

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Penelitian Sebelumnya**

Bahan referensi yang digunakan dalam penyusunan tugas akhir ini adalah beberapa penelitian terdahulu. Referensi untuk penyusunan tugas akhir didapat dari beberapa penelitian terdahulu yang tidak jauh dari topik penelitian mengenai perancangan sistem informasi pendataan penduduk berbasis *android*. Selain itu untuk menghindari anggapan kesamaan yang sudah dilaksanakan dan dikerjakan peneliti lain. Maka peneliti menulis beberapa hasil dari penelitian terdahulu.

Penelitian pertama yang berjudul “Perancangan Sistem Informasi Sensus Penduduk Berbasis *Android* Pada Desa Meranti” yang ditulis oleh Bella Ayu Listia, Iwan Purnama, Syaiful Zuhri Harahap pada tahun 2020. Penelitian ini memiliki tujuan untuk mengubah cara pendataan yang dilakukan dengan cara ditulis dibuku menjadi pendataan melalui aplikasi berbasis *android*. Pendekatan atau metode didalam penelitian ini adalah metode *Waterfall* dengan beberapa tahap yang biasa digunakan pada metode *Waterfall*. Hasil yang diperoleh penelitian ini salah satu contohnya pendataan penduduk yang semula masih dicatat melalui buku menjadi pendataan penduduk yang sudah menggunakan aplikasi *android* dan dimasukkan kedalam basis data. Kesimpulan dari penelitian pertama yaitu metode yang digunakan menunjukkan hasil data yang disimpan menjadi lebih akurat dan tepat waktu [16].

Penelitian kedua yang ditulis oleh Grasiana Oliva Londa, Ferdinandus Lidang Witi, Benediktus Yoseph Bhae pada tahun 2022 memiliki judul “Sistem Informasi Pendataan Penduduk Desa Detusoko Barat Kecamatan Detusoko Kabupaten Ende Berbasis *Web*”. Penelitian ini memiliki tujuan yaitu membantu mengatasi masalah-masalah pendataan penduduk di Desa Detusoko Barat. Metode yang digunakan yaitu metode *Waterfall* diimplementasikan untuk merancang aplikasi pendataan penduduk berbasis web menggunakan Bahasa pemrograman PHP. Hasil penelitian menunjukkan bahwa semua data yang dicatat menggunakan *website* yang dikembangkan oleh penulis langsung masuk kedalam *database*. Kesimpulan dari

penelitian kedua yaitu dengan adanya sistem informasi ini dapat membantu proses pengelolaan data penduduk yang terkait pada desa Detusoko Barat dan mempercepat proses penginputan data kependudukan [17].

Penelitian ketiga yang ditulis oleh Nicodemus Rahanra pada tahun 2022 yang memiliki judul “Sistem Informasi Data Penduduk Di Kantor Lurah Kabupaten Waropen Berbasis *Web*”. Tujuan dari penelitian ini adalah membangun sebuah website yang dapat membantu pendataan penduduk di kantor lurah Kabupaten Waropen. Penelitian ini menggunakan metode *Waterfall* untuk mengembangkan sistem berorientasi objek. Hasil penelitian berupa menyajikan beberapa informasi dan data penduduk dalam bentuk digital yang sudah disimpan kedalam *database*. Kesimpulan dari penelitian ketiga yaitu pada penelitian ini mengacu pada pembuatan *website* pengelolaan data penduduk di kantor lurah Kabupaten Waropen [18].

