

BAB II

LANDASAN TEORI

2.1. Data

Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas, dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. Data dapat berupa nilai yang terformat, teks, citra, audio, dan video

1. Data yang terformat adalah data dengan suatu format tertentu. Misalnya, data yang menyatakan tanggal atau jam, atau menyatakan nilai mata uang.
2. Data teks adalah sederetan huruf, angka, dan simbol – simbol khusus (misalnya + dan \$) yang kombinasinya tidak tergantung pada masing – masing item secara individual. Contoh teks adalah artikel Koran.
3. Data citra (image) adalah data dalam bentuk gambar. Citra dapat berupa grafik, foto, hasil rontgen, dan tanda tangan, ataupun gambar yang lain
4. Data audio adalah data dalam bentuk suara. Instrumen music, suara orang atau suara binatang, gemericik air, detak jantung merupakan beberapa contoh data audio.
5. Data video menyatakan data dalam bentuk sejumlah gambar yang bergerak dan bisa saja dilengkapi dengan suara. Video dapat digunakan untuk mengabadikan suatu kejadian atau aktivitas. (*Abdul Kadir, 2003*)

2.1.1 Informasi

McFadden, dkk (1999) mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Shannon dan Weaver, dua orang insinyur listrik, melakukan pendekatan secara matematis untuk mendefinisikan informasi (roenke, 1992). Menurut mereka, informasi adalah jumlah ketidakpastian yang dikurangi ketika sebuah pesan diterima. Artinya dengan adanya informasi, tingkat kepastian menjadi meningkat. Menurut Davis (1999), informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang. *(Abdul Kadir,2003)*

2.1.2 Sistem Informasi

Sistem informasi mencakup sejumlah komponen (manusia, computer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atas tujuan. *(Abdul Kadir,2003)*

2.2. Basis Data dan DBMS

Basis data (database) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas. Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS. DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien. DBMS dapat digunakan untuk

mengakomodasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda – beda. (*Abdul Kadir,2003*)

2.2.1 Pendekatan Tradisional

Pendekatan tradisional yang berhubungan dengan manajemen data, sumber data dikumpulkan dalam file-file yang tidak berhubungan satu dengan yang lainnya. Biasanya tiap-tiap file dirancang untuk aplikasi yang tertentu.

Pendekatan tradisional ini mempunyai beberapa kelemahan, yaitu :

1. Terjadi duplikasi data (*data redudancy*)

Karena tiap-tiap aplikasi membentuk file data tersendiri, akan dapat menimbulkan duplikasi data yang sama.

2. Tidak terjadi hubungan data (*data relatability*)

Karena tiap-tiap aplikasi menyelenggarakan file tersendiri, maka hubungan data ke file di aplikasi yang lain tidak ada. (*Lani Sidharta,1995*).

2.2.2 Pendekatan Database

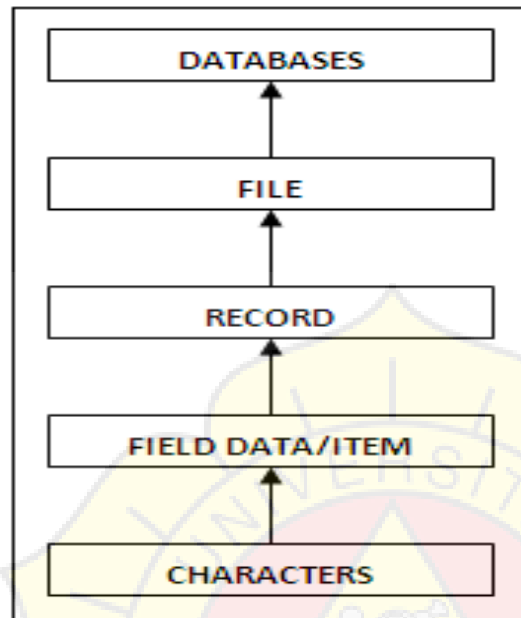
Pendekatan database mencoba memperbaiki kelemahan-kelemahan yang terjadi di pendekatan tradisional, yaitu :

1. Duplikasi data (*data redundancy*) dikurangi.

Karena database merupakan kumpulan dari semua data secara umum, maka dapat digunakan bersama-sama untuk semua aplikasi, sehingga duplikasi data dapat dikurangi.

