

## **BAB II LANDASAN TEORI**

### **2.1 Tinjauan Pustaka**

Pada penelitian ini penulis melakukan tinjauan pustaka pada beberapa penelitian sebelumnya untuk mendukung penelitian yang sedang dilakukan. Berikut adalah tinjauan pustaka yang digunakan penulis yang dapat dilihat pada Tabel 2.1:

**Tabel 2.1 Tinjauan Pustaka**

<b>Nomor Literatur</b>	<b>Penulis dan Tahun</b>	<b>Judul</b>
Literatur 1	(Febrianti, Muthmainnah and Sufyana, 2023)	Perancangan Sistem Informasi Akuntansi Kas Kecil (Petty Cash) Di PT. Koprima Sandysejahtera
Literatur 2	(Zahra and Devitra, 2023)	Sistem Informasi Akuntansi Kas Kecil Pada Yayasan Kria Yudha Nusantara
Literatur 3	(Yulistiawati, Apdian and Hasmizal, 2022)	Sistem Informasi Akuntansi Kas Kecil Dengan Metode Fluktuasi Pada Pt. Plastik Karawang Flexindo Berbasis Vb.Net
Literatur 4	(Bayu Santoso and Informasi, 2022)	Analisis Dan Perancangan Aplikasi Pengendalian Intern Petty Cash Berbasis Web
Literatur 5	(Annisa, Syahidin and Karyadi, 2022)	Perancangan Sistem Informasi Akuntansi Kas Kecil (Petty Cash) Berbasis Visual Dan Object Oriented Di Alfamart Kiaracondong Bandung

#### **1. Perancangan Sistem Informasi Akuntansi Kas Kecil (Petty Cash) Di PT. Koprima Sandysejahtera, Jurnal Manajemen Informatika dan Komunikasi (2023)**

Kas Kecil merupakan dana yang disediakan oleh perusahaan untuk kegiatan operasional yang nominalnya relatif kecil. Dalam pengolahan dana kas kecil di

PT. Koprime Sandysejahtera masih menggunakan Microsoft Excel, hal ini menjadikan kinerja perusahaan dalam pengolahan dana kas kecil kurang efektif. Oleh karena itu penulis melakukan perancangan sistem informasi akuntansi kas kecil berbasis web menggunakan bahasa pemrograman PHP dan My SQL sebagai database. Metode penelitian yang digunakan yaitu kualitatif. Data dan informasi didapatkan melalui observasi, wawancara, dokumentasi, dan studi pustaka yang berkaitan dengan penelitian. Metode pengembangan perangkat lunak yang digunakan yaitu model waterfall dimana tahapannya berurutan mulai dari analisis sistem sampai pemeliharaan sistem dan tidak dapat dilakukan secara acak. Dengan adanya sistem yang terkomputerisasi diharapkan kinerja perusahaan dalam hal pengolahan dana kas kecil semakin efektif.

## **2. Sistem Informasi Akuntansi Kas Kecil Pada Yayasan Kria Yudha Nusantara, Jurnal Manajemen Sistem Informasi (2023)**

Yayasan Kria Yudha Nusantara yaitu belum dapat menghasilkan penyajian laporan akuntansi terkait pengeluaran kas kecil secara cepat, tepat dan akurat dikarenakan proses pengelolaan laporan dilakukan secara manual menggunakan Microsoft Excel 2010 sehingga mengakibatkan sulitnya pencarian, menampilkan dan mencetak data laporan setiap bulannya. Oleh karena itu, dibutuhkan sistem informasi akuntansi kas kecil untuk mempermudah pihak admin Yayasan Kria Yudha Nusantara dalam mengelola laporan kas kecil. Menggunakan metode pendekatan berorientasi objek (*Object Oriented Analysis and Design*) menggunakan UML (*Unified Modelling Language*) untuk menghasilkan sebuah prototipe perancangan sistem informasi akuntansi kas kecil dengan metode fluktuasi yang dirancang untuk mengelola penerimaan dan pengeluaran kas kecil

yang memadai agar dapat mengawasi dan mengendalikan dana kas kecil. Penelitian ini menghasilkan sebuah prototype Sistem Informasi Akuntansi Kas Kecil Pada Yayasan Kria Yudha Nusantara berbasis database yang dapat diimplementasikan lebih lanjut sehingga menghasilkan Sistem Informasi Akuntansi Kas Kecil yang dapat diterapkan pada Yayasan Kria Yudha Nusantara

### **3. Sistem Informasi Akuntansi Kas Kecil Dengan Metode Fluktuasi Pada PT. Plastik Karawang Flexindo Berbasis Vb.Net, Jurnal Manajemen Sistem Informasi (2023)**

PT Plastik Karawang Flexindo merupakan perusahaan manufaktur yang menghasilkan barang plastik dengan bahan baku yang ramah lingkungan. Sama halnya dengan perusahaan yang lain PT Plastik Karawang Flexindo juga memiliki dana kas kecil yang berguna untuk membiayai pengeluaran operasional harian perusahaan yang jumlahnya relatif kecil didalam pencatatan pengeluaran dan kas kecil masih memanfaatkan sekaligus menggunakan microsoft excel. Namun belum dibuatkan penjurnalan untuk setiap laporan pengeluaran kas. Metode analisis data yang digunakan dalam penelitian ini adalah Design Science Research Methodology (DSRM), yaitu digunakan untuk mengembangkan sistem yang terkait dengan desain sebuah layanan dalam bentuk sistem informasi. Hasil penelitian menunjukkan bahwa metode yang digunakan dalam pengelolaan kas kecil, oleh PT. Plastik Karawang Flexindo yaitu metode sistem dana tidak tetap (fluktuatif) dimana dalam metode ini pengisian kembali dana kas kecil dilakukan sesuai dengan keperluan tidak berdasarkan jumlah pengeluaran sebelumnya dan dicatat dengan mendebit rekening dana kas kecil

