

**PERANCANGAN SISTEM INFORMASI *CUSTOMER
RELATIONSHIP MANAGEMENT* MENGGUNAKAN
METODE SCRUM (STUDI KASUS : PT RABANA
SEJAHTERA INDONESIA)**

TESIS

Disusun sebagai salah satu syarat untuk
memperoleh gelar Magister Komputer

Oleh :

ADIB

2017210093



**PROGRAM STUDI PASCASARJANA
MAGISTER SISTEM INFORMASI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER LIKMI
BANDUNG
2019**

**PERANCANGAN SISTEM INFORMASI *CUSTOMER
RELATIONSHIP MANAGEMENT* MENGGUNAKAN
METODE SCRUM (STUDI KASUS : PT. RABANA
SEJAHTERA INDONESIA)**

Oleh :

Adib

2017210093

Bandung, 15 Oktober 2019
Menyetujui,

Dr. Hery Heryanto, S.Kom, M.Kom
Pembimbing

**PROGRAM STUDI PASCASARJANA
MAGISTER SISTEM INFORMASI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER LIKMI
BANDUNG
2019**

ABSTRAK

PERANCANGAN SISTEM INFORMASI *CUSTOMER RELATIONSHIP MANAGEMENT* MENGUNAKAN METODE SCRUM (STUDI KASUS : PT. RABANA SEJAHTERA INDONESIA)

Oleh :

Adib
2017210093

Saat ini, strategi *marketing* sudah tidak lagi hanya berorientasi pada kualitas produk saja, tetapi mengharuskan untuk melayani pelanggan dengan baik. *Service* dan Kualitas layanan prima kepada pelanggan menjadi kunci terciptanya hubungan baik dengan pelanggan dan munculnya loyalitas pelanggan terhadap perusahaan. PT. Rabana Sejahtera Indonesia merupakan perusahaan swasta yang bergerak dibidang produk obat herbal yang teruji secara klinis.

Pelanggan yang dimiliki PT. Rabana Sejahtera Indonesia berasal dari seluruh Indonesia dan luar negeri karena menjual produk secara *online* dan penjualan langsung kepada pelanggan. Pelanggan yang banyak tersebut merupakan salah satu keunggulan yang mendukung dalam perancangan sistem informasi *CRM*. Perancangan sistem informasi *CRM* dilakukan dengan menggunakan metode *Agile Development Scrum* yang dapat mempercepat proses development sistem informasi *CRM* PT. Rabana Sejahtera Indonesia.

Pada perancangan sistem informasi *CRM*, terdapat 3 (tiga) aktor yaitu : admin, pelanggan dan supervisor. Sistem Informasi *CRM* dibangun menggunakan bahasa pemrograman *PHP Framework CodeIgniter* dan *MySQL* untuk perancangan *database*. Dengan penerapan metode *Scrum*. *Product Owner* adalah Kepala divisi pemasaran, *Scrum Master* adalah *manajer* pemasaran dan *Development Team* terdiri dari lima orang yang mana dua orang merupakan *backend programmer*, dua orang merupakan *frontend programmer* dan seorang *quality engineer*.

Penelitian ini menghasilkan 5 (lima) *sprint* dan setiap *sprint* mempunyai durasi 2 (dua) minggu. *Sprint* satu menghasilkan *login* dan inti sistem, *Sprint* dua menghasilkan fitur *master data* pelanggan dan pegawai. *Sprint* tiga menghasilkan menu *tasks* untuk menilai kinerja pegawai dan menu *cases* untuk menangani masalah pelanggan. *Sprint* empat menghasilkan menu *leads* untuk menyimpan data pelanggan yang potensial dan menu *opportunities* untuk pelanggan yang sudah membeli produk dari perusahaan. *Sprint* lima menghasilkan fitur *email reminder* dan *chatting* secara *realtime*. *Email reminder* berguna untuk mengingatkan pelanggan tentang produk yang dibutuhkan oleh pelanggan.

Perancangan sistem informasi *CRM* dapat untuk membantu PT. Rabana Sejahtera Indonesia mengetahui kebutuhan *customer* secara personal serta memberikan pelayanan yang lebih maksimal dan informasi yang dibutuhkan oleh *customer* sehingga diharapkan dapat meningkatkan kepuasan *customer*.

Kata Kunci : *Customer Relationship Management*, *Customer*, Sistem Informasi *CRM*, *Agile Development Methods Scrum*.

ABSTRACT

CUSTOMER RELATIONSHIP MANAGEMENT INFORMATION SYSTEM DESIGN USING SCRUM METHOD (CASE STUDY: PT. RABANA SEJAHTERA INDONESIA)

By :

**Adib
2017210093**

Nowadays, marketing strategies are no longer oriented solely on product quality, but require serving customers well. Service and Excellent service quality to customers is the key to creating good relationships with customers and the emergence of customer loyalty to the company. PT. Rabana Sejahtera Indonesia is a private company engaged in herbal medicine products that are clinically tested.

Customers owned by PT. Rabana Sejahtera Indonesia comes from all over Indonesia and abroad because it sells products online and direct sales to customers. That many customers is one of the advantages that support in designing CRM information systems. The design of CRM information systems is carried out using the Agile Development Scrum method which can accelerate the process of developing CRM information systems of PT. Rabana Sejahtera Indonesia.

In the design of CRM information systems, there are 3 (three) actors: admin, customer and supervisor. CRM Information Systems are built using the PHP Framework CodeIgniter and MySQL programming languages for database design. With the application of the Scrum method. Product Owner is the head of the marketing division, Scrum Master is the marketing manager and the Development Team consists of five people of which two are backend programmers, two are frontend programmers and a quality engineer.

This study produces 5 (five) sprints and each sprint has a duration of 2 (two) weeks. Sprint one generates login and system core, Sprint two produces master and employee data master features. Sprint three produces task menus to assess employee performance and menu cases to handle customer problems. Sprint four produces menu leads for storing potential customer data and opportunities menu for customers who have already purchased products from the company. Sprint Five produces email reminder and chat features in realtime. Email reminder is useful for reminding customers about the products needed by customers.

The design of a CRM information system can help PT. Rabana Sejahtera Indonesia knows customer needs personally and provides maximum service and information needed by customers so that it is expected to increase customer satisfaction.

Keywords: Customer Relationship Management, Customers, CRM Information Systems, Agile Development Scrum.

KATA PENGANTAR

Segala Puji dan Syukur bagi Allah S.W.T. Tuhan semesta alam yang telah memberikan Rahmat dan Karunia-Nya, sehingga penulis dapat menyelesaikan kegiatan penelitian yang berjudul " PERANCANGAN SISTEM INFORMASI *CUSTOMER RELATIONSHIP MANAGEMENT* MENGGUNAKAN METODE *SCRUM* (Studi Kasus : PT. Rabana Sejahtera Indonesia) ". Penulis juga mengucapkan terima kasih kepada semua pihak yang telah membantu dan memberi dukungan yang sangat berarti, baik itu dalam proses pra- penelitian maupun dalam tahap penelitiannya itu sendiri. Oleh karena itu tidak berlebihan kiranya dalam kesempatan ini penulis bermaksud mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. Hery Heryanto, S. Kom, M. Kom, selaku pembimbing utama yang telah bersedia mencurahkan segala pemikiran, tenaga, serta waktu untuk membimbing penulis sampai selesai.
2. Seluruh pegawai PT. Rabana Sejahtera Indonesia yang telah banyak memberikan kesempatan dan motivasi kepada penulis.
3. Seluruh Staf pengajar Program Pascasarjana (S2) STMIK LIKMI yang telah memberikan ilmu pengetahuan, semoga jadi ilmu yang bermanfaat dan Alloh S.W.T melipatgandakan balasannya. Aamiin.
4. Seluruh teman-teman seperjuangan di kelas Eksekutif.
5. Kedua Orang Tua yang dengan sabar dan penuh kasih sayang membantu penulis baik motivasi dan do'anya sehingga penulis tetap bersemangat menyelesaikan tesis ini.
6. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah memberikan bantuan sehingga tesis ini dapat diselesaikan. Semoga dorongan, bantuan serta bimbingan Bapak/Ibu sekalian dapat menjadi amal baik dan mendapat imbalan dari Allah SWT. Amin.

Penulis menyadari dalam penulisan tesis ini masih jauh dari sempurna, oleh karena itu dengan senang hati dan penuh keterbukaan, penulis mengharapkan saran dan masukan untuk perbaikan tesis ini. Atas perhatian, bantuan, serta bimbingannya, penulis mengucapkan terima kasih.

Akhir kata, dengan segala kerendahan hati penulis mohon maaf yang sebesar besarnya kepada semua pihak bila ada hal-hal yang tidak berkenan di hati selama penyusunan tesis ini, dan mudah-mudahan Allah S.W.T senantiasa melimpahkan rahmat, ampunan dan kasih sayang-Nya kepada kita semua. Amin.

Bandung, 15 Oktober 2019

Penulis

DAFTAR ISI

ABSTRAK.....	i
KATA PENGANTAR	ii
DAFTAR ISI.....	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
1.5 Hasil Penelitian	5
1.6 Metode Penelitian	5
1.7 Sistematika Penulisan.....	5
BAB II LANDASAN TEORI.....	7
2.1 Pengertian Sistem Informasi.....	7
2.2 <i>Customer Relationship Management</i>	8
2.2.1 Definisi <i>Customer</i>	10
2.2.2 Pelayanan.....	10
2.2.3 Definisi Keluhan Pelanggan	11
2.2.4 Definisi Pelayanan Prima	11
2.3 <i>Agile Development Method</i>	12
2.3.1 Model <i>Scrum</i>	13
2.3.2 Penggunaan <i>Scrum</i>	14
2.3.3 Teori <i>Scrum</i>	15
2.3.4 Tata Nilai <i>Scrum</i>	17
2.3.5 <i>Scrum Team</i>	17
2.3.5.1 <i>Product Owner</i>	18

2.3.5.2 <i>Development Team</i>	18
2.3.5.3 <i>Scrum Master</i>	19
2.3.6 <i>Acara Acara Scrum</i>	20
2.3.6.1 <i>Sprint</i>	20
2.3.6.2 <i>Sprint Planning</i>	22
2.3.6.3 <i>Daily Scrum</i>	22
2.3.6.4 <i>Sprint Review</i>	24
2.3.6.5 <i>Sprint Retrospective</i>	25
2.3.7 <i>Artefak Scrum</i>	26
2.3.7.1 <i>Product Backlog</i>	27
2.3.7.2 <i>Sprint Backlog</i>	28
2.3.8 <i>Increment</i>	29
2.3.9 <i>Transparansi Artefak</i>	29
2.4 <i>Unified Modelling Language (UML)</i>	30
2.4.1 <i>Use Case Diagram</i>	30
2.4.2 <i>Class Diagram</i>	30
2.4.3 <i>Deployment Diagram</i>	31
2.4.4 <i>Component Diagram</i>	31
2.4.5 <i>Activity Diagram</i>	32
2.4.6 <i>Package Diagram</i>	32
2.5 <i>Tools yang digunakan</i>	33
2.5.1 <i>Hypertext Preprocessor (PHP)</i>	33
2.5.2 <i>MySQL</i>	33
2.6 <i>Penelitian Terdahulu</i>	34
BAB III OBJEK DAN METODOLOGI PENELITIAN	35
3.1 <i>Profile PT. Rabana Sejahtera Indonesia</i>	35
3.1.1 <i>Visi dan Misi PT. Rabana Sejahtera Indonesia</i>	37
3.1.2 <i>Struktur Organisasi Perusahaan</i>	38
3.1.3 <i>Deskripsi Tugas</i>	38

3.1.4 Rekap Penjualan PT. Rabana Sejahtera Indonesia.....	40
3.2 Proses Bisnis	41
3.3 Metode Pengembangan Sistem Informasi.....	44
BAB IV PEMBAHASAN DAN HASIL PENELITIAN	47
4.1 Tinjauan Perancangan Sistem Informasi <i>Customer Relationship Management</i>	47
4.2 <i>Scrum Team CRM</i> Rabana	49
4.3 Pemilihan <i>Tools</i>	50
4.4 Lingkungan Kerja Fisik.....	50
4.5 <i>Sprint 1</i> Modul <i>Core</i> Sistem dan Login Sistem	50
4.6 <i>Sprint 2</i> Menu <i>Data Pegawai</i> dan Menu <i>Data Pelanggan</i>	54
4.6.1 Fitur Master Pegawai.....	57
4.6.2 Fitur Master Pelanggan	64
4.7 <i>Sprint 3</i> Menu <i>Cases</i> dan Menu <i>Tasks</i>	70
4.7.1 Menu <i>Cases</i>	70
4.7.2 Menu <i>Tasks</i>	77
4.8 <i>Sprint 4</i> Menu <i>Leads</i> dan Menu <i>Opportunities</i>	83
4.8.1 Menu <i>Leads</i>	83
4.8.2 Menu <i>Opportunities</i>	90
4.9 <i>Sprint 5</i> Menu <i>Email Reminder</i> dan Menu <i>Chatting</i>	98
4.9.1 Menu <i>Email Reminder</i>	98
4.9.2 Menu <i>Chatting</i>	102
BAB V KESIMPULAN DAN SARAN	107
5.1 Kesimpulan	107
5.2 Saran	108
DAFTAR PUSTAKA	109

DAFTAR GAMBAR

Gambar 3.1 Struktur Organisasi PT. Rabana Sejahtera Indonesia.....	38
Gambar 3.2 <i>Use Case Diagram</i> Sistem yang berjalan	41
Gambar 3.3 <i>Activity Diagram</i> Proses Konsultasi	42
Gambar 3.4 <i>Activity Diagram</i> Proses Pembelian.....	43
Gambar 3.5 Metodologi Penelitian.....	44
Gambar 4.1 <i>Use Case Diagram</i> Sistem Informasi CRM	47
Gambar 4.2 <i>Component diagram</i> <i>login</i> sistem	51
Gambar 4.3 <i>Package diagram</i> <i>core</i> sistem.....	52
Gambar 4.4 <i>Deployment diagram</i> <i>core</i> sistem	53
Gambar 4.5 <i>User Interface</i> Login sistem	54
Gambar 4.6 <i>User Interface</i> Menu Sistem Informasi CRM	54
Gambar 4.7 <i>Login</i> sistem	56
Gambar 4.8 Menu sistem informasi CRM.....	56
Gambar 4.9 <i>Activity Diagram</i> Master Data Pegawai	58
Gambar 4.10 <i>Class Diagram</i> Master Data Pegawai.....	59
Gambar 4.11 <i>Component Diagram</i> Master Data Pegawai	59
Gambar 4.12 <i>User Interface List</i> Master data pegawai	60
Gambar 4.13 <i>User Interface Form</i> Tambah data pegawai	60
Gambar 4.14 Fitur <i>List</i> data pegawai.....	62
Gambar 4.15 Fitur Form Tambah data pegawai.....	63
Gambar 4.16 Fitur Form Edit data pegawai.....	63
Gambar 4.17 <i>Activity Diagram</i> Master Data Pelanggan	65
Gambar 4.18 <i>Class Diagram</i> Master Data Pelanggan	66
Gambar 4.19 <i>Component Diagram</i> Master Data Pelanggan.....	66
Gambar 4.20 <i>User Interface List</i> Master data Pelanggan	67
Gambar 4.21 <i>User Interface Form</i> Tambah dan Edit data pelanggan	67
Gambar 4.22 Fitur <i>List</i> data pelanggan	69

Gambar 4.23 Fitur Form Tambah data pelanggan	70
Gambar 4.24 Fitur Form Edit data pelanggan	70
Gambar 4.25 <i>Activity Diagram</i> Menu <i>Cases</i>	71
Gambar 4.26 <i>Class Diagram</i> Menu <i>Cases</i>	72
Gambar 4.27 <i>Component Diagram</i> Menu <i>Cases</i>	73
Gambar 4.28 <i>User Interface List</i> Menu <i>Cases</i>	73
Gambar 4.29 <i>User Interface Form</i> Tambah data <i>Cases</i>	74
Gambar 4.30 Fitur <i>List</i> Menu <i>Cases</i>	75
Gambar 4.31 Fitur Form Tambah data <i>Cases</i>	76
Gambar 4.32 Fitur Form Edit data <i>Cases</i>	76
Gambar 4.33 <i>Activity Diagram</i> Menu <i>tasks</i>	78
Gambar 4.34 <i>Class Diagram</i> Menu <i>tasks</i>	79
Gambar 4.35 <i>Component Diagram</i> Menu <i>tasks</i>	79
Gambar 4.36 <i>User Interface List</i> Menu <i>tasks</i>	80
Gambar 4.37 <i>User Interface Form</i> Tambah data <i>tasks</i>	80
Gambar 4.38 Fitur <i>List</i> Menu <i>tasks</i>	82
Gambar 4.39 Fitur <i>Form</i> Tambah data <i>tasks</i>	82
Gambar 4.40 Fitur <i>Form</i> Edit data <i>tasks</i>	83
Gambar 4.41 <i>Activity Diagram</i> Menu <i>Leads</i>	84
Gambar 4.42 <i>Class Diagram</i> Menu <i>Leads</i>	85
Gambar 4.43 <i>Component Diagram</i> Menu <i>Leads</i>	86
Gambar 4.44 <i>User Interface List</i> Menu <i>Leads</i>	86
Gambar 4.45 <i>User Interface Form</i> Tambah data <i>Leads</i>	86
Gambar 4.46 Fitur <i>List</i> Menu <i>Leads</i>	88
Gambar 4.47 Fitur Form Tambah data <i>Leads</i>	89
Gambar 4.48 Fitur Form Edit data <i>Leads</i>	89
Gambar 4.49 <i>Activity Diagram</i> Menu <i>Opportunities</i>	91
Gambar 4.50 <i>Class Diagram</i> Menu <i>Opportunities</i>	92
Gambar 4.51 <i>Component Diagram</i> Menu <i>Opportunities</i>	93

Gambar 4.52 <i>User Interface List Menu Opportunities</i>	93
Gambar 4.53 <i>User Interface Form Tambah data Opportunities</i>	94
Gambar 4.54 <i>Fitur List Menu Opportunities</i>	96
Gambar 4.55 <i>Fitur Form Tambah data Opportunities</i>	96
Gambar 4.56 <i>Fitur Form Edit data Opportunities</i>	97
Gambar 4.57 <i>Activity Diagram Menu Email Reminder</i>	98
Gambar 4.58 <i>Class Diagram Menu Email Reminder</i>	99
Gambar 4.59 <i>Component Diagram Menu Email Reminder</i>	99
Gambar 4.60 <i>User Interface Menu Email</i>	100
Gambar 4.61 <i>Menu Email Reminder</i>	101
Gambar 4.62 <i>Activity Diagram Menu chatting</i>	102
Gambar 4.63 <i>Class Diagram Menu chatting</i>	103
Gambar 4.64 <i>Component Diagram Menu chatting</i>	103
Gambar 4.65 <i>User Interface Menu Chatting</i>	104
Gambar 4.66 <i>Menu Chatting</i>	106

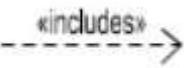
DAFTAR TABEL

Tabel 1.1 Produk Utama ekspor obat Herbal Indonesia	2
Tabel 2.1 Daftar Penelitian Terdahulu	34
Tabel 3.1 Rekap Penjualan Tahun 2018	40
Tabel 4.1 Scrum Team PT. Rabana Sejahtera Indonesia	49
Tabel 4.2 <i>Tools</i> yang digunakan untuk pengembangan	50
Tabel 4.3 <i>User Story core</i> sistem dan <i>login</i> sistem	51
Tabel 4.4 <i>Sprint execution core</i> sistem dan <i>login</i> sistem.....	55
Tabel 4.5 <i>Sprint review core</i> sistem dan <i>login</i> sistem.....	55
Tabel 4.6 <i>Sprint retrospective core</i> sistem dan <i>login</i> sistem	57
Tabel 4.7 <i>User Story</i> fitur master Pegawai	57
Tabel 4.8 Tabel Data Master Pegawai.....	61
Tabel 4.9 <i>Sprint execution</i> fitur master Pegawai	61
Tabel 4.10 <i>Sprint review</i> fitur master Pegawai	62
Tabel 4.11 <i>Sprint retrospective</i> fitur master Pegawai	63
Tabel 4.12 <i>User Story</i> fitur master pelanggan	64
Tabel 4.13 Tabel Data Master Pelanggan	68
Tabel 4.14 <i>Sprint execution</i> fitur master pelanggan	68
Tabel 4.15 <i>Sprint review</i> fitur master pelanggan	69
Tabel 4.16 <i>User Story</i> Menu <i>Cases</i>	71
Tabel 4.17 Tabel Data Menu <i>Cases</i>	74
Tabel 4.18 <i>Sprint execution</i> menu <i>cases</i>	74
Tabel 4.19 <i>Sprint review</i> menu <i>cases</i>	75
Tabel 4.20 <i>Sprint retrospective</i> fitur menu <i>cases</i>	77
Tabel 4.21 <i>User Story</i> Menu <i>tasks</i>	77
Tabel 4.22 Tabel Data Menu <i>tasks</i>	81
Tabel 4.23 <i>Sprint execution</i> menu <i>tasks</i>	81
Tabel 4.24 <i>Sprint review</i> menu <i>tasks</i>	81

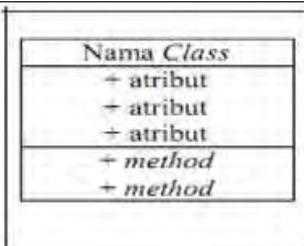
Tabel 4.25 <i>Sprint retrospective</i> fitur menu <i>tasks</i>	83
Tabel 4.26 <i>User Story</i> Menu <i>Leads</i>	84
Tabel 4.27 Tabel Data Menu <i>Leads</i>	87
Tabel 4.28 <i>Sprint execution</i> menu <i>Leads</i>	87
Tabel 4.29 <i>Sprint review</i> menu <i>Leads</i>	88
Tabel 4.30 <i>Sprint retrospective</i> fitur menu <i>Leads</i>	90
Tabel 4.31 <i>User Story</i> Menu <i>Opportunities</i>	90
Tabel 4.32 Tabel Data Menu <i>Opportunities</i>	94
Tabel 4.33 <i>Sprint execution</i> menu <i>Opportunities</i>	95
Tabel 4.34 <i>Sprint review</i> menu <i>Opportunities</i>	95
Tabel 4.35 <i>Sprint retrospective</i> fitur menu <i>Opportunities</i>	97
Tabel 4.36 <i>User Story</i> menu <i>email reminder</i>	98
Tabel 4.37 <i>Sprint execution</i> menu <i>email reminder</i>	100
Tabel 4.38 <i>Sprint review</i> menu <i>email reminder</i>	101
Tabel 4.39 <i>Sprint retrospective</i> menu <i>email reminder</i>	102
Tabel 4.40 <i>User Story</i> Menu <i>chatting</i>	102
Tabel 4.41 Tabel Data Menu <i>Chatting</i>	104
Tabel 4.42 <i>Sprint execution</i> menu <i>chatting</i>	105
Tabel 4.43 <i>Sprint review</i> fitur <i>chatting</i>	105
Tabel 4.44 <i>Sprint retrospective</i> fitur <i>chatting</i>	106

DAFTAR SIMBOL

Use Case Diagram

Simbol	Keterangan
	Actor : Merepresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. <i>Actor</i> hanya berinteraksi dengan <i>use case</i> tetapi tidak memiliki kontrol atas <i>use case</i> .
	Use Case : Adalah gambaran fungsionalitas dari suatu sistem, sehingga <i>customer</i> atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
	Association : Menghubungkan <i>link</i> antar elemen.
	Include : Yaitu kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, di mana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya.

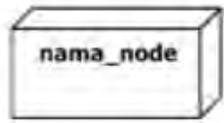
Class Diagram

Simbol	Keterangan
	Class : <i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut <i>class</i> . Bagian akhir mendefinisikan method-method dari sebuah <i>class</i> .
	Association : Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> . (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>).
	Composition : Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung

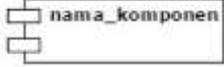
Class Diagram

	berbentuk jajaran genjang berisi/solid.
	Dependency : Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.
	Aggregation : <i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi.

Deployment Diagram

Simbol	Keterangan
	Node : digunakan untuk menggambarkan infrastruktur apa saja yang terdapat pada sistem. Biasanya <i>node</i> digambarkan sebagai server, pc, dan lain-lain.
	Communicates : digunakan untuk menghubungkan antar node yang saling berinteraksi.
	Dependency : digunakan untuk menggambarkan hubungan ketergantungan antar node atau komponen yang saling ketergantungan.

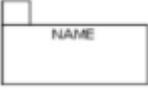
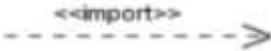
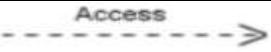
Component Diagram

Simbol	Keterangan
	Package : digunakan untuk menyimpan beberapa komponen
	Component : digunakan untuk mendeklarasikan komponen sistem.
	Communicates : digunakan untuk menghubungkan antar node yang saling berinteraksi.
	Dependency : digunakan untuk menggambarkan hubungan ketergantungan antar node atau komponen yang saling ketergantungan.

Activity Diagram

Simbol	Keterangan
	status awal : aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	aktivitas : yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	decision : Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.
	join : Asosiasi penggabungan di mana lebih dari satu dari aktivitas digabungkan menjadi akhir.
	Status akhir : yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Package Diagram

Simbol	Keterangan
	Package : adalah sekelompok elemen-elemen model
	Import : adalah suatu <i>dependency</i> yang mengindikasikan isi tujuan paket secara umum yang ditambahkan ke dalam sumber paket
	Access : adalah suatu <i>dependency</i> yang mengindikasikan isi tujuan paket secara umum yang bisa digunakan pada nama sumber paket

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di negara-negara sedang berkembang, sebagian besar penduduknya masih terus menggunakan obat tradisional, terutama untuk pemenuhan kebutuhan kesehatan dasarnya. Menurut resolusi *Promoting the Role of Traditional Medicine in Health Sistem : Strategy for the African Region*, sekitar 80% masyarakat di negara–negara anggota WHO (*World Health Organization*) di Afrika menggunakan obat tradisional untuk keperluan kesehatan. Beberapa negara Afrika melakukan pelatihan obat tradisional kepada farmasi, dokter dan para medik. Demikian pula penggunaan obat tradisional di Asia, terus meningkat meskipun banyak tersedia dan beredar obat-obat entitas kimia. Di RRC (Republik Rakyat China), penggunaan obat tradisional mencapai 90% penduduk di Jepang 60 sampai dengan 70% dokter meresepkan obat tradisional "kampo" untuk pasien.

Di Malaysia, obat tradisional Melayu, TCM dan obat tradisional India digunakan secara luas oleh masyarakatnya. Sementara itu, Kantor Regional WHO wilayah Amerika (*AMOR/PAHO*) melaporkan 71% penduduk Chile dan 40% penduduk Kolombia menggunakan obat tradisional. Di negara-negara maju, penggunaan obat tradisional tertentu sangat populer. Beberapa sumber menyebutkan penggunaan obat tradisional oleh penduduk di Perancis mencapai 49%, Kanada 70%, Inggris 40% dan Amerika Serikat 42%.

Beberapa produk ekstrak herbal mempunyai pasar global dengan nilai yang besar. *Ginko Biloba*, Ginseng, Garlic dan *Echinacae* adalah ekstrak yang memiliki pasar tergolong terbesar di dunia. Di Amerika Serikat, penjualan dan penggunaan obat herbal berupa *dietary supplement* juga meningkat cukup signifikan. Obat herbal Indonesia pada dasarnya dapat dikelompokkan dalam tiga kategori, yaitu :

1. Jamu;
2. Obat Herbal Terstandar; dan
3. Fitofarmaka.

Jamu sebagai warisan budaya bangsa perlu terus dikembangkan dan dilestarikan dengan fokus utama pada aspek mutu dan keamanannya (*safety*). Khasiat jamu sebagai obat herbal selama ini didasarkan pengalaman empirik yang telah berlangsung dalam kurun waktu yang sangat lama.

Menurut Badan Pusat Statistik Indonesia Nilai ekspor obat herbal Indonesia tahun 2013 mencapai US\$ 23,44 juta, sedangkan nilai ekspor pada periode Januari-Juni 2014 sebesar US\$ 29,13 juta, mengalami peningkatan 600% dari nilai ekspor pada periode Januari-Juni 2013. Pertumbuhan ekspor obat herbal Indonesia selama periode 2009- 2013 mengalami kenaikan sebesar 6,49% per tahun.

Produk utama ekspor obat herbal pada periode Januari-Juni 2014 adalah jahe (HS 091010) dengan nilai ekspor sebesar US\$ 25,8 juta, dengan pangsa ekspor sebesar 88,58% dari total ekspor obat herbal Indonesia. Rempah-rempah lainnya (HS 091099) dengan nilai ekspor sebesar US\$ 1,84 juta (6,33%), dan *curcuma* (HS 091030) dengan nilai ekspor US\$ 699 ribu (2,4%).

Negara tujuan ekspor obat herbal Indonesia pada periode Januari-Juni 2014 adalah Bangladesh dengan nilai US\$ 10,94 juta (pangsa ekspor obat herbal 37,55%), Pakistan US\$ 10,71 juta (36,76%), Malaysia US\$ 2,67 juta (9,17%), Vietnam sebesar US\$ 1,19 juta (4,12%) dan Jepang sebesar US\$ 806 ribu (2,77%), tabel 1.1 menjelaskan lebih detail tentang jenis obat herbal yang mempunyai nilai ekspor tinggi.

Tabel 1.1 Produk Utama ekspor obat Herbal Indonesia

No	Commodity	2013			% Trend 2009-2013	Jan-Jun 2014			% Change 2014/2013	
		Tons	US\$ 000	% Share		Tons	US\$ 000	% Share	Quantity	Value
1	Ginger	22.472	14.908	63,59	22,44	33.922	25.809	88,58	2.176,81	1.616,85
2	Spices, Nesol	1.188	4.343	18,52	-12,02	563	18,44	6,33	50,95	43,54
3	Tumeric (Curcuma)	1.947	2.101	8,96	-16,17	444	699	2,40	18,11	21,36
4	Mixtures Of Two Or More Spices Provided For Separately In Different Headings Of This Chapter	729	1.603	6,84	67,15	220	620	2,13	11,70	2,63
5	Saffron	794	490	2,09	47,34	274	166	0,57	-9,87	-13,40
	TOTAL	27.129	23.446	100,00	6,49	35.422	29.137	100,00	1.193.15	500,54

Sumber: Badan Pusat Statistik Indonesia

Perkembangan teknologi dan informasi saat ini semakin hari semakin berkembang ke berbagai sektor, khususnya dalam sebuah bisnis obat-obatan herbal yang semakin hari

semakin ketat. Sebuah perusahaan dituntut agar dapat memberikan pelayanan yang baik kepada pelanggan, karena pelanggan merupakan aset yang sangat penting dalam sebuah perusahaan, sehingga perusahaan harus melakukan strategi yang tepat agar dapat bersaing dalam dunia bisnis. Salah satu cara yang dapat dilakukan perusahaan agar mampu bersaing dalam dunia bisnis adalah dengan menerapkan sebuah sistem *Customer Relationship Management*. CRM merupakan salah satu sarana dalam menjaga hubungan baik antara perusahaan dengan pelanggan yang berfokus pada jangka panjang dan hubungan yang berkelanjutan. Melalui CRM perusahaan dapat memberikan kepuasan kepada pelanggan.

PT. Rabana Sejahtera Indonesia merupakan salah satu badan usaha milik swasta yang terletak di Kecamatan Klagenan, Kabupaten Cirebon. Perusahaan ini menjual obat herbal untuk berbagai penyakit. Adapun produk yang dijual lebih kurang berjumlah 100 item seperti obat jantung, obat penyakit paru-paru, obat metabolisme dan lain-lain. Setiap harinya pelanggan yang memesan produk obat herbal ini mencapai ratusan pesanan. Dengan jumlah karyawan yang terbatas dan jumlah pelanggan yang lumayan banyak serta ada pelanggan tetap yang menyebabkan karyawan perusahaan mengalami kesulitan dalam melayani pembelian dan promosi terhadap pelanggan. Hal ini disebabkan proses pelayanan yang masih manual, dan mengakibatkan pengelolaan hubungan dengan pelanggan menjadi kurang efektif.

Selanjutnya untuk pengembangan sebuah bisnis penjualan maka dibutuhkan pelanggan yang dapat meningkatkan kemajuan usaha. Saat ini upaya untuk melakukan pengelolaan dengan pelanggan terutama pelanggan tetap masih belum ada. Hal ini tentu akan membuat perusahaan kehilangan kesempatan untuk dapat mempertahankan pelanggan tetap dan loyal. Oleh karena itu, perlu ada sistem informasi yang dapat meningkatkan pengelolaan hubungan dengan pelanggan sehingga pelanggan tersebut puas dengan layanan dan akhirnya bisa menjadi pelanggan tetap yang loyal dan selalu bisa dipertahankan.

Berdasarkan Uraian di atas. Penelitian tesis ini mengambil judul "**PERANCANGAN SISTEM INFORMASI *CUSTOMER RELATIONSHIP MANAGEMENT* MENGGUNAKAN METODE *SCRUM* (STUDI KASUS : PT. RABANA SEJAHTERA INDONESIA)**".

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang yang dibahas di atas, maka dapat diambil sebuah rumusan permasalahan adalah sebagai berikut :

1. Bagaimana merancang sistem informasi *CRM* di PT. Rabana Sejahtera Indonesia dengan Metode *Scrum* ?
2. Bagaimana mengidentifikasi *service – service* sistem informasi *CRM* ?

1.3 Tujuan Penelitian

Berdasarkan permasalahan yang diteliti, maka tujuan penelitian yang akan dicapai adalah :

1. Membuat perancangan sistem informasi *CRM*.
2. Mengidentifikasi dan membuat *service – service* sistem informasi *CRM* dengan menggunakan Metode *Scrum*.

1.4 Batasan Masalah

Masalah-masalah yang akan dibahas sangat luas agar penyajian lebih terarah dan mencapai sasaran yang ditentukan, maka diperlukan suatu pembatasan masalah atau ruang lingkup kajian yang meliputi hal-hal sebagai berikut :

1. *CRM* yang diterapkan pada sistem ini adalah *CRM* analitis yang terfokus pada kegiatan penggalan data konsumen untuk tujuan strategis dan meningkatkan profit.
2. Metodologi yang digunakan adalah *Scrum Framework* dengan pendekatan rekayasa perangkat lunak *Agile Development methods*.
3. Penelitian ini dikhususkan untuk kepentingan di lingkungan PT. Rabana Sejahtera Indonesia

1.5 Hasil Penelitian

Hasil yang diharapkan dari kajian penelitian ini adalah :

1. Mendapatkan gambaran dan acuan membuat sistem informasi *CRM* yang efektif.
2. Dapat memberikan kontribusi berupa rekomendasi atau usulan untuk peneliti lain dalam membuat perancangan sistem informasi *CRM* dengan model *scrum*.

1.6 Metode Penelitian

Metodologi penelitian ini dilakukan secara sistematis yang dapat digunakan sebagai pedoman untuk peneliti dalam melaksanakan penelitian agar hasil yang dicapai tidak menyimpang dan tujuan yang diinginkan dapat terlaksana dengan baik dan sesuai dengan tujuan yang telah ditetapkan sebelumnya, metode yang akan digunakan adalah studi literatur, analisis sistem dan perancangan sistem. Berikut adalah uraian metode yang digunakan:

1. Mendefinisikan masalah dengan merancang sistem informasi *CRM* dalam hal ini di PT. Rabana Sejahtera Indonesia.
2. Pengumpulan data-data yang diperlukan untuk perancangan sistem informasi, dengan cara observasi dan wawancara.
3. Melakukan analisis sistem informasi *CRM* untuk kebutuhan pelanggan.

