

Pengukuran *Focus Factor* Tim Scrum pada Proyek Perangkat Lunak Skala Kecil

Ahlijati Nuraminah

Program Studi Ilmu Komputer, Universitas Ary Ginanjar
Menara 165, Jl. TB Simatupang Kav 1, Jakarta
Email: ahlijati.nuraminah@uag.ac.id
No Hp 081586176565

ABSTRAK

Scrum, metodologi *Agile* yang umum digunakan dalam pengembangan perangkat lunak, menekankan kerjasama, adaptabilitas, dan pengiriman produk bernilai tinggi. Meskipun *velocity* umumnya digunakan untuk mengukur kinerja tim, namun tidak memberikan gambaran yang komprehensif karena beberapa faktor seperti jumlah tim *developer* dan hari kerja yang diperhitungkan. Untuk mengatasi hal ini, *Focus Factor* digunakan untuk memberikan evaluasi yang lebih terhadap fokus dan konsentrasi tim selama *Sprint*. Penelitian ini menganalisis kinerja tim Scrum menggunakan *Focus Factor* dalam proyek perangkat lunak skala kecil, dengan tujuan memahami keuntungannya dalam meningkatkan efisiensi pengembangan perangkat lunak. Metode yang digunakan yaitu inisiasi *Scrum*, pengumpulan data, perhitungan *Focus Factor*, analisis, dan evaluasi. Hasil dari tujuh *Sprint* menunjukkan kinerja tim yang bervariasi, dengan beberapa mencapai nilai *Focus Factor* tinggi, mencerminkan fokus yang efisien, sementara yang lain menunjukkan kinerja lebih rendah. Secara keseluruhan, *Focus Factor* membuktikan adanya manfaat dalam menilai kinerja tim, memberikan wawasan tentang area-area yang dapat ditingkatkan. Penelitian mendatang dapat mengeksplorasi penerapan *Focus Factor* dalam proyek-proyek lebih besar untuk mengevaluasi kemampuan tim dalam mempertahankan fokus dan produktivitas yang tinggi pada *user stories* yang lebih kompleks.

Kata kunci: scrum, focus factor, velocity.

ABSTRACT

Scrum, an *Agile* methodology widely employed in software development, emphasizes collaboration, adaptability, and delivering high-value products. While *Velocity* is commonly used to measure team performance, it may not provide a comprehensive picture due to factors like the number of developers and working days considered. To address this, the *Focus Factor* is introduced, accounting for these factors and providing a more nuanced evaluation of team focus and concentration during each *Sprint*. This research analyzes a *Scrum Team's* performance using the *Focus Factor* in a small-scale software project, aiming to understand its advantages in enhancing software development efficiency. The method involves *Scrum* initiation, data collection, *Focus Factor* calculation, analysis, and evaluation. The results from seven *Sprints* indicate varying team performances, with some achieving high *Focus Factor* values, reflecting efficient focus, while others exhibit lower performance. In conclusion, the *Focus Factor* proves valuable in assessing team performance, offering insights into areas of improvement. Future research can explore its application in larger projects to evaluate teams' ability to maintain high focus and productivity amidst more extensive *user stories*.

Keywords: scrum, focus factor, velocity.

Pendahuluan

Scrum merupakan metodologi yang cukup populer diimplementasikan pada proyek pembangunan perangkat lunak (Billah & Nuraminah, 2022). Scrum didasarkan pada prinsip-prinsip Agile yang fokus pada kolaborasi tim, adaptabilitas terhadap perubahan, dan pengiriman produk yang bernilai tinggi (Sutherland, 2020). Pengembangan sistem dengan metode Scrum salah satunya dimanfaatkan dalam pembuatan aplikasi penjualan dengan memanfaatkan QRCode (Abidin, 2023). Salah satu cara untuk mengukur kinerja tim adalah dengan menghitung nilai *Velocity* (Putri & Fatichah, 2023). *Velocity* adalah jumlah pekerjaan yang selesai oleh tim dalam satu sprint (Vaid & Ghose, 2020). *Velocity* diperkirakan berdasarkan pengalaman tim dari sprint sebelumnya (Park & Noh, 2023), kemudian digunakan untuk membantu tim dalam perencanaan sprint berikutnya. Namun nilai *velocity* saja tidak mampu memberikan gambaran yang cukup untuk menilai kinerja tim karena ada faktor jumlah tim pengembang dan hari produktif yang dipertimbangkan (Park & Noh, 2023; Raicevic et al., 2023).

Untuk memberikan gambaran yang lebih lengkap mengenai kinerja tim, maka diperlukan perhitungan lainnya yang mempertimbangkan faktor jumlah tim pengembang dan hari produktif dalam setiap *Sprint*, yaitu dengan menghitung *Focus Factor* tim (Verwijs & Russo, 2023). *Focus Factor* pada *scrum* adalah pengukuran tingkat fokus atau konsentrasi tim dalam mengerjakan setiap *Sprint* yang telah ditentukan (Almeida & Carneiro, 2023). *Focus Factor* dihitung berdasarkan jumlah item *backlog* yang berhasil diselesaikan selama *Sprint* dan jumlah item yang seharusnya diselesaikan (Billah & Nuraminah, 2022). Nilai *Focus Factor* tersebut dapat menjadi estimasi dalam mengerjakan di setiap *sprint* berikutnya. Dengan menggunakan *Focus Factor* pada *scrum*, tim dapat lebih terfokus dan efisien dalam mengerjakan setiap *Sprint*, meningkatkan keefektifan dan efisiensi kerja, dan menghasilkan produk yang lebih baik (Sihombing & Ferdiando, 2023).

