

BAB II

KAJIAN PUSTAKA

II.1 Tinjauan Pustaka

Fujama Diapoldo Silalahi (2016) membuat aplikasi sistem pendukung keputusan untuk evaluasi kinerja karyawan untuk usulan gaji dan promosi jabatan berbasis web menggunakan metode AHP pada PT. HERCULON CARPET. Alasan dibangunnya sistem ini dikarenakan sistem penilaian di PT. HERCULON CARPET sebelumnya masih bersifat manual dengan pendataan penilaian karyawan menggunakan formulir penulisan tercetak dan rekap menggunakan Microsoft Excel dan data evaluasi kinerja karyawan yang tersimpan di masing-masing bagian dan belum terpusat dibagian HRD menyebabkan kurang efisiensinya laporan kinerja karyawan. Bahasa pemrograman yang digunakan PHP dan MySQL. PT. HERCULON CARPET ini mempunyai 3 cabang anak perusahaan yang berada di karawang dan semarang dengan jumlah karyawan 450 orang. Perangkat yang dibuat mampu mempermudah pihak HRD mengambil suatu sebagai dasar pertimbangan keputusan untuk menentukan karyawan yang akan memperoleh promosi jabatan.

Edianto Berutu (2015) membuat aplikasi sistem pendukung keputusan pengangkatan karyawan tetap dengan menggunakan metode AHP pada PT. PERKEBUNAN LEMBAH BHAKTI PROPINSI NAD KAB. ACEH SINGKIL. Adapun alasan yang melatarbelakangi dibuatnya sistem ini dikarenakan proses pengangkatan karyawan sebelumnya dianggap belum mencapai titik maksimal. Kenapa dibidang belum optimal karena cara pengangkatan karyawannya tidak profesional, dalam mengambil suatu keputusan layak atau tidaknya calon karyawan diangkat sebagai karyawan tetap diperusahaan tersebut. karena banyaknya calon karyawan yang harus diseleksi satu persatu berbagai kriteria kriterianya untuk diangkat menjadi karyawan tetap. Hal ini menjadi kendala PT. Perkebunan Lembah Bakti untuk mengambil suatu keputusan. Dengan cara pemilihan seperti ini yang bisa merugikan perusahaan, karena karyawan yang dipilih belum berpengalaman dan tidak bertanggung jawab dalam mengerjakan

pekerjaannya. Bahasa pemrograman yang digunakan adalah Microsoft Visual Basic 2008. Kelebihan dari sistem tersebut dapat memberikan beberapa solusi dalam menentukan karyawan yang memiliki nilai-nilai sesuai dengan yang diharapkan. Implementasi metode *Analytic Hierarchy process* (AHP) dalam Pengangkatan karyawan tetap ini memiliki kelemahan dalam pengangkatan karyawan tetap pada PT. Perkebunan Lembah Bakti tidak bisa digunakan untuk melakukan penilaian jika yang dinilai hanya satu calon Karyawan.

Berdasarkan uraian penelitian di atas, yang lebih menekankan pada analisis pemodelan *Analytic Hierarchy process* (AHP) untuk mendukung keputusan, tanpa mengembangkan solusi sistem pemodelan *Analytic Hierarchy process* (AHP) untuk mendukung sistem pengambilan keputusan. Penelitian yang akan dikembangkan adalah mengenai Sistem Pendukung Keputusan Pemilihan *Leader* Menggunakan Metode AHP (*Analytic Hierarchy Process*) Studi Kasus PT. Agung Cipta Jaya.

II.2 Landasan Teori

II.2.1 Pengertian Sistem

Menurut Jogiyanto, sistem adalah sekelompok elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. (Jogiyanto H.M, 2009) Pendekatan sistem yang menekankan pada komponen akan lebih mudah didalam mempelajari suatu sistem untuk tujuan analisis dan perancangan suatu sistem. Untuk menganalisis dan merencanakan suatu sistem, analisa dan perancangan sistem dibutuhkan pengertian dan pemahaman mengenai elemenelemen atau subsistem-subsistem dari sistem tersebut. Suatu sistem mempunyai maksud tertentu, yaitu untuk mencapai suatu tujuan (*goal*) dan untuk mencapai suatu sasaran (*objectives*). (Jogiyanto H.M, 2009).

