

## BAB I

---

### E-COMMERCE DALAM PENJUALAN PRODUK BERBASIS WEBSITE

---

#### *1.1 E-Commerce*



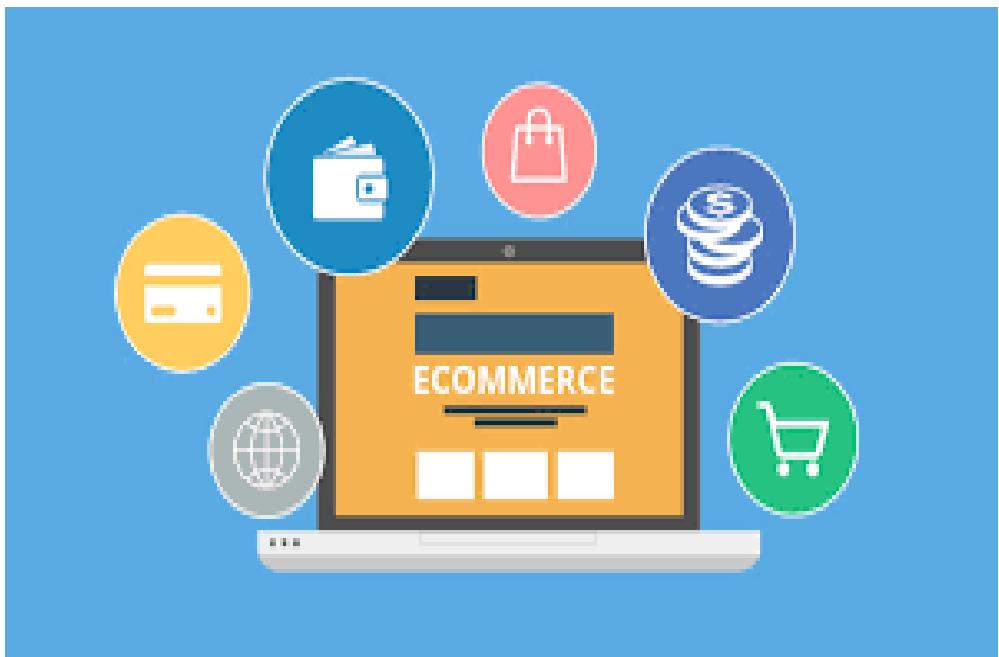
### **1.1.1 Definisi E-Commerce**

Secara umum, istilah E-commerce (perdagangan elektronik) adalah aktivitas pembelian dan penjualan barang / jasa atau pengiriman dana / data melalui jaringan elektronik, khususnya internet. Dengan perkembangan teknologi informasi dan perangkat lunak, ini telah memungkinkan transaksi elektronik konvensional. Situs web ini digunakan sebagai aplikasi toko offline. Situs web e-commerce mengelola berbagai fungsi seperti tampilan produk, pemesanan online, dan inventaris stok, untuk menjalankan fungsi utama sebagai e-commerce. Perangkat lunak yang digunakan diinstal pada server e-commerce dan digunakan bersama dengan sistem pembayaran online untuk pemrosesan transaksi. Secara umum e-commerce berarti berbisnis melalui jaringan yang saling berhubungan (network / internet interconnection).

Aktivitas menggunakan informasi digital dan teknologi pemrosesan komunikasi dalam melakukan transaksi bisnis untuk menciptakan, mengubah, dan mendefinisikan kembali hubungan antara penjual dan pembeli adalah sebuah definisi lain dari E-commerce. Sederhananya, E-commerce dapat diartikan sebagai transaksi pembelian dan penjualan barang, layanan atau transfer dana atau data menggunakan elektronik yang terhubung ke internet. Transaksi e-commerce tidak lagi baru di negara ini, bahkan perkembangannya

cukup cepat.

### 1.1.1 Sejarah E-Commerce

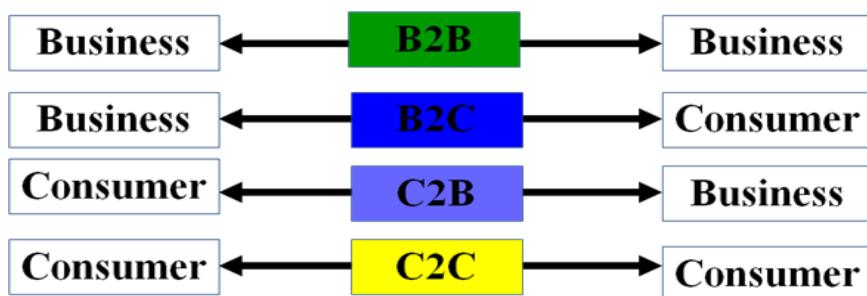


E-Commerce pertama kali diperkenalkan pada tahun 1994 ketika spanduk elektronik pertama kali dibeli untuk tujuan promosi dan iklan di halaman web (situs web). Menurut Forrester Research, perdagangan elektronik menghasilkan penjualan US \$ 12,2 juta pada tahun 2003. Istilah "perdagangan elektronik" telah mengubah persaingan dengan waktu. Awalnya, perdagangan elektronik, sarana komersial, seperti penggunaan EDI untuk pembelian dokumen komersial, seperti pesanan pembelian atau faktur elektronik. Kemudian berkembang menjadi sebuah kegiatan yang memiliki istilah "web commerce" yang lebih tepat - pembelian barang dan jasa melalui World Wide Web

melalui server yang aman (HTTPS), yaitu protokol khusus yang menggunakan enkripsi untuk data yang penting bagi pelanggan. Pada awalnya kompilasi web menjadi populer pada tahun 1994, banyak wartawan memperkirakan bahwa e-commerce akan menjadi sektor ekonomi baru. Namun, hanya sekitar empat tahun kemudian protokol aman seperti HTTPS dipasang dan digunakan secara luas. Kisaran antara 1998 dan 2000 banyak bisnis di AS dan Eropa mengembangkan situs web perdagangan ini.

### 1.1.2 Jenis-Jenis E-Commerce

Ada beberapa jenis transaksi dalam e-commerce, termasuk bisnis ke bisnis, bisnis ke konsumen, konsumen ke konsumen atau konsumen ke bisnis. Oleh karena itu, persyaratan e-commerce dan e-bisnis selalu disepakati.



#### 1. E-Commerce Bisnis ke Bisnis (B2B)

Transaksi e-commerce ini dilakukan oleh dua pihak yang memiliki kepentingan bisnis. Kedua belah pihak saling memahami dan mempelajari bisnis yang mereka jalankan.

Bisnis ini dijalankan secara berkelanjutan, atau hanya disetujui. Contoh sederhana B2B ini adalah produsen dan pemasok yang bertransaksi online baik untuk berkonsultasi dengan kebutuhan barang, hingga proses pembayaran.

#### 2. Bisnis E-commerce ke Konsumen (B2C)

Bisnis kepada konsumen dilakukan oleh bisnis dan konsumen. Transaksi e-commerce ini berlangsung seperti perdagangan normal. Pelanggan mendapatkan penawaran produk dan melakukan pembelian secara online.

### 3. C2C e-commerce (Konsumen ke Konsumen)

Untuk C2C, transaksi dilakukan oleh konsumen ke konsumen. Jika Anda sering menggunakan Tokopedia, Bukalapak, OLX dan sejenisnya, maka inilah yang disebut B2C e-commerce.

Transaksi jual beli dilakukan secara online melalui pasar. Jadi C2C di sini adalah partisipasi antara penjual dan pembeli.

#### 4. Consumen to Busines (C2B)

C2B adalah kebalikan dari B2C di mana pelanggan terakhir bertindak sebagai penjual dan perusahaan bertindak sebagai pembeli.

#### 5. Media atau Aplikasi E-commerce

Sebagaimana dibahas dalam kasus e-commerce di atas, transaksi bisnis ini kompatibel dengan aplikasi media online lainnya (baca: media online), seperti katalog, email, keranjang belanja, layanan eb, EDI dan protokol transfer file. B2B (bisnis ke bisnis).

### ***1.2 Penjualan***



### **1.1.2 Pengertian penjualan**

Penjualan adalah sebuah sumber kehidupan perusahaan, karena dari penjualan dapat diperoleh keuntungan dan bisnis menarik pelanggan yang berusaha mencari daya tarik, konsumen dapat mengetahui hasil dari produk yang dihasilkan. Promosi dilakukan untuk meningkatkan volume penjualan dan strategi untuk mendorong pembeli melakukan transisi. Sedangkan penjualan adalah suatu pemindahan barang dan jasa yang dilakukan atau dilaksanakan oleh penjual. Secara umum, perusahaan yang ingin menyelesaikan proses peningkatan volume penjualan akan melakukan kegiatan promosi melalui iklan, penjualan pribadi, dan publisitas. Istilah penjualan sering bingung dengan istilah promosi.

Sedangkan penjualan hanya mencakup penghapusan barang atau jasa, atau penggunaan penjual. Tidak Perlu Promosi. Jadi penjualan hanyalah bagian dari promosi. Sifat kegiatan dan struktur pelaporan dalam suatu perusahaan dapat berbeda. Namun, pada prinsipnya, semua itu dimaksudkan untuk memudahkan tugas penjualan. Untuk staf pemasaran, kegiatan pemasaran sangat penting. Mungkin salah satu yang paling penting adalah iklan, sementara efektivitasnya berhubungan langsung dengan kunjungan konsumen ke perusahaan. Dalam hal ini berkaitan langsung dengan jumlah penjualan yang dihasilkan. Dalam menjual masalah layanan yang sangat penting, kunjungan tenaga penjual ke pelanggan akan mempertimbangkan munculnya banyak pesanan. Fungsi tenaga penjualan adalah menghasilkan produk yang diproduksi oleh perusahaan. Selain itu bisa juga memberikan saran yang menarik. Ini sangat efektif untuk meningkatkan penjualan.

### **1.2.1 Sejarah Penjualan**



Penjualan dalam praktiknya sudah ada sejak zaman peradaban kuno. Orang Yunani dan Romawi Kuno mempraktikkan ilmu perdagangan dan secara aktif berkomunikasi secara persuasif dengan konsumen mereka. Demikian juga di peradaban lain yang mempromosikan perdagangan. Namun, konsep pemasaran modern yang dikenal saat ini hanya muncul dan dikembangkan selama Revolusi Industri yang terjadi pada abad ke-18 dan ke-19. Periode ini ditandai oleh perubahan sosial yang didorong oleh perkembangan teknologi dan inovasi pengetahuan.

Salah satu perubahan ini adalah industri yang memproduksi barang-barang konsumsi oleh massa. Hal ini juga didukung oleh pengembangan moda transportasi dan media massa yang dibutuhkan oleh produsen untuk menemukan cara untuk mengatur distribusi barang dan jasa. Selama Revolusi Industri, barang-barang

konsumen masih langka dan produsen dapat menjual hampir semua yang mereka produksi selama konsumen dapat membeli. Karena itu, mereka fokus pada pengembangan produksi dan distribusi dengan dukungan finansial paling sedikit. Ini juga berfokus pada pengembangan ilmu pemasaran pada saat itu, yang berkonsentrasi pada efisiensi biaya distribusi dan pembukaan pasar baru. Dari awal abad ke-20 hingga beberapa tahun setelah Perang Dunia II, persaingan bisnis telah meningkat dan fokus ilmu pemasaran telah bergeser dari fokus produksi ke fokus penjualan. Komunikasi, periklanan, dan sains merek mulai menjadi penting karena perusahaan mendapatkan banyak barang di pasar yang semakin ramai.

### **1.2.2 Strategi Penjualan**

Kata strategi dalam Kamus Besar Bahasa Indonesia (KBBI) adalah rencana yang cermat dalam mengembangkan program yang sejalan dengan tujuan dan sasaran tertentu. Strategi penjualan adalah cara untuk memperoleh pendapatan dari penjualan (sales) yang disusun berdasarkan kekuatan, keunggulan, yang mendukung perusahaan atau seseorang.

Kemudian diubah menjadi rencana yang dapat ditindaklanjuti untuk menemukan pelanggan baru atau mengembangkan pelanggan yang sudah ada. Sederhananya, strategi penjualan adalah kegiatan yang direncanakan yang bertujuan untuk meningkatkan

volume penjualan produk. Baik itu barang atau jasa.

### 1. Target Pasar

Hal pertama dalam menentukan strategi penjualan adalah Anda harus mengetahui target pasar untuk produk yang Anda tawarkan. Target pasar Anda adalah kumpulan orang yang ingin Anda fokuskan dalam promosi atau cara lain untuk meningkatkan volume penjualan Anda.



Contoh berikut ini akan memudahkan kita untuk memahami target pasar. Jika Anda seorang penjual yang menjual unit apartemen dengan harga terendah satu miliar rupiah. Jika Anda menawarkan unit apartemen kepada seseorang yang baru memulai karir profesionalnya, maka unit tersebut akan dijual dengan harga yang sangat murah. Lain halnya jika Anda menawarkan unit kepada CEO suatu perusahaan atau perusahaan, itu harus dijual lebih banyak.

Dalam dunia penjualan dan pemasaran tidak ada target pasar "semua orang" bahkan untuk barang-barang berharga murah. Semakin banyak Anda tahu tentang target pasar Anda, semakin baik peluang Anda untuk meningkatkan penjualan.

## 2. Jenis produk

Kedua adalah jenis produk, tidak semua produk sejenis memiliki target pasar yang sama. Coba ambil contoh jus buah sebagai produk.

Jus buah yang kita temukan di banyak restoran atau bahkan di jalan berfungsi hanya terbatas pada jenis minuman. Ini berbeda dari produk jus yang fokus pada nilai kesehatannya. Target pasar dan harga pasti akan berubah. Jus ini akan membantu hal-hal seperti tidak ada gula, diet yang efektif, manfaat kesehatan, kandungan gizi, dan vitamin dalam jus. Hal-hal menarik di atas akan sangat menarik bagi orang yang memprioritaskan kesehatan. Rahasia strategi penjualan sebenarnya adalah layanan dan bukan produk jus. Semua orang bisa membuat jus dengan mudah. Tetapi tidak semua orang memiliki waktu untuk membeli berbagai bahan dan mengolahnya menjadi jus dengan manfaat dan rasa yang sama.

### ***1.3 Website***



### 1.3.1 Pengertian Website

Web adalah kumpulan informasi / koleksi halaman yang biasanya diakses melalui internet. Setiap orang di berbagai tempat dan kapan saja dapat terhubung saat terhubung ke internet secara online di jaringan. Secara teknis, situs web adalah kumpulan halaman, yang tergabung dalam domain atau subdomain tertentu. Situs web di World Wide Web (WWW).

#### Memahami Situs Web Menurut Para Ahli

Halaman diakses dan dibaca melalui browser seperti Netscape Navigator, Internet Explorer, Mozilla

Firefox, Google Chrome dan aplikasi browser lainnya  
(pemahaman situs web dari Hakim Lukmanul, 2004)

Situs web adalah kumpulan halaman web yang saling terkait dan file terkait. Web terdiri dari halaman / halaman, dan halaman gabungan yang disebut homepage. Halaman depan berada di posisi teratas, dengan halaman yang ditautkan di bawah. Setiap halaman di beranda bawah disebut halaman anak, yang berisi hyperlink ke halaman lain di web (pemahaman situs web dari Gregory, 2000)

### **1.3.1 Sejarah Website**

Situs web ini pertama kali ditemukan oleh Sir Timothy John, Tim Berners-Lee. Pada tahun 1991 situs web terhubung ke jaringan. Tujuan situs web dibuat pada waktu itu untuk memfasilitasi pertukaran pertukaran dan memperbarui informasi bagi para peneliti di tempat kerja mereka. Dengan demikian pengertian situs web pada waktu itu masih sebatas bertukar informasi, bukan memahami terminologi situs web.



Situs web menyetujui publik setelah diumumkan dari CERN pada 30 April 1993. CERN menyatakan bahwa situs web dapat digunakan secara gratis oleh semua orang. Pada saat ini, ide situs web telah memasuki ruang publik karena dapat digunakan oleh semua orang di mana saja dan kapan saja.

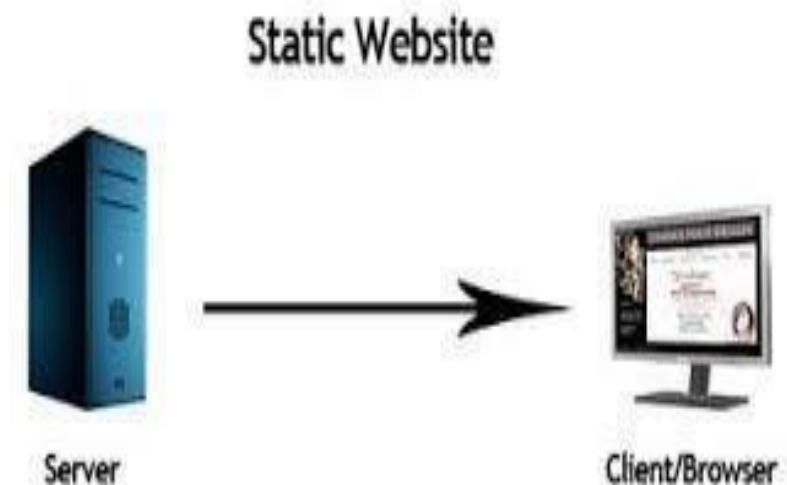
Dimbleweb mencoba menjelaskan jenis dan tipe situs web yang umumnya tersedia.

Secara garis besar, suatu situs web dapat digolongkan menjadi 3 bagian, yaitu:

## 1. Situs Web Statis

Web Statis adalah web di mana pengguna tidak dapat mengubah situs web secara langsung melalui browser. Jadi web statis ini hanya dapat

diubah secara manual. Sedangkan web dinamis adalah situs web yang kontennya dapat dengan mudah diperbarui dengan mudah



## 2. Situs Web Dinamis

Dynamic Website adalah jenis halaman web yang disusun oleh konten dan tata letak yang kaya akan informasi di dalamnya. Dinamakan situs web dinamis karena kontennya dapat berubah. ...

Kemudian, situs web Statis (Situs Web Statis) adalah situs web yang kontennya secara statistik / tidak berubah.



### 3. Situs Web Interaktif

Dynamic Website adalah jenis halaman web yang disusun oleh konten dan tata letak yang kaya akan informasi di dalamnya. Dinamakan situs web dinamis karena kontennya dapat berubah. ...

Kemudian, situs web Statis (Situs Web Statis) adalah situs web yang kontennya secara statistik / tidak berubah.

Statistik situs web adalah situs web yang memiliki halaman yang tidak berubah. Ini berarti membuat perubahan pada halaman yang dilakukan secara manual dengan mengubah kode ke dalam struktur situs.

Situs Web Dinamis adalah situs web yang dirancang untuk dimaksudkan untuk pembaruan sesering

mungkin. Apa pun yang dapat diakses oleh pengguna secara umum, juga menyediakan halaman backend untuk menginstal konten dari situs web. Contoh umum dari situs dinamis adalah berita web atau portal web yang mencakup berita, fasilitas pemungutan suara, dan sebagainya. Situs web interaktif adalah web yang saat ini sedang booming. Salah satu contoh situs web interaktif adalah blog dan forum. Di situs web para pengguna ini dapat berdiskusi dan bertabrakan argumen

tentang apa yang mereka anggap. Situs web biasa seperti memiliki moderator untuk topik yang disetujui yang dibahas tidak keluar jalur.

Tiga jenis situs di atas adalah jenis situs yang sering dibuat oleh pengembang di Indonesia seperti pembuatan situs web, di mana jenis yang paling diinginkan adalah DINAMIK, karena pemilik situs web dapat membuat perubahan konten secara mandiri.

### **1.3.2 Perkembangan Web**

Sejak web ditemukan pada tahun 1990, Web menjadi semakin populer dan layanan internet dengan pengguna terbanyak. Sebagian besar pengguna internet bahkan berpikir bahwa Web adalah satu-satunya layanan di internet. Teknologi web terus bergulir dari era web 1.0, web 2.0 dan tren menuju web

3.0. Sayangnya, karakteristik antara versi web 1.0, hingga 3.0 tidak disetujui oleh para ahli. Namun, bukan berarti kita tidak tahu polanya. Berikut ini adalah pola dari setiap versi web sesuai dengan sumber rock yang saya baca.

#### 1. Web 1.0



Web 1.0 dimulai di era web tahun 90-an.

Ciri utama era ini adalah informasi statistik.

#### Karakteristik web 1.0

Situs web umumnya statis yang berubah atau tidak berubah sama sekali. Situs web publik tidak interaktif.

Teknologi yang digunakan adalah teknologi tertutup.

Frame desain web menggunakan frame.

Font kombinasi warna dan teks terlihat norak karena hanya terbatas pada 16 warna dan 6 jenis font.

Informasi yang tersedia umumnya berupa teks dan gambar berita. Teknologi web 1.0

HTML dasar,  
CSS dasar, dan  
Javascript.

Aplikasi Flash  
dan Java.

Browser populer adalah Netscape dan Internet Explorer. Koneksi internet masih dihubungi hingga maksimal 56 kbps.



## 2. Web 2.0

Web 2.0 dimulai pada era 2005 hingga sekarang, era unik adiri ini adalah konten yang dibuat oleh pengguna dan jejaring sosial.

## Karakteristik Web 2.0

Situs web AJAX bersifat dinamis dan interaktif dengan teknologi.

Pengguna tidak hanya sebagai konsumen tetapi juga sebagai produsen informasi dalam istilah populer yang dibuat oleh pengguna konten (misalnya youtube atau wikipedia).

Teknologi yang digunakan umumnya open source.

Desain web menggunakan tata letak CSS.

Informasi terdiri dari teks, gambar, audio, video dan animasi. Web sebagai media komunikasi dan kolaborasi.

Jejaring sosial mendominasi di era ini.

Munculnya startup.

Munculnya jenis situs web seperti microblogging, berbagi foto dan video, bookmark online, blog dan jejaring sosial.

## Teknologi di Web 2.0

Pustaka Ajax dan jQuery dan Javascript mendominasi. Tingkatkan penggunaan HTML5 dan kurangi penggunaan Flash. Web Seluler dan desain web Responsif.  
· XML dan JSON untuk transfer data.

API · Web dan Mashup (Mashup adalah istilah yang berisi layanan bersama, misalnya menampilkan tweet di peta Google).

Browser populer adalah Chrome, Safari, dan Firefox.

· · Koneksi internet menggunakan broadband (> 1MB / s).

Android dan iOS menyetujui pengguna web seluler.

### 3. Web 3.0

Web 3.0 adalah generasi berikutnya dari teknologi web. Fitur Web 3.0 adalah akses seluler, web semantik, dan personalisasi. Beberapa situs web yang dianggap berasal dari web 3.0 adalah Instagram. Instagram

disebut sebagai permulaan generasi web 3.0 karena layanan ini hanya tersedia melalui perangkat seluler.

### Karakteristik Web 3.0

Integrasi web ke dalam produk-produk Rumah seperti TV, Kulkas atau Windows, smart atau home entertainment. Contoh dari produk ini adalah Smart Windows dari Samsung atau Google Nexus Q.

Web Semantik, Contoh penggunaan web semantik adalah grafik Pengetahuan di Google Search.

Memproses bahasa alami. Perintah suara. ada beberapa contohnya adalah Siri, Voice Action Android atau Voice search di Google.

Layanan berbasis lokasi dan personifikasi informasi. Web tidak lagi menyediakan informasi, tetapi solusi. Contoh layanan ini adalah Layar dan Google Now.

Komputer (dan web) yang dapat digunakan, contoh produk ini adalah Google glass.

### Teknologi Web 3.0

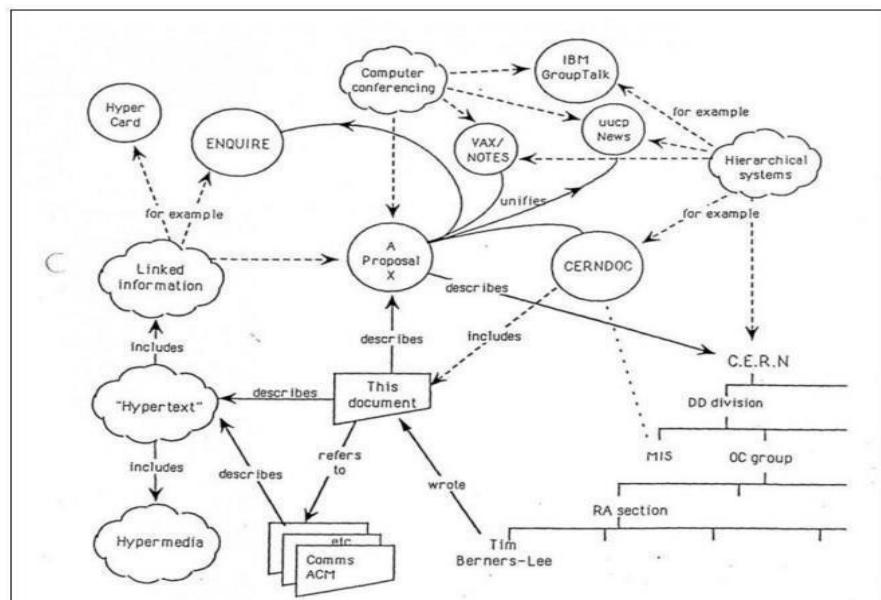
Untuk teknologi yang mendominasi web 3.0 adalah sebagai berikut: Semantic Web

(Contoh bagan pengetahuan).

Kecerdasan Buatan (Siri).

Komunikasi & kolaborasi waktu nyata (Google docs, Google Wave). Augment Reality (Layar, aplikasi di android).

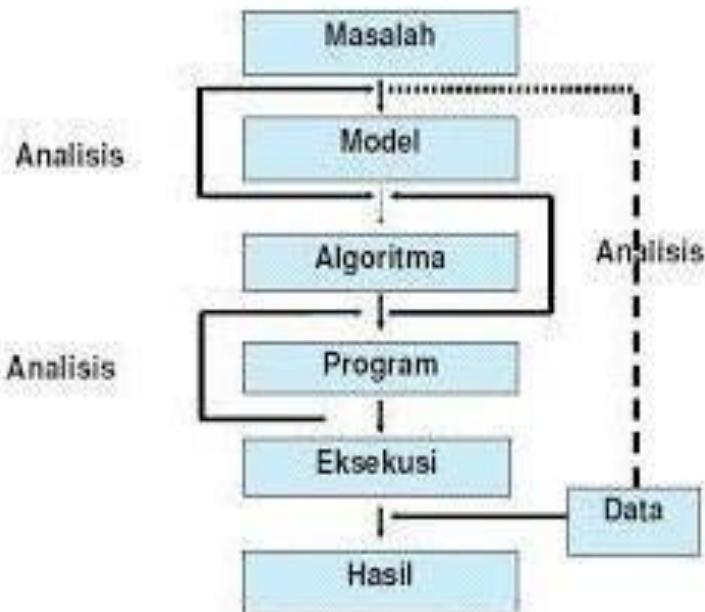
Informasi yang dihasilkan komputer  
(Contoh Wolfram Alpha). HTML5 dan teknologi Cloud.



## 1.4 Algoritma Metode Serakah

### 1.3.2 Metode Serakah

Algoritma serakah adalah algoritma yang mirip dengan pemrograman dinamis, bedanya jawabannya tidak perlu diketahui dalam setiap tahap; dan menggunakan pilihan “serakah” apa yang dilihat terbaik pada saat itu.



Algoritma serakah adalah algoritma yang mengikuti heuristik pemecahan masalah dalam membuat pilihan optimal pada setiap rak dengan tujuan menemukan global optimal. Dalam banyak masalah, strategi serakah biasanya tidak menghasilkan solusi optimal, tetapi heuristik serakah dapat menghasilkan solusi optimal lokal yang membutuhkan solusi optimal global dalam jumlah waktu yang wajar. contohnya, strategi serakah untuk masalah salesman keliling (yang memiliki kompleksitas komputasi tinggi) adalah heuristik berikut: "Pada setiap langkah, kunjungi kota terdekat yang belum dikunjungi." Heuristik ini tidak setuju untuk menemukan solusi terbaik, tetapi diakhiri

dengan menyimpulkan langkah yang masuk akal; Temukan solusi optimal untuk masalah yang sangat rumit. Dalam optimisasi matematis, suatu algoritma yang memecahkan masalah kombinatorial yang memiliki sifat matematika, dan memberikan estimasi faktor konstan untuk masalah optimisasi dengan struktur submodular.

Secara umum, algoritma serakah memiliki lima komponen:

1. Kumpulan kandidat, dari mana solusinya dibuat
2. Fungsi seleksi, yang memilih kandidat terbaik untuk ditambahkan ke solusi
3. Fungsi kelayakan, yang digunakan untuk menentukan apakah seorang kandidat dapat digunakan untuk berkontribusi pada solusi
4. Fungsi obyektif, yang memberi nilai pada solusi, atau solusi parsial, dan
5. Fungsi Solusi, yang akan ditampilkan setiap kali kami menemukan solusi yang lengkap

Algoritma memberikan solusi yang baik untuk beberapa masalah matematika, tetapi tidak untuk yang lain. Sebagian besar masalah di mana mereka bekerja akan memiliki dua properti:



### **1.4.1 Properti pilihan serakah**

Kita dapat membuat pilihan apa pun yang diselesaikan sekarang dan kemudian menyelesaikan submasalah yang muncul kemudian. Pilihan yang dibuat oleh algoritma serakah mungkin bergantung pada pilihan yang dibuat, tetapi tidak pada pilihan di masa depan atau semua solusi untuk submasalah. Iteratif membuat satu pilihan serakah demi satu, mengurangi setiap masalah ke tingkat yang lebih kecil. Dengan kata lain, algoritma serakah tidak pernah bisa memikirkan kembali pilihannya. Ini adalah perbedaan utama dari pemrograman dinamis, yang lengkap dan dijamin untuk menemukan solusi. Setelah setiap instalasi, pemrograman dinamis membuat keputusan berdasarkan semua keputusan yang dibuat dalam instalasi sebelumnya. Algoritma memiliki sejarah panjang dalam ilmu komputer kombinatorial dan teoritis. Heuristik rakus berhasil menghasilkan hasil pada masalah optimal,

[4] dan pertanyaan alami adalah:

- Untuk masalah apa algoritma serakah bekerja secara optimal?
- Untuk masalah apa algoritma serakah meminta solusi optimal?

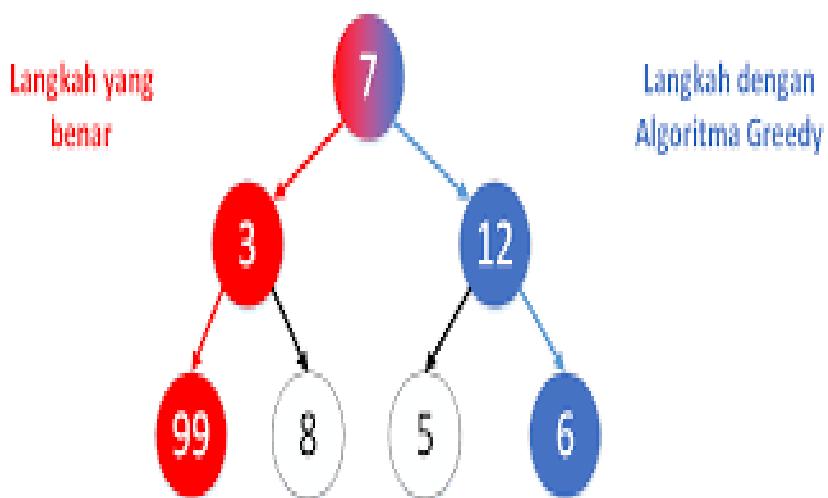
- Untuk masalah apa algoritma serakah yang optimal tidak menghasilkan solusi yang optimal?

Ada banyak literatur yang menjawab pertanyaan ini untuk kelas masalah umum, seperti matroid, serta untuk masalah khusus, seperti set covernya, dan dapat mempertimbangkan kembali jalur algoritmik.

Sebagian besar algoritma serakah (tetapi tidak selalu) gagal menemukan solusi global optimal yang mereka lakukan tidak dapat dilakukan pada semua data. Mereka dapat membuat komitmen untuk pilihan-pilihan tertentu sejak dini yang mencegah mereka menemukan solusi terbaik nanti. Sebagai

contoh, semua algoritma pewarnaan serakah yang dikenal untuk masalah pewarnaan grafik dan semua masalah NP lengkap lainnya tidak menemukan solusi optimal. Meskipun demikian, mereka berguna karena mereka berpikir cepat dan sering memberikan perkiraan yang optimal. Jika algoritma serakah dapat dibuktikan menghasilkan global yang optimal untuk kelas masalah yang diberikan, biasanya merupakan metode yang disukai karena lebih cepat daripada metode optimasi seperti pemrograman dinamis. Contoh dari algoritma serakah ini adalah algoritma Kruskal dan algoritma Prim untuk menemukan pohon rentang minimum, dan algoritma untuk menemukan pohon Huffman yang optimal.

## Pencarian Nilai Terbesar



Algoritma serakah juga muncul dalam perutean jaringan. Menggunakan rute serakah, pesan diteruskan ke jalur tetangga yang "paling dekat" ke tujuan. Lokasi node (dan "proximity" yang ditentukan) dapat ditentukan oleh lokasi fisiknya, seperti pada rute geografis yang digunakan

oleh jaringan ad hoc. Lokasi mungkin juga merupakan konstruksi yang dibuat seperti dalam routing dunia kecil dan tabel hash yang didukung.

Ada beberapa contoh gambar dari algoritma serakah, bias dilihat pada gambar diawah ini

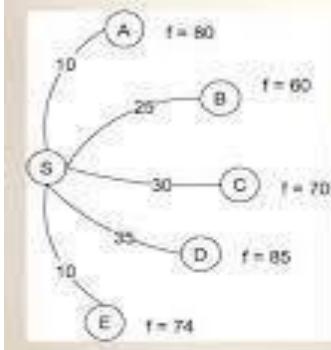
The screenshot shows a Java application window titled "Algoritma Greedy - Mengikuti Rupiah". The code in the editor is as follows:

```
public class GreedyBFS {
    public static void main(String[] args) {
        System.out.println("Masukkan jumlah uang : ");
        int uang = 1000;
        System.out.println("Masukkan jumlah uang : " + uang);
        uang = uang / 100;
        String[] uang_koin = {"Rupiah", "Puluhan", "Dua puluh", "Sepuluh", "Delapan", "Enam", "Empat", "Dua", "Satu"};
        String[] uang_koin_value = {"1000", "100", "200", "50", "20", "10", "5", "2", "1"};
        int i = 0;
        while (uang > 0) {
            if (uang % 100 == 0) {
                System.out.println(uang_koin[i] + " x " + uang_koin_value[i]);
                uang -= 100;
            } else {
                System.out.println(uang_koin[i] + " x " + uang_koin_value[i]);
                uang -= uang_koin_value[i];
            }
            i++;
        }
    }
}
```

The console output shows the breakdown of 1000 Rupiah into coins:

```
Temu yang logis adalah 1000
Puluhan x 10
Dua puluh x 5
Sepuluh x 2
Delapan x 1
Enam x 1
Empat x 1
Dua x 1
Satu x 1
```

## Greedy Best First Search



OPEN = [A,B,C,D,E]  
CLOSED = [S]

- Masalah pemilihan kegiatan adalah masalah dari kelas masalah ini, disepakati untuk memilih jumlah maksimum kegiatan yang tidak saling berbenturan.
- Dalam game komputer Macintosh, Quest Crystal, mengumpulkan kristal koleksi, dengan cara yang mirip dengan masalah salesman keliling. Gim ini memiliki mode demo, di mana ia menggunakan algoritma serakah untuk setiap kristal. Mode demo tidak disetujui, mode demo sering berakhir dengan cepat.
- Pengejaran yang cocok adalah contoh algoritma serakah yang diterapkan pada estimasi sinyal.
- Algoritma serakah menemukan solusi optimal untuk masalah Malfatti dalam menemukan tiga lingkaran terputus-putus dalam segitiga tertentu yang memaksimalkan luas total lingkaran; Terkait dengan algoritma serakah yang sama adalah optimal untuk lingkaran yang terbagi.
- Algoritma ini digunakan untuk membangun pohon Huffman selama pengkodean Huffman di mana ia

menemukan solusi optimal.

- Dalam pembelajaran pohon keputusan, algoritma serakah biasanya digunakan, tetapi tidak disetujui untuk menemukan solusi optimal.
- Salah satu algoritma yang populer adalah algoritma ID3 untuk konstruksi pohon keputusan.
- Algoritma Dijkstra dan algoritma pencarian terkait A \* adalah algoritma serakah yang benar-benar optimal untuk pencarian grafik dan pencarian jalur terpendek.

- Pencarian bersyarat opsional, membutuhkan "heuristik yang dapat diterima" yang tidak akan melebihi biaya jalur.
- Algoritma Kruskal dan algoritma Prim adalah algoritma serakah untuk membuat pohon rentang minimum dari grafik terhubung yang diberikan. Mereka selalu menemukan solusi optimal, yang mungkin tidak unik secara umum.

## **BAB 2**

---

### **TEORI**

---

#### **2.1 *Sejarah CI***

*CodeIgniter* adalah aplikasi sumber terbuka yang berupa kerangka kerja *PHP* dengan *Model*, *View*, *Controller*. Untuk membangun website *dinamis* dengan menggunakan *PHP*. *CodeIgniter* memudahkan pengembangan *web* untuk membuat aplikasi *web* dengan cepat mudah dibandingkan dengan membuatnya dari awal. *CodeIgniter* ini dirilis pertama kali pada 28 Februari 2006. *Versi* stabil terakhir adalah *versi* 3.1.11

##### **2.1.1 *Kerangka Kerja***

Kerangka kerja secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau prosedur-prosedur dan *class-class* untuk tujuan tertentu yang

sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang *programer*, tanpa harus membuat fungsi atau *class* dari awal.

Ada beberapa alasan mengapa menggunakan Kerangka kerja:

- Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
- Relatif memudahkan dalam proses *maintenance* karena sudah ada pola

Umumnya kerangka kerja menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya *validasi*, *ORM*, *pagination*, *multiple database*, *scaffolding*, *pengaturan session*, *error handling*, dll.

