MODUL SITEM INFORMASI MANAGEMEN

(MAN 611)

MODUL PERTEMUAN 08

Foundation Business Intelligence

DISUSUN OLEH

Dr. Fransiskus Adikara, S.Kom, MMSI

UNIVERSITAS ESA UNGGUL

2019

FOUNDATION OF DATABASE AND BI

1. Kemampuan Akhir Yang Diharapkan

After reading this session, you will be able to answer the following questions:

1. What are the problems of managing data resources in a traditional file environment and how are they solved by a database management system?
2. What are the major capabilities of database management systems (DBMS) and why is a relational DBMS so powerful?
3. What are some important principles of database design?

2. Uraian dan Contoh

1. ORGANIZING DATA IN A TRADITIONAL FILE ENVIRONMENT

An effective information system provides users with accurate, timely, and relevant information. Accurate information is free of errors. Information is timely when it is available to decision makers when it is needed. Information is relevant when it is useful and appropriate for the types of work and decisions that require it.

You might be surprised to learn that many businesses don't have timely, accurate, or relevant information because the data in their information systems have been poorly organized and maintained. That's why data management is so essential. To understand the problem, let's look at how information systems arrange data in computer files and traditional methods of file management.

FILE ORGANIZATION TERMS AND CONCEPTS

A computer system organizes data in a hierarchy that starts with bits and bytes and progresses to fields, records, files, and databases (see Figure 6.1). A **bit** represents the smallest unit of data a computer can handle. A group of bits, called a **byte**, represents a single character, which can be a letter, a number, or another symbol. A grouping of characters into a word, a group of words, or a complete number (such as a person's name or age) is called a **field**. A group of related fields, such as the student's name, the course taken, the date, and the grade, comprises a **record**; a group of records of the same type is called a **file**.

For example, the records in Figure 6.1 could constitute a student course file. A group of related files makes up a database. The student course file illustrated in Figure 6.1 could be grouped with files on students' personal histories and financial backgrounds to create a student database.

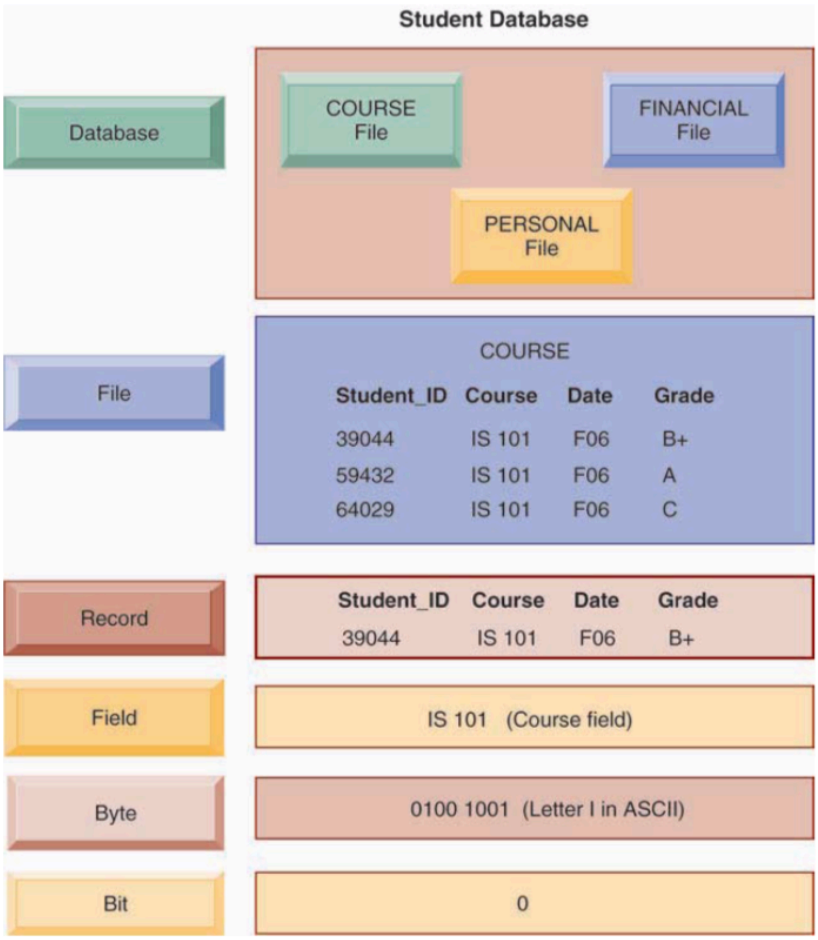
A record describes an entity. An **entity** is a person, place, thing, or event on which we store and maintain information. Each characteristic or quality describ- ing a particular entity is called an **attribute**. For example, Student_ID, Course, Date, and Grade are attributes of the entity COURSE. The specific values that these attributes can have are found in the fields of the record describing the entity COURSE.