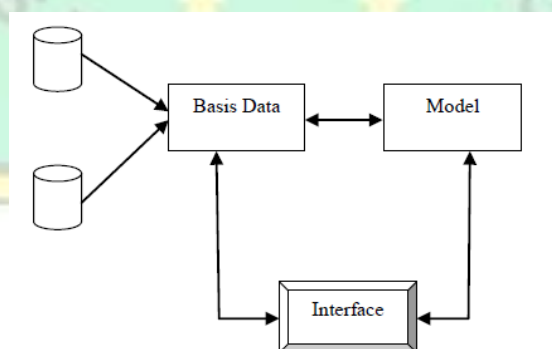
II.2.2 Sistem Pendukung Keputusan

Sistem pendukung keputusan (*decision support systems* disingkat DSS) adalah bagian dari sistem informasi berbasis komputer termasuk sistem berbasis pengetahuan (manajemen pengetahuan) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi perusahaan, atau lembaga

pendidikan. Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah yang spesifik. (Turban, 2005:1)

Kegiatan merancang sistem pendukung keputusan merupakan sebuah kegiatan untuk menemukan, mengembangkan dan menganalisis berbagai alternatif tindakan yang mungkin untuk dilakukan. Tahap perancangan ini meliputi pengembangan dan mengevaluasi serangkaian kegiatan alternatif. Sedangkan kegiatan memilih dan menelaah ini digunakan untuk memilih satu rangkaian tindakan tertentu dari beberapa yang tersedia dan melakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor-faktor yang perlu dipertimbangkan dalam pengambilan keputusan. (Turban, 2005:3)

Konsep Sistem Pendukung Keputusan pertama kali diperkenalkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System*. Konsep pendukung keputusan ditandai dengan sistem interaktif berbasis komputer yang membantu pengambil keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tidak terstruktur. Pada dasarnya SPK dirancang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam prosen pengambilan keputusan, sampai mengevaluasi pemilihan alternative. (Hilyah Magdalena : 2012)



Gambar II.1 Komponen Sistem Pendukung Keputusan

a. Tahap-Tahap Pembuatan Keputusan

Dalam mengambil keputusan dilakukan langkah-langkah sebagai berikut :

1. Identifikasi masalah
2. Pemilihan metode pemecahan masalah
3. Pengumpulan data yang dibutuhkan untuk melaksanakan model keputusan tersebut
4. Mengimplementasikan model tersebut
5. Mengevaluasi sisi positif dan setiap alternatif yang ada
6. Melaksanakan solusi terpilih.(Kusrini, 2007 hal 09)

b. Langkah-Langkah Pemodelan dalam DSS (*Decision Support Systems*)

Saat melakukan pemodelan dalam pembangunan DSS dilakukan langkah-langkah sebagai berikut:

1. Studi kelayakan (*Intelligence*)

Pada langkah ini, sasaran ditentukan dan dilakukan pencarian prosedur, pengumpulan data, identifikasi masalah, identifikasi kepemilikan masalah, klasifikasi masalah, hingga akhirnya terbentuk sebuah pernyataan masalah. Kepemilikan masalah berkaitan dengan bagian apa yang akan dibangun oleh DSS dan apa tugas dari bagian tersebut sehingga model tersebut bisa relevan dengan kebutuhan si pemilik masalah.

2. Perancangan (*Design*)

Pada tahapan ini akan di formulasikan model yang akan digunakan dan kriteria-kriteria yang ditentukan. Setelah itu, dicari alternative model yang bisa menyelesaikan permasalahan tersebut. Langkah selanjutnya adalah memprediksi keluaran yang mungkin. Kemudian, ditentukan variabel-variabel model.

3. Pemilihan (*Choice*)

Setelah pada tahap *design* ditentukan berbagai alternatif model beserta variabel-variabelnya, pada tahapan ini akan dilakukan pemilihan modelnya, termasuk solusi dari model tersebut. Selanjutnya, dilakukan analisis sensitivitas, yakni dengan mengganti beberapa variabel.

