



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Auji Hamizan dalam penelitiannya tentang sistem informasi penggajian karyawan mengembangkan sebuah aplikasi penggajian karyawan. Pada penelitian tersebut Auji Hamizan mengembangkan sebuah aplikasi penggajian karyawan untuk mempermudah perusahaan dalam manajemen waktu dan keamanan terhadap perhitungan maupun keamanan data penggajian karyawan serta menyajikan informasi yang akurat untuk tiap karyawan. Aplikasi ini juga digunakan untuk meningkatkan efisiensi kerja, kecepatan, dan ketepatan dalam suatu pengambilan keputusan. Pada akhir penelitian tersebut diperoleh hasil bahwa aplikasi tersebut hanya dapat melakukan proses pengolahan data *online* dan mengolah slip gaji *online* [9].

Penelitian serupa dilakukan oleh Riovan Styx dengan membangun sebuah aplikasi penggajian karyawan untuk membantu bagian HRD (*Human Resource Development*) dalam menangani masalah dan mempercepat proses penggajian. Penelitian ini menghasilkan sebuah aplikasi informasi penggajian karyawan berbasis *client-server* dengan menggunakan metode kualitatif untuk mengumpulkan data mengenai faktor-faktor yang mendukung terhadap perancangan sistem. Sistem ini menghasilkan laporan-laporan data gaji yang terdiri dari laporan data gaji seluruh dan laporan data lembur akan tetapi belum terintegrasi dengan aplikasi lain [10].

Pada penelitian lain Victor Marudut Mulia Siregar merancang sebuah aplikasi penggajian berbasis *desktop* untuk membantu dalam pengelolaan data penggajian pegawai sehingga menghasilkan informasi yang valid, membantu dalam memudahkan pencarian informasi penggajian pegawai dengan pencarian informasi yang lebih cepat, dan dengan diterapkannya aplikasi ini kendala tentang keterlambatan maupun ketidak-akuratan laporan-laporan yang berhubungan dengan masalah penggajian pegawai dapat diatasi. Perancangan sistem ini dilakukan dengan dua tahapan yakni membuat diagram konteks dan membuat *ERD* basis data yang akan dirancang. Setelah itu, dilakukan perancangan form dan laporan yang dibutuhkan. Setelah tahap perancangan selesai, maka dilanjutkan dengan tahap implementasi sistem dengan menggunakan *PHP* dan *DBMS MySQL* [11].

Serupa dengan Victor Marudut Mulia Siregar, Cosmas Eko Suharyanto merancang sebuah aplikasi penggajian berbasis *web* dengan pemrograman *PHP* dan *database MySQL* yang mampu mengolah data penggajian sekaligus menyajikan laporan-laporan secara akurat dan cepat serta meminimalkan terjadinya kesalahan akibat kelalaian manusia. Efektifitas dan efisiensi sistem penggajian akan berdampak pada pelayanan rumah sakit yang lebih optimal. Sistem informasi yang telah dirancang mampu menyelesaikan permasalahan yang dihadapi, *DBMS* mampu mengatasi redundansi data dan integrasi tabel-tabel. Bahasa pemrograman *PHP* dengan kolaborasi *MySQL* memungkinkan sistem informasi terintegrasi dan berbasis *web* yang dapat diakses dimana saja dan kapan saja [12].

Penelitian lainnya dilakukan oleh Hamid Kurniawan merancang sistem informasi penggajian yang dapat memudahkan proses pengolahan data gaji dan meminimalisasi kesalahan pada saat pengolahan data gaji sehingga dapat menghasilkan laporan yang lebih akurat dan mengolah data secara efisien. Aplikasi ini menggunakan metode wawancara dan observasi sebagai metode pengumpulan data, *waterfall* sebagai metode pengembangan *software*, dan *software Netbeans* yang digunakan dalam merancang dan membuat sistem informasi pengolahan data gaji. Peneliti sudah melengkapi menu program penggajian akan tetapi menu transaksi belum memiliki rekap lembur yang dapat membatasi perhitungan gaji oleh *user* itu sendiri, oleh karena itu untuk kedepannya diharapkan aplikasi ini akan dilengkapi dengan fasilitas rekap lembur agar lebih memperoleh perhitungan yang akurat dan efisien dalam menggunakan program penggajian yang ada [13].

