

LAPORAN TUGAS KECIL

“Penyelesaian *Word Search Puzzle* dengan Algoritma Brute Force”

Laporan Ini Dibuat untuk Memenuhi Tugas Perkuliahan

Mata Kuliah Strategi Algoritma (IF2211)

KELAS 01

Dosen : Dr. Masayu Leylia Khodra, S.T., M.T.



Rio Alexander Audino / 13520088

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

SEMESTER II TAHUN 2021/2022

A. ALGORITMA BRUTE FORCE

Pada program yang saya buat, proses pengaplikasian Algoritma *Brute Force* terbagi ke dalam 4 langkah utama, yakni pembacaan file, validasi arah, validasi kata, dan terakhir penampilan jawaban.

a. Pembacaan File

Agar bisa menerima input file dari user, program harus bisa melakukan pembacaan dari sebuah file txt. File txt disesuaikan dengan standar pada spesifikasi tugas kecil. Proses pembacaan file akan menghasilkan dua buah matrix, matrix *random alphabet (puzzles)* dan matrix *searched words (words)*. Proses pembacaan file di-handle library “fstream”.

b. Validasi Arah

Setelah berhasil melakukan pembacaan file, selanjutnya program akan masuk pada bagian algoritma *Brute Force*. Pada awalnya, akan dilakukan validasi arah terlebih dahulu. Proses validasi arah merupakan proses validasi arah- pengecekan. Proses validasi arah dilakukan untuk mengurangi jumlah komparasi huruf. Misal sekarang berada pada ujung kanan atas papan. Melalui proses validasi arah, hanya arah yang valid seperti kiri, kiri-bawah, atau bawah yang akan dicek, bukan arah atas, atas.

c. Validasi Kata

Setelah didapat arah- pengecekan yang valid, selanjutnya akan masuk pada proses pencocokan tiap huruf (validasi kata). Proses validasi kata mengambil porsi terbanyak dalam proses komputasi. Tiap huruf pertama dan arah yang telah valid, selanjutnya akan di-cek satu persatu sesuai dengan arah- pengecekannya. Sesuai dengan tema laporan kali ini – Algoritma *Brute Force* – tidak ada cara singkat dalam proses validasi kata. Tiap huruf harus dicocokkan satu-persatu dari setiap posisi yang valid.

d. Penampilan Jawaban

Proses penampilan jawaban adalah akhir dari program ini. Program pada akhir prosesnya akan menampilkan kata-kata yang ditemukan dalam *puzzles* tadi. Penampilan jawaban disesuaikan dengan format pada spesifikasi tugas besar. Tiap kata akan ditampilkan satu-persatu pada layar.

e. Waktu Komputasi

Di luar 4 langkah utama ini, ada satu proses lagi untuk mengecek waktu komputasi. Waktu komputasi dihitung setelah proses pembacaan file, dari validasi arah sampai akhir program. Proses perhitungan waktu komputasi di-handle oleh library “chrono”

B. KODE PROGRAM

Pada program ini, saya menggunakan bahasa C++. Pemilihan bahasa C++ didasarkan oleh waktu komputasi bahasa C++ umumnya lebih cepat dibandingkan C maupun Java. Kode program akan saya tampilkan sesuai dengan pembagian pada bagian sebelumnya, yakni pembacaan file, validasi arah, validasi kata, penampilan jawaban, dan waktu komputasi.

No	Proses	Gambar	Deskripsi Kode/Fungsi
1	Pembacaan File	<pre> void readFile(vector<vector<char>> *puzzles, vector<vector<char>> *words, int *m, int *n) { ifstream inFile; char readChar; bool isNewLine; int i; //Opening directory string fileName; cout << "Input filename : "; cin >> fileName; inFile.open("puzzles\\" + fileName); //Error handling if (!inFile) { cerr << "Can't find the specified file"; exit(1); } </pre>	Membuka nama file sesuai input user
		<pre> //Read words per char i = 0; while (inFile.get(readChar)) { if (readChar == '\n') { (*words).push_back({}); i++; } else if (readChar != ' ') { (*words)[i].push_back(readChar); } } inFile.close(); </pre>	Membuat matriks puzzle huruf acak
		<pre> //Read puzzles per char while (inFile.get(readChar)) { if (readChar == '\n') { if (isNewLine) { break; } (*puzzles).push_back({}); isNewLine = true; i++; (*n)++; } else if (readChar != ' ') { (*puzzles)[i].push_back(readChar); (*m)++; isNewLine = false; } } </pre>	Membuat matriks penyimpanan kata yang akan dicari

