

BAB II

LANDASAN TEORI

2.1. Konsep Dasar

2.1.1. Konsep Dasar Sistem

Di dalam konsep dasar ini, penulis menambahkan beberapa teori dari para ahli yang berkaitan dengan penyusunan tugas akhir ini. Termasuk hal-hal yang berkaitan dengan penulisan tugas akhir, diantaranya pengertian dari konsep dasar sistem dan program.

1. Sistem

“Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.” (Abdussomad, dkk, 2016)

Menurut (Abdussomad, dkk, 2016) mengemukakan bahwa suatu sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu”.

Dari pengertian di atas dapat ditarik kesimpulan bahwa sistem merupakan kumpulan suatu komponen sistem yang saling berhubungan satu dengan yang lain untuk mencapai tujuan suatu kegiatan pokok perusahaan.

2. Karakteristik Sistem

Model umum sebuah sistem adalah *input*, proses dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana, sebab sebuah sistem dapat mempunyai

beberapa masukan dan keluaran. Adapun karakteristik sistem menurut (Tohari, 2014) adalah sebagai berikut:

a. Komponen Sistem (*component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem tersebut dapat berupa suatu sub-sistem atau bagian dari sistem.

b. Batasan Sistem (*boundary*)

Batasan sistem merupakan batasan suatu sistem dengan suatu sistem lainnya atau dengan lingkungan luarnya, batasan sistem ini memungkinkan suatu sistem dipandang satu kesatuan yang tidak dapat dipisahkan. Batasan suatu sistem menunjukkan ruang lingkup dari sistem tersebut

c. Lingkungan Luar Sistem (*environment*)

Merupakan bentuk apapun yang ada di luar sistem yang mempengaruhi operasi sistem tersebut. Di dalam lingkungan luar sistem ini bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

d. Penghubung Sistem (*interface*)

Adalah sebagai media yang menghubungkan sistem dengan subsistem yang lainnya.

e. Masukan Sistem (*input*)

Adalah energi yang dimasukkan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan masukan sinyal (*signal input*).

f. Keluaran Sistem (*output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna keluaran ini merupakan masukan untuk subsistem yang lain.

g. Pengolah Sistem (*process*)

Merupakan suatu sistem yang dapat mempunyai suatu proses yang akan merubah masukan menjadi keluaran. Sebagai contoh sistem akuntansi, sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

h. Sasaran Sistem (*objective*)

Sasaran sistem merupakan suatu sistem yang mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik (dapat diprediksi). Sasaran sangat menentukan kebutuhan akan masukan dan keluaran yang diharapkan, jika suatu sistem tidak memiliki sasaran, maka operasi sistem tidak akan ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuan yang telah direncanakan.

3. Klasifikasi Sistem

Menurut (Jermias, 2016) sistem dapat diklasifikasikan, sebagai berikut :

1. Sistem abstrak (*Abstract system*)

Sistem abstrak adalah gagasan yang tidak tampak secara fisik.

2. Sistem Fisik (*Physical system*)

Sistem fisik adalah sistem dapat dilihat dengan mata dan tampak secara fisik

3. Sistem Alamiah (*Natural system*)

Sistem alamiah adalah terjadi karena proses alam, bukan buatan manusia.

4. Sistem Buatan Manusia (*human made system*)

Sistem buatan manusia adalah sistem yang terjadi melalui rancangan yang dibuat oleh manusia.

5. Sistem tertentu (*Deterministic system*)

Sistem tertentu yaitu sistem operasinya dapat diprediksi .

6. Sistem tak tentu (*Probabilistic system*)

Sistem tak tentu yaitu sistem operasinya tidak dapat diprediksi.

7. Sistem tertutup (*Closed system*)

Sistem tertutup yaitu sistem yang tidak berhubungan dengan lingkungan diluar sistem.

8. Sistem terbuka (*Open system*)

Sistem terbuka yaitu sistem yang menerima *input* dari sub sistem lain dan menghasilkan *output* untuk sub sistem lain

9. Sistem relatif tertutup

Biasanya memiliki masukan dan pengeluaran tertentu sehingga tidak dapat dipengaruhi oleh keadaan diluar sistem.

2.1.2. Konsep Dasar Informasi

Menurut Bodnar dan Hopwood dalam (Setiorini, dkk, 2018) “Informasi adalah data yang berguna yang diolah sedemikian rupa sehingga dapat menjadikan dasar untuk mengambil keputusan yang tepat”.

Menurut (Tohari, 2014) mengemukakan bahwa “Informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan”.

Sedangkan kualitas dari suatu informasi tergantung dari tiga hal menurut Tohari adalah:

1. Akurat artinya informasi harus bebas dari kesalahan-kesalahan.
2. Tepat artinya informasi yang datang pada penerima tidak boleh terlambat.
3. Relevan artinya informasi harus mempunyai manfaat bagi pemakainnya.

