BAB II
TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer *Computer Based Information System* (CBIS). Dalam praktik istilah sistem informasi lebih sering dipakai tanpa menggunakan kata komputer walaupun dalam kenyataannya komputer merupakan bagian yang penting. Dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja). Ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu tujuan. Sistem informasi terdiri dari beberapa komponen yaitu: [1]

a. Komponen *Input*

   Input mewakili data yang masuk ke dalam sistem informasi dan *input* ini termasuk metode dan media untuk menangkap data yang akan dimasukkan yang berupa dokumen-dokumen dasar.

b. Komponen Model

   Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di *database* dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

c. Komponen *Output*

   Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

d. Komponen Teknologi

   Teknologi merupakan "*toolbox"* dalam sistem informasi, teknologi digunakan untuk yang menerima *input*, menjalankan model, menyimpan dan mengakses *data*, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

e. Komponen *Hardware*

Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi yang berfungsi sebagai tempat untuk menampung *database* atau lebih mudah dikatakan sebagai sumber *data* dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

f. Komponen *Software*

Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi *data* yang diambil dari *hardware* untuk menciptakan suatu informasi.

g. Komponen Basis *Data*

Database (Basis *Data*) merupakan kumpulan *data* yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. *Data* perlu disimpan dalam basis *data* untuk keperluan penyediaan informasi lebih lanjut.

h. Komponen *Control*

Banyak hal yang dapat merusak sistem informasi seperti bencana alam, api, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ke tidak efisien dan lainnya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjut terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.2 *Prototype*

Prototype adalah satu versi dari sebuah sistem potensial yang memberikan ide dari para pengembang dan calon pengguna, bagaimana sistem akan berfungsi dalam bentuk yang telah selesai. [2]

Proses pembuatan *prototype* ini disebut *prototyping*. Dasar pemikirannya adalah membuat *prototype* secepat mungkin, bahkan dalam waktu semalam, lalu memperoleh umpan balik dari pengguna yang akan memungkinkan *prototype* tersebut diperbaiki kembali dengan sangat cepat. Terdapat dua jenis *prototype* evolusioner dan persyaratan yaitu: [2]

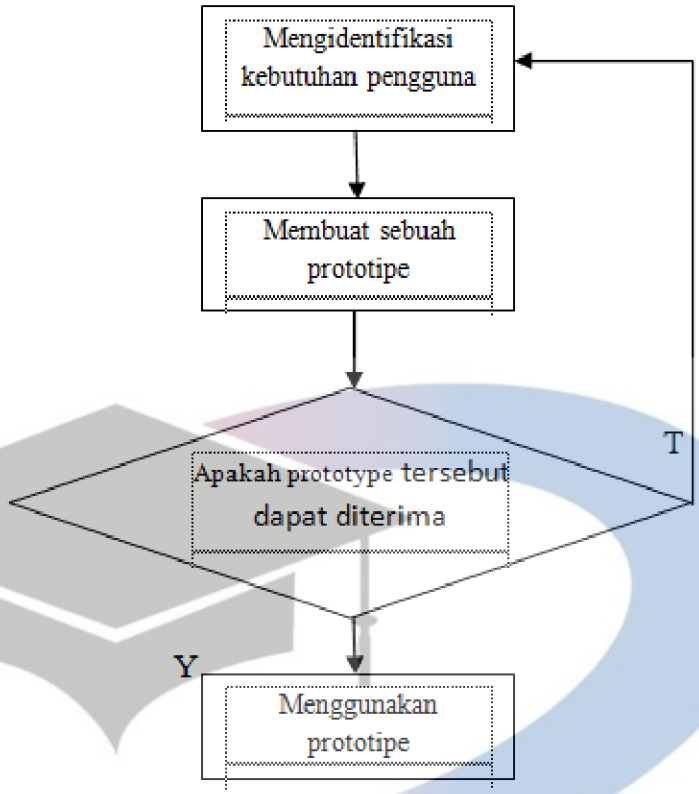
a. *Prototype* evolusioner (*evolutionary prototype*) terus-menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem

yang baru. *Prototype* ini kemudian dilanjutkan produksi. Jadi satu *prototype* evolusioner akan menjadi sistem aktual.

