

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Konsep Dasar Sistem**

Dalam mendefinisikan sistem terdapat dua kelompok pendekatan sistem yaitu :

1. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut :

Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Jogiyanto, 2005).

2. Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut :

Jerry F. Gerald dalam buku Jogiyanto Hartono (2005) mendefinisikan sistem yaitu suatu jaringan kerja dari prosedur-prosedur yang saling

berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan sasaran tertentu.

Sistem sebagai elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan (Kadir, 2003).

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :  
(Jogiyanto, 2005)

1) Memiliki Komponen Sistem (*Component*)

Suatu sistem terdiri dari komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen dari suatu sistem disebut subsistem. Subsistem ini mempunyai sifat-sifat dari sistem itu sendiri dalam menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

Suatu sistem juga mempunyai sistem yang lebih besar yang dikenal dengan suprasistem.

Contoh:

Jika suatu perusahaan dipandang sebagai suatu sistem, maka industri akan dipandang sebagai suprasistem.

2) Memiliki Batasan Sistem (*Boundary*)

Batasan sistem merupakan daerah yang membatasi antara sistem yang satu dengan sistem lainnya atau dengan lingkungan luarnya. Dengan

adanya batas sistem ini maka sistem dapat membentuk suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Dengan kata lain batas sistem ini merupakan ruang lingkup atau batasan dari sistem/subsistem itu sendiri.

Contoh :

Sistem Keuangan - Sistem Akuntansi

- Kasir

- Administrasi Keuangan

3) Personalia Memiliki Lingkungan Luar (*Environments*)

Segala sesuatu diluar dari batas sistem yg mempengaruhi operasi dari suatu sistem disebut Lingkungan luar sistem (*environment*).

Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang bersifat menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan yang bersifat merugikan harus dimusnahkan dan dikendalikan agar tidak mengganggu operasi dari sistem.

4) Memiliki Penghubung Sistem (*Interface*)

Penghubung Sistem merupakan suatu media penghubung antara satu subsistem dengan subsistem lainnya untuk membentuk satu kesatuan,

sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem lainnya. Dengan kata lain melalui penghubung ini output dari suatu subsistem akan menjadi input dari subsistem lainnya.

#### 5) Memiliki Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam suatu sistem disebut Input.

Masukan ini dapat berupa :

- Masukan Perawatan (*Maintenance Input*)

Yaitu energi yang dimasukkan supaya sistem itu dapat beroperasi.

Contoh : Program untuk mengoperasikan komputer.

- Masukan Sinyal (*Signal Input*)

Yaitu energi yang diproses untuk diperolehnya suatu keluaran.

Contoh : Data

#### 6) Memiliki Pengolahan Sistem (*Process*)

Suatu sistem mempunyai bagian pengolah yang akan mengubah input menjadi output.

Contoh : - CPU pada komputer

- Bagian produksi yang mengubah bahan baku menjadi barang jadi.

- Bagian akuntansi yang mengolah data transaksi menjadi laporan-laporan Keuangan.

7) Memiliki Keluaran Sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah. Keluaran ini dapat diklasifikasikan sebagai :

- Keluaran yang berguna

Contoh : Informasi yang dikeluarkan oleh komputer.

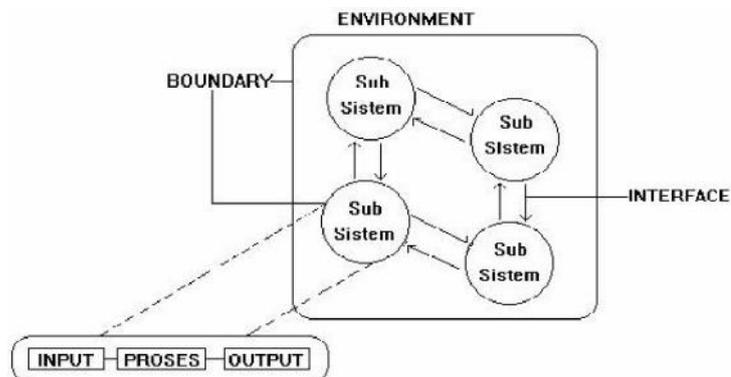
- Keluaran yang tidak berguna yang dikenal sebagai sisa pembuangan

Contoh : panas yang dikeluarkan oleh komputer.

