

Tugas Besar Pemrograman Berorientasi Objek - Milestone 1
Perancangan Struktur Objek Nimonscooked (Map Tipe B: Pasta)
IF2010 - K02 - Pemrograman Berorientasi Objek



Disusun oleh:

| | |
|-------------------------|----------|
| Muhammad Difa'urrahman | 18224028 |
| Muhammad Syauqi Azizi | 18224046 |
| R. Athalla Ananda Putra | 18224060 |
| Francis Galton | 18224072 |

Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2025

DAFTAR ISI

BAB I

| | |
|-------------------------|----------|
| PENDAHULUAN..... | 3 |
|-------------------------|----------|

| | |
|-------------------------|---|
| 1.1 Deskripsi Umum..... | 3 |
|-------------------------|---|

| | |
|-----------------------------|---|
| 1.2 Tujuan Perancangan..... | 3 |
|-----------------------------|---|

BAB II

| | |
|-------------------------------|----------|
| ARSITEKTUR SISTEM..... | 4 |
|-------------------------------|----------|

| | |
|---|---|
| 2.1 Arsitektur MVC (Model-View-Controller)..... | 4 |
|---|---|

| | |
|------------------------------|---|
| 2.2 Class Diagram Utama..... | 4 |
|------------------------------|---|

BAB III

| | |
|---|----------|
| DETAIL PERANCANGAN KOMPONEN..... | 5 |
|---|----------|

| | |
|--|---|
| 3.1 View Package (GUI & Observer Pattern)..... | 5 |
|--|---|

| | |
|---------------------------------------|---|
| 3.2 Controller (Command Pattern)..... | 5 |
|---------------------------------------|---|

| | |
|---|---|
| 3.3 MODEL CORE (Game engine, world, orders, recipes)..... | 6 |
|---|---|

| | |
|--|---|
| 3.4 Map Builder (Builder Pattern)..... | 7 |
|--|---|

| | |
|---|---|
| 3.5 Chef Logic (State Pattern for actions)..... | 7 |
|---|---|

| | |
|---|---|
| 3.6 Items & Interfaces (Factory Pattern)..... | 8 |
|---|---|

| | |
|---|----|
| 3.7 Tile System (Map tiles & stations)..... | 10 |
|---|----|

| | |
|---------------------------|----|
| 3.8 Utility/Gameplay..... | 11 |
|---------------------------|----|

BAB I

PENDAHULUAN

1.1 Deskripsi Umum

Dokumen ini berisi rancangan game *Nimonscooked*. Sebuah permainan memasak yang berfokus pada pengelolaan dapur dalam waktu terbatas. Pemain mengendalikan seorang koki yang dapat bergerak di area dapur, mengambil bahan, mengolah makanan, menggunakan peralatan, hingga menyajikan pesanan yang masuk. Mekanisme permainan menuntut pemain untuk bergerak dengan cepat dan tepat.

Area dapur pada game ini terdiri dari berbagai objek seperti stove, chopping board, meja, storage, piring, dan alat memasak lainnya. Setiap objek memiliki fungsi tertentu, dan karakter dapat memotong, memasak, membawa bahan, atau merakit dish. Selain itu, pesanan pelanggan akan muncul secara berkala, sehingga pemain harus mengatur prioritas dan alur kerja di dapur agar tidak terjadi keterlambatan. Perilaku karakter dan objek dapur mengikuti aturan logika yang sudah disusun di bagian Model, termasuk perubahan state item, validasi resep, dan manajemen pesanan. Dengan begitu, interaksi pemain terasa lebih konsisten dan realistik dengan mekanik game yang dijalankan di belakang layar.

Secara keseluruhan, game ini dirancang menggunakan MVC agar logika permainan, tampilan visual, dan kontrol input terpisah. Model menangani seluruh logika inti dan data permainan, View bertugas menampilkan dunia game, dan Controller mengubah input pemain menjadi aksi melalui pola Command.