1.7 Sistematika Penulisan

Sistematika penulisan laporan tesis ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi mengenai latar belakang masalah, rumusan masalah, tujuan penelitian, batasan masalah, hasil penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi mengenai teori yang mendukung terhadap pembahasan masalah meliputi konsep dasar pemodelan, pengertian *CRM*, sistem informasi, *Agile Development Methods*, *Scrum*, *Unified Modeling Language (UML)*. Aplikasi dan bahasa pemrograman pembuatan sistem dan penelitian sebelumnya.

BAB III OBJEK DAN METODOLOGI PENELITIAN

Bab ini berisi uraian tentang metode dan analisis perancangan sistem informasi *CRM* yang akan dibuat.

BAB IV PEMBAHASAN DAN HASIL PENELITIAN

Bab ini berisi uraian tentang hasil analisis perancangan sistem yang mencakup *database*, pemodelan dan perancangan sistem informasi dengan UML dalam menyelesaikan permasalahan yang ada.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dan saran hasil penyusunan penelitian yang telah disusun yang sesuai dengan rumusan masalah yang telah dibuat sebelumnya.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem yang saling teintegrasi di dalam sebuah organisasi yang menjembatani kebutuhan pengolahan operasional harian, mendukung operasi bersifat teknis dan kegiatan strategis dari suatu organisasi dan menyediakan laporan-laporan yang diperlukan untuk pihak luar (Jogiyanto, 2011).

Sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang saling terhubung berfungsi untuk mengolah data menggunakan perangkat keras dan perangkat lunak tersebut sehingga menghasilkan informasi yang bermanfaat (Kristanto, 2008).

Menurut Sutabri (2019) Sistem informasi adalah suatu sistem yang saling berkaitan dalam memproses mengolah kebutuhan pengolahan data harian yang mendukung fungsi manajerial di dalam operasionalnya dengan mengedepankan aspek strategis dalam sebuah organisasi yang akan menghasilkan laporan-laporan tertentu yang dibutuhkan untuk pihak luar.

Adapun komponen-komponen sistem informasi menurut Kristanto (2008) adalah sebagai berikut :

1. *Input*

Input adalah semua data yang akan dimasukkan dan akan diolah ke dalam sistem informasi.

2. *Proses*

Proses adalah merupakan sekumpulan tata cara atau algoritma yang akan memanipulasi masukan yang kemudian akan disimpan dalam basis data dan seterusnya akan diolah menjadi suatu keluaran yang akan digunakan oleh pengguna.

3. *Output*

Output merupakan semua keluaran atau hasil yang mempunyai nilai dari model yang sudah diolah menjadi suatu informasi yang berguna dan dapat dipakai pengguna.

4. Teknologi

Teknologi di sini adalah bagian yang berfungsi untuk memasukan *input*, mengolah *input* kemudian memproses *inputan* dari pengguna tersebut dan menghasilkan keluaran (*output*).

5. Basis data

Basis data merupakan kumpulan data-data yang saling berkaitan satu dengan yang lain yang terletak dalam perangkat keras komputer seperti *harddisk* dan akan diolah kembali dengan menggunakan perangkat lunak.

6. Kendali

Kendali merupakan semua tindakan yang diambil dan sudah dipertimbangkan dengan matang untuk menjaga sistem informasi tersebut agar bisa berjalan dengan lancar dan tidak mengalami gangguan karena penggunaan yang salah.

2.2 Customer Relationship Management

Menurut Buttle (2015) *Customer Relationship Management (CRM)* adalah sistem informasi terintegrasi mencakup semua aspek yang berurusan dengan prospek dengan pelanggan, termasuk pusat panggilan, tenaga penjualan, pemasaran, dukungan teknis dan layanan lapangan. Tujuan utama *CRM* adalah untuk meningkatkan jangka panjang pertumbuhan dan profitabilitas melalui pemahaman yang lebih baik tentang perilaku pelanggan. *CRM* bertujuan untuk memberikan umpan balik yang lebih efektif dan peningkatan integrasi untuk mengukur dengan lebih baik laba atas investasi (ROI) di daerah ini.

CRM adalah pendekatan terintegrasi untuk mengidentifikasi, memperoleh dan mempertahankan pelanggan. Organisasi untuk mengelola dan mengoordinasikan interaksi pelanggan lintas beragam saluran, departemen, lini bisnis dan geografi, *CRM* membantu organisasi memaksimalkan nilai setiap interaksi pelanggan dan mendorong perusahaan mempunyai kinerja yang lebih unggul.

Kerangka komponen *CRM* diklasifikasikan menjadi tiga: (Buttle, 2015) yaitu:

1. *CRM* Operasional

CRM Operasional mengotomatiskan proses bisnis yang dihadapi pelanggan. Aplikasi perangkat lunak *CRM* memungkinkan fungsi pemasaran, penjualan, dan layanan otomatis menjadi saling terintegrasi.

2. *CRM* Analitis

CRM Analitis adalah *CRM* yang berkaitan dengan menangkap, menyimpan, mengekstraksi, mengintegrasikan, memproses, menafsirkan, mendistribusikan, menggunakan dan melaporkan data terkait pelanggan untuk meningkatkan nilai pelanggan dan perusahaan.

CRM Analitis dibangun atas dasar informasi yang terkait dengan pelanggan. Data pelanggan terkait dapat ditemukan dalam repositori seluruh perusahaan: data penjualan (riwayat pembelian), data keuangan (riwayat pembayaran, skor kredit), data pemasaran (respons promosi, loyalitas data skema) dan data layanan. Untuk data internal ini dapat ditambahkan data dari sumber eksternal: data *geo-demografis* dan gaya hidup dari organisasi bisnis yang kompeten, misalnya. Ini biasanya adalah data terstruktur yang disimpan dalam database relasional. *Database* relasional adalah seperti *spreadsheet Excel* di mana semua data dalam baris apa pun adalah tentang pelanggan tertentu, dan kolom melaporkan variabel tertentu seperti nama, kode pos, dan sebagainya.

3. *CRM* Strategis

CRM Strategis difokuskan pada pengembangan budaya bisnis yang berpusat pada pelanggan didedikasikan untuk memenangkan dan menjaga pelanggan dengan menciptakan dan memberikan nilai lebih baik daripada pesaing. Budaya tercermin dalam perilaku kepemimpinan, desain sistem formal perusahaan, dan mitos dan cerita yang dibuat di dalam perusahaan. Dalam pelanggan- budaya sentral perusahaan akan mengharapkan sumber daya dialokasikan di mana mereka akan meningkatkan pelayanan terbaik kepada pelanggan, sistem penghargaan untuk mempromosikan perilaku karyawan yang meningkatkan kepuasan dan retensi pelanggan, dan informasi

pelanggan untuk dikumpulkan, dibagikan, dan diterapkan di seluruh bisnis. Faktor bisnis yang berpusat pada pelanggan memberikan nilai luar biasa atau layanan kepada pelanggan. Banyak bisnis mengklaim sebagai *customer-centric*, *customer-led*, *customer-focuss* atau berorientasi pelanggan tetapi memang hanya ada sedikit perusahaan yang mengukur dan mengklaim bahwa mereka sedang dalam misi untuk memenuhi persyaratan menguntungkan pelanggan. Sentralisasi pelanggan bersaing dengan logika bisnis lainnya. *CRM* mengidentifikasi tiga hal utama lainnya pada orientasi bisnis yaitu produk, produksi dan penjualan.

2.2.1 Definisi Customer

Customer adalah semua orang yang berhak mendapatkan pelayanan dari organisasi atau perusahaan dengan suatu standar kualitas tertentu, dan karena itu akan memberikan pengaruh pada performansi perusahaan sehingga dapat meningkatkan kepercayaan pelanggan terhadap perusahaan (Nasution, 2010).

Customer adalah orang yang paling diharapkan kedatangannya dalam suatu perusahaan. *Customer* tidak bergantung kepada perusahaan tetapi perusahaan bergantung kepada *customer*. *Customer* bukan menerima pekerjaan tetapi *customer* yang memberikan pekerjaan. Pembeli bukan seorang untuk menilai atau menghitung tetapi *customer* adalah seorang yang mengungkapkan. Apa yang diinginkan perusahaan adalah untuk memberikan kepuasan untuk *customer* dan untuk perusahaan itu sendiri (Nasution, 2010).

2.2.2 Pelayanan

Pelayanan merupakan hal yang paling penting dalam usaha untuk membuat pelanggan tetap merasa tenang dalam menggunakan jasa atau produk dari suatu perusahaan, jika perusahaan memberikan pelayanan yang baik maka pelanggan tidak akan pergi menggunakan jasa atau produk dari perusahaan lain. Adapun definisi pelayanan menurut Majid (2009), pelayanan adalah suatu tindakan nyata dan segera untuk

menolong orang lain (pelanggan, mitra kerja, mitra bisnis, dan sebagainya), disertai dengan senyuman yang hangat dan tulus.

2.2.3 Definisi Keluhan Pelanggan

Menurut Majid (2009) Mengeluh atau komplain adalah sesuatu yang manusiawi sebagai bentuk reaksi atas apa yang diterima yang tidak sesuai dengan apa yang telah dijanjikan dan diharapkan sebelumnya. Beberapa hal yang menyebabkan *customer* mengeluh, diantaranya:

1. Pelayanan yang diharapkan dari perusahaan tidak seperti yang *customer* harapkan.
2. *Customer* diacuhkan, misalnya dibiarkan menunggu tanpa ada kejelasan.
3. Tidak ada yang mau mendengarkan keluh kesan pelanggan.
4. Pegawai perusahaan berlaku tidak sopan atau tidak membantu pelanggan.
5. Tidak ada yang mau bertanggung jawab untuk suatu kesalahan yang dilakukan pegawai kepada pelanggan.
6. Ada kegagalan komunikasi antara pelanggan dan perusahaan.
7. Dan lain-lainnya.

Menurut Irawan (2002) Dalam menangani komplain, perusahaan harus mengucapkan terima kasih karena pelanggan yang komplain masih memberikan kesempatan kepada perusahaan untuk melakukan perbaikan kedepan agar kesalahan yang telah dilakukan tidak terjadi lagi dikemudian hari. Orang yang menangani komplain adalah *front-line staff*.

2.2.4 Definisi Pelayanan Prima

Dalam upaya untuk mempertahankan pelanggan dan meningkatkan loyalitas pelanggan di sebuah perusahaan memiliki cara tersendiri untuk menangani setiap masalah tersebut salah satunya dengan melakukan pelayanan prima yaitu bagaimana cara membuat pelanggan merasa dihargai dan diistimewakan. Seperti yang dikutip dalam Majid (2009) mengatakan bahwa pelayanan prima atau *Service Excellence* merupakan pengembangan dari *Customer Care* yang lebih efektif, dalam memberikan layanan terbaik

bagi pelanggan dan harus memberikan benefit kepada pelanggan, sehingga kita mampu memberikan kepuasan yang optimal.

Menurut Rusydi (2017) Pelayanan Prima adalah pelayanan yang sangat baik atau terbaik atau pelayanan yang sesuai dengan standar pelayanan yang berlaku atau dimiliki oleh instansi yang memberikan pelayanan. Bisa juga diterjemahkan sebagai pelayanan yang memuaskan pelanggan.

Konsep pelayanan prima akan tercapai bila keinginan pelanggan terpuaskan yaitu produk pelayanan yang diterima bermutu tinggi dan mutu diupayakan melalui penetapan standar pelayanan yang sudah disepakati oleh lembaga yang mengaturnya.

Akan tetapi terdapat beberapa rintangan yang terjadi pada organisasi atau perusahaan dalam mendengarkan permintaan pelanggan, seperti yang dikutip oleh Cook (2004) bahwa terdapat beberapa faktor penghambat organisasi menjadi dekat dengan pelanggannya yaitu:

1. Pendapat dari pelanggan yang mengajukan komplain tidak dibandingkan dengan pendapat dari pelanggan yang tidak mengajukan komplain, padahal sebagian besar pelanggan tidak mengajukan keluhan, pada saat terjadi komplain baiknya perusahaan langsung menganalisa komplain tersebut..
2. Opini dari kelompok pelanggan vokal yang sering menyebabkan organisasi atau perusahaan mengabaikan kebutuhan pelanggan yang lebih luas yang menyebabkan kurang tergalinya opini dari beberapa pelanggan lain.
3. Adanya prasangka dalam organisasi, khususnya terkait dengan sindrom “kami telah melakukannya dengan benar dan ini yang diinginkan oleh pelanggan” yang mana akan membuat perusahaan kurang dalam berinovasi.

2.3 Agile Development Method

Menurut Pressman (2012), *Agile Software Development* adalah sekumpulan metodologi pengembangan perangkat lunak yang berbasis pada pengembangan berkelanjutan, di mana persyaratan dan solusi berkembang melalui kolaborasi antar tim yang terorganisir. Istilah ini diciptakan pada tahun 2001 ketika *Agile Manifesto* dirumuskan.

Metode *Agile* umumnya mempromosikan disiplin proses manajemen proyek yang mendorong inspeksi dan adaptasi; filosofi kepemimpinan yang mendorong kerja sama dalam tim, pengorganisasian dan akuntabilitas; praktek rekayasa yang memungkinkan pengiriman perangkat lunak berkualitas tinggi dengan cepat; dan pendekatan bisnis yang sejalan dengan pengembangan kebutuhan pelanggan dan tujuan perusahaan.

Menurut Sarosa (2017), *Agile* dapat diterjemahkan ke dalam Bahasa Indonesia sebagai tangkas (bahasa Jawa : trengginas), *Agile* sendiri sebenarnya memiliki arti cepat dan terkoordinir dengan baik. *Agile* memiliki sifat dapat beradaptasi dengan cepat. Terkait dengan pengembangan sistem informasi, *Agile* memiliki karakteristik yang sangat berorientasi pada kecepatan penyelesaian sistem informasi maupun perangkat lunak sebagai sebagai suatu produk.

Agile methods memiliki banyak model yang bisa digunakan oleh pengembang, diantaranya sebagai berikut :

1. *Extreme Programming (XP)*
2. *Feature Driven Development (FDD)*
3. *Dynamic Sistem Development Method (DSDM)*
4. *Adaptive Software Development (ASD)*
5. *Scrum Methodology*
6. *Agile Modeling (AM)*
7. *Lean Software Development (LSD)*

2.3.1 Model Scrum

Menurut Sarosa (2017) *Scrum* berasal dari istilah dalam olahraga *Rugby* untuk memulai kembali pertandingan setelah dihentikan oleh wasit. Dalam *Scrum* team pengembang (seperti *team Rugby*) merupakan satu *team* yang multi guna dan bersama-sama mencapai suatu tujuan. *Team* pengembang harus berfungsi sebagai satu kesatuan yang utuh dalam menuntaskan pekerjaan.

Menurut Sutherland (2017) *Scrum* adalah Sebuah kerangka kerja di mana orang-orang dapat mengatasi masalah yang sangat rumit bersifat adaptif, di mana pada saat

yang bersamaan pula *Scrum Team* juga memberikan produk dengan nilai setinggi mungkin secara berkelanjutan dan kreatif. *Scrum* bersifat :

1. Ringan dan fleksibel
2. Sederhana dan mudah untuk dipahami
3. Sulit untuk diimplementasikan

Menurut Sutherland (2017) *Scrum* adalah kerangka kerja dalam memetakan beberapa aktifitas yang telah digunakan untuk mengelola pengembangan produk dengan tingkat kerumitan yang cukup tinggi sejak awal tahun 1990-an. *Scrum* bukanlah sebuah proses ataupun teknik. Akan tetapi *Scrum* adalah sebuah kerangka kerja di mana *scrum team* dapat mengkombinasikan berbagai proses dan teknik di dalamnya. *Scrum* menitikberatkan pada efektifitas dari manajemen produk dan teknik kerja *scrum team*, sehingga *scrum team* dapat secara terus-menerus meningkatkan kinerja dalam membuat produk, tim, dan lingkungan kerja *scrum team*.

Kerangka kerja *Scrum* terdiri dari *Scrum Team* seperti *Pemilik Produk*, *Scrum Master* dan *Tim pengembang*, acara-acara *Scrum* seperti *daily Scrum*, *sprint planning*, *sprint review*, *sprint retrospective*, artefak-artefak seperti *product backlog*, *sprint backlog* dan aturan-aturan terkait. Setiap bagian di dalam kerangka kerja *Scrum* memiliki fungsi tertentu dan sangat penting agar penggunaan *Scrum* bisa diterapkan dengan baik.

2.3.2 Penggunaan *Scrum*

Menurut Sutherland (2017) *Scrum* dibentuk untuk mengelola dan mengembangkan produk yang mengedepankan produktifitas tinggi. *Scrum* mulai dipergunakan di seluruh dunia, untuk:

1. Meneliti hal baru dan menggali potensi *market*, teknologi, dan kemampuan produk;
2. Mengembangkan produk dan meningkatnya nilai produk tersebut;
3. Merilis produk dan dengan mutu yang terus meningkat, sesering mungkin di setiap hari untuk memenuhi permintaan *market*;

4. Mengembangkan dan memelihara operasional sistem *cloud computing* (daring, keamanan, sesuai permintaan pengguna) dan lingkungan operasional lain untuk penggunaan produk sistem informasi; dan,
5. Mengelola dan memperbarui sebuah produk agar selalu sesuai dengan kebutuhan pengguna produk.

Scrum digunakan untuk mengembangkan perangkat lunak, perangkat keras, perangkat lunak yang saling terintegrasi, aplikasi dalam sistem jaringan yang saling terkoneksi, kendaraan tanpa awak, sekolah dan pusat pendidikan, pemerintahan, pemasaran, mengelola operasional perusahaan dan hampir semua hal yang digunakan di kehidupan sehari-hari sebagai seorang individu dan masyarakat.

Kehandalan *Scrum* dalam menyelesaikan masalah yang sangat rumit semakin terbukti setiap harinya seiring dengan semakin meningkatnya kompleksitas dan interaksi antara teknologi, pasar dan lingkungan yang menuntut untuk bisa bersinergi dan *Scrum* lebih efektif dalam melakukan *sharing* pengetahuan secara bertahap dan berkelanjutan. *Scrum* saat ini sudah digunakan secara luas untuk produk dibidang sistem informasi maupun non sistem informasi, layanan-layanan yang membutuhkan aksesibilitas tinggi, dan manajemen inti dari sebuah perusahaan.

Menurut Sutherland (2017) Makna dari *Scrum* adalah sebuah tim kecil yang terdiri dari beberapa orang. Tim bersifat sangat fleksibel dan mampu beradaptasi dengan sangat cepat. Kekuatan ini terus berlanjut dalam satu tim, beberapa tim maupun banyak tim yang saling berhubungan dalam mengembangkan produk, merilis produk, mengoperasikan dan memelihara produk dan produk hasil pekerjaan dari ribuan orang diintegrasikan. *Scrum Team* bersinergi dan saling berinteraksi melalui struktur pengembangan dan target rilis produk yang terkini.

2.3.3 Teori Scrum

Menurut Sutherland (2017) *Scrum* dibentuk berdasarkan teori proses pengendalian empiris atau bisa disebut empirisme. Empirisme menyatakan bahwa pengetahuan datang dari pengalaman dan pengambilan keputusan didasari oleh apa yang telah diketahui dari

dulu sampai sekarang. *Scrum* menggunakan pendekatan yang bertahap dan berkelanjutan untuk mengoptimalkan kemampuan estimasi dan pengelolaan risiko. Tiga pilar yang paling penting yang memperkuat setiap implementasi dari empirisme adalah: transparansi, inspeksi dan adaptasi.

Transparansi, Aspek detail dari sebuah proses harus dapat dilihat oleh semua orang yang bertanggung jawab terhadap efek yang ditimbulkan. Transparansi membutuhkan aspek-aspek tersebut untuk menetapkan standar baku sehingga para pemangku kepentingan memiliki pemahaman dan perspektif yang sama terhadap apa yang sedang dikerjakan. Sebagai contoh:

1. Semua peserta harus memiliki pemahaman dan perspektif yang sama terkait istilah yang digunakan mengacu pada proses; dan,
2. *Scrum Team* yang melakukan pekerjaan dan memeriksa *Increment* yang mempunyai *value* harus memiliki definisi "Selesai" yang sama.

Inspeksi, Penggunaan *Scrum* harus selalu melakukan inspeksi terhadap artefak *Scrum* dan perkembangan menuju Tujuan *Sprint* agar *Scrum Team* dapat mendeteksi adanya perbedaan hasil yang tidak diinginkan. Proses inspeksi juga disarankan tidak dilakukan terlalu sering sampai menghambat pekerjaan. Inspeksi akan sangat bermanfaat jika dilakukan oleh pemeriksa atau *tester* yang kredibel di saat di mana pekerjaan tersebut sedang atau selesai dilakukan.

Adaptasi, Jika pemeriksa menemukan bahwa ada satu hal atau lebih dari proses yang tidak sesuai di luar batas yang bisa diterima yang dapat menyebabkan produk tidak bisa diterima, maka proses atau materi yang sedang diproses harus diubah secepatnya. Perubahan harus dilakukan secepatnya untuk meminimalkan penyimpangan yang semakin jauh dari tujuan awal.

Scrum memiliki empat acara formal untuk melakukan inspeksi dan adaptasi dari produk yang akan diciptakan, yakni:

1. *Sprint Planning*
2. *Daily Scrum*
3. *Sprint Review*

4. *Sprint Retrospective*

2.3.4 Tata Nilai *Scrum*

Menurut Sutherland (2017) Saat tata nilai dari komitmen, keberanian, fokus, keterbukaan, dan respek diwujudkan dan hidup di dalam *Scrum Team*, maka pilar-pilar *Scrum* seperti: Transparansi, Inspeksi dan Adaptasi akan menjadi lebih terasa dan membangun rasa percaya dan saling menghargai satu sama lain. *Scrum Team* harus memahami dan menerapkan tata nilai tersebut bersama dengan penggunaan aspek lain di dalam *Scrum* pada saat *Scrum Team* bekerja.

Keberhasilan penggunaan *Scrum* bergantung pada orang-orang yang semakin menjiwai kelima tata nilai tersebut. Setiap orang berkomitmen untuk menciptakan tujuan dari *Scrum Team*. Anggota *Scrum Team* memiliki keberanian untuk melakukan hal yang terbaik dan bekerja sama untuk menyelesaikan masalah-masalah yang sangat sulit. Semua pihak fokus pada masalah yang harus dihadapi di dalam *Sprint* dan tujuan dari *Scrum Team*. *Scrum Team* dan pemangku kepentingan setuju untuk terbuka terhadap semua pekerjaan dan tantangan di dalam menyelesaikan pekerjaan. Anggota *Scrum Team* saling peduli satu sama lain bahwasanya *Scrum Team* adalah orang-orang yang memiliki kemampuan yang spesifik dan kemandirian yang tidak dapat diintervensi.

2.3.5 *Scrum Team*

Menurut Sutherland (2017) *Scrum Team* terdiri dari *Product Owner*, *Development Team* dan *Scrum Master*. *Scrum Team* bersifat lintas-fungsi dan mengelola diri mereka sendiri secara mandiri. Tim yang mengelola diri mereka sendiri memilih cara terbaik dalam menciptakan solusi di *Scrum Team*, tidak ada yang dapat memerintah di luar *Scrum Team*.

Tim yang lintas-fungsi memiliki semua keahlian yang diperlukan untuk menyelesaikan masalah *Scrum Team* tanpa ketergantungan pada individu lain di luar *scrum team*. Bentuk *Scrum team* dirancang untuk mengoptimalkan efektifitas memproduksi sebuah produk dengan kreativitas tinggi. Bentuk *Scrum Team* terbukti

menciptakan kondisi tim yang cepat dalam mengerjakan semua jenis pekerjaan yang telah ditentukan sebelumnya dan untuk semua jenis pekerjaan serumit apapun.

2.3.5.1 Product Owner

Menurut Sarosa (2017) *Product Owner* atau *Pemilik Produk* adalah perwakilan dari para pemangku kepentingan sistem informasi. Tanggung jawab Pemilik Produk adalah memastikan bahwa hasil pengembangan sistem informasi memberikan nilai tambah (*value*). Pemilik produk mewakili para pengguna maka harus berorientasi pada sisi non teknis (bagian proses bisnis) dan tidak mencampuri urusan teknis (kode – kode program Sistem Informasi).

Fungsi utama Pemilik Produk adalah berkomunikasi. Pemilik Produk menjadi penghubung antara pemangku kepentingan dan *team* pengembang. Kemampuan Pemilik Produk dalam berkomunikasi harus disertai keahlian teknis pengembangan sistem informasi, khususnya kemampuan mengidentifikasi kebutuhan pemangku kepentingan, negosiasi prioritas kebutuhan antar para pemangku kepentingan, dan berkolaborasi dengan pengembang untuk memastikan implementasi yang efektif.

2.3.5.2 Development Team

Menurut Sarosa (2017) *Development Team* atau *Team* Pengembang bertanggung jawab untuk mewujudkan solusi atas kebutuhan pemangku kepentingan. Setiap *team* terdiri atas 3 (tiga) orang sampai dengan 9 (sembilan) orang pengembang sistem. Pekerjaan pengembangan yang dilakukan meliputi semua tahapan pengembangan sistem informasi mulai dari analisis, perancangan, pengembangan, uji coba, komunikasi, sampai dengan penyusunan dokumentasi.

Team pengembang bersifat lintas fungsi yang memiliki keahlian yang dibutuhkan untuk menghasilkan produk berupa sistem informasi. *Team* pengembang dalam *scrum* bersifat mengorganisir dirinya sendiri agar proses kolaborasi di dalam tim menjadi lebih efektif.

2.3.5.3 *Scrum Master*

Menurut Sutherland (2017) *Scrum Master* bertanggung jawab untuk mengenalkan dan mendukung penggunaan *Scrum* di sebuah organisasi berfungsi untuk membantu orang-orang agar dapat memahami teori, praktik-praktik, aturan-aturan dan tata nilai dari implementasi *Scrum*.

Menurut Partogi (2015) *Scrum Master* adalah seorang yang berperan dibidang manajerial yang memiliki dampak yang cukup luas di dalam perusahaan. Walaupun *scrum master* biasanya berasal dari departemen Teknologi Informasi, dampaknya bisa terasa sampai departemen lainnya.

Scrum Master adalah seseorang yang melayani *Scrum Team*. *Scrum Master* menjembatani orang-orang di luar *Scrum Team* untuk dapat memahami komunikasi dan interaksi mana yang bermanfaat dan tidak bermanfaat. *Scrum Master* membantu orang-orang untuk mengubah interaksi dan komunikasi untuk memaksimalkan nilai bisnis yang dihasilkan oleh *Scrum Team*.

Menurut Sarosa (2017) *Scrum Master* adalah fasilitator dalam proses *scrum*. Tugas *Scrum Master* adalah menghilangkan penghambat yang menghalangi *team* pengembang dalam melaksanakan tugasnya. *Scrum Master* bukanlah pemimpin *team* seperti halnya dalam organisasi tradisional. *Scrum Master* berperan sebagai penyangga (*buffer*) antara *team* dan gangguan. *Scrum Master* memastikan proses berjalan dengan baik seperti yang telah disepakati. Tanggung jawab *Scrum Master* adalah sebagai berikut:

1. Membantu pemilik produk menjaga agar anggota *team* memahami tugas masing-masing dan dapat terus membuat kemajuan.
2. Membantu anggota *team* pengembang mendefinisikan produk yang dianggap selesai berdasarkan masukan dari pemangku kepentingan.
3. Melatih anggota *team* pengembang untuk dapat menghasilkan produk yang berkualitas tinggi.
4. Mempromosikan konsep mengorganisir diri sendiri dalam *team*
5. Membantu *team* pengembang menyingkirkan penghalang kemajuan pengerjaan proyek baik internal maupun eksternal.

6. Memfasilitasi *team* dalam bekerja sehingga terdapat kemajuan secara rutin
7. Mengedukasi jajaran pemangku kepentingan akan prinsip *Scrum*.

2.3.6 Acara acara Scrum

Menurut Sutherland (2017) Acara-acara wajib dalam *Scrum* dilaksanakan guna terciptanya rutinitas dan mengurangi pertemuan lain yang bukan merupakan bagian dari inti *Scrum*. Setiap acara memiliki durasi, artinya setiap acara memiliki durasi waktu maksimal. Ketika *Sprint* dimulai, durasinya tetap yang artinya tidak bisa disingkat maupun diperpanjang. Acara selain *Sprint* dapat berakhir kapanpun juga ketika tujuan dari acara tersebut telah tercapai, hal ini dilakukan guna memastikan waktu yang digunakan tidak berlebihan dan tidak ada waktu yang terbuang di *sprint* yang sedang berjalan.

Selain *Sprint* yang merupakan wadah untuk acara-acara lainnya, setiap acara di dalam *Scrum* adalah sebuah kesempatan secara resmi untuk melakukan inspeksi dan mengadaptasi sesuatu hal yang penting. Acara acara *scrum* secara khusus dirancang untuk mewujudkan transparansi dan inspeksi yang bersifat menyeluruh. Tidak diselenggarakannya acara-acara *scrum* akan mengurangi transparansi dan menghilangkan kesempatan untuk melakukan inspeksi dan adaptasi terhadap sebuah proses yang penting.

2.3.6.1 Sprint

Menurut Sutherland (2017) Jantung dari *Scrum* adalah *Sprint*, yaitu sebuah batasan waktu dengan durasi satu bulan atau perminggu, di mana terdapat proses pembuatan *Increment* yang bernilai, dapat digunakan dan berpotensi untuk dirilis ke *market*. *Sprint* memiliki durasi yang konsisten sepanjang daur hidup pengembangan produk sistem informasi, bila *Sprint* dilakukan setiap 2 (dua) minggu sekali maka seterusnya *Sprint* akan dilakukan setiap 2 (dua) minggu sekali. *Sprint* yang terbaru langsung dimulai setelah *Sprint* sebelumnya selesai.

Sprint berisi dan terdiri dari perencanaan *Sprint*, pertemuan harian *Scrum*, pengembangan produk, *Sprint Review* dan *Sprint Retrospective*. Pada saat *Sprint* berjalan:

1. Tidak boleh ada perubahan yang dapat mengancam tujuan *Sprint*;
2. Tingkat mutu tidak boleh menurun; dan,
3. Ruang lingkup dapat diperjelas dan dinegosiasi ulang antara *Pemilik Produk* dan *Tim Pengembang* setiap kali ada hal baru yang harus di pelajari;

Setiap *Sprint* bisa dianggap sebagai sebuah proyek dengan durasi tidak lebih dari satu bulan yang di jalankan untuk mencapai suatu tujuan. Setiap *Sprint* memiliki tujuan mengenai apa yang harus diciptakan, sebuah rancangan dan perencanaan fleksibel yang mengelola pembangunan sistem informasi, daftar pekerjaan, dan *output* dari *Increment*.

Sprint dibatasi tidak lebih dari satu bulan kalender. Ketika durasi *Sprint* terlalu lama maka definisi dari apa yang akan dikembangkan dapat berubah, kompleksitas dapat meningkat, dan risiko juga dapat meningkat. Dengan memastikan adanya inspeksi dan adaptasi, keberadaan *Sprint* menciptakan sebuah tingkat kemungkinan perkembangan pekerjaan menuju *Sprint Goal* setidaknya setiap satu bulan kalender. Dengan adanya *Sprint*, risiko pengeluaran biaya dibatasi menjadi maksimal satu bulan kalender.

Membatalkan *Sprint*, *sprint* dapat dibatalkan sebelum mencapai batasan waktu. Pemilik Produk memiliki kemampuan untuk membatalkan *Sprint* yang sedang berjalan, walaupun ia bisa saja membatalkannya karena dipengaruhi oleh pemegang kepentingan, Tim pengembang ataupun *Scrum Master*.

Sprint bisa saja dibatalkan bila *Sprint Goal* berubah. Hal ini dapat terjadi ketika perusahaan mengubah haluan ataupun ketika pasar atau teknologi telah berubah. *Sprint* seharusnya dibatalkan bila sudah tidak masuk akal untuk dilanjutkan yang disebabkan oleh keadaan-keadaan terkini. Akan tetapi karena waktu *Sprint* yang sangat singkat, pembatalan *Sprint* seringkali tidak masuk akal untuk dilakukan.

2.3.6.2 Sprint Planning

Menurut Sutherland (2017) Pekerjaan yang akan dikerjakan di *Sprint* direncanakan pada saat *Sprint Planning*. Perencanaan dilakukan secara kolaboratif oleh seluruh anggota *Scrum Team*. *Perencanaan Sprint* memiliki batasan waktu 8 (delapan) jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih singkat, acara *sprint planning* biasanya lebih singkat. *Scrum Master* memastikan acara perencanaan *sprint* diselenggarakan dan peserta memahami tujuannya. *Scrum Master* mengawasi dan mendidik *Scrum Team* untuk menjaganya di dalam waktu yang sudah ditetapkan.

Sprint Planning menjawab pertanyaan-pertanyaan berikut:

1. Apa yang dapat diberikan ke dalam *Increment* dari *Sprint* sekarang?
2. Bagaimana *Scrum Team* bisa menyelesaikan pekerjaan yang dibutuhkan untuk menghasilkan *Increment*?

Tujuan *Sprint* adalah sebuah objektif untuk *Sprint* yang dapat dicapai lewat penerapan *Product Backlog*. Tujuan *Sprint* merupakan panduan bagi *Development Team* untuk menjawab pertanyaan mengapa *Scrum Team* menghasilkan *Increment*. Tujuan *Sprint* dibuat pada pertemuan Perencanaan *Sprint*. *Sprint* memberikan ruang yang luwes mengenai fungsionalitas yang akan diimplementasikan di dalam *Sprint*. *Product Backlog* item terpilih merupakan satu fungsi yang sesuai yang bisa jadi Tujuan *Sprint*.

2.3.6.3 Daily Scrum

Menurut Sutherland (2017) *Daily Scrum* adalah acara untuk *Development Team* yang memiliki batasan waktu 15 menit. Acara *daily scrum* dilakukan setiap hari selama *Sprint* berlangsung. Di acara *daily scrum*, *Development Team* membuat rencana kerja untuk 24 jam ke depan. Acara *daily scrum* mengoptimalkan kolaborasi dan performa dari tim dengan melakukan inspeksi pada pekerjaan masing-masing yang dilakukan semenjak Pertemuan harian *Scrum* sebelumnya dan melakukan prakiraan terhadap pekerjaan selanjutnya di dalam *Sprint*. *Pertemuan harian Scrum* dilakukan di waktu dan tempat yang sama setiap harinya untuk mengurangi kerumitan dan dalam kondisi berdiri.

Development Team menggunakan Pertemuan harian *Scrum* untuk menginspeksi perkembangan pekerjaan menuju tujuan *Sprint* dan tren perkembangan penyelesaian pekerjaan di *Sprint Backlog*. *Pertemuan harian Scrum* meningkatkan kemungkinan *Tim Pengembang* untuk mencapai Tujuan *Sprint*. Setiap hari, *Tim Pengembang* harus memahami bagaimana *Scrum Team* bekerja sama sebagai tim yang mengorganisasi diri sendiri untuk mencapai tujuan *Sprint* dan membuat *Increment* yang diharapkan di akhir setiap *Sprint*.

Struktur dari pertemuan harian *scrum* ditentukan oleh *Development Team* dan bisa diadakan lewat berbagai macam cara selama pertemuan *daily scrum* fokus terhadap kemajuan menuju *Sprint Goal*. Beberapa tim pengembang akan menggunakan beberapa pertanyaan, beberapa akan lebih berdiskusi. Berikut adalah contoh pertanyaan yang mungkin saja digunakan:

1. Apa yang telah *Scrum Master* lakukan kemarin untuk membantu tim pengembang mencapai tujuan *Sprint*?
2. Apa yang akan *Scrum Master* lakukan hari ini untuk membantu tim pengembang mencapai tujuan *Sprint*?
3. Apakah *Scrum Master* melihat ada hambatan yang menghalangi *Scrum Master* ataupun tim pengembang dalam mencapai tujuan *Sprint*?

