

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian dan Pengabdian kepada Masyarakat oleh Perguruan Tinggi.

Ditjen Dikti dalam Panduan Pelaksanaan Penelitian dan Pengabdian kepada Masyarakat Oleh Perguruan Tinggi mengemukakan kegiatan penelitian dan pengabdian kepada masyarakat di Indonesia pada umumnya dikelola oleh setiap Departemen Teknis di bawah koordinasi Badan Penelitian dan Pengembangan (Badan Litbang) masing-masing. Namun, Departemen Pendidikan Nasional yang secara langsung terkait dengan Penyiapan Sumberdaya Manusia dari tingkat pendidikan dasar sampai pendidikan tinggi, Badan Litbangnya lebih menangani penelitian yang berhubungan dengan Pendidikan Dasar, Menengah dan Luar Sekolah. Direktorat Pembinaan Penelitian dan Pengabdian pada Masyarakat (Ditbinlitabmas) lebih difungsikan dalam aspek pembinaan penelitian dan kesinambungannya dengan pengabdian kepada masyarakat. (UNHAS, 2002)

Direktorat Pembinaan Penelitian dan Pengabdian pada Masyarakat dalam operasionalisasinya, bekerja secara koordinatif-integratif dengan direktorat-direktorat lain di bawah Direktorat Jenderal Pendidikan Tinggi (Ditjen-Dikti) untuk menumbuhkan sinergi yang positif baik dalam penyempurnaan ataupun pengembangan penyiapan sumberdaya manusia pada tingkat pendidikan tinggi.

Kehidupan dan perkembangan akademik di perguruan tinggi tidak terlepas dari perkembangan ilmu pengetahuan, teknologi dan seni (Ipteks) serta tuntutan masyarakat seiring dengan meningkatnya kualitas kehidupannya. Untuk mengantisipasi dan menyerasikannya, Ditbinlitabmas di dalam menyusun serta melaksanakan program kegiatannya senantiasa mengacu pada Undang-Undang

Pendidikan Nasional, yang mengamanatkan bahwa pendidikan tinggi harus diarahkan pada penciptaan sumberdaya manusia yang mempunyai kemampuan akademis, profesional dan kepemimpinan, serta tanggap terhadap kebutuhan ipteks. Dengan demikian program pembinaan Ditbinlitabmas mengarah pencapaian hasil optimal dari Tridarma Perguruan Tinggi yang mencakup Pendidikan, Penelitian dan Pengabdian kepada Masyarakat, utamanya yang bermanfaat bagi kemanusiaan dan selaras serasi dengan kebutuhan pembangunan. Di samping itu, diupayakan pula berbagai kegiatan peningkatan kualitas pendidikan dan pengajaran di lingkungan pendidikan tinggi.

Berpijak pada pola kebijakan di atas, dan dengan disepakatinya pola penyediaan dana penelitian secara berkesinambungan, maka sejak tahun 1992 diterapkan program penelitian berjangka panjang di lingkungan Ditbinlitabmas. Oleh karena itu program penelitian yang dilakukan di perguruan tinggi dituntut untuk menghasilkan produk yang benar-benar berkualitas dan bermanfaat. Sedangkan program pengabdian kepada masyarakat lebih diarahkan kepada pemanfaatan dan penerapan hasil penelitian maupun hasil pendidikan di perguruan tinggi bagi kesejahteraan dan kemajuan masyarakat.

Jurnal ilmiah perlu dikembangkan karena dua hal: motivasi dosen untuk menulis dengan memadai di jurnal ilmiah masih rendah dan wahana komunikasi yang pantas di kalangan masyarakat ilmiah masih terbatas. Program pembinaan dimulai pada tahun 1992 dengan mengkaji keberadaan jurnal ilmiah. Hasilnya menyatakan bahwa banyak jurnal bersifat bunga rampai, pengelolaan jurnal kurang profesional, dan mutu artikel yang dimuatnya masih belum memadai. Pembinaan yang diupayakan meliputi pelatihan penulisan artikel ilmiah, lokakarya pengelolaan jurnal, program bantuan penerbitan jurnal ilmiah, penjaminan mutu melalui akreditasi jurnal, program bantuan jurnal himpunan profesi, dan meningkatkan nilai kredit (kum) bagi penulis artikel. Masa akreditasi suatu jurnal ialah 3 tahun.

