

Laporan praktikum DAA (latihan Analisis Amortisasi)

Nama : Yohanes Yeningga

Nim : 20220047

Matkul : Prak. DAA

1. Analisis amortisasi adalah suatu metode perhitungan yang digunakan untuk mengestimasi dan mengalokasikan biaya atau nilai aset selama periode waktu tertentu. Dalam konteks keuangan, amortisasi mengacu pada proses mengurangi nilai suatu aset yang memiliki umur atau masa manfaat yang terbatas.

Contoh kasus kehidupan nyata yang dapat dijelaskan dengan analisis amortisasi adalah pembelian mobil. Misalkan seseorang membeli mobil baru seharga \$30.000 dengan umur pakai perkiraan 5 tahun. Dalam hal ini, pembelian mobil adalah investasi awal yang harus dialokasikan secara amortisasi selama periode 5 tahun.

Untuk menghitung amortisasi mobil tersebut, kita dapat menggunakan metode garis lurus. Dalam metode ini, nilai aset dikurangi secara merata dalam setiap periode selama masa manfaatnya. Dalam kasus ini, kita dapat menggunakan rumus berikut:

Amortisasi per tahun = (Nilai Awal - Nilai Sisa) / Umur Pakai

Dalam contoh ini, kita asumsikan nilai sisa mobil setelah 5 tahun adalah \$5.000. Maka, menggunakan rumus di atas, kita dapat menghitung amortisasi per tahun sebagai berikut:

Amortisasi per tahun = $(\$30.000 - \$5.000) / 5 = \$5.000$

Dengan demikian, seseorang dapat mengalokasikan biaya sebesar \$5.000 per tahun selama 5 tahun untuk mengkompensasi biaya pembelian mobil.

Dalam contoh ini, analisis amortisasi memungkinkan seseorang untuk memahami dan merencanakan pengeluaran dalam jangka waktu yang lebih panjang dengan memperhitungkan penyusutan nilai aset. Ini membantu dalam manajemen keuangan pribadi atau bisnis dengan menghitung secara akurat biaya yang diperlukan untuk memperoleh, mempertahankan, atau mengganti aset selama masa manfaatnya.

2. Dalam pemrograman, teknik analisis amortisasi digunakan untuk menghitung biaya atau kompleksitas operasi dalam pengembangan aplikasi. Hal ini membantu pengembang dalam memahami dan memperkirakan penggunaan sumber daya yang diperlukan oleh suatu operasi atau algoritma.

Salah satu contoh penggunaan analisis amortisasi dalam pengembangan aplikasi adalah ketika menghitung kompleksitas waktu atau ruang dari suatu algoritma. Misalnya, dalam struktur data seperti array dinamis atau daftar terkait, operasi penyisipan elemen di tengah struktur dapat memiliki biaya yang tinggi karena memerlukan pergeseran elemen yang ada.

Dalam hal ini, analisis amortisasi dapat membantu dalam memperkirakan biaya rata-rata dari operasi penyisipan elemen dalam jangka waktu yang lebih panjang. Sebagai contoh, kita dapat menggunakan analisis amortisasi untuk menghitung kompleksitas waktu rata-rata dari operasi penyisipan elemen dalam array dinamis.

Misalkan kita menggunakan metode "resize doubling" di mana array dinamis diperbesar dua kali lipat setiap kali kapasitas penuh. Dalam hal ini, analisis amortisasi dapat membantu kita memahami bahwa meskipun operasi penyisipan elemen tertentu memerlukan pergeseran elemen yang ada, biaya rata-rata dari operasi tersebut tetap terjaga karena array dinamis diperbesar secara eksponensial.

Dengan menggunakan analisis amortisasi, kita dapat memprediksi kompleksitas waktu rata-rata dari operasi penyisipan elemen dalam array dinamis dan memastikan bahwa penggunaan sumber daya yang dibutuhkan tetap efisien dalam jangka waktu yang lebih panjang.

Dalam pengembangan aplikasi, penggunaan analisis amortisasi membantu dalam merencanakan dan merancang struktur data atau algoritma yang efisien, mengoptimalkan penggunaan sumber daya, dan memastikan kinerja aplikasi secara keseluruhan.

