

# PENGEMBANGAN SISTEM OTOMATISASI TAGIHAN MENGUNAKAN METODE AGILE SOFTWARE DEVELOPMENT

Fahrul Nurzaman<sup>1</sup>

<sup>1</sup> Teknik Informatika UPI YAI  
Jl. Pangeran Diponegoro No. 74 Jakarta Pusat  
E-mail : fnurzaman@gmail.com<sup>1</sup>

## **ABSTRAK**

*Efektifitas dan efisiensi dalam proses operasional menjadi hal yang penting dalam peningkatan produktifitas bisnis perusahaan terutama dalam menghadapi persaingan bisnis. Perubahan proses administrasi dan proses kerja dari proses manual ke proses otomatisasi dan digitalisasi sudah menjadi kebutuhan yang penting bagi perusahaan. Berkembangnya Teknologi Informasi membantu adanya perubahan tersebut. Proses penagihan pembayaran kepada Pelanggan merupakan bagian dari operasional dalam perusahaan. Dimana proses tersebut merupakan proses pembuatan dan pengiriman tagihan yang menjadi kewajiban yang harus dibayar pelanggan atas barang atau jasa yang telah diberikan perusahaan kepada pelanggan. Agar proses pembuatan dan pengiriman tagihan tidak dilakukan secara manual maka dikembangkan sebuah sistem otomatisasi tagihan yang memiliki tujuan secara otomatis membuat dan mengirimkan tagihan beserta rincian lampiran dan invoice nya kepada pelanggan melalui email. Sistem otomatisasi tagihan dikembangkan dengan menggunakan agile software development yang merupakan metodologi pengembangan aplikasi yang memerlukan proses adaptasi cepat terhadap segala bentuk perubahan. Metode yang digunakan dalam agile software development adalah Metode Scrum , dimana metode ini terdiri dari sprint dan beban kerja yang mana masing-masing sprint didorong oleh Prioritized product backlog. Prioritized Product Backlog terdiri dari fitur terbaru, perbaikan bug, dan hal lain yang memberikan kontribusi pada produk akhir. Metode Scrum membuat beberapa blok waktu yang dinamakan sprint. Metode ini memiliki tiga artefak, yaitu Prioritized product backlog, sprint backlog, dan deliverable. Pada penulisan ini dibahas bagaimana mengembangkan sistem otomatisasi tagihan dengan menggunakan Metode Scrum agar lebih efektif dalam proses kerja dan mengganti proses manual dalam membuat dan mengirim tagihan kepada pelanggan.*

**Kata kunci : Sistem Otomatisasi Tagihan, Agile Software Development, Metode Scrum**

## **ABSTRACT**

*Effectiveness and efficiency in operational processes become important in increasing the productivity of the company's business, especially in the face of business competition. Changes in administrative processes and work processes from manual processes to automation and digitalization processes have become important needs for companies. The development of Information Technology helped bring about these changes. The process of billing payments to customers is part of operations within the company. Where the process is the process of making and sending bills that become obligations that must be paid by customers for goods or services that have been provided by the company to customers. So that the process of making and sending bills is not done manually, a bill automation system has been developed which aims to automatically create and send bills along with details of attachments and invoices to customers via*

*email. The billing automation system was developed using agile software development which is an application development methodology that requires a rapid adaptation process to all forms of change. The method used in agile software development is the Scrum Method, where this method consists of sprints and workloads, where each sprint is driven by the Prioritized product backlog. Prioritized Product Backlog consists of the latest features, bug fixes, and other things that contribute to the final product. The Scrum method creates several time blocks called sprints. This method has three artifacts, namely Prioritized product backlog, sprint backlog, and deliverables. At this writing discussed how to develop a billing automation system using the Scrum Method to make it more effective in the work process and replace the manual process in making and sending invoices to customers.*

**Keyword : Billing Automation System, Agile Software Development , Scrum Method**

## 1. PENDAHULUAN

Pembuatan dan pengiriman tagihan kepada pelanggan atas barang atau jasa yang diberikan perusahaan merupakan bagian dari proses kerja operasional dan administrasi yang harus dilakukan untuk demi kelancaran bisnis perusahaan. Tagihan kepada pelanggan mempengaruhi alur kas perusahaan dimana tagihan yang dibayar pelanggan dapat digunakan untuk biaya operasional atau biaya lain-lain pada perusahaan. Pentingnya pembuatan dan pengiriman tagihan kepada pelanggan harus dilakukan dengan tertib dan tepat waktu untuk menghindari adanya keterlambatan pengiriman atau bahkan tagihan yang tidak dibuat dan melewati batas waktu penagihan. Banyak nya pelanggan tidak memungkinkan proses pembuatan dan pengiriman dibuat secara manual. Jika dilakukan manual maka akan banyak membutuhkan waktu yang cukup lama dalam membuat tagihan , lampiran rinciannya dan mengirim tagihannya. Maka diperlukan otomatisasi dan digitalisasi dalam proses pembuatan dan pengiriman tagihan. Dengan adanya sistem tersebut maka tidak perlu lagi mengirimkan tagihan berupa cetak kertas ke alamat domisili atau perusahaan pelanggan tapi cukup dengan melalui elektronik surat dan tagihan dapat berupa digital dan rinciannya dapat berupa softcopy dalam bentuk spreadsheet (file excel). Dengan adanya

sistem tersebut akan menghemat dalam segi waktu dan biaya.