2.2. Lembaga Penelitian dan Pengembangan Unissula

Dalam rangka melaksanakan Tri Dharma Perguruan Tinggi, Lembaga Penelitian dan Pengembangan (LPP) Universitas Islam Sultan Agung (Unissula) Semarang telah melaksanakan kerjasama dengan Dinas / Instansi baik di tingkat Departemen, Pemerintah Propinsi, Pemerintah Kabupaten, dan Pemerintah Kota.

1. Bidang kajian

Bidang kajian penelitian yang ditangani oleh LPP Unissula, antara lain :

- Pusat Penelitian Lingkungan
- Pusat Penelitian Kesehatan dan Biomedis
- Pusat Penelitian Sosial, Ekonomi dan Agama
- Pusat Penelitian Teknologi
- Pusat Studi Pengembangan Ilmu dan Sistem Pengajaran
- Pusat Studi Wanita (PSW)
- Pusat Penelitian Sumber Daya Alam dan Wilayah

2. Sumber pendanaan

Berdasarkan segi pendanaan penelitian dibagi menjadi dua sumber:

- Penelitian dengan pendanaan Unissula
- Penelitian dengan pendanaan dari luar Unissula

2.3. HTML (HyperText Markup Language)

HyperText Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi di dalam sebuah *browser* internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C).

HTML berupa kode-kode *tag* yang menginstruksikan *browser* untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan *browser web* seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka *email* ataupun dari PDA (*Personal Digital Assistant*) dan program lain yang memiliki kemampuan *browser*. (wikipedia, 2011)

HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: **TAMPIL TEBAL**, maka penulisannya dilakukan dengan cara: `TAMPIL TEBAL`. Tanda `` digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh teks yang ingin ditebalkan, dan diakhiri dengan tanda `` untuk menonaktifkan cetak tebal tersebut.

Secara garis besar, terdapat 4 jenis elemen dari HTML:

- a. *Structural*, tanda yang menentukan level atau tingkatan dari sebuah teks (contoh, `<h1>Golf</h1>` akan memerintahkan browser untuk menampilkan "Golf" sebagai teks tebal besar yang menunjukkan sebagai Heading 1).
- b. *Presentational*, tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut (contoh, `boldface` akan menampilkan bold. Tanda presentational saat ini sudah mulai digantikan oleh CSS (*Cascading Style Sheets*) dan tidak direkomendasikan untuk mengatur tampilan teks).
- c. *Hypertext*, tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain (contoh, `Wikipedia` akan menampilkan Wikipedia sebagai sebuah hyperlink ke URL tertentu).
- d. Elemen *widget* yang membuat objek-objek lain seperti tombol (`<button>`), list (``), dan garis horizontal (`<hr>`).

Selain markup *presentational* , markup yang lain tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan *tag* HTML untuk menentukan tampilan telah dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan *Cascading Style Sheets* (Sumber: <http://id.wikipedia.org>).

2.4. Pemrograman PHP

PHP: *Hypertext Preprocessor* adalah “skrip yang berjalan dalam server side yang ditambahkan dalam HTML”. PHP itu sendiri merupakan singkatan dari *personal home page tools*. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman HTML tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* ini membuat pengerjaan skrip tersebut dikerjakan di *server* sedangkan yang dikirimkan kepada *browser* adalah hasil proses dari skrip tersebut yang sudah berbentuk HTML. (Peranginangin, 2006)

Keunggulan dari sifatnya yang *server side* adalah:

- a. Tidak diperlukan adanya kompatibilitas *browser* atau harus menggunakan *browser* tertentu, karena *server* yang akan mengerjakan skrip tersebut. Hasil yang dikirimkan kembali ke *browser* biasanya dalam bentuk teks ataupun gambar sehingga dapat dikenali oleh *browser* apapun.
- b. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh *server*, contoh: hubungan ke dalam *database*.
- c. Skrip tidak dapat dilihat sehingga keamanan lebih terjamin.
- d. PHP dibuat pada tahun 1994 oleh Rasmus Lerdorf. Tetapi kemudian dikembangkan oleh orang lain dan setelah melalui tiga kali karya penulisan akhirnya PHP menjadi bahasa pemrograman *web* seperti sekarang ini. *Domain* pengguna PHP jumlahnya saat ini dapat dilihat pada <http://www.php.net/usage.php>.

