PERANCANGAN SISTEM BERORIENTASI OBJEK

DENGAN PEMODELAN UML (UNIFIED MODELING LANGUAGE) TOOLS

SRI ANARDANI

Buku ini disusun untuk membantu memahami
bagaimana membuat perancangan berorientasi
objek menggunakan pemodelan Unified Modeling Language (UML)

Materi pembahasan buku meliputi
- > Konsep perancangan berorientasi objek
- > Pemodelan UML
- > Penerapan pemodelan UML pada studi kasus

Semoga Buku Ajar ini dapat memberikan manfaat bagi pembaca
Buku ini jauh dari sempurna, untuk itu kritik dan saran membangun
sangat diharapkan oleh penulis.

UNIPMA Press

Penerbit UNIPMA Press
Universitas PGRI Madiun
Jl. Setiabudi No. 85 Madiun Jawa Timur 63118
Telp. (0351) 462986, Fax. (0351) 45400
Email : upress@unipma.ac.id
Website : kwu.unipma.ac.id

SRI ANARDANI

Perancangan Sistem Berorientasi Objek
Dengan Pemodelan UML (Unified Modeling Language) Tools

UNIPMA Press

PERANCANGAN SISTEM BERORIENTASI OBJEK

DENGAN PEMODELAN UML (UNIFIED MODELING LANGUAGE) TOOLS

SRI ANARDANI

PERANCANGAN SISTEM BERORIENTASI OBJEK DENGAN PEMODELAN UML (*UNIFIED MODELING LANGUAGE*) *TOOLS*

PERANCANGAN SISTEM BERORIENTASI OBJEK DENGAN PEMODELAN UML (*UNIFIED MODELING LANGUAGE*) *TOOLS*

Sri Anardani

UNIPMA Press
WE GOT IT

PERANCANGAN SISTEM BERORIENTASI OBJEK DENGAN PEMODELAN UML (*UNIFIED MODELING LANGUAGE*) *TOOLS*

Penulis:
Sri Anardani, S.Kom, M.T

Editor:
Estuning Dewi Hapsari, S.Pd, M.Pd

Perancang Sampul:
Yoga Prisma Yudha, S.Kom, M.Kom

Penata Letak:
Slamet Riyanto, ST, MM

Cetakan Pertama, Oktober 2019

Diterbitkan Oleh:
UNIPMA Press (Anggota IKAPI)
Universitas PGRI Madiun
Jl. Setiabudi No. 85 Madiun Jawa Timur 63118
Telp. (0351) 462986, Fax. (0351) 459400
E-Mail: upress@unipma.ac.id
Website: kwu.unipma.ac.id

ISBN: 978-602-0725-58-1

KATA PENGANTAR

Bismillahirrohmanirrohim,

Puji dan syukur atas nikmat dan rahmat-Nya buku ini dapat diselesaikan dengan baik. Buku ini membahas perancangan pemodelan sistem informasi berorientasi obyek dengan menggunakan *Unified Modeling Language* (UML). Harapannya buku ini dapat memberikan tambahan wawasan dan memberi pengaruh positif bagi perkembangan ilmu pengethauna dibidang teknologi informasi.

Dalam kesempatan ini penulis menyampaikan terima kasih kepada:

1. LPPM Universitas PGRI Madiun yang telah memberikan dukungan luar biasa bagi penulis.
2. Kelurga besar Program Studi Informatika Universitas PGRI Madiun yang telah memberikan masukan dan bantuan pemikiran
3. Suamiku (Crismantoro) dan putraku (Bagas) terima kasih dukungannya selama ini.
4. Semua pihak yang telah mendorong dan membantu penulis dalam menyelesaikan buku ini.

Akhir kata, saya menyadari bahwa penyusunan buku ini jauh dari sempurna, oleh sebab itu saya sangat mengharapakan kritik dan saran untuk menyempurnakan.

