

Pengembangan *Rest API* Untuk Monitoring Daerah Rawan Narkoba Menggunakan *Framework ExpressJs* Dengan Metode *Scrum*

Diptya Bagus Sumantry¹, Muhammad Lulu Latif Usman^{2,*}, Yohani Setiya Rafika Nur³

^{1,2,3} Teknik Informatika, Teknik Informatika, Institut Teknologi Telkom Purwokerto, Banyumas, Indonesia
Email: ¹diptyasoemantry@gmail.com, ^{2,*}lulu@ittelkom-pwt.ac.id-, ³yohani@ittelkom-pwt.ac.id

^{*)} Email *Corresponding Author*

Abstrak—Narkoba adalah obat terlarang yang dapat menyebabkan banyak gejala negatif jika disalahgunakan oleh pengguna. Di Kabupaten Banyumas terdapat sekitar 23.985 jiwa penyalahgunaan dan peredaran gelap narkoba pada tahun 2022. Dengan banyaknya perkiraan tersebut, Badan Narkotika Nasional Kabupaten (BNNK) Banyumas mempunyai tanggung jawab dalam pencegahan, pemberantasan, penyalahgunaan dan peredaran gelap narkoba di wilayah Kabupaten Banyumas dengan melaksanakan program Desa Bersih Narkoba (Desa Bersinar). Desa Bersinar merupakan kegiatan yang digunakan untuk pencegahan, pemberantasan, penyalahgunaan dan peredaran gelap narkoba. Kegiatan tersebut dimulai dari pemetaan wilayah daerah rawan narkoba yang nantinya digunakan untuk monitoring daerah rawan narkoba. Dalam melaksanakan kegiatan Desa Bersinar BNNK Banyumas masih melakukannya secara manual maka berdasarkan permasalahan tersebut penulis akan memberikan solusi yaitu mengembangkan *backend* dengan hasil *Rest API* yang nantinya digunakan untuk mengelola data dan monitoring daerah rawan narkoba yang dapat digunakan pada pengembangan *frontend*. Pengembangan *Rest API* akan menggunakan metode *scrum* dengan *framework ExpressJs*. Agar pengembangan ini dapat berjalan sesuai yang diharapkan maka akan dilakukan pengujian pada *Rest API* dengan *whitebox testing* dan *blackbox testing*. *Sprint* pertama dilaksanakan selama 2 minggu pada tanggal 24 juli 2023 – 6 Agustus 2023 dengan 17 *product backlog* dan total *story point planed* 60 jam dengan *story point actual* 60 jam. *Sprint* Kedua dilaksanakan selama 2 minggu pada tanggal 14 agustus 2023 – 27 agustus 2023 dengan 22 *product backlog* dan total *story point planed* 60 jam dengan *story point actual* 60 jam. Hasil dari setiap *sprint* menghasilkan *Rest API*. *Story point planed* dan *story point actual* akan menghasilkan *burndown chart* untuk melihat proses pekerjaan selama *sprint*. Penelitian ini dilakukan sesuai dengan waktu yang direncanakan dan *Rest API* dapat diterapkan pada pengembangan *frontend*.

Kata Kunci: *Rest API*, Narkoba, Desa bebas narkoba, *Scrum*, *ExpressJs*

Abstract— Narcotics are illegal drugs that can cause many negative symptoms if misused by users. In Banyumas Regency, there will be around 23,985 people who abuse and illicit drug trafficking in 2022. With this estimate, the Banyumas Regency National Narcotics Agency (BNNK) has the responsibility to prevent, eradicate, abuse and illicit drug trafficking in the Banyumas Regency area by implementing the Village program. Clean from Drugs (Shining Village). Bersinar Village is an activity used to prevent, eradicate, abuse and illicit drug trafficking. This activity starts with mapping drug-prone areas which will later be used for monitoring drug-prone areas. In implementing the Bersinar Village activities, the Banyumas BNNK still does it manually, so based on this problem the author will provide a solution, namely developing a backend with Rest API results which will later be used to manage data and monitor drug-prone areas which can be used in frontend development. Rest API development will use the Scrum method with the ExpressJs framework. So that this development can run as expected, testing will be carried out on the Rest API with whitebox testing and blackbox testing. The first sprint was carried out for 2 weeks on July 24 2023 – August 6 2023 with 17 product backlogs and a total of 60 hours of planned story points with 60 hours of actual story points. The Second Sprint will be implemented for 2 weeks on August 14 2023 – August 27 2023 with 22 product backlogs and a total of 60 hours of story points with actual story points of 60 hours. The results of each sprint produce a *Rest API* for each feature to be developed. Planned story points and actual story points will produce a burndown chart for each sprint to see the work process during the sprint. This research was carried out according to the planned time and *Rest API* can be applied to frontend development.

Keywords: Rest API , Drugs, Drug free village, Scrum, ExpressJs

1. PENDAHULUAN

Penyalahgunaan Narkotika, Psikotropika dan Bahan Adiktif berbahaya lainnya (Narkoba) menjadi masalah serius di Kabupaten Banyumas, hal ini dapat terjadi karena kurangnya pemahaman masyarakat terhadap bahaya Narkoba. Menurut kepala Badan Narkotika Nasional Kabupaten (BNNK) Banyumas, Muhammad Fierza Mucharom terdapat sekitar 23.985 orang melakukan penyalahgunaan Narkoba pada tahun 2022, dimana estimasi tersebut diukur melalui Pusat Penelitian Data dan Informasi Badan Narkotika Nasional (Puslitdatin BNN) dengan prevalensi tahun 2021 sebesar 1,95% [1], dan jumlah penduduk Kabupaten Banyumas usia 15 sampai 64 tahun sekitar 1,23 juta jiwa dari data Badan Pusat Statistik (BPS) tahun 2022 [2]. Estimasi penyalahgunaan narkoba cukup besar berada di Kabupaten Banyumas. BNNK Banyumas mempunyai tanggung jawab dalam pencegahan, pemberantasan, penyalahgunaan dan peredaran gelap Narkoba di wilayah Kabupaten Banyumas. Desa Bersinar

merupakan satuan wilayah setingkat kelurahan atau desa yang memiliki kriteria tertentu untuk melaksanakan program Pencegahan Pemberantasan Penyalahgunaan dan Peredaran Gelap Narkotika (P4GN) yang dilaksanakan secara massif. Penentuan kerawanan dan keterpulihan daerah rawan narkoba masih dilakukan secara manual menggunakan kertas. Penulis menawarkan ide solusi untuk mempermudah BNNK Banyumas dalam melakukan pengelolaan data dan monitoring dalam bentuk *Representational State Transfer Application Programming Interface* (Rest API) yang nantinya akan digunakan untuk pengembangan *frontend* pada aplikasi desa bersinar.

