

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

A. Konsep Dasar Sistem Informasi

Sistem pada dasarnya adalah sekelompok unsur yang erat berhubungan dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.(Mulyadi, 2016). Sistem adalah sekumpulan komponen-komponen atau jaringan kerja dari prosedur-prosedur yang saling berkaitan dan saling bekerja sama membentuk suatu jaringan kerja untuk mencapai sasaran atau tujuan tertentu. (Muslihudin & Oktafianto, 2016).

B. Pengertian Sistem

Informasi adalah data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat. (Krismiaji, 2015).

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. (Hutahaean, 2015)

C. Karakteristik Sistem

Suatu sistem dapat dikatakan sebagai sistem yang baik apabila memiliki karakteristik-karakteristik tertentu. Karakteristik sistem (Hutahaean, 2015) yang dimaksud, yaitu:

1. **Komponen**

Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan sistem (*boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara system dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan luar sistem (*environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung sistem (*interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumberdaya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan sistem (*input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran sistem (*output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti system informasi. Keluaran yang dihasilkan adalah informasi.

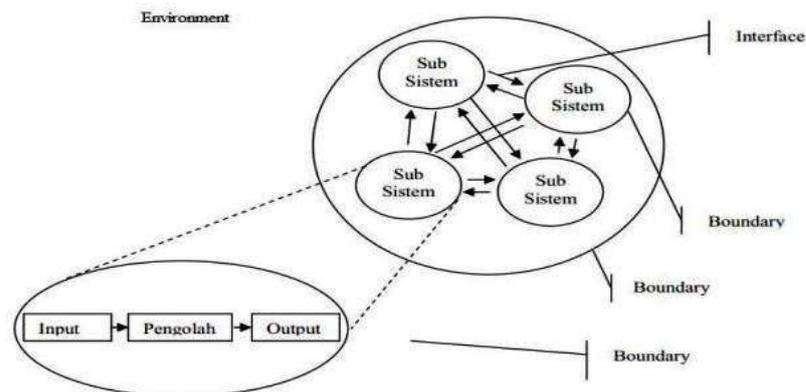
Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolah sistem

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran sistem

Suatu sistem mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.



Sumber: (Hutahaean, 2015)

Gambar II.1.

Karakteristik Dari Suatu Sistem

D. Klasifikasi Sistem

Sistem juga diklasifikasikan dari beberapa sudut pandang sistem. Klasifikasi sistem (Hutahaean, 2015) tersebut terdiri dari:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak merupakan sistem yang berupa pemikiran atau ide tidak tampak secara fisik, misalnya sistem telogi. Sedangkan sistem fisi diartikan

sebagai sistem yang nampak secara fisik sehingga setiap makhluk dapat melihatnya, misalnya sistem komputer.

2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain. Sedangkan sistem buatan manusia merupakan system yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

3. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem komputer, adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem robabilistik merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.

4. Sistem Terbuka dan Sistem Tertutup

Sistem terbuka merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih sepesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia. Sedangkan sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya.

Sistem ini bekerja secara otomatis tanpa danya campur tangan dari pihak luar. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

E. Informasi

Menurut (Hutahaean, 2015) mengemukakan bahwa “informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

Kriteria informasi yang terdiri dari:

1. Relevan

Informasi harus bisa dikatakan relevan apabila informasi yang termuat di dalamnya dapat mempengaruhi keputusan pengguna dengan membantu mereka mengevaluasi peristiwa masa lalu atau masa kini, dan memprediksi masa depan, serta menegaskan atau mengoreksi hasil evaluasi mereka di masa lalu.

2. Andal

Informasi harus bebas dari pengertian yang menyesatkan dan kesalahan material, menyajikan setiap fakta secara jujur, serta dapat diverifikasi.

3. Lengkap

Informasi disajikan selengkap mungkin, yaitu mencakup semua informasi yang dapat mempengaruhi pengambilan keputusan.

4. Tepat waktu

Informasi disajikan tepat waktu sehingga dapat berpengaruh dan berguna dalam pengambilan keputusan.