2.5. Keamanan

Keamanan bagi sebuah program menjadi sangat penting, di samping sistem operasi, misalnya, fasilitas fungsi lengkap dari bahasa pemrograman yang memberikan opsi ketika terjadi error. PHP ialah pemrograman berbasis web yang akan memberikan keamanan canggih jika dipadukan dengan webserver yang memberikan tingkat keamanan yang cukup tinggi, dalam hal ini Apache. Apache telah digunakan sebagai webserver karena terbukti benar-benar teruji dan dapat diandalkan. Hampir 70% dari jumlah server di dunia menggunakan Apache sebagai webserver (Syafii, 2005)

2.6. Program Basisdata MySQL

Menurut Bunafit Nugroho (Nugroho, 2008) mengemukakan bahwa “MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan standar SQL (*Structured Query Language*)”. Dengan menggunakan *MySQL server* maka data dapat diakses oleh banyak pemakai secara bersamaan sekaligus dapat membatasi akses para pemakai berdasarkan *privillage* (hak akses) yang diberikan. MySQL menggunakan bahasa SQL yaitu bahasa standar pemrograman *database*. MySQL dipublikasikan sejak tahun 1996, tetapi sebenarnya dikembangkan sejak tahun 1979. MySQL telah memenangkan penghargaan *Linux Journal Reader’s Choice Award* selama tiga tahun. MySQL sekarang tersedia di bawah ijin *open source*, tetapi juga ada ijin untuk penggunaan secara komersial.

Keunggulan dari MySQL adalah bersifat *open source*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli atau membayar lisensi kepada pembuatnya.

2.7. Pengantar SQL

Structured *Query Language* (SQL) adalah bahasa komputer standar ANSI (*American National Standard Institute*) untuk mengakses dan memanipulasi

sistem basisdata. Pernyataan SQL digunakan untuk mengambil dan meng-*update* data dalam basisdata. (Wikipedia, 2011)

Secara umum, SQL terdiri atas dua bahasa, yaitu Data Definition Language (DDL) dan Data Manipulation Language (DML).

1. *Data Definition Language (DDL)*

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya *tabel*, *view*, *user*, dan sebagainya. Secara umum, DDL yang digunakan adalah CREATE untuk membuat objek baru, USE untuk menggunakan objek, ALTER untuk mengubah objek yang sudah ada, dan DROP untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

CREATE digunakan untuk membuat basis data maupun objek-objek basis data. SQL yang umum digunakan adalah:

```
CREATE DATABASE nama_basis_data
```

CREATE DATABASE membuat sebuah basis data baru.

```
CREATE TABLE nama_tabel
```

CREATE TABLE membuat tabel baru pada basis data yang sedang aktif. Secara umum, perintah ini memiliki bentuk:

```
CREATE TABLE [nama_tabel]
(
nama_field1 tipe_data [constraints][,
```

atau

```
CREATE TABLE [nama_tabel]
(
nama_field1 tipe_data [,
nama_field2 tipe_data,
```

Keterangan :

- *nama_field* adalah nama kolom (*field*) yang akan dibuat. Beberapa sistem manajemen basis data mengizinkan penggunaan spasi dan karakter nonhuruf pada nama kolom.
- *tipe_data* tergantung implementasi sistem manajemen basis data. Misalnya, pada MySQL, tipe data dapat berupa VARCHAR, TEXT, BLOB, ENUM, dan sebagainya.
- *constraints* adalah batasan-batasan yang diberikan untuk tiap kolom. Ini juga tergantung implementasi sistem manajemen basis data, misalnya NOT NULL, UNIQUE, dan sebagainya. Ini dapat digunakan untuk mendefinisikan kunci primer (*primary key*) dan kunci asing (*foreign key*).

