

## **BAB 3**

### **METODE PENELITIAN**

Untuk mendapatkan hasil yang baik dalam sebuah penelitian, diperlukan perencanaan yang rapi, pengelolaan yang benar, pengolahan berbagai kebutuhan penelitian dan penggunaan metode yang tepat. Pembahasan mengenai metode penelitian sangat dibutuhkan dalam penelitian, berikut metode penelitian yang digunakan :

#### **3.1 Metode Pengumpulan Data**

Teknik pengumpulan data diperlukan dalam pembangunan Web pendukung menuju konsep Kota Pintar (*Smart City*), Teknik tersebut adalah :

1. Teknik Observasi

Teknik Observasi dilakukan secara langsung dengan pengamatan di jalan raya mengenai objek penelitian, lingkungan, serta sikap dan kebiasaan yang dilakukan pengendara di jalan raya untuk memenuhi kebutuhan informasi dan data dalam perancangan Web pendukung menuju konsep Kota Pintar (*Smart City*).

2. Teknik Interview

*Interview* dilakukan kepada beberapa responden yang berhubungan dengan penggunaan jalan raya, dengan mengajukan beberapa pertanyaan mengenai perilaku pengendara di jalan raya beserta pengalaman-pengalaman pribadi tentang kejadian pelanggaran lalu lintas di jalan raya disekitarnya sehingga didapatkan data yang mendukung fungsi Web pendukung menuju konsep Kota Pintar (*Smart City*).

3. Teknik Studi Pustaka

Studi Pustaka dilakukan dengan mencari beberapa penelitian, sumber terpercaya mengenai pentingnya konsep Kota Pintar (*Smart City*), WEB dengan *Responsive Website* dan metode rekayasa perancangan perangkat lunak untuk mendapatkan acuan bagaimana membangun pendukung menuju konsep Kota Pintar (*Smart City*) dengan baik dan benar.

### 3.2 Jenis Data

Data yang didapatkan masing-masing memiliki jenis, dalam bentuk apa data tersebut didapatkan dan dengan cara apa data dihasilkan mempengaruhi jenis setiap data yang dibutuhkan untuk membangun Web pendukung menuju konsep Kota Pintar (*Smart City*) ini, diantaranya adalah :

#### 1. Data Kuantitatif

Data Kuantitatif merupakan data yang berhubungan dengan angka ataupun bilangan yang berkaitan dengan Web pendukung menuju konsep Kota Pintar (*Smart City*) untuk memperkuat alasan pentingnya membangun Web yang dapat merubah sikap dan perilaku mengemudi di jalan raya. Seperti statistik jumlah kecelakaan yang terjadi di Indonesia untuk meyakinkan bahwa banyak korban yang meninggal di jalan raya akibat kecelakaan, selain itu juga ada peringkat faktor penyebab kecelakaan dimana data menunjukkan pelanggaran lalu lintas menjadi penyebab tertinggi terjadinya kecelakaan.

#### 2. Data Kualitatif

Data kualitatif juga diperlukan dalam penelitian ini sebagai pendukung dalam perancangan Web pendukung menuju konsep Kota Pintar (*Smart City*), data berupa analisis dokumen seperti penelitian terkait yang membahas pentingnya konsep Kota Pintar (*Smart City*), pembuatan Web dengan *responsive Website*, metode pengembangan perangkat lunak hingga mengenai pelanggaran lalu lintas dimana karena pelanggaran lalu lintas menjadi penyebab tertinggi sehingga faktor manusialah yang menjadi alasan pelanggaran lalu lintas terjadi. Selain itu *interview* menghasilkan data yang menjadi petunjuk dalam menentukan kategori-kategori pelanggaran lalu lintas yang sering terjadi.

### 3.3 Sumber Data

#### 1. Data Primer

Data primer didapatkan dengan observasi secara langsung dan *interview* kepada beberapa orang yang berhubungan dengan Web yang akan di bangun, sehingga keaslian dan keakuratan data terjamin dengan melihat hasil observasi

secara langsung maupun *interview* yang dilakukan kepada orang-orang tertentu yang berhubungan dengan Web.