Development Team atau beberapa anggota tim seringkali berkumpul setelah pertemuan harian *Scrum* untuk diskusi yang lebih mendalam di bagian teknis atau melakukan adaptasi atau melakukan rencana ulang terhadap sisa pekerjaan dari *Sprint* yang sudah berjalan.

Scrum Master memastikan bahwa tim pengembang menyelenggarakan pertemuan tersebut, tetapi tim pengembang yang bertanggung jawab untuk menjalankan Pertemuan Harian *Scrum*. *Scrum Master* mengajari tim pengembang untuk menjaga pertemuan harian *Scrum* tetap di dalam batasan waktu 15 menit perhari.

Daily Scrum adalah pertemuan internal untuk tim pengembang. Jika orang lain hadir, *Scrum Master* memastikan *Scrum Team* tidak mengganggu jalannya pertemuan *harian Scrum* tersebut.

Pertemuan harian Scrum meningkatkan kualitas komunikasi dan menghilangkan pertemuan-pertemuan lain yang kurang penting, mengidentifikasi kendala untuk dihilangkan, mengerucutkan dan mendukung pengambilan keputusan secara cepat, meningkatkan tingkat pengetahuan dari tim pengembang. Hal ini merupakan *point* penting dari pertemuan adalah inspeksi dan adaptasi.

2.3.6.4 Sprint Review

Menurut Sutherland (2017) *Sprint Review* diselenggarakan di akhir *Sprint* untuk menginspeksi *Increment* dan mengadaptasi *Product Backlog* bila diperlukan. Pada saat *Sprint Review*, *Scrum Team* dan pemangku kepentingan berkolaborasi untuk meninjau apa yang sudah diselesaikan di *Sprint sebelumnya*. Berdasarkan hasil pengamatan yang telah dilakukan dan perubahan terhadap *Product Backlog* di dalam *Sprint*, tim *Scrum* berkolaborasi untuk menentukan pekerjaan selanjutnya yang dapat dilakukan untuk mengoptimalkan nilai dari suatu proses bisnis. *Sprint review* adalah pertemuan informal, bukan pertemuan laporan status, dan presentasi *Increment* dilakukan guna mendapatkan *feedback* dan mengembangkan kemampuan yang bersinergi.

Pertemuan *Sprint review* paling lama diselenggarakan selama empat jam untuk *Sprint* berdurasi satu bulan. Untuk *Sprint* yang lebih singkat, durasi acara *Sprint review* biasanya lebih singkat. *Scrum Master* memastikan acara *Sprint review* terselenggara dan hadirin memahami tujuannya. *Scrum Master* menjelaskan kepada setiap hadirin untuk tepat waktu.

Sprint Review mengandung unsur-unsur berikut:

1. Hadirin yang di dalamnya adalah *Scrum Team* dan para pemegang kepentingan utama yang diundang oleh *Pemilik Produk*;
2. *Pemilik Produk* menjelaskan *Product Backlog* item yang sudah “Selesai” dan yang belum “Selesai”;
3. *Tim Pengembang* menjelaskan apa yang berjalan dengan baik sepanjang *Sprint*, mendeskripsikan masalah-masalah yang *Scrum Team* hadapi dan bagaimana cara menyelesaikannya;

4. Tim Pengembang mendemonstrasikan pekerjaan yang telah *Scrum Team* “Selesai”-kan dan menjawab pertanyaan-pertanyaan mengenai *Increment*;
5. Pemilik Produk menjelaskan keadaan *Product Backlog* hingga saat ini. Ia juga mengestimasi target dan tanggal penghantaran produk berdasarkan perkembangan hingga hari ini;
6. Seluruh hadirin berkolaborasi mengenai apa yang akan *Scrum Team* lakukan selanjutnya, artinya *Sprint Review* menghasilkan masukan berharga untuk perencanaan *Sprint* berikutnya;
7. Meninjau bagaimana keadaan pasar atau potensi penggunaan produk, yang mungkin telah berubah dan hal apa yang paling bernilai untuk dikerjakan berikutnya; dan,
8. Menganalisa jangka waktu, anggaran, potensi kemampuan, dan keadaan pasar untuk menulis fitur atau kemampuan produk yang sudah ditetapkan.

Hasil dari *Sprint Review* adalah *Product Backlog* yang sudah direvisi yang menjelaskan secara detail komponen *Product Backlog* yang mungkin diterapkan di *Sprint* berikutnya. *Product Backlog* juga dapat menyesuaikan secara menyeluruh untuk mendapatkan inovasi dari *market* yang potensial.

2.3.6.5 *Sprint Retrospective*

Menurut Sutherland (2017) *Sprint Retrospective* adalah sebuah kesempatan bagi *Scrum Team* baik *Scrum Master* maupun tim pengembang untuk menginspeksi dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya.

Sprint Retrospective dilaksanakan setelah *Sprint Review* dan sebelum perencanaan *Sprint* berikutnya. Acara *Sprint retrospective* diselenggarakan paling lama tiga jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih singkat misalnya 2 (dua) minggu, durasi acara *sprint retrospective* biasanya lebih singkat. *Scrum Master* memastikan acara *sprint retrospective* terselenggara dan mendukung setiap peserta untuk memahami tujuannya.

Scrum Master memastikan bahwa acara *Sprint retrospective* bernuansa positif dan produktif. *Scrum Master* menjelaskan kepada peserta untuk menjaga pertemuan ini tepat waktu. *Scrum Master* turut berpartisipasi sebagai rekan kerja yang bertanggung jawab terhadap proses *Scrum* di pertemuan *Sprint retrospective*. Tujuan utama dari *Sprint Retrospective* adalah:

1. Menginspeksi bagaimana jalannya *Sprint* terakhir yang terkait dengan orang-orang, hubungan antar *Scrum Team secara personal*, proses, dan alat-alat yang digunakan apakah sudah menunjang kinerja mereka;
2. Mengidentifikasi dan mengurutkan hal yang paling prioritas yang berjalan dengan baik dan peningkatan nilai yang berpotensi untuk dilakukan; dan,
3. Membuat rencana untuk penerapan peningkatan cara kerja *Scrum Team*

Scrum Master mendorong *Scrum Team* untuk membuat peningkatan dalam proses kerangka kerja *Scrum*, proses dan praktik-praktik pengembangan produk agar *Sprint* berikutnya lebih efektif dan membahagiakan. Di setiap *Sprint Retrospective*, *Scrum Team* merencanakan cara-cara yang terbaik untuk meningkatkan kualitas produk dengan cara meningkatkan proses kerja ataupun menyesuaikan isi dari definisi “Selesai”, apabila hal tersebut tidak menyangkut dengan standar mutu pengembangan produk di perusahaan.

Sebelum *Sprint Retrospective* berakhir, *Scrum Team* harus menyetujui peningkatan yang akan *Scrum Team* terapkan di *Sprint* berikutnya. Penerapan peningkatan di *Sprint* berikutnya adalah bentuk inspeksi dan adaptasi terhadap *Scrum Team* itu sendiri. Meskipun peningkatan dapat dilakukan kapan saja saat *Sprint* berjalan, *Sprint Retrospective* adalah sebuah kesempatan formal yang fokus pada inspeksi dan adaptasi setiap individu agar kinerja tim menjadi lebih optimal.

2.3.7 Artefak Scrum

Menurut Sutherland (2017) Artefak *Scrum* menjelaskan pekerjaan atau nilai proses bisnis guna terciptanya transparansi dan kesempatan untuk menginspeksi dan mengadaptasi pekerjaan. artefak yang dijabarkan *Scrum* dirancang sedemikian rupa untuk

memaksimalkan transparansi informasi utama agar setiap orang memiliki pemahaman dan perspektif yang sama mengenai artefak tersebut

2.3.7.1 Product Backlog

Menurut Sutherland (2017) *Product Backlog* adalah daftar sistematis semua hal yang telah diketahui hingga saat ini harus ada di dalam produk. *Product Backlog* adalah satu-satunya sumber kebutuhan untuk semua perubahan yang perlu diberlakukan terhadap produk. *Product Owner* bertanggung jawab terhadap *Product Backlog*, termasuk isi, ketersediaan dan skala prioritasnya.

Sebuah *Product Backlog* tidak pernah tuntas. Di awal pengembangan produk, *Product Backlog* berisi daftar kebutuhan yang diketahui dan dipahami saat ini. *Product Backlog* berubah seiring berkembangnya produk sistem informasi dan lingkungan di mana produk tersebut digunakan. *Product Backlog* bersifat dinamis dan berubah terus secara konstan agar produk menjadi layak, kompetitif dan bermanfaat. Selama produk masih ada, *Product Backlog* juga akan selalu ada.

Tim Pengembang bertanggung jawab untuk seluruh estimasi. Pemilik Produk dapat mempengaruhi *Tim Pengembang* dengan cara membantu *Scrum Team* untuk memahami dan membuat pertukaran, tetapi pada akhirnya orang-orang yang melakukan pekerjaan yang menentukan estimasi akhir.

Memantau Perkembangan Menuju Tujuan Akhir. Total sisa pekerjaan menuju tujuan akhir dapat dijumlahkan kapanpun juga. *Pemilik Produk* memantau total sisa pekerjaan ini setidaknya di setiap *Sprint Review*. *Pemilik Produk* membandingkannya dengan jumlah sisa pekerjaan dari *Sprint Review* sebelumnya untuk mengkaji proyeksi perkembangan pekerjaan menuju tujuan akhir di waktu yang telah ditetapkan. Informasi ini dibuat transparan untuk semua pemilik kepentingan.

2.3.7.2 Sprint Backlog

Menurut Sutherland (2017) *Sprint Backlog* adalah daftar *Product Backlog* item yang terpilih untuk *Sprint* ditambah perencanaan untuk menghasilkan *Increment* dan mencapai

tujuan dari *Sprint*. *Sprint Backlog* adalah perkiraan pekerjaan dari tim pengembang mengenai fungsionalitas yang akan masuk ke dalam *Increment* berikutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitas produk menjadi *Increment* yang “Selesai” dan mempunyai *value*.

Sprint Backlog menampilkan seluruh pekerjaan yang menurut *Development Team* perlu dikerjakan untuk mencapai tujuan dari *Sprint*. Untuk memastikan adanya *increment* berkelanjutan, *Sprint Backlog* berisi setidaknya satu peningkatan proses dengan prioritas yang paling penting dari hasil pertemuan *Sprint Retrospective* sebelumnya.

Sprint Backlog adalah perencanaan yang sangat detail sehingga perubahan yang sedang dikerjakan dapat dipahami pada saat *Pertemuan harian Scrum*. *Tim pengembang* mengubah *Sprint Backlog* sepanjang *Sprint*, dan *Sprint Backlog* muncul saat *Sprint* berlangsung. Seiring dengan tim pengembang bekerja sesuai perencanaan dan semakin bertambahnya pemahaman *Scrum Team* mengenai pekerjaan yang dibutuhkan untuk mencapai tujuan dari *Sprint*.

Jika ada pekerjaan baru yang diperlukan, maka *tim pengembang* menambahkannya ke *Sprint Backlog*. Jika ada yang sudah diselesaikan maka estimasi sisa pekerjaannya diperbarui. Jika ternyata ada pekerjaan yang tidak diperlukan, maka bisa dihilangkan. Hanya *tim pengembang* yang dapat mengubah isi dari *Sprint Backlog* milik *Scrum Team* sepanjang *Sprint*. *Sprint Backlog* dapat dilihat secara jelas, menggambarkan keadaan terkini mengenai sisa pekerjaan yang telah direncanakan tim pengembang untuk diselesaikan dalam *Sprint* dan merupakan otoritas tim pengembang sepenuhnya.

Tim pengembang di setiap saat di dalam *Sprint* selalu memeriksa *Sprint Backlog*, total sisa pekerjaan di dalam *Sprint Backlog* dapat dijumlahkan. Tim pengembang meninjau total sisa pekerjaan ini setidaknya setiap hari pada saat pertemuan harian *Scrum* untuk dapat memproyeksikan apakah kira-kira tujuan dari *Sprint* dapat tercapai. Dengan memantau sisa pekerjaan ini sepanjang *Sprint*, *tim pengembangan* dapat mengelola perkembangan pekerjaannya masih relevan dengan tujuan *sprint* atau tidak.

2.3.8 Increment

Menurut Sutherland (2017) adalah jumlah semua item *Product Backlog* yang diselesaikan selama Sprint dan nilai *Increment* semua *Sprint* sebelumnya. Di akhir *Sprint*, *Increment* baru harus menjadi "Selesai," yang berarti harus dalam kondisi dapat digunakan dan memenuhi standar selesai berdasarkan kesepakatan *team scrum*.

Sebuah *Increment* adalah hasil pekerjaan yang bisa diinspeksi dan telah selesai guna mendukung empirisme di akhir *Sprint*. *Increment* adalah sebuah langkah kecil menuju sebuah visi ataupun tujuan. *Increment* harus mempunyai nilai dan dapat digunakan walaupun *Product Owner* memutuskan untuk merilisnya ataupun tidak.

2.3.9 Transparansi Artefak

Menurut Sutherland (2017) *Scrum* mengutamakan nilai transparansi. Semua keputusan untuk memaksimalkan nilai dari sebuah proses bisnis dan mengelola risiko dibentuk dengan dasar keadaan yang sedang terjadi dari artefak-artefak *scrum*. Keputusan yang akan tercipta akan mempunyai nilai yang utuh bila sudah memenuhi sifat dari transparansi. Apabila artefak – artefak seperti *product backlog* dan lain-lainnya tidak transparan maka keputusan yang tercipta menjadi kurang valid. Value menjadi berkurang dan risiko yang akan timbul kemungkinan akan meningkat.

Scrum Master harus bekerja sama dengan *Product Owner*, *Development Team* dan pihak-pihak lainnya untuk memastikan bahwa alat-alat *Scrum* benar-benar transparan.. Seorang *Scrum Master* dapat mendeteksi transparansi yang tidak utuh dengan menginspeksi artefak *scrum*, merasakan terjadinya pola yang berulang-ulang, mendengarkan dengan seksama apa yang dikatakan orang-orang dan mendeteksi perbedaan antara apa yang diinginkan dengan apa yang dihasilkan.

Tugas seorang *Scrum Master* adalah bekerja-sama dengan *Scrum Team* dan organisasi untuk meningkatkan transparansi dari artefak *Scrum*. Pekerjaan ini melibatkan proses edukasi, meyakinkan orang-orang di dalam tim dan membuat perubahan yang berguna. Transparansi tidak terjadi dalam semalam, tetapi merupakan sebuah perjalanan jangka panjang.

2.4 Unified Modeling Language (UML)

Menurut Sarosa (2017) UML (*Unified Modeling Language*) adalah aturan – aturan pemodelan yang dapat digunakan untuk metodologi perancangan sistem informasi berorientasi objek.

Menurut Munawar (2018) UML adalah bahasa pemodelan yang baku yang berfungsi untuk membuat *blue print* atas visi mereka agar proses berbagi (*sharing*) dan berkomunikasi dengan tim menjadi lebih efektif

2.4.1 Use Case Diagram

Menurut Tohari (2014) Diagram *use case* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Hal ini ditekankan pada diagram ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. *Use case* mengaplikasikan sebuah interaksi antara *actor* dengan sistem secara sederhana. *Use case* ini digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi.

Salahuddin, (2011) Diagram *Use Case* menggambarkan semua aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar, yang menjadi persoalan apa yang dilakukan dan bagaimana melakukannya. Diagram *Use Case* dekat dengan kejadian-kejadian. Kejadian merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem. Diagram ini menjelaskan hubungan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasikan dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan penngguna *use case diagram* dapat membantu bila kita sedang menyusun kebutuhan sistem, mengkomuikasikan rancangan dengan klien, dan merancang *test* khusus untuk semua desain yang ada pada sistem.

2.4.2 Class Diagram

Kelas (*class*) adalah sebuah model statis yang jika diinstansiasi akan menghasilkan sebuah objek yang mempunyai nilai tertentu dan merupakan inti dari pengembangan dan

perancangan berorientasi objek. Kelas menggambarkan keadaan (*atribut* atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi). Menurut Tohari (2014).

Menurut Salahuddin (2011), diagram kelas atau *class diagram* adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sebuah sistem. *Class diagram* memiliki atribut dan metode atau operasi.

Kelas-kelas yang ada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada *class diagram* sebaiknya memiliki jenis yaitu kelas utama, kelas yang menangani tampilan sistem, kelas yang diambil dari pendefinisian *use case* dan kelas yang diambil dari pendefinisian data yang menjelaskan model dari kelas tersebut.

2.4.3 Deployment Diagram

Menurut Tohari (2014) *Deployment Diagram* menunjukkan tata letak sebuah sistem secara fisik. Diagram ini akan menampilkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* secara detail yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut.

Deployment Diagram menggambarkan detail bagaimana komponen *di deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisik. Hubungan antar titik (misalnya *IP Address*) dan kebutuhan *jaringan* dapat juga dijelaskan dengan menggunakan *deployment diagram*.

2.4.4 Component Diagram

Menurut Tohari (2014) Diagram Komponen (*Component Diagram*) digunakan untuk memodelkan aspek fisik dari suatu sistem. Aspek fisik ini berupa modul-modul yang berisikan *code atau syntax*, baik *library* maupun *file executable*, file atau dokumen yang ada di dalam *node*. Aspek fisik inilah yang disebut komponen dalam *UML*. Komponen

diagram ini digunakan untuk memvisualisasikan organisasi dan hubungan antar komponen dalam suatu sistem. Diagram ini juga digunakan untuk menjelaskan komponen sistem apa saja yang dapat diproses.

Component Diagram bila dikombinasikan dengan diagram lainnya dapat digunakan untuk menggambarkan pendistribusian secara fisik dari modul perangkat lunak sebuah jaringan. Misalnya, ketika merancang sistem *client-server*, hal ini berguna untuk menunjukkan mana kelas atau paket kelas yang akan berada pada *node client* dan mana yang akan berada di server. *Component diagram* juga dapat berguna dalam merancang model sistem informasi berbasis komponen. Karena berfokus pada analisis sistem berorientasi objek dan desain.

2.4.5 Activity Diagram

Menurut Tohari (2014) Activity Diagram memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses. Activity Diagram bermanfaat untuk menggambarkan interaksi antara beberapa *use case* yang sejalan dan menghasilkan bisnis proses yang terjadi dalam sebuah aktifitas.

Activity diagram menyediakan analisis dengan kemampuan untuk memodelkan proses dalam suatu sistem informasi. *Activity diagram* dapat digunakan untuk alur kerja model, *use case* individual, atau logika keputusan yang terkandung dalam metode individual. *Activity diagram* juga menyediakan sebuah pendekatan untuk proses pemodelan secara beriringan.

Pada dasarnya, *activity diagram* merupakan diagram canggih dan merupakan diagram aliran data yang terbaru. Secara teknis, *activity diagram* menggabungkan ide-ide proses pemodelan dengan teknik yang berbeda, termasuk model acara, *statecharts*.

2.4.6 Package Diagram

Menurut Tohari (2014) *Package* yang dalam bahasa Indonesia berarti paket atau sekumpulan data yang terbungkus dalam sebuah wadah dalam dunia *jaringan sistem* dimanfaatkan dalam komunikasi datanya agar lebih efektif di mana data tidak dikirim

secara langsung dalam bentuk binernya melainkan dikelompokkan terlebih dahulu dalam paket-paket tertentu. *Package* diagram merupakan salah satu dari diagram *UML* yang erat kaitannya dengan diagram *UML* yang lain.

Sedangkan dalam bahasa *Java* dan *C++*, *package* selalu diimport saat menuliskan *code* programnya. *Package* merupakan kumpulan dari *class*. Penggambaran diagram *package* mirip dengan simbol *package* dalam *Microsoft Windows*. *Package* Diagram dapat memvisualisasikan ketergantungan antar bagian dalam suatu sistem dan bermanfaat dalam mencari letak permasalahan kompilasi.

2.5 Tools yang digunakan

2.5.1 Hypertext Preprocessor (PHP)

PHP (*Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan *HTML* untuk membuat halaman *web* dinamis. Karena *PHP* merupakan sebuah bahasa *server-side scripting* yang mana sintaks atau baris *code PHP* akan diproses di sisi *server* kemudian output pemrosesan tadi dikirimkan ke *browser* dalam bentuk *HTML*. Dengan demikian kode program dalam bahasa pemrograman *PHP* tidak terlihat oleh pengguna, yaitu halaman *website* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data pengguna ke halaman *website* (Arief, 2011).

Menurut Raharjo (2015), *Hypertext Preprocessor (PHP)* adalah bahasa pemrograman untuk membuat *website* yang dapat digunakan untuk tujuan umum, sama seperti bahasa pemrograman dengan ruang lingkup *LAN*. Dalam proses penulisan kode program, *PHP* juga mengizinkan kita untuk menggunakan dua gaya pemrograman, yaitu gaya prosedural yaitu berurutan proses implementasinya dan gaya berorientasi objek, dan bisa juga gabungan dari keduanya.

2.5.2 MySQL

Menurut Aditya (2011) *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (*Structure Query Language*) yang *multithread*, *multiuser* berfungsi untuk

mengelola basis data sebuah sistem informasi dan merupakan tools basis data yang paling banyak digunakan di seluruh dunia.

MySQL AB membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi *GNU General Public License (GPL)*, tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan *GPL* seperti penggunaan dengan skala *enterprise* dan sistem informasi yang khusus penggunaannya.

2.6 Penelitian Terdahulu

Tabel 2.1 merupakan rangkuman dari penelitian yang pernah dilakukan sebelumnya yang melakukan penelitian yang serupa :

Tabel 2.1 Daftar Penelitian Terdahulu

Judul Penelitian	Peneliti	Hal yang diteliti	Tahun penelitian
<i>The Impact of Customer Relationship Management on Customer Loyalty in LG Company (Goldiran) Tehran City</i>	Karsalari	Menganalisis sales <i>growth</i> dan loyalitas pelanggan dengan menerapkan <i>CRM</i> di Perusahaan LG	2017
<i>Customer Loyalty in Collaborative Consumption Model: Empirical Study of CRM for Product-Service Sistem - Based e-Commerce in Indonesia</i>	Erdaka	Menganalisis Loyalitas pelanggan dari penerapan <i>CRM</i> di <i>E-Commerce</i> yang ada di Indonesia	2015
<i>Analisis Dan Perancangan Sistem Informasi Electronic Customer Relationship Management(E-CRM) PT. Delta Jaya Motor Singkawang</i>	Khairuzzaman	Perancangan Sistem Informasi berbasis web <i>e-CRM</i> yang dapat mengelola data pelanggan, promosi produk, produk data penjualan, dan pengelolaan layanan pelanggan yang digunakan untuk menyimpan pertanyaan pelanggan dan keluhan pelanggan.	2016
<i>Architecting a Geo-Enabled CRM: the Way to Seamless Integration</i>	Abdulmyanov , Sivtsov, Fomichyov	Menganalisis perancangan <i>CRM</i> dengan menggunakan metode <i>Agile</i>	2017

Perbedaan penelitian terdahulu dengan penelitian saat ini dilakukan adalah penelitian yang dilakukan lebih membahas tentang konsep dan perancangan sistem informasi *CRM* tanpa menggunakan metode *Agile Development Scrum*. pada penelitian dilakukan perancangan Sistem Informasi *CRM* dengan menggunakan metode *Agile Development Scrum* yang belum dilakukan pada penelitian sebelumnya.

BAB III

OBJEK DAN METODOLOGI PENELITIAN

3.1 *Profile* PT. Rabana Sejahtera Indonesia

Objek penelitian yang diambil adalah PT. Rabana Sejahtera Indonesia yang berada di daerah Kabupaten Cirebon Provinsi Jawa Barat. Pemilihan objek penelitian dilakukan secara sengaja dengan pertimbangan objek penelitian saat ini hanya memiliki metode pengelolaan hubungan dengan *customer* secara konvensional dan masih menulis di buku manual serta perusahaan ingin mengembangkan ke *media social* dan *online*. Waktu yang diambil pada proses penelitian dilakukan pada bulan Februari 2019 hingga Mei 2019.

PT. Rabana Sejahtera Indonesia merupakan perusahaan yang memproduksi berbagai macam obat herbal, seperti obat kanker, obat diabetes, teh herbal dan lain-lainnya, PT. Rabana Sejahtera Indonesia merupakan perusahaan produsen obat herbal, tentunya produk yang PT. Rabana Sejahtera Indonesia jual adalah obat herbal berkualitas karena pengembangan yang selalu melakukan pengembangan produk sesuai dengan permintaan dan kebutuhan konsumen.

Keunggulan Dari Produk PT. Rabana Sejahtera Indonesia, Indonesia kaya akan keanekaragaman hayati, dengan keanekaragaman itulah berlimpah bahan - bahan untuk ramuan herbal alami yang berkualitas namun tetap harus disentuh dan didukung oleh teknologi yang canggih.

PT. Rabana Sejahtera Indonesia hadir sebagai perusahaan yang memproduksi dan mendistribusikan Formula *Nanotechnology* Organik yang melibatkan berbagai bahan-bahan alami berkualitas terbaik asli dari tanah Indonesia, seluruh Formula yang diproduksi merupakan hasil dari penelitian panjang selama 15 tahun terakhir yang dilakukan oleh tim Formulator PT. Rabana Sejahtera Indonesia.

PT. Rabana Sejahtera Indonesia selalu menekankan niat untuk tidak membebani para pasiennya yang sedang berusaha ingin kembali sehat, dengan slogan "*We Care To Be Healthy*" PT. Rabana Sejahtera Indonesia memiliki Program-program sosial yaitu :

Adanya program Bergaransi 100% uang kembali jika tidak ada efek positif setelah penggunaan, namun harus dilengkapi dengan bukti Lab sebelum penggunaan Formula dan bukti Lab sesudah penggunaan Formula.

Bukan hanya itu. Selain menawarkan solusi untuk mengatasi berbagai permasalahan kesehatan, PT. Rabana Sejahtera Indonesia juga menyediakan fasilitas untuk konsultasi serta pemantauan selama penggunaan produk PT. Rabana Sejahtera Indonesia secara gratis dengan Konsultan Kesehatan dan para *Customer Care* dari PT. Rabana Sejahtera Indonesia.

PT. Rabana Sejahtera Indonesia juga menyediakan potongan harga bagi pasien pemilik kartu BPJS Kesehatan dan Asuransi Swasta dengan cukup menunjukkan bukti kartu keanggotaan Bpjs/Asuransi Swasta tersebut kepada *Customer Care* PT. Rabana Sejahtera Indonesia pada saat pemesanan/pembelian produk.

Selain itu sebagai bentuk nyata kepedulian sosial PT. Rabana Sejahtera Indonesia terhadap masyarakat luas, bagi pelanggan yang tergolong masyarakat ekonomi lemah (kurang mampu) PT. Rabana Sejahtera Indonesia juga memiliki program *Customer Social Responsibility*, cukup melampirkan Surat Keterangan Tidak Mampu (SKTM) pada saat pemesanan/pembelian produk melalui *Customer Care* PT. Rabana Sejahtera Indonesia maka pelanggan berhak mendapatkan potongan harga 50% hal ini PT. Rabana Sejahtera Indonesia lakukan agar produk-produk PT. Rabana Sejahtera Indonesia lebih banyak dapat menjangkau masyarakat yang kurang mampu.

3.1.1 Visi dan Misi PT. Rabana Sejahtera Indonesia

Visi PT. Rabana Sejahtera Indonesia adalah Mengukir Takdir Biologis, Menciptakan Subtansi Organik, Menemukan Solusi Adekuat Untuk Kemaslahatan Umat Manusia.

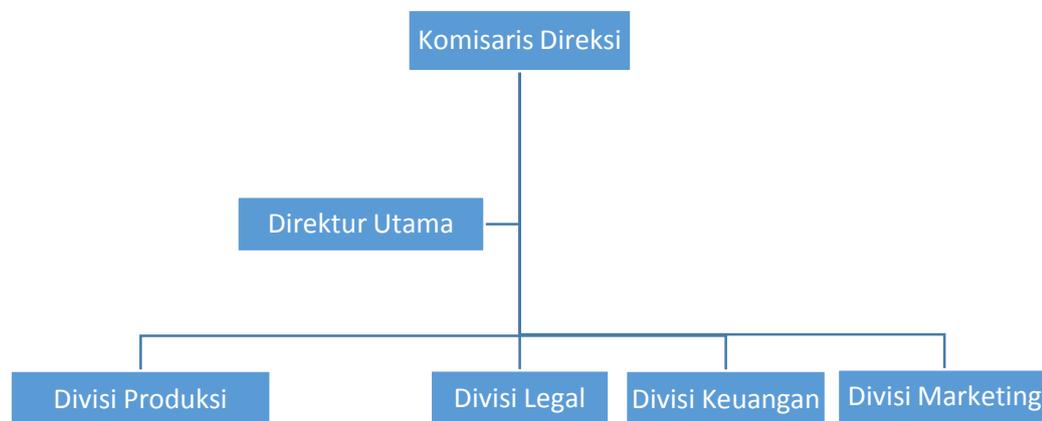
Sedangkan Misi dari PT. Rabana Sejahtera Indonesia adalah sebagai berikut :

1. Mengembangkan Produk-Produk Formula Herbal Aman, dan Berkualitas.
2. Mengembangkan Metode Penyembuhan yang Aman, Efektif, dan Efisien.
3. Selalu Terlibat Secara Eksperimental dalam Proses Pengayaan Khasanah maupun Penciptaan Bersama Formula Herbal Masyarakat Dunia.

4. Selalu Mencari dan Menciptakan Hal-Hal yang Paling Esensial Untuk Menjaga Kesehatan Manusia.
5. Membangun Hubungan Kerjasama Baik Rill maupun *Virtual* Guna Mewujudkan Dunia yang Sehat, Beriman, dan Beradab.

3.1.2 Struktur Organisasi Perusahaan

Struktur Organisasi yang baik dapat membantu setiap divisi / anggota perusahaan mengetahui batas tugas, wewenang dan tanggung jawab. Penelitian ini dilakukan pada departemen *Marketing* di sebuah perusahaan obat herbal yang memiliki struktur organisasi yang akan ditunjukkan pada gambar 3.1.



Gambar 3.1 Struktur Organisasi PT. Rabana Sejahtera Indonesia

3.1.3 Deskripsi Tugas

Deskripsi tugas merupakan keterangan dan pernyataan tertulis mengenai tugas yang harus dilaksanakan, wewenang, tanggung jawab dan hubungan kerja antar bagian pada suatu perusahaan.

Tugas yang dimiliki oleh seorang Direktur pada PT. Rabana Sejahtera Indonesia dapat didefinisikan sebagai berikut :

1. memimpin perusahaan dengan menerbitkan kebijakan-kebijakan perusahaan atau institusi

2. memilih, menetapkan, mengawasi tugas dari karyawan dan kepala divisi (*kadiv*) atau wakil direktur.
3. menyetujui anggaran tahunan perusahaan atau institusi.
4. menyampaikan laporan kepada pemegang saham atas kinerja perusahaan atau institusi.

Tugas yang dimiliki oleh seorang Kepala Divisi Produksi pada PT. Rabana Sejahtera Indonesia dapat didefinisikan sebagai berikut :

1. Mengawasi pelaksanaan proses produksi, mulai dari bahan baku awal sampai menjadi barang jadi.
2. Mengawasi pemakaian bahan baku, pemakaian *packing material* dan bahan pembantu lainnya dengan meminimalkan pemborosan dan kegagalan proses.
3. Menjaga dan mengawasi agar kualitas bahan baku dalam proses dan kualitas barang jadi sesuai dengan standar yang telah ditetapkan.
4. Mengawasi pembuatan laporan produksi, yang meliputi laporan absensi, pemakaian bahan baku, hasil produksi, dan jam berhenti tiap - tiap mesin.

Tugas yang dimiliki oleh seorang Kepala Divisi Legal pada PT. Rabana Sejahtera Indonesia dapat didefinisikan sebagai berikut :

1. Melakukan *update* regulasi nasional maupun lokal, khususnya yang terkait dengan perizinan perusahaan dibidang hukum.
2. Mampu menyusun dokumen perizinan dan perjanjian kerjasama dengan *stakeholder*.
3. Memahami dengan baik undang-undang dan segala peraturan tentang kesehatan..
4. Memiliki kemampuan advokasi, seperti mampu memahami urutan dan detil beracara di pengadilan.

Tugas yang dimiliki oleh seorang Kepala Divisi Keuangan pada PT. Rabana Sejahtera Indonesia dapat diidefinisikan sebagai berikut :

1. Mengkoordinasikan dan mengontrol perencanaan, pelaporan, serta pembayaran kewajiban pajak perusahaan agar efisien, akurat dan tepat waktu sesuai dengan aturan pemerintah.

2. Merencanakan dan mengkoordinasikan penyusunan anggaran perusahaan, serta mengontrol penggunaan anggaran tersebut untuk memastikan penggunaan dana secara efektif dan efisien dalam menunjang kegiatan operasional perusahaan.
3. Merencanakan, mengkoordinasikan, dan mengontrol arus kas perusahaan, terutama pengelolaan piutang dan utang. Sehingga, hal ini dapat memastikan ketersediaan dana untuk operasional perusahaan dan kondisi keuangan dapat tetap stabil.
4. Mengelola fungsi akuntansi dalam memproses data dan informasi keuangan untuk menghasilkan laporan keuangan yang dibutuhkan perusahaan secara akurat.

Tugas yang dimiliki oleh seorang Kepala Divisi Marketing pada PT. Rabana Sejahtera Indonesia dapat didefinisikan sebagai berikut :

1. Menyusun strategi *marketing* berdasarkan keadaan *market* dan kemampuan perusahaan.
2. Melakukan riset pemasaran sesuai trend yang terjadi di *market* khususnya pada produk kompetitor yang serupa.
3. Menyusun regulasi terkait kegiatan *marketing*.
4. Mengidentifikasi dan menganalisis potensi dan peluang usaha.

3.1.4 Rekap Penjualan PT. Rabana Sejahtera Indonesia

Rekap Penjualan pada PT . Rabana Sejahtera Indonesia berisikan bulan, jumlah order, omzet dan harga ongkos kirim. Data rekap penjualan ini ditulis di dalam buku penjualan kemudian diolah kembali di *Microsoft Excel* sehingga CS dan bagian keuangan melakukan pekerjaan yang berulang.