b. *Prototype* persyaratan (*requirement prototype*) dikembangkan sebagai satu cara untuk mendefinisikan persyaratan-persyaratan fungsional dari sistem baru ketika pengguna tidak mampu mengungkapkan apa yang diinginkan. Dengan meninjau *prototype* persyaratan seiring dengan ditambahkannya *fitur-fitur*, pengguna akan mampu mendefinisikan pemrosesan yang dibutuhkan dari sistem yang baru. Ketika persyaratan ditentukan, *prototype* persyaratan telah mencapai tujuan dan proyek lain akan dimulai untuk pengembangan sistem baru.

Empat langkah dalam pembuatan suatu *prototype* evolusioner, yaitu: [2]

a. Mengidentifikasi kebutuhan pengguna. Pengembang mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.
b. Membuat satu *prototype*. Pengembang mempergunakan satu alat *prototyping* atau lebih untuk membuat *prototype*.
c. Menentukan apakah *prototype* dapat diterima, pengembang mendemonstrasikan *prototype* kepada para pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan, jika sudah, langkah keempat akan diambil; jika tidak, *prototype* direvisi dengan mengulang langkah satu, dua dan tiga dengan pemahaman yang lebih baik mengenai kebutuhan pengguna.
d. Menggunakan *prototype*, *prototype* menjadi sistem produksi.



Gambar 2.1 Alur Pembuatan *Prototype* [2]

2.3 Rumah Sakit

Rumah sakit adalah institusi yang fungsi utamanya memberikan pelayanan kepada pasien, diagnostik dan terapeutik untuk berbagai penyakit dan masalah kesehatan, baik yang bersifat bedah maupun non bedah. Rumah sakit harus dibangun dan dilengkapi, serta dipelihara dengan baik untuk menjamin pelayanan kesehatan, keselamatan pasiennya, harus menyediakan fasilitas yang lapang, tidak berdesak desakan, dan terjamin sanitasinya untuk kesembuhan pasien. [3]

Berdasarkan Undang-undang Republik Indonesia No. 44 tahun 2009 tentang rumah sakit, Rumah Sakit adalah institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna yang menyediakan pelayanan rawat inap, rawat jalan, dan gawat darurat. Rumah Sakit Umum mempunyai misi memberikan pelayanan kesehatan yang bermutu dan terjangkau oleh masyarakat dalam rangka meningkatkan derajat kesehatan masyarakat. Berdasarkan Undang-undang Republik Indonesia No. 44 Tahun 2009

tentang Rumah Sakit, tugas Rumah Sakit yaitu memberikan pelayanan kesehatan perorangan secara paripurna. [4]

2.4 Rawat Inap

Rawat inap merupakan unit pelayanan non struktural yang menyediakan fasilitas dan menyelenggarakan kegiatan pelayanan kesehatan perorangan yang meliputi observasi, diagnosa, pengobatan, keperawatan dan rehabilitasi medik. Rawat inap adalah pemeliharaan kesehatan rumah sakit di mana penderita tinggal mondok sedikitnya satu hari berdasarkan rujukan dari pelaksanaan pelayanan kesehatan atau rumah sakit pelaksanaan pelayanan kesehatan lain. [5]

2.5 Standar Pelayanan Minimal Rawat Inap

Berdasarkan Keputusan menteri kesehatan nomor 129 Tahun 2008 Standar Pelayanan Minimal (SPM) adalah ketentuan tentang jenis dan mutu pelayanan dasar yang merupakan urusan wajib daerah yang berhak diperoleh setiap warga secara minimal. SPM juga merupakan spesifikasi teknis tentang tolak ukur pelayanan minimum yang diberikan oleh Badan Layanan Umum. SPM untuk jenis layanan rawat inap berdasarkan ketentuan Depkes adalah sebagai berikut: [6]

