

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Sistem Infomasi**

##### **2.1.1 Pengertian Dasar Sistem**

Secara umum yang dimaksud sistem adalah sekumpulan unsur / elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan. Definisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan.

Menurut L. Ackof sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya. Ludwig Von Bartalanfy mendefinisikan sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan. Sedangkan menurut Scott system terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*proses*), serta keluaran (*output*).

##### **2.1.2 Pengertian Dasar Informasi**

Informasi adalah data yang dirubah menjadi bentuk yang lebih berguna dan berarti bagi pihak yang menerimanya. Sedangkan data adalah suatu kenyataan yang menggambarkan suatu kejadian-kejadian dari kesatuan nyata.

Menurut Barry E. Cushing, informasi didefinisikan sebagai hasil dari pengolahan data yang diorganisasikan dan berguna bagi orang atau pihak yang menerimanya. Informasi dikatakan berguna apabila mempunyai kualitas yang baik dalam membantu seorang menejer mengambil keputusan dan dapat menentukan kebijaksanaan guna mencapai tujuan organisasi.

##### **2.1.3 Sistem Informasi**

Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut. Selain itu data juga memegang peranan yang penting dalam sistem informasi. Data yang akan dimasukkan dalam sebuah sistem

informasi dapat berupa formulir-formulir, prosedur-prosedur dan bentuk data lainnya.

#### **2.1.4 Pengertian Kantor Urusan Agama (KUA)**

Keputusan Menteri Agama No. 517 Tahun 2001 tentang Penataan Organisasi Kantor Urusan Agama Kecamatan Bukit Intan, tugas KUA adalah melaksanakan sebagian tugas Kantor Kementerian Agama Kabupaten dan Kota dibidang Urusan Agama Islam dalam wilayah Kecamatan. Dalam melaksanakan tugasnya tersebut, maka KUA melaksanakan fungsi:

- 1) Menyelenggarakan pendataan dan dokumentasi,
- 2) Menyelenggarakan surat menyurat, kearsipan, pengetikan, dan rumah tangga KUA Kecamatan Bukit Intan.
- 3) Melaksanakan pencatatan nikah, rujuk, mengurus dan membina masjid, zakat, wakaf, baitul maal dan ibadah sosial, kependudukan dan pengembangan keluarga sakinah sesuai dengan kebijaksanaan yang ditetapkan oleh Dirjen Bimas Islam berdasarkan peraturan perundang-undangan yang berlaku.

#### **2.2 Pengertian Analisa Berorientasi Objek**

##### **a. Activity Diagram**

Diagram memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktivitas pada suatu proses. Diagram ini sangat mirip dengan *flow chart* karena kita dapat memodelkan prosedur logika, proses bisnis dan alur kerja. Perbedaan utamanya adalah *flow chart* dibuat untuk menggambarkan alur kerja dari sebuah sistem, sedangkan *activity diagram* dibuat untuk menggambarkan aktivitas dari aktor.

Menurut Munawar (2004:109) "*Activity diagram* adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus".

*Activity diagram* mempunyai pesan seperti halnya *flow chart*, akan tetapi perbedaannya dengan *flow chart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flow chart* tidak bisa.

Simbol – simbol yang sering di gunakan pada saat pembuatan *activity diagram* adalah sebagai berikut :

- 1) *Start Point (initial node)*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
- 2) *End Point (Activity Final Node)*, akhir aktivitas.
- 3) *Activities*, menggambarkan proses bisnis dan dikenal sebagai *activitystate*.

Jenis – jenis *activity*, yaitu :

- 1) *Black Hole Activities*, ada masukan dan tidak ada keluaran.
- 2) *Miracle Activities*, tidak ada masukan dan ada keluaran dan dipakai waktu *start point*.
- 3) *Parallel Activities*, *activity* yang berjalan secara bersamaan terdiri dari:
  - a) *Fork* (percabangan)  
*Fork* digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
  - b) *Join* (penggabungan)  
*Join* yaitu mempunyai 2 atau lebih transisi masuk dan hanya 1 transisi keluar, dan *fork* harus berhubungan dengan *join*.
  - c) *Transition*  
*Transition* menggambarkan aliran perpindahan kontrol antara *state*.
  - d) *Decission Point*  
 Digambarkan dengan lambang wajik/belah ketupat mempunyai transisi (sebuah garis dari/ke *decision point*). Setiap transisi yang ada harus mempunyai *guard* (kunci).
  - e) *Swimlane*  
 Sebuah cara untuk mengelompokkan *activity* berdasarkan *actor* (mengelompokkan *activity* dalam sebuah urutan yang sama). *Actor* bisa ditulis nama *actor* ataupun sekaligus dalam lambang *actor (stick figure)* pada *use case diagram*. *Swimlane* digambar secara *vertical*, walaupun kadang - kadang digambar secara *horizontal*.
  - f) *Swimarea*  
 Ketika sebuah *activity diagram* mempunyai banyak *swimlane*, perlu dipikirkan dengan pendekatan *swimarea*. *Swimarea* mengelompokkan

