

PEMODELAN PERANGKAT LUNAK

(C3) KELAS XI

Abdul Munif

PT KUANTUM BUKU SEJAHTERA

PEMODELAN PERANGKAT LUNAK

SMK/MAK Kelas XI

© 2020

Hak cipta yang dilindungi Undang-Undang ada pada Penulis.

Hak penerbitan ada pada **PT Kuantum Buku Sejahtera**.

Penulis	: Abdul Munif
Editor	: Fourdina Ratnasari
Desainer Kover	: Achmad Faisal
Desainer Isi	: Putri Ari Kristanti
Tahun terbit	: 2020
ISBN	: 978-623-7216-99-5

Diterbitkan oleh

PT Kuantum Buku Sejahtera

Anggota IKAPI No. 212/JTI/2019

Jalan Pondok Blimbing Indah Selatan X N6 No. 5 Malang - Jawa Timur

Telp. (0341) 438 2294, Hotline 0822 9951 2221;

Situs web: www.quantumbook.id

*Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronis maupun mekanis, termasuk memfotokopi, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari **PT Kuantum Buku Sejahtera**.*

Daftar Isi

Prakata	v
Bab 1 Konsep Pemodelan Sistem Perangkat Lunak	1
A. Memahami Perangkat Lunak dan Kategorinya	2
B. Memahami Sistem Informasi	3
C. Memahami Teknik Pengembangan Sistem Perangkat Lunak.....	3
D. Memahami Pemodelan dalam Rekayasa Perangkat Lunak	4
E. Memahami Karakteristik Pemodelan Perangkat Lunak	5
F. Memahami Pemodelan Sistem Berorientasi Objek	6
G. Memahami Unified Modeling Language (UML)	7
Uji Kompetensi.....	10
Bab 2 Alur Kerja Sistem Berorientasi Objek	15
A. Memahami Ragam Jenis Model Proses Pengembangan Sistem	16
B. Memahami Alur Kerja Pengembangan Sistem Berorientasi Objek.....	20
Uji Kompetensi.....	25
Bab 3 Kebutuhan Sistem Berorientasi Objek	29
A. Memahami Ragam Jenis kebutuhan.....	30
B. Memahami Rekayasa Kebutuhan (Requirements Engineering)	32
C. Memahami Tahapan Requirements Engineering	33
D. Memahami Teknik-Teknik Analisis Kebutuhan	34
E. Mengamati Kebutuhan Fungsional.....	36
F. Menelaah Dokumen Kebutuhan	37
G. Pemodelan Bisnis (Business Modeling)	38
H. Model Diagram REAL (Resource, Events, Agent, and Locations).....	40
Uji Kompetensi.....	42
Bab 4 Pembuatan Model Berorientasi Objek	47
A. Memahami Analisis dan Desain Sistem Berorientasi Objek (OOAD)	48
B. Memahami Object Oriented Analysis (OOA).....	48
C. Memahami Object Oriented Design (OOD).....	51
D. Memahami Ragam Model Diagram UML	52
E. Memahami Proses-Proses Model UML	53
F. Memahami Tahapan Pembuatan Model UML	54
G. Membuat Use Case Diagram.....	54
H. Membuat Aktivitas Diagram	59
Uji Kompetensi.....	66
Bab 5 Relasi Kelas Sistem Berorientasi Objek	71
A. Memahami Struktur Class, Object, dan Package	72
B. Memahami Inheritance.....	75
C. Memahami Polymorphism dan Dynamic Binding.....	76

D.	Memahami Relasi Antarclass.....	77
E.	Memahami Abstract Class.....	81
F.	Memahami Port	81
G.	Memahami Class Responsibility Colaborator (CRC)	82
H.	Mengamati Ragam Contoh Class Diagram	85
	Uji Kompetensi.....	87
Bab 6	Interaksi Objek Sistem Berorientasi Objek	91
A.	Memahami Objek Diagram	92
B.	Memahami Sequence Diagram.....	94
C.	Memahami State Chart Diagram	98
	Uji Kompetensi.....	101
Bab 7	Komponen Sistem Berorientasi Objek.....	105
A.	Memahami Component Diagram	106
B.	Memahami Arsitektur Aplikasi	111
C.	Perancangan Arsitektur Aplikasi Sistem ATM	112
D.	Perancangan User Interface Sistem ATM	113
E.	Implementasi Desain Basis Data Sistem ATM	115
	Uji Kompetensi.....	117
Bab 8	Dokumen Laporan Pengembangan Sistem Berorientasi Objek.....	121
A.	Menjelaskan Dokumentasi Pengembangan Sistem.....	122
B.	Memahami Dokumentasi Perancangan Sistem	122
C.	Menjelaskan Dokumentasi Sistem Kontrol Real Time.....	124
D.	Menjelaskan Kerangka Laporan Pengembangan Sistem	127
	Uji Kompetensi.....	130
Bab 9	Sistem Dokumen Metadata	135
A.	Memahami Dokumen Metadata	136
B.	Memahami Implementasi Dokumen Metadata.....	140
	Uji Kompetensi.....	147
Bab 10	Pembuatan Sistem Dokumen Metadata	151
A.	Memahami XML.....	152
B.	Tujuan Pembuatan dan Penggunaan XML	152
C.	Struktur Dokumen XML	153
D.	Membangun Dokumen Metadata dengan XML.....	155
	Uji Kompetensi.....	164
	Glosarium	169
	Daftar Pustaka	172
	Biodata Penulis	176
	Biodata Konsultan	177
	Tim Kreatif.....	178

