Laporan Tugas Kecil
Algoritma Brute Force untuk Pencocokan Kata pada Papan Kata

Maria Khelli
K01 / 13520115

I. Spesifikasi Program

Bahasa yang digunakan: Java

Tautan GitHub: https://github.com/khelli07/IF2211-crossword-brute-force

Daftar kelas yang dibuat:

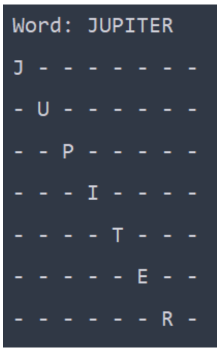
1. ColoredChar, sebagai konten dari matriks.
2. Coordinate, sebagai koordinat (baris, kolom) di matriks.
3. Matrix, representasi papan kata.
4. CrossWord, menyimpan papan kata dan daftar kata yang dicari.
5. Main, digunakan untuk menjalankan program utama.

Keberjalanan program:

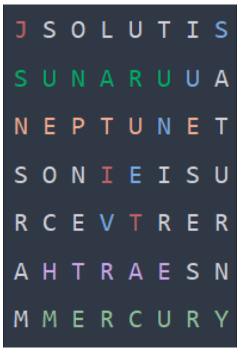
Poin	Ya	Tidak
Program berhasil dikompilasi	✓	
Program berhasil dijalankan	✓	
Program dapat membaca file dan menuliskan keluaran	✓	
Program berhasil menemukan semua kata dalam puzzle	✓	

II. Alur Algoritma Brute Force

1. Pertama, program akan menerima masukan String dari pengguna yang merupakan nama *file*.
2. Kemudian, program membaca *file*. Di dalam *file* masukan, papan kata yang dibaca akan disimpan di dalam Matrix. Lalu, diikuti baris kosong sebagai batas antara papan dan daftar kata. Selanjutnya, program membaca daftar kata (secara unik) yang disimpan dalam sebuah array.
3. Pada method searchWord yang dimiliki CrossWord, pencocokan papan kata dan daftar kata dilakukan. Sebelum itu, program meminta input pengguna, yaitu untuk memilih apakah menginginkan keluaran per kata atau seluruh kata seperti gambar berikut.



```
Word: JUPITER
J - - - - - - -
- U - - - - - -
- - P - - - - -
- - - I - - - -
- - - - T - - -
- - - - - E - -
- - - - - - R -
```



```
J S O L U T I S
S U N A R U U A
N E P T U N E T
S O N I E I S U
R C E V T R E R
A H T R A E S N
M M E R C U R Y
```

4. Kemudian, algoritma pencarian dijalankan. Program mengiterasi setiap kata.
5. Untuk setiap kata, program mengunjungi huruf-huruf matriks secara terurut, menguji kesamaan antara huruf yang dikunjungi dan huruf awal kata yang sedang diiterasi.

6. Jika sama, program akan masuk ke sebuah blok kondisi *if*. Di dalam blok *if* ini, program mengecek di sekeliling huruf pertama yang ditemukan (8 mata angin, dengan syarat indeks yang dicek masih valid). Misalkan pada gambar di atas, program ingin mencari kata "JUPITER". Artinya, ketika program bertemu huruf 'J' pertama, program akan melihat apakah 'S', 'U', 'S' sama dengan 'U' (huruf kedua "JUPITER").
7. Jika hurufnya sesuai dan estimasi indeks terakhir kata masih berada di dalam Matrix (*i.e.* belum *out of bounds*), maka arah (direpresentasikan dengan enum Direction) akan dimasukkan ke dalam array.
8. Kemudian, program akan mengiterasi ke setiap arah yang sesuai kriteria nomor 7.
9. Untuk setiap arah, apabila huruf yang ditemukan cocok sampai akhir, program akan memberi nilai warna pada karakter (jika dipilih berwarna) atau mencetak kata tersebut di dalam matriks (jika dipilih tidak berwarna). Apabila berbeda (*i.e.* tidak cocok sampai akhir), program akan mereset dan kembali ke awal huruf 'J' untuk mengiterasi arah berikutnya.
10. Jika telah ditemukan satu kata yang bersangkutan, program akan lanjut ke iterasi kata berikutnya.
11. Setelah selesai mengiterasi seluruh kata, dihitung waktu eksekusi, jumlah perbandingan yang dilakukan, dan banyak kata yang ditemukan. Data tersebut dicetak sebagai keluaran.

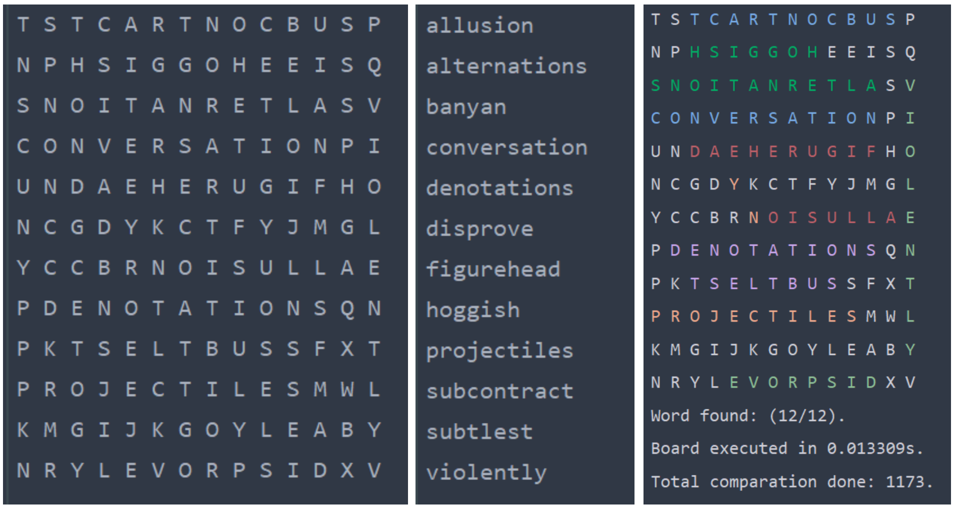
III. Sumber Kode Program

(Terlampir)

IV. Gambar Tangkapan Layar

Format (dari kiri ke kanan): Papan kata, daftar kata, keluaran program.

a. Ukuran 14 × 12 (Kecil)

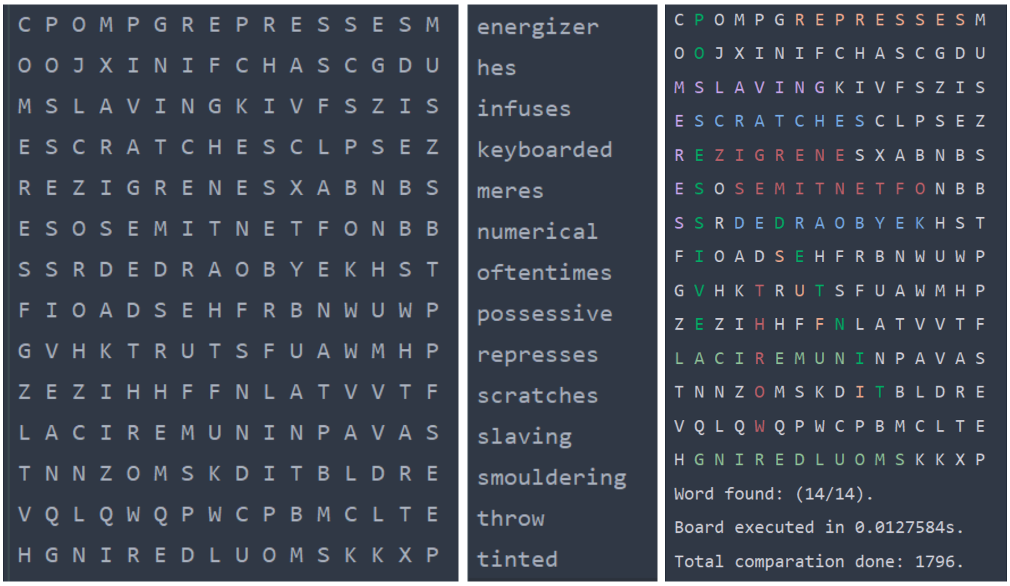


```
T S T C A R T N O C B U S P
N P H S I G G O H E E I S Q
S N O I T A N R E T L A S V
C O N V E R S A T I O N P I
U N D A E H E R U G I F H O
N C G D Y K C T F Y J M G L
Y C C B R N O I S U L L A E
P D E N O T A T I O N S Q N
P K T S E L T B U S S F X T
P R O J E C T I L E S M W L
K M G I J K G O Y L E A B Y
N R Y L E V O R P S I D X V
```

```
allusion
alternations
banyan
conversation
denotations
disprove
figurehead
hoggish
projectiles
subcontract
subtlest
violently
```

```
T S T C A R T N O C B U S P
N P H S I G G O H E E I S Q
S N O I T A N R E T L A S V
C O N V E R S A T I O N P I
U N D A E H E R U G I F H O
N C G D Y K C T F Y J M G L
Y C C B R N O I S U L L A E
P D E N O T A T I O N S Q N
P K T S E L T B U S S F X T
P R O J E C T I L E S M W L
K M G I J K G O Y L E A B Y
N R Y L E V O R P S I D X V
Word found: (12/12).
Board executed in 0.013309s.
Total comparation done: 1173.
```

b. Ukuran 16 × 14 (Kecil)