2. Hubungan data (*data relatability*) dapat ditingkatkan.

Karena data dikumpulkan bersama-sama, maka hubungan dari data dapat ditingkatkan, yang berarti data di file tertentu dapat dihubungkan dengan data di file-file lainnya.



Gambar 2.1 Jenjang dari data. (Jogiyanto,2000)

- **Characters** : merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (*special characters*) yang membentuk suatu item data / *field*.
- **Field** : merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya.

Kumpulan dari field membentuk suatu record.

- ❖ field name: harus diberi nama untuk membedakan field yang satu dengan lainnya
- ❖ field representation: tipe field (karakter, teks, tanggal, angka, dsb), lebar field (ruang maksimum yang dapat diisi dengan karakter-karakter data).

- ❖ field value: isi dari field untuk masing-masing record.
- **Record** : Kumpulan dari field membentuk suatu record. Record menggambarkan suatu unit data individu yang tertentu. Kumpulan dari record membentuk suatu file. Misalnya file personalia, tiap-tiap record dapat mewakili data tiap-tiap karyawan.
- **File**: File terdiri dari record-record yang menggambarkan satu kesatuan data yang sejenis. Misalnya file mata pelajaran berisi data tentang semua mata pelajaran yang ada.
- **Database** : Kumpulan dari file / tabel membentuk suatu database *(Jogiyanto,2000)*

2.3. HTML (Hyper Text Markup Language)

HTML kependekan dari Hyper Text Markup Language. Dokumen HTML adalah file teks murni yang dapat di buat dengan editor teks sembarang. Dokumen ini dikenal sebagai web page. Dokumen HTML merupakan dokumen yang disajikan dalam browser web surfer. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam internet. Ada dua cara untuk membuat sebuah web page: dengan HTML editor atau dengan editor teks biasa (misalnya notepad).

Dokumen HTML disusun oleh elemen-elemen. “Elemen” merupakan istilah bagi komponen-komponen dasar pembentuk dokumen HTML. Beberapa contoh elemen adalah: head, body, table, paragraf, dan list. Elemen dapat berupa teks murni, atau bukan teks, atau keduanya.

Elemen yang di butuhkan untuk membuat suatu dokumen HTML dinyatakan dengan tag <html>, <head>, dan <body> berikut tag-tag pasangannya.

Secara umum dokumen dibagi menjadi dua section(bagian), yaitu section head dan section body. Sehingga setiap dokumen html harus mempunyai pola dasar sebagai berikut :

```
<html>
<head>
---Informasi tentang dokumen html
</head>
<body>
---informasi yang ditampilkan dalam browser web
</body>
</html>
```

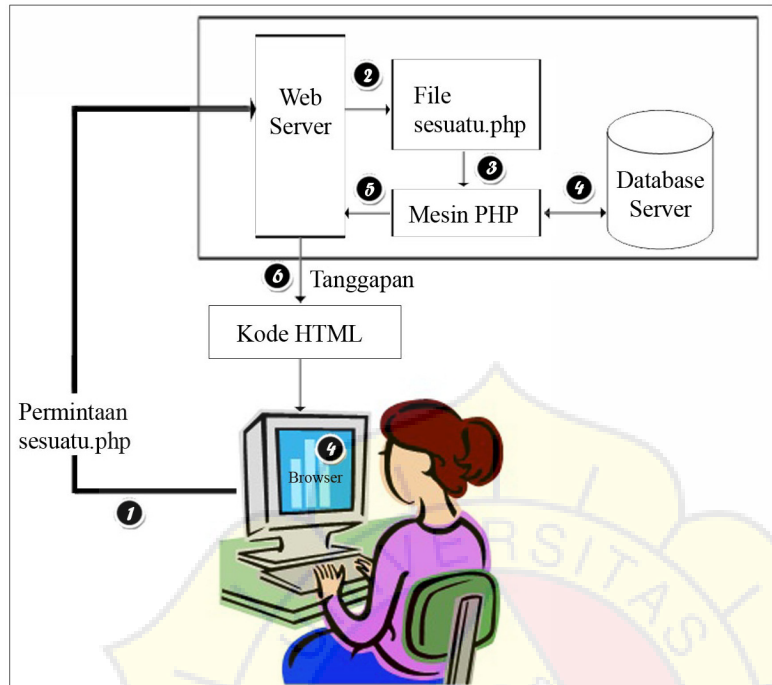
(Abdul Kadir, 2008)

