

BAB 2

TINJAUAN PUSTAKA

2.1 Media sosial

Media sosial adalah sebuah teknologi yang diperantarai oleh komputer interaktif yang memfasilitasi untuk pembuatan dan berbagi suatu informasi, ide, minat karir, dan bentuk ekspresi lain melalui komunitas dan jaringan virtual. Pengguna biasanya mengakses layanan media sosial melalui teknologi berbasis *web* di desktop dan laptop, atau mengunduh layanan yang menawarkan fungsionalitas media sosial ke perangkat seluler mereka (misalnya, ponsel cerdas, dan tablet). Pengguna membuat sebuah konten seperti, foto digital, video, atau teks dan data yang dihasilkan semua interaksi dilakukan secara *online*. Pengguna membuat identitas atau profil untuk *web* atau aplikasi yang dikelola oleh organisasi media sosial. Ketika menggunakan layanan ini, pengguna dapat membuat *platform* interaktif antar individu, komunitas, dan organisasi. pengguna dapat saling berbagi, mendiskusikan, dan memodifikasi konten buatan pengguna yang diposting via *online* [2]. Media sosial pada saat ini menjadi gaya hidup masyarakat untuk dapat bersosialisasi dan bentuk baru untuk berdiskusi antara “*consumer-to-consumer*” dan “*business-to-consumer*” yang memiliki implikasi besar terhadap pemasar [3].

2.1.1 Karakteristik unik media sosial

Menurut Wahyu Dian Purnomo, media sosial mempunyai karakteristik yang unik diantaranya adalah :

1. **Transparan:** Media sosial mempunyai sifat terbuka untuk menyebarkan informasi agar diketahui oleh publik baik itu kelompok sosial atau individu.
2. **Terdapat dialog:** Seseorang, brand bisnis, atau selebritis dapat menjalin komunikasi dengan *followernya* secara interaktif, selebritis, brand bisnis atau perseorangan dapat menjalin hubungan dan komunikasi interaktif dengan para *followernya* apalagi jika anda adalah pemilik akun

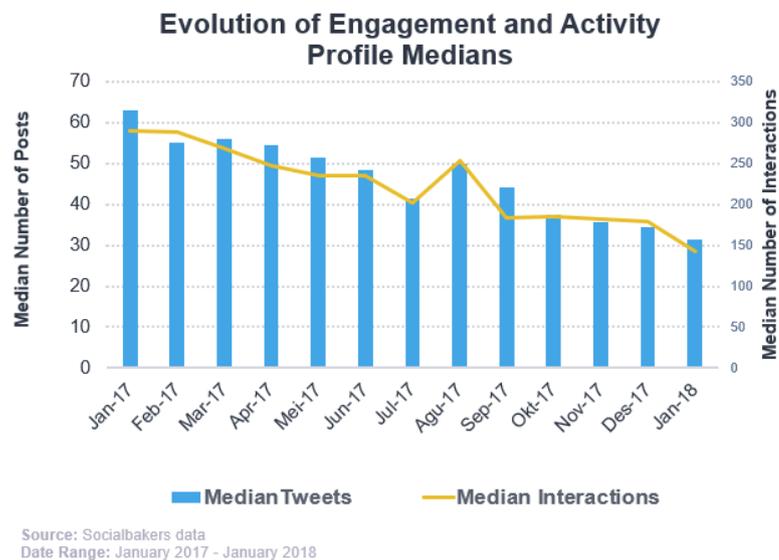
bisnis, produk atau jasa dan pelayanan Anda akan semakin disukai oleh calon konsumen apabila respon yang Anda tunjukkan semakin cepat.

3. Jejaring sosial: media sosial dapat membentuk suatu komunitas di yang mana setiap anggotanya dapat menjalin hubungan dan saling berkomunikasi satu sama lain seperti jaring laba-laba yang dinamakan jejaring sosial yang memiliki peranan yang sangat kuat untuk mempengaruhi audiense-nya.
4. Multi opini: setiap pengguna dapat mengutarakan pendapatnya dan berargumentasi dengan pengguna lainnya secara interaktif.
5. Multi form: media sosial mempunyai bentuk yang beraneka ragam dan mempunyai banyak kanal seperti *social media* press release, portal *web*, video, dan sebagainya.
6. Memiliki kelebihan sebagai *media* promosi *online*: media sosial mempunyai alat atau fitur yang dapat mewujudkan visi dan misi sebuah organisasi atau kelompok dan memunculkan sebuah peluang [4].

2.1.2 Teknik pemasaran dengan media sosial

Menurut Wahyu Dian Purnomo, Teknik pemasaran dengan media sosial atau *social media* marketing adalah suatu bentuk memperkenalkan atau mempromosikan suatu produk atau jasa melalui *media* digital seperti media sosial untuk meningkatkan komunikasi dengan konsumen demi meningkatkan penjualan dan mutu pelayanan yang tidak hanya satu arah melainkan bisa timbal balik [4]. Berdasarkan hasil wawancara Randi Eka dengan Moses Velasco, *Chief of Strategy* Socialbakers.com. mengatakan pendapatnya tentang bagaimana tren pemasaran melalui media sosial yang ada saat ini. Melihat perkembangannya di Indonesia, Moses mengatakan bahwa penggunaan yang sangat luas dari media sosial membuat konsumen beralih menjadi *mobile-centric* consumer. Sederhananya, kini semuanya serba menjadi consumable di *platform mobile apps*. Implikasinya sebuah merek dan publisher harus teliti dengan strateginya merangkul *audience* yang ditargetkan. Menurut Moses, generasi muda masih menjadi pangsa konsumen terbesar, dengan tren *mobile* saat ini, brand adalah hal terpenting yang memenangkan pangsa pasar saat ini. Misalnya memahami bahwa generasi muda lebih menyukai konten berupa visual daripada teks. Hal ini ditunjukkan oleh Moses dengan menampilkan data

tentang tren *engagement* di Twitter yang cenderung terus menurun, karena twitter adalah media sosial yang fokus utamanya menyampaikan konten dan informasi



**Gambar 2.1 Grafik pengguna twitter di Indonesia Januari
2017 – Januari 2018**

berupa teks [5]

Media sosial menjadi berkembang dan membuat paradigma baru, yaitu mendorong sebuah kepercayaan melalui konten yang dibagikan, setelah itu adalah membangun *engagement* setelah konten yang dibuat telah terpublikasi sesuai dengan target *audience* atau sampai kepada segmentasi yang tepat. Moses juga berkomentar seputar tren *platform* media sosial bahwa saat ini konten tidak ditampilkan secara kronologis melainkan dengan berdasarkan *algoritma* tertentu yang artinya untuk mencapai target yang *engagement* kebanyakan brand atau publisher melakukan promosi berbayar karena media sosial saat ini didesain sedemikian rupa agar orang-orang yang menginginkan target *audience* yang sesuai melakukan promosi berbayar. Sisi positifnya adalah mereka menyediakan targeting yang cukup relevan, keefektifitasnya tergantung dari strategi mengencarkan media sosial tersebut [5].

