PENGEMBANGAN *APPLICATION PROGRAMMING INTERFACE* (API) UNTUK LAYANAN PERTUKARAN DATA E-DOXID

SKRIPSI

Program Studi Sarjana Informatika
Jurusan Informatika

Oleh:

DONI WINATA

NIM D1041181039

FAKULTAS TEKNIK

UNIVERSITAS TANJUNGPURA

PONTIANAK

2025

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Doni Winata

NIM : D1041181039

menyatakan bahwa dalam skripsi yang berjudul "Pengembangan *Application Programming Interface* (API) Untuk Layanan Pertukaran Data E-Doxid" tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi manapun. Sepanjang pengetahuan Saya, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenar-benarnya. Saya sanggup menerima konsekuensi akademis dan hukum di kemudian hari apabila pernyataan yang dibuat ini tidak benar.

Pontianak, 30 Juni 2025



Doni Winata

NIM D1041181039

KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI
UNIVERSITAS TANJUNGPURA
FAKULTAS TEKNIK
Jalan Prof. Dr. H. Hadari Nawawi Pontianak 78124
Telepon (0561) 740186, WA: +6282152280907
Email : ft@untan.ac.id Website : http://teknik.untan.ac.id

HALAMAN PENGESAHAN

PENGEMBANGAN *APPLICATION PROGRAMMING INTERFACE* (API) UNTUK LAYANAN PERTUKARAN DATA E-DOXID

Program Studi Sarjana Informatika
Jurusan Informatika

Oleh:
Doni Winata
NIM D1041181039

Telah dipertahankan di depan Penguji Skripsi pada tanggal 30 Juni 2025 dan diterima sebagai salah satu persyaratan untuk memperoleh gelar sarjana

Susunan Penguji Skripsi:

Ketua,

Dr. Ir. Yus Sholva, S.T., M.T.
NIP 197410192003121002

Penguji Utama,

Morteza Muthahhari, S.Kom., M.T.I.
NIP 198607092019031008

Sekretaris,

Haried Novriando, S.Kom., M.Eng.
NIP 198611132020121005

Penguji Pendamping,

Rifqi Anugrah, S.Kom., M.Kom.
NIP 199803122024061001

Pontianak, 30 Juni 2025
Dekan,

Dr.-Ing. Ir. Slamet Widodo, M.T., IPM
NIP 196712231992031002

KATA PENGANTAR

Puji syukur saya panjatkan kepada Allah SWT, karena berkat dan rahmat-Nya saya dapat menyelesaikan skripsi ini yang berjudul **"Pengembangan *Application Programming Interface* (API) Untuk Layanan Pertukaran Data E-Doxid".** Skripsi ini dibuat untuk memenuhi syarat untuk menyelesaikan program Sarjana (S1) Jurusan Informatika Fakultas Teknik Universitas Tanjungpura.

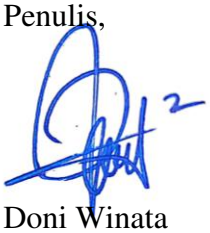
Dalam penelitian ini telah dikembangkan *backend* API untuk aplikasi E-Doxid untuk mendukung pengembangan aplikasi E-Doxid berbasis *mobile* maupun sistem lain yang memerlukan data E-Doxid.

Saya ucapkan terima kasih kepada Bapak Dr. Ir. Yus Sholva, S.T., M.T., selaku Ketua Jurusan Informatika Fakultas Teknik Universitas Tanjungpura Pontianak dan pembimbing utama yang telah mempercayakan judul skripsi ini pada penulis. Bapak Haried Novriando, S.Kom., M.Eng., selaku dosen pembimbing kedua skripsi yang telah bersedia meluangkan waktunya untuk membimbing penulis selama pengerjaan penelitian. Penulis juga mengucapkan terimakasih kepada Bapak Morteza Muthahhari, S.Kom., M.T.I, selaku penguji utama dan Bapak Rifqi Anugrah, S.Kom., M.Kom, selaku penguji kedua. Penulis juga ingin mengucapkan terimakasih pada orang tua, keluarga, teman dan sahabat, yang tidak dapat disebutkan satu persatu.

Semoga Allah yang Maha Esa membalas segala kebaikan semua pihak yang telah membantu dalam penelitian ini. Penulis menyadari terdapat kekurangan dalam tulisan ini, oleh karena itu dengan segenap kerendahan hati penulis mengharap kritik dan saran yang membangun guna menyempurnakan penelitian skripsi ini. Semoga penelitian ini dapat bermanfaat bagi semua pihak yang membutuhkan.