Tabel 2.1 Standar Pelayanan Minimal Menurut Departemen Kesehatan [6]

Pelayanan	Indikator	Standar
Rawat Inap	a. Pemberian pelayanan di rawat inap	a. Dr. Spesialis, perawat minimal pendidikan D3
	b. Dokter penanggung jawab pasien (DPJP) rawat inap	b. 100%
	c. Ketersediaan pelayanan rawat inap	c. Anak, penyakit dalam, kebidanan, bedah
	d. Jam *visite* Dr. Spesialis	d. 08.00 s/d 14.00 *wib* setiap hari kerja
		e. ≤ 1,5 %

	e. Kejadian infeksi pasca operasi f. Kejadian infeksi nosokomial g. Tidak adanya kejadian pasien jatuh yang berakibat kecacatan atau kematian h. Kematian pasien > 48 jam i. Kejadian pulang paksa j. Kepuasan pelanggan	f. ≤ 1,5 % g. 100 % h. ≤ 0.24 % i. ≤ 5 % j. ≥ 90 %

2.6 Use *Case Diagram*

Use *case Diagram* dipakai untuk menggambarkan relasi antara sistem dan sistem eksternal dan *user*, dengan kasus yang disesuaikan dengan langkah-langkah yang telah ditentukan. Use *case Diagram* merupakan cara atau metode yang cocok digunakan untuk dapat menggambarkan interaksi yang jelas antara sistem dengan pengguna. Notasi-notasi yang terdapat pada Use *case diagram* yaitu: [8]

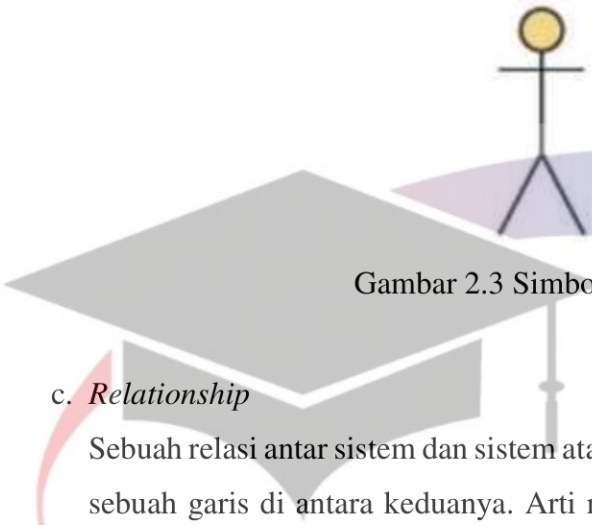
a. Use *Case*

Use *case* mendeskripsikan fungsi dari sebuah sistem dilihat dari sudut pandang pengguna.

Gambar 2.2 Simbol Use *Case* [8]

b. *Actor*

Actor merupakan sesuatu yang berinteraksi dengan sistem untuk saling betukar informasi. *Actor* tidak garu berupa manusia, tetapi dapat berupa suatu organisasi atau sistem informasi.



Gambar 2.3 Simbol *Actor* [8]

c. *Relationship*

Sebuah relasi antar sistem dan sistem atau *user* dan sistem digambarkan dengan sebuah garis di antara keduanya. Arti relasi yang digambarkan bisa beragam tergantung pada bagaimana garis itu digambarkan dan apa yang mereka hubungkan. Ada beberapa macam relasi, antara lain:

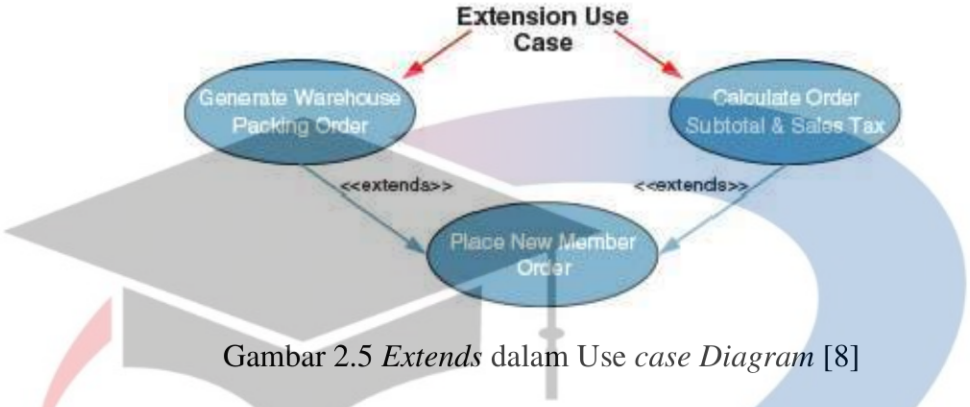
a. Associations

Associations adalah sebuah relasi antara seorang *actor* dengan sebuah Use *case* di mana terjadi interaksi antar mereka. Asosiasi dengan panah tertutup (1) di ujung yang menyentuh Use *case* mengindikasikan bahwa *actor* di ujung yang satu lagi melakukan Use *case* tersebut. Sedangkan asosiasi tanpa panah (2) mengindikasikan sebuah interaksi dari Use *case* ke *actor* yang menerima hasil dari Use *case* tersebut.

Gambar 2.4 *Associations* dalam Use *case Diagram* [8]

b. *Extends*

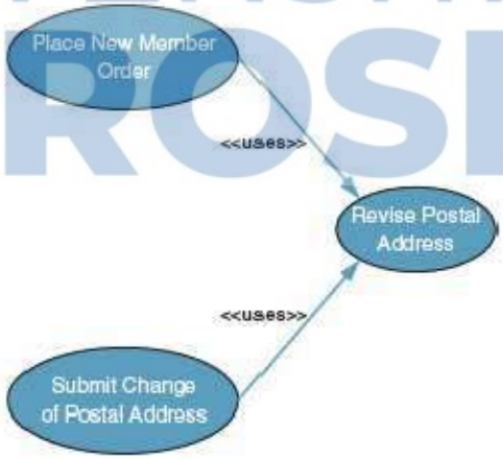
Extends perluasan dari Use *case* lain jika kondisi atau syarat terpenuhi. Kurangi penggunaan *Association Extend* ini, terlalu banyak pemakaian *association* ini membuat diagram sulit dipahami. Tanda panah terbuka harus terarah ke *parent* atau *base* use *case*.



Gambar 2.5 *Extends* dalam Use *case Diagram* [8]

c. *Uses (or Include)*

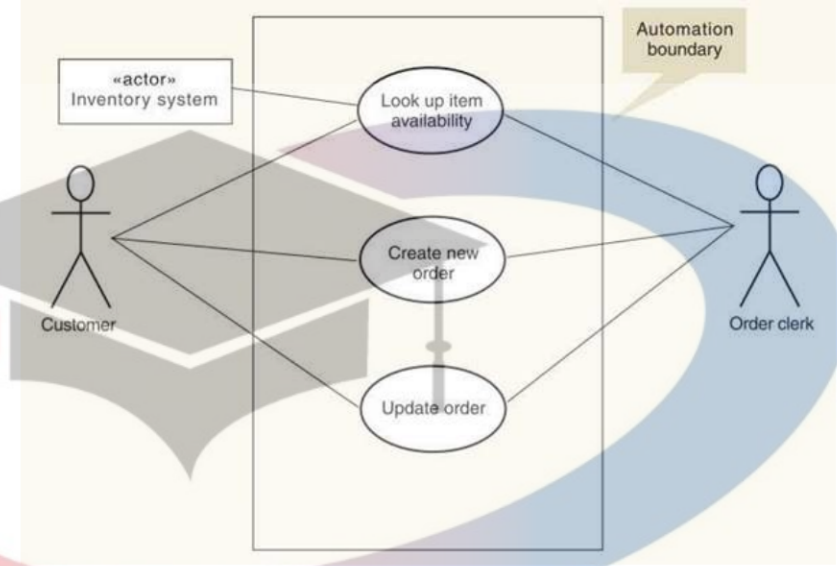
Uses (or Include) termasuk di dalam Use *case* lain (*required*) atau (diharuskan). *Uses* (*or Include)* yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, di mana pada kondisi ini sebuah Use *case* adalah bagian dari Use *case* lainnya.



