



# Kebutuhan Software

Deskripsi dan Spesifikasi Sistem

Arna Fariza

1

Rekayasa Perangkat Lunak




## Tujuan


- ☐ Memperkenalkan konsep kebutuhan user dan sistem
- ☐ Menggambarkan kebutuhan fungsional dan non-fungsional
- ☐ Menjelaskan dua teknik menggambarkan kebutuhan sistem
- ☐ Menjelaskan bagaimana kebutuhan software diorganisasikan dalam dokumen kebutuhan

Rekayasa Perangkat Lunak

2




# Materi




- ☐ Kebutuhan fungsional dan non-fungsional
- ☐ Kebutuhan user
- ☐ Kebutuhan sistem
- ☐ Dokumen kebutuhan software

Rekayasa Perangkat Lunak

3



# Rekayasa Kebutuhan



- ☐ Proses menetapkan layanan yang dibutuhkan konsumen terhadap sistem dan batasan operasi dan pengembangan
- ☐ Kebutuhan sendiri adalah diskripsi layanan sistem dan batasan yang dibangkitkan selama proses rekayasa kebutuhan

Rekayasa Perangkat Lunak

4



## Apakah Kebutuhan itu?




- ❑ Susunan pernyataan abstrak level tinggi dari layanan atau batasan sistem ke dalam spesifikasi fungsional matematis
- ❑ Tidak terelakkan bahwa kebutuhan mempunyai dua fungsi
  - o merupakan dasar untuk penawaran kontrak – sehingga harus terbuka untuk interpretasi
  - o merupakan dasar untuk kontrak itu sendiri – sehingga harus didefinisikan dengan detail
  - o Kedua pernyataan diatas disebut kebutuhan




## Abstraksi Kebutuhan (Davis)



“Jika sebuah perusahaan akan mengadakan kontrak untuk proyek pengembangan software besar, harus didefinisikan kebutuhan yang cukup dimana solusi belum terdefinisi. Kebutuhan harus ditulis sehingga beberapa kontraktor dapat menawarkan kontrak, penawaran, kemungkinan, secara berbeda dengan kebutuhan organisasi client. Bila kontrak sudah diserahkan, kontraktor harus menulis definisi sistem untuk client secara lebih detail sehingga client mengerti dan dapat mem-validasi software yang akan dikerjakan. Kedua dokumen ini disebut *dokumen kebutuhan* untuk sistem”




# Tipe-tipe Kebutuhan




- ❑ **Kebutuhan User**
  - o Pernyataan dalam bahasa natural plus diagram layanan yang tersedia dan batasan operasional. Ditulis oleh konsumen
- ❑ **Kebutuhan Sistem**
  - o Dokumen terstruktur berisi diskripsi detail dari layanan sistem. Ditulis sebagai kontrak antara klien dan kontraktor.
- ❑ **Spesifikasi Software**
  - o Diskripsi software detail yang sebagai dasar untuk desain atau implementasi. Ditulis oleh developer