Prakata

Buku teks pelajaran Pemodelan Perangkat Lunak ini disusun berdasarkan Kurikulum 2013 edisi revisi 2018. Buku ini diharapkan menjadi buku pedoman belajar bagi peserta didik di Sekolah Menengah Kejuruan (SMK) untuk Kompetensi Keahlian Rekayasa Perangkat Lunak. Metode pembelajaran pengetahuan menggunakan observasi dan diskusi yang mengembangkan kemampuan *collaborative*, yaitu kreativitas, intelektual, komunikasi, sikap, karakter, dan dinamika individu serta kelompok. Pembelajaran keterampilan menggunakan metode praktik dapat digunakan untuk menyelesaikan permasalahan dunia nyata. Melalui *discovery learning*, menggali pengetahuan dan keterampilan peserta didik melalui penyelidikan dan pengamatan, *project base learning* menuntut kompetensi *problems solving* dan pembuatan produk kreatif berdasarkan permasalahan nyata.


Buku ini tersusun dalam sepuluh bagian (bab), meliputi Konsep pemodelan sistem perangkat lunak, alur kerja pengembangan sistem berorientasi objek, rekayasa kebutuhan sistem, pembuatan model sistem, relasi kelas dalam sistem, interaksi objek dalam sistem, struktur komponen sistem, dokumen laporan pengembangan sistem, sistem dokumen metadata, dan rancang bangun sistem metadata. Setiap bab terdiri dari uraian materi yang menjelaskan pemahaman konsep dan urutan prosedur praktik. Pelatihan atau tugas menjelaskan tahapan langkah praktik uji kompetensi berisi soal bentuk pilihan ganda dan uraian untuk mengukur pencapaian kompetensi peserta didik.

Buku ini menampilkan contoh tahapan pembuatan sistem perangkat lunak berorientasi objek, mulai dari analisis kebutuhan dan desain sistem. Implementasi kode program menggunakan bahasa pemrograman Java, HTML, dan PHP. Buku ini juga menyajikan beberapa kasus permasalahan di masyarakat. Peserta didik dalam kelompok dapat mendiskusikan salah satu kasus untuk diselesaikan menggunakan model pendekatan yang telah dijabarkan dalam uraian materi.

Kritik, saran, dan masukan yang membangun terhadap penyempurnaan buku ini sangat diharapkan. Semoga buku ini dapat memberi manfaat bagi peserta didik, guru Sekolah Menengah Kejuruan khususnya, dan bagi semua pihak yang membutuhkan.

Malang, Januari 2020

Penulis

A large, pink, torn-edge paper graphic is centered on the page. It has a soft shadow and a white border, giving it a 3D effect. The text is written on this graphic.

Do not Pray
for an **Easy** life,
pray for the **strength** to
endure a difficult one

*Jangan kamu berdoa untuk hidup yang mudah,
Berdoalah agar diberi kekuatan supaya bisa
menghadapi hidup yang sulit.*

"Bruce Lee"

BAB

1

Konsep Pemodelan Sistem Perangkat Lunak

Kompetensi Dasar

- 3.1 Memahami konsep pemodelan berorientasi objek
- 4.1 Mempresentasikan konsep pemodelan berorientasi objek

Tujuan Pembelajaran

Setelah mengikuti pembelajaran ini diharapkan siswa mampu

- 1. menjelaskan berbagai ragam model diagram pemodelan berorientasi objek; dan
- 2. menganalisis karakteristik pemodelan berorientasi objek.

Industri manufaktur memproduksi barang secara massal berdasarkan suatu rancangan model produk. Seorang ahli bangunan membutuhkan rancangan model untuk membuat gedung bertingkat. Industri teknologi informasi membutuhkan pemodelan perangkat lunak untuk mengimplementasikan aplikasi sistem. Berdasarkan ilustrasi tersebut pemodelan merupakan suatu proses dalam menggambarkan secara abstrak suatu model. Model dapat dikatakan sebagai rencana, representasi, atau deskripsi yang menjelaskan suatu objek, sistem, atau konsep yang berupa penyederhanaan atau idealisasi.

A. Memahami Perangkat Lunak dan Kategorinya

Perangkat lunak (*software*) adalah kumpulan instruksi atau program komputer (*syntaks/code*) yang ditulis dalam bahasa pemrograman dan ketika dijalankan menyediakan fungsi dan performa yang diinginkan. Perangkat lunak menjadi struktur data yang memungkinkan program memanipulasi informasi dan dokumen yang menggambarkan bagaimana cara menggunakan dan mengoperasikan program tersebut (Pressman, 2012).