Gambar 2.6 *Uses* dalam Use *case Diagram* [8]

d. *Automation Boundary*

Automation Boundary adalah garis batas yang mengelilingi seluruh rangkaian Use *case*. *Boundary* ini menunjukkan batas antara lingkungan di mana letak *actor* dan komponen dari sistem. [7]



Gambar 2.7 *Automation Boundary* dalam Use *case Diagram* [7]

2.7 Use *Case Descriptions*

Informasi yang terperinci mengenai proses dari setiap Use *case* dideskripsikan menggunakan *Use case description.* Komponen di dalam *use case description*: [7]

a. Use *Case Name*

Use case name merupakan nama dari Use *case* yang akan dideskripsikan.

b. *Scenario*

Scenario merupakan sebuah kumpulan kegiatan internal yang unik dalam suatu Use *case*.

c. *Triggering Event*

Triggering event merupakan deskripsi mengenai peristiwa yang menyebabkan terjadinya Use *case* tersebut.

d. *Brief Description*

Brief description merupakan deskripsi singkat mengenai Use *case*.

e. *Actors*

Actors mengidentifikasi pelaku dari Use *case.*

f. *Related* Use *Case*

Related Use *case* menunjukkan Use *case* yang berhubungan dengan Use *case* yang sedang dideskripsikan dan bagaimana Use *case* tersebut saling berhubungan.

g. *Stakeholders*

Stakeholders menggambarkan pihak-pihak lain yang berkepentingan dengan Use *case* selain *actors. Stakeholders* tidak selalu yang berhubungan dengan Use *case* tetapi mungkin mereka memiliki kepentingan dengan hasil dari Use *case.*

h. *Preconditions*

Preconditions merupakan syarat yang harus terpenuhi sebelum Use *case* dijalankan.

i. *Postconditions*

Postconditions mengidentifikasi kondisi yang benar setelah menjalankan Use *case.*

j. *Flow of Activities*

Flow of activities menggambarkan secara rinci aliran kegiatan dari Use *case.* Terdapat dua kolom pada komponen ini, yang pertama yaitu kolom *actors* yang mendeskripsikan langkah-langkah yang dilakukan *actor* dan yang kedua kolom System yang merupakan *respon* yang diberikan oleh sistem.

k. *Exception Conditions*

Exception conditions merupakan kondisi-kondisi pengecualian pada Use *case*.

Use case name:	*Create customer account.*	
Scenario:	Create online customer account.	
Triggering event:	New customer wants to set up account online.	
Brief description:	Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card.	
Actors:	Customer.	
Related use cases:	Might be invoked by the *Check out shopping cart* use case.	
Stakeholders:	Accounting, Marketing, Sales.	
Preconditions:	Customer account subsystem must be available. Credit/debit authorization services must be available.	
Postconditions:	Customer must be created and saved. One or more Addresses must be created and saved. Credit/debit card information must be validated. Account must be created and saved. Address and Account must be associated with Customer.	
Flow of activities:	**Actor**	**System**
	1. Customer indicates desire to create customer account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.
	2. Customer enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.
	3. Customer enters credit/debit card information.	3.1 System creates account. 3.2 System verifies authorization for credit/debit card. 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details.
Exception conditions:	1.1 Basic customer data are incomplete. 2.1 The address isn't valid. 3.2 Credit/debit information isn't valid.	

