Penerapan Program Dinamis pada Penentuan Olahraga Harian

Made Prisha Wulansari 13518049
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
made.prisha@gmail.com

***Abstrak*—Olahraga merupakan kegiatan yang sangat penting yang perlu dilakukan oleh semua orang untuk menjaga kesehatan tubuh. Terdapat batasan waktu yang dianjurkan dalam melakukan olahraga. Untuk mengoptimalkan jumlah kalori yang terbakar pada saat olahraga, penentuan olahraga harian dapat menjadi solusinya. Pada makalah ini akan dibahas mengenai penentuan olahraga harian dengan program dinamis.**

Kata kunci—olahraga, kalori, optimal, program dinamis, knapsack (0/1) problem, maksimum, tahap, status

I. PENDAHULUAN

Olahraga merupakan aktifitas yang melatih tubuh agar tetap sehat dan bugar. Olahraga wajib dilakukan setiap orang. Kemampuan olahraga setiap orang tentunya berbeda. Dengan melakukan olahraga rutin, tubuh akan semakin terlatih dalam melakukan olahraga. Olahraga memiliki banyak manfaat, diantaranya adalah meningkatkan daya tahan tubuh, meningkatkan daya pikir, mencegah penyakit jantung, menangkal obesitas, dan tentunya membakar kalori dalam tubuh.

Gambar 1. Macam-macam Olahraga

(sumber: https://www.inverse.com/mind-body/exercise-coronavirus-how-to-workout-safely-in-a-pandemic)

Terdapat bermacam-macam jenis olahraga yang dapat dilakukan, diantaranya yaitu bersepeda, berenang, bermain basket, berlari, angkat beban, lompat tali dan lainnya. Kalori yang terbakar dengan melakukan olahraga tersebut tentunya berbeda-beda. Jika dilakukan dengan tepat, olahraga yang dilakukan dapat membakar kalori dengan maksimal.

Diperlukan batasan waktu tertentu dalam berolahraga. Berolahraga tentunya tidak baik jika dilakukan secara berlebihan. Daya tahan seseorang dalam berolahraga tentunya berbeda-beda. Beberapa orang juga mungkin memiliki kesibukan lain sehingga waktu untuk melakukan olahraga terbatas.

Untuk memaksimalkan pembakaran kalori melalui berbagai macam olahraga yang dilakukan dalam setiap sesi olahraga, dapat dilakukan penghitungan pada kalori dengan batas waktu yang diinginkan. Dengan banyaknya pilihan olahraga yang ada, terdapat masalah dalam pencarian rangkaian olahraga yang paling optimal dalam membakar kalori. Akan memakan waktu yang lama jika seseorang menghitung satu persatu rangkaian olahraga yang mungkin dilakukan dalam rentang waktu tertentu. Dengan pemrograman dinamis, waktu untuk mencari rangkaian olahraga yang paling optimal untuk membakar kalori dapat dipersingkat.

II. DASAR TEORI

A. *Program Dinamis*

Program dinamis merupakan algoritma untuk memecahkan masalah dengan cara menguraikan masalah yang kompleks menjadi lebih kecil yang saling berkaitan. Dari masalah yang sudah diuraikan tersebut, diambil keputusan dari setiap masalah, kemudian dihasilkan solusi dari persoalan secara keseluruhan.

Program dinamis memiliki beberapa karakteristik dalam penyelesaian persoala diantaranya adalah:

1. Mungkin memiliki sejumlah pilihan penyelesaian yang berhingga
2. Solusi atau keputusan yang diambil pada setiap tahap berkaitan dan dibangun dari solusi pada tahapan sebelumnya
3. Menggunakan persyaratan oprimalisasi serta batasan dalam pilihan yang perlu dipertimbangkan pada suatu tahap

Algoritma ini banyak digunakan untuk persoalan optimasi. Persoalan optimasi merupakan persoalan yang memerlukan sebuah atau serangkaian proses yang dilakukan agar mendapatkan hasil yang paling ideal atau optimal, bisa jadi hasil yang terkecil atau terbesar. Contoh persoalan optimasi seperti penjadwalan, *knapsack problem*, *Travelling*

