

BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Sistem

Menurut Karcianas dan Hessami yang dijelaskan dalam jurnal (Majd Saleh & Marie-Hélène Abel, 2018 : 3) sebuah sistem adalah organisasi interkoneksi objek yang tertanam dalam lingkungan tertentu. Sedangkan menurut Menurut Rusdiana dan Irfan yang di paparkan dalam jurnal (Heri Kuswara & Alvin Hidayat, 2018 : 2) kata “Sistem” berasal dari bahasa Yunani, yaitu *systema*, yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Selain itu, bisa diartikan sekelompok elemen yang independen, namun saling berkaitan sebagai satu kesatuan.

Suatu sistem pada dasarnya adalah suatu sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Elisabet Yunaeti A, 2017 : 11).

Menurut (Elisabet Yunaeti A, 2017 : 24-25) sistem mempunyai beberapa karakteristik tertentu, antara lain:

1. Komponen sistem (*Component*), komponen sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang saling berkerja sama membentuk komponen sistem.
2. Batasan sistem (*Boundary*), merupakan daerah yang membatasi suatu sistem dengan sistem yang lain atau dengan lingkungan kerjanya.

3. Subsistem (*sub system*), bagianbagian dari sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasarannya masing-masing.
4. Lingkungan luar sistem (*Environment*), suatu sistem yang ada diluar batas sistem yang dipengaruhi oleh operasi sistem.
5. Penghubung sistem (*interface*), media penghubung antara satu sub sistem dengan sub sistem yang lainnya. Adanya penghubung ini memungkinkan berbagai sumber daya mengalir dari suatu subsistem ke subsistem lainnya.
6. Masukan sistem (*input*), energi yang masuk kedalam sistem berupa perawatan dan sinyal. Masukan perawatan adalah energi yang dimasukkan tersebut agar sistem dapat berinteraksi.
7. Keluaran sistem (*output*), hasil energi yang diolah dan di klasifikasi menjadi keluaran yang berguna dan sisa pembuangan.
8. Pengolahan sistem (*process*), suatu sistem dapat mempunyai bagian suatu pengolahan yang akan mengubah masukan menjadi pengeluaran.
9. Sasaran sistem (*object*), tujuan yang ingin dicapai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran atau tujuan.

2.1.2. Informasi

Informasi adalah sekumpulan data atau fakta yang diorganisasi atau diolah dengan cara tertentu sehingga mempunyai arti bagi penerima (Elisabet Yunaeti A, 2017 : 13).



Gambar 2.1 Transformasi data menjadi informasi

Di dalam buku (Elisabet Yunaeti A, 2017 : 15) menuturkan bahwa ciri-ciri informasi yang berkualitas menurut Raymond Mc.Leod adalah:

1. Akurat, informasi harus mencerminkan keadaan yang sebenarnya dan informasi tersebut harus bebas dari kesalahankesalahan.
2. Tepat waktu, informasi tersebut harus tersedia atau ada pada saat informasi tersebut diperlukan dan tidak terlambat.
3. Relevan, informasi yang diberikan harus sesuai dengan yang dibutuhkan.
4. Lengkap, informasi harus diberikan secara lengkap karena jika informasi yang dihasilkan sebagian-sebagian akan memengaruhi dalam mengambil keputusan.

2.1.3. Sistem Informasi

Menurut (M. Rizki Alpiandi, 2016 : 2), Sistem informasi merupakan suatu sistem di dalam suatu organisasi yang berguna sebagai kebutuhan pengolahan transaksi harian, sebagai pendukung operasi proses organisasi. bersifat manajerial dan kegiatan strategi pada suatu organisasi.

Menurut (Elisabet Yunaeti A, 2017 : 2) fungsi sistem informasi antara lain:

1. Untuk meningkatkan aksesibilitas data yang ada secara efektif dan efisien kepada pengguna , tanpa dengan perantara sistem informasi.
2. Memperbaiki produktivitas aplikasi pengembangan dan pemeliharaan sistem.
3. Menjamin tersediannya kualitas dan keterampilan dalam memanfaatkan sistem informasi secara kritis.
4. Mengidentifikasi kebutuhan mengenai keterampilan penukung sistem informasi.
5. Mengantisipasi dan memahami akan konsekuensi ekonomi.
6. Menetapkan investasi yang akan diarahkan kepada sistem informasi.
7. Mengembangkan proses perencanaan yang efektif.

Di dalam buku (Elisabet Yunaeti A, 2017 : 4) juga mengemukakan beberapa contoh sistem informasi antara lain:

1. Sistem referansi pesawat terbang , digunakan dalam biro penerbangan untuk melayani pemesana atau pembelian tiket.

2. Sistem untung menangani penjualan kredit kendaraan bermotor sehingga dapat memantau hutang para pelanggan.
3. Sistem biometrik yang mencegah orang yang tidak berkenan memasuki fasilitas-fasilitas rahasia atau mengakses informasi yang bersifat rahasia dengan cara menganalisis sidik jari dan retina mata.
4. Sistem *Point of Sale* (POS), diterapkan banyak swalayan dengan dukungan pembaca barcode untuk untuk mempercepat pemasukan data.
5. Sistem telemetri atau pemantauan jarak jauh yang menggunakan teknologi radio, misalnya untuk mendapatkan suhu lingkungan pada gunung berapi atau untuk memantau getaran pilar jembatan rel kereta api.
6. Sistem berbasis kartu cerdas, dapat digunakan juru medis untuk mengetahui riwayat penyakit pasien karena di dalam kartu tersebut terekam data-data mengenai pasien.
7. Sistem layanan akademis berbasis web yang memungkinkan mahasiswa memperoleh data-data akademis atau bahkan dapat mendaftarkan mata kuliah yang diambil pada semester baru.
8. Sistem pertukaran data elektronik (*electronic data interchange* atau EDI) memungkinkan pertukaran dokumen antar perusahaan secara elektronik dan data yang terkandung dalam dokumen dapat diproses secara langsung oleh komputer.

2.1.4. UML

Menurut (M Teguh Prihandoyo, 2018 : 2) *Unified Modeling Language* merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. Menurut (Ibnu Akil, 2018:4) mendefinisikan bahwa UML adalah Bahasa pemodelan *visual* yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak.

Menurut Booch. G yang diuraikan dalam jurnal (M Teguh Prihandoyo, 2018:2) Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem, yaitu:

1. *Use case Diagram*, Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat *actor* yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.
2. *Activity Diagram*, Merupakan gambaran alir dari aktivitasaktivitas didalam sistem yang berjalan.
3. *Sequence Diagram*, Menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu.

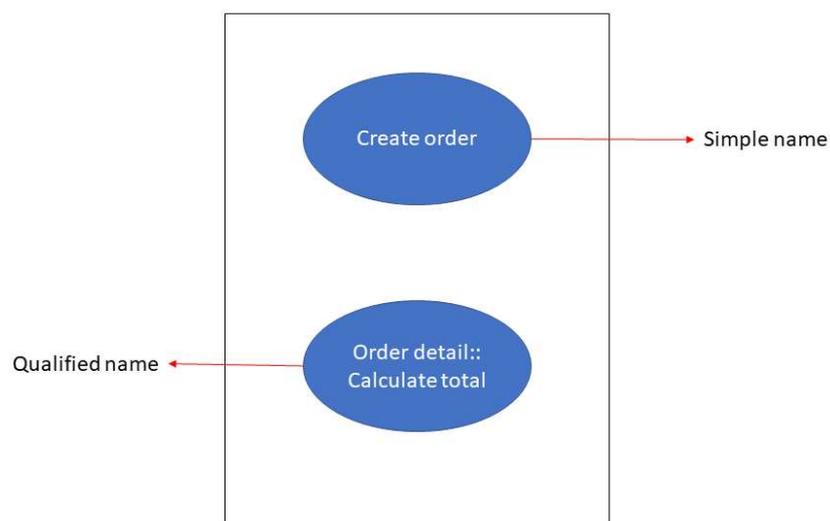
4. *Class Diagram*, Merupakan gambaran struktur dan deskripsi dari *class*, *package*, dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

1. *Use Case Diagram*

Menurut (Ibnu Akil, 2018 : 106-114) Diagram *use case* menangkap tingkah laku sistem, subsistem, kelas, atau komponen yang tampak kepada *eksternal entity*.