## 2. Data Sekunder

Data sekunder diperlukan dalam pemahaman masalah, penjelasan masalah, memberikan alternatif penyelesaian masalah, solusi yang ingin di ciptakan. Dengan memberikan data-data sebagai bukti kebenaran pentingnya mendukung konsep Kota Pintar (*Smart City*).

### 3.4 Populasi

Setiap penelitian pasti memiliki subjek yang berhubungan dengan apa yang menjadi pembahasan dalam penelitian, subjek tersebut merupakan populasi dari penelitian. Penelitian ini memiliki subjek Kota Pintar (*Smart City*) dimana konsep tersebut sangat berkaitan erat dengan beberapa hal, dalam penelitian ini hal tersebut adalah para pengguna jalan raya. Seperti pengendara, kepolisian, pejalan kaki, pedagang kaki lima dan masih banyak lagi yang bersangkutan dengan pengguna jalan raya yang memungkinkan bersangkutan dalam kejadian pelanggaran lalu lintas, mulai dari penyebab siapa yang melakukan pelanggaran hingga kepada siapa dampak akibat pelanggaran peraturan lalu lintas yang terjadi di jalan raya.

#### 3.4.1 Teknik Sampling Bertujuan

Teknik *Sampling* bertujuan atau biasa disebut dengan *purposive sampling* adalah pengambilan sample pada sebuah populasi yang dilakukan untuk mencapai sebuah tujuan. Pada web tertib berlalu lintas memiliki populasi seperti pengendara, kepolisian, pejalan kaki atau bisa disebut dengan pengguna jalan. Sedangkan teknik *perposive sampling* ini dilakukan untuk mencapai tujuan berikut ini :

1. Mengetahui apakah web tertib berlalu lintas dapat memberikan dampak sosial terhadap cara mengemudi kendaraan bermotor.
2. Mencari sebuah cara dan pendekatan yang dapat digunakan untuk memberikan efek jera dan perubahan cara mengemudi pelanggar peraturan lalu-lintas.

3. Memastikan bahwa cara dan pendekatan yang digunakan sudah sesuai dengan yang diinginkan kebanyakan pengguna.

### **3.5 Ruang Lingkup Penelitian**

Sistem ini merupakan Web yang dapat berjalan pada *smartphone* dan juga dapat dijalankan pada PC dengan kata lain, Web ini merupakan situs yang *responsive* atau dapat ditampilkan dengan baik di resolusi layar yang berbeda-beda, rekayasa perangkat lunak yang dilakukan penelitian ini menggunakan *ICONIX Process*, hingga nantinya aplikasi kampanye tertib berlalu lintas menjadi Web pendukung menuju konsep Kota Pintar (*Smart City*) dalam merubah kebiasaan masyarakat dalam berkendara untuk menjadi pengendara yang baik.

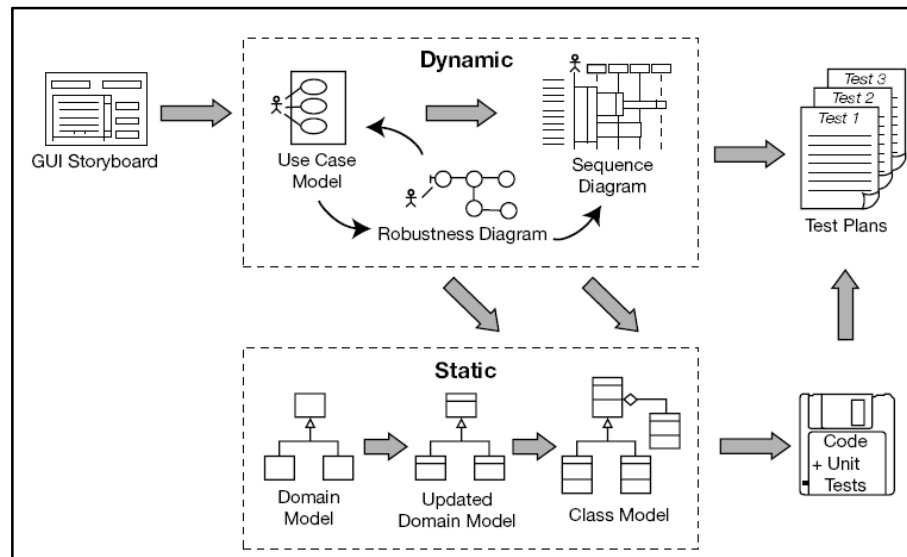
### **3.6 Metode Pengembangan Perangkat Lunak**

