ACC Workshop

Leonardo Valentino Kosasih

2024-04-06

Daftar Isi

1 Pengenalan Asuransi

Dalam kegiatan jual beli, terdapat suatu persamaan yang digunakan, yaitu

$$\text{Harga} := \text{Biaya} + \text{Keuntungan} \tag{1}$$

Sehingga dalam menentukan harga suatu barang/jasa, dapat dihitung dengan menjumlahkan biaya yang diperlukan untuk menyediakan barang/jasa tersebut dan keuntungan yang ingin diperoleh oleh penjual. Hal ini juga berlaku untuk produk asuransi dengan adanya suatu perbedaan mendasar. Perbedaan tersebut adalah, **tidak diketahuinya** biaya yang perlu dikeluarkan di masa depan.

Pada industri asuransi, **harga** pada persamaan (1) adalah premi dari suatu produk asuransi, **biaya** pada persamaan (1) adalah penjumlahan antara kerugian/klaim yang dilakukan oleh pemegang polis, dan juga pengeluaran perusahaan asuransi dalam memasarkan polis dan juga dalam menyelesaikan klaim yang diajukan dan keuntungan adalah arus kas net yang sering disebut dengan *underwriting profit*. Sehingga persamaan (1) dapat ditulis menjadi:

$$\text{Premi} := \text{Kerugian/Klaim} + \text{Biaya Operasional} + \textit{UW Profit} \tag{2}$$

yang dikenal sebagai *Fundamental Insruance Equation*.

Dalam merancang suatu produk asuransi, asumsi yang digunakan dapat dibagi menjadi 2 garis besar, yaitu **asumsi ekonomi** dan **asumsi non-ekonomi (demografis)**. Hal ini diperlukan karena pada umumnya, kontrak asuransi memiliki jangka waktu yang panjang. Sehingga, dalam melakukan valuasi pada persamaan (2), akan dihitung Ekspektasi dari Nilai Sekarang (*Expected Present Value*) untuk Premi, dan juga Kerugian atau Klaim untuk tiap tahunnya. Sehingga, agar perusahaan asuransi memperoleh keuntungan, maka ekspektasi dari **Premi** yang diperoleh di masa depan, harus setidaknya sama atau lebih besar dari ekspektasi dari **Klaim** yang akan diajukan di masa depan.

Modul ini, akan membahas mengenai asumsi-asumsi yang akan digunakan dalam merancang produk asuransi. Produk asuransi yang akan dipelajari adalah:

1. Asuransi Konvensional
2. Asuransi Syariah/Takaful
3. *Catasthrope* (CAT) *Bond*

Asuransi Syariah atau **Takaful** adalah asuransi yang dikelola dengan prinsip **Syariah**. Takaful, memiliki suatu kumpulan dana (*pooled fund*) yang digunakan untuk membayarkan klaim yang disebut dengan **Tabarru**. Sedangkan perusahaan asuransi (atau **Operator**) akan mendapat upah (**Ujrah**) dan/atau kontrak bagi hasil dalam mengelola dana **Tabarru**. Dalam operasionalnya, jika terdapat defisit dari saldo dana Tabarru, maka operator harus menyediakan utang tak berbunga (**Qardh**) kepada pemegang polis. Namun, jika terdapat surplus dari saldo dana Tabarru, maka keuntungan tersebut akan dibagi antara operator dan pemegang polis, sesuai dengan persentase yang telah ditentukan sebelumnya.

Asuransi adalah salah satu kontrak yang diatur dan diawasi oleh lembaga pemerintahan. Di Indonesia, lembaga yang mengawasi produk asuransi adalah Otoritas Jasa Keuangan (OJK). Sehingga dalam merancang suatu produk asuransi, selain memperhatikan kondisi pasar, perlu juga dilakukan studi mengenai peraturan yang berlaku.

