

BAB II

DASAR TEORI

2.1 *Microcontroller*

Microcontroller adalah sebuah *system computer* yang seluruh atau sebagian besar elemennya dikemas dalam satu chip IC, sehingga sering disebut *single chip microcomputer*. *Microcontroller* merupakan *system computer* yang mempunyai satu atau beberapa tugas yang sangat spesifik. Elemen *microcontroller* tersebut diantaranya adalah:

1. Pemroses (*processor*) fungsinya mengeksekusi program, memuat (*load*) program dari suatu tempat. Pada PC, misalkan *web browser* di muat dari dalam *harddisk*.
2. Memori, sebagai tempat untuk menyimpan variabel
3. *Input* dan *output* sebagai *interface* untuk berkomunikasi dengan manusia.

Microcontroller telah banyak digunakan di industri, walaupun penggunaannya masih kurang dibandingkan dengan penggunaan *Programmable Logic Control* (PLC), tetapi *microcontroller* memiliki beberapa keuntungan dibandingkan dengan PLC. Ukuran *microcontroller* lebih kecil dibandingkan dengan suatu modul PLC sehingga peletakkannya dapat lebih *flexible*. *Microcontroller* telah banyak digunakan pada berbagai macam peralatan rumah tangga seperti mesin cuci. Sebagai pengendali sederhana, *microcontroller* telah banyak digunakan dalam dunia medik, pengaturan lalu-lintas, dan masih banyak lagi. Contoh alat ini diantaranya adalah computer yang digunakan pada mobil untuk mengatur kestabilan mesin, alat untuk pengatur lampu lalu lintas.

Secara teknis hanya ada 2 *microcontroller* yaitu RISC dan CISC, dan masing-masing mempunyai keturunan/keluarga sendiri-sendiri. RISC kependekan dari *Reduced Instruction Set Computer*: instruksi terbatas tapi memiliki fasilitas yang lebih banyak CISC kependekan dari *Complex Instruction Set Computer* instruksi bisa dikatakan lebih lengkap tapi dengan fasilitas secukupnya. Tentang jenisnya banyak sekali ada keluarga Motorola dengan seri 68xx, keluarga MCS51

yang diproduksi Atmel, Philip, Dallas, keluarga PIC dari Micro chip, Renesas, Zilog. Masing-masing keluarga juga masih terbagi lagi dalam beberapa tipe. Jadi sulit sekali untuk menghitung jumlah *microcontroller*.

Microcontroller, sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, hadir memenuhi kebutuhan pasar (*market need*) dan teknologi baru. Sebagai teknologi baru, yaitu teknologi semi konduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara masal (dalam jumlah banyak) membuat harganya menjadi lebih murah (dibandingkan mikroprosesor). Sebagai kebutuhan pasar, *microcontroller* hadir untuk memenuhi selera industry dan para konsumen akan kebutuhan dan keinginan alat-alat bantu bahkan mainan yang lebih baik dan canggih.

Tidak seperti sistem komputer, yang mampu menangani berbagai macam program aplikasi (misalnya pengolah kata, pengolah angka dan lain sebagainya), *microcontroller* hanya bisa digunakan untuk suatu aplikasi tertentu saja (hanya satu program saja yang bisa disimpan). Perbedaan lainnya terletak pada perbandingan RAM dan ROM. Pada sistem komputer perbandingan RAM dan ROM-nya besar, artinya program-program pengguna disimpan dalam ruang RAM yang relatif besar, sedangkan rutin-rutin antarmuka perangkat keras disimpan dalam ruang ROM yang kecil. Sedangkan pada *Microcontroller*, perbandingan ROM dan RAM-nya yang besar, artinya program control disimpan dalam ROM (bisa Masked ROM atau Flash PEROM) yang ukurannya relative lebih besar, sedangkan RAM digunakan sebagai tempat penyimpanan sementara, termasuk *register-register* yang digunakan pada *microcontroller* yang bersangkutan.

Adapun kelebihan dari mikrokontroller adalah sebagai berikut :

1. Penggerak pada mikrokontoler menggunakan bahasa pemograman assembly dengan berpatokan pada kaidah digital dasar sehingga pengoperasian sistem menjadi sangat mudah dikerjakan sesuai dengan logika sistem (bahasa assembly ini mudah dimengerti karena menggunakan bahasa assembly aplikasi dimana parameter *input* dan *output* langsung bisa diakses tanpa menggunakan banyak perintah).

Desain bahasa assembly ini tidak menggunakan begitu banyak syarat penulisan bahasa pemrograman seperti huruf besar dan huruf kecil untuk bahasa assembly tetap diwajibkan.

2. *Microcontroller* tersusun dalam satu *chip* dimana prosesor, memori, dan I/O terintegrasi menjadi satu kesatuan kontrol sistem sehingga *microcontroller* dapat dikatakan sebagai komputer mini yang dapat bekerja secara inovatif sesuai dengan kebutuhan sistem.
3. Sistem *running* bersifat berdiri sendiri tanpa tergantung dengan komputer sedangkan parameter komputer hanya digunakan untuk *download* perintah instruksi atau program. Langkah-langkah untuk *download* komputer dengan *microcontroller* sangat mudah digunakan karena tidak menggunakan banyak perintah.
4. Pada *microcontroller* tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem.

