

## 2. TINJAUAN PUSTAKA

### 2.1. Parametric Design and Modeling

*Parametric modeling* mempunyai arti yang sama dengan *parametric variational*, yang berarti constraint-based atau feature based, dimana mempunyai pengetahuan dasar pada pemodelan menggunakan parameter-parameter dan batasan-batasan (*constraints*) untuk menciptakan hubungan secara geometris dan topologi pada sebuah model. Kemampuan *parametric* dari suatu model adalah kemampuan dari geometris suatu objek untuk dikendalikan oleh parameter-parameter (Anderl, 1994). Pada pemodelan *parametrik*, suatu geometri dan dimensi yang mempunyai hubungan dua arah (*bidirectional*) dimana jika mengubah dimensi-nya akan memberi efek pada geometri juga (Anderl and Mendgen, 1995)

*Model parametric* dikendalikan oleh batasan-batasan dan *parameter-parameter* yang dapat dikategorikan sebagai berikut:

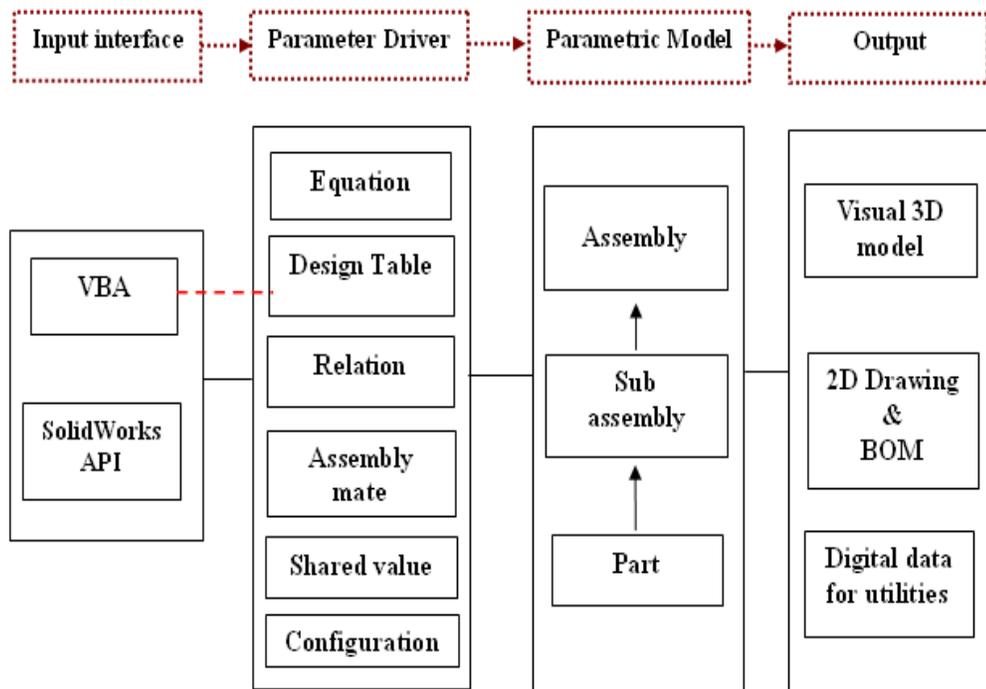
- *Numerical parameter*  
Berbentuk dalam dimensi dengan variabel dan besaran skalar. Seperti jarak dari titik ke titik, diameter dari lingkaran dan lain-lain. Bila kita mengubah besaran ini maka geometri dari bentuk akan ikut berubah.
- *Geometric parameter*  
Ini dapat dalam bentuk dari titik, garis, permukaan dan lain-lain. Sebagai contoh, dalam membentuk fitur *sweep*, maka profil garis dibutuhkan untuk membentuk alur dari *sweep* itu sendiri.
- *Geometric Constraint*  
Menggambarkan hubungan dari geometri dan bentuk yang mana kelihatan pada layar pemodelan. Seperti parallelism, tangency, dan lain-lain
- *Engineering Constraint*  
Bentuk dari persamaan dengan menggunakan rumus.

Pemodelan *parametric* mempunyai keuntungan, antara lain:

- Dapat melakukan perubahan pada desain parameter manapun kapanpun pada evolusi dari desain model.

- Membuat model fleksibel pada langkah awal pengembangan produk.
- Mudah menciptakan suatu keluarga dari komponen-komponen yang dapat menghemat waktu dan biaya.
- Mengurangi kebutuhan untuk menyimpan data-data dari kumpulan komponen-komponen.
- Memperluas performa untuk analisa toleransi, simulasi dan kinematik dari suatu model.