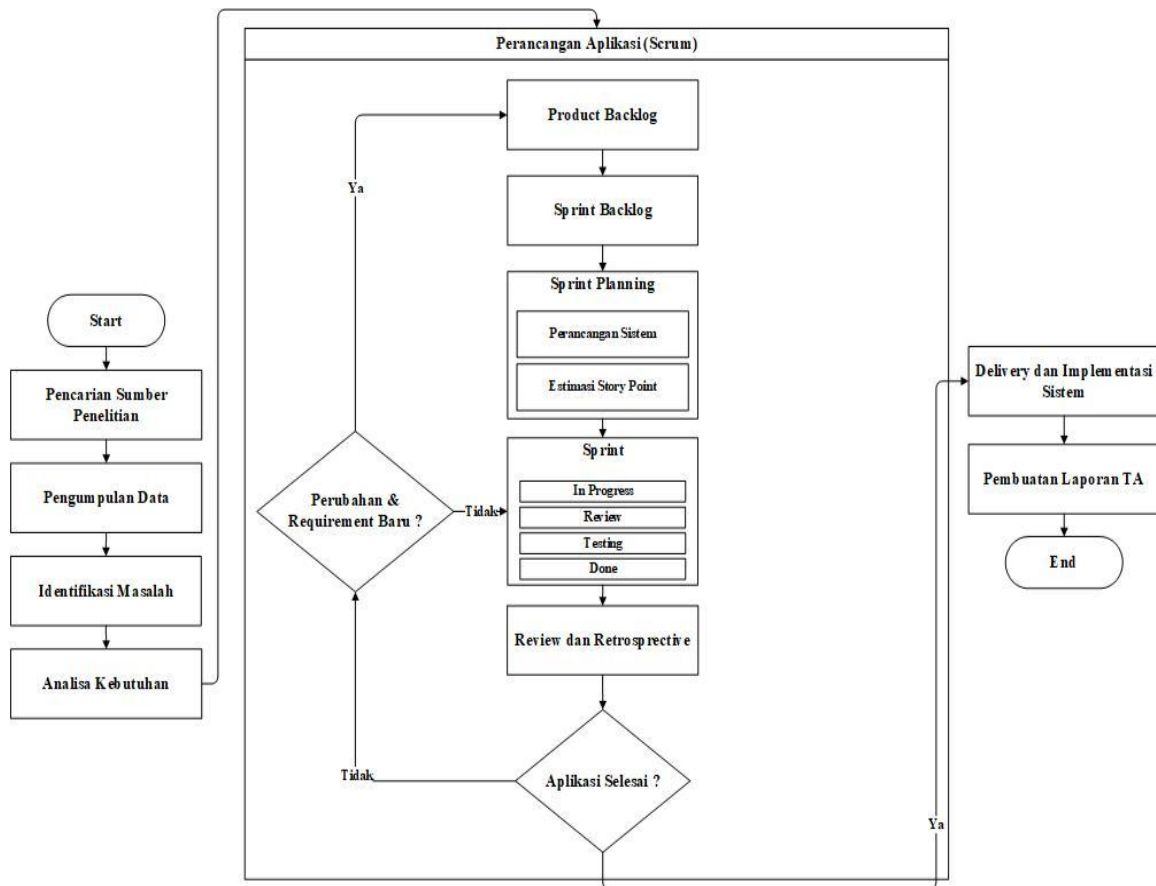
Dari permasalahan tersebut maka dapat dirumuskan yaitu bagaimanakah cara mengembangkan sebuah sistem yang dapat membuat dan mengirimkan tagihan secara otomatis ke email setiap pelanggan. Penelitian ini bertujuan untuk mengembangkan sebuah Sistem yang mengganti proses pembuatan dan pengiriman tagihan secara manual menjadi otomatis dimana proses itu semua dilakukan oleh mesin atau sistem tanpa melibatkan orang didalamnya. Dalam penelitian ini, pengembangan sistem dilakukan menggunakan prinsip *Agile Software Development* dengan metode *Scrum*.

## 2. METODOLOGI

Metode penelitian yang digunakan oleh penulis adalah metode deskriptif atau dikenal dengan metode survei. Metode deskriptif adalah suatu metode dalam meneliti status sekelompok manusia, suatu objek, suatu set kondisi, suatu sistem pemikiran, ataupun suatu peristiwa pada masa sekarang (Nasir,2003). Ada pun langkah-langkah yang dilakukan adalah diantaranya :

### 1. Observasi

Penelitian ini dilakukan dengan cara mensurvei dan mengamati sistem yang sedang berjalan untuk dapat menentukan aplikasi seperti apa yang nantinya akan



Gambar1. Kerangka Pemikiran Penelitian

dibuat serta menemukan kelemahan dari sistem yang sedang berjalan.

## 2. Wawancara

Penelitian ini dilakukan dengan mengajukan pertanyaan spesifik kepada pihak – pihak yang terkait di dalam sistem untuk mendapatkan penjelasan tentang sistem yang sedang berjalan dan menentukan kebutuhan user untuk aplikasi yang nantinya akan dibangun.

