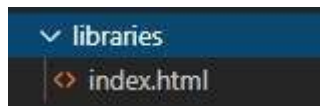
- Folder *Hooks*, merupakan suatu folder yang berisi kode untuk *script hook*.



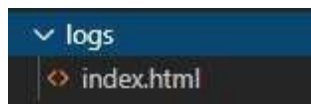
- Folder *language*, merupakan suatu folder yang berisi *string* untuk bahasa apabila web mendukung multi bahasa.



- Folder *Libraries*, merupakan suatu folder yang berisi *library*.



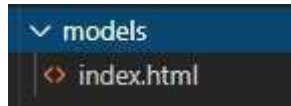
- Folder *Logs*, merupakan suatu folder yang berisi *logs* dari aplikasi.



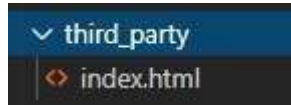
## SISTEM INVENTORY

---

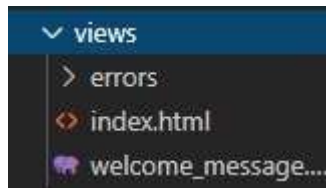
- Folder *models*, berisi kode-kode untuk model



- Folder *Third\_party*, merupakan suatu folder yang berisi *library* dari pihak ketiga

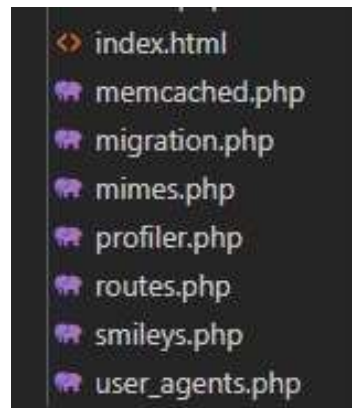
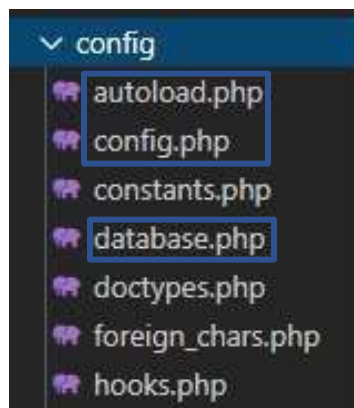


- Folder *Views*, merupakan suatu folder yang berisi kode suatu *view*.



### Konfigurasi Dasar *CodeIgniter*

Dalam memulai *codeigniter* ada beberapa konfigurasi yang harus dilakukan pada beberapa file yaitu file *autoload.php*, file *config*, dan *database.php*. Semua konfigurasi tersebut terletak didalam folder *application/config*.

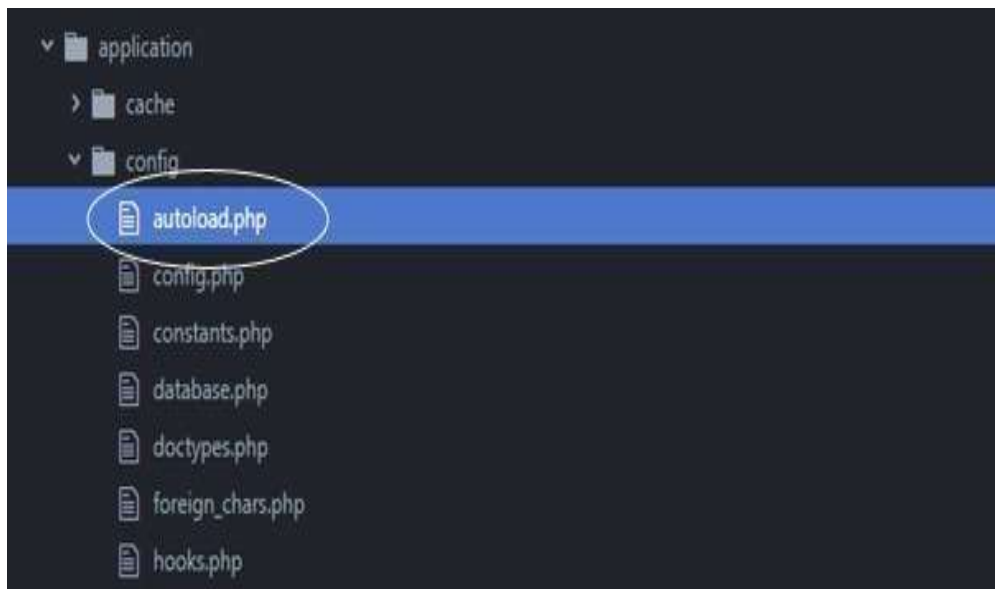


## SISTEM INVENTORY

---

- *Autoload.php*

Pada file ini digunakan untuk mengatur fungsi-fungsi yang akan di muat secara otomatis diawal ketika suatu program dijalankan. Untuk melakukan konfigurasi pada file *autoload.php* silahkan buka folder *application/config/autoload.php*.



Ada beberapa hal yang bisa di *load* secara otomatis dalam *codeigniter* yaitu *package*, *libraries*, *drivers*, *helper files*, *custom config files*, *language files*, dan *models*.

Untuk konfigurasi dasar yang perlu anda ketahui adalah *libraries* dan *helper files* hal tersebut memiliki tujuan agar beberapa *library* dan *helper* tertentu berjalan otomatis.

Untuk melakukan konfigurasi pada *libraries* buka file *autoload.php* dengan *text editor* kemudia temukan kode berikut :

```
$autoload['libraries'] = array();
```

## SISTEM INVENTORY

---

Setelah itu atur menjadi seperti gambar berikut:

```
$autoload['libraries'] = array('database');
```

Pada kode diatas artinya kita akan *meload library database* secara otomatis, sehingga anda dapat menggunakan fungsi-fungsi database pada *codeigniter*.

Selanjutnya kita akan melakukan konfigurasi pada *helper files*, kemudian temukan kode berikut:

```
$autoload['helper'] = array();
```

Setelah itu atur menjadi seperti gambar berikut:

```
$autoload['helper'] = array('url');
```

Pada kode diatas artinya kita akan *meload helper URL* secara otomatis. Dengan demikian anda dapat menggunakan fungsi-fungsi URL pada *codeigniter*, misalnya fungsi *base\_url()*, *site\_url()*, *URI Segment*, dan sebagainya.

- *Config.php*

Pada file ini terdapat beberapa konfigurasi yang secara standar sudah terkonfigurasi, namun terdapat beberapa konfigurasi yang perlu diperhatikan:

- *Base\_url*, merupakan url dasar dari *project* anda. Temukan kode berikut:

```
$config['base_url'] = '';
```

Kemudian atur menjadi gambar berikut:

## SISTEM INVENTORY

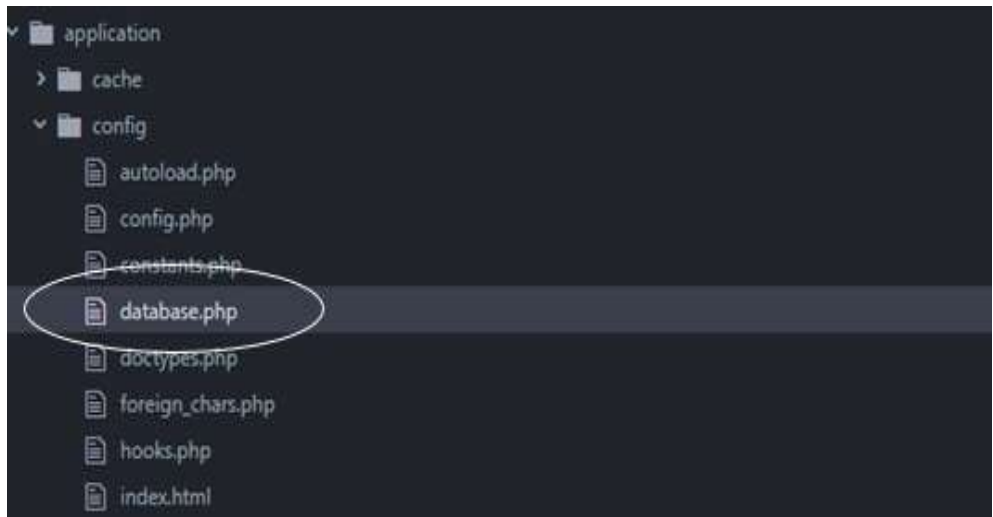
---

```
$config['base_url'] = 'http://localhost/contohci/';
```

- *Database.php*

Dilihat dari nama filenya anda pasti sudah mengetahui apa fungsi dari file ini. File *database.php* berfungsi untuk melakukan konfigurasi yang berkaitan dengan konfigurasi database dari *website* yang akan di buat. Konfigurasi yang perlu diperhatikan tersebut diantaranya: *hostname*, *username*, *password*, dan *database*.

Untuk melakukan konfigurasi pada *database.php* buka file *database.php* dengan *text* editor.



Kemudian temukan kode berikut :

## SISTEM INVENTORY

---

```
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => '',
    'password' => '',
    'database' => '',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

Setelah itu ubah kode tersebut seperti gambar dibawah ini :

```
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'contohci',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

### 4.1 Bahasa Pemrograman Yang Dapat Digunakan

#### 4.2.1 PHP (*Hypertext Preprocessor*)

PHP adalah singkatan dari *Hypertext Preprocessor* yang merupakan bahasa pemrograman yang bersifat *open source* yang berada didalam server yang diproses di server. PHP salah satu bahasa pemrograman yang memiliki script yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*).[9]

PHP merupakan bahasa pemrograman pelengkap HTML yang memungkinkan aplikasi web dinamis untuk pengolahan data, pemrosesan data dari *user via form*, membuat buku tamu, toko online, dan lain sebagainya. Dengan mudah PHP dapat melakukan koneksi ke database karen PHP memang dilengkapi fitur yang memungkinkan koneksi ke PHP dilakukan dengan mudah tanpa harus memusingkan (Tim EMS, 2016:55).

PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya akan dijalankan diserver tetapi disertakan pada dokumen HTML. Pembuatan web ini merupakan kombinasi antara php sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman web (Bimo sunarfrihantono, ST 2002:9).





*Gambar Hypertext Preprocessor*

Sumber : <https://www.webhozz.com/>

Menurut Andi Pramono dan M. Syafii (2004 : 2), PHP atau kependekan dari *Hypertext Preprocessor* adalah sebuah bahasa pemrograman berbasis *web* yang mempunyai banyak keunggulan dibandingkan dengan bahasa pemrograman berbasis *web* yang lain. PHP merupakan bahasa pemrograman yang bersumber dari Perl. Sedangkan Perl merupakan pengembangan dari bahasa C.

Oleh karenanya, struktur pemrograman yang ada di PHP sama dengan yang ada di bahasa C. Melihat bahwa PHP merupakan pengembangan dari bahasa C secara tidak langsung, maka PHP mempunyai banyak sekali fitur-fitur yang dapat digunakan. Misalnya, PHP dapat mengakses shell di Linux, mempunyai fungsi yang lengkap berhubungan dengan *networking*.

## SISTEM INVENTORY

Salah satu keunggulan PHP dibanding bahasa pemrograman lainnya adalah PHP dapat diperoleh secara gratis, meskipun bukan berarti karena gratis kemampuannya menjadi pas-pasan. PHP sangat powefull. Terbukti dengan banyaknya website yang dibangun menggunakan PHP.

PHP juga terkenal lebih aman daripada bahasa pemrograman website yang lain. PHP juga sudah mendukung OOP (*Object Oriented Programming*) sehingga *maintenance* kode menjadi jauh lebih muda dibandingkan procedural.

PHP bisa berinteraksi dengan database, file dan folder, PHP ini termaksud bahasa *cross-platform*, yang mana PHP ini bisa berjalan di sistem operasi yang berbeda-beda (Windows, Linux, ataupun MAC).[10]

Berdasarkan pernyataan diatas dapat disimpulkan bahwa PHP adalah bahasa pemrograman yang sangat fleksibel untuk digunakan dalam membuat sebuah website, dan juga PHP dapat dijalankan di bawah sistem operasi LINUX dan Windows.

## 4.2.2 HTML



## SISTEM INVENTORY

---

### *Gambar Script HTML*

Sumber : enimamedia.com

HTTP merupakan protokol yang digunakan untuk mentransfer data antara *web server* ke *web Browser*. Protokol ini mentransfer dokumen-dokumen web yang ditulis atau berformat HTML. Dikatakan markup karena HTML berfungsi untuk "mepercantik" file text biasa untuk ditampilkan pada program web Browser. Hal ini dilakukan dengan menambahkan tag-tag (perintah khusus) pada file teks biasa tersebut.

Tag HTML biasanya berupa tag-tag yang berpasangan pasangan dan ditandai dengan simbol `<dan>`. Pasangan dari sebuah tag ditandai dengan tanda `'/'`. misalnya awalnya perintah tagnya `<contoh>` maka diakhiri dengan `</contoh>`.

Supaya dapat menghasilkan tampilan wujud yang terintegrasi Pemformatan hiperteks sederhana ditulis dalam berkas format ASCII sehingga menjadi halaman web dengan perintah-perintah HTML.

HTML merupakan sebuah bahasa yang bermula bahasa yang sebelumnya banyak dipakai di dunia percetakan dan penerbitan yang disebut Standard Generalized Markup Language (SGML).

Sekarang ini HTML merupakan standar Internet yang dikendalikan dan didefinisikan pemakaiannya oleh World Wide Web Consortium (W3C). Pada tahun 1989, HTML dibuat oleh kolaborasi Berners-lee Robert dengan Caillau TIM pada saat mereka bekerja di CERN (CERN merupakan lembaga penelitian fisika energi tinggi di Jenewa).

### 4.2.2.1 Fungsi HTML (*HyperText Markup Language*)

HTML (*HyperText Markup Language*) adalah suatu bahasa yang menggunakan tanda-tanda tertentu (*tag*) untuk menyatakan kode-kode yang harus ditafsirkan oleh *Browser* agar halaman tersebut dapat ditampilkan secara benar.

Secara umum, fungsi HTML adalah untuk mengelola serangkaian data dan informasi sehingga suatu dokumen dapat diakses dan ditampilkan di Internet melalui layanan *web*.

Fungsi HTML yang lebih spesifik yaitu :

- Membuat halaman *web*.
- Menampilkan berbagai informasi di dalam sebuah *Browser Internet*.
- Membuat link menuju halaman web lain dengan kode tertentu (*hypertext*).
- HTTP atau *Hypertext Transfer* Protokol merupakan protokol yang digunakan untuk mentransfer data atau document yang berformat HTML dari *web server* ke *web Browser*. Dengan HTTP inilah yang memungkinkan Anda menjelajah internet dan melihat halaman *web*.

### 4.2.2.2 Sejarah HTML

HTML dibuat oleh Tim Berners-Lee, seorang ahli fisika di lembaga penelitian CERN yang berlokasi di Swiss. Dia memiliki ide tentang sistem *hypertext* yang berbasis internet.

*Hypertext* merujuk pada teks yang memuat referensi (link) ke teks lain yang bisa diakses langsung oleh viewer. Tim merilis versi pertama HTML pada tahun 1991, dan di dalamnya terdiri atas 18 HTML tag. Sejak saat itu, setiap kali bahasa HTML

merilis versi teranyarnya, selalu ada *tag* dan *attribute* (*tag modifier*) terbaru.

Berdasarkan *HTML Element Reference* milik Mozilla *Developer Network*, untuk saat ini, ada 140 HTL tag meskipun sebagiannya sudah usang (tidak lagi didukung oleh versi terbaru Browser).

Berkat popularitasnya yang terus meningkat, HTML kini dianggap sebagai web standard yang resmi. Spesifikasi HTML di-*maintain* dan dikembangkan oleh *World Wide Web Consortium* (W3C). Cek versi terbaru dari bahasa ini di website W3C.

*Upgrade* HTML besar-besaran terjadi pada tahun 2014, dan hasilnya adalah pengenalan HTML5. Pada *upgrade* tersebut, terdapat semantic baru yang memberitahukan arti dari kontennya sendiri, seperti <artcile>, <header>, dan <footer>.

### 4.2.2.3 Cara Kerja HTML

Dokumen HTML adalah file yang diakhiri dengan ekstensi .html atau .htm. Ekstensi file ini bisa dilihat dengan menggunakan web Browser apa pun (seperti Google Chrome, Safari, atau Mozilla Firefox). Browser tersebut membaca file HTML dan me-*render* kontennya sehingga user internet bisa melihat dan membacanya.

Biasanya, rata-rata situs web menyertakan sejumlah halaman HTML yang berbeda-beda. Contohnya, beranda utama, halaman ‘tentang kami’, halaman kontak yang semuanya memiliki dokumen HTML terpisah.

## SISTEM INVENTORY

---

Masing-masing halaman HTML terdiri atas seperangkat tags (bisa disebut juga elements), yang mengacu pada building block halaman website. Tag tersebut membuat hirarki yang menyusun konten hingga menjadi bagian, paragraf, heading, dan *block* konten lainnya.

Sebagian besar element HTML memiliki tag pembuka dan penutup yang menggunakan syntax `<tag></tag>`.

Berikut contoh kode dari susunan atau struktur HTML:

```
<div>
  <h1>The Main Heading</h1>
  <h2>A catchy subheading</h2>
  <p>Paragraph one</p>
  
  <p>Paragraph two with a <a
href="https://example.com">hyperlink</a></p>
</div>
```

Elemen teratas dan terbawah adalah division sederhana (`<div></div>`) yang bisa Anda gunakan untuk mark up bagian konten yang lebih besar.

Susunan HTML di atas terdiri atas heading (`<h1></h1>`), subheading (`<h2></h2>`), dua paragraf (`<p></p>`), dan satu gambar (`<img>`).

Paragraf kedua meliputi sebuah link (`<a></a>`) dengan attribute href yang terdiri atas URL tujuan.

Tag gambar memiliki dua attribute, src untuk path gambar dan alt untuk deskripsi gambar.

### 4.2.2.4 Gambaran Umum Tentang Tag HTML yang Paling Sering Digunakan

Tag HTML memiliki dua tipe utama: *block-level* dan *inline tags*. Elemen *block-level* memakai semua space yang tersedia dan selalu membuat line baru di dalam dokumen. Contoh dari *tag block* adalah *heading* dan *paragraf*.

Elemen *inline* hanya memakai *space* sesuai dengan kebutuhannya dan tidak membuat line baru di halaman. Biasanya elemen ini akan memformat isi konten dari elemen *block-level*. Contoh dari *tag inline* adalah *link* dan *emphasized strings*.

- *Tag Block-Level*

Tiga *tag block-level* yang harus dimiliki oleh setiap dokumen HTML adalah `<html>`, `<head>`, dan `<body>`.

Tag `<html></html>` adalah elemen level tertinggi yang menyertakan setiap halaman HTML.

Tag `<head></head>` menyimpan informasi meta, seperti judul dan charset halaman.

Tag `<body></body>` melampirkan semua konten yang muncul pada suatu halaman.

```
<html>
  <head>
    <!-- META INFORMATION -->
  </head>
  <body>
    <!-- PAGE CONTENT -->
  </body>
```

## SISTEM INVENTORY

---

</html>

*Heading* memiliki 6 level di HTML. Level tersebut bervariasi, mulai dari <h1></h1> sampai ke <h6></h6>, di mana h1 merupakan level heading tertinggi dan h6 adalah level terendah. Paragraf dibuka dan ditutup dengan tag <p></p>, sedangkan *blockquote* menggunakan tag <blockquote></blockquote>.

*Division* merupakan bagian konten yang lebih besar dan biasanya terdiri atas beberapa paragraf, gambar, kadang-kadang *blockquote*, dan elemen lebih kecil lainnya. Kita bisa membuat mark up dengan menggunakan tag <div></div>. Di dalam elemen div juga terdapat tag div lainnya.

Anda juga bisa menggunakan tag <ol></ol> untuk list yang berurutan dan <ul></ul> untuk list yang tidak berurutan. Masing-masing list item harus dibuka dan ditutup dengan tag <li></li>. Sebagai contoh, di bawah ini adalah tampilan dasar dari list tidak berurutan dalam HTML:

```
<ul>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ul>
```

- *Tag Inline*

Sebagian besar *tag inline* digunakan untuk memformat teks. Sebagai contoh, tag <strong></strong> akan *render* elemen ke format *bold*, sedangkan tag <em></em> akan ditampilkan dalam format *italic*.



*Hyperlink* adalah elemen *inline* yang mewajibkan adanya tag `<a></a>` dan attribute `href` untuk mengindikasi tujuan link:

```
<a href="https://example.com/">Click me!</a>
```

Gambar (*image*) juga merupakan elemen *inline*. Anda dapat menambahkan satu gambar dengan menggunakan `<img>` tanpa harus membubuhkan *tag* penutup. Hanya saja, Anda disarankan menggunakan *attribute* `src` untuk menentukan path gambar, misalnya:

```

```

Jika ingin mempelajari lebih dalam tentang tag HTML, silakan baca artikel kami tentang *HTML cheat sheet* (kami juga menyediakan link untuk diunduh).

### 4.2.2.5 Kelebihan dan Kekurangan HTML

Sama seperti hal teknis lainnya dalam dunia web, HTML juga punya kelebihan dan kekurangannya.

- Kelebihan
  - Bahasa yang digunakan secara luas dan memiliki banyak sumber serta komunitas yang besar.
  - Dijalankan secara alami di setiap *web Browser*.
  - Memiliki *learning curve* yang mudah.
  - *Open-source* dan sepenuhnya gratis.
  - Bahasa markup yang rapi dan konsisten.
  - *Standard web* yang resmi di-maintain oleh *World Wide Web Consortium (W3C)*.
  - Mudah diintegrasikan dengan bahasa *backend*, seperti PHP dan Node.js.

- Kekurangan

Paling sering digunakan untuk halaman *web* statis. Untuk fitur dinamis, Anda bisa menggunakan *JavaScript* atau bahasa backend, seperti PHP.

HTML tidak memungkinkan user untuk menjalankan logic.

Alhasil, semua halaman web harus dibuat terpisah meskipun menggunakan elemen yang sama, seperti header dan footer.

Fitur-fitur baru tidak bisa digunakan secara cepat di sebagian Browser.

Terkadang perilaku Browser susah untuk diprediksi (misalnya, Browser lama tidak selalu bisa render tag yang lebih baru).

### 4.2.3 CSS (*Cascading Style Sheets*)

CSS (*Cascading Style Sheets*) Pada versi HTML yang terdahulu, web Browser mengontrol tampilan (rendering) dari setiap halaman web. Jika menggunakan elemen H1 pada (*large heading*) pada web dokumen, Browser akan merender elemen tersebut.

Dengan adanya CSS, *programmer* dapat mengontrol bagaimana *Browser me-render* halaman web. Mengaplikasikan CSS pada halaman web dapat memberikan tampilan yang lebih menarik dan spesifik sesuai dengan tema pada sebuah web site.

Teknologi CSS memberikan fasilitas untuk menentukan style (misal; *spacing, margins*) dari elemen halaman web terpisah dari struktur dokumen web (*section headers, body text, links* ).

Pemisahan tersebut memberikan peningkatan yang lebih besar dalam pengaturan *web pages*, dan membuat perubahan – perubahan *style* dalam dokumen dapat dilakukan lebih cepat dan lebih mudah.

