BAB 2
TINJAUAN PUSTAKA

2.1 Profil Perusahaan

2.1.1 Sejarah Perusahaan

Ocean Seven berdiri sejak Tanggal 18 Mei 2009, *Ocean Seven* beralamat di Jl.Puri Cipageran Blok E No.88 Cimahi, ide didirikannya perusahaan ini adalah melihat daya beli masyarakat Indonesia terhadap pakaian berkualitas dengan harga yang terjangkau dan berbagai tema desain yang sangat dicari oleh berbagai komunitas, *Ocean Seven* menjual *Tshirt distro* dengan desain dari berbagai tema, mulai dari tema *club* sepakbola, *anime*, kartun, superhero, musik, *automotive* dan masih banyak tema desain lainnya

2.1.2 Visi dan Misi

a. Visi

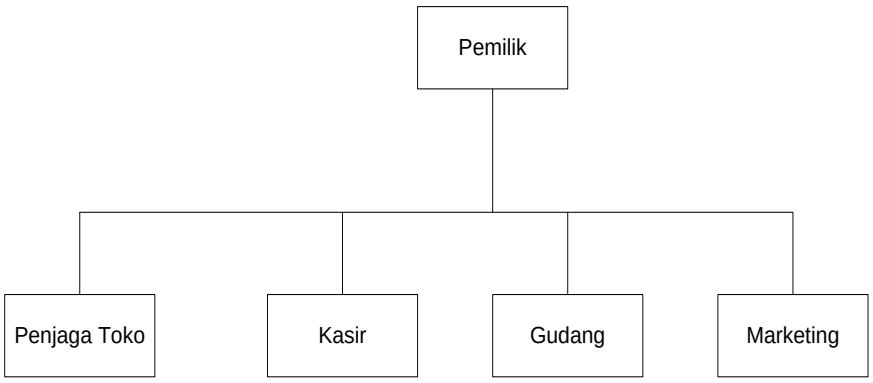
Ocean Seven menjadi *Distro Clothing* Nomor 1 di Indonesia

b. Misi

Menyediakan berbagai tema desain untuk berbagai komunitas di Indonesia dan memasarkan produk ke seluruh wilayah Indonesia

2.1.2 Struktur Organisasi

Struktur organisasi merupakan suatu susunan komponen-komponen dan hubungan antara tiap bagian serta posisi yang ada pada suatu organisasi dalam menjalankan kegiatan untuk mencapai tujuan. Struktur organisasi *Ocean Seven* dapat dilihat pada gambar dibawah ini :



Gambar 2.1 Struktur Organisasi *Ocean Seven*

2.1.4 Deskripsi Tugas :

1. Pemilik
   a. Menjalankan dan mengelola *Ocean Seven* secara keseluruhan.
   b. Mengkoordinir seluruh kinerja karyawan dalam melaksanakan tugas dan tanggung jawab pekerjaan.
   c. Melakukan pengawasan, pengarahan, dan penilaian terhadap karyawan.
2. Penjaga Toko
   a. Melayani pelanggan yang datang ke toko
   b. Memeriksa stok produk yang akan dibeli oleh pelanggan yang tersedia sesuai ukuran, gambar,warna yang diinginkan pelanggan.
   c. Mencatat stok produk yang ada di toko.
3. Bagian Gudang :
   a. Membantu Proses ke luar masuk barang.
   b. Bertanggung jawab terhadap pengadaan produk bila stok produk kosong.
   c. Bertanggung jawab terhadap penyimpanan, pemeliharaan produk dan pengiriman barang ke toko.
   d. Mencatat kartu gudang

4. Marketing

   Menyusun rencana promosi perusahaan, anggaran promosi dan melaksanakan kegiatan promosi.

5. Kasir

   a. Melakukan proses transaksi penjualan dengan pelanggan yaitu menerima pembayaran.
   b. Bertugas melayani pelanggan yang melakukan transaksi,bertanggung jawab atas hasil penjualan yaitu uang kas.
   c. Membuat laporan uang yang masuk dan uang yang keluar.

2.2 Landasan Teori

Penelitian yang dilakukan berlandaskan pada berbagai teori,teori tersebut menjadi acuan dalam pembangunan sistem. Beberapa teori yang diambil antara lain sebagai berikut:

2.2.1 Pengertian *E-commerce*

E-commerce adalah singkatan dari *elektronic commerce*, *e-commerce* atau bisa disebut perdagangan elektronik atau e-dagang adalah penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti *internet* atau televisi, *www*, atau jaringan komputer lainnya. *E-commerce* dapat melibatkan *transfer* data elektronik, pertukaran data elektronik, sistem manajemen inventori otomatis, dan sistem pengumpulan data otomatis.[4]

E-Commerce adalah suatu proses membeli dan menjual produk-produk secara elektronik oleh konsumen dan dari perusahaan ke perusahaan dengan komputer sebagai sebagai perantara transaksi bisnis. *E-Commerce* atau yang biasa disebut juga dengan istilah *Ecom* atau *Emmerce* atau *EC* merupakan pertukaran bisnis yang rutin dengan menggunakan transmisi *Electronic Data Interchange* (EDI), *email, electronic bulletin boards*, mesin faksimili, dan *Electronic Funds Transfer* yang berkenaan dengan transaksi-transaksi belanja di *internet shopping*.[13]

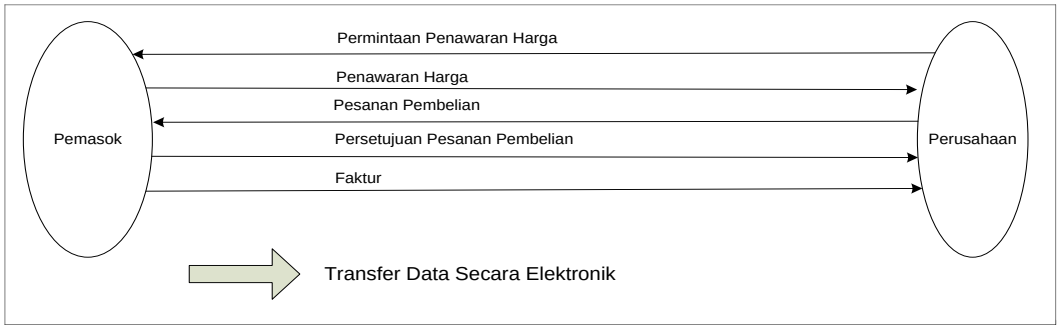
David Baum (1999, pp. 36-34) *yaitu "E-Commerce is a dynamic set of technologies, applications, and bussines process that link enterprises, consumers, and communities through electronics transactions and the electronic exchange of*

goods, services, and informations".E-Commerce merupakan suatu proses berbisnis dengan memakai teknologi elektronik, aplikasi, dan proses bisnis yang menghubungkan perusahaan, konsumen, dan komunitas tertentu melalui transaksi elektronik dan perdagangan barang, pelayanan, dan informasi yang dilakukan secara elektronik.[13]

2.2.1.1 Komponen *E-commerce*

1. *Electronic Data Interchange (EDI).*

Electronic Data Interchange (EDI) pertukaran data elektronik adalah proses *transfer* data terstruktur dengan format standar yang telah disetujui yang dilakukan dari satu sistem komputer ke sistem komputer yang lain melalui jaringan komputer dalam bentuk elektronik. Tujuan dari pemakaian teknologi *EDI*, adalah agar teknologi ini dapat membantu para pelaku bisnis. [14]



Gambar 2.2 *Electronic Data Interchange*

Manfaat *Electronic Data Interchange (EDI)* :

1. Mengurangi kesalahan
2. Mengurangi biaya
3. Meningkatkan efisiensi operasional
4. Meningkatkan kemampuan bersaing
5. Meningkatkan hubungan dengan mitra dagang
6. Meningkatkan pelayanan pelanggan

2. Digital *Currency*

   Digital *currency* dimaksudkan untuk memungkinkan user untuk memindahkan dana secara elektronik dalam lingkungan kerja tertentu. Saat ini, digital *currency* dirancang untuk versi elektronik dari uang kertas, memiliki atribut yang sama dengan media fisik sebenarnya baik secara anatomis maupun dari segi likuiditasnya.

