

**PERANCANGAN MODUL USER APLIKASI PENDONORAN  
DARAH BERBASIS WEB DENGAN MENGGUNAKAN  
METODE SCRUM PADA STARTUP E-VAMPIRE**

**Oleh**

**FADLI ANUGRAH EMAS**

**NIM : 1106130151**



**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS REKAYASA INDUSTRI  
UNIVERSITAS TELKOM  
BANDUNG  
2017**

**LEMBAR PENGESAHAN**

Tugas Akhir dengan judul:

**PERANCANGAN MODUL USER APLIKASI PENDONORAN  
DARAH BERBASIS WEB DENGAN MENGGUNAKAN  
METODE SCRUM PADA STARTUP E-VAMPIRE**

Telah disetujui dan disahkan pada Sidang Tugas Akhir  
Program Studi Strata 1 Sistem Informasi  
Fakultas Rekayasa Industri Universitas Telkom

**Oleh :**

**Fadli Anugrah Emas**

**1106130151**

Bandung, 11 Juli 2017

Disetujui oleh,

Pembimbing 1,

Pembimbing 2,

Soni Fajar Surya Gumilang, S.T., M.T

NIP: 17750069-2

Muhammad Azani Hasibuan, S.Kom., M.T.I

NIP: 14850055-1

## LEMBAR PERNYATAAN ORISINALITAS



Nama : Fadli Anugrah Emas

Nim : 1106130151

Alamat : Jl. Sukabirus, Kos Omega Dayeuhkolot,  
Kab. Bandung

No.Telp : 082126981085

Email : fadli.anugrah98@gmail.com

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri. Atas pernyataan ini, saya siap menanggung risiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.

Bandung, 11 Juli 2017

Fadli Anugrah Emas

**ABSTRAK**  
**PERANCANGAN MODUL USER APLIKASI**  
**PENDONORAN DARAH BERBASIS WEB DENGAN**  
**MENGGUNAKAN METODE SCRUM PADA STARTUP E-**  
**VAMPIRE**

Oleh

**FADLI ANUGRAH EMAS**

**NIM : 1106130151**

Ketersediaan darah untuk donor secara ideal adalah 2% dari jumlah penduduk yang disesuaikan dengan standar lembaga kesehatan internasional (WHO). Sehingga jika jumlah penduduk di Indonesia sebesar 258.705.000 jiwa (Badan Pusat Statistik , 2017) , maka idealnya dibutuhkan darah sebanyak 5.174.100 kantong darah. Akan tetapi pada tahun 2016 lalu, jumlah darah yang terkumpul dari pendonor sebanyak 4.600.000 kantong darah. Sehingga secara nasional terdapat kekurangan kebutuhan darah sejumlah 574.100 kantong darah. Dari hasil survei yang telah kami lakukan, terdapat beberapa alasan masyarakat takut atau tidak mau melakukan donor darah. Beberapa di antaranya berupa takut tertular penyakit, takut sakit disuntik, tubuh menjadi lemas, trauma dan takut gemuk maka dapat disimpulkan penyebab utama masyarakat tidak mendonorkan darahnya yaitu kurangnya edukasi masyarakat mengenai donor darah yang mengakibatkan kurangnya *awareness* (kesadaran) masyarakat untuk melakukan donor darah. Muncul sebuah ide untuk menutupi kekurangan dari aplikasi – aplikasi yang telah ada dengan membuat aplikasi yang baru dinamakan E-Vampire. Untuk menunjang peningkatan pemahaman mengenai donor darah, salah satu aplikasi E-Vampire akan diluncurkan berbasis aplikasi *website*. Perancangan aplikasi E-Vampire berbasis *website* akan menggunakan salah satu dari *software development life cycle agile* yaitu *scrum*. *Scrum* merupakan salah satu varian dari metode *agile* yang *iterative* dan *incremental*. Pengembangan aplikasi E-Vampire akan dibawa menuju tingkat selanjutnya dalam dunia bisnis menjadi sebuah perusahaan *startup* (Scrum Alliance , 2016). *Startup* E-Vampire mengadopsi salah satu model bisnis *Lean Canvas* yang berguna untuk merancang kebutuhan bisnis yang diperlukan oleh *startup* E-Vampire untuk beberapa tahun mendatang. Dapat diambil kesimpulan bahwa aplikasi E-Vampire berbasis *website* dirancang untuk menambah kesadaran masyarakat tentang pentingnya untuk donor darah.

Kata kunci : *scrum, website, startup*, donor darah

**ABSTRACT**  
**DESIGN OF APPLICATION USER MODULE BLOOD DONATION**  
**BASED WEB APPLICATION USING SCRUM METHOD ON**  
**STARTUP E-VAMPIRE**

**By**  
**FADLI ANUGRAH EMAS**  
**NIM: 1106130151**

The ideal blood supply for donors is 2% of the total population adjusted to international health standards (WHO) standards. So if the population in Indonesia amounted to 258,705,000 people (Badan Pusat Statistik , 2017), then ideally needed as many blood 5,174,100 blood bags. However, in 2016 and then, the amount of blood collected from donors as much as 4.600.000 blood bags. So nationally there is a lack of blood needs a total of 574,100 blood bags. From the results of the surveys we have done, there are several reasons people are afraid or unwilling to do blood donation. Some of them are the fear of contracting the disease, the fear of injected, the body becomes weak, trauma and fear of fat, it can be concluded that the main cause of the community is not donating their blood is the lack of public education about blood donation resulting in the lack of awareness (public awareness) to do blood donation. Appears an idea to cover the shortcomings of existing applications by creating a new application called E-Vampire. To support the increased understanding of blood donation, one of the E-Vampire applications will be launched based on website applications. E-Vampire website based design application will use one of life cycle development software agile that is scrum. Scrum is one of the variants of iterative and incremental agile methods. The development of E-Vampire applications will be brought to the next level in the business world into a startup company (Scrum Alliance, 2016). The E-Vampire Startup adopts one of Lean Canvas's business models that is useful for designing the business needs required by E-Vampire startup for the next few years. It can be concluded that a website-based E-Vampire application is designed to increase public awareness of the importance of blood donation.

*Keyword : scrum, websiter, startup, blood donation*

## **LEMBAR PERSEMBAHAN**

Tugas akhir ini penulis persembahkan kepada kedua orang tua dan saudara yang selalu memberikan semangat, dukungan dan doa yang membuat penulis menjadi seperti ini.

## **KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Allah SWT karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan tugas akhir ini. Shalawat dan salam semoga selalu tercurah kepada Rasulullah SAW beserta keluarganya. Tugas akhir ini disusun sebagai salah satu syarat kelulusan pada Program Studi Sistem Informasi Universitas Telkom.

Selama melaksanakan tugas akhir ini, penulis mendapatkan banyak bantuan dan dukungan dari berbagai pihak. Untuk itu penulis ingin mengucapkan terima kasih kepada:

1. Bapak Soni Fajar Surya Gumilang sebagai pembimbing I yang telah memberikan bimbingan dalam menyelesaikan tugas akhir.
2. Bapak Muhammad Azani Hasibuan sebagai pembimbing 2 yang telah memberikan bimbingan dalam menyelesaikan tugas akhir.
3. Kevin Dion dan Fitri Fauziah sebagai teman kelompok yang selalu berbagi pikiran dalam menyelesaikan tugas akhir ini.
4. Dan kepada pihak-pihak yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa penulisan dalam tugas akhir ini masih memiliki kekurangan, baik karena keterbatasan kemampuan maupun cara penyampaian. Untuk itu penulis menerima kritik dan saran dari para pembaca untuk bahan perbaikan dalam penulisan karya ilmiah.

Bandung, 11 Juli 2017

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
LEMBAR PERNYATAAN ORISINALITAS .....	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
LEMBAR PERSEMBAHAN .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	vii
DAFTAR LAMPIRAN.....	x
DAFTAR GAMBAR DAN ILUSTRASI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR SINGKATAN DAN LAMBANG .....	xvi
BAB I PENDAHULUAN .....	1
I.1 Latar Belakang .....	1
I.2 Perumusan Masalah .....	4
I.3 Tujuan Penelitian .....	4
I.4 Batasan Penelitian .....	4
I.5 Manfaat Penelitian .....	5
I.6 Sistematik Pelaporan.....	5
BAB II TINJUAN PUSTAKA .....	7
II.1. Rekayasa Perangkat Lunak .....	7
Gambar II.1 Ruang Lingkup Rekayasa Perangkat Lunak (Mulyarto, 2008) .....	8
II.2. Teknologi Yang Relevan Dengan Judul Penelitian .....	9
II.2.1 PHP(Hypertext Preprocessor) .....	9
II.2.2 MySQL.....	9
II.2.3 HTML5.....	10
II.2.4 CSS3.....	11
II.2.5 Unified Modelling Language (UML).....	12
II.5.5. Deployment Diagram .....	13
II.2.6 Laravel.....	13
II.3. Metode yang digunakan pada penelitian.....	14
II.3.1.1 Scrum .....	14
II.3.1.1 Product Backlog.....	15
II.3.1.2 Sprint planning.....	16

II.3.1.3	Sprint.....	16
II.3.1.4	Scrum Theory.....	17
II.4.	Black-box testing .....	18
II.5.	Teori Perancangan Bisnis.....	19
II.6.	Tool Yang Digunakan.....	21
II.6.1.	Microsoft Visio.....	21
II.6.2.	Proto.io .....	22
II.6.3.	Power Designer .....	22
II.7.	Skala Likert.....	22
II.8.	Penelitian Sebelumnya.....	23
II.8.1.	Rancang Bangun Aplikasi Komunitas Donor Darah Berbasis Web Dan Android Yang Dilengkapi Layanan Informasi Geografis .....	23
II.8.2.	Android Application For PMI With Synchronization Using XML-RPC web Service At PMI DIY .....	24
<b>BAB III METODOLOGI PENELITIAN .....</b>		<b>25</b>
III.1	Model Konseptual.....	25
III.2	Sistematika Penelitian .....	27
III.2.1.	Tahap Pendahuluan .....	27
III.2.2.	Tahap Pengembangan Produk .....	27
III.2.3.	Tahap Pengembangan Bisnis.....	28
III.2.4.	Tahap Penutup.....	30
<b>BAB IV ANALISIS DAN PERANCANGAN .....</b>		<b>31</b>
IV.1.	Analisis dan Perancangan Bisnis.....	31
IV.1.1.	Analisis Kelayakan Bisnis.....	31
IV.1.2.	Analisis Potensi Pasar .....	34
IV.1.3.	Analisis Bisnis Model .....	34
IV.1.4.	Analisis Kompetitor .....	36
IV.2.	Analisis Perancangan Produk .....	37
IV.2.1.	Sprint Planing.....	37
IV.2.2.	Kebutuhan Bisnis .....	40
IV.2.3.	Spesifikasi Profil Aktor .....	40
IV.2.4.	Usecase Diagram Global .....	41
IV.2.5.	Usecase Form .....	42
IV.2.6.	Activity Diagram.....	51
IV.2.7.	Class Diagram Global .....	63

IV.2.8. ER Diagram.....	64
IV.2.9. Sequence Diagram.....	65
IV.2.10. Mockup Aplikasi.....	75
IV.2.11. Deployment Diagram.....	78
<b>BAB V IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>80</b>
V.1. Implementasi.....	80
V.1.1. Hasil Antarmuka.....	80
V.1.2. Implementasi Database.....	84
V.1.3. Hasil Implementasi Bisnis.....	85
V.2. Pengujian.....	88
V.2.1. Pengujian Fungsionalitas.....	88
V.2.2. Cross-Browser .....	91
V.2.3. Stress Testing .....	95
V.2.4. User Acceptance Testing.....	108
VI.1. Kesimpulan.....	113
VI.2. Saran.....	113
<b>DAFTAR PUSTAKA .....</b>	<b>114</b>
<b>LAMPIRAN A PROYEKSI KEUANGAN.....</b>	<b>116</b>
<b>LAMPIRAN B SURVEI.....</b>	<b>119</b>
<b>LAMPIRAN C DOKUMEN SCRUM.....</b>	<b>126</b>
<b>LAMPIRAN D DOKUMENTASI WAWANCARA USER ACCEPTANCE TESTING</b> <b>.....</b>	<b>138</b>

## DAFTAR LAMPIRAN

LAMPIRAN A PROYEKSI KEUANGAN.....	116
LAMPIRAN B SURVEI.....	119
LAMPIRAN C DOKUMEN SCRUM.....	126
LAMPIRAN D DOKUMENTASI WAWANCARA USER ACCEPTANCE TESTING .....	138

## DAFTAR GAMBAR DAN ILUSTRASI

Gambar II 1 Information Technology Project Management   7e. Augsburg.....	14
Gambar II 2 www.scrumalliance.org .....	16
Gambar II 3 www.scrumalliance.org .....	16
Gambar II 4 www.scrumalliance.org .....	17
Gambar III 1 Metode Konseptual Penelitian .....	25
Gambar III.2 Sistematika Penelitian .....	27
Gambar IV2.4. Usecase Diagram .....	41
Gambar IV2.5. 1 Activity Diagram Registrasi.....	51
Gambar IV2.5. 2 Activity Diagram Login .....	52
Gambar IV2.5. 3 Activity Diagram Event .....	53
Gambar IV2.5. 4 Activity Diagram About .....	54
Gambar IV2.5. 5 Activity Diagram Blood Vault.....	55
Gambar IV2.5. 6 Activity Diagram Vam Playground .....	56
Gambar IV2.5. 7 Activity Diagram Vam House.....	57
Gambar IV2.5. 8 Activity Diagram Contact .....	58
Gambar IV2.5. 9 Activity Diagram Profil .....	59
Gambar IV2.5. 10 Activity Diagram Permintaan Darah.....	60
Gambar IV2.5. 11 Activity Diagram Blood Vault.....	61
Gambar IV2.5. 12 Activity Diagram Lupa Password .....	62
Gambar IV.2.8. 1 Sequence diagram Login.....	65
Gambar IV.2.8. 2 Sequence Diagram Registrasi .....	66
Gambar IV.2.8. 3 Sequence diagram Event.....	67
Gambar IV.2.8. 4 Sequence diagram About .....	68
Gambar IV.2.8. 5 Sequence diagram Contact.....	69
Gambar IV.2.8. 6 Sequence diagram Home .....	70
Gambar IV.2.8. 7 Sequence diagram Lupa Password.....	70
Gambar IV.2.8. 8 Sequence diagram Permintaan Darah .....	71
Gambar IV.2.8. 9 Sequence diagram Profil .....	72
Gambar IV.2.8. 10 Sequence diagram Vam House .....	73
Gambar IV.2.8. 11 Sequence diagram Vam Playgorund .....	73
Gambar IV.2.8. 12 Sequence diagram Logout.....	74

Gambar IV.2.10. 1 Mockup Home .....	75
Gambar IV.2.10. 2 Mockup Event.....	75
Gambar IV.2.10. 3 Mockup Blood Vault.....	76
Gambar IV.2.10. 4 Mockup Vam Playground .....	76
Gambar IV.2.10. 5 Mockup Vam House .....	77
Gambar IV.2.10. 6 Mockup Registrasi .....	77
Gambar IV.2.10. 7 Mockup Login.....	78
Gambar IV.2.10. 8 Mockup Lupa Password.....	78
Gambar IV.2.11. Deployment Diagram.....	79
Gambar V.1.1 1 Halaman Utama.....	80
Gambar V.1.1 2 Halaman Event .....	80
Gambar V.1.1 3 Halaman About .....	81
Gambar V.1.1 4 Halaman Blood Vault.....	81
Gambar V.1.1 5 Halaman Vam Playground .....	82
Gambar V.1.1 6 Halaman Vam House .....	82
Gambar V.1.1 7 Halaman Contact .....	83
Gambar V.1.1 8 Halaman Registrasi .....	83
Gambar V.1.1 9 Halaman Login.....	84
Gambar V.1.3. 1 Tampilan Facebook E-Vampire .....	85
Gambar V.1.3. 2 Tampilan Instagram E-Vampire.....	86
Gambar V.1.3. 3 Tampilan Youtube E-Vampire .....	86
Gambar V.1.3. 4 Tampilan Website E-Vampire.....	87
Gambar V.1.3. 4 Gantungan Kunci E-Vampire.....	87
Gambar V.1.3. 5 Pin E-Vampire.....	88
Gambar V.1.3. 6 Stiker E-Vampire.....	88
Gambar V.1.3.7 Logo E-Vampire.....	88
Gambar V.2.2. 1 Test Browser .....	92
Gambar V.2.2. 2 Test Browser .....	93
Gambar V.2.2. 3 Test Browser .....	93
Gambar V.2.2. 4 Test Browser .....	94
Gambar V.2.2. 5 Test Browser .....	94
Gambar V.2.2. 6 Test Browser .....	95
Gambar V.2.3. 1 Config Testing.....	96

Gambar V.2.3. 2 Config Testing .....	96
Gambar V.2.3. 3 Click Testing .....	97
Gambar V.2.3. 4 Click Testing .....	97
Gambar V.2.3. 5 Click Testing .....	97
Gambar V.2.3. 6 Click Testing .....	98
Gambar V.2.3. 7 Click Testing .....	98
Gambar V.2.3. 8 Click Testing .....	98
Gambar V.2.3. 9 Click Testing .....	99
Gambar V.2.3. 10 Click Testing .....	99
Gambar V.2.3. 11 Click Testing .....	99
Gambar V.2.3. 12 Click Testing .....	100
Gambar V.2.3. 13 Time Testing .....	100
Gambar V.2.3. 14 Time Testing .....	101
Gambar V.2.3. 15 Time Testing .....	101
Gambar V.2.3. 16 Time Testing .....	102
Gambar V.2.3. 17 Time Testing .....	102
Gambar V.2.3. 18 Time Testing .....	102
Gambar V.2.3. 19 Time Testing .....	103
Gambar V.2.3. 20 Time Testing .....	103
Gambar V.2.3. 21 Time Testing .....	103
Gambar V.2.3. 22 Time Testing .....	104
Gambar V.2.3. 23 Ramp Testing .....	104
Gambar V.2.3. 24 Ramp Testing .....	105
Gambar V.2.3. 25 Ramp Testing .....	105
Gambar V.2.3. 26 Ramp Testing .....	106
Gambar V.2.3. 27 Ramp Testing .....	106
Gambar V.2.3. 28 Ramp Testing .....	106
Gambar V.2.3. 29 Ramp Testing .....	107
Gambar V.2.3. 30 Ramp Testing .....	107
Gambar V.2.3. 31 Ramp Testing .....	108
Gambar V.2.3. 32 Ramp Testing .....	108

## DAFTAR TABEL

Tabel III.2.4.2. 1 Rencana Bisnis Selanjutnya .....	29
Tabel IV.1.1. 1 Estimasi Biaya Operasional .....	31
Tabel IV.1.1. 2 Revenue Stream .....	32
Tabel IV.1.1. 3 Kalkulasi Laba Rugi .....	32
Tabel IV.1.1. 4 Cash flow .....	33
Tabel IV.1.1. 5 Cash-Benefit .....	33
Tabel IV.1.1. 6 Analisis ROI .....	33
Tabel IV.1.1. 7 Keputusan Kelayakan .....	34
Tabel IV.1.4. 1 Analisis Kompetitor .....	37
Tabel IV.2.1. 1 Sprint Planing .....	38
Tabel IV.2.1. 2 Product Backlog .....	39
Tabel IV.2.3. 1 Spesifikasi Profil Aktor .....	40
Tabel.2.5. 1 Usecase Form Login .....	42
Tabel.2.5. 2 Usecase Form Registrasi .....	43
Tabel.2.5. 3 Usecase Form Reset Password .....	43
Tabel.2.5. 4 Usecase Form Lihat Home .....	44
Tabel.2.5. 5 Usecase Form Lihat Event .....	44
Tabel.2.5. 6 Usecase Form Ikut Event .....	45
Tabel.2.5. 7 Usecase Form Vam Playground .....	45
Tabel.2.5. 8 Usecase Form Masukan Komentar .....	46
Tabel.2.5. 9 Usecase Form Lihat Blood Vault .....	46
Tabel.2.5. 10 Usecase Form Lihat Vam House .....	47
Tabel.2.5. 11 Usecase Form Lihat Profil .....	47
Tabel.2.5. 12 Usecase Form Ubah Profil .....	48
Tabel.2.5. 13 Usecase Form Logout .....	48
Tabel.2.5. 14 Usecase Form Permintaan Darah .....	49
Tabel.2.5. 15 Usecase Form Lihat Contact .....	49
Tabel.2.5. 16 Usecase Form Lihat About .....	50
Tabel V.2.2. Cross Browser .....	91
Tabel V.2.4.1 Pertanyaan Kuesioner .....	108
Tabel V.2.4.2 Umpan Balik Responden .....	109

Tabel V.2.4.3 Bobot Penilaian .....	109
Tabel V.2.4.4 Perhitungan Data.....	110
Tabel V.2.4.5 Interval Penilaian .....	111

## DAFTAR SINGKATAN DAN LAMBANG

Tambah

SINGKATAN	Nama	Pemakaian pertama kali pada halaman
PMI	Palang Merah Indonesia	1
IEEE	Institute of Electrical and Electronics Engineers	7
SWEBOK	Software Engineering Body of Knowledge	7
PHP	Hypertext Preprocessor	8
SQL	Structured Query Language	8
CSS	Cascading Style Sheet	9
UML	Unified Modeling Language	11
ROI	Return on Investment	27

# **BAB I PENDAHULUAN**

## **I.1 Latar Belakang**

Berdasarkan dari visi misi Presiden ke-4 yaitu mewujudkan kualitas hidup manusia Indonesia yang tinggi, maju dan sejahtera dalam pelaksanaan visi misi tersebut Presiden membuat agenda prioritas yang disebut nawa cita. Nawa cita merupakan sembilan agenda prioritas dari Presiden dalam mewujudkan visi misi Presiden ke-4 tidak semua nawa cita berfokus pada visi misi tersebut. Nawa cita yang mendukung visi misi ke-4 ini ialah nawa cita ke-5 yaitu meningkatkan hidup manusia Indonesia melalui peningkatan kualitas Pendidikan dan pelatihan dengan program “Indonesia Pintar”. Dalam implementasi nawa cita ke-5 dapat diambil kesimpulan pembangunan kesehatan merupakan bagian dari pembangunan nasional salah satunya mengenai ketersediaan darah (“Masyarakat Hidup Sehat Indonesia Kuat”, 2016).

Ketersediaan darah untuk donor, secara ideal adalah 2% dari jumlah penduduk yang disesuaikan dengan standar lembaga kesehatan internasional (WHO). Berdasarkan Badan Pusat Statistika jumlah penduduk di Indonesia sebesar 258.705.000 jiwa, maka idealnya dibutuhkan darah sebanyak 5.14.100 kantong darah. Akan tetapi pada tahun 2016 lalu jumlah darah yang terkumpul dari donor sebanyak 4.600.000 kantong darah. Sehingga secara nasional terdapat kekurangan kebutuhan darah sejumlah 574.100 kantong darah. Apabila dalam pengambilan darah donor per orang sebanyak 250 cc—500 cc maka kekurangan kebutuhan ideal sebesar 574.100 x 250 = 143.525.000 cc atau sejumlah 466.938 liter darah. Akibatnya rumah sakit masih sering mengalami kesulitan dalam memenuhi kebutuhan akan transfusi darah.

untuk Kota Bandung, berdasarkan Badan Pusat Statistik Kota Bandung, jumlah penduduk Kota Bandung tahun 2016 sebesar 3.596.623 jiwa. Idealnya kantong darah membutuhkan 2 % dari jumlah penduduk yang ada, maka idealnya dibutuhkan darah sebanyak 71.932 kantong. Bila diambil asumsi setiap pendonor akan menghasilkan maksimal 4 kantong darah, maka dibutuhkan 17.983 pendonor. Donor darah kebanyakan masih bersifat donor musiman, hanya dilakukan berkaitan dengan acara tertentu saja.

Dalam penjelasan diatas maka dapat disimpulkan kebutuhan akan darah di Indonesia tinggi tetapi darah yang terkumpul dari donor darah masih rendah. Berdasarkan hasil survei yang telah kami lakukan ke 58 orang, darah yang terkumpul dari donor darah masih rendah disebabkan oleh beberapa alasan dari masyarakat yang takut atau tidak mau untuk melakukan donor darah. Beberapa di antaranya berupa takut akan tertular penyakit, takut sakit disuntik, takut lemas setelah donor, trauma dan takut gemuk yang dapat kami ambil kesimpulan penyebab utama dari masyarakat tidak mau mendonorkan darahnya yaitu kurangnya edukasi masyarakat mengenai donor darah yang mengakibatkan kurangnya *awareness* (kesadaran) masyarakat untuk melakukan donor darah. Untuk hasil survei dapat dilihat pada Lampiran B Hasil Survei.

Survei yang telah dilakukan oleh PMI pun tercatat kebutuhan darah di Kota Bandung belum banyak terpenuhi, kematian karena kekurangan darah pun masih marak terjadi. Banyaknya *broadcast* tentang kebutuhan darah untuk pasien penderita penyakit yang mengeluarkan banyak darah, juga menjadi perhatian penulis. Tapi bukankah sebuah ironi ketika banyak *charity fair* atau sejenisnya yang mengadakan acara donor darah di mana-mana, tetapi masih banyak pasien yang harus mencari darah untuk kebutuhan keselamatan hidupnya (INFODATIN, 2013). Peningkatan kesadaran masyarakat tentang donor darah sudah banyak dilakukan oleh beberapa situs mengenai informasi donor darah maupun informasi ketersediaan darah seperti [ayodonor.pmi.or.id](http://ayodonor.pmi.or.id) , [dondarbdg.org](http://dondarbdg.org), aplikasi *Reblood* mau pun aplikasi *Easy Blood* yang berisi informasi-informasi maupun berita terkait donor darah. Aplikasi maupun situs-situs tersebut hadir dalam upaya peningkatan *awareness* masyarakat terhadap donor darah. Tetapi dibutuhkan aplikasi yang dapat meningkatkan informasi donor darah, agar informasi merata dan tidak menimbulkan masalah terkait kekurangan darah. Serta, aplikasi yang dapat mempermudah masyarakat dalam melakukan donor darah. Digambarkan pada tabel analisis fitur aplikasi maupun situs terkait informasi ketersediaan darah belum adanya aplikasi yang mengintegrasikan pendistribusian darah serta aplikasi yang mempermudah para pendonor dalam melakukan donor darah.

Dilihat dari analisis fitur dari situs dan aplikasi donor darah, muncul sebuah ide untuk mengatasi masalah kebutuhan darah dengan sebuah aplikasi berbasis web dan

*mobile* yang dapat membantu pendistribusian darah serta mempermudah para pendonor dengan menyediakan layanan antar - jemput pendonor menuju tempat donor darah maupun bank darah rumah sakit yang dinamakan E-Vampire. Diambil dari tokoh mitologi dan legenda pada awal ke-18 pemburu darah, E-Vampire diharapkan dapat membantu para pasien yang sangat membutuhkan darah sehingga dapat mengurangi jumlah angka kematian akibat kehabisan darah. Mengejar segmen pasar para pasien kebutuhan darah dan pendonor usia 18 hingga 40 tahun. Dengan kelebihanannya pada fitur yang membantu mempermudah para pendonor maupun pasien yang membutuhkan darah serta peningkatan *awareness* bagi masyarakat dalam ketersediaan dan kebutuhan darah di Kota Bandung. E-Vampire sebagai aplikasi yang membawa solusi bagi permasalahan sosial, pengembangan aplikasi ini akan dibawa menuju *next levelnya* dalam dunia bisnis yaitu *startup* . Hadirnya aplikasi-aplikasi baru penyelesaian masalah di Indonesia lainnya membuat peluang *startup* menjadi sangat besar. Kemunculan *startup – startup* di Indonesia pula menjadikan banyak referensi yang dilakukan dalam perancangan aplikasi dan pengembangan bisnis E-Vampire ini.

Pada perancangan aplikasi dan pengembangan bisnisnya, E-Vampire memiliki kebutuhan perancangan yang berubah-ubah. Masih banyak perubahan yang akan dilakukan ke depannya terkait semua aspek yang berada dalam perancangan aplikasi maupun pengembangan bisnis. Terlebih hadirnya organisasi legal yang diakui pemerintah yang telah lebih dulu terikat dengan masalah kesehatan seperti PMI menjadi satu tolak ukur perubahan dalam perancangan dan pengembangan bisnis E-Vampire. Dari perubahan-perubahan yang nantinya mungkin terjadi menjadi dasar dari perancangan aplikasi berbasis web ini akan didukung dengan pengerjaan metode *scrum*.

Pengembangan aplikasi berbasis *website* ini didukung dengan pengerjaan *software development life cycle agile* yaitu *scrum*. Ketersediaan darah tentu tidak akan sama setiap saatnya ada hal-hal yang harus diperhatikan sebelumnya ketika akan melakukan pengembangan sebuah aplikasi pendukung pendistribusian dan penyedia informasi darah di kota Bandung. Scrum merupakan salah satu varian dari metode *agile* yang iteratif dan *incremental*. Satu iterasi pada *Scrum* disebut dengan Sprint, di mana untuk setiap akhir dari Sprint produk yang sesuai dengan definisi

”Done” harus diraih. Kedua pendekatan pengembangan *software* ini sama-sama menekankan pada pentingnya komunikasi dan kolaborasi antar berbagai pihak. Sehingga aplikasi E-Vampire dapat memenuhi kebutuhan segala pihak.

Pada perancangan aplikasi E-Vampire menggunakan aplikasi berbasis *website* dikarenakan pada tahun 2013 , pengguna internet di Indonesia sudah mencapai lebih dari 60 juta. Mereka menggunakan internet untuk menemukan informasi yang sedang dibutuhkan. Dengan adanya aplikasi berbasis *website* maka akan meningkatkan visibilitas atau keberadaan perusahaan itu sendiri. Dengan adanya aplikasi berbasis *website* kemudahan dalam penyampaian informasi jauh lebih mudah dari pada harus menyebarkan melalui TV , radio , atau media massa. Pada era digital *website* menjadi salah satu *trend* di era digital ini. Maka dengan demikian peluang yang tercipta cukup membantu perusahaan untuk memasarkan produk tanpa ada beban dalam memasarkan produk. Untuk menangkap peluang tersebut, startup E-Vampire mengeluarkan aplikasi berbasis *website*, agar dapat memanfaatkan kesempatan tersebut untuk menciptakan saluran pemasaran langsung.

## I.2 Perumusan Masalah

Perumusan masalah yang akan dibahas pada penelitian ini adalah sebagai berikut.

1. Bagaimana meningkatkan *awareness* masyarakat tentang informasi donor darah melalui aplikasi E-Vampire ?
2. Rancangan bisnis seperti apa yang dapat meningkatkan *awareness* masyarakat tentang informasi donor darah ?

## I.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut.

1. Merancang fitur aplikasi E-Vampire untuk meningkatkan *awareness* tentang informasi donor darah berbasis *website*.
2. Merancang perencanaan bisnis dalam pembuatan *startup* aplikasi E-Vampire.

## I.4 Batasan Penelitian

Ruang lingkup yang akan dibahas pada penelitian ini adalah sebagai berikut.

1. Perancangan aplikasi E-Vampire di peruntukan untuk kegiatan meningkatkan *awareness* dan penyedia informasi donor darah di kota Bandung.
2. Perangkat yang digunakan PC dan *gadget*.

3. Pengembangan menggunakan *scrum*.
4. Data yang ditampilkan dalam sistem ini meliputi stok, rumah sakit, dan donor darah.
5. Perancangan aplikasi E-Vampire di peruntukan untuk *user*.

#### **I.5 Manfaat Penelitian**

Dengan adanya penelitian ini diharapkan dapat memberikan manfaat sebagai berikut.

1. Memberikan rancangan *user friendly* sehingga dapat memberikan rasa nyaman pada pengguna dalam menggunakan aplikasi E-Vampire.
2. Memberikan rancangan sistem informasi yang informatif, sehingga dapat meningkatkan *awareness* pengguna.
3. Memberikan rancangan sistem informasi yang memberitahukan tempat pelayanan kesehatan secara tepat.

Manfaat E-Vampire bagi para perancang adalah sebagai berikut:

1. Menjadi tugas akhir perkuliahan di Universitas Telkom.
2. Membuat sebuah perusahaan *startup* terbaru dengan aplikasi E-Vampire.

#### **I.6 Sistematik Pelaporan**

Sistematika penulisan yang digunakan dalam laporan ini terdiri dari lima bab. Adapun garis besar penulisan dapat diuraikan sebagai berikut :

##### **Bab I.Pendahuluan**

Pada bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, tujuan penelitian, batasan penelitian, manfaat penelitian, dan sistematika penelitian.

##### **Bab II.Tinjauan Pustaka**

Pada bab ini berisi literatur yang relevan dengan permasalahan yang diteliti dan dibahas pula hasil-hasil penelitian terdahulu. Bagian kedua membahas hubungan antar konsep yang menjadi kajian penelitian dan uraian kontribusi penelitian.

##### **Bab III.Metode Penelitian**

Pada bab ini berisi tentang model konseptual dari penelitian dan sistematika penelitian berdasarkan metodologi pengembangan system yang telah ditentukan yaitu *scrum*.

#### Bab IV. Analisis dan Perancangan

Pada bab ini dibahas mengenai model bisnis yang akan dilakukan untuk permasalahan yang telah diambil, mengidentifikasi aktor yang terlibat dalam sistem ini serta merancang kebutuhan – kebutuhan yang diperlukan dalam pembuatan sistem ini. Perancangan system ini dimulai dari identifikasi keperluan user, *Usecase diagram*, *activity diagram*, *class diagram*, *sequence diagram*, dan *design interface*.

#### Bab V. Implementasi dan Pengujian

Bab ini berisi tentang implementasi dari rancangan serta desain tampilan dan pengujian untuk memastikan sistem ini layak untuk digunakan, maka dilakukan pengujian menggunakan metode *Black Box Testing*.

#### Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab terakhir, berisi kesimpulan dari hasil penelitian dari awal mula masalah, terbentuknya solusi, analisis sistem hingga implementasi sistem yang di lanjutkan dengan saran tentang sistem yang diharapkan dapat dikembangkan.

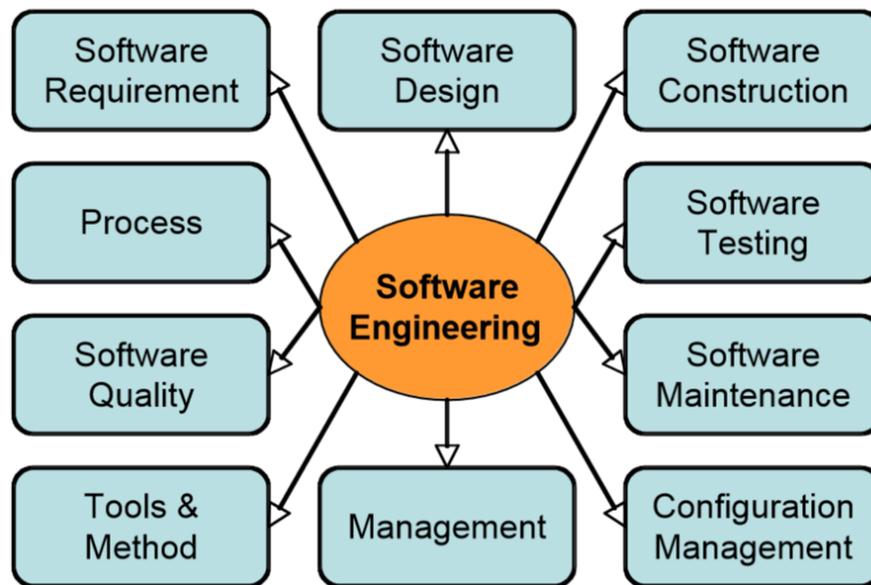
## BAB II TINJUAN PUSTAKA

### II.1. Rekayasa Perangkat Lunak

Istilah Rekayasa Perangkat Lunak (RPL) merupakan terjemahan dari istilah *Software Engineering*. Istilah *Software Engineering* mulai dipopulerkan tahun 1968 pada *Software Engineering Conference* yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat program komputer.

Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi. Jadi Rekayasa Perangkat Lunak (RPL) merupakan suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pelanggan, menentukan spesifikasi dari kebutuhan pengguna, desain, pembuatan, pengujian sampai pemeliharaan sistem setelah digunakan (Mulyarto, 2008).

Rekayasa perangkat lunak memiliki beberapa tujuan yaitu memperoleh biaya produksi perangkat lunak yang rendah, menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepat waktu, menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis *platform*, dan menghasilkan perangkat lunak yang biaya perawatannya rendah. Berdasarkan definisi Rekayasa Perangkat Lunak di atas, ruang lingkup Rekayasa Perangkat Lunak dapat dilihat pada gambar berikut.



**Gambar II.1 Ruang Lingkup Rekayasa Perangkat Lunak (Mulyarto, 2008)**

1. *Software requirements*, berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak.
2. *Software design*, mencakup proses penentuan arsitektur, komponen, antarmuka, dan karakteristik lain dari perangkat lunak.
3. *Software construction*, berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritme, pembuatan, pengujian, dan pencarian kesalahan.
4. *Software testing*, meliputi pengujian pada keseluruhan perilaku perangkat lunak.
5. *Software maintenance*, mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan.
6. *Software configuration management*, berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu.
7. *Software engineering tools and methods*, mencakup kajian teoretis tentang alat bantu dan metode RPL.
8. *Software engineering process*, berhubungan dengan definisi, implementasi, pengukuran, pengelolaan, perubahan dan perbaikan proses RPL.  
*Software quality*, menitikberatkan pada kualitas dan daur hidup *software*.

## **II.2. Teknologi Yang Relevan Dengan Judul Penelitian**

### **II.2.1 PHP(*Hypertext Preprocessor*)**

PHP adalah sebuah bahasa pemrograman scripting untuk membuat halaman web yang dinamis. PHP dikatakan sebagai sebuah *serve side embedded script language* artinya *syntaks* dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML yang seperti biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada tampilan web browser, tetapi prosesnya secara keseluruhan dijalankan di server. Terdapat beberapa pandangan dalam mengartikan kata PHP, kurang lebih dapat diartikan sebagai *Hypertext Preeprocessor*. PHP merupakan bahasa pemrograman yang hanya dapat berjalan pada server dan hasilnya dapat ditampilkan pada Client. PHP merupakan produk Open Source yang dapat digunakan secara gratis tanpa harus membayar untuk menggunakannya. PHP merupakan bahasa standar yang digunakan dalam dunia Website, PHP adalah bahasa pemrograman yang berbentuk skrip yang diletakkan di dalam server web. Jika kita lihat dari sejarah mulainya PHP diciptakan dari ide Rasmus Lerdof untuk kebutuhan pribadinya, skrip tersebut sebenarnya dimaksudkan untuk digunakan sebagai keperluan 21 membuat Website pribadi, akan tetapi kemudian dikembangkan lagi sehingga menjadi sebuah bahasa yang disebut “Personal Home Page”.

### **II.2.2 MySQL**

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasi nal (RD MS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama

untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Keandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja mengoptimasi dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data non transaksional. Pada modus operasi non transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non transaksional tidak ada jaminan atas *reliability* terhadap data yang tersimpan, karenanya modus non transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan *reliability* data seperti aplikasi *blogging* berbasis web (*wordpress*), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non transaksional.

### **II.2.3 HTML5**

HTML5 adalah sebuah bahasa markah untuk menstrukturkan dan menampilkan isi dari Waring Wera Wanua (*World Wide Web Consortium*, W3C), sebuah teknologi inti dari Internet. HTML5 adalah revisi kelima dari HTML (yang pertama kali diciptakan pada tahun 1990 dan versi keempatnya, HTML4, pada tahun 1997) dan hingga bulan Juni 2011 masih dalam pengembangan. Tujuan utama pengembangan HTML5 adalah untuk memperbaiki teknologi HTML agar mendukung teknologi multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

HTML5 merupakan salah satu karya Konsortium Waring Wera Wanua (*World Wide Web Consortium*, W3C) untuk mendefinisikan sebuah bahasa markah tunggal yang dapat ditulis dengan cara HTML ataupun XHTML. HTML5 merupakan jawaban atas pengembangan HTML 4.01 dan XHTML 1.1 yang selama ini berjalan terpisah, dan diimplementasikan secara berbeda-beda oleh banyak perangkat lunak pembuat web.