### 4.2.3.1 Kelebihan dan Kekurangan CSS

#### a. Kelebihan

- *Update* tampilan lebih mudah
- Beban *bandwith* lebih kecil
- Modifikasi *Web template* lebih mudah
- Lebih mudah digunakan pada *mobile phone*
- *Search engine friendly*
- Memisahkan desain dengan konten halaman web
- Mengatur desain seefisien mungkin
- Jika kita ingin mengubah suatu tema halaman web, cukup modifikasi pada css saja
- Menghadirkan sesuatu yang tidak dapat dilakukan oleh HTML
- Lebih mudah didownload karena lebih ringan ukuran filenya
- Satu CSS dapat digunakan banyak halaman web
- Menghemat penulisan kode, karena satu css dapat dipakai beberapa kali
- Mempersingkat waktu kerja, baik saat membuat maupun saat modifikasi halaman *Web/Blog*, dan masih banyak lagi.

#### b. Kekurangan

- Tampilan pada Browser berbeda-beda.

## SISTEM INVENTORY

---

- Kadang juga terdapat *Browser* yang tidak support CSS (*Browser* lama).
- Harus tahu cara menggunakannya.
- Dibutuhkan waktu lebih lama dalam membuatnya.
- Belum lagi ada *bug/error* dalam CSS.

### **BASIS DATA**

#### **1.1 Konsep Basis Data**

Jika misalnya kita memiliki banyak sekali buku dan anggap saja kita punya dua solusi untuk menyimpannya. Solusi pertama adalah menyimpan di sembarang tempat dan yang kedua disimpan di dalam rak yang tersusun rapi dengan kode rak dan seterusnya. Pada saat tersimpan solusi pertama tampak sangat memudahkan terserah kita mau simpan dimana namun pada saat penjualan buku, akan sangat sulit bagi kita untuk menemukan buku yang kita cari karena kita menyimpannya tidak beraturan. berbeda dengan solusi kedua, Pada saat penyimpanan masih diperlukan waktu yang sedikit lebih lama tapi kelebihan waktunya tidak signifikan jika dibandingkan dengan kemudahan pada saat pencarian. Dengan keteraturan pada saat penyimpanan mengakibatkan kemudahan dan kecepatan pada saat pencarian kembali buku yang disimpan

Nah, itulah kira-kira analogo betapa bermanfaatnya basis data. Buku itu kita ibaratkan sebagai data baris rak itu, kita ibaratkan sebagai tabel dan rak buku itu kita ibaratkan sebagai basis data.

#### **Definisi Basis Data**

Basis data dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikianrupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. prinsip utamanya adalah pengaturan data. tujuan utamanya kemudahan dan kecepatan dalam pengambilan kembali data.

# SISTEM INVENTORY

---

## Tujuan Basis Data

Secara lebih lengkap pemanfaatan basis data dilakukan untuk memenuhi tujuan berikut ini

- Kecepatan dan kemudahan (*speed*)
- Profisiensi ruang penyimpanan (*space*)
- Keakuratan (*accuracy*)
- Ketersediaan (*availability*)
- Kelengkapan (*completeness*)
- Keamanan (*security*)
- Pemakaian bersama (*sharability*)

## Penggunaan Basis Data

Setiap organisasi atau perusahaan maka untuk mencapai efisiensi, daya saing, dan kecepatan operasional organisasi atau perusahaan digunakanlah aplikasi sistem informasi yang biasanya menggunakan perangkat komputer. Basis data adalah suatu komponen utama dalam sistem informasi dan tidak ada sistem informasi yang bisa dijalankan tanpa adanya basis data. jadi untuk perusahaan menggunakan aplikasi berbasis komputer maka basis data diperlukan. konkritnya, beberapa pemanfaatan basis data antara lain

- Kepegawaian untuk berbagai perusahaan yang memiliki banyak pegawai.
- Pergudangan atau *inventory* dan penjualan untuk perusahaan manufaktur atau pabrik, supermarket, apotek dan lain-lain.

## SISTEM INVENTORY

---

- Akuntansi, untuk Bank dan perusahaan-perusahaan yang melibatkan uang.
- Reservasi, untuk hotel, pesawat, kereta api
- Catatan medik untuk rumah sakit
- Akademik, untuk perguruan tinggi atau sekolah.
- Penggajian, untuk perusahaan yang memiliki pegawai yang cukup banyak dll.

### **Perancangan Basis Data**

Sebelum kita membuat basis data terlebih dahulu dilakukan perancangan. proses perancangan ini bersifat konseptual. Kita belum mengetahui DBMS apa yang akan kita gunakan untuk mengimplementasikan rancangan basis data yang dibuat. Tujuan perancangan basis data adalah mendapatkan skema basis data yang meminimalisasi terjadinya redundansi dan duplikasi data serta menjaga integrasi data. Kebanyakan metode perancangan berbasis pada model basis data relasional. Pada basis data relasional, dan diatur melalui pembuatan tabel-tabel dan terdapat keterkaitan antar tabel yang satu dengan lainnya (relasi).

Salah satu pemodelan yang sering digunakan untuk merancang basis data relasional adalah *Entity Relationship Diagram*. Dua elemen fundamental pada ER adalah entitas dan *relationship* (keterhubungan).

### 1.2 Entitas

Entitas adalah suatu objek baik itu nyata maupun abstrak di dunia nyata yang dapat dibedakan dari objek lain berdasarkan karakteristik yang dimilikinya. Misalnya : pegawai perusahaan, barang yang dijual ditoko, mahasiswa, dosen, mata kuliah, jurusan, layanan suatu bank, dan sebagainya. Dalam basis data, kita harus menentukan entitas apa saja yang diinformasinya perlu disimpan. Pada ER Diagram, entitas memiliki symbol persegi Panjang.

Informasi yang ingin disimpan dari suatu entitas disebut dengan *property* . misalnya pada entitas pada mahasiswa, beberapa *property* yang mungkin adalah NPM, nama mahasiswa, tanggal lahir, alamat, nomor telepon, dan lainnya. Pada entitas barang, beberapa atribut yang mungkin adalah nomor identitas barang, nama barang, jumlah barang, harga beli, dan harga jual. Pada ER Diagram atribut dilambangkan dengan elips.

Untuk membedakan data pada suatu entitas, biasanya adalah suatu atribut digunakan sebagai *identifier* yang dapat membedakan antara satu data dengan lainnya. Atribut ini disebut dengan *primary key*. *Primary key*, karena NPM merupakan identitas unik dari seorang mahasiswa (tidak ada mahasiswa yang memiliki NPM sama). Pada kasus entitas barang, biasanya setiap barang memiliki nomor identitas barang yang digunakan sebagai *primary key*. Pada entitas mata



## SISTEM INVENTORY

---

kuliah, biasanya digunakan kode mata kuliah sebagai *primary key*.

Pertimbangan pemilihan atribut mana yang dijadikan sebagai *primary key* tergantung dari beberapa hal, antara lain:

- Atribut harus unik. Itu artinya untuk objek/barang yang berbeda atribut tersebut pasti beda juga.
- Nilai atribut tersebut jarang berubah.
- Mudah diingat dan biasanya memiliki format konsisten.

NIM	Nama_mhs	JK	Alamat	Tgl_Lahir	No_tlp
1174017	Muh. Rifky	L	Jln. Sariasih II	03/12/1999	0891638119
1174003	Dwi septiani	P	Jln. Sariasih II	19/04/1999	08531689190
.....	.....	...	.....	...	.....

Untuk tiap atribut, nilai atribut memiliki batasan yang disebut dengan domain constraint. misalnya, atribut merupakan string dengan panjang maksimal 9. jenis kelamin hanya boleh diisi dengan L atau P. atribut mana yang harus diisi dan mana yang tidak boleh diisi. jika suatu atribut nilainya tidak atau belum terdefinisi Dan diperbolehkan untuk tidak diisi dahulu, maka atribut tersebut adalah diberi nilai null. misalnya seorang mahasiswa tidak memiliki telepon, maka nilai kolom masih tersebut adalah *null*. contoh lainnya,

## SISTEM INVENTORY

---

misalkan alamat seorang mahasiswa belum diketahui maka kita bisa memberikan nilai null.

Pada beberapa kasus, suatu atribut dapat mengacu atau merefer ke atribut pada tabel lain ( atau atribut lain pada tabel itu sendiri). atribut yang diracun oleh atribut lain disebut *foreign key*. *Foreign key* adalah suatu cara untuk menjaga integritas data.

### **Relasi**

Relasi adalah sosialisasi yang menyatakan keterhubungan antar entitas. misalnya:

- Relasi mengajar yang menghubungkan antara entitas dosen dengan mata kuliah. Relasi ini menyatakan bahwa seorang dosen hanya boleh mengajar banyak mata kuliah dan satu mata kuliah bisa diajar oleh beberapa orang dosen.
- Relasi tempat antara entitas mata kuliah dengan ruangan. relasi ini menyatakan bahwa satu mata kuliah bisa diajarkan di beberapa ruang yang berbeda- beda dan satu ruangan bisa digunakan untuk mengajar mata kuliah yang berbeda-beda
- Relasi nilai, yang menghubungkan antara entitas mahasiswa dengan mata kuliah. Relasi ini menyatakan bahwa 1 orang mahasiswa memiliki satu nilai untuk satu mata kuliah, namun satu mata kuliah bisa memiliki banyak nilai mahasiswa.

### Kardinalitas

Relasi memiliki derajat keterhubungan. hubungan antar hubungan antar entitas pada suatu relasi seperti pada contoh diatas disebut dengan kardinalitas. Terdapat tiga jenis kardinalitas

- Satu ke satu, misalnya relasi registrasi
- Satu ke banyak, atau banyak ke-1. misalnya relasi tempat
- Banyak-banyak, misalnya relasi nilai.

contoh menentukan kardinalitas :

Misalnya saja di suatu kampus yang boleh menjadi admin adalah dosen

- Yang jadi pertanyaan, berapa maksimal dosen yang boleh menggunakan satu *login* admin tertentu? jawabannya 1. Bolehkah sebuah *login* admin tidak dimiliki oleh dosen (tidak Bertuan?) jawabannya tidak boleh karena munculnya sebuah admin adalah karena ada dosen yang registrasi sebagai admin ( minimal 1)
- Berapa maksimal *login* admin yang boleh dimiliki oleh seorang dosen? jawabannya 1, Bolehkah dosen tidak menjadi admin? jawabannya boleh

Kardinalitas sebuah relasi diambil dari nilai maksimalnya sehingga nilai relasi registrasi mempunyai kardinalitas satu kurang satu.

### Langkah-langkah Membuat ER Diagram

Langkah-langkah membuat er diagram dengan studi kasus *website* akademik. Identifikasi dan tetapkan seluruh himpunan entitas yang akan terlibat. misalnya studi kasus yang kita ambil adalah *website* akademik. objek-objek yang terlibat dengan proses akademik antara lain : dosen, mahasiswa, mata kuliah, ruangan, jadwal, dan admin.

Tentukan atribut-atribut key dari masing-masing himpunan entitas. atribut key adalah atribut yang menjadi Identifier ( pembeda atau identitas) dalam suatu entitas. dosen Keynya adalah NIP, mahasiswa keynya adalah NIM.

Identifikasi dan tetapkan seluruh relasi di antara entitas yang ada beserta *foreign key*nya.

- Antara entitas dosen dan mata kuliah dengan relasinya adalah mengajar
- Antara dosen dengan admin relasinya adalah registrasi
- Antara mahasiswa dengan mata kuliah relasinya adalah nilai
- Antara mata kuliah dengan jadwal relasinya adalah alokasi
- Antara jadwal dengan ruangan relasinya adalah tempat.

*Foreign key* adalah *key* pada relasi yang merupakan Primary Key dari entitas entitas yang terhubung oleh relasi tersebut

## SISTEM INVENTORY

---

- Menentukan derajat kardinalitas relasi untuk setiap himpunan relasi
- Relasi mengajar kardinalitasnya N-N
- Relasi registrasi kardinalitasnya 1-1
- Relasi nilai kardinalitasnya N-N
- Relasi alokasi kardinalitasnya 1-N
- Relasi tempat kardinalitasnya 1 -N

Melengkapi entitas dan relasi dengan atribut-atribut deskriptif :

- Menambahkan atribut nama\_ dosen pada entitas dosen
- Menambahkan atribut nama\_ mhs dan password pada entitas mahasiswa.
- Menambahkan atribut password\_ admin pada entitas admin.
- Menambahkan atribut nama\_ matkul, SKS\_matkul, semester pada entitas mata kuliah.
- Menambahkan atribut nama\_ ruangan pada entitas ruangan.

### 1.3 MySQL

*Database Management System* (DBMS) adalah aplikasi yang dipakai untuk mengelola basis data. DBMS biasanya menawarkan beberapa kemampuan yang terintegrasi seperti:

1. Membuat, menghapus, menambah dan memodifikasi basis data
2. Pada beberapa dbms pengelolaannya berbasis Windows (berbentuk jendela-jendela) sehingga lebih mudah digunakan
3. Tidak semua orang bisa mengakses basis data yang ada sehingga memberikan keamanan bagi data
4. Kemampuan berkomunikasi dengan program aplikasi yang lain. misalnya dimungkinkan untuk mengakses basis data mysql menggunakan aplikasi yang dibuat menggunakan PHP
5. Kemampuan pengaksesan melalui Komunikasi antar komputer (*client server*)

MySQL adalah salah satu aplikasi dbms yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. contoh DBMS lainnya adalah PostgreSQL (*freeware*), SQL server, MS Access, db2 dari IBM, Oracle dan Oracle Corp, dbase, foxpro, dsb.

Kelebihan dari MySQL adalah gratis, handal, selalu di-*update*, dan banyak forum yang memfasilitasi para pengguna jika memiliki Kendala. MySQL juga menjadi DBMS yang sering di bandling dengan *web server* sehingga proses instalasinya jadi lebih mudah.

## SISTEM INVENTORY

---

### Arsitektur Sistem

Arsitektur sistem maksudnya adalah konfigurasi sistem secara keseluruhan yang menjadi tempat hidup dari dbms, basis data dan aplikasi yang memfaatkannya.

Beberapa jenis arsitektur sistem yang dapat digunakan adalah:

a. Sistem tunggal/ Mandiri (*stand alone*)

Pada sistem ini DBMS, basis data serta aplikasi basis data ditempatkan pada komputer yang sama. arsitektur ini paling sederhana dan murah. Arsitektur ini dapat kita pilih jika basis data yang dikelola tidak terlalu besar dan lebih bersifat untuk membantu pekerjaan administratif. Sistem Mandiri dapat digunakan di rumah masing-masing seperti di apotek kecil, wartel, Hotel kecil, dan sebagainya

b. Sistem tersentralisasi (*centralized system*)

Jika yang disentralisasi adalah DBMSnya, maka servernya disebut *application server*. jika yang disentralisasi adalah basis datanya, maka servernya disebut *file server*. sistem tersentralisasi menggunakan *application server* kurang baik untuk spesifikasi *server*-yang rendah. karena pekerjaan pada *server* sangat berat. sistem tersentralisasi menggunakan *file server* kurang baik untuk jaringan yang terlalu luas karena transaksi dan data bisa sangat berat dan keamanan kurang terjaga.

c. Sistem *client server*

Arsitektur ini dibuat untuk menutupi kelemahan pada sistem tersentralisasi. Beban *server* jadi tidak terlalu berat

## SISTEM INVENTORY

---

karena klien juga memiliki dbms sehingga proses yang bisa dilakukan di klien akan dilakukan di *client*. lalu lintas data antara *server* dan Workstation pun dibuat lebih efisien.

### Tipe Data String

MySQL menggunakan seluruh tipe data numerik standar ANSI. Berikut ini adalah tipe data numerik yang biasanya digunakan beserta penjelasannya.

Tipe Data	Deskripsi
<b>INT</b>	nilai integer yang bisa bertanda atau tidak. jika bertanda, maka rentang yang diperbolehkan adalah -2147483648 sampai 2147483647, sedangkan jika tidak bertanda maka rentang nya dari 0 sampai 4294967295
<b>TINYINT</b>	nilai integer yang sangat kecil. Rentangnya 128-127 untuk yang bertanda dan 0- 255 Untuk yang tidak bertanda.
<b>SMALLINT</b>	nilai integer yang sangat kecil dengan rentang 31768 sampai 32767 untuk yang bertanda Sedangkan untuk yang tidak bertanda dari 0 sampai 65535.
<b>MEDIUMINT</b>	integer dengan ukuran sedang dengan rentang - 8388608 sampai 8388607 atau 0 sampai 16777215
<b>BIGINT</b>	Integer dengan ukuran besar dengan rentang - 922337203685775808 sampai



## SISTEM INVENTORY

---

	9223372036854775807 atau 0 sampai 18446744073709551615.
<b>FLOAT(M,D)</b>	bilangan pecahan dengan panjang ( termasuk jumlah desimal) M dan jumlah desimal D. presisi desimalnya bisa sampai 24 digit. defaultnya float ( 10,2). bilangan Float selalu bisa bertanda.
<b>DOUBLE(M,D)</b>	adalah bilangan pecahan dengan presisi dua kali lipat. panjang m dan jumlah desimal d. presisi desimalnya bisa sampai 53 digit. default-nya double (16,4) bilangan float selalu bisa bertanda. sinonim dari double adalah real.
<b>DECIMAL(M,D)</b>	adalah bilangan pecahan dan harus didefinisikan m&d nya. setiap desimal membutuhkan tempat 1 byte. sinonim dari desimal adalah numerik.

### Tipe Data Tanggal Dan Waktu

Berikut ini adalah tipe data tanggal dan waktu didalam MySQL.

Tipe Data	Deskripsi
<b>DATE</b>	Adalah tipe data tanggal dengan format YYYY-MM-DD, antara 1000-01-01 dan 9999-12-31.

## SISTEM INVENTORY

---

	Contoh: 17 Agustus 1945 akan disimpan sebagai 1945-08-17
<b>DATETIME</b>	Adalah kombinasi tanggal dan waktu dengan format YYYY-MM-DD HH:MM:SS dan rentang data antara 1000-01-01 00:00:00 sampai dengan 9999-12-31 23:59:59. Contoh pukul 10:00 pagi pada tanggal 17 Agustus 1945 akan disimpan sebagai 1945-08017 10:00:00.
<b>TIMESTAMP</b>	Sebuah penanda waktu antara 1 Januari 1970 tengah malam sampai dengan tahun 2037. Formatnya mirip dengan DATETIME tetapi tanpa pembatas diantara angkanya. Contoh: pukul 10:00 pagi pada tanggal 27 Agustus 1945 akan disimpan sebagai 19450817100000.
<b>TIME</b>	Menyimpan waktu dalam format HH:MM:SS. Contoh pukul 10:00 akan disimpan menjadi 10:00:00
<b>YEAR(M)</b>	Menyimpan data tahun dalam format 2 atau 4 digit. Jika M diisi dengan nilai 2, maka rentang tahunnya dari 1970-2069 sedangkan jika M diisi dengan nilai 4 maka YEAR bisa berisi 1901 sampai dengan 2155. Default nilai M adalah 4.

## SISTEM INVENTORY

---

### Tipe Data String

Berikut ini adalah tipe data string yang paling umum di dalam MySQL.

Tipe Data	Deskripsi
<b>CHAR</b>	string dengan ukuran tetap. ukurannya antara 1 sampai 255 karakter. Ukuran ditentukan dengan nilai M. Contoh CHAR(6).
<b>VARCHAR</b>	string dengan ukuran bervariasi antara 1 sampai dengan 255 karakter. Contoh VARCHAR(25)
<b>TEXT</b>	String dengan ukuran maksimum 65535 karakter. String yang tersimpan di dalam teks dianggap tidak <i>case sensitive</i> . Untuk kapasitas yang lebih kecil bisa menggunakan TINYTEXT dengan kapasitas maksimal 255 karakter sedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMTEXT ( 16777215 karakter) dan LONGTEXT ( 4294967295 karakter).
<b>BLOB</b>	<i>Binary Large Objects</i> (BLOB) adalah tipe data untuk menyimpan data binary dalam jumlah besar. biasa digunakan untuk menyimpan citra. Untuk penyimpanan data yang lebih kecil bisa menggunakan TINYBLOB (maksimal 255 karakter) Sedangkan untuk kapasitas yang lebih

## SISTEM INVENTORY

---

	besar bisa menggunakan MEDIUMBLOB (maksimal 16777215 karakter) dan LONGBLOB (maksimal 4294967295 karakter)
<b>ENUM</b>	enumerasi atau sebuah list atau daftar. jadi misalnya anda ingin bahwa sebuah nilai terbatas hanya boleh dengan nilai tertentu saja maka anda bisa membuat sebuah daftar. nilai itu hanya bisa terdiri dari a sampai e maka anda bisa membuatnya menjadi Enum (A,B,C,D,E)

### phpMyAdmin

Secara definisi, phpMyAdmin adalah *tool open source* yang ditulis dalam bahasa PHP untuk menangani administrasi my SQL berbasis *World Wide Web*.

Cara membuka phpmyadmin adalah sebagai berikut:

1. Bukalah XAMPP Control Panel dengan cara klik kanan XAMPP Run di administrator
2. setelah itu tekan Start pada baris Apache, tekan Start pada baris my SQL. kemudian tekan admin pada baris MySQL



## SISTEM INVENTORY

---

*Tools* ini digunakan untuk :

a. Membuat database

Misalnya kita ingin membuat basis data dengan nama db\_perpustakaan.

Langkah kerjanya adalah :

- Buka phpMyAdmin, kemudian isikan nama database yaitu db\_perpustakaan kemudian tekan Buat.



b. Membuat Tabel.

Pada dasarnya, data-data dalam basis data disimpan di dalam tabel. untuk merancang tabel, langkah kerjanya adalah:

- Pilih basis data tempat kita membuat tabelnya (db\_perpustakaan).
- Isikan mahasiswa pada name dan dua pada number of cplomns kemudian tekan go di sebelah kanan bawah
- Pada kolom pertama diisi dengan (column : id\_mahasiswa, type:CHAR, length: 6, index: primary).  
Isian yang lain Biarkan saja

### 1.4 LANDASAN TEORI WEB SERVICE

#### SOA (*Service Oriented Architecture*)

SOA adalah sebuah skema yang memungkinkan komunikasi antar sistem dilakukan secara *loosely coupled*, artinya masing-masing pihak tidak memiliki ketergantungan yang tinggi satu sama lain, dengan kata lain SOA sebuah arsitektur Kerangka kerja berbasis standar terbuka yang memungkinkan perusahaan-perusahaan untuk saling mengintegrasikan data satu sama lain dengan sebuah layanan yang diberikan, Sehingga aplikasi A bisa mendapatkan informasi dari aplikasi B begitu juga sebaliknya.

Menurut Rahmi Nur Shofa, et al. ( 2013) SOA sendiri merupakan suatu konsep gaya arsitektural yang memodularisasi informasi menjadi *services*, dengan demikian *web service* merupakan teknologi yang tepat untuk menerapkan konsep SOA, karena *cycle* proses *web service* sama dengan SOA. Adapun gambaran dari Soa dapat dilihat pada gambar 7.1.

