Aplikasi Kombinatorial Untuk Meningkatkan Keuntungan Dalam Kehidupan Sehari-Hari dan Berbisnis

Ziyad Dhia Rafi - 13520064
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*13520064@mahasiswa.itb.ac.id*

***Abstrak*—Makalah ini berisi tentang aplikasi teori kombinatorial untuk meningkatkan keuntungan dan mengurangi risiko kerugian dalam mengambil keputusan. Masalah-masalah yang dibahas berupa persoalan dasar yaitu bertaruh dalam *coin toss* yang mendasari perhitungan yang lebih kompleks yaitu penjualan tiket pesawat dengan Teknik *overbooking* untuk meningkatkan keuntungan maskapai. Tersedia perhitungan serta program untuk menghitung segala kombinasi yang ada.**

***Kata kunci*—Kombinasi, Permutasi, Bisnis, Risiko, Keuntungan, Kerugian, Tiket, Pesawat**

I. PENDAHULUAN

Segala sesuatu yang kita tempuh dalam dunia ini pasti memiliki risiko. Setiap keputusan yang diambil pasti memiliki risikonya masing-masing. Ada kalanya kpeutusan yang diambil memberikan sebuah hasil yang menguntungkan, atau bisa juga memberikan hasil yang merugikan.

Segala sesuatunya tampak terjadi secara acak. Semua terlihat memiliki kemungkinan yang sama untuk menghasilkan untung atau rugi. Hal ini membuat seseorang berpikir bahwa hasil akhir yang didapat hanya bergantung terhadap keberuntungan.

Namun faktanya, keputusan yang diambil di dunia ini dapat kita hitung risikonya menggunakan prinsip kombinatorial dan peluang. Banyak hal yang terlihat memiliki kemungkinan yang sama untuk mendapat keuntungan atau kerugian ternyata sebenarnya tidak demikian. Sebagai contoh, sebuah kasino di Las Vegas menghasilkan milyaran US Dollar dalam setahun. Hal ini terjadi karena keuntungan yang terjadi dalam setiap permainan di kasino selalu mengarah kepada pihak kasino. Kalau bukan karena hal tersebut mungkin tidak akan berdiri kasino-kasino megah seperti di Las Vegas atau Macau.

Gambar 1.1 Kasino di Las Vegas
https://idngamer.org/wp-content/uploads/2021/01/slot2.jpg

Dalam hal berbisnis juga tentu saja prinsip utamanya adalah mencari keuntungan sebanyak-banyaknya dan meminimalisasi risiko kerugian. Dengan menggunakan prinsip kombinatorial akan memudahkan untuk menghitung kemungkinan diapatkannya keuntungan atau kerugian.

Dalam makalah ini dibahas perhitungan berbagai perhitungan kombinatorial dan peluang dalam kehidupan sehari-hari dan berbisnis. Dimulai dari persoalan sederhana seperti dalam *coin toss*, hingga persoalan dalam dunia bisnis yaitu penjualan tiket pesawat. Dari perhitungan tersebut akan didapat keputusan terbaik yang perlu diambil sehingga keuntungan selalu mengarah kepada kita. Perhitungan juga akan dibuktikan melalui program komputer untuk mempermudah komputasi.

II. TEORI DASAR

A. Kombinatorial

Kombinatorial adalah sebuah cabang ilmu matematika untuk menghitung jumlah penyusunan objek-objek tanpa harus mengenumerasi semua kemungkinan susunannya.

- Kaidah dasar menghitung kombinatorial
  1. Kaidah perkalian (*rule of product*)
     Percobaan 1: p hasil
     Percobaan 2: q hasil
     Percobaan 1 **dan** Percobaan 2: p × q hasil

2. Kaidah penjumlahan (*rule of sum*)
   Percobaan 1: p hasil
   Percobaan 2: q hasil
   Percobaan 1 **atau** Percobaan 2: p × q hasil

- Perluasan kaidah dasar menghitung
  Misalkan ada n percobaan, masing-masing dengan p_i hasil:
  Kaidah perkalian (*rule of product*)
  $p_1 \times p_2 \times \ldots \times p_n$ *hasil*