2.1.3. Konsep Dasar Akuntansi

“Sistem Akuntansi adalah organisasi formulir, catatan dan laporan yang dikoordinasi sedemikian rupa untuk menyediakan informasi keuangan yang dibutuhkan oleh manajemen guna memudahkan pengelolaan perusahaan”. (Mulyadi, 2016)

Berdasarkan pengertian diatas maka dapat disimpulkan bahwa sistem informasi akuntansi adalah sebuah sistem yang memproses dan mengumpulkan data serta transaksi untuk menghasilkan informasi yang bermanfaat bagi pihak-pihak yang memerlukan.

2.1.4. Sistem Informasi Akuntansi

“Sistem Informasi Akuntansi adalah komponen organisasi yang dirancang untuk mengolah data keuangan menjadi informasi atau laporan keuangan, yang di tujukan kepada pihak internal maupun eksternal perusahaan”. (Tresnawati, dkk, 2017)

“Sistem Informasi Akuntansi adalah suatu sistem yang mengumpulkan mencatat, menyimpan, dan memproses data untuk menghasilkan informasi bagi para pengambil keputusan”. (Kurniawan dan Parparaga, 2014)

Menurut Romney dan Steinbart dalam (Jermias, 2016) menjelaskan bahwa “Sistem informasi akuntansi merupakan sistem yang mengumpulkan mencatat, menyimpan dan memproses data sehingga menghasilkan informasi untuk mengambil keputusan”.

Untuk dapat menghasilkan informasi yang diperlukan oleh para pembuat keputusan, sistem informasi akuntansi harus melaksanakan tugas-tugas sebagai berikut.

1. Mengumpulkan transaksi dan data lain dan memasukkannya kedalam sistem.

2. Memproses data transaksi.
3. Menyimpan data untuk keperluan dimasa mendatang.
4. Menghasilkan informasi yang diperlukan dengan memproduksi laporan, atau memungkinkan para pemakai untuk melihat sendiri data yang tersimpan di komputer.
5. Memngendalikan seluruh proses sedemikian rupa sehingga informasi yang dihasilkan akurat dan dapat dipercaya.

2.1.5. Jurnal

“Jurnal adalah buku harian untuk mencatat semua transaksi secara kronologis yang memuat nama bersama besarnya ke rekening/rekening debit maupun kredit. (Sujarweni, 2016)

2.1.6. Program

Abdul Kadir mengemukakan bahwa yang dimaksud “program adalah kumpulan intruksi yang digunakan untuk mengatur komputer agar melakukan suatu tindakan tertentu”. (Hardiyanto, dkk, 2019)

“Pemrograman terstruktur adalah konsep paradigma atau sudut pandang pemrograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program komputer”. (Sukamto dan Shalahuddin, 2018)

1. Netbeans IDE 8.1

“Java *Netbeans* IDE 8.1 merupakan pengembangan dari BASIC yang dibuat sebagai bahasa pemrograman yang mudah dipelajari dan digunakan”. (Enterprise, 2015). Java *Netbeans* IDE 8.1 memungkinkan proses *Rapid Application Development* (RAD) dari aplikasi antar muka, mengakses *database*, dan membuat kontrol dan objek.

2.1.7. Studi Literatur

Untuk menunjang Laporan Tugas Akhir ini, penulis memaparkan lima penelitian terkait mengenai Sistem Persediaan barang, diantaranya:

1. Pada Jurnal yang berjudul "Rancang Bangun Sistem Informasi Persediaan Barang *Spare Part* Mobil Berbasis Web Menggunakan Metode ABC", Tujuan Penelitian ini untuk membuat *standard operating procedure* menjadi lebih efektif dan pengendalian atas investasi tetap terjaga pada Bengkel Tubagus Ban. Hasil dari penelitian ini menunjukkan jika perlu nya membuat sebuah sistem yang terkomputerisasi agar proses bisnis pada Bengkel Tubagus ban ini dapat menjadi lebih baik. Hasil dari pengendalian persediaan menggunakan metode ABC pada Bengkel Tubagus ban ini menampilkan *spare part* jenis oli golongan A dengan 6 jenis *spare part* dan presentase permintaan sebesar 39,3%, golongan B dengan 8 jenis *spare part* dengan presentase permintaan sebesar 30,8% dan Golongan C yaitu 14 jenis *spare part* dengan presentase permintaan sebesar 29,9. Untuk jenis *tire* yaitu Golongan A dengan 5 jenis *spare part* dan presentasi permintaan sebesar 41%, Golongan B dengan 8 jenis *spare part* dan presentase permintaan sebesar 38,5% dan golongan C yaitu 12 jenis *spare part* dengan presentase permintaan sebesar 20,5%. (Wahyudin, 2019)
2. Pada Jurnal yang berjudul "Sistem Informasi Pengelolaan Persediaan Bahan Baku Menggunakan Model *Waterfall* Pada CV. Ginnozu Cipta Komunikasi". Sistem Pengelolaan Persediaan Bahan Baku merupakan salah satu fungsi manajerial yang sangat penting dalam operasional pada CV Ginnozu Cipta Komunikasi Selain merupakan investasi yang membutuhkan modal yang sangat besar, Sistem Pengelolaan persediaan dapat mempengaruhi pelayanan