Satu tabel boleh tidak memiliki kunci primer sama sekali, namun sangat disarankan mendefinisikan paling tidak satu kolom sebagai kunci primer.

Contoh:

```
CREATE TABLE user
(
username VARCHAR(30) CONSTRAINT PRIMARY KEY,
```

akan membuat tabel user seperti berikut

<u>username</u>	passwd	tanggal_lahir
-----------------	--------	---------------

2. *Data Manipulation Language (DML)*

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

- SELECT untuk menampilkan data
- INSERT untuk menambahkan data baru
- UPDATE untuk mengubah data yang sudah ada
- DELETE untuk menghapus data

SELECT adalah perintah yang paling sering digunakan pada SQL, sehingga kadang-kadang istilah query dirujuk pada perintah SELECT. SELECT digunakan untuk menampilkan data dari satu atau lebih tabel, biasanya dalam sebuah basis data yang sama. (Wikipedia, 2011)

```
SELECT [nama_tabel|alias.]nama_field1 [AS alias1] [, nama_field2, ...]
FROM nama_tabel1 [AS alias1] [INNER|LEFT|RIGHT JOIN tabel2 ON
kondisi_penghubung]
[, nama_tabel3 [AS alias3], ...]
```

keterangan:

- *kondisi* adalah syarat yang harus dipenuhi suatu data agar ditampilkan.

- *kondisi_aggregat* adalah syarat khusus untuk fungsi agregat.

INSERT digunakan untuk menyimpan data dalam tabel dipergunakan sintaks:

```
INSERT INTO [nama_tabel] ([daftar field]) VALUES ([daftar nilai])
```

Contoh :

```
INSERT INTO TEST (nama, alamat, password) VALUES ('test', 'alamat', 'pass');
```

UPDATE digunakan untuk mengubah data menggunakan sintaks:

```
UPDATE [nama_tabel] SET [nama_kolom]=[nilai] WHERE [kondisi]
```

Contoh :

```
UPDATE Msuser SET password="123456" WHERE username="abc"
```

DELETE digunakan untuk menghapus data dipergunakan sintaks:

```
DELETE FROM [nama_tabel] [kondisi]
```

contoh:

```
DELETE FROM test WHERE nama='test';
```

2.8. Unified Modeling Language (UML)

Unified Modeling Language adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO).

Menurut Quatrani (1998:1) pemodelan visual adalah suatu cara berpikir tentang persoalan menggunakan model-model yang diorganisasikan seputar dunia nyata. Model berguna untuk memahami persoalan, mengkomunikasikan dengan orang-orang yang terlibat dalam proyek (customer, ahli dibidangnya, analis, desainer, dan lain-lain), memodelkan perusahaan, menyiapkan dokumentasi, merancang program, dan merancang basis data.

