

Spesifikasi Tugas Besar 1

IF2111 Algoritma dan Struktur Data STI

PURRMART

Revisi

ver. 22 November 2024

Revisi 12/11/2024 - PURRMART THE PLATYPUS?!

Revisi 15/11/2024 - Mengubah nama save pada contoh struktur program

Revisi 15/11/2024 - Menambah keterangan kata valid pada *challenge* WORDL3

Revisi 20/11/2024 - Memperbaiki kesalahan input di WORK CHALLENGE

Revisi 22/11/2024 - Memperbaiki jumlah kesempatan mencoba di WORDL3

Revisi 24/11/2024 - Memperbaiki kesalahan input di LOAD, SAVE, dan QUIT

Revisi 25/11/2024 - Menambahkan form pengumpulan

Deadline

Milestone 1: **25 November 2024 21.11 WIB**

Form Pengumpulan: <https://forms.gle/Hkjc4a1L5xA1Gs8w5>

Daftar Isi

Daftar Isi	2
Latar Belakang	3
Spesifikasi Umum	5
System Mechanic	5
1. About the System	5
2. Menu Program	5
3. Command	6
a. START	6
b. LOAD <filename>	6
c. LOGIN	6
d. LOGOUT	7
e. REGISTER	7
f. WORK	8
g. WORK CHALLENGE	8
a) Tebak Angka	9
b) WORDL3	10
h. STORE LIST	13
i. STORE REQUEST	13
j. STORE SUPPLY	14
k. STORE REMOVE	14
l. HELP	15
m. SAVE <filename>	16
n. QUIT	16
Konfigurasi Sistem	16
Daftar ADT	17
Bonus	18
Catatan Tambahan	23

Latar Belakang



Agen Purry sedang menikmati tidur siangnya ketika dia tiba-tiba mendengar *alarm* dari belakang sofa. Suatu pintu rahasia terbuka di bawah dirinya dan ia jatuh ke ruang bawah tanah dan langsung disambut dengan misi terbarunya.



"Ah Agen Purry, maaf harus mengganggu waktu tidur kamu tapi kami mendapatkan laporan bahwa Dr. Asep Spakbor sedang membuat suatu mesin yang dinamakan 'Oppenheimer-inator' yang akan menghancurkan wilayah tiga negara bagian. Aku membutuhkan bantuanmu untuk menghentikan Dr. Asep Spakbor, ini merupakan ancaman terbesar yang pernah ia buat."

And indeed it was. Setelah pertarungan sengit selama 3 bulan 13 hari 2 jam 47 menit dan 2 detik, suplai senjata dan suplai peralatan yang dimiliki OWCA mulai menipis. Harapan kemenangan OWCA mulai memudar...

Tanpa disangka, Agen Purry mengeluarkan senjata rahasia miliknya: menjadi orang Bojongsoang yang memiliki kenalan pegawai Borma. Toko Borma adalah komponen penting yang dapat membawakan kemenangan untuk OWCA pada waktu-waktu kritis ini. Sebab, meskipun Borma terlihat seperti *supermarket* pada umumnya, mereka sebenarnya merupakan pemasok barang-barang perang. Namun terdapat satu masalah kecil, Borma masih beroperasi secara tatap muka dan OWCA tidak memiliki *transport* untuk pergi ke Bojongsoang.



(Reaksi Jujur Purry)

Untuk menyelesaikan permasalahan ini, OWCA mengontak tim *programmer* paling andalnya untuk merancang suatu sistem jual beli ke Borma dengan nama **PURRMART! Benar, tim tersebut adalah kalian!** Misi ini akan menantang dan menguji kalian. Namun, dengan kerja tim dan tekad yang kuat, kalian pasti dapat menghadapi tantangan ini.

Good luck, programmers! OWCA counts on you! (•̀ω•́) ✧

Spesifikasi Umum

Buatlah sebuah aplikasi simulasi berbasis CLI (*command-line interface*). Sistem ini dibuat dalam **bahasa C** dengan menggunakan **struktur data yang sudah kalian pelajari** di mata kuliah ini. Kalian boleh menggunakan (atau memodifikasi) struktur data yang sudah kalian buat untuk praktikum pada tugas besar ini. Daftar ADT yang wajib digunakan dapat dilihat pada bagian [Daftar ADT](#). *Library* yang boleh digunakan hanya **stdio.h**, **stdlib.h**, **time.h**, dan **math.h**.

System Mechanic

1. About the System

PURRMART adalah sebuah aplikasi yang dapat mensimulasikan aktivitas beli barang pada *e-commerce*. PURRMART memiliki beberapa fitur utama, yaitu:

- Menampilkan barang toko
- Meminta dan menyuplai barang baru ke toko
- Menyimpan dan membeli barang dalam keranjang
- Menampilkan barang yang sudah dibeli
- Membuat dan menghapus *wishlist*
- Bekerja untuk menghasilkan uang

2. Menu Program

Ketika program pertama kali dijalankan, PURRMART akan memperlihatkan *main menu* yang berisi **welcome menu** dan beberapa *command* yaitu **START**, **LOAD**, dan juga **HELP**.