2.4. Bahasa Pemrograman PHP (*Personal Home Page*)

PHP (PHP Hypertext Processor) merupakan salah satu bahasa pemrograman berbentuk skrip yang sangat populer dalam pembuatan aplikasi web. PHP tergolong sebagai *open source*, yang implisit berarti kita bisa menggunakan tanpa perlu membelinya.

PHP memungkinkan pembangunan halaman web secara dinamis. Hal seperti ini tidak bisa dilakukan hanya dengan menggunakan kode HTML. Sebagai gambaran, berkas yang hanya berisi kode HTML tidak bisa berhubungan dengan

database. Contoh pada gambar 2.2 memperlihatkan skema yang memungkinkan suatu aplikasi berinteraksi dengan database menggunakan PHP.



Gambar 2.2 Mekanisme pemanggilan aplikasi Web bertipe PHP. (Abdul Kadir, 2010)

Pada gambar diatas, setelah web server menemukan berkas yang diminta pengguna (sesuatu.php) maka berkas tersebut akan diserahkan ke mesin PHP untuk diproses lebih lanjut. Setelah semua isi berkas diproses maka hasilnya (berupa suatu kode HTML atau yang lain) diserahkan ke web server. Selanjutnya web server akan mengirimkan kode HTML atau dalam bentuk dokumen lain ke pengguna

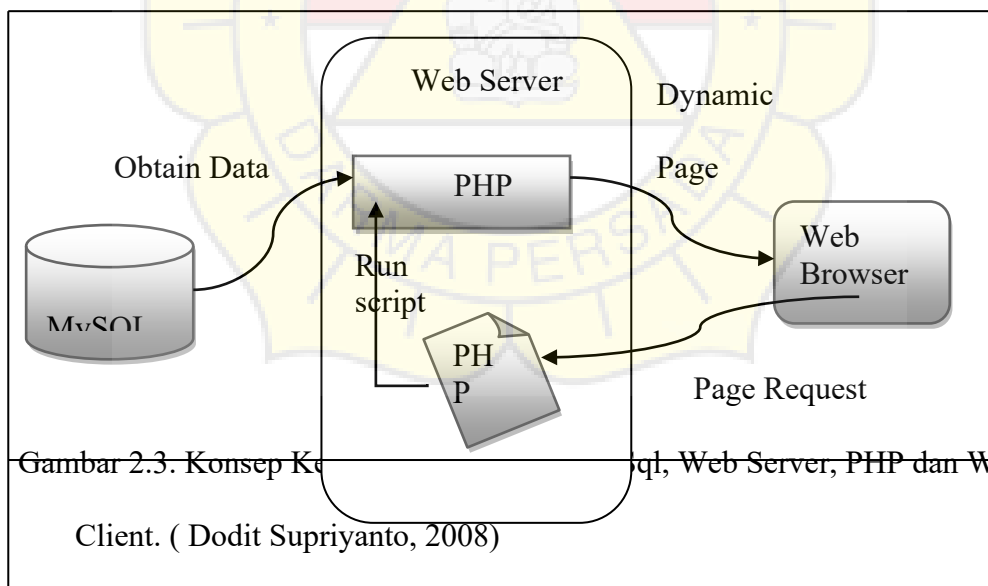
Berdasarkan penjelasan diatas, terlihat bahwa kode sumber PHP (sesuatu.php) tidak akan diketahui oleh pengguna karena mereka hanya menerima kode hasil pemrosesannya. Dengan cara seperti ini, kita tidak perlu khawatir bahwa pengguna bisa melihat password yang digunakan untuk mengakses database karena kerahasiaan kode sumber akan tetap terjaga. (Abdul Kadir, 2010)

2.5. MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *databasenya*. Selain itu ia bersifat *open source* (tidak perlu membayar untuk menggunakannya) pada berbagai platform (kecuali untuk jenis Enterprise, yang bersifat komersial).

