



BAB II
TINJAUAN PUSTAKA
DAN LANDASAN TEORI

BAB II

TEORI PENUNJANG

2.1 Tinjauan Pustaka

Penelitian sistem pendukung keputusan sebelumnya berjudul “*Sistem Pendukung Keputusan Penentuan Uang Kuliah Tunggal dengan Menggunakan Metode Weighted Product*” dilakukan setelah adanya perubahan sistem pembayaran uang kuliah yang diterapkan kepada seluruh Perguruan Tinggi Negeri (PTN), yang disebut Uang Kuliah Tunggal (UKT). Dalam penentuan UKT digunakan beberapa kriteria sebagai yaitu penghasilan orang tua, pekerjaan orang tua, biaya air, biaya listrik, usia orang tua dan tanggungan keluarga. Dengan adanya sistem ini dapat membantu bidang administrasi dan registrasi universitas dalam pengambilan keputusan penggolongan uang kuliah tunggal bagi mahasiswa dan untuk mengurangi terjadinya kesalahan saat melakukan perhitungan dalam penggolongan uang kuliah tunggal [2].

Penelitian yang selanjutnya dilakukan oleh Zulfi Azhar dari STMIK Royal Kisaran yang berjudul “*Penentuan Penempatan Karyawan Baru di PDAM Kisaran dengan Metode SMART*”. Sistem ini berbasis web. Tujuan pengamatan adalah untuk menemukan kelemahan dan kelebihan pada sistem lama, dan untuk merancang sebuah sistem pendukung keputusan yang baru yang dapat memudahkan kinerja karyawan sehingga mengatasi masalah-masalah yang terjadi pada pengelolaan data penempatan karyawan baru di PDAM Kisaran. Teknik pengambilan keputusan multi kriteria ini didasarkan pada teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai– nilai dan setiap kriteria memiliki bobot yang menggambarkan seberapa penting ia dibandingkan dengan kriteria lain. Pembobotan ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik. Dengan sistem ini, proses hasil keputusan lebih cepat dan efektif [7].

Penelitian yang berikutnya berjudul “*Sistem Pendukung Keputusan Penerimaan Beasiswa dengan Metode SAW*”. Penelitian ini dilakukan untuk membantu menentukan Penerima Beasiswa di SMK Al-Islam, Beasiswa yang diadakan yakni Beasiswa Prestasi (BP) dan Beasiswa Kurang Mampu (BKM). Adanya batasan jumlah kuota dan kriteria tertentu, sehingga menyulitkan pihak tim seleksi beasiswa.

Tujuan dari penelitian ini adalah Peneliti akan merancang suatu sistem pendukung keputusan untuk menyeleksi calon penerima beasiswa dengan mengimplementasikan metode SAW (*Simple Additive Weighing*). Metode ini dipilih karena metode ini mampu melakukan suatu proses perancangan yang dilanjutkan dengan proses pembobotan untuk setiap atribut, sehingga akan dapat menyeleksi alternatif terbaik dari sejumlah alternative (siswa) berdasarkan kriteria dan bobot tertentu. Desain sistem dibuat dengan model *Context Diagram*, DAD, HIPO, ERD. Hasil akhir dari perancangan sistem pendukung keputusan penerimaan beasiswa ini yaitu berupa laporan data pendaftar, laporan diterima BP dan laporan diterima BKM berdasarkan jumlah kuota yang telah ditentukan [8].

