

BAB III

METODOLOGI PENELITIAN

3.1 Subjek dan Objek Penelitian

Penelitian ini telah memusatkan perhatian pada pengembangan "Sistem Manajemen Laundry Berbasis *Website* dengan *Notifikasi Whatsapp*". Subjek utama dari penelitian ini adalah pengelola Bingo Laundry, termasuk admin dan pegawai, yang telah difokuskan dalam analisis utama dalam pengembangan sistem. Objek penelitian ini adalah rancang dan bangun website yang direncanakan untuk diimplementasikan dan diuji di Bingo Laundry. Penelitian ini mencakup semua aspek yang terlibat dalam operasional harian, layanan kepada pelanggan, manajemen pesanan, serta interaksi dengan sistem yang sedang dikembangkan. Dengan demikian, penelitian ini bertujuan untuk menggali hubungan antara pengembangan sistem manajemen laundry dengan integrasi *notifikasi WhatsApp* di Bingo Laundry sebagai pusat implementasi utama.

3.2 Alat dan Bahan Penelitian

Alat dan bahan yang akan digunakan pada penelitian ini meliputi :

3.2.1 Perangkat Keras (Hardware)

Beberapa perangkat keras yang diperlukan dalam penelitian ini.

1. Perangkat keras Acer Swift SF314-41
2. Processor AMD Ryzen 5 3500Uu (8 CPUs), 2.1GHz
3. Memory 8 Gb
4. Storage 512 Gb

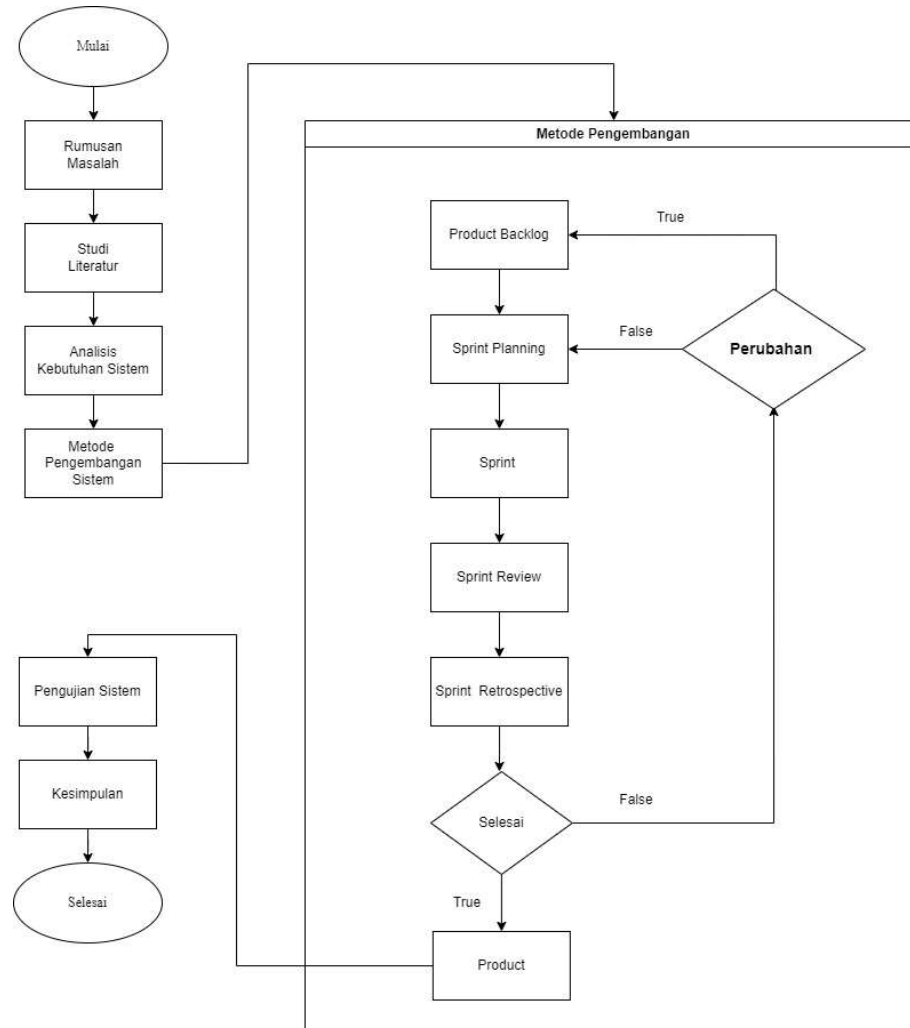
3.2.2 Perangkat lunak (Software)

1. Sistem Operasi *Windows 10 Home*
2. Visual Studio Code
3. Office Home and Student
4. Google Chrome
5. Figma
6. Google Form
7. Draw.io

3.2.3 Bahan

Bahan yang menjadi landasan dalam penelitian ini meliputi hasil wawancara yang dilakukan oleh peneliti dengan pegawai Bingo Laundry. Wawancara ini berfungsi sebagai sumber informasi utama untuk memahami proses operasional, tantangan yang dihadapi, dan kebutuhan yang perlu diatasi melalui perancangan sistem manajemen laundry berbasis website ini.