Penelitian keempat yang ditulis oleh Juli Budi Satya, Lilik Suhery, Alfry Aristo Jansen Sinlae, Uliyatinisa pada tahun 2021 yang berjudul “Pengembangan Sistem Pelayanan Publik Melalui Sistem Administrasi Kependudukan Menggunakan Metode *Extreme Programming*”. Penelitian ini memiliki tujuan untuk melakukan pengembangan sistem informasi yang mencakup sistem pelayanan publik. *Extreme Programming* menjadi sebuah metode yang digunakan untuk diimplementasikan pada aplikasi yang akan dikembangkan. Penelitian ini memiliki sebuah hasil yaitu sistem pelayanan publik untuk *monitoring* berkas dan pengelolaan administrasi kependudukan. Kesimpulan dari penelitian keempat yaitu hasil dari pengujian *usability*. Menghasilkan aplikasi yang dikembangkan untuk sistem administrasi kependudukan menjadi baik [13].

Penelitian kelima yang ditulis oleh Ahmad Habib, Ery Sadewa Yudha W, Ardy Januanto, Anton Breva Yunanda pada tahun 2021 yang berjudul “*Development of the Information System Management Population Data Using the Extreme Programming Methodology Approach, Case Study at Badal Village Government, Kediri, East Java, Indonesia*” [10]. Penelitian ini memiliki tujuan membangun sebuah sistem informasi kependudukan berbasis *web* menggunakan metode *Extreme Programming*. Hasil dari penelitian ini adalah sebuah sistem yang

mampu memenuhi semua kebutuhan pengguna secara detil dan akurat. Dapat disimpulkan bahwa *Extreme Programming* dapat diimplementasikan untuk merancang sebuah sistem informasi kependudukan [10].

Dari informasi yang penulis sajikan di atas, terdapat beberapa penelitian terdahulu yang relevan dengan penelitian ini adalah sebagai berikut:

Tabel 2.1 Tabel Penelitian Sebelumnya

No.	Judul Penelitian	Masalah	Metode	Hasil	Perbedaan dengan penelitian yang dilakukan
1.	“Perancangan Sistem Informasi Sensus Penduduk Berbasis <i>Android</i> Pada Desa Meranti.” [16]	Bagaimana cara merubah cara pendataan yang dilakukan pada desa meranti menjadi pendataan melalui aplikasi berbasis <i>android</i> .	<i>Waterfall</i>	Pendataan penduduk yang semula masih dicatat melalui buku. Menjadi, pendataan penduduk yang sudah menggunakan aplikasi <i>android</i> dan dimasukkan kedalam basis data	Metode <i>Waterfall</i> digunakan pada penelitian [16]. Sedangkan pada penulisan tugas akhir ini menggunakan metode <i>Extreme Programming</i>

No.	Judul Penelitian	Masalah	Metode	Hasil	Perbedaan dengan penelitian yang dilakukan
2.	“Sistem Informasi Pendataan Penduduk Desa Detusoko Barat Kecamatan Detusoko Kabupaten Ende Berbasis Web.”[17]	Pendataan penduduk di desa Detusoko Barat saat ini sudah menggunakan komputerisasi yaitu menggunakan Microsoft Word dan Microsoft Excel. Akan tetapi, masih banyak kesalahan dan kemungkinan terjadinya kehilangan data dan kesalahan <i>input</i> data sehingga menyebabkan sulitnya melakukan proses pencarian data penduduk.	<i>Waterfall</i>	Hasil penelitian menunjukkan bahwa semua data yang dicatat menggunakan website yang dikembangkan oleh penulis langsung masuk kedalam <i>database</i>	Metode yang digunakan pada penelitian [17] menggunakan metode <i>Waterfall</i> . Sedangkan pada penulisan tugas akhir ini menggunakan metode <i>Extreme Programming</i> .
3.	“Sistem Informasi Data Penduduk Di Kantor Lurah Kabupaten	Masih menggunakan cara konvensional dalam memasukkan data tentang kependudukan.	<i>Waterfall</i>	Hasil penelitian berupa menyajikan beberapa informasi dan data penduduk	Metode yang digunakan pada penelitian [18] menggunakan metode <i>Waterfall</i> . Sedangkan pada

