

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Pada penelitian terdahulu dengan judul Sistem Informasi Penyewaan Alat-alat Berat (Studi Kasus CV. Karya Bersama). Pada penelitian ini, penulis membuat suatu aplikasi penyewaan *online* yang berbasis *web* sehingga dapat digunakan pelanggan untuk menyewa alat-alat berat dan menghemat waktu serta biaya dalam melakukan transaksi antara perusahaan dan pelanggan. Dengan menggunakan bahasa pemrograman *PHP*, *CodeIgniter* sebagai *framework*-nya, dan *MySQL* sebagai *database*-nya. Metode pengembangannya menggunakan *SDLC* dengan Model *Waterfall* (Dwi Nafsiah, 2012)

Dan pada penelitian selanjutnya dengan judul Sistem Informasi Persewaan Alat Berat Dan Kendaraan Transportasi Pada CV. Barokah Abadi Semarang. Pada peneliti ini masih menyajikan informasi persewaan alat berat dan kendaraan transportasi secara cepat, tepat dan efisien akan sangat dibutuhkan. Dalam penelitian ini, peneliti menggunakan metode dalam penyusunan sistem Jogiyanto yang dimulai dari tahapan perencanaan, analisis sistem, desain sistem dan implementasi sistem (Ayu Kusuma Wardhani, 2012)

II.2 Landasan Teori

II.2.1 Sistem

Sistem merupakan suatu kumpulan dari komponen-komponen yang membentuk satu kesatuan. Sebuah organisasi dan sistem informasi adalah sistem fisik dan sosial yang ditata sedemikian rupa untuk mencapai tujuan tertentu. Seorang manajer yang mengorganisasikan pekerjaan unitnya adalah orang yang menciptakan sesuatu sistem pengembangan *personal computer* (PC) menciptakan suatu sistem yang terdiri dari perangkat keras dan lunak PC ini menjadi subsistem dari suatu perangkat konferensi elektronik. Sistem fisik dan sosial adalah sistem yang abstrak (*abstract system*) dari konsep dan ide. Contoh sistem yang abstrak ialah pengembangan daur hidup atau pengembangan sistem perangkat lunak (Tyoso ; 2016 : 1).

II.2.2. Informasi

Pengertian informasi masih bersifat kontradiktif dan belum mempunyai ciri-ciri khusus. Maka para pakar teori atau ilmu pengetahuan informasi belum dapat membuat satu definisi yang jelas dan lengkap mengenai apa yang disebut informasi.

Menurut para pakar informasi, informasi meliputi aspek abstrak dan khusus. Informasi dipandang sebagai ilmu pengetahuan tertulis atau yang di sampaikan secara lisan dan sebagai hasil dari data yang diolah, biasanya di olah secara formal. Ada pula yang menyebut informasi sebagai sarana pengurang ketidak pastian. Pengertian ini sama dengan arti informasi dalam istilah ekonomi,

yakni informasi adalah penghalau ketidakpastian, misalnya di dalam pasar, *consumer preference*, dan harga (Tyoso ; 2016 : 31).

II.2.3 Sistem Informasi Akuntansi

Sistem informasi akuntansi (SIA) merupakan gabungan dari tiga kata, yaitu sistem, informasi dan akuntansi. Sistem adalah serangkaian prosedur formal untuk mencapai suatu tujuan. Informasi adalah sekumpulan data yang telah diproses dan didistribusikan ke pemakai (*users*). Sistem informasi akuntansi terdiri dari tiga subsistem utama, yaitu:

- a. Sistem pemrosesan transaksi (*Transaction processing systems*),
- b. Sistem pelaporan buku besar/keuangan (*General ledger/financial reporting systems*) dan
- c. Sistem pengendalian manajemen (David Efendi ; 2013 : 73)

II.3. Basis AkruaI (*Accrual Basis*)

