

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Pengembangan suatu sistem informasi dengan menggunakan metode *Behavior Driven Development* (BDD) pernah dilakukan oleh Vance dan Cickovski (2012) yang dimuat pada jurnal yang berjudul “*A Case Study on Developing a Classroom Web Application Using Behavior-Driven Development*”. Dalam jurnal tersebut dilakukan pengembangan terhadap sebuah aplikasi *web* yang bernama Narwhal. Narwhal adalah sebuah aplikasi yang memiliki fungsi menyerupai papan tulis dalam ruang kelas dan juga memiliki fungsi yang menyerupai *slide* persentasi *powerpoint*. Vance dan Cickovski dalam penelitiannya tersebut mengevaluasi efektifitas dalam penggunaan *Behavior Driven Development* (BDD) untuk pengembangan Narwhal. Narwhal dikembangkan dengan menggunakan Rails yang merupakan *web framework* untuk bahasa pemrograman Ruby, penggunaan RSpec yang merupakan *framework Behavior Driven Development* (BDD) untuk penulisan *test case* dalam bahasa pemrograman Ruby juga dilakukan, serta *tool* bernama Capybara untuk mensimulasikan interaksi antara *client* dengan aplikasi. Pengembangan aplikasi tersebut berhasil dilakukan peneliti seorang diri dalam waktu 3 bulan. Penggunaan *Behavior Driven Development* (BDD) yang melibatkan *client* secara teratur dalam pengumpulan *feedback* untuk dijadikan *test case* berperan besar terhadap pengembangan aplikasi tersebut.

Maurya dan Kumar (2012) dalam penelitiannya yang berjudul “*Analytical Study on Manual vs. Automated Testing Using with Simplistic Cost Model*” membandingkan manfaat yang didapatkan dari penggunaan pengujian yang terotomatisasi dengan pengujian manual. Dari penelitian tersebut peneliti menggunakan metode *Simplistic Cost Model* untuk mendapatkan waktu yang diperlukan untuk mencapai *Break Event Point* (BEP) dalam pengembangan pengujian otomatis, serta bagaimana memanfaatkan kombinasi pengujian manual dan pengujian terotomatisasi untuk menyelesaikan pengujian dalam waktu yang telah ditentukan.

Pengembangan terhadap sistem pengelolaan data Lembaga Penelitian dan Pengabdian pada Masyarakat (LPPM) juga dilakukan Yanuardi dan Jananto (2005). Dalam jurnalnya yang berjudul “Rancang Bangun Sistem Informasi Penelitian dan Pengabdian kepada Masyarakat”. Peneliti melakukan penelitian untuk perancangan dan pengembangan terhadap sistem informasi penelitian dan pengabdian kepada masyarakat untuk universitas Stikubank Semarang. Pengembangan sistem informasi dilakukan peneliti karena pengelolaan data pada LPPM universitas Stikubank Semarang tersebut masih bersifat manual yang dapat menghambat penyusunan laporan akreditasi. Metode yang digunakan dalam pengembangan sistem informasi pengelolaan data tersebut adalah SDLC (*System Development Life Cycle*) dan berbasis *desktop* sehingga data penelitian tidak dapat diakses secara *online*. Metode pengujian yang dilakukan peneliti tidak terdapat pada jurnal tersebut.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Informasi merupakan salah satu sumber daya penting dalam suatu organisasi yang digunakan sebagai bahan pengambilan keputusan. Sehubungan dengan hal itu, informasi haruslah berkualitas. Menurut Burch dan Grudnitski, kualitas informasi ditentukan oleh tiga faktor yaitu relevansi, ketepatan waktu dan akurasi (Kadir, 2003).

Relevansi berarti bahwa informasi benar-benar berguna bagi suatu tindakan keputusan yang dilakukan seseorang. Tepat waktu berarti bahwa informasi ada saat dibutuhkan sehingga bermanfaat untuk mengambil keputusan.

