

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

Menurut Bocij (2008), Sistem dapat didefinisikan sebagai kumpulan komponen yang saling terkait yang bekerja sama untuk mencapai tujuan bersama. Fungsi sistem adalah untuk menerima masukan dan mengubah ini menjadi output

Seperti konsep data, ada beberapa definisi informasi yang umum digunakan, yaitu:

- a. Data yang telah diolah sehingga mereka bermakna
- b. Data yang telah diolah untuk tujuan
- c. Data yang telah dipahami dan dimengerti oleh penerima

Tiga hal penting dapat ditarik dari definisi ini pertama, ada proses yang jelas dan logis yang digunakan untuk menghasilkan informasi. Proses ini melibatkan pengumpulan data untuk sebuah proses transformasi dalam rangka menciptakan informasi. Kedua, informasi melibatkan dan menempatkan beberapa inisial data dalam bentuk konteks yang bermakna, sehingga mereka dapat dipahami dan ditindaklanjuti. Ketiga, informasi yang dihasilkan untuk suatu tujuan, untuk melayani kebutuhan informasi dari beberapa jenis

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Bocij, 2008).

Informasi dapat dikatakan memiliki sejumlah karakteristik yang berbeda dan dapat digunakan untuk menggambarkan kualitas. Informasi yang berkualitas menunjukkan bahwa informasi yang disajikan sesuai dengan harapan dan kebutuhan user berdasarkan dimensi kualitas informasi. Dimensi kualitas bisa disebut sebagai syarat sebuah informasi dikatakan berkualitas dilihat dari beberapa sudut. Dimensi tersebut adalah :

1. Dimensi Waktu

- a. *Timeliness* : Informasi harus ada saat dibutuhkan.
- b. *Currency* : Informasi harus selalu *up-to-date*.
- c. *Frequency* : Informasi disajikan berkali-kali sebanyak dibutuhkan.
- d. *Time Period* : Informasi dapat disajikan untuk periode sekarang-masa lalu – masa mendatang.

2. Dimensi Konten

- a. *Accuracy* : Informasi harus bebas dari kesalahan.
- b. *Relevance* : Informasi harus berhubungan dengan penggunanya pada situasi tertentu.
- c. *Completeness* : Informasi disajikan secara lengkap.
- d. *Conciseness* : Informasi disajikan hanya yang dibutuhkan.
- e. *Scope* : Informasi dapat disajikan untuk ruang lingkup luas maupun terbatas atau internal/eksternal.
- f. *Performance* : Informasi dapat menunjukkan nilai dengan pengukuran aktivitas yang telah diselesaikan.

3. Dimensi Bentuk

- a. *Clarity* : Informasi yang disajikan dalam bentuk yang mudah dimengerti.
- b. *Detail* : Informasi disajikan secara detail atau ringkasan.
- c. *Order* : Informasi dapat diatur dalam urutan tertentu.
- d. *Presentation* : Informasi dapat disajikan secara naratif, angka, grafik atau lainnya.
- e. *Media* : Informasi dapat disajikan dalam bentuk media cetak, video atau media lainnya.

2.2 Pertanahan

2.2.1 Pertanahan

Pertanahan berasal dari kata tanah. Dalam hukum tanah, kata sebutan “tanah” dipakai dalam arti yuridis, sebagai suatu pengertian yang telah diberi batasan sesuai oleh UUPA. Dalam Pasal 4 ayat (1) dinyatakan, bahwa :

“Atas dasar hak menguasai dari Negara ditentukan adanya macam-macam hak atas permukaan bumi, yang disebut tanah, yang dapat diberikan kepada dan dipunyai oleh orang-orang baik sendiri maupun bersama-sama dengan orang lain serta badan-badan hukum.”

Dengan demikian, tanah dalam pengertian yuridis adalah permukaan bumi, sedangkan hak atas tanah adalah hak atas sebagian tertentu permukaan bumi, yang terbatas, berdimensi dua dengan ukuran panjang dan lebar.

Pasal 4 ayat (2) menyatakan hak-hak atas tanah bukan hanya memberikan wewenang untuk mempergunakan sebagian tertentu permukaan bumi yang bersangkutan, yang disebut “tanah”, tetapi juga tubuh bumi yang ada di bawahnya dan air serta ruang yang ada di atasnya.²⁹ Dengan demikian, pengertian “tanah”

meliputi permukaan bumi yang ada di daratan dan permukaan bumi yang berada di bawah air, termasuk air laut.