## 3. Studi Pustaka

Mencari informasi perihal teori-teori yang berhubungan dengan penelitian dari berbagai sumber/literature seperti buku, jurnal, dan sebagainya.

Untuk kerangka pemikiran penelitian menggunakan kerangka kerja *Agile Software Development* dengan Metode Scrum. Pada Gambar 1 adalah kerangka pemikiran penelitian dengan mengikuti Kerangka Kerja Metode *Scrum* (Prastio & Ani, 2018).

Berikut di bawah ini penjelasannya :

### 1. Pencarian Sumber Penelitian

Proses Otomatisasi pembuatan dan pengiriman tagihan kepada pelanggan menjadi bahan penelitian yang akan dikerjakan oleh penulis.

### 2. Pengumpulan Data

Data yang dapat dikumpulkan diantaranya adalah data email pelanggan , data rincian penagihan untuk dijadikan lampiran, data isi pesan yang tercantum pada email yang dikirim.

### 3. Identifikasi Masalah

Identifikasi masalah pada penelitian ini adalah adanya proses manual dalam pembuatan dan pengiriman tagihan kepada pelanggan. Sehingga perlu diubah menjadi proses otomatisasi, dimana dengan adanya perubahan proses otomatisasi dapat menghasilkan efektifitas kerja, menghemat waktu dan biaya serta meminimalisir keterlambatan pembuatan dan pengiriman tagihan ke pelanggan yang dapat mengakibatkan terlambatnya pembayaran yang dilakukan pelanggan, sehingga mempengaruhi alur kas perusahaan. Perumusan masalahnya adalah bagaimana mengembangkan sistem otomatisasi dalam membuat dan mengirimkan tagihan ke pelanggan melalui email. Sistem ini secara otomatis memroses tagihan tanpa adanya keterlibatan orang sehingga tagihan secara otomatis dapat dibuat dan dikirim setiap hari. Dengan adanya sistem ini dapat menciptakan nilai tambah bagi perusahaan yaitu proses kerja menjadi efektif dan efisien, serta pelayanan yang prima dengan menerapkan Teknologi Informasi.

### 4. Analisa Kebutuhan

Analisa kebutuhan terdiri dari kebutuhan fungsional dan non fungsional pada sistem yang dikembangkan. Sistem harus mempunyai fungsi-fungsi yang dibutuhkan agar berjalan dengan baik seperti fungsi mengambil data rincian dari Sistem Core, fungsi mengambil data email pelanggan, membuat file excel, memasukkan file excel ke dalam lampiran email, membuat isi email berupa tagihan, dan mengirimkan tagihan beserta rincian lampiran ke email pelanggan.

### 5. Perancangan dan Pembuatan Aplikasi

Perancangan dan pembuatan aplikasi ini terdiri dari berbagai langkah menggunakan metode Scrum, berikut adalah langkah-langkah perancangan aplikasi serta penjelasannya:

#### a. Product Backlog

*Product backlog* berisi backlog item yang dibuat berdasarkan *requirements* yang didapat dari pengumpulan data (observasi,

wawancara dan studi literature). *Requirements* pada *product backlog* bersifat dinamis sehingga akan terus menerus bertambah apabila mendapatkan feedback dari pengguna pada saat review dan demo sistem otomatisasi tagihan.

#### b. Sprint Backlog

*Sprint backlog* adalah *product backlog* yang sudah dibagi menjadi beberapa bagian untuk dikerjakan pada fase *sprint* nanti. Durasi berlangsungnya *sprint* biasanya antara 1-4 minggu tergantung kesepakatan bersama Scrum Team.

#### c. Sprint Planning

*Sprint Planning* adalah perencanaan dalam pengerjaan *product backlog* pada *sprint*. Dalam tahap ini terdiri dari perancangan sistem, perancangan basisdata dan perancangan arsitektur sistem, dan alur proses sistem otomatisasi tagihan. Selain itu dalam *Sprint Planning* juga mengestimasi waktu pengerjaan fitur pada masing-masing *sprint*.

#### d. Sprint

*Sprint* terdiri dari unit kerja yang dibutuhkan untuk mencapai kebutuhan yang didefinisikan di dalam *backlog* yang harus diselesaikan dengan waktu yang telah ditentukan sebelumnya kepada tim. Pada tahap ini pembuatan Sistem Otomatisasi Tagihan sudah mulai dikerjakan sesuai dengan *Sprint Planning*. Tahapan *sprint* terdiri dari *In Progress*, *Review*, *Testing*, dan *Done*.

#### e. Review dan Retrospective

Setelah selesai dari fase *sprint*, sistem otomatisasi tagihan dilakukan pengujian terakhir. Pengujian sistem ini dilakukan dengan mencari kekurangan pada sistem otomatisasi tagihan. Selanjutnya dilakukan *retrospective* apakah ada masukan berupa feedback atas *functional requirements* yang telah direview. Jika ada perubahan atas suatu fungsi, maka akan dimasukkan ke

dalam *backlog* tambahan untuk dilakukan di sprint selanjutnya. Bila tidak ada aplikasi sudah siap untuk release.

