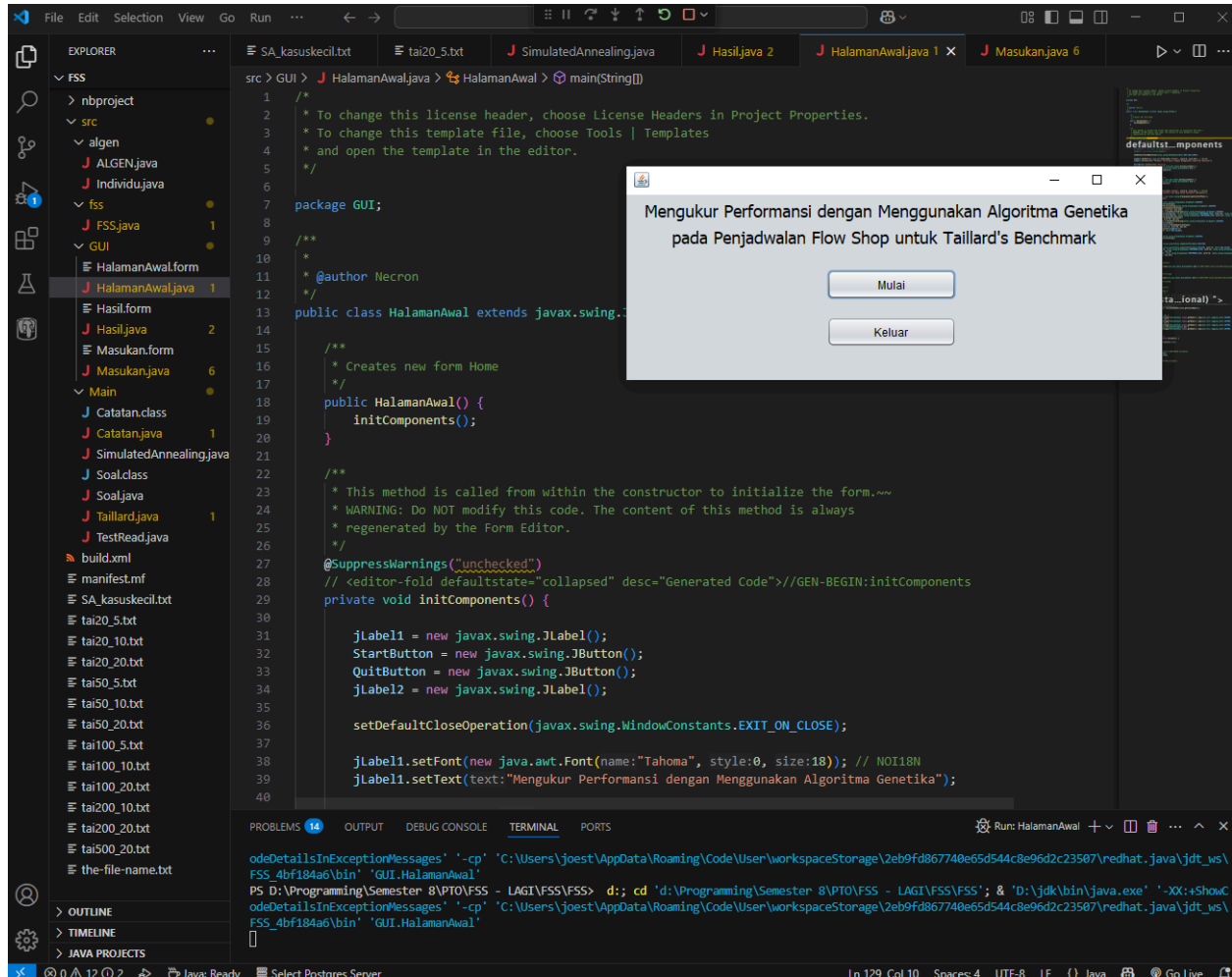
