BAB 2

TINJAUAN PUSTAKA

2.1 Profil Perusahaan

Profil perusahaan adalah untuk mengetahui keadaan diperusahaan diantaranya adalah sejarah berdirinya perusahaan, stuktur organisaasi perusahaanan serta visi dan misi perusahaan.

2.1.1 Sejarah Singkat Perusahaan

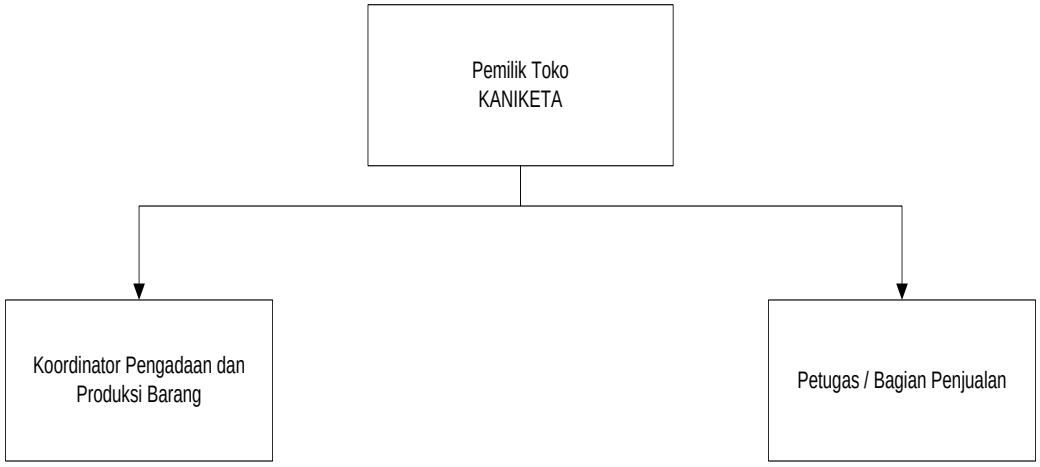
Toko Kaniketa *Clothing and Printing* adalah sebuah badan usaha yang bergerak di bidang produksi garmen dan percetakan. Toko Kaniketa *Clothing and Printing* berdiri sejak tahun 2003. Produk atau barang yang di jual disini terdiri dari banyak jenis. Untuk produksi garmen, toko Kaniketa menyediakan berbagai macam pembuatan dan pemesanan *t-shirt*, topi, training spak, kemeja, jaket almamater, tas, spanduk, bendera, celana, sweater,jaket dan sebagainya. Dan untuk produksi percetakan toko Kaniketa juga menyediakan berbagai jenis pembuatan dan pemesanan buku agenda, nota, faktur yang biasa kita peroleh sewaktu belanja di toko, kwitansi, kartu nama, undangan, kalender, *name tag* atau label, kop surat, amplop, plakat, plastik toko, *sticker*, *digital sticker*, poster, *ID card*, brosur, *leaflet*, dan sebagainya.Toko Kaniketa *Clothing and Printing* mulai membuka tempat sendiri di Bandung.

2.1.2 Struktur Organisasi Perusahaan

Struktur organisasi adalah suatu susunan dan hubungan antara tiap bagian serta posisi yang ada pada suatu organisasi atau perusahaan dalam menjalankan kegiatan operasional untuk mencapai tujuan. Struktur organisasi menggambarkan dengan jelas pemisahan kegiatan pekerjaan antara yang satu dengan yang lain dan bagaimana hubungan aktivitas dan fungsi dibatasi.

Struktur organisasi mempunyai arti penting karena struktur organisasi merupakan bentuk formal kegiatan dan hubungan antara berbagai subunit bagian-bagian didalam perusahaan, dengan mengetahui struktur organisasi dapat diperoleh suatu gambaran tentang pekerjaan dan hubungan-hubungan yang ada didalam perusahaan serta digunakan untuk merumuskan rencana kerja yang ideal sebagai pedoman untuk dapat mengetahui siapa bawahan dan atasannya dalam suatu organisasi perusahaan.

Adapun bagian-bagian pekerjaan yang ada di Toko Kaniketa *Clothing and Printing*:



Gambar 2.1 Struktur Organisasi Toko Kaniketa

2.1.3 Deskripsi Tugas

Deskripsi tugas digunakan untuk mengetahui tugas, wewenang dan tanggung jawab dari masing-masing bagian. Adapun deskripsi tugas yang ada di Toko Kaniketa *Clothing and Printing* akan di jelaskan apa halaman berikut :

1. Pemilik Toko / *Owner*
    a. Pemilik toko yaitu pemilik sekaligus atasan yang berkuasa penuh terhadap staff. Pemilik toko bertugas mengawasi serta ikut terlibat dalam berjalannya sistem prosedur operasional toko secara keseluruhan untuk menjaga konsisitensi kerja dalam mencapai tujuan perusahaan secara efektif dan efisien.

b. Melaksanakan audit yang meliputi keseluruhan aspek kegiatan manajemen keuangan operasional agar pengelolaan dapat dilaksanakan secara efektif dan efisien.

2. Bagian Koordinator Pengadaan dan Produksi Barang

   Bertugas untuk memeriksa barang pesanan konsumen dan stok barang yang akan dijual masih tersedia atau tidak, serta melakukan produksi barang pesanan dari tiap pelanggan yang telah mendapatkan persetujuan produksi dari pemilik toko.

3. Bagian Petugas / Bagian Penjualan

   Petugas / Bagian Penjualanbertugas untuk membuat laporan transaksi dan laporan barang baik laporan pengeluaran maupun laporan pemasukan. Dan memberikan hasil laporan nya kepada pemilik toko.

2.1.4 Visi dan Misi Perusahaan

Kaniketa *Clothing and Printing* memiliki visi dalam menjalankan usahanya yaitu menjadikan toko Kaniketa sebagai salah satu toko yang bertaraf nasional bahkan internasional dalam segi pengadaan serta produksi barang sehingga produk yang dihasilkan berkualitas, memuaskan dan semakin terpercaya.

Adapun misi dari Toko Kaniketa adalah :

a. Memberikan pelayanan yang terbaik bagi konsumen.

b. Mengembangkan usaha yang memiliki daya saing kuat.
c. Menjual Produk/barang yang bermutu tinggi, sesuai dengan yang di inginkan konsumen.

2.2Landasan Teori

Landasan teori merupakan kumpulan teori-teori yang menjadi dasar dari pembangunan aplikasi *e-commerce* ini yang dikutip dari berbagai buku.

2.2.1 Pengenalan *E-Commerce*

E-Commerce (*electronic commerce*) merupakan satu teknologi yang berkembang pesat seiring dengan kehadiran Internet dalam kehidupan kita. Banyak orang mendapatkan manfaat kemudahan berbisnis melalui Internet.

E-commerce merupakan bagian dari e-bisnis. Dimana *e-commerce* mengacu pada Internet untuk belanja *online* dan jangkauannya sempit. Sebagai sub perangkat dari e-bisnis. Cara pembayaran pada *e-commerce* dilakukan melalui transfer uang secara digital melalui *account paypal* atau kartu kredit. Sedangkan e-bisnis mengacu pada Internet tapi jangkauannya lebih luas. Hubungan bisnis terjadi ketika perusahaan atau individu berkomunikasi dengan klien atau nasabah melalui *email* akan tetapi pemasaran atau penjualan dilakukan melalui Internet. Cara ini dapat memberikan keuntungan fleksibilitas dan efisiensi. Pembayaran dilakukan melalui pembayaran

digital dengan menggunakan *e-gold* yang sudah diakui di seluruh dunia dalam melakukan transaksi *online*.

E-commerce sendiri berasal dari layanan EDI (*Electronic Data Interchange*).Layanan EDI ini telah berkembang sedemikian pesatnya di negara-negara yang mempunyai jaringan komputer dan telepon. Jika sebelumnya kita sering menggunakan media elektronik seperti telepon, *fax*, hingga *hand phone* untuk melakukan perniagaan atau perdagangan, sekarang ini kita dapat menggunakan Internet untuk melakukan perniagaan.

2.2.2. Definisi *e-commerce*

Definisi *e-commerce* menurut kamus Wikipedia adalah Perdagangan elektonik atau e-dagang (bahasa Inggris: *electronic commerce*) yaitu penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet atau televisi, www, dan jaringan komputer lainnya [2]. E-dagang dapat melibatkan transfer dan elektronik, pertukaran data elektronik, sistem manajemen inventaris otomatis, dan sistem pengumpulan data otomatis.

Menurut Mariza Arfina dan Robert Marpaung, *e-commerce* atau yang lebih dikenal *e-com* dapat diartikan sebagai cara berbelanja atau berdagang secara *online* atau *direct selling* yang memanfaatkan fasilitas Internet dimana terdapat *website* yang dapat menyediakan layanan "*get and deliver*" [2].

Definisi *e-commerce* yang sudah distandarkan dan disepakati bersama adalah "*E-commerce is a dynamic set of technologies, applications, and business process that link enterprise, consumers, and communities through electronic transactions and the electronic exchange of goods, services, and information*". *E-Commerce* merupakan satu set dinamis teknologi, aplikasi dan proses bisnis yang menghubungkan perusahaan, konsumen, dan komunitas tertentu melalui transaksi elektronik dan perdagangan barang, pelayanan, dan informasi yang dilakukan secara elektronik [2]. (David Baum)

Dengan demikian, dapat disimpulkan bahwa *e-commerce* merupakan proses pembelian dan penjualan jasa atau produk antara dua belah pihak melalui Internet (*Commerce net*) dan sejenis mekanisme bisnis elektronik dengan fokus pada transaksi bisnis berbasis individu dengan menggunkan internet sebagai media pertukaran barang atau jasa baik antar instansi atau individu dengan instansi (*Net-Ready*).

2.2.3. Manfaat *e-commerce*

Banyak manfaat dari penjualan secara *online* melalui media Internet, yaitu dengan adanya *e-commerce* suatu organisasi atau perusahaan dapat menjalankan bisnisnya untuk menjangkau seluruh Indonesia bahkan sampai ke luar negeri, beberapa manfaat *e-commerce* antara lain sebagai berikut :

1. Kemampuan grafis Internet mampu memperlihatkan produk apa adanya (natural) serta dapat membuat brosur berwarna dan menyebarkannya tanpa ongkos kirim.

2. Lebih aman membuka toko *online* dibanding membuka toko biasa.
3. Berjalan di dunia maya Internet tidak mengenal hari libur dan hari besar, semua transaksi bisa dilakukan kapan saja dan dimana saja.
4. Tanpa batas-batas wilayah dan waktu, sehingga memberikan jangkauan pemasaran yang luas dan tak terbatas oleh waktu.
5. *Revenue stream* (arus pendapatan) yang baru yang mungkin sulit atau tidak dapat diperoleh melalui cara konvensional.
6. Meningkatkan *market exposure* (pangsa pasar), dimana penggunaan *e-commerce* memungkinkan untuk meningkatkan pangsa pasar yang semula mempunyai pangsa pasar di dalam negeri saja, dengan adanya *e-commerce* maka pangsa pasar menjangkau luar negeri.
7. Menurunkan biaya operasi (*operating cost*). Penggunaan teknologi Internet memungkinkan kita untuk melakukan kegiatan perdagangan selama 24 jam sehari, 7 hari seminggu, akan tetapi tidak terpengaruh terhadap biaya yang dikeluarkan untuk biaya lembur karyawan atau pegawai, karena segala sesuatunya dikerjakan oleh komputer yang tidak membutuhkan operator untuk menjalankan proses perdagangan, cukup hanya dengan penggunaan *software* tertentu maka semua aktivitas dalam transaksi perdagangan dapat dilakukan.
8. Penghematan besar yang dimungkinkan melalui *e-mail*. Penghematan ini terjadi karena berkurangnya penggunaan kertas dalam segala proses transaksi, dimana segala sesuatunya didalam *e-commerce* menggunakan data digital sehingga tidak

membutuhkan kertas sebagai media yang pada akhirnya memberikan penghematan besar terhadap pengeluaran dalam proses transaksi.