#### 6. Delivery dan Implementasi Sistem

Langkah selanjutnya adalah *delivery* dan implementasi sistem, yaitu merilis sistem otomatisasi tagihan untuk siap diimplementasikan.

#### 7. Pembuatan Laporan

Langkah terakhir dari penelitian ini adalah membuat laporan penelitian setelah mendapatkan kesimpulan. Laporan ini berisi hal-hal yang dikerjakan selama penelitian dan hasil yang didapatkan pada saat melakukan penelitian ini.

### 3. LANDASAN TEORI

Dari segi filosofi, *agile* mempunyai arti antara lain: mendorong demi terciptanya kepuasan pelanggan; mempercepat *delivery* perangkat lunak secara bertahap (*incremental*); tim proyek yang ramping dan mempunyai motivasi yang sangat tinggi; minimasi pekerjaan; serta menyederhanakan (birokrasi) keseluruhan proses pembangunan perangkat lunak. Dari segi pedoman pengembangan perangkat lunak, *agile* mempunyai pengertian, bahwa secara aktif dan berkesinambungan, antara pengembang dengan pelanggan harus senantiasa menjalin kerjasama dan komunikasi dengan baik. *Agile software Development* adalah salah satu metodologi dalam pengembangan sistem perangkat lunak (*software*). Kata *Agile* berarti bersifat cepat, ringan, bebas bergerak, waspada. Kata ini digunakan sebagai kata yang menggambarkan konsep model proses yang berbeda dari konsep model-model proses yang sudah ada. Konsep *Agile software Development* dicetuskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa *Agile Software Development* adalah cara membangun software dengan melakukannya dan membantu orang lain membangunnya sekaligus (Ambler, An Introduction to Agile Modeling, 2001-2008). *Agile Software*

*Development* adalah jenis pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dan pengembang terhadap perubahan dalam bentuk apapun. Dalam *Agile Software Development*, interaksi dan personel lebih penting dari pada proses dan alat, software yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan klien lebih penting dari pada negosiasi kontrak, dan sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana.

*Agile Software Development* juga dapat diartikan sekelompok metodologi pengembangan software yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun (Dictio.id, 2016). Salah satu ciri dari *Agile Software Development* adalah tim yang tanggap terhadap perubahan karena perubahan adalah hal yang utama dalam membangun software: perubahan kebutuhan software, perubahan anggota tim, perubahan teknologi dan lain-lain. Selain itu *Agile Software Development* juga melihat pentingnya komunikasi antara anggota tim, antara orang-orang teknis dan businessmen, antara developer dan manajernya.

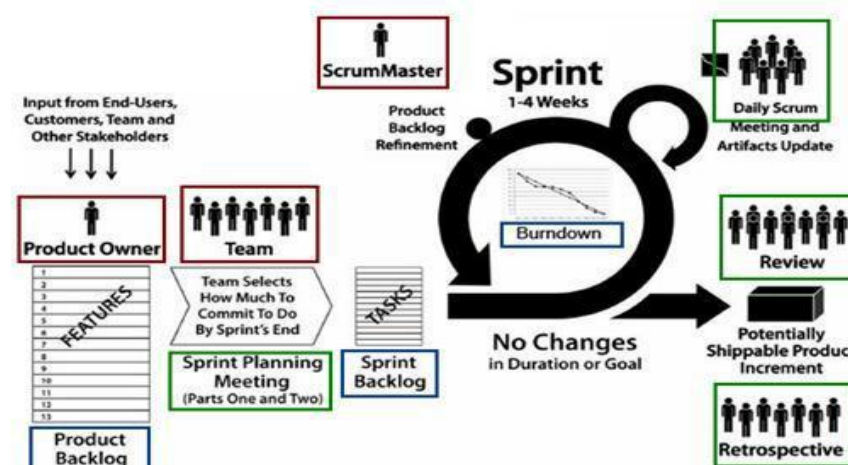
*Scrum* merupakan *framework* yang digunakan untuk implementasi metode pengembangan aplikasi *agile*. *Agile* ialah metodologi pengembangan aplikasi yang memerlukan proses adaptasi cepat terhadap segala bentuk perubahan. *Scrum* adalah sebuah kerangka kerja proses yang digunakan untuk mengelola pengembangan sebuah produk kompleks, berguna untuk menjadikan produk dengan nilai setinggi mungkin secara produktif dan kreatif (Schwaber & Jeff, 2013). Karena *scrum* adalah sebuah kerangka kerja, maka *scrum* bukanlah sebuah proses, teknik, ataupun metodologi, namun *scrum* ini bisa mencakup berbagai proses, teknik, ataupun metodologi di dalamnya (Schwaber & Jeff, 2013). *Scrum* sudah digunakan sejak awal tahun 1990. Beberapa alasan *scrum* banyak digunakan di seluruh dunia dalam

mengelola dan mengembangkan produk, adalah sebagai berikut :

- Untuk meneliti dan menggali potensi pasar, teknologi, dan kemampuan produk
- Untuk mengembangkan produk dan peningkatan-peningkatannya.
- Untuk merilis produk dan peningkatan-peningkatannya, sesering mungkin di setiap hari.
- Untuk mengembangkan dan memelihara operasional sistem komputasi awan (daring, keamanan, sesuai permintaan) dan lingkungan operasional lain untuk penggunaan produk.
- Untuk mengelola dan memperbarui sebuah produk. (Schwaber & Jeff, 2013)

