

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/334562380>

Pemodelan Sistem Menggunakan UML (Unified Modelling Language)

Conference Paper · July 2019

CITATIONS

2

READS

10,692

1 author:



Hendy Algorithm

Universitas Esa Unggul

10 PUBLICATIONS 2 CITATIONS

SEE PROFILE

Pemodelan Sistem Menggunakan UML (Unified Modelling Language)

Hendy

Universitas Esa Unggul
Jl. Arjuna Utara No.9, (021) 5674223
hendyalgorithm@programmer.net

Abstrak – Pengembangan sistem di zaman sekarang, baik sistem dalam bidang teknologi, bisnis, maupun bidang lainnya dapat dikatakan cukup kompleks sehingga membuat banyak pengembang sistem bingung ketika ingin mengembangkan sebuah sistem. Sistem yang dikembangkan oleh para pengembang biasanya mencakup pengembangan menggunakan sebuah atau beberapa bahasa pemrograman yang terkait dengan pengembangan sistem tersebut. Oleh sebab itu, dibutuhkan sebuah peralatan untuk merancang/memodelkan sistem sehingga membuat pengembangan sistem mudah ketika mengimplementasikannya ke sebuah atau beberapa bahasa pemrograman. Perancangan atau pemodelan sistem ini dapat dilakukan menggunakan peralatan (tools) yang disebut sebagai Unified Modelling Language (UML) dimana perancangan menggunakan peralatan ini biasanya dilakukan oleh seorang analis sistem.

Kata Kunci: pengembangan, pemodelan, sistem, UML

1. Pendahuluan

Perkembangan zaman memengaruhi cara seorang pengembang (*developer*) mengembangkan sebuah sistem. Perkembangan zaman mengubah tata cara untuk mengembangkan sistem tersebut sehingga secara langsung dapat mengubah pola pikir dari pengembang sistem untuk mengembangkan sistem tersebut. Perkembangan zaman pasti berdampak kepada jenis-jenis perangkat lunak yang akan dikembangkan oleh seorang pengembang. Dalam kenyataannya perkembangan zaman ini tidak membuat tata cara pengembangan menjadi semakin mudah, tetapi menjadi semakin kompleks (Pengantar *Unified Modeling Language* (UML), Sri Dharwiyanti, Romi Satria Wahono, 1).

Oleh sebab itu, perkembangan zaman ini dapat membuat pengembang sistem atau perangkat lunak menjadi sedikit bingung ketika ingin mengimplementasikan sistemnya secara nyata ke sebuah atau beberapa bahasa pemrograman yang digunakan untuk membuat sistem tersebut. Hal tersebut juga berdampak terhadap kinerja dari analis sistem ketika ingin merancang sistem untuk dapat diterapkan secara langsung oleh pengembang (*programmer*) karena sebelum pengembang dapat melakukan implementasi, diperlukan sebuah rancangan sistem atau perangkat lunak yang rancangannya ini dibuat oleh analis sistem.

UML (*Unified Modelling Language*) merupakan peralatan yang dapat digunakan atau dipilih oleh analis untuk merancang sistem yang nantinya akan diterapkan oleh pengembang. Penggunaan UML ini diperlukan dan merupakan hal yang sangat penting dan fundamental untuk digunakan untuk perancangan karena perancangan atau pemodelan menggunakan UML dapat digunakan untuk berbagai bidang, bahkan untuk merancang atau memodelkan sistem yang bukan dalam ruang lingkup IT. Walaupun penggunaan UML lebih sering terdengar untuk memodelkan sistem dalam ruang lingkup IT, UML juga dapat digunakan untuk memodelkan sistem ketika mempelajari bidang sejarah dan ilmu sosial (*Unified Modelling Language in History and Social Science Education*, Jan Bergandy, 2).

Oleh sebab itulah, penggunaan UML sangat diperlukan sebelum pengembang dapat mengimplementasikan sistemnya ke dalam bahasa pemrograman. Hal ini masuk akal mengingat kebutuhan sistem yang bahkan bukan dalam ruang lingkup IT juga memerlukan pemodelan

menggunakan UML. Agar dapat merancang atau memodelkan sebuah sistem menggunakan UML, kita perlu mencari tahu definisi dari UML jenis-jenis dari UML, dan cara penggunaannya.

