

DISERTAI
CD
GRATIS

Sebagai Penunjang
dalam Mempelajari
Buku ini

■ Rahmadya Trias Handayanto
■ Herlawati

PEMROGRAMAN BASIS DATA DI **MATLAB**



DENGAN
MYSQL
DAN
**MICROSOFT
ACCESS**

Bahasan Meliputi:

- Teori Dasar Sistem Basis Data
- Koneksi Matlab Ke Basis Data
- Akses Ke Basis Data Dengan Graphic User Interface (GUI)
- Contoh Penerapan Aplikasi Matlab Dengan Basis Data
- Dasar-Dasar Struktur Data di Matlab
- Membuat Menu di GUI Matlab
- Mengkompilasi Aplikasi Matlab



Penerbit **INFORMATIKA**

Pemrograman Basis Data di Matlab

dengan
MySQL dan Microsoft Access

Rahmadya Trias Handayanto
Herlawati



Penerbit **INFORMATIKA**

Pemrograman
Basis Data di Matlab
dengan
MySQL dan Microsoft Access

Rahmadya Trias Handayanto
Herlawati

***Pemrograman Basis Data di Matlab
dengan MySQL dan Microsoft Access***

Penyusun : Rahmadya Trias Handayanto
Herlawati

Penerbit : Informatika Bandung

Pemasaran : **BI-Obses**
Pasar Buku Palasari No. 82
Bandung 40264
Telp.(022)7317812
Fax. (022)7317896

Cetakan Pertama : Februari 2016

ISBN : 978-602-6232-06-9

Copyright © 2016 pada Penerbit INFORMATIKA Bandung

Persembahan

untuk yang tercinta:

Orangtua kami

Ladya Ramadhani — Putri Kami

Dimas Putra Rahmadya — Putra Kami

Panduan Penggunaan CD

Kode sumber yang tertulis di buku ini kami sertakan dalam bentuk CD. Dibutuhkan sistem operasi Windows, Microsoft Office dan aplikasi Matlab untuk menjalankannya. CD terdiri dari *folder-folder* sesuai dengan nama bab di dalam buku dan untuk menjalankannya diperlukan langkah-langkah sebagai berikut:

1. Pindahkan data CD ke komputer
2. Arahkan direktori kerja (*current directory*) ke folder bab yang sesuai
3. Buat koneksi ODBC terhadap dua database (Microsoft Access dan MySQL) yang ada di folder bernama *database* dengan mengikuti bab mengenai koneksi Matlab ke basis data.
4. Buka GUI dengan mengetik 'guide beasiswa' atau 'guide citra' pada command window untuk melihat jendela rancangan GUI.
5. Ketik 'edit beasiswa' atau 'edit citra' pada command window untuk melihat jendela m-file editor tempat kode sumber (*source code*) dibuat.
6. Diperlukan *software* Microsoft Access (biasanya sudah ada ketika instalasi Microsoft Office). Sementara untuk menghidupkan server apache dan *database* MySQL dapat Anda lakukan dengan menjalankan (klik ganda) file *xampp_start.exe* yang ada di *folder* *xamplite* (*xampp_stop.exe* untuk mematikan *server apache*). XAMPP lebih ringan dibanding dengan yang lain seperti Apache2triad karena tidak membebani kerja komputer (hanya aktif jika dijalankan). Tentu saja sangat diperbolehkan jika menggunakan *server* apache2triad, apalagi dengan komputer *server* sesungguhnya.

Prakata

Matlab yang merupakan bahasa komputasi teknis dan disukai oleh para peneliti sudah merambah ke berbagai disiplin ilmu, mulai dari ilmu-ilmu eksakta hingga ilmu-ilmu sosial. Terlebih lagi para mahasiswa maupun peneliti yang berkecimpung dengan data-data yang harus diolah. Di awal perkembangannya Matlab bekerja dengan data berupa matriks yang berasal dari Excel atau *file* berkecandian dat yang diketik dengan text editor (notepad, wordpad, dan sejenisnya). Namun dengan bertambahnya riset di bidang data mining yang melibatkan data berukuran besar serta dinamis, tentu saja program text editor seperti notepad atau *spreadsheet* seperti Microsoft Excel tidak sanggup menanganinya. Dibutuhkan *software Database Management System (DBMS)* seperti Oracle, MySQL, PostgreSQL, Microsoft Access, dan sebagainya. Sementara itu, Matlab membutuhkan sarana untuk terhubung ke *software DBMS* tersebut.

Sebenarnya buku ini muncul dari pertanyaan-pertanyaan pembaca buku Matlab sebelumnya tentang pengelolaan data yang berasal dari berbagai jenis *software database*. Berkembangnya *Graphic User Interface (GUI)* pada Matlab juga membuat pemakai membandingkan dengan bahasa pemrograman visual lainnya yang sudah terlebih dahulu mudah terhubung dan *user friendly* dengan aplikasi DBMS seperti Visual Basic, Java, C#, dan lain-lain. Walaupun Matlab memang tidak dibuat khusus untuk pengelolaan *database* tetapi alangkah lengkapnya jika bahasa pemrograman yang tangguh di dalam aplikasi komputasi juga tangguh dalam pengelolaan *database*.