Pada proses desain membuat produk dengan 3-D parametrik, dapat dibagi menjadi 4 bagian seperti bagan dibawah ini,



Gambar 2.1. Alur pemodelan *parametric*

Sumber: Ongkodjojo & Gunawan(2005, p.5)

### Input Interface

*Input interface* adalah gerbang untuk mengakses parameter dari sebuah model. *Input interface* adalah tempat dimana pengguna memasukan besaran-besaran (*values*) yang akan membuat modifikasi atau perubahan dari sebuah model. Sesuai dengan hubungannya dengan beberapa parameter, metode ini dapat dibagi menjadi dua, yaitu:

- *VBA interface*  
VBA (*Visual Basic for Application*) dapat digunakan sebagai alat untuk menciptakan hubungan dengan sebuah hubungan langsung dengan tabel desain sebagai *parameter driver*. Dengan bantuan dari VBE (*Visual Basic Editor*), macro dapat diciptakan dengan membangun program yang mengeset sebuah instruksi untuk memodifikasi beberapa sel tertentu di dalam tabel desain dimana bersesuaian dengan parameter dari model. VBA juga dapat membuat form yang akan memfasilitasi pengguna untuk memodifikasi atau *filling-in values*.
- *SolidWorks API*  
API (*Application Programming Interface*) adalah alat penghubung untuk mengakses semua fungsi dari *software* SolidWorks (SolidWorks 2004 *Online User's Guide*). Ini dapat dijalankan dengan berbagai bahasa program seperti VB (*Visual Basic*), VBA (*Visual Basic for Applications*), VB.NET, C++, atau SolidWorks *macro files*. Akses tak terbatas dan dapat mengatur setiap fungsi, *feature*, dan operasi dalam pada *software* termasuk mengatur parameter dari suatu model.

### **Parameter Driver**

*Items* yang dapat dikategorikan sebagai *parameter driver* adalah fungsi dari *software* yang mengeset beberapa peraturan untuk membuat *parametric* model. Ini dimana semua hubungan antara *values*, parameter-parameter, dan *entities* dari model dapat dibuat dan direncanakan. Inputan untuk driver-driver ini adalah *values* atau definisi dimana terhubung pada *output property* dari model.

### **Parametric Model**

Sebagai pusat aktivitas dari pemodelan, sebuah model dapat berisi dengan *features*, *entities*, *dimensions*, *properties*, *sub-assemblies*, atau *parts*. Setiap dari bagian-bagian itu dimana telah dilakukan pada model yang unik dan mempunyai ciri khas. Ciri khas ini adalah akses untuk mengatur sebuah model secara *parametric*, yang akan menjadikan itu ke dalam *parametric driver* yang akan

mengijinkan untuk diatur dengan mengubah value atau definisi pada tahap input *interface* atau langsung pada *parameter driver*.

## **Output**

Keluaran dari aktivitas pemodelan *parametric* dapat diklasifikasikan menjadi 3 tipe seperti yang dapat dilihat pada gambar 2.1. Visual 3-D model dapat diperoleh secara otomatis saat dimana model sedang dibuat. Perubahan-perubahan pada parameter-parameter akan menghasilkan pada visual geometri bila sudah di-*update*. Penggambaran 2-D bersama dengan BOM (*Bill of Material*) dapat dihasilkan dari model 3-D yang telah diciptakan sebelumnya. Keluaran-keluaran ini pada bagian penggambaran adalah sepenuhnya *parametric* pada 3-D model, dimana modifikasi telah dibuat pada model yang juga telah di-*update*. Keluaran-keluaran lain dapat diklasifikasikan sebagai data digital untuk kebutuhan, yang dapat dihubungkan dengan aplikasi lain yang *me-support* suatu data, sebagai contoh membuat *prototype* pada mesin RP (*Rapid Prototyping*), hubungan pada mesin CNC. *e-drawing*, dan lain-lain.

Komponen dirakit bersama-sama, batasan dan parameter diperlukan untuk mengatur tujuan dari desain untuk mengatur keseluruhan sistem secara serempak. Membangun parameter dapat dilakukan dengan metode berikut ini,

- *Assembly mates*

Sebagai komponen yang di *assembly*, *assembly mates* digunakan untuk menyatakan posisi dari setiap komponen pada komponen yang lain pada *assembly*.

