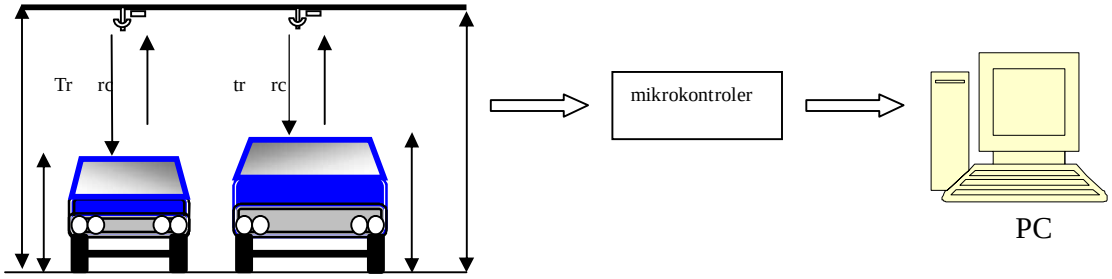
BAB II
LANDASAN TEORI

2.1. Dasar Teori Perangkat Keras

Pengontrolan lahan parkir yang belum terisi oleh kendaraan dapat di lakukan dengan menempatkan sensor di tempat parkir, tepat di atas lahan yang akan di gunakan mobil untuk parkir. Kendaraan yang berada tepat di tempat parkir tersebut akan dapat terdeteksi oleh sensor yang akan dihubungkan ke Komputer (PC).



Gambar 2.1. Diagram Perangkat Keras

Dalam perangkat kerasnya menggunakan beberapa alat yaitu :

1. **Sensor *InfraRed***

   Sensor *InfraRed* digunakan untuk mendeteksi apakah ada kendaraan (mobil). Dimana dalam satu lokasi parkir akan terdapat dua sensor *InfraRed*, pertama t*ransmitter* yaitu sensor yang berfungsi untuk mengirim sinyal, dan yang kedua adalah *receiver* yaitu sensor yang berfungsi untuk menerima sinyal pantulan dari atap mobil.

2. ***Mikrokontroler***

   Penggunaan *mikrokontroler* ini memiliki peranan yang cukup penting dalam perancangan sistem ini, karena digunakan untuk membaca sinyal yang dihasilkan oleh sensor *InfraRed* dan mengirimkan ke PC untuk di tampilkan dalam *database*

3. **RS485**

   Merupakan alat komunikasi yang menghubungkan antara 2 buah mikrokontrol dengan RS232 yang nantinya akan tersambung ke PC

4. **RS232**

   Saluran Komunikasi serial yang digunakan untuk menghubungkan antara alat dengan PC.

5. Personal Computer (PC)

Fungsi dari PC (*Personal Computer*) yaitu untuk menampilkan informasi yang dikirimkan dari sensor dan ditampilkan dalam bentuk simulasi denah 2 lantai, dimana tiap lantainya terdiri dari 32 lokasi parkir

2.2. Dasar teori Perangkat Lunak

2.2.1 Pengertian Sistem

Suatu sistem dapat didefinisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan.

Pada dasarnya hanya ada dua jenis sistem, yaitu :

a. Sistem alami seperti matahari, sistem luar angkasa dan lain sebagainya.
b. Sistem buatan manusia seperti organisasi (perpustakaan), sistem komunikasi (telepon, teleks), sistem keuangan (akuntansi, inventori, buku besar) dan lain sebagainya.

Salah satu bagian sistem buatan manusia adalah sistem terotomatisasi yang berinteraksi atau dikontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern.

Sistem terotomatisasi mempunyai sejumlah komponen, yaitu :

a. Perangkat keras, antara lain CPU, disk, terminal, printer.
b. Perangkat lunak, antara lain sistem operasi, sistem *database*, program pengontrol komunikasi, dan program aplikasi.
c. Personil, antara lain yang mengoprasikan sistem, menyediakan masukan (input), mengkonsumsi keluaran (*output*), dan melakukan aktivitas manual yang mendukung sistem.
d. Data, antara lain yang harus tersimpan dalam sistem selama jangka waktu tertentu.
e. Prosedur, antara lain intruksi dan kebijakan untuk mengoprasikan sistem.