2.1.1 Keluarga *Microcontroller* AVR

Microcontroller dapat bekerja dengan kendali suatu program yang dibuat sendiri. Program tersebut akan berinteraksi dengan arsitektur perangkat keras yang sudah *fix* di dalam *chip*. *Microcontroller* keluarga AVR secara umum dapat dibagi menjadi 6 kelompok, yaitu :

1. Keluarga ATtiny yaitu biasanya bentuk dimensi kecil, ukuran memori kecil, jumlah pin masukan dan keluaran juga sedikit.
2. Keluarga AT90Sxx yaitu merupakan *microcontroller* yang pertama kali dibuat oleh Atmel Corp.
3. Keluarga ATmega yaitu merupakan pengembangan *microcontroller* AT90Sxx dengan fitur yang lebih banyak.
4. Keluarga AT86RFxx yaitu merupakan *microcontroller* berorientasi desain minimal.
5. Keluarga AT90USBxx yaitu merupakan *microcontroller* yang berorientasi pemrograman USB.

6. Keluarga AVR 32 bit contohnya AP7000, UC3Axxxx, UC3Bxxx, UC3Lxx, dsb yaitu merupakan *microcontroller* dengan register dan instruksi dengan panjang 32 bit.

Masing-masing keluarga *microcontroller* mempunyai kelebihan dan fungsi tersendiri. Secara historis, *microcontroller* seri AVR pertama kali diperkenalkan ke pasaran sekitar tahun 1997 oleh perusahaan Atmel, yaitu sebuah perusahaan yang sangat terkenal dengan produk *microcontroller* seri AT89S51/52-nya yang sampai sekarang masih banyak digunakan di lapangan. *Microcontroller* seri AT89S51/52 masih mempertahankan arsitektur dan set instruksi dasar *microcontroller* 8031 dari perusahaan INTEL. *Microcontroller* AVR diklaim memiliki arsitektur dan set instruksi yang benar-benar baru dan berbeda dengan arsitektur *microcontroller* sebelumnya yang diproduksi oleh perusahaan tersebut. Walaupun demikian, bagi para *programmer* yang sebelumnya telah terbiasa dengan *microcontroller* seri AT89S51/52, dan berencana untuk beralih ke *microcontroller* AVR, maka secara teknis tidak akan banyak kesulitan yang berarti karena selain konsep dan istilah-istilah dasarnya hampir sama, pemrograman *level* assemblernya pun relatif tidak jauh berbeda.

AVR merupakan *microcontroller* dengan arsitektur RISC (*Reduced Instruction Set Computer*) dengan lebar bus data 8 bit. Frekuensi kerja *microcontroller* AVR ini pada dasarnya sama dengan frekuensi osilator sehingga hal tersebut menyebabkan kecepatan kerja AVR untuk frekuensi kerja osilator yang sama akan dua belas kali lebih cepat dibandingkan dengan *microcontroller* keluarga AT89S51/52.

Dengan instruksi yang sangat variatif (mirip dengan sistem *Complex Instruction Set Computer* (CISC) serta jumlah register serba guna (*General Purpose Register*) sebanyak 32 buah yang semuanya terhubung secara langsung ke *Arithmetic Logic Unit* (ALU), kecepatan operasi *microcontroller* AVR ini dapat mencapai 16 MIPS (enam belas juta instruksi per detik), sebuah kecepatan yang sangat tinggi untuk ukuran *microcontroller* 8 bit yang ada di pasaran sampai saat ini.

Untuk memenuhi kebutuhan dan aplikasi industri yang sangat beragam, *microcontroller* keluarga AVR ini muncul di pasaran dengan tipe seri utama, yaitu tinyAVR, ClassicAVR (AVR), dan megaAVR.

Keseluruhan seri AVR ini pada dasarnya memiliki organisasi memori dan set instruksi yang sama (dengan demikian jika telah mahir menggunakan salah satu seri AVR, untuk beralih ke seri yang lain akan relatif mudah). Perbedaan antara tinyAVR, AVR dan megaAVR pada kenyataannya hanya merefleksikan tambahan-tambahan fitur yang ditawarkannya saja (misal adanya tambahan ADC internal pada seri AVR tertentu, jumlah Port I/O serta memori yang berbeda, dan sebagainya). Di antara ketiganya, megaAVR umumnya memiliki fitur yang paling lengkap, disusul oleh AVR, dan terakhir tinyAVR.^[1]

Tabel 2.1 Perbedaan seri AVR berdasarkan jumlah memori

Microcontroller AVR		Memori (byte)		
Jenis	Paket	Flash	EEPROM	SRAM
TinyAVR	8-32 pin	1-2K	64-128	0-128
AVR (ClasiccAVR)	20-44 pin	1-8K	128-512	0-1K
MegaAVR	36-64 pin	8-128K	512-4K	512-4K

Seperti terlihat pada tabel 2.1, semua jenis AVR ini telah dilengkapi dengan memori *flash* sebagai memori program. Tergantung serinya, kapasitas memori *flash* yang dimiliki bervariasi dari 1KB sampai 128KB. Secara teknis, memori jenis ini dapat diprogram melalui saluran antarmuka yang dikenal dengan nama *Serial Peripheral Interface* (SPI) yang terdapat pada setiap seri AVR tersebut. Dengan menggunakan perangkat lunak *programmer (downloader)* yang tepat, pengisian memori *flash* dengan menggunakan SPI ini dapat dilakukan, bahkan ketika chip AVR telah terpasang pada sistem akhir (*end system*), sehingga pemrogramannya sangat fleksibel dan tidak merepotkan pengguna. Secara praktis metode ini dikenal dengan istilah *In System Programming* (ISP), sedangkan perangkat lunaknya dinamakan *In System Programmer*.

