

**PERANCANGAN ARSITEKTUR DAN PENERAPAN SOFTWARE
AS A SERVICE DAN BACKEND AS A SERVICE DENGAN
METODE AGILE FRAMEWORK SCRUM DAN
DEVOPS PADA STARTUP BARU
(Studi Kasus: CV. Monkey Teknologi Indonesia)**

TESIS

Disusun sebagai salah satu syarat untuk
memperoleh gelar Magister Komputer
dari Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI

Oleh:

ALI ABDUL WAHID

NPM: 2020210009



**PROGRAM STUDI PASCASARJANA
MAGISTER SISTEM INFORMASI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER LIKMI BANDUNG
2021**

**PERANCANGAN ARSITEKTUR DAN PENERAPAN *SOFTWARE AS A SERVICE* DAN *BACKEND AS A SERVICE* DENGAN
METODE *AGILE FRAMEWORK SCRUM* DAN
DEVOPS PADA STARTUP BARU
(Studi Kasus: CV. Monkey Teknologi Indonesia)**

Oleh:

ALI ABDUL WAHID

NPM: 2020210009

Bandung, 21 September 2021
Menyetujui,

Prof. Dr. Eng. H. Ana Hadiana.

Pembimbing

**PROGRAM STUDI PASCASARJANA
MAGISTER SISTEM INFORMASI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER LIKMI BANDUNG
2021**

ABSTRAK

PERANCANGAN ARSITEKTUR DAN PENERAPAN SOFTWARE AS A SERVICE DAN BACKEND AS A SERVICE DENGAN METODE AGILE FRAMEWORK SCRUM DAN DEVOPS PADA STARTUP BARU (Studi Kasus: CV. Monkey Teknologi Indonesia)

Oleh:
Ali Abdul Wahid
2020210009

Agar *startup* dapat mencapai valuasi yang tinggi maka diperlukan suatu perencanaan yang sangat matang, analisis perancangan diperlukan saat pengembangan sistem NgUang agar saat *development* bisa lebih cepat, bagus dengan biaya efektif. Saat ini dibutuhkan nya pengembangan *website* dan *mobile apps* pada produk NgUang maka dari itu dibutuhkan nya arsitektur teknologi, server, aplikasi dan juga bisnis, penerapan *cloud computing* juga dapat membantu dalam pembuatan arsitektur bisnis. Dalam penelitian ini, pengembangan sistem dilakukan menggunakan metode *agile* dengan *framework scrum* untuk membantu pengembangan sistem dengan total 5 *sprint* atau 5 minggu yang mencakup aktivitas *backlog*, *sprint*, *meeting*, *retrospective*, dan *review*. Selain penerapan metode *agile* dengan *framework scrum*, penerapan *devops* sangat membantu *scrum* tim karena *agile scrum* sendiri bisa sangat dinamis, dimana proses *deployment* bisa dilakukan per *sprint* ataupun kapan saja. Dengan diberlakukannya penerapan *devops* di awal, *scrum* tim tidak perlu memikirkan *deployment* lagi yang akan sangat memangkas waktu *development* sangat besar sehingga waktu *deployment* akan lebih efektif dan *deployment* dapat dilakukan secara berkala. Metode *devops* juga mampu mempersingkat atau mempercepat waktu antara pengoperasian dan pengembangan *software* tanpa mengurangi kualitas dari *software*, sehingga *deployment* dapat dilakukan secara berkala di setiap *sprint* dan mempersingkat proses *deployment*, lalu dengan menerapkan *cloud computing* model BaaS dapat membantu sisi internal NgUang untuk mempersingkat waktu dan mengurangi biaya saat pengembangan dan model SaaS membantu dalam segi marketing untuk menarik pengunjung jauh lebih banyak, metode *agile* dengan *framework scrum*, metode *devops* dan penerapan *cloud computing* yang diterapkan berhasil diterapkan dan cocok dikombinasikan pada *startup* baru khususnya CV Monkey Teknologi Indonesia dengan produk NgUang nya.

Kata kunci: *Agile*, *Scrum*, *Backend as a Service*, *Software as a Service*, *Startup*

ABSTRACT

PERANCANGAN ARSITEKTUR DAN PENERAPAN SOFTWARE AS A SERVICE DAN BACKEND AS A SERVICE DENGAN METODE AGILE FRAMEWORK SCRUM DAN DEVOPS PADA STARTUP BARU (Studi Kasus: CV. Monkey Teknologi Indonesia)

By:
Ali Abdul Wahid
2020210009

So that startups can achieve high valuations, a very careful planning is needed, design analysis is needed when developing the NgUang system so that development can be faster, better and cost-effective. Currently, there is a need to develop websites and mobile apps on NgUang products, therefore technology, server, application and business architecture is needed, the application of cloud computing can also help in making business architecture. In this study, system development was carried out using agile methods with the Scrum framework to assist system development with a total of 5 sprints or 5 weeks which included backlog, sprint, meeting, retrospective, and review activities. In addition to the application of agile methods with the scrum framework, the application of devops is very helpful for the scrum team because agile scrum itself can be very dynamic, where the deployment process can be carried out per sprint or at any time. With the implementation of devops implementation at the beginning, the scrum team does not need to think about deployment anymore which will greatly reduce development time greatly so that deployment time will be more effective and deployment can be carried out regularly. The DevOps method is also able to shorten or accelerate the time between operation and software development without compromising the quality of the software, so that deployments can be carried out periodically in every sprint and shorten the deployment process, then by implementing the BaaS cloud computing model can help the internal side of NgUang to shorten time and reduce costs. costs during development and the SaaS model helps in terms of marketing to attract more visitors, the agile method with the scrum framework, the Devops method and the application of cloud computing that has been implemented has been successfully implemented and is suitable to be combined with new startups, especially CV Monkey Teknologi Indonesia with its NgUang product.

Keywords: Agile, Scrum, Backend as a Service, Software as a Service, Startup

KATA PENGANTAR

Puji syukur ke hadirat Tuhan YME, atas segala karuniannya sehingga tesis ini dapat terselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada :

1. Bapak Prof. DR. Eng H. Ana Hadiana sebagai pembimbing Tesis yang telah memberikan perhatian, dukungan dan arahan sangat berarti.
2. Seluruh tim CV Monkey Teknologi Indonesia yang telah membantu dalam penyusunan laporan ini.
3. Seluruh tenaga pendidik dan tenaga kependidikan Program Magister Komputer STMIK LIKMI
4. Rekan – rekan mahasiswa STMIK LIKMI Angkatan 2020
5. Serta berbagai pihak yang tidak mungkin dapat penulis sebutkan satu per satu.

Demikian semoga laporan penelitian tesis ini memberikan manfaat positif bagi pribadi, lingkungan, almamater, masyarakat luas, bahkan ilmu pengetahuan. Semoga Allah S.W.T memberikan rahmat, dan keberkahan kepada semua pihak yang telah membantu penyelesaian laporan tesis ini. Amin.

Bandung, 14 Agustus 2021

Penulis

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR DAN ILUSTRASI	vii
DAFTAR TABEL.....	ix
DAFTAR SINGKATAN DAN LAMBANG	x
DAFTAR SIMBOL.....	xi
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Ruang Lingkup Penelitian	4
1.5 Jenis Penelitian & Teknik Pengumpulan Data	4
1.6 Sistematika Penulisan	5
BAB II LANDASAN TEORI.....	7
2.1 <i>Scrum</i>	7
2.2 <i>Devops</i>	10
2.2.1 <i>Gitlab</i>	12
2.2.2 <i>Firebase</i>	12
2.3 <i>Cloud Computing</i>	13
2.4 Arsitektur	14
2.5 <i>Startup</i>	15
2.6 Sistem Informasi.....	16
2.7 <i>Database</i>	17

2.8	<i>Unified Modelling Language (UML)</i>	17
2.9	Penelitian Terdahulu	18
BAB III OBJEK DAN METODELOGI PENELITIAN.....		20
3.1	CV. Monkey Teknologi Indonesia	20
3.1.1	Platform yang sudah berjalan	20
3.1.2	Platform yang akan dikembangkan	23
3.2	Kerangka Penelitian	23
3.3	Tahap Perencanaan.....	25
3.4	Studi Literatur	25
3.5	Tahap Pengumpulan Data	25
3.6	Implementasi <i>Scrum</i>	26
3.7	Perancangan Arsitektur.....	26
BAB IV HASIL DAN PEMBAHASAN.....		28
4.1	Perancangan Aplikasi.....	28
4.2	Implementasi <i>Scrum</i>	28
4.2.1	Scrum Team	28
4.2.2	Story Point	29
4.2.3	Product Backlog.....	29
4.2.4	Sprint Backlog.....	31
4.2.5	Sprint.....	43
4.2.6	Sprint Retrospective & Review	43
4.3	Perancangan Arsitektur Bisnis	55
4.4	Perancangan Arsitektur Teknologi	56
4.5	Perancangan Arsitektur <i>Server</i>	58
4.6	Perancangan Arsitektur Apikasi	58
4.6.1	Use Case Diagram.....	59
4.6.2	Class Diagram	62
4.6.3	Sitemap Aplikasi	69

4.7	Implementasi <i>Devops</i>	72
4.7.1	Continuous Integration.....	73
4.7.2	Continuous Deployment.....	74
BAB V KESIMPULAN DAN SARAN.....		76
5.1	Kesimpulan.....	76
5.2	Saran.....	77
DAFTAR PUSTAKA		xiv
LAMPIRAN		xvi

DAFTAR GAMBAR DAN ILUSTRASI

Gambar 2.1 <i>Sprint Workflow</i>	8
Gambar 2.2 <i>Sprint Retrospective</i>	9
Gambar 2.3 <i>Devops Lyfecycle</i>	11
Gambar 3.2 <i>Sitemap Dashboard Website</i>	22
Gambar 3.3 Kerangka Penelitian	24
Gambar 4.1 <i>Product Backlog</i>	30
Gambar 4.2 Perancangan <i>Sprint 1</i>	32
Gambar 4.3 Perancangan <i>Sprint 2</i>	34
Gambar 4.4 Perancangan <i>Sprint 3</i>	37
Gambar 4.5 Perancangan <i>Sprint 4</i>	39
Gambar 4.6 Perancangan <i>Sprint 5</i>	42
Gambar 4.7 <i>Sprint Retrospective Sprint 1</i>	44
Gambar 4.8 <i>Sprint Retrospective Sprint 2</i>	45
Gambar 4.9 <i>Website Panel</i>	46
Gambar 4.10 <i>Website Panel 2</i>	47
Gambar 4.11 <i>Payment Gateway Online</i>	47
Gambar 4.12 <i>Sprint Retrospective Sprint 3</i>	48
Gambar 4.13 Aplikasi <i>Mobile Nguang</i>	49
Gambar 4.14 <i>Sprint Retrospective Sprint 4</i>	50
Gambar 4.15 <i>Reporting Aplikasi Mobile Nguang</i>	51
Gambar 4.16 <i>Website Landing</i>	52
Gambar 4.17 <i>Website Guide</i>	52
Gambar 4.18 <i>Sprint Retrospective Sprint 5</i>	53
Gambar 4.19 <i>Pricing Nguang</i>	56
Gambar 4.20 <i>Arsitektur Teknologi</i>	57
Gambar 4.21 <i>Arsitektur Server</i>	58

Gambar 4.22 Use Login Dan Register	59
Gambar 4.23 Use Case Diagram Mobile Dashboard Nguang	59
Gambar 4.24 Use Case Diagram Panel Nguang	60
Gambar 4.25 Use Case Diagram Landing Nguang	61
Gambar 4.26 Use Case Diagram Guide Nguang.....	61
Gambar 4.27 Class Diagram Apps	63
Gambar 4.28 Class Diagram Database	65
Gambar 4.29 Class Diagram Globals	67
Gambar 4.30 Sitemap Dashboard Mobile.....	69
Gambar 4.31 Sitemap Website Guide	70
Gambar 4.32 Sitemap Website Panel.....	71
Gambar 4.33 Sitemap Website Landing	72
Gambar 4.34 Stage CI/CD	73
Gambar 4.35 Commit And Push	74
Gambar 4.36 Build Mobile Apps.....	74
Gambar 4.37 Firebase Hosting History Deployment.....	75
Gambar 4.38 Ionic App Flow History Deployment	75

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	18
Tabel 3.1 Fitur <i>Dashboard</i>	21
Tabel 4.1 <i>Scrum</i> Tim	29
Tabel 4.2 <i>Story Point</i>	29
Tabel 4.3 <i>Timeline Sprint</i>	31
Tabel 4.4 <i>User Story Sprint 1</i>	33
Tabel 4.5 <i>User Story Sprint 2</i>	35
Tabel 4.6 <i>User Story Sprint 3</i>	37
Tabel 4.7 <i>User Story Sprint 4</i>	40
Tabel 4.8 <i>User Story Sprint 5</i>	42
Tabel 4.9 Tabel <i>Domain</i>	45
Tabel 4.10 <i>Black Box</i>	54
Tabel 4.11 <i>Class Diagram Apps</i>	64
Tabel 4.12 <i>Class Diagram Database</i>	66
Tabel 4.13 <i>Class Diagram Globals</i>	68

DAFTAR SINGKATAN DAN LAMBANG

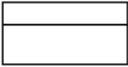
UKM	Usaha Kecil Menengah
SaaS	<i>Software as a Service</i>
BaaS	<i>Backend as a Service</i>
CI	<i>Continuous Integration</i>
CD	<i>Continuous Deployment</i>
PaaS	<i>Platform as a Service</i>
IaaS	<i>Infrastructure as a Service</i>
BRI	Bank Rakyat Indonesia
VA	<i>Virtual Account</i>
TOS	<i>Terms of Services</i>
IT	<i>Information Technology</i>
FAQ	<i>Frequently Asked Question</i>
QA	<i>Quality Assurance</i>
Devops	<i>Development dan Operation</i>
TI	Teknologi Informasi
UML	<i>Unified Modeling Language</i>

DAFTAR SIMBOL

1. Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan sebuah himpunan peran atau objek yang memainkan sebagai pengguna ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana suatu objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Untuk menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	mendeskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).