Sementara itu, pada penelitian ini peneliti bermaksud membangun sistem informasi penggajian karyawan berbasis *web* pada koperasi Primer Koperasi Veteran Republik Indonesia. Aplikasi ini dibangun untuk membantu memudahkan pengelolaan gaji karyawan sehingga kebutuhan informasi tentang gaji pegawai dapat diperoleh secara cepat dan akurat. Sistem ini menggunakan bahasa pemrograman *PHP*, metode pengembangan sistem menggunakan model *spiral*, dan tahapan *testing* menggunakan sistem *black-box*. Dalam aplikasi ini dapat menginputkan data karyawan, data gaji, laporan data presensi karyawan, dan slip gaji karyawan. Aplikasi ini telah dikembangkan dari yang sebelumnya menggunakan *microsoft excel* ke aplikasi berbasis *web*. Sistem ini mencakup data gaji, tunjangan, dan kehadiran. Hasil keluaran dari aplikasi sistem penggajian ini adalah slip gaji karyawan.

Berdasarkan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa tujuan penelitian ini sudah tercapai dan hasil yang diharapkan sudah sesuai.

2.2 Landasan Teori

2.2.1 Sistem

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan [14]. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu [15].

2.2.2 Sistem Informasi

Sistem informasi adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, mengolah serta menyimpan data, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan [16].

2.2.3 Sistem Penggajian

Sistem Penggajian adalah salah satu hal yang berkaitan dengan pengelolaan kesejahteraan tenaga kerja sehingga harus diberi perhatian khusus oleh perusahaan dalam rangka mencapai tujuannya [17]. Untuk penetapan komponen-komponen *payroll* atau daftar gaji umumnya ditetapkan berdasarkan *company policey* masing-masing perusahaan, berikut adalah contoh komponen-komponen *payroll*:

a. Pendapatan

Pendapatan adalah aliran masuk atau kenaikan lain aktiva suatu badan usaha atau pelunasan utang (atau kombinasi dari keduanya) selama suatu periode yang berasal dari penyerahan atau pembuatan barang, penyerahan jasa, atau kegiatan lain yang merupakan kegiatan utama badan usaha [18].

b. Absensi

Absensi adalah sebuah pembuatan data untuk daftar kehadiran yang bisa digunakan bagi sebuah lembaga atau instansi yang sangat perlu membutuhkan sistem seperti ini [19].

c. Tunjangan

Tunjangan adalah suplemen terhadap gaji. Pada dasarnya tunjangan punya fungsi yakni sebagai fungsi sosial maupun sebagai fungsi insentif [20].

d. Potongan

Potongan umumnya diambil disebabkan adanya beberapa hal terkait dengan pegawai. Berbagai macam potongan yang umum diberlakukan pada Primer Koperasi Veteran Republik Indonesia antara lain:

- a. Potongan PKHT Dansos;
- b. Potongan denda.

2.2.4 Pengujian *Usability*

Pada sistem informasi penggajian karyawan dalam penelitian ini menggunakan pengujian *usability*. Aspek *usability* adalah aspek yang untuk melihat keberhasilan sebuah *website* untuk mengetahui bagaimana persepsi pengguna apakah dapat diterima atau tidak dengan kata lain penerimaan pengguna [21]. Uji *usability* digunakan untuk mengetahui suatu ukuran kualitas pengalaman pengguna ketika berinteraksi dengan produk atau sistem (Nielsen,2012) [22].

