BAB II

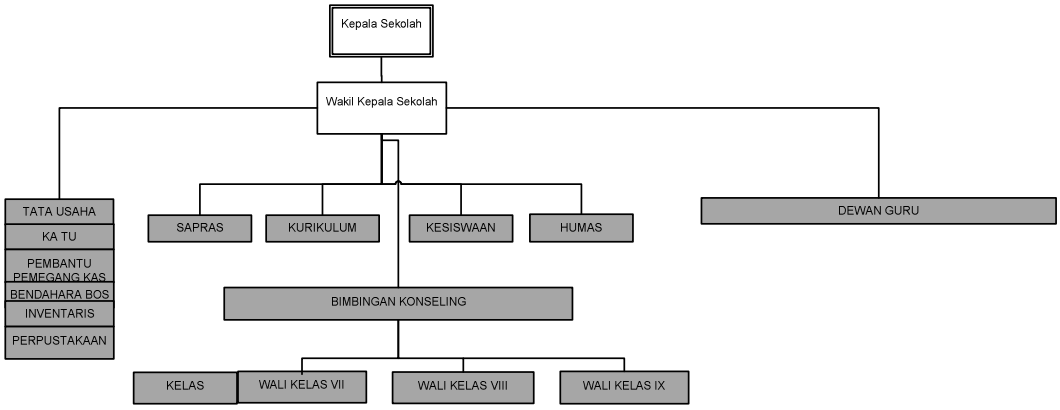
TINJAUAN PUSTAKA

2.1 Sejarah Singkat SMPN 1 Telukjambe

Awalnya SMP NEGERI 1 Telukjambe adalah SMP Yayasan Pembangunan TelukJambe (YPT) sekolah swasta yang dibangun untuk menampung lulusan SD di daerah Telukjambe Timur yang tidak mampu untuk bersekolah ke luar daerah. Sekolah ini pada mulanya berlokasi di areal Kantor Kecamatan Telukjambe Timur, dengan Kepala Sekolah Ibu Sriana Rusli istri dari Camat Telukjambe Timur yang menjabat saat itu. Namun karena kebutuhan akan sekolah lanjutan sangat mendesak, masyarakat mendorong untuk me-negeri-kan sekolah ini. Hingga pada tahun 1979 sekolah ini menempati gedung sendiri yang dibangun pemerintah di daerah pemukiman sepi di kawasan Jalan Raya Telukjambe, Desa Sukadana dan secara resmi menyandang nama SMP Negeri 1 Telukjambe dengan Kepala Sekolah Anang Busra, BA. Perjalanan yang begitu panjang telah menempa SMP ini menjadi institusi pendidikan yang mantap dan bahkan cukup disegani. Alumni sekolah ini tersebar diberbagai institusi, baik negeri maupun swasta. Dan secara traditif, sekolah ini juga terkenal melahirkan banyak olahragawan andal yang beberapa diantaranya turut mengharumkan Kabupaten Karawang. Letak sekolah ini pun sangat strategis, jauh dari kebisingan dan kesibukan. Hingga siswa dapat belajar dengan tenang.

Saat ini SMP Negeri 1 Telukjambe merupakan sekolah Potensial Mandiri yang menuju SSN.

2.1.1 Struktur Organisasi



Gambar 2.1 Struktur Organisasi SMPN 1 Telukjambe

2.1.2 Visi dan Misi

a. Visi

- Berprestasi berlandaskan Iman dan Taqwa serta Berwawasan Lingkungan

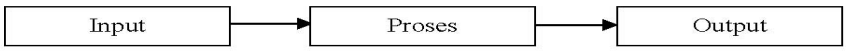
b. Misi

- Menunjang peningkatan mutu peserta didik dengan pembelajaran PAIKEM dan menumbuhkan kearifan dalam bertindak dan bersikap yang bersumber pada nilai – nilai luhur budaya bangsa.

2.2 Konsep Dasar Sistem

Mempelajari suatu sistem akan lebih memahami apabila mengetahui terlebih dahulu apakah sistem itu. Dimana definisi sistem mempunyai peranan penting di dalam pendekatan untuk mempelajari suatu sistem. Pendekatan sistem yang merupakan kumpulan dari elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang luas karena suatu sistem dapat terdiri dari beberapa sub sistem.

Berikut ini akan diperlihatkan gambar mengenai model sistem sederhana dan sistem yang memiliki beberapa input dan output.



Gambar 2.1 Model Sistem Sederhana

2.2.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu sebagai berikut:

1. Komponen-komponen (*components*)

   Setiap sistem baik sistem dalam skala besar maupun sistem dalam skala kecil sekalipun memiliki komponen-komponen atau elemen-elemen. Komponen-komponen ini saling berhubungan dan bekerja sama sehingga tercipta satu kesatuan fungsi dari sistem. Sehingga sistem dapat mencapai tujuannya.

2. Penghubung Sistem *(Sistem Interface)*

   Penghubung sistem merupakan media perantara antara subsistem yang satu dengan subsistem yang lainnya. Melalui penghubung sistem ini, maka subsistem-subsistem dapat saling meberi dan menerima sumber daya sehingga terjalin kerja sama dan dapat membentuk satu kesatuan fungsi dari sistem.

3. Lingkungan luar (*Environment*)

   Lingkungan luar dari sistem adalah segala sesuatu yang berada di luar batas sistem. Lingkungan luar ini bisa juga berupa ekosistem dimana sistem tersebut berada. Walaupun keberadaannya diluar sistem, tapi lingkungan luar dapat mempengaruhi sistem. Adanya ketidakserasian antara lingkungan luar dengan sistem dapat menyebabkan terganggunya fungsi sistem. Oleh karena itu harus senantiasa tercipta keharmonisan antara sistem dengan lingkungan luarnya.

4. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah pemisah antara satu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memberikan ruang lingkup yang jelas dari suatu sistem. Dengan adanya ruang lingkup yang jelas dari sistem tersebut, maka kita dapat memisahkan dan membedakan satu sistem dengan sistem yang lainnya maupun sistem dengan lingkungan luar.

5. Masukan Sistem (*Sistem Input*)

Masukan adalah bahan atau energi yang dimasukkan kedalam sistem. Energi ini dimasukkan kedalam sistem untuk diproses oleh sistem sesuai dengan fungsi dari sistem agar dapat menghasilkan keluaran.

6. Keluaran Sistem (*Sistem Output*)

Keluaran merupakan hasil dari pengolahan suatu sistem. Keluaran ini tentunya diharapkan dapat berguna sesuai dengan tujuan dari sistem. Selain sebagai hasil akhir, sebagian keluaran bisa juga dijadikan masukan untuk sistem lainnya.

7. Pengolah Sistem (*Sistem Processing*)

Pengolah sistem adalah mesin atau mekanisme yang digunakan untuk mengubah masukan menjadi keluaran. Pengolah memiliki peranan yang penting, karena disinilah proses perubahan dan pendayagunaan

masukan terjadi sehingga menghasilkan keluaran yang sesuai dengan tujuan sistem.

