

PERANCANGAN SISTEM INFORMASI BERORIENTASI
OBJEK DENGAN METODE UNIFIED MODELLING
LANGUAGE PADA SISTEM PENGGAJIAN
DI BPR SHINTA DAYA YOGYAKARTA



SKRIPSI

Oleh :

Nama : Ratna Furi Maulina Karebet

No. Mahasiswa : 99312227

FAKULTAS EKONOMI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA

2005

**PERANCANGAN SISTEM INFORMASI BERORIENTASI
OBJEK DENGAN METODE UNIFIED MODELLING
LANGUAGE PADA SISTEM PENGGAJIAN
DI BPR SHINTA DAYA YOGYAKARTA**

SKRIPSI

Disusun dan diajukan untuk memenuhi sebagai salah satu syarat untuk
Mencapai derajat Sarjana Strata -I Jurusan Akuntansi
Pada Fakultas Ekonomi UII

Oleh :

Nama : Ratna Furi Maulina Karebet

No. Mahasiswa : 99 312 227

**FAKULTAS EKONOMI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2005

PERNYATAAN BEBAS PLAGIARISME

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam referensi. Dan apabila dikemudian hari terbukti bahwa pernyataan ini tidak benar maka saya sanggup menerima hukuman/sangsi apapun sesuai peraturan yang berlaku.

Yogyakarta,2005

Penyusun,

(Ratna Furi Maulina Karebet)

**PERANCANGAN SISTEM INFORMASI BERORIENTASI
OBJEK DENGAN METODE UNIFIED MODELLING
LANGUAGE PADA SISTEM PENGGAJIAN
DI BPR SHINTA DAYA YOGYAKARTA**

Diajukan oleh

Nama : Ratna Furi Maulina Karebet

Nomor Mahasiswa : 99312227

Jurusan : Akuntansi

Telah disetujui oleh Dosen Pembimbing

Pada tanggal.....10/6/2005.....

Dosen Pembimbing,



(Drs, Dekar Urumsah, S, Si, M. Com)

BERITA ACARA UJIAN SKRIPSI

SKRIPSI BERJUDUL

PERANCANGAN SISTEM INFORMASI BERORIENTASI OBYEK DENGAN
METODE UNIFIED MODELLING LANGUAGE PADA SISTEM PENGGAJIAN DI
BPR SHINTA DAYA YOGYAKARTA

Disusun Oleh: RATNA FURI MAULINA KAREBET
Nomor mahasiswa: 99312227


Telah dipertahankan di depan Tim Penguji dan dinyatakan LULUS
Pada tanggal : 15 Juli 2005

Pembimbing Skripsi/Penguji : Drs. Dekar Urumsah, S.Si, M.Com

Penguji : Dra. Isti Rahayu, M.Si, Ak



Mengetahui
Dekan Fakultas Ekonomi
Universitas Islam Indonesia
Drs. Suwarsono, MA



PERSEMBAHAN

Dengan segenap cinta dan sayang

Skripsi ini kupersembahkan kepada:

Φ Papa dan mama yang telah membesarkanku dan memberi sedikit bekal untuk masa depanku dengan tulus

Φ Kakak dan saudara-saudaraku yang senantiasa membantu memberikan ide-idenya dan yang selalu sayang padaku

Φ Suami dan anakku yang telah memberikan bantuannya pada skripsiku dengan segala doa dan kesabarannya.

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat-Nya dan hidayah-Nya, sehingga Skripsi ini dapat terselesaikan dengan baik, guna memenuhi salah satu persyaratan untuk memperoleh gelar Sarjana Jurusan Akuntansi Fakultas Ekonomi UNIVERSITAS ISLAM INDONESIA.

Besar harapan penulis, dengan tersusunnya skripsi ini semoga bermanfaat bagi pihak PT. BPR Shinta Daya dan terlebih-lebih bagi penulis sendiri untuk dapat dijadikan sebagai tambahan ilmu tentang keadaan sebenarnya di lapangan. Adapun judul skripsi ini adalah “ Perancangan Berorientasi Obyek Dengan Metode Unified Modelling Language Pada Sistem Penggajian Di BPR Shinta Daya Yogyakarta. “

Semenjak persiapan hingga terselesainya skripsi ini, penulis mendapatkan pengawasan dan bimbingan serta bantuan orang lain, maka pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada :

1. Bapak Drs. H. Suwarsono, MA sebagai Dekan Fakultas Ekonomi UII yang Telah memberikan perlindungan dan fasilitas yang kondusif bagi mahasiswanya untuk menuntut ilmu.
2. Bapak Drs. Dekar Urumsah, S. Si, M. Com sebagai Dosen pembimbing yang telah memberikan arahan dan bimbingan dalam penyusunan skripsi ini.

3. Bapak Y. Soemantoro, BSc. Sebagai Komisaris PT. BPR Shinta Daya Yogyakarta yang telah memberikan ijin untuk melakukan penelitian di PT. BPR. Shinta Daya Yogyakarta.
4. Bapak Alex Sukono sebagai inspirator yang telah banyak meluangkan waktunya memberikan pengarahan selama penyusunan skripsi ini.
5. Papa dan Mama tercinta yang telah banyak memberikan dorongan serta doa kepada penulis sehingga penulis berhasil dalam penyusunan skripsi ini.
6. Guruku dari SD sampai SMA dan Dosenku UII yang telah membagi ilmunya dengan saya hingga jadi Sarjana.
7. Keluargaku yang telah banyak memberikan bantuan moril dan spiritual yang sangat berguna atas studiku ini.
8. Suamiku dan anakku Intan tercinta yang atas kesabarannya, untuk selalu menemani saya dalam pembuatan Skripsi ini.
9. Semua pihak yang terkait yang tak dapat saya sebutkan satu persatu yang telah ikut membantu terselesaikannya Skripsi ini.

Akhirnya penyusun menyadari bahwa skripsi ini masih jauh dari kesempurnaan, (untuk segala saran dan kritik dari para pembaca sangat diharapkan) dan semoga karya ini bermanfaat bagi para pembaca semua.

DAFTAR ISI

Hal	
HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN BEBAS PLAGIARISME	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah	5
1.4. Tujuan dan Manfaat Penelitian.....	5
1.5. Metode Pengumpulan Data.....	6
1.6. Sistematika Pembahasan.....	6
BAB II LANDASAN TEORI.....	8
2.1 Konsep Dasar Sistem Informasi	8
2.2 Definisi Sistem Informasi	8
2.3 Definisi Sistem dan Prosedur.....	9
2.4 Sistem dan Prosedur Penggajian.....	10
2.4.1 Fungsi Bagian Gaji.....	11
2.4.2 Sistem dan Prosedur Akuntansi Gaji.....	12
2.5 Pendekatan Berorientasi Objek.....	13
2.6 Object Oriented Analysis and Design (OOAD)	14
2.6.1 Konsep Dasar dalam OOAD	14
2.6.1.1 Objek	15
2.6.1.2 Kelas.....	15

2.6.1.3 Association dan Aggregation	16
2.7 Metodologi Berorientasi Objek	16
2.8 Pemilihan Metodologi.....	17
2.9 UML.....	18
2.9.1 Use Case Diagram	19
2.9.2 Class Diagram	23
2.9.3 Behaviour Diagram	29
2.9.3.1 Statechart Diagram.....	29
2.9.3.2 Activity Diagram.....	32
2.9.3.3 Interaction Diagram	33
2.9.3.3.1 Sequence Diagram	33
2.9.3.3.2 Collaboration Diagram.....	36
BAB III GAMBARAN UMUM PERUSAHAAN	38
3.1 Sejarah Berdirinya Perusahaan	38
3.2 Visi dan Misi Perusahaan	40
3.3 Struktur Organisasi	42
3.3.1 Job Description.....	44
3.4 Personalia.....	48
3.5 Bidang Usaha.....	50
3.6 Sistem Penggajian dan Pengupahan	51
BAB IV ANALISA DAN PEMBAHASAN	54
4.1 Diagram Class.....	54
4.2 Use Case Diagram (UCD)	56
4.3 Diagram Statechart	63
4.3.1 Diagram <i>Statechart</i> Kelas Departemen.....	63
4.3.2 Diagram <i>Statechart</i> Objek Kelas Absensi.....	65
4.4 Diagram Activity	67
4.4.1 Diagram <i>Activity</i> Use Case Melakukan Absensi.....	67
4.4.2 Diagram <i>Activity</i> Use Case Menghitung Gaji	68
4.5 Diagram Sequence	69
4.6 Diagram Collaborasi.....	71

BAB V KESIMPULAN DAN SARAN.....	72
5.1 Kesimpulan.....	72
5.2 Saran	74
DAFTAR PUSTAKA	75

DAFTAR TABEL

Tabel	Hal
2.1 Indikator Multiplicity	27
4.1 Atribut Class Diagram Sistem Penggajian Karyawan	55
4.2 Skenario Use Case.....	59

DAFTAR GAMBAR

Gambar	Hal
2.1 Diagram <i>Use Case</i> Sistem Pendaftaran Mata Kuliah	20
2.2 Diagram Use Case dengan Stereotype	22
2.3 Relasi Asosiasi dalam Diagram <i>Class</i>	25
2.4 Relasi Agregasi dalam Diagram <i>Class</i>	25
2.5 Relasi Dependency dalam Diagram <i>Class</i>	26
2.6 Relasi Generalisasi dalam Diagram <i>Class</i>	27
2.7 Diagram <i>Class</i> Sistem Pendaftaran Mata Kuliah.....	28
2.8 Diagram Statechart Sistem Pendaftaran Mata Kuliah.....	30
2.9 Diagram Activity.....	32
2.10 Diagram Sequence	34
2.11 Diagram <i>Collaboration</i>	36
3.1 Struktur Organisasi	43
4.1 Diagram <i>Class</i> untuk Sistem Penggajian Karyawan.....	55
4.2 Gambar Aktor dalam Sistem.....	56
4.3 Paket Diagram Use Case.....	57
4.4 Diagram Use Case untuk Sistem Absensi dan Perhitungan Gaji.....	58
4.5 Diagram Use Case Sistem Untuk Direktur	59
4.6 Diagram Statechart Objek Keuangan Untuk Update Gaji	64
4.7 Diagram Statechart Objek Keuangan Untuk Menghitung Gaji	65
4.8 Diagram Statechart Objek Kelas Absensi	66
4.9 Diagram Activity Use Case Melakukan Absensi.....	68
4.10 Diagram Activity Use Case Menghitung Gaji	69
4.11 Diagram Sequence Sistem Penggajian Karyawan	70
4.12 Diagram <i>Collaboration</i> Sistem Penggajian Karyawan.....	71



PT. Bank Perkreditan Rakyat

SHINTA DAYA

SURAT KETERANGAN

NO. 262/Dir.BSD/06/2005

Yang bertandatangan di bawah ini Direksi PT BPR Shinta Daya:

Nama : Y. Sumartono, BSc
Jabatan : Direktur
Alamat : Bogem, Kalasan, Sleman, Yogyakarta 55571

Dengan ini menerangkan bahwa

Nama : Ratna Furi M. K
No. Mahasiswa : 99.312.227
Fakultas : Ekonomi
Program Studi : Akuntansi
Universitas Islam Indonesia Yogyakarta

telah melakukan penelitian di PT BPR Shinta Daya sejak tanggal 1 April s/d 29 April 2005, untuk menyusun skripsi guna melengkapi persyaratan kelulusan sarjana, dengan judul "PERANCANGAN SISTEM INFORMASI BERORIENTASI OBJEK DENGAN METODE UNIFIED MODELLING LANGUAGE YANG DITERAPKAN PADA SISTEM PENGGAJIAN DI BPR SHINTA DAYA YOGYAKARTA".

Demikian surat keterangan ini dibuat, agar dapat dipergunakan sebagaimana mestinya.

Kalasan, 16 Juni 2005

PT BPR Shinta Daya

Y. Sumartono, BSc

Direktur

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Sistem informasi mempunyai peran penting dalam kehidupan manusia, ketergantungan manusia akan sistem informasi juga mempengaruhi dunia bisnis. Dengan semakin berkembangnya teknologi dan ilmu pengetahuan banyak inovasi dan teknik baru untuk mempermudah manusia dalam pekerjaannya termasuk dalam teknik rekayasa perangkat lunak diantaranya *visual modeling*, *visual programming*, serta pembuatan *framework* dan pola.

Meningkatnya kompleksitas dunia bisnis yang disebabkan bertambah luasnya tahapan proses merupakan suatu motivasi kunci untuk menerapkan metode perancangan perangkat lunak berorientasi objek menggunakan semantic dan notasi yang mampu menangani kerumitan arsitektural.

Selama ini metode struktur pendekatan *top-down* sangat populer dalam rekayasa desain perangkat lunak sehingga memacu peningkatan kualitas perangkat lunak, tapi dalam pembangunan sistem berskala besar pendekatan ini sering menemui banyak hambatan.

Metode terstruktur sering kali tidak dapat mendesain apa yang akan terjadi dalam sistem yang telah selesai tanpa mengimplementasikan sistem terlebih dahulu. Data dan fungsi disusun secara terpisah sering kali tidak dipikirkan sebelumnya untuk

mengetahui data mana yang digunakan dalam fungsi tertentu, jika ditemukan kesalahan dalam desain, sehingga harus disusun ulang dari awal sampai akhir, hal ini akan menjadi pemborosan biaya dan waktu. Para ahli mengakui bahwa pengembangan perangkat lunak metode pendekatan berorientasi objek terbaik yang ada saat ini, efektif dan efisien dalam membangun sistem perangkat lunak yang berskala besar serta kompleks.

Metode ini memecahkan suatu masalah dengan membuat model yang dibuat menurut konsep sekitar dunia nyata, dasar pembuatannya adalah obyek yang merupakan kombinasi antara struktur data dan perilaku dalam entitas. Implementasi membuat model ini menggunakan teknik *object oriented analysis and design* (*OOAD*) dan menggunakan bahasa unified modeling language (*UML*) yang konsisten untuk menentukan, visualisasi, mengkonstruksi, dan mendokumentasikan artifact yang terdapat dalam sistem perangkat lunak. Sebelum adanya *UML*, terdapat ketidakjelasan bahasa pemodelan apa yang paling unggul, para pengembang harus memilih diantara bahasa pemodelan dan tool pemodelan yang banyak dan mirip.

UML dibuat untuk membuat integrasi baru dalam bahasa pemodelan antara alat bantu (tool) dan proses. *UML* hanya mendefinisikan semantic dan notasi, bukan sebuah tool. Contoh tool yang menggunakan *UML* sebagai bahasanya adalah *Rational Rose* dan *Enterprise Architect*.

Bank Perkreditan Rakyat (BPR) bila dilihat dari lalu lintas informasinya memiliki tranfer data yang sangat tinggi dari hari ke hari, sedangkan jika dilihat

sebagai satuan sistem, BPR memiliki sub-sistem yang dapat menginteraksi satu sama lainnya, saling bekerja sama dengan tujuan memberikan pelayanan perbankan bagi konsumen dalam hal ini masyarakat, dengan demikian dibutuhkan suatu sistem informasi yang efisien, cepat dan akurat.

PT. BPR Shinta Daya, Yogyakarta, adalah bank perkreditan yang terfokus pada pelayanan jasa perbankan bagi masyarakat golongan ekonomi lemah pedesaan, dalam sistem informasinya masih banyak digunakan sistem semi-manual, ini bisa mengakibatkan tercecernya suatu informasi, kebocoran dana, dan transparannya data dan lain sebagainya, sehingga pihak manajemen mengalami kesulitan dalam kontrol manajemennya.

Sistem penggajian karyawan yang berdasarkan jam kerja membutuhkan suatu ketelitian, kesabaran, dan juga sumber daya manusia yang lebih banyak dalam perhitungan dan implementasinya, bila hal tersebut dilakukan secara semi otomatis maupun manual. Jika terjadi kesalahan dalam memasukkan data, maka salah satu pihak (perusahaan atau karyawan) akan diuntungkan dan pihak yang lain akan dirugikan. Kesalahan semacam ini sering terjadi sehingga untuk mengatasinya perlu diterapkan suatu sistem yang secara otomatis dapat memasukan dan menghitung atau memproses data.

Data yang harus dimasukkan secara terus menerus adalah data jam masuk dan jam pulang karyawan setiap harinya. Cara yang paling banyak dipakai untuk merekam data ini ada menggunakan kartu absensi dengan Check Clock.

Kelemahan sistem ini adalah pihak personalia perusahaan harus rajin dan telaten merekam data pada kartu absensi ke dalam komputer setiap harinya, karena jika tidak pekerjaan pemasukan data akan semakin banyak dan melelahkan. Akibatnya kemungkinan terjadinya pemasukan data yang salah akan semakin besar.

Pemecahan masalah untuk hal tersebut di atas adalah dengan menggunakan kartu absensi magnetik yang *On Line* ke komputer. Sistem seperti ini sebenarnya sudah ada akan tetapi masih terbatas untuk pengecekan absensi saja, bukan untuk keperluan yang lebih luas seperti perhitungan gaji untuk karyawan.

Perancangan dan pemodelan sistem informasi penggajian menggunakan *tool* Rational Rose mencoba untuk membangun sistem perangkat lunak yang mendukung otomatisasi serta mengintegrasikan data mengenai layanan penggajian karyawan secara transparan dalam aliran data maupun akurasi data di BPR Shinta Daya.

Berdasarkan uraian diatas, maka penulis mengambil judul **Perancangan Sistem Informasi Berorientasi Obyek dengan Metode Unified Modelling Language Pada Sistem Penggajian Di BPR Shinta Daya Yogyakarta.**

1.2 Rumusan Masalah

Berdasarkan latar belakang dan permasalahan yang dihadapi BPR Shinta Daya dapat dirumuskan suatu masalah yaitu:

Bagaimana merancang dan mendesain suatu Sistem Penggajian Karyawan di BPR, menggunakan bahasa standar analisis dan desain Objek yaitu UML.

1.3 Batasan Masalah

Skripsi ini tidak membahas mengenai kartu absensi magnetik dan alat untuk merekam data jam masuk dan pulangny karyawan, akan tetapi hanya membahas mengenai sistem informasi penggajian karyawan yang berlaku di PT. BPR Shinta Daya Yogyakarta.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Merancang suatu sistem absensi serta penggajian BPR yang saling terintegrasi.
2. Mencoba menerapkan teknik desain menggunakan tool yang dapat memudahkan dalam rekayasa perangkat lunak yang nantinya akan memudahkan dalam round-trip engineering.

1.5 Manfaat Penelitian

Dalam hasil dan penyusunan Tugas Akhir ini diharapkan dapat diperoleh manfaat bagi institusi bank dan juga pihak pengembang perangkat lunak yaitu:

1. Memperkenalkan paradigma baru dalam rekayasa perangkat lunak yang menjadi landasan bagi UML.
2. Memberikan pemahaman pemodelan berorientasi objek menggunakan tool analisis dan desain.

3. Sebagai bahan evaluasi dalam mempelajari pemodelan berorientasi objek menggunakan UML, yang diterapkan dalam pembuatan sistem informasi penggajian karyawan.

