

# Component Diagram

# Software Implementation Tools

- Dalam proyek iteratif, planning untuk implementation akan di mulai dari fase inception
- implementation workflow meliputi tasks untuk set up lingkungan untuk implementasi
- Beberapa alat, terutama alat pemodelan dan sistem manajemen konfigurasi akan digunakan pada kegiatan analisis dan desain
- Berbagai jenis perangkat lunak yang akan digunakan

# Software Implementation Tools

- Modelling Tools
  - Banyak alat sekarang mendukung UML
  - Memungkinkan untuk menghasilkan kode (Java, C++ dan VB) dari model
  - Dapat membuat reverse engineering dari kode yang mungkin, untuk memberikan teknik round-trip
  - Mungkin memetakan kelas untuk database relasional
  - Link ke alat manajemen konfigurasi

# Software Implementation Tools

- Compilers, Interpreters dan Run-times
  - Bahasa yang berbeda memerlukan alat yang berbeda
  - C + + memerlukan compiler dan linker untuk membangun executable
  - Java memerlukan compiler dan program run-time dan library untuk menjalankan byte-kode yang dihasilkan oleh kompilator
  - C # adalah seperti Java dan dikompilasi ke MSIL (Microsoft Intermediate Language)

# Software Implementation Tools

- Visual Editors
  - Menyediakan cara merancang antarmuka GUI dengan mendragn dan dropping tombol, teksarea dll ke jendela
  - Bisa juga menangani kontrol atau objek yang mewakili komponen non-visual seperti link ke database atau proses komunikasi

# Software Implementation Tools

- IDEs (Integrated Development Environments)
  - Mengelola banyak file dalam proyek dan dependensi di antara mereka
  - Link ke alat manajemen konfigurasi
  - Gunakan compiler untuk membangun proyek, hanya mengkompilasi ulang apa yang telah berubah
  - Menyediakan fasilitas debugging
  - Mungkin termasuk editor visual
  - Dapat dikonfigurasi untuk link dalam alat pihak ketiga

# Software Implementation Tools

- Configuration Management Tools
  - Disebut juga alat kontrol versi, meskipun manajemen konfigurasi adalah lebih dari sekedar kontrol versi
  - Mempertahankan catatan versi file dan perubahan dari satu versi ke yang berikutnya
  - Catat semua versi perangkat lunak dan alat-alat yang diperlukan untuk menghasilkan perangkat lunak berulang membangun

# Software Implementation Tools

- Class Browsers
  - Mungkin bagian dari IDE atau editor visual yang
  - Awalnya disediakan sebagai cara browsing melalui kelas yang tersedia di Smalltalk
  - Dokumentasi API Java disediakan dalam format hypertext browseable dihasilkan oleh Javadoc

# Software Implementation Tools

- Manajer komponen
  - Alat untuk mengelola komponen dan dependensi mereka
  - Menyediakan mekanisme untuk
    - menambahkan komponen
    - mencari komponen
    - browse untuk komponen
    - mempertahankan versi komponen

# Software Implementation Tools

- DBMS (Database Management Systems)
  - sistem server
  - Perangkat lunak klien (interface administrasi, ODBC dan JDBC driver)
  - Alat untuk mengelola database dan melakukan tuning kinerja
  - Besar DBMS, seperti Oracle, dengan banyak alat, bahkan server aplikasi mereka sendiri

# Software Implementation Tools

- CORBA
  - CORBA ORB untuk menangani menyusun dan unmarshalling permintaan dan benda-benda
  - IDL compiler
  - layanan Registry

# Software Implementation Tools

- Application Containers
  - web kontainer
    - seperti Tomcat
    - Jalankan servlet dan aplikasi skala kecil
  - aplikasi server
    - Seperti WebSphere, WebLogic atau Jboss
    - Menyediakan kerangka kerja di mana untuk menjalankan skala besar, aplikasi perusahaan

# Software Implementation Tools

- Testing Tools
  - Peralatan ditulis oleh pengembang untuk tes
  - Alat uji otomatis untuk menjalankan tes simultan berulang atau beberapa
  - Memungkinkan pengguna untuk menjalankan melalui tes sekali secara manual, kemudian menghasilkan sebuah script yang dapat diedit untuk memberikan variasi

# Software Implementation Tools

- Installation Tools
  - Mengotomatisasi ekstraksi file dari sebuah arsip dan pengaturan dari file konfigurasi dan entri registry
  - Beberapa mempertahankan informasi tentang ketergantungan pada bagian lain dari perangkat lunak dan akan menginstal semua paket yang diperlukan (misalnya Redhat RPM)
  - Uninstall software, file menghapus, direktori dan entri registri (jika Anda beruntung!)

