

## BAB II

### LANDASAN TEORI

#### 2.1 Sistem Informasi Piutang

Sistem dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan komponen yang saling berhubungan satu dengan yang lainnya, membentuk satu kesatuan untuk mencapai tujuan tertentu. (Nurjoko, 2018).

Menurut Anggi, zulkarnaini, ruki dan emilia (2022) Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategis dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan.

Piutang menurut A. Syafi'i (2015:104) mengungkapkan mengenai piutang sebagai berikut : "Piutang menunjukkan adanya klaim perusahaan kepada pihak (perusahaan) lain akibat kejadian di waktu sebelumnya dalam bentuk uang, barang, jasa atau dalam bentuk aktiva non kas lainnya yang harus dilakukan penagihan (collect) pada tanggal jatuh temponya".

A. Syafi'i (2015:104) membagi piutang menjadi tiga jenis, yaitu :

1. Piutang dagang (*account receivable*) atau A/R, merupakan klaim kepada pihak lain yang timbul karena penjualan kepadanya barang dagangan atau jasa yang dilakukan secara kredit. Piutang dagang merupakan suatu tagihan kepada pembeli yang tidak disertai dengan dokumen yang mempunyai kekuatan hukum yang dapat memaksa Debitor untuk melakukan pembayaran pada tanggal jatuh temponya.
2. Piutang wesel (*notes receivable*), merupakan tagihan atau piutang yang dinyatakan secara tertulis dalam bentuk surat perintah membayar (*wesel*) atau dalam bentuk surat kesanggupan membayar (*promes*).

3. Piutang lain-lain (*non account and receivable*), merupakan klaim kepada pihak ketiga yang timbul karena kejadian di masa lalu yang tidak terkait dengan kejadian penyerahan barang atau jasa yang diperdagangkan atau diproduksi. Yang termasuk dalam klasifikasi piutang lain - lain antara lain : Pinjaman tunai, beban dibayar di muka (*differal expenses*). (Achmad Sidik dan Iis Widiawati, 2016).

## 2.2 Tinjauan Pustaka

Riset terdahulu merupakan hasil penelitian terdahulu terkait dengan penelitian ini sebagai tinjauan pustaka untuk mendukung penelitian. Dapat dilihat pada tabel 2.1.

**Tabel 2.1 Riset Terdahulu tentang sistem informasi akuntansi piutang**

No.	Peneliti	Judul	Hasil riset	Objek
1.	Sidik & Widiawati, 2016	Rancangan Sistem Informasi Pengelolaan Piutang Dengan Metodologi Berorientasi Objek	Usulan prosedur baru, rancangan menggunakan StarUML. Rancangan Case Diagram terdiri dari 3 agen atau aktor yaitu sales accounting, finance manager, finance accounting. Case atau proses yang diciptakan adalah Login, Home, file, AR management, Report, Logout Rancangan tampilan : Form Login, Form Halaman utama, Form Halaman Menu File, Halaman menu AR Manajemen, Halaman Report. Pengembangan sistem menggunakan bahasa pemrograman PHP dan database MYSQL. Sistem dapat menghasilkan laporan outstanding piutang, jadwal pembayaran piutang, laporan uang masuk dan laporan saldo piutang.	PT Shinta Woo Sung

**Tabel 2.1 Riset Terdahulu tentang sistem informasi akuntansi piutang  
(lanjutan)**

No.	Peneliti	Judul	Hasil riset	Objek
2.	Arum & Nugroho, 2017	Perancangan Sistem Informasi Akuntansi Penerimaan Kas Berbasis Web	Penelitian menggunakan pendekatan kualitatif dan merupakan penelitian research and development dengan metode Rapid Application Development. Pembahasan menghasilkan prosedur, dokumen dan catatan yang dipakai (nota, faktur, catatan penjualan, catatan piutang, catatan persediaan), fungsi terkait (fungsi penjualan, fungsi persediaan, fungsi pencatatan). Teknik analisis adalah bussiness modelling, data modelling (menggunakan ERD), Proses Modelling(menggunakan DFD)	Batik Pramanca, Gamping, Sleman, Daerah Istimewa Yogyakarta
3.	Rada & Maulana, 2019	Rancang Bangun Aplikasi Akuntansi Perhitungan Piutang Dagang	Perancangan menggunakan metode Waterfall terhadap sistem yang berjalan, hasil perancangan berupa form login, menu utama, form data debitur, form input transaksi penjualan kredit, form transaksi penagihan, laporan piutang.	CV Anugrah Jaya Sentosa Bandar Lampung