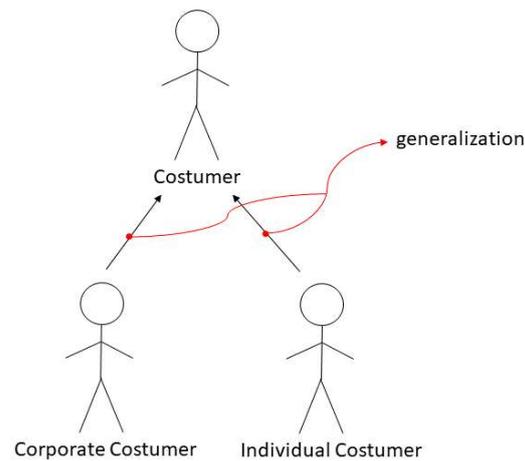
1. Notasi *use case*

- a. *Use case*, adalah sebuah unit eksternal dari sistem (berupa antar muka) yang akan menerima perintah dari seorang *actor* berupa sebuah *event*.



Gambar 2.2 Penamaan *use case*

- b. *Actor*, mempresentasikan suatu set peranan yang dimainkan oleh orang luar, unit kerja, atau hal-hal yang berinteraksi dengan sistem, subsistem atau kelas.

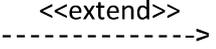
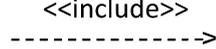


Gambar 2.3 Aktor

- c. *Relationships*, sebuah use case dapat berpartisipasi dengan use case lain dalam beberapa jenis hubungan sebagai tambahan Ketika berasosiasi dengan sebuah aktor.

Tabel 2.1 Jenisjenis *relationship* dalam *use case*

Relationship	Function	Notation
<i>Accociation</i>	Sebuah jalur komunikasi antara seorang <i>actor</i> dengan satu <i>use case</i> dimana <i>actor</i> tersebut berpartisipasi	_____

<i>Extend</i>	Penyisipan sebuah fungsionalitas tambahan kedalam sebuah <i>use case</i> dasar yang bersifat opsional.	
<i>Include</i>	Penyisipan sebuah fungsionalitas tambahan kedalam sebuah <i>use case</i> dasar dimana fungsionalitas tersebut bersifat <i>mandatory</i> (wajib).	
<i>Use case generalization</i>	Sebuah hubungan antara sebuah <i>use case</i> umum dengan <i>use case</i> yang lebih spesifik dimana <i>use case</i> yang lebih spesifik mewarisi <i>use case</i> umum tersebut.	

- d. Skenario *use case*, sebuah *use case* menjelaskan apa yang sistem atau sub sistem lakukan, akan tetapi *use case* tidak menjelaskan secara spesifik bagaimana sistem itu melakukannya.

Misalkan dalam sebuah sistem *e-commerce* salah satu *use case* yang ada mungkin *use case checkout* seperti berikut.

Tabel 2.2 Contoh skenario *use case*

<i>Use case name</i>	<i>Checkout</i>
<i>Goal</i>	<i>User</i> dapat melakukan pembelian secara online.
<i>Pre-conditions</i>	<i>User</i> telah memilih barang.
<i>Post-conditions</i>	Sistem mengirim rincian faktur penjualan via email.
<i>Failed end condition</i>	<i>User</i> membatalkan <i>checkout</i> . Kemungkinan <i>user</i> ingin menambah barang atau membatalkan pembelian.
<i>Primary actors</i>	<i>user</i>
<i>Main flow / basic path</i>	1. <i>User</i> memilih <i>icon checkout</i> . 2. Sistem menampilkan rincian belanja.

	<p>3. <i>User</i> menyetujui dan melanjutkan.</p> <p>4. Sistem mengecek apakah <i>user</i> sudah terdaftar atau belum. Jika belum maka sistem akan menampilkan halaman registrasi. Jika sudah sistem akan menampilkan halaman alamat pengiriman.</p> <p>5. <i>User</i> memasukan alamat pengiriman dan melanjutkan.</p> <p>6. Sistem menampilkan pilihan jasa kurir pengiriman dan biayanya.</p> <p>7. <i>User</i> memilih jasa kurir pengiriman dan melanjutkan.</p> <p>8. Sistem menampilkan pilihan metode pembayaran.</p>
--	---

	<p>9. <i>User</i> memilih metode pembayaran dan melanjutkan.</p> <p>10. Sistem memproses penjualan dan mengirimkan rincian penjualan ke email <i>user</i>.</p>
<i>Alternative flow</i>	<p>9. <i>User</i> memilih metode pembayaran menggunakan kartu kredit.</p> <p>10. Sistem menampilkan jenis pilihan kartu kredit.</p> <p>11. <i>User</i> memilih jenis kartu kredit dan memasukan nomor kartu kredit.</p> <p>12. Sistem memverifikasi kartu kredit.</p> <p>13. Jika valid sistem mendebet kartu kredit.</p> <p>14. Jika tidak valid sistem kembali ke halaman pemilihan metode pembayaran.</p>

<i>Use case name</i>	:	Adalah nama <i>use case</i> .
<i>Goal</i>	:	Adalah tujuan dari <i>use case</i> tersebut.
<i>Pre-condition</i>	:	Adalah kondisikondisi yang harus terpenuhi sebelum <i>use case</i> tersebut di eksekusi.
<i>Post-condition</i>	:	Adalah kondisikondisi yang harus terpenuhi setelah <i>use case</i> tersebut di eksekusi.
<i>Failed end condition</i>	:	Adalah kondisikondisi yang harus terpenuhi apabila terjadi kegagalan dalam eksekusi <i>use case</i> .
<i>Primary actor</i>	:	Adalah <i>actor</i> utama yang berpartisipasi dalam <i>use case</i> tersebut.
<i>Main flow/basic path</i>	:	Adalah arus kerja utama dengan asumsi semua berjalan lancar.
<i>Alternate flow</i>	:	Adalah arus kerja alternatif dari arus kerja utama.

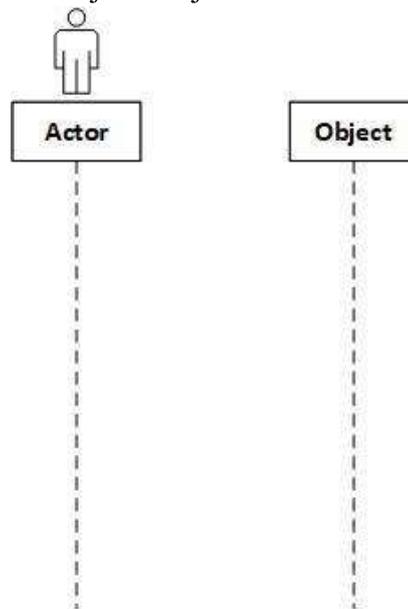
2. *Sequence Diagram*

Di dalam buku (Ibnu Akil, 2018 : 124-136) *Sequence diagram* menggambarkan tingkah laku dari suatu skenario tunggal. Diagram ini

menunjukkan objek-objek yang terlibat dalam proses tersebut dan bagaimana urutan penyampaian pesan antara objek-objek tersebut.