Setelah pendahuluan yang berisi pengenalan sistem basis data dibahas, bab selanjutnya membahas teknik menghubungkan aplikasi Matlab dengan sistem basis data yang sering digunakan, yaitu Microsoft Access yang ber-*platform* Windows. *Open Database*

Connectivity (ODBC) yang tersedia di sistem operasi windows juga akan coba dimanfaatkan mengingat *driver* penghubung *software* database dengan aplikasi ini membuat pengguna lebih fleksibel ketika beralih dari satu *platform software* DBMS satu ke yang lainnya. Untuk pengguna *open source software* seperti MySQL dan PostgreSQL akan coba dibahas di bab berikutnya. Terakhir adalah contoh-contoh aplikasi Sistem Pendukung Keputusan (*Decision Support System*) berbasis *soft computing* dengan koneksi terhadap sistem basis data sederhana yang sering kali dijumpai oleh mahasiswa dan peneliti dibahas dan dapat dicoba untuk dipraktikkan oleh pembaca.

Berbagai pihak telah banyak membantu penulis baik langsung maupun tidak langsung sehingga tulisan ini selesai. Beberapa bagian dari tulisan ini berasal dari riset-riset yang didanai oleh Direktorat Pendidikan Tinggi (DIKTI) lewat hibah-hibahnya. Juga terhadap almarhum Dr. Prabowo Pudjo Widodo, MS yang merupakan pembimbing dan sudah beberapa kali bersama penulis menerbitkan *paper* ilmiah dan buku-buku. Kampus tempat penulis mengambil doktor juga banyak memberikan kontribusi berupa *course*, sarana laboratorium (lisensi *software* Matlab), dan pelatihan-pelatihan rutin yang diadakan tentang Matlab. Dorongan moril dari keluarga juga sangat membantu motivasi penulis merampungkan penulisan buku ini.

Bekasi, 2015

Penulis

Daftar Isi

PRAKATA.....	vii
DAFTAR ISI.....	ix
BAB 1. TEORI DASAR SISTEM BASIS DATA.....	1
1.1 Data dan Manajemen Basis Data.....	2
1.2 Siklus Hidup Basis Data (<i>Database Life Cycle</i>)	3
1.3 Pemodelan Data Konseptual (<i>Conceptual Data Modeling</i>)..	5
1.4 Pemodelan Data Fisik (<i>Phisycal Data Modeling</i>).....	6
1.5 <i>Structure Query Language</i> (SQL).....	7
1.5.1 <i>Data Definition Language</i> (DDL).....	10
1.5.2 <i>Data Manipulation Language</i> (MDL).....	13
1.5.3 <i>Data Control Language</i> (DCL)	15
1.5.4 Perintah SQL Lanjut.....	16
BAB 2. KONEKSI MATLAB KE BASIS DATA.....	21
2.1 <i>Open Database Connectivity</i> (ODBC).....	22
2.2 Koneksi Matlab dengan Microsoft Office Access.....	25
2.2.1 Mempersiapkan <i>Database</i> dengan Microsoft Office Access	26
2.2.2 Koneksi Microsoft Access dengan Matlab.....	31
2.2.3 Menghubungkan Microsoft Access dengan Fungsi <i>Database</i> pada Matlab	35
2.2.4 Manipulasi Data SQL di Microsoft Access (<i>Select, Insert,</i> <i>Update dan Delete</i>) dengan Matlab	38

2.3	Koneksi Matlab dengan MySQL	45
2.3.1	Mempersiapkan <i>Database</i> dengan MySQL	46
2.3.2	Koneksi MySQL dengan Matlab.....	52
2.3.3	Menghubungkan MySQL dengan Matlab	54
2.3.4	Manipulasi Data SQL di MySQL (<i>Select, Insert, Update</i> dan <i>Delete</i>) dengan Matlab	56

BAB 3. AKSES KE BASIS DATA DENGAN *GRAPHIC USER*

***INTERFACE (GUI)*..... 63**

3.1	Dasar-dasar Pemrograman Visual (GUI) Matlab.....	64
3.1.1	Membuat Antarmuka (<i>Interface</i>) dengan GUI.....	65
3.1.2	Menghubungkan GUI dengan Basis Data.....	71
3.1.3	Memasukan Data ke Basis Data dengan GUI	76
3.1.4	Mengedit Data dengan GUI.....	80
3.1.5	Menghapus Data dengan GUI.....	89

BAB 4. CONTOH PENERAPAN APLIKASI MATLAB

DENGAN BASIS DATA 95

4.1	Enkripsi dan Dekripsi.....	95
4.1.1	Dasar-dasar Enkripsi dan Dekripsi.....	96
4.1.2	Contoh Penerapan Enkripsi dan Dekripsi	99
4.2	<i>Clustering</i>	104
4.2.1	Dasar-dasar Fuzzy C-Means (FCM)	105
4.2.2	Contoh Penerapan <i>Clustering</i> dengan Fuzzy C-Means (FCM)	114
4.2.3	Membuat Grafik dan <i>Insert</i> Hasil <i>Clustering</i> ke Basis Data	120