3.3 Diagram Alir Penelitian



Gambar 3. 1 Diagram Alir Penelitian

Gambar 3.1 menggambarkan proses pengembangan yang dimulai dengan tahap perumusan masalah yang kemudian diikuti dengan studi literatur guna merancang metode pengembangan menggunakan pendekatan *scrum*. Saat produk atau sistem telah selesai, proses berlanjut ke tahap pengujian menggunakan *blackbox testing* guna mengevaluasi tingkat kegunaan sebelum menarik kesimpulan akhir.

3.3.1 Rumusan Masalah

Penelitian ini berawal dari pertanyaan bagaimana merancang Sistem Manajemen Laundry dengan penambahan fitur dan fungsi yang memanfaatkan teknologi *web*. Fokus utama adalah memperluas kemampuan sistem yang digunakan dalam mengelola layanan laundry, dengan mengintegrasikan teknologi *web* guna memberikan lebih banyak fitur yang mendukung operasional bisnis *laundry*. Hal ini telah menjadi landasan dalam merancang sistem yang bisa merasakan pengalaman yang lebih baik untuk pelanggan serta mendukung operasional di Bingo Laundry. Dengan memanfaatkan teknologi *web*, diharapkan sistem ini mampu memberikan layanan yang lebih responsif.

3.3.2 Studi Literatur

Penelitian ini didahului oleh tinjauan mendalam terhadap literatur yang relevan dalam pengembangan Sistem Manajemen Laundry dengan pendekatan pemanfaatan teknologi *web*, khususnya pada lingkungan operasional Bingo Laundry. Fokus utama studi literatur mencakup eksplorasi terhadap sistem manajemen usaha layanan, terutama dalam konteks industri laundry di Bingo Laundry. Kajian tersebut mempertimbangkan penggunaan teknologi *web*, seperti integrasi *website* dan *notifikasi*, serta fitur-fitur lain yang meningkatkan operasional dan pengalaman pelanggan di lingkungan usaha tersebut. Studi literatur juga mencakup penelitian terkini, artikel-artikel, serta temuan terkait yang mendukung penerapan teknologi dalam konteks layanan dan manajemen laundry.

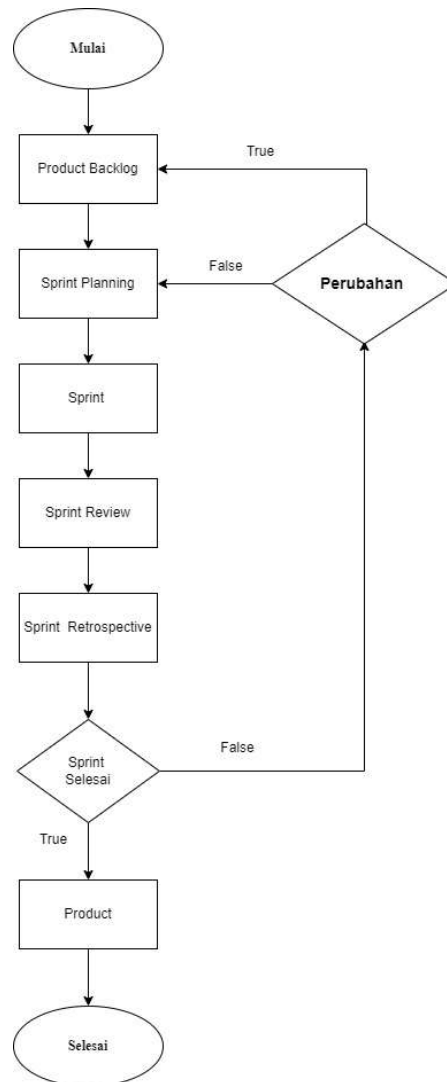
3.3.3 Analisis Kebutuhan

Analisis kebutuhan sistem dilakukan secara mendalam untuk memahami kebutuhan yang spesifik dari Bingo Laundry dalam pengelolaan sistem manajemen mereka. Fokusnya terutama pada aspek fungsi, sehingga pengembang dapat mengidentifikasi dengan jelas fitur-fitur dan fungsi yang penting bagi kebutuhan sistem yang diinginkan. Dapat dilihat pada tabel 3.1 berikut.

Tabel 3. 1 Daftar Kebutuhan Sistem

| Aktor | Deskripsi |
|-----------|--|
| Pegawai | Kebutuhan untuk pegawai menekankan fungsi dalam operasional sehari-hari, termasuk pencatatan pesanan, manajemen pesanan, <i>notifikasi</i> , dan komunikasi dengan pelanggan, serta pencatatan orderan cucian. |
| Pemilik | Pemilik toko lebih terfokus pada aspek manajemen, termasuk manajemen inventaris, rekapan orderan. |
| Pelanggan | Pelanggan berkaitan dengan keinginan mereka untuk menerima <i>notifikasi</i> yang tepat waktu tentang status pesanan mereka. |

