

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Rika Rosnelly, dkk (2021) mengenai “*Autoregressive Integrated Moving Average (ARIMA-Box Jenkins)* Pada Peramalan Komoditas Cabai Merah di Indonesia”. Dalam melakukan penelitian ini metode *time series* sering digunakan dalam memprediksi peramalan harga cabai, sehingga sering dilakukan untuk meramalkan harga cabai untuk di masa yang akan datang. Penelitian penulis menggunakan metode *Autoregressive Integrated Moving Average (ARIMA)* untuk memprediksi jumlah peserta didik di MTs. Swasta Tahfidzul Qur'an Nurul Azmi.

Berdasarkan penelitian yang dilakukan oleh Atmaja, dkk (2021) mengenai Peramalan Jumlah Penjualan Buku Menggunakan Metode *Autoregressive Integrated Moving Average (ARIMA)* Pada Toko Buku AGP Gramedia, Atmaja, dk menggunakan metode ARIMA untuk meramalkan jumlah penjualan buku. Metode ARIMA ini merupakan metode yang bisa mengatasi berbagai masalah peramalan maka penelitian ini menggunakan metode ARIMA untuk mengatasi masalah peramalan penjualan buku pada Toko Buku AGP Gramedia. Penelitian penulis menggunakan metode *Autoregressive Integrated Moving Average (ARIMA)* untuk memprediksi jumlah peserta didik di MTs. Swasta Tahfidzul Qur'an Nurul Azmi.

Berdasarkan penelitian yang dilakukan oleh Tan dan Astuti (2020) mengenai Metode *Autoregressive Integrated Moving Average* (ARIMA) untuk meramalkan penjualan. Kondisi *sales performance* dipengaruhi oleh *sales budget* (anggaran penjualan) yang telah ditetapkan di periode sebelumnya. Penetapan *sales budget* berdasarkan pada hasil ramalan penjualan (*sales forecast*) yang sudah dianalisa sebelumnya. Penetapan *sales forecast* di PT. Hibex Indonesia sebelumnya menggunakan metode kualitatif dimana ramalan dilakukan berdasarkan pada asumsi-asumsi yang bersifat subyektif baik dari segi internal maupun eksternal digabung data historis. Penelitian ini bertujuan untuk menerapkan ramalan penjualan di PT. Hibex Indonesia menggunakan metode kuantitatif. Penelitian menggunakan metode ARIMA dengan software Eviews 9.0 berdasarkan data historis sebelumnya, yaitu data penjualan periode Juli 2014 – Desember 2017. Penelitian penulis menggunakan metode *Autoregressive Integrated Moving Average* (ARIMA) untuk memprediksi jumlah peserta didik di MTs. Swasta Tahfidzul Qur'an Nurul Azmi menggunakan data tahun 2018-2020.

Berdasarkan penelitian yang dilakukan oleh Yunita (2019) mengenai peramalan jumlah penggunaan kuota internet menggunakan metode ARIMA. Tujuan dari penelitian ini yaitu untuk memprediksi jumlah penggunaan kuota internet dengan menggunakan metode *Autoregressive Integrated Moving Average* (ARIMA). Penelitian penulis menggunakan metode *Autoregressive Integrated Moving Average* (ARIMA) untuk membuat aplikasi yang dapat memprediksi jumlah peserta didik di MTs. Swasta Tahfidzul Qur'an Nurul Azmi.

Berdasarkan penelitian yang dilakukan oleh Zulhamidi dan Hardianto (2020) mengenai peramalan penjualan teh hijau dengan metode ARIMA (studi kasus pada PT. MK), Zulhamidi dan Hardianto menggunakan metode ARIMA untuk meramalkan penjualan teh hijau. Data yang digunakan adalah data periode Januari 2012 sampai dengan Desember 2016, dengan *software* Minitab sebagai alat bantu perhitungan. Penelitian penulis menggunakan metode *Autoregressive Integrated Moving Average* (ARIMA) untuk memprediksi jumlah peserta didik di MTs. Swasta Tahfidzul Qur'an Nurul Azmi dan membuatnya dengan pemrograman *web*.

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 (Ina Magdalena, dkk, 2021).

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).