5. Dapat dipahami

Informasi yang disajikan dalam bentuk serta istilah yang disesuaikan dengan batas pemahaman para pengguna.

6. Dapat diverifikasi

Informasi yang disajikan dapat diuji, dan apabila pengujian dilakukan lebih dari sekali oleh pihak yang berbeda, hasilnya tetap menunjukkan simpulan yang tidak berbeda jauh.

7. Dapat diakses

Informasi yang tersedia pada saat dibutuhkan dan dengan format yang dapat digunakan.

2.2. Teori Pendukung

A. Dana Pensiun

Dana Pensiun sangat penting dalam menggerakkan perekonomian karena selain menjamin kesejahteraan tenaga kerja di masa pensiun, juga membantu perkembangan sektor riil melalui investasi.

Menurut (Saefuloh, Alhusain, Silalahi, Surya, & Wirabrata, 2015) Dana Pensiun adalah sejumlah dana yang disiapkan oleh seseorang/lembaga untuk kepentingan seseorang pada saat ia tidak lagi bekerja. Berdasarkan definisi-definisi tersebut maka dapat dirangkum bahwa Dana Pensiun merupakan dana yang sengaja dihimpun secara khusus dengan tujuan untuk memberikan manfaat kepada karyawan pada saat mencapai usia pensiun, meninggal dunia, atau cacat.

B. UML (*Unified Modeling Language*)

Pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language (UML)*. UML muncul karena adanya kebutuhan

pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML Merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa & Shalahuddin, 2018).

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax/semantik*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya : Grady Booch OOD (*Object Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object Oriented Software Engineering*).

Abstraksi konsep dasar UML terdiri dari *structural classification*, *dynamic behavior* dan *model management*. Abstraksi konsep dasar UML terdiri dari *structural*, *class*, *ification*, *dynamic*, *behavior* dan *model manajemen*. UML mendefinisikan diagram-diagram sebagai berikut:

1. Use Case Diagram

Use case adalah pemodelan untuk kelakuan sistem informasi yang akan dibuat. Kesimpulannya *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi itu (Rosa & Shalahuddin, 2018).

Stereotype adalah sebuah model khusus yang terbatas untuk kondisi tertentu. Untuk menunjukkan *stereotype* digunakan symbol “<<” diawalnya dan ditutup “>>” diakhirnya. <<*extend*>> digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat

tertentu yang dipenuhi. Sedangkan `<<include>>` digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya. Biasanya `<<include>>` digunakan menghindari pengcopian suatu *use case* karena sering dipakai. Sebuah skema *level use case*, *use case* utama ada pada pada *level* „, *sea level*’. *Sea level use case* mewakili interaksi *diskret* diantara *actor* utama dan *system*. *Use case* ini akan mendeliver beberapa nilai ke *actor* utama dan biasanya membutuhkan beberapa menit sampai setengah jam bagi *actor* untuk melakukan. *Use case* yang ada karena di *include* oleh *use case sea level* disebut *fish level*.

2. **Activity Diagram**

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Aktivitas menggambarkan proses yang berjalan, sementara *Use Case* menggambarkan bagaimana *actor* menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, *standart* UML menggunakan segi empat dengan sudut membulat untuk menggambarkan aktivitas (Rosa & Shalahuddin, 2018).

Ilustrasikan proses paralel (*fork and join*) digunakan titik sinkronisasi yang dapat berupa titik, garis *horizontal* atau *vertikal*. *Activity Diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

3. **Class Diagram**

Menurut Whitten dalam (Suendri, 2018) Kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek. *Class* memiliki tiga area pokok yaitu :

- a. Nama, kelas harus mempunyai sebuah nama.
- b. Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.
- c. Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.

4. *Sequence Diagram*

Menurut Havaluddin dalam (Suendri, 2018) Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*.

C. Basis Data

1. *Entity Relationship Diagram*

Menurut Fathansyah dalam (Sukmaidrayana & Sidik, 2017) menyimpulkan bahwa:

Entity Relationship Diagram (Diagram E-R) adalah yang digunakan untuk menggambarkan model Entity Relationship yang berisi komponen-komponen. Himpunan Entitas dan Himpunan Relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempersentasikan seluruh fakta dari dunia nyata yang kita tinjau.