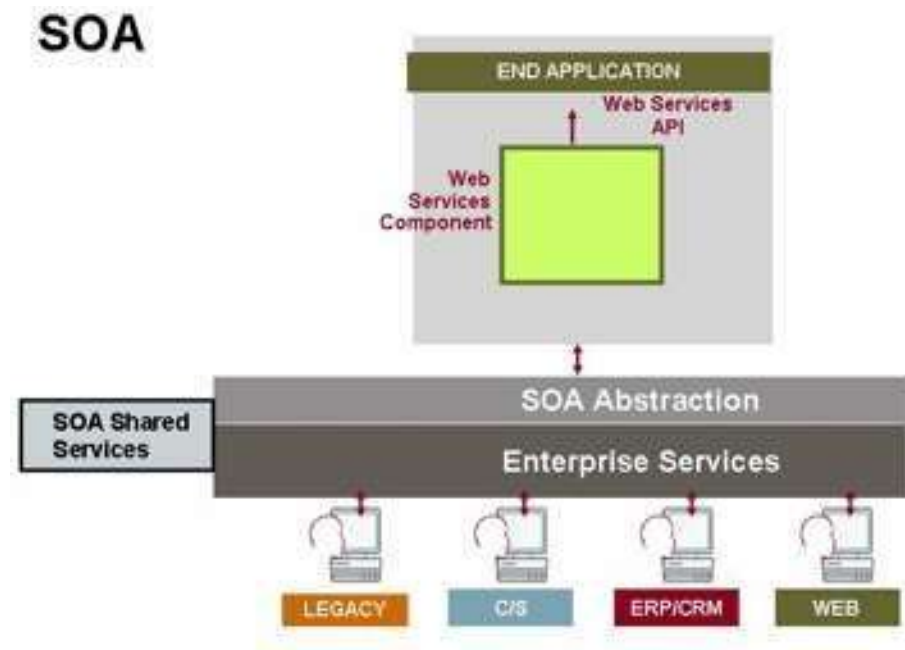
Dari gambar 7.1 arsitektur dapat dilihat SOA menyediakan sebuah layanan yang terdapat pada *web service*, di mana layanan tersebut setelah dibagikan kepada pengguna aplikasi, suatu konsep biasanya dikenal dari karakteristiknya. Dalam SOA terdapat beberapa karakteristik yaitu:

1. *Loose Coupled* adalah *service* yang dipanggil oleh *service* lainnya tanpa program pemanggil tersebut perlu memperhatikan Dimana lokasi servis yang dipanggil berada dan platform apa yang digunakan oleh *service* tersebut.

## SISTEM INVENTORY

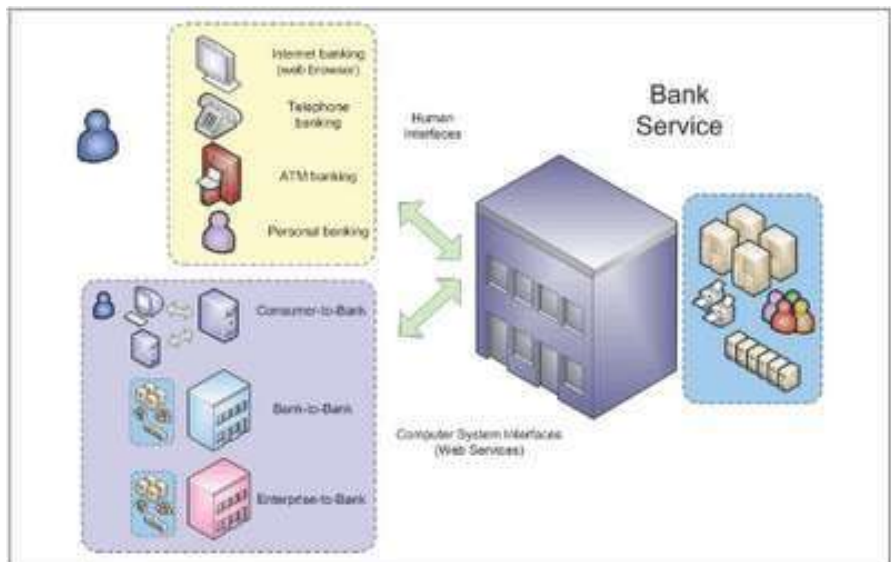
---

*Loose Coupling* sangat penting bagi SOA, karena dengan demikian pemanggilan sebuah *service* oleh *service* lainnya dapat dilakukan pada saat *run-time*. Untuk lebih jelasnya dapat dilihat pada gambar 7.2.



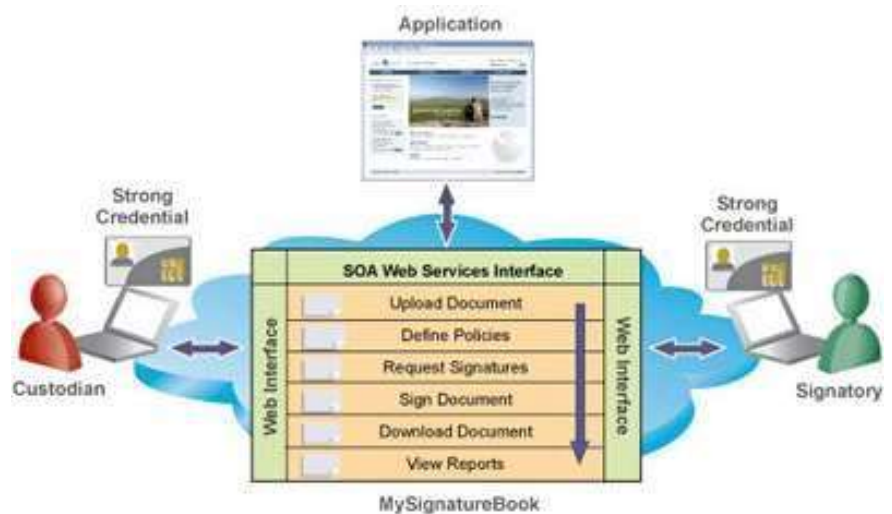
Gambat 7.1 Arsitektur SOA

## SISTEM INVENTORY



Gambar 7.2 Loose Coupled

2. Soa tersusun dari *service interface* dan *service implementation*. *Service interface* menyatakan Bagaimana parameter dari *input* dan *output* serta lokasinya berada, sedangkan *service implementation* adalah *logic* proses dari sebuah *service*. Adapun arsitektur dari *service interface* dan *service implementation* dapat dilihat pada gambar 7.3





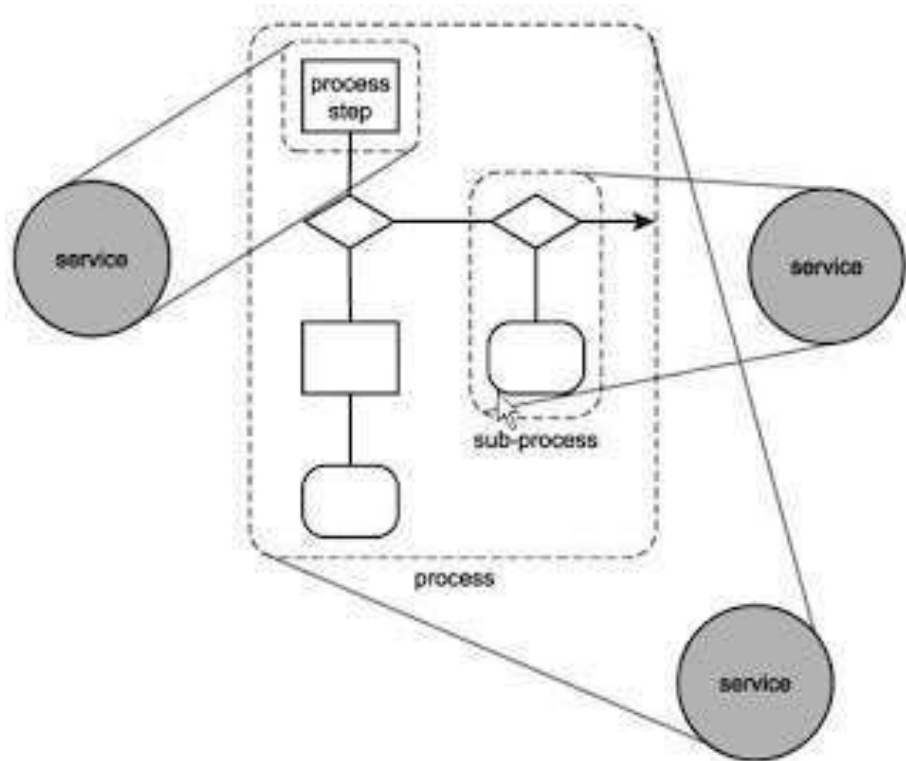
*Gambar 7.3*

Dari gambar 7.3 terlihat Bagaimana skema yang akan dibangun pada sistem informasi pembayaran uang kuliah online, aplikasi yang saling berkomunikasi antar beda *platform*, dari rancangan tersebut perlu adanya sebuah rancangan yang mengubah bisnis *logic* menjadi sebuah layanan yang dibungkus dalam sebuah wadah aplikasi jawabannya adalah *web service*.

3. *Service* harus *business oriented* adalah setiap *service* yang disediakan harus melakukan suatu aktivitas bisnis tertentu. Adapun Bagaimana enkapsulasi rojik dalam servis dapat dilihat pada gambar 7.4

Dari gambar 2.4 dapat dilihat Bagaimana sebuah servis kapsulasi sebuah bisnis *Logic* menjadi *service-service* kecil.

Manfaat utama dari SOA adalah *service oriented* atau berorientasi pada layanan untuk memungkinkan institusi Dalam penggunaannya secara simultan dan mudah untuk pertukaran data antar program dari *vendor* yang berbeda dengan menggunakan sebuah layanan, dan juga antar institusi tidak perlu menggunakan *platform* yang sama dalam pemrograman maupun database sehingga SOA menjadikan sebuah solusi bagi institusi-institusi untuk memberikan inovasi layanan tanpa memerlukan biaya yang tinggi dalam pencapaian inovasi tersebut.

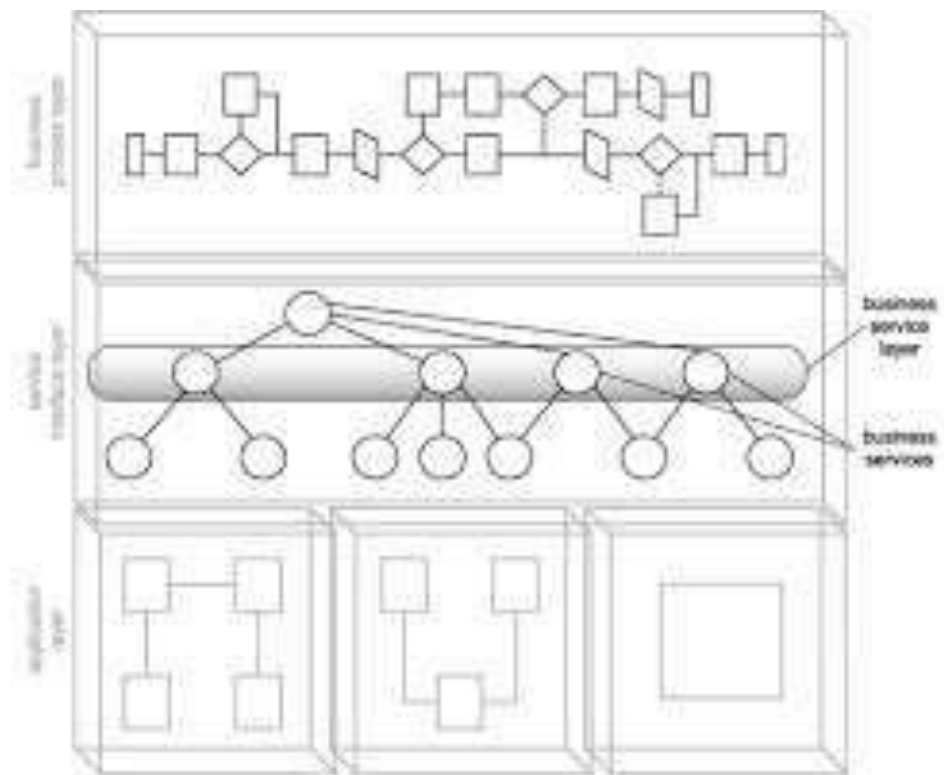


*Gambar 7.4 Enkapsulasi logic Dalam Service*

Riyanarto Sarno (2012) dalam sebuah pengembangan perangkat lunak menggunakan pendekatan konsep SOA berbeda dengan pengembangan tanpa menggunakan konsep SOA, di mana sebuah perangkat lunak yang dikembangkan tidak menggunakan SOA mengakibatkan aplikasi berjalan pada dua *layer* utama yaitu *application layer* dan *business process layer*, sementara itu dalam menggunakan konsep SOA diimplementasikan dalam sebuah *layer* di antara *application layer* dan *business process layer* yaitu *service interface layer*, fungsi ini adalah menggambarkan *application logic* dan *business process* yang ada di *business logic*. Lihat gamabr 7.5

## SISTEM INVENTORY

---



*Gambar 7.5 Layering Dalam SOA*

Gambar 7.5 terlihat bahwa teknologi aplikasi a b dan c terenkapsulasi oleh *service interface layer* ,terlihat bahwa application logik dan bisnis logic dibungkus dalam suatu layanan. dan *service interface layer* itu sendiri memiliki tiga lapisan abstraksi yaitu

1. *Application service layer*

Sebuah layanan yang letaknya berbatasan dengan *application layer* yang bertugas sebagai penyedia fungsi-fungsi untuk pemrosesan data yang nantinya akan dibutuhkan untuk menjalankan fungsi-fungsi yang ada pada proses bisnis.

## SISTEM INVENTORY

---

### 2. Business service layer

Digunakan untuk mengimplementasikan *logic* dan model bisnis menjadi sebuah *service*.

### 3. *Orchestration service layer*

Sebuah layanan yang memberikan abstraksi Bagaimana *service-service* yang berjalan sesuai dengan aturan dan urutannya yang sesuai dengan logika dan bisnis proses yang ada pada perusahaan.

Dengan menggunakan pendekatan metode SOA, langkah dalam pembuatan sistem informasi pembayaran uang kuliah online dapat lebih mudah dipahami, sehingga kita dapat mengetahui bagaimana pengembangan sistem informasi dapat terbentuk, yang akan menjadi sebuah acuan kerja dalam pengembangan sistem informasi secara teknis.

“Arif Firmansyah (2011), dalam dunia perbankan sebenarnya setelah ada arsitektur bisnis yang mirip dengan SOA. Bank besar di dunia telah diimplementasikannya untuk sistem komunikasi data transaksi *point of sale* (POS) untuk pembayaran kartu kredit di tokoh-tokoh dan auto *teller machine* untuk proses transfer pembayaran dan penarikan tunai.”

Dengan demikian, perancangan sebuah teknologi *web service* dalam pembayaran uang kuliah online ini hendaknya dapat memberikan kontribusi yang baik bagi bank, begitu juga dengan pihak kampus

### **Tentang *Web Service***

*Web Service* merupakan suatu komponen *software* yang dirancang untuk mendukung interaksi antar sistem pada suatu jaringan, *web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *web site* untuk menyediakan layanan dalam bentuk informasi kepada sistem lain, sehingga sistem lain dapat berkomunikasi dengan sistem tersebut melalui layanan layanan atau *service* yang disediakan oleh suatu sistem yang menyediakan *web service*.

“Utomo Budiyanto dan Habib Musthofa (2014), *web service* adalah sebuah antarmuka yang terletak di antara kode aplikasi dan pengguna kode tersebut, yang berperan sebagai lapisan abstrak yang memisahkan *platform* dan rincian spesifik bahasa pemrograman tentang bagaimana kode aplikasi seharusnya dipanggil”.

*Web service* merupakan komponen perangkat lunak yang dirancang untuk mendukung interaksi mesin ke mesin melalui sebuah jaringan melalui HTTP. Memungkinkan sebuah aplikasi untuk berkomunikasi jarak jauh melalui *internet* atau *intranet* secara independen menggunakan sebuah *platform* dari sebuah bahasa pemrograman menggunakan aturan pertukaran pesan berbasis XML.

“Menurut Edi Sultanta dan Khabib Musthofa (2013), *Web service* dapat dipahami sebagai *remote procedure call* (RPC) yang mampu memproses fungsi-fungsi yang didefinisikan pada sebuah aplikasi web serta mengekspos sebuah *user interface* melalui *web* dan juga sebagai jembatan penghubung dengan

database tanpa perlu *driver database* dan tidak harus mengetahui jenis DBMS”.

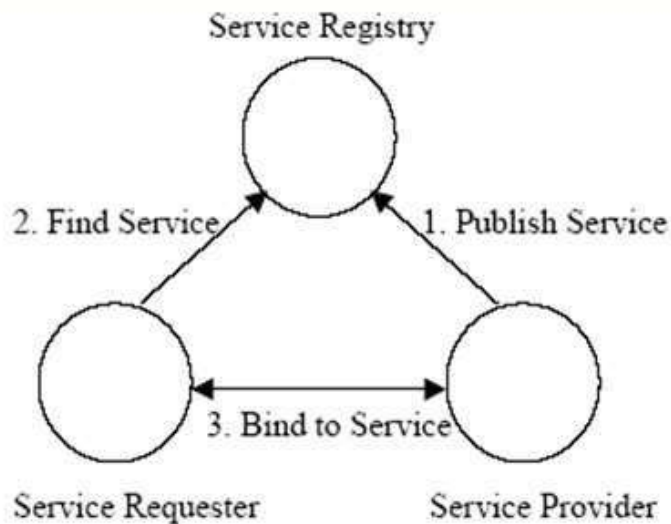
Berbeda dengan aplikasi *web* konvensional, *web service* tidak mengembalikan *interface* yang dapat langsung digunakan oleh *user* berupa layanan yang berbentuk kode HTML, tetapi *web service* mengembalikan layanan tersebut berupa kode XML yang berorientasi pada data yang langsung dapat digunakan oleh *user*.

Keunggulan teknologi *web service* sebenarnya menjadi suatu solusi yang tepat bagi pengembang sebuah sistem informasi, karena penggunaan *internet* yang sudah merasuk ke semua bidang termasuk dunia bisnis dan industri, *web service* menjawab akan tuntutan adanya aplikasi *business to business*, *application to application* yang memungkinkan untuk sebuah layanan *internet* yang dapat digunakan oleh aplikasi lain, karena *web service* dapat menjembatani sebuah aplikasi yang berbeda *framework* sekalipun berbeda dalam pemilihan *database* yang digunakan

Operasi suatu *web service* dapat digambarkan dengan sebuah arsitektur yang memiliki tiga komponen yang mempunyai peranan berbeda-beda. Lihat gambar 7.6

## SISTEM INVENTORY

---



*Gambar 7.6 Arsitektur Web Service*

Keterangan Gambar 7.6

- *Service provider* atau Penyedia Layanan  
Berfungsi sebagai lokasi Sentral yang mendeskripsikan semua layanan atau service yang sudah deregister
- *Service Registry* atau Daftar Layanan  
Meminta layanan yang mencari dan menemukan layanan yang dibutuhkan oleh *service requestor*, serta menggunakan layanan tersebut
- *Service Requestor* atau Peminta Layanan  
Adalah pihak yang membutuhkan sebuah layanan atau service dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia

Secara keseluruhan komponen-komponen dari web service dapat dilihat pada gambar 7.6



*Gambar 7.6 Blok Bangunan Web Service*

*Web Service* secara keseluruhan mempunyai empat komponen seperti pada gambar 6.7 yaitu:

1. Layar 1: protokol *internet* standar yang digunakan sebagai media transportasi adalah http, TCP/IP.
2. Layar 2: *Simple Object Access Protocol* (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk pertukaran data antar layanan.
3. Layar 3: *Web Service Definition Language* (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.
4. Layar 4: *Universal Description Discovery and Integration* (UUDI), yang mana merupakan direktori pusat untuk deskripsi layanan.



## SISTEM INVENTORY

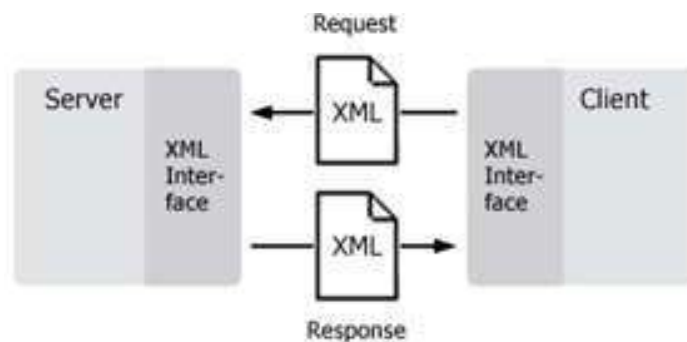
---

“Menurut Riyanto (2013), UDDI merupakan suatu layanan direktori hierarki dan terpusat yang menyediakan layanan untuk mempublikasikan informasi teknikal layanan web.

Dari beberapa penjabaran mengenai *web site* yang telah kita bahas di atas dapat disimpulkan beberapa kelebihan *web service* diantaranya:

- *Web service* mempunyai kelebihan untuk bisa diakses oleh aplikasi yang berjalan pada *platform* yang berbeda-beda.
- *Web service* menggunakan standar dan protokol terbuka pada internet dengan menggunakan HTTP.
- *Web Service* memungkinkan fungsi-fungsi pada banyak perangkat lunak di *internet* untuk dipadukan menjadi *web service* baru.

Disamping itu, web service mempunyai prinsip kerja yaitu mengembalikan hasil servis ke dalam bentuk XML yang berorientasi pada data yang belum bisa dibaca oleh pengguna, sehingga untuk menampilkan data tersebut diperlukan sebuah *interface*. Lihat Gambar 7.8



Request/response scheme (Client → Server → Client)

Gambar 7.8 Pengiriman Data XML

## SISTEM INVENTORY

---

*Application* pada gambar 7.8 berfungsi untuk menampilkan data XML ke *database* bentuk *interface*, sehingga data dalam *interface* tersebut diberikan kepada pengguna dalam bentuk *user interface*.

Untuk merepresentasikan kontrak antara *requestor* dan *provider* atau antara kode *client* dan kode *server*, secara teknis dapat digunakan beberapa standar protokol yang sudah ada yaitu SOAP WSDL. Dengan menggunakan SOAP WSDL, *client* dapat memanfaatkan fungsi-fungsi *public* yang disediakan oleh *server*.

WSDL (*web service description language*) merupakan sebuah dokumen dalam format XML yang isinya menjelaskan informasi detail sebuah *web service*. Di dalam WSDL di jelaskan metode-metode apa saja yang tersedia, parameter apa saja yang diperlukan untuk memanggil sebuah metode, apa hasil atau tipe data yang dikembalikan oleh metode yang dipanggil tersebut.

Dalam sebuah pengembangan dan pembuatan sistem informasi dibutuhkan sebuah perancangan yang baik agar sistem informasi yang dikembangkan dapat berjalan sesuai dengan yang dibutuhkan sehingga sistem tersebut menjadi konsisten, terarah sesuai dengan aturan, terkonsep dan dapat dikembangkan dengan mudah di kemudian harinya.

### 1.5 XML (*Extensible Markup Language*)

Xml merupakan sekumpulan aturan-aturan yang mendefinisikan suatu syntax yang digunakan untuk menjalankan suatu teks atau data dalam sebuah dokumen melalui penggunaan *tag*.

Di dalam *web service*, XML berfungsi sebagai komunikasi antar aplikasi dan integrasi data walaupun antar aplikasi yang berkomunikasi tersebut berbeda *platform*. Di sini juga terletak kelebihan dari XML tersebut, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antar satu dengan yang lain.