Sistem terotomasi terbagi dalam sejumlah kategori, yaitu :

a. *On-line systems*

   Sistem on-line adalah sistem yang menerima langsung input pada area dimana input tersebut direkam, dan menghasilkan output yang dapat berupa hasil komputasi pada area, yang dibutuhkan. Area sendiri terpisah-pisah dalam skala, misalnya ratusan kilometer. Sistem ini biasanya digunakan bagi *reservasi* angkutan udara, *reservasi* kereta api, perbankan dan lain-lain.

b. *Real-time systems*

Sistem real-time adalah mekanisme pengontrolan, perekaman data, pemrosesan yang sangat cepat sehingga output yang dihasilkan dapat diterima dalam waktu yang relatif sama. Perbedaannya dengan sistem *online* adalah satuan waktu yang digunakan *real-time* biasanya seperatus atau seperibu detik sedangkan *online* masih dalam sekala detik atau bahkan menit. Perbedaan lainnya, *online* biasanya hanya berinteraksi dengan pemakai, sedangkan *real-time* berinteraksi langsung dengan pemakai dan lingkungan yang dipetakan.

c. *Decision support system + strategic planing systems*

Sistem yang memproses transaksi organisasi secara harian, dan membantu para manajer mengambil keputusan, mengevaluasi dan menganalisa tujuan organisasi. Biasanya berbentuk paket statistik, paket pemasaran dan lain-lain. Sistem ini tidak hanya merekam dan menampilkan data tapi juga fungsi matematik, data analisa statistik dan menampilkan informasi dalam bentuk grafik (tabel, chart) sebagai laporan konvensional.

d. *Knowledge-based systems*

Program komputer yang dibuat mendekati kemampuan dan pengetahuan seorang pakar umumnya menggunakan perangkat keras dan perangkat lunak khusus seperti LISP dan PROLOG.

2.2.2 Pengertian Informasi

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan [2].

Kualitas dari suatu informasi tergantung dari tiga hal yaitu :

a. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan.
b. Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi.
c. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya.

2.2.3. Pengertian Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dalam menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik.

Sistem informasi dapat terdiri dari komponen-komponen yang disebut dengan blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok dasar data (*database block*) dan blok kendali (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainya membentuk suatu kesatuan untuk mencapai tujuannya.

2.3. Analisis Perancangan Sistem

2.3.1. Analisis Sistem

Analisis sistem bekerja dalam pengembangan sistem dengan karakteristik personil yang bervariasi. Karena itu ketika analisa sistem dilakukan, dibutuhkan pengetahuan yang lebih dari sekedar pemahaman tentang teknologi komputer misalnya kemampuan berkomunikasi dengan orang lain.

Pelaku sistem terdiri dari tujuh kelompok [2], yaitu :

1. Pemakai

   Pemakai adalah pelaku terpenting karena sistem diciptakan untuk pemakai dan melalui komunikasi dengan pemakailah maka sistem dibuat dan dirancang hingga mencapai bentuk akhir.

2. Manajemen

   Umumnya terdiri dari tiga jenis manajemen yaitu manajemen pemakai yang bertugas menangani pemakaian dimana sistem baru ditetapkan, manajemen sistem yang terlibat dalam pengembangan sistem itu sendiri dan manajemen umum yang terlibat dalam strategi perancangan sistem dan sistem pendukung pengambilan keputusan. Kelompok manajemen biasanya terlibat dengan keputusan yang berhubungan dengan orang, waktu dan uang.

3. Pemeriksa

   Ukuran dan kerumitan sistem yang dikerjakan dan bentuk alami organisasi dimana sistem tersebut diimpementasikan dapat menentukan kesimpulan perlu tidaknya pemeriksaan. Pemeriksa biasanya menentukan segala sesuatunya berdasarkan ukuran-ukuran standar yang dikembangkan pada banyak perusahaan sejenis.

4. Penganalisa Sistem

   Fungsi-fungsinya adalah sebagai berikut :

   a. *Arkeolog*, yaitu yang menelusuri bagaimana sebenarnya sistem berjalan, bagaimana sistem tersebut dijalankan, dan segala hal yang menyangkut sistem lama.
   b. *Inovator*, yaitu yang mengembangkan dan membuka wawasan pemakai bagi kemungkunan-kemungkinan yang lain.
   c. *Mediator*, yaitu yang menjalankan fungsi komunikasi semua level, antara lain pemakai, manajer, programmer, pemeriksa, dan pelaku sistem lainnya yang mungkin belum punya sikap dan cara pandang yang sama.
   d. Pimpinan proyek, penganalisa sistem haruslah personil yang lebih berpengalaman dari *disainer* dan *programmer*.

5. Pendisain sistem

   Pendisain sistem menerima hasil penganalisa sistem berupa kebutuhan pemakai yang tidak berorientasi pada teknologi tertentu yang kemudian ditransformasikan ke disain arsitektur tingkat tinggi dan dapat diformulasikan oleh *programmer*.

6. *Programmer*

   Setelah penganalisa sistem memberikan hasil kerjanya dan kemudian diolah oleh pendisain sistem baru *programmer* dapat mulai bekerja. Karena itu *programmer* baru mulai bekerja setelah penganalisa sistem selesai dengan pekerjaannya.