## II.2.4 CSS3

CSS3 adalah *Cascading Style Sheet* versi ke 3, yaitu pengatur dan pengendali tampilan sebuah halaman *blog / web*. CSS3 melakukan penataan terhadap komponen HTML maupun XHTML pada halaman web sehingga menghasilkan tampilan yang ramah di mata atau retina friendly. CSS versi ketiga dapat melakukan berbagai pembaruan terhadap kemampuannya beradaptasi dengan teknologi multimedia yang semakin modern. Kemampuan yang di miliki CSS3 antara lain:

1. CSS3 mampu membuat efek animasi secara independen tanpa membutuhkan software tambahan seperti Adobe Flash dan Microsoft Silverlight. Karena hanya menggunakan kode - kode maka web tidak akan berat atau tidak lama dan tidak boros kuota untuk membukanya.
2. CSS3 mampu memberikan efek grafis terhadap teks, kolom, *table* serta penataan huruf.
3. CSS3 mampu menampilkan berbagai macam dan jenis huruf selain *web-safe fonts*.
4. CSS3 mampu merekonstruksi secara visual tampilan *block*, seperti mengubah ukuran kotak, transformasi 2D/ 3D, memberikan efek sudut (radius) dan *shadow* (bayangan).
5. CSS3 mampu memanipulasi secara visual warna, desain maupun tekstur suatu halaman HTML atau XHTML.

Kelebihan dan Kekurangan CSS3, Kelebihan :

1. Dapat Mengatur dan mempercantik *interface website*.
2. Jika ingin membuat file CSS3 sebaiknya menggunakan CSS *external* atau terpisah dari file HTML dan dapat dipanggil di file HTML. Hal ini dapat mempermudah kita jika terjadi pengeditan coding.
3. Dengan CSS3, situs website dapat berkembang dan lebih interaktif lagi.
4. Bisa mengurangi ukuran *file* yang akan di muat dan akan menjadi lebih ringan.
5. Lebih mudah dan simpel serta dapat menghindari penggunaan tag yang berulang-ulang.

6. Banyak yang beranggapan bahwa peran jQuery sudah tidak dibutuhkan lagi setelah CSS3 semakin berkembang ke depannya. Apalagi dari segi *size*, jQuery jauh lebih besar. Tidak hanya jQuery saja, peran Flash saja sudah tergantikan dengan adanya CSS3.

Kekurangan :

1. Penulisan kode berbeda dengan versi sebelumnya.
2. Penulisan kode dari setiap *browser* berbeda. karena kita harus membuat penulisan kode dari seluruh *browser* satu persatu.

### II.2.5 Unified Modelling Language (UML)

*Unified Modeling Language* (UML) adalah notasi grafis untuk menggambar diagram konsep perangkat lunak. UML dapat digunakan untuk menggambar diagram domain masalah, rancangan perangkat lunak yang diusulkan, atau implementasi perangkat lunak (Martin,2002). Awalnya standar bahasa permodelan dibuat oleh metodologis dan perusahaan pengusul standar permodelan bahasa yaitu *Object Management Group* (OMG) yang merupakan sebuah *consortium* yang menciptakan dan mempertahankan standar untuk perindustrian komputer. Di tahun 1997, OMG mengadopsi UML sebagai sebuah standar.

#### 1. Use case Diagram

*Use case* adalah teknik untuk menangkap persyaratan fungsional suatu sistem. *Use case* berguna untuk menggambarkan interaksi antar pengguna dan sistem, dan memberikan narasi tentang bagaimana sistem digunakan. Setiap *use case* memiliki aktor utama yang meminta sistem untuk memberi sebuah layanan. Aktor utama adalah aktor dengan tujuan yang dipenuhi *use case* dan biasanya merupakan sebuah inisiator dari *use case* (Fowler, 2004)

#### 2. Activity Diagram

*Activity diagram* adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan alur kerja. Dalam banyak hal, *activity diagram* memiliki peran yang mirip dengan diagram alir (*flowcharts*), namun perbedaan utama antara *activity diagram* dengan diagram alir adalah dukungan perilaku paralel (Fowler, 2004). *Activity diagram* telah terjadi beberapa perubahan pada versi UML, sehingga

tidak mengherankan dan telah diperluas secara signifikan dan diubah lagi pada UML 2.

### 3. Class Diagram

*Class diagram* menggambarkan jenis objek dalam sistem dan berbagai jenis hubungan statis yang ada di antar *class diagram*. *Class diagram* juga menunjukkan sifat dan operasi *class* dan batasan yang berlaku terhadap objek yang terhubung. UML menggunakan fitur “*term*” sebagai istilah umum yang meliputi properti dan operasi dari sebuah kelas (Fowler, 2004).

### 4. Sequence Diagram

*Interaction diagram* menggambarkan kelompok objek berkolaborasi dalam beberapa perilaku. UML mendefinisikan beberapa bentuk diagram interaksi, yang paling umum adalah *sequence diagram*. Biasanya, *sequence diagram* menangkap perilaku dari sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah contoh objek dan pesan yang dilewatkan di antara benda – benda di dalam *use case* (Fowler, 2004).

#### **II.5.5. Deployment Diagram**

*Deployment diagram* menggambarkan tata letak fisik dari sistem, yang memperlihatkan perangkat lunak dan perangkat keras mana yang digunakan (Fowler, 2004). *Deployment diagram* juga dapat berfungsi sebagai:

1. Petunjuk di mana setiap komponen – komponen perangkat keras dan perangkat lunak di – *install*.
2. Petunjuk hubungan komunikasi antara komponen perangkat keras.
3. Petunjuk struktur dari sistem *run-time*.

#### **II.2.6 Laravel**

Framework Laravel dibuat oleh **Taylor Otwell**, proyek Laravel dimulai pada April 2011. Awal mula proyek ini dibuat karena Otwell sendiri tidak menemukan framework yang *up-to-date* dengan versi PHP. Mengembangkan framework yang sudah ada juga bukan merupakan ide yang bagus karena keterbatasan sumber daya. Dikarenakan beberapa keterbatasan tersebut, Otwell membuat sendiri framework dengan nama Laravel. Oleh karena itu Laravel mengisyaratkan PHP versi 5.3 keatas (Abdul, 2014).

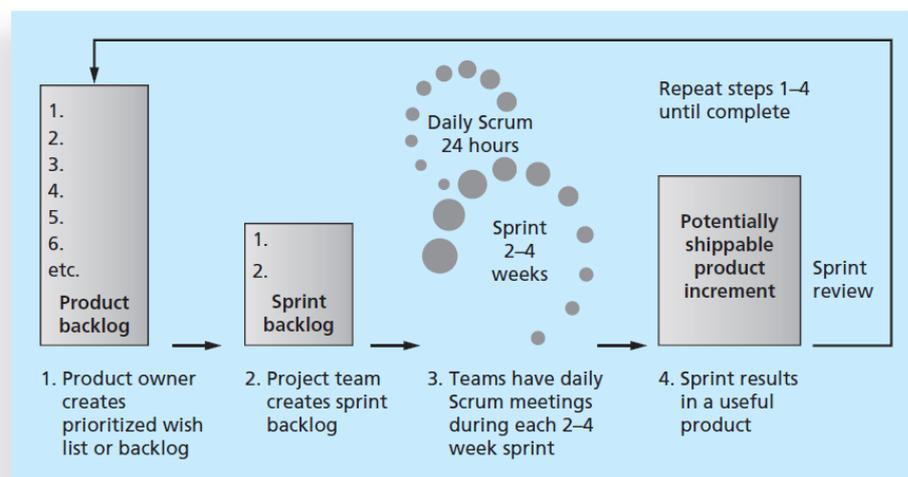
Laravel merupakan framework aplikasi web yang ekspresif dan sintaks yang elegan. Kami membangun framework ini dengan menyenangkan dan dari pengalaman-pengalaman kreatif agar menjadi produk yang memuaskan. Laravel berusaha mencoba menjadi framework yang mudah digunakan dengan mengurangi tugas-tugas umum yang sering digunakan dalam sebagian besar proyek-proyek web seperti *otentikasi*, *routing*, *session*, dan *caching* (Abdul Rahman, 2014).

Laravel membuat proses development yang menyenangkan bagi pengembang tanpa mengurangi fungsionalitas aplikasi. Dengan harapan, pengembang dapat membuat rangkaian kode-kode terbaik. Laravel berusaha untuk menggabungkan yang terbaik dari apa ada dalam framework web lain, termasuk framework yang menggunakan bahasa lain, seperti Ruby on Rails, ASP.NET MVC, dan Sinatra (Abdul, 2014).

## II.3. Metode yang digunakan pada penelitian

### II.3.1.1 Scrum

Scrum adalah metode untuk menyelesaikan proyek-proyek dengan kompleks, lingkup inovatif kerja. Kerangka dasar Scrum diringkas dalam ilustrasi pada Gambar II 1:



© Cengage Learning 2014

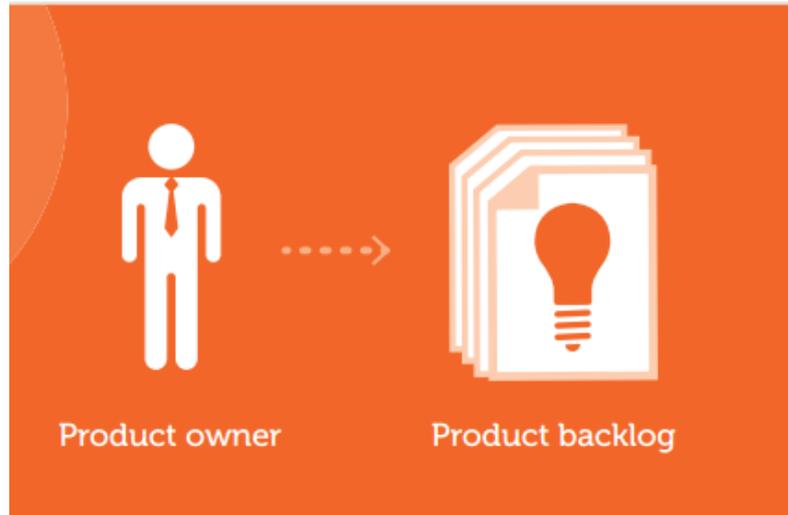
**Gambar II 1** *Information Technology Project Management / 7e. Augsburg*

1. Seorang product owner menciptakan daftar keinginan diprioritaskan disebut *product backlog*.
2. Selama sprint planning, tim mengambil bagian dari *product backlog* yaitu *sprint backlog*, dan memutuskan cara menerapkan bagian-bagian tersebut.
3. Tim ini memiliki sejumlah waktu atau *sprint* untuk menyelesaikan pekerjaan, biasanya setiap 2-4 minggu memenuhi setiap hari untuk menilai kemajuan (*daily Scrum*).
4. Sepanjang proses, *Scrum Master* membuat tim fokus pada sprint yang telah ditetapkan.
5. Pada akhir sprint, pekerjaan harus berpotensi dapat digunakan atau siap untuk diserahkan kepada pelanggan atau di tujukan kepada setiap stakeholdernya.
6. Sprint berakhir dengan *review sprint* dan *retrospektif*.
7. Sebagian *sprint* berikutnya dimulai, tim memilih bagian lain dari *produk backlog* dan mulai bekerja kembali.

Siklus berulang sampai bagian – bagian *product backlog* telah selesai, anggaran habis, atau tenggat waktu tiba. Proses pada *scrum* memastikan bahwa pekerjaan yang paling berharga telah selesai ketika proyek berakhir. Metode *scrum* awalnya diterapkan untuk proyek-proyek pengembangan perangkat lunak, tetapi hari ini jenis lain dari proyek menggunakan teknik ini untuk membantu fokus pada kerja tim, menyelesaikan pekerjaan yang paling penting pertama, dan menambahkan nilai bisnis.

#### **II.3.1.1 Product Backlog**

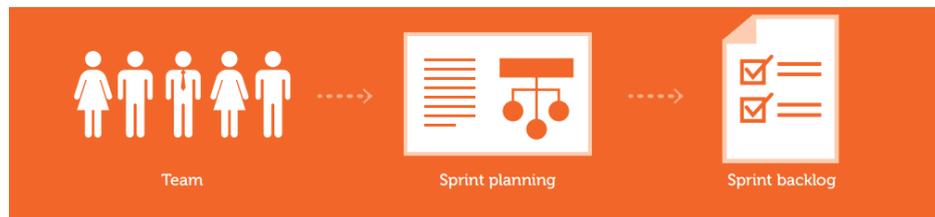
*Product owner* membuat daftar prioritas ide untuk rancangan produk, disebut *product backlog*. *Product backlog* membantu tim memecah rancangan produk menjadi lebih kecil, bagian kecil tersebut lebih mudah dikelola dan membangunnya secara bertahap dalam serangkaian periode waktu yang singkat disebut *sprint*. *Sprint* biasanya berlangsung satu sampai empat minggu



Gambar II 2 [www.scrumalliance.org](http://www.scrumalliance.org)

### II.3.1.2 *Sprint planning*

Selama *sprint planning*, tim mengeluarkan bagian kecil dari *product backlog* untuk membuat *sprint backlog*, dan kemudian memutuskan bagaimana menyelesaikan item - item selama sprint berikutnya.



Gambar II 3 [www.scrumalliance.org](http://www.scrumalliance.org)

### II.3.1.3 *Sprint*

Selama *sprint*, tim bertemu setiap hari, dalam *daily scrum*, untuk menilai kemajuan dan membuat penyesuaian yang diperlukan. Dalam proses *scrum* yang berlangsung, *Scrum Master* membuat tim fokus pada *sprint* yang telah di tentukan. Scrum Master juga menghilangkan hambatan bagi tim, sehingga semua orang bisa fokus dalam pekerjaan mereka. Pada akhir *sprint*, pekerjaan harus berpotensi dipakai dan siap untuk diserahkan kepada pelanggan atau di tujukan kepada setiap stakeholder . *Sprint* berakhir dengan review produk dan *retrospektif* tim proses kerja, hubungan, dan alat-alat. Setelah sprint selesai, tim memilih potongan lain dari *backlog* produk, dan sprint berikutnya dimulai.



Gambar II 4 [www.scrumalliance.org](http://www.scrumalliance.org)

Scrum memiliki kekuatan untuk mengubah manajemen proyek di setiap industri, setiap bisnis, dan bahkan di kehidupan secara umum. Dengan menggunakan Scrum, Anda akan menjadi lebih tangkas, bereaksi lebih cepat dan merespon lebih akurat dengan perubahan yang tak terelakkan yang datang dalam proses pengerjaan. Fokus, bekerja sama, dan berkomunikasi, dapat mencapai apa yang benar-benar perlu dilakukan - berhasil. *Scrum* adalah kerangka *Agile* solid dan sukses yang telah diterapkan pada berbagai proyek dan tim.

#### ***II.3.1.4 Scrum Theory***

Scrum didirikan pada teori kontrol proses empiris, atau empirisme. Empirisme menegaskan bahwa pengetahuan berasal dari pengalaman dan pengambilan keputusan berdasarkan apa yang diketahui. Scrum mempekerjakan berulang, pendekatan *incremental* untuk mengoptimalkan prediktabilitas dan pengendalian risiko. Tiga pilar menegaskan setiap pelaksanaan pengendalian empiris proses: transparansi, inspeksi, dan adaptasi.

##### **1. *Transparency***

Aspek penting dari proses harus terlihat untuk mereka yang bertanggung jawab untuk hasilnya. Transparansi memerlukan aspek yang didefinisikan oleh standar

umum sehingga pengamat memiliki pemahaman yang sama tentang apa yang sedang dilihat.

## 2. *Inspection*

Pengguna Scrum harus sering memeriksa Scrum artifak dan kemajuan menuju Sasaran Sprint untuk mendeteksi variasi yang tidak diinginkan. inspeksi mereka tidak boleh begitu sering bahwa inspeksi mendapat di jalan pekerjaan. Inspeksi yang paling bermanfaat ketika rajin dilakukan oleh inspektur terampil pada titik kerja.

## 3. *Adaptation*

Jika seorang inspektur menentukan bahwa satu atau lebih aspek dari suatu proses menyimpang di luar batas yang dapat diterima, dan bahwa produk yang dihasilkan akan diterima, proses atau bahan yang diproses harus disesuaikan. Penyesuaian harus dibuat sesegera mungkin untuk meminimalkan penyimpangan lebih lanjut.

Scrum menetapkan empat acara formal untuk inspeksi dan adaptasi, seperti yang dijelaskan dalam Scrum Events bagian dari dokumen ini:

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

## **II.4. Black-box testing**

*Black-box testing* adalah pengujian yang mengabaikan mekanisme internal suatu sistem atau komponen dan hanya berfokus pada keluaran yang dihasilkan sebagai respon terhadap *input* dan kondisi eksekusi yang dipilih (Williams, 2006). Berikut beberapa jenis dari *black-box testing*:

### 1. *Functional and system testing*

Menggunakan teknik *black-box testing*, penguji memeriksa rancangan tingkat tinggi dan spesifikasi persyaratan pengguna untuk merencanakan uji kasus untuk memastikan kode tersebut melakukan apa yang dimaksudkan. *Functional testing* memastikan bahwa fungsi yang ditentukan dalam spesifikasi kebutuhan bekerja. *System testing* adalah pengujian yang dilakukan pada sistem yang lengkap untuk mengevaluasi sistem terhadap persyaratan yang ditentukan. *System testing*

melibatkan penempatan program baru di lingkungan yang berbeda untuk memastikan program bekerja di lingkungan pelanggan yang khas dengan berbagai versi dan jenis sistem operasi dan aplikasi. Berikut beberapa contoh dari *system testing*:

a. *Stress testing*

Merupakan pengujian yang dilakukan untuk mengevaluasi suatu sistem atau komponen pada atau di luar batas spesifikasi atau persyaratan.

b. *Performance testing*

Merupakan pengujian yang dilakukan untuk mengevaluasi pemenuhan suatu sistem atau komponen dengan persyaratan kinerja yang spesifik.

c. *Usability testing*

Merupakan pengujian yang dilakukan untuk mengevaluasi sejauh mana pengguna dapat belajar untuk mengoperasikan, memasukkan, dan menafsirkan keluaran dari sistem atau komponen.

2. *Acceptance testing*

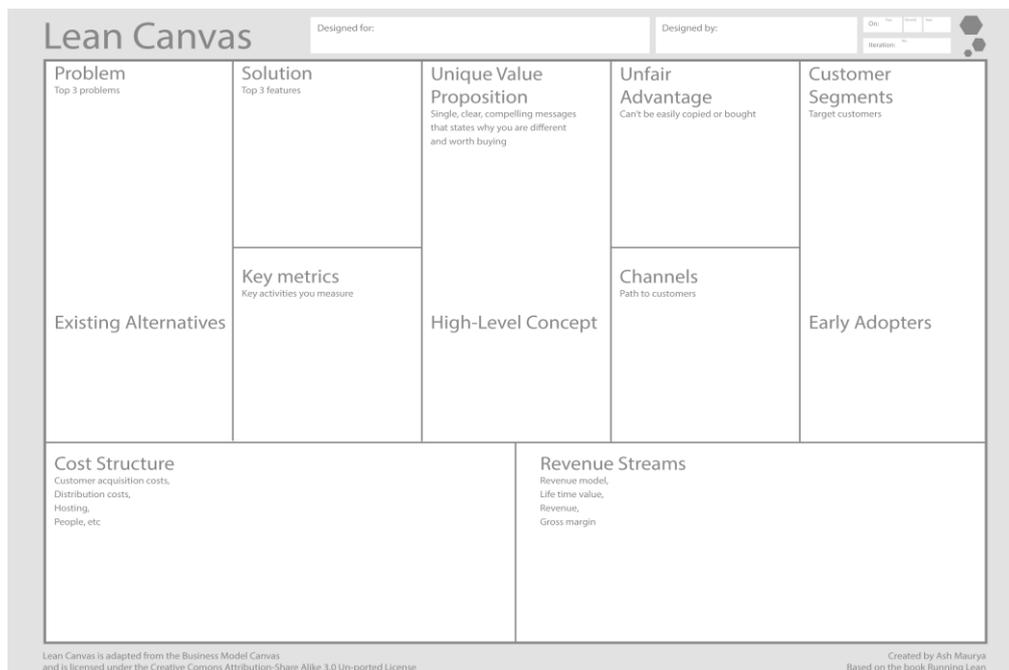
*Acceptance testing* adalah pengujian formal yang dilakukan untuk menentukan apakah suatu sistem memenuhi kriteria penerimaannya dan untuk memungkinkan pelanggan menentukan apakah akan menerima sistem atau tidak. Pengujian ini sering dilakukan sebelumnya oleh pengguna dan diberikan kepada tim penguji untuk dilakukan sebelum produk diberikan pada pengguna. Pengguna berhak menolak perangkat lunak jika tidak lulus dari *acceptance testing*.

## **II.5. Teori Perancangan Bisnis**

*Business model canvas* adalah (1) Sebuah bahasa bersama untuk menggambarkan, visualisasi, menilai, dan mengubah model bisnis. (2) Sebuah model bisnis yang menggambarkan pemikiran tentang bagaimana sebuah organisasi menciptakan, memberikan, dan menangkap nilai. (Alexander Osterwalder, 2010)

Pada *business model canvas*, ada sembilan hal yang menggambarkan elemen utama dalam setiap bisnis. Kesembilan elemen itu antara lain *customer segment*, *value proposition*, *channel*, *customer relationship*, *revenue stream*, *key resource*,

*key activities, key partners, dan cost.* Konsep sederhana ini memudahkan perusahaan-perusahaan besar tetapi tidak sesuai untuk perusahaan startup. Maka, Ash Maurya mengadopsi *business model canvas* milik Alexander Osterwalder menjadi *lean canvas*. *Lean canvas* adalah alat pengujian hipotesis model bisnis dan validasi yang menyederhanakan bagaimana mendokumentasikan model bisnis, mengukur kemajuan, dan berkomunikasi belajar dengan para *stakeholder* internal dan eksternal. (Ash Maurya, 2010) *Lean canvas* memiliki sembilan elemen utama dalam setiap bisnisnya. Kesembilan elemen itu antara lain *problem, customer segment, unique value proposition, solution, channel, revenue stream, cost structure, key metrics, dan unfair advantage.*



**Gambar II.5. 1 Lean canvas (Ash Maurya, 2010)**

**1. Problem**

*Problem* dapat dijabarkan menjadi tiga masalah utama untuk segmen pelanggan.

**2. Customer segment**

Target pelanggan. Identifikasikan tiga sampai empat jenis pengguna yang dapat beralih kepada solusi yang ditawarkan

**3. Unique value proposition**

Sebuah pesan yang jelas dan menarik. Pesan ini harus menjelaskan apa yang dilakukan, bagaimana perbedaan, dan mengapa layak untuk berinvestasi pada solusi bisnis dan jelaskan apa yang menonjol tentang produk.

#### 4. *Solution*

Solusi apa yang diberikan untuk masalah pelanggan? Hadirkan penjelasan mengenai layanan yang ditawarkan.

#### 5. *Channel*

Jalan menuju pelanggan. Bagaimana interaksi dengan pelanggan, dan menginformasikan perkembangan dan layanan?

#### 6. *Revenue Stream*

Sumber keuangan apa yang akan memberi masukan pada perusahaan? Bagaimana menghasilkan pendapatan? Hadirkan model penetapan harga untuk produk atau layanan yang diberikan, lalu sorot sumber pendapatan, penjualan ikan, biaya berlangganan, atau penjualan aset lainnya.

#### 7. *Cost Structure*

Berapa biaya untuk memulai dan mempertahankan bisnis? Pertimbangkan setiap tahap *startup* untuk membuat situs web dan memperoleh pengguna, untuk mempekerjakan karyawan dan memproduksi barang, memasarkan produk dan mengarahkan produk ke pelanggan.

#### 8. *Key Metrics*

Bagaimana melacak keterlibatan pelanggan, kegembiraan, dan penggunaan produk ?

#### 9. *Unfair Advantage*

Bagaimana produk melebihi dari pesaing? Apa yang menempatkan produk yang menjadi solusi lebih bagus dari kompetitor lainnya?

## **II.6. Tool Yang Digunakan**

### **II.6.1. Microsoft Visio**

Microsoft Visio adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir(*flowchart*), *brainstorm*, dan skema jaringan yang dirilis oleh Microsoft Corporation. Aplikasi ini menggunakan grafik vektor untuk membuat diagram – diagramnya. Visio bukanlah buatan Microsoft Corporation tetapi merupakan hasil akuisisi dari

Visio Corporation pada tahun 2000. Diagram – diagram yang dibuat oleh penulis menggunakan *software* Microsoft Visio 2013.

### II.6.2. Proto.io

Proto.io merupakan aplikasi *prototyping* yang diluncurkan pada tahun 2011 dan dikembangkan oleh laboratorium divisi SNQ Digital. Proto.io awalnya dirancang untuk membuat prototipe pada perangkat *mobile*, tetapi sekarang telah diperluas sehingga dapat membuat prototipe apa saja seperti kamera digital, mobil, pesawat terbang, hingga game *smart TV*. Penulis menggunakan proto.io sebagai alat bantu pembuatan *mockup* aplikasi *mobile* E-Vampire.

### II.6.3. Power Designer

Power Designer adalah sebuah perangkat lunak yang memiliki banyak fungsi di antaranya adalah untuk merancang serta mengelola *database* (Rizak, 2015). Power Designer sangat cocok untuk *database* yang berukuran besar serta memiliki tingkat kompleksitas yang cukup rumit. Pembuatan UML menggunakan Power Designer memiliki beberapa keuntungan yaitu:

1. Perancangan *database* mayoritas menggunakan *interface* berupa tampilan grafik.
2. *Database* dapat diciptakan independen dari RDBMS.
3. Migrasi *database* menjadi mudah.
4. Hampir tidak membutuhkan pengetahuan mengenai bahasa SQL.

### II.7. Skala Likert

Merupakan teknik *self-report* bagi pengukuran sikap dimana subjek diminta untuk mengidentifikasi tingkat kesetujuan atau ketidaksetujuan mereka terhadap masing – masing pernyataan (Hendri, 2009). Dalam pembuatan skala *likert*, pemilik aplikasi yang akan diuji membuat beberapa pernyataan yang berhubungan dengan suatu isu atau objek, lalu subjek atau responden diminta untuk mengidentifikasi tingkat kesetujuan atau ketidaksetujuan mereka terhadap masing – masing pernyataan. Teknik ini memiliki kelebihan dan kekurangan, di antaranya:

Kelebihan	Kekurangan
1. Sederhana untuk digunakan	1. Kecenderungan terpusat

2. Memungkinkan untuk menghasilkan skala yang andal	2. Sulit untuk menunjukkan validitas
3. Mudah dibaca dan diselesaikan oleh responden	3. Keinginan social
	4. Persetujuan

## II.8. Penelitian Sebelumnya

### II.8.1. Rancang Bangun Aplikasi Komunitas Donor Darah Berbasis Web Dan Android Yang Dilengkapi Layanan Informasi Geografis

(Sumber: Saputra, K. Y. (2016). RANCANG BANGUN APLIKASI KOMUNITAS DONOR DARAH BERBASIS WEB DAN ANDROID YANG DILENGKAPI LAYANAN INFORMASI GEOGRAFIS . Bali: Universitas Udayana.)

Pada penelitian ini dilakukan rancang bangun aplikasi komunitas donor darah berbasis web dan mobile Android yang juga dilengkapi dengan layanan informasi geografis. Aplikasi komunitas donor darah yang dibangun merupakan perantara antara pendonor dan penerima donor darah yang penyebaran informasinya memanfaatkan media internet. Aplikasi yang dibangun memiliki beberapa fitur yang dapat memberikan informasi dengan jelas kepada pendonor maupun penerima darah, seperti fitur *notification* yang terdapat pada aplikasi berbasis *mobile* (Android) sehingga setiap user dapat mengetahui informasi mengenai kebutuhan darah atau informasi mengenai event donor darah. Aplikasi komunitas donor darah ini juga dilengkapi dengan fitur layanan informasi geografis, sehingga pengguna dapat mengetahui lokasi rumah sakit dan lokasi suatu event donor darah. Pengembangan aplikasi atau perangkat lunak yang telah dilakukan menerapkan model *waterfall* yang prosesnya didahului dengan analisis kebutuhan sistem, perancangan sistem, pembuatan sistem, hingga pengujian system.

Dari penelitian ini didapatkan donor darah akan sangat terbantu jika penyebaran informasi di dukung oleh aplikasi dan internet. Layanan - layanan yang diberikan dapat di kembangkan agar tingkat *awareness* pendonor dapat di tingkatkan. Dimana layanan-layanan tersebut ialah layanan informasi tempat donor darah dan rumah sakit.

### ***II.8.2. Android Application For PMI With Synchronization Using XML-RPC web Service At PMI DIY***

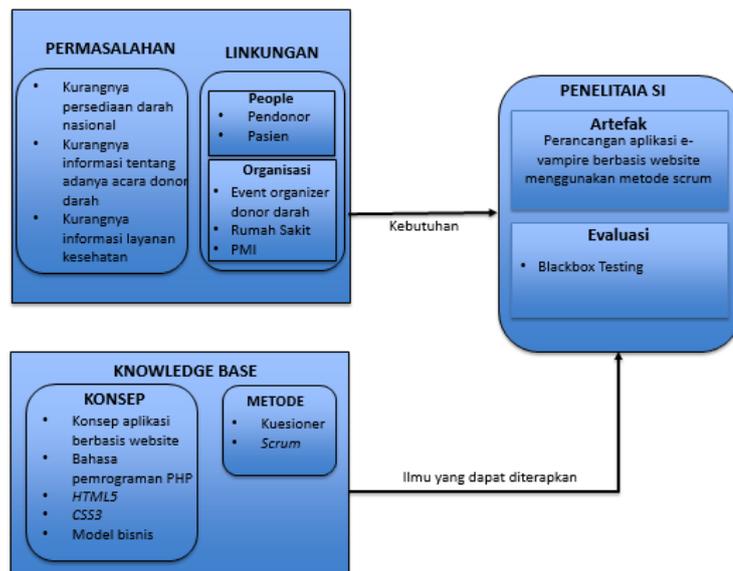
(Sumber: Ikhsanudin, R. (2013). *ANDROID APPLICATION FOR PMI WITH SYNCHRONIZATION USING XML-RPC WEB SERVICE AT PMI DIY* . Yogyakarta: AMIKOM Yogyakarta.)

Aplikasi yang dikembangkan adalah aplikasi Androidbased. Android open source dan telah banyak beredar di masyarakat. Aplikasi ini akan dapat untuk menampilkan informasi tentang PMI. Kami akan bisa mendapatkan informasi tentang suplai darah, berita penting, lokasi transfusidarah dan informasi lainnya tentang PMI.

## BAB III METODOLOGI PENELITIAN

### III.1 Model Konseptual

Model konseptual sangat erat hubungannya dengan teori literatur yang digunakan. Dengan menggunakan model konseptual dapat membantu dalam penataan masalah, mengidentifikasi faktor – faktor relevan dan kemudian memberikan koneksi yang membuatnya lebih mudah untuk memetakan bingkai masalahnya. Jika dipetakan dengan benar maka model konseptual dapat menjadi representasi yang benar dari masalah yang sedang dipelajari dalam penelitian. Model konseptual dibangun berdasarkan teori atau pengertian teoritis. Dengan adanya teori dapat memberitahukan kita dimana harus mencari, apa yang harus dicari, dan bagaimana melihat suatu masalah. (Jonker, dkk, 2011)



Gambar III 1 Metode Konseptual Penelitian

Berdasarkan gambar III.1 dapat dilihat bahwa model konseptual terdiri dari tiga bagian yaitu permasalahan dan lingkungan, *knowledge base* dan penelitian.

Dalam bagian metode konseptual merupakan faktor yang mempengaruhi dalam pengembangan aplikasi *E-Vampire* yaitu permasalahan dan lingkungan, pada bagian ini dibahas mengenai keadaan eksisting dalam lingkup permasalahan yang dibahas dalam aplikasi *E-Vampire*. Permasalahan pada model konseptual yaitu kurangnya persediaan darah nasional, kurangnya informasi tentang adanya event donor darah, dan kurangnya informasi layanan kesehatan maka dalam

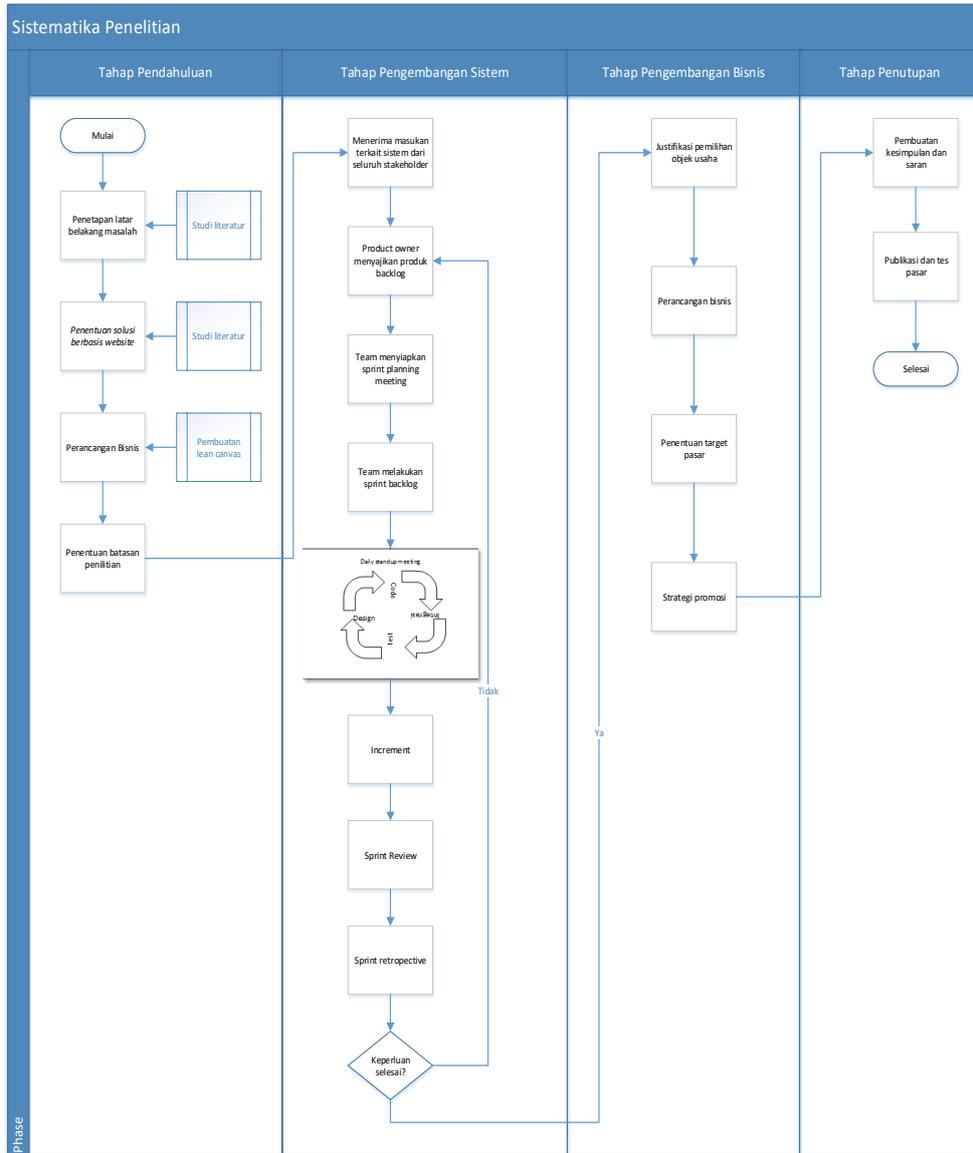
pengembangan aplikasi E-Vampire diharap dapat menangani permasalahan yang terjadi pada pelayanan kesehatan. Lingkungan dalam metode konseptual terbagi menjadi dua yaitu *people* dan organisasi. *People* dan organisasi ini merupakan aktor yang berperan penting dalam pengembangan aplikasi E-Vampire yaitu menjadi user yang akan menggunakan aplikasi tersebut.

Penelitian sistem informasi, dalam penelitian sistem informasi terdapat bagian – bagian yaitu artifak dan evaluasi. Artifak disini yaitu merupakan hasil dari penelitian terhadap aplikasi E-Vampire. Evaluasi merupakan bagian dalam metode konseptual yang berisi bagaimana jalan aplikasi E-Vampire setelah pembuatan dengan *Black-box testing*. *Black-box testing* ini di peruntukan untuk mengetes jalanya aplikasi E-Vampire.

*Knowledge base* pada metode konseptual terdiri dari konsep dan metode. Metode dalam penelitian ini yaitu kuesioner dan scrum dimana metode tersebut memberikan data dan alur pada pengembangan aplikasi E-Vampire. Data yang didapat secara metode kuesioner diharap dapat memenuhi data dan permintaan aplikasi yang diharap dalam pengembangan aplikasi E-Vampire. Alur – alur pengerjaan dalam pengembangan aplikasi E-Vampire diharap dapat sesuai dengan bagian – bagian yang ada pada metode *scrum*.

Lebih di jelaskan lagi

### III.2 Sistematika Penelitian



Gambar III.2 Sistematika Penelitian

#### III.2.1. Tahap Pendahuluan

Pada tahap pendahuluan dilakukan penetapan latar belakang masalah yang dilakukan dengan cara studi literatur yang menjadi dasar dari penelitian ini. Lalu dilakukan penetapan solusi yang dilakukan juga dengan cara studi literatur dari data yang telah tersedia sebelumnya. Setelah latar belakang masalah dan solusi ditemukan, maka ada perencanaan bisnis ke depan menggunakan *Lean Canvas Model*. Setelah terbentuknya perencanaan bisnis, di batasilah penyelesaian masalah dalam penelitian ini.

#### III.2.2. Tahap Pengembangan Produk

Dalam tahap pengembangan sistem, penulis menggunakan kerangka kerja *scrum*. Seperti yang dijelaskan oleh Jeff Sutherland, Ph.D. tentang kerangka kerja *scrum* yang diawali dari masukan terkait sistem dari *customer*, *team*, dan *manager* menuju *product owner* yang akan dijadikan *product backlog* hingga menjadi *sprint planning meeting* hingga terbentuknya *task breakdown* dalam *sprint backlog* yang akan mengawali kegiatan *sprint* berupa *design*, *code*, *integrated*, *test* yang akan diawasi oleh *scrum master* setiap harinya pada *daily standup meeting*. Setelah *sprint backlog* dirasa cukup, tahap selanjutnya akan dilakukan *increment* yang merupakan hasil dari seluruh hal dalam *product backlog* yang telah selesai dikerjakan pada seluruh *Sprint*. *Product backlog* akan di *review* serta melewati *sprint retrospective*.

### **III.2.3. Tahap Pengembangan Bisnis**

#### **III.2.4.1. Justifikasi Pemilihan Objek Usaha**

Pemilihan objek usaha ditentukan oleh solusi dari masalah yang telah penulis sampaikan pada latar belakang. Solusi dari masalah stok darah tersebut merupakan aplikasi berbasis *web* dan *mobile application* yang penulis beri nama “E-Vampire”. Aplikasi E-Vampire berbasis *web* dan *mobile application* yang mengintegrasikan seluruh pendistribusian darah di Bandung terhadap bank darah rumah sakit, PMI dan pihak terkait lainnya. *Save living app* ini akan membantu banyak membantu bagi pendonor, penerima darah serta *event organizer* donor darah. E-Vampire juga merupakan aplikasi yang membantu penerima menuju tempat darah yang telah tersedia dan pendonor yang akan mendonorkan darahnya untuk pasien.

#### **III.2.4.2. Rencana Bisnis Selanjutnya**

Perencanaan pengembangan bisnis selanjutnya akan dilakukan setelah produk di pasarkan. Pengembangan bisnis selanjutnya akan dilakukan setelah kurang lebih satu tahun melihat tanggapan dari pengguna setelah menggunakan aplikasi ini. Beberapa penambahan fitur dan penambahan cakupan wilayah akan dilakukan untuk mendukung pertumbuhan *startup* aplikasi E-Vampire. Berikut perencanaan bisnis selanjutnya pada *startup* aplikasi E-Vampire:

**Tabel III.2.4.2. 1 Rencana Bisnis Selanjutnya**

Tahun	Rencana Pengembangan Pasar	Rencana Pengembangan Produk
Tahun 1	<ol style="list-style-type: none"> <li>1. Pembuatan CV</li> <li>2. Cakupan aplikasi untuk kota Bandung</li> <li>3. Menjalin kerja sama</li> <li>4. Pemenuhan target PMI sebesar 15% jumlah kekurangan darah</li> </ol>	Pembuatan Aplikasi.
Tahun 2	<ol style="list-style-type: none"> <li>1. Pengembangan perusahaan menjadi PT</li> <li>2. Pengembangan cakupan aplikasi untuk Bandung Raya dan Kota Bogor</li> <li>3. Pemenuhan target PMI sebesar 25% jumlah kekurangan darah</li> </ol>	<ol style="list-style-type: none"> <li>1. Pembuatan fitur “<i>Your Mate</i>”</li> <li>2. Pembuatan fitur “<i>Reward</i>”</li> <li>3. Pembuatan fitur “<i>Vam Delivery</i>”</li> </ol>
Tahun 3	<ol style="list-style-type: none"> <li>1. Pengembangan cakupan aplikasi untuk provinsi Jawa Barat</li> <li>2. Pemenuhan target PMI sebesar 35% jumlah kekurangan darah</li> </ol>	Pengembangan aplikasi berbasis iOS.