Rekayasa Perangkat Lunak
7



# Definisi dan Spesifikasi



**Definisi Kebutuhan**

1. Software harus menyediakan ketentuan menampilkan dan mengakses file yang dibuat oleh tool lain

**Spesifikasi Kebutuhan**

1.1 User diberikan fasilitas untuk mendefinisikan tipe file eksternal

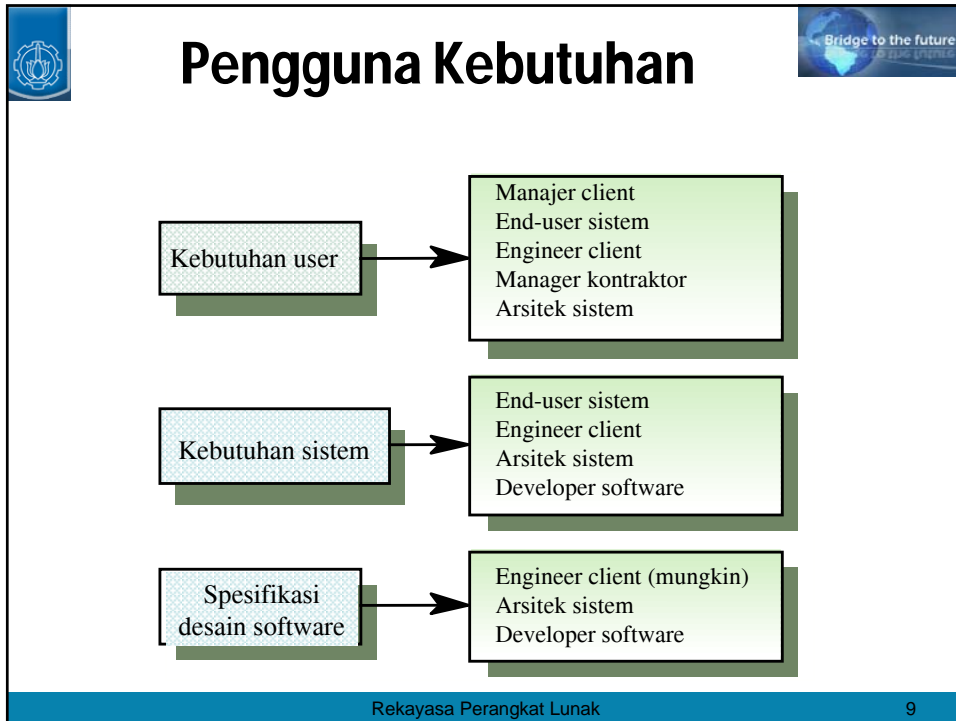
1.2 Setiap tipe file eksternal mempunyai alat untuk dihubungkan yang dapat diaplikasikan ke file

1.3 Setiap tipe file eksternal direpresentasikan sebagai icon tertentu pada tampilan user

1.4 Fasilitas disediakan untuk icon yang merepresentasikan tipe file eksternal yang didefinisikan oleh user

1.5 Jika user memilih icon untuk merepresentasikan file eksternal, efek pemilihan mengaplikasikan alat yang menghubungkan antara tipe file eksternal ke file yang direpresentasikan oleh icon terpilih

Rekayasa Perangkat Lunak
8



- 
- Kebutuhan Fungsional dan non-fungsional**
- ❑ Kebutuhan fungsional
    - o Pernyataan layanan sistem yang harus disediakan, bagaimana sistem bereaksi pada input tertentu dan bagaimana perilaku sistem pada situasi tertentu
  - ❑ Kebutuhan non-fungsional
    - o Batasan layanan atau fungsi yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses, standarisasi dll
  - ❑ Kebutuhan domain
    - o Kebutuhan yang datang dari domain aplikasi dari sistem dan yang menyatakan karakteristik dari domain tersebut
- Rekayasa Perangkat Lunak 10



## Kebutuhan Fungsional



- ☐ Menggambarkan fungsionalitas atau layanan sistem
- ☐ Tergantung pada tipe software, harapan user dan tipe sistem dimana software digunakan
- ☐ Kebutuhan fungsional user merupakan pernyataan level tinggi dari apa yang seharusnya dilakukan sistem tetapi kebutuhan fungsional sistem menggambarkan layanan sistem secara detail



## Contoh Kebutuhan Fungsional



- ☐ User dapat mencari semua kumpulan database inisial atau memilih subset dari database tersebut
- ☐ Sistem menyediakan tampilan yang tepat untuk user yang membaca dokumen dalam penyimpanan dokumen
- ☐ Setiap pesanan dapat dialokasikan sebagai identifier yang unik (ORDER\_ID) dimana user dapat meng-copy daerah penyimpanan account permanen



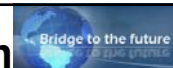
## Ketidaktepatan Kebutuhan



- ☐ Permasalahan timbul jika kebutuhan tidak ditetapkan dengan jelas
- ☐ Kebutuhan yang sama mungkin diinterpretasikan dengan cara yang berbeda oleh developer dan user
- ☐ Viewer yang tepat
  - o Kemauan user - viewer tujuan khusus untuk tipe dokumen yang berbeda
  - o Interpretasi developer - menyediakan text viewer yang menunjukkan isi dokumen