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya, istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau beberapa kolom.

Perlu dipahami bagaimana konsep kerja antara *database* MySQL, Web Server (apache), PHP, dan Browser Client., yaitu sebagai berikut:



Gambar 2.3. Konsep Kerja MySQL, Web Server, PHP dan Web Client. (Dodit Supriyanto, 2008)

Penjelasan gambar :

1. Pengunjung *web browser* melakukan permintaan pengiriman halaman *web* (*web request*) dengan menggunakan URL standar.

2. *Web server*, dalam hal ini Apache, mengenali permintaan halaman web tersebut, kemudian menerjemahkan file yang mengandung *skript PHP*.
3. Halaman *web* yang terkirim harus mengandung *skript PHP* berisi perintah untuk menciptakan koneksi ke *database MySQL* berupa *SQL query*.
4. *Database MySQL* memberikan tanggapan atas permintaan sambungan ke *MySQL* dan *SQL query* sesuai dengan isi *skript PHP* yang dikirim.
5. *Skript PHP* menyimpan hasil *query database MySQL* menjadi satu atau lebih *variabel PHP*, dengan *echo*, maka hasil *request* dikeluarkan.
6. Selanjutnya, PHP mengkopi hasilnya dalam bentuk HTML yang telah diciptakan oleh *web server*.
7. Web server mengirim HTML yang sudah jadi ke *web browser client*. (Dodit Supriyanto, 2008).

2.6. Dreamweaver 8

Dreamweaver merupakan adalah sebuah HTML editor profesional untuk mendesain secara visual dan mengelolah situs maupun halaman web.

Dreamweaver merupakan software utama yang digunakan oleh web desainer maupun web programmer dalam mengembangkan suatu situs. Hal ini disebabkan ruang kerja, fasilitas, dan kemampuan Dreamweaver yang mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun suatu situs. Dreamweaver juga dilengkapi dengan fasilitas untuk manajemen situs

yang cukup lengkap. Untuk mengetahui versi dari dreamweaver yang digunakan, klik menu Help > About Dreamweaver.

Secara standar, program Dreamweaver 8 akan menampilkan halaman pembuka (Start Page). Ada tiga bagian pada halaman pembuka Dreamweaver 8, yaitu:

1. Open Recent Item

Menampilkan daftar file yang terakhir kali dibuka di Dreamweaver. Melalui bagian ini, pengguna juga bisa membuka file lainnya dengan cara mengklik Open.

2. Create New

Untuk membuat file baru di Dreamweaver dengan cara mengklik HTML. Untuk membuat file baru berbasis pemrograman, klik ColdFusion atau PHP, dan lainnya.

3. Create from Samples

Untuk membuat file baru menggunakan contoh/template yang sudah disediakan oleh Dreamweaver dengan cara mengklik salah satu kategori template tersebut.

2.7. CSS (*Cascading Style Sheet*)

CSS merupakan singkatan dari *Cascading Style Sheet*. Kegunaannya adalah untuk mengatur tampilan dokumen HTML, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar.

CSS dikembangkan oleh W3C, organisasi yang mengembangkan teknologi internet. Tujuannya tak lain untuk mempermudah proses penataan halaman web. Perlu diingat, CSS hanyalah berupa kumpulan skript yang tujuannya bukan untuk menggantikan HTML, melainkan pelengkap agar dokumen HTML bisa tampil lebih cantik dan dinamis. Sejak awal ditemukan CSS pada awal dekada 90an, CSS terus dikembangkan dan diserap oleh web developer. Hingga saat buku ini ditulis, versi CSS telah mencapai versi ke-3

Kode CSS bersifat lintas platform, yang berarti skript ini dapat dibaca oleh berbagai macam system operasi dan browser. Hanya saja browser seperti Internet Explorer, seringkali salah mengartikan script CSS yang menyebabkan tidak sempurnanya tampilan dokumen HTML. Script CSS perlu dioptimalkan agar tampil maksimal pada browser Internet Explorer. *(Jayan,2010)*

