

ANALISIS DAN PEMENUHAN KEBUTUHAN PERANGKAT LUNAK DENGAN METODE KANO MELALUI PENGEMBANGAN BERBASIS KOMPONEN

Made Hanindia Prami Swari¹, I Gede Winaya², Azhari SN³

^{1,2,3}Jurusan IKE Fakultas MIPA, Universitas Gadjah Mada Jogjakarta

E-mail : little_nienz@yahoo.co.id, igedewinaya@gmail.com, arism.softcomp@gmail.com

ABSTRAK

Seringkali terdapat kesulitan membedakan antara kebutuhan dan keinginan pemakai dalam perancangan perangkat lunak. Terkadang keinginan pemakai justru menurunkan kinerja sistem yang akan dibuat atau keinginan yang berlebihan dari pengguna dapat membuat pengerjaan perangkat lunak yang akan dibuat menjadi lebih lama. Terdapat beberapa metode yang dapat digunakan dalam menentukan kebutuhan pengguna, salah satunya dengan menggunakan pendekatan metode Kano. Metode Kano dapat merupakan metode yang digunakan untuk menentukan tingkat kepuasan terhadap suatu atribut (layanan) berdasarkan seberapa baik atribut-atribut tersebut dapat memuaskan kebutuhan pengguna. Pengembangan perangkat lunak berbasis komponen dapat membantu memenuhi kebutuhan pengguna karena memakai komponen perangkat lunak yang sudah ada sehingga pengembang tidak semata-mata terlalu memikirkan kebutuhan dan keinginan pengguna.

Kata Kunci : *Komponen, Kebutuhan, Metode Kano*

1. PENDAHULUAN

Sistem yang baik merupakan sistem yang mampu mengakomodasi kebutuhan dari pemakai. Namun menentukan kebutuhan yang harus dipenuhi oleh suatu sistem tidaklah mudah. Seringkali akan sulit membedakan kebutuhan yang benar-benar harus dipenuhi ataupun memenuhi kebutuhan yang hanya berupa “keinginan” saja. Menurut Sauerwein (1996), metode Kano diperkenalkan untuk menentukan komponen-komponen ataupun layanan yang dapat mempengaruhi kepuasan pengguna. Sehingga metode kano ini dapat digunakan untuk memilah-milah kebutuhan pengguna sehingga sistem yang dibuat akan lebih bermanfaat dan dapat memuaskan keinginan pengguna. Secara umum hal-hal yang penting untuk dilakukan dalam melakukan analisis kebutuhan dengan metode Kano adalah mengerti kecenderungan kesukaan pengguna, menentukan prioritas dari kebutuhan, dan mengklasifikasikan kebutuhan [9].

Pembangunan sistem dengan pendekatan berbasis komponen merupakan suatu pendekatan yang dapat membantu dalam pengembangan sistem. Pembangunan sistem dengan menggunakan metode ini menjanjikan *reusabilitas* dan *maintainabilitas* yang lebih baik disamping waktu yang diperlukan untuk mengembangkan perangkat lunak sejenis berikutnya menjadi lebih efisien [2]. Sebuah arsitektur komponen dapat didefinisikan sebagai “satu set komponen perangkat lunak hubungan structural dan kebergantungan antar komponen” [8]. Pengefisienan waktu yang dihasilkan oleh metode ini tentunya akan membuat pemenuhan kebutuhan pengguna dalam pengembangan suatu sistem dapat lebih cepat, sehingga pengembang dapat memenuhi kebutuhan ataupun keinginan dari pengguna lebih maksimal.

2. METODE KANO

Ada hal yang perlu disadari oleh sebuah perusahaan, bahwa tidak semua atribut layanan yang diberikan dapat memuaskan kebutuhan pelanggan. Nurhayati dalam papernya menyatakan bahwa untuk mengetahui kebutuhan yang benar-benar diperlukan maka harus dilakukan penyelidikan atau analisis terhadap setiap daftar kebutuhan yang dibuat sedetil mungkin. Salah satu metode analisis kebutuhan adalah dengan menggunakan Model Kano.

