



TUGAS BESAR
IF2010 PEMROGRAMAN
BERORIENTASI OBJEK



BY K02-A IF2010



K02-A



RAHMAT
PUJIATNO
18219062

- PLATESTORAGE,
INGREDIENTSTORAGE,
TRASHSTATION,
SERVINGCOUNTER



EMIR MUHAMMAD
FIRASSIYAN
18224066

- MAIN, FLOW, TIMER, MAP,
POSITION, TILE, GUI



SALSABILA
SHOFIYAH
18224088

- CUTTINGSTATION,
COOKINGSTATION,
ASSEMBLYSTATION,
WASHINGSTATION,
COOKINGDEVICE,
KITCHENUTENSILS,
PLATE



NICHOLAS
PUTRA HALIM
18224098

- ENTITY, CHEF, INPUTHANDLER,
DIRECTION, COLLISIONCHECKER



GIFTA MOOIRA
PRASHARNISSA
18224106

- ITEM, INGREDIENT, DISH,
PREPARABLE, RECIPE, ORDER,
ORDERLIST





USER MANUAL



NimonsCooked adalah game cooking simulation berbasis Java dimana pemain berperan sebagai chef dan harus menyelesaikan pesanan pelanggan dengan memasak berbagai hidangan dalam waktu terbatas.

ALUR GAME

Pemain mengambil ingredient dari storage, lalu ke cutting station untuk memotong bahan, cooking station untuk memasak, assembly station untuk menyusun dish di plate, lalu serve dish di serving counter. Washing station digunakan untuk membersihkan plate kotor. Trash station untuk membuang item atau dish yang salah.



OBJECTIVE



Selesaikan sebanyak mungkin order sebelum waktu habis untuk mendapatkan score tertinggi!



USER MANUAL

CARA MEMAINKAN GAME

General Information:

W W untuk menggerakkan chef ke atas

A A untuk menggerakkan chef ke kiri

S S untuk menggerakkan chef ke bawah

D D untuk menggerakkan chef ke kanan

E E untuk mengambil/meletakkan item

SPACE SPACE untuk Switch Chef

SHIFT Shift untuk dash

Interaksi Station:

C C untuk interact dengan cutting station, cooking station, dan washing station

E E untuk interact dengan assembly station, serving station, dan plate storage

Tampilan Layar Game:





DESKRIPSI GAMEPLAY

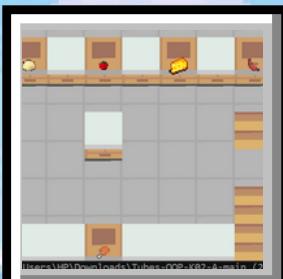
**MAIN MENU**

Tampilan pertama dalam game NimonsCooked. Terdapat opsi yang dapat dipilih :

- New Game untuk memulai permainan baru
- How to Play untuk petunjuk bermain
- Quit untuk keluar dari permainan

**KITCHEN VIEW**

Ketika memulai permainan baru, muncul tampilan kitchen yang memiliki dua chef, cutting station, cooking station, assembly station, washing station, trash station, serving station, ingredient storage, dan lain-lain.

**INGREDIENT STORAGE**

Terdapat beberapa ingredient (adonan, tomat, keju, sosis) yang diletakkan di ingredient storage. Klik E pada keyboard untuk mengambil ingredient dari ingredient storage.

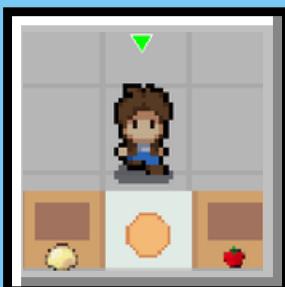
E**CUTTING STATION**

Semua ingredients harus dipotong sebelum dimasak. Untuk mempotong ingredients, gunakan Cutting Station dengan klik C pada keyboard. Cutting selesai ketika progress bar penuh

C

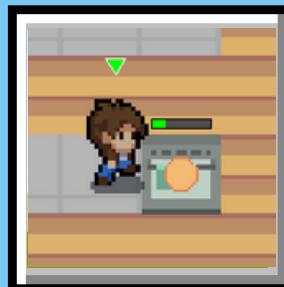


DESKRIPSI GAMEPLAY



ASSEMBLY STATION

Semua ingredients yang sudah dipotong (Chopped) pada cutting station, disusun di assembly station sesuai menu yang disajikan. Klik E pada assembly station untuk mulai menyusun

E

COOKING STATION

Setelah semua ingredients disusun, ingredients tersebut akan dimasak di cooking station dengan oven. Klik C pada cooking station untuk mulai memasak. Cooking selesai ketika progress bar penuh

C

PLATE STORAGE

Semua dish harus disajikan dalam piring sebelum diserve. Untuk mengambil piring, klik E pada plate storage lalu letakkan dish di atasnya. Dish yang sudah diletakkan pada piring sudah dapat diserve.