4.3 Pengolahan Citra Digital (<i>Digital Image Processing</i>).....	127
4.3.1 Contoh Penerapan Pengolahan Citra Digital dengan Basis Data Microsoft Access.....	128
4.3.2 Contoh Penerapan Pengolahan Citra Digital dengan Basis Data MySQL.....	141
LAMPIRAN	151
A. Dasar-dasar Struktur Data di Matlab	151
A.1 Tipe Data Numerik	152
A.2 Tipe Data <i>String</i>	154
A.3 Tipe <i>Data Cell</i>	156
B. Membuat Menu di GUI Matlab	159
C. Mengompilasi Aplikasi Matlab	167
DAFTAR PUSTAKA	177

BAB

1

Teori Dasar Sistem Basis Data

Teknologi basis data telah berkembang secara pesat dan didominasi oleh sistem basis data relasional. Sistem ini banyak dikembangkan oleh praktisi-praktisi yang biasanya berkecimpung dalam bidang bisnis. Sementara sistem berorientasi obyek, spasial, multimedia, dan sistem spesifik lainnya banyak dikembangkan oleh praktisi-praktisi yang berkecimpung dalam bidang rekayasa.

Sistem basis data relasional telah banyak dibuatkan alat bantu yang dikenal dengan istilah *Computer-aided Software Engineering (CASE)*. Alat bantu ini berfungsi membantu perancang basis data yang telah menggunakan metode manual sebelum ada alat bantu ini. Misalnya untuk perancangan logika, praktisi sudah terbiasa menggunakan aplikasi seperti ERwin Data Modeller, Rational Rose, Enterprise Architect (EA), Open Modelsphere dan sejenisnya yang juga menyediakan fasilitas untuk standar *Unified Modeling Language (UML)*. Database administrator kebanyakan sudah

menggunakan CASE yang biasanya disediakan oleh vendor penyedia *software* basis data seperti Oracle, Microsoft, IBM, dan lain-lain.

1.1 Data dan Manajemen Basis Data

Komponen dasar dari sistem berkas adalah item data yang merupakan representasi dari dunia nyata, misalnya nama depan, nama belakang, alamat, nomor identitas, dan sebagainya. Sekumpulan *item* data membentuk satu kesatuan unit yang dikenal dengan nama *record*. Contoh dari tipe *record* adalah pesanan, daftar sales, pelanggan, produk, departemen, dan sejenisnya. Kumpulan *record* dikumpulkan dan satu berkas. Untuk sistem relasional, item data dikenal dengan istilah kolom atau atribut/*field*, sementara *record* diberi istilah baris atau *tuple*. Berkas sendiri dikenal dengan istilah tabel.

Sekumpulan data menghasilkan suatu informasi. Misalnya data siswa, data hasil belajar, dan sebagainya membentuk satu sistem informasi akademik. Untuk mengatur data-data tersebut dibutuhkan suatu sistem manajemen basis data agar dihasilkan sistem yang tangguh. Beberapa pengembang telah mengembangkan sistem manajemen basis data seperti Oracle, Microsoft (SQL Server dan Access), IBM, dan juga beberapa pengembang lain yang *open source* seperti MySQL, PostgreSQL, dan lain-lain.

Satu sistem manajemen basis data bisa memiliki lebih dari satu basis data (*database*). Satu basis data sendiri memiliki lebih dari satu tabel. Jadi data, basis data, informasi, dan sistem informasi merupakan istilah yang berbeda. Secara fisik antara satu *database* terkadang berada di lokasi yang berbeda dan saling terhubung lewat mekanisme jaringan komputer (*computer network*). Berdasarkan sumbernya, basis data bisa berasal dari transaksi atau dikenal dengan istilah *Online Transaction Processing* (OLTP) atau berasal dari gudang data (*data warehouse*) yang lebih dikenal dengan istilah *Online Analytical Processing* (OLAP) yang biasanya diproses lewat mekanisme *data mining*.