PROBLEMS WITH THE TRADITIONAL FILE

ENVIRONMENT

In most organizations, systems tended to grow independently without a company-wide plan. Accounting, finance, manufacturing, human resources, and sales and marketing all developed their own systems and data files. Figure 6.2 illustrates the traditional approach to information processing.

FIGURE 6.1 THE DATA HIERARCHY



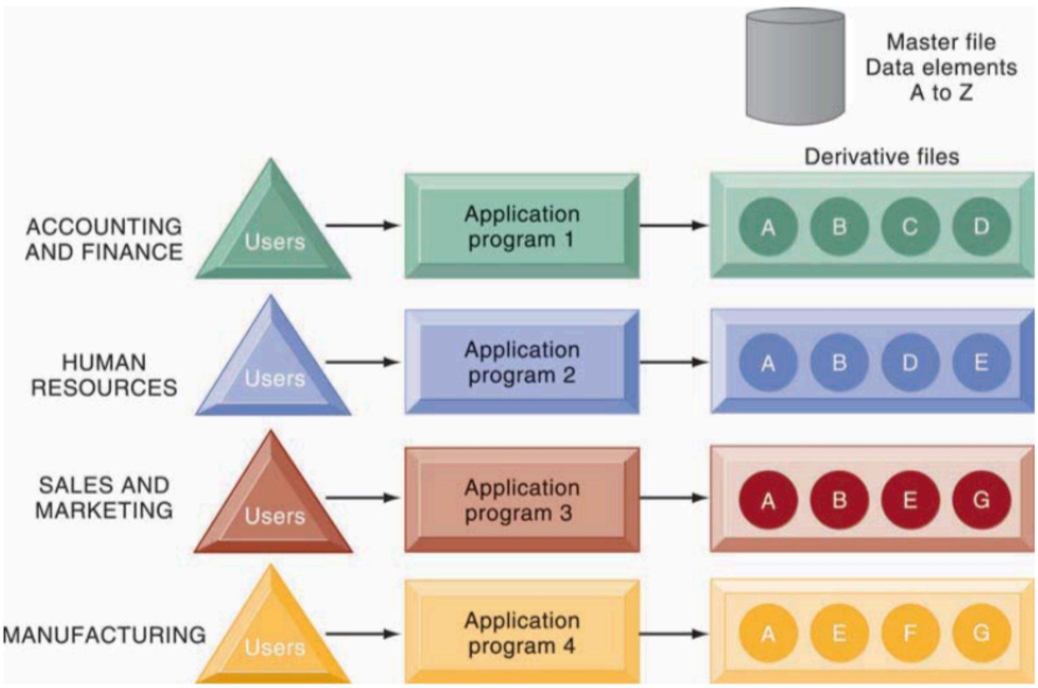
A computer system organizes data in a hierarchy that starts with the bit, which represents either a 0 or a 1. Bits can be grouped to form a byte to represent one character, number, or symbol. Bytes can be grouped to form a field, and related fields can be grouped to form a record. Related records can be collected to form a file, and related files can be organized into a database.

Each application, of course, required its own files and its own computer program to operate. For example, the human resources functional area might have a personnel master file, a payroll file, a medical insurance file, a pension file, a mailing list file, and so forth until tens, perhaps hundreds, of files and programs existed. In the company as a whole, this process led to multiple master files created, maintained, and operated by separate divisions or depart- ments. As this process goes on for 5 or 10 years, the organization is saddled with hundreds of programs and applications that are very difficult to maintain and manage. The resulting problems are data redundancy and inconsistency, program-data dependence, inflexibility, poor data security, and an inability to share data among applications.

Data Redundancy and Inconsistency

Data redundancy is the presence of duplicate data in multiple data files so that the same data are stored in more than one place or location. Data redun- dancy occurs when different groups in an organization independently collect the same piece of data and store it independently of each other. Data redun- dancy wastes storage resources and also leads to **data inconsistency**, where the same attribute may have different values. For example, in instances of the entity COURSE illustrated in Figure 6.1, the Date may be updated in some systems but not in others. The same attribute, Student_ID, may also have different names in different systems throughout the organization. Some systems might use Student_ID and others might use ID, for example.