Di dalam jurnal implementasi *extensible markup language web service*, XML *web service* memiliki beberapa karakteristik diantaranya:

- Standar dari XML *web service* adalah Standar Industri yang independen.
- XML *web service* tidak menyediakan *interface*, tetapi hanya menyediakan fungsi.
- XML *web service* menggunakan *hypertext Transfer Protocol* (HTTP) sebagai protokol standar.
- XML *web service* menggunakan model *request-response* yang sama seperti pada aplikasi *web*.
- XML *web service* ditempatkan di dalam sebuah *web server*.
- XML *web service* lebih ditujukan untuk kepentingan dari *programmer to programmer*, sehingga penggunaan XML *web service* yang utama bukanlah pengguna akhir tapi

*programmer* yang menggunakan bahasa pemrograman yang berbeda.

- Hasil dari XML *web service* merupakan *plain text* yang berupa berkas XML.
- Fungsi dari sebuah XML *web service* butuh pengolahan lebih lanjut dalam membentuk sebuah *interface* aplikasi.
- Secara umum hampir semua bahasa pemrograman yang dapat terkoneksi ke *internet* akan mampu terkoneksi dengan XML web service, namun tidak semua bahasa pemrograman tersebut mampu memproduksi XML *web service*.

Pada dasarnya, XML tidak jauh berbeda dengan HTML, dimana elemen yang digunakan sama-sama menggunakan pembuka dan penutup, masing-masing dikembangkan untuk tujuan yang berbeda-beda. HTML digunakan sebagai pemberi informasi yang dapat terlihat, sedangkan XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri.

### **SOA (*Simple Object Access Protocol*)**

SOAP merupakan protokol untuk pertukaran informasi dengan desentralisasi dan terdistribusi. SOAP merupakan gabungan antara HTTP dengan XML, karena umumnya menggunakan protokol HTTP sebagai sarana *transport* datanya dan data akan dipertukarkan atau ditulis dalam format XML.

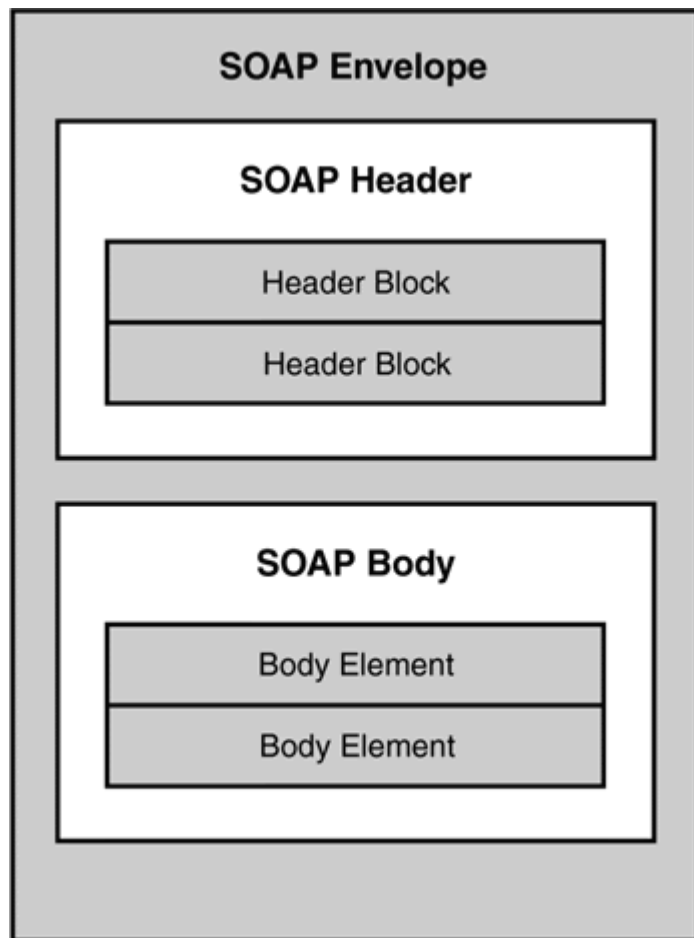
Karena SOAP menggunakan HTTP dan XML maka memungkinkan pihak-pihak yang mempunyai *platform*, seperti sistem operasi dan perangkat lunak yang berbeda dapat saling mempertukarkan datanya. SOAP mengatur bagaimana *request*

dan respon dari suatu *web service* bekerja. (Hartati Deviana, 2013).

“Riyanto,*et al.* (2013) pesan SOAP tersusun dari sebuah amplop yang berisi tubuh pesan dan informasi *header* yang akan dipakai untuk menerapkan pesan tersebut. elemen *envelope*. Dalam amplop tersebut, pesan SOAP yang sah dapat berisi elemen-elemen anak lainnya”. Adapun struktur pesan SOAP sebagai berikut:

- *Envelope Element* ( yang mengidentifikasi dokumen XML sebagai pesan SOAP).
- *Elemen Header* ( berisi informasi *header*).
- *Elemen body* (berisi panggilan dan respon informasi)

Jika digambarkan dalam sebuah diagram maka akan terlihat seperti pada gambar berikut.



*Gambar 2.10 Diagram Struktur SOAP*

### **WSDL (*Web Service Description Language*)**

WSDL merupakan bahasa berbasis XML yang digunakan untuk mendefinisikan *service* dan menggambarkan Bagaimana cara mengakses *web service* tersebut.

Ketika aplikasi *client* meminta *service*, UDDI akan memberikan informasi tentang tata letak dari dokumen WSDL, WSDL berisi sebuah pesan dengan skenario XML dengan skema tersebut pesan yang diminta dari *client* akan diproses dengan menggunakan WSDL, *client* dapat memanfaatkan fungsi-fungsi

publik yang sudah disediakan oleh *server* berupa *method-method* dan parameter-parameter apa saja yang tersedia dalam sebuah *web service*.

“Menurut Danny sasmoko (2013), WSDL adalah dokumen XML yang *machine readable*. Oleh karena itu, dokumen digunakan untuk melakukan otomatisasi proses Pengintegrasian layanan ke aplikasi *requestor*, aplikasi *requestor* tersebut langsung berhubungan dengan *service provider*.

Elemen-elemen yang ada pada WSDL adalah:

- *Message*

Definisi tipe data yang akan dikomunikasikan.

Berikut contoh struktur kode message WSDL:

```
34 <message name="searchRequest">
35   <partname="param" type="tns:TypeDataInput"/></message>
36
37 <message name="searchResponse">
38   <partname="return" type="tns:TypeDataOupptut"/></message>
39
```

Pada contoh *script message Element* WSDL di atas diperlihatkan tersebut adalah sekumpulan argumen yang di dalamnya terdapat sejumlah elemen pesan yang digunakan sebagai *input output* atau *fault message* dalam suatu operasi

- *Operation*

Nama operasi dari sebuah *web site* atau *method* untuk proses *input output*.

## SISTEM INVENTORY

---

Berikut contoh struktur *coding Operation WSDL*:

```
41 <operation name="search">
42   <documentation>Pemba Uang Kuliah</documentation>
43   <input message="tns:searchRequest"/>
44   <output message="tns:searchResponse"/>
45 </operation>
```

Contoh *script Operation element WSDL* di atas diperlihatkan di mana pers merupakan nama *method* yang akan digunakan pada proses *request* data bagi *client* ke *server*, penamaan ini tidak menjadi sebuah ketentuan bagi operasi, kita dapat memberikan nama sesuai dengan kebutuhan kita.

*Documentation* merupakan sebab penamaan atau *header* dari sebuah pengiriman *file*.

*Input* merupakan operasi masukan bagi parameter yang dikirimkan.

*Output* merupakan operasi keluaran atau *response* yang diberikan kepada *server* kepada *client*.

- *Port Type*

Mendeskripsikan sebuah *web service*, operasi-operasi yang telah dijalankan dan pesan-pesan sebuah *web service*.

Berikut contoh struktur kode *port tupe WSDL*:

```
8
9 <portType name="serverportType">
10   <operation name="search">
11     <documentation>Pmebayaran Uang Kuliah</documentation>
12     <input message="tns:searchRequest"/>
13     <output message="tns:serachResponse"/>
14   </operation>
15 </portType>
16
```



Pada contoh *script code port type* elemen WSDL di atas penjelasannya persis dengan penjelasan pada *operation*, namun yang menjadi di tambah adalah penamaan *port type* yang digunakan

- *Binding*

Protokol komunikasi yang digunakan oleh *web service*.

Berikut contoh struktur kode dinding WSDL:

```
17
18 <binding name="serverBinding" type="tns:unitasserverPorttype">
19   <soap:binding style="rpc"
20     transport="http://schemas.xmlsoap.org/soap/http"/>
21
22   <operation name="search">
23     <soap:operation
24       soapAction="urn:server#search" style="rpc"/>
25     <input><soap:body use="encoded" namespace="urn:server"
26       encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
27     </input>
28     <output><soap:body use="encoded" namespace="urn:server"
29       encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
30     </output>
31   </operation>
32 </binding>
```

Pada contoh *script code billing* elemen WSDL di atas adalah menentukan pada *web service request* tentang bagaimana membuat pesan anak untuk protokol tertentu, setiap *port type* dapat memiliki satu atau lebih elemen yang terkait. Untuk *port type* yang telah ditentukan, elemen yang mengikat harus menentukan pesan dan pasangan *transportnya* (SOAP / HTTP / . SOAP / SMTP).

- *Port*

*Port* merupakan definisi dari sebuah alamat jaringan..

Berikut contoh struktur *code port* WSDL:

## SISTEM INVENTORY

---

```
1 <port name="serverPort" binding="tns:serverPort">
2 <soap:address location=
3 "http://localhost/pembayarankuliah/trxinputquiry.php"/>
4 </port>
```

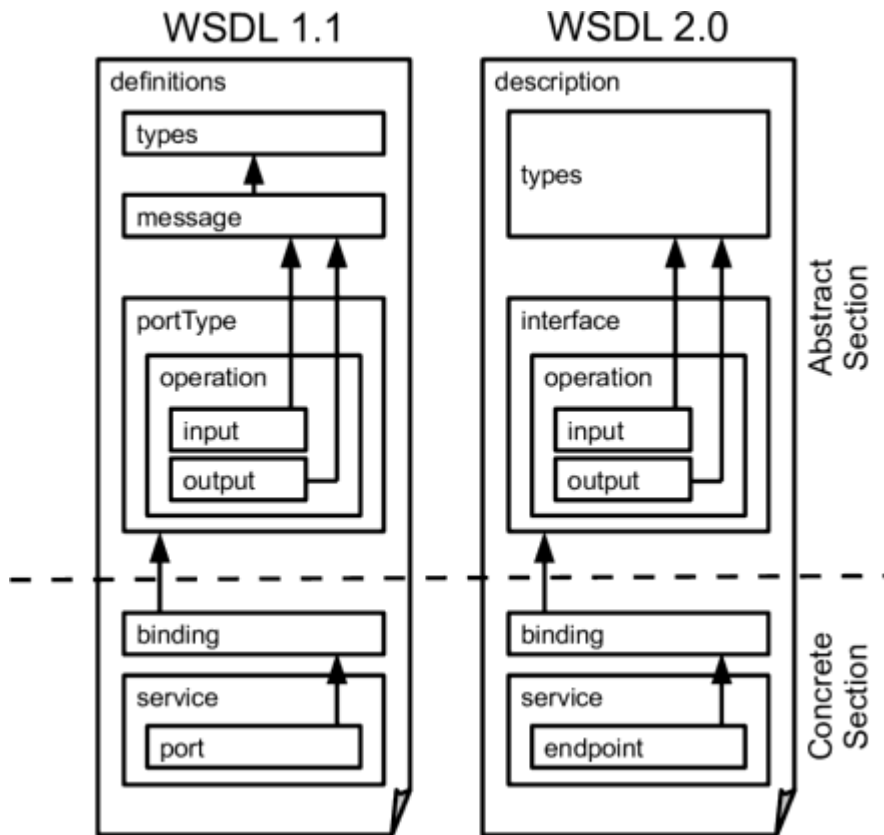
Pada contoh *script code port* elemen WSDL diatas diperlihatkan setelah *binding* ditentukan, maka pesan SOAP akan mengarahkan ke alamat mana, nantinya *client* harus masuk untuk menyisipkan parameter *input*.

- *Service*

Sekumpulan *endpoint* yang saling berhubungan yang menunjukkan sebuah *file / path* mana yang akan ditempatkan pada file WSDL, setiap penamaan *service* harus diberikan penamaan unik.

Berikut contoh struktur *code service* WSDL:

```
1 <service name="server">
2   <port name="serverPort" binding="tns:serverBinding">
3     <soap:address location=
4       "http://localhost/pembayaranuangkuliah/trxnquiry.php"/>
5   </port>
6 </service>
```



Gambar 2.11 Diagram WSDL

### 1.6 NuSOAP

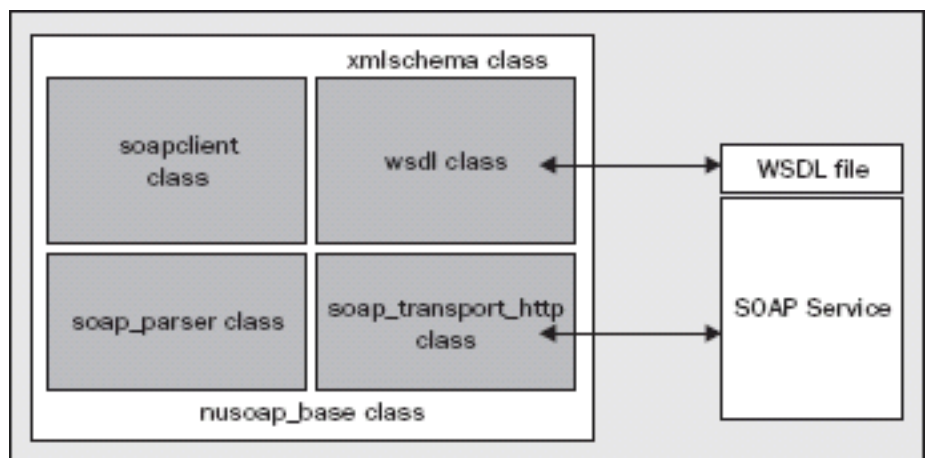
NuSOAP merupakan sebuah *library open source* dibawah lisensi NULGPL yang didistribusikan oleh NuSphere Corporation yang berisikan *class PHP* yang memungkinkan *user* untuk mengirim dan menerima pesan SOAP melalui *class client* yang disebut dengan *class “soapclient”* dan *class server* yang disebut dengan *class “soap server”* dan dengan beberapa *class* pendukung lainnya melalui protocol HTTP.

## SISTEM INVENTORY

---

Class “*soapclient*” memanggil data WSDL dari sebuah *file* PHP untuk menerjemahkan parameter-parameternya sekaligus menyusun *SOAP envelope*, ketika pemanggilan tersebut dieksekusi maka *class* “*soapclient*” menggunakan “*soap\_transport\_http*” untuk mengirim pesan *SOAP request* lalu pesan *SOAP response* yang diterima di parsing dengan menggunakan “*soap parser*”.

Operasi-operasi pengiriman pesan *SOAP* dari *client* melalui *method call* () yang mana *method* tersebut berisikan parameter-parameter dari *client* yang akan dibutuhkan untuk menggali sebuah informasi dari *server*. Berikut bentuk diagram proses *web service* menggunakan NuSOAP. Lihat gambar 2.12



Gambar 2.12 Proses web Service Dengan NuSoap

NuSOAP ini juga adalah SOAP Toolkit untuk PHP yang tidak memerlukan ekstensi PHP. Setelah mengunduh NuSOAP, kita perlu mengekstrak konten file zip ke direktori penyimpanan. Setelah diekstraksi, Anda akan mereferensikan “lib / nusoap.php” di aplikasi SOAP PHP Anda.

## SISTEM INVENTORY

---

Contoh :

```
48
49 <?php
50
51 //require NuSOAP
52 require_once("../lib/nusoap.php");
53
54 //retrieve WSDL
55 $client = new nusoap_client("http://{site_url}/service/v4/soap.php?wsdl", 'wsdl');
56
57
```

### Perbedaan PHP SOAP atau NuSOAP

PHP SOAP tersedia sejak PHP 5.0.1. Jika kita menggunakan PHP4, Anda harus menggunakan NuSOAP.

Kode PHP asli biasanya lebih baik kinerjanya & relatif bebas bug, jadi jika SOAP PHP tersedia, gunakan. Terlebih lagi, NuSOAP tidak memiliki banyak dokumentasi di situs *web* resminya.

Meskipun ada beberapa nuansa untuk disebutkan, NuSOAP juga sangat banyak digunakan, karena :

- Nusoap memiliki beberapa metode yang telah ditentukan bahwa dalam hal menggunakan Sabun Anda harus menulis sendiri.
- Karena bottleneck kinerja SOAP adalah waktu respons server, tidak ada rasa takut untuk menggunakan kelas yang sudah ditentukan Seperti Nusoap.
- Menangani UTF-8 jauh lebih mudah di Nusoap.
- Nusoap menawarkan beberapa fungsi yang baik untuk membuat server SOAP.

### 1.7 UML (*Unified Modelling Language*)

UML merupakan sebuah bahasa spesifikasi standar yang memvisualisasi berdasarkan gambar untuk menspesifikasikan, membangun dan mendokumentasikan dari sebuah pengembangan *software*.

“UML adalah notasi grafis oleh meta model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek(OO)(Flower, M,2004).

Penggunaan UML (*Unified Modeling Language*) merupakan langkah pertama dalam mengembangkan metodologi desain berorientasi objek untuk aplikasi komputer serta sistem basis data, UML ini sebagai notasi standar untuk pemodelan objek dunia nyata. Perancang sistem perangkat lunak dan arsitek diberikan banyak pilihan untuk menyediakan ketekunan objek yang andal, fleksibel dan efisien untuk aplikasi komputer dan sistem basis data. Mereka dapat memilih antara *Object-Oriented*, *Object-Relational hybrids*, *Relational* murni dan solusi khusus berdasarkan format file terbuka atau eksklusif.

UML dianggap sebagai bahasa pemodelan standar industri dengan notasi grafis yang kaya, dan serangkaian diagram dan elemen yang komprehensif.

UML adalah alat untuk menentukan dan memvisualisasikan sistem perangkat lunak. Ini termasuk tipe diagram standar yang menggambarkan dan memetakan aplikasi komputer secara visual atau desain dan struktur sistem basis data. Penggunaan UML sebagai alat untuk mendefinisikan struktur

## SISTEM INVENTORY

---

suatu sistem adalah cara yang sangat berguna untuk mengelola sistem yang besar dan kompleks. Memiliki struktur yang terlihat jelas membuatnya mudah untuk memperkenalkan orang baru ke proyek yang sudah ada. Ini menawarkan cara standar untuk memvisualisasikan cetak biru arsitektur sistem, termasuk elemen-elemen seperti :

- *Activities*
- *Actors*
- *Business*
- *Processes*
- *Database Schemas*
- *(Logical) Components*
- *Programming Language Statements*
- *Reusable Software Components.*

UML menggabungkan teknik dan proses dari pemodelan data (*entity relationship diagrams*), pemodelan bisnis (alur kerja/ *works flow*), pemodelan objek, dan pemodelan komponen. Ini dapat digunakan dengan semua proses, sepanjang siklus hidup pengembangan perangkat lunak, dan di berbagai teknologi implementasi.

### **Arsitektur Perangkat Lunak Dikembangkan Menggunakan UML Memberikan Manfaat Berikut :**

- *Preliminary Analysis Model* / Model Analisis Pendahuluan

Di luar diagram *use case*, pemodelan UML dapat sangat membantu dalam formalisasi model analisis pendahuluan.

Pemodelan proses bisnis dapat didasarkan pada diagram aktivitas dan model pertama aplikasi Anda dapat direalisasikan menggunakan diagram kelas. Persyaratan *Objectteering* / UML menyediakan asisten untuk membantu dalam pembuatan model dari analisis persyaratan dan kamus. Dengan menggunakan diagram alir, Persyaratan *Objectteering* / UML memberi Anda representasi unik dari definisi persyaratan Anda melalui arus informasi yang dipertukarkan antara elemen sistem

- *Component-Based Development* / Pengembangan Berbasis Komponen

Untuk pengembangan sistem bisnis yang kompleks, pengembangan berbasis komponen adalah cara yang sudah terbukti. Dengan *Rational Rose*, pengguna sekarang dapat memodelkan komponen dan antarmuka mereka dengan lebih efektif. Komponen desain yang diperlukan dapat dengan mudah direkayasa ulang untuk menjelajahi antarmuka dan keterkaitan antara komponen lain dalam model, nafsu tarik dan lepas komponen dari sistem file ke diagram komponen. Kemudian Anda dapat merekayasa balik, menggunakan kembali, memvisualisasikan, beradaptasi, memperoleh, dan membuat komponen untuk sistem Anda dengan cepat dan efisien.



## SISTEM INVENTORY

---

- *Multi-Language and Multi-Platform Development / Pengembangan Multi-Bahasa dan Multi-i /*

Proyek perangkat lunak berskala besar sering melibatkan berbagai bahasa. *Rational Rose* menyediakan dukungan multi bahasa sehingga Anda dapat membangun komponen dalam bahasa campuran. Rose mendukung C ++, Visual C ++, Java, *Visual Basic*, Ada, dan dapat menghasilkan *Interface Definition Language* (IDL) untuk aplikasi CORBA, serta *Data Description Language* (DDL) untuk aplikasi database. Rose yang rasional juga *platform* independen, yang berarti tim Anda dapat bekerja di lingkungan Windows dan UNIX

- *Complete Team Support / Dukungan Tim Lengkap*  
*Rational Rose* menyediakan cara mudah untuk mengoordinasikan tim pengembangan yang terdiri dari sepuluh hingga seratus atau lebih. Analis, arsitek, dan *engineers* dapat beroperasi di ruang kerja pribadi yang berisi tampilan individual dari keseluruhan model, dan memodifikasi bagian modelnya sendiri hingga siap untuk digabung dengan model lain menggunakan Integrator Model. Perubahan dibuat tersedia untuk orang lain dengan memeriksanya ke dalam *configuration management and version control* (CMVC). *Rational Rose* terintegrasi dengan alat CMVC utama termasuk *Rational ClearCase* dan *Microsoft SourceSafe*. Rose juga sangat terintegrasi dengan *Microsoft Repository* dan Unisys UREP dan terbuka untuk yang lain

repository.

- *Ease of Use* / Kemudahan Penggunaan

*Rational Rose* adalah alat pemodelan visual pilihan untuk berbagai jenis proyek perangkat lunak. Selain menggunakan UML, Rose mendukung elemen GUI yang sudah dikenal seperti *drag* and *drop* dengan otomatisasi OLE. Ini memungkinkan Anda berintegrasi dengan aplikasi seperti Microsoft Word, dan mendorong Rose dari antarmuka server otomatisasi OLE. Kompatibilitas Rose yang terbukti dengan *platform Windows* telah diuji di lab kegunaan Microsoft.