2. Simbol Class Diagram

No	Simbol	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

DAFTAR LAMPIRAN

Lampiran 1	<i>Devops CI</i>
Lampiran 2	<i>Devops CD</i>
Lampiran 3	<i>Screenshot Website Landing Page</i>
Lampiran 4	<i>Screenshot Website Guide</i>
Lampiran 5	<i>Screenshot Mobile Apps</i>

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di zaman *millenial* ini, peran dari teknologi informasi (TI) telah berubah dari sekedar sebagai sarana untuk meningkatkan operasional *startup* atau perusahaan, hingga menjadi dorongan untuk perubahan mendasar dalam struktur operasional dan manajemen *startup*. Hal ini dikarenakan teknologi informasi memiliki kemampuan untuk mengurangi biaya, meningkatkan produktivitas, meningkatkan hubungan pelanggan, meningkatkan pengambilan keputusan, dan mengembangkan aplikasi strategis yang baru di perusahaan atau *startup* (Irfan. et,al, 2018).

Istilah *startup* selalu diidentikkan dengan *startup* yang menggunakan teknologi informasi dalam produknya, ini biasanya mengacu pada bisnis atau perusahaan baru dan identik dengan aplikasi atau teknologi. Usaha yang tidak menerapkan teknologi biasanya dikatakan sebagai UKM (Usaha Kecil Menengah) (Baskoro, 2013). Saat ini *startup* sudah dikenal secara luas oleh masyarakat terutama masyarakat Indonesia yang memudahkan kegiatan sehari hari, seperti Gojek, Grab, Tokopedia, dll.

Penerapan teknologi informasi atau sistem informasi dalam organisasi atau *startup* memang tidak mudah. Perlu sekali adanya perencanaan yang baik, kolaborasi antar bagian yang kompak, kemauan yang kuat untuk berubah, dan sejalan dengan visi dan misi organisasi untuk mewujudkannya (Aryani dan Assegaff, 2017). Apalagi *startup* sekarang sering dikaitkan dengan teknologi, dengan perencanaan yang baik dapat membuat *startup* tumbuh lebih cepat untuk meningkatkan valuasinya, istilah yang sering di dengar yaitu *startup unicorn* yang sudah mencapai USD\$ 1 Juta atau setara Rp14 triliun ataupun *startup decacorn* yaitu valuasi senilai 10 kali lipat dari *unicorn*.

Agar *startup* dapat mencapai valuasi yang tinggi maka diperlukan suatu perencanaan yang sangat matang. Salah satu cara perencanaan adalah dengan merancang sebuah arsitektur *server*, teknologi, bisnis, dan aplikasi. Arsitektur melibatkan perencanaan di mana bagian atau

komponen sesuatu ditarik dari sudut yang berbeda. Ide arsitektur digunakan dalam berbagai bidang, diibaratkan mulai dari perencanaan kota hingga konstruksi (Aryani dan Assegaff, 2017).

Selain membuat perancangan arsitektur, penerapan *cloud computing* dapat diterapkan juga pada *startup*, sehingga sumber daya komputasi mudah diperoleh dan diakses, mudah digunakan, murah, dan segera dijalankan (Marks dan Lozano, 2010). Salah satu contoh model *cloud computing* yaitu SaaS. SaaS menyediakan cara yang mudah, cepat, dan hemat biaya untuk mengembangkan dan menerapkan perangkat lunak yang memenuhi kebutuhan bisnis (Utomo, 2015). Salah satu *startup* yang sudah menerapkan model SaaS ini antara lain Gadjian, Akuntansi Online, Jurnal, Jojonomic, Kata.ai, MokaPos, Pawoon, dll. Lalu model BaaS juga sebagai layanan *cloud* yang memudahkan pengembang untuk mempercepat pembuatan aplikasi, salah satu layanan yang menerapkan model BaaS yaitu *Firebase*. Dengan menggunakan *Firebase*, *developer* bisa fokus dalam mengembangkan aplikasi tanpa memberikan *effort* yang besar untuk urusan *backend* dengan harga yang murah bahkan disediakan layanan gratis dengan ketentuan yang berlaku hal ini sangat cocok diaplikasi disebuah perusahaan baru / *startup*.

CV. Monkey Teknologi Indonesia merupakan *startup* yang berdiri pada tahun 2020 dan bergerak dibidang teknologi yang tergolong baru dan memiliki beberapa produk yang baru. Produk-produk yang dimiliki CV. Monkey Teknologi Indonesia ialah sebuah *platform* aplikasi yang dapat digunakan oleh perusahaan ataupun perorangan untuk membantu kegiatan sehari hari. Produk tersebut ialah NgUang, Ngood, Ngiklan, Ngantre, dan Ngodings (Monkey Admin, 2020). Saat ini untuk produk NgUang belum ada *website landing*, *panel*, *guide*, dan juga *mobile apps*. Maka dari itu, perlu dilakukan pengembangan serta membuat perancangan arsitektur, *server*, teknologi, bisnis, dan aplikasi, yang nantinya arsitektur tersebut dapat digunakan untuk memudahkan saat pengembangan aplikasi ataupun menjadi acuan dokumentasi. CV. Monkey Teknologi Indonesia sendiri menerapkan konsep model *cloud computing* SaaS pada sistem aplikasinya. Agar penerapan model *cloud computing* tersebut dapat diterapkan, dibutuhkan perancangan arsitektur yang bisa memenuhi penerapan modelnya.

Agar pengembangan saat *development* bisa menjadi lebih cepat, bagus, dan menggunakan biaya yang efektif, pendekatan inovasi yang berbeda telah dikembangkan untuk

ini. Dalam hal ini, metode yang digunakan yaitu metode *agile* dengan *framework scrum* (Dobrigkeit, Paul, dan Carroll, 2020). Selain penerapan metode *agile* dengan *framework scrum*, metode *DevOps* juga mampu mempersingkat atau mempercepat waktu antara pengoperasian dan pengembangan *software* tanpa mengurangi kualitas dari *software* (Erich, Amrit dan Daneva, 2017). Dalam hal ini, *DevOps* dapat memperkecil kesalahpahaman antara tim operasi, pengembang, dan pengguna aplikasi sehingga dapat mengidentifikasi masalah dari awal (Erich, Amrit, dan Daneva, 2014). Dengan penerapan metode *agile* dengan *framework scrum* ini dapat mengurangi waktu pengembangan juga meningkatkan fleksibilitas proses dan kualitas pada produk (Dobrigkei, Paula, & Carroll 2020). Maka dari itu, dalam laporan ini penulis mengambil judul “Perancangan arsitektur dan penerapan *software as a service* dan *backend as a service* dengan metode *agile framework scrum* dan *devops* pada *startup* baru (Studi kasus CV. Monkey Teknologi Indonesia)”.

1.2 Rumusan Masalah

Rumusan masalah dari permasalahan yang peneliti uraikan diatas, sebagai berikut:

1. Bagaimana perancangan arsitektur meliputi *server*, teknologi, bisnis, dan aplikasi pada pengembangan produk NgUang di CV. Monkey Teknologi Indonesia?
2. Bagaimana perancangan *cloud computing* pada pengembangan produk NgUang di CV. Monkey Teknologi Indonesia sehingga dapat meningkatkan dari segi bisnis?
3. Bagaimana mengimplementasikan metode *agile* dengan *framework scrum* ke dalam pengembangan produk NgUang di CV. Monkey Teknologi Indonesia?
4. Bagaimana mengimplementasikan metode *devops* ke dalam pengembangan produk NgUang di CV. Monkey Teknologi Indonesia?

1.3 Tujuan Penelitian

Tujuan dari penelitian dari rumusan masalah yang peneliti uraikan, sebagai berikut:

1. Merancang arsitektur *server*, teknologi, bisnis, dan aplikasi untuk memudahkan *developer* atau semua yang terlibat saat pengembangan untuk dokumentasi kedepannya.

2. Perancangan *cloud computing* pada Produk NgUang CV. Monkey Teknologi Indonesia sehingga dapat meningkatkan dari segi bisnis NgUang.
3. Mengimplementasikan metode *agile dengan framework Scrum* untuk meningkatkan tingkat akurasi dan kecepatan dalam tahap *development*, hingga mengatasi masalah di masa mendatang.
4. Mengimplementasikan metode *devops* untuk memudahkan *developer* dengan *automation build*, sehingga pengembangan sistem jauh lebih mudah dan cepat, *deployment* aplikasi dapat dilakukan secara berkala dan sejalan dengan metode *agile* yang diterapkan.

1.4 Ruang Lingkup Penelitian

Ruang lingkup dari penelitian ini adalah:

1. Perancangan arsitektur *server* di CV. Monkey Teknologi Indonesia.
2. Perancangan arsitektur bisnis di CV. Monkey Teknologi Indonesia.
3. Perancangan arsitektur teknologi di CV. Monkey Teknologi Indonesia.
4. Perancangan arsitektur aplikasi di CV. Monkey Teknologi Indonesia.
5. *Cloud computing* di CV. Monkey Teknologi Indonesia.
6. Penerapan *devops* di CV. Monkey Teknologi Indonesia.
7. Penerapan *agile dengan framework scrum* di CV. Monkey Teknologi Indonesia.
8. Perancangan dan penerapan hanya di salah satu produk dari CV. Monkey Teknologi Indonesia yaitu NgUang.

1.5 Jenis Penelitian & Teknik Pengumpulan Data

Jenis penelitian yang akan dilakukan adalah kuantitatif, yaitu metode penelitian ilmiah yang meliputi penggunaan prosedur statistik eksperimental untuk pengumpulan informasi dan analisis data. (Jannah dan Prasetyo, 2011). Berikut adalah teknik pengumpulan data yang dilakukan pada penelitian ini:

1. Pengamatan atau Observasi

Tersiana (2018:12) mendefinisikan observasi yaitu mengamati perilaku dalam kondisi tertentu di seluruh dunia. Pada hakikatnya, observasi adalah kegiatan individu atau makna kasus berdasarkan pendapat pribadi. Teknik observasi ini menggunakan alat bantu untuk membuat draf atau membuat daftar hasil pencarian.

2. Riset Kepustakaan

Merupakan data yang penulis cari di buku atau informasi yang didapat dengan membaca dan meneliti buku-buku yang berkaitan dengan judul penelitian. Selain itu, penulis juga membaca jurnal ilmiah yang berkaitan dengan masalah dan publikasi yang ada di perpustakaan, sebagai bahan pelengkap penelitian yang bermanfaat untuk melengkapi penelitian.

1.6 Sistematika Penulisan

Dalam sistematika penulisan ini agar memudahkan penyusunan pada laporan, maka dibagi menjadi lima bagian dalam sistematika penulisan, sebagai berikut:

BAB I PENDAHULUAN.

Menguraikan tentang latar belakang penelitian, perumusan masalah, tujuan dari penelitian, ruang lingkup dari penelitian, jenis penelitian, teknik pengumpulan data, dan sistematika penulisan.

BAB II LANDASAN TEORI.

Menjelaskan tentang landasan teori pendukung dari konsep perancangan arsitektur, penerapan *cloud computing*, metode *agile dengan framework scrum*, metode *devops*, *gitlab*, dan *firebase*.

BAB III OBJEK DAN METODOLOGI PENELITIAN.

Menjabarkan pilihan metode yang dipilih dalam kerangka berpikir seperti pengumpulan data dan pengembangan sistem untuk merancang sistem yang diteliti.

BAB IV HASIL DAN PEMBAHASAN.

Merancang arsitektur *database*, teknologi, *server*, aplikasi, dan juga implementasi *cloud computing* SaaS dan BaaS untuk CV. Monkey Teknologi Indonesia pada produk NgUang dengan metode *Agile* dengan *framework Scrum* dan *Devops*.

BAB V KESIMPULAN DAN SARAN.

Penulis membuat penutup berupa kesimpulan dari hasil penelitian dan pembahasan mengenai perancangan arsitektur *database*, *server*, teknologi, aplikasi, dan juga implementasi *cloud computing* SaaS dan BaaS untuk CV. Monkey Teknologi Indonesia pada produk NgUang dengan metode *agile* dengan *framework scrum* dan *devops*.

BAB II

LANDASAN TEORI

2.1 *Scrum*

Dalam penelitian ini, penulis menggunakan model *scrum*. Model *scrum* ialah metode pengembangan perangkat lunak yang fleksibel dan juga cepat (Pressman, 2010). Prinsip *scrum* konsisten dengan prinsip *agile* yaitu cepat yang digunakan untuk mengelola pengembangan *software*, seperti kepuasan permintaan, analisis, desain, dan pengiriman (Azdy dan Azhary, 2012).

Scrum bukan sebuah cara untuk menciptakan produk, tetapi untuk membuat kerangka kerja dimana dapat mengintegrasikan metode dan proses. *Scrum* mendemonstrasikan hasil praktik manajemen dan pengembangan produk untuk menciptakan produk yang lebih baik (Jha dan Khan, 2015).

Scrum sendiri merupakan salah satu model dari *agile*. *Agile* adalah metode pengembangan tambahan yang berfokus pada pengembangan cepat, merilis perangkat lunak secara bertahap, menghasilkan kode berkualitas tinggi, dan mengurangi *overhead* proses. Pelanggan mengambil bagian langsung dalam proses pengembangan (Sommerville, 2011).

Scrum memiliki kerangka kerja yang mencakup tim *scrum* dan peran, artefak, kegiatan, dan aturan main. Aturan main *scrum* ini menghubungkan peran, kegiatan, dan artefak serta menjelaskan interaksi dan hubungan antara peserta dan orang lain (Spchwaber, et al, 2011). *Scrum* memiliki tiga komponen sebagai berikut (Azdy dan Azhary, 2012):

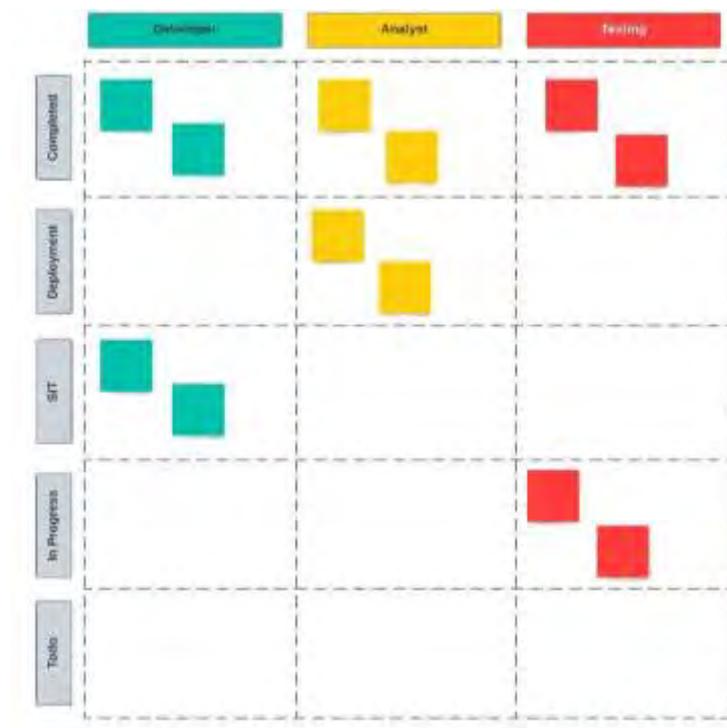
1. Peranan
 - a. *Product owner* memiliki bertanggung jawab atas *value* bisnis dari sebuah produk.
 - b. *Scrum master* memastikan atau menyediakan tim agar dapat bekerja secara fungsional dan produktif selama kegiatan *sprint*.
 - c. *Scrum team member* ialah organisasi individu / *self-organize* untuk menyelesaikan pekerjaan.

