

## **BAB II**

### **LANDASAN TEORI**

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

Sistem merupakan kumpulan dari satu kesatuan unsur-unsur yang berinteraksi dan terorganisir untuk mencapai suatu tujuan tertentu yang sama. Sistem tidak selalu identik dengan komputer, karena pada dasarnya dan pada mulanya sistem bisa dilakukan secara manual. Adapun sistem yang dilakukan dengan alat bantu bisa mempermudah, menghemat waktu, tenaga dan biaya. Dari defenisi ini dapat dirinci lebih lanjut pengertian sistem secara umum,yaitu:

1. Setiap sistem terdiri dari unsur-unsur.
2. Unsur-unsur tersebut merupakan bagian terpadu sistem yang bersangkutan.
3. Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem.

##### **2.1.1 Sistem Pendukung Keputusan**

Menurut Nofriansyah dalam Budi Sudrajat (2018:203), “Sistem Pendukung Keputusan biasanya dibangun untuk mendukung solusi atau masalah atau untuk suatu peluang”.

Menurut Kusriani dalam Rinianty dan Sukardi (2018:49), “Sistem Pendukung Keputusan adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem pendukung keputusan digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur”.

Keputusan dibuat untuk memecahkan masalah. Keputusan merupakan tindakan atau rangkaian tindakan yang harus diikuti untuk memecahkan suatu masalah (Deni,2013:137). Tindakan tersebut bisa berupa pengurangan sesuatu untuk menghindari risiko dari suatu hal atau pemanfaatan suatu kesempatan.

SPK atau *Decision Support System* (DSS) dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka (Deni,2013:145). SPK ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau pada keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma. Sebagai istilah umum SPK digunakan untuk menggambarkan semua sistem terkomputerisasi yang mendukung pengambilan keputusan pada suatu organisasi. Tujuan utama yang harus dicapai oleh SPK yaitu;

1. Dapat membantu manajer dalam membuat keputusan saat memecahkan berbagai masalah semi terstruktur.
2. Dapat mendukung penilaian yang dilakukan oleh manajer dan tidak mencoba menggantikannya.
3. Dapat meningkatkan efektivitas pengambilan keputusan yang dilakukan oleh manajer dan bukan meningkatkan efisiensinya (Peter dalam Deni,2013:144-145).

## **2.2 Bantuan Sosial**

Bantuan Sosial adalah bantuan berupa uang, barang, atau jasa kepada seseorang, keluarga, kelompok atau masyarakat miskin, tidak mampu, dan/atau rentan terhadap risiko sosial. Risiko Sosial adalah kejadian atau peristiwa yang dapat menimbulkan potensi terjadinya kerentanan sosial yang ditanggung oleh

seseorang, keluarga, kelompok, dan/atau masyarakat sebagai dampak krisis sosial, krisis ekonomi, krisis politik, fenomena alam, dan bencana yang jika tidak diberikan Bantuan Sosial akan semakin terpuruk dan tidak dapat hidup dalam kondisi wajar.

### **2.2.1 Penerima Bantuan Sosial**

Penerima Bantuan Sosial adalah seseorang, keluarga, kelompok atau masyarakat miskin, tidak mampu, dan/atau rentan terhadap risiko sosial.

Syarat – syarat penerima bantuan sembako pandemi Covid-19 berdasarkan keputusan Menteri Sosial Republik Indonesia nomor: 146 / HUK / 2013 tentang Penetapan Kriteria dan Pendataan Fakir Miskin dan Orang Tidak Mampu, terdiri atas 14 kriteria Kemiskinan :

1. Luas lantai bangunan tempat tinggal kurang dari 8 meter per orang.
2. Jenis lantai tempat tinggal terbuat dari tanah/bambu/kayu murahan.
3. Jenis dinding tempat tinggal terbuat dari bambu, kayu, tembok tanpa diplester.
4. Tidak memiliki fasilitas buang air besar/bersama-sama dengan rumah tangga lain.
5. Sumber penerangan rumah tangga tidak menggunakan listrik.
6. Sumber air minum berasal dari sumur.
7. Bahan bakar untuk memasak sehari-hari menggunakan kayu bakar/arang/minyak tanah.
8. Hanya mengkonsumsi daging/susu/ayam dalam satu kali seminggu.
9. Hanya membeli satu stel pakaian baru dalam setahun.
10. Hanya sanggup makan sebanyak satu atau dua kali dalam sehari.