7. Personil Pengoperasian

   Pelaku ini bertugas dan bertanggung jawab di pusat komputer misalnya jaringan, keamanan perangkat keras, keamanan perangkat lunak, dan *back-up*. Kadang-kadang pelaku sistem ini memang tidak diperlukan karena sistem yang berjalan mungkin tidak cukup besar dan tidak membutuhkan klasifikasi khusus untuk menjalankan sistem.

2.3.2. Perancangan Sistem

Pada dunia permodelan sistem terdapat sejumlah cara yang digunakan untuk mempresentasikan sistem, antara lain :

1. Statment of Purpose (STP)

Model pertama adalah *statement of purpose* (STP), yang berisi deskripsi tekstual fungsi sistem. Dalam hal ini berguna hampir semua level antara lain level puncak, level pemakai, dan level lain yang tidak terlibat secara langsung dalam pengembangan sistem.

2. Diagram Conteks (Conteks diagram)

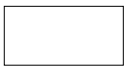
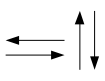
Conteks diagram adalah diagram yang menggambarkan hubungan antara sistem dengan entitas dari luar . Diagram konteks berfungsi sebagai transformasi dari suatu proses yang melakukan transformasi data input menjadi data *output.*

Conteks diagram menyoroti sejumlah karakteristik penting dalam sistem, antara lain :

a. Kelompok pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi yang disebut juga sebagai *terminator*.
b. Data masuk, data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
c. Data keluar, data yang dihasilkan sistem dan diberikan ke dunia luar.
d. Penyimpan data (*data store*) yang digunakan secara bersamaan antara sistem kita dengan *terminator*. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkunganatau sebaliknya, dibuat oleh lingkungan dan digunakan oleh sistem.
e. Batasan antara sistem kita dan lingkungan.

Berikut ini merupakan komponen-komponen dalam konteks diagram:

Tabel 2.1. komponen Pembentuk Konteks Diagram

No.	Simbol	Nama	Keterangan
1.		Menunjukan Entitas luar sistem	Menunjukan entitas luar sistem yang menghubungkan data dengan sistem
2.		Simbol Proses	Menunjukan adanya kegiatan proses dalam sistem
3.		Aliran data atau kontrol	Menunjukan arah aliran data atau kontrol.

4.	_____ _____	Data store	Menunjukan media penyimpanan data yang dapat berupa suatu file/tabel di sistem computer.

3. *Even List* (EL)

Even List (EL) adalah daftar narasi stimuli (daftar kejadian) yang terjadi dalam lingkungan dan mempunyai hubungan dengan respon yang diberikan sistem.

Aturan-aturan EL antara lain daftar kejadian yang kita buat dan digambarkan dalam bentuk tekstual sederhana yang berfungsi memodelkan kejadian dalam lingkungan dimana sistem harus memberikan respon. Ketika membuat EL maka kita harus yakin perbedaan kejadian (*event*) dan kejadian yang berelasi dengan aliran.

4. Data flow Diagram Levelled (DFD)

DFD adalah suatu jaringan sistem , DFD sering dipakai untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut disimpan atau DFD merupakan alat dimana data tersebut disimpan

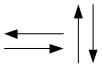
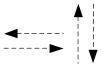
Model ini menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran penyimpanan data.

Terdapat empat komponen dalam model ini, yaitu :

a. Proses, menunjukan transformasi dari masukan menjadi keluaran, dalam hal ini sejumlah masukan dapat menjadi hanya satu keluaran ataupun sebaliknya. Proses direpresentasikan dalam bentuk lingkaran, umumnya didefinisikan dengan kata tunggal atau kalimat sederhana.
b. Aliran, komponen ini direpresentasikan dengan menggunakan panah yang menuju ke/dari proses. Digunakan untuk menggambarkan gerakan paket data atau informasi dari suatu bagian ke bagian yang lain dari sistem dimana penyimpanan mewakili lokasi penyimpanan data.
c. Penyimpanan, komponen ini digunakan untuk memodelkan kumpulan data atau paket data. Notasi yang digunakan adalah garis sejajar, segiempat dengan sudut melengkung, atau persegi panjang.

d. Terminator, direpresentasikan menggunakan persegi panjang, yang mewakili eniti luar dimana sistem berkomunikasi. Biasanya notasi ini melambangkan orang atau kelompok orang misalnya organisasi diluar sistem.