- Lebih bebas dalam pengembangan jika dibandingkan *Cms*.

### **2.1.2 Design Patern : MVC ( Model, View, Controller )**

*Model View Controller* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berasal pada bahasa *pemrograman Small Talk*, *MVC* memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu *MVC pattern* dalam suatu aplikasi yaitu:

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa *file template HTML*, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert*, *update*, *delete*, *search*), menangani *validasi* dari bagian *controller*, tetapi tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip *MVC* suatu aplikasi dapat dikembangkan sesuai dengan kemampuan *developer*nya, yaitu programmer yang menangani bagian model dan *controller*, sedangkan *designer* yang menangani bagian *view*, sehingga penggunaan arsitektur *MVC* dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara programmer dan *designer* dalam menangani variabel-variabel yang akan ditampilkan.

Ada beberapa kelebihan *CodeIgniter* (*CI*) dibandingkan dengan Kerangka kerja *PHP* lain,

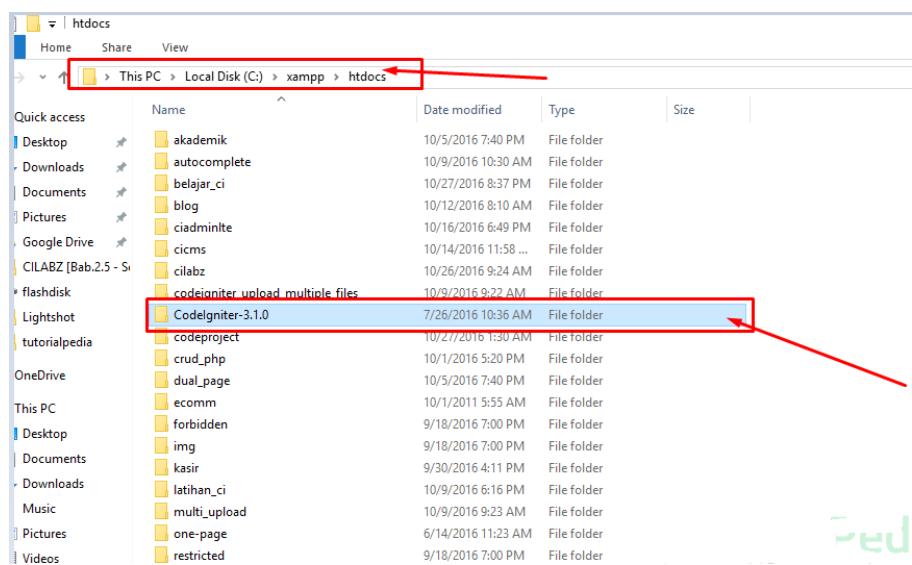
- Performa sangat cepat: salah satu alasan tidak menggunakan kerangka kerja adalah karena eksekusinya yang lebih lambat daripada *PHP from the scratch*, tapi *Codeigniter* sangat cepat bahkan mungkin bisa dibilang *codeigniter* merupakan kerangka kerja yang paling cepat dibanding kerangka kerja yang lain.
- *Konfigurasi* yang sangat minim (*nearly zero configuration*): tentu saja untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetap diizinkan melakukan *konfigurasi* dengan mengubah beberapa *file konfigurasi* seperti *database.php* atau *autoload.php*, tetapi untuk menggunakan *codeigniter* dengan *setting standard*, anda hanya perlu mengubah sedikit saja *file* pada

*folder config.*

- Banyak komunitas: dengan banyaknya komunitas *CI* ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
- Dokumentasi yang sangat lengkap: Setiap paket *instalasi codeigniter* sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

## 2.2 Installasi Codeigniter

- Download Package codeigniter pada situs resminya. Download Codeigniter
- Saat ini versi stabil dari codeigniter adalah 3.1.1.
- Setelah di download ekstrak file codeigniter pada direktori



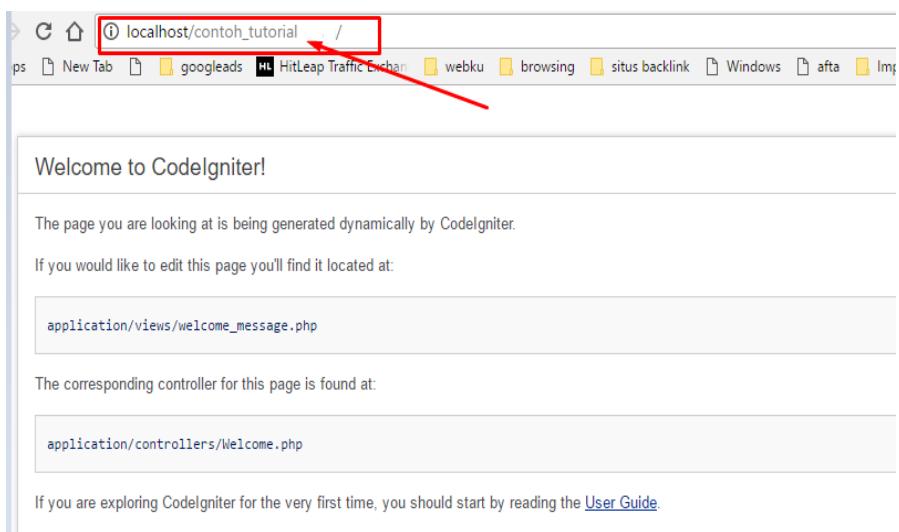
Gambar 2.1

- lalu ubah folder **Codeigniter** menjadi sesuai keinginan kita. disini saya akan ubah folder **codeigniter** tersebut menjadi **contoh\_tutorial**

Name	Date modified	Type	Size
akademik	10/5/2016 7:40 PM	File folder	
autocomplete	10/9/2016 10:30 AM	File folder	
belajar_ci	10/27/2016 8:37 PM	File folder	
blog	10/12/2016 8:10 AM	File folder	
ciaminlte	10/16/2016 6:49 PM	File folder	
cicms	10/14/2016 11:58 ...	File folder	
cilabz	10/26/2016 9:24 AM	File folder	
codeigniter_upload_multiple_files	10/9/2016 9:22 AM	File folder	
codeproject	10/27/2016 1:30 AM	File folder	
contoh_tutorial	10/27/2016 8:59 PM	File folder	
crud_php	10/1/2016 5:20 PM	File folder	
dual_page	10/5/2016 7:40 PM	File folder	
ecomm	10/1/2011 5:55 AM	File folder	
forbidden	9/18/2016 7:00 PM	File folder	
img	9/18/2016 7:00 PM	File folder	
kasir	9/30/2016 4:11 PM	File folder	
latihan_ci	10/9/2016 6:16 PM	File folder	

Gambar 2.2

- Setelah itu, sekarang kita coba akses folder tersebut melalui *browser*.
- jangan lupa untuk menyalakan *XAMPP* terlebih dahulu.
- lalu akses dengan *url*  
 $http://localhost/nama\_folder\_project$  pada  
 tutorial kali ini saya beri nama dengan  
 contoh\_tutorial



Welcome to CodeIgniter!

The page you are looking at is being generated dynamically by CodeIgniter.

If you would like to edit this page you'll find it located at:

```
application/views/welcome_message.php
```

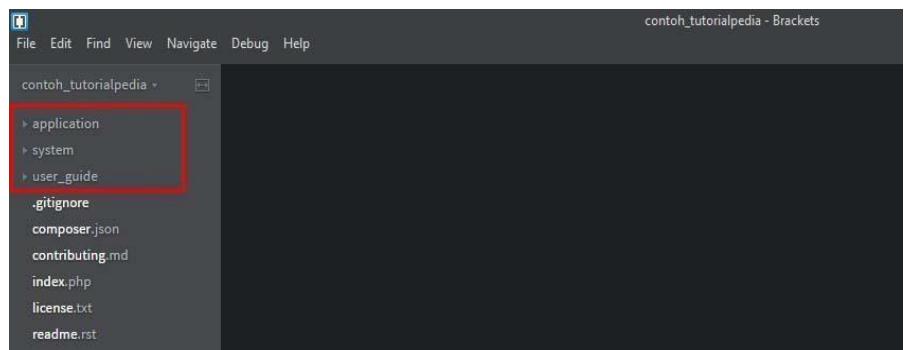
The corresponding controller for this page is found at:

```
application/controllers/Welcome.php
```

If you are exploring CodeIgniter for the very first time, you should start by reading the [User Guide](#).

Gambar 2.3

- Jika sudah tampil halaman **Welcome Codeigniter**, itu artinya anda telah berhasil melakukan *instalasi codeigniter*.
- Penjelasan Tentang Struktur Folder *Codeigniter*
- Didalam folder *codeigniter* terdapat 3 folder utama yaitu *Application, System, User Guide*. Folder *application* berguna sebagai tempat untuk mendevelop aplikasi kita nanti, Folder *system* berisi library atau kumpulan fungsi-fungsi dasar CI, sedangkan *User Guide* berisi sebagai buku panduan atau dokumentasi dari *codeigniter*.



Gambar 2.4

Lalu kita fokus pada folder *application/config* dan semua berawal dari file *routes.php*. Apabila kita buka file tersebut, maka kita akan menemukan *default\_controller* yang telah terisi dengan nama “*welcome*”, yang mana *default\_controller* ini berfungsi ketika user

tidak melakukan atau memasukkan apapun pada URL browser, maka secara otomatis mencari controller dengan nama “*welcome*”. Seperti dibawah ini :

```
contoh_tutorialpedia - [14] | example.com/class/method/id/
+ application [15] | In some instances, however, you may want to remap this relationship
+   + cache [16] | so that a different class/function is called than the one
+   + config [17] | corresponding to the URL.
+     + config.php [18] | Please see the user guide for complete details:
+       + autoload.php [19] | https://codeigniter.com/user_guide/general/routing.html
+         + config.php [20] | RESERVED ROUTES
+           + constants.php [21] | -----
+             + database.php [22] | There are three reserved routes:
+               + doctypes.php [23] |   $route['default_controller'] = 'welcome';
+                 + foreign_chars.php [24] | This route indicates which controller class should be loaded if the
+                   + hooks.php [25] | URL contains no data. In the above example, the "welcome" class
+                     + index.html [26] | would be loaded.
+                       + memcached.php [27] |   $route['404_override'] = 'errors/page_missing';
+                         + migration.php [28] | This route will tell the Router which controller/method to use if those
+                           + mimes.php [29] | provided in the URL cannot be matched to a valid route.
+                             + profiler.php [30] |
+                               + routes.php [31] |   $route['translate_uri_dashes'] = FALSE;
+                                 + smileys.php [32] | This is not exactly a route, but allows you to automatically route
+                                   + user_agents.php [33] | controller and method names that contain dashes. '-' isn't a valid
+                                     + controllers [34] | class or method name character, so it requires translation.
+                                       + core [35] | When you set this option to TRUE, it will replace ALL dashes in the
+                                         + helpers [36] | controller and method URI segments.
+                                           + hooks [37] | Examples: my-controller/index -> my_controller/index
+                                             + language [38] |   my-controller/my-method -> my_controller/my_method
+                                               + libraries [39] |   $route['default_controller'] = 'welcome';
+                                                 + logs [40] |   $route['404_override'] = '';
+                                                   + models [41] |   $route['translate_uri_dashes'] = FALSE;
```

Gambar 2.5

**Controller Welcome** itu sendiri bisa anda temukan pada folder **application/controllers**. Pada file **welcome\_controller** tersebut terdapat fungsi **index** yang gunanya **meload** bagian **view** atau memanggil halaman **view** dengan nama **welcome\_messages** sebagai berikut.

```
20      */
21      public function index()
22  {
23          $this->load->view('welcome_message');
24      }
25 }
```

Gambar 2.6

Sekarang kita lihat pada bagian **views**

yang terletak pada *application/views*. Pada folder *view* tersebut terdapat file dengan nama ***welcome\_message***. Pada bagian inilah semua *html* dan *css* akan diletakkan yang nantinya akan di tampilkan ke pengunjung.

### **2.3 XAMPP**

XAMPP adalah sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari *akronim* kata *Apache*, *MySQL* (dulu) / *MariaDB* (sekarang), *PHP*, dan *Perl*. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti *OS Linux*, *OS Windows*, *Mac OS*, dan juga *Solaris*.

Sejarah mencatat, *software* XAMPP pertama kali dikembangkan oleh tim proyek bernama *Apache Friends* dan sampai saat ini sudah masuk dalam rilis versi **7.3.9** yang bisa didapatkan secara gratis dengan *label GNU (General Public License)*.

Jika dijabarkan secara gamblang, masing-masing huruf yang ada di dalam nama XAMPP memiliki arti sebagai berikut ini:

**X = Cross Platform**

Merupakan kode penanda untuk *software cross platform* atau yang bisa berjalan di banyak sistem operasi.

*A = Apache*

*Apache* adalah aplikasi web server yang bersifat gratis dan bisa dikembangkan oleh banyak orang (*open source*).

*M = MySQL / MariaDB*

*MySQL* atau *MariaDB* merupakan aplikasi database server yang dikembangkan oleh orang yang sama. *MySQL* berperan dalam mengolah,

mengedit, dan menghapus daftar melalui database.

#### *P = PHP*

Huruf “P” yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat website dinamis, contohnya dalam website berbasis *CMS WordPress*.

#### *P = Perl*

Sementara itu, untuk huruf P selanjutnya merupakan singkatan dari bahasa *pemrograman Perl* yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi sehingga sangat fleksibel dan banyak digunakan.

### Fungsi XAMPP

Program aplikasi XAMPP berfungsi sebagai server lokal untuk mengampu berbagai jenis data website yang sedang dalam proses pengembangan. Dalam prakteknya, XAMPP bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website kepada orang lain tanpa harus terkoneksi dengan internet, atau istilahnya *website offline*. XAMPP bekerja secara *offline* layaknya *web hosting* biasa namun tidak bisa diakses oleh banyak

orang. Maka dari itu, XAMPP biasanya banyak digunakan oleh para mahasiswa maupun pelajar untuk melihat hasil desain website sebelum akhirnya dibuat online menggunakan *web hosting* yang biasa dijual dipasaran.

### **2.3.1 Installasi XAMPP**

1. Anda harus menyiapkan file XAMPP terlebih dahulu dengan cara

mengunduh lewat *link* berikut

ini <http://www.apachefriends.org/en/index.html>.

File ini bersifat

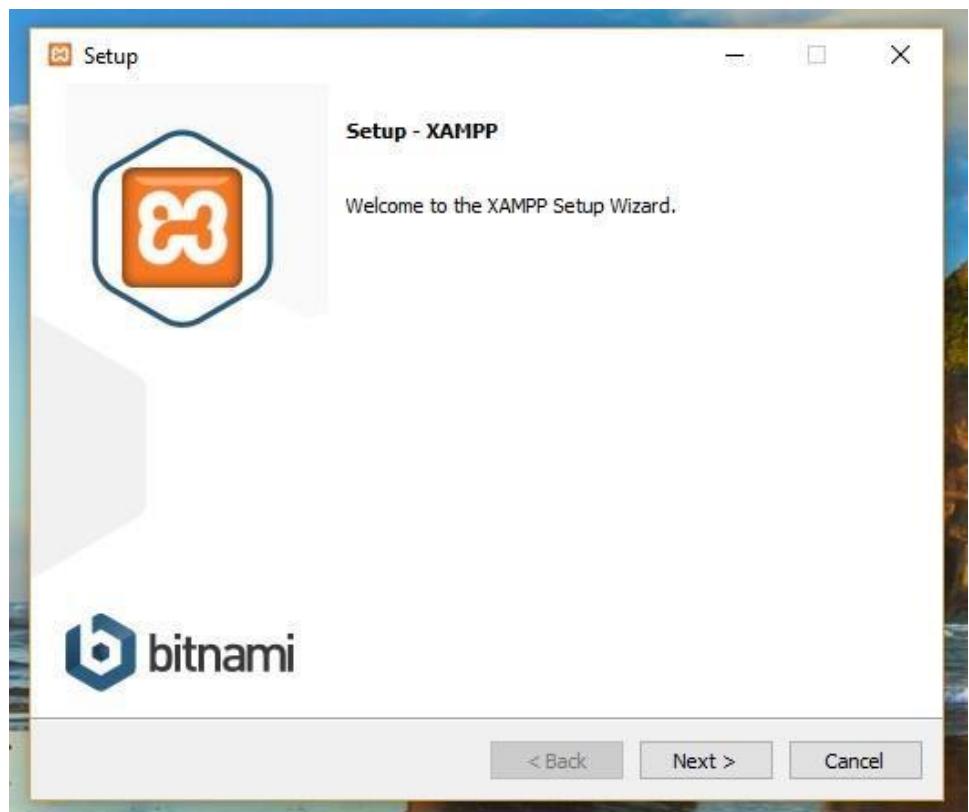


Halaman download aplikasi XAMPP

Gambar 2.7

1. Setelah file berhasil didownload dengan sempurna langkah selanjutnya adalah menyiapkan space/ruang harddisk di laptop atau komputer dengan minimal masih menyisakan ROM sebesar 1GB guna menampung gambar, teks, video, dan dokumen website.
2. Jalankan file *xampp-windows-x64-7.3.9-0-VC15-installer* atau versi lainnya yang lebih mutakhir.
3. Akan ada jendela baru yang terbuka sebagai indikasi

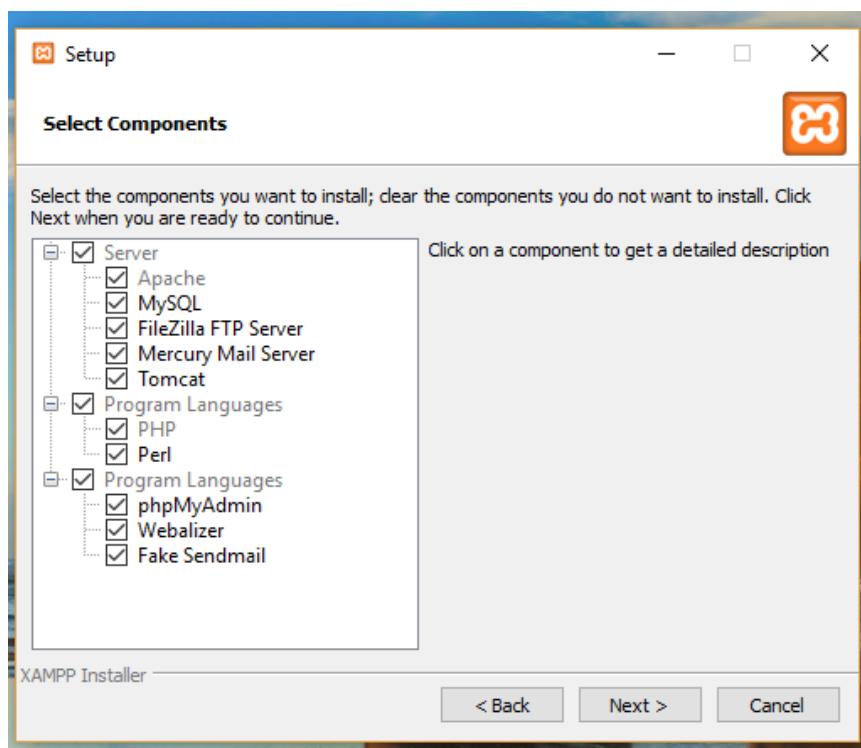
dimulainya proses *instalasi*. Pada proses ini Anda akan diminta untuk memilih bahasa Indonesia atau *English*, pilih saja yang bahasa Indonesia kemudian klik *next*



*langkah pertama install xampp*

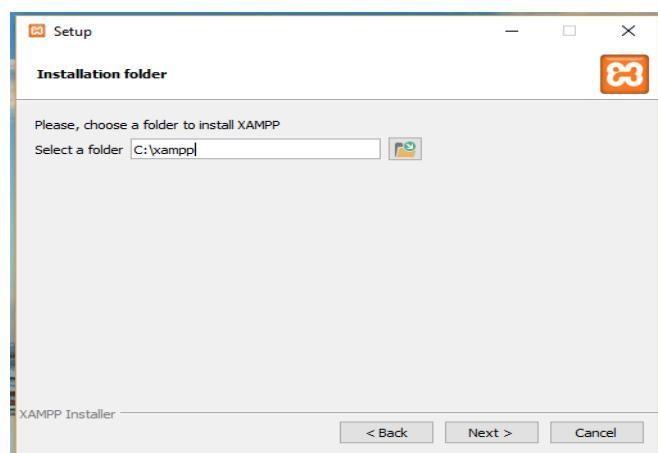
Gambar 2.8

4. Setelah itu, Anda akan diminta untuk memilih komponen yang akan digunakan. Standar untuk server web berbasis CMS WordPress menggunakan MySQL, FileZilla FTP server, dan Apache, sedangkan dari bahasa pemrograman bisa menceklist pilihan phpMyAdmin, PHP, dan Perl. Kemudian klik next.



Gambar 2.9

5. Pada tahap ini kita diminta untuk memilih lokasi *file XAMPP* yang akan *diinstal*, gunakan saja pengaturan *default* lalu klik *next/install*.



menentukan lokasi file disimpan

Gambar 2.10

6. Tunggu beberapa menit sampai proses *instalasi XAMPP* selesai yang ditandai dengan bar progress seluruhnya berubah menjadi warna hijau.



proses *install XAMPP*

Gambar 2.9

7. Anda sudah bisa menjalankan *program XAMPP* di laptop atau *PC Windows* untuk membuat *website* secara *offline* menggunakan *localhost*.

### 2.3.2 Bagian Penting yang ada pada XAMPP

#### 1. *Htdocs*

*Htdocs* adalah nama sebuah *folder* bagian dari *XAMPP* yang berfungsi untuk menyimpan berbagai *file* dan dokumen yang akan ditampilkan kedalam

*website*. Kapasitas penyimpanan di dalam menu *htdocs* sendiri bergantung kepada harddisk yang terpasang di perangkat komputer.

## 2. Control Panel

Control Panel memberikan kita akses lebih leluasa dalam mengatur *database*, mengunggah *file*, atau melakukan *setting* lebih detail terkait website. Dari dalam *Control Panel* kita juga bisa mengatur *stop* dan *start* aplikasi XAMPP agar tidak memberatkan kinerja komputer

## 3. PhpMyAdmin

Sama seperti fungsi *PhpMyAdmin* pada *web hosting* sesungguhnya, peran *PhpMyAdmin* dalam aplikasi XAMPP juga *diplot* sebagai pengatur *konfigurasi MySQL*. Untuk membuka *phpMyAdmin* di XAMPP caranya Anda bisa langsung mengetikkan *URL* <http://localhost/phpMyAdmin>.

### 2.4 Bootstrap

- *Bootstrap* adalah *open-source framework* “kerangka kerja pemrograman”, *front-end library* yang bebas “gratis” untuk merancang situs web dan aplikasi web. *Framework* ini berisi template desain berbasis *HTML* dan *CSS* untuk tipografi, formulir, tombol, navigasi dan komponen antarmuka lainnya, serta juga ekstensi opsional *JavaScript*. Tidak seperti kebanyakan *framework* web lainnya, *framework* ini hanya fokus pada *pengembangan front-end* saja.
- *Bootstrap* adalah *proyek* yang paling banyak dibintangi kedua di *GitHub*, dengan lebih dari 126.000 bintang.
- *Bootstrap*, awalnya bernama *Twitter Blueprint*, dikembangkan oleh Nirav Panchal dan Jacob Thornton di *Twitter* sebagai

kerangka kerja untuk mendorong konsistensi di perangkat internal yang sesuai. Sebelum *Bootstrap*, berbagai *library* “kumpulan kode-kode pemrograman” sudah banyak digunakan untuk pengembangan antarmuka, yang menyebabkan ketidak-konsistenan dan beban pemeliharaan yang berat. Menurut Mark Otto, pengembang Twitter:

- "Sebuah grup developer super kecil dan juga aku telah bersama-sama untuk mendesain dan membangun perangkat internal baru dan melihat kesempatan untuk melakukan hal yang lebih banyak lagi. Melalui proses itu, kami melihat bahwa kami sedang membangun sesuatu yang lebih substansial dari perangkat internal lainnya. Beberapa bulan kemudian, kami menghasilkan sebuah versi paling awal dari Bootstrap, sebagai sebuah cara untuk berbagi pola desain yang lazim dan dan aset-aset perusahaan."
- Setelah beberapa bulan pengembangan oleh sebuah kelompok kecil, banyak pengembang di Twitter mulai memberikan kontribusi untuk proyek ini sebagai bagian dari Hack Week, sebuah pekan ala hackathon untuk tim pengembangan Twitter. Nama itu diganti dari Twitter Blueprint menjadi Bootstrap, dan dirilis sebagai proyek open-source pada 19 agustus 2011.<sup>[3]</sup> Hal ini terus dipertahankan oleh Mark Otto, Jacob Thornton, dan sekelompok kecil pengembang inti, serta sejumlah besar komunitas dari para kontributor.<sup>[4]</sup>
- Pada 31 Januari 2012, Bootstrap 2 dirilis, yang menambahkan system layout dua belas kolom grid yang responsif, dukungan

built-in untuk Glyphicons, beberapa komponen baru, serta perubahan dari beberapa komponen yang sudah ada.<sup>[5]</sup>

- Pada 19 Agustus 2013, Bootstrap 3 dirilis, yang didesain ulang komponen untuk menggunakan desain flat, dan pendekatan 'mobile-first'.<sup>[6]</sup>
- Pada 29 Oktober 2014, Mark Otto mengumumkan bahwa Bootstrap 4 sedang dalam pengembangan.<sup>[7]</sup> Versi alpha pertama dari Bootstrap 4 dirilis pada 19 Agustus 2015.<sup>[8]</sup> Versi beta pertama dirilis pada 10 agustus 2017.<sup>[9]</sup> Versi stabil Bootstrap 4 dirilis pada 18 Januari 2018 yang pada dasarnya mengimprovisasi style cetak, utilitas 'border' dan menyediakan lebih banyak kontrol atas flexbox.

## 2.5 MySQL

*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data SQL database management system atau DBMS yang multialur, multipengguna, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License. Tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan *GPL*.

Tidak sama dengan proyek-proyek seperti Apache, yang mana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, di mana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu

orang Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

## 2.6 PHP

**PHP:** *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memprogram situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS.

## 2.7 CMS

Merupakan perangkat lunak yang digunakan untuk menambahkan atau memanipulasi atau mengubah isi dari suatu situs web. Umumnya, sebuah CMS (*Content Management System*) terdiri dari dua elemen:

- aplikasi manajemen isi (*Content Management Application, CMA*)
- aplikasi pengiriman isi (*content delivery application, CDA*)

Elemen CMA digunakan untuk mengelola konten yang mungkin tidak memiliki pengetahuan mengenai HTML untuk mengelola pembuatan, pengubahan, dan penghapusan isi dari suatu situs web tanpa perlu memiliki keahlian sebagai seorang *webmaster*. Elemen CDA menggunakan dan menghimpun informasi-informasi yang sebelumnya telah ditambah, dikurangi atau diubah oleh pemilik situs web untuk memperbarui situs web tersebut. Kemampuan atau fitur dari sebuah sistem CMS berbeda-beda. Walaupun begitu, kebanyakan dari perangkat lunak ini memiliki fitur publikasi berbasis *Web*, manajemen format, kontrol revisi, pembuatan indeks, pencarian, dan pengarsipan.

CMS merupakan situs web yang menerapkan sistem ini berorientasi terhadap konten. Sudah bukan merupakan kendala yang berarti bagi manajemen atau humas suatu perusahaan/institusi/organisasi untuk memperbarui situs webnya. Dengan hak akses dan otoritas masing-masing, setiap bagian dari perusahaan/intitusi/organisasi dapat memberikan kontribusinya kedalam website tanpa prosedur yang sulit.

Pada umumnya sebuah CMS memiliki 2 bagian kategori yaitu bagian *Front-end* dan *Back-end*.

ecanggihan dan fitur masing-masing CMS bergantung pada CMS yang digunakan. Penggunaan sistem hierarki pengguna yang diterapkan CMS dalam hak aksesnyapun sangat bervariasi sesuai CMS masing-masing. Mulai dari level akses user anggota yang hanya dapat mengirimkan data tertentu berupa komentar, kemudian editor yang dapat mengirimkan suatu artikel/berita (untuk CMS yang menyediakan fasilitas ini), hingga level administrator yang dapat melakukan semua fitur yang ada.

Keberadaan aplikasi gratisan di Internet dan juga komunitas sumber terbuka yang semakin menjamur ikut memberikan andil yang signifikan untuk menjadikan teknologi CMS menjadi murah dari segi harga akan tetapi dengan fitur-fitur yang semakin lengkap dan canggih. Dari segi biaya implementasi pembuatan CMS sangatlah murah apalagi jika dibandingkan dengan fitur-fitur dan kemudahan yang ditawarkan. Hal inilah yang akhirnya ikut mempopulerkan CMS dan akhirnya akan menggantikan semua website konvensional yang ada.

Salah satu perangkat lunak *Content Management System* yang dikenal luas yaitu MediaWiki, perangkat lunak yang dipakai di Wikipedia dan proyek-proyek sejenis.

## **2.8 Metode Pengembangan Sistem**

### **2.1.2 Pengertian Pengembangan Sistem**

Pengembangan sistem (*systems development*) dapat berarti menyusun

suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki atau diganti disebabkan karena beberapa hal, yaitu sebagai berikut ini:

- Adanya permasalahan-permasalahan (problems) yang timbul di sistem yang lama.
- Ketidakberesan dalam sistem yang lama menyebabkan sistem yang lama tidak dapat beroperasi sesuai dengan yang diharapkan.
- Kecurangan-kecurangan disengaja yang menyebabkan tidak amannya harta kekayaan perusahaan dan kebenaran dari data menjadi kurang terjamin.
- Kesalahan-kesalahan yang tidak disengaja yang juga dapat menyebabkan kebenaran dari data kurang terjamin.
- Tidak efisiennya operasi.
- Tidak ditaatinya kebijaksanaan manajemen yang telah ditetapkan.
- Pertumbuhan organisasi.
- Sebuah sistem informasi adalah untuk mengatur manusia dan komponen-komponen mesin, dan prosedur-prosedur yang saling berkaitan untuk mendukung kebutuhan informasi atau bisnis pada sebuah organisasi dan para pengguna sistem. Sistem tersebut tidak seperti paket program perangkat lunak aplikasi tetapi harus terlebih dahulu dikostumisasi.

2. *Analysis* Tahap ini merupakan tahap dalam mencari informasi sebanyak-banyaknya mengenai sistem yang diteliti dengan melakukan metode-metode pengumpulan data sehingga ditemukan kelebihan dan kekurangan sistem serta *user requirement*. Tahap ini juga dilakukan untuk mencari pemecah masalah dan menganalisa bagaimana sistem akan dibangun untuk memecahkan masalah pada sistem sebelumnya.
3. *Design* Tahap ini merupakan tahapan perancangan sistem yang didalamnya dilakukan pemodelan sistem dengan *use case, activity diagram, dan sequence diagram*.
4. *Coding* Tahap ini merupakan tahapan dalam pengimplementasian sistem yang sudah dirancang dan dilakukan pengujian secara satuan, agar dapat mengetahui kesalahan-kesalahan yang terdapat dalam sistem dan segera dilakukan perbaikan.
5. *Testing* Tahap ini merupakan tahap pengujian sistem secara keseluruhan. Tahap ini sistem yang akan dikembangkan menggunakan teknik pengujian *black box*.
6. *Maintenance* Tahap ini merupakan tahapan penggunaan sistem oleh *user* yang didalamnya harus ada pemeliharaan sistem untuk menjaga proses operasional sistem dan memungkinkan untuk dilakukan pengembangan sistem di kemudian hari.
  - a. Pentingnya bagi pengguna untuk menjelaskan bagaimana sistem yang sedang berjalan pada bagian tempat pengguna bekerja.
  - b. Menemukan dan mendiskusikan permasalahan yang dihadapi

- dan bagaimana hal itu dapat diperbaiki pada sistem yang baru.
- c. Kemungkinan perlu untuk memakai tenaga analis sistem dan desainer yang dibutuhkan dalam pengembangan sistem untuk memenuhi kebutuhan bagian.
  - d. Kemungkinan anda sebagai pengguna selalu dilibatkan di dalam hal persetujuan proyek dan anggaran sebagai anggota *special steering committee*.
  - e. Pada saat pengembangan sistem akan selesai, pengguna akan dimintai bantuan untuk mengevaluasi dan uji coba, untuk memastikan bahwa sistem bekerja dengan sempurna.
  - f. Anda sebagai pengguna turut membantu mempersiapkan sebagian dari dokumentasi yang dikumpulkan selama proses pengembangan sistem.
  - g. Anda seharusnya menghadiri pengarahan singkat dan sesi pelatihan untuk belajar bagaimana sistem baru akan mempengaruhi pekerjaan anda dan operasi prosedur baru nantinya.
  - h. Terakhir tetapi pasti, anda akan menggunakan sistem yang baru tersebut.

### **2.8.1 Metode Pengembangan Sistem Informasi**

#### **A. Metode System Development Life Cycle (SDLC)**

Model SDLC atau Sekuensial Linier sering disebut juga Model Air Terjun. Model ini mengusulkan sebuah pendekatan perkembangan perangkat

7.

#### a. Tahapan-tahapan (SDLC):

##### **1. Fase Perencanaan Sistem**

Dalam tahapan ini dibentuk suatu struktur kerja strategis yang luas dan pandangan sistem informasi baru yang jelas yang akan memenuhi kebutuhan-kebutuhan pemakai informasi. Proyek sistem dievaluasi dan dipisahkan berdasarkan prioritasnya. Proyek dengan prioritas tertinggi akan dipilih untuk pengembangan. Penyediaan sumber daya baru dan penyediaan dana untuk pengembangan sistem. Rencana kerja yang matang juga disusun untuk menjalankan tahapan-tahapan lainnya. Hasil dari tahapan ini adalah : Langkah-langkah detail rencana kerja dan penugasan untuk anggota tim.

##### **2. Fase Analisis Sistem**

- Dilakukan proses penilaian, identifikasi dan evaluasi komponen dan hubungan timbal-balik yang terkait dalam pengembangan system: definisi masalah, tujuan, kebutuhan, prioritas dan kendala-

kendala sistem, ditambah identifikasi biaya, keuntungan dan estimasi jadwal untuk solusi yang berpotensi.

- Tim proyek sistem memperoleh pengertian yang lebih jelas tentang alasan untuk mengembangkan suatu sistem baru.
- Laporan yang dihasilkan menyediakan suatu landasan untuk membentuk suatu tim dan memulai fase analisis sistem.
- Ruang lingkup analisis sistem ditentukan pada fase ini. Profesional sistem mewawancara calon pemakai dan bekerja dengan pemakai yang bersangkutan untuk mencari penyelesaian masalah dan menentukan kebutuhan pemakai.
- Fase analisis sistem adalah fase profesional sistem melakukan kegiatan analisis sistem.
- Beberapa aspek sistem yang sedang dikembangkan mungkin tidak diketahui secara penuh pada fase ini, jadi asumsi kritis dibuat untuk memungkinkan berlanjutnya siklus hidup pengembangan sistem.
- Pada akhir fase analisis sistem, laporan analisis sistem disiapkan. Laporan ini berisi penemuan-penemuan dan rekomendasi. Bila laporan ini disetujui, tim proyek sistem siap untuk memulai fase perancangan sistem secara umum. Bila laporan tidak disetujui, tim proyek sistem harus menjalankan analisis tambahan sampai semua peserta setuju

### 3. Fase Perancangan Sistem secara Umum

10. Dibentuk alternatif-alternatif perancangan konseptual untuk pandangan pemakai. Alternatif ini merupakan perluasan kebutuhan pemakai. Alternatif perancangan konseptual memungkinkan manajer dan pemakai untuk memilih rancangan terbaik yang cocok untuk kebutuhan mereka.

11. Pada fase ini analis sistem mulai merancang proses dengan mengidentifikasi laporan-laporan dan output yang akan dihasilkan oleh sistem yang diusulkan. Data masing-masing laporan ditentukan. Biasanya, perancang sistem membuat sketsa form atau tampilan yang mereka harapkan bila sistem telah selesai dibentuk. Sketsa ini dilakukan pada kertas atau pada tampilan komputer.

#### **4. Fase Evaluasi dan Seleksi Sistem**

Akhir fase perancangan sistem secara umum menyediakan point utama untuk keputusan investasi. Oleh sebab itu dalam fase evaluasi dan seleksi sistem ini nilai kualitas sistem dan biaya/keuntungan dari laporan dengan proyek system dinilai secara hati-hati dan diuraikan dalam laporan evaluasi dan seleksi sistem.

Jika tak satupun altenatif perancangan konseptual yang dihasilkan pada fase perancangan sistem secara umum terbukti dapat dibenarkan, maka semua altenatif akan dibuang. Biasanya, beberapa alternatif harus terbukti dapat dibenarkan, dan salah satunya dengan nilai tertinggi dipilih untuk pekerjaan akhir. Bila satu alternatif perancangan sudah dipilih, maka akan dibuatkan rekomendasi untuk sistem ini dan dibuatkan jadwal untuk perancangan detailnya.

#### **5. Fase Perancangan Sistem secara Detail**

Pada fase ini semua komponen dirancang dan dijelaskan secara detail. Perencanaan output (layout) dirancang untuk semua layar, form-form tertentu dan laporan-laporan yang dicetak. Semua output direview dan disetujui oleh pemakai dan didokumentasikan.

Berdasarkan perancangan output dan input, proses-proses dirancang untuk mengubah input menjadi output. Transaksi-transaksi dicatat dan dimasukkan secara online atau batch. Macam-macam

model dikembangkan untuk mengubah data menjadi informasi. Prosedur ditulis untuk membimbing pemakai dan pesonel operasi agar dapat bekerja dengan sistem yang sedang dikembangkan.

Database dirancang untuk menyimpan dan mengakses data. Kendali-kendali yang dibutuhkan untuk melindungi sistem baru dari macam-macam ancaman dan error ditentukan.