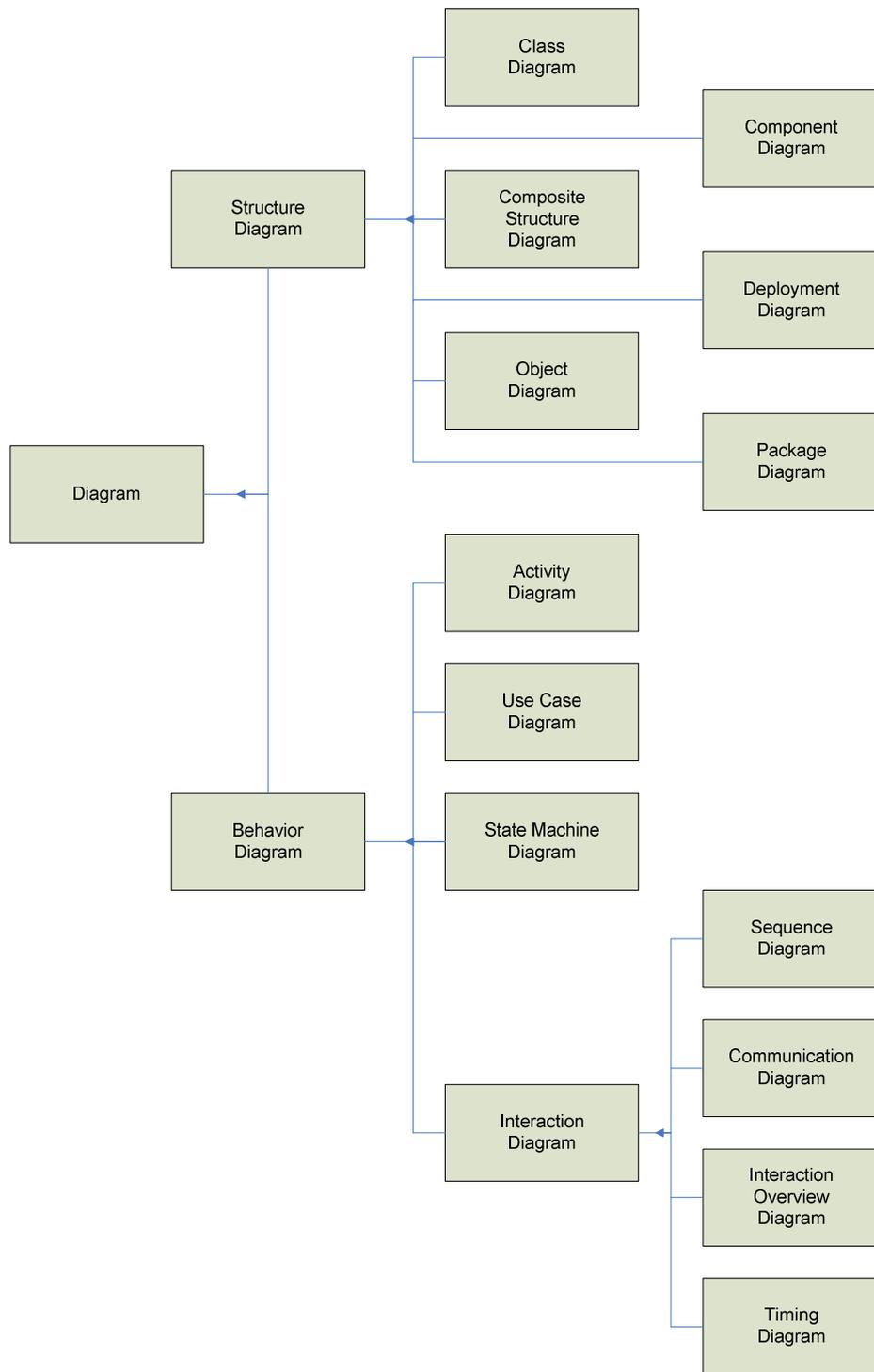
Sedangkan menurut Boggs (2002:6) pemodelan visual didefinisikan sebagai proses pemodelan sistem informasi menggunakan pengaturan standar elemen grafik. Tujuan utama pemodelan visual adalah untuk memungkinkan adanya komunikasi antara pengguna, pengembang, penganalisis, tester, manager, dan siapapun yang terlibat dalam proyek, menunjukkan interaksi antara pengguna dengan sistem, obyek-obyek dalam sistem, dan antar sistem itu sendiri. (Hakim, 2010)

Pengembangan UML hingga saat ini mencapai versi yang kedua. UML 1 dengan UML 2 memiliki penyempurnaan dari segi pengurangan bahasa yang luas dan kompleks dan terbukti UML 2 lebih efektif dalam prakteknya. (Sholih, 2006)

Tabel 2. 1 Jenis diagram resmi UML

Diagram	Kegunaan	Turunan
Aktivitas	<i>Behavior procedural</i> dan paralel.	Di UML 1
Kelas	<i>Class</i> , fitur, dan hubungan-	Di UML 1