Tabel 3.1 Rekap Penjualan Tahun 2018

No	Bulan	Jumlah Order	Omzet	Ongkos Kirim
1	Januari	1102	413,005,500	5,514,799
2	Februari	1117	442,423,125	8,693,431
3	Maret	1243	483,272,488	10,406,229
4	April	1241	503,607,750	10,572,900
5	Mei	1362	488,780,450	11,416,667
6	Juni	839	292,076,450	6,838,704
7	Juli	1302	487,221,000	13,159,334

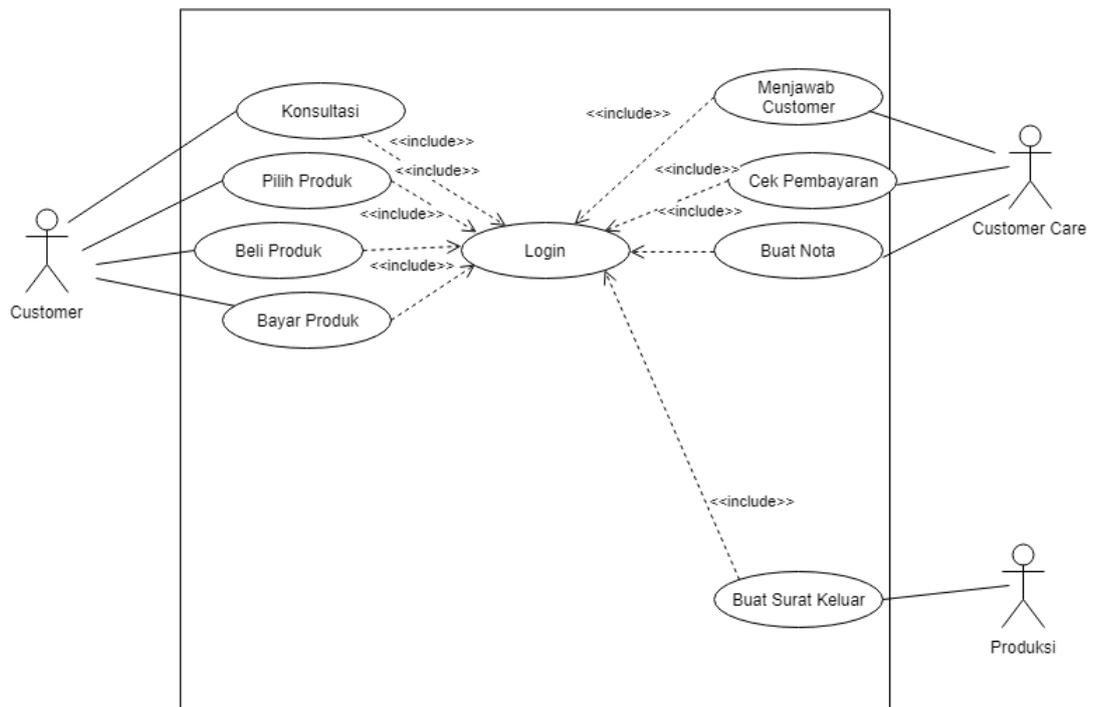
Tabel 3.1 Rekap Penjualan Tahun 2018

8	Agustus	1279	432,166,500	10,812,380
9	September	1247	405,576,990	12,850,942
10	Oktober	1333	398,559,700	13,096,780
11	November	1265	386,393,777	13,205,203
12	Desember	1140	326,433,853	10,643,590
Total			5,059,517,583	127,210,959
Grand Total (Omzet + Ongkos Kirim)		5,186,728,542		

Sumber : Laporan Keuangan tahun 2018 PT. Rabana Sejahtera Indonesia

3.2 Proses Bisnis

Proses bisnis yang sedang berjalan pada PT. Rabana Sejahtera Indonesia ditunjukkan dengan menggunakan *Use Case Diagram*.



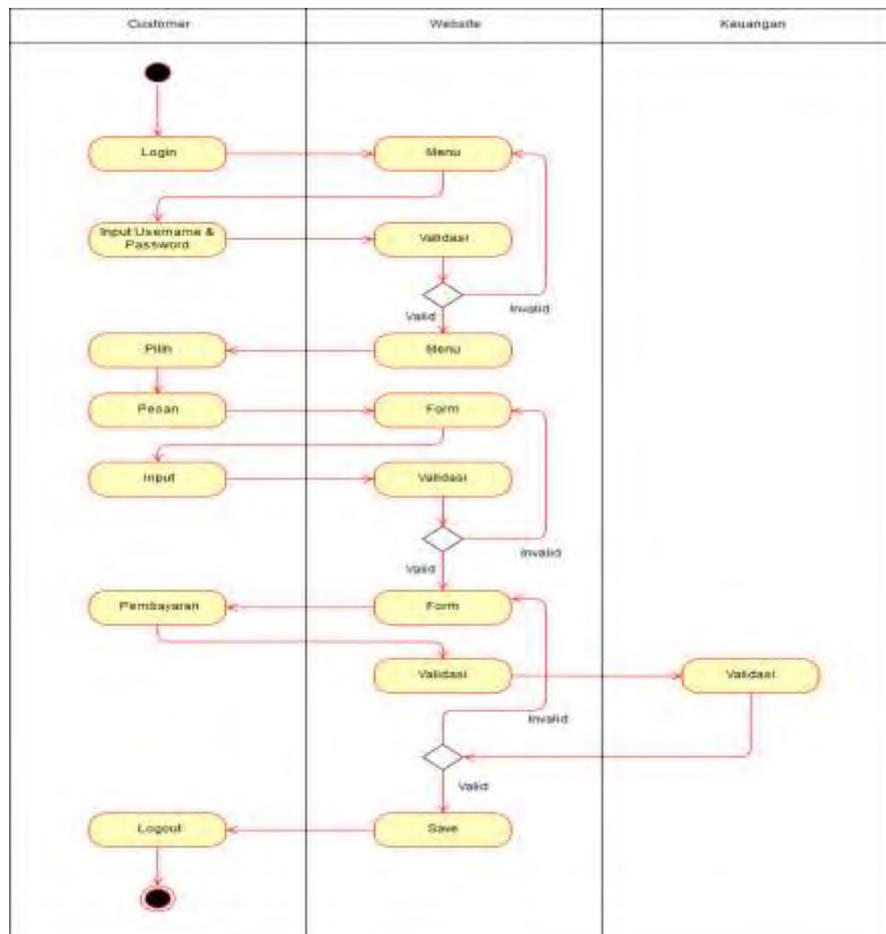
Gambar 3.2 Use Case Diagram Sistem yang berjalan

Gambar 3.2 menunjukkan proses bisnis yang berkaitan dengan *website* pada PT. Rabana Sejahtera Indonesia diawali dengan konsumen melakukan konsultasi ke CS atau CS menelpon *Customer* untuk menawarkan produk, *Customer* melakukan pembelian dengan memilih produk yang sesuai dengan penyakitnya dan melakukan proses pembayaran. Pada proses pembayaran memiliki beberapa metode pembayaran yaitu *Virtual Account BNI* dan Bank Transfer. Pada metode pembayaran Bank Transfer, *Customer* diwajibkan melakukan upload bukti transfer pada *website CRM* PT. Rabana

Sejahtera Indonesia. Setelah bukti transfer diterima, *Customer Care* akan membuat nota dan selanjutnya pesanan akan disimpan di *website CRM*.

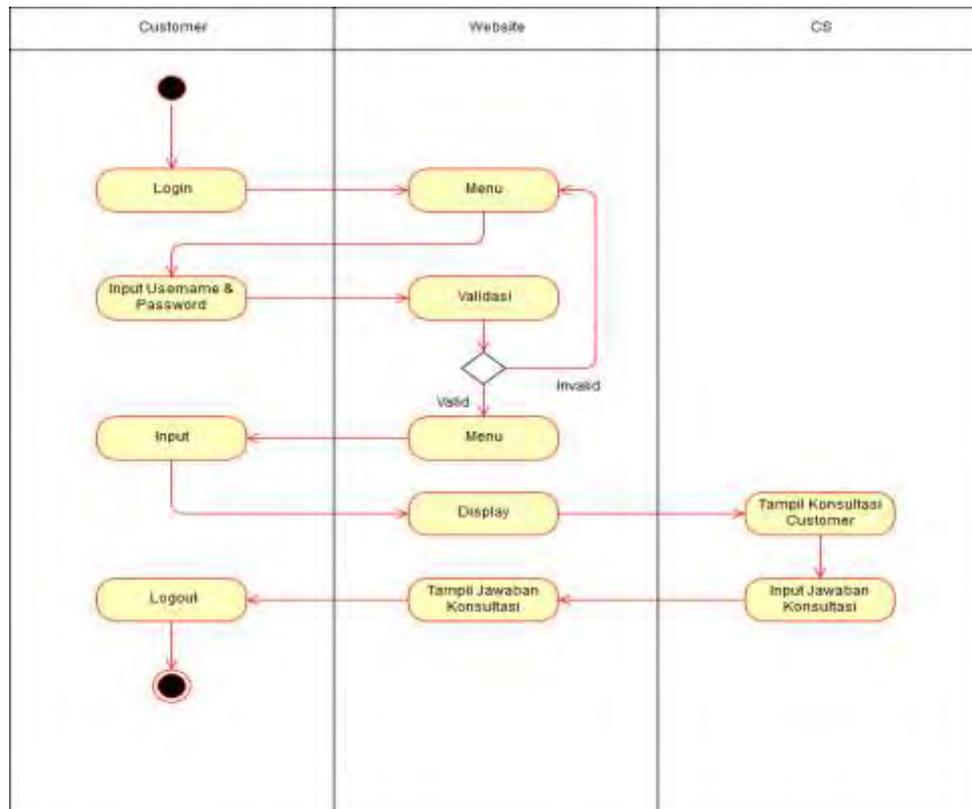
Bagian produksi akan menerima suatu notifikasi pemesanan melalui *sistem CRM* dan melakukan *packing* produk dan mengirimkan produk ke *Customer* dengan memberikan status kepada *Customer* bahwa produk telah dikirim. Pada tahap ini *Customer* akan memperoleh laporan pengiriman pesanan dan konsumen melakukan konfirmasi penerimaan kepada pihak *Customer Care*.

Menu konsultasi digunakan untuk mencatat konsultasi yang mungkin terjadi seperti, konsultasi produk, *Komplain Customer*, *refund* atau *return* produk dengan status dari kasus tersebut. Menu *Task* digunakan untuk menginputkan data tugas *customer care* setiap harinya seperti menelpon *customer*, memantau perkembangan kesehatan *customer*, acara - acara yang dilakukan dengan *customer* dan sebagainya.



Gambar 3.3 Activity Diagram Proses konsultasi

Gambar 3.3 merupakan gambaran proses bisnis konsultasi *customer* kepada CS PT. Rabana Sejahtera Indonesia. Tim CS akan melakukan pengecekan data konsultasi yang dilakukan oleh *customer*, konsultasi dapat melalui konsultasi langsung dengan datang ke kantor PT. Rabana Sejahtera Indonesia, telepon, *email*, *media social* dan lain-lainnya. Proses konsultasi ini berhubungan dengan bagian CS karena bagian ini mempunyai tugas utama melayani *customer* dengan sebaik-baiknya.



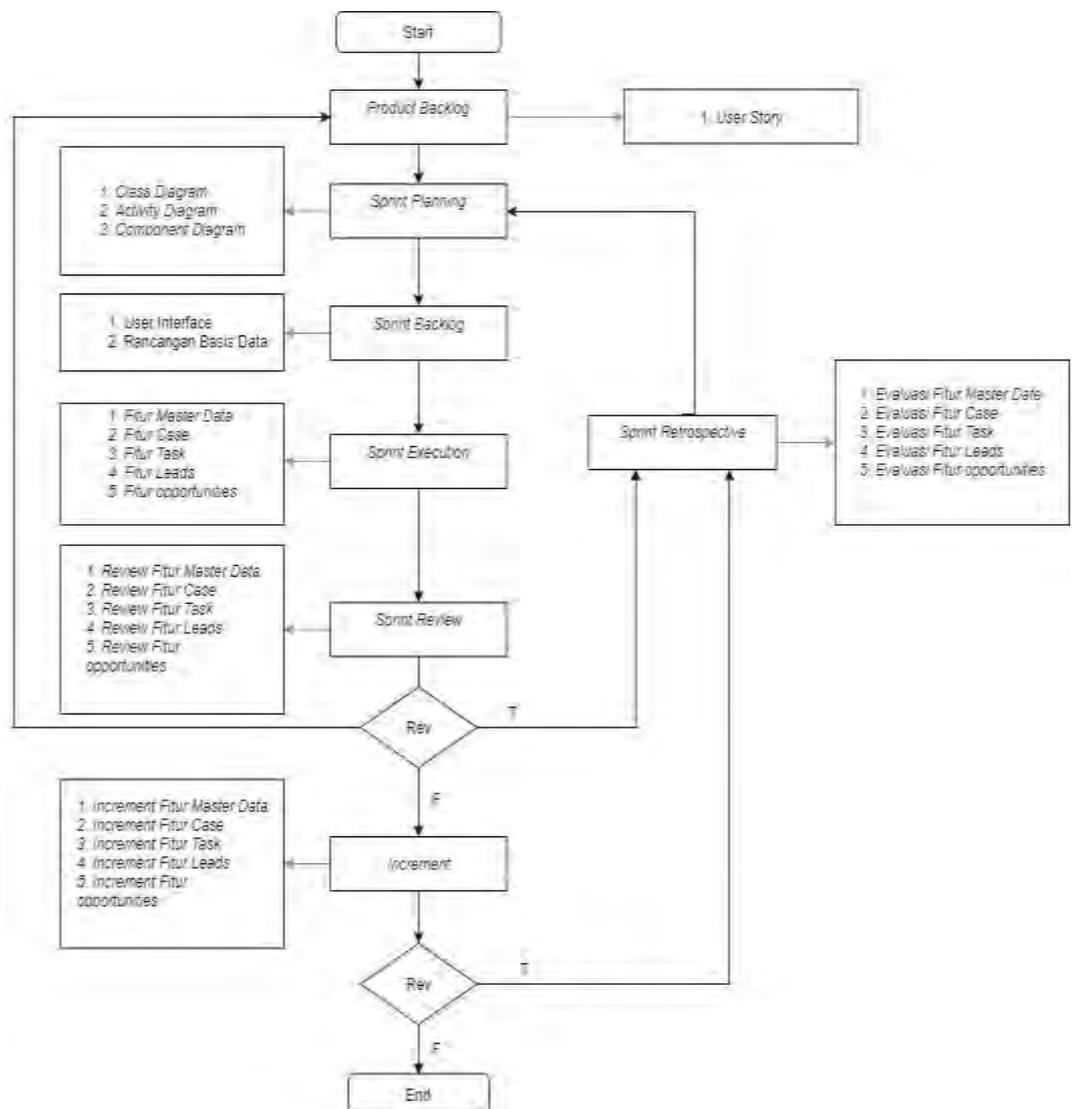
Gambar 3.4 Activity Diagram Proses pembelian

Gambar 3.4 merupakan gambaran proses bisnis dari *customer* yang digambarkan menggunakan *activity diagram*. Proses ini dilakukan saat *customer* melakukan proses transaksi pembelian di situs PT. Rabana Sejahtera Indonesia. Proses pembelian diawali dengan melakukan pemilihan produk dengan melihat katalog. Dilanjutkan dengan proses pembelian obat yang sesuai dengan kebutuhannya di mana pada proses ini *customer* melakukan pengisian alamat pengiriman dan keterangan profil *customer*. Pada proses ini juga dilakukan kalkulasi atas nominal transaksi yang dilakukan beserta ongkos kirim yang harus dibayar. Proses selanjutnya adalah proses pembayaran di mana konsumen akan memilih metode pembayaran yang akan digunakan (*Virtual Account dan Bank Transfer*).

Proses pembayaran akan melalui proses pengecekan data valid yang dilakukan oleh bagian keuangan. Jika proses validasi ini tidak berhasil, maka proses akan kembali kepada proses pembayaran di mana *customer* harus mencoba kembali atau menggunakan metode pembayaran lain. Jika proses validasi dana berhasil, maka pesanan obat *customer* akan diproses ke proses berikutnya.

3.3 Metode Pengembangan Sistem Informasi

Pada penelitian ini metode yang digunakan adalah *Scrum Framework* untuk pengembangan sistem informasi. Dalam membangun sebuah sistem informasi maka dibutuhkan beberapa langkah penelitian seperti yang akan diuraikan pada gambar 3.5.



Gambar 3.5 Metodologi Penelitian

Proses kerja pengembangan sistem informasi CRM menggunakan *Scrum Framework* yaitu :

1. *Product backlog*

Menentukan produk sistem informasi CRM yang telah disepakati oleh seluruh anggota tim *Scrum*. *Output* dari *product backlog* adalah beberapa *user story*.

2. *Sprint planning*

Pada tahap ini proses merencanakan apakah yang dapat dihantarkan ke dalam *increment* di *sprint* yang telah berlangsung dan bagaimana menyelesaikan pekerjaan yang dibutuhkan untuk menghantarkan *increment*. *Output* dari *sprint planning* adalah *Use Case Diagram, Class Diagram, Activity Diagram dan Sequence Diagram*.

3. *Sprint backlog*

Dari hasil *sprint planning* yang telah ditetapkan terdapat 5 fitur yang akan dikerjakan untuk menyelesaikan *sprint* yang sedang berlangsung dengan membagi ke beberapa tim *developer*. *Output* dari *Sprint Backlog* adalah *User Interface* dan Rancangan Basis Data.

4. *Sprint Execution*

Pada tahap ini adalah melakukan *meeting* harian selama *sprint* berlangsung dengan waktu dan tempat yang sama. Menginspeksi perkembangan pekerjaan dari setiap tim *developer*. *Output* dari *Sprint Execution* adalah Fitur Master data, fitur konsultasi, fitur *task*, fitur *leads* dan fitur *opportunities*.

5. *Sprint review*

Menginspeksi *Increment* dan mengadaptasi *product backlog* bila diperlukan. Dihadiri oleh *stakeholder* dan *Product owner* menjelaskan keadaan *product backlog*, proyeksi target yang sudah di *delivery* dan *developer team* menyampaikan apa yang terjadi sepanjang *sprint* berlangsung dan mendemonstrasikan pekerjaan yang "Done". *Output* dari *Sprint Review* adalah *review* fitur Master data, *review* fitur konsultasi, *review* fitur *task*, *review* fitur *leads* dan *review* fitur *opportunities*.

6. *Increment*

Mengantarkan hasil dari *Sprint Execution* berupa fitur – fitur program yang dapat

digunakan dan sesuai dengan fungsi yang telah ditentukan pada *product backlog* dan dapat digunakan oleh pengguna.

7. *Sprint retrospective*

Menginspeksi bagaimana jalannya *sprint* terakhir yang terkait dengan orang-orang, hubungan antar *Scrum Team*, proses, dan alat-alat yang digunakan, mengidentifikasi dan mengurutkan hal utama yang berjalan dengan baik dan peningkatan yang berpotensi untuk dilakukan dan membuat perencanaan untuk implementasi peningkatan cara kerja *scrum team*. ". *Output* dari *Sprint retrospective* adalah evaluasi fitur Master data, evaluasi fitur konsultasi, evaluasi fitur *task*, *evaluasi* fitur *leads* dan evaluasi fitur *opportunities*.

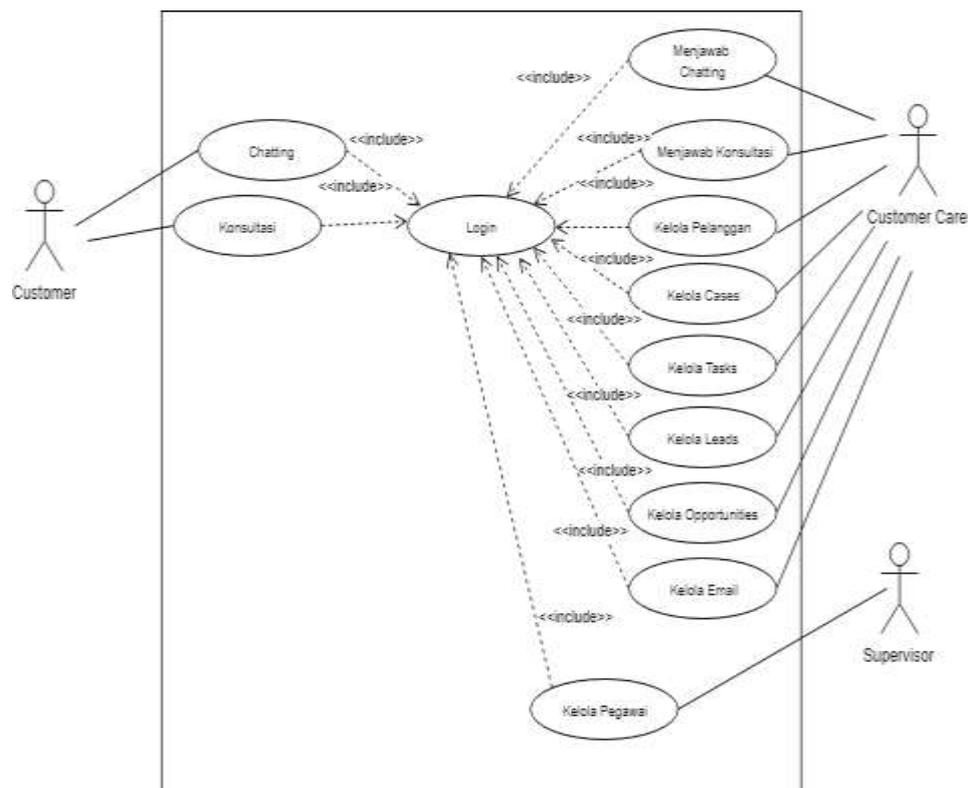
BAB IV

PEMBAHASAN DAN HASIL PENELITIAN

4.1 Tinjauan Perancangan Sistem Informasi *Customer Relationship Management*

Sistem informasi CRM yang dirancang pada PT. Rabana Sejahtera Indonesia bertujuan untuk menciptakan efektifitas pengelolaan data *customer* yang dilakukan oleh *customer service* yang sebelumnya masih menulis di buku besar kemudian dipindahkan ke dalam aplikasi *spreadsheet* agar data bisa diolah. Metode yang digunakan dalam perancangan sistem informasi CRM adalah *Scrum dengan pendekatan Agile Development*. *Agile development* merupakan metode iteratif dan bertahap dalam mengelola rancangan dan pembangunan sebuah sistem informasi.

Untuk menggambarkan kebutuhan – kebutuhan agar mencapai solusi penulis menggunakan diagram *use case*. Diagram *use case* ini menggambarkan rancangan dari sistem informasi CRM. Pada gambar 4.1 adalah gambaran diagram *use case* dari sistem informasi CRM di PT. Rabana Sejahtera Indonesia.



Gambar 4.1 *Use Case Diagram* Sistem Informasi CRM

Metode *Scrum* dipilih karena akan membuat proses membangun sistem informasi menjadi lebih cepat, karena sistem informasi dibangun berdasarkan skala prioritas. Fitur yang dibangun merupakan fitur yang sangat dibutuhkan oleh pengguna terlebih dahulu dan siap digunakan. Dari user *story product owner* maka *scrum master* melakukan analisis kepada kebutuhan yang ada lalu *scrum master* membuat *use case diagram*, *activity diagram*, *component diagram*, rancangan *user interface* serta rancangan *databasenya*. *Scrum master* mengkomunikasikan semua analisis dan desain setiap fitur kepada *Developer team* untuk dilakukan proses *development* sistem informasi CRM PT. Rabana Sejahtera Indonesia.

Gambar 4.1 menunjukkan *service – service* yang diusulkan berkaitan dengan Sistem Informasi CRM PT. Rabana Sejahtera Indonesia. Bagian *Customer Care* melakukan pengelolaan data *Customer* dengan menggunakan Metode CRM analitis agar data *Customer* dapat diolah dengan maksimal untuk meningkatkan keuntungan perusahaan dan menjaga hubungan baik dengan *Customer*. Langkah pertama adalah Supervisor menginput semua data pegawai kemudian bagian *Customer Care* menginputkan kontak *Customer* di menu pelanggan, kemudian dari semua kontak yang ada *Customer Care* melakukan komunikasi dengan *Customer* dengan target harian yang sudah ditentukan oleh perusahaan, dari komunikasi tersebut didapatkan beberapa *Customer* yang potensial. *service Tasks* digunakan untuk menyimpan data kinerja pegawai di PT. Rabana Sejahtera Indonesia dan *service Cases* digunakan untuk menangani kasus atau keluhan dari pelanggan. *Data Customer* yang potensial kemudian diinputkan ke menu *Lead*, komunikasi dengan *Customer* berlanjut bila *Customer* akan membeli produk maka data *Customer* tersebut dimasukkan di menu *Opportunities* di dalam menu ini status *Customer* akan diinputkan, *Customer* tersebut membeli produk “*Closing*” atau tidak. *Service email reminder* digunakan untuk mengirimkan *email* perperiode kepada pelanggan dan *service chatting* digunakan untuk melakukan komunikasi dengan pelanggan secara mendalam .

Rekomendasi untuk sistem informasi CRM PT. Rabana Sejahtera Indonesia dalam sisi perangkat lunak adalah *web server* menggunakan *Apache*, basis data menggunakan *MariaDB*, Sistem informasi CRM dibentuk menggunakan Bahasa pemrograman *PHP*

Framework Code Igniter sedangkan dari sisi perangkat keras adalah menggunakan Komputer server dengan spesifikasi *processor Intel Core I3 7th*, *RAM* minimal 4 (empat) *Giga Byte*, *Harddisk* Minimal 2 (dua) *Tera Byte*.

4.2 Scrum Team CRM Rabana

Scrum Team yang akan terlibat pada proses pengembangan sistem tersebut. Pada Tabel 4.1 akan dijelaskan nama, peran, jabatan dan tugas terhadap pengembangan sistem informasi *CRM*.

Tabel 4.1 *Scrum Team* PT. Rabana Sejahtera Indonesia

Nama	Peran	Jabatan	Tugas
<i>Hisyam</i>	<i>Product Owner</i>	Direktur <i>Marketing</i>	Menerima permintaan dari <i>client</i> , menerima kriteria produk, memprioritaskan <i>value</i> yang dibutuhkan dan menjalin hubungan baik dengan <i>stakeholders</i>
<i>Hardi</i>	<i>Scrum Master</i>	<i>Scrum Master</i>	Mengedukasi dan Melatih menggunakan <i>Scrum Framework</i> , memfasilitasi <i>meeting Scrum Team</i> , menghilangkan rintangan, menerima umpan balik dari <i>product owner</i> dan membantu menterjemahkan permintaan <i>client</i>
<i>Adib</i>	<i>Development Team</i>	<i>Backend Developer 1</i>	Membuat <i>service-service</i> sistem di sisi <i>backend</i> Dan bisa beradaptasi dengan perubahan yang sangat cepat
<i>Adik Darmadi</i>	<i>Development Team</i>	<i>Backend Developer 2</i>	Membuat <i>service-service</i> dan merancang <i>database</i> di sisi <i>backend</i> Dan bisa beradaptasi dengan perubahan yang sangat cepat
<i>Adi Setiawan</i>	<i>Development Team</i>	<i>Fronddend Developer 1</i>	Membuat <i>User Interface</i> dengan dinamis, mengintegrasikan dengan bagian <i>backend</i> dan bisa beradaptasi dengan perubahan yang sangat cepat
<i>Syamsudin</i>	<i>Development Team</i>	<i>Fronddend Developer 2</i>	Membuat <i>User Interface</i> dengan dinamis, mengintegrasikan dengan bagian <i>backend</i> Dan bisa beradaptasi dengan perubahan yang sangat cepat
<i>Deska</i>	<i>Development Team</i>	<i>Quality Engineer</i>	Menguji fitur fitur yang sudah dikembangkan oleh <i>backend developer</i> dan <i>fronddend developer</i> apakah sudah sesuai dengan kebutuhan <i>client</i> dan menghasilkan sebuah <i>value</i>

4.3 Pemilihan *Tools*

Sebelum masuk pada fase *sprint*. Langkah selanjutnya adalah mengelola *tools* baik itu perangkat lunak atau perangkat keras yang akan digunakan selama fase *sprint* berlangsung. Tabel 4.2 menjelaskan tentang *tools* apa saja yang akan digunakan untuk membangun sistem informasi CRM PT. Rabana Sejahtera Indonesia.

Tabel 4.2 *Tools* yang digunakan untuk pengembangan

No	Nama	Fungsi
1	<i>Visual Studio Code</i>	Teks editor kode Program
2	<i>Draw io</i>	Membuat rancangan <i>User Interface</i>
3	<i>Apache</i>	<i>Web server</i> pada <i>localhost</i>
4	<i>MariaDB</i>	<i>Database server</i>
5	<i>PHP Framework Code Igniter</i>	Bahasa Pemrograman yang digunakan
6	<i>Google Chrome</i>	Browser untuk menampilkan sistem informasi yang sedang dibuat.
7	<i>Windows 10</i>	Sistem operasi
8	<i>Visual Paradigm 15.2</i>	Membuat rancangan UML (<i>Unified Modelling Language</i>)

4.4 Lingkungan Kerja Fisik

Lingkungan kerja untuk mengerjakan *sprint* perlu ditentukan untuk mendukung *scrum* tim yang sudah dibentuk. Sebaiknya seluruh anggota tim termasuk *product owner* memiliki ruangan khusus untuk mengerjakan *sprint* dan *mereview sprint* tersebut. Ruang kerja yang efektif untuk *scrum team* adalah sebagai berikut :

1. Memiliki ruang yang cukup luas untuk masing-masing anggota
2. Papan tulis yang cukup luas
3. Adanya kamera digital
4. Adanya perlengkapan *scrum* seperti *sticky notes* dan spidol
5. Lemari buku untuk menyimpan buku referensi dan pemodelan
6. Meja yang besar
7. *Projector*
8. Buku referensi

4.5 *Sprint 1 Core Sistem dan Login Sistem*

Core sistem berfungsi untuk mengelola semua fitur yang ada di sistem informasi

CRM PT. Rabana Sejahtera Indonesia serta login sistem berdasarkan hak aksesnya di dalam Sistem Informasi CRM Rabana Sejahtera Indonesia.

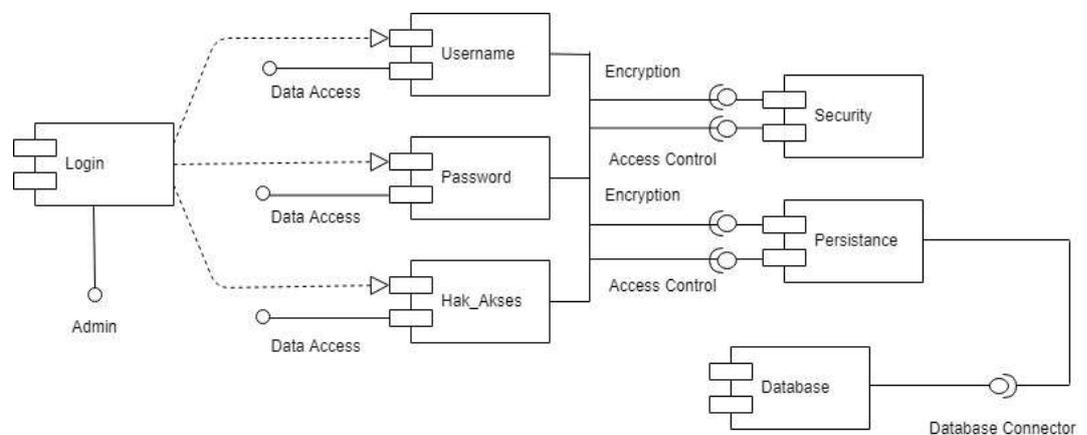
1. Product Backlog

Tabel 4.3 *User Story* core sistem dan login sistem

ID	User Story
US0001	Core sistem informasi CRM
US0002	Menu login sistem informasi CRM

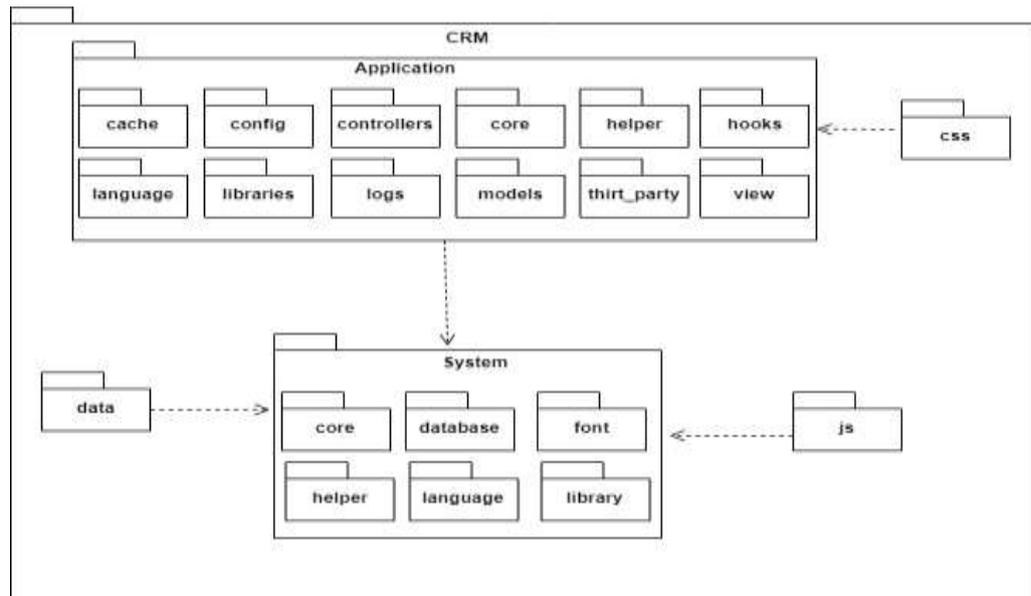
2. Sprint Planning

Pada tahap ini merupakan desain sistem informasi CRM dari modul core sistem dan login sistem menghasilkan *component* diagram yang dapat di lihat di Gambar 4.2, *package* diagram yang dapat di lihat di Gambar 4.3 dan *deployment* diagram yang dapat di lihat di Gambar 4.4 untuk memudahkan *software developer* dalam membangun kode program sistem informasi CRM.



Gambar 4.2 *Component diagram* login sistem

Gambar 4.2 menunjukkan node-node yang ada dalam proses *login*, di dalam menu pegawai terdapat *component user login* yang berisikan data diri pengguna sistem seperti *username*, *password* dan hak akses.

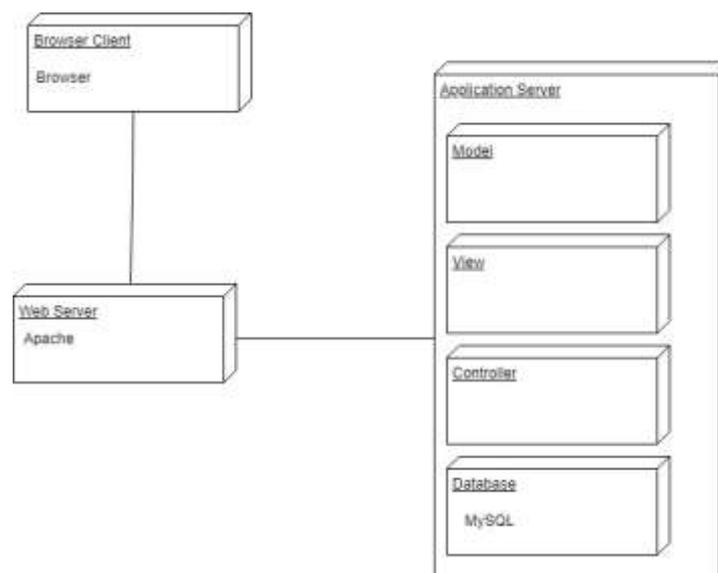


Gambar 4.3 Package diagram core sistem

Pada gambar 4.3 menjelaskan tentang package diagram CRM. *Package diagram* (diagram paket) merupakan salah satu jenis *UML* yang digunakan untuk mengelompokkan elemen-elemen model dari *use case* ataupun *class diagram*. *Package diagram* digunakan pada kumpulan sistem yang besar. atau *package diagram* disebut sekelompok elemen-elemen model. sebuah paket dapat berisi elemen-elemen model yang berlainan, termasuk paket-paket untuk menciptakan/menggambarkan sifat hirarki. sebuah paket diberi nama yang menggambarkan isinya. Susunan dari *package* Sistem Informasi CRM adalah sebagai berikut :

1. *Package application*: di sinilah aplikasi yang akan kita bangun diletakkan.
 - a. *Package config* – tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari *database*, *router* dan *autoload* aplikasi.
 - b. *Package controllers* – tempat menyimpan semua file controller.
 - c. *Package errors* – tempat menyimpan semua *template error* aplikasi.
 - d. *Package helpers* – tempat menyimpan helper yang bukan berasal dari CI.
 - e. *Package hooks* – tempat menyimpan hook yang digunakan untuk mengubah alur fungsi dari *core Code Igniter*.
 - f. *Package language* – tempat menyimpan bahasa-bahasa yang akan di gunakan.
 - g. *Package libraries* – tempat menyimpan semua *library* buatan kita sendiri.

