

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Damayanti dan Mesran (2020) mengenai sistem pendukung keputusan penerimaan member merah untuk pedagang yang layak pada indogrosir dengan menggunakan metode ARAS, Damayanti dan Mesran menyimpulkan bahwa penerepan metode ARAS dilakukan dengan cara menghitung nilai alternatif berdasarkan algoritma Aras dan hasilnya akan diperoleh pada nilai minimumnya.

Berdasarkan penelitian yang dilakukan oleh Susanto (2018) mengenai Penerapan Metode *Additive Ratio Assessment* (ARAS) Dalam Pendukung Keputusan Pemilihan Susu *Gym* Terbaik Untuk Menambah Masa Otot, Susanto menyimpulkan bahwa dalam proses perhitungan untuk menghasilkan suatu keputusan yang tepat dan akurat dengan membandingkan kriteriayang ada dari beberapa merk susu *gym* yang di bahas.

Berdasarkan penelitian yang dilakukan oleh Sitompul dan Hasibuan (2018) mengenai Sistem Pendukung Keputusan Seleksi Tenaga Kerja Untuk *Security Service* Menggunakan Metode ARAS, Sitompul dan Hasibuan menyimpulkan bahwa proses seleksi tenaga kerja *security* yang dilakukan oleh PT. ISS harus memenuhi beberapa kriteria yang ditentukan oleh pihak perusahaan.

Berdasarkan penelitian yang dilakukan oleh Simarmata, dkk (2019) mengenai Implementasi Metode *Additive Ratio Assesment* (ARAS) Untuk

Rekomendasi Pasien Kunjungan Sehat Pada Fasilitas Kesehatan Tingkat Pertama
Dr Josepb Nugroho H. S. Simarmata, dkk, menyimpulkan bahwa sistem mampu memberikan rekomendasi pasien kunjungan sehat dengan menggunakan metode ARAS berdasarkan data-data yang dimiliki FKTP dr. Josepb Nugroho H.S. Berdasarkan data bulan Juni 2019, maka terdapat 1674 pasien yang direkomendasikan, dengan pasien bernama Aswar Fahmi sebagai prioritas pertama karena memiliki nilai preferensi tertinggi yaitu satu.

Berdasarkan penelitian yang dilakukan oleh Addenan dan Susanto (2021) mengenai Penerapan Metode *Rank Order Centroid* dan *Additive Ratio Assessment* Pada Aplikasi Rekomendasi *Supplier*, Addenan dan Susanto menyimpulkan bahwa pemilihan rekomendasi *supplier* dalam memenuhi kebutuhan perusahaan sudah dapat menggunakan sistem pendukung keputusan berbasis komputer sehingga tidak memakan waktu yang lama dalam proses penilaiannya karena tidak dilakukan secara manual dalam proses penilaiannya dan hasil yang di dapat juga maksimal karena dilakukan oleh sistem.

II.2. Landasan Teori

Landasan teori peneliti kutip dari beberapa teori yang berkaitan dengan penelitian ini yaitu:

II.2.1. Penerapan

Penerapan (implementasi) merupakan suatu proses penerapan ide, konsep, kebijakan atau inovasi dalam suatu tindakan praktis sehingga memberikan dampak baik berupa perubahan pengetahuan, keterampilan maupun nilai dan

sikap. Pengertian implementasi secara sederhana implementasi diartikan sebagai pelaksanaan atau penerapan. Implementasi merupakan aktivitas yang saling menyesuaikan. Implementasi merupakan sistem rekayasa. Pengertian-pengertian tersebut memperlihatkan bahwa kata implementasi bermuara pada aktivitas, adanya aksi, tindakan atau mekanisme suatu sistem. Ungkapan mekanisme mengandung arti bahwa implementasi bukan sekedar aktivitas tetapi suatu kegiatan yang terencana dan dilakukan secara sungguh-sungguh berdasarkan acuan norma tertentu untuk mencapai tujuan kegiatan (Magdalena, dkk, 2021 : 120).

Penerapan (implementasi) merupakan proses penterjemahan dari pemodelan menjadi bentuk aplikasi sesuai dengan kebutuhan informasi pengguna. Proses implementasi adalah proses penterjemahan dari pemodelan ke dalam pengkodean atau pembentukan antarmuka (Oktaviani dan Sauda, 2019 : 182).

