

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Telah ada beberapa penelitian yang dilakukan terkait dengan Analisa perbandingan metode SMART dalam Penerimaan Siswa Baru Pada SMA Hang Tuah Belawan, diantaranya adalah :

Menurut penelitian Sundari (2019), “Penerapan Metode SMART Dalam Pengambilan Keputusan Penerima Beasiswa Yayasan AMIK Tunas Bangsa”. Pada penelitian ini AMIK Tunas Bangsa belum memiliki sistem pendukung keputusan dalam menentukan penerima beasiswa yayasan sehingga kemungkinan terjadinya kesalahan dalam merekomendasikan calon menerima beasiswa dan sangat rentan dalam praktik nepotisme. Oleh karena itu, penulis melakukan penelitian dalam menentukan penerima beasiswa yayasan. Metode penelitian yang diterapkan dalam menentukan penerima beasiswa yayasan AMIK Tunas Bangsa adalah metode SMART (*Simple Multi Attribute rating Technique*), karena metode ini mampu menyelesaikan masalah pengambilan keputusan multikriteria berdasarkan teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai-nilai dan setiap kriteria memiliki bobot yang menggambarkan seberapa penting kriteria tersebut dengan kriteria lain. Kriteria yang digunakan dalam menentukan penerima beasiswa yayasan pada penelitian ini adalah IPK, penghasilan orang tua dan jumlah tanggungan. Hasil dari penelitian ini adalah rekomendasi penerima beasiswa.

Dengan menggunakan metode SMART, maka rekomendasi yang dihasilkan lebih akurat.

Menurut penelitian Safii (2018), “Penerapan Metode *Simple Multi Attribute Rating Technique* (SMART) Sebagai Motivasi Pegawai Dalam Peningkatan Prestasi”. Pada penelitian ini masalah yang di hadapin adalah dalam menentukan prestasi pegawai, belum adanya alat yang dapat membantu dalam mengolah data kriteria tersebut. Solusinya adalah dengan memanfaatkan metode SMART dengan pemecahan masalah multikriteria sebagai alat bantu dalam pengambilan keputusan. Dengan pemanfaatan metode ini diharapkan dapat memberikan solusi dan rekomendasi yang tepat bagi pegawai yang berprestasi sehingga hasilnya dapat diterima oleh seluruh pegawai secara objektif. Dengan adanya model sistem pendukung keputusan ini diharapkan dapat meningkatkan motivasi pegawai dalam melakukan pekerjaannya dan membantu *top level management* dalam mengolah data pegawai.

Menurut penelitian Irawan (2019), “Sistem Pendukung Keputusan Penerimaan Siswa Baru Di SMK Negeri 1 Tapung Hulu Menggunakan Metode *Simple Multi Attribut Rating Technique* (SMART)”. Kegiatan penerimaan siswa baru di SMK Negeri 1 Tapung Hulu masih menggunakan sistem manual yaitu peserta mengisi formulir pendaftaran pada form yang telah disediakan oleh pihak sekolah, kemudian data-data peserta tersebut diolah kembali oleh panitia menggunakan *Microsoft Excel*. Penggunaan *Microsoft Excel* sebagai aplikasi yang digunakan untuk proses seleksi cukup efektif dikarenakan proses seleksi hanya berpedoman pada satu kriteria yaitu nilai Ujian Nasional (UN) dan proses ini