8. Sasaran dan Tujuan (*goal objective*)

   Suatu sistem pasti mempunyai tujuan *(goal)* atau sasaran (*objective*). Tujuan merupakan hal akhir yang ingin dicapai oleh suatu sistem, sedangkan sasaran merupakan hal-hal yang menjadi objek dan titik fokus untuk meraih tujuan. Suatu sistem bisa dikatakan berhasil menjalankan fungsinya bila berhasil mencapai sasaran dan tujuan dari sistem tersebut.

2.2.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari berberapa sudut pandang, diantaranya sebagai berikut:

1. Sistem diklasifikasikan sebagai abstark (*abstract system*) dan sistem fisik (*physical system*).

   Sistem Abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*).

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia melibatkan interaksi manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebutnya dengan *man-machine sytem.*

3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tertentu (*probabilistic system*).

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interkasi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem tak tertentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*)

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh oleh lingkungan luarnya.

2.2.3 Konsep Dasar Data dan Informasi

Data adalah bahan yang akan diolah atau diproses, dapat berupa huruf--huruf, angka-angka, simbol-simbol yang menunjukkan suatu fungsi berdiri sendiri, data disusun untuk diolah dalam bentuk database misalnya untuk

mendukung sistem informasi terutama dalam penyediaan dan pengolahan data.

Betapapun kecilnya suatu sistem selalu mengandung sub sistem-sub sistem. Setiap sub sistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

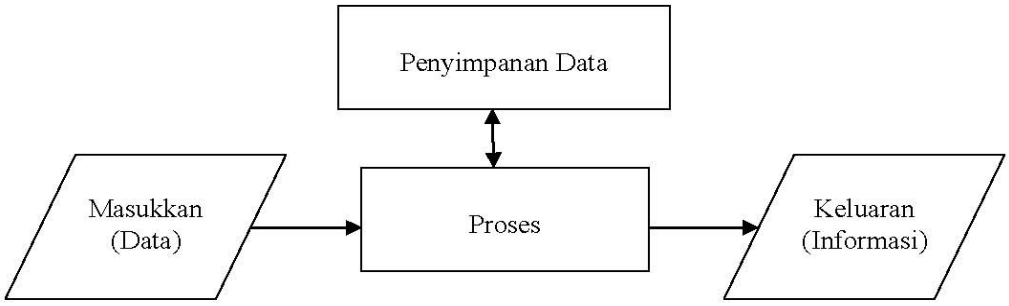
Antar aktivitas di dalam sistem memerlukan hubungan kerja yang biasa diwakili dengan munculnya informasi dari sistem, sehingga manajemen dapat melakukan pengawasan terhadap aktivitas yang ada.

Sebuah sistem mewakili suatu kehidupan tertentu pada suatu lingkungan, dimana didalamnya terdapat sejumlah aktivitas kerja. Biasanya aktivitas yang tampak adalah aktivitas yang sebagian besar dilakukan oleh manusia dan juga aktivitas lain yang dibantu oleh teknologi komputer.

Interaksi antara elemen-elemen sistem maupun subsistem-subsistem membentuk sebuah informasi yang berguna bagi penerimanya. Informasi sangat penting artinya bagi sebuah organisasi dan dari informasi inilah sebuah organisasi dapat mengetahui dan mengatur kehidupannya serta dapat mengembangkan diri.

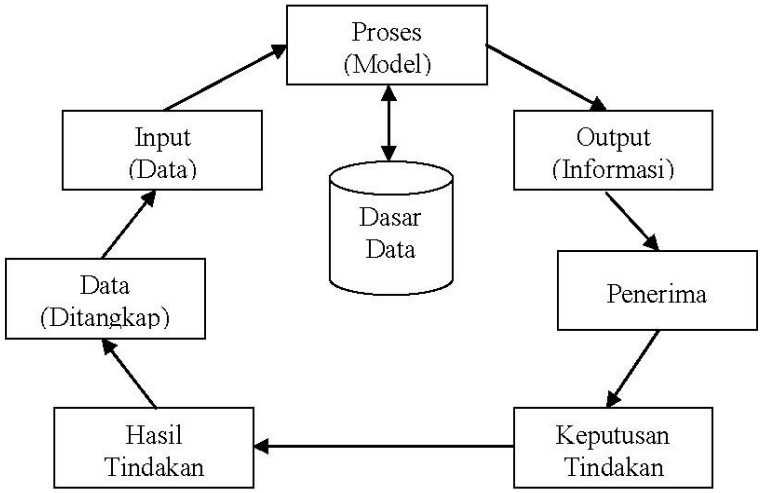
Dari suatu pendapat dikatakan bahwa: "*Informasi adalah data yang diolah menjadi suatu bentuk tertentu yang lebih berguna dan lebih berarti bagi penerimanya"*.

Berikut akan diperlihatkan gambar mengenai hubungan antara data dengan informasi :



Gambar 2.2 Hubungan Data dan Informasi

Suatu sumber informasi adalah data. Data adalah kenyataan-kenyataan yang menggambarkan kejadian atau fakta kehidupan. Untuk menghasilkan informasi, data harus diolah melalui suatu proses. Setelah itu informasi akan diterima oleh yang membutuhkan untuk bahan pengambilan keputusan dan melakukan tindakan. Dari tindakan akan muncul sejumlah kejadian baru dan ditangkap kembali sebagai data. Kemudian data baru ini akan diproses ulang melalui proses tertentu dan seterusnya membentuk suatu siklus seperti dipelihatkan pada gambar dibawah ini



Gambar 2.3 Siklus Informasi

Kualitas sebuah informasi (*quality of information*) ditentukan oleh beberapa faktor, yaitu informasi harus akurat (*accurate*), tepat pada waktunya (*timelines*) dan relevan (*relevance*).

1. Akurat Artinya informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi.

2. Tepat Pada Waktunya Artinya informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah lama tidak akan mempunyai nilai lagi, karena informasi merupakan landasan dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi.

3. Relevan Artinya informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk setiap orang akan berbeda.

Nilai suatu informasi (*value of information*) ditentukan oleh dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan biaya mendapatkannya. Namun demikian sangat sukar untuk menaksir nilai keuntungan sebagian informasi dengan satuan mata uang. Penilaian hanya dapat dilakukan dari nilai efektifitasnya. Pengukuran nilai informasi biasanya dihubungkan dengan nilai efektifitasnya (*Cost Efectifiness*) atau nilai keuntungan (*Cost Benefit*).

Informasi merupakan hal yang sangat penting bagi manajemen didalam pengambilan keputusan. Informasi dapat diperoleh dari sistem informasi (*information system*) atau disebut juga dengan *processing system* atau *information processing system* atau *Information Generating System*.

