BAB 2
LANDASAN TEORI

2.1 Program Pensiun

2.1.1 Definisi Program Pensiun

Berdasarkan Undang-Undang Republik Indonesia Nomor 11 tahun 1992 tentang Dana Pensiun didefinisikan bahwa program pensiun adalah setiap program yang mengupayakan manfaat pensiun bagi peserta. Manfaat pensiun itu berupa pembayaran berkala yang diberikan setelah peserta mencapai usia pensiun. Definisi tentang Dana Pensiun adalah badan hukum yang mengelola dan menjalankan program yang menjanjikan manfaat pensiun.

Ada dua jenis program pensiun, yaitu:

1. Program Pensiun Manfaat Pasti (*Defined Benefit Plan*)

   Yaitu program pensiun dimana manfaat yang akan didapatkan telah ditentukan, dan iuran berdasarkan penghitungan sehingga akan mencukupi untuk membayar manfaat yang dijanjikan.

2. Program Pensiun Iuran Pasti (*Defined Contribution Plan*)

   Yaitu program pensiun dimana iuran untuk pegawai telah ditetapkan terlebih dahulu, dan manfaat yang diperoleh adalah semua jumlah iuran ditambahkan dengan hasil investasi.

[Amin, 1995: 14-18]

Sedangkan metode evaluasi pendanaan dana pensiun terbagi menjadi dua, yaitu:

1. *Accrued Benefit Cost Method (ABCM)*

   Metode biaya manfaat yang disisihkan *(Accrued Benefit Cost Method).* Metode biaya aktuaria, yaitu iuran dalam satu tahun merupakan nilai sekarang dari tambahan jaminan dalam tahun itu.

   Pada metode ini, manfaat yang diperoleh adalah iuran yang umumnya lebih rendah dibandingkan dengan metode lainnya, dan hutangnya akan konsisten dengan target pengembangan manfaat karena kenaikan gaji yang digunakan dalam perhitungan biaya pensiun adalah sesuai dengan realisasinya. Kelemahan metode ini

adalah iuran akan cenderung naik apabila rata-rata umur peserta naik.

2. *Projected Benefit Cost Method (PBCM)*

Metode biaya manfaat yang diproyeksi *(Projected Benefit Cost Method).* Metode biaya aktuaria, yaitu iuran menggambarkan jaminan yang akan datang dan tingkat besarnya iuran (presentase gaji) sepanjang tahun.

Metode ini disertai asumsi kenaikan gaji setiap tahun yaitu secara periodik harus dilakukan peninjauan kembali mengenai dana pensiun yang ada. Peninjauan ini diadakan karena kemungkinan adanya penyimpangan asumsi yang menyebabkan kurangnya dana pensiun yang tersedia terutama dalam asumsi kenaikan gaji. Manfaat dari metode ini adalah iuran lebih stabil dan penyelenggaraan program pensiun lebih aman (sepanjang asumsi kenaikan gaji tidak menyimpang terlalu jauh).

[Amin, 1995: 17-18]

2.1.2 Manfaat Pensiun

Manfaat pensiun adalah pembayaran berkala yang dibayarkan kepada peserta pada saat dan dengan cara yang ditetapkan dalam peraturan Dana Pensiun.

Macam-macam manfaat pensiun antara lain:

1. Manfaat Pensiun Normal

   Manfaat pensiun bagi peserta, yang mulai dibayarkan pada saat peserta pensiun setelah mencapai usia pensiun normal atau sesudahnya.

2. Manfaat Pensiun Dipercepat

   Manfaat pensiun bagi peserta yang dibayarkan bila peserta pensiun pada usia tertentu sebelum usia normal.

3. Manfaat Pensiun Cacat

   Manfaat pensiun bagi peserta, yang dibayarkan bila peserta menjadi cacat.

4. Manfaat Pensiun Ditunda

> Merupakan hak atas manfaat pensiun bagi peserta yang berhenti bekerja sebelum mencapai usia pensiun normal, yang ditunda pembayarannya sampai pada saat peserta pensiun sesuai dengan Peraturan Dana Pensiun.
>
> [Amin, 1995: 12-13]

2.2 Asumsi Aktuaria

Asumsi Aktuaria merupakan harapan berdasarkan pengalaman masa lalu yang diperkirakan sesuai dengan keadaan sekarang atau masa depan. Terdapat tiga asumsi aktuaria yang dibutuhkan dalam program pensiun, yaitu:

1. Asumsi yang berhubungan dengan berbagai kecepatan penurunan peserta program pensiun, yang dikenal dengan asumsi penurunan (*decrement assumption*).
2. Asumsi mengenai gaji peserta di masa yang akan datang, yang dikenal sebagai asumsi gaji (*salary assumption*).
3. Asumsi mengenai nilai uang yang dikaitkan dengan waktu yang dikenal dengan asumsi bunga (*interest assumption*).

2.2.1 Asumsi Penurunan (*Decrement Assumption*)

Peserta program pensiun yang merupakan pegawai aktif, mempunyai kemungkinan untuk meninggal, berhenti, cacat, dan pensiun, sedangkan pegawai yang tidak aktif (telah pensiun) mempunyai kemungkinan untuk meninggal. Semua kemungkinan yang dapat terjadi pada peserta pensiun adalah berbagai tingkat *decrement*. Ada dua jenis *decrement*, yaitu *single-decrement* dan *multiple-decrement*. Pada *single-decrement*, hanya ada satu kemungkinan yang terjadi pada peserta, sedangkan pada *multiple-decrement* terdapat lebih dari satu kemungkinan yang terjadi. Sebagai contoh, sejak pegawai pensiun masuk dalam *single-decrement*, hanya mempunyai kemungkinan untuk meninggal. Pegawai yang masuk dalam *multiple-decrement* mempunyai kemungkinan untuk meninggal, keluar, cacat, dan pensiun.

Tingkat *decrement* biasanya disajikan dalam bentuk tabel yang disebut dengan tabel *decrement*. Tabel *decrement* adalah sebuah model matematika yang menganggap bahwa sekelompok orang menjadi

sasaran beberapa penyebab *decrement* yang *independent*, yang berlangsung terus-menerus. Sekelompok orang tersebut membentuk kelompok tertutup, dimana tidak ada peserta baru dan tidak ada peserta lama (sudah keluar) yang masuk kembali setelah terjadinya beberapa *decrement.*

Beberapa notasi yang digunakan:

 banyaknya orang yang mencapai usia (x) dalam kelompok karena berlangsungnya (1),(2),..(m) penyebab *decrement*.

$q_x^{(T)} =$ total probabilita bahwa (x) akan meninggalkan kelompok dalam 1 tahun disebabkan semua faktor penyebab, tanpa memperhatikan penyebabnya.

$p_x^{(T)} =$ total probabilita bahwa (x) akan tetap berada dalam kelompok dalam 1 tahun tanpa memperhatikan penyebabnya.

Pada *single decrement*, q_x didefinisikan sebagai tingkat *decrement* populasi dalam 1 tahun. Besaran q_x juga dapat diartikan sebagai probabilita *decrement* populasi dalam 1 tahun (*probability of decrement*). Jadi dalam *single decrement*, *probability of decrement* sama dengan tingkat *decrement*. $q'^{(k)}$ adalah notasi tingkat *decrement* dari kemungkinan ke (k) yang diambil dari *single decrement*, yang bebas dari sistem penurunan lainnya, sedangkan $q^{(k)}$ adalah notasi untuk probabilita *decrement* dari jenis ke (k) dalam *multiple decrement*, yang nilainya tidak bebas dari pengaruh sistem *decrement* lainnya.

Transformasi dari tingkat *decrement* pada probabilita *decrement* dalam *double-decrement* (k=1,2) di bawah asumsi UDD (*uniform distribution of death*) adalah

$$q^{(1)} = q'^{(1)}[1 - 1/2q'^{(2)}]$$

untuk tiga *decrement*, nilai dari $q^{(1)}$adalah

$$q^{(1)} = q'^{(1)}[1 - 1/2(q'^{(2)} + q'^{(3)}) + 1/3q'^{(2)}q'^{(3)}]$$

nilai untuk $q^{(1)}$ untuk tiga *decrement* dapat diaproksimasikan dengan

$$q^{(1)} \approx q'^{(1)}[1 - 1/2q'^{(2)}][1 - 1/2q'^{(3)}].$$

Jenis *decrement* berdasarkan penyebabnya antara lain:

2.2.1.1 *Decrement* karena Faktor Kematian (*Mortality Decrement*)

Kematian pada peserta aktif akan mencegah pencapaian status pensiun dan menerima manfaat pensiun normal, yaitu manfaat pensiun yang mulai diberikan pada usia pensiun normal. Kematian pada peserta nonaktif akan mengakhiri penerimaan manfaat pensiun normal.

Walaupun kematian akan mencegah atau mengakhiri penerimaan manfaat pensiun normal, tetapi akan menciptakan bentuk manfaat lain, seperti manfaat kematian yang dibayarkan secara sekaligus, atau berupa anuitas yang diberikan kepada ahli waris.

