**TUGAS KECIL III**

**IF2211 STRATEGI ALGORITMA**

**PENYELESAIAN PERSOALAN 15-PUZZLE DENGAN ALGORITMA *BRACH AND BOUND***

**LAPORAN TUGAS KECIL**

Diajukan untuk memenuhi tugas mata kuliah IF2211 Strategi Algoritma

Oleh

**MUHAMMAD FAUZAN AL-GHIFARI**

**(13518112 - K01)**

****

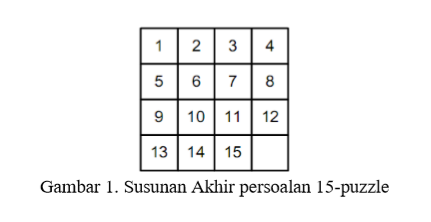
**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2020**

1. **Algoritma *Branch and Bound 15 Puzzle***

Nilai bound tiap simpul adalah penjumlahan cost yang diperlukan untuk sampai suatu simpul x dari akar, dengan taksiran cost simpul x untuk sampai ke goal. Taksiran cost yang digunakan adalah jumlah ubin tidak kosong yang tidak berada pada tempat sesuai susunan akhir (goal state). Untuk semua instansiasi persoalan 15-puzzle, susunan akhir yang diinginkan sesuai dengan Gambar 1.



Cuplikan program bagian algoritma branch and bound

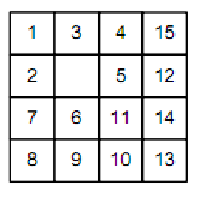
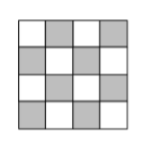
|  |
| --- |
| if (Y % 2 == 1):      print("Permasalahan Puzzle tidak dapat diselesaikan")  else:      matGen.add(str(matExpand.matrix))      heapq.heapify(matHidup)  *# Priority : Up, Right, Down, Left*      while not(matExpand.myBobot() == 0):          empty = matExpand.search('')  *# Ekspan Up*          if (empty.x != 0 and stepBefore != "down"):              matUp = matExpand.moveUp()              if not(str(matUp.matrix) in matGen):                  matUp.aras = matExpand.aras + 1                  matUp.bobot = matUp.myBobot() + matUp.aras                  matUp.stepBefore = "up"                  index += 1                  matUp.index = index                  heapq.heappush(matHidup,matUp)                  matGen.add(str(matUp.matrix))                  matUp.path.append("U")    *# Ekspan Right*          if (empty.y != 3 and stepBefore != "left"):              matRight = matExpand.moveRight()              if not(str(matRight.matrix) in matGen):                  matRight.aras = matExpand.aras + 1                  matRight.bobot = matRight.myBobot() + matRight.aras                  matRight.stepBefore = "right"                  index += 1                  matRight.index = index                  heapq.heappush(matHidup,matRight)                  matGen.add(str(matRight.matrix))                  matRight.path.append("R")  *# Ekspan Down*          if (empty.x != 3 and stepBefore != "up"):              matDown = matExpand.moveDown()              if not(str(matDown.matrix) in matGen):                  matDown.aras = matExpand.aras + 1                  matDown.bobot = matDown.myBobot() + matDown.aras                  matDown.stepBefore = "down"                  index += 1                  matDown.index = index                  heapq.heappush(matHidup,matDown)                  matGen.add(str(matDown.matrix))                  matDown.path.append("D")  *# Ekspan Left*          if (empty.y != 0 and stepBefore != "right"):              matLeft = matExpand.moveLeft()              if not(str(matLeft.matrix) in matGen):                  matLeft.aras = matExpand.aras + 1                  matLeft.bobot = matLeft.myBobot() + matLeft.aras                  matLeft.stepBefore = "left"                  index += 1                  matLeft.index = index                  heapq.heappush(matHidup,matLeft)                  matGen.add(str(matLeft.matrix))                  matLeft.path.append("L")  *# get matExpand from matHidup*          matExpand = heapq.heappop(matHidup)          stepBefore = matExpand.stepBefore    *# Jumlah Simpul*      print("Jumlah simpul yang dibangkitkan = " + str(matExpand.index))      print("Urutan penyelesaian = " + str(matExpand.path) + "\n")      matAwal.path = deepcopy(matExpand.path)      matAwal.printPath()  *# End Execution*  end\_time = time.time() |

