

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Pengambilan 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 manipulasi data. Sistem pendukung keputusan digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur”.

Berdasarkan beberapa pendapat yang dikemukakan di atas maka dapat disimpulkan bahwa Sistem Pendukung Keputusan adalah sebuah sistem berbasis komputer yang dibangun bertujuan untuk membantu pihak manajemen dalam pengambilan keputusan.

2.1.1 Karakteristik Sistem Pengambilan Keputusan

Menurut Nofriansyah dalam Ardhi Bagus Primahudi (2016:59), karakteristik sistem pendukung keputusan ada enam, sebagai berikut:

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.
2. Adanya interface manusia atau mesin dimana manusia tetap memegang control proses pengambilan keputusan.

3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur , semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
5. Memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai suatu kesatuan sistem.
6. Memiliki dua komponen utama yaitu data dan model.

2.1.2 Komponen Sistem Pengambilan Keputusan

Adapun komponen dari SPK adalah sebagai berikut :

1. *Data Management*

Termasuk database, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh software yang disebut Database Management System (DBMS).

2. *Model Management*

Melibatkan model finansial, statistikal, management science, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan sistem suatu kemampuan analitis, dan manajemen software yang dibutuhkan.

3. *Communication*

User dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antar muka.

4. *Knowledge Management*

Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang terdiri sendiri.

2.1.3 Tujuan Sistem Pengambilan Keputusan

Menurut Nofriansyah (2014:4), Sistem Pendukung Keputusan memiliki beberapa tujuan, yaitu:

1. Membantu dalam pengambilan keputusan atas masalah yang terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektifitas keputusan yang diambil lebih dari pada perbaikan efisiensinya.
4. Kecepatan komputasi komputer memungkinkan para pengambil keputusan untuk banyak melakukan komputasi secara cepat dengan biaya rendah.
5. Peningkatan produktivitas membangun suatu kelompok pengambilan keputusan, terutama para pakar, bisa sangat mahal.

2.2 Pemilihan Teknisi Terbaik

Aset paling penting yang harus dimiliki oleh organisasi atau perusahaan dan harus diperhatikan dalam manajemen adalah tenaga kerja atau manusia (sumber daya manusia). Manajemen sumber daya manusia merupakan kegiatan yang dilaksanakan agar sumber daya manusia dalam organisasi dapat didayagunakan secara efektif dan efisien guna mencapai berbagai tujuan.

Kegiatan manajemen sumber daya manusia adalah kegiatan untuk menyediakan dan mempertahankan tenaga kerja yang efektif dan berkualitas bagi organisasi dan perusahaan. Salah satu kegiatan manajemen sumber daya manusia yang dilakukan adalah seleksi (pemilihan) karyawan.

Pemilihan karyawan pada penelitian ini ditunjukkan untuk karyawan teknisi. Untuk mengetahui informasi kualitas dan kemampuan teknisi dengan cara mengukur prestasi teknisi dalam bekerja. Teknisi dipilih berdasarkan penilaian prestasi teknisi yang terbaik. Penilaian dilakukan oleh tim penilai dari perusahaan. Tim penilai biasanya terdiri dari manajer, kepala bagian (instansi), atau orang yang ditunjuk oleh perusahaan untuk menilai karyawan.

Perusahaan menerapkan pemilihan teknisi terbaik (berprestasi) untuk meningkatkan motivasi teknisi dalam bekerja. Bagi teknisi, penilaian tersebut berperan sebagai umpan balik tentang berbagai hal seperti kemampuan, kelebihan, kekurangan, dan potensi yang pada gilirannya bermanfaat untuk menunjukkan tujuan, jalur, rencana, dan pengembangan karir.

Teknisi yang terpilih menjadi teknisi terbaik akan mendapatkan penghargaan (awards) dari perusahaan. Pemberian penghargaan teknisi terbaik secara periodik dikenal juga dengan istilah *Employee of the Month (EOM)*. Penghargaan yang diberikan perusahaan dapat berupa penambahan gaji atau kenaikan jabatan.