2.2.4. Internet dan *e-commerce*

Dalam beberapa waktu terakhir ini, dengan begitu merebaknya media Intenet dimana-mana. Khususnya di Indonesia, dengan semakin menjamurnya warnet yang bertebaran hampir seperti wartel, ternyata telah membuat banyak perusahaan mulai mencoba menawarkan berbagai macam produk mereka dengan menggunakan media ini.

Kehadiran Internet yang walaupun masih merupakan industri baru yang dalam fase pertumbuhan, yang masih terus berubah serta penuh ketidakpastian, telah memperkokoh keyakinan akan pentingnya teknologi dalam pencapaian tujuan finansial perusahaan melalui modifikasi dan efisiensi proses bisnis, yaitu dengan memanfaatkan *E-Commerce*. Kemampuan Internet untuk menjangkau pelanggan baru dan penghematan biaya yang cukup signifikan untuk distribusi dan pelayanan pelanggan merupakan keuntungan yang bisa didapat perusahaan dengan memindahkan roda *commerce* ke media Internet.

Perkembangan teknologi Internet yang sangat cocok berubah menjadi strategi atau model bisnis yang cocok hari ini mungkin tidak relevan untuk diterapkan besok. Didalam lingkungan Internet yang kompetitif ini, di samping perencanaan yang baik serta harapan yang realistis terhadap teknologi baru, juga diperlukan kesiapan

infrastruktur yang cukup memadai dan visi yang komperhensif untuk bisa berjalan melewati ketidakpastian yang begitu cepat.

E-Commerce merupakan salah satu keunggulan dari Internet, hingga akhirnya di era sekarang ini (banyak disebut era digital) nampaknya tiada hari yang terlewatkan tanpa mendengar atau membaca kata *E-Commerce* di berbagai media informasi.

2.2.5. Sistem transaksi *e-commerce*

Untuk saat ini sistem transaksi di *e-commerce* bisa dibilang sudah lengkap dan terbilang aman untuk melakukan transaksi. Karena dengan adanya keamanan ini pelanggan dapat tertarik untuk melakukan transaksi secara *online* di Internet. Beberapa model pembayaran *online* di Internet untuk saat ini yang masih sering digunakan sebagai berikut :

1. Paypal

   Paypal merupakan salah satu alat pembayaran (*payment process*) teraman dan terbanyak digunakan di dunia Internet. Saat ini, orang Indonesia sudah dapat memindahkan uang yang berada di *account Paypal* langsung ke rekeningnya. Pengguna Internet dapat membeli barang di *ebay*, lisensi *software original*, keanggotaan situs, urusan bisnis, mengirim dan menerima donasi atau sumbangan dan mengirimkan uang ke pengguna *Paypal* lain di seluruh dunia. *Paypal* mengatasi kekurangan dalam pengiriman uang tradisional seperti cek atau *money order* yang prosesnya memakan waktu yang lama. *Paypal* tidak berbeda dengan

rekening bank. Hal ini pertama yang perlu Anda lakukan adalah membuat *account* tersebut dengan dana dari kartu kredit atau mentransfer dana dari *accountPaypal* orang lain ke *balance Paypal* Anda. Jika proses ini sudah diselesaikan, maka *account Paypal* sudah dapat digunakan untuk bertransaksi.

Pada bulan Februari 2008 *Paypal* sudah menerima 190 negara dan 16 mata uang. Pengguna kartu *Paypal* di Indonesia masih harus menggunakan hitungan US dollar karena rupiah masih belum ada di *Paypal*. *Ebay* adalah tempat jual-beli *online* terbesar di dunia. Anda dapat membeli dan mencari apapun, mulai dari barang elektronik, kendaraan hingga barang-barang langka dari seluruh pengguna *ebay* di dunia. Cara pembayaran utama di *ebay* adalah dengan *Paypal* karena lebih aman dan otomatis, meskipun penjual yang lain juga menerima pembayaran dengan cara lain.

2. Kartu Kredit

Kartu kredit dapat digunakan sebagai alat pembayaran dengan terlebih dahulu mendaftarkan diri Anda sebagai konsumen pada *web* tersebut. Selain kartu kredit sebagai alat pembayaran alternatif, Anda bisa menggunakan *e-cash,* yaitu sebuah *account* khusus untuk pembayaran melalui Internet. *Account* tersebut dibuka dengan kartu kredit yang dimiliki sebelumnya. Konsumen hanya perlu mengisi *account e-cash*-nya untuk digunakan.

3. Smartcard

Di Singapura, *smartcard* dikenal dengan istilah *cash card*. Pemakaian *smartcard* hampir sama dengan pemakaian kartu ATM yang biasa digunakan untuk berbelanja. Pada saat transaksi, uangnya di-*debet* langsung dari *account* di bank. Untuk pembayaran di Internet, konsumen harus memiliki *smart card reader*. Dalam penggunaanya, alat khusus ini disambungkan ke *port serial* di komputer. Ketika melakukan transaksi, kartu *smartcard* harus digesekan pada alat tersebut agar *chip* yang terdapat di kartu tersebut dapat dibaca oleh komputer. Penggunaan sistem pembayaran ini menggunakan *software* bernama *e-wallet*. Contoh *website* yang telah menerima *smartcard* sebagai alat pembayaran adalah http://www.discvault.com.

4. iCheck

iCheck adalah metode pembayaran menggunakan cek. Pembayaran ini membutuhkan nomor cek milik konsumen.

5. E-Gold

E-gold dapat diperoleh dengan membuka rekening gratis di www.E-gold.com. *E-gold* adalah pembayaran *virtual* yang cukup digemari oleh penjual *e-book* dan *software* di Internet. Dari segi keamanan, ada robot yang selalu mengganti PIN Anda dan mengirimnya ke *e-mail* saat Anda membutuhkannya. *E-gold* dapat dibeli atau dijual pada *web-web* penyedia jenis pembayaran ini, misalnya www.greatachiever.com.

2.3. Internet dan *World Wide Web*

E-Commerce bergantung pada keberadaan Internet sebagai media utama bertransaksi. Tanpa Internet, maka perdagangan secara elektronik ini tidak akan pernah terlaksana.

2.3.1. Pengertian Internet

Istilah Internet berasal dari bahasa latin, *inter,* yang berarti ”*antara*”. Secara kata per kata Internet berarti jaringan antara atau penghubung. Memang itulah fungsinya, Internet menghubungkan berbagai jaringan yang tidak saling bergantung pada satu sama lain dengan sedemikian rupa, sehingga mereka dapat berkomunikasi.

Internet adalah jaringan komputer yang saling terhubung ke seluruh dunia tanpa mengenal batas teritorial, hukum dan budaya. Secara fisik dianalogikan sebagai jaring laba-laba (*The Web*) yang menyelimuti bola dunia dan terdiri dari titik-titik (*node*) yang saling berhubungan [2].

Node bisa berupa komputer, jaringan lokal atau peralatan komunikasi, sedangkan garis penghubung antar simpul disebut tulang punggung (*back-bone*)yaitu media *terestrial* (kabel, serat optic, microwave, radio link) maupun satelit. *Node* terdiri dari pusat informasi dan database, peralatan komputer dan perangkat interkoneksi jaringan serta peralatan yang dipakai pengguna untuk mencari, menempatkan dan bertukar informasi di Internet.

Menurut Lani Sidharta (1996) walaupun secara fisik Internet adalah interkoneksi antar jaringan komputer namun secara umum Internet harus dipandang sebagai

sumber informasi. Isi Internet adalah informasi, dapat dibayangkan sebagai suatu database atau perpustakaan multimedia yang sangat besar dan lengkap. Bahkan Internet dipandang sebagai dunia dalam bentuk lain (maya) karena hampir seluruh aspek kehidupan di dunia nyata ada di Internet seperti bisnis, hiburan, olah raga, politik dan lain sebagainya.

Salah satu isu terpenting salah satu isu terpenting di Internet adalah penerapan *standar* komputerisasi terbuka (*open computing standart*). Karena *internetworking* dan Internet mengintegrasikan semua sistem, jenis dan tipe komputer yang ada di dunia, maka harus ada *standar* yang menjamin komputer dapat saling berbicara satu sama lain dalam bahasa yang sama. Menurut Drew Heywood (1966): *standar* bahasa komputer universal telah dikembangkan sejak 1969, terdiri dari serangkaian protocol komunikasi disebut *Transfer Control Protocol (TCP)* yang bertugas mengendalikan transmisi paket data, koreksi kesalahan dan kompresi data dan Internet Protocol (IP) yang bertugas sebagai pengenal (*identifier*) dan pengantar paket data ke alamat yang dituju.

Protokol TCP / IP menyatukan bahasa dan kode berbagai komputer di dunia, sehingga menjadi *standar* utama jaringan komputer. TCP / IP berkembang cepat dan kaya fasilitas karena bersifat terbuka dan bebas digunakan, oleh siapa pun. Oleh karena itu, hingga saat ini jaringan-jaringan tersebut saling terhubung di seluruh belahan dunia.

Seluruh jaringan yang tersebar di setiap penjuru dunia sudah terkondisikan dengan internet. Lalu lintas data Internet ke seluruh Amerika dan penjuru dunia saat ini disebarkan oleh jaringan pusat yang disebut VBNS (*Very High Speed Backbone Network Service*). Adapun seluruh biaya yang dikeluarkan di biayai oleh badal Ilmu Pengetahuan Nasional Amerika Serikat. Dari jaringan itu, komputer-komputer yang terhubung dengan internet akan berkomunikasi dan saling mengirimkan data sehingga internet dapat dinikmati oleh seluruh umat manusia untuk berkomunikasi melalui komuputer baik di rumah maupun di kantor.

Dengan demikian, internet dapat didefinisikan dengan hubungan berbagai jenis komputer dan jaringan di dunia, yang berbeda sistem operasi maupun aplikasinya di mana hubungan tersebut memanfaatkan kemajuan media komunikasi (telepon dan satelit) yang menggunakan protocol *standar* dalam berkomunikasi yaitu protocol TCP / IP. Internet berfungsi sebagai media komunikasi dan informasi modern.

2.3.2. Pengertian *World Wide Web*

World Wide Web adalah fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen *web* disebut *web page* dan *link* dalam *web* membuat *user* bisa pindah dari satu *page* ke *page* lain (*hyper text*), baik antar *page* yang disimpan dalam *server* yang sama atau *server* di seluruh dunia. *Pages* diakses dan dibaca melalui *web browser* seperti *netscape navigator* atau *internet explorer.*

World Wide Web sering disingkat WWW atau *web* saja, yaitu sebuah sistem dimana informasi dapat kita nikmati dalam bentuk teks, gambar, suara, dan lain-lain serta ikut dapat diakses oleh perangkat lunak yang disebut *browser*. Informasi di *web* pada umumnya ditulis dalam bentuk HTML. Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam bentuk format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World). WWW dijalankan dalam *server* yang disebut HTTPD.

2.3.3. Universal Resource Locator (URL)

URL adalah konsep nama file standar yang diperluas dengan jaringannya. Nama file ini tidak hanya menunjukan direktori dan nama filenya saja, tetapi juga nama mesinnya dalam jaringan. URL dapat disediakan (ada atau diakses) dengan berbagai metode dan bisa jadi bukan sekedar file. Karena URL dapat menunjukan queri-queri dokumen dapat disimpan dalam database, hasil dari suatu perintah finger atau achie atau apapun yang berkaitan dengan data hasil proses.