Faktor-faktor yang mempengaruhi tingkat kematian antara lain:

1. Usia, yaitu semakin tua seseorang maka semakin tinggi tingkat kematiannya,
2. Gender, berdasarkan studi empirik, wanita cenderung mempunyai tingkat kematian yang lebih rendah dibandingkan pria, dan
3. Pekerjaan, karena ada jenis pekerjaan yang mempunyai resiko kematian yang tinggi.

Jika $q_x'^{(m)}$ adalah notasi untuk tingkat kematian pada usia x, dan $p_x'^{(m)}$ adalah notasi untuk probabilitas seseorang berusia x akan bertahan hidup hingga usia $x+1$, dengan asumsi tidak ada faktor *decrement* lainnya,

$$p_x'^{(m)} = 1 - q_x'^{(m)} \quad ,$$

maka probabilitas seseorang berusia x akan bertahan hidup hingga n tahun kemudian, dengan asumsi tidak ada faktor *decrement* lainnya adalah ${}_np_x'^{(m)}$. ${}_np_x'^{(m)}$ diformulasikan sebagai berikut:

$${}_np_x'^{(m)} = \prod_{t=0}^{n-1} p_{x+t}'^{(m)}$$

$$= \prod_{t=0}^{n-1} (1 - q_{x+t}^{\prime(m)})$$

2.2.1.2 *Decrement* karena Faktor Keluar (*Withdrawal Decrement*)

Keluar atau berhentinya peserta aktif akan mencegah pencapaian status pensiun dan mencegah penerimaan manfaat pensiun normal. Faktor yang mempengaruhi tingkat keluar adalah faktor usia dan masa kerja. Semakin tua usia peserta atau semakin lama masa kerja peserta maka semakin kecil tingkat keluarnya.

Jika $q_x^{\prime(w)}$ adalah notasi untuk tingkat keluar pada usia x, dan $p_x^{\prime(w)}$ adalah notasi untuk probabilitas seseorang yang berusia x akan tetap bekerja 1 tahun kemudian, dengan asumsi tidak ada *decrement* lainnya, $p_x^{\prime(w)} = 1 - q_x^{\prime(w)}$

maka probabilitas seseorang berusia x akan tetap bekerja n tahun kemudian, dengan asumsi tidak ada *decrement* lainnya adalah ${}_np_x^{\prime(w)}$ ${}_np_x^{\prime(w)}$ diformulasikan dengan

$${}_np_x^{\prime(w)} = \prod_{t=0}^{n-1} p_{x+t}^{\prime(w)} = \prod_{t=0}^{n-1} (1 - q_{x+t}^{\prime(w)}).$$

2.2.1.3 *Decrement* karena Faktor Kecacatan (*Disability Decrement*)

Seperti pada *decrement* karena faktor kematian dan *decrement* karena faktor keluar, *decrement* karena faktor kecatatan akan mencegah peserta aktif mencapai status pensiun dan menerima manfaat pensiun normal. Ketika peserta mengalami kecacatan maka ia bisa memperoleh manfaat kecacatan (*disability benefit*).

Faktor yang mempengaruhi tingkat kecacatan yaitu faktor usia, gender dan pekerjaan.

Jika $q_x^{\prime(d)}$ adalah notasi untuk tingkat kecacatan pada usia x, dan $p_x^{\prime(d)}$ adalah notasi untuk probabilitas seseorang yang berusia x tidak mengalami cacat 1 tahun kemudian, dengan asumsi tidak ada *decrement* lainnya, $p_x^{\prime(d)} = 1 - q_x^{\prime(d)}$

Maka probabilitas seseorang berusia x tidak mengalami cacat n tahun kemudian, dengan asumsi tidak ada *decrement* lainnya adalah ${}_np_x^{\prime(d)}$, ${}_np_x^{\prime(d)}$ diformulasikan dengan

$${}_np_x^{\prime(d)} = \prod_{t=0}^{n-1} p_{x+t}^{\prime(d)} = \prod_{t=0}^{n-1} (1 - q_{x+t}^{\prime(d)})$$

2.2.1.4 *Decrement* karena Faktor Pensiun (*Retirement Decrement*)

Tidak seperti pada *decrement* lainnya yang mencegah peserta aktif mencapai status pensiun dan menerima manfaat pensiun normal, *decrement* karena faktor pensiun akan mengawali peserta aktif untuk menfaat pensiun normal.

Pensiun yang terjadi pada usia pensiun normal disebut pensiun usia normal. Adakalanya peserta lebih memilih pensiun sebelum mencapai usia pensiun normal, yang disebut dengan pensiun dipercepat. Terjadinya pesiun dipercepat dipengaruhi oleh banyak faktor seperti faktor lamanya masa kerja, status kesehatan, besar manfaat pensiun, pekerjaan, gender dan uisa. Di Indonesia, peserta program pensiun dapat memilih pensiun dipercepat apabila berusia sekuang-kurangnya 10 tahun dari usia pensiun normal. Notasi untuk tingkat pensiun pada usia x adalah $q_x^{\prime(r)}$.

[Winklevoss, 1976:10-22]

2.2.2 Asumsi Aktuaria Ekonomi

2.2.2.1 Asumsi Tingkat Bunga

Asumsi tingkat bunga memberi pengaruh yang kuat dalam pembiayaan pensiun, karena asumsi ini digunakan untuk mencari *present value* dari nilai uang di masa depan.

Asumsi tingkat bunga terdiri dari tiga komponen, yaitu komponen suku bunga murni, komponen yang dikaitkan dengan resiko investasi dan komponen yang berkaitan dengan inflasi.

1. Komponen suku bunga murni

Komponen suku bunga murni adalah komponen suku bunga yang dibuat untuk mengatasi apabila terjadi kondisi tidak terjadi inflasi atau apabila 100% tidak ada resiko dalam investasi.

2. Komponen yang dikaitkan dengan resiko investasi

   Komponen yang dikaitkan dengan resiko investasi yaitu komponen yang dibuat untuk mengatasi resiko investasi dari portofolio aset program di masa sekarang dan masa depan. besarnya resiko investasi (*risk premium*) tergantung pada jenis investasinya, yang biasanya dibagi dalam beberapa kategori seperti saham, obligasi perusahaan dan obligasi pemerintah. *Risk premium* idealnya berdasarkan pengalaman masa lalu, *yield* sekarang dan antisipasi dari pengembalian investasi di masa depan.

3. Komponen yang berkaitan dengan inflasi

   Komponen ini dibuat untuk mengantisipasi kerugian akibat adanya inflasi. Asumsi ini mempunyai subyektifitas yang tinggi dibandingkan asumsi aktuaria lainnya karena sejarah tingkat inflasi cenderung bukan prediksi tingkat inflasi masa depan yang baik.

[Winklevoss, 1976:26-28]

2.2.2.2 Asumsi Tingkat Kenaikan Upah

Seringkali besar manfaat pensiun dan iuran peserta pensiun merupakan fungsi dari gaji, karena itu harus dilakukan estimasi terhadap berapa besar kenaikan gaji peserta pensiun di masa depan. Estimasi tersebut mempertimbangkan tiga komponen, yaitu:

1. Peningkatan gaji karena peningkatan jasa

Peningkatan gaji karena peningkatan jasa adalah peningkatan gaji yang akan diterima seorang pekerja karena kemajuan dalam karirnya dan kemampuan yang semakin meningkat seiring dengan bertambahnya usia dan masa kerja.

2. Peningkatan gaji karena produktivitas keuntungan perusahaan

   Peningkatan gaji karena peningkatan produktivitas perusahaan adalah peningkatan gaji yang akan diterima seorang pekerja karena perusahaan tempatnya bekerja memperoleh laba akibat peningkatan produktivitas. Besarnya peningkatan gaji ini tergantung pada laba yang diperoleh perusahaan dan seberapa andil peserta dalam laba yang diperoleh perusahaan tersebut.
3. Peningkatan gaji karena adanya inflasi

   Faktor yang paling signifikan terhadap kenaikan gaji adalah inflasi. Setiap program pensiun dapat mengasumsikan tingkat inflasi yang berbeda.

[Winklevoss, 1976:23-26]

2.3 Dasar-Dasar Aktuaria

2.3.1 Composite Survival Function

Misalkan $T(x)$ adalah peubah acak dengan fungsi distribusi peluang, maka berlaku:

$${}_tq_x = P(T(x) \leq t) \quad t \geq 0,$$

dan

$${}_tp_x = 1 - {}_tq_x = P(T(x) > t) \quad t \geq 0,$$

dengan ${}_tq_x$ adalah peluang (x) meninggal dalam jangka waktu t tahun dan ${}_tp_x$ adalah peluang (x) akan hidup t tahun lagi atau mencapai usia $x + t$ tahun.

[Bowers, dkk., 1997:53]

Misalkan l_x adalah jumlah orang yang hidup pada usia x, maka peluang (x) mencapai usia $x+t$ tahun adalah ${}_tp_x = \frac{l_{x+t}}{l_x}$ dan peluang (x) meninggal dalam jangka waktu t tahun adalah

${}_tq_x = 1 - {}_tp_x = \frac{l_x - l_{x+t}}{l_x}$.