Selain seri-seri di atas yang sifatnya lebih umum, perusahaan Atmel juga memproduksi beberapa jenis *microcontroller* AVR untuk tujuan yang lebih khusus

dan terbatas, seperti seri AT86RF401 yang khusus digunakan untuk aplikasi *wireless remote control* dengan menggunakan gelombang radio (RF); seri AT90SC yang khusus digunakan untuk peralatan sistem-sistem keamanan kartu SIM GSM, pembayaran via internet, dan lain sebagainya.

Microcontroller AVR memiliki beberapa fitur. Fitur-fitur tersebut antara lain adalah sebagai berikut:

- ❖ *Flash* mempunyai kegunaan menyimpan kode-kode instruksi sehingga merupakan memori dengan kapasitas terbesar diantara memori yang ada di dalam sebuah chip *microcontroller* memori yang ada di dalam sebuah chip *microcontroller*.
- ❖ *Max. Operating Frequency* adalah frekuensi kerja maksimal dari suatu *microcontroller*.
- ❖ *Central Processing Unit (CPU)* atau unit pengolah pusat keseluruhan sistem, yaitu menjalankan perangkat lunak yang disimpan pada memori program, mengatur jalur pengiriman data dari atau ke piranti-piranti (memori, I/O, dan sebagainya), mengolah data-data yang ada pada perangkat lunak, dan sebagainya.
- ❖ *# of Touch Channels* adalah kanal atau saluran sentuh *microcontroller*.
- ❖ *Hardware QTouch Acquisition* suatu sistem sentuh yang berfungsi untuk mengambil, mengumpulkan dan menyiapkan data, hingga memprosesnya untuk menghasilkan data yang dikehendaki.
- ❖ *Max I/O Pins* adalah suatu jumlah maksimal perangkat yang menghubungkan proses di dalam *microcontroller* dengan dunia luar.
- ❖ *External Interrupts* adalah fitur yang sangat penting bagi *microcontroller* untuk memenuhi kebutuhan sistem yang *real time*. Interupsi adalah kondisi yang membuat CPU berhenti dari rutinitas yang sedang dikerjakan (rutin utama) untuk mengerjakan rutin lain (rutin interupsi). Pada saat terjadi interupsi, *microcontroller* menunda operasi yang sedang dilakukan dengan mencatat alamat kode instruksi kemudian menyimpannya, lalu mengidentifikasi interupsi dan menjalankan rutin pelayanan interupsi (*interrupts service routine-ISR*). Rata-rata *microcontroller* memiliki interupsi eksternal. Interupsi yang dimiliki bisa dipacu oleh “*edge*” atau

“level”. *Edge triggered interrupt* bekerja tidak tergantung pada waktu terjadinya interupsi, tetapi interupsi terjadi karena *glitch* (adalah lonjakan tegangan sangat singkat (sehingga berbentuk seperti paku) yang terjadi akibat ketakserempakan pensaklaran tiap-tiap bit). Sementara itu, *Level triggered interrupt* harus tetap pada logika *high* atau *low* sepanjang waktu tertentu agar dapat terjadi interupsi, interupsi ini tahan terhadap *glitch*.

- ❖ *USB Converter. Microcontroller* tertentu memiliki fitur ini sehingga memungkinkan komunikasi serial dengan karakteristik USB. Fitur dari USB ini ada dua yaitu :
 - 1) *USB Speed* seberapa sering komunikasi serial mengkonversi dengan karakteristik USB pada selang waktu tertentu.
 - 2) *USB Interface* adalah komunikasi serial dengan karakter USB dengan *user interface*.
- ❖ *Serial Peripheral Interface (SPI)* merupakan port komunikasi serial atau bisa disebut juga salah satu mode komunikasi serial *synchronous* (suatu kejadian yang terjadi pada waktu bersamaan dengan rate yang sama, dan kejadian ini terjadi berkelanjutan dan dapat diprediksi) kecepatan tinggi antara prosesor dan peripheral.
- ❖ *TWI (I2C)* adalah *I2C* singkatan dari *Inter Integrated Circuit*, sangat ideal untuk diterapkan pada aplikasi menggunakan *microcontroller*. Protokol ini mengizinkan desain sistem untuk saling berkoneksi sampai 128 piranti yang berlainan menggunakan hanya 2 jalur dua arah, satu untuk clock (SCL) dan satunya untuk jalur data (SDA).
- ❖ *Universal Asynchronous Receiver Transmitter (UART)* adalah protokol komunikasi yang umum digunakan dalam pengiriman data serial antara *device* satu dengan yang lainnya. Sebagai contoh komunikasi antara sesama mikrokontroler atau mikrokontroler ke PC.
- ❖ *LIN (Local Interconnecting Network)* digunakan dalam jaringan *Electronic Control Unit* berkecepatan rendah sebagai pengendali *sensor* dan *actuator* yang sederhana.