Secara umum sistem informasi dapat dikatakan sebagai sistem manusia mesin yang terintegrasi, yang menyediakan informasi serta mendukung pelaksanaan operasional manajemen dan fungsi pengambilan keputusan dalam sebuah organisasi.

Secara teori, komputer tidak harus digunakan didalam suatu sistem informasi, tetapi pada kenyataannya tidaklah mungkin suatu sistem informasi yang kompleks dapat berfungsi tanpa melibatkan elemen komputer dan tentunya juga elemen non-komputer yaitu manusia itu sendiri.

2.2.4 Konsep Dasar Sistem Informasi

Sistem Informasi (SI) merupakan sistem pembangkit informasi, artinya sistem yang menyediakan informasi. Dengan integrasi yang dimiliki antar sub-sistemnya, Sistem Informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat, dan akurat sesuai dengan manajemen yang membutuhkannya.

Sistem Informasi Berbasis Komputer (*Computer Based Information Sistem*-CBIS) mengandung arti bahwa komputer memainkan peranan penting dalam sebuah Sistem Informasi. Lebih jelasnya, CBIS merupakan sistem pengolah data

menjadi sebuah informasi yang berkualitas dan dipergunakan untuk suatu alat bantu pengambilan keputusan. Beberapa istilah yang terkait dengan CBIS antara lain adalah data, informasi, sistem, sistem informasi, dan "basis komputer" sebagai kata kuncinya.

Dengan semakin majunya teknologi sekarang saat ini, diperusahaan-perusahaan selau diterapkan suatu sistem informasi yang baru dengan mengikuti perkembangan jaman. Dengan diterapkannya sistem yang dirancang dengan baik akan mempermudah didalam pengoreksian jika terjadi kesalahan-kesalahan atau kendala yang terjadi di dalam perusahaan.

Informasi dihasilkan oleh suatu proses sistem informasi dan bertujuan menyediakan informasi untuk membantu pengambilan keputusan manajemen, operasi perusahaan dari hari ke hari dan informasi yang layak untuk pihak perusahaan.

Menurut *Robert A. Leitch* dan *K. Roscoe Davis* di dalam bukunya *Accounting Informatioon Systems* mendefinisikan sistem informasi sebagai berikut:

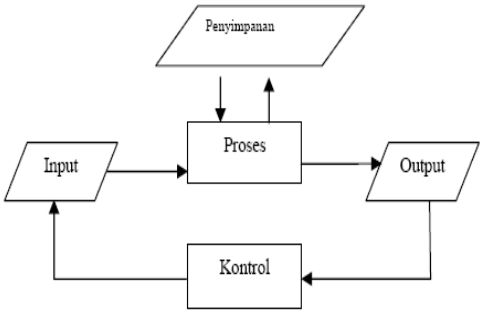
"Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan".

Sedangkan menurut *Susanto Azhar* di dalam bukunya *Pengantar Aplikasi Komputer* mendefinisikan sistem informasi sebagai berikut:

"*Sistem Informasi adalah kumpulan dari sub-sub sistem komponen baik phisik maupun non phisik yang saling berhubungan satu sama lain dan bekerjasama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berguna*".

Dari definisi diatas dapat diambil kesimpulan bahwa sistem informasi merupakan perpaduan antara manusia, alat teknologi, media, prosedure dan pengendalian yang bertujuan untuk menata jaringan komunikasi sehingga dapat membantu dalam pengambilan keputusan yang tepat. Kegiatan yang terdapat pada sistem informasi antara lain :

1. Input, menggambarkan suatu kegiatan untuk menyediakan data yang akan diproses
2. Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah
3. Output, suatu kegiatan untuk menghasilkan laporan dari proses diatas
4. Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data
5. Kontrol, suatu aktifitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan



Gambar 2.4 Kegiatan Sistem Informasi

2.2.5 Peranan Sistem Informasi

Sistem informasi pada hakekatnya adalah suatu pelayanan yang harus dapat melayani kebutuhan informasi yang ada dalam organisasi. Pentingnya peran informasi bagi organisasi menyebabkan pentingnya identifikasi terhadap karakteristik kebutuhan.

Peran dari setiap informasi harus disesuaikan dengan pekerjaan setiap tingkatan manajerial di organisasi. Setiap aktivitas manajerial pada suatu organisasi memiliki target yang berbeda-beda dan memiliki kebutuhan informasi yang sifatnya *intern* maupun *ekstern*.

Bila ditinjau dari tugas masing-masing tingkat manajerial, maka semestinya sistem informasi untuk manajemen yang dirancang sesuai dengan tugas dan wewenang masing-masing tingkat manajerial yang ada. Jenis informasi untuk setiap tingkatan manajerial bermacam-macam. Berikut ini adalah penjelasan tentang informasi yang sesuai dengan tingkat manajerial :

1. Manajemen Level Atas, informasi digunakan untuk perencanaan

strategis, kebijakan dan pengambilan keputusan.

2. Manajemen Level Menengah, informasi digunakan untuk perencanaan taktis dan pengambilan keputusan.

3. Manajemen Level Bawah, untuk pemrosesan transaksi dan merespon permintaan.

2.3 Analisis Sistem

Analisis sistem (*system analysis*) dapat didefinisiskan sebagai berikut : "*Penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya*"

Tahap analisis sistem dilakukan setelah tahap perencanaan (*system planning*), dan sebelum tahap desain sistem (*system design*). Tahap analisis merupakan tahap yang sangat kritis dan sangat penting, karena kesalahan pada tahap ini akan menyebabkan kesalahan pada tahap selanjutnya.

Analsis yang dilakukan menggunakan metode analisis terstruktur, dimana metode analisis terstruktur ini banyak memakai *tool-tool* dalam mempresentasikan suatu sistem, sehingga penguraian sistem dapat terstruktur dengan baik, jelas dan lebih mudah dipahami.

2.3.1 Diagram Konteks

Diagram konteks merupakan diagram yang mengandung satu proses yang menggambarkan hubungan keterkaitan antara sistem dengan pihak-pihak diluar lingkungan sistem dan posisi sistem didalam lingkungan tersebut. Pihak-pihak tersebut merupakan pihak-pihak yang membutuhkan informasi dan data dari sistem ataupun pihak-pihak yang menjadi sumber informasi dan data bagi sistem. Hubungan keterkaitannya digambarkan sebagai aliran informasi dan data yang masuk ke dalam sistem dan keluar dari sistem.