membutuhkan waktu yang cukup lama. Agar proses seleksi lebih efektif panitia pemilihan siswa baru akan melibatkan beberapa kriteria yang digunakan sebagai pedoman untuk proses seleksi calon siswa baru, diantaranya nilai Ujian Nasional (UN), nilai Ujian Akhir Sekolah (UAS), serta nilai Tes Akademik sehingga dalam proses penyelesaiannya diperlukan sebuah sistem pendukung keputusan untuk multikriteria. Metode *Simple Multi Attribute Rating Technique* atau yang biasa disingkat SMART merupakan metode dalam pengambilan keputusan multi atribut. Teknik pengambilan keputusan multi atribut ini digunakan untuk mendukung pembuat keputusan dalam memilih beberapa alternatif. Setiap pembuat keputusan harus memiliki sebuah alternatif yang sesuai dengan tujuan yang dirumuskan. Sistem pendukung keputusan yang dibangun sangat membantu untuk mempercepat pengolahan data dalam pengambilan keputusan untuk menentukan calon siswa baru yang layak dan tidak layak untuk diterima. Sistem pendukung keputusan menggunakan metode SMART yang dibangun sangat membantu untuk mempercepat pengolahan data dalam pengambilan keputusan untuk menentukan calon siswa baru yang layak dan tidak layak untuk diterima.

Menurut penelitian Herti (2019), “Perancangan Sistem Pendukung Keputusan Seleksi Penerimaan Beras Sejahtera (Rastra) Pada Kelurahan Tanjung Pinang Jambi”. Permasalahan Kelurahan Tanjung Pinang Jambi yaitu membutuhkan waktu yang cukup lama dan banyak calon penerima yang mendaftarkan diri untuk mendapatkan bantuan beras sejahtera sehingga terjadinya ketidaktepatan dalam seleksi penerimaan yang menyebabkan terjadinya kecemburuan sosial antara masing-masing calon penerima. Oleh karena itu,

penelitian ini bertujuan memberikan solusi untuk permasalahan yang terjadi dengan menawarkan sistem pendukung keputusan seleksi penerimaan beras sejahtera menggunakan bahasa pemrograman PHP, *database* MySQL, dan metode SMART dimana penulis melakukan pengembangan sistem dengan metode *Waterfall* dan menggunakan pendekatan model sistem *Unified Model Language* menggunakan *Usecase Diagram*, *Activity Diagram*, *Class Diagram* dan *Flowchart*. Sehingga aplikasi dapat digunakan oleh dapat memberikan kemudahan memilih calon penerima yang layak mendapatkan bantuan beras dengan cepat dan tepat.

Menurut penelitian Ismail (2019), “Sistem Pendukung Keputusan Penerimaan Siswa Baru Menggunakan Algoritma SMART Pada Sekolah Menengah Atas Negeri 2 Soppeng”. Penentuan kelulusan penerimaan siswa baru pada SMAN 2 Soppeng memakan waktu cukup lama karena penilaian multi kriteria yang masih manual dan banyaknya pendaftaran siswa baru. Untuk itu dibuat suatu sistem yang dapat membantu proses penentuan kelulusan siswa berdasarkan nilai kriteria yang digunakan. Adapun sistem yang dibuat yaitu sistem penunjang keputusan menggunakan metode matematis. Metode yang digunakan pada penelitian ini yaitu metode SMART. Metode SMART merupakan metode matematis yang digunakan untuk menghitung nilai kriteria. Keunggulan dari metode tersebut cukup sederhana sehingga mudah untuk dikembangkan. Dengan penerapan sistem penunjang keputusan penerimaan siswa baru menggunakan metode SMART, pekerjaan penentuan kelulusan menjadi mudah, penyeleksian menjadi cepat dan akurat. Hal ini dirasakan sangat membantu panitia penerimaan siswa baru karena menjadi mudah dan cepat.

Perbedaan penelitian yang penulis lakukan dengan penelitian sebelumnya adalah penulis menggunakan metode SMART dalam penerimaan dan penyeleksian siswa baru. Hasil dari penelitian ini dapat menjadi solusi yang optimal dan akurat bagi SMA Hang Tuah Belawan dalam pengambilan keputusan penerimaan dan penyeleksian siswa baru.

II.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dan satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan). (Novriansyah, 2015 : 1).

Menurut Novriansyah (2015), karakteristik sistem pendukung keputusan adalah :

1. Mendukung proses pengambilan keputusan dalam suatu perusahaan.
2. Adanya *interface* manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai kebutuhan.

5. Memiliki sub sistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
6. Memiliki dua komponen utama yaitu data dan model.