Akurasi berarti bahwa informasi bebas dari kesalahan (Kadir, 2003). Untuk mempermudah bagi para pekerja di suatu organisasi dalam memperoleh informasi, teknologi informasi bisa dilibatkan. Secara lebih khusus, organisasi umumnya menerapkan sistem informasi (Kadir, 2003).

Menurut Alter (1992), sistem informasi adalah kombinasi antarprosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

2.2.2 Data

Data sebagai elemen penting dalam menghasilkan informasi didefinisikan sebagai keterangan tertulis mengenai suatu fakta (kenyataan) yang masih berdiri sendiri, belum memiliki pengertian sebagai kelompok, belum

terkoordinasi satu sama lainnya, belum diolah sesuai keperluan tertentu (Zulkifli, 2000).

2.2.3 WWW (World Wide Web)

World Wide Web adalah sekelompok dokumen multimedia yang saling terkoneksi menggunakan *hypertext link*. Dengan mengklik pada *hot link*, anda dapat berpindah dari suatu dokumen ke dokumen lainnya. (Bustami, Ahmad 1999)

Ada dua hal penting yang harus diketahui bila belajar dengan WWW yaitu *software web browser* dan *software web server*. Dimana *software web browser* ini bertindak sebagai klien yang memungkinkan anda untuk melihat dan mendapatkan informasi dari *server web*, sedangkan *software web server* bertindak sebagai *server* yang memberikan/menyediakan informasi yang diminta oleh *browser* (Bustami, Ahmad 1999).

2.2.3.1 HTML (HyperText Markup Language)

Web didasarkan pada sebuah bahasa *hypertext* standar dengan nama *HyperText Markup Language* (HTML). HTML ini akan menyusun struktur dokumen dan memberikan dokumen tersebut suatu *dynamic hypertext link* ke dokumen lainnya, baik yang disimpan dalam komputer yang sama maupun di komputer berbeda. Menggunakan *hypertext link* ini (umumnya berwarna biru,

dicetak tebal, dan diberi garis), pengguna dapat mengklik *link* tersebut dan selanjutnya akan dipindahkan ke dokumen lain (Turban et al, 2003:214).

Ragget (1999) menjelaskan istilah HTML ini sebagai bahasa publikasi universal yang digunakan oleh *web* dan umumnya dimengerti oleh semua komputer.

2.2.3.2 CSS (*Cascading Style Sheet*)

Cascading Style Sheet (CSS) adalah sebuah dokumen yang berdiri sendiri dan dapat dimasukkan ke dalam kode HTML atau sekedar menjadi rujukan oleh HTML dalam pendefinisian *style*. CSS menggunakan kode – kode yang tersusun untuk menetapkan *style* pada elemen HTML atau dapat juga digunakan untuk membuat *style* baru yang biasa disebut *class*. (Sulistiyawan, Rubianto, & Saleh, 2008: 33)

Cascading Style Sheet (CSS) pelengkap HTML. CSS membentuk aturan tampilan atau *style* yang digunakan pada *website*, sehingga dapat dipisahkan antara isi *website* yang ditulis dengan HTML dan tampilan *web* yang ditulis dengan CSS. Biasanya CSS digunakan untuk menentukan warna, *font*, *layout* dan berbagai macam hal yang menyangkut tampilan. (Ollie, 2008: 50)

2.2.3.3 Javascript

Pada awalnya JavaScript dikembangkan pada *web browser* Netscape oleh *Brendan Eich* dengan nama *Mocha*, kemudian berubah menjadi *LiveScript* dan

yang akhirnya sampai sekarang ini menjadi JavaScript, JavaScript adalah bahasa skrip (*Scripting Language*) yaitu kumpulan instruksi perintah yang digunakan untuk mengendalikan beberapa bagian dari sistem operasi. Bentuk bahasa skrip dari JavaScript mengambil model penulisan pada pemrograman C dan JAVA, yang terdiri dari variabel, fungsi dan lainnya (Sibero, 2011).