1.2 Tujuan Perancangan

Perancangan ini bertujuan untuk:

1. Mendefinisikan hubungan antar objek menggunakan prinsip *Object-Oriented Programming* (OOP).
2. Memisahkan logika permainan, tampilan visual, dan kontrol input menggunakan MVC.
3. Menerapkan *Design Pattern* untuk mendukung mekanik permainan dan struktur program

BAB II

ARSITEKTUR SISTEM

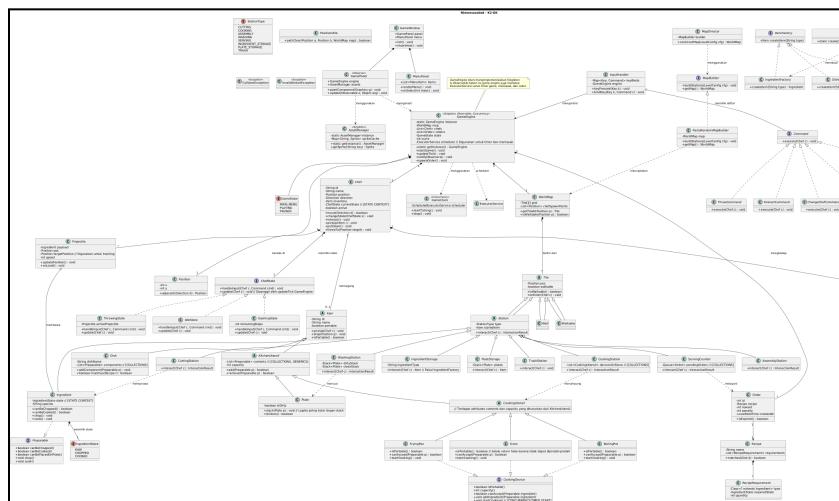
2.1 Arsitektur MVC (Model-View-Controller)

Sistem dibangun menggunakan pola arsitektur MVC untuk menjaga kode tetap terstruktur, mudah dikembangkan, dan memisahkan dengan jelas antara logika permainan, tampilan, serta pengolahan input

- **Model:** Mengatur logika permainan, seperti pergerakan chef, state karakter, proses memasak, item, station, map, resep, serta manajemen pesanan. Semua perubahan kondisi game diproses pada Model.
- **View:** Menampilkan dunia permainan melalui GUI. View hanya menggambar saat menerima notifikasi dari Model, sehingga tampilan mengikuti kondisi game secara real time.
- **Controller:** Mengolah input pemain dan mengubahnya menjadi Command seperti move, dash, interact, atau throw. Controller meneruskan perintah dari input player dan dieksekusi oleh Model.

2.2 Class Diagram Utama

Berikut adalah gambaran lengkap struktur kelas "Nimonscooked".



Lebih jelasnya dapat dilihat pada [Class Diagram OOP](#)

BAB III

DETAIL PERANCANGAN KOMPONEN

3.1 View Package (GUI & Observer Pattern)

Paket View menangani seluruh tampilan antarmuka game, termasuk window utama, panel dunia game, dan panel menu.

| Class | Tipe | Atribut | Method | Penjelasan |
|--------------|-------------------------------|--|--|--|
| GameWindow | Concrete | GamePanel panel MenuPanel menu | +init() : void +showView() : void | Window utama (GUI) untuk menginisialisasi panel, menu, dan listener |
| GamePanel | Concrete, implements Observer | GameEngine engine AssetManager assets | +paintComponent(Graphics g) : void +update(Observable o, Object arg) : void | Panel yang digunakan untuk merender world game, digunakan untuk mengobservasi GameEngine untuk redraw (Observer) |
| MenuPanel | Concrete | List<MenuItem> items | +renderMenu() : void +onSelect(int index) : void | Panel menu (Start/HowTo/Exit/StageSelect) |
| AssetManager | Concrete | private static AssetManager instance cache sprite/strings | +static getInstance() : AssetManager +getSprite(String key) : Sprite | Singleton untuk mengelola aset yang ada (sprite / teks / suara) |

3.2 Controller (Command Pattern)

Paket Controller bertanggung jawab menerjemahkan input pemain menjadi aksi melalui Command Pattern.