No.	Judul Penelitian	Masalah	Metode	Hasil	Perbedaan dengan penelitian yang dilakukan
	Waropen Berbasis Web” [18]			dalam bentuk digital yang sudah disimpan kedalam <i>database</i>	penulisan tugas akhir ini menggunakan metode <i>Extreme Programming</i> .
4.	“Pengembangan Sistem Pelayanan Publik Melalui Sistem Administrasi Kependudukan Menggunakan Metode <i>Extreme Programming</i> .” [13].	Pendataan penduduk menggunakan cara ditulis di kertas.	<i>Extreme Programming</i>	Sebuah sistem pelayanan publik berbasis <i>web</i> untuk <i>monitoring</i> berkas dan pengelolaan administrasi kependudukan.	Basis yang digunakan pada penelitian [13] adalah basis <i>website</i> . Sedangkan pada penulisan tugas akhir ini menggunakan basis <i>android</i> .
5.	“ <i>Development of the Information System</i> ”	Pendataan penduduk masih menggunakan cara dicatat dan ditulis dibuku.	<i>Extreme Programming</i>	Penelitian ini memiliki hasil sebuah sistem yang	Basis yang digunakan pada penelitian [10] adalah menggunakan basis

No.	Judul Penelitian	Masalah	Metode	Hasil	Perbedaan dengan penelitian yang dilakukan
	<p><i>Management Population Data Using the Extreme Programming Methodology Approach, Case Study at Badal Village Government, Kediri, East Java, Indonesia.” [10]</i></p>			<p>mampu mencakup semua kebutuhan <i>user</i>.</p>	<p><i>website</i>. Sedangkan pada penulisan tugas akhir ini menggunakan basis <i>android</i>.</p>

Berdasarkan tabel penelitian terdahulu yang dijadikan sebuah referensi sekaligus bahan pembandingan pada penelitian ini terdapat penelitian yang serupa yaitu penelitian dengan judul “Pengembangan Sistem Pelayanan Publik Melalui Sistem Administrasi Kependudukan Menggunakan Metode *Extreme Programming*.” yang ditulis oleh Juli Budi Satya, Lilik Suhery, Alfry Aristo Jansen Sinlae, Uliyatunisa pada tahun 2021 [13]. Perbedaan dengan penelitian ini yaitu implementasi pada desa dan basis yang digunakan. Sehingga peneliti tertarik untuk melakukan penelitian tentang pendataan penduduk menggunakan metode *Extreme Programming* yang berbasis *android*.

## **2.2. Dasar Teori**

### **2.2.1. Pendataan Penduduk**

Pendataan penduduk adalah sebuah kegiatan untuk mendapatkan informasi penduduk. Pendataan penduduk biasanya dilakukan oleh petugas pendataan yang bertugas pada daerah setempat. Hasil dari kegiatan tersebut adalah sebuah data penduduk. Tidak hanya itu pendataan penduduk mencakup semua proses seperti proses pengumpulan data, proses pengolahan data, proses penyimpanan data. Data penduduk yang sudah dikumpulkan akan digunakan untuk mencapai kebutuhan dari Badan Pusat Statistik [19].

### **2.2.2. Aplikasi**

Aplikasi adalah sebuah perangkat lunak yang dikembangkan guna untuk memenuhi kebutuhan pengguna. Kebutuhan pengguna yang dapat diatasi oleh aplikasi yaitu kebutuhan untuk proses memasukkan data, proses pengolahan data dan proses penyimpanan data yang dilakukan oleh manusia menggunakan aplikasi. Aplikasi dapat diinstal dan dijalankan di berbagai perangkat antara lain komputer, *smartphone* atau perangkat lainnya [20].