3. Electronic *Catalogs.*

   Electronic Catalogs (*e-catalogs*) telah berada pada aplikasi komersil yang dirancang untuk *internet* dan merupakan komponen utama dari sistem *e-commerce*. *E-catalogs* merupakan antar muka grafis (*Graphical User Interface*) yang umumnya berbentuk halaman *Word Wide Web* dimana menyediakan informasi tentang penawaran produk dan jasa. *E-catalogs* umumnya mendukung *online shopping* dan kemampuan pemesanan dan pembayaran barang. Suatu *website* bisa juga merupakan suatu koleksi *catalog*, misalnya electronik mall merupakan suatu *catalog* atas *catalog.*

4. *Intranets dan Extranets*

   Umumnya *intranet* digambarkan hanya sebagai *web server* di dalam perusahaan (internal), padahal sebenarnya *intranet* adalah kumpulan *website* yang dimiliki oleh suatu kelompok (biasanya perusahaan) yang bisa diakses hanya oleh anggota kelompok tersebut. Sedangkan *extranet* merupakan area tertentu dari *intarnet* yang bisa diakses oleh kelompok di luar anggota kelompok *intranet*, tapi dengan otorisasi tertentu. Fitur *intranet* standar dalam suatu organisasi memiliki 4 kemamapuan dasar *e-mail, on line publishing, on line searches dan application distribution*.[14]

2.2.1.2 Sejarah *E-commerce*

Istilah *e-commerce* adalah perdagangan elektronik berarti pemanfaatan transaksi komersial, seperti penggunaan *EDI* untuk mengirim dokumen komersial seperti pesanan pembelian atau *invoice* secara elektronik, kemudian dia berkembang menjadi suatu aktivitas yang mempunya istilah yang lebih tepat "perdagangan *web*" pembelian barang dan jasa melalui *World Wide Web* melalui

server aman (HTTPS), protokol *server* khusus yang menggunakan enkripsi untuk merahasiakan data penting pelanggan (12).

2.2.1.3 Kelebihan *E-commerce*

E-commerce memiliki beberapa manfaat, baik itu organisasi, perusahaan dan masyarakat itu sendiri, berikut beberapa manfaat dari *e-commerce* [11]

1. Menggantikan konsep manual sehingga mempercepat pelayanan ke pelanggan.
2. Menurunkan biaya operasional (*operating cost*), mengurangi biaya yang berhubungan dengan kertas, seperti biaya percetakan, laporan dan sebagainya.
3. Meningkatkan efisiensi proses pemasaran secara global
4. Mempermudah pertukan data dan informasi menggunakan *transfer* elektronik.

2.2.1.4 Kelemahan *E-commerce*

E-commerce menawarkan keuntungan jangka pendek dan jangka panjang kepada perusahaan. *E-commerce* membuka pasar baru bagi produk atau jasa yang ditawarkan, mencapai konsumen baru, mempermudah perusahaan melakukan bisnis. Perbedaan antara proses perdagan secara manual denggan *e-commerce*, pada proses *e-commerce* terjadi efisiensi pada penggunaan *fax*, pencetakan dokumen, *entry* ulang dokumen, serta jasa kurir. Efisiensi tersebut akan menunjukkan pengurangan biaya dan waktu kecepatan proses.[8]

Kelemahan yang terdapat pada *E-commerce* yaitu :

1. Produk yang di jual kadang tidak sesuai dengan yang ditampilkan
2. Kurang aman dalam transaksi, karena kerahasiaan informasi seseorang tanpa melakukan penetrasi, dapat dilakukan seperti mengubah infornasi transaksi di tengah jalan atau membuat sistim s*erver* palsu yang dapat menipu banyak orang. Menghalangi seseorang dalam mengakses informasi, sumber, dan fasilitas-fasilitas lainnya.

2.2.1.5 Klasifikasi *E-commerce*

Penggolongan *e-commerce* berdasarkan sifat transaksinya antara lain:

1. *Business to Business* (B2B)

   Jenis transaksi dimana pembeli biasanya membeli dalam jumlah besar karena akan dijual kembali. Contoh penjualan grosir

2. *Business to Consumer* (B2C)

   Jenis transaksi dimana pembelinya perorangan dan tidak punya tujuan untuk menjualnya kembali biasanya semacam toko *online* yang menjual berbagai macam barang.

3. *Consumer to Consumer* (C2C)

   Jenis transaksi dimana pembelinya perorangan yang tidak mempunyai tujuan untuk dijual kembali dan penjualnya juga perorangan yang tidak menyediakan bermacam-macam barang melainkan hanya beberapa barang saja.

   Contoh: *advertising online*

4. *Consumer to Business* (C2B)

   Termasuk kedalam kategori ini adalah perseorangan yang menjual produk atau layanan kepada organisasi, dan perseorangan yang mencari penjual, berinteraksi dengan mereka dan menyepakati suatu transaksi .[12]

2.2.2 Pengertian *Internet*

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

Node bisa berupa komputer, jaringan lokal atau peralatan komunikasi, sedangkan garis penghubung antar simpul disebut tulang punggung (*back-bone*) yaitu media *terestrial* (kabel, serat optik, *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 salah satu yang 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 protokol 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. 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 badan Ilmu Pengetahuan Nasional Amerika Serikat.

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 protokol *standar* dalam berkomunikasi yaitu *protocol TCP/IP*. *Internet* berfungsi sebagai media komunikasi dan informasi *modern*.[3]

2.2.3 *Email*

Email atau elektronik *mail* adalah media surat menyurat secara *online*, *email* memungkinkan kita untuk berkirim surat melalui *internet* dengan waktu yang cepat. Dalam perkembangannya, *email* berkembang dari sekedar teks ke media yang bisa mengirimkan berbagai media seperti *file* dokumen dan gambar. *Email* bisa di akses dengan jasa *mail service* seperti *Yahoo* dan *Google*, alamat *email* biasanya akan memakai tanda @.**[**6**]**

2.2.4 *World Wide Web (WWW)*

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 dinikmati 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 *HTTP*.[6]

2.2.5 Secure Soket Layer (SSL)

Secure Soket Layer (*SSL*) adalah protokol yang digunakan untuk *browsing web* secara aman, *Secure soket layer* (*SSL*) adalah *protocol* yang mengamankan komunikasi antara *client* dan *server*, *Secure soket layer* (SSL) dikembangkan oleh *Netscape Communications* pada tahun 1994, *SSL* bekerja diantara protokol komunikasi *TCP/IP* di*internet* dan perangkat lunak aplikasi.[12]

2.2.6 *Web Browser*

Dalam dunia *web* perangkat lunak *client*, yaitu *browser web* mempunyai tugas yang sama yaitu menterjemahkan informasi yang diterima oleh *server web* dan menampilkannya pada layar komputer pengguna, oleh karena HTTP

memungkinkan *server web* mengirimkan beragam data, seperti teks atau gambar, *browser* harus bisa mengenali berbagai macam data yang akan diterimanya, dan selanjutnya harus tahu cara untuk menampilkan dengan benar.