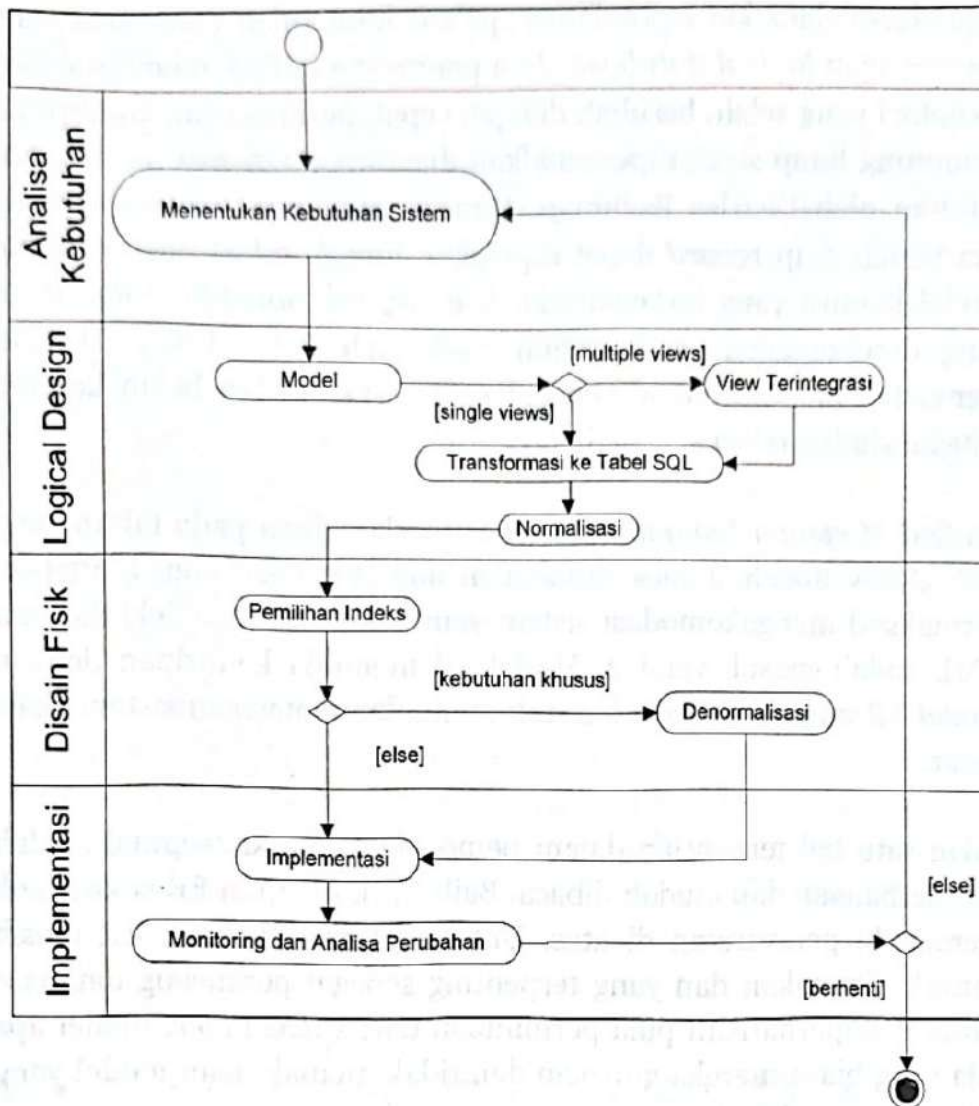
1.2 Siklus Hidup Basis Data (*Database Life Cycle*)

Siklus hidup basis data bermaksud mendukung perancangan basis data dimulai dari perancangan skema logika basis data, pengalokasian data melewati jalur jaringan komputer, serta pendefinisian skema spesifik sistem basis data lokal. Setelah itu proses implementasi dan perawatan (*maintenance*) bermaksud menjaga keandalan sistem basis data yang telah direncanakan sebelumnya. Jadi, siklus hidup basis data antara lain: menentukan kebutuhan informasi, perancangan logika, perancangan fisik, dan implementasi (Toerey, et al, 2011).

- a. Analisis Kebutuhan (*requirement analysis*). Tahap ini dilakukan dengan mewawancarai baik pemasok dan pengguna data serta informasi-informasi lainnya untuk menghasilkan spesifikasi formal kebutuhan. Spesifikasi-spesifikasi tersebut adalah: data yang dibutuhkan untuk diproses, hubungan antardata, dan platform perangkat lunak yang diperlukan.
- b. *Logical Design*. Dimulai dari perancangan skema global dan dengan konsep model data hingga konversi dari model menjadi tabel-tabel. Model harus bisa menunjukkan hubungan-hubungan antardata dan biasanya dikembangkan dengan *entity-relationship* (ER) atau UML. Tahap ini sangat penting sebelum masuk ke tahap berikutnya. Tahapan yang biasanya dilalui oleh perancang basis data di fase ini adalah: *Conceptual data modeling*, *View integration*, Transformasi ke tabel SQL dan normalisasi tabel.
- c. *Physical design*. Tahap ini mulai merealisasikan rancangan yang telah dibuat pada tahap sebelumnya ke sistem dengan melibatkan pemilihan indeks, partisi, kluster terhadap data. Sebisanya mungkin tahapan ini membuat rancangan pada tahap *logic* menjadi nyata dalam suatu sistem. Pada tahap fisik ini dilakukan juga proses optimasi agar dihasilkan sistem yang cepat dan mampu menangani komputasi untuk data real yang besar. Terkadang diperlukan proses denormalisasi untuk mengurangi proses join yang membutuhkan proses yang banyak. Waktu yang diperlukan untuk tiap query dicatat untuk mengetahui kinerjanya. Penting

untuk memperhatikan integritas data agar terhindar dari kesalahan saat mengolah data, terutama untuk sistem *online analytical processing* (OLAP). OLAP merupakan sistem yang berfokus ke analisa data yang berada di *database* yang berukuran besar, berbeda dengan sistem transaksi yang hanya sebatas keluaran dan masukan dengan komputasi sederhana.