Analisis terhadap *Focus Factor* telah dilakukan dalam penelitian terdahulu oleh Charkie (2020) yang menghitung *focus factor* and *velocity increase* pada 8 proyek *open source*. Hasil ditemukan bahwa secara umum jumlah *developer* dan durasi sprint tidak memberikan hasil statistik yang signifikan terhadap *Focus Factor* tim. Namun jika perhitungan ini diterapkan untuk *commercial project* bukan *open source*, kemungkinan memiliki perbedaan hasil karena faktor lainnya. Penelitian lainnya oleh Harahap (2023) menganalisis produktivitas tim scrum dalam pengembangan aplikasi Digital MSME *Empowerment*. Dalam setiap tahapan Sprint dihitung *Focus Factor* dari tim untuk melakukan estimasi terhadap durasi Sprint berikutnya. Namun jumlah *backlog* yang terbatas dan jumlah Sprint yang hanya 2 kali tidak dapat memberikan gambaran yang lebih lengkap terhadap produktifitas tim.

Berdasarkan latar belakang tersebut maka tujuan dari penelitian ini adalah untuk menganalisis kinerja Tim Scrum dengan teknik *Focus Factor* pada proyek perangkat lunak skala kecil untuk dapat melihat kelebihan metode *Focus Factor* dalam meningkatkan efisiensi pengembangan perangkat lunak

Metode Penelitian

Penelitian dilakukan dengan langkah awal yaitu Inisiasi Scrum. Pada tahap ini, project dimulai dengan pembagian *role* dalam tim, penentuan *product backlog* serta estimasi *story point* setiap *backlog*. Hasil dari tahap ini adalah estimasi durasi *project*, rencana jumlah Sprint yang akan dilaksanakan, durasi setiap Sprint, dan estimasi

story point. Kemudian selama *project* berlangsung, dilakukan pengumpulan data berupa *story point* yang berhasil diselesaikan dalam setiap *sprint*, durasi penyelesaian setiap backlog, dan jumlah hari kerja setiap *developer*. Kemudian dilakukan perhitungan *velocity* tim, dan analisis *sprint burndown chart*. Dari data *story point* setiap *sprint*, kemudian dilakukan penghitung *Focus Factor* dengan membagi jumlah pekerjaan yang selesai dengan jumlah hari pekerjaan diselesaikan dikali jumlah tim *developer*. Kemudian dari data *focus factor* setiap *Sprint* yang sudah diperoleh, kemudian dianalisis untuk mengidentifikasi tren atau pola yang muncul. Hasil perhitungan *focus factor* ini dijadikan masukan bagi tim untuk estimasi *sprint* berikutnya. Pada tahap akhir dilakukan evaluasi terhadap kinerja tim, serta analisis rekomendasi untuk perbaikan.

Hasil dan Pembahasan

Inisiasi Scrum dilakukan dengan pembagian peran. *Product Owner* diwakili 1 perwakilan dari pemangku kepentingan bisnis. *Scrum Master* terdiri dari 1 orang di mana ia bertanggung jawab memastikan tim menerapkan dan menjalankan praktik Scrum dengan benar. Tim pengembang terdiri dari 3 orang di mana individu tersebut bertanggung jawab menerjemahkan *backlog* produk menjadi inkremen yang siap untuk dirilis.

Penyusunan *Product Backlog* dilakukan dengan cara planning poker (Pozenel, et.al, 2023). *Story Point* ditentukan dengan *cara sizing relatives*. Metode ini melibatkan perbandingan dua atau lebih *user stories* dalam hal kompleksitas (Mallidi & Sharma, 2021), bukan dalam hal waktu yang dibutuhkan. Daftar *produk backlog* dapat dilihat pada Tabel 1.

Tabel 1. *Product backlog*

No	Backlog	Story Point
1	menginput pesanan	8
2	melihat daftar pesanan	4
3	mengubah pesanan	6
4	membatalkan pesanan	4
5	menghapus pesanan	2
6	mencetak <i>invoice</i> pesanan (pdf)	8
7	melakukan pembayaran pesanan	4
8	melihat data item pesanan	4
9	menambah item pesanan	4
10	mengubah item pesanan	4
11	menghapus item pesanan	2
12	melihat data kategori	2
13	menambah kategori	2
14	mengubah kategori	2
15	menghapus kategori	2
16	melihat data pelanggan	2
17	menambah data pelanggan	4
18	mengubah data pelanggan	4
19	menghapus data pelanggan	2
20	melihat transaksi (pemasukan/pengeluaran)	2
21	menambah transaksi (pemasukan/pengeluaran)	4
22	mengubah transaksi (pemasukan/pengeluaran)	4
23	menghapus transaksi (pemasukan/pengeluaran)	2