2 *Term Structure of Interest Rate*

2.1 Teori Dasar

Term Structure of Interest Rate merujuk nilai suku bunga yang bergantung dari jangka waktu investasi. Hal ini biasa disajikan dengan kurva imbal hasil (*yield curve*). Di Indonesia sendiri, terdapat suatu lembaga yang menerbitkan data imbal hasil ini, yaitu PHEI. Perlu diingat bahwa imbal hasil yang disajikan pada situs tersebut merupakan imbal hasil dari obligasi berkupon (*coupon paying bond*), sehingga imbal hasil yang ada

bukan merupakan imbal hasil efektif. Untuk dapat memperoleh imbal hasil efektif (atau yang dikenal dengan istilah *spot rate*), perlu dilakukan metode *bootstrapping*.
Definisikan *spot rate*, r_t, sebagai imbal hasil efektif dari uang yang diinvestasikan pada waktu 0 hingga waktu t, dan *yield rate*, y_t, sebagai imbal hasil dari obligasi dengan kupon tahunan. Dari pendefinisian ini, diperoleh bahwa $r_1 = y_1$ (Mengapa?), dan untuk menghitung r_k dapat diperoleh dari persamaan berikut:

$$1 = \sum_{i=1}^{k-1} \frac{y_k}{(1+s_i)^i} + \frac{1+y_k}{(1+r_k)^k} \tag{3}$$

Definisikan kembali *forward rate*, $f_{n,m}$, sebagai imbal hasil efektif dari uang yang diinvestasikan pada waktu n hingga waktu m, sehingga $f_{n,m}$ dapat dihitung dengan persamaan berikut:

$$(1+f_{n,m})^{m-n} = \frac{(1+r_m)^m}{(1+r_n)^n} \tag{4}$$

3 *Survival Model*

3.1 Notasi yang akan digunakan:

1. l_x: jumlah orang yang hidup pada usia-x;
2. d_x: banyak orang yang meninggal pada usia-x;
3. ${}_tp_x$: peluang orang dengan usia-x tetap hidup hingga usia $x+t$;
4. q_x: peluang orang usia x meninggal.

3.2 Pemodelan

Umumnya, model kesintasan dicatat dalam sebuah tabel yang disebut dengan tabel mortalita (*life table* atau dalam kasus khusus *service table*). Terdapat beberapa tabel mortalita yang dapat digunakan, salah satunya adalah Tabel Mortalita Indonesia IV.

Karena karakteristik dari suatu kontrak asuransi yang memiliki jangkauan jangka panjang, sangat mungkin untuk seorang pemegang polis untuk mengubah pikirannya untuk mencari polis dengan karakteristik yang berbeda. Sehingga perlu juga diasumsikan, peluang seseorang untuk membatalkan polis.

Dalam kasus ini, akan dicoba konstruksi suatu *service table* untuk sembarang usia, dengan peluang pembatalan polis sesuai dengan tabel berikut:

Umur Polis	Peluang
$<$ 1 tahun	20 %
1 - 2 tahun	5 %
$>$ 2 tahun	1 %

```
# Package
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)

# Algoritma
lapse_table <- data.frame(
              'year' = c(0,1,2),
              'prob' = c(0.2, 0.05, 0.01))

mort_table <- read.csv('TMI IV.csv')
mort_table <- mort_table %>%
  mutate('average' = 0.5*(Male+Female))
svc_table <- function(age, policy_age, coverage){
  # input:
  # age: usia pemegang polis
  # policy_age: usia polis
  # coverage: jangka waktu polis

  # output:
  # table:service table
  table_ <- data.frame(age_policyholder = seq(age, age + coverage),
                       policy_year = seq(policy_age, policy_age + coverage))
  table_$mort_rate <- mort_table %>%
    filter(Exact.Age >= age, Exact.Age <= age+coverage) %>%
    pull(average)
  table_$lapse_rate <- unlist(sapply(X = table_$policy_year,
                              function(x){
                                lapse_table[lapse_table$year == min(x,2), 2]
                                })
  )
  table_$survive <- cumsum(c(1,-table_$mort_rate - table_$lapse_rate))[1:nrow(table_)]
  table_$mort_ <- table_$mort_rate*table_$survive
  table_$lapse <- table_$lapse_rate*table_$survive
return(table_)
}
svc_table(20,0,10)
```

```
##    age_policyholder policy_year mort_rate lapse_rate  survive        mort_
## 1                20           0  0.000380       0.20 1.000000 0.0003800000
## 2                21           1  0.000385       0.05 0.799620 0.0003078537
## 3                22           2  0.000395       0.01 0.749235 0.0002959478
## 4                23           3  0.000405       0.01 0.738840 0.0002992302
## 5                24           4  0.000420       0.01 0.728435 0.0003059427
## 6                25           5  0.000450       0.01 0.718015 0.0003231067
## 7                26           6  0.000485       0.01 0.707565 0.0003431690
## 8                27           7  0.000530       0.01 0.697080 0.0003694524
## 9                28           8  0.000570       0.01 0.686550 0.0003913335
## 10               29           9  0.000610       0.01 0.675980 0.0004123478
## 11               30          10  0.000655       0.01 0.665370 0.0004358173
##         lapse
## 1  0.20000000
## 2  0.03998100
## 3  0.00749235
## 4  0.00738840
## 5  0.00728435
## 6  0.00718015
```

```
## 7  0.00707565
## 8  0.00697080
## 9  0.00686550
## 10 0.00675980
## 11 0.00665370
```