terhadap pelanggan dan fungsi produksi, fungsi pemasaran dan fungsi keuangan pada CV Ginnozu Cipta Komunikasi Persediaan bahan baku bertujuan untuk memperlancar kegiatan produksi. Kendala yang terjadi selama ini belum sempurnanya sistem informasi pengelolaan persediaan bahan baku, yaitu dengan menggunakan sistem konvensional, CV Ginnozu Cipta Komunikasi sering kali mengalami masalah terutama dalam kegiatan operasional pengelolaan persediaan bahan baku. Begitu pula untuk informasi laporan dimana pimpinan ingin mengetahui data sisa bahan baku yang tersedia sewaktu-waktu tidak bisa dilakukan secara cepat karena data terlebih dahulu harus direkap dari kartu *stock* sehingga membutuhkan waktu yang lama dan mengalami keterlambatan pembuatan laporan. Dalam menyelesaikan permasalahan diatas dan pentingnya data yang akurat tentang pengelolaan persediaan bahan baku, maka penulis melakukan penelitian analisa dan perancangan sistem informasi pengelolaan bahan baku dengan tujuan membantu CV Ginnozu Cipta Komunikasi dalam perancangan sistem informasi pengelolaan persediaan bahan baku dengan menggunakan komputer sebagai alat bantu pengolahan data. Pada saat pembuatan laporan ke pimpinan apa bila menghendaki sewaktu-waktu bisa langsung dicetak. Diharapkan sistem informasi pengelolaan persediaan bahan baku dapat memberi informasi yang cepat untuk kelancaran proses pengelolaan persediaan bahan baku. Secara sederhana konsep dasar sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. (Solecha, 2019)

3. Pada Jurnal yang berjudul "Rancang Bangun Sistem Informasi Pengolahan Data Persediaan Barang Berbasis Dekstop Dengan Model *Waterfall*". Sistem persediaan barang adalah suatu sistem untuk mengelola persediaan barang di gudang. Sistem persediaan barang kini sudah banyak digunakan oleh perusahaan-perusahaan berkembang pada umumnya, terutama dalam hal pengolahan data barang. Kampung Kemang adalah usaha dari perusahaan PT. Sakura Yasa Prima yang bergerak dibidang penjualan minuman di kota Jakarta dimana aktivitas kinerja yang baik dan efektif haruslah dijaga dan ditingkatkan. Seperti pengelolaan, pencarian, alur keluar masuknya barang, dan laporan data barang ketika dibutuhkan perusahaan. Berdasarkan permasalahan tersebut, perusahaan memiliki keinginan untuk memanfaatkan teknologi informasi secara baik melalui pembangunan sistem persediaan barang, dikarenakan sistem kegiatan pengolah data barang di gudang saat ini belum dirasa maksimal oleh perusahaan. Sistem *inventory* menggunakan metode FIFO (*First In First Out*) untuk mengatur alur keluar masuknya barang di gudang. Metode aliran yang digunakan adalah UML (*Unified Modeling Language*). Bahasa pemrograman menggunakan PHP, dan *database* MySQL. Sistem dibangun dengan *user friendly*, sehingga sistem dilengkapi dengan *Interface* yang mmudah dipahami dan dibuat semenarik mungkin. (Nawang dan Kurniawati, 2017)
4. Pada Jurnal yang berjudul "Rancang Bangun Sistem Informasi Persediaan Barang Pada PT. Bumi Permata Medika". Dalam mengerjakan laporan *inventory* alat kesehatan PT. Bumi Permata Medika masih di kerjakan secara konvensional, sehingga menimbulkan beberapa kesulitan dan kesalahan dalam pembuatan laporan. Banyaknya stok, nama barang yang harus dicatat,

menyebabkan terjadinya kerangkapan data dan membutuhkan waktu yang relatif lebih lama untuk mengerjakannya, maka harus dikerjakan dengan secara teliti. Kurangnya informasi yang akurat menyebabkan perusahaan tidak mengetahui batas minimum penyimpanan *inventory* alat kesehatan untuk periode selanjutnya. Sebagai solusi untuk memecahkan masalah pada PT. Bumi Permata Medika dalam penyediaan alat medis, penulis merancang program sistem informasi inventori untuk memudahkan dalam mengelola persediaan alat kesehatan. Metode pengembangan sistem yang digunakan dalam perancangan sistem informasi ini menggunakan metode *system development life cycle* (SDLC). (Safudin, dkk, 2018)

2.2. Peralatan Pendukung

Di dalam rancangan sebuah sistem diperlukan suatu peralatan yang dapat mendukung terciptanya sebuah rancangan. Peralatan pendukung (*Tools System*) merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol, lambang, diagram yang menunjukkan secara tepat arti dan fungsinya.