4. Membuat DSS

Setelah menentukan modelnya, berikutnya adalah mengimplementasikannya dalam aplikasi DSS. (Kusrini,2007 hal 30).

c. Tujuan Sistem Pendukung Keputusan

Banyaknya defenisi yang dikemukakan mengenai pengertian dan penerapan dari sebuah sistem pendukung keputusan, maka terdapat tujuan sistem pendukung keputusan yaitu sebagai berikut :

1. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektivitas keputusan yang diambil manajer lebih dari perbaikan efisiensinya.
4. Kecepatan komputasi, komputer memungkinkan para pengambil keputusan untuk melakukan banyak komptasi secara cepat dengan biaya rendah.
5. Peningkatan produktifitas, membangun satu kelpok pengambil keputusan, terutama para pakar bias sangat mahal. Pendukung terkomputerisasi bias megurangi ukurn kelompok dan memungkinkan para anggotanya untuk berada diberbagai lokasi yag berbedabeda (menghemat biaya perjalanan). Selain itu, produktivitas staf pendukung (misalnya analisis keuangan dan hukum) bisa ditingkatkan produktivitas juga bisa ditingkatkan menggunakan peralatan optimalisasi yang menentukan cara terbaik untuk menjalanka sebuah bisnis.
6. Dukungan kualitas, komputer bisa meningkatkan kualitas keputusan yang dibuat. Sebagai contoh, semakin banyak datayang diakses, maka banyak juga alternate yang bisa dievaluasi. Analisis risiko bisa dilakukan dengan cepat dan pandangan dari para pakar (beberapa dari mereka berada dilokasi yang jauh) bisa dikumpulkan dengan cepat dan dengan biaya yang lebih rendah. Keahlian bahkan bisa diambil langsung dari sebuah sistem komputer, para pengambil keputusan bisa melakukan simulasi yang

kompleks, memeriksa banyak scenario yang memungkinkan, dan menilai berbagai pengaruh secara cepat dan ekonomis, semua kapabilitas tersebut mengarah kepada keputusan yang lebih baik.

7. Berdaya saing. Manajemen dan pemberdayaan sumber daya perusahaan. Tekanan persaingan menyebabkan tugas pengambilan keputusan menjadi sulit. Persaingan didasarkan tidak hanya pada harga, tetapi juga pada kualitas, kecepatan, kustomasi produk, dan dukungan pelanggan. Organisasi harus mampu secara sering dan cepat mengubah mode operasi, merencanakan ulang proses dan struktur, memberdayakan karyawan, serta berinovasi. Teknologi pengambilan keputusan bisa menciptakan pemberdayaan yang signifikan dengan cara memperbolehkan seseorang untuk membuat keputusan yang baik secara cepat, bahkan jika mereka memiliki pengetahuan yang kurang.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan. Menurut Simon (1977), otak manusia memiliki kemampuan yang terbatas untuk memproses dan menyimpan informasi. Orang-orang kadang sulit mengingat dan menggunakan sebuah informasi dari cara yang bebas dengan kesalahan. (Kusrini, 2007 hal 16)

d. Pembagian Sistem Pendukung Keputusan

Ditinjau dari tingkat teknologinya, sistem pendukung keputusan dibagi menjadi 3, yaitu:

1. SPK spesifik

SPK spesifik bertujuan membantu memecahkan suatu masalah dengan karakteristik tertentu. Misalnya, SPK penentuan harga satuan barang.

2. Pembangkit SPK

Suatu software yang khusus digunakan untuk membangun dan mengembangkan SPK. Pembangkit SPK akan memudahkan perancang dalam membangun SPK spesifik.

3. Perlengkapan SPK

Berupa software dan hardware yang digunakan atau mendukung pembangunan SPK spesifik maupun pembangkit SPK. (Kusrini, 2007 hal 18).