- *Shared Values*

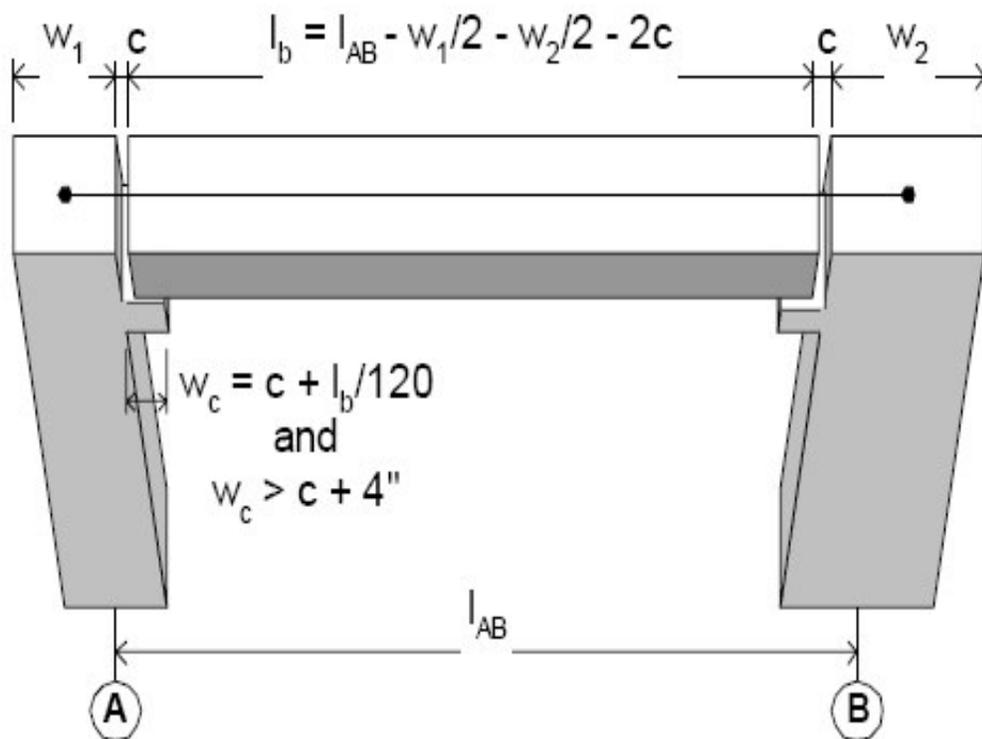
Cara paling mudah untuk menciptakan hubungan antara 2 atau lebih dengan menggunakan nilai-nilai yang menghubungkan dengan yang lain.

- *Equation*

*Equation* digunakan untuk menyesuaikan dari *parameter* yang ada, sebab beberapa rumus matematika digunakan untuk menghubungkan satu dengan lainnya.

Dalam penggantian tradisional menggambar 2-D ke pemodelan 3-D dengan menggunakan metode *top-down parametric* dalam merencanakan

bangunan, mereka percaya bahwa penggantian cara ini dapat mendekati desain yang mempunyai *error* hampir tidak ada dan juga peningkatan pada kualitas, biaya dan juga jangka waktu pengerjaan *project*. Pada perakitan *parametric assembly* dan pemodelan 3-D, terdapat 2 prinsip untuk memodelkan dalam *software* yang terpusat untuk mengurangi *error* dan konsekuensi pengerjaan ulang, salah satunya adalah memperbaiki integritas dari *assembly* ke bagian-bagian daripada dari bagian-bagian ke *assembly*. Dalam memodelkan suatu bangunan yang terdiri dari beberapa bagian dengan geometris tetap, geometris dari setiap bagian harus berhubungan dengan bagian yang lain. Sebagai contoh (lihat gambar 2.2.)