**Tabel 2.1 Riset Terdahulu tentang sistem informasi akuntansi piutang  
(lanjutan)**

No.	Peneliti	Judul	Hasil riset	Objek
4.	Rahma Dwi Kusumaratri & Purwanto,	PERANCANGAN SISTEM INFORMASI AKUNTANSI PIUTANG PADA CV MENANG SENTOSA	Prancangan menggunakan metode waterfall. Hasil perancangan adalah a Form Daftar Toko, Form Transaksi Penjualan, Form Transaksi Retur, Form Transaksi Pembayaran.	CV MENANG SENTOSA
5.	Retnosari & Ayunda Putri Nilasari (2021).	SISTEM INFORMASI PENGENDALIAN PIUTANG TAK TERTAGIH PADA PT HISHENG LUGGAGE ACCESSORY	Penelitian ini menggunakan metode penelitian dan pengembangan Research and Development (R&D). Pengertian Research and Development adalah salah satu metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut. Hasil perancangan menghasilkan form login, dashboard admin & pimpinan, data pelunasan, data penjualan, data laporan.	PT.HISHENG LUGGAGE ACCESSORY

### 2.3 Pengertian Website

Website application merupakan aplikasi yang menggunakan user interface berbasis teknologi web, dikodekan menggunakan bahasa pemrograman didukung teknologi penjelajah web. Bahasa pemrograman tersebut diantaranya php, javascript, asp, perl. Aplikasi ini memiliki banyak kelebihan, dapat diakses

dengan berbagai macam perangkat seperti personal computer, smartphone, tablet dan laptop dengan syarat memiliki program peramban web.

Website adalah keseluruhan halaman halaman yang terdapat dari sebuah domain yang mengandung informasi atau yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, dan video). Domain adalah nama unik yang dimiliki oleh sebuah institusi sehingga bisa diakses melalui internet. (Bagus Prihadi, Ruki Rizal, Hendra Kurniawan & Melda Agarina, 2020).

Menurut Pressman, keuntungan menggunakan aplikasi berbasis web ini diantaranya,

- a. Kemudahan akses data yang tidak dibatasi tempat dan waktu.
- b. Pemakai mendapatkan kemudahan dari sisi penggunaan.
- c. Tidak memerlukan platform khusus dalam proses akses.
- d. Kemudahan dalam kegiatan *maintenance*, karena kegiatan tersebut hanya dilakukan pada sistem server. Sedangkan dari sisi pengguna atau *klien* tidak memerlukan *maintenance* khusus. Ketika terjadi perubahan atau tambahan fasilitas modul sistem, cukup dilakukan pada unit server. (Subianto, 2020).

## **2.4 Bahasa Pemrograman Pendukung**

### **2.4.1 PHP (Hypertext Pre Processor)**

Menurut Subianto, (2020) PHP singkatan dari Hypertext Preprocessor adalah Suatu bahasa pemrograman berupa skrip untuk membuat halaman web dinamis, meskipun pengguna lain juga dapat menggunakannya (Betha Sidik dan Husni I. Pohan, 2010: 12). Pemrograman berbasis web biasanya menggunakan bahasa pemrograman PHP dan berpasangan dengan aplikasi manajemen database MySQL.

### **2.4.2 PHP My Admin**

Menurut Muhammad Susilo, Rezki Kurniati & Kasmawi, (2018) 1PHPMYAdmin adalah perangkat lunak yang bebas ditulis dalam bahasa pemrograman PHP yang

digunakan untuk menangani administrasi MYSQL melalui Jejaring jagat Jembar (World Wide Web). PHPMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengolah basis data, tabel-tabel, bidang (fields), relasi (relation), indeks, pengguna (users), perjanjian (permissions), dan lain-lain

### **2.4.3 XAMPP**

XAMPP merupakan sebuah software yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolah data MySQL dikomputer local. XAMPP berperan sebagai server web pada komputer anda. XAMPP juga dapat disebut sebuah CPanel server virtual, yang dapat membantu anda melakukan preview sehingga dapat memodifikasi website tanpa harus online atau terakses dengan internet. (Halimah & Bobby Bachry, 2018)

Menurut Muhammad Susilo, Rezki Kurniati & Kasmawi, (2018) XAMPP adalah aplikasi yang berfungsi sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain : Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X empat sistem operasi, yang meliputi Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU (General Public License), merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis.