Setelah itu, program akan memasuki **login menu** yang memiliki *command* **LOGIN**, **REGISTER**, dan juga **HELP**. Jika pengguna berhasil memasuki kredensial suatu akun, maka mereka akan masuk ke menu selanjutnya.

Main menu menerima masukan berupa *command* yang akan dijelaskan pada bagian berikutnya. Program akan terus menerima *command* sampai diberikan *command* **QUIT** yang berlaku pada seluruh menu.

3. Command

Pengguna dapat memasukkan *command-command* berikut.

a. START

START merupakan salah satu command yang dimasukkan pertama kali dalam Toko Purrmart. Setelah menekan Enter, dibaca file konfigurasi *default* yang berisi daftar barang pada toko.

```
>> START
File konfigurasi aplikasi berhasil dibaca. PURRMART berhasil
dijalankan.
```

b. LOAD <filename>

LOAD merupakan salah satu *command* yang dimasukkan pertama kali dalam PURRMART. Command ini memiliki satu argumen yaitu *filename* yang merepresentasikan suatu *save file* yang ingin dibuka. *File* didapatkan dari *folder* tertentu, contohnya *save*. Setelah menekan *Enter*, akan dibaca *save file* <*filename*> yang berisi daftar barang pada toko. Lebih detailnya bisa dilihat pada [Konfigurasi Aplikasi](#).

```
>> LOAD filenihil.txt
// Meload /save/filenihil.txt

Save file tidak ditemukan. PURRMART gagal dijalankan.

>> LOAD savefile.txt
// Meload /save/savefile.txt

Save file berhasil dibaca. PURRMART berhasil dijalankan.
```

c. LOGIN

Login merupakan *command* yang baru dapat dipanggil setelah pengguna memulai sesi. *Login* berguna untuk masuk ke akun di sistem PURRMART yang sudah didaftarkan sebelumnya.

```
// Contoh login yang berhasil untuk username johndoe dan password
```

```
janedoe
```

```
>> LOGIN
```

```
Username: johndoe
```

```
Password: janedoe
```

Anda telah login ke PURRMART sebagai johndoe.

```
// Contoh login yang gagal karena password salah
```

```
>> LOGIN
```

```
Username: johndoe
```

```
Password: johndoe
```

Username atau password salah.

```
// Contoh login yang gagal karena belum ada pengguna dengan username  
janedoe
```

```
>> LOGIN
```

```
Username: janedoe
```

```
Password: johndoe
```

Username atau password salah.

```
// Contoh login yang gagal karena pengguna belum LOGOUT
```

```
>> LOGIN
```

```
Username: janedoe
```

```
Password: johndoe
```

Anda masih tercatat sebagai johndoe. Silakan LOGOUT terlebih dahulu.

d. LOGOUT

LOGOUT merupakan salah satu *command* yang baru dapat digunakan setelah pengguna telah memasuki sebuah sesi.

```
>> LOGOUT
```

johndoe telah logout dari sistem PURRMART. Silakan REGISTER/LOGIN kembali untuk melanjutkan.

e. REGISTER

Register merupakan *command* yang baru dapat dipanggil setelah pengguna memulai sesi. *Register* berguna untuk mendaftarkan akun baru ke dalam sistem PURRMART. Sebuah akun setidaknya memiliki atribut *username* dan *password*. **Username dan password hanya terdiri dari 1 kata.**

```
// Contoh register yang berhasil
```

```
>> REGISTER
```

```
Username: johndoe
```

```
Password: janedoe
```

Akun dengan username johndoe telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.

```
// Contoh register yang gagal karena username sudah ada
```

```
>> REGISTER
```

```
Username: johndoe
```

```
Password: janedoe
```

Akun dengan username johndoe gagal dibuat. Silakan lakukan REGISTER ulang.

f. WORK

WORK merupakan *command* yang digunakan pengguna untuk mendapatkan uang. Terdapat sejumlah pekerjaan yang bisa dipilih. Setiap pekerjaan memiliki waktu tunggu yang berbeda-beda dan dengan nominal pendapatan yang berbeda-beda pula. Selama pengguna sedang bekerja, maka sistem tidak bisa digunakan hingga pekerjaan selesai dilakukan.

```
>> WORK
```

Daftar pekerjaan:

1. Evil Lab Assistant (pendapatan=100, durasi=14s)
2. OWCA Hiring Manager (pendapatan=4200, durasi=21s)
3. Cikapundunginator Caretaker (pendapatan=7000, durasi=30s)
4. Mewing Specialist (pendapatan=10000, durasi=22s)
5. Inator Connoisseur (pendapatan=997, durasi=15s)

Masukkan pekerjaan yang dipilih: **Mewing Specialist**

Anda sedang bekerja sebagai Mewing Specialist... harap tunggu.