Sedangkan peluang seseorang berusia 0 tahun dapat bertahan hidup hingga x tahun ke depan merupakan fungsi bertahan hidup *(survival functions)* yaitu $s_x = \frac{l_x}{l_0} = {}_xp_0$

[Futami T., 1993:34]

Probabilitas pegawai aktif berusia (x) tahun bertahan untuk 1 tahun dinotasikan dengan $p_x^{(T)}$ dan dinotasikan dengan:

$$p_x^{(T)} = p_x^{\prime(m)} p_x^{\prime(w)} p_x^{\prime(d)} p_x^{\prime(r)}$$

equivalen dengan:

$$p_x^{(T)} = 1 - q_x^{(T)}$$

$$= (1 - q_x^{\prime(m)})(1 - q_x^{\prime(w)})(1 - q_x^{\prime(d)})(1 - q_x^{\prime(r)})$$

$$= 1 - \left(q_x^{\prime(m)} + q_x^{\prime(w)} + q_x^{\prime(d)} + q_x^{\prime(r)}\right).$$

dengan $q_x^{(T)}$ adalah notasi dari total probabilitas bahwa (x) akan meninggalkan kelompok dalam 1 tahun disebabkan semua faktor penyebab, tanpa memperhatikan penyebabnya.

Probabilita pegawai aktif berusia (x) tahun, bertahan untuk n tahun kemudian adalah

$${}_np_x^{(T)} = \prod_{t=0}^{n-1} p_{x+t}^{(T)},$$

Misal total banyaknya orang yang bertahan pada usia (x) tahun adalah $l_x^{(T)}$. Total banyaknya pegawai yang meninggalkan pekerjaan aktifnya sepanjang tahun ($d_x^{(T)}$) adalah: $d_x^{(T)} = l_x^{(T)} q_x^{(T)}$

total populasi aktif yang mengalami *decrement* adalah

$$d_x^{(T)} = d_x^{(m)} + d_x^{(t)} + d_x^{(d)} + d_x^{(r)}$$

$$= l_x^{(T)} (q_x^{(m)} + q_x^{(t)} + q_x^{(d)} + q_x^{(r)})$$

[Winklevoss, 1976:29-30]

2.3.2 Fungsi Anuitas Hidup (*Life Annuity)*

Anuitas hidup (*Life Annuity)* adalah serangkaian pembayaran yang dilakukan kepada seseorang selama orang tersebut hidup. Anuitas hidup terbagi menjadi dua bagian, yaitu: anuitas hidup kontinu dan anuitas hidup diskrit. Anuitas hidup diskrit sendiri terbagi lagi menjadi dua bagian berdasarkan sistem pembayaran yang dilakukan, yaitu: anuitas hidup diskrit dengan anuitas akhir (*annuity-immediate)* dan anuitas hidup diskrit dengan anuitas awal (*annuity-due)*. *Annuity due* adalah pembayaran anuitas yang dilakukan di awal periode sedangkan *annuity immediate* adalah pembayaran anuitas yang dilakukan pada akhir periode. Anuitas seumur hidup (*whole life annuity)* merupakan anuitas yang dilakukan hingga meninggal. *n-year deffered whole life annuity* merupakan anuitas yang dilakukan sejak n tahun yang akan datang hingga meninggal. Fungsi anuitas hidup sendiri dapat digunakan dalam *single-life* dan *multi-life*.

[Bowers, 1997:135-148]

2.3.2.1 Anuitas Seumur Hidup Kontinu (*Whole Life Annuity kontinu)*

Actuarial present value dari *whole life annuity* sebesar 1 setiap tahun, yang dibayarkan secara kontinu dari usia (x) hingga meninggal dinotasikan dengan $\bar{a}_x$.

Misalkan Y adalah variabel acak yang berupa *present value*, dan T adalah variabel acak kontinu untuk *time until death*:

$$Y = \bar{a}_{\overline{T}|}, T \geq 0$$

probabilita yang bersesuaian dengan $\bar{a}_{\overline{T}|}$ adalah

$$\bar{a}_x = E[Y] = \int_0^\infty \bar{a}_{\overline{T}|} \; {}_tp_x \; \mu_{x+t} dt = \int_0^\infty v^t \; {}_tp_x \; dt$$

[Bowers, dkk., 1997:134-135]

2.3.2.2 *n-year deferred whole life annuity*

Actuarial present value dari *n-year deferred whole life annuity* sebesar 1, yang dibayarkan secara kontinu dan dimulai dari usia $(x+n)$ hingga (x) meninggal dinotasikan dengan ${}_{n|}\bar{a}_x$.

Misalkan Y adalah variabel acak yang berupa *present value*, dan T adalah variabel acak kontinu untuk *time until death*:

$$Y = \begin{cases} 0 & 0 \leq T < n \\ v^n \; \bar{a}_{\overline{T-n}|} & T \geq n \end{cases}$$

$${}_{n|}\bar{a}_x = E[Y] = \int_0^\infty v^n \bar{a}_{\overline{T-n}|} \; {}_tp_x \; \mu_{x+t} dt = v^n \; {}_np_x \; \bar{a}_{x+n}$$

[Bowers, dkk., 1997:145-146]

2.3.2.3 Anuitas Seumur Hidup dengan Anuitas Awal

Actuarial present value dari *whole life annuity* sebesar 1, yang dibayarkan setiap awal tahun dan dimulai dari usia (x) hingga meninggal dinotasikan dengan $\ddot{a}_x$.

Misalkan Y adalah variabel acak yang berupa *present value*, dan K adalah variabel acak diskrit untuk *time until death*: $Y = C$ jika K=0,1,...

probabilita yang bersesuaian dengan $\ddot{a}_{\overline{k+1}|}$ adalah

$$f_k(k) = P_r(K = k) = {}_kp_x \; q_{x+k}$$

$$\ddot{a}_x = E[Y] = \sum_{k=0}^{\infty} \ddot{a}_{\overline{k+1}|} \; {}_kp_x \; q_{x+k} = \sum_{k=0}^{\infty} v^k \; {}_kp_x$$

$$= \sum_{k=0}^{\infty} \frac{v^{x+k} \, l_{x+k}}{v^x \, l^x} .$$

Misalkan $D_x = v^x l_x$ dan $N_x = \sum_{k=0}^{\infty} D_{x+k}$, maka $\ddot{a}_x = \frac{N_x}{D_x}$

[Bowers, dkk., 1997:143-144]

2.3.2.4 *n-year temporary life annuity due*

Anuitas hidup yang dibayarkan pada awal tahun selama jangka waktu n tahun *(n-year temporary life annuity due)* adalah anuitas hidup yang dibayarkan pada awal tahun selama jangka waktu n tahun. *Present value* dari *n-year temporary life annuity due* sebesar 1 per tahun adalah

$$Y = \ddot{a}_{\overline{k+1}|} \quad ,untuk \; k = 0,1,2,\dots,n-1$$
$$= \ddot{a}_{\overline{n}|} \; ,untuk \; k = n, n+1, n+2, \dots$$

$\ddot{a}_{\overline{x:n}|}$ adalah *actuarial present value* dari anuitas ini, yang diformulasikan sebagai berikut:

$$\ddot{a}_{\overline{x:n}|} = E[Y]$$
$$= \sum_{k=0}^{n-1} \ddot{a}_{\overline{k+1}|} P_r(K = k) + \sum_{k=n}^{\infty} \ddot{a}_{\overline{n}|} P_r(K = k)$$
$$= \sum_{k=0}^{\infty} v^k \; {}_kp_x \; .$$

Variasi penting dari $\ddot{a}_{\overline{x|n}}$ adalah ${}^{s}\ddot{a}_{\overline{x|r-x|}}$, di mana *supersscript* s menunjukan bahwa anuitas tersebut berdasarkan gaji. Jika s_x adalah notasi dari gaji seseorang saat berusia x, maka *present value* dari anuitas hidup awal berjangka berdasarkan gaji seseorang di masa depan mulai dari usia x hingga usia r, per satuan gajinya pada usia x adalah

$$Y = \frac{s_x}{s_x} + v\frac{s_{x+1}}{s_x} + v^2\frac{s_{x+2}}{s_x} + \cdots + v^k\frac{s_{x+k}}{s_x}$$

$$= \sum_{t=0}^{k} v^t \frac{s_{x+t}}{s_x} \quad , k = 0,1, \dots , r-x-1$$

$$= \frac{s_x}{s_x} + v\frac{s_{x+1}}{s_x} + v^2\frac{s_{x+2}}{s_x} + \cdots + v^{r-x-1}\frac{s_{r-1}}{s_x}$$

$$= \sum_{t=0}^{r-x-1} v^t \frac{s_{x+t}}{s_x} \quad , k = r-x, r-x+1, \dots$$

${}^{s}\ddot{a}_{\overline{x|r-x|}}$ adalah *actuarial present value* dari *life annuity due* berdasarkan gaji seseorang di masa depan mulai dari usia x hingga usia r, per satuan gajinya pada usia x.