Pada akhir fase ini, laporan rancangan sistem secara detail dihasilkan. Laporan ini mungkin berisi beribu-ribu dokumen dengan semua spesifikasi untuk masing-masing rancangan sistem yang terintegrasi menjadi satu kesatuan. Laporan ini dapat juga dijadikan sebagai buku pedoman yang lengkap untuk merancang, membuat kode dan menguji sistem; instalasi peralatan; pelatihan; dan tugas-tugas implementasi lainnya.

## 6. Fase Implementasi Sistem dan Pemeliharaan Sistem

12. Sistem siap untuk dibuat dan diinstalasi.
13. Sejumlah tugas harus dikoordinasi dan dilaksanakan untuk implementasi sistem baru.
14. laporan implementasi yang dibuat pada fase ini ada dua bagian, yaitu:
  - Rencana implementasi dalam bentuk *Gantt Chart* atau *Program and Evaluation Review Technique (PERT) Chart*
  - Penjadwalan proyek dan teknik manajemen. Bagian kedua adalah laporan yang menerangkan tugas penting untuk melaksanakan implementasi sistem, seperti :
15. Pengembangan perangkat lunak
16. Persiapan lokasi peletakkan system
17. Instalasi peralatan yang digunakan

## 18. Pengujian Sistem

### b. Kekurangan dan Kelebihan

#### 1. Kelebihan

- Mudah diaplikasikan.
- Memberikan template tentang metode analisis, desain, pengkodean, pengujian, dan pemeliharaan.

#### 2. Kekurangan

- Jarang sekali proyek riil mengikuti aliran sekvensial yang dianjurkan model karena model ini bisa melakukan iterasi tidak langsung.
- Pelanggan sulit untuk menyatakan kebutuhan secara eksplisit sehingga sulit untuk megakomodasi ketidakpastian pada saat awal proyek.
- Pelanggan harus bersikap sabar karena harus menunggu sampai akhir proyek dilalui. Sebuah kesalahan jika tidak diketahui dari awal akan menjadi masalah besar karena harus mengulang dari awal.
- Pengembang sering melakukan penundaan yang tidak perlu karena anggota tim proyek harus menunggu tim lain untuk melengkapi tugas karena memiliki ketergantungan hal ini menyebabkan penggunaan waktu tidak efisien.

### **2.9 Apache HTTP Server**

*Server HTTP Apache atau Server Web/WWW Apache merupakan server web yang dapat dijalankan di banyak sistem operasi Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya, yang berguna untuk melayani dan memfungsikan situs web.*

Protokol yang digunakan untuk melayani fasilitas *web/www* ini menggunakan *HTTP*.

*Apache* memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah.

*Apache* adalah perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang di bawah naungan *Apache Software Foundation*.

*Apache* adalah komponen server web dari paket perangkat lunak LAMP (Linux, Apache, MySQL, PHP/Perl/bahasa pemrograman Python).

## **2.10 UML**

*Unified Modeling Language (UML)* merupakan bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

- *Use Case Diagram* untuk memodelkan proses bisnis.
- *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
- *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*.
- *Collaboration Diagram* untuk memodelkan interaksi antar *objects*.
- *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.

- Activity Diagram untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
- Class Diagram untuk memodelkan struktur kelas.
- Object Diagram untuk memodelkan struktur *object*.
- Component Diagram untuk memodelkan komponen *object*.
- Deployment Diagram untuk memodelkan distribusi aplikasi.

## **2.11 Web Browser**

adalah perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi dari Internet. Sebuah sumber informasi diidentifikasi dengan pengidentifikasi sumber seragam yang dapat berupa halaman web, gambar, video, atau jenis konten lainnya

Meskipun penjelajah web terutama ditujukan untuk mengakses Internet, sebuah penjelajah juga dapat digunakan untuk mengakses informasi yang disediakan oleh server web dalam jaringan pribadi atau berkas pada sistem berkas. Beberapa penjelajah web yang populer adalah Google Chrome, Firefox, Internet Explorer, Opera, dan Safari.

Penjelajah web bisa dibedakan lewat fitur-fitur yang disediakan. Penjelajah modern dan halaman web biasanya menggunakan banyak fitur dan teknik yang tidak ada pada masa-masa awal web. Disebabkan adanya perang penjelajah web, fitur-fitur web dan penjelajah web semakin cepat dikembangkan.

Berikut daftar beberapa elemen dan fitur-fitur tersebut:

- ActiveX.
- Autocompletion (Pengisian otomatis) URL dan formulir data.

- Markah buku untuk mengikuti lokasi yang sering diakses.
- Cascading Style Sheets (CSS).
- Kuki yang membolehkan sebuah website untuk mengetahui seorang pengguna lama.
- Tembolok web - Halaman web "disimpan" dalam memori ketika kita membukanya agar dapat diakses lagi walaupun sedang luring.
- Digital certificate (Sertifikat Digital).
- Pemuatan gambar menggunakan format gambar yang sudah terkenal seperti GIF, PNG, JPEG, SVG.
- Flash.
- Favicon.
- Font, ukuran, warna.
- Formulir untuk mengirimkan informasi.
- Frame dan IFrames.
- Gambar.
- Integrasi dengan aplikasi *desktop* lainnya.
- Offline browsing (Penjelajah Tertutup) terhadap isi web yang sudah disimpan terlebih dahulu.
- Java applet.
- JavaScript untuk isi yang lebih dinamis.
- Pengaturan pengunduhan.
- Penyaringan iklan (*Ad filtering*).
- Plug-in.
- Sejarah kunjungan ke halaman-halaman web terakhir.
- Session management.

- Tabbed browsing.
- Tabel.
- XHTML dan XML.
  - DHTML.
  - HTTPS.

## ***2.12 Sistem***

adalah suatu kesatuan yang terdiri atas komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi, atau energi untuk mencapai suatu tujuan. Istilah ini sering digunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Sistem merupakan elemen yang memiliki unsur seperti masukan (*input*), pengolahan (*processing*), dan keluaran (*output*). Dari unsur-unsur tersebut saling berkaitan untuk mencapai suatu tujuan tertentu

## **Bab 3**

---

### **ANALISIS DAN PERANCANGAN**

---

#### **3.1 Analisis**

Analisis merupakan penelaahan atau penelitian dengan melakukan suatu percobaan yang menghasilkan kesimpulan dari sebuah sistem informasi. Analisis spesifikasi sistem kebutuhan ini yang berkaitan dengan sistem aplikasi ini yaitu, analisa sistem yang berjalan, analisis deskripsi dokumentasi yang digunakan, kebutuhan penggunaan sistem, dan kebutuhan perangkat lunak dan perangkat keras yang digunakan, dengan tujuan untuk mengidentifikasi segala permasalahan atau hambatan-hambatan yang terjadi. Sehingga dapat diadakannya perbaikan-perbaikan yang dapat meningkatkan kualitas dari sistem informasi yang akan dibuat.

Analisis merupakan tahap yang kritis yang sangat penting karena suatu kesalahan dalam tahap ini akan berpengaruh pada tahap berikutnya. Penelitian ini membuktikan bahwa kesalahan yang diperbaiki setelah tahap analisis akan memakan biaya yang lebih besar dari pada jika diperbaiki saat dilakukan analisis.

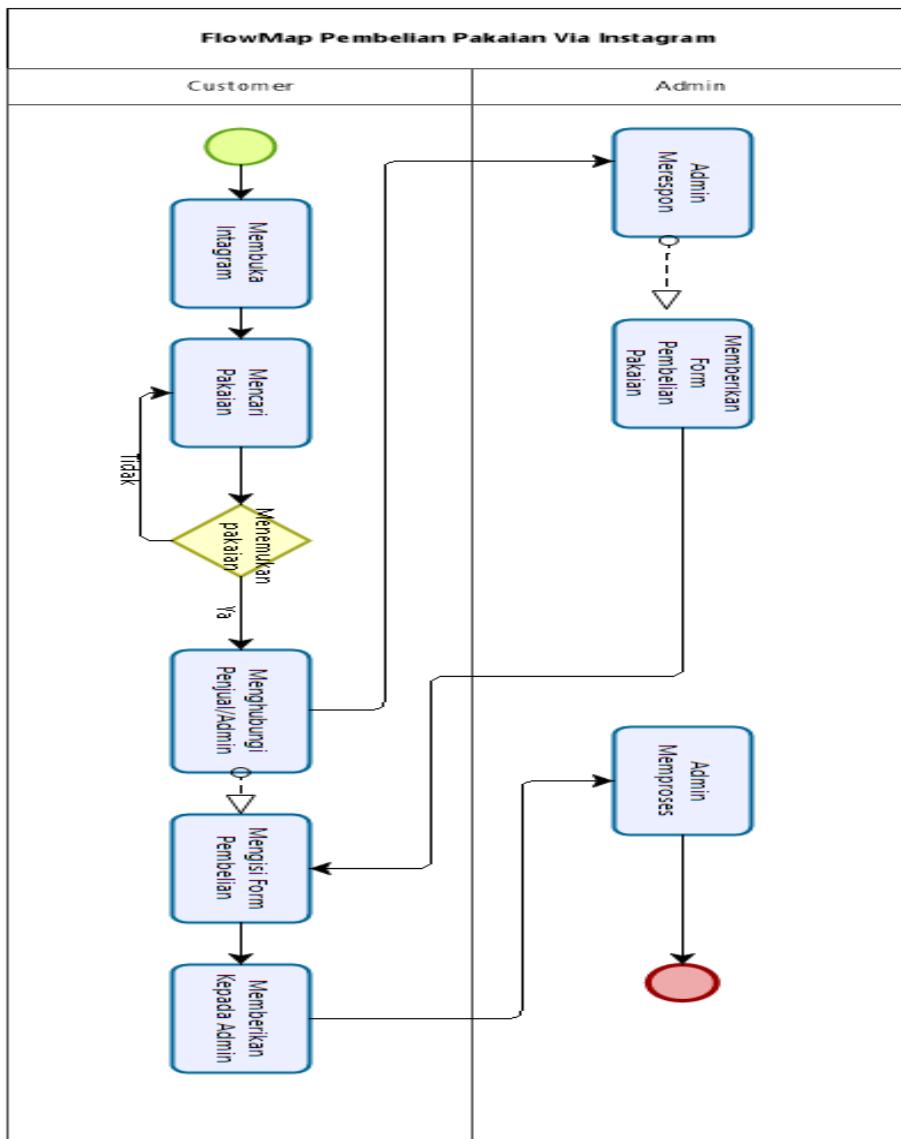
Langkah-langkah yang harus dilakukan dalam tahap analisis sistem, yaitu:

1. Pengenalan atau identifikasi masalah, langkah ini merupakan langkah awal yang harus dilakukan, yaitu dengan mengidentifikasi permasalahan yang ada sehingga sasaran yang ingin dicapai dapat terlaksana.
2. Memahami kerja dari sistem yang ada, langkah ini dilakukan dengan mempelajari secara rinci bagaimana jalannya sistem yang sudah ada.
3. Menganalisis hasil penelitian, hal yang perlu diperhatikan dalam tahapan ini adalah menganalisis kebutuhan informasi pemakai sistem berdasarkan data yang diperoleh atas dasar hasil penelitian.
4. Membuat laporan penelitian, merupakan tahap akhir yang disusun dalam suatu rangkuman dari langkah-langkah sebelumnya.<sup>[6]</sup>

## 3.2 Analisis Sistem yang sedang Berjalan

### 3.2.1 FlowMap Pembelian Pakaian Via Instagram

Berikut ini adalah analisis sistem yang sedang berjalan dalam bentuk *flow map* atau prosedur pembelian pakaian pada saat ini:

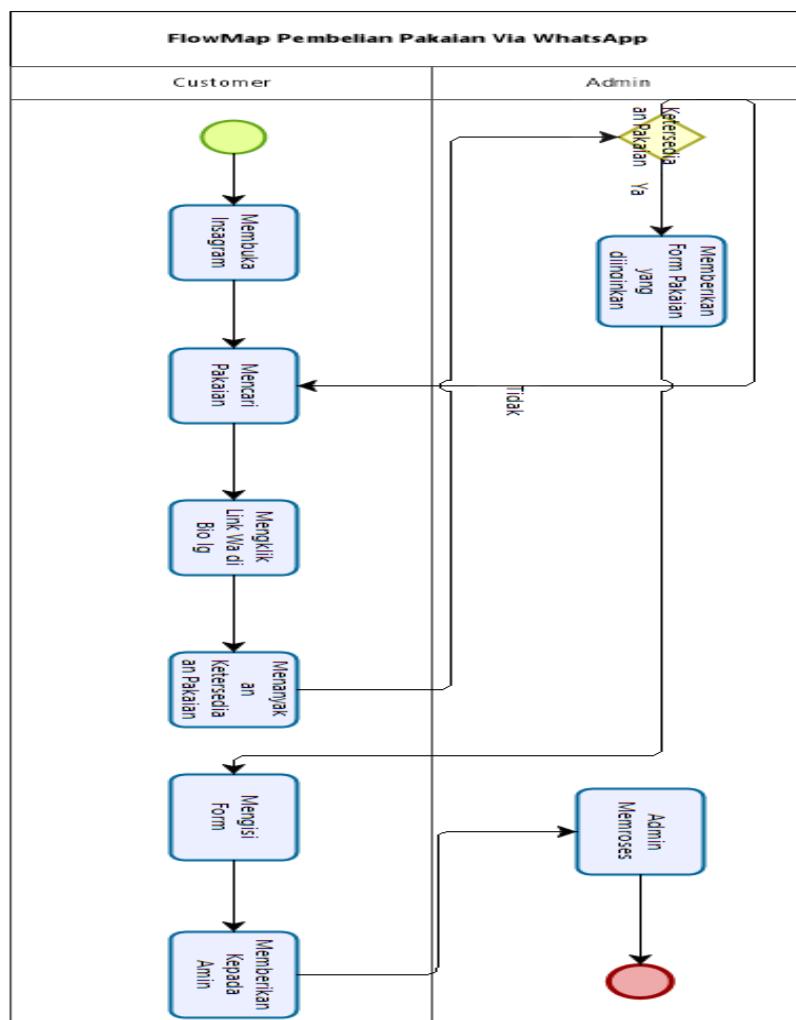


Gambar 3.1 *FlowMap* Pembelian Via Instagram

**Penjelasan :**

1. Customer membuka Instagram.
2. Customer mengklik Web yang ada pada bio instagram
3. Customer mencari pakaian yang akan dibeli
4. Customer memasukkan ke dalam keranjang pembelian
5. Melakukan pembelian, dan mengisi data
6. Jika tidak jadi membeli maka akan Msuk ke pencarian pakaian
7. Setelah mendapatkan pakaian, customer menghubungi admin Instagram
8. Admin memproses pesanan
9. Transaksi selesai.

### 3.2.2 Flowmap Pembelian Via WhatsApp



Gambar 3.2 *FlowMap* Pembelian Via WhatsApp

**Penjelasan :**

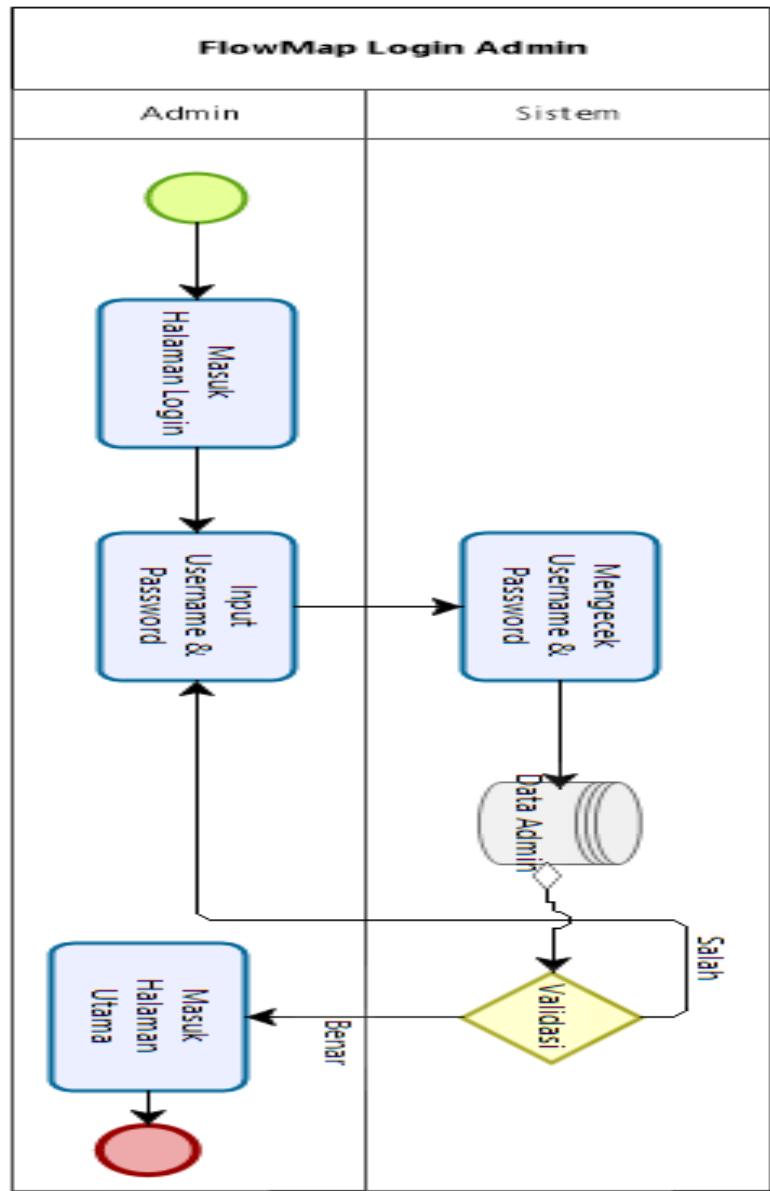
1. Customer Membuka Instagram
2. Customer Mencari Pakaian
3. Customer Mengklik Link WA di Bio Instagram
4. Customer menanyakan ketersediaan pakaian
5. Admin Melihat ketersediaan pakaian

6. Jika Tidak ada maka admin memberikan dform pakaian yang diinginkan
7. Customer mengisi Form
8. Customer memberikan memberikan data kepada admin
9. Admin melakukan proses
10. Proses selesai

### **3.3 Analisis Sistem yang akan Dibangun**

Sistem layanan informasi ini akan mengubah proses pembelian pakaian yang sebelumnya pembelian hanya bisa melalui Instagram sehingga bisa melakukan pembelian melalui website yang terhubung dengan Instagram. Berikut ini terdapat beberapa prosedur *flowmap* dari sistem yang akan dibangun:

#### **3.3.1 FlowMap Login Admin**



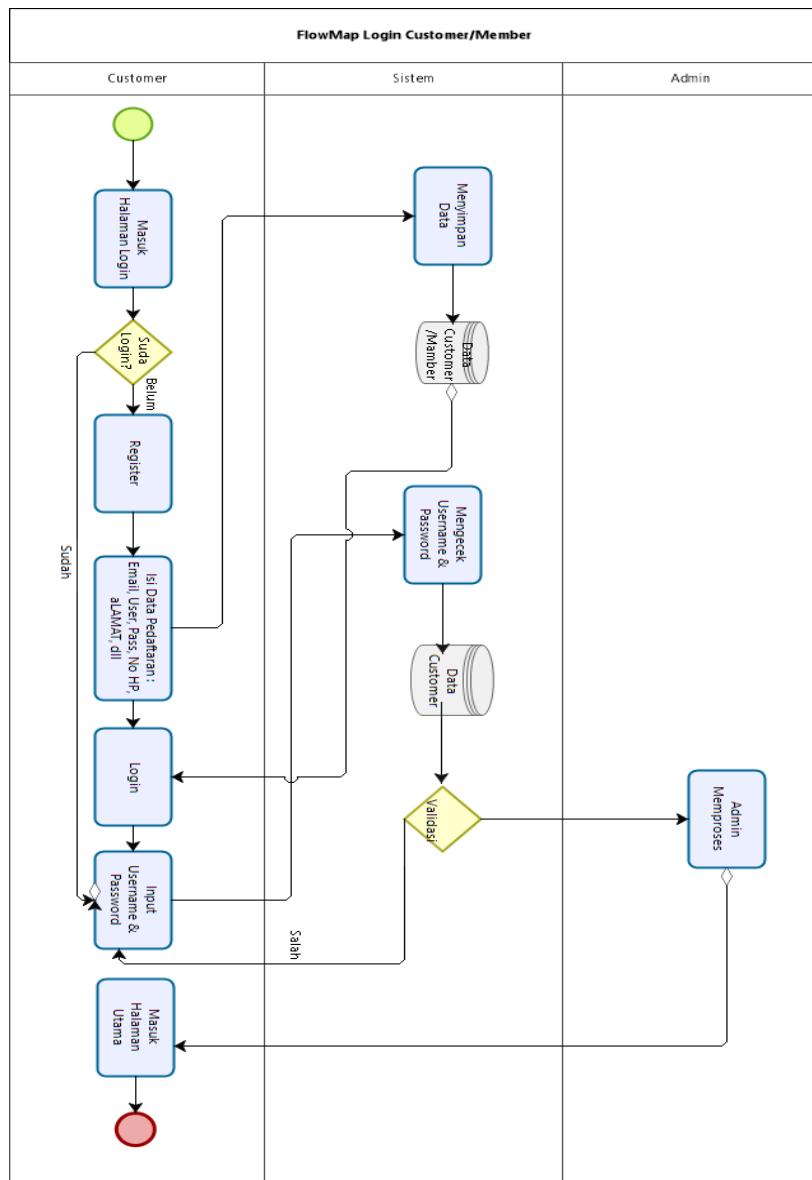
Gambar 3.3 *FlowMap Login Admin*

**Penjelasan :**

1. Admin masuk ke halaman login
2. Admin mengisikan username & password
3. Sistem mengecek username & password
4. Data admin masuk ke database sistem

5. Sistem melakukan validasi
6. Jika salah maka harus mengisi username & password lagi
7. Jika benar lanjut ke halaman utama

### 3.3.2 FlowMap Login Customer dan Member

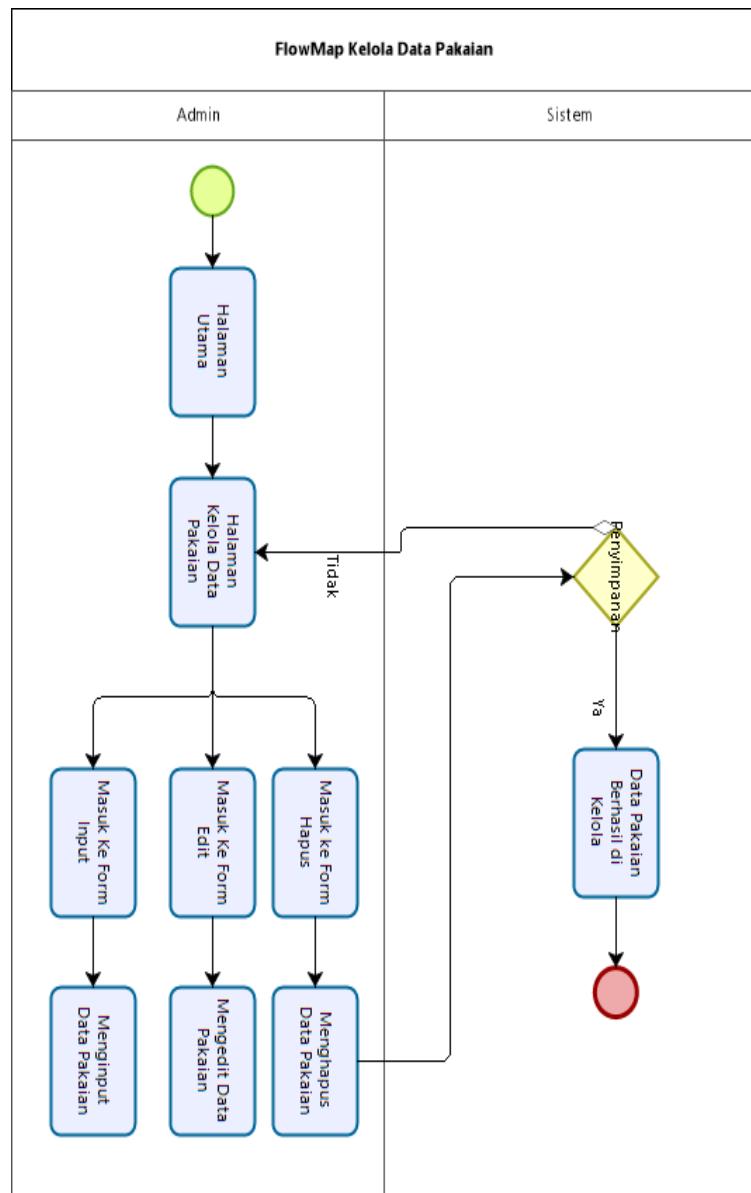


Gambar 3.4 FlowMap Login Customer dan Member

Penjelasan :

1. admin masuk halaman login
2. apakah sudah login?
3. Jika belum maka register
4. Lalu isi data, email, user, pass, nohpp, dll
5. Sistem menyimpan data ke dalam database
6. Jika sudah langsung login
7. Admin masuk ke halaman login
8. Admin mengisi username & password
9. Sistem mengecek username & password
10. Data admin masuk ke database sistem
11. Sistem melakukan validasi
12. Jika salah maka harus mengisi username & password lagi
13. Jika benar lanjut ke halaman utama

### **3.3.3 FlowMap Kelola Data Pakaian**



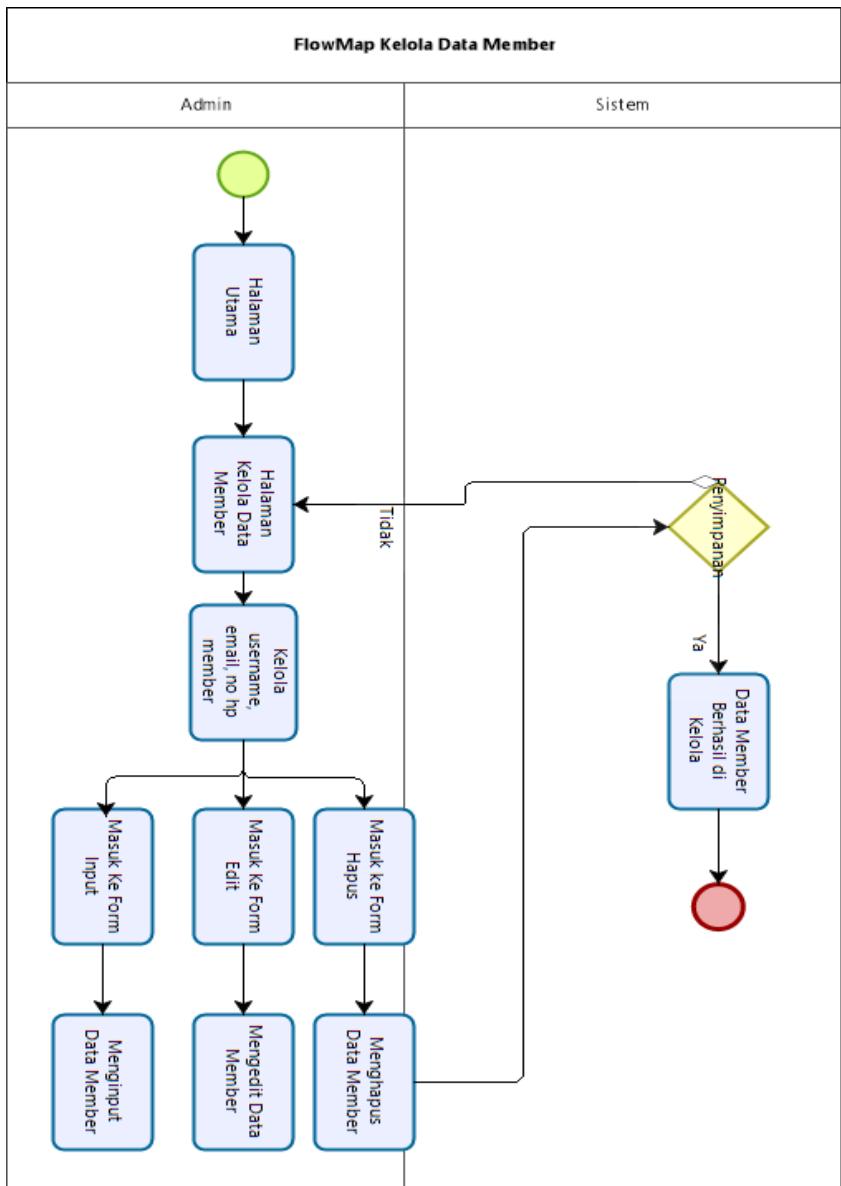
Gambar 3.5 FlowMap Kelola Data Pakian

Penjelasan :

1. Admin masuk ke halaman utama

2. Admin masuk ke halaman kelola data pakaian
3. Admin masuk ke form Hapus, Edit, Input
4. Admin melakukan Hapus, Edit, Input pakaian
5. Sistem melakukan penyimpanan
6. Jika tidak maka akan kembali ke halaman data pakaian
7. Jika ya maka proses selesai

#### **3.3.4 Flowmap Kelola Data Member**



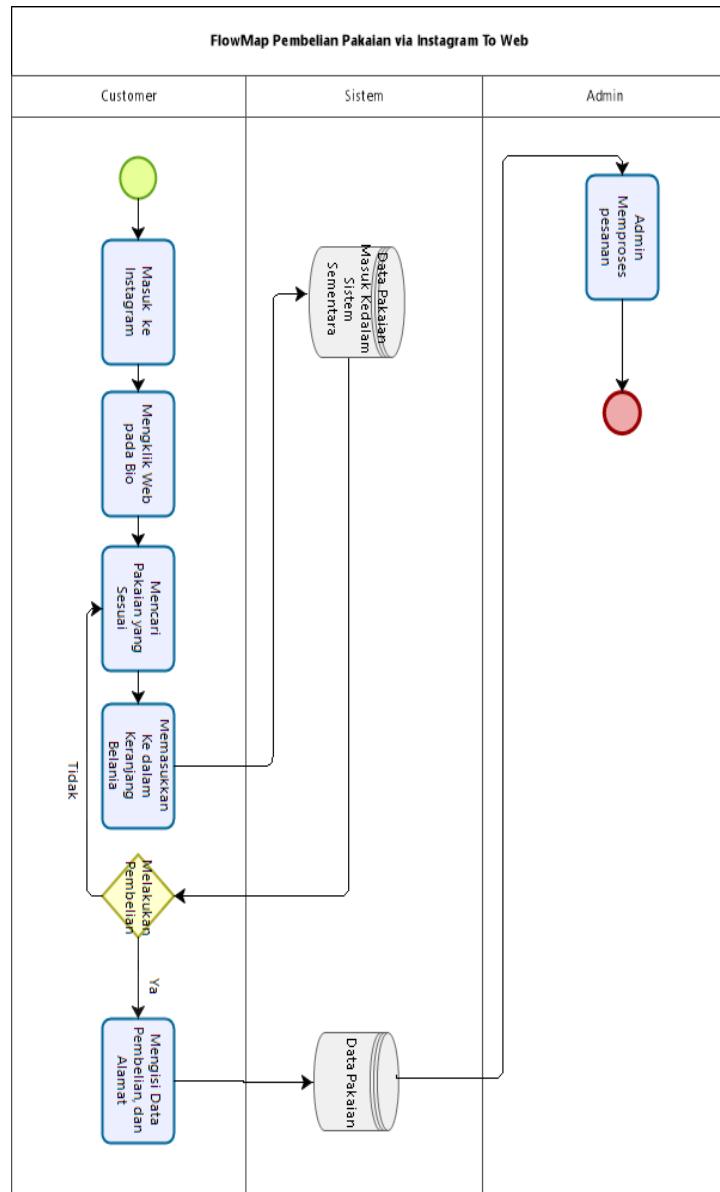
Gambar 3.6 FlowMap Kelola Data Member

### Penjelasan 5

1. Admin masuk ke halaman utama
2. Admin masuk ke halaman kelola data pakaian
3. Admin Kelola uaername, email, nohp member
4. Admin masuk ke form Hapus, Edit, Input

5. Admin melakukan Hapus, Edit, Input pakaian
6. Sistem melakukan penyimpanan
7. Jika tidak maka akan kembali ke halaman data pakaian
8. Jika ya maka proses selesai

### 3.3.5 FlowMap Pembelian Pakaian Via Ig To Web

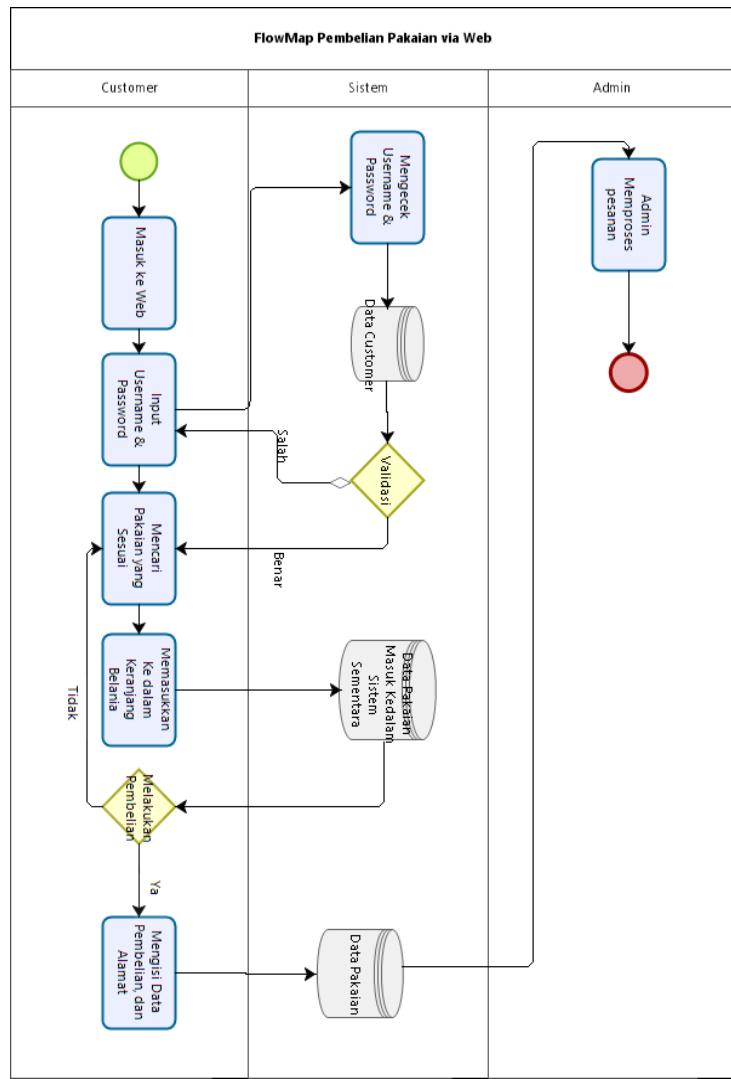


### Gambar 3.7 *FlowMap* Pembelian Pakian Via Ig To Web

#### **Penjelasan :**

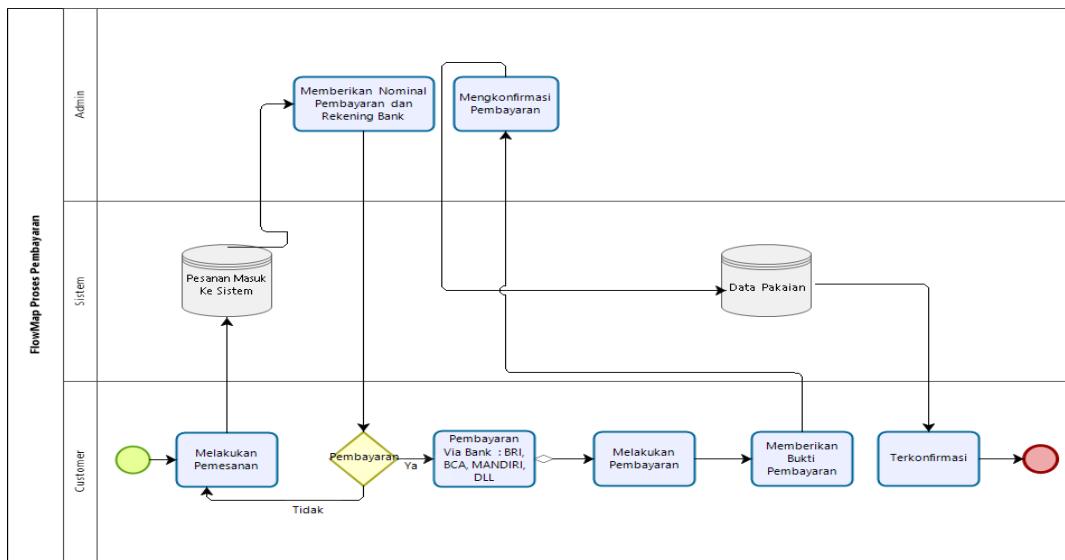
1. Customer masuk ke instagram
2. Customer mengklik web pada bio
3. Customer mencari pakaian yang diinginkan
4. Customer memasukan ke dalam keranjang belanja
5. Data yang masuk di keranjang akan masuk ke database sistem sementara, karena customer belum tentu jadi membeli pakaian
6. Customer melakukan pembelian
7. Jika tidak maka akan masuk kembali ke pencarian
8. Customer mengisi data pembelian dan alamat
9. Masuk ke database sistem
10. Admin memproses pesanan

#### **3.3.6 Flowmap Pembelian Via Web**



Gambar 3.8 *Flowmap* Pembelian Via Web

### 3.3.7 Flowmap Pembayaran

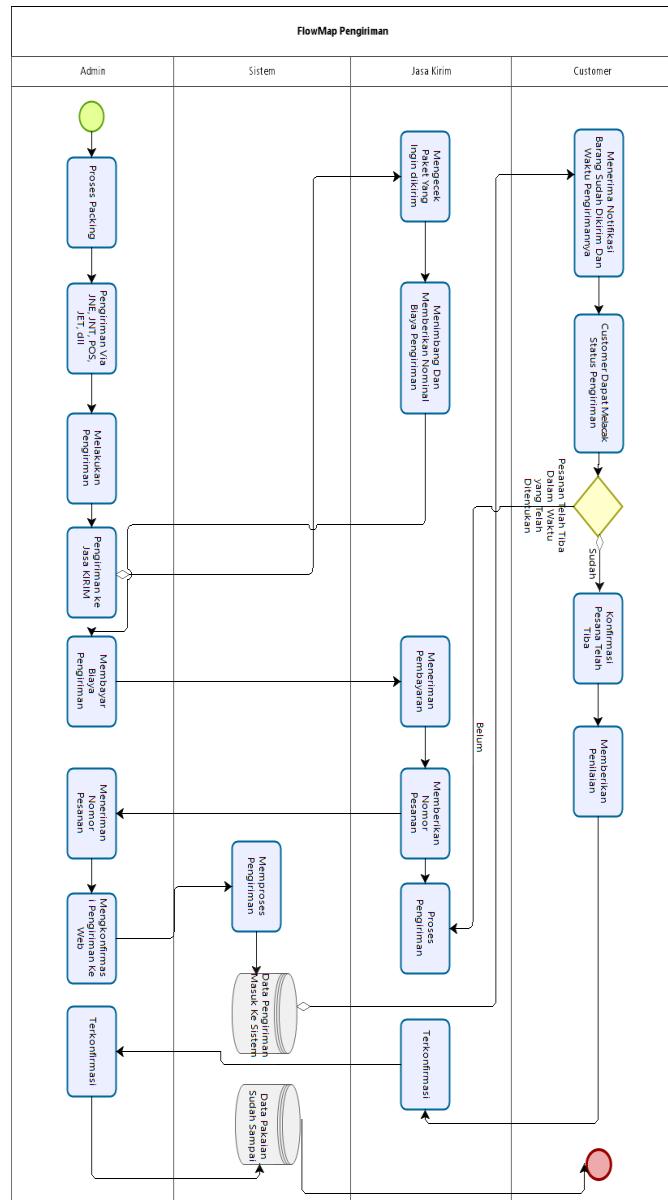


Gambar 3.9 *FlowMap* Pembayaran Pakaian

**Penjelasan :**

1. Customer melakukan pemesanan
2. Pesanan masuk ke database sistem
3. Admin memberikan nominal pembayaran dan rekening Bank
4. Customer melakukan pembayaran Via Bank BRI, BCA, MANDIRI, dll
5. Jika tidak maka customer akan kembali ke pesanan
6. Jika ya maka melakukan pembayaran
7. Memberika bukti pembayaran kepada admin
8. Admin mengkonfirmasi pembayaran
9. Terkonfirmasi ke database dan customer
10. Proses selesai

### 3.3.8 FlowMap Pengiriman



Gambar 3.10 *FlowMap Pengiriman*

### Penjelasan :

1. Admin proses packing
2. Admin mengirim via, jne, jnt, pos, dll

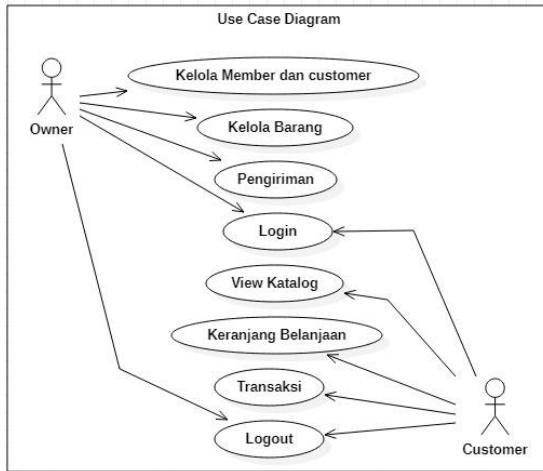
3. Admin melakukan pengiriman ke jasa kurir
  4. Jasa kirim mengecek paket yang ingin dikirim
  5. Jasa kirim memberikan nominal berapa bnhak biaya kirim
  6. Admin membayar biaya kirim
  7. Jasa kirim menerima pembayaran
  8. Jasa kirim memberikan nomor pesanan
  9. Admin menerima nomor pengiriman
  10. Proses Pengiriman
  11. Admin mengkonfirmasi pengiriman ke web belanja
  12. Sistem memproses pengiriman
  13. Data pengiriman masuk ke sistem
  14. Customer menerima notifikasi barang sudah dikirim dan berapa lama waktu pengiriman
  15. Customer dapat melacak status pengiriman
  16. Pesanan sudah tiba dalam waktu yang ditentukan ?
  17. Jika belum maka kurir masih dalam proses pengriman
  18. Jika sudah maka konfirmasi pesanan telah tiba
  19. Memberikan nilai
  20. Jasa kurir admin dan sistem terkonfirmasi brng telah tiba
21. Proses selesai

### **3.4 Perancangan**

Untuk mempermudah proses pengelolaan data barang maka dirancanglah sebuah sistem informasi *online* untuk membuat proses pengelolaan barang menjadi lebih efektif, memberikan kemudahan bagi petugas untuk melakukan pendataan, serta penempatan tataletak barang.