Pekerjaan selesai, +10000 rupiah telah ditambahkan ke akun Anda.

g. WORK CHALLENGE

WORK CHALLENGE merupakan *command* alternatif sebagai cara mendapatkan uang dengan melakukan *challenge-challenge* di OWCA. Pemain membutuhkan uang dengan jumlah tertentu untuk bisa memainkan challenge. Uang yang dibayarkan untuk bermain *challenge* tidak akan dikembalikan, meskipun pemain kalah dalam permainan. Terdapat dua *challenge* yang dapat dipilih:

a) Tebak Angka

Challenge Tebak Angka merupakan permainan yang meminta pemain menebak sebuah angka yang ditentukan oleh program. Pemain memiliki 10 (sepuluh) kesempatan untuk menebak angka yang benar. Program akan memberikan *feedback* apakah angka tebakkan lebih besar, lebih kecil, atau sama dengan angka target. Jumlah kesempatan yang dipakai oleh pengguna akan mempengaruhi uang yang didapatkan.

>> WORK CHALLENGE

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. WORDL399 (biaya main=500)

Masukan challenge yang hendak dimainkan: **1**

Tebak angka: **50**

Tebakanmu lebih kecil!

Tebak angka: **75**

Tebakanmu lebih besar!

Tebak angka: **65**

Tebakanmu lebih besar!

Tebak angka: **62**

Tebakanmu benar! +350 rupiah telah ditambahkan ke akun anda.

>> **WORK CHALLENGE**

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. WORDL399 (biaya main=500)

Masukan challenge yang hendak dimainkan: **1**

Tebak angka: **50**

Tebakanmu benar! +500 rupiah telah ditambahkan ke akun anda.

b) WORDL3

Challenge WORDL3 merupakan permainan tebak kata berjumlah lima karakter. Pemain memiliki 6 (enam) kesempatan untuk menebak kata yang benar. Kata harus berupa kata valid, tidak boleh sekadar *string* acak, bahasa dibebaskan (disarankan bahasa Indonesia/Inggris). Pada setiap giliran, program akan mencetak ulang kata yang dimasukan, tetapi dengan penanda tertentu. Huruf yang benar dan berada pada tempat yang tepat dicetak biasa. Huruf yang benar, tetapi berada di tempat yang salah diberi tanda "*" setelah hurufnya. Huruf yang tidak ada sama sekali pada kata diberi tanda "%" setelah hurufnya.

>> **WORK CHALLENGE**

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. WORDL399 (biaya main=500)

Masukan challenge yang hendak dimainkan: **2**

WELCOME TO WORDL3, YOU HAVE 5 CHANCES TO ANSWER BEFORE YOU LOSE!

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Masukan kata tebakan Anda: **ADIEU**

Hasil:

A% D% I% E% U*

- - - - -

- - - - -

- - - - -

- - - - -

- - - - -

Masukan kata tebakan Anda: **OMBRE**

Hasil:

A% D% I% E% U*

O% M% B% R* E%

- - - - -

- - - - -

- - - - -

- - - - -

Masukan kata tebakan Anda: **PLUCK**

Hasil:

A% D% I% E% U*

O% M% B% R* E%

P% L* U C% K%

- - - - -

- - - - -

- - - - -

Masukan kata tebakan Anda: **SERIN**

Hasil:

A% D% I% E% U*

O% M% B% R* E%

P% L* U C% K%

S% E% R* I% N%

- - - - -

- - - - -

Masukan kata tebakan Anda: **TRULY**

Hasil:

A% D% I% E% U*

O% M% B% R* E%

P% L* U C% K%

S% E% R* I% N%

T R U L Y

- - - - -

Selamat, Anda menang!

+1500 rupiah telah ditambahkan ke akun Anda.

>> **WORK CHALLENGE**

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. W0RDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 2

WELCOME TO W0RDL3, YOU HAVE 5 CHANCES TO ANSWER BEFORE YOU LOSE!

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Masukan kata tebakan Anda: **ADIEU**

Hasil:

A% D% I% E* U%

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Masukan kata tebakan Anda: **TRULY**

Hasil:

A% D% I% E* U%

T% R% U% L% Y%

- - - - -
- - - - -
- - - - -
- - - - -

Masukan kata tebakan Anda: **LUCKY**

Hasil:

A% D% I% E% U*

T% R% U% L% Y%

L% U% C% K% Y%

- - - - -
- - - - -
- - - - -

Masukan kata tebakan Anda: **SLICK**

Hasil:

A% D% I% E% U*

T% R% U% L% Y%

L% U% C% K% Y%

S* L% I% C% K%

- - - - -

```

- - - - -

Masukan kata tebakan Anda: LEAKY
Hasil:
A% D% I% E% U*
T% R% U% L% Y%
L% U% C% K% Y%
S* L% I% C% K%
L% E A% K% Y%

- - - - -

Masukan kata tebakan Anda: BUILD
Hasil:
A% D% I% E% U*
T% R% U% L% Y%
L% U% C% K% Y%
S* L% I% C% K%
L% E A% K% Y%
B% U% I% L% D%

Boo! Anda kalah.