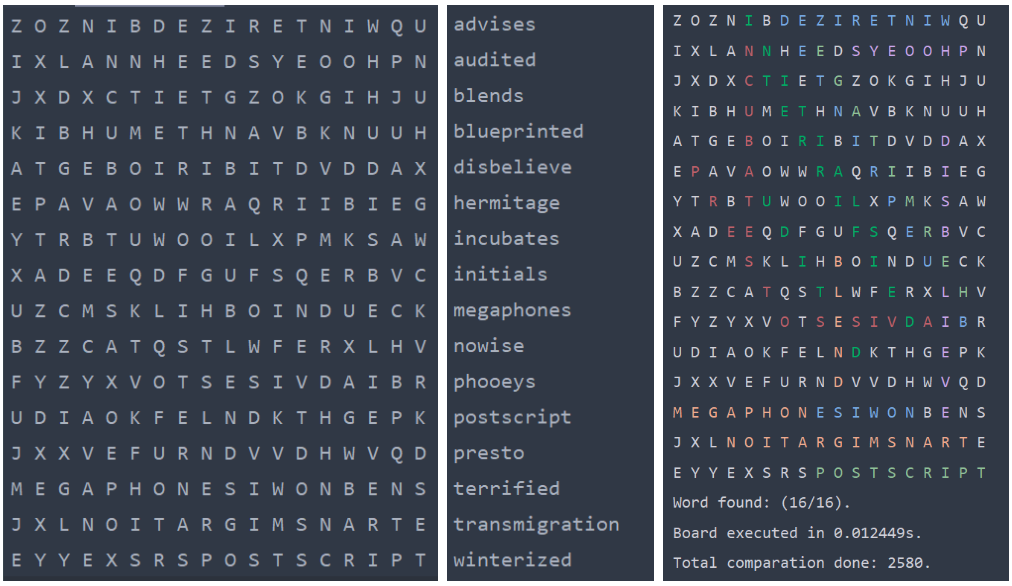


```
C P O M P G R E P R E S S E S M
O O J X I N I F C H A S C G D U
M S L A V I N G K I V F S Z I S
E S C R A T C H E S C L P S E Z
R E Z I G R E N E S X A B N B S
E S O S E M I T N E T F O N B B
S S R D E D R A O B Y E K H S T
F I O A D S E H F R B N W U W P
G V H K T R U T S F U A W M H P
Z E Z I H H F F N L A T V V T F
L A C I R E M U N I N P A V A S
T N N Z O M S K D I T B L D R E
V Q L Q W Q P W C P B M C L T E
H G N I R E D L U O M S K K X P
```

```
energizer
hes
infuses
keyboarded
meres
numerical
oftentimes
possessive
represses
scratches
slaving
smouldering
throw
tinted
```

```
C P O M P G R E P R E S S E S M
O O J X I N I F C H A S C G D U
M S L A V I N G K I V F S Z I S
E S C R A T C H E S C L P S E Z
R E Z I G R E N E S X A B N B S
E S O S E M I T N E T F O N B B
S S R D E D R A O B Y E K H S T
F I O A D S E H F R B N W U W P
G V H K T R U T S F U A W M H P
Z E Z I H H F F N L A T V V T F
L A C I R E M U N I N P A V A S
T N N Z O M S K D I T B L D R E
V Q L Q W Q P W C P B M C L T E
H G N I R E D L U O M S K K X P
Word found: (14/14).
Board executed in 0.0127584s.
Total comparation done: 1796.
```

c. Ukuran 18 × 16 (Kecil)

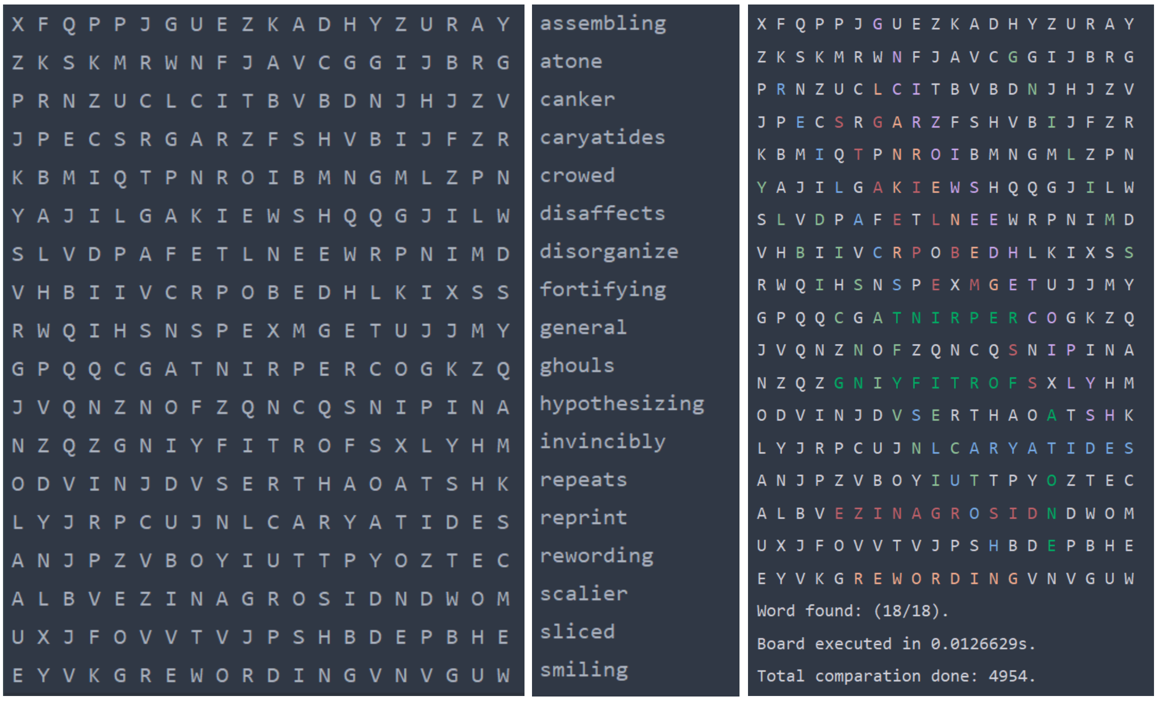


```
Z O Z N I B D E Z I R E T N I W Q U
I X L A N N H E E D S Y E O O H P N
J X D X C T I E T G Z O K G I H J U
K I B H U M E T H N A V B K N U U H
A T G E B O I R I B I T D V D D A X
E P A V A O W W R A Q R I I B I E G
Y T R B T U W O O I L X P M K S A W
X A D E E Q D F G U F S Q E R B V C
U Z C M S K L I H B O I N D U E C K
B Z Z C A T Q S T L W F E R X L H V
F Y Z Y X V O T S E S I V D A I B R
U D I A O K F E L N D K T H G E P K
J X X V E F U R N D V V D H W V Q D
M E G A P H O N E S I W O N B E N S
J X L N O I T A R G I M S N A R T E
E Y Y E X S R S P O S T S C R I P T
```

```
advises
audited
blends
blueprinted
disbelieve
hermitage
incubates
initials
megaphones
nowise
phooeys
postscript
presto
terrified
transmigration
winterized
```

```
Z O Z N I B D E Z I R E T N I W Q U
I X L A N N H E E D S Y E O O H P N
J X D X C T I E T G Z O K G I H J U
K I B H U M E T H N A V B K N U U H
A T G E B O I R I B I T D V D D A X
E P A V A O W W R A Q R I I B I E G
Y T R B T U W O O I L X P M K S A W
X A D E E Q D F G U F S Q E R B V C
U Z C M S K L I H B O I N D U E C K
B Z Z C A T Q S T L W F E R X L H V
F Y Z Y X V O T S E S I V D A I B R
U D I A O K F E L N D K T H G E P K
J X X V E F U R N D V V D H W V Q D
M E G A P H O N E S I W O N B E N S
J X L N O I T A R G I M S N A R T E
E Y Y E X S R S P O S T S C R I P T
Word found: (16/16).
Board executed in 0.012449s.
Total comparation done: 2580.
```

d. Ukuran 20 × 18 (Sedang)

