

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Menurut **Fatmawati Isnaini**, 2017, yang berjudul “ **Perancangan Sistem Informasi Akuntansi Penyusutan Aktiva Tetap Menggunakan Metode Garis Lurus Pada Kopkar Bina Khatulistiwa**” Penelitian ini bertujuan untuk perancangan sistem informasi akuntansi penyusutan aktiva tetap metode garis lurus pada KOPKAR Bina Khatulistiwa, maka dapat ditarik simpulan dengan menggunakan aplikasi terkomputerisasi, karyawan bagian pembukuan dapat mengolah data penyusutan aktiva tetap yang akan menghasilkan laporan penyusutan yang dibutuhkan bagi pengguna. Selain itu, data penyusutan aktiva tetap dilaporkan oleh koordinator dalam RUPS untuk menilai penyusutan aktiva tetap sebagai oposional dan aset perusahaan yang mengeluarkan biaya.

Menurut Fransiskus Zoromi, 2017, yang berjudul “ **Perancangan Sistem Pengelolaan Data Aktiva Tetap dan Penyusutan Nilai Aset dengan Metode Straight Line (Studi Kasus di STMIK Amik Riau)**” Penelitian ini bertujuan untuk pengelolaan data aset dan perhitungan nilai penyusutan aset maka dapat membantu bagian rumah tangga di STMIK Amik Riau dalam menghitung total nilai aset yang ada. Dengan adanya sistem aplikasi pengelolaan data aset dan perhitungan nilai penyusutan aset ini lebih mudah dalam pembuatan laporan

sehingga dapat dilakukan dengan baik dan cepat oleh bagian rumah tangga di STMIK Amik Riau, Dengan adanya sistem aplikasi ini bagian staf rumah tangga / administrator lebih mudah dalam melihat laporan tanpa harus dibatasi oleh ruang dan waktu.

Menurut Agustin, 2017, yang berjudul “ **Pengelolaan Data Aktiva Tetap Dan Penyusutan Nilai Aset Dengan Metode *Straight Line***” Penelitian ini bertujuan untuk penyusutan sebuah aset dapat dihitung dengan cepat. Dengan adanya sistem, pengolahan data tentang aset ini lebih mudah dalam pembuatan laporan sehingga dapat dilakukan dengan baik dan cepat serta dengan adanya sistem aplikasi ini bagian staf rumah tangga / administrator lebih mudah dalam melihat laporan tanpa harus dibatasi oleh ruang dan waktu.

Menurut Arif Maulana Yusuf, 2021, yang berjudul “Sistem Informasi perhitungan Penyusutan Aktiva Tetap Menggunakan Metode Garis Lurus Berbasis VB.NET Pada CV Ginanjar Sejahteramandiri Karawang” Penelitian ini bertujuan untuk perhitungan penyusutan aktiva tetap menggunakan metode garis lurus untuk memudahkan perusahaan dalam membuat laporan penyusutan aktiva tetap.

Menurut **Eni Endaryati**, 2020, yang berjudul “**Perancangan Sistem Informasi Akuntansi Aset Tetap Dengan Metode Garis Lurus Sebagai Perhitungan Penyusutan Aset Tetap**” Penelitian ini bertujuan untuk menghasilkan aplikasi sistem informasi akuntansi penyusutan aset tetap dengan metode garis lurus dapat digunakan dan membantu dalam mengelola aset tetap di

Koperasi Semarang dan memberikan informasi yang lengkap dan akurat mengenai nilai aset setiap periode.

Berdasarkan analisa dari penelitian sebelumnya dimana pada penelitian yang dilakukan oleh penulis dalam penerapan di aplikasi dapat secara mudah melakukan penyusutan secara otomatis sehingga lebih memudahkan pengguna dalam melakukan penyusutan.

II.2. Uraian Teoritis

II.2.1. Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau. Adapun Karakteristik dari sistem adalah menyelesaikan suatu sasaran tertentu

Berdasarkan definisi tersebut, dapat disimpulkan bahwa pengertian sistem yaitu merupakan gabungan dari berbagai elemen yang bekerja sama untuk mencapai suatu target atau tujuan.