```

h. STORE LIST

STORE LIST adalah *command* yang digunakan untuk melihat barang-barang apa saja yang ada di dalam toko. **Setiap barang yang ditampilkan haruslah bersifat *unique*.**

```

>> STORE LIST
List barang yang ada di toko :
- Platypus Laser
- Shrink Ray
- Net Shooter
- Camouflage Cloak
- Sleep Dart Gun
- Bubble Blaster

>> STORE LIST
TOKO KOSONG

```

i. STORE REQUEST

STORE REQUEST adalah *command* yang digunakan untuk meminta penambahan barang baru ke dalam toko. Barang-barang yang diminta akan

disimpan di dalam sebuah antrian dan akan dimasukkan ke toko menggunakan *command* selanjutnya. **Nama barang yang masuk tidak boleh sama dengan nama barang yang sudah ada di toko atau di antrian.**

>> STORE REQUEST Nama barang yang diminta: AK47
>> STORE REQUEST Nama barang yang diminta: Adaditoko Barang dengan nama yang sama sudah ada di toko!
>> STORE REQUEST Nama barang yang diminta: Adadiantrian Barang dengan nama yang sama sudah ada di antrian!

j. **STORE SUPPLY**

STORE SUPPLY adalah command yang digunakan untuk menambahkan barang baru ke dalam toko berdasarkan antrian permintaan. Barang yang berada pada antrian paling depan akan dimasukan ke toko. Pengguna dapat menerima, menunda, atau menolak permintaan.

- Jika diterima, maka program akan meminta harga dari barang dan dimasukan ke toko.
- Jika ditunda, maka barang akan kembali masuk ke antrian
- Jika ditolak, maka barang akan dihapus dari antrian

Harus terdapat validasi agar harga barang merupakan angka yang valid (berupa angka dan bernilai lebih dari nol).

>> STORE SUPPLY Apakah kamu ingin menambahkan barang AK47: Terima Harga barang: 100 AK47 dengan harga 100 telah ditambahkan ke toko.
>> STORE SUPPLY Apakah kamu ingin menambahkan barang AK47: Tunda AK47 dikembalikan ke antrian.
>> STORE SUPPLY Apakah kamu ingin menambahkan barang AK47: ToLak

AK47 dihapuskan dari antrian.

>> **STORE SUPPLY**

Apakah kamu ingin menambahkan barang AK47: **Purrry**

< Balik ke menu >

k. STORE REMOVE

STORE REMOVE adalah *command* yang dapat menghapus barang yang ada di toko. Akan dilakukan *input* akan barang yang akan dihapus. Beri tahu apabila proses berhasil (barang terdapat pada toko dan berhasil dihapus) ataupun tidak (barang tidak terdapat di toko).

>> **STORE REMOVE**

Nama barang yang akan dihapus: **Sticky Web Gun**

Sticky Web Gun telah berhasil dihapus.

>> **STORE REMOVE**

Nama barang yang akan dihapus: **Inator Neutralizer**

Toko tidak menjual Inator Neutralizer

l. HELP

HELP merupakan *command* yang digunakan menampilkan daftar *command* yang mungkin untuk dieksekusi dengan deskripsinya. Penjelasan dari deskripsi dibebaskan selama masih mendeskripsikan *command* sesuai dengan spek.

// Ketika perintah dipanggil pada welcome menu

>> HELP

====[Welcome Menu Help PURRMART]=====

1. START → Untuk masuk sesi baru
2. LOAD → Untuk memulai sesi berdasarkan file konfigurasi
3. QUIT → Untuk keluar dari program

// Ketika perintah dipanggil pada login menu

>> HELP

====[Login Menu Help PURRMART]=====

1. REGISTER → Untuk melakukan pendaftaran akun baru

2. LOGIN → Untuk masuk ke dalam akun dan memulai sesi
3. QUIT → Untuk keluar dari program

```
// Ketika perintah dipanggil pada main menu
>> HELP

=====[ Menu Help PURRMART]=====
  1. WORK → Untuk bekerja
  2. WORK CHALLENGE → Untuk mengerjakan challenge
  3. STORE LIST → Untuk melihat barang-barang di toko
  4. STORE REQUEST → Untuk meminta penambahan barang
  5. STORE SUPPLY → Untuk menambahkan barang dari permintaan
  6. STORE REMOVE → Untuk menghapus barang
  7. LOGOUT → Untuk keluar dari sesi
  8. SAVE → Untuk menyimpan state ke dalam file
  9. QUIT → Untuk keluar dari program
```