  Kaidah penjumlahan (*rule of sum*)
  $p_1 + p_2 + \ldots + p_n$ *hasil*

- Permutasi
  Permutasi dalah jumlah urutan berbeda dari pengaturan objek-objek
  Nilai permutasi r dari n elemen dapat dihitung sebagai berikut:
  $$P(n,r) = \frac{n!}{(n-r)!}$$

- Kombinasi
  Kombinasi adalah bentuk khusus dari permutasi. Pada kombinasi urutan penempatan tidak berpengaruh.
  Nilai kombinasi r dari n elemen dapat dihitung sebagai berikut:
  $$C(n,r) = \frac{n!}{r!\,(n-r)!}$$

- Permutasi dengan unsur sama
  Merupakan salah satu bentuk khusus permutasi. Pada permutasi dengan unsur sama, perbedaan urutan pada unsur yang sama tidak berpengaruh
  Nilai permutasi unsur sama r dari n elemen, dengan jumlah elemen ke-1 = a1, ke-2 = a2, …, ke-n=an dapat dihitung sebagai berikut:
  $$Px(n,r) = \frac{P(n,r)}{a1!\,a2!\,\ldots an!}$$

B. Peluang

Peluang adalah kebolehjadian atau besar kemungkinan suatu kasus akan terjadi. Peluang memiliki jangkauan antara 0 hingga 1. Peluang bernilai 0 menyatakan kemustahilan sedangkan peluang bernilai 1 menyatakan sebuah kepastian.

Misalkan banyak terjadinya kondisi A adalah n(A), banyak ruang sampel adalah n(S), maka besar peluang P(A) dapat dihitung sebagai berikut:

Peluang dapat dinyatakan sebagai berikut:

$$P(A) = \frac{n(A)}{n(S)}$$

III. Perhitungan Kombinatorial Dalam Persoalan Di Dunia Nyata

A. Bertaruh dalam *Coin Toss*

Gambar 3.1 Permainan *Coin Toss*, Koin memiliki sisi *head* dan *tail*
https://i.stack.imgur.com/vuIvO.jpg

Sebuah koin memiliki dua buah sisi yaitu *head* dan *tail* atau dalam mata uang di Indonesia adalah sisi uang dan sisi gambar. Kedua sisi ini memiliki kemungkinan kemunculan yang sama yaitu 50% *head* dan 50% *tail*.

Dalam pertaruhan dengan *coin toss* aturannya sederhana, Seorang petaruh akan mendapatkan sejumlah uang apabila koin yang jatuh menghadap sisi yang dipilihnya, sedangkan petaruh akan kehilangan sejumlah uang apabila koin yang jatuh menghadap sisi yang berlawanan dengan yang dipilihnya. Permainan dapat diulang hingga berkali-kali *flips*.

Misalan seseorang ditawari untuk bertaruh dengan *coin toss*, dia akan mendapat 10 Dollar jika menang dan akan kehilangan 10 Dollar. Mungkin sebagian besar orang akan menolak untuk bermain karena seseorang cenderung untuk melihat kasus terburuk yaitu kehilangan uangnya.

Hal ini dapat dihitung dengan prinsip kombinatorial dan peluang sederhana

Kemunculan kasus menang:

$$P(1,1) = \frac{1!}{(1-1)!\,1!} = 1\ kasus$$

Kemunculan kasus kalah:

$P(1,1) = 1$ kasus

Total kasus adalah:

1 + 1 = 2 kasus

Sehingga peluangnya:

$$Peluang\ untung = Peluang\ rugi = \frac{Jumlah\ Kasus\ Khusus}{Total\ Kasus} = \frac{1}{2}$$

Bagaimana jika keuntungan menang dinaikkan menjadi 20 Dollar, sementara apabila kalah tetap 10 Dollar? Dalam satu kali *flips*, tanpa memedulikan jumlah uang yang didapat:

Total kasus untung: $P(1,1) = 1\ kasus$

Total kasus rugi: $P(1,1) = 1\ kasus$

Sehingga peluangnya menjadi:

Peluang untung = 50%

Peluang rugi = 50%

Terlihat kemungkinan untuk mendapatkan keuntungan masih sama dengan sebelumnya. Namun, jika dilakukan lebih banyak *flips* yang akan terjadi adalah sebagai berikut:

Dalam duua kali *flips*

Kasus 1: Menang, Menang (+40$)
Kasus 2: Menang, Kalah (+10$)
Kasus 3: Kalah, Menang (+10$)
Kasus 4: Kalah, Kalah (-20$)

Kemungkinan-kemungkinan kasus juga dapat dihitung dengan permutasi:

2 Menang (Permutasi unsur sama):

$$\frac{P(2,2)}{2!} = 1$$

1 Menang, 1 Kalah:

$$P(2,2) = 2$$

2 Kalah (Permutasi unsur sama):

$$\frac{P(2,2)}{2!} = 1$$

Hal ini menghasilkan:
Total kasus untung = 3 kasus
Total kasus rugi = 1 kasus
Peluang untung = 75%
Peluang rugi = 25%

4 kali *flips*:
4 Menang, 0 Kalah (Untung):

$$\frac{P(4,4)}{4!} = 1$$

3 Menang, 1 Kalah: (Untung)

$$\frac{P(4,4)}{3!} = 4$$

2 Menang, 2 Kalah (Untung)

$$\frac{P(4,4)}{2!\,2!} = 6$$

1 Menang, 3 Kalah: (Sama)

$$\frac{P(4,4)}{3!} = 4$$

0 Meanang, 4 Kalah: (Rugi)

$$\frac{P(4,4)}{4!} = 1$$

Total kasus untung = 11 Kasus
Total kasus rugi = 1 Kasus
Total kasus netral (tidak untung/rugi) = 4 Kasus
Peluang untung = 68,75%
Peluang rugi = 6,25%
Peluang netral = 25%
Pemain hanya perlu menang sekali agar tidak rugi

Kemungkinannya dapat dibuktikan dengan program komputer. Berikut adalah algoritma untuk menghitung semua kemungkinan yang menghasilkan kasus untung, rugi atau netral. Program lengkap dapat diakses di https://github.com/ziyaddr/PaperMatdis2021/blob/main/CoinBruteForce.py

```
# konfigurasi
cases = 2 # jumlah kemungkinan yang terjadi dalam satu proses
caseEffect = [-10, 20] # efek yang didapat caseEffect[0] = kasus-0(kalah); caseEffect[1] = kasus-1(menang)
flips = 4 # jumlah pengulangan
totalCases = cases**flips
countRugi = 0
countUntung = 0
a = [0 for i in range(flips)] # inisialisasi kasus pertama ([0, ..., 0])

start_time = time.time()
# proses iterasi
for x in range(totalCases):
    Money = 0
    for y in range(flips):
        Money += caseEffect[a[y]]
    if (Money < 0):
        countRugi += 1
    elif (Money > 0):
        countUntung += 1
    addOne(a, flips, cases) # next case

countSama = totalCases - countUntung - countRugi # jumlah kasus sama
pUntung = round(((countUntung*100)/totalCases), 2) # persen kasus untung
pRugi = round(((countRugi*100)/totalCases), 2) # presen kasus rugi
pSama = round(((countSama*100)/totalCases), 2) # persen kasus sama
```

Gambar 3.2 Program menghitung kemungkinan pada permainan *Coin Toss* secara *Brute Force*
Arsip penulis

Setiap kasus direpresentasikan sebagai bilangan berbasis 2, dengan setiap digit disimpan sebagai elemen array. Program melakukan iterasi untuk setiap kasus yang mungkin (*brute force*)

Pada flips = 4 menghasilkan:

```
Coin flip
Runtime =  0.0
Total kasus = 16
Kasus untung = 11 (68.75%)
Kasus rugi = 5 (31.25%)
Kasus sama = 0 (0.0%)
```

Gambar 3.3 Hasil perhitungan *coin toss* dalam 4 flips
Arsip penulis

Pada flips = 10 menghasilkan

```
Coin flip
Runtime =  0.0
Total kasus = 1024
Kasus untung = 848 (82.81%)
Kasus rugi = 176 (17.19%)
Kasus sama = 0 (0.0%)
```