2.2.2 Buku Letter C

Buku Letter C adalah sebagai alat bukti yang dimiliki oleh seseorang pada saat orang tersebut ingin memperoleh hak akan tanahnya, dan ingin melakukan pendaftaran tanah atas namanya. Dan tidak dapat dilupakan pula bahwa buku Letter C juga merupakan syarat yang harus ada untuk pengonversian tanah milik adat, sebagai bukti hak milik adat. (Soeprapto, 1986). Contoh Letter C Desa dapat dilihat pada Gambar 2.1

		SAWAH				TANAH KERING					
Nomor seri dan huruf bagian seri	Kelas desa	Menurut daftar perindian				Sebab dan tanggal perubahan	Nomor seri dan huruf bagian seri	Menurut daftar perindian			
		Luas milik		Ipe da				Luas milik		Ipe da	
		ha	da	R.	S.			ha	da	R.	S.
						√3	02	0 122	0 11		
						67	01	0 163	0 11		
						86	02	0 182	0 12		
						√3	02	0 280	0 14		
Nama: Pandri P. Aroen NO. 303											
		√6	√8	0 129	0 27	√5	02	0 024	0 03		
		71	√8	0 090	0 16	78	02	0 114	0 14		
		102	√4	0 122	0 07	92	02	0 179	0 12		
Nama: Pandong P. Venon NO. 304											
						90	02	0 057	0 10		

Gambar2.1 Contoh Letter C Desa

2.2.3 Petok D

Petok D merupakan surat tanda bukti kepemilikan tanah yang ada di tangan pemilik tanah dimana isinya adalah Kutipan Buku Letter C desa yang terdapat di desa/kelurahan. Jadi dapat disimpulkan buku Letter C aslinya itu di Kantor Desa/Kelurahan, sedangkan kutipannya berupa girik, petok D, kekitir diberikan pada pemilik tanah. Contoh Petok D dapat dilihat pada Gambar 2.2

DAFTAR MUTASI OBYEK DAN WAJIB PAJAK

KELURAHAN/DESA : RANDUPADANGAN
KECAMATAN : MENGANTI
DATI II : GRESIK

No.	NOMOR SPPT	NAMA DAN ALAMAT WAJIB PAJAK	DATA LAMA		LETAK OBYEK PAJAK B L O K	SEBAB TERJADI MUTASI	DATA BARU		KETERANGAN		
			TANAH	BANGUNAN			TANAH	BANGUNAN			
1.	0038	H. ABDUL MAJID	1.152	---	008	E I B A H	0038 C. desa No. 3969	K G Y I M A H	1.152	---	Asal persil 3 C. asal dari - Nomor : 905 S. II

RANDUPADANGAN, TANGGAL : 26 JUN 2001

Laporan ini Telah Ditanda-tangani pada tanggal 28 JUN 2001
KEPALA KANTOR PELAYANAN DESA GRESIK
SEKSI PENDATAAN DAN PENILAIAN

SUDIOKO, SH.
NIP. 260 242 648

KEPALA DESA
RANDUPADANGAN
M A M A D

Gambar 2.2 Contoh Petok D

2.2.4 Peta Blok

Peta Blok merupakan peta bidang tanah yang dipergunakan untuk keperluan permohonan hak, pemilik tanah, lokasi tanah dan mengetahui luas. Contoh Petok D dapat dilihat pada Gambar 2.3



Gambar 2.3 Contoh Peta Blok

2.3 Poligon

Poligon digunakan untuk merepresentasikan objek-objek dua dimensi. Suatu danau, batas propinsi, batas kota, batas – batas persil tanah milik adalah tipe-tipe entitas yang pada umumnya direpresentasikan sebagai poligon. Suatu poligon paling sedikit dibatasi oleh tiga garis yang saling terhubung diantara ketiga titik tersebut. (Nasruddin Unigha, 2016, from <http://hjtfruity.blogspot.co.id/2015/02/makalah-poligon.html#.Vs6wIdJ97IU>, 9 Februari 2015)

2.3.1 Macam-macam polygon

1. Atas dasar titik ikat:

a. Poligon terikat sempurna

Poligon yang ujung-ujungnya terikat pada dua titik yang diketahui koordinatnya,

b. Poligon terikat sepihak

Poligon yang salah satu titik ujungnya terikat atau diketahui koordinatnya,

c. Poligon bebas

Poligon yang ujung-ujungnya tidak terikat.