II.2.2. Sistem Pendukung Keputusan (SPK)

Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan manipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tidak seorang pun tahu secara pasti bagaimana keputusan itu seharusnya dibuat. Sistem pendukung keputusan biasanya dibangun untuk mendukung solusi atas suatu masalah atau mengevaluasi suatu peluang. Sistem pendukung keputusan merupakan sistem

informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data (Azmi, 2020 : 160).

II.2.2.1. Karakteristik Utama Sistem Pendukung Keputusan

Dari pengertian Sistem Pendukung Keputusan maka dapat ditentukan karakteristik antara lain:

1. Sistem yang berbasis komputer.
2. Dipergunakan untuk membantu para pengambil keputusan.
3. Melalui cara simulasi yang interaktif.
4. Dimana data dan model analisis sebagai komponen utama (Labolo, 2020 : 32).

II.2.2.2. Proses Pengambilan Keputusan

Pengambilan keputusan meliputi beberapa tahap dan melalui beberapa proses, pengambilan keputusan meliputi empat tahap yang saling berhubungan dan berurutan. Empat proses tersebut adalah:

1. *Inteligenci*

Tahap ini merupakan proses penelusuran dan pendekatan dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diperoses, dan diuji dalam rangka mengidentifikasikan masalah.

2. *Design*

Tahap ini merupakan proses menemukan dan mengembangkan alternatif. Tahap ini meliputi proses untuk mengerti masalah, menurunkan solusi dan menguji kelayakan solusi.

3. *Choice*

Pada tahap ini dilakukan proses pemilihan di antara berbagai alternatif tindakan yang mungkin dijalankan. Tahap ini meliputi pencarian, evaluasi, dan rekomendasi solusi yang sesuai untuk variabel hasil pada alternatif yang dipilih.

4. *Implementation*

Tahap implementasi adalah tahap pelaksanaan dari keputusan yang telah di ambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana, sehingga hasil keputusan dapat dipantau dan disesuaikan apabila diperlukan perbaikan. Dalam hal ini, model simon juga menggambarkan kontribusi Sistem Informasi Manajemen (SIM) dan Ilmu Manajemen/*Operations Research* (IM/OR) terhadap proses pengambilan keputusan (Panggabean, 2016 : 2).

II.2.2.3. Karakteristik Sistem Pendukung Keputusan

Karakteristik dan kapabilitas kunci dari Sistem Pendukung Keputusan adalah sebagai berikut:

1. Dukungan untuk pengambilan keputusan, terutama pada situasi semiterstruktur dan tak terseteruktur.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer.
3. Dukungan untuk individu dan kelompok.
4. Dukungan untuk semua keputusan independen dan sekuensial.

5. Dukungan di semua *fase* proses pengambilan keputusan: inteligensi, desain, pilihan, dan implementasi.
6. Dukungan pada berbagai proses dan gaya pengambilan keputusan.
7. Kemampuan sistem beradaptasi dengan cepat dimana pengambilan keputusan dapat menghadapi masalah-masalah dengan cara mengadaptasi sistem terhadap kondisi-kondisi perubahan yang terjadi.
8. Pengguna merasa seperti rumah. *User-friendly*, kapabilitas grafis yang kuat, dan sebuah bahasa interaktif yang alami.
9. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, *timelines*, kualitas) dari pada efisiensi (biaya).
10. Pengambil keputusan mengontrol penuh semua langkah proses pengambilan keputusan dalam memecahkan masalah.
11. Pengguna akhir dapat mengembangkan dan memodifikasi sistem sederhana.
12. Menggunakan model-model dalam penganalisisan situasi pengambilan keputusan.
13. Disediakkannya akses untuk berbagai sumber data, format, dan tipe, mulai dari Sistem informasi geografis (GIS) sampai sistem berorientasi objek.
14. Dapat dilakukan sebagai alat *standalone* yang digunakan oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan di satu organisasi keseluruhan dan di beberapa organisasi sepanjang rantai persediaan (Panggabean, 2016 : 2).