Teks ditampilkan sebagai teks dan gambar ditampilkan sebagai gambar. *browser web* menerima data dalam bentuk *HTML*. *File HTML* sebenarnya adalah *file* teks biasa yang selain berisi informasi yang hendak ditampilkan kepada pengguna, juga mempunyai perintah-perintah untuk mengatur tampilan data tersebut. *Browser* yang menterjemahkan perintah-perintah.Telah dibuat *consensus* untuk menstandarkan format dan elemen-elemen *HTML*, setiap jenis *browser* bisa menterjemahkan *file HTML* secara berbeda. Beberapa *server web* memiliki *feature* seperti *server side programming*, *security control* dan lain sebagainya. *Web browser* yang bisa digunakan untuk mengakses web, antaraa lain *internet explorer, mozilla firefox, opera, safari,* dan lain-lain [9]

2.2.7 *Web Server*

Menurut Minoli berpendapat bahwa sebuah *web server* adalah sesuatu program untuk menawarkan pelayanan yang bisa diperoleh seluruh jaringan.*Web server* merupakan suatu tipe server khusus yang dapat berkomunikasi langsung dengan client menggunakan HTTP, *web server* menerima permintaan dari *client* dan meresponnya, biasanya dengan mengembalikan sebuah dokumen atau gambar [7].

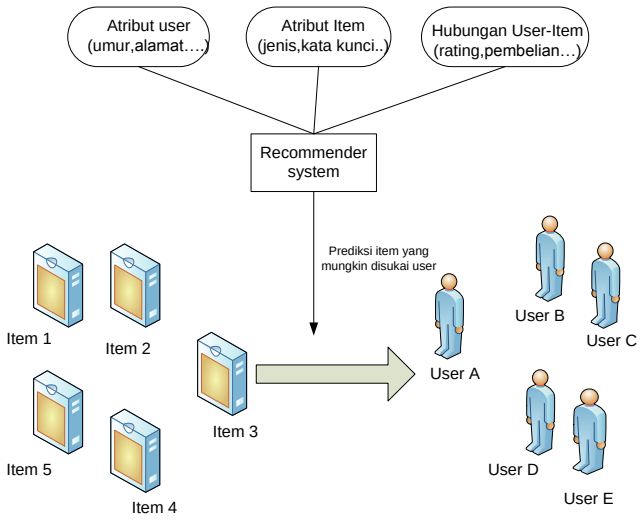
2.2.8 *Paypal*

PayPal adalah salah satu alat pembayaran (*Payment procesors*) menggunakan *internet* yang terbanyak digunakan didunia. *Paypal* mengatasi kekurangan dalam pengiriman uang tradisional seperti Cek atau *Money order* yang prosesnya dapat memakan waktu *Paypal* seperti rekening bank, pertama *member* membuat *account*, lalu mengisi *account* tersebut dengan dana dari kartu kredit atau transferan dana yang dapat diterima *paypal* setelah itu *paypal* dapat digunakan untuk bertransaksi.Berikut adalah kartu kredit di Indonesia yang sudah dicoba dan diterima oleh *Paypal*: *HSBC Visa*, BNI Visa, Mandiri *Visa*, *Citibank Mastercard*, *BCA Mastercard*, BRI *Mastercard*.[8]

2.2.9 … (*Smart Recommendation System*)

2.2.9.1 *Smart Recommendation System*

Smart Recommender system merupakan sebuah metoda untuk menampilkan informasi mengenai suatu hal (film, musik, buku, berita, gambar, dan sebagainya) yang sesuai dengan minat *user*. *Recommender system* akan membandingkan profil *user* dengan referensi yang dimilikinya lalu menampilkan informasi kepada *user* berdasarkan prediksi yang dilakukan sebelumnya. [16]



Gambar 2.3 *Recommendation System*.[16]

Ada dua tipe pengumpulan data untuk membangun *recommender system* sebenarnya. [16]

1. Secara eksplisit
   a. Meminta *user* untuk merating sebuah *item*.
   b. Meminta *user* untuk merating sekumpulan *item*.
   c. Meminta *user* untuk memilih salah satu *item* dari beberapa *item* yang diberikan.
   d. Meminta *user* untuk membuat daftar *item* yang dia suka.
2. Secara implisit
   a. Mengobservasi *item* yang sedang dilihat oleh *user* secara *online.*

b. Menganalisis jumlah *user* yang melihat suatu *item*.

c. Menyimpan catatan pembelian *user*.

Ada beberapa cara untuk menyajikan rekomendasi, yaitu:

1. Rekomendasi *Non-Personalized*

   Rekomendasi *non-personalized* akan merekomendasikan item yang memiliki tingkat popularitas yang tinggi berdasarkan rating *user* lain atau data transaksi. Contoh *output* dari teknik *non-personalized* misalnya "*20-most popular software"*

2. Rekomendasi *Demographic*

   Rekomendasi demographic memanfaatkan fitur/atribut *user*. Teknik ini mencari *user-user* yang memiliki fitur yang mirip dan merekomendasikan *item* yang disukai satu *user* kepada *user* lain yang fiturnya mirip.

3. *Rekomendasi Content-Based*

   Kebalikannya dengan rekomendasi *demographic,* rekomendasi *content-based*

   memanfaatkan fitur dari *item*. Teknik ini akan mencari kemiripin dari setiap

   item. Jika *user* X memilih *item* A dan *item* A mirip dengan *item* N, maka sistem akan merekomendasikan item N kepada *user* X.

4. *Rekomendasi User-Based Collaborative*

   Rekomendasi *user-based collaborative* memanfaatkan rating *user* atau data transaksi. Sistem mencari *user-user* yang memiliki korelasi yang tinggi kemudian merekomendasikan *item-item* yang disukai oleh *user-user* itu. Misalnya *user* X menyukai *item* A, *item* B, dan *item* C sementara *user* Y menyukai *item* B, *item* C dan *item* D. Maka sistem akan merekomendasikan *item* D pada user X dan *item* A pada *user* Y.

5. Rekomendasi *Item-Based Collaborative*

   Mirip seperti rekomendasi *user-based collaborative*, rekomendasi *item-based collaborative* memanfaatkan rating *user* data transaksi. yang membedakan adalah korelasi yang dicari. Rekomendasi *item-based collaborative* mencari korelasi diantara *item-item* yang dipilih *user*

kemudian merekomendasikan *item-item* yang berkolerasi itu pada *user* yang lain.

2.2.9.2 Konsep Dasar *Collaborative Filtering*

Collaborative filtering merupakan proses penyaringan atau pengevaluasian *item* menggunakan opini pelanggan lain.[16] *Collaborative filtering* melakukan penyaringan data berdasarkan kemiripan karakteristik pelanggan sehingga mampu memberikan informasi yang baru kepada pelanggan karena sistem memberikan informasi berdasarkan pola satu kelompok pelanggan yang hampir sama. Perbedaan minat pada beberapa anggota kelompok menjadikan sumber informasi baru yang mungkin bermanfaat bagi anggota kelompok lainnya. Secara umum proses pemberian rekomendasi terdiri atas tiga langkah, yaitu: penemuan *similar user*, pembuatan ketetanggaan (*neighborhood*), dan penghitungan prediksi berdasarkan tetangga yang dipilih. *Collaborative filtering* menghasilkan prediksi atau rekomendasi bagi pengguna atau pelanggan yang dituju terhadap satu *item* atau lebih. *Item* dapat terdiri atas apa saja yang dapat disediakan manusia seperti misalnya buku, film, seni, artikel, atau tujuan wisata. Rating dalam *collaborative filtering* dapat berbentuk dari model-model sebagai berikut:

a. Model rating skalar yang terdiri atas rating numerik seperti 1 (satu) sampai 5 (Lima)
b. Model rating *biner* dengan memilih antara setuju atau tidak setuju, atau dapat pula baik atau buruk
c. *Mating unary* dapat mengindikasikan bahwa pengguna telah mengobservasi atau membeli *item* atau me-rating *item* dengan positif.