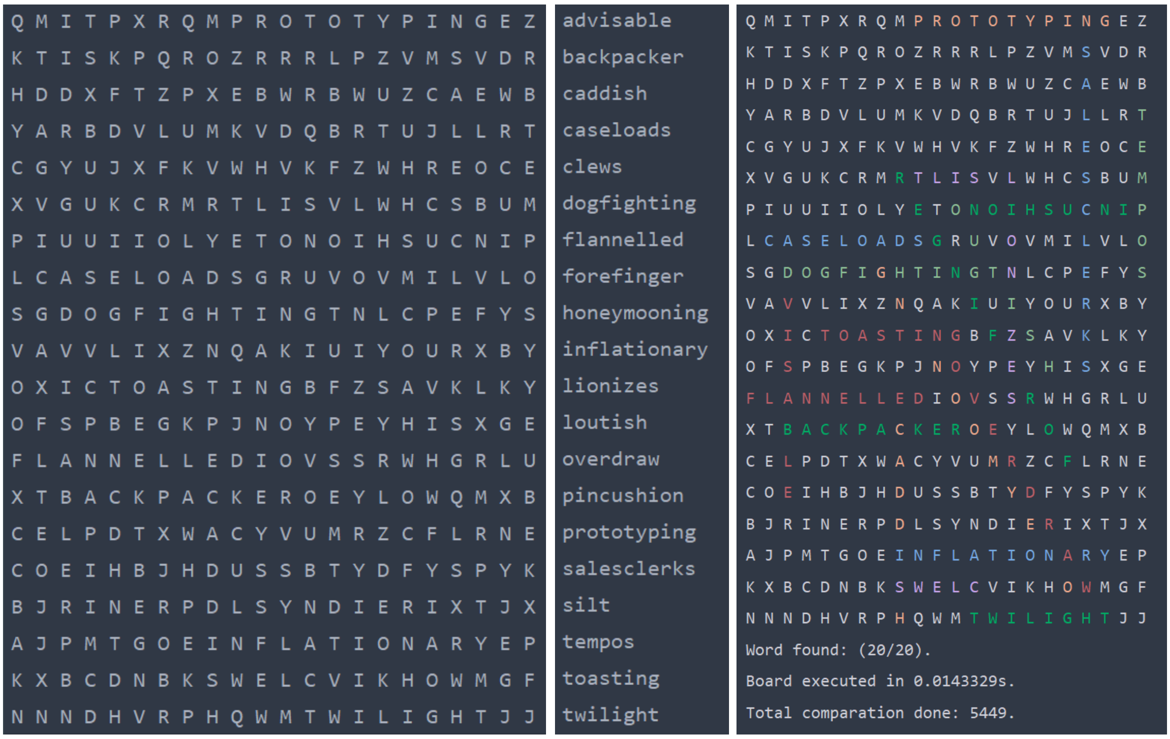


```
X F Q P P J G U E Z K A D H Y Z U R A Y
Z K S K M R W N F J A V C G G I J B R G
P R N Z U C L C I T B V B D N J H J Z V
J P E C S R G A R Z F S H V B I J F Z R
K B M I Q T P N R O I B M N G M L Z P N
Y A J I L G A K I E W S H Q Q G J I L W
S L V D P A F E T L N E E W R P N I M D
V H B I I V C R P O B E D H L K I X S S
R W Q I H S N S P E X M G E T U J J M Y
G P Q Q C G A T N I R P E R C O G K Z Q
J V Q N Z N O F Z Q N C Q S N I P I N A
N Z Q Z G N I Y F I T R O F S X L Y H M
O D V I N J D V S E R T H A O A T S H K
L Y J R P C U J N L C A R Y A T I D E S
A N J P Z V B O Y I U T T P Y O Z T E C
A L B V E Z I N A G R O S I D N D W O M
U X J F O V V T V J P S H B D E P B H E
E Y V K G R E W O R D I N G V N V G U W
```

```
assembling
atone
canker
caryatides
crowed
disaffects
disorganize
fortifying
general
ghouls
hypothesizing
invincibly
repeats
reprint
rewording
scalier
sliced
smiling
```

```
X F Q P P J G U E Z K A D H Y Z U R A Y
Z K S K M R W N F J A V C G G I J B R G
P R N Z U C L C I T B V B D N J H J Z V
J P E C S R G A R Z F S H V B I J F Z R
K B M I Q T P N R O I B M N G M L Z P N
Y A J I L G A K I E W S H Q Q G J I L W
S L V D P A F E T L N E E W R P N I M D
V H B I I V C R P O B E D H L K I X S S
R W Q I H S N S P E X M G E T U J J M Y
G P Q Q C G A T N I R P E R C O G K Z Q
J V Q N Z N O F Z Q N C Q S N I P I N A
N Z Q Z G N I Y F I T R O F S X L Y H M
O D V I N J D V S E R T H A O A T S H K
L Y J R P C U J N L C A R Y A T I D E S
A N J P Z V B O Y I U T T P Y O Z T E C
A L B V E Z I N A G R O S I D N D W O M
U X J F O V V T V J P S H B D E P B H E
E Y V K G R E W O R D I N G V N V G U W
Word found: (18/18).
Board executed in 0.0126629s.
Total comparation done: 4954.
```

e. Ukuran 22 × 20 (Sedang)

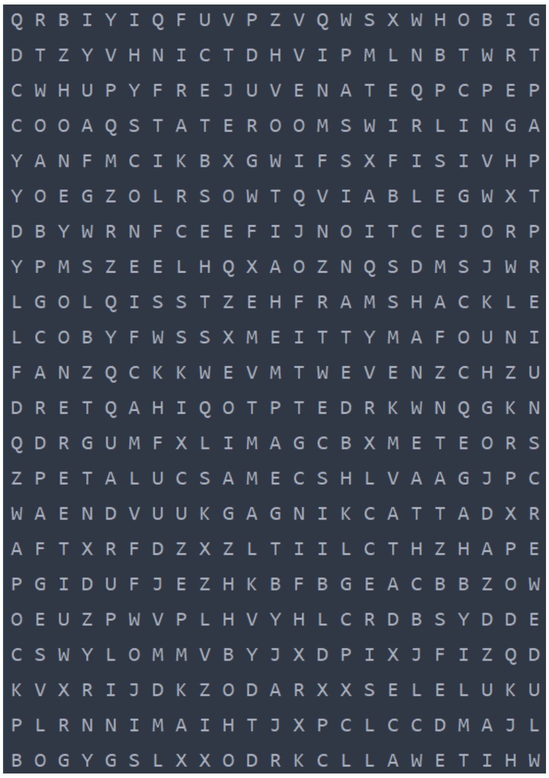


```
Q M I T P X R Q M P R O T O T Y P I N G E Z
K T I S K P Q R O Z R R R L P Z V M S V D R
H D D X F T Z P X E B W R B W U Z C A E W B
Y A R B D V L U M K V D Q B R T U J L L R T
C G Y U J X F K V W H V K F Z W H R E O C E
X V G U K C R M R T L I S V L W H C S B U M
P I U U I I O L Y E T O N O I H S U C N I P
L C A S E L O A D S G R U V O V M I L V L O
S G D O G F I G H T I N G T N L C P E F Y S
V A V V L I X Z N Q A K I U I Y O U R X B Y
O X I C T O A S T I N G B F Z S A V K L K Y
O F S P B E G K P J N O Y P E Y H I S X G E
F L A N N E L L E D I O V S S R W H G R L U
X T B A C K P A C K E R O E Y L O W Q M X B
C E L P D T X W A C Y V U M R Z C F L R N E
C O E I H B J H D U S S B T Y D F Y S P Y K
B J R I N E R P D L S Y N D I E R I X T J X
A J P M T G O E I N F L A T I O N A R Y E P
K X B C D N B K S W E L C V I K H O W M G F
N N N D H V R P H Q W M T W I L I G H T J J
```

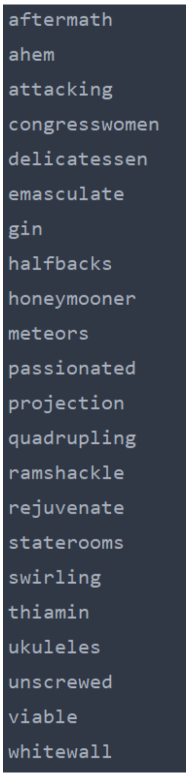
```
advisable
backpacker
caddish
caseloads
clews
dogfighting
flannelled
forefinger
honeymooning
inflationary
lionizes
loutish
overdraw
pincushion
prototyping
salesclerks
silt
tempos
toasting
twilight
```

```
Q M I T P X R Q M P R O T O T Y P I N G E Z
K T I S K P Q R O Z R R R L P Z V M S V D R
H D D X F T Z P X E B W R B W U Z C A E W B
Y A R B D V L U M K V D Q B R T U J L L R T
C G Y U J X F K V W H V K F Z W H R E O C E
X V G U K C R M R T L I S V L W H C S B U M
P I U U I I O L Y E T O N O I H S U C N I P
L C A S E L O A D S G R U V O V M I L V L O
S G D O G F I G H T I N G T N L C P E F Y S
V A V V L I X Z N Q A K I U I Y O U R X B Y
O X I C T O A S T I N G B F Z S A V K L K Y
O F S P B E G K P J N O Y P E Y H I S X G E
F L A N N E L L E D I O V S S R W H G R L U
X T B A C K P A C K E R O E Y L O W Q M X B
C E L P D T X W A C Y V U M R Z C F L R N E
C O E I H B J H D U S S B T Y D F Y S P Y K
B J R I N E R P D L S Y N D I E R I X T J X
A J P M T G O E I N F L A T I O N A R Y E P
K X B C D N B K S W E L C V I K H O W M G F
N N N D H V R P H Q W M T W I L I G H T J J
Word found: (20/20).
Board executed in 0.0143329s.
Total comparation done: 5449.
```