Pontianak, 30 Juni 2025

Penulis,



Doni Winata

HALAMAN PERSEMBAHAN

Skripsi ini saya persembahkan kepada Allah SWT yang senantiasa memberikan nikmat, rahmat dan anugerah-Nya melalui orang-orang yang membimbing dan mendukung sehingga saya dapat menulis dan menyelesaikan skripsi ini.

Saya persembahkan karya kecil ini untuk Ayahanda **Aswandi** dan Ibunda tercinta **Marina** sebagai tanda bakti, hormat dan rasa terimakasih kepada yang selalu memberikan kasih sayang, dukungan dan doa tulus yang tiada henti. Terimakasih telah menjadi orang tua yang terbaik.

Terimakasih juga untuk teman-teman Informatika Untan Reg A Angkatan 2018 yang menghadirkan berbagai warna selama saya menempuh pendidikan di perguruan tinggi ini.

ABSTRAK

Program Studi Informatika memiliki sistem berbasis *web* bernama E-Doxid (Elektronik Dokumen Sidang) yang digunakan untuk mengelola berkas-berkas administrasi seperti proposal, sidang hasil, dan sidang akhir. Namun, sistem ini belum memiliki kemampuan integrasi lintas *platform* yang diperlukan untuk mendukung pengembangan aplikasi *mobile* dan kolaborasi sistem di masa depan. Penelitian ini bertujuan untuk merancang *Application Programming Interface* (API) menggunakan gaya arsitektur *Representational State Transfer* (REST) guna memungkinkan integrasi antara aplikasi E-Doxid berbasis *web* dan rencana pengembangan aplikasi mobile. Pengembangan sistem dilakukan dengan menggunakan metode *prototype*. Hasil penelitian menunjukkan bahwa API yang dikembangkan mampu menampilkan data profil, pengajuan, pengguna, dan pengaturan sesuai dengan peran masing-masing *user*, seperti admin, bagian administrasi, dosen, dan mahasiswa, dalam format respons JSON. Pengujian dilakukan menggunakan aplikasi Postman dan sistem dokumentasi sederhana, yang menunjukkan bahwa API dapat menerima dan memproses permintaan dari sistem berbasis *web* dengan baik. Dengan demikian, API ini diharapkan dapat menjadi jembatan komunikasi data yang efektif untuk pengembangan lebih lanjut sistem E-Doxid lintas *platform*.

Kata kunci: API, E-Doxid, REST, *Prototype*, Postman

ABSTRACT

The Informatics Study Program has a web-based system called E-Doxid (Electronic Document for Seminars), which is used to manage administrative documents such as proposals, results seminars, and final seminars. However, the system currently lacks cross-platform integration capabilities needed to support mobile application development and future system collaboration. This study aims to design an Application Programming Interface (API) using the Representational State Transfer (REST) architectural style to enable integration between the web-based E-Doxid application and the planned mobile application. The system was developed using the prototype method. The results show that the developed API can display profile, submission, user, and settings data in JSON response format according to the roles of each user, including administrators, administrative staff, lecturers, and students. Testing was conducted using the Postman application and a simple documentation system, demonstrating that the API can successfully receive and process requests from the web-based system. Therefore, the API is expected to serve as an effective data communication bridge for the future cross-platform development of the E-Doxid system.

Keywords : API, E-Doxid, REST, Prototype, Postman

DAFTAR ISI

HALAMAN PERNYATAAN ii
HALAMAN PENGESAHAN iii
KATA PENGANTAR iv
HALAMAN PERSEMBAHAN v
ABSTRAK vi
ABSTRACT vii
DAFTAR ISI viii
DAFTAR TABEL x
DAFTAR GAMBAR xi
DAFTAR KODE PROGRAM xiii

BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 2
1.3 Tujuan Penelitian 2
1.4 Pembatasan Masalah 2
1.5 Sistematika Penulisan 3