Tidak tersedianya rating mengindikasikan tidak terdapat informasi yang menghubungkan pengguna dengan *item*. Rating dapat dikumpulkan secara eksplisit, implisit, ataupun gabungan antara eksplisit dan implisit. Rating eksplisit yaitu rating yang didapatkan pada saat pelanggan/pengguna diminta menyediakan opini terhadap *item* tertentu. Rating implisit yaitu rating yang didapatkan melalui aksi yang dilakukan pelanggan. Analisis yang dipakai untuk memperoleh rekomendasi yaitu memakai algoritma yang hanya dapat menghitung rata-rata

rating dari setiap produk. Berikut adalah rumus perhitungan rata-rata rating dari setiap produk.

$$Rproduk = \frac{\sum R}{n} \quad (2.1)$$

Keterangan :

Rproduk = nilai rata-rata *rating* produk yang dihasilkan dari perhitungan.

∑R = Perhitungan rata-rata *rating*

n = banyaknya pelanggan yang me*rating*.

2.2.9.3 Algoritma *Collaborative Filtering*

Schafer membagi algoritma *collaborative filtering* ke dalam dua kelas yang berbeda 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*. [16]

2.2.9.4 *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*. [16]

2.2.9.5 *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.[16] 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-based collaborative 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.

2.2.9.6 *Item-Based Collaborative Filtering*

Item-based collaborative filtering memanfaatkan rating *user* atau data transaksi untuk membuat rekomendasi. Teknik ini akan mencari korelasi diantara *item-item* yang dipilih *user* kemudian merekomendasikan *item-item* yang berkolerasi itu pada *user* yang lain. Pada awalnya, *item-based collaborative filtering* akan menghitung nilai kemiripan antara *item* yang satu dengan *item* yang lainnya berdasarkan rating yang diberikan oleh *user*. Nilai kemiripan antara dua *item* itu didapat dengan menghitung rating kedua *item* tersebut menggunakan rumus *Pearson Correlation atau Adjusted-Cosine.* [16]

Persamaan *Pearson Correlation atau Adjusted-Cosine*

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_u)^2}} \quad [2.2]$$

Keterangan :

S(i,j) = Nilai kemiripan antara *item* i dengan *item* j.

u ϵ U = Himpunan *user* yang me-rating baik *item* i maupun *item* j.

Ru,i = Ratinguser u pada *item* i.

$\bar{R}i$ = Nilai rating rata-rata *item* i.

$\bar{R}u,j$ = Rating user u pada *item* j.

j = Nilai rating rata-rata *item* j.

$\bar{r}_u$ = Nilai rating rata-rata *user* u.

Nilai yang dihasilkan oleh rumus adalah antara -1. hingga +1. Jika nilai koefisien semakin mendekati -1 atau +1, maka hubungan antara kedua variabel itu akan semakin kuat. Jika nilai koefisiennya adalah 0, maka kedua variabel itu tidak ada hubungannya (*independen*). Pada kasus *collaborative filtering* nilai koefisien lebih populer disebut *similarity* (kemiripan). Jika nilai *similarity* antara kedua *item* mendekati +1, maka kedua *item* akan semakin mirip satu sama lain. Sebaliknya, jika mendekati -1, kedua *item* itu akan semakin bertolak belakang. Tahap berikutnya adalah menghitung prediksi. Tahapan ini dilakukan untuk memperkirakan rating yang akan diberikan oleh seorang *user* pada suatu *item* yang belum pernah di-rate oleh *user* itu. Penghitungan prediksi menggunakan rumus *weighted sum.*

Persamaan *weighted sum*:

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

Keterangan :

P(a,j) = Prediksi rating *item* j oleh *user* u.

i ϵ I = Himpunan *item* yang mirip dengan *item* j.

Ru,i = Rating *user* u pada *item* i.

Si,j = Nilai *similarity* antara *item* i dan *item* j.

Tahap terakhir adalah pembuatan rekomendasi. Pada tahap ini, teknik *collaborative filtering* berperan untuk menyediakan nilai-nilai yang akan dijadikan bahan pokok pembuatan rekomendasi. Sistem secara keseluruhan memiliki peran yang lebih besar. Sistem dapat membuat dua jenis rekomendasi, yaitu:

1. Rekomendasi yang bersifat umum

   Pembuatan rekomendasi umum memanfaatkan nilai kemiripan yang sudah dihitung sebelumnya. Setiap *user* siapapun itu, akan mendapatkan rekomendasi yang sama
2. Rekomendasi yang bersifat khusus

Rekomendasi khusus bersifat personal. artinya, setiap *user* akan mendapatkan rekomendasi yang berbeda. Pembuatan rekomendasi memanfaatkan nilai prediksi yang telah dihitung dengan menggunakan rumus *weigthed sum*.

2.2.9.7 Karakteristik Teknik *Item Based Collaborative Filtering*

Karakteristik dari teknik *item based collaborative filtering* dari sebenarnya yaitu:

1. *Scalable*

   Di dunia nyata, teknik *item-based collaborative filtering* digunakan untuk membuat rekomendasi bagi jutaan pengguna dengan jutaan *item-item* yang tersedia. Salah satu situs yang memanfaatkan teknik ini adalah *amazon.com*.

2. *Cold Start*

   Pada tahap awal, akan sulit untuk membuat rekomendasi dengan kualitas yang baik karena sumber data yang digunakan tidak banyak.

3. *Sparsity Problem*

   Sparsity problem (masalah kekosongan data) adalah kondisi ketika lebih banyak sel yang kosong dibandingkan dengan sel yang terisi dalam suatu tabel. Teknik *item-based collaborative filtering* tidak akan berurusan dengan masalah seperti ini karena teknik ini hanya berurusan dengan sel yang terisi.