Gambar 2.2. Model *parametric* dari sebuah balok di antara 2 kolom

Sumber: Sacks, Eastman and Lee (2003, p.46-55)

Balok akan secara otomatis menge-pas-kan jarak (*constraint*) antara 2 kolom, dan penyangga juga otomatis menge-pas-kan dimensi (*constraint*) dari balok. Jadi bila salah satu dimensi berubah, maka yang lain akan berubah, karena masing-masing telah terhubung oleh suatu rumus. Inilah yang dimaksud dengan pemodelan

*parametric top-down*. Desainer akan terfokus pada struktur keseluruhan *assembly* baru ke bagian-bagian yang menyusun *assembly*. Tidak sama dengan CAD file tradisional, sifat tiap part pada bangunan *top-down parametric* model lekat mencerminkan konsep berpikir dari suatu insinyur di dalam meletakkan bersama-sama suatu disain bangunan berbeton. Insinyur pertama terfokus pada struktur secara keseluruhan perakitan, berikutnya dengan bagian-bagian yang membuat dari *assembly*, kemudian koneksi yang menghubungkan dan akhirnya dengan detail dari tiap part-nya. Bila perubahan dilakukan pada tingkat manapun, unsur-unsur pada suatu tingkat yang lebih rendah perlu menyesuaikan pada perubahan yang dibuat. Potensi untuk melakukan penyesuaian otomatis seperti ini lebih besar terjadi pada tingkat yang lebih rendah.

Bisa jadi perdebatan bahwa membangun *modeling* dengan metode *top-down* bisa dilakukan menggunakan sistem CAD konvensional, bahkan di 2-D, dengan canggih dan mengatur penggunaan *layer-layer* gambar dan pandangan separasi model. Bagaimanapun, pada umumnya sistem, potongan individu dimasukkan dalam *assembly* sebagai potongan rangkaian produksi (*piece-marks*). Bila memakai metode *bottom-up*, maka bila terdapat perubahan dalam salah satu bagiannya, yang dilakukan harus membuat bagian itu dengan ukuran yang baru. Ini sangat berbeda bila menggunakan metode *parametric top-down* dimana semua geometri-nya berhubungan.

Penulis menemukan bahwa dengan metode ini ditemukan bahwa mereduksi sangat banyak waktu dalam membuat *assembly* dan bagian-bagiannya.

Model dari suatu kumpulan komponen terintegrasi pada pemodelan 3-D dan sistem dengan kaya pengetahuan dapat berperan untuk menghapuskan atau mengurangi kesalahan, antara lain:

1. Hubungan yang logis antara sambungan dan bagian yang melekat pada sistem. Ini bersama-sama dengan perilaku *parametric* dari *assembly* dan bagian yang berarti integritas dari model 3-D dapat berubah tanpa kebutuhan untuk melakukan intervensi pada part dari pengguna.
2. Model 3-D menjadi satu-satunya sumber untuk semua informasi produk, gambar 2-D adalah gambar yang dihasilkan dari informasi model 3-D model. Ke-tidak-konsistenan yang muncul antara berbagai tempat penyimpanan (

seperti pada paradigma gambar 2-D ) tidak bisa terjadi. Kesalahan koordinasi antar gambar *assembly* dan *shop tickets* sangat utama dihapuskan.

3. Detail yang diotomatiskan, seperti penempatan penghubung perangkat keras dan membuat semua geometri yang perlu menyesuaikan kepada bagian yang dihubungkan, ini dapat membantu untuk menghilangkan kesempatan untuk terjadinya kesalahan manusia. Bahkan di situasi disain unik, dimana detail diotomatiskan, tidak bisa diterapkan dan pendetailan harus dilakukan dengan manual, pendetailan dapat diselesaikan pada konteks semua bagian lain, maka kesempatan untuk kesalahan dapat benar-benar dikurangi.
4. Sistem pembangunan manapun yang berdampak pada bagian yang berbeton dapat diimport atau secara langsung dimodelkan dalam 3-D (yang memungkinkan menuntut signifikan pekerjaan tambahan). Jika ini diperbaharui dari waktu ke waktu untuk mencerminkan semua perubahan, maka semua bagian gambar diproduksi dapat benar-benar menunjukkan lubang manapun yang diperlukan, dan prosedur pendetailan otomatis manapun dapat meliputi sistem komponen bangunan. Suatu manfaat yang didapat adalah waktu penggambaran suatu komponen dapat dikurangi dari beberapa bulan menjadi beberapa hari.
5. Pemodelan bangunan 3-D menyediakan suatu *platform* untuk didesain secara otomatis melakukan pengecekan rutin. Sebagai contoh, bagian tanpa koneksi cukup, konflik mengenai ruang, dan kesalahan lain dapat secara otomatis dikenali dan dilaporkan kepada pengguna.