Penelitian ini membutuhkan metode pengembangan perangkat lunak untuk mempermudah dalam menemukan objek-objek yang berkaitan dengan Web yang akan di bangun, serta mempermudah dalam perancangan alur sistem yang akan berjalan dan siapa saja yang bersangkutan selama proses sistem berlangsung

#### **3.6.1 ICONIX Process**

*ICONIX Process* digunakan sebagai panduan untuk membantu dalam pembangunan serta perancangan Web pendukung menuju konsep Kota Pintar (*Smart City*) untuk menghasilkan aplikasi yang baik dan cepat dari segi waktu yang dibutuhkan untuk pembangunannya. Dalam penggunaan UML pada *ICONIX Process* pun tidak berlebihan, karena hanya membutuhkan beberapa diagram saja yang dianggap sudah cukup untuk melakukan analisa perancangan berbasis objek.

Pemodelan ini diawali dengan pembuatan model abstrak sesuai dengan dunia nyata berupa objek-objek yang bersangkutan dengan Web pendukung menuju konsep Kota Pintar (*Smart City*) yang akan dibangun. Hingga pemodelan menjadi sebuah Web dengan analisa perancangan berbasis objek menggunakan *ICONIX Process* yang digambarkan pada gambar dibawah ini[23].



**Gambar 3.1 ICONIX Process**

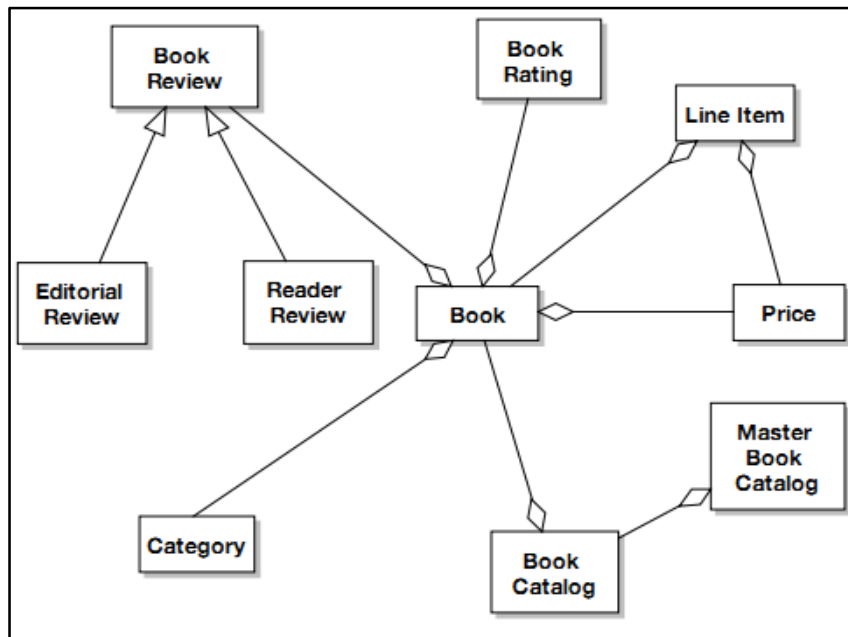
Dari gambar tersebut nampak terdapat beberapa proses yang harus dilakukan secara berurutan untuk membangun serta perancangan Web Tertib Berlalu Lintas, maka tahap tersebut terdiri dari proses berikut ini :

### 1. Functional Requirement Analysis

Tahap pertama dalam perancangan Web pendukung menuju konsep Kota Pintar (*Smart City*) dilakukan analisis kebutuhan fungsional secara detail untuk menentukan fungsi-fungsi yang akan tersedia pada aplikasi. Kebutuhan Fungsional yang sudah ditetapkan diberikan kode dan nomor untuk membuat dokumen perancangan menjadi lebih terstruktur dan rapi. Hal tersebut sangat penting dilakukan karena dengan kode dan nomor membuat deskripsi kebutuhan fungsional yang detail dan besar akan menjadi lebih mudah untuk melanjutkan ke tahap-tahap berikutnya secara berurutan.