*activity* berdasarkan kegiatan didalam *use case*.

**b. Analisa Dokumen Keluaran**

Menurut Sutopo (2002:12) “Analisa keluaran adalah analisa mengenai dokumen – dokumen keluaran yang dihasilkan dari sebuah sistem”.

**c. Analisa Dokumen Masukan**

Menurut Sutopo (2002:12) “Analisa masukan adalah bagian dari pengumpulan informasi tentang sistem yang sedang berjalan”. Tujuan analisa masukan adalah memahami prosedur berjalan.

**d. Use Case Diagram**

*Use Case Diagram* menggambarkan sebuah fungsionalitas yang diharapkan dari sebuah sistem dan bagaimana sistem berinteraksi dengan dunia luar. Yang ditekankan dalam *Use Case Diagram* adalah “apa” yang diperbuat sistem, dan bukan “bagaimana” sistem itu melakukannya. Sebuah *Use Case* merepresentasikan sebuah interaksi antara *actor* dengan sistem. *Use Case Diagram* juga menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*). *Use Case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, *create* sebuah daftar belanja, dan sebagainya.

Secara umum *Use Case Diagram* terdiri dari :

1) *Actor*

Menurut Munawar (2004:64) “*Actor* adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem”. Untuk mengidentifikasi *actor* harus ditentukan pembagian kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. *Actor* dilukiskan dengan peran yang mereka mainkan dalam *use case*, seperti staff penjualan, pelanggan, dll.

2) *Use Case*

Menurut Munawar (2004:62) “*Use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna”. *Use case* dibuat berdasarkan keperluan *actor*, merupakan “apa” yang dikerjakan sistem bukan “bagaimana” sistem

mengerjakannya. Setiap *use case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan *actor*. Nama *use case* boleh terdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama.

### 3) *Relationship* (Relasi) / *Association* (Asosiasi)

Menurut Jeffery L. Whitten (2004:274) "Asosiasi adalah sebuah relasi antara *actor* dengan *use case* dimana sebuah interaksi terjadi diantara mereka."

Relasi (*relationship*) digambarkan sebagai bentuk garis antara dua simbol dalam *use case diagram*.

Ada empat jenis relasi / asosiasi yang dapat timbul pada *use case diagram*, yaitu :

#### a) Asosiasi antara *Actor* dan *Use Case*

Ujung panah pada *association* antara *actor* dan *use case* mengindikasikan siapa / apa yang meminta interaksi dan bukannya mengindikasikan aliran data. Sebaiknya gunakan garis tanpa panah untuk *association* antara *actor* dan *use case*. *Association* antar *actor* dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila *actor* berinteraksi secara pasif dengan sistem.

#### b) Asosiasi antara *Use Case*

Relasi antara *use case* dengan *use case* :

(1) *Include*, menggambarkan suatu *use case* termasuk di dalam *use case* lain (diharuskan). Contohnya adalah pemanggilan sebuah fungsi program. Digambarkan dengan garis lurus berpanah dengan tulisan <<include>>.

(2) *Extend*, digunakan ketika hendak menggambarkan variasi pada kondisi perilaku normal dan menggunakan lebih banyak *control form* dan mendeklarasikan ekstension pada *use case* utama atau dengan kata lain adalah perluasan dari *use case* lain jika syarat atau kondisi terpenuhi. Digambarkan dengan garis lurus berpanah dengan tulisan <<extend>>.

(3) *Generalization / Inheritance* antar *Use Case* dipakai ketika ada

sebuah perlakuan khusus (*single condition*) dan merupakan pola hubungan *base-parent use case*. Digambarkan dengan *generalization / inheritance* antar *use case* secara vertikal dengan *inheriting use case* dibawah *base / parent use case*.