E

SERVING COUNTER

Dish yang sudah diletakkan pada piring kemudian diserve pada serving counter. Serving order tidak harus sesuai pada urutan order. Setelah makanan diserve, chef akan mendapatkan point.

E



DESKRIPSI GAMEPLAY



WASHING STATION

Setelah piring digunakan untuk serve dish, piring akan kotor dan harus dicuci di washing station. Untuk mencuci piring, klik C saat berada pada washing station. Piring akan menjadi bersih dan dapat digunakan kembali untuk serve.

C

TRASH STATION

Jika player salah memasak resep dan ingin membuang ingredients atau dish, keduanya dapat dibuang di trash. Untuk membuang, klik E saat berada pada trash station. Tidak dapat membuang kitchen utensils seperti piting dan oven.

E

ORDER LIST

Pada bagian kanan layar, terdapat order list dan durasi waktu untuk menyelesaikan order tersebut. Rincian ingredients setiap menu juga dapat dilihat di bawah setiap menu.

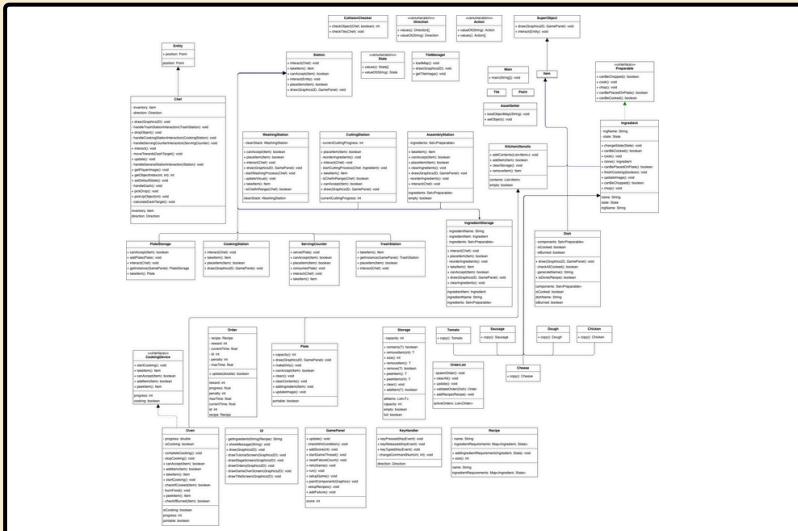


SCORING SYSTEM

Untuk memenangkan permainan, skor harus mencapai 300 point. Setiap player berhasil menyelesaikan 1 order, point bertambah 100 point. Apabila gagal, skor dikurangi 20 poin.



CLASS DIAGRAM



Class diagram menggambarkan struktur sistem NimonsCooked dengan Entity sebagai parent class yang mewariskan atribut position ke Chef, dimana Chef memiliki atribut inventory, direction, dan isAlive serta method drawGraphics, handleInput, dan update untuk mengatur koki. SuperObject, interface dengan method drawGraphics dan interactWith, diimplementasikan oleh Station, Tile, dan Item, dimana Station memiliki berbagai turunan seperti CuttingStation, WashingStation, AssemblyStation, dan CookingStation yang masing-masing memiliki atribut dan method spesifik. Item sebagai parent class diturunkan ke Point dan ContainerStation, sementara Ingredient sebagai parent diturunkan ke Dish yang memiliki turunan seperti Tomato, Sausage, Dough, and Chicken. Main berelasi dengan TileManager dan AssistantBar sebagai entry point aplikasi, Direction dan Action sebagai interface mengatur arah dan aksi, KitchenIntents mengelola item makanan yang berelasi dengan IngredientStorage, serta class pendukung seperti Order, Plate, Recipe, OrderList, GamePanel, UI, KeyHandler, Storage, Oven, dan Cheese untuk gameplay NimonsCooked.



DESIGN PATTERN

**BONUS**

Design pattern yang diimplementasikan pada game adalah sebagai berikut.

SINGLETON PATTERN

Pattern ini dipakai pada TrashStation dan PlateStorage supaya hanya ada satu instance yang digunakan seluruh game. Dengan begitu, tempat sampah dan penyimpanan piring punya state yang konsisten, lebih hemat memory, dan gampang diakses dari mana saja. Objeknya dibuat lewat getInstance(), bukan constructor biasa.

STRATEGY PATTERN

CookingStation tidak terikat pada satu cara memasak karena menggunakan interface CookingDevice. Oven hanyalah salah satu implementasinya.

Pattern ini membuat CookingStation fleksibel, karena bila ada perangkat masak baru (misalnya stove atau microwave), kita tinggal membuat class baru tanpa mengubah kode station.

Sebuah operasi memasak didelegasikan ke objek device tersebut.