2.2.4 Database

Menurut Connolly, dan Begg (2010, p65), *Database* adalah data logikal yang saling terhubung dan dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi. *Database* digunakan dalam kehidupan sehari-hari baik secara sadar maupun tidak sadar, contoh *database* yang sering kita gunakan dalam kehidupan sehari-hari yaitu:

1. Pembayaran kartu kredit.
2. Pembayaran liburan pada agen *travel*.
3. Pembayaran belanja di pasar swalayan.
4. Mencari buku di perpustakaan.
5. Peminjaman DVD.
6. Penggunaan *internet*.

2.2.4.1 DBMS (*Database Management System*)

Menurut Connolly, dan Begg (2010, p66), *Database Management System* adalah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan melakukan kontrol akses ke *database*. DBMS adalah perangkat lunak yang berinteraksi dengan program aplikasi dan pengguna *database*. Biasanya DBMS menyediakan fasilitas sebagai berikut:

1. DDL (*Data Definition Language*)

DDL memungkinkan pengguna untuk menentukan tipe data dan struktur dan kendala pada data yang akan disimpan dalam *database*.

2. DML (*Data Manipulation Language*)

DML memungkinkan pengguna untuk memasukkan, mengubah, dan menghapus *data* dari *database*.

3. Keamanan

Manajemen keamanan *database* yang meliputi:

1. Keamanan sistem: Pencegahan terhadap pengguna yang tidak berhak mengakses *database*.
2. Integritas system: Menjaga konsistensi data yang tersimpan.
3. *Concurrency control system*: yang memungkinkan berbagi akses *database*.
4. Pemulihan sistem: Mengembalikan *database* ke keadaan sebelumnya yang konsisten setelah kegagalan perangkat keras atau kegagalan *software*.

5. *User access catalog*: Berisi informasi dari data pengguna yang mengakses *database*.

2.2.4.2 MySQL

Menurut Kadir (2008, p2), MySQL merupakan *software* yang tergolong sebagai DBMS(*Database Management System*) yang bersifat *open source*. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di *internet* secara gratis.

MySQL awalnya dibuat oleh perusahaan konsultan bernama TcX yang berlokasi di Swedia. Saat ini pengembangan MySQL berada dibawah naungan perusahaan MySQL AB. Sebagai *software* DBMS, MySQL memiliki sejumlah fitur seperti:

1. *Multiplatform*

MySQL tersedia pada beberapa *platform* (Windows, Linux, Unix, dan lain-lain).

2. Andal, cepat, dan mudah digunakan

MySQL tergolong sebagai *database server* yang andal, dapat menangani *database* yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, dan sekaligus mudah untuk digunakan.

3. Jaminan keamanan akses

MySQL mendukung pengamanan *database* dengan berbagai kriteria pengaksesan.

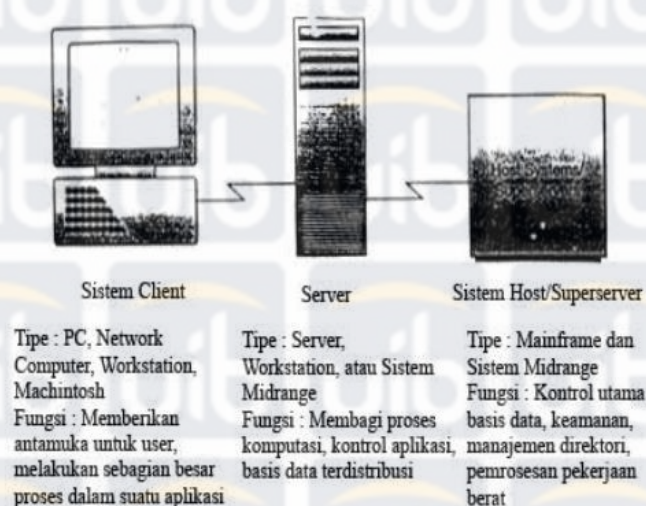
4. Dukungan SQL

SQL merupakan standar dalam pengaksesan *database* relasional.

Pengetahuan SQL akan memudahkan siapapun yang menggunakan MySQL.