#### **4. Analisis Dan Perancangan Aplikasi Pengendalian Intern Petty Cash Berbasis Web, Jurnal Teknologi (2022)**

TOP Finance adalah sebuah perusahaan swasta nasional yang mengkhususkan diri dalam kegiatan peminjaman dana dengan jaminan BPKB kendaraan bermotor. Top Finance menggunakan kas kecil dengan metode Imprest system atau disebut dana tetap yang digunakan untuk keperluan sehari-hari pada kantor. Top Finance cabang Tanjung Bintang menunjuk admin dalam mengelola kas kecil. Kurangnya dokumen yang dimiliki yaitu dokumen bukti pengeluaran kas kecil yang digunakan sebagai bukti pertanggung jawaban pemakai kas kecil yang mengakibatkan tidak ada dokumen yang memberikan informasi rincian rekapitulasi pengeluaran kas kecil dan nomor bukti setiap terjadi transaksi, serta tidak memiliki laporan kas kecil yang mengakibatkan tidak adanya pengendalian intern dalam pengawasan pada penerimaan kas kecil dan pengeluaran kas kecil sehingga dapat terjadi penyimpangan dana kas kecil. Pengolahan pencatatan pengeluaran kas kecil dan pengisian kembali kas kecil yang masih manual membuat data informasi yang diberikan tidak lengkap dan sering terjadi keterlambatan dalam pengisian kembali kas kecil karena tidak adanya hak akses yang diberikan untuk pertanggung jawaban kepada admin dan kacab sebagai pengendalian intern. Desain sistem berbasis web yang dapat diakses oleh pusat dan cabang tanjung bintang dengan mudah dan tepat waktu yang memberikan informasi data kas kecil secara lengkap dan terinci yaitu penerimaan kas kecil, pengeluaran kas kecil, saldo akhir dan dapat mencetak laporan kas kecil.

**5. Perancangan Sistem Informasi Akuntansi Kas Kecil (Petty Cash) Berbasis Visual Dan Object Oriented Di Alfamart Kiarcondong Bandung, Jurnal Co Management (2022)**

Pesatnya perkembangan teknologi mempengaruhi banyak aspek, salah satunya adalah aspek ekonomi. Persaingan antar pelaku usaha semakin ketat. agar dapat bersaing di pasar, diperlukan upaya peningkatan kualitas kerja, kualitas pelayanan dan akurasi data. Pengelolaan Sistem Informasi Akuntansi pada perusahaan mini market masih belum optimal yaitu menggunakan buku pengawasan yang dibuat menggunakan microsoft excel sehingga sering terjadi kesalahan karena lupa mencatat dan waktu yang digunakan juga tidak efisien. Pada penelitian ini akan dirancang sistem informasi akuntansi kas kecil berbasis object oriented dengan menggunakan UML dan diimplementasikan ke dalam bahasa pemrograman visual dengan tujuan untuk memberikan kebutuhan pengelolaan data keuangan kas kecil pada perusahaan retail dan menghasilkan pelaporan yang dibutuhkan secara lebih efektif dan efisien.

Berdasarkan penelitian yang dilakukan sebelumnya terdapat beberapa perbedaan dalam peneliti yaitu:

1. Metode yang digunakan dalam peneltiian yaitu metode *extreme programming*.
2. Sistem yang dibangun diakses oleh pimpinan dan keuangan.
3. Sistem yang dibangun menghasilkan laporan secara periode (perbulan, tanggal, dan tahun).
4. Sistem yang dibangun menampilkan grafik *petty cash*.

## 2.2. *Petty Cash*

*Petty Cash* merupakan sejumlah uang kas atau uang tunai yang disediakan perusahaan untuk membayar pengeluaran-pengeluaran yang jumlahnya relative kecil dan tidak ekonomis bila dibayar dengan. *Petty Cash* adalah pengeluaran kas dengan uang tunai, dimana dapat diselenggarakan dengan dua cara sistem saldo berfluktuasi dan imprest sistem (Bayu Santoso and Informasi, 2022).

Dana Kas Kecil atau biasa disebut *Petty Cash* adalah dana khusus yang disediakan oleh perusahaan yang harus segera dilakukan yang jumlah relatif kecil serta tidak ekonomis praktis apabila cek atau giro. Pengelolaan kas kecil diserahkan kepada pemegang kas kecil yang bertanggungjawab terhadap yang diambil dari dana kas kecil. Pengelola dana kas kecil tidak berhak menerima pembayaran dari pihak luar atau dana yang dikelola hanya yang diterima dari pemegang kas besar atau kas umum (Abdus Salam *et al.*, 2022).

Metode kas kecil mencakup tahapan-tahapan pengelolaan dalam penggunaan dana. Secara garis besar, metode kas kecil dibagi menjadi 2, yaitu:

### 1. *Imprest Fund System* (Metode Tetap)

Metode kas kecil yang pertama adalah metode tetap. Metode kas kecil ini membuat pembukuan kas kecil dengan jumlah rekening sama atau tetap. Ciri-ciri metode tetap antara lain:

- a. Pemegang kas kecil adalah orang yang mengumpulkan bukti penggunaan dana kas kecil.

- b. Pengisian dana kas kecil dilakukan melalui penarikan cek dimana cek tersebut memiliki jumlah sama dengan dana yang telah digunakan. Sehingga jumlah kas kecil kembali ke jumlah semula.

## 2. *Fluctuating Fund System* (Metode Berubah-ubah)

Metode kas kecil yang kedua adalah metode berubah-ubah. Metode kas kecil satu ini melakukan pengisian dan pengendalian jumlah dana kas secara berubah-ubah (sesuai dengan kebutuhan). Berikut adalah ciri-ciri metode kas:

- a. Pembentukan dan pengisian kembali dananya dicatat di debit dalam akun kas kecil.
- b. Besar dana yang disediakan menyesuaikan kegiatan dari divisi yang menggunakan dana.
- c. Bukti pengeluaran kas kecil yang dicatat dalam buku kas kecil adalah transaksi penggunaan kredit akun kas kecil.