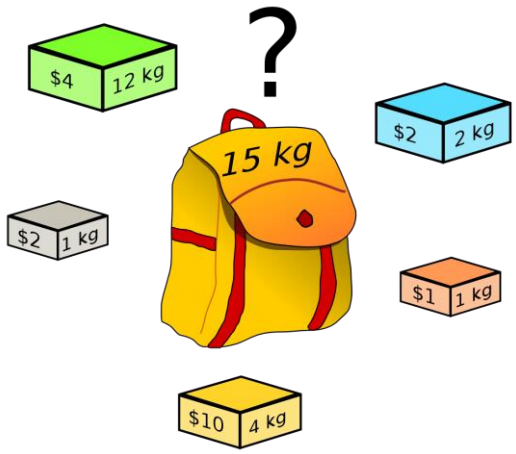
Salesperson Problem (TSP), dan banyak lainnya.Integer (0/1) Knapsack Problem

Rangkaian keputusan yang dibuat berdasarkan Prinsip Optimalitas karena mayoritas aplikasi program dinamis pada persoalan optimalitas. Prinsip optimalitas yaitu solusi optimal dari suatu persoalan optimalitas dibangun dari solusi optimal tiap tahapannya. Ini berarti bahwa untuk mendapatkan hasil optimal pada tahap k+1 menggunakan hasil optimal pada tahap k, atau ongkos pada tahap k+1 adalah ongkos pada tahap k ditambah dengan ongkos dari tahap k ke k+1.

Terdapat dua pendekatan program dinamis yaitu program dinamis maju (*forward* atau *up-down*) dan mundur (*backward* atau *bottom-up*). Pada program dinamis maju, tahapan dimulai dari tahap pertama hingga tahap n. Untuk menghitung ongkos pada tahap k+1, yaitu dengan menambahkan ongkos pada tahap k dengan ongkos dari tahap k ke k+1. Sedangkan pada program dinamis mundur, tahapan dimulai dari tahap n ke tahap 1. Untuk menghitung ongkos pada tahap k yaitu dengan menambahkan ongkos pada tahap k+1 dengan ongkos dari tahap k+1 ke k.

Ada beberapa langkah dalam mengembangkan algoritma ini. Pertama, perlunya dibuat karakteristik dari struktur solusi optimal. Kemudian, nilai solusi optimal didefinisikan secara rekursif. Lalu, solusi optimal dihitung sesuai dengan pendekatan program dinamis yang dipilih. Langkah terakhir yaitu melakukan konstruksi agar didapat solusi optimal.

B. Knapsack (0/1) Problem



Gambar 2. Persoalan Knapsack (0/1)

(sumber: https://en.wikipedia.org/wiki/Knapsack_problem)

Sesuai dengan namanya, persoalan ini seperti permasalahan dalam menentukan barang yang dapat dibawa dalam sebuah ransel. Setiap pilihan barang memiliki nilai dan berat masing-masing. Ransel juga memiliki kekuatan menahan beban dengan berat maksimal tertentu. Solusi yang dicari dari permasalahan ini yaitu barang-barang apa saja yang harus dipilih agar nilai barang pada ransel maksimal dan ransel tidak menahan beban lebih dari berat maksimal yang bisa ditahan. *Knapsack (0/1) Problem* merupakan persoalan optimasi kombinasi dari sejumlah objek, manakah objek yang harus dipilih dan tidak agar menghasilkan solusi yang optimal dengan tetap memenuhi syarat dari persoalan.

Persoalan ini dapat diselesaikan dengan berbagai algoritma. Pada algoritma *Exhaustive Search* persoalan ini diselesaikan dengan mengenumerasi semua kemungkinan yang terjadi, mengevaluasi nilai dari setiap kemungkinan, kemudian dipilih kemungkinan yang memberikan nilai paling optimal. Pada algoritma *Greedy*, ada tiga strategi dalam menyelesaikannya yaitu *Greedy* dengan keuntungan, *Greedy* dengan berat, dan *Greedy* dengan kepadatan atau menghitung keuntungan per unit berat. Dengan algoritma *Greedy*, objek ditambahkan ke penyelesaian berdasarkan strategi tersebut agar memaksimumkan keuntungan atau memasukkan sebanyak mungkin objek ke dalam *knapsack*. Tetapi tidak dijamin bahwa algoritma ini akan menghasilkan solusi yang paling optimal. Persoalan *knapsack* (0/1) ini juga dapat diselesaikan dengan algoritma program dinamis. Penyelesaian dengan algoritma ini yaitu dengan memasukkan objek ke knapsack dengan mempertimbangkan kapasitas sisa dan apakah lebih optimal jika sebuah objek dimasukkan atau tidak.