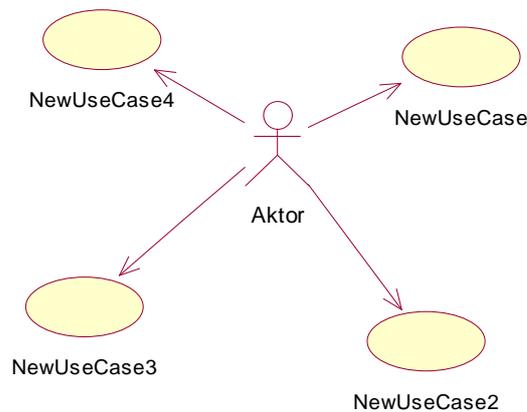
	hubungan.	
Komponen	Struktur dan koneksi komponen.	Di UML 1
<i>Composite Structure</i>	Dekomposisi <i>runtime</i> sebuah <i>class</i> .	Baru di UML 2
<i>Deployment</i>	Pemindahan artifak ke node.	Di UML 1
<i>Interaction overview</i>	Campuran diagram sekuensial dan aktivitas.	Baru di UML 2
<i>Object</i>	Contoh konfigurasi dari contoh-contoh.	Tidak resmi di UML 1
Sekuensial	Interaksi antar objek; penekanan pada sekuensial.	Di UML 1
<i>State machine</i>	Menjelaskan objek secara lebih dinamis.	Di UML 1
<i>Timing</i>	Interaksi antar objek; penekanan pada waktu.	Baru di UML 2
<i>Use case</i>	Bagaimana pengguna berinteraksi dengan sebuah sistem.	Di UML 1



Gambar 2.1 *Klasifikasi jenis diagram UML*

2.9. Diagram *use case*

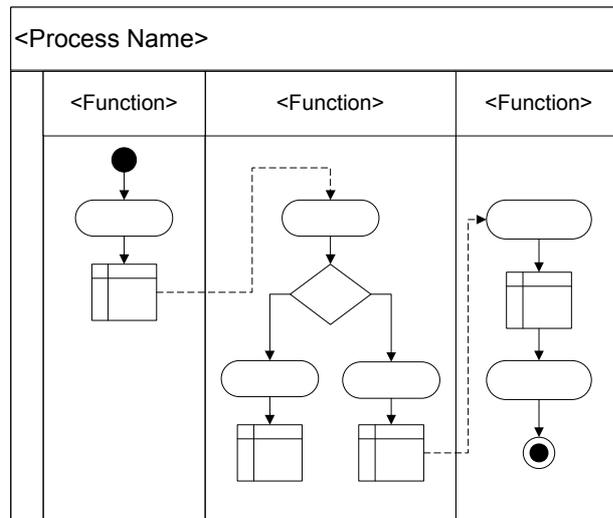
Diagram *use case* menyajikan interaksi antara *use case* dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.



Gambar 2. 2 Skema diagram *use case*

2.10. Diagram aktivitas

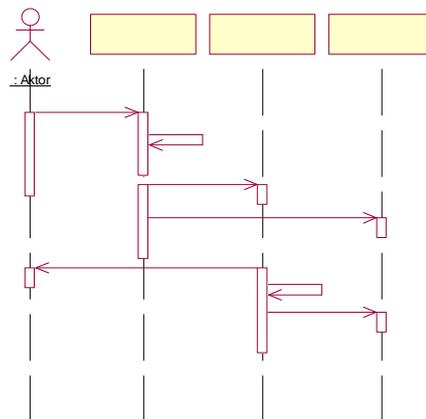
Diagram aktivitas menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business workflow*). Dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of events*) dalam *use case*.



Gambar 2.3 Skema diagram aktivitas

2.11. Diagram sekuensial

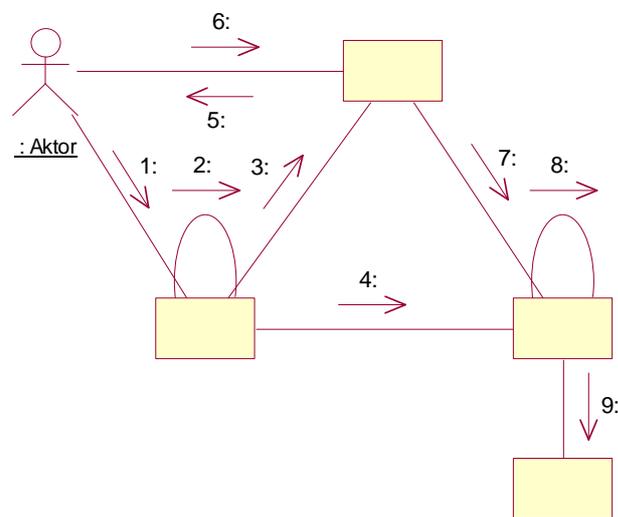
Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Keseluruhan interaksi berdasarkan urutan waktu.