Komponen yang terdapat dalam *Entity Relationship Diagram* adalah sebagai berikut:

1. *Entity* (entitas)

Entitas adalah suatu data yang dapat disimpan dan berguna bagi badan atau perusahaan, dengan kata lain suatu obyek yang dapat dibedakan dengan objek lainnya. Entitas digambarkan dengan kotak persegi panjang.

Terdapat juga Entitas Lemah (*Weak Entity*), yaitu suatu entitas sangat bergantung dengan entitas biasa, dengan kata lain, entitas lemah tidak akan ada apabila tidak

ada entitas biasa. Entitas lemah digambarkan dengan kotak persegi panjang dengan garis ganda.

2. Atribut

Atribut menunjukkan karakteristik dari tiap-tiap entitas. Atribut digambarkan dengan bentuk *oval*.

3. Relasi

Relasi menunjukkan hubungan yang terjadi diantara entitas. Relasi digambarkan dengan bentuk belah ketupat atau *diamond*.

4. Line Connector

Line Connector digambarkan dengan bentuk garis tunggal.

5. Atribut Utama

Atribut utama digambarkan dengan bentuk *oval*, dengan keterangan diberi garis bawah *absolute*.

6. Atribut Pilihan

Atribut Pilihan digambarkan dengan bentuk *oval* dengan keterangan diberi garis bawah putus-putus.

7. Kardinalitas

Kardinalitas merupakan tingkat hubungan yang terjadi diantara entitas di dalam sebuah sistem. Terdapat tiga tingkat hubungan yang terjadi, yaitu:

a. Hubungan Satu pada Satu (*One to One* atau 1:1)

Tingkat hubungan dinyatakan satu pada satu jika satu kejadian pada entitas pertama hanya mempunyai satu hubungan dengan suatu kejadian pada entitas kedua. Demikian juga sebaliknya satu kejadian pada entitas kedua hanya bisa mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

b. Hubungan Satu pada Banyak (*One to Many* atau 1:M)

Tingkat hubungan satu pada banyak (1:M) adalah sama dengan banyak pada satu (M:1), tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua, hanya bisa mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

c. Hubungan Banyak pada Banyak (*Many to Many* atau M:N)

Tingkat hubungan banyak pada banyak (M:N) terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi entitas yang kedua.

2. LRS (*Logical Relationship Structure*)

LRS (Logical Record Structure) merupakan representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas. (Sukmaidrayana & Sidik, 2017). Diagram hubungan entitas atau lebih dikenal dengan sebutan Entity Relationship Diagram merupakan suatu model jaringan yang menggambarkan rancangan atau susunan data store dari sistem pada level yang tinggi. Jadi didalamnya terdapat informasi apa saja yang terkandung didalam data store dan juga hubungan apa yang ada diantara data store. (Sukmaidrayana & Sidik, 2017)

LRS merupakan hasil pemodelan *Entity Relationship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas.

Dalam pembuatan LRS terdapat tiga hal yang dapat mempengaruhi, yaitu:

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one to one*), maka digabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.

2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one to many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many to many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

3. MySQL (*My Structure Query Language*)

MySQL adalah salah satu jenis *database server* yang menggunakan SQL sebagai bahasa dasar untuk mengakses *databasenya*. Dengan menggunakan *script* PHP dan PERL *Software database* ini dapat berfungsi atau berjalan pada semua *platform* sistem operasi yang biasa digunakan (*Windows, Linux, OS/2, berbagai varian Unix*) (Susanti, 2016).

Terdapat beberapa alasan pokok mengapa MySQL sangat populer dikalangan *programmer web*, yaitu:

- a. MySQL tersedia di berbagai *platform*, baik itu *linux* atau *windows* serta juga dalam berbagai varian unik, misalnya MySQL dirilis oleh *Microsoft*.
- b. Fitur-fitur yang dimiliki oleh MySQL sangat banyak di butuhkan dalam aplikasi *web*, Contohnya: Klausur Limit yang berfungsi untuk pengaturan halaman.
- c. MySQL memiliki *overhand* koneksi yang rendah. Karakteristik inilah yang menjadikan MySQL cocok bekerja dengan aplikasi CGI, dimana di setiap *request script* akan melakukan koneksi, mengirimkan satu atau lebih perintah SQL, lalu memutuskan koneksi lagi.