2. Atas dasar bentuk:

a. Poligon Terbuka

poligon yang ujungnya tidak saling bertemu satu dengan yang lain,

b. Poligon tertutup

poligon yang ujungnya saling bertemu (titik awal dan titik akhir menjadi satu) dan membentuk suatu loop atau kring,

c. Poligon cabang

Poligon yang merupakan cabang dari poligon yang lain.

3. Atas dasar hirarki dalam pemetaan :

a. Poligon yang utama

Poligon yang koordinat titik-titiknya diperoleh langsung dari penentuan koordinat titik local atau diikatkan langsung melalui pengukuran dari titik kontrol terdekat.

b. Poligon cabang

Poligon yang koordinat titik-titiknya diikatkan dari poligon utama.

2.4 Menggambar Poligon dengan PHP

PHP memiliki beberapa fungsi grafis yang mendukung untuk membuat shape (bangun ruang) seperti garis, kotak, lingkaran dan polygon. Cara Mudah

membuat garis, kotak, lingkaran dan polygon di PHP yaitu menggunakan fungsi-fungsi yang ada pada PHP.

Beberapa fungsi yang dapat digunakan untuk membuat shape pada PHP, yaitu;

- *imageline()*; untuk membuat garis
- *imagerectangle()*; untuk membuat kotak
- *imagearc()*; untuk membuat garis lentur seperti curva
- *imageellipse()*; untuk membuat lingkaran
- *imagepolygon()*; untuk membuat polygon

dari fungsi-fungsi tersebut kita dapat membuat garis, kotak, lingkaran dan polygon di PHP dengan mudah. (Finki Dona Marleny, 2016, from http://findomedia.blogspot.co.id/2016/01/cara-mudah-membuat-garis-kotak_3/, 25 Januari 2016)

2.5 – Software Engineering Body of Knowledge (SWEBOK)

Menurut England & Sons (2004) menuliskan : “SWEBOK menggambarkan pengetahuan secara umum tentang rekayasa perangkat lunak yang dibagi kedalam 10 area pengetahuan (*Knowledge Areas*) atau disebut Kas.” *Software Engineering Body of Knowledge (SWEBOK)* adalah produk dari komite koordinasi rekayasa perangkat lunak di sponsori oleh *IEEE Computer Society*. SWEBOK sendiri mempunyai panduan yang disebut *Guide to the SWEBOK*, panduan ini dibuat untuk lima tujuan yaitu:

- a. Untuk memperlihatkan kesamaan pandangan tentang rekayasa perangkat lunak di seluruh dunia.

- b. Untuk memperjelas tempat dan menetapkan batas dari rekayasa perangkat lunak dan hubungannya dengan disiplin ilmu lain seperti ilmu komputer, manajemen proyek, teknik komputer dan matematika.
- c. Untuk memberi karakter isi dari disiplin ilmu rekayasa perangkat lunak.
- d. Untuk memberikan akses topik ke SWEBOK
- e. Untuk memberikan pengetahuan dasar bagi pengembangan kurikulum dan sertifikasi serta perizinan.

Berikut adalah penjabaran tentang ruang lingkup pengetahuan atau yang disebut juga *Knowledge Area (KA)* yang digunakan sebagai panduan dalam mengembangkan aplikasi oleh penulis.

1. Requirement

Tahapan awal dalam membangun aplikasi, *Software Requirements* merupakan sebuah properti yang disajikan untuk memenuhi kebutuhan dalam menyelesaikan permasalahan yang ada akan diselesaikan oleh aplikasi tersebut. Menjabarkan bagaimana mengotomasikan sebuah permasalahan sebuah tugas yang dihadapi oleh pengguna, membantu menganalisis proses bisnis perusahaan yang telah menggunakan aplikasi, menganalisis kekurangan yang ada, dan lainnya. Berikut penjabaran tentang beberapa tahapan yang ada pada *software requirement*:

a. Requirement Elicitation

Tahapan awal dalam pemenuhan *software requirements* makna dari kebutuhan mendatang ini berhubungan dengan dari mana kebutuhan perangkat lunak itu sendiri dan bagaimana para pengembangan perangkat lunak dapat mengumpulkannya. Pada dasarnya, kegiatan yang dijabarkan adalah dari tiap

individu dan tiap pemegang kendali sistem tersebut untuk membangun kesinambungan antara pihak pengembang dan pengguna perangkat lunak itu sendiri.