Menurut Wahyu Dian Purnomo, sebagai brand atau publisher wajib memiliki akun di media sosial sebagai *media* promosi, meskipun sudah memiliki *website* sendiri dengan alasan berikut:

1. Sebagai informasi kepada publik tentang bisnis yang sedang dijalankan, bertujuan untuk memperkenalkan produk atau jasa anda untuk menjangkau pasar yang lebih luas dan biaya yang relatif murah serta untuk mendapatkan *follower* atau target *audience*.
2. Media sosial juga dapat digunakan untuk mendapatkan ulasan atau *feedback* yang cepat dari konsumen yang telah membeli produk atau menggunakan jasa yang telah ditawarkan secara *online* hal ini berguna untuk meningkatkan kualitas pelayanan serta mengantisipasi apabila ada komentar negatif yang dapat merusak reputasi, komentar positif juga dapat digunakan sebagai testimoni untuk menarik minat calon konsumen.
3. Media sosial lebih komunikatif dan interaktif dengan *follower* atau pengguna produk anda.
4. Biaya promosi di media sosial relatif lebih murah dibandingkan biaya promosi konvensional seperti promosi di TV atau reklame.
5. Media sosial membuat promosi meluncurkan jasa atau produk baru lebih mudah [4].

2.1.3 Optimasi Pemasaran dengan media sosial

Moses Velasco mengatakan bahwa ada dua hal yang harus diperhatikan sebagai brand dan publisher untuk memenangkan strategi pemasaran di media sosial.

- 1) Cara yang pertama adalah brand dan publisher harus benar-benar mengetahui secara detail siapa *audience* yang ditargetkan sehingga bisa tahu strategi selanjutnya yang akan diterapkan pada tahapan selanjutnya. Media sosial sendiri saat ini mempunyai beranekaragam karakteristik pengguna dan sudah sangat terfragmentasi.
- 2) Cara yang kedua adalah menentukan *platform* dan strategi konten yang akan digunakan berdasarkan pada karakteristik *audience* tadi. Contohnya adalah

brand saat ini gencar mencari pangsa pasar generasi muda, karena mereka lebih menyukai konten visual berupa gambar atau video pendek untuk penyebaran konten. seperti instagram. Moses berpendapat bahwa *platform* media sosial dengan fitur Live Video atau Story adalah fitur yang sedang populer saat ini, dan dapat dimanfaatkan untuk melakukan pendekatan dengan *audience*. Live Video dapat menyebarkan informasi one-to-many broadcast dengan keuntungan dapat berinteraksi langsung dengan *audience*. Moses Velasco juga berpendapat bahwa bagian penting untuk membantu meningkatkan pemasaran digital melalui media sosial adalah teknologi Artificial Intelligence (AI) berupa komputasi pintar karena memiliki *algoritma* yang dapat diterapkan untuk membantu pengguna menganalisis dan memperoleh hasil sesuai dengan *engagement* yang diharapkan [5].

2.2 Instagram

2.2.1 Profil Instagram

Instagram adalah sebuah aplikasi dan *web* untuk ponsel pintar berupa media sosial yang merupakan salah satu dari *media* digital yang berfungsi menyerupai twitter, namun perbedaan instagram dengan twitter terletak pada fitur pengambilan foto atau video dalam bentuk serta menampilkan tempat untuk berbagi informasi terhadap penggunanya, Instagram juga dapat meningkatkan kreatifitas, karena mempunyai fitur untuk mengubah foto menjadi lebih indah, lebih mempunyai syarat seni, dan menjadi lebih bagus. [6] Instagram diluncurkan pada tanggal 6 Oktober 2010 dan dirilis pertama kali untuk *platform* iOS. Sebanyak 25 ribu pengguna berhasil mendaftar di hari pertama. Pada tanggal 13 Oktober 2010 penggunanya mencapai 100 ribu dan pada tanggal 21 Desember 2010 mencetak rekor jumlah pengguna mencapai 1 juta dan terus bertambah hingga saat ini [6]. Kini, instagram sudah dapat digunakan dalam program smartphone lainnya seperti *Android*. Ini juga membuat penggunaan Instagram semakin luas. Instagram berasal dari kata “insta” dan “gram”, “insta” berasal dari kata “instan” yang terinspirasi dari kamera polaroid yaitu foto instan, karena instagram dapat menampilkan foto instan. “sedangkan kata “gram” berasal dari “telegram” yang mempunyai arti seperti cara

kerja telegram yaitu mengirimkan informasi kepada orang lain dengan cepat. Seperti telegram, instagram dapat mengunggah foto dengan jaringan *internet*, sehingga informasi yang ingin disampaikan dapat diterima oleh orang lain dengan cepat. [7]

2.2.2 Sejarah Instagram

Instagram pertama kali didirikan dan dirancang oleh Kevin Systrom dan Mike Krieger di San Francisco pada tahun 2010, mereka berdua adalah CEO dari sebuah perusahaan startup yang hanya berfokus dalam HTML5 peranti bergerak dan pengembangan aplikasi untuk telepon genggam, Burbn, Inc. Pada mulanya mereka menciptakan sebuah aplikasi bernama Burbn yang sudah dapat digunakan oleh iPhone, namun didalamnya masih ada beberapa hal yang belum sempurna, dan di versi akhirnya, terlalu banyak fitur-fitur, akhirnya Kevin Systrom dan Mike Krieger memulai ulang membuat aplikasi dari awal yang berfokus pada bagian foto, yang mempunyai fitur komentar, dan kemampuan untuk menyukai sebuah foto, dan pada akhirnya aplikasi itu diberi nama, instagram. Instagram rilis perdana pada tanggal 6 Oktober 2010 dan pertama kali hanya beroperasi pada sistem operasi ios, 6 bulan kemudian instagram untuk *android* resmi dirilis. Facebook mengakuisisi Instagram pada tanggal 9 April 2012 seharga hampir \$1 miliar dalam bentuk tunai dan cash.