### III.2.4.3. Target Pasar

Ketersediaan darah dan kebutuhan darah di Bandung masih tidak berimbang. Ketersediaannya masih jauh dari kata ideal untuk setiap daerah yang membutuhkan. Target pasar dari *startup* E-Vampire ialah pendonor efektif (umur 17 – 40 tahun), pendonor tidak efektif (umur > 40 tahun) dan *event organizer*.

### III.2.4.4. Strategi Promosi

Promosi penjualan melalui komunikasi pemasaran terintegrasi melalui periklanan, hubungan masyarakat, personal maupun secara langsung akan menjadi strategi promosi E-Vampire.

1. Periklanan (Media Online) : *website, social media* (Facebook, Instagram dan Youtube).
2. Hubungan Masyarakat : *workshop, seminar, partnership* dengan *event* donor darah.
3. Penjualan langsung personal : media cetak, *talk-to-talk*, dll.

#### **III.2.4. Tahap Penutup**

Pada tahap penutup kegiatan yang dilakukan adalah penarikan kesimpulan mengenai hasil penelitian yang telah dilakukan dan memberikan saran untuk penelitian selanjutnya, serta melakukan publikasi hasil penelitian.

## BAB IV ANALISIS DAN PERANCANGAN

### IV.1. Analisis dan Perancangan Bisnis

#### IV.1.1. Analisis Kelayakan Bisnis

Analisis kelayakan bisnis membahas tentang *Return on Investment* (ROI) yaitu merupakan salah satu bentuk dari rasio profitabilitas yang dimaksudkan untuk mengukur kemampuan perusahaan dengan keseluruhan dana yang ditanamkan dalam aktiva yang digunakan untuk operasi perusahaan untuk menghasilkan keuntungan (Munawir, 2007). Dalam analisis ini merupakan tahapan untuk menilai ROI *startup* aplikasi E-Vampire untuk tahun pertama. Pendiri E-Vampire menanamkan dan menjalin kerja sama untuk mendapatkan modal investasi sebesar Rp. 50.000.000,00 sebagai modal awal. Investasi tersebut digunakan untuk pembuatan dan pengembangan *startup* E-Vampire sebagai berikut:

*Tabel IV.1.1. 1 Estimasi Biaya Operasional*

	Tahun		
	1	2	3
Internet	IDR 3,600,00	IDR 3,600,000	IDR 3,600,000
Domain	IDR 500,000	IDR 500,000	IDR 500,000
Biaya Pembuatan CV	IDR 5,000,000	-	-
Biaya Pembuatan PT	-	IDR 8,000,000	-
Biaya Kendaraan	IDR 5,000,000	IDR 10,000,000	IDR 15,000,000
Biaya Sewa Tempat	IDR 5,000,000	IDR 8,000,000	IDR 8,000,000
Biaya Peralatan Kantor	IDR 5,000,000	IDR 5,000,000	IDR 5,000,000
Biaya Gaji Pegawai	-	IDR 72,000,000	IDR 72,000,000
Total	IDR 24,100,000	IDR 107,100,000	IDR 104,100,000

Untuk mengetahui lebih lanjut mengenai estimasi biaya operasional dapat dilihat pada Lampiran A pada Tabel A.2 Proyeksi Pengeluaran.

**Tabel IV.1.1. 2 Revenue Stream**

Item	Tahun		
	1	2	3
Iklan	RP 14,400,000	RP 21,600,000	RP 28,800,000
Adsense	RP 31,852,800	RP 78,624,000	RP 124,185,600
Deliv Darah		RP 32,400,000	RP 108,000,000
Total	RP 46,252,800	RP 132,624,000	RP 260,985,600

Keterangan :

1. Untuk mengetahui lebih lanjut mengenai *Revenue Stream* dapat dilihat pada Lampiran A: Tabel A.1.1 Proyeksi Pendapatan Iklan Web Apps, A.1.2 Proyeksi Pendapatan Admob Mobile Apps, A.1.3 Proyeksi Pendapatan Google Adsense Web Apps, A.1.4 Proyeksi Pendapatan YouTube, dan A.1.5 Proyeksi Pendapatan *Delivery* Darah
2. Jumlah pendapatan Adsense merupakan jumlah penambahan dari hasil tabel A.1.2 Proyeksi Pendapatan Admob Mobile Apps, A.1.3 Proyeksi Pendapatan Google Adsense Web Apps dan A.1.4 Proyeksi Pendapatan YouTube per tahun

**Tabel IV.1.1. 3 Kalkulasi Laba Rugi**

	Tahun		
	1	2	3
Pendapatan	RP 46,252,800	RP 132,624,000	RP 260,985,600
Total Pendapatan	RP 46,252,800	RP 132,624,000	RP 260,985,600
Biaya Operasi	RP 24,100,000	RP 107,100,000	RP 104,100,000
Total Biaya Operasi	RP 24,100,000	RP 107,100,000	RP 104,100,000
Profit Sebelum Pajak	RP 22,152,800	RP 25,524,000	RP 156,885,600
Pajak 2%	RP 443,056	RP 510,480	RP 3,137,712

**Tabel IV.1.1. 4 Cash flow**

	Tahun		
	1	2	3
Pendapatan	RP 46,252,800	RP 132,624,000	RP 260,985,600
Total Pendapatan	RP 46,252,800	RP 132,624,000	RP 260,985,600
Biaya Operasi	RP 24,100,000	RP 107,100,000	RP 104,100,000
Total Biaya Operasi	RP 24,100,000	RP 107,100,000	RP 104,100,000
Net Inflow	RP 21,709,744	RP 25,013,520	RP 153,747,888
Saldo Kas Awal		RP 21,709,744	RP 46,723,264
Saldo Kas Akhir	RP 21,709,744	RP 46,723,264	RP 200,471,152

**Tabel IV.1.1. 5 Cash-Benefit**

	Tahun		
	1	2	3
Pendapatan	RP 46,252,800	RP 132,624,000	RP 260,985,600
Biaya Operasi	RP 24,100,000	RP 107,100,000	RP 104,100,000
Total	RP 22,152,800	RP 25,524,000	RP 156,885,600

Berikut tabel hasil analisis ROI:

**Tabel IV.1.1. 6 Analisis ROI**

Perhitungan ROI	Nilai
Total Investasi	Rp. 50.000.000,00
Total Pendapatan	Rp. 24.100.000,00
ROI (%)	93

Dari hasil analisis tabel di atas, menunjukkan bahwa *startup* aplikasi E-Vampire dapat menghasilkan keuntungan yang akan digunakan untuk menutupi investasi yang telah dikeluarkan sebesar 93 %. ROI digunakan oleh sebuah perusahaan untuk efisiensi penggunaan modal yang bekerja, produksi dan penjualan. Dari hasil perhitungan proyeksi finansial untuk tahun pertama *startup* aplikasi E-Vampire, maka dapat diambil kesimpulan dari indikator finansial sebagai berikut:

*Tabel IV.1.1. 7 Keputusan Kelayakan*

Parameter	Hasil Perhitungan	Kriteria Kelayakan	Keputusan Kelayakan
ROI	60 %	ROI > 2 %	Layak

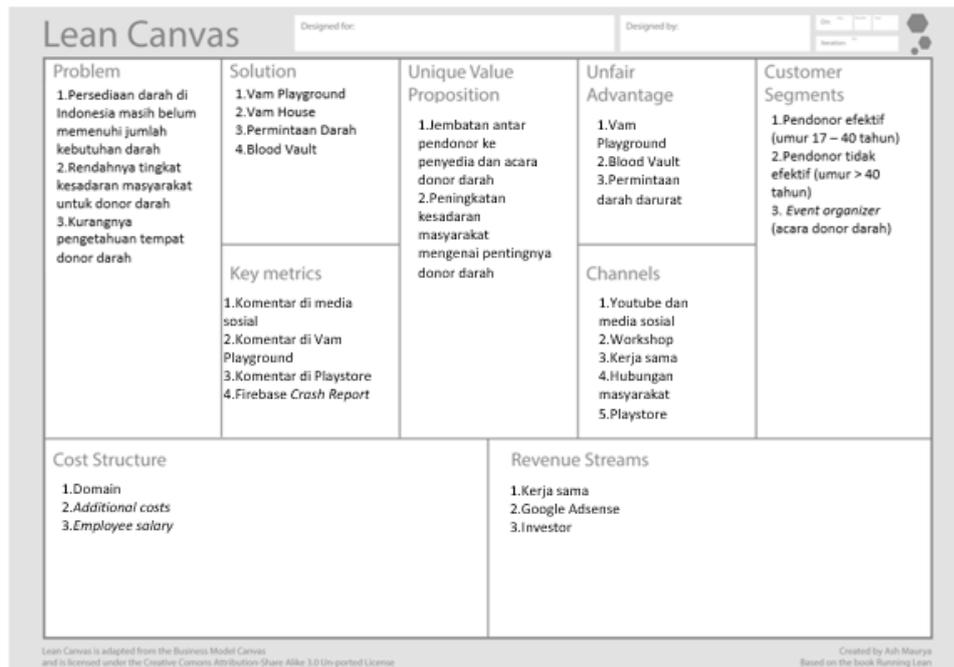
Dari hasil keputusan kelayakan di atas, dapat diambil kesimpulan bahwa secara finansial produk aplikasi E-Vampire **layak** untuk dikembangkan.

#### **IV.1.2. Analisis Potensi Pasar**

Berdasarkan Badan Pusat Statistik Kota Bandung, jumlah penduduk Bandung periode bulan April tahun 2016 sebesar 2.483.977 jiwa. Idealnya kantong darah membutuhkan 2,5 % dari jumlah penduduk yang ada, maka idealnya dibutuhkan darah sebanyak 62.100 kantong. Bila diambil asumsi setiap pendonor akan menghasilkan maksimal 4 kantong darah, maka dibutuhkan 15.525 pendonor. *Startup* aplikasi E-Vampire menargetkan membantu PMI untuk mendapatkan pendonor untuk tahun pertama sebanyak 20 % yaitu sebanyak 3104 pendonor. Untuk memenuhi target tersebut, *startup* aplikasi E-Vampire berencana untuk melakukan kerja sama dengan PMI Kota Bandung, dan beberapa perusahaan, komunitas, universitas, hingga organisasi masyarakat.

#### **IV.1.3. Analisis Bisnis Model**

Bisnis model merupakan alat bantu yang menjelaskan bagaimana suatu organisasi atau perusahaan menciptakan, memberikan, dan menangkap suatu nilai tambah. *Startup* aplikasi E-Vampire menggunakan bisnis model *Lean Canvas* sebagai acuan. Gambar *Lean Canvas* dapat dilihat sebagai berikut:



Pada Gambar IV.1.3 merupakan gambaran model bisnis pada *startup* E-Vampire yang terbagi menjadi sembilan komponen, yaitu:

1. *Problem dan existing alternatives*. Suatu masalah merupakan hal yang paling penting dan menjadi dasar dari pembuatan suatu aplikasi. Pada perancangan aplikasi E-Vampire mendapat tiga masalah utama yang menjadi dasar adanya aplikasi ini. Tiga masalah tersebut adalah :
  1. Persediaan darah di kota Bandung masih belum memenuhi jumlah kebutuhan darah
  2. Rendahnya tingkat kesadaran (*awareness*) masyarakat untuk donor darah
  3. Kurangnya pengetahuan tempat donor darah
2. *Customer segments*. Pelanggan merupakan target utama yang diinginkan perusahaan untuk mengembangkan bisnis, untuk itu dibutuhkan identifikasi *customer segments* agar tepat sasaran. Pada *startup* E-Vampire memiliki tiga jenis target customer yang akan dituju, yaitu pendonor efektif , pendonor tidak efektif, dan *event oragnizer*.
3. *Unique Value Proposition*. Dalam pengembangan suatu aplikasi pada startup perlunya diferensiasi dengan *startup* yang memiliki maksud yang sama maka ada dua pembeda pada startup E-Vampire yaitu menjadi jembatan antara pendonor ke penyedia dan acara donor darah dan meningkatkan kesadaran.

4. *Solution*. Aplikasi E-Vampire memiliki fitur yang dapat menangani masalah yang merupakan dasar dari pembuatan aplikasi ini. Fitur-fitur yang dimaksud yaitu Vam Playground, Vam House, Permintaan Darah, Blood Vault.
5. *Channel*. Merupakan tempat dimana *startup* dapat memasarkan dan menyampaikan informasi tentang aplikasi dimana startup E-Vampire memiliki kanal-kanal sebagai berikut, Youtube, media sosial, kerja sama dan masyarakat.
6. *Revenue Stream*. merupakan penghasilan yang akan diterima dari produk yang dijual yang berhasil ditawarkan pada pelanggan. Penghasilan yang didapatkan oleh *startup* E-Vampire di antaranya adalah dari hasil Kerja sama dan Google AdSense.
7. *Cost Structure*. merupakan biaya yang diperlukan perusahaan untuk menunjang kegiatan pengembangan sistem yang dibagi menjadi *available cost*, *fixed cost* dan *variable cost*.
8. *Key Metrics*. Aktivitas-aktivitas yang di lakukan dalam perancangan aplikasi yaitu proses pembuatan aplikasi, proses pembuatan *startup*, proses pengajuan hak cipta, proses promosi dan proses kerja sama.
9. *Unfair Advantage*. Di sini merupakan tahapan perbandingan aplikasi dengan aplikasi lainnya yang sudah telah dikenal dan dipublikasikan yaitu aplikasi E-Vampire memiliki fitur seperti berikut permintaan darah darurat dan Blood Vault.

#### **IV.1.4. Analisis Kompetitor**

Aplikasi E-Vampire telah memiliki pesaing dengan meluncurkan aplikasi berbasis dan bertema yang sama yaitu Reblood oleh Start Surabaya dan Easy Blood oleh PT. Telekomunikasi Indonesia, TBK. Berikut hasil analisis kompetitor dari pesaing E-Vampire:

**Tabel IV.1.4. 1 Analisis Kompetitor**

No.	Fitur Yang Dibangun	Aplikasi Pendororan Darah		Aplikasi Penelitian E-Vampire
		Easy Blood	Reblood	
1	Memiliki Aplikasi Website	X	√	√
2	Memiliki Aplikasi Mobile	√	√	√
3	Informasi Kegiatan Donor Darah	√	√	√
4	Informasi Stok Darah	√	X	√
5	Informasi Bank Darah	X	X	√
6	Informasi Edukatif	X	√	√
7	Informasi Lokasi Rumah Sakit Dan PMI	√	X	√
8	Pemberian Notifikasi	√	X	√
9	Permintaan Butuh Darah Darurat	X	X	√

#### **IV.2. Analisis Perancangan Produk**

Dalam membantu kekurangan persediaan kantung darah tersebut, dibutuhkan solusi untuk menambah kesadaran masyarakat Indonesia atas pentingnya donor darah. Salah satu solusi tersebut ialah E-Vampire. Aplikasi E-Vampire berbasis situs web dan aplikasi *mobile* akan memudahkan penggunaanya mengakses aplikasi. Aplikasi ini akan dipasarkan secara gratis dan memiliki fitur utama seperti Vam Playground, permintaan darah darurat dan lain-lain. Aplikasi E-Vampire ini nanti diharapkan dapat digunakan oleh seluruh pendonor efektif maupun non efektif.

##### **IV.2.1. Sprint Planing**

Dalam pembuatan aplikasi E-Vampire berbasis *website* ini, menggunakan metode Scrum. Pada metode Scrum terdapat salah satu proses yaitu *sprint planning*. Selama *sprint planning*, tim mengambil potongan kecil item dari *backlog* produk yaitu *sprint backlog*, dan memutuskan cara menerapkan potongan tersebut. Berikut daftar *backlog* produk pada aplikasi E-Vampire berbasis *website* :

**Tabel IV.2.1. 1 Sprint Planing**

<i>User Story ID</i>	Saya ingin/butuh/dapat untuk	<i>Task</i>	<i>Complexity (1-5)</i>
US-01	Saya dapat <i>login</i>	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	3
US-02	Saya dapat melakukan registrasi	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-03	Saya dapat mengembalikan kata sandi	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-04	Saya dapat melihat tampilan Home	Desain <i>mockup</i>	2
		Desain konten	3
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-05	Saya ingin melihat daftar acara donor darah	Desain <i>mockup</i>	2
		Desain konten	3
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-06	Saya ingin mengikuti acara donor darah dan batal mengikuti acara	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	4
		Implementasi <i>back-end code</i>	5
		<i>Testing</i>	2
US-07	Saya ingin melihat informasi persediaan darah	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4

<i>User Story ID</i>	<i>Saya ingin/butuh/dapat untuk</i>	<i>Task</i>	<i>Complexity (1-5)</i>
	Saya ingin melihat informasi persediaan darah pada bank darah	<i>Testing</i>	2
US-08	Saya dapat membaca informasi edukatif mengenai donor darah dan memberikan komentar	Desain <i>mockup</i>	2
		Desain konten	5
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-09	Saya ingin melihat lokasi rumah sakit dan PMI	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-10	Saya dapat mengubah data akun dan melihat data akun	Desain <i>mockup</i>	1
		Implementasi <i>front-end code</i>	1
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	3
US-11	Saya ingin melakukan permintaan kebutuhan darah darurat	Desain <i>mockup</i>	1
		Implementasi <i>front-end code</i>	2
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-12	Saya ingin menghubungi pihak E-Vampire	Desain <i>mockup</i>	1
		Implementasi <i>front-end code</i>	2
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-13	Saya dapat <i>logout</i>	Implementasi <i>back-end code</i>	1
		<i>Testing</i>	1
US-14	Saya dapat melihat informasi tentang aplikasi E-Vampire	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		<i>Testing</i>	2

Dalam pengerjaan *backlog* produk aplikasi berbasis *website* di atas, akan dibagi dalam empat tahapan *sprint* sebagai berikut:

**Tabel IV.2.1. 2 Product Backlog**

	<i>Product Backlog</i>
<i>Sprint 1</i>	Login, Registrasi, Logout, dan Lupa Password
<i>Sprint 2</i>	Home, About, Contact, dan Vam House
<i>Sprint 3</i>	Vam Playground, Komentar, Permintaan Darah, dan Blood Vault
<i>Sprint 4</i>	Event, Ikut Event, Lihat Profil, Ubah Profil

#### **IV.2.2. Kebutuhan Bisnis**

Pada tahun 2016 , pengguna internet di Indonesia sudah mencapai lebih dari 60 juta. Mereka menggunakan internet untuk menemukan informasi yang sedang dibutuhkan. Dengan adanya aplikasi berbasis *website* maka akan meningkatkan visibilitas atau keberadaan perusahaan itu sendiri. Dengan adanya aplikasi berbasis *website* kemudahan dalam penyampaian informasi jauh lebih mudah dari pada harus menyebarkan melalui TV , radio , atau media massa. Pada era digital *website* menjadi salah satu *trend* di era digital ini. Maka dengan demikian peluang yang tercipta cukup membantu perusahaan untuk memasarkan produk tanpa ada beban dalam memasarkan produk. Untuk menangkap peluang tersebut, *startup* aplikasi E-Vampire mengeluarkan aplikasi berbasis *website*, perusahaan dapat memanfaatkan kesempatan tersebut untuk menciptakan saluran pemasaran langsung.

#### **IV.2.3. Spesifikasi Profil Aktor**

Dalam pembuatan aplikasi E-Vampire berbasis *website*, dapat dipisahkan menjadi dua jenis aktor yang terlibat. Tiga jenis aktor tersebut memiliki peran yang berbeda seperti yang dijelaskan pada tabel berikut ini:

*Tabel IV.2.3. 1 Spesifikasi Profil Aktor*

No.	Nama Aktor	Deskripsi
1	Pengguna Terdaftar	Aktor yang memiliki hak akses untuk fitur yang memiliki transaksi.
2	Pengguna Tidak Terdaftar	Aktor yang tidak memiliki hak akses untuk fitur yang memiliki transaksi.
3	Admin	Aktor yang memiliki hak akses untuk mengatur fitur – fitur pada aplikasi E-Vampire.



#### IV.2.5. Usecase Form

*Usecase form* dibuat berdasarkan *usecase* diagram yang telah dibuat sebelumnya. Pembuatan *usecase form* dibutuhkan untuk menjelaskan tentang aktivitas apa saja yang dilakukan oleh aktor pada setiap *usecase* yang ada.

Berikut *usecase form* :

##### 1. *Usecase form Login*

**Tabel.2.5. 1 Usecase Form Login**

<b>Nama</b>	<b><i>Login</i></b>
<b>Aktor</b>	Pengguna , pasien dan admin
<b>Deskripsi</b>	Proses untuk mengakses aplikasi dengan memasukkan identitas dari akun pengguna dan kata sandi untuk mendapatkan hak akses menggunakan aplikasi
<b>Prasyarat</b>	1. Aplikasi terhubung dengan internet 2. Pengguna telah terdaftar
<b>Kondisi Akhir</b>	Pengguna dan pasien mendapat hak izin masuk aplikasi , masuk ke menu <i>Home</i> dan menu <i>login</i> berubah menjadi menu Profil
<b>Alur Normal</b>	1. Pengguna membuka aplikasi 2. Pengguna mengisi email dan kata sandi 3. Pengguna menekan tombol login 4. Pengguna masuk ke aplikasi
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	1. Pengguna mengisi email dan kata sandi yang salah 2. Pengguna belum terdaftar 3. Aplikasi tidak terhubung dengan internet
<b>Mockup</b>	MA-07

## 2. Usecase Form Registrasi

*Tabel.2.5. 2 Usecase Form Registrasi*

<b>Nama</b>	<b>Registrasi</b>
<b>Aktor</b>	Pengguna dan pasien
<b>Deskripsi</b>	Proses untuk mendaftarkan pengguna dengan memasukkan informasi pribadi untuk mendapatkan hak akses menggunakan aplikasi
<b>Prasyarat</b>	Aplikasi terhubung dengan internet
<b>Kondisi Akhir</b>	Pengguna dan pasien telah terdaftar untuk mendapatkan hak akses aplikasi dan masuk ke menu <i>Home</i>
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna membuka aplikasi</li> <li>2. Pengguna menekan menu “Registrasi”</li> <li>3. Pengguna memasukkan data sesuai dengan form registrasi</li> <li>4. Pengguna menekan tombol “Daftar”</li> <li>5. Pengguna masuk ke aplikasi pada halaman Home</li> </ol>
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan email yang telah terdaftar</li> <li>2. Aplikasi tidak terhubung dengan internet</li> </ol>
<b>Mockup</b>	MA-06

## 3. Usecase form Lupa Password

*Tabel.2.5. 3 Usecase Form Reset Password*

<b>Nama</b>	<b>Lupa Password</b>
<b>Aktor</b>	Pengguna dan pasien
<b>Deskripsi</b>	Proses pemulihan atau menyetel ulang kata sandi
<b>Prasyarat</b>	<ol style="list-style-type: none"> <li>1. Pengguna telah terdaftar pada aplikasi</li> <li>2. Aplikasi terhubung dengan internet</li> </ol>
<b>Kondisi Akhir</b>	Pengguna dan pasien telah memulihkan atau menyetel ulang kata sandi
<b>Alur Normal</b>	-
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan email yang salah</li> <li>2. Aplikasi tidak terhubung dengan internet</li> </ol>
<b>Mockup</b>	MA-08

#### 4. Usecase form lihat Home

**Tabel.2.5. 4 Usecase Form Lihat Home**

<b>Nama</b>	<i>Lihat Home</i>
<b>Aktor</b>	Pengguna , admin dan pasien
<b>Deskripsi</b>	-
<b>Prasyarat</b>	Aplikasi terhubung dengan internet
<b>Kondisi Akhir</b>	Tertampilkan Home
<b>Alur Normal</b>	1. Pengguna menekan tombol menu “Home” 2. Aplikasi menampilkan halaman <i>Home</i>
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	Aplikasi tidak terhubung dengan internet
<b>Mockup</b>	MA-01

#### 5. Usecase form lihat Event

**Tabel.2.5. 5 Usecase Form Lihat Event**

<b>Nama</b>	<i>Lihat Event</i>
<b>Aktor</b>	Pengguna dan admin
<b>Deskripsi</b>	Proses menampilkan acara – acara donor darah yang akan diselenggarakan
<b>Prasyarat</b>	1. Pengguna telah <i>Login</i> pada aplikasi 2. Aplikasi terhubung dengan internet
<b>Kondisi Akhir</b>	Tertampilkan acara – acara donor darah yang akan diselenggarakan
<b>Alur Normal</b>	1. Pengguna menekan tombol menu “Event” 2. Aplikasi mengirim permintaan data menggunakan Controller 3. Aplikasi menerima data dari View 4. Aplikasi menampilkan acara – acara yang akan diselenggarakan
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	Aplikasi tidak terhubung dengan internet
<b>Mockup</b>	MA-02

## 6. Usecase form ikut Event

**Tabel.2.5. 6 Usecase Form Ikut Event**

<b>Nama</b>	<b><i>Ikut Event</i></b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang ingin ikut serta pada acara
<b>Prasyarat</b>	<ol style="list-style-type: none"> <li>1. Pengguna telah <i>Login</i> pada aplikasi</li> <li>2. Aplikasi terhubung dengan internet</li> </ol>
<b>Kondisi Akhir</b>	Pengguna berhasil
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan menu “Event”</li> <li>2. Pengguna memilih event</li> <li>3. Pengguna mengisi data diri</li> <li>4. Pengguna menekan tombol “follow event”</li> <li>5. Aplikasi mengirim permintaan</li> <li>6. Aplikasi menerima data</li> <li>7. Aplikasi menampilkan status sukses</li> </ol>
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	<ol style="list-style-type: none"> <li>1. Aplikasi tidak terhubung dengan internet</li> <li>2. Pengguna dan admin belum melakukan login</li> </ol>
<b>Mockup</b>	MA-09

## 7. Usecase form Vam Playground

**Tabel.2.5. 7 Usecase Form Vam Playground**

<b>Nama</b>	<b><i>Vam Playground</i></b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang telah atau belum terdaftar pada aplikasi
<b>Prasyarat</b>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu Vam Playground pada aplikasi</li> <li>2. Aplikasi terhubung dengan internet</li> </ol>
<b>Kondisi Akhir</b>	Pengguna berhasil
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan menu “Vam Playground”</li> <li>2. Pengguna memilih konten edukasi</li> <li>3. Aplikasi mengirim permintaan</li> <li>4. Aplikasi menampilkan halaman Vam Playground</li> </ol>
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	<ol style="list-style-type: none"> <li>1. Aplikasi tidak terhubung dengan internet</li> </ol>
<b>Mockup</b>	MA-03

8. *Usecase form* masukan Komentar

*Tabel.2.5. 8 Usecase Form Masukan Komentar*

<b>Nama</b>	<b>Masukan Komentar</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang tertarik memberikan pendapat
<b>Prasyarat</b>	1. Pengguna telah <i>Login</i> pada aplikasi 2. Aplikasi terhubung dengan internet
<b>Kondisi Akhir</b>	Pengguna berhasil
<b>Alur Normal</b>	1. Pengguna menekan menu “Vam Playground” 2. Pengguna memilih konten edukasi 3. Pengguna mengisi komentar atau pendapat 4. Pengguna menekan tombol “kirim” 5. Aplikasi mengirim permintaan 6. Aplikasi menerima data 7. Aplikasi menampilkan komentar pada kolom komentar
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	1. Aplikasi tidak terhubung dengan internet 2. Pengguna dan admin belum melakukan login
<b>Mockup</b>	MA-10

9. *Usecase form* lihat *Blood Vault*

*Tabel.2.5. 9 Usecase Form Lihat Blood Vault*

<b>Nama</b>	lihat <i>Blood Vault</i>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang telah atau belum terdaftar pada aplikasi
<b>Prasyarat</b>	Aplikasi terhubung dengan internet
<b>Kondisi Akhir</b>	Pengguna berhasil
<b>Alur Normal</b>	1. Pengguna menekan menu “Blood Vault” 2. Pengguna memilih jenis kantong pada kolom jenis darah 3. Aplikasi mengirim permintaan 4. Aplikasi menampilkan jumlah kantong darah
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	Aplikasi tidak terhubung dengan internet
<b>Mockup</b>	MA-04

10. Usecase form lihat Vam House

**Tabel.2.5. 10 Usecase Form Lihat Vam House**

<b>Nama</b>	<i>Lihat Vam House</i>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang telah atau belum terdaftar pada aplikasi
<b>Prasyarat</b>	Aplikasi terhubung dengan internet
<b>Kondisi Akhir</b>	Pengguna berhasil
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan menu “Vam House”</li> <li>2. Pengguna memilih <i>marker</i> yang tertera pada map</li> <li>3. Aplikasi mengirim permintaan</li> <li>4. Aplikasi menampilkan informasi sesuai dengan <i>marker</i></li> </ol>
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	Aplikasi tidak terhubung dengan internet
<b>Mockup</b>	MA-05

11. Usecase form lihat profil

**Tabel.2.5. 11 Usecase Form Lihat Profil**

<b>Nama</b>	<i>Lihat Profil</i>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang telah terdaftar dalam aplikasi
<b>Prasyarat</b>	<ol style="list-style-type: none"> <li>1. Pengguna telah <i>Login</i> pada aplikasi</li> <li>2. Aplikasi terhubung dengan internet</li> </ol>
<b>Kondisi Akhir</b>	Pengguna berhasil
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan menu “Profil”</li> <li>2. Aplikasi mengirim permintaan</li> <li>3. Aplikasi menampilkan halaman profil</li> </ol>
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	<ol style="list-style-type: none"> <li>1. Aplikasi tidak terhubung dengan internet</li> <li>2. Pengguna dan admin belum melakukan login</li> </ol>
<b>Mockup</b>	MA-11

## 12. Usecase form ubah profil

**Tabel.2.5. 12 Usecase Form Ubah Profil**

<b>Nama</b>	<b>Ubah Profil</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang telah terdaftar dalam aplikasi
<b>Prasyarat</b>	<ol style="list-style-type: none"> <li>1. Pengguna telah <i>Login</i> pada aplikasi</li> <li>2. Aplikasi terhubung dengan internet</li> </ol>
<b>Kondisi Akhir</b>	Pengguna berhasil
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pengguna menekan menu “Profil”</li> <li>2. Pengguna mengisi data pada kolom ubah profil</li> <li>3. Pengguna menekan tombol “ubah”</li> <li>4. Aplikasi mengirim permintaan</li> <li>5. Aplikasi menerima data</li> <li>6. Aplikasi menampilkan status sukses</li> </ol>
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	<ol style="list-style-type: none"> <li>1. Aplikasi tidak terhubung dengan internet</li> <li>2. Pengguna dan admin belum melakukan login</li> </ol>
<b>Mockup</b>	MA-11

## 13. Usecase form Logout

**Tabel.2.5. 13 Usecase Form Logout**

<b>Nama</b>	<b>Logout</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang telah terdaftar dalam aplikasi
<b>Prasyarat</b>	<ol style="list-style-type: none"> <li>1. Pengguna telah <i>Login</i> pada aplikasi</li> <li>2. Aplikasi terhubung dengan internet</li> </ol>
<b>Kondisi Akhir</b>	Pengguna berhasil Logout
<b>Alur Normal</b>	Pengguna menekan menu “Logout”
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	<ol style="list-style-type: none"> <li>1. Aplikasi tidak terhubung dengan internet</li> <li>2. Pengguna dan admin belum melakukan login</li> </ol>
<b>Mockup</b>	

#### 14. Usecase form Permintaan Darah

*Tabel.2.5. 14 Usecase Form Permintaan Darah*

<b>Nama</b>	<b>Permintaan Darah</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang telah terdaftar dalam aplikasi
<b>Prasyarat</b>	1. Pengguna telah <i>Login</i> pada aplikasi 2. Aplikasi terhubung dengan internet
<b>Kondisi Akhir</b>	Pengguna berhasil mengirim permintaan darah
<b>Alur Normal</b>	1. Pengguna menekan menu “Permintaan Darah” 2. Pengguna mengisi data pada <i>form</i> Permintaan Darah 3. Pengguna menekan tombol “Kirim” 4. Aplikasi mengirim permintaan 5. Aplikasi menerima data 6. Aplikasi menampilkan status sukses
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	1. Aplikasi tidak terhubung dengan internet 2. Pengguna dan admin belum melakukan login
<b>Mockup</b>	

#### 15. Usecase form Lihat Contact

*Tabel.2.5. 15 Usecase Form Lihat Contact*

<b>Nama</b>	<b>Lihat Contact</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang telah terdaftar dalam aplikasi
<b>Prasyarat</b>	Aplikasi terhubung dengan internet
<b>Kondisi Akhir</b>	Pengguna berhasil Melihat Contact
<b>Alur Normal</b>	Pengguna menekan menu “Contact ”
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	Aplikasi tidak terhubung dengan internet
<b>Mockup</b>	

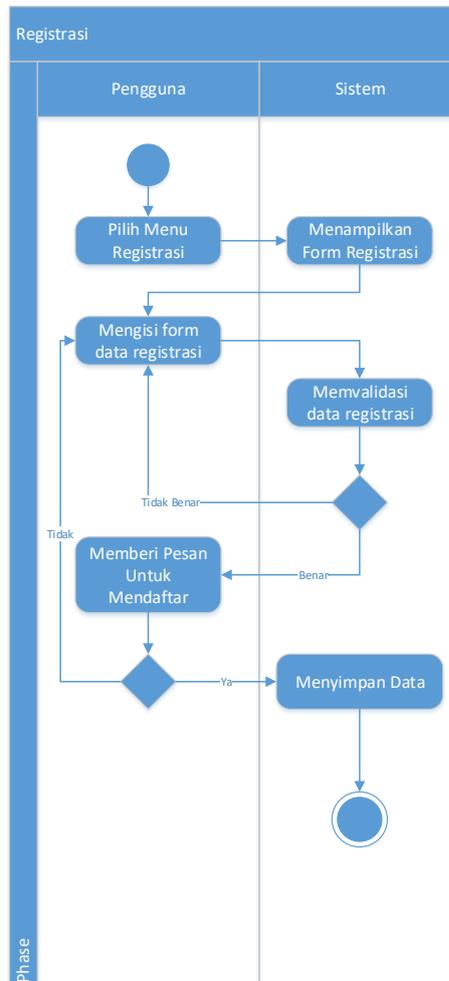
## 16. Usecase form Lihat About

*Tabel.2.5. 16 Usecase Form Lihat About*

<b>Nama</b>	<b>Lihat Contact</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Proses bagi pengguna yang telah terdaftar dalam aplikasi
<b>Prasyarat</b>	Aplikasi terhubung dengan internet
<b>Kondisi Akhir</b>	Pengguna berhasil Melihat Contact
<b>Alur Normal</b>	Pengguna menekan menu “Contact ”
<b>Alur Alternatif</b>	-
<b>Pengecualian</b>	Aplikasi tidak terhubung dengan internet
<b>Mockup</b>	

#### IV.2.6. Activity Diagram

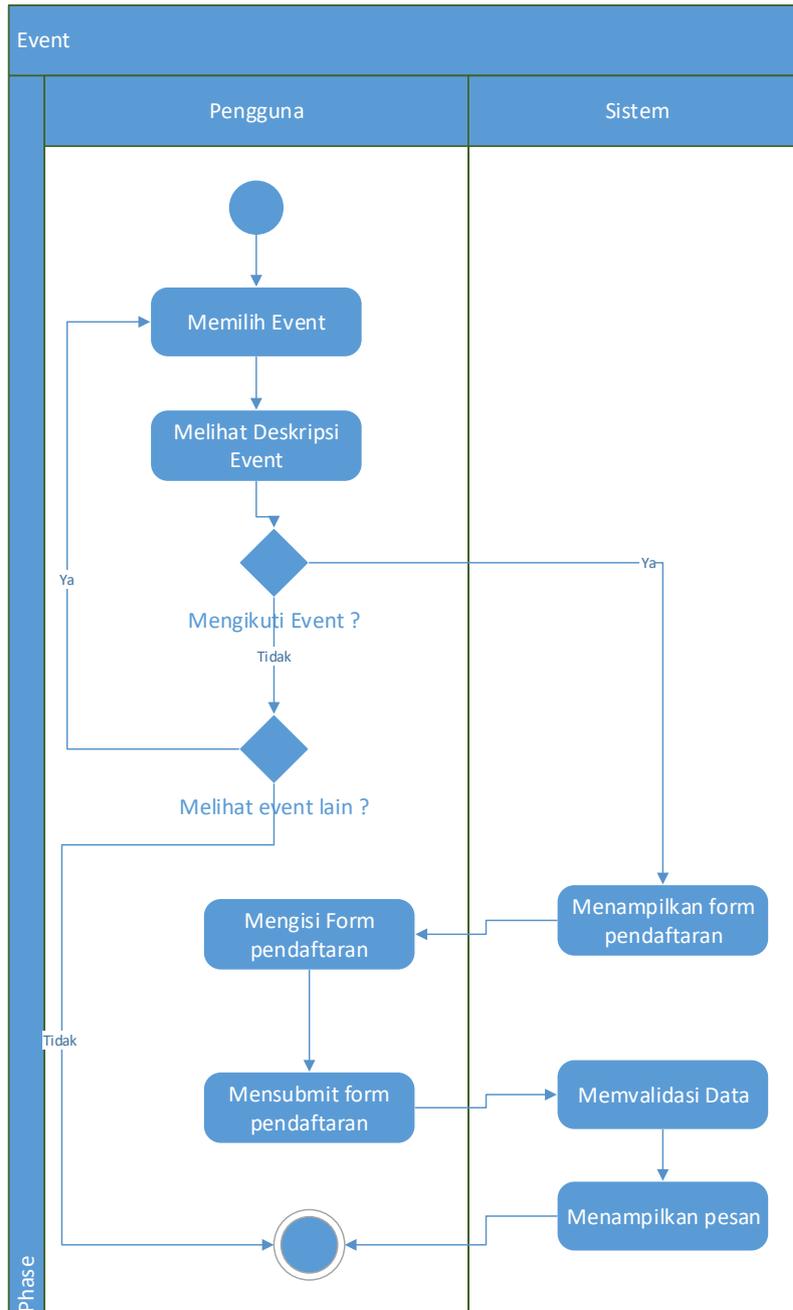
*Activity diagram* adalah sebuah diagram yang menggambarkan berbagai alur aktivitas dalam sistem yang akan dirancang, bagaimana alur berawal, kemungkinan yang akan terjadi, dan bagaimana alur berakhir. *Activity diagram* pada perancangan aplikasi berbasis *website* E-Vampire adalah sebagai berikut:



Gambar IV2.5. 1 Activity Diagram Registrasi

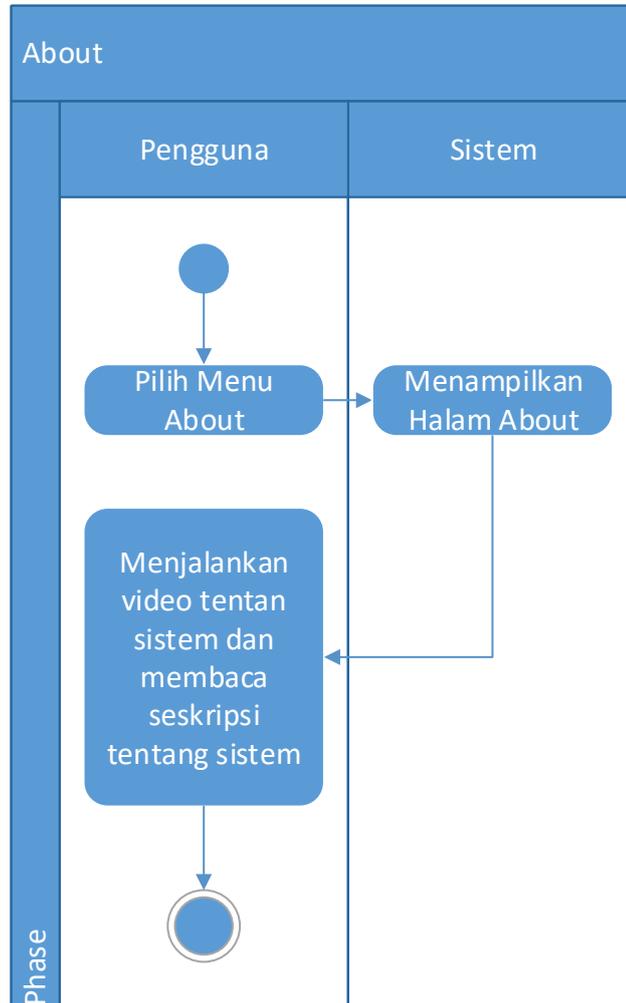
Menggambarkan alur proses registrasi ke dalam aplikasi agar mendapatkan hak akses. Pertama pengguna harus memilih menu registrasi lalu sistem akan memproses permintaan dan pengguna langsung mengisi data registrasi setelah form registrasi muncul. Setelah pengguna telah mengisi maka pengguna menekan tombol registrasi maka sistem mengadakan pengecekan apabila data valid maka akan langsung menampilkan halaman utama.