### **2.4.4 Cascading Style Sheets (CSS)**

Menurut Muhammad Susilo, Rezki Kurniati & Kasmawi, (2018) CSS adalah singkatan dari Cascading Style-Sheet, yaitu sebuah pengembangan atas kode HTML yang sudah ada sebelumnya. Dengan CSS, bisa menentukan sebuah struktur dasar halaman web secara lebih mudah dan cepat, serta irit size.

### **2.4.5 MYSQL dan Basis Data**

Menurut Subianto, (2020) MySQL merupakan *Relational Database Management System* (RDBMS) yaitu hubungan antar tabel yang berisi data-data pada suatu database. Basis data MySQL terdiri dari setiap tabel. Setiap tabel memiliki kolom, baris, dan record untuk menyimpan data. Tabel-tabel ini dihubungkan oleh sebuah

hubungan. Ketika seorang pengguna ingin menampilkan informasi dalam database, hubungan tersebut dapat menggabungkan data dari beberapa tabel. Penggunaan MySQL biasanya digabungkan dengan aplikasi PHP, karena penggunaan kedua program ini telah terbukti kehandalannya dalam memproses permintaan data.

Menurut lanjut Ochi Marsela dan Laili Ambarwati, (2015) Basis data atau *database* merupakan kumpulan data yang saling berhubungan satu dengan yang lain tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulsinya. Data perlu disimpan didalam database untuk keperluan penyediaan informasi lebih.

Terdapat banyak istilah yang digunakan dalam database seperti file, record, field, entity dan attribute. Definisi lain dari masing-masing istilah tersebut sebagai berikut:

- a) File Kumpulan *Record-record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda-beda data volume, Kristanto (2002).
- b) *Record* Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entity secara lengkap, Kristanto (2002).
- c) *Field-Field* adalah item informasi diantara item informasi lain yang membentuk suatu record Tribisono (2001).
- d) Attribute Setiap entity mempunyai attribute atau sebutan mewakili entity.

### Sistem Basis Data

Dalam sebuah sistem basis data, secara lengkap akan terdapat komponen-komponen utama sebagai berikut:

- a) Perangkat Keras (*Hardware*)
- b) System Operasi (*Operating System*)
- c) Basis Data (*Database*)
- d) Perankat Lunak (*Software*)
- e) Pemakai (*User*)

File didalam pemrosesan aplikasi dapat dikategorikan kedalam beberapa tipe diantara sebagai berikut:

a. File Induk (*Master File*) Didalam aplikasi, file ini merupakan file yang penting. File ini tetap harus ada selama sistem hidup. File induk ini dapat dibedakan menjadi dua jenis yaitu :

b. File Induk Acuan (*Reference Master File*) Adalah file induk yang nilai recordnya relative statis.

c. File Induk Dinamik (*Dynamic Master file*)

.File transaksi File ini digunakan untuk merekam data hasil dari suatu transaksi. d. File laporan (*report file*) File laporan ini merupakan file yang berupa informasi yang akan ditampilkan.

d. File pelindung (*backup file*) File ini merupakan salinan dari file-file yang masih aktif dibasis data pada suatu periode.

e. File sejarah (*history file*) Disebut juga sebagai file arsip (*archival file*) yaitu file yang berisi data masa lalu yang sudah tidak aktif lagi, tetapi perlu disimpan untuk keperluan dimasa mendatang.

f. File kunci Digunakan sebagai identifikasi data yang terdapat didalam tabel. Field kunci harus bersifat unik. Field kunci terdiri dari dua macam yaitu :

a. *Primary key* adalah suatu atau beberapa field yang digunakan untuk mengidentifikasi record pada suatu tabel. Primary key bersifat unik, tidak ada record yang dalam sebuah tabel yang memiliki primary key yang sama.