- d. *Database implementation, monitoring, and modification.* Ini merupakan tahap terakhir yang tidak kalah pentingnya dengan tahap sebelumnya. Implementasi menyiapkan tabel-tabel yang akan digunakan oleh pengguna baik dengan menggunakan *data definition language* (DDL) agar siap diolah dengan *data manipulation language* (DML). Ketika *database* mulai beroperasi, tahap monitoring dimulai untuk memantau apakah performa dan kebutuhan sistem telah terpenuhi. Jika tidak memenuhi maka perlu dilakukan modifikasi agar sesuai dengan kebutuhan sistem.



Gambar 1.1. Siklus Hidup Basis Data

1.3 Pemodelan Data Konseptual (*Conceptual Data Modeling*)

Sebelum basis data terbentuk, dilakukan terlebih dahulu pemodelan yang nantinya dengan mudah dikonversi menjadi data fisik beserta perangkat lunak yang digunakannya. Tahap ini sangat menentukan karena jika salah di tahap ini maka implementasi ke basis data fisik akan salah juga dan akan menghabiskan biaya yang besar.

Pemodelan data konseptual merupakan komponen penggerak dari perancangan *logical database*. Jika perancangan fisik membutuhkan teknologi yang selalu berubah dengan cepat, perancangan konseptual cenderung tetap sejak diperkenalkan diagram skemanya pada tahun 1960-an oleh Charles Bachman. Dengan menggunakan segi empat dan panah, tiap *record* dapat dipetakan hingga relasi *one-to-many*. Model lainnya yang terkenal adalah *entity-relationship* (ER) model yang diperkenalkan pada tahun 1976 oleh Peter Chen. Dengan menggunakan segi empat sebagai *record*, garis, dan belah ketupat sebagai simbol relasi antara dua *record*.

Unified Modeling Language (UML) diperkenalkan pada tahun 1997 oleh Grady Booch, James Rumbaugh dan Ivar Jacobson. Model ini bermaksud mengakomodasi sistem yang berorientasi objek. Saat ini UML sudah masuk versi 2. Model ini memiliki kemiripan dengan model ER dan sangat cocok untuk mendokumentasikan sistem skala besar.

Salah satu hal terpenting dalam pemodelan data konseptual adalah kesederhanaan dan mudah dibaca. Baik UML maupun ER cukup baik memenuhi persyaratan di atas. Untuk sistem *database*, ER masih banyak digunakan dan yang terpenting sebagai perancang database harus memperhatikan pula permintaan dari *stakeholder*, model apa saja yang biasa mereka gunakan dan tidak memaksakan model yang sulit mereka pahami.

1.4 Pemodelan Data Fisik (*Physical Data Modeling*)

Sistem relasional memiliki langkah-langkah yang standar dan biasa dilakukan yaitu mengonversi disain logika berupa model ER (atau sering disebut ER Diagram/ERD) menjadi struktur tabel lengkap dengan atribut/*field*-nya (sering diistilahkan dengan konversi ERD ke *Logical Record Structure/LRS*). Normalisasi sangat diperlukan untuk

menghindari adanya anomali-anomali dan redundansi yang terjadi ketika mengonversi menjadi tabel.

Tahap ini memiliki tanggung jawab apakah implementasi basis data ke perangkat lunak sudah sesuai dengan fase sebelumnya (*logical*). Beberapa alat bantu (CASE) dijumpai di pasaran yang dengan tepat mampu mengkonversi tabel yang berasal dari model ER menjadi tabel fisik di perangkat lunak. Rational Rose, Datanamic Deziqn, System Architech, dan lain-lain merupakan contoh CASE yang membantu pengembang mengonversi model (ER atau UML) menjadi basis data fisik, bahkan bahasa pemrograman tertentu memiliki *Integrated Development Environment (IDE)* yang sudah menyertakan CASE untuk menghasilkan satu basis data fisik, misalnya Netbeans yang berbasis Java.

Selain itu, fase pemodelan fisik bertanggung jawab juga terhadap keamanan, kecepatan, dan *backup system*. Penggunaan sistem indeks dan denormalisasi dapat mempercepat waktu akses terhadap basis data. *Backup system* dan *sistem redundancy*, misalnya RAID, diperlukan untuk mengamankan data ketika terjadi kerusakan pada sistem.

1.5 *Structure Query Language (SQL)*

Sistem relasional memiliki langkah-langkah yang standar dan biasa dilakukan yaitu mengonversi disain logika berupa model ER (atau sering disebut ER Diagram/ERD) menjadi struktur tabel lengkap dengan atribut/field-nya (sering diistilahkan dengan konversi ERD ke *Logical Record Structure/LRS*). Normalisasi sangat diperlukan untuk menghindari adanya anomali-anomali dan redundansi yang terjadi ketika mengonversi menjadi tabel-tabel. Tabel-tabel yang telah terbentuk dan sudah tidak ada perubahan yang berarti lagi kemudian dikonversi menjadi *file-file* dalam sistem basis data dengan menggunakan bahasa standar *Structure Query Language (SQL)*.

Ketika mempublikasi tulisan tentang *Teori Database Relational*, Dr. Edgar F. Codd di tahun 1969 mengajukan sebuah bahasa yang disebut DSL/Alpha. Bahasa ini untuk mengelola data dalam basis data bertipe relational. Berdasarkan ide Dr.Codd ini, setelah itu IBM mencoba merancang bahasa prototipe sederhana DSL/Alpha yang disebut SQUARE.