Uji *usability* pada penelitian ini dilakukan berdasarkan hasil pengisian instrument oleh responden. Instrument terbagi menjadi beberapa butir (item), dimana butir tersebut dikelompokkan menjadi lima (5) variabel sesuai dengan yang diutarakan Nielsen (2012), yang menyebutkan bahwa dalam mengukur *usability*, perlu menggunakan pertanyaan-pertanyaan yang memiliki komponen-komponen tertentu untuk menggali pendapat responden, antara lain *Learnability*, *Efficiency*, *Memorability*, *Errors*, dan *Satisfaction* [23].

Sugiyono (2009) menyebutkan bahwa skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan skala *Likert*, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Jawaban setiap butir instrumen yang menggunakan skala *Likert* mempunyai gradasi dari sangat setuju sampai sangat tidak setuju. Menurut Sugiyono, pada penelitian kuantitatif, jawaban item instrumen dapat diberi skor seperti Tabel 2.1 [24].

Tabel 2. 1 Skala *Likert*

Jawaban	Skor
Sangat Setuju/Selalu/SangatPositif	5
Setuju/Sering/Positif	4
Ragu-ragu/Kadang-kadang/Netral	3
Tidak Setuju/Hampir Tidak Pernah/Negatif	2
Sangat Tidak Setuju/Tidak Pernah/Sangat Negatif	1

Nilai penerimaan *user* atas kemudahan penggunaan sistem dihitung dengan rumus berikut ini:

$$Usability = \frac{\text{Nilai perolehan}}{\text{Nilai maksimal}} \times 100\%$$

Guritno dkk (2011), dalam Latifah (2018), menyebutkan kategori yang dipakai dalam mengelompokan nilai uji *usability*, seperti terlampir pada Tabel 2.2 [25].

Tabel 2. 2 Interpretasi Predikat Sistem

Presentase Hasil Pengujian	Predikat
81 % - 100 %	Sangat Baik
61 % - 80 %	Baik
41 % - 60 %	Cukup Baik
21 % - 40 %	Kurang Baik
0 % - 20 %	Tidak Baik

2.2.5 Blackbox Testing

Black box testing yaitu pengujian yang dilakukan dengan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian ini dianggap paling tepat untuk mewakili penggunaan menu yang dibuat [26].

2.2.6 Pengertian Kuesioner

Kuesioner merupakan alat pengumpulan data primer dengan metode survei untuk memperoleh opini responden. Kuesioner dapat

digunakan untuk memperoleh informasi pribadi misalnya sikap, opini, harapan dan keinginan responden. Idealnya semua responden mau mengisi atau lebih tepatnya memiliki motivasi untuk menyelesaikan pertanyaan ataupun pernyataan yang ada pada kuesioner penelitian. Kuesioner merupakan daftar pertanyaan yang diajukan kepada responden untuk mencari jawaban dari permasalahan yang diteliti [27].

2.2.6 Pengertian Berbasis Web

Berbasis *web* merupakan perangkat lunak yang dapat diakses dengan menggunakan *browser*. *Web* itu sendiri merupakan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi *internet* [28].

2.2.6 Perangkat Lunak (Software)

Perangkat lunak adalah [29]:

- a. Kumpulan instruksi (program komputer) yang jika dieksekusi akan menyediakan fungsi dan daya guna yang diinginkan.
- b. Kumpulan struktur data yang memungkinkan program untuk memanipulasi informasi secukupnya.
- c. Kumpulan dokumen yang menggambarkan operasi dan penggunaan program.

Perangkat lunak tidak hanya berupa program tetapi juga semua dokumen yang berhubungan dan data konfigurasi yang diperlukan agar program beroperasi dengan benar [30].

2.2.7 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sampai pemeliharaan sistem. Rekayasa perangkat lunak berhubungan dengan masalah-masalah praktis dalam memproduksi perangkat lunak. Rekayasa perangkat lunak juga mencakup masalah pemilihan metode yang paling sesuai dengan suatu keadaan [31].