II.2.2. Metode ARIMA

Metode ARIMA merupakan metode peramalan dengan menggunakan serangkaian data masa lalu yang digunakan untuk mengamati terhadap suatu kejadian, peristiwa, atau suatu variabel pada data tersebut. Kelebihan pada metode ARIMA ini adalah dapat digunakan untuk menganalisis situasi yang acak, tren, musim bahkan sifat siklis dalam deret waktu yang dianalisis. (Adri Arisena, 2020)

Metode *Autoregressive Integrated Moving Average* (ARIMA) yang biasa disebut dengan metode *Box-Jenkins* merupakan metode yang dikembangkan oleh George Box dan Gwilym Jenkins pada tahun 1970. Metode ARIMA (*Autoregressive Integrated Moving Average*) adalah metode yang digunakan untuk peramalan jangka pendek. Penggunaan metode ARIMA dalam peramalan jangka pendek sangat tepat digunakan karena metode ARIMA memiliki ketepatan yang sangat akurat. Dan juga menentukan hubungan statistik yang baik antar variabel yang akan diramal dengan nilai yang digunakan untuk peramalan. Sedangkan untuk peramalan jangka panjang ketepatan peramalannya kurang baik. Biasanya nilai peramalan akan cenderung konstan untuk periode yang cukup panjang. Model

Autoregressive Integrated Moving Average (ARIMA) adalah model yang secara penuh mengabaikan variabel independen dalam membuat peramalan. Nilai yang digunakan oleh ARIMA untuk peramalan yaitu menggunakan nilai masa lalu dan sekarang dari variabel dependen untuk menghasilkan peramalan jangka pendek yang akurat. (Yunita, 2019).

Model ARIMA diaplikasikan untuk deret waktu yang mempunyai mean atau varians yang tidak konstan, atau mempunyai kedua sifat tersebut. Sifat ini disebut sebagai proses yang tidak stasioner. (Paridi, 2018).

Autoregressive integrated moving average (ARIMA) adalah metode peramalan yang digunakan untuk deret waktu yang memiliki variasi yang tidak tetap atau memiliki dua sifat yaitu mean dan varians. ARIMA juga adalah kombinasi dari dua metode yaitu *autoregressive* dan *moving average*. *Autoregressive* merupakan peramalan menggunakan deret waktu berdasarkan periode pada kasus yang terjadi. *Moving average* merupakan peramalan dengan nilai rata-rata jumlah dari beberapa periode waktu.

Persamaan matematika dari metode ARIMA dengan kombinasi dua metode yaitu *autoregressive* dan *moving average* yaitu:

Autoregressive:

$$AR = \Phi_1.P_t - 1 + \Phi_2.P_t - 2 + \Phi_n.P_t - n + e_t \quad (1)$$

Dimana:

AR : *Autoregressive*

Φ_n : Koefisien parameter

P_{t-n} : Nilai Data

e_t : Nilai Error [13].

Untuk mencari koefisien parameter maka digunakan rumus:

$$\Phi_n = \frac{P_{t-n}}{P_{t-n-1}} \quad (2)$$

Untuk mencari nilai error maka digunakan rumus:

$$e_t = (1 - \Phi_1 \cdot B) \quad (3)$$

Dimana B adalah jumlah periode.

Moving Average:

$$MA = \frac{P_t + P_{t-1} + P_{t-n}}{n} \quad (4)$$

Dimana:

MA: *Moving Average*

P_t : Nilai Data

n : Jumlah Data.

ARIMA:

$$H = P_t - \Phi_1 \cdot P_{t-1} + \Phi_2 \cdot P_{t-1} + \Phi_n \cdot P_{t-n} + e_t - \left(\frac{1}{n}\right) \cdot P_t + P_{t-1} + P_{t-n} \quad (5)$$

Dimana:

H_t : Hasil Ramal ARIMA

Φ_n : Koefisien parameter

P_{t-n} : Nilai Data

e_t : Nilai Error

P_t : Nilai Data

n : Jumlah Data. (Atmaja, dkk, 2021).

II.2.3. Prediksi

Prediksi merupakan tindakan memperkirakan sebuah nilai di masa depan berdasarkan nilai yang terdapat pada saat ini. (Atmaja, dkk, 2021).

Prediksi adalah proses untuk meramalkan suatu variabel di masa mendatang dengan berdasarkan pertimbangan data pada masa lampau. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin dengan kejadian. (Evi Sinaga, 2018)

Prediksi adalah input dasar dalam proses pengambilan keputusan manajemen operasi pada menaruh liputan tentang permintaan di masa mendatang menggunakan tujuan buat memilih berapa kapasitas atau persediaan yang dibutuhkan buat menciptakan keputusan *staffing*, *budget* yang wajib disiapkan, pemesanan barang berdasarkan *supplier* dan partner menurut rantai pasok yang diharapkan membuat suatu perencanaan. (Ngantung dan Jan, 2019).

II.2.4. Tujuan Prediksi

Tujuan Prediksi adalah untuk meramalkan permintaan dan *item-item independent demand* di masa yang akan datang, sedangkan menurut Subagyo (2002), tujuan Prediksi adalah mendapatkan Prediksi yang bisa meminimalkan kesalahan (*error*) yang bisa diukur dengan *Mean Square Error* (MSE). Dengan adanya Prediksi penjualan berarti manajemen perusahaan telah mendapatkan gambaran perusahaan dimasa yang akan datang, sehingga manajemen perusahaan memperoleh masukan yang sangat berarti dalam menentukan kebijaksanaan perusahaan. (Zulhamidi dan Hardianto, 2020).