Pada program kali ini saya menggunakan kelas Matrix yang memiliki beebrapa atribut

1. matrix menyimpan array of array of char
2. bobot menyimpan bobot matriks
3. stepBefore menyimpan langkah sebelumnya untuk mencapai matriks tersebut
4. Index menyimpan urutan pembangkitan matriks pada program.
5. Aras menyimpan posisi aras (tingkatan) matriks pada pohon
6. Path berupa array pembangkitan matriks dari posisi awal ke posisi matriks tersebut

Langkah penyelesaian 15 Puzzle pada program adalah sebagai berikut

1. Program membaca input dari file txt sesuai pilihan user dan menghitung nilai Y. Nilai Y adalah nilai dari ∑KURANG(i) + X. Jika nilainya ganjil artinya posisi awal puzzle tidak dapat mencapai posisi goal, program akan menampilkan pesan puzzle tidak dapat diselesaikan. Sedangkan jika nilai Y genap, artinya puzzle memiliki solusi mencapai posisi goal. Nilai X ditentukan dengan melihat posisi kosong pada puzzle. JIka posisinya berapa blok yang diarsir maka X bernilai 1, sedangkan jika di blok putih x bernilai 0. Contoh di bawah ini X bernilai 0



1. Pada program utama terdapat matHidup untuk menyimpan simpul hidup. matHidup adalah heap (PrioQueue) dengan pengurutan prioritas berdasarkan bobot matriks, jika bobot sama maka diambil simpul yang paling baru atau dengan indeks yang terbesar. matGen untuk menyimpan matriks yang sudah di generate agar tidak degenerate 2 kali.

1. File di load ke dalam matEkspand. Pertama-tama matExpand dimasukka ke dalam matGen dan ekspan matriks tersebut. Cari posisi ubin kosong pada matEkspan untuk menentukan arah pengekspanan matriks.
2. Matriks dapat diekspan paling banyak menjadi empat matriks, yaitu saat ubin kosong di geser ke up, right, down dan left. Aturan pengekspanan adalah sebagai berikut
3. Matriks dapat diekspan dengan up jika sebelumnya matriks tidak didapat melalui ekspan down, dan ubin kosong tidak berada pada baris paling atas
4. Matriks dapat diekspan dengan right jika sebelumnya matriks tidak didapatkan melalui ekspan left dan ubin kosong tidak berada pada kolom paling kanan
5. Matriks dapat diekspan dengan down jika sebelumnya matriks tidak didapatkan melalui ekspan up dan ubin kosong tidak berada pada baris paling bawah
6. Matriks dapat diekspan dengan left jika sebelumnya matriks tidak didapatkan malalui ekspan right dan ubin kosong tidak berada pada kolom paling kiri
7. Copy matExpand yang sudah diubah posisi ubin kosongnya sesuai arah pengekspanan ke suatu matriks mat. Jika mat belum pernah ada (tidak ada di dalam matGen) maka matriks tersebut merupakan matriks yang dibangkitkan. Set aras matriks +1 dari aras orang tuanya. Set bobot matriks dengan c(P) = f(P) + g(P). f(P) adalah panjang lintasan dari simpul akar ke P (aras matriks tersebut) dan g(P) adalah taksiran lintasan terpendek dari P ke su=impul solusi pada upapohon yang akarnya P, g(P) dihitung dengan method myBobot().
8. Set stepBefore sesuai dengan arah pengekspanan dan indeks +1 dari indeks matriks terakhir yang diekspan. Push matriks tersebut ke dalam matHidup dan matGen, tampahkan juga informasi pada path matriks sesuai arah pengekspanan.
9. Ulangi langkah 5 dan 6 untuk semua arah pengekspanan yang mungkin sesuai dengan aturan pada poin 4 a-d. Setelah itu pilih sebuah simpul dari matHidup untuk dijadikan matEkspan selanjutnya, yaitu memilih matriks dengan bobot terkecil, dan set langkah sebelumnya adalah langkah yang diambil untuk pembangkitan simpul ekspan agar tidak menghasilkan simpul parent lagi di pengekspanan selanjutnya.
10. Ulangi langkah 5-7 sampai nilai g(matEkspan) bernilai 0 yaitu dimana matriks yang dipilih adalah matriks yang sudah mencapai goal state. Cetak informasi yang dibutuhkan sesuai dengan spesifikasi tugas
11. **Screenshot input-output program**