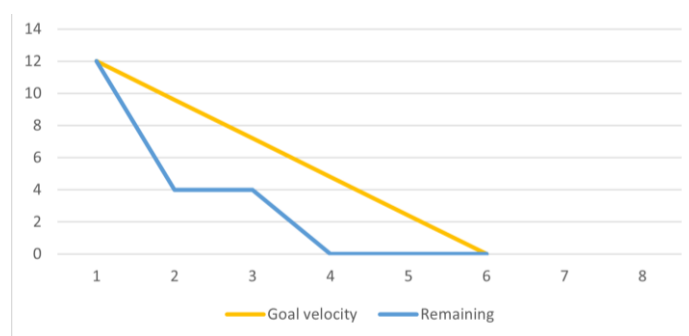
No	Backlog	Story Point
24	melihat laporan pemasukan/pengeluaran (harian)	12
25	melihat laporan pemasukan/pengeluaran (bulanan)	12
26	melihat laporan pemasukan/pengeluaran (tahunan)	12
27	melihat <i>dashboard</i> pemasukan/pengeluaran	12
28	melihat laporan laba rugi (bulanan, tahunan)	12
29	grafik pemasukan pengeluaran akumulatif setahun	20
30	grafik pemasukan pengeluaran (grafik pie)	20
31	Login, logout	4
32	Pasang Template	2

Sprint merupakan sebuah batasan waktu (*timebox*) yang berisi periode kerja yang sudah ditentukan dan setiap sprint fokus terhadap *delivery* produk berdasarkan item – item yang dipilih dari *product backlog* yang akan dilaksanakan dengan batasan waktu yang telah ditentukan (Rizaldi et al., 2022). Setiap Sprint diawali dengan *Sprint Planning*, di mana tim memilih item dari *Product Backlog* untuk dimasukkan ke dalam *Sprint Backlog* (Tahyudin & Zidni Iman Sholihati, 2022) . Untuk seluruh *product backlogs*, proyek berlangsung dari 8 Februari hingga 8 Mei 2023 yang dibagi menjadi 12 Sprint. Setiap Sprint akan berdurasi 7 hari kerja. Daftar *sprint backlog* yang dikerjakan pada Sprint 1 dapat dilihat pada Tabel 2.

Tabel 2. Sprint 1 backlog

No	Backlog	Story point	Tanggal selesai
1	Tambah pesanan	8	09-02-2023
2	Lihat pesanan	4	13-02-2023

Sprint 1 memiliki 2 *backlog item* dengan 12 *Story Point* dengan durasi pengerjaan 6 hari kerja. Dari data yang ditunjukkan pada Gambar 1, **Error! Reference source not found.** garis *remaining* berada di bawah garis *goal velocity* hingga akhir yang menunjukkan bahwa *progress* pengerjaan berjalan dengan baik dan lancar tanpa ada kendala sehingga pengerjaan *backlog* dapat diselesaikan lebih cepat pada hari ke 4. Grafik *burndown* Sprint 1 dapat dilihat pada Gambar 1.



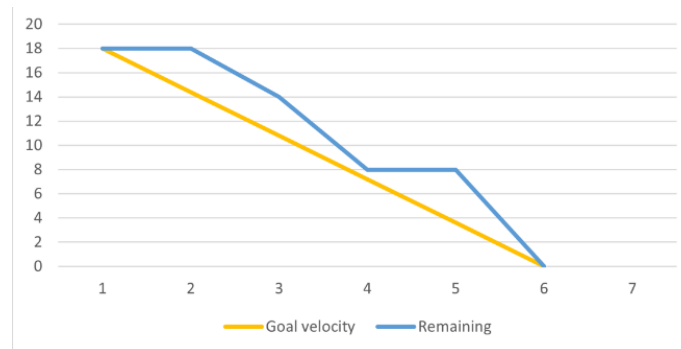
Gambar 1. Burndown chart sprint 1

Sprint ke-2 tim *scrum* menentukan 6 *backlog* yang dikerjakan dengan total 18 *Story Point* dengan *rincian* yang dapat dilihat pada Tabel 3.

Tabel 3. Sprint 2 backlog

No	Backlog	Story Point	Tanggal Selesai
1	Pasang Template	2	20-02-2023
2	Ubah Pesanan	6	22-02-2023
3	Hapus Pesanan	2	20-02-2023
4	Bayar Pesanan	4	23-02-2023
5	<i>Bug fixing</i> , Tambah pesanan pelanggan	2	23-02-2023
6	<i>Bug fixing</i> , Tambah pesanan jumlah	2	23-02-2023

Sprint 2 memiliki 18 *Story Point* dengan durasi pengerjaan 6 hari kerja. Dari data yang ditunjukkan pada Gambar 1, garis *remaining* berada diatas garis *goal velocity*, ini menandakan bahwa kinerja tim tidak berjalan dengan baik pada hari ke 4 hingga ke 5 garis *remaining* berbentuk lurus yang berarti tim tidak memiliki *progress* sama sekali, walaupun begitu semua pada akhirnya semua *sprint backlog* yang dipilih telah selesai dikerjakan. Grafik *burndown* Sprint 2 dapat dilihat pada Gambar 2.