3.3.4 Metode Pengembangan dengan *Scrum*



Gambar 3. 2 Diagram Alir metode Scrum

Pada gambar 3.2 menjelaskan bagaimana metode *scrum* bekerja. dengan menampilkan diagram alir *scrum* dengan memiliki keunggulan mencakup yang peningkatan publikasi dan penerapannya yang luas, kemampuan untuk bekerja lebih cepat dan efisien, popularitasnya di industri perangkat lunak karena

sifat praktisnya, fokus pada perencanaan yang kuat dan kolaborasi yang efektif, serta orientasi pada aspek kualitas. Pendekatan yang terintegrasi dan berfokus pada tahapan-tahapan seperti *product backlog*, *sprint planning*, *sprint review*, *sprint retrospect*, dan *sprint product*, yang menjadi bagian penting dalam proses implementasinya.

a. Product Backlog

Dari fitur kebutuhan yang telah direncanakan pada tahapan perancangan Sistem Manajemen Laundry Berbasis Website, dibuatlah *product backlog* yang mencantumkan deskripsi fitur, perkiraan durasi pengerjaan dalam hari, dan detail fitur-fitur tersebut tercantum dalam tabel 3.2 berikut.

Tabel 3. 2 Daftar *Product Backlog*

| No | Fitur | Nilai Prioritas | Perkiraan Waktu (hari) | Keterangan |
|----|---------------------------|-----------------|------------------------|---|
| 1 | <i>Login</i> | 100 | 2 | Fitur <i>Login</i> memungkinkan pengguna (Admin dan Pegawai) untuk mengakses sistem manajemen laundry dengan memasukkan kredensial yang valid. |
| 2 | Dashboard Pegawai & Admin | 100 | 4 | Dasbor untuk pegawai menampilkan daftar tugas harian, status pesanan yang sedang diproses. Sedangkan Dasbor untuk admin menampilkan ringkasan keseluruhan operasional |

| No | Fitur | Nilai Prioritas | Perkiraan Waktu (hari) | Keterangan |
|----|---|-----------------|------------------------|---|
| | | | | dan manajemen pengguna. |
| 3 | Daftar Paket yang tersedia (Pegawai) | 90 | 2 | Halaman ini memungkinkan pegawai untuk melihat daftar paket layanan laundry yang tersedia. |
| 4 | Manajemen Pesanan (Pegawai) | 100 | 3 | Fitur ini memungkinkan pegawai untuk membuat, melihat, mengedit, menghapus, dan memproses pesanan laundry. |
| 5 | Riwayat Transaksi (Pegawai) | 80 | 4 | Fitur ini memungkinkan pegawai untuk melihat semua transaksi, baik yang sedang diproses maupun yang sudah selesai, dalam satu tampilan. |
| 6 | <i>Notifikasi</i> Pelanggan & Invoice (Pegawai) | 80 | 4 | Fitur ini memungkinkan pegawai untuk mencetak invoice untuk pelanggan dan mengirimkan <i>notifikasi</i> melalui <i>Whatsapp</i> ketika orderan sudah selesai. |
| 7 | Pengajuan Barang (Pegawai) | 90 | 3 | Fitur ini memungkinkan pegawai untuk mengajukan permintaan barang seperti deterjen dan perlengkapan lain yang digunakan dalam proses layanan laundry. |

| No | Fitur | Nilai Prioritas | Perkiraan Waktu (hari) | Keterangan |
|-------------------|--|-----------------|------------------------|--|
| 8 | Daftar Akun Pegawai (Admin) | 100 | 2 | Memungkinkan admin untuk melihat, menambah, mengedit, dan menghapus akun pegawai yang memiliki akses ke sistem manajemen laundry. |
| 9 | Manajemen Paket yang tersedia (Admin) | 90 | 2 | Fitur ini memungkinkan admin untuk mengelola daftar paket layanan laundry yang tersedia, termasuk menambah, mengedit, menghapus, dan mengatur harga. |
| 10 | Rekapan Order (Admin) | 80 | 2 | Fitur ini memungkinkan admin untuk melihat rekapan jumlah pelanggan yang menggunakan berbagai jenis layanan seperti paket express, reguler, dan setrika. |
| 11 | Monitoring Pesanan & Pengajuan (Admin) | 90 | 4 | Memungkinkan admin untuk memonitor dan menindaklanjuti pesanan serta pengajuan yang diajukan oleh pegawai terkait kebutuhan operasional laundry. |
| Total hari | | | 32 | |

Keterangan:

1. Nilai prioritas berkisar antara 1 hingga 100, dengan 100 menunjukkan tingkat kepentingan tertinggi.
2. Nilai dari satu story point setara dengan nilai dari satu *man-day*, yang merupakan satu hari kerja oleh satu orang.

Pada tahapan *product backlog* di awal menghasilkan kesepakatan dengan *product owner* mengenai sistem manajemen laundry dan perancangan sistem telah ditetapkan. Terdapat 11 total kebutuhan sistem yang telah dirancang, yang kemudian disusun dalam *product backlog*.

b. Sprint

Pembagian sprint dari product backlog tersebut dilakukan berdasarkan prioritas dan kompleksitas fitur-fitur yang tercantum. Dalam pengembangan sistem kali ini, terbagi menjadi 4 sprint yang bertujuan untuk menyusun dan menyelesaikan fitur-fitur yang paling penting terlebih dahulu. Hal ini memastikan fokus pada fitur utama dan memberikan kemungkinan untuk penyelesaian bertahap dengan urutan prioritas yang telah ditetapkan.

