

BAB II

LANDASAN TEORI

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan di gunakan sebagai sumber acuan yang relavan dan terkini yaitu :

Tabel II.1. Penelitian Terkait

No.	Peneliti Jurnal	Jurnal	Hasil Penelitian
1	Adam Hasbi Nurgroho (2016)	Sistem Pendukung Keputusan Penentuan Nilai Kenaikan Pangkat TNI Angkatan Darat Menggunakan Metode <i>Composite Performance Index</i> (CPI) (Jurnal Matematika, Volume 19, Nomor 3, Desember 2016)	Untuk menentukan penerima bantuan program bedah rumah mengalami kesulitan dalam menentukan siapa yang berhak menerima dan siapa yang tidak. Penerapan Sistem Pendukung Keputusan (SPK) menggunakan metode <i>Composite Performance Index</i> . Kriteria berupa kriteria yang terdapat pada penentuan kenaikan pangkat adalah Kesamaptaan Jasmani, Kepribadian, Kecakapan
2	Budi Santoso (2020)	Penerapan Metode <i>Composite Performance Index</i> (CPI) Dalam Proses Penentuan Penerima Bantuan Program Bedah Rumah Bagi Keluarga Miskin Dikota Lubuklinggau (Jurnal Ilmiah Betrik, Volume.11, No.02, Agustus 2020)	Penelitian ini menghasilkan sebuah sistem pendukung keputusan yang mampu menampilkan hasil dari keputusan untuk pemilihan pemenang tender pengadaan barang/jasa. Kriteria yang digunakan Luas Lantai, Jenis Lantai Yang digunakan, Jenis dinding rumah, Fasilitas BAB yang dimiliki, Sumber penerangan,

			Sumber air minum, Bahan bakar untuk memasak dan Frekuensi makan daging/ayam/ikan dalam satu minggu
3	Nur S Tanjung (2018)	Sistem Pendukung Keputusan Pemilihan Guru Teladan Dengan Menggunakan Metode Composite Performance Index (CPI) (Jurnal Riset Komputer (JURIKOM), Vol. 5 No. 1, Februari 2018)	Pemilihan guru teladan diharapkan dapat meningkatkan semangat, motivasi dan profesionalisme kegiatan pemilihan guru teladan yang terjadi biasanya hanya dengan menggunakan penilaian yang di dasarkan pada analisa pribadi. Hasil dari penelitian ini diharapkan sistem dapat dimanfaatkan pihak sekolah untuk menentukan dengan tepat siapa saja yang berhak menjadi guru teladan dengan di dasarkan pada beberapa aspek yaitu kemampuan bidang studi, tanggung jawab, kreatifitas, komunikasi.
4	Diana (2018)	Implementasi Composite Performance Index pada Multi Criteria Decision Making (MCDM) untuk Memilih Lokasi Usaha UMKM (Jurnal Ilmiah Matrik Vol.20 No.3, Desember 2018:169-178)	Pemilihan lokasi Usaha Mikro Kecil Menengah (UMKM) yang tepat akan memberi pengaruh positif terhadap perkembangan UMKM dan diharapkan dapat memberikan hasil pendapatan yang optimal. Kriteria yang digunakan untuk memilih usaha UMKM adalah biaya sewa tempat, ketersediaan listrik, ketersediaan air bersih, akses jalan, luas lahan parkir , tingkat keamanan, kedekatan dengan suplier, usaha serupa dan kepadatan penduduk implementasi. Composite Performace Index (CPI) untuk memilih

			lokasi menghasilkan lokasi 4 sebagai lokasi terbaik.
5	Andi Anto Tri Susilo, 2017	Penerapan Metode CPI Pada Pemilihan Hotel Di Kota Lubulinggau (JURNAL RESTI, Vol. 1 No. 3 (2017) 204 - 210 ISSN Media Elektronik : 2580-0760)	Penelitian ini menghasilkan sebuah sistem informasi mengenai peringkat hotel yang dapat dijadikan tempat refrensi untuk tetap memperhatikan beberapa kriteria, termasuk tarif kamar, fasilitas dan layanan sedangkan dalam penelitian yang dibahas oleh penulis untuk memilih laptop terbaik dengan kondisi baru sesuai dengan permintaan dari konsumen