Notasi *sequence diagram*

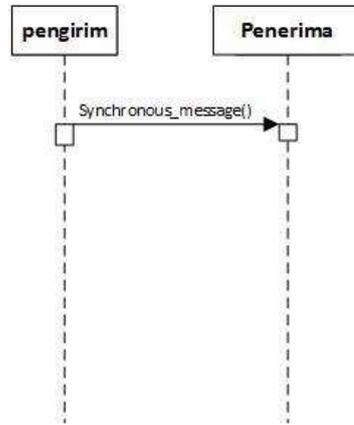
- a. *Lifeline stereotype icon*, bagian kepalanya untuk mengklasifikasikan jenis objek.



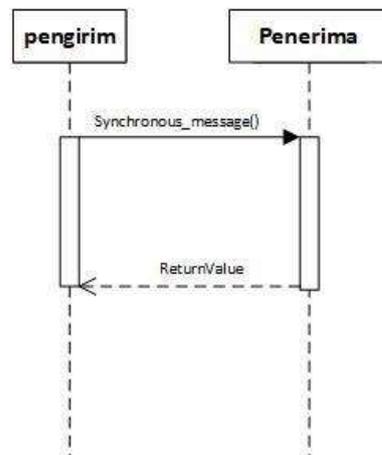
Gambar 2.4 *Stereotip lifeline*

- b. *Messages* (pesan), Ketika satu objek menyampaikan pesan ke objek lain ini digambarkan dengan tanda panah diantara garis hidup.

- c. *Synchronous message*, digunakan Ketika sipengirim menunggu sampai penerima selesai memproses pesan.

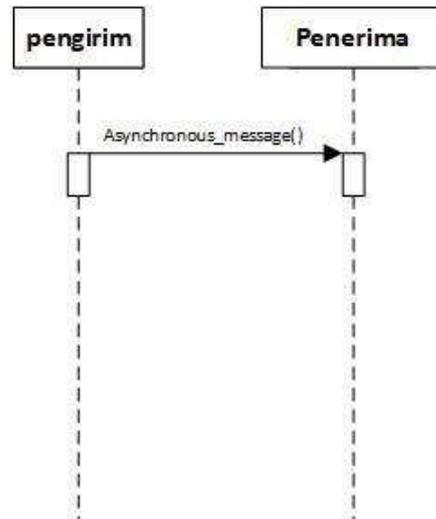


Gambar 2.5 *Synchronous message*



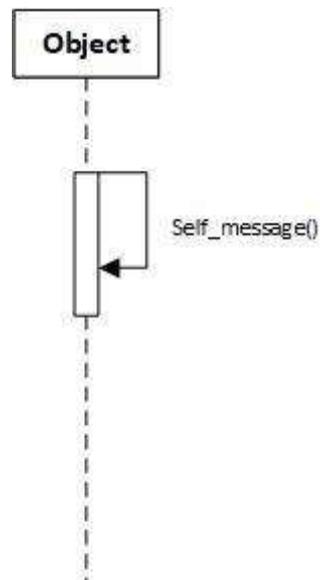
Gambar 2.6 *Synchoronous message dengan return value*

- d. *Asynchronous message*, si pengirim tidak menunggu si penerima selesai memproses pesan.



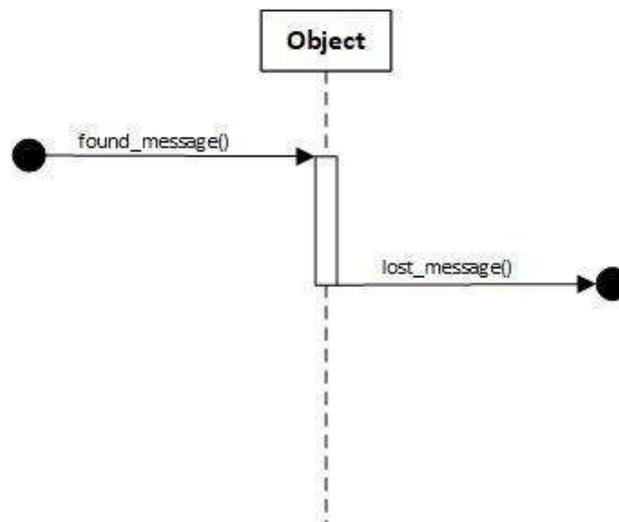
Gambar 2.7 *Asynchronous message*

- e. *Self message*, sebuah pesan yang dikirim kepada objek itu sendiri.



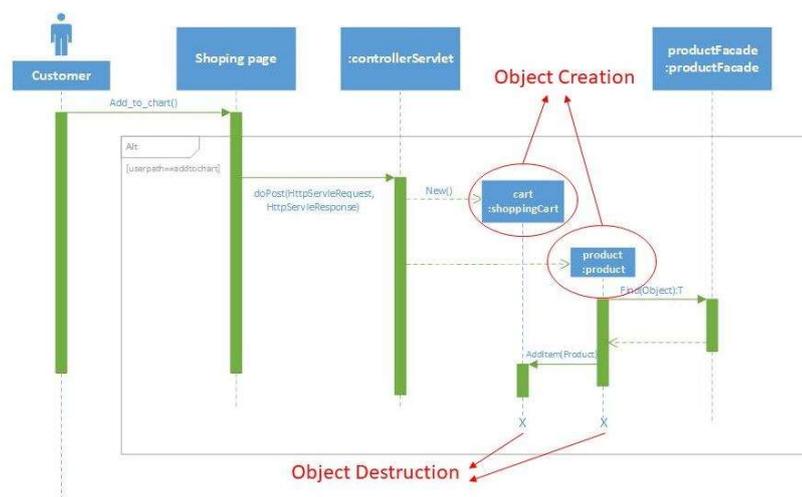
Gambar 2.8 *Self message*

- f. *Lost and found message*, pesan dimana sipemanggil tidak diperlihatkan atau tidak penting.



Gambar 2.9 *Lost and found message*

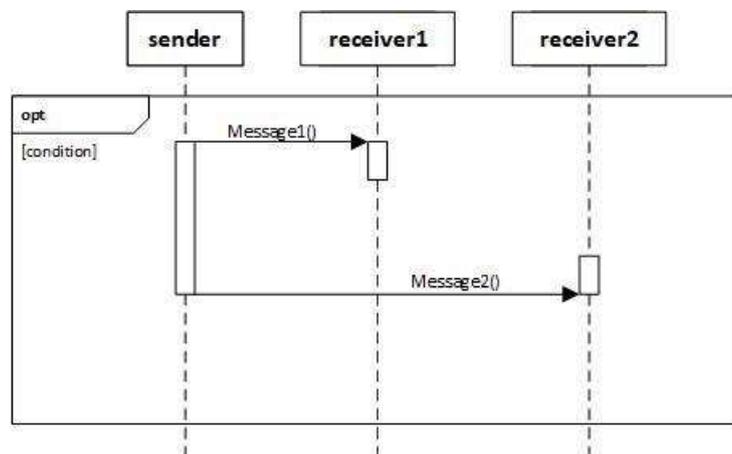
- g. Penciptaan dan penghapusan objek, objek yang ada pada awal sebuah interaksi ditempatkan diatas diagram. Sedangkan objek lain ditempatkan selama interaksi dibawah pada saat objek tersebut diciptakan.