BAB II TINJAUAN PUSTAKA 4
2.1 Penelitian Terkait 4
2.2 Landasan Teori 6
2.2.1 Elektronik Dokumen Sidang (E-Doxid) 6
2.2.2 Perangkat Lunak 7
2.2.3 Bahasa Pemrograman 7
2.2.3.1 *Hypertext Markup Language* (HTML) 7
2.2.3.2 *Hypertext Preprocessor* (PHP) 7
2.2.4 Laravel 8
2.2.5 Laravel Sanctum 8
2.2.6 Laravel Valet 9
2.2.7 Postman 10
2.2.8 *Application Programming Interface* (API) 14
2.2.9 *Representational State Transfer* (REST) 16
2.2.10 *Javascript Object Notation* (JSON) 18
2.2.11 *Flowchart* 19
2.2.12 *Unified Modelling Language* 21
2.2.12.1 *Use Case Diagram* 21
2.2.12.2 *Activity Diagram* 22
2.2.12.3 *Class Diagram* 23
2.2.12.3.1 Notasi *Class Diagram* 23
2.2.12.3.2 Hubungan *Class Diagram* 24
2.2.13 Metode *Prototype* 25

BAB III METODOLOGI PENELITIAN 26
3.1 Langkah Penelitian 26
3.2 Metode Pengembangan Perangkat Lunak 27
3.2.1 Analisis Kebutuhan Dan Pengumpulan Data 27
3.2.2 Desain Cepat 28

3.2.3 Membangun *Prototype* 28
3.2.4 Evaluasi 28
3.2.5 Menyempurnakan *Prototype* 29
3.3 Alat Bantu Pengembangan 30
3.3.1 Perangkat Keras (*Hardware*) 30
3.3.2 Perangkat Lunak (*Software*) 30
3.4 Perancangan Sistem 31
3.4.1 Arsitektur Sistem 31
3.4.2 *Unified Modeling Language* (UML) 32
3.4.2.1 Perancangan *Use Case Diagram* 32
3.4.2.2 Perancangan *Activity Diagram* 34
3.4.2.3 Perancangan *Class Diagram* 40
3.5 Database 40
3.6 Perancangan Pengujian 50
3.6.1 Skenario Pengujian dengan Postman 50
3.6.2 Skenario Pengujian dengan *Web* 55
3.6.3 Skenario Pengujian API Publik 56
BAB IV HASIL DAN PEMBAHASAN 57
4.1 Gambaran Umum API 57
4.2 Struktur dan Hak Akses Berdasarkan Level Pengguna 58
4.2.1 Admin 58
4.2.2 Bagian Administrasi 60
4.2.3 Dosen 61
4.2.4 Mahasiswa 62
4.3 Implementasi Otentikasi dan *Middleware* 63
4.4 Pengujian API 65
4.4.1 Postman 65
4.4.1.1 Otentikasi 65
4.4.1.2 Profil 66
4.4.1.3 Admin 67
4.4.1.4 Bagian Administrasi 72
4.4.1.5 Dosen 73
4.4.1.6 Mahasiswa 75
4.4.2 *Web* 76
4.4.2.1 Akses 76
4.4.2.2 *Authentication* 76
4.4.2.3 Profil 77
4.4.2.4 Admin 77
4.4.2.5 Bagian Administrasi 79
4.4.2.6 Dosen 80
4.4.2.7 Mahasiswa 82
4.4.2.8 *Forbidden* 84
4.4.3 API Publik 84
BAB V PENUTUP 89
5.1 Kesimpulan 89
5.2 Saran 89
DAFTAR PUSTAKA 90