m. **SAVE <filename>**

SAVE merupakan *command* yang digunakan untuk menyimpan *state* aplikasi terbaru ke dalam suatu *file*. Command SAVE memiliki satu argumen yang merepresentasikan nama *file* yang akan disimpan. Penyimpanan dilakukan pada *folder* tertentu, misal *folder save*.

```
>> SAVE savefile.txt

Save file berhasil disimpan.
// File disimpan pada /save/savefile.txt
```

n. **QUIT**

QUIT merupakan *command* yang digunakan untuk keluar dari sesi aplikasi PURRMART.

```
>> QUIT

Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? N

Kamu keluar dari PURRMART.
Dadah ^_^/
```



```
>> QUIT
```

```
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? Y
// memanggil fitur Save
```

Konfigurasi Sistem

File konfigurasi akan dibaca saat memulai permainan. File ini menyimpan data-data yang disimpan ketika sistem dijalankan sebelumnya atau data-data *default*. Spesifikasi dari *file* konfigurasi adalah sebagai berikut:

1. Barisan pertama adalah bilangan bulat positif **N** yang menunjukkan berapa banyak barang di dalam sistem
2. Selanjutnya, sejumlah **N** baris menyatakan nama barang beserta harganya dengan format **<Harga barang> <Nama barang>**
3. Baris selanjutnya adalah bilangan bulat positif **M** yang menunjukkan berapa banyak pengguna di dalam sistem
4. Selanjutnya, sejumlah **M** baris menyatakan nama pengguna, password, jumlah uang dengan format **<Uang> <Nama> <Password>**

Berikut adalah contoh file konfigurasi yang dimuat di awal sebuah session sebagai inisialisasi:

```
3
10 AK47
20 Lalabu
20 Ayam Goreng Crisbar
2
100 admin alstrukdatkeren
25 praktikan kerenbangetkak
```

Perlu diperhatikan bahwa antrian permintaan barang tidak disimpan di konfigurasi! Jika sistem dimulai, maka antrian akan dibuat lagi dari 0.

Daftar ADT

1. ADT Kustom

ADT Kustom dibuat untuk merepresentasikan seorang pengguna (User) dan sebuah barang (Barang). ADT User akan berisi nama serta *password* berupa *string* dan uang berupa *integer*.

```
typedef struct {  
    char name[MAX_LEN];  
    char password[MAX_LEN];  
    integer money;  
} User;
```

Sedangkan ADT Barang akan berisi nama berupa *string* dan harga berupa *integer*.

```
typedef struct {  
    char name[MAX_LEN];  
    integer price;  
} Barang;
```

2. ADT List

ADT ini digunakan dengan implementasi *array* yang terbagi berdasarkan tipe pengalokasiannya, yaitu statis dan dinamis. *List* statis digunakan untuk merepresentasikan daftar pengguna yang terdapat dalam program. Sedangkan *list* dinamis digunakan untuk merepresentasikan daftar barang yang ada pada toko.

3. ADT Mesin Karakter dan Mesin Kata

ADT ini digunakan untuk melakukan *parsing command* dalam aplikasi dan membaca ataupun membuat *file* konfigurasi dari aplikasi. Dilarang menggunakan **scanf** ataupun **fgets** ketika memproses *command* dan *file* konfigurasi.

4. ADT Queue

ADT ini digunakan untuk merepresentasikan urutan permintaan penambahan barang.

Bonus

Pada tugas besar ini, terdapat beberapa fitur tambahan yang bisa diimplementasikan. Fitur-fitur ini tidak wajib untuk dikerjakan dan bobotnya lebih kecil dibanding fitur utama. **Utamakan fitur-fitur utama yang diminta sebelum mengerjakan bonus.** Berikut adalah penjelasan dari masing-masing fitur bonus:

1. Quantum WORDL3

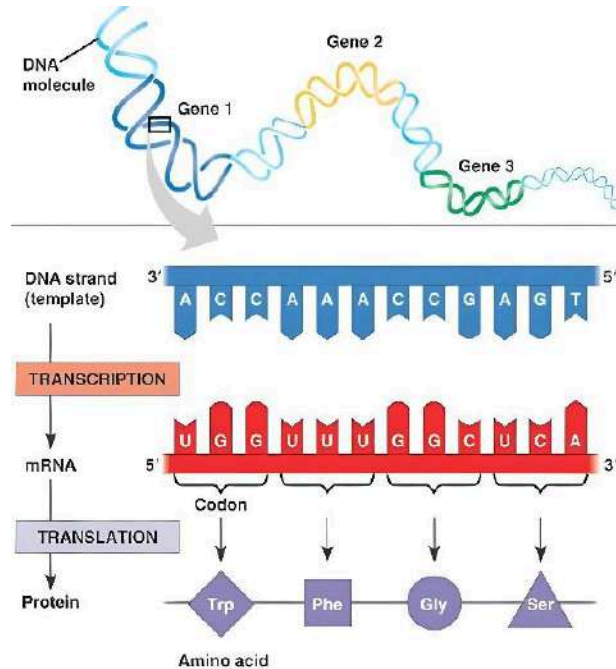
Fitur tambahan berupa *work challenge* ekstra bagi pemain. Berbeda dengan WORDL3 yang terdapat pada spesifikasi wajib, fitur Quantum WORDL3 memungkinkan pemain untuk menebak empat kata sekaligus dalam satu waktu (keempat kata harus dijamin berbeda).