Dalam pembuatan sistem ini, contoh kasus yang diambil adalah pada PT. Telekomunikasi Indonesia, Tbk yaitu dalam pemilihan teknisi terbaik yang dilakukan setiap 6 (enam) bulan. Dalam proses pemilihan teknisi terbaik, penilaian dilakukan oleh leader team teknisi.

Selama 6 (enam) bulan leader teknisi melakukan pengamatan kinerja teknisi, kriteria yang dinilai oleh leader team terdiri dari:

1. Character Integritas

Penilaian ini dilakukan berdasarkan sikap kejujuran dari masing-masing teknisi sesuai antara perkataan dengan perilaku dan mampu memberikan pengaruh yang baik di lingkungan kerja.

2. Competency Skill

Penilaian ini dilakukan berdasarkan kemampuan teknis maupun administrative yang diperlukan untuk penyelesaian pekerjaan.

3. Colaboration Team Work

Penilaian ini dilakukan berdasarkan kemampuan menjalin kerjasama dengan rekan kerja dan bekerja dalam kelompok.

4. Contribution Target.

Penilaian ini dilakukan berdasarkan pencapaian hasil/target yang menjadi tanggung jawab.

Pemilihan teknisi terbaik ini diharapkan dapat memacu kinerja teknisi dalam menjalankan tugas-tugas atau pekerjaan yang diberikan dan dapat memberikan hasil yang terbaik dalam setiap pekerjaannya.

2.3 Analytic Network Process (ANP)

Metode Analytic Network Process (ANP) merupakan pengembangan dari metode Analytical Hierarchy Process (AHP). Metode AHP (Analytical Hierarchy Process) merupakan sebuah hirarki fungsional dengan input utamanya persepsi manusia. Suatu masalah yang kompleks dan tidak terstruktur dipecah ke dalam kelompok-kelompok kemudian diatur menjadi suatu bentuk hierarki (Saaty, 1998).

AHP dapat menata bagian atau variabel ini dalam suatu susunan hirarki, memberi nilai numerik pada pertimbangan subjektif tentang pentingnya tiap variabel dan mensintesis berbagai pertimbangan untuk menetapkan variabel mana yang memiliki prioritas paling tinggi dan bertindak untuk mempengaruhi hasil pada situasi tersebut.

Metode AHP ini membantu memecahkan persoalan yang kompleks dengan menstruktur suatu hirarki kriteria dan menarik berbagai pertimbangan guna mengembangkan bobot atau prioritas. Metode ini juga menggabungkan kekuatan perasaan dan logika yang bersangkutan pada berbagai persoalan, lalu mensintesis berbagai pertimbangan yang beragam menjadi hasil yang cocok dengan perkiraan kita secara intuitif sebagaimana yang dipresentasikan pada pertimbangan yang telah dibuat (Saaty, 1994). Menurut Saaty, ada beberapa prinsip dalam memecahkan persoalan dengan AHP, yaitu prinsip menyusun hirarki (Decompostion), prinsip menentukan prioritas (Comparative Judgement), dan prinsip konsistensi logis (Logical Consistency).

Perbedaan AHP dan ANP berawal dari struktur hierarki yang tidak berlaku untuk ANP. Perbedaan ANP dengan AHP dapat dilihat pada tabel dibawah ini.

Tabel 2.1 Perbedaan AHP dengan ANP

NO	PERBEDAAN	AHP	ANP
1	Kerangka	Hierarki	Jaringan
2	Hubungan	Dependensi	Dependensi dan Feedback
3	Prediksi	Kurang Akurat	Lebih Akurat
4	Komparasi	Preferensi/Kepentingan Lebih Subjektif	Pengaruh Lebih Objektif

5	Hasil	Matriks, Eigenvector Kurang Stabil	Supermatriks Lebih Stabil
6	Cakupan	Sempit/Terbatas	Luas

Perbedaan **pertama** terletak pada struktur kerangka model yang berbentuk hierarki pada AHP dan berbentuk jaringan pada ANP. Hal ini membuat ANP dapat diaplikasikan lebih luas dari ANP. Bentuk jaringan ANP juga bisa sangat bervariasi dan lebih dapat mencerminkan permasalahan seperti keadaan yang sesungguhnya.