| Class | Tipe | Atribut | Method | Penjelasan |
|--------------|-----------------------------|--|--|--|
| InputHandler | Concrete | Map<Key, Command> keyBinds GameEngine engine | +keyPressed(Key k) : void +bindKey(Key k, Command c) : void | Menerjemahkan input pemain ke objek Command dan menjalankannya |
| Command | Interface | - | +execute(Chef c) : void | Interface command umum (W / A / S / D) |
| MoveCommand | Concrete implements Command | Direction dir | +execute(Chef c) : void | Perintah pindah satu langkah atau dash jika dikombinasikan |

| Class | Tipe | Atribut | Method | Penjelasan |
|-------------------|-----------------------------|---|-------------------------|---|
| DashCommand | Concrete implements Command | Direction dir int distance int cooldown | +execute(Chef c) : void | Perintah dash dengan bergerak lebih dari 1 tile dan ada handling cooldown |
| ThrowCommand | Concrete implements Command | Position target Int range | +execute(Chef c) : void | Melempar item dengan cara memvalidasi rute (tembok) lalu menciptakan Projectile |
| InteractCommand | Concrete implements Command | - | +execute(Chef c) : void | Interaksi dengan seperti station / pickup / putdown sesuai konteks tile depan. |
| ChangeChefCommand | Concrete implements Command | String chefId | +execute(Chef c) : void | Mengganti active chef |

3.3 MODEL CORE (Game engine, world, orders, recipes)

Bagian Model menjadi pusat logika permainan.

| Class | Tipe | Atribut | Method | Penjelasan |
|------------|----------|---|--|---|
| GameEngine | Concrete | private static GameEngine instance WorldMap map List<Chef> chefs List<Order> orders GameState state int score ExecutorService scheduler | +static getInstance() : GameEngine +startGame() : void +updateTick() : void +notifyObservers() : void +spawnOrder() : void | Mengontrol game mulai dari loop, state, tick scheduler, observer notifier |
| GameState | Enum | MAIN_MENU, PLAYING, PAUSED | - | Berisi enum status game |
| WorldMap | Concrete | Tile[][] grid (14x10) List<Position> chefSpawnPoints | +getTileAt(Position p) : Tile +isWalkable(Position p) : boolean | Representasi peta dalam bentuk grid dan operasi query tile |
| Order | Concrete | int id Recipe recipe int reward int penalty LocalDateTime createdAt | +isExpired() : boolean | Unit satuan untuk pesanan pelanggan |
| Recipe | Concrete | String name List<RecipeRequirement> | +matches(Dish d) : boolean | Spesifikasi untuk recipe yang berisi |

| Class | Tipe | Atribut | Method | Penjelasan |
|-------------------|-------------|---|-----------------------------------|---|
| | | requirements | | RecipeRequirement untuk menjelaskan ingredient dan required state |
| RecipeRequirement | Concrete | Class<? extends Ingredient> type IngredientState requiredState int quantity | - | Unit requirement per recipe |
| Position | Value Class | int x, y | +adjacent(Direction d) : Position | Representasi koordinat |
| Direction | Enum | UP, DOWN, LEFT, RIGHT | - | Berisi enum untuk arah |

3.4 Map Builder (Builder Pattern)

Paket Map Builder menggunakan Builder Pattern untuk membangun peta permainan secara fleksibel berdasarkan konfigurasi level.

| Class | Tipe | Atribut | Method | Penjelasan |
|-----------------------|--------------------------------|--------------------|--|---|
| MapDirector | Concrete | MapBuilder builder | +constructMap(LevelConfig cfg) : WorldMap | Mengarahkan proses pembuatan map dengan builder. |
| MapBuilder | Interface | - | +buildStations(LevelConfig cfg) : void +getMap() : WorldMap | Interface builder |
| PastaRandomMapBuilder | Concrete implements MapBuilder | WorldMap map | +buildStations(LevelConfig cfg) : void +getMap() : WorldMap | Builder spesifik untuk Pasta Map. Disini akan diterapkan generator random dan juga validation |

3.5 Chef Logic (State Pattern for actions)

Class pada tabel dibawah ini digunakan untuk perilaku chef dengan *State Design Pattern*. Adapun rincian setiap classnya adalah sebagai berikut.