2. Artefak

- a. *Product backlog* ialah daftar pekerjaan yang harus diselesaikan dari proyek yang akan dicapai.
- b. *Sprint backlog* ialah kumpulan kegiatan yang sudah disetujui oleh tim dan juga atasan yang diselesaikan dalam satu *sprint*. Satu *sprint* sendiri dapat berlangsung selama 1 – 2 minggu atau pun sesuai kesepakatan tim dan dapat dibagi menjadi beberapa *subtask* dan *task*.
- c. *Burndown chart* ialah ringkasan tampilan terhadap kegiatan yang belum dikerjakan yang disajikan dengan *chart*.

3. Kegiatan

- a. *Sprint planning* ialah kegiatan tim dimana melakukan pertemuan dengan *product owner* untuk menentukan kegiatan yang dilakukan dan diselesaikan selama *sprint*. Saat *sprint* berjalan, peneliti menggunakan *sprint workflow* seperti pada gambar 2.1.

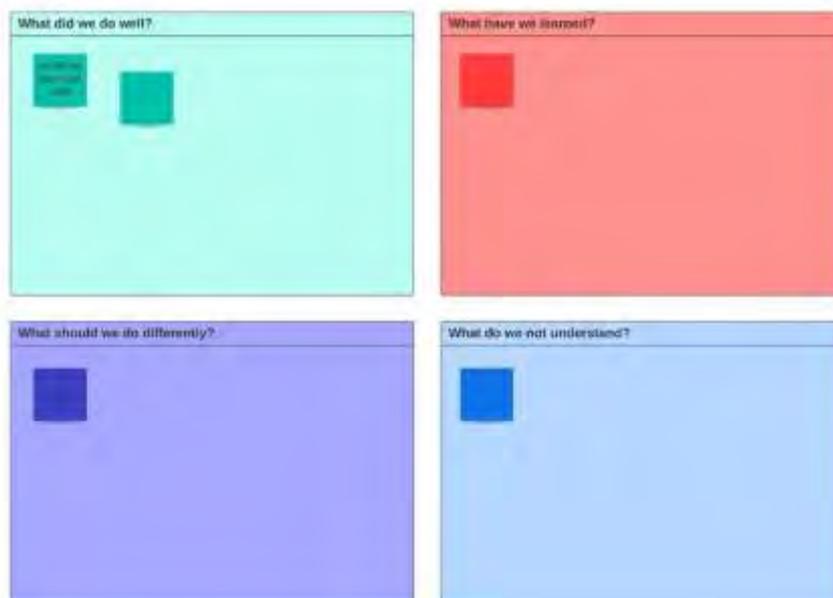


Gambar 2.1 *Sprint Workflow*

Sumber: <https://8allocate.com/article/agile-sprint-workflow-outputs-and-definitions/>

Pada *sprint workflow* ini menjelaskan lima kategori pengerjaan, yaitu:

1. *To do* dimana *tasks* yang ada dalam *sprint* berjalan dengan status belum dikerjakan.
 2. *In progress* dimana *task* yang sedang tersebut sedang dikerjakan.
 3. *SIT* dimana *task* sudah selesai dikerjakan dan belum ada masuk ke *production*.
 4. *Deployment* dimana *task* sudah selesai dikerjakan dan sudah ada pada *production*.
 5. *Completed* dimana *task* tersebut sudah selesai.
- b. *Daily scrum* ialah kegiatan berkumpul setiap harinya untuk menyampaikan *update* atau *progress* dari pekerjaan yang sudah dikerjakan atau mengemukakan permasalahan untuk sebuah tim.
- c. *Sprint review* ialah kegiatan tim untuk *update* apa yang telah diselesaikan atau *progress* selama *sprint* kepada *product owner* dan juga kepada seluruh tim yang terlibat.
- d. *Sprint retrospective* ialah kegiatan mencari bagaimana cara untuk meningkatkan produk yang dilakukan oleh tim dan proses dari *sprint* sebelumnya, agar *sprint* selanjutnya dapat lebih baik dari *sprint* sebelumnya. setelah selesai melakukan *sprint review* dilanjutkan dengan *sprint retrospective* yang digambarkan pada gambar 2.2.



Gambar 2.2 *Sprint Retrospective*

Sumber: <https://www.scrum.org/resources/what-is-a-sprint-retrospective>

Pada *sprint retrospective* ini semua *team member* menuliskan apa saja yang sudah dilakukan dengan baik, sudah belajar hal baru apa saja, apa yang bisa dikembangkan pada sprint yang berjalan, dan apa saja kendala-kendala yang tidak bisa diatasi untuk diperbaiki.

2.2 Devops

DevOps (pengembangan dan operasi) adalah studi konseptual pengembangan dan pengiriman *software* untuk infrastruktur menggunakan pendekatan kolaborasi dan integrasi antara pengembang (*Dev*) dan operator *software* (*Ops*) (Jha dan Khan, 2015). *DevOps* juga merupakan pendekatan organisasi yang berupaya menciptakan komunikasi, kolaborasi, dan empati lintas fungsi dan department (Dyck, Penners dan Lichter 2015).

Menerapkan *devops* adalah langkah untuk mendorong kolaborasi antar tim IT di mana pemeliharaan dan pengembangan *software* sangatlah penting (Dobrigkeit, Paula, dan Carroll, 2020). *DevOps* dapat mengurangi beberapa kegiatan pengembangan yang dilakukan dengan cara lama. (Erich, Amrit dan Daneva, 2017). *DevOps* dapat mengurangi waktu antara rilis *software* dan pengembangan tanpa mengorbankan kualitas *software* itu sendiri. *DevOps* menjembatani kesalahpahaman antara tim pengembangan, tim operasi, dan pengguna aplikasi dengan mengaktifkan deteksi masalah dini. Diperkenalkannya *devops*, membuat pengguna dapat terus berkembang dan versi aplikasi yang lebih baik sering dapat dirilis (Erich, Amrit dan Daneva, 2014). *DevOps* tidak hanya memodifikasi proses dan praktik pengembangan perangkat lunak untuk dipercepat, tetapi juga meningkatkan keandalan, stabilitas, ketahanan, dan keamanan lingkungan *production*.

Sebagai bonus tambahan, menerapkan *devops* membantu mengurangi stres atau beban kerja pada karyawan karena *agile* memerlukan fase *production* yang sering. Perangkat lunak yang rilis lebih awal dan frekuensi tinggi menjadikan *startup* memiliki peluang yang bagus untuk memenangkan persaingan di Industri.

Metode *devops* diharapkan dapat menjadi solusi mempercepat rilis pengembangan produk di CV. Monkey Teknologi Indonesia seperti dapat dilihat pada gambar 2.3 mengenai *devops lifecycle* untuk penjelasan bagaimana *devops* bekerja.

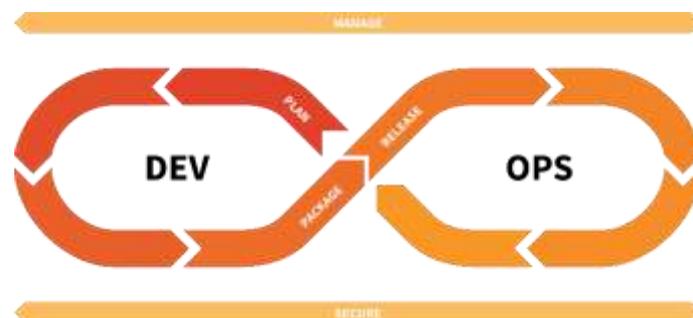
DevOps sendiri dibagi menjadi dua bagian yaitu (Dicoding, 2019):

1. *Continuous Integration* (CI)

Continuous integration adalah salah satu teknik *devops* yang dapat digunakan oleh pengembang untuk mengintegrasikan *coding* ke dalam repositori kode seperti GitHub atau Gitlab yang saat ini menjadi *remote* yang sering digunakan dan menjalankan pengujian dengan cepat dan otomatis.

2. *Continuous Deployment* (CD)

Continuous deployment atau penyebaran berkelanjutan adalah praktik di mana setelah kode berhasil diintegrasikan, aplikasi dapat dibuat dan kemudian diterbitkan secara otomatis.



Gambar 2.3 *Devops Lyfecycle*

Sumber: <https://about.gitlab.com/images/press/devops-lifecycle.svg>

Dev dan *Ops* ini saling berhubungan dimulai dari *plan*, *package* hingga *release*, dengan bantuan *platform gitlab*, *gitlab* membantu yang mengubah cara tim untuk pengembangan, keamanan, dan kolaborasi antar tim dan membangun perangkat lunak.

2.2.1 Gitlab

Istilah *devops* sendiri sering dikaitkan dengan perangkat *git*, seperti *gitlab*. *gitlab* adalah platform *devops* lengkap, dikirimkan sebagai aplikasi tunggal yang secara fundamental mengubah cara tim pengembangan, keamanan, dan *ops* berkolaborasi untuk membangun perangkat lunak dan sudah banyak digunakan oleh para *developer*. Dari ide hingga produksi, *gitlab* membantu tim meningkatkan waktu siklus pengembangan lebih cepat dari minggu ke menit, mengurangi biaya pengembangan, dan waktu ke pasar sekaligus meningkatkan produktivitas pengembang. *Gitlab* sendiri sangat memudahkan *developer* berkolaborasi dengan tim dengan perintah *git* (*Gitlab*, 2021). Dengan penggunaan *gitlab* sendiri maka *developer* dapat melakukan *deployment* ke *production* ataupun *UAT* jauh lebih cepat. *Developer* hanya perlu integrasi *code* dengan *gitlab* dan *gitlab* di konfigurasi dengan *server* contohnya *firebase*. Sebagai contoh jika *developer* melakukan perintah *git push origin sit* maka akan secara otomatis *deployment* ke *environment UAT* atau melakukan perintah *git push origin master* maka akan otomatis *deployment* ke *environment production*.

2.2.2 Firebase

Firebase merupakan buatan dari google dengan model *backend as a service* (BaaS). *Firebase* menjadi solusi yang diberikan google untuk memfasilitasi pengembangan aplikasi seluler. *Remote config* dan *real time database* adalah dua fitur menarik. Selain itu, fitur yang baru *firebase release* yaitu *firestore* yang merupakan pengembangan dari *firebase realtime database*. *Firebase* juga mendukung fitur untuk sistem yang membutuhkan *push* notifikasi, dimana dapat menggunakan fitur *firebase notification*. (Sandy, et.al, 2017).

Selain fitur yang disebutkan sebelumnya, *Firebase* sendiri memiliki fitur *hosting* yang sudah *support* untuk mendukung *devops*. Menggunakan CI/CD dari *gitlab* dan konfigurasi dari *code* maka secara otomatis dapat langsung *deployment* menuju *hosting* yang sudah ditentukan hingga *support* untuk *custom domain* dengan konfigurasi menggunakan DNS.

2.3 Cloud Computing

Model komputasi di mana sumber daya seperti prosesor atau daya komputasi, jaringan, penyimpanan, dan *software* diabstraksi dan disajikan sebagai layanan di jaringan atau internet menggunakan mode akses jarak jauh (Alfatih dan Marco, 2015).

Beberapa atribut penting dari *cloud computing* antara lain adalah ketersediaan sesuai permintaan, manajemen yang mudah, praktis tidak terbatas, dan skalabilitas dinamis. *Cloud computing* menawarkan layanan tak terbatas kepada penggunanya untuk mengakses aplikasi atau sistem tanpa batasan ruang dan waktu.

Cloud computing dapat dikategorikan menjadi 3 bagian yaitu, *infrastructure as a service* (IaaS), *platform as a service* (PaaS), dan *software as a service* (SaaS) (Sulistyo dan Agustina, 2013). Seiring berjalannya waktu, mulai bermunculan jenis-jenis baru dari *cloud computing*, seperti *backend as a service* (BaaS) yang dibuat dan dikembangkan oleh google dengan produknya yang bernama *firebase* (Alfatih dan Marco, R. 2015).

1. Platform as a service (Paas)

PaaS ialah layanan atau jasa yang menyediakan perangkat keras, sehingga *developer* aplikasi tidak perlu mempertimbangkan sistem operasi, skalabilitas infrastruktur, penyeimbangan beban, dan masalah lainnya. Oleh karena itu, *developer* dapat fokus pada *website* atau aplikasi yang dikembangkan, misalnya aplikasi tersebut menjadi tanggung jawab pemasok, seperti microsoft azure.

2. Infrastructure as a service (IaaS)

IaaS ialah layanan atau jasa yang menyediakan sumber daya komputasi dasar yang dapat digunakan pelanggan untuk menjalankan aplikasi. Model ini tampak seperti ruang yang disewakan dari *vendor* pusat data, tetapi lebih berskala mikro. Keuntungan dari layanan atau jasa jenis ini adalah tidak perlu membeli fisik komputer, tetapi dapat membuat mesin *virtual* yang dapat dimodifikasi dengan mudah. Misalnya: amazon EC2, windows azure, dll.

3. Software as a service (SaaS)

SaaS ialah layanan atau jasa komputasi awan yang memungkinkan pelanggan menggunakan perangkat lunak yang disediakan oleh *vendor*. Pengguna tidak perlu lagi membeli

lisensi *software*, hanya perlu berlangganan dan membayar sesuai kebutuhan. Contohnya termasuk *adobe creative cloud*, *yahoo messenger*, *twitter*, *facebook*, *skype*, *office365*, dll.