2.2.5 *Server dan Web Server*

Menurut Nugroho (2004 : 5) secara global *server* dapat diartikan sebagai pusat dan difungsikan sebagai “pelayan” yang berguna untuk pengiriman data dan / atau penerimaan data serta mengatur pengiriman dan penerimaan data di antara komputer-komputer yang tersambung atau dengan kata lain *server* berfungsi menyediakan pelayanan terhadap klien.



Gambar 2.1 *Sistem Client Server*

Sedangkan *web server* menurut Nugroho (2004, p6) adalah sebuah bentuk *server* yang khusus digunakan untuk menyimpan halaman *website* atau *home page*. Komputer dapat dikatakan sebagai *web server* jika komputer tersebut memiliki suatu program *server* yang disebut *Personal Web Server (PWS)*. PWS ini difungsikan agar halaman *web* yang ada di dalam sebuah komputer *server* dapat dipanggil oleh komputer klien. Bentuk hubungan *server*, *client*, dan *host* dapat dilihat pada Gambar 2.1.

Macam-macam *web server*:

- Apache (*Open Source*)
- Xitami
- IIS
- PWS

2.2.5.1 Apache

Menurut Firdaus (2007 : 5) *Web server* merupakan sebuah bentuk *server* yang khusus digunakan untuk menyimpan halaman *website* atau *homepage*. Apache merupakan turunan dari *web server* yang dikeluarkan oleh NSCA yaitu NSCA HTTPd sekitar tahun 1995-an. Pada dasarnya, Apache adalah “APatCHy” (*patch*) dan pengganti dari NCSA HTTPd.

Apache *web server* merupakan tulang punggung permintaan dari klien yang menggunakan *browser*, seperti Netscape Navigator, Internet Explorer, Mozilla, lynx dan lain-lain. *Web server* dalam berkomunikasi dengan kliennya

menggunakan protokol HTTP (*Hyper Text Transfer Protocol*). Apache berada di bawah GNU, *General Public License* yang bersifat gratis sehingga Apache dapat diunduh secara gratis. Adapun pertimbangan dalam memilih Apache adalah :

1. Apache termasuk dalam kategori *free software* (*software* gratis).
2. Instalasi Apache sangat mudah.
3. Mampu beroperasi pada banyak *platform* sistem operasi, seperti Linux, Windows dan lain-lain.

Apache Web Server merupakan *web server* yang bersifat *open source* dan mempunyai *performance* yang sangat bagus, fleksibel dan mendukung berbagai macam *platform* sistem operasi seperti Windows NT/9x, UNIX, Netware 5x, OS/2 dan berbagai macam sistem operasi lainnya. Apache sangat cepat sekali mengeluarkan *update* terbarunya, sehingga mengurangi munculnya *bugs* dan kelemahan program.

2.2.5.2 PHP (*Hypertext Preprocessor*)

Menurut Firdaus (2007 : 18) PHP (*Hypertext Preprocessor*) merupakan bahasa yang hanya dapat berjalan pada *server* yang hasilnya dapat ditampilkan pada klien. Dalam mengeksekusi kode PHP pada sisi *server*(disebut *server side*) berbeda dengan mesin maya Java yang mengeksekusi program pada sisi klien (*client side*). Proses eksekusi kode PHP yang disisipkan pada halaman HTML.

PHP merupakan bahasa standar yang digunakan dalam dunia *website*. PHP adalah bahasa pemrograman yang berbentuk *script* yang diletakkan dalam *server*

web. Sekitar tahun 1994, Rasmus Lerdorf telah meletakkan bersama *Perl script* untuk membuat siapa yang telah melihat resumennya terkesan. Kemudian sedikit demi sedikit user mulai menyukai *script* ini.

2.2.5.3 *Yii Framework 2*

Yii adalah *framework* PHP yang memiliki performa tinggi, berbasis komponen untuk mempercepat proses pengembangan aplikasi *web* modern.