2.3.2 Data Flow Diagram (DFD)

DFD adalah sebuah teknik grafik yang menggambarkan aliran data dan transformasi yang digunakan sebagai perjalanan data dari masukan menuju ke keluaran. DFD dapat diartikan juga sebagai model jaringan dari sebuah sistem. DFD dapat menggambarkan proses-proses yang terjadi dan aliran data diantaranya.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut disimpan. Secara umum DFD dapat diartikan sebagai salah satu *tools* untuk analisis sistem yang dapat bermanfaat untuk menggambarkan proses, aliran data, *entity* yang terlibat serta *data store* yang digunakan dalam sistem yang dipelajari. Dengan menuangkan hasil analisis ke dalam DFD, seorang analis dapat memahami sistem yang sedang dipelajari

dengan mudah dan baik.

Dibawah ini terdapat simbol-simbol untuk DFD yang diusulkan oleh Yourdon :

1. *Proses*, suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, prosedur atau alat yang digunakan untuk mentransformasikan data.

2. *Data Flow* (Arus Data), data yang mengalir dengan arah tertentu dari asal ke tujuan. Data yang mengalir dapat berupa dokumen, surat atau bentuk lainnya.

3. *Data Store* (Penyimpanan Data), digunakan untuk menyimpan dan mengambil data oleh proses. Data yang disimpan dapat berupa data yang terkomputerisasi maupun tidak terkomputerisasi.

4. *Source* atau *destination* atau dikenal juga dengan *external entity*, berupa orang, organisasi atau sistem lain yang berada diluar batas sistem yang berinteraksi dengan sistem yang sedang dikembangkan.

Dalam DFD terdapat beberapa aturan dasar penyusunan DFD yang dapat membantu untuk mempermudah penggambaran diagram arus data, diantaranya :

1. Setiap data yang dihasilkan atau keluar dari proses harus didasarkan pada data yang masuk ke proses tersebut.

2. Semua aliran data yang dihasilkan harus diberi nama, nama yang

diberikan harus mencerminkan aliran data antara proses, *data store* dan *entity* yang ada.

3. Hanya data yang diperlukan untuk melakukan suatu proses saja yang harus digunakan sebagai input suatu proses.

4. Suatu proses harus tergantung pada input dan output yang masuk ke dalam proses itu saja, tidak perlu memperhatikan apa yang terjadi pada proses lainnya.

5. Setiap proses yang ada harus merupakan proses yang berjalan terus menerus, setiap proses harus diasumsikan siap menangani fungsi atau kerja setiap proses.

Agar representasi sistem dalam DFD mudah dipahami, maka DFD disusun dalam bentuk bertingkat (*leveled*) yang merupakan rincian lanjut dari proses pada level yang sebelumnya. Dimulai dari tingkat yang tertinggi dilakukan identifikasi proses apa saja yang ada dan data apa saja yang mengalir antar proses yang ada. Selanjutnya, dilakukan analisis terhadap setiap proses yang ada pada level tersebut, apakah sudah merupakan proses yang melakukan satu fungsi saja atau masih melakukan beberapa fungsi yang berbeda. Jika proses yang dianalisis masih melakukan lebih dari satu fungsi yang berbeda, maka perlu dilakukan pemecahan terhadap proses tersebut menjadi level lebih tinggi. Demikian seterusnya sampai didapat semua proses yang ada pada level paling tinggi dilakukan satu fungsi saja.

Jika DFD untuk suatu sistem yang dianalisis telah dibuat, sebelum melanjutkan kegiatan lainnya selalu dilakukan evaluasi atau pemeriksaan terhadap DFD tersebut. Pemeriksaan dilakukan terhadap munculnya kesalahan yang dapat diakibatkan oleh salah gambar, tidak konsistennya penggunaan simbol dan analisis terhadap proses yang terjadi.

2.3.3 Perancangan Sistem

Perancangan sistem adalah suatu fase kreatif dalam siklus pengembangan sistem, dimana sistem yang dirancang harus memenuhi tujuan sistem secara teknis dan operasional. Pada tahap ini perlu dikembangkan spesifikasi untuk keluaran, database, operasi pemrosesan data, masukkan data, pengendalian serta pengamanan.

Perancangan sistem informasi yang dilakukan, bertujuan untuk menghilangkan beberapa kelemahan sistem dan meningkatkan kemampuan sistem dalam menyelesaikan permasalahan yang ada dengan acuan hasil-hasil analisis yang telah dilakukan.

2.3.3.1 Basis Data

Basis Data terdiri dari dua 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 mewakili suatu objek seperti manusia, barang, hewan, peristiwa dan sebagainya.

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan lainnya. Basis data adalah kumpulan-kumpulan file yang saling berkaitan.

Penerapan basis data dalam sistem informasi disebut dengan sistem basis data (*database sistem*). Sistem basis data ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lain dan tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi.

Tujuan dari desain basis data ini adalah untuk menentukan data-data yang dibutuhkan dalam sistem, sehingga informasi yang dihasilkan dapat terpenuhi dengan baik. Perancangan *database* yang digunakan adalah untuk memudahkan dalam mengetahui *file-file database* yang digunakan dalam perancangan sistem, sekaligus untuk mengetahui hubungan antara *file* dari *database* tersebut.

Beberapa kriteria yang harus dipenuhi dalam *database* adalah sebagai berikut :

1. Menyimpan seluruh data dan informasi secara terpusat.

2. Mengurangi redudansi data atau duplikasi data.

3. Melakukan perubahan-perubahan data untuk menyelesaikan dan untuk pengembangan yang akan datang.

4. Menjamin keamanan data.

2.3.3.2 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) adalah suatu model diagram yang menyatakan keterhubungan suatu entity dengan entity yang lain. Atau juga dapat dikatakan sebagai sebuah teknik untuk menggambarkan informasi yang dibutuhkan dalam sistem dan hubungan antar data-data tersebut.

Secara terjemahan dalam bahasa Indonesia, *Entity Relationship Diagram* adalah diagram relasi atau keterhubungan entitas. Dari model *Entity Relationship Diagram* akan didapatkan data-data yang dibutuhkan sistem. Dengan begitu maka akan didapatkan pula kejelasan aktivitas yang dilakukan dalam sistem.

Didalam *Entity Relationship Diagram* (ERD) dikenal beberapa komponen, yaitu sebagai berikut :

1. Entitas (*Entity*)

   Adalah suatu objek yang memiliki hubungan dengan objek lain. Dalam ERD digambarkan dengan bentuk persegi panjang.

2. Hubungan (*Relationship*)

Dimana entitas dapat berhubungan dengan entitas lain, hubungan ini disebut dengan *entity relationship* yang digambarkan dengan garis. Berikut adalah beberapa jenis hubungan yang dapat terjadi dalam suatu basis data :

a. Relasi 1-1 (*One to One*) Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu.

b. Relasi 1-N (*One to Many*) Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak, tetapi tidak sebaliknya, dimana *file* kedua hanya berhubungan dengan satu entitas pada *file* pertama.

c. Relasi N-1 (*Many to One*) Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding satu, tetapi tidak sebaliknya, dimana *file* pertama hanya berhubungan dengan satu entitas pada *file* kedua.

d. Relasi N-N (*Many to Many)* Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak.