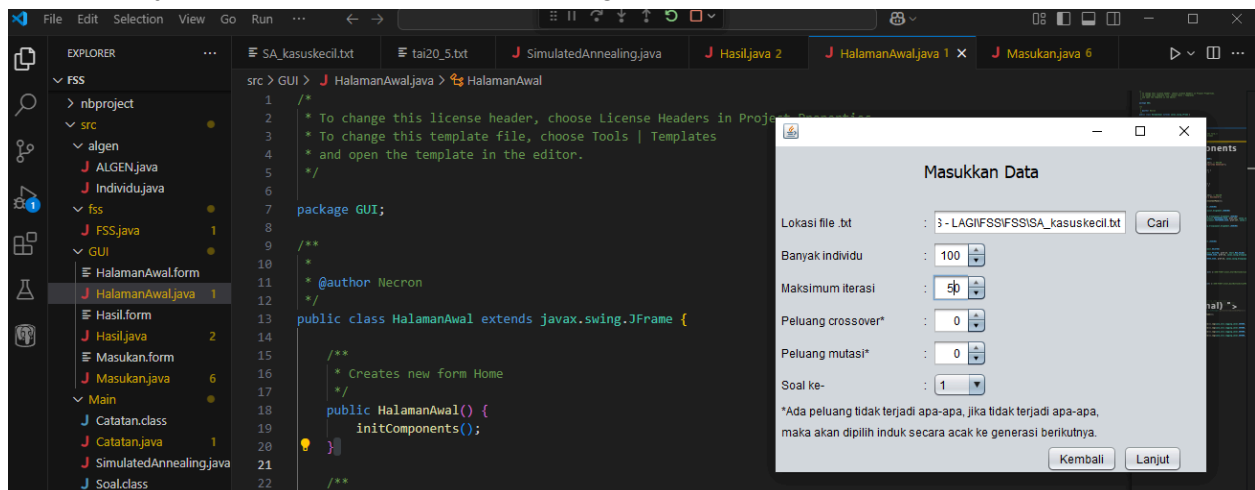