2.2.3 Logo Instagram



Gambar 2.1 Logo Instagram (20 September 2011 – 11 Mei 2016)



Gambar 2.2 Logo Instagram (12 Mei 2016- Sekarang)

2.2.4 Fitur-fitur Instagram

Instagram mempunyai banyak fitur dasar sebagai jejaring sosial dan juga fitur untuk menunjang teknik pemasaran melalui media sosial di antaranya:

1. Posting foto dan video

Pengguna dapat memposting foto dan video yang dapat diperoleh dari gallery di device mereka maupun menangkap langsung dari kamera di device mereka. Foto atau video yang sudah di unggah akan muncul di feed instagram yang juga akan muncul di halaman profil instagram pengguna dengan rasio 1:1, 4:5, dan 16:9,

pengguna juga dapat memposting foto dan video pada story instagram yang akan otomatis hilang selama 24 jam, dengan rasio 9:16. Durasi video di feed instagram maksimal 60 detik, dan 15 detik untuk story instagram. Pengguna dapat menambahkan filter pada foto atau video yang akan diposting.

2. *Followers* (Pengikut)

Sistem sosial yang ada di dalam Instagram adalah dengan cara mengikuti akun pengguna lainnya, atau diikuti oleh pengguna Instagram lain. Dengan demikian komunikasi yang terjadi antar pengguna Instagram sendiri dapat terjalin dengan cara *memberikan* tanda suka (*like*) dan juga pengguna bisa mengomentari foto-foto yang telah diposting oleh pengguna lainnya. Pengikut juga menjadi salah satu unsur yang penting, dan jumlah tanda suka dari para pengikut sangat mempengaruhi apakah foto tersebut dapat menjadi sebuah foto yang populer atau tidak. Untuk menemukan teman-teman yang ada di dalam Instagram, pengguna juga dapat menemukan teman-teman mereka yang sudah menggunakan Instagram melalui jejaring sosial seperti twitter dan juga facebook.

3. *Influencer*

Influencer merupakan orang-orang yang mempunyai *audience* atau *followers* yang cukup banyak di media sosial dan mereka mempunyai pengaruh yang kuat terhadap *followers* mereka, seperti artis, selebgram, blogger, youtuber, dan lain sebagainya. Mereka disukai dan dipercaya oleh *audience* atau *followers* mereka sehingga apa yang mereka pakai, sampaikan atau lakukan, bisa menginspirasi dan mempengaruhi para *followers*nya, termasuk untuk mencoba atau membeli sebuah produk. *Influencer* marketing merupakan cara untuk promosi yang efektif karena konsumen jaman sekarang sudah kurang suka terhadap iklan perhatian mereka sehari-hari berada di media sosial dan mereka suka melihat konten-konten menarik dari *influencer* yang mereka ikuti [8].

4. *Arroba* (@)

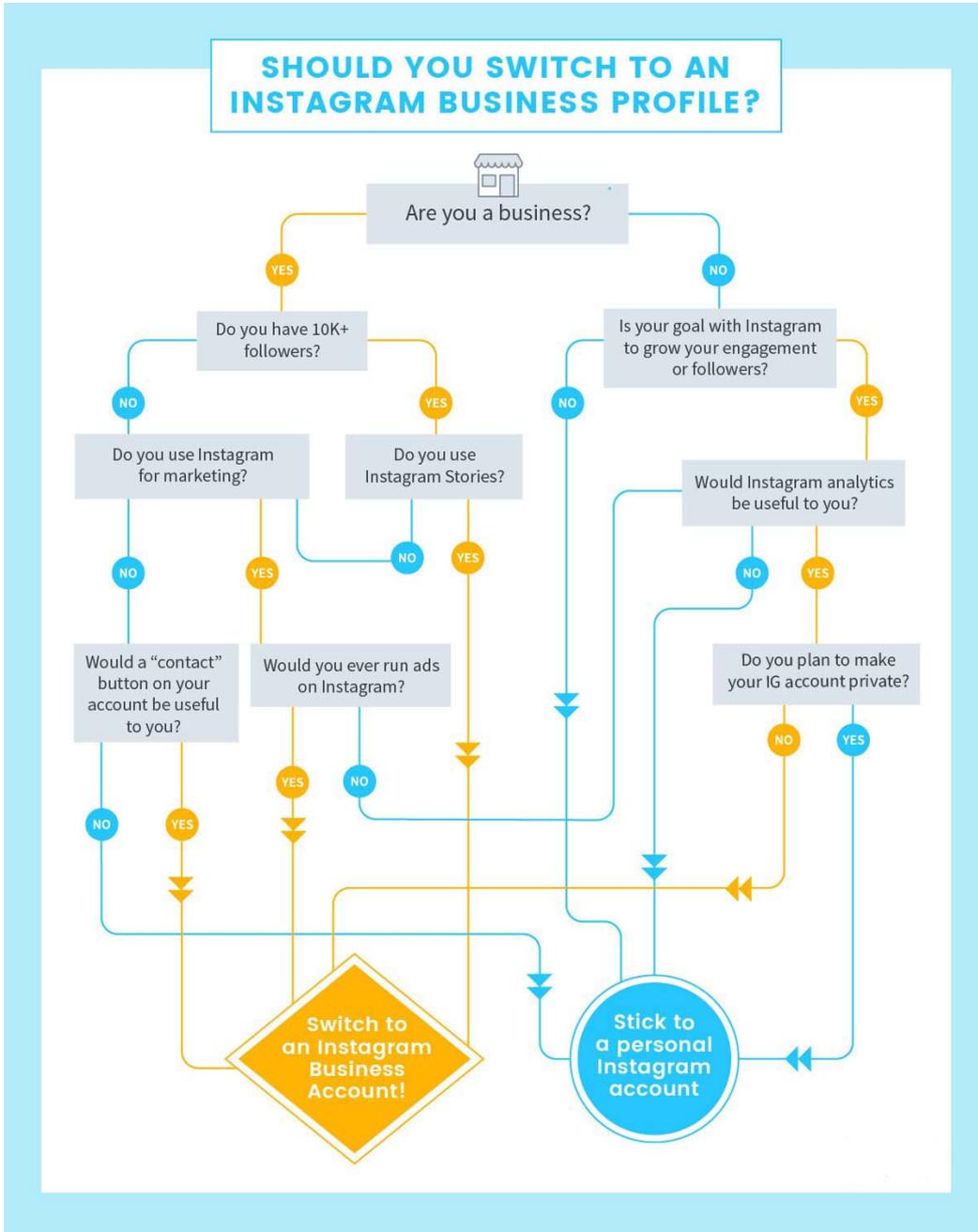
Instagram mempunyai sistem menggunakan tanda *arroba* (@) seperti twitter pada bagian sebelum nama pengguna untuk menyinggung pengguna lain. Para Pengguna dapat menyinggung pengguna lainnya di kolom komentar untuk menjalin komunikasi antar pengguna, Para pengguna juga dapat mencantumkan *arroba* nama pengguna lain di judul foto atau video dan juga di profil instagram.