1.6 Sistematika Penulisan

Sistematika Penulisan dalam laporan tugas akhir ini adalah sebagai berikut:

BAB I Pendahuluan

Berisi latar belakang masalah, perumusan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II Landasan Teori

Pada bab ini diuraikan tentang teori-teori yang digunakan sebagai dasar penelitian, analisis, dan desain sistem.

BAB III Gambaran Umum Perusahaan

Berisikan tentang riwayat singkat perusahaan, visi dan misi organisasi, struktur organisasi dan *job description*, personalia, bidang usaha perusahaan, dan sistem penggajian perusahaan.

BAB IV Perancangan Sistem, Hasil, dan Pembahasan

Membahas tentang langkah-langkah analisis dan perancangan dari sistem penggajian yang akan dibuat dari pemodelan kelas, pemodelan fungsional, masukan, dan keluaran serta membahas tentang hasil perancangan sistem penggajian dengan UML .

BAB V Kesimpulan dan Saran

Pada bab ini berisikan kesimpulan dari penyusunan skripsi serta saran-saran yang bermanfaat bagi perbaikan kinerja perusahaan.

BAB II LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

Dengan berkembangnya suatu organisasi perusahaan, maka masalah yang dihadapi suatu perusahaan semakin kompleks dan ruang lingkungannya akan semakin luas. Agar perusahaan dapat merencanakan, mengkoordinasikan, dan mengendalikan berbagai aktivitas perusahaan, maka diperlukan suatu sistem informasi yang akan menjadi sarana bagi perusahaan di dalam pengambilan keputusan. Suatu sistem informasi yang baik dalam perusahaan, merupakan sumber utama bagi manajemen didalam memenuhi kebutuhan akan informasi.

Sebelum melangkah lebih jauh untuk memberikan tinjauan secara menyeluruh tentang perancangan sistem, pertama-tama akan dipaparkan terlebih dahulu mengenai apa itu definisi dari Sistem Informasi tersebut.

2.2 Definisi Sistem Informasi

Sistem Informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer, serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut. Selain itu data juga memegang peranan yang penting dalam sistem informasi (Andri Kristanto, 2003). Data yang akan dimasukkan dalam sebuah sistem informasi dapat berupa formulir-formulir, prosedur-prosedur dan bentuk data lainnya.

Sistem Informasi dapat juga didefinisikan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen - komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Menurut Andri Kristanto (2003) definisi sistem informasi adalah “Sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan –laporan yang diperlukan”.

Hal ini menunjukkan atau mengindikasikan bahwa begitu penting dan perlu adanya suatu sistem informasi dalam suatu perusahaan, yang nantinya digunakan sebagai salah satu alat untuk pencapaian tujuan dari suatu perusahaan.

2.3 Definisi Sistem dan Prosedur

Menurut Zaki Baridwan (1991), Sistem adalah suatu kerangka dari prosedur-prosedur yang saling berhubungan yang disusun dengan skema yang menyeluruh untuk melaksanakan suatu kegiatan atau fungsi utama dari perusahaan. Sedangkan Prosedur adalah urutan pekerjaan kerani (clerical), biasanya melibatkan beberapa orang dalam bagian atau lebih, disusun untuk menjamin adanya perlakuan yang seragam terhadap transaksi-transaksi perusahaan yang sering terjadi.

Dari definisi diatas dapat diperjelas bahwa sistem adalah kerangka dari prosedur-prosedur yang saling berhubungan untuk melaksanakan kegiatan perusahaan,

sedangkan prosedur adalah urutan pekerjaan yang melibatkan beberapa orang untuk melakukan kegiatan perusahaan yang telah ditetapkan perusahaan.

2.4 Sistem dan Prosedur Penggajian

Pengertian gaji menurut Hani Handoko (1992) adalah pembayaran tetap secara bulanan atau mingguan kepada karyawan klerikal, administratif, manajerial, dan professional.

Sedangkan menurut A .S Ruky (1990) mendefinisikan Gaji sebagai pembayaran yang ditetapkan bulanan, misalnya sebagai pembayaran yang ditetapkan bulanan yaitu untuk pegawai administratif, supervisor, manajer dan lain-lain.

Walaupun cara pembayarannya diatur dalam dua kali sebulan, gaji biasanya ditetapkan berdasarkan perjanjian antara majikan dan pegawai, disamping itu ada yang ditetapkan oleh pemerintah berdasarkan golongan. Sehingga ada banyak faktor yang menentukan baik secara langsung maupun tidak langsung tingkat gaji untuk bermacam-macam jabatan. Meskipun pengaruh masing-masing faktor tersebut sulit ditentukan secara tepat, namun faktor-faktor tersebut harus dipertimbangkan dengan seksama.

Adapun faktor-faktor yang menentukan besarnya gaji adalah sebagai berikut (Rukhyat Kosasih, 1992) :

- a. Tingkat upah umum dalam masyarakat khususnya ditingkat industri yang bersangkutan.

- b. Kebutuhan pokok tenaga kerja dan tingkat biaya hidup atau standar hidup beserta statistik perubahan-perubahannya.
- c. Kualitas tenaga kerja yang diperlukan oleh perusahaan secara efektif.
- d. Persaingan dari perusahaan-perusahaan lain dalam usaha memperoleh tenaga kerja dengan kualitas yang diinginkan.

Dari beberapa faktor yang disebutkan diatas, bahwa faktor yang menentukan besarnya gaji adalah tingkat upah umum yang berlaku di perusahaan sejenis dan tempat perusahaan beroperasi, tingkat biaya hidup, kualitas tenaga kerja dan persaingan antar perusahaan untuk mendapatkan kualitas tenaga kerja yang diinginkan. Hal ini sangat penting sehingga nantinya baik karyawan maupun perusahaan tidak saling merugikan.

2.4.1 Fungsi Bagian Gaji

Fungsi bagian gaji adalah sebagai berikut (Zaki Baridwan,1991):

- a. Menghitung gaji. Adapun prosedurnya adalah sebagai berikut:
 - 1. Mengumpulkan catatan waktu hadir dari bagian pencatat waktu
 - 2. Mengumpulkan data yang diperlukan untuk menghitung pendapatan karyawan.
 - 3. Menambah tunjangan-tunjangan pada gaji.
 - 4. Mengalikan waktu hadir dengan tarif upah.
 - 5. Memperhitungkan potongan-potongan pada gaji seperti angsuran pinjaman, asuransi dan lain sebagainya.

- b. Membuat formulir dan laporan yang diperlukan sebagai berikut:
 - 1. Jurnal gaji
 - 2. Cek gaji dan amplop gaji.
 - 3. Laporan gaji karyawan.
 - 4. Catatan gaji karyawan.
 - 5. Formulir atau laporan yang diperlukan untuk menghitung pajak dan asuransi.
- c. Menyusun statistik gaji.
- d. Memelihara arsip-arsip yang diperlukan.

2.4.2 Sistem dan Prosedur Akuntansi Gaji

Sistem penggajian dalam perusahaan meliputi:

- a. Fungsi Kepegawaian

Fungsi ini berfungsi bertanggung jawab terhadap pengangkatan karyawan, penetapan jabatan, penetapan gaji, promosi, dan penurunan pangkat, mutasi karyawan, pembarhantian karyawan dari pekerjaannya dan penetapan berbagai tunjangan kesejahteraan karyawan serta penghitungan gaji karyawan.
- b. Fungsi Keuangan

Fungsi ini bertanggung jawab atas pelaksanaan pembayaran gaji dan berbagai tunjangan kesejahteraan karyawan.

c. Fungsi Akuntansi

Fungsi ini bertanggung jawab atas pencatatan biaya tenaga kerja dan distribusi tenaga kerja untuk kepentingan harga pokok produk penyediaan informasi guna pengawasan biaya tenaga kerja, (Mulyadi, 1993).

Dari sistem akuntansi gaji diatas dapat diperjelas bahwa sistem akuntansi ini meliputi fungsi kepegawaian yang bertanggung jawab terhadap pengangkatan sampai pemberhentian karyawan, fungsi keuangan bertanggung jawab terhadap pembayaran gaji dan fungsi akuntansi bertanggung jawab atas pencatatan biaya tenaga kerja. Semua fungsi ini telah mempunyai tugas dan tanggung jawab masing-masing.

2.5 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek adalah cara memandang persoalan menggunakan model-model yang diorganisasikan seputar konsep objek yang mengkombinasikan struktur data dan perilaku suatu entitas. Pendekatan ini menawarkan pemakaian bersama di beberapa tingkat berbeda. Pendekatan berorientasi objek tidak hanya memungkinkan informasi dipakai bersama didalam satu aplikasi, tapi juga penggunaan ulang (reuse) rancangan dan kode pada proyek-proyek berikutnya.

Pengembangan sistem berorientasi objek memerlukan keterampilan untuk analisis, perancangan, dan pemrograman berorientasi objek. Pemahaman pendekatan objek sangat diperlukan dalam pengembangan sistem berorientasi objek. Analisis dan perancangan berorientasi objek merupakan bagian kritis dalam pengembangan sistem berorientasi objek.

2.6 Object Oriented Analysis and Design (OOAD)

Object-oriented mencakup bidang aplikasi yang sangat luas. Para pengguna sistem komputer dan sistem lain berbasis teknologi komputer merasakan efek *object-oriented* dalam bentuk meningkatnya aplikasi software yang mudah digunakan dan servis yang lebih flexibel, yang muncul dalam berbagai industri, seperti dalam perbankan, telekomunikasi, dan sebagainya. Sedangkan bagi software engineer, *object-oriented* berpengaruh dalam bahasa pemrograman, metodologi rekayasa, manajemen proyek, hardware dan sebagainya.

Object-Oriented analysis adalah metode analisis yang memeriksa *requiremen* (syarat yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek - objek yang ditemui dalam ruang lingkup permasalahan. Sedangkan *object-oriented design* adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek- objek sistem atau subsistem (A. Suhendar dan Hariman Gunadi, 2002).

Proses analisis dan perancangan ini merupakan suatu proses yang berkelanjutan: proses analisis kemudian proses perancangan. Tujuannya untuk mendapatkan domain masalah dan pemecahan logis atas masalah dari kaca mata teknologi objek.

2.6.1 Konsep Dasar dalam OOAD

Menurut A.Suhendar dan Hariman Gunadi (2002) teknologi *object-oriented* merupakan paradigma baru dalam rekayasa software yang didasarkan pada objek dan kelas. Dibawah ini akan dijelaskan secara umum mengenai:

2.6.1.1 Objek

Objek adalah benda yang secara fisik atau konseptual dapat ditemui di sekeliling kita. Sebuah objek memiliki keadaan sesaat (state) dan perilaku (behavior). Keadaan sesaat dari sebuah objek adalah kondisi objek tersebut atau himpunan dari keadaan yang menggambarkan objek tersebut. Perilaku suatu objek mendefinisikan bagaimana sebuah objek beraksi dan memberi reaksi. Perilaku ini ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan dalam objek itu sendiri.

Sedangkan Erwin (2001) menyebutkan bahwa objek adalah segala sesuatu yang mempunyai ciri-ciri sebagai berikut :

1. *Identitas* adalah nama obyek yang digunakan untuk membedakan objek yang satu dengan objek yang lain
2. *Atribut* merupakan karakteristik dari suatu objek. Dalam bahasa pemrograman, misalnya java, atribut diimplementasikan sebagai data atau variable dari suatu obyek.
3. *Perilaku*. Karakteristik dari perilaku disebut dengan operasi. Dalam bahasa pemrograman, misalnya java, perilaku atau operasi diimplementasikan sebagai fungsi atau metode dari suatu obyek.

2.6.1.2 Kelas

Menurut Alhir (1998), kelas adalah deskripsi dari objek-objek dengan implementasi umum. Kelas berkaitan dengan implementasi dari karakteristik struktural dan perilaku yang sama. Dengan demikian kelas merupakan deskripsi

dari objek-objek dengan atribut umum, implementasi operasi, semantik, asosiasi, dan interaksi. Sebagai contoh, orang termasuk dalam kelas manusia dan mobil termasuk dalam kelas kendaraan bermotor.

2.6.1.3 Association dan Aggregation

Association adalah hubungan antar objek yang saling membutuhkan. Sedangkan, *aggregation* adalah bentuk khusus dari asosiasi yang menggambarkan seluruh bagian suatu objek merupakan bagian dari objek lainnya. Sebagai contoh, objek tanggal dapat disusun dari objek hari, objek bulan, dan objek tahun.

2.7 Metodologi Berorientasi Objek

Metodologi berorientasi objek adalah suatu metodologi pengembangan perangkat lunak dimana dalam melakukan analisis, perancangan, dan pemrograman serta implementasinya meniru kehidupan nyata, yaitu segala sesuatunya dipandang sebagai objek (Erwin, 2001).

Metodologi berorientasi objek mempunyai keuntungan, yaitu (Bray, 1997) :

1. *Maintainability*. Dibangun sesuai dengan kenyataan yang ada (dunia nyata) sehingga memudahkan dalam menganalisa, dapat mengurangi kerumitan dalam perancangan sistem, dan mudah diverifikasi (dibuktikan) oleh pengguna.
2. *Reusability* ini dapat menghemat waktu dan biaya karena rancangan yang sudah ada, baik model maupun implementasinya, dapat digunakan kembali dalam pengembangan sistem.

3. *Productivity*. Meningkatkan produktivitas dimana keuntungan ini diperoleh langsung dari penggunaan *Object Oriented Programming Language* (OOPL).

2.8 Pemilihan Metodologi

Berikut ini merupakan beberapa metodologi yang sering digunakan dalam *object-oriented analysis and design* (OOAD) adalah sebagai berikut :

1. *Object Oriented Design with Applications* (OODA) oleh Booch.
2. *Object Oriented Analysis and Object Oriented Design* (OOA/OOD) oleh Coud dan Yourdon.
3. *Object Oriented Analysis and Design* (OOAD) oleh Martin dan Odell.
4. *Object Modeling Technique* (OMT) oleh James Rumbaugh.
5. *Object Oriented Systems Analysis* (OOSA) oleh Shlaer dan Mellor.
6. *Designing Object Oriented Software* (DOOS) oleh Wirfs Brock.
7. *Object-Oriented Software Engineering* (OOSE) oleh Jacobson.

Menurut Quatrani (2001), metodologi OMT milik Rumbaugh mempunyai keunggulan dalam penotasian yang mendukung semua konsep berorientasi objek. Metode ini sangat kuat dalam analisis tetapi lemah dalam perancangan. Sedangkan metodologi milik Booch kelebihanannya adalah sangat detil dan kaya dengan notasi dan elemen gabungan dari metode lain. Metode ini sangat kuat dalam perancangan tetapi lemah dalam analisis.

Untuk metodologi milik Jacobson menurut Quatrani (2001), keunggulannya adalah mudah untuk dipelajari karena memiliki notasi yang sederhana, dan mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Ketiga orang ini kemudian bergabung dengan Rational, sebuah perusahaan perangkat lunak, dan saling bekerja sama sehingga menghasilkan suatu bahasa pemodelan yang standart, yaitu *Unified Modeling Language* (UML). UML ini kemudian dikirim ke *Object Management Group* (OMG), sebuah organisasi internasional yang dibentuk pada tahun 1989 dan didukung oleh lebih dari 800 anggota yang terdiri dari perusahaan sistem informasi, pengembang perangkat lunak, dan para pemakai sistem komputer, untuk distandarisasi (Quatrani, 2001).

OMG menyebutkan bahwa UML adalah bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasi artifak dari suatu sistem perangkat lunak. Bahasa standart ini menurut Quatrani (2001) dapat mengatasi masalah penotasian dan mekanisme tukar menukar model antar metodologi yang sering dilakukan dalam proses pengembangan Software.

2.9 UML

Dalam memodelkan suatu sistem menetapkan bahwa UML menggunakan diagram-diagram sebagai berikut :

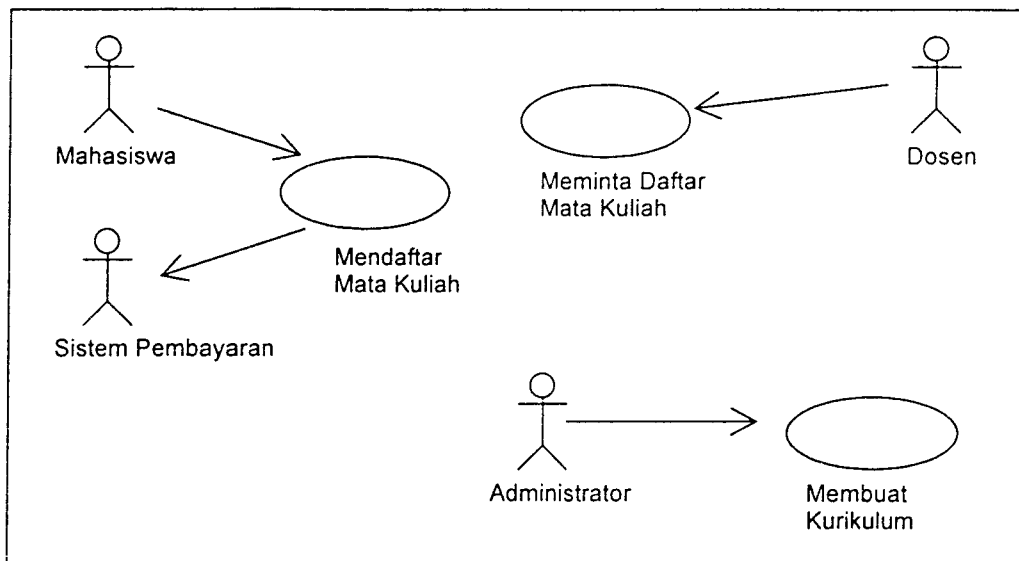
1. *Use Case Diagram*
2. *Class Diagram*
3. *Behaviour Diagram*, yang terdiri dari :

- a. *Statechart Diagram*
 - b. *Activity Diagram*
 - c. *Interaction Diagram*, yang terdiri dari *Sequence Diagram* dan *Collaboration Diagram*.
4. *Implementation Diagram*, yang terdiri dari :
- a. *Component Diagram*
 - b. *Deployment Diagram*

2.9.1 Use Case Diagram

Use Case Diagram (UCD) menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem (Julius Hermawan, 2004). UCD ini memberikan gambaran statis dari sistem yang dibangun dan dapat digunakan selama proses analisis untuk menangkap kebutuhan-kebutuhan sistem dan untuk memahami bagaimana sistem seharusnya bekerja.

Dalam diagram ini diperlihatkan relasi yang terjadi di antara *Use Case-Use Case* sistem dan aktor-aktor yang terlibat dalam sistem serta komunikasi yang terjadi. *Use Case Diagram* ini bermanfaat untuk menentukan semua kebutuhan dari sistem dan untuk berkomunikasi dengan pelanggan.