11. Tidak sanggup membayar biaya pengobatan di puskesmas/klinik.
12. Sumber penghasilan kepala rumah tangga adalah petani/buruh tani/nelayan/buruh bangunan/buruh perkebunan atau pekerja lainnya dengan penghasilan dibawah 600.000 perbulan.
13. Pendidikan tertinggi kepala keluarga adalah tidak sekolah/tidak tamat SD/ tamat SD.
14. Tidak memiliki tabungan/barang yang mudah dijual dengan minimal 500.000.

### **2.2.2 Pandemi Covid-19**

Pandemi COVID-19 adalah peristiwa menyebarnya Penyakit koronavirus 2019 (Bahasa Inggris: *Coronavirus disease 2019*, disingkat **COVID-19**) di seluruh dunia untuk semua Negara. Penyakit ini disebabkan oleh koronavirus jenis baru yang diberi nama SARS-CoV-2. Wabah COVID-19 pertama kali dideteksi di Kota Wuhan, Provinsi Hubei, Tiongkok pada tanggal 1 Desember 2019, dan ditetapkan sebagai pandemi oleh Organisasi Kesehatan Dunia (WHO) pada tanggal 11 Maret 2020.

### **2.3 Metode Fuzzy Multi Attribute Decision Making (Fuzzy MADM)**

Masalah ketidaktepatan dan ketidakpastian bisa disebabkan oleh beberapa hal, seperti; informasi yang tidak dapat dihitung, informasi yang tidak lengkap, informasi yang tidak jelas serta pengabaian parsial. Dalam menetapkan sebuah kriteria atau variabel untuk sebuah sistem seleksi, tidak jarang didapati beberapa variabel yang juga sulit untuk diukur secara jelas. Untuk mengatasi masalah tersebut, maka beberapa penelitian tentang penggunaan metode *Fuzzy* MADM mulai banyak dilakukan, dan terbukti memiliki kinerja yang sangat baik.

*Fuzzy MADM* adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu (Ryza, 2014 :28) . Inti dari FMADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan.

Untuk menyelesaikan masalah FMADM, dibutuhkan 2 tahap, yaitu; (Ryza, 2014:29)

- 1 Membuat rating pada setiap alternatif berdasarkan agregasi derajat kecocokan pada semua kriteria.
- 2 Meranking semua alternatif untuk mendapatkan alternatif terbaik. Ada 2 cara yang dapat digunakan dalam proses perankingan, yaitu melalui *defuzzy* atau melalui relasi *preferensi fuzzy*. Metode *defuzzy* dilakukan dengan pertama-tama membuat bentuk *crisp* dari bilangan *fuzzy*, proses perankingan didasarkan atas bilangan *crisp* tersebut; model ini memang mudah untuk diimplementasikan, namun kita sangat dimungkinkan untuk kehilangan beberapa informasi terutama yang menyangkut ketidakpastian. Penggunaan relasi preferensi *fuzzy* lebih menjamin ketidakpastian yang melekat pada bilangan *fuzzy* hingga proses perankingan

### **2.3.1 Fuzzy MADM Model Yager**

Fuzzy FMADM model Yager ini merupakan bentuk standar dari fuzzy MADM (Kusumadewi, 2006:137). Langkah-langkah penyelesaian untuk model ini adalah;

1. Tetapkan matriks perbandingan berpasangan antar atribut,  $M$ ,

berdasarkan prosedur hirarki Saaty.

2. Menentukan bobot tingkat kepentingan yang konsisten untuk setiap atribut berdasarkan metode eigenvector dari Saaty.
3. Hitung nilai;  $C_j(x_i)^{w_j}$ .
4. Tentukan interseksi dari semua  $C_j(x_i)^{w_j}$ .
5. Pilih  $x_i$  dengan derajat keanggotaan terbesar sebagai alternatif optimal.

### 2.3.2 Model MADM Klasik Untuk Penyelesaian Fuzzy MADM

Mekanisme menyelesaikan masalah *fuzzy* MADM dapat dengan mengaplikasikan metode MADM klasik untuk melakukan perangkingan, setelah terlebih dahulu dilakukan konversi data *fuzzy* ke data *crisps* (Chen,1992 dalam Kusumadewi,2006:145). Apabila data *fuzzy* diberikan dalam bentuk linguistik, maka data tersebut harus dikonversi terlebih dahulu ke bentuk bilangan Fuzzy, baru kemudian dikonversi lagi ke bilangan *crisps* (Kusumadewi,2006:145).