2.3.4. Domain Name Service (DNS)

DNS digunakan untuk mempermudah penggunaan internet, dengan memetakan IP *address* ke nama host. Agar data nama host dapat didistribusikan di banyak server, format data yang digunakan harus mencerminkan terdistribusinya data tersebut. Untuk itu, digunakan format tree dengan masing-masing nodenya disebut domain. Penulisan nama host secara lengkap disebut sebagai *Full Qualifield Domain Name (FQDN).*

DNS bekerja secara hirarki dan berbentuk seperti pohon (tree). Bagian atas adalah *Top Level Domain* (TLD) seperti COM, ORG, EDU, MIL dsb. Seperti pohon DNS mempunyai cabang-cabang yang dicari dari pangkal sampai ke ujung. Pada waktu kita mencari alamat misalnya linux.or.id pertama-tama DNS bertanya pada TLD server tentang DNS Server yang melayani domain .id misalnya dijawab ns1.id, setelah itu dia bertanya pada ns1.id tentang DNS Server yang bertanggung jawab atas .or.id misalnya ns.or.id kemudian dia bertanya pada ns.or.id tentang linux.or.id dan dijawab 64.29.24.175

Sedangkan untuk mengubah IP menjadi nama *host* melibatkan domain in-addr.arpa. Seperti domain lainnya domain in-addr.arpa pun bercabang-cabang. Yang penting diingat adalah alamat IP-nya ditulis dalam urutan terbalik di bawah in-addr.arpa. Misalnya untuk alamat IP 64.29.24.275 prosesnya seperti contoh linux.or.id: cari server untuk arpa, cari server untuk in-addr.arpa, cari server untuk 64.in-addr.arpa, cari server 29.64.in-addr.arpa, cari server untuk 24.29.64.in-addr.arpa. Dan cari informasi untuk 275.24.29.64.in-addr.arpa. Pembalikan urutan angkanya memang bisa membingungkan.

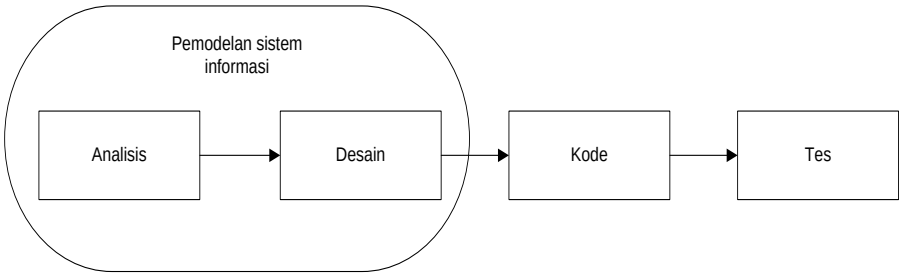
2.3.5. HTTP (*Hypertext Transfer Protocol*)

Server dan *browser web* berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini yaitu HTTP. HTTP bertugas menangani permintaan-permintaan *(request)* dan *browser* untuk mengambil dokumen-dokumen *web*. HTTP bisa dianggap sebagai sistem yang bermodel *client-server*.

2.4. SDLC *Model Sequential Linear*

System Development Life Cycle (SDLC) adalah suatu metodologi yang digunakan untuk mengembangkan, memelihara dan menggunakan perangkat Lunak (O'briend, 2001), metodologi penelitian ini mencakup sejumlah fase atau tahapan.

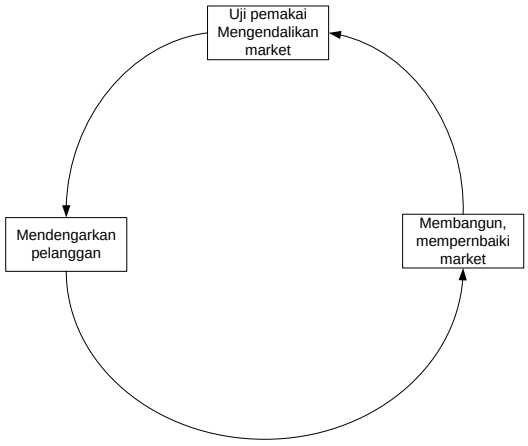
Linear Sequential Model (model sequensial linear) dapat dilihat pada gambar 2.1, ditemukan pertama kali oleh Royce. Model ini sering disebut model klasik atau *waterfall.* Model ini menyarankan pendekatan pengembangan secara sequen dan sistematik untuk pengembangan perangkat lunak [3]. Model ini merupakan model tertua. Model ini terdiri atas beberapa tahapan, yaitu rekayasa, dan pemodelan sistem informasi, analisis kebutuhan perangkat lunak, desain, generasi kode, pengujian dan pemeliharaan.



Gambar 2.2 Model Sequensial Linear

Berikut metode pengembangan perangkat lunak lainnya :

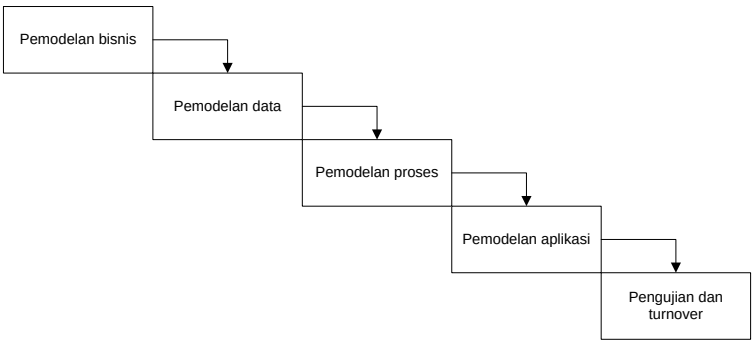
1. *Prototyping Model* (model paradigma) dimulai dengan pengumpulan kebutuhan, pengembangan dan pelanggan bertemu dan mendefinisikan objek keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui dan area garis besar dimana definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat [3]. Perancangan kilat membawa kepada konstruksi sebuah prototipe yang kemudian dievaluasi oleh pemakai. Dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. Siklus dari model prototipe dapat dilihat pada gambar 2.2 dibawah ini :



Gambar 2.3 Model Prototipe

2. *Rapid Application Development* (RAD), adalah sebuah model proses pengembangan perangkat lunak sequensial linear yang menekankan siklus pengembangan yang sangat pendek. Model RAD merupakan adaptasi "berkecepatan tinggi" dari linear sequential model pengembangan yang cepat

dapat diperoleh dengan menggunakan pendekatan konstruksi berbasis komponen [3]. Lihat gambar 2.3 berikut :



Gambar 2.4 Model *Rapid Application Development* (RAD)

3. *Evolutionary Software Process Model* adalah perulangan. Model ini dicirikan dengan pengembang versi-versi sistem yang lebih lengkap sedikit demi sedikit [3]. Model telah mempertimbangkan untuk mengkoordinasikan evaluasi produk, secara lengkap model ini terdiri dari :
   a. *Incremental model*, model ini mengkombinasikan antara linear sequential model dengan filosofi iteratif pada *prototyping*. Pada masing-masing sequen linear, yang menghasilkan perangkat lunak yang semakin meningkat kompleksitasnya.
   b. *Sipral model*, model ini digunakan oleh Boehm. Model ini menggabungkan antar sifat alami iterasi dati *prototyping* dengan aspek sistematik dan

terkendali dari linear sequential model. Model ini memberi peluang untuk pengembangan cepat.

2.5. SEO

SEO (*Search EngineOptimization*) adalah serangkaian proses yang dilakukan secara sistematis yang bertujuan untuk meningkatkan volume dan kualitas trafik dari mesin pencari menuju alamat situs web tertentu dengan memanfaatkan mekanisme kerja alami algoritma mesin pencari tersebut [5].

Praktisi SEO terkenal seperti *Rand Fishkin, Barry Schwartz, Aaron Wall and Jill Whalen* mencoba mempelajari beberapa pendekatan berbeda untuk *Search EngineOptimization*, dan telah menerbitkan opini mereka di forum online dan blog. Praktisi SEO mencoba mempelajari pola yang diterapkan oleh beragam *Search Engine* untuk melihat lebih dalam algoritma tersebut.

Berdasarkan *Jerri L. Ledford* (*Search EngineOprimization*), SEO adalah teknik pencarian yang menggunakan kata kunci atau frase yang mengandung indikator yang terkandung dalam halaman halaman web, informasi tersebut yang akan di index oleh mesin pencari [1].

Berdasarkan *David Viney* (*Get to the top on Google*), SEO adalah sebuah teknik pengoptimasian sebuah halam yang mengandung kata kunci atau frase yang bersangkutan yang akan di index oleh mesin pencari sebagai kata kunci pecarian [7].

Berdasarkan *Peter Kent* (*Search EngineOprimization for Dummies*), SEO adalah teknik yang di gunakan untuk mendapatkan posisi yang menguntungkan di mesin pencari yang sesui dengan kata kunci yang berada di halaman website [8].

Tujuan dari SEO adalah agar web/blog selalu berada di halaman terdepan dan teratas dari suatu *Search Engine*, sehingga selalu berada di posisi tearatas, maka besar kemungkinan web/blog sering dikunjungi.

2.5.1. Sejarah SEO [7]

Pada pertengahan tahun 1990-an ketika para webmaster dan penyedia konten mulai mengoptimalkan situs mereka di *Search Engine* yang mulai menggunakan katalog. Cara optimasi situs (SEO) saat itu adalah dengan mengirimkan alamat sebuah situs atau halaman situs dalam bentuk URL ke berbagai *search*.

Pada periode tersebut **algoritma *Search Engine*** belum terlalu kompleks sehingga mudah dimanipulasi. Versi awal algoritma pencarian didasarkan sepenuhnya pada informasi yang diperoleh pada saat "*crawling*" oleh "*spider*" pada bagian "*meta tag*" pada kode html situs web. "*Meta tag*" menyediakan informasi tentang konten yang terkandung pada suatu halaman web dengan serangkaian ***keyword***. Bagian inilah yang kemudian dimanipulasi dengan cara memasukkan *keyword* yang tidak relevan, tidak lengkap, dan tidak konsisten dengan konten situs, sehingga *Search Engine* salah menempatkan dan memeringkat situs tersebut. Hal ini menyebabkan hasil pencarian menjadi tidak akurat dan menimbulkan kerugian baik

bagi *Search Engine* maupun bagi pengguna internet yang mengharapkan informasi yang relevan dan berkualitas.

Search Engine yang yang dikembangkan oleh *Larry Page* dan *Sergey Brin*, dua orang mahasiswa doktoral ilmu komputer *Universitas Stanford*, cukup berhasil mengatasi permasalahan tersebut dengan mengembangkan *Backrub*, sebuah *Search Engine* yang sepenuhnya mengandalkan algoritma matematis untuk memeringkat halaman web.

Bisnis dan layanan SEO berkembang pesat seiring dengan pertumbuhan web, yang menyebabkan suatu situs web harus berjuang sekuat tenaga agar urlnya mudah ditemukan calon pelanggan di antara jutaan alamat situs lain dari seluruh dunia yang menjadi kompetitornya. *Search Engine* merupakan pintu masuk utama, karena pengguna internet tidak lagi sanggup (dan tidak ingin) menghafalkan alamat-alamat situs web, dan sebagai gantinya mereka mengandalkan indeks yang dibuat oleh *Search Engine*.

2.5.2. Fungsi SEO

Fungsi dari SEO antara lain :

a. Memberikan posisi terbaik di mesin pencari.
b. Memberikan trafic sesui dengan target pengunjung.
c. Mempopulerkan situs web site dengan cepat.

2.5.3. Kelebihan SEO

Dengan adanya mesin pencari saat ini, maka user akan di permudah dalam mencari dan mengetahui situs – situs yang akurat sesuai dengan informasi yang di butuhkan oleh user.

SEO dapat memberikan pengunjung informasi yang lebih akurat yang sesuai dengan isi *content* situs serta mempermudah user untuk mendapatkan informasi yang akurat dan sesuai dengan informasi yang di butuhkan.

2.5.4. Kekurangan SEO

Banyaknya situs web di seluruh dunia membuat mesin pencari harus semakin selektif dalam proses *indexing* sebuah halaman situs, maka mesin pencari membuat standarisasi dari konten situs web.

Maka memerlukan waktu yang cukup lama untuk mendapatkan pengunjung pada situs web yang di rekomendasikan oleh mesin pencari sesuai sesuai dengan isi konten situs web.