Gambar 2.1. *Use Case Diagram* Sistem Pendaftaran Mata Kuliah

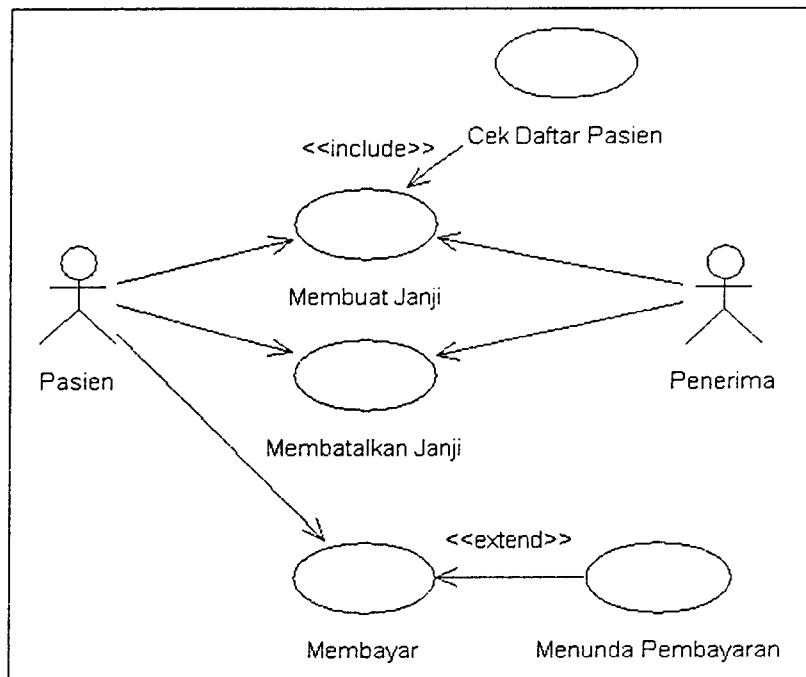
Sebelum membuat *Use Case Diagram*, pertama-tama yang dilakukan adalah dengan mengetahui aktor yang terlibat dalam sistem. Aktor adalah sesuatu yang berinteraksi dengan sistem, baik pengguna, mesin, maupun alat. Aktor bahkan bisa berupa proses lain yang berinteraksi dengan sistem yang bersangkutan. Dalam Gambar 2.1 terlihat aktor-aktor yang terlibat dalam sistem, dalam hal ini sistem pendaftaran mata kuliah. Pertama kali yang dilakukan dalam membuat *Use Case Diagram* adalah siapa (aktor) yang berinteraksi dengan sistem. Contoh dalam Gambar 2.1, aktor yang berinteraksi dengan sistem adalah Administrasi, Dosen, Mahasiswa, dan Sistem Pembayaran. Dengan demikian jelas bahwa aktor tidak harus manusia akan tetapi juga dapat berupa apa saja dimana dia berinteraksi dengan sistem dan berada diluar dari sistem tersebut.

Setelah mengetahui aktor-aktor yang terlibat dalam sistem, barulah dibuat *Use Case*. Caranya adalah dengan melihat satu persatu dari aktor yang ada dan

kemudian menganalisis mengapa aktor-aktor tersebut ingin menggunakan sistem. Menurut Quatrani (2001), *Use Case* adalah rangkaian atau urutan dari transaksi-transaksi (aktifitas) yang saling berelasi dan dimainkan atau dilakukan oleh aktor dalam sistem. *Use Case* bukan merupakan sebuah modul dari perangkat lunak, akan tetapi sesuatu yang bernilai dan dilakukan oleh aktor. Dalam sistem pendaftaran mata kuliah ini Administrator membuat kurikulum, Dosen meminta daftar mata kuliah, Mahasiswa melakukan pemilihan mata kuliah, dan Sistem Pembayaran menerima informasi pengajuan mata kuliah. Selanjutnya dengan melihat dari sudut pandang pelanggan, dibuat *Use Case* dengan cara mengetahui mengapa aktor-aktor tersebut ingin menggunakan sistem dan keuntungan apa yang didapat dari sistem tersebut.

Antara *Use Case* yang satu dengan *Use Case* yang lainnya dapat mempunyai hubungan asosiasi (Sparks, 2004). Hubungan asosiasi tersebut biasanya mempunyai *Stereotype* yang digunakan untuk menerangkan sub klasifikasi dari suatu elemen model. *Stereotype* biasanya mempunyai definisi “*extend*” atau “*include*”. *Stereotype* “*extend*” mempunyai arti bahwa suatu *Use Case* merupakan kelanjutan dari *Use Case* lain jika suatu syarat dipenuhi. Sedangkan *Stereotype* “*include*” berarti suatu *Use Case* termasuk dalam alur kerja dari *Use Case*.

Gambar 2.2 berikut ini merupakan contoh *Use Case Diagram* dengan hubungan asosiasi *Stereotype* “*extend*” dan “*include*”.



Gambar 2.2. Contoh *Use Case Diagram* dengan *Stereotype*

Pada Gambar 2.2 diatas hubungan asosiasi antara *Use Case* Cek Daftar Pasien dan *Use Case* Membuat Janji adalah hubungan asosiasi dengan *Stereotype* “include”. Hal ini berarti bahwa dalam *Use Case* Membuat Janji penerima juga melakukan pengecekan pada daftar pasien, yaitu apakah pasien tersebut merupakan pasien baru atau tidak. Dengan demikian *Use Case* Cek Daftar Pasien termasuk dalam alur kerja dari *Use Case* Membuat Janji.

Sedangkan hubungan asosiasi dengan *Stereotype* “extend” antara *Use Case* Menunda Pembayaran dan *Use Case* Membayar menunjukkan bahwa *Use Case* Menunda Pembayaran merupakan kelanjutan dari *Use Case* Membayar apabila pasien yang bersangkutan ingin melakukan pembayaran di lain waktu.

Langkah selanjutnya adalah mendokumentasikan *Use Case* setelah interaksi antara aktor dan sistem diidentifikasi. Setiap *Use Case* perlu didokumentasikan bersamaan dengan aliran kejadiannya. Dengan demikian dari sudut pandang aktor, sistem yang dirancang telah selesai. Sebaiknya detail atau perincian dari sistem harus tersedia untuk aktor ketika *Use Case* dieksekusi atau diterapkan, contohnya kapan *Use Case* dimulai dan kapan akan berakhir. Oleh karena itu *Use Case Diagram* sangat erat hubungannya dengan skenario (Miller, 2003). Skenario adalah suatu deskripsi apa yang terjadi ketika seseorang berinteraksi dengan sistem.

Dengan demikian dapat diambil kesimpulan bahwa *Use Case Diagram* merupakan cara yang baik untuk menyakinkan dan menunjukkan bahwa sistem yang dibuat adalah benar-benar merupakan keinginan dari pelanggan karena *Use Case Diagram* memperlihatkan aktor, *Use Case*, dan hubungan yang terjadi diantara mereka. *Use Case Diagram* juga merupakan diagram yang cocok untuk melihat sistem secara keseluruhan. memperlihatkan eksternal dari sistem (aktor) dan kemampuan yang dihasilkan atau disediakan oleh sistem (*Use Case*).

2.9.2 Class Diagram

Class Diagram memberi gambaran dari sistem dengan memperlihatkan kelas-kelas yang ada dan relasi atau interaksi yang terjadi diantara kelas-kelas tersebut (Miller, 2003). *Class Diagram* adalah diagram yang statis karena menampilkan interaksi yang terjadi tetapi tidak menampilkan apa yang terjadi ketika kelas-kelas

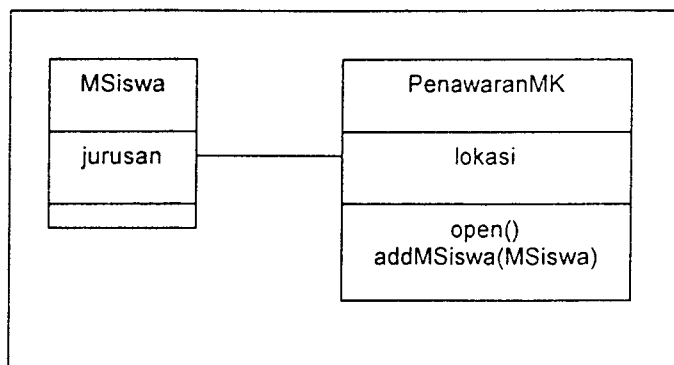
tersebut berinteraksi. *Class Diagram* juga memperlihatkan atribut dan operasi pada setiap kelas yang ada.

Dalam UML sebuah kelas mempunyai notasi berupa empat persegi panjang yang dibagi menjadi tiga bagian, yaitu untuk nama kelas, atribut, dan operasi. Atribut dan operasi dari suatu kelas dapat memiliki salah satu sifat berikut ini, yaitu :

1. *Private*, hanya dapat digunakan oleh kelas yang bersangkutan.
2. *Protected*, hanya dapat digunakan oleh kelas yang bersangkutan dan kelas-kelas turunannya.
3. *Public*, dapat digunakan oleh siapa saja, baik dari kelas yang bersangkutan maupun dari luar kelas.

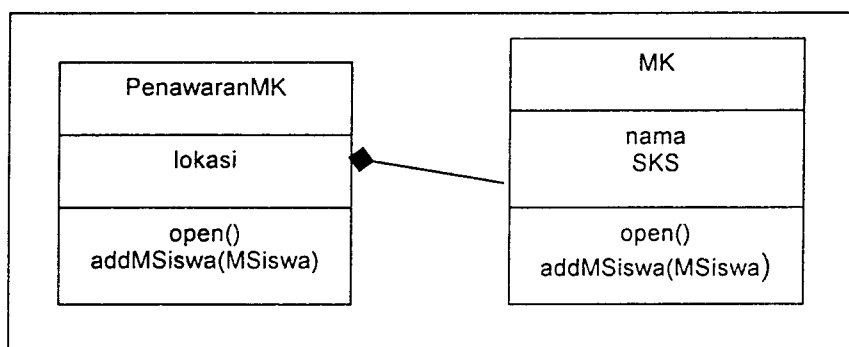
Dalam *Class Diagram* relasi yang terjadi dapat terdiri dari empat jenis, yaitu :

1. *Assosiation* adalah relasi yang terjadi antara dua kelas dimana relasi ini merupakan relasi dua arah dan hanya dapat terjadi jika salah satu kelas memahami tentang kelas yang lainnya. Hal tersebut menurut Quatrani (2001) berarti “Saya dapat mengirim kepada anda sebuah pesan, karena jika saya berelasi asosiasi dengan anda, saya tahu anda ada”. Asosiasi digambarkan sebagai garis lurus. Pada Gambar 2.3 berikut merupakan contoh relasi asosiasi dimana pesan yang dikirim dari kelas MSiswa ke kelas PenawaranMK adalah mengambil mata kuliah.



Gambar 2.3. Relasi Asosiasi dalam *Class Diagram*

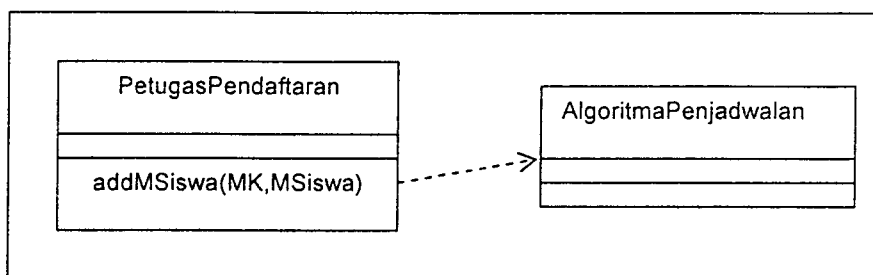
2. *Aggregation* adalah suatu asosiasi dimana suatu kelas merupakan bagian dari kelas yang lain atau boleh dikatakan suatu kelas “terdiri atas” (Miller, 2003). Misalnya pada Gambar 2.4 berikut terlihat bahwa kelas MK dibuat berdasarkan dari kelas PenawaranMK. Agregasi digambarkan sebagai garis lurus dimana salah satu ujungnya terdapat belah ketupat.



Gambar 2.4. Relasi Agregasi dalam *Class Diagram*

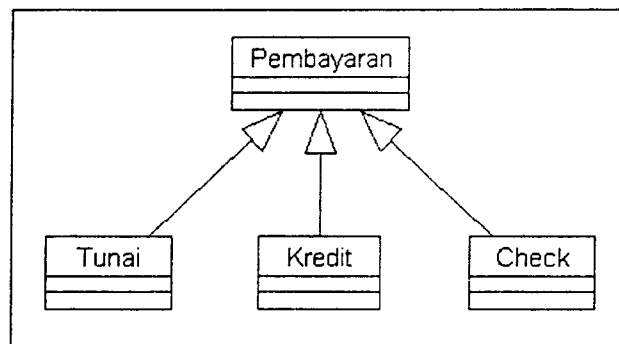
Asosiasi dan agregasi merupakan komunikasi dua arah, akan tetapi seringkali diinginkan untuk membatasinya dalam satu arah. Jika demikian maka harus ditambahkan kepala panah untuk menunjukkan arah komunikasi. Inilah yang disebut dengan Navigasi.

3. *Dependency* digunakan apabila terdapat hubungan yang lemah antara dua kelas (Quatrani, 2001). Misalnya hubungan antara kelas Petugas Pendaftaran dan kelas Algoritma Penjadwalan pada Gambar 2.5 di bawah ini, dimana kelas Petugas Pendaftaran tidak mengetahui keberadaan dari kelas Algoritma Penjadwalan dengan pasti. *Dependency* digambarkan dengan anak panah dengan garis putus-putus. Dengan demikian *Dependency* dapat dijelaskan sebagai berikut : Saya membutuhkan pelayanan anda, tetapi saya tidak tahu anda ada atau tidak.



Gambar 2.5. Relasi *Dependency* dalam *Class Diagram*

4. *Generalization* adalah merupakan hubungan pewarisan yang menunjukkan bahwa suatu kelas merupakan *Superclass* dari kelas yang lain (Miller, 2003). Misalnya pada Gambar 2.6 di bawah ini menunjukkan bahwa kelas Pembayaran merupakan *Superclass* dari kelas Kredit, kelas Tunai, dan kelas Check. Generalisasi digambarkan sebagai garis lurus dimana salah satu ujungnya terdapat segitiga.

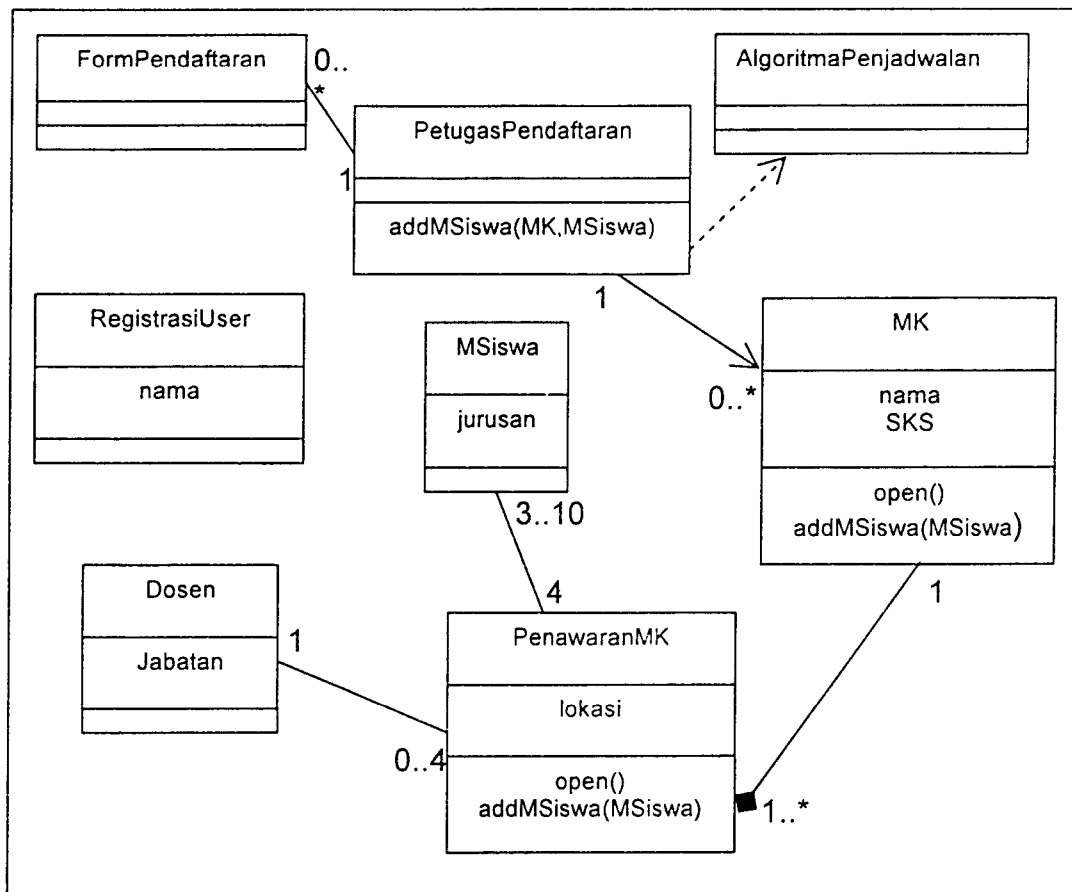


Gambar 2.6. Relasi Generalisasi dalam *Class Diagram*

Dalam *Class Diagram* terdapat elemen *Multiplicity* (keserbaragaman) dan Navigasi. *Multiplicity* menentukan berapa banyak objek yang dibuat dalam suatu relasi. Dengan demikian *Multiplicity* menunjukkan berapa banyak *Instance* dari suatu kelas yang dapat dihubungkan kepada satu *Instance* dari kelas lain. Untuk asosiasi dan agregasi harus terdapat *Multiplicity* pada setiap ujung dari relasi. *Multiplicity* direpresentasikan dengan nomor dan tanda * yang mempunyai arti banyak (Miller, 2003). Pada Tabel 2.1 berikut ini menampilkan indikator *Multiplicity*.

Tabel 2.1. Indikator *Multiplicity*

Relasi	Artinya
0..1	Nol atau Satu <i>Instance</i> . Untuk notasi n..m mengindikasikan n sampai dengan m <i>Instance</i> .
0..* atau *	Tidak terbatas <i>Instance</i> -nya, termasuk Nol.
1	Tepat Satu <i>Instance</i> .
1..*	Paling sedikit Satu <i>Instance</i> .



Gambar 2.7. *Class Diagram* Sistem Pendaftaran Mata Kuliah

Gambar 2.7 diatas merupakan contoh *Class Diagram* dari sistem pendaftaran mata kuliah. Relasi asosiasi dapat dilihat antara kelas Dosen dan kelas PenawaranMK dimana terdapat hubungan atau komunikasi dua arah antara dua kelas tersebut. Dari relasi kedua kelas tersebut juga dapat diketahui bahwa satu objek Dosen dapat berelasi dengan nol sampai dengan empat objek PenawaranMK. Sebaliknya satu PenawaranMK hanya dapat berelasi dengan satu objek Dosen. Dari relasi antara kelas MSiswa (mahasiswa) dan kelas PenawaranMK dapat diketahui bahwa satu mata kuliah yang ditawarkan minimal harus diikuti oleh tiga mahasiswa dan maksimal sepuluh mahasiswa.