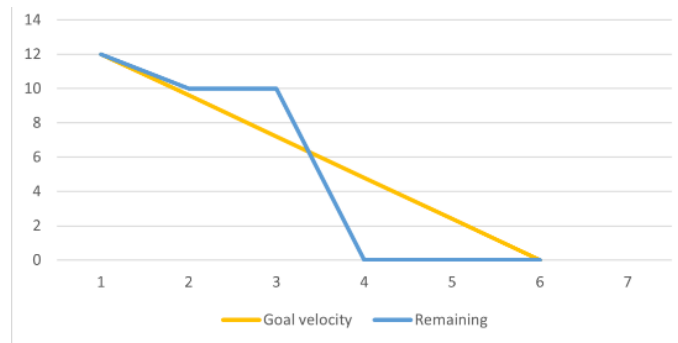
Gambar 2. Burndown chart sprint 2

Sprint ke-3 ditetapkan ada 4 *sprint backlog* dengan total 12 *Story Point* dapat dilihat dengan rincian pada Tabel 4.

Tabel 4. Sprint 3 backlog

No	Backlog	Story point	Tanggal selesai
1	Tambah transaksi	4	28-02-2023
2	Lihat transaksi	2	26-02-2023
3	Ubah transaksi	4	28-02-2023
4	Hapus transaksi	2	28-02-2023
5	<i>Bug fixing</i> bayar pesanan	2	26-02-2023

Sprint 3 memiliki 14 *Story Point* dengan durasi pengerjaan 5 hari kerja . Dari data yang ditunjukkan pada Gambar 2, pada hari ke 1 - 2 garis *remaining* berada sedikit di atas garis *goal velocity* di mana tim telah menyelesaikan beberapa *sprint backlog* untuk diselesaikan, pada hari ke 2 - 3 garis *remaining* tidak mengalami penurunan di mana tim memasuki hari libur sehingga tidak ada pengerjaan *sprint backlog* pada hari tersebut, dari hari ke 3 - 4 garis *remaining* turun drastis sehingga berada di bawah garis *goal velocity* yang menandakan tim telah menyelesaikan semua *sprint backlog* lebih awal yaitu pada hari ke 4. Grafik *burndown* Sprint 3 dapat dilihat pada Gambar 3.



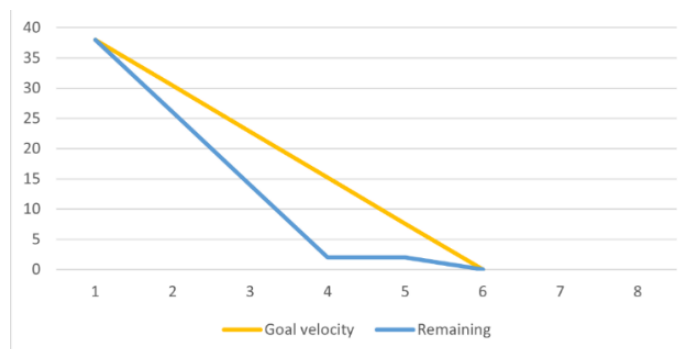
Gambar 3. Burndown chart sprint 3

Pada *sprint 4* ada 4 *backlog* dengan 38 *Story Point* dengan rincian yang dapat dilihat pada Tabel 5.

Tabel 5. Sprint 4 backlog

No	Backlog	Story point	Tanggal selesai
1	Bug fixing tambah transaksi	2	9-03-2023
2	Laporan pemasukan/ pengeluaran harian	12	7-03-2023
3	Laporan pemasukan/ pengeluaran bulanan	12	6-03-2023
4	Laporan pemasukan/ pengeluaran tahunan	12	4-03-2023

Sprint 4 memiliki 38 *Story Point* dengan durasi pengerjaan 6 hari kerja. Dari data yang ditunjukkan pada Gambar 3, garis *remaining* pada hari ke-1 hingga ke 4 menurun dengan drastic dengan posisi di bawah garis *goal velocity* yang menandakan tim sedang sangat produktif sehingga menampilkan kinerja yang luar biasa dalam menyelesaikan *sprint backlog*, kemudian pada hari 4-5 garis *remaining* menjadi datar karena tim masih mencari solusi untuk menyelesaikan *sprint backlog* sisanya, pada hari 5-6 *sprint backlog* terakhir telah diselesaikan tepat pada waktunya. Grafik *burndown* Sprint 4 dapat dilihat pada Gambar 4.