## Konsistensi dan Kelengkapan Kebutuhan



- ☐ Kebutuhan prinsip harus lengkap dan konsisten
- ☐ Lengkap
  - o Harus mendiskripsikan semua fasilitas yang dibutuhkan
- ☐ Konsisten
  - o Seharusnya tidak ada konflik atau kontradiksi dalam diskripsi fasilitas sistem
- ☐ Dalam prakteknya, mungkin untuk memproduksi dokumen kebutuhan yang lengkap dan konsisten



## Kebutuhan Non-fungsional



- ❑ Mendefinisikan properti sistem dan batasan sistem, seperti kehandalah, waktu respon, kebutuhan penyimpanan. Batasan misalnya kapabilitas perangkat I/O, representasi sistem dll
- ❑ Kebutuhan proses juga menetapkan penggunaan sistem CASE khusus, bahasa pemrograman atau metode pengembangan
- ❑ Kebutuhan non-fungsional lebih kritis daripada kebutuhan fungsional. Jika tidak dapat bertemu, sistem menjadi tidak berguna

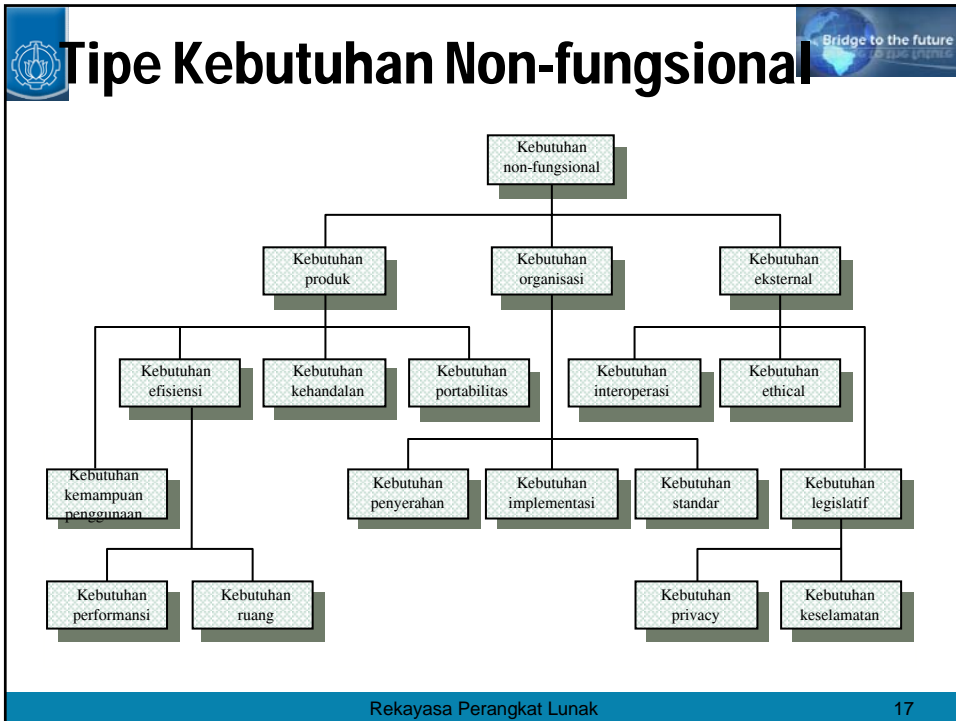


## Klasifikasi Non-fungsional



- ❑ Kebutuhan Produk
  - o Kebutuhan yang menetapkan bahwa produk yang dikirim harus berjalan dengan cara tertentu, contoh kecepatan eksekusi, kehandalan dll
- ❑ Kebutuhan Organisasi
  - o Kebutuhan sebagai akibat dari kebijakan organisasi dan prosedur misalnya standar proses yang digunakan, kebutuhan implementasi dll
- ❑ Kebutuhan Eksternal
  - o Kebutuhan yang muncul dari faktor eksternal sistem dan proses pengembangan misalnya kebutuhan antar operasi, kebutuhan legislatif dll