Kedua, dalam struktur hierarki hanya ada dependensi level yang lebih rendah kepada level yang lebih tinggi, sementara dalam struktur jaringan terdapat juga *feedback*. Dengan *feedback* alternatif dapat dependen terhadap kriteria, seperti pada hierarki, tetapi dapat pula dependen satu sama lain. Sementara kriteria sendiri dapat dependen pada alternatif dan pada satu sama lain.

Ketiga, *feedback* memperbaiki prioritas yang dihasilkan dari penilaian, dan membuat prediksi lebih akurat.

Keempat, untuk melakukan komparasi dalam AHP seseorang bertanya mana yang lebih disukai atau lebih penting? Keduanya lebih kurang subyektif dan personal. Sementara itu untuk komparasi dalam ANP seseorang bertanya mana yang lebih berpengaruh? Hal ini membutuhkan observasi faktual dan pengetahuan sehingga menghasilkan jawaban valid yang lebih obyektif.

Kelima, hasil AHP adalah matriks dan eigenvector yang menunjukkan skala prioritas, sedangkan hasil ANP berupa supermatriks skala prioritas yang lebih stabil karena adanya *feedback*. Kestabilan hasil ANP telah dibuktikan oleh Iwan J. Azis dalam papernya (Azis, 2003), dimana masalah Trans Sumatra Highway dianalisis

dengan menggunakan AHP dan ANP. Dari analisa sensitivitas yang dilakukan diperoleh kesimpulan bahwa hasil ANP lebih stabil dan *robust* dari pada hasil AHP.

Keenam, Cakupan AHP terbatas pada struktur yang hierarkis, sedangkan cakupan ANP meluas tak terbatas. AHP dengan asumsi-asumsi dependensinya tentang cluster dan elemen merupakan kasus khusus ANP.

Metode ANP mampu memperbaiki kelemahan AHP berupa kemampuan mengakomodasi keterkaitan antar kriteria atau alternatif (Saaty, 2004). Pada AHP semua kriteria yang ada harus saling berkaitan secara hirarki, sedangkan pada ANP semua kriteria bisa berkaitan dan tidak berkaitan, jika ada kriteria yang tidak berkaitan maka kriteria itu bernilai 0.