Framework *ExpressJs* merupakan salah satu kerangka kerja pada pengembangan *Rest API* yang berjalan pada *runtime NodeJs*. Dari hasil pengujian kinerja pada *framework Express.js* dengan *framework Codeigniter* menunjukkan bahwa rata-rata *respons* waktu dari *framework ExpressJs* adalah 420,72 *milisecon* lebih cepat dibandingkan *framework Codeigniter* yang memperoleh rata-rata *respons* waktu 555,90 *milisecon* [3]. Berdasarkan pemaparan tersebut, pemilihan *framework ExpressJs* pada pengembangan *Rest API* untuk aplikasi desa bersinar yaitu memiliki kecepatan dalam melakukan proses *respons* data.

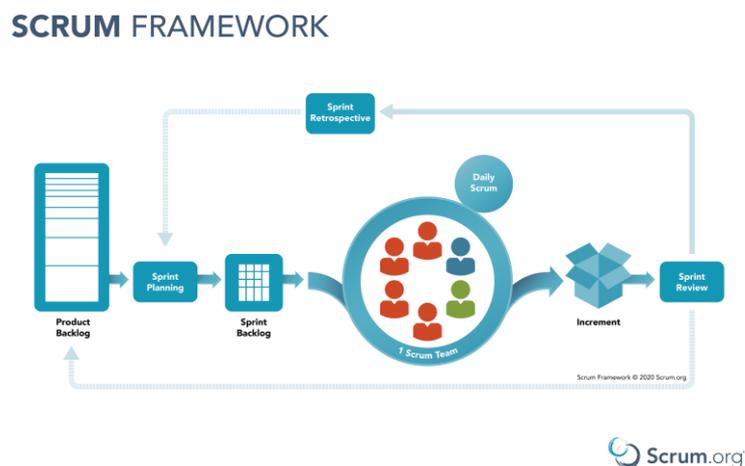
Metode *Scrum* merupakan kerangka kerja pada pengembangan perangkat lunak yang dilakukan secara berulang dengan mengidentifikasi penyebab masalah dan dilanjutkan berdiskusi dengan tim untuk mencari solusi dari setiap permasalahan [4]. Pada metode *scrum* akan menentukan waktu yang akan dibutuhkan untuk mengembangkan suatu fitur. Dalam penggunaan metode *scrum* akan fokus pada fitur yang akan dikembangkan dan melakukan diskusi mengenai apa yang sudah dilakukan dan apa yang akan dilakukan pada *sprint* berikutnya sesuai dengan waktu yang sudah ditentukan. Pada Penelitian ini merupakan bagian *backend* yang nantinya akan menghasilkan *Rest API*.

Pengujian dalam tahap pengembangan aplikasi sangat diperlukan untuk mengetahui apakah aplikasi dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Rencana pengujian yang digunakan pada *Rest API* yaitu menggunakan pengujian *whitebox* dan *blackbox*. Apabila terdapat kesalahan akan diperbaiki secepatnya dengan metode pengembangan *scrum*.

2. METODE PENELITIAN

2.1 Metode Pengembangan *Scrum*

Scrum merupakan kerangka kerja pada pengembangan perangkat lunak yang dilakukan secara berulang dengan mengidentifikasi penyebab masalah dan dilanjutkan berdiskusi dengan tim untuk mencari solusi dari setiap permasalahan [4]. Terdapat tahapan dalam proses pengembangan perangkat lunak pada metode *scrum*, tahapan tersebut yaitu sebagai berikut:



Gambar 1. Metode Pengembangan *Scrum* [5]

a. *Product Backlog*

Tahap *product backlog* adalah hal yang paling penting dalam pengembangan perangkat lunak menggunakan metode *scrum*. Pada tahap ini akan merubah ide menjadi suatu nilai yang nantinya akan dikerjakan untuk mencapai *product goal* [6].

b. *Sprint Planning*

Sprint Planning merupakan tahapan penting dalam memulai pekerjaan selama *sprint*. Pada tahap ini *product owner* akan melakukan rapat dengan anggota tim untuk memastikan *product backlog* yang akan dikerjakan dan membahas hubungan antara *product backlog* dengan *product goal* [6].

c. *Daily Scrum*

Daily Scrum merupakan tahapan pada *sprint* yang dilakukan untuk memastikan implementasi *product backlog* terus berjalan sesuai dengan *product goal* [6].

d. *Sprint Review*

Sprint Review adalah tahapan untuk mempresentasikan hasil dari *sprint* yang sudah dikerjakan kepada pemangku kepentingan [6].

e. *Sprint Retrospective*

Sprint Retrospective merupakan tahap yang dilakukan untuk merencanakan cara agar dapat meningkatkan kualitas dan efektivitas pada perangkat lunak yang sudah dibangun [6].

2.2 Pengujian Sistem

Pengujian dilakukan untuk mengetahui apakah *rest api* yang sudah dikembangkan sesuai dengan fungsi dan berjalan dengan baik. Pengujian dilakukan menggunakan metode *whitebox* dan *blackbox testing*.

1. *WhiteBox Testing*

Whitebox adalah metode pengujian perangkat lunak dengan cara menganalisa dan meneliti stuktur internal dan kode dari perangkat lunak [7]. Dalam pengujian menggunakan *whitebox* akan terdapat *flowchart*, *flowgraph*, *cycleomatic complexity*, *independet path* dan uji skenario. Perhitungan *cycleomatic complexity* yaitu dengan menghitung jumlah *edges* pada *flowgraph* dan jumlah *nodes* pada *flowgraph* [8][9]. Hasil dari jumlah *edges* dikurangkan dengan jumlah *nodes* dan ditambahkan dengan nilai 2. Dari hasil tersebut merupakan *independent patah* yang merupakan jumlah jalur yang dapat dilakukan untuk melakukan testing.