Gambar 4. Burndown chart sprint 4

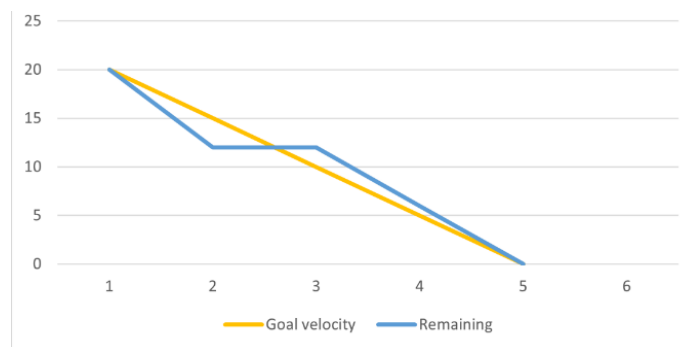
Pada *sprint 5* ada 8 *backlog* dengan 20 *Story Point* dengan rincian yang dapat dilihat pada Tabel 6.

Tabel 6. Sprint 5 backlog

No	Backlog	Story point	Tanggal selesai
1	Tambah Kategori	2	15-03-2023
2	Lihat Kategori	2	13-03-2023
3	Ubah Kategori	2	16-03-2023

No	Backlog	Story point	Tanggal selesai
4	Hapus Kategori	2	16-03-2023
5	Tambah Pelanggan	4	13-03-2023
6	Lihat Pelanggan	2	13-03-2023
7	Ubah Pelanggan	4	15-03-2023
8	Hapus Pelanggan	2	16-03-2023

Sprint 5 memiliki 18 *Story Point* dengan durasi pengerjaan 5 hari kerja. Dari data yang ditunjukkan pada Gambar 4 menunjukkan garis *remaining* pada hari ke 1-2 mengalami *progress* di mana tim dapat menyelesaikan beberapa *sprint backlog* yang diberikan, pada hari 2-3 garis *remaining* tidak mengalami penurunan sehingga garis *remaining* berada di atas garis *goals velocity* yang berarti tim mengalami kemunduran karena mengalami kendala selama proses pengerjaan sehingga *sprint backlog* belum bisa memenuhi *acceptance criteria*, garis *remaining* pada hari ke 3-5 mengalami penurunan drastis sehingga *sprint backlog* berhasil diselesaikan tepat waktu. Grafik *burndown* *Sprint 5* dapat dilihat pada Gambar 5.



Gambar 5. Burndown chart sprint 5

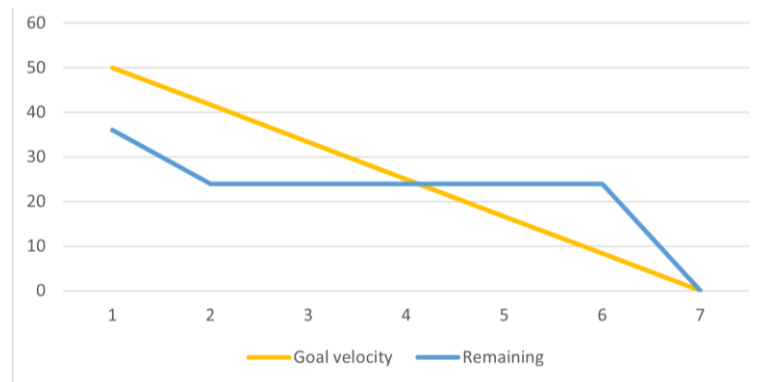
Pada *sprint 6* dikerjakan 8 *backlog* dengan 50 *Story Point* dengan rincian yang dapat dilihat pada Tabel 7.

Tabel 7. Sprint 6 backlog

No	Backlog	Story point	Tanggal selesai
1	Tambah Item Pesanan	4	17-03-2023
2	Lihat Item Pesanan	4	17-03-2023
3	Ubah Item Pesanan	4	17-03-2023
4	Hapus Item Pesanan	2	17-03-2023
5	Card <i>Dashboard</i>	12	19-03-2023
6	Grafik Bar <i>Dashboard</i>	20	24-03-2023
7	<i>Bug fixing</i> ubah transaksi	2	24-03-2023
8	<i>Bug fixing</i> tambah transaksi	2	24-03-2023

Sprint 6 memiliki 50 *Story Point* dengan durasi pengerjaan 6 hari kerja. Dari data yang ditunjukkan pada Gambar 5 garis *remaining* pada hari pertama langsung berada di bawah garis *goals velocity*. Ini disebabkan karena tim langsung menuntaskan beberapa *sprint backlog* lebih awal yaitu *sprint backlog* tambah, ubah, lihat, hapus item pesanan agar *sprint backlog* lainnya dapat cepat selesai. Pada hari ke 2-6 garis *remaining* berbentuk datar hingga berada di atas garis *goals velocity*. Pada hari ke 4-7 pengerjaan sisa *sprint backlog* yang ada sedikit kompleks pengerjaannya, dikarenakan kurangnya pengetahuan tim dalam menghadapi problem tersebut sehingga tim *scrum* memerlukan waktu lebih untuk menuntaskan *sprint backlog*

tersebut. Pada hari 6-7 garis *remaining* mengalami penurunan hingga menjadi 0 yang artinya tim *scrum* dapat menuntaskan semua *sprint backlog* tepat pada waktunya. Grafik *burndown* Sprint 6 dapat dilihat pada Gambar 6.