4. *Content Analysis*

   Analisis konten tidak diperlukan karena *user* akan me-*rate* suatu *item* berdasarkan pengalaman mereka. [16]

2.2.10 Kuesioner

Menurut Sugiyono menyatakan bahwa “Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya”. Kuesioner atau angket merupakan teknik pengumpulan data yang digunakan untuk mendapatkan informasi responden, dalam arti laporan tentang pendapat dari hal-hal yang diketahuinya. Kuesioner disebarkan menggunakan teknik quota sampling dimana

pemilihan sampel dengan memilih sejumlah tertentu (kuota) dan paling mudah diperoleh seperti yang dikehendaki oleh peneliti. Kuesioner disebarkan kepada 30 responden, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat mengambil kesimpulan terhadap penilaian penerapan sistem yang baru. [18]

2.2.10.1 Skala Pengukuran

2.2.10.2 Skala *Likert*

Skala *likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseoran atau sekelompok orang tentang fenomena sosila. Dalam penelitian, fenomena sosial ini telah ditetapkan secara spesifik, yang selanjutnya disebut sebagai variabel penelitian. Dengan skala *likert*, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun *item-item* instrumen yang dapat berupa pernyataan atau pertanyaan. Jawaban setiap *item* instrumen yang menggunakan skala *likert* mempunyai gradiasi dari sangat positif sampai sangat negatif, yang ada berupa kata-kata antara lain:

Tabel 2.1. Gradiasi Jawaban Skala *Likert*

Keterangan	Skala
Sangat setuju	1
Setuju	2
Ragu-ragu	3
Tidak setuju	4
Sangat tidak setuju	5

Instrumen penelitian yang menggunakan skala *likert* dapat dibuat dalam bentuk *checklist* ataupun pilihan ganda. Setelah itu langkah berikutnya adalah

1. Pertama kita menentukan jumlah skor jawaban responden terhadap tiap soal. Dengan menggunakan rumus

$$P = \sum_{i=1}^{5} Si \times Ri \quad (2.4)$$

2. Kemudian untuk mencari prosentasi dari masing-masing jawaban kuesioner digunakan rumus skala *likert* sebagai berikut:

$$Y = \frac{S}{\text{skorideal}} \times 100 \quad \ldots.(2.5)$$

Keterangan:

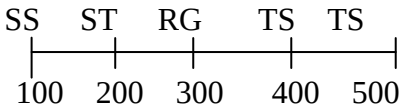
Y = Nilai prosentasi yang dicari.

S = Total yang diperoleh $S = \sum_{i=1}^{5} ni.ri$

ri = Banyak responden yang menjawab jawaban ke-i.

ni = Nilai jawaban ke-i.

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel atau responden. Dibawah ini merupakan skala untuk mengmbil kesimpulan dari setiap petanyaan, datap dilihat pada Tabel 2.4 [15]



Gambar 2.4 Skala *Likert.* [15]

2.2.11 *Message Digest Algorithm 5 (MD5*)

2.2.11.1 Prinsip Dasar Kriptografi

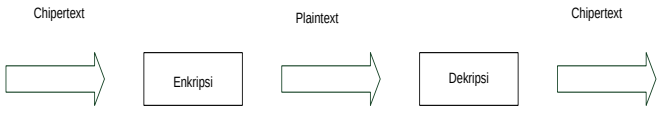
Kriptografi (*cryptography*) merupakan ilmu dan seni untuk menjaga pesan agar aman. (*Cryptography is the art and science of keeping messages secure.*) "*Crypto*" berarti "*secret*" (rahasia) dan "*graphy*" berarti "*writing*" (tulisan). Para pelaku atau praktisi kriptografi disebut *cryptographers*. Sebuah algoritma kriptografik (*cryptographic algorithm*), *disebutcipher*, merupakan persamaan matematik yang digunakan untuk proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (untuk enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat.[15]

Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut *plaintext*) menjadi pesan yang tersembunyi (disebut *ciphertext*) adalah enkripsi

(*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah "*encipher*". Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext*, disebut dekripsi (*decryption*).

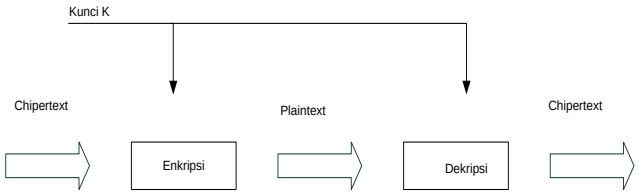
Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah "*decipher*". *Cryptanalysis* adalah seni dan ilmu untuk memecahkan ciphertext tanpa bantuan kunci. *Cryptanalyst* adalah pelaku atau praktisi yang menjalankan *cryptanalysis. Cryptology* merupakan gabungan dari *cryptography* dan *cryptanalysis*.

Ilmu kriptografi adalah ilmu yang mempelajari tentang penyembunyian huruf atau tulisan sehingga membuat tulisan tersebut tidak dapat dibaca oleh orang yang tidak berkepentingan. Kriptografi mempunyai 2 (dua) bagian yang penting, yaitu enkripsi dan dekripsi. Enkripsi adalah proses dari penyandian pesan asli menjadi pesan yang tidak dapat diartikan seperti aslinya. Dekripsi sendiri berarti merubah pesan yang sudah disandikan menjadi pesan aslinya. Pesan asli biasanya disebut *plaintext*, sedangkan pesan yang sudah disandikan disebut *ciphertext*. Pada Gambar 2.5 dapat dilihat bahwa masukan berupa *plaintext* akan masuk ke dalam blok enkripsi dan keluarannya akan berupa *ciphertext*, kemudian *ciphertext* akan masuk ke dalam blok dekripsi dan keluarannya akan kembali menjadi *plaintext* semula.



Gambar 2.5 Proses Enkripsi dan Dekripsi

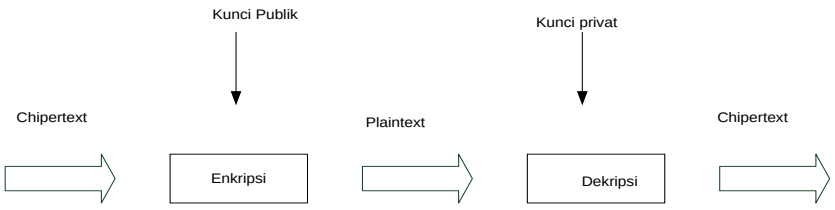
Ada 2 (dua) model algoritma enkripsi yang menggunakan kunci, yaitu kunci simetrik dan kunci asimetrik. Enkripsi kunci simetrikyang biasanya disebut enkripsi konvensional adalah enkripsi yang menggunakan kunci yang sama untuk enkripsi maupun dekripsi, dari Gambar 2.6 terlihat bahwa untuk mengenkripsi maupun mendekripsi pesan hanya menggunakan satu buah kunci (K) saja.



Gambar 2.6 Enkripsi-dekripsi Kunci Simetrik

Penggunaan metode ini membutuhkan persetujuan antara pengirim dan penerima tentang kunci sebelum mereka saling mengirim pesan. Keamanan dari kunci simetrik tergantung pada kerahasiaan kunci, apabila seorang penyusup dapat menemukan kunci maka dengan mudah dapat membaca pesan yang sudah dienkripsi.

Enkripsi kunci simetrik dapat dibagi kedalam 2 (dua) kelompok yaitu metode *stream cipher* dan metode *block cipher*. Enkripsi kunci asimetrik (biasa disebut enkripsi kunci publik) dibuat sedemikian rupa sehingga kunci yang dipakai untuk enkripsi berbeda dengan kunci yang dipakai untuk dekripsi. Enkripsi kunci publik disebut demikian karena kunci untuk enkripsi boleh disebar luaskan kepada umum sedangkan kunci untuk mendekripsi hanya disimpan oleh yang bersangkutan. Contohnya seperti pada Gambar 2.7 bila seseorang ingin mengirim pesan kepada orang lain maka orang tersebut menggunakan kunci publik orang tersebut untuk mengenkripsi pesan yang dikirim kepadanya lalu orang tersebut akan mendekripsi pesan tersebut dengan kunci privat yang dimilikinya.