Ubin kosong direpresentasikan dengan ` ` spasi kosong

1. File test1.txt

|  |
| --- |
| 2 3 4 8  1 7 11  5 9 6 12  13 10 14 15 |

output

|  |
| --- |
| pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 1  Kondisi Awal Puzzle  -------------  | 2| 3| 4| 8|  | 1| 7|11|  |  | 5| 9| 6|12|  |13|10|14|15|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           1     3           1     4           1     5           0     6           0     7           2     8           4     9           1     10          0     11          4     12          1     13          1     14          0     15          0     16          8  -------------------------  X = 0  Sigma (KURANG(i)) + X = 24  Jumlah simpul yang dibangkitkan = 47  Urutan penyelesaian = ['U', 'L', 'L', 'L', 'D', 'D', 'R', 'R', 'U', 'L', 'D', 'D', 'R', 'R']  Step ke 1 : U  -------------  | 2| 3| 4|  |  | 1| 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 2 : L  -------------  | 2| 3|  | 4|  | 1| 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 3 : L  -------------  | 2|  | 3| 4|  | 1| 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 4 : L  -------------  |  | 2| 3| 4|  | 1| 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 5 : D  -------------  | 1| 2| 3| 4|  |  | 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 6 : D  -------------  | 1| 2| 3| 4|  | 5| 7|11| 8|  |  | 9| 6|12|  |13|10|14|15|  -------------  Step ke 7 : R  -------------  | 1| 2| 3| 4|  | 5| 7|11| 8|  | 9|  | 6|12|  |13|10|14|15|  -------------  Step ke 8 : R  -------------  | 1| 2| 3| 4|  | 5| 7|11| 8|  | 9| 6|  |12|  |13|10|14|15|  -------------  Step ke 9 : U  -------------  | 1| 2| 3| 4|  | 5| 7|  | 8|  | 9| 6|11|12|  |13|10|14|15|  -------------  Step ke 10 : L  -------------  | 1| 2| 3| 4|  | 5|  | 7| 8|  | 9| 6|11|12|  |13|10|14|15|  -------------  Step ke 11 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|  |11|12|  |13|10|14|15|  -------------  Step ke 12 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|  |14|15|  -------------  Step ke 13 : R  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|  |15|  -------------  Step ke 14 : R  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|15|  |  -------------  Waktu eksekusi = 0.10163068771362305 seconds |