# Software Implementation Tools

- Conversion Tools
  - Mengekstrak data dari sistem yang ada
  - Format data untuk sistem yang baru
  - Masukkan ke dalam database untuk sistem yang baru
  - Mungkin memerlukan intervensi manual untuk 'membersihkan' data-menghapus duplikasi atau nilai-nilai yang tidak valid

# Software Implementation Tools

- Documentation Generators
  - Model Dokumen dan kode
  - Mengekstrak informasi standar atau ditetapkan pengguna informasi ke dalam dokumen template
  - Menghasilkan HTML untuk mendokumentasikan API kelas dalam aplikasi

# Coding dan Standar Dokumentasi

- Standar penamaan yang disepakati di awal proyek
- Sebuah standar khas berorientasi obyek:
  - kelas dengan huruf kapital: `Campaign`
  - atribut dan operasi dengan awal huruf kecil: `title`, `recordPayment ( )`
  - kata-kata yang digabung bersamaan dengan huruf kapital untuk menunjukkan di mana mereka bergabung: `InternationalCampaign`, `campaignFinishDate`, `getNotes ( )`

# Coding dan Standar Dokumentasi

- Notasi Hungaria
- Digunakan dalam C dan C + +
- Nama diawali dengan singkatan untuk menunjukkan jenis variabel anggota
  - b for boolean: **bOrderClosed**
  - i for integer: **iOrderLineNumber**
  - btn for button: **btnCloseOperation**

# Coding dan Standar Dokumentasi

- Satu standar lainnya:
  - Menggunakan garis bawah untuk memisahkan bagian nama bukan huruf kapital  
**Order\_Closed**
  - sering digunakan untuk nama kolom dalam database, karena lebih mudah untuk menggantikan menggarisbawahi dengan spasi untuk menghasilkan judul kolom dalam laporan

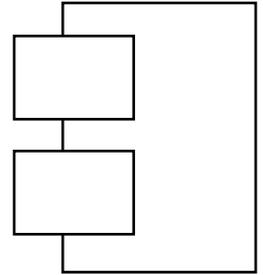
# Coding dan Standar Dokumentasi

- Document code
  - Pikirkan orang-orang yang akan menggunakan kode Anda
  - Orang lain mungkin dapat menggunakan kode Anda untuk belajar praktek yang baik, tapi hanya jika jelas didokumentasikan
  - Bahasa ada yang membantu mendokumentasikan diri, konvensi dan standar
  - Sesuai dengan standar dokumentasi Java, jika coding di Java (Javadoc)
  - Gunakan tag XML jika coding di C #
  - Anda dapat mengambil keuntungan dari alat yang mengotomatisasi produksi dokumentasi dari komentar

# Implementation Diagrams

- Component Diagrams
  - digunakan untuk mendokumentasikan ketergantungan antara komponen, yaitu unit perangkat lunak modular dengan antarmuka yang terdefinisi dengan baik
- Deployment Diagrams
  - digunakan untuk menunjukkan konfigurasi elemen pemrosesan run-time dan artefak perangkat lunak serta letak proses

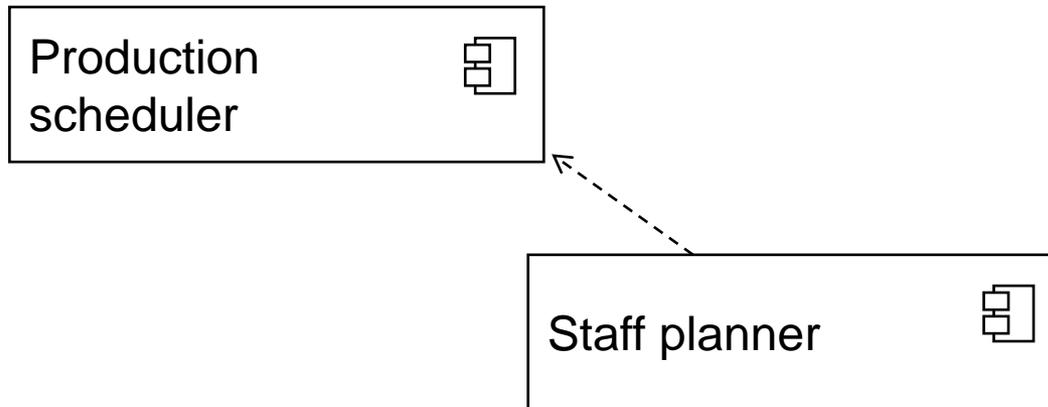
# Notasi Komponen Diagram



- Components
  - persegi panjang dengan ikon komponen di sudut kanan atas
  - dapat memberikan atau memerlukan interface
- Dependencies

