

Berikut adalah **contoh soal dan jawaban** yang sesuai dengan materi **Strategi Algoritma: Divide and Conquer dan Greedy Algorithm**, khusus untuk siswa SMA kelas 11, disajikan dengan bahasa yang mudah dimengerti dan penjelasan langkah demi langkah.

✓ Contoh Soal 1: Menggunakan Strategi *Divide and Conquer* (Merge Sort)

📌 Soal:

Urutkan daftar angka berikut dari terkecil ke terbesar menggunakan strategi **Divide and Conquer**:

[12, 5, 8, 20, 3, 15]

Jelaskan langkah-langkahnya!

✓ Jawaban:

Kita akan gunakan **Merge Sort**, yaitu algoritma pengurutan yang menggunakan strategi *Divide and Conquer*.

Langkah 1: Bagi (Divide)

Kita bagi daftar menjadi dua bagian yang hampir sama besar, terus-menerus, sampai hanya tersisa satu elemen.

Daftar awal:

[12, 5, 8, 20, 3, 15]

- Bagi menjadi:
[12, 5, 8] dan [20, 3, 15]
- Bagi lagi:
[12], [5, 8] → [5], [8]
[20], [3, 15] → [3], [15]

Sekarang semua sudah menjadi bagian kecil (1 elemen), yang otomatis terurut.

Langkah 2: Kuasai & Gabung (Conquer and Combine)

Gabungkan kembali bagian-bagian kecil dengan mengurutkannya.

- Gabung [5] dan [8] → [5, 8]
- Gabung [3] dan [15] → [3, 15]

Sekarang kita punya:

- [12], [5, 8] → gabung:
Bandingkan 12 dan 5 → ambil 5
Bandingkan 12 dan 8 → ambil 8
Sisa 12 → hasil: [5, 8, 12]

- [20] dan [3, 15] → gabung:
Bandingkan 20 dan 3 → ambil 3
Bandingkan 20 dan 15 → ambil 15
Sisa 20 → hasil: [3, 15, 20]

Sekarang kita gabung dua bagian besar:

- [5, 8, 12] dan [3, 15, 20]

Langkah penggabungan:

1. Bandingkan 5 dan 3 → ambil 3 → [3]
2. Bandingkan 5 dan 15 → ambil 5 → [3, 5]
3. Bandingkan 8 dan 15 → ambil 8 → [3, 5, 8]
4. Bandingkan 12 dan 15 → ambil 12 → [3, 5, 8, 12]
5. Sisa 15 dan 20 → tambahkan → [3, 5, 8, 12, 15, 20]

✓ Hasil Akhir:

[3, 5, 8, 12, 15, 20]

🔍 Kesimpulan:

Dengan strategi **Divide and Conquer**, kita memecah masalah besar (mengurutkan 6 angka) menjadi sub-masalah kecil, menyelesaikannya, lalu menggabungkan hasilnya. Ini sangat efisien untuk data besar!

✓ Contoh Soal 2: Menggunakan Strategi *Greedy Algorithm* (Penukaran Uang)

📌 Soal:

Seorang kasir harus memberi kembalian sebesar **Rp 7.800**.

Koin yang tersedia:

- Rp 5.000
- Rp 1.000
- Rp 500
- Rp 200
- Rp 100

Gunakan **Greedy Algorithm** untuk menentukan kombinasi koin yang digunakan, dengan tujuan **jumlah koin sekecil mungkin**.

✓ Jawaban:

Kita gunakan pendekatan *greedy*: selalu pilih koin **terbesar** yang masih bisa digunakan.

Sisa kembalian: Rp 7.800

1. **Ambil koin Rp 5.000**

Sisa: $7.800 - 5.000 = \text{Rp } 2.800$

(Koin: 1×5.000)

2. **Ambil koin Rp 1.000** → bisa 2 kali

$2.800 - 1.000 = 1.800$

$1.800 - 1.000 = 800$

Sisa: Rp 800

(Koin: 2×1.000)

3. **Ambil koin Rp 500**

Sisa: $800 - 500 = \text{Rp } 300$

(Koin: 1×500)

4. **Ambil koin Rp 200**

Sisa: $300 - 200 = \text{Rp } 100$

(Koin: 1×200)

5. **Ambil koin Rp 100**

Sisa: $100 - 100 = 0$

(Koin: 1×100)

✓ Hasil:

- $1 \times \text{Rp } 5.000$
- $2 \times \text{Rp } 1.000$
- $1 \times \text{Rp } 500$
- $1 \times \text{Rp } 200$
- $1 \times \text{Rp } 100$

Total koin: **6 keping**

🔍 Catatan:

Algoritma *Greedy* berhasil di sini karena sistem mata uang Indonesia **memungkinkan** solusi optimal dengan memilih koin terbesar terlebih dahulu. Tapi, jika koin Rp 400 tersedia, mungkin solusi *greedy* tidak lagi optimal. Jadi, *greedy* **tidak selalu** memberi solusi terbaik, tapi **sering cepat dan praktis**.

✓ Soal Latihan Mandiri (Bonus!)

📌 Soal:

Gunakan **strategi Greedy** untuk memilih **jumlah maksimal aktivitas** yang bisa dilakukan oleh seseorang, jika hanya bisa melakukan satu aktivitas dalam satu waktu.

Daftar aktivitas (nama, waktu mulai, waktu selesai):

- A1: 9–11
- A2: 10–12
- A3: 11–13
- A4: 13–15

Petunjuk: Gunakan pendekatan *greedy* berdasarkan **waktu selesai tercepat**.

✓ Jawaban:

Langkah:

- Urutkan aktivitas berdasarkan waktu selesai:
 - A1: 9–11
 - A2: 10–12
 - A3: 11–13
 - A4: 13–15
- Pilih aktivitas pertama (A1, selesai pukul 11)
- Cari aktivitas berikutnya yang **mulai setelah pukul 11** → A3 (mulai pukul 11, boleh karena selesai tepat saat mulai)
- Setelah A3 (selesai 13), pilih A4 (mulai 13)

✓ Aktivitas yang dipilih: A1, A3, A4

Total: **3 aktivitas**

🔍 Ini adalah contoh klasik dari *Activity Selection Problem* yang cocok dengan **strategi Greedy!**

🎯 Kesimpulan:

Strategi	Kapan Digunakan	Kelebihan	Kekurangan
Divide and Conquer	Masalah bisa dipecah, lalu hasil digabung (contoh: pengurutan)	Efisien untuk data besar	Lebih kompleks, butuh rekursi

Strategi	Kapan Digunakan	Kelebihan	Kekurangan
Greedy Algorithm	Pilihan lokal terbaik membawa ke solusi global terbaik (contoh: kembalian, pemilihan aktivitas)	Cepat dan sederhana	Tidak selalu optimal

Semoga contoh soal dan jawaban ini membantumu memahami strategi algoritma dengan lebih baik! 😊