4. *Backend as a service* (Baas)

BaaS adalah model layanan cloud di mana pengembang melakukan *outsourcing* semua aspek tersembunyi dari *web* atau aplikasi seluler. Jadi, yang harus dilakukan adalah menulis dan *maintenance frontend*. Penyedia BaaS menawarkan perangkat lunak yang telah ditulis sebelumnya untuk tindakan yang dilakukan di *server*, misalnya otentikasi pengguna, manajemen basis data, pembaruan jarak jauh, dan pemberitahuan *push* (untuk aplikasi seluler), penyimpanan, dan *hosting* di *cloud* (Cloudflare, 2021).

2.4 Arsitektur

Istilah "arsitektur" umumnya mengacu pada desain fisik, tapi juga digunakan dalam konteks arsitektur bisnis dan arsitektur pengembangan *software*.

Beberapa definisi yang berkaitan dengan arsitektur, yaitu (Subaeki, 2014):

1. Arsitektur ialah komponen sistem yang tersusun dari jaringan terstruktur yang meliputi *software* dan *hardware*.
2. Desain atau rancangan semua jenis struktur, baik fisik maupun kontekstual, nyata maupun *virtual*.

Kesimpulan yang dijelaskan dapat disimpulkan bahwa arsitektur pada dasarnya digunakan untuk merancang bentuk konstruksi sistem yang termasuk dalam model (*blueprint*) yang dapat dilihat dari berbagai sisi. Arsitektur juga dibagi menjadi beberapa bagian antara lain (*The Open Group*, 2011):

1. Arsitektur teknologi

Arsitektur teknologi ini menjelaskan bagaimana pengembangan arsitektur teknologi untuk mendukung visi arsitektur yang telah disepakati, maksudnya bagaimana untuk mendeskripsikan struktur dan interaksi layanan antar *platform* atau komponen teknologi *login* dan fisik.

2. Arsitektur bisnis

Arsitektur bisnis ini menjelaskan bagaimana pengembangan arsitektur bisnis untuk mendukung visi arsitektur yang telah disepati, maksudnya untuk mendeskripsikan interaksi antara strategi bisnis, organisasi, proses bisnis, dan kebutuhan Informasi yang berkaitan dengan visi bisnis yang ada.

3. Arsitektur aplikasi

Arsitektur aplikasi ini untuk menjelaskan mengenai jenis-jenis utama dari sistem aplikasi atau fitur-fitur utama untuk mendukung bisnis dan memproses data.

4. Arsitektur *database*

Arsitektur *database* ini untuk menjelaskan atau menguraikan kebutuhan data yang dibuat dengan berbagai tabel yang saling terintegrasi yang berkaitan untuk mendukung bisnis.

2.5 **Startup**

Istilah "*startup*" selalu identik dengan *startup* yang menggunakan teknologi informasi dalam produknya, umumnya merujuk pada perusahaan yang masih baru beroperasi dan identik dengan perusahaan yang berkaitan dengan teknologi. Perusahaan yang tidak menerapkan teknologi biasanya disebut sebagai UKM (Usaha Kecil Menengah) (Baskoro, 2013).

Sejarah *startup* sendiri berawal ketika *startup* menjadi populer di seluruh dunia sekitar tahun 1998 sampai tahun 2000. Perusahaan-perusahaan membuat situs *website* pada waktu bersamaan, dikarenakan saat itu sedang maraknya membuka *website* untuk perusahaan baru demi memulai bisnisnya (Liputan6.com, 2019).

Dari kejadian ini, banyak orang semakin mengenal *internet* sebagai tempat baru untuk berbisnis. Dalam kurun waktu tersebut, perusahaan *startup* juga menjadi perusahaan yang terkait dengan dunia *digital* atau *internet*, teknologi informasi dan aplikasi (Liputan6.com, 2019).

Startup memiliki beberapa karakteristik sebagai berikut (Liputan6.com, 2019):

1. Usia dari perusahaan biasanya kurang dari 3 tahun, diibaratkan seperti usia bayi, dimana saat berusia 3 tahun sedang berkembang untuk menjadi anak-anak dan menjadi dewasa.
2. Jumlah karyawan kurang dari 30 orang pada umumnya untuk sebuah *startup*.

3. Sumber daya manusia *multitasking* atau seseorang karyawan yang sebisa mungkin bisa mengerjakan banyak hal. Meskipun seseorang memulai dengan jumlah karyawan yang relatif kecil, biasanya memiliki bakat dan ahli terbaik di bidangnya.
4. Selain itu, *startup* biasanya mempekerjakan milenial, kaum muda, dan profesional teknologi. Pada umumnya, karyawan kebanyakan anak muda atau berusia antara 20 dan 35 tahun. Karena pada kelompok usia ini masih memiliki semangat yang menggebu, inovatif, dan juga kreatif.
5. Biasanya aktif dalam teknologi meskipun arti sebenarnya dari sebuah *startup* tidak selalu dalam teknologi. Tetapi saat ini *startup* yang sering dikaitkan dengan teknologi saat ini.
6. *Website* merupakan sarana utama bagi perusahaan atau *startup* yang harus memiliki sistem aplikasi atau *website* untuk menjalankan segala aktivitas yang bertujuan menyediakan dan mempromosikan produk yang perusahaan jual. Kegiatan yang dilakukan antara lain yakni layanan *online* berbasis aplikasi yang selalu terkoneksi dengan *database* dengan *website* perusahaan.
7. Prioritas *startup* yaitu untuk menjangkau konsumen sebanyak-banyaknya. Para *startup* selalu menawarkan inovasi untuk memanjakan setiap konsumennya agar dapat menarik konsumen sebanyak mungkin.

2.6 Sistem Informasi

Sistem informasi terdiri dari sekumpulan komponen yang memproses, mengumpulkan, menyimpan, dan menghasilkan semua informasi yang diperlukan untuk semua proses bisnis dan *software* terkait, *database*, dan bahkan aplikasi yang digunakan oleh proses manual. (Satzinger dan Burd, 2012:4).

Menurut Irwan Purwanto dan Dini Destiani (2012:02) mendefinisikan: "Sistem informasi adalah sistem buatan dari sekumpulan komponen komputer pasif yang dirancang untuk mengumpulkan, menyimpan, dan mengelola data serta memberikan informasi kepada pengguna."

Berdasarkan beberapa konsep yang disajikan diatas, sistem informasi dapat diartikan sebagai sekumpulan sumber daya untuk pengumpulan, pemrosesan, penyimpanan, analisis, dan distribusi informasi dalam suatu organisasi untuk tujuan tertentu.

2.7 Database

Menurut Bambang Hariyanto dalam (Minarni dan Susanti, 2014), *database* ialah sekumpulan data atau *element* yang secara *logic* saling berkaitan dalam mempresentasikan fenomena atau peristiwa secara terstruktur dalam domain tertentu untuk mendukung aplikasi dalam sistem tertentu.

Berdasarkan pengertian di atas, dapat ditarik kesimpulan bahwa *database* adalah kumpulan data-data yang saling berkaitan yang diurutkan dalam suatu struktur atau skema dan dapat digunakan kembali dengan mudah dan cepat. Bahkan saat ini database sudah ke *level* yang lebih tinggi yaitu *big data*.

Jadi, beberapa alasan mengapa perlu menggunakan *database* adalah sebagai berikut (Minarni dan Susanti, 2014):

1. Menentukan kualitas informasi. Jika manfaatnya melebihi biaya perolehan informasi, maka informasi yang akurat, tepat waktu dan relevan akan sangat berharga.
2. Salah satu elemen penting dari sistem informasi, ialah sebagai dasar penyediaan informasi.
3. Mengurangi persamaan data atau duplikasi data (redudansi data).
4. Berbagai data dapat dihubungkan dengan mudah.
5. Menghemat tempat dan mengurangi kehilangan data.

2.8 Unified Modelling Language (UML)

Menurut Rosa dan Salahuddin (2015: 133), "*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang paling banyak digunakan di dunia industri untuk mendefinisikan persyaratan, melakukan analisis dan desain, dan menggambarkan arsitektur dalam pemrograman berorientasi objek.

Jadi, *UML (Unified Modeling Language)* dapat diartikan sebagai bahasa visual untuk menggambarkan definisi persyaratan, melakukan analisis, dan merancang dan menggambar arsitektur dalam pemrograman berorientasi objek dengan bantuan teks yang menyertainya. Menurut Rosa dan Salahuddin (2015: 140), UML ini terdiri dari 13 jenis diagram, namun hanya beberapa diagram yang digunakan, antara lain: *class diagram* dan *use case diagram*.

2.9 Penelitian Terdahulu

Hasil penelitian terdahulu dari berbagai literatur yang peneliti dapatkan terdapat pada tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti & Tahun Publikasi	Publikasi	Judul dari Penelitian	Hasil dari Penelitian
1	Andy Prasetyo Utomo, 2015	Jurnal SIMETRIS, Vol 6 No 1	Penerapan Konsep SaaS (<i>Software as A Service</i>) pada Aplikasi Penggajian	Konsep SaaS dapat di terapkan di aplikasi penggajian berbasis <i>website</i>
2	Rifky Lana Rahardian, Linawati, Made Sudarma, 2018	Majalah Ilmiah Teknologi Elektro, Vol. 17, No. 3	Implementasi Layanan <i>Cloud Computing Software as a Service</i> Pada Usaha Mikro Kecil dan Menengah	Dengan menggunakan SaaS, tidak perlu berinvestasi besar dan membayar hanya jika diperlukan.
3	Ika Nuril Abidah, Mufty Ali Hamdani, Yusuf Amrozi, 2020	Jurnal Sains dan Teknologi, Vol.1 (2), 77-84	Implementasi Sistem Basis Data <i>Cloud Computing</i> pada Sektor Pendidikan	Dapat meningkatkan efisiensi dan efektivitas, Manfaat <i>cloud computing</i> terutama bagi lembaga pendidikan di antaranya ketersediaan yang tinggi, kapasitas penyimpanan yang besar, aksesibilitas yang baik, keamanan data yang terjamin, stabilitas dan sistem yang handal, dan hemat biaya operasional.
4	Mirza Ilhami, 2017	Jurnal IT CIDA Vol. 3 No. 1	Pengenalan Google Firebase Untuk <i>Hybrid Mobile Apps</i> Berbasis Cordova	<i>Firebase</i> sangat cocok untuk digunakan dengan aplikasi <i>hybrid</i> karena menawarkan fitur yang sangat berguna seperti: <i>cloud messaging</i> , autentikasi, <i>database</i>

No	Nama Peneliti & Tahun Publikasi	Publikasi	Judul dari Penelitian	Hasil dari Penelitian
				waktu nyata, pengarsipan, dan lainnya
5	Irfan Mahendra, Deny Tresno Eby Yanto ,2018	Jurnal Teknologi dan Open Source	<i>Agile Development Methods</i> Dalam Pengembangan Sistem Informasi Pengajuan Kredit Berbasis Web (Studi Kasus: Bank Bri Unit Kolonel Sugiono)	Penerapan <i>agile development methods</i> , khususnya model <i>scrum</i> , dalam pengembangan sistem informasi pengajuan kredit berbasis web pada bank BRI Unit Kolonel Sugiono,
6	Ninda Lutfiani, Eka Purnama Harahap, Qurotul Aini, Alfian Dimas Ahsanul Rizki Ahmad, Untung Rahardja, 2020	InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)	Inovasi Manajemen Proyek <i>I-Learning</i> Menggunakan Metode <i>Agile</i> Scrumban	Dengan <i>agile</i> , inovasi dapat ditingkatkan melalui produktivitas tinggi dan kemudian melalui disiplin yang kuat dalam tim yang memberikan nilai bisnis dengan keterlibatan pelanggan langsung. Tidak hanya melalui kerja tim yang lebih baik, kolaborasi yang lebih baik, dan komunikasi yang lebih baik, yang mengarah pada produk berkualitas tinggi.
7	Tohirin Tohirin, Sri Farida Utami, Septian Rheno Widiyanto, Widhy Al Mauludyansah, 2020	Jurnal Multimedia dan virtual <i>Networking Informatics</i>	Implementasi <i>DevOps</i> pada Pengembangan Aplikasi e-Skrining Covid-19	<i>DevOps</i> menyederhanakan proses pengembangan perangkat lunak menjadi satu antarmuka web menggunakan alat <i>gitlab</i> .

Perbedaan penelitian terdahulu dengan penelitian penulis yaitu, penulis mengkombinasikan metode *agile* dengan *framework scrum* dan juga metode *devops*. Selain itu juga menggunakan metode *agile* dengan *framework scrum* agar pengembangan saat *development* bisa menjadi lebih cepat, bagus, dan biaya yang efektif. Metode *devops* juga dapat mempersingkat atau mempercepat waktu antara pengoperasian dan pengembangan *software* tanpa mengurangi kualitas dari *software* (Erich, Amrit, dan Daneva, 2017). Penulis juga menerapkan *cloud computing* SaaS dan BaaS untuk perancangan arsitektur CV. Monkey Teknologi Indonesia.