### 2.2.1. Tahapan Pengelolaan

Berikut ini adalah langkah-langkah atau prosedur dalam pengelolaan kas kecil, yaitu:

1. Menetapkan batas saldo kas kecil, dari awal dibuatnya akun kas kecil, pihak manajemen harus sudah menetapkan saldo kas kecil yang disesuaikan dengan kebutuhan operasional usaha dalam jangka waktu tertentu. Ada dua jenis metode pencatatan kas kecil yang bisa digunakan sebagai dasar dalam menentukan jumlah saldo petty cash, yaitu metode imprest dan fluktuatif. Metode imprest di mana saldo kas kecil selalu tetap sebesar jumlah saldo yang sudah ditetapkan. Biasanya pada akhir periode kasir kas kecil akan

meminta pengisian kembali kas kecil sejumlah yang sudah dikeluarkan. Dengan demikian pengeluaran kas kecil baru dicatat pada saat pengisian kembali. Metode kedua adalah fluktuatif, yaitu saldo kas kecil tidak tetap, namun berfluktuasi sesuai dengan jumlah pengeluaran kas kecil.

2. Menentukan kasir kas kecil, setelah batas saldo ditetapkan, maka selanjutnya harus ada personel yang bertanggung jawab terhadap pengelolaan kas kecil atau biasa disebut dengan kasir kas kecil. Karena kasir kas kecil ini menangani transaksi kecil yang sifatnya rutin, tidak semua personel dapat menempati posisi ini. Ada beberapa kriteria yang harus dipertimbangkan dalam menentukan posisi kasir, seperti:
  - a. Menguasai dasar-dasar akuntansi
  - b. Mampu meng-handle transaksi dalam jumlah kecil
  - c. Jujur
  - d. Konsisten
  - e. Mampu menguasai program komputer, seperti Excel.

Selain itu pihak manajemen sebaiknya juga memberikan training sebelum kasir memulai tugasnya, seperti memberi arahan prosedur pengisian kembali kas kecil hingga cara rekonsiliasi kas kecil serta prosedur pembelian.

3. Pengisian kembali kas kecil, setelah menentukan batas saldo dan metode pencatatan kas kecil, selanjutnya *Financial Controller* memerintahkan pengisian kepada kasir umum dengan menarik kas dari bank. Kemudian uang diserahkan kepada kasir kas kecil. Langkah selanjutnya akan dilakukan penghitungan secara fisik dan harus ada bukti serah terima. Pada proses ini, kasir kas kecil menandatangani tanda terima atas dana yang diserahkan serta

sebagai tanda serah terima tanggung jawab terhadap dana tersebut. Jika suatu saat saldo kas kecil minim, maka kasir kas kecil harus mengajukan permohonan pengisian kembali kepada Financial Controller.

4. Penggunaan kas kecil, kasir kas kecil hanya diperbolehkan melakukan pembayaran kas kecil untuk permohonan pembayaran atas pembelian yang sudah disetujui oleh *Financial Controller*. Kasir kas kecil harus membuat bukti pengeluaran kas kecil untuk setiap pengeluaran yang ditandatangani oleh penerima dana. Kemudian transaksi dicatat dalam buku kas kecil dan bukti pengeluaran harus disimpan atau diarsip. Kasir kas kecil harus menghitung nominal kas kecil secara fisik setiap selesai melakukan transaksi. Setelah dihitung, fisik kas kecil dicocokkan dengan pencatatan kas kecil yang sudah dibuat oleh kasir kas kecil. Dengan demikian akan mengurangi beban kerja saat rekonsiliasi penutupan kas kecil setiap harinya. Financial Controller juga harus melakukan pemeriksaan kas kecil dengan membandingkan antara fisik kas kecil dengan catatan yang sudah dibuat demi menghindari terjadinya kecurangan.

#### **2.2.2. Dokumen Terkait**

Dalam pencatatan kas kecil terdapat beberapa dokumen penting yang gunanya sebagai penunjang saat pengisian. Setidaknya terdapat 5 dokumen penting yang wajib ada dalam pencatatan kas kecil, yaitu:

1. Bukti Kas Keluar, dokumen ini berfungsi untuk memungkinkan pengeluaran kas dari kas besar yang tercantum dalam dokumen pembukuan. Dalam praktiknya, dokumen ini digunakan ketika pencatatan atau pengisian kembali dana kas kecil.

2. Cek merupakan dokumen yang digunakan untuk meminta bank melakukan sejumlah pembayaran kepada pihak tertentu yang namanya tertera pada cek tersebut.
3. Permintaan Pengeluaran Kas Kecil menggunakan dokumen ini untuk meminta uang kepada pihak pemegang dana kas kecil. Sedangkan bagi pihak pemegang kas kecil, dokumen ini nantinya akan digunakan sebagai dokumen bukti pengeluaran.
4. Bukti Pengeluaran Kas Kecil, pengguna dana kas kecil membuat dokumen bukti pengeluaran sebagai bukti pertanggungjawaban atas pemakaian dana kas kecil. Dalam membuat dokumen ini wajib melampirkan dengan bukti-bukti pengeluaran kas kecil yang sudah digunakan. Selanjutnya, ketika dokumen sudah lengkap, maka akan diserahkan oleh pengguna dana kas kecil ke pemegang dana kas kecil.
5. Permintaan Pengisian Kembali Kas Kecil membuat dokumen ini untuk meminta kepada bagian utang agar menyediakan bukti kas keluar. Bukti ini kemudian digunakan untuk pengisian kembali dana kas kecil.

### **2.2.3. Laporan**

1. Jurnal kas kecil bisa dibuat dan disusun berdasarkan dua metode laporan kas kecil yang sudah dibahas tadi. Untuk bisa memahami dengan lebih jelas, berikut ada beberapa contoh pencatatan laporan kas kecil.
2. Laporan Kas kecil adalah pembukuan perusahaan yang berupa pencatatan setiap pengeluaran dengan nominal yang tidak terlalu banyak. Contohnya adalah pembukuan untuk pembayaran klaim atau *reimbursement* karyawan, pembelian ATK dan fotokopi dokumen, konsumsi, dan masih banyak lagi.