### **2.2.3. Android**

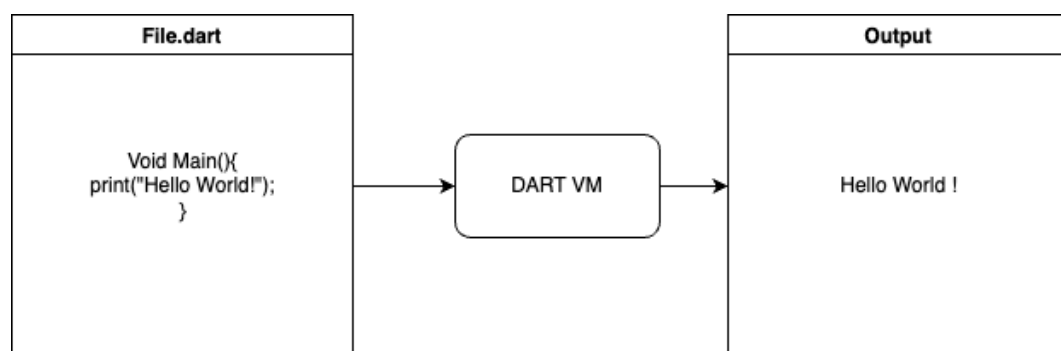
Sebuah *operating system* berbasis *linux* pada *smartphone*. *Android* sendiri memiliki tiga elemen yaitu *operating system*, *middleware* dan *main application*.



Seperti halnya *linux android* bersifat *open source*. Tujuan pertama kali *android* dikembangkan adalah untuk perangkat kamera dan alat untuk komunikasi. Namun, sebelum munculnya era *smartphone* pasar *android* tidaklah terlalu besar, lalu pihak pengembang mengimplementasikan sistem operasi *android* untuk *smartphone* [21].

#### 2.2.4. Dart

*Dart* merupakan sebuah Bahasa Pemrograman yang dikembangkan oleh perusahaan Google dan Lars Bak, Kasper Lund. *Dart* disebar oleh Google pada tahun 2011 [8]. Bahasa pemrograman *dart* terkenal mudah digunakan maupun untuk dikembangkan oleh siapapun. Bahasa pemrograman *Dart* mempunyai fitur yang sangat lengkap untuk pengembangan di beberapa *platform* seperti halnya *website*, *mobile*, *server* dan *desktop*. Khususnya pengembangan pada platform *mobile*, *dart* memiliki keunggulan yaitu dapat menjalankan dua sistem operasi pada *mobile Android* dan *iOS*. *Dart* adalah Bahasa pemrograman yang bersifat dinamis yang artinya *dart* bersifat fleksible dalam segi pengembangan [22]. Aplikasi yang dikembangkan menggunakan Bahasa pemrograman *Dart* dieksekusi langsung *direct* ke *DartVM*. Hal ini menyebabkan proses *compile* pada *dart* terbilang sangat cepat [23].



Gambar 2.1 *Process Compile Dart*

### 2.2.5. Framework Flutter

*Framework* atau biasa disebut dengan lembar kerja yang berisikan kumpulan koleksi potongan program yang dapat digunakan oleh para pengembang. Keuntungan menggunakan *framework* ada beberapa yaitu:

1. Menghemat Waktu Pengembangan

Dengan menggunakan *library* yang telah disediakan pada *framework* para pengembang tidak perlu untuk menyusun ulang karena sudah tersedia.

2. *Refuse of Code*

Kode yang dikembangkan oleh para pengembang yang menggunakan *framework* lebih terstruktur dan rapih karena *framework* sudah menyediakan lembar kerja yang dibutuhkan para pengembang.

3. *Community Help*

Para pengembang yang menggunakan *framework* yang sama pasti akan mengalami suatu permasalahan yang sama, karena itu para pengembang yang sudah mengalami permasalahan tersebut akan memberikan solusi pada dokumentasi yang telah disediakan [24].

*Flutter* adalah sebuah *Framework* yang dikembangkan oleh sebuah perusahaan Google. *Flutter* dikembangkan untuk menjadi sebuah *framework* yang dapat melakukan *crossplatform* hanya dengan *flutter*. Kelebihan menggunakan *framework flutter* adalah tidak perlu mempelajari Bahasa pemrograman *native* pada *android* maupun *iOS*. Bahasa pemrograman *native* pada *android* contohnya adalah *Kotlin* dan *Java*. Pada *iOS* Bahasa pemrograman *native* ada dua yaitu *Swift* dan *Objective-C*. [23].