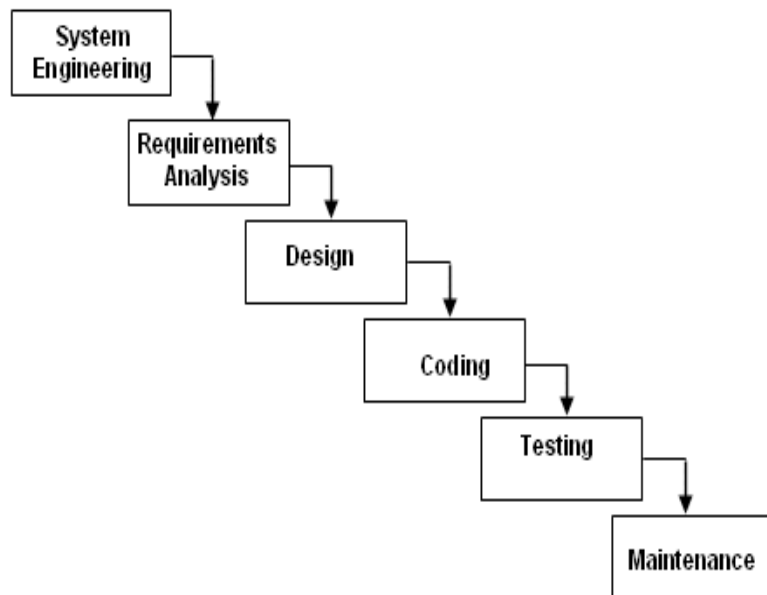
# SISTEM INVENTORY

---

## METODE PENGEMBANGAN SISTEM DAN IMPLEMENTASI

### 1. WATERFALL

Metode yang digunakan dalam pengembangan sistem adalah metode *waterfall*. Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan system ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan.



Gambar 2.1 Metode Pengembangan Sistem Waterfall

## SISTEM INVENTORY

---

Tahap dalam metode waterfall adalah sebagai berikut [4]:

1. Tahap Perencanaan

Merencanakan penambahan fitur pemesanan air secara online, sebisa mungkin ada informasi dari pemilik usaha tentang kebutuhan di aplikasi terkait modul online pemesanan air, dan dijadikan kalimat-kalimat yang menjelaskan apa yang akan dibuat dan apa yang dibutuhkan.

2. Tahap Analisis Sistem

Tahap analisis sistem adalah studi domain masalah untuk merekomendasikan perbaikan dan menspesifikasi persyaratan dan prioritas untuk solusi.

3. Tahap Desain Sistem (Perancangan)

Merupakan spesifikasi atau konstruksi solusi yang teknik dan berbasis komputer untuk persyaratan yang diidentifikasi dalam analisis sistem. Tujuan dari desain sistem ini adalah memenuhi kebutuhan pemakai sistem serta memberikan gambaran yang jelas dan lengkap kepada pemrograman komputer dan ahli-ahli teknik lainnya yang terlibat.

4. Tahap *Coding*

Penulisan kode program atau *coding* merupakan penerjemah desain dalam bahasa yang bisa dikenali oleh *computer*. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh *user*.

## SISTEM INVENTORY

---

### 5. Tahap Implementasi (*Testing*)

Fokus pada perangkat lunak dari segi logik dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

### 6. Tahap Pemeliharaan (*Maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

#### - Kelebihan Metode *Waterfall*

Kelebihan menggunakan metode air terjun (*waterfall*) adalah metode ini memungkinkan untuk departementalisasi dan kontrol. proses pengembangan model fase *one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi dan pemeliharaan.

## SISTEM INVENTORY

---

### - Kekurangan Metode *Waterfall*

Kekurangan menggunakan metode *waterfall* adalah metode ini tidak memungkinkan untuk banyak revisi jika terjadi kesalahan dalam prosesnya. Karena setelah aplikasi ini dalam tahap pengujian, sulit untuk kembali lagi dan mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep sebelumnya.

## 2. *USE CASE MODELLING*

Menurut Windu Gata “*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak”. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. [4]


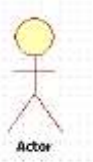


Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

### a. *Use Case Diagram*

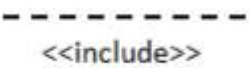

*Use Case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu : [5]

## SISTEM INVENTORY

Tabel 2.1 Simbol *Use Case Diagram*

Simbol	Penjelasan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.</p>
	<p>Aktor atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan pesan terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>



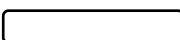
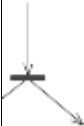
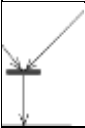
## SISTEM INVENTORY

	<i>include</i> , merupakan didalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

### b. Diagram Aktivitas (*Activity Diagram*)



*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu : [5]

Tabel 2.2 Simbol *Activity Diagram*

Simbol	Penjelasan
	Start Point, diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	End Point, akhir aktivitas
	Activities, menggambarkan suatu proses/kegiatan bisnis.
	Join/percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Split (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.







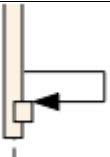
## SISTEM INVENTORY

	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan <i>true</i> atau <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

### c. Diagram Urutan (*Sequence Diagram*)



*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu : [5]

Tabel 2.3 Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun baris data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form <i>entry</i> dan form cetak
	<i>Control Class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar class
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

## SISTEM INVENTORY

---

	ctivation, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitasi sebuah operasi.
	eline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

### d. Diagram Kelas (*Class Diagram*)

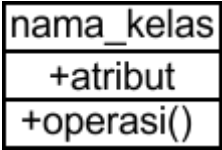
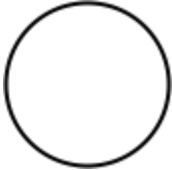


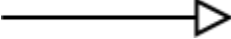
Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.


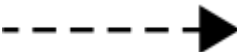
*Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

*Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Assosiations*, *Generalitiation*, dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*. [5]

## SISTEM INVENTORY

---


No	Simbol	Keterangan
1		Simbol ini adalah sibol untuk sebuah kelas pada struktur sistem. penulisan disana tidak diperbolehkan menggunakan spasi. simbol ini memiliki 3 susunan, yaitu kotak pertama adalah nama kelas, kedua atribut dan terakhir operasi.
2		Lingkaran ini adalah simbol untuk interface atau dalam bahasa indonesianya antar muka. konsep yang digunakan pun sama dengan pemrogramman berorientasi object (OOP).
3		Simbol ini sering disebut dengan simbol Association atau dalam bahasa indonesianya yaitu asosiasi. Garis ini adalah garis yang digunakan untuk menghubungkan atau merelasikan kelas satu dengan kelas yang lainnya dengan makna umum.
4		Nama dari sibol ini adalah indirected association atau dalam bahasa indonesianya adalah asosiasi berarah. Simbol ini merupakan simbol relasi antar kelas seperti yang diatas, namun yang membedakan pada relasi ini adalah cara penggunaannya. Simbol ini digunakan jika kelas yang satu digunakan oleh kelas yang lainnya.
5		Simbol ini bernama Generalisasi. Generalisasi digunakan untuk menghubungkan antar kelas dengan arti umum-khusus. Jadi jika ada kelas bermakna umum dan kelas bermakna

		<p>husus dapat menggunakan simbol ini.</p>
6		<p>Nama dari simbol ini adalah Aggregation atau dalam bahasa indonesia nya Agregasi. Simbol ini adalah simbol yang menghubungkan antar kelas dengan makna untuk semua bagian. Jadi relasi ini digunakan jika kelas yang satu adalah semua bagian dari kelas yang lainnya.</p>
7		<p>Nama dari simbol ini adalah Dependency atau dalam bahasa indonesia nya ketergantungan. Kadangkala sebuah class menggunakan class yang lain. Umumnya penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain. Sebuah dependency dilambangkan sebagai sebuah panah bertitik-titik.</p>



## e. Deployment Diagram

*Deployment Diagram* digunakan untuk menggambarkan detail bagaimana komponen disusun di infrastruktur sistem. [5]

Tabel 2.4 Simbol *Deployment Diagram*




Simbol	Terangn
	<p>la <i>deployment diagram</i>, komponen-komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.</p>

## SISTEM INVENTORY



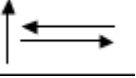
	<p><i>node</i> menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk <i>node</i> digambarkan sebagai sebuah kubus 3 dimensi.</p>
	<p>Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara element-element <i>hardware</i>.</p>

### f. Statechart Diagram

*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimulus yang diterima. Pada umumnya statechart diagram menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). [6]

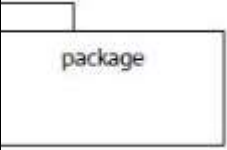
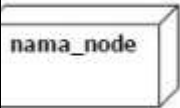
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
		<i>Initial Node</i>	Bagaimana objek dibentuk

## SISTEM INVENTORY

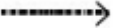

			atau diawali.
4		<i>Activity Final Node</i>	gaimana objek dibentuk dan diakhiri.
5		<i>Decision</i>	gunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
		<i>Flow Connector</i>	gunakan untuk menghubungkan satu simbol dengan simbol lainnya.

### g. Component Diagram

*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. [6]

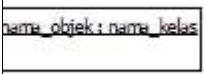

O	GAMBAR	NAMA	KETERANGAN
		<i>Package</i>	Package merupakan sebuah bungkus dari satu atau lebih komponen.
		Komponen	komponen sistem.

## SISTEM INVENTORY

		bergantungan/ <i>Dependency</i>	bergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
		<i>Link</i>	lasi antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai

### h. Collaboration Diagram

*Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan kepada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor 1. *Message* dari level yang sama memiliki prefiks yang sama. [6]

O	GAMBAR	NAMA	KETERANGAN
1		Objek	objek yang melakukan interaksi pesan
2		<i>Link</i>	lasi antar objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri.

### 3. ANALISIS DAN PERANCANGAN

#### 3.1 Analisis

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Analisis adalah langkah awal untuk pengembangan aplikasi, karena perancangan aplikasi tidak akan berjalan dengan baik tanpa adanya analisa terhadap aplikasi yang akan digunakan.

Tahap analisis adalah tahap pemahaman terhadap suatu aplikasi yang telah dibuat. Dalam tahap ini bertujuan untuk mengetahui sistem dari aplikasi tersebut, proses-proses yang terlibat dalam aplikasi serta hubungan antar proses. Analisis juga dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan tujuan serta maksud untuk mengidentifikasi dan mengevaluasi masalah-masalah, kesempatan-kesempatan, hambatan-hambatan yang terjadi serta kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan *recovery* atau perbaikan.

##### 3.1.1 Analisis Sistem Berjalan (*Current System*)

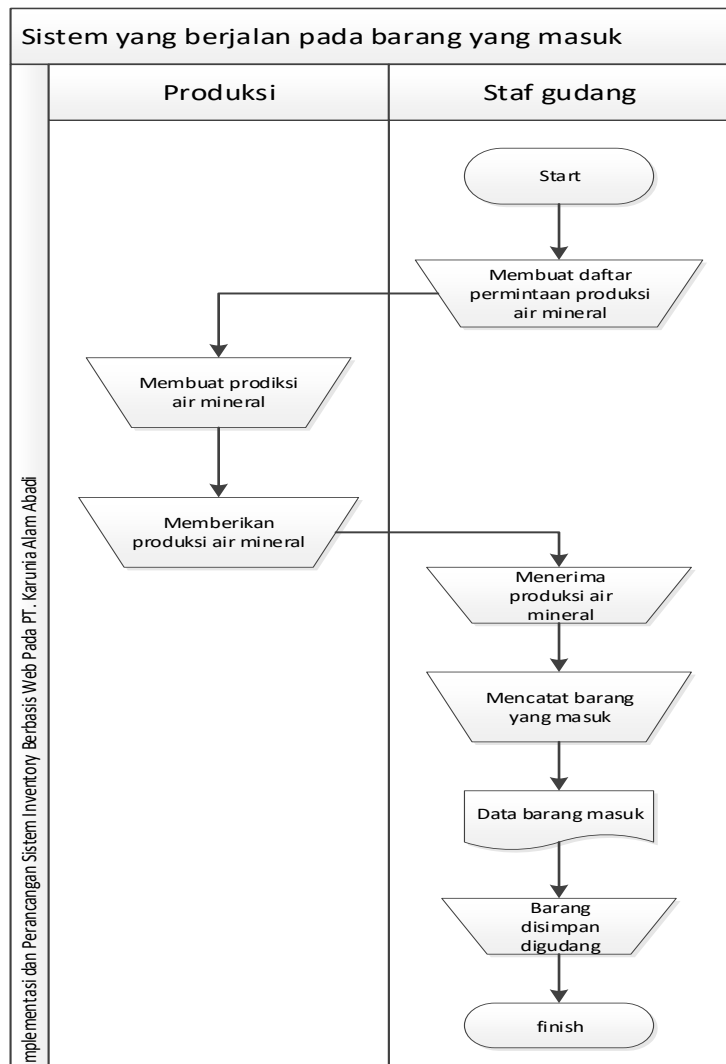
Analisis sistem yang sedang berjalan merupakan sebuah analisis tahap awal dalam penentuan dan perancangan sistem. Di dalam analisis ini terdapat dua metode yang digunakan, yaitu : Analisis Prosedur (*Flowmap*) dan Analisis Dokumen. Dengan demikian,



aplikasi yang akan dibangun akan benar-benar sesuai dengan prosedur serta sistem kerja yang dibutuhkan.

### 3.1.1.1 Analisis Prosedur (*Flowmap*)

#### 3.1.1.1.1 Analisis Sistem Yang Berjalan Pada Barang Yang Masuk



*Gambar 3.1 Flowmap Yang Sedang Berjalan Pada  
Barang Yang Masuk*

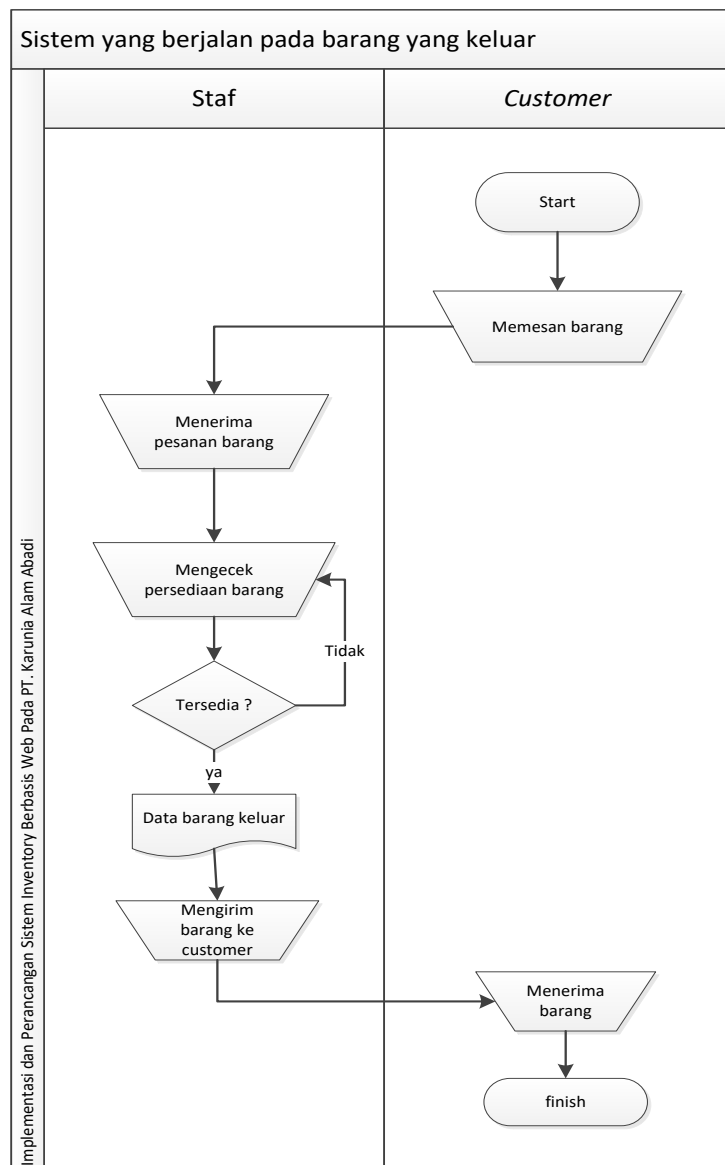
## SISTEM INVENTORY

---

Keterangan :

1. Dimulai dari Staf Gudang membuat daftar permintaan produksi air mineral. Daftar permintaan yang dimaksud disini adalah jumlah air mineral yang dibutuhkan dalam gudang. Daftar permintaan itu kemudian diberikan ke bagian produksi.
2. Lalu, Produksi membuat air mineral sesuai dengan daftar permintaan produksi.
3. Setelah proses produksi selesai, bagian produksi memberikan air mineral kepada staf.
4. Staf menerima hasil produksi air mineral, lalu staf mencatat barang yang masuk dalam buku. Buku yang dipakai dalam mencatat, hanya menggunakan satu buku sehingga besar kemungkinan buku yang digunakan hilang.
5. Kemudian, barang disimpan di dalam gudang.

## 3.1.1.1.2 Analisis Sistem Yang Berjalan Pada Barang Yang Keluar



Gambar 3.2 Flowmap Yang Sedang Berjalan

### *Pada Barang Yang Keluar*

Keterangan :

1. Dimulai dari *customer* memesan barang menggunakan via telpon.
2. Staf menerima pesanan dari *customer*.
3. Staf mengecek persediaan barang yang ada digudang dengan cara melihat buku barang yang masuk, satu persatu sehingga membutuhkan waktu yang lama dalam mengecek persediaan.
4. Jika tersedia, maka staf memberitahukan ke *customer* mengenai pesanannya, dan mencatat transaksi barang keluar di dalam buku. Sama seperti sebelumnya, buku yang digunakan dalam mencatat transaksi barang keluar juga menggunakan satu buku, sehingga besar kemungkinan buku akan hilang dan data dalam buku itu akan hilang juga.
5. Staf mengirim barang ke *customer*.
6. *Customer* menerima barang sesuai pesanan.

#### **3.1.1.1.3 Analisis Dokumen Yang Digunakan**

Dari hasil analisis yang dilakukan, dokumen yang digunakan oleh PT.Karunia Alam Abadi diantaranya

## SISTEM INVENTORY

---

adalah dokumen barang yang masuk dan barang yang keluar. Untuk mempermudah dokumentasi suatu barang (produk) maka dibutuhkan suatu sistem yang dapat mempermudah dalam proses pencarian data dan pemesanannya. Adapun dokumen yang dimaksud adalah sebagai berikut :

Tabel 3.1 Dokumen Barang Yang Masuk

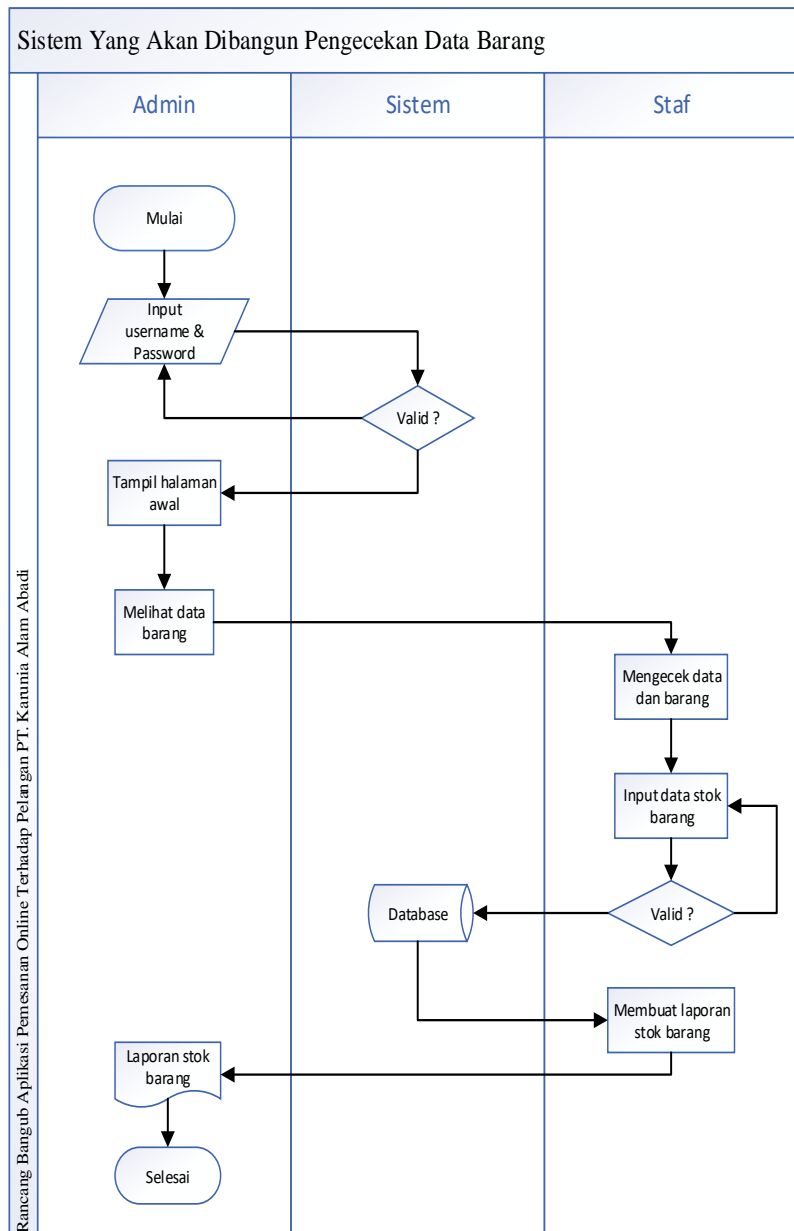
<b>Dibuat oleh</b>	Staf
<b>Dibuat untuk</b>	Staf sebagai bukti persediaan barang yang ada digudang
<b>Isi</b>	Berupa list / daftar barang yang masuk dari produsen
<b>Frekuensi</b>	Dibuat sesuai barang yang masuk
<b>Tujuan</b>	Menyediakan barang untuk <i>customer</i>

Tabel 3.2 Dokumen Barang Yang Keluar

<b>Dibuat oleh</b>	Staf
<b>Dibuat untuk</b>	Staf sebagai bukti adanya transaksi penjualan ke <i>customer</i>
<b>Isi</b>	Laporan data barang yang dibeli oleh <i>customer</i>
<b>Frekuensi</b>	Dibuat sesuai data barang yang dibeli <i>customer</i>
<b>Tujuan</b>	Menyediakan laporan data barang yang dibeli oleh <i>customer</i>

# SISTEM INVENTORY

## 3.1.2 Analisis Sistem Yang Akan Dibangun Pengecekan Data Persediaan Barang

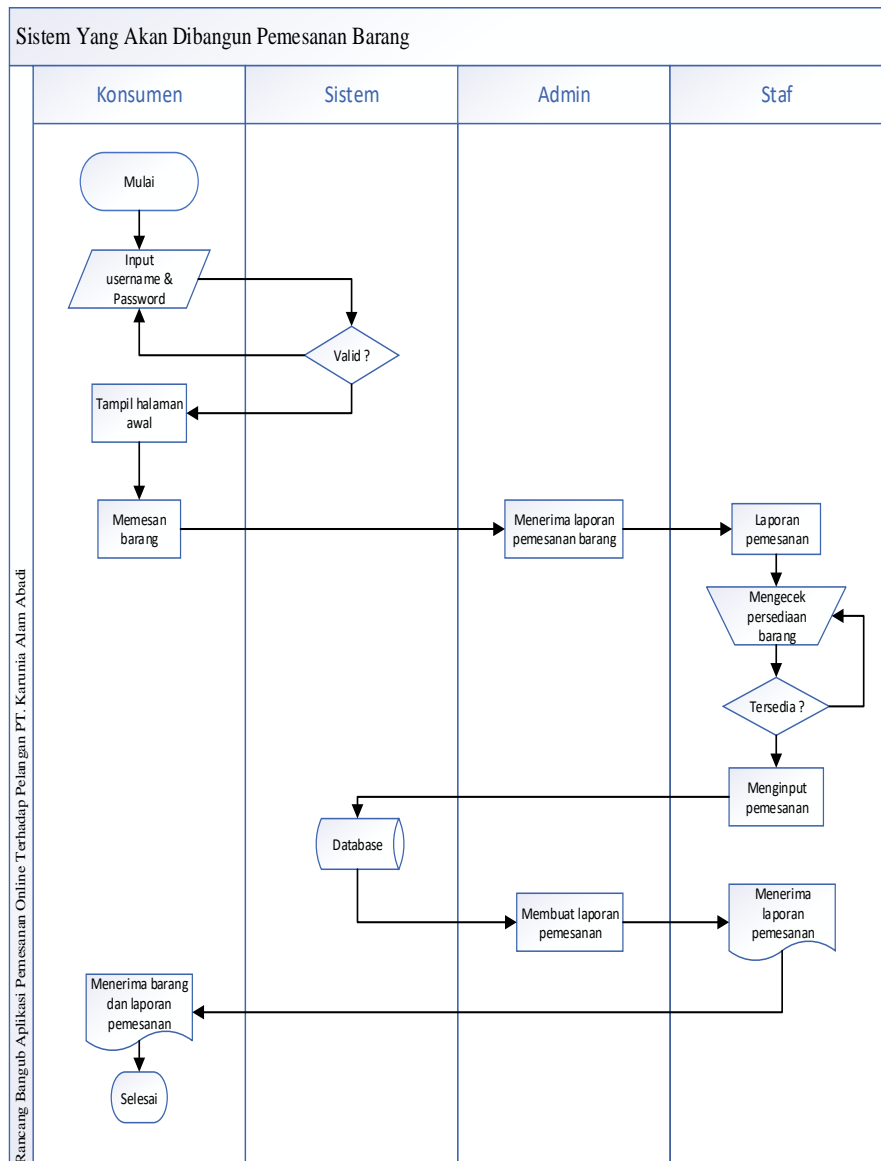


*Gambar 3.3 Flowmap Sistem Yang Akan Dibangun  
Pengecekan Data Barang*

Keterangan :

1. Admin membuka *website* Sistem *Inventory*.
2. Admin dan memasukkan *username* dan *password*.
3. Jika benar, maka masuk ke sistem tampil halaman awal. Jika tidak, maka memasukkan ulang *username* dan *password*
4. Kemudian, Admin melihat data barang. Selanjutnya, maka akan masuk di tampilan mengecek data dan barang yang dilakukan oleh staf.
5. Staf menginput data stok barang air mineral.
6. Jika benar, maka akan tersimpan di dalam database. Jika tidak, maka kembali di input data stok barang.
7. Staf membuat laporan stok barang.
8. Admin menerima laporan stok barang.