Tabel 1. Tingkar Resiko Kode Program

Path	Tipe Prosedur	Tingkat Resiko
1-4	Sederhana	Rendah
5-10	Struktur dan Stabil	Rendah
11-20	Kompleks	Menengah
21-50	Kompleks dan Kritis	Tinggi
>50	Rentan kesalahan, mengganggu, tidak dapat diuji	Sangat Tinggi

2. *BlackBox Testing*

Black box testing merupakan salah satu metode yang digunakan untuk melakukan pengujian fungsionalitas dengan melihat hasil eksekusi melalui data uji dan memastikan fungsi dari *software* [10]. Pengujian menggunakan *blackbox* dapat mengetahui masukan dan keluaran yang di masukan oleh user dapat berjalan dengan baik atau tidak [11], dan mengevaluasi kesesuaian dengan kebutuhan pengguna tanpa mengetahui struktur kode atau program.

2.3 Alat dan Bahan

Dalam penelitian ini memerlukan pegguaan alat dan bahan sebagai pendukung keberhasilan penelitian. Adapun alat dan bahan yang diperlukan yaitu sebagai berikut:

a. Alat

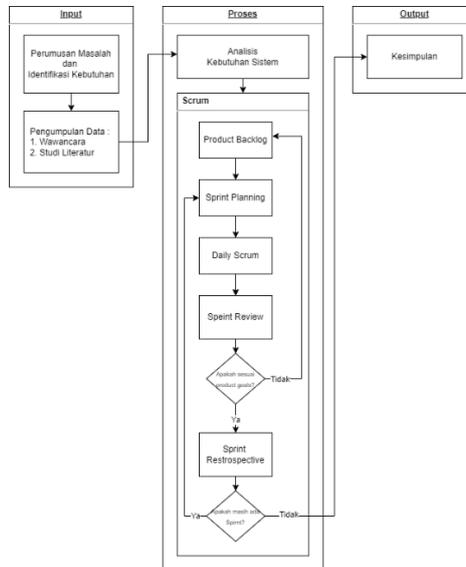
Device : Asus Vivobook S14
 Processor : Intel(R) Core (TM) i5-8265U CPU @ 1.80GHz
 RAM : 8GB

b. Bahan

Bahan yang digunakan dalam penelitian ini adalah formulir Indeks Keterpulihan Kawasan Rawan Narkoba (IKKR) [12].

2.4 Diagram Alir Penelitian

Diagram Alir merupakan salah satu tahap untuk melakukan penyusunan suatu penelitian. Penyusunan tahapan ini penting untuk dilakukan, agar penelitian dapat berjalan dengan lancar dan mencapai hasil yang diharapkan. Berikut merupakan diagram alir yang akan dilakukan untuk penyusunan laporan penelitian ini:



Gambar 2. Diagram Alir Penelitian

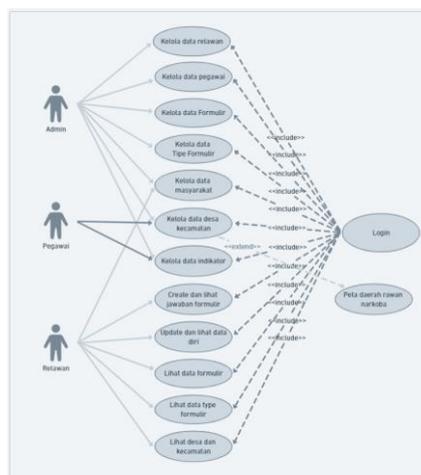
3. HASIL DAN PEMBAHASAN

3.1 Hasil

3.1.1 Analisis Kebutuhan Sistem

Tahap Analisis Kebutuhan Sistem merupakan tahapan penting dalam pengembangan aplikasi desa bersinar. Pada tahap ini peneliti akan melakukan desain sistem yang akan digambarkan dalam bentuk *User Modeling Language* (UML) dan *Entity Relationship Diagram* (ERD) [13].

a. Use Case Diagram

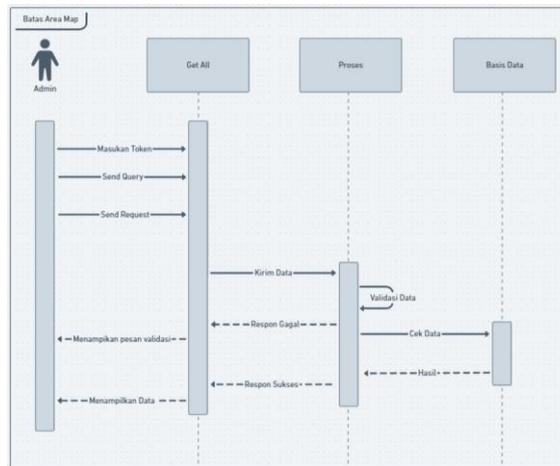


Gambar 3. Use Case Diagram

Pada gambar 3 terlihat role admin dapat mengakses data relawan, pegawai, formulir, tipe formulir, masyarakat, desa kecamatan dan indikator. Role pegawai dapat mengakses data desa kecamatan dan

indikator. Role Relawan dapat mengakses data Masyarakat, membuat dan melihat jawaban formulir, edit dan lihat data pribadi, lihat data formulir, lihat tipe formulir, dan lihat data desa dan kecamatan

b. Sequence Diagram Pemetaan



Gambar 4. Sequence Diagram Pemetaan

Rest API batas area map dapat diakses oleh role admin. Admin akan memasukkan *authorization* token dan *query* untuk melakukan filter data berdasarkan rentan waktu. Token dan *query* yang telah dimasukan lalu dikirimkan untuk proses validasi oleh sistem. Jika terdapat data yang tidak sesuai maka akan menampilkan pesan validasi gagal. Jika data yang dimasukan sesuai maka, akan melakukan cek data pada basis data dan menampilkan area desa dan status desa tersebut.

c. Entity Relationship Diagram (ERD)

Entity relationship diagram yang menggambarkan relasi antar *table* pada basis data aplikasi desa bersinar. Terdapat 9 *entity* yang digunakan dalam pengembangan aplikasi desa bersinar. Setiap *entity* mempunyai atribut yang dapat saling berelasi antar *table* dan mempunyai *primary key* serta *foreign key*.