2	Validasi Arah	<pre> bool validDirection(int i, int j, int wordSize, int direction, int m, int n) { /* 1-8 menunjukkan arah, dimulai dari 1 (atas), 2 (atas kanan), ..., 8 (atas kiri). -> perputaran searah jarum jam */ switch(direction) { case 1: if (i - wordSize + 1 >= 0) { return true; } break; case 2: if (i - wordSize + 1 >= 0 && j + wordSize <= m) { return true; } break; case 3: if (j + wordSize <= m) { return true; } break; case 4: if (j + wordSize <= m && i + wordSize <= n) { return true; } break; case 5: if (i + wordSize <= n) { return true; } break; case 6: if (i + wordSize <= n && j - wordSize + 1 >= 0) { return true; } break; case 7: if (j - wordSize + 1 >= 0) { return true; } break; case 8: if (j - wordSize + 1 >= 0 && i - wordSize + 1 >= 0) { return true; } break; } return false; } </pre>	<p>Memvalidasi arah- pengecekan. Fungsi akan mengembalikan true jika arah tersebut valid</p>
3	Validasi Kata	<pre> bool validWord(vector<vector<char>> puzzles, vector<char> word, int wordSize, int i, int j, int direction, int *numComp) { int verMove = 0, horMove = 0; //Configuring move direction switch(direction) { case 1: verMove = -1; break; case 2: verMove = -1; horMove = 1; break; case 3: horMove = 1; break; case 4: verMove = 1; horMove = 1; break; case 5: verMove = 1; break; case 6: verMove = 1; horMove = -1; break; case 7: horMove = -1; break; case 8: verMove = -1; horMove = -1; break; } } </pre>	<p>Memastikan arah pengecekan. verMove dan horMove adalah arah pengecekan.</p>

		<pre> //Checking the puzzles per char int iCheck = i + verMove; int jCheck = j + horMove; bool isFound = true; for(int k = 1; k < wordSize; k++) { *numComp+=1; //---COMPARING--- if (puzzles[iCheck][jCheck] != word[k]) { isFound = false; break; } iCheck += verMove; jCheck += horMove; } return isFound; } </pre>	Melakukan pengecekan per karakter/huruf.
4	Penampilan Jawaban	<pre> void showAnswer(int n, int m, int direction, vector<char> word, int i, int j, int wordSize) { int verMove = 0, horMove = 0; switch(direction) { case 1: verMove = -1; break; case 2: verMove = -1; horMove = 1; break; case 3: horMove = 1; break; case 4: verMove = 1; horMove = 1; break; case 5: verMove = 1; break; case 6: verMove = 1; horMove = -1; break; case 7: horMove = -1; break; case 8: verMove = -1; horMove = -1; break; } } </pre>	Memastikan arah penampilan jawaban
		<pre> //Print the word for (int iPrint = 0; iPrint < wordSize; iPrint++) { cout << word[iPrint]; } cout << endl; for (int iPrint = 0; iPrint < n; iPrint++) { iCheck = iPrint - 1; iMag = iCheck * verMove; //Magnitude of vertical and horizontal movement for (int jPrint = 0; jPrint < m; jPrint++) { jCheck = jPrint - j; jMag = jCheck * horMove; if (jMag >= 0 && jMag < wordSize && iMag >= 0 && iMag < wordSize) { //Print kanan-kiri if (verMove == 0 && iCheck == 0) { temp = jCheck/horMove; cout << word[temp] << ' '; } //Print Atas-bawah else if (horMove == 0 && jCheck == 0) { temp = iCheck/verMove; cout << word[temp] << ' '; } //Print Diagonal else if (verMove != 0 && horMove != 0 && iMag == jMag) { temp = iCheck/verMove; cout << word[temp] << ' '; } else { cout << "- "; } } else { cout << "- "; } } cout << endl; } cout << endl; } </pre>	Menampilkan kata yang dicari Melakukan iterasi di tiap huruf puzzles.
		<pre> if (jMag >= 0 && jMag < wordSize && iMag >= 0 && iMag < wordSize) { //Print kanan-kiri if (verMove == 0 && iCheck == 0) { temp = jCheck/horMove; cout << word[temp] << ' '; } //Print Atas-bawah else if (horMove == 0 && jCheck == 0) { temp = iCheck/verMove; cout << word[temp] << ' '; } //Print Diagonal else if (verMove != 0 && horMove != 0 && iMag == jMag) { temp = iCheck/verMove; cout << word[temp] << ' '; } else { cout << "- "; } } else { cout << "- "; } } cout << endl; } cout << endl; } </pre>	Memastikan huruf yang ditampilkan masih dalam jangkauan panjang kata Membagi ke beberapa kasus, untuk kanan-kiri, atas-bawah, dan diagonal.