BAB III

OBJEK DAN METODELOGI PENELITIAN

3.1 CV. Monkey Teknologi Indonesia

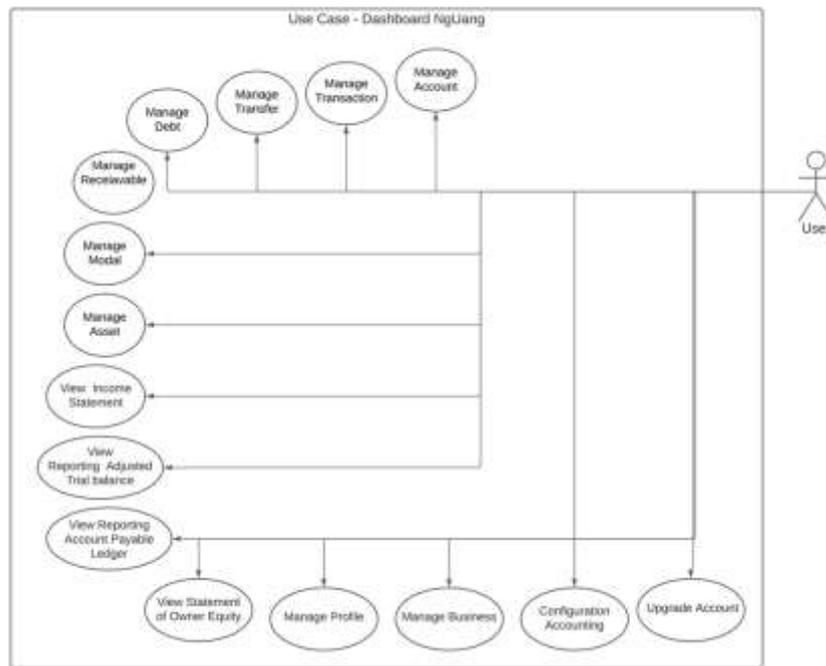
CV. Monkey Teknologi Indonesia berdirinya sejak Mei 2020. *Startup* ini berfokus pada pembuatan *platform* terbaik untuk meningkatkan bisnis dan membantu dalam segala urusan kebutuhan sehari-hari dengan sistem aplikasi. CV. Monkey Teknologi Indonesia sendiri saat ini sudah memiliki beberapa produk salah satunya yaitu NgUang (*Monkey Admin*, 2020):

NgUang merupakan salah satu produk yang berfokus pada layanan pencatatan keuangan untuk pribadi maupun bisnis. NgUang sendiri menerapkan *cloud computing* model SaaS dan juga BaaS dan memiliki fitur utama seperti kelola bisnis atau perusahaan multi, kelola pendapatan, kelola biaya hutang piutang, kelola modal, kelola aset, *transfer* rekening (bank / tunai / dompet), laporan laba rugi, laporan jurnal umum, laporan perubahan modal, dan laporan neraca keuangan. NgUang tersedia versi *website* dan juga versi *android* untuk penggunaannya.

3.1.1 Platform yang sudah berjalan

Saat ini aplikasi NgUang sudah berjalan dengan berbasis *website* (*dashboard* NgUang) yang dapat digunakan oleh *customer* secara gratis dengan *use case diagram* pada gambar 3.1 dan *sitemap* pada gambar 3.2.

1. Use Case Diagram Dashboard NgUang



Gambar 3.1 Use Case Diagram Dashboard NgUang

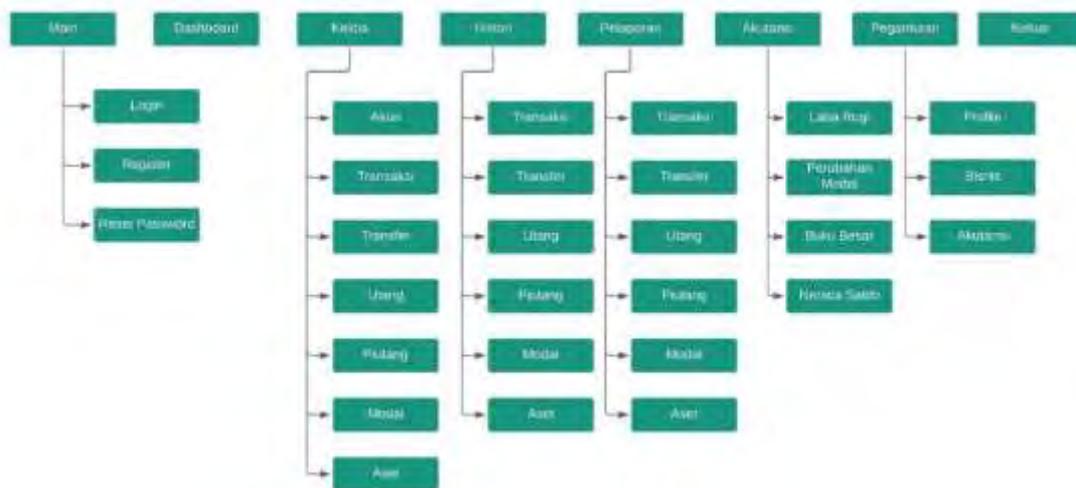
Use case diagram dashboard NgUang memiliki satu aktor yaitu user, user dapat register terlebih dahulu, dan login untuk dapat mengelola fitur dashboard NgUang yang ada seperti pada tabel 3.1.

Tabel 3.1 Fitur Dashboard

No	Fitur	Deskripsi
1	Manage Account	User dapat mengelola account/wallet.
2	Manage Transaction	User dapat mengelola transaksi masuk dan transaksi keluar.
3	Transfer	User dapat melakukan transfer antar account/wallet.
4	Manage Debt	User dapat mengelola utang.
5	Manage Receivable	User dapat mengelola piutang.
6	Modal	User dapat mengelola modal.
7	Asset	User dapat mengelola asset perusahaan.

No	Fitur	Deskripsi
8	<i>Business</i>	User dapat mengelola <i>business</i> .
9	<i>Configuration Accounting</i>	User dapat mengelola <i>format</i> uang sesuai kebutuhan.
10	<i>Profile</i>	User dapat mengelola informasi pribadi.
11	<i>Upgrade Account</i>	User dapat upgrade ke akun <i>premium</i> .
12	<i>Reporting Income Statement</i>	User dapat melihat laporan laba rugi.
13	<i>Reporting Adjusted Trial Balance</i>	User dapat melihat laporan neraca saldo.
14	<i>Reporting Statement of Owner Equity</i>	User dapat melihat laporan penyesuaian modal.
15	<i>Reporting Account Payable Ledger</i>	User dapat melihat laporan kas besar.

2. Sitemap Dashboard Website



Gambar 3.2 Sitemap Dashboard Website

Website dashboard NgUang memiliki 8 menu utama yaitu:

- Menu *main*, user dapat *login*, *register* dan *reset password*.
- Menu *dashboard*, user dapat melihat ringkasan semua fitur di *dashboard*.

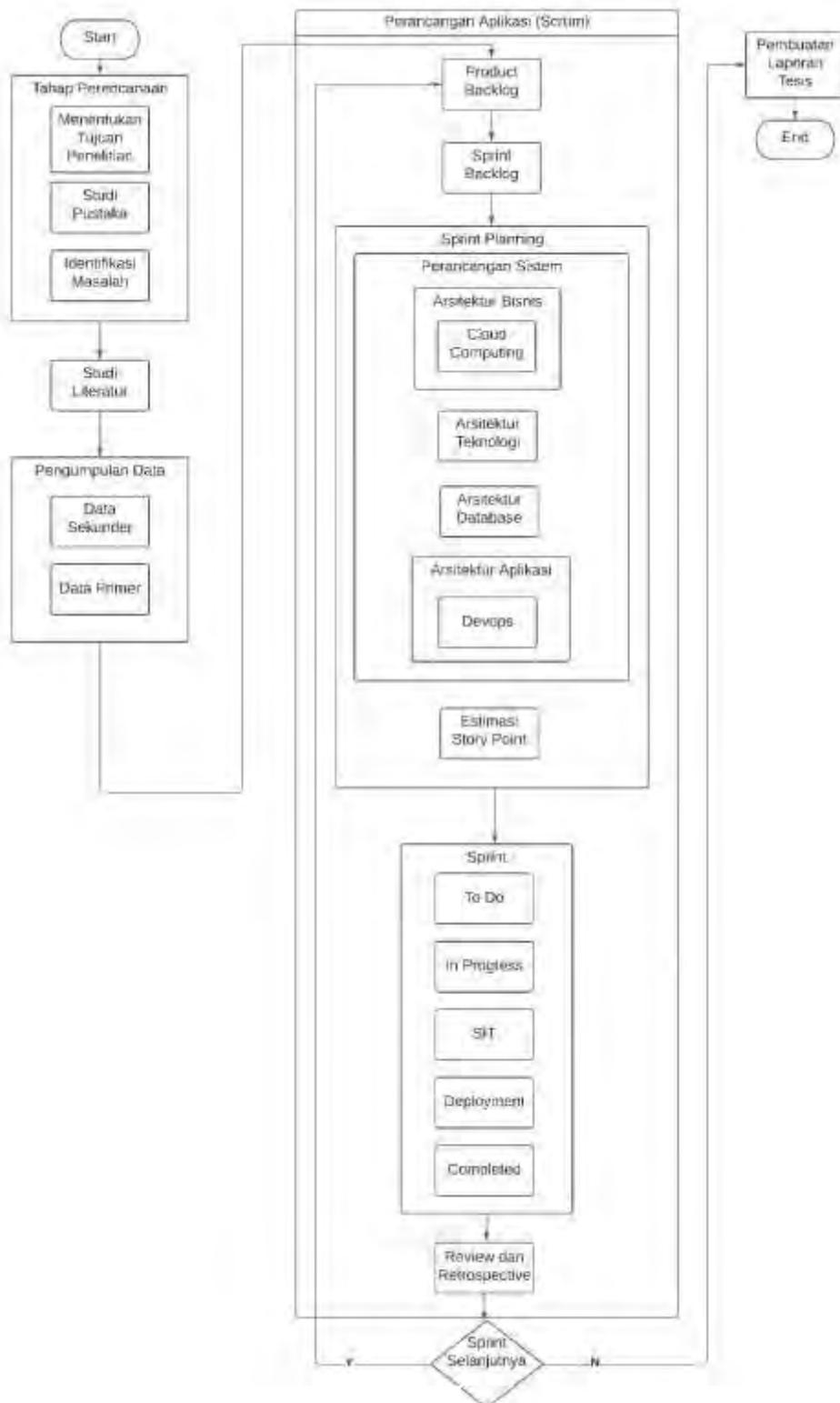
- c. Menu kelola, *user* dapat mengelola akun, transaksi, *transfer*, utang, piutang, modal dan *asset*.
- d. Menu histori, *user* dapat melihat histori transaksi, *transfer*, utang, piutang, modal dan *asset*.
- e. Menu laporan, *user* dapat melihat laporan transaksi, transfer, utang, piutang, modal dan *asset*.
- f. Menu akuntansi, *user* dapat melihat laporan kas besar, laba rugi, perubahan modal dan neraca saldo.
- g. Menu pengaturan, *user* dapat mengelola *profile*, dan bisnis.
- h. Menu keluar aplikasi.

3.1.2 Platform yang akan dikembangkan

Pada *platform* yang sudah berjalan, perlu adanya pengembangan pembuatan *website landing* yang berfungsi sebagai media informasi kepada *customer* atau dengan fungsi utama sebagai media *marketing*. Selain itu, butuh juga pengembangan *website guide* sebagai media panduan penggunaan kepada *customer*, lalu *website panel* yang berfungsi untuk pihak internal NgUang sebagai tempat mengelola data-data yang ditampilkan di *website landing* ataupun *website guide*, dan yang terakhir pengembangan *website dashboard* ke versi *mobile apps* untuk meningkatkan kepuasan *customer* dan memperluas jangka saing.

3.2 Kerangka Penelitian

Pada bab ini penulis akan menjelaskan bagaimana cara melakukan penelitian dengan rincian alat bantu analisis dan bahan serta materi dan urutan alur penelitian yang dibuat secara sistematis. Sehingga dapat menjadi pedoman dalam pemecahan masalah yang akan dihadapi dan melakukan perancangan arsitektur. Berikut alur penelitian yang sudah peneliti buat yang dapat dilihat pada gambar 3.3.



Gambar 3.3 Kerangka Penelitian

3.3 Tahap Perencanaan

Sebelum merancang arsitektur, dimulai dengan kebijakan dan perencanaan. Tanpa perencanaan yang tepat, desain arsitektur tidak akan memiliki efek yang diinginkan. Tahap perencanaan merupakan panduan dalam pengembangan sistem. Inilah yang perlu dilakukan untuk rencana tersebut:

1. Menentukan judul, tujuan yang ingin dicapai, batasan masalah atau ruang lingkup penyelidikan, dan sistem pembuatan laporan.
2. Rencana pengumpulan data (waktu pengumpulan data).
3. Mengidentifikasi data yang diperlukan untuk desain arsitektur.
4. Merancang alat pengumpulan data (berupa wawancara).

3.4 Studi Literatur

Pada tahap ini, dilakukan pencarian referensi teori yang relevan dengan kasus atau permasalahan mengenai perancangan arsitektur dari server, aplikasi dan bisnis, dan juga mencari teori yang relevan dengan penelitian yang peneliti ambil yaitu metode agile dengan framework scrum dan juga metode devops agar perancangan yang dilakukan dapat lebih baik dari penelitian yang sudah ada.

3.5 Tahap Pengumpulan Data

Terdapat dua jenis teknik pengumpulan data yang dilakukan oleh peneliti, antara lain:

1. Data sekunder, yaitu data yang diperoleh langsung dari subjek penelitian, dapat digunakan sebagai tinjauan pustaka, buku, dan hasil-hasil yang mendukung rancangan yang dibuat oleh peneliti dan tinjauan penelitian yang ada.
2. Data primer, yaitu data yang diperoleh dari dokumen berupa data yang dilakukan pada observasi. Observasi adalah suatu metode penggalan data dengan memeriksa catatan dokumen pendukung data tambahan.

3.6 Implementasi Scrum

Pada tahap ini dilakukan implementasi *scrum* sebagai berikut:

1. *Product backlog* yaitu harus mampu menganalisa atau menentukan apa saja yang dibutuhkan atau diperlukan untuk produk.
2. *Sprint backlog* yaitu perencanaan *sprint* dilakukan pada saat pertemuan antara *customer* dengan tim *developer*. Tim pengembangan bekerja sama untuk memilih *backlog* untuk disertakan dalam *sprint*. Hasil pertemuan ini disebut *sprint backlog*.
3. Aktivitas *sprint* adalah periode di mana suatu produk dapat dikembangkan sesuai dengan daftar persyaratan. *Sprint* terdiri dari dua bagian yaitu:
 - a. *Meeting* atau pertemuan adalah rapat harian yang diadakan oleh tim pengembang untuk membahas kegiatan dalam proses pengembangan.
 - b. *Reflection sprint* adalah pertemuan bulanan untuk membahas *sprint backlog* saat ini dan meningkatkan banyak fitur untuk meningkatkan kualitas produk.
4. *Sprint review* adalah kegiatan untuk memverifikasi fungsi-fungsi yang diimplementasikan untuk memastikan bahwa fungsi-fungsi yang diimplementasikan berjalan dengan benar.
5. *Sprint retrospective* ialah proses dimana tim melakukan *review* kembali untuk perbaikan dan untuk meningkatkan *sprint* berikutnya. Selain itu, *sprint retrospective* dilakukan untuk melihat kembali bagaimana pekerjaan berjalan di *sprint* sebelumnya.