### 2. Domain Modelling

Perancangan Web pendukung menuju konsep Kota Pintar (*Smart City*) pada tahap kedua ini adalah dengan membuat model dengan domain diagram dimana domain model ini menjadi awal dari permodelan secara statis yang menggambarkan objek-objek yang saling berkaitan dengan aplikasi dan dunia nyata [23].



**Gambar 3.2 Contoh Domain Model**

Gambar diatas merupakan contoh dari domain model yang merepresentasikan objek dan relasi yang saling berkaitan. Objek-objek tersebut telah ditetapkan pada proses analisis kebutuhan fungsional. Dari kebutuhan fungsional dan domain model yang telah terbentuk diharapkan pembangunan Web pendukung menuju konsep Kota Pintar (*Smart City*) tetap berada pada ruang lingkungannya, dalam arti ruang lingkup tidak meluas dan tidak terlalu kecil sehingga konsep Kota Pintar (*Smart City*) terwujud dengan baik dan jelas. Dalam membangun Domain model terdapat pedoman berupa prinsip-prinsip membangun domain model, yaitu :

- a) Analisa diharapkan untuk tetap fokus terhadap objek yang ada di dunia nyata yang berhubungan dengan permasalahan.
- b) Gunakan generalisasi (*is-a*) dan agregasi (*has-a*) untuk menggambarkan hubungan antar model dimana hubungan tersebut menggambarkan mana yang termasuk komponen dan mana yang termasuk atribut.
- c) Dalam memodelkan domain model harus membatasi waktu pengerjaan, agar tidak terlalu lama untuk beranjak ke langkah selanjutnya karena domain model akan berkembang disetiap prosesnya.

- d) Mengelola kata kunci *abstract* atau kandidat domain objek sebagai domain objek untuk dijadikan domain model.
- e) Pada saat membuat Domain Model jangan menganggap bahwa domain objek merupakan *class* objek.
- f) Pembuatan domain model hanya berfokus untuk mencari kata kunci yang dapat dijadikan domain objek yang berhubungan dengan permasalahan bukan pembuatan *database* yang berisi kumpulan beberapa hal dengan beberapa atribut.
- g) Gunakan domain model sebagai *glossary* proyek, yang memberikan penjelasan yang mudah dimengerti sekaligus berhubungan dengan permasalahan.
- h) Domain model bisa dikatakan untuk menghindari ambiguitas nama agar dapat melanjutkan ke tahap selanjutnya, maka dari itu harus dilakukan di awal.
- i) Jangan pernah berharap bahwa domain model yang dibuat sama persis dengan *Class* diagram nantinya, namun setidaknya pasti ada sedikit kemiripan.
- j) Pastikan membuat domain yang berhubungan dengan permasalahan bukan yang berhubungan dengan implementasi GUI.

### 3. Story Board The User Interface

ICONIX *Process* menganjurkan untuk tetap membuat gambaran berupa antarmuka pengguna, dengan desain secara kasar memberikan kemudahan untuk menentukan aktifitas pengguna dan di gambarkan dengan *Use Case* diagram.

### 4. Use Case Modelling

Memodelkan *Use Case* merupakan tahapan dimana membentuk sebuah diagram yang menggambarkan aktifitas yang dapat dilakukan dengan Web pendukung konsep Kota Pintar (*Smart City*) dan siapa saja aktor yang terlibat. Berikut adalah tahapan dalam memodelkan *Use Case* yang di tawarkan ICONIX *Process* :