2.8. JQuery (*Javascript Library*)

Semenjak dirilis pertama kali pada tahun 2006 oleh John Resig, JQuery telah mencuri perhatian para developer web. Buktinya, pada tahun-tahun berikutnya JQuery telah banyak digunakan oleh website-website terkemuka di dunia seperti Google, Microsoft, Intel, Nokia, Oracle, IBM, DELL, BBC, NBC, ESPN, EA Sport, Twitter, Facebook, Amazon, Time, Youtube, Apple, Wordpress, Mozilla, Netflix, Digg, Technorati, dan masih banyak lagi, bahkan website local pun tidak mau ketinggalan, lihat saja Detik, Studio 21 (21cineplex.com), Indosiar, Vivanews, Kompas, dan lain-lain

JQuery adalah Javascript Library yaitu kumpulan kode/fungsi Javascript siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode

Javascript. Secara standar, apabila kita membuat kode Javascript, maka diperlukan kode yang cukup panjang, bahkan terkadang sangat sulit dipahami. Disinilah peran JQuery sebagai Javascript Library, dimana kita bisa langsung memanggil fungsi yang terdapat didalam Library tersebut. Kesimpulannya, JQuery menyederhanakan kode Javascript. Hal ini sesuai dengan slogannya *write less, do more* cukup tulis sedikit, tapi bisa melakukan banyak hal. (Lukmanul Hakim, 2010)

2.9. UML (*Unified Modelling Language*)

UML itu singkatan dari *Unified Modelling Language*. Sesuai dengan kata terakhir dari kepanjangannya, UML merupakan salah satu bentuk *language* atau bahasa. Menurut pencetusnya, UML didefinisikan sebagai bahasa *visual* untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem.

Karena tergolong bahasa *visual*, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem yang sedang dimodelkan. Memahami UML itu sebagai bahasa *visual* itu penting, karena penekanan tersebut membedakannya dengan bahasa pemrograman yang lebih dekat ke mesin. Bahasa visual lebih dekat ke mental model pikiran kita, sehingga pemodelan menggunakan bahasa visual bisa lebih mudah dan lebih cepat dipahami dibandingkan apabila dituliskan dalam sebuah bahasa pemrograman.

UML adalah salah satu bentuk notasi atau bahasa yang sama yang digunakan oleh professional dibidang *software* untuk menggambarkan atau memodelkan sebuah sistem *software*. Sebelumnya ada banyak notasi atau bahasa lain untuk

mencapai keperluan yang sama misalnya DFD (*Data Flow Diagram*). Tetapi sejak matang dan populernya teknologi pemrograman, perancangan, dan analisis berorientasi objek, UML telah menjadi *de facto standard language*.

Ada tiga cara dalam memakai UML dalam melakukan pemodelan sistem:

1. UML sebagai sketsa

UML digambarkan dalam sketsa coretan-coretan dalam kertas atau *whitboard* secara tidak formal. Biasanya digunakan dalam sesi diskusi tim untuk membahas aspek tertentu dalam tahap analisis dan perancangan.

2. UML sebagai *blueprint* sistem

Seperti diagram kelistrikan adalah *blueprint* dari komponen atau produk yang akan dihasilkan, UML juga bisa menggambarkan *blueprint* yang identik untuk sebuah sistem *software*.

3. UML sebagai bahasa pemrograman

UML berfungsi sebagai bahasa pemrograman mencoba melakukan semuanya dengan UML sampai kepada produk jadinya. Analisis dan perancangan dilakukan dengan diagram-diagram yang ada dalam UML, sementara sebuah *tool* atau generator bisa menghasilkan produk akhir dari diagram-diagram ini.