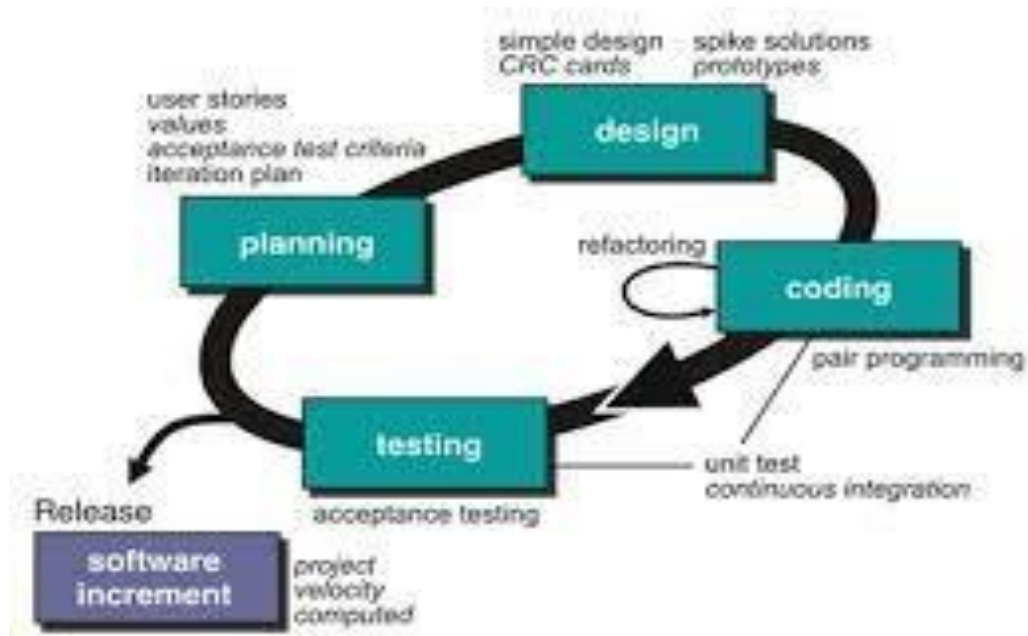
b. *Foreign Key* adalah primary key dari suatu tabel yang menjadi non primary key ditabel yang lain.

## **2.5 Metode Pengembangan Sistem**

Metode pengembangan yang digunakan dalam membuat system informasi penagihan piutang adalah dengan menggunakan metode Extreme Programming. Metode Extreme Programming adalah melakukan penyederhanaan berbagai tahapan pengembangan sistem informasi menjadi lebih efisien, adaptif



dan fleksibel. Nilai dasar extreme programming antara lain communication, Courage, Simplicity, Feedback dan Quality work. (Mukrodin 2021).



Gambar 2.1 Metode Extreme Programming

Metode Extreme Programming memiliki tahapan-tahapan sebagai berikut:

1. *Planning* (Perencanaan)

Dimulai dengan pemahaman konteks bisnis dari aplikasi, mendefinisikan output, fitur yang ada pada aplikasi, fungsi dari aplikasi yang dibuat, serta alur pengembangan aplikasi. Dapat dikatakan bahwa tahapan ini menentukan fungsionalitas keseluruhan yang akan dikembangkan dalam sistem.

2. *Design* (Perancangan)

Pada tahapan perancangan fokus pada design aplikasi sederhana, alat untuk mendesign pada tahapan ini yaitu CRC (*Class Responsibility Collaborator*). CRC memetakan kelas kelas yang akan dibangun dalam Use Case Diagram, Class Diagram, Activity Diagram dan juga Sequence Diagram.

3. *Coding* (Pengkodean)

Coding atau pengkodean merupakan penerjemahan dari perancangan dalam bahasa pemrograman yang dikenali oleh komputer. Pada penelitian ini aplikasi dibagi menjadi dua, yaitu untuk front-end dan back-end. Pengkodean

dengan menggunakan bahasa pemrograman PHP dengan *compiler visual studio code* dan *database MySQL*.

#### 4. *Testing* (Pengujian)

Sistem yang telah dibangun harus diuji terlebih dahulu agar dapat menemukan kesalahan-kesalahan. Pada penelitian ini menggunakan pengujian terhadap *usability*, dimana pengujian dilakukan untuk mengetahui pengguna apakah dapat mempelajari dan menggunakan produk untuk mencapai tujuannya dan seberapa kepuasan pengguna terhadap penggunaan aplikasi serta kegunaannya.