		<pre> int main() { vector<vector<char>> puzzles {}, words {}; vector<char> word; int m = 0, n = 0; bool isFound; //Read Config File readFile(&puzzles, &words, &m, &n); //Readjusting m and n m = (m/n); //Greedy Algorithm auto start = high_resolution_clock::now(); //iterates per word int i, j; int direction, wordIndex = 0, wordSize, wordsSize = words.size(); int numComp = 0; </pre>	<p>Melakukan pembacaan file</p> <p>Memulai waktu perhitungan</p>
5	Program Utama	<pre> while(wordIndex < wordsSize) { isFound = false; i = 0; word = words[wordIndex]; wordSize = word.size(); //iterates per char in puzzle while (i < n && !isFound) { j = 0; while(j < m && !isFound) { //check the first alphabet ///---COMPARING--- numComp += 1; if (word[0] != puzzles[i][j]) { j++; //skipping continue; } </pre>	<p>Melakukan iterasi dari setiap kata yang dicari</p> <p>Melakukan iterasi ke seluruh huruf puzzle</p> <p>Mengecek huruf pertama</p>
		<pre> } //Validating direction for (int direction = 1; direction <= 8; direction++) { //cout << i << " " << j << " " << direction << endl; if(validDirection(i, j, wordSize, direction, m, n)) { //Check if word correct isFound = validWord(puzzles, word, wordSize, i, j, direction, &numComp); if (isFound) { //printf("Pos %d %d Words %d Size %d direction %d Correct %d\n", i, j, wordIndex, wordSize, direction, isFound); showAnswer(n, m, direction, word, i, j, wordSize); break; } } } </pre>	<p>Melakukan iterasi ke setiap arah</p> <p>Melakukan validasi arah, validasi kata, dan penampilan jawaban</p>
		<pre> auto stop = high_resolution_clock::now(); auto duration = duration_cast<microseconds>(stop - start) / 1000; cout << duration.count() << " ms" << endl; cout << "No. of Comparison : " << numComp << endl; return 0; } </pre>	<p>Mengakhiri waktu komputasi</p> <p>Menampilkan statistik komputasi</p>

C. DEMO PROGRAM

a. Medium 1 (1316ms / 5182 Comparison)

[illegible]

b. Medium 2 (1288ms / 6212 comparison)

[illegible]

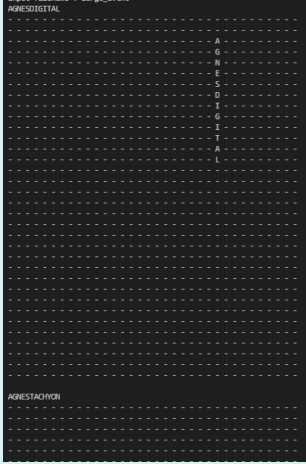
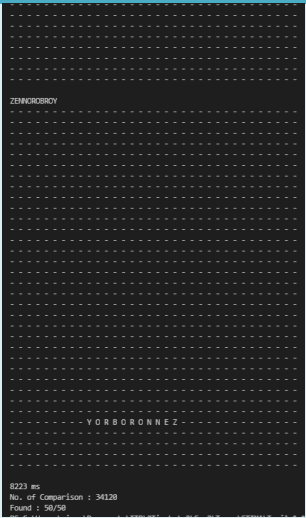
c. Medium 3 (1307ms / 5802 Comparison)