Adapun kesimpulan yang diperoleh dari penelitian terdahulu diatas adalah metode *Composite Performance Index* (CPI) dapat mengatasi dalam penentuan Kenaikan Pangkat TNI Angkatan Darat, Penerima Bantuan Program Bedah Rumah, Pemilihan Guru Teladan, pemilihan Memilih Lokasi Usaha UMKM, Pemilihan Hotel Di Kota Lubulinggau. Sedangkan dalam penelitian yang dilakukan oleh peneliti adalah menentukan laptop terbaik untuk masyarakat dengan layar 14 Inchi dengan kondisi baru, sistem yang akan dirancang oleh penulis menggunakan aplikasi PHP dan database MySQL.

II.2. Sistem

Sistem dapat terdiri dari sistem-sistem bagian (*subsystem*). Sebagai misal, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat

keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar, dan kemudian subsistem-subsistem tersebut akan berinteraksi sedemikian rupa sehingga dapat mencapai satu kesatuan yang terpadu. Dalam buku *Analisa dan Design Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu (Han & Goleman, Daniel; Boyatzis, Richard; McKee, 2019)

II.3. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem informasi berbasis komputer yang interaktif, dengan cara mengolah data dengan berbagai model untuk memecahkan masalah-masalah yang tidak terstruktur sehingga dapat memberikan informasi yang bisa digunakan oleh para pengambil keputusan dalam membuat sebuah keputusan. Dalam sebuah sistem pendukung keputusan, sumber daya intelektual yang dimiliki seseorang dipadukan dengan kemampuan komputer untuk membantu meningkatkan kualitas dari keputusan yang diambil. Pengambilan keputusan merupakan sebuah proses memilih sebuah tindakan diantara beberapa alternatif yang ada, sehingga tujuan yang diinginkan dapat tercapai. (Alfina & Harahap, 2019)

Sistem pendukung keputusan (SPK) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang

semiterstruktur dan situasi yang tidak terstruktur. Tujuan dari SPK adalah untuk membantu pengambil keputusan memilih berbagai alternatif keputusan yang merupakan pengolahan informasi-informasi yang diperoleh atau tersedia dengan menggunakan model pengambilan keputusan. Ciri utama sekaligus keunggulan dari sistem pendukung keputusan tersebut adalah kemampuannya untuk menyelesaikan masalah-masalah yang tidak terstruktur. Pengambilan keputusan merupakan proses pemilihan alternatif tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor-faktor yang perlu dipertimbangkan dalam pengambilan keputusan. Tahap-tahap yang harus dilalui dalam proses pengambilan keputusan sebagai berikut :

- a. Tahap Pemahaman (*Intelligence Phase*) Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.
- b. Tahap Perancangan (*Design Phase*) Tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan atau solusi yang dapat diambil. Tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.
- c. Tahap Pemilihan (*Choice Phase*) Pada tahap ini dilakukan pemilihan terhadap berbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar

ditentukan atau dengan memperhatikan kriteria-kriteria berdasarkan tujuan yang akan dicapai. Tahap Implementasi (Implementation Phase) Pada tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan. (Khairina et al., 2016)

II.4. Metode *Composite Performance Index*

Composite Performance Index merupakan indeks gabungan atau *composite index* yang dapat digunakan untuk menentukan penilaian atau peringkat dari berbagai alternatif (i) berdasarkan beberapa kriteria (j) (Nur S Tanjung, 2018).