II.2.5. Pendekatan Prediksi

Terdapat dua pendekatan untuk melakukan Prediksi, yaitu dengan pendekatan kualitatif dan pendekatan kuantitatif.

1. Metode Prediksi kualitatif yang menggabungkan faktor-faktor seperti intuisi pengambilan keputusan, emosi, dan pengalaman pribadi.
2. Metode Prediksi kuantitatif yang menggunakan satu atau lebih model matematis dengan data masa lalu dan variabel sebab akibat untuk meramalkan permintaan.

Metode Prediksi kuantitatif juga dapat dikelompokkan menjadi dua jenis, yaitu model deret waktu/*time series* dan model kausal/*explanatory*. (Yunita, 2019).

II.2.6. *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).

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).

II.2.7. *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).

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).

II.2.8. *My Structure Query Language (MySQL)*

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan data dengan cepat dengan menggunakan perintah-perintah SQL. MySQL adalah sebuah program *server database* yang bersifat gratis (MySQL *Community Server*). (Wijaya, 2018).

Salah satu contoh *database management system* adalah MySQL. MySQL adalah *database open source* terpopuler di dunia. Dengan kinerja, kehandalan dan kemudahan penggunaan yang terbukti, MySQL telah menjadi pilihan *database* terdepan untuk aplikasi berbasis *web*, yang digunakan oleh properti *web* profil tinggi termasuk Facebook, Twitter, YouTube, Yahoo! dan banyak lagi. Kata “SQL” dari “MySQL” adalah singkatan dari “*Structured Query Language*”. SQL adalah bahasa standar yang paling umum digunakan untuk mengakses database. Bergantung pada lingkungan pemrograman yang kita pakai, kita bisa memasukkan SQL secara langsung (misalnya, untuk *men-generate* laporan), memasukkan pernyataan SQL ke dalam kode yang ditulis dalam bahasa lain, atau menggunakan API khusus yang dapat menyembunyikan sintaks SQL. (Afrianto dan Nugrahanti, 2018).

II.2.9. *Word Elektrik Browser (Web)*

Web merupakan salah satu aplikasi untuk mengelola data *online* yaitu yang terhubung dalam jaringan internet. *Web* adalah halaman situs sistem informasi yang dapat diakses secara cepat. *Web* adalah pengelolaan data sesuai dengan layar pengguna dan dikembangkan lagi menjadi *Responsive Web Design* (RWD), yang

dapat mengoptimalkan kegunaan dari *web* itu sendiri dengan tampilan yang dapat dikelola sesuai dengan layar pengguna dan dapat dilakukan dimana saja dan kapan saja. (Anita, dkk, 2020).

Web merupakan kumpulan halaman *web* yang saling berkaitan. *Web* adalah sebuah sistem yang berisi beragam informasi baik berupa teks, gambar, audio maupun video dan dapat diakses melalui perangkat yang biasa disebut *web browser*. *Web* merupakan kumpulan informasi baik yang bersifat statis maupun dinamis yang terdiri dari halaman yang dibuat. (Oktasari dan Kurniadi, 2019).

II.2.10. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa untuk menentukan visualisasi, konstruksi, dan mendokumentasikan artefak bagian dari informasi yang digunakan atau dihasilkan dalam proses pembuatan perangkat lunak. Artefak dapat berupa model, deskripsi atau perangkat lunak pada sistem perangkat lunak, seperti dalam pemodelan bisnis dan perangkat lunak non-sistem lainnya. UML adalah kumpulan teknik terbaik yang telah terbukti berhasil dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses perangkat lunak pemodelan, tetapi di hampir semua bidang yang membutuhkan pemodelan. (Linda Wahyuni dan Rika Rosnelly, 2021).

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:

1. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
4. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
6. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Omni Alfina dan Fitriana Harahap: 2019).

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

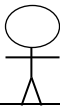
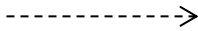
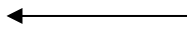
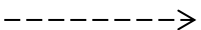
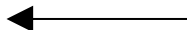
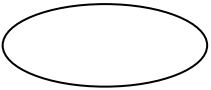
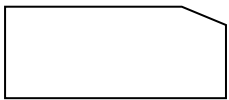
1. *Use Case Diagram*

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 Diagram*

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Depedency</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>	Menspesifikan 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 (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber: Andikos, 2019)

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*

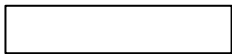
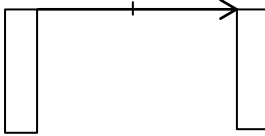
Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana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

(Sumber: Andikos, 2019)

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

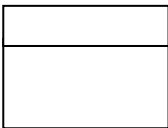
Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

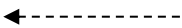
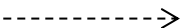
(Sumber: Andikos, 2019)

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)