- h. *Package models* – tempat menyimpan semua model.
 - i. *Package views* – tempat menyimpan semua *file view* aplikasi.
2. *Package system*: menyimpan semua file baik itu file aplikasi yang dibuat maupun *core framework*-nya.
 - a. *Package core* – tempat menyimpan semua *core* yang ada di CI
 - b. *Package database* – tempat menyimpan semua *database drivers* dan class yang akan digunakan.
 - c. *Package fonts* – tempat menyimpan semua *font* yang digunakan *image manipulation library*.
 - d. *Package helpers* – tempat menyimpan semua helper *core* CI.
 - e. *Package language* – tempat menyimpan semua *language core* CI.
 - f. *Package libraries* – tempat menyimpan semua *library core* CI
 3. *Package css*: tempat menyimpan semua file *css*.
 4. *Package js*: tempat menyimpan semua file *javascript*.
 5. *Package data*: tempat menyimpan semua data.



Gambar 4.4 *Deployment diagram core sistem*

Deployment diagram adalah diagram yang digunakan untuk memetakan *software* ke *processing node*. Menunjukkan konfigurasi elemen pemroses pada saat *run time* dan *software* yang ada di dalamnya. Pada gambar 4.4 merupakan *Deployment diagram* pada

sistem informasi CRM PT. Rabana Sejahtera Indonesia menjelaskan tentang alur pemrosesan tiap *node* dimulai dari *browser client* melakukan *request* kepada web server kemudian diproses oleh *application server* dan menyimpannya kedalam *database*.

3. *Sprint Planning*

a. *User Interface Login*



Gambar 4.5 *User Interface Login sistem*

Gambar 4.5 merupakan rancangan *user interface* dari menu *login*. Di rancangan ini terdapat kolom *header* yang berisikan tulisan menu *customer relationship management* dan sebuah form yang berisikan *username* dan *password* serta tombol yang bertuliskan *login* untuk dapat masuk kedalam sistem informasi.

b. *User Interface Menu*



Gambar 4.6 *User Interface Menu Sistem Informasi CRM*

Gambar 4.6 merupakan rancangan *user interface* dari menu utama. Di rancangan ini terdapat kolom *header* yang berisikan tulisan menu *home* dan sebuah *sidebar menu* yang berisikan menu-menu utama dari sistem informasi CRM.

4. *Sprint Execution*

Pada *sprint execution* berisikan beberapa *service* untuk *core* sistem dan login sistem. Tabel 4.4 adalah daftar *service* dari *sprint execution core* sistem dan login sistem.

Tabel 4.4 *Sprint execution core* sistem dan login sistem

ID	Story	Nama Service	To Do List
US0001	Core sistem informasi CRM	Menu Service	<ol style="list-style-type: none"> 1. Membuat table data pegawai 2. Membuat <i>base url</i> dan koneksi ke <i>database</i> 3. Menyetting <i>dependency</i> yang dibutuhkan 4. Menyetting tampilan CSS untuk sistem Informasi
US0002	Menu login sistem informasi CRM	Login Service	<ol style="list-style-type: none"> 1. Membuat form login untuk user 2. Membuat validasi data untuk <i>login</i> ke sistem informasi CRM

5. *Sprint Review*

Pada *sprint review* ditabel 4.5 berisi *review – review* dari *product owner* dan *stakeholders* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan oleh *product owner* sebelumnya.

Tabel 4.5 *Sprint review core* sistem dan login sistem

ID	Story	Review
US0001	Core sistem informasi CRM	Core sistem informasi CRM berjalan dengan baik
US0002	Menu <i>login</i> sistem informasi CRM	User bisa <i>login</i> ke dalam sistem

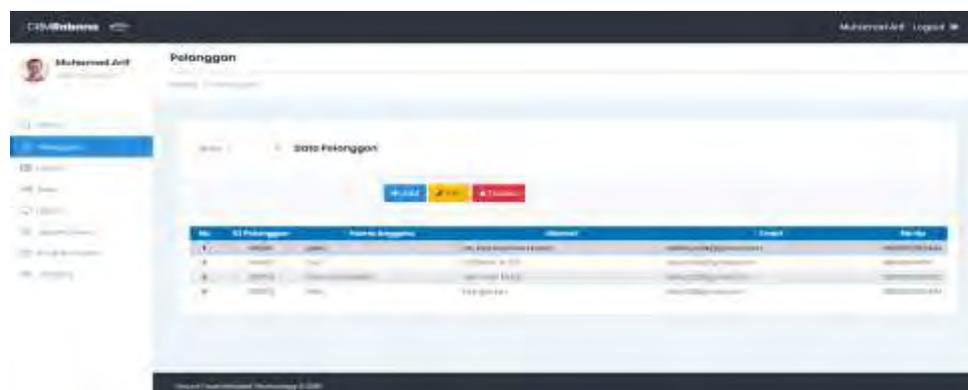
6. *Increment*

Pada tahap *increment* adalah pendistribusian fitur yang sudah dikerjakan oleh *scrum team* kepada pengguna dan *stakeholder*. Jika ada fitur yang masih kurang sesuai dengan kebutuhan pengguna maka fitur dapat dimasukkan kembali pada *sprint* berikutnya. Fitur yang sudah selesai dikerjakan adalah login menu yang bisa ditunjukkan pada gambar 4.7 dan menu sistem informasi CRM pada gambar 4.8 :



Gambar 4.7 Login sistem

Gambar 4.7 merupakan *User Interface* dari menu login Sistem Informasi CRM yang digunakan untuk masuk kedalam sistem.terdapat 3 (tiga) hak akses di sistem informasi CRM yaitu : admin, supervisor dan pelanggan.



Gambar 4.8 Menu sistem informasi CRM

Gambar 4.8 merupakan *User Interface* dari menu utama Sistem Informasi CRM yang digunakan untuk memetakan menu yang ada di sistem informasi CRM. Letak menu berada pada sisi kiri layar agar pengguna lebih mudah dalam menggunakan sistem.

7. Sprint Retrospective

Pada *sprint retrospective* ditabel 4.6 berisi *evaluasi* dari *scrum team* dan *product owner* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan sebelumnya.

Tabel 4.6 *Sprint retrospective core sistem dan login sistem*

ID	Story	Evaluasi
US0001	Core sistem informasi CRM	Sudah sesuai dengan <i>product backlog</i>
US0002	Menu <i>login</i> sistem informasi CRM	Sudah sesuai dengan <i>product backlog</i>

4.6 *Sprint 2 Modul Master Data*

4.6.1 *Fitur Master Pegawai*

Fitur master pegawai berfungsi untuk mengelola data pegawai yang ada di PT. Rabana Sejahtera Indonesia serta memetakan hak aksesnya di dalam Sistem Informasi CRM Rabana Sejahtera Indonesia.

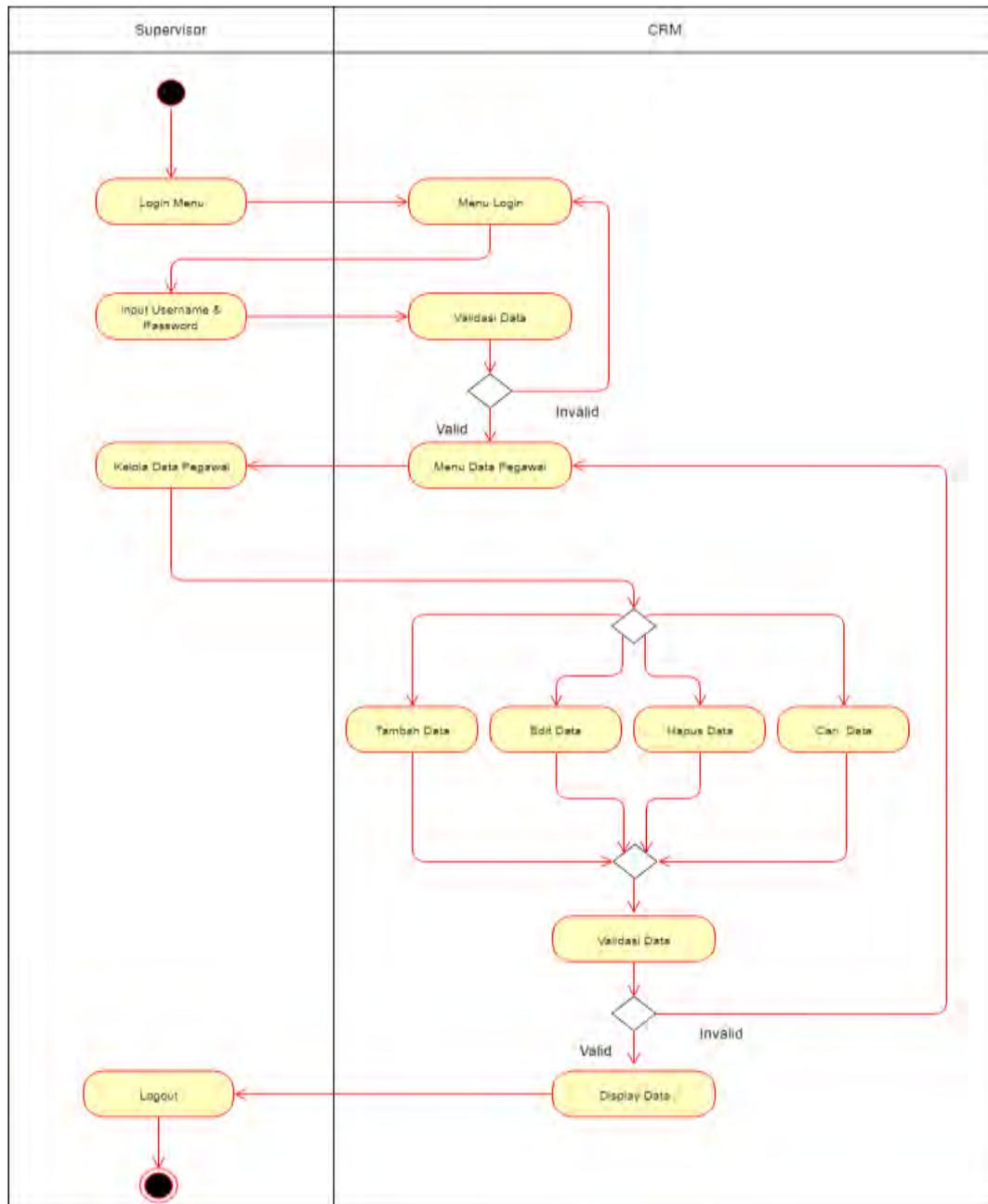
1. *Product Backlog*

Tabel 4.7 *User Story* fitur master Pegawai

ID	User Story
US0003	Supervisor dapat menginput data pegawai
US0004	Supervisor dapat melakukan <i>editing</i> terhadap data pegawai bila terjadi kesalahan
US0005	Supervisor dapat menonaktifkan pegawai jika sudah tidak aktif di sistem
US0006	Supervisor dapat mencari data pegawai di kolom pencarian

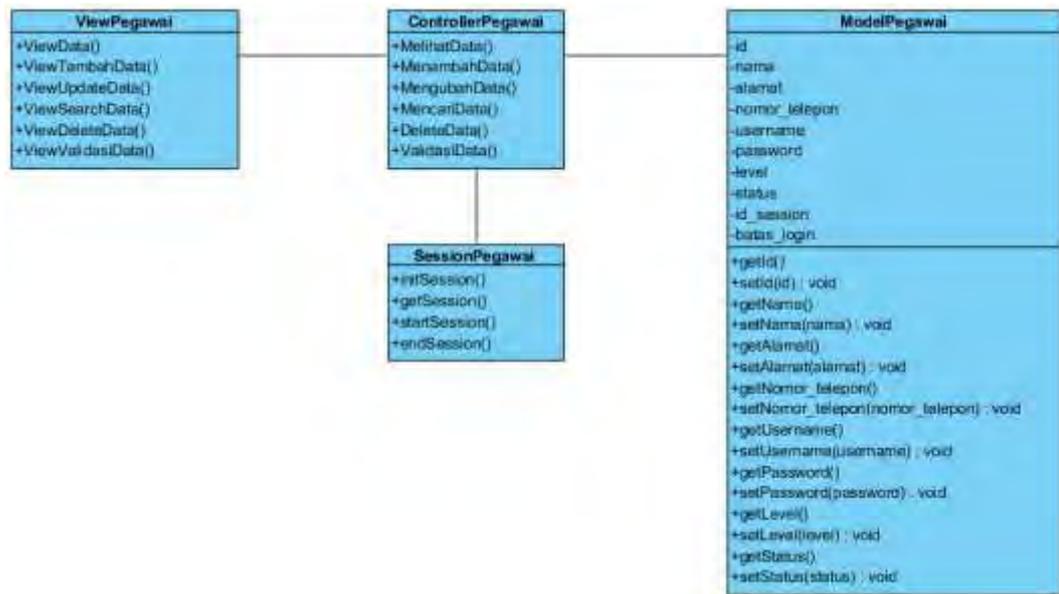
2. *Sprint Planning*

Pada tahap *sprint Planning* merupakan desain sistem informasi CRM dari modul master data dan menghasilkan *activity* diagram yang dapat di lihat di Gambar 4.9, *class* diagram yang dapat di lihat di Gambar 4.10 dan *component* diagram yang dapat di lihat di Gambar 4.11 untuk memudahkan *software developer* dalam membangun kode program.



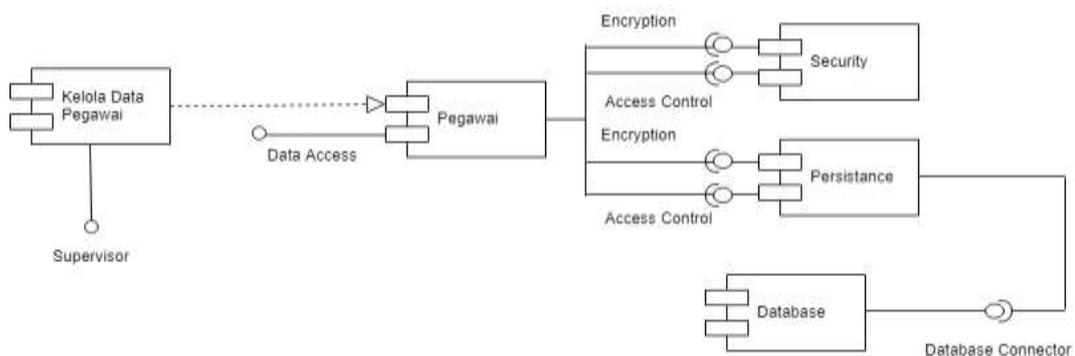
Gambar 4.9 Activity Diagram Master Data Pegawai

Gambar 4.9 menunjukkan *activity diagram* pengelolaan menu pegawai yang dilakukan oleh Admin CS. Sebelumnya masuk ke dalam sistem. Jika proses validasi berhasil maka sistem akan menampilkan halaman *home* dan admin dapat memilih menu pengelolaan pegawai. Ketika admin memilih menu pengelolaan pegawai maka admin dapat melakukan pengelolaan pegawai dengan menambahkan, merubah dan menghapus pegawai. Sistem akan melakukan pengecekan bila terjadi duplikasi data dan akan memberikan notifikasi kepada pengguna sistem.



Gambar 4.10 *Class Diagram* Master Data Pegawai

Gambar 4.10 menunjukkan operasi dari setiap *class* di mana pada *class* Pegawai memiliki beberapa *class* yang mempunyai peran masing-masing seperti *class* *viewPegawai* berfungsi untuk menampilkan *display* pegawai, form tambah dan edit pegawai, *class* *controllerPegawai* berfungsi untuk menjalankan logika dari proses yang di *request* oleh *class* *view*, *class* *ModelPegawai* berfungsi untuk menyimpan entitas-entitas yang berkaitan dengan *menu* pegawai seperti nama, alamat, nomor telepon, tanggal lahir, tempat lahir, email, website dan status pegawai, *class* *SessionPegawai* berfungsi untuk menyimpan *session user*.



Gambar 4.11 *Component Diagram* Master Data Pegawai

Gambar 4.11 menunjukkan node-node yang ada dalam proses pegawai, di dalam menu pegawai terdapat *component* pegawai yang berisikan data diri pegawai seperti nama, alamat, tanggal lahir, tempat lahir, hak akses dan status pelanggan.

3. *Sprint Backlog*

a. *User Interface*

HEADER					
DATA MASTER PEGAWAI					
		+ Tambah		Search	
No	ID Pegawai	Nama	Alamat	Level	Aksi
1	XXXXXXXX	XXXXXXXX	XXXXXXXX	admin	Edit
2	XXXXXXXX	XXXXXXXX	XXXXXXXX	supervisor	Edit
3	XXXXXXXX	XXXXXXXX	XXXXXXXX	admin	Aksi
				Previous	Next
FOOTER					

Gambar 4.12 *User Interface List* Master data pegawai

Gambar 4.12 merupakan rancangan *user interface* dari menu pegawai. Di rancangan ini ada sebuah *list* yang berisikan id pegawai, nama pegawai, alamat, level dan beberapa tombol untuk memanipulasi data pegawai.

HEADER	
DATA MASTER PEGAWAI	
ID Pegawai	x (50)
Nama	x (50)
Alamat	x (100)
Telepon	x (15)
Username	x (20)
Password	x (50)
Level	x (10)
Status	enum { Aktif / Tidak Aktif }
<input type="button" value="Batal"/> <input type="button" value="Simpan"/>	
FOOTER	

Gambar 4.13 *User Interface Form* Tambah data pegawai

Gambar 4.13 merupakan rancangan *user interface* dari menu pegawai. Di rancangan ini ada sebuah form yang berisikan id pegawai, nama pegawai, alamat, nomor telepon, *username*, *password*, level dan status pegawai, kolom *header* yang berisikan tulisan menu pelanggan dan tombol simpan.

2. Rancangan Basis Data

Tabel 4.8 Tabel Data Master Pegawai

No	Nama field	Tipe data	Panjang Record	Index
1	id_pegawai	Varchar	50	<i>primary key</i>
2	Nama	Varchar	50	
3	Alamat	Varchar	100	
4	no_telp	Varchar	15	
5	Username	Varchar	20	
6	Password	Varchar	50	
7	Level	Varchar	10	
8	Status	Enum	-	
9	id_session	Varchar	100	<i>foreign key</i>
10	Batas_login	Int	1	

4. Sprint Execution

Pada *sprint execution* berisikan beberapa *service* untuk master data pegawai. Tabel

4.9 adalah daftar *service* dari *sprint execution* fitur master data pegawai

Tabel 4.9 *Sprint execution* fitur master Pegawai

ID	Story	Nama Service	To Do List
US0003	Supervisor dapat menginput data pegawai	<i>POST</i> Tambah data pegawai	<ol style="list-style-type: none"> 1. Membuat <i>table</i> master data pegawai 2. Membuat <i>list</i> data pegawai 3. Membuat form input master data pegawai
US0004	Supervisor dapat melakukan <i>editing</i> terhadap data pegawai bila terjadi kesalahan	<i>PUT</i> Update data pegawai	<ol style="list-style-type: none"> 1. Membuat form edit master data pegawai
US0005	Supervisor dapat menonaktifkan pegawai jika sudah tidak aktif di sistem	<i>DELETE</i> data pegawai	<ol style="list-style-type: none"> 1. Membuat form blokir pegawai yang sudah tidak aktif
US0006	Supervisor dapat mencari data pegawai di kolom pencarian	<i>GET</i> Cari data pegawai	<ol style="list-style-type: none"> 1. Membuat form pencarian untuk data pegawai

5. Sprint Review

Pada *sprint review* ditabel 4.10 berisi *review – review* dari *product owner* dan *stakeholders* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan oleh *product owner* sebelumnya.

Tabel 4.10 *Sprint review* fitur master Pegawai

ID	Story	Review
US0003	Supervisor dapat menginput data pegawai	Data pegawai berhasil di inputkan kedalam sistem
US0004	Supervisor dapat melakukan <i>editing</i> terhadap data pegawai bila terjadi kesalahan	Data pegawai berhasil di edit
US0005	Supervisor dapat menonaktifkan pegawai jika sudah tidak aktif di sistem	Data pegawai berhasil di nonaktifkan serta tidak bisa login ke dalam sistem
US0006	Supervisor dapat mencari data pegawai di kolom pencarian	Pencarian data pegawai dapat menemukan data sesuai dengan parameter yang diinputkan

6. *Increment*

Pada tahap *increment* adalah pendistribusian fitur yang sudah dikerjakan oleh *scrum team* kepada pengguna dan *stakeholder*. Jika ada fitur yang masih kurang sesuai dengan kebutuhan pengguna maka fitur dapat dimasukkan kembali pada *sprint* berikutnya. Fitur yang sudah selesai dikerjakan adalah fitur *list* data pegawai yang bisa ditunjukkan pada gambar 4.14, fitur form tambah data pegawai pada gambar 4.15 dan fitur edit data pegawai pada gambar 416 :

Gambar 4.14 *Fitur List* data pegawai

Gambar 4.14 merupakan *User Interface* dari menu *list* data pegawai yang digunakan untuk melihat data pegawai PT. Rabana Sejahtera Indonesia. Di dalam menu *list* pegawai mempunyai 3 (tiga) tombol penting yaitu tombol *add* untuk menambah data pegawai, tombol *edit* untuk mengedit data pegawai dan tombol *delete* untuk menghapus data pegawai. Pengguna menu kelola data pegawai adalah supervisor PT. Rabana Sejahtera Indonesia.

Gambar 4.15 *Fitur Form* Tambah data pegawai

Gambar 4.15 merupakan *User Interface* dari menu tambah data pegawai yang digunakan untuk menambahkan data pegawai. Pengguna menu tambah data pegawai adalah supervisor PT. Rabana Sejahtera Indonesia.

Gambar 4.16 *Fitur Form* Edit data pegawai

Gambar 4.16 merupakan *User Interface* dari menu edit data pegawai yang digunakan untuk mengedit data pegawai jika terjadi kesalahan menginput data pegawai. Pengguna menu *edit* data pegawai adalah supervisor PT. Rabana Sejahtera Indonesia.

7. *Sprint Retrospective*

Pada *sprint retrospective* ditabel 4.11 berisi *evaluasi* dari *scrum team* dan *product owner* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan sebelumnya.

Tabel 4.11 *Sprint retrospective* fitur master Pegawai

ID	Story	Evaluasi
US0003	Supervisor dapat menginput data pegawai	Sudah sesuai dengan <i>product backlog</i>

Tabel 4.11 *Sprint retrospective* fitur master Pegawai

US0004	Supervisor dapat melakukan <i>editing</i> terhadap data pegawai bila terjadi kesalahan	Sudah sesuai dengan <i>product backlog</i>
US0005	Supervisor dapat menonaktifkan pegawai jika sudah tidak aktif di sistem	Sudah sesuai dengan <i>product backlog</i>
US0006	Supervisor dapat mencari data pegawai di kolom pencarian	Sudah sesuai dengan <i>product backlog</i>

4.6.2 Fitur Master Pelanggan

Fitur master Pelanggan berfungsi untuk mengelola data Pelanggan yang ada di PT. Rabana Sejahtera Indonesia serta memetakan hak aksesnya di dalam Sistem Informasi CRM Rabana Sejahtera Indonesia.

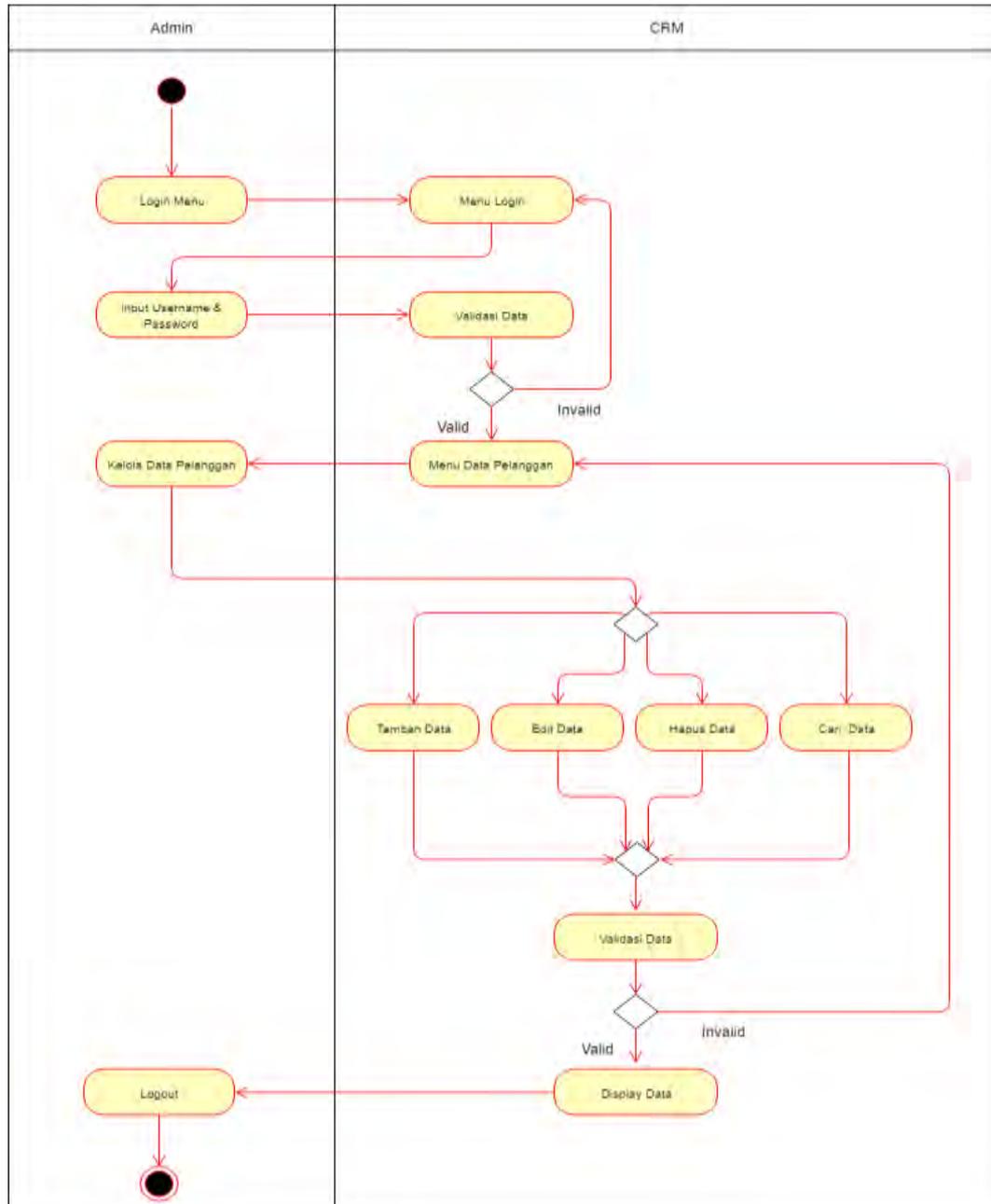
1. *Product Backlog*

Tabel 4.12 *User Story* fitur master pelanggan

<i>ID</i>	<i>User Story</i>
US0007	Admin dapat menginput data Pelanggan
US0008	Admin dapat melakukan <i>editing</i> terhadap data pelanggan bila terjadi kesalahan
US0009	Admin dapat menonaktifkan pegawai jika sudah tidak aktif di sistem
US00010	Admin dapat mencari data pegawai di kolom pencarian

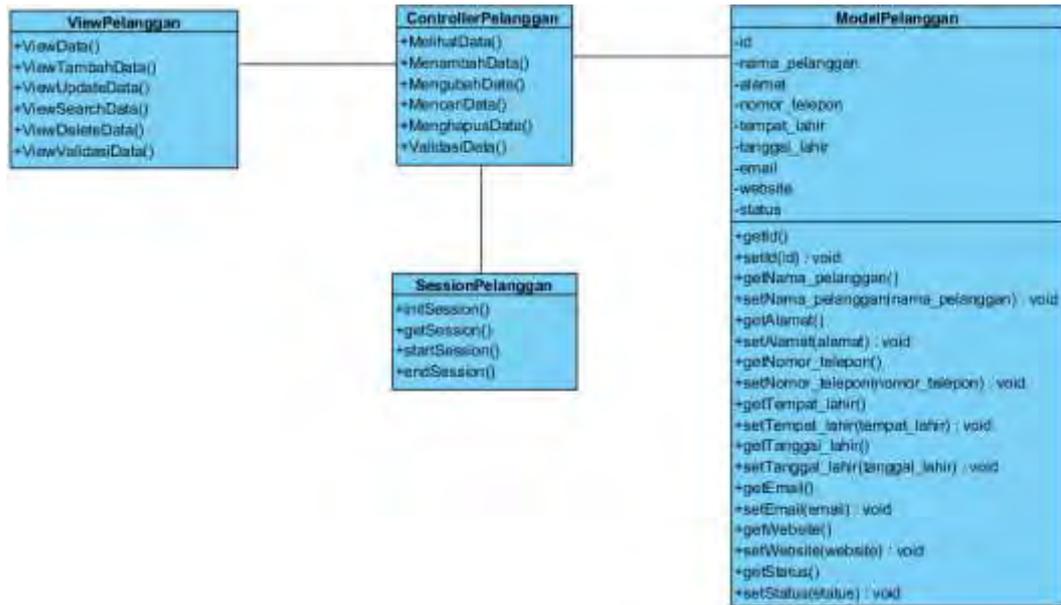
2. *Sprint Planning*

Pada tahap *sprint planning* merupakan desain sistem informasi CRM dari modul master data dan menghasilkan *activity* diagram yang dapat di lihat di Gambar 4.17, *class* diagram yang dapat di lihat di Gambar 4.18 dan *component* diagram yang dapat di lihat di Gambar 4.19 untuk memudahkan *software developer* membangun kode program sistem informasi CRM.



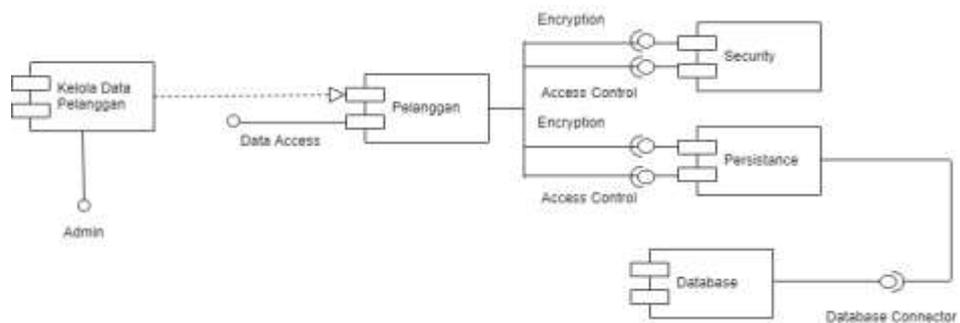
Gambar 4.17 Activity Diagram Master Data Pelanggan

Gambar 4.17 menunjukkan *activity diagram* pengelolaan menu pelanggan yang dilakukan oleh Admin CS. Sebelumnya masuk ke dalam sistem. Jika proses validasi berhasil maka sistem akan menampilkan halaman *dashboard* dan admin dapat memilih menu pengelolaan pelanggan. Ketika admin memilih menu pengelolaan pelanggan maka admin dapat melakukan pengelolaan pelanggan dengan menambahkan, merubah dan menghapus pelanggan. Sistem akan melakukan pengecekan bila terjadi duplikasi data dan akan memberikan notifikasi kepada pengguna sistem.



Gambar 4.18 *Class Diagram* Master Data Pelanggan

Gambar 4.18 menunjukkan operasi dari setiap *class* di mana pada *class* Pelanggan memiliki beberapa *class* yang mempunyai peran masing-masing seperti *class* *viewPelanggan* berfungsi untuk menampilkan *display* pelanggan, form tambah dan edit pelanggan, *class* *controllerPelanggan* berfungsi untuk menjalankan logika dari proses yang di *request* oleh *class* *view*, *class* *ModelPelanggan* berfungsi untuk menyimpan entitas-entitas yang berkaitan dengan *menu* pelanggan seperti nama, alamat, nomor telepon, tanggal lahir, tempat lahir, *email*, *website* dan status pelanggan, *class* *SessionPelanggan* berfungsi untuk menyimpan *session user*.



Gambar 4.19 *Component Diagram* Master Data Pelanggan

Gambar 4.19 menunjukkan node-node yang ada dalam proses pelanggan, di dalam menu pelanggan terdapat *component* pelanggan yang berisikan data diri pelanggan seperti nama, alamat, tanggal lahir, tempat lahir dan status pelanggan.

3. *Sprint Backlog*
 - a. *User Interface*

HEADER							
							Edit Profile
Menu Pelanggan							
Tambah Data						Cari	
No	Id Pelanggan	Nama	Tempat Lahir	Tanggal	Alamat	No Kontak	Aksi
1	xw/9999	xxxxxxxxxxxxxx	xxxxxxxxxx	99/99/9999	xxxxxxxxxx	99999999999999	Edit

Previous | 1 | Next

FOOTER

Gambar 4.20 *User Interface List* Master data Pelanggan

Gambar 4.20 merupakan rancangan *user interface* dari menu pelanggan. Di rancangan ini ada sebuah *list* yang berisikan id pelanggan, nama, alamat, tempat dan tanggal lahir dan beberapa tombol untuk memanipulasi data pelanggan.

HEADER	
Edit Profile	
Menu Pelanggan	
id Pelanggan	xw/9999
Nama	xxxxxxxxxxxxxxxxxxxxxx
Website	xxxxxx
Alamat	xxxx xxxxx xxxxxxx
Kontak	9999-9999-9999
Email	xxxxxxxxxx
Tempat Lahir	xxxxxxxxxx
Tanggal Lahir	99/99/9999
Simpan	
FOOTER	

Gambar 4.21 *User Interface Form* Tambah dan Edit data pelanggan

Gambar 4.21 merupakan rancangan *user interface* dari menu pelanggan. Di rancangan ini ada sebuah form yang berisikan id pelanggan, nama pelanggan, website, alamat pelanggan, kontak pelanggan, email, tempat lahir dan tanggal lahir pelanggan, kolom *header* yang berisikan tulisan menu pelanggan dan tombol simpan.

b. Rancangan Basis Data

Tabel 4.13 Tabel Data Master Pelanggan

No	Nama field	Tipe data	Panjang Record	Index
1	Id	Varchar	50	<i>primary key</i>
2	nama_pelanggan	Varchar	100	
3	Alamat	Varchar	100	
4	nomor_telepon	Varchar	20	
5	tempat_lahir	Varchar	50	
6	tanggal_lahir	Date	-	
7	Website	Varchar	50	
8	Email	Varchar	50	
9	Status	Enum	-	

4. *Sprint Execution*

Pada *sprint execution* berisikan beberapa *service* untuk master data pelanggan.

Tabel 4.14 adalah daftar *service* dari *sprint execution* fitur master data pelanggan

Tabel 4.14 *Sprint execution* fitur master pelanggan

ID	Story	Nama Service	To Do List
US0007	Admin dapat menginput data pelanggan	<i>POST</i> Tambah data pelanggan	<ol style="list-style-type: none"> 1. Membuat table master data pelanggan 2. Membuat list data pelanggan 3. Membuat form input master data pelanggan
US0008	Admin dapat melakukan <i>editing</i> terhadap data pelanggan bila terjadi kesalahan	<i>PUT</i> Update data pelanggan	<ol style="list-style-type: none"> 1. Membuat form edit master data pelanggan
US0009	Admin dapat menonaktifkan pelanggan jika sudah tidak aktif di sistem	<i>DELETE</i> data pelanggan	<ol style="list-style-type: none"> 1. Membuat form blokir pelanggan yang sudah tidak aktif
US00010	Admin dapat mencari data pelanggan di kolom pencarian	<i>GET</i> Cari data pelanggan	<ol style="list-style-type: none"> 1. Membuat form pencarian untuk data pelanggan

5. *Sprint Review*

Pada *sprint review* ditabel 4.15 berisi *review – review* dari *product owner* dan *stakeholders* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan oleh *product owner* sebelumnya.