*Scrum* menjamin transparansi dalam komunikasi dan menciptakan lingkungan dengan tanggung jawab bersama-sama untuk evolusi secara berlanjut (Satpathy, 2016). Di dalam *Scrum Team* hanya terdiri dari 3 peran, yaitu *Product Owner*, *Development Team*, dan *Scrum Master* (Rad Turley, 2013). *Scrum Team* bersifat swakelola, dan

lintas-fungsi. Arti dari swakelola adalah tim dapat memilih cara terbaik dalam mengerjakan pekerjaan mereka, bukan diperintah oleh orang lain di luar tim ini. Sedangkan lintas-fungsi berarti anggota dari tim memiliki semua keahlian yang dibutuhkan untuk menyelesaikan pekerjaan mereka tanpa bergantung pada orang lain di luar tim ini. (Schwaber & Jeff, 2013). Dalam kerangka kerja *scrum*, disarankan untuk membagi proses pengembangan menjadi beberapa rangkaian *sprint*. Proses *sprint* rata-rata diadakan selama satu hingga empat minggu. Selama *sprint*, tim *scrum* mengatur fitur atau *backlog* ide ke fungsi. Fitur-fitur ini dikodekan, diuji, dan diimplementasikan ke dalam perangkat lunak atau produk yang terus berkembang. Proses *scrum* ditunjukkan pada Gambar 2.



Gambar2. Proses *Scrum*

#### 4. HASIL DAN PEMBAHASAN

Dalam pengembangan sistem otomatisasi tagihan menggunakan metode

*scrum*, langkah pertama yang dilakukan adalah menyusun *product backlog* berdasarkan requirement yang didapat melalui observasi dan pengumpulan data. Berikut *product backlog* yang terdapat pada sistem otomatisasi tagihan ditampilkan pada tabel 3.

**Tabel 1.** Product Backlog Sistem Otomatisasi Tagihan

| ID | Backlog Item Sistem Otomatisasi Tagihan                                     |
|----|---|
| 1  | Mengambil data dari Basis Data Sistem <i>Core</i>                           |
| 2  | Mengecek adanya tagihan yang dibuat di Sistem <i>Core</i>                   |
| 3  | Menyimpan data ke Basis Data Sistem Tagihan Otomatis                        |
| 4  | Membuat file Excel Lampiran yang diambil dari Basis Data Sistem <i>Core</i> |
| 5  | Mengambil basis data email pelanggan  |
| 6  | Menampilkan tagihan ke <i>message body</i> email                            |
| 7  | Melampirkan file excel ke dalam email                                       |
| 8  | Mengirimkan email secara otomatis   |

Langkah kedua adalah Sprint Backlog yaitu dengan membuat daftar product backlog item yang terpilih dan terurut berdasarkan prioritasnya untuk dikerjakan dalam sprint dengan perencanaan dan

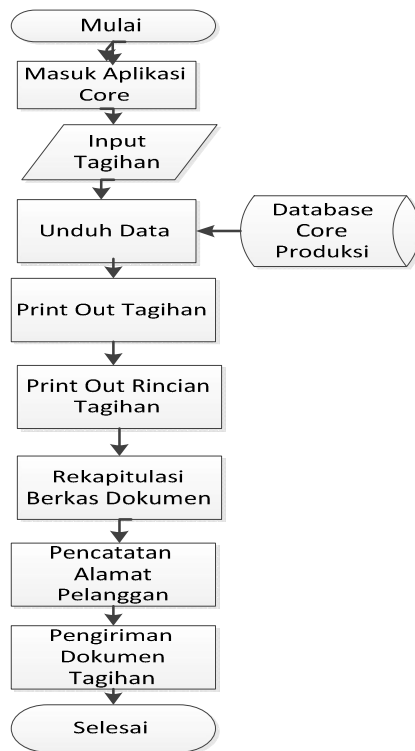
perkiraan mengenai fungsionalitasnya untuk menghasilkan produk yang diinginkan (Schwaber & Jeff, 2013). Berikut ini adalah tabel 4 yang berisi Sprint Backlog.

**Tabel 2.** Sprint Backlog Sistem Otomatisasi Tagihan

| Sprint Backlog Sistem Otomatisasi Tagihan |  |          |          |
|---|--|----------|----------|
| ID  | Story  | Estimate | Priority |
| 2   | Sistem dapat mengecek ada nya tagihan di Sistem <i>Core</i>                              | 1        | 1        |
| 1   | Sistem dapat mengambil data di Basis Data Sistem <i>Core</i>                             | 3        | 2        |
| 3   | Sistem dapat menyimpan data ke Basis Data Sistem Tagihan Otomatis                        | 2        | 3        |
| 6   | Sistem dapat menampilkan tagihan di <i>message body</i>                                  | 2        | 4        |
| 4   | Sistem dapat membuat file Excel Lampiran yang diambil dari Basis Data Sistem <i>Core</i> | 2        | 5        |
| 5   | Sistem dapat mengambil basis data email pelanggan  | 1        | 6        |
| 8   | Sistem dapat mengirim email secara otomatis  | 2        | 7        |
| 7   | Sistem dapat melampirkan file excel ke dalam email                                       | 2        | 8        |