DAFTAR TABEL

Tabel 2. 1 Simbol *Flowchart* 19
Tabel 2. 2 Simbol *Use Case Diagram* 21
Tabel 2. 3 Hubungan Kardinalitas 25
Tabel 3. 1 Deskripsi *Use Case* 33
Tabel 3. 2 Kamus Data *User* 40
Tabel 3. 3 Kamus Data *Submissions* 41
Tabel 3. 4 Kamus Data *Skill Group* 42
Tabel 3. 5 Kamus Data *Types* 43
Tabel 3. 6 Kamus Data *Verificators* 43
Tabel 3. 7 Kamus Data *Settings* 44
Tabel 3. 8 Kamus Data *Temporary Files* 44
Tabel 3. 9 Kamus Data *Formats* 44
Tabel 3. 10 Kamus Data *Required Files* 45
Tabel 3. 11 Kamus Data *Requires Values* 46
Tabel 3. 12 Kamus Data Mahasiswa *Type Access Permission* 46
Tabel 3. 13 Kamus Data *Submission Required Files* 46
Tabel 3. 14 Kamus Data *Submission Required Values* 47
Tabel 3. 15 Kamus Data *Get Type Access Permissions* 47
Tabel 3. 16 Kamus Data *Submission Consultations* 47
Tabel 3. 17 Kamus Data *Skill Group Lecturers* 48
Tabel 3. 18 Kamus Data *Activity User Logs* 48
Tabel 3. 19 Kamus Data *Notifications* 48
Tabel 3. 20 Kamus Data *Activity Submission Logs* 49
Tabel 3. 21 Kamus Data *Gradings* 49
Tabel 3. 22 Kamus Data *Personal Access Tokens* 49
Tabel 4. 1 Struktur Umum *Endpoint* 57
Tabel 4. 2 *Endpoint* Yang Dapat Diakses Oleh Admin 58
Tabel 4. 3 *Endpoint* Yang Dapat Diakses Bagian Administrasi (TU) 60
Tabel 4. 4 *Endpoint* Yang Dapat Diakses Oleh Dosen 61
Tabel 4. 5 *Endpoint* Yang Dapat Diakses Oleh Mahasiswa 62
Tabel 4. 6 Pengujian *Endpoint Users* yang Diakses oleh Admin 67
Tabel 4. 7 Pengujian *Endpoint* Tipe Pengajuan yang Diakses oleh Admin 68
Tabel 4. 8 Pengujian *Endpoint* Pengajuan yang Diakses oleh Admin 69
Tabel 4. 9 Pengujian *Endpoint* Kelompok Keahlian yang Diakses oleh Admin . 70
Tabel 4. 10 Pengujian *Endpoint* Pengaturan yang Diakses oleh Admin 71
Tabel 4. 11 Pengujian *Endpoint* Bagian Administrasi (TU) 72
Tabel 4. 12 Pengujian *Endpoint* Dosen 73
Tabel 4. 13 Pengujian *Endpoint* Mahasiswa 75
Tabel 4. 14 Contoh *Endpoint* API Publik 85

DAFTAR GAMBAR

Gambar 2. 1 Dashboard E-Doxid Level Mahasiswa 6
Gambar 2. 2 Contoh *Request* GET untuk Mendapatkan Serangkaian Data..... 10
Gambar 2. 3 Contoh *Request* GET *S*ederhana dengan "Echo" 11
Gambar 2. 4 Contoh *Visualize* untuk Menampilkan Data dalam Bentuk Tabel.. 12
Gambar 2. 5 Contoh *Request* Variabel Dinamis dan Responnya..... 12
Gambar 2. 6 Contoh Skrip yang Digunakan pada Ketiga *Request* saat *Run*..... 13
Gambar 2. 7 Hasil *Run* dari Memasukan Skrip pada Post-res..... 13
Gambar 2. 8 *Artisan: Make Controller* 15
Gambar 2. 9 Membuat ContohAPIController 15
Gambar 2. 10 API *Controller*..... 15
Gambar 2. 11 Folder API 15
Gambar 2. 12 Contoh Respons dengan Format JSON 18
Gambar 2. 13 Notasi *Activity Diagram* 22
Gambar 2. 14 Notasi *Class Diagram* 23
Gambar 2. 15 Simbol *Assosiation* 24
Gambar 2. 16 Hubungan Asosiasi 24
Gambar 2. 17 Simbol *Aggregation*..... 24
Gambar 2. 18 Hubungan Agregasi 24
Gambar 2. 19 Simbol *Generalization*..... 25
Gambar 2. 20 Hubungan Generalisasi..... 25
Gambar 3. 1 *Flowchart* Metodologi Penelitian.....26
Gambar 3. 2 Tahapan Pada *Prototype* (Pressman, 2010)..... 27
Gambar 3. 3 Arsitektur Sistem 31
Gambar 3. 4 *Use Case Diagram*..... 32
Gambar 3. 5 *Activity Diagram* – Profil 34
Gambar 3. 6 Admin - Kelola *User* 34
Gambar 3. 7 Admin - Kelola Kelompok Keahlian..... 35
Gambar 3. 8 Admin - Kelola Pengaturan 35
Gambar 3. 9 Admin - Kelola Pengajuan (1)..... 36
Gambar 3. 10 Admin - Kelola Pengajuan (2)..... 37
Gambar 3. 11 Bagian Administrasi – Kelola Pengajuan..... 38
Gambar 3. 12 Dosen – Kelola Pengajuan..... 39
Gambar 3. 13 Mahasiswa – Kelola Pengajuan..... 39
Gambar 3. 14 *Class Diagram* Hubungan *User* Dengan Token..... 40
Gambar 3. 15 Membuat *Collection* Baru..... 50
Gambar 3. 16 Menambahkan *Request* Baru 50
Gambar 3. 17 Contoh URL API..... 51
Gambar 3. 18 Mengatur Tipe *Authentication*..... 51
Gambar 3. 19 Menambahkan *Headers*..... 51
Gambar 3. 20 Mengisi Parameter..... 52
Gambar 3. 21 Mengisi *Body* Berupa form-data..... 52
Gambar 3. 22 Contoh Respons API dengan Format JSON..... 53
Gambar 3. 23 *Run Collection* 53
Gambar 3. 24 *Run Collection* Tes API..... 54
Gambar 3. 25 Hasil dari Menjalankan *Collection Runner* 54
Gambar 3. 26 Memasukan *Username* dan *Password*..... 55