Relasi agregasi terlihat antara kelas MK (mata kuliah) dan kelas PenawaranMK, dimana kelas MK dibuat berdasarkan kelas PenawaranMK. Relasi agregasi ini menunjukkan pada perancang bahwa jika mereka ingin merubah kelas MK, mereka mungkin juga harus melakukan sesuatu yang khusus (merubah juga) terhadap kelas PenawaranMK.

2.9.3 Behaviour Diagram

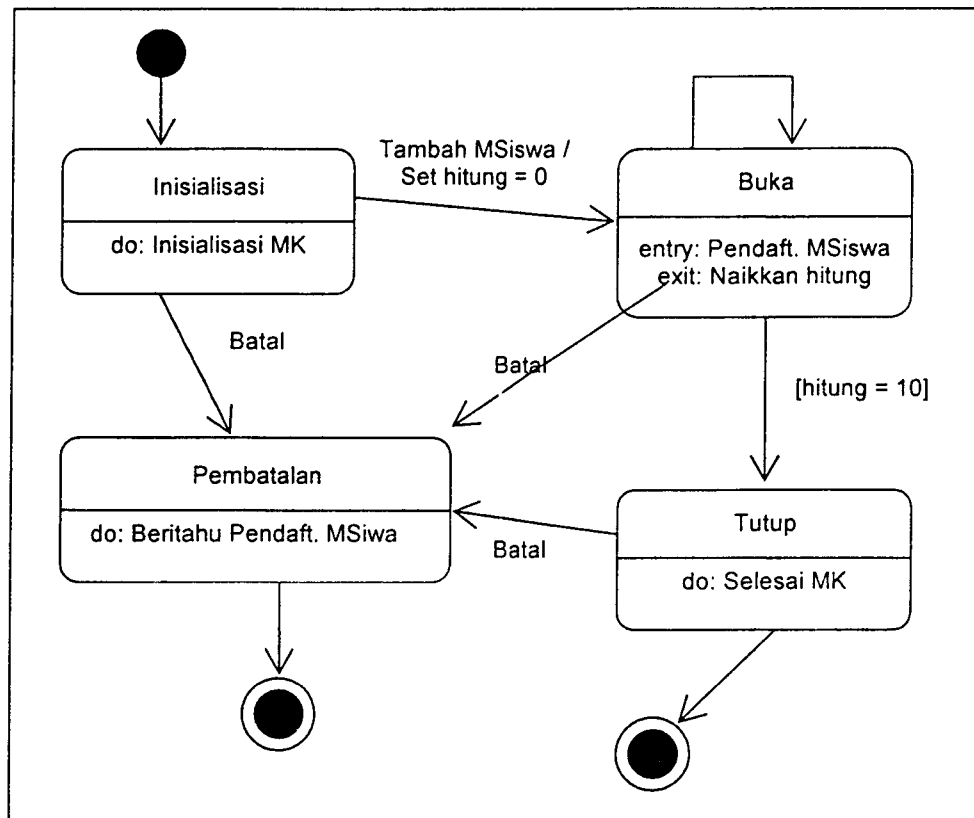
Suatu objek mempunyai perilaku dan keadaan (Miller, 2003). Keadaan dari objek tersebut sangat tergantung pada kondisi maupun aktifitasnya pada suatu saat.

Adapun Behaviour diagram meliputi:

- Statechart Diagram
- Activity Diagram
- Interaction Diagram yang meliputi Sequence diagram dan collaboration diagram

2.9.3.1 Statechart Diagram

Statechart Diagram memperlihatkan kejadian yang menyebabkan transisi dari suatu keadaan ke keadaan yang lain dan aksi atau perilaku yang dihasilkan dari perubahan keadaan tersebut. Diagram ini digunakan untuk kelas yang banyak mempunyai perilaku yang dinamis, sebagaimana disajikan dalam gambar berikut.



Gambar 2.8. *Statechart Diagram* Sistem Pendaftaran Mata Kuliah

Gambar 2.8 di atas merupakan contoh *Statechart Diagram* pendaftaran mata kuliah, dimana keadaan digambarkan sebagai empat persegi panjang dengan keempat sudut-sudutnya membulat, transisi digambarkan dengan anak panah yang menghubungkan antara satu keadaan ke keadaan yang lain, kejadian atau kondisi yang mencerminkan transisi ditulis disamping anak panah, lingkaran hitam penuh menggambarkan titik awal, dan lingkaran hitam setengah menggambarkan titik akhir.

Statechart Diagram dimulai dengan keadaan inisialisasi. Bila seseorang berada di dalam diagram tersebut, maka orang tersebut akan berada dalam keadaan-inisialisasi ini sampai dia mendapat pesan “Tambah MSiswa”. Ketika orang

tersebut mendapat pesan, maka dia mulai penghitungan dari nol dan bertransisi ke keadaan Buka. Pada keadaan Buka ini terdapat *Entry* (masuk) sehingga orang tersebut dapat berada didalamnya. Keadaan Buka tidak perlu mengetahui bagaimana orang tersebut bisa masuk. Keadaan Buka hanya menerima perintah untuk mencatat atau menerima mahasiswa. Ketika orang tersebut keluar dari keadaan Buka, maka penghitung akan menyimpan banyak mahasiswa yang telah masuk atau mendaftar dalam suatu mata kuliah. Sistem dapat menambah mahasiswa sampai sepuluh orang dan berpindah ke keadaan Tutup. Jika kita memilih transisi Batal, maka sistem harus memberitahu mahasiswa dan transisi untuk menghentikan keadaan.

Untuk kelas-kelas yang mempunyai banyak perilaku dinamis menurut Quatrani (2001) hendaknya dibuat *Statechart Diagram* untuk menangani segala sesuatu yang terjadi. Kunci dari diagram ini adalah pertanyaan-pertanyaan sebagai berikut:

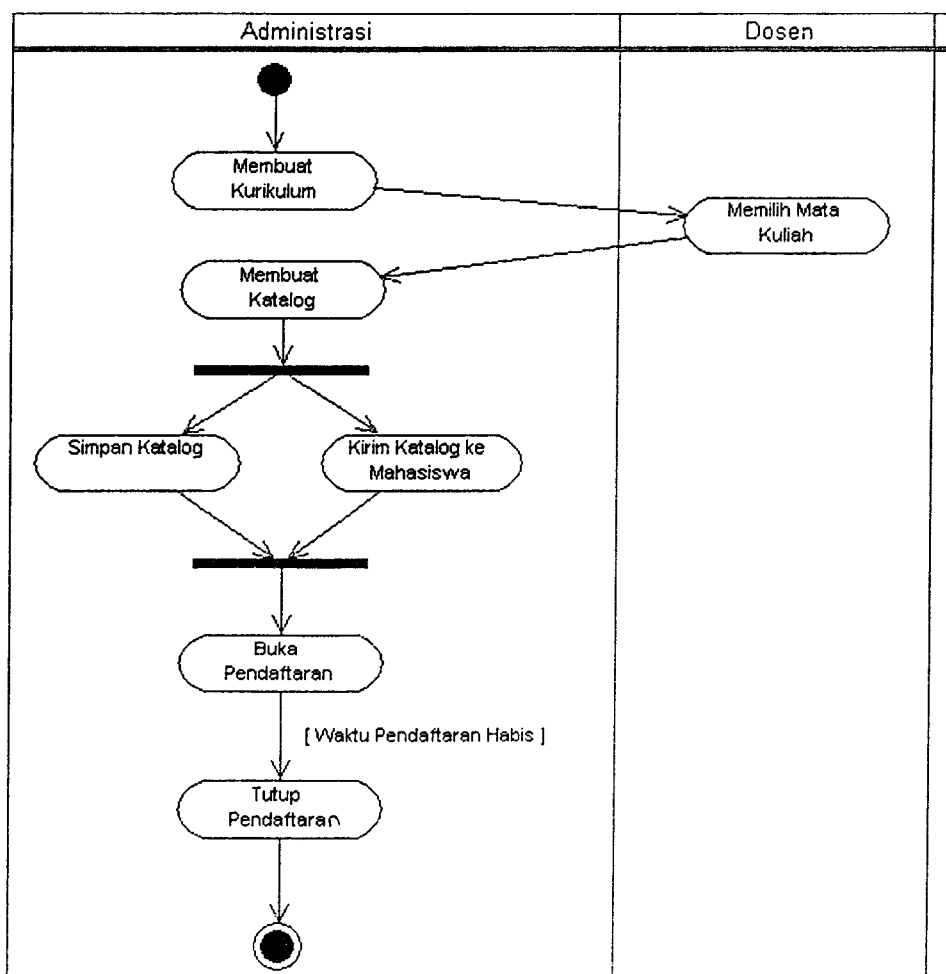
- a. Apa yang terjadi bila kita mendapat atau menerima suatu pesan ?
- b. Apa yang kita kerjakan ketika menerima suatu pesan ?
- c. Pesan apa yang harus kita kirim ?

Pesan-pesan yang ada akan menjadi operasi dari kelas, seperti dalam contoh dimana penambahan mahasiswa (Tambah MSiswa) merupakan suatu operasi. Aksi-aksi seperti perhitungan mahasiswa, penambahan mahasiswa, dan mengecek banyak mahasiswa akan menjadi operasi tersendiri dari kelas yang bersangkutan dan *Statechart Diagram* dimana hal tersebut terjadi. Untuk mengetahui kelas yang

dinamis adalah dengan melihat *Sequence Diagram*. Jika ada kelas yang terdapat dalam banyak *Sequence Diagram* dan mengirim banyak pesan, hal tersebut merupakan indikasi bahwa kelas tersebut merupakan kelas yang dinamis.

2.9.3.2 Activity Diagram

Activity Diagram digunakan untuk memperlihatkan setiap *Use Case* yang berada dalam suatu aktifitas agar dapat diketahui apa yang terjadi pada *Use Case* tersebut. Selain itu juga digunakan untuk memperlihatkan bagaimana aliran yang terjadi diantara *Use Case-Use Case* tersebut (Quatrani, 2001).



Gambar 2.9. Activity Diagram

Gambar 2.9 di atas merupakan contoh *Activity Diagram*. Seperti halnya *Statechart Diagram*, *Activity Diagram* dimulai dengan lingkaran hitam penuh yang menggambarkan titik awal, dan lingkaran hitam setengah menggambarkan titik akhir. Aktifitas digambarkan sebagai empat persegi panjang dengan ujung membulat, dimana aktifitas tersebut secara otomatis akan berlanjut atau berpindah ke aktifitas selanjutnya jika aktifitas yang dilakukan telah selesai. Transisi digambarkan sebagai anak panah yang melukiskan pergerakan dari satu aktifitas ke aktifitas yang lain. Sedangkan Sinkronisasi digambarkan sebagai garis lurus tebal melukiskan suatu aktifitas terjadi secara paralel. Kelebihan dari UML adalah ditambahkannya *Swimlanes* pada *Activity Diagram* ini. *Swimlanes* ini digunakan untuk memperlihatkan kepemilikan dari suatu aktifitas seperti yang digambarkan pada Gambar 2.9.

2.9.3.3 Interaction Diagram

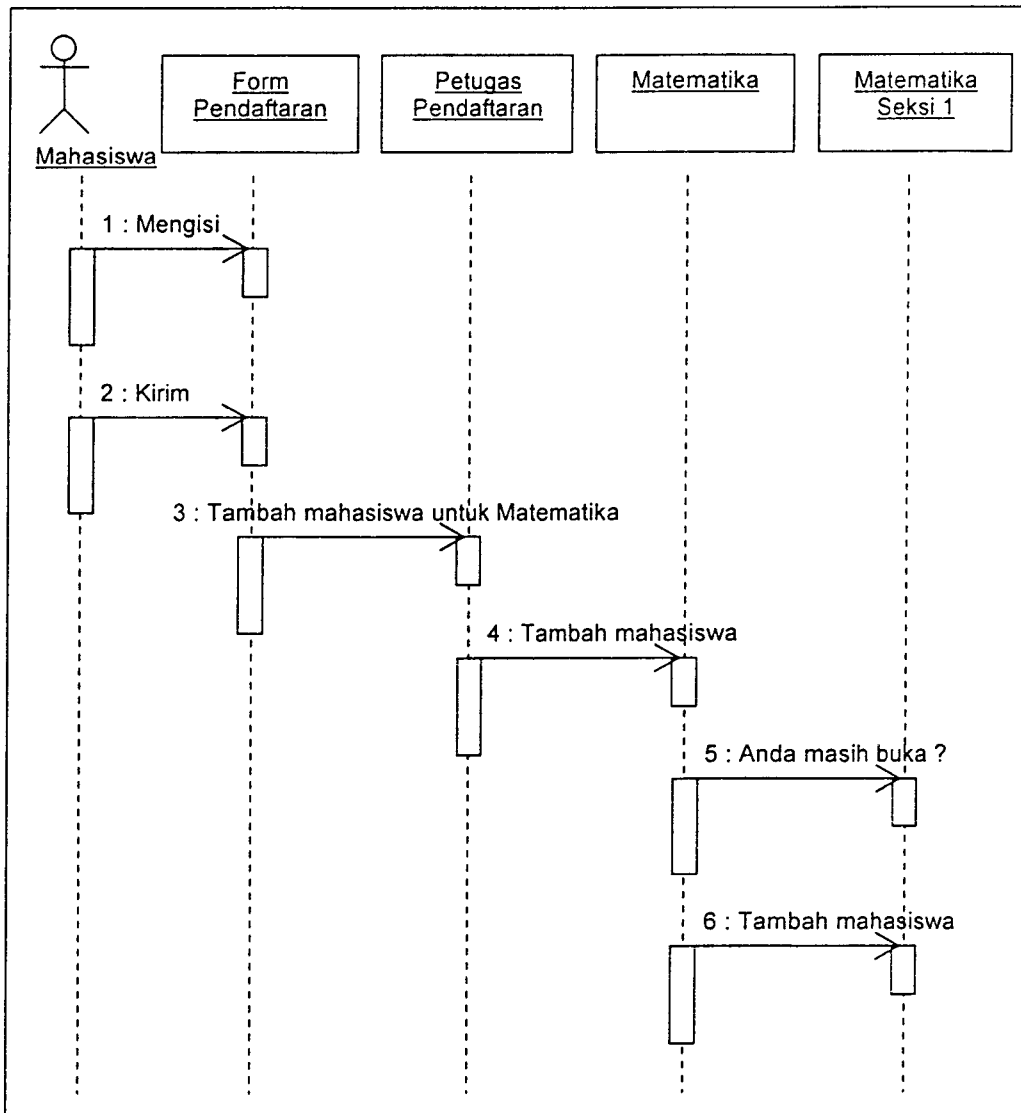
Interaction diagram merupakan aliran kejadian yang dapat menentukan objek dan interaksi yang dibutuhkan. Hal ini meliputi :

- Sequence diagram
- Collaboration Diagram

2.9.3.3.1 Sequence Diagram

Sequence Diagram memperlihatkan interaksi objek yang tersusun secara berurutan berdasarkan waktu kejadian (Quatrani, 2001). Aliran kejadian dapat digunakan untuk menentukan objek dan interaksi apa yang dibutuhkan untuk

menyelesaikan fungsionalitas yang spesifik oleh aliran kejadian. Sebagaimana dicontohkan pada gambar 2.10.



.Gambar 2.10. Sequence Diagram

Contoh *Sequence Diagram* pada Gambar 2.10 memperlihatkan alur kerja mahasiswa dalam memilih mata kuliahnya. Mahasiswa mengisi informasi dan mengirimkan formulir. Formulir tersebut digunakan untuk memerintahkan kepada

petugas pendaftaran agar menambahkan nama mahasiswa pada suatu mata kuliah tertentu yang dipilihnya, misalnya Matematika.

Petugas pendaftaran kemudian memberitahu Matematika bahwa ada penambahan mahasiswa. Matematika mengkonfirmasi kepada Seksi 1 apakah penambahan mahasiswa dapat dilakukan atau tidak. Dalam contoh ini Seksi 1 memberi jawaban bahwa penambahan mahasiswa dapat dilakukan, dengan demikian Matematika memerintahkan Seksi 1 untuk memasukkan atau menambahkan mahasiswa tersebut.

Dengan demikian jelas bahwa *Sequence Diagram* merupakan diagram yang sangat berguna karena dapat memperlihatkan pada perancang dan pelanggan langkah demi langkah apa yang seharusnya terjadi dalam sistem.

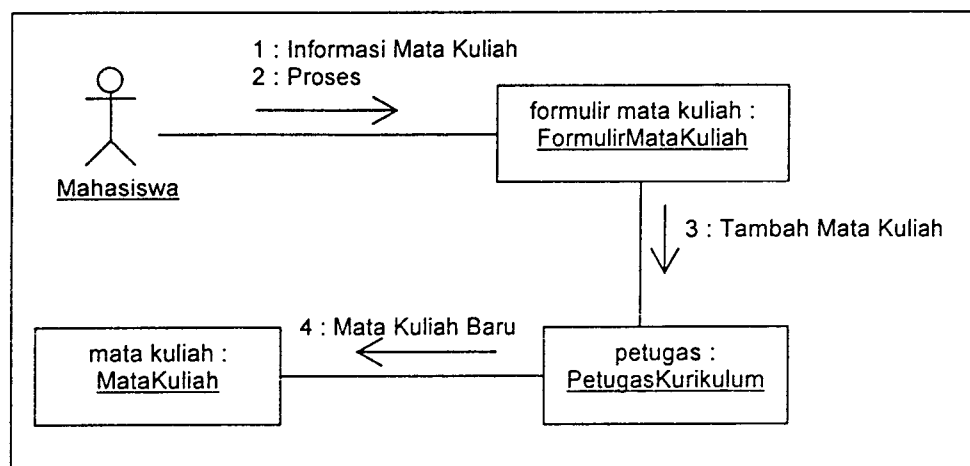
Sequence Diagram sangat membantu dalam menentukan kebutuhan, terutama kebutuhan yang sangat sulit untuk ditemukan, seperti kebutuhan *User Interface* yang dikenal sangat sulit karena sangat tergantung kepada yang menggunakannya. Hal ini disebabkan *User Interface* haruslah *User Friendly*. Akan tetapi *Sequence Diagram* ini dapat membantu dalam menentukan *User Interface* yang cocok atau sesuai.

Dalam perancangan sistem, pembuatan *Sequence Diagram* tidak dapat dipastikan, hanya sampai dirasa telah mencukupi dimana tidak ditemukan lagi objek yang baru. *Sequence Diagram* tidak boleh berulang-ulang. Sebagai contoh pendaftaran mahasiswa pada Matematika akan sama prosesnya dengan pendaftaran mahasiswa

pada Fisika. Aturan lain yang dapat dijadikan acuan adalah membuat *Sequence Diagram* untuk setiap aliran dasar dari setiap *Use Case*.

2.9.3.3.2 Collaboration Diagram

Menurut Quatrani (2001), Collaboration Diagram memperlihatkan interaksi objek yang diorganisasikan disekitar objek dan hubungannya dengan objek yang lain. *Collaboration Diagram* merupakan suatu skenario yang berbeda yang menggambarkan urutan kejadian (pesan) dari satu objek ke objek yang lainnya. *Collaboration Diagram* dan *Sequence Diagram* dapat digunakan secara bergantian karena keduanya sangat mirip. Adapun contoh dari C.D. disajikan pada gambar 2.11.