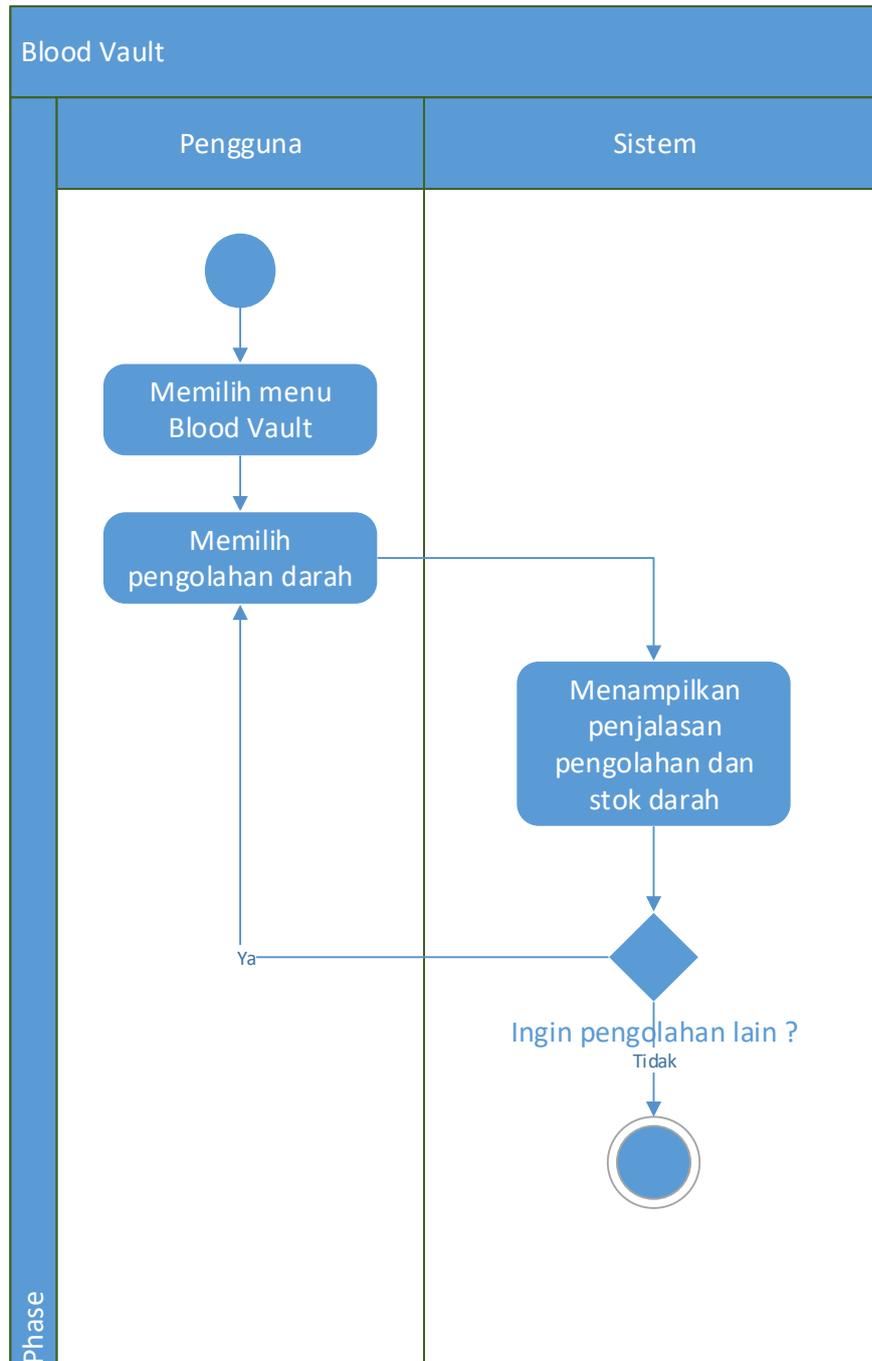
**Gambar IV2.5. 3 Activity Diagram Event**

Menggambarakan proses event diawali pengguna memilih menu event dan sistem akan menampilkan event yang telah tercantum pada aplikasi. Pengguna dapat memilih event yang akan di ikuti oleh pengguna. Setelah pengguna memilih event maka sistem akan membuka halaman event yang berisi deskripsi singkat event tersebut dan pengguna dapat mengikuti event tersebut. Setelah pengguna memilih untuk ikut event maka pengguna menuliskan data diri untuk mengikuti event yang telah dipilih.



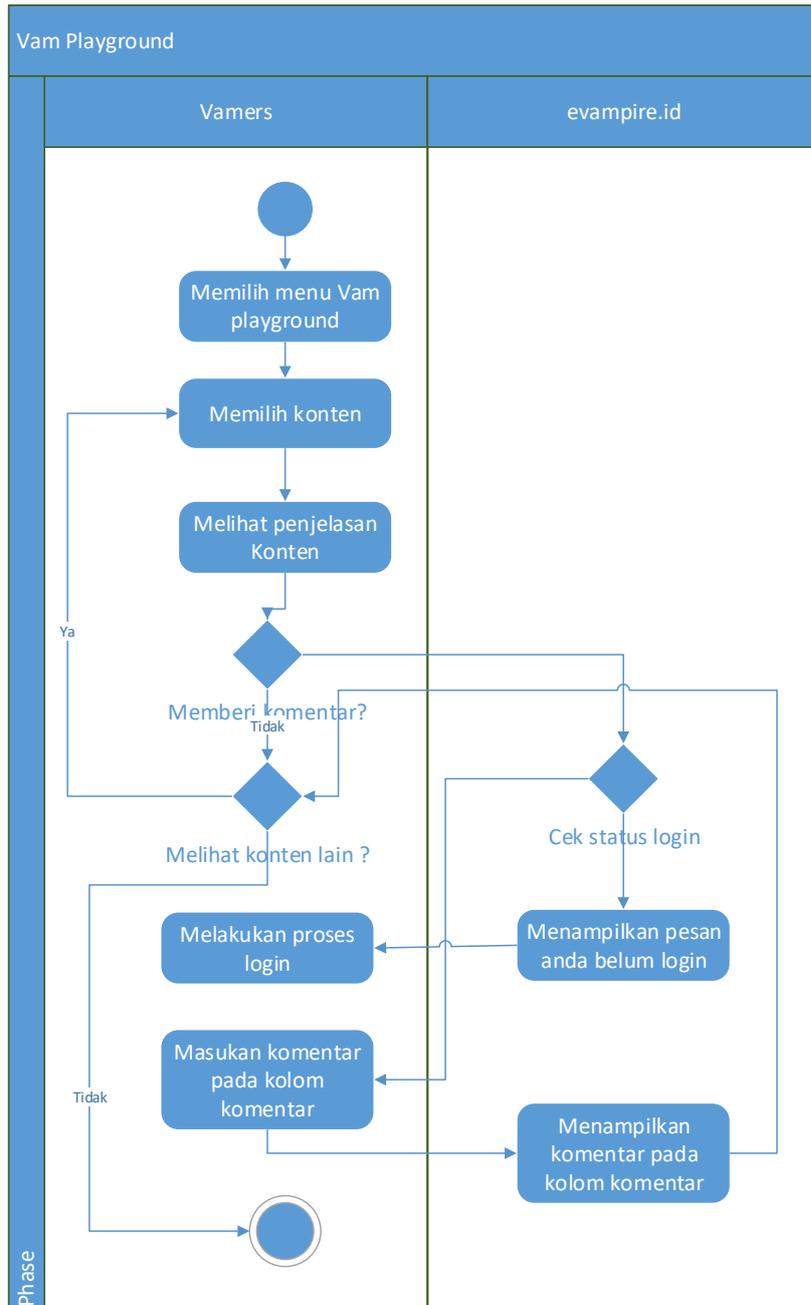
**Gambar IV2.5. 4 Activity Diagram About**

Menggambarkan proses *about* di awali pengguna memilih menu *about* maka sistem menampilkan halaman *about*. Pengguna dapat melihat video dan deskripsi singkat tentang aplikasi berbasis website ini.



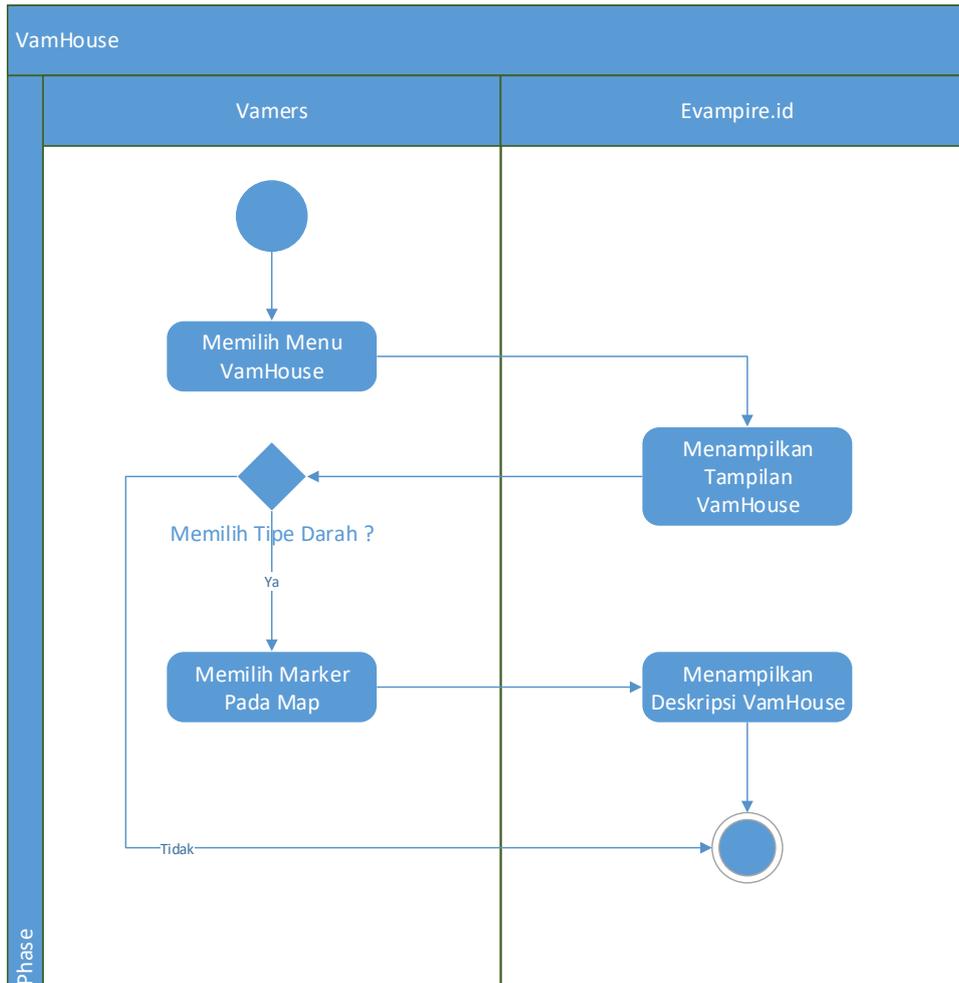
**Gambar IV2.5. 5 Activity Diagram Blood Vault**

Menggambarakan proses Blood Vault di awali pengguna memilih menu Blood Vault maka sistem akan menampilkan informasi tentang stok darah dan sesuai dengan macam-macam kantong darah.



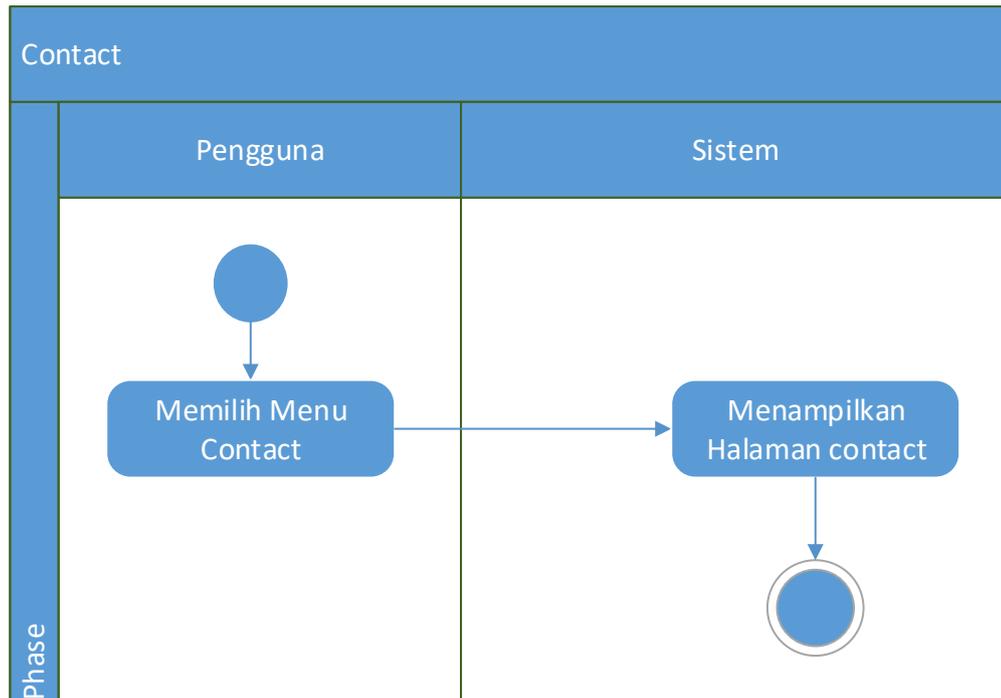
**Gambar IV2.5. 6 Activity Diagram Vam Playground**

Menggambarkan proses Vam Playground diawali dengan pengguna memilih menu Vam Playground maka sistem akan menampilkan halaman Vam Playground. Dalam halaman Vam Playground terdapat informasi yang dapat meningkatkan *awareness* pengguna. Pengguna dapat memilih informasi dan memberikan komentar tentang informasi tersebut.



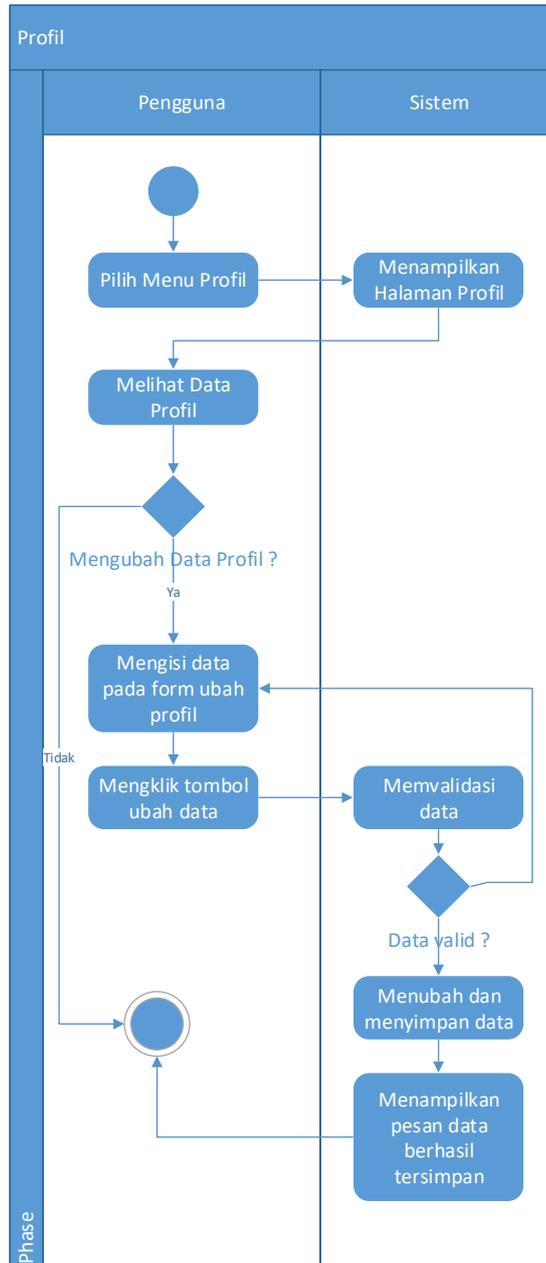
**Gambar IV2.5. 7 Activity Diagram Vam House**

Menggambarkan proses Vam House diawali dengan pengguna memilih menu Vam House maka sistem akan menampilkan halaman Vam House. Dalam halaman Vam House terdapat informasi mengenai alamat penyedia stok darah seperti PMI kota Bandung, rumah sakit. Pengguna memilih *marker* yang terdapat dalam map yang disediakan aplikasi berbasis website ini. Setelah pengguna memilih *marker* maka aplikasi akan memunculkan informasi alamat dan nama tempat penyedia darah tersebut.



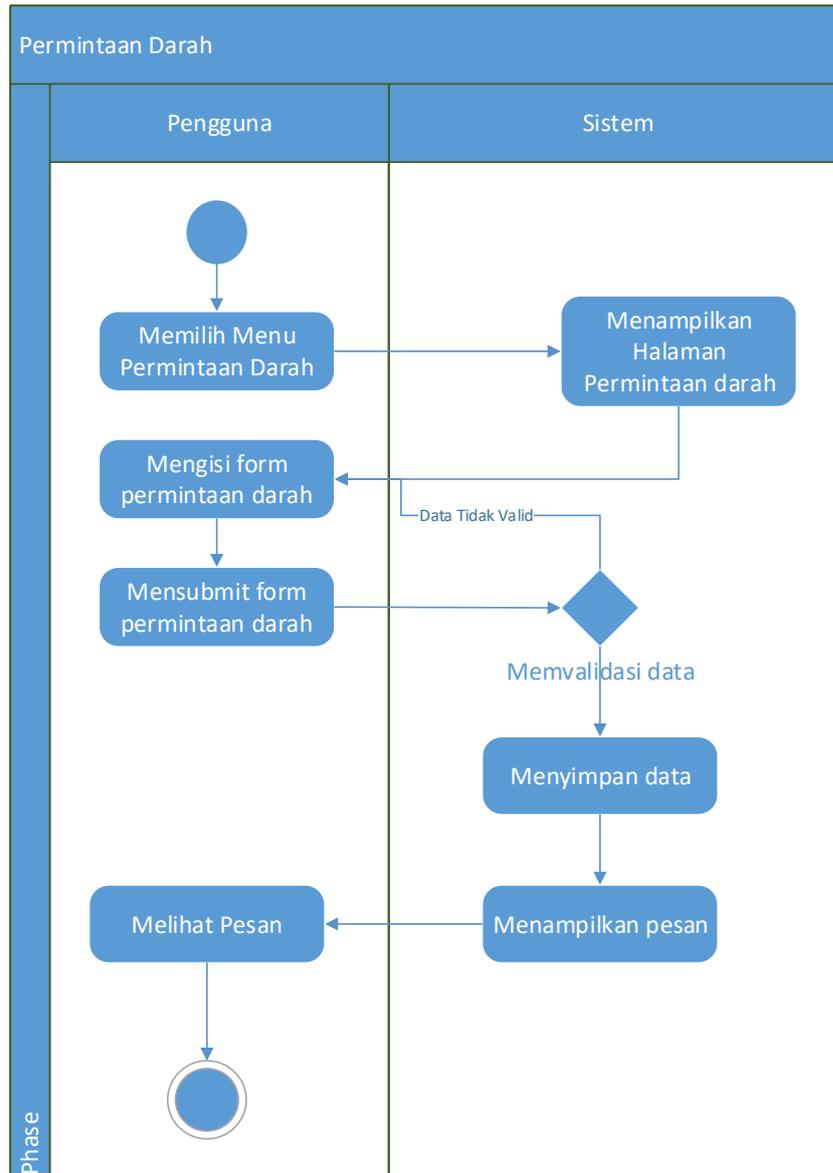
**Gambar IV2.5. 8 Activity Diagram Contact**

Menggambarkan alur proses Contact di awali dengan pengguna memilih menu Contact maka sistem akan menampilkan halaman untuk Contact. Dalam halaman Contact pengguna dapat memberikan saran dan kesan dalam penggunaan aplikasi berbasis website ini dan pengguna dapat mencantumkan event yang akan dilakukan pengguna. Setelah mengisi form Contact maka pengguna akan mendapatkan konfirmasi via email.



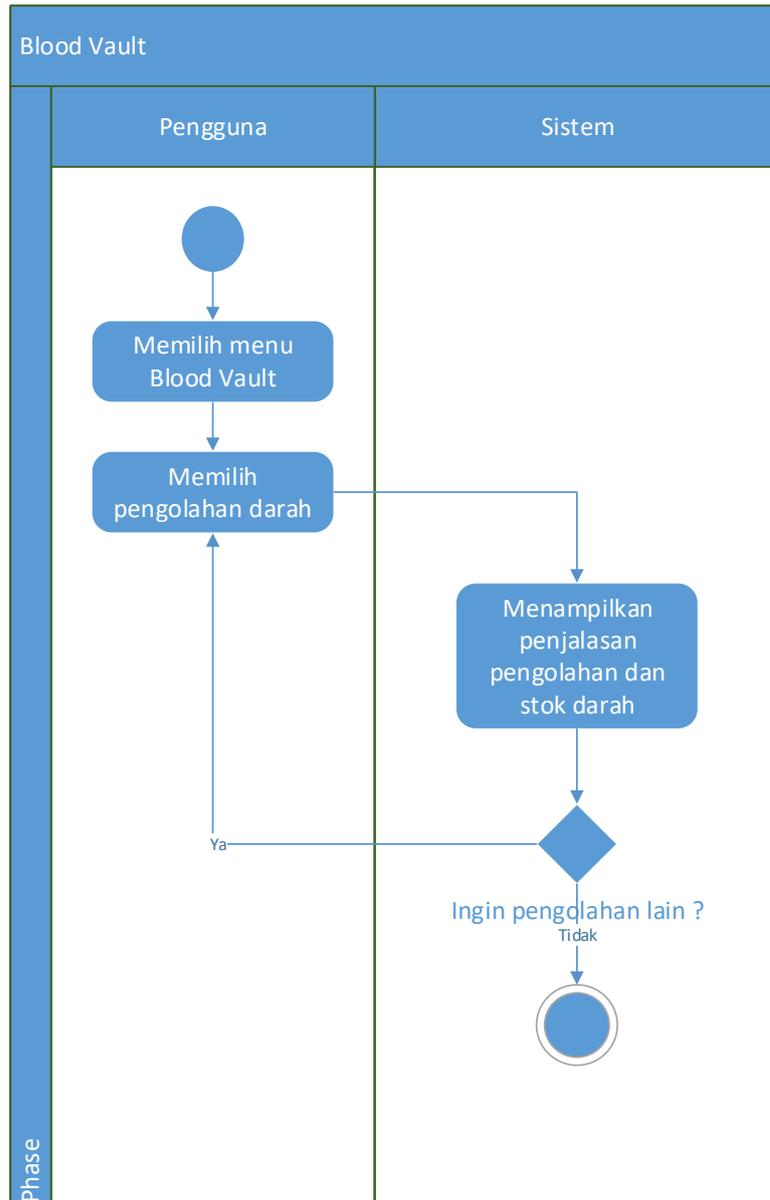
**Gambar IV2.5. 9 Activity Diagram Profil**

Menggambarkan alur proses profil dimana pengguna harus telah menyelesaikan proses login. Setelah pengguna menyelesaikan proses login maka pengguna dapat memilih menu profil maka sistem akan menampilkan halaman profil yang berisi data diri dari pengguna. Dalam halaman profil pengguna dapat mengubah data diri dengan mengisi form ubah data pada halaman profil. Setelah mengisi form ubah data maka sistem akan melakukan pengecekan terhadap data. Data dinyatakan valid maka sistem akan menyimpan perubahan dan memberikan pesan data berhasil di ubah.



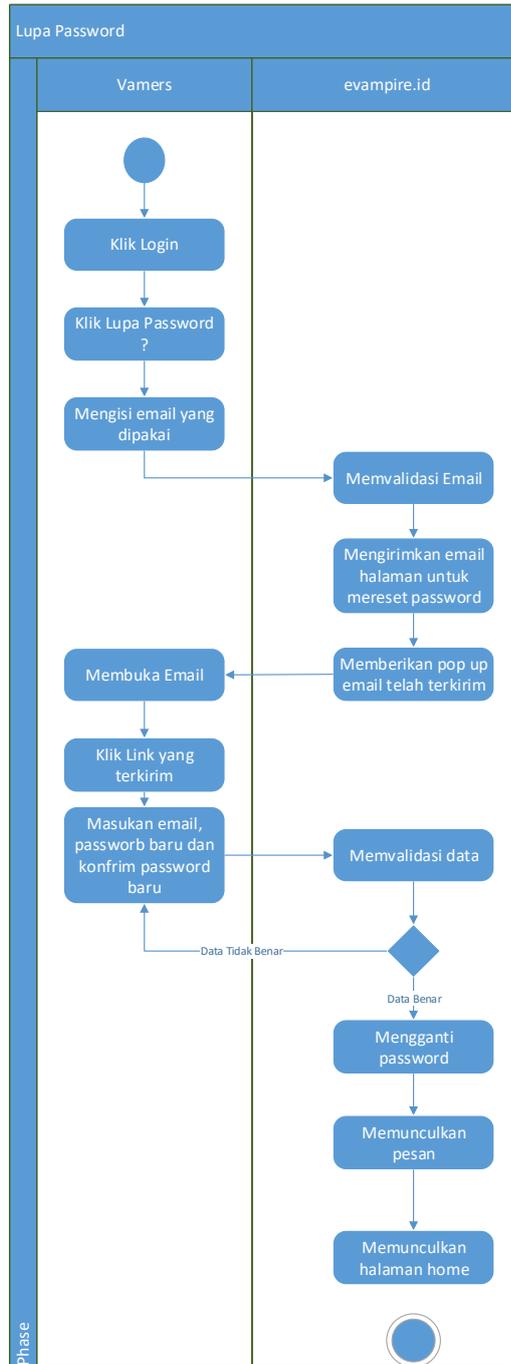
**Gambar IV2.5. 10 Activity Diagram Permintaan Darah**

Mengambarkan alur proses Permintaan Darah di mana pengguna harus telah menyelesaikan proses login. Setelah pengguna menyelesaikan proses login maka pengguna dapat mengisi form permintaan darah. Data dinyatakan valid maka sistem akan menyimpan permintaan dan memberikan pesan data berhasil di ubah.



**Gambar IV2.5. 11 Activity Diagram Blood Vault**

Menggambarkan alur proses Blood Vault di mana pengguna harus telah menyelesaikan proses login. Dalam fitur Blood Vault dapat melihat jenis – jenis kantong dan dapat melihat jumlah dan tempat di mana dapat mendapatkan data.



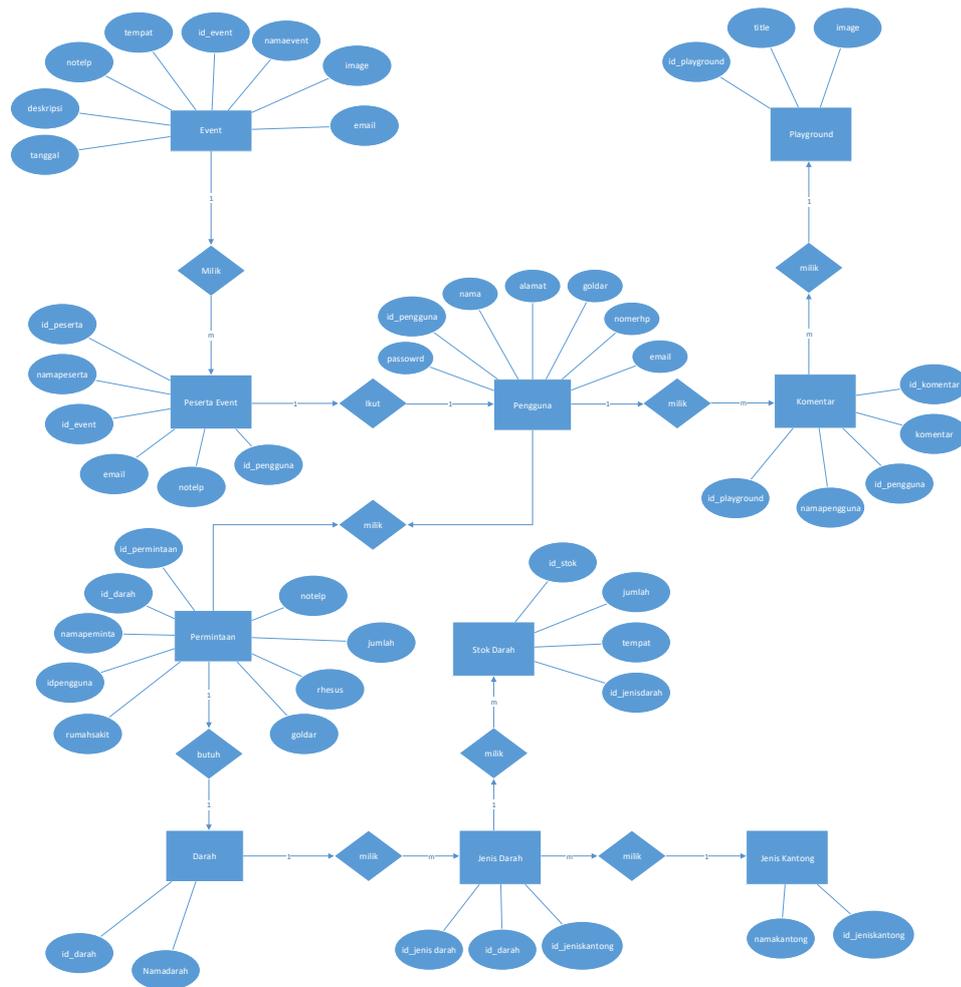
**Gambar IV2.5. 12 Activity Diagram Lupa Password**

Menggambarakan alur proses Lupa Password. Dalam fitur Lupa Password dapat melihat jenis – jenis kantong dan dapat melihat jumlah dan tempat di mana dapat mendapatkan data.



#### IV.2.8. ER Diagram

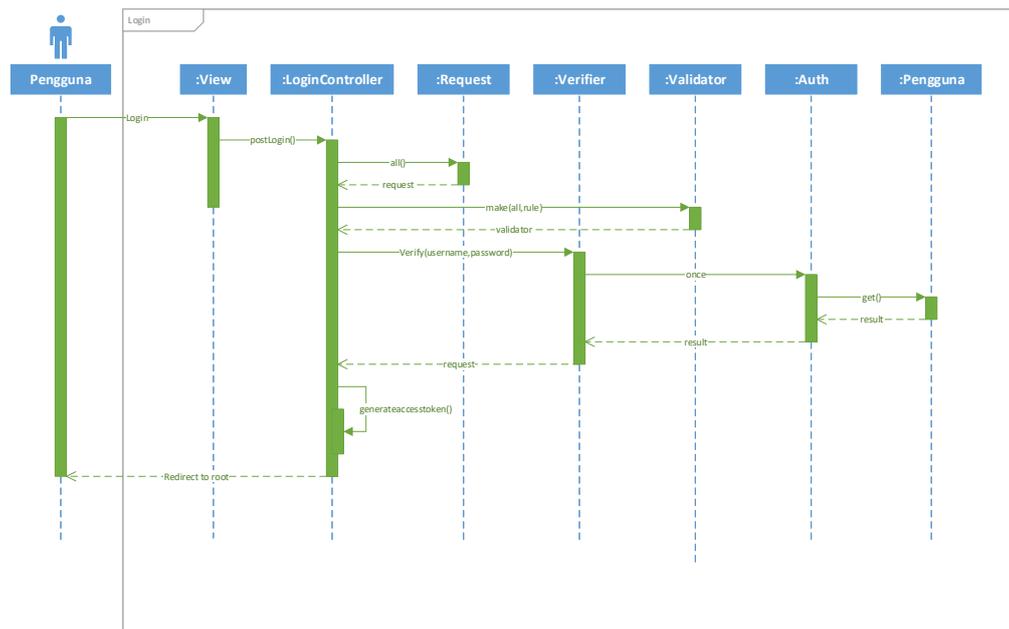
*Entity relationship diagram* adalah representasi grafis dari entitas dan hubungan satu sama lain, biasanya digunakan sebagai acuan perancangan *database* untuk melakukan proses *coding*. Berikut *ER Diagram* dalam pembuatan aplikasi E-Vampire berbasis website:



**Gambar IV.2.8. ER Diagram**

## IV.2.9. Sequence Diagram

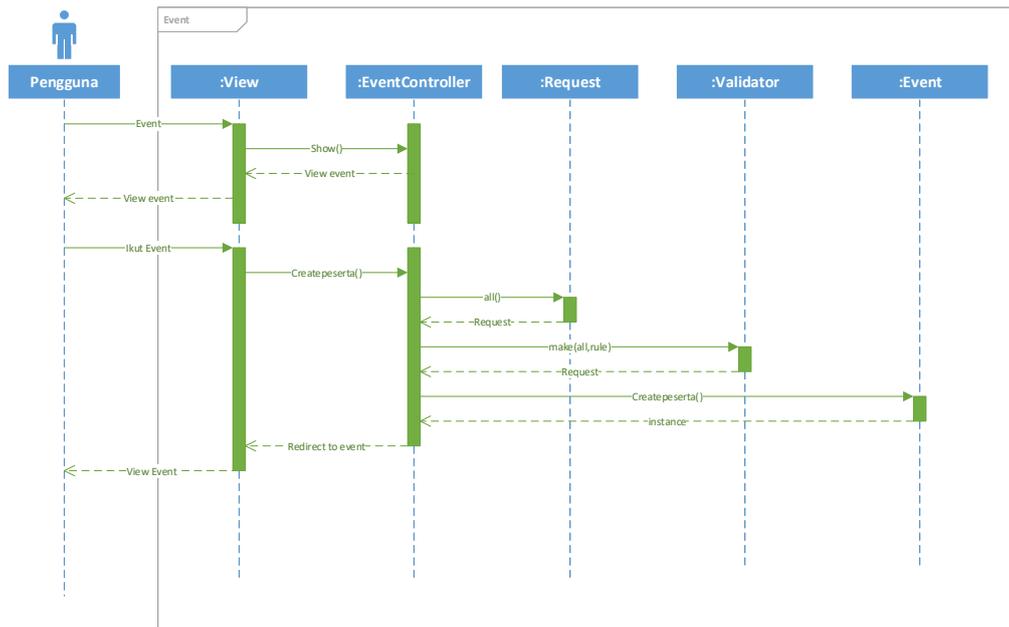
*Sequence Diagram* adalah suatu diagram yang memperlihatkan atau menampilkan interaksi – interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan / message. Berikut *sequence diagram* dalam pembuatan aplikasi E-Vampire berbasis *website*:



**Gambar IV.2.8. 1 Sequence diagram Login**

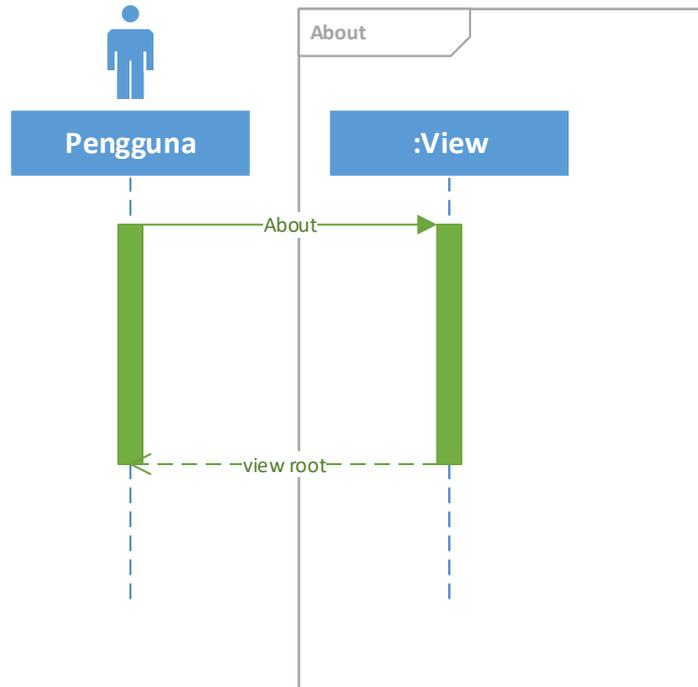
Pada Gambar IV.2.8.1 merupakan *sequence digram* Login yang menjelaskan interaksi antar setiap objek yang menjalankan proses Login. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*. Pada proses tersebut terdapat beberapa method buatan sendiri yaitu `postLogin()` dan *method* pada objek yang disediakan Laravel yaitu `all()`, `make(all,rule)`, `get()`, `verify(username, password)`, `generateaccesstoken()`.





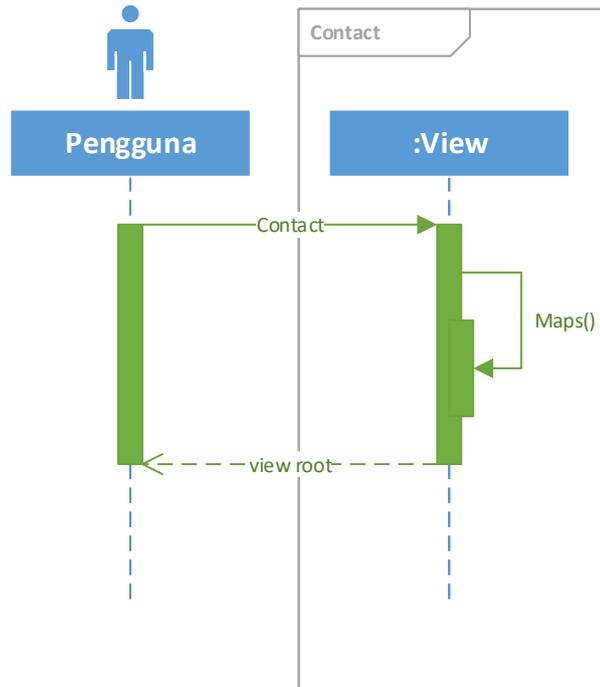
Gambar IV.2.8. 3 Sequence diagram Event

Pada Gambar IV.2.8.3 merupakan *sequence digram* Event yang menjelaskan interaksi antar setiap objek yang menjalankan proses Event. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*. Pada proses tersebut terdapat beberapa method buatan sendiri yaitu `show()`, `createPeserta()` dan *method* pada objek yang disediakan Laravel yaitu `all()`, `make(all,rule)`.



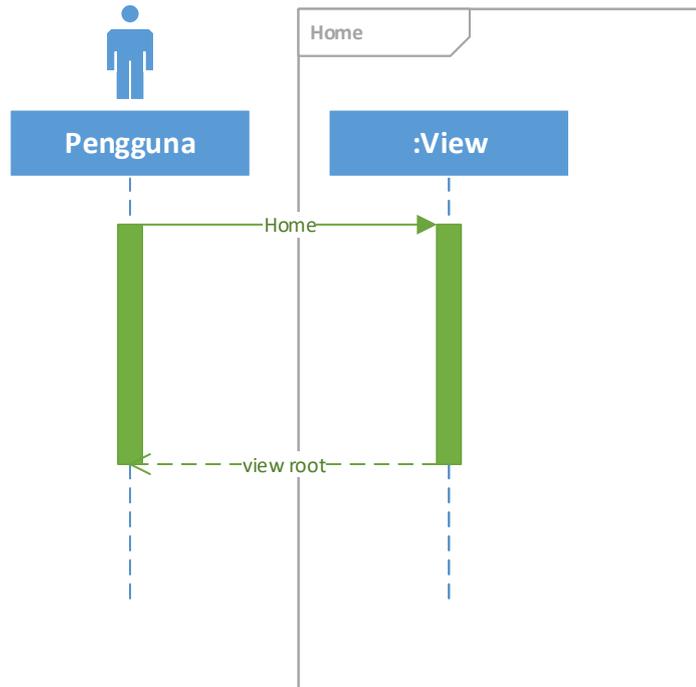
**Gambar IV.2.8. 4 Sequence diagram About**

Pada Gambar IV.2.8.4 merupakan *sequence digram* About yang menjelaskan interaksi antar setiap objek yang menjalankan proses About. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*.