Model MADM yang digunakan penulis dalam menyelesaikan penelitian ini adalah dengan model SAW. Adapun langkah-langkah penyelesaian model ini adalah (Apriansyah dan Dinna, 2011:17-18);

1. Menentukan kriteria yang dijadikan acuan pengambilan keputusan.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (Z), kemudian melakukan normalisasi metriks berdasarkan persamaan yang disesuaikan dengan jenis atribut sehingga diperoleh matriks ternormalisasi R.

$$N = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1j} \\ \vdots & & & \vdots \\ r_{i1} & r_{i2} & \dots & r_{ij} \end{bmatrix}$$

Ketentuan Normalisasi:

- a. Dikatakan atribut keuntungan apabila atribut banyak memberikan keuntungan bagi pengambil keputusan, sedangkan atribut biaya yang banyak memerlukan pengeluaran jika nilainya semakin besar bagi pengambil keputusan.
  - b. Apabila berupa atribut keunganan maka nilai ( $x_{ij}$ ) dari setiap kolom atribut dibagi dengan nilai ( $\text{Max } x_{ij}$ ) dari setiap kolom, sedangkan untuk atribut biaya, nilai ( $\text{Min } x_{ij}$ ) dari setiap kolom atribut dibagi dengan nilai ( $x_{ij}$ ) setiap kolom
4. Hasil akhir ( $V_i$ ) diperoleh dari proses perangkinayan yaitu penjumlahan dari perkalian matrik ternormalisasi  $N$  dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik sebagai solusi.

## **2.4 Software Untuk Membangun Aplikasi Web**

Untuk membangun aplikasi web pada perancangan system penilaian teknisi terbaik pada PT. Telekomunikasi Indonesia Tbk, menggunakan metode ANP penulis menggunakan software sebagai berikut:

1. Xampp
2. Text Editor
3. Mozila Firefox

### **2.4.1 Aplikasi Website Dan kelebihanannya**

Aplikasi web adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti internet atau intranet. Ia juga merupakan suatu

aplikasi perangkat lunak computer yang dikodekan dalam Bahasa yang didukung penjelajah web (seperti ASP, Perl, Java, Java Script, PHP, Python, Ruby, dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

Kelebihan aplikasi web adalah akses informasi mudah, setup server lebih mudah, informasi mudah didistribusikan, bebas platform.

## **2.4.2 Script Pada Aplikasi Web**

### **2.4.2.1 HTML**

HTML adalah singkatan dari Hyper-Text Markup Language. HTML merupakan tulang punggung aplikasi web. Sekali pun sejumlah teknologi pendukung aplikasi web telah muncul, tetapi peran HTML tak tergantikan. Kehadiran PHP merupakan pelengkap yang memungkinkan aplikasi web tidak lagi bersifat statis, melainkan dinamis. Perubahan-perubahan pada sisi data tidak perlu membuat aplikasi diubah. Cara seperti itu terutama dilakukan pada sisi server. Artinya, kode seperti PHP berjalan di server. Selain cara seperti itu, penambahan kode pada HTML juga bisa dilakukan pada sisi klien. (Abdul Kadir, 2011).

### **2.4.2.2 CSS**

*Cascading Style Sheet* (CSS) merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. Sama halnya styles dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa HTML dan XHTML.

### **2.4.2.3 Java Script**

Java Script adalah kode untuk menyusun halaman web yang memungkinkan dijalankan di sisi klien (pada browser yang digunakan pemakai). Karena dijalankan di sisi klien, maka JavaScript digunakan untuk membuat tampilan lebih dinamis. (Abdul Kadir, 2011).

### **2.4.2.4 PHP**

PHP adalah teknologi yang diperkenalkan tahun 1994 oleh Rasmus Lerdorf. Beberapa versi awal yang tidak dipublikasikan digunakan pada situs pribadinya untuk mencatat siapa saja yang mengakses daftar riwayat hidup onlinennya. Versi pertama digunakan oleh pihak lain pada awal tahun 1995 dan dikenal sebagai Personal Home Page Tools. Terkandung didalamnya sebuah parser engine (mesin pengurai) yang sangat disederhanakan, yang hanya mampu mengolah macro khusus dan beberapa utilitas yang sering digunakan pada pembuatan home page, seperti buku tamu, pencacah, dan hal semacamnya. Parser tersebut ditulis ulang pada pertengahan 1995 dan dinamakan PHP/FI Versiaon 2. FI(Form Interpreter) sendiri berasal dari kode lain yang ditulis juga oleh Rasmus, yang menterjemahkan HTML dari data. Ia menggabungkan script Personal Home Page Tools dengan Form Interpreter dan menambahkan dukungan terhadap server database yang menggunakan format mSQL sehingga lahirlah PHP/FI. PHP/FI tumbuh dengan pesat, dan orang-orang mulai menyiapkan kode-kode programnya supaya bisa didukung oleh PHP. (Abdul Kadir, 2011).