Gambar 2.4 Skema diagram sekuensial

2.12. Diagram kolaborasi

Diagram kolaborasi menunjukkan informasi yang sama persis dengan diagram sekuensial, tetapi dalam bentuk dan tujuan yang berbeda. Pada diagram sekuensial, keseluruhan interaksi berdasarkan urutan waktu, tetapi pada diagram kolaborasi, interaksi antar objek atau aktor ditunjukkan dengan arah panah tanpa keterangan waktu.

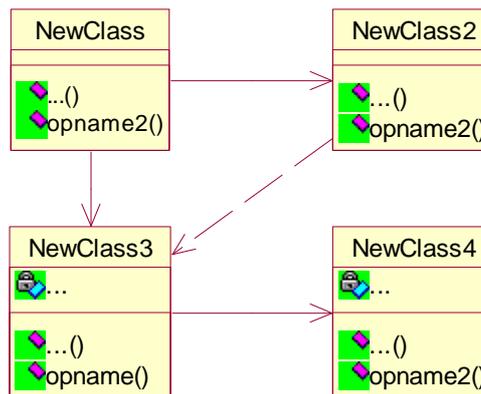


Gambar 2.5 Skema diagram kolaborasi

Pada dasarnya diagram ini sama dengan diagram sekuensial, tetapi orang menggunakan diagram ini untuk alasan yang berbeda. Tenaga ahli jaminan kualitas dan arsitek sistem menggunakan diagram ini untuk melihat proses distribusi antar objek. Diagram ini berbentuk seperti bintang, dengan beberapa objek yang berkomunikasi dengan sebuah objek pusat. Arsitek sistem menggunakan diagram ini untuk menyimpulkan bahwa sistem yang dibangun terlalu tergantung pada objek pusat, dan merancang ulang objek-objek untuk mendistribusikan proses secara merata. Interaksi demikian akan sulit dilihat jika menggunakan diagram sekuensial saja.

2.13. Diagram kelas

Diagram kelas menunjukkan interaksi antar kelas dalam sistem. Sebuah kelas pada diagram kelas dibuat untuk setiap tipe objek pada diagram sekuensial atau diagram kolaborasi.



Gambar 2. 6 Skema diagram kelas

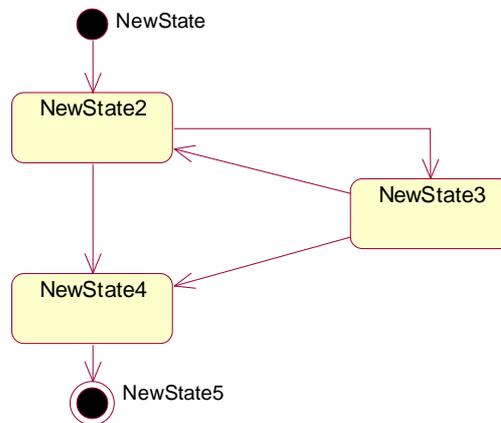
Sebuah kelas dibuat dalam bentuk bujur sangkar yang terbagi dalam tiga bagian. Bagian pertama menunjukkan nama kelas. Bagian kedua menunjukkan anggota kelas yang memuat informasi atau atribut. Bagian ketiga menunjukkan operasi-operasi dari sebuah kelas, dimana operasi dari sebuah kelas adalah tingkah laku yang disediakan oleh kelas. Garis yang menghubungkan antar kelas menunjukkan hubungan komunikasi antar kelas.