3.1.2 Product Backlog

Pada tahap ini akan menghasilkan fitur yang akan dikerjakan berdasarkan *product goals* dan terdapat prioritas dan waktu untuk mengerjakan *product backlog* dalam setiap *sprint*. Berikut tabel *product goals*:

Tabel 2. Product backlog

No	Product Goals	Product backlog	Prioritas	Waktu
1	Menghasilkan <i>Environment</i> yang bersih	Perancangan <i>Framework</i> <i>Add Models & Migrate table database</i>	Pertama	60 Jam (<i>Sprint</i>)
2	Data otomatis ditambahkan ketika program pertama kali dijalankan	Otomatis menambahkan akun admin, menambahkan data desa kecamatan dan tipe formulir	Pertama	Pertama
3	Melakukan otorisasi dan autentikasi	Membuat <i>login</i> Membuat <i>fetch</i> <i>Endpoint logout</i>	Pertama	
4	Dapat Melakuaan pembuatan, melihat, edit dan hapus data pada relawan	Membuat <i>create</i> relawan Membuat <i>get all</i> relawan Membuat <i>get detail</i> relawan Membuat edit relawan Membuat <i>delete</i> relawan	Pertama	
5	Dapat Melakuaan pembuatan, melihat, edit dan hapus data pada masyarakat	Membuat <i>create</i> masyarakat Membuat <i>get</i> masyarakat Membuat detail masyarakat Membuat edit masyarakat	Pertama	

6	Dapat Melakukan pembuatan, melihat, edit dan hapus data pada pegawai	Membuat <i>delete</i> masyarakat Membuat <i>create</i> pegawai Membuat <i>get all</i> pegawai Membuat detail pegawai Membuat edit pegawai Membuat <i>delete</i> pegawai	Kedua	60 Jam (<i>Sprint</i> Kedua)
7	Dapat Melakukan pembuatan, melihat, edit dan hapus data pada <i>Type</i> Formulir	Membuat <i>create</i> tipe formulir Membuat <i>get</i> tipe formulir Membuat <i>get detail</i> tipe formulir Membuat edit tipe Formulir Membuat <i>delete</i> <i>Type</i> Formulir	Kedua	
8	Dapat Melakukan pembuatan, melihat, edit dan hapus data pada Formulir	Membuat <i>create</i> dan edit formulir Membuat <i>get</i> formulir IKKR Membuat <i>get</i> formulir <i>by</i> tipe formulir Membuat <i>delete</i> formulir	Kedua	
9	Dapat Melakukan pembuatan dan melihat jawaban	Membuat <i>Create Answer</i> Membuat <i>get Answer by</i> masyarakatId	Kedua	
10	Dapat Melakukan melihat data desa kecamatan rawan narkoba	Membuat <i>get list</i> desa dan kecamatan Membuat <i>get</i> batas area peta desa	Kedua	
11	Dapat Melakukan pembuatan, melihat, edit dan hapus data pada Indikator	Membuat <i>create</i> indikator desa Membuat <i>get</i> indikator <i>by</i> desaKecamatan Membuat <i>delete</i> indikator	Kedua	
12	Dapat menampilkan total data pada dashboard	Membuat <i>get</i> menampilkan Jumlah Relawan, Jumlah Masyarakat, Jumlah Formulir	Kedua	

3.1.3 Sprint Pertama

a. *Sprint Planning*

Sprint pertama dilaksanakan selama 2 minggu yaitu dari tanggal 24 Juli – 6 Agustus 2023. Berikut merupakan perencanaan pada *sprint* pertama pada pengembangan *rest api* aplikasi desa bersinar:

Tabel 3. *Sprint Planning – Sprint Pertama*

No	<i>Product backlog</i>	<i>Date</i>	<i>Story Point Planed</i>
1	Perancangan Framework	24 Juli 2023	5
2	Add Models & Migrate ke database	25 Juli 2023	5
3	Otomatis menambahkan akun admin	26 Juli 2023	5
4	Otomatis menambahkan akun admin, data desa kecamatan dan tipe formulir	27 Juli 2023	5
5	Membuat <i>login</i>	28 Juli 2023	5
6	Membuat <i>endpoint fetch</i>	29 Juli 2023	5
7	Membuat <i>logout</i>	30 Juli 2023	5
8	Membuat <i>create</i> relawan	31 Juli 2023	5
9	Membuat <i>get</i> semua relawan	31 Juli 2023	5
10	Membuat <i>get</i> detail relawan	31 Juli 2023	5

11	Membuat edit relawan	1 Agustus 2023	5
12	Membuat <i>delete</i> relawan	2 Agustus 2023	5
13	Membuat <i>create</i> masyarakat	3 Agustus 2023	5
14	Membuat <i>get</i> semua masyarakat	4 Agustus 2023	5
15	Membuat detail masyarakat	5 Agustus 2023	5
16	Membuat edit masyarakat		
17	Membuat <i>delete</i> masyarakat		
Total			60

b. *Daily Scrum*

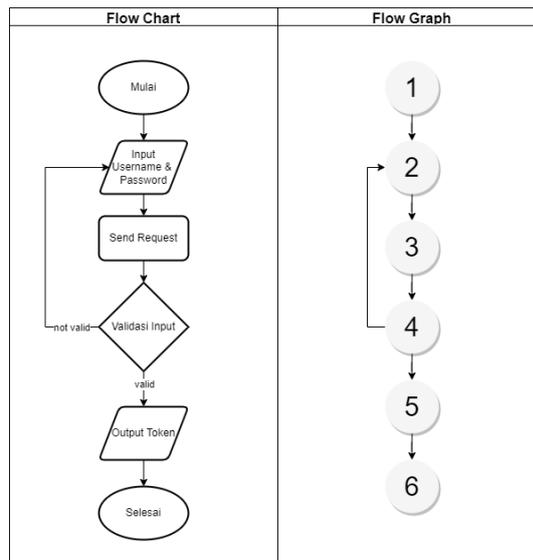
Berikut merupakan hasil daily scrum pada *sprint* pertama yang sudah dikerjakan:

Tabel 4. Hasil *Daily Scrum* – *Sprint* Pertama

No	Product backlog	API						Date	Story Point Actual
		M	P	Q	H	B	A		
1	Perancangan <i>Framework</i>	x	x	x	x	x	x	24 Juli 2023	8
2	<i>Add Models & Migrate</i> ke database	x	x	x	x	x	x	25 Juli 2023	8
3	Otomatis menambahkan akun admin	x	x	x	x	x	x	26 Juli 2023	3
4	Otomatis menambahkan data desa kecamatan dan tipe formulir	x	x	x	x	x	x	27 Juli 2023	5
5	Membuat <i>login</i>	<i>POST</i>	x	x	x	v	x	28 Juli 2023	3
6	Membuat <i>endpoint fetch</i>	<i>GET</i>	x	x	x	x	v		1
7	Membuat <i>logout</i>	<i>POST</i>	x	x	x	x	v	29 Juli 2023	1
8	Membuat <i>create</i> relawan	<i>POST</i>	x	x	x	v	v		5
9	Membuat <i>get</i> semua relawan	<i>GET</i>	x	v	x	x	v	31 Juli 2023	3
10	Membuat <i>get</i> detail relawan	<i>GET</i>	v	x	x	x	v		3
11	Membuat edit relawan	<i>PUT</i>	v	x	x	v	v	1 Agustus 2023	5
12	Membuat <i>delete</i> relawan	<i>DELETE</i>	v	x	x	x	v	3 Agustus 2023	1
13	Membuat <i>create</i> masyarakat	<i>POST</i>	x	x	x	v	v		5
14	Membuat <i>get</i> semua masyarakat	<i>GET</i>	x	v	x	x	v	3 Agustus 2023	3

							4		
15	Membuat detail masyarakat	GET	v	x	x	x	v	Agustus 2023	2
16	Membuat edit masyarakat	PUT	v	x	x	v	v	5 Agustus 2023	3
17	Membuat delete masyarakat	DELETE	v	x	x	x	v		1
Total									60

Pengujian menggunakan *WhiteBox Testing* pada endpoint *login*:



Gambar 5 . Flow Login

Menentukan jumlah jalur ($v(G)$) dilakukan perhitungan sebagai berikut:

$$V(G) = Edges - Nodes + 2$$

$$V(G) = 6 - 6 + 2$$

$$v(G) = 2$$

Dari hasil perhitungan $v(G)$ terdapat 2 jalur untuk pengujian dengan skenario sebagai berikut:

Jalur 1 = 1-2-3-4-5-6

Jalur 2 = 1-2-3-4-2-3-4-5-6

Tabel 5. White Box Login

Path	1	2
Jalur	1-2-3-4-5-6	1-2-3-4-2-3-4-5-6
Skenario	1. Mulai 2. <i>Input username dan password</i> 3. <i>Send request</i> 4. <i>Validasi Input</i> 5. <i>Output token</i> 6. Selesai	1. Mulai 2. <i>Input username dan password</i> 3. <i>Send request</i> 4. <i>Validasi Input</i> 2. <i>Input username dan password</i> 3. <i>Send request</i> 4. <i>Validasi Input</i>

		5. <i>Output</i> token
		6. Selesai
Hasil Pengujian	Berhasil	Berhasil

Padaa tabel 1 pengujian whitebox login pada 2 jalur berhasil di jalankan.

Tabel 6. *Black Box Login*

No	Role	Skenario	Hasil yang Diharapkan	Keterangan
1	Admin, Pegawai, Relawan	Memasukan <i>username</i> dan <i>password</i> yang sesuai pada database	Menampilkan access token	Berhasil
		Memasukan <i>username</i> tidak terdaftar	Menampilkan pesan <i>error</i> " <i>username</i> tidak ditemukan"	Berhasil
2	Admin, Pegawai, Relawan	Memasukan <i>password</i> yang salah	Menampilkan pesan <i>error</i> " <i>password</i> anda salah"	Berhasil
		Memasukan <i>username</i> dan <i>password</i> kosong	Menampilkan pesan <i>error</i> " <i>username</i> tidak boleh kosong" dan " <i>password</i> tidak boleh kosong"	Berhasil

Pengujian fungsional pada tabel 6 blackbox login berhasil di jalankan sesuai dengan skenario.

c. Sprint Review

Pada tahap *sprint review* peneliti dan tim akan mendemonstrasikan setiap *product backlog* yang sudah dikerjakan kepada pemangku kepentingan. Hasil dari diskusi tersebut, dihasilkan bahwa pada *sprint* pertama *product backlog* berhasil diselesaikan sesuai dengan *product goals* dan *product backlog* diselesaikan dengan tepat waktu.

d. Sprint Retrospective

Pada tahap *sprint retrospective*, peneliti dan tim melakukan evaluasi selama mengerjakan *sprint* pertama. Hasil pada tahap ini dapat berjalan dengan baik dan tidak ada masalah yang ditemukan. Peneliti menyelesaikan *sprint* dengan tepat waktu dan akan melanjutkan *sprint* berikutnya.