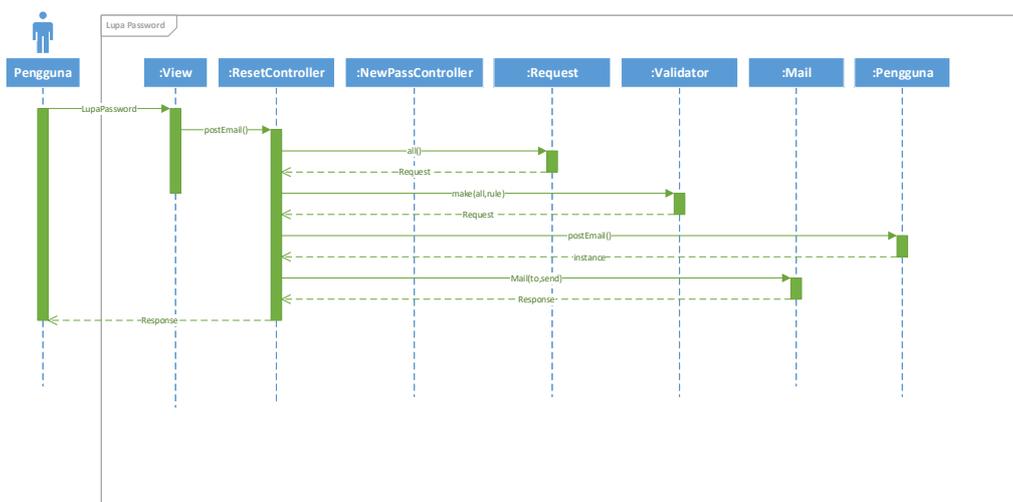
**Gambar IV.2.8. 5 Sequence diagram Contact**

Pada Gambar IV.2.8.5 merupakan *sequence digram* Contact yang menjelaskan interaksi antar setiap objek yang menjalankan proses Contact. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*.



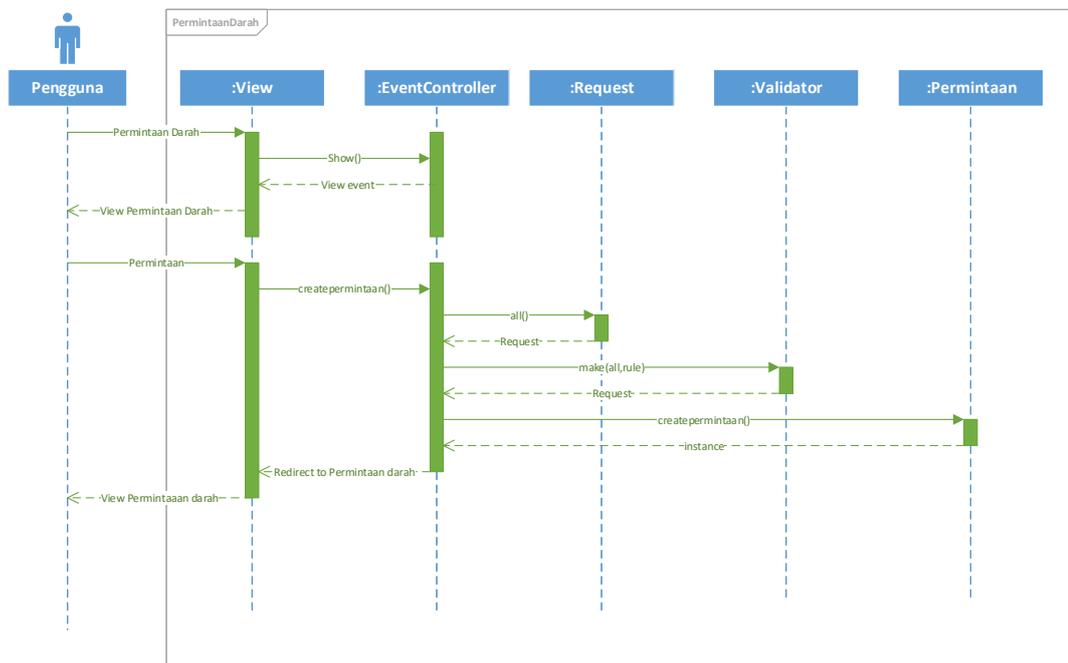
Gambar IV.2.8. 6 Sequence diagram Home

Pada Gambar IV.2.8.6 merupakan *sequence digram* Home yang menjelaskan interaksi antar setiap objek yang menjalankan proses Home. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*.



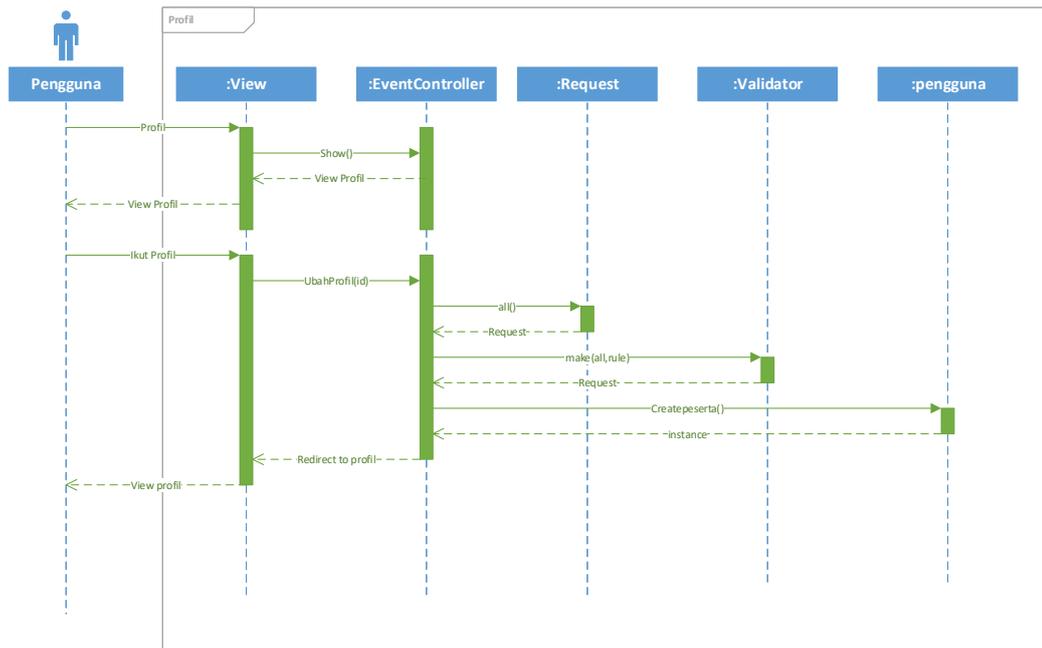
Gambar IV.2.8. 7 Sequence diagram Lupa Password

Pada Gambar IV.2.8.7 merupakan *sequence digram* Login yang menjelaskan interaksi antar setiap objek yang menjalankan proses Login. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*. Pada proses tersebut terdapat beberapa *method* buatan sendiri yaitu `postEmail()` dan *method* pada objek yang disediakan Laravel yaitu `all()`, `make(all,rule)`, `Mail(to,send)`.



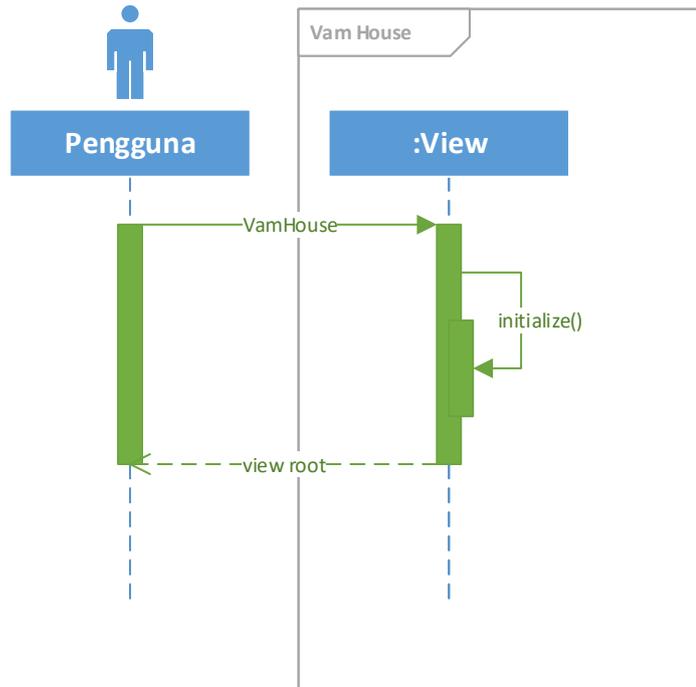
**Gambar IV.2.8. 8 Sequence diagram Permintaan Darah**

Pada Gambar IV.2.8.8 merupakan *sequence digram* Permintaan Darah yang menjelaskan interaksi antar setiap objek yang menjalankan proses Permintaan Darah. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*. Pada proses tersebut terdapat beberapa *method* buatan sendiri yaitu `show()`, `createPermintaan()` dan *method* pada objek yang disediakan Laravel yaitu `all()`, `make(all,rule)`.



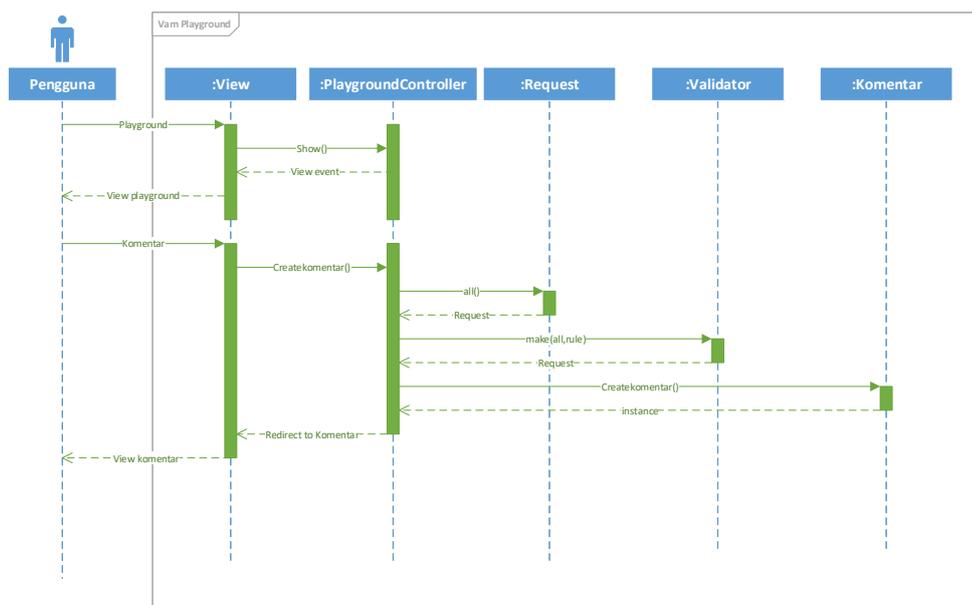
**Gambar IV.2.8. 9 Sequence diagram Profil**

Pada Gambar IV.2.8.9 merupakan *sequence digram* Profil yang menjelaskan interaksi antar setiap objek yang menjalankan proses Profil. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*. Pada proses tersebut terdapat beberapa method buatan sendiri yaitu `show()`, `postProfil()` dan *method* pada objek yang disediakan Laravel yaitu `all()`, `make(all,rule)`.



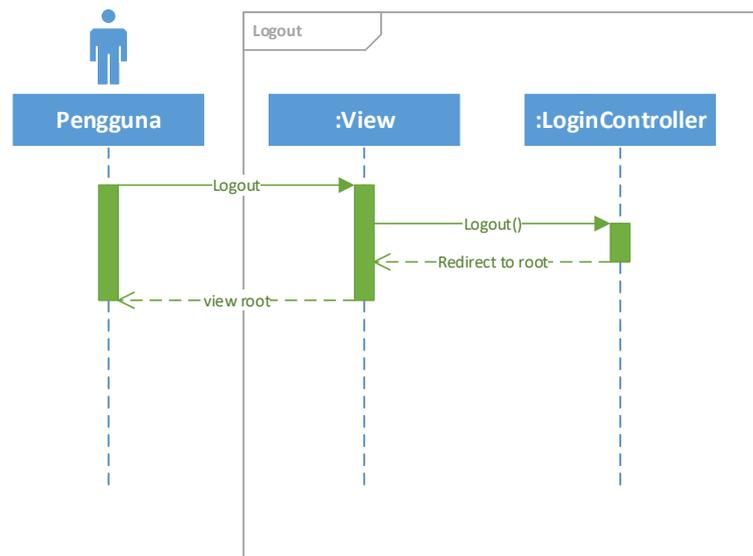
**Gambar IV.2.8. 10 Sequence diagram Vam House**

Pada Gambar IV.2.8.10 merupakan *sequence digram* Vam House yang menjelaskan interaksi antar setiap objek yang menjalankan proses Vam House. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*.



**Gambar IV.2.8. 11 Sequence diagram Vam Playgrund**

Pada Gambar IV.2.8.11 merupakan *sequence digram* Vam Playground yang menjelaskan interaksi antar setiap objek yang menjalankan proses Vam Playground. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*. Pada proses tersebut terdapat beberapa *method* buatan sendiri yaitu `show()`, `createPlayground()` dan *method* pada objek yang disediakan Laravel yaitu `all()`, `make(all,rule)`.



**Gambar IV.2.8. 12 Sequence diagram Logout**

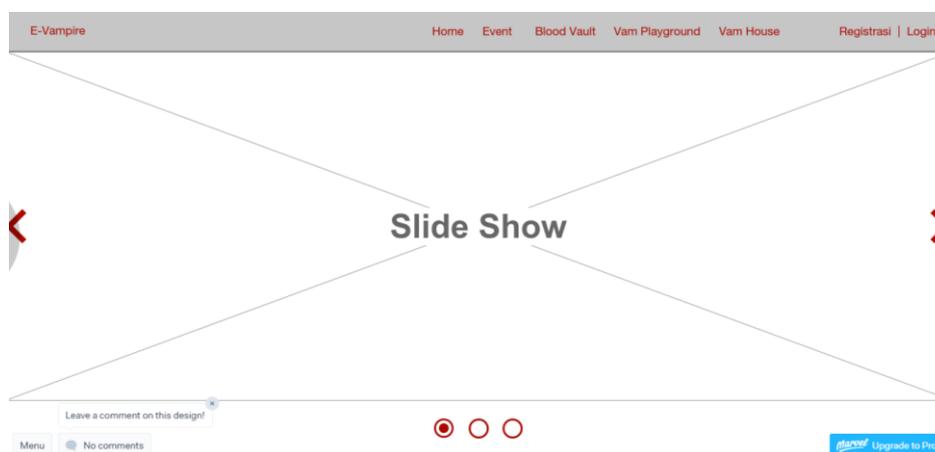
Pada Gambar IV.2.8.12 merupakan *sequence digram* Login yang menjelaskan interaksi antar setiap objek yang menjalankan proses Login. Dengan menggunakan Framework Laravel maka pada objek terpengaruh oleh setiap *method* yang ada pada Laravel dan *method* yang kita buat pada *controller* untuk mengakses pada *view* dan *model*. Pada proses tersebut terdapat beberapa *method* buatan sendiri yaitu `logout()`.

#### IV.2.10. Mockup Aplikasi

Berikut merupakan rancangan *mockup* pada aplikasi E-Vampire berbasis *website* :

ID *Mockup Interface*: MA-01

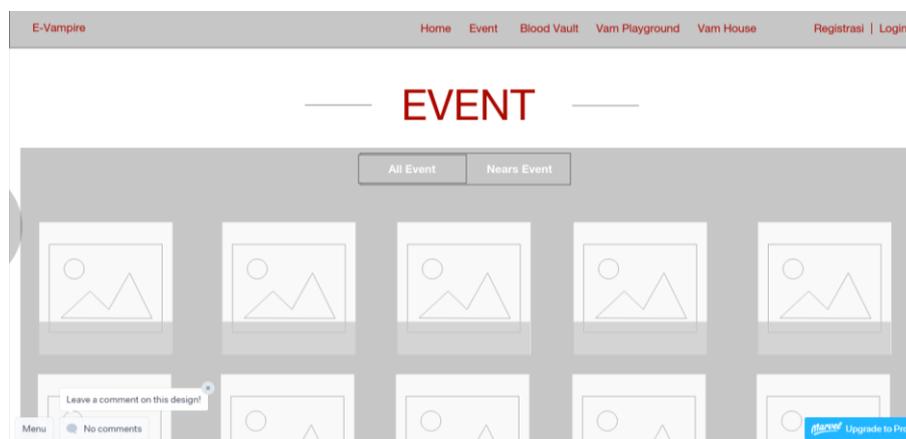
Nama *Mockup*: Halaman Home



**Gambar IV.2.10. 1 Mockup Home**

ID *Mockup Interface*: MA-02

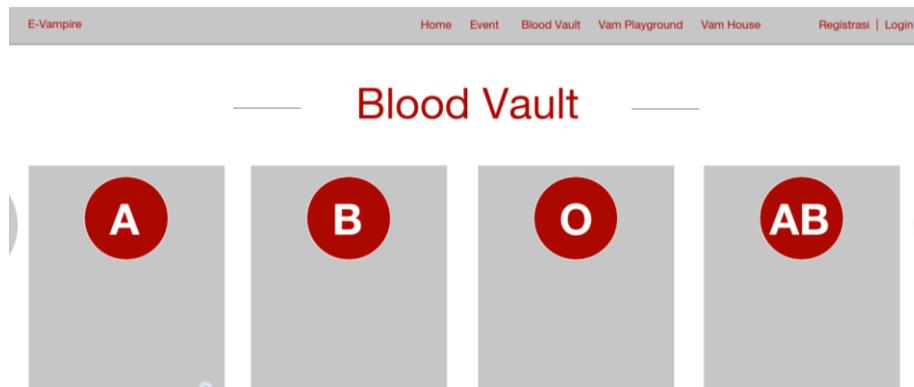
Nama *Mockup*: Halaman Event



**Gambar IV.2.10. 2 Mockup Event**

ID *Mockup Interface*: MA-03

Nama *Mockup*: Halaman Blood Vault



*Gambar IV.2.10. 3 Mockup Blood Vault*

ID *Mockup Interface*: MA-04

Nama *Mockup*: Halaman Vam Playground



*Gambar IV.2.10. 4 Mockup Vam Playground*

ID *Mockup Interface*: MA-05

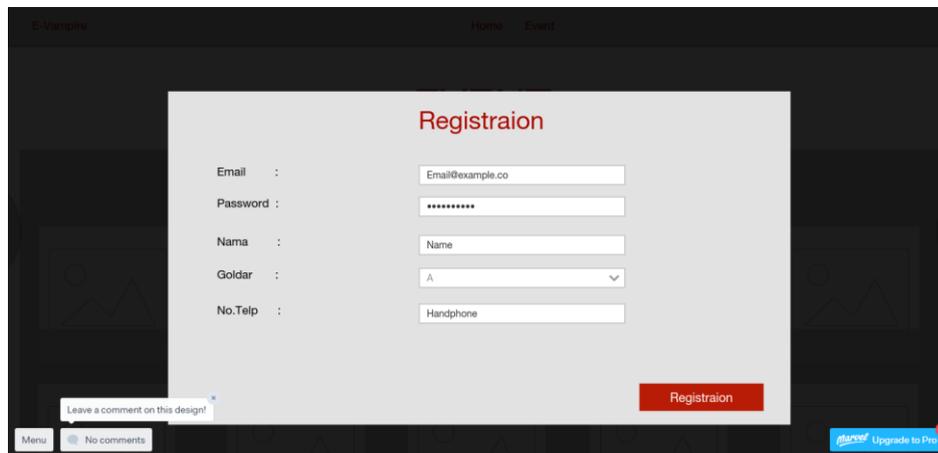
Nama *Mockup*: Halaman Vam House



**Gambar IV.2.10. 5 Mockup Vam House**

ID *Mockup Interface*: MA-06

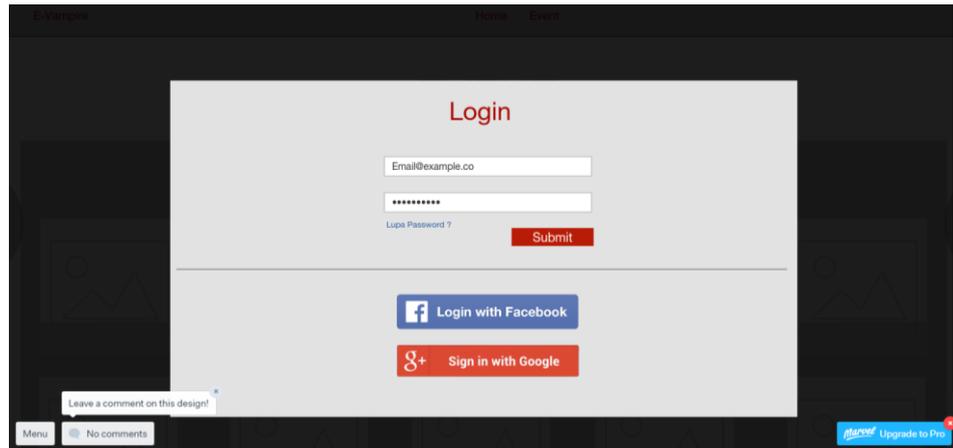
Nama *Mockup*: Pop-up Registrasi



**Gambar IV.2.10. 6 Mockup Registrasi**

ID *Mockup Interface*: MA-07

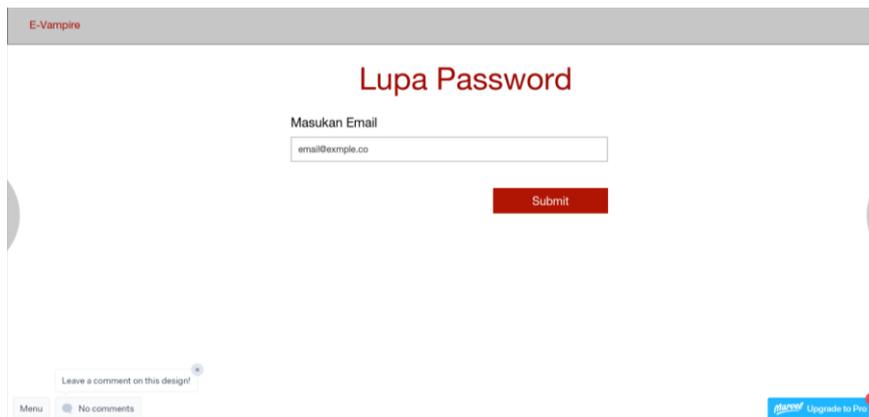
Nama *Mockup*: Pop-up Login



**Gambar IV.2.10. 7 Mockup Login**

ID *Mockup Interface*: MA-08

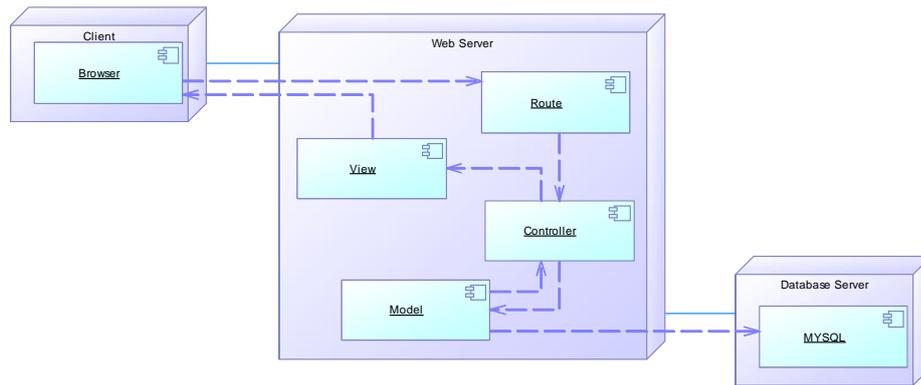
Nama *Mockup*: Halaman Lupa Password



**Gambar IV.2.10. 8 Mockup Lupa Password**

#### **IV.2.11. Deployment Diagram**

*Deployment diagram* menggambarkan tata letak fisik dari sistem, yang memperlihatkan perangkat lunak dan perangkat keras mana yang digunakan. Berikut gambar *deployment diagram* untuk aplikasi E-Vampire berbasis website:



Gambar IV.2.11. Deployment Diagram

## BAB V IMPLEMENTASI DAN PENGUJIAN

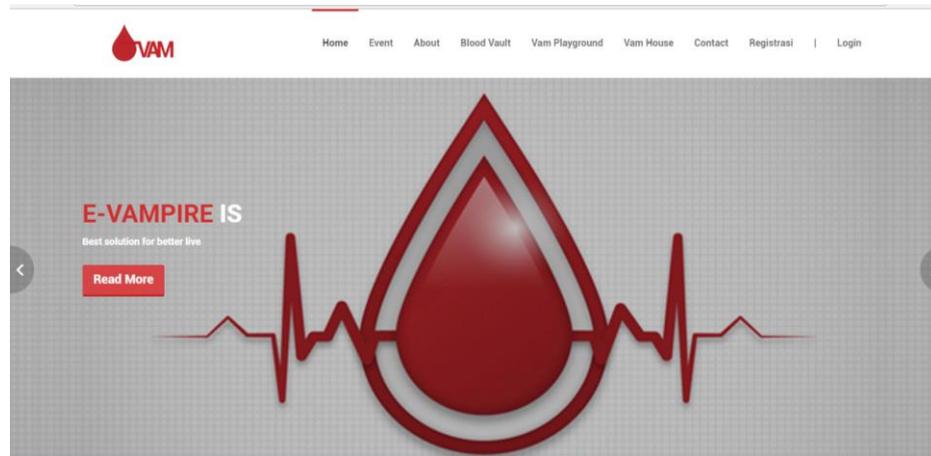
### V.1. Implementasi

Pada tahap ini merupakan hasil analisis desain yang di kembangkan menjadi sebuah aplikasi.

#### V.1.1. Hasil Antarmuka

Berikut hasil antarmuka aplikasi :

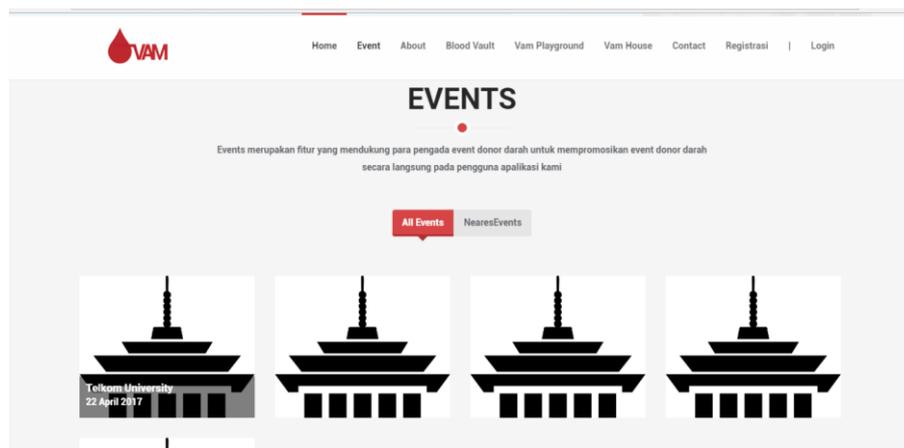
##### 1. Halaman Utama



*Gambar V.1.1 1 Halaman Utama*

Halaman ini merupakan halaman pertama atau bisa disebut Home disini bertujuan untuk memberikan beberapa informasi tentang kegiatan dan proses yang sedang dilakukan oleh pihak pengembang aplikasi ini.

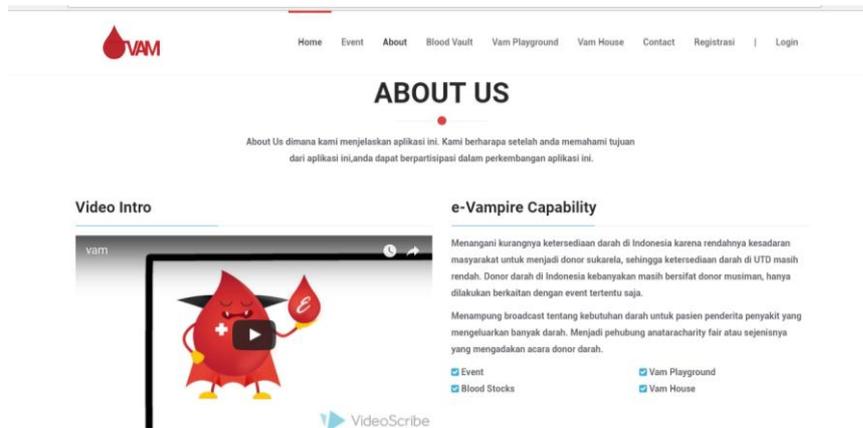
##### 2. Halaman Event



*Gambar V.1.1 2 Halaman Event*

Halaman ini merupakan halaman event dimana event dibagi menjadi dua yaitu Acara Terbaru dan Acara Mendatang.

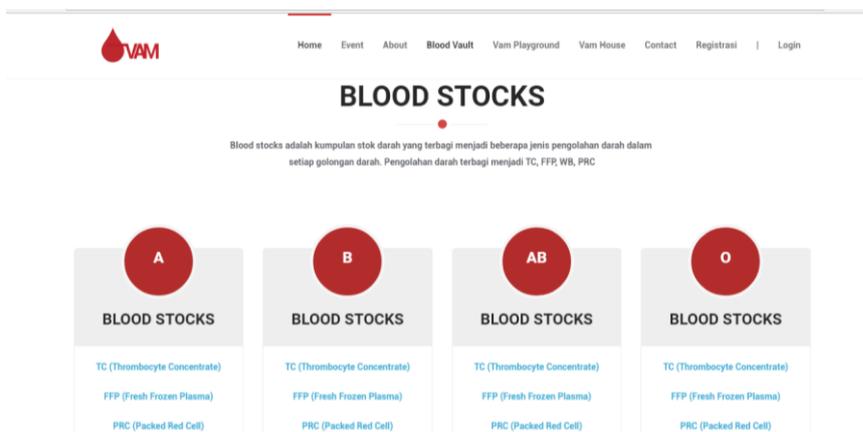
### 3. Halaman About



Gambar V.1.1 3 Halaman About

Halaman ini merupakan halaman About yang menjelaskan tentang aplikasi E-Vampire berbasis website ini.

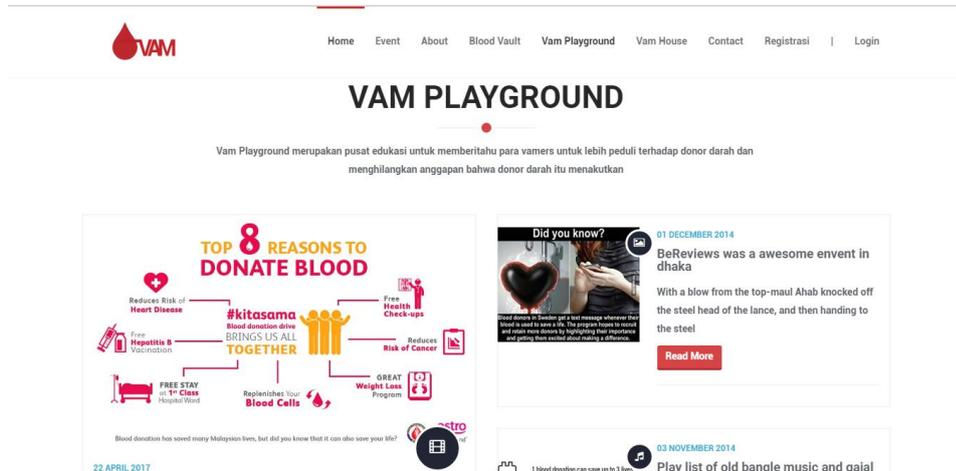
### 4. Halaman Blood Vault



Gambar V.1.1 4 Halaman Blood Vault

Halaman ini merupakan halaman Blood Vault yang menjelaskan tentang stok darah dan pengelompokan kantong darah sesuai dengan pemrosesan kantong darah tersebut.

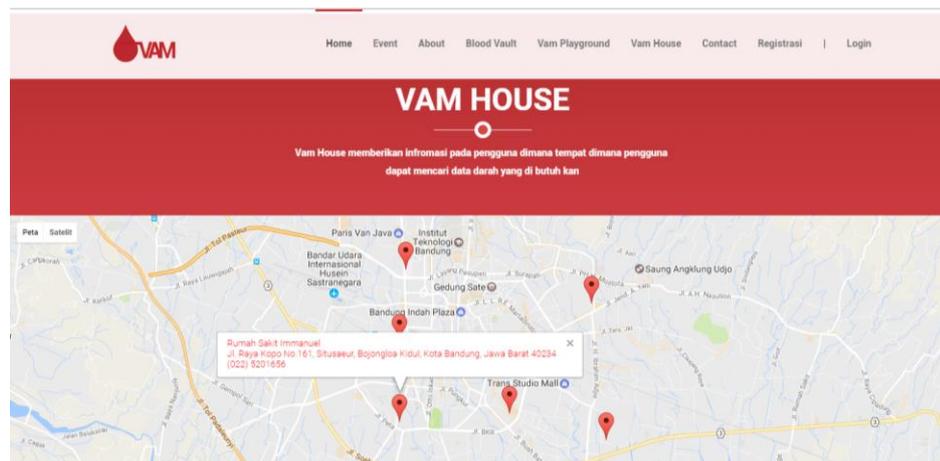
### 5. Halaman Vam Playground



Gambar V.1.1 5 Halaman Vam Playground

Halaman ini menjelaskan halaman Vam Playground dan berisi informasi-informasi yang dapat meningkatkan *awareness* pengguna terhadap donor darah dan merupakan tujuan dari pembuatan aplikasi berbasis website ini.

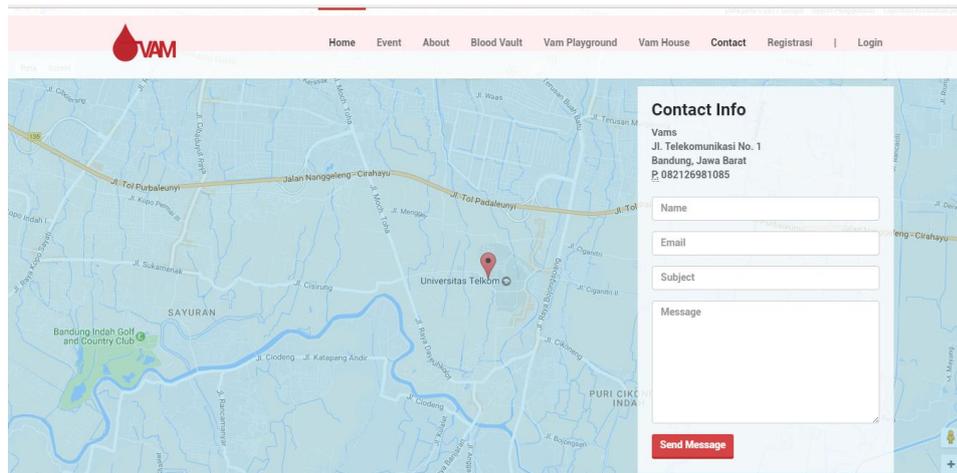
## 6. Halaman Vam House



Gambar V.1.1 6 Halaman Vam House

Halaman ini menjelaskan halaman Vam House yang berisi informasi tempat yang mempunyai andil besar dalam penyebaran kantong darah dan pusat kesehatan.

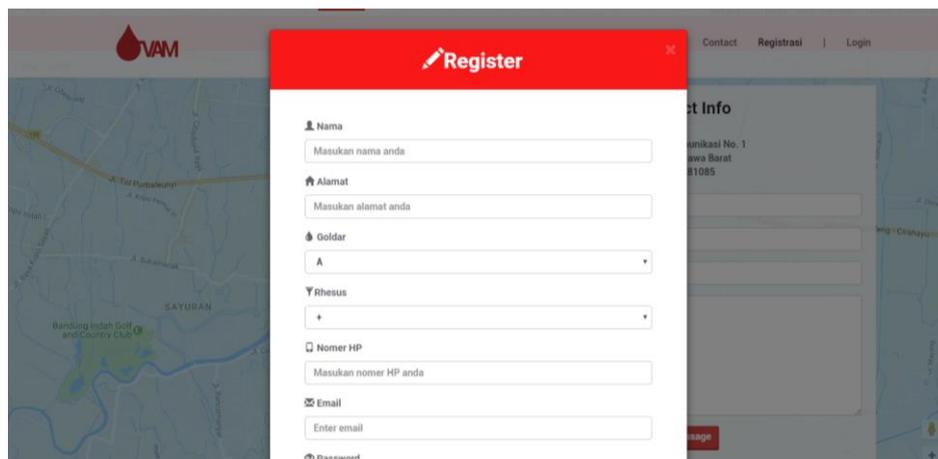
## 7. Halaman Contact



**Gambar V.1.1 7 Halaman Contact**

Halaman ini menjelaskan halaman Contact yang berisi form Contact dan alamat pengembang aplikasi berbasis website ini.

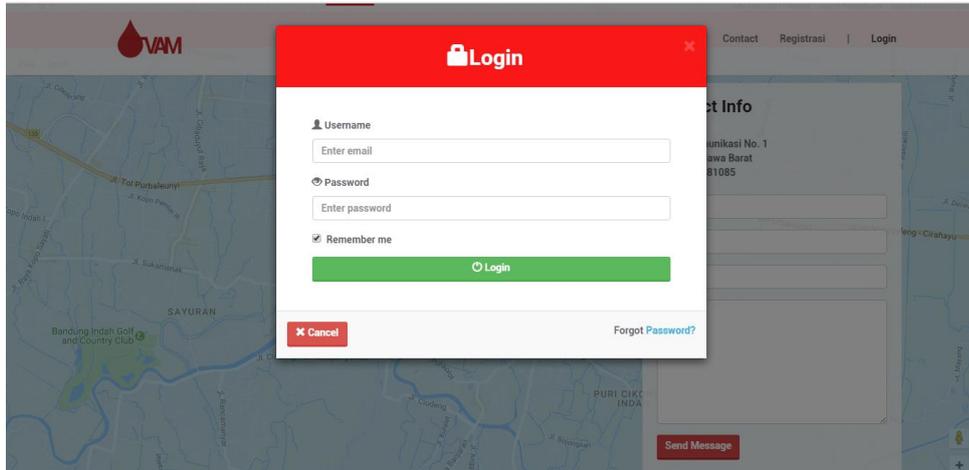
### 8. Pop-up Registrasi



**Gambar V.1.1 8 Halaman Registrasi**

Pop-up ini menjelaskan *pop-up* registrasi dimana pengguna dapat mengisi form registrasi agar mendapatkan akses untuk lebih nyaman dalam menggunakan aplikasi ini.

### 9. Pop-up Login



Gambar V.1.1 9 Halaman Login

Pop-up ini menjelaskan pop-up login dimana agar pengguna dapat masuk dalam aplikasi ini dan dapat mengakses lebih jauh dari aplikasi ini.

### V.1.2. Implementasi Database

Database dibuat melalui proses perancangan data yang menggunakan er-model dan class diagram. Pada perancangan er-model dan class diagram dapat menyusun setiap table pada database. Berikut merupakan implementasi database dari aplikasi E-Vampire berbasis website.

<b>evam event</b> id_event : int(10) namaevent : varchar(100) eimage : varchar(100) email : varchar(30) etempat : text enotelp : varchar(12) edesc : text tanggal : varchar(30)	<b>evam tb_stokdarah</b> id_stok : int(11) jumlah : int(11) lokasi : text id_jenisdarah : int(11)	<b>evam tb_komentar</b> id_komentar : int(11) komentar : text id_vamers : int(11) namavamers : varchar(30) id_playground : int(11)	<b>evam tb_vamers</b> id_vamers : int(11) nama : varchar(30) alamat : text goldar : varchar(2) nomerhp : varchar(12) email : varchar(30) password : varchar(8) remember_token : varchar(100) reset_token : varchar(32) expired : int(11)
<b>evam tb_jenisdarah</b> id_jenisdarah : int(11) id_darah : int(11) id_jenis : int(11)	<b>evam tb_rhesus</b> id_rhesus : int(11) namarhesus : varchar(10)	<b>evam tb_playground</b> id_playground : int(11) title : varchar(30) image : varchar(100)	<b>evam tb_permintaan</b> id_permintaan : int(11) namapeminta : varchar(30) rumahsakit : varchar(100) goldar : varchar(10) rhesus : varchar(10) jeniskantong : varchar(30) jumlah : int(10) nomer : varchar(12)
<b>evam tb_jeniskantong</b> id_jenis : int(11) namajenis : varchar(11)	<b>evam tb_login</b> id_login : int(11) username : varchar(30) password : varchar(8) role : varchar(10)	<b>evam peserta_event</b> id_peserta : int(11) nama_peserta : varchar(30) id_event : int(11) email : varchar(30) no_telp : varchar(12) nama_event : varchar(30)	
<b>evam tb_darah</b> id_darah : int(11) namadarah : varchar(10)			

Gambar V.I.2. Database E-Vampire

### V.1.3. Hasil Implementasi Bisnis

E-Vampire merupakan aplikasi pendonoran darah berbasis *website* yang membantu untuk meningkatkan *awareness* terhadap pendonor darah menangani kekurangan kantong darah. Bisnis yang sedang dijalankan saat ini termasuk pada jenis bisnis atau kewirausahaan sosial teknologi aplikasi. Bisnis atau kewirausahaan sosial merupakan inovasi sosial yang mampu mengubah system yang ada di masyarakat. E-Vampire tergolong pada bisnis atau kewirausahaan sosial teknologi aplikasi karena E-Vampire merupakan sebuah aplikasi yang dapat membantu meningkatkan *social value*, *civil society*, dan *awareness* pada masyarakat.