Tabel 2.2 Kompunen– komponen pembentuk DFD

No.	Simbol	Nama	Keterangan
1.		Menunjukan Entitas	Menunjukan bagian yang melakukan proses
2.		Simbol Proses	Menunjukan adanya kegiatan proses dalam sistem
3.		Aliran data atau kontrol	Menunjukan arah aliran data atau kontrol.
4.		Data store	Menunjukan media penyimpanan data yang dapat berupa suatu file/tabel di sistem computer.
5.		Aliran data berupa sinyal	Menunjukan arah aliran data yang berupa sinyal digital/analog.

5. Data dictionary (DD)

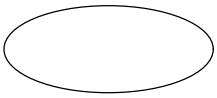
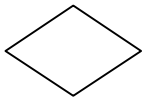
Data dictionary atau kamus data tidak menggunakan notasi grafis sebagaimana halnya DFD. Kamus data berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detail dam mereorganisasi semua elemen data yang digunakan dalam sistem secara presisi sehingga pemakai dan penganalisa sistem punya dasar pengertian yang sama tentang masukan, keluaran, penyimpanan, dan proses.

6. Entity-Relationship Diagram (ERD)

ERD adalah model konseptual yang mendeskripsikan hubungan antara penyimpanan (dalam DFD) ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada tiga macam simbol yang digunakan yaitu :

a. *Entity : Entity* adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang dibuat. *Enitiy* digambarkan menggunakan persegi empat.
b. Atribut : berfungsi mendeskripsikan karakter *enitity*. Atribut diawali oleh simbol elips.
c. Hubungan (relasi) : Pada suatu hubungan tidak masalah berapapun derajat hubungannya, antar *enitity* ada tiga jenis hubungan biner seperti satu ke satu, satu ke banyak, atau banyak ke satu dan banyak ke banyak. Dalam ERD ada *enitity* yang disebut *enitiy* lemah, yaitu *enitiy* yang kehadirannya dalam suatu basis data tergantung pada kehadiran *enitity* lain.

Tabel 2.3 Kompunen– komponen pembentuk ER-D

No.	Simbol	Nama	Keterangan
1.		Menunjukan Himpunan Entitas	Menunjukan bagian yang melakukan proses
2.		Simbol elips	Menunjukan sebagai atribut (atribut yang berfungsi sebagai key
3.		Garis	Sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya
4.		Belah Ketupat	Menyatakan Himpunan relasi.

Langkah- langkah teknis yang dapat dilakukan untuk membentuk diagram E-R menurut [1] adalah:

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
2. Menentukan atribut-atribut *key* yang akan terlibat.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan-himpunan entitas yang ada

4. Menentukan derajat kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non key*)

7. State Transition Diagram (STD)

STD berfungsi untuk model tingkah laku sebuah sistem. Komponen utama diagram adalah keadaan (*state*) dan panah (*arrow*) yang merepresentasikan perubahan keadaaan.

State dinotasikan dengan persegi panjang yang merepresentasikan keadaan atau lebih tepatnya diasumsikan sebagai kumpulan atribut yang menggambarkan sesuatu pada suatu saat (kondisi) sistem pada saat tertentu. Pada sejumlah sistem dengan keadaan akhir tidak ada, maka sistem akan terus bekerja dan tetap bekerja sampai ada aksi yang tidak didefinisikan dalam sistem untuk menghentikan proses. Tapi pada umumnya sistem selalu punya keadaan akhir dan keadaan awal.

Kondisi awal umumnya selalu digambarkan pada bagian atas diagram yang didefinisikan dari panah awal yang menuju keadaan awal. Kondisi akhir digambarkan dengan tanda panah yang menuju ke satu keadaan akhir pada bagian bawah diagram dan tidak selalu keadaan paling bawah menjadi kondisi akhir suatu sistem. Secara umum hanya ada satu kondisi akhir yang masing-masing eksklusif.

Untuk melengkapi STD, kita membutuhkan dua hal yaitu pertama, sebagai penyebab perubah keadaan; dan yang kedua,aksi yang harus dilakukan ketika akan berubah keadaan.

2.4. Sistem Basis Data

2.4.1. Pengertian Basis Data (*Database)*

Basis data atau *database* adalah merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Penerapan *database* dalam sistem informasi disebut *database system*.

Sistem basis data (*database system*) adalah suatu sistem informasi yng mengintegrasikan kumpulan dari data-data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia beberapa aplikasi yang bermacam-macam di dalam suatu organisasi.

Karakter-karakter dari basis data merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (*special characters*) yang membentuk suatu item data antara lain :

a. *Field*; menggambarkan suatu atribut dari *record* yang menunjukan suatu item dari data.
b. *Record*; merupakan kumpulan dari *field-field*. *Record* menggambarkan suatu unit data individu tertentu.
c. *File*; terdiri dari *record-record* yang menggambarkan suatu kesatuan data yang sejenis.
d. *Database*; merupakan kumpula dari *file-file*.