Gambar 2.10 Instansiasi dan Penghapusan

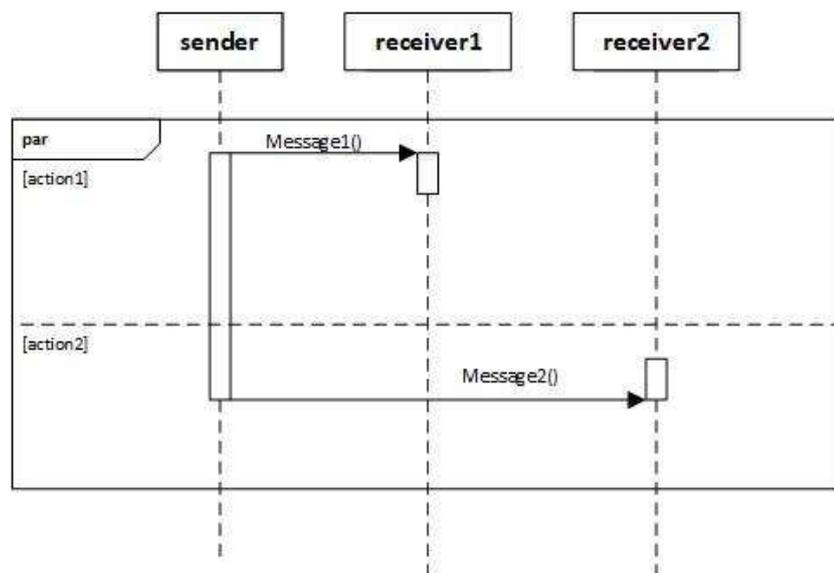
- h. *Fragment*, digambarkan sebagai kotak dengan label dipojok kiri atas yang berisi jenis operator control.

Eksekusi Opsional

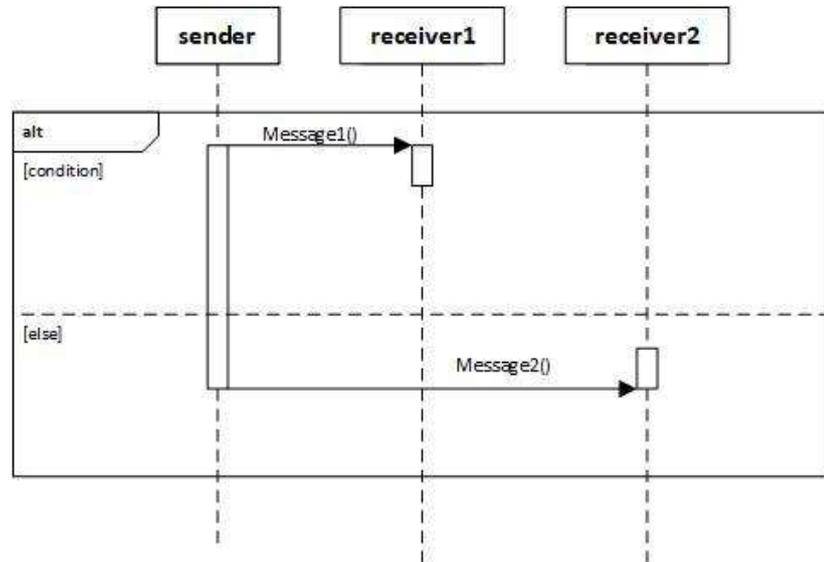
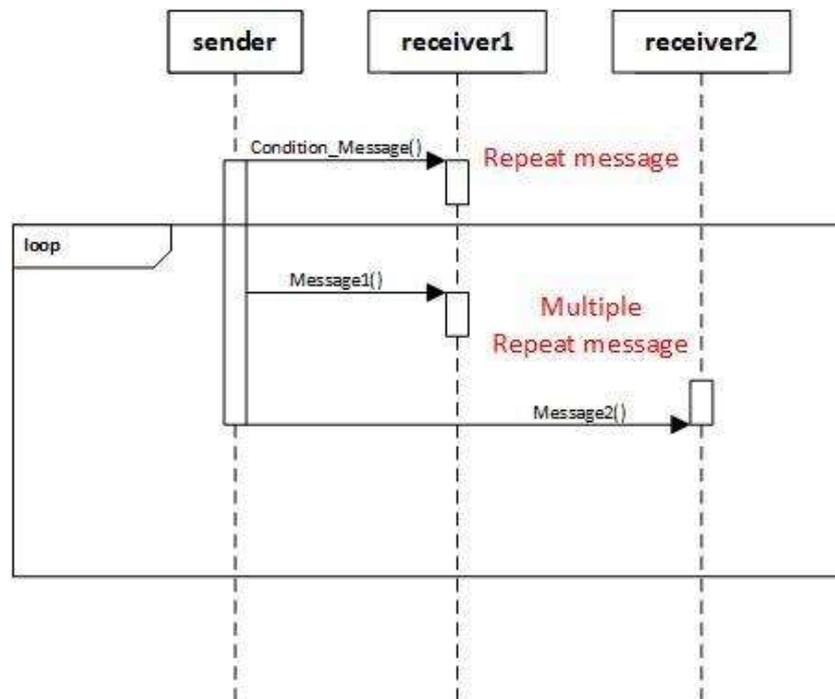


Gambar 2.11 Eksekusi opsional

Eksekusi Kondisional



Gambar 2.12 Eksekusi Kondisional

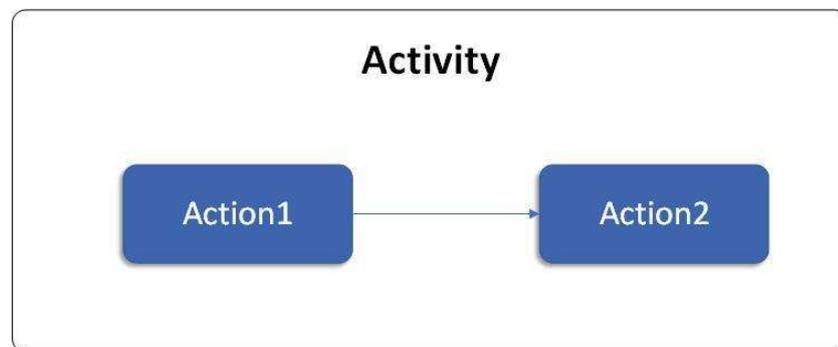
Eksekusi *Parallel*Gambar 2.13 Eksekusi *parallel*Eksekusi *Loop*Gambar 2.14 Eksekusi *loop*

3. *Activity Diagram*

Menurut (Ibnu Akil, 2018 : 143-152) kelebihan *activity diagram* dibandingkan dengan *flowchart* adalah *activity diagram* mendukung proses yang berjalan secara *parallel*, sedangkan *flowchart* tidak.

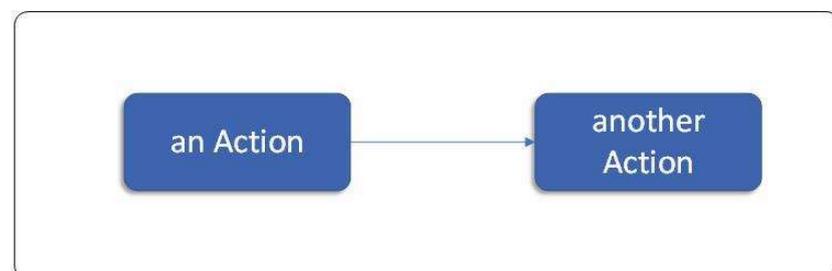
Notasi Elemen

- a. *Action dan activity node, activity* dapat berisi *group-group* dan *action-action* atau kumpulan *activity* yang lain.