M2_Jonathan Henry_ 6182101047

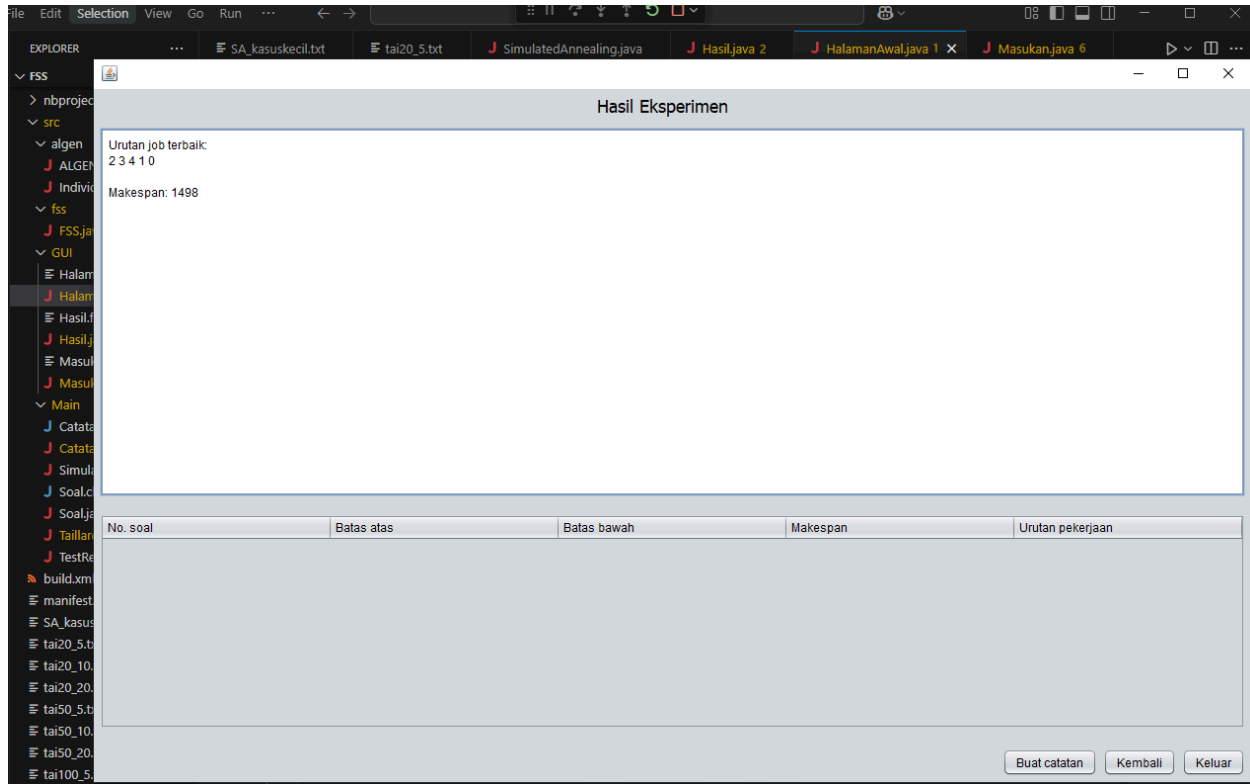


1. Menjalankan HalamanAwal untuk mengakses GUI



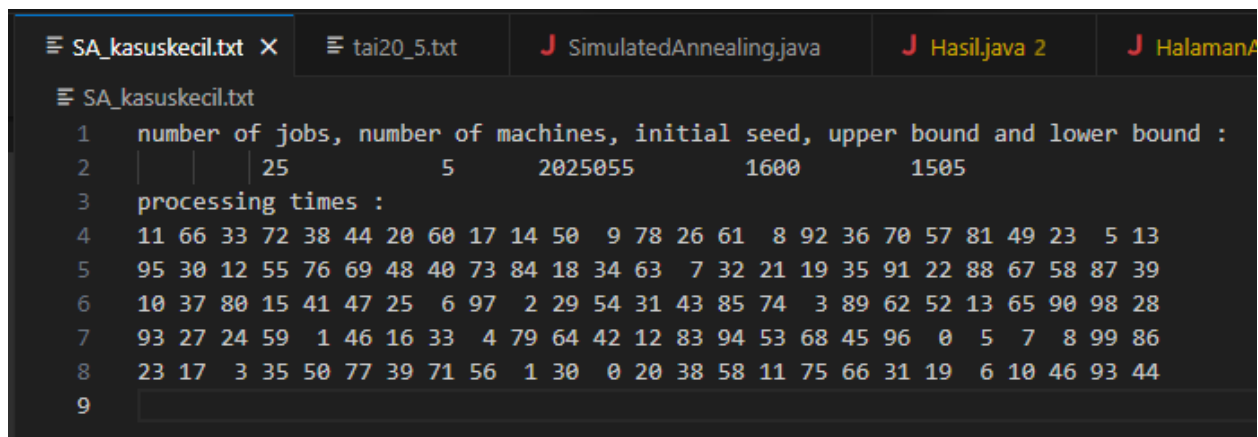
2. Klik 'Mulai' lalu masukan file berisi kasus kecil yang sudah dirancang, lalu input banyak individu serta maksimum iterasinya