## 3.1.3 Analisis Sistem Yang Akan Dibangun Pemesanan barang



Gambar 3.4 Flowmap Sistem Yang Akan Dibangun



### *Pemesanan Barang*

Keterangan :

1. Konsumen membuka *website* Sistem *Inventory*.
2. Konsumen memasukkan *username* dan *password*.
3. Jika benar, maka masuk ke sistem tampil halaman awal. Jika tidak, maka memasukkan ulang *username* dan *password*
4. Kemudian, Konsumen memesan barang.
5. Admin menerima pemesanan barang. Selanjutnya akan diteruskan ke staf.
6. Kemudian staf mengecek persediaan barang. Jika tersedia, staf menginput pemesanan barang yang disimpan di database, jika tidak maka akan kembali di mengecek persediaan barang.
7. Admin membuat laporan pemesanan barang dan diberikan ke staf.
8. Staf mengantarkan barang dan laporan pemesanan barang ke konsumen.

#### **3.1.2.1 Kebutuhan Fungsional**

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan fungsional juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan oleh sistem.

## SISTEM INVENTORY

---

Adapun kebutuhan fungsional yang akan dibuat yaitu:

1. *Login* admin dan staf.
2. Kelola data admin
3. Kelola data staf
4. Kelola data stok barang
5. Kelola laporan stok barang

Setiap proses memiliki representasi masing-masing pada sebuah tabel atau data yang terdapat pada *database* yang telah dirancang sebelumnya. Dan setiap proses berhubungan langsung dengan entitas atau *user*.

### 3.1.2.2 Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, analisis pengguna/*user*.

Adapun kebutuhan non fungsional yang akan digunakan adalah sebagai berikut:

#### A. Kebutuhan Perangkat Keras

Pembuatan aplikasi ini menggunakan perangkat sebagai berikut :

No.	Jenis		Keterangan
1	Processor	:	Intel® core™i3
2	Memory	:	4 GB
3	Monitor	:	LCD 14,1 Inchi
4	Mouse dan keyboard	:	Standard

Tabel 3.3 Deskripsi Perangkat Keras

### B. Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan adalah sebagai berikut :

Tabel 3.4 Deskripsi Perangkat Lunak

OS	Windows 10, dll.
System Operasi	Windows 10, dll.
Bahasa Pemrograman	(Code Igniter)
Software Pendukung	Blime Text 3, Notepad++, dll
Browser	Google Chrome, Mozilla, dll

### C. Analisis Pengguna/User

Aplikasi yang akan dibuat ini digunakan dalam Persediaan Usaha Air Mineral PT.Karunia Alam Abadi antara lain melibatkan Admin dan Staf.

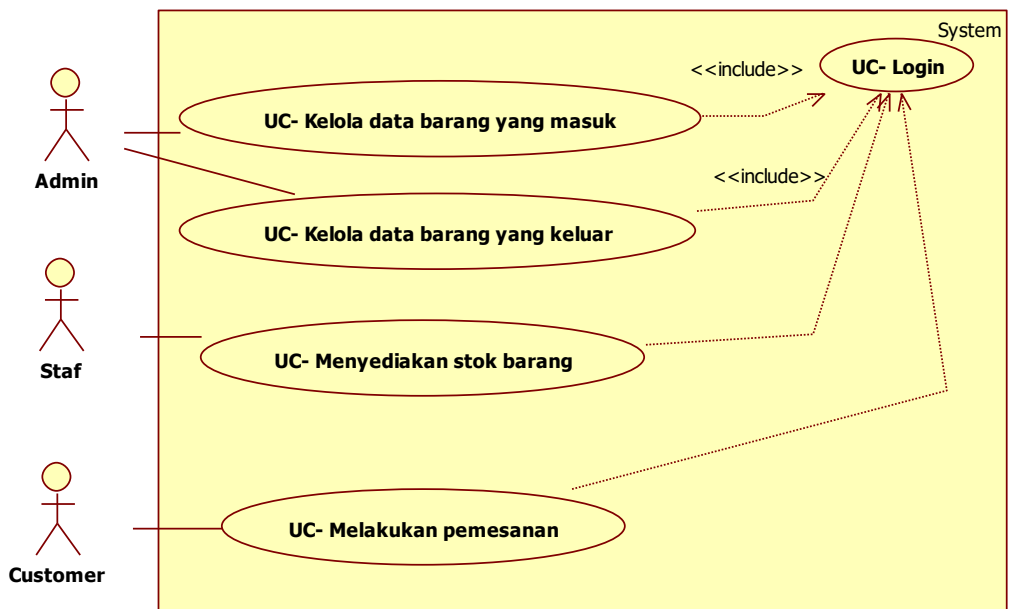
## 3.2 Perancangan

Pada kasus pembuatan *website* ini mekanisme yang telah dirancang yaitu: Admin hanya bisa mengelola data *user*. Staf disini memiliki hak akses mengelola data barang, mengelola pemesanan, melihat laporan penjualan, dan mencetak faktur pemesanan. Sedangkan *customer* hanya memiliki hak akses melakukan pembelian.

### 3.2.1 Use Case Diagram

*Use case* adalah metode berbasis teks untuk menggambarkan dan mendokumentasikan proses yang kompleks. *Use case* menambahkan detail untuk kebutuhan yang telah dituliskan pada definisi sistem kebutuhan. *Use case* dikembangkan oleh analisis sistem bersama-sama dengan pengguna. Pada tahapan selanjutnya, berdasarkan *use case* ini, analis menyusun model data dan model proses.

Diagram *Use Case* dari aplikasi ini adalah sebagai berikut :



Gambar 3.4 Use Case Diagram

### 3.2.1.1 Definisi *Use Case*

Berikut ini merupakan tabel definisi *use case* yang terlibat:

Tabel 3.5 Definisi Aktor

Admin	Penjelasan
Admin	melakukan pengelolaan data barang masuk dan data barang yang keluar.
Staf	menyediakan stok barang , mengelola pesanan
Customer	melakukan pemesanan barang

Tabel 3.6 Definisi *Use Case*

Admin	Penjelasan
Login	melakukan Proses <i>Login</i>
Admin Data Barang yang masuk	melakukan pengelolaan terhadap data barang yang masuk
Admin Data Barang yang keluar	melakukan pengelolaan terhadap data barang yang keluar
Customer	melakukan pemesanan barang

### 3.2.1.2 Skenario *Use Case*

Tabel 3.7 Skenario *Use Case Login*

<b>Identifikasi</b>	
Aktor	Admin, Staf, dan Customer
Media	Internet
Use Case	melakukan <i>login</i> untuk masuk halaman utama
Algoritma	<i>login</i> dengan memasukkan <i>Username</i> dan <i>Password</i>
<b>Detail Skenario</b>	
Keadaan Awal	Menampilkan <i>form login</i>
<b>Aksi Aktor</b>	
<b>Aksi Sistem</b>	
1. Aktor meminta fasilitas <i>login</i> kepada sistem	2. Menampilkan <i>form login</i>
3. Memasukkan <i>Username</i> dan <i>Password</i>	4. Melakukan proses validasi jika salah keduanya, maka akan muncul pesan " <i>Username</i> dan <i>Password</i> yang anda masukkan tidak cocok".
	5. Jika masih salah kembali ke <i>form login</i>

## SISTEM INVENTORY

---

	6. Jika benar maka akan ditampilkan <i>form</i> halaman utama.
Kondisi Akhir	Menampilkan <i>form</i> halaman utama

Tabel 3.8 Skenario *Use Case* Pemesanan

<b>Identifikasi</b>	
Uraian	U-02
Modul	Modul melakukan pemesanan
Fungsi	Modul melakukan pemesanan barang
Prosedur	Modul melakukan pemesanan barang melalui system yang tersedia
Peran	Customer
<b>Skenario</b>	
Kondisi Awal	Modul menampilkan <i>form</i> pemesanan
<b>Aksi Aktor</b>	
<b>Aksi Sistem</b>	
1. Pilih menu pemesanan	2. Menampilkan <i>form</i> pemesanan
3. Customer memilih fasilitas pemesanan yaitu : pesan, edit, tambah, hapus, dan simpan pemesanan barang.	4. Sistem menampilkan halaman yang dipanggil beserta fasilitas yang ada.
5. Customer memilih fasilitas pemesanan barang	6. Sistem memunculkan <i>datagridview</i> sesuai dengan yang telah diinputkan Customer.
	7. Jika benar maka akan ditampilkan <i>form</i> halaman utama.
Kondisi Akhir	Menampilkan <i>form</i> halaman utama

Tabel 3.9 Skenario *Use Case* Kelola Data Barang Yang Masuk

<b>Identifikasi</b>	
Uraian	U-03
Modul	Modul Kelola Data Barang Yang Masuk
Fungsi	Modul melakukan pengelolaan data barang yang masuk
Prosedur	Modul melakukan pengelolaan data barang yang masuk
Peran	Admin
<b>Skenario</b>	
Kondisi Awal	Modul menampilkan <i>form</i> kelola Staf
<b>Aksi Aktor</b>	
<b>Aksi Sistem</b>	
Memulai Aplikasi	Menampilkan <i>form</i> kelola data barang yang masuk

## SISTEM INVENTORY

1. Pilih menu kelola data barang yang masuk	2. Menampilkan halaman kelola data barang yang masuk
3. Aktor memilih fasilitas pengelolaan barang yang masuk yaitu : tambah barang, <i>edit</i> barang, hapus barang, dan simpan barang	4. Sistem menampilkan hasil di halaman yang diminta sesuai dengan fasilitas yang dilakukan.
5. Aktor memilih fasilitas tambah barang	6. Sistem memunculkan <i>datagridview</i> sesuai yang diinputkan aktor.

Tabel 3.10 Scenario *Use Case* Kelola Data Barang Yang Keluar

<b>Identifikasi</b>	
Id	U-04
Nama	Kelola Data Barang Yang Keluar
Fungsi	Lakukan pengelolaan data barang yang keluar
Deskripsi	Lakukan pengelolaan data barang yang keluar
Aktor	Admin
<b>Scenario</b>	
Kondisi Awal	Menampilkan <i>form</i> kelola admin
Aksi Aktor	Aksi Sistem
Memulai Aplikasi	Menampilkan <i>form</i> kelola data barang yang keluar
1. Pilih menu kelola data barang yang keluar	2. Menampilkan halaman kelola data barang yang keluar
3. Aktor memilih fasilitas pengelolaan barang yang keluar yaitu : <i>edit</i> barang, hapus barang.	4. Sistem menampilkan hasil ke halaman yang diminta sesuai dengan fasilitas yang dilakukan.
5. Aktor memilih verifikasi	6. Sistem memunculkan <i>datagridview</i> sesuai yang diinputkan aktor.

### 3.2.2 Class Diagram

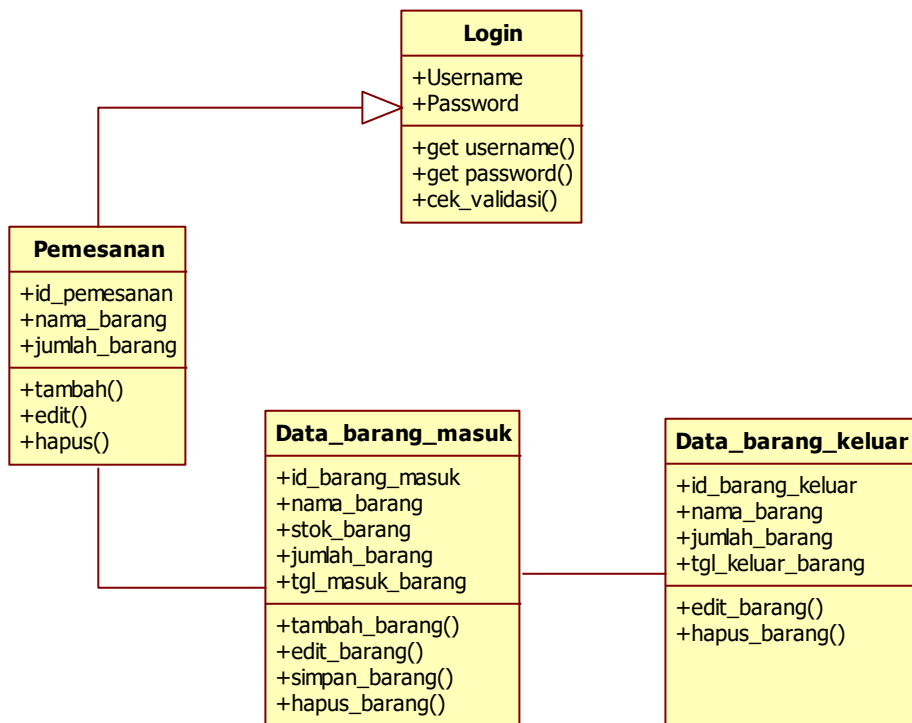
*Class Diagram* menggambarkan *class* dan hubungan antar *class* didalam sistem. *Class* digambarkan dengan sebuah kotak dibagi menjadi tiga bagian. Bagian paling atas diisikan nama *class*,

## SISTEM INVENTORY

---

bagian tengah diisikan variabel yang dimiliki *class*, dan bagian bawah diisikan method-method dari *class*.

Dalam proses pembuatan aplikasi ini *class diagram* yang akan digunakan adalah sebagai berikut :



Gambar 3.5 Class Diagram

### 3.2.3 Sequence Diagram

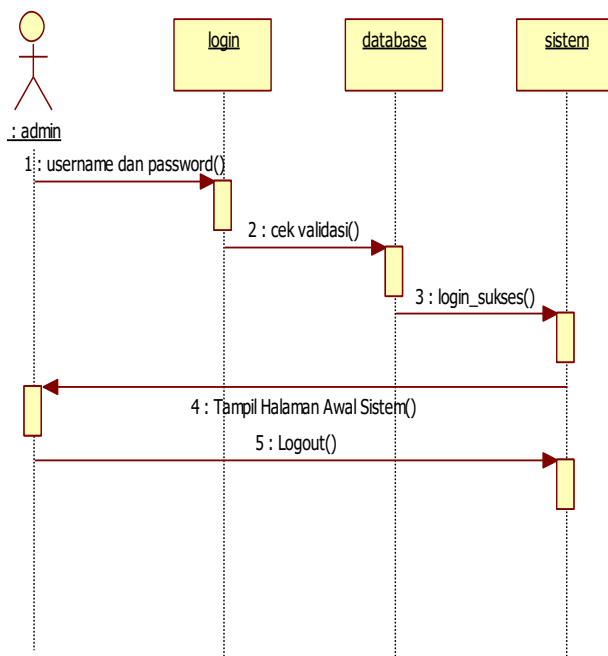
*Sequence Diagram* adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Objek-objek tersebut kemudia diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya



ditaruh di paling kiri dari diagram, pada diagram ini, dimensi vertikal merepresentasikan waktu.

### 3.2.3.1 Sequence Diagram Login

Gambar 3.6 Sequence Diagram

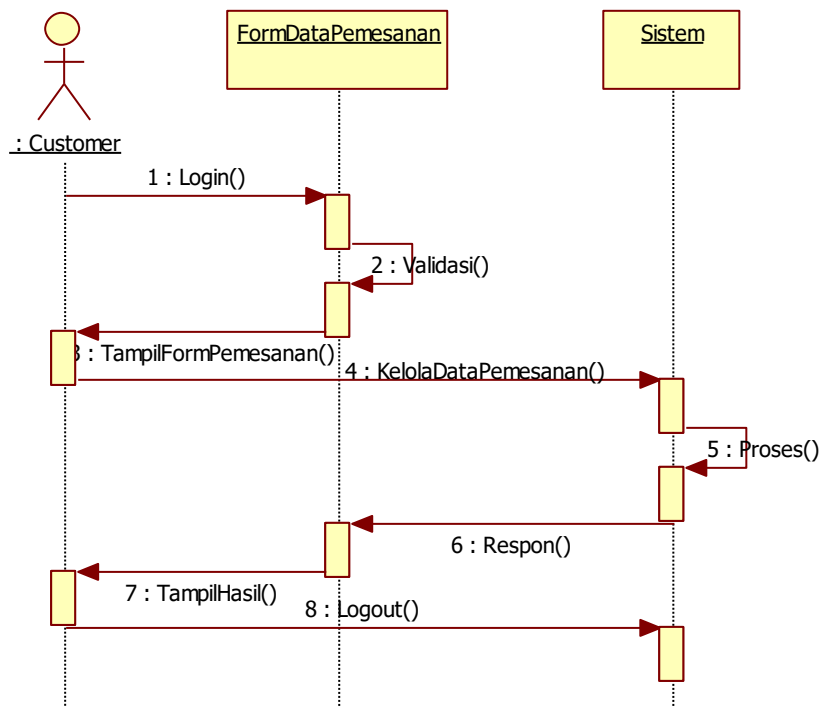


### Login

Sequence diagram ini menjelaskan proses login. Admin mulai memasukkan *username* dan *password*. Kemudian *database* akan mengecek validasi, jika *username* dan *password* benar maka akan masuk ke

sistem dan tampil halaman awal sistem. Jika *username* dan *password* salah maka kembali memasukkan *username* dan *password*.

### 3.2.3.2 Sequence Diagram Pemesanan

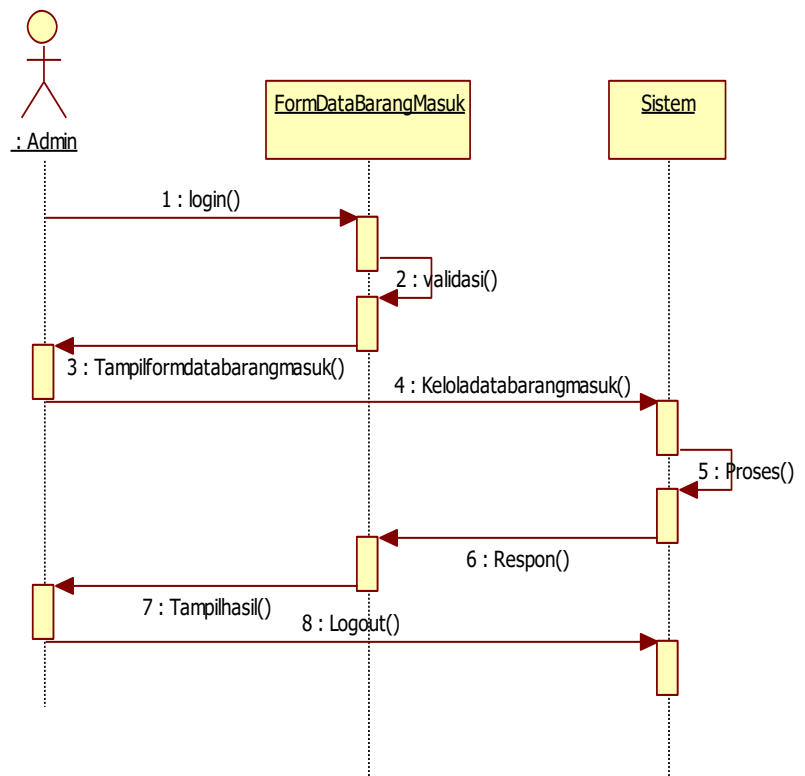


Gambar 3.7 Sequence Diagram Pemesanan

*Sequence diagram* ini menjelaskan proses pemesanan, customer melakukan *login* terlebih dahulu. Selanjutnya akan masuk *form* pemesanan. Untuk memulai kelola ini, masuk ke menu pemesanan, kemudian akan tampil *form* pemesanan. Setelah itu,

melakukan pemesanan. Kemudian sistem akan memproses. Setelah itu akan muncul tampilan sesuai dengan instruksi yang diberikan.