C. Kalori

Kalori masuk ke tubuh manusia dalam bentuk makanan. Makanan yang dikonsumsi masuk ke dalam tubuh dan diuraikan menjadi asam lemak. Kemudian, asam lemak akan masuk ke aliran darah dan diserap oleh usus halus.

Lemak terdapat pada jaringan tubuh. Jumlah sel dari lemak sendiri tidak bertambah ataupun berkuran, tetapi ukuran dari lemak ini yang dapat membesar dan mengecil. Jika asupan kalori yang dikonsumsi lebih sedikit dari biasanya, maka lemak akan mengecil. Tetapi jika kalori yang dikonsumsi lebih banyak, maka ukuran lemak menjadi lebih besar.

Olahraga dan aktifitas yang memacu pernafasan akan membuat ukuran sel lemak ini mengecil. Hal ini terjadi karena lemak keluar dari tubuh lebih banyak dalam bentuk molekul lewat pernapasan, sedangkan sisanya keluar dari cairan tubuh seperti urin, keringat, dan cairan tubuh lainnya.

III. Implementasi dan Pembahasan

Persoalan pemilihan olahraga agar kalori yang terbakar dapat maksimal pada rentang waktu tertentu sama seperti Knapsack (0/1) Problem. Pada persoalan ini, memiliki syarat batas waktu dalam melakukan olahraga. Pada persoalan ini juga mencari kalori optimal yang terbakar dari rangkaian pilihan olahraga. Algoritma yang digunakan dalam penyelesaian persoalan ini adalah Program Dinamis. Pendekatan pada program dinamis yang digunakan yaitu dengan program dinamis maju, sehingga tahapan meninjau objek dimulai dari objek pertama hingga terakhir.

Pada penyelesaian persoalan ini dengan program dinamis, proses memilih suatu olahraga merupakan tahapan, sedangkan durasi yang tersisa pada dari durasi awal setelah pemilihan olahraga sebelumnya merupakan status. Misalkan status dinyatakan dengan y dan tahap dinyatakan dengan k. Ketika memilih olahraga pada tahap ke k, durasi olahraga tersebut dinyatakan dengan w_k.

Untuk menyelesaikan persoalan ini, pertama-tama perlu dibuat daftar pilihan olahraga yang mungkin berserta durasi dan kalori yang terbakar dalam durasi yang ditentukan. Contoh list pilihan olahraga ada pada Tabel I.

TABEL I. CONTOH PILIHAN OLAHRAGA

Olahraga	Durasi	Kalori
Jogging	30	340
Lompat tali	30	375
Push up	10	130
Bersepeda	30	136
Basket	20	148
Berenang cepat	20	235
Aerobik	30	340
Sit up	10	60
Squat	10	98
Bulu tangkis	20	91
Pull up	10	130
Yoga	30	185
Back up	10	60
Sepakbola	30	260
Tenis	30	250