2.9.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan

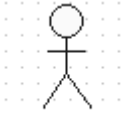

“bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

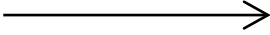
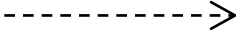
Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

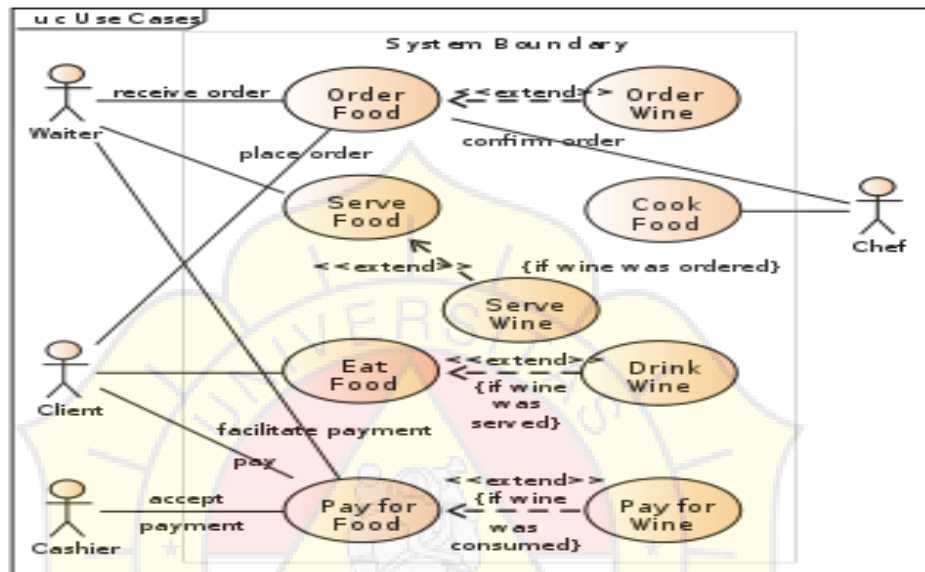
Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Berikut adalah elemen dalam *use case* :

Tabel 2.1 Simbol *Class Diagram*

Penjelasan	Notasi UML
Aktor : Mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan <i>use case</i> .	 Pelanggan
<i>Use Case</i> : Abstraksi dari interaksi antara sistem dan <i>actor</i>	 Mengisi soal

<i>Association</i> : adalah abstraksi dari penghubung antara <i>actor</i> dan <i>use case</i>	
Generalisasi : menunjukkan spesialisasi actor untuk dapat berpartisipasi dalam <i>use case</i>	



Gambar 2.4 Use Case Diagram

2.9.2. Activity Diagram



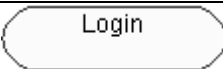

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem)

secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Simbol-simbol yang terdapat dalam *Activity diagram*, sebagai berikut :

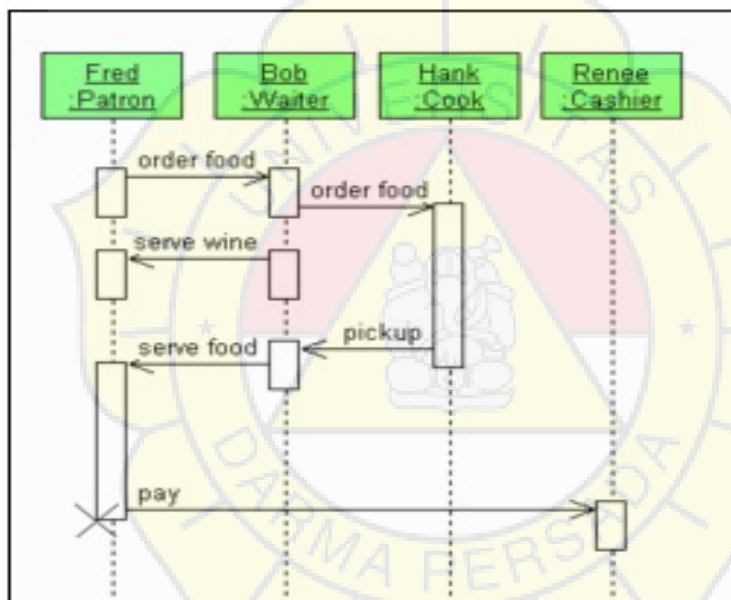
Tabel 2.2 Simbol *Activity Diagram*

Keterangan	Simbol
Titik Awal atau permulaan.	
Titik Akhir atau akhir dari aktivitas.	
Aktiviti, atau aktivitas yang dilakukan oleh aktor.	
Decision, atau pilihan untuk mengambil keputusan.	

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*.

Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.



Gambar 2.6 *Sequence Diagram*. (Munawar, 2005)