### 3.2.3.3 Sequence Diagram Kelola Data Barang Masuk

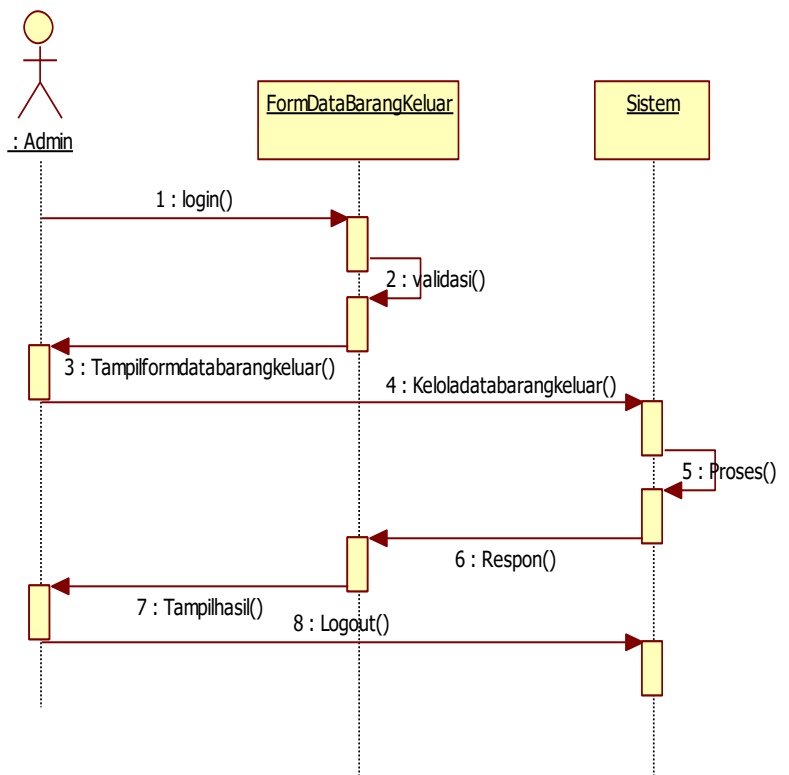


Gambar 3.8 Sequence Diagram Kelola Data Barang  
Masuk

Sequence diagram ini menjelaskan proses kelola data barang masuk, admin melakukan *login* terlebih dahulu. Selanjutnya akan masuk *form* menu admin. Untuk

memulai kelola ini, masuk ke menu data barang masuk, kemudian akan tampil *form* data barang masuk. Setelah itu, kelola data barang masuk. Kemudian sistem akan memproses. Setelah itu akan muncul tampilan sesuai dengan instruksi yang diberikan.

### 3.2.3.4 Sequence Diagram Kelola Data Barang Keluar



Gambar 3.9 Sequence Diagram Kelola data Barang Keluar

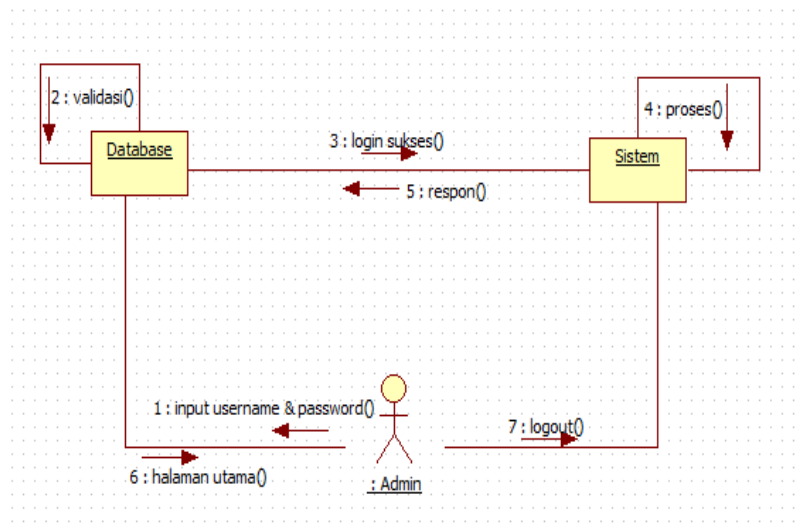
Sequence diagram ini menjelaskan proses

kelola data barang keluar, admin melakukan *login* terlebih dahulu. Selanjutnya akan masuk *form* menu admin. Untuk memulai kelola ini, masuk ke menu data barang keluar, kemudian akan tampil *form* data barang keluar. Setelah itu, kelola data barang keluar. Kemudian sistem akan memproses. Setelah itu akan muncul tampilan sesuai dengan instruksi yang diberikan.

### 3.2.4 Collaboration Diagram

*Collaboration Diagram* adalah suatu diagram yang memperlihatkan atau menampilkan pengorganisasian interaksi yang terdapat disekitar objek (seperti halnya *sequence diagram*) dan hubungannya terhadap yang lainnya. *Collaboration Diagram* lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan/*message*.

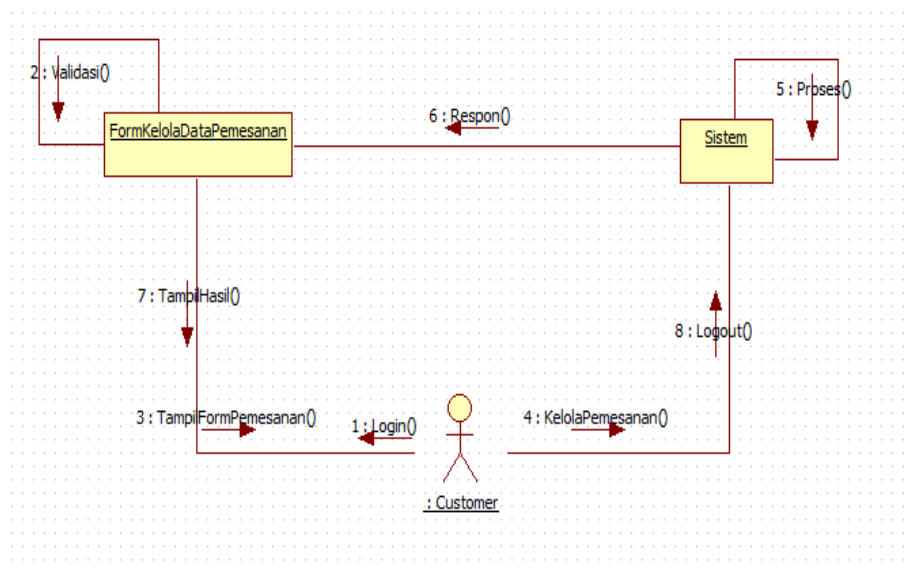
#### 3.2.4.1 Collaboration Diagram Login



*Gambar 3.10 Collaboration Diagram Login*

*Collaboration diagram* ini menjelaskan proses *login*. Admin *login* terlebih dahulu dengan memasukkan *username* dan *password*. Jika *username* dan *password* benar maka *login* sukses. Dan akan muncul halaman utama.

### 3.2.4.2 Collaboration Diagram Pemesanan

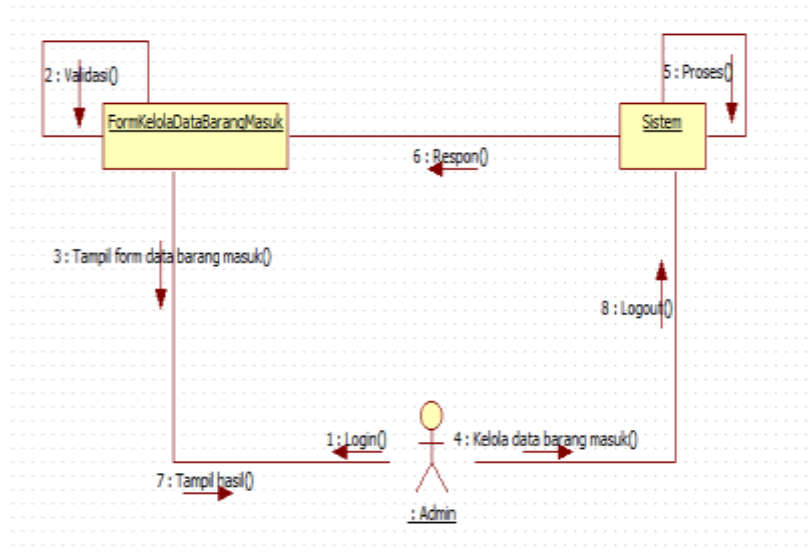


*Gambar 3.11 Collaboration Diagram Kelola Pemesanan*

*Collaboration diagram* ini menjelaskan proses pemesanan. Customer *login* terlebih dahulu, selanjutnya akan masuk *form* menu pemesanan. Untuk memulai kelola ini, masuk ke menu pemesanan, kemudian akan tampil *form* data pemesanan. Setelah itu, isi data pemesanan. Kemudian sistem akan memproses. Setelah itu akan muncul tampilan sesuai dengan

instruksi yang diberikan.

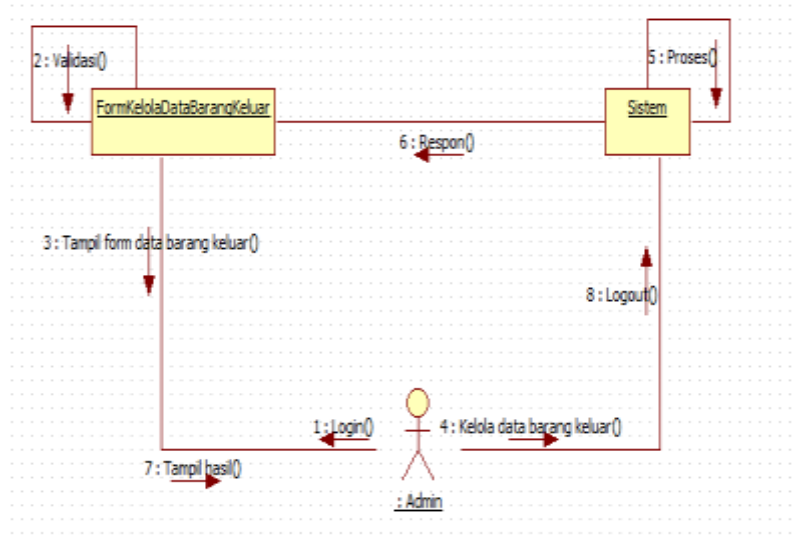
### 3.2.4.3 Collaboration Diagram Kelola Data Barang Masuk



Gambar 3.12 Collaboration Diagram Kelola Data  
Barang Masuk

*Collaboration diagram* ini menjelaskan proses kelola data barang masuk, admin melakukan *login* terlebih dahulu. Selanjutnya akan masuk *form* menu admin. Untuk memulai kelola ini, masuk ke menu data barang masuk, kemudian akan tampil *form* data barang masuk. Setelah itu, isi data barang masuk. Kemudian sistem akan memproses. Setelah itu akan muncul tampilan sesuai dengan instruksi yang diberikan.

### 3.2.4.4 Collaboration Diagram Kelola Data Barang Keluar



*Gambar 3.13 Collaboration Diagram Kelola Data  
Barang Keluar*

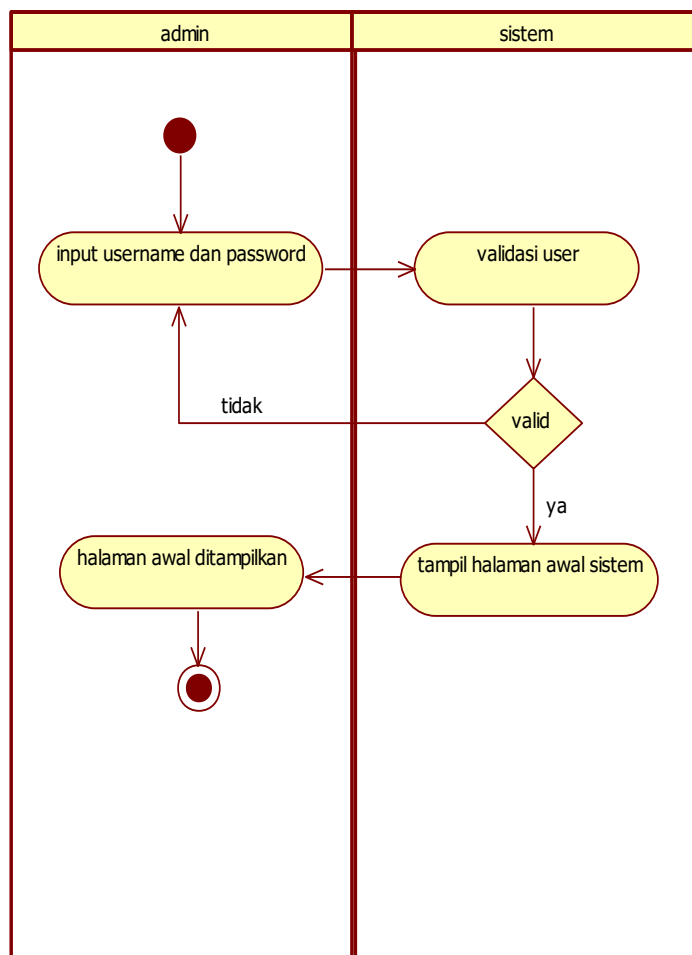
*Collaboration diagram* ini menjelaskan proses kelola data barang keluar, admin melakukan *login* terlebih dahulu. Selanjutnya akan masuk *form* menu admin. Untuk memulai kelola ini, masuk ke menu data barang keluar, kemudian akan tampil *form* data barang keluar. Setelah itu, isi data barang keluar. Kemudian sistem akan memproses. Setelah itu akan muncul tampilan sesuai dengan instruksi yang diberikan.



### 3.2.5 Activity Diagram

*Activity diagram* adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.

#### 3.2.5.1 Activity Diagram Login



*Gambar 3.14 Activity Diagram Login*

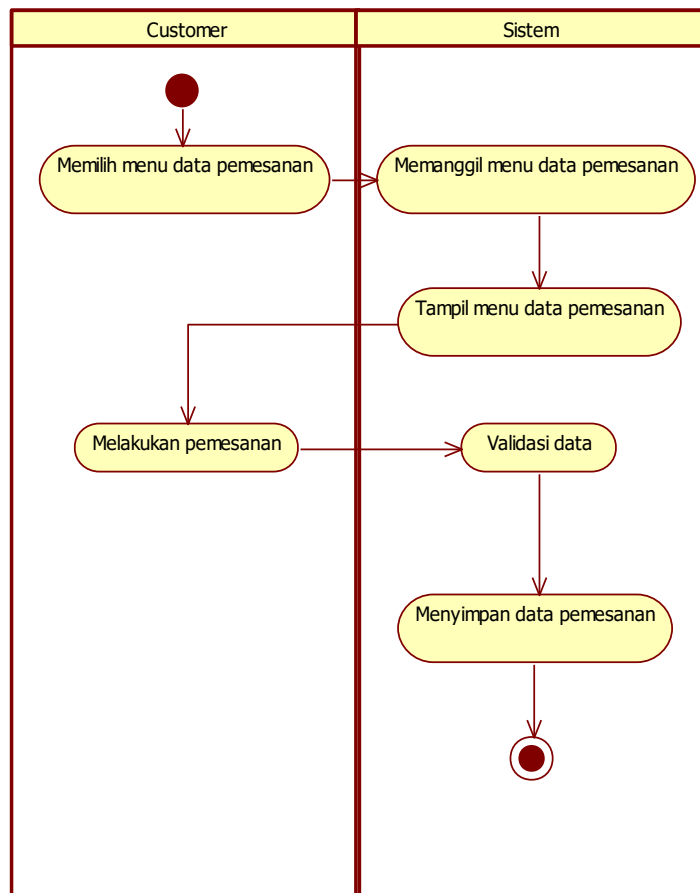
Aplikasi menampilkan *form login*. Kemudian *admin*

## SISTEM INVENTORY

---

memasukkan *username* dan *password*. Sistem melakukan pengecekan pada *database* yang kemudian akan dilakukan validasi *login user*. Apabila *username* dan *password* benar (*valid*) maka sistem akan menampilkan halaman menu awal *user*, apabila salah, *user* akan diminta memasukkan kembali *username* dan *password* yang benar. Setelah proses *login* sukses, *user* dapat masuk ke aplikasi sesuai dengan level penggunaanya. Dan admin dapat *logout* setelah berhasil mengakses semua *interface*.

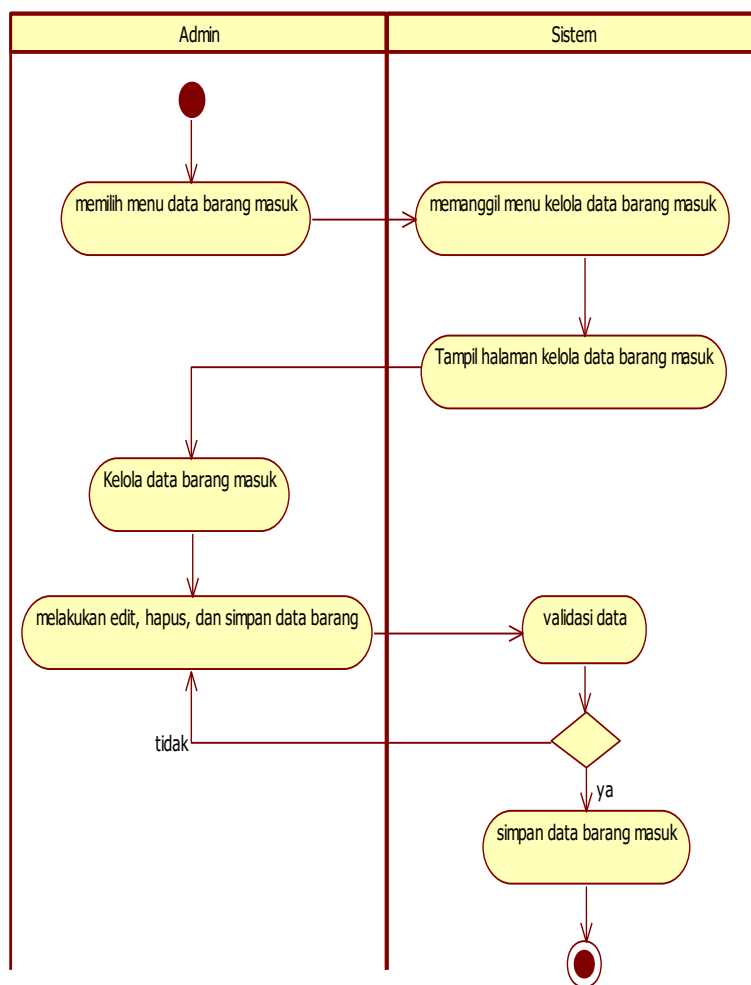
### 3.2.5.2 Activity Diagram Pemesanan



*Gambar 3.15 Activity Diagram Pemesanan*

Untuk mengakses kelola data *pemesanan*, customer memilih menu data pemesanan. Setelah sistem terbuka maka akan muncul halaman data kelola pemesanan. *Customer* melakukan pemesanan. Dan melakukan validasi data yang sudah dimodifikasi. Apabila benar, maka data *user* akan tersimpan.

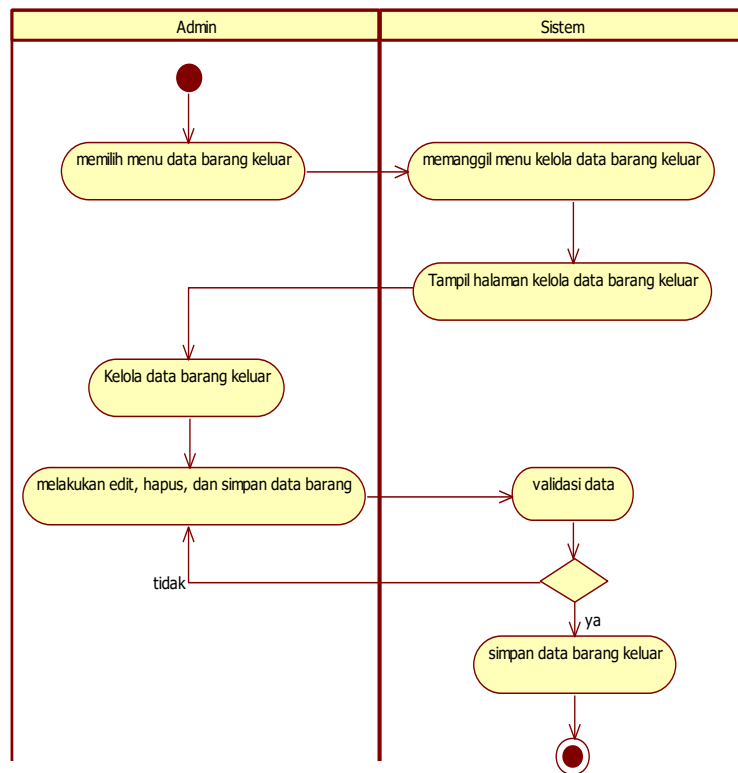
### 3.2.5.3 Activity Diagram Kelola Data Barang Masuk



*Gambar 3.16 Activity Diagram Kelola Data Barang  
Masuk*

Untuk mengakses kelola data barang masuk, admin memilih menu data barang. Setelah sistem terbuka maka akan muncul halaman data kelola barang masuk. Admin dapat melakukan modifikasi data barang masuk sesuai dengan permintaan yaitu tambah, *edit*, dan hapus. Dan melakukan validasi data yang sudah dimodifikasi. Apabila benar, maka data barang masuk akan tersimpan. Apabila salah, maka akan kembali ke tampilan tambah, *edit*, dan hapus. Dan melakukan penyimpanan data yang sudah dimodifikasi.

### 3.2.5.4 Activity Diagram Kelola Data Barang Keluar



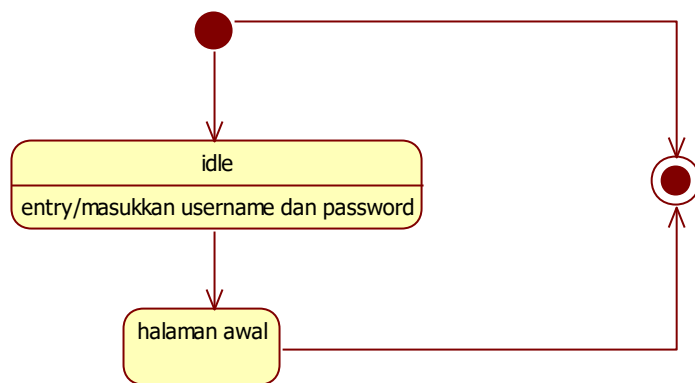
*Gambar 3.17 Activity Diagram Kelola Data Barang Keluar*

Untuk mengakses kelola data barang keluar, admin memilih menu data barang keluar. Setelah sistem terbuka maka akan muncul halaman data kelola barang keluar. Admin dapat melakukan modifikasi data pemesanan sesuai dengan permintaan yaitu verifikasi dan hapus data barang keluar. Dan melakukan validasi data yang sudah dimodifikasi. Apabila benar, maka data barang keluar akan tersimpan. Apabila salah, maka akan kembali ke tampilan verifikasi dan hapus data barang keluar. Dan melakukan penyimpanan data yang sudah dimodifikasi.

### 3.2.6 Statechart Diagram

*Statechart Diagram* menggambarkan transisi dan perubahan keadaan (dari suatu *state* ke *state* lainnya) suatu objek pada *system* sebagai akibat dari stimulan yang diterima *statechart diagram* mendeskripsikan bagaimana suatu objek mengalami perubahan status adanya *trigger* dari *event-event*. Menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek.