(4) *Generalization / Inheritance* antar *Actors*, digambarkan *generalization* antar *actors* secara vertikal dengan *inheriting actor* dibawah *base / parent use case*.

**e. Deskripsi Use Case ( Use Case Description)**

Deskripsi *use case* adalah resume langkah atau tahapan dalam *usecase*.

Kegunaannya untuk mendeskripsikan secara rinci mengenai *usecase diagram*.

Deskripsi *usecase diagram* memiliki tiga komponen yang umum, yaitu :

- 1) *Use case*, berisi nama *use case*.
- 2) *Actor*, berisi nama *actor* yang menjalankan sistem.
- 3) Deskripsi, menjelaskan bagaimana sistem berjalan.

### 2.3 Perancangan Sistem Berorientasi Objek

Menurut Jeffery L. Whitten (2004:686) “Perancangan sistem berorientasi obyek (*Object-Oriented Design*) adalah suatu pendekatan yang digunakan untuk menspesifikasikan kebutuhan – kebutuhan sistem dengan mengkolaborasikan obyek – obyek, atribut –atribut dan metode – metode yang ada”.

Fokus dari desain objek adalah perencanaan struktur data dan algoritma yang diperlukan untuk implementasi setiap kelas.

Adapun diagram - diagram UML yang digunakan penulis dalam merancang sistem berorientasi objek adalah :

**a. Entity Relationship Diagram (ERD)**

Menurut Jeffery L. Whitten et al (2004:295) “Diagram-ER adalah sebuah model data yang menggunakan beberapa notasi untuk menggambarkan data dalam hal entitas dan relasi yang digambarkan oleh data tersebut”.

Yang pertama kali mendeskripsikan diagram-ER adalah Peter Chan yang dibuat sebagai bagian dari perangkat lunak case. Model ERD adalah suatu penyajian data

dengan menggunakan entity dan relationship. Diagram-ER menggambarkan hubungan antara data yang ada dan tidak menggambarkan proses – proses yang terjadi.

Simbol – simbol / notasi yang biasa digunakan dalam ERD, antara lain :

- 1) Menurut Jeffery L. Whitten et al (2004:295) "*Entity* (entitas) adalah sebuah kelas dari orang, tempat obyek, kejadian dan sebagainya atau sebuah konsep yang mana kita perlukan untuk menagkap dan menyimpan data".

Pada entity terdapat dua jenis, yaitu :

- a) *Strong Entity* adalah *entity* yang memiliki *primary key*
- b) *Weak Entity* adalah suatu *entity* yang tidak memiliki *primary key* dan keberadaan *entity* tersebut tergantung dari keberadaan *entity* lain. *Entity* yang merupakan induknya disebut *identifying owner* dan relasinya disebut *identifying relationship*.

- 2) *Relationship* ( Hubungan / relasi ), menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda.

- 3) *Cardinality* / Kardinalitas

*Cardinality* adalah tingkat hubungan atau derajat relasi. Tingkat *cardinality* yang terjadi pada sebuah ERD adalah sebagai berikut :

- a) *One To One* ( 1 : 1 )

Hubungan relasi *one to one* yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

- b) *One To Many* ( 1 : M )

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

- c) *Many To One* ( M : 1 )

Setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A, tetapi setiap entitas pada entitas A dapat berhubungan dengan satu entitas pada himpunan entitas B.

- d) *Many To Many* ( M : M )

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak

entitas pada himpunan entitas B, begitu juga sebaliknya.

- 4) Menurut Jeffery L. Whitten et al (2004:298) "Atribut adalah suatu deskripsi karakteristik dari entitas". Nilai atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut dalam suatu *entity* atau *relationship*.

Terdapat dua jenis atribut, yaitu :

- a) *Identifier (key)*, menentukan *entity* secara unik (beda antara satu sama lain dan tidak mungkin sama).
- b) *Descriptor (non key atribut)*, menentukan *entity* yang tidak unik. Atribut *relationship* sangat ditentukan oleh *cardinality*, yaitu :
  - (1) Jika *cardinality* 1 : 1 dan 1 : M, atribut *relationship* diambil dari identifier dari entitas di kiri dan kanan.
  - (2) Jika *cardinality* M : N, atribut *relationship* diambil dari identifier dari entitas di kiri dan kanan ditambah dengan atribut lain yang bukan milik entitas di kiri ataupun di kanan.

Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

#### b. Logical Record Structure (LRS)

Diagram-ER (ERD) harus dikonversi ke bentuk *structure* (struktur record secara logik). Sebuah model sistem yang digambarkan dengan sebuah Diagram-ER akan mengikuti pola/aturan pemodelan tertentu. Dalam kaitannya dengan konversi ke LRS, maka perubahan yang terjadi adalah mengikuti aturan-aturan berikut ini :

- 1) Setiap entitas akan diubah ke bentuk kotak
- 2) Sebuah *relationship* kadang disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada Diagram-ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi), sebuah *relationship* dipisah dalam sebuah kotak tersendiri jika tingkat hubungannya M:N (*Many to many*).

### c. Tabel / Relasi

Tabel adalah koleksi objek yang terdiri dari sekumpulan elemen yang diorganisasi secara kontigu, artinya memori yang dialokasi antara satu elemen dengan elemen yang lainnya mempunyai adress yang berurutan.

Untuk transformasi LRS ke table / relasi, berpedoman pada hal - hal sebagai berikut :

- 1) Tiap satu LRS akan menjadi satu tabel
- 2) Nama LRS menjadi satu tabel
- 3) Tiap 1 ( satu ) atribut akan menjadi 1 ( satu ) kolom
- 4) Nama atribut akan menjadi nama kolom.

### d. Spesifikasi Basis Data

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lain dan tersimpan di luar komputer serta digunakan perangkat lunak ( *software* ) tertentu untuk memanipulasinya.

Sedangkan sistem berbasis data adalah suatu sistem penyusunan dan pengelolaan *record - record* dengan menggunakan komputer dengan tujuan untuk menyimpan atau merekam serta melihat data operasional lengkap pada sebuah organisasi, sehingga mampu menyediakan informasi yang diperlukan untuk kepentingan proses pengambilan keputusan.

## 2.4 Teori Pendukung

Perkahwinan atau nikah menurut bahasa ialah berkumpul dan bercampur. Menurut istilah syarak pula ialah ijab dan qabul („aqad) yang menghalalkan persetubuhan antara lelaki dan perempuan yang diucapkan oleh kata-kata yang menunjukkan nikah, menurut peraturan yang ditentukan oleh Islam. Perkataan *zawaj* digunakan di dalam al-Quran bermaksud pasangan dalam penggunaannya perkataan ini bermaksud perkahwinan Allah s.w.t. menjadikan manusia itu berpasang-pasangan, menghalalkan perkahwinan dan mengharamkan zina.

Adapun nikah menurut syari“at nikah juga berarti akad. Sedangkan pengertian hubungan badan itu hanya metafora saja. Islam adalah agama yang syumul (universal). Agama yang mencakup semua sisi kehidupan. Tidak ada suatu masalah pun, dalam

kehidupan ini, yang tidak dijelaskan. Dan tidak ada satu pun masalah yang tidak disentuh nilai Islam, walau masalah tersebut nampak kecil dan sepele. Itulah Islam, agama yang memberi rahmat bagi sekalian alam. Dalam masalah perkawinan, Islam telah berbicara banyak. Dari mulai bagaimana mencari kriteria calon calon pendamping hidup, hingga bagaimana memperlakukannya kala resmi menjadi sang penyejuk hati. Islam menuntunnya. Begitu pula Islam mengajarkan bagaimana mewujudkan sebuah pesta pernikahan yang meriah, namun tetap mendapatkan berkah dan tidak melanggar tuntunan sunnah Rasulullah shallallahu „alaihi wa sallam, begitu pula dengan pernikahan yang sederhana namun tetap penuh dengan pesona. Melalui makalah yang singkat ini insyaallah kami akan membahas perkawinan menurut hukum islam.

Pernikahan adalah sunnah karuniah yang apabila dilaksanakan akan mendapat pahala tetapi apabila tidak dilakukan tidak mendapatkan dosa tetapi dimakruhkan karna tidak mengikuti sunnah rosul. Arti dari pernikahan disini adalah bersatunya dua insane dengan jenis berbeda yaitu laki-laki dan perempuan yang menjalin suatu ikatan dengan perjanjian atau akad.

Suatu pernikahan mempunyai tujuan yaitu ingin membangun keluarga yang sakinah mawaddah warohmah serta ingin mendapatkan keturunan yang solihah. Keturunan inilah yang selalu didambakan oleh setiap orang yang sudah menikah karena keturunan merupakan generasi bagi orang tuanya.