Sprint 1 :

Pada *Sprint 1* ini, menjelaskan tentang daftar tugas yang akan dikerjakan berdasarkan daftar *product backlog* yang telah disusun. Berikut merupakan rincian *Sprint 1* :

Sprint Planning

Pada perancangan *Sprint 1*, dihasilkan keputusan sebagai berikut:

- a) Durasi *sprint*: 2 minggu (10 hari efektif)
- b) Langkah pertama untuk menentukan *story point* yang telah dimasukkan ke dalam *sprint* adalah dengan melakukan estimasi terhadap kecepatan tim. Berikut adalah perhitungan kecepatan tim:

Man-days : 1 (orang) x 10 (hari efektif) = 10

Focus Factor : 80%

Perkiraan kecepatan tim = *Mans-days* yang tersedia x

Focus Factor

= 10 x 80%

= 8

Dengan perkiraan kecepatan tim, maka terdapat 8 *story point* dari *product backlog* yang dimasukkan pada *Sprint* ini. *Sprint Backlog* pada *Sprint* 1 dijelaskan pada tabel 3.3 dibawah ini:

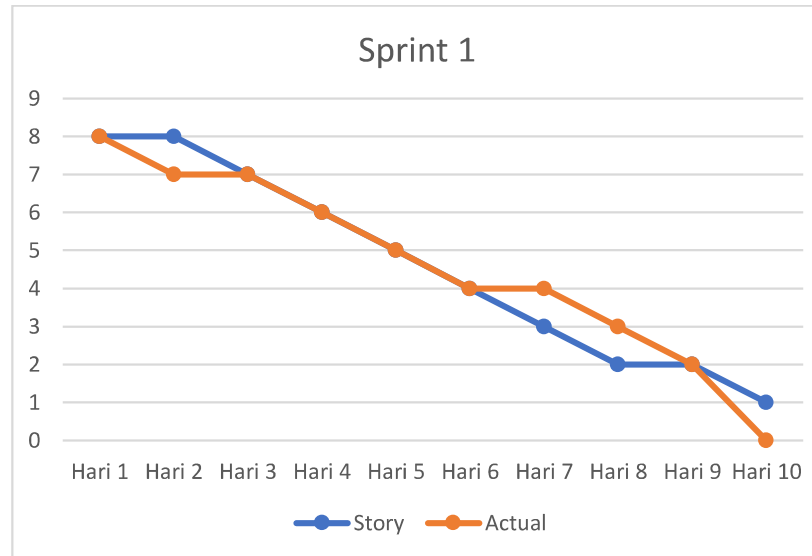
Tabel 3. 3 *Sprint Backlog Sprint 1*

| <i>Sprint Backlog</i> | Pengerjaan | <i>Story Point</i> |
|---------------------------|--|--------------------|
| Login | <ul style="list-style-type: none"> • Merancanng database Login • Buat endpoint API di Express.js untuk menerima permintaan registrasi untuk fitur halaman Login sebagai admin / pegawai • Lakukan pengujian unit terhadap endpoint API registrasi | 2 |
| Dashboard Pegawai & Admin | <ul style="list-style-type: none"> • Desain UI/UX untuk dashboard Pegawai & Admin | 4 |

| | | |
|--------------------------------------|--|---|
| | <ul style="list-style-type: none"> • Validasi token akses dan peran pengguna untuk mengakses dashboard • Membuat <i>Rest-API</i> registrasi untuk fitur halaman daftar • Pengujian unit testing untuk memastikan middleware berfungsi dengan baik | |
| Daftar Paket yang tersedia (Pegawai) | <ul style="list-style-type: none"> • Desain tampilan halaman untuk daftar paket yang tersedia • Buat endpoint API di backend untuk mengambil data paket • Kirim data paket ke frontend untuk ditampilkan | 2 |
| Total | | 8 |

Daily Scrum

Pada tahapan ini, Daily Scrum digunakan untuk memonitor kemajuan setiap pekerjaan yang telah diselesaikan oleh pengembang. Pertemuan ini dilakukan setiap hari selama 15 menit, dengan tujuan untuk melacak perkembangan terhadap *sprint backlog*. Selama *sprint* 1, grafik *Burndown* telah digunakan untuk memvisualisasikan estimasi sisa pekerjaan berbanding dengan waktu yang tersisa, yang telah diperbarui secara teratur untuk memantau kemajuan selama periode *sprint* tersebut. Dapat dilihat pada gambar 3.3 berikut.