2.2.1. Metode Perancangan Program

“Model air terjun sering disebut juga dengan model sekuensial linier (*sequence linier*) atau alur hidup klasik”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*).

Berikut ini langkah-langkah metode *waterfall* dalam penelitian ini menurut (Sukamto dan Shalahuddin, 2018):

1. Analisa Kebutuhan

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang focus terhadap desain pembuatan program perangkat lunak.

3. Pembuatan Kode Program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program computer sesuai dengan desain yang telah dibuat pada tahap desain. Kode program dibuat menggunakan *software* Java *Netbeans* IDE 8.1.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah di uji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Pengujian dalam web ini menggunakan teknik *blackbox testing*.

5. Pendukung (*Support*) atau pemeliharaan (*Maintenance*)

Tidak menuntut kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk

perubahan perangkat lunak, yang sudah ada tapi tidak untuk membuat perangkat lunak baru.

2.2.2. Basis Data (*Database*)

“Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. (Sukamto dan Shalahuddin, 2018)

“Basis data dapat dipahami sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan bersama-sama dalam suatu media, tanpa mengatap satu sama lain atau tidak perlu satu kerangkapan data”. (Nurrahman, dkk, 2018)

1. MySQL

“SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS”. (Sukamto dan Shalahuddin, 2018)

“SQL merupakan singkatan dari *Structured Query Language*. Bahasa ini didasarkan pada Bahasa Inggris dan digunakan dalam database MySQL, Oracle, dan Microsoft SQL Server”. (Enterprise, 2015)

Menurut Taufik dalam (Fachrurrozy, 2018) “MySQL merupakan *software* yang tergolong *database server* dan bersifat *Open Source*”.

Kelebihan dari MySQL adalah gratis, handal, selalu di-*update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah.

Saat ini MySQL banyak digunakan berbagai kalangan untuk melakukan penyimpanan dan pengolahan data karena beberapa keuntungan diantaranya:

a. Fleksibel

MySQL dapat digunakan untuk mengembangkan aplikasi *desktop* aplikasi web dengan menggunakan teknologi yang bervariasi.

b. Performa tinggi

MySQL memiliki mesin *query* dengan performa tinggi, dengan demikian proses transaksional dapat dilakukan dengan sangat cepat.

c. Lintas Platform

MySQL dapat digunakan pada platform atau dengan sistem informasi yang beragam.

d. Gratis

MySQL dapat digunakan secara gratis, sehingga kalangan dapat menggunakannya.

e. Proteksi Data Handal

MySQL menyediakan mekanisme yang *powerfull* untuk keamanan yaitu dengan menyediakan fasilitas manajemen *user*, enkripsi data, dan lain sebagainya.

f. Komunitas Luas

MySQL sudah memiliki banyak penggunaan sehingga mysql sudah memiliki banyak komunitas yang dapat memudahkan kita untuk berbagi solusi untuk penanganan masalah yang timbul.

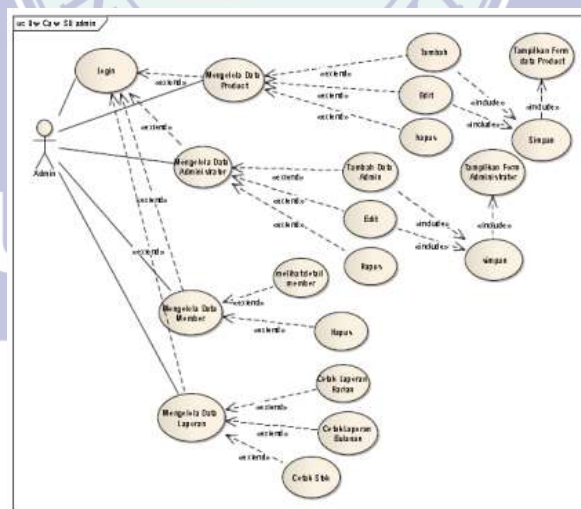
2. XAMPP

Menurut Kristania dalam (Fachrurrozy, 2018) “*Xampp* merupakan sebuah aplikasi perangkat lunak pemrograman dan *database* yang di dalamnya terdapat berbagai macam aplikasi pemrograman seperti : *Apache HTTP server*, *MySQL*, *database*, Bahasa pemrograman *PHP* dan *perl*.”