Pada penelitian kali ini, penulis akan membangun sebuah sistem pendukung keputusan untuk menentukan rekomendasi besaran UKT bagi mahasiswa baru Politeknik Negeri Cilacap. Panitia PMB akan memasukkan data-data mahasiswa baru pada sistem, kemudian sistem akan secara otomatis melakukan perhitungan. Bagian Keuangan dapat menentukan kriteia dan subkriteria apa saja yang diperlukan. BAAK dapat mencetak laporan rekomendasi untuk kemudian dirapatkan lagi. Pada sistem ini selain terdapat fitur cetak laporan, terdapat juga fitur history laporan sehingga dapat melihat laporan pada periode sebelum-sebelumnya. Sistem ini dibuat dengan menggunakan bahasa pemrograman *PHP* dan *database MySQL*. Sistem akan dibangun menggunakan metode *Simple Multi-Attribute Rating Technique (SMART)*. Metode *SMART* digunakan karena metode ini memiliki kelebihan antara lain dapat melakukan penambahan atau pengurangan alternatif, sehingga setiap penambahan alternatif atau data mahasiswa tidak akan mempengaruhi perhitungan pembobotan, karena setiap penilaian alternatif tidak saling bergantung. Kelebihan lainnya adalah perhitungan *SMART* sangat sederhana sehingga tidak memerlukan perhitungan matematis yang rumit yang memerlukan pemahaman matematika yang kuat[9].

2.2 Landasan Teori

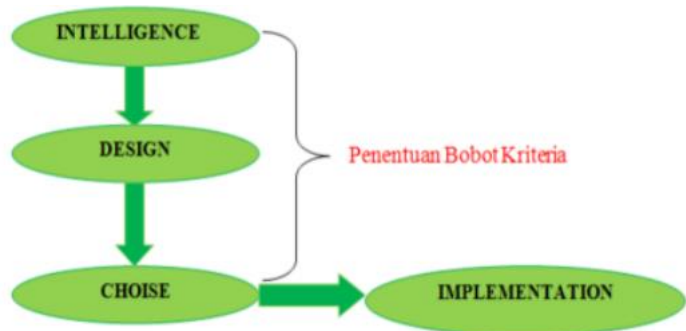
2.2.1. Sistem

Sistem adalah kumpulan orang yang saling bekerja sama dengan dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan fungsi untuk mencapai

tujuan. Sistem memiliki beberapa karakteristik atau sifat yang terdiri dari komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem, dan sasaran sistem.

2.2.2. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) adalah sistem berbasis komputer yang interaktif, yang membantu pengambil keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tak terstruktur dan semi terstruktur. Sistem pendukung keputusan tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, namun memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia. Berikut dapat dilihat pada Gambar 2.1 tahapan dalam penerapan SPK[10] :



Gambar 2. 1 Fase Pengambilan Keputusan

1. *Intelligence*
Kemampuan untuk mempresepsikan sebuah informasi, dan mempertahankannya sebagai pengetahuan yang diterapkan.
2. *Design*
Membuat rencana dalam bentuk prototipe atau spesifikasi untuk membangun objek atau sistem sebelum melakukan kegiatan atau sebuah proses.
3. *Choice*

Pada tahap ini ditentukan sebuah pilihan dari berbagai aspek yang diperlukan yang sesuai dengan model yang telah dirancang.

4. *Implementation*

Implementasi diterapkan pada teknologi untuk menggambarkan interaksi unsur-unsur dalam Bahasa pemrograman. Penerapan digunakan untuk mengenali sumber daya pemrograman yang ditulis ke dalam suatu program.

Bobot kriteria pada sistem ini ditentukan berdasarkan tingkat prioritas dari masing-masing kriteria, yang mana penentuannya dilakukan oleh bagian Keuangan dan BAAK.

2.2.3. Metode SMART

Di dalam Sistem Pendukung Keputusan terdapat berbagai metode yang dimanfaatkan untuk menyelesaikan suatu permasalahan, salah satunya ialah metode *Simple Multi Attribute Rating Technique* atau bisa disingkat SMART. Metode SMART merupakan teknik pengambilan keputusan multikriteria ini didasarkan pada teori bahwa setiap alternatif terdiri dari sejumlah kriteria memiliki bobot yang menggambarkan seberapa penting dibandingkan dengan kriteria lain. Pembobotan ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik. Alasan menggunakan metode *SMART* karena metode ini memiliki **kelebihan** antara lain dapat melakukan penambahan atau pengurangan alternatif, sehingga setiap penambahan alternatif atau data mahasiswa tidak akan mempengaruhi perhitungan pembobotan, karena setiap penilaian alternatif tidak saling bergantung. **Kelebihan lainnya** adalah perhitungan *SMART* sangat sederhana sehingga tidak memerlukan perhitungan matematis yang rumit yang memerlukan pemahaman matematika yang kuat[9].