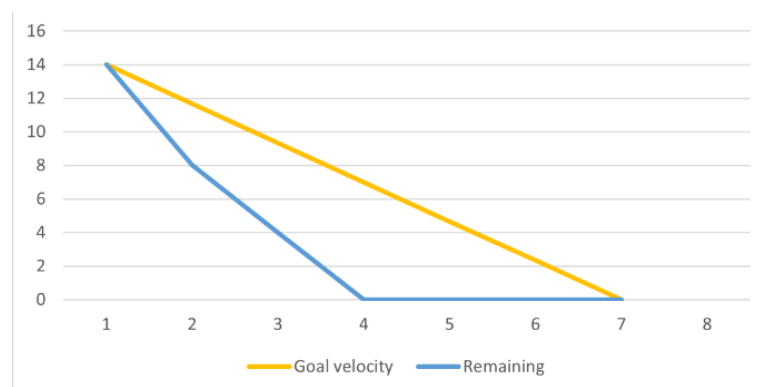
Gambar 6. Burndown chart sprint 6

Pada *Sprint* ke-7 ada 6 *backlog* yang dikerjakan dengan total 14 *Story Point* dengan rincian yang dapat dilihat pada Tabel 8.

Tabel 8. Sprint 7 backlog

No	Backlog	Story point	Tanggal selesai
1	<i>Bug fixing</i> Format tanggal (d-m-Y)	4	28-03-2023
2	<i>Bug fixing</i> Format Rupiah	2	28-03-2023
3	<i>Bug fixing</i> onChange pelanggan	2	29-03-2023
4	<i>Bug fixing</i> onChange ItemPesanan	2	29-03-2023
5	<i>Improvement</i> Tipe data no. hp	2	30-03-2023
6	<i>Bug fixing</i> Tanggal diisi otomatis	2	30-03-2023
7	<i>Bug fixing</i> Format tanggal (d-m-Y)	4	28-03-2023

Sprint 7 memiliki 14 *Story Point* dengan durasi pengerjaan 6 hari kerja. Dari data pada Gambar 6 menunjukkan garis *remaining* berada di bawah garis *goals velocity* dari awal hingga akhir. Hal ini dikarenakan tim sudah memahami apa yang harus dilakukan untuk *sprint backlog* yang ditentukan sehingga proses pengerjaan bisa selesai tepat waktu dan juga garis *remaining* sudah menyentuh 0 *Story Point* pada hari ke-4 sehingga tim sudah menuntaskan semua *sprint backlog* yang diberikan lebih cepat. Grafik *burndown* Sprint 7 dapat dilihat pada Gambar 7.



Gambar 7. Burndown chart sprint 7

Hasil Implementasi

Sistem dikembangkan dalam bahasa pemrograman PHP dengan memanfaatkan *framework* Bootstrap untuk mengatur tampilan, serta penyimpanan data menggunakan MySQL. Tampilan hasil implementasi halaman dashboard sistem dapat dilihat pada Gambar 8.



Gambar 8. Hasil implementasi halaman dashboard

Sistem dapat digunakan oleh beberapa aktor pengguna yaitu pemilik dan admin. Salah satu fitur yang dapat digunakan oleh admin yaitu menu Pesanan, untuk menangani pesanan yang masuk. Tampilan halaman pesanan dapat dilihat pada Gambar 9.

No.	Id	Tanggal	Pelanggan	Item	Jumlah	Subtotal	Pajak	Total	Status	Aksi
1	2	03-12-2002	Dinas Komunikasi dan Informasi Kota Depok	Sneck Box	100	Rp 1.500.000	Rp 150.000	Rp 1.650.000		Bayar Ubah Hapus
2	3	20-02-2023	Dinas Kesehatan Kota Depok	Sneck Box	21	Rp 315.000	Rp 31.500	Rp 346.500		Bayar Ubah Hapus
3	4	23-02-2023	Dinas Kesehatan Kota Depok	Sneck Box	20	Rp 300.000	Rp 30.000	Rp 330.000	Lunas	Bayar Ubah Hapus
4	5	16-03-2023	ESQ	Nasi Box	6	Rp 150.000	Rp 15.000	Rp 165.000	Lunas	Bayar Ubah Hapus
5	6	25-03-2023	Cianju Corp	Bakwan (Satuan)	10	Rp 10.000	Rp 1.000	Rp 11.000	Belum Lunas	Bayar Ubah Hapus
6	9	30-03-2023	Cianju Corp	Bakwan (Satuan)	100	Rp 100.000	Rp 10.000	Rp 110.000	Lunas	Bayar Ubah Hapus

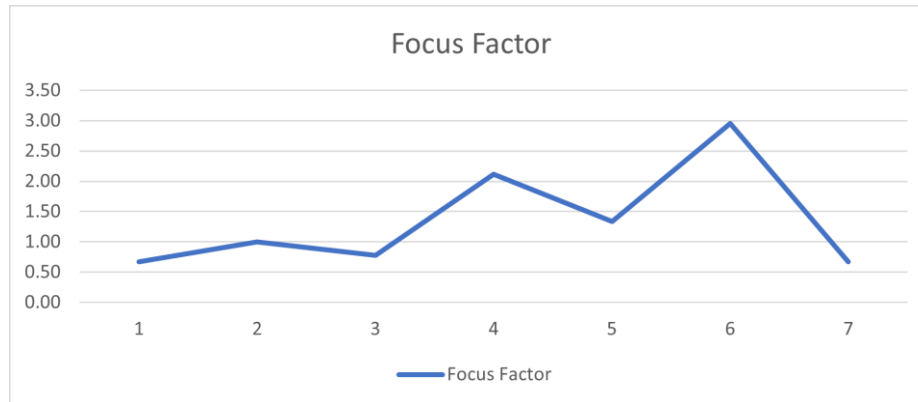
Gambar 9. Hasil implementasi halaman pesanan