Diperlukan penyimpanan nilai optimal setiap status pada setiap tahap untuk kebutuhan di tahap selanjutnya dan juga penyimpanan solusi setiap tahap dan setiap status. Hal ini bisa dipersingkat dengan membuat 2 array integer untuk nilai optimal yaitu f dan f_before dan juga 2 array yang berisi array integer sepanjang banyaknya pilihan olahraga yang akan diisikan dengan 0 jika tidak dipilih dan 1 jika dipilih untuk solusi yaitu sol dan sol_before. 2 array ini merupakan array pada tahap yang sedang dilakukan dan tahapan sebelumnya, karena ketika menghitung setiap status pada sebuah tahapan, nilai optimal dan solusi yang perlu diperhatikan hanya berdasarkan tahapan yang tepat sebelumnya. Nilai optimal pada tahapan 0 dianggap 0 karena belum ada olahraga yang dipilih. Begitu juga dengan rangkaian solusi pada tahapan 0 dianggap 0 karena belum ada olahraga yang dipilih. Diperlukan juga durasi maksimal sebagai batasan, Di program ini dimasukan durasi maksimal pada TimeMax dengan nilai 60. Daftar pilihan olahraga yang ada pada Tabel 1 juga dimasukan pada sport dan SportMax berisi banyaknya olahraga yang ada pada sport. Pada program yang telah dibuat oleh penulis, program dibuat dengan bahasa Python.

```
sport = [("Jogging",30,340), ("Lompat
tali",30,375), ("Push up",10,130),
("Bersepeda",30,136), ("Basket",20,148),
("Berenang cepat",20,235), ("Aerobik",30,340),
("Sit up",10,60), ("Squat",10,98), ("Bulu
tangkis",20,91), ("Pull up",10,130),
("Yoga",30,185), ("Back up",10,60),
("Sepakbola",30,260), ("Tenis",30,250)]

TimeMax = 60

SportMax = len(sport)

sol = [[0 for i in range (SportMax)] for j in
range (TimeMax + 1)]

sol_before = [[0 for i in range (SportMax)] for j
in range (TimeMax + 1)]

f = [0 for i in range (TimeMax + 1)]

f_before = [0 for i in range (TimeMax + 1)]
```

Hal yang harus diperhatikan pertama yaitu mengecek apakah $w_k \geq y$. Jika terpenuhi, maka sisa kapasitas setelah objek dimasukan ke dalam ransel yaitu $y - w_k$. Untuk mengisi sisa kapasitas tersebut, digunakan nilai dari tahapan sebelumnya dengan status $y - w_k$, sehingga didapatkan nilai sementara pada tahap k dan status y. sedangkan jika $w_k \geq y$ tidak terpenuhi, maka nilai sementara berupa nilai negatif karena tidak akan dipilih sebagai nilai optimal, atau dalam program ini diasumsikan 0 karena nilai optimal pada tahapan sebelumnya pasti lebih besar atau sama dengan 0 sehingga nilai terbesar yang terpilih menjadi nilai optimal pada status tersebut pasti lebih besar atau sama degan 0. Nilai sementara tersebut kemudian dibandingkan dengan nilai optimal yang telah didapat pada tahap sebelumnya dengan status yang sama yaitu pada tahap ke $k - 1$ dan status y. Nilai terbesar yang didapatkan dari kedua nilai tersebut merupakan nilai optimum pada tahap tersebut dan status tersebut.

Program dinamis seharusnya dilakukan dengan rekurens dengan basis 2 basis yaitu

$f_0(y) = 0, y = 0, 1, 2, ..., M$

$f_k(y) = -\infty, y < 0$

Sedangkan rekurens pada program dinamis yang dilakukan yaitu sebagai berikut

$f_k(y) = max \{f_{k-1}(y), p_k + f_{k-1}(y - w_k)\}, k = 1, 2, 3, ..., N$

M adalah total status yang pada program ini sejumlah TimeMax dan N adalah total tahap yang pada program ini sejumlah SportMax. Tetapi pada program ini diubah tidak melakukan rekurens tapi melakukan pengulangan dari $k = 1$ hingga $k = N$ pada fungsi program_dinamis(), basis pertama telah diinisialisasi di awal program pada array f_before, dan basis kedua merupakan pengecekan yang ada di dalam pengulangan.

Fungsi program_dinamis() melakukan pengulangan pada tahapan. Pengulangan pada tahapan melakukan updateF() yaitu melakukan pemindahan f yang berisi array integer dari nilai optimal semua status pada tahap ke – i ke f_before kemudian melakukan updateSol() yaitu melakukan pemindahan sol yang berisi array dari array integer (0/1) yang menggambarkan diambil atau tidaknya pilihan olahraga pada tahap ke – i ke sol_before. Setelah itu melakukan pengulangan pada status atau pada durasi dari 0 hingga TimeMax.