b. Requirement Analysis

Tahapan ini membahas tentang kegiatan menganalisa kegiatan menganalisa kebutuhan untuk :

1. Mendeteksi dan menyelesaikan ketidakcocokan yang ada pada tiap-tiap kebutuhan.
2. Menggali batasan yang ada pada perangkat lunak yang dikembangkan dan bagaimana perangkat lunak tersebut akan berinteraksi dengan sistem.
3. Menguraikan kebutuhan sistem yang akan digunakan sebagai kebutuhan perangkat lunak

c. Requirements specification

Secara teknis pada kata “*specification*” mengacu pada banyaknya jumlah pekerjaan atau kemampuan perangkat lunak tersebut dalam mencapai tujuannya. Dalam sebuah istilah pengembangan perangkat lunak, “*software requirements specification*” secara khusus mengarah kepada hasil ketepatan, atau penyamaan elektronik, yang dapat ditinjau, dinilai, dan dibenarkan.

d. Requirement Verification and Validation

Beberapa dokumen requirements di atas menjadi bahan dari tahapan validasi dan verifikasi. Kebutuhan yang ada di validasi untuk menjamin bahwa pengembang dari perangkat lunak tersebut dapat memahami kebutuhan yang akan dicapai. Penyesuaian kebutuhan untuk standar perusahaan sangat penting untuk diperhatikan bahwa kebutuhan tersebut dimengerti, konsisten, dan lengkap.

2. Analisis

Tahap Analisis merupakan tahap identifikasi, seleksi, dan perencanaan sistem yang bertujuan untuk mendeteksi dan memberikan solusi antar kebutuhan serta mengetahui ruang lingkup perangkat lunak dan bagaimana perangkat lunak tersebut berinteraksi dengan lingkungan.

Tahapan analisis kebutuhan, menunjukkan tahapan-tahapan di dalam analisis kebutuhan. Pada dasarnya, aktivitas analisis dibutuhkan dalam setiap proses dalam daur hidup pengembangan perangkat lunak. Dalam proses rekayasa kebutuhan, analisis pun dilakukan dalam setiap aktivitas-aktivitasnya. Aktivitas tersebut antara lain sebagai berikut :

1. *Domain Understanding* : Dalam tahapan ini, pengembang harus mengetahui bagaimana organisasi perusahaan beroperasi dan apa yang menjadi permasalahan pada sistem yang berjalan.
2. *Requirements Collection* : Tahapan ini merupakan tahapan pengumpulan kebutuhan akan sistem yang akan dibangun sehingga diperlukan adanya interaksi secara intensif dengan *stakeholder*.
3. *Classification* : Tahapan ini mengelompokkan hasil dari tahap kebutuhan sehingga menjadi lebih terstruktur untuk selanjutnya diorganisir kedalam kelompok-kelompok yang koheren.
4. *Conflict Resolution* : Tahapan ini berguna untuk menemukan dan menyelesaikan kebutuhan yang di dalamnya terdapat konflik. Konflik tersebut dapat terjadi antara dua *stakeholder* yang saling terkait tetapi memiliki fasilitas yang tidak sesuai, atau dapat terjadi antara kebutuhan dan sumber daya.

5. *Prioritisation* : Tahap ini melakukan interaksi dengan *stakeholder* untuk mengidentifikasi kebutuhan-kebutuhan prioritas dari masing-masing kebutuhan agar memenuhi sumber daya yang tersedia pada organisasi.
6. *Requirements Checking*: Menganalisis sekumpulan kebutuhan dari hasil tahapan sebelumnya untuk memverifikasi dan memvalidasi berdasarkan aspek kelengkapan, konsistensi, dan kebutuhan nyata.

Semua jenis kebutuhan yang telah diperoleh tersebut kemudian dituangkan dalam bentuk dokumen yang berisi tentang kebutuhan sistem secara keseluruhan. Dokumen ini menjelaskan secara rinci tentang kesepakatan antara pengembang dengan klien, desain perangkat lunak yang akan dibangun, segala resiko yang akan dihadapi dan jadwal pembuatan perangkat lunak. Secara umum dokumen ini biasa disebut dengan *Software Requirements Specification (SRS)*.