2.2.4. Uang Kuliah Tunggal

Berdasarkan Peraturan Menteri Riset, Teknologi, dan Pendidikan Tinggi Nomor 39 Tahun 2017 tentang biaya kuliah tunggal dan uang kuliah tunggal pada Perguruan Tinggi Negeri, maka yang dimaksud dengan biaya kuliah tunggal adalah keseluruhan biaya operasional yang terkait langsung dengan proses pembelajaran mahasiswa per semester pada program studi di PTN, sedangkan uang kuliah tunggal atau yang biasa disingkat UKT adalah sebagian biaya kuliah tunggal yang ditanggung setiap mahasiswa berdasarkan kemampuan ekonominya[11].

Uang kuliah tunggal ditentukan dengan melihat kemampuan ekonomi mahasiswa, mahasiswa dengan kemampuan ekonomi yang tinggi tentu seharusnya membayar uang kuliah tunggal yang lebih tinggi daripada mahasiswa yang kemampuannya kurang. Berikut pembagian 8 golongan UKT yang ada di Politeknik Negeri Cilacap dapat dilihat pada Tabel 2.1 :

Tabel 2. 1 Golongan UKT

No.	Nama Golongan	Nominal UKT
1.	Golongan I	500.000
2.	Golongan II	1.000.000
3.	Golongan III	2.000.000 – 2.400.000
4.	Golongan IV	3.000.000 – 3.500.000
5.	Golongan V	4.000.000 – 4.500.000
6.	Golongan VI	5.000.000 – 5.500.000
7.	Golongan VII	6.500.000
8.	Golongan VIII	7.500.000 – 8.000.000

2.2.5. Situs Web (*Website*)

Web merupakan bagian dari layanan yang dapat berjalan di atas teknologi internet. Halaman web dapat diakses dengan adanya teknologi *web server* sebagai penyedia halaman web, HTML sebagai bahasa baku dan HTTP sebagai jalur pengiriman dokumen web [12].

Website adalah halaman informasi yang disediakan melalui jalur internet, sehingga bisa diakses di manapun dan selama terkoneksi dengan jaringan internet. *Website* merupakan komponen yang terdiri dari teks, gambar, suara, animasi, sehingga lebih menyediakan informasi yang menarik untuk dikunjungi. Berdasarkan sifatnya, *Website* terbagi menjadi 2 jenis :

1. *Website* dinamis : sebuah *Website* yang menyediakan konten atau isi yang selalu berubah-ubah setiap saat. Biasanya *Website* ini

digunakan oleh *Website* surat kabar seperti kompas.com , detik.com, dan merdeka.com.

2. *Website* statis : sebuah *Website* yang isinya sangat jarang diubah. Biasanya *Website* ini digunakan oleh organisasi atau instansi, seperti profil organisasi [13].

2.2.6. Rekayasa Perangkat Lunak (RPL)

Adalah suatu disiplin rekayasa yang berkonsentrasi terhadap seluruh aspek produksi perangkat lunak. Mengadopsi pendekatan yang sistematis dan terorganisir terhadap pekerjaannya dan menggunakan tool yang sesuai serta teknik yang ditentukan berdasarkan masalah yang akan dipecahkan, kendala pengembangan dan sumber daya yang tersedia [14].