2.5.5. Strategi Penerapan SEO

Strategi penerapan SEO merupakan teknik yang paling efektif, untuk memaksimalkan kinerja situs web hosting sehingga situs web site yang menggunakan teknik ini akan lebih maksimal, berikut stategi yang di terapkan.

a. Usia *Domain*

Penerapan teknik SEO ini akan berjalan dengan baik jika usia *domain* usah memiliki usia minimal 6 bulan dari tanggal registrasi kepemilikan nama *domain*.

b. *Uniq Page Title*

Page Title adalah pemberian judul halaman web yang akan menjadi informasi pertama yang akan direkam oleh mesin, sehingga bisa di mengeti oleh pengunjung situs website.

c. *Meta Description*

Meta Description adalah tag untuk memberikan penjelasan mengenai halaman tersebut, ini akan menjadi info penting yang akan di simpan oleh mesin pencari yang akan di berikan pada user yang mencari sesui dengan kategori yang di maksud.

d. *URL*

URL adalah alamat sebuah halaman website yang akan memudahkan mensin pencari untuk membaca halaman yang sesui dengan kata kunci yang dimaksud.

e. *Site Navigation*

*Site Navigation*adalah untuk memberikan petunjuk kepada pengunjung bahwa alur link di dalam situs tersebut.

f. *Anchot Text*

*Anchot Text*berfungsi untuk memberikan tanda bahwa kategori tersebut berhubungan dengan halaman yang lainnya.

g. *Tags Appropriatly*

*Tags Appropriatly*berfungsi untuk menampilkan struktur halaman kepada pengguna.

h. *Images*

*Images*berfungsi untuk memudahkan pengunjung jika suatu file gambar tidak bisa di akses maka akan ada keterangan mengenai gambar tersebut.

i. *Robots*

Robots berfungsi untuk memberitahukan pada robot google bahwa halaman atau direktori mana saja yang tidak boleh di akses.

2.6. Teori Basis Data

2.6.1. Definisi basis data

Basis data terdiri dari 2 kata, yaitu basis dan data, basis dapat diartikan sebagai maskas atau gudang tempat bersarang atau berkumpul [4]. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli dan lain-lain), barang hewan, peristiwa, konsep keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang, seperti :

1. Himpunan kelompok data (*arsip*) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.
3. Kumpulan file, tabel, atau arsip yang saling berhubungan yang akan disimpan dalam media penyimpanan elektronis.

Basis data dalam lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengeluaran data dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengembalian data. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Maka basis data menggunakan media penyimpanan elektronik seperti disk (disket atau *hardisk*). Hal ini karena lemari arsip langsung dikelola oleh manusia sementara basis data dikelola melalui perantara alat atau mesin pintar elektronis (yang kita kenal komputer).

Satu hal yang harus diperhatikan, bahwa basis data bukan hanya sekedar tidak menyimpan data secara elektronis (dengan bantuan komputer), artinya tidak semua bentuk penyimpanan data secara elektronis bisa disebut basis data. Kita dapat menyimpan dokumen berisi data dalam file teks (dengan program pengolah data), *file spread* sheet dan lain-lain. Tetapi tidak bisa disebut basis data karena didalamnya tidak ada pemilahan dan pengelompokan data sesuai jenis dan fungsi data. Yang

sangat ditonjolkan basis data adalah pengaturan, pengelompokan, pengorganisasian data yang akan kita simpan sesuai dengan fungsinya.

2.6.2. Operasi dasar basis data

Didalam sebuah disk, basis data dapat diciptakan dan dapat pula ditiadakan. Sementara dalam sebuah basis data, kita dapat menempatkan satu atau lebih file ataupun table. Setiap basis data pada umumnya dibuat untuk mewakili sebuah basis data sebuah semesta data yang spesifik.

Operasi-operasi yang dapat kita lakukan berkenaan dengan basis data dapat meliputi :

1. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru.
2. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus dengan isinya, jika ada).
3. Pembuatan file atau tabel dari suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
4. Penghapusan file atau tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
5. Penambahan atau pengisian data baru kesebuah file atau tabel disebuah basis data (*insert*) yang identik dengan penambahan lembaran arsip kesebuah map arsip.
6. Pengambilan data dari sebuah file atau tabel (*retrieve atau search*), yang identik dengan pencarian lembaran arsip yang ada disebuah map arsip.

7. Pengubahan data dari sebuah file atau tabel (*update*).

2.6.3. Database Management System (DBMS)

Berikut ini akan dikemukakan definisi dari *DatabaseManagement System* (*DBMS*) :

"Manajemen Sistem Basis Data (Database Management System / DBMS) adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar".[7]

"DBMS diartikan sebagai suatu program komputer yang digunakan untuk memasukkan, mengubah, mengapus, memanipulasi dan memperoleh data/informasi dengan praktis dan efisien".[8]

DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, semisal penyimpanan data dalam file dan menulis kode aplikasi yang spesifik untuk pengaturannya.

Penggunaan *DBMS* untuk suatu aplikasi tergantung pada kemampuan dan dukungan *DBMS* yang beroperasi secara efisien. Sehingga agar bisa menggunakan *DBMS* dengan baik, perlu diketahui cara kerja dari *DBMS* tersebut. Pendekatan yang dilakukan untuk menggunakan *DBMS* secara baik meliputi : implementasi *DBMS* dan arsitektur secara mendetail untuk dapat memahami desain dari suatu basis data

2.6.4 Pemodelan Data

Pemodelan sistem memainkan peranan yang penting dalam pengembangan sistem. Pemodelan data kadang-kadang disebut pemodelan *database* karena model data kadang-kadang diimplementasikan sebagai sebuah *database*. Pemodelan data dapat digambarkan dengan ERD (*Entity Relationship Diagram*).

2.6.5. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram adalah suatu model atau teknik pendekatan yang dapat menyatakan atau menggambarkan hubungan *entity* didalam sebuah sistem, dimana hubungan tersebut dinyatakan sebagai *one-to-one*, *one-to-many* dan *many-to-many*. [5]

Entity relationship bertujuan untuk menentukan objek-objek data (*Entity*) dan *relationship* (hubungan) yang ada pada objek-objek tersebut.

Komponen *Entity Relationship Diagram* :

1. *Entity*

Entity adalah suatu yang dapat dibedakan atau diidentifikasikan secara unik, objek disini sifatnya berupa orang, tempat, peristiwa atau konsep yang informasinya direkam.

2. Atribut

Atribut adalah kumpulan dari beberapa elemen data yang membentuk suatu entity. Atribut merupakan ciri sebutan atau karakteristik yang dapat mewakili suatu simbol yang digunakan untuk menggambarkan atribut pada *entity relationship diagram.*

3. *Relationship*

Relationship adalah relasi atau hubungan yang terjadi diantara beberapa *entity*. Simbol yang digambarakan untuk menggambarkan *relationship* pada teknik *entity relationship diagram* adalah berupa belah ketupat.

Setiap *relationship* mempunyai derajat *relationship*, yaitu tingkat atau hubungan yang terdiri dari tiga derajat, yaitu :

a. *Unary* (derajat 1)

Suatu *relationship* yang dihubungkan dengan suatu *entity* set, dimana penghubungnya ada dua.

b. *Binary* (derajat 2)

Relationship derajat dua adalah *relationship* yang biasa terjadi yaitu dua *entity* set dihubungkan dengan satu *entity relationship*.

c. *Ternary* (derajat 3)

Relationship derajat tiga.

Relationship diagram juga mempunyai *Cardinalitas* yaitu batasan pada jumlah *entity* yang terdiri dari:

1. Relasi satu ke satu (*one-to-one*)

Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa satu berbanding satu.

2. Relasi satu ke banyak (*one-to-many*)

Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa banyak berbanding banyak atau banyak berbanding satu.

3. Relasi Banyak ke Banyak (*many-to-many*)

Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa banyak berbanding banyak.

2.7.Analisis Sistem

Penggambaran dan analisis model sistem yang digunakan untuk menggambarkan pembangunan perangkat lunak adalah analisis terstruktur. Analisis terstruktur adalah aktivitas pembangunan model dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional. Analisis terstruktur bukan merupakan metode tunggal yang diaplikasikan secara konsisten oleh semua yang menggunakannya. Pada model ini analisis terhadap perangkat lunak dapat digambarkan dalam bentuk *Flow Map*, Diagram konteks, *Data Flow Diagram* (DFD) dan Kamus Data.

2.7.1. Flowmap

Flowmap adalah representasi grafik dari sistem informasi, proses-proses, aliran-aliran data logis, masukan, keluaran, dan *file-file* serta entitas-entitas sistem operasi yang berhubungan dengan sistem informasi [5].

2.7.2. Diagram konteks

Diagram Konteks adalah bagian dari DFD yang berfungsi memetakan model model lingkungan, yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Diagram ini adalah diagram *level* tertinggi dari DFD. Diagram konteks menyoroti sejumlah karateristik penting sistem, yaitu :

1. Kelompok Pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi (sebagai terminator).
2. Data Masuk, yaitu data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
3. Data Keluar, yaitu data yang dihasilkan sistem dan diberikan kedunia luar.
4. Penyimpanan Data (*storage*), yaitu digunakan secara bersama antara sistem dengan terminator. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya dibuat oleh lingkungan dan digunakan oleh sistem. Hal ini berarti pembuatan simbol *data storage* dalam diagram konteks dibenarkan, dengan syarat simbol tersebut merupakan bagian dari luar sistem.
5. Batasan, yaitu batasan antara sistem dan lingkungan.

2.7.3. Data Flow Diagram (DFD)

DFD atau singkatan dari *Data Flow Diagram* merupakan representasi grafik dari suatu sistem yang menunjukan proses atau fungsi, aliran data, tempat penyimpan data dan *entitas eksternal*. DFD juga digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan. Dengan menggunakan DFD, rancangan yang akan kita buat akan lebih terarah dan lebih rinci. Sehingga kita tidak akan mengalami kesulitan dalam melakukan perancangan. *Data flow diagram* memiliki empat komponen, antara lain akan dijelaskan dibawah ini [6] :

1. *Terminator (external exitity)*

   Terminator mewakili *entity external* yang berkomunikasi dengan sistem yang sedang dikembangkan. *Terminator* merupakan kesatuan dilingkungan sistem. Biasanya terminator ini dikenal dengan nama entitas (*external*) sumber atau tujuan (*source* dan *sink*). Terminator dapat juga berupa departemen, divisi atau sistem diluar sistem yang berkomunikasi dengan sistem yang dikembangkan.

2. Proses

   Proses sering dikenal dengan nama *Bubble*, fungsi atau informasi. Komponen proses menggambarkan bagian dari sistem yang mentransformasikan *input* ke *output*, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi satu *input* atau lebih menjadi *output.* Setiap proses harus diberikan penjelasan yang lengkap sebagai berikut :

a. Identifikasi Proses

Umumnya berupa angka yang menunjukan nomor dari proses atau ditulis pada bagian atas simbol proses.

b. Nama Proses

Menunjukan apa yang dikerjakan oleh proses tersebut. Nama proses harus jelas dan lengkap menggambarkan bagian prosesnya nama proses diletakan dibawah identifikasi proses.

3. Penyimpanan data *(data store)*

Data store digunakan sebagai sarana untuk mengumpulkan data. *Data store* disimbolkan dengan dua garis horizontal yang paralel dimana tertutup pada salah satu ujungnya atau dua garis horizontal saja. *Data store* ini biasanya berkaitan dengan penyimpanan-pemyimpanan seperti *file* atau *database* yang berkaitan dengan penyimpanan secara komputerisasi, contohnya *file* pita magnetic, *file* disket atau *file harddisk*. *Data store* juga berkaitan dengan pemyimpanan data.

4. Alur Data *(Data Flow)*

Suatu data *flow* atau alur data dapat dipresentasikan dengan anak panah yang menunjukan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau satu paket data atau informasi dari suatu bagian sistem ke bagian lainnya. Selain menunjukan arah, alur data pada model yang dibuat dapat merepresentasikan bit, karakter, pesan,

formulir, bilangan *real* dan macam-macam informasi yang berkaitan dengan komputer.

2.7.4. Kamus data

Kamus data adalah suatu daftar atau kamus dari seluruh elemen-elemen data yang diperlukan oleh suatu sistem. Kamus data dibuat berdasarkan arus data yang terdapat pada DFD. Arus data dalam DFD bersifat global sehingga hanya dapat ditunjukkan nama arus datanya saja dan keterangan lebih lanjut mengenai struktur arus data dapat dilihat dari kamus data [5].