Pada dokumen SRS akan dijelaskan juga mengenai kebutuhan fungsional dan non-fungsional dimana kebutuhan non-fungsional dibuat berdasarkan dokumen *IEEE standard 803:1993*. *IEEE 803:1993* mengelompokkan kebutuhan non-fungsional kedalam sejumlah kategori kualitas dari suatu perangkat lunak. Kategori-kategori tersebut secara umum dibagi kedalam 2 kelompok, yaitu faktor kualitas eksternal dari perangkat lunak dan faktor kualitas internal perangkat lunak. Faktor kualitas eksternal merupakan kategori kualitas yang dapat diobservasi atau menjadi ketertarikan utama dari pelanggan. Kategori-kategori yang termasuk di dalam kelompok ini antara lain :

- a. Ketepatan (*correctness*),
- b. *Robustness*,
- c. Unjuk Kerja (*performance*),

- d. Ketersediaan dan kualitas antar muka (*interface*),
- e. Keandalan (*reliability*), dan
- f. Ketersediaan (*availability*)

Sedangkan kualitas faktor internal merupakan kategori kualitas yang dapat di observasi atau menjadi ketertarikan utama dari pengembang. Seperti :

- a. Kemudahan membaca/memahami struktur perangkat lunak (*readability*),
 - b. Kemampuan untuk dilakukan pengujian (*testability*),
 - c. Ketersediaan dan kualitas dokumentasi (*documentation*),
 - d. Kemudahan pemeliharaan (*maintainability*), dan
- Adaptasi terhadap lingkungan berbeda (*portability*)

3. Desain

Desain adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

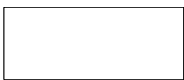




Analisis sistem dapat mendesain model dari sistem informasi yang diusulkan dalam bentuk *physical system* dan *logical model*. Bagan alir sistem (*system flowchart*) merupakan alat yang tepat digunakan untuk menggambarkan *physical system*.

Logical model dari sistem informasi lebih menjelaskan kepada *user* bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja. *Logical model* dapat digambarkan dengan menggunakan diagram arus data (*data flow diagram*). (Burch & Grudnitski, 1986).

a. *System Flow*

Menurut Kendal (2003), *System flow* adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem dimana bagan ini menjelaskan urutan prosedur-prosedur yang ada di dalam sistem dan biasanya dalam membuat *system flow* ditentukan oleh fungsi yang melaksanakan dan yang bertanggung jawab. Simbol-simbol dari *system flow* antara lain:

Tabel 2.1 *System Flow*

Simbol	Keterangan
	Simbol dari proses, dimana merupakan proses secara terkomputerisasi.
	Simbol dari keputusan atau <i>decision</i>
	Sistem penghubung, menunjukkan penghubung ke halaman yang masih sama.
	Simbol dari alur, menunjukkan alur dari proses
	Sistem penghubung, menunjukkan penghubung ke halaman yang berbeda atau halaman lain.

b. *Data Flow Diagram*

DFD adalah diagram yang menggunakan notasi-notasi ini untuk menggambarkan arus dari data sistem, sekarang di kenal dengan nama diagram arus data (*data flow diagram*). DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir.

a. *External entity*

External entity merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem.

b. *Data flow*

Data flow menunjukkan arus dari data yang berupa masukan untuk sistem atau hasil dari proses sistem dan dapat berbentuk sebagai berikut ini.

c. *Process* adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses.

d. *Data Store*

Data store adalah simpanan dari data yang berupa, suatu file database di sistem komputer, arsip atau catatan manual, dan suatu tabel acuan manual.

c. ***Entity Relationship Diagram***

Attribute adalah kolom di sebuah relasi. Macam-macam *attribute* yaitu :

a. *Simple Attribute*

Attribute ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya, misalnya *entity* mahasiswa yang *attribute*-nya NIM.

b. *Composite Attribute*

Composite Attribute adalah *Attribute* yang memiliki dua nilai harga, misalnya nama besar (nama keluarga) dan nama kecil (nama asli).

c. *Single Value Attribute*

Attribute yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *Attribute*-nya umur (tanggal lahir).

d. *Multi Value Attribute*

Attribute yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *Attribute*-nya pendidikan (SD, SMP, SMA).

e. *Null Value Attribute*

Attribute yang tidak memiliki nilai harga, misalnya *entity* tukang becak dengan *Attribute*-nya pendidikan (tanpa memiliki ijazah).

ERD ini diperlukan agar dapat menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah *entity* dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh perancang *database*.