Pada permainan ini, pemain diberi kesempatan untuk menebak sebanyak 9 kali. Ketentuan lain sama seperti fitur WORDL3 pada spesifikasi wajib. Format visualisasi dibebaskan selama mampu menunjukkan *state* dari permainan dengan jelas.

2. Bio Weapon

Fitur tambahan yang memungkinkan pemain untuk menambahkan barang baru berupa senjata biologis ke PURRMART. Pada dasarnya, senjata biologis merupakan jenis senjata yang menggunakan patogen untuk menyerang target, mulai dari bakteri, fungi, maupun virus. Namun, apabila tidak berhati-hati, senjata biologis rentan disabotase dan justru dapat berakibat buruk bagi pemiliknya. Maka, dalam kasus ini pabrik telah memutuskan untuk **merekayasa genom patogen dan menyisipkan kode tersembunyi** di dalamnya.

Dalam biologi molekuler, dikenal istilah [*central dogma*](#) sebagai alur aliran informasi genetik. Informasi disimpan dalam bentuk DNA, ditranskripsi menjadi RNA, kemudian diterjemahkan menjadi protein.



Gambar 1.1 Ilustrasi *Central Dogma*

DNA (*deoxyribonucleic acid*) terdiri dari empat basa nukleotida: adenina (A), guanina (G), timina (T), dan sistein (C). RNA (*ribonucleic acid*) mirip seperti DNA, hanya saja timina (T) diganti dengan urasil (U). Pada proses transkripsi (DNA→RNA), Anda perlu mengganti nukleotida pada DNA dengan nukleotida komplementernya. Adenina (A) berpasangan dengan Urasil (U) sementara Guanina (G) berpasangan dengan Sistein (C). Berikut adalah ilustrasi transkripsi DNA menjadi RNA:

DNA: TACTAGAA

RNA: AUGAUCUU

Perhatikan bahwa T pada DNA menjadi A di RNA, A pada DNA menjadi U di RNA. G pada DNA menjadi C di RNA dan sebaliknya.

		Second letter				
		U	C	A	G	
First letter	U	UUU } Phe UUC } UUA } Leu UUG }	UCU } UCC } Ser UCA } UCG }	UAU } Tyr UAC } UAA Stop UAG Stop	UGU } Cys UGC } UGA Stop UGG Trp	U C A G
	C	CUU } CUC } Leu CUA } CUG }	CCU } CCC } Pro CCA } CCG }	CAU } His CAC } CAA Gln CAG }	CGU } CGC } Arg CGA } CGG }	U C A G
	A	AUU } AUC } Ile AUA } AUG Met	ACU } ACC } Thr ACA } ACG }	AAU } Asn AAC } AAA Lys AAG }	AGU } Ser AGC } AGA } Arg AGG }	U C A G
	G	GUU } GUC } Val GUA } GUG }	GCU } GCC } Ala GCA } GCG }	GAU } Asp GAC } GAA Glu GAG }	GGU } GGC } Gly GGA } GGG }	U C A G

Gambar 1.2 Tabel Kodon-Asam Amino

Setelah DNA [ditranskripsikan](#) ke RNA, maka selanjutnya akan ditranslasi oleh ribosom dengan tabel kodon. Protein terdiri dari 20 asam amino yang tersusun sebagai rantai peptida. Perhatikan pada gambar 1.2 bahwa pembacaan RNA dilakukan secara triplet (kelipatan tiga). Satu asam amino dapat dikode oleh lebih dari satu kodon. Apabila terdapat kodon *stop* pada sekuens, maka lewati saja (anggap tidak ada apa-apa). Abaikan pula mekanisme kodon *start*.

G Glycine	Gly	P Proline	Pro
A Alanine	Ala	V Valine	Val
L Leucine	Leu	I Isoleucine	Ile
M Methionine	Met	C Cysteine	Cys
F Phenylalanine	Phe	Y Tyrosine	Tyr
W Tryptophan	Trp	H Histidine	His
K Lysine	Lys	R Arginine	Arg
Q Glutamine	Gln	N Asparagine	Asn
E Glutamic Acid	Glu	D Aspartic Acid	Asp
S Serine	Ser	T Threonine	Thr

Gambar 1.3 Singkatan Asam Amino

Seperti yang dapat Anda lihat pada gambar 1.3, asam amino memiliki singkatan tiga huruf dan singkatan satu huruf. Singkatan asam amino satu huruf inilah yang akan digunakan oleh pabrik untuk menyisipkan kode di senjata biologis.