Menurut Mc Leod “Sistem adalah sekelompok elemen-elemen yang terntegritas dengan tujuan yang sama untuk mencapai tujuan.” (Yunahar Heriyanto : 2018) :

Adapun beberapa karakteristik sistem menurut H.A.Rusdiana DKK karakteristik sistem

a. Komponen (*components*)

Komponen sistem adalah segala sesuatu yang menjadi bagian penyusun sistem.

b. Batas (bondary)

Batas sistem diperlukan untuk membedakan satu sistem dengan sistem yang lain.

c. Lingkungan (*environoments*)

d. Lingkungan sistem adalah segala sesuatu yang berbeda di luar sistem lingkungan sistem yang dapat menguntungkan ataupun merugikan.

e. Penghubung/antarmuka(*interface*)

Penghubung/antar muka merupakan sarana memungkinkan setiap komponen sistem, yaitu segala sesuatu yang bertugas menjabatani hubungan antar komponen dalam sistem.

f. Masukan (*input*)

Masukan merupakan komponen sistem, yaitu segala sesuatuyang perlu dimasukan kedalam sistem sebagai bahan yang akan diolah lebih lanjut lagi untuk menghasilkan keluaran (output) yang berguna.

g. Pengolahan (*processing*)

Pengolahan merupakan komponen sistem yang mempunyai peran utamamengolah masukan agar menghasilkan output yang berguna bagi para pemakainya.

h. Keluaran (*output*)

Keluaran merupakan komponen sistem yang berupa berbagai macam bentuk keluaran yang dihasikan bentuk kluaran yang dihasilkan oleh komponen pengolahan.

i. Sasaran (objektif) dan tujuan (*goal*)

Setiap komponen dalam sistem perlu juga agar cepat bekerja sesuai dengan agar mampu mencapai sasaran dan tujuan sistem.

j. Kendali (control)

Setiap komponen dalam sistem perlu dijaga agar tetap bekerja sesuai dengan peran dan fungsinya masing-masing.

k. Umpan balik (feed back)

Umpan balik diperlukan oleh bagian kendali (control) sistem untuk mengecek terjadinya penyimpanan proses dalam sistem dan mengembalikannya pada kondisi normal. (Otto Fajariantono, 2017).

II.2.2. Informasi

Menurut Deni Darmawan mengungkapkan “Informasi merupakan hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengolahan tersebut bisa menjadi informasi, hasil pengolahan data yang tidak memberikan makna atau arti serta tidak bermanfaat bagi seseorang bukanlah merupakan informasi bagi orang tersebut. Berdasarkan definisi tersebut, dapat disimpulkan bahwa informasi merupakan hasil dari pengolahan data atau data yang sudah di proses.”

Menurut Davis didalam buku Abdul Kadir mengungkapkan “Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang”. (Yunahar Heriyanto : 2018)

II.2.3. Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung fungsi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan pihak luar tertentu. (Yeyi Gusla Nengsih : 2020).

II.2.3.1 Komponen Sistem Informasi

Sistem informasi memiliki beberapa komponen, antara lain :

1. Blok Masukan (*Input Block*), data yang masuk ke dalam sistem informasi.
2. Blok Model (*Model Block*), blok yang akan memanipulasi data input.
3. Blok Keluaran (*Output Block*), keluaran yang merupakan informasi yang berkualitas.
4. Blok Teknologi (*Technology Block*),. Blok teknologi terdiri dari *user (brainware)*, perangkat lunak (*software*), dan perangkat keras (*hardware*).
5. Basis Data (*Database Block*), basis data merupakan tempat data dikumpulkan yang tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. (Muhamad Alda : 2019)

II.2.4. Depresiasi (Penyusutan)

Depresiasi (Penyusutan) adalah alokasi secara periodik dan sistematis dari harga perolehan aset selama periode-periode berbeda yang memperoleh manfaat dari penggunaan aset bersangkutan. Faktor-faktor yang mempengaruhi beban penyusutan adalah:

1. Nilai Perolehan Aset (*asset cost*), mencakup seluruh pengeluaran yang terkait dengan perolehannya dan persiapan sampai aset dapat digunakan
2. Nilai Residu/Nilai Sisa (*residual or salvage value*), merupakan estimasi nilai realisasi pada saat aset tidak dipakai lagi.
3. Umur Ekonomis (*economic life*), dapat diartikan sebagai suatu periode atau umur fisik dimana perusahaan dapat dimanfaatkan aset tetapnya.
4. Pola Pemakaian (*pattern of use*), pola pemakaian ini seringkali diabaikan dalam menghitung besarnya beban penyusutan periodek mengingat sulitnya dalam mengidentifikasi pola pemakaian dimaksud.

Beban penyusutan adalah pengakuan atas penggunaan manfaat potensial dari suatu asset. Menurut Hery, Sifat beban penyusutan secara konsep tidak berbeda dengan beban yang mengakui pemanfaatan atas premi asuransi ataupun sewa yang dibayar dimuka selama periode berjalan. Beban penyusutan merupakan beban yang tidak memerlukan pengeluaran uang kas (*non cash outlay expense*). (Dian Indah Sari : 2018).