#### **3.4.1 Use Case Diagram**

Diagram *use case* menggambarkan interaksi antar *use case* dan aktor dalam suatu sistem.



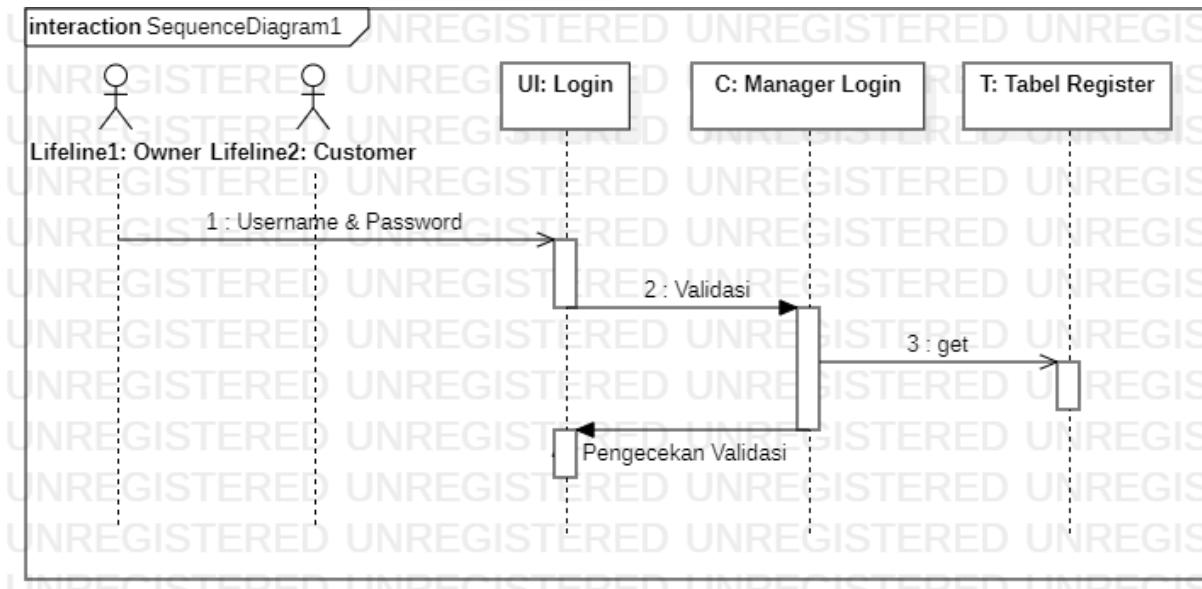
Gambar 3.11 Use Case Diagram

### 3.4.2 Sequence Diagram

*Sequence Diagram* merupakan penggambaran keterhubungan atau interksi antar objek dalam suatu jangka waktu. *Sequence Diagram* terutama menampilkan interaksi Antara pengguna (*user*) dengan sistem.

#### 3.4.2.1 Sequence Diagram Login

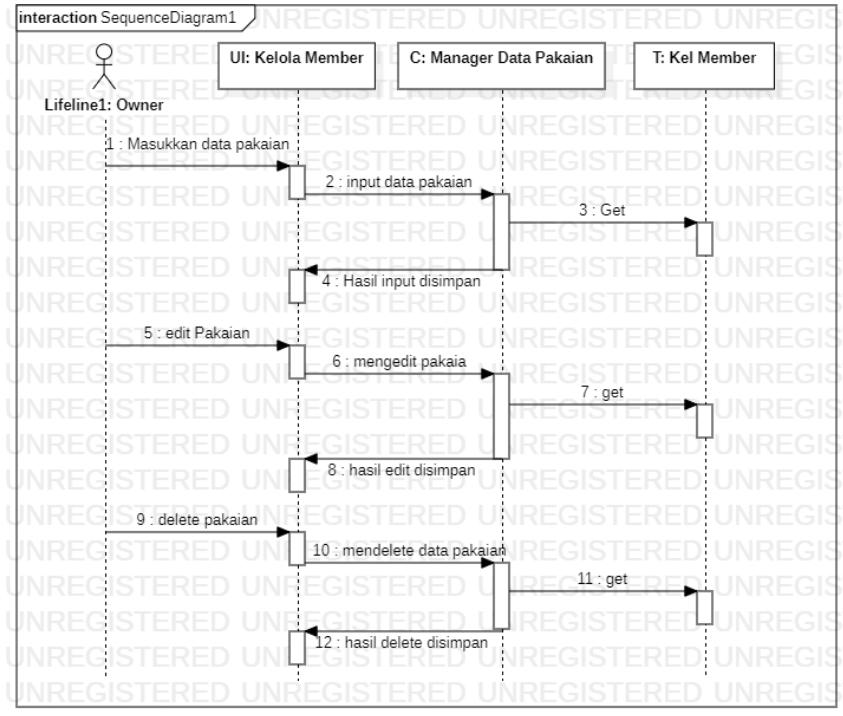
*Owner* dan *customer* menginput *username* & *password* untuk *login*, Sistem akan mengecek benar tidaknya *username* & *password* tersebut. Jika benar kemudia sistem akan masuk kehalaman utamam namun jika salah akan kembali kehalaman *login*.



Gambar 3.12 *Sequence Diagram Login*

### 3.4.2.2 *Sequence Kelola Pakaian*

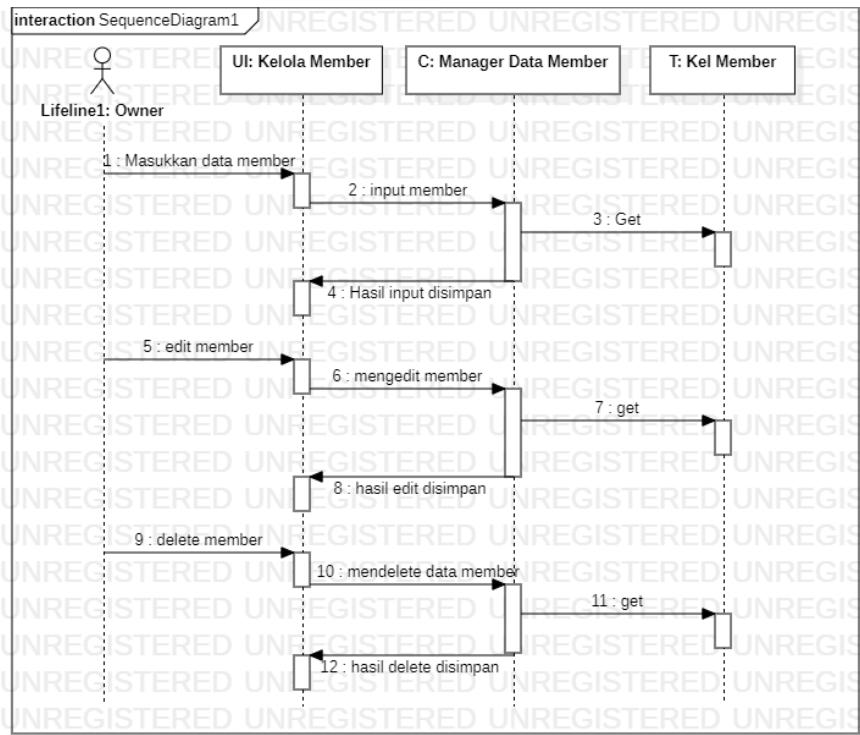
*Owner* masuk kedalam kelola pakaian, untuk melakukan *input* pakaian *owner* memilih menu *input* pakaian, kemudian memasukkan data pakaian, setelah selesai menginputkan data pakaian, data akan disimpan.



Gambar 3.13 *Squence kelola pakaian*

#### 3.4.2.3 *Sequence Kelola Member*

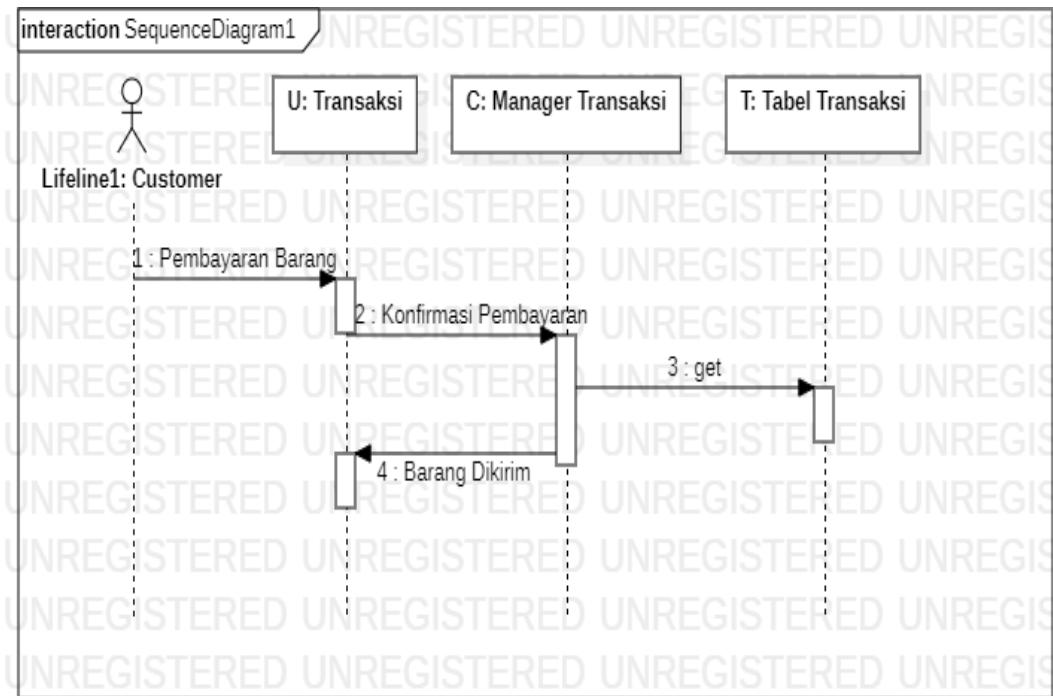
*Owner* masuk kehalaman kelola member, untuk melakukan *input member* petugas memilih menu *input member*, kemudian masukkan data *member*.



Gambar 3.14 *sequence* kelola member.

#### **3.4.2.4 Sequence Diagram Transaksi**

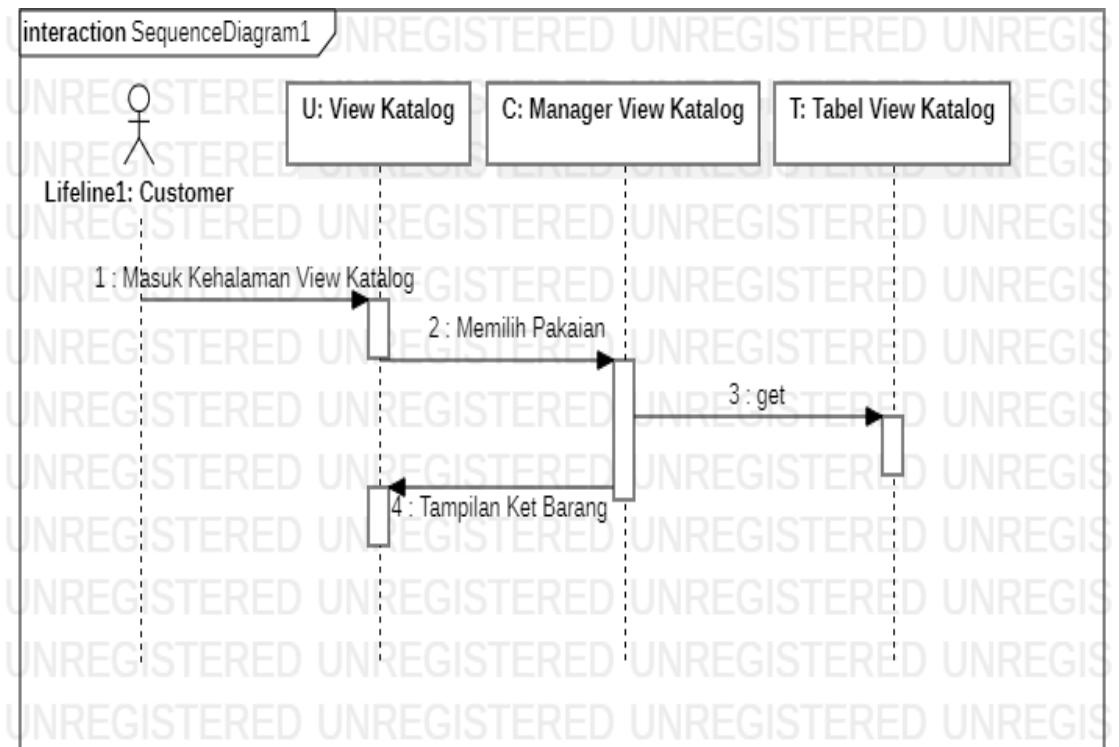
Untuk pembelian barang, masuk kemenu utama, kemudian mencari barang yang akan dibeli, ketika ingin membeli barang, *customer* harus menginputkan *username & password*. Jika *username & password* benar langsung saja ke transaksi, dan jika tsalah akan kembali kehalaman cari barang.



Gambar 3.15 *sequence transaksi*

#### 3.4.2.5 Sequence View Katalog

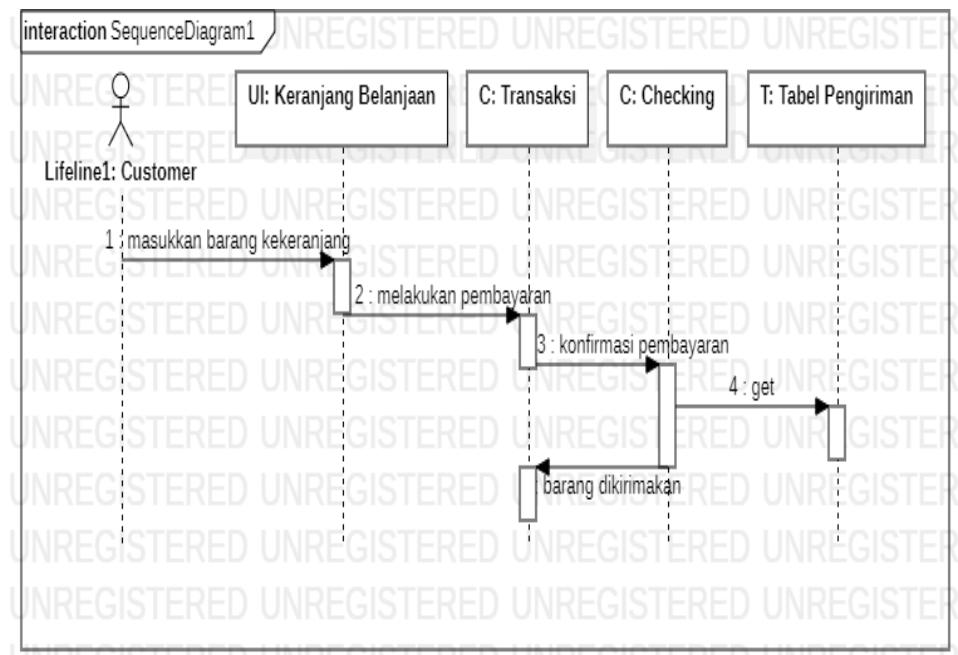
Untuk *view katalog*, *customer* memasuki halaman utama lalu memasuki *username* dan *password*, memilih barang, tampilan keterangan pakaian yang dibeli, melakukan pembayaran dan selesai.



Gambar 3.16 *Sequence diagram view catalog*

#### 3.4.2.6 Sequence Keranjang Belanjaan

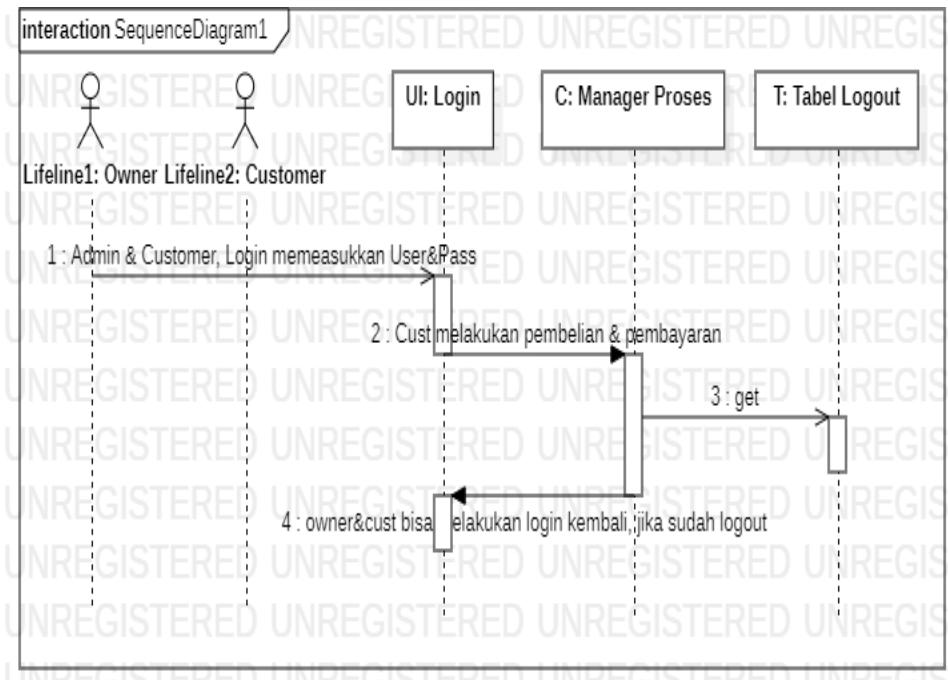
Setelah *customer* selesai memilih pakaian, lalu membeli pakaian yang dipilih dan melakukan pembayaran, jika selesai maka akan kembali ke menu pembelian.



Gambar 3.17 *sequence diagram* pembelian

#### 3.4.2.7 Sequence Logout

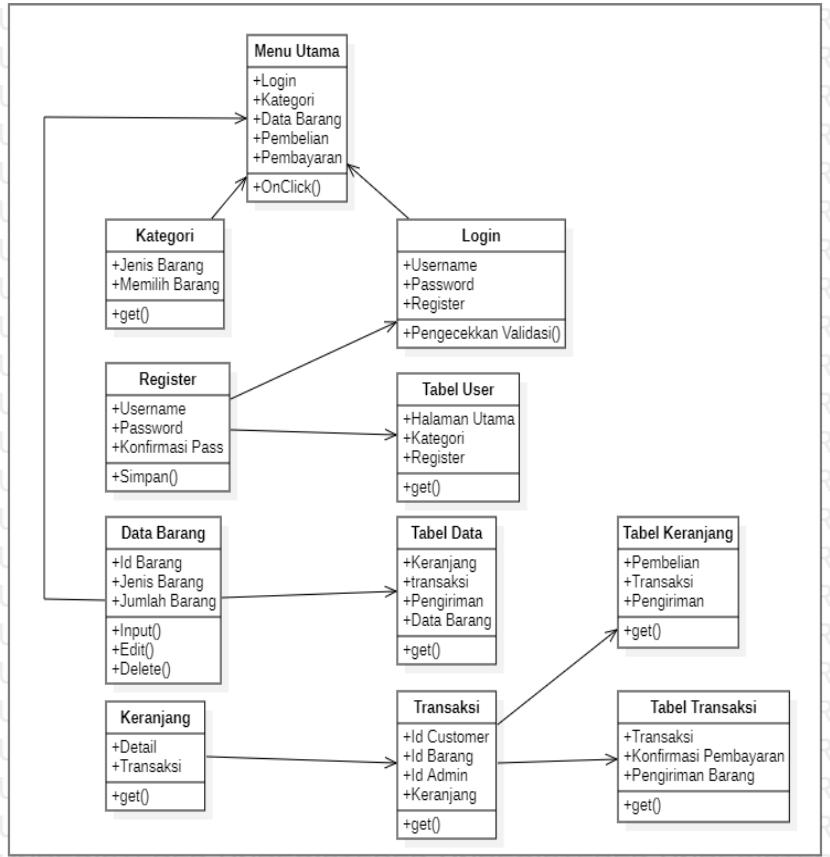
Setelah melakukan semua yang diinginkan maka bisa menglogout aplikasi, dan akan kembali ke halaman *login*.



Gambar 3.18 sequence diagram logout

### 3.4.3 Class Diagram

*Class diagram* merupakan struktur tetap yang akan digunakan dalam sebuah aplikasi. Dalam proses pembuatan aplikasi ini, *class diagram* yang akan digunakan sebagai berikut:



Gambar 3.19 *Class Diagram*.

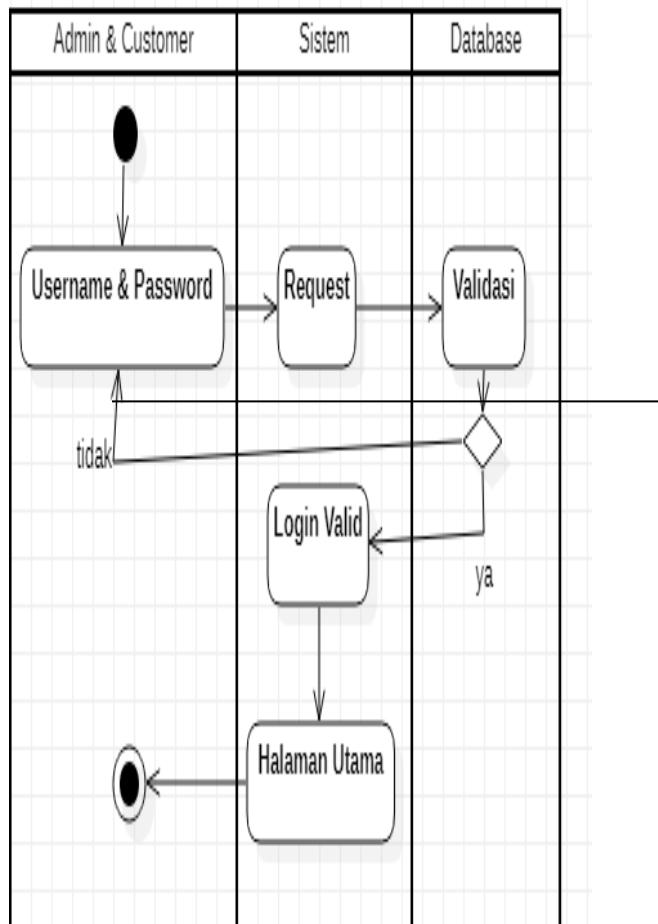
### 3.4.4 Activity Diagram

*Activity diagram* menggambarkan *workflow* suatu sistem atau proses bisnis. Diagram ini menunjukkan suatu alur kegiatan secara berurutan. Diagram ini digunakan untuk mendeskripsikan kegiatan-kegiatan dalam sebuah operasi meskipun juga dapat digunakan untuk mendeskripsikan alur kegiatan yang lainnya.

#### 3.4.4.1 Activity Diagram Login

User menginputkan *username & password* untuk *login*, kemudian sistem akan mengvalidasi *username & password* yang telah diinputkan. Sistem akan mengecek benar tidaknya *username & password* tersebut. Jika benar kemudian sistem akan masuk kehalaman utama, namun jika salah akan kembali kehalaman

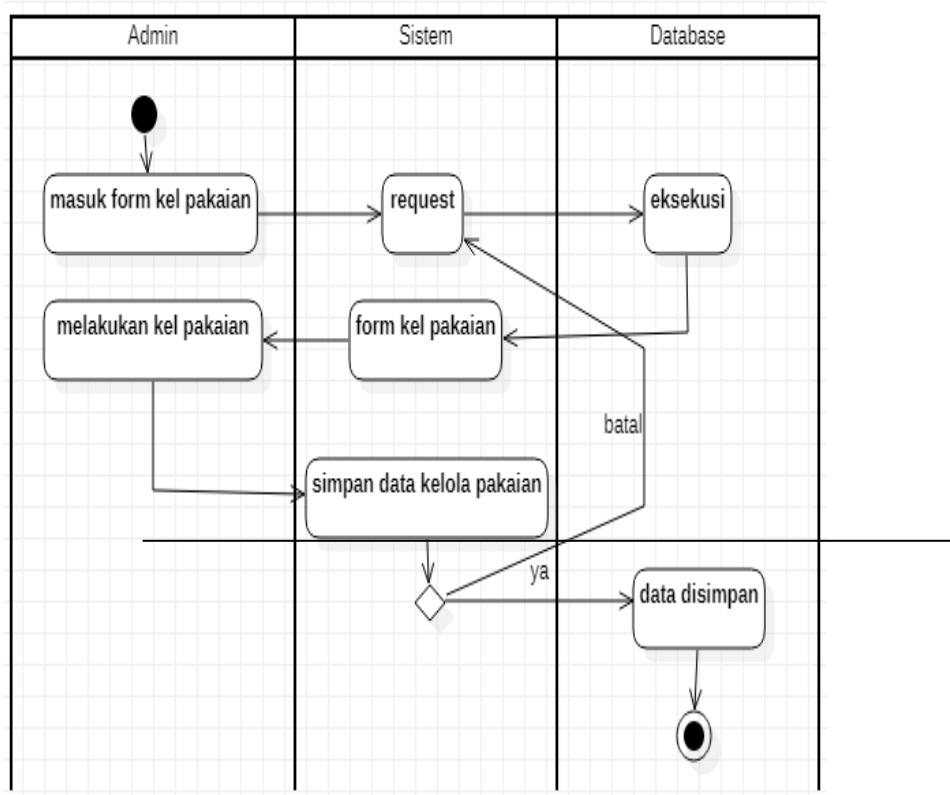
*login.*



Gambar 3.20 *Activity diagram login.*

#### 3.4.4.2 Activity kelola pakaian

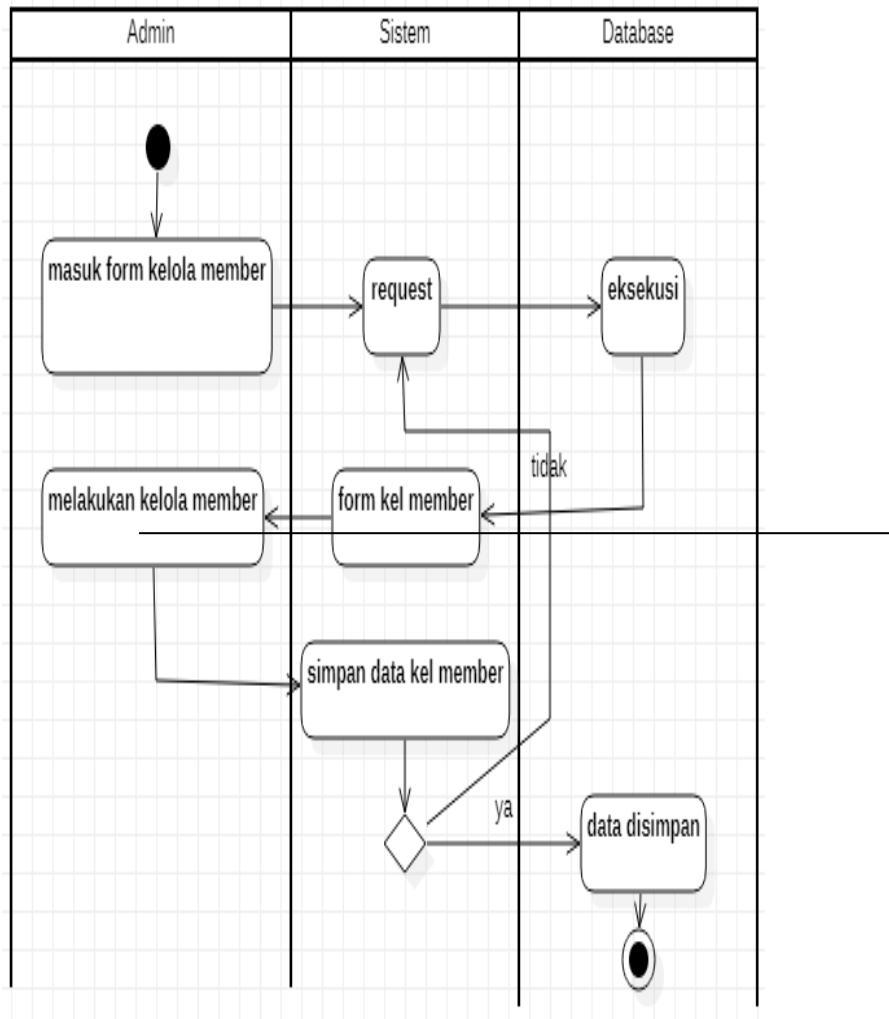
*Admin* masuk ke halaman kelola barang, untuk melakukan *input* barang dan *admin* memilih menu *input* barang, kemudian memasukkan data barang, setelah selesai menginputkan data barang, data akan disimpan.



Gambar 3.21 Activity kelola pakaian.

#### 3.4.4.3 Activity kelola member

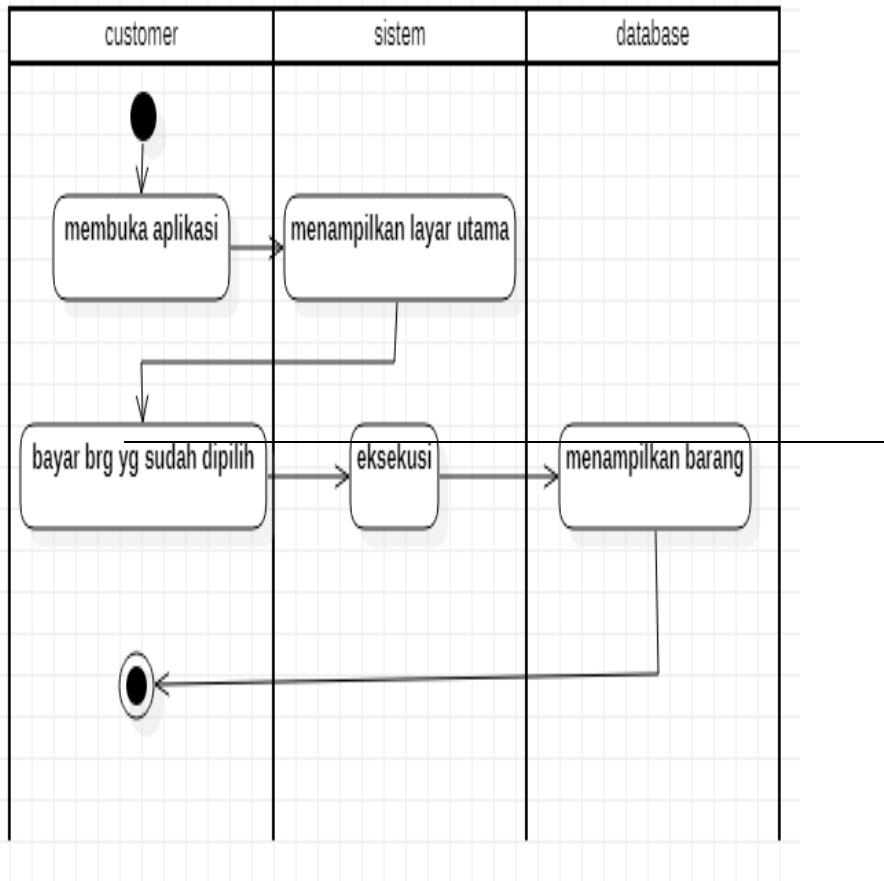
Admin akan masuk kedalam kelola *member*, untuk melakukan *input member* admin memilih menu *input member*, kemudian memasukkan data *member*, setelah selesai menginputkan data *member*, data kan disimpan.



Gambar 3.22 Activity kelola member.

#### 3.4.4.4 Activity Transaksi

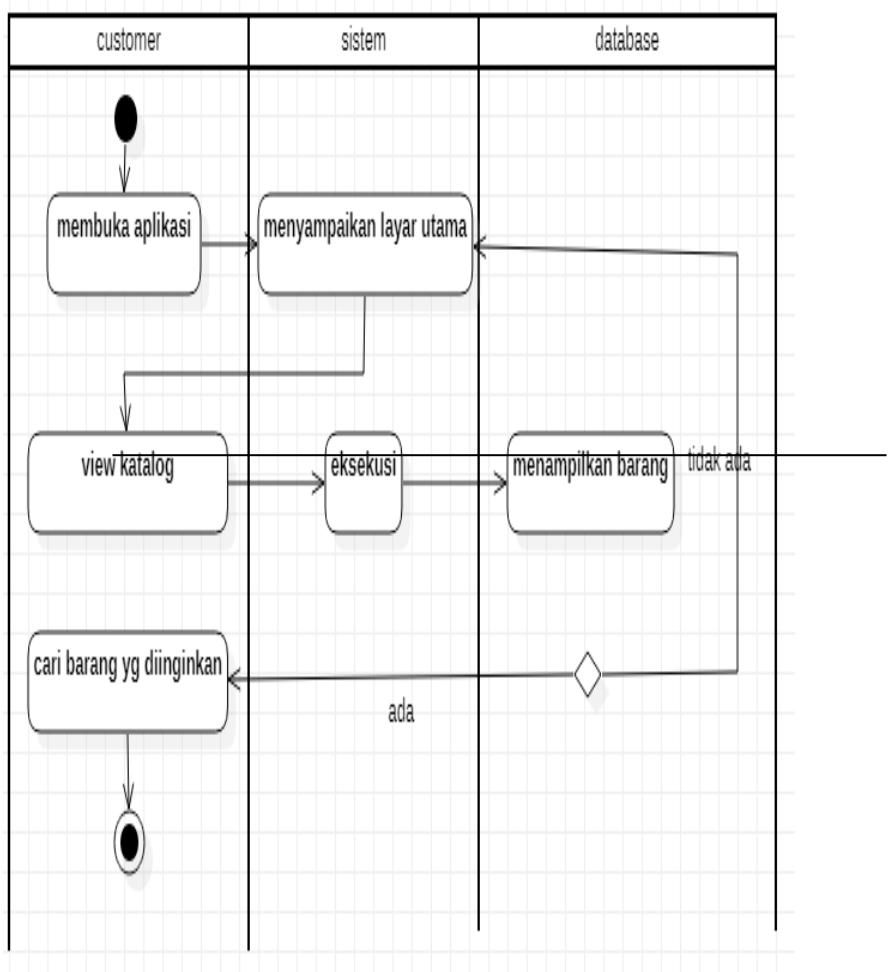
*Customer* mencari barang yang akan dibeli, lalu menampilkan data barang, mengecek barang yang akan dibeli, lalu *transaksi* dan selesai.