2. Metode Penelitian

Metode penelitian merupakan sebuah cara untuk mendapatkan sebuah data yang dapat dijadikan informasi dan disajikan dalam bentuk penelitian. Cara untuk melakukan penelitian ini terdiri dari beragam cara atau metode. Berikut ini merupakan beragam cara atau metode untuk melakukan penelitian berdasarkan masalahnya menurut modul yang ditulis tahun 2010 oleh Prof. Dr. Suryana, M.Si:

- a) Penelitian historis,
- b) Penelitian deskriptif,
- c) Penelitian perkembangan,
- d) Penelitian kasus,
- e) Penelitian korelasional,
- f) Penelitian eksperimen,
- g) Penelitian kausal komparatif
- h) Penelitian tindakan.

Karena pada makalah ini penulis membahas masalah cara untuk memodelkan sistem menggunakan UML, maka penulis akan menggunakan metode penelitian deskriptif. Penelitian deskriptif merupakan sebuah penelitian terstruktur yang berisi penjelasan-penjelasan yang akurat mengenai fakta-fakta yang terdapat di dunia nyata (Metodologi Penelitian, Prof. Dr. Suryana, M.Si, 2010, 18).

Alasan penulis menggunakan metode penelitian ini adalah karena masalah yang dibahas perlu untuk dijelaskan secara lengkap dan sesuai fakta. Penjelasan yang akan dipaparkan oleh penulis bertujuan agar pembaca mengerti cara untuk memodelkan sistem menggunakan UML dan mengerti apa UML itu sendiri. Adapula pembahasan yang akan dijelaskan di makalah ini adalah mengenai pengertian dari UML, jenis-jenis dari UML, dan cara menggunakan UML untuk memodelkan sistem.

3. Pembahasan

UML (*Unified Modelling Language*) adalah sekelompok peralatan yang pada biasanya berupa diagram untuk merancang atau memodelkan bagaimana sistem tersebut bekerja, bagaimana pengguna dapat berinteraksi dengan sistem, bagaimana tata cara kerja dari sistem, dan fitur-fitur yang terdapat di sebuah sistem yang nantinya akan diimplementasikan. UML penting digunakan oleh analis sistem karena berbagai macam manfaat yang beragam yaitu dapat manajemen kompleksitas dari sistem, mendeteksi kesalahan yang mungkin terjadi ketika diimplementasikan, menjelaskan tata kerja dari sistem kepada para pihak yang berkepentingan (*stakeholders*), dsb (*Introduction to UML 2.0*, Terry Quatrani, 6).