3. Atribut

Adalah elemen dari entitas yang berfungsi sebagai deskripsi karakter entitas dan digambarkan dengan bentuk elips.

2.3.3.3 Kamus Data

"*Kamus data atau data directory adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi*".

Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap perancangan sistem, kamus data dapat digunakan untuk merancang *input*, *output*, dan merancang *database* program. Kamus data dibuat berdasarkan arus data yang ada.

2.3.3.4 Model-Model SDLC

Proses pengembangan sistem informasi adalah sekumpulan tahap, tugas dan aktivitas yang dibutuhkan untuk secara efisien mentransformasikan kebutuhan pemakai ke suatu solusi sistem informasi yang efektif. Pemodelan proses sistem informasi bertujuan untuk mempresentasikan aktivitas yang terjadi selama pembuatan sistem informasi dan perubahan-perubahan (evolusi). Latar belakang penggunaan model-model tersebut adalah kebutuhan untuk menghasilkan suatu sistem yang benar sedini mungkin di dalam proses pengembangannya. Alasan utama adalah biaya, semakin dini suatu kesalahan bisa dideteksi dalam pengembangan sistem, maka perbaikannya semakin rendah. Suatu model proses yang baik dan terstruktur akan menentukan urutan tahap dalam suatu proses dan menciptakan kriteria transisi untuk maju ke tahap selanjutnya.

Model-model *Sistem Development Life Cycle* (SDLC) adalah :

1. Model *Waterfall*

   Model *Waterfall* ini juga dikenal dengan nama model tradisional atau model klasik. Model inipaling banyak dikenal dan digunakan. Langkah-langkah yang penting dalam model ini adalah :

   a. Desain sistem dan sistem informasi.

   b. Implementasi dan uji coba unit.

   c. Integrasi dan ujicoba sistem.

   d. Operasi dan pemeliharaan.

2. Model *Prototyping*

   Model ini adalah suatu proses yang memungkinkan pengembangan aplikasi untuk menciptakan suatu model dari sistem informasi yang harus dikembangkan. Tujuan dari *prototyping* adalah untuk mengurangi resiko dan ketidakpastian dari tahap-tahap awal dari *life cycle* pengembangan sistem informasi.

3. Model *Exploratory*

   Model ini dipakai dengan awal seadanya. Pengembangan sistem tidak sepenuhnya dimengerti dan menguasai *requirement* (kebutuhan) dari sistem. Sistem dikembangkan dengan adanya *requirement* baru.

4. Model *Incremental*

Model ini adalah pengembangan lebih lanjut dari model *exploratory* yang didesain untuk menjawab kritik terhadap *prototyping*. *Requirement* didefinisikan dan diserahkan kepada pemakai dalam bentuk bertahap.

5. Model *Transformational*

   Model ini adalah model yang berorientasi pada spesifikasi formal. Spesifikasi ini ditransformasikan melalui beberapa tahap yang telah diuji kebenarannya sebelum menjadi sistem seutuhnya.

6. Model *Spiral*

   Model ini dikembangkan oleh Boehm. Model ini cocok digunakan untuk organisasi yang besar.

7. Model 4GT

   Pada alat bantu *Fourth Generation Technique* (4GT) memungkinkan pengembangan spesifikasi sistem informasi secara otomatis sesuai dengan yang akan diterjemahkan ke program. Dengan menggunakan 4GT, *software engineering* memusatkan perhatian hanya kepada spesifikasi sistem informasi yang diinginkan.

2.4 Perangkat Lunak Penunjang

2.4.1 Visual Basic

Visual basic merupakan bahasa pemograman yamg tercepat dan termudah untuk membuat satu aplikasi dalam microsoft windows. Dengan menggunakan metode *Graphical User Interface (GUI).* Visual basic memudahkan pemograman untuk berinteraksi langsung dengan elemen-elemen untuk setiap bentuk pemograman. Visual basic dibuat sebagai langkah pengembangan untuk menyesuaikan BASIC *(Beginners All-Purpose Symbolic Intruction Code*) yang berbasis DOS yang tidak mempunyai kemampuan menggunakan metode GUI dalam basis windows.

Sebagai programs yang berbasis windows, Visual Basic mempunyai kemampuan untuk berinteraksi dengan seluruh aplikasi Windows seperti Microsoft Word, Microsoft Excel, Microsoft Acces dan sebagainya. Dengan kemampuan yang hampir tidak terbatas, Visual basic dapat digunakan untuk semau jenis aplikasi pemograman. Anda dapat membuat aplikasi windows swperti word atau excel ataupun game, multimedia, program penghitungan dan sebagainya. Visual basic dapat digunakan untuk membuat objek-objek pembantu program seperti misalnya Control Activex, file help, aplikasi internet dan sebagainya. Selain itu visual basic juga dapat digunakan untuk menguji program (debugging) dan menghasilkan program akhir berakhiran exe yang bersifat executable atau dapat langsung digunakan.

2.4.2 Microsoft Access

Microsoft Access (atau Microsoft Office Access) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah Microsoft Office Access 2007 yang termasuk ke dalam Microsoft Office System 2007.

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/*programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

2.4.3 Algoritma Taboosearch

Taboosearch berasal dari Tongan, suatu bahasa Polinesia yang digunakan oleh suku Aborigin Pulau tonga untuk mengindikasikan suatu hal yang tidak boleh "disentuh" karena sakralnya. Menurut kasus Webster, Taboo berarti larangan yang dipaksakan oleh kebudayaan social sebagai suatu tindakan pencegahan atau

sesuatu yang dilarang karena berbahaya. Bahaya yang harus dihindari dalam *Taboosearch* adalah penjadualan yang tidak layak, dan terjebak tanpa ada jalan keluar. Dalam konteks lebih luas, larangan perlindungan dapat diganti jika terjadi tuntutan yang mendadak.

Taboosearch adalah sebuah metode optimasi yang berbasis pada *local search*. Proses pencarian bergerak dari satu solusi ke solusi berikutnya, dengan cara memilih solusi terbaik *neighbourhood* solusi sekarang (*current*) yang tidak tergolong solusi terlarang (*tabu*). Ide dasar dari algoritma *taboosearch* adalah mencegah proses pencarian dari *local search* agar tidak melakukan pencarian ulang pada ruang solusi yang sudah pernah ditelusuri, dengan memanfaatkan suatu struktur memori yang mencatat sebagian jejak proses pencarian yang telah dilakukan.