Tentu saja, pada pemodelannya, belum tentu usia seorang pemegang polis berupa bilangan bulat sehingga dapat dilakukan interpolasi. Beberapa metode interpolasi yang umum digunakan adalah:

1. Interpolasi Linear (*Uniform Distribution of Death*)
2. Interpolasi Eksponensial (*Constant Force of Mortality*)
3. Interpolasi Harmonic (*Hyperbolic / Balducci*)

4 *Gross Policy Values*

Gross Policy Values (GPV) adalah suatu formula untuk menentukan nilai cadangan premi dari sebuah polis. Notasikan L_t sebagai peubah acak yang menyatakan *present value of gross future loss* pada waktu ke-t yang dapat dihitung dengan formula berikut:

$$L_t = b_t + e_t - p_t \tag{5}$$

dengan

1. b_t: *Present value of future benefits* pada waktu ke-t
2. e_t: *Present value of future expense* pada waktu ke-t
3. p_t: *Present value of future gross premiums.* pada waktu ke-t

Definisikan kembali suatu peubah acak yang menyakatan nilai polis pada waktu ke-t sebagai ${}_tV$ yang dapat dihitung dengan formula berikut:

$${}_tV + EPV_{P_t} = EPV_{B_t} + EPV_{e_t} \tag{6}$$

yang dapat dihitung secara rekursif dengan persamaan berikut:

$${}_tV = \frac{\left(p_{x+t t+1}V + q_{x+t}\left(b_{t+1} + e_{t+1}^{(b)}\right)\right)}{(1+i_t)} - P_t + e_t^{(p)} \tag{7}$$

dengan:

1. P_t: premi pada waktu ke-t
2. $e_t^{(p)}/e_t^{(b)}$: *expense* yang berkaitan dengan premi/manfaat pada waktu ke-t
3. b_{t+1}: manfaat yang dibayarkan pada waktu ke-t

Perhatikan bahwa untuk $t > n$ dengan n adalah masa pertanggungan polis, diperoleh ${}_tV = 0$. Sehingga dalam menghitung GPV akan lebih mudah jika menggunakan formula rekursif.