Gambar 3.23 *Activity transaksi.*

#### **3.4.4.5 Activity View Katalog.**

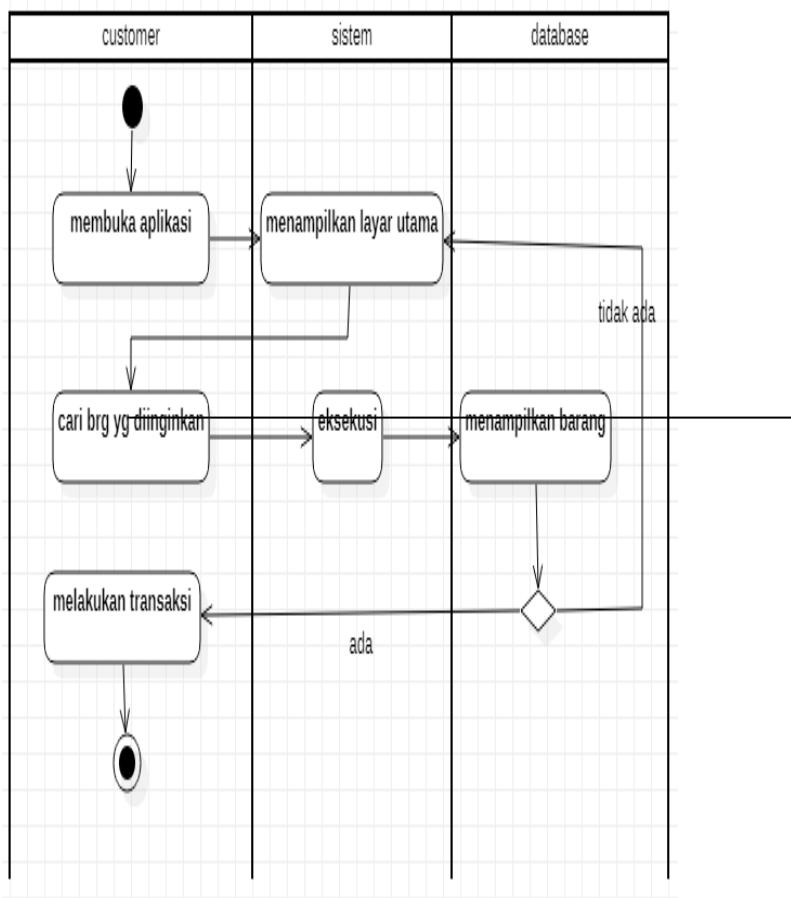
*Customer* membuka aplikasi, lalu sistem menampilkan layar utama, dan *customer* menekan tombol *search*, cari barang dan sistem mengeksekusi, lalu database mencari barang, jika barang tidak ditemukan maka kembali ke layar utama, jika berhasil menampilkan barang yang diinginkan maka selesai.



Gambar 3.24 *Activity View Katalog.*

#### **3.4.4.6 Activity Kernjang Belanjaan**

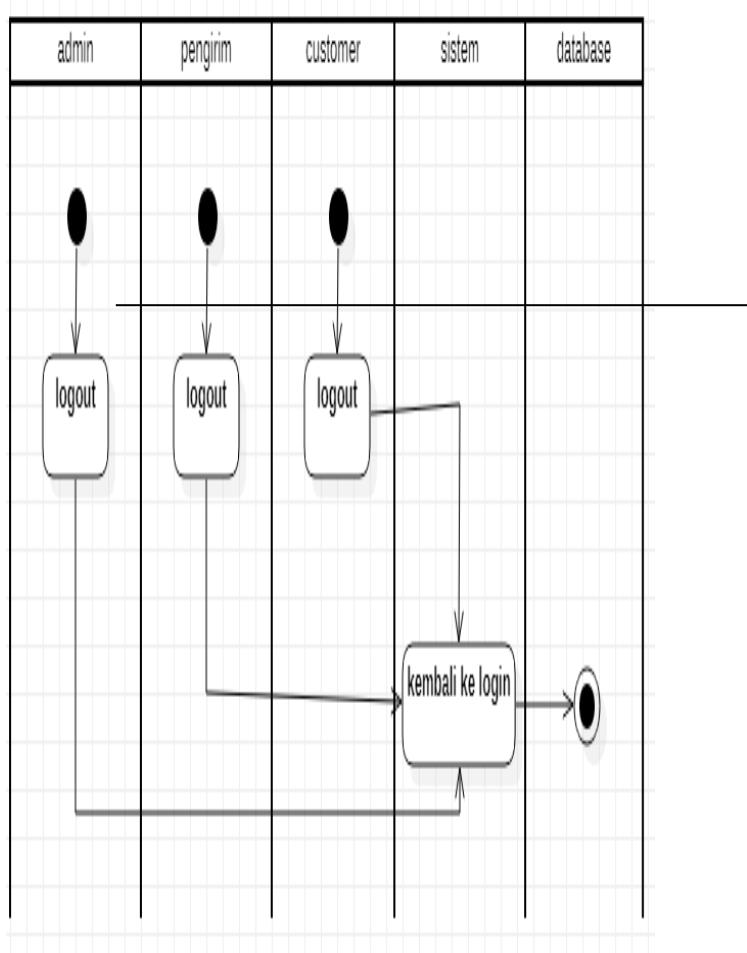
Customer membuka aplikasi, lalu system menampilkan layer utama, dan customer mencari barang yang diinginkan, dan system mengeksekusi, database menampilkan barang, jika barangnya ada maka customer melakukan pembayaran, dan jika tidak maka balik lagi ke layer utama.



Gambar 3.25 Activity Keranjan Belanjaan.

#### 3.4.4.7 Activity Logout

Admin, pengirim, dan customer menglogout aplikasi dan sistem akan menampilkan halaman *login*



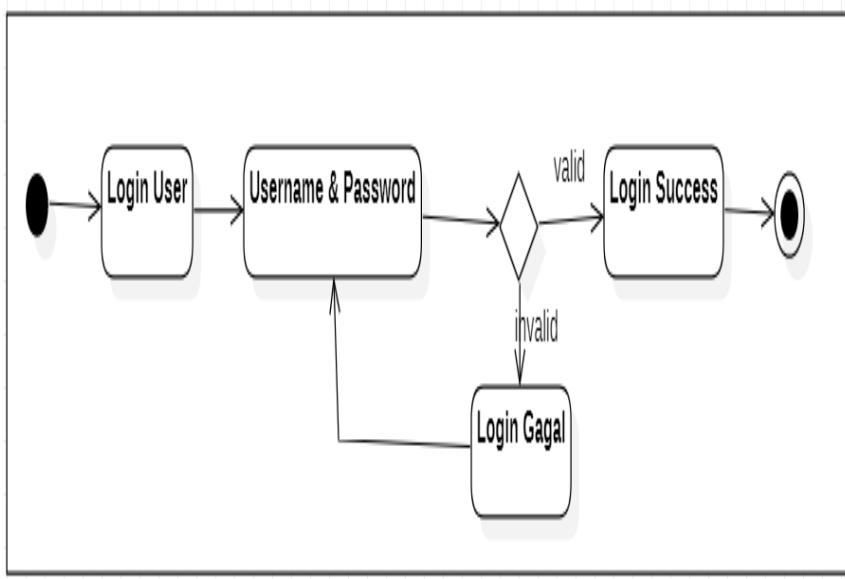
Gambar 3.26 Activity Logout.

### 3.4.5 Statechart Diagram

*Diagram* status digunakan untuk menyatakan kondisi sebuah objek pada saat sistem informasi berjalan. *Diagram* ini diadopsi dari penggambaran kondisi mesin status yang menggambarkan status apa saja yang dialami oleh mesin.

#### 3.4.5.1 Statechart Diagram Login

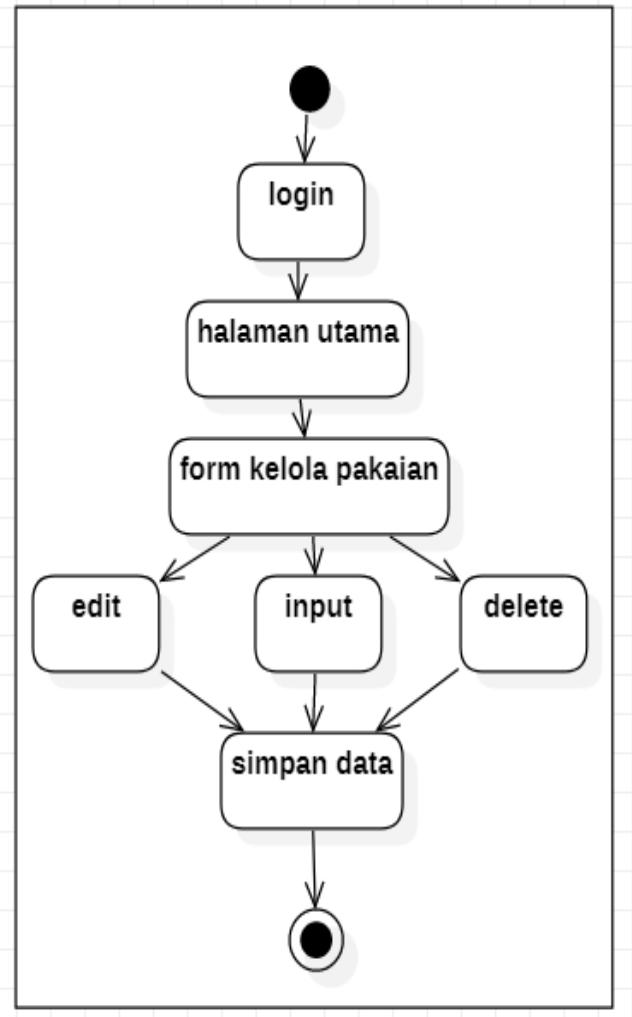
User menginputkan *username* & *password* untuk *login*, kemudian sistem akan mengvalidasi *username* & *password* untuk *login*, kemudian sistem akan mengvalidasi *username* & *password* yang telah diinputkan.



Gambar 3.27 Statechart Diagram Login.

#### 3.4.5.2 StateChart Kelola Pakaian

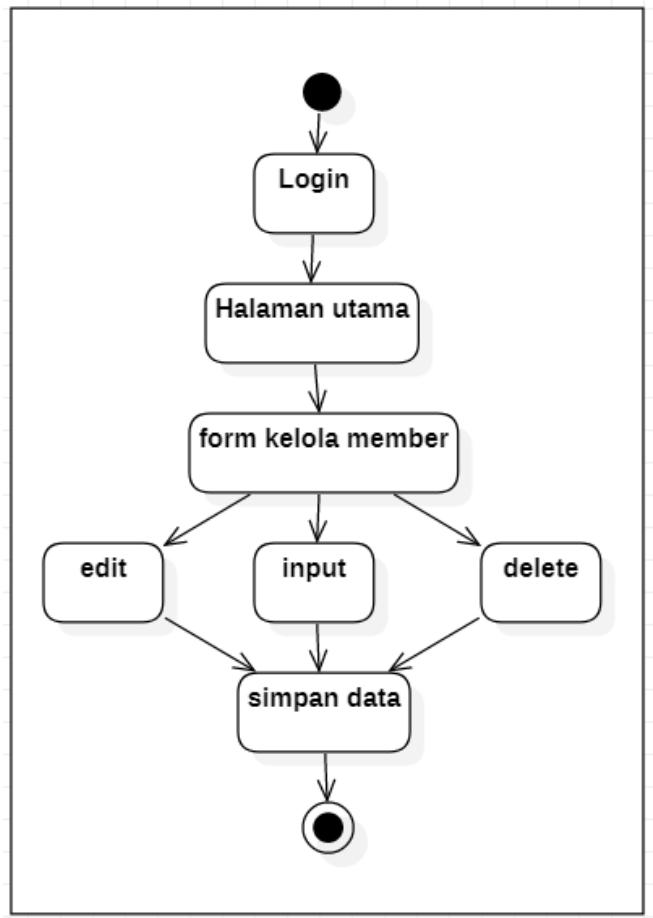
*Customer* mencari barang yang akan dibeli, lalu menampilkan data barang, mengecek barang yang akan dibeli, lalu *transaksi* dan selesai.



Gambar 3.28 Statechart Kelola Pakaian.

#### 3.4.5.3 StateChart Kelola Member

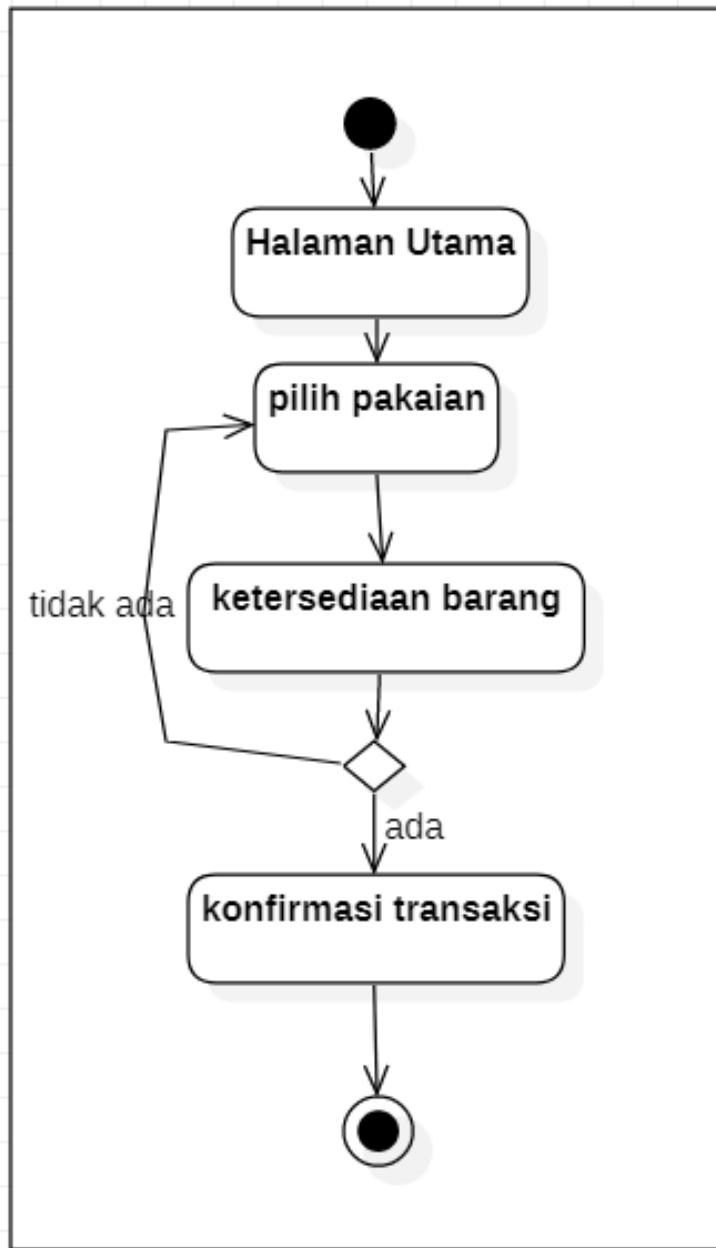
Admin akan masuk kedalam kelola *member*, untuk melakukan *input member* admin memilih menu *input member*, kemudian memasukkan data *member*, setelah selesai menginputkan data *member*, data kan disimpan.



Gambar 3.29 Statechart Kelola Member.

#### 3.4.5.4 Statechart Transaksi

*Customer* mencari barang yang akan dibeli, lalu menampilkan data barang, mengecek barang yang akan dibeli, lalu *transaksi* dan selesai.

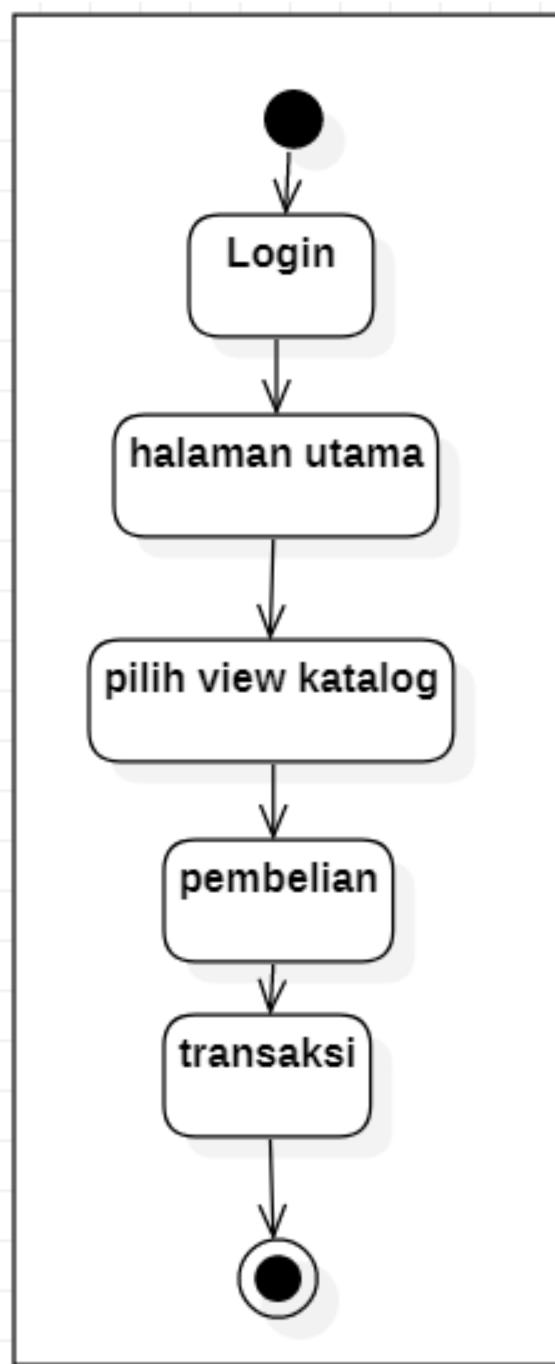


Gambar 3.30 Statechart Transaksi.

#### 3.4.5.5 Statechart View Katalog

*Customer* membuka aplikasi, lalu sistem menampilkan layar utama, dan pilih *view catalog*, cari barang dan sistem mengeksekusi, lalu database mencari barang, jika barang tidak ditemukan makan kembali ke layar utama, jika

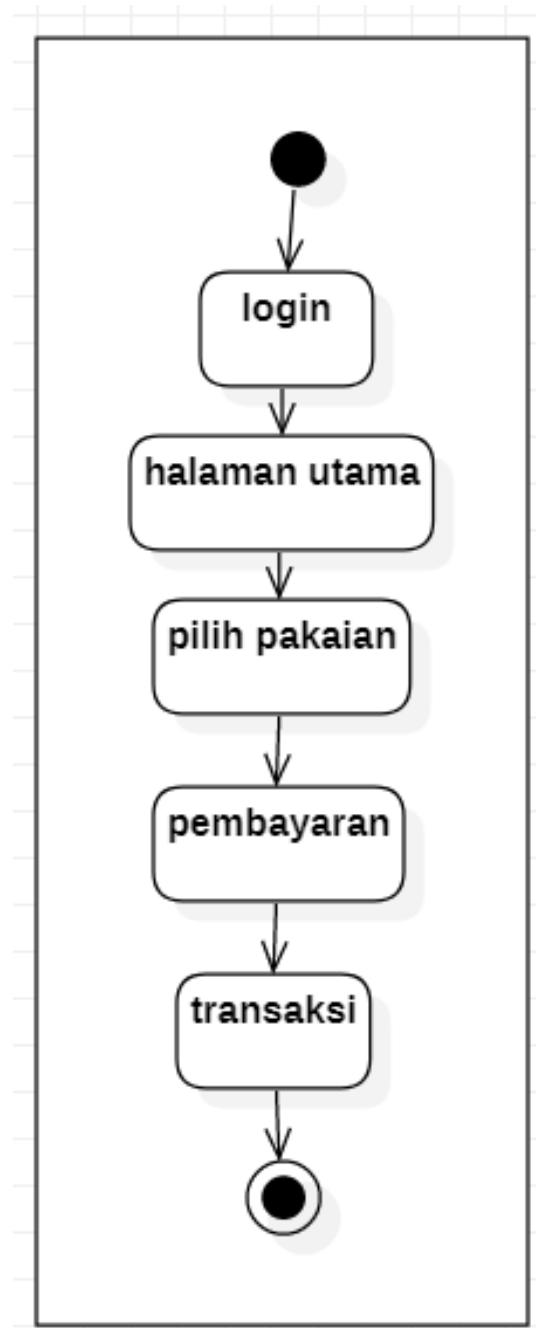
berhasil menampilkan barang yang diinginkan maka selesai.



Gambar 3.31 *Statechart View Katalog.*

#### 3.4.5.6 Statechart Keranjang Belanjaan.

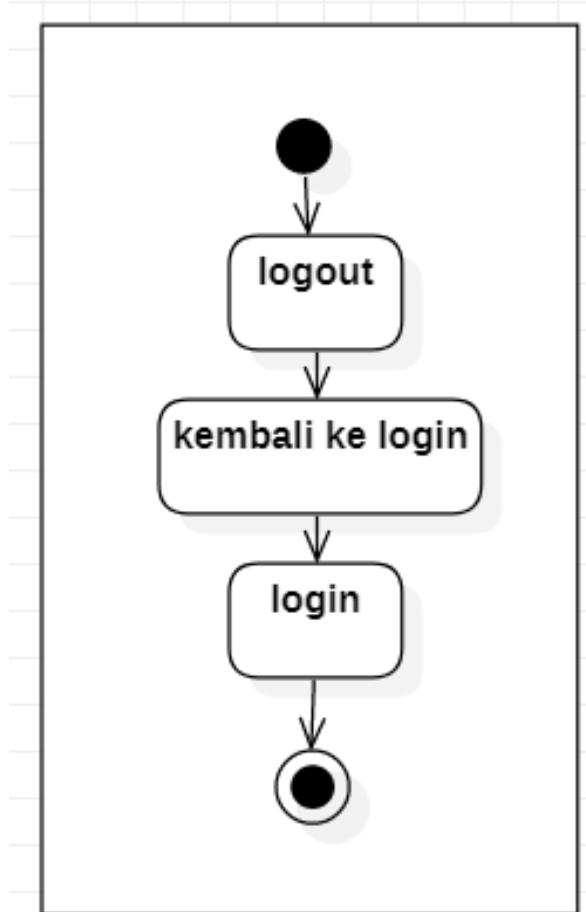
Login, kehalaman utama, lalu customer memilih pakaian, dan melakukan pembelian lalu pembayaran.



Gambar 3.32 Statechart Keanjang Belanjaan.

### **3.4.5.7 Statechart Logout.**

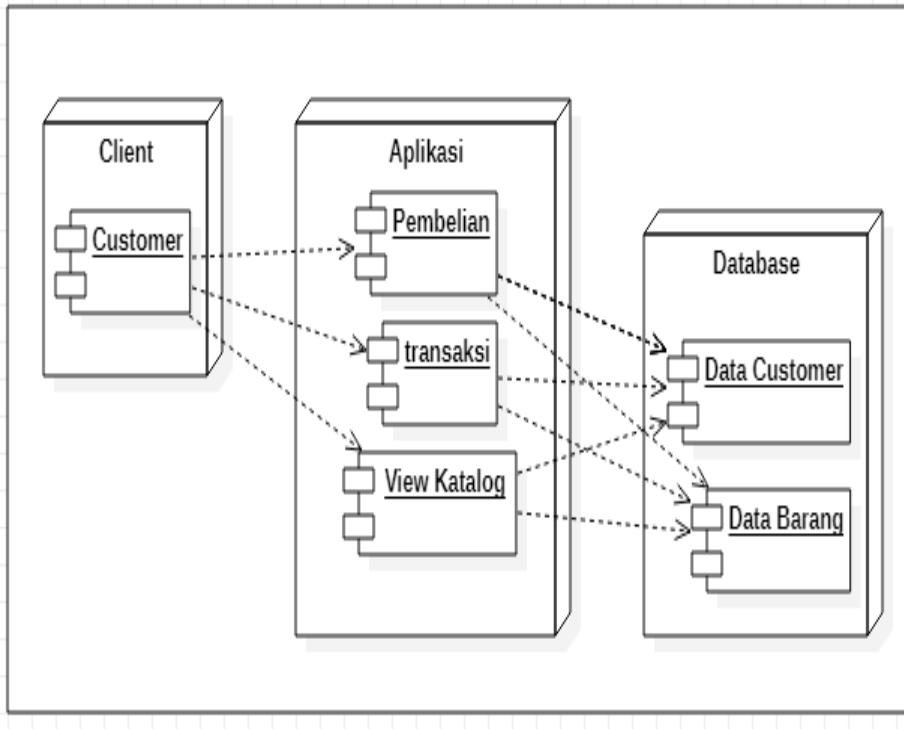
Jika ingin *logout*, lalu system akan menampilkan menu *login*



Gambar 3.33 *Statechart Logout.*

### **3.4.6 Component Diagram**

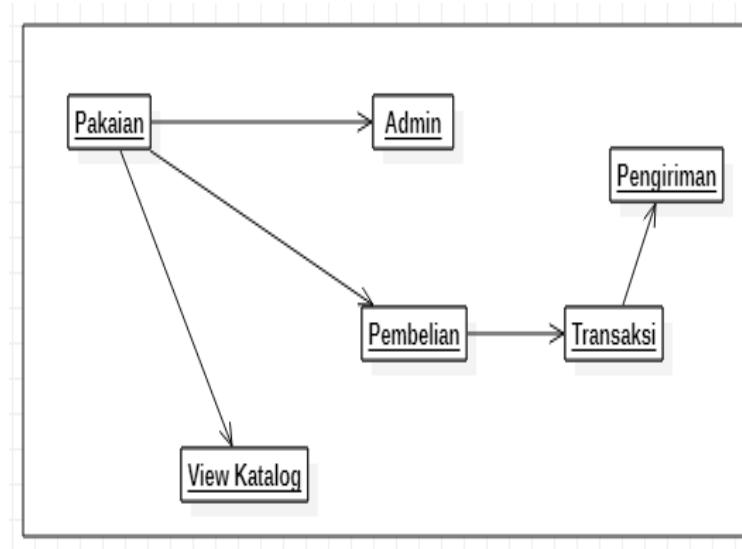
*Component Diagram* digunakan untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.



Gambar 3.34 *Component Diagram*.

### 3.4.7 *Object Diagram*

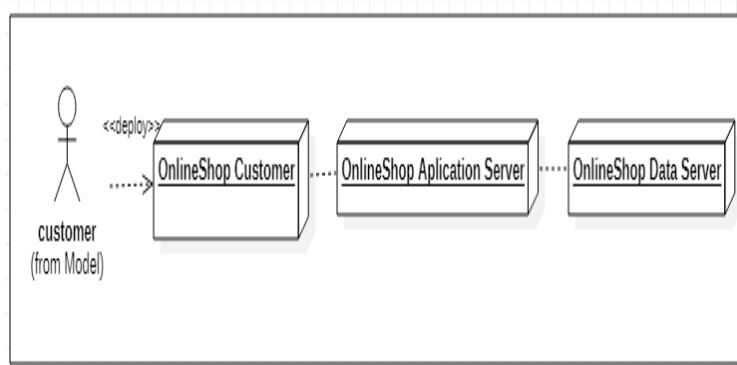
*Object Diagram* merupakan sebuah gambaran tentang objek-objek dalam sebuah sistem pada satu titik waktu. Karena lebih menonjolkan perintah-perintah daripada *class*, *object diagram* lebih sering disebut sebagai sebuah *diagram* perintah.



Gambar 3.35 *Object Diagram*.

### 3.4.8 Deployment Diagram

*Deployment Diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. *Diagram deployment* juga dapat digunakan untuk memodelkan hal-hal seperti sistem tambahan yang menggambarkan rancangan *device*, *node*, dan *hardware*, sistem *client/server*, sistem terdistribusi murni, dan rekayasa ulang aplikasi.



Gambar 3.36 *Deployment Diagram*

## Bab 4

---

## **Implementasi dan Pengujian**

---

### **4.1 Teori**

#### **4.1.1 Implementasi**

Implementasi sistem dilakukan ketika kebijakan telah dibuat, pekerjaan selanjutnya adalah mengatur, melakukan kepemimpinan untuk proses kepemimpinan dan melaksanakan proses kontrol.

Kegiatan terperinci dalam mengelola implementasi kebijakan dapat diatur melalui:

- Implementasi strategi
- Pengorganisasian
- Penggerak dengan baik
- Kontrol

Implementasi yang melibatkan upaya pembuat kebijakan untuk mengubah apa yang disebut "birokrat tingkat jalan" oleh Lipsky untuk menyediakan layanan sesuai dengan kebijakan kelompok sasaran. Untuk kebijakan sederhana, implementasi gratis yang melibatkan badan yang mengimplementasikan sebagai implementor, misalnya, kebijakan pengembangan infrastruktur publik untuk membantu orang-orang yang memiliki kehidupan yang baik, untuk keberhasilan implementasi kebijakan akan ditentukan dengan jumlah variabel atau aspek, dan masing-masing variabel tersebut dibahas secara setara satu sama lain Kegiatan implementasi kebijakan sejak awal implementasi strategi, organisasi, gerakan kepemimpinan dan pengawasan dilakukan sesuai dengan apa yang diinginkan. Bab ini merupakan implementasi dari “Aplikasi Pembelian Pakaian Berbasis

Website” Implementasi merupakan tahap menerapkan hasil analisis dan perancangan ke tahap pembuatan program sesuai rancangan yang dilakukan.

#### **4.2.1 Prosedur Implementasi**

Dan pengorganisasian berikut merumuskan prosedur pelaksanaan, yang disusun dalam tipe dasar

- Desain organisasi dan struktur organisasi.
- Divisi pekerjaan dan desain pekerjaan.
- Integrasi dan koordinasi.
- Rekrutmen dan penempatan sumber daya manusia.
- Hak, otoritas juga kewajiban.
- Delegasi.
- Pengembangan kapabilitas organisasi dan kapabilitas sumber daya manusia.
- Budaya organisasi.
- Aspek berikutnya adalah mobilisasi dan kepemimpinan yang mengalokasikan sumber daya, mengelola prosedur implementasi dengan sumber daya.

Dalam fase kebijakan ini, ada dasar untuk fleksibilitas atau ruang bagi pelaksana individu untuk memilih tindakan otonom mereka sendiri dalam batas kewenangan

Dengan beberapa rencana:

- Efektivitas kepemimpinan.
- Motivasi.
- Norma.
- Kualitas.

- Kerja tim.
- Komunikasi organisasional.
- Perundingan
- Aspek selanjutnya adalah membahas proses dengan melakukan pemantauan berkala dengan beberapa rencana:
- Desain kontrol.
- Sistem manajemen informasi.
- Pemantauan
- Kontrol biaya atau keuangan.
- Audit.
- Implementasi kebijakan pada prinsipnya adalah langkah kebijakan yang dapat diambil oleh seseorang untuk mengambil keputusan.

### Implementasi Kebijakan Umum

Untuk menerapkan kebijakan umum, ada dua langkah yang tersedia, yang dilaksanakan langsung dalam bentuk program atau melalui perumusan kebijakan turunan atau berasal dari kebijakan umum, Implementasi adalah tindakan untuk mencapai tujuan yang diuraikan dalam ketentuan kebijakan,tindakan itu dilakukan baik oleh individu, baik pejabat pemerintah swasta. Sehubungan dengan uraian ini dapat ditarik kesimpulan jika implementasinya merupakan proses yang dinamis, di mana pelaksana kebijakan mengatur kegiatan atau kegiatan, hingga akhirnya hasil yang sesuai dengan maksud atau tujuan kebijakan akan diperoleh.

Menurut Hanifah Harsono (2002: 67)

Menurut Hanifah Harsono dalam bukunya yang berjudul Implementasi Kebijakan dan Kebijakan Politik mengenai implementasi atau proses sebagai berikut: Implementasi adalah proses implementasi kebijakan menjadi kebijakan kebijakan dalam mengembangkan kebijakan administrasi dalam rangka

meningkatkan program.

### **Definisi Implementasi Umum**

Gagasan di atas menunjukkan apakah implementasi kata bermuara pada kegiatan, perilaku, tindakan, atau sistem. Arti pemindahan ekspresi berarti jika diterapkan pada pelaksanaan suatu kegiatan, tetapi suatu kegiatan yang direncanakan dan dilaksanakan dengan perhatian penuh pada referensi ke norma-norma tertentu untuk mencapai tujuan kegiatan. Oleh karena itu, implementasi tidak berdiri sendiri tetapi dipengaruhi oleh objek berikutnya, kurikulum. Sesuai dengan kesepakatan, implementasi kurikulum Fullan adalah proses koordinasi, pemisahan program atau kegiatan dengan harapan orang lain dapat menerima dan melakukan perubahan. Dalam proses penyelesaian, evaluasi yang disebutkan di atas memberikan tekanan pada proses. Inti dari implementasi adalah proses, kegiatan yang digunakan untuk mentransfer / ide, program atau harapan sebagaimana diuraikan dalam kurikulum desain (disediakan) yang disetujui sesuai dengan desain. Masing-masing demi mencerminkan tingkat proses yang berbeda. Dalam diskusi dengan perjanjian, Nurdin dan Usman (2004) menjelaskan bahwa jika ditanya sebelumnya, evaluasi pelaksanaan dilakukan sebelum distribusi (diseminasi) desain kurikulum.

Proses program terkait dengan kegiatan yang berkaitan dengan menjelaskan tujuan program, menjelaskan sumber-sumber baru dan menunjukkan bagaimana menemukan apa yang digunakan. Keputusan kedua, menurut Nurdin dan Usman (2002), memprioritaskan fase perbaikan. Kata proses dalam dukungan menekankan hubungan antara pengembang dan guru (memfasilitasi pendidikan). Pengembang melakukan kontrol atas program yang baru disetujui, sumber baru, dan menggabungkan konten / materi program baru dengan sumber yang ada berdasarkan hasil uji coba lapangan dan pengalaman para guru. Hubungan antara pengembang dan guru terjadi dalam konteks penyempurnaan program, pengembang menyelenggarakan lokakarya atau diskusi

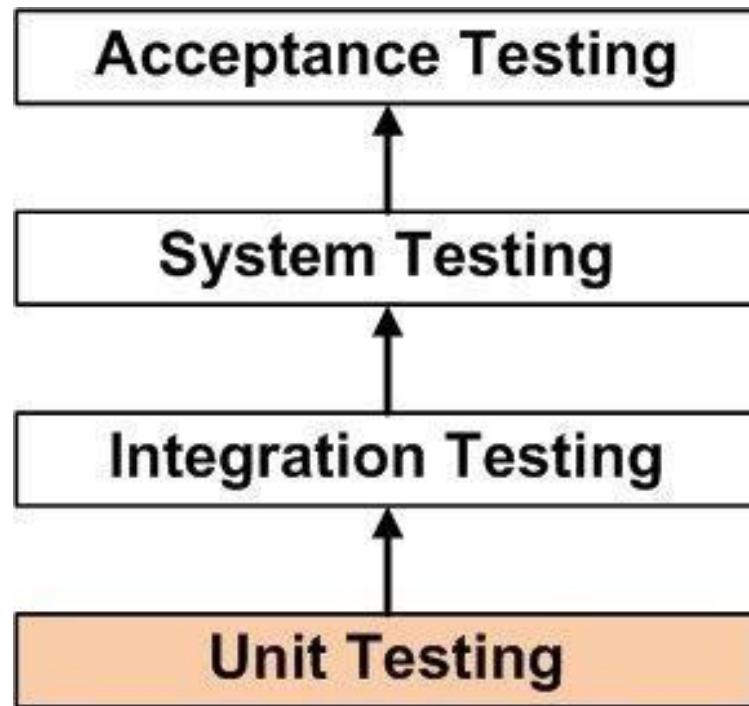
dengan guru untuk mendapatkan masukan. Implementasi dianggap selesai ketika proses peningkatan program baru tampaknya telah selesai. Sementara menyetujui tiga, Nurdin dan Usman (2002) melihat implementasi sebagai sisi dari program kurikulum. Implementasi dalam Kamus Besar Bahasa Indonesia disimpulkan sebagai suatu proses atau aplikasi. Berharap bahwa apa yang dilakukan dengan baik diimplementasikan dalam kurikulum yang telah dirancang / dirancang untuk diimplementasikan sepenuhnya.