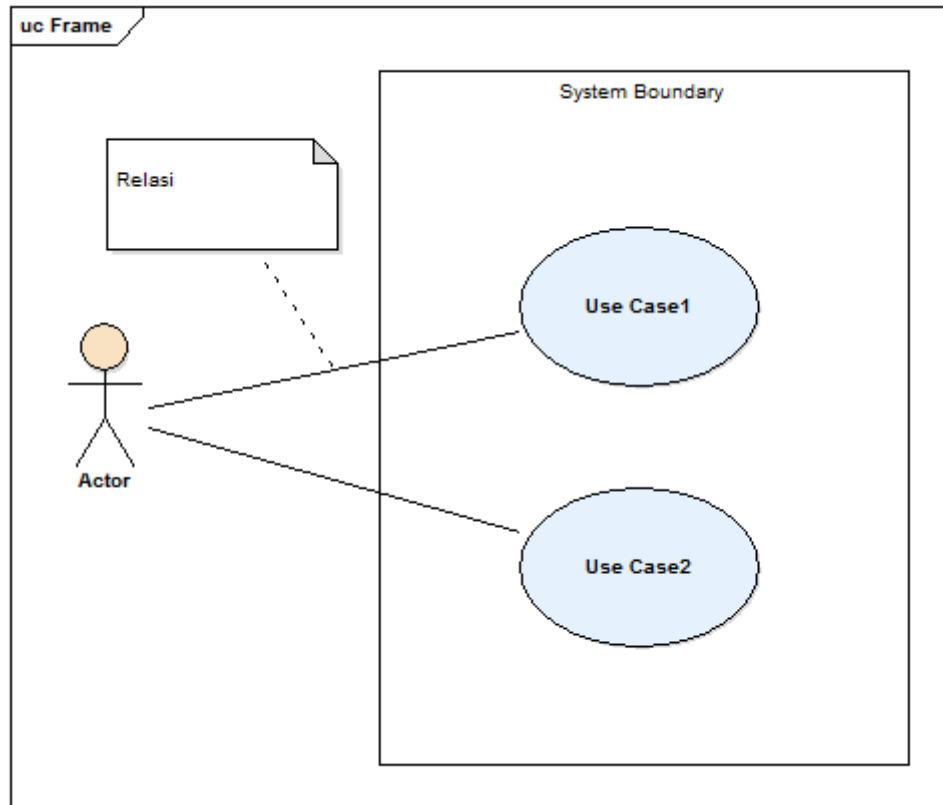
Seperti yang telah dipaparkan diatas UML merupakan peralatan yang terdiri dari beberapa diagram. Pada kesempatan ini, penulis akan membahas cara menggunakan UML untuk memodelkan sistem menggunakan dua diagram UML yang paling umum dan akan dipaparkan di bagian-bagian berikut ini:

3.1 Diagram Use Case

Diagram Use Case merupakan sebuah diagram yang digunakan oleh seorang analis untuk memodelkan kebutuhan atau skenario dari pengguna ketika nantinya pengguna berinteraksi dengan sistem tersebut (Analisis Perancangan Sistem Berorientasi Objek dengan UML, Munawar, 89). Secara umum, diagram ini digunakan untuk mendeskripsikan fitur-fitur apa saja yang dapat digunakan oleh pengguna ketika berinteraksi dengan sistem. Selain untuk memudahkan pengembang untuk mengembangkan sistem atau perangkat lunak, diagram use case

juga dapat digunakan oleh para pemangku kepentingan (*stakeholders*) untuk melihat skenario apa saja yang terdapat di sistem tersebut.

Berikut ini merupakan gambar diagram, definisi, dan deskripsi mengenai cara penggunaan ketika melakukan pemodelan skenario sistem menggunakan diagram use case:



Gambar 1. Model Use Case

Actor adalah sebuah subjek yang merupakan representasi dari pengguna. Skenario sistem yang didefinisikan pada hal tersebut terkait erat dengan *actor*. Skenario-skenario sistem yang terdapat di sistem tersebut didefinisikan dengan cara membuat sebuah border yang disebut sebagai *system boundary*. Skenario-skenario sistem yang terdapat di dalam *system boundary* didefinisikan dalam bentuk diagram yang biasa disebut sebagai use case. Adapula relasi yang menghubungkan *actor* dengan use case. Relasi ini berfungsi untuk mendefinisikan skenario sistem apa saja yang dapat digunakan oleh *actor* yang terkait dengan sistem tersebut.

Sebenarnya ada beberapa relasi umum lainnya yang sering digunakan untuk memodelkan skenario sistem menggunakan diagram ini. Relasi tersebut adalah *extend* dan *include*. Relasi *extend* berfungsi sebagai relasi yang mendefinisikan perluasan fitur atau skenario dari sistem tersebut, sedangkan relasi *include* berfungsi sebagai relasi yang mendefinisikan fitur atau skenario yang diperlukan sebelum menjalankan fitur atau skenario yang berikutnya.

Model use case ini terdapat di dalam sebuah bingkai yang didefinisikan sebagai *frame*. Penamaan *frame* biasanya adalah nama dari sistem yang sedang dirancang skenarionya di dalam model use case tersebut. Perancangan diagram use case ini menggunakan perangkat lunak *Enterprise Architect v12.0*. Perancangan model use case ini dapat dilakukan dengan mudah di perangkat lunak ini, yaitu dengan cara mengklik kanan dan menambahkan elemen-elemen yang akan digunakan untuk merancang skenario sistem (seperti *actor*, use case, relasi).