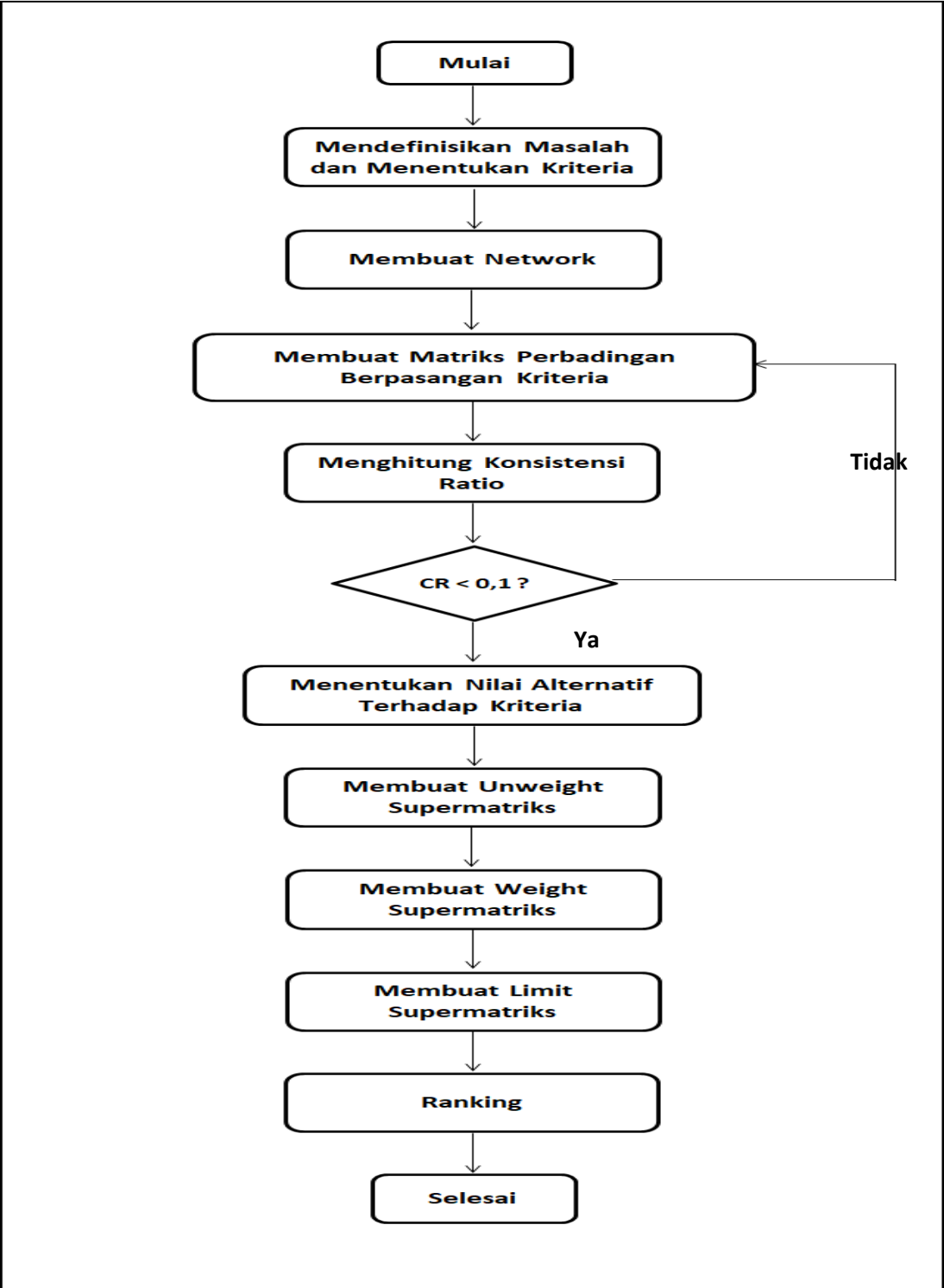
Keterkaitan pada metode ANP ada 2 jenis yaitu keterkaitan dalam satu set cluster (inner dependence) dan keterkaitan antar cluster yang berbeda (outer dependence). Adanya keterkaitan tersebut menyebabkan metode ANP lebih kompleks dibanding metode AHP.

2.3.1 Langkah – Langkah Metode ANP

Secara umum langkah-langkah yang harus dilakukan dalam menggunakan ANP adalah sebagai berikut:

1. Mendefinisikan masalah dan menentukan kriteria solusi yang diinginkan.
2. Menentukan pembobotan komponen dari sudut pandang manajerial.
3. Membuat matriks perbandingan berpasangan yang menggambarkan kontribusi atau pengaruh setiap elemen atas setiap kriteria. Perbandingan dilakukan berdasarkan penilaian dari pengambil keputusan dengan menilai tingkat kepentingan suatu elemen.

4. Setelah mengumpulkan semua data perbandingan berpasangan dan memasukkan nilai-nilai kebalikannya serta nilai satu di sepanjang diagonal utama, prioritas masing-masing kriteria dicari dan konsistensi diuji.
5. Menentukan eigen vector dari matriks yang telah dibuat pada langkah ketiga.
6. Mengulangi langkah 3, 4, dan 5 untuk semua kriteria.
7. Membuat unweighted supermatriks dengan cara memasukkan semua eigen vector yang telah dihitung pada langkah 5 ke dalam sebuah supermatriks.
8. Membuat weighted supermatriks dengan cara melakukan perkalian setiap isi unweighted supermatriks terhadap matriks perbandingan kriteria (cluster matrix).
9. Membuat limiting supermatriks dengan cara mengangkat supermatriks secara terus menerus hingga angka disetiap kolom dalam satu baris sama besar.
10. Ambil nilai dari alternatif yang dibandingkan setelah dilakukan limiting supermatriks.



Gambar 2.1 Flowchart Analisa Model ANP

2.3.1.1 Mendefinisikan Masalah

Mendefinisikan masalah yang dihadapi dan menentukan solusi yang diinginkan. Masalahnya harus dinyatakan dengan jelas dan menguraikannya menjadi system rasional seperti jaringan.