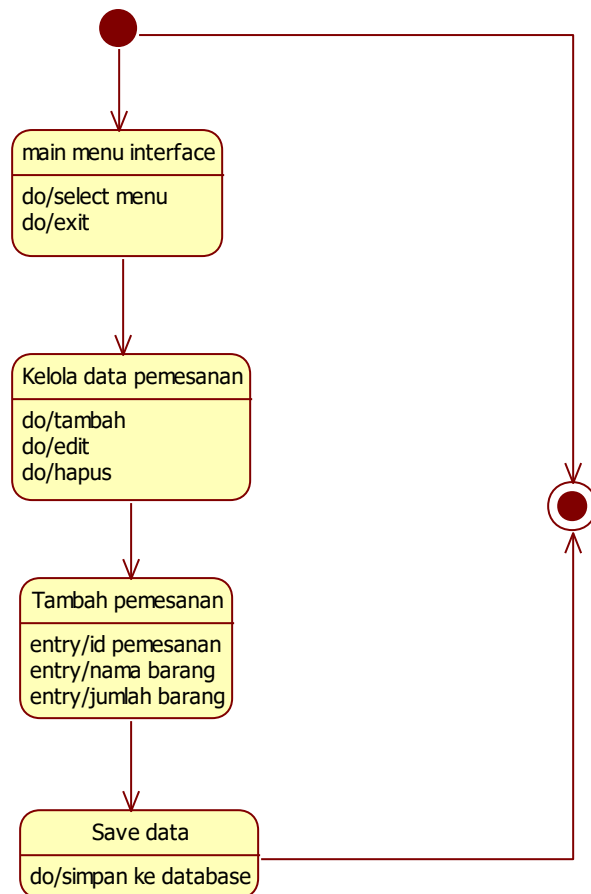
#### 3.2.6.1 Statechart Diagram Login



Gambar 3.18 Statechart Diagram Login

Aktor yang dapat *login* adalah admin dan staf. Aktor mulai dengan memasukkan *username* dan *password*, kemudian sistem akan memvalidasi apakah *username* dan *password* yang dimasukkan valid atau tidak valid. Jika valid maka akan tampil menu admin, tetapi jika tidak valid akan kembali ke *form login*.

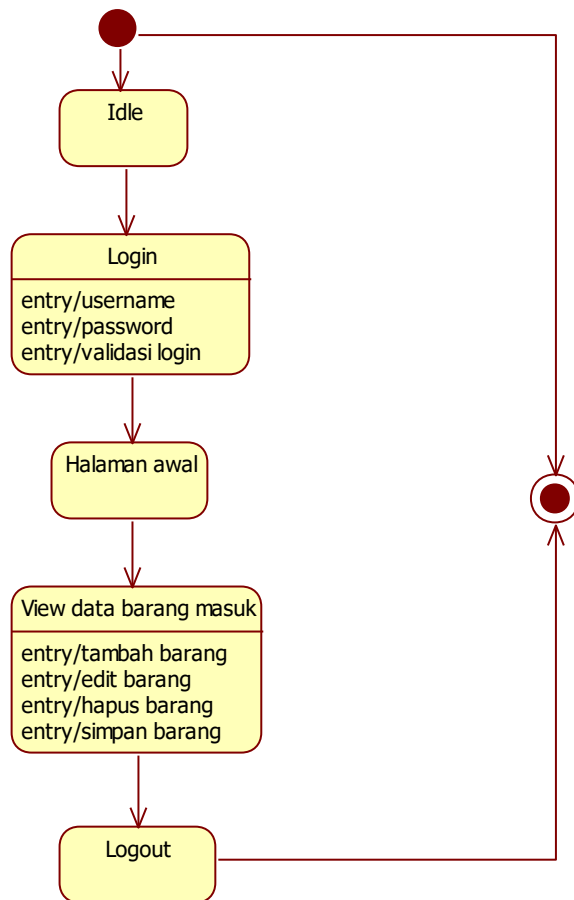
### 3.2.6.2 Statechart Diagram Pemesanan



Gambar 3.19 Statechart Diagram Pemesanan

*Statechart diagram* ini menjelaskan proses kelola data pemesanan. Customer masuk ke halaman pemesanan. Untuk memulai kelola ini, masuk ke menu data pemesanan, kemudian akan masuk *form* data pemesanan dimana dapat melakukan operasi melakukan pemesanan, tambah, edit, dan kurang.

### 3.2.6.3 Statechart Diagram Kelola Data Barang Masuk

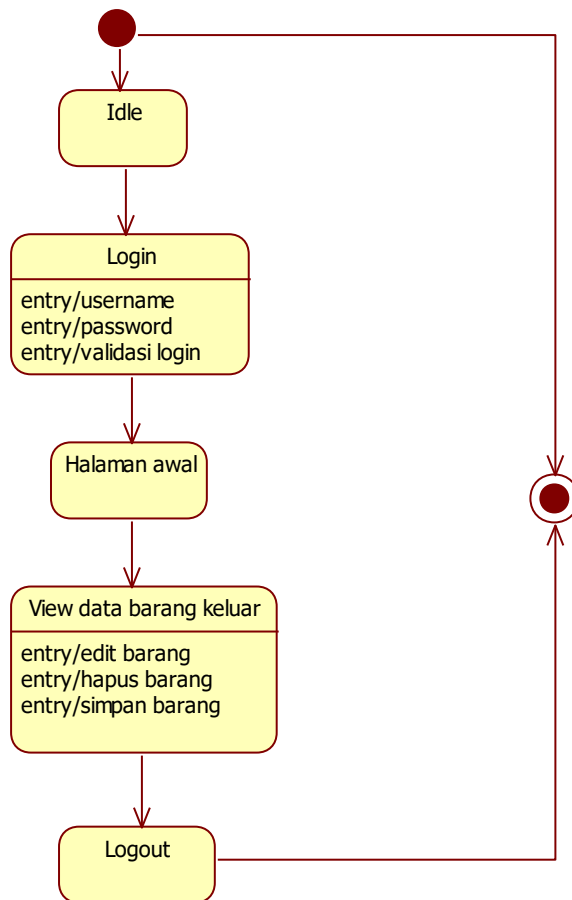


*Gambar 3.20 Statechart Diagram Kelola Data  
Barang Masuk*

*Statechart diagram* ini menjelaskan proses kelola data barang masuk. Admin masuk ke halaman Admin. Untuk memulai kelola ini, masuk ke menu data barang masuk, kemudian akan masuk *form* data barang masuk dimana dapat melakukan operasi tambah barang, *edit* barang, hapus barang, dan simpan barang.



### 3.2.6.4 Statechart Diagram Kelola Data Barang Keluar



*Gambar 3.21 Statechart Diagram Kelola Data Barang Keluar*

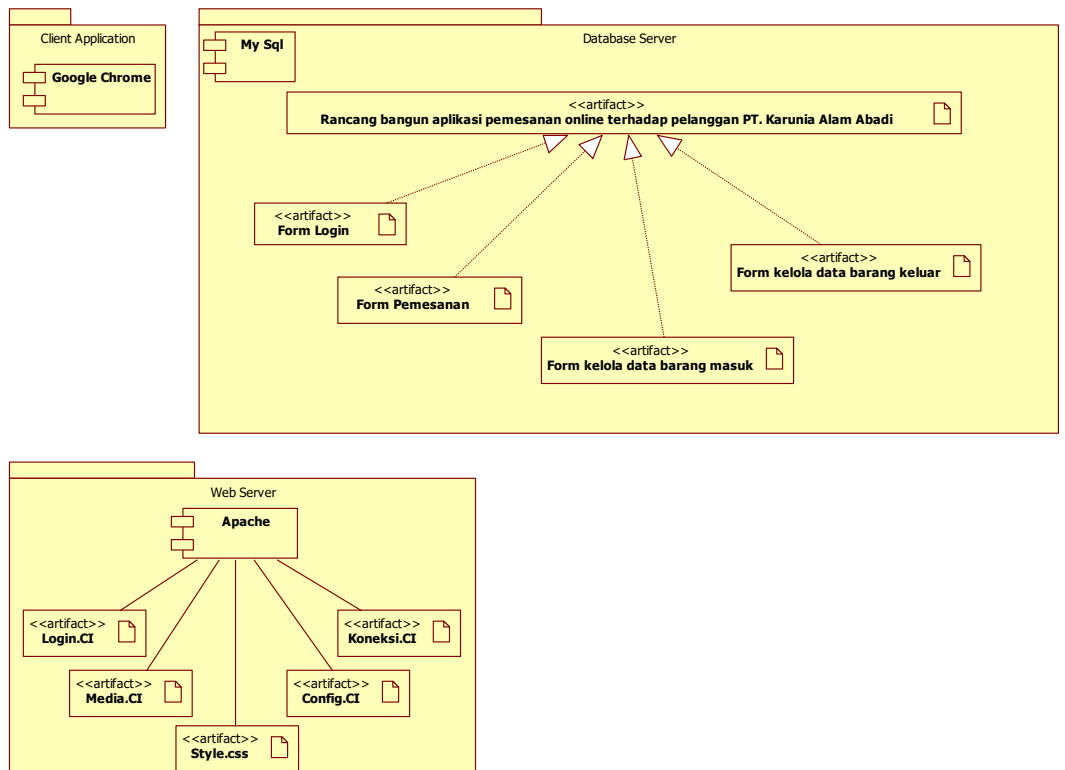
*Statechart diagram* ini menjelaskan proses kelola data barang keluar. admin masuk ke halaman admin. Untuk memulai kelola ini, masuk ke menu data barang keluar, kemudian akan masuk *form* data barang keluar dimana dapat melakukan operasi *edit* barang, hapus barang, dan simpan barang.

## SISTEM INVENTORY

### 3.2.7 Component Diagram

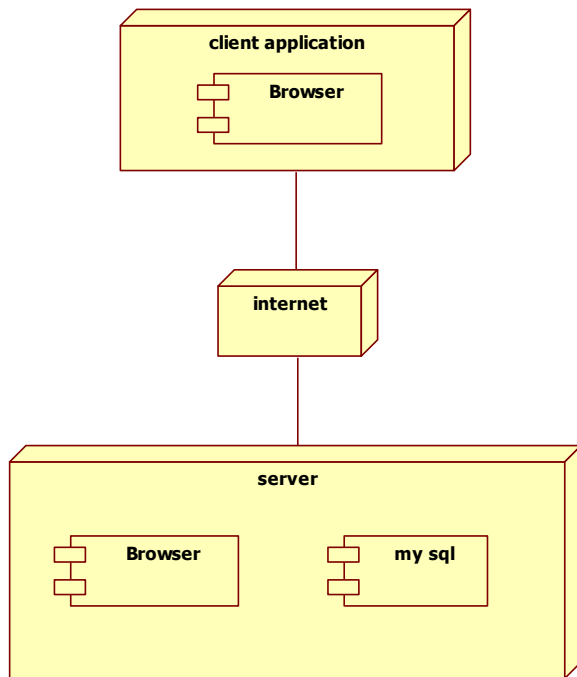
*Component diagram* yaitu salah satu jenis diagram pada UML yang menggambarkan *software* pada suatu sistem. *Component diagram* merupakan penerapan *software* dari satu ataupun lebih *class*, dan biasanya berupa *file* data atau *.exe*, *source code*, *table*, dokumen, dll.

Berikut ini merupakan *component diagram* yang menjelaskan semua komponen yang ada pada sistem.



Gambar 3.22 Component Diagram

### 3.2.8 Deployment Diagram



*Gambar 3.23 Deployment Diagram*

# SISTEM INVENTORY

---

## 4. IMPLEMENTASI

Implementasi merupakan sistem/aplikasi yang dibuat dengan merinci komponen-komponen pendukung berupa program, lingkungan implementasi, tampilan antarmuka, dan petunjuk pemakaian.

### 4.1 Lingkungan Implementasi

Perancangan aplikasi ini dapat berjalan dengan baik karena didukung oleh perangkat pendukung yaitu perangkat lunak dan perangkat keras.

#### 4.1.1 Kebutuhan Perangkat Lunak

Perangkat lunak pendukung yang digunakan adalah sebagai berikut.

Tabel 4.1 Deskripsi Perangkat Lunak *Server*

	Tools/Software	Fungsi
	Microsoft Windows 7 Ultimate 32-bit	Sistem Operasi
	Microsoft SQL Server 2008 R2	Database
	Microsoft Visual Studio 2010	Visual Object Oriented Programming
	PHP CI	Bahasa Pemrograman
	.NET Framework 4.0	Software Pendukung Object Oriented Programming

Tabel 4.2 Deskripsi Perangkat Lunak *Client*

No	Tools / Software	Fungsi
1.	Windows 7	Sistem Operasi
2.	.NET Framework 4.0	Software Pendukung Object Oriented Programming

## SISTEM INVENTORY

---

3.	SQL Server 2008	Database
4.	Microsoft Visual Studio 2010	Visual Object Oriented Programming

### 4.1.2 Kebutuhan Perangkat Keras

Perangkat keras pendukung yang digunakan adalah sebagai berikut.

Tabel 4.3 Deskripsi Perangkat Keras *Server*

No	Nama Perangkat	Spesifikasi	Keterangan
1.	<i>Processor</i>	<i>Intel® core™i5 CPU T6570 @2,10 Ghz</i>	Media untuk menyimpan data aplikasi yang dibuat
2.	<i>Memory</i>	4 GB	<i>Memory System</i> yang digunakan
3.	<i>Hardisk</i>	320 GB	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer
4.	Mouse dan keyboard	Standard	Alat pendukung
5.	Monitor	LCD 14,1 Inchi	Menampilkan data
6.	Infrastruktur jaringan		Bisa dianalogikan sebagai alur proses dari titik awal proses sampai pada akhir proses

Tabel 4.4 Deskripsi Perangkat Keras *Client*

No	Nama Perangkat	Spesifikasi	Keterangan
1.	<i>Hardisk</i>	160 GB	Sebagai tempat untuk menyimpan data yang dibutuhkan tetapi pada sisi

## SISTEM INVENTORY

---

			client tidak diharuskan memiliki ketersediaan space yang besar
2.	<i>Memory</i>	1 GB	Kecepatan client dalam mengakses system ini
3.	<i>Processor</i>	<i>Intel® Pentium® CPU P6200 @2,13 Ghz(2 CPUs), ~2.1GHz</i>	Untuk performansi komputer
4.	Infrastruktur Jaringan		<i>Server dan Host</i>
5.	Mouse dan keyboard	Standard	Alat pendukung
7.	Monitor	Ukuran 14,1 inchi	Menampilkan data

### 4.2 Pembahasan Hasil Implementasi

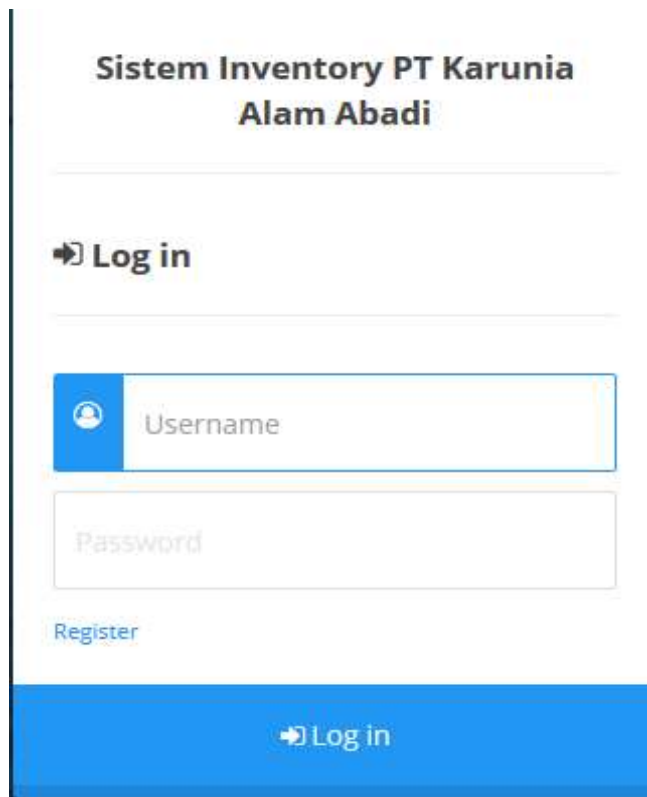
Berdasarkan perancangan yang telah dibuat, didapat hasil dari implementasi yang menjadi tujuan pembuatan perangkat lunak ini yaitu dapat mengelola seluruh data barang di PT. Karunia Alam Abadi.

#### 4.2.1 *FormLogin*

Untuk menggunakan aplikasi ini, user/admin harus login terlebih dahulu dengan memasukkan username dan password pada form *Login* dibawah ini. Hak akses *user* terbagi menjadi 2 level, yaitu admin dan *staff*. Untuk *Login* klik *login*.

## SISTEM INVENTORY

---



The login form is titled "Sistem Inventory PT Karunia Alam Abadi". It features a "Log in" button with a right-pointing arrow icon. Below this is a "Register" link. The form includes two input fields: "Username" and "Password". A blue bar at the bottom contains a "Log in" button with a right-pointing arrow icon.

*Gambar 4.1 Form login*

### 4.2.2 Form Halaman Utama Admin

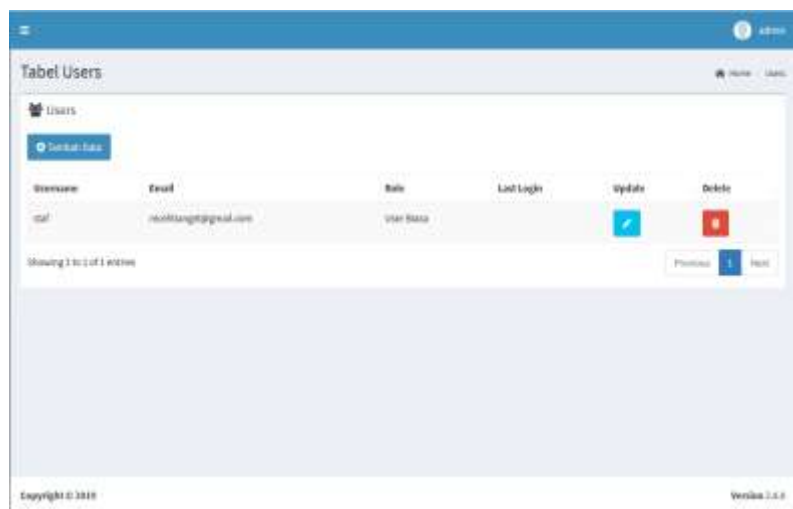


The admin dashboard is titled "PT Karunia Alam". It features a sidebar with navigation links: "Dashboard", "Form", "Table", "Profile", and "Logout". The main content area displays three cards: "33 Stock Barang Gudang" (blue), "82 Stock Barang Pelajar" (green), and "7 User" (orange). Each card has a "Detail" link. The footer shows "Copyright © 2019" and "Version 1.0".

*Gambar 4.2 Form halaman Utama Admin*

Pada menu utama Admin terdapat menu untuk mengelola Data *user*, data barang yang masuk, data barang yang keluar, dan lihat laporan data pengeluaran barang. Admin hanya dapat melihat atau memanipulasi data. Fungsi form laporan adalah melaporkan seluruh kegiatan yang dilakukan oleh User, dimana staff akan melakukan proses pendataan barang.

### 4.2.3 Form Halaman kelola user

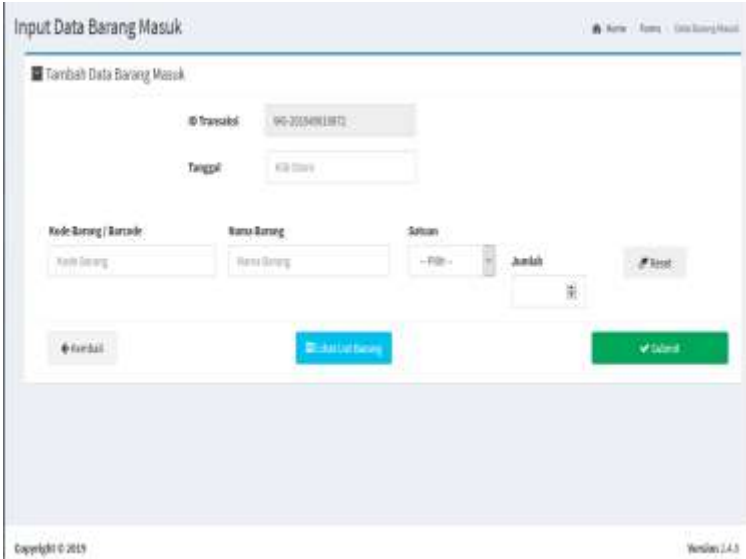


*Gambar 4.4 Form halaman Kelola user*

Untuk mengelola user, Admin hanya cukup mengklik menu kelola data user pada submenu *users* pada form utama admin. Maka sistem akan memunculkan form kelola Data user seperti gambar diatas. Pada form kelola user ini terdapat beberapa navigasi diantaranya tambah, edit, hapus, bersih, keluar. Jika admin ingin melakukan penambahan data setelah mengisi form yang ada, admin hanya melakukan klik *button* tambah dan seterusnya untuk *button-button* yang ada pada form kelola user.



### 4.2.4 *Form Halaman kelola data barang masuk*



The screenshot displays a web form titled "Input Data Barang Masuk" with a sub-header "Tambah Data Barang Masuk". The form includes the following elements:

- ID Transaksi:** A dropdown menu currently showing "WO-20240118072".
- Tanggal:** A date picker showing "18/01/2024".
- Kode Barang / Barcode:** A text input field.
- Nama Barang:** A text input field.
- Satuan:** A dropdown menu showing "Pcs".
- Jumlah:** A text input field.
- Buttons:** "Kembali" (grey), "Tambah Data Barang" (blue), and "Simpan" (green).
- Footer:** "Copyright © 2019" and "Versi 1.0.0".

*Gambar 4.5 Form halaman Kelola Data Barang Masuk*

Untuk mengelola Data barang yang masuk, Admin hanya cukup mengklik menu kelola data barang masuk pada form utama admin dan Staff. Maka sistem akan memunculkan form kelola Data barang masuk seperti gambar diatas. Pada form kelola data ini terdapat beberapa navigasi diantaranya tambah, edit, hapus, bersih, keluar. Jika admin ingin melakukan penambahan data setelah mengisi form yang ada, admin hanya melakukan klik *button* tambah dan seterusnya untuk *button-button* yang ada pada form kelola data barang.

### 4.2.5 Form Halaman kelola data barang keluar

The screenshot shows a web form titled "Tambah Barang Keluar". At the top right, there are navigation links: "Home", "Tabel", and "Tambah Barang Keluar". The form contains the following fields and controls:

- ID Transaksi:** A text input field containing "WO-20200318426".
- Tanggal Masuk:** A date input field containing "2020/03/18".
- Tanggal Keluar:** A date input field with the placeholder text "Klik Enter".
- Kode Barang / Barcode:** A text input field containing "12345".
- Nama Barang:** A text input field containing "BHM Air Minum Cup".
- Satuan:** A dropdown menu with "Btl" selected.
- Jumlah:** A numeric input field containing "12".
- Buttons:** A "Kembali" button with a left arrow icon and a "Simpan" button with a checkmark icon.

At the bottom of the page, there is a footer with "Copyright © 2019" on the left and "Versi 1.0.0" on the right.

*Gambar 4.6 Form halaman Kelola Data Barang Keluar*

Untuk mengelola Data barang yang keluar, Admin hanya cukup mengklik menu kelola data barang keluar pada form utama admin dan Staff. Maka sistem akan memunculkan form kelola Data barang keluar seperti gambar diatas. Pada form kelola data ini terdapat beberapa navigasi diantaranya tambah, edit, hapus, bersih, keluar. Jika admin ingin melakukan penambahan data setelah mengisi form yang ada, admin hanya melakukan klik *button* tambah dan seterusnya untuk *button-button* yang ada pada form kelola data barang.