Sistem perangkat lunak komputer dapat diklasifikasikan menjadi tiga jenis utama, yaitu perangkat lunak sistem, perangkat lunak pemrograman, dan perangkat lunak aplikasi. Perangkat lunak sistem terdiri dari sistem operasi, perangkat *driver*, *server*, dan komponen perangkat lunak. Perangkat lunak pemrograman sangat membantu dalam menulis kode program melalui alat-alat, seperti *editor*, *integrator*, *debugger*, *compiler* atau *interpreter*.

Perangkat lunak aplikasi komputer adalah kode program yang digunakan untuk mengerjakan tugas tertentu pada sistem. Perangkat lunak menggunakan kapasitas komputer secara langsung untuk mengetik dokumen, memanipulasi foto, merancang rumah, grafik, dan angka. Perangkat lunak aplikasi ini biasanya dikolaborasikan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer. Perangkat lunak aplikasi juga dapat digolongkan menjadi beberapa kelas berikut.

1. Perangkat lunak perusahaan (*enterprise*), meliputi perangkat lunak akuntansi, pendukung (*back office*), perangkat lunak bisnis, dan manajemen sumber daya manusia.
2. Perangkat lunak infrastruktur perusahaan, meliputi perangkat lunak alur kerja bisnis, sistem manajemen basis data, serta manajemen aset digital dan dokumen.
3. Perangkat lunak informasi kerja, meliputi manajemen waktu dan sumber daya, manajemen data, dokumentasi, perangkat lunak analitik, dan kolaborasi.
4. Perangkat lunak media dan hiburan, misalnya media digital dan game.
5. Perangkat lunak pendidikan, meliputi manajemen ruang kelas, manajemen survei, manajemen pelatihan, dan perangkat lunak kesiapan penjualan.
6. Perangkat lunak pengembangan media, meliputi perangkat lunak seni grafis dan penyuntingan media.
7. Perangkat lunak rekayasa produk, meliputi rekayasa perangkat keras dan rekayasa perangkat lunak.

B. Memahami Sistem Informasi

Menurut Wikipedia Indonesia, Sistem Informasi (SI) adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen. O' Brien (2004) mendefinisikan sistem informasi sebagai gabungan komponen terintegrasi dan terorganisasi dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi, dan sumber data yang digunakan dalam mengumpulkan, mengelola, mengubah, menyebarkan data, dan informasi dalam organisasi. Laudon (2010) menambahkan bahwa sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi serta menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

C. Memahami Teknik Pengembangan Sistem Perangkat Lunak

1. Pendekatan Terstruktur

Pendekatan terstruktur adalah suatu proses yang mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program dan memecah aktivitas bisnis menjadi bagian kecil yang dapat diatur dan berhubungan. Aktivitas dapat disatukan kembali menjadi satu kesatuan. Dalam pengembangan sistem informasi, pendekatan terstruktur sebagai proses yang berorientasi kepada teknik dan rekayasa. Jika suatu proses telah sampai pada suatu langkah tertentu, proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya. Ciri-ciri pendekatan terstruktur sebagai berikut.

- Merancang berdasar modul (*modularisasi*), yaitu proses yang membagi suatu sistem menjadi beberapa modul yang dapat beroperasi secara independen.
- Bekerja dengan pendekatan *top-down* yang dimulai dari level atas (secara global) kemudian diuraikan sampai ke tingkat modul (rinci).
- Kegiatan-kegiatan dilakukan secara bersamaan paralel dan iterasi.
- Dalam implementasinya antara dimensi data (variabel) dan proses (prosedur dan fungsi) dipisahkan.

2. Pendekatan Objek

Pendekatan objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Pendekatan objek menjadi suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem. Pendekatan objek memiliki beberapa keuntungan, yaitu *maintenance*. Dalam hal ini program lebih mudah dibaca dan dipahami, mudah dalam perubahan program (penambahan atau penghapusan fitur tertentu), seperti perubahan database atau program dan dapat digunakan objek sebanyak mungkin sesuai dengan yang diinginkan. Pendekatan objek memiliki beberapa karakteristik atau sifat berikut.

- Abstraksi
Abstraksi yaitu prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model sederhana dengan mengabaikan aspek-aspek yang tidak sesuai dengan permasalahan.

- b. Enkapsulasi
Enkapsulasi yaitu pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki oleh objek tersebut.
- c. Pewarisan (*inheritance*)
Pewarisan (*inheritance*) mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.
- d. Reusability
Reusability yaitu pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada kasus lain yang melibatkan objek tersebut.
- e. Generalisasi dan spesialisasi
Generalisasi dan spesialisasi yaitu menunjukkan hubungan antara kelas dan objek umum dengan kelas dan objek khusus.
- f. Komunikasi antarobjek
Komunikasi antarobjek yaitu dilakukan lewat pesan yang dikirim dari satu objek ke objek lainnya.
- g. *Polymorphism*
Polymorphism adalah kemampuan suatu objek untuk digunakan di banyak tujuan berbeda dengan nama yang sama sehingga menghemat baris program.