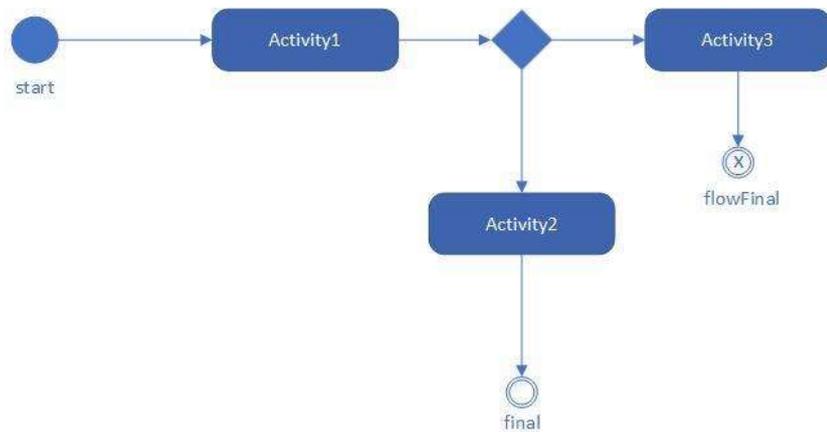
Gambar 2.15 *Action dan activity node*

- b. *Control Flow*, menunjukkan arus kendali dari satu *action* ke *action* lainnya. Ketika satu *action* atau *activity* selesai dieksekusi *flow* akan berlanjut ke *action* atau *activity* lainnya.



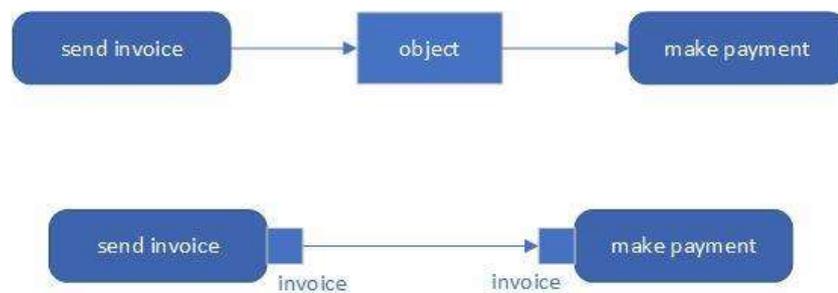
Gambar 2.16 *Control flow*

- c. *Initial node, final node dan flow final, initial node* adalah titik awal serangkaian *activity* atau *actions*. *Final node* adalah titik akhir dari serangkaian *activity* atau *action*.



Gambar 2.17 *Initial Node*

- d. *Object dan object flows*, jalur dimana *object* atau data dilewatkan. Objek ditunjukkan sebagai kotak.



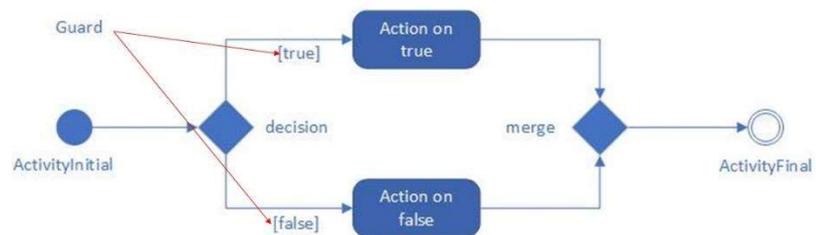
Gambar 2.18 *object dan object flows*

- e. *Data store*, elemen yang digunakan untuk menyimpan data secara permanen.



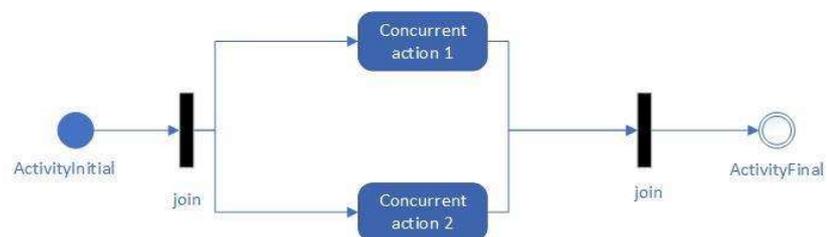
Gambar 2.19 *Data store*

- f. *Decision dan merge nodes*, memiliki notasi yang sama yaitu belah ketupat. Sebuah *flow* akan membagi *decision* menjadi dua diaman hanya ada satu *flow* yang dilalui apabila kondisi *guard* terpenuhi, sedangkan *merge* adalah tempat bergabung kembali *flow* yang terpisah karena decision.



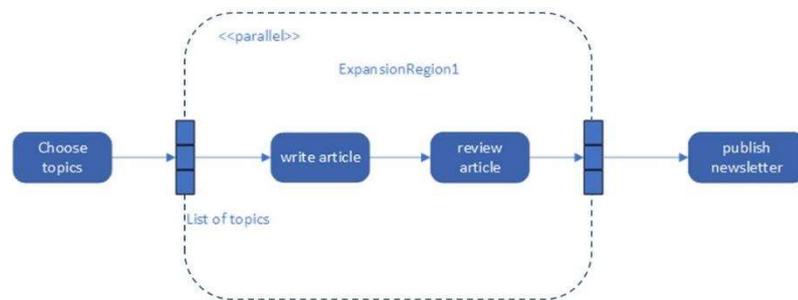
Gambar 2.20 *Decision dan merge nodes*

- g. *Fork dan join nodes*, mengindikasikan proses yang *parallel* atau *concurrent*. *Fork* untuk percabangan proses yang dieksekusi secara *parallel* atau bersamaan, sedangkan *join* sebagai titik temu prosesproses yang *parallel* menjadi satu *flow*, diaman *flow* ini tidak akan berjalan sebelum semua proses selesai.



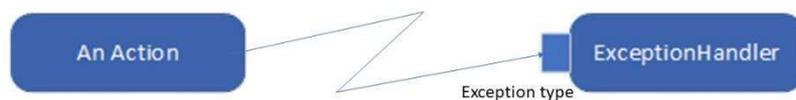
Gambar 2.21 *Fork dan Join nodes*

- h. *Expansion region*, sebuah area *activity* yang terstruktur yang dieksekusi berkali-kali. Contoh ketika anda akan menulis sebuah artikel, anda akan memilih topik atau kategori, kemudian mereviewnya baru di *publish*. Mungkin anda akan menulis lagi dengan topik yang berbeda dan melakukan aktivitas yang sama (menulis dan mereview) secara berulang.



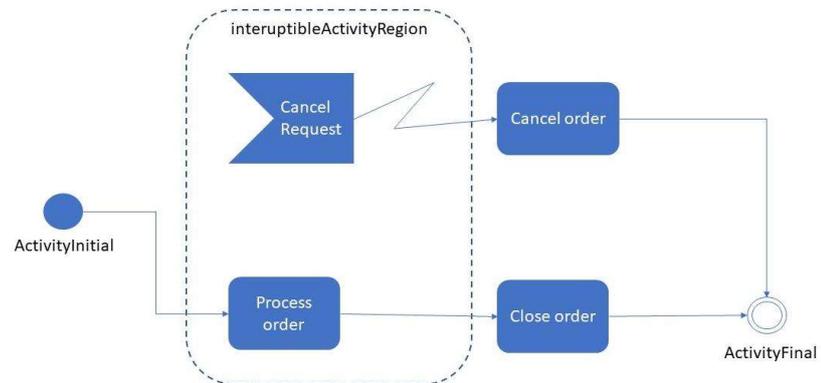
Gambar 2.22 *Expansion Region*

- i. *Exception handles*, penanganan kesalahan.



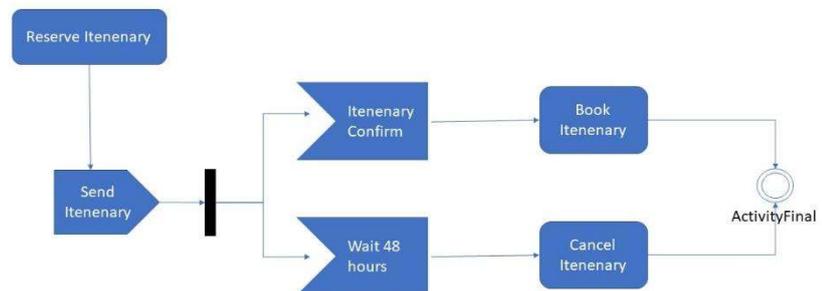
Gambar 2.23 *Expansion handles*

- j. *Interruptible activity region*, biasanya berisi *actions* yang dapat di interupsi.