Madiun, September 2019

Sri Anardani

DAFTAR ISI

Kata Pengantar i
Daftar Isi ii
Bab 1. Perancangan Sistem 1
1.1. Konsep Dasar Perancangan Sistem 1
1.2. Tujuan Perancangan Sistem 2
1.3. Personil Yang Terlibat Dalam Perancangan Sistem 3
Bab 2. Sistem Berorientasi Obyek 5
2.1. Pengertian dan Konsep Berorientasi Obyek 5
2.2. Metodologi Berorientasi Obyek 7
2.3. Analisis Berorientasi Obyek 9
2.4. Perancangan Berorientasi Obyek 10
Bab 3. Unified Modeling Language (UML) 13
3.1. Sejarah dan Pengertian Dasar UML 13
3.2. Pemanfaatan UML dalam Pengembangan Perangkat Lunak 14
3.3. Klasifikasi Diagram UML 16
Bab 4. Use Case System 18
4.1. Konsep Dasar Pemodelan Use Case Diagram 18
4.2. Bagaimana Mengidentifikasi Aktor 20
4.3. Bagaimana Mengidentifikasi Use Case 21
4.4. Relasi Diagram Use Case 23
4.4.1. Relasi Asosiasi 24
4.4.2. Relasi Include 24
4.4.3. Relasi Extend 25
4.4.4. Relasi Generalisasi 26
4.5. Use Case Spesification 27

4.6. Contoh Studi Kasus 28
4.7. Latihan Soal 55
Bab 5. Interaction Diagram 56
5.1. Diagram Interaksi 56
5.2. Diagram Sekuensial 56
5.3. Diagram Kolaborasi 66
5.4. Latihan Soal 67
Bab 6. Class Diagram 68
6.1. Pengertian Class Diagram 68
6.2. Menemukan Class 70
6.3. Multiplicity Class 70
6.4. Relasi Antar Class 71
6.5. Contoh Studi Kasus 75
6.6. Latihan Soal 79
Bab 7. Diagram Aktivitas 80
7.1. Pengertian Diagram Aktifitas 80
7.2. Implementasi Diagram Aktifitas 81
7.3. Contoh Studi Kasus 84
7.4. Latihan Soal 85
Bab 8. Studi Kasus Sistem Informasi Prestasi Mahasiswa 86
8.1. Definisi Permasalahan 86
8.2. Diagram Use Case 86
8.3. Spesifikasi Use Case 88
8.4. Pemodelan Diagram Aktifitas 104
8.5. Pemodelan Class Diagram 105
8.6. Pemodelan Diagram Sekuensial 107
Bab 9. Studi Kasus Sistem Informasi Kepegawaian 115

9.1. Definisi Permasalahan 115
9.2. Use Case Diagram 115
9.3. Class Diagram 136
9.4. Diagram Kolaborasi 142
9.5. Perancangan Antarmuka 148
Daftar Pustaka 155
Indeks 157

BAB 1 PERANCANGAN SISTEM

Tujuan Instruksional:

1. Dapat menjelaskan paradigma konsep perancangan sistem.
2. Dapat menjelaskan tujuan perancangan sistem

1.1. KONSEP DASAR PERANCANGAN SISTEM

Tahap perancangan adalah merupakan bagian dari langkah-langkah siklus pengembangan sistem informasi. Tahap perancangan dilaksanakan setelah tahapan analisis kebutuhan sistem dilalui. Pada tahapan analisis kebutuhan sistem akan digali kebutuhan pengguna dari perangkat lunak yang akan dibangun. Proses perancangan adalah bagaimana sebuah perangkat lunak akan dibangun. Fase perancangan harus mampu menampilkan kerjasama berbagai bagian yang akan terlibat dalam sistem.

Proses perancangan adalah sebuah tempat berisi batasan kreativitas berupa kebutuhan stakeholder, kebutuhan bisnis dan pertimbangan teknikal yang secara bersamaan menjadi formula sebuah sistem informasi (Pressman dan Maxim, 2015)

Perancangan sistem berkaitan dengan perancangan masukan, proses, dan keluaran. Perancangan merupakan kumpulan dari beberapa elemen yang berhubungan dan melaksanakan tanggungjawabnya dalam proses yang menghasilkan keluaran (Munawar, 2018) . Hasil dari rancangan akan berdampak pada arsitektur sistem secara keseluruhan. Proses perancangan memegang peranan penting sebagai dasar untuk mengambil keputusan apakah perangkat lunak akan dibangun atau tidak kedepannya.