#### **4.4 Analogi Implementasi**

Ketika misalkan dengan desain bangunan yang dibuat oleh insinyur bangunan tentang desain perumahan di kertas kalkir. Sehingga implementasi yang dilakukan oleh beberapa pengrajin yang telah dirancang telah dibuat dan tidak mungkin dilakukan atau direncanakan akan terdistorsi atau tidak sesuai dengan desain. Jika apa yang dilakukan oleh beberapa pabrikan tidak sama dengan hasil desain yang akan terjadi pada bangunan yang telah dibuat karena hasil dari proses yang panjang, rumit, sulit dan sempurna dalam hal desainer dan desain. Implementasi kurikulum juga diperlukan untuk melakukan apa yang telah dilakukan dalam kurikulum untuk dilakukan dengan sepenuh hati dan semangat. Masalah besar akan terjadi yang dilakukan dengan menentang atau menyimpang dari apa yang telah dibuat. Akan ada pemborosan antara desain dan implementasi.

##### **4.1.1 Lingkungan Implementasi**

Lingkungan implementasi dari suatu program aplikasi adalah perangkat keras dan perangkat lunak serta program aplikasi lainnya yang mendukung implementasi program aplikasi tersebut. Agar perancangan program aplikasi dapat berjalan dengan baik maka dibutuhkan perangkat keras pendukung dan perangkat lunak yang sesuai.



#### **4.1.2 pengujian**

##### **4.1.2.1 Proses pengujian**

Sistem berfungsi dengan baik dan mencari masalah yang mungkin terjadi sistem.

Tujuan pengujian adalah untuk bereksperimen:

1. Kesalahan bahasa (kesalahan bahasa), kesalahan yang disebabkan oleh jawab dalam pembaruan sintaksis.
2. Kesalahan waktu pemrosesan (kesalahan runtime), kesalahan kompilasi program dimulai. Kesalahan ini akan menyebabkan proses program berhenti sebelum berhasil berhenti.
3. Kesalahan logika, kesalahan yang disebabkan oleh logika program dibuat. Kesalahan ini sulit ditemukan karena tidak ada Ingat lokasi kesalahan.

##### **4.1.2.2 Rencana Tes**

Penilaian dilakukan dalam menguji penggunaan perangkat lunak ini mendiskusikan pengujian kotak hitam. Setuju dengan ini perangkat lunak fokus.

#### **4.1.2.3 Metode Pengujian**

Metode pengujian adalah metode atau teknik pengujian perangkat lunak, memiliki mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak sepenuhnya dan memiliki kemungkinan tinggi menemukan kesalahan. Perangkat lunak dapat dirilis dalam dua cara, yaitu:

1. Pengujian menggunakan data uji untuk menguji semua elemen program (data internal, loop, logika, keputusan, dan jalur). Uji data dimunculkan dengan memahami struktur internal (kode sumber) dari perangkat lunak.
2. Pengujian dilakukan dengan mengeksekusi data uji dan memeriksa apakah Perangkat fungsional yang berfungsi dengan baik. Data uji dihasilkan dari spesifikasi perangkat lunak.

##### **1. UJI KOTAK PUTIH**

Pengujian kotak putih (kotak kaca) adalah tes yang didasarkan pada periksa detail desain, menggunakan struktur control program desain untuk pengujian procedural beberapa kasus pengujian. Penentuan masalah tes disesuaikan dengan struktur sistem, pengetahuan program yang digunakan untuk mengidentifikasi kasus uji tambahan.

Tujuan menggunakan kotak putih adalah untuk memprogram semua pernyataan.

Penggunaan metode pengujian kotak putih dilakukan untuk:

1. menyediakan modul
2. digunakan setidaknya sekali
3. gunakan semua keputusan logis untuk semua kondisi benar atau salah
4. mengeksekusi semua pengulangan nilai dan kendala operasional pada
5. setiap kondisi.
6. sengan digunakan struktur data internal untuk menjamin validitas jalur keputusan.

##### **A. Pengujian Jalur Dasar**

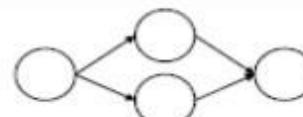
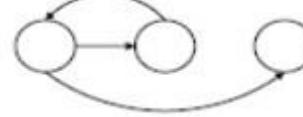
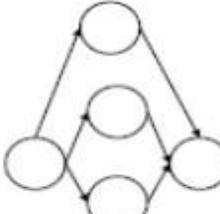
Tes jalur dasar adalah tes kotak putih pertama yang direkomendasikan oleh Tom

McCabe. Metode ini memungkinkan penguji untuk mengukur rumit seperti logis dari desain prosedural dan rumit panduan untuk membangun satu set dasar dari semua jalur eksekusi.

#### a. Notasi Diagram Alir

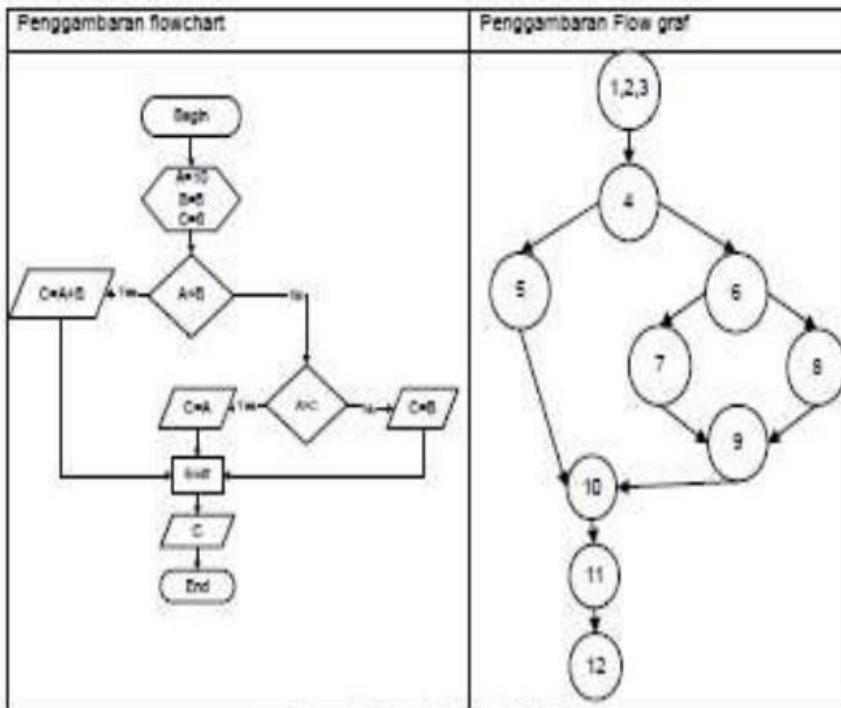
Notasi digunakan untuk menggambarkan jalur eksekusi notasi diagram alir (atau grafik program), yang menggunakan notasi loop atau simpul) dan panah. Notasi ini menggambarkan aliran kontrol logika yang digunakan dalam bahasa pemrograman.

Tabel – Notasi Diagram Alir

NOTASI	ARTI
	Langkah sequence
	Skema If
	Skema While (...) Do (...)
	Skema Repeat (...) Until (...)
	Skema Case (...) Of

## Contoh Suatu PDL

```
A :- 10;          (1)
B  5;           (2)
C:- 6;          (3)
If A>B        (4)
    then C:-A+B   (5)
    Else ifA»C    (6)
        then Can  (7)
        Else C:-B; (8)
    Endif         (9)
Endif            (10)
writeln(*Nilai C - ',C); (11)
End.          (12)
```



Gambar – Contoh Metode Basis Path

## B. Kompleksitas siklus

The Complexity Cycle adalah perangkat lunak metrik yang disediakannya Pengukuran kuantitatif program logis, nilai yang diperoleh akan ditentukan oleh jumlah jalur independen di jalur yang ditentukan, juga memberikan nilai batas atas untuk jumlah tes yang harus dilakukan dilakukan, untuk memastikan bahwa semua yang telah disepakati telah dieksekusi lihat satu kali. Jalur independen adalah jalur yang termasuk dalam program mengintegrasikan satu atau beberapa proses.

### Perhitungan siklus

1. Diagram alir diagram area (area wajah)

2.  $V(G) = E - N + 2$ .

E adalah jumlah tepi, dan N adalah jumlah node

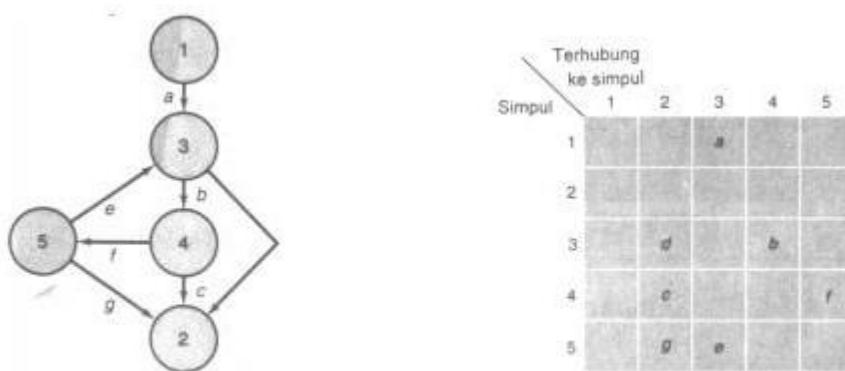
3.  $V(G) = P + 1$

P adalah simpul predikat Predicate Nodes adalah penggambaran simpul yang

memiliki satu atau lebih banyak input, dan lebih dari satu output.

### C. Graph Matrix (Graph Matrix)

Bentuk struktur data ini sering digunakan untuk pengujian adalah dengan matriks grafis. Matriks grafis adalah matriks persegi Jumlahnya sama dengan jumlah dalam bagan alur. Masukkan ke dalam matriks harus sesuai dengan arah. Matriks grafis selanjutnya disebut sebagai koneksi matriks, dan dijelaskan Sama dengan matriks tetangga dengan memperhatikan arah masuk dan keluar dari tepi.



Gambar – Contoh Matriks Koneksi

Dalam gambar flowgraph, setiap node ditandai dengan nomor klan dengan huruf kecil, kemudian terjemahkan ke diagram matriks. Contoh hubungan antara simpul 3 dan simpul 4 pada grafik ditandai dengan huruf b. Timbangan bobot memberikan informasi tambahan tentang aliran kontrol. Dalam hubungan yang sederhana seseorang dapat dengan mudah diberi nilai 1 jika ada hubungan di antara mereka simpul atau nilai 0 jika tidak ada hubungan.

Simpul	Terhubung ke simpul				
	1	2	3	4	5
1	1		1		
2		1			
3		1		1	
4		1			1
5		1	1		

**Gambar - Hubungan bobot**

Koneksi:

Simpul1: 1-1=0

Simpul2: 2-1=1

Simpul3: 2-1=1

Simpul4: 2-1=1

Cyclometric complexity = jumlah simpul +1

$$=3+4=7$$

### **Pengujian Struktur Kontrol**

Teknik pengujian jalur dasar adalah salah satu dari beberapa teknik untuk Memeriksa kontrol struktur, tetapi jalur pengujian dasar tidak memadai beberapa kasus uji.

#### **A. Kondisi Pengujian**

Ketentuan Penggunaan Pengujian program. Kondisi sederhana menggunakan variabel Boolean, dengan bentuk persamaan. E1 (operator-relasional) E2 Jika

suatu kondisi salah, maka akan ada komponen yang tidak ada dari kondisi yang salah, menghasilkan jenis kesalahan ketika kondisi berikut adalah:

1. operator kesalahan Boolean
2. Kesalahan variabel Boolean
3. tanda kurung Boolean Error
4. Kesalahan operator relasional
5. Kesalahan Persamaan Aritmatika

Contoh:

C1: B1 & B2

B1 & B2 adalah dua variabel Boolean sehingga mereka menjadi 2<sup>2</sup>

$B_1 = T; B_2 = T$	
$B_1 = T; B_2 = F$	
$B_1 = F; B_2 = T$	
$B_1 = F; B_2 = F$	$\{(T,T), (T,F), (F,T), (F,F)\}$

b. Tes Aliran Data

Metode pengujian untuk menguji aliran data menggunakan resolusi variabel dalam program, efektif digunakan untuk melindungi kesalahan, Tetapi akan memiliki penilaian dan pemilihan jalur uji kompleks.m

c. Pengujian loop

Ada beberapa jenis loop, yaitu:

1. Sederhana

Tes harus memperhatikan hal-hal berikut:

- a) Mengabaikan seluruh loop

- b) Hanya ada satu jalan yang dilalui lingkaran
- c) Suatu variabel akan melewati loop jika nilainya lebih besar dari nilainya bertekad
- d) Harus memperhatikan batas variabel pada loop. (Kondisi: n-1, n, n + 1)

## 2. Nested Loop

Jumlah yang akan bertambah sesuai geometrik sesuai dengan jumlahnya lingkaran bersarang. Untuk menyederhanakan pengujian, itu harus pertimbangkan hal berikut:

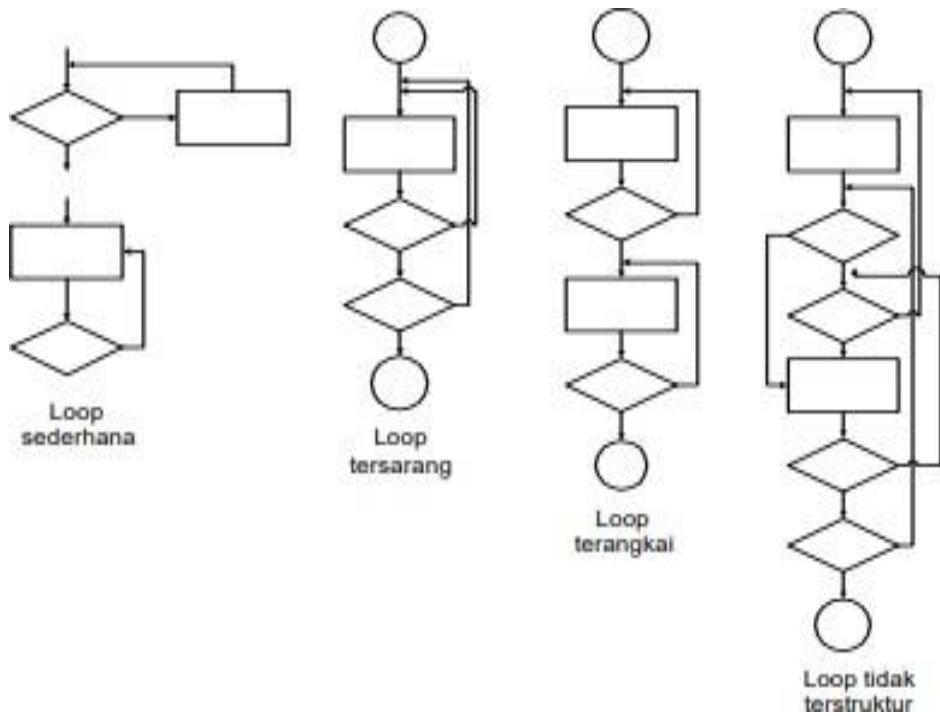
- a) memulai pengujian dari loop terdalam, dengan memulai pengujian dari minimumnya.
- b) lakukan tes loop sederhana dari loop terdalam ke loop terluar, dengan mempertimbangkan parameter yang digunakan.

## 3. Lingkaran Tegang

Tes untuk lingkaran harus disesuaikan sesuai dengan independensi variabel antara loop ini. Jika variabel digunakan di loop kedua tidak dihitung dengan loop pertama, maka pengujian loop digunakan sederhana, setiap kali lingkaran kedua bergantung pada nilai loop kedua, kemudian lakukan tes bersarang.

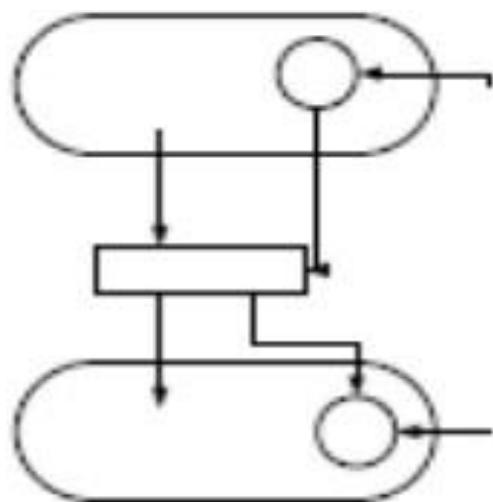
## 4. Loop Tidak Terstruktur

Sama sekali tidak direkomendasikan untuk digunakan dalam pembuatan program.



## B. MENGUJI KOTAK HITAM

Pengujian black box adalah pengujian pelengkap teknik putih kotak, karena pengujian kotak hitam diharapkan untuk mengungkapkan kelas kesalahan yang lebih luas daripada teknik kotak putih. Pengujian kotak hitam Sesuai dengan persyaratan masukan sesuai dengan persyaratan sesuai dengan persyaratan Program fungsional.



**Gambar – Black Box**

Pengujian black box adalah aspek mendasar dari pengujian sistem tanpa memperhatikan struktur logis internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak menggunakan dengan benar. Pengujian black box adalah metode desain data berbasis tes spesifikasi perangkat lunak. Uji data yang dihasilkan, jalankan di perangkat perangkat lunak dan kemudian output dari perangkat lunak diperiksa jika cocok seperti yang diharapkan.

Pengujian kotak hitam

1. fungsi yang salah atau hilang
2. Kesalahan antarmuka
4. kesalahan kinerja
5. kesalahan inisialisasi dan terminasi.

Berbeda dengan pengujian kotak putih, pengujian kotak hitam digunakan diterapkan di akhir ujian. Pengujian kotak hitam harus dimungkinkan jawab pertanyaan berikut:

- a. Apa validitas lisensi fungsional
- b. Kelas input apa yang akan membuat tes lebih baik

- c. Cara membatasi dari data yang terisolasi
- d. Kecepatan dan volume data apa yang akan ditoleransi system

Apa kombinasi data tertentu dengan sistem operasi.

### 3. Pengujian Berbasis Grafik

Langkah pertama dalam menguji kotak hitam adalah menguji suatu objek Ada dalam model perangkat lunak dan menentukan hubungan mana Dimaksudkan di antara benda-benda ini. Pengujian berdasarkan model grafis dilakukan pada perilaku sistem. Pengujian Grafis. Gambarlah grafik yang mewakili hubungan antara objek pada modul sehingga setiap objek dapat dipesan. Pengujian ini Dimulai dengan semua transisi, lalu jatuhkan lingkaran, di mana objek dan atribut diidentifikasi, juga memberikan indikasi titik awal dan berhenti.

Tabel – Notasi Pengujian berbasis graf

NOTASI	ARTI
	Simpul atau node Menggambarkan suatu objek
	Link Menggambarkan hubungan antar objek
	Node weight Menggambarkan properti atau nilai dari data
	Link weight Menggambarkan karakteristik link
	Link paralel Menggambarkan hubungan yang berbeda yang dibangun antar simpul
	Link simetris Menggambarkan hubungan dua arah antara dua objek

Ada beberapa pola berat tautan, yaitu:

- objek yang ditentukan

- Simetris, yaitu hubungan antara dua benda dalam dua arah
- Refleksif, yaitu hubungan yang mengarah ke simpul itu sendiri atau loop dibatalkan

Beizer membahas beberapa metode pengujian kotak hitam yang digunakan grafik, yaitu:

1. Transaction Flow Modeling, metode ini menggunakan simpul sebagai representasi langkah pada transaksi, dan tautan sebagai representasi hubungan logis antara langkah-langkah
2. Pemodelan keadaan terbatas, metode ini menggunakan node sebagai status dan tautan sebagai representasi. Statechart atau Grafik transfer status dapat digunakan untuk membuat grafik.
3. Pemodelan aliran data, metode ini menggunakan node sebagai repositori objek data dan tautan sebagai transformasi dari satu objek data ke objek data yang lain.
4. Pemodelan waktu, metode ini menggunakan simpul sebagai representasi objek program dan tautan sebagai hubungan berurutan antara objek.

### C. Partisi Kesetaraan

Partisi ekivalensi adalah metode yang membagi domain input dari a program ke dalam kelas data, tentukan pengujian dengan pengujian mengungkap kelas kesalahan, sehingga akan mengurangi jumlahnya Kasus pengujian keseluruhan. Jika tautan memiliki pola transitivitas, simetris, dan refleksif akan ada kelas ekivalensi. Kelas ekivalensi mewakili Provisi Provisi valid dan tidak valid untuk kondisi input. Terutama, kondisi input dapat berupa harga numerik, kisaran harga, Harga terkait, atau kondisi Boolean.

#### Penentuan Kelas Kesetaraan

1. Ketika Input menentukan kisaran, maka kelas yang setara valid dan dua tidak valid
2. Jika input membutuhkan harga khusus, maka satu kelas valid dan dua yang

setara tidak valid

3. Jika suatu kondisi menentukan anggota mana yang ditentukan, maka satu kelas  
Satu atau dua padanan tidak valid
4. Jika kondisi input adalah Boolean, maka satu kelas valid dan satu ditentukan  
ditentukan.

Contoh:

Nomor telepon untuk SLJJ

Kondisi:

Kode area: kosong atau ada 3-4 digit dimulai dengan 0

Awalan: 3 digit tanpa awalan dengan 1 atau 0

Sufi: 4 digit

Kondisi input:

Kode area: Kondisi input Boolean - tidak ada yang bisa ada

Kondisi rentang input - nilai > 200 atau nilai < 999

Awalan: rentang rentang kondisi - nilai > 200 atau nilai < 999

Suffix: Kondisi harga input - 4 digit

## B. Analisis Nilai Batas (Analisis Nilai Batas)

Analisis nilai batas adalah teknik desain proses yang melengkapi partisi kesetaraan, berdasarkan kesepakatan pada output domain. Pedoman untuk menentukan analisis nilai batas:

1. Jika input input adalah rentang yang ditentukan oleh nilai a dan b, pengujian harus dirancang dengan nilai a dan b, secara terus-menerus di atas dan di bawah ini dan sesuai
2. Jika input mengevaluasi jumlah input, maka uji harus dikembangkan menggunakan jumlah minimum dan Maksimum. Nilai tepat di atas dan di bawah minimum dan maksimum juga ditolak.
3. Pedoman 1 dan 2 juga berlaku untuk kondisi keluaran.
4. Jika struktur program data telah memerintahkan batas, maka Tes akan

dilakukan sesuai dengan ketentuan struktur data.

#### 2.1.2.3 Pengujian Komparatif

Tes yang diambil adalah metode pembuatan data yang dilakukan pada perangkat lunak yang dibuat berlebihan. Kekuatan Perangkat Lunak memiliki dua tim pengembangan, masing-masing berkembang Perangkat lunak individual untuk spesifikasi yang sama. Metode evaluasi digunakan untuk mengevaluasi data untuk salah satu perangkat lunak, yang kemudian digunakan sebagai input dalam tes perangkat lunak lain. Pengujian Komparatif digunakan untuk sistem yang menguji redundansi, Kasing uji dirancang untuk satu versi perangkat lunak yang dapat digunakan dalam menyetujui versi perangkat lunak lain, dan hasil dari versi kedua diharapkan perangkat lunaknya harus sama. Jika hasil pengujian kedua perangkat lunak ini berbeda maka kedua perangkat lunak akan digunakan untuk pencarian Shalat.

#### 4.1.1.1 Perangkat Lunak

Kebutuhan perangkat lunak dalam system ini adalah:

- a. Sistem Operasi : Windows
- b. Pemrograman : Bootstrap

#### 4.1.1.2 Perangkat Keras

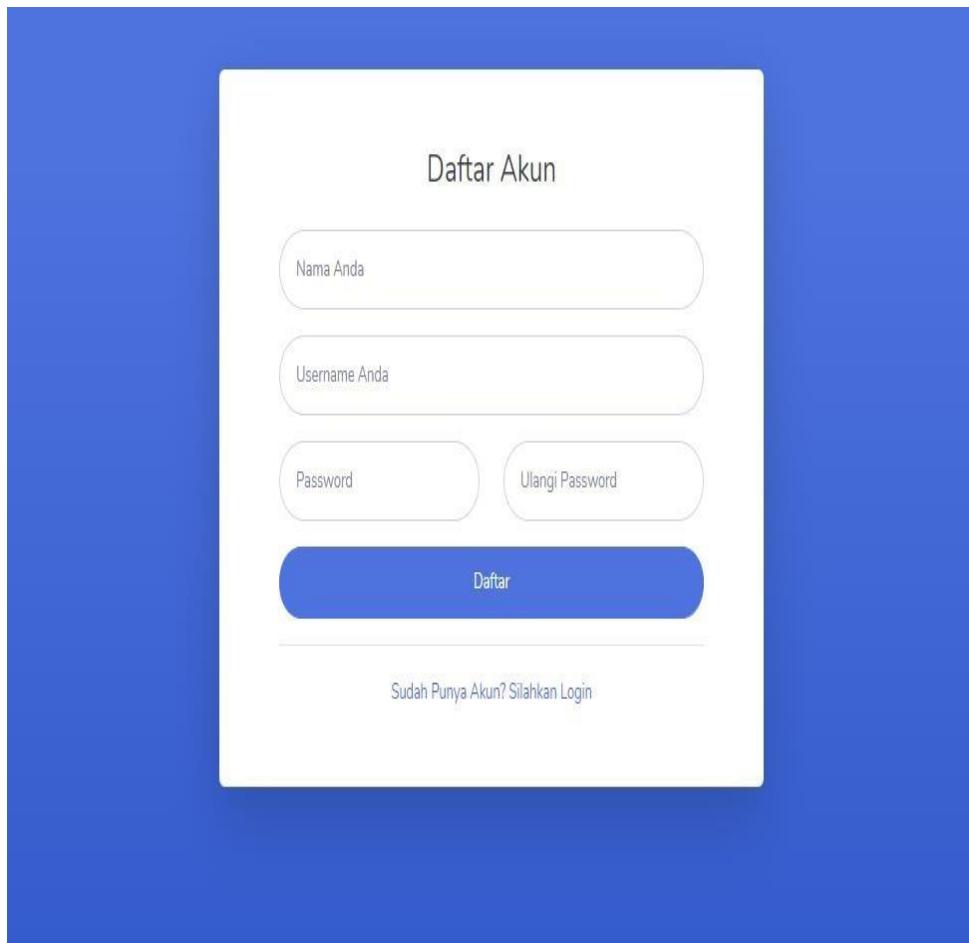
Kebutuhan perangkat lunak dalam system ini adalah:

- a. Prosesor : Intel® Core i5-8250U CPU @ 1.60GHz
- b. Memori : 4 GB

#### 4.1.2 Pengujian

### 4.2 Tampilan *User Interface*

#### 4.2.1 Tampilan Register

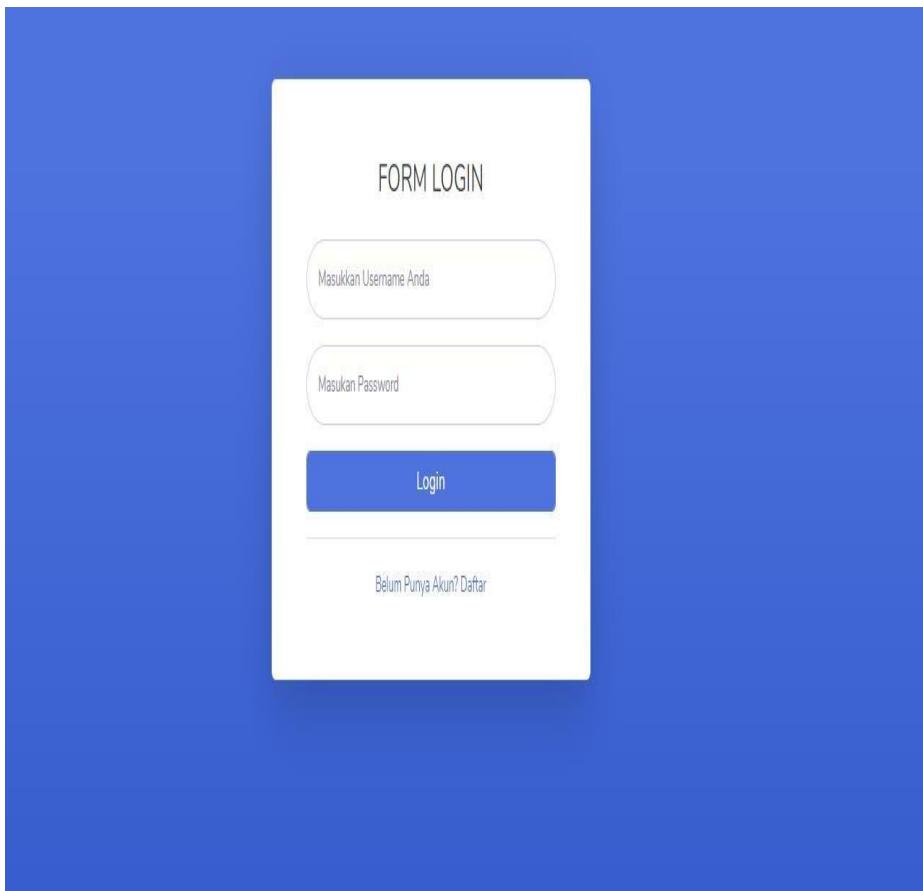


Gambar 4.1 Tampilan *Register*

**Penjelasan:**

pada UI di atas adalah UI untuk register *customer* dimana ketika *customer* ingin melakukan pembelian atau pemesanan pakaian *customer* harus mendaftarkan diri agar bisa login dengan akun masing-masing yang telah diregister.

**4.2.2 Tampilan Login**



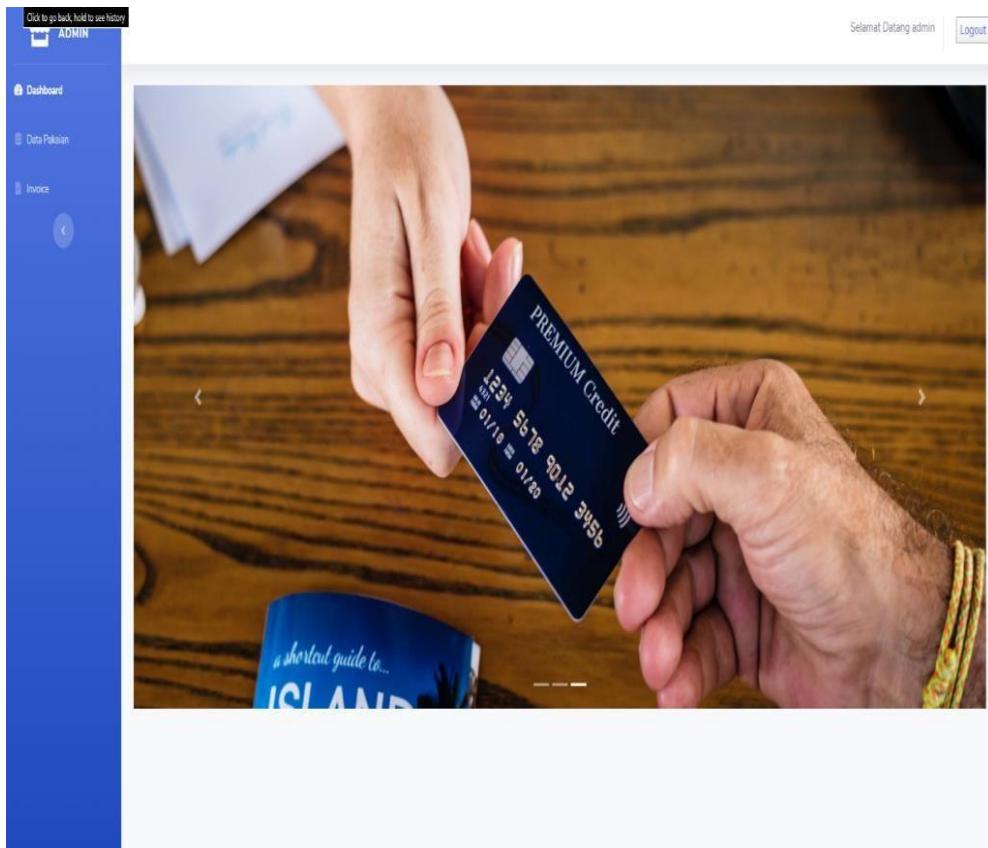
Gambar 4.2 Tampilan *Login*

**Penjelasan :**

pada tampilan di atas adalah tampilan untuk *login customer* dan *admin* dimana ketika *admin* dan *customer* ingin melakukan pembelian atau pemesanan pakaian *customer* dan *admin* harus login dengan akun masing-masing yang telah deregister.

**4.2.3 Tampilan Halaman Utama atau *Dashboard***

**4.2.3.1 Tampilan Halaman Utama *Dashboard Admin***

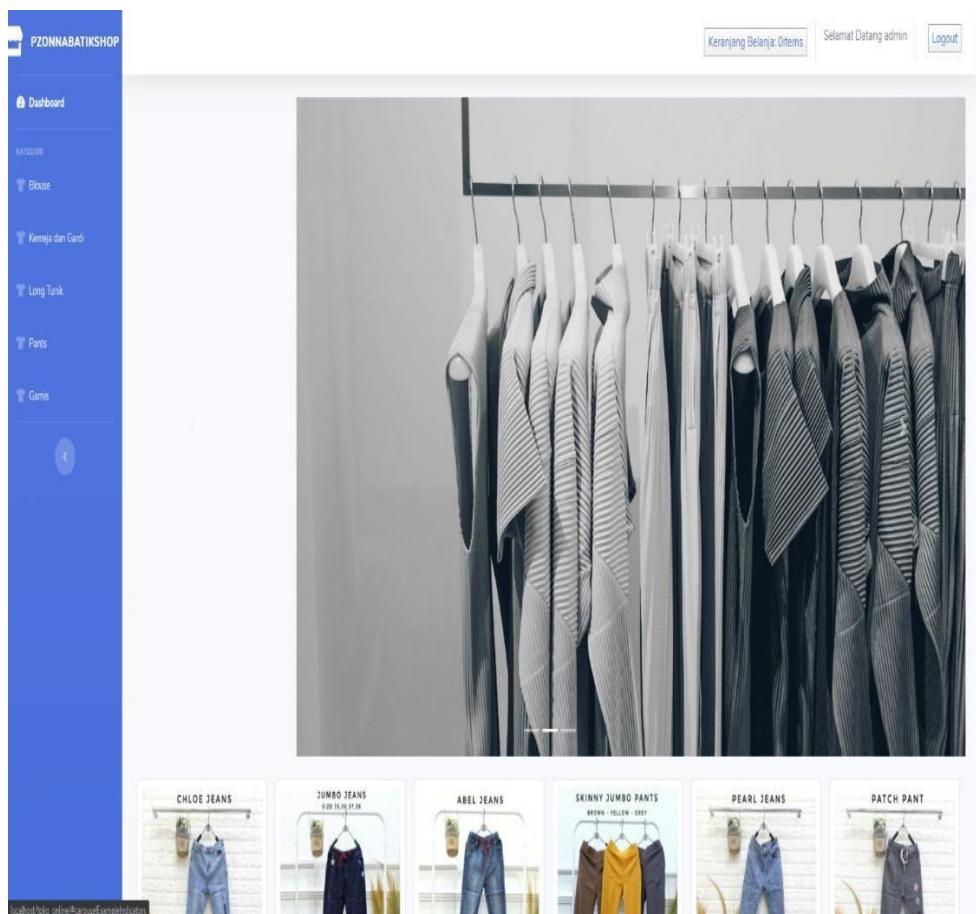


Gambar 4.3 Tampilan Halaman Utama *Dashboard Admin*

**Penjelasan:**

Pada halaman utama atau dashboard admin ini hanyalah tampilan biasanya untuk sambutan selamat datang atau tampilan memperindah saja yang isinya gambar-gambar slide

#### 4.2.3.2 Tampilan Halaman Utama Dashboard Customer

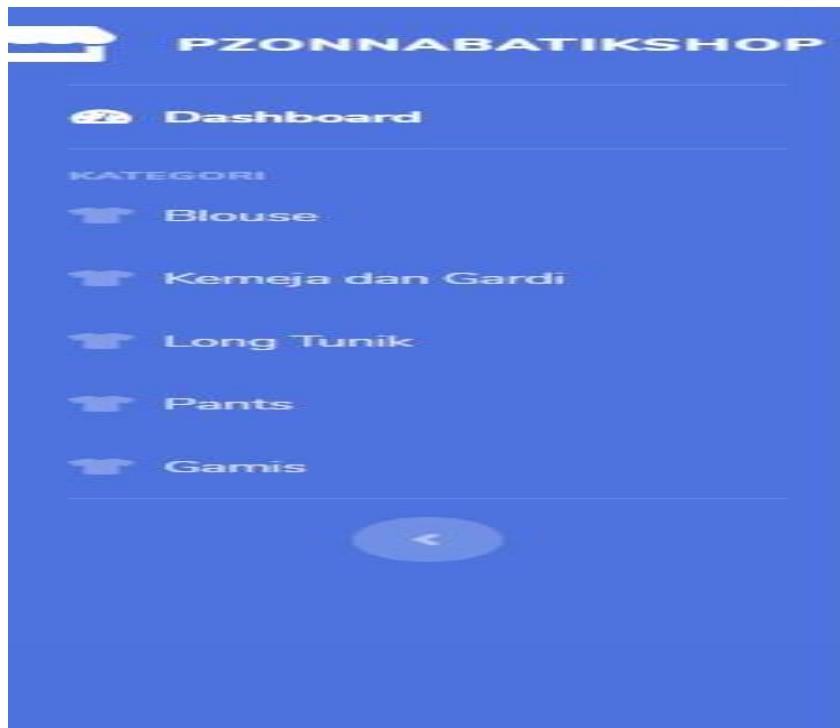


Gambar 4.4 Tampilan Halaman Utama *Dashboard Customer*

#### Penjelasan:

Tampilan diatas adalah gambaran dashboard untuk *customernya*, dimana isinya adalah produk-produk terbaru, dan promo, jadi *customer* bias melihat pada tampilan *dashboardnya*

