BAB II

LANDASAN TEORI

2.1 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem yang dapat memberikan informasi kepada orang lain atau orang tertentu mengenai hasil pengolahan data-data yang telah dilakukan di suatu perusahaan atau instansi dan dibutuhkan sebagai bahan acuan untuk memutuskan suatu hal.

Telah diketahui sebelumnnya bahwa informasi suatu hal yang sangat penting dalam menggambil keputusan, dan informasi tersebut dapat diperoleh dari sistem informasi (*Information system*) atau disebut juga dengan *processing system* atau *information processing system* atau *Information generating system*. Menurut *Robert A. Leitch* dan *K. Roscoe Davis* Sistem Informasi didefinisikan sebagai berikut : “Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan.

Menurut *Robert* dan *Donald Symazky* Sistem Informasi sendiri mempunyai sejumlah komponen yang berbeda yaitu : manusia, data, hardware, dan software. Setiap suatu sistem, setiap komponen tersebut berinteraksi untuk mencapai sasarannya.

Sistem informasi terdiri atas dua gabungan kata yang pada awalnya berdiri sendiri, yaitu sistem dan informasi.

2.1.1 Pengertian Sistem

Pendefinisian sistem dikelompokan menjadi dua macam, yaitu yang menekankan pada komponen-komponen atau elemen-elemennya serta sistem yang menekankan pada prosedur. Sistem yang menekankan pada prosedur mendefinisikan sistem sebagai berikut;

“Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul, bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

Pendekatan yang menekankan pada elemen atau komponen adalah :

“Sistem adalah kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”.

Dari pernyataan penulis menyimpulkan bahwa sasaran atau tujuan dalam sistem yaitu untuk menentukan operasi yang akan dilaksanakan. Selain itu juga mempengaruhi jenis masukan yang dibutuhkan dan keluaran yang dihasilkan oleh suatu sistem. Suatu sistem dapat dikatakan berhasil apabila sasaran atau tujuan tersebut telah tercapai.

Suatu sistem mempunyai karakteristik-karakteristik, yang mana semua karakteristik tersebut tidak dapat dipisahkan antara satu karakteristik dengan karakteristik lainnya. Karakteristik atau sifat sistem adalah sebagai berikut :

1. Komponen-komponen (*components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama untuk membentuk satu kesatuan.

2. Batas sistem (*boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan luar sistem (*environment*)

   Lingkungan luar dari suatu sistem adalah apapun dari luar batas dari sistem yang mempengaruhi operasi sistem.

4. Perhubungan (*interface*)

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lain.

5. Masukan (*input*)

   Masukan adalah energi yang dimasukan kedalam sistem.

6. Keluaran (*output*)

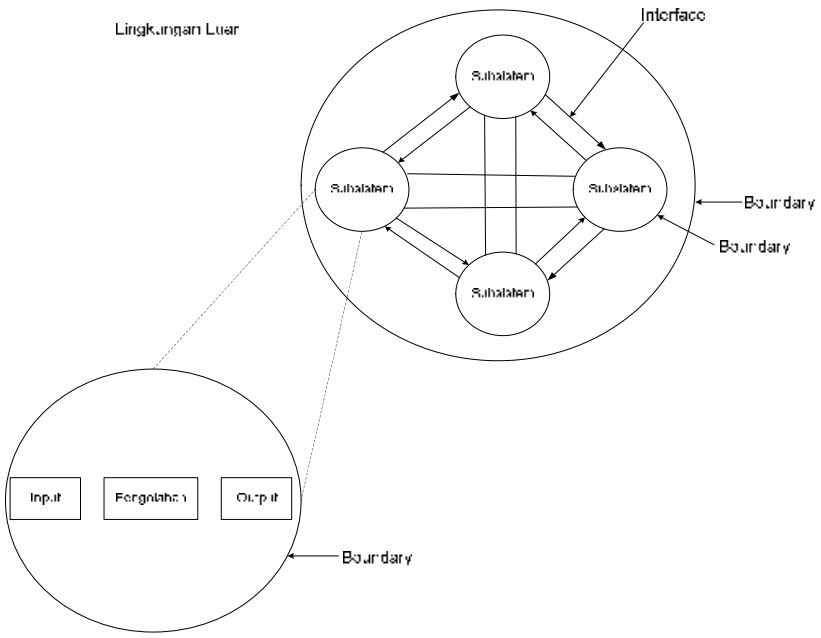
   Keluaran adalah hasil dari energi yang diolah dan diklarifikasikan menjadi keluaran yang berguna.