### 2.2.6. Database

*Database* mempunyai sebuah kolom dan garis untuk menyimpan berbagai data yang akan siap untuk diolah. Basis data mempunyai beberapa jenis seperti contohnya *MySQL*. *MySQL* adalah sebuah basis data yang mengutamakan dalam relasi-relasi yang ada didalam tabel, garis dan kolom. Tujuan menggunakan

*MySQL* karena *MySQL* menggunakan sebuah prinsip relasi yang ada hubungan maupun garis dan kolom yang tersedia pada tabel *database* [25].

### 2.2.7. ERD

*Entity Relationship Diagram* (ERD) adalah sebuah visualisasi grafis yang menggambarkan bagaimana entitas (objek) berinteraksi di dalam basis data dan bagaimana data disimpan. ERD menggunakan tiga elemen utama:


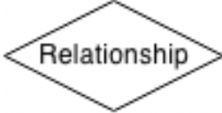
1. Entitas: Representasi dari objek dunia nyata atau aplikasi yang memiliki atribut untuk menyimpan data. Entitas ini biasanya direpresentasikan sebagai kotak dalam diagram ERD.

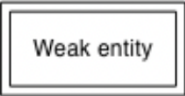


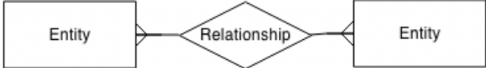
2. Atribut: Informasi yang terkait dengan entitas tersebut. Atribut ditampilkan sebagai elips di dalam kotak entitas dan mewakili data yang bisa disimpan untuk entitas tersebut.

3. Hubungan: Gambaran tentang bagaimana sebuah entitas saling terhubung dan berinteraksi satu sama lain. Hubungan ini diilustrasikan oleh garis yang menghubungkan dua entitas, seringkali disertai dengan keterangan yang menjelaskan sifat hubungan tersebut.

ERD adalah sebuah teknik yang sangat berguna dalam fase perancangan dan pengembangan basis data karena membantu menggambarkan struktur data secara jelas dan mudah dipahami [26]. Berikut merupakan daftar simbol yang ada didalam ERD:

Tabel 2.2 Simbol ERD

Simbol	Keterangan
	<i>Entity</i> : objek atau benda nyata yang memiliki atribut yang dapat disimpan dalam basis data
	<i>Relationship</i> : hubungan atau kaitan antara dua atau lebih entitas dalam basis data

	<p><i>Weak entity</i>: Entitas lemah adalah jenis entitas yang tidak memiliki identitas unik sendiri.</p>
	<p><i>One to one</i>: jenis relasi antara dua entitas di mana satu entitas pada sisi satu hanya berhubungan dengan satu entitas pada sisi lain, dan sebaliknya</p>
	<p><i>One to many</i>: jenis relasi antara dua entitas di mana satu entitas pada sisi satu berhubungan dengan banyak entitas pada sisi lain, tetapi entitas pada sisi lain hanya berhubungan dengan satu entitas pada sisi satu</p>
	<p><i>Many to many</i>: jenis relasi antara dua entitas di mana banyak entitas pada sisi satu berhubungan dengan banyak entitas pada sisi lain</p>

### 2.2.8. Unified Modeling Language (UML)


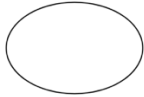
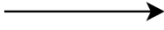
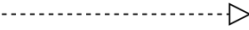
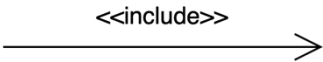
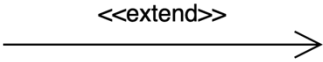
UML adalah suatu metode dalam pemodelan secara *visual* yang digunakan sebagai sarana perancangan sistem berorientasi objek [27]. UML sendiri juga memberikan standar penulisan sebuah sistem yang meliputi penulisan kelas-kelas dalam bahasa program yang spesifik [28]. Berikut merupakan simbol-simbol yang ada didalam UML:

#### 1. Usecase Diagram

*Usecase Diagram* adalah salah satu jenis diagram yang digunakan dalam rekayasa perangkat untuk menggambarkan interaksi antara aktor atau pengguna

dengan sistem dalam situasi tertentu. Tujuan utama dari *Usecase Diagram* adalah untuk menggambarkan fungsionalitas sistem dari perspektif pengguna dan mengilustrasikan bagaimana pengguna akan berinteraksi dengan sistem untuk mencapai tujuan tertentu [29]. Berikut merupakan simbol-simbol dari Usecase Diagram:



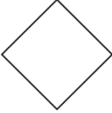


Tabel 2.3 Simbol *Usecase Diagram*

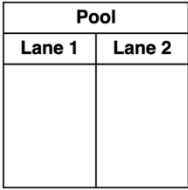
Simbol	Keterangan
 Actor	Aktor : Mewakili peran orang, sistem atau alat berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

## 2. *Activity Diagram*

*Activity Diagram* adalah salah satu bentuk diagram yang digunakan dalam rekayasa perangkat lunak untuk mengilustrasikan alur aktivitas atau aliran kerja dari suatu proses atau sistem. Tujuan utama *Activity diagram* adalah untuk menggambarkan serangkaian tindakan atau aktivitas yang terjadi dalam suatu proses, serta bagaimana aktivitas-aktivitas tersebut terkait dan saling berhubungan [30]. Berikut merupakan simbol-simbol *Activity Diagram*:

Tabel 2.4 Simbol *Activity Diagram*

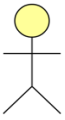
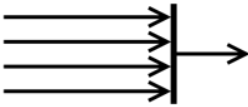

Simbol	Nama	Keterangan
	Start / status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Percabangan / <i>decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
	<i>Join</i>	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu
	End / Akhir dari proses	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir


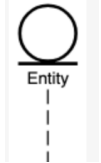
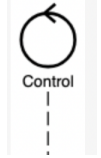


	<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
---	-----------------	---

### 3. *Sequence Diagram*

*Sequence Diagram* adalah sebuah bentuk diagram yang digunakan dalam rekayasa perangkat lunak untuk mengilustrasikan interaksi antara objek-objek dalam suatu skenario atau proses tertentu. Fungsi utama dari diagram ini adalah untuk memvisualisasikan urutan pesan (*message*) yang dikirim antara objek-objek dalam urutan kronologis, sehingga menggambarkan bagaimana objek-objek berinteraksi dan berkomunikasi satu sama lain selama eksekusi proses berlangsung [31]. Berikut merupakan simbol yang terdapat pada *Sequence Diagram*:

Tabel 2.5 Simbol *Sequence Diagram*

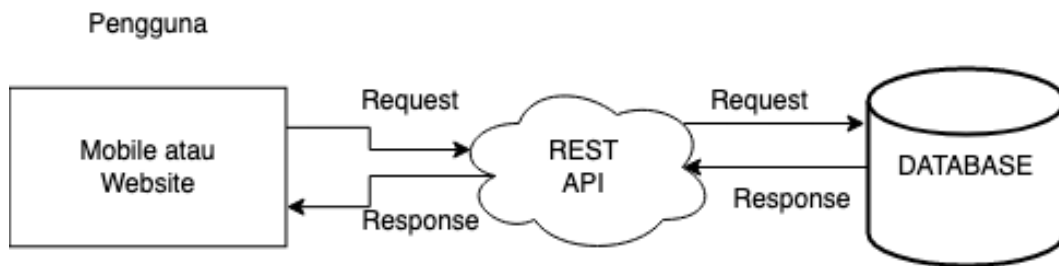
Simbol	Keterangan
	Aktor : orang, proses atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari sistem tersebut.
	<i>Join</i> : menunjukkan bagaimana pemrosesan paralel yang terjadi sebelumnya Bersatu Kembali dan menjadi satu alur tunggal untuk melakukan proses secara bersama.
	<i>Fork</i> : menggambarkan paralelisme atau pemrosesan secara bersamaan dalam suatu skenario.