Model Kano dikembangkan pada tahun 1980 oleh Prof. Noriaki Kano, model ini dikembangkan dengan tujuan memahami hubungan antara pemenuhan (atau tidak) pada kebutuhan (fitur produk) dan kepuasan atau ketidakpuasan yang dialami oleh pengguna layanan. Model Kano dapat mengategorikan atribut-atribut layanan berdasarkan seberapa baik layanan tersebut dapat memuaskan kebutuhan pelanggan. Pengkategorian tersebut dapat berguna untuk panduan keputusan desain baru sebagai solusi atas unsur inovasi. Namun, untuk merealisasikan kebutuhan tersebut harus diperhatikan pula kapasitas dari perusahaan [9]. Adapun dua keuntungan utama mengklasifikasikan kebutuhan pengguna dengan menggunakan metode Kano adalah dapat memprioritaskan pengembangan produk atau layanan, atribut-atribut produk dapat diketahui lebih baik [5]. Metode Kano membedakan atribut dari persyaratan produk kedalam tiga tipe, yaitu [5]:

- Persyaratan yang bersifat *Must-Be* (Harus Ada)

Pada kategori ini, pengguna atau pelanggan menjadi tidak puas apabila kinerja dari atribut yang bersangkutan rendah. Tetapi kepuasan pelanggan tidak akan meningkat jauh diatas netral meskipun kinerja dari atribut tersebut tinggi.

- *One Dimensional* atau *Percormance Needs*
Pada kategori ini, tingkat kepuasan pelanggan linear dengan kinerja atribut, dalam artian semakin tinggi kinerja atribut maka akan mengakibatkan semakin tingginya kepuasan pelanggan begitu pula sebaliknya.
- *Attractive* atau *Excitement Needs*
Sedangkan pada kategori ini, tingkat kepuasan pelanggan akan meningkat tinggi seiring dengan meningkatnya kinerja atribut. Namun penurunan kinerja atribut tidak akan menurunkan tingkat kepuasan. Persyaratan *attractive* tidak diungkapkan secara eksplit dan tidak pula diharapkan oleh pelanggan. Pemenuhan persyaratan ini lebih menekankan pada tingkat kepuasan pelanggan.

2.1 Metode Penelitian

Metode penelitian yang digunakan untuk metode kano adalah dengan menggunakan survey dengan cara mengambil sampel dari suatu populasi dan menggunakan kuisioner. Kuisioner yang disebarakan memiliki bentuk yang khusus. Bentuk ini disesuaikan dengan metode yang diperkenalkan oleh Kano, dimana setiap pertanyaan mengandung komponen pilihan jawaban yang jawabannya antara lain : suka, harus, netral, boleh, atau tidak suka. Setiap pertanyaan ditanyakan dua kali kepada responden, dimana pertanyaan pertama merupakan pertanyaan positif (functional) dan pertanyaan kedua merupakan pertanyaan negative (*dysfunctional*) [12]. Adapun langkah-langkah untuk mengklasifikasikan atribut berdasarkan metode kano adalah sebagai berikut [10] :

1. Menentukan kategori atribut tiap responden berdasarkan tabel evaluasi Metode Kano sebagai berikut :

Tabel 1. Evaluasi Metode Kano

Kebutuhan Konsumen		Dysfunctional				
		1	2	3	4	5
		Suka	Mengharap	Netral	Toleransi	Tidak Suka
Functional	1. Suka	Q	A	A	A	O
	2. Mengharap	R	I	I	I	M
	3. Netral	R	I	I	I	M
	4. Toleransi	R	I	I	I	M
	5. Tidak Suka	R	R	R	R	M

Dari tabel diatas dapat disimpulkan ke dalam beberapa tipe klasifikasi seperti berikut :

- A = Attractive (menarik)
- M = Must be (harus ada)
- O = One Dimensional (Satu dimensi)
- R = Reverse (kebalikan)
- Q = Questionable (diragukan)
- I = Indifferent (biasa saja)

2. Menghitung jumlah masing-masing kategori kano dalam tiap-tiap atribut
3. Menentukan kategori kano tiap-tiap atribut dengan menggunakan Blauth's formula sebagai berikut :
 - Jika jumlah nilai (**one dimensional + attractive + must be**) > (**indifferent + reverse + questionable**) maka grade diperoleh dari nilai terbesar antara one dimensional atau attractive atau must be.
 - Jika jumlah nilai (**one dimensional + attractive + must be**) < (**indifferent + reverse + questionable**) maka grade akan diperoleh dari nilai terbesar antara indifferent atau reverse atau questionable.
 - Jika jumlah nilai (**one dimensional + attractive + must be**) = (**indifferent + reverse + questionable**) maka grade akan diperoleh dari nilai yang paling maksimum diantara semua kategori kano yaitu one dimensional, attractive, must be, indifferent, reverse, atau questionable.

3. PENGEMBANGAN PERANGKAT LUNAK BERBASIS KOMPONEN

Metode pengembangan perangkat lunak berbasis komponen merupakan suatu metode yang menarik untuk dikaji. Dikatakan menarik karena tujuan dari metode pengembangan ini adalah untuk mengurangi waktu dan usaha yang

diperlukan untuk pengembangan suatu perangkat lunak. Hal ini disebabkan karena suatu komponen dimungkinkan untuk melakukan komunikasi dengan komponen lainnya sehingga dapat digunakan dalam banyak dan beragam aplikasi [11].