7. Pengolahan (*process*)

   Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.

8. Sasaran (*objective*) dan Tujuan (*goal*)

   Suatu sistem mempunyai tujuan atau sasaran. Sasaran dari sistem yang sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.



Gambar 2.1 Karakteristik Suatu Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

1. sistem alami dan sistem buatan.
2. sistem fisik dan sistem konseptual.
3. sistem terbuka dan sistem tertutup.
4. sistem lingkaran terbuka dan sistem lingkaran tertutup (pengendalian).
5. sistem dinamis dan sistem setatis.

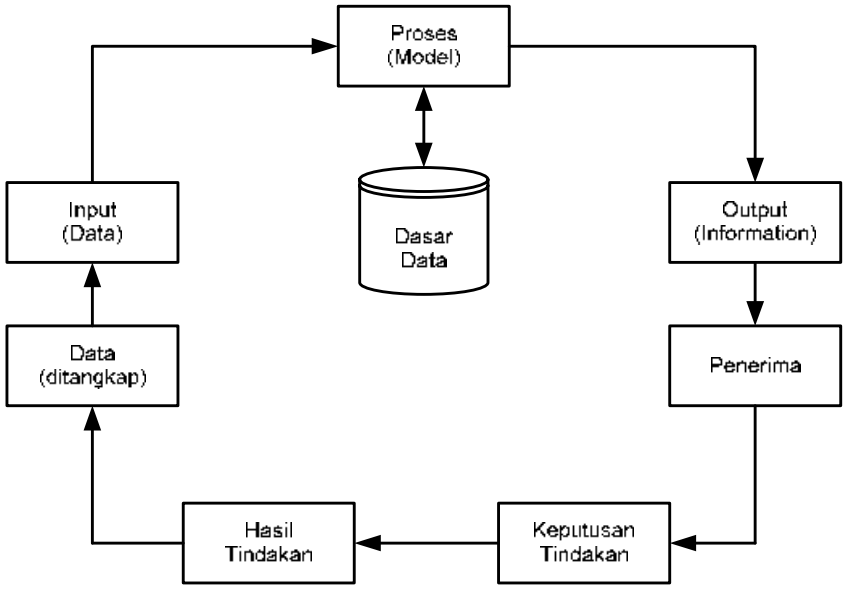
2.1.2 Pengertian Informasi

Informasi adalah data yang telah diolah menjadi suatu bentuk yang lebih penting bagi yang menerimanya dan bermanfaat bagi yang mengambil keputusan saat ini atau yang mendatang. Sumber informasi ini adalah data. Adapun pengertian data adalah fakta atau apapun yang dapat digunakan sebagai input dalam menghasilkan informasi. Saat ini data tidak harus selalu dalam bentuk

kumpulan huruf-huruf, dalam bentuk kata atau kalimat, tapi bisa juga dalam bentuk suara, gambar diam dan bergerak, baik dalam bentuk dua atau tiga dimensi.

2.1.2.1 Siklus Informasi

Informasi merupakan suatu proses perubahan data menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk siklus. Siklus pada gambar 2.1 ini oleh John Burch disebut dengan siklus informasi (*Information Cycle*).



Gambar 2.2 Siklus Informasi

2.1.2.2 Kualitas Informasi

Dalam mendapatkan suatu informasi, informasi tersebut harus yang berkualitas, yaitu yang memiliki ciri-ciri sebagai berikut :

1. Akurat, artinya informasi harus mencerminkan keadaan yang sebenarnya.
2. Tepat waktu, artinya informasi itu harus tersedia atau ada pada saat informasi tersebut diperlukan.
3. *Relevan*, artinya informasi yang diberikan harus sesuai dengan yang dibutuhkan.
4. Lengkap, artinya informasi harus diberikan secara lengkap.