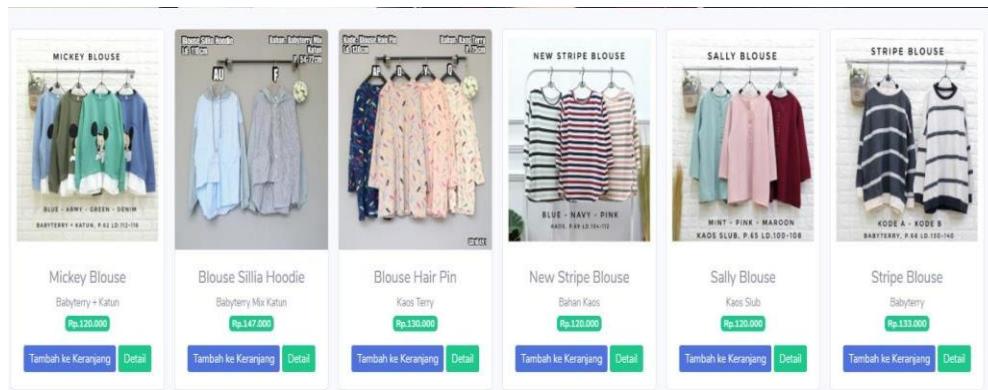
#### 4.2.4 Tampilan kategori



Gambar 4.5 Tampilan kategori

#### Penjelasan:

Tampilan di atas adalah tampilan untuk macam-macam kategori produk dimana ketika kita klik salah satu dari kategori diatas maka akan muncul pakaian dengan jenis kategori tersebut seperti misalnya kita klik kategori blouse maka akan tampil seperti dibawah ini



Gambar 4.6 Contoh Kategori

Dan pada button detail berisi detail pakaian seperti tampilan seperti dibawah ini:

Detail Pakaian

<b>MICKEY BLOUSE</b>	Nama Pakaian	Mickey Blouse
	Keterangan	Babyterry + Katun
	Kategori	Blouse
	Stok	20
	Harga	Rp.120.000
	<a href="#">Tambah ke Keranjang</a>	<a href="#">Kembali</a>

Gambar 4.7 Detail Pakaian

Jika di klik kembali maka akan kembali ke tampilan halaman utama, Pada button tambah keranjang jika di klik akan masuk ke dalam keranjang belanja

#### 4.2.5 Tampilan Keranjang Belanja

Keranjang Belanja		Keranjang Belanja: 1 items	Selamat Datang fili	Logout
NO	Nama Produk	Jumlah	Harga	Sub-Total
1	Mickey Blouse	1	Rp. 120.000	Rp. 120.000
				Rp. 120.000
<a href="#">Hapus Keranjang</a> <a href="#">Lanjutkan Belanja</a> <a href="#">Pembayaran</a>				

Gambar 4.8 Tampilan Keranjang Belanja

#### Penjelasan:

Pada tampilan diatas adalah tampilan keranjang belanja, tampilan keranjang belanja memiliki 3 button hapus keranjang lanjutkan belanja dan pembayaran. Jika salah satu kita klik maka akan masuk ke dalam tampilannya masing-masing

#### 4.2.6 Tampilan Pembayaran

Gambar 4.9 Ta

Total Belanja Anda: Rp. 120.000

### Input Pengiriman dan Pembayaran

Nama Lengkap

Nama Lengkap Anda

Alamat Lengkap

Alamat Lengkap Anda

No. Telepon

Nomor Telepon Lengkap Anda

Jasa Pengiriman

JNE

Pilih BANK

BCA - XXXXXXXX

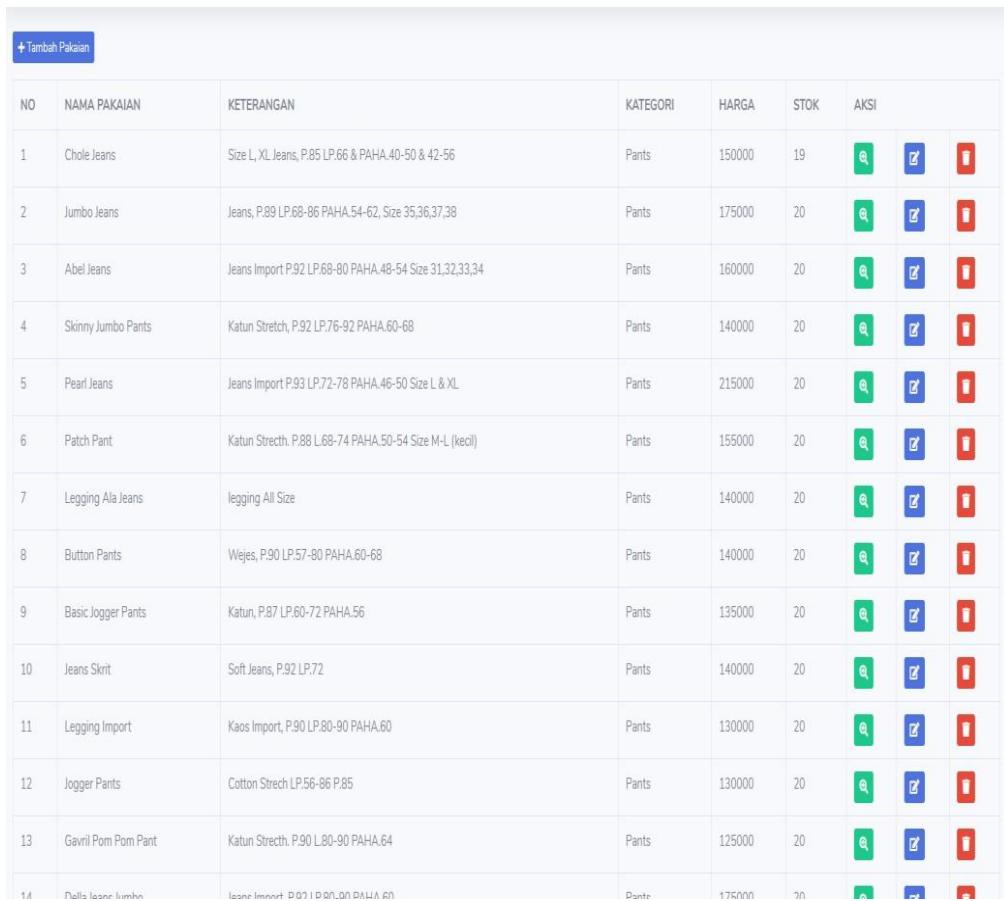
Pesan

### Tampilan Pembayaran

#### Penjelasan:

Pada tampilan diatas adalah tampilan pembayaran dimana kita bisa mengisi data lengkap kita dan jasa pengiriman yang kita inginkan, serta rekening bank yang harus kita transfer sesuai dengan pembelanjaan

#### 4.2.7 Tampilan Data Pakaian pada Admin



The screenshot shows a table of apparel items with the following data:

NO	NAMA PAKAIAN	KETERANGAN	KATEGORI	HARGA	STOK	AKSI
1	Chole Jeans	Size L, XL Jeans, P.85 LP.66 & PAHA.40-50 & 42-56	Pants	150000	19	
2	Jumbo Jeans	Jeans, P.89 LP.68-86 PAHA.54-62, Size 35,36,37,38	Pants	175000	20	
3	Abel Jeans	Jeans Import P.92 LP.68-80 PAHA.48-54 Size 31,32,33,34	Pants	160000	20	
4	Skinny Jumbo Pants	Katun Stretch, P.92 LP.76-92 PAHA.60-68	Pants	140000	20	
5	Pearl Jeans	Jeans Import P.93 LP.72-78 PAHA.46-50 Size L & XL	Pants	215000	20	
6	Patch Pant	Katun Strech, P.88 LP.68-74 PAHA.50-54 Size M-L (kecil)	Pants	155000	20	
7	Legging Ala jeans	legging All Size	Pants	140000	20	
8	Button Pants	Wejes, P.90 LP.57-80 PAHA.60-68	Pants	140000	20	
9	Basic Jogger Pants	Katun, P.87 LP.60-72 PAHA.56	Pants	135000	20	
10	Jeans Skrit	Soft Jeans, P.92 LP.72	Pants	140000	20	
11	Legging Import	Kaos Import, P.90 LP.80-90 PAHA.60	Pants	130000	20	
12	Jogger Pants	Cotton Strech LP.56-86 P.85	Pants	130000	20	
13	Gavril Pom Pom Pant	Katun Strech, P.90 LP.80-90 PAHA.64	Pants	125000	20	
14	Polla loose lumha	jeans Import P.92 LP.60,60 PAHA.60	Pants	175000	20	

Gambar 4.10 Tampilan Data Pakaian pada Admin

### Penjelasan:

Pada tampilan diatas adalah tampilan dimana kita bisa melihat data-data pakaian yang telah kita input, dan juga kita bisa mengedit datanya serta menghapus datanya sesuai dengan keinginan admin pada button tambah pakaian diatas disitulah tempat kita menambah data-data pakaian

#### 4.2.8 Tampilan Input Data Pakaian

The screenshot shows a modal window titled "FORM INPUT PAKAIAN". The window contains the following fields:

- Nama Pakaian: An input field.
- Keterangan: An input field.
- Kategori: A dropdown menu currently set to "Blouse".
- Harga: An input field.
- Stok: An input field.
- Gmbar Pakaian: A file input field with the placeholder "Choose File No file chosen".

At the bottom right of the modal are two buttons: "Close" (red) and "Simpan" (blue).

Gambar 4.11 Tampilan Input Data Pakaian

#### Penjelasan:

Tampilan diatas adalah tampilan dimana admin menginput data pakaian yang akan tampil pada data pakaian admin, dan kategori customer

#### 4.2.9 Tampilan Invoice

Invoice Pemesanan Produk					
Id Invoice	Nama Pemesan	Alamat Pengiriman	Tanggal Pemesanan	Batas Pembayaran	Aksi
28			2019-12-25 20:32:27	2019-12-26 20:32:27	<button>Detail</button>
29	a	a	2020-01-09 14:53:57	2020-01-10 14:53:57	<button>Detail</button>
30	dini	jl.sariasih	2020-01-10 09:53:01	2020-01-11 09:53:01	<button>Detail</button>

Gambar 4.12 Tampilan Invoice

### Penjelasan:

Tampilan di atas adalah tampilan untuk history bagi customer yang sudah membayar belanjaan dan detail pesanannya

## Bab 5

---

### TEORI BASIS DATA

---

#### 5.1 Teori Dasar

##### 5.1.1 Basis Data

Basis Data adalah kumpulan data yang saling berhubungan secara logikal serta deskripsi dari data tersebut, yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Basis Data adalah sebuah penyimpanan data yang besar yang bisa digunakan oleh banyak pengguna dan departemen. Semua data terintegrasi dengan jumlah duplikasi yang minimum. Basis Data tidak lagi dipegang oleh satu departemen, tetapi dibagikan ke seluruh departemen pada perusahaan. Basis Data itu sendiri tidak hanya memegang data operasional organisasi tetapi juga penggambaran dari data tersebut (Connolly & Begg, 2010:64).

Basis data adalah kumpulan data store yang terintegrasi yang diatur dan di kontrol secara sentral. Sebuah basis data biasanya menyimpan ribuan class. Informasi yang disimpan termasuk class attribute dan relasi antar class. Basis data juga menyimpan informasi yang dekriptif seperti nama atribut, pemberian batasan suatu nilai, dan kontrol akses untuk data-data yang sensitif (Satzinger, Robert, & Stephen, 2005: 398). Basis data juga diartikan sebagai sekumpulan file dikomputer yang saling terhubung. File file ini diatur sesuai kesamaan elemennya, sehingga data yang diinginkan dapat dicari secara mudah (Williams

& Sawyer, 2007: 181)

Basis data adalah dua atau lebih simpanan data dengan elemenelemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan teknik-teknik formal dan manajemen basis data. Dari definisi diatas, maka dapat dikatakan bahwa basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya (Iskandar & Rangkuti, 2008: 3). Basis data adalah dua atau lebih simpanan data dengan elemen-elemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan teknik-teknik formal dan manajemen basis data (Abdillah, 2012: 1).

Dapat disimpulkan basis data adalah penyimpanan data yang terstruktur, terintegrasi dan saling berkaitan dengan elemen-elemen penghubungnya dan dapat di akses dengan berbagai cara, oleh karena itu basis data juga bisa didefinisikan sebagai kumpulan yang menggambarkan sendiri dari catatan yang terintegrasi dan penggambaran dari data dikenal sebagai sistem katalog (atau kamus data atau metadata). Definisi data disini dibedakan dari program aplikasi, yang umumnya sama dengan pendekatan pengembangan modern perangkat lunak, dimana definisi internal dan eksternal dari sebuah objek dipisahkan. Salah satu keuntungan dari pendekatan tersebut adalah abstraksi data dimana kita dapat mengubah definisi internal dari sebuah objek tanpa mempengaruhi pengguna dari objek jika definisi eksternal objek tersebut tidak berubah.

### **5.1.2 *Database Management System (DBMS)***

Database Management System adalah sistem perangkat lunak yang memungkinkan pengguna dapat mendefinisikan, membuat, merawat, dan mengatur akses ke Basis Data. Biasanya DBMS memungkinkan pengguna untuk mendefinisikan Database melalui sebuah Data Definition Language (DDL), menspesifikasikan tipe data, struktur dan batasan pada data yang

disimpan pada Database. Kemudian juga memungkinkan insert, update, delete, dan mengambil data dari Database melalui Data Manipulation Language (DML), mempunyai pusat penyimpanan untuk semua data dan deskripsi data memungkinkan DML untuk menyediakan fasilitas umum untuk data tersebut yang umumnya disebut bahasa query (Connolly & Begg, 2010:6).

### **5.1.2.1 Keuntungan DBMS**

Berikut ini adalah beberapa keuntungan dari Database Management System, yakni:

1. Kontrol terhadap redudansi data Sistem berbasis file tradisional membuang tempat penyimpanan dengan menyimpan informasi yang sama lebih dari satu file.
2. Konsistensi data Dengan menghilangkan atau mengendalikan redudansi, kita bisa mengurangi resiko dari inkonsistensi yang akan terjadi. Apabila sebuah item data disimpan hanya sekali dalam Database, jika terjadi pembaruan pada nilainya yang harus dilakukan hanya sekali maka nilai yang baru tersebut akan langsung bisa digunakan untuk semua pengguna.
3. Lebih banyak informasi dari sumber yang sama. Dengan integrasi dari data operasional, memungkinkan bagi organisasi untuk mengambil data tambahan dari informasi yang sama.
4. Pembagian Data Biasanya file dimiliki oleh departemen atau yang menggunakan. Dilain hal, Database seharusnya berada diseluruh organisasi dan bisa di-share pada seluruh pengguna yang diizinkan.
5. Meningkatkan integritas data Integritas Database mengacu pada validitas dan konsistensi data yang tersimpan.
6. Meningkatkan keamanan Keamanan Database adalah perlindungan dari Database dari pengguna yang tidak sah. Tanpa langkah-langkah

keamanan yang sesuai, integrasi membuat data lebih rentan daripada sistem Basis Data.

7. Penegakan Standar Integrasi memungkinkan DBA untuk mendefinisikan dan menegakan standar yang diperlukan. Termasuk departemen, organisasi, standar nasional, atau standar internasional untuk hal-hal seperti format data untuk memfasilitasi pertukaran data antara sistem, konvensi penamaan, standar dokumentasi, prosedur update, dan aturan akses.
8. Skala Ekonomi Menggabungkan semua data operasional organisasi ke dalam satu Database, dan menciptakan kumpulan aplikasi yang bekerja pada salah satu sumber data, yang berdampak pada penghematan biaya.
9. Keseimbangan pada persyaratan yang bertentangan Setiap pengguna atau departemen memiliki kebutuhan yang mungkin bertentangan dengan kebutuhan pengguna lain. Karena Database berada di bawah kendali DBA, DBA dapat membuat keputusan mengenai penggunaan desain dan operasional dari Database yang menyediakan penggunaan terbaik dari sumber daya bagi organisasi secara keseluruhan.
10. Meningkatkan aksebilitas data dan data responsif Sebagai akibat dari integrasi, data yang melintasi batas-batas departemen menjadi dapat diakses secara langsung ke pengguna akhir. Dengan demikian menyediakan suatu sistem dengan potensi yang lebih banyak mengenai fungsionalitas, misalnya, dapat digunakan untuk memberikan layanan yang lebih baik kepada pengguna akhir atau klien organisasi.
11. Peningkatan Produktifitas DBMS menyediakan banyak fungsi standar yang biasanya seorang programmer harus menulis dalam aplikasi berbasis file. Pada tingkat dasar, DBMS menyediakan semua

rutinitas penanganan file tingkat rendah yang khas dalam program aplikasi. Penyediaan dari fungsi tersebut memungkinkan programmer untuk berkonsentrasi pada fungsi khusus yang diperlukan oleh pengguna tanpa harus khawatir tentang detil implementasi tingkat rendah.

12. Peningkatan pemeliharaan melalui independensi data Dalam sistem berbasis file, deskripsi data dan logika untuk mengakses data dibangun ke dalam setiap program aplikasi, menjadikan program berketergantungan pada data. Perubahan struktur data, misalnya membuat alamat 41 karakter bukan 40 karakter, atau perubahan dengan bagaimana cara data disimpan pada disk, memerlukan perubahan besar untuk program yang terpengaruh oleh perubahan.

13. Peningkatan Konkurensi Dalam beberapa sistem berbasis file, jika dua atau lebih pengguna diizinkan untuk mengakses file yang sama secara bersamaan, sangat mungkin akses tersebut saling mengganggu satu sama lain, sehingga dapat menyebabkan hilangnya informasi atau bahkan hilangnya integritas.

14. Peningkatan backup dan Jasa Pemulihan Banyak system berbasis file menempatkan tanggung jawab pada pengguna untuk memberikan langkah-langkah untuk melindungi data dari kerusakan pada sistem komputer atau program aplikasi. Ini mungkin melibatkan backup dari data semalam. Jika terjadi kerusakan di keesokan harinya, backup dipulihkan dan pekerjaan yang telah dikerjakan sebelum backup ini hilang dan harus kembali dikerjakan. (Connolly & Begg, 2010: 77-80)

### **5.1.2.2 Kerugian DBMS**

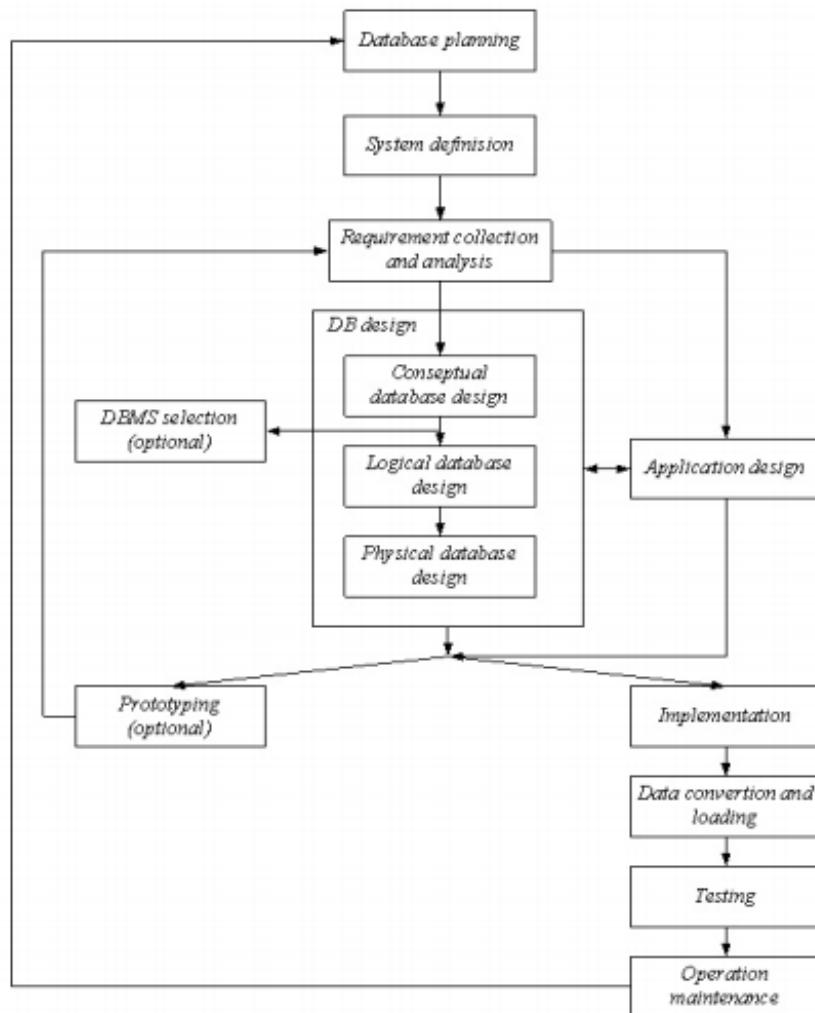
Disamping keuntungan yang begitu banyak akan manfaat, DBMS juga mempunyai kerugian. Berikut adalah pembahasan mengenai kerugian dari DBMS :

1. Kompleksitas Penyediaan fungsi yang diharapkan dari DBMS yang baik membuat DBMS menjadi bagian yang sangat kompleks dari perangkat lunak. Desainer Basis Data dan developer, data dan Database administrator, dan pengguna akhir harus memahami fungsi tersebut untuk bisa mengambil keunggulan secara penuh. Kegagalan untuk memahami sistem dapat mengarah pada keputusan desain yang buruk, yang nantinya menjadi konsekuensi serius bagi suatu organisasi.
2. Ukuran Kompleksitas dan luasnya fungsionalitas membuat DBMS menjadi bagian software yang sangat besar, menggunakan banyak megabytes pada ruang disk dan membutuhkan sejumlah besar memori untuk menjalankannya secara efisien.
3. Biaya dari DBMS Biaya DBMS bervariasi, tergantung pada lingkungan dan fungsi yang disediakan.
4. Biaya Tambahan Perangkat Keras Persyaratan penyimpanan disk untuk DBMS dan Database mungkin memerlukan pembelian disk tambahan untuk memperbanyak tempat penyimpanan.
5. Biaya Konversi Dalam beberapa situasi, biaya perangkat keras dari DBMS dan perangkat keras tambahan mungkin tidak signifikan dibandingkan dengan biaya konversi aplikasi yang ada untuk dijalankan pada DBMS dan perangkat keras baru.
6. Performa Sebuah sistem berbasis file ditulis untuk aplikasi tertentu, seperti faktur. Sebagai hasilnya, kinerja umumnya sangat baik. Namun, DBMS ditulis lebih umum, untuk melayani banyak aplikasi bukan hanya satu. Efeknya adalah bahwa beberapa aplikasi tidak dapat berjalan secepat yang biasanya mereka gunakan.
7. Dampak dari kerusakan yang tinggi Pemusatkan sumber daya meningkatkan kerentanan pada sistem. Karena semua pengguna dan

aplikasi bergantung pada ketersediaan DBMS, kerusakan komponen tertentu dapat menyebabkan operasi tidak jalan. (Connolly & Begg, 2010:80-81)

### 5.1.3 Siklus Hidup Basis Data

Dalam basis data juga terdapat siklus hidup Database yang digambarkan sebagai berikut :



Gambar 5.1 Siklus Hidup Basis Data

Berikut ini adalah penjelasan mengenai tahap-tahap dari gambar siklus hidup Database di atas disertai dengan faktor-faktor dan langkah-langkah pendukung lainnya :

- **Tahap 1 Perencanaan Basis Data (*Database Planning*)**

Kegiatan manajemen yang memungkinkan tahapan dari siklus hidup pengembangan sistem Database untuk direalisasikan secara efisien dan seefektif mungkin. Perencanaan Database harus terintegrasi dengan strategi sistem informasi dalam organisasi.

Tiga hal yang terlibat dalam merumuskan strategi sistem informasi :

1. Mengidentifikasi rencana dan tujuan perusahaan yang menerapkan kebutuhan sistem informasi.
2. Mengevaluasi sistem informasi yang ada untuk memastikan kelebihan dan kekurangannya.
3. Menilai kesempatan teknologi informasi yang mungkin menghasilkan keunggulan kompetitif.

Langkah penting dalam perencanaan Database adalah mendefinisikan mission statement untuk sistem Basis Data dan mendefinisikan mission objectives.

Fungsi mission statement adalah sebagai berikut :

1. Mendefinisikan tujuan utam dari pengembangan Database.
2. Membantu memperjelas atau mengklarifikasi proyek Database.
3. Menyediakan arah yang jelas terhadap aplikasi Database yang akan dibangun.

Mission objective mengidentifikasi bagian dari tugas-tugas utama yang harus didukung.

- **Tahap 2 Definisi Sistem**

Definisi Sistem menggambarkan ruang lingkup dan batasan aplikasi

Database yang akan dibuat dan diterjemahkan ke major *user view*. *User View* adalah mendefinisikan apa yang akan dibutuhkan dari sistem Basis Data dari perspektif peran pekerjaan tertentu (seperti manajer atau supervisor) atau area aplikasi perusahaan (seperti marketing, personalia dan bagian duang.)

#### • **Tahap 3 Pengumpulan Persyaratan dan Analisis**

Proses mengumpulkan dan menganalisis informasi tentang bagian dari organisasi yang harus didukung oleh sistem Basis Data dan menggunakan informasi tersebut untuk mengidentifikasi persyaratan untuk sistem yang baru. Tahap ini melibatkan pengumpulan dan analisis informasi tentang bagian dari perusahaan yang akan didukung oleh basis data. Banyak teknik digunakan untuk mendapatkan informasi tersebut yang disebut teknik fact-finding. Teknik fact-finding adalah teknik mengumpulkan fakta untuk merancang basis data.

Informasi atau fakta yang dikumpulkan adalah :

1. Deskripsi dari data yang digunakan atau yang dihasilkan.
2. Rincian tentang bagaimana data akan digunakan atau dihasilkan.
3. Persyaratan tambahan untuk sistem Database yang baru. Informasi tersebut kemudian dianalisa untuk mengidentifikasi persyaratan untuk dimasukkan ke dalam sistem Database yang baru.

#### • **Tahap 4 Desain Basis Data Konseptual**

Suatu kegiatan untuk membangun model semua data yang digunakan oleh perusahaan dan tidak bergantung dari semua pertimbangan fisik. Model data dibangun menggunakan informasi terdokumentasi dalam spesifikasi kebutuhan pengguna. Desain Basis Data Konseptual sepenuhnya tidak bergantung pada target DBMS, program aplikasi, perangkat keras.

Langkah-langkah pada tahap Desain Basis Data Konseptual :

1. Mengidentifikasi tipe entitas.

Tujuan pada tahap ini adalah untuk mengidentifikasi jenis entitas yang dibutuhkan. Langkah pertama membangun model data konseptual adalah untuk menentukan dan mendefinisikan objek utama. Objek-objek tersebut adalah entitas untuk model data. Salah satu metode untuk mengidentifikasi entitas adalah memeriksa spesifikasi kebutuhan pengguna.

2. Mengidentifikasi Tipe Relasi

Tujuan pada tahap ini adalah untuk mengidentifikasi hubungan penting yang ada antara jenis entitas.

Langkah-langkah yang dilakukan :

a. Menggunakan E-R Diagram

E-R Diagram digunakan untuk menggambarkan entitas data hubungan antara satu entitas dengan yang lain.

b. Menentukan Batasan Multiplisitas Dari Tipe Relasi

Batasan multiplisitas digunakan untuk memeriksa dan menjaga kualitas data.

c. Memeriksa fan dan chasm trap.

Fan trap adalah model yang merupakan hubungan antara tipe entitas, tetapi jalur antara kejadian entitas tertentu tidak ada. Chasm trap adalah model yang menunjukkan adanya hubungan antara tipe entitas tetapi jalur antara kejadian entitas tertentu tidak ada.

d. Mendokumentasi Tipe Relasi

3. Mengidentifikasi dan mengasosiasikan atribut dengan entitas atau tipe relasi.

Tujuan pada tahap ini adalah untuk mengasosiasikan atribut dengan entitas atau tipe hubungan yang sesuai. Langkah berikut dalam metodologi ini adalah untuk mengidentifikasi jenis fakta tentang entitas dan relasi yang telah dipilih untuk diwakili dalam Database.

#### 4. Menentukan domain atribut.

Tujuan pada tahap ini adalah untuk menentukan domain untuk atribut dalam model data konseptual. Tujuan dari langkah ini adalah untuk menentukan domain untuk semua atribut dalam model.

#### 5. Menentukan attribute candidate, primary key dan alternate key.

Tujuan pada tahap ini adalah untuk mengidentifikasi *candidate key* dari setiap jenis entitas, jika ada lebih dari satu *candidate key* maka dipilih salah satu menjadi primary key dan yang lainnya sebagai *alternate key*.

#### 6. Mempertimbangkan penggunaan konsep pemodelan ditingkatkan.

Tujuan pada tahap ini adalah untuk mempertimbangkan penggunaan konsep pemodelan seperti spesialisasi atau generalisasi, agregasi dan komposisi.

#### 7. Periksa model untuk redudansi

Tujuan pada tahap ini adalah untuk memeriksa adanya redudansi apapun dalam model. Ada tiga kegiatan dalam langkah ini :

- Memeriksa kembali hubungan one-to-one
- Menghapus relasi yang redudansi
- Mempertimbangkan dimensi waktu

#### 8. Memvalidasi model data konseptual terhadap transaksi pengguna

Tujuan pada tahap ini adalah untuk memastikan bahwa model data konseptual mendukung transaksi yang dibutuhkan. Dalam menggunakan model model inim, akan dilakukan operasi secara manual. Namun jika tidak dapat melakukan transaksi secara manual artinya ada masalah

dengan model data yang harus diselesaikan. Ada dua langkah untuk memastikan bahwa model data konseptual mendukung transaksi yang diperlukan :

- Menggambarkan transaksi
- Menggunakan jalur transaksi

#### 9. Memeriksa data model konseptual dengan pengguna

Tujuan pada tahap ini adalah untuk memeriksa model data konseptual dengan pengguna untuk memastikan bahwa mereka menganggap model adalah “benar” merupakan representasi dari persyaratan data perusahaan. Model data konseptual mengandung E-R Diagram dan dokumentasi pendukung yang mendeskripsikan model data. Jika terdapat anomaly dalam model data, harus membuat perubahan yang sesuai yang mungkin memerlukan untuk mengulangi langkah sebelumnya.

#### • **Tahap 5 Desain Basis Data Logikal**

Untuk menerjemahkan model data konseptual ke dalam model data logikal dan kemudian untuk memvalidasi model ini untuk memeriksa bahwa secara struktural benar dan mampu mendukung transaksi yang diperlukan. Pada tahap ini bertujuan membuat desain basis data logikal dari model data konseptual terdiri dari beberapa tahap yakni :

##### 1. Mendapatkan relasi untuk model data logical

Tujuan pada tahap ini adalah untuk membuat relasi untuk model data logical yang mewakili entitas, relasi, relasi dan atribut yang telah diidentifikasi. Ada Sembilan hal yang menggambarkan bagaimana hubungan yang diturunkan untuk struktur yang mungkin terjadi dalam model data konseptual :

- *Strong entity types*

- *Weak entity types*
- *One to many binary relationship types*
- *One to one binary relationship types*
- *Superclass / subclass relationship types*
- *Many to many binary relationship types*
- *Complex relationship types*
- *Multi-valued attributes*

## 2. Memvalidasi relasi menggunakan normalisasi

Tujuan pada tahap ini adalah memvalidasi relasi pada model data logical dan untuk memastikan bahwa relasi memiliki sejumlah atribut yang diperlukan untuk mendukung kebutuhan data perusahaan dan relasi juga harus memiliki redundansi minimal data untuk menghindari masalah anomali.

## 3. Memvalidasi relasi terhadap transaksi pengguna

Tujuan pada tahap ini adalah untuk memastikan bahwa relasi dalam model data logikal mendukung transaksi yang diperlukan seperti yang dirincikan dalam *user requirements specifications*. Pada tahap ini juga diperiksa apakah relasi yang dibuat pada langkah sebelumnya juga mendukung transaksi dan memastikan bahwa tidak ada kesalahan pada saat menciptakan relasi.

## 4. Cek batasan integritas.

Tujuan pada tahap ini adalah untuk memeriksa apakah batasan integritas terwakili dalam model data logikal. Jenis batasan integritas adalah :

- Data yang diperlukan.
- Batasan atribut *domain*.
- *Multiplisitas*.

- *Integritas entitas.*
- Integritas referensial.
- Batasan Umum.

#### 5. Memeriksa kembali model data logikal dengan pengguna

Tujuan pada tahap ini adalah untuk memeriksa model data logikal dengan pengguna untuk memastikan bahwa model yang dibuat adalah representasi yang benar dari persyaratan data perusahaan. Jika pengguna tidak puas dengan model, maka beberapa pengulangan langkah sebelumnya dalam metodologi mungkin diperlukan.

#### 6. Menggabungkan model data logikal ke dalam model global (langkah opsional)

Tujuan pada tahap ini adalah untuk menggabungkan model data *logical* ke dalam model data *logikal global* yang mewakili pandangan semua pengguna *Database*. Model data logikal lokal mewakili satu atau lebih pandangan pengguna dari semua *Database* sedangkan model data logikal global mewakili pandangan semua pengguna *Database*. Meskipun masing-masing model adalah representasi hanya dari satu atau lebih tetapi tidak pandangan semua *Database*. Beberapa aktivitas dalam pendeketan ini :

- Memeriksa ulang nama dan isi dari entitas atau relasi dan *candidate key*-nya.
- Memeriksa nama dan isi dari relasi atau *foreign key*.
- Menggabungkan entitas atau relasi dari model data lokal.
- Menyertakan (tanpa penggabungan) entitas atau relasi unik untuk setiap model data lokal.
- Menggabungkan relasi atau *foreign key* dari model data lokal.
- Menyertakan (tanpa penggabungan) relasi atau *foreign key* yang unik

untuk setiap model data lokal.

- Memeriksa entitas atau relasi atau *foreign key* yang hilang.
- Memeriksa *foreign key*.
- Memeriksa batasan integritas.
- Menggambar diagram global ER
- Memperbarui dokumentasi

#### 7. Cek untuk pertumbuhan dimasa yang akan datang

Tujuan pada tahap ini adalah untuk menentukan apakah ada perubahan signifikan kemungkinan di masa mendatang dan untuk menilai apakah model data logikal dapat mengakomodasi perubahan-perubahan. Jika model yang dibuat hanya menjawab kebutuhan pada saat ini saja, maka model hanya memiliki ‘hidup’ yang singkat dan membutuhkan perubahan-perubahan untuk menjawab kebutuhan-kebutuhan yang baru di masa yang akan datang.

#### • **Tahap 6 Desain Basis Data Fisik**

Proses menghasilkan deskripsi implementasi *Database* pada penyimpanan sekunder yang menggambarkan relasi dasar, organisasi file dan indeks yang digunakan untuk mencapai akses yang efisien ke data, setiap batasan integritas terkait dan langkah-langkah kemanan.

##### 1. Menerjemahkan model data logikal untuk target DBMS

- Mendesain relasi dasar.
- Mendesain representasi dari *derived data*
- Mendesain batasan umum.

##### 2. Mendesain file organisasi dan indek

Tujuan pada tahap ini adalah untuk menentukan file organisasi yang optimal untuk menyimpan basis relasi dan indek yang diperlukan untuk mencapai kinerja yang dapat diterima Pada tahap ini dibagi menjadi :

- Menganalisis Transaksi.
- Memilih File Organisasi.
- Memilih Indeks
- Mengestimasi kebutuhan kapasitas penyimpanan

### 3. Mendesain pandangan pengguna

Tujuan pada tahap ini adalah untuk merancang pandangan pengguna yang diidentifikasi selama tahap pengumpulan dan analisis persyaratan sistem siklus hidup pengembangan *Database*.

### 4. Mendesain mekanisme Keamanan

Tujuan pada tahap ini adalah untuk mendesain mekanisme keamanan untuk *Database* seperti yang ditentukan oleh pengguna selama tahap persyaratan dan pengumpulan dari sistem siklus hidup pengembangan *Database*. *Database* merupakan sumber daya perusahaan yang penting dan keamanan sumber daya ini sangat penting. Tujuan pada tahap ini adalah untuk memutuskan bagaimana persyaratan keamanan akan terwujud. Beberapa sistem menawarkan fasilitas keamanan yang berbeda dengan yang lainnya. Ada dua sistem yang harus diterapkan yaitu :

- Keamanan Sistem
- Keamanan Data

### 5. Mempertimbangkan Pengenalan Kontrol Redudansi

Tujuan pada tahap ini adalah untuk menentukan redundansi dalam cara yang dikendalikan, dengan cara melonggarkan normalisasi akan meningkatkan kinerja sistem. Salah satu tujuan dasar dari desain *Database* relasional adalah mengelompokkan atribut bersama-sama dalam satu relasi karena ada functional dependency diantara itu. Hasil dari normalisasi adalah desain basis data logikal yang secara struktural

konsisten dan memiliki redundansi minimal.

## 6. Memonitor Dan Menyesuaikan Sistem Operasional

Tujuan pada tahap ini adalah memonitor sistem operasional dan meningkatkan kinerja dari sistem untuk memperbaiki desain yang tidak pantas atau mencerminkan kebutuhan perubahan. Pada aktifitas ini, harus diingat bahwa salah satu tujuan utama dari mendesain Basis Data fisik adalah untuk menyimpan dan mengakses data dalam cara yang efisien. Ada beberapa faktor yang digunakan untuk mengukur efisiensi :

- *Transaksi Throuhput*
- Waktu Respon
- Kapasistas Penyimpanan
- Penyesuaian *Database* dapat mencegah pembelian perangkat keras tambahan.
- Memungkinkan untuk menghemat konfigurasi perangkat keras.
- Penyesuaian yang baik menghasilkan waktu respon yang cepat dan hasil yang baik dan berdampak pada peningkatan produktivitas organisasi.
- Meningkatnya waktu respon dapat meningkatkan moral karyawan.
- Meningkatnya waktu respon dapat meningkatkan respon pelanggan.