Penyelesaian CPI dalam mencari alternatif yang layak, yaitu :

1. Identifikasi kriteria tren positif (semakin tinggi nilainya semakin baik) dan tren negatif (semakin rendah nilainya semakin buruk).
2. Untuk kriteria tren positif, nilai minimum pada setiap kriteria ditransformasi ke seratus, sedangkan nilai lainnya ditransformasi secara profesional lebih tinggi.
3. Untuk kriteria tren negatif, nilai minimum pada setiap kriteria ditransformasi ke seratus, sedangkan nilai lainnya ditransformasi secara profesional lebih rendah.

Index yang digunakan untuk menentukan penilaian atau peringkat dari berbagai alternatif keputusan berdasarkan beberapa kriteria dari setiap alternatif, dirumuskan sebagai berikut :

1. Normalisasi Matrix

Normalisasi matrix dilakukan dengan menyesuaikan Nilai Kriteria dengan nilai Bobot yang sudah ditentukan.

2. Menentukan Nilai Kriteria :

$$A_{ij} = X_{ij} / X_{ij}(\min) * 100 \quad ; \quad i = 1,2,\dots, n \text{ dan } j = 1,2,\dots, n$$

3. Menentukan Nilai CPI :

$$I_i = \sum_{j=i}^m A_{ij}B_j \quad ; \quad i = 1,2,\dots, n \text{ dan } j = 1,2,\dots, n$$

Dimana :

A_{ij} = Nilai alternatif ke i pada kriteria ke j

X_{ij} = Nilai awal alternatif ke i pada kriteria ke j

$X_{ij}(\min)$ = Nilai alternatif ke i pada kriteria minimum ke j

B_j = Bobot kepentingan kriteria ke j

I_i = indeks gabungan kriteria pada alternatif ke-i

Prosedur di *Composite Performance Index* disebutkan sebagai berikut :

1. Identifikasi kriteria trenpositif (semakin tinggi nilainya semakin baik) dan tren negatif (semakin rendah nilainya semakin baik).
2. Untuk kriteria tren positif, nilai minimum pada setiap kriteria ditransformasi ke seratus, sedangkan nilai lainnya ditransformasikan secara proporsional lebih tinggi.

3. Untuk kriteria tren negatif, nilai minimum pada setiap kriteria ditransformasi ke seratus, sedangkan nilai lainnya ditransformasikan secara proporsional lebih rendah.
4. Perhitungan nilai alternatif merupakan jumlah dari perkalian antara nilai kriteria dengan bobot kriteria.

II.5. PHP (*Hypertext Preprocessor*)

PHP merupakan kependekan dari kata *Hypertext Preprocessor*. PHP tergolong sebagai perangkat lunak open source yang diatur dalam aturan *general purpose licences* (GPL). Bahasa pemrograman PHP sangat cocok dikembangkan dalam lingkungan *web*, karena PHP bisa diletakkan pada *script* HTML atau sebaliknya (Hikmah, 2015 : 2).

II.6. MySQL (*My Structure Query Language*)

MySQL adalah salah satu *Database Management System (DBMS)* dari sekian banyak *DBMS* seperti *Oracle*, *MS SQL*, *PostgreSQL*, dan lainnya. *MySQL* berfungsi untuk mengolah *Database* menggunakan bahasa *SQL*. *MySQL* bersifat *open source* sehingga bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung dengan *database MySQL* (Hikmah, 2015 : 2).

II.7. Database

Database adalah basis dari data, dengan kata lain database merupakan kumpulan data, dasar yang digunakan untuk menampilkan data atau informasi, sekumpulan data atau informasi teratur berdasarkan criteria tertentu yang saling berhubungan. Selain itu juga database dapat di defenisikan sebagai susunan *record* data operasional lengkap dari suatu organisasi perusahaan, yang terorganisir dan disimpan secara terintegrasi. (Herlina Trisnawati, 2016)