Gambar 3.4 Hasil perhitungan *coin toss* dalam 10 flips
Arsip penulis

Pada flips = 20 menghasilkan

```
Coin flip
Runtime =  2.9
Total kasus = 1048576
Kasus untung = 988116 (94.23%)
Kasus rugi = 60460 (5.77%)
Kasus sama = 0 (0.0%)
```

Gambar 3.5 Hasil perhitungan *coin toss* dalam 20 flips
Arsip penulis

Karena program ini menggunakan algoritma $O(2^n)$ maka runtimenya akan meningkat secara eksponensial. Menggunakan algoritma dengan menghitung kombinasi tentu akan lebih cepat daripada melakukan iterasi satu per satu.

Dengan menggunakan algoritma permutasi, dapat dengan mudah didapatkan hasil bahkan hingga 1000 flips dalam kurang dari 0.1 detik. Program lengkap dapat diases di https://github.com/ziyaddr/PaperMatdis2021/blob/main/CoinSpecial.py

Pada flips = 100 menghasilkan

```
Coin flip
Runtime =  0.0
Total kasus = 1.268e+30
Kasus untung = 1.267e+30 (99.9563140082%)
Kasus rugi = 5.538e+26 (0.0436859918%)
Kasus sama = 0.000e+00 (0.0%)
```

Gambar 3.6 Hasil perhitungan *coin toss* dalam 100 flips
Arsip penulis

Pada flips = 200 mennghasilkan

```
Coin flip
Runtime =  0.0
Total kasus = 2.582e+120
Kasus untung = 2.582e+120 (99.999999999%)
Kasus rugi = 2.472e+109 (1e-09%)
Kasus sama = 0.000e+00 (0.0%)
```

Gambar 3.7 Hasil perhitungan *coin toss* dalam 200 flips
Arsip penulis

Dari data yang telah disajikan, dapat terlihat dengan jelas bahwa apabila keuntungan kemenangan ditingkatkan maka kemungkinan untuk untung setelah bermain akan semakin besar.

B. Penjualan Tiket Pesawat

Prinsip kombinatorial juga dipakai dalam penjualan tiket pesawat. Maskapai ingin mendapatkan profit sebesar-besarnya oleh karena itu berbagai cara dan perhitungan digunakan untuk mencapai profit maksimum.

Banyak maskapai memberlakukan promosi agar penumpang dapat membatalkan penerbangan dengan refund 100% tanpai biaya tambahan. Tentunya maskapai akan mengalami kerugian jika penumpang ada yang membatalkan penerbangan. Oleh karena itu, maskapai memberlakukan suatu Teknik yaitu *overbooking* atau menjual tiket melebihi kapasitas yang disediakan.

Apabila maskapai memberlakukan *overbooking*, dengan asumsi tiket terjual habis, kemungkinan yang terjadi akan terbagi menjadi 3 kasus. Ketiga kasus ini menggunakan prinsip untung atau rugi dibandingkan dengan tanpa *overbooking* seperti pada permainan *coin toss*.

Kasus yang pertama adalah jika penumpang tidak ada yang membatalkkan penerbangan. Pada kasus ini pihak maskapai harus membayar penalti setiap penumpang yang "ditendang" dari penerbangan. Tentunya pada kasus ini maskapai akan mendapatkan profit lebih sedikit dibanding menjual tiket sesuai kapasitas.

Kasus yang kedua adalah jika jumlah penumpang yang membatalkan penerbangan lebih banyak dibanding jumlah yang melebihi kapasitas. Pada kasus ini semua penumpang yang tidak membatalkan penerbangan dapat masuk semua ke dalam pesawat. Tentunya maskapai akan mendapat profit lebih jika melakukan *overbooking* karena lebih banyak penumpang yang masuk ke dalam pesawat.