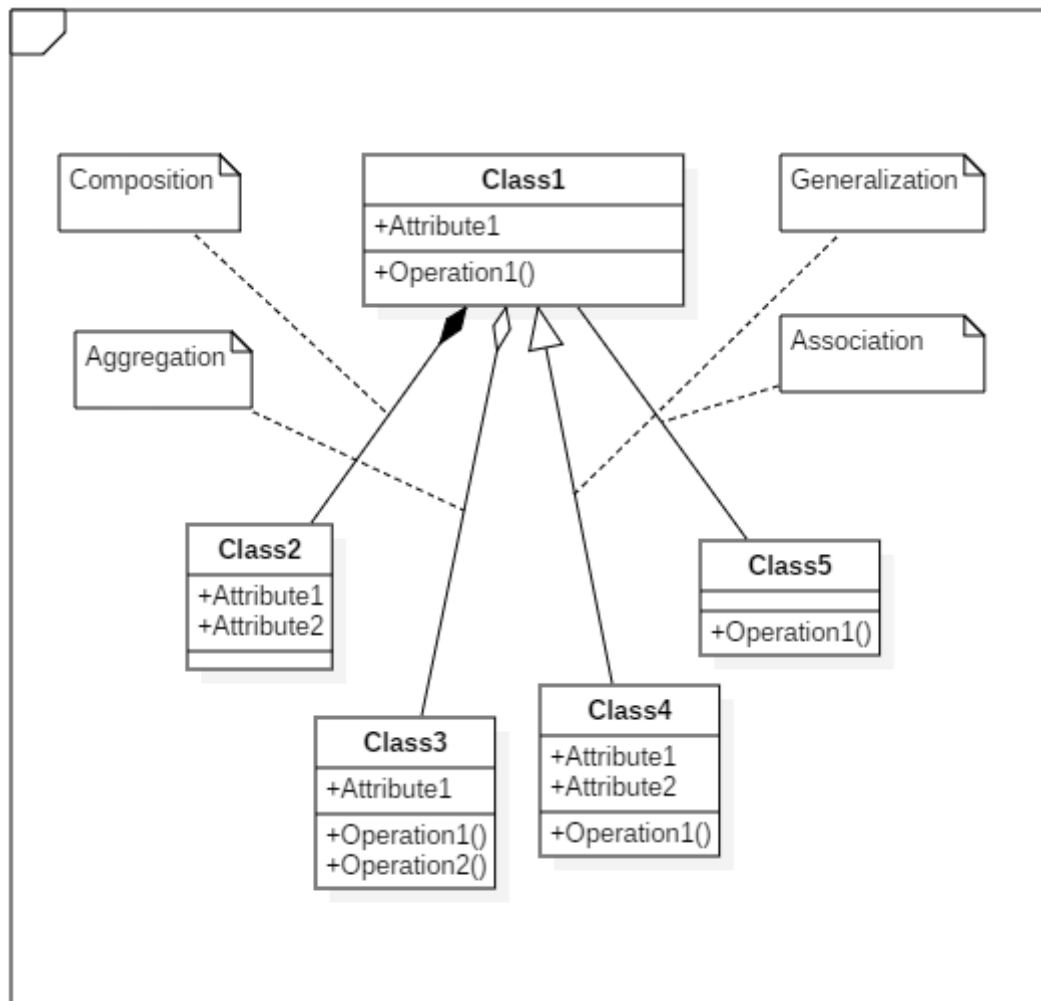
Cara menggunakan perangkat lunak *Enterprise Architect v12.0* adalah dengan cara menambahkan sebuah atau beberapa *actor* yang terkait dengan sistem tersebut dan use case yang merupakan representasi dari skenario-skenario yang terdapat di dalam sistem tersebut.