II.3. Penerimaan Siswa Baru

Penerimaan siswa baru merupakan salah satu kegiatan yang pertama dilakukan, biasanya dengan mengadakan seleksi calon siswa. Pengelolaan siswa baru ini harus dilakukan secara terorganisir dan terencana, sehingga kegiatan pembelajaran dapat dilaksanakan pada hari pertama setiap tahun ajaran baru. (Ismail, 2019 : 4).

II.4. Metode *Simple Multi Attribute Rating Technique* (SMART)

Simple Multi-Attribute Rating Technique (SMART) merupakan metode pengambilan keputusan multikriteria yang dikembangkan oleh Edward pada tahun 1977. SMART merupakan teknik pengambilan keputusan multikriteria yang 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 (Ismail, 2019 : 4).

SMART menggunakan *linear additive model* untuk meramal nilai setiap alternatif. SMART merupakan metode pengambilan keputusan yang fleksibel. SMART lebih banyak digunakan karena kesederhanaannya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon. Analisa yang

terlibat adalah transparan sehingga metode ini memberikan pemahaman masalah yang tinggi dan dapat diterima oleh pembuat keputusan. Pembobotan pada SMART menggunakan skala antara 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif.

Model fungsi *utility linear* yang digunakan oleh SMART seperti pada Persamaan (Herti, 2019 : 2):

$$\text{SMART} = \sum_{j=1}^k w_j u_{ij} \dots\dots\dots (1)$$

Di mana :

- 1) w_j adalah nilai pembobotan kriteria ke- j dari k kriteria,
- 2) u_{ij} adalah nilai *utility* alternatif i pada kriteria j .
- 3) Pemilihan keputusan adalah mengidentifikasi mana dari n alternatif yang mempunyai nilai fungsi terbesar. Nilai fungsi ini juga dapat digunakan untuk meranking alternatif.

Langkah-langkah dalam melakukan metode ini adalah:

1. W_j adalah nilai pembobotan kriteria ke $-j$ dari k kriteria,
2. U_{ji} adalah nilai *utility* alternatif i pada kriteria j ,
3. Pemilihan keputusan adalah mengidentifikasi mana dari n alternatif yang mempunyai nilai fungsi terbesar.
4. Nilai fungsi ini dapat digunakan untuk meranking n alternatif

Menghitung nilai normalisasi bobot, seperti pada Persamaan (Herti, 2019 : 2):

$$Nw_j = \frac{w_j}{\sum_{n=1}^k w_j} \dots\dots\dots (2)$$

Di mana:

1. Nw_j adalah normalisasi bobot kriteria ke $-j$

2. W_j adalah nilai bobot kriteria ke $-j$
3. K adalah jumlah kriteria
4. n adalah bobot kriteria ke $-n$

Menghitung nilai *utility*, seperti pada Persamaan (Herti, 2019 : 2):

$$u_{ij}=(v_{ij}) \dots\dots\dots (3)$$

Di mana:

1. U_{ij} adalah *utility* kriteria ke $-j$ untuk alternatif i
2. V_{ij} adalah nilai kriteria ke $-j$ untuk alternatif i
3. $F(v_{ij})$ adalah fungsi kriteria ke $-j$ untuk alternatif i .

II.5. *Unified Modeling Language (UML)*


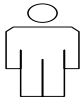

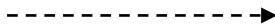

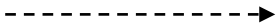
UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. Pemodelan yang peneliti gunakan sebanyak 4 pemodelan sebagai berikut. (Simatupang, 2018 : 5).

1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja

yang berhak menggunakan fungsi-fungsi tersebut. Adapun simbol dari *usecase diagram* dapat dilihat pada Tabel II.1.