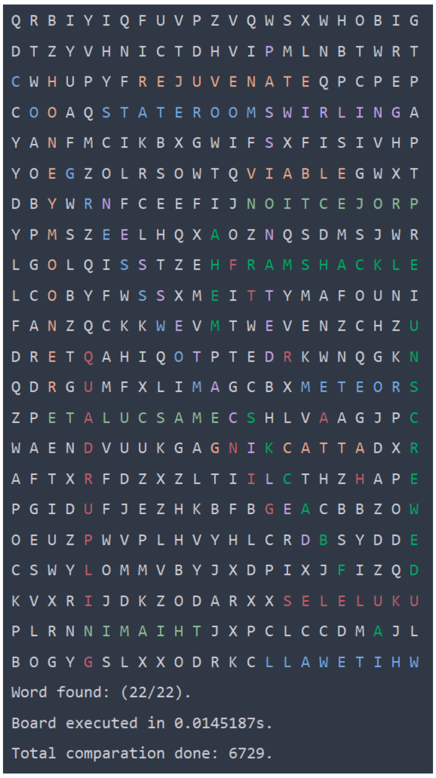
f. Ukuran 24 × 22 (Sedang)



```
Q R B I Y I Q F U V P Z V Q W S X W H O B I G
D T Z Y V H N I C T D H V I P M L N B T W R T
C W H U P Y F R E J U V E N A T E Q P C P E P
C O O A Q S T A T E R O O M S W I R L I N G A
Y A N F M C I K B X G W I F S X F I S I V H P
Y O E G Z O L R S O W T Q V I A B L E G W X T
D B Y W R N F C E E F I J N O I T C E J O R P
Y P M S Z E E L H Q X A O Z N Q S D M S J W R
L G O L Q I S S T Z E H F R A M S H A C K L E
L C O B Y F W S S X M E I T T Y M A F O U N I
F A N Z Q C K K W E V M T W E V E N Z C H Z U
D R E T Q A H I Q O T P T E D R K W N Q G K N
Q D R G U M F X L I M A G C B X M E T E O R S
Z P E T A L U C S A M E C S H L V A A G J P C
W A E N D V U U K G A G N I K C A T T A D X R
A F T X R F D Z X Z L T I I L C T H Z H A P E
P G I D U F J E Z H K B F B G E A C B B Z O W
O E U Z P W V P L H V Y H L C R D B S Y D D E
C S W Y L O M M V B Y J X D P I X J F I Z Q D
K V X R I J D K Z O D A R X X S E L E L U K U
P L R N N I M A I H T J X P C L C C D M A J L
B O G Y G S L X X O D R K C L L A W E T I H W
```

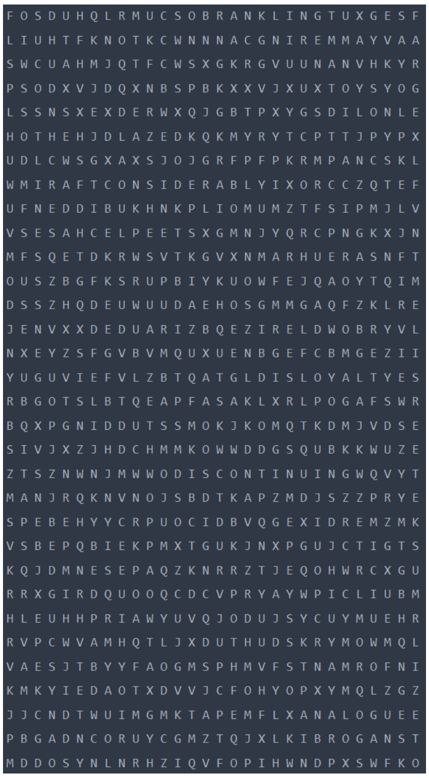


```
aftermath
ahem
attacking
congresswomen
delicatessen
emasculate
gin
halfbacks
honeymooner
meteors
passionated
projection
quadrupling
ramshackle
rejuvenate
staterooms
swirling
thiamin
ukuleles
unscrewed
viable
whitewall
```

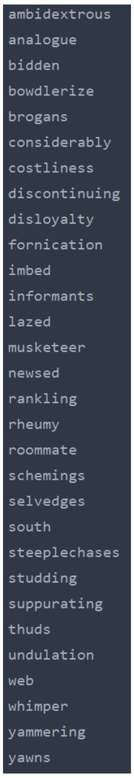


```
Q R B I Y I Q F U V P Z V Q W S X W H O B I G
D T Z Y V H N I C T D H V I P M L N B T W R T
C W H U P Y F R E J U V E N A T E Q P C P E P
C O O A Q S T A T E R O O M S W I R L I N G A
Y A N F M C I K B X G W I F S X F I S I V H P
Y O E G Z O L R S O W T Q V I A B L E G W X T
D B Y W R N F C E E F I J N O I T C E J O R P
Y P M S Z E E L H Q X A O Z N Q S D M S J W R
L G O L Q I S S T Z E H F R A M S H A C K L E
L C O B Y F W S S X M E I T T Y M A F O U N I
F A N Z Q C K K W E V M T W E V E N Z C H Z U
D R E T Q A H I Q O T P T E D R K W N Q G K N
Q D R G U M F X L I M A G C B X M E T E O R S
Z P E T A L U C S A M E C S H L V A A G J P C
W A E N D V U U K G A G N I K C A T T A D X R
A F T X R F D Z X Z L T I I L C T H Z H A P E
P G I D U F J E Z H K B F B G E A C B B Z O W
O E U Z P W V P L H V Y H L C R D B S Y D D E
C S W Y L O M M V B Y J X D P I X J F I Z Q D
K V X R I J D K Z O D A R X X S E L E L U K U
P L R N N I M A I H T J X P C L C C D M A J L
B O G Y G S L X X O D R K C L L A W E T I H W
Word found: (22/22).
Board executed in 0.0145187s.
Total comparation done: 6729.
```

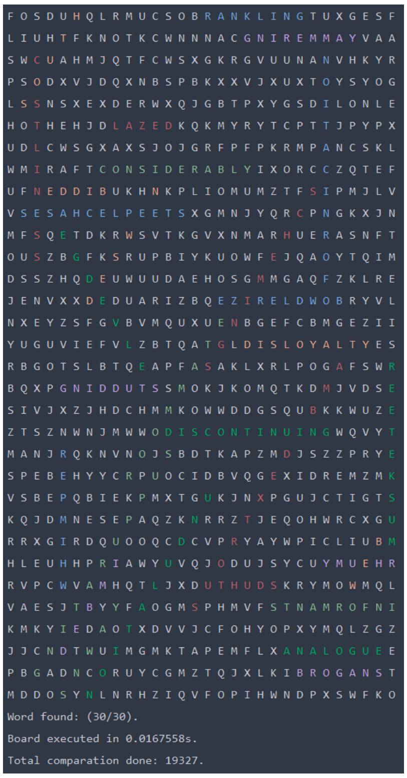
g. Ukuran 32 × 30 (Besar)



```
F O S D U H Q L R M U C S O B R A N K L I N G T U X G E S F
L I U H T F K N O T K C W N N N A C G N I R E M M A Y V A A
S W C U A H M J Q T F C W S X G K R G V U U N A N V H K Y R
P S O D X V J D Q X N B S P B K X X V J X U X T O Y S Y O G
L S S N S X E X D E R W X Q J G B T P X Y G S D I L O N L E
H O T H E H J D L A Z E D K Q K M Y R Y T C P T T J P Y P X
U D L C W S G X A X S J O J G R F P F P K R M P A N C S K L
W M I R A F T C O N S I D E R A B L Y I X O R C C Z Q T E F
U F N E D D I B U K H N K P L I O M U M Z T F S I P M J L V
V S E S A H C E L P E E T S X G M N J Y Q R C P N G K X J N
M F S Q E T D K R W S V T K G V X N M A R H U E R A S N F T
O U S Z B G F K S R U P B I Y K U O W F E J Q A O Y T Q I M
D S S Z H Q D E U W U U D A E H O S G M M G A Q F Z K L R E
J E N V X X D E D U A R I Z B Q E Z I R E L D W O B R Y V L
N X E Y Z S F G V B V M Q U X U E N B G E F C B M G E Z I I
Y U G U V I E F V L Z B T Q A T G L D I S L O Y A L T Y E S
R B G O T S L B T Q E A P F A S A K L X R L P O G A F S W R
B Q X P G N I D D U T S S M O K J K O M Q T K D M J V D S E
S I V J X Z J H D C H M M K O W W D D G S Q U B K K W U Z E
Z T S Z N W N J M W W O D I S C O N T I N U I N G W Q V Y T
M A N J R Q K N V N O J S B D T K A P Z M D J S Z Z P R Y E
S P E B E H Y Y C R P U O C I D B V Q G E X I D R E M Z M K
V S B E P Q B I E K P M X T G U K J N X P G U J C T I G T S
K Q J D M N E S E P A Q Z K N R R Z T J E Q O H W R C X G U
R R X G I R D Q U O O Q C D C V P R Y A Y W P I C L I U B M
H L E U H H P R I A W Y U V Q J O D U J S Y C U Y M U E H R
R V P C W V A M H Q T L J X D U T H U D S K R Y M O W M Q L
V A E S J T B Y Y F A O G M S P H M V F S T N A M R O F N I
K M K Y I E D A O T X D V V J C F O H Y O P X Y M Q L Z G Z
J J C N D T W U I M G M K T A P E M F L X A N A L O G U E E
P B G A D N C O R U Y C G M Z T Q J X L K I B R O G A N S T
M D D O S Y N L N R H Z I Q V F O P I H W N D P X S W F K O
```