Gambar 2.8 Contoh *Use Case Descriptions* [7]

2.8 *Database Management System (DBMS)*

Databases Management System (DBMS) adalah sebuah perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, mengontrol akses ke basis *data*. DBMS menyediakan beberapa fasilitas yang bisa digunakan, yaitu: [8]

a. *Data Definition Language* (DDL)

DDL Memungkinkan pengguna untuk menentukan tipe *data*, struktur *data*, dan batasan aturan dari *data* yang disimpan dalam *database*.

b. *Data Manipulation Language* (DML)

Melalui DML ini, pengguna diperbolehkan untuk menambah *data*, mengubah *data*, menghapus dan mengambil *data* kembali dari *database*. Terdapat juga sebuah fasilitas yang melayani pengaksesan *data* yang disebut *Query Language*. *Structured Query Language* (SQL) merupakan bahasa yang diakui merupakan standar bagi DBMS.

c. Menyediakan akses kontrol ke *database*, seperti:

a. Sistem keamanan, mencegah pengguna yang tidak memiliki hak untuk mengakses *data*.

b. Sistem integritas, memelihara konsistensi *data* yang tersimpan dalam *database*.

c. Sistem kontrol konkurensi, memungkinkan akses bersamaan dalam *database*.

d. Sistem kontrol pemulihan, mengembalikan *database* keadaan konsisten sebelumnya setelah mengalami kegagalan perangkat keras atau perangkat lunak.

e. Katalog yang bisa diakses oleh pengguna yang berisi deskripsi dari *database*.

DBMS menyediakan beberapa fungsi dan layanan di bawah ini: [8]

a. *Data storage*, *retrieval*, *and update*

Sebuah DBMS harus memiliki kemampuan untuk menyimpan, mengambil dan melakukan perubahan *data* dalam *database*.

b. A *user-accessible catalog*

DBMS harus memberikan katalog yang berisi deskripsi dari *data* yang disimpan dan bisa diakses oleh pengguna.

c. *Transacton support*

DBMS harus memberikan mekanisme yang memastikan bahwa semua *update* yang berhubungan dengan transaksi yang diberikan adalah transaksi yang cocok dengan transaksi yang sebenarnya.

d. *Conccurency control service*

DBMS dapat memastikan bahwa *database* diubah dengan benar ketika beberapa pengguna mengubah *database* secara bersamaan.

e. *Recovery service*

Sebuah DBMS harus memberikan mekanisme untuk memulihkan *database* yang rusak dengan cara apa pun.

f. *Authorization service*

DMBM harus dapat memastikan bahwa hanya pengguna yang sah yang dapat mengakses *database*.

g. *Support for data communication*

DBMS mampu berintegrasi dengan *software* komunikasi.

h. *Integrity service*

DBMS memastikan *data* dan perubahan sesuai dengan aturan-aturan tertentu

i. *Service to promote data independece*

DBMS harus memiliki fasilitas untuk mendukung program yang tidak bergantung pada struktur *database* yang sebenarnya.

j. *Utility service*

DBMS harus menyediakan layanan utilitas, contohnya fasilitas untuk mengimpor, fasilitas untuk melakukan pengawasan, dan program analisa statistik.

2.9 *MySQL*

MySQL tergolong sebagai DBMS (*DataBase Management System*). Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat. Berikut adalah sejumlah aktivitas yang terkait dengan data yang didukung oleh perangkat lunak tersebut: [9]

a. Menyimpan data ke dalam tabel.
b. Menghapus data dalam tabel.
c. Mengubah data dalam tabel.
d. Mengambil data yang tersimpan dalam tabel.
e. Memungkinkan untuk memilih data tertentu yang diambil.
f. Memungkinkan untuk melakukan pengaturan hak akses terhadap data.