M2_Jonathan Henry_ 6182101047



3. Setelah klik 'Lanjut' akan membuka halaman hasil perhitungan dengan Simulated Annealing

Isi dari SA_KasusKecil.txt



Potongan Kode SimulatedAnnealing.java

M2_Jonathan Henry_ 6182101047

```
src > Main > J SimulatedAnnealing.java > SimulatedAnnealing > solusiTerbaik
1  package Main;
2
3  import java.util.Random;
4
5  public class SimulatedAnnealing {
6      private int[][] durasiProses;
7      private int jumlahPekerjaan;
8      private int jumlahMesin;
9      private double suhuAwal;
10     private double lajuPendinginan;
11     private int iterasiMaksimum;
12
13     private int[] solusiTerbaik;
14     private int waktuSelesaiTerbaik;
15
16     public SimulatedAnnealing(String pathFile, int populasi, int iterasi, int peluangCO, int peluangMutasi, int i
17         TestRead pembaca = new TestRead(pathFile);
18         this.durasiProses = pembaca.getKumpulanSoal()[indeksSoal].getSoal();
19         this.jumlahPekerjaan = durasiProses.length;
20         this.jumlahMesin = durasiProses[0].length;
21
22         this.suhuAwal = 1000.0;
23         this.lajuPendinginan = 0.95;
24         this.iterasiMaksimum = iterasi;
25
26         // Langsung eksekusi algoritma
27         this.solusiTerbaik = jalankanAlgoritma();
28         this.waktuSelesaiTerbaik = hitungMakespan(this.solusiTerbaik);
29     }
30
31     public int[] jalankanAlgoritma() {
32         int[] solusiSekarang = buatSolusiAwal();
33         int[] solusiOptimum = solusiSekarang.clone();
34         int nilaiSekarang = hitungMakespan(solusiSekarang);
35         int nilaiOptimum = nilaiSekarang;
36
37         double suhu = suhuAwal;
38         Random acak = new Random();
39
40         for (int langkah = 0; langkah < iterasiMaksimum; langkah++) {
41             int[] tetangga = buatTetangga(solusiSekarang.clone());
42             int nilaiTetangga = hitungMakespan(tetangga);
43
44             if (probabilitasDiterima(nilaiSekarang, nilaiTetangga, suhu) > acak.nextDouble()) {
45                 solusiSekarang = tetangga;
46                 nilaiSekarang = nilaiTetangga;
47
48                 if (nilaiSekarang < nilaiOptimum) {
49                     solusiOptimum = solusiSekarang.clone();

```

M2_Jonathan Henry_ 6182101047

```
src > Main > J SimulatedAnnealing.java > SimulatedAnnealing > SimulatedAnnealing(String, int, int, int, int)
5 public class SimulatedAnnealing {
6     private double probabilitasPenerimaan(int deltaE, int nilaiAwal, double suhu) {
63     }
64
65     private int[] buatSolusiAwal() {
66         int[] urutan = new int[jumlahPekerjaan];
67         for (int i = 0; i < jumlahPekerjaan; i++) urutan[i] = i;
68
69         // Acak posisi pekerjaan
70         Random rng = new Random();
71         for (int i = jumlahPekerjaan - 1; i > 0; i--) {
72             int j = rng.nextInt(i + 1);
73             int tmp = urutan[i];
74             urutan[i] = urutan[j];
75             urutan[j] = tmp;
76         }
77
78         return urutan;
79     }
80
81     private int[] buatTetangga(int[] urutan) {
82         Random rng = new Random();
83         int a = rng.nextInt(urutan.length);
84         int b = rng.nextInt(urutan.length);
85         int tmp = urutan[a];
86         urutan[a] = urutan[b];
87         urutan[b] = tmp;
88         return urutan;
89     }
90
91     public int hitungMakespan(int[] urutan) {
92         int[][] waktuSelesai = new int[jumlahPekerjaan][jumlahMesin];
93
94         for (int i = 0; i < jumlahPekerjaan; i++) {
95             int pekerjaan = urutan[i];
96             for (int j = 0; j < jumlahMesin; j++) {
97                 if (i == 0 && j == 0) {
98                     waktuSelesai[i][j] = durasiProses[pekerjaan][j];
99                 } else if (i == 0) {
100                     waktuSelesai[i][j] = waktuSelesai[i][j - 1] + durasiProses[pekerjaan][j];
101                 } else if (j == 0) {
102                     waktuSelesai[i][j] = waktuSelesai[i - 1][j] + durasiProses[pekerjaan][j];
103                 } else {
104                     waktuSelesai[i][j] = Math.max(waktuSelesai[i - 1][j], waktuSelesai[i][j - 1]) + durasiProses[pekerjaan][j];
105                 }
106             }
107         }
108
109         return waktuSelesai[jumlahPekerjaan - 1][jumlahMesin - 1];
110     }
111 }
```