Tabel 4.15 *Sprint review* fitur master pelanggan

ID	Story	Review
US0007	Admin dapat menginput data pelanggan	Data pelanggan berhasil di inputkan kedalam sistem
US0008	Admin dapat melakukan <i>editing</i> terhadap data pelanggan bila terjadi kesalahan	Data pelanggan berhasil di edit
US0009	Admin dapat menonaktifkan pelanggan jika sudah tidak aktif di sistem	Data pelanggan berhasil di nonaktifkan
US00010	Admin dapat mencari data pelanggan di kolom pencarian	Pencarian data pelanggan dapat menemukan data sesuai dengan parameter yang diinputkan

6. *Increment*

Pada tahap *increment* adalah pendistribusian fitur yang sudah dikerjakan oleh *scrum team* kepada pengguna dan *stakeholder*. Jika ada fitur yang masih kurang sesuai dengan kebutuhan pengguna maka fitur dapat dimasukkan kembali pada *sprint* berikutnya. Fitur yang sudah selesai dikerjakan adalah fitur *list* data pelanggan yang bisa ditunjukkan pada gambar 4.22, fitur form tambah data pelanggan pada gambar 4.23 dan fitur edit data pelanggan pada gambar 4.24 :

Gambar 4.22 *Fitur List* data pelanggan

Gambar 4.22 merupakan *User Interface* dari menu *list* data pelanggan yang digunakan untuk melihat data pelanggan PT. Rabana Sejahtera Indonesia. Di dalam menu *list* pelanggan mempunyai 3 (tiga) tombol penting yaitu tombol *add* untuk menambah data pelanggan, tombol *edit* untuk mengedit data pelanggan dan tombol *delete* untuk menghapus data pelanggan. Pengguna menu kelola data pelanggan adalah admin CS PT. Rabana Sejahtera Indonesia.

Gambar 4.23 *Fitur Form* Tambah data pelanggan

Gambar 4.23 merupakan *User Interface* dari menu tambah data pelanggan yang digunakan untuk menambah data pelanggan kedalam sistem. Pengguna menu tambah data pelanggan adalah admin CS PT. Rabana Sejahtera Indonesia.

Gambar 4.24 *Fitur Form* Edit data pelanggan

Gambar 4.24 merupakan *User Interface* dari menu edit data pelanggan yang digunakan untuk mengedit data pelanggan jika terjadi kesalahan menginput data pelanggan. Pengguna menu *edit* data pelanggan adalah admin CS PT. Rabana Sejahtera Indonesia.

4.7 *Sprint 3* Menu *Cases* dan Menu *Tasks*

4.7.1 Menu *Cases*

Menu *Cases* berfungsi untuk mengelola data *cases* yang diterima dari pelanggan yang ada di PT. Rabana Sejahtera Indonesia seperti konsultasi produk, masalah produk, pengembalian produk dan sebagainya..

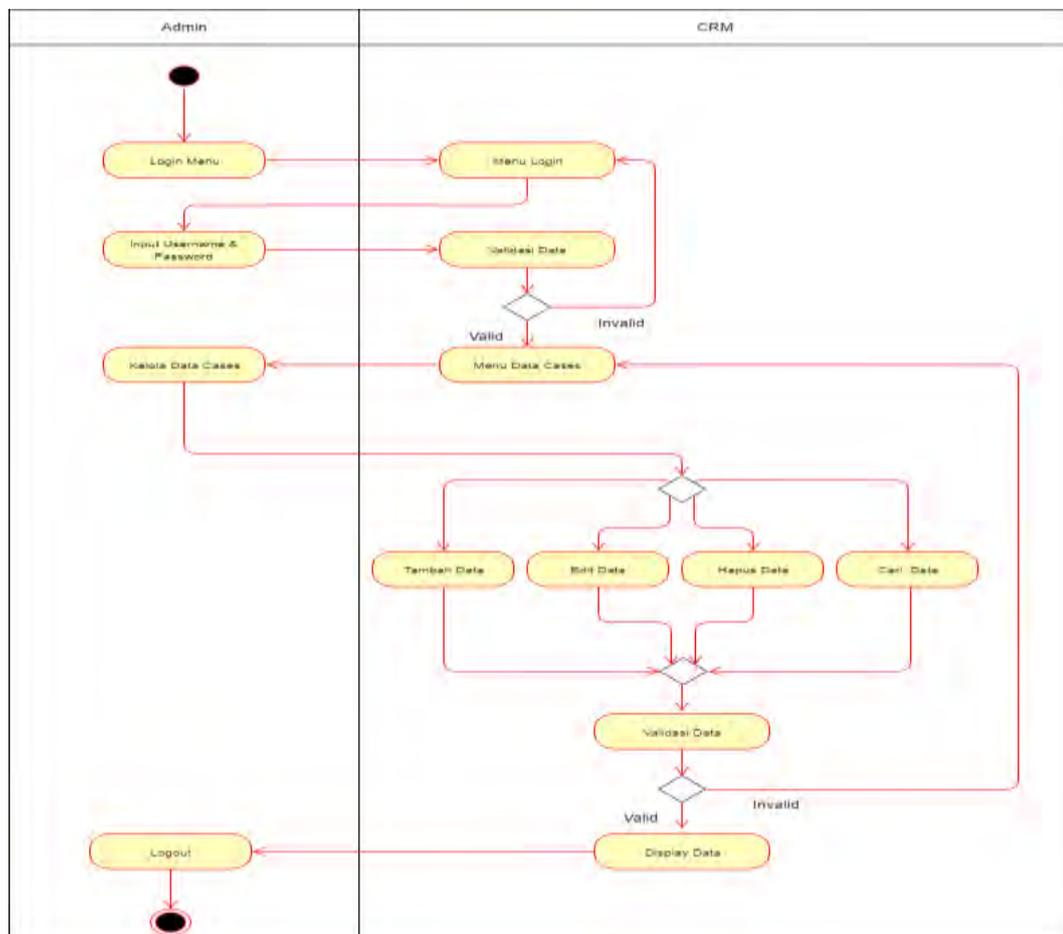
1. Product Backlog

Tabel 4.16 User Story Menu Cases

ID	User Story
US0011	Admin dapat menginput data <i>cases</i>
US0012	Admin dapat melakukan <i>editing</i> terhadap data <i>cases</i> bila terjadi kesalahan
US0013	Admin dapat menghapus data <i>cases</i> jika sudah tidak aktif di sistem
US0014	Admin dapat mencari data <i>cases</i> berdasarkan area pelanggan

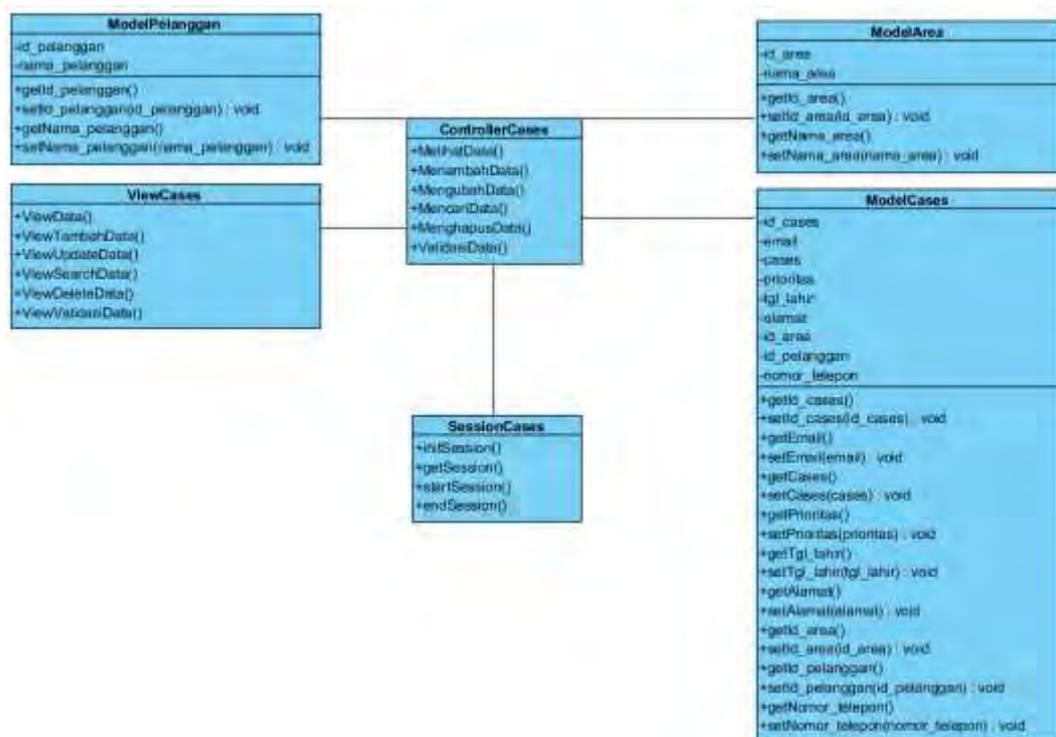
2. Sprint Planning

Pada tahap *sprint Planning* merupakan desain sistem informasi CRM dari menu *cases* dan menghasilkan *activity diagram* yang dapat di lihat di Gambar 4.25, *class diagram* yang dapat di lihat di Gambar 4.26 dan *component diagram* yang dapat di lihat di Gambar 4.27 untuk memudahkan *software developer* dalam membangun kode program sistem informasi CRM.



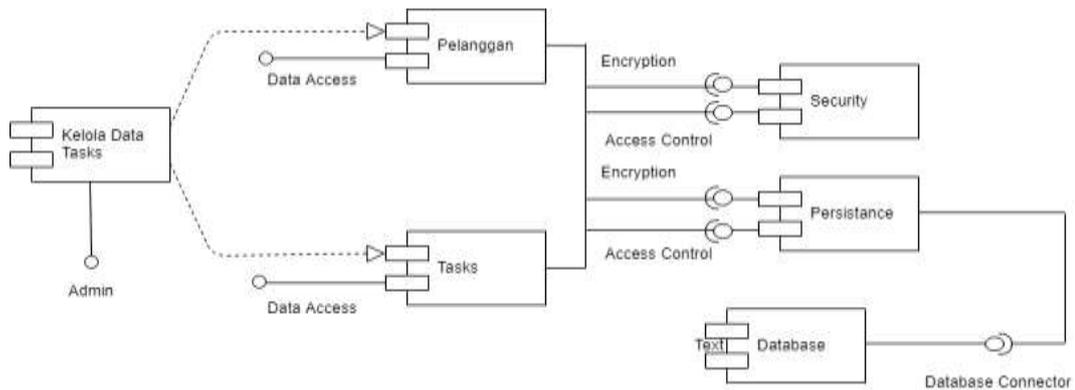
Gambar 4.25 Activity Diagram Menu Cases

Gambar 4.25 menunjukkan *activity diagram* pengelolaan menu cases yang dilakukan oleh Admin CS. Sebelumnya masuk ke dalam sistem. Jika proses validasi berhasil maka sistem akan menampilkan halaman *dashboard* dan admin dapat memilih menu pengelolaan cases. Ketika admin memilih menu pengelolaan cases maka admin dapat melakukan pengelolaan cases dengan menambahkan, merubah dan menghapus cases. Sistem akan melakukan pengecekan bila terjadi duplikasi data dan akan memberikan notifikasi kepada pengguna sistem.



Gambar 4.26 *Class Diagram* Menu Cases

Gambar 4.26 menunjukkan operasi dari setiap *class* di mana pada *class cases* memiliki beberapa *class* yang mempunyai peran masing-masing seperti *class viewCases* berfungsi untuk menampilkan *display cases*, form tambah dan edit cases, *class controllerCases* berfungsi untuk menjalankan logika dari proses yang di *request* oleh *class view*, *class ModelCases* berfungsi untuk menyimpan entitas-entitas yang berkaitan dengan *menu cases* seperti nama, cases dan status cases, *class SessionCases* berfungsi untuk menyimpan *session user*.



Gambar 4.27 Component Diagram Menu Cases

Gambar 4.27 menunjukkan node-node yang ada dalam proses cases, di dalam menu cases terdapat *component cases* yang berisikan cases yang telah diterima dari pelanggan, *component pelanggan* yang berisi data pelanggan dan *component area* yang berisi area.

3. Sprint Backlog

a. User Interface



Gambar 4.28 User Interface List Menu Cases

Gambar 4.28 merupakan rancangan *user interface* dari menu cases pelanggan. Di rancangan ini ada sebuah *list* yang berisikan id cases, nama pelanggan, kasus, tanggal, prioritas, status dan beberapa tombol untuk memanipulasi data kasus.

Gambar 4.29 *User Interface Form* Tambah data Cases

Gambar 4.29 merupakan rancangan *user interface* dari menu cases. Di rancangan ini ada sebuah form yang berisikan id cases, nama pelanggan, kontak, prioritas, area dan status, kolom *header* yang berisikan tulisan menu cases dan tombol simpan.

b. Rancangan Basis Data

Tabel 4.17 Tabel Data Menu Cases

No	Nama field	Tipe data	Panjang Record	Index
1	Id	Varchar	30	<i>primary key</i>
2	Email	Varchar	30	
3	Nama_kasus	Varchar	100	
4	Prioritas	Varchar	30	
5	Tgl_kasus	Date	15	
6	Id_area	Varchar	15	<i>foreign key</i>
7	Id_pelanggan	Varchar	15	<i>foreign key</i>
8	Keterangan	Enum	250	
9	Status	Varchar	10	

4. *Sprint Execution*

Pada *sprint execution* berisikan beberapa *service* untuk menu cases. Tabel 4.18 adalah daftar *service* dari *sprint execution* menu cases.

Tabel 4.18 *Sprint execution* menu cases

ID	Story	Nama Service	To Do List
US0011	Admin dapat menginput data cases	POST Tambah data cases	1. Membuat <i>table cases</i> 2. Membuat <i>list data cases</i> 3. Membuat form input cases

Tabel 4.18 *Sprint execution* menu cases

US0012	Admin dapat melakukan <i>editing</i> terhadap data cases bila terjadi kesalahan	<i>PUT</i> Update data cases	1. Membuat form edit cases
US0013	Admin dapat menghapus data cases	<i>DELETE</i> data cases	1. Membuat form <i>delete cases</i>
US0014	Admin dapat mencari data cases berdasarkan area pelanggan	<i>GET</i> Cari data by area	1. Membuat form pencarian berdasarkan area pelanggan

5. *Sprint Review*

Pada *sprint review* ditabel 4.19 berisi *review – review* dari *product owner* dan *stakeholders* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan oleh *product owner* sebelumnya.

Tabel 4.19 *Sprint review* menu cases

ID	Story	Review
US0011	Admin dapat menginput data cases	Data cases berhasil di inputkan kedalam system
US0012	Admin dapat melakukan <i>editing</i> terhadap data cases bila terjadi kesalahan	Data cases berhasil di edit
US0013	Admin dapat menghapus data cases jika sudah tidak aktif di sistem	Data cases berhasil di hapus
US0014	Admin dapat mencari data cases berdasarkan area pelanggan	Pencarian data cases dapat menemukan data sesuai dengan area yang dipilih

6. *Increment*

Pada tahap *increment* adalah pendistribusian fitur yang sudah dikerjakan oleh *scrum team* kepada pengguna dan *stakeholder*. Fitur yang sudah selesai dikerjakan adalah fitur *list* menu cases yang bisa ditunjukkan pada gambar 4.30, fitur form tambah data cases pada gambar 4.31 dan fitur edit data cases pada gambar 4.32 :



No	Nomor Cases	Kasus	Nama Pelanggan	Pekerjaan	Nomor Handphone	Status
1	10001	Komputer	Andi	Marketing	123456789	Selesai
2	10002	Komputer	Budi	IT Support	123456789	Pending
3	10003	Perawatan	Cici	Marketing	123456789	Selesai
4	10004	Perawatan	Dani	Marketing	123456789	Selesai

Gambar 4.30 *Fitur List* Menu Cases

Gambar 4.30 merupakan *User Interface* dari menu *list* data *Cases* yang digunakan untuk melihat data *Cases* pelanggan PT. Rabana Sejahtera Indonesia. Di dalam menu *list* *Cases* mempunyai 3 (tiga) tombol penting yaitu tombol *add* untuk menambah data *Cases*, tombol edit untuk mengedit data *Cases* dan tombol *delete* untuk menghapus data *Cases*. Pengguna menu kelola data *Cases* adalah admin CS PT. Rabana Sejahtera Indonesia.

Gambar 4.31 *Fitur Form* Tambah data *Cases*

Gambar 4.31 merupakan *User Interface* dari menu tambah data *cases* yang digunakan untuk menambah data *cases* pelanggan kedalam sistem. Pengguna menu tambah data *cases* adalah admin CS PT. Rabana Sejahtera Indonesia.

Gambar 4.32 *Fitur Form* Edit data *Cases*

Gambar 4.32 merupakan *User Interface* dari menu edit data *Cases* yang digunakan untuk mengedit data *Cases* pelanggan jika terjadi kesalahan menginput data *Cases*. Pengguna menu *edit* data *Cases* adalah admin CS PT. Rabana Sejahtera Indonesia.

7. *Sprint Retrospective*

Pada *sprint retrospective* ditabel 4.20 berisi *evaluasi* dari *scrum team* dan *product owner* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan sebelumnya.

Tabel 4.20 *Sprint retrospective* fitur menu *cases*

ID	Story	Evaluasi
US0011	Admin dapat menginput data <i>cases</i>	Sudah sesuai dengan <i>product backlog</i>
US0012	Admin dapat melakukan <i>editing</i> terhadap data <i>cases</i> bila terjadi kesalahan	Sudah sesuai dengan <i>product backlog</i>
US0013	Admin dapat menghapus data <i>cases</i> jika sudah tidak aktif di sistem	Sudah sesuai dengan <i>product backlog</i>
US0014	Admin dapat mencari data <i>cases</i> berdasarkan area pelanggan	Sudah sesuai dengan <i>product backlog</i>

4.7.2 Menu *Tasks*

Menu *Tasks* berfungsi untuk mengelola data *tugas tiap tiap admin cs* yang ada di PT. Rabana Sejahtera Indonesia seperti agenda menelpon pelanggan, agenda bertemu pelanggan, *meeting* dan sebagainya.

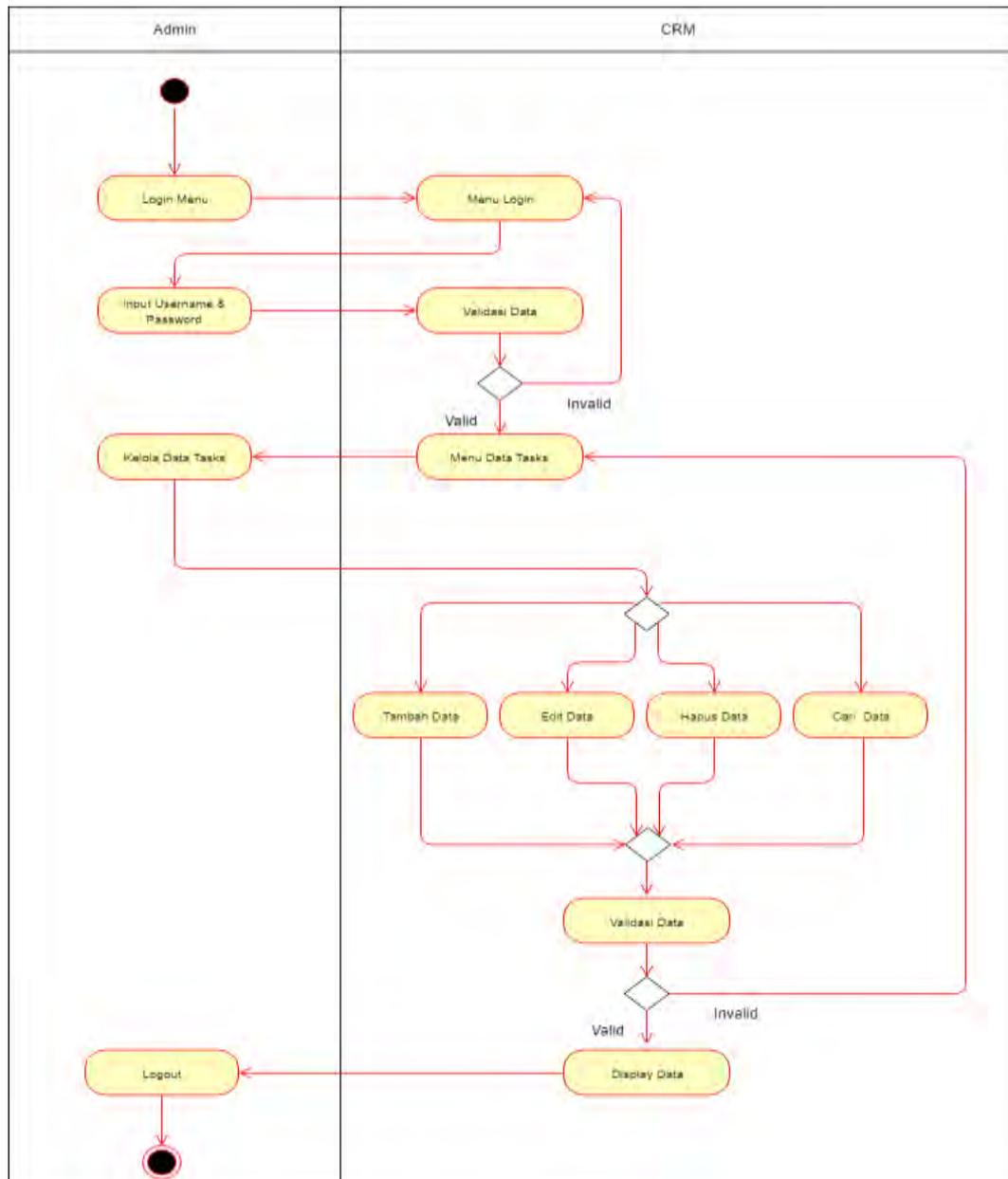
1. *Product Backlog*

Tabel 4.21 *User Story* Menu *tasks*

ID	User Story
US0015	Admin dapat menginput data <i>tasks</i>
US0016	Admin dapat melakukan <i>editing</i> terhadap data <i>tasks</i> bila terjadi kesalahan
US0017	Admin dapat menghapus data <i>tasks</i> jika sudah tidak aktif di sistem
US0018	Admin dapat mencari data <i>tasks</i> berdasarkan area pelanggan

2. *Sprint Planning*

Pada tahap *sprint Planning* merupakan desain sistem informasi *CRM* dari menu *tasks* dan menghasilkan *activity diagram* yang dapat di lihat di Gambar 4.33, *class diagram* yang dapat di lihat di Gambar 4.34 dan *component diagram* yang dapat di lihat di Gambar 4.35 untuk memudahkan *software developer* dalam membangun kode program sistem informasi *CRM*.



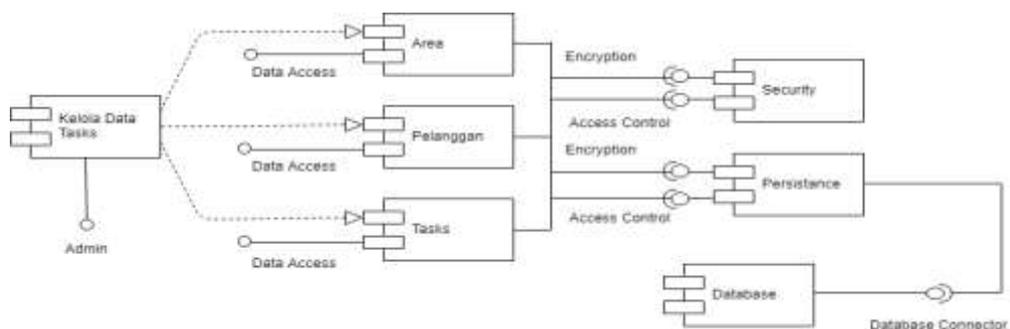
Gambar 4.33 Activity Diagram Menu tasks.

Gambar 4.33 menunjukkan *activity diagram* pengelolaan menu *tasks* yang dilakukan oleh Admin CS. Sebelumnya masuk ke dalam sistem. Jika proses validasi berhasil maka sistem akan menampilkan halaman *dashboard* dan admin dapat memilih menu pengelolaan *tasks*. Ketika admin memilih menu pengelolaan *tasks* maka admin dapat melakukan pengelolaan *tasks* dengan menambahkan, merubah dan menghapus *tasks*. Sistem akan melakukan pengecekan bila terjadi duplikasi data dan akan memberikan notifikasi kepada pengguna sistem.



Gambar 4.34 Class Diagram Menu tasks

Gambar 4.34 menunjukkan operasi dari setiap class di mana pada class tasks memiliki beberapa class yang mempunyai peran masing-masing seperti class viewTasks berfungsi untuk menampilkan display tasks, form tambah dan edit tasks, class controllerTasks berfungsi untuk menjalankan logika dari proses yang di request oleh class view, class ModelTasks berfungsi untuk menyimpan entitas-entitas yang berkaitan dengan menu tasks seperti nama pelanggan, tasks, tanggal tasks dan status tasks, class SessionTasks berfungsi untuk menyimpan session user.



Gambar 4.35 Component Diagram Menu tasks

Gambar 4.35 menunjukkan node-node yang ada dalam proses *Tasks*, di dalam menu *Tasks* terdapat *component Tasks* yang berisikan *Tasks yang telah dikerjakan*, *component pelanggan yang berisi data pelanggan* dan *component area yang berisi area*.

3. *Sprint Backlog*

a. *User Interface*

Gambar 4.36 *User Interface List Menu tasks*

Gambar 4.36 merupakan rancangan *user interface* dari menu *tasks* pegawai. Di rancangan ini ada sebuah *list* yang berisikan id *tasks*, nama pelanggan, tugas, tanggal, prioritas, status dan beberapa tombol untuk memanipulasi data tugas.

Gambar 4.37 *User Interface Form Tambah data tasks*

Gambar 4.37 merupakan rancangan *user interface* dari menu *tasks*. Di rancangan ini ada sebuah form yang berisikan id *tasks*, nama pelanggan, kontak, prioritas, area dan status, kolom *header* yang berisikan tulisan menu *tasks* dan tombol simpan.

b. Rancangan Basis Data

Tabel 4.22 Tabel Data Menu *tasks*

No	Nama field	Tipe data	Panjang Record	Index
1	Id	Varchar	30	<i>primary key</i>
2	Email	Varchar	30	
3	Nama_ <i>tasks</i>	Varchar	100	
4	Prioritas	Varchar	30	
5	Tgl_kasus	Date	15	
6	Id_pelanggan	Varchar	15	<i>foreign key</i>
7	Keterangan	Enum	250	
8	Status	Varchar	10	

4. *Sprint Execution*

Pada *sprint execution* berisikan beberapa *service* untuk menu *tasks*. Tabel 4.23 adalah daftar *service* dari *sprint execution* menu *tasks*.

Tabel 4.23 *Sprint execution* menu *tasks*

ID	Story	Nama Service	To Do List
US0015	Admin dapat menginput data <i>tasks</i>	<i>POST</i> Tambah data <i>tasks</i>	1. Membuat <i>table tasks</i> 2. Membuat <i>list data tasks</i> 3. Membuat form input <i>tasks</i>
US0016	Admin dapat melakukan <i>editing</i> terhadap data <i>tasks</i> bila terjadi kesalahan	<i>PUT</i> Update data <i>tasks</i>	1. Membuat form edit <i>tasks</i>
US0017	Admin dapat menghapus data <i>tasks</i>	<i>DELETE</i> data <i>tasks</i>	1. Membuat form <i>delete tasks</i>
US0018	Admin dapat mencari data <i>tasks</i> berdasarkan area	<i>GET</i> Cari data by area	1. Membuat form pencarian berdasarkan area

5. *Sprint Review*

Pada *sprint review* ditabel 4.24 berisi *review – review* dari *product owner* dan *stakeholders* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan oleh *product owner* sebelumnya.

Tabel 4.24 *Sprint review* menu *tasks*

ID	Story	Review
US0015	Admin dapat menginput data <i>tasks</i>	Data <i>tasks</i> berhasil di inputkan kedalam sistem
US0016	Admin dapat melakukan <i>editing</i> terhadap data <i>tasks</i> bila terjadi kesalahan	Data <i>tasks</i> berhasil di edit

Tabel 4.24 *Sprint review* menu *tasks*

US0017	Admin dapat menghapus data <i>tasks</i> jika sudah tidak aktif di sistem	Data <i>tasks</i> berhasil di hapus
US0018	Admin dapat mencari data <i>tasks</i> berdasarkan area pelanggan	Pencarian data <i>tasks</i> dapat menemukan data sesuai dengan area yang dipilih

6. *Increment*

Pada tahap *increment* adalah pendistribusian fitur yang sudah dikerjakan oleh *scrum team* kepada pengguna dan *stakeholder*. Jika ada fitur yang masih kurang sesuai dengan kebutuhan pengguna maka fitur dapat dimasukkan kembali pada *sprint* berikutnya. Fitur yang sudah selesai dikerjakan adalah fitur *list* menu *tasks* yang bisa ditunjukkan pada gambar 4.38, fitur form tambah data *tasks* pada gambar 4.39 dan fitur edit data *tasks* pada gambar 4.40 :

Gambar 4.38 *Fitur List* Menu *tasks*

Gambar 4.38 merupakan *User Interface* dari menu *list* data *tasks* yang digunakan untuk melihat data *tasks* pegawai PT. Rabana Sejahtera Indonesia. Di dalam menu *list tasks* mempunyai 3 (tiga) tombol penting yaitu tombol *add* untuk menambah data *tasks*, tombol *edit* untuk mengedit data *tasks* dan tombol *delete* untuk menghapus data *tasks*. Pengguna menu kelola data *tasks* adalah admin CS PT. Rabana Sejahtera Indonesia.

Gambar 4.39 *Fitur Form* Tambah data *tasks*

Gambar 4.39 merupakan *User Interface* dari menu tambah data *tasks* yang digunakan untuk menambah data *tasks* pegawai kedalam sistem. Pengguna menu tambah data *tasks* adalah admin CS PT. Rabana Sejahtera Indonesia.

Gambar 4.40 Fitur Form Edit data *tasks*

Gambar 4.40 merupakan *User Interface* dari menu edit data *tasks* yang digunakan untuk mengedit data *tasks* pegawai jika terjadi kesalahan menginput data *tasks*. Pengguna menu *edit* data *tasks* adalah admin CS PT. Rabana Sejahtera Indonesia.

7. *Sprint Retrospective*

Pada *sprint retrospective* ditabel 4.25 berisi *evaluasi* dari *scrum team* dan *product owner* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan sebelumnya.

Tabel 4.25 *Sprint retrospective* fitur menu *tasks*

ID	Story	Evaluasi
US0015	Admin dapat menginput data <i>tasks</i>	Sudah sesuai dengan <i>product backlog</i>
US0016	Admin dapat melakukan <i>editing</i> terhadap data <i>tasks</i> bila terjadi kesalahan	Sudah sesuai dengan <i>product backlog</i>
US0017	Admin dapat menghapus data <i>tasks</i> jika sudah tidak aktif di sistem	Sudah sesuai dengan <i>product backlog</i>
US0018	Admin dapat mencari data <i>tasks</i> berdasarkan area pelanggan	Sudah sesuai dengan <i>product backlog</i>

4.8 *Sprint 4 Menu Leads dan Menu Opportunities*

4.8.1 Menu *Leads*

Menu *Leads* berfungsi untuk mengelola data *leads* pelanggan potensial yang ada di PT. Rabana Sejahtera Indonesia.

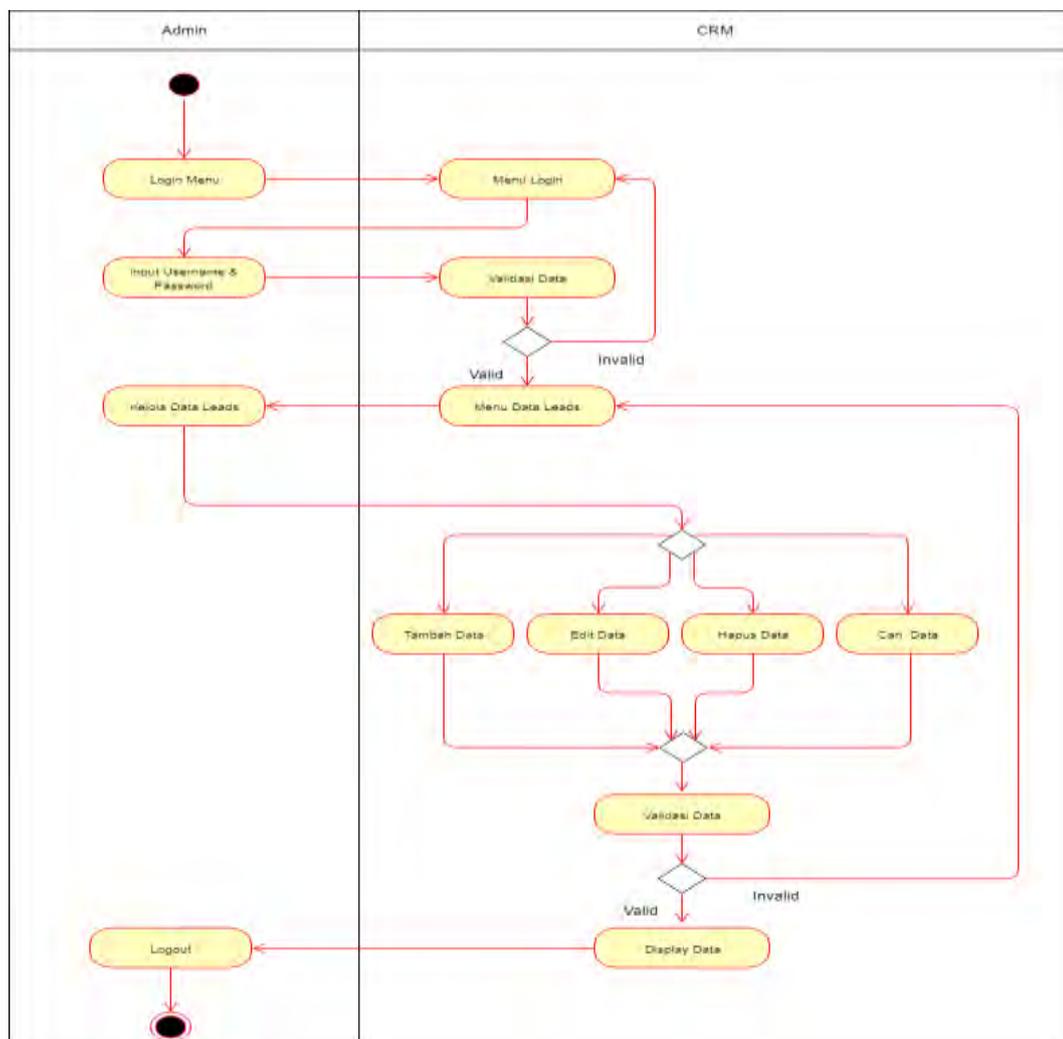
1. Product Backlog

Tabel 4.26 User Story Menu Leads

ID	User Story
US0019	Admin dapat menginput data <i>leads</i>
US0020	Admin dapat melakukan <i>editing</i> terhadap data <i>leads</i> bila terjadi kesalahan
US0021	Admin dapat menghapus data <i>leads</i> jika sudah tidak aktif di system
US0022	Admin dapat mencari data <i>leads</i> berdasarkan area pelanggan

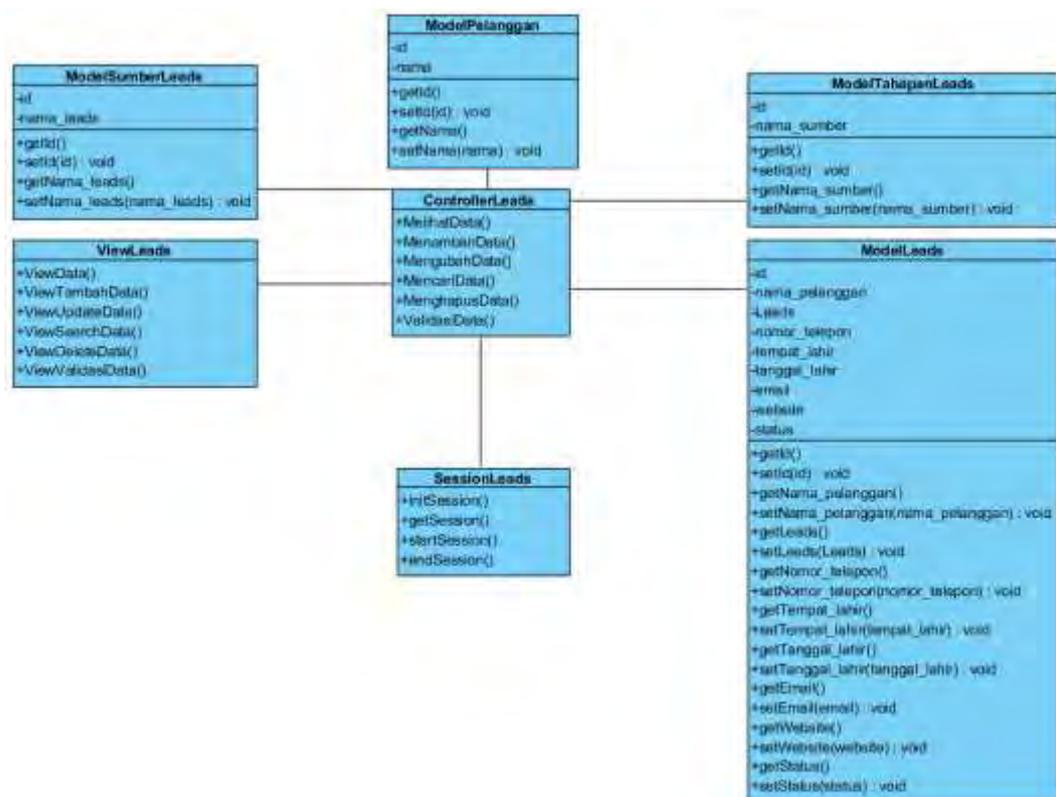
2. Sprint Planning

Pada tahap *sprint Planning* merupakan desain sistem informasi CRM dari menu *Leads* dan menghasilkan *activity diagram* yang dapat di lihat di Gambar 4.41, *class diagram* yang dapat di lihat di Gambar 4.42 dan *component diagram* yang dapat di lihat di Gambar 4.43.