Saat ini E-Vampire telah memiliki beberapa media sosial guna memperkenalkan E-Vampire pada masyarakat, diantaranya :

1. Facebook :

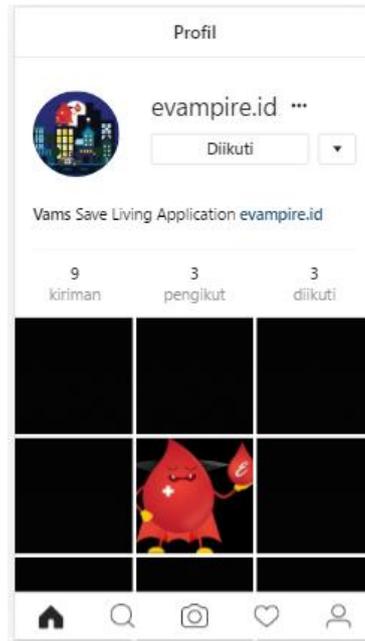
<https://www.facebook.com/E-Vampire-635462789971673/?fref=ts>



Gambar V.1.3. 1 Tampilan Facebook E-Vampire

2. Instagram

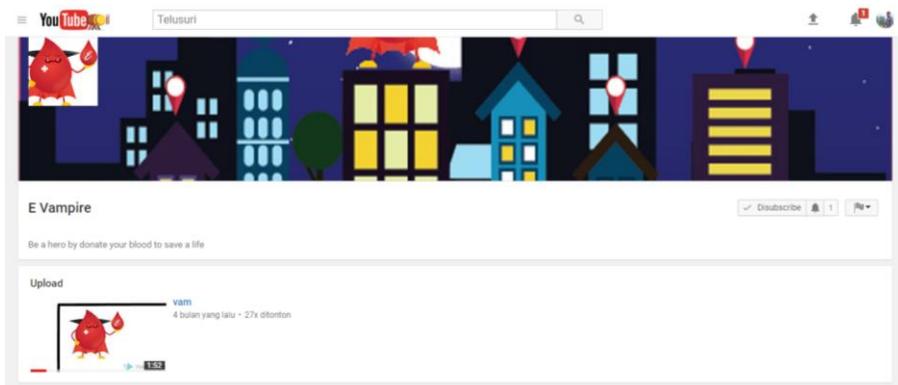
<https://www.instagram.com/evampire.id/>



**Gambar V.1.3. 2 Tampilan Instagram E-Vampire**

3. Youtube Chanel

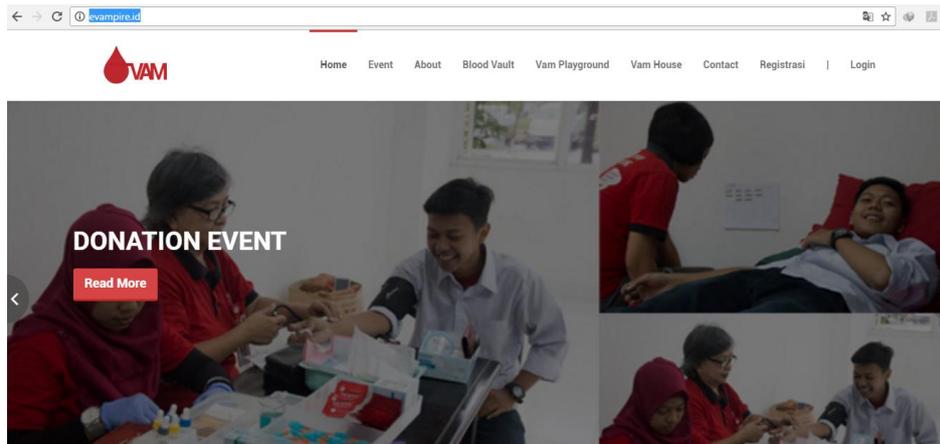
<https://www.youtube.com/channel/UCCCuZhAuxJi8gSD3nS2R18w>



**Gambar V.1.3. 3 Tampilan Youtube E-Vampire**

4. Website

<http://evampire.id/>

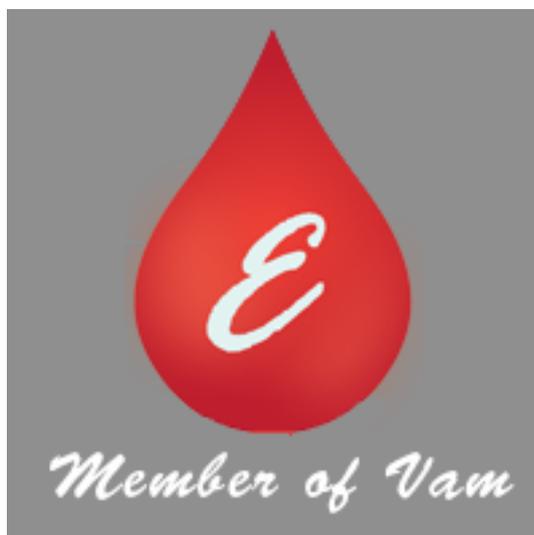


**Gambar V.1.3. 4 Tampilan Website E-Vampire**

Selain menggunakan medial sosial untuk memasarkan dan mengenalkan E-Vampire dan untuk mendapatkan nilai unik agar mudah diingat oleh masyarakat E-Vampire juga membuat pin dan *sticker*. Gambar dibawah ini merupakan pin, gantungan kunci, dan sticker yang E-Vampire gunakan sebagai media promosi.



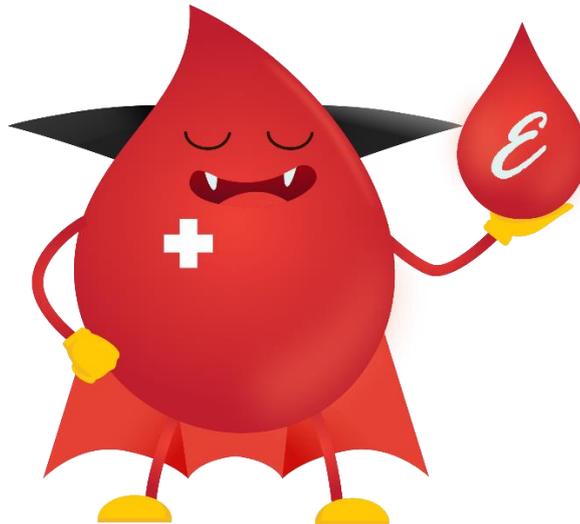
**Gambar V.1.3. 4 Gantungan Kunci E-Vampire**



**Gambar V.1.3. 5 Pin E-Vampire**



**Gambar V.1.3. 6 Stiker E-Vampire**



**Gambar V.1.3.7 Logo E-Vampire**

## **V.2. Pengujian**

Pada tahap ini, dilakukan beberapa pengujian untuk memastikan apakah aplikasi yang telah dibuat dapat berjalan sesuai dengan yang diharapkan.

### **V.2.1. Pengujian Fungsionalitas**

Pengujian fungsionalitas dilakukan untuk memastikan bahwa setiap fungsi yang terdapat pada *usecase* dapat berjalan sesuai dengan yang diharapkan. Skenario pengerjaan pengujian dapat dilihat pada tabel berikut:

ID	Deskripsi	Prosedur	Masukan	Harapan	Hasil	
					Berhasil	Gagal
TC-01	Pendaftaran pengguna	Mengisi form pendaftaran, dan mengisi form informasi pengguna.	Nama: Fadli Alamat: Jl. Sukabirus Goldar: B Nomer hp: 082126981085 Email: <a href="mailto:fadli.anugrah98@gmail.com">fadli.anugrah98@gmail.com</a> Password: *****	Proses pendaftaran berhasil dan menampilkan halaman Home.	√	
TC-02	Login pengguna	Mengisi <i>email</i> dan kata sandi lalu menekan menu login.	Email : <a href="mailto:fadli.anugrah98@gmail.com">fadli.anugrah98@gmail.com</a> Password: *****	Pengguna dapat masuk ke aplikasi dan menampilkan halaman Home.	√	
TC-03	Logout pengguna	Menekan menu logout	-	Pengguna dapat keluar dari aplikasi.	√	
TC-04	Reset Password	Mengisi form reset <i>password</i> , membuka link email, dan mengganti <i>password</i> baru.	Email : <a href="mailto:fadli.anugrah98@gmail.com">fadli.anugrah98@gmail.com</a>	Pengguna dapat mengganti <i>password</i> dengan yang baru.	√	
TC-05	Login dengan media sosial	Menekan logo media sosial pada halaman Login.	-	Pengguna dapat masuk aplikasi dengan akun media sosial.	√	
TC-06	Perbarui profil	Mengisi form, dan menekan tombol “Ubah Profil”,	Nama: Fadli Alamat: Jl. Sukabirus Goldar: B Nomer hp: 082126981085 Email:	Pengguna dapat mengganti data profil.	√	

ID	Deskripsi	Prosedur	Masukan	Harapan	Hasil	
					Berhasil	Gagal
			<a href="mailto:fadli.anugrah98@gmail.com">fadli.anugrah98@gmail.com</a> Password: *****			
TC-07	Melihat daftar acara donor darah	Menekan menu Event	-	Pengguna dapat melihat daftar acara donor darah.	√	
TC-08	Mengikuti acara donor darah	Menekan salah satu acara, menekan tombol "Ikuti" pada halaman detail acara.	Nama: Fadli Alamat: Jl. Sukabirus Nomer hp: 082126981085	Pengguna dapat mengikuti sebuah acara donor darah.	√	
TC-09	Batal mengikuti acara donor darah	Menekan salah satu acara, menekan tombol "Batal Ikuti" pada halaman detail acara.	-	Pengguna dapat membatalkan mengikuti sebuah acara donor darah.	√	
TC-10	Melihat tanggal terakhir melakukan donor darah	Menekan tombol menu "Profil".	-	Pengguna dapat melihat tanggal terakhir melakukan donor darah.	√	
TC-11	Melihat tanggal rekomendasi melakukan donor darah	Menekan tombol menu "Profil".	-	Pengguna dapat melihat tanggal rekomendasi melakukan donor darah.	√	
TC-12	Melihat informasi persediaan darah	Menekan menu "Blood Vault".	-	Pengguna dapat melihat informasi persediaan darah	√	
TC-13	Melihat informasi jumlah darah pada bank darah	Menekan menu "Blood Vault" menekan macam-macam kantong darah	-	Pengguna dapat melihat informasi jumlah	√	

ID	Deskripsi	Prosedur	Masukan	Harapan	Hasil	
					Berhasil	Gagal
				darah pada bank darah		
TC-14	Membaca informasi edukatif	Menekan tombol menu "Vam Playground".	-	Pengguna dapat membaca informasi edukatif	√	
TC-15	Melihat lokasi rumah sakit dan kantor PMI	Menekan tombol menu "Vam House" .	-	Pengguna dapat melihat lokasi rumah sakit dan kantor PMI di peta Google Maps	√	
TC-16	Memesan kebutuhan darah darurat	Menekan menu permintaan darah mengisi form dan menekan tombol "Butuh Darah Darurat?".	Nama: Fadli Alamat: Jl. Sukabirus Nomer hp: 082126981085	Pengguna dapat memesan kebutuhan darah darurat	√	

### V.2.2. *Cross-Browser*

*Cross-Browser* atau *browser testing* salah satu bagian perancangan aplikasi E-Vampire berbasis website guna untuk mendapatkan hasil tampilan pada setiap browser di setia OS. Dalam pengujian ini aplikasi E-Vampire dijalankan pada BrowserShots yang merupakan aplikasi open source untuk memperoleh hasil tampilan aplikasi pada setiap browser. Berikut merupakan browser yang di coba untuk menguji aplikasi E-Vampire berbasis website dengan mengakses URL aplikasi E-Vampire berbasis website.

**Tabel V.2.2. Cross Browser**

OS	Linux	
Browser	Chrome 58.0.3029.96	Arora 0.11.0
Dimensi	1440x900	1024x768
Durasi	1 m 8 s	29 s

OS	Windows	
Browser	Chrome 39.0.2171.95	Chrome 39.0.2171.95
Dimensi	1280x1224	1280x1224
Durasi	1 m 22 s	1 m 22 s
OS	Mac OS	
Browser	Safari 9.1.3	Safari 9.1.3
Dimensi	1600x940	1600x940
Durasi	12 m 18 s	12 m 18 s

<b>Browser:</b>	Chrome 45.0.2454.101
<b>Engine:</b>	AppleWebKit 537.36
<b>Javascript:</b>	1.7
<b>Java:</b>	disabled
<b>Flash:</b>	disabled
<b>Factory:</b>	mac-x-11
<b>Operating System:</b>	Mac OS X 10.8 (Mountain
<b>Dimensions:</b>	1600x900
<b>File Size:</b>	97,5 KB
<b>Requested:</b>	2017-06-06 04:32:07
<b>Started:</b>	2017-06-06 04:44:33
<b>Page Loaded:</b>	2017-06-06 04:44:54
<b>Uploaded:</b>	2017-06-06 04:45:30

***Gambar V.2.2. 1 Test Browser***

<b>Browser:</b>	Safari 9.1.3
<b>Engine:</b>	AppleWebKit 601.7.8
<b>Javascript:</b>	1.7
<b>Java:</b>	enabled
<b>Flash:</b>	disabled
<b>Factory:</b>	mac-x-11
<b>Operating System:</b>	Mac OS X 10.8 (Mountain
<b>Dimensions:</b>	1600x940
<b>File Size:</b>	710,9 KB
<b>Requested:</b>	2017-06-06 04:32:07
<b>Started:</b>	2017-06-06 04:45:29
<b>Page Loaded:</b>	2017-06-06 04:45:38
<b>Uploaded:</b>	2017-06-06 04:46:25

***Gambar V.2.2. 2 Test Browser***

<b>Browser:</b>	Safari 9.1.3
<b>Engine:</b>	AppleWebKit 601.7.8
<b>Javascript:</b>	1.7
<b>Java:</b>	enabled
<b>Flash:</b>	disabled
<b>Factory:</b>	mac-x-11
<b>Operating System:</b>	Mac OS X 10.8 (Mountain
<b>Dimensions:</b>	1600x940
<b>File Size:</b>	710,9 KB
<b>Requested:</b>	2017-06-06 04:32:07
<b>Started:</b>	2017-06-06 04:45:29
<b>Page Loaded:</b>	2017-06-06 04:45:38
<b>Uploaded:</b>	2017-06-06 04:46:25

***Gambar V.2.2. 3 Test Browser***

<b>Browser:</b>	Chrome 39.0.2171.95
<b>Engine:</b>	AppleWebKit 537.36
<b>Javascript:</b>	1.7
<b>Java:</b>	enabled
<b>Flash:</b>	disabled
<b>Factory:</b>	hz-win-2
<b>Operating System:</b>	Windows 2008 R2 (Server)
<b>Dimensions:</b>	1280x1224
<b>File Size:</b>	92,2 KB
<b>Requested:</b>	2017-06-06 04:32:07
<b>Started:</b>	2017-06-06 04:32:22
<b>Page Loaded:</b>	2017-06-06 04:32:55
<b>Uploaded:</b>	2017-06-06 04:33:29

***Gambar V.2.2. 4 Test Browser***

<b>Browser:</b>	Arora 0.11.0
<b>Engine:</b>	AppleWebKit 534.34
<b>Javascript:</b>	1.7
<b>Java:</b>	disabled
<b>Flash:</b>	disabled
<b>Factory:</b>	ny-linux-2
<b>Operating System:</b>	Debian 6.0 (squeeze)
<b>Dimensions:</b>	1024x768
<b>File Size:</b>	481,6 KB
<b>Requested:</b>	2017-06-06 04:32:07
<b>Started:</b>	2017-06-06 04:32:09
<b>Page Loaded:</b>	2017-06-06 04:32:18
<b>Uploaded:</b>	2017-06-06 04:32:36

***Gambar V.2.2. 5 Test Browser***

<b>Browser:</b>	Chrome 53.0.2785.143
<b>Engine:</b>	AppleWebKit 537.36
<b>Javascript:</b>	1.7
<b>Java:</b>	enabled
<b>Flash:</b>	11
<b>Factory:</b>	chicago-linux-3
<b>Operating System:</b>	Ubuntu 12.04 LTS (Precise
<b>Dimensions:</b>	1024x768
<b>File Size:</b>	597,5 KB
<b>Requested:</b>	2017-06-06 04:32:07
<b>Started:</b>	2017-06-06 04:32:10
<b>Page Loaded:</b>	2017-06-06 04:32:18
<b>Uploaded:</b>	2017-06-06 04:32:40

**Gambar V.2.2. 6 Test Browser**

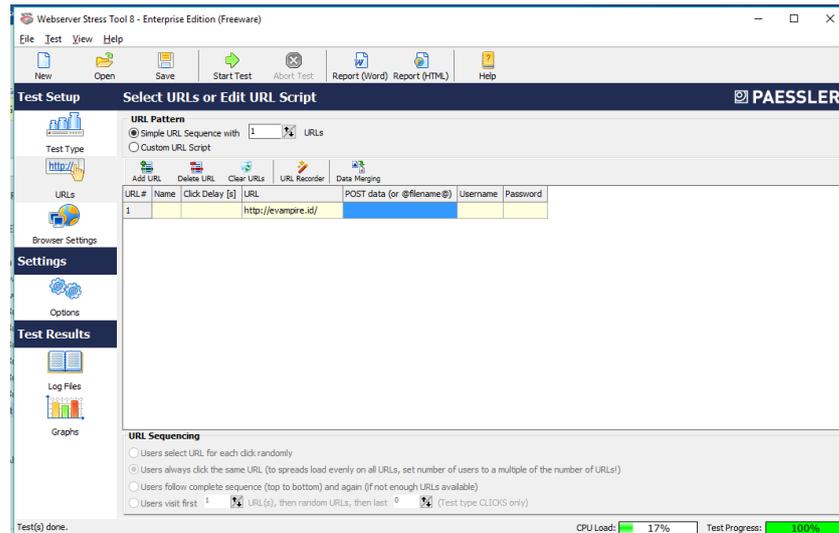
Dalam pengujian yang telah dilakukan diatas hasil dari captured dari setiap pengujian dapat dilihat dalam gambar diatas. Pengujian *cross browser* yang telah dilakukan maka dapat disimpulkan untuk mengakses aplikasi E-Vampire berbasis *website* lebih disarankan untuk menggunakan *browser* Chrome yang dapat dijalankan pada setia OS.

### **V.2.3. Stress Testing**

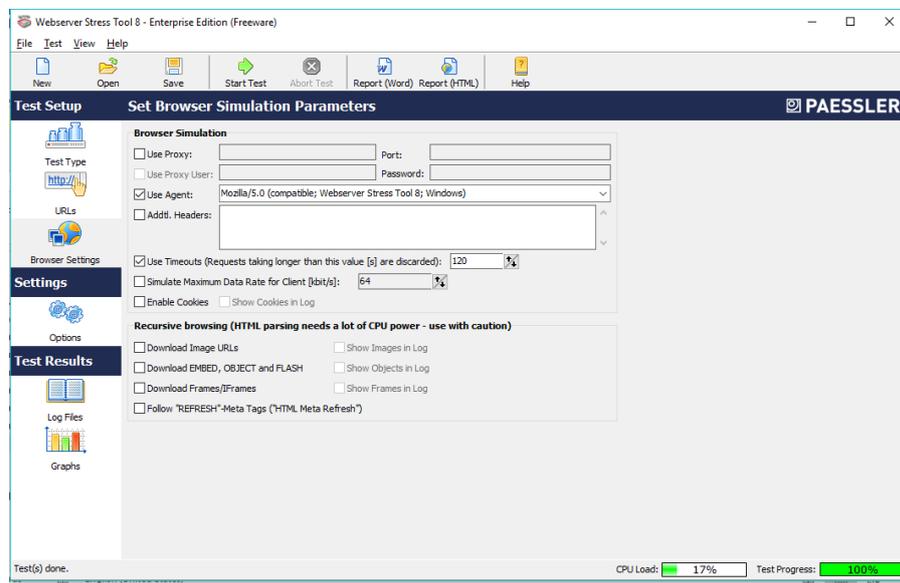
Stress testing adalah salah satu jenis pengujian system. Pengujian ini bertujuan untuk melihat apakah aplikasi mampu menangani kebutuhan sumberdaya yang tidak normal. Sumberdaya yang dimaksud ialah data dalam jumlah sangat besar, dengan frekuensi sangat tinggi apakah berakibat mengganggu fungsionalitas dari aplikasi. Pada tahap pengujian ini *tool* yang digunaka ialah Webserver Stress Tool 8 dari PAESSLER. Dalam *tool* ini memiliki tiga type pengujian yaitu: Clicks, Time, Ramp. Berikut tahapan *stress testing*.

#### **1. Config Testing**

Config Tesing disini penguji memasukan konfigurasi dari URL aplikasi E-Vampire dan memasukan browser yang akan menguji aplikasi dengan cara mengakses URL yang telah di konfigurasi sebelum nya.



**Gambar V.2.3. 1 Config Testing**

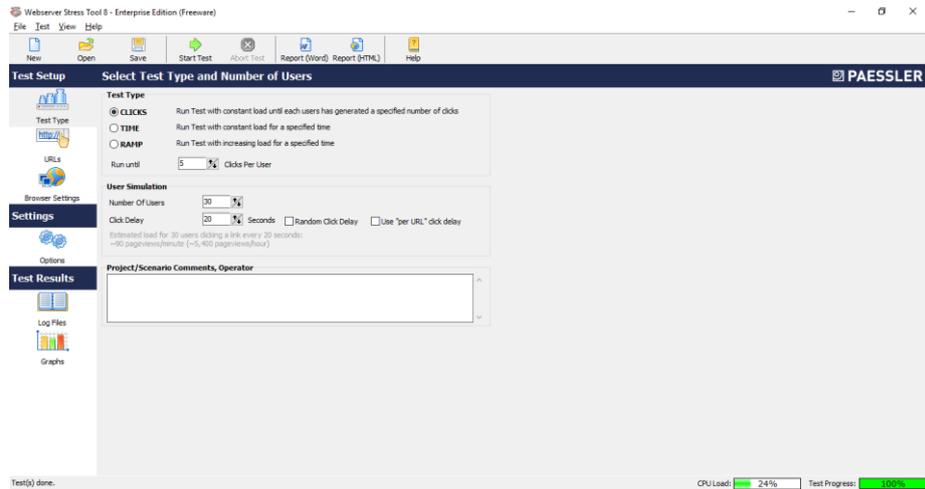


**Gambar V.2.3. 2 Config Testing**

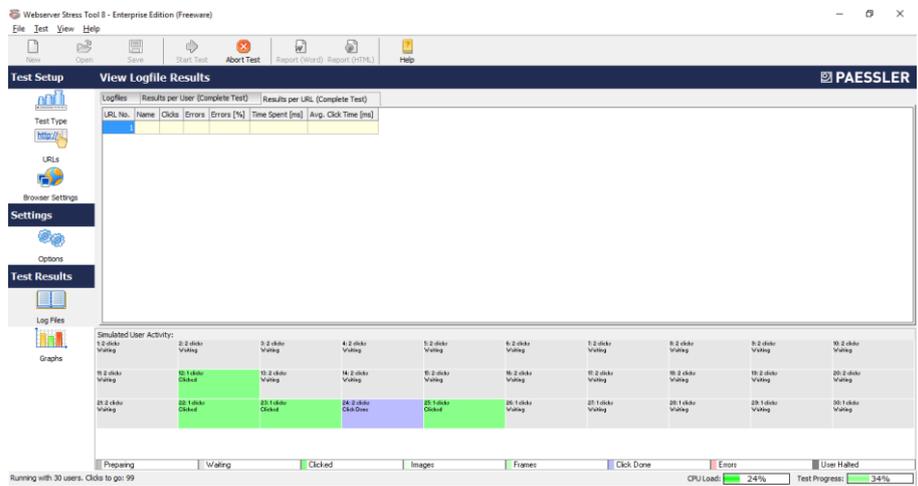
## 2. Click Tesing

Click Tesing digunakan untuk mengetes urutan dari URL aplikasi E-Vampire berbasis *website*. Pada Testing ini penguji menkonfigurasi Click Tesing sebagai berikut:

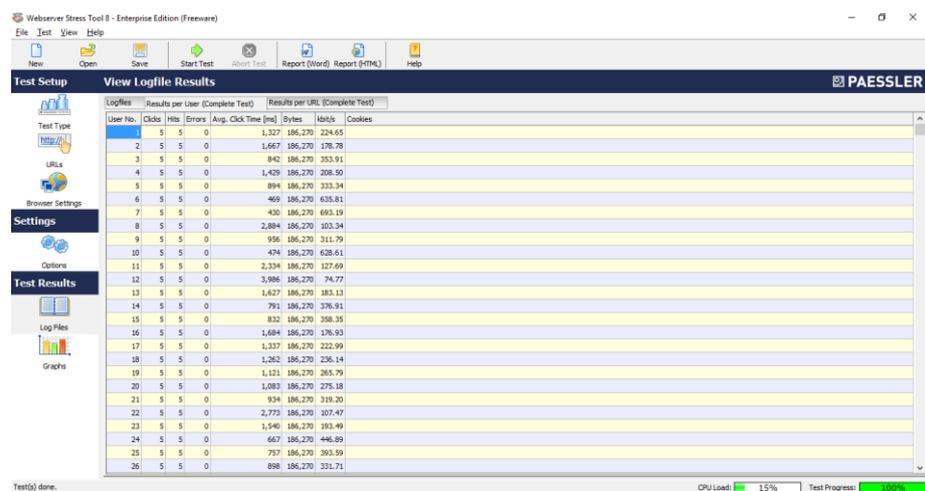
1. *Number of users: 30*
2. *Run until: 5 clicks per user*
3. *Click delay: 20 seconds*



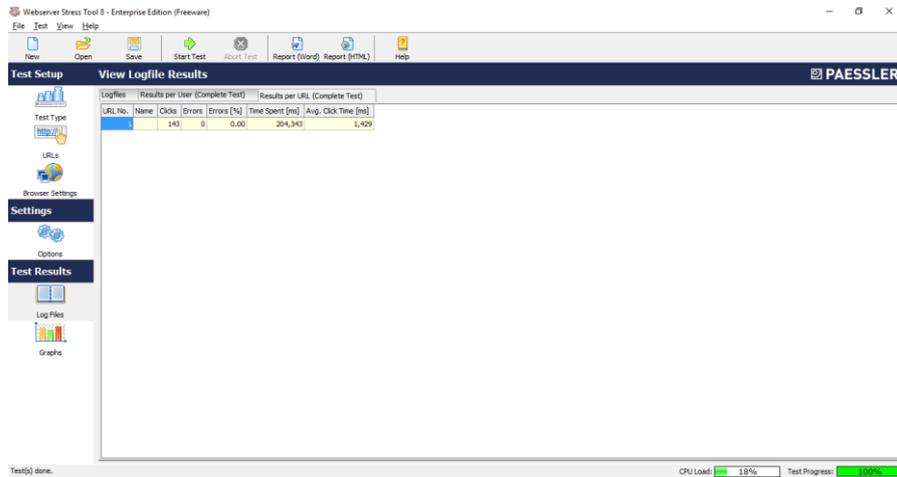
Gambar V.2.3. 3 Click Testing



Gambar V.2.3. 4 Click Testing



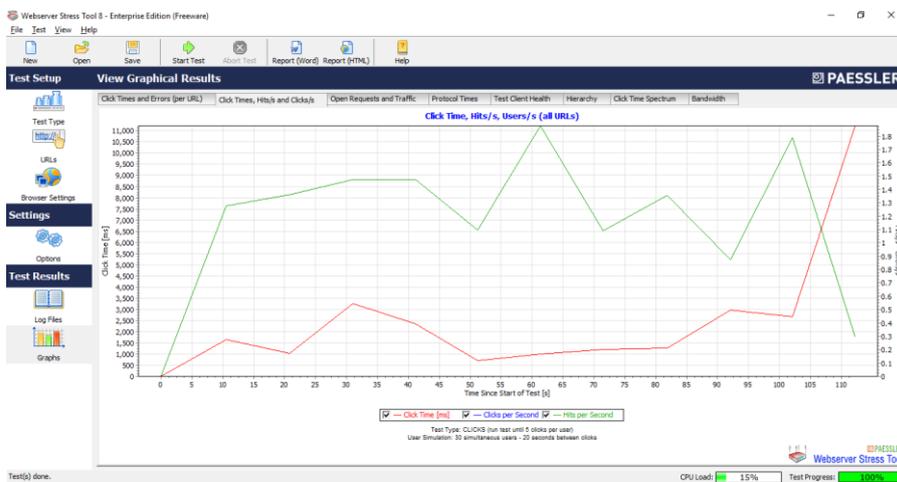
Gambar V.2.3. 5 Click Testing



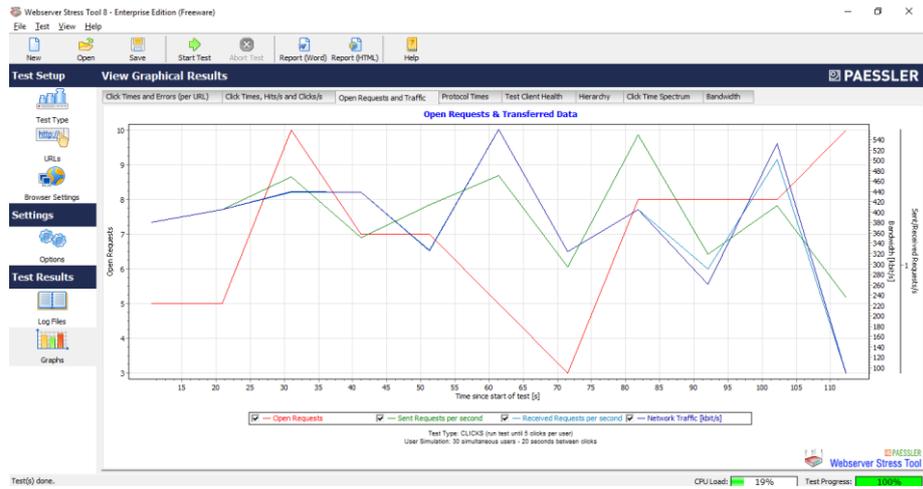
Gambar V.2.3. 6 Click Testing



Gambar V.2.3. 7 Click Testing



Gambar V.2.3. 8 Click Testing



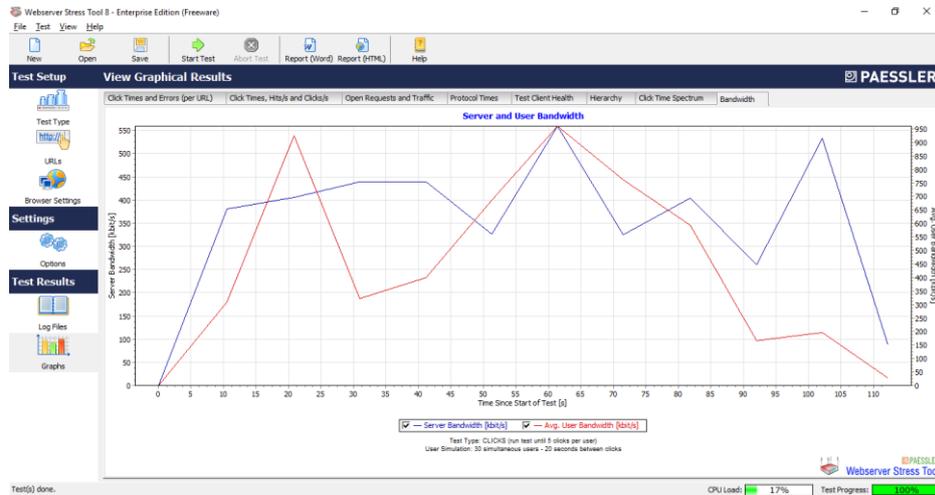
Gambar V.2.3. 9 Click Testing



Gambar V.2.3. 10 Click Testing



Gambar V.2.3. 11 Click Testing

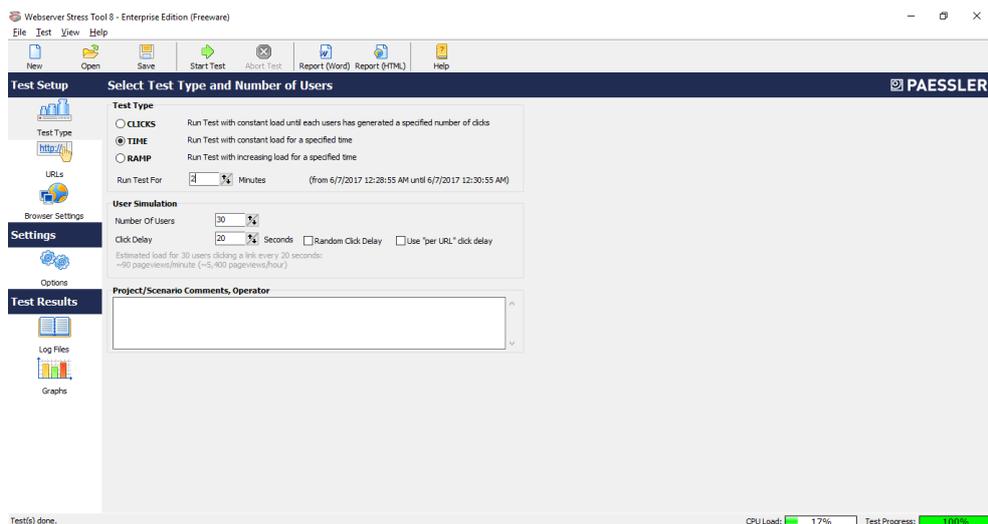


**Gambar V.2.3. 12 Click Testing**

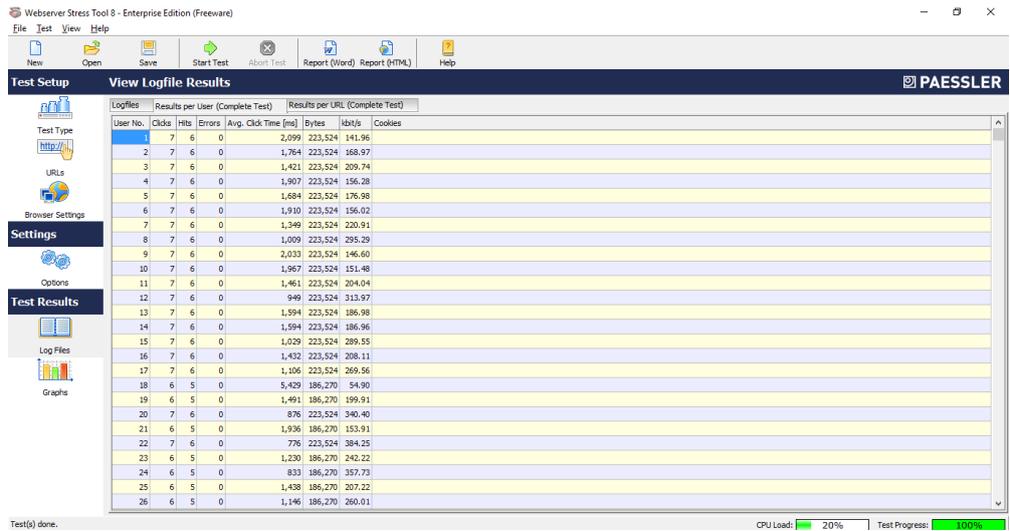
### 3. Time Testing

Time Tesing ini digunakan untuk mengetes server aplikasi E-Vampire berbasis website dalam jangka waktu yang ditentukan. Pada testing ini penguji menkonfigurasi sebagai berikut:

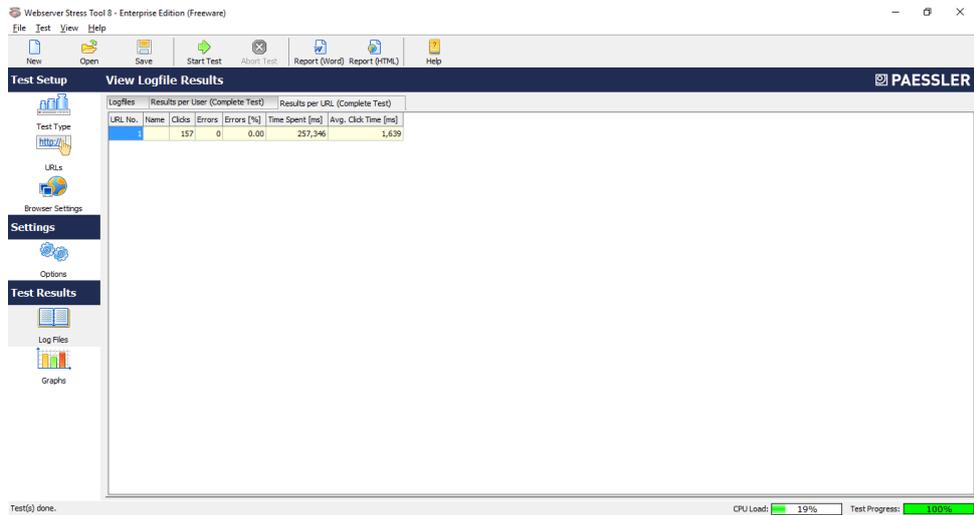
1. *Run test for: 2 minutes*
2. *Number of users: 30*
3. *Click delay: 20 seconds*