Gambar 3.3 Grafik *Daily Sprint 1*

Sprint Review

Dalam pertemuan evaluasi *sprint*, dilakukan penilaian terhadap kinerja pengembang selama *sprint* 1. Perkiraan durasi untuk *sprint* 1 terbukti akurat karena semua *sprint backlog* telah berhasil diselesaikan oleh pengembang. Kinerja yang stabil tercermin dari grafik *burndown*. Pada hari ketiga hingga keenam, jumlah *story point* aktual sesuai dengan perkiraan. Namun, pada hari ketujuh dan kedelapan, terjadi revisi yang mengembalikan *story point* menjadi 4 dan 3. Pada hari kesembilan, *story point* turun menjadi 2, dan akhirnya pada hari kesepuluh mencapai 0, menandakan semua tugas telah selesai.

Sprint Retrospective

Pada pertemuan evaluasi *sprint*, dilaksanakan penilaian terhadap performa pengembang selama

sprint 1. Perkiraan durasi terbukti akurat, karena semua item dalam *sprint backlog* berhasil diselesaikan. Kinerja yang stabil terlihat dari grafik *burndown*, yang menunjukkan pengurangan 1 *story point* setiap hari.

Sprint 2 :

Pada *sprint 2* ini akan dijelaskan daftar pekerjaan yang telah dikerjakan berdasarkan daftar *product backlog* yang telah di buat. Berikut adalah pada *sprint 2* :

Sprint Planning

Pada perancangan *sprint 2*, dihasilkan keputusan yang telah dikerjakan sebagai berikut:

- a) Durasi *sprint*: 2 minggu (11 hari efektif)
- b) Langkah pertama untuk menentukan *story point* yang akan dimasukkan ke dalam *sprint* adalah dengan melakukan estimasi terhadap kecepatan tim. Berikut adalah perhitungan kecepatan tim:

Man-days : 1 (orang) x 11 (hari efektif) = 11

Focus Factor : 80%

Perkiraan kecepataannya = *Mans-days* yang tersedia x
Focus Factor

= 11 x 80%

= 9

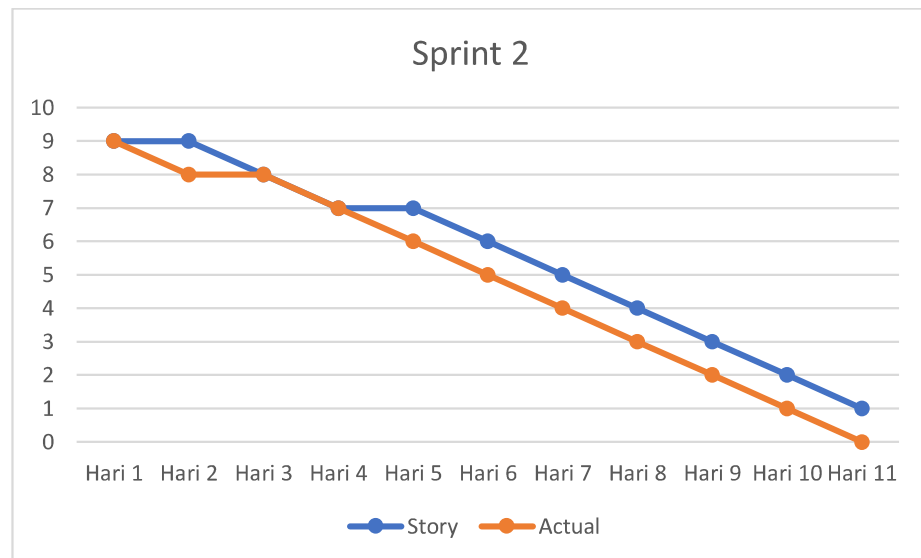
Dengan perkiraan durasi tersebut, terdapat 9 *story point* dari *product backlog* yang akan dimasukkan ke dalam *Sprint* ini. Berikut adalah *sprint backlog* untuk *sprint 2* yang dijelaskan pada tabel 3.4 dibawah ini:

Tabel 3. 4 *Sprint Backlog Sprint 2*

| <i>Sprint Backlog</i> | Pengerjaan | <i>Story Point</i> |
|---------------------------------------|--|--------------------|
| Manajemen Pesanan | <ul style="list-style-type: none"> • Desain tampilan halaman manajemen pesanan dengan membuat form sesuai paket yang tersedia. • Buat <i>Endpoint API</i> di Express.js untuk menyediakan data pesanan. • Lakukan pengujian unit terhadap komponen Pesanan ini pastikan sudah berfungsi | 4 |
| Riwayat Transaksi | <ul style="list-style-type: none"> • Desain UI/UX untuk dashboard Pegawai & Admin • Validasi token akses dan peran pengguna untuk mengakses dashboard • Membuat <i>Endpoint API</i> registrasi untuk fitur halaman daftar • Pengujian unit testing untuk memastikan middleware berfungsi dengan baik | 2 |
| <i>Notifikasi</i> Pelanggan & Invoice | <ul style="list-style-type: none"> • Desain tampilan halaman untuk daftar paket yang tersedia • Buat endpoint API di backend untuk mengambil data paket • Kirim data paket ke frontend untuk ditampilkan | 3 |
| Total | | 9 |

Daily Scrum

Pada tahapan ini, *daily scrum* digunakan untuk memonitor kemajuan setiap pekerjaan yang telah diselesaikan oleh pengembang. Pertemuan ini dilakukan setiap hari selama 15 menit, dengan tujuan untuk melacak perkembangan terhadap *product backlog*. Selama *sprint 2*, grafik *burndown* telah digunakan untuk memvisualisasikan estimasi sisa pekerjaan berbanding dengan waktu yang tersisa, yang telah diperbarui secara teratur untuk memantau kemajuan selama periode *sprint* tersebut.