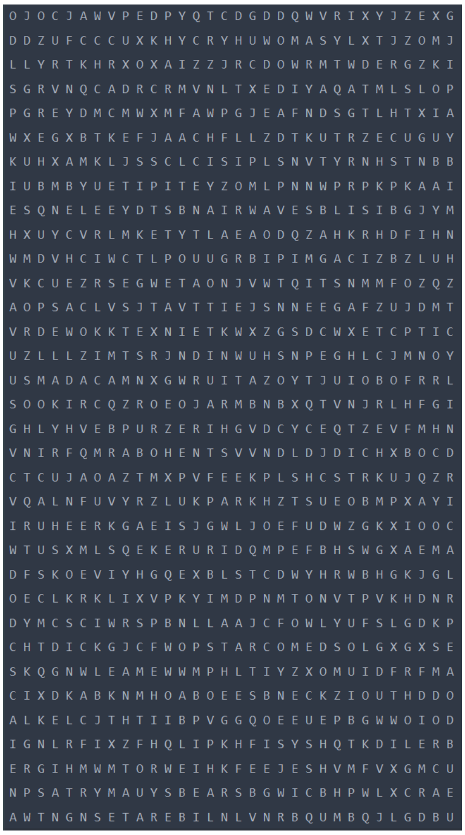


```
ambidextrous
analogue
bidden
bowdlerize
brogans
considerably
costliness
discontinuing
disloyalty
fornication
imbed
informants
lazed
musketeer
newsed
rankling
rheumy
roommate
schemings
selvedges
south
steeplechases
studding
suppurating
thuds
undulation
web
whimper
yammering
yawns
```

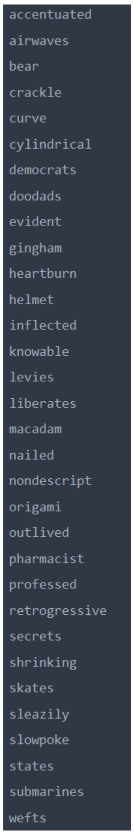


```
F O S D U H Q L R M U C S O B R A N K L I N G T U X G E S F
L I U H T F K N O T K C W N N N A C G N I R E M M A Y V A A
S W C U A H M J Q T F C W S X G K R G V U U N A N V H K Y R
P S O D X V J D Q X N B S P B K X X V J X U X T O Y S Y O G
L S S N S X E X D E R W X Q J G B T P X Y G S D I L O N L E
H O T H E H J D L A Z E D K Q K M Y R Y T C P T T J P Y P X
U D L C W S G X A X S J O J G R F P F P K R M P A N C S K L
W M I R A F T C O N S I D E R A B L Y I X O R C C Z Q T E F
U F N E D D I B U K H N K P L I O M U M Z T F S I P M J L V
V S E S A H C E L P E E T S X G M N J Y Q R C P N G K X J N
M F S Q E T D K R W S V T K G V X N M A R H U E R A S N F T
O U S Z B G F K S R U P B I Y K U O W F E J Q A O Y T Q I M
D S S Z H Q D E U W U U D A E H O S G M M G A Q F Z K L R E
J E N V X X D E D U A R I Z B Q E Z I R E L D W O B R Y V L
N X E Y Z S F G V B V M Q U X U E N B G E F C B M G E Z I I
Y U G U V I E F V L Z B T Q A T G L D I S L O Y A L T Y E S
R B G O T S L B T Q E A P F A S A K L X R L P O G A F S W R
B Q X P G N I D D U T S S M O K J K O M Q T K D M J V D S E
S I V J X Z J H D C H M M K O W W D D G S Q U B K K W U Z E
Z T S Z N W N J M W W O D I S C O N T I N U I N G W Q V Y T
M A N J R Q K N V N O J S B D T K A P Z M D J S Z Z P R Y E
S P E B E H Y Y C R P U O C I D B V Q G E X I D R E M Z M K
V S B E P Q B I E K P M X T G U K J N X P G U J C T I G T S
K Q J D M N E S E P A Q Z K N R R Z T J E Q O H W R C X G U
R R X G I R D Q U O O Q C D C V P R Y A Y W P I C L I U B M
H L E U H H P R I A W Y U V Q J O D U J S Y C U Y M U E H R
R V P C W V A M H Q T L J X D U T H U D S K R Y M O W M Q L
V A E S J T B Y Y F A O G M S P H M V F S T N A M R O F N I
K M K Y I E D A O T X D V V J C F O H Y O P X Y M Q L Z G Z
J J C N D T W U I M G M K T A P E M F L X A N A L O G U E E
P B G A D N C O R U Y C G M Z T Q J X L K I B R O G A N S T
M D D O S Y N L N R H Z I Q V F O P I H W N D P X S W F K O
Word found: (30/30).
Board executed in 0.0167558s.
Total comparation done: 19327.
```

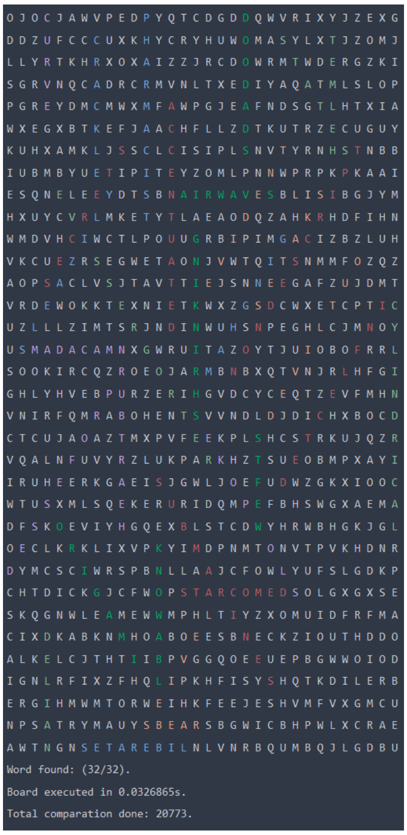
h. Ukuran 34 × 32 (Besar)



```
O J O C J A W V P E D P Y Q T C D G D D Q W V R I X Y J Z E X G
D D Z U F C C C U X K H Y C R Y H U W O M A S Y L X T J Z O M J
L L Y R T K H R X O X A I Z Z J R C D O W R M T W D E R G Z K I
S G R V N Q C A D R C R M V N L T X E D I Y A Q A T M L S L O P
P G R E Y D M C M W X M F A W P G J E A F N D S G T L H T X I A
W X E G X B T K E F J A A C H F L L Z D T K U T R Z E C U G U Y
K U H X A M K L J S S C L C I S I P L S N V T Y R N H S T N B B
I U B M B Y U E T I P I T E Y Z O M L P N N W P R P K P K A A I
E S Q N E L E E Y D T S B N A I R W A V E S B L I S I B G J Y M
H X U Y C V R L M K E T Y T L A E A O D Q Z A H K R H D F I H N
W M D V H C I W C T L P O U U G R B I P I M G A C I Z B Z L U H
V K C U E Z R S E G W E T A O N J V W T Q I T S N M M F O Z Q Z
A O P S A C L V S J T A V T T I E J S N N E E G A F Z U J D M T
V R D E W O K K T E X N I E T K W X Z G S D C W X E T C P T I C
U Z L L L Z I M T S R J N D I N W U H S N P E G H L C J M N O Y
U S M A D A C A M N X G W R U I T A Z O Y T J U I O B O F R R L
S O O K I R C Q Z R O E O J A R M B N B X Q T V N J R L H F G I
G H L Y H V E B P U R Z E R I H G V D C Y C E Q T Z E V F M H N
V N I R F Q M R A B O H E N T S V V N D L D J D I C H X B O C D
C T C U J A O A Z T M X P V F E E K P L S H C S T R K U J Q Z R
V Q A L N F U V Y R Z L U K P A R K H Z T S U E O B M P X A Y I
I R U H E E R K G A E I S J G W L J O E F U D W Z G K X I O O C
W T U S X M L S Q E K E R U R I D Q M P E F B H S W G X A E M A
D F S K O E V I Y H G Q E X B L S T C D W Y H R W B H G K J G L
O E C L K R K L I X V P K Y I M D P N M T O N V T P V K H D N R
D Y M C S C I W R S P B N L L A A J C F O W L Y U F S L G D K P
C H T D I C K G J C F W O P S T A R C O M E D S O L G X G X S E
S K Q G N W L E A M E W W M P H L T I Y Z X O M U I D F R F M A
C I X D K A B K N M H O A B O E E S B N E C K Z I O U T H D D O
A L K E L C J T H T I I B P V G G Q O E E U E P B G W W O I O D
I G N L R F I X Z F H Q L I P K H F I S Y S H Q T K D I L E R B
E R G I H M W M T O R W E I H K F E E J E S H V M F V X G M C U
N P S A T R Y M A U Y S B E A R S B G W I C B H P W L X C R A E
A W T N G N S E T A R E B I L N L V N R B Q U M B Q J L G D B U
```