$${}^{s}\ddot{a}_{\overline{x|r-x|}} = E(Y)$$

$$= \sum_{k=0}^{r-x-1} \sum_{t=0}^{k} v^t \frac{s_{x+t}}{s_x}(-\Delta_k p_x) + \sum_{k=r-x}^{\infty} \sum_{t=0}^{r-x-1} v^t \frac{s_{x+t}}{s_x}(-\Delta_k p_x)$$

$$= \sum_{k=0}^{\infty} {}_k p_x \; v^k \frac{s_k}{s_x}.$$

[Bowers, 1997:144-145]

2.3.3 Fungsi Tingkat Suku Bunga

Definisi Bunga

- Bunga adalah kompensasi pembayaran dari peminjam suatu modal kepada yang meminjamkan modal tersebut
- Nilai pokok adalah sejumlah uang yang diinvestasikan pada saat awal

- Nilai akumulasi adalah jumlah total uang yang diterima sesudah periode waktu tertentu
- Besar bunga adalah selisih nilai akumulasi sesudah periode waktu tertentu dengan nilai pokok pada saat awal periode

[Kellison, 1991:1-2]

DefinisiTingkat Diskonto

Tingkat diskon efektif (d) adalah rasio dari besarnya diskonto yang diperoleh selama periode tertentu terhadap besarnya nilai akumulasi pada akhir periode. Dimana d dapat dinyatakan sebagai

$$d = \frac{i}{1+i}$$

[Kellison, 1991:14]

Definisi Faktor Diskonto

Nilai saat ini adalah investasi sebesar 1 yang akan terakumulasi menjadi $1+i$ pada akhir peiode ke 1. Nilai saat ini juga bisa disebut dengan faktor diskonto yang dinotasikan dengan v dan dapat dinyatakan sebagai $v = \frac{i}{1+i}$

[Kellison, 1991:10]

Fungsi tingkat suku bunga digunakan untuk mendiskontokan pembayaran yang akan datang pada saat ini. Jika i adalah tingkat suku bunga yang diasumsikan untuk tahun ke t, maka nilai masa kini *(present value)* dari 1 yang harus dibayar setelah n tahun adalah:

$$\frac{1}{(1+i_1)(1+i_2)\dots(1+i_n)}$$

dan jika $i_1 = i_2 = \cdots = i_n = i$, maka menjadi $\frac{1}{(1+i)^n} = v^n$

dimana v^n menyatakan *present value* dari nilai 1 pada tahun ke-n pada sistem pembungaan *compound interest*, dengan suku bunga efektif i.

dimana:

v^n = nilai sekarang dari pembayaran sebesar 1 satuan yang dilakukan n tahun kemudian.

[Winklevoss, 1976:33]

2.3.4 Fungsi Gaji

Jika program pensiun dikaitkan dengan gaji peserta, maka diperlukan pengembangan notasi untuk gaji dan prosedur untuk menaksir gaji yang akan datang. Gaji saat ini untuk peserta berusia x dilambangkan dengan s_x, dan S_x merupakan akumulasi gaji dari usia masuk (y) sampai usia $x-1$, dimana $x \geq y$ diformulasikan dengan :

$$S_x = \sum_{t=y}^{x-1} s_t, \qquad x \geq y.$$

Apabila karyawan mendapat proporsi kenaikan gaji sebesar s setiap tahun, maka besarnya gaji karyawan pada saat berusia $x+t$, berdasarkan gaji pada saat x adalah

$$s_{x+t} = s_x(1+s)^t.$$

[Winklevoss, 1976:34-35]

2.3.5 Fungsi Manfaat

Fungsi Manfaat (*benefit function)* berfungsi untuk menentukan besar gaji yang akan diterima pegawai pada saat pensiun normal, pensiun dini, keluar (*vested)*, cacat atau meninggal. Misalkan b_x menyatakan *benefit accrual* yang dibayarkan pada setiap tahun untuk jangka waktu x sampai dengan $x+1$, maka jumlah dari manfaat-manfaat pensiun (*accrued benefit*) seseorang dari sejak usia masuk y sampai dengan usia x adalah sebagai berikut:

$$B_x = \sum_{t=y}^{x-1} b_t.$$

Pada perumusan program pensiun dikenal tiga perumusan manfaat pensiun, yaitu:

1. *Flat benefit*, merupakan jumlah *benefit accrual* yang dibayarkan setiap tahunnya sama, sehingga *benefit accrual* kumulatifnya hanya perkalian dengan masa kerja, sebagai berikut: $B_x = (x-y)b_x$.

2. *Career average*, dimana jumlah *benefit accrual* yang dibayarkan setiap tahunnya berdasarkan presentase tetap dari rata-rata gaji pegawai dalam setahun.

$$b_x = k\, s_x$$
$$= k\, S_x$$

dimana:

k = merupakan proporsi gaji setiap tahun yang dibayarkan sebagai *benefit accrual* ($0 \leq k \leq 1$)

s_x = gaji pada saat usia x

S_x = jumlah gaji dari usia y sampai dengan usia $x - 1$

3. *Final average*, adalah penentuan jumlah *benefit accrual* berdasarkan rata-rata gaji beberapa tahun terakhir.

$$B_x = k(r - y)\frac{1}{n}\sum_{t=r-n}^{r-1} s_t$$

dimana:

r = usia pensiun normal karyawan

n = jumlah tahun untuk penghitungan rata-rata gaji terakhir.

Terdapat dua modifikasi dari *benefit accrual*, yaitu:

1. Modifikasi CA (*Constant Amount*); merupakan perkembangan dari fungsi benefit berdasarkan *flat benefit*, di mana *pension benefit* ditentukan berdasarkan nilai yang sama setiap bulannya (biasanya dalam dollar). Dalam modifikasi ini bx ditransformasikan ke dalam *pro rate share* dari *projected benefit*, dilambangkan dengan ${}^{CA}b_x$.
2. Modifikasi CS (*Constant Percentage of Salary*); merupakan perkembangan dari fungsi benefit berdasarkan *career average*, di mana pension benefit ditentukan berdasarkan persentase tertentu dari gaji. Dalam modifikasi ini bx ditransformasikan ke nilai yang setara dengan persentase gaji konstan setiap tahun, dilambangkan dengan ${}^{CS}b_x$.

[Winklevoss, 1976:36-40]

2.4 Konsep Dasar dan Pendanaan Program Pensiun

2.4.1 *Projected Benefit Cost Method (*PBCM*)*

*Projected Benefit Cost Method (*PBCM*)* merupakan metode penilaian aktuarial yang menunjukan nilai *benefit*, berdasarkan jasa yang telah diberikan peserta sampai dengan tanggal penilaian. Metode ini mengalokasikan biaya dari *benefit* secara merata (dinyatakan dalam jumlah tertentu atau sebagai presentase dari gaji) selama masa kerja karyawan. Pada *Projected Benefit Cost Method*, setiap karyawan di asumsikan telah menjadi peserta pensiun ketika pertama kali dipekerjakan atau segera setelah karyawan tersebut memenuhi syarat. Biaya jasa kini didefinisikan sebagai tingkat jumlah tahunan atau presentase tetap dari gaji, yang jika diinvestasikan pada tingkat bunga yang diasumsikan, cukup untuk mebayar *benefit* sesuai dengan yang ditetapkan. Biaya jasa lalu didefinisikan sebagai nilai sekarang dari kelebihan proyeksi *benefit* terhadap jumlah yang diharapkan tersedia dari iuran di masa mendatang berdasarkan biaya masa kini.

[Winklevoss, 1976:92-95]

2.4.2 Nilai Sekarang (*Present Value Future Benefit*)

Present Value of Future Benefit adalah nilai sekarang dari manfaat pensiun berkala, yang akan diterima oleh peserta program pensiun di masa yang akan datang.

Present Value of Future Benefit untuk seorang peserta sekarang berusia x, masuk program pada usia y, dan akan pensiun pada usia r, dinotasikan dengan $(PVFB)_x$.

$$(PVFB)_x = B_r \; {}_{r-x}p_x^{(T)} \; v_L^{r-x} \; \ddot{a}_r.$$

Dengan :

B_r = manfaat pensiun berkala yang akan diterima oleh peserta setelah pensiun

${}_{r-x}p_x^{(T)}$ = total probabilitas bahwa peserta berusia x akan tetap bertahan dalam program pensiun hingga usia r

v_L^{r-x} = diskonto bunga kewajiban pensiun dari usia x ke r

$\ddot{a}_r$ = actuarial present value dari anuitas seumur hidup dengan pembayaran segera sebesar 1 yang dibayarkan mulai dari usia r.