Kasus yang ketiga adalah apabila sebagian penumpang membatalkan penerbangan tetapi tidak melebihi jumlah *overcapacity* sehingga masih ada penumpang yang dibatalkan penerbangannya. Pada kasus ini maskapai bisa saja mengalami keuntungan atau kerugian. Kasus inilah yang akan menjadi pembahasan utama untuk menyatakan apakah *overbookingI* efektif untuk meningkatkan keuntungan maskapai.

Pembahasan kali ini menggunakan berbagai asumsi. Tiket diasumsikan terjual habis karena jika tidak terjual habis maka tidak aka nada perbedaan antara melakukan *overbooking* atau tidak. Asumsi kedua adalah tidak ada denda jika penumpang melakukan pembatalan penerbangan. Kapasitas pesawat adalah 200 kursi dengan harga tiket yang sama yaitu 200$. Diasumsikan kompensasi yang didapatkan penumpang jika penerbangannya dibatalkan oleh maskapai adalah sebesar 100$ sehingga orang tersebut akan mendapatkan 200$ refund + 100$ kompensasi. Selanjutnya akan divariasikan jumlah penumpang yang membatalkan penerbangan.

Perhitungan pada masalah *overbooking* ini menggunakan prinsip yang sama dengan *coin toss* yang sebelumnya sudah di bahas, namun kali ini basis kasusnya adalah 20.

Perhitungan keuntungan atau kerugian dapat terlihat pada tabel sebagai berikut:

Harga tiket	Compensation	Capacity	Passenger Cancel	Tickets sold	Tickets income	Refund	Total income	Tickets sold	Tickets income	Cancelled by airline	Refund	Cancel Penalty	Total Income	Margin	Untung/Rugi
200	100	200	1	200	40000	200	39800	220	44000	19	4000	1900	38100	-1700	RUGI
200	100	200	2	200	40000	400	39600	220	44000	18	4000	1800	38200	-1400	RUGI
200	100	200	3	200	40000	600	39400	220	44000	17	4000	1700	38300	-1100	RUGI
200	100	200	4	200	40000	800	39200	220	44000	16	4000	1600	38400	-800	RUGI
200	100	200	5	200	40000	1000	39000	220	44000	15	4000	1500	38500	-500	RUGI
200	100	200	6	200	40000	1200	38800	220	44000	14	4000	1400	38600	-200	RUGI
200	100	200	7	200	40000	1400	38600	220	44000	13	4000	1300	38700	100	UNTUNG
200	100	200	8	200	40000	1600	38400	220	44000	12	4000	1200	38800	400	UNTUNG
200	100	200	9	200	40000	1800	38200	220	44000	11	4000	1100	38900	700	UNTUNG
200	100	200	10	200	40000	2000	38000	220	44000	10	4000	1000	39000	1000	UNTUNG
200	100	200	11	200	40000	2200	37800	220	44000	9	4000	900	39100	1300	UNTUNG
200	100	200	12	200	40000	2400	37600	220	44000	8	4000	800	39200	1600	UNTUNG
200	100	200	13	200	40000	2600	37400	220	44000	7	4000	700	39300	1900	UNTUNG
200	100	200	14	200	40000	2800	37200	220	44000	6	4000	600	39400	2200	UNTUNG
200	100	200	15	200	40000	3000	37000	220	44000	5	4000	500	39500	2500	UNTUNG
200	100	200	16	200	40000	3200	36800	220	44000	4	4000	400	39600	2800	UNTUNG
200	100	200	17	200	40000	3400	36600	220	44000	3	4000	300	39700	3100	UNTUNG
200	100	200	18	200	40000	3600	36400	220	44000	2	4000	200	39800	3400	UNTUNG
200	100	200	19	200	40000	3800	36200	220	44000	1	4000	100	39900	3700	UNTUNG
200	100	200	20	200	40000	4000	36000	220	44000	0	4000	0	40000	4000	UNTUNG

Tabel 3.1 Tabel hasil perhitungan perbandingan keuntungan dan kerugian *overbooking* dalam sekali penerbangan
Arsip penulis