3. Untuk menghitung besarnya pembayaran cicilan bulanan dan bunga pada pembiayaan kredit mobil menggunakan teknik analisis amortisasi, kita perlu memperhatikan beberapa faktor seperti jumlah pinjaman, suku bunga, periode pembayaran, dan jenis penghitungan bunga (misalnya, bunga tetap atau bunga menurun). Berikut adalah

contoh perhitungan menggunakan metode bunga tetap dan periode pembayaran bulanan:

```
+ Kode + Teks
def hitung_cicilan(jumlah_pinjaman, suku_bunga, periode):
    bunga_per_bulan = suku_bunga / 12 / 100 # Konversi suku bunga tahunan menjadi suku bunga bulanan
    jumlah_cicilan = jumlah_pinjaman * (bunga_per_bulan * (1 + bunga_per_bulan)**periode) / ((1 + bunga_per_bulan)**periode - 1)
    return jumlah_cicilan

def hitung_bunga(jumlah_pinjaman, suku_bunga, sisa_pinjaman):
    bunga_per_bulan = suku_bunga / 12 / 100 # Konversi suku bunga tahunan menjadi suku bunga bulanan
    bunga = sisa_pinjaman * bunga_per_bulan
    return bunga

# Contoh penggunaan
jumlah_pinjaman = 30000 # Jumlah pinjaman dalam dolar
suku_bunga = 5 # Suku bunga tahunan
periode = 5 # Periode pembayaran dalam tahun
sisa_pinjaman = jumlah_pinjaman

total_cicilan = hitung_cicilan(jumlah_pinjaman, suku_bunga, periode)
print("Jumlah cicilan bulanan: $", round(total_cicilan, 2))

for bulan in range(1, periode * 12 + 1):
    bunga = hitung_bunga(jumlah_pinjaman, suku_bunga, sisa_pinjaman)
    pembayaran_pokok = total_cicilan - bunga
    sisa_pinjaman -= pembayaran_pokok
    print("Bulan", bulan, "- Pembayaran Pokok: $", round(pembayaran_pokok, 2), "- Bunga: $", round(bunga, 2), "- Sisa Pinjaman: $", round(sisa_pinjaman, 2))

pembayaran_pokok = total_cicilan - bunga
sisa_pinjaman -= pembayaran_pokok
print("Bulan", bulan, "- Pembayaran Pokok: $", round(pembayaran_pokok, 2), "- Bunga: $", round(bunga, 2), "- Sisa Pinjaman: $", round(sisa_pinjaman, 2))

Bulan 3 - Pembayaran Pokok: $ 5999.9 - Bunga: $ 75.31 - Sisa Pinjaman: $ 12074.9
Bulan 4 - Pembayaran Pokok: $ 6024.9 - Bunga: $ 50.31 - Sisa Pinjaman: $ 6050.0
Bulan 5 - Pembayaran Pokok: $ 6050.0 - Bunga: $ 25.21 - Sisa Pinjaman: $ 0.0
Bulan 6 - Pembayaran Pokok: $ 6075.21 - Bunga: $ 0.0 - Sisa Pinjaman: $ -6075.21
Bulan 7 - Pembayaran Pokok: $ 6100.52 - Bunga: $ -25.31 - Sisa Pinjaman: $ -12175.73
Bulan 8 - Pembayaran Pokok: $ 6125.94 - Bunga: $ -50.73 - Sisa Pinjaman: $ -18301.67
Bulan 9 - Pembayaran Pokok: $ 6151.46 - Bunga: $ -76.26 - Sisa Pinjaman: $ -24453.13
Bulan 10 - Pembayaran Pokok: $ 6177.1 - Bunga: $ -101.89 - Sisa Pinjaman: $ -30630.23
Bulan 11 - Pembayaran Pokok: $ 6202.83 - Bunga: $ -127.63 - Sisa Pinjaman: $ -36833.06
Bulan 12 - Pembayaran Pokok: $ 6228.68 - Bunga: $ -153.47 - Sisa Pinjaman: $ -43061.74
Bulan 13 - Pembayaran Pokok: $ 6254.63 - Bunga: $ -179.42 - Sisa Pinjaman: $ -49316.37
Bulan 14 - Pembayaran Pokok: $ 6280.69 - Bunga: $ -205.48 - Sisa Pinjaman: $ -55597.07
Bulan 15 - Pembayaran Pokok: $ 6306.86 - Bunga: $ -231.65 - Sisa Pinjaman: $ -61903.93