Team yang beranggotakan peneliti IBM Donald D. Chamberlin dan Raymond F. Boyce, mengembangkan *SQUARE*, yang kemudian berubah namanya menjadi *SEQUEL (Structured English Query Language)* di tahun 1970. *SEQUEL* digunakan untuk mengoperasikan prototipe RDBMS pertama IBM, *System R*. Dikemudian hari, *SEQUEL* berubah nama menjadi SQL karena permasalahan merk dagang (*trademark*) dengan sebuah perusahaan pesawat di Inggris yang terlebih dahulu telah memakai nama *SEQUEL*. Di tahun yang sama, perusahaan *Relational Software, Inc.* (yang sekarang dikenal dengan nama Oracle Corporation) melihat potensi bahasa SQL dan mengembangkan sendiri versi SQL untuk RDBMS mereka. *Oracle V2* (versi 2) yang dirilis Juni 1979 adalah RDBMS komersial pertama yang mengimplementasikan SQL.

Dengan kemudahan yang ditawarkan, SQL mulai diimplementasikan oleh berbagai *Relational Database Management System (RDBMS)* dengan versi SQL mereka masing-masing. Namun hal ini menimbulkan permasalahan karena perbedaan penerapan SQL dari satu aplikasi dengan aplikasi database lainnya yang tidak seragam. Sehingga pada tahun 1986, badan standar Amerika, *ANSI (American National Standards Institute)* merancang sebuah standar untuk SQL. Satu tahun setelahnya, *ISO (International Organization for Standardization)* juga mengeluarkan standar untuk SQL. Dengan standar ini diharapkan ada keseragaman SQL antar-aplikasi RDBMS.

Namun demikian, walaupun sudah ada standar tentang SQL, banyak perusahaan RDBMS yang menambahkan 'fitur' SQL selain standar yang ada. PostgreSQL juga memiliki SQL yang tidak standar, yang tidak ada pada Oracle, begitu juga sebaliknya. Namun setidaknya

bahasa SQL hampir sama untuk perintah-perintah dasar antar-RDBMS. Perintah SQL untuk membuat tabel misalnya, dapat digunakan baik di Oracle, PostgreSQL, MySQL, maupun sistem RDBMS yang lainnya.

Dalam membangun suatu aplikasi komputer yang berbasis RDBMS, terdapat suatu struktur bahasa yang sudah standar untuk membangun basis datanya. Standar bahasa tersebut adalah *Structure Query Language* (selanjutnya disingkat SQL). Standar ini digunakan ketika membangun basis data berdasarkan tabel-tabel yang sudah dibentuk sebelumnya. Hal ini dapat dilakukan melalui suatu konversi dari tabel menjadi bahasa program, dengan menggunakan aturan umum dalam perintah SQL.

Perintah atau instruksi SQL dapat dikelompokkan berdasarkan jenis dan fungsinya. Terdapat 3 jenis perintah dasar SQL : *Data Definition Language*, *Data Manipulation Language* dan *Data Control Language*.

1. *Data Definition Language (DDL)* adalah jenis instruksi SQL yang berkaitan dengan pembuatan struktur tabel maupun *database*. Termasuk diantaranya : CREATE, DROP, ALTER, dan RENAME.
2. *Data Manipulation Language (DML)* adalah jenis instruksi SQL yang berkaitan dengan data yang ada dalam tabel, tentang bagaimana menginput, menghapus, memperbarui serta membaca data yang tersimpan di dalam *database*. Contoh perintah SQL untuk DML: SELECT, INSERT, DELETE, dan UPDATE.
3. *Data Control Language (DCL)* adalah jenis instruksi SQL yang berkaitan dengan manajemen hak akses dan pengguna (*user*) yang dapat mengakses *database* maupun tabel. Termasuk di antaranya: GRANT dan REVOKE.

Selain ketiga jenis perintah SQL, terdapat juga 2 jenis SQL tambahan: *Transaction Control Language*, dan *Programmatic SQL*.

1. *Transaction Control Language (TCL)* adalah perintah SQL untuk proses transaksi. Proses transaksi ini digunakan untuk perintah yang lebih dari satu, namun harus berjalan semua, atau tidak sama sekali. Misalnya untuk aplikasi *critical* seperti transfer uang

dalam sistem database perbankan. Setidaknya akan ada 2 perintah, yaitu mengurangi uang nasabah A, dan menambah uang nasabah B. Namun jika terjadi kesalahan sistem, kedua transaksi ini harus dibatalkan. Tidak bisa hanya satu perintah saja. Termasuk ke dalam TCL adalah perintah: COMMIT, ROLLBACK, dan SET TRANSACTION.

2. *Programmatic SQL* berkaitan dengan sub program (*stored procedure*) maupun penjelasan mengenai struktur *database*. Contoh perintah seperti: DECLARE, EXPLAIN, PREPARE, dan DESCRIBE.