Dalam satu penerbangan didapatkan hasil sebagai berikut:
Terdapat 20 kasus:
Kasus 1: -1700$ (rugi)
Kasus 2: -1400$ (rugi)
Kasus 3: -1100$ (rugi)
Kasus 4: -800$ (rugi)
Kasus 5: -500$ (rugi)
Kasus 6: -200$ (rugi)
Kasus 7: 100$ (untung)
Kasus 8: 400$ (untung)
Kasus 8: 700$ (untung)
Kasus 8: 1000$ (untung)
Kasus 8: 1300$ (untung)
Kasus 8: 1600$ (untung)
Kasus 8: 1900$ (untung)
Kasus 8: 2200$ (untung)
Kasus 8: 2500$ (untung)
Kasus 8: 2800$ (untung)
Kasus 8: 3100$ (untung)
Kasus 8: 3400$ (untung)
Kasus 8: 3700$ (untung)
Kasus 8: 4000$ (untung)

Dalam sekali penerbangan, terdapat 6 kasus rugi dan 14 kasus untung sehingga peluangnya menjadi:
Peluang untung: 70%
Peluang rugi: 30%

Apabila dilakukan n kali penerbangan maka peluangnya akan berbeda. Banyaknya kasus yang dapat terjadi adalah sebanyak 20^n buah kasus.

Perhitungan menggunakan komputer dapat menggunakan kombinatorial yang dioptimasikan untuk keperluan khusus seperti ini, namun kali ini akan digunakan algoritma *brute force*, yaitu melakukan iterasi sebanyak jumlah total kasus agar dapat berlaku pada asumsi apapun sama seperti permainan *coin toss*. Program lengkap dapat diakses di https://github.com/ziyaddr/PaperMatdis2021/blob/main/FlightBruteForce.py

```
print("Flight")
# konfigurasi
cases = 20 # jumlah kemungkinan yang terjadi dalam satu proses
caseEffect = [-1700, -1400, -1100, -800, -500, -200, 100, 400, 700, 1000,
1300, 1600, 1900, 2200, 2500, 2800, 3100, 3400, 3700, 4000]
# efek yang didapat caseEffect[0] = kasus-0, dst
flips = 10 # jumlah pengulangan
totalCases = cases**flips
countRugi = 0
countUntung = 0
a = [0 for i in range(flips)] # inisialisasi kasus pertama ([0, ..., 0])

start_time = time.time()
# proses iterasi
for x in range(totalCases):
    Money = 0
    for y in range(flips):
        Money += caseEffect[a[y]]
    if (Money < 0):
        countRugi += 1
    elif (Money > 0):
        countUntung += 1
    addOne(a, flips, cases) # next case

countSama = totalCases - countUntung - countRugi # jumlah kasus sama
pUntung = round(((countUntung*100)/totalCases), 2) # persen kasus untung
pRugi = round(((countRugi*100)/totalCases), 2) # presen kasus rugi
pSama = round(((countSama*100)/totalCases), 2) # persen kasus sama
```

Gambar 3.8 Program perhitungan keuntungan dan kerugian *overbooking* tiket pesawat
Arsip penulis

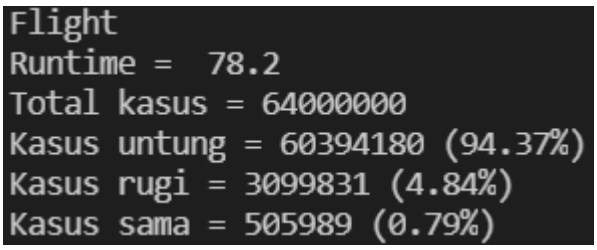
Setiap kasus dinyatakan sebagia bilangan berbasis 20 dan digit-digitnya disimpan sebagai elemen array.

Dalam 3 kali penerbangan akan didapat hasil sebagai berikut:

```
Flight
Runtime =  0.0
Total kasus = 8000
Kasus untung = 6860 (85.75%)
Kasus rugi = 969 (12.11%)
Kasus sama = 171 (2.14%)
```

Gambar 3.9 Hasil perhitungan *overbooking* dalam 3 penerbangan
Arsip penulis

Dalam 6 kali penerbangan akan didapat hasil sebagai berikut



Gambar 3.10 Hasil perhitungan *overbooking* dalam 6 kali penerbangan
Arsip penulis

Program seperti ini menggunakan algoritma $O(20^n)$ sehingga setipa peningkatan 1 penerbangan, program akan memakan waktu 20 kali lebih lama. Sehingga perhitungan dibatasi hingga 6 penerbangan.