Struktur memori fundamental dalam *taboosearch* dinamakan *taboolist*. *Taboolist* menyimpan atribut dari sebagian *move* (transisi solusi) yang telah diterapkan pada iterasi-iterasi sebelumnya. *Taboosearch* menggunakan *taboolist* untuk menolak solusi-solusi yang memenuhi atribut tertentu guna mencegah proses pencarian mengalami *cycling* pada daerah solusi yang sama, dan menuntun proses pencarian menelusuri daerah solusi yang belum dikunjungi. Tanpa menngunakan strategi ini, *local search* yang sudah menemukan solusi optimum local dapat terjebak pada daerah solusi optimum local tersebut pada iterasi-iterasi berikutnya. *List* ini mengikuti aturan LIFO dan biasanya sangat pendek (panjangnya biasanya sebesar $O(\sqrt{N})$, dimana N adalah jumlah total dari operasi). Stiap saat ada langkah itu akan ditempatkan dalam *tau list*.

Perekaman solusi secara lengkap dalam sebuah *forbidden list* dan pengecekan apakah sebuah kandidat solusi tercatat dalam list tersebut merupakan cara yang mahal, baik dari sisi kebutuhan memori maupun kebutuhan waktu komputasi. Jadi, *taboolist* hanya menyimpan langkah transisi (*move*) yang merupakan lawan atu kebalikan dari langkah yang telah digunakan dalam iterasi sebelumnya utnuk bergerak dari satu solusi ke solusi berikutnya. Dengan kata lain *taboolist* berisi langkah-langkah yang membalikan solusi yang baru ke solusi yang lama.

Pada tiap iterasi, dipilih solusi baru yang merupakan solusi terbaik dalam *neighbourhood* dan tidak tergolong sebagai tabu. Kualitas solusi baru ini tidak harus lebih baik dari kualitas solusi sekarang. Apabila solusi baru ini memiliki nilai fungsi objektif lebih baik dibandingkan solusi terbaik yang telah dicapai sebelumnya, maka solusi baru ini dicatat sebagai solusi terbaik yang baru.

Sebagai tambahan dari *taboo list,* dikenal adanya *criteria aspirasi*, yaitu suatu Penanganan khusus terhadap *move* yang dinilai dapat menghasilkan solusi yang dinilai dapat menghasilkan solusi yang baik namun *move* tersebut berstatus tabu. Dalam hal ini, jika *move* tersebut memenuhi criteria aspirasi yang telah ditetapkan sebelumnya, maka *move* tersebut dapat digunakan utnuk membentuk solusi berikutnya (status tabunya dibatalkan). Berikut ini diberikan kerangka umum algoritma *taboosearch* dalam notasi bahasa pascal.

```
begin
{Buat solusi awal s yang feasibel dengan
menggunakan
suatu metode heuristik tertentu atau secara
acak}
best := cost(s);
s* := s; {s* adalah solusi terbaik yang
diperoleh}
tabu_list := null;
repeat
Candidate(s) := (s' ∈ N(s): merupakan move
dari s
```

2.4.5.1 Ide Dasar Taboosearch

Sebuah informasi akan digunakan sebagai petunjuk untuk bergerak dari i ke solusi selanjutnya dalam N(i). Penggunaan memori sebagai pembatas dalam pemilihan beberapa subset dari N(i) dengan mencegah pergerakan ke beberapa solusi tetangga. Sebuah informasi akan digunakan sebagai petunjuk untuk bergerak dari i ke solusi selanjutnya dalam N(i). Penggunaan memori sebagai pembatas dalam pemilihan beberapa subset dari N(i) dengan mencegah pergerakan ke beberapa solusi tetangga.

Secara formal,kita dapat menganggap masalah optimalisasi dalam cara berikut: Diberikan sebuah himpunan solusi *S* dan sebuah fungsi *f : S*, temukan solusi *i** dalam *S* sehingga *f(i*)* dapat diterima dengan beberapa kriteria. Secara umum kriteria untuk dapat diterima sebagai solusi *i** harus memenuhi *f(i*)=f(i)*

untuk setiap *i* dalam *S*. Dalam situasi metode pencarian tabooakan menjadi sebuah algoritma minimimasi secara pasti yang menyediakan proses eksplorasi yang menjamin setelah sejumlah langkah-langkah terhingga *i** dapat dicapai.

Untuk mendalami lagi prinsip kerja metode pencarian tabu, kita dapat memformulakan metoda menurun klasik (*classical descent method*) dalam beberapa langkah, yaitu:

1. Memilih sebuah solusi awal *i* dalam *S*
2. Membangkitkan subset *V** sebagai solusi dalam *N(i)*
3. Mencari j 'terbaik' dalam *V** dan menetapkan *i=j*
4. Jika *f(j)=f(i)* maka berhenti, namun jika tidak kembali ke langkah ke-2

Dalam metode menurun secara umum akan dapat langsung ditetapkan bahwa *V** = *N(i)*. Tetapi hal ini seringkali membutuhkan waktu yang lama. Untuk itulah cara pemilihan *V** yang tepat seringkali dijadikan sebagai peningkatan yang penting dalam metode pencarian. Dalam metode menurun secara umum akan dapat langsung ditetapkan bahwa *V** = *N(i).* Tetapi hal ini seringkali membutuhkan waktu yang lama. Untuk itulah cara pemilihan *V** yang tepat seringkali dijadikan sebagai peningkatan yang penting dalam metode pencarian. Pada kasus yang berlawanan, akan ditetapkan *|V*|=1*. Hal ini akan menurunkan fase pemilihan *j* 'terbaik'. Solusi *j* akan diterima jika *f(j)=f(i*), sebaliknya hal ini akan diterima dengan kemungkinan tertentu yang bergantung pada nilai-nilai *f* pada *i* dan *j* serta pada sebuah parameter yang disebut temperatur. Tidak ada temperatur dalam metode pencarian tabu, namun pemilihan *V** akan menjadi hal yang penting guna mendefinisikannya dalam

setiap langkah di mana akan terjadi penggunaan memori secara sistematis untuk memanfaatkan informasi yang ada di luar fungsi f dan lingkungan *N(i).*

Sebagai pengecualian pada kasus-kasus istimewa, penggunaan prosedur menurun (*descent procedure*) secara umum lebih rumit karena kita akan terperangkap pada sebuah minimum lokal yang mungkin masih jauh dari minimum global. Maka untuk proses iterasi dalam eksplorasi apapun sebaiknya dalam beberapa hal juga menerima langkah-langkah yang tidak akan memberikan perkembangan dari i ke j dalam *V** (misal *f(j)>f(i)*). Metode pencarian taboosecara berbeda memilih *j* 'terbaik' dalam *V**.