Tabel II.1. Simbol Use Case Diagram

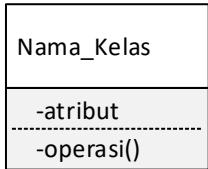


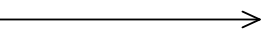

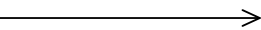

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor: biasanya dinyatakan dengan menggunakan kata kerja diawal di awal frase nama use case.</p>
<p>Aktor/actor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.</p>
<p>Asosiasi / association</p> 	<p>Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.</p>
<p>Ekstensi / extend</p> <p><<extends></p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan misal Arah panah mengarah pada use case yang ditambahkan.</p>
<p>Generalisasi / generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan / <i>include</i></p> 	<p>Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan</p>

(Sumber : Julianto Simatupang, 2019)

2. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Adapun simbol dari *class diagram* dapat dilihat pada Tabel II.2.

Tabel II.2. Simbol Class Diagram


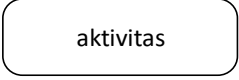


Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antar muka/interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi / association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
Asosiasi berarah / directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi 	Relasi antar kelas dengan makna generalisasi- generalisasi- spesialisasi (umum khusus).
Kebergantungan / dependency 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi /aggregation 	Relasi antar kelas dengan makna semua-bagian (whole- part).

(Sumber : Julianto Simatupang, 2019)

3. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan *actor*. Adapun simbol *activity diagram* dapat dilihat pada Tabel II.3.

Tabel II.3. Simbol Activity Diagram




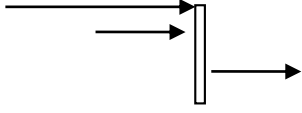
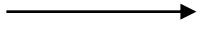
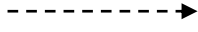
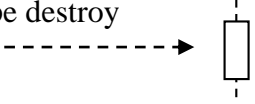
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari Satu
Penggabungan / Join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

(Sumber : Julianto Simatupang, 2019)

4. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Adapun simbol *usecase diagram* dapat dilihat pada Tabel II.4.

Tabel II.4. Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor  atau nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Garis hidup / lifeline 	Menyatakan kehidupan suatu objek.
Objek nama objek	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
Pesan tipe create << create >> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan tipe send 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 1 : keluaran 	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.
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah yang mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

(Sumber : Julianto Simatupang, 2019)

II.6. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah *web* dan bisa digunakan pada dokumen HTML. PHP dirancang untuk dapat bekerja sama dengan *database server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses *database* menjadi begitu mudah (Anwar, 2016 : 3).

Untuk mencoba PHP, tuliskan sintaks bahasa pemrograman PHP dan berikan nama pertama.php, kemudian file ini disimpan ke dalam folder C:\xampp\htdocs. Untuk memudahkan manajemen file sebaiknya dibuatkan folder tersendiri. Dalam hal ini misalnya folder diberi nama coba menjadi C:\xampp\htdocs\coba. Untuk melihat hasil program PHP, cobalah buka *browser* dan ketikkan <http://localhost/coba/pertama.php>.

Berikut sintaks dasar bahasa pemrograman PHP dengan namapertama.php.

File pertama.php:

```
<html>

<head>

    <title>Contoh Sederhana</title>

</head>

<body>

    <?php

        echo("Hallo apa kabar? Nama saya PHP script");

    ?>

</body>
```

</html>

Hasilnya seperti tampak pada halaman <http://localhost/coba/pertama.php> pada Gambar II.1.



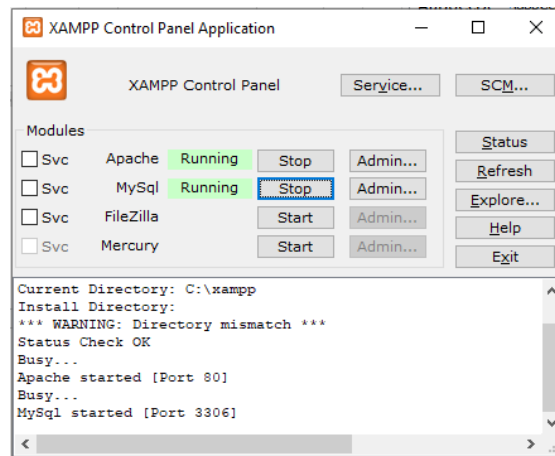
Gambar II.1. Hasil Sintak Dasar PHP
(Sumber : Anwar, 2016 : 3)

II.7. MySQL

MySQL (*My Structure Query Language*) merupakan sebuah program pembuat *database* yang bersifat *open source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *windows* maupun *linux*. MySQL juga merupakan sebuah perangkat lunak sistem manajemen basis data SQL yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *multiuser*. MySQL juga sering dikenal dengan nama sistem manajemen *database* relasional. (Anwar, 2016 : 3).