- ❖ *Graphics LCD* berfungsi untuk menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi *microcontroller*.
- ❖ *Video Decoder* digunakan untuk menampilkan video.
- ❖ *Camera Interface* digunakan untuk mengolah data visual.
- ❖ *Analog to Digital Converter* (ADC) adalah salah satu fasilitas *microcontroller* yang berfungsi untuk mengubah data *analog* menjadi data *digital*. Ada 3 macam ADC yaitu :
 - 1) *ADC Channels* adalah kanal atau saluran ADC.
 - 2) *ADC Resolution* menentukan ketelitian nilai hasil konversi ADC.
 - 3) *ADC Speed* suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu.
- ❖ *Resistive Touch Screen* device dengan *resistive screen* dioperasikan dengan ditekan, karena *resistive screen* ini terdiri dari beberapa layer. Pada bagian ini berfungsi untuk mendeteksi sentuhan sinyal-sinyal listrik.
- ❖ *Digital to Analog Converter* (DAC) adalah salah satu fasilitas mikrokontroller yang berfungsi untuk mengubah data *digital* menjadi data *analog*. Ada 2 macam DAC yaitu :
 - 1) *DAC Channels* kanal atau saluran DAC.
 - 2) *DAC Resolution* ketelitian nilai hasil konversi DAC.
- ❖ *Temperature Sensor* adalah sensor suhu yang berbentuk rangkaian terintegrasi dan mempunyai output berupa tegangan yang berubah secara linier dan proporsional terhadap suhu (skala celcius).
- ❖ *Crypto Engine* adalah keamanan mesin dalam melakukan pengiriman data.
- ❖ *Static Random Access Memory* (SRAM) adalah jenis memory yang tidak perlu penyegaran oleh CPU agar data yang terdapat didalamnya tetap tersimpan dengan baik.
- ❖ EEPROM merupakan kependekan dari *Electrically Erasable Programmable Read-Only Memory*. EEPROM adalah memori yang dapat mempertahankan data yang disimpan, meskipun catu daya dihilangkan serta cara penulisan dan penghapusannya dengan menggunakan elektrik.

- ❖ *Self Program Memory* fitur yang dapat mengisi dirinya sendiri fitur inilah yang dimanfaatkan untuk mengubah isi *flash* melalui *bootloader*. *Bootloader* sendiri bisa menerima data yang akan di-*flash* melalui jalur *serial* UART yang dikirim dari komputer.
- ❖ *External Bus Interface* adalah bus computer untuk interfacing perangkat peripheral kecil seperti flash memori dengan prosesor. Hal ini digunakan untuk memper luas bus internal prosesor untuk memungkinkan koneksi dengan memori-memori eksternal atau peripheral lainnya.
- ❖ *Dinamic Random Access Memory* (DRAM) adalah jenis memory yang bersifat volatile yang menyimpan setiap bit rate dalam kapasitor yang terpisah dalam sebuah sirkuit terpadu.
- ❖ *NAND Interface* yaitu fungsi memory non volatile yang disusun dalam gerbang NAND dan untuk operasi *input/output* dilakukan secara *interface*.
- ❖ *picoPower* adalah pemanfaatan listrik dengan daya yang sangat rendah.
- ❖ *Temp. Range* (deg C) digunakan untuk mengetahui besarnya suhu. IC ini akan mengubah nilai suhu menjadi besaran tegangan dengan jangkauan (range) suhu yang mampu dirasakan.
- ❖ *I/O Supply Class* untuk *supply input/output* kumpulan data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu.
- ❖ *Operating Voltage* (Vcc) merupakan masukan sumber tegangan positif catu daya pada *microcontroller*.
- ❖ *Floating Point Unit* (FPU) adalah bagian dari sistem komputer yang dirancang khusus untuk melakukan operasi pada angka *floating point*.
- ❖ MPU/MMU pengertiannya adalah sebagai berikut :
 - 1) *Microprocessor Unit* (MPU) adalah sebuah IC (*integrated circuit*) yang digunakan sebagai pengendali utama pada system berbasis *microcontroller*.
 - 2) *Manajemen Memori Unit* (MMU) merupakan pusat kegiatan pada komputer, sebab setiap proses yang akan dijalankan, harus melalui memori terlebih dahulu.
- ❖ *Timer /counter* adalah seperangkat pencacah biner biner (*binary counter*) yang terhubung langsung ke saluran-data *microcontroller* sehingga

microcontroller bisa membaca kedudukan pencacah tersebut. Seperti layaknya pencacah biner, bilamana sinyal denyut (*clock*) yang diumpangkan sudah melebihi kapasitas pencacah maka pada bagian akhir untaian pencacah akan timbul sinyal limpahan.