3.1 Definisi Komponen

Putra (2005) dalam papernya menyatakan bahwa komponen merupakan entitas yang dapat digunakan oleh beberapa program yang berbeda. Komponen menyediakan model standar untuk pemaketan layanan-layanan. Pada sebuah aplikasi yang menggunakan komponen, komponen hanyalah sebuah kotak hitam, karena semua data dan implementasi detail yang dimiliki oleh sebuah komponen akan disembunyikan. Layanan dari sebuah komponen dibuka melalui *public interfaces* [11].

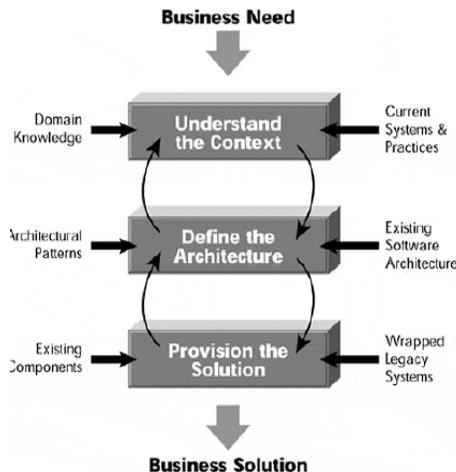
Menurut Sommerville (2004) Pengembangan Berbasis Komponen (CBD) didefinisikan sebagai proses penembangan sistem yang dilakukan dengan mengintegrasikan komponen (koleksi objek) yang sesuai dengan model komponen standar. Suatu komponen dianggap sebagai *language-neutral*, artinya bahwa sejumlah komponen dapat ditulis dalam bahasa yang berbeda. Hal ini berarti bahwa meskipun komponen-komponen tersebut dirancang dan disebarluaskan dalam bahasa yang berbeda namun mereka dapat bekerja sama [4].

Komponen dapat diimplementasikan terbebas dari komponen lainnya. Hal ini dimungkinkan sebab suatu komponen terenkapsulasi, dimana masing-masing komponen memiliki unit kecil dari pengembangan dan pengujian. Suatu komponen tidak dibatasi untuk dapat dijalankan pada platform tunggal, karena dimungkinkan untuk membuat *deployment* yang berbeda untuk suatu komponen agar dapat dijalankan pada beberapa platform [11].

Dalam papernya, Putra (2005) menyatakan bahwa komponen harus memiliki mekanisme yang memungkinkan untuk berintegrasi dengan sistem tanpa memodifikasi dan mengembangkan ulang sistem. Untuk membangun komponen-komponen perangkat lunak yang bisa digunakan berulang-ulang (*reuse*) dibutuhkan waktu dan usaha yang lebih besar dibandingkan dengan membangun komponen yang digunakan untuk memenuhi kebutuhan yang spesifik. Hal ini disebabkan komponen-komponen tersebut harus dapat digunakan untuk kasus yang cukup bervariasi.

3.2 Pendekatan Umum Pengembangan Sistem Berbasis Komponen

Dibutuhkan tiga langkah utama untuk menterjemahkan kebutuhan bisnis menjadi solusi bisnis, yaitu : memahami konteks, mendefinisikan arsitektur, dan menentukan solusi. Berikut ini merupakan diagram yang menggambarkan pendekatan umum dalam pengembangan sistem berbasis komponen



Gambar 1 : Pendekatan Pengembangan Sistem Berbasis Komponen [3]

Adapun penjelasan proses pengembangan komponen melalui pendekatan diatas adalah sebagai berikut :

- Memahami Konteks
Pada tahap ini, pengembang harus memahami perilaku statis dan dinamis (*static and dynamic behavior*). Pada pendekatan UML, kedua perilaku ini dapat dimodelkan melalui kombinasi dari model *class*, *use-case*, dan sekuensial (atau *collaborative*).

- Mendefinisikan Arsitektur
Setelah aspek statis dan dinamis dari sebuah domain dimodelkan, maka pengembang harus memutuskan bagaimana perilaku tersebut harus dikemas dalam unit sehingga dapat dikembangkan secara independen, dibagi kedalam seluruh proyek, dan dijalankan pada mesin yang berbeda.
- Menentukan Solusi
Sistem yang telah selesai (*final system*) akan diimplementasikan melalui desain dan implementasi dari masing-masing komponen penyusunnya. Hal ini akan membutuhkan desain tambahan untuk mengimplementasikan masing-masing komponen tersebut. Hal ini akan menghasilkan sebuah arsitektur implementasi yang menggambarkan komponen fisik dan realisasinya dalam teknologi tertentu. Terdapat banyak teknologi potensial yang dapat dipilih dalam pengimplementasian komponen ini. Terkadang pilihan dari implementasi teknologi sudah dibuat sebelumnya (dengan menggunakan kembali komponen yang telah ada).