II.7.1. Mengakses MySQL

Untuk mengakses MySQL buka aplikasi XAMPP, tekan tombol *Start* pada *Apache* dan *MySql* seperti pada Gambar II.2.



Gambar II.2. Aplikasi XAMPP
(Sumber : Anwar, 2016 : 3)

Selanjutnya melalui *web browser* dengan menyetikkan URL <http://localhost/phpmyAdmin/> pada *address bar*, maka tampilan phpMyAdmin akan seperti Gambar II.3.



Gambar II.3. Tampilan phpMyAdmin
(Sumber : Anwar, 2016 : 3)

Tampilan phpMyAdmin dibagi menjadi 2, bagian pertama berada pada sebelah kiri berisi daftar *database* yang telah ada pada instalasi MySQL. Lalu pada bagian kedua berada pada sebelah kanan tampilan berisi fitur dari aplikasi phpMyAdmin dan informasi dari *server* MySQL, (Anwar, 2016 : 3).

II.7.2. Membuat *Database, Table* dan *Field*

Berikut ini adalah penjelasan membuat *database, table* dan *field* yaitu:

1. Setelah phpMyTata Usaha tampil pada *web browser*, isikan nama *database* pada kolom “*Create new database*”, kemudian klik tombol *Create*. Jika berhasil maka *database* yang dibuat akan tampil pada daftar *database* dan tampil halaman “*Create Table*”.
2. Halaman berikutnya isikan nama tabel pada kolom *Name*, dan jumlah kolom pada kolom *Number of fields*, kemudian klik tombol *Go*.
3. Halaman berikutnya “*Create Field*” untuk membuat kolom dalam *database*. Sebelum memulai pembuatan *field table*, terdapat beberapa hal yang perlu diperhatikan dalam menentukan tipe data yang digunakan untuk menyimpan data. Tipe data yang umum digunakan adalah *Varchar, Int, Date* dan *Text*. Tipe data *Varchar* digunakan untuk menyimpan bentuk data berupa karakter meliputi angka, huruf dan tanda baca dengan jumlah maksimal 255 karakter. *Int* digunakan untuk menyimpan bentuk data hanya angka dengan *range* antara -2147483648 dan 2147483647. Penyimpanan bentuk data tanggal dengan format tahun-bulan-tanggal (yyyy-mm-dd) menggunakan tipe *Date*. Tipe *Text* digunakan menyimpan bentuk data karakter meliputi angka, huruf dan tanda baca dengan jumlah maksimal karakter tidak terbatas.

Isikan halaman “*Create Field*” berdasarkan tipe data yang di gunakan. Setelah selesai klik tombol *Save*, jika berhasil maka akan tampil halaman “*Tables*”. Dengan tampilnya halaman “*Tables*” maka proses pembuatan *Database, Table* dan *Field* selesai, (Anwar, 2016 : 3).

II.8. Normalisasi

Normalisasi adalah salah satu cara untuk meminimalisir pengulangan data (*data redundancy*), normalisasi akan diperlukan jika ada indikasi bahwa tabel yang kita buat tidak baik (terjadi pengulangan informasi, potensi inkonsistensi data pada operasi pengubahan, tersembunyinya informasi tertentu dan lain sebagainya) dan diperlukan supaya jika tabel-tabel yang didekomposisi kita gabung kembali dapat menghasilkan tabel awal sebelum didekomposisi, sehingga diperoleh tabel yang baik. Hasil dari normalisasi adalah himpunan-himpunan data (tabel-tabel) dalam bentuk normal (*normal form*) (Mulyani, 2016:132).