II.2.5. Garis Lurus

Dalam metode *Garis Lurus* (*Garis Lurus*) lebih melihat aspek waktu daripada aspek kegunaan. Metode ini paling banyak diterapkan oleh perusahaanperusahaan karena paling mudah diaplikasikan dalam akuntansi. Dalam metode penyusutan *Garis Lurus*, beban penyusutan untuk tiap tahun nilainya sama besar dan tidak dipengaruhi dengan hasil atau output yang diproduksi. Metode *Garis Lurus* mempertimbangkan penyusutan sebagai fungsi dari waktu, bukan fungsi dari penggunaan. Metode ini telah digunakan secara luas

dalam prakteknya disebabkan kemudahannya dan secara koseptual merupakan prosedur yang paling sesuai untuk menghitung penyusutan aset. Rumus metode *Garis Lurus* ditunjukkan pada rumus berikut : (Iqbal Hanafi, 2018)

$$\text{Penyusutan per tahun} = \frac{\text{Harga Perolehan} - \text{Nilai Residu}}{\text{Estimasi Manfaat (dalam tahun)}} \dots\dots\dots(1)$$

II.2.6. SQL Server

SQL Server merupakan suatu *Relational Database Management Systems* (RDBMS) yang digunakan untuk menyimpan data. Data yang disimpan pada *database* bisa dalam skala kecil maupun besar. Selain itu, penyajiannya merupakan penyajian pada level fisik karena kita akan menyimpan langsung data pada *database* dengan kondisi yang sebenarnya, yaitu disimpan pada tabel apa, kolom mana, dan menggunakan tipe data saat penyimpanan. (Benardo, 2015).

Database merupakan suatu tempat untuk menyimpan data. Pada sebuah *database* bisa terdapat satu atau lebih tabel dan *query*. Operasi yang biasanya dilakukan *database* berhubungan erat dengan pengaksesan tabel atau *query* Pada *SQL Server 2008* terdapat fitur-fitur yang dapat mengembangkan performa dari *database* tersebut. Beberapa fitur tersebut, yaitu (Benardo, 2015) :

- 1) *Date Data Type*: Digunakan untuk menyimpan data tanggal saja sehingga akan menghemat *space* pada *server*.
- 2) *Data Compression*: Digunakan untuk melakukan compress data sehingga ukuran data yang disimpan dalam hal *space hardisk* akan lebih kecil.
- 3) *Sparse Column*: Digunakan untuk menyimpan data yang memiliki lebih banyak data NULL dengan lebih efisien.

- 4) *Row Constructor*: Digunakan untuk melakukan insert beberapa data sekaligus dengan satu perintah INSERT.
- 5) *Table-Valued Parameter*: Digunakan untuk melakukan parsing array pada bahasa pemrograman, dimana satu variable diberikan data-data yang akan diproses setelahnya.

II.2.7. Visual Studio 2010

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, *Software Development Kit* (SDK), *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. (Herpendi, 2016).

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*). (Herpendi, 2016).

Visual Studio sebelumnya versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19

November 2007, yang ditujukan untuk *platform* Microsoft *.NET Framework 3.5*. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform *.NET Framework 2.0* dan *3.0*. Visual Studio 2003. ditujukan untuk *.NET Framework 1.1*, dan Visual Studio 2002 ditujukan untuk *.NET Framework 1.0*. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio *.NET*, karena memang membutuhkan Microsoft *.NET Framework*. Sementara itu, sebelum muncul Visual Studio *.NET*, terdapat Microsoft Visual Studio 6.0 (VS1998). (Herpendi, 2016).

II.2.8. Database

Basis data (*database*) merupakan kumpulan berkas (*file*) atau data yang saling berkaitan (mempunyai relasi yang ditunjukkan dengan kunci dari tiap berkas), dimana penyimpanannya disusun secara sistematis di dalam komputer untuk dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) sehingga dapat menghasilkan informasi. *Database* biasanya dipakai untuk mengelola berkas-berkas penting dalam lingkungan perusahaan, instansi atau organisasi. Berkas- berkas pada *database* pada suatu badan usaha, instansi atau organisasi disusun atau diurutkan secara rapih dan terstruktur, sehingga dapat dengan mudah diambil sewaktu-waktu untuk ditampilkan dalam bentuk laporan.

Untuk memasukkan dan mengambil data ke dan dari *database*, diperlukan perangkat lunak yang dikenal dengan istilah sistem manaje men basis data (*database management system*)

(DBMS). Perangkat lunak DBMS merupakan sistem yang memungkinkan pengguna dari basis data (*database user*) untuk memelihara, mengontrol dan

mengakses data secara lebih praktis dan efisien. DBMS ini menjadi lapisan yang menjadi jembatan penghubung antara basis data dengan program aplikasi dan memastikan bahwa basis data tetap terorganisasi secara konsisten dan dapat diakses dengan mudah.