Yii adalah *generic web programming framework*, yang berarti *Yii* dapat digunakan untuk pengembangan berbagai jenis aplikasi *web* berbasis PHP. Karena arsitekturnya yang berbasis komponen dan *caching support* yang mutakhir, *Yii* sangat cocok digunakan untuk mengembangkan aplikasi yang ruang lingkupnya besar seperti *portal*, *forum*, *content management systems* (CMS), *e-commerce*, *RESTful Web Service*, dan lain-lainnya (Y. S. LLC, 2014).

2.2.6 *Behavior Driven Development (BDD)*

Behavior Driven Development (BDD) menurut Hatko, Mersmann, dan Puppe (2014 : 1) adalah sebagai berikut:

“Behaviour-Driven Development (BDD) is an agile Software Engineering (SE) methodology, that conforms to the Test-First principle: Test cases are developed previous to the software system itself. In BDD, executable test cases are directly derived from requirements stated in natural language, by employing a pattern matching approach. This enables stakeholders to actively participate in

the definition of requirements, which are a fundamental design input for the system under development.”

Berarti *Behavior Driven Development* (BDD) adalah salah satu bentuk metode pengembangan perangkat lunak *agile*, yang menganut prinsip *test-first*, dimana *test case* dikembangkan terlebih dahulu sebelum pengembangan sistem.

Test case pada BDD didasarkan pada *requirement* yang dikumpulkan dari *stakeholders*.

2.2.7 Software Testing

Quadri dan Farooq (2010) mendefinisikan pengujian pada perangkat lunak adalah sebagai berikut:

“Software testing is a process of verifying and validating that a software application or program meets the business and technical requirements that guided its design and development and works as expected and also identifies important errors or flaws categorized as per the severity level in the application that must be fixed. Software testing is also used to test the software for other software quality actors like reliability, usability, integrity, security, capability, efficiency, portability, maintainability, compatibility etc”

Dapat diambil kesimpulan bahwa pengujian terhadap perangkat lunak adalah sebuah proses untuk membuktikan bahwa perangkat lunak tersebut memenuhi persyaratan bisnis dan teknis yang digunakan sebagai rancangannya. Pengujian juga bertujuan untuk mengidentifikasi masalah dalam perangkat lunak yang harus segera diperbaiki, dan juga berfungsi untuk menguji faktor-faktor lain

dalam perangkat lunak yang berhubungan dengan kualitas seperti tingkat keandalan, kegunaan, integritas, keamanan, kemampuan, efisiensi, portabilitas, perawatan, kecocokan, dan lainnya.

Sedangkan *automated testing* menurut Mailewa dan Herath (2015) adalah proses dimana peralatan untuk melakukan pengujian secara otomatis digunakan untuk melakukan pengujian yang melakukan tindakan yang telah ditentukan secara berulang-ulang. Pengujian dilakukan dengan membandingkan *output* yang dihasilkan dengan *output* yang diharapkan dalam pengujian. Apabila *output* pengujian dan *output* yang diharapkan tidak sama, maka ada permasalahan yang harus diidentifikasi dan dilakukan perubahan pada perangkat lunak sampai mendapat *output* yang sama.

2.2.7.1 Codeception

Codeception adalah *testing framework* yang dikembangkan untuk PHP. Codeception diinspirasi oleh BDD (*Behavior Driven Development*), yang memperkenalkan cara baru dalam menulis *acceptance*, *functional*, dan bahkan *unit tests*. Codeception memanfaatkan PHPUnit dalam pengujiannya. (Davert, 2015:1)

2.2.8 Flowchart

Menurut Krismiaji (2010:71) dalam bukunya yang berjudul Sistem Informasi Akuntansi, menjelaskan bahwa:

Bagan alir merupakan teknik analitis yang digunakan untuk menjelaskan aspek-aspek sistem informasi secara jelas, tepat dan logis. Bagan alir menggunakan serangkaian simbol standar untuk menguraikan prosedur pengolahan transaksi yang digunakan oleh sebuah perusahaan, sekaligus menguraikan aliran *data* dalam sebuah sistem.

Terdapat beberapa jenis bagan alir yang biasa digunakan, yaitu sebagai berikut:

A. Bagan Alir Sistem (*System Flowchart*)

Bagan alir sistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan dalam sistem. Bagan alir sistem digambarkan dengan menggunakan simbol-simbol yang telah ditentukan.