1. File test2.txt

|  |
| --- |
| 2 14 10 4  1 7 3 8  5 6 12  9 13 11 15 |

Output

|  |
| --- |
| pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 2  Kondisi Awal Puzzle  -------------  | 2|14|10| 4|  | 1| 7| 3| 8|  |  | 5| 6|12|  | 9|13|11|15|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           1     3           0     4           2     5           0     6           0     7           3     8           2     9           0     10          8     11          0     12          2     13          1     14          12     15          0     16          7  -------------------------  X = 0  Sigma (KURANG(i)) + X = 38  Jumlah simpul yang dibangkitkan = 4063  Urutan penyelesaian = ['R', 'U', 'U', 'R', 'D', 'D', 'L', 'U', 'U', 'L', 'D', 'D', 'D', 'R', 'U', 'U', 'R', 'D', 'D', 'R']  Step ke 1 : R  -------------  | 2|14|10| 4|  | 1| 7| 3| 8|  | 5|  | 6|12|  | 9|13|11|15|  -------------  Step ke 2 : U  -------------  | 2|14|10| 4|  | 1|  | 3| 8|  | 5| 7| 6|12|  | 9|13|11|15|  -------------  Step ke 3 : U  -------------  | 2|  |10| 4|  | 1|14| 3| 8|  | 5| 7| 6|12|  | 9|13|11|15|  -------------  Step ke 4 : R  -------------  | 2|10|  | 4|  | 1|14| 3| 8|  | 5| 7| 6|12|  | 9|13|11|15|  -------------  Step ke 5 : D  -------------  | 2|10| 3| 4|  | 1|14|  | 8|  | 5| 7| 6|12|  | 9|13|11|15|  -------------  Step ke 6 : D  -------------  | 2|10| 3| 4|  | 1|14| 6| 8|  | 5| 7|  |12|  | 9|13|11|15|  -------------  Step ke 7 : L  -------------  | 2|10| 3| 4|  | 1|14| 6| 8|  | 5|  | 7|12|  | 9|13|11|15|  -------------  Step ke 8 : U  -------------  | 2|10| 3| 4|  | 1|  | 6| 8|  | 5|14| 7|12|  | 9|13|11|15|  -------------  Step ke 9 : U  -------------  | 2|  | 3| 4|  | 1|10| 6| 8|  | 5|14| 7|12|  | 9|13|11|15|  -------------  Step ke 10 : L  -------------  |  | 2| 3| 4|  | 1|10| 6| 8|  | 5|14| 7|12|  | 9|13|11|15|  -------------  Step ke 11 : D  -------------  | 1| 2| 3| 4|  |  |10| 6| 8|  | 5|14| 7|12|  | 9|13|11|15|  -------------  Step ke 12 : D  -------------  | 1| 2| 3| 4|  | 5|10| 6| 8|  |  |14| 7|12|  | 9|13|11|15|  -------------  Step ke 13 : D  -------------  | 1| 2| 3| 4|  | 5|10| 6| 8|  | 9|14| 7|12|  |  |13|11|15|  -------------  Step ke 14 : R  -------------  | 1| 2| 3| 4|  | 5|10| 6| 8|  | 9|14| 7|12|  |13|  |11|15|  -------------  Step ke 15 : U  -------------  | 1| 2| 3| 4|  | 5|10| 6| 8|  | 9|  | 7|12|  |13|14|11|15|  -------------  Step ke 16 : U  -------------  | 1| 2| 3| 4|  | 5|  | 6| 8|  | 9|10| 7|12|  |13|14|11|15|  -------------  Step ke 17 : R  -------------  | 1| 2| 3| 4|  | 5| 6|  | 8|  | 9|10| 7|12|  |13|14|11|15|  -------------  Step ke 18 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|  |12|  |13|14|11|15|  -------------  Step ke 19 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|  |15|  -------------  Step ke 20 : R  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|15|  |  -------------  Waktu eksekusi = 0.6677596569061279 seconds |

1. File test3.txt

|  |
| --- |
| 1 2 7 3  11 9 10 6  12 14 4  5 13 15 8 |

Output

|  |
| --- |
| A  pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 3  Kondisi Awal Puzzle  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  |12|  |14| 4|  | 5|13|15| 8|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           0     3           0     4           0     5           0     6           2     7           4     8           0     9           4     10          4     11          6     12          3     13          1     14          4     15          1     16          6  -------------------------  X = 1  Sigma (KURANG(i)) + X = 36  Jumlah simpul yang dibangkitkan = 212534  Urutan penyelesaian = ['L', 'D', 'R', 'U', 'U', 'L', 'D', 'R', 'R', 'D', 'L', 'U', 'U', 'R', 'R', 'D', 'D', 'L', 'U', 'L', 'U', 'R', 'U', 'R', 'D', 'D', 'D']  Step ke 1 : L  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  |  |12|14| 4|  | 5|13|15| 8|  -------------  Step ke 2 : D  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  | 5|12|14| 4|  |  |13|15| 8|  -------------  Step ke 3 : R  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  | 5|12|14| 4|  |13|  |15| 8|  -------------  Step ke 4 : U  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  | 5|  |14| 4|  |13|12|15| 8|  -------------  Step ke 5 : U  -------------  | 1| 2| 7| 3|  |11|  |10| 6|  | 5| 9|14| 4|  |13|12|15| 8|  -------------  Step ke 6 : L  -------------  | 1| 2| 7| 3|  |  |11|10| 6|  | 5| 9|14| 4|  |13|12|15| 8|  -------------  Step ke 7 : D  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  |  | 9|14| 4|  |13|12|15| 8|  -------------  Step ke 8 : R  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|  |14| 4|  |13|12|15| 8|  -------------  Step ke 9 : R  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|14|  | 4|  |13|12|15| 8|  -------------  Step ke 10 : D  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|14|15| 4|  |13|12|  | 8|  -------------  Step ke 11 : L  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|14|15| 4|  |13|  |12| 8|  -------------  Step ke 12 : U  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|  |15| 4|  |13|14|12| 8|  -------------  Step ke 13 : U  -------------  | 1| 2| 7| 3|  | 5|  |10| 6|  | 9|11|15| 4|  |13|14|12| 8|  -------------  Step ke 14 : R  -------------  | 1| 2| 7| 3|  | 5|10|  | 6|  | 9|11|15| 4|  |13|14|12| 8|  -------------  Step ke 15 : R  -------------  | 1| 2| 7| 3|  | 5|10| 6|  |  | 9|11|15| 4|  |13|14|12| 8|  -------------  Step ke 16 : D  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|11|15|  |  |13|14|12| 8|  -------------  Step ke 17 : D  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|11|15| 8|  |13|14|12|  |  -------------  Step ke 18 : L  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|11|15| 8|  |13|14|  |12|  -------------  Step ke 19 : U  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|11|  | 8|  |13|14|15|12|  -------------  Step ke 20 : L  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|  |11| 8|  |13|14|15|12|  -------------  Step ke 21 : U  -------------  | 1| 2| 7| 3|  | 5|  | 6| 4|  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 22 : R  -------------  | 1| 2| 7| 3|  | 5| 6|  | 4|  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 23 : U  -------------  | 1| 2|  | 3|  | 5| 6| 7| 4|  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 24 : R  -------------  | 1| 2| 3|  |  | 5| 6| 7| 4|  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 25 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7|  |  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 26 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|  |  |13|14|15|12|  -------------  Step ke 27 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|15|  |  ------------- |