8) Memiliki Sasaran Sistem (*Objectives*) dan tujuan sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan output yang akan dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya.

**Karakteristik sistem ini dilihat pada gambar 2.1**



**Gambar 2.1** Karakteristik suatu sistem ( Jogiyanto, 2005)

Suatu sistem dapat diklasifikasikan sebagai : (Jogiyanto, 2005)

1) Sistem Abstrak (*Abstract System*)

Sistem ini merupakan sistem yang tidak tampak secara fisik, karena hanya berupa pemikiran atau ide-ide.

Contoh : Sistem Teologi yang merupakan suatu sistem yang menggambarkan hubungan Tuhan dengan manusia.

2) Sistem Fisik (*Physical System*)

Sistem fisik merupakan sistem yang tampak secara fisik sehingga setiap makhluk dapat melihatnya.

Contoh :

- Sistem Komputer

- Sistem Akuntansi

3) Sistem Alamiah (*Natural System*)

Sistem alamiah ini adalah sistem yang terjadi dari proses proses alam dalam arti tidak dibuat oleh manusia.

Contoh : Sistem Geologi seperti sungai, pegunungan, Sistem Solar seperti *galaxy*, tata surya

4) Sistem Buatan Manusia (*Human Made System*)

Sistem ini merupakan sistem yang dirancang dan didesain oleh manusia.

Contoh : Sistem Informasi-manusia-komputer

*Man-machine system/human-machine system* Interaksi antara manusia dan mesin

5) Sistem Deterministik (*Deterministic System*)

Sistem yang beroperasi dengan tingkah laku yang dapat diramalkan disebut sistem deterministik. Interaksi antar tiap-tiap bagian dapat dideteksi, sehingga *outputnya* juga dapat diramalkan.

Contoh : Sistem Komputer

6) Sistem Tak Tentu (*Probabilistic System*)

Sistem ini adalah sistem dimana kondisi masa depannya tak dapat diramalkan karena mengandung probabilitas.

Contoh : Sistem Manusia

7) Sistem Tertutup (*Closed System*)

Sistem tertutup adalah sistem yang tidak berhubungan dan tidak dipengaruhi oleh lingkungan luarnya. Sebenarnya didunia ini tidak ada sistem yang benar-benar tertutup. Yang ada hanyalah sistem yang secara relatif tertutup (*relatively closed system*)

#### 8) Sistem Terbuka (*Open System*)

Sistem ini kebalikan dari sistem tertutup, karena sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungannya. Oleh sebab itu sistem ini harus mempunyai suatu sistem pengendalian (*control system*) yang baik, agar yang masuk hanya pengaruh-pengaruh yang baik saja.

Contoh : Sistem Kebudayaan Indonesia

## **2.2 Konsep Dasar Data dan Informasi**

### **2.2.1 Definisi Data dan Informasi**

Menurut Teguh Wahyono data adalah bahan baku informasi, didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda dan sebagainya. Data terbentuk dari karakter dapat berupa alfabet, angka, maupun simbol khusus. Sedangkan informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan (Wahyono, 2004).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Jogiyanto, 2005).

Menurut (Kadir, 2003) informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang. Definisi tersebut merupakan definisi informasi dalam pemakaian sistem informasi. Ada tiga pilar yang menentukan kualitas dari suatu informasi, yaitu (Jogiyanto, 2005):

1. Keakuratannya (*Accurate*)

Yaitu Informasi harus bebas dari suatu kesalahan-kesalahan

2. Ketepatan pada waktunya (*Timeline*)

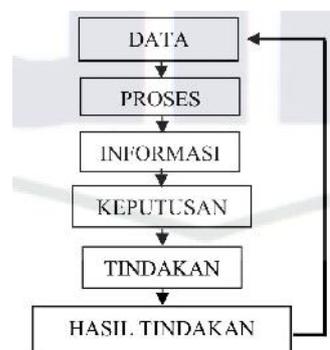
Yaitu Informasi yang datang pada penerima tidak boleh terlambat

3. Relevansinya (*Relevant*)

Yaitu berarti Informasi tersebut mempunyai manfaat untuk pemakainya.