D. Memahami Pemodelan dalam Rekayasa Perangkat Lunak

Pemodelan merupakan suatu proses dalam menggambarkan secara abstrak suatu model. Model diartikan sebagai rencana, representasi, atau deskripsi yang menjelaskan suatu objek, sistem, atau konsep yang berupa penyederhanaan atau idealisasi. Bentuknya dapat berupa model fisik (maket dan prototipe), model citra (gambar rancangan dan citra komputer), atau rumusan matematis. Proses pemodelan menampilkan deskripsi suatu proses dari beberapa perspektif tertentu. Proses pemodelan perangkat lunak menjadi aktivitas yang saling terkait (koheren) untuk menspesifikasikan, merancang, implementasi kode program, dan pengujian sistem perangkat lunak (Romi, 2006).

Pemodelan dalam rekayasa perangkat lunak menjadi aktivitas yang dilakukan di tahapan awal pengembangan aplikasi dan akan memengaruhi pekerjaan lain. Beberapa pemodelan yang dikenal dalam pengembangan aplikasi atau rekayasa perangkat lunak sebagai berikut.

1. Pemodelan Proses

Model yang digunakan dalam pengembangan sistem perangkat lunak (*software development life cycle process*) terdiri dari tahapan spesifikasi kebutuhan perangkat lunak sampai perawatan sistem.

2. Pemodelan Analisis dan Desain

Model ini berisi spesifikasi lengkap dari persyaratan representasi desain yang komprehensif bagi perangkat lunak. Model ini harus memenuhi tiga sasaran utama, yaitu menggambarkan yang dibutuhkan pelanggan, membangun dasar bagi pembuatan desain perangkat lunak, dan membatasi persyaratan yang dapat divalidasi.

3. Pemodelan Data

Model ini merupakan sarana melakukan abstraksi data, relasi antarentitas dalam basis data, dan konsep membuat deskripsi struktur basis data. Model ini memuat

spesifikasi operasi dasar (*basic operation*) dalam pengaksesan dan pembaharuan data, serta tabiat data (*data behavior*).

4. **Pemodelan Fungsional dan Aliran Informasi**

Informasi ditransformasikan saat mengalir melalui sebuah sistem berbasis komputer. Sistem menerima input dengan berbagai cara dan menghasilkan output sehingga dapat menciptakan suatu model aliran bagi setiap sistem tanpa melihat ukuran dan kompleksitasnya. Contoh: *Data Flow Diagram* (DFD).

5. **Pemodelan Unified Modeling Language (UML)**

Pemodelan Unified Modeling Language (UML) merupakan suatu bahasa yang berisi aturan-aturan, notasi-notasi, dan gambar untuk membuat visualisasi model seperangkat lunak dengan pendekatan sistem berorientasi objek.

E.

Memahami Karakteristik Pemodelan Perangkat Lunak

Pengembangan perangkat lunak merupakan proses yang kompleks dan melibatkan banyak aktivitas. Proses pemodelan analisis dalam pengembangan perangkat lunak memiliki atribut dan karakteristik spesifik dibandingkan dengan pemodelan bidang lain. Menurut Sitemaster (2017), karakteristik perangkat lunak sebagai berikut.

1. **Understandability**

Understandability yaitu sejauh mana proses secara eksplisit ditentukan dan bagaimana kemudahan definisi proses itu dimengerti.

2. **Visibility**

Visibility apakah aktivitas-aktivitas proses mencapai titik akhir dalam hasil yang jelas sehingga kemajuan dari proses tersebut dapat terlihat nyata.

3. **Supportability**

Supportability yaitu sejauh mana aktivitas proses dapat didukung oleh *Computer Aided Software Engineering* (CASE).

4. **Acceptability**

Acceptability adalah proses yang telah ditentukan dapat diterima dan digunakan serta mampu dipertanggungjawabkan selama pembuatan produk perangkat lunak.

5. **Reliability**

Reliability adalah proses yang didesain sedikikan rupa sehingga kesalahan proses dapat dihindari sebelum terjadi kesalahan pada produk.

6. **Robustness**

Robustness adalah proses yang terus berjalan walaupun terjadi masalah yang tidak diduga.

7. **Maintainability**

Maintainability adalah proses berkembang untuk mengikuti kebutuhan atau perbaikan.

8. **Rapidity**

Rapidity adalah kecepatan proses pengiriman sistem dapat secara lengkap memenuhi spesifikasi.