Seperti bahasa pemrograman lain, perintah SQL juga memiliki aturan-aturan tertentu yang khas antara lain:

1. Perintah dapat ditulis dalam huruf besar maupun kecil.
2. Setiap perintah diakhiri dengan tanda titik koma “;”
3. Perintah dapat ditulis satu atau beberapa baris untuk memberikan komentar pada perintah SQL. Gunakan tanda minus “-” untuk komentar 1 baris, atau di antara tanda “*” dan “*\” untuk komentar beberapa baris.

1.5.1 *Data Definition Language (DDL)*

Data Definition Language (DDL) merupakan perintah SQL yang digunakan untuk melakukan definisi awal suatu basis data dan tabel pada konsep RDBMS. Secara sederhana, penulisan perintah SQL pada kelompok ini terdiri dari *create*, *alter* dan *drop*. Berikut ini merupakan pembahasan untuk perintah SQL tersebut.

1. *Create*

Perintah SQL ini digunakan untuk membuat suatu basis data dan tabel pendukung di dalam pembangunan basis data tersebut. Tabel merupakan objek yang sangat penting dalam suatu basis data.

Berikut ini merupakan aturan umum penulisan yang digunakan pada *create database*.

CREATE DATABASE nama basis data;

Contoh: CREATE DATABASE sekolah;

Keterangan: Membuat *database* baru dengan nama sekolah

Berikut ini merupakan aturan umum penulisan yang digunakan pada *create table*.

```
CREATE TABLE namatabel
(
    Field, data type, length, attribute key
)
```

Contoh: create table siswa (nis varchar(15), nama varchar(35), kelas int(3));

Keterangan: Membuat tabel baru dengan nama siswa yang terdiri dari 3 fields/kolom. Kolom nis bertipe *string*, kolom nama bertipe *string* dan kelas bertipe *integer*.

2. *Alter*

Perintah SQL ini digunakan untuk mengubah struktur tabel yang terdapat di dalam basis data. Hal ini dapat saja terjadi, jika ingin melakukan penambahan atau penghapusan suatu *field* atau atribut *key* (*primary/foreign*), tanpa mendefinisikan struktur baru pada tabel. Melalui perintah SQL ini, jika sudah terisikan *record* pada tabel tersebut, maka mengenai hilangnya *record* yang terdapat pada suatu tabel tidak akan terjadi.

Berikut ini merupakan aturan umum penulisan yang digunakan pada *alter table*.

```
ALTER TABLE nama tabel ADD nama field tipe data(size);
```

Contoh : ALTER TABLE siswa ADD alamat varchar(50);

Keterangan : Menambahkan kolom baru dengan nama alamat bertipe *string* panjang 50 karakter.

```
ALTER TABLE nama tabel CHANGE nama field yang diganti
nama field baru tipe data(size);
```

Contoh: ALTER TABLE siswa CHANGE alamat alamat_rumah varchar(50);

Keterangan: Mengubah kolom alamat menjadi alamat_rumah bertype string dengan panjang 50 karakter.

ALTER TABLE nama tabel DROP nama field;

Contoh : alter table siswa drop alamat_rumah;

Keterangan: Menghapus Kolom alamat_rumah dari tabel siswa.

3. Drop

Perintah SQL ini digunakan untuk menghapus tabel yang terdapat di dalam basis data. Bisa digunakan juga untuk menghapus basis data.

Berikut ini merupakan aturan umum penulisan yang digunakan pada *drop* tabel.

DROP TABEL nama tabel

Contoh : DROP TABEL siswa;

Keterangan : Menghapus tabel siswa

Berikut ini merupakan aturan umum penulisan yang digunakan pada drop basis data.

DROP DATABASE nama basis data

Contoh : DROP DATABASE sekolah;

Keterangan : Menghapus *database* sekolah

4. Rename

Perintah SQL ini digunakan untuk mengganti nama tabel yang terdapat di dalam basis data. Berikut ini merupakan aturan umum penulisan yang digunakan pada *rename* nama tabel.

RENAME nama tabel yang diganti TO nama tabel baru;

Contoh : RENAME siswa TO mahasiswa;

Keterangan : Mengubah tabel siswa menjadi mahasiswa



1.5.2 Data Manipulation Language (MDL)

Data Manipulation Language (MDL), merupakan perintah SQL yang digunakan untuk melakukan pengolahan *record* atau manipulasi *record* pada tabel dalam suatu basis data. Secara sederhana, penulisan perintah SQL pada kelompok ini terdiri dari *insert*, *select*, *update*, dan *delete*. Berikut ini adalah deskripsi mengenai kelompok perintah DML tersebut.

