

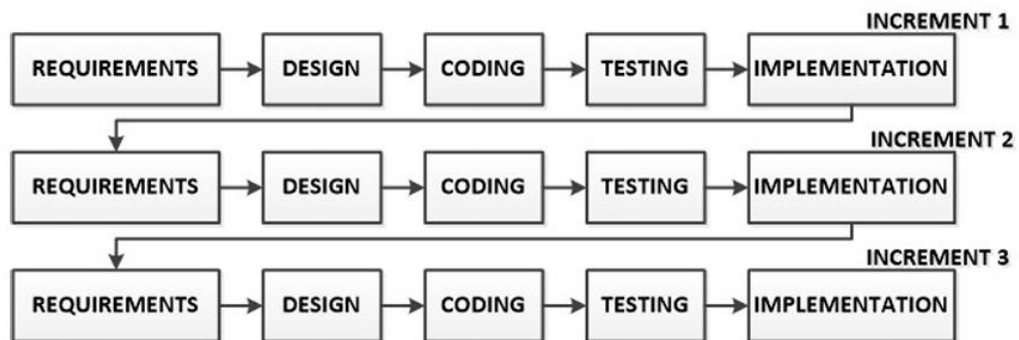
BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian ini menerapkan metode *Research and Development* (R&D) untuk merancang dan membangun sistem verifikasi identitas (e-KYC) berbasis arsitektur *microservices*. Pemilihan metode R&D ini memungkinkan peneliti untuk secara sistematis mengembangkan, menguji, dan menyempurnakan setiap fungsionalitas sistem. Dalam konteks ini, peneliti berfokus pada isolasi pengujian dan integrasi layanan e-KYC yakni *Optical Character Recognition* (OCR), *Face Match*, dan *Liveness Detection* dengan aplikasi SIPADU, guna meminimalisasi risiko kegagalan sistem (*breaking changes*) pada aplikasi SIPADU serta memastikan keandalan sistem sebelum diimplementasikan secara penuh kepada pengguna akhir.

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah Metode Inkremental (*Incremental Model*). Metode ini mengintegrasikan struktur sekuensial dengan pendekatan pengembangan yang memecah sistem ke dalam beberapa siklus rilis fungsional (inkremen). Penggunaan metode ini didasarkan pada kompleksitas arsitektur *microservices* yang diterapkan pada sistem e-KYC; di mana portal antarmuka operasional dipisahkan secara logis dari layanan e-KYC seperti *OCR*, *Face Match* dan *Liveness Detection*. Melalui pendekatan inkremental, setiap layanan (servis) dapat dirancang, dikodekan, dan diuji secara independen. Isolasi siklus pengembangan ini secara efektif meminimalisasi potensi gangguan sistem berskala besar (*breaking changes*) akibat ketidakcocokan antarmuka pemrograman aplikasi (API), serta memastikan stabilitas integrasi antar-servis sebelum sistem diimplementasikan sepenuhnya kepada pengguna akhir. Adapun gambaran metode Inkremental yang digunakan oleh peneliti adalah seperti berikut:



Gambar 3.1 Metode Pengembangan

Sumber: Rachman dkk. (2020)

1. Pelaksanaan *Increment 1: Microservices* Ekstraksi Identitas (Modul OCR KTP)

Iterasi pertama berfokus pada pembuatan layanan untuk membaca teks dari dokumen identitas (KTP) secara otomatis yang berdiri sendiri (*standalone*).

a) *Requirement Analysis*

Tahap ini berfokus pada penentuan format data masukan berupa gambar KTP dan target keluaran berupa teks (NIK, Nama, Tempat/Tanggal Lahir, Alamat, dan lain-lain). Selain itu, tahap ini juga mengidentifikasi cara memisahkan proses OCR ini agar beban komputasinya yang berat tidak mengganggu kinerja aplikasi SIPADU.

b) *Design*

Berdasarkan analisis tersebut, tahap perancangan mendefinisikan alur pemrosesan gambar sebelum masuk ke sistem OCR (*image preprocessing*). Selain itu, dirancang pula titik akhir (*endpoint*) API khusus yang berdiri sendiri agar sistem siap menerima unggahan gambar KTP dari pengguna.