Rekayasa pemodelan sistem merupakan pendefinisian kebutuhan sistem yang akan dibangun untuk mengetahui kebutuhan yang diperlukan analisis sistem. Analisis didefinisikan sebagai penguraian suatu pokok atas berbagai bagiannya dan penelaahan bagian-bagian serta hubungannya untuk memperoleh pengertian yang tepat dan pemahaman arti secara keseluruhan. Pemanfaatan suatu metode analisis menjadi hal penting agar kebutuhan sistem dapat diketahui. Metodologi sebagai cara sistematis untuk mengerjakan pekerjaan analisis dan desain. Manfaat penggunaan metodologi bagi pihak yang membangun dapat merencanakan dan mengulangi pekerjaan di lain waktu. Metodologi dapat menghilangkan kesalahpahaman dan menghilangkan perbedaan notasi untuk suatu hal yang sama. Menurut Fowler (2015), metode yang digunakan dalam analisis berorientasi objek sebagai berikut.

1. Metode Booch (Metode Desain Object Oriented)

Proses analisis dan desain dilakukan dalam empat tahap iteratif, yaitu identifikasi kelas/objek, identifikasi semantik dan hubungan kelas/objek, perincian *interface*, dan implementasi.

2. Metode Rumbaugh (Object Modelling Technique/OMT)

Metode ini berdasarkan analisis terstruktur dan pemodelan *entity-relationship*. Tahapan utama berupa analisis, desain sistem, desain objek, dan implementasi. Keunggulan metode ini terletak pada penotasiannya yang mendukung semua konsep *object oriented*.

3. Metode Jacobson (Object Oriented Software Engineering/OOSE)

Metode ini berisi elemen dari *object oriented* yang menekankan pada *use case*. Metode ini memiliki tiga tahap, yaitu membuat model *requirement* dan analisis, desain implementasi, serta model pengujian. Keunggulan metode ini mudah dipelajari karena memiliki notasi yang sederhana dan mencakup seluruh tahapan dalam rekayasa perangkat lunak.

4. Metode Coad dan Yourdon

Metode ini berdasarkan pemodelan *object oriented* dan *entity relationship*. Model ini mempunyai perancangan yang berfokus pada empat komponen, yaitu *problem domain component*, *human interaction component*, *data management component*, dan *task management component*.

5. Metode Wirfs/Brock Responsibility Driven Design/Class Responsibility Collaboration (RDD/CFC)

Metode ini diarahkan pada desain yang sangat berguna untuk memunculkan ide dalam tahap analisis. Keunggulan metode ini mudah digunakan dan mengidentifikasi hierarki kelas dan subsistemnya.

6. Metode Shlair Mellor/Object Oriented Analysis/Design (OOA/D)

Metode ini menggunakan teknik pemodelan informasi tradisional, menjelaskan entitas sistem, *state diagram* memodelkan keadaan entitas, dan *data flow diagram* yang memodelkan alur data sistem. Metode ini meliputi *information model*, *state model*, dan *process model*. Keunggulannya mudah dikonversi dari metode struktural.

UML merupakan suatu pemodelan atau sebuah bahasa standar untuk merancang, mendokumentasikan, dan membuat model semua jenis sistem perangkat lunak khususnya sistem berorientasi objek. Aplikasi ini dapat berjalan pada semua perangkat keras, semua sistem operasi dan jaringan, serta dapat ditulis dalam semua bahasa pemrograman. UML menggunakan *class* dan *operation* sehingga lebih tepat digunakan untuk penulisan perangkat lunak dalam bahasa berorientasi objek, seperti C++, Java, C#, atau VB.NET. Notasi UML menggambarkan berbagai diagram peranti lunak dan diturunkan dari tiga notasi, yaitu OOD, OMT, dan OOSE. Karakteristik dan fungsi UML dalam rekayasa pengembangan perangkat lunak sebagai berikut.

1. Bahasa Pemodelan

UML sebagai bahasa pemodelan yang memiliki perbendaharaan kata, fokus pada konseptual, dan fisik dari suatu sistem.

2. Mengambarkan Sistem

UML tidak hanya menjadi rangkaian simbol grafis, tetapi penerapan semantik yang baik dan mudah dimengerti.

3. Menspesifikasikan Sistem

UML membangun model yang sesuai dan lengkap, meliputi spesifikasi analisis, desain, dan implementasi.

4. Mendokumentasikan Sistem

UML menunjukkan dokumentasi dari arsitektur sistem dan detail dari semuanya. Tujuan penggunaan model UML untuk memberikan model yang siap digunakan, mendukung bahasa pemodelan visual ekspresif, mengembangkan sistem, saling tukar-menukar model dengan mudah dan dimengerti secara umum, memberikan bahasa pemodelan yang bebas dari bahasa pemrograman dan proses rekayasa, dan menyatukan praktik terbaik dalam pemodelan. Tahun 1995 diluncurkan UML versi 0.8 dan pada tahun 1997 munculnya UML versi 1.1 Tahun 1999 diterbitkan buku UML oleh Booch, Rumbaugh, dan Jacobson. Pada tahun 1999 UML menjadi standar bahasa pemodelan sistem berorientasi objek. Tahun 2003 diluncurkan UML versi 1.5, lalu tahun 2004 diluncurkan UML versi 2.0. Menurut Dharwiyanti (2003) berikut ragam jenis model UML versi 1.5.