Adapun fungsi koneksi ke *server database* menggunakan pola yang sama yaitu *server, port, user, password*. Fungsi-fungsi tersebut adalah:

a. *Mysql_connect (host,user, password)*

Fungsi ini digunakan untuk membuka koneksi dengan *server* MySQL.

b. *Mysql_select_db (nama_database, nama_koneksi)*

Fungsi ini digunakan untuk memilih *database* yang akan digunakan.

c. *Mysql_query (perintah_query, nama_koneksi)*

Perintah ini digunakan untuk mengirimkan *query* ke *server database* melalui link nama koneksi.

d. *Mysql_fetch_array (hasil_query)*

Fungsi ini digunakan untuk melakukan pembacaan *record* hasil *query* yang dilakukan.

e. *Mysql_fetch_row (hasil_query)*

Fungsi ini mempunyai hampir sama dengan *mysql_fetch_array* tetapi *array* hasil pembacaan data hanya menggunakan indeks *numeric* saja yang dimulai dari 0 untuk kolom pertama sampai n-1 untuk kolom terakhir hasil *query*.

f. *Mysql_result (hasil_query, no_record, nama_field)*

Fungsi ini berguna untuk mengambil langsung nilai hasil *query* pada suatu baris dan kolom tertentu (satu sel) dengan menyebutkan parameter *variable* hasil proses *query*, *no_record*, untuk nomor baris (dimulai dari 0) dan nama field.

g. *Mysql_num_field (hasil_query)*

Fungsi ini berguna untuk mendapatkan jumlah *field* dari hasil *query*.

h. *Mysql_num_rows (hasil_query)*

Fungsi ini berguna untuk mendapatkan jumlah baris dari hasil *query*.

Secara garis besar, *database* MySQL mempunyai 3 macam tipe data, yaitu:

1. Tipe Data *Numeric*

Tipe Data *Numeric* pada *database* MySQL terbagi atas beberapa macam tipe data, yaitu:

a. *Int*

Digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -2.147.483.648 s/d 2.147.483.647. Tipe data ini mempunyai ukuran 4 *byte* (32 *bit*).

Contoh : TOTAL_MAHASISWA INT;

b. *Tinyint*

Digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -128 s/d 127. Tipe data ini mempunyai ukuran 1 *byte* (8 *bit*)

c. *Smallint*

Digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -32.768 s/d 32.767. Tipe data ini mempunyai ukuran 2 *byte* (16 *bit*)

d. *Mediumint*

Digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -8.388.608 s/d 8.388.607. Tipe data ini mempunyai ukuran 3 *byte* (24 *bit*)

e. *Bigint*

Digunakan untuk menyimpan data yang berupa bilangan bulat positif dan negatif dengan jangkauan antara -8.388.608 s/d 8.388.607. Tipe data ini mempunyai ukuran 8 *byte* (64 *bit*)

f. *Float dan Double*

Float digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif presisi tunggal. Tipe data ini mempunyai ukuran 4 *byte* (32 *bit*).

Double Digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif presisi ganda. Tipe data ini mempunyai ukuran 8 *byte* (64 *bit*).

g. *Decimal*

Digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif presisi ganda. Tipe data ini mempunyai ukuran 8 *byte* (64 *bit*).

h. *Real*

Digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif. Tipe data ini mempunyai ukuran 8 *byte* (64 *bit*).

i. *Numeric*

Digunakan untuk menyimpan data yang berupa bilangan pecahan positif dan negatif. Tipe data ini mempunyai ukuran 8 *byte* (64 *bit*).