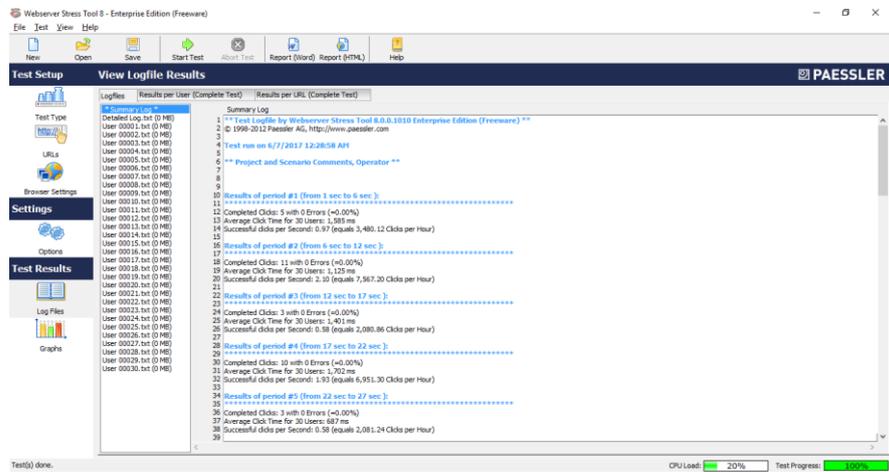
**Gambar V.2.3. 13 Time Testing**



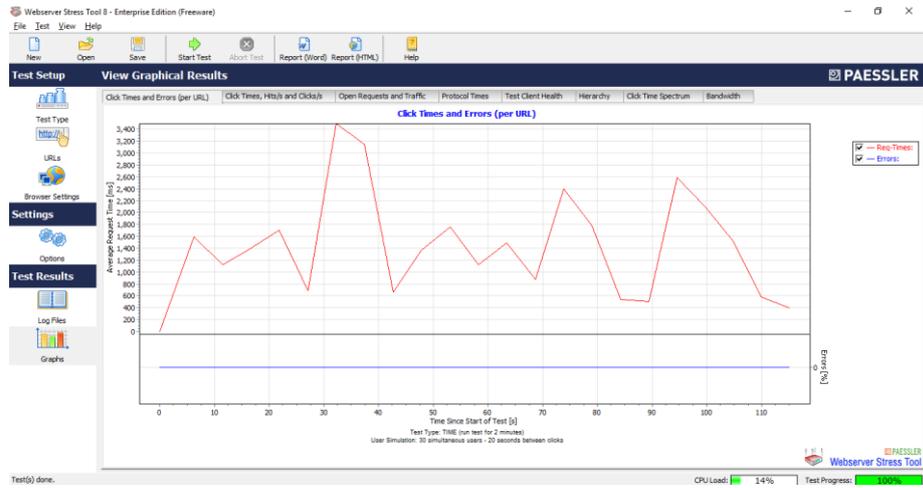
Gambar V.2.3. 14 Time Testing



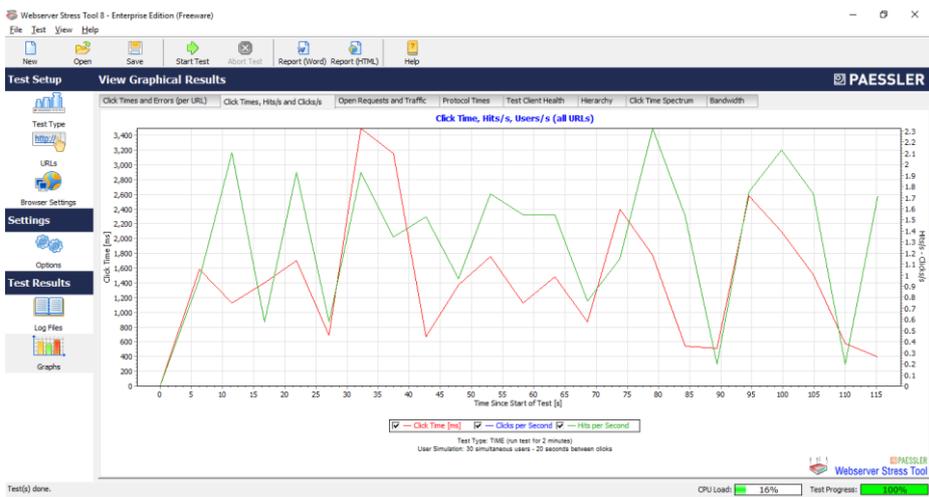
Gambar V.2.3. 15 Time Testing



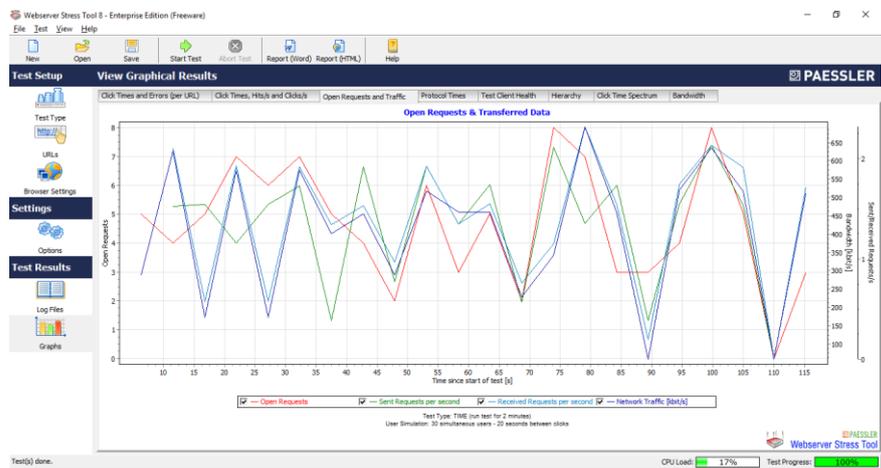
### Gambar V.2.3. 16 Time Testing



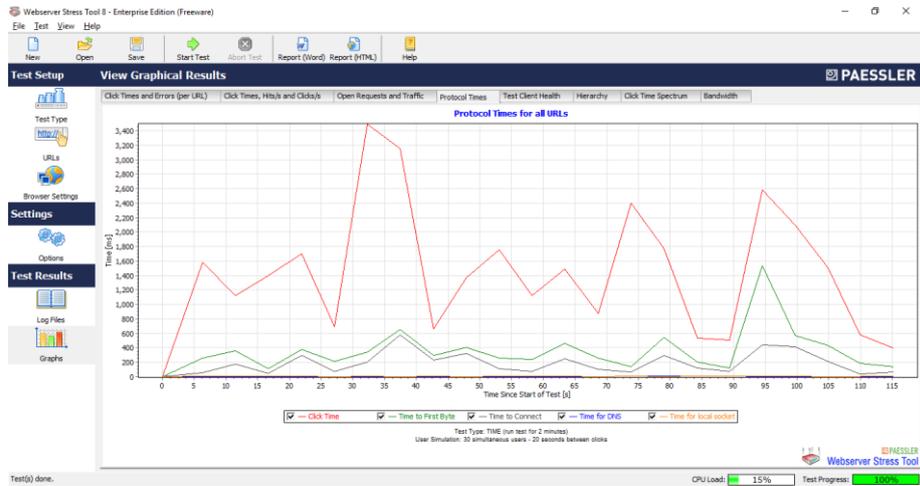
### Gambar V.2.3. 17 Time Testing



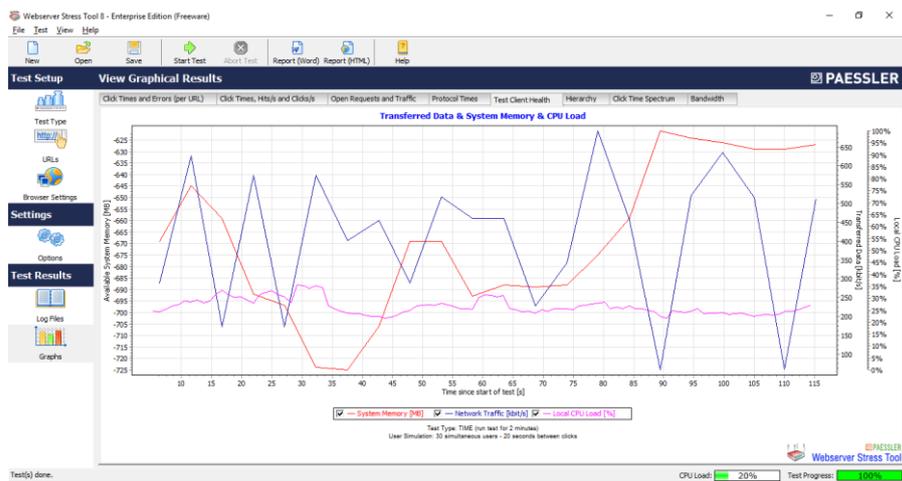
### Gambar V.2.3. 18 Time Testing



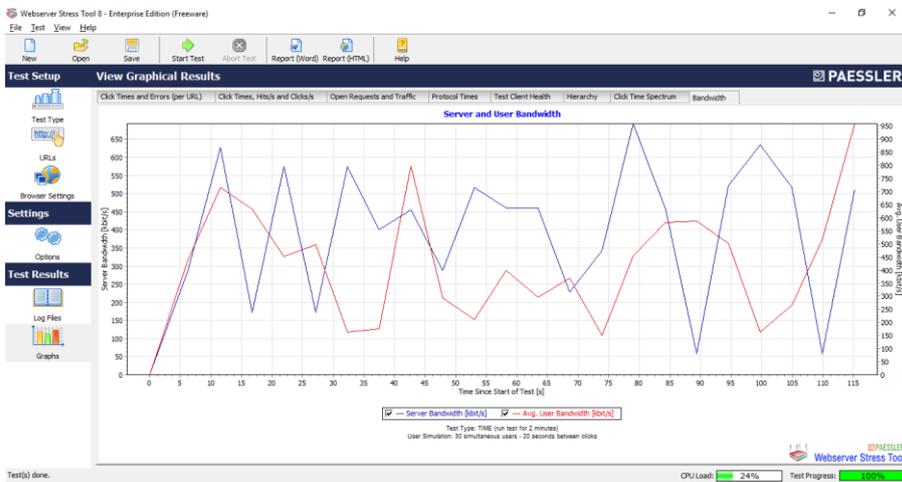
### Gambar V.2.3. 19 Time Testing



### Gambar V.2.3. 20 Time Testing



### Gambar V.2.3. 21 Time Testing

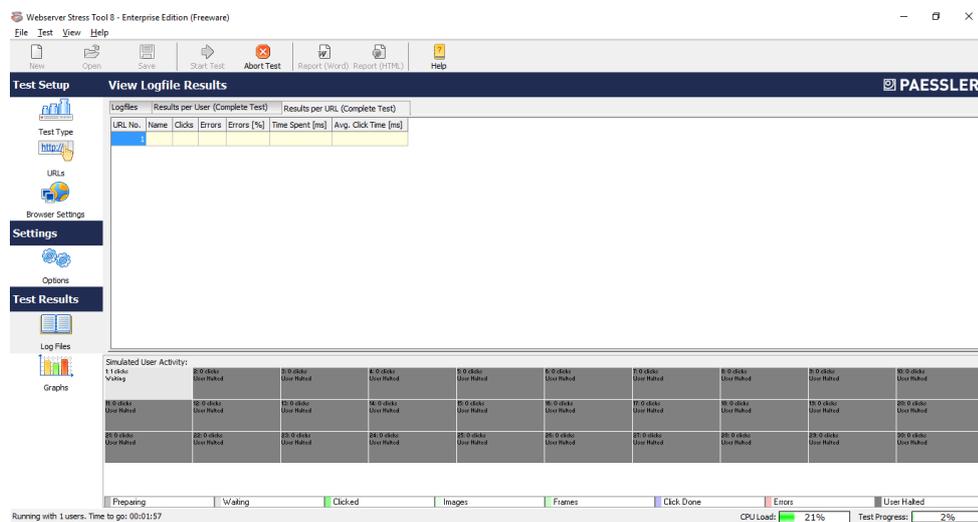


**Gambar V.2.3. 22 Time Testing**

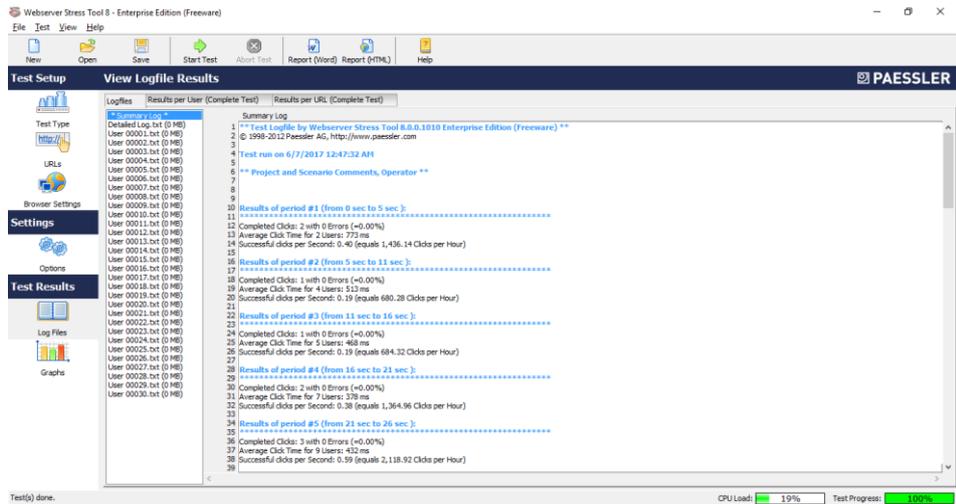
#### 4. Ramp Testing

Ramp testing digunakan penguji untuk menguji aplikasi E-Vampire berbasis website dalam menguji beban server yang digunakan aplikasi E-Vampire. Pengujian ini dapat memberikan seberapa batas beban yang dapat server aplikasi E-Vampire terima. Penguji mengkonfigurasi dan mendapatkan hasil sebagai berikut

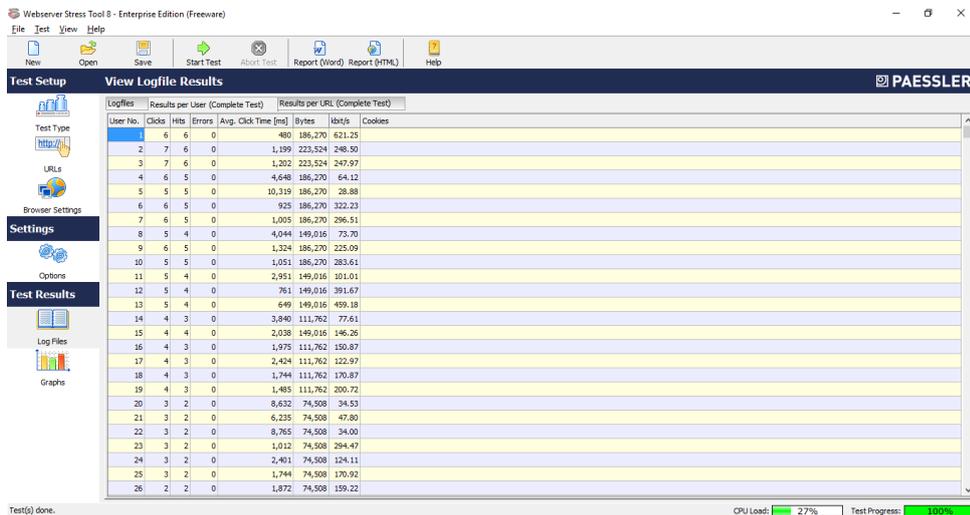
1. *Run test for: 2 minutes*
2. *Number of users: 30*
3. *Click delay: 20 seconds*



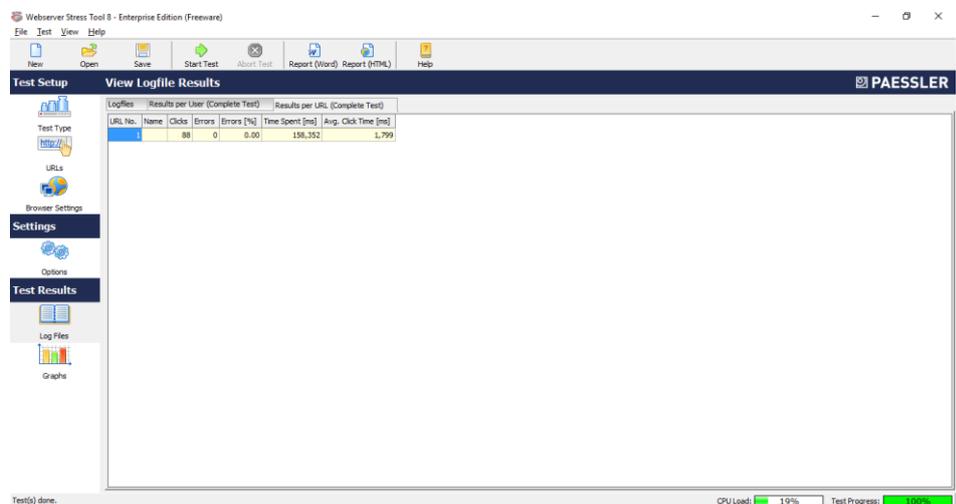
**Gambar V.2.3. 23 Ramp Testing**



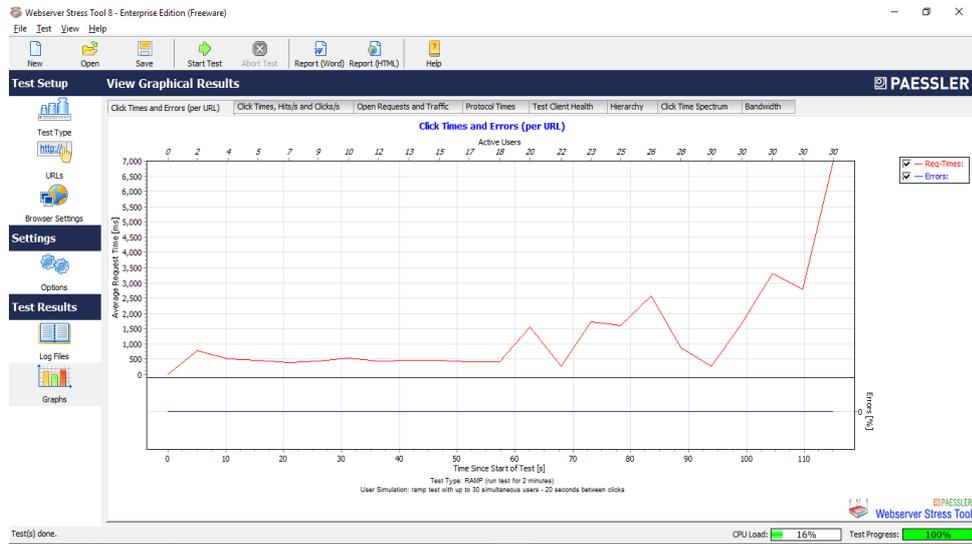
Gambar V.2.3. 24 Ramp Testing



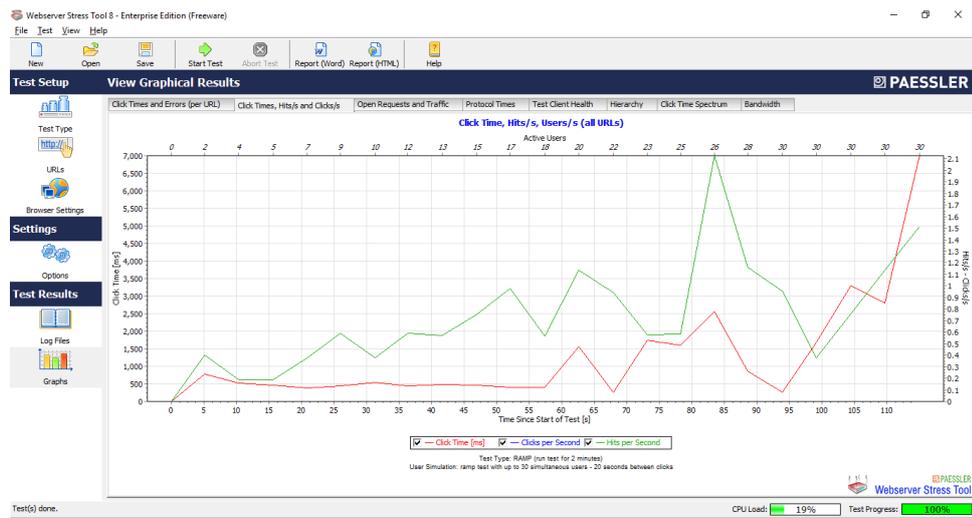
Gambar V.2.3. 25 Ramp Testing



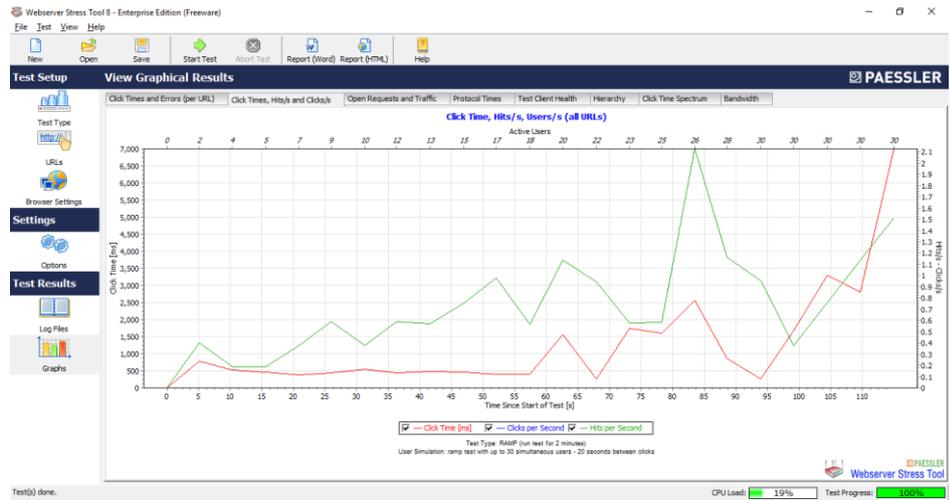
### Gambar V.2.3. 26 Ramp Testing



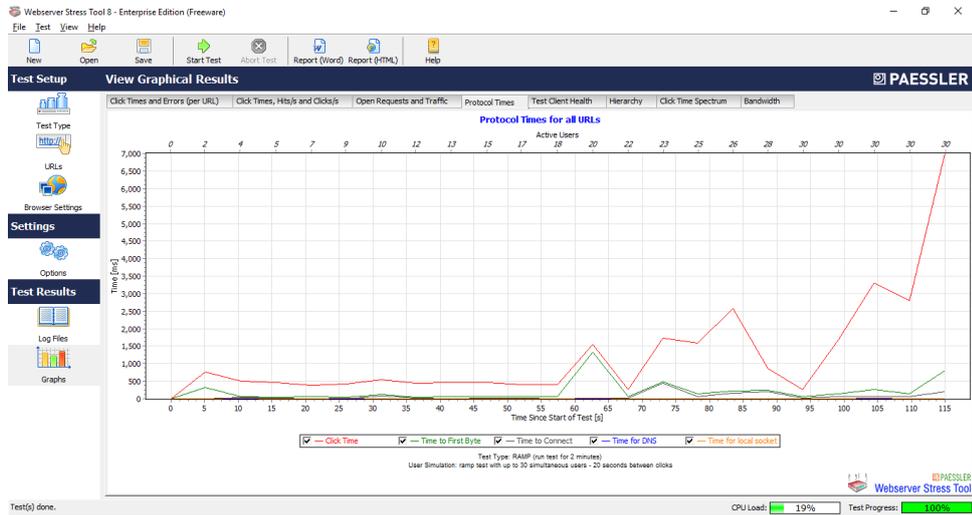
### Gambar V.2.3. 27 Ramp Testing



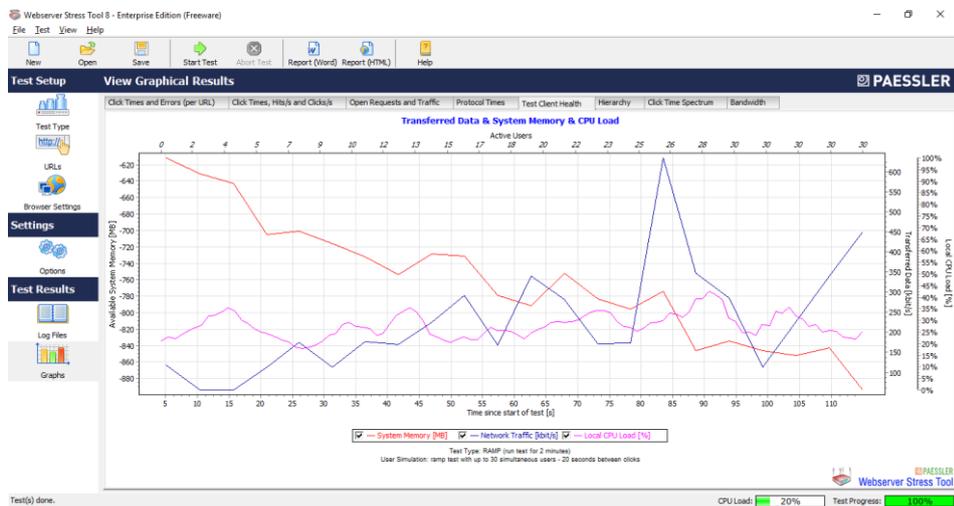
### Gambar V.2.3. 28 Ramp Testing



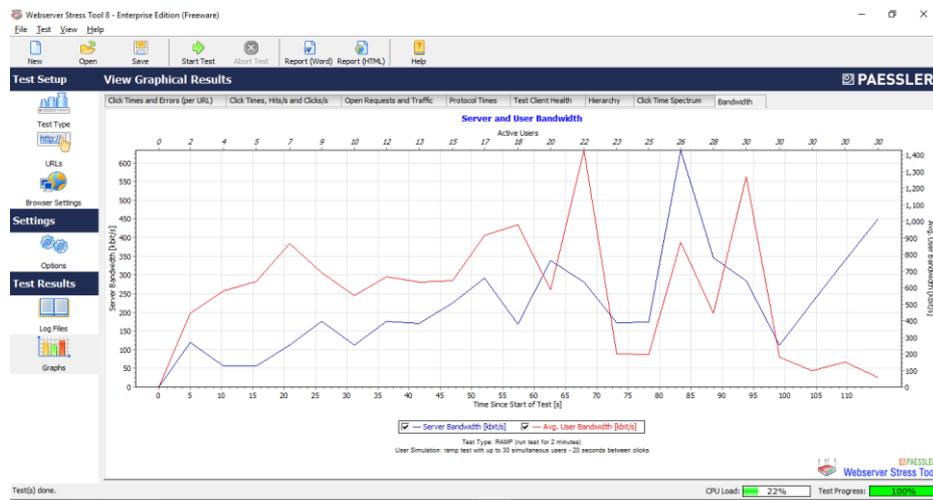
Gambar V.2.3. 29 Ramp Testing



Gambar V.2.3. 30 Ramp Testing



**Gambar V.2.3. 31 Ramp Testing**



**Gambar V.2.3. 32 Ramp Testing**

Setelah melakukan stress test diatas maka penguji mendapat keimpulan dengan *requirement* yang telah dibuat oleh penguji server dari aplikasi E-Vampire mampu menerima beban yang disebabkan requirement oleh penguji.

#### **V.2.4. User Acceptance Testing**

*User acceptance testing* atau pengujian penerimaan pengguna dilakukan dengan metode kuesioner. Kuesioner dilakukan untuk mengukur tingkat kenyamanan pengguna dalam menggunakan aplikasi E-Vampire berbasis *website* dan diterima atau tidaknya oleh pengguna. Berikut daftar pertanyaan yang diajukan kepada responden:

**Tabel V.2.4.1 Pertanyaan Kuesioner**

Kode	Pertanyaan
Q1	Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan?
Q2	Apakah antarmuka aplikasi mudah dipelajari?
Q3	Apakah komponen dan warna yang digunakan konsisten?
Q4	Apakah menu – menu yang terdapat di aplikasi dapat dimengerti dengan baik?
Q5	Apakah aplikasi dapat diakses dengan baik?

Q6	Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna?
Q7	Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi?
Q8	Apakah aplikasi memberikan umpan balik dari setiap aksi yang diberikan?
Q9	Apakah aplikasi memberikan nilai tambah kepada penggunanya?
Q10	Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbarui dikemudian hari?

Berdasarkan pertanyaan yang telah disusun, dapat dilihat umpan balik yang diberikan oleh responden sebagai berikut:

**Tabel V.2.4.2 Umpan Balik Responden**

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	4	5	4	4	5	5	3	3	4	4
2	5	4	4	4	3	5	5	5	5	5
3	4	4	4	4	3	5	5	3	4	5
4	5	4	5	4	4	4	4	4	5	5
5	5	4	4	5	5	5	5	5	4	4
6	3	4	5	5	4	3	5	4	4	5
7	5	4	5	4	5	5	5	5	5	5
8	5	5	5	4	4	5	4	5	3	5
9	3	5	5	3	5	5	3	5	3	5
10	4	5	4	5	4	5	5	5	5	5

Perhitungan dilakukan untuk mendapatkan total penilaian yang telah diberikan oleh responden. Bobot penilaian dapat dilihat sebagai berikut:

**Tabel V.2.4.3 Bobot Penilaian**

Kategori Penilaian	Bobot
--------------------	-------

Sangat Setuju (SS)	5
Setuju (S)	4
Cukup Setuju (CS)	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju (STS)	1

Proses perhitungan data dilakukan dengan cara menghitung persentase dari setiap pertanyaan yang diajukan, perhitungan data dapat dilihat sebagai berikut:

**Tabel V.2.4.4 Perhitungan Data**

Kode	Umpan Balik Responden									
	S	%	S	%	C	%	T	%	ST	%
	S		S		S		S		S	
Q1	3	30	5	50	2	20	0	0	0	0
Q2	5	50	3	30	2	20	0	0	0	0
Q3	3	30	5	50	2	20	0	0	0	0
Q4	4	40	6	60	0	0	0	0	0	0
Q5	6	60	4	40	0	0	0	0	0	0
Q6	4	40	4	40	2	20	0	0	0	0
Q7	8	80	2	20	0	0	0	0	0	0
Q8	6	60	3	30	1	10	0	0	0	0
Q9	6	60	0	0	4	40	0	0	0	0
Q10	7	70	3	30	0	0	0	0	0	0
Tot al	5 2	520	3 5	350	1 3	130	0	0	0	0

Perhitungan yang dilakukan menggunakan skala *likert* untuk mengetahui apakah aplikasi yang ditampilkan dapat diterima oleh pengguna. Setelah mendapat data umpan balik dari responden, data tersebut diolah untuk mendapatkan total skor, skor terendah, dan skor tertinggi.

#### **Rumus Perhitungan**

$$\begin{aligned} \text{Total Skor} &= (\Sigma SS \times 5) + (\Sigma S \times 4) + (\Sigma CS \times 3) + (\Sigma TS \times 2) \\ &+ (\Sigma STS \times 1) \end{aligned}$$

$$\text{Skor Terendah (X)} = \Sigma \text{Skor terendah likert} \times \text{jumlah responden}$$

$$\text{Skor Tertinggi (Y)} = \Sigma \text{Skor tertinggi} \times \text{jumlah responden}$$

Dengan menggunakan rumus yang telah dijelaskan di atas, maka didapatkan hasil sebagai berikut:

$$\text{Total Skor} = (52 \times 5) + (35 \times 4) + (13 \times 3) + (0 \times 2) + (0 \times 1) = 439$$

$$\text{Skor Terendah (X)} = 0 \times 10 = 0$$

$$\text{Skor Tertinggi (Y)} = 52 \times 10 = 520$$

Tahap selanjutnya adalah mengetahui skala interval dengan satuan persen (%) agar dapat diketahui bahwa aplikasi dapat dikatakan baik atau tidak.

#### **Rumus Interval**

$$I = \frac{100 \%}{\text{Jumlah Kategori (Likert)}}$$

Dengan menggunakan rumus yang telah dijelaskan di atas, maka didapatkanlah jarak interval untuk mengukur hasil pengujian sebagai berikut:

$$I = \frac{100 \%}{5} = 20 \%$$

Berdasarkan hasil jarak interval dan dengan nilai terendah 0 % dan nilai tertinggi 100 %, skala interval penilaian dapat dilihat sebagai berikut:

**Tabel V.2.4.5 Interval Penilaian**

Skala	Nilai (%)
Sangat Baik	81 - 100
Baik	61 - 80
Rata - rata	41 - 60
Kurang Baik	21 - 40
Buruk	0 - 20

Tahap terakhir adalah menghitung skor interpretasi, lalu hasil perhitungan tersebut dibandingkan dengan tabel interval penilaian sehingga dapat ditarik kesimpulan bagaimana nilai yang didapatkan dari aplikasi.

#### **Rumus Skor Interpretasi**

$$Skor Interpretasi = \frac{Total Skor}{Y} \times 100\%$$

Dengan menggunakan rumus yang telah dijelaskan di atas, maka didapatkanlah skor interpretasi sebagai berikut:

$$Skor Interpretasi = \frac{439}{520} \times 100\% = 84,4 \%$$

Berdasarkan hasil skor interpretasi sebesar 84,4 %, dapat diambil kesimpulan aplikasi dapat dikatakan termasuk kategori sangat baik. Jadi aplikasi E-Vampire berbasis *website* yang diberikan dapat diterima dengan baik oleh pengguna.

## **Bab VI KESIMPULAN DAN SARAN**

### **VI.1. Kesimpulan**

Kesimpulan yang dapat diambil dari pengembangan aplikasi E-Vampire adalah:

1. Fitur Event, Vam Playground, dan Vam House diharapkan dapat meningkatkan *awareness* masyarakat pada aplikasi E-Vampire berbasis *website*.
2. *Start-up* E-Vampire menggunakan *Lean Canvas* sebagai pendukung perancangan bisnis *start-up*.

### **VI.2. Saran**

Untuk pengembangan selanjutnya, aplikasi E-Vampire berbasis *website* sebaiknya ditambahkan beberapa fitur lainnya seperti Facebook *Login*, komentar dan berbagi ke media sosial, dan tampilan yang lebih menarik sehingga setiap pengguna merasa nyaman saat menggunakan, membantu segala kebutuhan dalam donor darah dan memberikan informasi – informasi edukatif yang dapat membantu menambah kesadaran akan pentingnya donor darah.

## DAFTAR PUSTAKA

- iNFODATIN. (n.d.). *SITUASI DONOR DARAH DI INDONESIA*.
- Kadir, A. (2008). *Tuntunan Praktis: Belajar Database Menggunakan MySQL*. Yogyakarta: Andi.
- Kasman, A. D. (2013). *Kolaborasi Dahsyat Android dengan PHP dan MYSQL*. Yogyakarta: Lokomedia.
- Kathy Schwalbe, P. P. (n.d.). *Information Technology Project Management | 7e*. Augsburg , Jerman: Course Technology.
- Menhennett, A. (2014). *A Guide to HTML5 and CSS3*. Australia: Zenva.
- RAHMAYADI, A. P. (2016). *PERANCANGAN USER INTERFACE DASHBOARD APLIKASI*. BANDUNG: RAHMAYADI, A. P.
- SCRUM ALLIANCE®, I. (2016). *Scrum Value* . Retrieved Form Scrum Alliance: <https://www.scrumalliance.org/why-scrum/core-scrum-values-roles>
- ScrumInc., S. a. (2014). *The Scrum Guide*. Retrieved Form Scrum Alliance : <https://www.scrumalliance.org/why-scrum/scrum-guide>
- SN, R. (2012). *IMPLEMENTASI SCRUM PADA PENGEMBANGAN SOFTWARE TERDISTRIBUSI*.
- Jan Jonker, Bartjan J.W. Pennink, Sari Wahyuni. 2011. *Metodologi Penelitian. Panduan Untuk Master Ph.D di bidang Manajemen*. Jakarta : Salemba Empat (Jan Jonker, 2011)
- Kementrian Kesehatan Republik Indonesia. (2016). *Ketersediaan Darah Ditentukan Partisipasi Masyarakat Menjadi Donor*. Jakarta: Kementrian Kesehatan Republik Indonesia.
- Martin, R. C. (2002). *UML for Java Programmers*. New Jersey: Alan Apt.
- Fowler, M. (2004). *A Brief Guide TO The Standard Object Modeling Language*. Massachusetts: Addison Wesley.
- Maurya, A. (2012). *Running Lean A Systematic Process For Iterating Your Web Application Form Plan A To A Plan That Works*. Texas: O'Reilly Media.
- Munawir, S. (2007). *Analisis Laporan Keuangan*. Yogyakarta: Liberty.

Statista. (2017, Mei 31). *Indonesia: Total population Form 2010 to 2020 (in million inhabitants)*. Diambil kembali dari Statista:

<https://www.statista.com>

Statista. (2017, Mei 31). *Number of smartphone users in Indonesia Form 2011 to 2021 (in millions)*. Diambil kembali dari Statista: <https://www.statista.com>

Badan Pusat Statistik. (2017, Juni 06). *Badan Pusat Statistik Tabel Dinamis*.

Diambil kembali dari Badan Pusat Statistik:

<https://www.bps.go.id/site/resultTab>

Bhasin, H. (2014). Black Box Testing. *Black Box Testing Based On Requirement Analysis And Design Specification*.

## LAMPIRAN A PROYEKSI KEUANGAN

### A.1. Proyeksi Pendapatan

**Tabel A.1.1 Proyeksi Pendapatan Iklan Web Apps**

Tahun	Harga	Keuntungan Per bulan	Keuntungan Per Tahun
1	IDR 400.000	IDR 1.200.000	IDR 14.400.000
2	IDR 600.000	IDR 1.800.000	IDR 21.600.000
3	IDR 800.000	IDR 2.400.000	IDR 28.800.000

Keterangan:

Jumlah iklan pada aplikasi web berjumlah tiga iklan.

**Tabel A.1.2 Proyeksi Pendapatan Admob Mobile Apps**

Tahun	Jumlah User	Keuntungan Per bulan	Keuntungan Per Tahun
1	1.000	IDR 840.000	IDR 10.080.000
2	3.000	IDR 2.520.000	IDR 30.240.000
3	5.000	IDR 4.200.000	IDR 50.400.000

Keterangan:

- 1) Jumlah iklan pada aplikasi *mobile* berjumlah sepuluh iklan.
- 2) Biaya per klik = 0,006 \$ = IDR 84
- 3) Kalkulasi pendapatan per bulan (tahun 1):

$$\begin{aligned} \text{Pendapatan Per Bulan} &= \text{User} \times \text{Jumlah Iklan} \times \text{Biaya Per klik} \\ \text{Pendapatan Per Bulan} &= 1000 \times 10 \times 84 = 840.000 \end{aligned}$$

- 4) Kalkulasi pendapatan per tahun (tahun 1):

$$\text{Pendapatan Per Tahun} = 840.000 \times 12 \text{ bulan} = 10.080.000$$

**Tabel A.1.3 Proyeksi Pendapatan Google AdSense Web Apps**

Tahun	Jumlah User Per Acara	Keuntungan Per Tahun
1	40	IDR 9.676.800
2	100	IDR 24.192.000
3	180	IDR 43.545.600

Keterangan:

- 1) Jumlah acara per tahun = 144 Acara
- 2) Jumlah rata – rata klik per *user* = 20 klik
- 3) Biaya per klik = 0,006 \$ = IDR 84
- 4) Kalkulasi pendapatan per tahun (tahun 1):

$$\begin{aligned} \text{Pendapatan Per Tahun} &= \text{Jumlah Acara} \times \text{User Per Acara} \times \text{Klik Per User} \times \text{Biaya Per klik} \\ \text{Pendapatan Per Tahun} &= 144 \times 40 \times 20 \times 84 = 9.676.800 \end{aligned}$$

**Tabel A.1.4 Proyeksi Pendapatan Youtube**

Tahun	Target Viewer	Keuntungan Per Tahun
1	1.000	IDR 12.096.000
2	2.000	IDR 24.192.000
3	2.500	IDR 30.240.000

Keterangan:

- 1) Jumlah acara per tahun = 144 Acara
- 2) Biaya per *view* = 0,006\$ = IDR 84
- 3) Kalkulasi pendapatan per tahun (tahun 1):

$$\text{Pendapatan Per Tahun} = \text{Jumlah Acara} \times \text{Viewer} \times \text{Biaya Per klik}$$

$$\text{Pendapatan Per Tahun} = 144 \times 1000 \times 84 = 12.096.00$$

**Tabel A.1.5 Proyeksi Pendapatan Delivery Darah**

Tahun	Transaksi Per bulan	Keuntungan Per bulan	Keuntungan Per Tahun
2	30 Transaksi	IDR 2.700.000	IDR 32.400.000
3	100 Transaksi	IDR 9.000.000	IDR 108.000.000

Keterangan:

- 1) Keuntungan 1 kantung = IDR 30.000
- 2) Keuntungan 1 transaksi = IDR 90.000
- 3) 1 Transaksi minimal 4 kantung dan keuntungan 1 kantung diberikan kepada pengantar.
- 4) *Delivery* darah mulai direncanakan dilakukan pada tahun ke-2.
- 5) Kalkulasi pendapatan per bulan (tahun 1):

$$\text{Pendapatan Per Bulan} = \text{Keuntungan 1 Transaksi} \times 30 \text{ Hari}$$

$$\text{Pendapatan Per Bulan} = 90.000 \times 30 = 2.700.000$$

- 6) Kalkulasi pendapatan per tahun (tahun 1):

$$\text{Pendapatan Per Tahun} = 2.700.000 \times 12 \text{ Bulan} = 32.400.000$$

## A.2. Proyeksi Pengeluaran

**Tabel A.2 Proyeksi Pengeluaran**

Biaya Lain - Lain		
Tahun 1	Biaya Pembuatan CV =	IDR 5,000,000
	Biaya Kendaraan =	IDR 5,000,000
	Biaya Sewa Tempat =	IDR 5,000,000
	Biaya Peralatan Kantor =	IDR 5,000,000
	Total =	IDR 20,000,000
Tahun 2	Biaya Pembuatan PT =	IDR 8,000,000

	Biaya Kendaraan =	IDR 10,000,000
	Biaya Sewa Tempat =	IDR 8,000,000
	Biaya Peralatan Kantor =	IDR 5,000,000
	Biaya Gaji =	IDR 72,000,000
	Total =	IDR 103,000,000
Tahun 3	Biaya Kendaraan =	IDR 15,000,000
	Biaya Sewa Tempat =	IDR 8,000,000
	Biaya Peralatan Kantor =	IDR 5,000,000
	Biaya Gaji =	IDR 72,000,000
	Total =	IDR 100,000,000

Keterangan:

- 1) Biaya gaji per pegawai = IDR 3.000.000
- 2) Pada tahun kedua dan ketiga *start-up* aplikasi E-Vampire memiliki dua orang pegawai
- 3) Kalkulasi biaya gaji per tahun:

$$\text{Biaya Gaji Per Tahun} = 3.000.000 \times 2 \text{ orang} \times 12 \text{ bulan} = 72.000.000$$

## LAMPIRAN B SURVEI

### B.1. Form Survey

### E-Vampire

---

Halo, kami dari startup E-Vampire dalam rangka mengerjakan tugas akhir dalam perancangan aplikasi E-Vampire membutuhkan responden dalam validasi pasar. Evampire.id merupakan aplikasi medikal yang mempunyai fitur pencarian darah darurat untuk pasien kebutuhan darah, pencarian tempat donor darah maupun bank darah terdekat, informasi stok darah dan konten edukasi donor darah. Terima Kasih.