c) *Coding*

Layanan ini dikembangkan menggunakan bahasa pemrograman Python. Kode program ditulis untuk memproses gambar menggunakan pustaka OpenCV (seperti *grayscale* dan penyesuaian kontras), lalu

teksnya dibaca menggunakan Tesseract OCR. Hasil bacaan tersebut kemudian dibersihkan dan diformat menggunakan pola *Regular Expression* (Regex) agar menghasilkan format data JSON yang rapi.

d) *Testing*

Pengujian dilakukan secara mandiri (*unit testing*) untuk mengukur tingkat akurasi pembacaan teks OCR. Pengujian menggunakan metrik *Character Error Rate* (CER) dengan memasukkan berbagai sampel gambar KTP.

e) *Implementation*

Modul OCR yang telah selesai dibuat kemudian dibungkus ke dalam sebuah kontainer menggunakan Docker (*Dockerfile*). Hal ini bertujuan agar layanan beserta seluruh pustaka bawaannya dapat berjalan dengan stabil di server tanpa ada masalah bentrok versi (*compatibility*).

2. Pelaksanaan *Increment 2: Microservices Verifikasi Biometrik (Modul Face Match & Liveness)*

Iterasi kedua bertujuan menambahkan lapisan keamanan pengenalan wajah guna mencegah manipulasi identitas, yang juga dibangun secara terpisah dari aplikasi utama.

a) *Requirement Analysis*

Tahap ini mengidentifikasi kebutuhan data berupa gambar wajah dari KTP (hasil potongan dari inkremen sebelumnya) dan gambar wajah asli dari kamera (*selfie*). Tahap ini juga menganalisis kebutuhan kapasitas memori (RAM) pada server, mengingat proses *deep learning* untuk pengenalan wajah membutuhkan sumber daya yang besar.

b) *Design*

Perancangan difokuskan pada pembuatan *Sequence Diagram* untuk menggambarkan alur sistem saat mendeteksi objek hidup (*liveness*), seperti memberikan instruksi berkedip kepada pengguna. Selain itu, dirancang juga alur penyimpanan data ciri wajah (*vektor embedding*) agar aman dan terpisah dari basis data utama.

c) *Coding*

Pembuatan kode program menggunakan pustaka DeepFace pada Python untuk mengekstrak fitur wajah dan menghitung persentase kemiripan wajah di KTP dengan wajah asli pengguna. Selain itu, algoritma deteksi *liveness* juga dibuat untuk memastikan bahwa objek yang ada di depan kamera adalah manusia asli, bukan sekadar foto atau layar gawai yang dihadapkan ke kamera.

d) *Testing*

Pengujian tingkat akurasi dilakukan secara kuantitatif menggunakan *Confusion Matrix*. Sistem diuji dengan cara memberikan wajah palsu (*spoofing*) untuk melihat rasio kesalahan penerimaan (*False Positive*), serta diuji dengan wajah asli untuk melihat rasio kesalahan penolakan (*False Negative*). Tujuannya adalah mencari nilai batas (*threshold*) yang paling seimbang (*Equal Error Rate*).

e) *Implementation*

Modul pengenalan wajah ini kemudian dikontainerisasi menggunakan Docker. Pada tahap ini, dilakukan konfigurasi pembatasan penggunaan memori pada Docker (*resource limits*) untuk mencegah server mati mendadak (*Out-of-Memory*) saat memproses wajah secara intensif.

3. Pelaksanaan *Increment 3*: Orkestrasi Antar-Layanan, Integrasi Arsitektur, dan Deployment

Iterasi terakhir ini berfungsi untuk menggabungkan aplikasi utama (SIPADU) dengan layanan e-KYC menjadi satu sistem yang siap digunakan.

a) *Requirement Analysis*

Fokus pada tahap ini adalah menentukan format pertukaran data antar-layanan. Karena proses e-KYC membutuhkan waktu komputasi yang tidak sebentar, tahap ini juga menganalisis kebutuhan sistem antrean asinkron agar aplikasi klien tidak terhenti atau *hang* saat menunggu hasil.