Dari uraian tersebut diatas maka sistem informasi dapat didefinisikan sebagai suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan-kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan.

Kegiatan sistem informasi mencakup :

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

2.2 Alat-Alat Pengembangan Sistem

Untuk dapat melakukan langkah-langkah sesuai dengan yang diberikan oleh metodologi analisis dan desain terstruktur, maka dibutuhkan alat dan teknik untuk melaksanakannya. Alat-alat pengembangan sistem umumnya berupa suatu gambar atau diagram atau grafik.Alat-alat yang dipakai antara lain :

1. Alat berbentuk grafik, yaitu :

   a. Data Flow Diagram (DFD)

   Adalah diagram yang menggunakan arus notasi-notasi tertentu untuk menggambarkan data sistem. *Data flow diagram* digunakan untuk mengambarkan suatu sistem yang telah ada atau baru dikembangkan secara logika tanpa memperdulikan lingkungan *fisik* dimana data tersebut mengalir atau disimpan. *Data flow diagram* merupakan alat yang dapat menggambarkan arus data didalam sistem secara terstruktur dan jelas.

   Didalam DFD terdapat beberapa level, diantaranya adalah DFD level 0 atau disebut dengan Diagram Konteks (*Context Diagram*), yaitu diagram yang digunakan untuk mendefinisikan dan memperlihatkan ruang lingkup atau batasan dari sistem yang ditelaah dan hubungannya dengan sistem lain yang terkait.

   Untuk memudahkan pembacaan DFD, maka penggambaran DFD dibantu dengan beberapa simbol diantaranya :

   1. Kesatuan Luar (*External Entry*)

      Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Namun sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya.



Gambar 2.3 Notasi External Entity

2. Arus data (*Data Flow*)

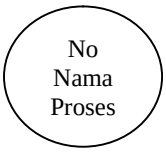
   Suatu data flow atau alur data dapat dipresentasikan dengan anak panah yang menunjukan arah menuju kedalam dan keluar dari suatu proses. Aliran daya ini digunakan untuk menerangkan perpindahan data atau paket data atau informasi dari suatu bagian sistem ke bagian lainnya aliran data dapat disimbolkan dengan panah.



Gambar 2.4 Notasi *Data Flow*

3. Proses (*Process*)

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



Gambar 2.5 Notasi *Process*

4. Penyimpanan data (*Data Store*)

   Penyimpanan data (*data store*) merupakan simpanan dari atau yang dapat berupa sebagai berikut ini :

   a. suatu file / *database* di dalam sistem komputer
   b. suatu arsip atau catatan manual
   c. suatu agenda atau buku



Gambar 2.6 Notasi *Data Store*

b. Entity Relationship Diagram (ERD)

Adalah relasi antara file atau tabel atau lebih yang saling berkaitan, yang direlasikan dengan kunci (*relation key*), kunci relasi merupakan kunci utama dari masing-masing file.

Entity Relationship Diagram merupakan suatu model jaringan (*network*) yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Tujuan utama penggambaran ERD adalah untuk menunjukan struktur objek data *(entity)* dan hubungan *(Relationship)* yang menunjukan struktur objek data (*entity*) dan hubungan yang ada pada objek tersebut.

Komponen – komponen pembentuk model ERD ini adalah entitas *(entity)* dan relasi *(relation).* Dan berikut ini adalah penjelasan dari komponen pembentuk ERD :

1. Entity

   Entity adalah sesuatu yang dapat diberikan dalam dunia nyata dengan keberadaan yang bebas baik secara fisik maupun secara abstrak (konsep), mempunyai karakteristik tertentu, dimana informasi yang berkaitan dengannya dikumpulkan. Tipe entitas adalah sekumpulan entitas yang menggunakan sifat dan karakteristik yang sama, sedangkan instance entitas adalah empat persegi panjang serta pemberian nama biasanya dengan menggunakan kata benda.