Bulan 16 - Pembayaran Pokok: $ 6333.14 - Bunga: $ -257.93 - Sisa Pinjaman: $ -68237.07
Bulan 17 - Pembayaran Pokok: $ 6359.53 - Bunga: $ -284.32 - Sisa Pinjaman: $ -74596.6
Bulan 18 - Pembayaran Pokok: $ 6386.03 - Bunga: $ -310.82 - Sisa Pinjaman: $ -80982.63
Bulan 19 - Pembayaran Pokok: $ 6412.64 - Bunga: $ -337.43 - Sisa Pinjaman: $ -87395.26
Bulan 20 - Pembayaran Pokok: $ 6439.35 - Bunga: $ -364.15 - Sisa Pinjaman: $ -93834.62
Bulan 21 - Pembayaran Pokok: $ 6466.19 - Bunga: $ -390.98 - Sisa Pinjaman: $ -100300.8
Bulan 22 - Pembayaran Pokok: $ 6493.13 - Bunga: $ -417.92 - Sisa Pinjaman: $ -106793.93
Bulan 23 - Pembayaran Pokok: $ 6520.18 - Bunga: $ -444.97 - Sisa Pinjaman: $ -113314.11
Bulan 24 - Pembayaran Pokok: $ 6547.35 - Bunga: $ -472.14 - Sisa Pinjaman: $ -119861.46
Bulan 25 - Pembayaran Pokok: $ 6574.63 - Bunga: $ -499.42 - Sisa Pinjaman: $ -126436.09
Bulan 26 - Pembayaran Pokok: $ 6602.02 - Bunga: $ -526.82 - Sisa Pinjaman: $ -133038.12
Bulan 27 - Pembayaran Pokok: $ 6629.53 - Bunga: $ -554.33 - Sisa Pinjaman: $ -139667.65
Bulan 28 - Pembayaran Pokok: $ 6657.16 - Bunga: $ -581.95 - Sisa Pinjaman: $ -146324.81
Bulan 29 - Pembayaran Pokok: $ 6684.89 - Bunga: $ -609.69 - Sisa Pinjaman: $ -153009.7
Bulan 30 - Pembayaran Pokok: $ 6712.75 - Bunga: $ -637.54 - Sisa Pinjaman: $ -159722.45
Bulan 31 - Pembayaran Pokok: $ 6740.72 - Bunga: $ -665.51 - Sisa Pinjaman: $ -166463.17
Bulan 32 - Pembayaran Pokok: $ 6768.8 - Bunga: $ -693.6 - Sisa Pinjaman: $ -173231.97
Bulan 33 - Pembayaran Pokok: $ 6797.01 - Bunga: $ -721.8 - Sisa Pinjaman: $ -180028.98
Bulan 34 - Pembayaran Pokok: $ 6825.33 - Bunga: $ -750.12 - Sisa Pinjaman: $ -186854.31
Bulan 35 - Pembayaran Pokok: $ 6853.77 - Bunga: $ -778.56 - Sisa Pinjaman: $ -193708.08
Bulan 36 - Pembayaran Pokok: $ 6882.32 - Bunga: $ -807.12 - Sisa Pinjaman: $ -200590.4
Bulan 37 - Pembayaran Pokok: $ 6911.0 - Bunga: $ -835.79 - Sisa Pinjaman: $ -207501.4
Bulan 38 - Pembayaran Pokok: $ 6939.8 - Bunga: $ -864.59 - Sisa Pinjaman: $ -214441.2
Bulan 39 - Pembayaran Pokok: $ 6968.71 - Bunga: $ -893.51 - Sisa Pinjaman: $ -221409.91
Bulan 40 - Pembayaran Pokok: $ 6997.75 - Bunga: $ -922.54 - Sisa Pinjaman: $ -228407.66
Bulan 41 - Pembayaran Pokok: $ 7026.91 - Bunga: $ -951.7 - Sisa Pinjaman: $ -235434.57
Bulan 42 - Pembayaran Pokok: $ 7056.19 - Bunga: $ -980.98 - Sisa Pinjaman: $ -242490.76
Bulan 43 - Pembayaran Pokok: $ 7085.59 - Bunga: $ -1010.38 - Sisa Pinjaman: $ -249576.34
Bulan 44 - Pembayaran Pokok: $ 7115.11 - Bunga: $ -1039.9 - Sisa Pinjaman: $ -256691.45
Bulan 45 - Pembayaran Pokok: $ 7144.76 - Bunga: $ -1069.55 - Sisa Pinjaman: $ -263836.21
Bulan 46 - Pembayaran Pokok: $ 7174.53 - Bunga: $ -1099.32 - Sisa Pinjaman: $ -271010.73
Bulan 47 - Pembayaran Pokok: $ 7204.42 - Bunga: $ -1129.21 - Sisa Pinjaman: $ -278215.15
Bulan 48 - Pembayaran Pokok: $ 7234.44 - Bunga: $ -1159.23 - Sisa Pinjaman: $ -285449.59
```