2.2.7. Metode *Waterfall*

Metode SDLC air terjun (*Waterfall*) sering disebut juga model sekuensial linear (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Kelebihan dari metode ini yakni kualitas dari sistem yang dihasilkan akan baik karena pelaksanaannya dilakukan secara bertahap, proses pengembangan model fase *one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi, dan dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya [15] . Berikut adalah tahapan metode *waterfall* menurut Ian Sommerville, 2011 :

1. *Requirement Definition*

Proses pendefinisian persyaratan dimulai dengan melakukan observasi dengan *user system*, yang kemudian hasil observasi tersebut didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem yang akan dibuat.

2. *System and Software Design*

Desain perangkat lunak dan sistem adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Proses ini juga harus didokumentasikan.

3. *Implementation and Unit Testing*

Pada tahap ini adalah proses dari menerjemahkan perancangan ke dalam serangkaian program yang disebut unit program. Pengujian dilakukan untuk memverifikasi setiap unit dapat bekerja dengan baik dan sesuai dengan spesifikasinya.

4. *Integration and System Testing*

Unit program diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah terpenuhi sesuai dengan yang diinginkan *user*. Setelah selesai, sistem dikirimkan ke *user*.

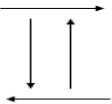
5. *Maintenance* (Pendukung atau Pemeliharaan)









Tidak memungkinkan sebuah perangkat lunak akan mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap ini mengulangi dari proses pengembangan mulai dari analisis spesifikasi untuk perangkat lunak yang sudah ada, tapi tidak membuat yang baru.

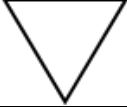

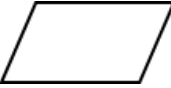


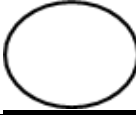


2.2.8. Flowchart

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada programmer. Dengan begitu, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem[16]. *Flowchart* digambarkan dengan menggunakan simbol-simbol. Berikut simbol-simbol pada *flowchart* dapat dilihat pada Tabel 2.2 :

Tabel 2. 2 Simbol-simbol pada *Flowchart*

Simbol	Nama	Fungsi
	<i>Connecting Line</i>	Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses

	<p><i>Communication Link</i></p>	<p>Berfungsi untuk transmisi data dari satu lokasi ke lokasi lain</p>
	<p><i>Connector</i></p>	<p>Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang sama</p>
	<p><i>Offline Connector</i></p>	<p>Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman berbeda</p>
	<p><i>Processing</i></p>	<p>Digunakan untuk menunjukkan pengolahan yang akan dilakukan oleh komputer</p>
	<p><i>Manual Operation</i></p>	<p>Digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer</p>
	<p><i>Decision</i></p>	<p>Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu</p>
	<p><i>Predefined Process</i></p>	<p>Digunakan untuk mempersiapkan penyimpanan yang sedang/akan digunakan dengan memberi harga awal</p>
	<p><i>Terminal</i></p>	<p>Digunakan untuk memulai atau mengakhiri program</p>

	<i>Offline Storage</i>	Berfungsi untuk menunjukkan bahwa data akan disimpan ke media tertentu
	<i>Manual Input Simbol</i>	Digunakan untuk menginputkan data secara manual dengan keyboard
	<i>Input/Output</i>	Digunakan untuk menyatakan input dan output tanpa melihat jenisnya
	<i>Punched Card</i>	Digunakan untuk menyatakan masukan dan pengeluaran yang berasal dari card
	<i>Disk Storage</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari disk
	<i>Magnetic Tape</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari pita magnetis
	<i>Document</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari dokumen
	<i>Display</i>	Digunakan untuk menyatakan masukan dan keluaran melalui layer monitor

2.2.9. Unified Modeling Language (UML)




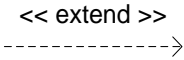
Unified Modeling Language (UML) adalah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. UML


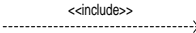
merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung [5].