1. *Insert*

Perintah SQL ini digunakan untuk melakukan entry atau penambahan suatu *record* (informasi/data baru) pada tabel dalam basis data. Berikut ini merupakan aturan umum penulisan yang digunakan pada insert suatu tabel.

```
INSERT INTO nama tabel VALUES (data-data nilai yang diisikan pada tabelnya)
```

Contoh:

```
INSERT INTO siswa (nis,nama,kelas) VALUES ('103362','LADYA RAMADHANI','3C'); atau
```

```
INSERT INTO siswa VALUES ('103362','LADYA RAMADHANI','3C');
```

2. *Select*

Perintah SQL ini digunakan untuk memilih *record* yang akan ditampilkan berdasarkan data pada tabel dalam basis data. Berikut ini merupakan aturan umum dan variasi penulisan yang digunakan pada *select*.

```
SELECT nama field, nama field, ... FROM nama tabel;
```

```
SELECT nama field, nama field, ... FROM nama tabel WHERE kondisi yang diinginkan;
```

```
SELECT nama field, nama field, ... FROM nama tabel ORDER BY aturan urutan data;
```

Berikut ini contoh-contoh penerapannya:

```
SELECT nis,nama,kelas FROM siswa;
```

```
SELECT * FROM siswa;  
SELECT nis,nama,kelas FROM siswa WHERE kelas = '4D';  
SELECT * FROM siswa WHERE nama like 'AN%' ORDER BY  
nama ASC;
```

Keterangan:

Perintah WHERE merupakan Optional tidak harus disertakan seperti SELECT dan FROM.

Contoh 'a' menampilkan semua informasi nis,nama,kelas dari data siswa.

Contoh 'b' menampilkan semua data siswa.

Contoh 'c' menampilkan semua informasi nis,nama,kelas dari data siswa kelas 4D.

Contoh 'd' menampilkan semua data siswa yang namanya diawali dengan "AN" dan diurutkan berdasarkan nama secara Ascending.

3. *Update*

Perintah SQL untuk mengubah data dalam suatu tabel pada *field* tertentu, dengan *record* baru berdasarkan suatu *field* sebagai kriteria perubahan *record*-nya. Berikut ini merupakan aturan umum penulisan yang digunakan pada *update*.

```
UPDATE nama tabel  
SET nama field = record yang diubah  
WHERE kondisi field yang dijadikan patokan;  
Contoh: UPDATE siswa SET nama='Dimas' WHERE  
nis='103361';
```

Keterangan: Mengganti nama menjadi "Dimas" untuk siswa yang mempunyai nis "103361"

4. *Delete*

Perintah SQL ini digunakan untuk menghapus data dalam suatu tabel, berdasarkan suatu *field* sebagai kriteria penghapusan



record-nya. Berikut ini merupakan aturan umum penulisan yang digunakan pada *delete*.

DELETE FROM nama tabel WHERE kondisi *field* yang dijadikan patokan;

Contoh:

DELETE FROM siswa WHERE nama = 'Ladya Ramadhani';

Keterangan:

Menghapus data siswa yang memiliki nama "Ladya Ramadhani"
Perintah yang digunakan untuk manipulasi data seperti menambah, mengedit, menghapus data. Perintah yang digunakan INSERT, DELETE, UPDATE, MERGE.

1.5.3 Data Control Language (DCL)

Data Control Language (DCL) merupakan perintah SQL yang digunakan untuk melakukan pengaturan hak akses suatu objek data para pengguna dalam basis data. Secara sederhana, penulisan perintah SQL pada kelompok ini terdiri dari *grant* dan *revoke*.

1. *Grant*

Perintah SQL ini, digunakan oleh seorang administrator basis data untuk memberikan hak aksesnya kepada pengguna tertentu, agar dapat mengakses suatu tabel dalam basis data. Hak akses tersebut adalah *insert*, *delete*, *update*, dan *select*. Berikut ini merupakan aturan umum penulisan yang digunakan pada *grant*.

grant hak akses
on nama tabel
to pengguna tertentu;

2. *Revoke*

Perintah SQL ini digunakan oleh seorang administrator basis data, untuk membatalkan/menghentikan hak akses yang telah diberikan kepada pengguna tertentu, agar tidak dapat mengakses tabel dalam basis data. Berikut ini merupakan aturan umum penulisan yang digunakan pada *revoke*.

Daftar Pustaka

- Bramer, Max. 2007. *Principles of Data Mining*. London: Springer – Verlag.
- Elmasri, Ramez, Shamkant B. Navathe. 2003. *Fundamentals of Database Systems*. New York: Pearson Education, Inc.
- <http://www.mathworks.com>
- <https://www.apachefriends.org>
- Gorunescu, Florin. 2011. *Data Mining – Concepts, Models, and Techniques*. Berlin: Springer – Verlag.
- Han, Jiawei, Michelin Kamber. 2006. *Data Mining: Concepts and Techniques – 2nd Edition*. New Jersey: Prentice Hall.
- Kadir, Abdul, Adhi Susanto. 2013. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Penerbit Andi.
- Larose, Daniel T. 2006. *Data Mining, Methods and Models*. New Jersey: John Wiley & Sons.
- Miyamoto, Sadaaki, Hideotomo Ichihashi, Katsuhiro Honda. 2008. *Algorithms for Fuzzy Clustering – Methods C-Means Clustering with Applications*. Germany: Springer – Verlag.
- Toerey, Toby, Sam Lightstone, Tom Nadeau, H. V. Jagadish. 2011. *Database Modeling and Design Fifth Edition*. USA: Elsevier, Inc.
- Widodo, Prabowo Pudjo, Rahmadya Trias Handayanto, Herlawati. 2013. *Penerapan Data Mining dengan Matlab*. Bandung: Rekayasa Sains.