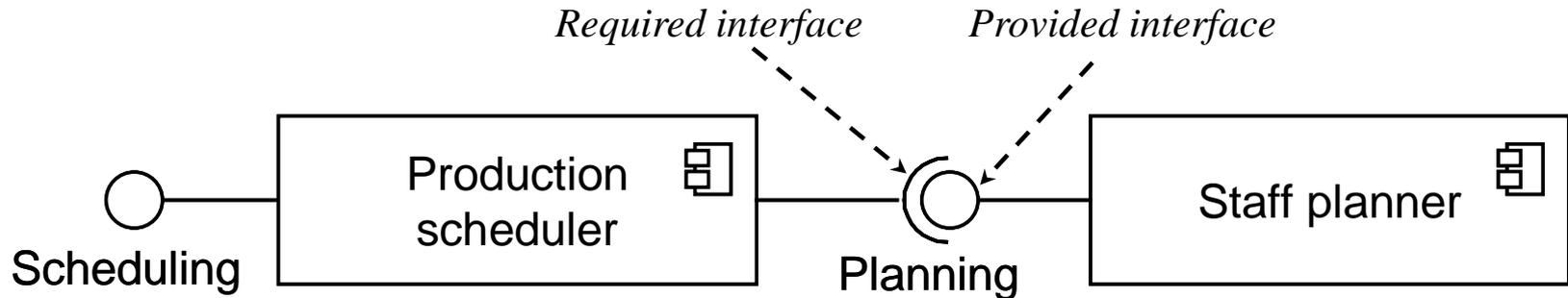
# Notasi Komponen Diagram

- Dependencies antara components



# Notation of Component Diagrams

- Wiring connection antara required dan provided interfaces



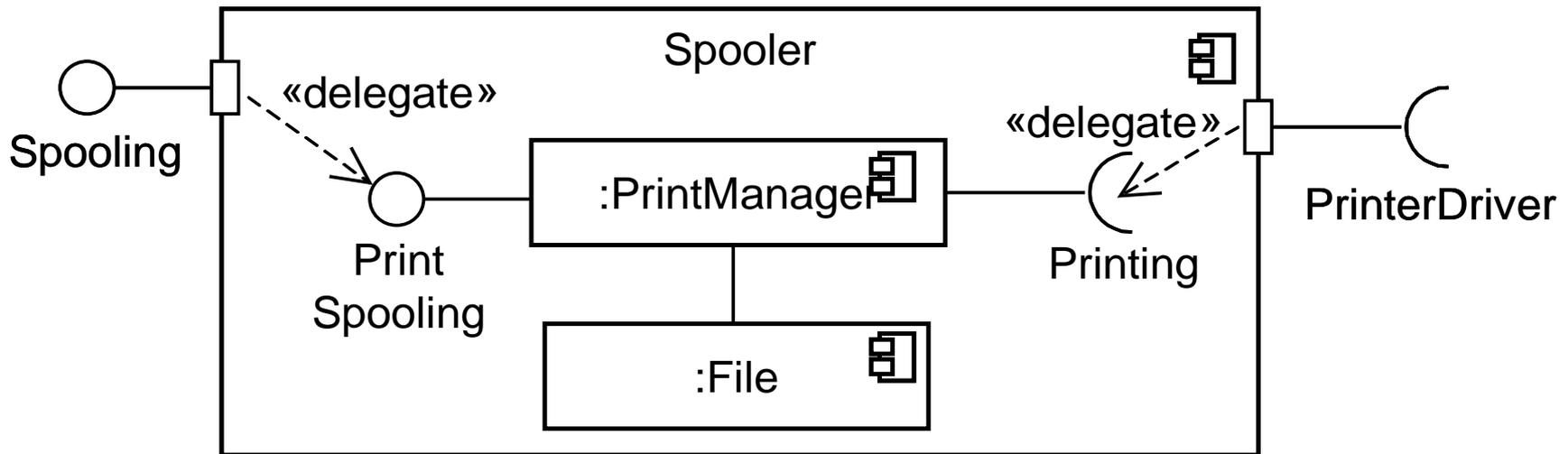
# Notation of Component Diagrams

- Component dengan ports
  - Menunjukkan bahwa komponen delegasi tanggung jawab atas perilaku antarmuka yang ke subkomponen



# Notation of Component Diagrams

- Component dengan ports
  - Menunjukkan tanggung jawab yang didelegasikan



# Components

- Komponen telah berubah dalam UML 2.0
- Mereka tidak lagi ditampilkan di Diagram Deployment, di mana mereka telah digantikan oleh Artefak
- Komponen yang khusus unit perangkat lunak modular dengan antarmuka yang terdefinisi dengan baik
- Bisa menjadi logis atau fisik

# Forms Component

- Cheesman dan Daniels membuat perbedaan yang jelas antara berbagai jenis komponen:
  - Component specification
  - Component implementation
  - Installed component
  - Component object

# Referensi

- Deitel and Deitel (1997)  
(For full bibliographic details, see Bennett, McRobb and Farmer)