Perancangan berbeda dengan fase analisis, pada fase analisis lebih menekankan pada penyelidikan permasalahan dan solusi yang dibutuhkan sedangkan perancangan lebih menekankan pada solusi konseptual sesuai persyaratan implementasi sistem. Contoh dari proses perancangan adalah deskripsi dari skema database yang akan di implementasikan. Fase perancangan fokus pada bagaimana sistem baru akan di implementasikan, sedangkan fase analisis fokus pada bagaimana proses bisnis dapat berjalan dengan cara yang lebih baik dengan sistem baru yang akan dikembangkan.

Perubahan lingkungan bisnis yang cepat sangat mempengaruhi proses bisnis organisasi. Perubahan ini berdampak pada kebutuhan informasi bagi organisasi. Kebutuhan akan informasi yang berubah-ubah ini membutuhkan penyesuaian sistem lama ke sistem baru. Penyesuaian sistem membutuhkan analisis kebutuhan sistem, hal inilah yang mengakibatkan proses perancangan sistem bergantung pada kejelian tahapan analisis kebutuhan sistem. Hasil akhir analisis kebutuhan sistem adalah sebuah spesifikasi usulan kebutuhan pembangunan sistem yang akan dikembangkan.

Fase perancangan adalah tahap memutuskan bagaimana sistem akan beroperasi dan bagaimana mengidentifikasi kebutuhan perangkat keras, perangkat lunak, infrastruktur jaringan, antarmuka pengguna, formulir dan laporan yang akan digunakan (Gould, 2012)

1.2. TUJUAN PERANCANGAN SISTEM

Perangkat lunak memainkan peranan penting pada hampir semua aspek kehidupan sehari-hari di semua bidang kehidupan. Produk perangkat lunak membantu manusia untuk lebih efisien dan produktif. Proses perancangan sistem adalah merupakan salah satu tahapan siklus dari pengembangan perangkat lunak. Proses perancangan sistem memegang

peranan penting dalam siklus pengembangan perangkat lunak. Hasil dari proses perancangan akan memegang peranan penting pada kondisi arsitektur sistem secara keseluruhan.

Tujuan utama dari tahapan perancangan sistem adalah memberikan gambaran kepada para pengguna terkait sistem baru yang akan dibangun. Dokumen perancangan sistem adalah jembatan komunikasi antara tim pengembang dengan pengguna untuk meminimalkan kesalahan dalam tahapan pembangunan sistem.

Dokumen perancangan yang dihasilkan akan memuat model sistem yang akan dibangun, sehingga solusi atas permasalahan dari sistem lama dapat diwujudkan ke dalam proses pembangunan sistem. Pemodelan yang dihasilkan dari proses perancangan terdiri atas model input, model database, model output. Hasil dari perancangan ini akan menjadi alat penerjemah dari tim untuk proses penulisan bahasa pemrograman.

1.3. PERSONIL YANG TERLIBAT DALAM PERANCANGAN SISTEM.

Sebuah pengembangan perangkat lunak adalah kegiatan tim, karena permasalahn yang begitu kompleks tidak mungkin dikerjakan oleh perorangan. Kemampuan dari tim ini akan berdampak pada produktivitas perangkat lunak. Di dalam tim pengembangan perangkat lunak ada beberapa keahlian yang dibutuhkan dari masing-masing personil yaitu keahlian menganalisis masalah, keahlian perancangan dan arsitektur, keahlian pemrograman serta keahlian pengujian sistem. Keahlian dari tiap-tiap anggota tim diharapkan dapat membantu menetapkan kebutuhan secara efektif dan efisien.

Personil untuk tim yang terlibat dalam proses perancangan sistem adalah yang memiliki beberapa kemampuan, diantaranya adalah:

1. Keahlian untuk mengendalikan teknologi informasi.
2. Keahlian sebagai teknisi komunikasi data dan jaringan komputer.
3. Keahlian dalam mengolah struktur data dalam basis data.