 <p>Boundary</p>	<p><i>Boundary</i> : mengacu pada elemen yang menunjukkan interaksi antara sistem yang sedang dibangun (<i>subject</i>) dengan aktor-aktor eksternal atau sistem lain di lingkungan eksternal</p>
 <p>Entity</p>	<p><i>Entity</i> : Entitas mewakili bagian dari sistem yang berinteraksi satu sama lain dengan mengirim dan menerima pesan.</p>
 <p>Control</p>	<p><i>Control</i> : Control adalah entitas yang berfungsi untuk mengatur aliran interaksi antara entitas lain dalam sistem.</p>
 <p>Participant</p>	<p><i>Participant</i> : Partisipan adalah entitas yang terlibat dalam interaksi dalam diagram urutan.</p>
 <p>Database</p>	<p><i>Database</i> : Database mengacu pada entitas atau komponen dalam sistem yang bertugas menyimpan dan mengelola data.</p>

### 2.2.9. Application Programming Interface (API)

*API* merupakan sebuah perangkat lunak yang memiliki fungsi sebagai jembatan antara tampilan dengan data yang ada didalam basis data. *API* berfungsi untuk menerima permintaan dari pengguna dan memberikan tanggapan atau *response* dari data. Alasan *API* dikembangkan dan digunakan untuk perancangan aplikasi adalah karena *API* dapat memberikan jembatan atau akses dalam mengintegrasikan data baik *website* ataupun *mobile*.



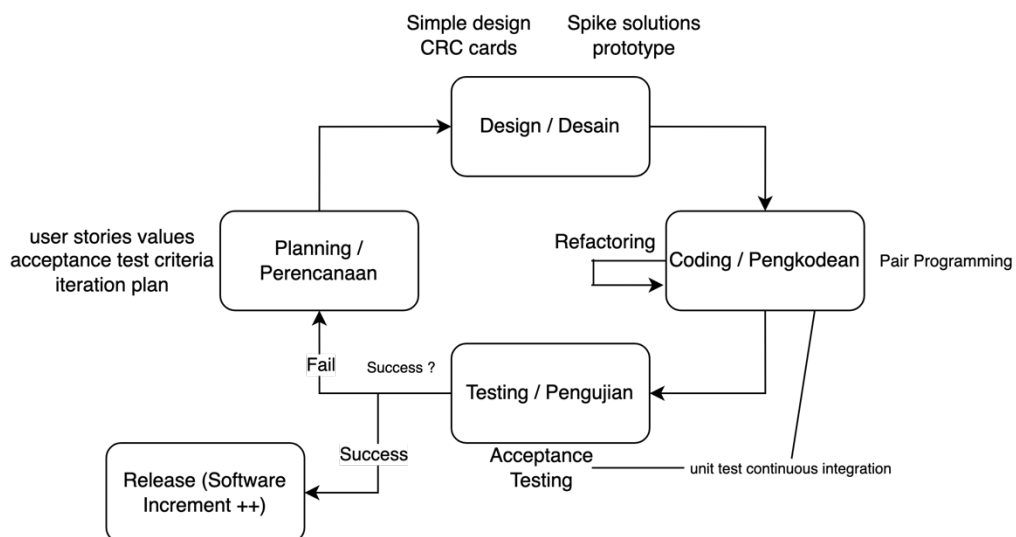


Gambar 2.2 Cara Kerja API

API memiliki beberapa *method* atau metode dalam pemanggilan fungsinya. Seperti *method GET* untuk memberikan *request* pada *REST API* lalu diteruskan pada *database*. Lalu pada *method POST* yaitu berfungsi sebagai *request insert data* ataupun *create data*. Lalu pada *method PUT* biasa digunakan untuk penggantian data yang ada didalam *database*. Terakhir yaitu *method DELETE* memiliki fungsi untuk menghapus sebuah data didalam *database*[32].