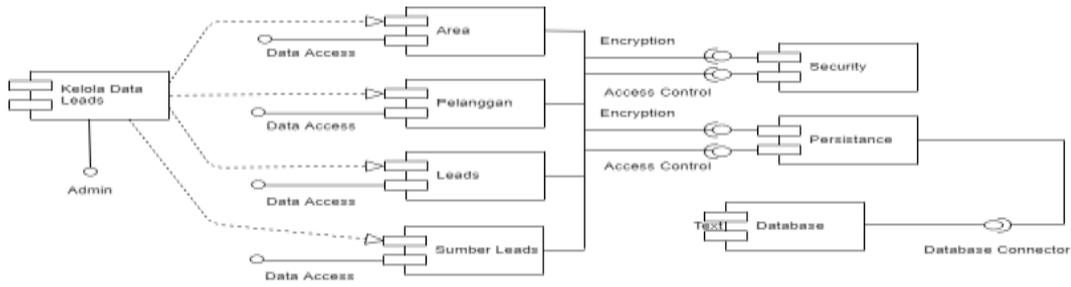
Gambar 4.41 Activity Diagram Menu Leads

Gambar 4.41 menunjukkan *activity diagram* pengelolaan menu *leads* yang dilakukan oleh Admin CS. Sebelumnya masuk ke dalam sistem. Jika proses validasi berhasil maka sistem akan menampilkan halaman *dashboard* dan admin dapat memilih menu pengelolaan *leads*. Ketika admin memilih menu pengelolaan *leads* maka admin dapat melakukan pengelolaan *leads* dengan menambahkan, merubah dan menghapus *leads*. Sistem akan melakukan pengecekan bila terjadi duplikasi data dan akan memberikan notifikasi kepada pengguna sistem.



Gambar 4.42 Class Diagram Menu Leads

Gambar 4.42 menunjukkan operasi dari setiap *class* di mana pada *class leads* memiliki beberapa *class* yang mempunyai peran masing-masing seperti *class viewLeads* berfungsi untuk menampilkan *display leads*, form tambah dan edit *leads*, *class controllerLeads* berfungsi untuk menjalankan logika dari proses yang di *request* oleh *class view*, *class ModelLeads* berfungsi untuk menyimpan entitas-entitas yang berkaitan dengan *menu leads* seperti nama, *leads*, sumber *leads* dan status *leads*, *class SessionLeads* berfungsi untuk menyimpan *session user*.



Gambar 4.43 Component Diagram Menu Leads

Gambar 4.43 menunjukkan node-node yang ada dalam proses Leads, di dalam menu Leads terdapat component Leads yang berisikan pesan Leads, component pelanggan yang berisi data pelanggan, component area yang berisi area dan component sumber leads yang berisi sumber leads pelanggan tersebut.

3. Sprint Backlog

a. User Interface



Gambar 4.44 User Interface List Menu Leads

Gambar 4.44 merupakan rancangan user interface dari menu leads pelanggan. Di rancangan ini ada sebuah list yang berisikan id leads, nama pelanggan, leads, tanggal, prioritas, status dan beberapa tombol untuk memanipulasi data leads.



Gambar 4.45 User Interface Form Tambah data Leads

Gambar 4.45 merupakan rancangan *user interface* dari menu *leads*. Di rancangan ini ada sebuah form yang berisikan id *leads*, nama pelanggan, kontak, prioritas, area dan status, kolom *header* yang berisikan tulisan menu *leads* dan tombol simpan.

b. Rancangan Basis Data

Tabel 4.27 Tabel Data Menu *Leads*

No	Nama field	Tipe data	Panjang Record	Index
1	Id	Varchar	30	<i>primary key</i>
2	Email	Varchar	30	
3	Nama_Leads	Varchar	100	
4	Prioritas	Varchar	30	
5	Tgl_leads	Date	15	
6	Id_area	Varchar	15	<i>foreign key</i>
7	Id_pelanggan	Varchar	15	<i>foreign key</i>
8	Id_sumber_leads	Varchar	15	<i>foreign key</i>
9	Keterangan	Enum	250	
10	Status	Varchar	10	

4. *Sprint Execution*

Pada *sprint execution* berisikan beberapa *service* untuk menu *Leads*. Tabel 4.28 adalah daftar *service* dari *sprint execution* menu *Leads*.

Tabel 4.28 *Sprint execution* menu *Leads*

ID	Story	Nama Service	To Do List
US0011	Admin dapat menginput data <i>Leads</i>	<i>POST</i> Tambah data <i>Leads</i>	1. Membuat <i>table Leads</i> 2. Membuat <i>list</i> data <i>Leads</i> 3. Membuat form input <i>Leads</i>
US0012	Admin dapat melakukan <i>editing</i> terhadap data <i>Leads</i> yang salah	<i>PUT</i> Update data <i>Leads</i>	1. Membuat form edit <i>Leads</i>
US0013	Admin dapat menghapus data <i>Leads</i>	<i>DELETE</i> data <i>Leads</i>	1. Membuat form <i>delete Leads</i>
US0014	Admin dapat mencari data <i>Leads</i> berdasarkan area pelanggan	<i>GET</i> Cari data by area	1. Membuat form pencarian berdasarkan area pelanggan

5. *Sprint Review*

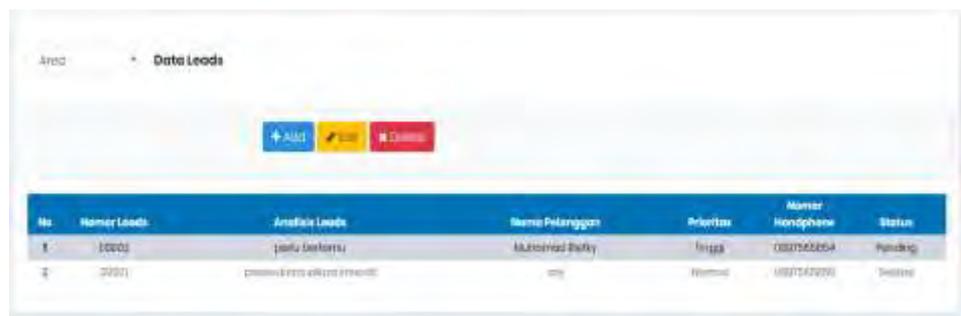
Pada *sprint review* ditabel 4.29 berisi *review – review* dari *product owner* dan *stakeholders* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan oleh *product owner* sebelumnya.

Tabel 4.29 *Sprint review* menu *Leads*

ID	Story	Review
US0019	Admin dapat menginput data <i>leads</i>	Data <i>Leads</i> berhasil di inputkan kedalam sistem
US0020	Admin dapat melakukan <i>editing</i> terhadap data <i>leads</i> bila terjadi kesalahan	Data <i>Leads</i> berhasil di edit
US0021	Admin dapat menghapus data <i>leads</i> jika sudah tidak aktif di sistem	Data <i>Leads</i> berhasil di hapus
US0022	Admin dapat mencari data <i>leads</i> berdasarkan area pelanggan	Pencarian data <i>Leads</i> dapat menemukan data sesuai dengan area yang dipilih

6. *Increment*

Pada tahap *increment* adalah pendistribusian fitur yang sudah dikerjakan oleh *scrum team* kepada pengguna dan *stakeholder*. Jika ada fitur yang masih kurang sesuai dengan kebutuhan pengguna maka fitur dapat dimasukkan kembali pada *sprint* berikutnya. Fitur yang sudah selesai dikerjakan adalah fitur *list* menu *Leads* yang bisa ditunjukkan pada gambar 4.46, fitur form tambah data *Leads* pada gambar 4.47 dan fitur edit data *Leads* pada gambar 4.48 :

Gambar 4.46 *Fitur List* Menu *Leads*

Gambar 4.46 merupakan *User Interface* dari menu *list* data *Leads* yang digunakan untuk melihat data *Leads* pelanggan PT. Rabana Sejahtera Indonesia. Di dalam menu *list Leads* mempunyai 3 (tiga) tombol penting yaitu tombol *add* untuk menambah data *Leads*, tombol edit untuk mengedit data *Leads* dan tombol *delete* untuk menghapus data *Leads*. Pengguna menu kelola data *Leads* adalah admin CS PT. Rabana Sejahtera Indonesia.

Gambar 4.47 *Fitur Form Tambah data Leads*

Gambar 4.47 merupakan *User Interface* dari menu tambah data *leads* yang digunakan untuk menambah data *leads* pelanggan ke dalam sistem. Pengguna menu tambah data *leads* adalah admin CS PT. Rabana Sejahtera Indonesia.

Gambar 4.48 *Fitur Form Edit data Leads*

Gambar 4.48 merupakan *User Interface* dari menu edit data *Leads* yang digunakan untuk mengedit data *Leads* pelanggan jika terjadi kesalahan menginput data *Leads*. Pengguna menu *edit* data *Leads* adalah admin CS PT. Rabana Sejahtera Indonesia.

7. *Sprint Retrospective*

Pada *sprint retrospective* ditabel 4.30 berisi *evaluasi* dari *scrum team* dan *product owner* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan sebelumnya.

Tabel 4.30 *Sprint retrospective* fitur menu *Leads*

ID	Story	Evaluasi
US0019	Admin dapat menginput data <i>leads</i>	Sudah sesuai dengan <i>product backlog</i>
US0020	Admin dapat melakukan <i>editing</i> terhadap data <i>leads</i> bila terjadi kesalahan	Sudah sesuai dengan <i>product backlog</i>
US0021	Admin dapat menghapus data <i>leads</i> jika sudah tidak aktif di sistem	Sudah sesuai dengan <i>product backlog</i>
US0022	Admin dapat mencari data <i>leads</i> berdasarkan area pelanggan	Sudah sesuai dengan <i>product backlog</i>

4.8.2 Menu *Opportunities*

Menu *Opportunities* berfungsi untuk mengelola data pelanggan yang akan membeli produk obat herbal yang ada di PT. Rabana Sejahtera Indonesia.

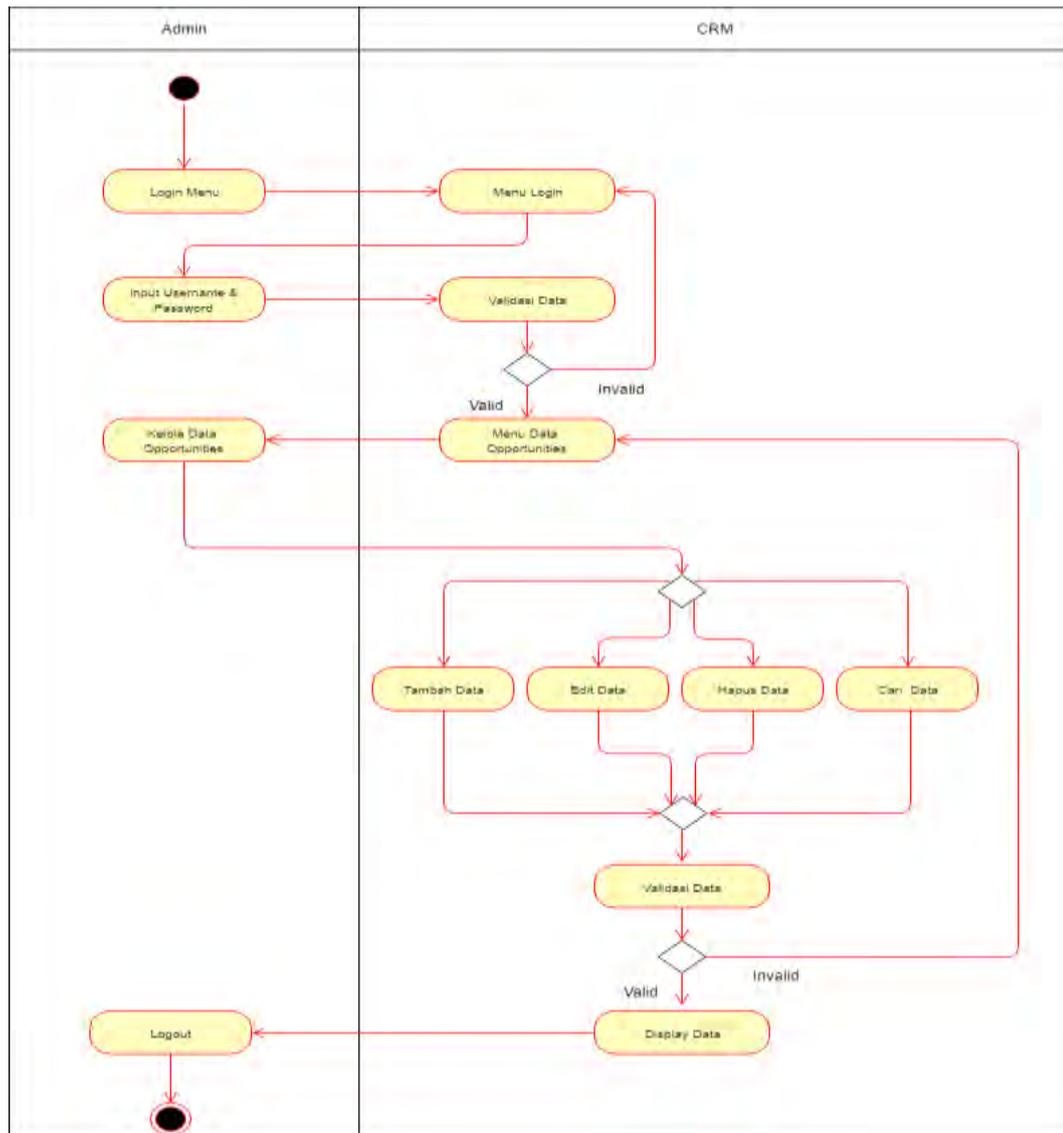
2. *Product Backlog*

Tabel 4.31 *User Story* Menu *Opportunities*

ID	User Story
US0023	Admin dapat menginput data <i>Opportunities</i>
US0024	Admin dapat melakukan <i>editing</i> terhadap data <i>Opportunities</i> bila terjadi kesalahan
US0025	Admin dapat menghapus data <i>Opportunities</i> jika sudah tidak aktif di system
US0026	Admin dapat mencari data <i>Opportunities</i> berdasarkan area pelanggan

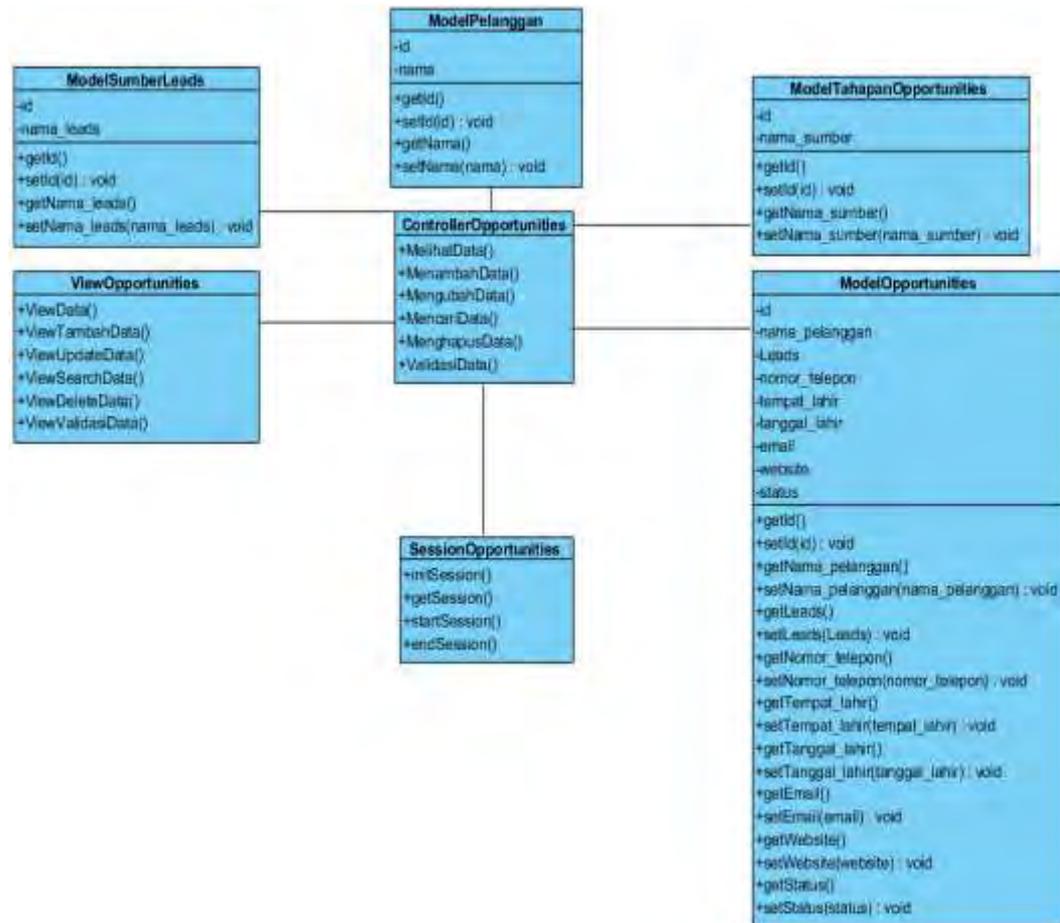
3. *Sprint Planning*

Pada tahap *sprint Planning* merupakan desain sistem informasi *CRM* dari menu *Opportunities* dan menghasilkan *activity* diagram yang dapat di lihat di Gambar 4.49, *class* diagram yang dapat di lihat di Gambar 4.50 dan *component* diagram yang dapat di lihat di Gambar 4.51.



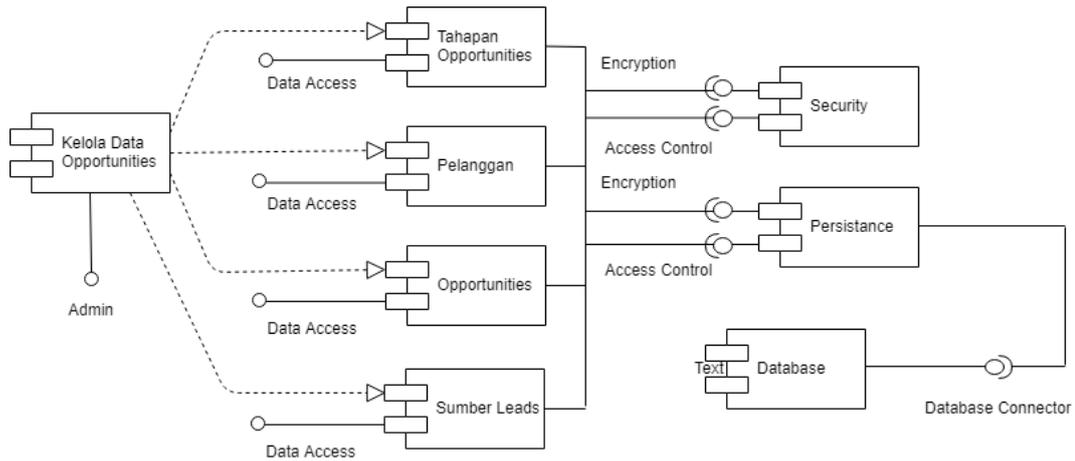
Gambar 4.49 Activity Diagram Menu Opportunities

Gambar 4.49 menunjukkan *activity diagram* pengelolaan menu *opportunities* yang dilakukan oleh Admin CS. Sebelumnya masuk ke dalam sistem. Jika proses validasi berhasil maka sistem akan menampilkan halaman *dashboard* dan admin dapat memilih menu pengelolaan *opportunities*. Ketika admin memilih menu pengelolaan *opportunities* maka admin dapat melakukan pengelolaan *opportunities* dengan menambahkan, merubah dan menghapus *opportunities*. Sistem akan melakukan pengecekan bila terjadi duplikasi data dan akan memberikan notifikasi kepada pengguna sistem.



Gambar 4.50 Class Diagram Menu Opportunities

Gambar 4.50 menunjukkan operasi dari setiap *class* di mana pada *class opportunities* memiliki beberapa *class* yang mempunyai peran masing-masing seperti *class viewOpportunities* berfungsi untuk menampilkan *display opportunities* dan form tambah dan edit *opportunities*, *class controllerOpportunities* berfungsi untuk menjalankan logika dari proses yang di *request* oleh *class view*, *class ModelOpportunities* berfungsi untuk menyimpan entitas-entitas yang berkaitan dengan *menu opportunities* seperti nama, *leads*, nama pelanggan dan status *opportunities*, *class SessionOpportunities* berfungsi untuk menyimpan *session user*.

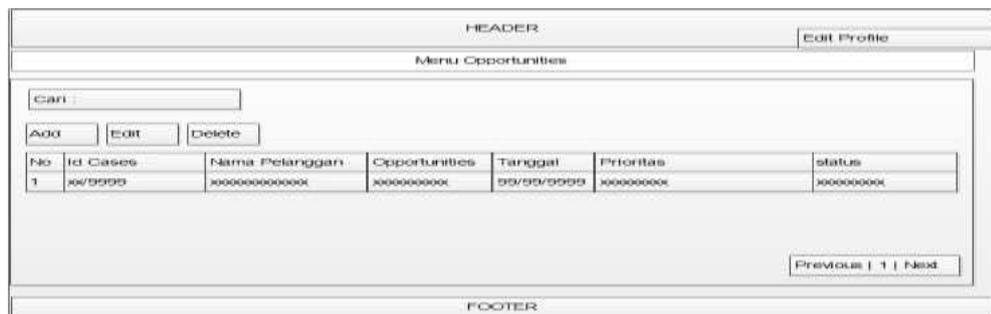


Gambar 4.51 Component Diagram Menu Opportunities

Gambar 4.51 menunjukkan node-node yang ada dalam proses *Opportunities*, di dalam menu *opportunities* terdapat *component opportunities* yang berisikan pesan *opportunities*, *component pelanggan* yang berisi data pelanggan, *component area* yang berisi area, *component tahapan opportunities* yang berisikan status opportunities dan *component sumber leads* yang berisi sumber leads pelanggan tersebut.

4. Sprint Backlog

a. User Interface



Gambar 4.52 User Interface List Menu Opportunities

Gambar 4.52 merupakan rancangan *user interface* dari menu *opportunities* pelanggan. Di rancangan ini ada sebuah *list* yang berisikan id *opportunities*, nama pelanggan, *opportunities*, tanggal, prioritas, status dan beberapa tombol untuk memanipulasi data *opportunities*.

Gambar 4.53 User Interface Form Tambah data Opportunities

Gambar 4.53 merupakan rancangan *user interface* dari menu *opportunities*. Di rancangan ini ada sebuah form yang berisikan id *opportunities* nama pelanggan, kontak, prioritas, area dan status, kolom *header* yang berisikan tulisan menu *Opportunities* dan tombol simpan.

c. Rancangan Basis Data

Tabel 4.32 Tabel Data Menu Opportunities

No	Nama field	Tipe data	Panjang Record	Index
1	Id	Varchar	30	<i>primary key</i>
2	Email	Varchar	30	
3	Nama_ Opportunities	Varchar	100	
4	Prioritas	Varchar	30	
5	Tgl_kasus	Date	15	
6	Id_tahapan_opportunities	Varchar	15	<i>foreign key</i>
7	Id_sumber_leads	Varchar	15	<i>foreign key</i>
8	Id_pelanggan	Varchar	15	<i>foreign key</i>
9	Keterangan	Enum	250	
10	Status	Varchar	10	

5. Sprint Execution

Pada *sprint execution* berisikan beberapa *service* untuk menu *Opportunities*. Tabel 4.28 adalah daftar *service* dari *sprint execution* menu *Opportunities*.

Tabel 4.33 *Sprint execution* menu *Opportunities*

ID	Story	Nama Service	To Do List
US0023	Admin dapat menginput data <i>Opportunities</i>	<i>POST</i> Tambah data <i>Opportunities</i>	1. Membuat <i>table Opportunities</i> 2. Membuat <i>list data Opportunities</i> 3. Membuat form input <i>Opportunities</i>
US0024	Admin dapat melakukan <i>editing</i> terhadap data <i>Opportunities</i> bila terjadi kesalahan	<i>PUT</i> Update data <i>Opportunities</i>	1. Membuat form edit <i>Opportunities</i>
US0025	Admin dapat menghapus data <i>Opportunities</i> jika sudah tidak aktif di system	<i>DELETE</i> data <i>Opportunities</i>	1. Membuat form <i>delete Opportunities</i> yang sudah tidak aktif
US0026	Admin dapat mencari data <i>Opportunities</i> berdasarkan area pelanggan	<i>GET</i> Cari data by area	1. Membuat form pencarian berdasarkan area pelanggan

8. *Sprint Review*

Pada *sprint review* ditabel 4.29 berisi *review – review* dari *product owner* dan *stakeholders* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan oleh *product owner* sebelumnya.

Tabel 4.34 *Sprint review* menu *Opportunities*

ID	Story	Review
US0023	Admin dapat menginput data <i>Opportunities</i>	Data <i>Opportunities</i> berhasil di inputkan kedalam sistem
US0024	Admin dapat melakukan <i>editing</i> terhadap data <i>Opportunities</i> bila terjadi kesalahan	Data <i>Opportunities</i> berhasil di edit
US0025	Admin dapat menghapus data <i>Opportunities</i> jika sudah tidak aktif di system	Data <i>Opportunities</i> berhasil di hapus
US0026	Admin dapat mencari data <i>Opportunities</i> berdasarkan area pelanggan	Pencarian data <i>Opportunities</i> dapat menemukan data sesuai dengan area yang dipilih

9. *Increment*

Pada tahap *increment* adalah pendistribusian fitur yang sudah dikerjakan oleh *scrum team* kepada pengguna dan *stakeholder*. Jika ada fitur yang masih kurang sesuai dengan kebutuhan pengguna maka fitur dapat dimasukkan kembali pada *sprint* berikutnya. Fitur

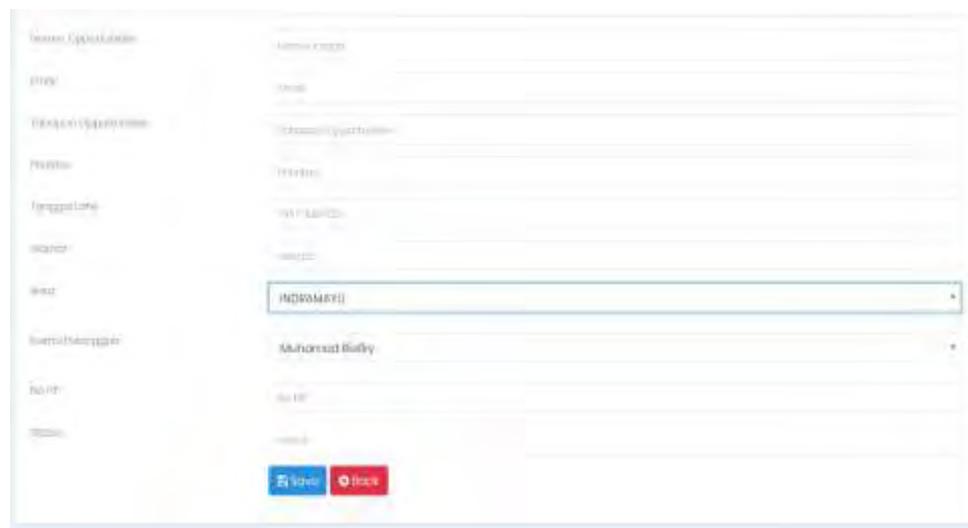
yang sudah selesai dikerjakan adalah fitur *list* menu *Opportunities* yang bisa ditunjukkan pada gambar 4.46, fitur form tambah data *Opportunities* pada gambar 4.47 dan fitur edit data *Opportunities* pada gambar 4.48:



No	Nomor Opportunities	Tahapan Opportunities	Nama Pelanggan	Prioritas	Nomor Handphone	Status
1	00001	risong	Muhammad Bilal	Tinggi	0887900664	Pending
2	00002	0001123456789	Ida	Rendah	08123456789	Selesai

Gambar 4.54 Fitur List Menu *Opportunities*

Gambar 4.54 merupakan *User Interface* dari menu *list* data *opportunities* yang digunakan untuk melihat data *opportunities* pelanggan PT. Rabana Sejahtera Indonesia. Di dalam menu *list opportunities* mempunyai 3 (tiga) tombol penting yaitu tombol *add* untuk menambah data *opportunities*, tombol *edit* untuk mengedit data *opportunities* dan tombol *delete* untuk menghapus data *opportunities*. Pengguna menu kelola data *opportunities* adalah admin CS PT. Rabana Sejahtera Indonesia.



Gambar 4.55 Fitur Form Tambah data *Opportunities*

Gambar 4.55 merupakan *User Interface* dari menu tambah data *opportunities* yang digunakan untuk menambah data *opportunities* pelanggan kedalam sistem. Pengguna menu tambah data *opportunities* adalah admin CS PT. Rabana Sejahtera Indonesia.

Gambar 4.56 Fitur Form Edit data *Opportunities*

Gambar 4.56 merupakan *User Interface* dari menu edit data *opportunities* yang digunakan untuk mengedit data *opportunities* pelanggan jika terjadi kesalahan menginput data *opportunities*. Pengguna menu *edit data opportunities* adalah admin CS PT. Rabana Sejahtera Indonesia.

10. *Sprint Retrospective*

Pada *sprint retrospective* ditabel 4.35 berisi *evaluasi* dari *scrum team* dan *product owner* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan sebelumnya.

Tabel 4.35 *Sprint retrospective* fitur menu *Opportunities*

ID	Story	Evaluasi
US0023	Admin dapat menginput data <i>Opportunities</i>	Sudah sesuai dengan <i>product backlog</i>
US0024	Admin dapat melakukan <i>editing</i> terhadap data <i>Opportunities</i> bila terjadi kesalahan	Sudah sesuai dengan <i>product backlog</i>
US0025	Admin dapat menghapus data <i>Opportunities</i> jika sudah tidak aktif di system	Sudah sesuai dengan <i>product backlog</i>
US0026	Admin dapat mencari data <i>Opportunities</i> berdasarkan area pelanggan	Sudah sesuai dengan <i>product backlog</i>

4.9 Sprint 5 Menu *Email Reminder* dan Menu *Chatting*

4.9.1 Menu *Email Reminder*

Menu *email reminder* berfungsi untuk mengirim *email* notifikasi kepada pelanggan di PT. Rabana Sejahtera Indonesia untuk memudahkan pelanggan dalam mendapatkan informasi terbaru dari PT. Rabana Sejahtera Indonesia.

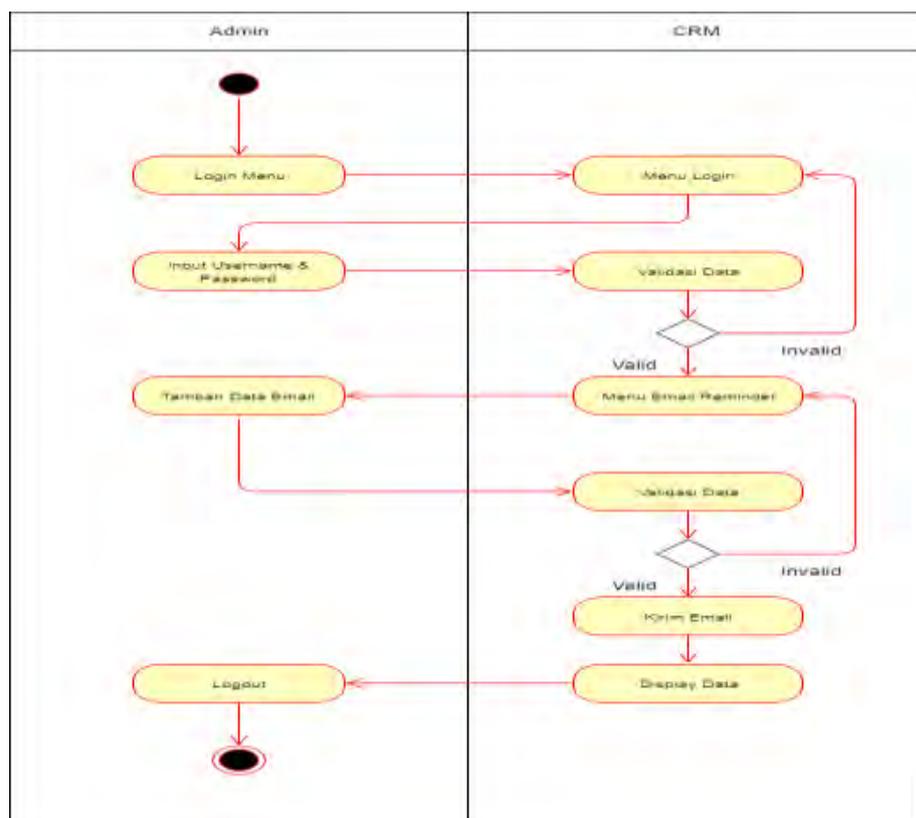
1. *Product Backlog*

Tabel 4.36 *User Story* menu *email reminder*

ID	User Story
US0027	Admin dapat mengirim email kepada pelanggan

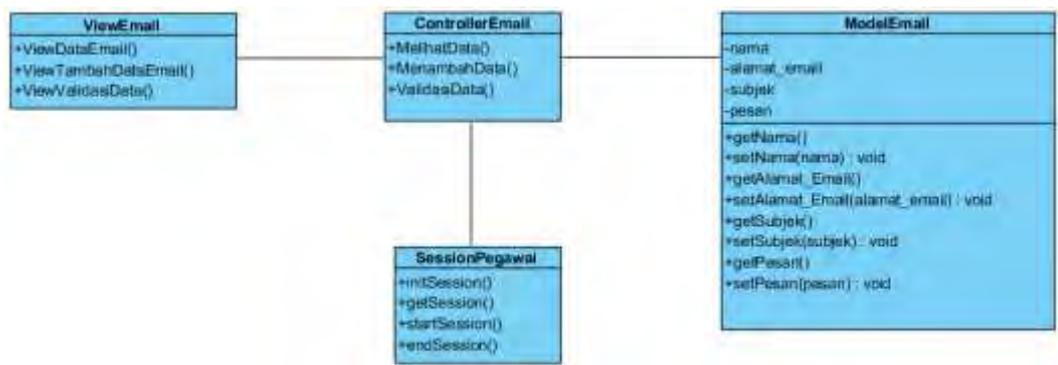
2. *Sprint Planning*

Pada tahap *sprint Planning* merupakan desain sistem informasi CRM dari modul *email reminder* dan menghasilkan *activity diagram* yang dapat di lihat di Gambar 4.57, *class diagram* yang dapat di lihat di Gambar 4.58 dan *component diagram* yang dapat di lihat di Gambar 4.59.