Accrual basis merupakan salah satu konsep yang sangat penting dalam akuntansi dimana Pencatatan akrual basis adalah teknik pencatatan ketika transaksi terjadi dimana uang benar-benar diterima atau dikeluarkan. Dengan kata lain akuntansi akrual basis adalah basis akuntansi yang mengakui pengaruh transaksi dan peristiwa lainnya pada saat kas atau setara kas diterima atau dibayar yang digunakan untuk pengakuan pendapatan, belanja dan pembiayaan. AkruaI basis akan mencatat kegiatan keuangan saat kas atau uang telah diterima misalkan perusahaan menjual produknya akan tetapi uang pembayaran belum diterima

maka pencatatan pendapatan penjualan produk tersebut tidak dilakukan, jika kas telah diterima maka transaksi tersebut baru akan dicatat seperti halnya dengan “dasar akrual” hal ini berlaku untuk semua transaksi yang dilakukan, jika menggunakan dasar akrual maka penjualan produk perusahaan yang dilakukan secara kredit akan menambah piutang dagang sehingga berpengaruh pada besarnya piutang dagang (H.Rahmansyah Ritonga; 2013).

II.4. *Visual Basic*

VB.NET adalah salah satu bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa *VB.NET* ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada platform .NET yang diluncurkan tahun 2002 dan untuk menjembatani programmer *Visual Basic*. Bahasa *VB.NET* secara teknis mengadopsi sintak bahasa *Visual Basic*. Konsistensi *API (Application Programming Interface)* membuat bahasa *VB.NET* menjadi pilihan dalam membuat kode program diatas *platform Windows*. Fitur baru bahasa *VB.NET* dibandingkan *Visual Basic* bahwa bahasa *VB.NET* mendukung *object-oriented* dan juga *dynamics programming*. Ini menambah daftar kemudahan untuk belajar bahasa *VB.NET* (Agus Kurniawan ; 2013 : 10).

II.5. *Basis Data (Database)*

Secara sederhana *database* (basis data/pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat

disimpan dalam media pengingat yang disebut *hard disk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database*. (Agustinus Mujilan ; 2012 : 23)

II.6. *SQL Server*

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang *database*. *SQL Server* adalah sebuah *Database Management System (DBMS)* yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahuluan seperti *IBM* dan *Oracle*. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data (Wahana Komputer ; 2013 : 2).

II.7. Kamus Data

Kamus data umumnya berguna pada pengembangan model sistem dan dapat digunakan untuk menangani semua informasi dari semua tipe model sistem. Kamus data sederhananya adalah daftar alfabetis dari nama-nama termasuk pada berbagai model sistem. Seperti namanya, kamus harus mencakup deskripsi yang berhubungan dengan entitas bernama tersebut dan, jika nama itu merepresentasikan objek komposit, mungkin saja ada deskripsi mengenai komposisinya. Informasi lain seperti tanggal pembuatan, pembuatnya dan

representasi entitas juga dapat dimasukkan, tergantung pada tipe model yang sedang dikembangkan. Keuntungan penggunaan kamus data adalah :

1. Kamus data merupakan mekanisme untuk manajemen nama. Banyak orang yang harus menciptakan nama untuk entitas dan relasi ketika mengembangkan model sistem yang besar. Nama-nama ini harus dipakai secara konsisten dan tidak boleh bentrok. Perangkat lunak kamus data dapat memeriksa keunikan nama dan memberitahu analisis persyaratan sekiranya terjadi duplikasi nama.
2. Kamus data berfungsi sebagai tempat penyimpanan informasi organisasional yang dapat menghubungkan analisis, desain, implementasi dan evolusi. Sementara sistem dikembangkan, informasi diambil untuk memberitahu perkembangan. Informasi baru ditambahkan pada sistem. Semua informasi mengenai entitas berada pada satu tempat (Ian Sommerville ; 2012 : 151).
Adapun tabel entri kamus data dapat dilihat pada Tabel II.1.