FIGURE 6.2 TRADITIONAL FILE PROCESSING



The use of a traditional approach to file processing encourages each functional area in a corporation to develop specialized applications. Each application requires a unique data file that is likely to be a subset of the master file. These subsets of the master file lead to data redundancy and inconsistency, processing inflexibility, and wasted storage resources.

Additional confusion might result from using different coding systems to represent values for an attribute. For instance, the sales, inventory, and manufacturing systems of a clothing retailer might use different codes to represent clothing size. One system might represent clothing size as “extra large,” whereas another might use the code “XL” for the same purpose. The resulting confusion would make it difficult for companies to create customer relationship management, supply chain management, or enterprise systems that integrate data from different sources.

Program-Data Dependence

Program-data dependence refers to the coupling of data stored in files and the specific programs required to update and maintain those files such that changes in programs require changes to the data. Every traditional computer program has to describe the location and nature of the data with which it works. In a traditional file environment, any change in a software program could require a change in the data accessed by that program. One program might be modified from a five-digit to a nine-digit zip code. If the original data file were changed from five-digit to nine-digit zip codes, then other programs that required the five-digit zip code would no longer work properly. Such changes could cost millions of dollars to implement properly.

Lack of Flexibility

A traditional file system can deliver routine scheduled reports after extensive programming efforts, but it cannot deliver ad hoc reports or respond to unantic- ipated information requirements in a timely fashion. The information required by ad hoc requests is somewhere in the system but may be too expensive to retrieve. Several programmers might have to work for weeks to put together the required data items in a new file.

Poor Security

Because there is little control or management of data, access to and dissemina- tion of information may be out of control. Management may have no way of knowing who is accessing or even making changes to the organization's data.

Lack of Data Sharing and Availability

Because pieces of information in different files and different parts of the organization cannot be related to one another, it is virtually impossible for information to be shared or accessed in a timely manner. Information cannot flow freely across different functional areas or different parts of the organi- zation. If users find different values of the same piece of information in two different systems, they may not want to use these systems because they cannot trust the accuracy of their data.

2. THE DATABASE APPROACH TO DATA MANAGEMENT

Database technology cuts through many of the problems of traditional file organization. A more rigorous definition of a **database** is a collection of data organized to serve many applications efficiently by centralizing the data and controlling redundant data. Rather than storing data in separate files for each application, data appears to users as being stored in only one location. A single database services multiple applications. For example, instead of a corporation storing employee data in separate information systems and separate files for personnel, payroll, and benefits, the corporation could create a single common human resources database.

DATABASE MANAGEMENT SYSTEMS

A **database management system (DBMS)** is software that permits an organization to centralize data, manage them efficiently, and provide access to the stored data by application programs. The DBMS acts as an interface between application programs and the physical data files. When the applica- tion program calls for a data item, such as gross pay, the DBMS finds this item in the database and presents it to the application program. Using traditional data files, the programmer would have to specify the size and format of each data element used in the program and then tell the computer where they were located.

The DBMS relieves the programmer or end user from the task of under- standing where and how the data are actually stored by separating the logical and physical views of the data. The *logical view* presents data as they would be perceived by end users or business specialists, whereas the *physical view* shows how data are actually organized and structured on physical storage media.

The database management software makes the physical database available for different logical views required by users. For example, for the human resources database illustrated in Figure 6.3, a benefits specialist might require a view consisting of the employee's name, social security number, and health insurance coverage. A payroll department member might need data such as the employee's name, social security number, gross pay, and net pay. The data for all these views are stored in a single database, where they can be more easily managed by the organization.

FIGURE 6.4 RELATIONAL DATABASE TABLES

SUPPLIER

Columns (Attributes, Fields)

Supplier_Number	Supplier_Name	Supplier_Street	Supplier_City	Supplier_State	Supplier_Zip
8259	CBM Inc.	74 5th Avenue	Dayton	OH	45220
8261	B. R. Molds	1277 Gandolly Street	Cleveland	OH	49345
8263	Jackson Composites	8233 Micklin Street	Lexington	KY	56723
8444	Bryant Corporation	4315 Mill Drive	Rochester	NY	11344

Rows (Records, Tuples)

Key Field (Primary Key)

PART

Part_Number	Part_Name	Unit_Price	Supplier_Number
137	Door latch	22.00	8259
145	Side mirror	12.00	8444
150	Door molding	6.00	8263
152	Door lock	31.00	8259
155	Compressor	54.00	8261
178	Door handle	10.00	8259

Primary Key

Foreign Key

A relational database organizes data in the form of two-dimensional tables. Illustrated here are tables for the entities SUPPLIER and PART showing how they represent each entity and its attributes. Supplier_Number is a primary key for the SUPPLIER table and a foreign key for the PART table.