“UML (*Unified Modeling Language*) adalah bahasa pemodelan dalam sistem atau perangkat lunak yang berparadigma berorientasi objek”. (Abdussomad, dkk, 2016)

“UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement* (kebutuhan), membuat analisis dan *desain*, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”. (Sukamto dan Shalahuddin, 2018)

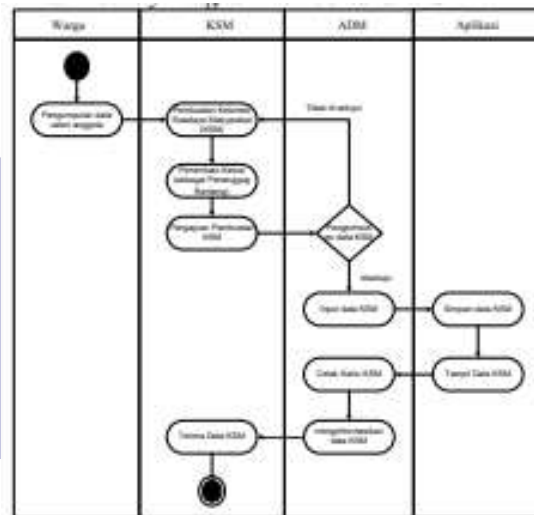
“Use case atau diagram use case merupakan pemodelan atau kelakuan (*behavior*) sistem informasi yang akan dibuat”. (Sukanto dan Shalahuddin, 2018)



Gambar II.1
Use Case Diagram

2. Activity Diagram

“*Activity Diagram* digunakan untuk model perilaku dalam independen proses bisnis suatu objek. Dalam banyak hal, *Activity Diagram* dapat dipandang sebagai diagram 17 aliran data yang canggih yang dapat digunakan dalam hubungannya dengan analisis terstruktur”. (Abdussomad, dkk, 2016)



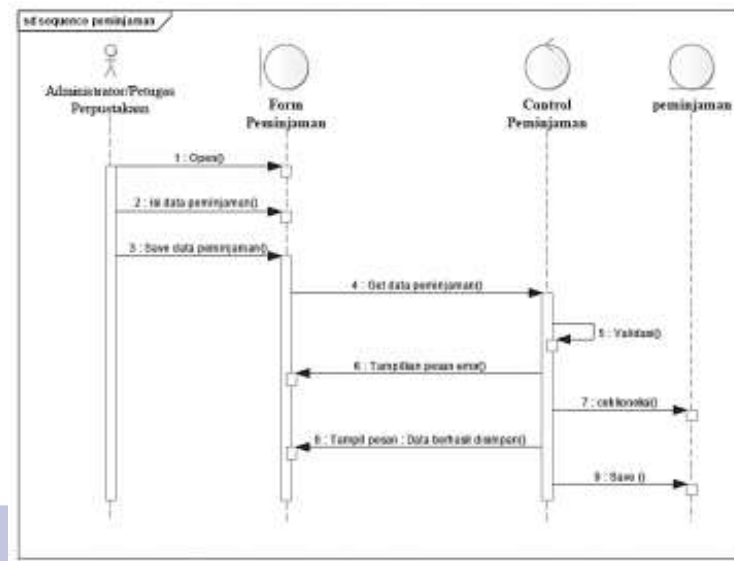
Sumber: (Abdussomad, 2018)

Gambar II.2
Diagram Activity

“Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. (Sukamto dan Shalahuddin, 2018)

3. Sequence Diagram

“*Diagram Sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”. (Sukamto dan Shalahuddin, 2018)



Sumber: (Sukamto dan Shalahuddin, 2018)

Gambar II.3
Sequence Diagram

4. Deployment Diagram

“Diagram deployment atau deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi”. (Sukamto dan Shalahuddin, 2018)

2.2.4. Entity Relationship Diagram (ERD)

“Entity Relationship Diagram (ERD) merupakan pemodelan basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional”. (Sukamto dan Shalahuddin, 2018).

“Sebuah diagram ER/ER_D tersusun atas tiga komponen, yaitu entitas, atribut, dan kerelasian antar entitas. Secara garis besar, entitas merupakan objek dasar yang terlibat dalam sistem. Atribut berperan sebagai penjelas entitas, sedangkan kerelasian menunjukkan hubungan yang terjadi diantara dua entitas” (Nurrahman, dkk, 2018).

“*Entity Relation Diagram* adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas”. (Abdussomad, dkk, 2016).