Selama pergerakan yang tidak memberi perkembangan itu mungkin, resiko pengunjungan kembali sebuah solusi atau lebih umumnya disebut sebagai siklus mungkin untuk terjadi. Dalam hal inilah penggunaan memori sangat diperlukan untuk mencegah terjadinya pergerakan ke solusi yang telah dikunjungi. Jika memori seperti itu diperkenalkan, maka kita dapat menganggap struktur *N(i)* tergantung pada pengelilingan yang merupakan pengulangan k. Jadi kita dapat merujuk ke *N(i,k)* daripada *N(i).*

Dengan perujukan ini kita dapat mencoba untuk melakukan peningkatan algoritma menurun dalam beberapa hal untuk lebih mendekatkan metode ini ke prosedur dalam metode pencarian taboosecara umum. Hal ini dapat didefinisikan dalam beberapa langkah (catatan *i** adalah solusi 'terbaik' yang ditemukan dan k adalah penghitung dalam pengulangan) :

1. Memilih solusi awal *I* dalam *S*. Tetapkan *i*=I* dan *k=0.*

2. Tetapkan nilai $k=k+1$ dan membagkitkan subset V^* sebagai solusi dalam $N(i,k)$
3. Pilih j 'terbaik' dalam V^* dan tetapkan $i=j$.
4. Jika $f(i) < f(i^*)$ maka tetapkan $i^*=i$.
5. Jika kondisi berhenti ditemukan, maka proses dihentikan, sedangkan jika belum kembali ke langkah 2.

Amati bahwa langkah-langkah pada metode penurunan klasik termasuk dalam formula ini (kondisi berhenti secara sederhana *jika* $f(i)=f(i^*)$ dan i^* akan selalu menjadi solusi akhir). Selain itu kita juga dapat mempertimbangkan penggunaan f yang telah dimodifikasi daripada f yang dalam beberapa keadaan di deskripsikan kemudian.

Dalam metode pencarian taboo, kondisi berhenti dapat berupa:

1. $N(i,k+1)$ = tidak terdefinisi
2. k lebih besar daripada bilangan maksimum pada pengulangan
3. Banyaknya pengulangan selama peningkatan terakhir i^* lebih besar dari bilangan tertentu.
4. Pembuktian dapat diberikan daripada solusi optimum yang telah didapatkan.

Selama aturan-aturan berhenti ini memungkinkan untuk memiliki beberapa pengaruh dalam prosedur pencarian dan pada hasil-hasilnya, penting untuk menyadari bahwa pendefinisian $N(i,k)$ dalam tiap pengulangan k dan pemilihan V^* adalah hal yang krusial. Definisi dari $N(i,k)$ menyatakan secara tidak langsung bahwa beberapa solusi yang telah dikunjungi dihapus dari $N(i)$, mereka dianggap sebagai solusi-solusi tabooyang harus dihindari dalam pengulangan selanjutnya.

Sebagai contoh, pemeliharaan pada pengulangan *k* sebuah list *T*(*list taboo*) pada solusi yang telah dikunjungi terakhir |*T*| akan mencegah terjadinya siklus pada ukuran paling banyak sebesar |T|. Pada kasus ini, kita bisa mengambil *N(i,k)=N(i)-T*. Namun list *T* kemungkinan tidak dapat digunakan secara praktis. Oleh karena itu, kita akan mendeskripsikan proses eksplorasi pada *S* dalam masa pergerakan dari satu solusi ke solusi lainnya. Untuk setiap solusi *I* dalam *S,* kita dapat mendefinisikan *M(i)* sebagai himpunan gerak yang dapat digunakan oleh *i* untuk memperoleh solusi baru *j* (notasi: $j=i \nabla m$). Secara umum kita dapat menggunakan gerakan-gerakan yang dapat dibalik. Untuk setiap m terdapat gerakan m^{-1} sehingga $(i \nabla m) \nabla m^{-1} = i$.

Jadi daripada mempertahankan list *T* dari solusi-solusi yang telah dikunjungi, kita dapat secara sederhana memelihara jalur gerak terakhir |*T*| atau gerak balik terakhir |*T*| yang diasosiasikan dfengan gerakan-gerakan yang sebenarnya telah dilakukan. Maka dengan jelas bahwa pembatasan yang ada adalah kehilangan informasi, dan hal itu tidak menjamin tidak terjadinya siklus dengan panjang paling banyak |*T*|. Untuk kepentingan efisiensi, maka diperlukan penggunaan beberapa list *Tγ* dalam satu waktu maka beberapa unsur pokok *tγ* (dari *i* atau dari *m*) akan diberikan taboostatus untuk mengindikasi bahwa unsur pokok ini sedang tidak diperbolehkan untuk terlibat dalam pergerakan. Secara umum pergerakan untuk taboostatus adalah fungsi taboostatus pada unsur-unsur pokoknya yang dapat diubah pada setiap pengulangan. Gerak taboom digunakan pada solusi *i* yang mungkin tampak menarik karena itu diberikan sebagai contoh sebuah solusi yang lebih baik dari pada yang telah ditemukan. Kita akan dapat

menerima *m* tanpa memperhatikan statusnya. Kita akan melakukan hal tersebut jika *m* memiliki tingkat aspirasi (*aspiration level*) *a(i,m)* yang lebih baik daripada permulaan *A(i,m).*

Sekarang kita dapat mendefinisikan karakteristik dari prosedur pencarian taboodalam langkah-langkah berikut, antara lain:

1. Memilih solusi awal i dalam S. Tetapkan i*=I dan k=0.
2. Tetapkan k=k+1 dan bangkitkan sebuah subset V* dari solusi dalam N(I,k) sehingga salah satu dari kondisi tabootγ yang melanggar (γ=1,2,…,t) atau setidaknya satu kondisi aspirasi aγ yang memiliki (γ=1,2,…,a).
3. Pilih j terbaik melalui j=i∇m dalam V* dan tetapkan i=j.
4. Jika f(i)<f(i*) maka tetapkan i*=i.
5. Perbaharui kondisi Taboodan aspirasi.
6. Jika kondisi berhenti ditemukan, maka proses dihentikan. Jika tidak, kembali ke langkah 2.

Hal-hal di atas dapat dikatakan sebagai bahan-bahan dasar dari metode pencarian tabooyang nantinya akan digunakan untuk memecahkan masalah penjadualan pekerjaan (job scheduling) pada pembahasan selanjutnya.

2.4.4 Graf

2.4.4.1 Definisi Graf

Teori graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan sampai saat ini. Graf digunakan untuk mempresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek sebagai noktah, bulatan atau titik, sedangkan hubungan antara objek dinyatakan dengan garis.

Secara metematis, graf didefinisikan sebagai berikut:

DEFINISI 1. Graf *G* didefinisikan sebgai pasangan himpunan (V, E), yang dalam hal ini:

V = himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*)

$= \{ v_1 , v_2 , ... , v_n \}$

dan

E = himpunan sisi (*edges* atau *arc*) yang menghubungkan sepasang simpul

$=\{e_1 , e_2 , ... , e_n \}$

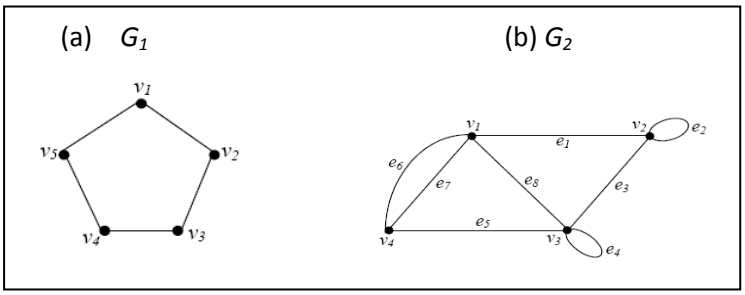
atau dapat ditulis singkat notasi *G = (V, E)*.