### 2.2.4. Kode Transaksi

Berikut ini adalah tabel kode transaksi dalam pengelolaan kas kecil yaitu:

**Tabel 2. 2 Kode Transaksi**

No	Kelas, Kelompok Dan Jenis Akun	Sistem Desimal	Sistem Inemorik	Sistem Kombinasi
1	<b>HARTA</b>	<b>1</b>	<b>H</b>	<b>H 1</b>
	<b>Harta lancar</b>	01.01	HL	H 11
	1. Kas	01.01.01	HL-K	HL 111
	2. Surat-surat berharga	01.01.02	HL-SB	HL 112
	3. Piutang usaha	01.01.03	HL-PU	HL 113
	4. Wesel tagih	01.01.04	HL-WT	HL 114
	5. Persekot / uang usaha	01.01.05	HL-P	HL 115
	6. Pendapatan yang masih harus diterima.	01.01.06	HL-Pmhd	HL 116
	7. Perlengkapan tokoh	01.01.08	HL-PT	HL 118
	8. Perlengkapan kantor	01.01.09	HL-PK	HL 119
2	<b>Investasi</b>	01.02	HI	HI 12
	1. Berupa uang tunai	01.02.01	HI-BUT	HI 121
	2. Surat berharga	01.02.02	HI-SD	HI 122
3	<b>Harta tetap berwujud</b>	01.03	HTB	HTB 13
	1. Peralatan tokoh	01.03.01	HTB-PT	HTB 131
	2. Peralatan kantor	01.03.02	HTB-PK	HTB 132
	3. Kendaraan	01.03.03	HTB-K	HTB 133
	4. Mesin	01.03.04	HTB-M	HTB 134
	5. Bangunan	01.03.05	HTB-B	HTB 135
	6. Tanah	01.03.06	HTB-T	HTB 136

Tabel 2.2 Kode Transaksi (Lanjutan)

No	Kelas, Kelompok Dan Jenis Akun	Sistem Desimal	Sistem Inemorik	Sistem Kombinasi
4	<b>Harta tetap tak berwujud</b>	14	HTTB	HTTB 14
	1. Copyright	141	HTTB-CR	HTTB 141
	2. Hak paten	142	HTTB-HP	HTTB 142
	3. Hak merek	143	HTTB-HM	HTTB 143
	4. Hak cipta	144	HTTB-HC	HTTB 144
	5. Good will	145	HTTB-GW	HTTB 145
	6. Paransais	146	HTTB-P	HTTB 146
5	<b>Harta lain-lain</b>	15	HLL	HLL 15
	1. Semua yang rusak	151	HLL-SYR	HLL 151
1	<b>UTANG</b>	<b>2</b>	<b>U</b>	<b>U 2</b>
	<b>Utang jangka pendek</b>	02.01	UJPD	UJPD 21
	1. Utang usaha	02.01.01	UJPD-UU	UJPD 211
	2. Utang bunga	02.01.02	UJPD-UB	UJPD 212
	3. Utang gaji	02.01.03	UJPD-UG	UJPD 213
	4. Utang sewa	02.01.04	UJPD-US	UJPD 214
	5. Beban yang harus dibayar	02.01.05	UJPD-BHB	UJPD 215
	6. Beban yang diterima dimuka	02.01.06	UJPD-BDM	UJPD 216
	7. Utang listrik	2..1.7	UJPD-UL	UJPD 217
	8. Utang bayar	02.01.08	UJPD-UIB	UJPD 218
2	<b>Utang jangka panjang</b>	22	UJP	UJP 22
	1. Utang bank	02.02.01	UJP-UB	UJP 221
	2. Utang hipotik	02.02.02	UJP-UI	UJP 222

Tabel 2.2 Kode Transaksi (Lanjutan)

No	Kelas, Kelompok Dan Jenis Akun	Sistem Desimal	Sistem Inemorik	Sistem Kombinasi
2	<b>Utang jangka panjang</b>	22	UJP	UJP 22
	3. Utang KIK	02.02.03	UJP-KIK	UJP 223
	4. Utang KUK	02.02.04	UJP-KUK	UJP 224
	5. Utang KUM	02.02.05	UJP-KUM	UJP 225
	6. Utang KRIDA	02.02.06	UJP-KRIDA	UJP 226
1	<b>MODAL</b>	<b>3</b>	<b>M</b>	<b>M 3</b>
	<b>Modal sendiri</b>	03.01	MSD	MS 31
	1. Modal berupa uang tunai	03.01.01	MSD-UT	MS 311
	2. Modal berupa surat berharga	03.01.02	MSD-SB	MS 312
2	<b>Modal pinjaman</b>	03.02	MP	MP 32
	1. Modal pinjaman berupa uang tunai	03.02.01	MP-UT	MP 321
	2. Modal pinjaman surat berharga	03.02.02	MP-SB	MP 322
3	<b>Modal saham</b>	03.03	MS	MS 33
	1. Modal saham biasa	03.03.01	MS-SB	MS 331
	2. Modal saham istimewa	03.03.02	MS-SI	MS 332
1	<b>PENDAPATAN</b>	<b>4</b>	<b>P</b>	<b>P 4</b>
	<b>Pendapatan usaha</b>	04.01	PU-PJ	PU 41
	1. Pendapatan jasa salon	04.01.01	PU-PJS	PU 411
	2. Pendapatan jasa angkutan	04.01.02	PU-PJA	PU 412
	3. Pendapatan jasa bengkel	04.01.03	PU-PJB	PU 413
	4. Pendapatan jasa servis	04.01.04	PU-PJS	PU 414

Tabel 2.2 Kode Transaksi (Lanjutan)