2.8. Normalisasi

Normalisasi adalah suatu teknik dalam menstruktur data dalam cara-cara tertentu untuk mengurangai atau mencegah timbulnya masalah yang berhubungan dengan pengolahan data dalam *database*. Normalisasi juga diartikan sebagai proses pengelompokan data elemen menjadi tabel-tabel yang menunjukan entitas dan relasinya.

Konsep-konsep pada normalisasi, antara lain :

1. Kunci atribut (*Key field* atau *Key attribut*) yaitu suatu kunci *field* yang mewakili *record* atau *tupple*
2. Kunci kandidat (*Candidate key*) yaitu suatu atribut atau satu set atribut yang mengidentifikasi secara unik suatu *entity*.

3. Kunci primer (*Primary key*) yaitu satu atribut atau satu set atribut yang yang mengidentifikasi secara unik dan mewakili setiap kejadian pada satu *entity*.
4. Kunci alternatif (*Alternate key*) yaitu kunci kandidat yang dipakai sebagai kunci primer.
5. Kunci tamu (*Foreign key*) yaitu suatu atribut atau satu set atribut dan melengkapi hubungan yang menunjukan ke induknya.

Bentuk-bentuk normalisasi yaitu :

1. Normal satu (1NF atau *First Normal Form*)

   Relasi berada pada normal kesatu jika semua atribut mempunyai nilai yang bersifat atomic.

2. Normal kedua (2NF atau *Second Normal Form*)

   Relasi berada pada normal kedua jika relasi tersebut merupakan normal satu dan atribut bukan merupakan kunci tergantung penuh pada kunci primer.

3. Normal ketiga (3NF atau *Three Normal Form*)

   Relasi berada pada normal ketiga jika relasi tersebut merupakan normal kedua dan atribut bukan kunci tidak tergantung secara transitif pada kunci primer.

4. BCNF (*Boyce Cood Normal Form*)

   Relasi berada pada BCNF jika dan hanya jika faktor penentunya adalah kunci kandidat dan relasi tersebut merupakan normal ketiga.

2.9. Hypertext Markup Language (HTML)

Dokumen HTML adalah file teks regular (disebut juga ASCII) yang diciptakan dengan menggunakan editor teks (misalnya, Emacs dalam Unix, Gnotepad pada Linux, atau Notepad pada Windows) atau dengan menggunakan word processor, tetapi harus disimpan dengan format *"text only with line breaks".*

Sesuai dengan namanya, bahasa ini menggunakan tanda (MarkUp) untuk menandai perintah-perintahnya. Saat ini banyak sekali aplikasi-aplikasi yang dapat digunakan untuk membuat web page, seperti Microsof Frontpage, Adobe GoLive, Macromedia Dreamweaver, Ultradev dan lain-lainnya. Namun untuk seorang web develover kemampuan dasar perintah HTML tetap dibutuhkan.

Homepage dibangun oleh sebuah file yang didalamnya terdapat kode-kode dan informasi dan dibaca oleh sebuah aplikasi yang bernama browser. Oleh browser, kode dan informasi akan disajikan sebagai sebuah halaman web yang kita kenal sebagai homepage. Kode tersebut haruslah ditulis menurut suatu aturan tertentu yang dapat dimengerti browser. Aturan tersebut dikenal dengan "bahasa pengkodean".

Untuk membuat homepage website, bahasa pengkodean yang digunakan adalah HTML. Sekalipun banyak orang menyebutnya sebagai salah satau bahasa pemrograman, HTML sebenarnya sama sekali bukan bahasa pemrograman karena HTML merupakan bahasa Mark Up (pengkodean) yang digunakan untuk menentukan format atau style dan teks yang ditandai. Simbol Mark Up yang digunakan oleh

HTML ditandai dengan tanda lebih kecil (<) dan lebih besar (>), keduanya dinamai tag. Sebagai contoh akan ditampilkan teks yang tercetak tebal, maka mark up yang digunakan adalah sebagai berikut:

<B> Teks ini bercetak tebal </B>

Untuk menandai bahwa sebuah file teks merupakan file HTML, maka ciri yang terlihat jelas adalah ekstensi filenya yaitu .htm atau .html. Namun labih jauh dari pada itu didalam file tersebut harus mengandung struktur sebagai berikut:

<HTML>

…….

…….

</HTML>

Tag <HTML> harus diletakan pada bagian awal dan tag </HTML> harus diletakan pada bagian terakhir dari suatu file HTML. Tag-tag HTML tidak bersifat case sensive artinya penggunaan huruf besar maupun huruf kecil tidak menjadi masalah. Jadi tag <HTML> akan sama dengan tag <html>.

2.10. Hypertext Preprocessor (PHP)

PHP (Personal Home Page Tools) adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang berjalan disisi *server*. Sebagian besar perintahnya berasal dari bahasa *C, Java* dan *Perl* dengan beberapa tambahan fungsi khusus PHP.

Bahasa ini memungkinkan para pembuat aplikasi web yang menyajikan HTML yang dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server. PHP biasa berinteraksi dengan hampir semua teknologi *web* yang sudah ada. *Developer* biasa menulis sebuah program PHP yang mengeksekusi suatu program CGI di *server* lain. Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs web yang besar dan sibuk, karena pemilik masih menggunakan aplikasi-aplikasi yang sudah terlanjur dibuat dimasa lalu dengan menggunakan CGI, ISAP atau dengan *script* seperti *Perl, awk* atau *Phyton* selama proses migrasi ke aplikasi baru yang dibuat dengan menggunakan PHP.

Kode program PHP menyatu dengan tag-tag HTML dalam satu file. Kode PHP diawali dengan tag <? Dan ditutup dengan tag ?>. File yang berisi tag HTML dan kode PHP ini diberi ekstensi .php atau ekstensi lainnya yang ditetapkan pada *Apache/web server*. Berdasarkan ektensi ini, pada saat file diakses, server akan tahu bahwa file ini mengandung kode PHP. Server akan menerjemahkan kode ini dan menghasilkan output dalam bentuk tag HTML yang akan dikirim ke *browserclient* yang mengakses file tersebut. Contohnya dapat dituliskan seperti ini :

```
<?
echo (“<html>
<head>
<title>Pemrograman PHP</title>
```

```
</head>

<body>

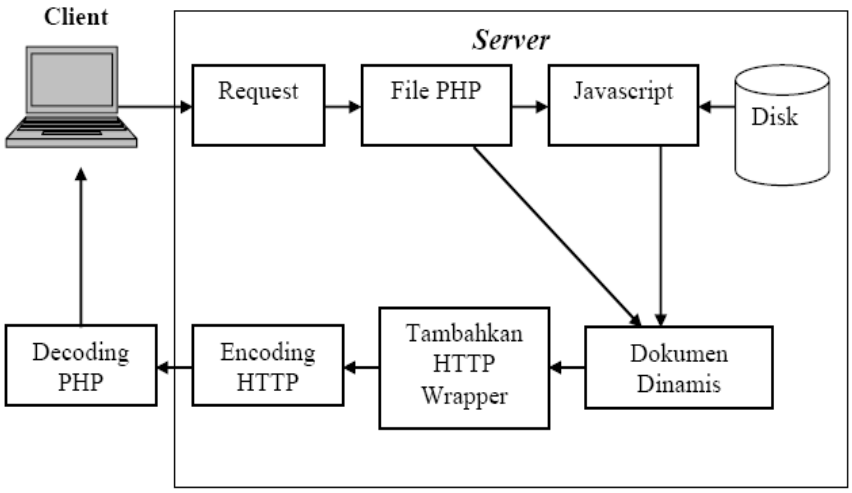
<center>Selamat menggunakan PHP</center>

</body>

</html>”);

?>
```

Dengan PHP, *developer* tidak perlu lagi berurusan dengan dua buah file yang terpisah seperti pada CGI. *Browser web* mengacu secara langsung ke file yang dituju, yang lalu dibaca oleh *server* sebagaimana file HTML statis biasa. Bedanya, sebelum dikirim balik ke *browser web*, *server web* memeriksa isi file dan menentukan apakah ada kode didalam file tersebut yang harus dieksekusi. Bila ada, kode-kode tersebut akan dieksekusi. Hasilnya akan dimasukkan kedalam dokumen yang sama. *Serverweb* bekerja secara langsung terhadap file yang bersangkutan tidak memanggil *script* terpisah seperti pada metode CGI. Seluruh kode di eksekusi di *server* (oleh karena itu disebut *server-side script*). Lihat gambar 2.4 dibawah ini untuk melihat kerja dari PHP dan Apache.



Gambar 2.5 Prinsip Kerja PHPdan*Apache*

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu :

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* biasa berkonsentrasi langsung pada penampilan webnya.
2. Tidak ada proses *compiling* dan *linking.*
3. Berorientasi objek *(object Oriented).*
4. Sintaksis pemrogramannya udah dipelajari, sangat menyerupai *C* dan *Perl.*
5. Integrasi yang sangat luas ke berbagai *server database*. Menulis *web* yang terhubung ke *database* menjadi sangat sederhana. Database yang didukung oleh PHP : *Oralce, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL,Adabas D, FilePRo, Velocis, Informic, dBase, UNIX dbm.*

2.11. Cascading Style Sheets (CSS)

tampilan suatu dokumen yang ditulis dalam bahasa *markup*. Penggunaan yang paling umum dari CSS *Cascading Style Sheets* (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur adalah untuk memformat halaman *web* yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh World Wide Web Consortium (W3C).

CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markup lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada stuktur isi.

CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh browser basis-suara atau pembaca layar), dan juga alat pembaca *braille*. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

2.12. JavaScript

JavaScript adalah bahasa *script* yang digunakan pada halaman HTML. *JavaScript* adalah varian dari *java* yang sangat terkenal dalam lingkungan pemrograman *web*. *JavaScript* mempunyai keunggulan pada sisi *client*, maupun diakses lebih cepat dari *script server*. Tetapi kelemahannya pada *script* ini belum tentu mampu diproses oleh *browser client* tergantung kompetibilitas *browser* tersebut. Cara untuk menjalankan j*avascript* hanya dengan *javascript-enabled* yaitu *browser* yang mampu menjalankan *javascript* seperti *Netscape Navigator* (versi 2.0 keatas) atau *internet ekplorer* (MSIE-versi 3.0 ke atas).

Beberapa hal mengenai JavaScript :

1) *JavaScript* didesain untuk menambah interaktif suatu *web*.
2) *JavaScript* merupakan sebuah bahasa *scripting* dengan bahasa pemrograman yang ringan.
3) *JavaScript* biasanya disisipkan (*embedded*) dalam halaman HTML.
4) *JavaScript* adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi.

Ada beberapa cara untuk menuliskan JavaScript yaitu :

a) *Script* pada bagian *head*

Skrip ini dieksekusi ketika dipanggil (biasanya berbentuk *function*) atau dipanggil berdasarkan *event* tertentu. Peletakan *script* di *head* akan menjamin *script* di-*load* terlebih dahulu sebelum digunakan.

```
<html>
<head>
<script type=”text/javascript”>
Function message ()
{
Alert (“Alert ini dipanggil saat even dijalankan”);
}
<script>
</head>
</html>
```

b) *Script* pada *Body*

Skrip akan dieksekusi ketika halaman di-*load* sampai dibagian <body> dan ketika ditempatkan berarti antara isi dan *javacript* dijadikan satu bagian.

```
<html>
<head>
</head>
<body>
<script type=”text/javascript”>
```

```
Document.write(“Pesan ini akan tampil”);
</script>
</body>
</html>
```

c) *External JavaScript*

Untuk memudahkan menulis *script* yang berulang-ulang di setiap halaman maka *JavaScript* dapat ditulis di *file* secara *ekternal*. Jadi, antara HTML dan *JavaScript* dipisah, kemudian *file* tersebut dipanggil dari dokumen HTML. *File JavaScript* tersebut disimpan dengan ektensi .js.