## • **Tahap 7 Pemilihan DBMS**

Tujuan pada tahap ini adalah pemilihan DBMS yang tepat untuk mendukung sistem Basis Data. Langkah-langkah yang dilakukan adalah sebagai berikut :

1. Mendefinisikan kerangka acuan studi.
2. Daftar dua atau tiga produk.
3. Mengevaluasi produk.

## • **Tahap 8 Desain Aplikasi**

Tujuan pada tahap ini adalah mendesain antarmuka pengguna dan aplikasi yang digunakan dan memproses sistem Basis Data. Namun sebelum tahap ini dilakukan, desain Database harus sudah jadi. Pada tahap ini juga harus dipastikan bahwa semua fungsi yang dinyatakan dalam spesifikasi kebutuhan pengguna harus ditampilkan dalam desain aplikasi untuk sistem Basis Data. Ada dua aspek dalam mendesain aplikasi :

### 1. Desain Transaksi

Tujuan dari desain interaksi adalah untuk menentukan dan mendokumentasikan karakteristik tingkat tinggi dari transaksi yang diperlukan pada *Database* termasuk data yang digunakan oleh transaksi, fungsional karakteristik transaksi dan hasil dari transaksi.

### 2. Pedoman mendesain antarmuka pengguna.

Sebelum mengimplementasikan *form* atau laporan, penting untuk mendesain *layout*. Pedoman yang perlu diperhatikan yaitu :

- Judul yang bermakna
- Instruksi yang mudah dipahami
- Pengelompokkan secara logis dan urutan isian
- Tata letak *form* yang menarik secara *visual*
- Label isian yang akrab dengan pengguna
- Terminologi dan singkatan yang konsisten
- Penggunaan warna yang konsisten.
- Terlihat kurang dan batas-batas untuk menginput kotak isian data
- Gerakan kursor yang nyaman
- Koreksi kesalahan
- Pesan kesalahan untuk nilai yang tidak diijinkan
- Pengisian opsional diberi tanda dengan jelas

- Pesan yang dikeluarkan untuk kotak pengisian

- Sinyal tanda selesai

- **Tahap 9 Prototipe**

Pada beberapa titik sepanjang porses mendesain, memiliki pilihan untuk sepenuhnya mengimplementasikan sistem Basis Data atau membangun *prototype*. Tujuan pada tahap ini adalah membangun sebuah model kerja untuk sistem Basis Data. *Prototype* adalah model kerja yang biasanya tidak memiliki semua fitur yang biasanya diperlukan atau menyediakan semua fungsi dari sistem akhir. Dengan cara inim dapat memperjelas kebutuhan *user* dan pengembang sistem khususnya mengevaluasi kelayakan dari desain sistem

- **Tahap 10 Implementasi**

Tujuan pada tahap ini adalah untuk merealissasikan sistem Basis Data dan desain aplikasi. Pada tahap penyelesaian desain, tahap selanjutnya adalah untuk mengimplementasikan sistem Basis Data dan aplikasi. Mengimplementasikan *Database* dapat menggunakan *Data Definition Language* (DDL) dan DBMS yang dipilih atau *Graphical User Interface* (GUI). Program aplikasi yang akan diimplementasikan menggunakan bahas pemrograman generasi ketiga atau ke empat seperti *Visual Basic* (VB), *VB.netm Phyton,Delphi, C, C++, Java ,COBOL, Fortran, Ada, atau Pascal.*

- **Tahap 11 Data konversi dan Pemuatan (*Data Conversion and Loading*)**

Tujuan pada tahap ini adalah men-transfer data yang ada ke dalam *Database* baru dan menkonversi setiap aplikasi yang ada untuk dijalankan pada *Database* yang baru. Tahap ini dibutuhkan hanya ketika sistem *Database* yang baru akan menggantikan sistem yang lama. Saat

ini, adalah hal yang biasa bagi DBMS untuk memiliki utilitas yang memuat file yang ada ke dalam *Database* yang baru. *Utilitas* biasanya membutuhkan spesifikasi dari sumber file dan taget *Database* dan secara otomatis menkonversi data ke format yang dibutuhkan oleh *Database* yang baru.

#### • **Tahap 12 Pengujian**

Tujuan pada tahap ini adalah untuk menjalankan *Database* dengan maksud menemukan kesalahan. Sebelum dijalankan, sistem basis data yang baru harus melewati proses pengujian. Pengujian dicapai menggunakan strategi pengujian yang direncanakan dan data realistik sehingga proses pengujian tidak boleh memiliki pandangan bahwa kesalahan tidak ada. Ada beberapa kriteria yang digunakan untuk tahap pengujian yakni :

1. *Learnability* : berapa lama yang dibutuhkan pengguna baru untuk menjadi produktif dengan sistem.
2. *Performance* : seberapa baik respon sistem terhadap pengguna.
3. *Robustness* : seberapa toleran sistem terhadap kesalahan pengguna.
4. *Recoverability* : seberapa baik sistem dapat mengaktifkan fitur recovery atas kesalahan pengguna
5. *Adaptability* : seberapa dekat sistem dengan model satu pekerjaan setelah pengujian selesai, maka sistem Basis Data siap untuk ‘ditandatangani’ dan diserahkan kepada pengguna.

#### • **Tahap 13 Pemeliharaan Operasional**

Tujuan pada tahap ini adalah untuk memantau dan memelihara sistem *Database*. Pada tahap sebelumnya, sistem Basis Data sudah diimplementasikan dan diuji. Tahap selanjutnya adalah tahap pemeliharaan yang melibatkan kegiatan-kegiatan seperti :

1. Memantau kinerja dari sistem
2. Pemeliharaan dan peningkatan kinerja sistem Basis Data (jika diperlukan)

Setelah sistem basis data sepenuhnya berfungsi untuk kegiatan operasional, pengawasan yang ketat terjadi untuk memastikan kinerja yang tetap dalam tingkat yang dapat diterima. DBMS menyediakan berbagai utilitas untuk membantu Database administrator termasuk utilitas untuk memuat data ke dalam Database dan untuk memonitor sistem. Utilitas tersebut juga memungkinkan sistem monitoring untuk memberikan informasi tentang penggunaan *Database*, melihat efisiensi (termasuk melihat *deadlock* yang terjadi) dan strategi eksekusi *query*. (Connolly & Begg, 2010:335)

#### **5.1.4 *Database Language***

Database Language adalah sebuah data sublanguage terdiri atas dua bagian yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). DDL digunakan untuk menentukan skema Database dan DML digunakan untuk membaca dan mengupdate *Database*. Keduanya disebut data sub language karena kedua data tersebut tidak membangun semua kebutuhan pemrograman komputer seperti pernyataan kondisi dan *iterative* yang digunakan pada beberapa bahasa pemrograman tingkat tinggi lainnya. (Connolly & Begg, 2010:91)

- *Data Definition Language*

*Data Definition Language* (DDL) adalah suatu bahasa yang memperoleh DBA atau pengguna untuk mendeskripsikan dan memberi nama entitas, atribut dan relationship yang diperlukan untuk aplikasi. DDL berfungsi untuk mengubah suatu data menjadi suatu data yang bermanfaat bagi pengguna. (Connolly & Begg, 2010:92)

- *Data Manipulation Language*

*Data Manipulation Language* (DML) adalah suatu bahasa yang memberikan sekumpulan operasi untuk mendukung operasi dasar dari manipulasi data yang ada dalam *Database*. Operasi manipulasi data biasanya termasuk dalam data berikut :

- Penyisipan data bari ke dalam *Database* (*insertion*)
- Mengubah atau memodifikasi data yang disimpan dalam *Database* (*modify*)
- Pemanggilan data yang ada dalam *Database* (*retrieve*)
- Menghapus data dari *Database* (*delete*)

(Connolly & Begg, 2010:92)

*Data Manipulation Language (DML)* terbagi atas dua tipe yang berbeda, yakni *Prosedural DML* dan *Non Prosedural DML*. Berikut adalah penjelasan mengenai *Prosedural DML* dan *Non Prosedural DML* :

a. *Prosedural DML*

*Prosedural DML* adalah suatu bahasa yang memungkinkan pengguna untuk memberikan instruksi ke sistem tentang data apa saja yang dibutuhkan serta bagaimana cara memanggilnya (*retrieve*).

b. *Non Prosedural DML*

*Non Prosedural DML* adalah suatu bahasa yang memungkinkan pengguna untuk menyatakan apakah suatau data itu dibutuhkan daripada bagaimana data tersebut diambil.

(Connolly & Begg, 2010:93)

### **5.1.5 Struktur Data Relasional**

Struktur data relasional dibagi menjadi beberapa bagian, yakni :

1. Relasi : merupakan sebuah tabel dengan baris dan kolom. Digunakan untuk

menyimpan informasi tentang objek yang digambarkan dalam *Database*.

2. Atribut : merupakan nama kolom dari relasi. Atribbut dapat ditampilkan dalam berbagai perintah dan dalam relasi yang sama sehingga menyampaikan arti yang sama.

3. *Domain*: merupakan sekelompok nilai yang diijinkan bagi satu atau lebih atribut. Setiap atribut dalam relasi didefinisikan pada sebuah *domain*. *Domain* dapat berbeda bagi setiap atribut, atau dua atau lebih atribut dapat didefinisikan pada domain yang sama. Konsep domain sangat penting karena memungkinkan pengguna menjelaskan arti dan sumber nilai yang ada pada atribut.

4. Tuple: merupakan baris dari sebuah relasi. Tuple dapat disebut intention jika struktur relasi, *domain* serta batasan-batasan yang lainnya pada nilai yang mungkin bersifat tetap,namun sebaliknya jika relasi berubah setiap waktu ini disebut *extension*.

5. *Degree* : merupakan jumlah atribut yang terdapat dalam relasi. Jika relasi mempunyai satu atribut akan mempunyai derajat satu yang disebut relasi *unary* / satu tup

6. *Cardinality* : merupakan jumlah tuple yang terdapat dalam relasi. Merupakan *property* dari *extension* relasi dan ditentukan dari *instance* tertentu.

7. *Database relasional* : merupakan kumpulan dari relasi yang ternormalisasi dengan nama relasi yang berbeda.

(Connolly & Begg, 2010:144 - 146)

### 5.1.6 *Relational Key*

*Relational keys* dibagi menjadi beberapa jenis, yakni:

1. *Super key* : merupakan sebuah atribut atau sekelompok atribut yang mengidentifikasi secara unik tuple dalam relasi. *Superkey* yang mudah

identifikasi adalah yang hanya berisi jumlah minimum atribut yang diperlukan.

2. *Candidate key* : merupakan superkey dalam relasi, *Candidate key* (K), bagi sebuah relasi (R) mempunyai dua sifat , yaitu :

a. Keunikan : dalam setiap tuple dari R, nilai dari K secara unik mengidentifikasi tuple tersebut.

b. *Irreducibility* : tidak ada subset yang sesuai dari K yang mempunyai keunikan sifat. Ketika sebuah key terdiri dari satu atribut ,ini disebut sebagai *composite key*.

3. *Primary key* : merupakan candidate key yang terpilih untuk identifikasi tuple secara unik dalam satu relasi. Sementara candidate key yang tak terpilih sebagai primary key disebut alternate key.

4. *Foreign key* : merupakan sebuah atribut atau sekelompok atribut dalam relasi yang dibandingkan dengan *candidate key* pda beberapa relasi.

(Connolly & Begg, 2010:150)

### 5.1.7 *Integrity Constraint*

*Relational Integrity* terbagi atas beberapa tipe, yakni:

- *Null*

Merupakan gambaran sebuah nilai bagi sebuah atribut yang tidak diketahui atau tidak digunakan bagi tuple tersebut. Null tidak sama dengan nilai numerik nol atau spasi, tapi nul menggambarkan ketidak adaan nilai.

- *Integritas entitas*

Pada relasi dasar, tidak ada atribut *primary key* yang bernilai null. Bedasarkan definisi diatas, primary key minimal berperan sebagai

identifier yang digunakan untuk mengidentifikasi tuple secara unik . ini berarti tidak ada subset dari *primary key* yang cukup untuk menyediakan perngidentifikasi tuple ang unik.

- *Integritas Referensi*

Jika terdapat *foreign key* dalam relasi, maka nilai *foreign key* tersebut akan dibandingkan dengan nilai *candidate key* dari beberapa tuple pada relasi itu sendiri atau nilai *foreign key* harus null semua.

(Connolly & Begg, 2010:153 - 155)

### 5.1.8 Normalisasi

Normalisasi adalah suatu teknik untuk memproduksi satu set hubungan dnegan kebutuhan yang diinginkan, memberi kebutuhan data dari suatu perusahaan. Tujuan lain dari normalisasi adalah mencegah adanya redudansi atau pengulangan data yang nantinya akan menghemat tempat pada penyimpanan. Proses dalam normalisasi terbagi menjadi beberapa tahap, yakni :

- *Unnormalized (UNF)*

Suatu table dikatakan sebagai bentuk yang *unnormalized* bila di dalamnya terdapat kelompok berulang atau yang biasa dikenal *repeating group*.

- *First Normal Form (1NF)*

Untuk mengubah bentuk Unnormalized Form menjadi 1NF yang harus dilakukan adalah mengidentifikasi dan menghilangkan kelompok berulang agar setiap pertemuan antara baris dan kolom berisi satu dan hanya satu nilai.

Langkah-langkah membuat normal dari bentuk *unnorma*l :

1. Menentukan 1 atau lebih atribut sebagai atribut kunci dari *table unnormal*.

2. Mengidentifikasi *repeating group* dari suatu *table unnormal* yang mengulang atribut kunci di atas.

3. Menghilangkan *repeating group*, dapat dilakukan dengan 2 cara :

mengisi kolom dari baris yang kosong dengan data-data yang sesuai atau menempatkan data yang berulang beserta *key*-nya dalam tabel baru.

- *Second Normal Form (2NF)*

Pada tahap normalisasi 2NF dihilangkan setiap *Partial Dependence* yang ada pada bentuk 1NF. Yang dimaksud dengan *Partial Dependence* adalah atribut *non primary key* yang merupakan sebagian fungsi dan *primary key*, atau dapat dijelaskan demikian apabila terdapat atribut-atribut dalam suatu relasi yang memiliki ketergantungan fungsional misalnya A (Kd\_dosen\_Nama) dan atribut B (KdMtk), dikatakan partial dependence apabila ada sebagian atribut dari A dihiangkan namun ketergantungan masih ada. Langkah-langkah membuat 2NF dari 1 NF :

1. Mengidentifikasi *primary key* dari bentuk 1NF

2. Mengidentifikasi *functional dependency* pada bentuk 1 NF

3. Bila terdapat *partial dependency* dalam *primary key* maka tempatkan *primary key* tersebut dalam suatu tabel baru beserta *field* lain yang berkaitan dengannya.

- *Third Normal Form (3NF)*

Pengujian terhadap bentuk normal ketiga dilakukan dengan cara melihat apakah terdapat atribut bukan *key* tergantung fungsionalnya terhadap atribut dengan bukan *key* yang lain

(disebut ketergantungan *transitive*). Pada tahap normalisasi 3 NF dihilangkan setiap *Transitive Dependence* yang terdapat pada bentuk 2 NF. Kondisi *transitive dependence* dapat diterangkan sebagai berikut : misalkan terdapat atribut A, B dan C yang mempunyai relasi AB , dan B C, C adalah *transitive dependence* terhadap A melalui B. Ketergantungan *transitif* (*transitive dependency*) terjadi ketika ada atribut yang bukan merupakan *primary key*, yang memiliki ketergantungan pada atribut lain yang juga bukan merupakan *primary key*. Langkah-langkah membuat 3NF dari 2NF :

1. Mengidentifikasi *primary key* pada bentuk 2NF
2. Mengidentifikasi *functional dependency* dalam tabel tersebut
3. Bila terdapat *transitive dependency* pada *primary key* maka tempatkan *primary key* beserta *field* lain yang berkaitan pada tabel baru. Proses normalisasi secara umum dilakukan sampai tahap 3NF karena sudah tidak terdapat data pengulangan dan *anomaly* yang ada sudah sangat sedikit.

(Connolly & Begg, 2010:416-461)

### 5.1.8 Data Flow Diagram

*Data Flow Diagram* (DFD), adalah sebuah alat yang menggambarkan aliran data melalui sistem dan kerja atau pengolahan yang dilakukan oleh sistem tersebut (Whitten, Bentley, & Dittman, 2004:344). Beberapa simbol yang digunakan dalam DFD antara lain :

1. *Process* adalah kerja yang dilakukan pada atau sebagai *respons* terhadap aliran data masuk atau kondisi.
2. *Data Flow* menunjukkan input data ke proses atau output data (atau informasi)

dari proses. *Data flow* juga digunakan untuk menunjukan pembuatan, pembacaan, penghapusan atau pembaruan data dalam file atau *Database*.

3. *External Agent* mendefinisikan orang, unit organisasi, sistem, atau organisasi luar yang berinteraksi dengan sistem.
4. *Data Store* adalah penyimpanan data yang ditujukan untuk penggunaan selanjutnya. Sinonimnya adalah file dan *Database*.

## **5.2 Teori Pendukung**

### **5.2.1 Operasional**

Operasional adalah sebuah fungsi atau sistem yang dapat mengubah *input* ke dalam *output* atau keluaran yang lebih besar. Operasional meliputi perancangan, pengoprasioan dan meningkatkan sistem menjadi lebih produktif (Russel & Taylor, 2009: 2).

### **5.2.2 Pengertian Logistik Pihak Ketiga**

Sebuah perusahaan 3PL adalah perusahaan firma pribadi yang menyediakan layanan logistik dibawah kontrak yang utamanya manufaktur, vendor, atau pengguna yang memiliki produk atau jasa. Ini disebut pihak ketiga karena penyedia layanan logistik tidak mempunyai produk sendiri namun berpartisipasi pada *supply-chain* diantara pembuatan manufaktur dan pengguna yang menerima produknya. Perusahaan 3PL mempunyai fungsi logistik dari pelanggan mereka secara lengkap atau sebagian. Biasanya, perusahaan 3PL adalah perusahaan pengangkutan, perusahaan pergudangan atau agen *forwarding*. Sekarang ini perusahaan 3PL menambahkan banyak variasi dengan menawarkan berbagai macam jasa dan menjamin berbagai kegiatan jasa tersebut. (Aguezzoul, 2007 : 1-7)

### **5.2.3 Pengertian Manajemen Material**

Manajemen material dapat diartikan sebagai *konfederasi* aktivitas material yang dibatasi oleh ide umum atau gagasan pendekatan manajemen terpadu untuk merencanakan aliran dan distribusi bahan produksi siap pakai. Manajemen material terhubung dengan menjaga pasokan yang memadai dari sesuatu seperti bahan baku, komponen, perlengkapan operasional untuk memenuhi kebutuhan. Materi merupakan elemen tunggal terbesar dari biaya. Umumnya lebih dari setengah biaya dari setiap sistem produksi karena material saja. Oleh karena itu mengurangi biaya bahan sangat penting bagi peningkatan produksi. Fungsi dari materi manajemen adalah memastikan pasokan bahan tiba pada waktu, tempat, kuantitas yang tepat dan berasal dari sumber yang tepat dengan harga yang tepat (Singh, 2008:1-2).

## **BAB 6**

---

### **TEORI MYSQL**

---

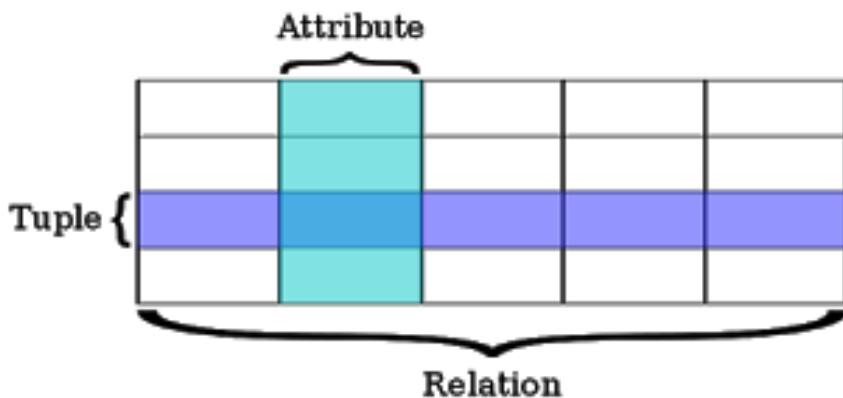
#### **6.1 Teori Dasar Database MySQL**

MySQL adalah salah satu RDBMS (*Relational Database Management System*). Maksud RDBMS adalah aplikasi database yang menggunakan prinsip relasional / saling terhubung satu sama lainnya. Alasan menggunakan *database* MySQL dari pada RDBMS lainnya adalah :

- *Speed*
- *Reliability*
- *Skalability*
- *User friendly*
- *Portability and Standard Compliance*
- *Multiuser Support*
- *Internationalization*
- *Wide Application Support*
- *Open Source Code*

Dalam relasional database model, sebuah database merupakan kumpulan relasi yang saling terhubung satu sama lainnya. Relasi adalah istilah dalam

*relational database*, tapi kita lebih familiar jika menyebutnya sebagai tabel. Selayaknya tabel yang memiliki kolom dan baris, dalam *relational database*, kolom (*column*) disebut attribute, sedangkan baris (*row*) disebut tuple. Hal ini hanya sekedar penamaan, dan agar lebih gampang, kita hanya akan menggunakan istilah tabel, kolom dan baris dalam tutorial ini, namun jika anda menemui istilah relation, attribut dan tuple, itu hanya penamaan lain dari tabel, kolom, dan baris.



Gambar 6.1 Istilah Tabel

### 6.1.1 Jenis-Jenis *Key Field* Pada MySQL

#### 1. *Candidat Key (Kunci Kandidat)*

Database dalam relational database dapat diserhanakan sebagai sekumpulan tabel yang saling terhubung. Setiap baris dari dalam tabel setidaknya harus memiliki sebuah kolom yang unik. Unik disini maksudnya tidak boleh sama. Candidate Key adalah satu atau beberapa kolom dalam tabel yang bisa mengidentifikasi tiap baris dari tabel tersebut. Beberapa karakteristik *Candidate key* : unik (tidak boleh berulang), tidak boleh memiliki nilai null (kosong), nilai dari candidate key akan sangat jarang berubah. Contohnya tabel data\_mahasiswa, kolom NIM (Nomor Induk Mahasiswa) akan menjadi

kandidat yang bagus, karena tidak mungkin ada 2 mahasiswa yang memiliki NIM yang sama. Contoh lain juga seperti Nomor KTP juga merupakan candidate key yang bagus, setidaknya setiap orang akan memiliki Nomor KTP yang berbeda-beda. Namun dalam beberapa kasus, nomor KTP tidak selalu ada, karena bisa saja seseorang belum memiliki KTP karena sesuatu dan lain hal.

## **2. Primary Key (*Kunci Utama*)**

Dalam sebuah tabel, akan terdapat beberapa candidate key, namun hanya ada 1 *Primary key* (*kunci utama*). Primary key adalah salah satu *candidate key* yang kita nobatkan sebagai kolom unik untuk identifikasi baris dalam tabel. Kolom ini tidak boleh berulang, dan tidak boleh kosong (*null*). Dari tabel data\_mahasiswa, NIM dapat kita tetapkan sebagai *primary key*.

## **3. Foreign Key (*Kunci Tamu*)**

Adalah kolom atau *field* pada suatu tabel yang berfungsi sebagai kunci tamu dari tabel lain. *Foreign Key* sangat berguna bila kita bekerja dengan banyak tabel yang saling berelasi satu sama lain. Dalam kata lain Kunci Penghubung merupakan sarana untuk menghubung antara dua tabel yang berbeda, tetapi masih punya kaitan yang sama.

### **6.1.2 Jenis-Jenis Tipe Data Pada MySQL**

#### **1. Tipe Data Numerik**

Tipe data numerik yaitu tipe data yang digunakan untuk menyimpan data numerik (angka).

No	Nama	Fungsi	Jangkauan	Ukuran
1	TINYINT	Menyimpan data bilangan bulat positif <u>dan</u> negatif.	-128 s/d 127	1 byte (8 bit).
2	SMALLINT	menyimpan data bilangan bulat positif <u>dan</u> negatif.	-32.768 s/d 32.767	2 byte (16 bit).
3	MEDIUMINT	menyimpan data bilangan bulat positif <u>dan</u> negatif.	-8.388.608 s/d 8.388.607	Ukuran : 3 byte (24 bit).
4	INT	menyimpan data bilangan bulat positif <u>dan</u> negatif.	-2.147.483.648 s/d 2.147.483.647	4 byte (32 bit).
5	BIGINT	menyimpan data bilangan bulat positif dan negatif.	$\pm 9,22 \times 10^{18}$	8 byte (64 bit).
6	FLOAT	menyimpan data bilangan pecahan positif dan negatif presisi tunggal	-3.402823466E+38 s/d -1.175494351E-38, 0, dan 1.175494351E-38 s/d 3.402823466E+38.	4 byte (32 bit)
7	DOUBLE	menyimpan data bilangan pecahan positif dan negatif presisi ganda.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit)
8	REAL	menyimpan data bilangan pecahan positif dan negatif presisi ganda.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).
9	DECIMAL	menyimpan data bilangan pecahan positif dan negatif.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).
10	NUMERIC	menyimpan data bilangan pecahan positif dan negatif.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).

Gambar 6.2 Tipe Data Numerik

## 2. Tipe Data *Date* dan *Time*

Tipe data *date* dan *time* yaitu tipe data yang digunakan untuk menyimpan data tanggal dan waktu

No	Nama	Fungsi	Jangkauan	Ukuran
1	DATE	menyimpan data tanggal	1000-01-01 s/d 9999-12-31 (YYYY-MM-DD)	3 byte.
2	TIME	menyimpan data waktu	-838:59:59 s/d +838:59:59 (HH:MM:SS)	3 byte
3	DATETIME	menyimpan data tanggal dan waktu.	'1000-01-01 00:00:00' s/d '9999-12-31 23:59:59'	8 byte
4	YEAR	menyimpan data tahun dari tanggal	1900 s/d 2155	1 byte

Gambar 6.3 Tipe data *date* dan *time*

### **3. Tipe Data String (Text)**

*Tipe data string* yaitu tipe data yang digunakan untuk menyimpan data *string* (text).

No	Nama	Fungsi	Jangkauan
1	CHAR	menyimpan data string ukuran tetap.	0 s/d 255 karakter
2	VARCHAR	menyimpan data string ukuran dinamis.	0 s/d 255 karakter (versi 4.1), 0 s/d 65.535
3	TINYTEXT	menyimpan data text.	0 s/d 255 karakter (versi 4.1), 0 s/d 65.535
4	TEXT	menyimpan data text.	0 s/d 65.535
5	MEDIUMTEXT	menyimpan data text	0 s/d 224 - 1 karakter
6	LONGTEXT	menyimpan data text.	0 s/d 232 - 1 karakter

6.4 Gambar *Tipe data string*

### **4. Tipe Data BLOB (Biner)**

*Tipe data blob* digunakan untuk menyimpan data biner.

No	Nama	Fungsi	Jangkauan
1	BIT	Menyimpan data biner.	64 digit biner
2	TINYBLOB	menyimpan data biner/ Gambar ukuran kecil	255 byte
3	BLOB	Menyimpan data biner/ Gambar	4
4	MEDIUMBLOB	Menyimpan data biner/ Gambar kurang sedang	224-1 byte
5	LONGBLOB	Menyimpan data biner/ Gambar ukuran besar	232-1 byte

6.5 Gambar *Tipe data blob*

### **5. Tipe Data yang Lain**

Selain tipe data di atas, MySQL juga menyediakan tipe data yang lain, diantaranya adalah :

No	Nama	Fungsi	Jangkauan
1	ENUM	enumerasi (kumpulan data).	sampai dengan 65535 string.
2	SET	combination (himpunan data).	sampai dengan 255 string anggota

6.6 Gambar Tipe Data yang Lain

### **6.1.3 DDL - Data Definition Language**

Seperti definisi yang telah dijelaskan di atas, DDL adalah kumpulan perintah *SQL* yang digunakan untuk membuat (*create*), mengubah (*alter*) dan menghapus (*drop*) struktur dan definisi tipe data dari objek-objek *database*. Objek-objek database pada yang dimaksud - pada *MySQL* - adalah sebagai berikut :

- *Database*
- *Table*
- *View*
- *Index*
- *Procedure (Stored Procedure)*
- *Function*
- *Trigger*

### **6.1.4 DML - Data Manipulation Language**

DML sendiri adalah kumpulan perintah *SQL* yang berhubungan dengan pekerjaan mengolah data di dalam table - dan tidak terkait dengan perubahan struktur dan definisi tipe data dari objek database seperti *table*, *column*, dan sebagainya.

### **6.1.5 Sejarah MySQL**

*MySQL* adalah pengembangan lanjutan dari proyek *UNIREG* yang dikerjakan oleh Michael Monty Widenius dan TcX (perusahaan perangkat lunak asal Swedia). Sayangnya, *UNIREG* belum terlalu *kompatibel* dengan

database dinamis yang dipakai di *website*. TcX kemudian mencari alternatif lain dan menemukan perangkat lunak yang dikembangkan oleh David Hughes, yaitu *miniSQL* atau *mSQL*. Namun, ditemukan masalah lagi karena *mSQL* tidak mendukung indexing sehingga belum sesuai dengan kebutuhan TcX.

Pada akhirnya muncul kerjasama antara pengembang *UNIREG* (Michael Monty Widenius), *mSQL* (David Hughes), dan TcX. Kerjasama ini bertujuan untuk mengembangkan sistem database yang baru, dan pada 1995 dirilislah *MySQL* seperti yang dikenal saat ini. Saat ini pengembangan *MySQL* berada di bawah *Oracle*.

### **6.1.6 Kelebihan dan Kekurangan MySQL**

Setelah mempelajari pengertian *MySQL* dan sejarahnya, ada baiknya jika Anda juga mengetahui kelebihan dan kekurangannya juga.

Meskipun menjadi database yang cukup populer, *MySQL* tentu mempunyai beberapa kelebihan dan kekurangan dibandingkan dengan *database* server lainnya. Salah satu kekurangan *MySQL* adalah performanya turun di saat beberapa database manajemen sistem mampu bekerja baik pada pengelolaan database yang besar.

Adapun kelebihan dan kekurangan *MySQL* lain, di antaranya:

#### **Kelebihan MySQL**

*MySQL* mempunyai beberapa kelebihan yang bisa Anda manfaatkan untuk mengembangkan perangkat lunak yang andal seperti:

## **1. Mendukung Integrasi Dengan Bahasa Pemrograman Lain.**

Website atau perangkat lunak terkadang dikembangkan dengan menggunakan berbagai macam bahasa pemrograman, jadi Anda tidak perlu khawatir jika menggunakan *MySQL*. Maka dari itu, *MySQL* bisa membantu Anda untuk mengembangkan perangkat lunak yang lebih efektif dan tentu saja lebih mudah dengan integrasi antara bahasa pemrograman.

## **2. Tidak Membutuhkan RAM Besar.**

*MySQL* dapat dipasang pada server dengan spesifikasi kecil. Jadi tidak perlu khawatir jika Anda hanya mempunyai *server* dengan kapasitas 1 GB karena Anda masih bisa menggunakan *MySQL* sebagai *database* Anda.

## **3. Mendukung *Multi User*.**

*MySQL* dapat dipakai oleh beberapa user dalam waktu bersamaan tanpa membuatnya crash atau berhenti bekerja. Ini dapat Anda manfaatkan ketika mengerjakan proyek yang sifatnya tim sehingga seluruh tim dapat bekerja dalam waktu bersamaan tanpa harus menunggu user lain selesai.

## **4. Bersifat *Open Source***

*MySQL* adalah sistem manajemen *database* gratis. Meskipun gratis, bukan berarti database ini mempunyai kinerja buruk. Apalagi lisensi gratis yang dipakai adalah GPL di bawah pengelolaan *Oracle* sehingga kualitasnya termasuk baik. Selain itu, Anda juga tidak perlu khawatir jika terjadi masalah karena banyak komunitas dan dokumentasi yang membahas soal *MySQL*.

## **5. Struktur Tabel yang *Fleksibel*.**

MySQL mempunyai struktur tabel yang mudah dipakai dan fleksibel. Contohnya saat MySQL memproses *ALTER TABLE* dan lain sebagainya. Jika dibandingkan dengan *database* lain seperti *Oracle* dan *PostgreSQL*, MySQL tergolong lebih mudah.

## **6. Tipe Data yang *Bervariasi*.**

Kelebihan lain dari MySQL adalah mendukung berbagai macam data yang bisa Anda gunakan di MySQL. Contohnya *float*, *integer*, *date*, *char*, *text*, *timestamp*, *double*, dan lain sebagainya. Jadi manajemen *database* sistem ini sangat membantu Anda untuk mengembangkan perangkat lunak yang berguna untuk pengelolaan database di server.

## **7. Keamanan yang Terjamin.**

*Open source* bukan berarti MySQL menyediakan keamanan yang buruk. Malah sebaliknya, MySQL mempunyai fitur keamanan yang cukup apik. Ada beberapa lapisan keamanan yang diterapkan oleh MySQL, seperti level nama *host*, dan *subnetmask*. Selain itu MySQL juga dapat mengatur hak akses user dengan *enkripsi password* tingkat tiniggi.

## **Kekurangan MySQL**

Sayangnya, meskipun memiliki segudang kelebihan, masih ada beberapa kelemahan yang dimiliki oleh MySQL sehingga Anda perlu mempertimbangkannya juga sebelum memakainya.

## **1. Kurang Cocok untuk Aplikasi *Game* dan *Mobile***

Anda yang ingin mengembangkan aplikasi game atau perangkat mobile ada baiknya jika mempertimbangkan lagi jika ingin menggunakan *MySQL*. Kebanyakan pengembang *game* maupun aplikasi mobile tidak menggunakannya karena memang database manajemen sistem ini masih kurang bagus dipakai untuk sistem aplikasi tersebut.

## **2. Sulit Mengelola *Database* yang Besar**

Jika Anda ingin mengembangkan aplikasi atau sistem di perusahaan dengan database yang cukup besar, ada baiknya jika menggunakan *database* manajemen sistem selain *MySQL*. *MySQL* dikembangkan supaya ramah dengan perangkat yang mempunyai spesifikasi rendah, itulah mengapa *MySQL* tidak memiliki fitur yang lengkap seperti aplikasi lainnya

## **3. *Technical Support* yang Kurang Bagus**

Sifatnya yang open source terkadang membuat aplikasi tidak menyediakan *technical support* yang memadai. *Technical support MySQL* diklaim kurang bagus. Hal ini membuat pengguna kesulitan. Apalagi jika pengguna mengalami masalah yang berhubungan dengan pengoperasian perangkat lunak tersebut dan membutuhkan bantuan *technical support*.

Anda sudah belajar pengertian, sejarah, kekurangan, dan kelebihan mysql. Anda bisa melanjutkan membaca untuk belajar *MySQL* dari dasar.

## 6.1.7 Mengenal *SQL*

*SQL* merupakan bahasa pemrograman yang perlu Anda pahami karena dapat merelasikan antara beberapa tabel dengan database maupun antar database. Ada tiga bentuk *SQL* yang perlu Anda ketahui, yaitu *Data Definition Language* (DDL), *Data Manipulation Language* (DML), dan *Data Control Language* (DCL)

<b>CREATE</b>	Dipakai untuk membuat tabel dan database.
<b>DROP</b>	Dipakai untuk menghapus database dan tabel.
	Dipakai untuk mengubah struktur tabel yang sudah ada.
<b>ALTER</b>	Alter dapat mengganti field menggunakan perintah “Change”, menambahkan field menggunakan perintah “Add”, atau menghapus field menggunakan perintah “drop”, dan mengubah namanya menggunakan perintah “Rename”.

Gambar 6.7 Mengenal *SQL*

## 6.1.8 Belajar MySQL Lengkap

Mempelajari *MySQL* cukup mudah. Sesuai dengan pengertian *MySQL* yang sudah dibahas sebelumnya, bahwa sistem manajemen server ini menggunakan *SQL* sebagai bahasa pemrogramannya. Jadi akan lebih baik jika Anda harus memahami bagaimana *SQL* bekerja dan cara menggunakannya.

## 6.1.9 *Data Definition Language* (DDL)

DDL berguna pada saat Anda ingin mendefinisikan data di dalam database. Terdapat beberapa query yang dikelompokkan ke dalam DDL, yaitu:

Argumen DDL di atas perlu Anda pahami karena merupakan dasar penggunaan *SQL* di bagian awal pembuatan *database*. Contohnya saja jika belum menjalankan perintah “*CREATE*”, Anda belum bisa melanjutkan penggunaan argumen yang lainnya.

#### **6.1.10 Data Manipulation Language (DML)**

DML dapat Anda pakai setelah menjalankan perintah DDL. DML berfungsi untuk memanipulasi, mengubah, atau mengganti isi dari database (tabel) yang sudah ada.

Terdapat beberapa perintah DML yang perlu Anda ketahui, yaitu:

<b>INSERT</b>	Dipakai untuk memasukkan data ke dalam tabel pada database.
<b>UPDATE</b>	Dipakai untuk mengubah data yang ada di dalam tabel pada database.
<b>DELETE</b>	Dipakai untuk menghapus data di dalam tabel pada database.

Gambar 6.8 Mengenal DML

#### **6.1.11 Data Control Language (DCL)**

Jika Anda sudah mempunyai user dan ingin mengatur hak akses masing-masing user, Anda sebaiknya memahami berbagai macam jenis DCL dan cara penggunaannya. DCL berguna untuk memberikan hak akses *database*, mendefinisikan *space*, mengalokasikan *space*, dan melakukan audit penggunaan *database*.

Terdapat beberapa perintah DCL yang perlu Anda ketahui, yaitu:

<b>GRANT</b>	Dipakai untuk memberikan izin kepada user untuk mengakses database.
<b>REVOKE</b>	Dipakai untuk membatalkan izin user untuk mengakses database.
<b>COMMIT</b>	Dipakai untuk menetapkan penyimpanan pada database.
<b>ROLLBACK</b>	Dipakai untuk membatalkan penyimpanan pada database.

Gambar 6.8 Mengenal DCL