How a DBMS Solves the Problems of the Traditional File Environment
A DBMS reduces data redundancy and inconsistency by minimizing isolated files in which the same data are repeated. The DBMS may not enable the organization to eliminate data redundancy entirely, but it can help control redundancy. Even if the

organization maintains some redundant data, using a DBMS eliminates data inconsistency because the DBMS can help the organization ensure that every occurrence of redundant data has the same values. The DBMS uncouples programs and data, enabling data to stand

n their own. Access and availability of information will be increased and program development and maintenance costs reduced because users and programmers can perform ad hoc queries of data in the database. The DBMS enables the organization to centrally manage data, their use, and security.

Relational DBMS

Contemporary DBMS use different database models to keep track of entities, attributes, and relationships. The most popular type of DBMS today for PCs as well as for larger computers and mainframes is the **relational DBMS**. Relational databases represent data as two-dimensional tables (called relations). Tables may be referred to as files. Each table contains data on an entity and its attributes. Microsoft Access is a relational DBMS for desktop systems, whereas DB2, Oracle Database, and Microsoft SQL Server are relational DBMS for large mainframes and midrange computers. MySQL is a popular open source DBMS, and Oracle Database Lite is a DBMS for mobile computing devices.

Let's look at how a relational database organizes data about suppliers and parts (see Figure 6.4). The database has a separate table for the entity SUPPLIER and a table for the entity PART. Each table consists of a grid of columns and rows of data. Each individual element of data for each entity is stored as a separate field, and each field represents an attribute for that entity. Fields in a relational database are also called columns. For the entity SUPPLIER, the supplier identification number, name, street, city, state, and zip code are stored as separate fields within the SUPPLIER table and each field represents an attribute for the entity SUPPLIER.

The actual information about a single supplier that resides in a table is called a row. Rows are commonly referred to as records, or in very technical terms, as **tuples**. Data for the entity PART have their own separate table.

The field for Supplier_Number in the SUPPLIER table uniquely identifies each record so that the record can be retrieved, updated, or sorted. It is called a **key field**. Each table in a relational database has one field that is designated as its **primary key**. This key field is the unique identifier for all the information in any row of the table and this primary key cannot be duplicated. Supplier_ Number is the primary key for the SUPPLIER table and Part_Number is the primary key for the PART table. Note that Supplier_Number appears in both the SUPPLIER and PART tables. In the SUPPLIER table, Supplier_Number is the primary key. When the field Supplier_Number appears in the PART table, it is called a **foreign key** and is essentially a lookup field to look up data about the supplier of a specific part.

Operations of a Relational DBMS

Relational database tables can be combined easily to deliver data required by users, provided that any two tables share a common data element. Suppose we wanted to find in this database the names of suppliers who could provide us with part number

137 or part number 150. We would need information from two tables: the SUPPLIER table and the PART table. Note that these two files have a shared data element: Supplier_Number.

In a relational database, three basic operations, as shown in Figure 6.5, are used to develop useful sets of data: select, join, and project. The *select* operation creates a subset consisting of all records in the file that meet stated criteria. Select creates, in other words, a subset of rows that meet certain criteria. In our example, we want to select records (rows) from the PART table where the Part_Number equals 137 or 150. The *join* operation combines relational tables to provide the user with more information than is available in individual tables. In our example, we want to join the now-shortened PART table (only parts 137 or 150 will be presented) and the SUPPLIER table into a single new table.

The *project* operation creates a subset consisting of columns in a table, permit- ting the user to create new tables that contain only the information required. In our example, we want to extract from the new table only the following columns: Part_Number, Part_Name, Supplier_Number, and Supplier_Name.

Non-Relational Databases and Databases in the Cloud

For over 30 years, relational database technology has been the gold standard. Cloud computing, unprecedented data volumes, massive workloads for Web services, and the need to store new types of data require database alterna- tives to the traditional relational model of organizing data in the form of tables, columns, and rows. Companies are turning to "*NoSQL*" non-relational database technologies for this purpose. **Non-relational database management systems** use a more flexible data model and are designed for managing large data sets across many distributed machines and for easily scaling up or down. They are useful for accelerating simple queries against large volumes of structured and unstructured data, including Web, social media, graphics, and other forms of data that are difficult to analyze with traditional SQL-based tools.

