

DAFTAR TABEL

Tabel.2.1.	Posisi Penelitian	8
Tabel 3.1.	Perbandingan sistem <i>slot</i> dan <i>hole</i>	10
Tabel 3.2.	Simbol pada Diagram Kelas	15
Tabel 3.3.	Simbol-simbol dalam <i>use case diagram</i>	19
Tabel 4.1.	Data Ukuran <i>Baseplate</i> untuk Pemilihan dari <i>Database</i>	27
Tabel 4.2.	Data Ukuran <i>Support Cylinder</i> untuk Pemilihan dari <i>Database</i>	30
Tabel 4.3.	Data Ukuran <i>Strap Clamp</i> untuk Pemilihan dari <i>Database</i>	31
Tabel 4.4.	Definisi aktor yang terlibat	38
Tabel 4.5.	Definisi <i>Use Case</i> yang digunakan	38
Tabel 4.6.	Definisi <i>Use Case</i> Login	40
Tabel 4.7.	Definisi <i>Use Case</i> Logout	41
Tabel 4.8.	Definisi <i>Use Case</i> menyimpan gambar 3D	41
Tabel 4.9.	Definisi <i>Use Case</i> Membuat perhitungan <i>fixturing</i>	42
Tabel 4.10.	Definisi <i>Use Case</i> menyimpan desain hasil perhitungan <i>fixturing</i>	44
Tabel 4.11.	Definisi <i>Use Case</i> menghapus desain hasil perhitungan <i>fixturing</i>	44
Tabel 4.12.	Definisi <i>Use Case</i> mencetak laporan desain hasil perhitungan	44

BAB I PENDAHULUAN

1.1. Latar Belakang

Dalam siklus proses produksi, *fixturing* merupakan salah satu tahapan penting dalam proses manufaktur yang berfungsi untuk mendukung, menempatkan, dan menahan *workpiece* selama pemesinan (Krishnakumar dan Melkote, 1999). Sebelum dikenalnya *Computer Aided Manufacture* (CAM) untuk mendukung proses *fixturing*, desain *fixture* dikerjakan secara manual dengan berdasar kepada pengetahuan heuristik desainer *fixture* untuk menempatkan lokator, *clamp*, dan *support* yang sesuai pada *workpiece*. Akan tetapi, desain *fixture* dengan metode manual membutuhkan waktu yang tidak sedikit. Menurut Kumar et al. (1999) pada kasus dimana desain *fixture* mengandalkan kemampuan dari desainer, produktifitas akan menurun seiring dengan banyaknya waktu yang dibutuhkan untuk melakukan iterasi konfigurasi terhadap jenis *workpiece* dan proses pemesinannya.

Seiring dengan perkembangan teknik dan metode *fixturing*, penggunaan *fixture* modular mulai mendapat perhatian sebagai bagian dari *Flexible Manufacturing System* (FMS) dan *Computer-Integrated Manufacturing Systems* (CIMS). *Modular fixture* adalah jenis *fixture* yang menyediakan alternatif konfigurasi dengan menggunakan variasi dari kombinasi elemen *fixture* (Kang dan Peng, 2009). Berdasarkan jenis komponen utamanya (*tool body*), *fixture modular* terbagi kedalam dua jenis, yaitu *T-slot* dan *dowel pin* atau *hole-based fixture*. Diantara kedua jenis tersebut, *modular fixture* jenis *hole-based* memiliki kegunaan yang lebih fleksibel untuk berbagai jenis benda yang akan dimesin, terutama untuk digunakan pada industri dengan jenis produksi *job* dan *batch* (Segal et al., 2001).

Untuk mendukung sistem produksi yang terintegrasi, aplikasi *modular fixture* yang berbasis desain *Computer Aided Manufacturing* (CAD) semakin dibutuhkan. Penelitian mengenai pengembangan sistem cerdas yang mampu

mengotomasi pemilihan elemen *modular fixture* dan mengatur tata letak *fixture*-nya sudah banyak diteliti seperti pada penelitian yang dilakukan oleh Chou (1992), Ariastuti et al. (1998), Hartono dan Herliansyah (2003) dan Farhan (2013). Namun, pengembangannya masih bersifat searah karena masing-masing belum meninjau suatu perangkat lunak dari sisi pendekatan sistem yang memadukan aspek mekanis, rekayasa perangkat lunak, serta perancangan dan pengembangan produk. Padahal, saat ini perangkat lunak pada dasarnya memiliki peran ganda. Satu sisi perangkat lunak merupakan produk dan pada sisi lainnya perangkat lunak adalah sarana untuk mengantarkan suatu produk (Pressman, 2010).

Peninjauan terhadap sisi perangkat lunak sebagai produk dalam sudut pandang sistem bukan merupakan suatu hal yang sederhana. Pressman (2010) menyatakan bahwa produk perangkat lunak membutuhkan sebuah tim yang terdiri dari berbagai disiplin ilmu untuk bekerja sama dalam setiap fase perancangan dan pengembangan produk perangkat lunak. Tujuannya agar tercipta suatu produk yang berkualitas dan tepat guna.

Untuk mencapai tujuan tersebut, suatu produk perangkat lunak harus memiliki perancangan yang baik. Fase ini merupakan tahapan yang krusial dalam melakukan pengembangan perangkat lunak. Tahapan perancangan membuat satu representasi atau memodelkan sistem yang bekerja pada perangkat lunak, tapi tidak seperti model kebutuhan yang berfokus pada bagaimana cara mendeskripsikan data, fungsi, serta perilaku (Pressman, 2010).

Pada dasarnya, model perancangan perangkat lunak menyediakan detail dari hal-hal yang berkaitan dengan arsitektur perangkat lunak yang bersangkutan. Adanya suatu model yang mendeskripsikan sistem arsitektur yang bekerja dalam perangkat lunak dapat meminimalisir kecacatan produk dan langsung diperbaiki sebelum kode-kode program dieksekusi.

Penelitian yang telah ada sebelumnya seperti yang dilakukan oleh Hartono dan Herliansyah (2003) dan Farhan (2011) mampu untuk mendesain sebuah *Computer Aided Fixture Design (CAFD)* yang berbasis simulasi pada sistem 3D CAD. Akan tetapi dalam sistem yang telah dikembangkan tersebut, pemilihan