### **2.4.3 Database**

Basis data merupakan suatu kumpulan data yang berhubungan secara logis

dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi (Indrajani,2014:2). Basis data digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer. Untuk mengelolah database diperlukan suatu perangkat lunak yang disebut DBMS (*Database Management System*).

#### **2.4.3.1 Konsep DBMS (*Database Management System*).**

*Database Management System* (DBMS) merupakan suatu perangkat lunak yang ditujukan untuk menangani penciptaan, pemeliharaan, dan pengendalian akses data (Idam, 2011: 37) . DBMS terdiri dari Database dan set program pengolah untuk menambah data, mengolah data, mengambil dan membaca data. Database sendiri merupakan kumpulan data yang pada umumnya menggambarkan aktifitas dari pelakunnya (*user*). Dalam penelitian ini, penulis menggunakan DBMS berupa MySQL sebagai implementasi perancangan database.

#### **2.4.3.2 MySQL**

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *Database Management System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia (Solichin, 2010:8). Beberapa alasan menggunakan MySQL sebagai *server database* dalam pengembangan sistem adalah;

1. Fleksibel, MySQL memiliki fleksibilitasterhadap terknologi yang akan digunakan sebagai pengembangan aplikasi.
2. Performa tinggi, sebagai database dari beberapa aplikasi web yang memiliki *traffic* yang tinggi.

3. Lintas *Platform*, MySQL dapat digunakan pada sistem operasi yang beragam.
4. Gratis, MySQL dapat digunakan secara gratis.
5. Proteksi data yang handal, MySQL menyediakan mekanisme yang *powerfull* untuk menangani hal ini, yaitu dengan menyediakan fasilitas manajemen *user*, enkripsi data dan lain sebagainya. Komunitas luas, Hal ini berguna jika kita menemui suatu permasalahan dalam pengolahan data menggunakan MySQL (Budi Raharjo, 2011: 304 dalam Dian, 2014:31).

## 2.5 Pemodelan Sistem dengan UML (Unified Modeling Language)

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standar. UML memiliki sintakas dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara peangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan system yang ada, proses-proses dan organisasinya (Prabowo Pudjo Widodo, 2011).

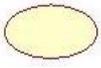
## 2.5.1 Usecase Diagram

### 1. Tabel Komponen *Use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor. Dimana aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan – persyaratan yang harus dipenuhi sistem dari pandangan pemakai.

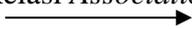
Beberapa komponen dalam *use case* antara lain :

**Tabel 2.1** Komponen *Use Case Diagram* (Hamilton dan Russel, 2006)

Komponen Use case	Penjelasan
 Aktor	Merupakan sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi dengan sistem.
 <i>Use case</i>	Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Ada beberapa relasi yang terdapat dalam use case, antaran lain :

**Tabel 2.2** Relasi Tabel *Use Case* (Hamilton dan Russel, 2006)

Relasi <i>Use case</i>	Penjelasan
Relasi <i>Association</i> 	<i>Association</i> , menghubungkan link antar element
Relasi  <i>Generalization</i>	<i>Generalization</i> disebut juga <i>inheritance</i> (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.

Relasi <i>Depedency</i> ----->	<i>Dependency</i> , sebuah elemen bergantung dalam beberapa cara ke elemen lainnya.
-----------------------------------	---

Tipe relasi / *stereotype* yang mungkin terjadi pada *use case diagram* :

**Tabel 2.3** Tabel *Stereotype* yang mungkin terjadi (Hamilton dan Russel, 2006)

Relasi/ <i>Stereotype</i>	Penjelasan
<b>&lt;&lt;include&gt;&gt;</b>	Kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya.
<b>&lt;&lt;extends&gt;&gt;</b>	Kelakuan yang hanya berjalan dibawah kondisi tertentu seperti menggerakkan alarm.
<b>&lt;&lt;Communicates&gt;&gt;</b>	Ditambahkan untuk asosiasi yang mungkin menunjukkan asosiasinya adalah <i>communicates association</i> .