3.7 Perancangan Arsitektur

Pada saat perancangan sistem terbagi menjadi beberapa bagian tahap dalam perancangan arsitekturnya. Terdapat empat perancangan arsitektur sebagai berikut:

1. Perancangan arsitektur *database*

Pada perancangan ini dilakukan perancangan *database* dengan konsep NoSQL dengan menggunakan *firebase*. Dengan menggunakan *firebase* memiliki fitur unggulan seperti *realtime database*. Perancangan *database* juga sudah meliputi perancangan dengan model SaaS. Dengan adanya *product* baru ataupun *client* baru pada NgUang, tidak perlu merubah skema *database* yang ada dan data antar *product* dapat saling terintegrasi.

2. Perancangan arsitektur bisnis

Pada perancangan ini dijelaskan bagaimana bisnis dapat lebih *efisien* dengan menerapkan *cloud computing* SaaS dan juga BaaS. Model *cloud computing* SaaS menjadi pilihan bagi pengguna atau *client* untuk menggunakan produk NgUang dengan hemat biaya, lebih cepat waktu pengerjaan, tidak perlu memikirkan biaya *maintenance*, hanya cukup membayar biaya bulanan/tahunan kepada CV. Monkey Teknologi Indonesia. *Cloud computing* BaaS menjadi kemudahan bagi CV. Monkey Teknologi Indonesia sebagai *startup* yang baru tanpa harus memikirkan memiliki *server*, dengan BaaS, semua data dan *server* menggunakan *cloud* dari *firebase* dengan kelebihan lebih mudah, tidak mengeluarkan biaya yang besar.

3. Perancangan arsitektur aplikasi

Pada perancangan ini dilakukan perancangan aplikasi berupa diagram *use case diagram*, *site map* aplikasi, lalu penerapan *devops* dilakukan saat perancangan arsitektur aplikasi. Implementasi *devops* yang peneliti gunakan yaitu gitlab sebagai perangkat git untuk menghubungkan otomatis *deployment* ke *server* secara otomatis ke *server Firebase*. Implementasi *devops* ini dapat mempersingkat waktu pengembangan sistem dari waktu ke waktu. *Firebase hosting* sebagai layanan *hosting* yang dapat dikelola penuh untuk konten statis dan dinamis bahkan *microservice*. *Firebase hosting* sendiri sudah dilengkapi SSI agar konten akan selalu dikirim dengan aman. *Firebase hosting* dapat diintegrasikan dengan gitlab dan penggunaannya juga sangat mudah. Maka dari itu penulis menggunakan layanan ini.

4. Perancangan Arsitektur Teknologi

Pada perancangan ini dilakukan perancangan teknologi apa saja yang digunakan *product* NgUang dan bagaimana semuanya bisa saling terintegrasi dari *server*, *code*, *devops*, dan *database* yang akan dijelaskan melalui sebuah gambar.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Aplikasi

Dalam perancangan aplikasi, peneliti menerapkan metode *scrum agile* agar dalam perancangannya dan saat *development* bisa menjadi lebih cepat, bagus, dan menggunakan biaya yang efektif. Maka dari itu, sebelum dilakukan perancangan, peneliti membuat *product backlog* terlebih dahulu agar mengetahui seberapa luas cakupan sistem yang akan dirancang. Sebelum menentukan *product backlog*, peneliti menentukan terlebih dahulu berapa banyak *developer* yang ada, lalu menentukan berapa lama 1 *sprint* berjalan dan juga menentukan tingkat kesulitan dari *story point*.

4.2 Implementasi Scrum

4.2.1 Scrum Team

Scrum team untuk membuat aplikasi NgUang hanya berjumlah tiga orang dikarenakan sebuah perusahaan ini ialah perusahaan baru atau *startup* yang baru di rintis dengan tidak adanya *income* sama sekali. Tim *scrum* yang kecil idealnya yaitu sedikitnya 2 orang dan sebanyaknya 7 orang, jika lebih dari itu biasanya dipisah menjadi *multiple scrum team*. Harapan dari *scrum* sendiri bukan terfokus pada satu skill dalam satu orang, tetapi setiap individu harus memiliki keterampilan yang seimbang agar mendapat kinerja terbaik dan optimal. Berdasarkan hal tersebut, *scrum team* akan mampu menghadapi tantangan yang selalu berubah. Hal ini juga berarti dalam *Scrum framework* tidak dibedakan antara "*tester*" dan "*arsitek*". Semua *role* yang sama yaitu "*anggota scrum team*". Bahkan, jika keterampilan utamanya bukan untuk mengembangkan kode produksi atau membuat sebuah program. Maka, disusunlah *scrum team* dipenelitian ini yang dijelaskan pada tabel 4.1.

Tabel 4.1 Scrum Tim

No	Scrum Team	Role
1	Ali Abdul Wahid	Scrum product owner, scrum team member (software engineer, QA, devops)
2	Uki Satriana	Scrum team member (software engineer, QA, devops)
3	Harry Nuril Anwar	Scrum team member (softwareeEngineer, QA, devops)

4.2.2 Story Point

Story point sendiri digunakan untuk setiap *product backlog* pada *sprint*. Story point berfungsi untuk memberikan estimasi dari cakupan *backlog*, agar memudahkan tim untuk mengukur kemampuan tim dan tugas yang akan dilakukan/diselesaikan. Agar memudahkan *range* dari story point peneliti juga membagi story point menjadi 3 point seperti pada tabel 4.2.

Tabel 4.2 Story Point

No	Point	Deskripsi	Estimasi
1	3	Sangat mudah	3 jam kerja
2	5	Mudah	5 jam kerja
3	8	Normal	8 jam kerja
4	16	Sulit	16 jam kerja
5	24	Sulit sekali	24 jam kerja

Dengan story point yang dijelaskan pada tabel, dapat ditarik perhitungan 1 story point sama dengan 1 jam kerja. Dengan estimasi ini akan sangat memudahkan *scrum team* dalam mengambil *task* saat *sprint* berjalan.

4.2.3 Product Backlog

Peneliti membuat 7 item *product backlog* utama dalam pembuatan aplikasi NgUang yang digambarkan pada gambar 4.1 dengan total story point 664 story point, alias 664 jam kerja

dengan *standard* jam kerja sehari 8 jam kerja dan dilakukan oleh 3 orang, yang berarti estimasi dari seluruh *product backlog* yaitu $664 / 8 / 3$ yang hasilnya 27 hari dimana 1 minggu 5 hari kerja dan dapat disimpulkan untuk menyelesaikan *product backlog* ini kurang lebih diselesaikan dalam 5 minggu.



Gambar 4.1 *Product Backlog*

Dari 7 *product backlog* yang ada peneliti membagi menjadi lima *sprint* dan dikelompokan agar setiap *sprint* nya memiliki *load task* yang seimbang seperti pada table 4.3.

Tabel 4.3 *Timeline Sprint*

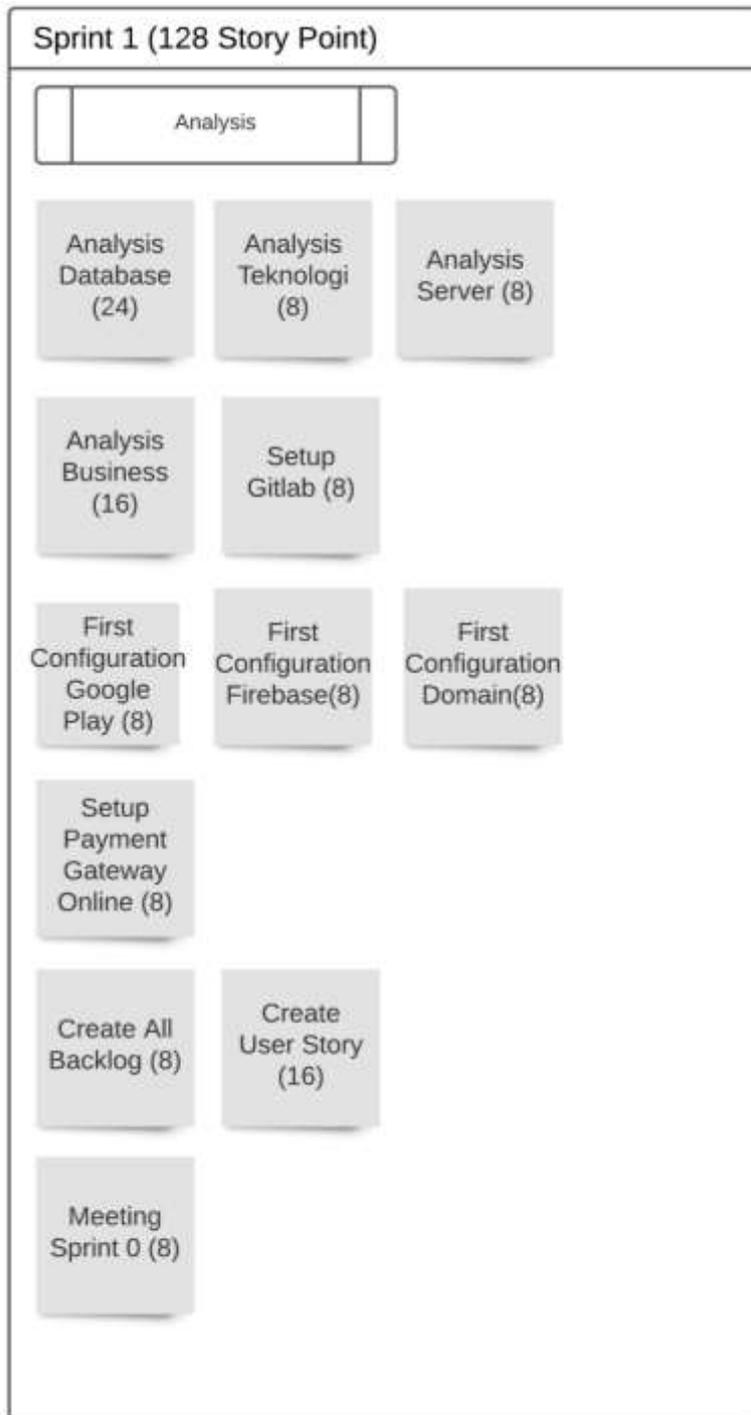
No	Sprint				
Week	1	2	3	4	5
Story Point	128	139	133	113	128
Task	<i>Analysis & Setup Apps</i>	<i>Integration Payment Gateway Online</i>	<i>Mobile Dashboard for User</i>	<i>Website Landing</i>	<i>Testing</i>
		<i>Website Panel for Admin</i>		<i>Website Guide</i>	
				<i>Reporting</i>	

4.2.4 *Sprint Backlog*

Sprint backlog berisi fitur-fitur hasil seleksi dari *product backlog* yang akan di selesaikan pada *sprint*. Peneliti membagi menjadi 5 *sprint* dengan catatan 1 *sprint* berlangsung dalam 1 minggu atau 5 hari kerja.

1. Perancangan *sprint* 1

Perancangan pada *sprint* 1 digambarkan pada gambar 4.2. dan *user story* dijelaskan pada tabel 4.4.



Gambar 4.2 Perancangan *Sprint 1*

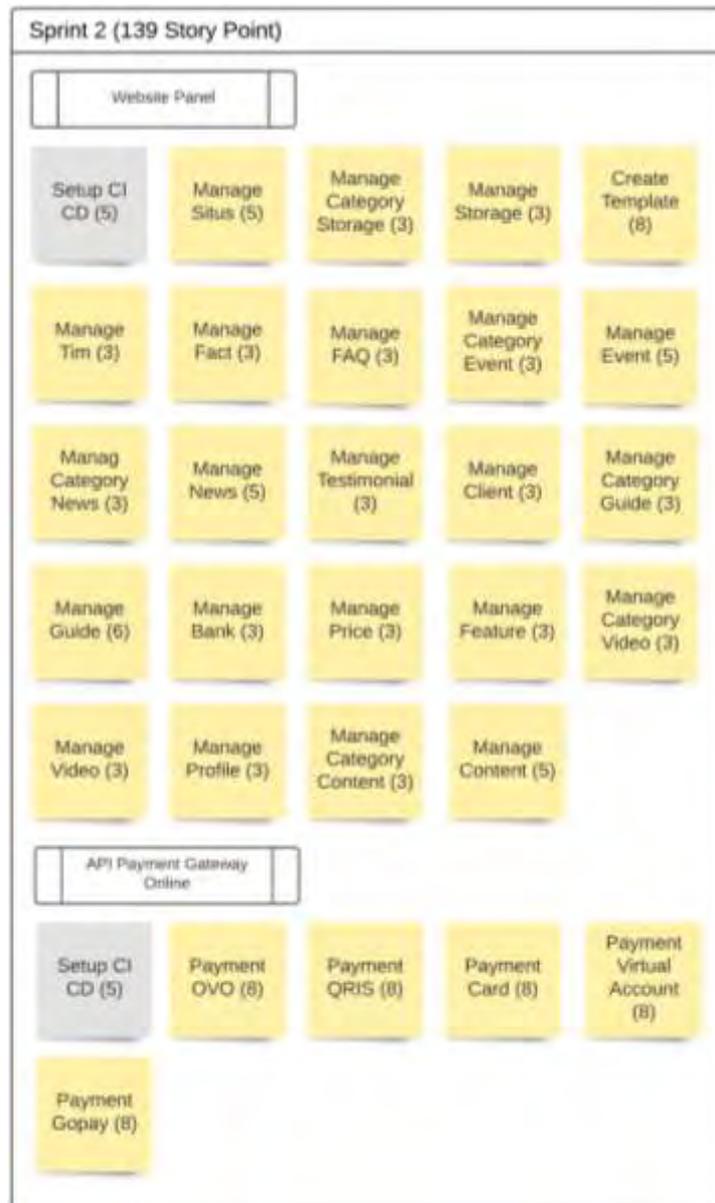
Pada *sprint 1* berfokus pada *analysis system* terlebih dahulu, lalu beberapa konfigurasi awal seperti konfigurasi *remote git*, *google play* untuk *publishing apps*, *firebase*, dan *domain* untuk konfigurasi *hosting* dan membuat semua *backlog* dan *user story*.