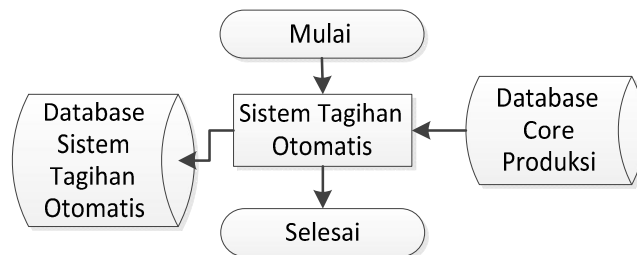
Langkah ketiga adalah membuat Sprint Planning yang terdiri dari alur proses manual, arsitektur sistem, alur proses kerja sistem , Perancangan basis data , dan arsitektur aplikasi. Di bawah ini adalah alur

proses yang berjalan untuk pembuatan dan pengiriman tagihan jika dilakukan secara manual :



Gambar3. Proses Manual Proses Tagihan

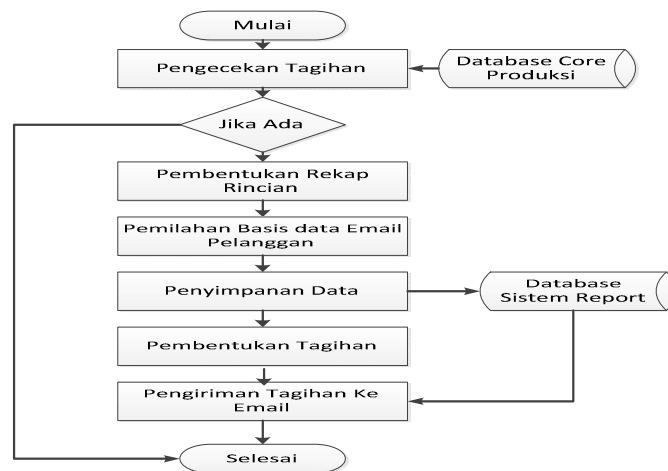
Di bawah ini adalah arsitektur sistem proses untuk pembuatan dan pengiriman tagihan dilakukan secara otomatis :



Gambar4. Arsitektur Sistem Tagihan Otomatis

Proses kerja Sistem Tagihan Otomatis adalah sebagai berikut seperti digambarkan dengan flow chart di bawah ini :





Gambar5. Alur Proses Sistem Tagihan Otomatis

Sistem melakukan pengecekan data secara otomatis setiap hari dari database Core produksi, jika ada nya tagihan maka sistem akan mengambil data dari database Core untuk membentuk rekap rincian lalu disimpan ke dalam database Sistem Tagihan Otomatis, lalu membentuk file spreadsheet (excel) lalu memilah database email pelanggan, lalu membentuk tagihan dan

mengirimkan tagihan beserta lampiran rinciannya ke email pelanggan. Basis data Sistem Tagihan Otomatis merupakan Basis Data Staging yang terpisah dengan Basis data Core Aplikasi Produksi. Basis Data Sistem Tagihan Otomatis terdiri dari Tabel Tagihan yang memiliki Struktur di bawah ini :

Tabel 3. Struktur Tabel Tagihan

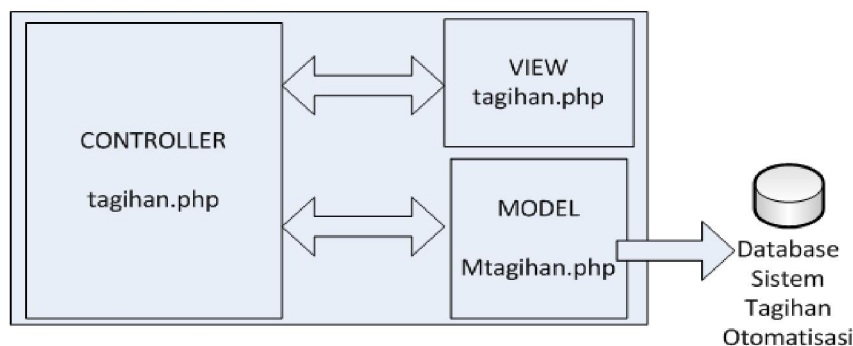
| Nama Fields       | Type Data    | Default    | Keterangan                        |
|-------------------|--------------|------------|-----------------------------------|
| no_tagihan        | char(8)      | NOT NULL   | Primary key                       |
| tgl_tagihan       | date         | dd/mm/yyyy | Tanggal Tagihan dibuat            |
| tgl_jatuh_tempo   | date         | dd/mm/yyyy | Tanggal Jatuh Temp Taggihan       |
| nama_pelanggan    | Varchar(100) | NOT NULL   | Nama Pelanggan                    |
| email_1           | Varchar(100) | NOT NULL   | Alamat email pelanggan            |
| email_2           | Varchar(100) | NULL       | Alamat email pelanggan (opsional) |
| memo              | Varchar(100) |            | Penjelasan Tagihan                |
| tgl_input_tagihan | date         | dd/mm/yyyy | Tanggal Proses Sistem             |
| tgl_kirim_email   | date         | dd/mm/yyyy | Tanggal Pengiriman Email          |
| nama_file         | Varchar(100) |            | Nama file lampiran                |
| status_kirim      | char(1)      | F          | Status data yang sudah/belum      |

Arsitektur Aplikasi Sistem Tagihan Otomatis menggunakan konsep MVC yaitu menggunakan pola Model, View, dan Controller. Konsep MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang terdiri dari manipulasi data, user interface, dan bagian

yang menjadi kontrol aplikasi (Supono,2012). Terdapat 3 jenis komponen yang membangun suatu pola MVC aplikasi yaitu View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller.