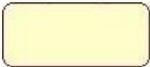
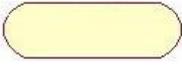
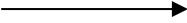
## 2.5.2 Activity Diagram

### 1. Tabel Activity Diagram

Diagram aktivitas menggambarkan aliran fungsionalitas sistem. Dapat juga digunakan untuk menggambarkan aliran kejadian dalam *use case*. Berikut beberapa komponen yang terdapat dalam *Activity Diagram* :

**Tabel 2.4** *Activity Diagram* (Hamilton dan Russel, 2006)

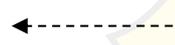
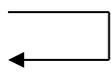
<i>Activity Diagram</i>	Penjelasan
 <i>Start State</i>	<i>Start State</i> , Sebagai tanda awal proses dari <i>activity diagram</i> .

 <i>State</i>	<i>State</i> , Berfungsi menampung event dalam <i>activity diagram</i> .
 <i>Activity</i>	<i>Activity</i> , Memiliki fungsi yang sama dengan <i>state</i> . Menampung event atau aktifitas pada proses sistem.
 <i>State Transition</i>	<i>State Transition</i> , Berfungsi untuk menunjukkan aliran atau urutan dari event atau aktifitas pada diagram.
 <i>Transition to self</i>	<i>Transition to Self</i> , Berfungsi untuk menunjukkan transisi sebuah event yang mengarah ke event itu sendiri.
 <i>Horizontal Synchronization</i>	<i>Horizontal Synchronization</i> , Berfungsi untuk mengsinkronisasikan 2 cabang event yang posisinya horizontal.
 <i>Decision</i>	<i>Decision</i> , Digunakan ketika terjadi pemilihan 2 kondisi event pada diagram.
 End State	<i>End State</i> , Sebagai tanda akhir dari <i>activity diagram</i> .

### 2.5.3 Sequence Diagram

Diagram sekuensial umumnya digunakan untuk menggambarkan suatu skenario atau urutan langkah – langkah yang dilakukan baik oleh *actor* maupun sistem yang merupakan respon dari sebuah kejadian untuk mendapatkan hasil atau output. Berikut adalah beberapa komponen yang terdapat dalam *sequence diagram*:

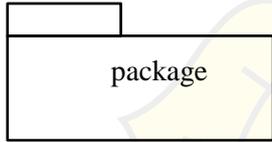
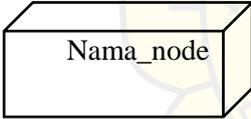
**Tabel 2.5** *Sequence Diagram* (Hamilton dan Russel, 2006)

<i>Sequence Diagram</i>	Penjelasan
 <i>Actor</i>	<i>Actor</i> , Menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem.
 <i>Return Message</i>	<i>Return Message</i> , menggambarkan pesan atau hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.
 <i>Lifeline</i>	<i>Lifeline</i> , Eksekusi obyek selama <i>sequence</i> ( <i>message</i> dikirim atau diterima dan aktifasinya)
 <i>Message to Self</i>	<i>Message to Self</i> , Menggambarkan pesan atau hubungan obyek itu sendiri yang menunjukkan urutan kejadian yang terjadi.
 <i>Object Message</i>	<i>Object Message</i> , Menggambarkan pesan atau hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.

### 2.5.4 Deployment Diagram

Deployment diagram adalah diagram yang digunakan memetakan software ke processing node. Menunjukkan konfigurasi elemen pemroses pada saat run time dan software yang ada di dalamnya. Diagram ini adalah salah satu diagram paling penting dalam tingkat implementasi perangkat lunak. Berikut adalah beberapa komponen yang terdapat dalam *deployment diagram* :

**Tabel 2.6** *Deployment Diagram*

<i>Deployment Diagram</i>	Penjelasan
<p><i>Package</i></p> 	<p><i>Package</i>, merupakan sebuah bungkus dari satu atau lebih node.</p>
<p><i>Node</i></p> 	<p>Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam node disertakan komponen untuk mengkonsistensikan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.</p>
<p>Kebergantungan/<i>dependency</i></p> 	<p>Kebergantungan antar node, arah panah mengarah pada node yang dipakai.</p>
<p><i>Link</i></p> 	<p>Relasi antar node</p>