Tabel 4.4 User Story Sprint 1

No	Task	Detail Task
1	<i>Analysys Database</i>	<i>Scrum team member</i> melakukan <i>analysis database</i> terkait aplikasi NgUang yang sudah dapat mencakup semua <i>website</i> dan <i>mobile apps</i> yang akan dibuat.
2	<i>Analysis Teknologi</i>	<i>Scrum team member</i> melakukan <i>analysis</i> teknologi dengan memperkirakan teknologi yang sedang <i>trend</i> dan mudah untuk melakukan <i>maintenance</i> di jangka panjang
3	<i>Analysis Server</i>	<i>Scrum team member</i> melakukan <i>analysis server</i> dengan menganalisis <i>server</i> yang mudah diterapkan dengan biaya yang murah untuk perusahaan baru/ <i>startup</i> baru.
4	<i>Analysis Business</i>	<i>Scrum team member</i> melakukan <i>analysis business</i> dengan menganalisis <i>cloud computing</i> apa yang bisa diterapkan untuk <i>startup</i> baru agar <i>profit</i> yang dihasilkan dapat maksimal
5	<i>Setup Gitlab</i>	<i>Scrum team member</i> melakukan <i>setup</i> pada <i>remote git</i> agar proses <i>development</i> tim menjadi lebih mudah dan cepat.
6	<i>First Configuration Google Play</i>	<i>Scrum team member</i> melakukan <i>configuration</i> untuk persiapan <i>publish</i> aplikasi <i>mobile</i> NgUang.
7	<i>First Configuration Firebase</i>	<i>Scrum team member</i> melakukan <i>configuration</i> untuk menyambungkan domain yang ada ke <i>server hosting firebase</i> , <i>setting auth login</i> yang akan digunakan dan <i>setup database rule</i> yang ada.
8	<i>First Configuration Domain</i>	<i>Scrum team member</i> melakukan pembelian domain dan konfigurasi <i>dns domain</i> untuk disambungkan dengan <i>server firebase</i> .
9	<i>Setup Payment Gateway Online</i>	<i>Scrum team member</i> menentukan <i>payment gateway online</i> yang akan digunakan dan cocok dengan bisnis yang ada dan melakukan konfigurasi untuk versi <i>sandbox</i> dan <i>production</i> .
10	<i>Create All Backlog</i>	<i>Scrum product owner</i> membuat <i>product backlog</i> dan <i>task-task</i> pada <i>product backlog</i> untuk memudahkan <i>scrum team member</i> saat proses <i>sprint</i> berjalan.
11	<i>Create User Story</i>	<i>Scrum product owner</i> membuat <i>user story</i> untuk memudahkan pengerjaan di setiap <i>task</i> yang sudah dibuat dengan detailnya.
12	<i>Meeting Sprint 0</i>	<i>Scrum team member</i> melakukan <i>meeting</i> untuk membahas semua <i>task-task</i> dan <i>user story</i> yang ada dengan detail agar saat proses <i>development</i> sudah mengetahui apa yang harus dikerjakan dan harus di selesaikan

2. Perancangan *Sprint 2*

Perancangan pada *sprint 2* digambarkan pada gambar 4.3 dan *user story* dijelaskan pada tabel 4.4.



Gambar 4.3 Perancangan *Sprint 2*

Pada *sprint* ini berfokus pada *integration payment gateway online* untuk *user* melakukan pembayaran pada NgUang dan *website panel* yang digunakan *admin* untuk mengelola *website landing* yang berisi informasi-informasi mengenai NgUang untuk *user*.

Tabel 4.5 User Story Sprint 2

No	Task	Detail Task
1	Setup CI CD Website Panel	Scrum team member melakukan setup auto CI/CD sebelum melakukan <i>development</i> pada aplikasi. Konfigurasi dilakukan dengan membuat <i>code git</i> yang dapat secara otomatis setiap adanya perubahan pada <i>git remote</i> akan otomatis <i>terbuild</i> pada server <i>firebase</i> .
2	Manage Situs	Sebagai <i>admin</i> dapat mengelola informasi-informasi mengenai situs NgUang seperti nama, deksripsi, logo, alamat, <i>social media</i> , dan kontak dari NgUang.
3	Manage Category Storage	Sebagai <i>admin</i> dapat mengelola kategori <i>storage</i> .
4	Manage Storage	Sebagai <i>admin</i> dapat mengelolas <i>storage</i> . Di fitur ini semua yang berhubungan dengan file, dokumen, gambar, video dapat dikelola pada menu ini.
5	Create Template	Scrum team member membuat <i>template</i> UI untuk <i>website panel</i> , dari standarisasi <i>layout</i> , <i>button</i> , warna, tema, dll.
6	Manage Team	Sebagai <i>admin</i> dapat mengelola daftar tim dari NgUang yang nanti ditampilkan pada halaman <i>website landing</i> .
7	Manage Fact	Sebagai <i>admin</i> dapat mengelola data data <i>factual</i> seperti jumlah <i>user</i> yang telah daftar atau jumlah <i>user</i> yang mengunjungi <i>website</i> .
8	Manage FAQ	Sebagai <i>admin</i> dapat mengelola <i>FAQ</i> berupa pertanyaan dan jawaban yang nantinya sering ditanyakan akan di kelola disini.
9	Manage Category Event	Sebagai <i>admin</i> dapat mengelola kategori <i>event</i> .
10	Manage Event	Sebagai <i>admin</i> dapat mengelola daftar <i>event</i> agar pengunjung <i>website</i> bisa mengetahui kegiatan yang akan dating dan histori kegiatan yang ada pada NgUang.
11	Manage Category News	Sebagai <i>admin</i> dapat mengelola kategori <i>news</i> .
12	Manage News	Sebagai <i>admin</i> dapat mengelola berita yang sedang <i>trend</i> atau berita yang berhubungan dengan keuangan agar menarik pengunjung NgUang.
13	Manage Testimonial	Sebagai <i>admin</i> dapat mengelola testimoni dari <i>user</i> yang nanti ditampilkan di halaman <i>website landing</i> .
14	Manage Client	Sebagai <i>admin</i> dapat mengelola <i>client</i> dari <i>user</i> yang nanti ditampilkan di halaman <i>website landing</i>

No	Task	Detail Task
15	<i>Manage Category Guide</i>	Sebagai <i>admin</i> dapat mengelola kategori panduan, agar memudahkan <i>user</i> saat melakukan pencarian bahan panduan.
16	<i>Manage Guide</i>	Sebagai <i>admin</i> dapat mengelola panduan panduan yang memudahkan <i>user</i> saat menggunakan <i>platform</i> dari NgUang.
17	<i>Manage Bank</i>	Sebagai <i>admin</i> dapat mengelola daftar bank yang dimiliki NgUang agar <i>user</i> mengetahui pembayaran manual ke bank yang mana.
18	<i>Manage Pricing</i>	Sebagai <i>admin</i> dapat mengelola daftar harga penggunaan <i>platform</i> NgUang.
19	<i>Manage Feature</i>	Sebagai <i>admin</i> dapat mengelola daftar fitur yang ada pada <i>platform</i> NgUang.
19	<i>Manage Category Video</i>	Sebagai <i>admin</i> dapat mengelola kategori <i>video</i> .
20	<i>Manage Video</i>	Sebagai <i>admin</i> dapat mengelola <i>video</i> .
21	<i>Manage Profile</i>	Sebagai <i>admin</i> dapat merubah informasi pribadi mengenai <i>admin</i> sendiri dan dapat mereset password untuk <i>login</i> .
22	<i>Manage Category Content</i>	Sebagai <i>admin</i> dapat mengelola kategori <i>content</i> .
23	<i>Manage Content</i>	Sebagai <i>admin</i> dapat mengelola konten yang dinamis.

3. Perancangan *Sprint* 3

Perancangan pada *sprint* 3 digambarkan pada gambar 4.4 dan *user story* dijelaskan pada tabel 4.6.



Gambar 4.4 Perancangan *Sprint 3*

Sprint ini berfokus pada pembuatan *mobile dashboard* untuk *user* atau pengembangan *website dashboard* yang sudah ada, agar memudahkan segmen pasar pelanggan dimana saat ini pengguna *mobile* sudah tinggi.

Tabel 4.6 *User Story Sprint 3*

No	Task	Detail Task
1	<i>Setup CI CD Mobile Dashboard</i>	<i>Scrum team member</i> melakukan <i>setup auto CI/CD</i> sebelum melakukan <i>development</i> pada aplikasi. Konfigurasi dilakukan dengan membuat <i>code git</i> yang dapat secara otomatis setiap adanya perubahan pada <i>git remote</i> akan otomatis <i>terbuild</i> pada <i>ionic app flow</i> .
2	<i>Login</i>	Sebagai <i>user</i> dapat melakukan <i>login</i> dengan menggunakan email dan <i>password</i> yang sudah terdaftar.

No	Task	Detail Task
3	<i>Register</i>	Sebagai <i>user</i> dapat mendaftar pada aplikasi NgUang secara gratis dengan fitur yang terbatas.
4	<i>Reset Password</i>	Sebagai <i>user</i> dapat melakukan <i>reset password</i> . Setelah <i>user</i> melakukan <i>reset password</i> , <i>server</i> akan mengirim email ke <i>user</i> untuk memasukkan <i>password</i> yang baru.
5	<i>Dashboard</i>	Sebagai <i>user</i> dapat melihat jumlah saldo dari akun dompet yang ada.
6	<i>Manage Account</i>	Sebagai <i>user</i> dapat mengelola akun akun dompet.
7	<i>Manage Transaction</i>	Sebagai <i>user</i> dapat mengelola pendapatan dan pengeluaran keuangan.
8	<i>Transfer</i>	Sebagai <i>user</i> dapat memindahkan uang dari satu akun ke akun lainnya.
9	<i>Create Template</i>	<i>Scrum team member</i> membuat <i>template</i> UI untuk <i>website panel</i> , dari standarisasi <i>layout</i> , <i>button</i> , warna, tema, dll
10	<i>Configuration Accounting</i>	Sebagai <i>user</i> dapat mengubah <i>format</i> dari keuangan di aplikasi.
11	<i>Manage Debt</i>	Sebagai <i>user</i> dapat mengelola utang keuangan.
12	<i>Manage Receivable</i>	Sebagai <i>user</i> dapat mengelola piutang keuangan.
13	<i>Manage Profile</i>	Sebagai <i>user</i> dapat mengelola data pribadi
14	<i>Manage Business</i>	Sebagai <i>user</i> dapat mengelola banyak bisnis yang ada dengan keuangan yang terpisah.
15	<i>Manage Modal</i>	Sebagai <i>user premium</i> dapat mengelola modal keuangan.
16	<i>Upgrade Account</i>	Sebagai <i>user</i> dapat beralih ke <i>user premium</i> agar semua fitur terbuka.
17	<i>Manage Asset</i>	Sebagai <i>user premium</i> dapat mengelola aset keuangan.
18	<i>History Transfer</i>	Sebagai <i>user</i> dapat melihat histori transaksi transfer.
19	<i>History Receivable</i>	Sebagai <i>user</i> dapat melihat histori transaksi piutang.
20	<i>History Debt</i>	Sebagai <i>user</i> dapat melihat histori transaksi utang.
21	<i>History Modal</i>	Sebagai <i>user premium</i> dapat melihat histori transaksi modal.
22	<i>History Asset</i>	Sebagai <i>user premium</i> dapat melihat histori transaksi aset.

No	Task	Detail Task
23	History Transaction	Sebagai <i>user</i> dapat melihat histori pendapatan dan pengeluaran keuangan.

4. Perancangan *Sprint* 4

Perancangan pada *sprint* 4 digambarkan pada gambar 4.5 dan *user story* dijelaskan pada tabel 4.7.



Gambar 4.5 Perancangan *Sprint* 4

Sprint ini berfokus pada pembuatan *reporting* keuangan untuk *user* dan membuat *website landing* untuk memudahkan memberi informasi kepada *user*.

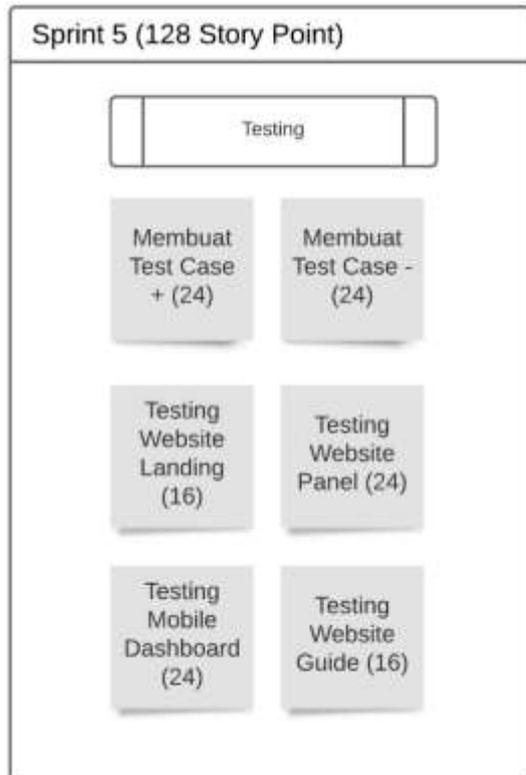
Tabel 4.7 User Story Sprint 4

No	Task	Detail Task
1	Setup CI CD Website Guide	Scrum team member melakukan setup auto CI/CD sebelum melakukan development pada aplikasi. Konfigurasi dilakukan dengan membuat code git yang dapat secara otomatis setiap adanya perubahan pada git remote akan otomatis terbuild pada server firebase.
2	Setup CI CD Website Landing	Scrum team member melakukan setup auto CI/CD sebelum melakukan development pada aplikasi. Konfigurasi dilakukan dengan membuat code git yang dapat secara otomatis setiap adanya perubahan pada git remote akan otomatis terbuild pada server firebase.
3	Create Template Website Guide	Scrum team member membuat template UI untuk website panel, dari standarisasi layout, button, warna, tema, dll
4	Statement of Owner Equity	Sebagai user dapat melihat laporan mengenai perubahan modal pada dashboard mobile.
5	Reporting Account Payable Ledger	Sebagai user dapat melihat laporan mengenai buku besar keuangan pada dashboard mobile.
6	Reporting Adjusted Trial Balance	Sebagai user dapat melihat laporan mengenai neraca saldo keuangan pada dashboard mobile.
7	Income Statement	Sebagai user dapat melihat laba rugi keuangan pada dashboard website maupun mobile
8	Category Guide Page	Sebagai user dapat melihat kategori panduan penggunaan aplikasi NgUang.
9	Guide Page	Sebagai user dapat melihat daftar panduan penggunaan aplikasi NgUang.
10	Detail Guide Page	Sebagai user dapat melihat daftar detail panduan penggunaan aplikasi NgUang.
11	Create Template Website Landing	Scrum team member membuat template UI untuk website panel, dari standarisasi layout, button, warna, tema, dll.
12	Home Page	Sebagai user dapat melihat ringkasan informasi mengenai NgUang seperti apa itu NgUang, fitur fitur, screenshot mobile, tim. testimonial, berita terbaru, informasi fact, harga aplikasi, dan klien yang bekerjasama dengan NgUang.
13	Feature Page	Sebagai user dapat melihat detail fitur-fitur yang ada pada Platform NgUang.
14	Pricing Page	Sebagai user dapat melihat detail harga yang ada pada platform NgUang.