View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view. Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan

apa yang akan diproses oleh aplikasi. Berikut di bawah ini Arsitektur Aplikasi Sistem Tagihan Otomatis dimana dibangun dengan menggunakan Pemrograman PHP dengan framework Code Igniter serta Bootstrap sebagai framework Cascade Style Sheet. Implementasi sistem report otomatis menggunakan bahasa pemrograman PHP dengan menggunakan framework CodeIgniter dan Bootstrap sebagai framework CSS.



Gambar6. Alur Arsitektur Aplikasi Sistem Tagihan Otomatis

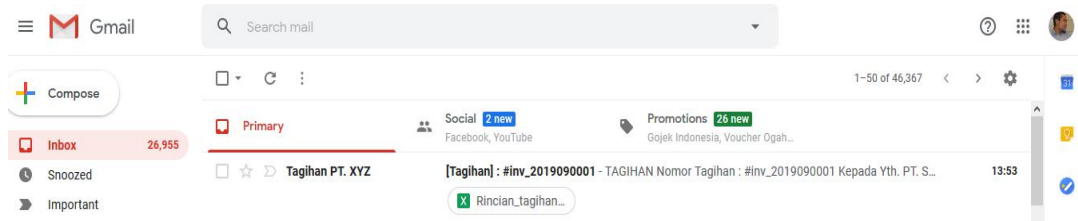
Setelah pemilihan item product backlog pada tahap sprint backlog, penulis merancang sprint yang berisi perencanaan banyaknya proses sprint yang diperlukan berdasarkan sprint backlog serta mengestimasi waktu pengerjaannya melalui Scrum Meeting dengan Bagian Administrasi

Operasional selaku Product Owner. Pada Pengembangan Sistem tagihan Otomatis dapat dilakukan 1 kali proses sprint dengan pengerjaannya selama 8 hari kalender dengan estimasi point 15. Detail estimasi waktu dan pengerjaan backlog item sprint dijelaskan pada di bawah ini :