Gambar 3. 4 Grafik *Daily Sprint 2*

Sprint Review

Dalam evaluasi *sprint 2*, semua item dalam *product backlog* diselesaikan sesuai ekspektasi. Hasil *sprint 2* akan diuraikan lebih lanjut, sementara *product backlog* yang tersisa tidak memerlukan revisi.

Berdasarkan grafik, *story point* pada hari pertama tetap diangka 9, dan tidak berkurang hingga hari kedua. Pada hari ketiga, *story point* berkurang menjadi 8 setelah revisi. Mulai hari keempat hingga hari kesebelas, *story point* berkurang secara konsisten sebesar 1 setiap hari.

Sprint Retrospective

Dalam evaluasi *sprint*, kinerja selama *sprint 2* dievaluasi. Perkiraan durasi untuk *sprint 2* terbukti akurat karena semua item dalam *sprint backlog* berhasil diselesaikan oleh pengembang. Kinerja terlihat stabil, seperti yang tercermin dalam grafik *burndown* yang memperlihatkan bahwa terjadi pengurangan sekitar 1 *story point* setiap harinya. Untuk *sprint* berikutnya, diharapkan pengembang bisa mempertahankan kestabilan dalam pekerjaan mereka dan menyelesaikan semua *sprint backlog* yang ditetapkan.

Sprint 3 :

Pada *sprint 3* ini akan dijelaskan daftar pekerjaan yang telah dikerjakan berdasarkan daftar *product backlog* yang telah di buat. Berikut adalah pada *sprint 3 :*

Sprint Planning

Pada perancangan *sprint 3*, dihasilkan keputusan sebagai berikut:

- a) Durasi *sprint*: 2 minggu (11 hari efektif)

b) Langkah pertama untuk menentukan *story point* yang akan dimasukkan ke dalam *sprint* adalah dengan melakukan estimasi terhadap kecepatan tim. Berikut adalah perhitungan kecepatan tim:

Man-days : 1 (orang) x 11 (hari efektif) = 11

Focus Factor : 80%

Perkiraan kecepatannya = *Mans-days* yang tersedia x
Focus Factor

= 11 x 80%

= 9

Dengan perkiraan durasi tersebut, sebanyak 9 *story point* dari *product backlog* telah dimasukkan ke dalam *sprint* ini. *sprint backlog* untuk *sprint* 3 diuraikan pada tabel 3.5 dibawah ini:

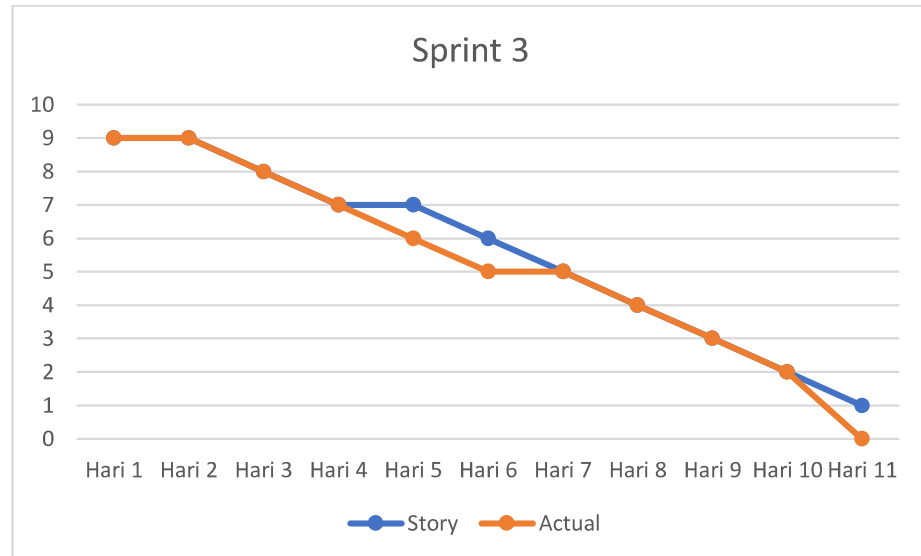
Tabel 3. 5 *Sprint Backlog Sprint 3*

| <i>Sprint Backlog</i> | Pengerjaan | <i>Story Point</i> |
|-----------------------|--|--------------------|
| Pengajuan Barang | <ul style="list-style-type: none"> • Membuat komponen dan menentukan tata letak form pengajuan sesuai UI/UX yang diperlukan. • Menambahkan validasi input dan fitur interaksi antarmuka pengguna implementasikan CRUD. • Melakukan pengujian unit dan integrasi untuk memastikan fungsionalitas | 4 |
| Daftar Akun Pegawai | <ul style="list-style-type: none"> • Membuat komponen dan menentukan tata letak form pengajuan sesuai UI/UX yang diperlukan. | 2 |