Input	Hasil
<p>JUTTVVLCQZRSMEOGWHZVKI KRSRBRIIMYXHTGNYXQBGHUG HJEIXEFFETEVGNIFYGYVCP FJTUEUXIBTSSOOIJCBUAI DNDMITPPYERSPBHWAUZPHM DNLPEOJNSRSSQDYSOBFMDZ LUOHPUPSYZAHVHZWIHECZ PMMACBEIDEVORPPANUSKLT UEPNWSOAXFFELIFVUMKLYB URHTBNVRFHRHEWMVXPRLRV ZGZONOCCEBCOMPPEPLHUKW DEZBWKIDLJBUAIFFUAMABX KROELSODBPPOFJUIUVXVOV UYDUFQNAAVVSOLRNQEKIWZ NGNPRQVLTQHNFAASKWMCDED SREOQZZPCNNBVXDQOGIAYV CDVWJTBIABJAFONTNRPJLC VKELYPOWLPGIGQRREPETMN EKLIMZXABECKIYVOMKXNCW KTEZFAMDSJSLICEDIYSAYU</p> <p>AVID AVOMED BRIM BROADNESS COPE EFFETE ELEVEN HAZY KNIFE LADDER MERGER OBSESSES PRIMEVAL SELECTABLE SHOWINGS SLICED SLOUCH TRIUMPHANT UNAPPROVED</p>	<p>Input filename : medium_3.txt AVID</p> <p>TRIUMPHANT - T - - R - - I - - U - - M - - P - - H - - A - - N - - T -</p> <p>UNAPPROVED</p> <p>DEAVORPPANU</p> <p>1307 ms No. of Comparison : 5802 Found : 19/19</p>

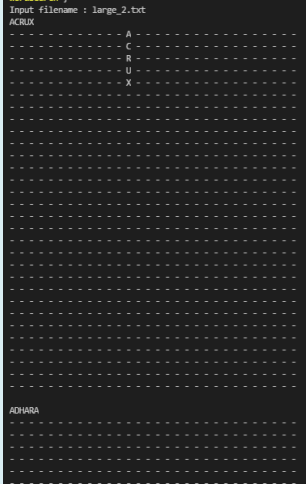
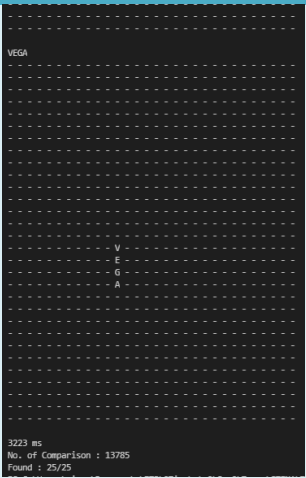
d. Medium 4 (3557ms / 14626 Comparison)