- ❖ *Output Compare channel* skanal atau saluran yang digunakan untuk menghasilkan frekuensi atau detak dari timer/counter ke sebuah pin *output* di *microcontroller*.
- ❖ *Input Capture Channels* kanal atau saluran yang digunakan untuk mengukur ketebalan detak (*pulse width*) atau waktu penangkapan (*capturing time*).
- ❖ *Pulse Width Modulation (PWM) Channels* adalah teknik mendapatkan efek sinyal analog dari sebuah sinyal digital yang terputus-putus. PWM dapat dibangkitkan hanya dengan menggunakan digital i/o yang difungsikan sebagai output.
- ❖ 32kHz *Real Time Clock (RTC)* adalah jenis pewaktu yang bekerja berdasarkan waktu yang sebenarnya atau dengan kata lain berdasarkan waktu yang ada pada jam.
- ❖ *Calibrated RC Oscillator* adalah berupa jaringan resistor dan kapasitor yang terkalibrasi.
- ❖ *Watchdog* merupakan timer yang terpisah dengan sistem CPU *microcontroller* sehingga tidak akan terpengaruh bagaimanapun keadaan CPU bekerja.
- ❖ *Analog Comparators* digunakan untuk membandingkan nilai masukan pada pin tak membalik (*non-inverting*) AIN0 dan pin membalik (*inverting*) AIN1 di mana kedua nilai tegangan yang dibandingkan adalah tegangan analog.^[2]

2.1.2 Database

Database adalah sekumpulan *file* data yang saling berhubungan dan diorganisasi sedemikian rupa sehingga data-data tersebut dapat diakses dengan mudah dan cepat dan diproses menjadi sebuah informasi yang lebih bermanfaat.

Dalam *database*, data yang ada tidak hanya disimpan begitu saja dalam sebuah media penyimpanan, tetapi dikelola dan diolah oleh sebuah sistem *database*

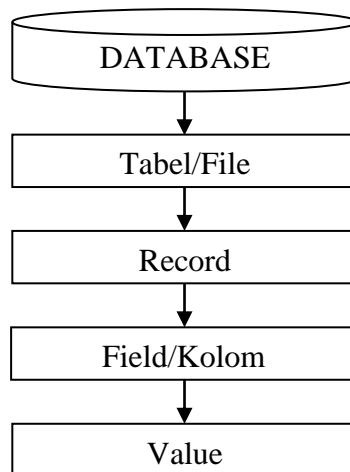
yang disebut *Database Management System* (DBMS). Dengan demikian, data yang begitu besar dan kompleks dapat tersusun dengan rapi dan terstruktur sehingga meningkatkan kecepatan akses datanya (membuat aplikasi database dengan java dan MySQL).

Manfaat *database* banyak dijumpai saat ini. ATM (Anjungan Tunai Mandiri) merupakan sebuah contoh teknologi informasi yang pada dasarnya memanfaatkan *database*, yang memungkinkan seseorang bisa mengambil uang di mana saja dan kapan saja. Di dalam *database* tersimpan data yang menyangkut rekening nasabah, *password* yang *valid* untuk nasabah, dan juga saldo tabungan nasabah. Aplikasi *database* yang lain dapat dijumpai pada toko-toko swalayan, perpustakaan, dan bahkan pada *Web*, seseorang bisa melihat buku-buku pada situs-situs toko buku *online* dengan hanya memasukkan kata kunci tertentu seperti *database* atau *Access*. Aplikasi tersebut akan mencocokkan dengan *database* yang tersedia dan kemudian menampilkan judul-judul buku beserta atribut lainnya (nama pengarang, ISBN, dan sebagainya) ke layar komputer pemakai.

Untuk mengelola database diperlukan suatu perangkat lunak yang disebut *Database Management System* (DBMS). DBMS merupakan suatu *system* perangkat lunak yang memungkinkan *user* (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses *database* secara praktis dan efisien. Dengan DBMS, *user* akan lebih mudah mengontrol dan memanipulasi data yang ada.

2.2.1 Komponen *Database*

Sebelum merancang *database*, sebaiknya mengenal terlebih dahulu istilah-istilah yang biasa digunakan dalam *database*. Istilah-istilah tersebut memiliki urutan bertingkat dalam struktur sebuah *database*, yaitu dari komponen yang kecil hingga yang terbesar. Tingkatan data dapat dilihat pada Gambar 2.1 berikut.



Gambar 2.1 Tingkatan data

Adapun penjelasan dari tingkatan data tersebut adalah sebagai berikut :

1. *Database* merupakan kumpulan dari *field*/tabel yang saling berhubungan. *Database* menduduki urutan tertinggi karena di dalamnya semua data disimpan dan dikelola.

Gambar 2.2 Simbol *database*

2. Tabel sering disebut entitas atau *entity*. Tabel atas *record-record* yang menggambarkan kesatuan data-data yang sejenis.
3. *Record* merupakan kumpulan *field* yang membentuk suatu *record*. Suatu *record* menggambarkan informasi tentang individu tertentu.
4. *Field*/Kolom merupakan atribut dari *record* yang menunjukkan suatu *value*/item data. Kumpulan *field* yang membentuk suatu *record* harus diberi nama untuk membedakan antara *field* satu dengan yang lain. Pada *field* ini harus didefinisikan tipe data dan panjang maksimal data yang akan disimpan.
5. *Value* adalah jenjang terkecil yang merupakan isi dari *field* yang dapat berupa karakter, huruf, dan angka. *Value* dapat juga disebut data yang tersimpan dalam setiap *field*/kolom.

2.2.2 Database Management System

Database Management System atau yang biasa disingkat DBMS. DBMS merupakan perangkat lunak atau program komputer yang dirancang secara khusus untuk memudahkan pengelolaan *database*. Salah satu macam DBMS yang populer dewasa ini berupa *Relational Database Management System* (RDBMS), yang menggunakan model basis data relasional atau dalam bentuk tabel-tabel yang saling terhubung.