B. Bagan Alir Dokumen (*Document Flowchart*)

Bagan alir dokumen (*document flowchart*) atau disebut dengan bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan didalam bagan alir sistem.

C. Bagan Alir Program (*Program Flowchart*)

Bagan alir program (*program flowchart*) terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowchart*) dan bagan alir program *computer* terinci (*detailed computer program flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika. Bagan alir logika program ini dipersiapkan oleh analis sistem.

D. Bagan Alir Skematik (*Schematic Flowchart*)

Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu menggambarkan prosedur dalam sistem. Perbedaannya adalah bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan dalam menjelaskan simbol-simbol bagan alir kepada orang yang kurang paham.





E. Bagan Alir Proses (*Process Flowchart*)

Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur.

2.2.9 Data Flow Diagram (DFD)

Data Flow Diagram yang nantinya akan di singkat dengan DFD adalah representasi dari sebuah sistem secara grafis yang digambarkan dengan sejumlah simbol tertentu untuk menunjukkan perpindahan data dalam proses-proses suatu sistem (Jogiyanto,1995).

Tabel 2.1 4 simbol dasar *Data Flow Diagram*

Simbol	Nama	Keterangan
	<i>Process</i>	Sebuah proses yang menunjukkan aksi atau sekumpulan aksi yang akan terjadi.
	<i>Data Flow</i>	Sebuah proses yang menunjukkan aksi atau sekumpulan aksi yang akan terjadi.
	<i>Data Source</i>	Sebuah tempat penyimpanan data, memungkinkan terjadinya pengambilan data untuk dimanipulasi.
	<i>External Entity</i>	Sebuah entitas yang dapat berupa seseorang, sekelompok orang, setiap aktor yang menerima informasi ataupun memasukkan data ke dalam sistem

Sumber: Jogiyanto (1995)

2.2.10 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram yang menggambarkan hubungan data antara obyek yang terdapat di dalam suatu sistem tanpa memberikan informasi apapun tentang fungsi yang menghasilkan atau menggunakan data tersebut. Komponen ERD terdiri dari :

1. Entitas adalah sesuatu yang digambarkan dengan sekumpulan atribut yang dimilikinya, dimana atribut tersebut akan dimanipulasi didalam sistem. Untuk memberi nama entitas digunakan kata benda, dimana masing-masing entitas tidak boleh memiliki nama yang sama.
2. *Relationship* mengindifikasikan hubungan antara dua atau lebih entitas dan menggunakan kata kerja untuk menggambarkan hubungan tersebut.
3. *Cardinality* menggambarkan banyaknya kemungkinan maksimum dari *relationship occurrences* sebuah entitas yang berpartisipasi dalam sebuah *relationship*.

Menurut J.G.Zheng (2010) hubungan (*relationship*) menyatakan keterkaitan antara beberapa entitas. Ada tiga jenis relasi di dalam diagram ERD, yaitu :

1. *One to One* (1-1)

Mempunyai pengertian setiap baris data pada tabel pertama dihubungkan hanya ke satu baris data pada tabel ke dua.





2. *One to Many* (1-M)

Mempunyai pengertian setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih data pada tabel ke dua.

3. *Many to Many* (M-M)

Mempunyai pengertian satu baris atau lebih data pada tabel pertama bisa dihubungkan ke satu baris atau lebih data pada tabel ke dua.

Tabel 2.2 4 simbol dasar *Entity Relationship Diagram*

No	Simbol	Keterangan
1	Entitas 	Entitas adalah sebuah kesatuan objek lain, setiap entitas dibatasi oleh atribut.
2	Atribut 	Atribut merupakan sifat atau karakteristik dari suatu entitas yang menyediakan penjelasan secara rinci.
3	Relasi 	Menyatakan himpunan relasi.
4	<i>Link</i> 	<i>Link</i> sebagai penghubung antara himpunan relasi dari himpunan entitas dengan atributnya.

Sumber : Shelly et al (2002)