3.1.4 Sprint Kedua

a. Sprint Planning

Sprint kedua dilaksanakan selama 2 minggu yaitu dari tanggal 14 Agustus – 26 Agustus 2023. Berikut merupakan perencanaan pada *sprint* kedua di aplikasi desa bersinar:

Tabel 7. *Sprint Planing – Sprint Kedua*

No	Product backlog	Date	Story Point Planed
1	Membuat <i>create</i> pegawai		
2	Membuat <i>get all</i> pegawai		
3	Membuat detail pegawai	14 Agustus 2023	5
4	Membuat edit pegawai		
5	Membuat <i>delete</i> pegawai		
6	Membuat <i>create</i> tipe formulir	15 Agustus 2023	5

7	Membuat <i>get</i> tipe formulir		
8	Membuat <i>get detail</i> tipe formulir		
9	Membuat edit tipe Formulir		
10	Membuat <i>delete Type</i> Formulir		
11	Membuat <i>create</i> dan edit formulir		
12	Membuat <i>get</i> formulir IKKR	16 Agustus 2023	5
13	Membuat <i>get</i> formulir <i>by</i> tipe formulir		
14	Membuat <i>delete</i> formulir	17 Agustus 2023	5
15	Membuat <i>Create Answer</i>	18 Agustus 2023	5
16	Membuat <i>get Answer by</i> masyarakatId	19 Agustus 2023	5
17	Membuat <i>get list</i> desa dan kecamatan	21 Agustus 2023	5
18	Membuat <i>get</i> batas area peta desa	22 Agustus 2023	5
19	Membuat <i>create</i> indikator desa	23 Agustus 2023	5
20	Membuat <i>get</i> indikator <i>by</i> desaKecamatan	24 Agustus 2023	5
21	Membuat <i>delete</i> indikator	25 Agustus 2023	5
	Membuat <i>get</i> menampilkan		
22	Jumlah Relawan, Jumlah Masyarakat, Jumlah Formulir	26 Agustus 2023	5
Total			60

b. *Daily Scrum*

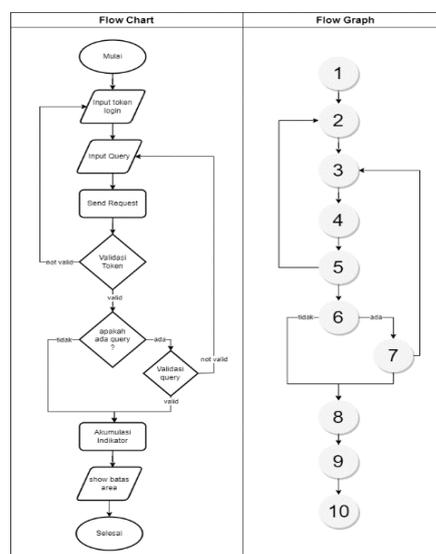
Berikut merupakan hasil daily scrum pada *sprint* pertama:

Tabel 8. Hasil Daily Scrum – Sprint Pertama

No	Product backlog	API						Date	Story Point Actual
		M	P	Q	H	B	A		
1	Membuat <i>create</i> pegawai	POST	x	x	x	v	v		3
2	Membuat <i>get all</i> pegawai	GET	x	v	x	x	v		1
3	Membuat detail pegawai	GET	v	x	x	x	v	14 Agustus 2023	1
4	Membuat edit pegawai	PUT	v	x	x	v	v		3
5	Membuat <i>delete</i> pegawai	DELETE	v	x	x	x	v		1
6	Membuat <i>create</i> tipe formulir	POST	x	x	x	v	v		3
7	Membuat <i>get</i> tipe formulir	GET	x	x	x	x	v	15 Agustus 2023	1
8	Membuat <i>get detail</i> tipe formulir	GET	v	x	x	x	v		3
9	Membuat edit tipe Formulir	PUT	v	x	x	v	v		3

10	Membuat <i>delete Type</i> Formulir	<i>DELETE</i>	v	x	x	x	v		1
11	Membuat <i>create</i> dan edit formulir	<i>POST</i>	x	x	x	v	v	16 Agustus 2023	5
12	Membuat <i>get</i> formulir IKKR	<i>GET</i>	x	x	x	x	v		3
13	<i>Get</i> formulir <i>by tipe</i> formulir	<i>GET</i>	v	x	x	x	v		3
14	Membuat <i>delete</i> formulir	<i>DELETE</i>	v	x	x	x	v	17 Agustus 2023	1
15	Membuat <i>Create Answer</i>	<i>POST</i>	v	x	x	v	v	18 Agustus 2023	7
16	<i>Get Answer by</i> masyarakatId	<i>GET</i>	v	x	x	x	v	19 Agustus 2023	3
17	Membuat <i>get list</i> desa dan kecamatan	<i>GET</i>	x	v	x	x	v	21 Agustus 2023	3
18	Membuat <i>get</i> batas area peta desa	<i>GET</i>	x	v	x	x	v	22 Agustus 2023	7
19	Membuat <i>create</i> indikator desa	<i>POST</i>	x	x	x	v	v	23 Agustus 2023	3
20	Membuat <i>get</i> indikator <i>by</i> desaKecamatan	<i>GET</i>	v	x	x	x	v	24 Agustus 2023	3
21	Membuat <i>delete</i> indikator	<i>DELETE</i>	v	x	x	x	v		1
22	Membuat <i>get</i> menampilkan Jumlah Relawan, Jumlah Masyarakat, Jumlah Formulir	<i>GET</i>	x	x	x	x	v	25 Agustus 2023	1
Total									60

Pengujian menggunakan *WhiteBox Testing* pada *endpoint* batas area map:



Gambar 6. Flow Batas Area Map

Menentukan jumlah jalur ($v(G)$) dilakukan perhitungan sebagai berikut:

$$V(G) = E - N + 2$$

$$V(G) = 12 - 10 + 2$$

$$v(G) = 4$$

Dari hasil perhitungan $v(G)$ terdapat 4 jalur untuk pengujian dengan skenario sebagai berikut:

Jalur 1 = 1-2-3-4-5-6-7-8-9-10

Jalur 2 = 1-2-3-4-5-6-8-9-10

Jalur 3 = 1-2-3-4-5-6-7-3-4-5-6-7-8-9-10

Jalur 4 = 1-2-3-4-5-2-3-4-5-6-7-8-9-10

Tabel 9. White Box Batas Area Map

Path	1	2
Jalur	1-2-3-4-5-6-7-8-9-10	1-2-3-4-5-6-8-9-10
Skenario	1. Mulai 2. <i>Input token login</i> 3. <i>Input query</i> 4. <i>Send request</i> 5. <i>Validasi token</i> 6. Apakah ada query 7. <i>Validasi query</i> 8. Akumulasi Indikator 9. Show batas area 10. Selesai	1. Mulai 2. <i>Input token login</i> 3. <i>Input query</i> 4. <i>Send request</i> 5. <i>Validasi token</i> 6. Apakah ada query 8. Akumulasi Indikator 9. Show batas area 10. Selesai
Hasil Pengujian	Berhasil	Berhasil
Path	3	4
Jalur	1-2-3-4-5-6-7-3-4-5-6-7-8-9-10	1-2-3-4-5-2-3-4-5-6-7-8-9-10
Skenario	1. Mulai 2. <i>Input token login</i> 3. <i>Input query</i> 4. <i>Send request</i> 5. <i>Validasi token</i> 6. Apakah ada query 7. <i>Validasi query</i> 3. <i>Input query</i> 4. <i>Send request</i> 5. <i>Validasi token</i> 6. Apakah ada query 7. <i>Validasi query</i>	1. Mulai 2. <i>Input token login</i> 3. <i>Input query</i> 4. <i>Send request</i> 5. <i>Validasi token</i> 2. <i>Input token login</i> 3. <i>Input query</i> 4. <i>Send request</i> 5. <i>Validasi token</i> 6. Apakah ada query 7. <i>Validasi query</i> 8. Akumulasi Indikator

	8. Akumulasi Indikator	9. Show batas area
	9. Show batas area	10. Selesai
	10. Selesai	
Hasil Pengujian	Berhasil	Berhasil

Hasil pengujian pada tabel 9 white box testing sebanyak 4 jalur berhasil dijalankan.

Tabel 10. *Black Box* Batas Area Map

No	Role	Skenario	Hasil yang Diharapkan	Keterangan
1	Admin	Memasukan token <i>login</i> yang sudah kadaluarsa	Menampilkan pesan <i>error</i> " <i>Unauthorized</i> "	Berhasil
		Tidak memasukan token <i>login</i>	Menampilkan pesan <i>error</i> " <i>forbidden</i> "	Berhasil
2	Admin	Memasukan token <i>login</i> admin dan <i>Input query</i>	Menampilkan data berdasarkan waktu	Berhasil
		Memasukan token <i>login</i> admin dan tidak menambahkan <i>query</i>	Menampilkan data semua batas area	Berhasil
		Memasukan token <i>login</i> selain <i>role</i> admin dan relawan	Menampilkan pesan <i>error</i> <i>Your role is not allowed!</i>	Berhasil

Pengujian fungsional pada tabel 10 *blackbox* batas area map berhasil dijalankan sesuai dengan skenario.

c. Sprint Review

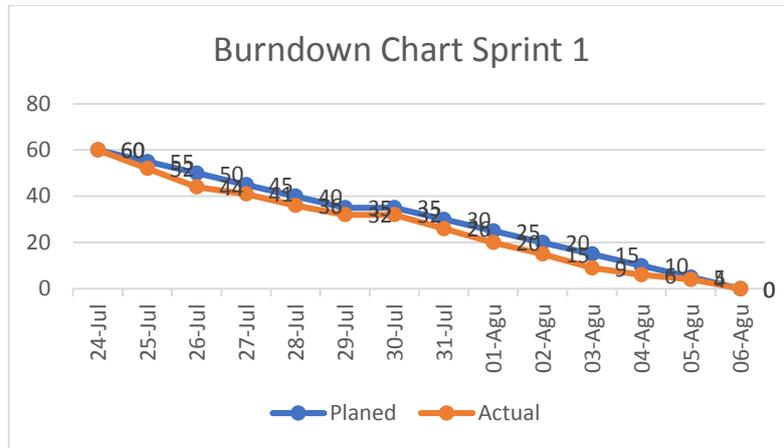
peneliti berdiskusi dengan pemangku kepentingan apakah *product backlog* yang sudah dikerjakan sesuai dengan *product goals* dan apakah masih ada *product backlog* yang tersisa. Hasil dari diskusi tersebut, dihasilkan bahwa pada *sprint* kedua *product backlog* berhasil diselesaikan sesuai dengan *product goals* dan *product backlog* diselesaikan dengan tepat waktu.

d. Sprint Retrospective

Pada tahap *sprint retrospective* di *sprint* kedua, peneliti dan tim melakukan evaluasi selama mengerjakan *sprint* kedua. Hasil pada tahap ini dapat berjalan dengan baik dan tidak ada masalah yang ditemukan. Peneliti menyelesaikan *sprint* kedua dengan tepat waktu.

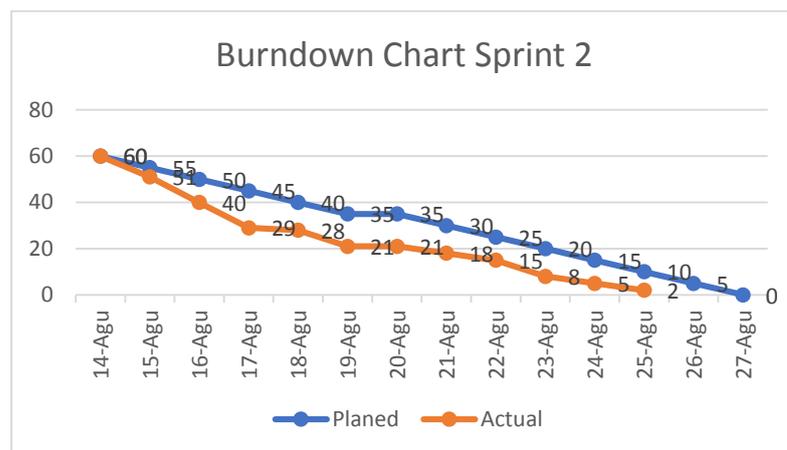
3.2 Pembahasan

3.2.1. Penerapan Metode Pengembangan SCRUM



Gambar 7. Burndown Chart Sprint 1

Garis warna *orange* menggambarkan waktu yang dikerjakan sebenarnya dan garis biru menggambarkan waktu perencanaan *sprint* selesai. *Sprint* 1 dilakukan sesuai dengan perencanaan yaitu selesai pada tanggal 6 agustus 2023



Gambar 8. Burndown Chart Sprint 2

Garis warna *orange* menggambarkan waktu pengerjaan sebenarnya dan garis biru waktu perencanaan *sprint* selesai. *Sprint* 2 selesai lebih cepat yaitu pada tanggal 25 agustus 2023 dari waktu perencanaan 27 agustus 2023.