Gambar 2.7 Enkripsi Kunci Asimetrik

keamanan data saja, tetapi sekarang semakin banyak tujuan-tujuan yang akan dicapai yaitu:

2.2.11.2 Metode Fungsi *Hash*

Salah satu cara untuk menguji integritas sebuah data adalah dengan memberikan "*checksum*" atau tanda bahwa data tersebut tidak berubah. Cara yang paling mudah dilakukan adalah dengan menjumlahkan karakter-karakter atau data-data yang ada sehingga apabila terjadi perubahan, hasil penjumlahan menjadi berbeda. Cara ini tentunya mudah dipecahkan dengan menggunakan kombinasi data yang berbeda akan tetapi menghasilkan hasil penjumlahan yang sama. Pada sistem digital biasanya ada beberapa mekanisme pengujian integritas seperti antara lain:

a. *Parity checking*
b. *Checksum*
c. *Hash function*

Fungsi *Hash* (*Hash function*) merupakan fungsi yang bersifat satu arah dimana jika kita masukkan data, maka dia akan menghasilkan sebuah "*checksum*" atau "*fingerprint*" dari data tersebut. Sebuah pesan yang dilewatkan ke fungsi hash akan menghasilkan keluaran yang disebut *Message Authenticated Code* (MAC). Dilihat dari sisi matematik, *hash function* memetakan satu set data ke dalam sebuah *set* yang lebih kecil dan terbatas ukurannya. Ada beberapa fungsi *hash* yang umum digunakan saat ini antara lain:

a. MD5 (*Message-Digest Algorithm 5*)
b. SHA (*Secure Hash Algorithm*).

1. MD5 (*Message-Digest Algorithm 5*)

Salah satu dari bagian kriptografi adalah fungsi *hash* satu arah. Fungsi *hash* satu arah adalah dimana dapat dengan mudah melakukan enkripsi untuk mendapatkan *cipher*-nya tetapi sangat sulit untuk mendapatkan *plaintext*-nya. Salah satu fungsi hash yang paling banyak digunakan adalah *Message Digest Algorithm* 5 (MD-5). *MD-5* merupakan fungsi hash satu arah. *MD-5* adalah salah

satu aplikasi yang digunakan untuk mengetahui bahwa pesan yang dikirim tidak ada perubahan sewaktu berada di jaringan.

Algoritma *MD-5* secara garis besar adalah mengambil pesan yang mempunyai panjang variabel diubah menjadi ‘sidik jari’ atau ‘intisari pesan’ yang mempunyai panjang tetap yaitu 128 bit.‘Sidik jari’ ini tidak dapat dibalik untuk mendapatkan pesan, dengan kata lain tidak ada orang yang dapat melihat pesan dari ‘sidik jari’ *MD-5.Message digest* atau intisari pesan harus mempunyai tiga sifat penting, yaitu:

1. Bila P diketahui, maka MD(P) akan dengan mudah dapat dihitung.
2. Bila MD(P) diketahui, maka tidak mungkin menghitung P.
3. idak seorang pun dapat memberi dua pesan yang mempunyai intisari pesan yang sama. H(M) H(M’). ≠

Salah satu aplikasi yang lazim menggunakan *MD5* adalah pengamanan berkas *password (/etc/passwd atau/etc/shadow)* di sistem *UNIX*. Berkas *password* menyimpan data *password* dalam bentuk yang sudah terenkripsi dengan menggunakan *DES*. Implementasi awal dari sistem *UNIX* adalah menyimpan data *password* yang sudah terenkripsi tersebut langsung pada salah satu *field* di berkas *password*. Meskipun sudah terenkripsi, penyimpanan data *passwor*d yang sudah terenkripsi tersebut masih menimbulkan potensi lubang keamanan karena *DES* merupakan enkripsi yang *reversible*.

Panjang data yang dihasilkan oleh proses enkripsi juga bergantung kepada panjang data yang dimasukkan. Jika seseorang berhasil mendapatkan berkas *password* tersebut, maka bisa mencoba proses dekripsi yaitu dengan melakukan *brute force attack* dengan mencoba melakukan proses dekripsi, *MD5* menambahkan satu tingkat keamanan lagi. Data *password* yang disimpan bukanlah *password* yang terenkripsi saja, melainkan data yang terenkripsi yang sudah dilewatkan oleh *MD5*. Karena sifatnya yang satu arah, sangat sulit untuk mencari data *password* terenkripsi dengan basis data hasil fungsi *MD5*. Jadi skema penyimpanan data tersebut contohnya sebagai berikut:

password > DES > password terenkripsi > MD5 > hashed encrypted password

Dengan cara ini,potensi untuk melakukan serangan *brute force* terhadap *encrypted password* menjadi lebih sulit. Satu-satunya cara untuk melakukan serangan *brute force* adalah dengan melakukan enkripsi juga dengan melalui jalan maju seperti di atas dan kemudian membandingkan hasil *hashed encrypted password*nya. Jika sama persis, maka kata yang dipilih sebagai percobaan sama dengan *password* yang ingin dipecahkan tersebut. *MD5* juga digunakan dalam autentifikasi dengan menggunakan *protocol CHAP* (RFC 1994).[15]

2.3 Basis Data (*Database*)

Basis data terdiri atas 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, konsumen), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, *symbol*, teks, gambar, bumi, atau kombinasinya.

Struktur *file database*:

1. Data adalah satuan informasi yang akan diolah, dimana sebelum diolah dikumpulkan di dalam suatu *file database*. Pengumpulan data dilakukan secara sistematis menurut struktur *file database* tersebut.
2. *Record* adalah data yang isinya merupakan satu kesatuan seperti nama, alamat, nomor telepon. Setiap keterangan yang mencakup nama, alamat dan nomor telepon dinamakan satu *record*. Dan setiap *record* diberi nomor urut yang disebut nomor *record* (*Record Number*). Ukuran suatu *file database* ditentukan oleh jumlah *record* yang tersimpan di dalamnya.
3. *Field* adalah sub bagian dari *record*. Dari contoh isi *record* di atas maka terdiri dari 3 *field*, yaitu *field* nama, *field* alamat dan *field* nomor telepon.

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (arsip) saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembalidengan cepat dan mudah.
2. Kumpulan *file/table*/arsip saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Ada beberapa tahapan yang digunakan dalam proses mendesain suatu basis data yaitu:

1. Tentukan tujuan dari basis data yang akan dibuat.
2. Tentukan tabel – tabel yang sekiranya akan dperlukan.
3. Tentukan *field* – *field* (kolom – kolom) yang diperlukan oleh tabel.
4. Tentukan sebuah kolom yang bersifat unik untuk dijadikan sebagai kunci (*key*).
5. Tentukan relasi antar tabel.
6. Periksa ulang dan sempurnakan desain basis data.
7. Isi datadalam basis data dan buat objek – objek *database* yang sekiranya diperlukan**.[**10**]**

2.3.1 Sistem Basis Data (*Database System*)

Sistem *database* adalah sebuah sistem yang terdiri atas kumpulan *file* (tabel) yang saling berelasi atau berhubungan dan sekumpulan program (*DBMS*) yang memungkinkan beberapa pemakai atau program lain untuk mengakses dan memanipulasi *file-file* tersebut. Komponen utama Sistem Basis data yaitu:

1. Data yang disimpan dalam basis data

   Yaitu masukan yang dibutuhkan oleh sistem *database* untuk diolah dan dimodifikasi menjadi informasi yang berguna.

2. *Hardware :storage, processor, memory*

   Adalah peralatan fisik *database* untuk menyimpan dan mengolah data menjadi informasi.

1. *Software : DBMS, Report-writer, design, arts, dll*

   Diantara *database physical* dan *end user* terdapat *software database* manajemen (*DBMS*) yang mengatur permintaan akses ke dalam sistem. *Database* manajemen sistem (*DBMS*) adalah kumpulan *file* yang saling berkaitan bersama program pengelolanya.