BAB 2 SISTEM BERORIENTASI OBYEK

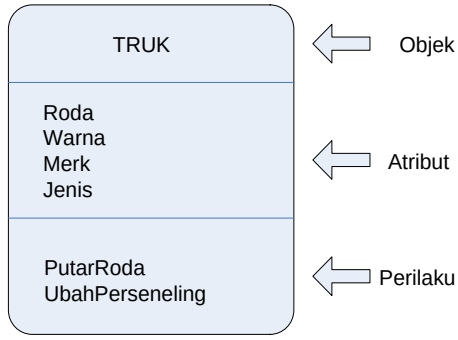
Tujuan Instruksional:

1. Dapat menjelaskan paradigma konsep analisa dan perancangan sistem berorientasi obyek.
2. Dapat menjelaskan metodologi sistem berorientasi obyek.
3. Dapat menjelaskan beberapa istilah pada sistem berorientasi obyek

2.1. PENGERTIAN DAN KONSEP BERORIENTASI OBYEK

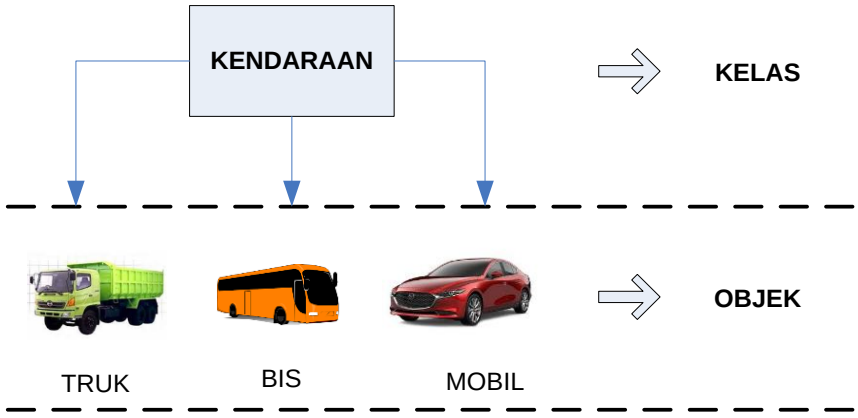
Berorientasi objek atau yang dikenal dengan objek oriented merupakan trend baru dalam rekayasa dalam rekayasa perangkat lunak dimana sistem dipandang sebagai kumpulan-kumpulan objek yang saling berinteraksi (Sholiq, 2006). Kumpulan objek-objek yang saling bekerjasama ini diatur oleh perilaku (*behavior*).

Berorientasi objek menyelesaikan masalah menggunakan pemodelan yang dibuat berdasarkan kondisi dunia nyata. Objek adalah segala sesuatu yang ada di lingkungan sekitar kita yang menyususn dunia. Setiap objek memiliki atribut (*attribute*) dan perilaku (*behavior*). Atribut adalah informasi yang berkaitan dengan objek, sedangkan perilaku adalah operasi yang mengatur objek. Contoh dari objek adalah truk, bis, mobil. Berikut adalah struktur dari sebuah objek dengan atribut dan perilakunya



Gambar 2.1.1. Struktur Objek

Beberapa objek yang memiliki kesamaan atribut perilaku dapat dikelompokkan dalam satu kategori misalkan objek truk, bis dan mobil dapat dikelompokkan ke dalam kategori kendaraan. Pengelompokan beberapa objek yang memilki kesamaan atribut dan operasi dalam satu kategori disebut dengan kelas. Objek adalah contoh dari sebuah kelas, objek dan kelas sering sama sebagai benda dalam deskripsi masalah.



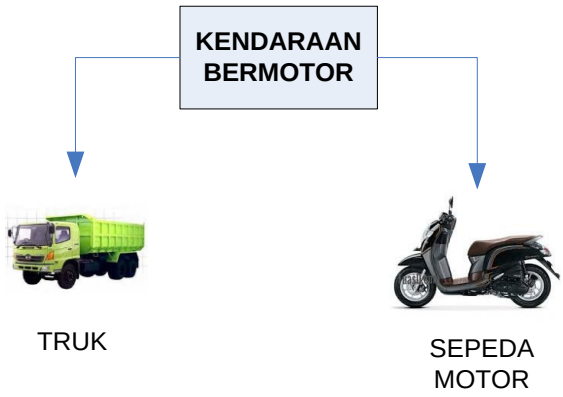
Gambar 2.1.2 Hubungan Kelas dan Objek

2.2. METODOLOGI BERORIENTASI OBYEK

Metodologi pengembangan berorientasi objek memiliki beberapa karakteristik utama yaitu:

1. Pewarisan (Inheritance)

   Pewarisan adalah sebuah metodologi yang menyatakan bahwa anak dari sebuah objek akan mewarisi atribut dan perilaku dari induknya. Dengan demikian apapun atribut dan perilaku dari class akan dimiliki semua objek yang diturunkan dari kelas tersebut. Suatu kelas ditentukan secara umum, kemudian untuk subkelas ditentukan secara spesifik. Masing-masing subkelas mewarisi semua sifat yang dimiliki oleh kelas induknya dan bisa ditambahi dengan sifat unik yang dimiliki. Metode pewarisan menggambarkan generalisasi sebuah kelas. Konsep pewarisan dapat dimisalkan truk dan sepeda motor yang merupakan subkelas dari induk kendaraan bermotor, kedua subkelas tersebut memiliki sifat induk yaitu memiliki mesin, dapat berjalan, memiliki roda. Setiap subkelas memiliki sifat yang berbeda misal jenis mesin, kecepatan, dan jumlah roda. Konsep pewarisan dapat digambarkan sebagai berikut:



Gambar 2.1.3 Konsep Pewarisan

2. Pengkapsulan (*Encapsulation*)

   Proses pengkapsulan adalah proses membatasi ruang lingkup program terhadap data yang sedang diproses. Konsep pengkapsulan adalah menyembunyikan informasi. Data dan prosedur dikemas secara bersamaan dalam bungkus suatu objek, sehingga data atau prosedur dari luar tidak dapat mengaksesnya. Contoh di dunia nyata seperti proses ketika kita memperbesar volume televisi, pengguna tidak perlu tahu bagaimana proses volume tersebut bekerja. Pengguna televisi cukup memperhatikan tombol navigasi apa saja yang digunakan untuk mengoperasikan televisi. Pengkapsulan diperlukan untuk melindungi data dari prosedur atau objek lain, sehingga masalah bisa dilokalisisr.

3. Banyak Bentuk (*Polymorphism*)

   Polymorphism adalah konsep yang menyebutkan bahwa sebuah operasi yang sama dapat mempunyai bentuk dan perilaku yang berbeda. Kelas yang berbeda bisa memiliki nama operasi yang sama. Misalnya operasi "membuka" bisa digunakan untuk mebuka jendela, membuka pintu, membuka buku. Operasi "membuka" digunakan pada objek yang berbeda sehingga memiliki makna yang berbeda. Setiap objek memiliki kemampuan yang berbeda-beda untuk melakukan suatu metode dalam merespon pesan yang sama.

4. Asosiasi (*Association*)

   Asosiasi adalah hubungan antar objek yang saling membutuhkan. Sebuah kelas dapat diasosiasikan dengan beberapa kelas sekaligus. Asosiasi membutuhkan hubungan dua arah. Selain sesama objek asosiasi juga bisa diterapkan pada kelas, dimana sebuah kelas bisa berasosiasi dengan lebih

dari satu kelas. Contoh seorang sopir dapat mengendarai bis sekaligus dapat mengendarai mobil, maka bisa dikatakan kelas sopir berasosiasi dengan kelas bis dan kelas mobil.

5. Agregasi (Aggregation)

   Agresasi merupakan bentuk khusus dari asosiasi dan lebih kuat. Agregasi menggambarkan relasi antara satu objek dan objek-objek lainnya sebagai komponen pembentuknya dengan sifat relasi yang kuat. Misalkan objek gabungan komputer tersusun dari bagian CPU, Monitor, keyboard. Apakah bisa disebut komputer tanpa CPU. Komputer memiliki hubungan yang kuat dengan komponen pembentuknya yaitu CPU, monitor dan keyboard.

2.3. ANALISIS BERORIENTASI OBYEK

Pada proses pengembangan sistem perangkat lunak harus dibedakan antara proses analisis dengan proses perancangan. Proses analisis adalah kegiatan untuk menjawab pertanyaan tentang kondisi sistem lama atau sistem saat ini dan menjawab kebutuhan dari sistem baru yang akan dibangun. Hasil akhir dari kegiatan analisis adalah spesifikasi usulan berkaitan dengan kebutuhan sistem baru. Salah satu tugas analis adalah menggali kebutuhan dari sistem yang akan dibangun. Banyak aspek yang harus digali terkait kebutuhan pengguna. Tahap menganalisa adalah mendeskripsikan apa yang bisa dilakukan sistem atas kebutuhan pengguna.