b) *Design*

Desain jaringan diarahkan pada perancangan *API Gateway* sebagai pintu masuk sekaligus penyaring keamanan. Selain itu, dirancang alur pertukaran data antrean (*Message Queuing*). File gambar KTP dan swafoto akan disimpan di layanan penyimpanan eksternal, sedangkan referensi tautan (URL) file tersebut akan dimasukkan ke dalam antrean Redis untuk diproses oleh layanan e-KYC.

c) *Coding*

Pemrograman difokuskan pada pengaturan rute *API Gateway* menggunakan sistem otentikasi JWT. Di sisi aplikasi utama, dibuat antarmuka (UI) untuk verifikasi wajah dan kode *worker* untuk mengeksekusi data dari antrean Redis. Hasil verifikasi dari layanan e-KYC akan dikirimkan kembali ke aplikasi klien secara *real-time* menggunakan WebSockets.

d) *Testing*

Sistem yang sudah terintegrasi ini diuji kinerjanya menggunakan *Stress Testing* untuk melihat apakah antrean Redis mampu menahan banyak permintaan verifikasi secara bersamaan. Pengujian ketahanan (*fault tolerance*) juga dilakukan dengan mematikan layanan e-KYC sementara untuk memastikan aplikasi utama tidak ikut *crash*.

e) *Implementation*

Seluruh layanan yang telah dikontainerisasi diunggah (*deploy*) ke server *cloud*. Setelah aplikasi berjalan di lingkungan produksi, dilakukan pemantauan (*monitoring*) pada penggunaan CPU dan RAM secara langsung untuk memastikan bahwa isolasi beban kerja pada arsitektur *microservices* ini benar-benar berjalan dengan efektif.

B. Setting Penelitian

1. Lokasi Penelitian

Penelitian ini dilakukan di BAZNAS Kota Padang, yaitu sebuah badan zakat resmi nasional terletak di Jl. By Pass No.KM 12, Sungai Sapih, Kec. Kuranji, Kota Padang, Sumatera Barat.

2. Jadwal Penelitian

Tabel 3.1 Waktu Penelitian

Kegiatan/Waktu	Bulan 1				Bulan 2				Bulan 3				Bulan 4			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Studi Literatur																
Increment 1																
1. Requirement Analysis																
2. Design																
3. Coding																
4. Testing																
5. Implementation																
Increment 2																
1. Requirement Analysis																
2. Design																
3. Coding																
4. Testing																
5. Implementation																
Increment 3																
1. Requirement Analysis																
2. Design																
3. Coding																
4. Testing																
5. Implementation																

Adapun kegiatan yang dilakukan dalam penelitian ini menggunakan pendekatan pengembangan perangkat lunak secara inkremental. Pengembangan dibagi menjadi tiga *increment* (inkremen) utama yang dilaksanakan secara berkesinambungan selama empat bulan. Setiap inkremen memiliki siklus pengembangannya masing-masing untuk memastikan sistem dibangun dan dievaluasi secara bertahap.

- a) Studi Literatur dilaksanakan pada minggu pertama bulan pertama. Tahap ini difokuskan pada pengkajian teori, dokumentasi teknis, dan pencarian literatur ilmiah terkait perancangan arsitektur *microservices*, algoritma pengolahan gambar untuk ekstraksi teks (OCR), pengenalan wajah (*Face Match*) dan validasi objek hidup (*Liveness Detection*), serta strategi integrasi API ke dalam aplikasi *front-end* (SIPADU) yang sudah ada..
- b) Pelaksanaan *Increment Pertama*
Iterasi pertama berfokus pada pengembangan *Microservices* Ekstraksi Identitas (Modul OCR KTP).
 - a. *Requirement Analysis*: Dilaksanakan pada minggu kedua bulan pertama, difokuskan pada identifikasi *format* masukan gambar KTP dan parameter keluaran teks.
 - b. *Design*: Dilaksanakan pada minggu ketiga bulan pertama.
 - c. *Coding*: Dilaksanakan pada minggu keempat bulan pertama, mencakup penulisan skrip pemrosesan gambar digital dan *endpoint* layanan.
 - d. *Testing*: Dilaksanakan pada minggu pertama bulan kedua, menguji tingkat akurasi pembacaan karakter pada berbagai kondisi pencahayaan gambar KTP.
 - e. *Implementation*: Dilaksanakan pada minggu kedua bulan kedua.
- c) Pelaksanaan *Increment Kedua*
Iterasi kedua berfokus pada pengembangan *Microservices* Verifikasi Biometrik (Modul *Face Match* dan *Liveness Detection*).