2.3.1.2 Menentukan Pembobotan Komponen

Pembobotan komponen atau kriteria dilakukan oleh pihak perusahaan yang bersangkutan.

2.3.1.3 Membuat Matriks Perbandingan Berpasangan

Menyusun matriks perbandingan berpasangan merupakan salah satu bagian yang penting dan perlu ketelitian didalamnya. Pada bagian ini akan ditentukan skala kepentingan suatu elemen terhadap elemen lainnya. Langkah pertama dalam menyusun perbandingan berpasangan, yaitu membandingkan dalam bentuk berpasangan seluruh untuk setiap sub sistem hirarki. Perbandingan tersebut kemudian ditransformasikan dalam bentuk matriks untuk maksud analisis numerik, yaitu matriks $n \times n$.

Misalkan terdapat suatu sub sistem hirarki dengan suatu kriteria A dan sejumlah elemen dibawahnya. B1 sampai Bn. Perbandingan antar elemen untuk sub sistem hirarki itu dapat dibuat dalam bentuk matriks $n \times n$. Matriks ini disebut matriks perbandingan berpasangan.

Tabel 2.2 Matriks Perbandingan Berpasangan (Saaty, 1999)

A	B ₁	B ₂	B ₃	...	B _n
B ₁	B ₁₁	B ₁₂	B ₁₃	...	B _{1n}
B ₂	B ₂₁	B ₂₂	B ₂₃	...	B _{2n}
B ₃	B ₃₁	B ₃₂	B ₃₃	...	B _{3n}
...
B _n	B _{n1}	B _{n2}	B _{n3}	...	B _{nn}

Nilai b_{ij} adalah nilai perbandingan elemen B_i terhadap B_j yang menyatakan hubungan :

1. Seberapa jauh tingkat kepentingan B_i bila dibandingkan dengan B_j , atau
2. Seberapa besar kontribusi B_i terhadap kriteria A dibandingkan dengan B_j , atau
3. Seberapa jauh dominasi B_i dibandingkan dengan B_j , atau
4. Seberapa banyak sifat kriteria A terdapat pada B_i dibandingkan dengan B_j

Bila diketahui nilai b_{ij} maka secara teoritis nilai $b_{ij} = 1 / b_{ji}$, sedangkan b_{ij} dalam situasi $i = j$ adalah mutlak.

Nilai numerik yang digunakan untuk perbandingan di atas diperoleh dari skala perbandingan yang dibuat Saaty dan Vargas. Berdasarkan tabel di bawah ini kita dapat menentukan skala perbandingan antar elemen dalam proses pengambilan keputusan.

Tabel 2.3 Penilaian Perbandingan Berpasangan (Saaty, 1999)

Tingkat Kepentingan	Definisi	Keterangan
1	Kedua elemen sama penting/disukai	Elemen A1 dan A2 sama disukai/penting
3	Elemen yang satu sedikit Lebih penting/disukai daripada elemen lainnya	Elemen A1 cukup disukai/penting dibanding elemen A2
5	Elemen yang satu lebih penting/disukai daripada elemen lainnya	Elemen A1 lebih disukai/penting dibanding elemen A2
7	Satu elemen lebih penting / disukai daripada elemen lainnya	Elemen A1 sangat disukai / penting dibanding elemen A2
9	Satu elemen mutlak penting / disukai daripada elemennlainnya	Elemen A1 mutlak disukai / penting dibanding elemen A2
2,4,6,8	Nilai tengah diantara dua penilaian yang berdampingan	Nilai ini diberikan jika diperlukan kompromi /ragu-ragu dalam memilih skala. Misal : memilih sangat disukai & mutlak disuka
Kebalikan	Bila elemen ke-ij pada faktor I	Misal : Jika A1 dibanding A2

	mendapat nilai nilai x maka elemen ke-ji pada faktor ke-j mendapat nilai 1/x	katakanlah mempunyai skala 7,maka A2 dibanding A1 mempunyai skala 1/7
--	---	---