Pada saat menganalisa sangat penting untuk memahami bagaimana cara kerja sistem yang ada saat ini sehingga kekurangan-kekurangan pada sistem lama dapat diperbaiki dengan sistem yang baru. Beberapa manfaat dari proses penggalian data kondisi sistem lama adalah:

1. Kondisi sistem lama bisa menjadi dasar acuan target kinerja untuk proses pembangunan sistem yang baru.
2. Kekurangan di sistem lama harus dihindari pada saat membangun sistem baru.
3. Fungsionalitas sistem lama yang teridentifikasi kemungkinan akan diperlukan di sistem yang baru.
4. Beberapa data dari sistem lama dapat dimigrasikan ke sistem baru.
5. Kondisi sistem lama bisa memberikan informasi situasi organisasi secara umum.

Analisa Berorientasi Objek (OOA) mampu memodelkan suatu masalah dengan mempresentasikan objek, atribut dan operasi sebagai komponen pemodelan utama. Proses OOA dimulai dengan mendokumentasikan fungsionalitas sistem menggunakan *use case*. Diagram *use case* mampu menggambarkan skenario fungsionalitas sistem yang akan dibangun. Langkah selanjutnya adalah pengelompokan objek dan membuat hirarki kelas. Hubungan antar objek dapat menggambarkan bagaimana kelas-kelas tersebut berhubungan satu dengan yang lain serta keseluruhan tingkah laku sistem.

Fungsi dari analisa berorientasi obyek adalah memastikan bahwa perangkat lunak dapat mengerjakan apa yang menjadi kebutuhan pelanggan serta dapat menerapkan prinsip-prinsip dasar berorientasi obyek sehingga perangkat lunak lebih fleksibel (McLaughlin, dkk, 2007)

2.4. PERANCANGAN BERORIENTASI OBJEK

Fase perancangan adalah adalah tahap yang harus dilalui oleh pengembang sistem perangkat lunak. Fase perancangan akan menjawab pertanyaan bagaimana sistem perangkat lunak yang akan dibangun. Ada 2

jenis pendekatan yang diterapakan pada tahap perancangan, yaitu perancangan terstruktur dan perancangan berorientasi objek.

Metode perancangan terstruktur adalah konsep lama yang biasa diterapkan pada teknik perakitan di dunia industri. Pada pengembangan sistem perangkat lunak permasalahan yang kompleks pada organisasi dapat dipecahkan dan hasil dari sistem yang dibangun akan lebih mudah untuk dipeliharan. Salah satu tools yang digunakan dalam perancangan sistem terstruktur adalah menggunakan DFD (*Data Flow Diagram*), Kamus Data, *Entity Relationship Diagram* (ERD). Metodologi yang digunakan pada perancangan terstruktur lebih menekankan pada pemecahan fungsional sistem ke dalam subsistem-subsistem yang lebih kecil dan fokus pada karakteristik data yang akan diproses oleh sistem. Kekurangan dari perancangan terstruktur adalah lebih mengutamakan pada proses sehingga kebutuhan non-fungsional terabaikan.

Pendekatan perancangan berorientasi objek (OOD) akan memandang sistem perangkat lunak yang dikembangkan sebagai suatu kumpulan objek yang berhubungan dengan objek-objek di dunia nyata. Pada proses rekayasa perangkat lunak konsep OOD dapat diterapkan pada fase analisis, perancangan, pemrograman dan pengujian. Relasi objek dengan entitas dapat dipetakan dengan mudah seperti kondisi di dunia nyata sehinga proses desain dapat dengan mudah dipahami.

Metodologi pengembangan dalam perspektif OOD adalah penggunaan kembali (*reuse*) dengan teknik pengkapsulan fungsi dan data secara bersamaaan. *Reuse* merupakan salah satu kelebihan dari OOD, namun apabila dalam penerapannya tidak menggunakan prosedur yang tepat maka konsep ini sangat sulit diterapkan pada skala besar. Pada metodologi OOD

standarisasi objek kemungkinan dapat dibentuk sehingga akan mengurangi resiko proses pembangunan sistem.