2. Relasi

   “Relasi adalah hubungan yang terjadi antara *instance* dari satu atau lebih tipe entitas. *Relationship* tidak mempunyai keberadaan fisik kecuali yang mewarisi dari hubungan entitas tersebut. *Relationship set* adalah kumpulan *relatioanship* yang sejenis. Simbol yang menggunakan kata kerja.

3. Kardinalitas Pemetaan (*mapping constraint*)

   Kardinalitas pemetaan atau rasio kardinalitas menunjukkan jumlah *entity* yang dapat dihubungkan ke satu *entity* lain dengan suatu *relationship sets*.

   Kardinalitas pemetaan meliputi :

   a. Hubungan satu ke satu (*one to one*)

      Yaitu satu *entity* dalam A dihubungkan dengan maksimum satu *entity* dalam B.

   b. Hubungan satu ke banyak (*one to many*)

      Yaitu satu *entity* dalam A dihubungkan dengan sejumlah *entity* dalam B. Satu *entity* dalam B dihubungkan dengan maksimum satu *entity* dalam A.

   c. Hubungan banyak ke satu (*many to one*)

      Yaitu satu *entity* dalam A dihubungkan dengan maksimum satu *entity* dalam B. Satu *entity* dalam B dapat dihubungkan dengan sejumlah *entity* dalam A.

   d. Hubungan banyak ke banyak (*many to many*)

Satu *entity* dalam A dihubungkan dengan sejumlah *entity* dalam B.

2. Alat berbentuk bagan, yaitu

   a. *Organization Chart* (Bagan Organisasi)

      Adalah bagan yang menujukan bagaimana departemen-departemen didalam suatu organisasi dikoordinasikan secara bersama-sama melalui suatu jalur dan bertanggung jawab.

   b. *Flowchart Document* (Bagan Air Dokumen)

      Merupakan bagan yang menunjukan arus dari laporan dan formulir termasuk tembusan-tembusan yang dikeluarkan dan hasil bagan-bagan tertentu dalam sistem.

   c. *Relation Database* (Bagan Hubungan Database)

      Adalah bagan yang menggambarkan hubungan atau relasi dari file-file *database* yang digunakan dalam sistem.

   d. *Layout Chart* (Bagan Tata Letak Layar)

      Adalah bagan yang menggambarkan bentuk input dan output pada layar terminal monitor (*screen*).

3. Alat yang bukan berbentuk gambar atau grafik, yaitu :

   a. *Data Dictionary* (Kamus Data)

      Kamus data atau *data dictionary* disebut juga dengan istilah sistem dari *dictionary* adalah katalog fakta tentang data dan kebutuhan–kebutuhan dari suatu sistem informasi dengan menggunakan kamus data analis sistem cepat mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data dibuat pada sikap analis sistem dan digunakan pada setiap analis maupun pada tahap perencanaan sistem.

"Kamus data adalah dokumentasi yang mendukung data flow diagram terdiri dari definisi dari setiap data flow dan *data store* yang berada pada *data flow diagram* tersebut. Kamus data terdiri dari dua macam yaitu kamus data komposit dan kamus data elementer.

"Kamus data komposit digunakan untuk menjelaskan komposisi dari paket data yang komplek yang dapat dipecah menjadi beberapa item – item elementer".

"Kamus data elementer digunakan untuk menjelaskan data – data elementer, yang tidak dapat dipecah – pecah lagi"

Kamus data mendefinisikan elemen data dengan fungsi sebagai berikut :

1. Menjelaskan arti aliran data dan penyimpanan dalam data flow diagram.
2. Mendefinisikan komponen paket data yang bergerak melalui aliran.
3. Mendeskripsikan komposisi penyimpanan data.

b. *Form* (Formulir)

Adalah formulir yang ada pada sistem yang sedang berjalan atau pada sistem yang dirancang. Formulir ini berisi data item, bentuk, sumber, fungsi, jumlah dan frekuensinya.