II.2.2.4. Komponen-Komponen Sistem Pendukung Keputusan

Sistem Pendukung Keputusan terdiri dari empat subsistem, yaitu:

1. Manajemen Data, meliputi basis data yang berisi data-data yang relevan dengan keadaan dan dikelola oleh perangkat lunak yang disebut dengan *Database Management System* (DBMS).
2. Manajemen Model berupa sebuah paket perangkat lunak yang berisi model-model *financial*, *statistic*, *management science*, atau model kuantitatif, yang menyediakan kemampuan analisa dan perangkat lunak manajemen yang sesuai.
3. Subsistem Dialog atau Komunikasi, merupakan subsistem yang dipakai oleh *user* untuk berkomunikasi dan member perintah (menyediakan *user interface*).
4. Manajemen *knowledge* yang mendukung subsistem lain atau sebagai komponen yang berdiri sendiri (Panggabean, 2016 : 3).

II.2.2.5. Komponen Besar Sistem Pendukung Keputusan

Secara umum DSS dibangun oleh tiga komponen besar yaitu:

1. *Database Management*.
2. *Model Base*.
3. *Software System/User Interface* (Labolo, 2020 : 32).

II.2.3. Metode Additive Ratio Assessment (ARAS)

Metode *Additive Ratio Assessment* (ARAS) adalah sebuah metode yang digunakan untuk perbandingan kriteria, secara konsep metode ARAS ini

digunakan dengan metode lain yang menggunakan konsep perangkingan seperti SAW atau TOPSIS, dimana proses penentuan ranking harus di olah kembali dengan menggunakan metode ARAS sehingga hasil rangking dengan metode SAW dan metode SAW+ARAS bisa berberda hasilnya. Langkah-langkah dalam melakukan proses perangkingan dengan metode ARAS sebagai berikut:

1. Pembentukan *Decisio Making Matrik*

$$X = \begin{pmatrix} X_{01} & X_{0j} & \dots & X_{0n} \\ X_{11} & X_{1j} & \dots & X_{1n} \\ \dots & \dots & \dots & \dots \\ X_{n1} & X_{nj} & \dots & X_{nn} \end{pmatrix} \quad (i=0, m=1, \dots, n)$$

Dimana

m = Jumlah Alternatif

n = Jumlah Kriteria

X_{ij} = Nilai Perfoma Dari Alternatif i Terhadap Kriteria j

X_{0j} = Nilai Optimum Dari Kriteria j

Jika nilai optimal kriteria j (X_{0j}) tidak diketahui, maka :

$$X_{0j} = \frac{Max}{X_{ij}} \quad \text{Jika Benefit}$$

$$X_{0j} = \frac{Min}{X_{ij}} \quad \text{Jika Cost}$$

2. Penormalisasian matriks keputusan untuk semua kriteria.

Jika kriteria Benefical (max), maka dilakukan normalisasi mengikuti :

$$X_{ij}^* = \frac{X_{ij}}{\sum_{i=0}^m X_{ij}}$$

Dimana X_{ij}^* adalah nilai normalisasi.

Jika kriteria non Benefical (min), maka dilakukan normalisasi :

$$X_{ij} = \frac{1}{X_{ij}}$$

$$R = \frac{X_{if}}{\sum_{i=0}^m X_{ij}}$$

Menentukan bobot matriks yang sudah dilakukan normalisasi.

$$D = [D_{ij}] m \times n R_{ij} \cdot W_j$$

Dimana W_j = Bobot

3. Menentukan nilai fungsi optimalisasi (S_i)

$$S_i = \sum_{j=1}^n D_{ij} : (i = 1, 2, \dots, m; j = 1, 2, \dots, n)$$

Dimana S_i adalah nilai fungsi optimalis alternatif i . Nilai terbesar adalah nilai terbaik, dan nilai yang paling sedikit adalah yang terburuk. Dengan memperhitungkan proses hubungan proposional dengan nilai bobot dan kriteria yang diketahui berpengaruh pada hasil akhir.

4. Menentukan tingkatan peringkat tertinggi alternatif

$$K_i = S_i / S_0$$

Dimana S_i dan S_0 merupakan nilai kriteria optimalitas (Sipayung, 2020 : 485).