Kemudian, perancang sistem dapat mengkoneksikan relasi antara *actor* dan use case yang telah dibuat untuk mendefinisikan skenario apa saja yang dapat dilakukan oleh *actor*.

3.2 Diagram Class

Diagram class adalah sebuah diagram yang digunakan untuk merancang bagaimana sistem dapat memiliki variabel atau atribut penyimpan data beserta metode yang mendefinisikan cara data tersebut dikelola di dalam sebuah sistem. Metode ini adalah sebuah proses yang menggambarkan bagaimana sistem tersebut dapat bekerja untuk mengelola data yang ada.

Berikut ini merupakan diagram, definisi, dan cara untuk menggunakan diagram class untuk merancang proses-proses yang terdapat di sebuah sistem:



Gambar 2. Diagram Class

Pada diagram class yang terdapat pada gambar 2, terdapat sebuah diagram yang berbentuk kotak dan memiliki komponen-komponen di dalamnya. Diagram tersebut disebut dengan class. Class merepresentasikan elemen yang terdapat di sistem tersebut. Di dalam class terdapat komponen yang disebut *attribute*, dimana hal ini berfungsi sebagai variabel penyimpan data dari elemen sistem. Ada juga komponen di dalam diagram class yang disebut sebagai *operation* yang berfungsi sebagai fungsi yang mendefinisikan metode atau cara untuk mengelola sebuah data yang terdapat di elemen tersebut atau elemen yang lainnya.

Ada berbagai macam tipe relasi yang dapat digunakan di dalam diagram class, tetapi yang paling umum dan sering digunakan adalah relasi *composition* (relasi yang mendefinisikan bahwa

class tersebut masih merupakan bagian yang terikat dari kelas yang diasosiasikan), relasi *aggregation* (relasi yang mendefinisikan bahwa class tersebut masih merupakan bagian yang tidak terikat dari kelas yang diasosiasikan), relasi *generalization* (relasi yang mendefinisikan bahwa class tersebut merupakan turunan dari kelas yang diasosiasikan), dan relasi *association* (relasi biasa antar class).

Model diagram class yang dirancang di bagian ini merupakan model yang dirancang menggunakan perangkat lunak *StarUML v3.0.2*. Cara menggunakan perangkat lunak ini untuk merancang diagram class sangatlah mudah karena sudah disediakan menu di bagian kiri perangkat lunak untuk menambahkan komponen-komponen yang terdapat di diagram class. Cara lainnya yang dapat digunakan adalah melakukan klik kanan di file yang telah disediakan dan memilih menu 'Add'.

4. Kesimpulan

UML (*Unified Modelling Language*) merupakan sekelompok diagram yang diperlukan untuk merancang sistem atau perangkat lunak sebelum melakukan pengembangan. Terdapat banyak diagram yang diperlukan untuk merancang sebuah sistem, tetapi terdapat dua diagram fundamental yang diperlukan untuk merancang skenario dan pengelolaan data atau proses dari sebuah sistem yang akan dikembangkan. Diagram ini adalah diagram use case dan diagram class.

Perancangan dari kedua diagram ini dapat dilakukan dengan menambahkan komponen-komponen dasar yang terdapat pada diagram ini, kemudian menambahkan relasi antar komponen-komponen dasar yang sudah ditambahkan terlebih dahulu. Beberapa perangkat lunak yang dapat digunakan untuk merancang sistem antara lain adalah *Enterprise Architect* dan *StarUML*.

5. Daftar Pustaka

- [1] Sri Dharwiyanti, Romi Satria Wahono. Pengantar Unified Modeling Language (UML). IlmuKomputer.com. 2003.
- [2] Jan Bergandy. Unified Modelling Language in History and Social Science Education. GSTF Journal on Computing (JoC). 2013; 3(3)
- [3] Prof. Dr. Suryana, M.Si. Metodologi Penelitian. Universitas Pendidikan Indonesia. 2010.
- [4] Terry Quatrani. Introduction to UML 2.0. IBM Corporation. 2005.
- [5] Munawar. Analisis Perancangan Sistem Berorientasi Objek dengan UML. Penerbit Informatika. 2018.