2.3.1.4 Menentukan Nilai *Eigenvector*

Setelah dilakukan matriks perbandingan berpasangan, selanjutnya menentukan nilai eigen dari matriks tersebut. Perhitungan eigenvector dengan cara menjumlahkan nilai setiap kolom dari matriks kemudian membagi setiap nilai sel kolom dengan total kolom dan menjumlahkan nilai-nilai dari setiap baris dan dibagi n. Nilai eigen dihitung dengan langkah-langkah sebagai berikut :

$$X = \Sigma(W_{ij} / \Sigma W_j) / n$$

Keterangan :

X : *eigenvector*

W_{ij} : nilai sel kolom dalam satu baris (i,j = 1....n)

ΣW_j : jumlah total kolom

n : jumlah matriks yang dibandingkan

2.3.1.5 Memeriksa Konsistensi Ratio

Setelah mendapatkan nilai eigen, selanjutnya memeriksa rasio konsistensi.

Langkah pertama mencari nilai λ_{maks} dengan cara:

$$\lambda_{maks} = (\text{nilai eigen } 1 \times \text{jumlah kolom } 1) + (\text{nilai eigen } 2 \times \text{jumlah kolom } 2) \dots n.$$

Setelah mendapatkan λ_{maks} kemudian mencari Consistency Index (CI) sebagai berikut :

$$CI = (\lambda_{maks} - n) / (n - 1) \quad (2.3)$$

Keterangan :

CI : Consistency Index

λ_{maks} : nilai eigen terbesar

n : jumlah matriks yang dibandingkan

Nilai CI tidak akan berarti apabila terdapat standar untuk menyatakan apakah CI menunjukkan matriks konsisten. Saaty memberikan patokan dengan melakukan perbandingan secara acak atas 500 buah sampel. Saaty berpendapat bahwa suatu matriks yang dihasilkan dari perbandingan yang dilakukan secara acak merupakan suatu matriks yang mutlak tidak konsisten. Dari matriks acak tersebut didapatkan juga nilai Consistency Index, yang disebut juga dengan Random Index (RI).

Dengan membandingkan CI dengan RI maka didapatkan patokan untuk menentukan tingkat konsistensi suatu matriks, yang disebut dengan Consistency Ratio (CR), dengan rumus :

$$CR = CI / RI \quad (2.4)$$

Keterangan :

CR : Consistency Ratio

CI : Consistency Index

RI : Random Index

untuk beberapa orde matriks mendapatkan nilai rata-rata RI sebagai berikut :

Tabel 2.4 Nilai *Random Index* (Saaty, 1999)

Orde Matriks	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

Saaty menerapkan bahwa suatu matriks perbandingan adalah konsisten bila nilai CR tidak lebih dari 10%. Apabila rasio konsistensi semakin mendekati ke angka nol berarti semakin baik nilainya dan menunjukkan kekonsistensian matriks perbandingan tersebut.

2.3.1.6 Membuat Supermatriks

Supermatriks merupakan matriks yang terdiri dari beberapa matriks. Supermatriks digunakan dalam ANP karena adanya hubungan keterkaitan antar elemen dalam network. Menurut Saaty, terdapat 3 jenis supermatriks dalam ANP.

1. Unweight Supermatriks

Membuat unweight supermatriks dengan cara memasukkan semua nilai eigen vector yang diperoleh dari matriks perbandingan berpasangan antar elemen. Jika diasumsikan suatu sistem memiliki N cluster dimana elemen-elemen dalam tiap I saling berinteraksi atau memiliki pengaruh terhadap beberapa atau seluruh cluster yang ada. Jika cluster dinotasikan dengan C_h , dimana $h = 1, 2, 3, \dots, N$. Dengan elemen sebanyak n_h yang dinotasikan dengan $e_{h1}, e_{h2}, \dots, e_{hn}$. Pengaruh dari satu set elemen dalam suatu cluster pada elemen yang lain dalam suatu sistem dapat direpresentasikan melalui vektor prioritas berskala rasio yang diambil dari perbandingan berpasangan. Jaringan pada metode ini memiliki kompleksitas yang tinggi dibanding dengan jenis lain, karena adanya fenomena feedback dari cluster satu ke cluster yang lain., bahkan dengan cluster-nya sendiri. Setelah model dibuat, maka dilakukan pentabelan dari hasil data pairwise comparison dengan menggunakan tabel supermatriks.