Gambar 2.11. Collaboration Diagram

Pada gambar 2.11 diatas dijelaskan bahwa Diagram ini sangat membantu jika ingin melihat semua pesan yang terjadi antara dua objek dalam suatu Use Case atau Skenario. Hal ini digambarkan bahwa Seorang Mahasiswa ingin mendapatkan suatu informasi mata kuliah dan proses dengan mendapatkan

formulir mata kuliah, formulir tersebut juga dapat digunakan untuk mengambil tambahan mata kuliah dengan bantuan petugas kurikulum, serta mata kuliah baru.

BAB III GAMBARAN UMUM PERUSAHAAN

3.1 Sejarah Berdirinya Perusahaan

PT. Bank Shinta Daya pernah dikenal pula sebagai Bank Desa Shinta Daya, kini berkembang menjadi sebuah bank perkreditan rakyat (BPR) yang mempunyai jaringan luas di tingkat pedesaan. Pada masa awalnya, 1970 – 1972, nama lengkap PT. BPR Madya Shinta Daya. Nama ini mengandung pengertian , bahwa misi pokok adalah membantu pemerintah meningkatkan taraf hidup social ekonomi pedesaan, melalui jasa perbankan. Pada masa itu, masih banyak masyarakat lebih mengenal jasa rentenir daripada jasa perbankan.

Pada saat awal beroperasi, para pendiri lebih memilih bentuk usaha perseroan terbatas (PT) agar dapat memberikan pelayanan secara lebih mandiri. Dan juga ciri positif dalam bentuk perseroan terbatas menjadi lebih bermakna, setelah dilengkapi nama diri yang dipilih, yakni: Shinta. Berdasarkan falsafah pewayangan, Dewi Shinta adalah tokoh yang setia-jujur-percaya diri. Dari ketiga karakter tersebut, diharapkan dapat menjadi kekuatan (daya) bagi prinsip prinsip usaha Shinta Daya dalam menjalin kerjasama dengan seluruh lapisan masyarakat.

Seiring dengan program pemerintah pada waktu itu, yakni dalam rangka menggalakkan 'kewiraswastaan' maka daerah kecamatan Kalasan-Prambanan semakin semarak dengan berbagai jenis usaha. Usaha pertanian semakin luas atas dukungan kesuburan lahan, program bimas dan teknologi. Sedangkan lalu-lintas

perdagangan semakin padat, karena lokasi Kalasan-Prambanan dan antara lintas perdagangan Yogyakarta-Klaten/Boyolali/Solo - Sleman – Wonosari.

Bahkan usaha kerajinan dan cinderamata juga semakin meningkat, mengingat adanya obyek wisata internasional Candi Prambanan, Ratu Boko dan lain-lainnya.

Namun sudah barang tentu usaha-usaha tersebut akan sulit tumbuh dan berkembang bila tidak didukung oleh faktor 'pemodalannya'. Apalagi para wiraswastawan tersebut kebanyakan terdiri dari para pengusaha golongan ekonomi lemah. Melihat peluang itu, memacu gagasan para pendiri Bank Shinta Daya, yang terdiri dari, Bapak Otho Soebijanto Atmosoedarmo dan Bapak AL. Teguh Soebardjar, SH. Kemudian pacuan ini melahirkan PT. BPR Desa Shinta Daya pada tanggal 5 Juni 1970 dengan akta notaries Nomor 6, jasa perbankan telah lahir di Kalasan-Prambanan, melangkah bersama tumbuhnya era kewiraswastaan.

PT. BPR Shinta Daya berkeyakinan bahwa, suka atau duka dalam operasional bukan hanya tergantung dari faktor intern juga faktor ekstern. Faktor ekstern sangat besar pengaruhnya, sesungguhnya terletak pada: Potensi Lingkungan yang meliputi faktor kependudukan, ekonomi, teknologi, budaya maupun sarana dan prasarana lain bagi kehidupan masyarakat setempat. Menyadari bahwa potensi lingkungan adalah potensi Bank Shinta Daya maka potensi lingkungan tersebut harus benar-benar kami pahami, kami hayati dan kami amalkan.

Sekedar gambaran masa lalu, pada awal operasinya tanggal 20 Agustus 1970, PT BPR Desa Shinta Daya hanya didukung 8 personel. Modal awalnya pun hanya Rp. 1.000.000 (satu juta rupiah). Kantor operasionalnya pun menempati dua buah kios yang dijadikan satu seluas 80 meter persegi. Bahkan, pada tutup buku 1971 masih mengalami kerugian.

Ternyata keprihatinan dan jerih payah tersebut tidak sia-sia. Terlebih didorong oleh karakteristik tokoh Dewi Shinta, yakni setia-jujur-percaya diri, maka saat ini terbukti bahwa situasi dan kondisi tersebut di atas telah jauh berlalu. Totalitas nasabah tersebut mencakup areal domisili di: daerah Yogyakarta - Gunungkidul - Bantul - Kulonprogo – Sleman – Klaten –dan sekitarnya.

3.2 Visi dan Misi Perusahaan

PT. BPR Shinta Daya sudah memasuki usia yang ke 35 tahun, memang penuh dengan suka duka dalam bergumul untuk memperjuangkan Ekonomi Kerakyatan. Sejak berdiri dan beroperasinya PT. BPR Shinta Daya tetap kokoh dan solid dikanca perbankan. Hal ini diperkuat dengan pengakuan sehat dengan klasifikasi A berdasarkan pada pengakuan Kantor Wilayah Bank Rakyat Indonesia-Yogyakarta, melalui surat keputusan nomor B. 1242. WAS/bpr/1986 tertanggal 17 April 1986 dan mendapatkan predikat sehat dari Bank Indonesia Cabang Yogyakarta dengan surat No. 29/15/ubpr/Adr/Yk/Rahasia.

Kondisi sehat tersebut, bukan hanya berlangsung untuk saat ini. Namun secara terus menerus, karena PT. BPR Shinta Daya mempunyai visi dan misi yang jelas dan bersifat membangun.

Adapun visi dari PT. BPR Shinta Daya adalah:

1. Meningkatkan taraf hidup / perekonomian masyarakat pada umumnya yang dihimpun dalam bentuk deposito dan tabungan yang dikembalikan dalam bentuk kredit.
2. PT. BPR Shinta Daya menjadi perusahaan yang kokoh dalam finansial dan kuat dalam kemitraan.
3. PT. BPR Shinta Daya memberikan kepercayaan dan kemudahan pelayanan terhadap masyarakat dan menjadikan wadah pengembangan diri serta kekerabatan yang memberikan efek bermanfaat bagi lingkungannya.

Sedangkan misi dari PT. BPR Shinta Daya adalah:

1. Membantu pemerintah dalam meningkatkan perekonomian masyarakat baik regional maupun nasional
2. Membangun jalinan hubungan kemitraan antara nasabah dan PT. BPR Shinta Daya yang lebih kokoh untuk menuju kesejahteraan bersama.
3. Memberikan kesejahteraan yang seimbang bagi para karyawan sesuai dengan kinerja dan sumbangsih pada perusahaan.
4. Menciptakan laba/keuntungan yang terus meningkat melalui kepuasan konsumen/customer akan produk perbankan yang ditawarkan dengan suku bunga yang menarik.

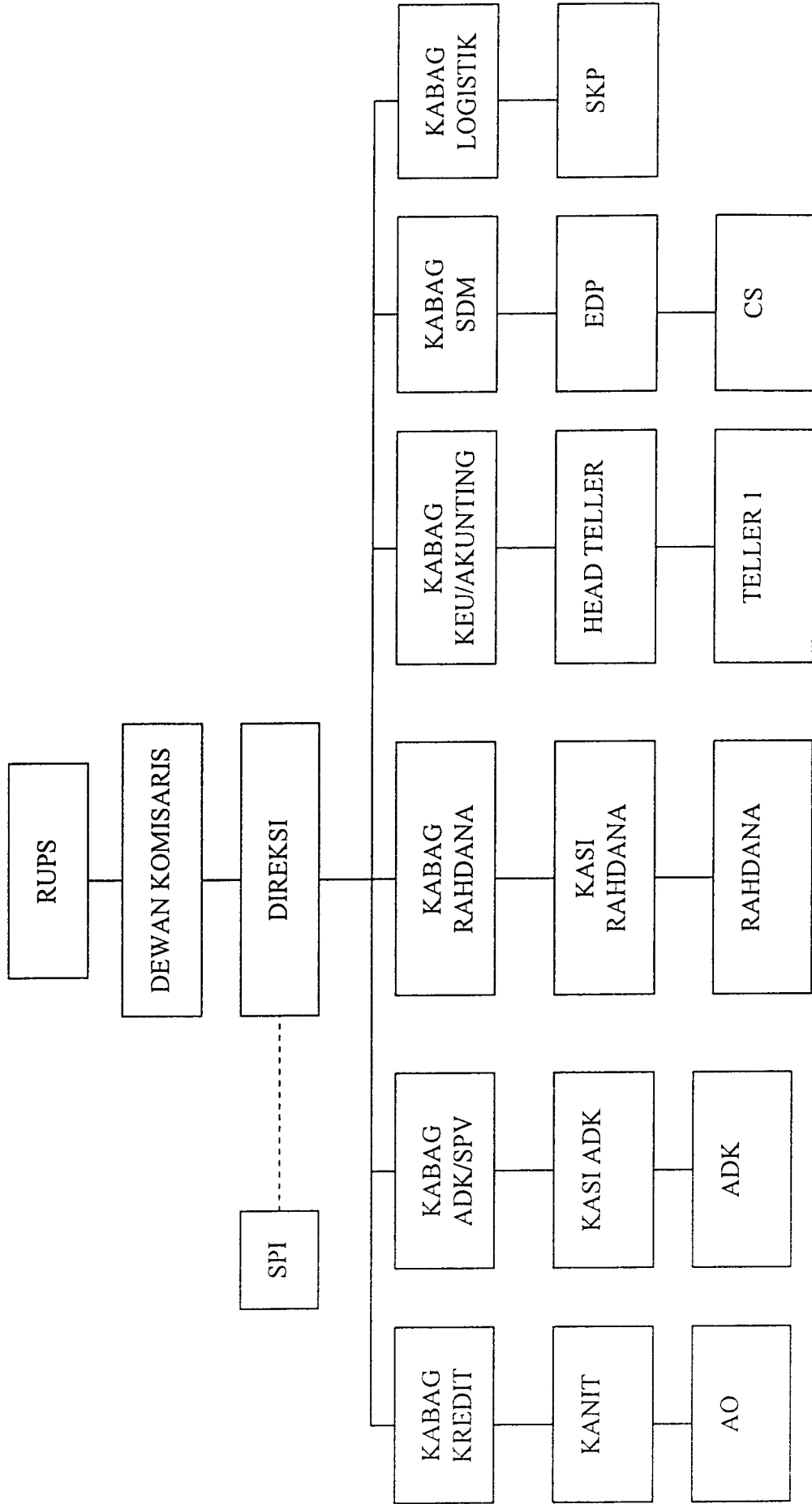
3.3 Struktur Organisasi

Struktur organisasi perusahaan memegang peranan penting, sebab tanpa adanya struktur organisasi, perusahaan tidak akan berjalan dengan baik. Struktur organisasi perusahaan Bank BPR Shinta Daya merupakan struktur organisasi lini staf. Dalam konsep struktur organisasi lini staf, semua jenis posisi atau bagian fungsional dapat dikategorikan ke dalam dua kelompok, yaitu kelompok lini dan kelompok staf. Kelompok lini yaitu para pejabat yang mengambil keputusan dan yang melaksanakan fungsi-fungsi manajemen.

Sedangkan kelompok staf yaitu bagian yang dalam menjalankan tugas operasionalnya dibantu oleh para bawahan. Adapun fungsi dari staf adalah membantu pimpinan dalam melaksanakan pekerjaan yang menjadi tanggung jawabnya, sehingga tersedia cukup waktu bagi pimpinan membuat perencanaan dan menentukan kebijakan dalam operasional perusahaan.

Sistem pertanggungjawaban di PT. BPR Shinta Daya sesuai dengan hirarki di perusahaan yang tercantum dalam struktur organisasi. Alur pertanggungjawabannya adalah dari bawah ke atas sementara alur pendelegasian tugas dan wewenang adalah dari atas ke bawah. Sejalan dengan itu, pengelola usaha distrukturkan ke dalam organisasi seperti tertera pada gambar3.2.

Struktur Organisasi PT. BPR Shinta Daya



3.3.1 Job Description

Berdasarkan struktur organisasi tersebut, maka hubungan antara bagian diperusahaan yang memiliki posisi sejajar bersifat koordinasi dan kolaborasi informasi. Masing-masing bagian menjalankan fungsinya sesuai dengan uraian kerja yang ditetapkan oleh perusahaan. Adapun fungsi-fungsi yang ada di dalam bagian tersebut adalah sebagai berikut:

1. RUPS

- a. Membahas arah dan tujuan perusahaan
- b. Menyetujui program-program perusahaan yang akan dijalankan
- c. Berhak mengangkat dan memberhentikan Direktur

2. Dewan Komisaris

- a. Bertugas mengawasi Direktur dalam Operasional Bank
- b. Berhak mengusulkan kepada rapat pleno untuk mengangkat dan memberhentikan Direktur
- c. Membuat laporan ke Bank Indonesia setiap 6 bulan sekali

3. Direktur

- a. Berkewajiban untuk menentukan kebijaksanaan perusahaan dan memimpin perusahaan sesuai dengan garis-garis kebijaksanaan yang telah ditetapkan oleh rapat pleno.
- b. Mengadakan pembagian tugas.
- c. Membimbing dan mengkoordinasikan semua bagian dalam melaksanakan operasi perusahaan.

- d. Memberikan laporan pertanggungjawaban kepada Bank Indonesia dan Dewan Direksi.

4. SPI

- a. Mengawasi operasional Bank yang dilaksanakan oleh Direksi dan karyawan.
- b. Membuat laporan tiap akhir bulan tentang kinerjanya.

5. Kepala Bagian Kredit

- a. Melakukan perencanaan dan membuat rekomendasi pada karyawan mengenai tugas.
- b. Melakukan pembinaan dan mengadakan pengawasan atas team kerja sehubungan dengan masalah transaksi di bidang perkreditan, baik hubungan dengan nasabah dan tanggungjawab terhadap kelancaran operasional maupun tentang moral karyawan yang dipimpinnya.

Dalam melaksanakan tugasnya, Kabag Kredit dibantu oleh dua bagian, yaitu:

- a. Kepala Unit (Kanit)
 - 1. Melaksanakan tugas dan rencana kerja masing-masing sesuai yang telah ditetapkan oleh atasnya.
 - 2. Mengawasi pelaksanaan yang dilakukan oleh bawahannya masing-masing.
- b. Account Officer
 - 1. Bertanggungjawab terhadap produk perbankan yang akan dikenal oleh nasabah.

2. Pelaksana dilapangan sebagai ujung tombak dalam mencari nasabah, menawarkan dan mengajak nasabah agar mau menabung dan meminjam di bank.

6. Kepala Bagian Keuangan

- a. Bertanggungjawab merencanakan dan mengendalikan keuangan perusahaan.
- b. Bertanggungjawab terhadap segala aktifitas keuangan dalam perusahaan.
- c. Bertanggungjawab menyusun laporan keuangan harian dan bulanan.
- d. Bertanggungjawab mengawasi dan membimbing teller.
- e. Mencatat mutasi karyawan.

Dalam menjalankan tugasnya, Kepala Bagian Keuangan dibantu oleh head teller yang bertugas:

- a. Bertanggungjawab atas uang yang keluar dan masuk dalam bank yang mempunyai wewenang diatas teller
- b. Mengarsip dokumen-dokumen yang berkaitan dengan kegiatan keluar-masuknya uang dalam bank
- c. Berkewajiban membuat laporan keuangan kepada Kepala Bagian Keuangan

7. Kepala Bagian Sumber Daya Manusia (SDM)

- a. Bertanggungjawab atas penerimaan pegawai
- b. Bertanggungjawab mengawasi kinerja karyawan
- c. Bertanggungjawab membuat daftar gaji karyawan
- d. Bertanggungjawab atas rotasi karyawan, kenaikan pangkat dan pendidikan

Dalam menjalankan tugasnya, Kepala Bagian SDM dibantu oleh:

a. Elektronik Data Processing

1. Bertanggungjawab mengolah data input/output karyawan
2. Bertanggungjawab mengarsipkan data-data karyawan

b. Customer Service

1. Bertanggungjawab atas segala hal yang berkaitan dengan administrasi kantor
2. Bertanggungjawab menerima tamu dan menghubungi contact personnya
3. Bertanggungjawab mendistribusikan arus telepon baik keluar maupun masuk (sebagai operatornya)

8. Kepala Bagian Logistik

- a. Bertanggungjawab menangani setiap pembelanjaan dan pembayaran segala fasilitas kantor
- b. Bertanggungjawab mengenai setiap keluhan yang berhubungan dengan fasilitas kantor
- c. Bertanggungjawab memelihara kebersihan dan perawatan bagi operasional

Dalam menjalankan tugasnya, Kepala Bagian Logistik dibantu oleh Sekretariat Keamanan dan Pengamanan (SKP) :

1. Sekretaris:

- a. Berkewajiban untuk membantu pekerjaan Direktur
- b. Mencatat dan menyiapkan apa yang diperlukan Direktur
- c. Mengarsip surat-menyurat untuk Direksi

2. Keamanan dan Pengamanan

- a. Bertanggungjawab menjaga dan mengawasi keamanan kantor
- b. Menjaga kebersihan, kerapian dan keamanan kantor
- c. Bertanggungjawab mengawasi arus telepon (masuk dan keluar) pada malam hari.

3.4 Personalia

Setiap perusahaan mempunyai tata personalia yang berbeda tetapi pada umumnya mempunyai tujuan yang sama. Adapun tujuan diadakannya personalia adalah untuk memperoleh dan mengelola sumber daya manusia agar tujuan tiap individu karyawan sejalan dengan tujuan perusahaan. Untuk mendapatkan partisipasi dan produktivitas karyawan, maka PT. BPR Shinta Daya mengambil kebijaksanaan mengenai ketenagakerjaan yang meliputi:

1. Waktu/jam kerja
2. Gaji karyawan
3. Tunjangan hari raya
4. Cuti karyawan
5. Asuransi
6. dan lain sebagainya.

a. Jumlah Karyawan

Dalam usahanya untuk mencapai tujuan perusahaan, maka karyawan merupakan salah satu faktor yang penting, karena karyawan merupakan motor penggerak dalam kegiatan yang menunjang tercapainya tujuan.