Data adalah penggambaran suatu fakta atau keadaan. Informasi merupakan sekumpulan data yang telah diolah menjadi suatu bentuk sehingga berguna bagi penerima dan pemakai. Dalam dunia komputer *database* bisa dikategorikan bisa sangat spesial karena selalu menjadi hal utama dalam perancangan sistem sustu perusahaan, tentunya ada alasan tertentu mengapa *database* menjadi prioritas sendiri dalam kinerja manejemen perusahaan. (Herlina Trisnawati, 2016)

II.8 Normalisasi

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data relasional yang mengelompokkan atribut dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redudansi data). (Puspitasari et al., 2016)

Tujuan dari normalisasi adalah untuk :

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh *base relations (file)*.
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

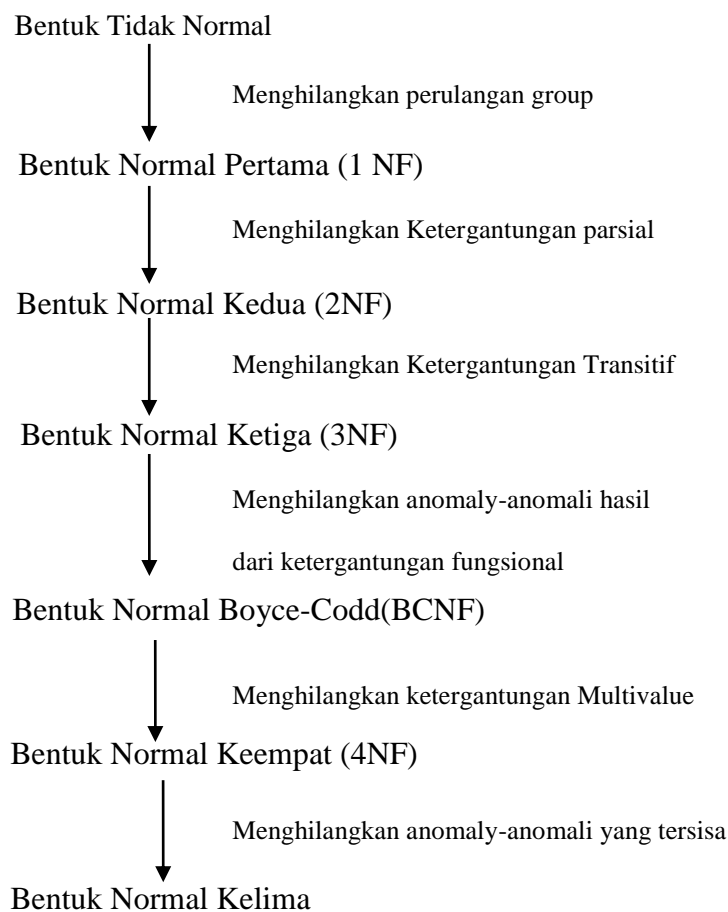
II.8.1. Proses Normalisasi

Gambaran proses normalisasi yaitu:

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat, kemudian.
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal. Untuk melakukan proses tersebut dibutuhkan beberapa tahapan.

Tahapan dalam normalisasi dimulai dari tahap paling ringan (*1NF*) hingga paling ketat (*5NF*). Biasanya hanya sampai pada tingkat *3NF* atau *BCNF* karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik.

Urutan tahapan normalisasi tampak seperti gambar II.1.



Gambar II.1. Tahapan pada Normalisasi
 (Sumber : Dwi Puspitasari, dkk ; 2016)

Adapun aturan dalam normalisasi adalah suatu tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar *Boyce-Codd Normal Form (BCNF)*.

Jika kriteria ketiga (*BCNF*) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (*3rd Normal Form / 3NF*). Pada penelitian ini formula yang dibuat sampai memenuhi bentuk normal ke 3 yaitu *3NF*.