3.2.2. Performa Rest API



Gambar 9. Speed Endpoint Batas Area Map

Performa waktu yang dihasilkan ketika mengakses *endpoint* batas area map untuk melakukan *monitoring* daerah rawan narkoba mendapatkan waktu 231.4 *millisecond* untuk menampilkan data rawan narkoba dari setiap desa dengan 100 indikator. Dari hasil yang sudah dilakukan pada pengujian *whitebox* mendapatkan hasil bahwa semua jumlah jalur terkecil dua jalur dan jalur terbesar terdapat enam jalur. Hasil jalur tersebut memiliki tingkat resiko rendah. Hasil pengujian *blackbox* yang sudah dilakukan pada semua *endpoint* dapat berfungsi sesuai dengan skenario dan hasil yang diharapkan. Setiap *endpoint* akan menampilkan pesan *error* jika pengujian tidak sesuai dengan skenario dan akan menampilkan pesan berhasil sesuai dengan *request* yang dilakukan.

4. KESIMPULAN

Dari penelitian yang sudah dilakukan oleh peneliti pada pengembangan *rest api* menggunakan *framework expressjs*, pengujian *whitebox testing*, *blackbox testing* dan metode pengembangan *scrum* dengan melakukan 2 kali *sprint*, memberikan kesimpulan sebagai berikut: (1) Pengembangan *rest api* yang sudah dilakukan menggunakan *framework expressjs* dapat melakukan penambahan, menampilkan, mengedit dan menghapus data pada basis data. *Rest api* yang sudah dikembangkan dapat digunakan tim *frontend* dan *mobile* untuk pengelolaan data dan *monitoring* daerah rawan narkoba. (2) Pengembangan *rest api* menggunakan metode pengembangan *scrum* yang dimulai dari menentukan *product backlog* hingga *sprint retrospective* berhasil dilakukan. *Sprint* dilakukan 2 kali dalam pekerjaan *product backlog* dengan melakukan pengujian *whitebox* dan *blackbox testing*. Hasil dari pengujian setiap *product backlog* berhasil sesuai dengan skenario dan memiliki tingkat resiko yang rendah.

REFERENCES

- [1] Indonesia. Badan Narkotika Nasional. Pusat Penelitian Data dan Informasi, *National survey on drug abuse, 2021*.
- [2] Badan Pusat Statistik Kabupaten Banyumas, "Jumlah Penduduk Menurut Kelompok Umur dan Jenis Kelamin di Kabupaten Banyumas." Accessed: Jun. 12, 2023. [Online]. Available: <https://banyumaskab.bps.go.id/statictable/2016/09/29/108/jumlah-penduduk-menurut-kelompok-umur-dan-jenis-kelamin-di-kabupaten-banyumas-2015.html>
- [3] L. Mulana, K. Prihandani, A. Rizal, U. Singaperbanga, and K. Abstract, "Analisis Perbandingan Kinerja Framework Codeigniter Dengan Express.Js Pada Server RESTful Api," *Jurnal Ilmiah Wahana Pendidikan*, vol. 8, no. 16, pp. 316–326, 2022, doi: 10.5281/zenodo.7067707.
- [4] H. A. Lalu Mutawali, Buyung Kurnia Fathoni, "Implementasi Scrum Dalam Pengembangan Sistem Informasi Jasa Desain Grafis," *Manajemen Informatika Dan Sistem Informasi*, vol. 3, no. 2, pp. 116–122, 2020.
- [5] "Gambar Metode Scrum," <https://www.scrum.org/node/646>.
- [6] Ken Schwaber & Jeff Sutherland, *Panduan Scrum*. 2020. [Online]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Indonesian.pdf>
- [7] H. Gusdevi, S. Kuswayati, M. Iqbal, M. F. Abu Bakar, N. Novianti, and R. Ramadan, "Pengujian White-Box Pada Aplikasi Debt Manager Berbasis Android," *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika*, vol. 4, no. 1, pp. 11–22, 2022, doi: 10.53580/naratif.v4i1.147.
- [8] H. Gusdevi, S. Kuswayati, M. Iqbal, M. F. Abu Bakar, N. Novianti, and R. Ramadan, "Pengujian White-Box Pada Aplikasi Debt Manager Berbasis Android," *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika*, vol. 4, no. 1, pp. 11–22, 2022, doi: 10.53580/naratif.v4i1.147.
- [9] A. Setiawan, M. R. D. Ananda, R. Alvario, and ..., "Pengujian Sistem Informasi Aplikasi Perpustakaan Berbasis Web Di SMAN 1 Gunung Sindur Dengan White Box Testing," ... *Sacra: Jurnal Sains ...*, vol. 2, no. 1, pp. 180–188, 2022, [Online]. Available: <http://www.pijarpemikiran.com/index.php/Scientia/article/view/148>
- [10] Rangga Gelar Guntara and V. Azkarin, "Implementasi dan Pengujian REST API Sistem Reservasi Ruang Rapat dengan Metode Black Box Testing," *Jurnal Minfo Polgan*, vol. 12, no. 1, pp. 1229–1238, 2023, doi: 10.33395/jmp.v12i1.12691.
- [11] Rangga Gelar Guntara and V. Azkarin, "Implementasi dan Pengujian REST API Sistem Reservasi Ruang Rapat dengan Metode Black Box Testing," *Jurnal Minfo Polgan*, vol. 12, no. 1, pp. 1229–1238, 2023, doi: 10.33395/jmp.v12i1.12691.
- [12] "Formulir IKKR," https://perpustakaan.bnn.go.id/sites/default/files/Buku_Digital_2020-08/Juknis%20IKKR.pdf.
- [13] K. 'Afifah, Z. F. Azzahra, and A. D. Anggoro, "Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review," *Intech*, vol. 3, no. 2, pp. 18–22, 2022, doi: 10.54895/intech.v3i2.1682.