---

**Name \***

First Name Last Name

**E-mail \***

**Tanggal Lahir \***

-  -

**Gambar B.1 Form Survey 1**

Apakah anda pernah melakukan donor darah? \*

- YA  
 TIDAK

Apakah anda mengetahui jika Indonesia mengalami kekurangan ketersediaan darah? \*

- YA  
 TIDAK

Alasan anda belum pernah melakukan donor darah?

isi jika anda belum pernah melakukan donor darah, lanjut pada sesi pernyataan

Alasan anda melakukan donor darah?

isi jika anda sudah pernah melakukan donor darah

**Gambar B.1 Form Survey 2**

Sudah berapa kali anda melakukan donor darah?

ex: 23

Dimana anda biasa melakukan donor darah

- PMI
- Event mingguan organisasi
- Event kampus
- Other

Darimana anda mendapat informasi donor darah?

- PMI
- Teman
- Media Sosial
- Other

good people, setiap pernyataan di bawah dapat dijawab dengan memilih tingkat tidak setuju hingga setuju pada pernyataan yang tertera. Setiap angkanya berarti 1 = Sangat Tidak Setuju | 2= Tidak Setuju | 3= Biasa | 4= Setuju | 5= Sangat Setuju

### Gambar B.1 Form Survey 3

Pengetahuan tentang kurangnya ketersediaan darah serta donor darah mempengaruhi saya dalam melakukan donor darah \*

1 2 3 4 5

Tidak setuju      Setuju

Edukasi donor darah dibutuhkan dalam meningkatkan minat masyarakat dalam donor darah. \*

1 2 3 4 5

Tidak setuju      Setuju

Saya mengalami kesulitan dalam mencari informasi tempat donor darah dan bank darah \*

1 2 3 4 5

Tidak setuju      Setuju

Saya mengalami kesulitan mencari Informasi stok darah \*

1 2 3 4 5

Tidak setuju      Setuju

### Gambar B.1 Form Survey 4

Pencarian darah darurat masih sulit dilakukan \*

1 2 3 4 5

Tidak setuju      Setuju

Untuk memudahkan pencarian informasi terkait donor darah hadirnya aplikasi berbasis web akan membantu pendonor. \*

1 2 3 4 5

Tidak setuju      Setuju

Untuk memudahkan pencarian informasi terkait donor darah hadirnya aplikasi berbasis mobile akan membantu pendonor. \*

1 2 3 4 5

Tidak setuju      Setuju

Aplikasi yang menyediakan fitur pencarian darah darurat untuk pasien kebutuhan darah akan membantu pasien kekurangan darah. \*

1 2 3 4 5

Tidak setuju      Setuju

### Gambar B.1 Form Survey 5

Aplikasi yang menyediakan informasi terkait tempat donor darah serta bank darah terdekat akan membantu pendonor. \*

1 2 3 4 5

Tidak setuju      Setuju

Aplikasi yang menyediakan fitur informasi stok darah akan membantu pendonor. \*

1 2 3 4 5

Tidak setuju      Setuju

Aplikasi yang berisikan fitur konten edukasi donor darah akan meningkatkan minat masyarakat dalam mendonasikan darahnya. \*

1 2 3 4 5

Tidak setuju      Setuju

Submit

### Gambar B.1 Form Survey 6

## B.2. Hasil Survei

**Tabel B.2 Nama Peserta Survei**

No.	Nama	No.	Nama
1	Ahmad Luthfi	30	Imam W
2	Ahmad Taufiq Hidayat	31	Indra Bayu
3	Amira Cita Muthia	32	Kafilia Vidai
4	Amri Hanif	33	Khairan Niza Mefid
5	Andhika Pratama	34	Lukas Laksamana
6	Annisa Aninditya	35	M.Zahid Syafnel
7	Annisa Azmi Abdullah	36	Muhammad Haekal Luthfi S
8	Benny Hutabarat	37	Naufal Afra Firdaus
9	Cut Vadea Aqila	38	Naufal Al-Hadid
10	Dea Rizky Pangestika	39	Naufal Bayhaqi
11	Deri Pran Kurniawan	40	Naufal Cahyo
12	Dias Fithry	41	Nida Nafisah
13	Dicky Prasetya	42	Puspita Sari
14	Dieta Driantama	43	Putri Dwi Aprilia
15	Dimas Bagus Prakoso	44	Rahmadina Arifazhari
16	Dini Indra Cahyani	45	Rahman
17	Dwi Paramitha	46	Rieke Amaluthvie
18	Fadhli Aulia S	47	Rivan Radian
19	Faisal Rapsanjani	48	Rizka Alifiani S
20	Faishal W Pradhana	49	Ruhimat Imat
21	Fajrin Pramestiningrum	50	Suci Wulanda Agusti
22	Farah Raniah	51	Syifa Kinasih
23	Ferryansa	52	Tasya Fauzia
24	Furqon Muhammad	53	Vian Rasyid Diyatma
25	Ginanjari Dewi Girang	54	Viqqy Rianda
26	Ginanjari Widya	55	Vita Dwi
27	Gita Desca Purnama	56	Wildan Faudzan Maulan Haryono
28	Gita Riesta	57	Wimbajaya Hamukti
29	Gustarina Nurfadilah	58	Yunita Prabowo

Pertanyaan 1. Apakah anda pernah melakukan donor darah?

Sebanyak 22 peserta pernah melakukan donor darah dan sebanyak 36 peserta tidak pernah melakukan donor darah.

Pertanyaan 2. Apakah anda mengetahui jika Indonesia mengalami kekurangan ketersediaan darah?

Sebanyak 40 peserta mengetahui Indonesia mengalami kekurangan ketersediaan darah dan 18 tidak mengetahui Indonesia mengalami kekurangan ketersediaan darah.

Pertanyaan 3. Alasan anda belum pernah melakukan donor darah?

Alasan peserta belum pernah melakukan donor darah yaitu takut jarum suntik, karena memiliki tekanan darah rendah, takut kekurangan darah atau anemia, berat badan, tidak berani, trauma, dan waktu yang belum tepat.

Pertanyaan 4. Alasan anda melakukan donor darah?

Alasan peserta melakukan donor darah yaitu ingin berbagi kepada yang membutuhkan, bagus untuk kesehatan, keinginan sesaat dan cek kesehatan gratis.

Pertanyaan 5. Sudah berapa kali anda melakukan donor darah?

Peserta telah melakukan donor darah rata – rata sebanyak 6 kali.

Pertanyaan 6. Dimana anda biasa melakukan donor darah?

Peserta biasa melakukan donor darah di PMI sebanyak 13 peserta , *event* kampus sebanyak 13 peserta dan *event* mingguan organisasi sebanyak 5 peserta.

Pertanyaan 7. Dari mana anda mendapat informasi donor darah?

Peserta mendapat informasi donor darah dari PMI sebanyak 14 peserta, media sosial sebanyak 10 peserta, dan teman sebanyak 13 peserta.

Pertanyaan 8. Pengetahuan tentang kurangnya ketersediaan darah serta donor darah mempengaruhi saya dalam melakukan donor darah.

Hasil jawaban :

1 = 1 peserta, 2 = 7 peserta, 3 = 19 peserta, 4 = 22 peserta, 5 = 9 peserta.

Pertanyaan 9. Edukasi donor darah dibutuhkan dalam meningkatkan minat masyarakat dalam donor darah.

Hasil jawaban:

1 = 0 peserta, 2 = 0 peserta, 3 = 12 peserta, 4 = 16 peserta, 5 = 30 peserta.

Pertanyaan 10. Saya mengalami kesulitan dalam mencari informasi tempat donor darah dan bank darah.

Hasil jawaban:

1 = 2 peserta, 2 = 6 peserta, 3 = 23 peserta, 4 = 18 peserta, 5 = 9 peserta.

Pertanyaan 11. Saya mengalami kesulitan mencari informasi stok darah.

Hasil jawaban:

1 = 0 peserta, 2 = 4 peserta, 3 = 15 peserta, 4 = 23 peserta, 5 = 16 peserta.

Pertanyaan 12. Pencarian darah darurat masih sulit dilakukan.

Hasil jawaban:

1 = 0 peserta, 2 = 0 peserta, 3 = 13 peserta, 4 = 26 peserta, 5 = 19 peserta.

Pertanyaan 13. Untuk memudahkan pencarian informasi terkait donor darah hadirnya aplikasi berbasis web akan membantu pendonor.

Hasil jawaban:

1 = 0 peserta, 2 = 0 peserta, 3 = 7 peserta, 4 = 11 peserta, 5 = 40 peserta.

Pertanyaan 14. Untuk memudahkan pencarian informasi terkait donor darah hadirnya aplikasi berbasis mobile akan membantu pendonor.

Hasil jawaban:

1 = 0 peserta, 2 = 0 peserta, 3 = 6 peserta, 4 = 15 peserta, 5 = 37 peserta.

Pertanyaan 15. Aplikasi yang menyediakan fitur pencarian darah darurat untuk pasien kebutuhan darah akan membantu pasien kekurangan darah.

Hasil jawaban:

1 = 0 peserta, 2 = 0 peserta, 3 = 7 peserta, 4 = 5 peserta, 5 = 46 peserta.

Pertanyaan 16. Aplikasi yang menyediakan informasi terkait tempat donor darah serta bank darah terdekat akan membantu pendonor.

Hasil jawaban:

1 = 0 peserta, 2 = 0 peserta, 3 = 9 peserta, 4 = 8 peserta, 5 = 41 peserta.

Pertanyaan 17. Aplikasi yang menyediakan fitur informasi stok darah akan membantu pendonor.

Hasil jawaban:

1 = 0 peserta, 2 = 0 peserta, 3 = 8 peserta, 4 = 12 peserta, 5 = 38 peserta.

Pertanyaan 18. Aplikasi yang berisikan fitur konten edukasi donor darah akan meningkatkan minat masyarakat dalam mendonasikan darahnya.

Hasil jawaban:

1 = 0 peserta, 2 = 0 peserta, 3 = 8 peserta, 4 = 14 peserta, 5 = 35 peserta.

## LAMPIRAN C DOKUMEN SCRUM

### C.1. Sprint 1

*Sprint 1* merupakan iterasi pertama dalam pengembangan aplikasi. Dalam tahap ini, ada empat fungsional yang akan dikembangkan yaitu Login, Registrasi, Logout dan Lupa Password.

#### C.1.1. Analisis

Analisis setiap *sprint* akan disesuaikan dengan *user stories* dan digambarkan menjadi *usecase diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

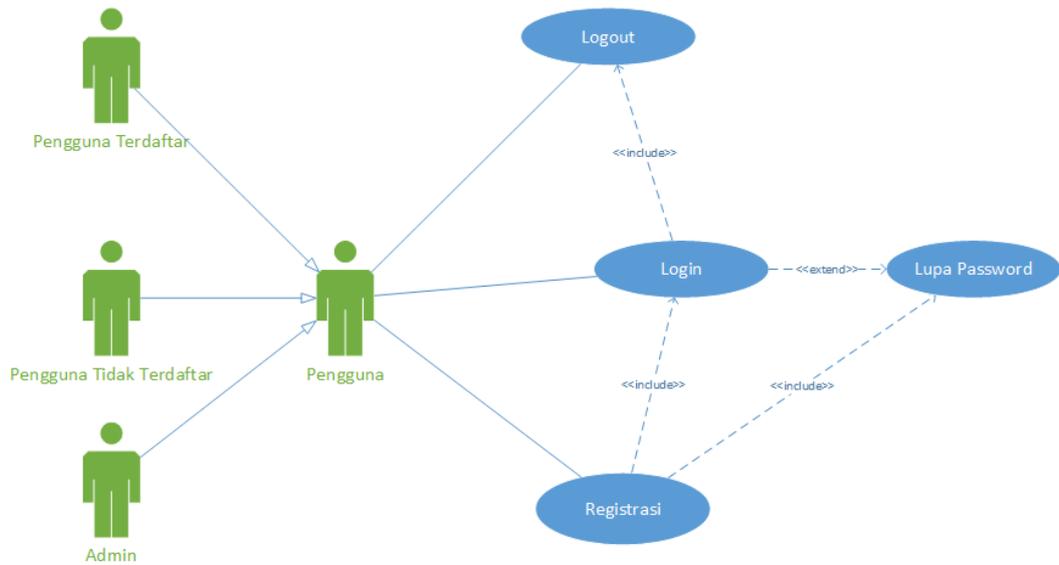
#### C.1.2. Sprint Backlog

Pada *sprint 1*, diambil tiga *product backlog* yang akan dijadikan *sprint backlog*. *Sprint backlog* tersebut yaitu:

*Tabel C.1.2 Sprint Backlog Sprint 1*

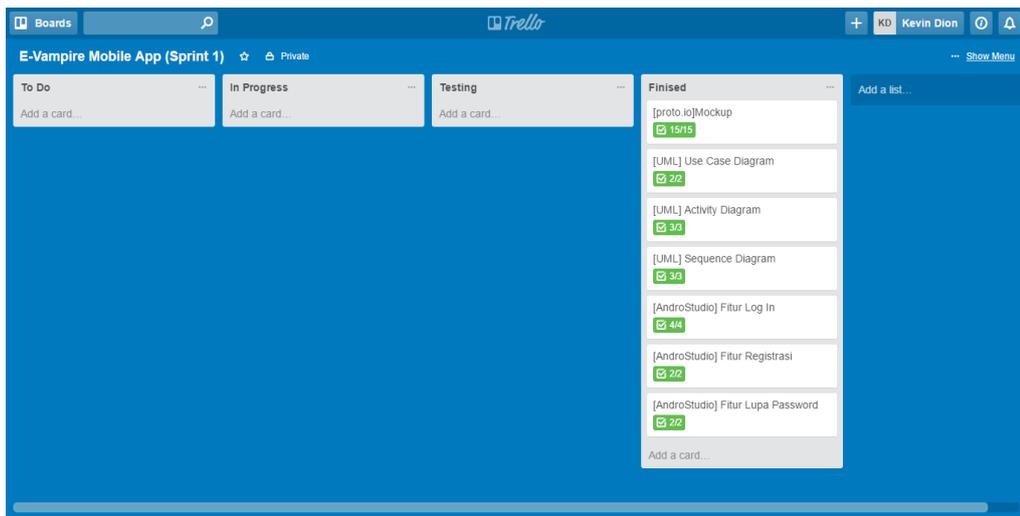
<i>User Story ID</i>	Saya ingin/butuh/dapat untuk	<i>Task</i>	Estimasi (Jam)
US-01	Saya dapat <i>login</i>	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
	Saya ingin login menggunakan akun media sosial	Implementasi <i>back-end code</i>	4
		<i>Testing</i>	3
US-02	Saya dapat melakukan registrasi	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-03	Saya dapat mengembalikan kata sandi	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-13	Saya dapat <i>logout</i>	Implementasi <i>back-end code</i>	1
		<i>Testing</i>	1
Total Jam Pengerjaan			36

#### C.1.1. Usecase Diagram



Gambar C.1.3 Usecase Diagram Sprint 1

### C.1.2. Sprint Board



Gambar C.1.4 Trello Sprint 1

## C.2. Sprint 2

*Sprint 1* merupakan iterasi kedua dalam pengembangan aplikasi. Dalam tahap ini, ada tiga fungsional yang akan dikembangkan yaitu Home, About, Contact, Vam House.

### C.2.1. Analisis

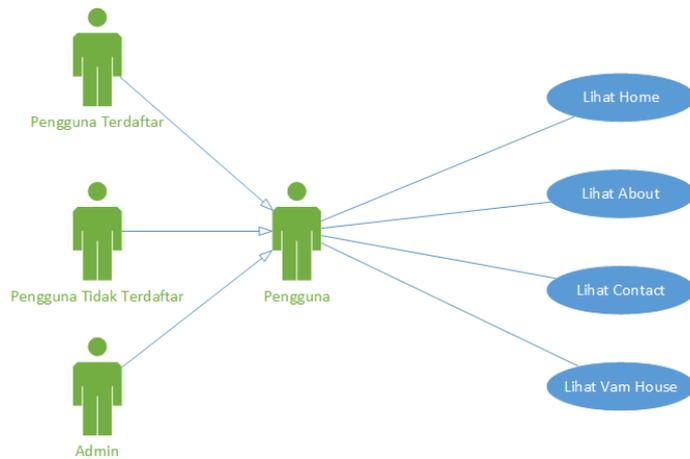
Analisis setiap *sprint* akan disesuaikan dengan *user stories* dan digambarkan menjadi *usecase diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

### C.2.2. Sprint Backlog

Tabel C.2.2 Sprint Backlog Sprint 2

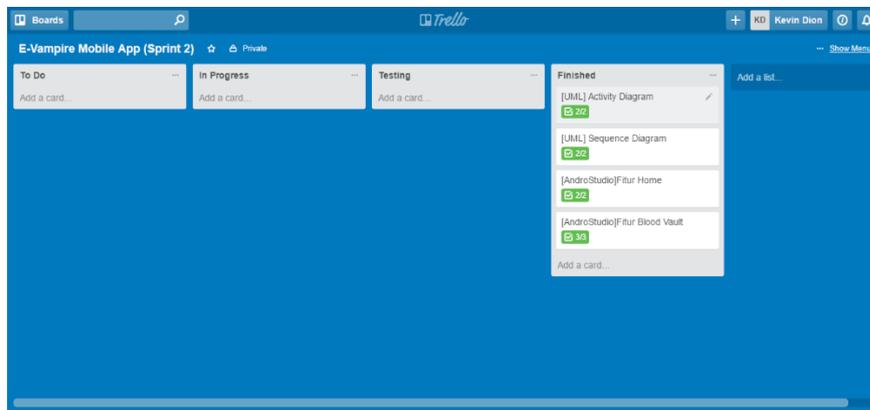
<i>User Story ID</i>	Saya ingin/butuh/dapat untuk	<i>Task</i>	Estimasi (Jam)
US-04	Saya dapat melihat tampilan Home	Desain <i>mockup</i>	2
		Desain konten	3
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-12	Saya ingin menghubungi pihak E-Vampire	Desain <i>mockup</i>	1
		Implementasi <i>front-end code</i>	2
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-09	Saya ingin melihat lokasi rumah sakit dan PMI	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
US-14	Saya dapat melihat informasi tentang aplikasi E-Vampire	Desain <i>mockup</i>	2
		Implementasi <i>front-end code</i>	3
		<i>Testing</i>	2
Total Jam Pengerjaan			41

### C.2.3. Usecase Diagram



Gambar C.2.3 Usecase Diagram Sprint 2

### C.2.4. Sprint Board



Gambar C.2.4 Trello Sprint 2

### C.3. Sprint 3

*Sprint 3* merupakan iterasi ketiga dalam pengembangan aplikasi. Dalam tahap ini, ada dua fungsional yang akan dikembangkan yaitu Vam Playground, Komentar, Permintaan Darah, dan Blood Vault.

#### C.3.1. Analisis

Analisis setiap *sprint* akan disesuaikan dengan *user stories* dan digambarkan menjadi *usecase diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

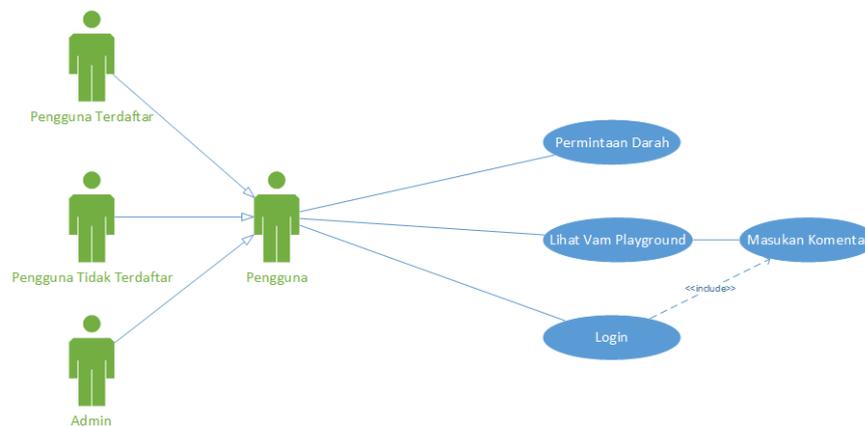
#### C.3.2. Sprint Backlog

Tabel C.3.2 Sprint Backlog Sprint 3

User Story ID	Saya ingin/butuh/dapat untuk	Task	Estimasi (Jam)
US-07	Saya ingin melihat informasi persediaan darah	Desain <i>mockup</i>	2

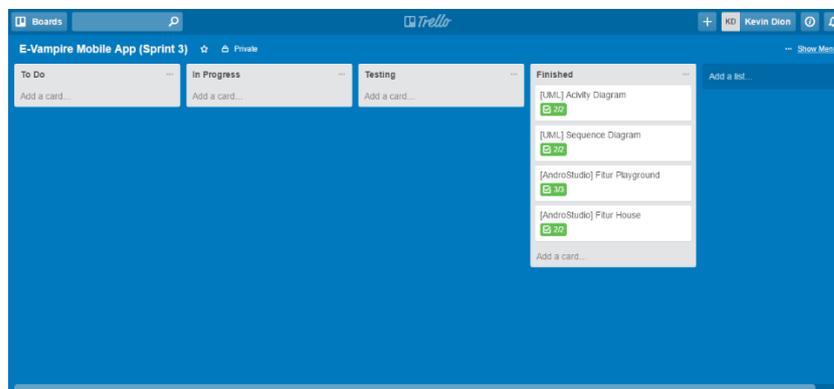
	Saya ingin melihat informasi persediaan darah pada bank darah	Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
		Desain <i>mockup</i>	2
US-08	Saya dapat membaca informasi edukatif mengenai donor darah dan memberikan komentar	Desain <i>mockup</i>	2
		Desain konten	5
		Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
US-11	Saya ingin melakukan permintaan kebutuhan darah darurat	Desain <i>mockup</i>	1
		Implementasi <i>front-end code</i>	2
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	2
Total Jam Pengerjaan			36

### C.3.3. Usecase Diagram



Gambar C.3.3 Usecase Diagram Sprint 3

### C.3.4. Sprint Board



Gambar C.3.4 Trello Sprint 3

#### C.4. Sprint 4

*Sprint 4* merupakan iterasi keempat dalam pengembangan aplikasi. Dalam tahap ini, ada enam fungsional yang akan dikembangkan yaitu Event, Ikut Event, Lihat Profil dan Ubah Profil.

##### C.4.1. Analisis

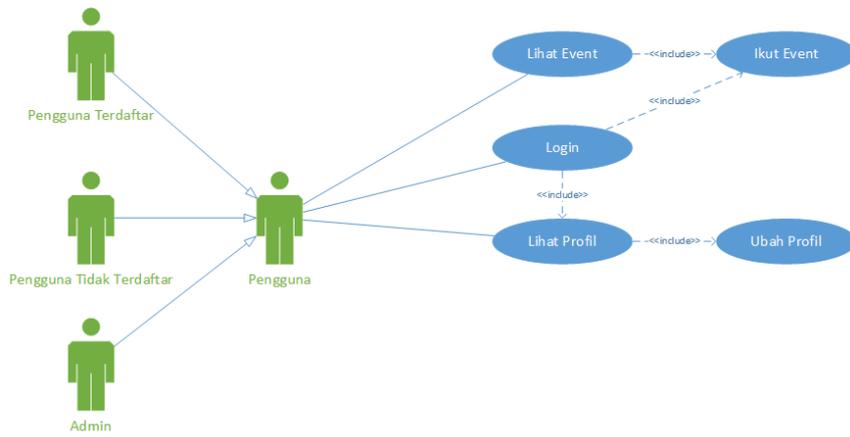
Analisis setiap *sprint* akan disesuaikan dengan *user stories* dan digambarkan menjadi *usecase diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

##### C.4.2. Sprint Backlog

Tabel C.4.2 Sprint Backlog Sprint 4

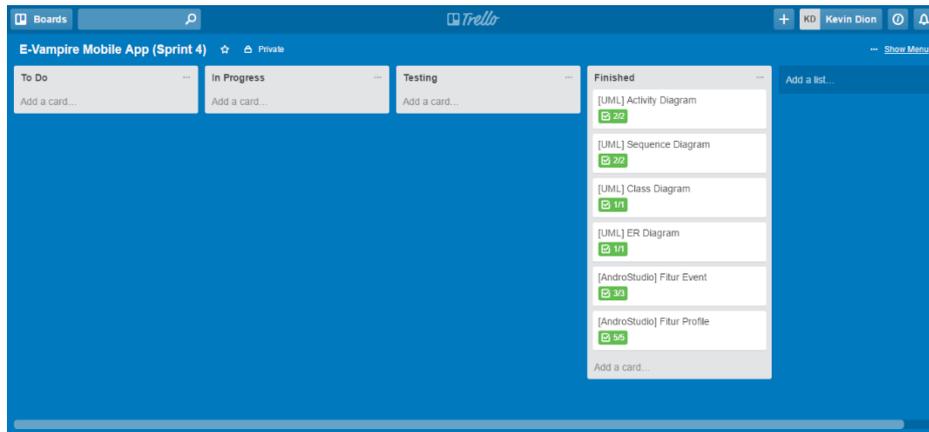
<i>User Story ID</i>	Saya ingin/butuh/dapat untuk	<i>Task</i>	Estimasi (Jam)
US-10	Saya dapat mengubah data akun dan melihat data akun	Desain <i>mockup</i>	1
		Implementasi <i>front-end code</i>	1
		Implementasi <i>back-end code</i>	4
		<i>Testing</i>	3
		Desain <i>mockup</i>	1
US-05	Saya ingin melihat daftar acara donor darah	Desain <i>mockup</i>	2
		Desain konten	3
	Saya ingin batal mengikuti acara donor darah	Implementasi <i>front-end code</i>	3
		Implementasi <i>back-end code</i>	4
US-06	Saya ingin mengikuti acara donor darah dan batal mengikuti acara	<i>Testing</i>	2
		Desain <i>mockup</i>	2
	Saya ingin melihat tanggal rekomendasi melakukan donor darah selanjutnya	Implementasi <i>front-end code</i>	4
		Implementasi <i>back-end code</i>	5
Total Jam Pengerjaan			35

##### C.4.3. Usecase Diagram



Gambar C.4.3 Usecase Diagram Sprint 4

#### C.4.4. Sprint Board



Gambar C.4.4 Trello Sprint 4

### C.1. Tabel Pengerjaan *Sprint*

Tabel C.5.1 Hasil Pengerjaan *Sprint 1*

User Story ID	Saya ingin/butuh/dapat untuk	Task	Estimasi (Jam)	Sprint 1													
				H-1	H-2	H-3	H-4	H-5	H-6	H-7	H-8	H-9	H-10	H-11	H-12	H-13	H-14
US-01	Saya dapat login	Desain <i>mockup</i>	3	1	1	1											
		Implementasi <i>front-end code</i>	4				1	1	1	1							
	Saya ingin login menggunakan akun media sosial	Implementasi <i>back-end code</i>	5							1	1	1	1	1			
		Testing	4											1	1	1	1
US-02	Saya dapat melakukan pendaftaran	Desain <i>mockup</i>	3	1	1	1											
		Implementasi <i>front-end code</i>	4				1	1	1	1							
		Implementasi <i>back-end code</i>	5							1	1	1	1	1			
		Testing	3											1	1	1	
US-03	Saya dapat mengembalikan kata sandi	Desain <i>mockup</i>	3	1	1	1											
		Implementasi <i>front-end code</i>	4				1	1	1	1							
		Implementasi <i>back-end code</i>	5							1	1	1	1	1			
		Testing	3											1	1	1	
U-13	Saya dapat logout	Implementasi <i>back-end code</i>														1	1
		Testing														1	1
<i>Actual Remaining Hours</i>			204	201	198	195	192	189	186	180	177	174	171	165	162	157	154
<i>Estimated Remaining Hours</i>			204.0	201.8	198.8	195.8	192.8	189.8	186.8	181.5	177.8	174.8	171.8	166.5	162.8	158.3	154.8

Tabel C.5.2 Hasil Pengerjaan Sprint 2

User Story ID	Saya ingin/butuh/dapat untuk	Task	Estimasi (Jam)	Sprint 2													
				H-15	H-16	H-17	H-18	H-19	H-20	H-21	H-22	H-23	H-24	H-25	H-26	H-27	H-28
US-04	Saya dapat melihat tampilan Home	Desain <i>mockup</i>	3	1	1	1											
		Desain konten	4			1	1	1	1								
		Implementasi <i>front-end code</i>	4				1	1	1	1							
		Implementasi <i>back-end code</i>	5								1	1	1	1			
		Testing	3										1	1	1	1	1
US-12	Saya ingin menghubungi pihak E-Vampire	Desain <i>mockup</i>	2	1	1										1	1	1
		Implementasi <i>front-end code</i>	3			1	1	1									
		Implementasi <i>back-end code</i>	5						1	1	1	1	1				
		Testing	3										1	1	1		
US-09	Saya ingin melihat lokasi rumah sakit dan PMI	Desain <i>mockup</i>	3	1	1	1											
		Implementasi <i>front-end code</i>	4			1	1	1	1								
		Implementasi <i>back-end code</i>	5							1	1	1	1	1			
		Testing	3											1	1		
US-14	Saya dapat melihat informasi tentang aplikasi E-Vampire	Desain <i>mockup</i>	3	1	1	1											
		Implementasi <i>front-end code</i>	4				1	1	1	1							
		Testing	3								1	1	1	1			
<i>Actual Remaining Hours</i>			154	150	146	140	136	131	126	122	117	113	107	101	97	95	93
<i>Estimated Remaining Hours</i>			154.8	150	146	140	136	131	126	122	117	113	107	101	97	95	93

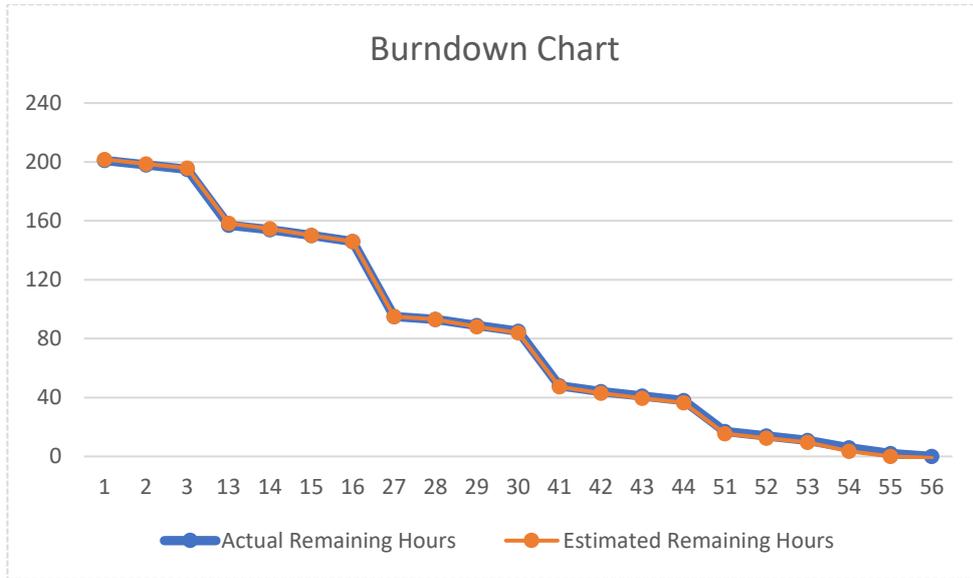
Tabel C.5.3 Hasil Pengerjaan Sprint 3

User Story ID	Saya ingin/butuh/dapat untuk	Task	Estimasi (Jam)	Sprint 3													
				H-28	H-29	H-30	H-31	H-32	H-33	H-34	H-35	H-36	H-37	H-38	H-39	H-40	H-41
US-08	Saya dapat membaca informasi edukatif mengenai donor darah	Desain <i>mockup</i>	2	1	1	1											
		Desain konten	3			1	1	1	1								
		Implementasi <i>front-end code</i>	4							1	1	1	1	1			
		Implementasi <i>back-end code</i>	2											1	1	1	
		<i>Testing</i>	2	1	1	1											
US-11	Saya ingin melakukan permintaan kebutuhan darah darurat	Desain <i>mockup</i>	2	1	1	1											
		Implementasi <i>front-end code</i>	5			1	1	1	1	1							
		Implementasi <i>back-end code</i>	3								1	1	1				
		<i>Testing</i>	4										1	1	1	1	
US-07	Saya ingin melihat informasi persediaan darah	Desain <i>mockup</i>	1	1	1												
	Saya ingin melihat informasi persediaan darah pada bank darah	Implementasi <i>front-end code</i>	2			1	1	1									
		Implementasi <i>back-end code</i>	4						1	1	1	1					
		<i>Testing</i>	2										1	1	1		
<i>Actual Remaining Hours</i>			93	89	85	80	77	74	71	68	65	62	59	55	51	48	44
<i>Estimated Remaining Hours</i>			93	88	84	78.75	76.25	73.25	70.25	67.25	64.25	61.25	58.25	54	50	47.25	43

Tabel C.5.4 Hasil Pengerjaan Sprint 4

User Story ID	Saya ingin/butuh/dapat untuk	Task	Estimasi (Jam)	Sprint 4														
				H- 42	H- 43	H- 44	H- 45	H- 46	H- 47	H- 48	H- 49	H- 50	H- 51	H- 52	H- 53	H- 54	H- 55	
US-10	Saya dapat mengubah data akun dan melihat data akun	Desain <i>mockup</i>	2	1	1													
		Implementasi <i>front-end code</i>	2			1	1											
		Implementasi <i>back-end code</i>	5					1	1	1	1	1						
		<i>Testing</i>	4										1	1	1	1		
US-05	Saya ingin melihat daftar acara donor darah	Desain <i>mockup</i>	3	1	1	1												
		Desain konten	4				1	1	1	1								
	Saya ingin batal mengikuti acara donor darah	Implementasi <i>front-end code</i>	4								1	1	1	1				
		Implementasi <i>back-end code</i>	5										1	1	1	1	1	1
US-06	Saya ingin mengikuti acara donor darah dan batal mengikuti acara	<i>Testing</i>	3												1	1	1	
		Desain <i>mockup</i>	3	1	1	1												
		Implementasi <i>front-end code</i>	5				1	1	1	1	1							
		Implementasi <i>back-end code</i>	6									1	1	1	1	1	1	1
<i>Actual Remaining Hours</i>			44	41	38	35	32	29	26	23	20	17	14	11	6	2	0	
<i>Estimated Remaining Hours</i>			43	39.5	36.5	33.5	30.5	27.5	24.5	21.5	18.5	15.5	12.5	9.5	3.5	0	1	

C.5. Burndown Chart



Gambar C.6 Burndown Chart



## D.2. Hasil Wawancara

Berikut merupakan hasil dari form pertanyaan kuesioner oleh para responden:

Data Responden

Nama : Abadila Herditya Rosadi

Umur : 27

Jenis Kelamin : Laki-laki

Penilaian

**Tabel D.2.1 Hasil Kuesioner Responden 1**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?		✓			
Apakah antarmuka aplikasi mudah dipelajari ?	✓				
Apakah komponen dan warna yang digunakan konsisten ?		✓			
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?		✓			
Apakah aplikasi dapat diakses dengan baik?	✓				
Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?	✓				
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?			✓		
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?			✓		
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?		✓			
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?		✓			

Data Responden

Nama : Ilamana Bintang Lazuardi  
 Umur : 23  
 Jenis Kelamin : Laki-laki

Penilaian

**Tabel D.2.2 Hasil Kuesioner Responden 2**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?	✓				
Apakah antarmuka aplikasi mudah dipelajari ?		✓			
Apakah komponen dan warna yang digunakan konsisten ?		✓			
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?		✓			
Apakah aplikasi dapat diakses dengan baik?			✓		
Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?	✓				
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?	✓				
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?	✓				
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?			✓		
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?	✓				

Data Responden

Nama : Andinia Vigianti  
 Umur : 31  
 Jenis Kelamin : Perempuan

Penilaian

**Tabel D.2.3 Hasil Kuesioner Responden 3**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?		✓			
Apakah antarmuka aplikasi mudah dipelajari ?		✓			
Apakah komponen dan warna yang digunakan konsisten ?		✓			
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?		✓			
Apakah aplikasi dapat diakses dengan baik?			✓		
Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?	✓				
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?	✓				
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?			✓		
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?		✓			
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?	✓				

Data Responden

Nama : Yusuf Rizaldi  
 Umur : 22  
 Jenis Kelamin : Laki – laki

Penilaian

**Tabel D.2.4 Hasil Kuesioner Responden 4**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?	✓				
Apakah antarmuka aplikasi mudah dipelajari ?		✓			
Apakah komponen dan warna yang digunakan konsisten ?	✓				
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?		✓			
Apakah aplikasi dapat diakses dengan baik?		✓			
Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?		✓			
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?		✓			
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?		✓			
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?	✓				
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?	✓				

Data Responden

Nama : Ripaldhi Aldiana  
 Umur : 22  
 Jenis Kelamin : Laki - Laki

Penilaian

**Tabel D.2.5 Hasil Kuesioner Responden 5**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?	✓				
Apakah antarmuka aplikasi mudah dipelajari ?		✓			
Apakah komponen dan warna yang digunakan konsisten ?		✓			
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?	✓				
Apakah aplikasi dapat diakses dengan baik?	✓				
Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?	✓				
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?	✓				
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?	✓				
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?		✓			
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?		✓			

Data Responden

Nama : Satrio Mukti  
 Umur : 40  
 Jenis Kelamin : Laki - Laki

Penilaian

**Tabel D.2.6 Hasil Kuesioner Responden 6**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?			✓		
Apakah antarmuka aplikasi mudah dipelajari ?		✓			
Apakah komponen dan warna yang digunakan konsisten ?	✓				
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?	✓				
Apakah aplikasi dapat diakses dengan baik?		✓			
Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?			✓		
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?	✓				
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?		✓			
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?		✓			
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?	✓				

Data Responden

Nama : Patria Darwis  
 Umur : 30  
 Jenis Kelamin : Laki – laki

Penilaian

**Tabel D.2.7 Hasil Kuesioner Responden 7**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?	✓				
Apakah antarmuka aplikasi mudah dipelajari ?		✓			
Apakah komponen dan warna yang digunakan konsisten ?	✓				
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?	✓				
Apakah aplikasi dapat diakses dengan baik?		✓			
Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?	✓				
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?	✓				
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?	✓				
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?	✓				
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?	✓				

Data Responden

Nama : Ganesha Inugraha  
 Umur : 22  
 Jenis Kelamin : Laki – laki

Penilaian

**Tabel D.2.8 Hasil Kuesioner Responden 8**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?	✓				
Apakah antarmuka aplikasi mudah dipelajari ?	✓				
Apakah komponen dan warna yang digunakan konsisten ?	✓				
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?		✓			
Apakah aplikasi dapat diakses dengan baik?		✓			
Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?	✓				
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?		✓			
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?	✓				
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?			✓		
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?	✓				

Data Responden

Nama : Qori Winata  
 Umur : 23  
 Jenis Kelamin : Perempuan

Penilaian

**Tabel D.2.9 Hasil Kuesioner Responden 9**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?			✓		
Apakah antarmuka aplikasi mudah dipelajari ?	✓				
Apakah komponen dan warna yang digunakan konsisten ?	✓				
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?			✓		
Apakah aplikasi dapat diakses dengan baik?	✓				
Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?	✓				
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?			✓		
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?	✓				
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?			✓		
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?	✓				

Data Responden

Nama : Galang Surya Putra  
Umur : 22  
Jenis Kelamin : Laki-laki

Penilaian

**Tabel D.2.10 Hasil Kuesioner Responden 10**

Pertanyaan	Penilaian				
	SS (5)	S (4)	CS (3)	TS (2)	STS (1)
Apakah aplikasi memiliki tampilan yang menarik secara keseluruhan ?		✓			
Apakah antarmuka aplikasi mudah dipelajari ?	✓				
Apakah komponen dan warna yang digunakan konsisten ?		✓			
Apakah menu-menu yang terdapat di aplikasi dapat dimengerti dengan baik ?	✓				
Apakah aplikasi dapat diakses dengan baik?		✓			

Apakah data yang ditampilkan sesuai dengan kebutuhan pengguna ?	✓				
Apakah aplikasi menyediakan informasi yang jelas mengenai penggunaan aplikasi ?	✓				
Aplikasi memberi umpan balik dari setiap aksi yang diberikan ?	✓				
Apakah aplikasi memberikan nilai tambah kepada penggunanya ?	✓				
Apakah aplikasi menyimpan informasi pengguna serta menyediakan fitur untuk memperbaruinya dikemudian waktu?	✓				