Para *programmer* menggunakan diagram ini untuk mengembangkan kelas. Para Analis menggunakan diagram ini untuk menunjukkan detail sistem, sedangkan arsitek sistem menggunakan diagram ini untuk melihat rancangan sistem.

2.14. Diagram statechart

Diagram *statechart* menyediakan sebuah cara untuk memodelkan bermacam-macam keadaan yang mungkin dialami oleh sebuah objek. Jika dalam

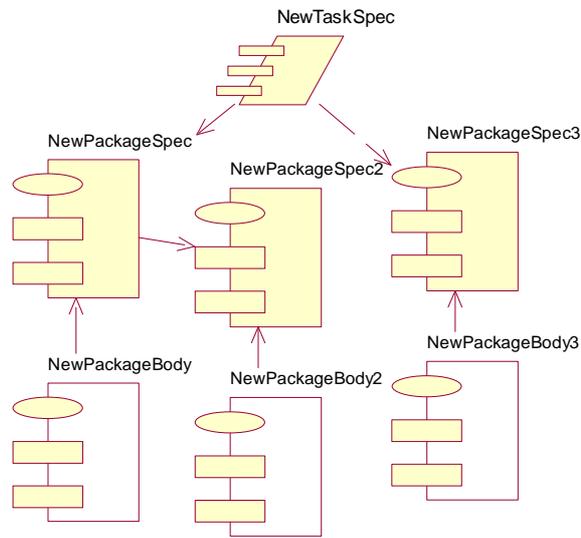
diagram kelas menunjukkan gambaran statis kelas-kelas dan relasinya, diagram *statechart* digunakan untuk memodelkan tingkah laku dinamik sistem.



Gambar 2. 7 Skema diagram statechart

2.15. Diagram komponen

Diagram komponen menunjukkan model secara fisik komponen perangkat lunak pada sistem dan hubungannya antar mereka. Ada dua tipe komponen dalam diagram yaitu komponen *executable* dan kode pustaka (*libraries code*). Masing-masing kelas dalam model akan dipetakan ke sebuah komponen kode pustaka. Setelah komponen dibuat, mereka ditambahkan dalam diagram komponen dengan memberikan relasi antara komponen-komponen. Relasi yang terjadi antar komponen hanya satu tipe relasi yaitu dependensi yang menunjukkan ketergantungan *compile-time* dan *run-time* antara komponen-komponen tersebut.

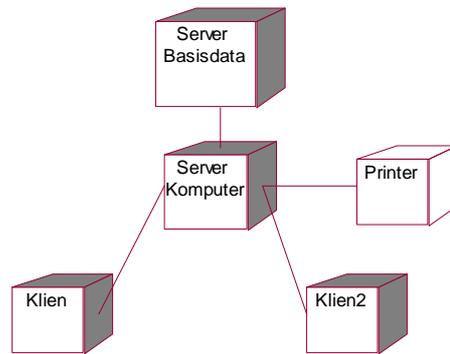


Gambar 2. 8 Skema diagram komponen

Komponen terhubung oleh garis putus-putus yang menampilkan hubungan dependensi antar komponen. Diagram komponen digunakan oleh siapapun yang bertanggung jawab untuk melakukan kompilasi sistem. Diagram ini juga menunjukkan komponen apa yang dibutuhkan saat proses kompilasi dan menampilkan komponen *run-time* apa saja yang dibuat sebagai hasil proses kompilasi. Komponen diagram memperlihatkan pemetaan dari kelas-kelas ke komponen-komponen sebagai implementasi kelas.

2.16. Diagram deployment

Diagram *deployment* menampilkan rancangan fisik jaringan dimana berbagai komponen akan terdapat disana.



Gambar 2. 9 Skema diagram deployment

Diagram *deployment* digunakan oleh manajer proyek, arsitek sistem, dan karyawan distribusi untuk memahami rancangan fisik sistem dan di mana saja terdapat subsistem yang akan dibuat. Diagram ini membantu manajer proyek mengkomunikasikan tentang apa yang sistem inginkan terhadap pemakai, juga membantu bagian pengembangan untuk merencanakan distribusi yang akan ditawarkan.