2. Adalah pemakai yang berinteraksi dengan sistem *database* melalui pemanggilan satu program aplikasi permanen.

2.3.2 Bahasa *Database* (*Database Language*)

DBMS merupakan perantara bagi pemakai dengan *database* dalam disk. Cara berinteraksi antara pemakai dengan *database* tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa tersebut dapat disebut bahasa *database* yang terdiri dari sejumlah perintah (*statement*) yang diformulasikan dan dapat diberikan *user* dan dikenali atau diproses oleh *DBMS* untuk melakukan suatu aksi atau pekerjaan tertentu. Bahasa *database* dapat dipilih ke dalam 2 (dua) bentuk yaitu :

1. *Data Definition Language* (DDL).

   Struktur atau skema *database* yang menggambarkan desain *database* secara keseluruhan. Dengan bahasa inilah dapat dibuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel, dan sebagainya.

2. *Data Manipulation Language* (DML).

   Merupakan bentuk bahasa *database* yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu *database*. Manipulasi data dapat berupa penyisipan atau penambahan data baru ke suatu basis data, penghapusan data, dan pengubahan data di suatu *database* [10].

2.3.3 *Structure Query Language (SQL)*

SQl adalah bahasa standar yang digunakan oleh *Database Management system* untuk memanipulasi dan memperoleh data dari sebuah *database* relasional. Perintah *SQL* :

1. Data *definition Language (DDL*) yaitu perintah untuk mendifinisikan kerangka basis data, perintah tersebut adalah :
   a. *Create* ,Untuk membuat atau menciptakan objek basis data.
   b. *Alter*, Untuk memodifikasi atau mengahpus objek basis data
   c. *Drop*,Untuk menghapus objek basis data
2. Data *Manipulation Language* (*DML*), digunakan untuk mengoperasikan atau memanipulasi isi basis data yaitu:
   a. *Select*, digunakan untuk mengambil data dari basis data
   b. *Delete*, digunakan untuk menghapus data pada basis data

c. *Insert*, digunakan untuk menambahkan data kedalam table
d. *Update*, digunakan untuk modifikasi data pada basis data
e. *Security*, *Security* perintah yang digunakan untuk menjamin keamanan data yaitu: *Grant*, digunakan untuk memberikan akses kepada *user* tertentu ke basis data dan *Revoke*, digunakan untuk mencabut hak akses dari *user*.

2.3.4 *Struktur Dasar SQL*

Ekpresi *SQL* dasar sebenarnya hany terdiri atas 3 klausa, yaitu: *Select*, *From* dan *Where.*

1. Klausa select digunakan untuk menetapkan daftar atribut (*field*) yang diinginkan sebagai hasil *query*.
2. Klausa *from* digunakan untuk menetapkan tabel yang akan ditelusuri selama *query* data dilakukan.
3. Klausa *where*, yang sifatnya opsional, digunakan sebagai predikat (kriteria) yang harus dipenuhi dalam memperoleh hasil *query*.

Sintaks (cara penulisan) dari ekpresi *SQL* dasar dengan 3 klausa tersebut adalah :

Select A1 [, A2,..,An]

From T1 [, T2...T2]

[*Where* P]

Keterangan:

1. A1, A2...An merupakan daftar atribut
2. T1, T2...,Tn merupakan daftar *table*
3. P merupakan predikat *query*

2.3.5 Konsep Dasar Analisis Sistem

Sistem diartikan sebagai sekumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu, yaitu menyajikan informasi. Elemen atau komponen tersebut secara umum adalah *hardware*, *software*, *brainware*, data, dan prosedur. Dalam mengembangkan sistem informasi dibutuhkan perangkat atau alat-alat pengembang sistem. Alat-alat yang digunakan dalam pengembangan sistem adalah *Entity Relational Diagram* (ERD), *flowmap*, diagram konteks, *Data Flow Diagram (DFD),* dan kamus data

2.3.5.1 Bagian Alur Dokumen (*Flowmap*)

Flowmap merupakan diagram alir yang menunjukan arus bagi dokumen, aliran data fisik entitas sistem informasi dan kegiatan operasi yang berhubungan dengan sistem informasi. Penggambaran biasanya diawali dengan mengamati dokumen apa yang menjadi media data atau informasi dan selanjutnya ditelusuri bagaimana dokumen termasuk ke bagian entitas mana dokumen tersebut, proses apa yang terjadi terhadap dokumen tersebut dan seterusnya.

2.3.5.2 *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*. [2]. *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 informasinnya 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.

2.3.5.3 Diagram Konteks

Diagram konteks merupakan alat pemodelan atau suatu diagram yang menggambarkan sistem berbasis komputer yang dirancang secara global dan merupakan suatu diagram alir data tingkat atas, dimana didalam diagram konteks ini menggambarkan seluruh jaringan, baik masukan maupun sebuah keluaran sebuah sistem.

Diagram konteks terdiri dari sebuah simbol proses tunggal yang menggambarkan sebuah sistem dan menunjukan data aliran utama untuk dan dari terminator. Diagram ini merupakan dasar yang digunakan untuk menentukan aliran data yang mengalir menuju sistem (*input system*) dan keluar dari sistem (*output system*), yang meliputi objek berupa kesatuan luar (*eksternal entity*). Diagram konteks dapat mendefinisikan jangkauan proses penurunan sistem informasi yaitu menentukan apa yang menjadi bagian dari sistem informasi dan apa yang tidak menjadi bagian sistem informasi.

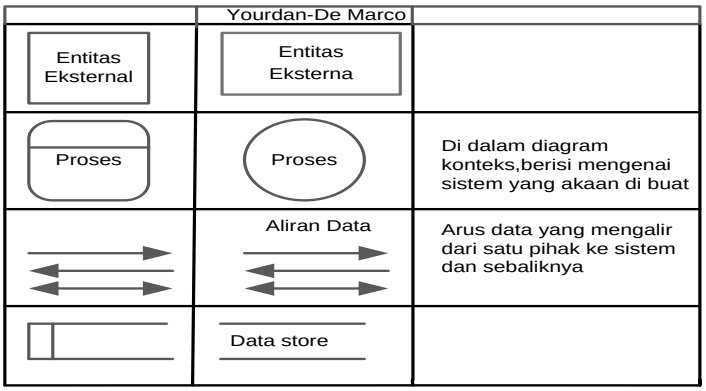


Gambar 2. 8 Komponen Diagram Konteks

2.3.5.4 *Data Flow Diagram* (DFD)

Data flow Diagram (DFD) adalah diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari sistem. DFD sering digunakan untuk menggambarkan sustu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telepon, surat, dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya *file* kartu, *harddisk*, *tape*, disket, dan lain sebagianya).

Komponen-komponen DFD yang digunakan dapat dilihat pada gambar berikut ini, yaitu:



Gambar 2.9 Komponen *DFD*

1. *External entity* (kesatuan Luar) atau *boundary* (batas sistem)

   Setiap sistem pasti memiliki batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar (*external entity*)

merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainya yang berada di lingkungan luarnya yang memberikan *input* atau menerima *output* dari *sistem*.

2. *Dataflow* (arus data)

   Arus data di DFD diberi simbol panah. Arus data ini mengalir diantara proses, simpanan, dan kesatuan luar.

3. *Process* (proses)

   Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.