1. File test4.txt

|  |
| --- |
| 3 9 1 15  14 11 4 6  13 10 12  2 7 8 5 |

Output

|  |
| --- |
| pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 4  Kondisi Awal Puzzle  -------------  | 3| 9| 1|15|  |14|11| 4| 6|  |13|  |10|12|  | 2| 7| 8| 5|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           0     3           2     4           1     5           0     6           2     7           1     8           1     9           7     10          4     11          7     12          4     13          6     14          10     15          11     16          6  -------------------------  X = 1  Sigma (KURANG(i)) + X = 63  Permasalahan Puzzle tidak dapat diselesaikan  Waktu eksekusi = 0.0 seconds |

1. File test5.txt

|  |
| --- |
| 1 3 4  15 2 13 9  10 11 12 6  7 8 5 14 |

Output

|  |
| --- |
| pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 5  Kondisi Awal Puzzle  -------------  | 1| 3|  | 4|  |15| 2|13| 9|  |10|11|12| 6|  | 7| 8| 5|14|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           0     3           1     4           1     5           0     6           1     7           1     8           1     9           4     10          4     11          4     12          4     13          8     14          0     15          11     16          13  -------------------------  X = 0  Sigma (KURANG(i)) + X = 53  Permasalahan Puzzle tidak dapat diselesaikan  Waktu eksekusi = 0.0 seconds |

**DISCLAIMER**

Secara teori program ini dapat menyelesaikan semua permasalahan 15 Puzzle karena saya sudah mencocokkan hasilnya dengan pekerjakan manual yang saya kerjakan berdasar pada test1.txt, tetapi untuk beberapa kasus matriks yang dibangkitkan terlalu banyak sehingga membutuhkan waktu yang sangat lama atau sebelum mencapai solusi terjadi memory error. Jika ada test case yang tidak berhasil diselesaikan oleh program ini tetapi bisa diselesaikan oleh program lain itu disebabkan oleh pemilihan simpul dengan bobot terkecil saat bobot terkecil jumlahnya lebih dari satu (pemilihan simpul yang diekspan dibebaskan), saya sendiri memilih untuk mengekspan simpul yang paling baru jika ada beberapa simpul dengan bobot yang sama.

**Spesifikasi komputer yang digunakan**

Nama Laptop : Asus E202SA

OS : Ubuntu 18.04

RAM : 2048 MB / 2GB

Processor : Intel(R) Celeron(R) CPU N3060 @1.60Ghz (2 CPUs), ~1.6Ghz

|  |  |  |  |
| --- | --- | --- | --- |
| No | Point | Ya | Tidak |
| 1 | Program berhasil dikompilasi | ✓ |  |
| 2 | Program berhasil *running* | ✓ |  |
| 3 | Program dapat menerima input dan menuliskan output | ✓ |  |
| 4 | Luaran sudah benar untuk semua data uji | ✓ |  |