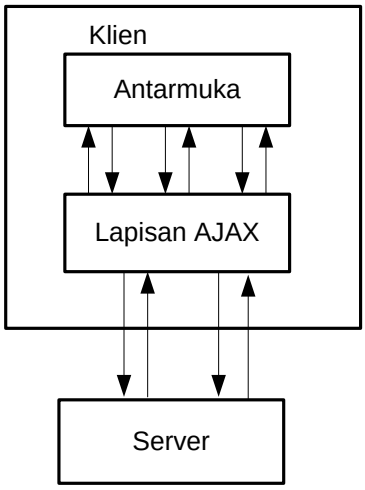
```
<html>
<head>
</head>
<body>
<script src=”xxx.js”>
</script>
</body>
</html>
```

2.13. Asynchronous JavaScript and XML (AJAX)

AJAX (*Asynchronous JavaScript and XML*) sesungguhnya bukanlah suatu bahasa pemrogarman, melainkan hanya suatu teknik yang memungkinkan untuk membuat aplikasi *web* yang interaktif. Dengan menggunakan AJAX, aplikasi *web* dapat berinteraksi dengan server di latar belakang sehingga tidak mempengaruhi halaman

web secara keseluruhan. Sebagai gambaran, Anda mungkin sering mendaftarkan diri pada suatu situs *web* sebagai anggota baru. Setelah Anda selesai mengetikan *user ID* saat registrasi, sistem bisa memberitahukan bahwa nama yang Anda pakai sudah digunakan oleh orang lain atau belum.

Bagaimana aplikasi *web* ini bisa tahu hal itu padahal Anda belum mengklik tombol apapun dalam halaman *web*. Rahasianya terletak pada pemrosesan di latar belakang yang dilakukan oleh AJAX. Gambar 2.5 Menunjukan aplikasi *web* yang berbasis AJAX.



Gambar 2.6 Model Aplikasi Web dengan AJAX

Nampak bahwa pada aplikasi *web* yang berbasis AJAX terdapat bagian yang disebut lapisan AJAX (atau terkadang disebut mesin AJAX). Lapisan inilah yang mengatur komunikasi antara *client* dan *server*.

Butir penting yang perlu diperhatikan dalam lapisan AJAX :

1. Lapisan AJAX meminta layanan ke *server* atau tidak, tergantung pada jenis layanan antarmuka. Sebagai contoh, untuk validasi terhadap objek formulir tidak perlu melakukan permintaan ke *server*. Cukup ditangani di sisi *client*. Namun untuk pemeriksaan keabsahan *password*, mau tidak mau harus dilayangkan ke *server* karena harus berhubungan dengan database.
2. Sekalipun penambahan lapisan AJAX berkesan memperlambat proses (karena komunikasi ke server tidak lagi dilakukan secara langsung), pada kenyataanya yang terjadi adalah sebaliknya. Hal ini karena permintaan yang dilakukan oleh lapisan AJAX menghasilkan informasi yang jauh lebih sedikit dibanding kalau yang diminta adalah keseluruhan halaman.
3. Lapisan AJAX bekerja dengan pendekatan asinkron. Asinkron berarti bahwa *client* bisa meminta layanan dari *server* dan tidak perlu menunggu sampai *server* melayaninya. Dengan demikian, saat informasi dari *server* belum didapat, layanan kepada pemakai tetap bisa dilakukan.
4. Komunikasi antara *client* dan *server* tidak perlu mengubah keseluruhan antarmuka dalam halaman *web*, tetapi cukup hanya bagian tertentu yang terpengaruh. Hal seperti ini tidak bisa dilakukan pada aplikasi *web* yang tidak menggunakan AJAX.

AJAX pertama kali ditemukakan pada tahun 2005 oleh Jesse James Garret, presiden dan pendiri perusahaan Adaptive Path. Menurut beliau, nama AJAX diberikan untuk memudahkan dalam berkomunikasi dengan klien daripada kalau menyebutnya "*Asynchronous JavaScript + CSS + DOM + XMLHttp Request*". Teknik

ini sebenarnya sudah ada sejak 1999. Pada masa itu Microsoft membuat objek bernama *XMLHttpRequest* (Objek yang kelak akan dilibatkan dalam membuat AJAX) sebagai control *ActiveX* pada Internet Explorer 5. AJAX menjadi sangat populer baru setelah digunakan intensif oleh Yahoo! Dan Google.

Menurut Jesse James Garret, di dalam artikelnya yang berjudul "*AJAX A New Approach to Web Applications*", AJAX yang lengkap mencakup [9]:

1. XHTML (*Extensible HyperText Markup Language*), sebagai peluasan dari HTML.
2. CSS (*Cascading Style Sheets*) sebagai pemformat bagian dari dokumen.
3. DOM (*Document Object Model*) untuk mengatur interaksi dan penampilan isi HTML dan XHTML secara dinamis.
4. XML (*Extensible Markup Language*) memungkinkan untuk melakukan pertukaran data.
5. XSLT (*Extensible Stylesheet Language Transformation*) adalah bahasa berbasis XML yang ditujukan untuk melakukan transformasi dokumen XML ke bentuk yang lain.
6. *XMLHttpRequest* untuk melakukan pertukaran data secara asinkron.
7. *JavaScript* sebagai bahasa untuk membentuk lapisan AJAX.

Memang nampaknya sangat kompleks karena melibatkan banyak hal. Namun tidak pelu khawatir bahwa hal itu akan menyusahkan dalam mewujudkannya. Dalam

praktik, AJAX bisa diimplementasikan tanpa menggunakan XML, XSLT, maupun CSS.

Software yang digunakan untuk membangun aplikasi *web* dengan AJAX :

1. *Browser* (misalnya *Internet Explorer* dan *Mozilla Firefox*) pada sisi klien.
2. *Web Server* (misalnya *Apache*) pada sisi server.

Tentu saja yang dicantumkan di atas adalah kebutuhan minimal untuk membuat aplikasi *web* yang melibatkan AJAX. Jika aplikasi juga melibatkan database, di sisi server juga perlu dipasang *database server.* Selain itu, agar *web server* dapat berkomunikasi dengan *database server*, perantara seperti PHP diperlukan.

2.14. MySQL

MySQL adalah multi *user database* yang menggunakan bahasa *Strured Query Language* (SQL). MySQL merupakan database server multi user dan multi *threaded* yang tangguh. Dengan memiliki banyak feature MySQL bisa bersaing dengan database komsersial sekalipun. Tidak mengejutkan MySQL menjadi database pilihan untuk banyak pengguna PHP. MySQL termasuk RDBMS (*Relational Database Management System*) itulah sebabnya istilah seperti tabel, baris dan kolom digunakan pada MySQL. Pada MySQL sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri dari sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

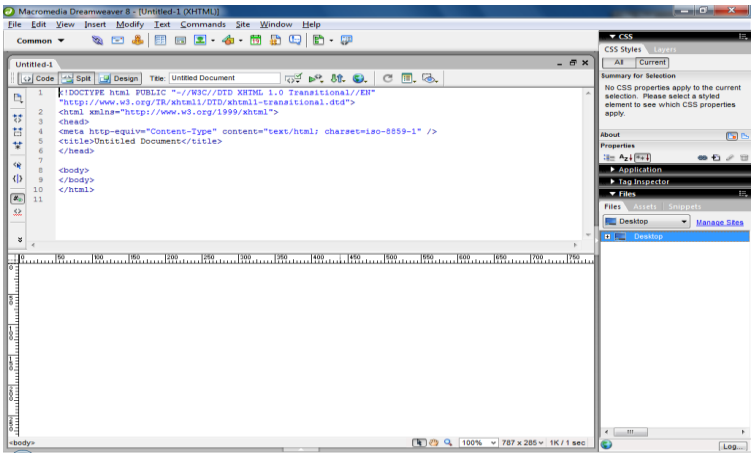
MySQL mampu menangani data yagn cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data. SQL adalah sebuah standar yang digunakan untuk mengakses *databaseserver*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industry.

Dengan menggunakan MySQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan menggunakan *dBASE* atau *Clipper* yang masih menggunakan perintah-perintah pemrograman. MySQL merupakan *softwaredatabase* yang masih populer di lingkungan Linux, kepopuleran ini sangat ditunjang oleh performa *query* dari databasenya yang saat ini bisa dikatakan paling cepat dan jarang bermaslah. MySQL ini juga sudah berjalan pada lingkungan windows.

2.15. Pengenalan *Macromedia Dreamweaver*

Macromedia Dreamweaver adalah sebuah HTML editor professional untuk mendesain secara visual dan mengelola situs *web* maupun halaman *web*. *Macromedia Dreamweaver* merupakan *software* utama yang digunakan oleh *Web Designer* maupun *Web Progremer* guna mengembangkan situs *web*. Ruang kerja, fasilitas dan kemampuan *Macromedia Dreamweaver* mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun situs *web*.

Salah satu kelebihan *Dreamweaver* adalah ruang kerja *Dreamweaver* serta *tool* yang tersedian dapat digunakan dengan sangat mudah dan cepat, sehingga dapat memudahkan membangun suatu *website* dengan cepat dan tanpa harus melakukan *coding*. *Dreamweaver* juga memiliki dua bentuk layer, yaitu bentuk halaman *Design* dan halaman *Code*. Dua bentuk layer ini akan mempermudah dalam menambahkan *script* yang berbasis PHP maupun javascript. *Dreamweaver*, selain mendukung pembuatan *web* yang berbasis HTML, juga dapat mendukung program–program *web* yang lain diantaranya PHP, ASP, Perl, Javascript, dan lain-lain.



Gambar 2. 7 Tampilan Macromedia Dreamweaver 8

2.16. Smart Recommendation system

Dalam buku Mass Customization (Pine, 1993), Joe Pine berpendapatbahwa perusahaan perlu untuk beralih dari dunia lama pada massaproduksi di mana "produk standar, pasar homogen,dan jangka waktu produk serta pengembangan siklus adalah aturan"untukdunia baru dalam perusahaan dimana "keragaman dan kustomisasi

menggantikan standar produk" Pine berpendapat bahwa membangunan satu produk tidak dapat menguntungkan lagi. Perusahaan perlu untuk dapat Minimal, mengembangkan beberapa produk yang memenuhi beberapakebutuhan dari beberapa pelanggan. Gerakan ke arah *e-commerce*telah memungkinkan perusahaan untuk menyediakan pelanggan dengan lebih banyak pilihan.

Namun, dalam memperluas ke tingkat baru kustomisasi,bisnis meningkatkan jumlah informasi bahwa pelangganharus mengikuti proses sebelum mereka dapat memilih item yang telah memenuhi Sistem Recommender yang telah digunakan yang di e-commerce untuk menyarankanproduk kepada pelanggan mereka. Produk dapat direkomendasikanberdasarkan penjual keseluruhan atas pada situs, berdasarkandemografi pelanggan, atau berdasarkan analisa dari masa laluperilaku pembelian pelanggan sebagai prediksi untuk membeli masa depanperilaku. Secara umum, teknik ini merupakan bagian dari personalisasi di situs, karena dapat membantu aplikasi menyesuaikan diri dengan setiap pelanggan.Sistem Recommender mengotomatisasi personalisasi di Web,memungkinkan personalisasi individu untuk setiap pelanggan.Personalisasi sampai batas ini adalah salah satu cara untuk merealisasikan ide Pinedi Web. Dengan demikian, Pine mungkin akan setuju dengan Jeff Bezos,CEO Amazon.com ™, saat ia mengatakan "Jika saya memiliki 2 jutapelanggan di Web, saya harus memiliki 2 juta toko diWeb. "Sistem Recommender meningkatkan penjualan *e-commerce* dalam tiga cara:

1. Browser menjadi pembeli: Pengunjung ke situs Web dapat melihat produk tanpa harus membeli sesuatu. Recommender system dapat membantu pelanggan menemukan produk yang mereka ingin membeli.
2. Cross-menjual: sistem Recommender meningkatkan penjualan silang olehmenunjukkan produk tambahan untuk pelanggan untuk membeli. Jikarekomendasi bagus, rata-rata pesanan harusmeningkat. Misalnya, situs mungkin merekomendasikan tambahanproduk dalam proses penyelesaian, berdasarkan produk tersebut sudahdi keranjang belanja.
3. Loyalitas: Dalam dunia di mana pesaing sebuah situs hanya satu klik ataudua jauh, memperoleh loyalitas pelanggan adalah bisnis yang pentingstrategi (Reichheld dan Sesser, 1990) (Reichheld, 1993).Sistem Recommender meningkatkan loyalitas dengan menciptakan nilai tambahhubungan antara situs dan pelanggan. Situs berinvestasi dalambelajar tentang pengguna mereka, menggunakan sistem recommender untukmengoperasionalkan bahwa belajar, dan tampilan antarmuka yang hadirsesuai dengan kebutuhan pelanggan. Pelanggan membayar situs tersebut dengan kembalike yang paling sesuai kebutuhan mereka. Semakin banyak pelangganmenggunakan sistem rekomendasi - mengajarkannya apa yang mereka inginkan -yang lebih setia mereka ke situs. Dan pada akhirnya, menciptakan hubungan antara pelanggan juga dapatmeningkatkan loyalitas. Pelanggan akan kembali ke situs yangmerekomendasikan orang dengan siapa mereka akan menyukai untuk berinteraksi.