2. Tipe Data *Date & Time*

Tipe Data *Date & Time* pada *database* MySQL terbagi atas beberapa macam tipe data, yaitu:

a. *Date*

Digunakan untuk meyimpan data tanggal dalam format YY:MM:DD

b. *Datetime*

Digunakan untuk menyimpan data tanggal dan waktu dalam format YY:MM:DD
HH:MM:SS

c. *Time*

Digunakan untuk menyimpan data waktu dalam format HH:MM:SS

d. *Year*

Digunakan untuk menyimpan data tahun.

3. Tipe Data *String*

Tipe Data *String* pada *database* MySQL terbagi atas beberapa macam tipe data, yaitu:

a. *Char*

Digunakan untuk menyimpan data karakter/*string* dengan ukuran tetap. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 255 karakter.

b. *Varchar*

Digunakan untuk menyimpan data karakter/*string* dengan ukuran dinamis. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 255 untuk MySQL versi 4.1. Dan mempunyai jangkauan antara 0 s/d 65.535 untuk MySQL versi 5.0.3

c. *Blob*

BLOB (*Binary Large Object*) adalah tipe data yang digunakan untuk menyimpan data biner seperti *file*, *image*, suara, dll. Tipe data ini mempunyai jangkauan 216-1 *byte*.

d. *Tinyblob*

Digunakan untuk menyimpan data biner seperti *file*, *image*, dan suara. Tipe data ini mempunyai jangkauan 255 *byte*.

e. *Mediumblob*

Digunakan untuk menyimpan data biner seperti *file*, *image*, dan suara. Tipe data ini mempunyai jangkauan 244-1 *byte*.

f. *Longblob*

Digunakan untuk menyimpan data biner seperti *file*, *image*, dan suara. Tipe data ini mempunyai jangkauan 232-1 *byte*

g. *Text*

Digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 65.535 (216-1) karakter.

h. *Tinytext*

Digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 s/d 255 untuk MySQL versi 4.0, dan mempunyai jangkauan antara 0 s/d 65.535 untuk MySQL versi 5.0.3

i. *Mediumtext*

Digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 224-1 karakter

j. *Longtext*

Digunakan untuk menyimpan data *text*. Tipe data ini mempunyai jangkauan antara 0 sampai dengan 232-1 karakter

k. *Enum*

Digunakan untuk menyimpan data enumerasi (kumpulan data)

l. *Set*

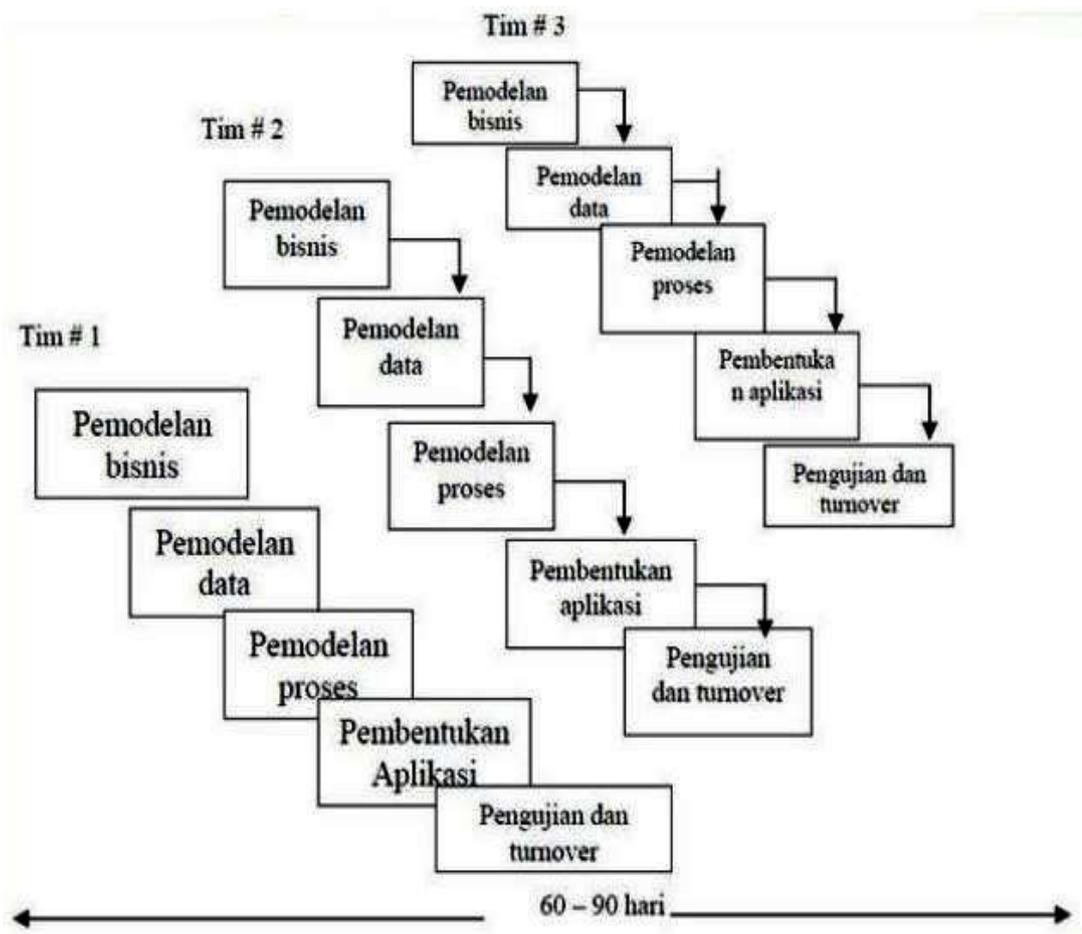
Digunakan untuk menyimpan data himpunan data.