- a. Use case diagram
Menggambarkan diagram fungsionalitas (kebutuhan atau *requirement*) yang diharapkan dari sebuah sistem.
- b. Class diagram
Menggambarkan struktur dan deskripsi dari *class* beserta hubungan antara satu *class* dengan lainnya (pewarisan, asosiasi, dan lain-lain).
- c. Object diagram
Menggambarkan struktur dan deskripsi objek-objek yang terkait dengan sistem.
- d. Statechart diagram
Menggambarkan perilaku/*behaviour* dari suatu *case* atau *class* tertentu dalam sistem.
- e. Activity diagram
Menggambarkan proses yang berjalan dari suatu sistem atau *case* tertentu.

- f. Sequence diagram
Menggambarkan interaksi antarobjek di dalam dan di sekitar sistem.
- g. Collaboration diagram
Menggambarkan interaksi antarobjek yang lebih menekankan peran objek.
- h. Component diagram
Menggambarkan struktur dan hubungan antarkomponen terkait dengan sistem perangkat lunak.
- i. Deployment diagram
Menggambarkan deskripsi detail bagaimana komponen dibangun (*deploy*) dalam infrastruktur sistem perangkat lunak.

Tugas Kelompok

1. Refleksi dan Orientasi Permasalahan

Pemerintah daerah “XXX” bekerja sama dengan pengembang perangkat lunak “YYY” dan akan membuat aplikasi sistem informasi Usaha Kecil Menengah (UKM). Pengembang perangkat lunak “YYY” menggunakan konsep objek dan dalam implementasinya. Perangkat lunak akan digunakan membantu proses bisnis atau aktivitas pada UKM yang dikelola oleh pemerintah daerah “XXX”. Aplikasi dibuat menggunakan bahasa pemrograman Java. Langkah awal yang dilakukan pengembang perangkat lunak untuk membuat model sistem aplikasi perangkat lunak.

2. Organisasi Belajar

Berdasarkan uraian pada orientasi masalah, buatlah kelompok diskusi yang terdiri dari 3–5 orang. Diskusikan dan organisasikan anggota diskusi untuk menyelesaikan permasalahan dan buatlah deskripsi tugas setiap anggota kelompok dengan membagi tugas kepada setiap individu dalam kelompok. Jawablah pertanyaan di bawah ini, kemudian identifikasikan rumusan permasalahan dan langkah-langkah yang harus dilakukan oleh pengembang perangkat lunak tersebut.

- a. Mengapa harus membuat model perangkat lunak dan apa manfaatnya bagi pembuat perangkat lunak (developer), pemerintah daerah “XXX”, dan pengguna?
- b. Apa kategori aplikasi yang akan dibuat dan data yang dibutuhkan?
- c. Apa informasi yang didapatkan dan siapa yang terlibat pengembangan aplikasi tersebut?
- d. Apa metode atau pendekatan yang digunakan untuk pembuatan model sistem?

3. Penyelidikan Individu atau Kelompok (Research)

Merujuk kepada uraian materi, sumber bacaan, dan referensi lainnya, serta hasil pembagian tugas, lakukan langkah-langkah dan tampilkan hasilnya dalam tabel berikut.

- a. Identifikasi aplikasi perangkat lunak dan sistem informasi
Identifikasi serta jelaskan tentang perbedaan konsep aplikasi perangkat lunak dan aplikasi sistem informasi. Identifikasi konsep yang meliputi konsep, tipe/ jenis/klasifikasi, contoh aplikasi, serta fungsi dan kegunaan setiap aplikasi.
- b. Identifikasi jenis, fungsi dan kegunaan pemodelan sistem
Identifikasikan dan jelaskan alasan dibutuhkannya pemodelan sistem. Jelaskan fungsi dan kegunaannya bagi pembuat perangkat lunak (developer) dan pengguna (*user*), serta metode atau pendekatan apa saja yang dapat diterapkan.