5. *Hashtag*/Tagar (#)

Hashtag adalah simbol atau kode yang digunakan untuk menelusuri, menemukan, dan mengelompokkan pesan, kata kunci. Untuk kasus Instagram, berupa gambar atau video. Misalnya, jika mengetik #kucing, #cat atau #cats di salah satu foto atau video, akan menemukan foto atau video yang menggunakan tagar yang sama. Para pengguna di mana pun yang mencari "kucing" atau "cat" akan dengan mudah menemukan gambar atau video pengguna lainnya yang menggunakan kata kunci # yang sama [9].

6. Profil Instagram Bisnis

Instagram mempermudah pengguna yang menggunakan instagram untuk keperluan bisnis dengan *memberikan* beberapa fitur yang lebih dibanding profil instagram biasa dengan cara beralih ke profil Instagram bisnis. Setelah mengubah profil menjadi instagram bisnis, pengguna dapat mengakses fitur khusus seperti, Instagram Insights yaitu kemampuan sederhana untuk melacak jumlah tayangan, jumlah jangkauan, dan jumlah keterlibatan terhadap postingan, kemampuan untuk mempromosikan konten postingan, melihat data dan grafik *engagement*, memunculkan tombol "kontak" di profil, dan menambahkan tautan atau link ke Instagram Stories [10].



Gambar 2.3 Skema Kebutuhan Instagram Bisnis

2.3 Landasan Teori

Landasan Teori *memberikan* gambaran dari teori yang terkait dengan pembangunan perangkat lunak. Landasan teori ini akan menjadi acuan dasar penulis dalam melakukan penelitian melalui uraian-uraian secara teoritis.

2.3.1 Teknologi Website

World Wide Web (WWW) merupakan suatu sistem penyedia informasi dengan skala yang besar yang mengelola informasi tersebut secara terdistribusi dalam *internet* dengan menggunakan teknologi *hypermedia*. WWW mempunyai karakteristik sebagai berikut:

1. Dukungan *interface* yang terintegrasi WWW menyediakan suatu *interface* yang terintegrasi terhadap protokol, format data, sistem pengalamatan, dan lain-lain. Hal ini menyebabkan berbagai macam layanan dan basis data yang ada di *internet* dapat diakses secara langsung.
2. Kemudahan pada sisi pengguna WWW mendukung secara transparan sebagian besar aplikasi-aplikasi yang terdapat di *internet* seperti telnet, gopher, anonymous *file* transfer protocol, finger, dan aplikasi lainnya.
3. Kemudahan dalam perkembangan Kapabilitas *server* WWW dapat dikembangkan secara mudah dengan menggunakan standar PHP antara *server* WWW dengan aplikasi yang lain. Program PHP juga memungkinkan perubahan informasi secara dinamis yang dapat diperoleh secara real time.
4. Tidak tergantung pada platform tertentu WWW memungkinkan seseorang dapat membangun *server* WWW di berbagai sistem yang berlainan dan *memberikan* informasi dalam bentuk *hypermedia*. WWW tidak didesain untuk mendukung sistem tertentu.

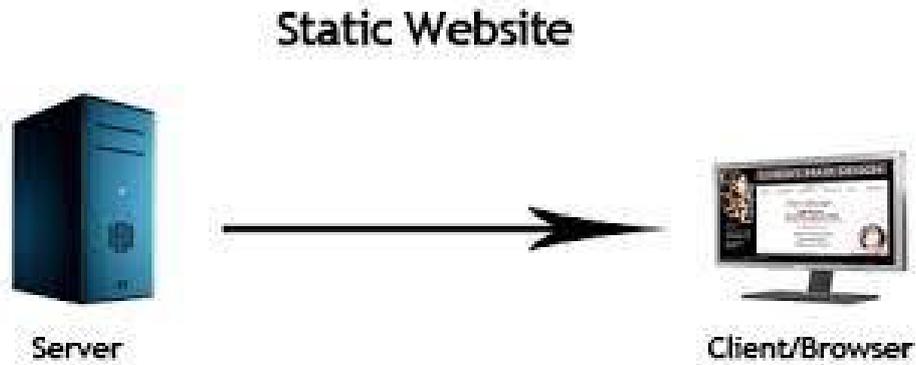
WWW menggunakan model *client server*. WWW juga menggunakan ekspresi informasi, transfer informasi, metode penamaan informasi yang standar sehingga dapat memproses dan mentransfer informasi secara terdistribusi yang dilakukan secara sistematis. Informasi yang terdistribusi disimpan di dalam *server* WWW dan pengguna mengakses informasi tersebut menggunakan suatu software yang disebut

dengan *browser*. *Server* menyimpan informasi di dalam *file-file* terstruktur menggunakan *Hypertext Markup Language (HTML)*. *Hypertext* merupakan kumpulan teks-teks yang saling berhubungan satu sama lainnya. *Hypermedia* serupa dengan *hypertext*, hanya saja *media* yang digunakan bukan hanya teks akan tetapi meliputi juga audio, image, dan video. WWW menggunakan standar untuk mencapai konsistensi dalam menghasilkan dan melakukan transfer informasi. WWW menggunakan HTML sebagai standar produksi informasi dengan menggunakan *Hypertext Transfer Protocol (HTTP)* sebagai standar transfer informasi. WWW menggunakan standar penamaan yang disebut dengan *Universal Resource Locator (URL)* untuk mendukung penyediaan informasi secara universal dalam *server* yang terdistribusi.