Misalkan dengan contoh pada algoritma sebelumnya, terdapat pembayaran premi sebanyak 100 pada waktu ke 0 dan manfaat yang dibayarkan hanya jika pemegang polis meninggal dengan besaran 500 pada tiap waktu. Untuk penyederhanaan, misalkan pula suku bunga yang digunakan sebesar 5%

```
n <- 10
i = 0.05
table_cf <- svc_table(20,0,10)
table_cf <- table_cf %>%
  mutate(disc_factor = (1+i)^-table_cf$policy_year,
```

```r
         premium = c(1000, rep(0,nrow(table_cf)-1)),
         benefit = rep(500, nrow(table_cf))) %>%
  mutate(expected_premium = premium*disc_factor*survive,
         expected_benefit = benefit*disc_factor*mort_)
table_cf
```

```
##    age_policyholder policy_year mort_rate lapse_rate  survive        mort_
## 1                20           0  0.000380       0.20 1.000000 0.0003800000
## 2                21           1  0.000385       0.05 0.799620 0.0003078537
## 3                22           2  0.000395       0.01 0.749235 0.0002959478
## 4                23           3  0.000405       0.01 0.738840 0.0002992302
## 5                24           4  0.000420       0.01 0.728435 0.0003059427
## 6                25           5  0.000450       0.01 0.718015 0.0003231067
## 7                26           6  0.000485       0.01 0.707565 0.0003431690
## 8                27           7  0.000530       0.01 0.697080 0.0003694524
## 9                28           8  0.000570       0.01 0.686550 0.0003913335
## 10               29           9  0.000610       0.01 0.675980 0.0004123478
## 11               30          10  0.000655       0.01 0.665370 0.0004358173
##         lapse disc_factor premium benefit expected_premium expected_benefit
## 1  0.20000000   1.0000000    1000     500             1000        0.1900000
## 2  0.03998100   0.9523810       0     500                0        0.1465970
## 3  0.00749235   0.9070295       0     500                0        0.1342167
## 4  0.00738840   0.8638376       0     500                0        0.1292431
## 5  0.00728435   0.8227025       0     500                0        0.1258499
## 6  0.00718015   0.7835262       0     500                0        0.1265813
## 7  0.00707565   0.7462154       0     500                0        0.1280390
## 8  0.00697080   0.7106813       0     500                0        0.1312815
## 9  0.00686550   0.6768394       0     500                0        0.1324350
## 10 0.00675980   0.6446089       0     500                0        0.1329015
## 11 0.00665370   0.6139133       0     500                0        0.1337770
```

```r
policy_value <- function(t){
  if (t > nrow(table_cf)){
    return(0)
  }
  return((ifelse(t=nrow(table_cf),
                 1,
                 table_cf$survive[(t+1)])*policy_value(t+1)+
            table_cf$expected_benefit[t])/(1+i))
}
table_cf$gpv <- unlist(lapply(1:nrow(table_cf), policy_value))
table_cf
```

```
##    age_policyholder policy_year mort_rate lapse_rate  survive        mort_
## 1                20           0  0.000380       0.20 1.000000 0.0003800000
## 2                21           1  0.000385       0.05 0.799620 0.0003078537
## 3                22           2  0.000395       0.01 0.749235 0.0002959478
## 4                23           3  0.000405       0.01 0.738840 0.0002992302
## 5                24           4  0.000420       0.01 0.728435 0.0003059427
## 6                25           5  0.000450       0.01 0.718015 0.0003231067
## 7                26           6  0.000485       0.01 0.707565 0.0003431690
## 8                27           7  0.000530       0.01 0.697080 0.0003694524
## 9                28           8  0.000570       0.01 0.686550 0.0003913335
## 10               29           9  0.000610       0.01 0.675980 0.0004123478
```

```
## 11              30          10  0.000655       0.01 0.665370 0.0004358173
##         lapse disc_factor premium benefit expected_premium expected_benefit
## 1  0.20000000   1.0000000    1000     500             1000        0.1900000
## 2  0.03998100   0.9523810       0     500                0        0.1465970
## 3  0.00749235   0.9070295       0     500                0        0.1342167
## 4  0.00738840   0.8638376       0     500                0        0.1292431
## 5  0.00728435   0.8227025       0     500                0        0.1258499
## 6  0.00718015   0.7835262       0     500                0        0.1265813
## 7  0.00707565   0.7462154       0     500                0        0.1280390
## 8  0.00697080   0.7106813       0     500                0        0.1312815
## 9  0.00686550   0.6768394       0     500                0        0.1324350
## 10 0.00675980   0.6446089       0     500                0        0.1329015
## 11 0.00665370   0.6139133       0     500                0        0.1337770
##          gpv
## 1  1.1542807
## 2  1.0219947
## 3  0.9264975
## 4  0.8386056
## 5  0.7512928
## 6  0.6630075
## 7  0.5695766
## 8  0.4700164
## 9  0.3622358
## 10 0.2479126
## 11 0.1274067
```