Sistem basis data merupakan sistem yang terdiri atas kumpulan *file* (tabel) yng saling berhubungan (dalam sebuah basis data di sebuah komputer) dan sekumpulan program (*Database Management System*) yang memungkinkan beberapa pemakai atau program lain untuk mengakses memanipulasi *file-file* (tebel-tabel) tersebut.

Dalam sebuah sistem basis data terdapat komponen-komponen utama sebagai berikut :

1. Perangkat Keras (*Hardware*)

Perangkat Keras yang biasanya terdapat dalam sebuah sistem basis data adalah:

a. Komputer (satu untuk sistem yang stand-alone atau lebih dari satu untuk sistem jaringan).
b. Memori sekunder yang *on-line (Harddisk*).
c. Memori sekunder yang *off-line* (*Tape* atau *Removable Disk* untuk keperluan *backup data*.
d. Media/perangkat komunikasi (untuk sistem jaringan).

2. Sistem Operasi (*Operating System*)

Secara sederhana, Sistem Operasi merupakan program yang mengaktifkan/memfungsikan sistem komputer, mengendalikan seluruh sumber daya (resource) dalam komputer dam malakukan operasi-operasi dasar dalam komputer (operasi I/O, pengelolaan file, dan lain-lain). Sejumlah Sistem Operasi yang banyak digunakan seperti: MS-DOS, MS-Windows 3.1, MS-Windows 95 (untuk komputer stand-alone atau untuk komputer client dalam sistem jaringan) atau Novel-Netware, MS-Windows NT, Uniks dan Sun-Solaris (untuk komputer *server* dalam sistem jaringan). Program pengelola basis data hanya dapat aktif (*running*) jika Sistem Operasi yang dikehendakinya (sesuai) telah aktif.

3. Basis Data (*Database*)

Sebuah sistem basis data dapat memilki beberapa basis data. Setiap basis data dapat berisi selumlah objek basis data (seperti file, tabel, indeks, dan lain-lain). Di samping berisi/menyimpan setiap basia data juga mengandung atau menyimpan definisi struktur (baik untuk basis data maupun objek-objeknya secara detail).

4. Sistem Pengelola Basis Data (*Database Management System*/DBMS)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah Perangkat Lunak (Sistem) yang khusus. Perangkat lunak inilah (disebut DBMS) yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan/konsistensi data, dan sebagainya.

Perangkat lunak yang termasuk DBMS seperti dBase III+, dBase IV, FoxBase, Rbase, MS-Access dan Borland-Paradox (untuk kelas sederhana) atau Borland-Interbase, MS-SQLServer, CA-Open Ingres, Oracle, Informix dan Sybase (untuk kelas kompleks/berat).

5. Pemakai (*User*)

Ada beberapa jenis/tipe pemakai terhadap suatu sistem basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem:

a. Programmer Aplikasi

   Pemakai yang berinteraksi dengan basis data melalui *Data Manipulation language* (DML), yang disertakan (embedded) dalam program yang ditulis dalam bahasa pemprograman induk (seperti C, Pascal, Cobol, dan lain-lain).

b. User mahir (*Casual User*)

   Pemakai yang berinteraksi dengan sistem tanpa menulis modul program. Mereka menyatakan *query* (untuk akses data) dengan bahasa *query* yang telah disediakan oleh suatu DBMS.

c. User Umum (*End User Naive User*)

   Pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanen (*executable* program) yang telah ditulis sebelumnya.

d. User Khusus (*Specialized User*)

   Pemakai yang menulis aplikasi basis data non konvensional tetapi untuk keperluan-keperluan khusus, seperti untuk aplikasi AI, Sisitem Pakar,

pengolahan Citra, dan lain-lain, yang bisa saja mengakses basis data dengan atau tanpa DBMS yang bersangkutan.

Untuk sebuah basis data yang *stand-alone*, maka pada suatu saat hanya ada satu pemakai yang hanya dapat bekerja. Sedang untuk sistem basis data dalam jaringan, maka pada suatu saat ada banyak pemakai yang dapat berhubungan (menggunakan) basis data yang sama. Pilihan untuk *stand-alone* atau jaringan (*multiuser*) tergantung pada (ditentukan oleh) kebutuhan pemakai, perangkat keras yang tersedia, sistem operasi, yang digunakan serta DBMS yang dipilih.