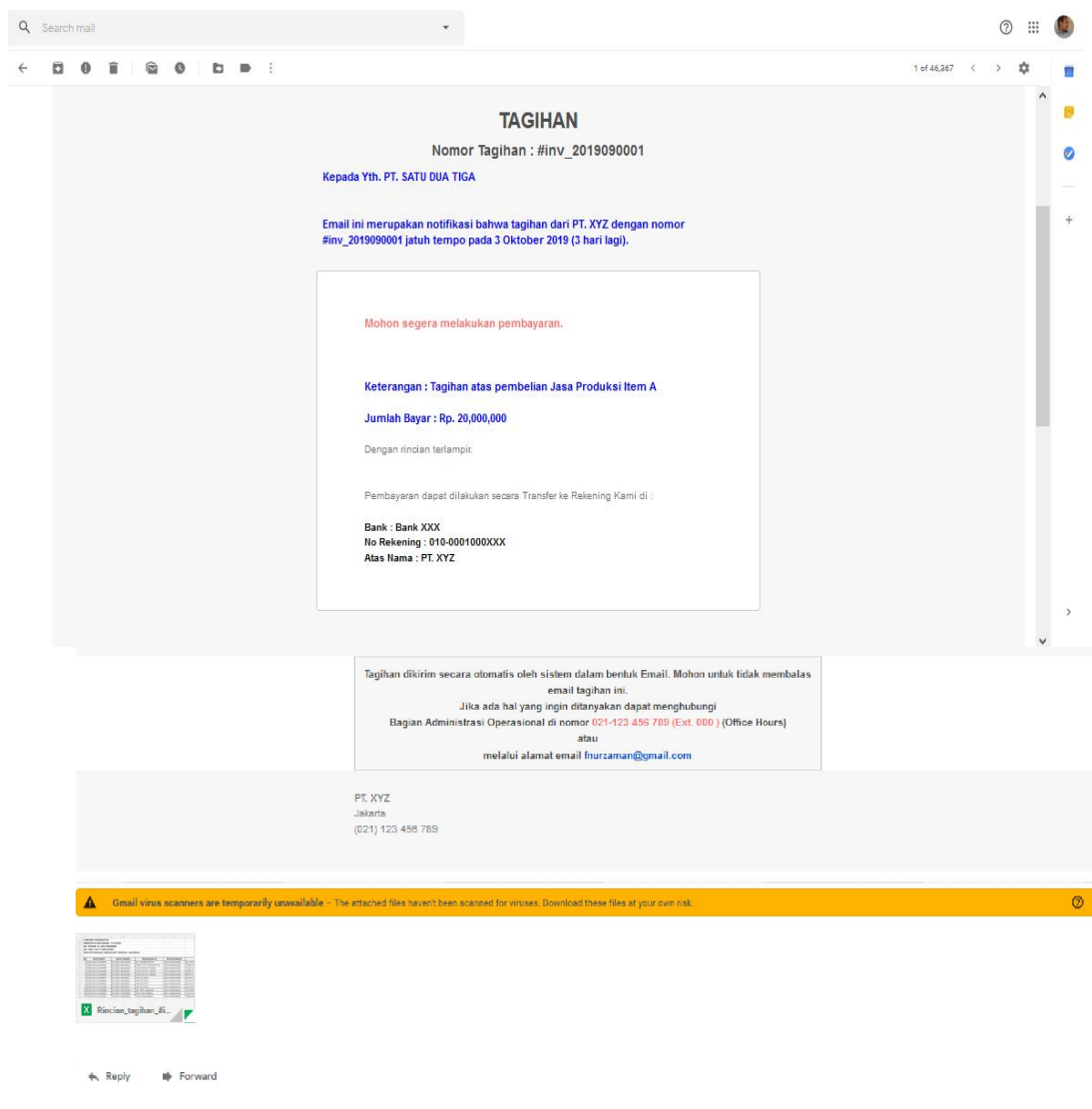
Tabel 3. Sprint

| Backlog Item (User Story)   | Start      | Days | End        | Story Point |
|---|------------|------|------------|-------------|
| Sprint  | 01-09-2019 | 8    | 09-09-2019 |             |
| Sistem dapat mencek ada nya tagihan di Sistem Core                                |            |      |            | 1           |
| Sistem dapat mengambil data di Basis Data Sistem Core                             |            |      |            | 3           |
| Sistem dapat menyimpan data ke Basis Data Sistem Tagihan Otomatis                 |            |      |            | 2           |
| Sistem dapat menampilkan tagihan di <i>message body</i>                           |            |      |            | 2           |
| Sistem dapat membuat file Excel Lampiran yang diambil dari Basis Data Sistem Core |            |      |            | 2           |
| Sistem dapat mengambil basis data email pelanggan                                 |            |      |            | 1           |
| Sistem dapat mengirimkan email secara otomatis                                    |            |      |            | 2           |
| Sistem dapat melampirkan file excel ke dalam email                                |            |      |            | 2           |

Hasil pengembangan Sistem Tagihan Otomatis ditampilkan pada gambar di bawah ini :



Gambar7. Tampilan Notifikasi Email Tagihan



Gambar8. Tampilan Tagihan Otomatis

Langkah terakhir yang dilakukan adalah Review dan Retrospective dimana Sistem Tagihan Otomatis direview kembali oleh product owner (bagian Administrasi Operasional) dan penulis. Setelah melakukan evaluasi dan pengujian akhir terhadap Sistem Tagihan Otomatis, maka Sistem Tagihan Otomatis sudah siap di-deploy ke Server Aplikasi dengan men-setting Task Schedule pada Cpanel Web Server yang secara otomatis berjalan sendiri.

## 5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut yaitu dalam pengembangan Sistem Tagihan Otomatis dibagi dalam beberapa tahap pengembangan, yaitu Product Backlog, Sprint Backlog, Sprint Planning, Sprint, kemudian Review dan Retrospective. Sistem Tagihan Otomatis memiliki fungsi utama membuat tagihan secara elektronik, membuat lampiran rincian dalam bentuk file spreadsheet dan mengirimkan tagihan ke email pelanggan secara otomatis. Setiap fungsi atau tugas Sistem Tagihan Otomatis hanya memerlukan proses sprint sebanyak 1 kali selama 8 hari. Hasil implementasi Sistem Tagihan Otomatis memudahkan perusahaan dalam pembuatan dan pengiriman tagihan kepada pelanggan sehingga prosesnya menjadi efektif dan efisien.

Pengembangan Sistem lebih lanjut dapat dilakukan dengan memberikan beberapa penambahan fungsi seperti file lampiran dalam bentuk file pdf dan notifikasi email secara otomatis untuk Tagihan yang sudah melewati jatuh tempo.

## DAFTAR PUSTAKA

Chrismanto Eka Prastio & Ani, Nur. (2018). *Aplikasi self service menu menggunakan metode Scrum berbasis android (case study : warkobar café cikarang)*. Jurnal Petir, 11(2).

Dictio.id. (2016). *Apa yang dimaksud dengan Scrum pada pengembangan perangkat lunak? - Komputer / Perangkat Lunak - Dictio Community*. Retrieved September 13, 2019, from <https://www.dictio.id/t/apakah-yang-dimaksud-dengan-agile-software-development/15199>

Nasir, M. (2003). *Metode Penelitian*. Jakarta : Ghalia Indonesia.

Satpathy, T. (2016). *A Guide to the Scrum Body of Knowledge (SBOKTM Guide)*. SCRUMstudyTM.

Schwaber, K., & Jeff, S. (2013). *Panduan Scrum TM*, (November).

Supono dan Virdiandry Putratama. (2012). *Pemrograman Web Dengan Menggunakan PHP dan Framework Codeigniter*. Yogyakarta: Deepublish.