Gambar 3. 27 Menentukan Token yang Digunakan 55
Gambar 3. 28 Keterangan Menu yang Dapat Diakses Pengguna 56
Gambar 3. 29 Memasukan URL API Publik 56
Gambar 3. 30 Contoh Respons dari API Publik 56
Gambar 4. 1 Contoh Respons dari *Endpoint Login* 64
Gambar 4. 2 *Environment* 64
Gambar 4. 3 POST /login 65
Gambar 4. 4 Gagal *Login* 65
Gambar 4. 5 POST /logout 66
Gambar 4. 6 GET /profile 66
Gambar 4. 7 PUT /profile/signature 67
Gambar 4. 8 Contoh GET /users/{role} (role = admin) 68
Gambar 4. 9 Halaman Akses 76
Gambar 4. 10 *Authentication* 76
Gambar 4. 11 *Profile* 77
Gambar 4. 12 Admin - Proposal 77
Gambar 4. 13 Admin - Sidang Hasil 78
Gambar 4. 14 Admin - Sidang Akhir 78
Gambar 4. 15 Admin - *Users* (Dosen) 79
Gambar 4. 16 Admin - Detail Proposal 79
Gambar 4. 17 Bagian Administrasi - Proposal 80
Gambar 4. 18 Bagian Administrasi - Berkas Proposal 80
Gambar 4. 19 Dosen - Sidang Hasil 81
Gambar 4. 20 Dosen - Detail Sidang Hasil 81
Gambar 4. 21 Dosen - Verifikator 82
Gambar 4. 22 Mahasiswa - Sidang Hasil 82
Gambar 4. 23 Mahasiswa – Berkas 83
Gambar 4. 24 Mahasiswa - Ajukan Proses 83
Gambar 4. 25 Tampilan *Error - Forbidden* 84
Gambar 4. 26 Respons *Endpoint* /daftar-mahasiswa 85
Gambar 4.27 Respons *Endpoint* /daftar-mahasiswa?angkatan=2018 86
Gambar 4.28 Respons *Endpoint* /daftar-pengajuan 87
Gambar 4.29 Respons *Endpoint* /daftar-pengajuan?angkatan=2018 88

DAFTAR KODE PROGRAM

Kode Program 2. 1 Contoh *Fuction* Ambil Data 16
Kode Program 2. 2 Contoh *Method* GET *by* ID 16
Kode Program 2. 3 Contoh *Method* POST 17
Kode Program 2. 4 Contoh *Method* PUT..... 17
Kode Program 2. 5 Contoh *Method* DELETE 18
Kode Program 4. 1 *Route* 63
Kode Program 4. 2 *Controller* 63

BAB I
PENDAHULUAN

1.1 Latar Belakang

Program Studi Informatika merupakan salah satu program studi jenjang sarjana (S1) di lingkungan Fakultas Teknik Universitas Tanjungpura, berdiri pada tanggal 18 mei 2014 dengan SK Dirjen DIKTI Nomor 1664/D/T/2004. Saat ini penyelenggaraan Program Studi Informatika berdasarkan SK Penyelenggaraan Nomor 7325/D/T/K-N/2011 tanggal 6 Juni 2011, dan Terakreditasi B sesuai SK BAN-PT Nomor 0770/SK/BAN-PT/Akred/S/III/2017 tanggal 21 Maret 2017.