2. Weight Supermatriks

Supermatriks ini terbentuk dari tiap blok vektor prioritas dibobot berdasarkan matriks perbandingan berpasangan antar cluster.

3. Limit Supermatriks

Membuat limiting supermatriks dengan cara memangkatkan weighted supermatriks secara terus menerus hingga angka disetiap kolom dalam satu baris sama besar, yaitu dengan cara memangkatkan weighted supermatriks dengan pangkat k dimana $k= 1,2,..n$.

2.4 Software Untuk Membangun Aplikasi Web

Untuk membangun aplikasi web pada perancangan system penilaian teknis 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

Menurut Didik Setiawan (2017:16), “HTML (*Hypertext Markup Language*) adalah sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman website yang dapat diakses atau ditampilkan menggunakan *Web Browser*.

Menurut Much Aziz Muslim dan Atikah Ari Pramesti dalam *Scientifi Journal of Informatics* (2014), “HTML adalah sebuah bahasa untuk menampilkan halaman sebuah website. Html merupakan singkatan dari *Hypertext Markup Language*.

Menurut R.H. Sianipar (2015), “HTML (*Hypertext Mark up Language*) adalah sebuah mark up untuk menstrukturkan dan menampilkan isi dari halaman web.

Berdasarkan beberapa definisi maka bisa disimpulkan bahwa, HTML (*Hypertext Mark up Language*) merupakan script pemrograman standar yang kegunaannya untuk membuat sebuah website agar dapat dijelajahi dengan leluasa didunia maya.

2.4.2.2 CSS

Menurut Johni S Pasaribu dalam Jurnal Ilmiah Teknologi Informasi Terapan (2017:158), “CSS dalah singkatan dari *Cascading Style Sheet* yang merupakan kumpulan perintah yang dibentuk dari berbagai sumber yang disusun menurut urutan tertentu sehingga mampu mengatasi konflik style. CSS atau yang disebut *cascading style sheet* yaitu salah satu bahasa pemrograman web yang mengatur komponen dalam suatu web supaya lebih terstruktur dan lebih seragam.

Menurut Didik Setiawan (2017:166), “CSS adalah kependekan dari *Cascading Style Sheet*. CSS merupakan salah satu kode pemrograman yang

bertujuan untuk menghias dan mengatur gaya tampilan atau layout halaman web supaya lebih elegan dan menarik.

Berdasarkan pendapat diatas dapat disimpulkan bahwa CSS (*Cascading Style Sheet*) merupakan script. Program yang kegunaanya untuk memberikan gaya tampilan yang menarik dihalaman HTML (*Hypertext Mark up Language*).

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, 2013).

2.4.2.4 PHP

Menurut Budi Raharjo (2016:38), “PHP adalah salah satu bahasa pemrograman script yang dirancang untuk membangun aplikasi web”.

Menurut Jubilee *Enterprise* (2015:1), “PHP singkatan dari *hypertext preprocessor* adalah bahasa komputer atau bahasa pemrograman atau koding atau script yang digunakan untuk mengolah data dari server untuk ditampilkan di website”.

Menurut Priyo Sutopo dkk dalam Jurnal Infomatika Mulawarman (2016:25), “PHP adalah salah satu server side yang dirancang khusus untuk aplikasi web. PHP disisipkan diantara bahasa HTML dan karena bahasa *server side*, maka bahasa PHP akan dieksekusi di server, sehingga yang dikirimkan ke browser adalah hasil jadi dalam bentuk html, dan kode PHP tidak akan terlihat. PHP termasuk *open source product*. Jadi, dapat diubah source code dan mendistribusikannya secara bebas.

Berdasarkan beberapa definisi diatas dapat disimpulkan bahwa, PHP adalah bahasa pemrograman yang berbentuk script HTML yang prosesnya terjadi pada client-server dan bersifat open source.

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, 2014: 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

Menurut Priyo Sutopo dkk dalam Jurnal Informatika Mulawarman (2016:25), "MySQL adalah salah satu jenis *database* server yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL termasuk jenis RDBMS (*Relational Database*

Management System). Pada MySQL, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

Menurut Mukhamad Masrur (2016:124), mengatakan bahwa “MySQL adalah salah satu *relation database management system* yang bersifat *open source*”.