2.3.2 Aplikasi Program Berbasis Web

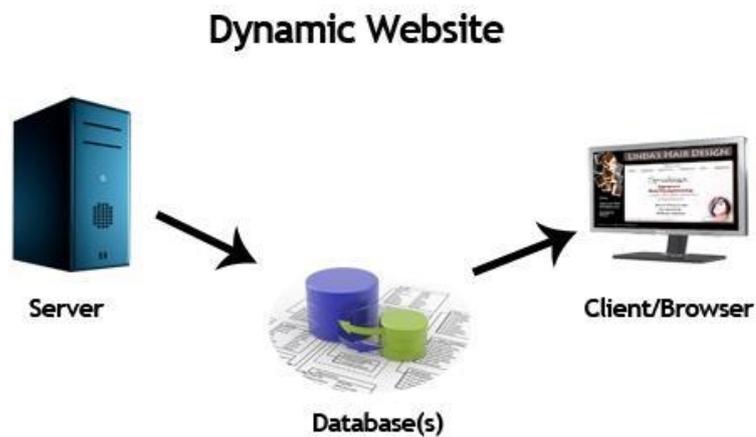
Banyak situs *internet* yang memiliki halaman dengan sifat statis seperti profil perusahaan, artikel, dan keterangan-keterangan lain. Situs ini mempunyai dokumen dengan teks yang sederhana, image dan *hyperlinks* ke dokumen yang dimilikinya. Untuk mengembangkan situs yang bersifat statis, kita menggunakan teknologi *client side*. HTML dan *Cascading Style Sheet (CSS)* dapat digunakan untuk mengatur struktur dan menampilkan halaman isi. Seandainya ingin diperindah dapat ditambahkan script yang sifatnya *client side*, seperti JavaScript, Jscript ataupun VBScript. Dengan berkembangnya *internet*, situs yang ada di *internet* tidak hanya berfungsi untuk mempresentasikan *content* tetapi cenderung berupa aplikasi yang kebanyakan terhubung ke suatu basis data. Pada tahapan ini situs akan bersifat dinamis, karena *content* yang dipresentasikan akan bervariasi dan berubah-ubah sesuai dengan data yang diminta dan action dari pengguna. Untuk mengembangkan situs yang dinamis diperlukan teknologi *server side* seperti PHP, ASP, Perl dan CGI yang lain. Dengan teknologi *server side* kita dapat

mengembangkan suatu aplikasi berbasis *internet* yang dapat menghasilkan dan menampilkan *content* secara dinamis.



Gambar 2.4 Website Statis

Pada Gambar 2.4 di atas diperlihatkan cara kerja situs dengan *content* yang statis. Pada saat terjadi suatu request dari *browser*, *server web* selanjutnya membaca request yang dikirim oleh *browser*, mencari dan menemukan halaman di *server*, dan mengirim halaman yang diminta melalui *internet* ke *browser*. Selanjutnya halamantersebut ditampilkan di *browser*.



Gambar 2.5 Website Dinamis

Pada Gambar 2.5 diatas diperlihatkan cara kerja situs dengan *content* yang dinamis:

- a. *Browser (client)* melakukan request halaman *web* ke *WEB Server*.
- b. *WEB Server* kemudian membaca request yang dikirim dari *browser*, mencari dan menemukan halaman di *server*.
- c. Menterjemahkan perintah yang diberikan oleh bahasa program *server side* menjadi halaman HTML.
- d. Setelah konversi ke halaman HTML selesai kemudian mengirim halaman yang diminta ke *browser*.
- e. Setelah halaman HTML yang dikirim dari *WEB server* sampai, selanjutnya *browser client* akan menampilkan halaman *web*.

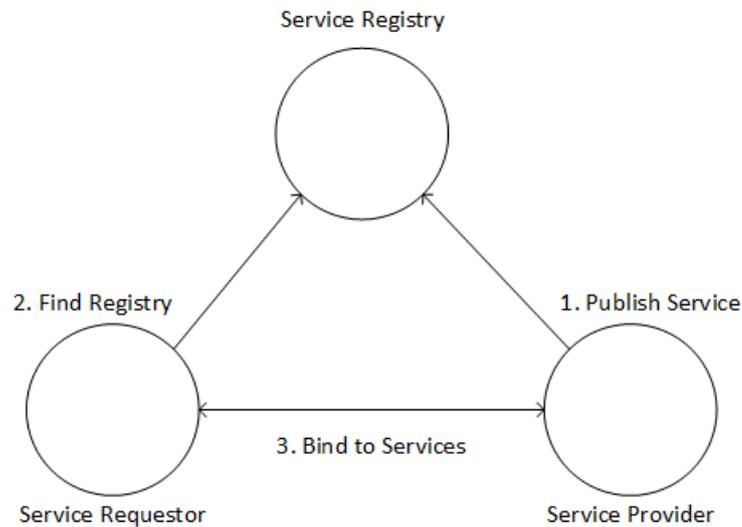
2.3.3 *Web Service*

Web Service atau API (*Application Programming Interface*) adalah suatu sistem perangkat lunak yang dirancang untuk mendukung *interaction* and *interoperability* antar sistem pada suatu jaringan.

Web service digunakan sebagai suatu fasilitas yang menyediakan layanan (dalam bentuk informasi atau data) kepada sistem lain, sehingga dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan. *Web service* menyimpan data informasi dalam format JSON atau XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, dan bahasa pemrograman [11].