2.3 Model Proses Rekayasa Perangkat Lunak

Model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan alat-alat bantu yang akan dipakai, dan kontrol serta penyampaian yang dibutuhkan. Perkembangan perangkat lunak bisa

dianggap sebagai lingkaran pemecahan masalah dimana terdapat empat keadaan berbeda, yaitu status quo, definisi masalah, perkembangan teknis memecahkan masalah di keseluruhan aplikasi dari banyak aplikasi, dan integrasi pemecahan menyampaikan hasil kepada siapa yang membutuhkan pertama kali. Bermacam-macam model proses yang berbeda pada perangkat lunak sebagai berikut:

1. Model Sekuensial Linier atau *Waterfall*

   Sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan.

2. Model Prototipe

   Prototyping paradigm dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut,yang akan nampak bagi pelanggan / pemakai (contohnya pendekatan *input* dan format *output*).

3. Model RAD

   Rapid Application Development (RAD) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier di mana

perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.

4. Model Proses Perangkat Lunak Evolusioner

Model evolusioner adalah model iteratif. Model itu ditandai dengan tingkah laku yang memungkinkan perekayasa perangkat lunak mengembangkan versi perangkat lunak yang lebih lengkap sedikit demi sedikit.

a. Model Pertambahan

Model inkeremental menggabungkan elemen-elemen model sekuensial linier (diaplikasikan secara berulang) dengan filosofi prototipe iteratif. Model pertambahan memakai urutan-urutan linier di dalam model yang membingungkan, seiring dengan laju waktu kalender. Model pertambahan berfokus pada penyampaian produk operasional dalam setiap pertambahannya.

b. Model Spiral

Model spiral (*spiral model*) adalah model proses perangkat lunak yang evolusioner yang merangkai sifat iteratif dari prototipe dengan cara komputer dan aspek sistematis dari model sekuensial linier.

c. Model Rakitan Komponen

Model ratikan komponen menggabungkan beberapa karakteristik model spiral. Model ini bersifat evolusioner, sehingga membutuhkan pendekatan iteratif untuk mencapai perangkat lunak. Tetapi model rakitan komponen merangkai aplikasi dari komponen perangkat lunak sebelum dipaketkan (kadang-kadang disebut “kelas”).

d. Model Perkembangan Konkuren

Model proses yang konkuren dapat disajikan secara skematis sebagai sederetan aktivitas teknik mayor, tugas-tugas, dan keadaannya yang lain. Contohnya aktivitas rekayasa yang dibatasi untuk model spiral dipenuhi dengan melakukan tugas-tugas sebagai berikut: *prototyping* dan atau pemodelan analisis, spesifikasi kebutuhan, dan rancangan.

5. Model Formal

Model metode formal mencakup sekumpulan aktivitas yang membawa kepada spesifikasi matematis perangkat lunak komputer. Metode formal memungkinkan perekayasa perangkat lunak untuk mengkhususkan, mengembangkan, dan memverifikasi sistem berbasis komputer dengan menggunakan notasi matematis yang tepat.

2.4 Pengertian *Database* (Basis Data)

Database adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk suatu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. *Database* merupakan salah satu komponen penting dalam sistem informasi, karena *database* adalah dasar untuk menyediakan informasi bagi para pemakai.

Adapun komponen-komponen basis data adalah sebagai berikut:

a. Perangkat keras (*Hadware*)
b. Perangkat lunak (*Software*)
c. Sistem operasi (*Operating System*)
d. Basis data (*Database*)

e. Pengelola basis data (DBMS)

f. Pemakai (*User*)

2.4.1 Tujuan Basis Data

Tujuan basis data erat hubungannya dengan masalah-masalah yang timbul dalam file basis data, yaitu :

1. Penyediaan akses yang *fleksibel*, yaitu untuk memberikan kemudahan dalam menampilkan semua informasi yang diperlukan.
2. Pemeliharaan integritas data.
3. Keamanan data.
4. Menghilangkan ketergantungan data.
5. Mengurangi kerangkapan data.
6. Penggunaan data bersama-sama.
7. Standarisasi data, yaitu memberikan fasilitas-fasilitas kamus data untuk mendefinisikan nama-nama data secara rinci.