+ Kode	+ Teks
0d	<p>Bulan 44 - Pembayaran Pokok: \$ 7115.11 - Bunga: \$ -1039.9 - Sisa Pinjaman: \$ -256691.45</p> <p>Bulan 45 - Pembayaran Pokok: \$ 7144.76 - Bunga: \$ -1069.55 - Sisa Pinjaman: \$ -263836.21</p> <p>Bulan 46 - Pembayaran Pokok: \$ 7174.53 - Bunga: \$ -1099.32 - Sisa Pinjaman: \$ -271010.73</p> <p>Bulan 47 - Pembayaran Pokok: \$ 7204.42 - Bunga: \$ -1129.21 - Sisa Pinjaman: \$ -278215.15</p> <p>Bulan 48 - Pembayaran Pokok: \$ 7234.44 - Bunga: \$ -1159.23 - Sisa Pinjaman: \$ -285449.59</p> <p>Bulan 49 - Pembayaran Pokok: \$ 7264.58 - Bunga: \$ -1189.37 - Sisa Pinjaman: \$ -292714.17</p> <p>Bulan 50 - Pembayaran Pokok: \$ 7294.85 - Bunga: \$ -1219.64 - Sisa Pinjaman: \$ -300009.02</p> <p>Bulan 51 - Pembayaran Pokok: \$ 7325.25 - Bunga: \$ -1250.04 - Sisa Pinjaman: \$ -307334.27</p> <p>Bulan 52 - Pembayaran Pokok: \$ 7355.77 - Bunga: \$ -1280.56 - Sisa Pinjaman: \$ -314690.03</p> <p>Bulan 53 - Pembayaran Pokok: \$ 7386.42 - Bunga: \$ -1311.21 - Sisa Pinjaman: \$ -322076.45</p> <p>Bulan 54 - Pembayaran Pokok: \$ 7417.19 - Bunga: \$ -1341.99 - Sisa Pinjaman: \$ -329493.64</p> <p>Bulan 55 - Pembayaran Pokok: \$ 7448.1 - Bunga: \$ -1372.89 - Sisa Pinjaman: \$ -336941.74</p> <p>Bulan 56 - Pembayaran Pokok: \$ 7479.13 - Bunga: \$ -1403.92 - Sisa Pinjaman: \$ -344420.87</p> <p>Bulan 57 - Pembayaran Pokok: \$ 7510.29 - Bunga: \$ -1435.09 - Sisa Pinjaman: \$ -351931.17</p> <p>Bulan 58 - Pembayaran Pokok: \$ 7541.59 - Bunga: \$ -1466.38 - Sisa Pinjaman: \$ -359472.75</p> <p>Bulan 59 - Pembayaran Pokok: \$ 7573.01 - Bunga: \$ -1497.8 - Sisa Pinjaman: \$ -367045.77</p> <p>Bulan 60 - Pembayaran Pokok: \$ 7604.57 - Bunga: \$ -1529.36 - Sisa Pinjaman: \$ -374650.33</p>

Dalam contoh di atas, kita menggunakan fungsi `hitung_cicilan` untuk menghitung besarnya pembayaran cicilan bulanan berdasarkan jumlah pinjaman, suku bunga, dan periode pembayaran. Fungsi `hitung_bunga` digunakan untuk menghitung jumlah bunga yang harus dibayar setiap bulan berdasarkan sisa pinjaman.

Setelah itu, kita melakukan iterasi untuk setiap bulan dalam periode pembayaran dan menghitung pembayaran pokok, bunga, dan sisa pinjaman. Hasilnya akan dicetak dalam bentuk tabel yang menunjukkan pembayaran bulanan untuk setiap bulan.

Pastikan untuk memodifikasi nilai `jumlah_pinjaman`, `suku_bunga`, dan `periode` sesuai dengan kebutuhan Anda.

4. Keuntungan dari menggunakan teknik analisis amortisasi dalam menghitung kompleksitas waktu dan ruang suatu algoritma adalah sebagai berikut:

Perhitungan yang lebih akurat: Dengan menggunakan analisis amortisasi, kita dapat memperoleh perkiraan yang lebih akurat tentang kompleksitas waktu dan ruang suatu algoritma. Hal ini membantu dalam memahami dan memprediksi kinerja algoritma secara keseluruhan.