Program Studi Informatika saat ini memiliki sebuah sistem bernama E-Doxid (Elektronik Dokumen Sidang). E-Doxid merupakan aplikasi pemberkasan *online* berbasis *web* yang digunakan untuk melihat dan mengelola data proposal, sidang hasil dan sidang akhir. Pengguna dapat melakukan pengunggahan berkas yang kemudian dapat diverifikasi oleh staf Tata Usaha. Aplikasi E-Doxid adalah aplikasi *web* dimana memiliki kelemahan dalam hal notifikasi secara cepat, untuk menangani notifikasi E-Doxid menggunakan WA *Gateway* dimana notifikasi dikirim ke pengguna sesuai dengan proses yang ditangani. Selain fitur chat, jurusan Informatika berencana untuk mengembangkan aplikasi E-Doxid berbasis *mobile app*. Oleh sebab itu, diperlukan sebuah API pada aplikasi E-Doxid untuk komunikasi data antara aplikasi E-Doxid berbasis *web* dan aplikasi berbasis *mobile app*.

Perancangan *software* berbasis API akan memungkinkan sebuah *backend* dimanfaatkan secara lebih luas karena pembuatan *logic* pada sistem akan dilakukan secara terpisah dengan *logic* pada tampilan antarmuka pengguna. Penggunaan API juga mempermudah sebuah sistem untuk berkolaborasi dengan sistem lain, karena dengan menggunakan API adalah sebuah "penghubung" yang memungkinkan suatu sistem untuk berinteraksi dengan sistem lain. Hal ini juga berguna jika pada proses yang akan datang proses bisnis yang ada membutuhkan interaksi dengan sistem lain, atau dalam masa pengembangan modul-modul baru pada sistem terkait.

Begitu pula dengan gaya arsitektur REST yang akan diterapkan pada pengembangan API. REST adalah sebuah standar arsitektur berbasis *web* yang menggunakan protokol HTTP untuk berkomunikasi data. Arsitektur REST dapat digunakan oleh berbagai macam bahasa pemrograman dan *platform*, selain itu juga struktur kerja pada arsitektur ini juga sangat mudah untuk dipahami. Hal ini berguna untuk para pengembang lain dalam proses pemeliharaan dan pengembangan sistem.

Gagasan-gagasan di atas menjadi landasan perlunya dilakukan riset untuk merancang dan membangun *Application Programming Interface* (API) yang terintegrasi sebagai bagian dari pengembangan aplikasi E-Doxid.

1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah dalam penelitian ini adalah bagaimana membangun RESTful *Application Programming Interface* (API) untuk layanan pertukaran data E-Doxid.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang *Application Programming Interface* (API) menggunakan gaya arsitektur *Representational State Transfer* (REST) untuk pengembangan aplikasi E-Doxid agar data dapat terintegrasi dengan aplikasi lainnya.

1.4 Pembatasan Masalah

Adapun beberapa hal yang menjadi batasan masalah pada penelitian ini adalah sebagai berikut :

a. Pengembangan API dirancang berdasarkan format aplikasi E-Doxid Fakultas Teknik Universitas Tanjungpura.
b. Sistem yang dikembangkan berbasis *web* yang menggunakan bahasa pemrograman PHP dan MySql sebagai *database*.
c. API yang dibangun mencakup data-data yang dapat diakses dari pihak admin, bagian administrasi/tata usaha (TU), Ketua Jurusan, dosen, dan mahasiswa.
d. Data yang digunakan dalam pengembangan menggunakan data *dummy* dengan format sesuai *database* E-Doxid.

1.5 Sistematika Penulisan

Adapun sistematika dalam penulisan laporan skripsi ini ada lima bab yang terdiri dari Bab I Pendahuluan, Bab II Tinjauan Pustaka, Bab III Metodologi Penelitian, Bab IV Hasil dan Analisis, serta Bab V Penutup.

BAB I PENDAHULUAN, bab ini berisi tentang pendahuluan penelitian yang meliputi latar belakang, perumusan masalah, tujuan penelitian, dan pembatasan masalah.

BAB II TINJAUAN PUSTAKA, bab ini berisi tentang tinjauan pustaka yang meliputi kajian teori mengenai penelitian terkait dan landasan teori mengenai hal-hal yang berkaitan dengan penelitian ini.

BAB III METODOLOGI PENELITIAN, bab ini menjelaskan tentang metodologi penelitian yang meliputi langkah penelitian, metode pengambangan perangkat lunak, alat yang digunakan, dan rancangan sistem.

BAB IV HASIL DAN PEMBAHASAN, pada bab ini menjelaskan tentang hasil dan pembahasan yang berisi hasil pembuatan API dan pengujian untuk mendapatkan suatu kesimpulan.

BAB V PENUTUP, pada bab ini berisi tentang penutup yang meliputi kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian kedepannya.