| Class | Tipe | Atribut | Method | Penjelasan |
|-------|----------------|--|--|---|
| Chef | Concrete Class | String: id String: name Position: position Direction direction Item inventory (nullable) | +move(Direction d) : boolean +changeState(ChefState s) : void +interact() : void | Representasi pemain. Semua perubahan posisi/inventori harus atomik dan juga synchronize untuk |

| | | | | |
|---------------|----------------|---|---|---|
| | | ChefState currentState boolean active | +pickUp(Item i) : void +putDown() : void +throwTo(Position target) : void | concurrency |
| ChefState | Interface | - | +handleInput(Chef c, Command cmd) : void +update(Chef c) : void | State pattern interface |
| IdleState | Concrete Class | - | +handleInput(...) : void | State default yang digunakan menerima perintah normal |
| DashingState | Concrete Class | int remainingSteps | +update(Chef c) : void | State sementara saat dash berlangsung |
| ThrowingState | Concrete Class | Projectile activeProjectile | +update(Chef c) : void | State saat melempar. Handles cooldown atau animasi |
| Projectile | Concrete Class | Ingredient payload Position pos Position dir int speed | +updatePosition() : void +onLand() : void | Representasi objek lempar yang bergerak beberapa tick |

3.6 Items & Interfaces (Factory Pattern)

Sistem item menggunakan kombinasi inheritance dan Factory Pattern untuk menciptakan berbagai objek permainan.

| Class | Tipe | Atribut | Method | Penjelasan |
|-----------------|--|--|--|--|
| Item | Abstract Class | String id String name boolean portable | +pickUp(Chef c) : void +drop(Position p) : void +isPortable() : boolean | Superclass semua item / benda fisik |
| Preparable | Interface | - | +boolean canBeChopped() +boolean canBeCooked() +boolean canBePlacedOnPlate() +void chop() +void cook() | Interface untuk Ingredient yang dapat diproses |
| Ingredient | Concrete extends Item, implements Preparable | IngredientState state String species | +canBeChopped() : boolean +canBeCooked() : boolean +chop() : void +cook() : void | Ingredient dasar |
| IngredientState | Enum | RAW, CHOPPED, COOKED | - | Status untuk ingredient |
| Dish | Concrete extends Item | String dishName List<Preparable> components | +addComponent(Preparable p) : void +boolean matches(Recipe r) : | Hidangan akhir yang akan dibuat melalui DishFactory. |

| Class | Tipe | Atribut | Method | Penjelasan |
|----------------|---|--|--|---|
| | | | boolean | |
| KitchenUtensil | Abstract extends Item | List<Preparable> contents int capacity | +add(Preparable p) : boolean +remove(Preparable p) : boolean | Superclass untuk alat makan / masak |
| Plate | Concrete extends KitchenUtensil | boolean isDirty | +stack(Plate p) : void +isClean() : boolean | Piring dapat disajikan atau dicuci jika isDirty bernilai true |
| CookingUtensil | Abstract extends KitchenUtensil | - | (inherits) | Superclass untuk pot / pan / oven |
| CookingDevice | Interface | - | +boolean isPortable() +int capacity() +boolean canAccept(Preparable ingredient) +void addIngredient(Preparable ingredient) +void startCooking() : void | Interface alat masak |
| BoilingPot | Concrete extends CookingUtensil implements CookingDevice | int capacity | +isPortable() : boolean +canAccept(Preparable p) : boolean +addIngredient(Preparable p) : void +startCooking() : void | Hanya menerima pasta / rice. Bersifat portable. |
| FryingPan | Concrete implements CookingDevice | int capacity | +isPortable() : boolean +canAccept(Preparable p) : boolean +addIngredient(Preparable p) : void +startCooking() : void | Wajan untuk bahan CHOPPED tertentu |
| Oven | Concrete implements CookingDevice | int capacity | +isPortable() : boolean +canAccept(Preparable p) : boolean +addIngredient(Preparable p) : void +startCooking() : void | Oven bersifat tidak portable |