II.2.3 Promosi Jabatan

Promosi Jabatan menurut Manullang (2001:153) "Promosi berarti kenaikan jabatan, yakni menerima kekuasaan dan tanggung jawab lebih besar dari kekuasaan dan tanggung jawab sebelumnya". Sedangkan menurut Hasibuan (2002:108) "Promosi adalah perpindahan yang memperbesar authority dan responsibility karyawan ke jabatan yang lebih tinggi didalam suatu organisasi sehingga kewajiban, hak, status, dan penghasilan semakin besar". Menurut Hasibuan (2002:113)

Ada beberapa tujuan dilaksanakannya promosi jabatan yaitu:

1. Untuk memberikan pengakuan, jabatan, dan imbalan jasa yang semakin besar kepada karyawan yang berprestasi kerja tinggi.
2. Dapat menimbulkan kepuasan dan kebanggaan pribadi, status sosial yang semakin tinggi, dan penghasilan yang semakin besar.
3. Untuk merangsang agar karyawan lebih bergairah bekerja, berdisiplin tinggi, dan memperbesar kinerjanya.
4. Untuk menjamin stabilitas kepegawaian.
5. Memperbaiki status karyawan dari karyawan sementara menjadi karyawan tetap setelah lulus dalam masa percobaan.

II.2.4 Analytical Hierarchy Process (AHP)

Analytical Hierarchy Process (AHP) dikembangkan oleh Thomas L. Saaty pada tahun 1970-an. Metode ini merupakan salah satu model pengambilan keputusan multikriteria yang dapat membantu kerangka berpikir manusia dimana faktor logika, pengalaman pengetahuan, emosi dan rasa dioptimalkan ke dalam suatu proses sistematis. Pada dasarnya, AHP merupakan metode yang digunakan untuk memecahkan masalah yang kompleks dan tidak terstruktur ke dalam

kelompok–kelompoknya, dengan mengatur kelompok tersebut ke dalam suatu hierarki, kemudian memasukkan nilai numerik sebagai pengganti persepsi manusia dalam melakukan perbandingan relatif. Dengan suatu hipotesa maka akan dapat ditentukan elemen mana yang mempunyai prioritas tertinggi. (Hilyah Magdalena : 2012)

a. Prinsip Dasar *Analytical Hierarchy Process* (AHP)

Dalam menyelesaikan persoalan dengan metode *Analytic Hierarchy Process* (AHP) ada beberapa prinsip dasar yang harus dipahami (Kusrini,2007:133) antara lain:

1. Membuat hierarki

Sistem yang kompleks bisa dipahami dengan memecahnya menjadi elemen-elemen pendukung, menyusun elemen secara hierarki, dan menggabungkannya atau mensistesisnya.

2. Penilaian kriteria dan alternatif

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty (1998), untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan saaty bisa diukur menggunakan tabel analisis seperti ditunjukkan pada tabel berikut.

Tabel II.1 Skala Penilaian Perbandingan Pasangan

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
5	Elemen yang satu lebih penting daripada elemen lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen jelas mutlak penting daripada elemen lainnya

2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
Kebalikannya	Jika aktivitas I mendapat satu angka dibandingkan dengan aktivitas j, maka I memiliki nilai kebalikannya dibandingkan dengan i

(Sumber : Kusrini, Konsep dan Aplikasi Sistem Pendukung Keputusan 2007 hal 134)

3. *Synthesis of Priority* (menentukan prioritas)

Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan, berpasangan (*pairwise Comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan judgement yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi matriks atau melalui penyelesaian persamaan matematika.

4. *Logical Consistency* (Knsistensi Logis)

Konsentensi memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antar objek yang didasarkan pada kriteria tertentu.

b. Prosedur *Analytical Hierrchy Process* (AHP)

Pada dasarnya, prosedur atau langkah-langkah dala metode *Analytical Hierrchy Process* (AHP) (Kusrini,2007:135) meliputi :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas.
2. Menentukan prioritas elemen
 - a. langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan pasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan.

- b. matriks perbandingan berpasangan diisi dengan menggunakan bilangan untuk mempresentasikan kepentingan relatif dari suatu elemen terhadap elemen yang lainnya.