Proses perangkat lunak sebagai kerangka kerja untuk tugas-tugas yang dibutuhkan dalam membangun perangkat lunak dengan kualitas yang baik [32]. Proses perangkat lunak juga meliputi teknologi yang mempopulasikan metode-metode (*method*) teknis, alat-alat bantu (*tools*)

otomatis dan prosedur-prosedur (*procedure*) atau sering disebut dengan proses. Berikut penjelasannya:

a. Metode

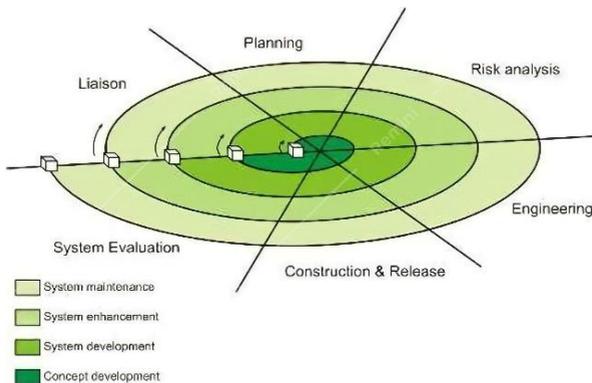
Metode dapat diartikan sebagai cara bagaimana secara teknis membangun perangkat lunak yang harus berada pada sebuah komitmen dasar menuju kualitas.

b. Alat Bantu

Alat bantu atau *tools-tools* perangkat lunak memberikan topangan yang otomatis ataupun semi-otomatis pada proses-proses dalam metode-metode.

c. Prosedur

Prosedur merupakan penggabungan antara metode dan alat bantu. Berikut merupakan prosedur atau urutan (*sequence*) metode yang digunakan oleh peneliti yakni model proses *software Spiral Model* atau *Model Spiral* seperti ditunjukkan pada Gambar 2.1 berikut ini:



Gambar 2.1 *Spiral Model* [33]

Dari gambar *spiral model* di atas, proses dimulai dari inti bergerak searah dengan jarum jam mengelilingi *spiral*. Lintasan pertama putaran menghasilkan perkembangan spesifikasi produk. Putaran selanjutnya digunakan untuk mengembangkan sebuah *prototype*, dan secara progresif mengembangkan versi perangkat lunak yang lebih canggih. Masing-masing lintasan yang melalui daerah perencanaan menghasilkan penyesuaian pada rencana proyek. Biaya dan jadwal

disesuaikan berdasarkan umpan balik yang disimpulkan dari evaluasi pelanggan. Manajer proyek akan menambah jumlah iterasi sesuai dengan yang dibutuhkan. Model *spiral* ini melingkupi aktivitas-aktivitas seperti berikut [33]:

- a. Komunikasi Pelanggan (*Liaison*)
Aktivitas yang dibutuhkan untuk membangun komunikasi yang efektif antara pengembang dengan pengguna terutama mengenai kebutuhan dari pengguna.
- b. Perencanaan (*Planning with Estimation, Scheduling*)
Aktivitas perencanaan ini dibutuhkan untuk menentukan sumberdaya, perkiraan waktu pengerjaan, analisis risiko, dan informasi lainnya yang dibutuhkan untuk pengembangan perangkat lunak.
- c. Analisis Resiko (*Risk Analysis*)
Menganalisis tentang suatu keadaan yang akan terjadi di masa depan dengan cara menentukan suatu keputusan yang akan diambil berdasarkan dari pengamatan dan pertimbangan dari informasi yang telah didapat.
- d. Perencanaan (*Engineering*)
Pembuatan daftar tugas yang nantinya akan dibutuhkan untuk membangun satu atau lebih representasi dari aplikasi yang akan dibangun.
- e. Konstruksi dan Peluncuran (*Construction and Release*)
Membangun sistem dengan berbagai langkah-langkah dalam pelaksanaannya. Seperti membuat basis data, implementasi desain sistem, implementasi pemrograman/*coding*, instalasi, mengkonstruksi, memasang dan memberi pelayanan kepada pelanggan.
- f. Evaluasi Pelanggan (*System Evaluation*)
Aktivitas penyebaran untuk mendapatkan respon dari pelanggan atau mendapat umpan balik berdasarkan evaluasi pelanggan selama representasi perangkat lunak pada tahap konstruksi.