1. *Use case Diagram*

Use case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [5]. Berikut adalah simbol-simbol yang ada pada *Use case Diagram* pada Tabel 2.3 :

Tabel 2. 3 Simbol-simbol pada *Use case*

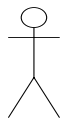
Simbol	Nama	Fungsi
	<i>Use case</i>	Fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor. Biasanya dinyatakan dengan menggunakan kata kerja
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat sendiri
	<i>Association</i>	Komunikasi antar actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat




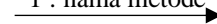


		berdiri sendiri walau tanpa <i>use case</i> tambahan itu
	<i>Generalization</i>	Hubungan generalisasi dari spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

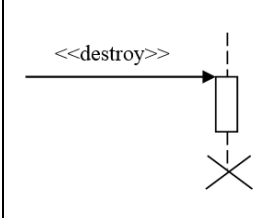
2. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terdapat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansi menjadi objek itu. Berikut adalah simbol-simbol yang ada pada diagram sekuen dapat dilihat pada Tabel 2.4 :

Tabel 2. 4 Simbol-simbol pada *Sequence Diagram*

Simbol	Nama	Fungsi
	<i>Aktor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat sendiri




	<i>Lifeline</i>	Menyatakan kehidupan suatu objek
<div style="border: 1px solid black; padding: 2px; display: inline-block;">Objek : kelas</div>	<i>Objek</i>	Menyatakan objek yang berinteraksi pesan
	<i>Waktu Aktif</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
<p style="text-align: center;"><<create>></p> 	<i>Pesan Tipe Create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
<p style="text-align: center;">1 : nama metode</p> 	<i>Pesan Tipe Call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
	<i>Pesan Tipe Send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
	<i>Pesan Tipe Return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian

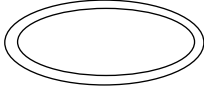
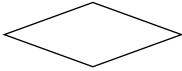
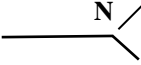
	<p><i>Pesan Tipe Destroy</i></p>	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka destroy</p>
---	----------------------------------	---

3. *Entity Relationship Data (ERD)*

Entity Relationship Data (ERD) adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODBMS maka perancangan ERD tidak perlu dilakukan. ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Berikut merupakan simbol-simbol yang digunakan pada ERD dapat dilihat pada Tabel 2.5 :

Tabel 2. 5 Simbol-simbol yang Digunakan ERD

Simbol	Nama	Fungsi
	<p><i>Entitas /entity</i></p>	<p>Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.</p>
	<p><i>Atribut</i></p>	<p>Field atau kolom data yang butuh disimpan dalam suatu entitas.</p>
	<p><i>Atribut primary key</i></p>	<p>Field atau kolom data yang butuh disimpan dalam suatu entitas dan</p>

		digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom; asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
	<i>Atribut multivalai / multivalue</i>	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
	<i>Relasi</i>	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
	<i>Asosiasi / association</i>	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many yang

		menghubungkan entitas A dan entitas B.
--	--	--

2.2.10. Basis Data

Basis data merupakan sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Jadi, pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

SQL adalah bahasa pemrograman yang digunakan untuk mengolah basis data, sedangkan MySQL adalah sebuah *brand software database management system* (DBMS) untuk mengolah basis data menggunakan Bahasa SQL itu sendiri [17].

2.2.11. Pengujian Perangkat Lunak

Pengujian perangkat lunak memiliki tujuan untuk memastikan bahwa perangkat lunak yang sedang dikembangkan sudah berjalan sesuai dengan fungsionalitas yang diharapkan oleh para *user*. Pengujian perangkat lunak ini *mrnjamin* kualitas perangkat lunak yang merupakan bagian dari daur hidup pengembangan perangkat lunak. (M. Sidi Mustaqbal, 2015). Jenis pengujian perangkat lunak terbagi menjadi dua yaitu *white box testing* dan *black box testing*.

Black box testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian *black box testing* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi.

Dalam pengujian *black box testing* digunakan alat untuk pengumpulan data yang disebut dengan *user acceptance test*, dokumen ini terdiri deskripsi indikator dari prosedur – prosedur pengujian fungsionalitas dari perangkat lunak [18].

-HALAMAN INI SENGAJA DIKOSONGKAN-