Scrum dievaluasi dengan mengukur nilai *Focus Factor* pada setiap *sprint*. Pada rencana awal *project* akan dilaksanakan selama 12 *sprint*, namun *project* dapat diselesaikan pada *sprint* ke-7. Data pelaksanaan *sprint* dapat dilihat pada Tabel 9.

Tabel 9. Data pelaksanaan sprint

Sprint	Jumlah backlog	Story point	Jumlah hari	Focus factor
1	2	12	6	0,67
2	6	18	6	1,00
3	5	14	6	0,78
4	4	38	6	2,11
5	8	20	5	1,33
6	8	50	7	2,95
7	6	14	7	0,67

Grafik kinerja tim direpresentasikan dalam *Focus Factor* dapat dilihat pada Gambar 10.



Gambar 10. Kinerja tim

Berdasarkan grafik *Focus Factor* pada Gambar 10, terlihat bahwa terjadi fluktuasi dalam kinerja tim dari *sprint* ke *sprint*. Terdapat beberapa *sprint* yang mencapai *Focus Factor* yang tinggi, seperti *sprint* 4 dan 6, yang masing-masing memiliki *Focus Factor* sebesar 2,11 dan 2,95. Namun, ada juga beberapa *sprint* yang memperlihatkan kinerja yang kurang baik, seperti *sprint* 1 dan 3, yang masing-masing memiliki *Focus Factor* sebesar 0,67 dan 0,78. *Sprint* 2 dan 5 menunjukkan kinerja tim yang stabil, dengan *Focus Factor* yang relatif tinggi, yaitu masing-masing sebesar 1,00 dan 1,33. *Sprint* 7 memperlihatkan sedikit peningkatan dari *sprint* sebelumnya, dengan *Focus Factor* sebesar 1,17. Penyebab *sprint* 1 & 3 memiliki nilai terendah karena *Story Point* yang dibebankan kepada tim terbilang kecil, faktor tersebut terjadi karena tim masih membiasakan diri dan masih mempelajari dasar dari proyek yang akan dikerjakan. Sedangkan penyebab *sprint* 4 & 6 memiliki nilai *focus factor* tinggi dikarenakan *capacity* tim sangat besar yang berarti tim scrum dapat menanggung *Story Point* lebih dari seharusnya.

Faktor utama adalah tim mampu untuk konsentrasi penuh pada pengerjaan *backlog* karena tidak terdapat pekerjaan lain yang harus dikerjakan pada waktu bersamaan. Kemudian proses pengerjaan *sprint* jauh lebih mudah karena pemahaman seiring pengerjaan proyek semakin meningkat. Faktor lainnya antara lain saat tim berhasil menyelesaikan semua *Sprint Backlog* dalam satu *sprint*, kepercayaan diri tim meningkat sehingga lebih produktif pada *sprint* berikutnya.

Oleh karena itu, dapat disimpulkan bahwa fokus tim dalam mengimplementasikan proyek tersebut tidak konsisten di setiap *sprint*, namun secara umum tim berhasil mencapai *Focus Factor* yang tinggi pada beberapa *sprint* dan secara keseluruhan dapat membawa proyek menuju ke arah yang lebih baik.

Berdasarkan analisis *focus factor* di atas, berikut beberapa evaluasi dan rekomendasi terhadap tim untuk meningkatkan kinerjanya. Tim dapat melakukan evaluasi *sprint-sprint* dengan nilai *Focus Factor* rendah untuk mengidentifikasi faktor-faktor yang mempengaruhi penurunan efisiensi tim, membuat perencanaan *sprint* yang realistis dengan memanfaatkan data *Focus Factor* dari *sprint* sebelumnya. Anggota tim dapat meningkatkan kolaborasi dan komunikasi dalam mengembangkan sistem. Tim dapat menerapkan siklus pembelajaran dan peningkatan terus-menerus melalui retrospektif *sprint*. Tim juga dapat memberikan pelatihan dan pengembangan yang sesuai kepada anggota tim dalam penerapan *Scrum*, dan melakukan *monitoring* dan evaluasi berkala terhadap kinerja tim serta perkembangan *Focus Factor*.