There are several different kinds of NoSQL databases, each with its own technical features and behavior. Oracle NoSQL Database is one example, as is Amazon's SimpleDB, one of the Amazon Web Services that run in the cloud. SimpleDB provides a simple Web services interface to create and store multiple data sets, query data easily, and return the results. There is no need to pre-define a formal database structure or change that definition if new data are added later.

Amazon and other cloud computing vendors provide relational database services as well. Amazon Relational Database Service (Amazon RDS) offers MySQL, SQL Server, or Oracle Database as database engines. Pricing is based on usage. Oracle has its own Database Cloud Service using its relational Oracle Database 11g, and Microsoft SQL Azure Database is a cloud-based relational database service based on Microsoft's SQL Server DBMS. Cloud-based data management services have special appeal for Web-focused start-ups or small to medium-sized businesses seeking database capabilities at a lower price than in-house database products.

TicketDirect, which sells tickets to concerts, sporting events, theater performances, and movies in Australia and New Zealand, adopted the SQL Azure Database cloud platform in order to improve management of peak system loads during major ticket sales. It migrated its data to the SQL Azure database. By moving to a cloud solution, TicketDirect is able to scale its computing resources in response to real-time demand while keeping costs low.

In addition to public cloud-based data management services, companies now have the option of using databases in private clouds. For example, Sabre Holdings, the world's largest software as a service (SaaS) provider for the aviation industry, has a private database cloud that supports more than 100 projects and 700 users. A consolidated database spanning a pool of stan- dardized servers running Oracle Database 11g provides database services for multiple applications. Workload management tools ensure sufficient resources are available to meet application needs even when the workload changes. The shared hardware and software platform reduces the number of servers, DBMS, and storage devices needed for these projects, which consist of custom airline travel applications along with rail, hotel, and other travel industry applications (Baum, 2011).

Private clouds consolidate servers, storage, operating systems, databases, and mixed workloads onto a shared hardware and software infrastructure. Deploying databases on a consolidated private cloud enables IT departments to improve quality of service levels and reduce capital and operating costs. The higher the consolidation density achieved, the greater the return on investment.

CAPABILITIES OF DATABASE MANAGEMENT SYSTEMS

A DBMS includes capabilities and tools for organizing, managing, and accessing the data in the database. The most important are its data definition language, data dictionary, and data manipulation language.

DBMS have a **data definition** capability to specify the structure of the content of the database. It would be used to create database tables and to define the characteristics of the fields in each table. This information about the database would be documented in a data dictionary. A **data dictionary** is an automated or manual file that stores definitions of data elements and their characteristics.

Microsoft Access has a rudimentary data dictionary capability that displays information about the name, description, size, type, format, and other properties of each field in a table (see Figure 6.6). Data dictionaries for large corporate databases may capture additional information, such as usage, ownership (who in the organization is responsible for maintaining the data), authorization, security, and the individuals, business functions, programs, and reports that use each data element.

Querying and Reporting

DBMS includes tools for accessing and manipulating information in databases. Most DBMS have a specialized language called a **data manipulation language** that is used to add, change, delete, and retrieve the data in the database. This language contains commands that permit end users and programming specialists to extract data from the database to satisfy information requests and develop applications. The most prominent data manipulation language today is **Structured Query Language**, or **SQL**. Figure 6.7 illustrates the SQL query that would produce the new resultant table in Figure 6.5. You can find out more about how to perform SQL queries in our Learning Tracks for this chapter.

Users of DBMS for large and midrange computers, such as DB2, Oracle, or SQL Server, would employ SQL to retrieve information they needed from the database. Microsoft Access also uses SQL, but it provides its own set of user- friendly tools for querying databases and for organizing data from databases into more polished reports.

In Microsoft Access, you will find features that enable users to create queries by identifying the tables and fields they want and the results, and then select- ing the rows from the database that meet particular criteria. These actions in turn are translated into SQL commands. Figure 6.8 illustrates how the same query as the SQL query to select parts and suppliers would be constructed using the Microsoft query-building tools.

Microsoft Access and other DBMS include capabilities for report generation so that the data of interest can be displayed in a more structured and polished format than would be possible just by querying. Crystal Reports is a popular report generator for large corporate DBMS, although it can also be used with Access. Access also has capabilities for developing desktop system applications. These include tools for creating data entry screens, reports, and developing the logic for processing transactions.