3. Sintesis

Pertimbangan-pertimbangan terhadap perbandingan berpasangan di sintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini adalah :

- a. menjumlahkan nilai-nilai dari setiap kolom pada matriks
- b. membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks
- c. menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata

4. Mengukur Konsistensi

Dalam pembuatan keputusan , penting untuk mengetahui seberapa baik konsistensi yang ada karena kita tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah :

- b. kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya
- c. jumlahkan setiap baris dibagi dengan elemen prioritas relatif yang bersangkutan
- d. jumlahkan hasil bagi diatas dengan banyaknya elemen yang ada, hasilnya disebut λ maks.

5. Hitung Consistency Index CI dengan rumus :

$$CI = (\lambda \text{ maks} - n)/n$$

dimana n = banyaknya elemen

6. Hitung Rasio Konsistensi/Consistency Rasio (CR) dengan rumus :

$$CR = CI/RC$$

dimana CR = Consistency Ratio

CI = Consistency Index

IR = Indeks Random Consistency

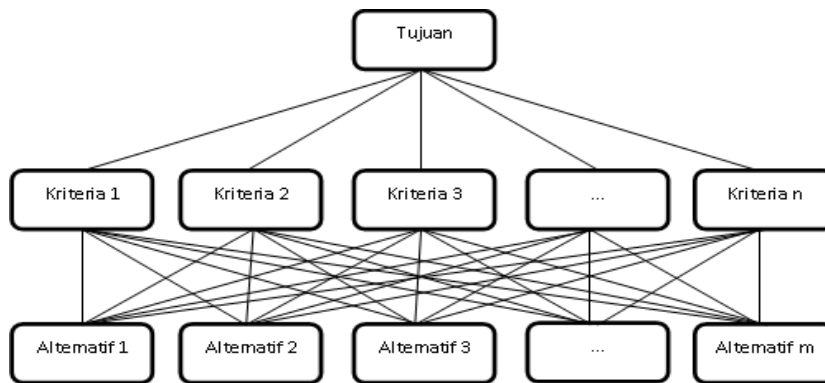
7. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10%, maka penilaian data judgement harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar.

Daftar Indeks Random Konsistensi (IR) bisa dilihat dalam tabel berikut

Tabel II.2 Daftar Indeks Random Konsistensi

Ukuran Matriks	Nilai IR
1,2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49
11	1,51
12	1,48
13	1,56
14	1,57
15	1,59

- c. Tahapan - tahapan dalam AHP (Analytic Hierarchy Process)
 1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
 2. Membuat struktur hierarki yang diawali dengan tujuan umum, dilanjutkan dengan kriteria-kriteria dan alternatif- alternatif pilihan.



Gambar II.2 Struktur hierarki AHP

3. Membuat matrik perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap tujuan atau kriteria yang setingkat di atasnya. Perbandingan dilakukan berdasarkan pilihan atau judgement dari pembuat keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya.
4. Menormalkan data yaitu dengan membagi nilai dari setiap elemen di dalam matrik yang berpasangan dengan nilai total dari setiap kolom
5. Menghitung nilai eigen vector dan menguji konsistensinya, jika tidak konsisten maka pengambilan data (preferensi) perlu diulangi. Nilai eigen vector yang dimaksud adalah nilai eigen vector maksimum yang diperoleh.
6. Mengulangi langkah 3, 4 dan 5 untuk seluruh tingkat hirarki.
7. Menghitung eigen vector dari setiap matriks perbandingan berpasangan. Nilai eigen vector merupakan bobot setiap elemen.
8. Menguji konsistensi hirarki. Jika tidak memenuhi dengan $CR < 0,100$ maka penilaian harus diulangi kembali.

II.2.5 Perangkat Lunak Pendukung

1. PHP

Php adalah bahasa *scripting* yang ditempelkan (*embedded*) ketika digunakan dalam halaman *web* (Janner, 2006:32).

2. MySQL

Menurut Bunafit (2004:1) MySQL merupakan sebuah program *database* server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*) dengan pengembang *web* (*Web Developers*).

3. Adobe Dreamweaver CS3

Menurut devisi penelitian MADCOMS (2007:3) dreamweaver merupakan software utama yang digunakan oleh *web designer* maupun *web programmer* dalam mengembangkan suatu situs web.