Jumlah karyawan di perusahaan PT BPR Shinta Daya untuk saat ini berjumlah 69 orang, yang semuanya sudah diangkat menjadi pegawai tetap.

b. Upah Karyawan

Perusahaan PT. BPR Shita daya di dalam memberikan upah karyawan berdasarkan sifat pekerjaannya. Perusahaan dalam hal ini mengadakan penggolongan upah kepada masing-masing karyawan. Besarnya upah dibayarkan berdasarkan upah minimum pemerintah. Upah yang diberikan atas dasar bulanan dari Rp.900.000,- sampai dengan Rp. 15.000.000,-

c. Kesejahteraan Karyawan

Perusahaan PT. BPR Shinta Daya selain memberikan upah kepada karyawannya juga memberikan perhatian dalam hal kesejahteraan karyawannya. Hal ini dimaksudkan agar para karyawan tersebut dengan tingkat kesejahteraan yang memadai diharapkan akan memberikan sumbangsih yang besar kepada perusahaan.

Adapun macam-macam kesejahteraan yang diberikan oleh PT. BPR Shinta Daya kepada para karyawannya adalah sebagai berikut:

1. Tunjangan hari raya
2. Tunjangan keluarga
3. Tunjangan pensiun
4. Tunjangan kemahalan
5. Pinjaman tiap bulan

d. Jam Kerja

Karyawan PT. BPR Shinta Daya bekerja selama 8 jam pada hari senin sampai dengan kamis, yaitu pukul 07.30 sampai pukul 16.05, untuk hari jumat bekerja mulai pukul 06.30 sampai dengan pukul 14.00 dan khusus hari sabtu bekerja selama 6 jam, yaitu dari pukul 07.30 sampai dengan 12.00 WIB, yang termasuk juga waktu istirahat antara pukul 12.00 sampai pukul 13.00 WIB. Selama satu minggu karyawan bekerja selama 6 hari.

3.5 Bidang Usaha

PT. BPR Shinta Daya, adalah salah satu Bank Perkreditan Rakyat yang ditunjuk oleh Bank Indonesia, sebagai pelaksana Pilot Proyek hubungan Bank dengan Kelompok Swadaya Masyarakat (PPHBKSM). Penunjukan tersebut didasarkan pertimbangan bahwa PT. BPR Shinta Daya memiliki kondisi cukup sehat dan memiliki kemampuan pelayanan cukup baik.

Untuk sementara itu telah terjalin kerjasama dengan beberapa Lembaga Pengembangan Swadaya Masyarakat (LPSM) yang antara lain: Yayasan Sosial Tunggal Jati, Yogyakarta – Yayasan Indonesia Sejahtera, Surakarta – Bina Sadaya, Surakarta. Tujuan utama jalinan kerjasama ini ialah membina kelompok-kelompok usaha anggota LPSM yang antara lain melalui bantuan kredit usaha. Misalnya bagi usaha pertanian - peternakan - industri kecil – dan lain sebagainya.

PT. BPR Shinta Daya memasuki usianya yang ke 30 tahun (Tiga Dasa Warsa), telah menyalurkan kredit berupa:

1. Kredit pegawai
2. Kredit Umum
3. Kredit KSM
4. Kredit Pasar
5. Kredit PKM

Sedangkan pada sisi nasabah deposan atau Penabung, meliputi:

1. Deposito
2. Tabungan Umum
3. Tabungan Pegawai
4. Dan Tabungan Taberna

3.6 Sistem Penggajian dan Pengupahan

Di dalam sistem penggajian dan pengupahan ini bagian-bagian yang terkait adalah sebagai berikut:

1. Bagian SDM

Bertugas membuat Rekap Daftar Hadir Karyawan, membuat Laporan Kehadiran Karyawan Tahunan yang akan dilaporkan kepada Direksi. Setelah menerima Laporan Penilaian Kerja Bulanan dan menginput gaji karyawan setelah itu membuat Daftar Gaji Karyawan.

2. Bagian Accounting

Menginput ulang Daftar Gaji Karyawan yang diterima dari bagian SDM kemudian membuat Daftar Gaji Per Karyawan yang digunakan sebagai dasar membuat slip kas bank sebagai permintaan pengeluaran uang gaji.

3. Bagian keuangan

Bertugas membuat cek berdasarkan slip bank yang diterima dari bagian accounting dan mengambil uang di bank.

Sedangkan dokumen yang digunakan dalam sistem penggajian dan pengupahan adalah sebagai berikut:

1. Absen Karyawan/ Daftar Hadir

Digunakan sebagai dasar pembuatan daftar gaji karyawan karena di dalam absen karyawan tercantum jam kehadiran dan jam pulang karyawan.

2. Laporan Kehadiran Karyawan Bulanan

Digunakan sebagai dasar membuat daftar gaji yang akan diterima karyawan dalam suatu periode tertentu.

3. Laporan Penilaian Kerja Karyawan Tahunan

Laporan ini akan dikirimkan ke direksi sebagai catatan terhadap prestasi kerja karyawan.

4. Laporan Daftar Gaji Karyawan

Digunakan sebagai dasar otorisasi atas slip kas bank untuk perintah pengeluaran cek sesuai dengan jumlah yang tercantum dalam daftar gaji karyawan.

5. Slip Kas Bank

Digunakan sebagai dasar perintah pengeluaran cek atas sejumlah uang untuk pembayaran gaji karyawan.

6. Kartu Gaji Karyawan

Berisi jumlah gaji yang akan diterima karyawan dan digunakan sebagai bukti penerimaan amplop gaji yang telah diterima oleh karyawan sehingga tidak terjadi kecurangan didalam pengambilan gaji.

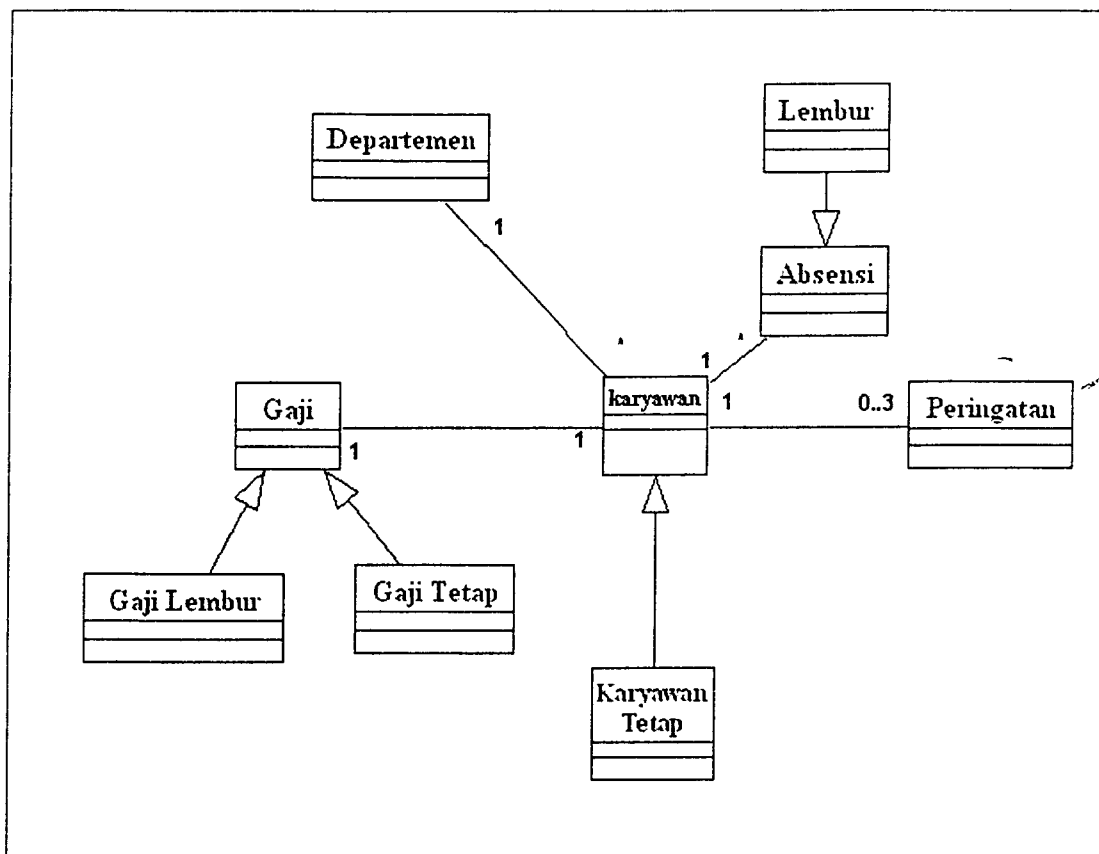
BAB IV ANALISA DAN PEMBAHASAN

Pada bab sebelumnya telah diuraikan pengertian sistem informasi dan prosedur penggajian, pendekatan berorientasi objek, UML, dan gambaran umum perusahaan. Dalam bab ini akan disajikan mengenai pembahasan dan analisa sistem penggajian dengan menggunakan UML pada BPR Shinta Daya. Analisa dilakukan setelah melihat hasil wawancara dan observasi dengan pihak perusahaan dan mempelajari bukti-bukti dokumen yang ada di perusahaan tersebut.

Pembahasan dan analisa sistem dengan berbasis komputer pada BPR Shinta Daya meliputi perancangan sistem penggajian karyawan diterapkan pada komputersasi yang berorientasi obyek berbasis UML.

4.1 Diagram Class

Diagram *Class* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem. Dalam tahap desain, *class diagram* berperan menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Diagram *Class* dalam sistem penggajian disajikan dalam Gambar 4.1



Gambar 4.1 Diagram *Class* untuk Sistem Penggajian Karyawan

Diagram *Class* juga memperlihatkan Atribut dan perilaku atau operasi pada setiap kelas yang ada. Adapun atribut dalam sistem penggajian disajikan dalam tabel 4.1.

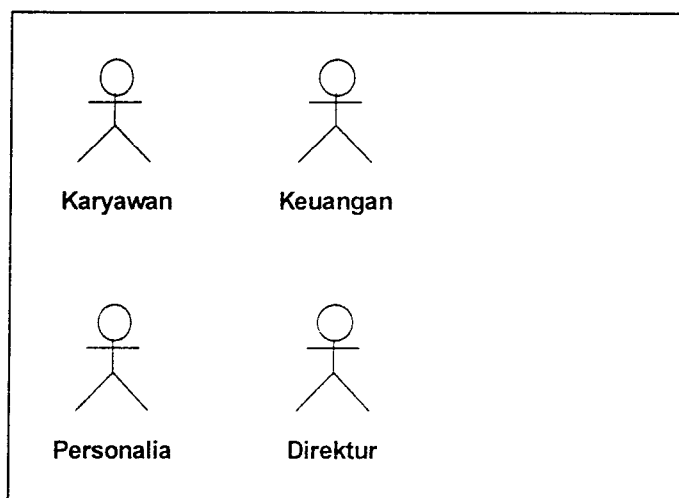
Tabel 4.1 Atribut diagram *Class* sistem penggajian karyawan

No	Nama Kelas	Atribut	Operasi	Keterangan
1.	Karyawan	NoKartu, NIK, Nama, Jabatan, TempatLahir, TglLahir, JenisKelamin, StatusKawin, Alamat, Kota, Propinsi, Telepon, Pendidikan, TglMasuk	Tambah, Edit, Hapus	SuperClass
2.	KaryawanTetap	Tgl Pensiun	TambahTgl Pensiun	SubClass

3.	Gaji	Gaji pokok, Potongan, Tunjangan	Tambah, Edit, Hapus	SuperClass
4.	Departemen	Nama Departemen	Tambah, Edit, Hapus	-
5.	Absensi	Tanggal, JamKerja, JamMasuk, JamPulang, KetAbsensi	Tambah, Edit, Hapus, TotalMasukKerja	SuperClass

4.2 Use Case Diagram (UCD)

UCD menjelaskan fungsionalitas suatu sistem dan bagaimana sistem berinteraksi dengan dunia luar. Selama proses analisis, UCD menangkap persyaratan-persyaratan sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Sedangkan pada proses desain, UCD menetapkan perilaku (behavior) sistem saat diimplementasikan. Dari sistem penggajian karyawan di atas, maka dapat diidentifikasi aktor yang terlibat dalam sistem, yaitu Karyawan, Personalia, Keuangan, dan Direktur. Gambar 4.2 berikut ini merupakan gambar dari aktor-aktor yang terlibat dalam Sistem.

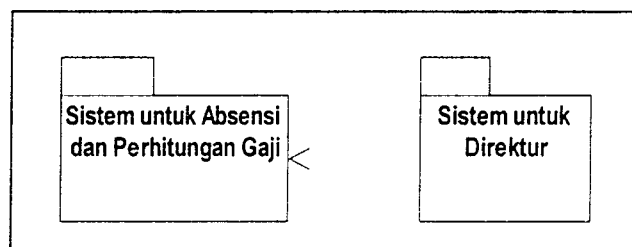


Gambar 4.2. Gambar Aktor dalam Sistem

Dari aktor tersebut di atas, *Use Case* yang dapat diidentifikasi adalah sebagai berikut :

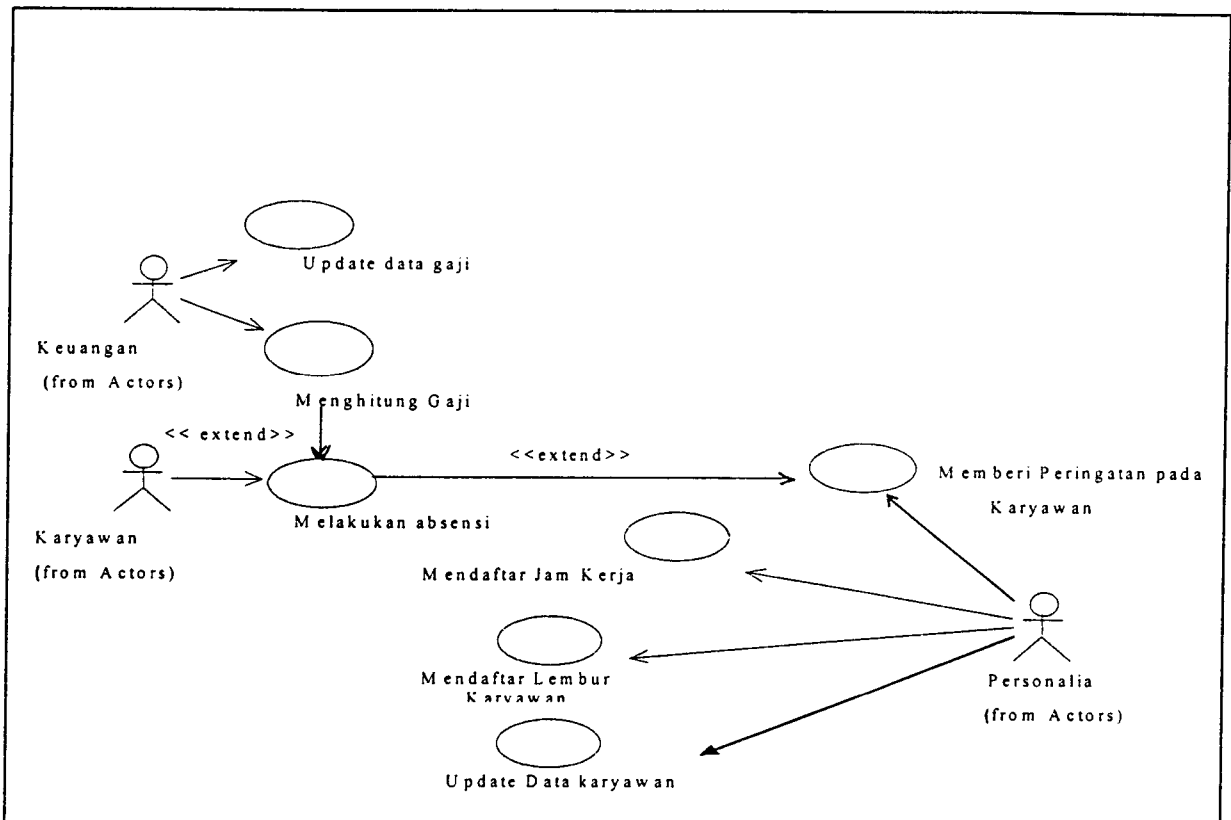
- a. Melakukan Absensi
- b. *Update* Data Gaji
- c. Menghitung Gaji
- d. Memberi Peringatan Pada Karyawan
- e. *Update* Data Karyawan dan Absensi
- f. Mendaftar Jam Kerja
- g. Mendaftar Lembur Karyawan
- h. Meminta Informasi Karyawan
- i. Meminta Informasi Gaji Karyawan

Pada dasarnya *Use Case-Use Case* tersebut di atas terbagi dalam dua sistem. Pertama yaitu *Use Case* sistem untuk absensi dan perhitungan gaji, dan yang kedua adalah *Use Case* sistem untuk Direktur dimana Direktur tidak terlibat secara langsung dalam sistem penggajian karyawan. Gambar 4.3 menggambarkan kedua paket *Use Case* tersebut di atas dimana sistem untuk Direktur tergantung kepada sistem untuk absensi dan perhitungan gaji.



Gambar 4.3 Paket Diagram *Use Case*

Untuk Gambar 4.4 berikut ini menunjukkan Diagram *Use Case* sistem untuk absensi dan perhitungan gaji dimana pada Diagram *Use Case* tersebut terlihat beberapa hubungan asosiasi antar *Use Case* yang satu dengan *Use Case* yang lainnya dengan *Stereotype extend*”.

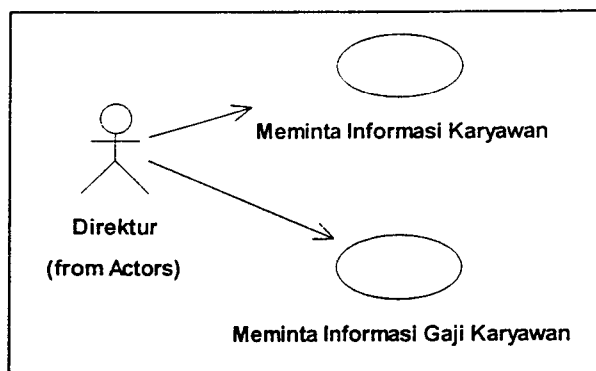


Gambar 4.4 Diagram *Use Case* untuk Sistem Absensi dan Perhitungan Gaji

Hubungan asosiasi dengan *Stereotype* “extend” yang terjadi adalah antara *Use Case* Memberi Peringatan pada Karyawan dan *Use Case* Melakukan Absensi menunjukkan bahwa *Use Case* Memberi Peringatan pada Karyawan merupakan kelanjutan dari *Use Case* Melakukan Absensi apabila ditemukan karyawan yang melanggar peraturan.

Antara *Use Case* Menghitung Gaji dan *Use Case* Melakukan Absensi menunjukkan bahwa *Use Case* Menghitung Gaji merupakan kelanjutan dari *Use Case* Melakukan Absensi jika Keuangan melakukan perhitungan besar gaji karyawan.

Gambar 4.5 berikut ini menunjukkan Diagram *Use Case* sistem untuk Direktur dimana Direktur dapat melakukan permintaan atau mengakses informasi baik informasi karyawan maupun gaji karyawannya.