Berikut adalah contoh alur lengkap DNA→RNA→Protein.

<p>DNA: TACTAGAAGCATTAT</p> <p>↓</p> <p>RNA: AUGAUCUUCGUAUA</p> <p>↓</p> <p>Protein: Methionine - Isoleucine - Phenylalanine - Valine - Isoleucine</p> <p>Protein dengan asam amino representasi 1 huruf: MIFVI</p>
<p>DNA: AGATAACCATACCGG</p> <p>↓</p> <p>RNA: UCUAUUGGUAUGGCC</p> <p>↓</p> <p>Protein: Serine - Isoleucine - Glycine - Methionine - Alanine</p> <p>Protein dengan asam amino representasi 1 huruf: SIGMA</p>

Program akan meminta Anda untuk memasukan sekuens DNA dan kode rahasia yang disepakati antara Anda dan pabrik. Apabila kode rahasia tersebut terdeteksi, maka barang dimasukan ke *queue* seperti pada fitur STORE REQUEST. Namun, jika kode rahasia tidak ditemukan, maka barang tidak akan dimasukan ke dalam *queue*.

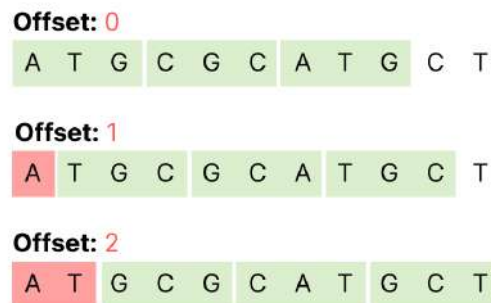
Berikut adalah contoh interaksi program.

<p>>> STORE REQUEST BIOWEAPON</p> <p>Masukan nama senjata biologis: Batuksius fififafae</p> <p>Masukan sekuens DNA: TAAATGATGAGATAACCATACCGGGCCGCAATT</p> <p>Masukan kode rahasia: SIGMA</p> <p>Senjata biologis mengandung kode, barang akan ditambahkan ke dalam queue!</p> <p>// Batuksius fififafae masuk ke dalam queue barang</p>
<p>>> STORE REQUEST BIOWEAPON</p> <p>Masukan nama senjata biologis: Achoosius pilekae</p> <p>Masukan sekuens DNA: CCAATGAACCGCTCGGCATACCATGCCGCTGAT</p> <p>Masukan kode rahasia: SKIBIDI</p>

Kode rahasia tidak ditemukan, maka senjata biologis sudah disabotase, barang ditolak!

// Achoosius pilekae tidak masuk ke dalam *queue* barang

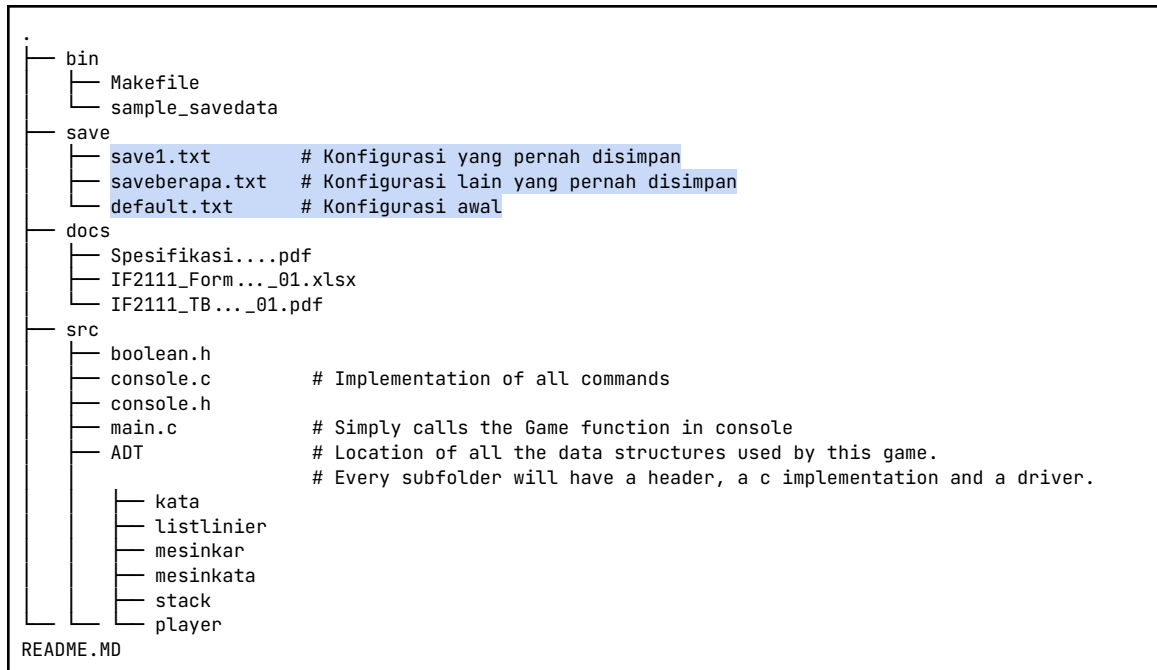
Hint: Apabila Anda perhatikan pada gambar 1.4 di bawah, pembacaan RNA secara triplet memiliki konsekuensi adanya *window/frame* pembacaan. Pastikan Anda mengecek seluruh *window* pembacaan.