2.4.2 Teknik Normalisasi

Didalam proses perancangan *database*, untuk menghindari database dari redudansi dan inkonsistensi data, duplikasi dan sejenisnya, maka dilakukan teknik normalisasi. Ini merupakan salah satu pendekatan (*approach*) model konseptual didalam proses perancangan database yang efektif dan efisien. Teknik konseptual dalam hal ini teknik normalisasi merupakan proses pengelompokan data elemen menjadi tabel yang menunjukan entity dan reaksinya. Sebelum mengenal teknik

normalisasi lebih jauh, ada beberapa hal atau konsep yang harus diketahui lebih dahulu, yaitu :

a. *Field atribute* kunci

Setiap file selalu terdapat kunci atau *key* dari *file* berupa satu *field* atau satu set *field* yang dapat mewakili *record*. Misalnya NIS merupakan kunci dari *file* siswa/i. Didalam *field* atau atribut ini ada juga yang disebut dengan kunci utama (*primary key*), *candidate key, alternate key* dan *foreign key*.

1. *Candidate key*

   Merupakan satu atribut atau satu set minimal atribut yang mengidentifikasikan secara unik suatu kejadian-kejadian spesifik dari suatu entity.

2. *Primary key*

   Merupakan satu atribut yang tidak hanya mengidentifikasikan secara unik satu kejadian spesifik tapi juga dapat mewakili setiap kejadian dari suatu entity.

3. *Alternate key*

   Merupakan kunci kandidat yang tidak dipakai sebagai *primary key*.

4. *Foreign key*

   Merupakan salah satu atribut yang melengkapi satu *relationship* yang menunjukan ke induknya.

b. Kebergantungan fungsi (*functional dependency*)

Didalam *database* kita sering menjumpai kebergantungan, fungsi ini merupakan salah satu yang harus diperhatikan dalam membuat aplikasi

database. Bentuk skematika dari kebergantungan fungsi ini adalah sebagai berikut :

Nama Siswa/i ⟶ NIS atau Nama Siswa/i = f [NIS]

Artinya bahwa nama siswa/i tergantung pada NIS. Jadi bisa dikatakan atribut nama siswa/i bergantung fungsi pada NIS.

2.5 Borland Delphi 7.0.

Delphi adalah suatu program berbasis bahasa *Pascal* yang berjalan dalam lingkungan *Windows*. Delphi telah memanfaatkan suatu teknik pemrograman yang disebut RAD yang telah membuat pemrograman menjadi lebih mudah. Delphi adalah suatu bahasa pemrograman yang telah memanfaatkan metode pemrograman *Object Oriented Programming* (OOP).

Fungsi dari elemen-elemen pada delphi adalah :

1. *Object Inspector* : suatu window yang berguna untuk mengatur suatu object baik properti, *events* dan *method*.
2. *Form* : Digunakan sebagai layar/window yang digunakan sebagai lembar kerja kita. Di form-lah semua komponen seperti tombol dan komponen lainnya disimpan.
3. *Window Unit*/*Source Code* : Window/layar yang berisi perintah-perintah yang akan dieksekusi oleh komputer. Di layar inilah kita mengisikan program-program.
4. *Component Palette* : Layar yang berisikan komponen-komponen yang dipakai dipakai dalam program kita.

Perbedaan konsep pemrograman berbasis objek dengan konsep pemrograman biasa adalah sebagai berikut :

1. Dalam pemrograman biasa, suatu benda hanya memiliki properti (ciri) yang membentuk dirinya.

   Contoh : Objek manusia memiliki nama, tinggi, berat, warna kulit dan lain-lain.

2. Dalam pemrograman berorientasi objek, suatu benda tidak hanya memiliki properti (ciri) tetapi juga memiliki method dan event.