MySQL merupakan salah satu contoh produk RDBMS yang sangat populer di lingkungan Linux, tetapi juga tersedia pada Windows. Banyak situs *Web* yang menggunakan MySQL sebagai *database server* (*server* yang melayani permintaan akses terhadap *database*).

Syntax dasar dalam DBMS adalah menghapus suatu *record* atau data dari tabel, mengubah struktur tabel, menambah *record* atau data ke dalam suatu tabel, dan mengedit suatu *record* atau data dari suatu tabel berikut adalah beberapa format penggunaan *syntax* dari DBMS :

1. Proses *delete* atau hapus dilakukan jika terdapat data atau *record* dalam suatu tabel yang perlu dihapus atau dihilangkan. Perubahan yang terjadi dalam proses *delete* bersifat permanen, artinya setelah perintah dijalankan tidak dapat dibatalkan (*undo*). *Syntax* untuk menghapus suatu *record* atau data dari tabel adalah sebagai berikut :

DELETE FROM nama_tabel [**WHERE** kondisi];

Keterangan pada perintah untuk *delete* adalah :

- a) DELETE FROM merupakan perintah dasar untuk menghapus suatu *record* dari tabel.
- b) nama_tabel merupakan nama tabel yang *record* atau datanya akan dihapus.
- c) Perintah WHERE diikuti oleh kondisi tertentu yang menentukan *record* mana yang akan dihapus (*delete*). Perintah WHERE ini boleh ada boleh juga tidak. Namun demikian, jika WHERE tidak ditambahkan pada perintah *delete* maka semua *record* dalam tabel bersangkutan akan terhapus.

2. *Syntax* untuk mengubah struktur tabel adalah sebagai berikut :

ALTER TABLE nama_tabel alter_options;

Dimana :

- a) Alter Table merupakan perintah dasar untuk mengubah tabel.
- b) nama_tabel merupakan nama tabel yang akan diubah strukturnya.
- c) alter_option merupakan pilihan perubahan tabel. *Option* yang bisa digunakan, beberapa di antaranya sebagai berikut :

- 1) **ADD** definisi_field_baru

Option ini digunakan untuk menambahkan *field* baru dengan “definisi_field_baru” (nama *field*, tipe dan *option* lain).

- 2) **ADD INDEX** nama_index

Option ini digunakan untuk menambahkan *index* dengan nama “nama_index” pada tabel.

- 3) **ADD PRIMARY KEY** (*field_kunci*)

Option untuk menambahkan primary key pada tabel.

3. *Syntax* untuk menambah *record* atau data ke dalam suatu tabel adalah sebagai berikut :

INSERT INTO nama_tabel **VALUES** ('nilai1', 'nilai2', ...);

atau dapat dengan bentuk sebagai berikut :

INSERT INTO nama_tabel (*field1*, *field2*, ...) **VALUES** ('nilai1', 'nilai2', ...);

atau dapat juga dengan bentuk sebagai berikut :

INSERT INTO nama_tabel **SET** *field1*='nilai1', *field2*='nilai2', ...;

4. Proses *update* bisa sewaktu-waktu dilakukan jika terdapat data atau *record* dalam suatu tabel yang perlu diperbaiki. Proses update ini tidak menambahkan data (*record*) baru, tetapi memperbaiki data yang lama. Perubahan yang terjadi dalam proses update bersifat permanen, artinya setelah perintah dijalankan tidak dapat dibatalkan (*undo*). *Syntax* untuk mengedit suatu *record* atau data dari suatu tabel adalah sebagai berikut :

UPDATE nama_tabel **SET** *field1*='nilaibaru' [**WHERE** kondisi];

Dimana :

- a) UPDATE merupakan perintah dasar untuk mengubah *record* tabel.
- b) nama_tabel merupakan nama tabel yang akan diubah *recordnya*.
- c) Perintah SET diikuti dengan *field-field* yang akan diubah yang mana diikuti juga dengan perubahan isi dari masing-masing *field*. Untuk mengubah nilai dari beberapa *field* sekaligus, gunakan koma (,) untuk memisahkan masing-masing *field*.
- d) Perintah WHERE diikuti oleh kondisi tertentu yang menentukan record mana yang akan diedit (diubah). Perintah WHERE ini boleh ada boleh juga tidak. Jika WHERE tidak ditambahkan pada perintah update maka semua *record* dalam tabel bersangkutan akan berubah.^[3]

2.2.3 SQL dan MySQL

Structured Query Language (SQL) adalah salah satu dari sekian banyak bahasa pemrograman *database* yang paling *popular*. SQL merupakan bahasa pemrograman yang gratis dan mudah digunakan. SQL dibangun di laboratorium IBM San Jose California pada tahun 1970-an. SQL pertama kali dikembangkan sebagai bahasa pada *database* DB2 hingga saat ini masih merupakan produk andalan IBM. SQL memungkinkan untuk membuat *database* sekaligus mengelolanya, yaitu menambah, menghapus, mengubah, mencari data, dan lain sebagainya (membuat aplikasi database dengan java dan MySQL).

Dalam bahasa SQL pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi terdiri dari baris (*row* atau *record*) dan kolom (*column* atau *field*). Sedangkan dalam sebuah *database* dapat terdiri dari beberapa *table*.