## Contoh Kebutuhan Non-fungsional


- ❑ Kebutuhan Produk
  - o 4.C.8 Dimungkinkan untuk semua komunikasi yang diperlukan antara APSE dan user diekspresikan dalam karakter Ada standar
- ❑ Kebutuhan Organisasi
  - o 9.3.2 Proses pengembangan sistem dan penyerahan dokumen seharusnya sesuai dengan proses dan penyerahan yang didefinisikan dalam XYZCo-SP-STAN-95
- ❑ Kebutuhan Eksternal
  - o 7.6.5 Sistem seharusnya tidak tertutup untuk segala informasi personal tentang konsumen lepas dari nama dan nomor referensi ke operator sistem

Rekayasa Perangkat Lunak

18




## Tujuan dan Kebutuhan




- ❑ Kebutuhan Non-fungsional kemungkinan sangat sulit untuk ditetapkan dan kebutuhan yang tidak tepat sulit diverifikasi
- ❑ Tujuan
  - o Tujuan umum dari user misalnya kemudahan penggunaan
- ❑ Kebutuhan non-fungsional yang dapat diverifikasi
  - o Pernyataan menggunakan beberapa ukuran yang dapat dites secara obyektif
- ❑ Tujuan sangat membantu pengembangan sesuai penyampaian maksud user sistem

Rekayasa Perangkat Lunak
19




## Contoh




- ❑ Tujuan Sistem
  - o Sistem seharusnya mudah digunakan oleh pengguna dan diorganisasikan sehingga error user dapat diminimalkan
- ❑ Kebutuhan non-fungsional yang dapat diverifikasi
  - o Pengguna seharusnya dapat menggunakan semua fungsi sistem setelah training selesai. Setelah training ini, jumlah rata-rata error yang dibuat oleh user yang berpengalaman tidak lebih dari 2 setiap hari

Rekayasa Perangkat Lunak
20




## Ukuran Kebutuhan




Properti	Ukuran
Kecepatan	Transaksi proses/detik Waktu respon user/event Waktu refresh screen
Ukuran	K byte Jumlah chip RAM
Kemudahan penggunaan	Waktu training Jumlah frame bantuan
Kehandalan	Rata-rata waktu kegagalan Kemungkinan ketidak tersediaan Rata-rata kejadian kegagalan Ketersediaan
Kekuatan	Waktu restart setelah kegagalan Persentase event yang menyebabkan kegagalan Kemungkinan korupsi data pada saat kegagalan
Portabilitas	Persentase pernyataan ketergantungan target Jumlah sistem target

Rekayasa Perangkat Lunak
21




## Interaksi Kebutuhan




- ☐ Konflik antara kebutuhan non-fungsional bersifat umum dalam sistem yang kompleks
- ☐ Sistem kendaraan luar angkasa
  - o Untuk meminimalkan berat, jumlah chip yang terpisah pada sistem harus diminimalkan
  - o Untuk meminimalkan konsumsi listrik, listrik chip lebih rendah harus digunakan
  - o Sehingga, menggunakan chip listrik rendah berarti lebih banyak chip yang harus digunakan. Mana yang merupakan kebutuhan kritis?

Rekayasa Perangkat Lunak
22



## Kebutuhan Domain



- ☐ Diperoleh dari domain aplikasi dan menggambarkan karakteristik sistem dan fitur yang merefleksikan domain
- ☐ Berupa kebutuhan fungsional baru, batasan pada kebutuhan yang sudah ada atau mendefinisikan komputasi tertentu
- ☐ Jika kebutuhan domain tidak terpenuhi, sistem mungkin tidak dapat bekerja

Rekayasa Perangkat Lunak

23



## Kebutuhan Domain sistem Perpustakaan



- ☐ Terdapat antar muka standar user untuk semua database berdasarkan standar X39.50
- ☐ Karena pembatasan copyright, beberapa dokumen harus dihapus segera setelah datang. Tergantung kebutuhan user, dokumen tersebut dapat dicetak secara lokal pada server sistem untuk dikirim secara manual ke user atau dilewatkan ke network printer

Rekayasa Perangkat Lunak

24



## Sistem Proteksi Kereta Api



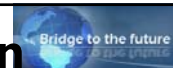
- ❑ Perlambatan kereta dapat dihitung sbb:

- o  $D_{\text{train}} = D_{\text{control}} + D_{\text{gradient}}$


dimana  $D_{\text{gradient}}$  sebesar  $9.81\text{ms}^2 * \text{ganti\_rugi}$   
gradient/alpha dan dimana nilai  $9.81\text{ms}^2$   
/alpha diketahui dari tipe kereta api yang  
berbeda