Untuk mempermudah pengembangan sistem berbasis komponen diperlukan suatu pedoman untuk merepresentasikan karakteristik yang diperlukan untuk membuat komponen yang berpotensi untuk digunakan kembali [4].

3.3 Kelebihan Penggunaan Kembali Komponen (*Reuse*)

Nugraheni (2010) dalam papernya menyatakan software *reuse* merupakan suatu proses untuk membangun sistem dari sistem lainnya bukan membuatnya dari awal. Atau dapat dikatakan bahwa software reuse adalah proses mengimplementasikan atau mengupdate sistem perangkat lunak dengan menggunakan software yang telah dibuat sebelumnya.

Adapun kelebihan dari Penggunaan Kembali Perangkat Lunak adalah [4] :

- Meningkatkan efisiensi. Hal ini berarti bahwa dengan menggunakan komponen maka dapat mengurangi waktu untuk desain dan implementasi (*coding*).
- Pengembangan berbasis komponen dapat menguntungkan karena komponen-komponen yang telah dibuat dapat diperjual belikan.
- Penggunaan ulang komponen dapat lebih menjamin komponen yang digunakan karena komponen-komponen tersebut tentunya sudah berulang kali diuji sebelumnya.
- Komponen dapat dibangun, dites dan dikembangkan secara individual. Hal ini dapat mempermudah dalam pengembangan dan meningkatkan kualitas serta efisiensi.
- Pemakaian tampilan antarmuka standar dapat meningkatkan kehandalan, karena pengguna lebih terbiasa menggunakan antarmuka yang telah dikenal.

3.4 Masalah Pengembangan Perangkat Lunak Berbasis Komponen

Selain memiliki keuntungan, pengembangan perangkat lunak berbasis komponen ini juga memiliki beberapa masalah dalam proses pengembangan perangkat lunak, diantaranya [1] :

- Cara menemukan komponen yang sesuai dengan kebutuhan
- Jaminan komponen berfungsi dengan baik (tersertifikasi)
- Standarisasi untuk sertifikasi komponen dan pengujian kebenaran dari komposisi komponen belum ditetapkan secara pasti.
- Teknik dalam mengintegrasikan komponen-komponen yang ada menjadi satu sistem yang utuh
- Pemeliharaan komponen

4 PERANAN METODE PENGEMBANGAN PERANGKAT LUNAK BERBASIS KOMPONEN DALAM MEMBANTU MEMENUHI KEBUTUHAN PENGGUNA

Pengembangan sistem berbasis komponen difokuskan pada mengidentifikasi entitas yang dapat digunakan kembali (*reuse*) dan hubungan antara komponen-komponen tersebut. Hal ini dimulai dari penentuan kebutuhan sistem dan mendefinisikan komponen-komponen yang telah tersedia [6]. Menurut Putra (2005) dalam papernya menyatakan meskipun sebagian besar komponen dibuat untuk memenuhi kebutuhan dari suatu aplikasi, namun setiap kali suatu komponen dibuat dan disebar, maka sangatlah mungkin untuk menggunakan komponen tersebut pada aplikasi yang berbeda, selama *interface* dari komponen tersebut memenuhi kebutuhan pengguna. Berdasarkan hal tersebut maka penggunaan komponen-komponen dalam pembuatan suatu sistem yang utuh dapat mengefisiensi waktu dalam desain, testing, dan deployment. Melalui penggunaan komponen-komponen yang telah ada, maka pengembang hanya tinggal mengintegrasikan komponen yang satu dengan komponen yang lainnya untuk membuat sebuah sistem yang utuh, tanpa harus membuatnya dari awal. Komponen-komponen yang ada kemungkinan besar sudah di test oleh beberapa pengguna sebelumnya, sehingga tingkat kebenarannya menjadi lebih baik. Masalah yang mungkin muncul adalah bagaimana memilih komponen yang dapat dipakai dalam sistem

yang dibuat. Masalah lain yang mungkin muncul adalah ketidakpercayaan *programmer* terhadap komponen yang tidak dibuat oleh dirinya sendiri, dan terkadang akan lebih sulit mengerti potongan kode yang dibuat oleh orang lain dibandingkan dengan membuat komponen dari awal. Namun hal ini dapat ditanggulangi dengan membuat dokumentasi program selengkap-lengkapnyanya sehingga pengembang selanjutnya dapat lebih mudah untuk menentukan komponen yang akan dipilih serta mengerti akan potongan kode dari masing-masing komponen tersebut.