   Contoh: Objek manusia tidak hanya memiliki ***properti*** nama, tinggi, berat, warna kulit tetapi juga mempunyai metode / langkah yang dapat dikerjakan dan juga event (suatu kejadian yang bisa terjadi pada objek).

   a. ***Event*** dapat berupa : Event ketika lapar, ketika haus, ketika ngantuk dan lain-lain.
   b. ***Method*** dapat berupa : Makan, minum, tidur.

Suatu *method* bisa saja dipanggil ketika suatu *event* terjadi. Contoh ketika manusia lapar, maka dia akan melakukan method makan. Contoh nyata dalam pemrograman adalah sebagai berikut : Misalkan program yang kita buat memiliki sebuah *form* dan sebuah tombol. Kita dapat memberikan suatu perintah ke masing-masing *event* dari kedua objek tersebut. Misalkan ketika *form* diklik, maka warna dari *form* berganti menjadi merah, dan ketika tombol ditekan maka program akan ditutup.

2.6 MySQL

MySQL adalah bahasa komputer standar ANSI (*American National Standard Institute*) untuk mengakses dan memanipulasi sistem basisdata. Pernyataan SQL digunakan untuk mengambil dan meng-update data dalam basisdata.

SQL adalah kependekan *Structured Query Language* yang merupakan bahasa basisdata standar industri. SQL menyediakan Data Definition Language (DDL) dan *Data Manipulation Language* (DML). Walaupun dalam beberapa bagian saling berkaitan, perintah DDL memungkinkan dapat membuat dan mendefinisikan tabel (CREATE TABLE), dan indeks (CREATE INDEKS). Sebaliknya, perintah DML memungkinkan dapat memanipulasi data dan membangun query untuk mengambil data (SELECT) bebrapa tabel, menyisipkan (INSERT) data baru, memperbaiki (UPDATE) data, dan menghapus (DELETE) data.

SQL akan dikelompokan menjadi tiga :

1. Query untuk mengelola *database*
2. Query untuk mengakses data dalam satu table
3. Query yang melibatkan lebih dari satu table

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang *software* dan konsultan *database* bernama MySQL AB yang bertempat di Swedia. Waktu itu perusahaan tersebut masih bernama TcX Data Konsult AB, dan tujuan awal dikembangkannya MySQL adalah untuk mengembangkan aplikasi berbasis web pada *client*.

Awalnya Micheal Widenius “Monty”, pengembang satu-satunya TcX, memiliki sebuah aplikasi UNIREG dan rutin ISAM buatannya sendiri dan sedang mencari antarmuka SQL yang cocok untuk diimplementasikan kedalamnya. Mula-mula Monty memakai miniSQL (mSQL) pada percobaannya itu, tetapi mSQL dirasa kurang sesuai, karena terlalu lambat dalam pemrosesan *query* data.

Akhirnya Monty menghubungi David Hughes, pembuat mSQL yang sedang merilis versi kedua dari mSQL. Kemudian Monty mencoba membuat sendiri mesin SQL yang memiliki antarmuka mirip dengan SQL, tetapi dengan kemampuan yang lebih sesuai, dan lahirlah MySQL.

2.7 Konsep Dasar *Client/Server*

Sistem *Client* / *server* mempunyai dua komponen utama, yaitu komputer *client* dan komputer *server*. *Server* merupakan komputer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintan-permintaan dari komputer *client* dan bertindak sebagai *sever database* yang menyimpan data. *Client* yaitu komputer / w*orkstation* yang melakukan pengiriman permintaan-permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya.

Sistem *client* / *server* merupakan suatu sistem komputer yang melibatkan proses-proses *client* yang meminta suatu pelayanan data kepada komputer *server* yang menyediakan layanan data tersebut, sehingga baik *client* maupun *server* sama-sama melakukan suatu pekerjaan. Dengan adanya kombinasi *client* dan *server* ini maka kumpulan dari modul-modul program tidak dieksekusi dalam memori yang sama namun terbagi dalam komputer *client* dan *server*. Dengan

arsitektur seperti ini, modul yang memanggil menjadi *client* yang meminta suatu layanan data dan modul yang dipanggil mencari *server* yang menyediakan layanan tersebut.

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.