Pada dasarnya ada tiga simbol yang digunakan sebagai berikut (Abdussomad, dkk, 2016):

1. Entitas digambarkan dengan kotak persegi panjang dan digunakan untuk menunjukkan sekumpulan orang, tempat, objek atau konsep dan sebagainya yang menunjukkan dimana data dicatat atau disimpan.
2. Hubungan atau Relasi digambarkan dengan kotak berbentuk *diamond* atau belah ketupat dengan garis yang menghubungkan ke entitas yang terkait. Maka *relationship* diberi nama dengan kata kerja. Hubungan atau relasi menunjukkan abstraksi dari sekumpulan hubungan yang mengaitkan antara entitas yang berbeda.
3. Atribut digambarkan dengan bentuk elips. Atribut menunjukkan karakteristik dari tiap entitas atau sesuatu yang menjelaskan entitas atau hubungan. Sehingga atribut dikatakan elemen dari entitas dan relasi. Dari setiap atribut entitas terdapat satu atribut yang dijadikan sebagai kunci (*key*). Beberapa jenis kunci tersebut antara lain : *Primary key*, *Candidate key*, *Composite key*, *Secondary key*, *Alternate key* dan *Foreign key*.
4. Tingkat Hubungan (*Cardinality*) *Entity Relation Diagram* (ERD) juga menunjukkan tingkat hubungan yang terjadi.

Fathansyah mengemukakan bahwa “ERD merupakan Model *Entity Relationship* yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari “Dunia Nyata” yang kita tinjau, dapat

digambarkan dengan sistematis dengan menggunakan Diagram *Entity Relationship*". (Hardiyanto, dkk, 2019)

1. Notasi ERD

Ada sejumlah konvensi mengenai notasi ERD. Notasi klasik sering digunakan untuk model konseptual. Berbagai notasi lain juga digunakan untuk menggambarkan secara logis dan fisik dari suatu basis data. Salah satunya adalah IDEFIX. Notasi-notasi simbolik yang digunakan dalam *Entity Relationship Diagram* sebagai berikut:

a. Entitas (*Entity*)

Entity adalah objek yang dapat dibedakan dalam dunia nyata. *Entity* diberi nama dengan kata benda, lokasi dan kejadian. Sedangkan *entity set* adalah kumpulan dari *entity* yang sejenis. *Entity set* terdiri dari:

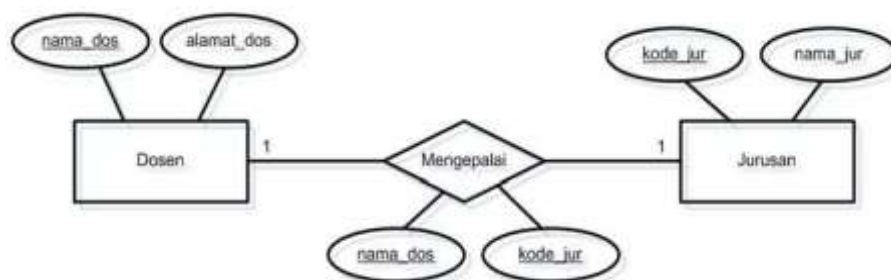
- 1) *Strong entity set* yaitu *entity set* yang satu atau lebih atributnya digunakan oleh *entity set* lain sebagai *key*. Digambarkan dengan empat persegi panjang. Misal: E adalah sebuah *entity set* dengan atribut-atribut $a_1, a_2, \dots, a_n$, maka *entity set* tersebut direpresentasikan dalam bentuk tabel E yang terdiri dari n kolom, di mana setiap kolom berkaitan dengan atribut-atributnya. Entitas yang mandiri, yang keberadaannya tidak bergantung pada keberadaan entitas yang lainnya. Instansiasi entitas kuat selalu memiliki karakteristik yang unik disebut *identifier* (sebuah atribut tunggal atau gabungan atribut-atribut yang secara unik dapat digunakan untuk membedakannya dari entitas kuat yang lain).
- 2) *Weak Entity Set*, *entity set* yang bergantung terhadap *strong entity set*. Digambarkan dengan empat persegi panjang bertumpuk. Misal: A adalah *weak entity set* dari atribut-atribut $a_1, a_2, \dots, a_r$ dan B adalah *strong entity set* dengan atribut-atribut $b_1, b_2, \dots, b_s$, di mana b_1 adalah

atribut *primary key*, maka *weak entity set* direpresentasikan berupa tabel A, dengan atribut-atribut $\{b_1\}$ u $\{a_1, a_2, \dots, a_r\}$, entitas yang keberadaanya sangat bergantung pada keberadaan entitas yang lainnya. Entitas lemah tidak memiliki arti apa-apa dan tidak dikehendaki kehadirannya dalam diagram ER tanpa kehadiran entitas di mana mereka bergantung. Entitas di mana entitas lemah bergantung dinamakan *identifying owner*. Entitas lemah tidak memiliki *identifier* sendiri. Secara umum, dalam diagram ER entitas lemah memiliki atribut yang berperan sebagai *partial identifier* (*identifier* yang berfungsi secara bagian).

b. Relasi (*Relation*)