#### 2.2.10. Extreme Programming

Salah satu dari cabang metode *agile* adalah *Extreme Programming*. Dasar dari keseluruhan pengembangan metode *extreme programming* adalah pengujian dan penulisan kode yang diperoleh dari tahap kedua dalam metode ini.



Gambar 2.3 Tahapan Metode *Extreme Programming*

Gambar diatas merupakan sebuah tahapan dari metode yang digunakan dalam penelitian ini berikut penjelasan mengenai tahapan diatas.

1. *Planning* atau perencanaan dalam tahap pertama adalah tahap untuk memulai dalam perancangan. Perencanaan diambil dari sebuah keluhan dan kebutuhan pengguna yang membutuhkan sebuah sistem informasi untuk memenuhi kebutuhan pengguna.
2. *Design* atau desain sistem dalam tahap kedua adalah tahap untuk mendesain sistem yang dibutuhkan oleh kebutuhan pengguna. Seperti contohnya pengguna membutuhkan sebuah sistem informasi yang dapat mengisikan data. Jadi sistem yang didesain ada sistem untuk memasukkan data.
3. *Coding* atau tahap pengkodean disini adalah tahap ketiga setelah kebutuhan pengguna didesain dengan terstruktur akan melakukan tahap ini. Tahap ini adalah implementasi kode yang digunakan untuk memenuhi kebutuhan pengguna dalam pengembangan aplikasi.
4. *Testing* atau tahap keempat dalam metode ini adalah berarti jika tahap *coding* sudah dilakukan. Tahapan selanjutnya adalah tahap *testing* bertugas untuk melakukan uji coba apakah aplikasi yang dikembangkan sudah memenuhi syarat dan kebutuhan pengguna.
5. *Release* atau tahap terakhir jika semua tahap sudah dilalui dan memenuhi kebutuhan pengguna maka masuk ke tahap terakhir yaitu aplikasi yang dikembangkan siap untuk didistribusikan kepada pengguna [11].

#### 2.2.11. *Black Box Testing*

Ada beberapa metode yang digunakan dalam pengujian sistem salah satunya adalah metode *Black Box Testing*. *Black Box Testing* merupakan metode pengujian yang memiliki fokus dalam mengujikan apakah fungsi-fungsi yang ada bekerja dengan baik atau tidak. Pengujian ini berfokus untuk mengamati *input* dan *output* terhadap fungsi didalam sistem informasi. Metode pengujian ini tidak menghiraukan kode-kode didalam sistem informasi. Setelah melakukan pengujian ini mendapatkan hasil antara lain adalah tabel yang berisikan deskripsi mengenai prosedur pengujian fungsi. Tabel yang dihasilkan dari metode pengujian ini

meliputi deskripsi, hasil pengujian, hasil yang diharapkan dan kesimpulan dari hasil pengujian fungsi ini [33]

#### 2.2.12. Teknik Analisis Deskriptif

Teknik Analisis Deskriptif memiliki fungsi untuk menghitung nilai persentase keberhasilan dari responden pengujian [34]. Nilai persentase keberhasilan didapatkan dari perhitungan sebagai berikut sesuai dengan rumus 2.1 beserta rincian perhitungan dari skor yang diobservasi dibagi skor yang diharapkan [15]

$$\textit{Persentase Keberhasilan} = \frac{\textit{Skor yang diobservasi}}{\textit{Skor yang diharapkan}} \times 100\% \quad (2.1)$$