No	Kelas, Kelompok Dan Jenis Akun	Sistem Desimal	Sistem Inemorik	Sistem Kombinasi
1	<b>Pendapatan usaha</b>	04.01	PU-PJ	PU 41
	5. Pendapatan jasa tailor	04.01.05	PU-PJT	PU 415
	6. Pendapatan jasa laundry	04.01.06	PU-PJL	PU 416
	7. Pendapatan jasa komisi	04.01.07	PU-PJK	PU 417
2	<b>Pendapatan di luar usaha</b>	42	PLU	PLU 42
	1. Pendapatan bunga	04.02.01	PLU-PB	PLU 421
	2. Laba penjualan harta	04.02.02	PLU-PH	PLU 422
1	<b>BEBAN</b>	<b>5</b>	<b>B</b>	<b>B 5</b>
	<b>Beban usaha</b>	05.01	BU	BU 51
	1. Beban gaji	05.01.01	BU-BG	BU 511
	2. Beban sewa	05.01.02	BU-BS	BU 512
	3. Beban listrik	05.01.03	BU-BL	BU 513
	4. Beban asuransi	05.01.04	BU-BA	BU 514
	5. Beban perlengkapan	05.01.05	BU-BP	BU 515
	6. Beban penyusutan harga tetap	05.01.06	BU-Bpht	BU 516
	7. Beban rupa-rupa	05.01.07	BU-R	BU 517
2	<b>Beban di luar usaha</b>	05.02	BLU	BLU 52
	1. Beban bunga	05.02.01	BLU-BB	BLU 521
	2. Beban penjualan harta	05.02.02	BLU-BPH	BLU 522
	3. Kecurian uang/barang	05.02.03	BLU-KU	BLU 523

### 2.3. Metode Analisis *PIECES*

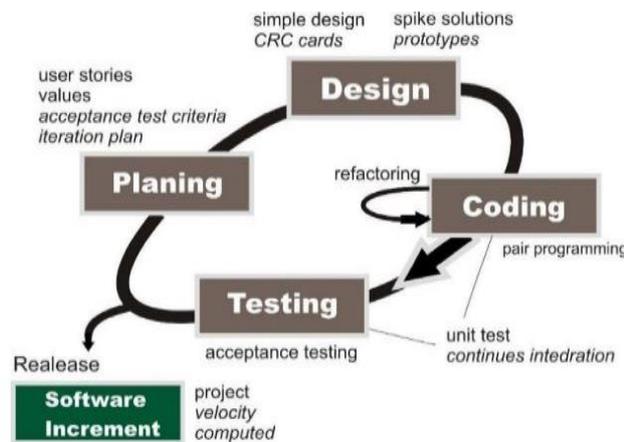
Analisis pieces juga sangat banyak diterapkan untuk penelitian pada suatu perusahaan atau organisasi. Selain mudah dan dapat dipahami analisis pieces juga bersifat ringan tidak membutuhkan data yang banyak. Berikut ini kriteria yang wajib ada pada analisis pieces menurut (Sanusi, 2012) :

1. Analisis Kinerja (*Performance*), kinerja sistem saat ini masih jauh dari yang diharapkan. Pekerja Toko harus melakukan pengecekan data secara berulang agar tidak terjadi kesalahan dalam pengolahan data, hal ini menunjukkan kinerja yang belum optimal dan membutuhkan waktu yang cukup lama dalam pengolahan data penjualan. Akhirnya bisa memperburuk kinerja karyawan. Jika kinerja karyawan menurun, maka performance perusahaan pun akan menurun.
2. Analisis Informasi (*Information*), sistem yang berjalan belum dapat memberikan informasi yang akurat yang terbebas dari kesalahan, dan informasi yang dihasilkan sering terjadi kesalahan. Selain itu pelanggan mengalami kesulitan untuk memperoleh informasi mengenai produk baru. Hal ini juga terjadi karena karyawan lebih sedikit dan pelanggan lebih banyak sehingga *input* dan *output* nya kurang diperhatikan sehingga bisa merusak informasi yang tepat. Pelanggan dari kalangan bisnis juga banyak dan belum ter-handle secara sistem.
3. Analisis Ekonomi (*Economy*), proses pengelolaan yang dilakukan saat ini masih dilakukan dengan cara pencatatan dokumen. Dalam pembuatan laporan masih banyak pengolahan data yang berulang dan mengakibatkan pemborosan biaya.

4. Analisis Kendali (Control), pengelolaan sistem yang berjalan sudah dirasa cukup baik dalam pengendalian, hal ini ditunjukkan dengan adanya dokumen yang dibuat lengkap dalam prosesnya.
5. Analisis Efisiensi (*Efficiency*), penggunaan waktu yang belum maksimal dikarenakan sering terjadinya pengecekan data secara berulang agar tidak terjadi kesalahan dalam pengolahan datanya, sehingga mempengaruhi proses pembuatan laporan.
6. Analisis Layanan (*Services*) dalam hal pelayanan masih dirasa belum baik, perlu ditingkatkan dengan cara meningkatkan keakuratan serta kecepatan dalam proses pelayanan sehingga sangatlah diperlukan sistem yang berbasis database dan real-time.

#### **2.4. Metode Pengembangan Sistem**

*Extreme Programming* (XP) adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak yang dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. Tahapan - tahapan dari *Extreme Programming* terdiri dari planning seperti memahami kriteria pengguna dan perencanaan pengembangan, designing seperti perancangan prototype dan tampilan, coding termasuk pengintegrasian, dan yang terakhir adalah testing (Ependi, Usman and Widayati, 2013).



Sumber : (Ependi, Usman and Widayati, 2013)

**Gambar 2. 1 Tahapan pada *Extreme Programming (XP)*.**

Dibawah ini adalah penjelasan tahapan Extreme Programming yaitu:

### 1. *Planning* (Perencanaan)

Kegiatan Perencanaan (disebut juga *planning game*) biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan mengumpulkan kebutuhan kebutuhan untuk memahami konteks bisnis dan perlunya keluaran-keluaran (*output*), fungsi utama, dan fungsionalitas. Pada perencanaan terdapat *user stories values* yaitu *story* dengan *value* tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama, *acceptance test criteria integration plan* melakukan perhitungan kecepatan *project* selama *development*, customer dapat menambah *story*, merubah *value*, membagi *story* atau menghapusnya.