II.2.6 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

UML dapat digunakan untuk membuat model segala jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Karena konsep dasar UML menggunakan *class* dan *operation*, maka lebih cocok untuk penulisan perangkat lunak dalam bahasa berorientasi objek.

Notasi UML adalah sekumpulan bentuk khusus yang menggambarkan berbagai diagram piranti lunak, dan tiap-tiap bentuk memiliki makna tertentu. Notasi UML merupakan gabungan dari 3 notasi yang telah ada sebelumnya yaitu *Grady Booch OOD*, *Jim Rumbaugh OMT*, dan *Ivar Jacobson OOSE*.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan merupakan notasi dari metode yang digunakan untuk mendesain

secara cepat. Bagian utama dari *UML* adalah *view*, diagram, model elemen dan *general mechanism*.

1. *View*

Berfungsi untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan digunakan untuk melihat grafik, tetapi *view* merupakan suatu abstraksi yang berisi sejumlah diagram. Jenis-jenis *view* antara lain *use case view*, *logical view*, komponen *view*, *concurrent view* dan *deployment view*.

a. *Use case view*

Use case diagram berfungsi untuk memahami dan menggunakan sistem yang dimodelkan. *View* ini melihat pada bagaimana *actor* dan *use case* berinteraksi.

b. *Logical view*

Logical view merupakan persyaratan (*requirements*) fungsional sistem. *View* ini melihat pada kelas-kelas dan hubungan antar kelas-kelas tersebut.

c. *Komponen view*

Component view bertujuan untuk pengaturan software. *View* ini mengandung informasi mengenai komponen-komponen *software*, komponen tereksekusi (*executable*) dan *library* dan sistem yang dimodelkan.

e. *Deployment view*

Deployment view memperlihatkan pemetaan setiap proses ke dalam *hardware*. *View* ini sangat bermanfaat ketika membuat model suatu sistem yang diterapkan dalam lingkungan arsitektur yang terdistribusi yang diterapkan pada aplikasi dan server pada lokasi yang berbeda.

2. *Diagram*

Diagram berbentuk grafik menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan jika digambar biasanya dialokasikan untuk *view* tertentu.

Jenis-jenis diagram antara lain *use case diagram*, *class diagram*, *activity diagram*, *sequence diagram*, *package diagram* dan *deployment diagram*.

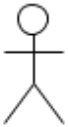
a. *Use case diagram*



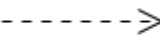



Usecase diagram mendiskripsikan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem, dan bukan bagaimana. Sebuah *usecase* mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem. Seorang/ sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.



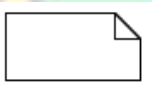
Use case diagram sangat membantu pada saat penyusunan *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat menginclude fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *usecase* yang di *include* akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal.

Sebuah *use case* dapat di *include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviournya* sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari *use case* yang lain.

Tabel II.3 Daftar simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		Actor	Actor adalah pengguna system. Actor tidak terbatas pada manusia saja, jika sebuah system berkomunikasi dengan

			aplikasi lain dan membutuhkan input atau meberikan output, maka aplikasi tersebut bisa juga dianggap sebagai actor.
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		Generalization	Hubungan dimana objek anak (desendent) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk.
4		Include	Menspesifikasikan bahwa usecase sumber secara eksplisit.
5		Extend	Menspesifikasikan bahwa usecase target memperkuat perilaku dari usecase sumber pada suatu titik diberikan.
6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		System	Memspesifikasikan paket yang menampilkan system secara terbatas.


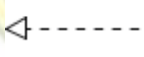
8		Use case	Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (bersinergi).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.


b. *Class diagram*

Class adalah sebuah spesifikasi yang apabila di instansiasikan akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan atribut atau properti suatu sistem dan memberikan layanan untuk memanipulasi keadaan tersebut (fungsi/metoda). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi dan lain-lain.

Class dapat berupa implementasi sebuah *interface*, yaitu kelas abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplemantasikan dahulu menjadi sebuah *class*. Dengan demikian interface mendukung resolusi metoda pada saat *run-time*.

Table II.4 Daftar simbol *class diagram*

No	Gambar	Nama	Keterangan
1		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang berada di atasnya objek induk (ancestor).
2		Nary association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor.
5		Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
6		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.