II.2.4. Rekomendasi

Secara harfiah pengertian dari rekomendasi adalah saran yang menganjurkan atau membenarkan atau menguatkan. Sistem rekomendasi adalah suatu perangkat piranti lunak dan teknik yang memberikan saran untuk sesuatu yang dapat dimanfaatkan oleh pengguna, rekomendasi yang diberikan bisa berupa informasi, layanan atau produk yang sudah diprediksi sebelumnya. Sebuah

perangkat lunak yang dibangun guna membantu pengguna dengan memberikan rekomendasi saat pengguna dihadapkan pada jumlah informasi yang besar merupakan garis besar yang menjelaskan sistem rekomendasi. Sistem rekomendasi sebagai model aplikasi dari hasil observasi terhadap keadaan dan keinginan pelanggan (Hidayatullah, dkk, 2020 : 20).

II.2.5. Produk

Produk adalah seperangkat atribut baik berwujud, termasuk di dalamnya masalah warna, harga, nama baik pabrik, nama baik tokoh yang menjual (pengecer), dan pelayanan pabrik serta pelayanan pengecer, yang diterima oleh pembeli guna memuaskan keinginannya. Produk adalah segala sesuatu yang dapat ditawarkan di pasar, untuk memuaskan kebutuhan dan keinginan konsumen. Produk terdiri atas barang, jasa, pengalaman, *events*, orang, tempat, kepemilikan, organisasi, informasi dan ide. Jadi produk itu bukan hanya berbentuk suatu yang berwujud saja, seperti makanan, pakaian, dan sebagainya akan tetapi juga sesuatu yang tidak berwujud seperti pelayanan jasa. Semua diperlukan bagi pemuasan kebutuhan dan keinginan (*need and wants*) dari konsumen. Konsumen tidak hanya membeli produk sekedar memuaskan kebutuhan (*need*), akan tetapi juga bertujuan memuaskan keinginan (*wants*). Misalnya membeli bentuk sepatu, gaya warna, merek, dan harga yang menimbulkan/mengangkat *prestige* (Hasanudin, 2018 : 28).

II.2.6. *Hypertext Preprocessor (PHP)*

PHP merupakan singkatan dari “*Hypertext Preprocessor*”. PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrograman C, Java, ASP dan Perl ditambah beberapa fungsi PHP yang Spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan *web* menjadi lebih dinamis, dengan PHP anda bisa menampilkan atau menjalankan beberapa *file* dalam 1 *file* dengan cara di *include* dan *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa *database* walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle (Rahmasari, 2019 : 415).

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted (FI)*, yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script/program* tersebut akan dijalankan/diproses oleh *server*. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari (Sugijanto, dkk, 2020 : 2).

II.2.7. *Hyper Text Markup Language (HTML)*

HTML adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang *user* untuk membuat dan menyusun bagian paragraf, *heading*, *link* atau tautan, dan *blockquote* untuk halaman *web* dan aplikasi. HTML bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan *user* untuk mengorganisir dan memformat dokumen, sama seperti *Microsoft Word* (Sugijanto, dkk, 2020 : 2).

HTML ialah kepanjangan dari *Hypertext Markup Language*. Definisi HTML adalah bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui *tag-tag* yang ditulis dalam HTML (Rahmasari, 2019 : 415).

II.2.8. MySQL

Definisi MySQL merupakan *software RDMS (Relational Database Management System)* yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat di akses oleh banyak pengguna dan dapat melakukan suatu proses secara sinkron atau bersamaan (Rahmasari, 2019 : 415).

MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database* pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis

dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL (Sugijanto, dkk, 2020 : 2).

II.2.9. *Unified Modeling Language* (UML)

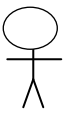
UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Andikos, 2019 : 39).

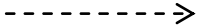
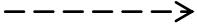
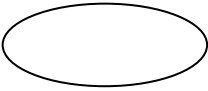
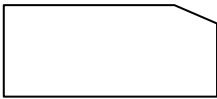
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol *Use Case*

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

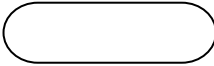
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber : Andikos, 2019 : 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II.2. Simbol *Activity Diagram*

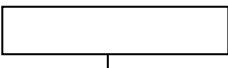
Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber : Andikos, 2019 : 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.

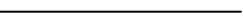
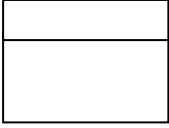
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

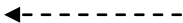
(Sumber : Andikos, 2019 : 39)

4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Assocation</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Andikos, 2019 : 39)