Manfaat yang diberikan kepada peserta pensiun merupakan *n-year deffered whole life annuity due* sebesar B_r, yang dibayarkan mulai dari usia r. Diketahui bahwa *actuarial present value* dari *n-year defferred whole life annuity due* sebesar 1, yang dibayarkan pada awal tahun, selama seseorang berusia$x$ bertahan hidup mulai dari usia $x+n$ adalah

$${}_{n|}\ddot{a}_x = v^n {}_np_x \ \ddot{a}_{x+n}.$$

Maka *actuarial present value* dari *n-year deffered whole life annuity due* sebesar B_r, yang dibayarkan setiap awal tahun selama peserta (x) bertahan hidup mulai dari usia $x+n(n=r-x)$, yang dihitung dengan tingkat bunga atas kewajiban pensiun i_L adalah

$$B_r\, {}_{r-x|}\ddot{a}_x = B_r v_L^{r-x} {}_{r-x}p_x \ \ddot{a}_{x+(r-x)} = B_r v_L^{r-x} {}_{r-x}p_x \ \ddot{a}_r.$$

Hanya saja ${}_{r-x}p_x$ yang digunakan adalah ${}_{r-x}p_x^{(T)}$, yaitu total probabilitas bahwa peserta program berusia x akan tetap bertahan dari segala tingkat *decrement* hingga usia r. Setelah usia pensiun $(x \geq r)$, *Present Value of Future Benefit*nya akan menjadi

$$(PVFB)_x = B_r \ddot{a}_x, \quad x \geq r,$$

dengan

$\ddot{a}_x$ = actuarial present value dari anuitas seumur hidup dengan pembayaran segera sebesar 1 yang dibayarkan mulai dari usia x.

Present Value of Future Benefit untuk suatu program adalah penjumlahan atas *Present Value of Future Benefit* untuk masing-masing peserta program pensiun.

[Winklevoss, 1976:65-69]

2.4.3 Normal Cost

Dalam PBCM untuk mendapatkan *benefit* setelah pensiun, maka peserta harus mencicil sejumlah *normal cost* yang dibayarkan pada saat

usia x sampai dengan pensiun pada usia k. Besarnya *normal cost* yang ditentukan dengan PBCM ini bisa dengan dua cara, yaitu yang sifatnya konstan terhadap jumlah ($^{CA}PBCM$) dan konstan terhadap presentase gaji ($^{CS}PBCM$).

Berikut persamaan dari normal cost untuk $^{CA}PBCM$:

$$^{CA}(NC)_x = \frac{(PVFB)_y}{\ddot{a}_{\overline{yr-y|}}}$$

Berikut ini persamaan untuk $^{CS}PBCM$:

$$^{CS}(NC)_x = K.s_x$$

Dengan :

$\ddot{a}_{\overline{yr-y|}}$ = anuitas hidup berjangka yang dibayarkan di awal tahun, nilainya sesuai dengan gaji berdasarkan gaji pada saat masuk (y).

K = presentase konstan yang ditetapkan

s_x = gaji pada usia x

[Winklevoss, 1976:72-74]

2.5 Landasan Teori Perancangan Program

2.5.1 *Unified Modelling Language (*UML*)*

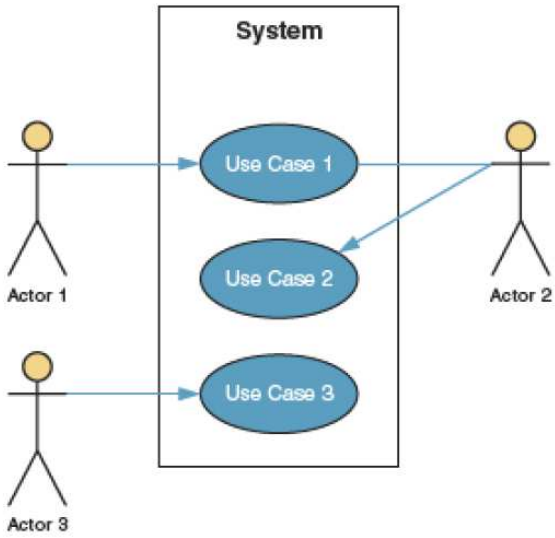
Whitten dan Bentley (2007) mengatakan *Unified Modelling Language (UML)* adalah *blueprint* dari sistem informasi yang akan dibuat dalam pengembangan aplikasi. Setiap UML memiliki fungsi dan tujuan yang berbeda dalam pengembangan aplikasi sesuai dengan jenisnya.

UML memiliki banyak struktur dan diagram dalam pemodelannya. Pada pembuatan aplikasi ini akan digunakan beberapa diagram seperti *use case diagram, activity diagram, class diagram,* dan *sequence diagram.*

2.5.1.1 *Use Case*

2.5.1.1.1 *Use Case Diagram*

Use case diagram adalah diagram yang menggambarkan interaksi antara sistem, eksternal sistem, dan pengguna. Diagram ini menyediakan informasi mengenai siapa saja yang akan menggunakan sistem tersebut dan bagaimana cara untuk menggunakannya. Komponen-komponen dalam *use case diagram* adalah sebagai berikut :



Gambar 2.1 *Use Case Diagram*

(Sumber : Whitten & Bentley, 2007:246)

a) Use cases

Use cases memperlihatkan fungsi-fungsi dari sistem. Use case digambarkan dengan bentuk elips horisontal dengan nama dari *use case* tersebut terletak di bawah, di atas, atau di dalam elips tersebut.

Gambar 2.2 *Use Case*

(Sumber : Whitten dan Bentley, 2007 : 246)

b) *Actor*

Actor adalah *user* yang akan berinteraksi dalam sistem dengan melakukan *use case* untuk bertukar informasi. *Actor* digambarkan dalam bentuk *stick figure* dengan label peran *actor* tersebut dalam sistem.

Gambar 2.3 *Actor*

(Sumber : Whitten dan Bentley, 2007:247)

c) *Relationship*

Relationship adalah hubungan dari dua symbol dalam *use case diagram* yang digambarkan dalam bentuk garis. Arti dari hubungan tersebut berbeda-beda tergantung jenis garis dan symbol yang dihubungkan. Berikut adalah beberapa hubungan yang ada dalam *use case diagram*:

1. *Associations*

   Komunikasi antara *use case* dan *actor* digambarkan sebagai *associations.* Garis *associations* dapat memiliki anak panah yang berarti *actor* berperan sebagai pelaku dari *use case* tersebut, sedangkan garis tanpa anak panah berarti *actor* hanya berperan sebagai *external database* atau penerima dari *use case* tersebut.

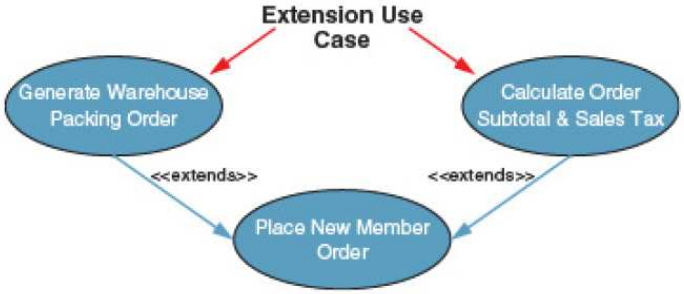
Gambar 2.4 Association

(Sumber : Whitten dan Bentley, 2007:248)

2. *Extends*

   Sebuah *use case* dari fungsi-fungsi yang kompleks, terdiri dari beberapa tahap-tahap yang dapat membuat *use case* tersebut sulit untuk dipahami.

   Untuk menyederhanakan *use case* tersebut agar mudah dimengerti, dapat di lakukan pemisahan menjadi *use case* tersendiri yang dinamakan *extension use case*. Hubungan antara *extension use case* dengan *use case* itu sendiri dinamakan *extends relationship*.

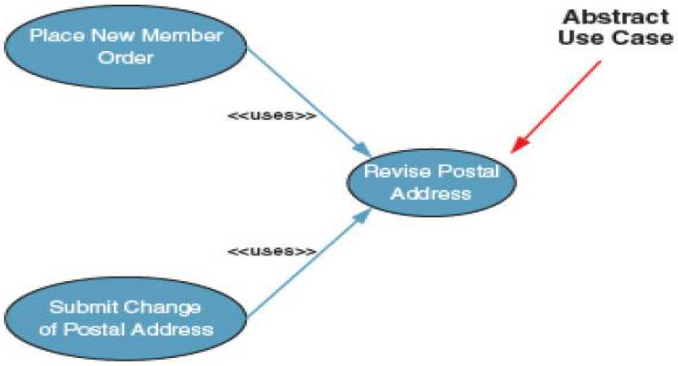


Gambar .5 *Extends*

(Sumber : Whitten dan Bentley, 2007 : 249)

e) Uses (Includes)

Uses (includes) adalah *relationship* antara *use case* dengan *abstract use case* yang digunakan. *Abstract use case* adalah sebuah *use case* yang digunakan untuk mengurangi *redundancy* dari dua *use case* atau lebih yang memiliki langkah yang sama dengan menggabungkan langkah-langkah yang terdapat dalam *use case* tersebut. *Abstract use case* ditunjukan dengan arah panah yang mengarah dari *use case* ke arah *abstract use case* tersebut.



Gambar 2.6 *Includes*

(Sumber : Whitten dan Bentley, 2007:249)

2.5.1.2 Use Case Narrative

Use case narrative adalah deskripsi tertulis dari setiap *event* pada *use case* dan bagaimana *user* berinteraksi dengan sistem untuk menyelesaikan tugas.