2.4.2. Bahasa Basis Data (*Database Language*)

Database manajement system (DBMS) merupakan perantara bagi pemakai dengan basis data dalam *disk*. Cara berinteraksi atau berkomunikasi antar pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa itu dapat kita sebut sebagai Bahasa Basis Data yang terdiri atas sejumlah perintah (*statement*) yang diformulasikan dan dapat diberikan user dan dikenali atau diproses oleh DBMS untuk melakukan suatu pekerjaan tertentu. Contoh-contoh bahasa basis data adalah SQL, dBase, QUEL dan sebagainya. Sebuah Bahasa Basis Data biasanya dapat dipilih ke dalm 2 bentuk menurut [2] yaitu:

1. *Data Definition Language* (DDL)

Struktur atau skema basis data yang menggambarkan desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yang disebut *Data Definition Language* (DDL). Dengan bahasa inilah kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel, dan sebagainya. Hasisl dari kompilasi perintah DDL adalah kumpulan tabel yang disimpan dalam file khusus yang disebut Kamus Data (*Data Dictionary*). Kamus Data merupakan suatu *metadata* (superdata) yaitu data yang mendeskripsikan data sesungguhnya. Kamus Data ini akan selalu diakses dalam suatu operasi basis data sebelum suatu file data yang sesungguhnya diakses.

2. Data Manipulation Language (DML)

Merupakan bentuk Bahasa Basis Data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa:

- Penyisipan atau Penambahan data baru ke suatu basis data.
- Penghapusan data dari suatu basis data.
- Pengubahan data di suatu basis data.

Pada level fisik, kita harus mendefinisikan algoritma yang memungkinkan pengaksesan yang efisien terhadap data. Pada level yang lebih tinggi, yang dipentingkan bukan hanya efisiensi akses, tetapi juga efisiensi interaksi manusia (pemakai) dengan sistem (kemudahan permintaan akses).

Data Manipulation Languge (DML) merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data. Ada 2 jenis DML, yaitu:

a. *Prosedural*, yang mensyaratkan agar pemakai menentukan data apa yang diinginkan serta bagaimana cara mendapatkannya.
b. *Nonprosedural*, yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

2.4.3. Normalisasi

Dalam merancang basis data kita dapat melakukannya dengan :

a. Menetapkan normalisasi terhadap struktur tabel yang telah diketahui.
b. Membuat model *enitity-relationship*

Dalam pendekatan normalisasi, perancang/desainer basis data bertitik tolak dari situasi yang nyata. Ia telah memiliki *item-item* data yang siap ditempatkan dalam baris dan kolom pada tabel-tabel relasional. Demikian juga dengan sejumlah aturan tentang keterhubungan antar item-item data tersebut. Sementara pendekatan kedua, dengan langsung membuat model data lebih tepat dilakukan jika yang telah diketahui baru prinsip-prinsip sistem secara keseluruhan.

Normalisasi, seperti yang telah dikemukakan sebelumnya telah difokuskan pada tinjauan komperhensif terhadap setiap kelompok dan tabel secara individual. Dalam perancangan basis data terdapat istilah baru yaitu *key* dan atribut. Atribut identik dengan pemakaian istilah kolom data. Ada atribut yang dijadikan *key* dan yang lainnya disebut atribut deskriptif. Ada pula atribut yang tergolong atribut sederhana ataupun atribut komposit, dan sebagainya.

Pada dasarnya, *key* adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*row*) dalam tabel secara unik. Artinya, jika suatu atribut dijadikan sebagai *key* , maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut itu.

Ada 3 (tiga) macam *key* yang dapat diteteapkan dalam suatu tabel, yaitu :

a. *Superkey*, merupakan satu atau lebih atribut (kumpulan atribut)yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.
b. *Candidat-key*, merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik. Sebuah *candidat-key* tidak boleh berisi atribut atau kumpulan atribut yang telah menjadi *Superkey* yang lain. Jadi, sebuah *Candidat-key* pastilah *Superkey*, tapi belum tentu sebaliknya.
c. Key primer(*primary-key*) pada sebuah tabel kemungkinan adanya lebih dari satu *Candidat-key*, dimana salah satunya dapat dijadikan sebagai *primary-key*. Pemilihan Primary-key dari sejumlah *candidat-key* umumnya didasari oleh :
    a. *Key* tersebut lebih sering (lebih natural)untuk dijadikan sebuah acuan.
    b. *Key* tersebut lebih ringkas.
    c. Jaminan keunikan *key* tersebut lebih baik.