| <i>Sprint Backlog</i> | Pengerjaan | <i>Story Point</i> |
|-------------------------------|--|--------------------|
| | <ul style="list-style-type: none"> • Menyediakan <i>Endpoint API</i> di Express.js operasi CRUD akun pegawai. • Logika backend untuk validasi dan penyimpanan data akun pegawai dan melakukan testing memastikan fungsionalitas. | |
| Manajemen Paket yang tersedia | <ul style="list-style-type: none"> • Desain tampilan halaman untuk daftar paket yang tersedia • Buat endpoint API di backend untuk mengambil data paket • Kirim data paket ke frontend untuk ditampilkan | 3 |
| Total | | 9 |

Daily Scrum

Pada tahapan ini, *Daily Scrum* digunakan untuk memonitor kemajuan setiap pekerjaan yang telah diselesaikan oleh pengembang. Pertemuan ini dilakukan setiap hari selama 15 menit, dengan tujuan untuk melacak perkembangan terhadap *product backlog*. Selama *sprint* 3, grafik *burndown* telah digunakan untuk memvisualisasikan estimasi sisa pekerjaan berbanding dengan waktu yang tersisa, yang telah diperbarui secara teratur untuk memantau kemajuan selama periode *sprint* tersebut.



Gambar 3. 5 Grafik *Daily Sprint 3*

Sprint Review

Dalam evaluasi *sprint 3*, semua item dalam *product backlog* dinyatakan selesai sesuai dengan ekspektasi yang ditetapkan. Hasil dari *sprint 3* akan diuraikan lebih lanjut pada bagian berikutnya. Pada *sprint 3*, hari pertama dan kedua mengalami kendala sehingga *story point* tetap diangka 9. Namun, pada hari ketiga, pengerjaan mulai membaik, dan beberapa revisi dilakukan pada hari kelima, menyebabkan *story point* naik kembali menjadi 7. Mulai hari keenam dan seterusnya, proses antara *actual* dan *story point* berjalan linear hingga di hari terakhir tidak ada *product backlog* yang tersisa.

Sprint Retrospective

Pada evaluasi *sprint*, pengerjaan selama *sprint 3* dievaluasi. Perkiraan kecepatan untuk *sprint 3* terbukti akurat karena semua item dalam *product backlog*

berhasil diselesaikan oleh pengembang. Kinerja terlihat stabil, seperti yang tercermin dalam grafik *burndown* yang memperlihatkan bahwa pengurangan sekitar 1 *story point* setiap harinya. Untuk *sprint* berikutnya, diharapkan pengembang bisa mempertahankan kestabilan dalam kinerja mereka dan menyelesaikan semua *sprint backlog* yang ditetapkan.

Sprint 4 :

Pada *sprint 4* ini akan dijelaskan daftar pekerjaan yang akan dikerjakan berdasarkan daftar *product backlog* yang telah di buat. Berikut adalah pada *sprint 4* :

Sprint Planning

Pada perancangan *sprint 4*, dihasilkan keputusan sebagai berikut:

- a) Durasi *sprint*: 1 minggu (6 hari efektif)
- b) Langkah pertama untuk menentukan *story point* yang akan dimasukkan ke dalam *sprint* adalah dengan melakukan estimasi terhadap kecepatan tim. Berikut adalah perhitungan kecepatan tim:

Man-days : 1 (orang) x 6 (hari efektif) = 6

Focus Factor : 80%

Perkiraan kecepatannya = *Mans-days* yang tersedia x

Focus Factor

= 6 x 80%

= 5

Perkiraan durasi maka *product backlog* terdapat 9 *story point* yang dimasukkan pada *sprint*. Berikut

sprint backlog pada *sprint 4* dijelaskan pada tabel 3.6 dibawah ini:

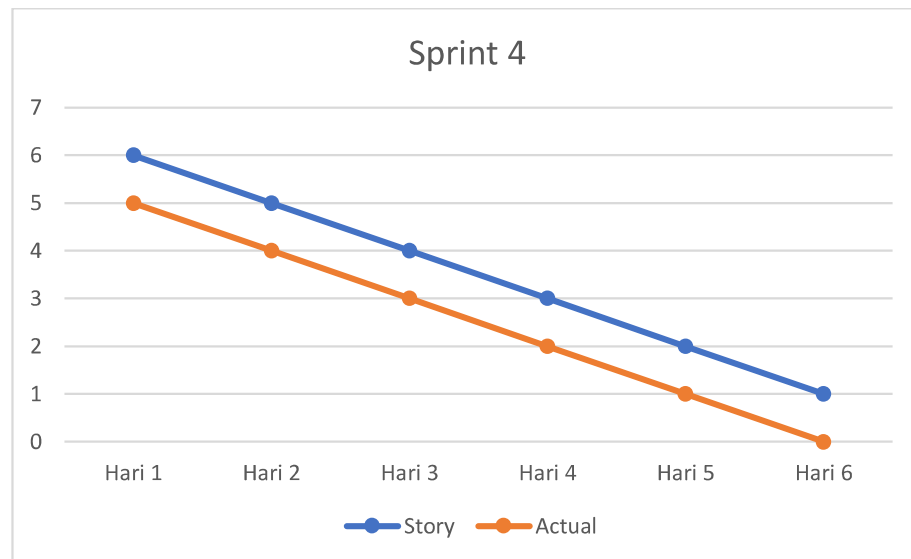
Tabel 3. 6 *Sprint Backlog Sprint 4*

| <i>Sprint Backlog</i> | Pengerjaan | <i>Story Point</i> |
|--------------------------------|---|--------------------|
| Rekapan Order | <ul style="list-style-type: none"> • Membuat komponen dan menentukan tata letak form pengajuan sesuai UI/UX yang diperlukan. • Implementasi fitur untuk menampilkan detail order dengan chart. • Menyediakan endpoint API di Express.js dengan cara mendapatkan rekapan order dengan method GET. | 2 |
| Monitoring Pesanan & Pengajuan | <ul style="list-style-type: none"> • Membuat komponen dan menentukan tata letak form pengajuan sesuai UI/UX yang diperlukan. • Logika backend untuk mengambil data dari database dan menyediakannya ke frontend. • Pengujian unit dan integrasi untuk memastikan fungsionalitas. | 3 |
| Total | | 5 |

Daily Scrum

Pada tahapan ini, *daily scrum* digunakan untuk memonitor kemajuan setiap pekerjaan yang telah diselesaikan oleh pengembang. Pertemuan ini dilakukan setiap hari selama 15 menit, dengan tujuan untuk melacak perkembangan terhadap *product backlog*. Selama *sprint 4*, grafik *burndown* akan digunakan untuk memvisualisasikan estimasi sisa

pekerjaan berbanding dengan waktu yang tersisa, yang akan diperbarui secara teratur untuk memantau kemajuan selama periode *sprint* tersebut.



Gambar 3. 6 Grafik *Daily Sprint 4*

Sprint Review

Dalam evaluasi *sprint 4*, semua bekerja secara efisien dan sesuai dengan harapan yang telah ditetapkan. Bahwa semua item dalam *product backlog* telah diselesaikan sesuai ekspektasi. Hasil dari *sprint* ini akan diperjelas pada bagian berikutnya dari pertemuan tersebut. Selain itu, dalam pertemuan juga dibahas mengenai *product backlog* yang masih tersisa, dan evaluasi menunjukkan bahwa tidak ada revisi yang diperlukan terhadap bagian tersebut. Setiap hari, dari hari pertama hingga hari keenam, *story point* dari *product backlog* berkurang satu tanpa ada kendala yang signifikan.

Sprint Retrospective

Dalam evaluasi *sprint*, kinerja selama *sprint* 4 dievaluasi. Perkiraan kecepatan untuk *sprint* 4 terbukti akurat karena semua item dalam *product backlog* berhasil diselesaikan oleh pengembang. Kinerja terlihat stabil, seperti yang tercermin dalam grafik *burndown* yang memperlihatkan bahwa pengurangan sekitar 1 *story point* setiap harinya. Untuk *sprint* ini, diharapkan pengembang dapat mempertahankan stabilitas kinerja mereka dan menyelesaikan semua *sprint backlog* yang telah ditetapkan.

3.3.5 Pengujian Sistem

a. Black Box Testing

Black Box testing memberikan sejumlah keuntungan yang signifikan. Pertama-tama, pendekatan ini mengharuskan penggunaan yang lebih sedikit cakupannya dalam hal sumberdaya jika dibandingkan dengan pengujian *White Box*, karena perhatian utamanya terfokus pada fungsi-fungsi sistem tanpa memerlukan pemahaman mendalam tentang implementasi internal. Hal ini memungkinkan tim pengujian untuk bekerja secara lebih efisien. Selain itu, keunggulan lainnya terletak pada potensi otomatisasi pengujian *Black Box*, yang dapat meningkatkan efektivitas penggunaan sumber daya dan mempersingkat periode pengujian secara signifikan.



Gambar 3. 7 *Black Box Testing*

Pada gambar 3.3 menunjukkan pengujian dalam proses melibatkan adanya inputan yang berasal dari gambaran umum sistem, dilanjutkan dengan pengujian, dan evaluasi hasilnya sebagai output. bertujuan untuk mencapai cakupan uji yang komprehensif, melibatkan berbagai kelompok uji seperti ketersediaan, keandalan, ketahanan beban, dan kelompok uji terkait operasi, revisi, dan transisi. Pengujian ini bertujuan memastikan aplikasi berkinerja optimal dalam berbagai kondisi operasional. Pendekatan *Black Box* tidak hanya mengidentifikasi kesalahan fungsional tetapi juga mendukung pengujian aspek non-fungsional, seperti kinerja dan ketahanan sistem. Ini adalah pilihan yang solid untuk memvalidasi sistem secara menyeluruh tanpa memerlukan pengetahuan mendalam tentang struktur internal[31].