2.5.2 Activity Diagram

Activity Diagram digunakan untuk menggambarkan aliran berurutan dari sebuah proses *use case* atau *business process*. Selain itu, dapat juga digunakan untuk logika model dengan sistem yaitu, menggambarkan tindakan (*action*) yang akan dijalankan ketika suatu proses sedang berjalan dan beserta hasil dari proses yang dijalankan tersebut.Terdapat beberapa komponen dalam menggambarkan *activity diagram*, yaitu :

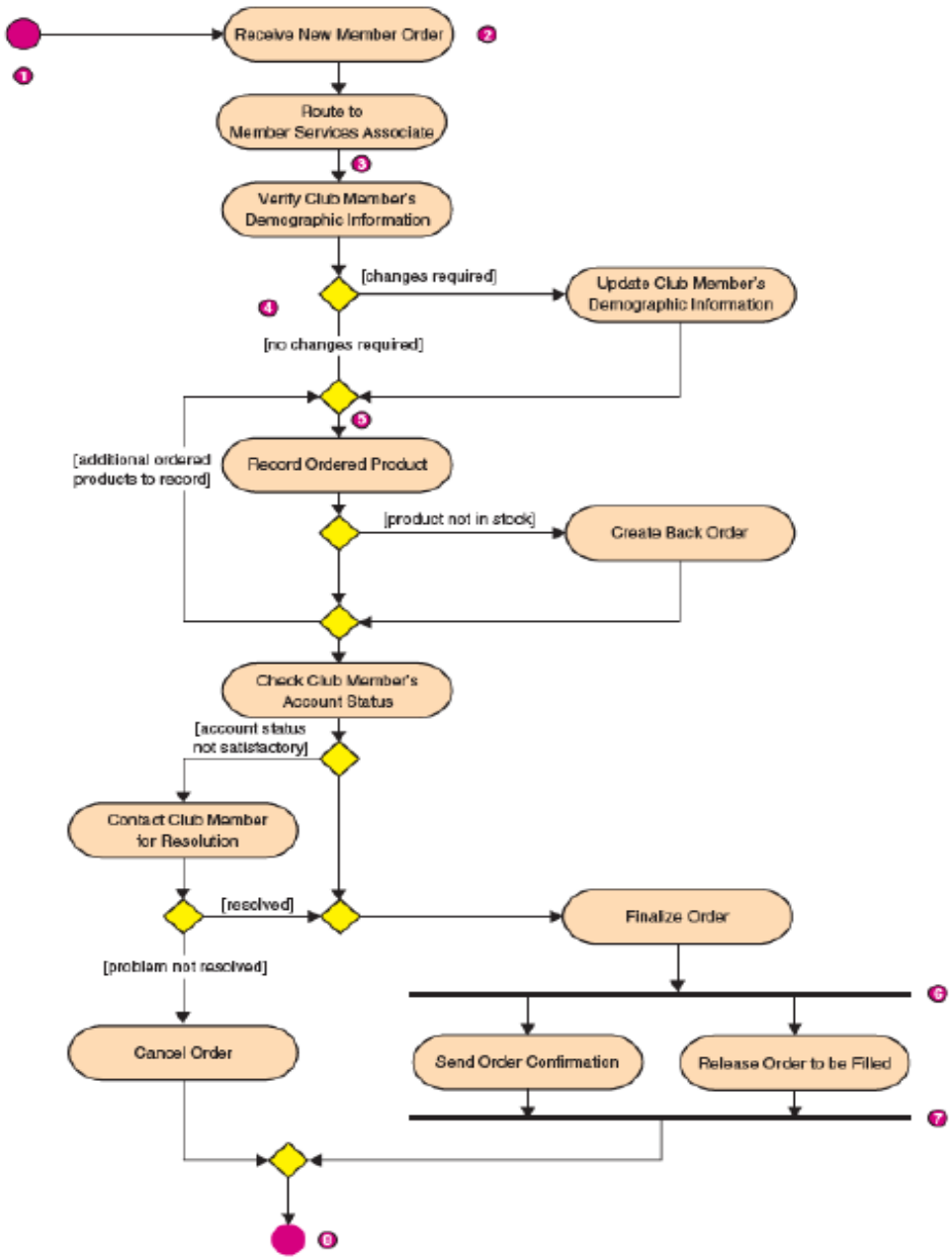
1. *Initial Node*, bentuk lingkaran berisi penuh melambangkan awal dari suatu proses.
2. *Actions*, bentuk persegi panjang yang mempunyai ujung lingkaran yang melambangkan tahap-tahap per individu. *Sequence* dari *actions* menunjukan total dari aktivitas yang dilihat dari diagram.
3. *Flows*, panah pada diagram mengindikasikan kemajuan dari sebuah actions. Kebanyakan *flow* tidak membutuhkan kata untuk mengidentifikasikan mereka kecuali kata tersebut keluar dari *decision*.

4. *Decision*, bentuk berlian dengan satu *flow* masuk dan dua atau lebih *flow* keluar. *Flow* keluar menandakan untuk indikasi sebuah kondisi.
5. *Merge*, bentuk berlian dengan dua atau lebih *flow* masuk dan satu *flow* keluar. Merupakan penggabungan *flow* yang sebelumnya dipisahkan oleh *decision*.
6. *Fork*, bar hitam dengan satu *flow* yang masuk beserta dua atau lebih *flow* yang keluar. *Actions* dengan *flow pararel* di bawah *fork* dapat terjadi dengan adanya urutan secara bersamaan.
7. *Join*, bar hitam dengan dua atau lebih *flow* yang masuk beserta satu *flow* yang keluar, tercatat pada akhir dari proses secara bersamaan. Semua *actions* yang menuju *join* harus lengkap sebelum proses dapat berlanjut.
8. *Activity Final*, lingkaran solid di dalam lingkaran berongga yang menandakan akhir dari proses

Selain komponen – komponen di atas, terdapat dua tambahan komponen dari activity diagram, yaitu :

1. *Subactivity Indicator*, simbol seperti sisir terbalik yang berada pada *actions* mengindikasikan bahwa *actions* telah keluar menuju *activity diagram* yang lain. Hal ini dapat membantu *activity diagram* agar tidak menjadi terlalu kompleks.
2. *Connector*, huruf yang berada di dalam lingkaran dan memberikan alat untuk mengarut kompleksitas. *Flow* yang menuju *connector* melompati *flow* yang keluar dari *connector* dengan huruf yang sama.

[Whitten & Bentley, 2007: 382-392]

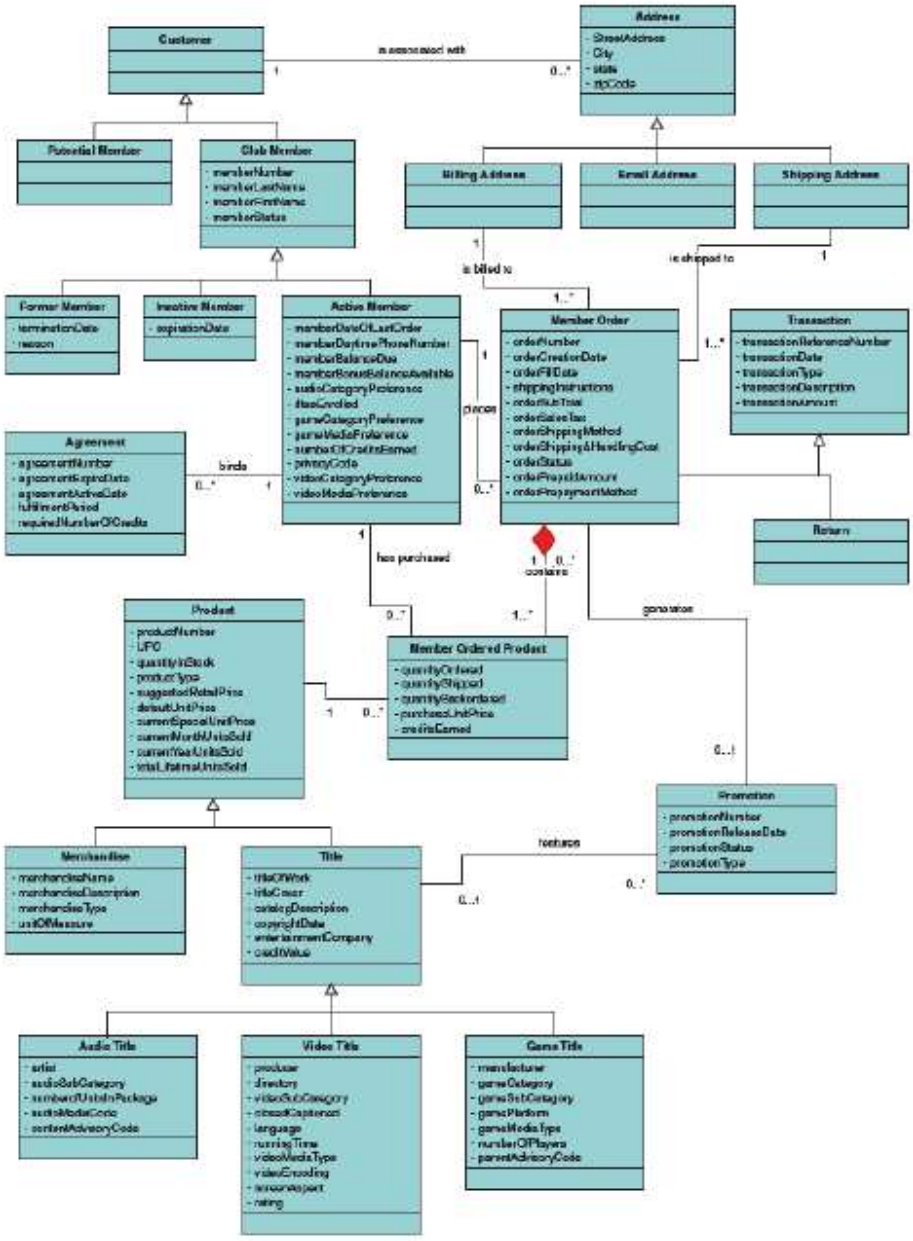


Gambar 2.7 Contoh *Activity Diagram*

(Whitten & Bentley, 2007: 392)

2.5.3 *Class Diagram*

Class Diagram adalah sebuah diagram yang memberikan gambaran grafis dari sistem struktur object statis, menunjukkan kelas objek bahwa sistem tersebut tersusun atas hubungan – hubungan antara kelas object.