Contoh:

Penentuan *superkey, candidat-key, primary-key*

Tabel mahasiswa memiliki 4 buah atribut, yaitu nim, nama_mhs, alamat_mhs dan tgl_ lahir

Didalam tabel Mahasiswa, yang dapat menjadi *superkey* adalah:

- (Nim, nama_mhs, alamat_mhs dan tgl_ lahir)
- (Nim, nama_mhs, alamat_mhs)
- (Nim, nama_mhs)
- (Nama_mhs), jika kita bisa menjamin tidak ada nilai yang sama untuk atribut ini.
- (Nim). Merupakan *Superkey*

Didalam tabel Mahasiswa, yang dapat menjadi *candidat-key* adalah:

- (Nim)
- (Nama_mhs), jika kita bisa menjamin tidak ada nilai yang sama untuk atribut ini.

Sedangkan untuk primary-key adalah (Nim)

2.4.5 Operasi Basis Data

Di dalam sebuah *disk*, basis data dapat diciptakan dan dapat pula ditiadakan. Di dalam sebuah disk, kita dapat pula menempatkan beberapa (lebih dari satu) basis data. Sementara dalam sebuah basis data, kita dapat menempatkan satu atau lebih file atau tabel. Pada file atau tabel inilah sesungguhnya data disimpan dan ditempatkan. Setiap basis data umumnya dibuat untuk mewakili sebuah semesta data yang spesifik. Misalnya, ada basis data kepegawaian, basis data akademik, basis data inventori (pergudangan), dan sebagainya. Sementara dalam basis data akademik, misalnya, kita dapat menempatkan file mahasiswa, file mata kuliah, file dosen, file jadual, file kehadiran, file nilai dan seterusnya.

Karena itu, operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data menurut [1] dapat meliputi:

a. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru.
b. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus beserta isinya, jika ada).
c. Pembuatan file atau tabel baru ke suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
d. Pembuatan file atau tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
e. Penambahan atau pengisian data baru ke sebuah tabel di sebuah basis data (insert), yang identik dengan penambahan lembaran arsip ke sebuah map arsip.
f. Pengambilan data dari sebuah tabel (*retrieve/search*), yang identik dengan pencarian lembaran arsip dari sebuah map arsip.
g. Pengubahan data dari sebuah tabel (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
h. Penghapusan data dari sebuah file atau tabel (*delete*), yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

Operasi yang berkenaan dengan pembuatan objek (basis data dan tabel) merupakan opersi awal yang hanya dilakukan sekali dan berlaku seterusnya. Sedang operasi-operasi yang berkaitan dengan isi tabel (data) merupakan operasi rutin yang akan berlangsung berulang-ulang dan karena itu operasi-operasi inilah yang lebih tepat mewakili aktivitas pengelolaan (*management*) dan pengolahan (*processing*) data dalam basis data.

2.4.5 Objektif Database

Telah disebutkan di awal bahwa tujuan awal dan utama dalam pengelolaan data dalam sebuah data adalah agar kita dapat memperoleh menemukan kembali data (yang kita cari) dengan mudah dan cepat. Di samping itu, pemanfaatan basis data untuk pengelolaan data, juga memiliki tujuan-tujuan lain.

Secara lebih lengkap, pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) seperti berikut ini:

a. Kecepatan dan Kemudahan (Speed)

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, daripada jika kita menyimpan data secara manual (non elektronis) atau secara elektronis (tetapi tidak dalam bentk penerapan basis data, misalnya dalam bentuk *spread sheet* atau dokumen teks biasa).

b. Efisiensi Ruang Penyimpanan (Space)

Karena keterkaitan yang erat antar kelompok data dalam sebuah basus data, maka redundansi (pengulangan) dat pasti akan selalu ada. Banyaknya redundansi ini tentu akan memperbesar ruangan penyimpanan (baik di memori utama maupun memori sekunder) yang harus disediakan. Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena kita dapat melakukan penekanan jumlah redundansi data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk *file*) antar kelompok data yang saling berhubungan.

c. Keakuratan (*Accurasy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (*constraint*) tipe data, domain data keunikan data, dan sebagainya, yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

d. Ketersediaan (*Avaibility*)

Pertumbuhan data (baik dari sisi jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu selalu kita gunakan/butuhkan. Karena itu kita dapat memilah adanya data utama/master/referensi, data transaksi, data histori hingga data kadaluarsa. Data yang sudah jarang atau bahkan tidak pernah lagi kita gunakan, dapat kita atur untuk dilepaskan dari sistem basis data yang sedang aktif (menjadi *off-line*) baik dengan

cara penghapusan atau dengan memindahkannya ke media penyimpanannya *off-line* (seperti *removabledisk*, atau *tape*). Di sisi lain, karena kepentingan pemakaian data, sebuah basis data dapat memiliki data yang disebar di banyak lokasi geografis. Data nasabah sebuah bank, misalnya, dipisah-pisah dan disimpan di lokasi yang sesuai dengan keberadaan nasabah. Dengan pemanfaatan teknologi jaringan komputer, data yang berada di sebuah lokasi/cabang, dapat juga diakses (menjadi tersedia/available) bagi lokasi/cabang lain.