Input	Hasil
<p>TCSDNALREHTENDNALQPSYH ARRATNAUHTILGIDSKBDLO QSBYVQUALDIBATRENNKRON HMLBYTIFESTIKAREANSAYA AELPARATRAGLUBATEUOLAC COADTDPKZANVYVSSWRCPRKO EERSONIRAMNASHSPNUUEER DNUUMLODOVACKPYIENZAU QASDOMANTASAECAKEDTBNV NBOSNTAHERZEGOVINAMIVA EIZHNDORQGRANDPKARUTHTT ATETWEREKACOGIETEISOI HYTTUIONWPLZUNNGORJVMK HANTSRXTAUBHOLKEVAPDA AUAJAGNKIXVLTAEILCSIONN IBNIMHEECFSRBNAAALBRTIC HAGAGARNTVEKHGBTTDOETET CDULAREEHULDQVEAAMUNT ENTQORRGKCOATWPFSTERGEY ZAKUESYEAMCANJYGINAAGA CARRODNAZHLIRDNUTIELRW AGSDNALBTFAUQLVSSCAKZON</p> <p>ALBANIA GERMANY NORWAY AROBRIA GREECE POLAND ARMENIA HUNGARY PORTUGAL AUSTRIA ICELAND ROMANIA AZERBAIJAN IRELAND RUSSIA BELARUS ITALY SAMARINDO BELGIUM KAZAKHSTAN SERBIA ROMANIAHERZAGOVIA LATVIA SLOVAKIA BULGARIA LIECHTENSTEIN SLOVENIA CROATIA LITHUANIA SPAIN CYPRUS LUXEMBOURG SWEDEN CZECHIA HUNGARY SWITZERLAND DENMARK MALTA TURKEY ESTONIA MOLDOVA UKRAINE FINLAND NORWAY UNITEDKINGDOM FRANCE</p>	<p>Input filename : medium_4.txt ALBANIA</p> <p>GERMANY</p> <p>NETHERLANDS SDNALREHTEN</p> <p>3557 ms No. of Comparison : 14626 Found : 50/50</p>

e. Large 1 (8223ms / 34120 Comparison)

Input	Hasil	
<p>PJHMH8OXNAOVMSPWJRYWEYCCUQJINIJIEA NISAHAADICQBSJQTEKTVGVIJLOSJ3KXYVM OPHYZYYYRDIQKRISRQFYTPCOAVOKEPUADE B0BSIAARFVRVQUASXEXYTL00S5DXXRVGI SHNCQMAOCPSYAHISHTAKER0M000IUECUS UY00ZLYOORVOMASOIBNQVHVHVED0UULZRH GVFYUXDPTRIINMNUQOLAHGNKISG0RNLTEIG KFFMAIAGT0NTATP4SECEGBJ0H0BARVLTIC0 GSLBZMHCYAPRESISPTHYVEYQKIHOLPTUHA0 MNBXAJECCTKSGMMRGIA0ZKEAJ0SLFOAHAPT QUMTHSGEAFGRUWCFNDYJUMHINQ0DENSKIG HIZBHIQURHBVNRHRENZKXNYVYTVIVIIAEX IENOUAQJTEUATFEXMQITKFKLAZMACXXUM KSHFYIYNDSERYFHOKACBELYIN0PRJULT QEZISKYJENJGAMKDTMZZEZFBDAKCFPAFXN RQTOISKNSOFFQSNUTHFVNNPADMZQPTZKN JAHUSKXGZFMQLSQ0PHATTAGCCITANXKOT HFEUYUNDUSN7ZRRMNGAED0PTMNCHEAUKCN SKLJWEIXSKNDIEENALYU0G0EAD0VMYHG IRACUSJITIANPNTTEKQ0GAGFYV8ZIT00K0ET PEQNCXUVNRLCIBSUAT0UEH2BZRENTQD3GI HYHOABAFGERATZCKLXXIAERUPVCAUKAC USWJYFAASS0BAUSTISYK0PZ000CK0ENNAK SVPTXBRISDSCSARKNARITATATISIHINHANDE JLOEKAATOCHEMTFFEL00UREL00RHYSTCCAT QPHRCQ08YICRATITKANFFUKUMITABUKBYH QDQWIEB0NAVJ3NEEUQCMORIJEMB0PCYJJI JMIZCRANEDONTHSUKABARUKASTAATVYSOL NARTFALCON0M0H0AD0P0HATTOSTLCC QV5AMFIMYORBORONNEZKACVIGF3VYSCAM H00LISRVFEIATVTIPAJLXCAQ0H0ENKHNVP SPTMPOH0E7ZNYT5S0F0JAFH0Q0EP0P WQNETSV3RYKEV00RGRJAN0YITC0LOGYRA PJJACN0KQ0GFG0B0FVETSIMINFLASHLXXME</p> <p>AGNESDIGITAL AGNESTACHON AGNESONE ROMANVAVIDE CLARENCHEW DANESCARLET EESHIMPLASH EELCONOPASA FENDITION FUTIKISEKI GOLDICITY GOLDIOP GRASSHORDER H000J0000 H000000000</p>		

f. Large 2 (3223ms / 13785 Comparison)