2.17. Collaborative Filtering

Collaborative filtering(CF) adalah teknik yang digunakan oleh beberapa sistem recommender . *Collaborative filtering* memiliki dua pengertian, yang sempit dan yang lebih umum. [1] Secara umum, kolaboratif penyaringan adalah proses penyaringan untuk informasi atau pola menggunakan teknik yang melibatkan kolaborasi antara beberapa agen, sudut pandang, sumber data, dll [1]Aplikasi penyaringan kolaboratif biasanya melibatkan sangat besar set data. Metode penyaringan kolaboratif telah diterapkan untuk berbagai jenis data termasuk sensing dan data pemantauan - seperti dalam eksplorasi mineral, penginderaan lingkungan di daerah yang luas atau beberapa sensor, data keuangan - seperti lembaga jasa keuangan yang mengintegrasikan sumber keuangan banyak, atau secara elektronik *commerce* dan web 2.0 aplikasi di mana fokusnya adalah pada data pengguna, dll sisa diskusi ini berfokus pada penyaringan kolaboratif untuk data pengguna, meskipun beberapa metode dan pendekatan berlaku untuk aplikasi besar lainnya juga.

Dalam arti, lebih baru sempit, kolaboratif penyaringan adalah metode membuat prediksi otomatis (*filtering*) tentang kepentingan pengguna dengan mengumpulkan preferensi atau selera informasi dari banyak pengguna (berkolaborasi). Asumsi yang mendasari pendekatan penyaringan kolaboratif adalah bahwa jika seseorang *A* memiliki pendapat yang sama sebagai orang *B* pada suatu masalah, adalah lebih mungkin untuk memiliki pendapat B pada *x* isu yang berbeda daripada memiliki pendapat pada x dari orang yang dipilih secara acak. Misalnya,

penyaringan kolaboratif rekomendasi sistem untuk televisi selera bisa membuat prediksi tentang yang televisi menunjukkan pengguna harus ingin memberikan sebagian daftar selera pengguna tersebut (suka atau tidak suka). [2] Perhatikan bahwa prediksi ini khusus untuk pengguna, tapi menggunakan informasi yang diperoleh dari banyak pengguna. Hal ini berbeda dengan pendekatan yang lebih sederhana dari memberikan rata skor (tidak spesifik) untuk setiap item yang menarik, misalnya berdasarkan jumlah kantor suara .

2.17.1. Pendahuluan

Pertumbuhan Internet telah membuatnya menjadi jauh lebih sulit untuk secara efektif mengekstrak informasi yang berguna dari semua informasi yang tersedia secara *online*. Jumlah besar data memerlukan mekanisme untuk penyaringan informasi yang efisien. Salah satu teknik yang digunakan untuk menangani masalah ini disebut penyaringan kolaboratif.

Motivasi untuk penyaringan kolaboratif berasal dari gagasan bahwa orang sering mendapatkan rekomendasi terbaik dari seseorang dengan rasa serupa. *Collaborative filtering* mengeksplorasi teknik untuk mencocokkan orang dengan minat yang sama dan membuat rekomendasi atas dasar ini.

Algoritma penyaringan kolaboratif sering membutuhkan (1) partisipasi aktif, (2) cara mudah untuk mewakili pengguna pengguna kepentingan ke sistem, dan

(3) algoritma yang dapat mencocokkan orang dengan minat yang sama.Biasanya, alur kerja dari sistem penyaringan kolaboratif adalah:

1. Seorang pengguna mengungkapkan nya atau preferensi nya dengan item rating (misalnya buku, film atau CD) sistem. Peringkat ini dapat dilihat sebagai representasi perkiraan kepentingan pengguna dalam domain yang sesuai.

2. Sistem ini cocok peringkat pengguna ini terhadap pengguna lain dan menemukan orang dengan paling "mirip" selera.

3. Dengan pengguna yang sama, sistem merekomendasikan item bahwa pengguna serupa telah dinilai sangat namun belum sedang dinilai oleh pengguna ini (mungkin tidak adanya Peringkat sering dianggap sebagai ketidakbiasaan dari item)

Masalah utama dari penyaringan kolaboratif adalah bagaimana menggabungkan dan bobot preferensi tetangga pengguna. Terkadang, pengguna segera dapat menilai item yang direkomendasikan. Akibatnya, sistem keuntungan representasi semakin akurat dari preferensi pengguna dari waktu ke waktu.

2.17.2. Metodologi

Sistem penyaringan Kolaborasi memiliki banyak bentuk, tetapi sistem yang umum dapat dikurangi menjadi dua langkah:

1. Carilah pengguna yang berbagi pola rating yang sama dengan user aktif (*user* yang prediksi adalah untuk).

2. Gunakan peringkat dari orang-orang yang berpikiran pengguna ditemukan pada langkah 1 untuk menghitung prediksi bagi pengguna aktif

Ini jatuh di bawah kategori pengguna berbasis penyaringan kolaboratif. Sebuah aplikasi khusus ini adalah pengguna berbasis algoritma *Nearest Neighbor* .Atau, barang berbasis penyaringan kolaboratif ditemukan oleh *Amazon.com* (pengguna yang membeli x juga membeli y), seluruh penerimaan secara item-sentris:

1. Membangun matriks item-item yang menentukan hubungan antara pasangan item

2. Menggunakan matriks, dan data pada pengguna saat ini, menyimpulkan seleranya dalam melihat, misalnya, Lereng Satu item yang berbasis keluarga penyaringan kolaboratif.

Bentuk lain dari penyaringan kolaboratif dapat didasarkan pada pengamatan implisit dari perilaku user biasa (yang bertentangan dengan perilaku buatan dikenakan oleh tugas rating). Dalam sistem ini Anda mengamati apa yang user telah dilakukan bersama-sama dengan apa semua pengguna telah dilakukan (apa musik yang mereka telah mendengarkan, barang apa yang mereka telah membeli) dan menggunakan data tersebut untuk memprediksi perilaku pengguna di masa depan

atau untuk memprediksi bagaimana pengguna mungkin ingin berperilaku jika saja mereka diberi kesempatan. Prediksi ini kemudian harus disaring melalui logika bisnis untuk menentukan bagaimana prediksi ini dapat mempengaruhi apa sistem bisnis harus dilakukan. Hal ini, misalnya, tidak berguna untuk menawarkan untuk menjual musik seseorang jika mereka sudah telah menunjukkan bahwa mereka sendiri yang musik atau, mengingat contoh lain, tidak berguna untuk menyarankan panduan wisata lebih untuk Paris untuk seseorang yang sudah membeli panduan perjalanan untuk kota ini.

Mengandalkan sistem penilaian atau *rating* yang rata-rata dari semua pengguna mengabaikan tuntutan spesifik dari pengguna, dan sangat miskin di tugas di mana ada variasi besar dalam kepentingan, misalnya dalam rekomendasi dari musik. Namun, ada metode lain untuk memerangi ledakan informasi, misalnya *web* pencarian, pengelompokan data , dan banyak lagi.

2.17.3. Jenis-jenis *collaborative filtering*

Schafer membagi algoritma *collaborative filtering* ke dalam dua kelas yangberbeda menurut teori dan kepraktisannya, yaitu algoritma non-probabilistik dan algoritma probabilistik. Suatu algoritma dianggap probabilistik bila algoritma tersebut berdasarkan model probabilistik. Algoritma tersebut mewakili distribusi probabilitas saat menghitung prediksi rating atau daftar rangking rekomendasi. Algoritma non-probabilistik yang terkenal yaitu *nearest neighbours algorithm*.

Algoritma ini dibagi menjadi dua kelas yaitu *user-based* dan *item-based* (Schafer dkk, 2007).

(a) *User-Based Collaborative Filtering*

User-based nearest neighbour algorithm menggunakan teknik statistika untuk menemukan sekumpulan pengguna, dikenal sebagai tetangga (*neighbour*), yang memiliki sejarah setuju dengan pengguna yang menjadi sasaran. Setelah sekumpulan tetangga terbentuk, sistem menggunakan algoritma yang berbeda untuk menggabungkan kesukaan *neighbours* untuk menghasilkan prediksi atau rekomendasi N-teratas untuk *active user*.

(b) *Item-to-Item Collaborative Filtering*

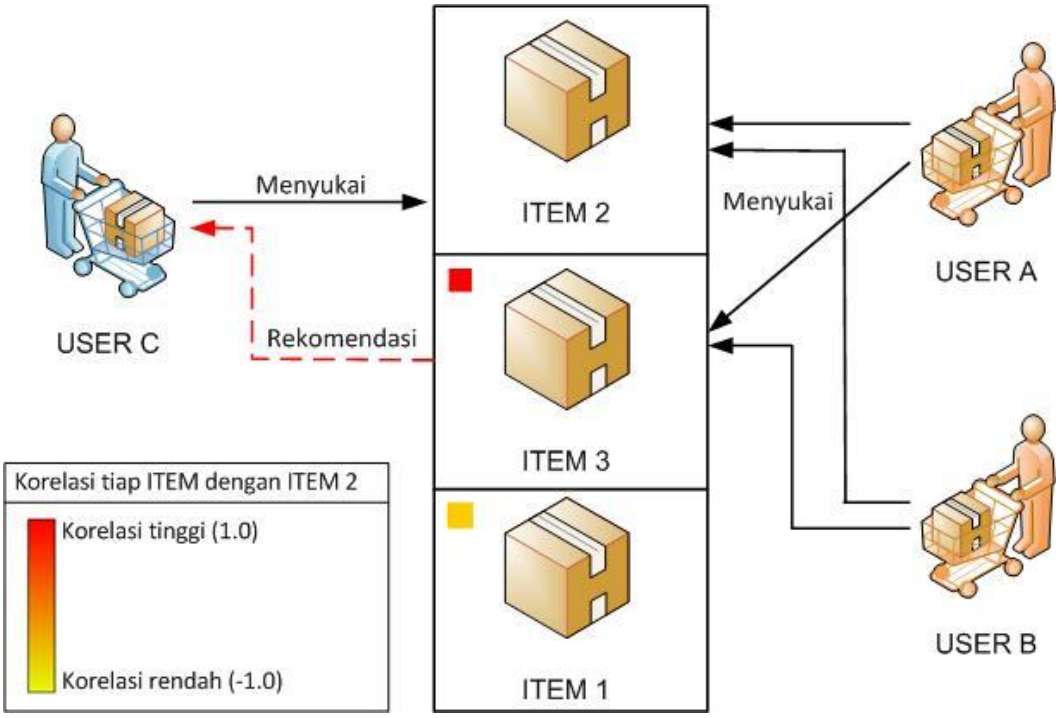
Item-based collaborative filtering merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian rating terhadap suatu produk dengan produk yang dibeli. Dari tingkat kesamaan produk, kemudian dibagi dengan parameter kebutuhan pelanggan untuk memperoleh nilai kegunaan produk. Produk yang memiliki nilai kegunaan tertinggilah yang kemudian dijadikan rekomendasi. Metode ini muncul sebagai solusi untuk beberapa permasalahan pada *user-based collaborative filtering* yaitu pada masalah keterbatasan (*sparsity*) dan skalabilitas serta masalah waktu dan memori.