Gambar 4.5. Diagram *Use Case* Sistem Untuk Direktur

Skenario dari *Use Case-Use Case* tersebut disajikan dalam tabel 4.2 :

Tabel 4.2 Skenario *Use Case*

1.	Nama	:	Melakukan Absensi
	Aktor	:	Karyawan
	Deskripsi	:	<ul style="list-style-type: none"> • Use Case ini dimulai ketika karyawan melakukan absensi pada saat masuk dan pulang kerja dan berakhir setelah karyawan selesai melakukan absensi

2.	Nama : Aktor : Deskripsi :	Update Data Gaji Keuangan <ul style="list-style-type: none"> • Use Case ini dimulai ketika Keuangan melakukan perubahan-perubahan terhadap gaji karyawan dan akan berakhir setelah selesai melakukan perubahan-perubahan tersebut. Perubahan-perubahan yang dilakukan adalah pada besar gaji pokok dan item-item dari potongan dan tunjangan gaji.
3.	Nama : Aktor : Deskripsi :	Hitung Gaji Keuangan <ul style="list-style-type: none"> • Use Case ini dimulai ketika bagian Keuangan melakukan perhitungan gaji dan berakhir setelah selesai melakukan perhitungan. • Use Case ini setiap hari dapat menghitung besar gaji karyawan perusahaan baik gaji tetap maupun tidak tetap. Informasi besar gaji dapat dicetak untuk tanda terima (slip gaji) karyawan tetap. Informasi besar gaji juga dapat dicetak untuk keperluan laporan yang berisi informasi tentang total gaji baik untuk hari kerja biasa maupun kerja lembur.

4.	Nama : Aktor : Deskripsi :	Memberi Peringatan pada Karyawan Personalia <ul style="list-style-type: none"> ● Use Case ini dimulai ketika Use Case Mengevaluasi Karyawan menemukan karyawan yang melanggar peraturan perusahaan. Use Case akan berakhir setelah bagian Personalia membuat surat peringatan. ● Use Case ini untuk menampilkan karyawan yang tidak masuk kerja baik karena sakit, ijin maupun tanpa ijin. ● Use Case akan menyimpan informasi mengenai surat peringatan yang diterima oleh seorang karyawan.
5.	Nama : Aktor : Deskripsi :	Update Data Karyawan dan Absensi Personalia <ul style="list-style-type: none"> ● Use Case ini dimulai ketika terjadi penambahan karyawan maupun perubahan-perubahan baik pada data karyawan, data status hari, dan data status absensi. Use Case berakhir setelah penambahan atau perubahan selesai dilakukan. ● Untuk menambah dan merubah data karyawan perusahaan. ● Merubah status hari dan status absen. ● Mencetak laporan data karyawan dan data absensi.
6.	Nama : Aktor : Deskripsi :	Mendaftar Jam Kerja Personalia <ul style="list-style-type: none"> ● Use Case ini dimulai setiap ada perubahan jam kerja untuk karyawan tetap dan berakhir setelah bagian Personalia selesai memasukkan data tersebut. ● Memasukkan data jam kerja.

7.	Nama Aktor Deskripsi	: : :	Mendaftar Lembur Karyawan Personalia <ul style="list-style-type: none"> • Use Case ini dimulai setiap dilakukan kerja lembur yang dilakukan oleh setiap departemen dan berakhir setelah Personalia selesai memasukkan data lembur. • Mengetahui karyawan-karyawan yang melakukan kerja lembur yang diajukan oleh setiap departemen untuk mencegah karyawan yang tidak kerja lembur masuk kerja.
8.	Nama Aktor Deskripsi	: : :	Meminta Informasi Karyawan Direktur <ul style="list-style-type: none"> • Use Case ini dimulai saat Direktur melihat informasi karyawan dan berakhir setelah Direktur selesai melihat informasinya. • Mengetahui informasi mengenai karyawan perusahaan.
9.	Nama Aktor Deskripsi	: : :	Meminta Informasi Gaji Direktur <ul style="list-style-type: none"> • Use Case ini dimulai saat Direktur melihat informasi gaji yang dikeluarkan perusahaan dan berakhir setelah Direktur selesai melihat informasinya. • Mengetahui semua informasi mengenai gaji yang dibayar oleh perusahaan kepada karyawannya.

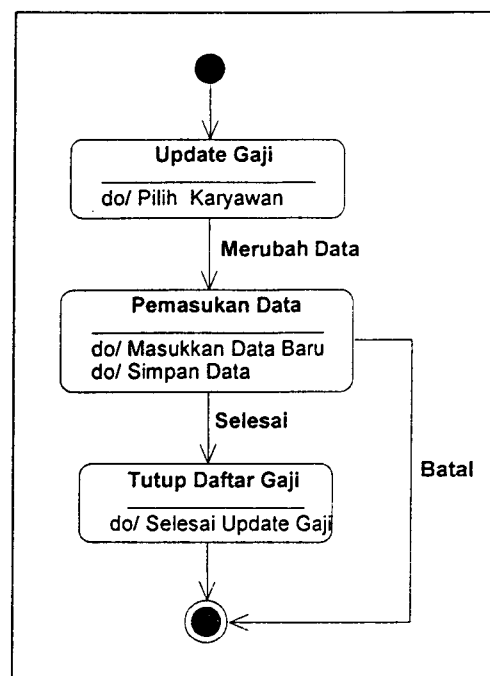
4.3 Diagram *Statechart*

Dalam pembuatan Diagram *Statechart* ini tidak semua objek kelas dibuat *Statechart Diagram*-nya. Hanya objek kelas yang penting (sangat diperlukan sistem) dan mempunyai perilaku yang sangat dinamis saja yang dibuat *Statechart Diagram*-nya. Pada kasus ini Diagram *Statechart* yang dibuat adalah :

- a. Kelas departemen yaitu untuk objek keuangan (*Update* data gaji dan menghitung gaji).
- b. Objek kelas absensi.

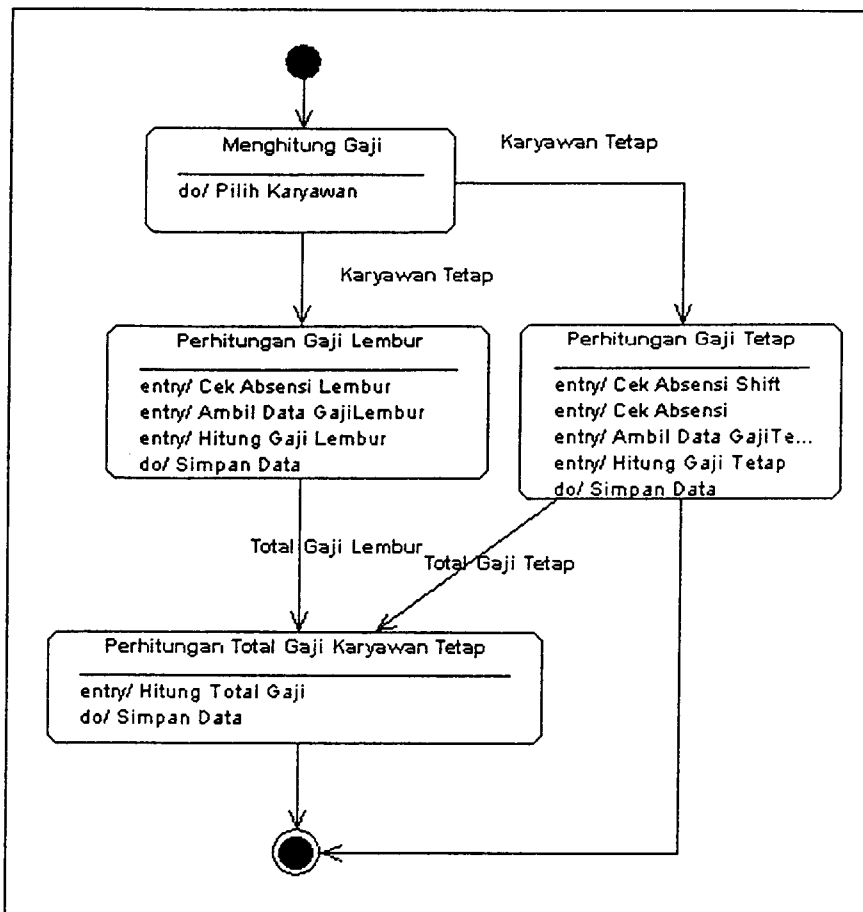
4.3.1 Diagram *Statechart* Kelas Departemen

Untuk objek keuangan mempunyai dua Diagram *Statechart*, yaitu Diagram *Statechart* untuk meng-*Update* gaji dan Diagram *Statechart* untuk mencetak laporan mengenai gaji karyawan sebagaimana disajikan dalam Gambar 4.6 dan 4.7. Gambar 4.6 merupakan Diagram *Statechart* objek Keuangan untuk meng-*Update* gaji dan Gambar 4.7 merupakan Diagram *Statechart* untuk mencetak laporan.



Gambar 4.6. Diagram *Statechart* Objek Keuangan Untuk Update Gaji

Pada Gambar 4.6 di atas objek keuangan melakukan *Update* gaji karyawan dengan memilih kelompok karyawan mana yang akan di-*Update*. Kemudian melakukan perubahan-perubahan terhadap item-item gaji seperti gaji pokok, tunjangan, dan potongan.



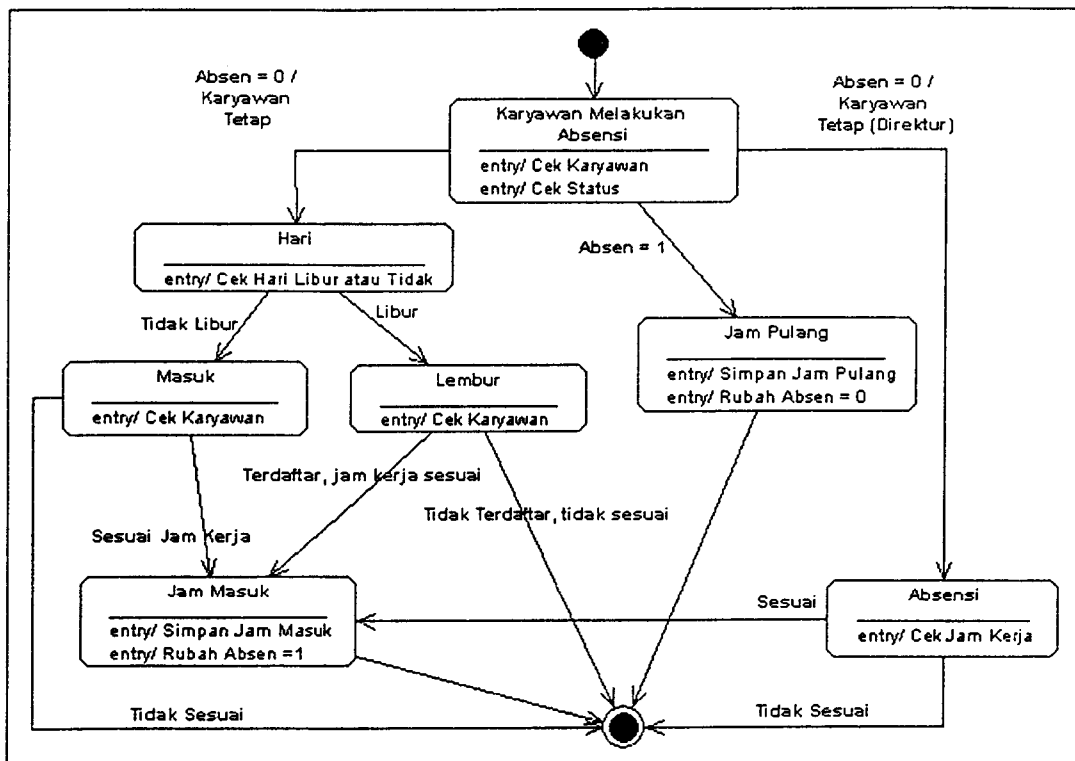
Gambar 4.7. Diagram *Statechart* Objek Keuangan Untuk Menghitung Gaji

Gambar 4.7 diatas merupakan Diagram *Statechart* objek keuangan untuk menghitung gaji karyawan. Diagram *Statechart* ini dimulai dengan keadaan Menghitung Gaji dimana pada keadaan ini dilakukan pemilihan karyawan mana yang akan dihitung gajinya.

4.3.2 Diagram *Statechart* Objek Kelas Absensi

Diagram *State* untuk objek kelas Absensi ini dimulai dengan keadaan karyawan melakukan absensi. Pada keadaan ini dilakukan pengecekan pada karyawan dan absensi. Apabila absen bernilai 0 maka akan dilakukan pengecekan hari. Bila

masuk sesuai dengan jam kerjanya maka jam masuk karyawan akan disimpan dan nilai absen dirubah menjadi 1. Bila karyawan masuk tidak sesuai dengan jam kerjanya maka akan langsung selesai. Apabila absen bernilai 1 pada saat karyawan melakukan absensi, maka secara otomatis data jam dianggap sebagai jam pulang karyawan.



Gambar 4.8. Diagram *Statechart* Objek Kelas Absensi

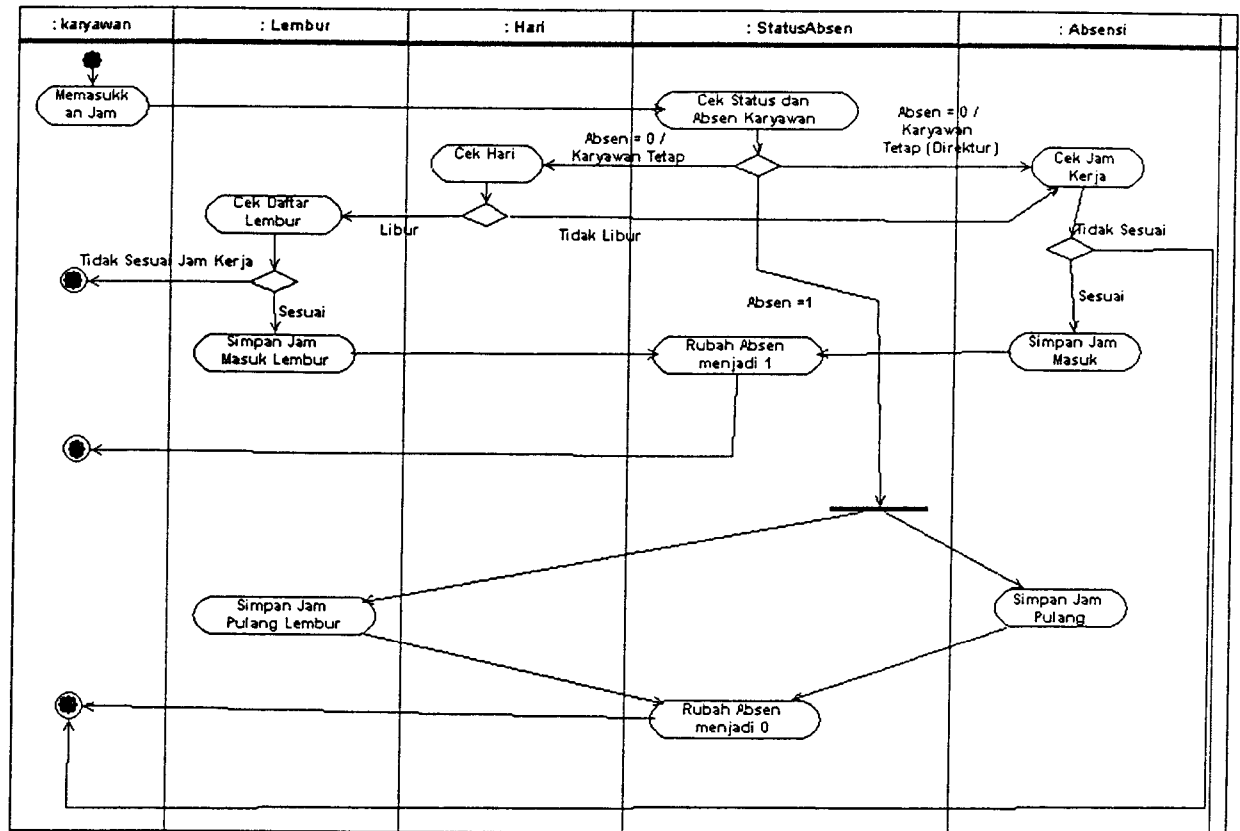
4.4 Diagram Activity

Activity diagram memodelkan alur kerja sebuah proses bisnis dan urutan aktivitas dalam suatu proses, yang berguna untuk menggambarkan bagaimana perilaku dalam berbagai *use case* berinteraksi. Activity diagram yang dibuat meliputi:

- Diagram *Activity Use Case* Melakukan Absensi.
- Diagram *Activity Use Case* Menghitung Gaji

4.4.1 Diagram *Activity Use Case* Melakukan Absensi

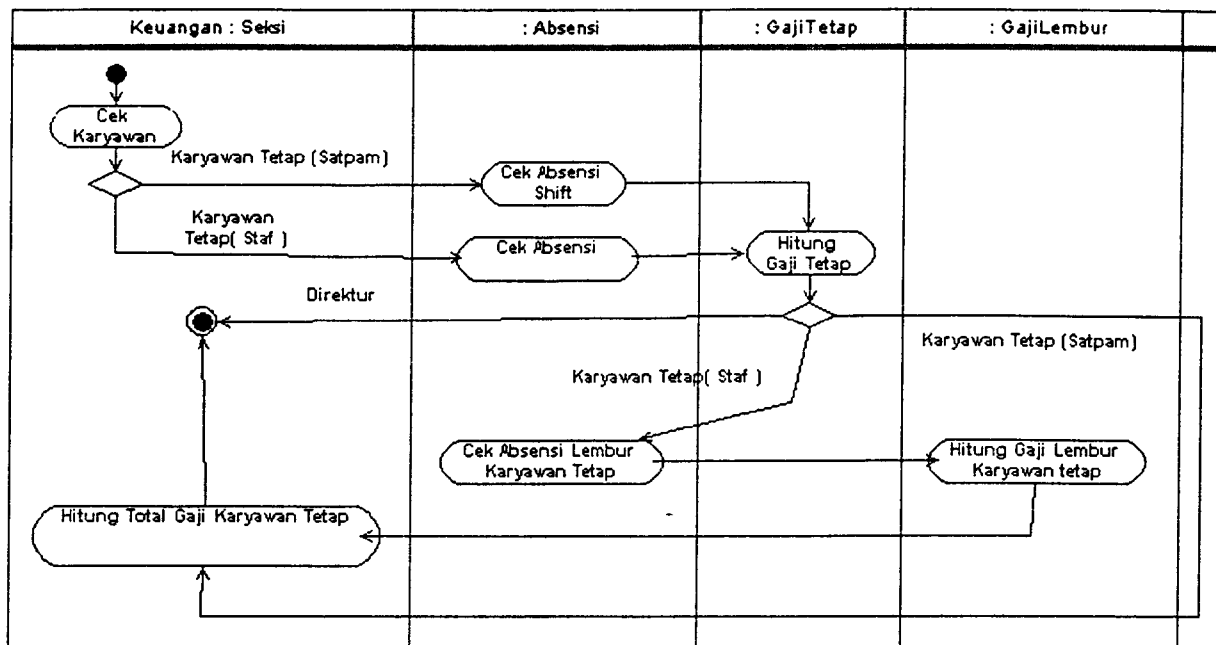
Diagram *Activity* untuk *Use Case* Melakukan Absensi di atas dimulai dengan aktifitas memasukkan jam yang dilakukan oleh karyawan dengan cara menggesekkan kartu absensinya pada alat perekam. Kemudian akan dilakukan aktifitas pengecekan karyawan dan absen karyawan pada kelas Absen. Apabila absen bernilai 0, maka dilakukan pengecekan hari. Untuk hari libur maka dilakukan pengecekan lagi pada daftar lembur karyawan. Proses yang sama juga dilakukan untuk hari biasa, akan tetapi pengecekan dilakukan pada jam masuk karyawan untuk mengetahui jam masuk kerja karyawan sudah sesuai atau tidak. Apabila terdaftar maka jam masuk karyawan akan disimpan dan absen dirubah menjadi 1. Bila masuk kerja sesuai dengan jam kerjanya maka data jam masuk disimpan dan absen dirubah menjadi 1. Pada waktu dilakukan pengecekan absensi karyawan ditemukan absen bernilai 1, maka data akan dimasukkan sebagai jam pulang karyawan dan absen dirubah kembali menjadi 0.