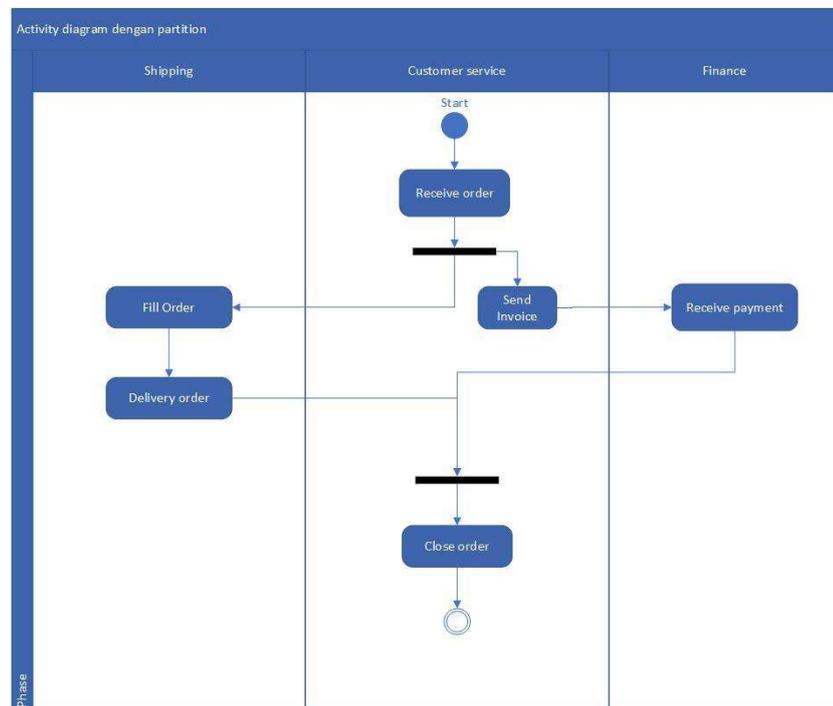
Gambar 2.24 *Interruptible activity region*

- k. *Signals*, signal bahwa *activity* menerima sebuah *event* dari luar proses. Hal ini berarti sebuah *activity* dapat dilanjutkan apabila telah menerima signal.



Gambar 2.25 *signals*

- l. *Partition* atau *swimlane*, untuk menunjukkan *actor* atau objek yang bertindak atas suatu *activity* atau *action* dan menempatkan aktivitas tersebut pada *partition* yang bersangkutan.



Gambar 2.26 activity diagram dengan partition

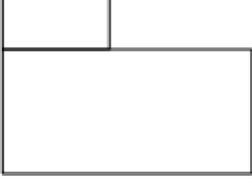
4. Class Diagram

Menurut (Yuni Sugiarti, 2018 : 120-123) *class diagram* menggambarkan struktur sistem dari segi pendefinisian *class* yang akan dibuat untuk membangun sistem. *class diagram* menggambarkan struktur dan deskripsi kelas, *package*, dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lainlain.

Berikut simboisymbol yang ada dalam *class diagram*.

Tabel 2.3 simboisymbol *class diagram*

Simbol	Deskripsi
<i>Package</i>	Merupakan sebuah bungkus dari satu atau lebih <i>class</i> .

	
<p>Operasi</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>Nama Kelas</p> <p>+ Attribute 1</p> <p>+ Attribute 2</p> <p>+ Operation()</p> </div>	<p><i>Class</i> pada struktur sistem.</p>
<p>Antarmuka/<i>interface</i></p> 	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>
<p>Asosiasi</p> 	<p>Relasi antar <i>class</i> dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah/<i>directed</i></p> 	<p>Relasi antar <i>class</i> dengan makna <i>class</i> yang satu digunakan oleh <i>class</i> yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Generalisasi</p> 	<p>Relasi antar <i>class</i> dengan makna generalisasi spesialisasi (umum khusus).</p>
<p>Kebergantungan/<i>dependency</i></p> 	<p>Relasi antar <i>class</i> dengan makna kebergantungan.</p>
<p>Agresasi</p>	<p>Relasi antar <i>class</i> dengan makna semua bagian (<i>whole part</i>).</p>



2.2. Teori Khusus

2.2.1. Sistem informasi akademik

Menurut Triyono yang dikemukakan di dalam jurnal (Saut Horas Tua Siagian & Effiyaldi, 2018 : 3) Kata akademik berasal dari bahasa Yunani yakni *Academos* yang berarti sebuah taman umum (plasa) di sebelah barat laut kota Athena. Nama *Academos* adalah nama seorang pahlawan yang terbunuh pada saat perang legendaris Troya. Pada plasa inilah filosof Socrates berpidato dan membuka arena perdebatan tentang berbagai hal. Tempat ini juga menjadi tempat Plato melakukan dialog dan mengajarkan pikiran-pikiran filosofinya kepada orang-orang yang datang. Sesudah itu, kata *Academos* berubah menjadi akademik, yaitu semacam tempat perguruan. Para pengikut perguruan tersebut disebut *Academist*, sedangkan perguruan semacam itu disebut *Academia*. Berdasarkan hal ini, inti dari pengertian akademik adalah keadaan orang-orang bisa menyampaikan dan menerima gagasan, pemikiran, ilmu pengetahuan, dan sekaligus dapat mengujinya secara jujur, terbuka dan leluasa.

Menurut (Riandy Pratama et al., 2017 : 2) Sistem Informasi Akademik adalah sistem yang dirancang untuk mengolah data-data yang berhubungan dengan informasi akademik, seperti data mahasiswa, karyawan, rekaman prestasi, kurikulum, dan jadwal kelas. Sedangkan menurut (Ahmad Homaidi, 2016 : 2) Sistem informasi akademik adalah sistem yang dirancang untuk melakukan

pengolahan data akademik baik melalui *software* ataupun *hardware* sehingga proses kegiatan akademik dapat dikelola dengan baik sehingga menjadi informasi yang bermanfaat untuk manajemen perguruan tinggi dan pengambilan keputusan oleh eksekutif.

2.2.2. Website

Menurut (Andi Christian et al., 2018 : 1) website, merupakan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi berbentuk data digital baik berupa teks, gambar, video, audio dan animasi lainnya yang diakses dengan koneksi internet.

Menurut Hidayat yang dijelaskan dalam jurnal (Juansyah, 2019 : 3) *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, animasi, suara atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

Menurut (Nurhadi, 2017 : 8-9) alasan banyak orang membuat *website*, diantaranya:

1. Memperluas jangkauan promosi sesuatu, dengan memiliki *website* maka produk kita lebih bisa dikenal oleh masyarakat khususnya pengguna internet.
2. Bisa menjadi media tanpa batas, sebab internet adalah media informasi yang tanpa batas.

3. Promosi yang luas, internet adalah suatu media promosi terluas di dunia jika dilihat dari jangkauan area.
4. Media pengenalan perusahaan, jika kita memiliki suatu perusahaan akan lebih mudah kita mengenalkan perusahaan melalui *website*.

Di dalam buku (Nurhadi, 2017 : 12-15) juga menjelaskan bahwa terdapat macam-macam bahasa pemrograman *website* antara lain:

1. HTML (*HyperText Markup Language*), merupakan bahasa standar yang digunakan untuk membuat suatu dokumen atau halaman web, terdiri dari kode-kode singkat tertentu yang akan ditampilkan oleh *web browser*.
2. JavaScript, adalah bahasa *script* yang dapat berkerja di sebagian besar *browser*. JavaScript dapat disisipkan dalam halaman web menggunakan tag *SCRIPT*.
3. PHP (*Hypertext Preprocessor*), adalah bahasa pemrograman yang digunakan secara luas untuk penggunaan, pembuatan, dan pengembangan sebuah situs web yang bisa digunakan bersamaan dengan HTML.
4. JavaServlet (JSP), merupakan teknologi *server side java*.
5. XML (*Extensible Markup Language*), adalah bahasa *markup* untuk keperluan umum yang disarankan oleh W3C, untuk membuat dokumen *markup* keperluan pertukaran data antar sistem yang beraneka ragam.