Input	Hasil	
<p>ZBEYDAJNYSPEAAEFNH0EEIFSEBSNCOR LVZB0J3CLRSQCLNAN0SS5TVFCU0NNRC RIGILKENTUMSRNMBYRYUREOPNRPYK KTZAGSYTECCFUIEEYACHU0AMRNCI XUQTHIP0UXZAXLHFAHBM0NUH0Q0YCD WLGJQLTISDYZARLFPFARQ0VPTIQ H0XY6ETRFV0QVHGADELWCXPFZGVYST KPAETQTOKQFANRMFNGTATQNDQKMLA SJOENAVBZCQQTORTKYVOLCWEZGAUJAP EZ0CLBNFH0HNF0B0M0MSIAXJ0NVEY JHDL6NYF0GTHBKTBROVIOUDRBEXLSJ GXEVBCUJJTXQZJABMMARHC6FUZQLUZL BBSRWDTERUDORAMD0EXTMYD0SAVHDK SERATMAELNATM0KIG0MYAKJLKYLZM KXSGUUMJCCSX0YH0MS0N0R50YNPPIYS MVEQBWHFFUGTVLEGI0DFWENUREXVUXS UMKLPHYABAZEENOC0RUSIGEQKUNT0VH VLP0QDF0UUGI0GYTRAMTAFANUICP0R SAMYDQX0HAKNAPTQTNCTULEYYSLZNNU UURTQF0LXKHDLZKFUXILUDQB0JOPERM ICQCPH0STHADAB0QNYPSAQTVPTJUN SS0CTE0Q0VZNEG0SQTHSXA0IUAHSG IYBGBUAFHMHATGAJMRGRPEXTHVVSNL SQSAZARZVPNNQSHRSLNXPMD0ADPXHW CNRJLTJUT0J0JLSRMPAPWPL0RLL0ZR W0MNP1RDSMNNCKRRE0DDKKK0FTR0RX NGAUMFYUWQLCSWDBAHC0PREJVP5LBAR UILDHUVXSVVQIZPAAZWK3YQQXHADEN RDHUKKTSF0UMYRQRK0ZV0RGUMHUY0G KMFHZFWZQFHZF0H0GEP0PXT0I0H0G0</p> <p>ACRUX ADHARA ALDEBARAN ALNIAIR ALNILLAN ALTAR ANTARES ARCTURUS BELLATRIX BETELGEUSE CANOPUS CAPELLA</p>	<p>Input filename : large_2.txt ACRUX</p> 	

g. Small 1 (954ms / 3996 Comparison)

[illegible]

h. Small 2 (298ms / 1332 Comparison)

Input		Hasil
I P J B B U Z P J J E G V B L C L V C I A V Q O E N X V C O Q Z S T L B Z Z S U D F Q G N A R E G N A T K D A W K U K A L H A V W P O N M N X E K G C A O Q O C P A Q P B A M E D A N X V D E B E U J R T N Z B I I B C D K Z P L Y K Z B V R Z C T A M Z L S M I B A Y R H F N T Z X L T O C A L T B O B K M M J V F Q L D I W E A G J J S L E F I B O T Y R E X O D Z M V Q G E R N U V Y F X B Q V I	Input filename : small_1.txt BALI - - - - B - - - - - A - - - - - L - - - - - I - BANDUNG G N U D N A B	- S - o - - L - - o - - - - - - TANGERANG - G N A R E G N A T - 298 ms No. of Comparison : 1332 Found : 10/10

D. SOURCE PROGRAM

Link Github : https://github.com/Audino723/Tucil1_13520088

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)	√	
2. Program berhasil running	√	
3. Program dapat membaca file masukan dan menuliskan luaran	√	
4. Program berhasil menemukan semua kata di dalam puzzle	√	