Relation adalah hubungan yang terjadi antara satu atau lebih *entity*. Simbol dari relasi yaitu belah ketupat. Macam-macam relasi antar tabel, yaitu:

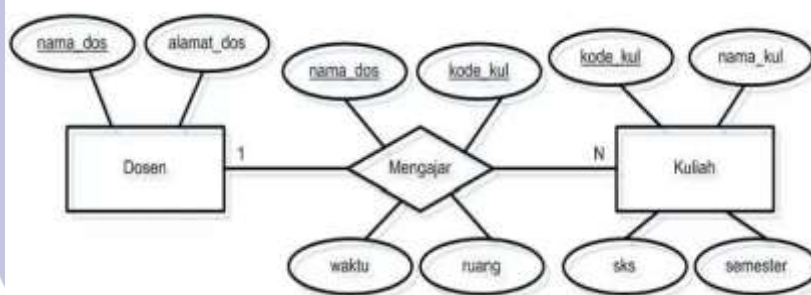
- 1) *One-to-one*. Jika dua tabel berelasi *one-to-one* artinya setiap *record* di entitas pertama hanya akan berhubungan dengan satu *record* di entitas kedua begitu pula sebaliknya. Contoh pada relasi tabel dosen dan tabel jurusan. Satu *record* dosen hanya berhubungan dengan satu *record* jurusan begitu pula sebaliknya. Entitas 3 merupakan atribut yang unik di entitas.



Sumber: (Fathansyah, 2015)

Gambar II.4.
Contoh One-to-one

- 2) *One-to-many*. Misalkan terdapat relasi antara tabel dosen dan tabel kuliah dengan nama relasi “mengajar” dan relasinya *one-to-many*. Artinya satu *record* pada tabel dosen boleh berelasi (mengajar) dengan banyak *record* pada tabel kuliah. Namun satu *record* pada tabel kuliah hanya boleh berelasi dengan satu *record* saja pada tabel dosen.

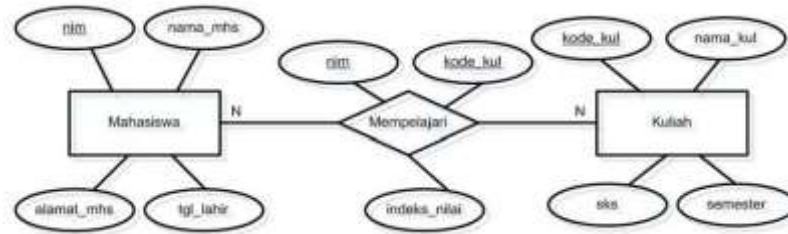


Sumber: (Fathansyah, 2015)

Gambar II.5.
Contoh One-to-many

- 3) *Many-to-many*. Jika tabel satu berelasi dengan tabel dua dengan relasi *many-to-many* artinya ada banyak *record* di entitas satu dan entitas dua yang saling berhubungan satu sama lain. Contoh pada relasi *many-to-*

many antara tabel mahasiswa dan tabel kuliah. Satu *record* mahasiswa bisa berhubungan dengan banyak *record* kuliah, begitu pula sebaliknya.



Sumber: (Fathansyah, 2015)

Gambar II.6.

Contoh *Many-to-many*

c. Atribut (*Attribute*)

Atribut adalah karakteristik dari *entity* atau *relationship*, yang menyediakan penjelasan detail tentang *entity* atau *relationship* tersebut. Jenis-jenis atribut diantaranya:

- 1) Atribut sederhana atau *Simple Attribute* adalah atribut yang tidak dapat dibagi-bagi menjadi atribut yang lebih mendasar.

Contoh: atribut harga dari *entity* barang.

- 2) Atribut komposit atau *Composite Attribute* adalah atribut yang terdiri dari beberapa atribut yang lebih mendasar.

Contoh: *Entity* mahasiswa memiliki atribut nama yang terdiri dari nama depan (*first name*), nama tengah (*middle name*) dan nama belakang (*last name*).

- 3) *Single Valued Attribute* dan *Multi Valued Attribute*

Atribut Berharga Tunggal (*Single Valued Attribute*) adalah atribut yang hanya mempunyai satu harga untuk suatu entitas tertentu.

Contoh: atribut umur.

- 4) Atribut Berharga Banyak (*Multi Valued Attribute*) adalah atribut yang dapat terdiri dari sekumpulan harga untuk suatu entitas tertentu.

Contoh: atribut hobi.

- 5) *Derived Attribute* (Atribut Turunan)

Atribut turunan adalah suatu atribut yang dihasilkan dari atribut lain.

Contoh: atribut umur yang dapat dihasilkan dari atribut *tgl_lahir*.

- 6) *Key Attribute* (Atribut Kunci)

Satu atau beberapa atribut yang mempunyai nilai unik sehingga dapat digunakan untuk membedakan data pada suatu baris atau *record* dengan baris lain pada suatu entitas.

d. Derajat Relasi atau Kardinalitas

Menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.