6. WML (*Wireless Markup Language*), bahasa *markup* yang digunakan untuk mengimplementasikan WAP dokumen WML berbasis XML, di mana versinya mempunyai DTD dokumen type definition. WML merupakan analog dari HTML yang berjalan pada protokol nirkabel.
7. CSS (*Cascading Style Sheet*), merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Sama halnya Microsoft Word yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersamaan dalam beberapa berkas.
8. AJAX (*Asynchronous JavaScript And XML*), adalah bahasa pemrograman yang membuat *website* lebih interaktif.
9. ASP (*Active Server Pages*), adalah pengaturan web yang digunakan untuk membina halaman web yang dinamik dan bersifat interaktif. *Active server pages* adalah fasilitas yang diberikan oleh Microsoft untuk memudahkan pembuatan aplikasi aplikasi *web server*.
10. ColdFusion, bahasa pemrograman ini mirip dengan HTML menggunakan tag untuk membentuk blok-blok programnya.
11. JQuery, adalah pustaka JavaScript kecil bersumber terbuka yang menekankan pada interaksi antara JavaScript dan HTML.

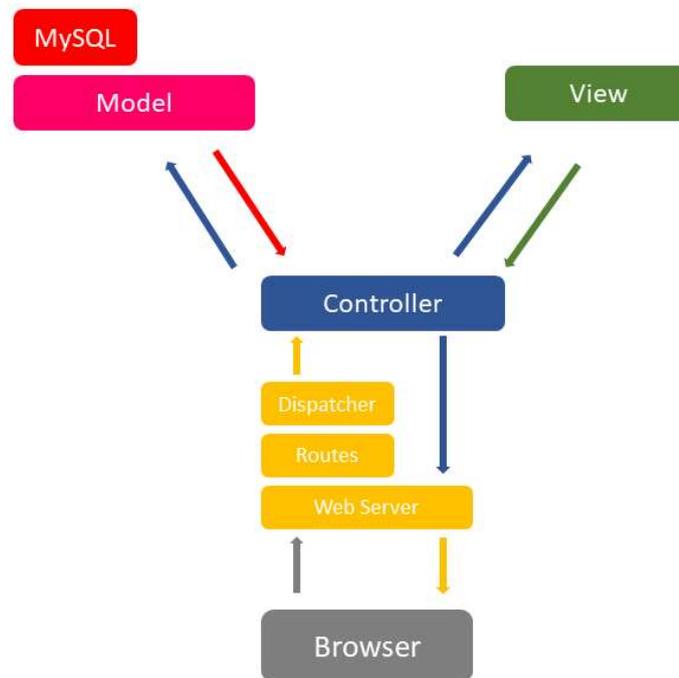
2.2.3. CodeIgniter

Menurut (Supono Vidiandry P, 2016 : 109) CodeIgniter adalah aplikasi open source berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP.

1. **Komponen MVC *pattern***

Di dalam (Supono Vidiandry P, 2016 : 110) menjelaskan bahwa terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam satu aplikasi yaitu:

- a. *Model*, biasanya berhubungan langsung dengan database untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
- b. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan mempresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian *model*.
- c. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari user kemudian menentukan apa yang akan diproses aplikasi.



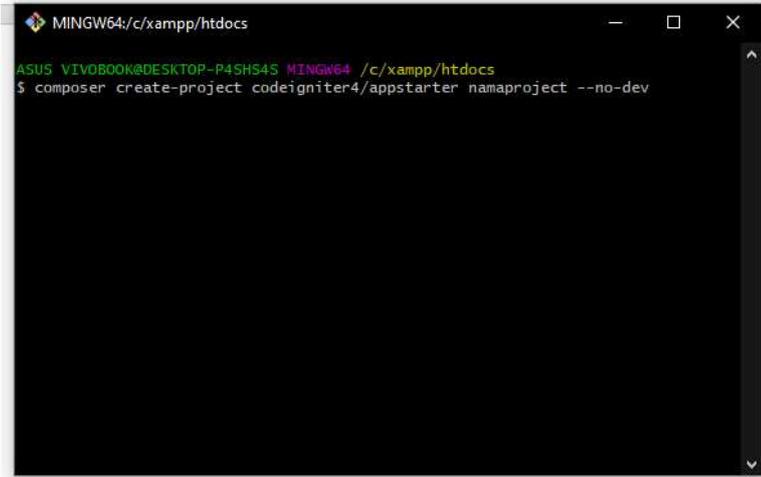
Gambar 2.27 Cara Kerja MVC

2. Instalasi CodeIgniter

CodeIgniter yang penulis gunakan untuk merancang aplikasi adalah yang versi terbaru yaitu CodeIgniter 4. Tahap instalasi CodeIgniter 4 berbeda dengan CodeIgniter 3 dimana CodeIgniter 4 bisa di instal secara online melalui terminal sehingga jika ada pembaruan dari *framework* maka kita tidak perlu lagi mengganti *framework* dan mengkonfigurasikannya lagi dengan website yang sudah kita rancang melainkan kita cukup melakukan update melalui terminal atau composer secara online dan langsung terintegrasi otomatis kepada website kita.

Adapun tahap peng-installan framework CI4 adalah sebagai berikut:

- a. Sebelum menginstall CI4 pastikan kita sudah menginstal terlebih dahulu composer di computer kita. yaitu bisa di dapatkan di <https://getcomposer.org/>.
- b. Dan pastikan kita sudah memiliki *local server* di computer kita. yang paling sering digunakan adalah XAMPP untuk pengguna Windows dan Linux serta MAMPP untuk pengguna Mac.
- c. Arahkan folder melalui command prompt ke C:\xampp\htdocs atau di tempat kita menginstall *local server* nya dan ketikkan `$ composer create-project codeigniter4/appstarter namaproject --no-dev`. Lalu tekan enter dan tunggu prosesnya.

A screenshot of a terminal window titled 'MINGW64:/c/xampp/htdocs'. The prompt shows the user 'ASUS VIVOB00K@DESKTOP-P4SHS45' in the directory 'MINGW64 /c/xampp/htdocs'. The command entered is '\$ composer create-project codeigniter4/appstarter namaproject --no-dev'. The terminal background is black with green text for the prompt and command, and white text for the output area which is currently empty.

Gambar 2.28 installasi CodeIgniter 4

- d. *Framework* berhasil diinstal.