Pada metode ini akan diketahui nilai similaritas antar item dengan tingkat persebaran rating kecil dan nilai similaritas antar item cenderung lebih jarang berubah dibandingkan dengan nilai similaritas antar pengguna.*Item-*

basedcollaborative filtering melakukan similaritas dengan membentuk suatu model similaritas secara *offline* yang secara otomatis akan menghemat waktu dan memori yang digunakan untuk penghitungan pada saat pengguna mengakses halaman situs.

2.17.4 Formula *item-base collaborative filtering*

Menu rekomendasi merupakan menu yang akan menampilkan barang yang akan direkomendasikan kepada pelanggan tersebut. Rekomendasi tersebut berasal dari perhitungan algoritma pembangkit rekomendasi. Barang yang memiliki nilai kegunaan tertinggilah yang kemudian dijadikan rekomendasi. Pelanggan akan mendapatkan rekomendasi buku apabila pelanggan tersebut telah melakukan rating satu atau lebih item yang disediakan oleh sistem rekomendasi. Jadi jika pelanggan tersebut belum melakukan rating terhadap item di sistem, maka pelanggan tersebut akan mendapatkan rekomendasi rata-rata dari jumlah keseluruhan rating pelanggan.



Gambar 2.8 *Ilustrasi Skema Item-based Collaborative Filtering*

Gambar 2.3 diatas menggambarkan bagaimana *item-based collaborative filtering*memberikan rekomendasi terhadap *user* C, *item* yang direkomendasikan terhadap*user*C adalah *item*3. Hal tersebut dikarenakan*item* 3 mempunyai korelasi yang tinggi dengan *item* 2 yang telah disukai sebelumnya oleh *user* C.

Berikut adalah persamaan *adjusted cosine similarity* yang digunakan untuk menghitung nilai kemiripan diantara *item.*

Persamaan *adjusted cosine* :

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_u)^2}} \quad \ldots(2.1)$$

Keterangan :

$sim(i,j)$ = Nilai kemiripan antara *item* i dan *item* j

$u \in U$ = Himpunan *user* u yang merating *item* i dan *item* j

$R_{u,i}$ = *Ratinguser* u pada *item* i

$R_{u,j}$ = *Rating user* u pada *item* j

$\bar{R}_u$ b = Nilai rata-rata *rating user* u

Dalam menghitung nilai kemiripan, nilai yang akan dihasilkan oleh persamaan *adjusted-cosine similarity* adalah berkisar antara +1.0 dengan -1.0, sedangkan informasi korelasi antara dua *item* diketahui jika (Marmanis H, 2009):

- Nilai kemiripan 0 : Kedua *item* tidak berkorelasi (independen).
- Nilai kemiripan mendekati +1.0 : Kedua *item* cenderung akan mirip antara satu dengan yang lainnya, jadi apabila *rating* suatu *item* telah diketahui maka *rating item* yang lainnya dapat diketahui dan disimpulkan dengan probabilitas yang tinggi.
- Nilai kemiripan mendekati -1.0 : Kedua *item* saling bertolak belakang dan dalam kasus ini juga *rating* suatu item bisa ditentukan berdasarkan

rating dari *item* lainnya, tapi keadaannya sekarang apabila *rating item* pertama meningkat maka *rating item* kedua justru akan sebaliknya, yaitu menurun.

Tahapan selanjutnya yang paling penting dalam proses *collaboratve filtering* adalah membuat prediksi. Setelah mendapatkan sekumpulan *item* yang sangat mirip berdasarkan perhitungan kemiripan, dilakukan proses prediksi yang nantinya akan memperkirakan nilai *rating* dari *user* bagi suatu *item* yang belum pernah dirating sebelumnya oleh *user* tersebut.

Teknik yang digunakan untuk mendapatkan nilai prediksi adalah dengan persamaan *weighted sum*, teknik ini memprediksi *item* j untuk *user* u dengan menghitung jumlah nilai *rating* yang diberikan oleh *user* terhadap *item* yang berkorelasi dengan *item* j. Setiap *rating* yang diberikan pada *item* yang berkorelasi, akan dikalikan dengan nilai kemiripannya. Kemudian dibagi dengan jumlah nilai absolut kemiripan seluruh *item* yang berkorelasi.

Persamaan *weighted sum* :

$$P_{(u,j)} = \frac{\sum_{i \in I}(R_{u,i} * S_{i,j})}{\sum_{i \in I}|S_{i,j}|} \quad \ldots\ldots\ldots\ldots (2.2)$$

Keterangan :

$P_{(u,j)}$ = Prediksi untuk *user* u pada *item* j

$i \in I$ = Himpunan *item* yang mirip dengan *item* j

$R_{u,i}$ = *Ratinguser* u pada *item* i

$S_{i,j}$ = Nilai kemiripan antara *item* i dan *item* j

Sebagai contoh dalam menghitung prediksi dengan persamaan *weightedsum*, diketahui tabel 2.1 dibawah dengan data *rating* dari *user u* untuk setiap *item* diantaranya *item* 1, *item* 2, dan *item* 3. Untuk *item* yang akan diprediksi ratingnya adalah *item* 4

2.17.4.1 AlgoritmaK-*Nearest Neighbor* (KNN)

Algoritma K-*nearestneighbor* (KNN) merupakan metode pengklasifikasian data yang bekerja relatif dengan cara yang lebih sederhana dibandingkan dengan metode pengklasifikasian data lainnya. Algoritma ini berusaha mengklasifikasikan data baru yang belum diketahui *class*-nya dengan memilih data sejumlah K yang letaknya terdekat dari data baru tersebut. *Class*terbanyak dari data terdekat sejumlah K tersebut dipilih sebagai *class* yang diprediksikan untuk data yang baru (Yudi Agusta, 2008).

Dalam konteks penelitian ini, yang dimaksud dengan data terdekat adalah data yang mempunyai nilai kemiripan yang tinggi, dimana proses untuk penghitungan nilai kemiripan sudah dijelaskan pada poin sebelumnya yaitu pada

persamaan 2.1.

Algoritma KNN ini diterapkan untuk menentukan berapa jumlah *neighbor* (*item*) terdekat yang digunakan dalam proses menghitung prediksi *rating* suatu *item* oleh *user*sehingga menghasilkan tingkat akurasi prediksi yang tinggi. Implementasi dari algoritma KNN pada metode *item-based collaborativefiltering* ini akan dijelaskan pada poin selanjutnya.

2.17.4.2 *Item-based* K-*Nearest Neighbor* (KNN)

Sebenarnya teknik ini merupakan pengembangan dari teknik sebelumnya, namun pada *item-based* KNN ini diterapkan algoritma KNN dalam menetukan prediksinya (Zheng Wen, 2008). KNN menentukan sejumlah K *neighbor* terdekat untuk setiap *item* yang memiliki korelasi.*Weightedsum* masih digunakan untuk mendapatkan prediksi, tapi dengan menambahkan sedikit perubahan :

$$P_{(u,j)} = \frac{\sum_{i \in I_u^K(j)} (R_{u,i} * S_{i,j})}{\sum_{i \in I_u^K(j)} |S_{i,j}|} \quad \ldots\ldots\ldots\ldots (2.3)$$

Keterangan :

$P_{(u,j)}$ = Prediksi untuk *user* u pada *item* j

$i \in I_u^K(j)$ Himpunan K *item* yang mirip dengan *item* j

$R_{u,i}$ = *Ratinguser* u pada *item* i

$S_{i,j}$ = Nilai kemiripan antara *item* i dan *item* j

2.17.4.3 Karakteristik *Item-based Collaborative Filtering*

Ada beberapa karakteristik yang dimiliki oleh teknik *item-based collaborative filtering* (Alex Lin, 2009), yaitu :

1. *Data Sparsity Problem*

   Keadaan dimana banyak data yang kosong untuk setiap sel dalam tabel matriks *rating user-item*. Untuk teknik *item-based*, hanya sel yang terisi yang akan diproses dalam mengkalkulasi prediksi.

2. *Cold Start Problem*

   Keadaan ini terjadi saat awal *item* atau *user* baru saja dimasukan kedalam sistem, sehingga mesin sistem rekomendasi tidak mempunyai pengetahuan yang banyak untuk menghasilkan rekomendasi yang akurat.

3. Scalable

   Teknik ini sering digunakan untuk sistem rekomendasi yang menggunakan banyak sekali jumlah data, seperti pada Amazon.com, dan terbukti teknik ini sangat efisien untuk masalah banyaknya jumlah data yang dikalkulasi dalam membangun rekomendasi.

Penelitian tentang sistem rekomendasi telah menggunakan beberapa jenis pengukuran untuk mengevaluasi kualitas dari sistem rekomendasi. Salah satunya jenis*statistical accuracy metrics* dimana mengukur akurasi sistem dengan membandingkan nilai yang diprediksi dengan nilai yang sebenarnya. *Mean absolute*

error (MAE) merupakan persamaan yang termasuk jenis *statistical accuracy metrics* dimana sesuai dengan namanya yaitu menghitung nilali rata-rata selisih antara nilai prediksi dengan nilai yang sebenarnya (George Karypis, 2001).

Persamaan MAE :

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \quad \ldots\ldots\ldots (2.4)$$

Keterangan :

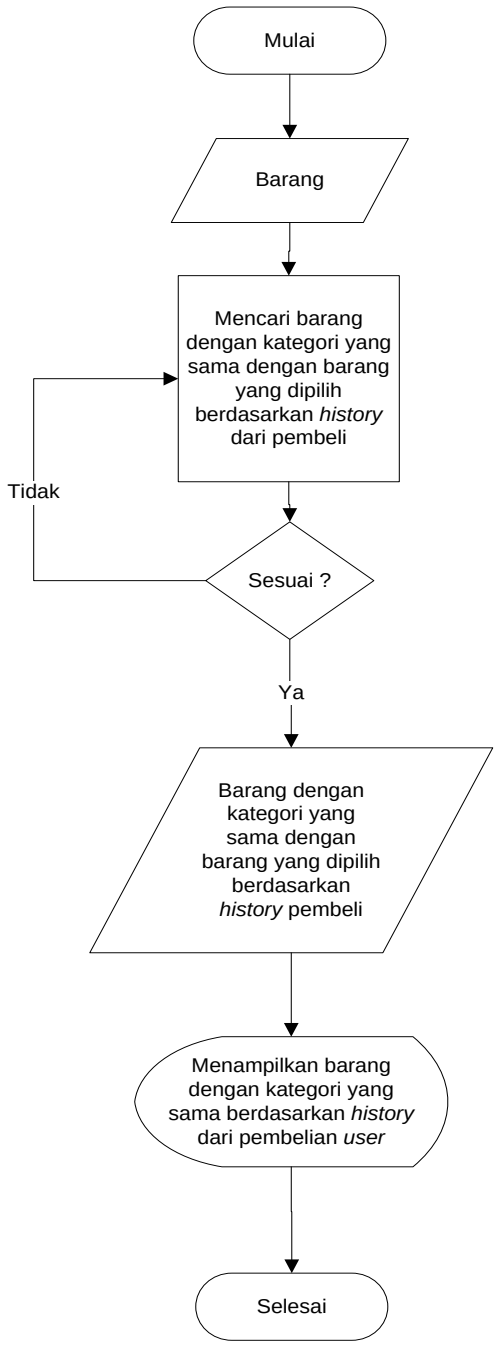
MAE = Nilai rata-rata kesalahan hitung

N = Jumlah *item* yang dihitung

p_i = Nilai prediksi *item* ke-i

q_i = Nilai *rating* sebenarnya *item* ke-i

Semakin rendah nilai MAE yang didapat, maka sistem semakin akurat dalam memprediksi *rating* untuk setiap *user*.



Gambar 2.9*FlowChart Proses Sistem Rekomendasi*