- a. *Requirement Analysis*: Dilaksanakan pada minggu ketiga bulan kedua.
 - b. *Design*: Dilaksanakan pada minggu keempat bulan kedua.
 - c. *Coding*: Dilaksanakan pada minggu pertama bulan ketiga, berfokus pada algoritma pendeteksian pola wajah dan validasi objek hidup.
 - d. *Testing*: Dilaksanakan pada minggu kedua bulan ketiga, untuk memvalidasi tingkat kecocokan antara wajah pada KTP dengan wajah tangkapan kamera.
 - e. *Implementation*: Dilaksanakan pada minggu ketiga bulan ketiga.
- d) Pelaksanaan *Increment Ketiga*
- Iterasi ketiga berfokus pada koordinasi antar-layanan, Integrasi Arsitektur, dan *Deployment*.
- a. *Requirement Analysis*: Dilaksanakan pada minggu keempat bulan ketiga. Tahap ini berfokus pada penentuan mekanisme orkestrasi dan alur komunikasi ekosistem, termasuk perancangan integrasi API *Gateway* sebagai titik masuk komunikasi dari klien, implementasi *Message Broker* (Redis Queue) untuk mengelola antrian tugas asinkron antar-*microservices*, serta perancangan mekanisme notifikasi *real-time* berbasis *WebSockets* (melalui Supabase) untuk memberikan pembaruan status verifikasi kepada pengguna akhir.
 - b. *Design*: Dilaksanakan pada minggu pertama bulan keempat.
 - c. *Coding*: Dilaksanakan pada minggu kedua bulan keempat, mencakup konfigurasi jembatan komunikasi (*routing*) yang menghubungkan aplikasi klien dengan ekosistem *microservices* E-KYC, penulisan *worker script* untuk mengeksekusi antrian Redis, dan implementasi pendengar kejadian (*event listener*) pada sisi klien.
 - d. *Testing*: Dilaksanakan pada minggu ketiga bulan keempat, difokuskan pada pengujian beban (*load testing*), latensi respons API, dan penanganan kegagalan sistem (*fault tolerance*) saat terjadi antrian pemrosesan data biometrik yang panjang

e. Implementation: Dilaksanakan pada minggu keempat bulan keempat, merilis layanan ke lingkungan server.

e) *Dokumentasi* (Berkelanjutan)

Dokumentasi dilaksanakan secara terus-menerus mulai dari minggu pertama bulan pertama hingga minggu keempat bulan keempat. Tahapan ini mencakup pencatatan progres harian/mingguan dari setiap inkremen, penyusunan draf bab-bab skripsi, serta pelaporan hasil akhir dari seluruh rangkaian kegiatan penelitian.

C. Alat/Instrumen Penelitian

Adapun alat atau instrumen yang digunakan dalam penelitian ini antara lain sebagai berikut:

1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini antara lain:

- a. Processor Intel Core i3-1005G1
- b. RAM 8 GB
- c. SSD 512 GB
- d. GPU Intel UHD Graphics

2. Perangkat Lunak

Perangkat Lunak yang digunakan dalam penelitian ini antara lain:

- a. Sistem operasi Windows 11 64-bit
- b. Visual Studio Code
- c. React
- d. Express JS
- e. Docker
- f. Supabase
- g. Message Broker: Redis

D. Teknik Pengumpulan Data

Pada penelitian ini, penulis menggunakan beberapa metode pengumpulan data sebagai berikut:

1. Observasi (Pengamatan)

Observasi dilakukan secara langsung terhadap struktur aplikasi SIPADU yang saat ini masih dalam tahap pengembangan (prototipe) di BAZNAS Kota Padang. Pengamatan difokuskan pada analisis rancangan arsitektur awal, keterbatasan kapabilitas integrasi pada prototipe saat ini, serta kesiapan infrastruktur untuk mengadopsi model *microservices*. Data hasil observasi ini digunakan sebagai dasar (analisis kesenjangan) untuk merancang arsitektur baru yang dapat mengakomodasi layanan E-KYC secara independen dan terukur (*scalable*).