Gambar 1.4 Ilustrasi *Sliding Window* Pembacaan

Catatan Tambahan

1. Tampilan program boleh dibuat sesuai keinginan kalian, tampilan pada spesifikasi ini hanya merupakan contoh.
2. **Diwajibkan** untuk membuat **driver** untuk masing-masing ADT. Driver berisi sebuah main file yang memanggil fungsi/prosedur yang ada di ADT tersebut. Kegunaan driver adalah untuk *testing* ADT yang sudah dibuat.
3. Sebagai saran, manfaatkan **Makefile** untuk mempermudah proses kompilasi dan penjalanan program. Bila sulit dalam menggunakan **Makefile**, bisa diakali dengan menggunakan **shell script/batch file**.
4. Gunakan **GitHub** sebagai *version control*, lalu undang asisten kalian sebagai *collaborator*. Pastikan asisten sudah masuk ke dalam *repository* sebelum asistensi pertama.
5. Buat *file* README yang minimal mengandung deskripsi singkat program, identitas anggota kelompok dan cara kompilasi program. Readme dapat dibuat dengan menggunakan [markdown](#).
6. Buat struktur program yang serapi mungkin. Jangan buat semuanya pada *file* yang sama. Contoh struktur program (tidak harus diikuti):



7. Manfaatkan ADT yang sudah kalian buat dalam praktikum semaksimal mungkin.
8. Perhatikan bahwa nilai untuk bonus akan **lebih kecil** dibandingkan dengan fitur utama. Silakan prioritaskan fitur-fitur yang lebih penting terlebih dahulu.
9. Jika ada yang kurang jelas, silahkan bertanya melalui [QnA](#).

Extras

"Semua soal susah di prak itu buatan Rayhan Maheswara Pramanda "

-Abil-

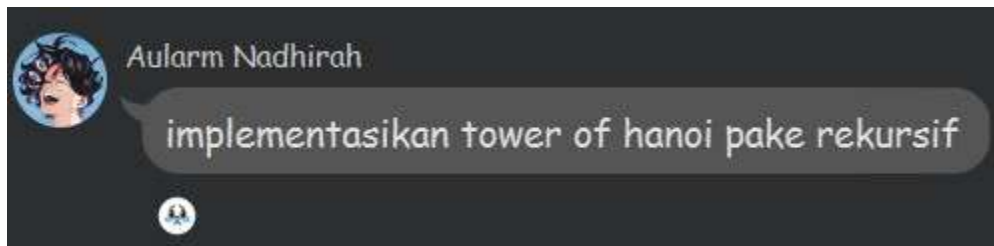
"Fitnah ini"

-Ray-



"Semangat para STI 23, kalian pasti bisa! $L(*^{\nabla}^*L)$ "

-Aul-



Preview prak selanjutny

-Aularm-

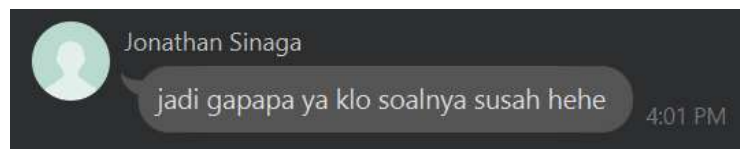
"Fitnah ini"

-Aul-



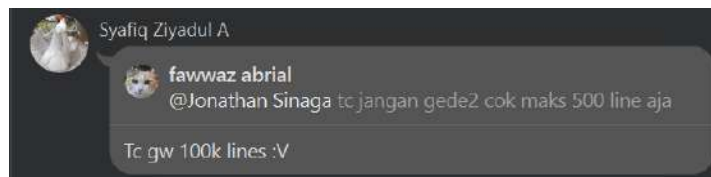
"Cipher C-nya cemangat dan cinta C \(_>_<_),♡"

-Ray-



"Pelaku sebenarnya guys"

-asisten yang bukan jon-



"Barang bukti"

-asisten yang bukan sapiq-

"Alangkah baiknya setelah *hunting* 11.11, mengerjakan tubes alstrukdat"

-asisten tercinta-

Semangat guys kalian pasti bisa!