### 2.2.2 Siklus Informasi

Pengolahan data menjadi suatu informasi pada gambar 2.2



**Gambar 2.2** Siklus Informasi (Wahyono, 2004)

## 2.3 Konsep Dasar Sistem Informasi

### 2.3.1 Definisi Sistem Informasi

Definisi atau Pengertian Sistem Informasi Secara umum merupakan kegiatan atau aktifitas yang melibatkan serangkaian proses, berisi informasi-informasi yang digunakan untuk mencapai tujuan. Pengertian Sistem Informasi menurut Robert dalam buku Jogiyanto Hartono (2005) adalah informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dalam suatu organisasi dan menyediakan pihak luar tertentu laporan-laporan yang diperlukan (Jogiyanto, 2005).

Ada beberapa sifat informasi yang harus dipahami, yaitu (Jogiyanto, 2005):

1. Mudah dicapai merupakan sifat yang menunjukkan mudahnya, cepatnya informasi dapat diperoleh, kecepatan memperoleh dapat diukur misalnya satu menit *versus* 24 jam
2. Mudah dan lengkap merupakan sifat yang menunjukkan lengkapnya informasi
3. Ketepatan merupakan sifat yang berhubungan dengan waktu yang dilalui yang lebih pendek dari pada siklus untuk mendapatkan informasi.

Sistem informasi terdiri dari komponen-komponen yang saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya. Adapun komponen-komponen sistem informasi meliputi (Jogiyanto, 2005) :

a. Blok Masukan (*input block*)

Masukan mewakili data yang masuk ke dalam sistem informasi. Masukan disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

b. Blok Model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dari model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data.

c. Blok Keluaran (*output block*)

Produk dari Sistem Informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen dan semua pemakai sistem.

d. Blok Teknologi (*technology block*)

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran serta membantu pengendalian dari sistem secara keseluruhan.

e. Blok Basis Data (*database block*)

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya. Data perlu disimpan di dalam basis data dan diorganisasikan supaya informasi yang dihasilkan berkualitas.

f. Blok Kendali (*controls block*)

Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau kesalahan yang telah terjadi dapat diatasi.

## **2.4 Konsep Dasar Penjadwalan**

### **2.4.1 Definisi Penjadwalan**

Penjadwalan sebagai kumpulan kebijaksanaan dan mekanisme di setiap operasi yang berkaitan dengan urutan kerja yang dilakukan sistem komputer. Penjadwalan memutuskan proses yang harus berjalan dan kapan dan selama berapa lama proses itu berjalan (Hariyanto, 1997).

### **2.4.2 Aktifitas Penjadwalan**

Aktifitas penjadwalan pada dasarnya dapat dibedakan menjadi lima tingkatan, yaitu (Wahyuni, 2009):

- a. *Long-range planning* (waktu 2 sampai 5 tahun).
- b. *Middle-range planning* (waktu 1 sampai 2 tahun).

- c. *Short-range planning* (waktu 3 sampai 6 bulan).
- d. Penjadwalan (waktu 2 sampai 6 minggu).
- e. Penjadwalan reaktif atau *control* (waktu 1 sampai 3 hari).

### 2.4.3 Performasi Penjadwalan

Terdapat tiga tujuan pembuatan keputusan yang umum dalam penjadwalan dan ketiganya menunjukkan ukuran dasar performansi jadwal, yaitu (Wahyuni, 2009) :

- a. Pemanfaatan sumber daya yang efisien : minimum maksimum saat selesai.
- b. Respon yang cepat terhadap permintaan konsumen : minimum rata-rata saat selesai (*completion time*), minimum rata-rata waktu tinggal (*flow time*), atau minimum rata-rata waktu tunggu (*waiting time*).
- c. Sesuai dengan batas waktu yang ditentukan : minimum rata-rata keterlambatan (*tardiness*), minimum maksimum keterlambatan, dan minimum jumlah *job* yang terlambat (*the number of tardy jobs*).