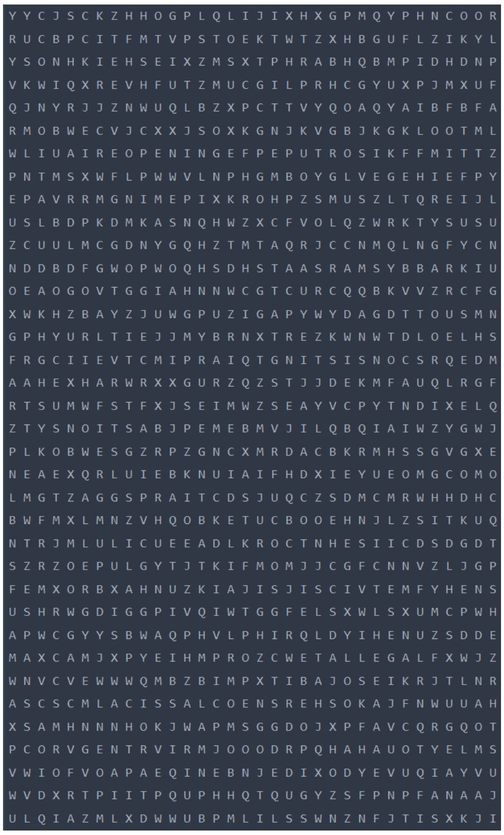
```
accentuated
airwaves
bear
crackle
curve
cylindrical
democrats
doodads
evident
gingham
heartburn
helmet
inflected
knowable
levies
liberates
macadam
nailed
nondescript
origami
outlived
pharmacist
professed
retrogressive
secrets
shrinking
skates
sleazily
slowpoke
states
submarines
wefts
```



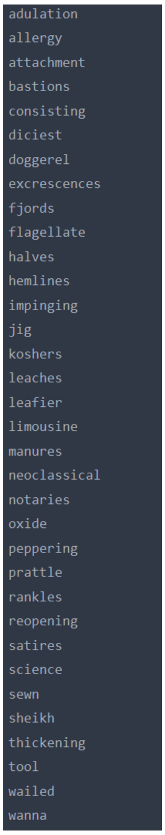
```
O J O C J A W V P E D P Y Q T C D G D D Q W V R I X Y J Z E X G
D D Z U F C C C U X K H Y C R Y H U W O M A S Y L X T J Z O M J
L L Y R T K H R X O X A I Z Z J R C D O W R M T W D E R G Z K I
S G R V N Q C A D R C R M V N L T X E D I Y A Q A T M L S L O P
P G R E Y D M C M W X M F A W P G J E A F N D S G T L H T X I A
W X E G X B T K E F J A A C H F L L Z D T K U T R Z E C U G U Y
K U H X A M K L J S S C L C I S I P L S N V T Y R N H S T N B B
I U B M B Y U E T I P I T E Y Z O M L P N N W P R P K P K A A I
E S Q N E L E E Y D T S B N A I R W A V E S B L I S I B G J Y M
H X U Y C V R L M K E T Y T L A E A O D Q Z A H K R H D F I H N
W M D V H C I W C T L P O U U G R B I P I M G A C I Z B Z L U H
V K C U E Z R S E G W E T A O N J V W T Q I T S N M M F O Z Q Z
A O P S A C L V S J T A V T T I E J S N N E E G A F Z U J D M T
V R D E W O K K T E X N I E T K W X Z G S D C W X E T C P T I C
U Z L L L Z I M T S R J N D I N W U H S N P E G H L C J M N O Y
U S M A D A C A M N X G W R U I T A Z O Y T J U I O B O F R R L
S O O K I R C Q Z R O E O J A R M B N B X Q T V N J R L H F G I
G H L Y H V E B P U R Z E R I H G V D C Y C E Q T Z E V F M H N
V N I R F Q M R A B O H E N T S V V N D L D J D I C H X B O C D
C T C U J A O A Z T M X P V F E E K P L S H C S T R K U J Q Z R
V Q A L N F U V Y R Z L U K P A R K H Z T S U E O B M P X A Y I
I R U H E E R K G A E I S J G W L J O E F U D W Z G K X I O O C
W T U S X M L S Q E K E R U R I D Q M P E F B H S W G X A E M A
D F S K O E V I Y H G Q E X B L S T C D W Y H R W B H G K J G L
O E C L K R K L I X V P K Y I M D P N M T O N V T P V K H D N R
D Y M C S C I W R S P B N L L A A J C F O W L Y U F S L G D K P
C H T D I C K G J C F W O P S T A R C O M E D S O L G X G X S E
S K Q G N W L E A M E W W M P H L T I Y Z X O M U I D F R F M A
C I X D K A B K N M H O A B O E E S B N E C K Z I O U T H D D O
A L K E L C J T H T I I B P V G G Q O E E U E P B G W W O I O D
I G N L R F I X Z F H Q L I P K H F I S Y S H Q T K D I L E R B
E R G I H M W M T O R W E I H K F E E J E S H V M F V X G M C U
N P S A T R Y M A U Y S B E A R S B G W I C B H P W L X C R A E
A W T N G N S E T A R E B I L N L V N R B Q U M B Q J L G D B U
```

```
Word found: (32/32).
Board executed in 0.0326865s.
Total comparation done: 20773.
```