## Permasalahan Kebutuhan Domain



- ❑ Kemampuan untuk dimengerti
  - o Kebutuhan dinyatakan dalam bahasa domain aplikasi
  - o Biasanya tidak dimengerti oleh software engineer yang mengembangkan sistem
- ❑ Ketidaklengkapan
  - o Domain spesialis mengerti area dengan baik sehingga mereka tidak berfikir untuk membuat kebutuhan domain secara eksplisit




## Kebutuhan User




- ☐ Menjelaskan kebutuhan fungsional dan non-fungsional sehingga user yang tidak mempunyai pengetahuan teknis detail dapat mengerti sistem
- ☐ Kebutuhan user didefinisikan menggunakan bahasa natural, tabel dan diagram

Rekayasa Perangkat Lunak

27



## Permasalahan dengan Bahasa Alami



- ☐ Ketidakjelasan
  - o Kecermatan sulit diwujudkan tanpa membuat dokumen yang sulit dibaca
- ☐ Kebutuhan yang membingungkan
  - o Kebutuhan fungsional dan non-fungsional cenderung dicampur aduk
- ☐ Penggabungan kebutuhan
  - o Beberapa kebutuhan yang berbeda dinyatakan bersama-sama

Rekayasa Perangkat Lunak

28



## Kebutuhan Database



**4.A.5** Database harus mendukung pembangkitan dan kontrol dari konfigurasi obyek; sehingga obyek terkelompok sendiri dalam database. Fasilitas kontrol konfigurasi memungkinkan akses ke obyek menggunakan nama yang tidak lengkap



## Kebutuhan Editor grid



**2.6 Fasilitas Grid** untuk membantu meletakkan entiti pada diagram, user harus mengubah grid dalam cm atau inch, melalui pilihan pada control panel. Semula, grid off. Grid diubah on dan off selama waktu editing dan dapat diubah inch dan cm sepanjang waktu. Pilihan grid akan disediakan pilihan penurunan tetapi jumlah garis grid yang ditampilkan akan diturunkan untuk menghindari mengisi diagram lebih kecil dengan garis grid.



## Permasalahan Kebutuhan



- ❑ Kebutuhan database terdiri dari konseptual dan informasi detail
  - o Menggambarkan konsep fasilitas kontrol konfigurasi
  - o Terdiri dari detail obyek yang boleh diakses menggunakan nama yang tak lengkap
- ❑ Kebutuhan grid menggabungkan 3 kebutuhan yang berbeda
  - o Kebutuhan konseptual fungsional (kebutuhan untuk grid)
  - o Kebutuhan non-fungsional (unit grid)
  - o Kebutuhan antar muka non-fungsional (grid switching)



## Presentasi Terstruktur



### 2.6 Fasilitas Grid

#### 2.6.1 Editor menyediakan fasilitas grid dimana matriks baris horisontal dan vertikal merupakan background window editor.

Grid ini merupakan passive grid dimana ukuran entitas adalah tanggung jawab user

Alasan: grid membantu user membuat diagram teratur dengan entitas teratur. Meskipun grid aktif, dimana entitas garis grid dapat berguna, posisi tidak teliti. User adalah orang yang tepat untuk menentukan dimana entitas seharusnya diletakkan





## Kebutuhan User Detail



### 3.5.1 Penambahan node ke desain

#### 3.5.1.1 Editor menyediakan fasilitas untuk user menambah node untuk tipe tertentu pada desain

3.5.1.2 Urutan akse untuk menambah node sebagai berikut:

1. User memilih tipe node yang ditambahkan
2. User memindahkan kursor ke posisi node yang tepat pada diagram dan mengindikasikan bahwa simboh node ditambahkan pada titik tersebut
3. User menarik simbol node ke posisi akhir

Alasan: user adalah orang terbaik untuk memutuskan dimana posisi node pada diagram. Pendekatan ini memberikan kontrol langsung user ke semua pemilihan dan memosisikan tipe node