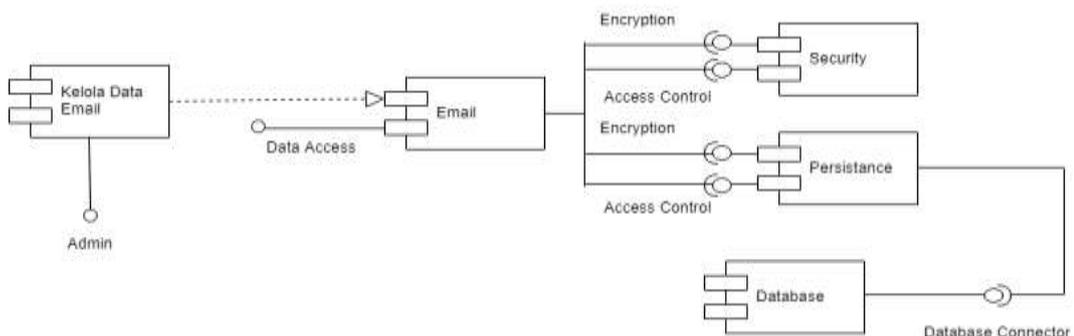
Gambar 4.57 *Activity Diagram* Menu *Email Reminder*

Gambar 4.57 menunjukkan *activity diagram* pengelolaan menu *email* yang dilakukan oleh Admin CS. Sebelumnya masuk ke dalam sistem. Jika proses validasi berhasil maka sistem akan menampilkan halaman *dashboard* dan admin dapat memilih menu pengelolaan *email*. Ketika admin memilih menu pengelolaan *email* maka admin dapat melakukan pengelolaan *email* dengan menambahkan, merubah dan menghapus *email*. Sistem akan melakukan pengecekan bila terjadi duplikasi data dan akan memberikan notifikasi kepada pengguna sistem.



Gambar 4.58 Class Diagram Menu Email Reminder

Gambar 4.58 menunjukkan operasi dari setiap *class* di mana pada *class email reminder* memiliki beberapa *class* yang mempunyai peran masing-masing seperti *class viewEmail* berfungsi untuk menampilkan *display email* dan form tambah *email*, *class controllerEmail* berfungsi untuk menjalankan logika dari proses yang di *request* oleh *class view*, *class ModelEmail* berfungsi untuk menyimpan entitas-entitas yang berkaitan dengan *menu email* seperti nama, alamat email, subjek dan isi pesan, *class SessionPegawai* berfungsi untuk menyimpan *session user*.



Gambar 4.59 Component Diagram Menu Email Reminder

Gambar 4.59 menunjukkan node-node yang ada dalam proses *email*, di dalam menu *email* terdapat *component* email yang berisikan pesan *email* dari admin yang akan dikirim kepada pelanggan PT. Rabana Sejahtera Indonesia.

3. *Sprint Backlog*

a. *User Interface*

Gambar 4.60 *User Interface Menu Email*

Gambar 4.60 merupakan rancangan *user interface* dari menu *email*. Di rancangan ini sebuah form yang berisikan nama, email *subject* dan pesan, kolom *header* yang berisikan tulisan menu *email* dan tombol submit.

4. *Sprint Execution*

Pada *sprint execution* berisikan *service* untuk mengirim pesan. Tabel 4.37 adalah *service* dari *sprint execution* menu *email reminder*.

Tabel 4.37 *Sprint execution menu email reminder*

ID	Story	Nama Service	To Do List
US0027	Admin dapat mengirim email kepada pelanggan	POST Kirim <i>Email Reminder</i>	<ol style="list-style-type: none"> 1. Membuat form kirim email 2. Integrasi dengan library email dari code igniter

5. *Sprint Review*

Pada *sprint review* ditabel 4.38 berisi *review – review* dari *product owner* dan *stakeholders* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan oleh *product owner* sebelumnya.

Tabel 4.38 *Sprint review* menu *email reminder*

ID	Story	Review
US0027	Admin dapat mengirim <i>email</i> kepada pelanggan	<i>Email</i> yang dikirim admin terkirim ke <i>email</i> pelanggan

6. *Increment*

Pada tahap *increment* adalah pendistribusian fitur yang sudah dikerjakan oleh *scrum team* kepada pengguna dan *stakeholder*. Fitur yang sudah selesai dikerjakan adalah fitur *email reminder* yang bisa ditunjukkan pada gambar 4.62 :

CRM PT. Rabana Sejahtera Indonesia

Form Email

Alamat Email
Masukkan Alamat Email

Subjek
Masukkan Subjek

Keterangan
Keterangan

Submit

Gambar 4.61 *Menu Email Reminder*

Gambar 4.61 merupakan *User Interface* dari menu *email reminder* yang digunakan untuk melakukan kirim *email* kepada pelanggan. Pengguna menu *email reminder* adalah admin CS PT. Rabana Sejahtera Indonesia.

7. *Sprint Retrospective*

Pada *sprint retrospective* ditabel 4.39 berisi *evaluasi* dari *scrum team* dan *product owner* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan sebelumnya, *sprint retrospective* pada *sprint* ini berjalan dengan baik dan *scrum team* saling terbuka terhadap tanggung jawabnya..

Tabel 4.39 *Sprint retrospective* menu *email reminder*

ID	Story	Evaluasi
US0027	Admin dapat mengirim email kepada pelanggan	Sudah sesuai dengan <i>product backlog</i>

4.9.2 Menu *Chatting*

Menu *chatting* berfungsi untuk admin melakukan *chatting* dengan pelanggan di PT. Rabana Sejahtera Indonesia yang dapat berkomunikasi secara *realtime*..

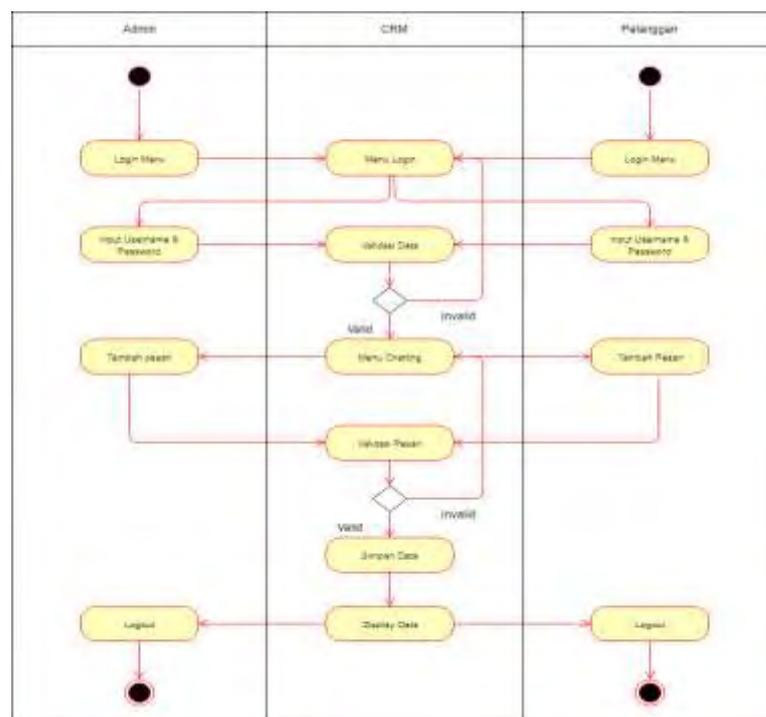
1. *Product Backlog*

Tabel 4.40 *User Story* Menu *chatting*

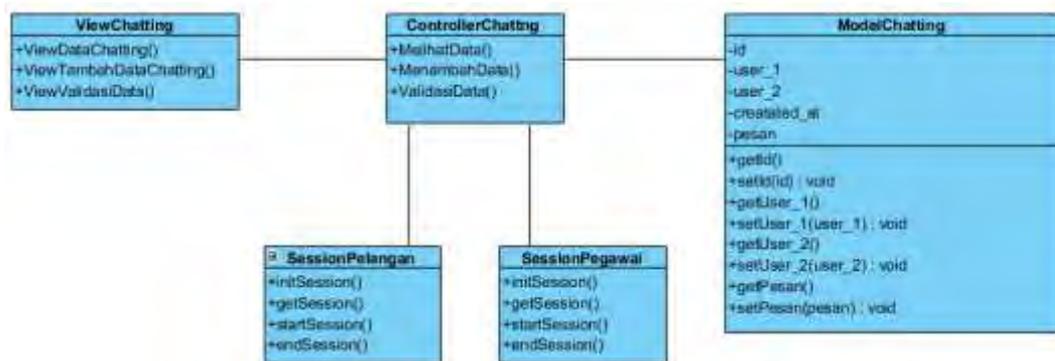
ID	User Story
US0028	Pelanggan dapat melakukan <i>chatting</i> dengan admin cs
US0029	Admin cs bisa membalas <i>chatting</i> dari pelanggan

2. *Sprint Planning*

Pada tahap *sprint Planning* merupakan desain sistem informasi CRM dari menu *chatting* dan menghasilkan *activity* diagram yang dapat di lihat di Gambar 4.62, *class* diagram yang dapat di lihat di Gambar 4.63 dan *component* diagram yang dapat di lihat di Gambar 4.64.

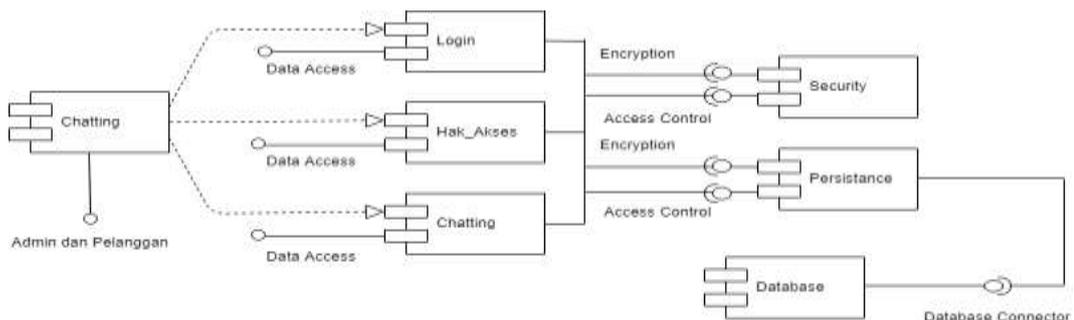
Gambar 4.62 *Activity Diagram* Menu *chatting*

Gambar 4.62 menunjukkan *activity diagram* pengelolaan menu *chatting* yang dilakukan oleh Admin CS dan pelanggan. Sebelumnya masuk ke dalam sistem. Jika proses validasi berhasil maka sistem akan menampilkan halaman *dashboard* dan admin dapat memilih menu pengelolaan *chatting*. Ketika admin memilih menu pengelolaan *chatting* maka admin dapat melakukan pengelolaan *chatting* dengan menambahkan, merubah dan menghapus *chatting*. Sistem akan melakukan pengecekan bila terjadi duplikasi data dan akan memberikan notifikasi kepada pengguna sistem.



Gambar 4.63 Class Diagram Menu *chatting*

Gambar 4.63 menunjukkan operasi dari setiap class di mana pada class *chatting* memiliki beberapa class yang mempunyai peran masing-masing seperti class *viewChatting* berfungsi untuk menampilkan *display chatting* dan form tambah *chatting*, class *controllerChatting* berfungsi untuk menjalankan logika dari proses yang di *request* dari class *view*, class *ModelChatting* berfungsi untuk menyimpan entitas-entitas yang berkaitan dengan menu *chatting* seperti *id*, *user pertama*, *user kedua*, tanggal diproses dan isi pesan, class *SessionPelanggan* dan *SessionPegawai* berfungsi untuk menyimpan *session user*.

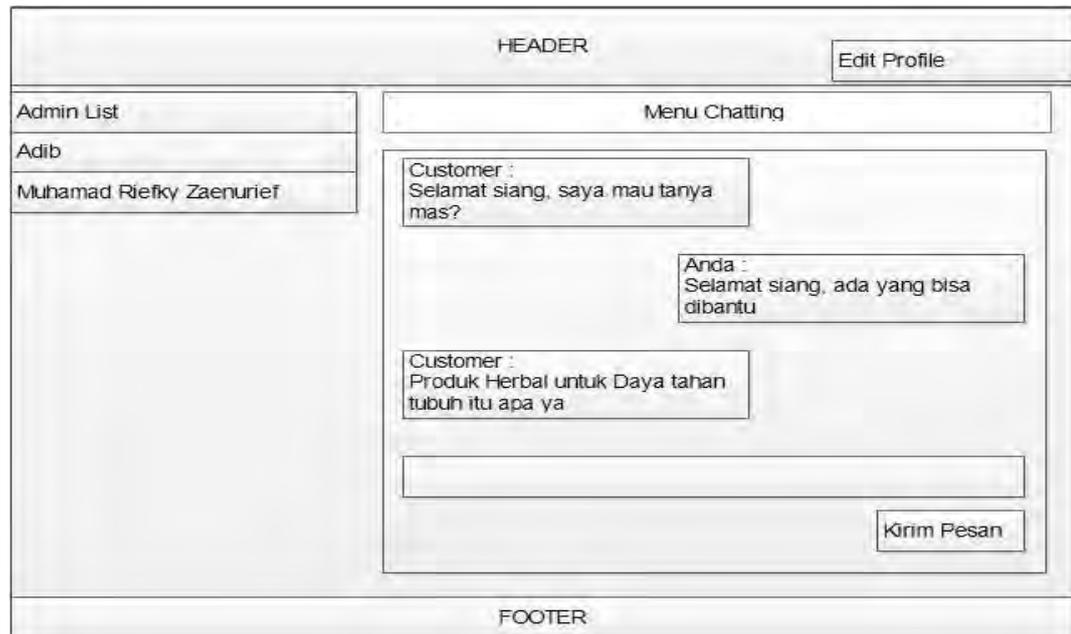


Gambar 4.64 Component Diagram Menu *chatting*

Gambar 4.64 menunjukkan node-node yang ada dalam proses *chatting*, di dalam menu *chatting* terdapat *component* login, *component* hak akses yaitu admin dan pelanggan dan *component* *chatting* yang berisikan pesan *chatting* dari admin dan pelanggan.

3. Sprint Backlog

a. User Interface



Gambar 4.65 User Interface Menu Chatting

Gambar 4.65 merupakan rancangan *user interface* dari menu *chatting*. Di rancangan ini terdapat admin list yang berisikan nama-nama admin CS, kolom *header* yang berisikan tulisan menu *chatting*, kotak dialog untuk menuliskan pesan kepada admin dan tombol kirim pesan

b. Rancangan Basis Data

Tabel 4.41 Tabel Data Menu Chatting

No	Nama field	Tipe data	Panjang Record	Index
1	id_chat	Varchar	50	primary key
2	User_1	Varchar	50	foreign key
3	User_2	Varchar	50	foreign key
4	Created_at	Timestamp	15	
5	Pesan	Text	0	

4. *Sprint Execution*

Pada *sprint execution* berisikan beberapa *service* untuk Menu *chatting*. Tabel 4.42 adalah daftar *service* dari *sprint execution* menu *chatting*

Tabel 4.42 *Sprint execution* menu *chatting*

ID	Story	Nama Service	To Do List
US0028	Pelanggan dapat melakukan <i>chatting</i> dengan admin cs	<i>POST chatting</i>	<ol style="list-style-type: none"> 1. Membuat <i>table chatting</i> 2. Membuat <i>list</i> data admin 3. Membuat form input pesan di sisi pelanggan
US0029	Admin cs bisa membalas <i>chatting</i> dari pelanggan	<i>POST chatting</i>	<ol style="list-style-type: none"> 1. Membuat <i>table chatting</i> 2. Membuat <i>list</i> data admin 3. Membuat form input pesan di sisi admin

5. *Sprint Review*

Pada *sprint review* ditabel 4.43 berisi *review – review* dari *product owner* dan *stakeholders* tentang kinerja *scrum team* untuk menu *chatting*.

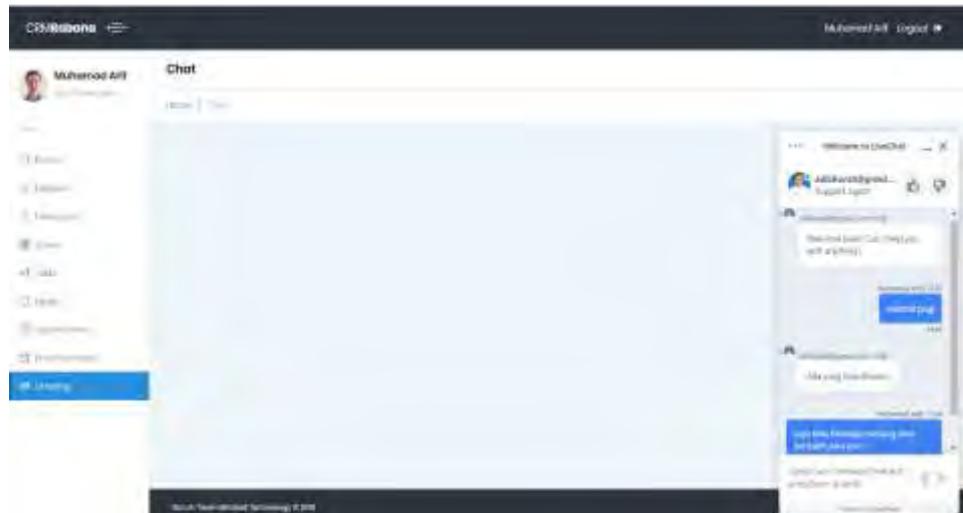
Tabel 4.43 *Sprint review* fitur *chatting*

ID	Story	Review
US0028	Pelanggan dapat melakukan <i>chatting</i> dengan admin cs	Pesan sampai kepada admin
US0029	Admin cs bisa membalas <i>chatting</i> dari pelanggan	Admin dapat membalas <i>chatting</i> dari pelanggan

6. *Increment*

Pada tahap *increment* adalah pendistribusian fitur yang sudah dikerjakan oleh *scrum team* kepada pengguna dan *stakeholder*. Jika ada fitur yang masih kurang sesuai dengan kebutuhan pengguna maka fitur dapat dimasukkan kembali pada *sprint* berikutnya. Fitur yang sudah selesai dikerjakan adalah fitur *chatting* yang bisa ditunjukkan pada gambar

4.67



Gambar 4.66 Menu *Chatting*

Gambar 4.66 merupakan *User Interface* dari menu *chatting* yang digunakan untuk melakukan *chatting* secara realtime. Pengguna menu *chatting* adalah CS dan pelanggan PT. Rabana Sejahtera Indonesia.

7. *Sprint Retrospective*

Pada *sprint retrospective* ditabel 4.44 berisi *evaluasi* dari *scrum team* dan *product owner* tentang kinerja *scrum team*, apakah sudah sesuai dengan *product backlog* yang telah ditentukan sebelumnya.

Tabel 4.44 *Sprint retrospective* fitur *chatting*

ID	Story	Evaluasi
US0028	Pelanggan dapat melakukan <i>chatting</i> dengan admin cs	Sudah sesuai dengan <i>product backlog</i>
US0029	Admin cs bisa membalas <i>chatting</i> dari pelanggan	Sudah sesuai dengan <i>product backlog</i>

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan rumusan masalah yang ada pada penelitian sistem informasi *CRM* pada PT. Rabana Sejahtera Indonesia maka ditemukan beberapa kesimpulan yaitu :

1. Tahapan proses analisis sistem informasi *CRM* menggunakan metode *Scrum* menghasilkan 5 (lima) *sprint* selama penelitian berlangsung. Aktor yang berperan dalam Metode *Scrum* adalah *Product Owner* yang mempunyai kepentingan untuk menggunakan sistem jika sudah tercipta dan memberikan spesifikasi sistem informasi *CRM*, *Scrum Master* bertugas untuk menjembatani antara *product owner* dan *Development team* sedangkan *development team* mempunyai peran untuk membangun sistem Informasi *CRM* PT. Rabana Sejahtera Indonesia.
2. Tahapan proses perancangan sistem informasi *CRM* setiap *sprint* berdurasi 2 (dua) minggu dan mempunyai *output* di setiap fase, pada fase *Product Backlog* menghasilkan *user story* dari *product owner*, kemudian pada fase *sprint planning* menghasilkan *activity diagram*, *class diagram* dan *component diagram*, pada fase *sprint backlog* menghasilkan *user interface* dan rancangan *database*, pada fase *sprint execution* menghasilkan fitur-fitur sistem informasi *CRM*, pada fase *sprint review* menghasilkan evaluasi tentang sistem informasi *CRM*, pada fase *increment* menghasilkan fitur sistem informasi *CRM* yang dapat digunakan dan pada fase *sprint retrospective* menghasilkan evaluasi yang lebih spesifik mengenai fitur sistem informasi *CRM* atau *development team*. *Service* yang dihasilkan dari penelitian adalah *service* pengelola data pelanggan, *service* pengelola data pegawai, *service* pengelola data *task* pegawai, *service* pengelola data *cases* pelanggan, *service* pengelola data *leads* pelanggan yang potensial, *service* pengelola data *opportunities* pelanggan, *service email reminder* untuk mengirim *email* secara berkala kepada pelanggan dan *service chatting* dengan pelanggan.

5.2 Saran

Adapun beberapa saran yang dapat diusulkan untuk membantu pengembangan dan penelitian sistem informasi *CRM* pada PT. Rabana Sejahtera Indonesia yaitu :

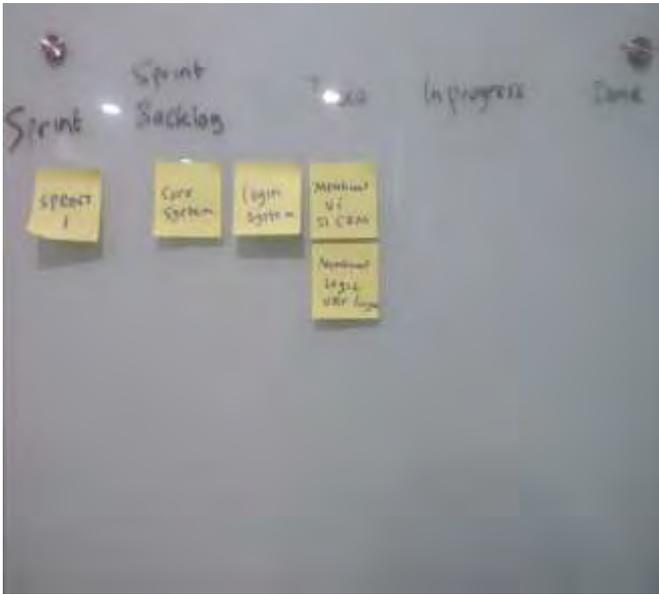
1. Pembuatan *mobile application* yang memiliki integrasi dengan sistem informasi *CRM* karena saat ini penggunaan perangkat mobile sangat tinggi dan mendukung penerapan teknologi yang semakin maksimal. Contoh teknologi yang dapat diterapkan adalah teknologi *GPS (Global Positioning System)* dan notifikasi terbaru secara *realtime*.
2. Perlunya diadakan penelitian lebih lanjut mengenai tingkat kesuksesan penerapan Sistem Informasi *CRM* pada suatu objek penelitian.

DAFTAR PUSTAKA

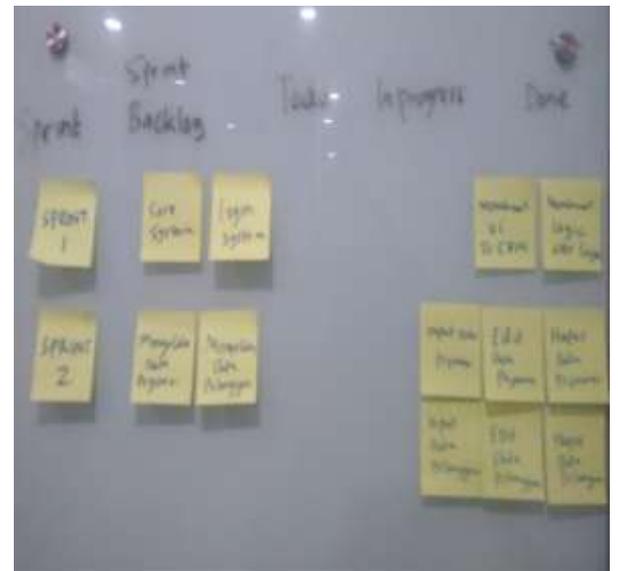
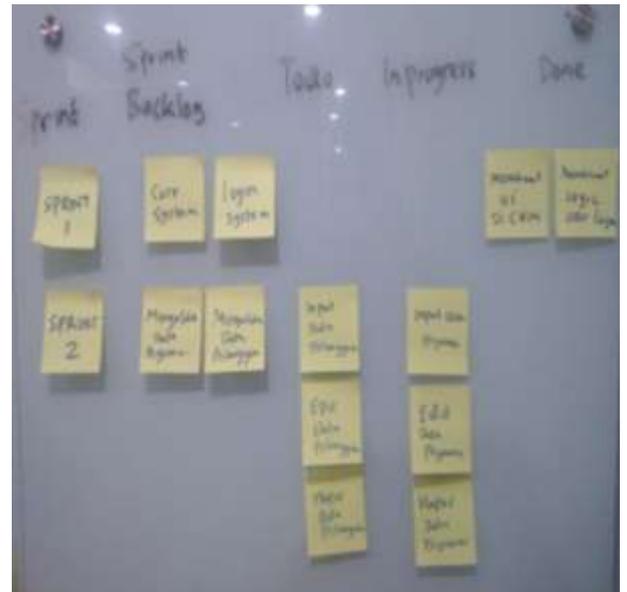
- Abdulmyanov, V., Sivtsov, A, & Fomichyov, N., 2017. *Architecting a Geo-Enabled CRM: The Way to Seamless Integration*. International Conference on Knowledge Based and Intelligent Information and Engineering System s, KES2017, 6-7 September 2017, Marseille, France.
- Adi, P. 2015. "Scrum Method Implementation in a Software Development Project Management". International Journal of Advanced Computer Science and Applications, 6(9), 198–204. <https://doi.org/10.14569/IJACSA.2015.060927>
- Aditya dan Alan, Nur. 2011. *Jago PHP dan MySQL*. Bekasi: DUNIA KOMPUTER.
- Arief, M. 2011. *Pemograman Web Dinamis menggunakan PHP dan MySQL*. Yogyakarta : Andi Offset.
- Buttle, Francis. 2015. "Customer Relationship Management Consept and Tools Third Edition". Penerbit Bayu Media Publishing, Malang.
- Cook, Sarah. 2004. "Customer Care Excellence". Jakarta: PPM
- Erdaka, A., 2015. *Customer Loyalty in Collaborative Consumption Model: Empirical Study of CRM for Product-Service Sistem -Based e-Commerce in Indonesia*. The Third Information Sistem s International Conference
- Karsalari, A. R, 2017. *The Impact of Customer Relationship Management on Customer Loyalty in LG Company (Goldiran) Tehran City*.
- Irawan, Handi. 2002. "10 Prinsip Kepuasan Pelanggan". Jakarta: Elex Media Komputindo.
- Jogiyanto, HM, 2011, "Analisis dan Desain Sistem Informasi : pendekatan terstruktur teori dan praktik aplikasi bisnis". Yogyakarta : Penerbit Andi..
- Khairuzzaman, M, Q., 2016. Analisis Dan Perancangan Sistem Informasi Electronic Customer Relationship Management (E-CRM) Pt. Delta Jaya Motor Singkawang. *Jurnal Khatulistiwa Informatika*, 4(1 Juni 2016).
- Kristanto, Andri. 2008. "Perancangan Sistem Informasi". Yogyakarta : Gava Media.
- Laudon, Kenneth C dan Traver, Carol Guercio., 2002, "E-Commerce; business technology society", International Edition. Addison Wesley, USA
- Majid, Abdul Suharto. 2009. "Customer Service dalam Bisnis Jasa Transportasi". Jakarta : Rajawali Pers.
- Munawar. 2018. "Analisis Perancangan Sistem Berorientasi Objek dengan Unified Modeling Language", Penerbit Informatika, Bandung.
- Nasution, Muhamad Nur. 2010, "Manajemen Perubahan". Ghalia Indonesia. Bogor.
- Oetomo, B. S. D., John P. S., dan Andreas A. S., 2003, "I-CRM Membina Relasi dengan Pelanggan.com", Yogyakarta : Penerbit Andi.
- Partogi, Joshua ., 2015, "Manajemen Modern dengan Scrum", Yogyakarta : Penerbit Andi.
- Pressman, Roger, S, 2012, "Rekayasa Perangkat Lunak: Pendekatan Praktisi (Edisi tujuh)". Yogyakarta : Penerbit Andi..

- Raharjo, Budi. 2015. *Belajar Otodidak MySQL (teknik pembuatan dan pengolahan database)*. Bandung : Penerbit Informatika..
- Rusydi, Mhd. 2017. *Belajar Customer excellence* . Yogyakarta : Gosyn Publishing..
- Salahuddin, Muhamad, 2011, "*Modul Pembelajaran Rekayasa Perangkat Lunak Terstruktur dab Berorientasi Objek*". Bandung : Modula..
- Sarosa, Samiaji, 2017, "*Metodologi Pengembangan Sistem Informasi*". Jakarta : Penerbit Indeks.
- Sutabri, Tata, 2019, "*Sistem Informasi Bisnis*". Yogyakarta : Penerbit Andi.
- Sutherland, Jeff. 2017, "*Scrum, Meningkatkan Produktivitas Dua Kali Lipat dalam Waktu Setengahnya Saja*". Yogyakarta : BENTANG..
- Tohari, Tamim, 2014, "*Sistem Informasi melalui pendekatan UML*", Yogyakarta : CV. Andi Offset.

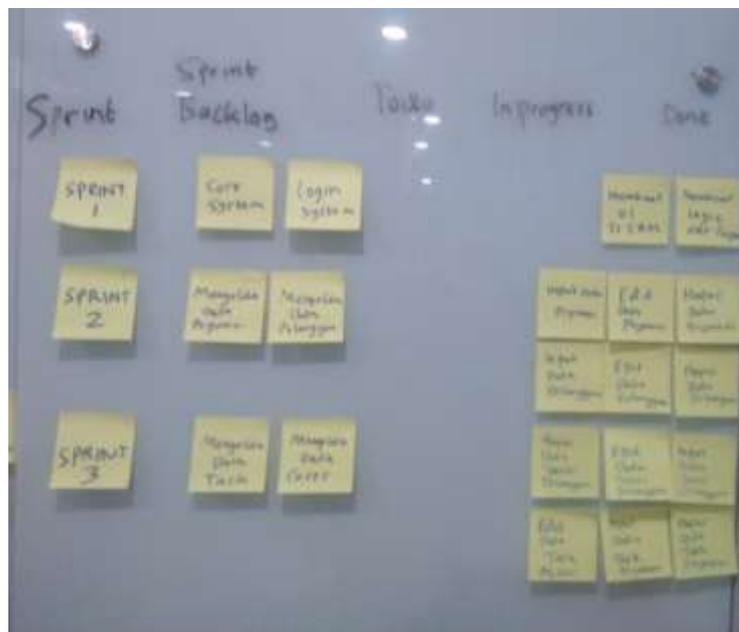
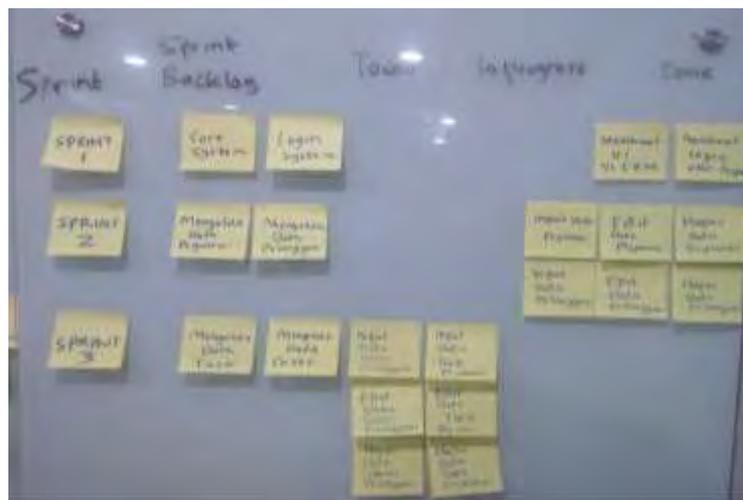
LAMPIRAN



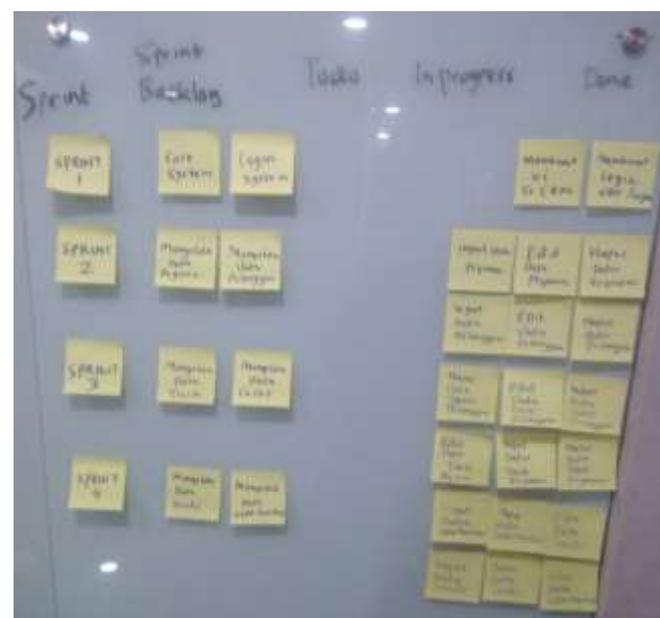
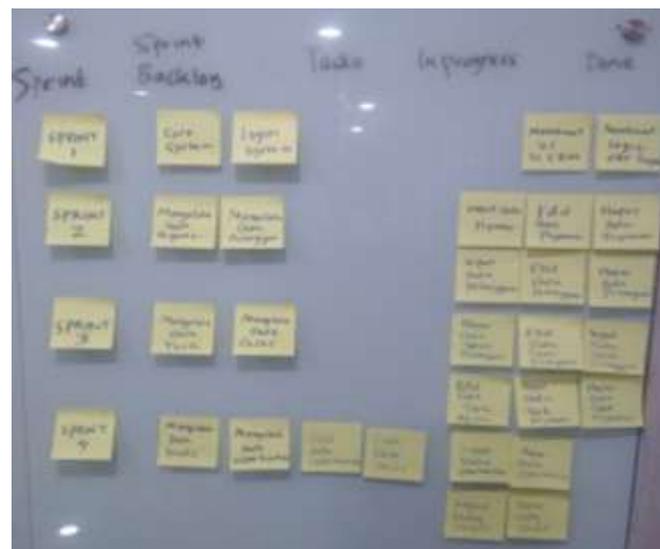
Doc Pribadi Foto Scrum Board Sprint 1



Doc Pribadi Foto Scrum Board Sprint 2



Doc Pribadi Foto Scrum Board Sprint 3



Doc Pribadi Foto Scrum Board Sprint 4



Doc Pribadi Foto Scrum Board Sprint 5



Doc Pribadi Foto *Sprint Planning*



Doc Pribadi Foto *Daily Scrum*



Doc Pribadi Foto *Sprint Review*



Doc Pribadi Foto *Sprint Retrospective*