Gambar 4.9 Diagram *Activity Use Case* Melakukan Absensi

4.4.2 Diagram *Activity Use Case* Menghitung Gaji

Gambar 4.10 merupakan Diagram *Activity Duntuk Use Case* Menghitung Gaji. Diagram *Activity* tersebut dimulai dengan aktifitas mengecek karyawan yang dihitung besar gaji yang diperolehnya. Bila karyawan berstatus sebagai karyawan tetap, maka akan dilakukan aktifitas menghitung gaji tetap. Gambar 4.10. Diagram *Activity Use Case* Menghitung Gaji



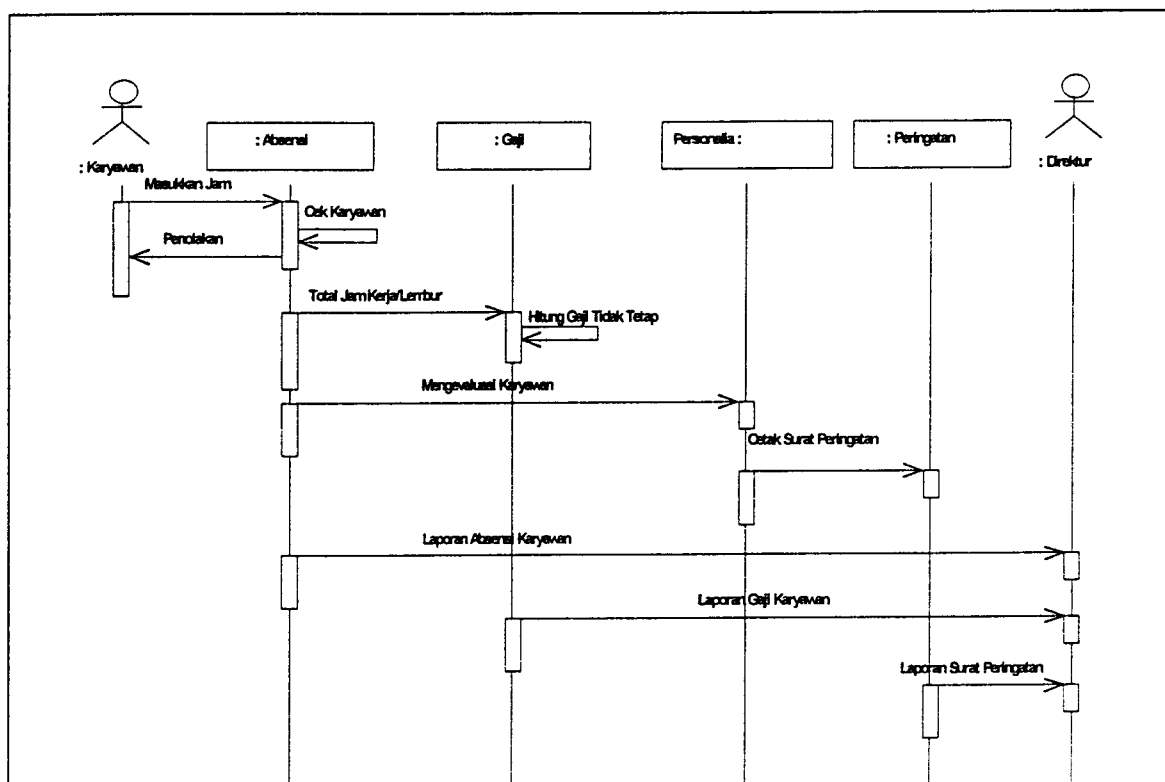
Gambar 4.10. Diagram *Activity Use Case* Menghitung Gaji

Untuk Karyawan Tetap perhitungan gaji yang diterimanya sudah selesai. Untuk menghitung gaji lemburnya maka harus dilakukan dahulu aktifitas untuk mengecek daftar absensi lemburnya. Setelah diketahui jumlah jam lemburnya, maka dilakukan aktifitas menghitung gaji lembur. Setelah diketahui jumlah total jam kerja dan jam lemburnya maka dilakukan aktifitas perhitungan gaji. Kemudian kedua gaji tersebut dijumlahkan pada aktifitas hitung total gaji karyawan untuk memperoleh besar gaji yang diterimanya.

4.5 Diagram *Sequence*

Diagram *Sequence* pada Gambar 4.11 berikut ini merupakan Diagram *Sequence* dari keseluruhan sistem perhitungan gaji karyawan yang menggambarkan urutan kejadian berdasarkan waktu. Pada diagram tersebut terlihat bahwa diagram dimulai dengan objek karyawan yang melakukan absensi (memasukkan jam),

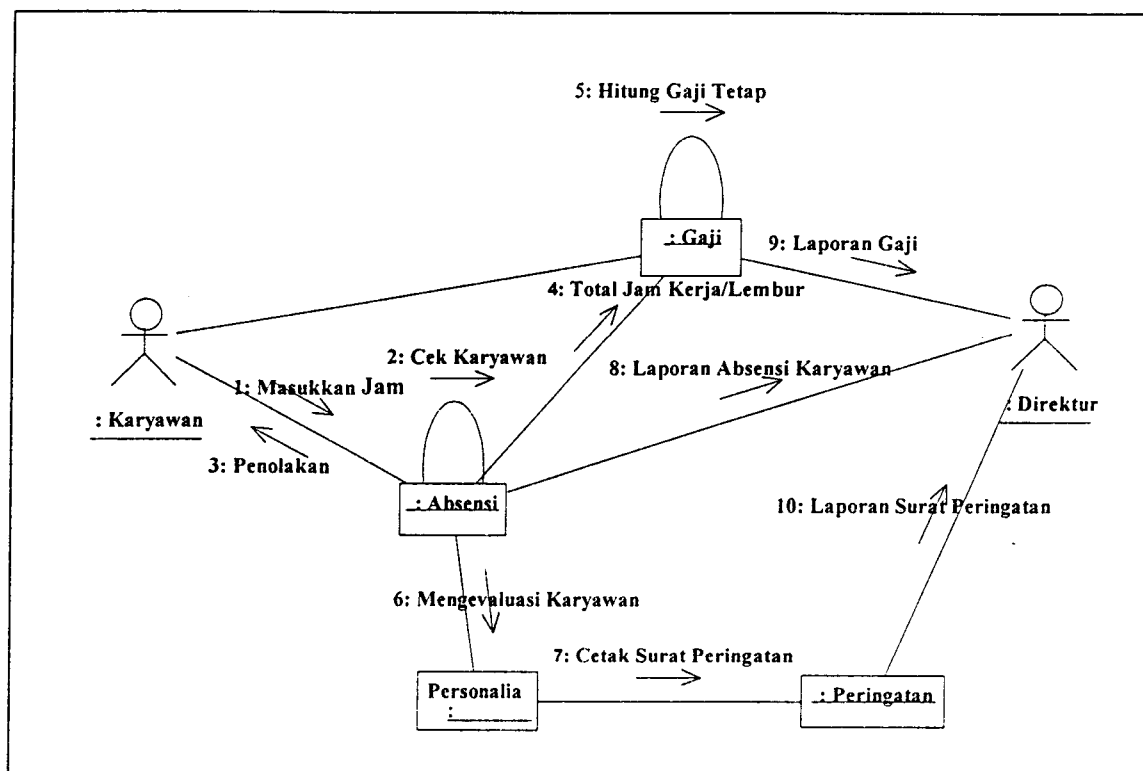
kemudian dilakukan pengecekan pada karyawan apakah masuk sesuai dengan jam kerjanya atau tidak. Selanjutnya dilakukan perhitungan gaji karyawan dengan menghitung jumlah jam kerja atau lembur karyawan dan mengalikannya dengan besar gaji atau upah lembur per jamnya. Absensi yang diperoleh juga dapat digunakan untuk mengevaluasi karyawan oleh Personalia dan kemudian Personalia akan menerbitkan surat peringatan jika ditemukan karyawan yang melanggar. Absensi, besar gaji, dan surat peringatan yang telah diperoleh kemudian dilaporkan ke Direktur untuk keperluan manajemen perusahaan.



Gambar 4.11 Diagram Sequence Sistem Penggajian Karyawan

4.6 Diagram Collaborasi

Diagram *Collaboration* sistem perhitungan gaji karyawan berikut ini merupakan gambaran mengenai urutan kejadian di dalam sistem dari satu objek ke objek yang lainnya. Diagram *Collaboration* berhubungan secara langsung dengan Diagram *Sequence* sehingga kedua diagram ini tidak dapat dipisahkan satu dengan yang lainnya. Dengan demikian jika dilakukan perubahan pada Diagram *Sequence* maka perubahan juga harus dilakukan pada Diagram *Collaboration*, demikian juga sebaliknya.



Gambar 4.12 Diagram Collaboration Sistem Penggajian Karyawan

BAB V KESIMPULAN DAN SARAN

Dalam bab ini akan diuraikan kesimpulan tentang penyusunan sistem penggajian dengan metode UML, sedangkan saran akan disajikan pada bagian 5.2.

5.1 Kesimpulan

Setelah penulis menguraikan beberapa permasalahan mengenai sistem penggajian karyawan yang ada pada PT. BPR Shinta Daya, maka pada bab ini penulis akan memberikan beberapa kesimpulan. Adapun kesimpulan yang dapat diambil adalah sebagai berikut :

1. Sistem informasi penggajian karyawan ini mempunyai fasilitas dan kemampuan sebagai berikut :
 - a. Fasilitas presensi karyawan yang langsung ke komputer akan memberi kemampuan penyimpanan dan pemrosesan data absensi karyawan secara otomatis. Kemampuan lain dari fasilitas ini adalah pencegahan terhadap karyawan yang masuk kerja tidak sesuai dengan jam kerjanya atau tidak termasuk dalam daftar lembur.
 - b. Fasilitas penghitungan gaji karyawan mempunyai kemampuan untuk menghitung gaji karyawan baik gaji tetap, maupun gaji lembur, dengan cepat dan kemudian mencetaknya sebagai slip gaji setiap saat bila diperlukan.

- c. Sistem informasi juga menyediakan fasilitas untuk Direktur sehingga Direktur dapat memperoleh informasi yang diperlukan dengan cepat dan *uptodate* tanpa harus menunggu laporan dari bagian Personalia atau bagian Keuangan.
 - d. Apabila dalam proses penghitungan gaji terdapat kesalahan, maka user dengan cepat dapat memperbaiki tanpa mengubah seluruh data yang telah dimasukkan sehingga data yang dihasilkan menjadi lebih akurat dan efisien dalam penggunaan waktu.
2. Setiap proses yang ada dalam sistem penggajian karyawan, dijelaskan secara visual dengan menggunakan UML, dalam bentuk diagram-diagram. Hal tersebut sangat membantu dalam memahami sistem sehingga setiap proses dalam sistem penggajian menjadi lebih jelas dan mudah dipahami oleh end-user maupun programmer. Oleh karena itu, maka diharapkan sistem penggajian ini dapat dibangun dan diimplementasikan secara efektif dan efisien.
3. Pengembangan dan implementasi dari perancangan sistem berorientasi objek mempunyai sifat dapat digunakan kembali (*Reuseable*) dan sifat pewarisan (*Inheritance*) sehingga pihak pengembang dapat menggunakan rancangan yang sudah ada ataupun membuat rancangan baru yang mewarisi sifat-sifat dari rancangan yang sudah ada.

5.2 Saran

Untuk menjamin terlaksananya sistem penggajian yang baik, penulis menganggap perlu dilakukan perbaikan. Adapun perbaikan dan pembenahan yang disarankan oleh penulis untuk perusahaan adalah sebagai berikut:

1. Mengadakan pengembangan sistem penggajian dengan metode UML sebagai tindak lanjut dari sistem penggajian yang sudah ada.
2. Perlu adanya pengawasan pada saat karyawan memasukkan kartu pada mesin absensi agar tidak terjadi kecurangan, untuk langkah preventif sebaiknya kartu presensi diganti dengan presensi yang melekat seperti sidik jari namun, dalam pelaksanaan sistem ini tetap dibutuhkan pengawasan baik saat presensi masuk, jam kerja maupun presensi pulang.
3. Diperlukan sosialisasi dan pelatihan terhadap sistem penggajian dengan UML sehingga dalam prakteknya tidak mengalami hambatan.
4. Untuk meningkatkan kinerja perusahaan perlu diimbangi antara peningkatan sumber daya manusia dengan sistem *software* yang memadai.

DAFTAR PUSTAKA

- Achmad S. Ruky, *Manajemen Penggajian dan Pengupahan untuk Karyawan Perusahaan*, edisi pertama, penerbit PT. Gramedia Pustaka Utama, Jakarta, 2002.
- A. Suhendar dan Hariman Gunadi, *Visual Modelling Menggunakan UML dan Rational Rose*, edisi pertama, penerbit Informatika, Bandung, 2002.
- Anies Basalamah, *Pengolahan data elektronik konsep untuk Manajer dan Auditor*, edisi pertama, penerbit Pustaka Binaman Presindo, Jakarta, 1995.
- Andri Kristanto, *Perancangan Sistem Informasi dan Aplikasinya*, edisi pertama, cetakan pertama, Bagian penerbit Gava Media, Yogyakarta, 2003.
- Blahe, Michael, *Object-Oriented Modeling and Design for Database Applications*, Prentice-Hall Inc, 1998
- Bray, Mike, *Object Oriented Analysis*, Carnegie Mellon University,
Download : tanggal 11 Februari, 2005,
http://www.sei.cmu.edu/str/descriptions/ooanalysis_body.html, 1997.
- _____, *Object Oriented Design*, Carnegie Mellon University, Download : tanggal 11 Februari 2005,
http://www.sei.cmu.edu/str/descriptions/oodesign_body.html, 1997.
- _____, *Object Oriented Programming Languages*, Carnegie Mellon University, Download: tanggal 11 Februari 2005,
http://www.sei.cmu.edu/str/descriptions/ooapl_body.html, 1997.
- Charibaldi, Novrido, *Model Berorientasi Objek Sistem Perkuliahan dan Evaluasi Pada Virtual University*, Program Pasca Sarjana Program Studi Ilmu Komputer Universitas Gadjah Mada Yogyakarta, 2001 .

- Erwin, Harry R., *History of Object Orientation*, University of Sunderland,
Download : tanggal 16 Desember 2004,
http://www.cet.sunderland.ac.uk/~cs0her/COM379_Lectures/Lecture1.pt,
2001.
- Erwin, Harry R., *History of Object Orientation*, University of Sunderland,
Download : tanggal 16 Desember 2004,
http://www.cet.sunderland.ac.uk/~cs0her/COM379_Lectures/Lecture3.ppt,
2001.
- Handoko, Tunggul, *Sistem Informasi Penggajian Pegawai Hotel*, Teknik
Informatika Universitas Kristen Duta Wacana Yogyakarta, 2002.
- Haryono Jusup, *Auditing*, Buku 1, Cetakan pertama, penerbit bagian penerbitan
Sekolah Tinggi Ilmu Ekonomi YKPN, Agustus 2001.
- Hasbi, Muhammad, *Analisis Perancangan Berorientasi Objek Sistem Informatika
Akutansi Keuangan di Sekolah Tinggi Manajemen Informatika dan
Komputer Sinar Nusantara Surakarta*, Program Pasca Sarjana Program
Studi Ilmu Komputer Universitas Gadjah Mada Yogyakarta, 2002.
- Ikatan Akuntansi Indonesia, *Standar Profesional Akuntansi Publik Per 1 Agustus
1994*, bagian penerbit Sekolah Tinggi Ilmu Ekonomi YKPN,
Yogyakarta, 1994.
- Julius Hermawan, *Analisa Desain dan Pemrograman Berorientasi Objek dengan
UML dan Visual Basic . Net*, Edisi pertama, cetakan pertama, penerbit Andi,
Yogyakarta, 2004.
- Kosasih, Ruchyat, (Trans). Barry E. Cushing, *Sistem Informasi Akuntansi dan
Organisasi Perusahaan*, edisi ketiga, Penerbit Erlangga, Jakarta, 1992.
- Michael Armstrong dan Helen Murlis, *Pedoman Praktis Sistem Penggajian*, edisi
kedua, penerbit PT. Pustaka Binaman Pressindo, 1993.
- Miller, Randy, *Practical UML : A Hands-On Introduction for Developers*,
Borland Software Corporation, Download: tanggal 11 Februari 2005
<http://bdn.borland.com/article/0,1410,31863,00.html>, 2003.

Mulyadi, *Sistem Akuntansi*, Edisi ke-3, Bagian Penerbit Sekolah Tinggi Ilmu Ekonomi YKPN, Yogyakarta, 1993.

_____, *Auditing*, Buku 2, cetakan pertama, penerbit bagian penerbitan Sekolah Tinggi Ilmu Ekonomi YKPN, Januari 2002.

Munawir H S, *Auditing Modern*, Edisi pertama, cetakan pertama, penerbit BPFE, Yogyakarta, Juli 1995.

Quatrani, Terry, *Introduction to the Unified Modelling Language*, Rational Software, Download: tanggal 6 Januari 2005
http://www.rational.com/media/uml/intro_rdn.pdf, 2001.

Rumbaugh, James et, *Object-Oriented Modeling and Design*, Prentice-Hall Inc. 1991.

Sinan Si Alhir, *The Object-Oriented Paradigm*, Download tanggal 16 Desember 2003
<http://home.earthlink.net/~salhir/TheObjectOrientedParadigm.PDF>, 1998.

Sinan Si Alhir, *Understanding the Unified Modeling Language (UML)*, Download: tanggal 16 Desember 2004
<http://home.earthlink.net/~salhir/UnderstandingTheUML.PDF>, 1999.

Sparks, Geoffrey, *An Introduction to UML : The Use Case Model*, Sparx Systems, Download: tanggal 11 Februari 2005
http://www.sparxsystems.com.au/Use_Case_Model.htm, 2004.

Zaki Baridwan, *Sistem Akuntansi*, Penyusun Prosedur dan Metode, edisi 5, cetakan ke-7, penerbit BPFE Yogyakarta, 1998.