2.2.4. Bootstrap

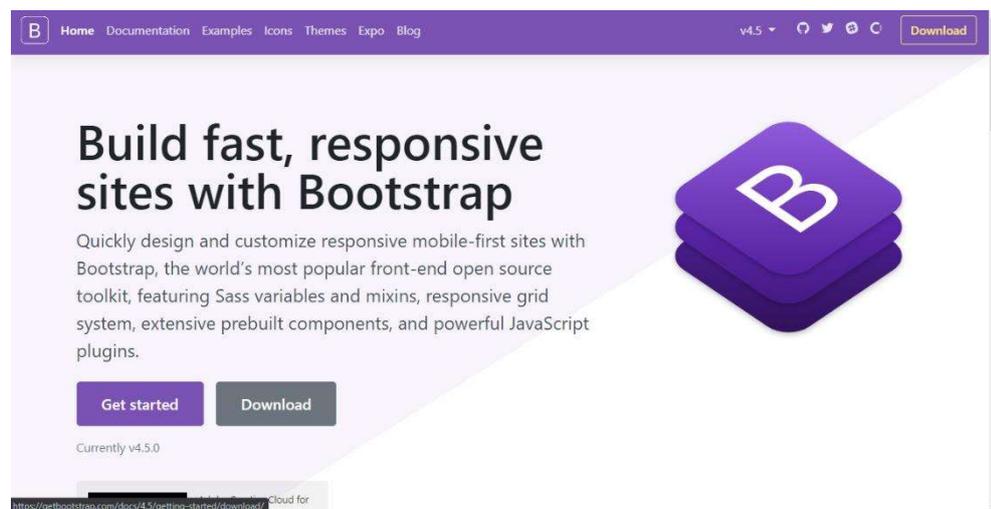
Menurut (Christian et al., 2018 : 1) Bootstrap adalah paket aplikasi siap pakai untuk membuat *interface* sebuah website, bisa dikatakan, bootstrap adalah *template* disain web dengan fitur lebih. Bootstrap dibuat adalah untuk mempermudah proses

disain web bagi berbagai tingkat pengguna, mulai dari tingkat pemula hingga tingkat profesional. Cukup memahami pengetahuan dasar mengenai HTML dan CSS.

Didalam buku (Wahana Komputer, 2016 : 115) mendefinisikan bahwa bootstrap adalah sebuah *framework* CSS yang menyediakan komponen-komponen *interface* siap pakai dan telah dirancang sedemikian rupa untuk keperluan disain halaman website yang artistik.

1. Instalasi bootstrap

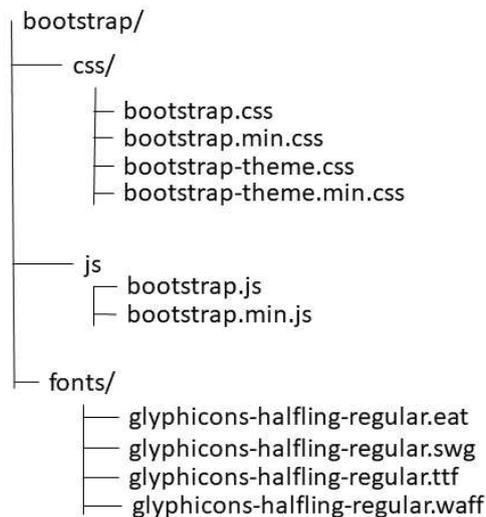
Download file *framework* di www.getbootstrap.com.



Gambar 2.29 instalasi bootstrap

Setelah itu ekstrak filenya dan *copy* folder css dan js kedalam *website* kita.

2. Struktur file bootstrap



Gambar 2.30 Struktur bootstrap

- a. Bootstrap.css, adalah file CSS utama yang harus disertakan dalam halaman HTML yang anda buat.
- b. Bootstrap.min.css, adalah hasil *minify/compress* dari bootstrap.css. file ini digunakan jika disain situs yang dibuat untuk online.
- c. Bootstrap-theme.css, diibaratkan suatu kebutuhan skunder. Fungsinya untuk memberikan efek 3D pada tombol dan beberapa elemen lainnya.
- d. Bootstrap-theme.min.css, adalah hasil *minify/compress* dari bootstrap-theme.css. file ini digunakan jika disain situs yang dibuat untuk online.
- e. Bootstrap.js, adalah library javascript untuk memberikan efek seperti modal, dropdown, scrollspy, dan lain sebagainya.
- f. Bootstrap.min.js, digunakan sebagai pengganti bootstrap.js jika situs yang anda buat untuk online.

2.2.5. PHP

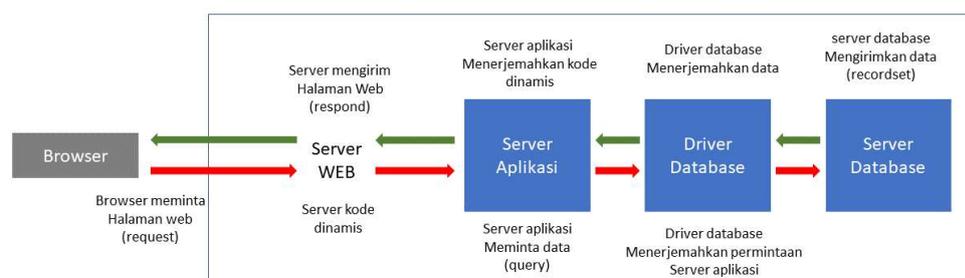
Menurut (Dewi Mustari et al., 2016 : 2) *Hypertext Preprocessor* (PHP) adalah bahasa skrip yang dapat dimasukkan atau disisipkan ke dalam HTML. PHP populer untuk memrogram website dinamis. PHP bisa digunakan untuk membangun sebuah CMS.

Sedangkan menurut (Supono Vidiandry P, 2016 : 3) PHP adalah bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat di tambahkan kedalam HTML.

Menurut (Tukino, 2018 : 3) Sistem kerja PHP diawali dengan permintaan yang berasal dari halaman website oleh browser. Berdasarkan URL atau alamat website dalam jaringan internet, browser akan menemukan sebuah alamat dari webserver, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh webserver.

1. Prinsip Kerja PHP

Di dalam buku (Supono Vidiandry P, 2016 : 4) menjelaskan bahwa prinsip kerja PHP memerlukan penerjemah dalam hal ini *web server* untuk menjalankannya.



Gambar 2.31 Prinsip Kerja PHP

Di dalam buku (Jubilee Enterprise, 2017 : 12) menjelaskan bahwa syntax dasar php yaitu sebagai berikut:

1. Cara menulis komentar. Terdapat dua cara menambahkan komentar pada PHP yaitu dengan menambahkan garis miring diawal komentar

```
2
3 // komentar 1 baris
4
5 /*komentar lebih dari
6 satu baris*/
7
```

dan menambahkan tanda garis miring dan bintang untuk komentar yang lebih dari 2 kata.

Gambar 2.32 komentar pada PHP

2. Titik koma. Baris perintah php harus diakhiri dengan titik koma (;).
3. Simbol \$. Digunakan untuk mendeklarasikan variable.

```
2
3 $i = 10;|
4
```

Gambar 2.33 aturan penulisan dasar PHP

2.2.6. MySQL

Menurut (Andi Christian et al., 2018 : 2) MySQL adalah *software* RDBMS (*server database*) yang dapat mengelola basis data dengan sangat cepat, dapat menampung data dalam jumlah sangat banyak, dapat diakses oleh banyak user

(*multi-user*) dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

Menurut (Iwan Purnama, 2019 : 3) menjelaskan bahwa website yang populer menggunakan MySQL adalah MediaWiki (perangkat lunak yang dipakai Wikipedia dan proyekproyek sejenis), PHP-Nuke yang berfungsi sebagai komponen database dalam LAMP. Popularitas MySQL sebagai aplikasi web dikarenakan kedekatannya dengan PHP, sehingga MySQL dan PHP seringkali disebut sebagai *Dynamic Duo*.

Menurut (Yanto, 2016 : 16-17) keunggulan dari MySQL adalah:

1. Cepat.
2. Memudahkan bagi user dalam penggunaannya.
3. Bersifat *open source*.
4. Support dengan Bahasa query.
5. Pengguna dapat mengakses lebih dari satu dalam satu waktu.
6. Pengaksesan dapat dilakukan di setiap tempat dengan menggunakan internet.
7. MySQL mudah di dapatkan karena source code yang dapat disebarluaskan.

MySQL bisa kita gunakan pada saat kita menginstal *local server* XAMPP kita pada komputer atau PC yang dapat diakses melalui browser yaitu PHPMyAdmin.