## **2.5 Alat dan Teknik Pengembangan**


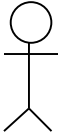

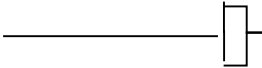



### ***UML (Unified Modelling Language)***

Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. *Unified Modeling Language (UML)* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. *Unified Modeling Language (UML)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. (Hendra Nusa Putra, 2018)

#### ***2.5.1 Use Case Diagram***

*Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.1 :


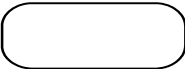
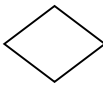

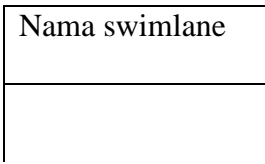

**Tabel 2.2 Simbol Use Case Diagram**

No	Simbol	Deskripsi
1.		<i>Use case</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi ( <i>association</i> ): merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Generalisasi ( <i>generalization</i> ): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5.	<< Include >> 	Include berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.
6.	<<Extend>> 	Ekstensi ( <i>extend</i> ) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5.	<< Include >> 	Include berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.

### 2.5.2 Activity Diagram

*Activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.3 berikut ini.

**Tabel 1.3 Simbol Activity Diagram**

No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan ( <i>Decision</i> ) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan ( <i>Join</i> ) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

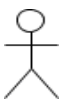
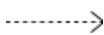

### 2.5.3 Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang


digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu.

Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar *UML* mendefinisikan icon khusus untuk *object boundary*, *controller* dan *persistent entity*. Komponen – komponen dari *Sequence Diagram* dapat dilihat pada tabel 2.3 berikut :

**Tabel 2.4 Sequence Diagram**

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Mendefinisikan entitas diluar sistem yang memakai sistem. Contohnya orang
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana antara dua use case generalisasi dan spesialisasi dimana fungsi yang satu lebih umum dan lainnya khusus

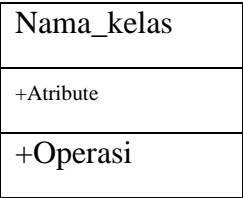


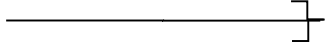

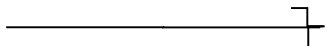
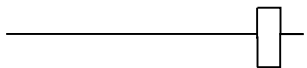
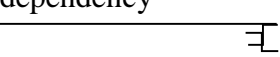
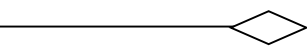
**Tabel 2.4** *Sequence Diagram* (lanjutan)

4		<i>Include</i>	Menspesifikasikan relasi use case tambahan ke sebuah use case yang ditambahkan dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya <b>atau sebagai syarat</b>
5		<i>Extend</i>	Menspesifikasikan bahwa relasi <i>use case</i> yang ditambahkan dapat berdiri <b>sendiri.</b>
6		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang terlibat didalam sistem
7		<i>System</i>	Menspesifikasikan paketyang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari fungsionalitas dari suatu sistem, sehingga pengguna mengerti kegunaan sistem yang akan <b>dibangun.</b>

#### 2.5.4 Class Diagram

Menurut, *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.4 sebagai berikut.

**Tabel 2.4 Simbol *Class Diagram***

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	<p>Antar Muka/Interface</p>  <p>Nama_Interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / Association</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / Directed Association</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
3.	<p>Asosiasi / Association</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / Directed Association</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Ketergantungan dependency</p> 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	<p>Agregasi / aggregation</p> 	Relasi antar kelas dengan maksna semua bagian ( <i>whole-part</i> )

## ***2.6 blackbox testing***

Metode *black box testing* merupakan pengujian program yang mengutamakan pengujian terhadap kebutuhan fungsi dari suatu program dengan menemukan kesalahan fungsi pada program (Neni Purwati, Halimah dan Agus Rahardi, 2018)

Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian yang dilakukan dengan membuat kasus yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji dilakukan harus dibuat dengan benar dan salah, seperti contoh pada proses *login* “Jika user memasukkan *username* dan *password* yang benar maka dapat *login* ” dan sebagainya.