- 1) *One to One* (1:1)

Tingkat hubungan dinyatakan *one to one* jika suatu kejadian pada *entity* pertama hanya mempunyai satu hubungan dengan satu kejadian pada entitas kedua. Demikian juga sebaliknya satu kejadian pada *entity* yang pertama.

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

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

3) *Many to Many* (M:M)

Tingkat hubungan *many to many* terjadi jika tiap kejadian pada sebuah entitas mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari entitas yang pertama maupun dilihat dari sisi entitas yang kedua.

2.2.5. *Logical Record Structure* (LRS)

Kursini mengungkapkan bahwa” Model relasional adalah kumpulan tabel-tabel untuk merepresentasikan data dan relasi antar data-data”. (Abdussomad, 2018)

“LRS merupakan sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola atau aturan pemodelan tertentu dalam kaitannya dengan konversi LRS, maka perubahan yang terjadi adalah mengikuti aturan-aturan tertentu”. (Puspitasari, 2016)

“*Logical Record Structure* (LRS) merupakan hasil dari pemodelan *entity relationship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antarentitas”. (Junaidi, 2016)

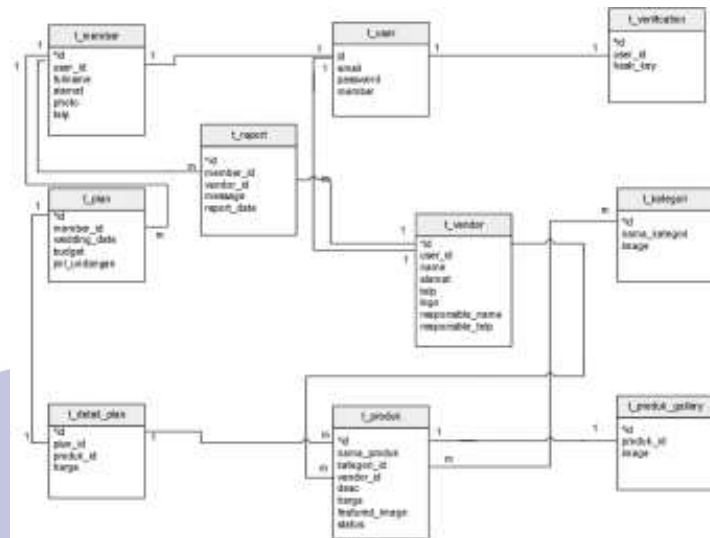
“*Logical Record Structure* (LRS) merupakan transpormasi dari *Entity Relationship Diagram* (ERD), *Logical Record Structure* (LRS) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas”. (Fridayanthie & Mahdiati, 2016)

Menentukan kardinalitas, jumlah tabel, dan *Foreign key* (fk).

Berikut adalah cara membentuk skema database atau LRS (*Logical Record Structure*) berdasarkan *Entitas Relationship Diagram*:

1. Jika relasinya satu-ke-satu, maka *foreign key* diletakan pada salah satu dari dua entitas yang ada atau menyatukan kedua entitas tersebut.
2. Jika relasinya satu-ke-banyak, maka *foreign key* diletakan pada entitas *many*.

3. Jika relasinya banyak-ke-banyak, maka dibuat “*file konektor*” yang berisi dua *foreign key* yang berasal dari kedua entitas.



Sumber : (Fridayanthie & Mahdiati, 2016)

Gambar II.7
Logical Relationship Structure

2.2.6. Spesifikasi File

Menurut (Candra Wijaya, 2015) spesifikasi *file* menjelaskan tentang *file* atau tabel yang terbentuk dari transformasi ERD (dan atau *file-file* penunjang untuk web).

2.2.7. User Interface

“*User Interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah user interface dapat di implementasikan oleh kelas lain. Sebuah kelas dapat mengimplemantasikan lebih dari satu antarmuka dimana kelas ini akan mendeklarasikan metode pada antarmuka yang dibutuhkan kelas itu sekaligus mendefinisikan isinya pada kode program kelas itu. Metode pada antarmuka yang di implementasikan harus sama persis dengan yang ada di antarmuka. Antarmuka pengguna atau *user interface* biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas, mengakses antarmukanya”. (Sukamto dan Shalahuddin, 2018)

2.2.8. *Code Generation*

Code generation adalah proses pembuatan *coding* atau pengkodean merupakan penterjemahan desain kedalam bahasa yang bisa dikenali oleh komputer yang akan menterjemahkan permintaan *user* ke sebuah *software*". (Maranti, 2018)

2.2.9. *Blackbox Testing*

"*Blackbox Testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan". (Sukamto dan Shalahuddin, 2018)

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat menurut (Sukamto dan Shalahuddin, 2018), adalah:

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.