- c. Identifikasi model sistem dengan pendekatan terstruktur
Identifikasikan beberapa jenis diagram model pendekatan terstruktur (DFD, ERD, dan *structure chart diagram*) dengan parameter, antara lain definisi/kegunaan/fungsi diagram, konsep utama atau *main concept*, dan notasi/symbol yang digunakan.
- d. Identifikasi model sistem menggunakan pendekatan objek
Identifikasikan beberapa jenis diagram model pendekatan berorientasi objek (diagram UML), meliputi *use case diagram*, *activity diagram*, *class diagram*, *component diagram*, dan lainnya dengan parameter konsep/kegunaan/fungsi diagram, konsep utama, *main concept*, dan notasi/symbol yang digunakan (tergantung *tools* yang digunakan).
- e. Menganalisis perbedaan model sistem terstruktur dan objek
Identifikasi dan analisis perbedaan model sistem menggunakan pendekatan terstruktur dan objek. Parameter yang digunakan adalah konsep, dekomposisi permasalahan, ragam jenis diagram, *main concept*, karakteristik, pengelolaan atau struktur data, proses (dapat ditunjukkan dalam suatu contoh), dan sebagainya.
- f. Menyajikan hasil karya (metode *windows shopping*)
Lakukan langkah-langkah berikut ini.
 - 1) Buatlah laporan dan kesimpulan hasil diskusi.
 - 2) Presentasikan hasil diskusi ke kelompok lain dengan cara berikut.
 - a) Setiap anggota kelompok mendapatkan uang kertas mainan atau stiker untuk membeli produk hasil diskusi kelompok.
 - b) Setiap kelompok ditunjuk satu orang untuk memaparkan hasil diskusi.
 - c) Anggota kelompok lain berkunjung ke kelompok untuk mendengarkan hasil presentasi dan melakukan tanya jawab (waktu 10 menit)
 - d) Setelah 10 menit, anggota kelompok tersebut berpindah ke kelompok lainnya untuk mendengarkan paparan hasil diskusi dan melakukan tanya jawab.
- g. Menganalisis dan mengevaluasi
Setelah mendengarkan paparan hasil dari semua kelompok, lakukan hal berikut.
 - 1) Setiap anggota kelompok melakukan analisis dan evaluasi paparan hasil diskusi. Hasilnya menetapkan kelompok yang paling baik. Kriteria analisis, meliputi ketepatan, kebenaran, kedalaman pembahasan, penyajian diskusi, kemampuan menjawab pertanyaan, serta kemampuan presentasi dan komunikasi.
 - 2) Setiap anggota kelompok memberikan stiker atau memberikan uang kertas mainan (membeli produk) ke kelompok yang paling unggul dan baik.
 - 3) Setiap kelompok menghitung hasil perolehan produk hasil diskusi.

Rangkuman

Pemodelan dalam suatu rekayasa perangkat lunak merupakan suatu aktivitas atau kegiatan yang dilakukan di tahapan awal pengembangan aplikasi perangkat lunak. Pemodelan analisis dan desain mendefinisikan spesifikasi lengkap dari persyaratan representasi dan representasi desain yang komprehensif bagi perangkat lunak yang akan dibangun. Pemodelan data menjadi sarana untuk melakukan abstraksi data dan relasi antarentitas dalam basis data. Hal ini menjadi sejumlah konsep untuk membuat diskripsi stuktur basis data. Arsitektur perangkat lunak adalah sebuah spesifikasi sistem abstrak yang utama dan terdiri dari komponen fungsional yang menjelaskan beberapa hal tentang perilaku sistem dan antarmuka serta hubungan komponen-komponen sistem. UML menjadi bahasa pemodelan standar untuk visualisasi, merancang, dan mendokumentasikan serta membuat model untuk semua jenis sistem aplikasi perangkat lunak khususnya pendekatan berorientasi objek.

Uji Kompetensi

A. Soal Pilihan Ganda

Pilihlah salah satu jawaban yang paling tepat!

1. Diagram yang menggambarkan proses yang berjalan dari suatu sistem atau kejadian tertentu disebut
 - a. *class diagram*
 - b. *use case diagram*
 - c. *activity diagram*
 - d. *statechart diagram*
 - e. *room diagram*
2. Diagram yang menggambarkan perilaku atau *behaviour* dari suatu kejadian (case) atau class tertentu dalam sistem adalah
 - a. *class diagram*
 - b. *use case diagram*
 - c. *activity diagram*
 - d. *statechart diagram*
 - e. *room diagram*
3. Pernyataan paling benar yang menjelaskan pendekatan objek adalah
 - a. pendekatan secara *top and down*
 - b. modularisasi program
 - c. pembungkusan data dan fungsi
 - d. kegiatan dilakukan secara paralel
 - e. pendekatan data dan fungsi

4. Kemampuan suatu objek untuk dapat digunakan pada banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program
 - a. pewarisan (*inheritance*)
 - b. enkapsulasi
 - c. reusability
 - d. polymorphisme
 - e. reuse
5. Penggunaan ulang suatu objek yang sudah didefinisikan untuk suatu permasalahan lain yang melibatkan objek tersebut adalah
 - a. pewarisan (*inheritance*)
 - b. enkapsulasi
 - c. reusability
 - d. polymorphisme
 - e. reuse
6. Pernyataan berikut menjelaskan entitas yang berupa konsep atau abstrak adalah
 - a. pegawai, penjualan, dan mata pelajaran
 - b. departemen, pekerjaan, dan mata pelajaran
 - c. departemen, peserta didik, dan peminjaman
 - d. pegawai, pekerjaan, dan pembelian
 - e. pegawai, pekerjaan, dan mata pelajaran
7. Pernyataan berikut menjelaskan entitas yang berupa *event* atau kejadian adalah
 - a. pegawai, penjualan, dan mata pelajaran
 - b. departemen, peminjaman, dan mata pelajaran
 - c. pembelian, penjualan, dan peminjaman
 - d. pegawai, pekerjaan, dan pembelian
 - e. departemen, peserta didik, dan peminjaman
8. Model diagram menggambarkan arsitektur perangkat lunak secara keseluruhan tanpa memperlihatkan proses pemilihan dan pengulangannya secara rinci adalah
 - a. class diagram
 - b. data flow diagram
 - c. state chart diagram
 - d. struktur chart diagram
 - e. model diagram
9. Pernyataan yang menjelaskan aplikasi infrastruktur perusahaan adalah
 - a. aplikasi manajemen aset digital, dan aplikasi manajemen dokumen
 - b. aplikasi akuntansi, aplikasi back office, dan aplikasi SDM
 - c. aplikasi akuntansi dan aplikasi manajemen dokumen
 - d. aplikasi back office dan aplikasi manajemen aset digital
 - e. aplikasi back office dan aplikasi SDM