Definisi 1 menyatakan bahwa *V* tidak boleh kosong, sedangkan *E* boleh kosong. Kadi, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, tetapi simpulnya harus ada, minimal satu. Graf yang hanya mempunyai satu buah simpul tanpa sebuah sisi pun dinamakan **graf trivial**.

Simpul pada graf dapat dinomori dengan huruf, seperti *a, b , c, ...v, w,....* dengan bilangan asli 1, 2, 3, ... atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul v_i dengan simpul v_j dinyatakan dengan pasangan (v_i, v_j) atau dengan lambing e_1, e_2, ... Dengan kata lain, jika *e* adalah sisi yang

menghubungkan simpul v_i dengan simpul v_j, maka e dapat ditulis sebagai $e = (v_i, v_j)$.

Secara geometri graf digambarkan sebagai sekumpulan noktah (simpul) di dalam bidang dwimatra yang dihubungkan dengan sekumpulan garis (sisi).



Gambar 2.5 Graf Sederhana dan Graf Ganda

2.4.5 Jenis-jenis Graf

Graf dapat dikelompokan menjadi beberapa kategori (jenis) bergantung pada sudut pandang pengelompokannya. Pengelompokan graf dapat dipandang berdasarkan ada tidaknya sisi ganda atau sisi kalang, berdasarkan jumlah simpul, atau berdasarkan orientasi arah pada sisi.

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka secara umum graf dapat digolongkan menjadi dua jenis:

1. **Graf sederhana** (*simple graph*).

Graf yang tidak mengandung gelang maupun sisi-ganda dinamakan graf sederhana. G_1 pada gambar 2.5(a) adalah contoh graf sederhana yang mempresentasikan jaringan komputer. Simpul menyatakan komputer, sedangkan sisi menyatakan saluran telepon untuk berkomunikasi. Saluran telepon dapat beroprasi pada sua arah.

2. Graf tak-sederhana (*unsimple – graph*).

Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (*unsimple graph*). Ada dua macam graf tak-sederhana, yaitu **graf ganda** (*multigraph*) dan **graf semu** (*pseudograph*). Graf ganda adalah graf yang mengandung sisi ganda. Sisi ganda yang menghubungkan sepasang simpul bias lebih dari dua buah. G_2 pada gambar 2.5(b) adalah graf ganda. Sisi ganda pada G_2 dapat diandaikan sebagai saluran telepon tambahan apabila beban komunikasi data antar komputer sangat padat. Graf semu adalah graf yang mengandung gelang. G_3 adalah graf semu (termasuk bila memiliki sisi ganda sekalipun). Sisi gelang pada G_3 dapat dianggap sebagai saluran telepon tambahan yang menghubungakan komputer dengan dirinya sendiri (mungkin untuk tujuan diagnostic). Graf semu lebih umum daripada graf ganda, karena sisi pada graf semu dapat terhubung ke dirinya sendiri.

Jumlah simpul pada graf disebut sebagai kardinalitas graf, dan sinyatakan dengan $n = |V|$, dan jumlah sisi dinyatakan dengan $m = |E|$. Berdasarkan jumlah simpul pada suatu graf, maka secara umum graf dapat digolongkan menjadi dua jenis:

2.4.5.4 Model Matematis Graf Sebagai Representasi *Hard Constraint*

Constrain adalah keadaan yang didefinisan agar penjadualan yang dihasilkan merupakan jadual yang baik dan tidak boleh dilanggar. Untuk mempresentasikan *constrain* yang ada dibangunlah sebuah model matematis dalam bentuk graf.

Graf G didefiniskan sebagai pasangan himpunan (V,E) dimana:

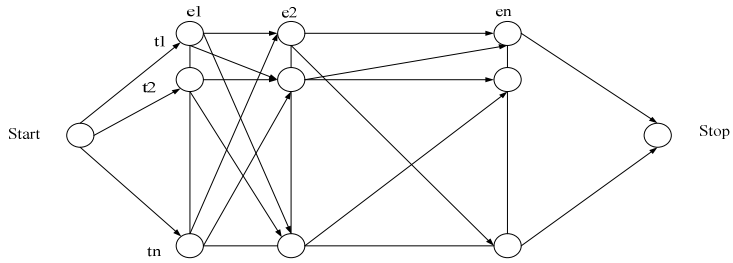
V= himpunan berhingga yang tidak kosong dari simpul-simpul atau vertex/ node $\{V_1,V_2, \ldots\ldots,V_n\}$.

E= himpunan sisi (edge atau arc yang menghubungkan sepasang simpul $\{e_1, e_2,\ldots\ldots, e_n\}$.

Untuk masalah penjadualan mata pelajaran yang akan diselesaikan menggunakan algoritma *Taboosearch*, graf yang dibangun merupakan graf berarah. Node-node yang terbentuk nantinya merupakan representasi dari *subject-subject* yang akan dijadualkan pada waktu t, *subject-subject* tersebut merupakan sekumpulan data yang terdiri dari kelas, dan mata pelajaran yang diajarkan.

Untuk masalah penjadualan mata pelajaran yang akan diselesaikan dengan menggunakan algoritma *Taboosearch* dibuat graf berarah. Node-node yang terbentuk merupakan representasi dari beberapa *subject* yang akan dijadualkan pada *timeslot* t, *subject-subject* tersebut merupakan sekumpulam *record* yang terdiri dari kelas, guru, dan mata pelajaran yang diajarkan. *Timeslot* adalah kumpulan waktu yang memungkinkan untuk dijadualkan pada *subject-subject* tadi. Sedangkan keterhubungan antar node-node yang terbentuk (edge) merupakan

representasi dari *constrain* yang ada, sebagai contoh dapat dilihat pada gambar di bawah ini:



Gambar 2.6 Graf sebagai Representasi *Constraint*

Maksud dari gambar di atas adalah mempresentasikan penelusuran node yang mencari pasangan antara *subject* dan *timeslot* yang memebuhi *constrain*. Pada setiap *subject* terdapat nilai *constrain* yaitu kelas, dan guru. *Subject* pertama sudah mendapatkan *timeslot* yang berimpit atau beririsan maka pemeriksaan dilakukan terhadap nilai *constrain* dari *subject* yaitu kelas dan guru. Jika salah satunya sama dengan *subject* sebelumnya maka *timeslot* tidak bisa terhubung dan harus mencari lagi *timelot* lain yang terbebas dari nilai *constrain* tadi.