e. Kelengkapan (*Completeness*)

Lengkap atau tidaknya data yang kita kelola dalam sebuah basis data bersifat relatif (baik terhadap kebutuhan pemakai maupun terhadap waktu). Bila seorang pemakai sudah menganggap bahwa data yang yang dipelihara sudah lengkap, maka pemakai yang lain belum tentu berpendapat sama. Atau, yang sekarang dianggap sudah lengkap, belum tentu di masa yang akan datang juga demikian. Dalam sebuah basis data, disamping data kita juga harus menyimpan struktur (baik yang mendefinisikan objek-objek dalam basis data maupun definisi detail dari tiap objek, seperti struktur *file*/tabel atau indeks). Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menmbah *record-record* data, tetapi juga dapat melakukan perubahan stuktur dalam basis data, baik dalam bentuk penembahan objek baru (tabel) atau dengan penambahan *field-field* baru dalam suatu tabel.

f. Keamanan *(Security*)

Memang ada sejumlah sistem (aplikasi) pengelola basis data yang tidak menerapkan aspek keamanan dalam penggunaan basis data. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan dengan ketat. Dengan begitu, kita dapat menentukan siapa-siapa (pemakai) yang boleh menggunakan basis data beserta objek-objek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukannya.

2.5. SQL (*Structured Query Language*)

Ada banyak sekali bahasa dalam basis data salah satunya yang akan kita bahas secara umum adalah SQL(*Structured Query Language*) SQL ini adalah bahasa *nonprocedural* untuk mengakses data pada database relasional. SQL mengurangi pekerjaan kita dalam *database* karena apabila kita menggunakan bahasa pemograman yang umum, kita harus benar-benar memulainya segalanya dari awal.

2.5.1. Fiture-Fiture MySQL

Berikut ini akan dipaparkan karakteristik dari enggine perangkat lunak database MySQL.

Ditinjau dari internal dan probabilitasnya :

- Ditulis dalam bahasa C dan C++.
- Dapat bekerja dalam berbagai *platform*.
- Tersedia dalam versi *client server*

Ditinjau dari tipe kolom atau *field*nya :

- *Fixed-length* dan *variable-length record*
- Tipe field yang tersedia : *signed* atau *unsigned integer*, 1, 2, 3, 4, dan 8 bytes, *float, double, char, varchar, text, BLOB, date, time, datetime, timestamp, year, set, dan enum*

Ditinjau dari konektivitasnya :

- *Client* dapat berhubungan dengan *MySQL server* menggunakan soket TCP/IP, *unix* atau *Named Pipes*(NT)

2.5.2 Bagian- Bagian Dari MySQL

SQL terbagi menjadi tiga bagian yaitu :

1) *Data Definition Language*(DDL)
   Terdiri atas perintah-perintah SQL untuk membuat suatu obyek seperti tabel, *index, view*, dan lain- lain dalam suatu *database*.
2) *Data Manipulation Language* (DML)
   Kumpulan perintah- perintah SQL untuk menentukan, memanggil dan memanipulasi isi dari suatu tabel.
3) *Data Control Language* (DCL)
   Terdiri dari perintah untuk menentukan izin akses dari *user* dalam suatu *database*

2.6. Borland Delphi

Delphi adalah salah satu dari pemograman secara visual, bahasa yang digunakan lebih mengarah ke bahasa Pascal. Seperti bahasa lainnya, Delphi mengalami perkembangan yang sangat pesat. Delphi 7, versi terbaru yang dikeluarkan oleh Borland, memiliki support yang tinggi terhadap *database-database* yang sudah terkenal seperti MS Access, Paradox, Foxpro, Dbase, Oracle, SQL server, dan lain sebagainya, dan dilengkapi dengan obyek-obyek yang baru sehingga memudahkan dalam pembuatan program, baik program database maupun program lainnya.

Bahasa Pemograman Delphi berorientasi pada *objec- oriented* dan *event –driven* yang strukturnya relatif teratur dibandingkan bahasa lain. Sedangkan objek yang ditangani dapat berupa BLOB *(Binary Large Object*), OLE, *Dataset* dan *generik*. Semua interface basis datanya melalui BDE/IDAPI (*Broland Database Enggine*) dengan masing-masing basis datanya memiliki *driver* khusus.

Kesimpulan umum, delphi cocok digunakan untuk aplikasi besar baik dengan basis data maupun yang berorientasi *Visual-interface*. Hal ini disebabkan karena bahasa pemograman pada delphi tergolong baik dan konsisten.