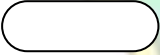
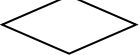
7		Association	Apa yang menghubungkan antara objek satu dengan objek.
---	---	-------------	--

c. *Activity diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di *trigger* oleh selesainya *state* sebelumnya (*internal processing*).

Table II.5 Daftar simbol *Activity Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Start state</i>	Titik awal atau permulaan
	<i>End state</i>	Titik akhir atau akhir dari aktivitas
	<i>Activity</i>	<i>Activity</i> atau aktivitas yang dilakukan oleh actor
	<i>Decision</i>	Pilihan untuk mengambil keputusan
	<i>Interaction</i>	Alur

II.2.7 Tinjauan Umum PT. AGUNG CIPTA JAYA

1. Sejarah Singkat PT. AGUNG CIPTA JAYA

PT. Agung Cipta Jaya adalah perusahaan Nasional yang mengelola program komunitas. Sebagai program anak negeri yang mengusung visi nasionalisme dan pemberdayaan ekonomi masyarakat telah bekerjasama dengan komunitas UKM di seluruh Indonesia untuk bisa saling bersinergi dan terhubung dengan layanan program *free call* dari INDOSAT sehingga setiap anggotanya bisa berbagi informasi dengan cepat bahkan gratis

PT. Agung Cipta Jaya berdiri di SEMARANG pada tanggal 06 Desember 2015 dengan sebuah kesungguhan dan komitmen sepenuh hati untuk dapat menghadirkan satu konsep komunitas yang akan menjadi Rumah Besar kita semua. lintas bisnis yang fair dan transparan, dimana secara sistem Anda akan ditempatkan sebagai subjek yang akan menikmati potensi dari kerjasama antar sub sub komunitas UKM , dikelola secara profesional oleh orang-orang yang kompeten dibidangnya, serta dukungan perusahaan IT Partner yang terpercaya untuk menjamin kenyamanan dan kelancaran pelayanan.

2. Visi dan Misi Perusahaan

Visi

- Menjadi perusahaan yang terpercaya dan dipilih untuk menjadi wadah bersama untuk saling terhubung dan mengakselerasi terbentuknya potensi bisnis baru antar sub komunitas
- PT. Agung Cipta Jaya menyediakan layanan pembayaran guna memudahkan masyarakat untuk melakukan one-stop payment untuk segala kebutuhan pembayaran/transaksi keuangan/bisnis sehari-hari. Dizaman yang semakin modern, kesibukan dan aktivitas bisnis yang semakin tinggi, diperlukan sistem pembayaran yang lebih cepat, mudah, aman, flexibel dan menguntungkan.

Misi

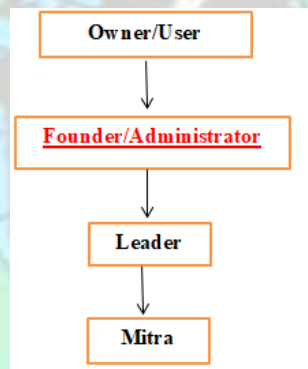
- Memudahkan anggota untuk saling berkomunikasi dengan cara yang murah, mudah dan menguntungkan

- Mendorong kemudahan berkomunikasi yang akan mengakselerasi potensi bisnis antar sub komunitas
- Menjadi fasilitator pengusaha-pengusaha baru dan mengurangi pengangguran
- Mewujudkan masyarakat yang mandiri dan memiliki penghasilan sendiri
- Memberikan kesejahteraan bagi para mitra bisnis UKM sebagai mitra strategis pengembangan bisnis.

3. Pendiri Perusahaan

Nama : R.M AGUNG PRIHARTONO
 TTL : Semarang, 27-11-1973
 Alamat : Merayu Utara Rt 007 / 002 Kembangan
 Jakarta

4. Struktur Organisasi



Gambar II.3 Struktur Organisasi

Pada gambar struktur organisasi diatas menunjukkan *Owner* adalah sebagai pimpinan utama/pemilik perusahaan dan sebagai salah satu user pada program, *Founder/Administrator* bertugas untuk melakukan proses pemilihan *leader* atau menjadi administrator pada program ini