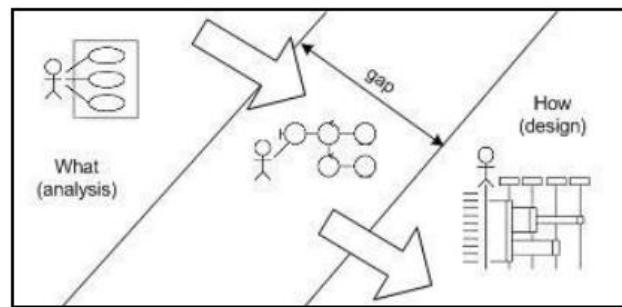
- a. Membuat kerangka kerja skenario *Use Case* yang berisi *normal flow* dan *alternate flow*.
- b. Menanyakan, “Apa yang akan terjadi?” pertanyaan ini akan menjadi pertanyaan yang menjawab tindakan awal yang akan dilakukan.
- c. Menanyakan, “Lalu apa yang terjadi?” terus menerus menanyakan apa yang akan terjadi sehingga mendapatkan detail dari aktifitas yang diinginkan.
- d. Menanyakan, “Apa lagi yang bisa terjadi?” lebih detailnya lagi untuk memunculkan aktifitas-aktifitas yang tidak terduga.

Mendefinisikan aktor menjadi langkah pertama yang harus dilakukan untuk memodelkan *Use Case*. Setelah aktor dan istilahnya telah ditetapkan tahap berikutnya adalah mendefinisikan aktifitas-aktifitas yang dilakukan aktor yang sudah ditetapkan. *ICONIX Process* menganjurkan untuk membuat desain antar muka pengguna atau *GUI Story Board* yang membantu memudahkan pemodelan *Use Case*.

## 5. Robustness Analysis

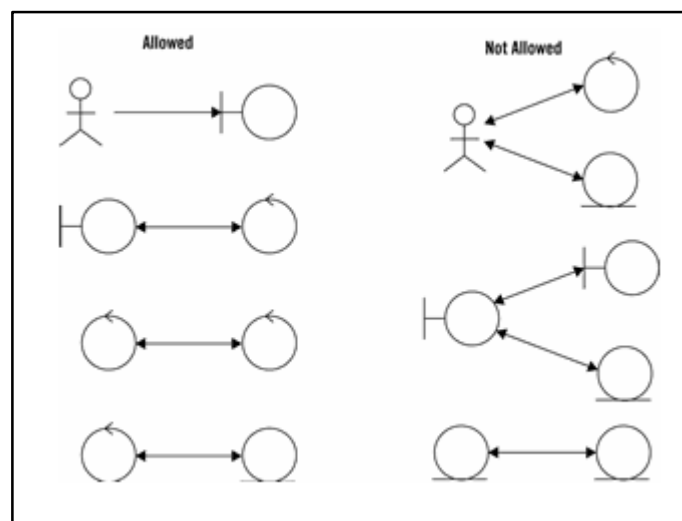
*Robustness analysis* menjadi tahap yang menggambarkan detail setiap kalimat dari skenario *use case* yang sudah terbentuk, setiap *use case* memiliki skenario yang berbeda-beda, dengan *robustness analysis* maka didapatkan dengan mudah objek-objek yang berhubungan dengan aplikasi yang akan di bangun. *Robustness analysis* menjembatani apabila terjadi kekurangan atau kesenjangan pada deskripsi skenario *use case* sehingga di buat menjadi lebih baik untuk tahap yang selanjutnya [23].





**Gambar 3.3 jembatan antara desain dan analisa**

Langkah *Robustness analysis* menjadi jembatan antara *use case* yang menggambarkan apa yang dapat dilakukan dengan aplikasi pendukung konsep Kota Pintar (*Smart City*) dan bagaimana desain yang dirancang untuk skenario setiap fungsi aplikasi. *Robustness Diagram* juga memiliki aturan dalam pembuatannya dimana aktor hanya boleh berhubungan langsung dengan *boundary*, *boundary* dengan *controller*, *controller* dengan *controller*, *controller* dengan *entity*. Seperti yang terlihat gambar di bawah ini [23].



**Gambar 3.4 Aturan Robustness Analysis**

## 6. Sequence Diagramming

Setelah melalui *robustness analysis* tahap selanjutnya adalah membuat *sequence diagram*, diagram ini menggambarkan secara rinci proses yang terjadi sesuai objek dengan skenarionya. Tidak boleh terdapat proses yang

terlewati saat membuat *sequence diagram*, *sequence diagram* mengacu pada skenario setiap *use case* mulai dari *normal flow* hingga *alternate flow*.