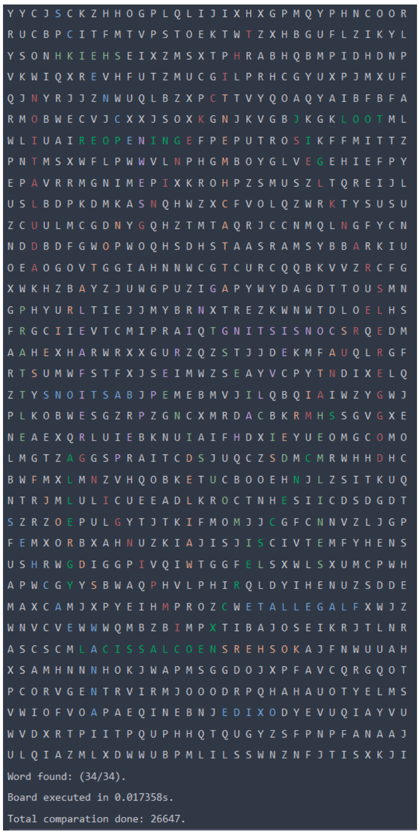
i. Ukuran 36 × 34 (Besar)



```
Y Y C J S C K Z H H O G P L Q L I J I X H X G P M Q Y P H N C O O R
R U C B P C I T F M T V P S T O E K T W T Z X H B G U F L Z I K Y L
Y S O N H K I E H S E I X Z M S X T P H R A B H Q B M P I D H D N P
V K W I Q X R E V H F U T Z M U C G I L P R H C G Y U X P J M X U F
Q J N Y R J J Z N W U Q L B Z X P C T T V Y Q O A Q Y A I B F B F A
R M O B W E C V J C X X J S O X K G N J K V G B J K G K L O O T M L
W L I U A I R E O P E N I N G E F P E P U T R O S I K F F M I T T Z
P N T M S X W F L P W W V L N P H G M B O Y G L V E G E H I E F P Y
E P A V R R M G N I M E P I X K R O H P Z S M U S Z L T Q R E I J L
U S L B D P K D M K A S N Q H W Z X C F V O L Q Z W R K T Y S U S U
Z C U U L M C G D N Y G Q H Z T M T A Q R J C C N M Q L N G F Y C N
N D D B D F G W O P W O Q H S D H S T A A S R A M S Y B B A R K I U
O E A O G O V T G G I A H N N W C G T C U R C Q Q B K V V Z R C F G
X W K H Z B A Y Z J U W G P U Z I G A P Y W Y D A G D T T O U S M N
G P H Y U R L T I E J J M Y B R N X T R E Z K W N W T D L O E L H S
F R G C I I E V T C M I P R A I Q T G N I T S I S N O C S R Q E D M
A A H E X H A R W R X X G U R Z Q Z S T J J D E K M F A U Q L R G F
R T S U M W F S T F X J S E I M W Z S E A Y V C P Y T N D I X E L Q
Z T Y S N O I T S A B J P E M E B M V J I L Q B Q I A I W Z Y G W J
P L K O B W E S G Z R P Z G N C X M R D A C B K R M H S S G V G X E
N E A E X Q R L U I E B K N U I A I F H D X I E Y U E O M G C O M O
L M G T Z A G G S P R A I T C D S J U Q C Z S D M C M R W H H D H C
B W F M X L M N Z V H Q O B K E T U C B O O E H N J L Z S I T K U Q
N T R J M L U L I C U E E A D L K R O C T N H E S I I C D S D G D T
S Z R Z O E P U L G Y T J T K I F M O M J J C G F C N N V Z L J G P
F E M X O R B X A H N U Z K I A J I S J I S C I V T E M F Y H E N S
U S H R W G D I G G P I V Q I W T G G F E L S X W L S X U M C P W H
A P W C G Y Y S B W A Q P H V L P H I R Q L D Y I H E N U Z S D D E
M A X C A M J X P Y E I H M P R O Z C W E T A L L E G A L F X W J Z
W N V C V E W W W Q M B Z B I M P X T I B A J O S E I K R J T L N R
A S C S C M L A C I S S A L C O E N S R E H S O K A J F N W U U A H
X S A M H N N N H O K J W A P M S G G D O J X P F A V C Q R G Q O T
P C O R V G E N T R V I R M J O O O D R P Q H A H A U O T Y E L M S
V W I O F V O A P A E Q I N E B N J E D I X O D Y E V U Q I A Y V U
W V D X R T P I I T P Q U P H H Q T Q U G Y Z S F P N P F A N A A J
U L Q I A Z M L X D W W U B P M L I L S S W N Z N F J T I S X K J I
```



```
adulation
allergy
attachment
bastions
consisting
diciest
doggerel
excrescences
fjords
flagellate
halves
hemlines
impinging
jig
koshers
leaches
leafier
limousine
manures
neoclassical
notaries
oxide
peppering
prattle
rankles
reopening
satires
science
sewn
sheikh
thickening
tool
wailed
wanna
```



```
Y Y C J S C K Z H H O G P L Q L I J I X H X G P M Q Y P H N C O O R
R U C B P C I T F M T V P S T O E K T W T Z X H B G U F L Z I K Y L
Y S O N H K I E H S E I X Z M S X T P H R A B H Q B M P I D H D N P
V K W I Q X R E V H F U T Z M U C G I L P R H C G Y U X P J M X U F
Q J N Y R J J Z N W U Q L B Z X P C T T V Y Q O A Q Y A I B F B F A
R M O B W E C V J C X X J S O X K G N J K V G B J K G K L O O T M L
W L I U A I R E O P E N I N G E F P E P U T R O S I K F F M I T T Z
P N T M S X W F L P W W V L N P H G M B O Y G L V E G E H I E F P Y
E P A V R R M G N I M E P I X K R O H P Z S M U S Z L T Q R E I J L
U S L B D P K D M K A S N Q H W Z X C F V O L Q Z W R K T Y S U S U
Z C U U L M C G D N Y G Q H Z T M T A Q R J C C N M Q L N G F Y C N
N D D B D F G W O P W O Q H S D H S T A A S R A M S Y B B A R K I U
O E A O G O V T G G I A H N N W C G T C U R C Q Q B K V V Z R C F G
X W K H Z B A Y Z J U W G P U Z I G A P Y W Y D A G D T T O U S M N
G P H Y U R L T I E J J M Y B R N X T R E Z K W N W T D L O E L H S
F R G C I I E V T C M I P R A I Q T G N I T S I S N O C S R Q E D M
A A H E X H A R W R X X G U R Z Q Z S T J J D E K M F A U Q L R G F
R T S U M W F S T F X J S E I M W Z S E A Y V C P Y T N D I X E L Q
Z T Y S N O I T S A B J P E M E B M V J I L Q B Q I A I W Z Y G W J
P L K O B W E S G Z R P Z G N C X M R D A C B K R M H S S G V G X E
N E A E X Q R L U I E B K N U I A I F H D X I E Y U E O M G C O M O
L M G T Z A G G S P R A I T C D S J U Q C Z S D M C M R W H H D H C
B W F M X L M N Z V H Q O B K E T U C B O O E H N J L Z S I T K U Q
N T R J M L U L I C U E E A D L K R O C T N H E S I I C D S D G D T
S Z R Z O E P U L G Y T J T K I F M O M J J C G F C N N V Z L J G P
F E M X O R B X A H N U Z K I A J I S J I S C I V T E M F Y H E N S
U S H R W G D I G G P I V Q I W T G G F E L S X W L S X U M C P W H
A P W C G Y Y S B W A Q P H V L P H I R Q L D Y I H E N U Z S D D E
M A X C A M J X P Y E I H M P R O Z C W E T A L L E G A L F X W J Z
W N V C V E W W W Q M B Z B I M P X T I B A J O S E I K R J T L N R
A S C S C M L A C I S S A L C O E N S R E H S O K A J F N W U U A H
X S A M H N N N H O K J W A P M S G G D O J X P F A V C Q R G Q O T
P C O R V G E N T R V I R M J O O O D R P Q H A H A U O T Y E L M S
V W I O F V O A P A E Q I N E B N J E D I X O D Y E V U Q I A Y V U
W V D X R T P I I T P Q U P H H Q T Q U G Y Z S F P N P F A N A A J
U L Q I A Z M L X D W W U B P M L I L S S W N Z N F J T I S X K J I
```

```
Word found: (34/34).
Board executed in 0.017358s.
Total comparation done: 26647.
```

Lampiran

File ColoredChar.java

```
public class ColoredChar {
    private final char character;
    private Color color;

    public enum Color {
        RESET,
        RED,
        GREEN,
        YELLOW,
        BLUE,
        PURPLE,
        CYAN,
    }

    ColoredChar(char c) {
        this.character = Character.toUpperCase(c);
        this.color = Color.RESET;
    }

    public void setColor(ColoredChar.Color color) {
        this.color = color;
    }

    public char getChar() {
        return this.character;
    }

    public void printChar() {
        String ANSI;
        switch (this.color) {
            case RED -> ANSI = "\u001B[31m";
            case GREEN -> ANSI = "\u001B[32m";
            case YELLOW -> ANSI = "\u001B[33m";
            case BLUE -> ANSI = "\u001B[34m";
            case PURPLE -> ANSI = "\u001B[35m";
            case CYAN -> ANSI = "\u001B[36m";
            default -> ANSI = "\u001B[0m";
        }

        System.out.print(ANSI + this.character + "\u001B[0m" + " ");
    }
}
```

File Coordinate.java

```
public class Coordinate {
    private int x;
    private int y;

    Coordinate(int i, int j) {
        this.x = i;
        this.y = j;
    }
    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public void setCoordinate(int i, int j) {
        this.x = i;
        this.y = j;
    }

    public Coordinate copy() {
        return new Coordinate(this.x, this.y);
    }
}
```