Spesifikasi: BCLIPSB/WS/Tools/DB/FS.Section 3.5.1




## Ketentuan Penulisan Kebutuhan



- ☐ Menggunakan format standard dan menggunakannya untuk semua kebutuhan
- ☐ Menggunakan bahasa secara konsisten. Penggunaannya untuk kebutuhan utama, untuk kebutuhan yang sangat diperlukan
- ☐ Menggunakan penanda teks untuk identifikasi bagian penting dari kebutuhan
- ☐ Hindari penggunaan jargon komputer




## Kebutuhan Sistem




- ☐ Spesifikasi yang lebih detail dari kebutuhan user
- ☐ Sebagai dasar mendesain sistem
- ☐ Biasanya digunakan sebagai bagian dari kontrak sistem
- ☐ Kebutuhan sistem diekspresikan menggunakan model sistem

Rekayasa Perangkat Lunak

35




## Kebutuhan dan Desain




- ☐ Secara prinsip, kebutuhan menyatakan apa yang seharusnya sistem lakukan dan desain menggambarkan bagaimana melakukannya
- ☐ Prakteknya, kebutuhan dan desain dibedakan
  - o Arsitektur sistem di desain untuk men-strukturkan kebutuhan
  - o Sistem dapat beroperasi dengan sistem lain yang membangkitkan kebutuhan desain
  - o Penggunaan desain tertentu mungkin menjadi kebutuhan domain

Rekayasa Perangkat Lunak

36




# Permasalahan dengan Spesifikasi Bahasa Alami




- ❑ Kemenduaan
  - o Pembaca dan penulis kebutuhan harus menginterpretasikan kata yang sama dg cara yang sama. Bahasa alami secara natural bersifat mendua sehingga sangat sulit
- ❑ Terlalu Fleksibel
  - o Hal yang sama mungkin dikatakan dengan beberapa cara yang berbeda dalam spesifikasi
- ❑ Tidak ada modularitas
  - o Struktur bahasa alami tidak cukup untuk men-strukturkan kebutuhan sistem

Rekayasa Perangkat Lunak

37




# Alternatif Spesifikasi Bahasa Alami




Notasi	Deskripsi
Bahasa natural terstruktur	Pendekatan ini tergantung definisi bentuk standard atau template untuk menggambarkan spesifikasi kebutuhan
Bahasa deskripsi desain	Pendekatan ini menggunakan bahasa seperti bahasa pemrograman tetapi lebih banyak fitur abstrak untuk menentukan kebutuhan dengan mendefinisikan model operasi dari sistem
Notasi grafis	bahasa grafis, ditambahkan dengan text digunakan untuk mendefinisikan kebutuhan fungsional untuk sistem Contohnya sebelumnya bahasa grafis SADT (Ross, 1977; Schoman and Ross, 1977), saat ini digunakan deskripsi use case (Jacobsen, Christerson et al.,1993)
Spesifikasi Matematis	Ada beberapa notasi berbasis konsep matematis seperti mesin finite-state atau himpunan. Spesifikasi yang ganda dapat mengurangi argumen antara konsumen dan kontraktor tentang fungsional sistem. Sebaliknya, sebagian besar konsumen tidak mengerti spesifikasi formal dan enggan menerimanya sebagai kontrak sistem

Rekayasa Perangkat Lunak

38




## Spesifikasi Bahasa Terstruktur




- ☐ Bentuk terbatas dari bahasa alami (Natural Language) dapat digunakan untuk menggambarkan kebutuhan
- ☐ Menghilangkan beberapa permasalahan yang menghasilkan kemenduaan dan fleksibilitas dan gangguan level keseragaman pada spesifikasi
- ☐ Biasanya menggunakan pendekatan berbasis form

Rekayasa Perangkat Lunak

39




## Spesifikasi Berbasis Form




- ☐ Definisi fungsi atau entiti
- ☐ Menggambarkan input dan dimana berasal
- ☐ Menggambarkan output dan dimana berjalan
- ☐ Mengindikasikan entiti lain yang dibutuhkan
- ☐ Kondisi sebelum dan sesudah (jika diperlukan)
- ☐ Efek lain (jika ada)