Proses Normalisasi Berdasarkan tahapan normalisasi, terdapat enam bentuk normal yaitu Elmasri & Navathe (2013):

1. Bentuk Normal Tahap Pertama (*1st Normal Form / 1NF*).
2. Bentuk Normal Tahap Kedua (*2nd Normal Form / 2NF*).
3. Bentuk Normal Tahap (*3rd Normal Form / 3NF*).
4. *Boyce-Code Normal Form (BCNF)*.
5. Bentuk Normal Tahap (*4th Normal Form / 4NF*) 6. Bentuk Normal Tahap (*5th Normal Form / 5NF*)

Bentuk Normal Pertama / *1NF*, memiliki aturan:

- a. Tidak adanya atribut *multi-value*, atribut komposit atau kombinasinya.
- b. Mendefinisikan atribut kunci.
- c. Setiap atribut dalam tabel tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi).

Bentuk Normal Kedua / *2NF*, memiliki aturan :

- a. Sudah memenuhi dalam bentuk normal kesatu (*1NF*).
- b. Semua atribut bukan kunci hanya boleh tergantung (*functional dependency*) pada atribut kunci. Jika ada ketergantungan parsial maka atribut tersebut harus dipisah pada tabel yang lain.

- c. Perlu ada tabel penghubung ataupun kehadiran foreign key bagi atribut-atribut yang telah dipisah tadi.

Bentuk Normal Ketiga / *3NF*, memiliki aturan:

- a. Sudah memenuhi dalam bentuk normal kedua (*2NF*)
- b. Tidak ada ketergantungan transitif (dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya). (Puspitasari et al., 2016)

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

UML (Unified Modelling Language) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena *UML* menyediakan Bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*Sharing*) dan mengkomunikasikan rancangan dengan baik. (Helmi Kurniawan : 2015)

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi.

Unified Modeling Language (UML) biasa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Alfina & Harahap, 2019)

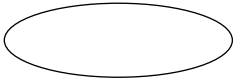
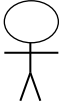


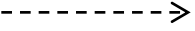
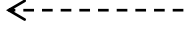
UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industry yang merupakan standar bahasa pemodelan umum dalam industri perangkat dan pengembangan sistem. (Urva et al., 2015)

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah Empat alat bantu sebagai berikut: *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*.

1. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan Use Case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Symbol-simbol yang digunakan dalam Use Case Diagram Yaitu

Tabel II.4. Simbol Use Case



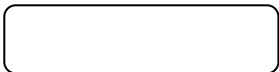
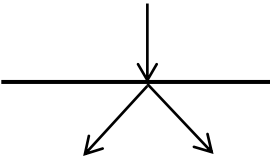
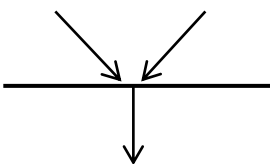
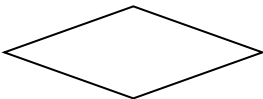

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva, dkk ; 2015)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol- simbol yang digunakan dalam *activity diagram* yaitu:

Tabel II.5. Simbol *Activity Diagram*

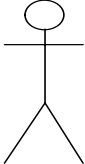
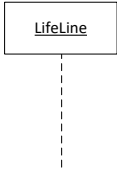

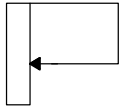
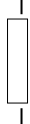

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau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

(Sumber : Gellysa Urva,dkk ; 2015)

3. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antara objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel II.6. Sequence Diagram

Gambar	Keterangan
 <p>Actor</p>	Orang, Proses atau sistem lain yang berinteraksi dengan sistem informasi.
	Merepresentasikan entitas tunggal dalam sequence diagram, digambarkan dengan kotak. Entitas ini memiliki nama, <i>Stereotype</i> atau berupa <i>instance</i> .
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gellysa Urva, dkk ; 2015)

4. Class Diagram (Diagram Kelas)

Merupakan hubungan antara kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan

tanggung jawab entitas yang merupakan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class Diagram secara khas meliputi: Kelas (*Class*), *Relasi*, *Associations*, *Generalization dan Aggregation*, Atribut (*Attributes*), Operasi (*Operations/ Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.7. Multiplicity Class Diagram

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

(Sumber : Gellysa Urva, dkk ; 2015)