Pemodelan bangunan di dalam komputer, sebagai ganti penggambaran yang mewakili dan komponen itu di dalam penggambaran (baik itu menggunakan CAD atau secara manual), memegang potensi untuk mengurangi terjadinya kesalahan dan kebutuhan untuk mengerjakan lagi pada proyek konstruksi. Menggunakan CAD 3-D perlu, tetapi tidak cukup, model bangunan harus dikembangkan di dalam suatu bentuk terintegrasi, harus menyeluruh, selengkap mungkin, dan harus mengendalikan produksi dari semua penggambaran dan laporan, jika keuntungan diwujudkan nyata (Sacks, Eastman, & Lee, 2003)

## 2.2. Assembly Modeling

Pada pemodelan suatu *assembly*, suatu model dibuat untuk mewakili suatu produk yang terdiri atas beberapa komponen yang lebih kecil. Oleh karena komponen lebih kecil ini, fokus di dalam pemodelan *assembly* tidak hanya pada komponen-komponen tersebut, tetapi juga pada hubungan antar komponen-komponen tersebut. Suatu komponen yang tidak bisa dibagi lagi ke dalam komponen lebih kecil adalah *single part* atau bagian tunggal. Suatu kelompok yang menggabungkan komponen bersama-sama disebut *assembly* atau suatu perakitan.

Terdapat dua pendekatan yang utama dalam memodelkan sebuah produk, yaitu pendekatan *top-down* dan pendekatan *bottom-up*, seperti yang dijelaskan (Libardi et al., 1988 dan Lim et al., 1995).

### 2.2.1. Top-Down Approach

Metode *top-down* adalah metode mendesain sesuatu yang dimulai dengan *complete item* kemudian dibagi-bagi menjadi komponen-komponen yang lebih kecil. Pada metode *top-down*, desain dilakukan lebih *natural* (alami). Pendekatan *top-down* berdasarkan pada pandangan dari seorang desainer. Desainer pada awalnya memulai dari berpikir secara abstrak, berpikir secara fungsional dalam mendapatkan kepuasan pada kebutuhan suatu produk. Produk harus memenuhi beberapa fungsi utama yang ditetapkan. Dengan berulang kali membagi fungsi yang utama ke dalam *sub-functions*, perancang menghasilkan suatu produk dengan metode *top-down* desain.

Dibawah ini merupakan langkah-langkah dalam membagi ke dalam sub-assemblies, yaitu:

1. *Plan*

Berpikir matang-matang dan membuat rencana pada suatu *assembly*.

2. *Define what is known*

Mencari requirements atau kebutuhan-kebutuhan tertentu yang diketahui sebelum memulai.

3. *Create a hand sketch of the assembly skeleton*

Sebelum memulai pemodelan assembly, buatlah sketsa simpel dengan tangan. Uraikan secara singkat tinjauan ide-ide suatu desain dan fungsi-fungsi yang harus dimasukkan di dalam desain. Catat bagaimana sistem seharusnya pindah.

4. *Create the structure*

Gambarkan *sub-assemblies* yang utama dan komponen kerangka.

5. *Review and capture the design intent*

Lengkapi dan uji *assembly* dari kerangka. Pastikan fitur-fitur diberi nama dan pergerakan *assembly* bekerja sesuai yang diharapkan.

6. *Create and mate the sub-assemblies and assembly components*

Buatlah komponen *sub-assembly* dengan menggunakan kerangka untuk menggambarkan dan menghubungkan antar komponen.

### 2.2.2. Bottom-Up Approach

Pada metode *bottom-up*, komponen-komponen *assembly* dimodelkan sebagai bagian yang terpisah terlebih dahulu tanpa ada referensi dari komponen *assembly* yang lain, kemudian menghubungkan (*mate*) satu komponen dengan yang lain. Diperlukan desain detail dari semua bagian dan *sub-assemblies* sebelum membuat desain untuk *assembly*. Untuk membuat perubahan, dengan cara membuka part tertentu, meng-*update*-nya dan lihat bagaimana perakitan berubah. Ini dapat sangat memakan waktu jika kita harus kembali dan membuat perubahan itu. Metode *bottom-up* didasarkan pada ketersediaan teknologi. Metode ini kini sudah secara luas digunakan dikarenakan kemungkinan teknis pada sistem CAD. Pada dekade terakhir, telah ada banyak riset di dalam single-part modeling dan menghasilkan banyak kemajuan. Riset ini dipusatkan pada bagaimana cara memodelkan geometris pada komponen yang mempunyai detail terperinci. Teknik pemodelan seperti *Constructive Solid Geometry* (CSG), *Boundary Representations* (B-Rep) atau kombinasi dari kedua metode ini, dapat digunakan.