DESIGNING DATABASES

To create a database, you must understand the relationships among the data, the type of data that will be maintained in the database, how the data will be used, and how the organization will need to change to manage data from a company-wide perspective. The database requires both a conceptual design and a physical design. The conceptual, or logical, design of a database is an abstract model of the database from a business perspective, whereas the physical design shows how the database is actually arranged on direct-access storage devices.

entity-relationship diagramming, and database design in the Learning Tracks for this chapter.

Relational database systems try to enforce **referential integrity** rules to ensure that relationships between coupled tables remain consistent. When one table has a foreign key that points to another table, you may not add a record to the table with the foreign key unless there is a corresponding record in the linked table. In the database we examined earlier in this chapter, the foreign key Supplier_Number links the PART table to the SUPPLIER table. We may not add a new record to the PART table for a part with Supplier_ Number 8266 unless there is a corresponding record in the SUPPLIER table for Supplier_Number 8266. We must also delete the corresponding record in the PART table if we delete the record in the SUPPLIER table for Supplier_ Number 8266. In other words, we shouldn't have parts from nonexistent suppliers!

Database designers document their data model with an **entity-relationship diagram**, illustrated in Figure 6.11. This diagram illustrates the relationship between the entities SUPPLIER, PART, LINE_ITEM, and ORDER. The boxes represent entities. The lines connecting the boxes represent relationships. A line connecting two entities that ends in two short marks designates a one-to- one relationship. A line connecting two entities that ends with a crow's foot topped by a short mark indicates a one-to-many relationship. Figure 6.11 shows that one ORDER can contain many LINE_ITEMs. (A PART can be ordered many times and appear many times as a line item in a single order.) Each PART can have only one SUPPLIER, but many PARTs can be provided by the same SUPPLIER.

It can't be emphasized enough: If the business doesn't get its data model right, the system won't be able to serve the business well. The company's systems will not be as effective as they could be because they'll have to work with data that may be inaccurate, incomplete, or difficult to retrieve. Understanding the organization's data and how they should be represented in a database is perhaps the most important lesson you can learn from this course.

For example, Famous Footwear, a shoe store chain with more than 800 locations in 49 states, could not achieve its goal of having "the right style of shoe in the right store for sale at the right price" because its database was not properly designed for rapidly adjusting store inventory. The company had an Oracle relational database running on a midrange computer, but the database was designed primarily for producing standard reports for management rather than for reacting to marketplace changes. Management could not obtain precise data on specific items in inventory in each of its stores. The company had to work around this problem by building a new database where the sales and inventory data could be better organized for analysis and inventory management.

3. Latihan dan Jawaban

1) What are the problems of managing data resources in a traditional file environment and how are they solved by a database management system?

   Traditional file management techniques make it difficult for organizations to keep track of all of the pieces of data they use in a systematic way and to organize these data so that they can be easily accessed. Different functional areas and groups were allowed to develop their own files indepen- dently. Over time, this traditional file management environment creates problems such as data redundancy and inconsistency, program-data dependence, inflexibility, poor security, and lack of data sharing and availability. A database management system (DBMS) solves these problems with software that permits centralization of data and data management so that businesses have a single consistent source for all their data needs. Using a DBMS minimizes redundant and inconsistent files.

2) What are the major capabilities of DBMS and why is a relational DBMS so powerful?

   The principal capabilities of a DBMS includes a data definition capability, a data dictionary capabil- ity, and a data manipulation language. The data definition capability specifies the structure and content of the database. The data dictionary is an automated or manual file that stores information about the data in the database, including names, definitions, formats, and descriptions of data elements. The data manipulation language, such as SQL, is a specialized language for accessing and manipulat- ing the data in the database.

The relational database has been the primary method for organizing and maintaining data in information systems because it is so flexible and accessible. It organizes data in two-dimensional tables called relations with rows and columns. Each table contains data about an entity and its attributes. Each row represents a record and each column represents an attribute or field. Each table also contains a key field to uniquely identify each record for retrieval or manipulation. Relational database tables can be combined easily to deliver data required by users, provided that any two tables share a common data element. Non-relational databases are becoming popular for managing types of data that can't be handled easily by the relational data model. Both relational and non-relational database products are available as cloud computing services.

4. Daftar Pustaka

1. Management Information Systems, Managing Digital Firm, 11th Ed, Kenneth C. Laudon, Jane. P. Laudon. (L&L)
2. Management Information Systems With Misource 2007, 8th Ed James A. O'brien, And George Marakas
3. Managing Information Technology 5th Edition Martin, Brown, Dehayes