#### **7. Updated Domain Model**

Pada tahap ini memastikan apakah design *robustness* dan *sequence diagram* menampilkan objek baru atau terdapat objek yang hilang. Domain model akan mengalami perubahan jika terdapat objek yang seharusnya dihilangkan atau ditambahkan.

#### **8. Finalize the Class model**

Perancangan *class diagram* menjadi lebih rinci dibandingkan domain diagram, diagram ini adalah gambaran mengenai database yang saling berkaitan. *Class diagram* memiliki informasi yang lebih detail karena mempunyai atribut dan operasi-operasi yang didapatkan setelah menggambarkan *sequence diagram* untuk masing-masing *use case*.

### **3.6.2 Implementation and Testing**

Desain perancangan yang sudah lengkap dengan beberapa *use case* beserta skenarionya hingga *class diagram* yang sudah dilengkapi dengan beberapa atribut dan operasi yang didapatkan dari pemodelan *robustness diagram* dan *sequence diagram* maka perancangan diimplementasikan dengan memulai menyusun menjadi kode program untuk membangun Web pendukung menuju konsep Kota Pintar (*Smart City*).

#### **a. Black-box Testing**

Pengujian perangkat lunak dilakukan setelah kode program disusun untuk membangun aplikasi, pengujian yang digunakan adalah *Black-Box Testing*, pengujian ini dilakukan untuk memeriksa fungsionalitas dari aplikasi tanpa harus mengujikan kode program. *Black-box Testing* juga digunakan untuk mengetahui setiap detail dari proses *input* dan *output* berjalan dengan normal dan sesuai dengan deskripsi yang sudah ditentukan.

## b. User Acceptance Testing

UAT merupakan cara formal yang dilakukan untuk menegaskan bahwa Web Tertib Berlalu-lintas benar-benar sudah memenuhi kebutuhan pengguna. UAT dilakukan berdasarkan kebutuhan fungsional dan dilakukan kepada pengguna yang bersangkutan dengan lalu lintas, seperti pengendara kendaraan bermotor, dan pejalan kaki yang menggunakan trotoar dan penyeberangan jalan.

UAT dilakukan dengan memberikan skala *likert* dengan angka 1 hingga 5. Dimana 1 merupakan skala jawaban sangat kurang, serta 5 merupakan skala jawaban sangat baik. Hingga didapat persentasi yang memiliki kriteria hasil pengujian. Untuk mendapatkan kriteria hasil pengujian digunakan rumus skor ideal sebagai berikut :

$$\text{Skor Kriteria} = \text{Nilai Skala} \times \text{Jumlah Responden} \quad (3.1)$$

**Tabel 3.1 Skor Ideal kriteria**

Kriteria	Nilai Skala x Jumlah Responden	Skor
Sangat Kurang	1 x 20	< = 20
Kurang	2 x 20	< = 40
Cukup	3 x 20	< = 60
Baik	4 x 20	< = 80
Sangat Baik	5 x 20	< = 100

*User Acceptance test* dilakukan dengan mengajukan pertanyaan berdasarkan 5 sudut pandang [25], yaitu :

**Tabel 3.2 Sudut pandang user acceptance test**

Sudut Pandang	Maksud dan tujuan
<i>Web Design</i>	Mengetahui apakah berdasarkan tampilan dari web sudah memenuhi keinginan pengguna.
<i>Reliability</i>	Mengetahui apakah web sudah memenuhi fungsi yang dibutuhkan pada waktu dan kondisi tertentu.

<i>Responsiveness</i>	Mengetahui seberapa respon dan kecepatan web untuk memenuhi kebutuhan pengguna.
<i>Trust</i>	Mengetahui tingkat kepercayaan pengguna dalam menggunakan web tertib berlalu lintas.
<i>Personalization</i>	Mengetahui seberapa besar manfaat yang diperoleh setelah menggunakan web dengan mengambil beberapa kesimpulan untuk mendapatkan keputusan.