### 2. *Design* (Perancangan)

Perancangan yang sederhana, menarik, dan sederhana selalu memberikan hasil disukai daripada gambaran-gambaran yang lebih kompleks. Perancangan XP memberikan panduan implementasi untuk suatu cerita ketika ditulis, tidak kurang, tidak lebih. Terdapat *simple design CRC Cards* untuk mengenali dan mengatur

*object oriented class* sesuai dengan *software increment* dan *spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.

### 3. *Coding* (Pengkodean)

Pengkodean ini dilanjutkan setelah cerita yang telah dikembangkan dan 16 rancangan yang telah dilakukan oleh tim perangkat lunak. Pengkodean ini tidak langsung mengarah ke kode-kode program. Tim akan mengembangkan serangkaian unit pengujian lalu beralih ke pengkodean. Pada tahapan pair programming melakukan kerja sama untuk membuat kode dari satu story. Dan refactoring adalah proses restrukturisasi kode program komputer yang ada tanpa mengubah perilaku eksternalnya.

### 4. Pengujian (Pengujian)

Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi sehingga dapat dijalankan dengan mudah dan dapat dijalankan berulang kali. Pada tahapan pengujian yaitu unit test *continuous integration* yaitu tahapan pengujian code yang diintegrasikan dengan kerja lainnya dengan pengujian yang dilakukan oleh customer dan focus pada keseluruhan dan fungsional sistem, dan *acceptance testing* yaitu pengujian yang dilakukan *customer stories* yang akan diimplementasikan sebagai bagian dari *software release*. Selanjutnya terdapat tahapan *software increment project velocity computed* yaitu tahapan yang telah diimplementasikan dari *software release* yang nantinya akan diterapkan dalam suatu sistem.

## 2.5. Unified Modeling Language (UML)

UML merupakan pengembangan dari teknik pemrograman berorientasi objek, menghasilkan bahasa pemodelan yang terstandarisasi untuk pengembangan perangkat lunak untuk membuat analisis dan perancangan, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa dan Shalahudin, 2018).

### 2.5.1. Use Case Diagram

*Use Case* adalah model untuk perilaku aplikasi yang akan dibuat. *Use case* mendeskripsikan interaksi satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi-fungsi apa saja yang ada dalam suatu sistem informasi dan siapa yang berhak menggunakan fungsi tersebut.

**Tabel 2.3 Simbol Use Case Diagram**

Simbol	Deskripsi
<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 frase nama <i>use case</i>
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang.
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor
Ekstensi/ <i>extend</i> << <i>extend</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

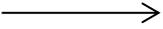
Simbol	Deskripsi
Menggunakan/ <i>Include/uses</i> <pre>&lt;&lt;include&gt;&gt;</pre> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

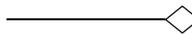
Sumber : (Rosa and Shalahudin, 2018)

### 2.5.2. Class Diagram

*Class diagram* adalah sebuah *class* yang menggambarkan struktur sistem dari segi pendefinisian kelas kelas yang akan dibuat untuk membangun sistem. *Class diagram* juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.

**Tabel 2. 4 Simbol Class Diagram**

No.	Simbol	Deskripsi
1	Kelas / <i>class</i> 	Kelas pada struktur sistem.
2	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

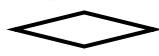
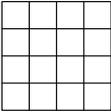
No.	Simbol	Deskripsi
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisai (umum-khusus).
6	Agresi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

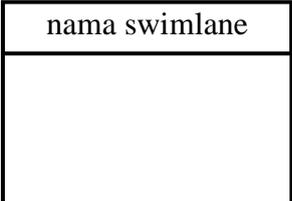
Sumber: Rosa and Shalahudin (2018)

### 2.5.3. Activity Diagram

*Activity diagram* atau diagram aktivitas ini menggambarkan rangkaian aliran dari aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

**Tabel 2. 5 Simbol Activity Diagram**

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis

Simbol	Deskripsi
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: (Rosa dan Shalahiddin, 2018)

## 2.6. Pengertian PHP

*PHP (Hypertext apareprocessor)* adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasil nya dikirimkan ke browser dalam format HTML. PHP juga dapat berjalan pada berbagai web server seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache, Xitami. PHP juga mampu lintas platform. Artinya PHP dapat berjalan dibanyak sistem operasi yang beredar saat ini, di antaranya: Sistem Operasi Microsoft Windows (semua versi), Linux, Mac OS, Solaris. PHP dapat dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai CGI (*Common Gateway Interface*). PHP dapat mengirim HTTP header, dapat mengatur cookies, mengatur *authentication* dan *redirect user* (Gilmore, 2015).

## **2.7. Pengertian *MySQL***

Gilmore (2015) *MySQL* adalah server database relasional yang menawarkan berbagai mekanisme untuk memproses data yang dikenal sebagai mesin penyimpanan.

Sheldon and Moes (2015) *MySQL* sama seperti sistem DBMS lainnya, seperti Oracle, DB2, dan SQL Server. Memungkinkan untuk mengakses, memanipulasi, melindungi, dan memelihara metadata yang diperlukan untuk menentukan data yang disimpan.

## **2.8. XAMPP**

Menurut Sheldon and Moes (2015) XAMPP merupakan paket program web lengkap yang dapat Anda gunakan untuk mempelajari pemrograman web khususnya PHP dan *MySQL*.