Simpulan

Berdasarkan data yang dianalisis, *Focus Factor* bervariasi dari *sprint* ke *sprint*. Terdapat fluktuasi dalam kinerja tim, dengan beberapa *sprint* mencapai *Focus Factor* tinggi, seperti *sprint* 4 dan 6 dengan masing-masing nilai 2,11 dan 2,95. Namun, ada juga *sprint* yang mencapai nilai terendah seperti *sprint* 1 dan 3 dengan masing-masing nilai 0,67 dan 0,78. Secara keseluruhan, fokus tim tidak konsisten di setiap *sprint*, tetapi tim berhasil memperoleh nilai *Focus Factor* yang baik di mana nilai *focus factor* berada di atas 0,6 merupakan nilai minimal yang bermakna bahwa tim memiliki fokus dan produktivitas yang tinggi. Penelitian ini dapat dikembangkan dengan menerapkan *focus factor* pada studi kasus proyek yang lebih besar *user story*-nya untuk mengukur apakah tim mampu mencapai fokus yang tinggi dan produktif jika dihadapkan dengan proyek yang lebih besar.

Daftar Pustaka

- Abidin, R. (2023). Pemanfaatan QRCode dan Map dalam Pengembangan Aplikasi Penjualan Menggunakan Metode Scrum. *JURNAL UNITEK*, 16(2), 162–171. <https://doi.org/10.52072/unitek.v16i2.612>
- Almeida, F., & Carneiro, P. (2023). Perceived Importance of Metrics for Agile Scrum Environments. *Information*, 14(6), 327. <https://doi.org/10.3390/info14060327>
- Billah, M., & Nuraminah, A. (2022). Pengembangan Sistem E-Commerce dengan Framework Laravel dan Vuejs Menggunakan Metode Scrum. *Jurnal Teknik Informatika Dan Elektro*, 4(2), 130-136.
- Charkie, K. N. (2020). *Focus Factor and Hyper-Productivity of Agile Teams : A Study of 8 Open-Source Projects* [University Of Tartu]. https://comserv.cs.ut.ee/home/files/nimr_charkie_software_engineering_2020.pdf?study=ATILoputoo&reference=C10614E890564DEF0DD9D51EEDA01452C575098B
- Harahap, E. S. W., Kurniawan, E., & Putri, P. (2023). Penggunaan Metode Scrum Dalam Membangun Aplikasi Sistem Monitoring Update Maintenance Site Dan Genset Pt. Sibatel. *J-Com (Journal of Computer)*, 3(2), 101–110. <https://doi.org/10.33330/j-com.v3i2.2505>
- Poženel, M., Fürst, L., Vavpotič, D., & Hovelja, T. (2023). Agile Effort Estimation: Comparing the Accuracy and Efficiency of Planning Poker, Bucket System, and Affinity Estimation Methods. *International Journal of Software Engineering and Knowledge Engineering*, 33(11n12), 1923–1950. <https://doi.org/10.1142/S021819402350064X>
- Mallidi, R. K., & Sharma, M. (2021). Study on Agile Story Point Estimation Techniques and Challenges. *International Journal of Computer Applications*, 174(13), 9–14. <https://doi.org/10.5120/ijca2021921014>

- Park, D. S., & Noh, J. Y. (2023). The Effect of Sprint Duration to the Velocity in a Large-Scale Embedded Software Project. *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1–5. <https://doi.org/10.1109/ESEM56168.2023.10304859>
- Putri, R. P. D., & Fatichah, C. (2023). Analysis of Level Team Effectiveness in The Implementation of Scrum Using Evidence-Based Management (Case Study: Company A as A Fintech Industry). *IPTEK The Journal for Technology and Science*, 34(2), 95. <https://doi.org/10.12962/j20882033.v34i2.15789>
- Raicevic, S. S., Garvanov, I., & Raicevic, N. (2023). Improvements and Shortcomings in The First Stage of Scrum Implementation in Medium-Sized IT Companies. *2023 XXXII International Scientific Conference Electronics (ET)*, 1–4. <https://doi.org/10.1109/ET59121.2023.10278704>
- Rizaldi, A., Maria, E., Wahyono, T., Purwanto, P., & Hartomo, K. D. (2022). Analisis Penerapan Metode Scrum Pada Pengembangan Sistem Informasi Akuntansi Koperasi. *Jurnal Media Informatika Budidarma*, 6(1), 57. <https://doi.org/10.30865/mib.v6i1.3349>
- Sihombing, D. J. C., & Ferdiando, F. (2023). Analysis of The Implementation of SCRUM in The Development of A BBLobster Content Management System. *Journal of Information Systems and Informatics*, 5(1), 15–30. <https://doi.org/10.51519/journalisi.v5i1.411>
- Sutherland, K. S. & J. (2020). *The Scrum Guide 2020*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- Tahyudin, I., & Zidni Iman Sholihati. (2022). Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 6(1), 169–176. <https://doi.org/10.29207/resti.v6i1.3793>
- Vaid, K., & Ghose, U. (2020). Predictive Analysis of Manpower Requirements in Scrum Projects Using Regression Techniques. *Procedia Computer Science*, 173, 335–344. <https://doi.org/10.1016/j.procs.2020.06.039>
- Verwijs, C., & Russo, D. (2023). A Theory of Scrum Team Effectiveness. *ACM Transactions on Software Engineering and Methodology*, 32(3), 1–51. <https://doi.org/10.1145/3571849>