Pengulangan pada status membandingkan antara 2 nilai yaitu nilai optimal tahapan sebelumnya (f1) dan nilai sementara pada tahap ke – i (f2). Caranya yaitu dengan menghitung durasi yang tersisa (s) jika olahraga pada tahap ke – i dipilh kemudian mengecek apakah s bernilai negatif atau tidak. Jika bernilai negatif maka f2 bernilai 0, sedangkan s bukan bilangan negatif maka f2 berisi kalori yang terbakar pada pilihan olahraga pada tahap ke – i. Setelah didapat f1 dan f2, kedua nilai tersebut dibandingkan. Jika f2 atau nilai sementara lebih besar, maka sol diubah sesuai dengan sol_before dengan status s dan sol pada status ini pada olahraga ke – (i-1) menjadi 1 atau terpilih. Nilai terbesar antara f1 dan f2 dimasukkan ke f pada tahap tersebut. Berikut fungsi program_dinamis() dan fungsi- fungsi lain yang mendukungnya:

```
# menyalin f ke f_before
def updateF():
  global f_before
  for i in range (TimeMax + 1):
    f_before[i] = f[i]

# menyalin sol ke sol_before
def updateSol():
  global sol_before
  for i in range (TimeMax + 1):
     for j in range(SportMax):
       sol_before[i][j] = sol[i][j]

# mengganti sol pada tahap ke y dengan sol_before
pada tahap ke s
def replaceSol(y,s):
  global sol
  for i in range (SportMax):
     sol[y][i] = sol_before[s][i]
```

```
# melakukan program dinamis
def program_dinamis():
  global f, sol
  for i in range (1,SportMax+1):
     updateF()
     updateSol()
     for j in range (TimeMax+1):
       f1 = f_before[j]
       s = j - sport[i-1][1] # j - (time di sport
ke i-1)
       if (s < 0):
         f2 = 0
       else:
         f2 = sport[i-1][2] + f_before[s]
       if (f2 > f1):
         replaceSol(j,s)
         sol[j][i-1] = 1
         f[j] = f2
       else:
         f[j] = f1
```

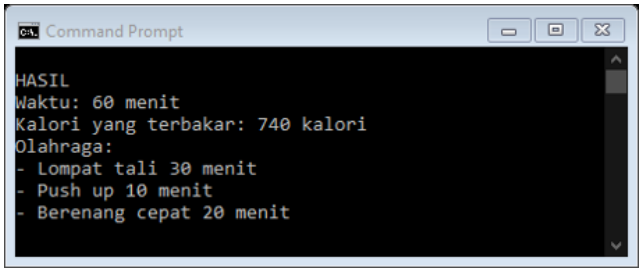
Jika program dinamis sudah dilakukan, maka f sudah berisi nilai optimal pada setiap durasi dan juga sol sudah berisi rangkaian pilihan olahraga yang paling optimal sesuai dengan f. Fungsi getMax() mencari nilai maksimal dari array f. Fungsi ini mengembalikan tuple yang berisi nilai maksimum dari f dan array berisi indeks pada yang memiliki nilai maksimum tersebut pada sol. Hal ini dilakukan karena adanya kemungkinan bahwa terdapat lebih dari 1 pilihan solusi yang memiliki nilai maksimum tersebut. Berikut fungsi getMax().

```
# mencari nilai maksimal pada f
def getMax():
  m = f[0]
  arr = [0]
  for i in range (1,TimeMax+1):
    if (m < f[i]):
         m = f[i]
         arr.clear()
         arr.append(i)
    elif (m == f[i]):
         arr.append(i)
  return (m,arr)
```

Untuk algoritma utama cukup dengan memanggil fungsi program_dinamis(), getMax(), dan printHasil() yang mencetak hasil solusi. Berikut algoritma utama beserta fungsi-fungsi lain yang mendukung algoritma utama.

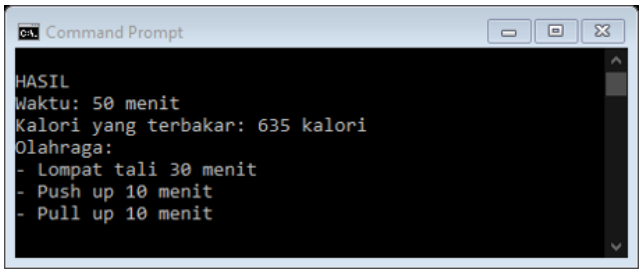
```
# menampilkan ke layar olahraga yang ada pada sol
indeks ke s
def printSol(s):
  for i in range (SportMax):
    if(s[i] == 1):
      print("-",sport[i][0],sport[i][1],"menit")
  print("")

# menampilkan ke layar olahraga yang terpilih
dari tuple final yang berisi jumlah kemungkinan
pilihan yang maksimal dan array yang berisi
indeks dari sol
def printHasil(final):
  kalories = final[0]
  result = final[1]
  print("\nHASIL")
  print("Waktu:",TimeMax,"menit")
  print("Kalori yang
terbakar:",kalories,"kalori")
  print("Olahraga:")
  for i in result:
        printSol(sol[i])

program_dinamis()
final = getMax()
printHasil(final)
```