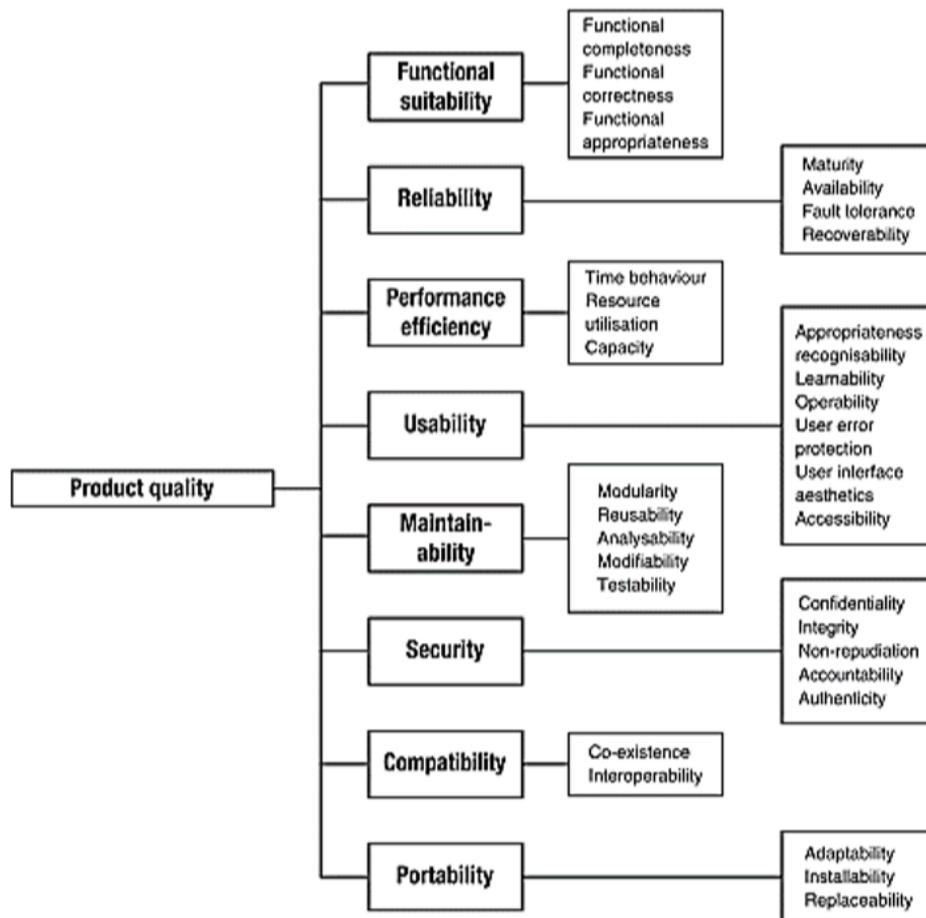
Keunggulan Xampp adalah performa tinggi, stabil, banyak fitur, mudah dikonfigurasi, hanya menggunakan sedikit resource di server, tidak bergantung pada thread untuk melayani klien.

Kekurangan Xampp adalah tidak support IPV6, update / patch terbaru sudah lama, Fast-CGI-nya tidak berfungsi maksimal, penggunaanya tidak sebanyak aktivitas paket program lainnya.

## **2.9. Pengujian *ISO 25010***

ISO/IEC 25010 merupakan bagian dari Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Systems and software quality models. ISO/IEC 25010 adalah pengembangan dari ISO/IEC 9126, serta secara resmi membatalkan dan menggantikan ISO / IEC 9126:2001. Edisi pertama ISO/IEC 25010:2011 yang telah direvisi secara teknis, menjadi

standar internasional terbaru dan relevan untuk menguji sistem informasi yang akan dikembangkan. ISO/IEC 25010 menjadi standar tolak ukur analisis kualitas perangkat lunak yang digunakan oleh perusahaan, instansi, ataupun organisasi. Dengan ISO/IEC 25010 evaluasi kualitas sistem perangkat lunak dapat dilakukan secara spesifik berdasarkan dimensi product quality yang terdiri dari 8 karakteristik yaitu *functional suitability*, *performance efficiency*, *compatibility*, *usability*, *reliability*, *security*, *maintainability*, dan *portability* (Tangkudung *et al.*, 2019).



Sumber : (Tangkudung *et al.*, 2019).

**Gambar 2.2** Model kualitas produk ISO/IEC 25010

Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya:

- 1) **Fungsionalitas (Fungsionalitas).** Kemampuan perangkat lunak berada pada level dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi tertentu, dalam hal ini perangkat lunak dapat memenuhi kelayakan suatu fungsi untuk melakukan pekerjaan tertentu bagi pengguna dan dapat memberikan hasil yang tepat. dan akurasi untuk kebutuhan pengguna tingkat. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu.
  - a. Kelengkapan fungsional, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna tertentu.
  - b. Ketepatan fungsional, sejauh mana produk atau sistem memberikan hasil yang benar sesuai kebutuhan.
  - c. Kesesuaian fungsional, sejauh mana fungsi yang diberikan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.
- 2) **Kompatibilitas,** sejauh mana produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen atau melakukan fungsi lain yang diperlukan secara bersamaan ketika berbagi lingkungan perangkat keras dan perangkat lunak yang sama. Ciri ini terbagi menjadi 2 ciri yaitu.
  - a. Koeksistensi, sejauh mana produk atau sistem dapat melakukan fungsi yang diperlukan secara efisien sambil berbagi sumber daya dengan produk atau sistem lain tanpa merusak produk atau sistem.
  - b. Interoperabilitas, sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.

- 3) *Usability*, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan secara efektif, efisiensi, dan kepuasan tertentu dalam konteks penggunaan. Ciri ini terbagi menjadi beberapa ciri yaitu.
- a. Ketepatan pengenalan, sejauh mana pengguna dapat mengetahui apakah sistem atau produk memenuhi kebutuhan mereka.
  - b. *Learnability*, sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu dengan belajar menggunakan sistem atau produk tersebut secara efisien, efektif, bebas dari risiko dan kepuasan dalam konteks tertentu.
  - c. Operabilitas, sejauh mana produk atau sistem mudah dioperasikan dan dikendalikan.
  - d. Perlindungan kesalahan pengguna, sejauh mana produk atau sistem melindungi pengguna dari membuat kesalahan.
  - e. Estetika antarmuka pengguna, sejauh mana antarmuka pengguna suatu produk atau sistem memungkinkan interaksi yang ramah pengguna dan memuaskan.
  - f. Aksesibilitas, sejauh mana suatu produk atau sistem dapat digunakan oleh semua kelompok untuk mencapai tujuan tertentu sesuai dengan konteks penggunaannya.
- 4) Reliabilitas Merupakan tingkat dimana perangkat lunak dapat bertahan pada tingkat tertentu ketika digunakan oleh pengguna dalam kondisi tertentu, dalam hal ini perangkat lunak dapat beroperasi dan siap pada saat dibutuhkan untuk digunakan dan juga dapat menahan tingkat

kemampuan tertentu. terhadap kegagalan, kesalahan dan pengembalian perangkat lunak sampai tingkat tertentu dalam memulihkan pengembalian data karena kegagalan atau kesalahan perangkat lunak.