Melalui efisiensi waktu yang tercipta dengan metode pengembangan perangkat lunak berbasis komponen ini, maka penggunaan metode ini akan sangat membantu pemenuhan kebutuhan-kebutuhan pengguna dalam pembuatan suatu sistem. Jika sebelumnya pengembang hanya akan membuat sistem yang memenuhi kebutuhan pengguna yang termasuk dalam kategori *must-be* dikarenakan keterbatasan waktu dan biaya, maka dengan metode ini pengembang tentunya dapat lebih banyak memenuhi kebutuhan pengguna. Sehingga nantinya pengembang tidak semata-mata harus memikirkan kebutuhan yang harus dipenuhi dengan jangka waktu tertentu. Waktu dan biaya yang lebih efisien dihabiskan melalui penggunaan metode ini tentunya dapat memperpanjang daftar kebutuhan yang dapat dipenuhi.

5 KESIMPULAN

Proses analisis kebutuhan perangkat lunak merupakan hal yang penting dalam proses pembuatan perangkat lunak. Salah satu metode yang dapat dipergunakan untuk analisis kebutuhan adalah dengan menggunakan metode kano. Melalui penggunaan metode ini, pengembang dapat mengetahui kebutuhan mana yang benar-benar harus dipenuhi dan kebutuhan mana yang tidak berpengaruh signifikan pada sistem. Selain itu, dalam pengembangan perangkat lunak berbasis komponen diperlukan penggunaan standarisasi dalam pengembangan komponen agar komponen yang satu dapat bekerja sama dengan komponen yang lainnya.

DAFTAR PUSTAKA

- [1] A. Santoso, D. Cahyadi, R. Lokasasmita, "Verifikasi dan Sertifikasi Perangkat Lunak Berbasis Komponen" dalam Seminar Nasional Aplikasi Teknologi Informasi 2007 (SNATI 2007) ISSN : 1907-5022. 2007
- [2] A. Y. Ridwan, "Pembangunan Perangkat Lunak Berbasis Komponen dengan Arsitektur 3-Tier (Contoh Kasus Sistem Informasi Rumah Sakit X)" pada Prosiding Konferensi Nasional Teknologi Informasi & Komunikasi untuk Indonesia Institut Teknologi Bandung. 2006
- [3] A. Brown, *Large-scale Componentbased Development*. Prentice Hall, 2000
diunduh dari : <http://armstrong.craig.free.fr/eBooks/Prentice%20Hall/Prentice%20Hall%20Large-Scale%20Component-Based%20Development.pdf>
- [4] D. M. K. Nugraheni, "Software Component Reuse and OOAD (Object Oriented Analysis and Design)" pada Seminar Nasional Ilmu Komputer Universitas Diponegoro, 2010
- [5] E. Sauerwein, F. Bailom, K. Matzler, H.H Heinterhuber, "The KANO Model : How to Delight Your Customer", pada International Working Seminar on Production Economic, Volume 1, hal 313-327. 1996
- [6] I. Crnkovic, S. Larsson, M. Chaudron. "Component-based Development Process and Component Lifecycle". Unpublished
- [7] I. Sommerville. 2004. *Software Engineering*, 7th Edition, ch:18. Massachusetts : Addison Wesley
- [8] J. Bertin, "*Best Practices for Component-Based Development and Model -Driven Architecture*". Softmetaware. 2003. Unpublished
- [9] Q. Xu, R. J. Jiao, X. Yang, M. Helander, H. M. Khalid. 2008. "*An Analytical Kano Model for Costumer Need Analysis*". Elsevier LTD
- [10] R. A. Anggoro. 2008. "Evaluasi Kualitas Pelayanan Pendidikan dengan Menggunakan Metode Kano (Studi Kasus di Pusat Pendidikan ISTIBANK Pabelan, Kartasura)". (Unpublished Undergraduated Thesis). Universitas Muhammadiyah, Surakarta
- [11] S. I. Putra, "Pembangunan Perangkat Lunak Berbasis Komponen Sudi Kasus : Sistem Informasi Akademik Terdistribusi" dalam Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005) ISBN : 979-756-061-6. 2005
- [12] S. Nurhayati, "Analisis Kebutuhan Proses Bisnis Menggunakan Metode Kano". Unpublished