Factory Classes

| Class | Tipe | Atribut | Method |
|------------------|---------------------------------|--|---|
| ItemFactory | Interface | +Item createItem(String type) : Item | Factory bersifat generic |
| IngredietFactory | Concrete implements ItemFactory | +Item createItem(String type) : Ingredient | Membuat Ingrediet (Tomato, PastaRaw, Meat, Shrimp, Fish). |
| UtensilFactory | Concrete implements | +Item createItem(String type) : | Membuat Plate, BoilingPot, |

| Class | Tipe | Atribut | Method |
|---------------------|---------------------------------|--|--|
| | ItemFactory | KitchenUtensil | FryingPan, Oven. |
| DishFactory | Concrete implements ItemFactory | +Item createItem(String type) : Dish | Membuat Dish berdasarkan Recipe |
| ItemFactoryProvider | Utiliy | +static Item create(String category, String type) : Item | Master factory yang memanggil factory sesuai kategori. |

3.7 Tile System (Map tiles & stations)

Tile System mendefinisikan berbagai jenis tile pada permainan.

| Class | Tipe | Atribut | Method | Penjelasan |
|-----------------|--------------------------|--|--|---|
| Tile | Concrete Class | Position pos boolean walkable | +isWalkable() : boolean +onEnter(Chef c) : void | Base untuk tile |
| Wall | Abstract | - | isWalkable() : false | Dinding/obstacle |
| Walkable | Concrete extends Tile | - | isWalkable() : true | Tile kosong |
| Station | Abstract extends Tile | StationType type Item storedItem (nullable) | +interact(Chef c) : InteractionResult | Basis station yang dapat diinteract untuk memproses logic |
| CuttingStation | Concrete extends Station | | +interact(Chef c) : InteractionResult | Memotong ingredient (mengubah state menjadi CHOPPED) dapat menerima/menaruh item. |
| CookingStation | Concrete extends Station | List<CookingUtensil> devicesOnStove | +interact(Chef c) : InteractionResult | Tempat memasak.; Jika ada pot / pan di atasnya maka dapat memroses pemasakan. |
| AssemblyStation | Concrete extends Station | | +interact(Chef c) : InteractionResult | Untuk plating dish. |

| Class | Tipe | Atribut | Method | Penjelasan |
|-------------------|--------------------------|--|---------------------------------------|--|
| WashingStation | Concrete extends Station | Stack<Plate> dirtyStack Stack<Plate> cleanStack | +interact(Chef c) : InteractionResult | Mencuci plate, mengubah state dari dirty menjadi clean. |
| ServingStation | Concrete extends Station | Queue<Order> pendingOrders | +interact(Chef c) : InteractionResult | Menyajikan dish. Jika sesuai order akan diberi reward lalu spawn order baru. |
| IngredientStorage | Concrete extends Station | String ingredientType | +interact(Chef c) : Item | Menghasilkan ingredient RAW |
| PlateStorage | Concrete extends Station | Stack<Plate> plates | +interact(Chef c) : Item | Menyediakan piring bersih / kotor |
| TrashStation | Concrete extends Station | | +interact(Chef c) : void | Menghapus item dari permainan |

3.8 Utility/Gameplay

Bagian Utility menyediakan fungsi pendukung untuk mekanik gameplay.

| Class | Tipe | Atribut | Method | Penjelasan |
|------------------------|-------------------------------|--------------------------|--|---|
| PositionUtils | Utility | - | +pathClear(Position a, Position b, WorldMap map) : boolean | Validasi jalur pelemparan (untuk Throw) |
| CollisionException | Exception (extends Exception) | - | - | Dilempar saat gerakan tidak valid (tile terblok) |
| InvalidActionException | Exception | - | - | Dilempar jika interaksi tidak dimungkinkan (misalnya: plating di plate kotor) |
| GameClock | Concrete (uses concurrency) | ScheduledExecutorService | +startTicking() +stop() | Mengatur tick game (physics / projectile) |

| Class | Tipe | Atribut | Method | Penjelasan |
|--------------|-------------|----------------|---------------|---------------------------|
| | | | | updates / cooking timers) |