D. Model Pengembangan Perangkat Lunak *RAD (Rapid Application Development)*

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adopsi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak. (Rosa & Shalahuddin, 2018).

Jika kebutuhan perangkat lunak dipahami dengan baik dan lingkup perangkat lunak dibatasi dengan baik sehingga tim dapat menyelesaikan pembuatan perangkat lunak dengan waktu yang pendek. Model RAD membagi tim pengembang menjadi beberapa tim untuk mengerjakan beberapa komponen masing-masing tim pengerjaan dapat dilakukan secara paralel.

Berikut adalah gambar dari model RAD:



Sumber : (Rini & Fatmariyani, 2017)

Gambar II.2.

Model *Rapid Application Development* (RAD)

Adapun tahapan- tahapan berdasarkan metode *Rapid Application Development* (RAD) adalah sebagai berikut (Rini & Fatmariyani, 2017):

1. *Requirements Planning* (Perencanaan Persyaratan)

Tahapan requirements planning bertujuan untuk mengidentifikasi kebutuhan, batasan dan objektifitas dari sistem yang akan dibangun, dengan cara mengumpulkan data dari stakeholder. Aktivitas yang dilakukan dengan melakukan pengamatan langsung dan mengumpulkan data dari buku-buku dan jurnal-jurnal yang menunjang dan relevan. Hasil yang didapatkan berupa mekanisme atau prosedur pengambilan data penelitian dan spesifikasi kebutuhan sistem.

2. *RAD Design Workshop* (Pemodelan)

Tahapan *RAD Design Workshop* bertujuan untuk merancang semua kegiatan dalam arsitektur sistem secara keseluruhan dengan melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya. Aktivitas yang dilakukan dengan melakukan identifikasi pelaku, analisis proses dan kinerja sistem, mengidentifikasi struktur objek dan relasinya, pemodelan interaksi obyek dan behavior, dan mendesain antarmuka. Hasil yang didapatkan berupa pemodelan sistem.

3. Implementasi

Tahap implementasi bertujuan untuk mengimplementasikan metode, program sesuai dengan kebutuhan sistem. Aktivitas yang dilakukan dengan membangun sistem sesuai dengan pemodelan yang dibangun. Hasil yang didapatkan berupa sistem pengajuan pengambilan data penelitian berbasis *website* dengan tahapan-tahapan berdasarkan metode *Rapid Application Development* (RAD).