- 1). Perhitungan yang lebih akurat: Dengan menggunakan analisis amortisasi, kita dapat memperoleh perkiraan yang lebih akurat tentang kompleksitas waktu dan ruang suatu algoritma. Hal ini membantu dalam memahami dan memprediksi kinerja algoritma secara keseluruhan.
- 2). Penentuan sumber daya yang efisien: Dengan memahami kompleksitas amortisasi, kita dapat menentukan alokasi sumber daya yang efisien untuk suatu operasi atau algoritma. Ini membantu dalam perencanaan dan pengoptimalan penggunaan waktu dan ruang.

- 3). Identifikasi hotspot: Analisis amortisasi membantu dalam mengidentifikasi bagian-bagian kritis dalam algoritma yang memiliki kompleksitas yang tinggi. Dengan mengetahui hotspot ini, kita dapat fokus pada optimasi dan perbaikan kinerja pada area yang membutuhkan perhatian lebih.

Contoh penggunaan analisis amortisasi dalam analisis algoritma adalah pada operasi penyisipan atau penghapusan elemen dalam struktur data seperti array dinamis atau heap binomial. Dalam kasus ini, analisis amortisasi membantu dalam memahami kompleksitas waktu dan ruang operasi tersebut secara rata-rata dalam jangka waktu yang lebih panjang

Misalnya, jika kita menggunakan operasi penyisipan elemen dalam array dinamis dengan metode *resize doubling*, kompleksitas waktu dalam satu operasi mungkin terlihat tinggi karena memerlukan pergeseran elemen. Namun, dengan menggunakan analisis amortisasi, kita dapat menunjukkan bahwa dalam jangka waktu yang lebih panjang, biaya operasi tersebut tetap terjaga secara rata-rata karena array dinamis diperbesar secara eksponensial.

Dengan demikian, analisis amortisasi membantu dalam memahami dan mengevaluasi kompleksitas sebenarnya dari suatu algoritma dengan mempertimbangkan interaksi antara operasi-operasi yang terlibat dalam jangka waktu yang lebih panjang. Ini memungkinkan pengembang untuk membuat keputusan yang lebih bijaksana dalam merancang dan mengoptimalkan algoritma untuk kinerja yang lebih baik.

5. Perbedaan antara analisis amortisasi dan analisis kasus terburuk dalam menghitung kompleksitas algoritma terletak pada fokus dan tujuan dari kedua pendekatan tersebut.

- Analisis Amortisasi:

Analisis amortisasi digunakan untuk menghitung rata-rata biaya atau kompleksitas operasi dalam jangka waktu yang lebih panjang. Tujuan utama dari analisis amortisasi adalah memberikan perkiraan yang lebih akurat tentang biaya rata-rata dari serangkaian operasi dalam algoritma tersebut. Analisis ini membantu dalam mengidentifikasi bagian-bagian kritis yang dapat mengakumulasi biaya tinggi dalam jangka waktu yang panjang.

Contoh: Dalam struktur data seperti array dinamis dengan metode resize doubling, operasi penyisipan elemen mungkin memiliki kompleksitas tinggi dalam satu operasi, tetapi dengan menggunakan analisis amortisasi, kita dapat menunjukkan bahwa dalam jangka waktu yang lebih panjang, biaya operasi tersebut tetap terjaga secara rata-rata karena ukuran array diperbesar secara eksponensial.

- Analisis Kasus Terburuk:

Analisis kasus terburuk menghitung kompleksitas maksimum dari suatu algoritma. Pendekatan ini fokus pada skenario terburuk di mana input terburuk yang mungkin akan memberikan kinerja paling buruk. Analisis kasus terburuk memberikan perkiraan paling pesimistis tentang kompleksitas algoritma dan memberikan batasan atas pada kinerja algoritma dalam situasi terburuk.

Contoh: Dalam algoritma pengurutan seperti algoritma Bubble Sort, kompleksitas kasus terburuk terjadi ketika elemen-elemen dalam array terurut secara terbalik. Dalam hal ini, analisis kasus terburuk menghasilkan kompleksitas $O(n^2)$, yang menunjukkan bahwa algoritma Bubble Sort memiliki kinerja yang buruk pada input terburuk.

Dalam rangka menggambarkan perbedaan antara kedua pendekatan tersebut, dapat dikatakan bahwa analisis amortisasi memberikan gambaran yang lebih realistis tentang kompleksitas rata-rata dalam jangka waktu yang lebih panjang, sementara analisis kasus terburuk memberikan batasan atas terhadap kinerja terburuk yang mungkin terjadi dalam suatu algoritma.