TEMPLATE METHOD PATTERN

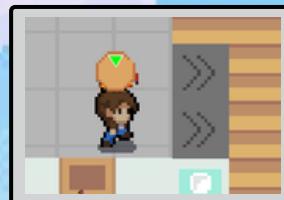
Abstract class Station menyediakan struktur dasar interaksi-misalnya cara menerima item, mengambil item, dan melakukan interact(). Subclass seperti CuttingStation, WashingStation, dan CookingStation hanya mengisi detail perilaku masing-masing dengan override method yang disediakan. Dengan cara ini, semua station mengikuti pola yang sama, tetapi tetap punya karakteristik unik tanpa menduplikasi kode.

PROTOTYPE PATTERN

Ingredient di storage bekerja dengan konsep prototipe. Setiap kali chef mengambil bahan, storage memberikan clone dari prototype ingredient yang disimpan. Ini membuat proses pembuatan item baru lebih cepat dan konsisten, serta memastikan state prototype tidak berubah meskipun bahan yang diambil telah dipotong atau dimasak di tempat lain.

**GUI****BONUS**

Menggunakan Java Swing dalam mengimplementasikan GUI dari game NimonsCooked seperti tampilan di bawah.





SOLID PRINCIPLE

BONUS

S

SINGLE RESPONSIBILITY PRINCIPLE (SRP)

Setiap class dibuat fokus pada satu tugas saja. Contohnya, CuttingStation hanya mengurus proses memotong, CookingStation hanya mengkoordinasikan proses memasak, WashingStation hanya mencuci, dan OrderList hanya menangani manajemen order. Dengan pemisahan tugas seperti ini, kode lebih mudah dirawat dan perubahan pada satu bagian tidak mengacaukan bagian lain.

O

OPEN/CLOSED PRINCIPLE (OCP)

Kode dapat ditambah fitur baru tanpa harus mengubah kode lama. Contohnya, Station sebagai abstract class sudah menyediakan struktur dasar, sehingga kita bisa membuat banyak station baru, seperti AssemblyStation atau ServingCounter, tanpa memodifikasi Station.java. Hal yang sama juga terjadi pada CookingDevice: kita bisa menambah Microwave atau Stove di masa depan tanpa mengubah CookingStation.

L

LISKOV SUBSTITUTION PRINCIPLE (LSP)

Semua subclass bisa digunakan sebagai pengganti superclass tanpa merusak program. Misalnya, seluruh station (Cutting, Cooking, Washing, Trash, PlateStorage, Assembly) bisa diperlakukan sebagai Station biasa. GamePanel dan Chef bisa berinteraksi dengan station apa pun lewat method yang sama, dan semuanya tetap bekerja karena perilaku dasarnya konsisten.





SOLID PRINCIPLE

BONUS



INTERFACE SEGREGATION PRINCIPLE (ISP)

I

Interface dipisah menjadi bagian kecil yang spesifik supaya class tidak dipaksa memakai method yang tidak relevan. Preparable hanya berisi operasi terkait bahan (chop, cook, dll.), sementara CookingDevice hanya berisi operasi perangkat memasak. Hasilnya, Ingredient tidak perlu punya method-device, dan Oven tidak perlu method chopping atau placing.

D

DEPENDENCY INVERSION PRINCIPLE (DIP)

Class level tinggi bergantung pada abstraksi, bukan implementasi konkret. CookingStation tidak bergantung pada Oven secara langsung, melainkan pada interface CookingDevice. Ini membuat CookingStation bebas dari detail implementasi dan memudahkan penambahan device baru. Sama halnya, Station bekerja dengan abstraksi Item, sehingga dapat menerima apa saja: Ingredient, Plate, atau KitchenUtensils.

DASH



BONUS

Menggunakan kombinasi tombol arah (W, A, S, D) dengan tombol Shift untuk meningkatkan jarak pergerakan chef.

W
A
S
D



SHIFT

Memperbesar jarak langkah chef dalam sekali tekan.



BEHIND THE SCENE



Kelompok kami, K02-A, terdiri dari Kak Rahmat, Emir, Nicholas, Salsa, dan Gifta. Pada Milestone 1, kami sempat berkumpul offline untuk membahas konsep dasar tugas, termasuk merancang class diagram dan membagi tugas agar penggerjaan lebih terarah. Memasuki Milestone 2, pekerjaan dilanjutkan dengan memodifikasi beberapa konsep dari tahap sebelumnya karena adanya tambahan requirement. Namun, di tahap ini kami tidak sempat berkumpul langsung karena bentrokkan dengan tubes lain, sehingga seluruh koordinasi dilakukan melalui GitHub dan group chat tanpa diskusi tatap muka. Selama keseluruhan proses, kami menjalani dua kali asistensi untuk memastikan arah penggerjaan sudah tepat. Big thanks kepada seluruh anggota kelompok, kakak asisten, Pak Riza selaku dosen OOP, dan Labpro yang telah membantu sepanjang penggerjaan tugas ini.



TUGAS BESAR
IF2010



LEVEL
JAVA



TYPE
OOP

NIMONSCOOKED