(Kadir, 2008)

Untuk itu ERD dibagi menjadi 2 jenis model, yaitu :

a. *Conceptual Data Model (CDM)*

Merupakan jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

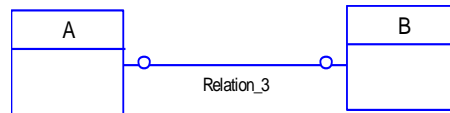
b. *Physical Data Model (PDM)*

Merupakan jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

ERD mempunyai 4 jenis hubungan antara lain :

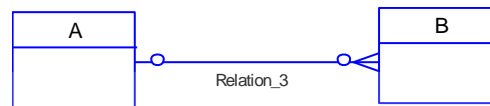
a. Hubungan *one-to-one* (1:1) menyatakan bahwa setiap entitas pada tipe entitas A paling banyak berpasangan dengan satu entitas pada tipe entitas B.

Begitu pula sebaliknya. Contoh :



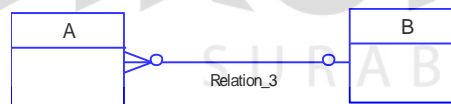
Gambar 2.4 Hubungan *one-to-one*

- b. Hubungan *one-to-many* (1:M) menyatakan bahwa setiap entitas pada tipe entitas A bisa berpasangan dengan banyak entitas pada tipe entitas B, sedangkan setiap entitas pada B hanya bisa berpasangan dengan satu entitas pada tipe entitas B. Contoh :



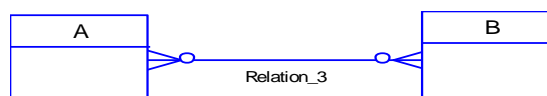
Gambar 2.5 Hubungan *one-to-many*

- c. Hubungan *many-to-one* (M:1) menyatakan bahwa setiap entitas pada tipe entitas A paling banyak berpasangan dengan satu entitas pada tipe entitas B dan setiap entitas pada tipe entitas B bisa berpasangan dengan banyak entitas pada tipe entitas A. Contoh :



Gambar 2.6 Hubungan *many-to-one*

- d. Hubungan *many-to-many* (M:N) Menyatakan bahwa setiap entitas pada suatu tipe entitas A bisa berpasangan dengan banyak entitas pada tipe entitas B dan begitu pula sebaliknya. Contoh :



Gambar 2.7 Hubungan *many-to-many*

Kardinalitas menggambar hubungan antara dua entitas dengan mengidentifikasi berapa banyak *instance* untuk setiap entitas yang nantinya dapat dihubungkan dengan setiap *instance* yang spesifik di entitas yang lain.

4. *Construction*

Software construction lebih di artikan sebagai pembuatan detail dari suatu pekerjaan, menciptakan satu software yang penting yang dikombinasikan dengan *code*, proses verifikasi, *testing unit*, dan testing yang terintegrasi, serta proses *debugging*. *Software construction* lebih sering di hubungkan dengan proses desain dan proses testing.

1. *Software Construction Fundamentals*

Pada tahap pertama, dilakukan pendefinisian dasar tentang prinsip-prinsip yang digunakan dalam proses implementasi seperti minimalisasi kompleksitas, mengantisipasi perubahan, dan standar yang digunakan.

2. *Managing Construction*

Bagian ini mendefinisikan tentang model implementasi yang digunakan, rencana implementasi, dan ukuran pencapaian dari implementasi tersebut.

3. *Practical Considerations*

Bagian ini membahas tentang desain implementasi yang digunakan, bahasa pemrograman yang digunakan, kualitas dari implementasi yang dilakukan, proses pengetesan dan integritas.

Dalam proses pengimplementasian saat ini, digunakan beberapa aplikasi pendukung yaitu :

a. PHP

PHP (*Personal Home Page*) adalah *script* bersifat server-side yang ditambahkan ke dalam HTML (*Hyper Text Markup Language*). *Script* ini akan membuat sesuatu aplikasi dapat diintegrasikan ke dalam HTML, sehingga suatu halaman web tidak lagi bersifat statis. Sifat server-side berarti pengerjaan *script* akan dilakukan di sever, baru kemudian hasilnya dikirim ke *browser* (Kurniawan, 2002)

Keunggulan dari sifatnya yang server-side tersebut adalah tidak diperlukan kompatibilitas *browser* atau harus menggunakan *browser* tertentu, karena server lah yang mengerjakan *script* PHP. Hasil yang dikirimkan kembali ke *browser* umumnya bersifat teks atau gambar saja sehingga bisa dikenal oleh *browser* apapun dan dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh server, misalkan koneksi ke database.