Beberapa tools yang digunakan dalam proses perancangan berorientasi objek diantaranya adalah

1. *Object Oriented Analysis* dan *Object Oriented Design* yang dikenal dengan tools OOA dan OOD dari Peter Coad dan Edward Yourdon, dikenalkan pada tahun 1990.
2. *Object Modelling Technique* dikenal dengan OMT dari James Rumbaugh, Michael Blaha, William Premerlan, Frederick Eddy dan William Lorensen pada tahun 1991.
3. *Object Oriented Software Engineering* (OOSE) dari Ivar Jacobson pada tahun 1992.
4. Booch Method dan Grady Booch pada tahun 1994.

   Unified Modeling Language (UML) dari James Rumbaugh, Grady Booch dan Ivan Jacobson pada tahun 1997.

BAB 3 UNIFIED MODELING LANGUAGE (UML)

Tujuan Instruksional:

1. Dapat menjelaskan sejarah Unified Modeling Language (UML).
2. Dapat menyebutkan dan menjelaskan diagram dalam UML

3.1. Sejarah dan Pengertian Dasar UML

Pemodelan atau modeling adalah serangkaian proses merancang sistem perangkat lunak sebelum masuk tahap pembangunan sistem atau *coding*. Mengunakan pemodelan dalam proses merancang sebuah sistem yang kompleks merupakan fase yang sangat penting. Semakin kompleks sistem yang akan dibangun maka semakin penting penggunaan teknik pemodelan yang baik dan tepat.

UML (Unified Modeling Language) adalah alat bantu yang sudah menjadi standar dalam dunia pengembangan sistem perangkat lunak berorientasi objek (Sugiarti, 2013). UML menjadi bahasa yang handal dalam memvisualisasi rancangan sitem perangkat lunak. UML memungkinkan para pengembang sistem membuat *blue print* dalam bentuk yang baku dan mudah dimengerti sehingga bisa hasil rancangan bisa dikomunikasikan dengan pihak lain.

UML merupakan penyatuan dari pemodelan *Booch Method*, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Dimulai pada tahun 1994 tiga pakar metodologi pemodelan berorientasi objek yaitu Booch, Rumbaugh dan Jacobson mempelopori proses penyatuan metodologi.

Pemodelan metode Booch yang dikenal dengan *Design Object Oriented* menggunakan 4 tahapan iterative yaitu identifikasi kelas dan objek,

identifikasi hubungan kelas dan objek, identifikasi antarmuka dan terakhir implementasi. Kelebihan dari metode Booch adalah sangat detail dan banyak notasinya.

Pemodelan OMT yang dipelopori Rumbaugh dikembangkan dengan dasar analisis perancangan terstruktur dan pemodelan entity relationship. Tahapan dalam pemodelan OMT dimulai dengan analisa, perancangan sistem, perancangan objek dan implementasi. Kelebihan dari metode ini adalah mampu mendukung konsep *Object Oriented* dengan sangat baik.

Metode OOSE milik Jacobson memiliki tiga tahapan yaitu membuat analisis kebutuhan, perancangan, implementasi dan pengujian. Metode ini lebih menekankan pada diagram *use case*. Kelbihan dari metode Jacobson adalah notasi yang sederhana namun sudah mencakup keseluruhan tahapan dalam rekayasa perangkat lunak.

Ketiga pemodelan tersebut diatas digabungkan oleh ketiga pakar dengan menambahkan elemen baru yang lebih seragam serta membuang elemen lama yang tidak praktis. Pada tahun 1995 draft pertama dari UML dipublikasikan dengan UML versi 8.0, kemudian sejak tahun 1996 pengembangan UML dibawah koordinasi Object Management Group (OMG).

3.2. Pemanfaatan UML dalam Pengembangan Perangkat Lunak

UML dapat digunakan sebagi pemodelan untuk semua jenis aplikasi perangkat lunak serta dalam bahasa pemrograman apapun, tetapi karena dalam konsep dasarnya UML juga menggunakan kelas dan operasi maka UML lebih cocok digunakan untuk bahasa berorientasi objek seperti C++, Java, C# atau VB.NET.