E. Website

Menurut Rahdian dalam (Surniandari & Gustaman, 2014) Website adalah suatu ruang informasi dimana sumber-sumber daya yang berguna diidentifikasi dengan pengenalan global yang disebut Uniform Resource Locator (URL)“. Secara termologi, pengertian website adalah kumpulan dari halaman-halaman situs atau link yang biasanya terangkum dalam sebuah domain atau subdomain yang tepatnya berada didalam World Wide Web (WWW) di Internet. Sebuah halaman web adalah dokumen yang ditulis dalam format HTML (Hyfer Text Markup Language), yang hampir selalu bisa di akses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser. Halaman-halaman dari sebuah website yang diidentifikasi oleh URL (Uniform Resource Locator) biasa Disebut homepage atau domain name.

Ada bermacam-macam jenis website, sama seperti halnya tempat di dunia nyata yang memiliki fungsinya masing-masing. Berdasarkan pada fungsinya, website terbagi atas hal – hal :

1. *Personal website*

Website-website yang berisi informasi pribadi seseorang.

2. *Commercial website*

Website yang dimiliki oleh sebuah perusahaan yang bersifat bisnis.

3. *Government website*

Website yang dimiliki oleh instansi pemerintah, pendidikan, yang bertujuan memberikan pelayanan kepada pengguna.

4. *Non-profit organization website*

Dimiliki oleh organisasi yang bersifat *non-profit* atau tidak bersifat bisnis.

Berdasarkan sifatnya, ada dua jenis website, yaitu sebagai berikut:

a. *Website Statis*

Website Statis merupakan jenis website yang halamannya statis dan tidak berubah-ubah. *Website statis* biasanya digunakan untuk *web* profil pribadi atau organisasi. Bahasa pemrograman yang digunakan untuk *website* jenis ini adalah HTML dan tidak menggunakan *database*. *Website* jenis ini menampilkan konten halaman yang sama bagi para pengunjungnya. Pemilih *website* sebenarnya dapat mengubah isi *website* tersebut secara periodik, namun perlu proses manual untuk mengubah konten dalam *website* tersebut.

b. *Website Dinamis*

Website Dinamis merupakan jenis *website* yang sering diubah-ubah atau ditambah kontennya. Bahasa pemrograman yang sering digunakan untuk *website* ini yaitu PHP, ASP serta memanfaatkan database MYSQL atau MS SQL. Contoh dari *website* jenis ini yaitu *website* berita seperti detik.com, nytime.com atau blog pribadi.

Aplikasi berbasis *web* adalah sebuah aplikasi yang dapat diakses melalui *internet* atau *intranet*. Banyak dari perusahaan-perusahaan berkembang yang menggunakan aplikasi berbasis *web* dalam merencanakan sumber daya mereka dan untuk mengelola perusahaan mereka yaitu sebagai berikut :

1. *Web Browser*

Web browser menurut (Hastanti, Puji, & Purnama, 2015) *Web Browser* adalah sebuah perangkat lunak atau *software* yang berfungsi untuk menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *server web*.

2. *Web Server*

Web server menurut (Supono & Putratama, 2016) adalah “perangkat lunak *server* yang berfungsi untuk menerima permintaan dalam bentuk situs web melalui HTTP atau HTTPS dari klien itu, yang di kenal sebagai browser web dan

mengirimkan kembali (reaksi) hasil dalam bentuk situs yang biasanya merupakan dokumen HTML”. *web server* yang digunakan dalam sistem adalah XAMPP. Dalam paket XAMPP sudah terdapat *Apache (Webserver)*, *MySQL (database)*, *PHP (server side scripting)*, *Perl*, *FTP server*, *PHP MyAdmin*.(Hastanti, Rulia Puji & Purnama, 2015)

3. *Internet*

Internet menurut (Sukamto & Shalahuddin, 2018) “*Internet* atau *Internetwork* adalah sekumpulan jaringan berbeda yang saling berhubungan bersama sebagai satu kesatuan dengan menggunakan berbagai macam protokol, salah satunya adalah *TCP/IP (Transmission Control Protokol/Internet Protocol)*”. *Internet* dapat diartikan sebagai jaringan komputer yang luas dan besar yang mendunia. Yaitu menghubungkan pemakai komputer dari suatu negara dengan negara lain di seluruh dunia, dimana di dalamnya berbagai sumber daya informasi dari mulai yang statis hingga yang dinamis dan interaktif.