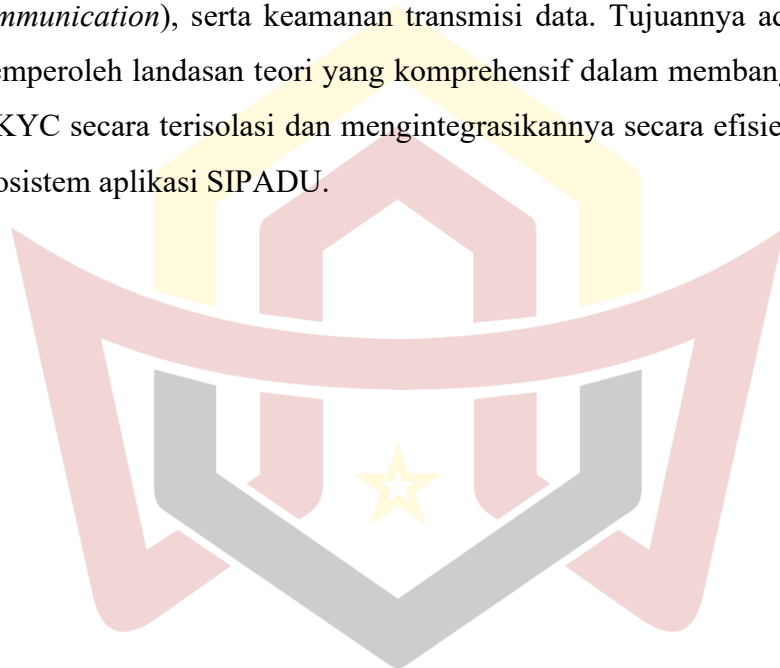
2. Wawancara (Interview)

Wawancara dilakukan dengan manajemen dan staf operasional BAZNAS Kota Padang yang menangani pendaftaran penerima bantuan, serta tim pengembang aplikasi SIPADU. Wawancara ini bertujuan untuk:

1. Memetakan alur bisnis (SOP) verifikasi data calon penerima bantuan yang saat ini sepenuhnya berjalan secara manual.
2. Melakukan analisis kebutuhan (*needs assessment*) untuk mengidentifikasi kendala atau titik jenuh (*bottleneck*) pada proses manual tersebut (misalnya potensi pemalsuan identitas, durasi layanan, atau beban kerja pegawai). Hal ini dilakukan untuk menentukan secara objektif apakah implementasi otomasi verifikasi identitas (E-KYC) benar-benar menjadi kebutuhan esensial bagi BAZNAS, atau belum diperlukan pada tahap ini.
3. Mengestimasi rata-rata volume pendaftaran penerima bantuan per bulan. Data kuantitatif ini krusial untuk memproyeksikan beban kerja sistem (*workload*), yang nantinya digunakan sebagai justifikasi teknis apakah pengembangan prototipe SIPADU memerlukan pemisahan layanan E-KYC ke dalam arsitektur *microservices* agar dapat menangani lonjakan data secara terukur (*scalable*). Pendekatan wawancara semi-terstruktur digunakan agar peneliti dapat merumuskan spesifikasi kebutuhan perangkat lunak (SRS) yang berangkat dari permasalahan nyata di lapangan.

3. Studi Pustaka

Studi pustaka dilakukan dengan mengumpulkan data dari literatur ilmiah, buku teks, dan dokumentasi teknologi terkait. Pencarian literatur difokuskan pada perancangan arsitektur *microservices*, pengembangan sistem E-KYC mandiri berbasis pengolahan gambar digital (seperti implementasi *OCR* untuk ekstraksi teks dokumen dan pengenalan wajah untuk verifikasi identitas), protokol komunikasi antar-layanan (*service-to-service communication*), serta keamanan transmisi data. Tujuannya adalah untuk memperoleh landasan teori yang komprehensif dalam membangun *service* E-KYC secara terisolasi dan mengintegrasikannya secara efisien ke dalam ekosistem aplikasi SIPADU.



UIN IMAM BONJOL
PADANG