b. MySQL

Menurut Kadir (2008), MySQL (dibaca: mi-se-kyu-el) merupakan software yang tergolong sebagai DBMS (database Management System) yang bersifat open source. Open source menyatakan bahwa software ini dilengkapi dengan source code (code yang dipakai untuk membuat MySQL). Selain tentu saja bentuk executable-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi. Dan bisa diperoleh secara gratis dengan mengunduh di internet. MySQL awalnya dibuat oleh perusahaan konsultan bernama TeX yang berlokasi di Swedia. Saat ini pengembangan MySQL berada dibawah naungan MySQL AB. Sebagai software DBMS, MySQL memiliki sejumlah fitur seperti yang akan dijelaskan dibawah ini.

a. Multiplatform

MySQL tersedia pada beberapa platform (windows, linux, unix, dan lain-lain)

b. Andal, cepat dan mudah digunakan.

MySQL tergolong sebagai database server (server yang melayani permintaan terhadap database) yang andal, dapat menangani database yang besar dengan kecepatan tinggi. Mendukung banyak sekali fungsi untuk mengakses database dan sekaligus mudah untuk digunakan.

c. Jaminan keamanan akses

MySQL mendukung pengamanan database dengan berbagai criteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur user tertentu agar bisa mengakses data yang bersifat rahasia (missal gaji pegawai), sedangkan user lain tidak boleh sesuai dengan hak aksesnya.

d. Dukungan SQL

Seperti tersirat namanya, SQL mendukung perintah SQL (Structured Query Language). Sebagaimana diketahui SQL merupakan bahasa standar dalam pengaksesan database rasional. Pengetahuan akan SQL akan memudahkan siapa pun untuk menggunakan MySQL.).

2.6 Testing dan Implementasi

Tahap ini mendemonstrasikan sistem perangkat lunak yang telah selesai dibuat untuk dijalankan, apakah telah sesuai dengan kebutuhan yang telah dispesifikasikan dan dapat diadaptasi pada lingkungan sistem yang baru(England & sons, 2004). Terdapat 5 tahapan, yaitu :

a) *Test Plan*

Membuat *Software Testing Fundamentals* yang berisi tentang penjelasan penting mengenai *terminology testing*

b) *Test Levels*

Merancang *Test Levels* yang terbagi antara target dan objektif dari pengetesan.

c) *Test Techniques*

Penjabaran terhadap teknik yang digunakan termasuk dasar-dasar pengetesan berdasarkan intuisi dan pengalaman serta teknik pengetesan secara teknik *coding*, teknik *kesalahan*, teknik penggunaan, dan teknik terkait lainnya.

d) *Test-Related Measures*

Merupakan ukuran-ukuran pencapaian *testing* yang telah dilakukan untuk kemudian dilakukan evaluasi kembali.

e) *Test Process*

Merupakan tahapan terakhir dari *Software Testing*, yaitu pendefinisian yang berisi tentang aktivitas *testing* yang dilakukan.

2.7 Maintenance

Pada tahap ini akan dilakukan pendeskripsian pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan pengguna termasuk implementasi akhir dan proses peninjauan kembali. Pemeliharaan sistem ini terdiri dari beberapa jenis yaitu:

- a. *Corrective*, yaitu memperbaiki desain dan *error* pada program.
- b. *Adaptive*, yaitu memodifikasi sistem untuk beradaptasi dengan perubahan lingkungan.

- c. *Perfective*, yaitu melibatkan sistem untuk menyelesaikan masalah baru atau mengambil kesempatan untuk penambahan fitur.
- d. *Preventive*, yaitu menjaga sistem dari kemungkinan masalah di masa yang akan datang.

Prosedur pemeliharaan tersebut disusun dalam beberapa tahapan. Tahap awal adalah menyusun *software maintenance fundamentals* yang berisi tentang dasar-dasar pemeliharaan, segala yang dibutuhkan untuk melakukan pemeliharaan, dan kategori pemeliharaan. Selanjutnya adalah mendefinisikan *Key Issues in Software Maintenance*, yang berisi tentang teknik pemeliharaan, manajemen pemeliharaan dan biaya, serta ukuran pemeliharaan perangkat lunak. Tahap selanjutnya adalah mendefinisikan proses dan aktivitas pemeliharaan tersebut ke dalam *Maintenance Process*.(England & Sons, 2004).