Rekayasa Perangkat Lunak

40




# Spesifikasi node Berbasis Form




BCLIPSB/Workstation/Tools/DB/3.5.1

<b>Function</b>	Tambah node
<b>Deskripsi</b>	Menambah node ke desain yang sudah ada. User memilih tipe node dan posisi bila ditambahkan ke desain, node menjadi pilihan saat ini. User memilih posisi node dan memindahkan kursor ke area dimana node ditambahkan
<b>Input</b>	Tipe node, posisi node, identifier desain
<b>Source</b>	Tipe node dan posisi node diinputkan oleh user, identifier desain dari database
<b>Output</b>	Identifier desain
<b>Tujuan</b>	Desain database. Desain membuat database menyelesaikan operasi
<b>Kebutuhan</b>	Desain graph akar dari identifier desain input
<b>Pre-kondisi</b>	Desain terbuka dan ditampilkan pada layar user
<b>Post-kondisi</b>	Desain tidak berubah terpisah dari tambahan node tipe tertentu pada posisi yang diberikan
<b>Efek lain</b>	Tidak ada

Rekayasa Perangkat Lunak
41



# Definisi Kebutuhan Berbasis Bahasa Pemrograman (PDL)



- ☐ Kebutuhan dapat didefinisikan secara operasional menggunakan bahasa seperti bahasa pemrograman tetapi lebih fleksibel dalam ekspresi
- ☐ Cocok untuk 2 situasi yaitu
  - o Dimana operasi ditentukan sebagai deretan aksi dan urutan sangat penting
  - o Dimana antar muka hardware dan software harus dispesifikasi
- ☐ Kelemahannya
  - o PDL tidak cukup ekspresif untuk mendefinisikan konsep domain
  - o Spesifikasi akan dianggap sebagai desain daripada spesifikasi

Rekayasa Perangkat Lunak
42



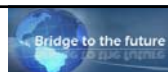
## Bagian dari spesifikasi ATM



```
class ATM{
    // declarations here
    public static void main (String args[]) throws InvalidCard {
        try {
            thisCard.read(); // may throw InvalidCard exception
            pin = KeyPad.readPin(); attempts = 1;
            while (!thisCard.pin.equals(pin) & attempts < 4)
            {
                pin = KeyPad.readPin(); attempts = attempts + 1;
            }
            if (thisCard.pin.equals(pin))
                throw new InvalidCard ("Bad PIN");
            thisBalance = thisCard.getBalance();
            do { Screen.prompt (" Please select a service ");
                service = Screen.touchKey();
                switch (service) {
                    case Services.withdrawalWithReceipt:
                        receiptRequired = true;
                }
            } while (true);
        } catch (InvalidCard e) {
            // handle exception
        }
    }
}
```



## Kelemahan PDL



- ☐ PDL tidak cukup ekspresif untuk menyatakan fungsional sistem yang dapat dimengerti
- ☐ Notasi hanya dapat dimengerti oleh orang yang mempunyai pengetahuan bahasa pemrograman
- ☐ Kebutuhan dianggap sebagai spesifikasi desain daripada model untuk membantu mengerti sistem



## Spesifikasi Antar Muka




- ❑ Sebagian besar sistem harus beroperasi dengan sistem lain dan antar muka operasi harus ditentukan sebagai bagian kebutuhan
- ❑ Tiga tipe antar muka
  - o Antar muka prosedural
  - o Struktur data yang dapat ditukar
  - o Representasi data
- ❑ Notasi formal adalah teknik efektif untuk spesifikasi antar muka




## Deskripsi Antar Muka PDL



```
interface PrintServer {  
  
    // defines an abstract printer server  
    // requires:      interface Printer, interface PrintDoc  
    // provides: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter  
  
    void initialize ( Printer p );  
    void print ( Printer p, PrintDoc d );  
    void displayPrintQueue ( Printer p );  
    void cancelPrintJob (Printer p, PrintDoc d);  
    void switchPrinter (Printer p1, Printer p2, PrintDoc d);  
} //PrintServer
```