Manfaat menggunakan *database* adalah *sharing data resources*, sehingga data pada satu *database* tadi bisa digunakan bersama-sama dan penggunaanya dapat terkontrol dengan baik dari pusat. *Database* terpusat dapat mempunyai efek positif dalam pengurangan biaya perangkat karena tiap divisi atau bidang dalam suatu perusahaan dapat mengakses *database* dari jaringan yang terhubung dengan *server* pusat. Keunggulan lain adalah *database* juga memberikan kemudahan dan kecepatan dalam mengakses dan memperoleh data.

Berdasarkan jenisnya, *database* dikelompokkan menjadi 2 bagian, yaitu *database flat – file* dan *database relasional*. *Database flat – file* umumnya digunakan pada data berukuran kecil dan dapat dirubah dengan mudah. *Database* jenis ini menggunakan table tunggal yang kolom-kolomnya merepresentasikan seluruh parameter. Sementara itu *database relasional* mempunyai struktur yang lebih logis terkait cara penyimpanan.

Relational database, merupakan penggabungan beberapa tabel dengan berbagai metode untuk dapat berkerjasama. Hubungan-hubungan antar tabel data dapat dibandingkan, disatukan, dan ditampilkan dalam formform *database*. Basis data relasional umumnya menggunakan sekumpulan tabel dua dimensi dimana masing-masing tabel tersebut tersusun atas baris (*tupel*) dan kolom (*atribut*). Hubungan antara dua atau lebih tabel dibuat dengan menggunakan *key* (atribut

kunci) yaitu *primary key* pada salah satu tabel dan *foreign key* di tabel yang lain. Basis data relasional saat ini biasa dijadikan pilihan yang utama karena beberapa keunggulannya. Beberapa contoh perangkat lunak basis data relasional diantaranya oracle, Microsoft SQL Server, MariaDB, dan Mysql. (Antonius Irianto Sukowati : 2017)

II.2.9. UML (*Unified Modelling Language*)

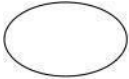
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut (Sinta Maria : 2021) :

II.2.9.1. Use Case Diagram

Diagram use case 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. Berikut ini adalah simbol-simbol diagram use case, seperti yang terlihat pada tabel II.1 :

Tabel II.1. Simbol *Use Case Diagram*

Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukaran pesan antara unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>Use Case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mendefinisikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem biasa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> tetapi tidak memiliki <i>control</i> terhadap <i>use case</i>.
	Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidentifikasi bila aktor berinteraksi secara pasif dengan sistem

	<i>Include</i> , merupakan didalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat dipenuhi.

(Sumber : Sinta Maria : 2021)

II.2.9.2. 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. Berikut adalah simbol- simbol class diagram, seperti terlihat pada tabel II.2. berikut ini.

Tabel II.2.Simbol Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0...*	Boleh tidak ada atau 1 atau lebih
1...*	1 atau lebih
0..1	Boleh tidak ada maksimal 1
n..n	Batasan antara Contoh 2..4 mempunyai arti minimal 2 maksimum 4

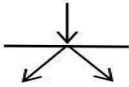
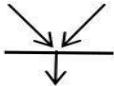
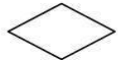
(Sumber : Sinta Maria : 2021)

II.2.9.3. Activity Diagram

Diagram aktivitas atau *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. Berikut adalah simbol-simbol *activity diagram*, seperti terlihat pada tabel II.3 :

Tabel II.3. Activity Diagram

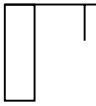
Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End Point</i> , akhir aktifitas
	<i>Activities</i> menggambarkan suatu proses atau kegiatan bisnis.
	<i>Fork</i> (Percabangan) digunakan untuk menunjukan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	<i>Join</i> (Penggabungan) atau rake, digunakan untuk menunjukan adanya dekomposisi.
	<i>Decision points</i> menggambarkan pilihan untuk pengambilan keputusan, true, false
	<i>Swimlane</i> pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

(Sumber : Sinta Maria : 2021)

II.2.9.4. 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 terlibat dalam sebuah use case beserta metode- metode yang dimiliki kelas yang diinstansiasi menjadi objek itu”. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case (Rosa dan Shalahuddin : 2018). Berikut adalah simbol-simbol activity diagram, seperti terlihat pada tabel II.4 berikut ini

Tabel II.4. Simbol Sequence Diagram

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari system yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal system dan menjadi landasan untuk menyusun basis data
	<i>Boundary Class</i> berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem. Seperti tampilan <i>formentry</i> dan <i>form cetak</i> .
	<i>Control Class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar kelas.
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

 	<i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
----------------	---

(Sumber : Sinta Maria : 2021)