Program di atas memberikan hasil seperti berikut.



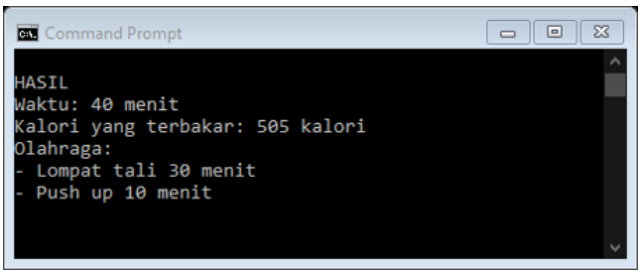
Gambar 3. Tampilan hasil program dengan

Jika durasi pada program (TimeMax) diganti dengan 50, program akan menghasilkan tampilan seperti berikut.



Gambar 4. Tampilan hasil program dengan durasi 50

Jika durasi pada program (TimeMax) diganti dengan 40, program akan menghasilkan tampilan seperti berikut.



Gambar 5. Tampilan hasil program dengan durasi 40

Ketiga hasil program pada Gambar 3, 4, dan 5 dengan data pilihan olahraga yang sama memberikan hasil yang berbeda-beda sesuai dengan durasinya. Hasil pada durasi yang lebih kecil belum tentu sama dengan hasil dengan durasi yang lebih besar. Hal ini disesbabkan oleh adanya pilihan olahraga yang durasinya mencukupi dan kalori yang terbakar lebih besar daripada durasi yang lebih kecil. Jika dicari secara manual, maka akan didapatkan hasil yang serupa.

IV. Kesimpulan dan Saran

Berdasarkan hasil dari implementasi yang dilakukan, dapat disimpulkan bahwa program dinamis dapat menyelesaikan persoalan ini. Program dinamis juga memberikan hasil yang optimal sehingga rangkaian olahraga pada solusi membakar kalori secara maksimal pada durasi tertentu. Untuk pilihan olahraga yang lebih banyak dan durasi yang lebih panjang, algoritma ini sangat berguna. Penyelesaian persoalan ini tentunya dapat membantu sesorang yang memiliki waktu terbatas untuk melakukan olahraga apalagi dengan durasi yang berbeda-beda setiap sesi olahraga. Dibandingkan dengan pencarian solusi secara manual, algoritma ini tergolong cepat. Saran dalam persoalan ini dari penulis yaitu jika pilihan olahraga hanya sedikit, persoalan ini lebih baik diselesaikan secara manual.

Ucapan Terima Kasih

Terima kasih dan puji syukur penulis sampaikan kepada Tuhan Yang Maha Esa yang telah memberikan kesempatan untuk membuat makalah ini juga atas kelancaran dalam pembuatan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada seluruh dosen pengampu mata kuliah IF2211 Strategi Algoritma atas ilmu yang telah diberikan sehingga penulis dapat menuliskan isi dari makalah ini dengan baik dan juga kepada semua pihak yang telah membantu dalam keberjalanan pembuatan makalah ini.

Referensi

[1] Rinaldi Munir, Diktat kuliah IF2251 Strategi Algoritmik, Teknik Informatika ITB, 2007.

[2] https://www.guru99.com/knapsack-problem-dynamic-programming.html diakses pada 3 Mei 2020 pukul 20.52 WIB.

[3] https://nationalgeographic.grid.id/read/131857186/ke-mana-perginya-lemak-yang-terbakar-saat-berolahraga?page=all diakses pada 4 Mei 2020 pukul 21.33 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, 4 Mei 2020

Made Prisha Wulansari 13518049