MySQL adalah sebuah program *Database server* yang mampu menerima dan mengirimkan data dengan cepat dengan menggunakan perintah-perintah SQL. MySQL memiliki dua bentuk lisensi, yaitu lisensi GNU/GPL (*General Public License*).

Di dalam MySQL, terdapat 3 sub bahasa, yaitu *Data Definition Language* (DDL), *Data Manipulation Language* (DML), dan *Data Control Language* (DCL). DDL digunakan untuk membangun objek-objek dalam *database*, seperti tabel dan

index. DML digunakan untuk menambah, mencari, mengubah, dan menghapus baris dalam tabel. Sedangkan DCL digunakan untuk menangani masalah *security*/keamanan dalam *database* (membuat aplikasi database dengan java dan MySQL). Perintah-perintah MySQL antara lain :

Tabel 2.1 Perintah-perintah MySQL^[4]

Perintah	Perintah Singkat	Kegunaan
<i>Help</i>	\h	Menampilkan daftar perintah
<i>Clear</i>	\c	Menghapus (<i>clear</i>)
<i>Connect</i>	\r	Menghubungkan kembali <i>database</i> MySQL
<i>Exit</i>	\q	Keluar dari MySQL
<i>Go</i>	\g	Mengirimkan perintah kepada MySQL
Perintah	Perintah Singkat	Kegunaan
<i>Ego</i>	\G	Mengirimkan perintah kepada MySQL dan menampilkan hasilnya secara <i>vertical</i>
<i>Print</i>	\p	Mencetak perintah saat ini
<i>Use</i>	\u	Membuat/mengganti koneksi kepada <i>database</i>

Ketentuan Memberikan Perintah

1. Perintah dalam MySQL mengenal *casein sensitive*, perintah dapat ditulis huruf besar (*uppercase*), ataupun dengan huruf kecil (*lowercase*).
2. Setiap perintah diakhiri dengan ; (tanda titik koma) atau dengan memberikan perintah \g diakhir perintah.
3. Perintah dapat berupa perintah SQL atau perintah khusus MySQL.
4. Jika Prompt mysql> berganti dengan -> berarti prompt tersebut menunggu kelengkapan perintah dari basis sebelumnya atau menunggu diberikan tanda ; atau \g.

Contoh : Misalkan perintah dibawah ini ditulis tanpa tanda titik-koma “ ; ”.

```
mysql> create database latihan1
```

```
->
```


Sistem MySQL akan menampilkan tanda panah ‘->’ yang menyatakan bahwa perintah MySQL tersebut dianggap belum selesai (karena belum diakhiri dengan tanda titik-koma ‘;’).

Sekarang jika perintah sebelumnya dilengkapi dengan tanda titik-koma ‘;’

```
mysql> create database latihan1
```

```
-> ;
```

```
Query OK, 1 row affected (0.02 sec). [5]
```

2.2 Bahasa Pemrograman

Bahasa pemrograman adalah teknik komunikasi standar untuk mengekspresikan instruksi kepada komputer. Seperti bahasa manusia, setiap bahasa memiliki tata tulis dan aturan tertentu.

Bahasa pemrograman memfasilitasi seorang programmer secara tepat menetapkan data apa yang sedang dilakukan oleh komputer selanjutnya, bagaimana data tersebut disimpan dan dikirim, dan apa yang akan dilakukan apabila terjadi kondisi yang variatif.

Bahasa pemrograman dapat diklasifikasikan menjadi tingkat rendah, menengah, dan tingkat tinggi. Pergeseran tingkat dari rendah menuju tinggi menunjukkan kedekatan terhadap ”bahasa manusia”.

2.3.1 Java

Java adalah bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rakan-rekannya seperti Patrick Naughton, Chris Warth, Ed Frank, dan Mike Sheridan di suatu perusahaan perangkat lunak yang bernama Sun Microsystems, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “Oak”, namun pada tahun 1995 diganti namanya menjadi “Java”. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, seperti *microwave oven* dan *remote control*, sehingga Java harus bersifat *portable* atau yang sering disebut dengan *platform-independent* (tidak tergantung pada *platform*). Itulah yang menyebabkan dalam dunia pemrograman Java, dikenal dengan istilah ‘*write once, run everywhere*’, yang berarti kode program hanya dapat ditulis sekali, namun

dapat dijalankan di bawah *platform* manapun, tanpa harus melakukan kode program.

Java adalah sebuah bahasa dan teknologi pemrograman yang dapat berjalan pada multiplatforms. Secara arsitektur, tidak ada perubahan sedikitpun pada Java semenjak awal mula bahasa pemrograman tersebut mulai dirilis. Kompiler Java (disebut dengan JavaC atau *Java Compiler*) akan mengubah kode-kode dalam bahasa Java ke dalam suatu *bytecode*. *Bytecode* adalah beberapa perintah hasil kompilasi dari bahasa Java yang kemudian dapat dieksekusi dengan mesin komputer abstrak, yang disebut dengan JVM (*Java Virtual Machine*). JVM disebut juga sebagai *interpreter*, karena sifat *interpreter* yang selalu menerjemahkan kode-kode yang tersimpan dalam *bytecode* secara baris demi baris.^[6]

a) *Java Development Kit (JDK)*

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java menjadi *bytecode* yang dapat dimengerti dan dapat dijalankan oleh *Java Runtime Environment (JRE)*.