Menurut Budi Raharjo (2015:16), “MySQL adalah software RDBMS (server database) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (*multi-user*) dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

Berdasarkan pendapat yang dikemukakan diatas, maka dapat disimpulkan bahwa “MySQL adalah *software* yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar yang diletakan pada beberapa *table* yang terpisah”.

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.

2.5 Pemodelan Sistem dengan UML (Unified Modeling Language)

Menurut Onu, Fergus U. dan Umeakuka, Chinelo V. dalam *International Journal of Computer Applications Technology and Research* (2016:506), “A UML is a standard modeling language to model the real world in the field of software engineering. A UML diagram is a partial graphical view of model of a system under design, implementation, or already in existence.

Menurut Ary Budi Warsito dkk dalam Jurnal CCIT (2015:29), “UML (Unified Modelling Language) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut.

Berdasarkan pendapat yang telah dikemukakan oleh para ahli, penulis dapat menarik kesimpulan bahwa UML (Unified Modelling Language) adalah suatu kumpulan pemodelan yang digunakan untuk menentukan atau menggambarkan atau mendokumentasikan sebuah sistem software yang terkait dengan objek.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :



1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat 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.

2.5.1 Usecase Diagram

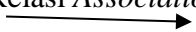

1. Tabel Komponen *Use case*

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 2.5 Komponen *Use Case Diagram* (Rosa dan Shalahuddin, 2014:156)

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 :

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

Tabel 2.6 Relasi Tabel *Use Case* (Rosa dan Shalahuddin, 2014:156)

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

Tabel 2.7 Tabel *Stereotype* yang mungkin terjadi (Rosa dan Shalahuddin, 2014:156)

Relasi/ <i>Stereotype</i>	Penjelasan
<< <i>include</i> >>	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.
<< <i>extends</i> >>	Kelakuan yang hanya berjalan dibawah kondisi tertentu seperti menggerakkan alarm.
<< <i>Communicates</i> >> >	Ditambahkan untuk asosiasi yang mungkin menunjukkan asosiasinya adalah <i>communicates association</i> .


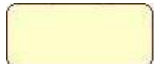






2.5.2 Activity Diagram

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 2.8 *Activity Diagram* (Rosa dan Shalahuddin, 2014:156)


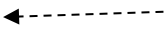

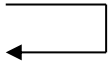
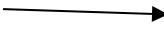
<i>Activity Diagram</i>	Penjelasan
-------------------------	------------

 <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.
 <i>End State</i>	<i>End State</i> , Sebagai tanda akhir dari <i>activity diagram</i> .

2.5.3 Squence 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 2.9 *Sequence Diagram* (Rosa dan Shalahuddin, 2014:156)

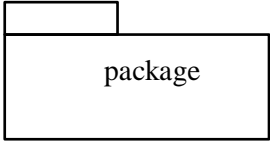
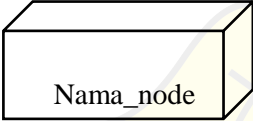
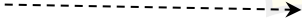
<i>Sequence Diagram</i>	Penjelasan
 <i>Actor</i>	<i>Actor</i> , Menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem.
 <i>Return Message</i>	<i>Return Message</i> , menggambarkan pesan atau hubungan antar obyek yang menunjukkan urutan kejadian yang 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 2.10 *Deployment Diagram*

<i>Deployment Diagram</i>	Penjelasan
<p data-bbox="313 520 418 548"><i>Package</i></p> 	<p data-bbox="680 520 1367 611"><i>Package</i>, merupakan sebuah bungkus dari satu atau lebih node.</p>
<p data-bbox="321 791 386 819"><i>Node</i></p> 	<p data-bbox="680 791 1367 1157">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 data-bbox="313 1194 643 1285">Kebergantungan/<i>dependenc</i> <i>y</i></p> 	<p data-bbox="680 1194 1367 1285">Kebergantungan antar node, arah panah mengarah pada node yang dipakai.</p>