Tabel II.1. Entri Kamus Data

Nama	Keterangan	Tipe	Tanggal
Has-labels	Relasi 1 N antara entitas bertipe node atau link dan entitas bertipe label	Relasi	5.10.1998
Label	Berisi informasi terstruktur atau tidak terstruktur mengenai node atau link. Label dipresentasikan oleh ikon (yang bisa berupa kotak transparan) dan teks yang berhubungan	Entitas	8.10.1998
Link	Relasi 1 : 1 antara entitas desain yang dipresentasikan sebagai node. Link diketikkan dan dapat diberi nama	Relasi	8.10.1998
Name (Label)	Setiap label memiliki nama yang mengidentifikasi tipe label. Nama harus unik dalam set tipe label yang dipakai pada desain	Atribut	8.10.1998
Name (Node)	Setiap node harus memiliki nama yang unik di dalam sebuah desain. Panjang nama bisa mencapai 64 karakter.	Atribut	15.10.1998

(Sumber : Ian Sommerville ; 2012 : 151)

II.8. Normalisasi

Normalisasi adalah suatu proses untuk membuat data yang tidak normal menjadi data yang normal. Bentuk data yang tidak normal / data mentah biasa disebut juga *unnormalized form*. Masing – masing level normalisasi mempunyai aturan tersendiri.

1. *First Normal Form* (1NF)

Suatu tabel dikatakan dalam keadaan *first normal form* (1NF) jika :

- a. Tidak ada perulangan record data dalam tabel.
- b. Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group dan array.
- c. Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

2. *Second Normal Form* (2NF)

Suatu tabel dikatakan dalam keadaan *Second Normal Form* (2NF) jika tabel tersebut sudah dalam keadaan *First Normal Form* (1NF) dan jika semua atribut yang bukan kunci tabel, baik *primary key* maupun *foreign key* tergantung pada semua kunci dalam tabel.

3. *Third Normal Form* (3NF)

Suatu tabel dikatakan dalam keadaan *third normal form* (3NF) jika tabel tersebut sudah dalam keadaan *second normal form* (2NF) dan jika tidak terdapat ketergantungan yang transitif. Artinya, data-data yang mungkin diisi berulang-ulang dapat dibuat sebuah tabel baru.

4. *Boyce-Codd Normal Form (BCNF)*

Tabel dikatakan dalam keadaan *boyce-codd normal form (BCNF)* jika tabel tersebut dalam keadaan *third normal form (3NF)* dan setiap determinan adalah kunci kandidat.

5. *Fourth Normal Form (4NF)*

Suatu tabel dikatakan dalam keadaan *fourth normal form (4NF)* jika tabel tersebut dalam keadaan *boyce-codd normal form (BCNF)* dan jika tidak terdapat ketergantungan nilai ganda.

6. *Fiveth Normal Form (5NF)*

Tabel dikatakan dalam keadaan *Fiveth Normal Form (5NF)* jika tabel tersebut dalam keadaan *fourth normal form (4NF)* dan jika setiap ketergantungan dalam join ada pada tabel sudah konsekuen dengan kunci kandidat pada tabel tersebut (Ema Utami ; 2012 : 73-76).

II.9. *Unified Modeling Language (UML)*

Menurut Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

Unified Modeling Language (UML) merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *Unified Modeling Language (UML)* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa


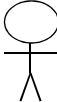


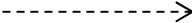
pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *Unified Modeling Language* (UML) adalah sebagai berikut :

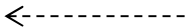
1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada Tabel

II.2.

Tabel II.2. Simbol *Use Case*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control 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 mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <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.



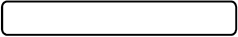
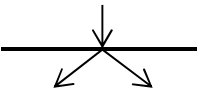
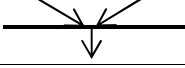
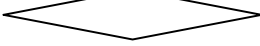

	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.
---	---

(Sumber : Gata, 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel II.3.

Tabel II.3. Simbol *Activity Diagram*

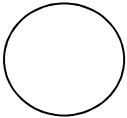
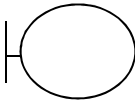
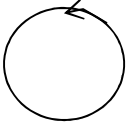
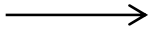
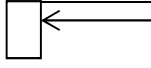


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau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

(Sumber : Gata, 2013 : 6)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada Tabel II.4.

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem 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</i> cetak.
	<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 <i>class</i>
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gata, 2013 : 7)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan

multiplicity atau kardinaliti. Adapun tabel *multiplicity Class Diagram* dapat dilihat pada Tabel II.5.

Tabel II.5. *Multiplicity 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 : Gata, 2013 : 9)