10. Perbedaan pemodelan prosedural dengan pemodelan berorientasi objek adalah
- pemodelan prosedural dapat menerapkan penggunaan beberapa prosedur dengan nama yang sama dan parameter berbeda
 - pemodelan berorientasi objek memecah dimensi proses menjadi beberapa bagian yang terpisah dari dimensi data
 - pemodelan berorientasi objek menggabungkan dimensi data dan proses menjadi satu kesatuan
 - pemodelan prosedural mengelompokkan variabel dan prosedur dalam suatu struktur data
 - pemodelan prosedural mengelompokkan beberapa bagian yang terpisah dari dimensi data
11. Representasikan dunia nyata yang kompleks menjadi bentuk model sederhana dan mengabaikan aspek yang tidak sesuai dengan permasalahan adalah
- abstraksi
 - enkapsulasi
 - reusability*
 - polimorfisme*
 - reuse*
12. Proses yang telah ditentukan dapat diterima, digunakan, dan mampu dipertanggungjawabkan selama pembuatan produk perangkat lunak adalah
- reliability*
 - visibility*
 - reusability*
 - acceptability*
 - polimorfisme*
13. Diagram yang menggambarkan fungsionalitas (kebutuhan atau *requirement*) yang diharapkan dari sebuah sistem adalah
- class diagram*
 - use case diagram*
 - activity diagram*
 - statechart diagram*
 - reusability diagram*
14. Program lebih mudah dipahami dan mudah untuk menambah atau menghapus fitur sehingga dapat digunakan pada banyak objek. Hal ini menjadi keuntungan pendekatan objek dalam hal
- validation*
 - verification*
 - maintenance*
 - implementation*
 - selection*

15. Aktivitas proses harus mencapai titik akhir dalam hasil yang jelas sehingga kemajuan proses dapat terlihat nyata merupakan karakteristik pemodelan
- reliability*
 - visibility*
 - reusability*
 - acceptability*
 - verification*

B. Soal Esai

Jawablah dengan tepat dan benar.

- Perangkat lunak adalah kumpulan ... yang ditulis menggunakan ... dan ketika dijalankan menyediakan ... yang diinginkan (tujuan/tugas tertentu).
- Perangkat lunak alur kerja bisnis, sistem manajemen aset digital, dan sistem manajemen dokumen termasuk dalam jenis perangkat lunak
- Sistem informasi adalah perangkat lunak aplikasi komputer untuk mendukung ... dan ... dari suatu organisasi.
- Dalam implementasinya salah satu ciri dari pemrograman berorientasi objek adalah menggabungkan antara dimensi ... dan
- ... adalah kemampuan suatu objek untuk banyak tujuan yang berbeda dengan nama yang sama sehingga mempunyai kelebihan dalam implementasi kode program dibandingkan dengan pendekatan terstruktur disebut
- ... mendefinisikan spesifikasi lengkap persyaratan dan representasi rancangan yang komprehensif bagi perangkat lunak yang akan dibangun.
- Metode ... memberikan penekanan pada *usecase* dan memiliki tiga tahap, yaitu model *requirement* dan analisis, desain implementasi, serta model pengujian.
- UML memiliki kosakata dan cara mempresentasikan rancangan yang berfokus pada konseptual dan fisik sistem. Hal ini merupakan karakteristik UML sebagai ...
- Use case diagram* menggambarkan ... dari sistem yang akan dibangun.
- Bisnis proses model merupakan suatu ... yang mengidentifikasi ... mengaturnya secara ... menampilkan ... secara grafis dari suatu organisasi atau sistem.

C. Soal Esai Uraian

Jawablah dengan ringkas dan benar.

1. Analisislah perbedaan model terstruktur dengan objek serta berikan satu kasus permasalahan yang diselesaikan dengan kedua pendekatan tersebut.
2. Jelaskan prinsip-prinsip atau karakteristik pemodelan perangkat lunak.
3. Jelaskan tahap pengembangan perangkat lunak (*software development life cycle*).
4. Jelaskan perbedaan model DFD dan *use case diagram* dari segi fungsi atau kegunaanya.
5. Jelaskan perbedaan prinsip-prinsip sistem berorientasi objek yang meliputi *encapsulation*, *inheritance*, dan *polimorfisme*.