Semakin banyak jumlah penerbangan, dengan melakukan *overbooking*, sama seperti permainan *coin toss*, peluang keuntungannya akan semakin menuju ke 100% yang artinya pasti untung. *Overbooking* yang dilakukan maskapai sukses untuk meningkatkan keuntungan sebuah maskapai. Dengan ribuan penerbangan yang dilakukan tentunya akan hampir pasti maskapai akan mendapatkan keuntungan lebih dengan melakukan overbooking.

Tentunya perhitungan seperti ini merupakan perhitungan yang sangat sederhana dan perhitungan yang dilakukan maskapai jauh lebih kompleks daripada itu. Dengan mempertimbangkan statistik seperti data persentase pembatalan penerbangan, melakukan *binomial probability* dsb, tentu hasil yang didapat akan jauh lebih akurat. Dengan hal itu tentunya dapat ditentukan jumlah tiket yang dijual, penalti pembatalan penerbangan, harga tiket dan banyak lainnya untuk mendapatkan keuntungan maksimum.

IV. KESIMPULAN

Dalam permainan *coin toss* serta penjualan tiket pesawat didapatkan hasil sebagia berikut:

1. Apabila terdapat k jenis kasus berbeda, serta n kali pengulangan maka jumlah kasus total adalah sebanyak k^n buah kasus
2. Apabila sebuah kondisi merupakan sesuatu yg menguntungkan dalam sekali pengulangan, maka semakin banyak pengulangan maka peluangnya keuntungannya akan menuju 100%

Dapat diambil kesimpulan, penerapan teori kombinatorial sangat berguna diterapkan dalam dunia bisnis seperti dalam penjualan tiket pesawat. Dengan memanfaatkan teori ini dapat tercapai prinsip utama bisnis yaitu mendapatkan keuntungan sebesar-besarnya dan meminimalisasi risiko kerugian. Menggunakan teori kombinatorial juga dapat mempersingkat perhitungan kasus-kasus tertentu, dalam hal ini kasus untung atau rugi dibanding dengan melakukan iterasi terhadap setiap kasus yang ada.

V. PENUTUP

Penulis mengucapkan rasa syukur kepada Allah SWT, karena dengan rahmat dan karunianya, penulis dapat menyelesaikan makalah ini dengan sebaik-baiknya

Penulis juga mengucapkan terimakasih yang sangat dalam kepada Orang Tua, Dosen, serta teman yang mendukung sepenuhnya atas ditulisnya makalah ini.

Penulis sadar bahwa mash banyak kekurangan yang dimiliki, oleh karena itu penulis sangat terbuka untuk kritik dan saran agar semakin baik kedepannya.

Demikian makalah ini dibuat oleh penulis. Semoga dengan adanya makalah ini dapat memberikan manfaat yang baik bagi perkembangan ilmu pengetahuan, khususnya di bidang Matematika Diskrit

REFERENSI

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Kombinatorial-2020-Bagian1.pdf
diakses pada 11 Desember 2021, pukul 21.03 WIB
[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Kombinatorial-2020-Bagian2.pdf
diakses pada 11 Desember 2021, pukul 21.03 WIB
[3] https://www.britannica.com/story/why-do-airlines-overbook-seats-on-flights
diakses pada 11 Desember 2021, pukul 23.30 WIB
[4] https://www.khanacademy.org/math/statistics-probability/probability-library/basic-theoretical-probability/a/probability-the-basics
diakses pada 12 Desember 2021, pukul 00.11 WIB
[5] https://stackoverflow.com/questions/54205990/generic-formula-for-build-an-array-of-numbers-of-base-n
diakses pada 13 Desember 2021, pukul 22.15 WIB)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 14 Desember 2021

Ziyad Dhia Rafi 13520064