1. *Arsitektur Web Service*

Secara umum, arsitektur *web service* dapat dilihat pada gambar 2.3.3 berikut



Gambar 2.6 Arsitektur *Web Service*

. Pada gambar 2.3.3 diatas, ada tiga komponen utama dari *web service* yaitu:

1. *Service Provider* adalah penyedia *web service* yang berfungsi menyediakan kumpulan *web service* yang dapat diakses oleh pengguna.
2. *Service Requestor* adalah aplikasi yang bertindak sebagai pengguna yang melakukan permintaan layanan (berupa *web service*) ke *service provider*
3. *Service Registry* adalah tempat dimana *service provider* mempublikasikan layanannya. Pada arsitektur *web service*, *service registry* bersifat opsional [12].

2. Strategi API

Dalam kebanyakan kasus, sangat bermanfaat bagi penyedia API untuk menentukan kasus penggunaan primer untuk API. Ini satu dari pertimbangan penyedia API yang paling penting untuk dibuat. API mungkin ditargetkan untuk penggunaan internal dalam bisnis, atau untuk mitra atau tujuannya adalah membuat data dapat diakses oleh pengembang perorangan di seluruh dunia. Tiga pendekatan utama untuk strategi API diidentifikasi

oleh Boyd (2014 & 2015) disebut *private*, *partner* dan *public*. Juga Willmott (2015) menggunakan klasifikasi yang sangat mirip. Pendekatan yang relevan bergantung pada model bisnis penyedia API dan penyedia API lingkungan bisnis beroperasi. Secara teknis API diimplementasikan di bawah pendekatan ini mungkin sama, namun pendekatan yang dipilih menentukan bagaimana penyedia API memprioritaskan berbagai fitur dan pemangku kepentingan API yang berbeda.

3. Private API

Private API ditargetkan untuk pengembang yang bekerja dalam internal perusahaan. *Private* API sering digunakan untuk membuat beberapa aplikasi atau kumpulan data dapat diakses dalam internal perusahaan sehingga mudah digunakan kembali di aplikasi lain atau unit bisnis. Dengan cara ini *Private* API dapat digunakan untuk mengurangi duplikat pemrosesan informasi antar departemen. Selain itu *Private* API dapat digunakan sebagai blok bangunan arsitektur *microservices*, yaitu varian yang lebih modern dan lebih ringan dari pendekatan arsitektur berorientasi layanan yang lebih lama (Bradbury, 2016). Dalam pendekatan ini model aplikasi tradisional dipecah menjadi beberapa layanan kecil yang saling berbicara melalui API. *Microservices* bisa membawa lebih banyak kecepatan dan kelincahan untuk pengembangan perangkat lunak. Aplikasi yang lebih kecil dapat lebih cepat berkembang dan mudah berubah, berkat API ada sedikit independensi bila dibandingkan dengan aplikasi model lama/tradisional. *Private* API juga sering digunakan saat ada kebutuhan untuk mengakses data melalui *mobile* seperti aplikasi iPhone atau *Android*. Banyak profesional API seperti Lane (2012a) merekomendasikan agar organisasi memulai eksperimen mereka dengan *Private* API lalu pindah ke *Partner* dan *Public* API saat mereka telah mengumpulkan pengalaman dalam pengelolaan dan pembangunan API yang dibutuhkan. Pendekatan *Private* API memungkinkan penyedia API untuk melihat masalah dan tantangan terkait dengan API, namun kegagalan hanya akan mempengaruhi penyedia API itu sendiri. Satu perusahaan yang menggunakan pendekatan ini adalah Amazon, salah satu tim mengekspos data dan fungsinya ke tim lain melalui API.

Pendekatan ini memungkinkan Amazon untuk tetap terorganisir dan tangkas, sementara ada beberapa tim yang bekerja dengan berbagai layanan infrastruktur sistem di waktu yang sama. Operasi dan dukungan memerlukan lebih sedikit sumber daya dalam pendekatan *Private API* daripada yang pendekatan lainnya. Penyedia API harus tetap berusaha membuat dokumentasi API berkualitas tinggi dan *memberikan* dukungan yang diperlukan kepada pengembang sehingga mereka dapat menggunakan API dan mewujudkan kesempatan yang diberikan oleh API. *Private API* biasanya tidak didokumentasikan secara terbuka.

4. Partner API

Dalam pendekatan *Partner API*, penyedia API menggunakan API untuk mengintegrasikan dan mengotomatisasi interaksi dengan mitra-mitranya. Pendekatan ini populer terutama di kalangan bisnis yang beroperasi di lingkungan pasar *business-to-business*. Dokumentasi yang terdefinisi dengan baik akan memudahkan integrasi proses bisnis saat hubungan bisnis baru sedang dibangun. API mengurangi kebutuhan akan penyesuaian integrasi sistem-ke-sistem dan memudahkan komunikasi antara mitra yang telah dibangun dan didokumentasikan terlebih dahulu dan diimplementasikan menggunakan teknologi yang standar. *Partner API* juga memungkinkan untuk memperdalam hubungan dengan mitra dengan mengintegrasikan proses bisnis lebih ketat dan dengan menawarkan data operasional *real time partner*. Kasus khusus penggunaan untuk *Partner API* adalah faktor otomatis, sistem pemesanan dan pembayaran, produk dan katalog harga dan berbagai jenis pencarian status. Pendekatan *Partner API* biasanya membutuhkan lebih banyak sumber daya dibanding pendekatan *Private API*, karena penyedia API harus mendukung juga eksternal pengembang yang bekerja untuk subkontraktor mitra atau mitranya. Di lingkungan *Partner API* saat ada pengembangan atau fitur baru perubahan tidak boleh merusak integrasi yang ada dengan sistem mitra yang sudah ada. Dokumentasi *Private API* mungkin bersifat publik, namun untuk mendapatkan akses ke API biasanya memerlukan hubungan bisnis dengan penyedia API.

5. *Public API*

Public API didokumentasikan secara terbuka dan dapat diakses oleh pihak manapun yang ingin mendaftar. Pada kasus paling baik *Public API* memungkinkan penyedia API untuk memanfaatkan kreativitas pengembang pihak ketiga yang tak terhitung jumlahnya. *Public API* memberi kesempatan kepada penyedia API untuk mengubah produknya menjadi *platform*, yang mana pengembang pihak ketiga dapat memperpanjang dan memasukkannya ke dalam aplikasi mereka sendiri. Itu satu alasan mengapa perusahaan *online* berorientasi *platform* besar seperti Facebook, Twitter dan Google memiliki banyak berinvestasi dalam pengembangan API. Juga penyedia *Software-as-service* yang melayani perusahaan kecil dan menengah biasanya memungkinkan untuk memperluas layanan mereka melalui *Public API*.