Java Development Kit harus terpasang pada komputer yang akan melakukan proses pembuatan aplikasi berbasis java. Namun *Java development Kit* tidak harus terpasang di komputer yang akan menjalankan aplikasi yang dibangun menggunakan java.

b) *Java Runtime Environment (JRE)*

Java Runtime Environment (JRE) merupakan perangkat lunak yang digunakan untuk menjalankan aplikasi yang dibangun menggunakan java. Versi JRE harus sama atau lebih tinggi dari JDK yang digunakan untuk membangun aplikasi agar aplikasi dapat berjalan sesuai dengan yang diharapkan.

c) NetBeans IDE

Netbeans IDE merupakan perangkat lunak yang digunakan untuk membangun perangkat lunak yang lain. Netbeans IDE dapat digunakan untuk membangun perangkat lunak berbasis *Java Standard Edition*, *Java Enterprise Edition*, *Java Micro Edition*, JavaFX, PHP, C/C++, Groovy dan Python.

Netbeans memiliki *Integrated Development Environment*, atau *Integrated Design Environment* dan *Intergrated Debugging Envieronmet* yang biasa disingkat IDE, yakni sebuah program atau alat bantu yang terdiri atas Editor, Compiler yang terintegrasi dalam satu aplikasi.^[5]

NetBeans mempunyai kelebihan diantaranya :

- 1) Dapat berjalan pada *multiplat form* sistem operasi termasuk *Windows, Linux, Mac OS, Solaris*.
- 2) Mendukung permodelan perangkat lunak dengan UML (*Unified Modeling Language*).
- 3) Terdapat banyak modul untuk mengembangkan lebih lanjut.
- 4) Merupakan produk *free* (bebas) tanpa ada batasan penggunaannya.
- 5) Merupakan produk *open source* (kode sumber terbuka).

d) Keunggulan Java

Berdasarkan white paper resmi dari SUN, Java memiliki karakteristik berikut :

1) Sederhana

Bahasa pemrograman java menggunakan sintaks mirip dengan C++ namun sintaks pada java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance*. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection*.^[6]

2) Berorientasi objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali.

Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

3) Dapat didistribusi dengan mudah

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries networking* yang terintegrasi pada java.

4) *Interpreter*

Program Java dijalankan menggunakan *interpreter* yaitu *Java Virtual Machine (JVM)*. Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi *Java byte codes* dapat dijalankan pada *platform* yang berbeda-beda.

5) *Robust*

Java mempunyai reliabilitas yang tinggi. *Compiler* pada java mempunyai kemampuan mendeteksi *error* secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling* untuk membantu mengatasi *error* pada pemrograman.

6) Aman

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

7) *Architecture Neutral*

Program java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada *platform* yang berbeda dengan *Java Virtual Machine*.

8) *Portabel*

Source code maupun program java dapat dengan mudah dibawa ke *platform* yang berbeda-beda tanpa harus dikompilasi ulang.

9) *Performance*

Performance pada java sering dikatakan kurang tinggi. Namun performance java dapat ditingkatkan menggunakan kompilasi java lain

seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan *Just In Time Compilers* (JIT).

10) *Multithreaded*

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

11) Dinamis

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan *properties* ataupun *method* dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.^[7]

e) **Fitur dari Java**

1) *Java Virtual Machine* (JVM)

JVM adalah sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi *hardware* dan *platform* dimana kompilasi kode java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis java menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh JVM.

Aplikasi program Java diciptakan dengan file teks berekstensi *.java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada system komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *javainterpreter* yang tersedia disemua *platform* sistem komputer dan sistem operasi.

2) *Garbage Collection*

Banyak bahasa pemrograman lain yang memungkinkan seorang *programmer* mengalokasikan memori pada saat dijalankan. Namun, setelah menggunakan alokasi memori tersebut, harus terdapat cara untuk menempatkan kembali blok memori tersebut supaya program lain dapat menggunakannya. Dalam C, C++ dan bahasa lainnya, adalah

programmer yang mutlak bertanggung jawab akan hal ini. Hal ini dapat menyulitkan bila mana *programmer* tersebut alpa untuk mengembalikan blok memori sehingga menyebabkan situasi yang dikenal dengan nama *memory leaks*.^[9]

Program Java melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek–objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh *programmer* dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat pada bahasa yang memungkinkan alokasi dinamis.

3) *Code Security*

Code Security terimplementasi pada java melalui penggunaan *Java Runtime Environment* (JRE). Java menggunakan model pengamanan 3 lapis untuk melindungi sistem dari *untrusted Java Code*.

- Pertama, *class-loader* menangani pemuatan kelas java ke *runtime interpreter*. Proses ini menyediakan pengamanan dengan memisahkan kelas–kelas yang berasal dari *local disk* dengan kelas–kelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas–kelas yang berasal dari *local disk* yang dimuat terlebih dahulu.
- Kedua, *bytecodeverifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan–aturan dasar bahasa java.
- Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem *file*, *port* jaringan, proses eksternal dan sistem windowing.

Setelah seluruh proses tersebut selesai dijalankan, barulah kode program dieksekusi.^[10]