Ciri-ciri tersebut terbagi menjadi beberapa sub ciri, yaitu:

- a. Kematangan, sejauh mana suatu produk atau sistem dapat memenuhi kebutuhan dengan andal dalam keadaan normal.
  - b. Ketersediaan, sejauh mana produk atau sistem siap untuk dioperasikan dan dapat diakses pada saat perlu digunakan.
  - c. Toleransi kesalahan, sejauh mana produk atau sistem tetap berjalan sebagaimana mestinya meskipun ada kesalahan pada perangkat keras atau perangkat lunaknya.
  - d. Dapat dipulihkan, sejauh mana produk atau sistem dapat memulihkan data yang terpengaruh secara langsung dan mengatur ulang kondisi sistem sesuai keinginan ketika terjadi gangguan.
- 5) Keamanan, sejauh mana suatu produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan tingkat otorisasi yang dimilikinya. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu:
- a. Kerahasiaan, sejauh mana produk atau perangkat lunak memastikan data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
  - b. Integritas, sejauh mana produk atau perangkat lunak dapat mencegah akses tidak sah untuk mengubah data.

- c. Non repudiation, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan atas peristiwa atau tindakan tersebut.
  - d. Akuntabilitas, sejauh mana tindakan suatu entitas dapat dilacak secara unik ke entitas tersebut.
  - e. Keaslian, sejauh mana identitas subjek atau sumber dapat dibuktikan menjadi yang diklaim.
- 6) Portabilitas, sejauh mana efektivitas dan efisiensi suatu sistem, produk atau komponen dapat dipindahkan dari satu perangkat keras, perangkat lunak atau digunakan dalam lingkungan yang berbeda. Ciri-ciri tersebut terbagi menjadi beberapa ciri yaitu:
- a. Adaptabilitas, sejauh mana produk atau sistem dapat secara efektif dan efisien diadaptasi ke perangkat lunak, perangkat keras, dan lingkungan yang berbeda.
  - b. Penginstalan, sejauh mana produk atau sistem berhasil diinstal atau dihapus di lingkungan tertentu.
  - c. *Replaceability*, sejauh mana suatu produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama di lingkungan yang sama.
- 7) Kinerja terhadap sejumlah sumber daya yang digunakan dalam kondisi tertentu, dalam hal ini efisiensi kinerja dapat memberikan reaksi dan waktu yang diperlukan ketika melakukan suatu tindakan dari suatu fungsi dan perangkat lunak dapat menggunakan jumlah sumber daya saat

melakukan tindakan. dari suatu fungsi. Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu. Ciri ini terbagi menjadi beberapa sub ciri yaitu :

- a. Perilaku waktu, sejauh mana respon dan waktu pemrosesan produk atau sistem dapat memenuhi persyaratan saat menjalankan suatu fungsi.
  - b. Pemanfaatan sumber daya, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem dapat memenuhi persyaratan saat menjalankan fungsinya.
  - c. Kapasitas, sejauh mana batas maksimum produk atau parameter sistem dapat memenuhi persyaratan.
- 8) Maintainability Adalah tingkat dimana perangkat lunak dapat dimodifikasi. Dalam hal ini, modifikasi adalah perbaikan, perubahan atau penyesuaian pada perangkat lunak untuk dapat mengubah lingkungan, persyaratan dan fungsionalitas tertentu. Selain itu, perangkat lunak dapat dianalisis untuk mengetahui penyebab kegagalan perangkat lunak dalam mengidentifikasi bagian-bagian yang dapat dimodifikasi. Sejauh mana efektivitas dan efisiensi suatu produk atau sistem dapat dicapai diobati. Ciri ini terbagi menjadi beberapa sub ciri yaitu.
- a. Modularitas, sejauh mana sistem terdiri dari komponen yang terpisah sehingga perubahan atau modifikasi salah satu komponen tersebut berdampak kecil pada komponen lainnya.
  - b. Dapat digunakan kembali, sejauh mana aset dapat digunakan lebih banyak oleh satu sistem atau digunakan untuk membangun aset lain.

- c. Analisis, tingkat efektivitas dan efisiensi untuk menilai dampak perubahan pada satu atau lebih bagian dari suatu produk atau sistem, untuk mendiagnosis kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan diubah.
- d. Modifiability, sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa mengurangi kualitas produk yang ada.
- e. Testability, tingkat keefektifan dan efisiensi untuk membentuk kriteria pengujian suatu produk, sistem atau komponen dan pengujian dapat dilakukan untuk mengetahui apakah kriteria tersebut telah terpenuhi.

## 2.10. Skala Pengukuran

Skala pengukuran yang digunakan adalah skala likert, yaitu skala yang didasarkan pada penjumlahan sikap responden dalam menanggapi pernyataan terkait dengan indikator suatu konsep atau variabel yang diukur (Sugiyono, 2017). Skala likert pada umumnya menggunakan lima titik dengan label netral di posisi tengah (ketiga). Skala Likert dapat dilihat pada Tabel 2.5.

**Tabel 2. 6 Skala Likert**

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

**Sumber :** (Sugiyono, 2017)

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat dibawah ini

$$\text{Persentase} = \frac{\text{Skor Aktual}}{\text{Skor Ideal}} \times 100\% \dots \dots \dots (1)$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.7.

**Tabel 2.7 Skala Konversi Nilai**

<i>Persentase Pencapaian (%)</i>	<i>Interpretasi</i>
$90 \leq x$	<i>Sangat Baik</i>
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang
$X < 60$	Sangat Kurang

**Sumber :** (Sugiyono, 2017)

Keterangan: x = persentase hasil pengujian.