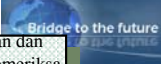
# Dokumen Kebutuhan



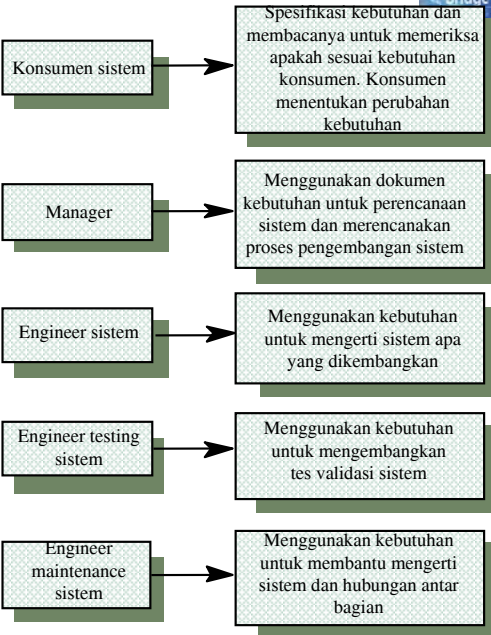
- ❑ Dokumen kebutuhan adalah pernyataan resmi dari apa yang dibutuhkan oleh developer sistem
- ❑ Terdiri dari definisi dan spesifikasi kebutuhan
- ❑ BUKAN dokumen desain. Sejauh mungkin, menentukan APA yang harus dikerjakan sistem daripada BAGAIMANA mengerjakannya

Rekayasa Perangkat Lunak
47





**User dari dokumen kebutuhan**




```

graph LR
    A[Konsumen sistem] --> B[Spesifikasi kebutuhan dan membacanya untuk memeriksa apakah sesuai kebutuhan konsumen. Konsumen menentukan perubahan kebutuhan]
    C[Manager] --> D[Menggunakan dokumen kebutuhan untuk perencanaan sistem dan merencanakan proses pengembangan sistem]
    E[Engineer sistem] --> F[Menggunakan kebutuhan untuk mengerti sistem apa yang dikembangkan]
    G[Engineer testing sistem] --> H[Menggunakan kebutuhan untuk mengembangkan tes validasi sistem]
    I[Engineer maintenance sistem] --> J[Menggunakan kebutuhan untuk membantu mengerti sistem dan hubungan antar bagian]
        
```

Rekayasa Perangkat Lunak
48






## Kebutuhan Dokumen Kebutuhan

Bridge to the future  
Building the future

- ☐ Menentukan perilaku sistem eksternal
- ☐ Menentukan batasan implementasi
- ☐ Mudah diubah
- ☐ Berlaku sebagai alat referensi untuk maintenance
- ☐ Menyimpan siklus hidup sistem, misalnya memprediksi perubahan
- ☐ Menentukan karakter respon untuk even yang tak diharapkan

Rekayasa Perangkat Lunak49



## Standard IEEE untuk Kebutuhan

Bridge to the future  
Building the future

- ☐ Pendahuluan
- ☐ Deskripsi Umum
- ☐ Kebutuhan spesifik
- ☐ Lampiran
- ☐ Indeks
- ☐ Merupakan struktur generik yang harus disesuaikan sistem spesifik

Rekayasa Perangkat Lunak50



## Struktur Dokumen Kebutuhan



- ☐ Pendahuluan
- ☐ Daftar Istilah
- ☐ Definisi Kebutuhan User
- ☐ Arsitektur Sistem
- ☐ Spesifikasi kebutuhan sistem
- ☐ Model sistem
- ☐ Evolusi sistem
- ☐ Lampiran
- ☐ Indeks



## Summary



- ☐ Kebutuhan menentukan apa yang seharusnya dilakukan sistem dan menentukan batasan operasi dan implementasi
- ☐ Kebutuhan fungsional menentukan servis sistem yang harus disediakan
- ☐ Kebutuhan non-fungsional membatasi pengembangan sistem atau pengembangan proses
- ☐ Kebutuhan user adalah pernyataan level tinggi apa yang sistem harus kerjakan



## Summary



- ☐ Kebutuhan sistem dapat ditulis dalam bahasa alami, tabel dan diagram
- ☐ Kebutuhan sistem dimaksudkan untuk mengkomunikasikan fungsi yang harus disediakan sistem
- ☐ Kebutuhan sistem dapat ditulis dalam bahasa natural terstruktur, PDL atau bahasa formal
- ☐ Dokumen kebutuhan software adalah pernyataan persetujuan dari kebutuhan sistem