2.2.8 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek merupakan suatu metode pemrograman yang berbasiskan pada objek. Pada pemrograman berorientasi objek, antara objek yang satu berinteraksi dengan objek lainnya dengan tujuan mendapatkan informasi antar objek. Pemrograman berorientasi objek merupakan bagian dari pengembangan sistem berorientasi objek [34].

2.2.9 Basis Data

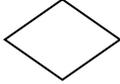
Basis Data merupakan himpunan kelompok data yang saling berhubungan dan disimpan dalam media penyimpanan elektronik setelah melalui proses pengelompokan dan pemilahan data sehingga data disimpan tanpa adanya pengulangan (*redudancy*) yang tidak perlu agar dapat dimanfaatkan kembali dengan cepat [35]. Pengelolaan data bertujuan untuk mempermudah dan mempercepat proses pengambilan data yang telah disimpan.

Basis data tidak akan berguna tanpa adanya pengelola dan penggerakannya karena basis data hanya sebuah objek pasif. Pengelola dan penggerak dalam bisnis data adalah program/aplikasi (*software*). Gabungan dari basis data dan elemen penggerakannya akan membentuk sistem basis data. Sistem basis data merupakan sistem yang terdiri dari kumpulan tabel yang saling berhubungan dan sekumpulan *Database Management System* (DBMS) yang memungkinkan beberapa pengguna untuk mengakses serta memanipulasi tabel-tabel data tersebut [35].

Tabel yang terbentuk merupakan implementasi dari rancangan yang telah dibuat dalam bentuk *Entity Relationship Diagram* (ERD). ERD menunjukkan hubungan antar entitas yang nantinya akan membentuk tabel di dalam DBMS. DBMS bertujuan untuk menyediakan antarmuka (*interface*) kepada pengguna dalam melakukan pengelolaan data [35]. Berikut adalah simbol-simbol ERD dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Simbol Entity Relationship Diagram (ERD)

No	Simbol	Nama	Keterangan
1.		Himpunan <i>Entity</i>	Sesuatu apa saja yang di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data.
2.		Atribut	Sifat atau karakteristik dari tiap-tiap entitas maupun tiap <i>relationship</i> .

No	Simbol	Nama	Keterangan
3.		Himpunan Relasi	Kumpulan semua relasi diantara entitas-entitas yang terdapat dalam himpunan entitas.
4.		<i>Link</i>	Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut.

2.2.10 Unified Modeling Language (UML)

UML (Unified Modeling Language) adalah sebuah bahasa untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. *UML* digunakan untuk merancang model sebuah sistem [36]. Dengan menggunakan *UML* dapat dibuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun [37]. Berikut jenis-jenis *UML*:

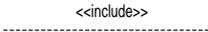
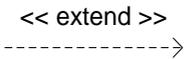
1. Use Case Diagram

Use Case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [38]. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [38]. Berikut simbol-simbol yang sering digunakan dalam *Use Case Diagram* beserta dengan fungsinya dapat dilihat pada Tabel 2.4.