File Matrix.java

```
import java.util.ArrayList;
public class Matrix {
    private final ColoredChar[][] contents;
    private int rows;
    private int cols;
    private int compareCount;

    public enum Direction {
        UPPER,
        UPPER_RIGHT,
        RIGHT,
        BOTTOM_RIGHT,
        BOTTOM,
        BOTTOM_LEFT,
        LEFT,
        UPPER_LEFT
    }

    Matrix(int rows, int cols) {
        this.contents = new ColoredChar[rows][cols];
        this.rows = rows;
        this.cols = cols;
        this.compareCount = 0;
    }

    public void setRows(int rows) {
        this.rows = rows;
    }

    public void setCols(int cols) {
        this.cols = cols;
    }

    public int getRows() {
        return this.rows;
    }
    public int getCols() {
        return this.cols;
    }

    public ColoredChar getElmt(Coordinate cr) {
        return this.contents[cr.getX()][cr.getY()];
    }

    public int getCompareCount() {
        return this.compareCount;
    }

    public void setElmt(Coordinate cr, char c, ColoredChar.Color color) {
        this.contents[cr.getX()][cr.getY()] = new ColoredChar(c);
    }

    public void printMatrix() {
        for (int i = 0; i < this.rows; i++) {
            for (int j = 0; j < this.cols; j++) {
                System.out.printf("%c ", this.getElmt(new Coordinate(i,
j)).getChar());
            }
            System.out.println();
        }
    }

    public void printColoredMatrix() {
        for (int i = 0; i < this.rows; i++) {
            for (int j = 0; j < this.cols; j++) {
                this.getElmt(new Coordinate(i, j)).printChar();
            }
            System.out.println();
        }
    }

    public void printWord(ArrayList<Coordinate> wordCoordinates, String word) {
        Matrix tmpMatrix = new Matrix(this.rows, this.cols);

        for (int i = 0; i < this.rows; i++) {
            for (int j = 0; j < this.cols; j++) {
                tmpMatrix.setElmt(new Coordinate(i, j), '-',
ColoredChar.Color.RESET);
            }
        }

        for (int i = 0; i < word.length(); i++) {
            tmpMatrix.setElmt(wordCoordinates.get(i), word.charAt(i),
ColoredChar.Color.RESET);
        }

        System.out.println("Word: " + word);
        tmpMatrix.printMatrix();
        System.out.println();
    }

    public boolean isCrValid(Coordinate cr) {
        return (cr.getX() >= 0 && cr.getY() >= 0 && cr.getX() < this.rows &&
cr.getY() < this.cols);
    }

    public boolean checkCharacter(Coordinate cr, char c) {
        this.compareCount++;
        return (this.getElmt(cr).getChar() == c);
    }

    private void moveRight(Coordinate cr) {
        cr.setCoordinate(cr.getX(), cr.getY() + 1);
    }

    private void moveBottomRight(Coordinate cr) {
        cr.setCoordinate(cr.getX() + 1, cr.getY() + 1);
    }

    private void moveBottom(Coordinate cr) {
        cr.setCoordinate(cr.getX() + 1, cr.getY());
    }

    private void moveBottomLeft(Coordinate cr) {
        cr.setCoordinate(cr.getX() + 1, cr.getY() - 1);
    }

    private void moveLeft(Coordinate cr) {
        cr.setCoordinate(cr.getX(), cr.getY() - 1);
    }

    private void moveUpperLeft(Coordinate cr) {
        cr.setCoordinate(cr.getX() - 1, cr.getY() - 1);
    }

    private void moveUpper(Coordinate cr) {
        cr.setCoordinate(cr.getX() - 1, cr.getY());
    }

    private void moveUpperRight(Coordinate cr) {
        cr.setCoordinate(cr.getX() - 1, cr.getY() + 1);
    }

    public ArrayList<Direction> decideDirection(Coordinate cr, char[] word) {
        ArrayList<Direction> dirList = new ArrayList<>();
        int len = word.length - 1;
        int x = cr.getX();
        int y = cr.getY();

        int[] moveX = {0, 1, 1, 1, 0, -1, -1, -1};
        int[] moveY = {1, 1, 0, -1, -1, -1, 0, 1};
        Direction[] availableDir = {
                Direction.RIGHT,
                Direction.BOTTOM_RIGHT,
                Direction.BOTTOM,
                Direction.BOTTOM_LEFT,
                Direction.LEFT,
                Direction.UPPER_LEFT,
                Direction.UPPER,
                Direction.UPPER_RIGHT
        };

        for (int i = 0; i < moveX.length; i++) {
            Coordinate secCr = new Coordinate(x + moveX[i], y + moveY[i]);
            Coordinate lastCr = new Coordinate(x + moveX[i] * len, y + moveY[i] *
len);
            if (this.isCrValid(secCr) && this.isCrValid(lastCr) &&
this.checkCharacter(secCr, word[1])) {
                dirList.add(availableDir[i]);
            }
        }
        return dirList;
    }

    public void moveOneStep(Coordinate cr, Direction dir) {
        switch (dir) {
            case RIGHT -> moveRight(cr);
            case BOTTOM_RIGHT -> moveBottomRight(cr);
            case BOTTOM_LEFT -> moveBottomLeft(cr);
            case BOTTOM -> moveBottom(cr);
            case LEFT -> moveLeft(cr);
            case UPPER_LEFT -> moveUpperLeft(cr);
            case UPPER -> moveUpper(cr);
            case UPPER_RIGHT -> moveUpperRight(cr);
        }
    }
}
```

File CrossWord.java

```
import java.io.*;
import java.util.ArrayList;

public class CrossWord {
    private Matrix board;
    private ArrayList<String> words;

    CrossWord() {
        this.board = null;
        this.words = null;
    }

    public Matrix getBoard() {
        return this.board;
    }

    public ArrayList<String> getWords() {
        return this.words;
    }

    public void readCWFile(String relFilePath) throws IOException {
        try {
            File file = new File(relFilePath);
            BufferedReader br = new BufferedReader(new FileReader(file));
            String ln;
            ArrayList<String> matLines = new ArrayList<>();

            // Read board first
            ln = br.readLine();
            while (!ln.isEmpty()) {
                matLines.add(ln);
                ln = br.readLine();
            }

            // Move board to matrix
            int rows = matLines.size();
            int cols = matLines.get(0).replaceAll("\\s", "").length();
            this.board = new Matrix(rows, cols);

            for (int i = 0; i < matLines.size(); i++) {
                char[] inchar = matLines.get(i).replaceAll("\\s",
"").toCharArray();
                for (int j = 0; j < inchar.length; j++) {
                    this.board.setElmt(new Coordinate(i, j), inchar[j],
ColoredChar.Color.RESET);
                }
            }

            // Read list of available words
            this.words = new ArrayList<>();
            ln = br.readLine();
            while (ln != null) {
                if (!this.words.contains(ln))
                    words.add(ln.toUpperCase());
                ln = br.readLine();
            }

            br.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found!");
            e.printStackTrace();
        }
    }

    public void searchWords(boolean printColored) {
        ArrayList<Coordinate> wordCoordinates = new ArrayList<>();
        ColoredChar.Color[] availableColor = {
                ColoredChar.Color.RED,
                ColoredChar.Color.GREEN,
                ColoredChar.Color.YELLOW,
                ColoredChar.Color.BLUE,
                ColoredChar.Color.PURPLE,
                ColoredChar.Color.CYAN
        };

        int wordFound = 0;
        int index = 0;

        // For every word in the list
        for (String word : this.words) {
            char[] inchar = word.toCharArray();
            int i = 0;
            boolean found = false;
            while (i < this.board.getRows() && !found) {
                int j = 0;
                while (j < this.board.getCols() && !found) {
                    Coordinate startCr = new Coordinate(i, j); // Starting
coordinate

                    // Check first character
                    if (this.board.checkCharacter(startCr, inchar[0])) {
                        // Proceed to check surrounding if initial character
match
                        ArrayList<Matrix.Direction> dirList =
this.board.decideDirection(startCr, inchar);

                        // Iterate through every direction possible
                        for (Matrix.Direction dir : dirList) {
                            // Initialize starting point
                            int itr = 1;
                            boolean walk = true;
                            wordCoordinates.add(startCr);
                            Coordinate cr = startCr.copy();

                            // Walk
                            while (itr < inchar.length && walk) {
                                this.board.moveOneStep(cr, dir);
                                if (!this.board.isCrValid(cr) ||
!this.board.checkCharacter(cr, inchar[itr])) {
                                    walk = false;
                                } else {
                                    wordCoordinates.add(cr.copy());
                                    itr++;
                                }
                            }

                            // If we found the word, print the board. Otherwise,
start again
                            if (itr == inchar.length) {
                                if (!printColored) {
                                    this.board.printWord(wordCoordinates, word);
                                } else {
                                    for (Coordinate ctemp : wordCoordinates) {

this.board.getElmt(ctemp).setColor(availableColor[index]);
                                    }
                                    index = (index + 1) % availableColor.length;
                                    found = true;
                                }
                                wordFound++;
                                wordCoordinates.clear();
                            } else {
                                wordCoordinates.clear();
                            }
                            if (found)
                                break;
                        }
                    }
                    j++;
                }
                i++;
            }
        }
        if (printColored) {
            this.board.printColoredMatrix();
        }
        System.out.println("Word found: (" + wordFound + "/" + this.words.size()
+ ").");
    }
}
```

File Main.java

```
import java.io.IOException;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) throws IOException {
        CrossWord cw = new CrossWord();
        Scanner reader = new Scanner(System.in);
        System.out.print("Input your filename: ");
        String inputFile = reader.next();
        cw.readCWFile("test/" + inputFile);

        System.out.println("Initial board:");
        cw.getBoard().printMatrix();
        System.out.println("Provided words:");
        System.out.println(cw.getWords() + "\n");

        // Ask if want to print colored
        boolean printColored = true;
        System.out.print("Want to print the board colored? (Y/N) ");
        char in = reader.next().charAt(0);
        if (in == 'N' || in == 'n')
            printColored = false;

        double startTime = System.nanoTime();
        cw.searchWords(printColored);
        double elapsedTime = (System.nanoTime() - startTime) / Math.pow(10, 9);

        System.out.println("Board executed in " + elapsedTime + "s.");
        System.out.println("Total comparation done: " +
cw.getBoard().getCompareCount() + ".");
    }
}
```