Pendekatan *Public API* juga disukai oleh organisasi sektor publik seperti pemerintah kota, karena data yang mereka miliki sering bersifat publik dan harus dibuat semudah diakses untuk semuanya. Mengembangkan dan mendukung *Public API* memerlukan lebih banyak sumber daya dari penyedia API daripada dua pendekatan sebelumnya. Penyedia API harus *memberikan* dukungan bagi pengembang yang mungkin tidak sangat akrab dengan produknya. Menurut banyak profesional API seperti Bortenschlager (2015) *Developer Experience (DX)* adalah salah satu faktor terpenting *program API* yang berhasil terutama dalam kasus *Public API*. Istilah pengalaman pengembang mengacu pada total pengembang bekerja dengan API yang diberikan. Pengalaman pengembang adalah sejumlah hal seperti kualitas Dokumentasi API, daya tanggap dukungan teknis, kualitas teknis API, kemudahan memulai dengan pekerjaan pengembangan dan kualitas alat pengembangan perangkat lunak yang tersedia untuk API. Untuk mendapatkan pengalaman pengembang mungkin memerlukan investasi yang signifikan dari pemberi API. Agar berhasil melibatkan pengembang pihak ketiga ke dalam pengembangan produk penyedia API harus memiliki data, yang mengilhami

pengembang atau perusahaan dapat *memberi* insentif seperti pembagian keuntungan kepada pengembang [13].

2.3.4 Javascript

JavaScript (sering disingkat menjadi JS) adalah sebuah bahasa pemrograman tingkat tinggi yang dinamis, berorientasi objek, dan paling populer sebagai bahasa scripting untuk halaman *web browser*, javascript juga sering digunakan untuk *non-browser*. JavaScript berjalan di sisi klien *web*, yang dapat digunakan untuk mendesain/memprogram bagaimana halaman *web* berperilaku pada terjadinya suatu peristiwa. JavaScript adalah bahasa yang mudah dipelajari dan juga bahasa scripting yang kuat, banyak digunakan untuk mengontrol perilaku halaman *web* [14].

2.3.5 PHP

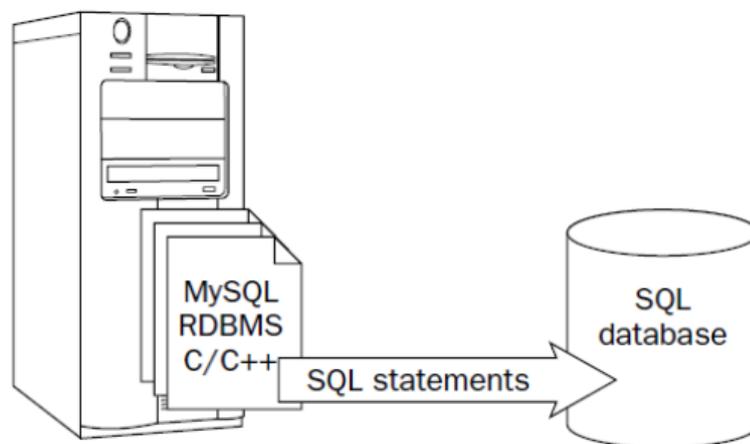
PHP atau *Hypertext Preprocessor* adalah bahasa scripting yang menyatu dengan HTML dan berjalan di sisi *server*. PHP diperkenalkan pada tahun 1995 pertama kali oleh Rasmus Lerdorf seorang *Software Developer*. dengan PHP adalah singkatan dari PHP *Hypertext Preprocessor*. PHP dapat di unduh dan digunakan gratis di situs resminya <http://www.php.net> karna merupakan software open-source yang lisensinya disebarikan secara gratis. PHP sering digunakan untuk proses pembangunnn aplikasi dan ditulis dengan bahasa C [15].

2.3.6 CodeIgniter

Code Igniter (CI) (www.codeigniter.com) adalah salah satu *framework* php yang tangguh dan populer. Code Igniter tergolong *framework* dengan ukuran kecil dan cukup mudah di kuasai. CI juga datang dengan manual yang tergolong lengkap. CI adalah sebuah php *framework* yang berupa kumpulan folder dan *file* php, java script,css,txt dan *file* berbasis *web* lainnya dengan *setting* tertentu untuk menggunakannya dan menyediakan library dan helper yang dapat di manfaatkan di dalam pemrograman php. CI di jalankan under *web* dan harus dengan *web server*. Program CI cukup di letakkan di bawah folder directory *web server* pengguna [16].

2.3.7 MySQL

MySQL adalah sebuah aplikasi *Relational Database Management Server* (RDBMS) bersifat *open source* yang memungkinkan data diakses dengan cepat oleh banyak pemakai secara bersamaan dan juga memungkinkan pembatasan akses pemakai berdasarkan *privilege* (hak akses) yang diberikan. MySQL menggunakan bahasa SQL (*structured query language*) yang merupakan bahasa standar pemrograman *database* [15].



Gambar 2.7 Interaksi MySQL RDMS dengan SQL

2.3.8 OOP (Object Oriented Programming)

Object Oriented Programming atau yang dikenal dengan Pemrograman Berorientasi Objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus ke dalam kelas-kelas atau objek-objek. Model data berorientasi objek dikatakan dapat *memberi* fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai

contoh anggap kita memiliki sebuah departemen yang memiliki *manager*, sekretaris, petugas administrasi data dan lainnya. Misal *manager* tersebut ingin memperoleh data dari bag administrasi maka *manager* tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang *manager* tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi *manager* bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri [17].

2.3.9 UML (Unified Modelling Language)

UML (Unified Modeling Language) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem. *UML* adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (*OOP*) serta aplikasinya. *UML* adalah metodologi untuk mengembangkan sistem *OOP* dan sekelompok perangkat *tool* untuk mendukung pengembangan system tersebut. *UML* mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar *OOP* sejak tahun 1980-an. Sekarang *UML* sudah mulai banyak digunakan oleh para praktisi *OOP*. *UML* suatu bahasa yang digunakan untuk menentukan memvisualisasikan membangun, dan mendokumentasikan suatu sistem informasi. *UML* dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian *UML* dapat digunakan untuk memahami dan mendokumentasikan setiap system informasi. Penggunaan *UML* dalam industri terus meningkat, ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem [18].

UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah [19]:

1. *Use case Diagram*

Use case Diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem secara keseluruhan yang akan dibuat. *Diagram use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, *diagram use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Yang ditekankan pada *diagram* ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. *Use case* menjelaskan secara sederhana fungsi sistem dari sudut pandang pengguna. Adapun komponen-komponen dalam *Use case Diagram* diantaranya:

a) *Actor*

Aktor adalah segala hal diluar sistem yang akan menggunakan sistem tersebut untuk melakukan sesuatu. Bisa merupakan manusia, sistem, atau *device* yang memiliki peranan dalam keberhasilan operasi dari sistem.

b) *Use case*

Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada *use case*, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. *Use case* menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem.

c) *System*

Menyatakan batasan sistem dalam relasi dengan *actor-actor* yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). Digambarkan dengan segi empat yang membatasi semua *use case* dalam sistem terhadap pihak mana sistem akan berinteraksi. Sistem disertai label yang menyebutkan nama dari sistem,

tapi umumnya tidak digambarkan karena tidak terlalu *memberi* arti tambahan pada *diagram*.

d) *Association*

Mengidentifikasi interaksi antara setiap *actor* tertentu dengan setiap *use case* tertentu. Digambarkan sebagai garis antara *actor* terhadap *use case* yang bersangkutan. Asosiasi bisa berarah (garis dengan anak panah) jika komunikasi satu arah, namun umumnya terjadi kedua arah (tanpa anak panah) karena selalu diperlukan demikian.

e) *Dependency*

Dependensi <<include>>

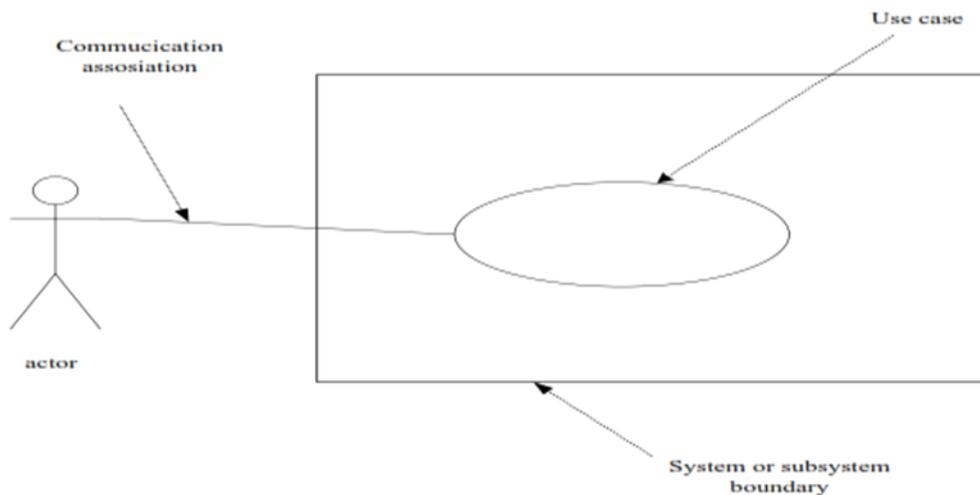
- 1) Mengidentifikasi hubungan antar dua *use case* di mana yang satu memanggil yang lain.
- 2) Jika pada beberapa *use case* terdapat bagian yang memiliki aktivitas yang sama maka bagian aktivitas tersebut biasanya dijadikan *use case* tersendiri dengan relasi dependensi setiap *use case* semula ke *use case* yang baru ini sehingga memudahkan pemeliharaan.
- 3) Digambarkan dengan garis putus-putus bermata panah dengan notasi <<include>> pada garis.
- 4) Arah mata panah sesuai dengan arah pemanggilan

Dependensi <<extend>>

- 1) Jika pemanggilan memerlukan adanya kondisi tertentu maka berlaku dependensi <<extend>>.
- 2) Digambarkan serupa dengan dependensi <<include>> kecuali arah panah berlawanan.
- 3) Note: konsep “extend” ini berbeda dengan “extend” dalam Java.

f) *Generalization*

Mendefinisikan relasi antara dua *actor* atau dua *use case* yang mana salah satunya meng-*inherit* dan menambahkan atau *override* sifat dari yang lainnya. Penggambaran menggunakan garis bermata panah kosong dari yang meng-*inherit* mengarah ke yang di-*inherit*.



Gambar 2.8 Generalization

2. *Activity Diagram*

Activity Diagram adalah sebuah tahapan yang lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dimana biasanya dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. *Activity Diagram* ini sendiri memiliki struktur *diagram* yang mirip *Flowchart* atau *data flow diagram* pada perancangan terstruktur. *Activity Diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *Use case Diagram*.

3. *Class Diagram*

Class diagram merupakan *diagram* yang selalu ada di pemodelan system berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam system yang sedang dibangun dan bagaimana mereka saling

berkolaborasi untuk mencapai satu tujuan. Kelas pada kelas *diagram* terdiri dari 3 bagian utama yaitu nama kelas, isi *property* dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan *generalisasi*. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing komponen penyusun kelas memiliki hak akses seperti *public*, *private* dan *protected*.

Tabel 2.1 Class Diagram

Nama <i>class</i>
Atribut
Operasi/method

Atribut dan Operasi/method dapat memiliki tiga sifat berikut:

1. *Public*, dapat dipanggil oleh *class* apa saja.
2. *Protected*, hanya dapat dipanggil atau diakses oleh *class* yang bersangkutan dan *class* turunannya.
3. *Private*, hanya dapat dipanggil oleh dirinya sendiri (tidak dapat diakses dari luar *class* yang bersangkutan).

4. *Sequence Diagram*

Sequence Diagram adalah suatu *diagram* yang digunakan untuk menggambarkan dan menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*. Adapun urutan proses yang dijelaskan yaitu interaksi yang terjadi antar *class*, operasi yang terlibat, urutan antar operasi, dan informs yang diperlukan oleh masing-masing operasi. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, message yang digambarkan dengan garis penuh, dan waktu yang ditunjukkan dengan progress vertical [20].