Definisi ukuran-ukuran performansi tersebut adalah (Wahyuni, 2009) :

- a. Saat selesai (*completion time*) : menunjukkan saat selesai pemrosesan.
- b. Waktu tinggal (*flow time*) : menunjukkan lamanya *job* berada dalam sistem.

- c. *Lateness* : menunjukkan perbedaan antara saat selesai dengan *due date* (mengukur kesesuaian antara jadwal dengan *due date* yang diberikan) .
- d. *Tardiness* atau *positive lateness* : menunjukkan keterlambatan yang terjadi.
- e. *Earliness* atau *negative lateness* : menunjukkan kondisi *job* selesai lebih awal dari *due date*.

## 2.5 Studi Literatur

Mempelajari buku atau jurnal hasil penelitian sejenis sebelumnya yang pernah dilakukan orang lain. Tujuannya ialah untuk mendapatkan landasan teori mengenai masalah yang akan diteliti. Teori ini merupakan pijakan bagi peneliti untuk memahami persoalan yang diteliti dengan benar dan sesuai dengan kerangka berfikir ilmiah.(Jonathan, 2006).

Studi Literatur yang dipakai dalam penelitian ini adalah sebagai berikut :

1. Studi Analisis *Rapid Application Development* Sebagai Salah Satu Alternatif Metode Pengembangan Perangkat Lunak (Noertjahyana, 2002).
2. Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS (Puspaningrum dkk, 2013).

3. Perancangan *WEB e-Commerce* dengan Metode *Rapid Application Development (RAD)* untuk Produk Unggulan Desa (Wahyuningrum, Januarita, 2014).
4. Sistem Informasi Akademik Berbasis *Sms Gateway* (Wiharto, 2011).
5. Buku “*Whole System Design An Integrated Approach to Sustainable Engineering*” (Peter Stasinopoulos, Michael H. Smith, Karlson ‘Charlie’ Hargroves and Cheryl Desha).
6. Buku “*Analisis dan Design*” (Jogiyanto, 2005).
7. *Implementation Sms Gateway In The Development Web Based Information System Schedule Seminar Thesis* (Nasution, 2012).

## **2.6 Perancangan Sistem**

### **2.6.1 Use Case Diagram**

Diagram yang menggambarkan interaksi antara sistem dengan eksternal sistem dan pengguna. Menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem (Whitten dkk, 2004). Dalam *use case diagram* memiliki pemodelan sebagai berikut:

### 1. *Use Case*

*Use case* merupakan tindakan saling terkait (*scenario*), baik otomatis maupun secara manual.

### 2. *Actor* (Pelaku)

*Actor* merupakan segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi.

### 3. *Relationship* (Hubungan)

Pada diagram *use case*, *relationship* digambarkan sebagai sebuah garis antara dua simbol. Berikut ini adalah perbedaan di antara *relationship* yang ada pada sebuah diagram *use case*:

#### a) *Association*

Relationship antara *actor* dengan *use case* dimana terjadi interaksi di antara mereka.

#### b) *Extends*

*Use case* yang terdiri dari langkah yang terekstraksi dari *use case* yang lebih kompleks untuk menyederhanakan masalah dan karena itu memperluas fungsinya.

#### c) *Uses (includes)*

Hubungan *uses* menggambarkan bahwa satu *use case* seluruhnya meliputi fungsionalitas dari *use case* lainnya.

d) *Depends On*

Terkadang suatu *use case* memiliki ketergantungan pada *use case* yang lainnya.

e) *Inheritance*

Hubungan *inheritance* terjadi ketika dua atau lebih *actor* menggunakan *use case* yang sama.

### **2.6.2 Activity Diagram**

*Activity diagram* (Fowler, 2005) adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, *activity diagram* memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah *activity diagram* mendukung *behavior paralel*. *Node* pada sebuah *activity diagram* disebut sebagai *action*, sehingga diagram tersebut menampilkan sebuah *activity* yang tersusun dari *action*.

### **2.6.3 Sequence Diagram**

*Sequence diagram* (Munawar, 2005) adalah grafik dua dimensi dimana obyek ditunjukkan dalam dimensi *horizontal*, sedangkan *lifeline* ditunjukkan dalam dimensi *vertikal*.