Tabel 2. 4 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Orang proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat

No	Simbol	Nama	Keterangan
			diluar sistem informasi yang akan dibuat itu sendiri.
2.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan menggunakan kata kerja di awali di awal frase nama nama <i>use case</i> .
3.		<i>Interaction</i>	Digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya <i>interaction</i> dilengkapi dengan <i>teks</i> bernama <i>operation signature</i> yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.
4.		<i>Dependency</i>	Merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi

No	Simbol	Nama	Keterangan
			pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah.
5.		<i>Include</i>	Menunjukkan bahwa suatu bagian dari elemen memicu eksekusi bagian dari elemen lain (yang ada pada tanda panah).
6.		<i>Extend</i>	Menunjukkan bahwa suatu bagian dari elemen bisa disisipkan ke dalam elemen yang lain.
7.		<i>Association</i>	Menggambarkan navigasi antar <i>class</i> , berapa banyak obyek lain yang bisa berhubungan dengan satu obyek, dan apakah suatu <i>class</i> menjadi bagian dari <i>class</i> lainnya.
8.		<i>Generalization</i>	Menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan <i>generalization</i> ,

No	Simbol	Nama	Keterangan
			<i>class</i> yang lebih spesifik (<i>subclass</i>) akan menurunkan atribut dan operasi dari <i>class</i> yang lebih umum (<i>superclass</i>). Notasi ini digunakan pada konsep <i>inheritance</i> .

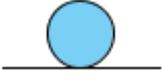
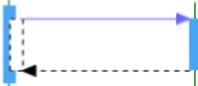
2. Sequence Diagram

Sequence Diagram merupakan diagram *UML* yang termasuk ke dalam *interaction diagram* [39]. *Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek [40]. Untuk menggambar *sequence diagram* harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansi menjadi objek itu [40].

Banyaknya *sequence diagram* yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* telah didefinisikan interaksinya pesan, sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak [40]. Berikut simbol-simbol yang sering digunakan dalam *sequence diagram* beserta dengan fungsinya dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Komponen Sequence Diagram

No	Simbol	Nama	Keterangan
1.		<i>Object Lifeline</i>	Menggambarkan <i>object</i> apa saja yang terlibat.
2.		<i>Actor</i>	Menggambarkan <i>actor</i> yang terlibat.

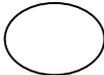
No	Simbol	Nama	Keterangan
3.		<i>Lifeline Control</i>	Digunakan untuk menggambarkan sebuah <i>form</i> .
4.		<i>Lifeline Entity</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.
5.		<i>Lifeline Control</i>	Digunakan untuk menghubungkan <i>boundary</i> dengan tabel.
6.		<i>Time Active</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
7.		<i>Synchronous</i>	Relasi ini digunakan untuk memanggil operasi atau <i>method</i> yang dimiliki oleh suatu objek. <i>Synchronous</i> mengharuskan kita menyelesaikan 1 proses baru kemudian memanggil proses berikutnya.
8.		<i>Asynchronous</i>	Relasi ini digunakan untuk memanggil operasi atau <i>method</i> yang dimiliki oleh suatu objek. <i>Asynchronous</i> memberikan kita fasilitas untuk menjalankan proses lain ketika proses

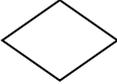
No	Simbol	Nama	Keterangan
			sebelumnya belum selesai.

2.2.11 Flowchart

Flowchart adalah bagian-bagian yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah [41]. Berikut simbol-simbol yang digunakan dalam *Flowchart* beserta dengan fungsinya dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Simbol Flowchart

No	Simbol	Nama	Keterangan
1.		<i>Input/Output</i>	Mempresentasikan <i>input</i> data atau <i>output</i> data yang diproses atau informasi.
2.		Proses	Mempresentasikan operasi.
3.		Penghubung	Keluar ke atau masuk dari bagian lain, khususnya halaman yang sama.
4.		Anak Panah	Mempresentasikan alur kerja.

No	Simbol	Nama	Keterangan
6.		Keputusan	Keputusan dalam program.
9.		<i>Terminal Port</i>	<i>Awal/akhir Flowchart.</i>
11.		Dokumen	<i>Input/Output</i> dalam format yang dicetak.
19.		<i>Manual Operation</i>	Operasi manual.