Gambar 2.8 Contoh *Class Diagram*

(Whitten & Bentley, 2007: 406)

Dalam penggunaannya, *class* diagram memiliki beberapa istilah, yaitu:

1. *Visibility*

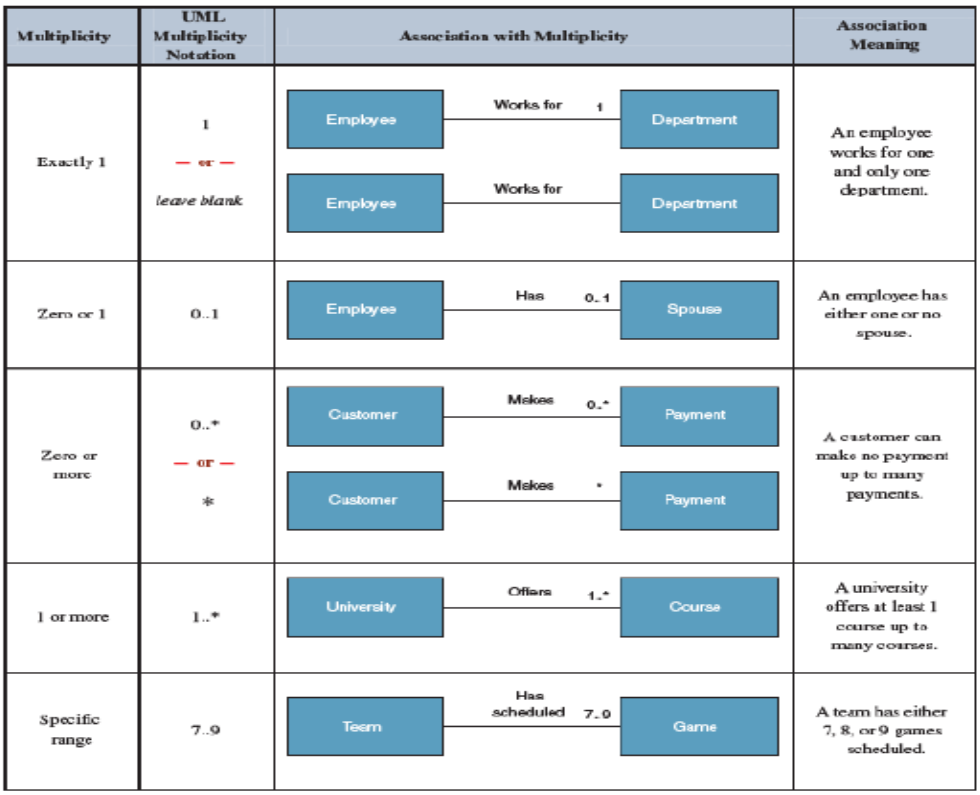
   Fungsi dari *visibility* dalam *class diagram* adalah untuk menentukan apakah atribut atau operasi dari suatu kelas dapat digunakan oleh kelas lain

Tabel 2.1 Penjelasan *Visibility*

Visibility	Simbol	Keterangan
Private	-	Hanya dapat digunakan oleh kelas yang mendefinisikan
Protected	#	Dapat digunakan oleh kelas yang mendefisinikan dan turunannya
Public	+	Dapat digunakan oleh kelas yang berhubungan

2. *Multiplicity and Associations*

   Fungsi dari *multiplicity* dalam *class diagram* adalah untuk menentukan banyaknya kelas yang berhubungan dengan kelas yang dimaksud.

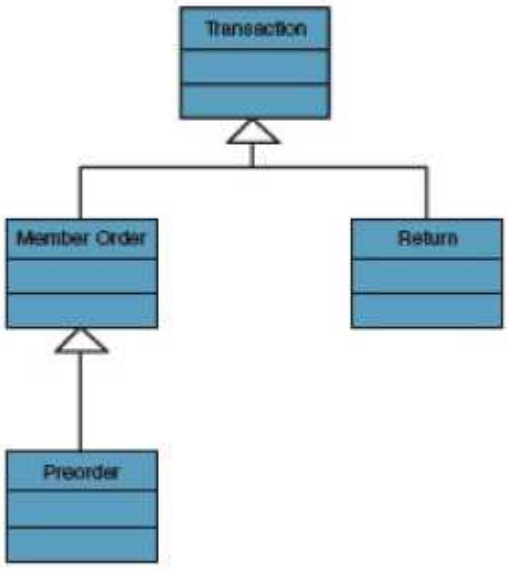


Gambar 2.9 Contoh *Multiplicity* dan *Associations*
(Whitten & Bentley, 2007: 377)

3. *Generalization*

   Fungsi dari *generalization* pada *class diagram* adalah untuk menggambarkan hubungan antara *superclass* dan *subclass. Superclass*

adalah bentuk umum dari *subclass, subclass* adalah bentuk spesifik dari *superclass*



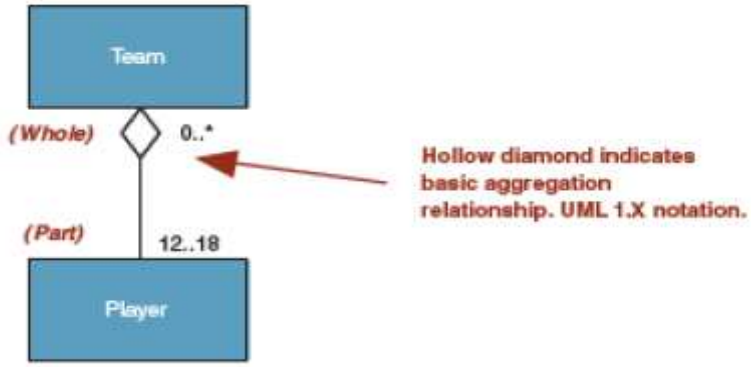
Gambar 2.10

Contoh *Generalization*

(Whitten & Bentley, 2007: 404)

4. *Aggregration*

Fungsi dari *aggregation* adalah untuk menggambarkan hubungan dimana satu kelas merupakan bagian dari kelas lain. Dalam agregasi tidak menggambarkan sebuh *inheritance*, tetapi bersifat *asimetris.* Misalnya terdapat dua buah kelas team dan play, kelas player merupakan bagian kelas dari team, tetapi team bukan bagian kelas dari player.



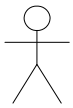
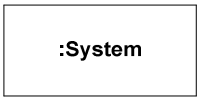
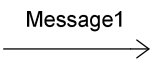
Gambar 2.11 Contoh *Agregration*

(Whitten & Bentley, 2007: 379)

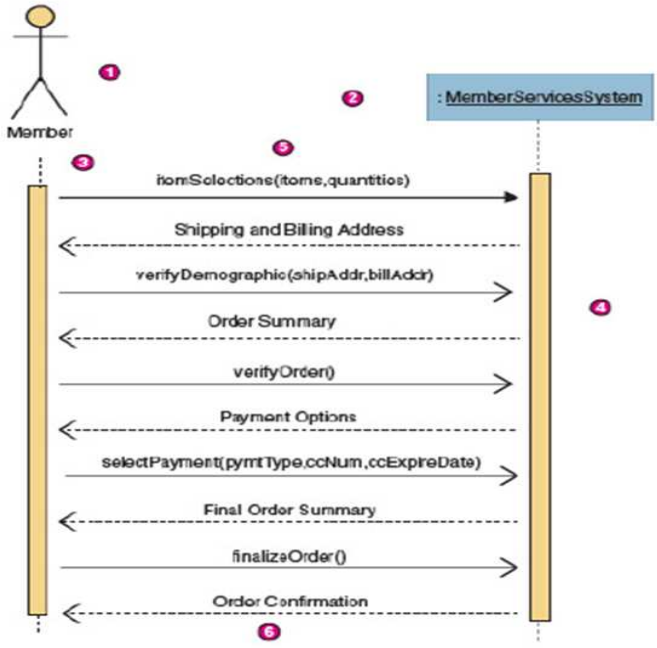
2.5.4 *Sequence Diagram*

Sequence Diagram adalah sebuah diagram yang menggambarkan interaksi antara *actor* dengan sistem dalam skenario *use case* yang sedang berlangsung. Diagram ini menggambarkan bagaimana pesan dikirim dan diterima antar objek dan urutannya. Komponen-komponen penting dalam *sequence diagram*, yaitu:

Gambar 2.12 Tabel Notasi *Sequence Diagram*

Notasi	**Keterangan**
 *Actor*	Notasi ini menggambarkan *user* yang berinteraksi dengan system
 *System*	Notasi ini menggambarkan kelas-kelas yang ada pada *class diagram*
 Lifelines	Notasi ini menggambarkan hidupnya objek dalam sebuah *sequence*
 *Activation Bars*	Notasi ini menggambarkan periode waktu saat proses aktif dalam interaksi
 *Input Message*	Notasi ini menggambarkan pesan masuk yang dikirimkan yaitu berupa *behavior.*
\	Notasi ini menggambarkan pesan yang dikirimkan sebagai balasan pesan masuk yaitu berupa

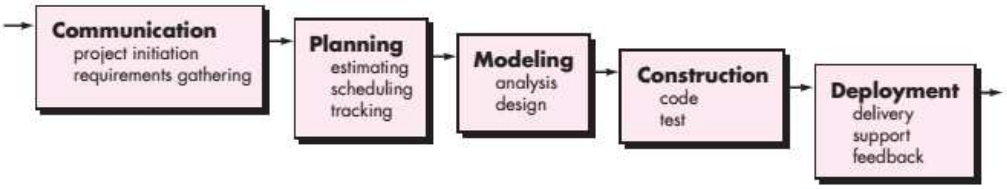
Message2 *Output Message*	*attribute*.
Loop *Frame*	Notasi ini menggambarkan area pada sistem yang mengalami perulangan (*loops),* seleksi (*alternate fragments),* atau kondisi opsional (*optional).*



Gambar 2.13 Contoh *Sequence Diagram*
(Whitten & Bentley,2007:396)

2.6 Waterfall Model

Model *waterfall* adalah model yang bersifat sistematis dan *sekuensial* dalam pengembangan perangkat lunak yang melalui tahapan *communication, planning, modelling, construction,* dan *deployment.* Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan secara berurutan. Tahapan pada model *waterfall* dapat dilihat pada gambar 2.14.



Gambar 2.14 Model *Waterfall*

(Pressman, 2010: 39)

Berikut penjelasan dari tahapan-tahapan tersebut:

1. *Communication*

   Tahapan ini terdiri dari *project initiation* dan *requirements gathering,* yaitu merupakan tahap pengumpulan informasi dan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software.*

2. *Planning*

   Tahapan ini terdiri dari *estimating, scheduling* dan *tracking.* Pada tahap ini pengembang *software* membuat perkiraan yang diperlukan serta penjadwalan agar pembuatan *software* dapat selesai sesuai dengan waktu yang diperkirakan.

3. *Modelling*

   Tahapan ini terdiri dari *analysis* dan *design.* Proses ini bertujuan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi s*oftware* sebelum proses penulisan kode (*coding*) dimulai.

4. *Construction*

   Terdiri dari proses *coding* dan pengujian *software,* pada tahap ini *design software* diterjemahkan menjadi bahasa yang dimengerti oleh mesin. Lalu semua fungsi *software* diuji coba agar *software* bebas dari *error* dan hasilnya sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

5. *Deployment*

   Terdiri dari *delivery, support* dan *feedback.* Pada tahap ini *software* digunakan langsung oleh *customer*. Pengembang juga menyediakan dokumentasi untuk semua fitur dan fungsi, dan pengembang mendapatkan umpan balik terhadap *software* untuk kepentingan modifikasi fitur dan fungsi.

[Pressman, 2010: 39]

2.7 Delapan Aturan Emas

Menurut Shneiderman & Plaisant (2010), terdapat delapan hal yang harus diperhatikan dalam membuat rancangan antar muka sebuah aplikasi atau yang sering disebut delapan aturan emas. Delapan aturan emas tersebut adalah :

a. Berusaha untuk konsisten

   Dalam perancangan *interface*, diperlukan konsistensi pada setiap aksi yang dilakukan. Contohnya seperti tampilan navigasi pada sebuah website, ketika berpindah ke halaman lain tetap sama (*font*, *color*, *layout*, dan apapun yang digunakan didalamnya).

b. Menyediakan fungsi yang bersifat umum

   Adanya jenis *User* yang beragam dari yang baru mengenal komputer hingga yang sudah ahli dengan komputer, dibutuhkan sebuah rancangan yang memiliki fungsi-fungsi yang mudah dikenali *User* yang beragam atau penjelasan pemakaian aplikasi juga memiliki fungsi tambahan yang mendukung aplikasi tersebut untuk para ahlinya.

c. Memberikan umpan balik yang informatif

   Untuk segala aksi yang dilakukan *user*, harus ada sistem yang memberikan umpan balik dengan respon yang berbeda di setiap kondisi yang ada.

d. Merancang dialog untuk menghasilkan penutupan

   Urutan aksi dikelompokan kedalam tiga kategori, yaitu aksi permulaan, pertengahan dan penutup. Pada saat *user* melakukan aksi penutup suatu proses, kotak dialog seharusnya meuncul untuk menegaskan bahwa *user* telah selesai melakukan proses tersebut.

e. Memberikan pencegahan terhadap kesalahan yang sederhana

   Sedapat mungkin sistem dirancang sehingga user tidak dapat melakukan kesalahan fatal. Jika kesalahan terjadi sistem dapat mendeteksi kesalahan dengan cepat dan memberikan mekanisme yang sederhana dan mudah dipahami untuk penanganan kesalahan.

f. Memungkinkan pengembalian aksi sebelumnya

   Sistem sebaiknya menyediakan fitur yang memungkinkan user untuk membatalkan aksi yang sudah dilakukannya dengan mudah. Fitur ini dapat mengurangi kecemasan karena *user* mengetahui bahwa kesalahan dapat

dibatalkan dan dapat mendorong user untuk mempelajari pilihan-pilihan yang tidak biasa.

g. Mendukung pengendalian internal

Menjadikan user sebagai pemegang kendali dari sistem bukan sistem yang mengendalikan *user*. Ketidak mampuan user dalam memperoleh informasi penting akan memberikan ketidak puasan kepada *user*.

h. Mengurangi beban ingatan jangka pendek

Keterbatasan ingatan manusia untuk mengolah informasi menyebabkan dibutuhkan rancangan tampilan yang sederhana agar informasi mudah dicerna. Desainer harus menghindari *interface* dimana *user* harus mengingat informasi dari satu tampilan yang akan dipakai di tampilan lainnya karena kapasitas ingatan manusia dalam hal merespon informasi jangka pendek terbatas.

[Shneiderman & Plaisant,2010:88-89]

2.8 PHP

PHP adalah bahasa pemrograman *open source* yang menempel pada HTML yang didukung oleh banyak *web servers* termasuk *Apache HTTP Server* dan *Microsoft's Internet Information Server*, dan pilihan bahasa pemrograman Linux *Web*. Salah satu keuntungan dari PHP adalah dapat diperpanjang, dan nomor modul perpanjangnya sudah menyediakan dukungan seperti koneksi *basis data*, *mail*, dan XML (Connolly dan Begg, 2005:1014).

2.9 SQL

SQL adalah sebuah bahasa komputer yang digunakan untuk mengelola dan berinteraksi dengan data dalam database relasional. SQL adalah bahasa yang paling universal database diimplementasikan digunakan, dan telah menjadi bahasa standar untuk manajemen database. SQL bekerja sama dengan RDBMS untuk menentukan struktur database, menyimpan data, mengontrol akses ke data, dan menjamin integrasi data. Meskipun bahasa lain telah dikembangkan untuk mengimplementasikan model relasional, SQL telah muncul sebagai pemenang (Sheldon, Moes, 2005;10).

2.10 Lima Faktor Manusia Terukur

Lima faktor manusia terukur yang dapat mendukung perancangan *interface*, yaitu (Shneiderman & Plaisant, 2010:32) :

1. *Time to Learn* (Waktu belajar)

   Waktu yang dibutuhkan oleh *user* untuk dapat memahami sistem.

2. *Speed of Performance* (Kecepatan kinerja)

   Kecepatan *user* dalam menjalankan aksi-aksi dalam sistem.

3. *Rate of Error* (Tingkat Kesalahan)

   Tingkat banyaknya kesalahan dan jenis kesalahan apa yang dilakukan oleh *user*. Pengendalian kesalahan merupakan komponen kritis dalam pembuatan *interface*.

4. *Retention Over Time* (Daya ingat)

   Kemampuan *user* untuk mengingat informasi yang diterima dari sistem dalam jangka waktu tertentu.

5. *Subjective Satisfaction* (Kepuasan subjektif)

   Keberhasilan suatu sistem yang dirancang, diukur melalui kepuasan *user* terhadap sistem tersebut.