4. Data *store* (simpanan data)

   Merupakan simpanan dari data yang dapat berupa suatu *file* atau *database* di komputer, suatu arsip atau catatan manual dan lain sebagainya.[2]

2.3.5.5 Kamus Data (Data *Dictionary*)

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. Kamus data dapat digunakan pada saat analisis sitem atau perancangan sistem, kamus data digunakan untuk mencatat terminologi bisnis, aturan standar (batasan panjang karakter, nilai, *system field*). [2]

2.4 *Software* yang digunakan

2.4.1 *Personal Home Page* (*PHP*)

PHP dibuat oleh *Rasmus Lerdorf* pada tahun 1994, pada awalnya tidak untuk didistribusikan dan hanya digunakan pada *homepage* pribadinya. Pada tahun 1995 dikeluarkan versi pertama yang dapat digunakan oleh umum dengan nama *personal home page tools*. Ditulis kembali pada pertengahan 1995 dan diberi nama sebagai *PHP/F1 version* 2. F1 berasal dari paket *Rasmus* yang mana merupakan html *interpreter* untuk data *form*. Pada hasil kombinasi tersebut juga

ditambah dukungan SQL.PHP/F1 terus berkembang dan banyak orang mulai memberikan kontribusi dalam pengembangannya.

Pada tahun 1996 *PHP/F1* diperkirakan telah digunakan 15.000 situs *web* dunia, dan pada pertengahan 1997 jumlah ini berkembang melebihi 50.000.pada pertengahan 1997 juga terjadi perubahan pada *PHP* dimana berubah menjadi proyek yang didukung oleh team yang lebih terorganisasi. Parsernya ditulis ulang oleh *Zeev Suraski* dan *Andi Gutmans* dan parser baru inilah yang membentuk basis untuk *PHP* versi 3. Banyak kode *utility* telah selesai ditulis kembali**.[**12**]**

2.4.2 *Cascading Style Sheet* (CSS)

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. Begitu pula sama halnya dengan *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, subbab, *bodytext, footer, images* dan *style* lain dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya *CSS* dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa *HTML* dan *XHTML*.CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan *CSS* dapat menampilkan halaman yang sama dengan format yang berbeda.[15]

2.4.3 *MYSQL*

SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses *server database*. Semenjak tahun 70-an bahasa ini telah dikembangkan oleh *IBM*, yang kemudian diikuti dengan adanya *Oracle, Informix dan Sybase*. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan misalnya *dBase* ataupun *Clipper* yang masih menggunakan perintah – perintah pemrograman murni. *MySQL* adalah sebuah *server database* SQL *multiuser* dan *multi-threaded*]. SQL sendiri adalah salah satu bahasa *database* yang paling populer di dunia. Implementasi program *server database* ini adalah program daemon '*mysql*' dan beberapa program lain serta beberapa pustaka. Sebagaimana *database* sistem yang lain, dalam SQL juga

dikenal hierarki *server* dengan *database-database*. Tiap-tiap *database* memiliki tabel-tabel. Tiap-tiap tabel memiliki *field-field*.

Umumnya informasi tersimpan dalam table-tabel yang secara lojik merupakan struktur 2 dimensi terdiri atas baris dan kolom. *Field-field* tersebut dapat berupa data seperti *int, realm char, date, time* dan lainnya. *SQL* tidak memiliki fasilitas pemrograman yang lengkap, tidak ada *looping* ataupun percabangan. Sehingga untuk menutupi kelemahan ini perlu digabung dengan bahasa pemrograman semisal *Pascal*. *MySQL* sering digunakan sebagai *SQL server* karena berbagai kelebihannya, antara lain[5] :

1. Source *MySQL* dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.
3. Pengaksesan *database* dapat dilakukan dengan mudah.

2.4.4 *Hypertext Markup Language* (*HTML*)

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi di dalam sebuah *browser internet*. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan *SGML (Standard Generalized Markup Language)*, *HTML* adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*.*HTML* saat ini merupakan standar *internet* yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C).

HTML berupa kode-kode tag yang menginstruksikan *browser* untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah *file* yang merupakan *file HTML* dapat dibuka dengan menggunakan *browser web* seperti *Mozilla Firefox* atau *Microsoft Internet Explorer. HTML* juga dapat dikenali oleh aplikasi pembuka *email* ataupun dari *PDA* dan program lain yang memiliki kemampuan *browser*.

HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan *TAG* tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: TAMPIL TEBAL, maka penulisannya

dilakukan dengan cara: **TAMPIL TEBAL**. Tanda digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh teks yang ingin ditebalkan, dan diakhiri dengan tanda untuk menonaktifkan cetak tebal tersebut. Secara garis besar, terdapat beberapa jenis elemen dari *HTML*:

a. *Structural*. tanda yang menentukan *level* atau tingkatan dari sebuah teks (contoh,<h1>Golf</h1> akan memerintahkan *browser* untuk menampilkan "Golf" sebagai teks tebal besar yang menunjukkan sebagai *Heading 1*

b. *Presentational*. tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan *level* dari teks tersebut (contoh, boldface akan menampilkan bold. Tanda *presentational* saat ini sudah mulai digantikan oleh *CSS* dan tidak direkomendasikan untuk mengatur tampilan teks,

c. *Hypertext*. tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain. Selain markup *presentational*, markup yang lin tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan *tag HTML* untuk menentukan tampilan telah dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan *Cascading Style Sheet.*

```
<!DOCTPE html>
<html>
<head>
    <title>Informatika : Belajar HTML<title>
<head>
<body>
     <p>Belajar HTML</p>
</body>
```

2.4.5 *Apache Web Browser*

Web server merupakan *server internet* yang mampu melayani koneksi transfer data dalam *protocol HTTP*. *Web server* merupakan hal yang terpenting dari *server* di internet dibandingkan server lainnya seperti *e-mail server*, ftp *server* ataupun *news server*. Hal ini disebabkan *web server* telah dirancang untuk dapat melayani beragam jenis data, dari *text* sampai grafis 3 dimensi. Kemampuan ini telah menyebabkan berbagai institusi seperti universitas maupun perusahaan dapat menerima kehadirannya dan juga sekaligus menggunakannya sebagai sarana di *internet*.

Web server juga dapat menggabungkan dengan dunia *mobile wireless* internet atau yang sering disebut sebagai *WAP* (*Wireless Access Protocol*), yang banyak digunakan sebagai sarana *handphone* yang memiliki fitur *WAP*. Dalam kondisi ini, *webserver* tidak lagi melayani data *file HTML* tetapi telah melayani *WML* (*Wireless Markup Language*). Salah satu *software* yang biasa digunakan oleh banyak *web master* di dunia adalah apache. *Software* tersebut dapat kita *download* secara gratis dari *web* resmi *apache*, yaitu *http://www.apache.org.* Dalam Penggunaannya *Apache* merupakan *software open source* yang sekarang ini sudah merebut pasar dunia lebih dari 50%. *Web server* ini fleksibel terhadap berbagai *system* operasi seperti *windows9x/NT* ataupun *unix/linux*. *Apache* merupakan turunan dari *web server* yang dikeluarkan oleh *NCSA* yaitu *NCSA HTTP*d pada sekitar tahun 1995.

Kelebihan *web server Apache* :

a. *Freeware* (software gratisan).
b. Mudah di *install*.
c. Mampu beroperasi pada berbagai *platform* sistem operasi.
d. Mudah mengkonfigurasinya.
e. *Apache Web server*

Mudah dalam menambahkan *periferal* lainnya ke dalam *platform web* servernya, misalnya untuk: menambahkan modul, cukup hanya menset *file* konfigurasinya agar mengikutsertakan modul itu ke dalam kumpulan modul lain yang sudah dioperasikan.