#### **2.6.4 Class Diagram**

*Class diagram* menurut Munawar (2005) merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute/properties*. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi.

#### **2.7 Web Browser**

*Web browser* sebagai perangkat lunak yang berguna untuk mengakses informasi *web* ataupun untuk melakukan transaksi via *web*. Beberapa contoh *browser* yang ada saat ini seperti : *Internet Explorer, Mozilla Firefox, Opera, Netscape, dan Safari*. (Kadir, 2003)

#### **2.8 Plugin**

*Plugin* yaitu sebuah program tambahan yang bisa diintegrasikan dengan memberikan fungsi-fungsi lain yang belum tersedia pada instalasi standar, bisa di sebut juga sebuah atau seperangkat program aplikasi tambahan yang berisi fungsi *script* dalam bahasa *PHP* yang memberikan fitur-fitur atau layanan yang spesifik untuk meningkatkan fungsi. Salah satunya adalah *plugin wordpress, Plugin wordpress* adalah sebuah atau seperangkat program aplikasi tambahan yang berisi

fungsi *script* dalam bahasa *PHP* yang memberikan fitur-fitur atau layanan yang spesifik untuk meningkatkan fungsi dalam penggunaan *wordpress*, yang dapat digabungkan dengan blog menggunakan akses poin dan metode yang disediakan oleh *wordpress* (<http://www.wordpress.org>).

## 2.9 Wordpress

*WordPress* adalah sebuah aplikasi sumber terbuka (*open source*) yang sangat populer digunakan sebagai mesin blog (*blog engine*). *WordPress* dibangun dengan bahasa pemrograman *PHP* dan basis data (*database*) *MySQL*. *PHP* dan *MySQL*, keduanya merupakan perangkat lunak sumber terbuka (*open source software*).

Selain sebagai *blog*, *WordPress* juga mulai digunakan sebagai sebuah *CMS* (*Content Management System*) karena kemampuannya untuk dimodifikasi dan disesuaikan dengan kebutuhan penggunanya. *WordPress* adalah penerus resmi dari *b2/cafelog* yang dikembangkan oleh Michel Valdrighi. Nama *WordPress* diusulkan oleh Christine Selleck, teman Matt Mullenweg. *WordPress* saat ini menjadi *platform content management system* (*CMS*) bagi beberapa situs *web* ternama seperti *CNN*, *Reuters*, *The New York Times*, *TechCrunch*, dan lainnya(<http://www.wordpress.org>).

## 2.10 Metode Pengujian

### Metode pengujian *Black Box*

*Black-Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program (Rosa, 2008).

Ciri-ciri *Black Box Testing* (Rosa, 2008) sebagai berikut:

- 1) *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
- 2) *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.
- 3) *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur *internal* dari sistem atau komponen yang dites, juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.

### *Equivalence Partitioning*

Merupakan metode *black box testing* yang membagi domain masukan dari suatu program ke dalam kelas-kelas data, dimana *test cases* dapat diturunkan.

*Equivalence partitioning* berdasarkan pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang diperlakukan sama (*Equivalence*) oleh komponen tersebut. Dapat juga diasumsikan bahwa masukan yang sama akan menghasilkan respon yang sama pula. Nilai tunggal pada suatu partisi ekuivalensi diasumsikan sebagai representasi dari semua nilai dalam partisi (Rosa, 2008).

**Analisa partisi pada *Equivalence Partitioning Black Box* (Rosa, 2008) :**

1. Tester menyediakan suatu model komponen yang dites yang merupakan partisi dari nilai masukan dan keluaran komponen.
2. Masukan dan keluaran dibuat dari spesifikasi dari tingkah laku komponen.
3. Partisi adalah sekumpulan nilai, yang dipilih dengan suatu cara dimana semua nilai di dalam partisi, diharapkan untuk diperlakukan dengan cara yang sama oleh komponen (seperti mempunyai proses yang sama).
4. Partisi untuk nilai *valid* dan tidak *valid* harus ditentukan.