No	Task	Detail Task
15	<i>Payment Info Page</i>	Sebagai <i>user</i> dapat melihat daftar bank untuk melakukan pembayaran manual yang ada pada <i>platform</i> NgUang.
16	<i>About Us Page</i>	Sebagai <i>user</i> dapat melihat informasi mengenai NgUang itu apa lalu kontak yang bisa dihubungi dan <i>social media</i> dari NgUang.
17	<i>Our Team Page</i>	Sebagai <i>user</i> dapat melihat daftar tim yang ada pada <i>platform</i> NgUang.
18	<i>Event Page</i>	Sebagai <i>user</i> dapat melihat detail kegiatan yang ada pada <i>Platform</i> NgUang.
19	<i>Inspiration Page</i>	Sebagai <i>user</i> dapat melihat berita yang ada pada <i>platform</i> NgUang.
20	<i>FAQ Page</i>	Sebagai <i>user</i> dapat melihat daftar pertanyaan yang sering ditanyakan tanpa harus menghubungi pihak <i>customer service</i> NgUang
21	<i>Terns of Service Page</i>	Sebagai <i>user</i> dapat melihat syarat dan ketentuan penggunaan <i>platform</i> NgUang
22	<i>Privacy Policy Page</i>	Sebagai <i>user</i> dapat melihat kebijakan penggunaan <i>platform</i> NgUang

5. Perancangan *Sprint* 5

Perancangan pada *sprint* 5 digambarkan pada gambar 4.6 dan *user story* dijelaskan pada tabel 4.8.



Gambar 4.6 Perancangan *Sprint 5*

Pada *sprint* ini berfokus pada pembuatan *test case* dan melakukan *testing* pada semua aplikasi yang sudah dibuat.

Tabel 4.8 *User Story Sprint 5*

No	Task	Detail Task
1	Membuat <i>Test Case +</i>	<i>Scrum team member</i> membuat daftar <i>test case</i> positif untuk melakukan <i>testing</i> pada <i>platform</i> NgUang
2	Membuat <i>Test Case -</i>	<i>Scrum team member</i> membuat daftar <i>test case</i> negatif untuk melakukan <i>testing</i> pada <i>platform</i> NgUang
3	<i>Testing Website Landing</i>	<i>Scrum team member</i> melakukan <i>testing</i> pada <i>website landing</i> dari daftar <i>test case</i> yang sudah dibuat.
4	<i>Testing Website Panel</i>	<i>Scrum team member</i> melakukan <i>testing</i> pada <i>website panel</i> dari daftar <i>test case</i> yang sudah dibuat.
5	<i>Testing Website Guide</i>	<i>Scrum team member</i> melakukan <i>testing</i> pada <i>website guide</i> dari daftar <i>test case</i> yang sudah dibuat.

No	Task	Detail Task
6	<i>Testing Mobile Dashboard</i>	<i>Scrum team member melakukan testing pada mobile dashboard dari daftar test case yang sudah dibuat.</i>

4.2.5 *Sprint*

Pada *sprint* ini dilakukan pembahasan mengenai pekerjaan-pekerjaan yang akan dilakukan selama *sprint* dan dinamakan *sprint planning*, melakukan *daily scrum*, melakukan *sprint review*, dan *retrospective*. *Sprint planning* sendiri sudah dapat dilihat dengan melihat *product backlog* dan *tasks* yang sudah dibuat. *Daily scrum* dilakukan setiap pagi saat memulai pekerjaan, tim *scrum* melakukan *daily standup* apa yang sudah diselesaikan pada hari sebelumnya dan apa yang akan di selesaikan pada hari ini. Dilakukan secara berdiri agar proses *daily scrum* dapat berlangsung dengan cepat dengan maksimal 15 menit.

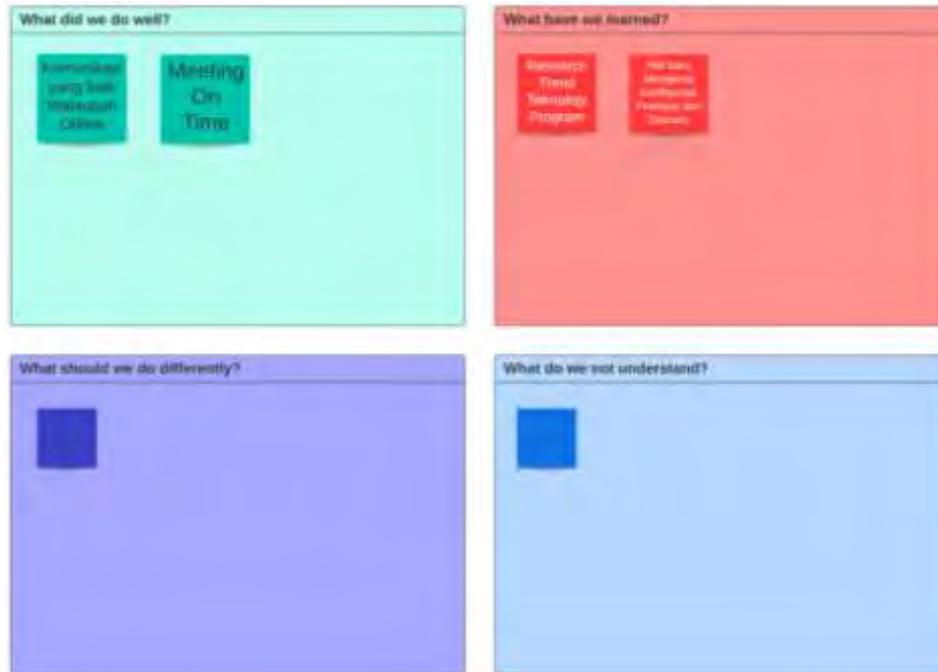
4.2.6 *Sprint Retrospective & Review*

Output dari 7 *sprint* yang sudah di jalankan:

1. Sprint 1

a. *Sprint Retrospective*

Pada gambar 4.7 dijelaskan mengenai hasil dari *sprint retrospective* pada *sprint 1*



Gambar 4.7 *Sprint Retrospective Sprint 1*

Pada *sprint 1* ini yang sudah dilakukan dengan baik yaitu setiap *meeting* selalu *on time* tidak ada yang telat dan komunikasi yang baik walaupun dilakukan secara online lalu belajar mengenai *trend* teknologi saat ini dan belajar mengenai konfigurasi firebase dan domain.

b. *Sprint Review*

Pada *sprint 1* menghasilkan analisis *database* yang dijelaskan di *class diagram* pada *point 4.8.2*, analisis teknologi pada *point 4.4*, arsitektur bisnis pada *point 4.3*, selain itu juga dilakukan nya konfigurasi awal untuk *payment gateway online*, google play, *firebase*, dan juga domain yang dijelaskan pada tabel 4.9.

Tabel 4.9 Tabel *Domain*

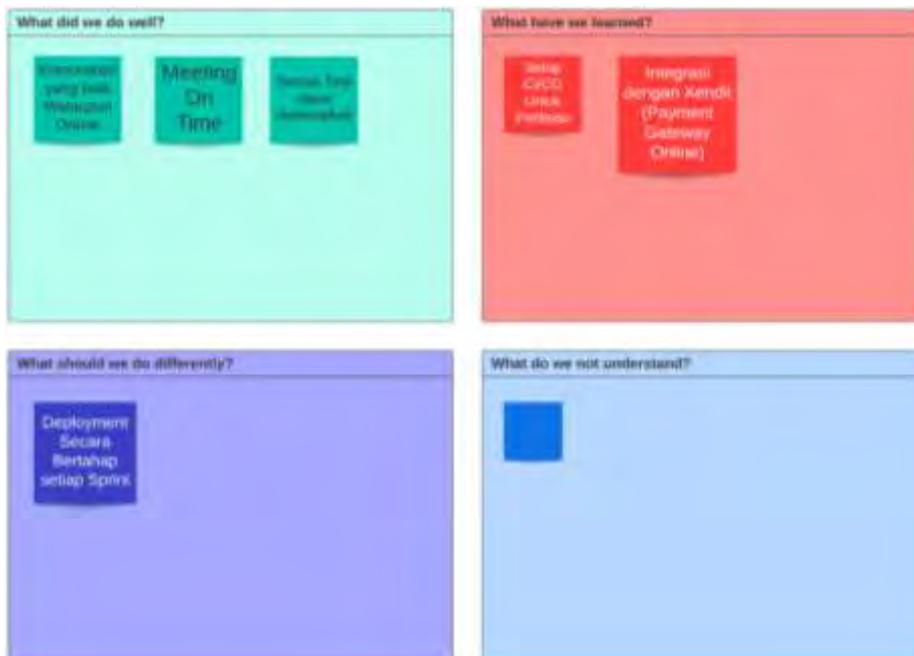
No	Nama	Domain
1	Website Landing SIT	www.sit.nguang.id
2	Website Landing	www.nguang.id
3	Website Panel SIT	www.sit.panel.nguang.id
4	Website Panel	www.panel.nguang.id
5	Website Guide SIT	www.sit.guide.nguang.id
6	Website Guide	www.guide.nguang.id

Environment SIT digunakan untuk kebutuhan *testing* menjadi lebih dengan *database* yang terpisah dengan data *production*, sehingga tidak perlu khawatir untuk melakukan *testing*.

2. Sprint 2

a. *Sprint Retrospective*

Pada gambar 4.8 dijelaskan mengenai hasil dari *sprint retrospective* pada *sprint 2*.

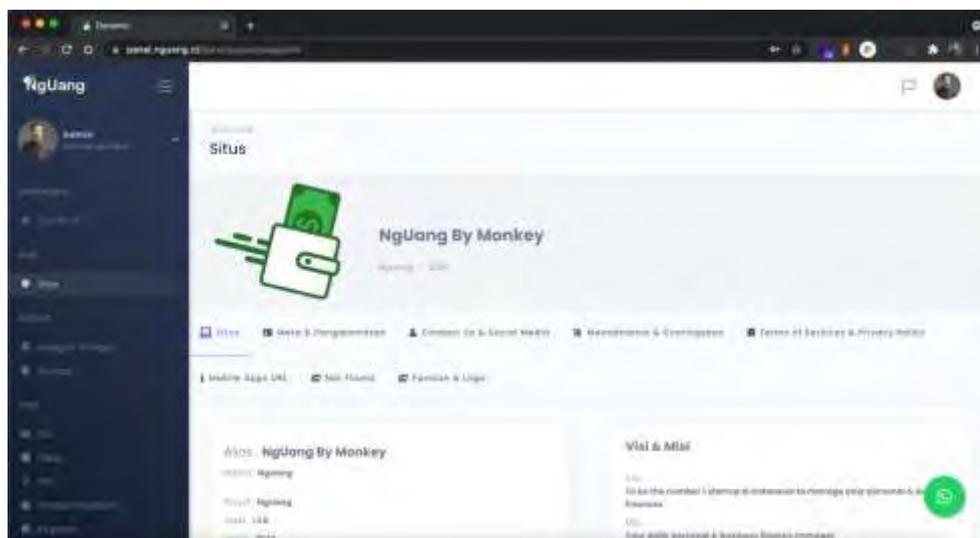


Gambar 4.8 *Sprint Retrospective Sprint 2*

Pada *sprint 2* ini yang sudah dilakukan dengan baik yaitu setiap *meeting* selalu *on time*, tidak ada yang telat dan komunikasi yang baik walaupun dilakukan secara online dan semua *task* dapat diselesaikan. Hal baru yang dipelajari mengenai *setup CI/CD* untuk *firebase* dan integrasi dengan *xendit* untuk kebutuhan *payment gateway online*. Lalu hal yang membuat berbeda yaitu *deployment* dapat dilakukan secara bertahap.

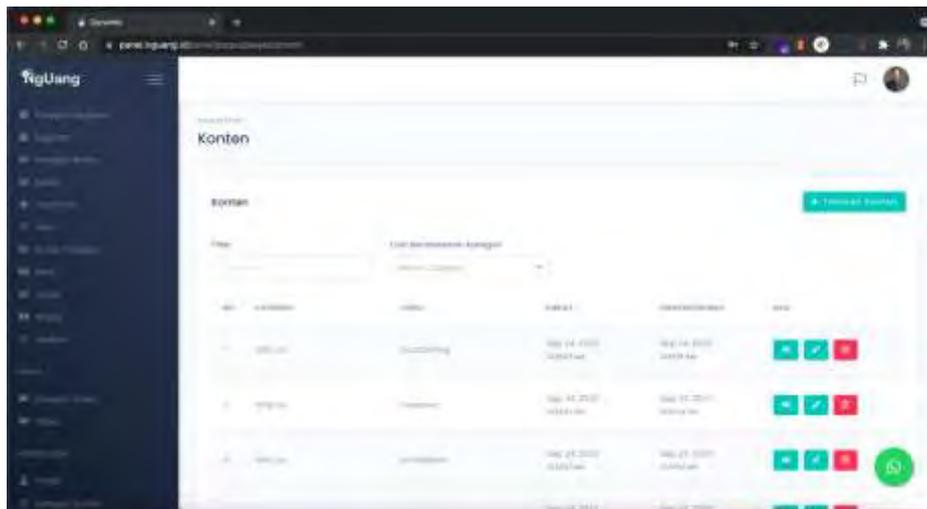
b. *Sprint Review*

Pada *sprint 2* ini dilakukan pembuatan *website panel* dan penerapan *payment gateway online* untuk pembayaran *customer* dan semua *task* dapat diselesaikan. *Website panel* sudah dapat diakses melalui www.sit.panel.nguang.id seperti yang terlihat pada *screenshot* tampilan gambar 4.9 dan 4.10. Selesaiannya *website panel* ini, tim *internal* dapat mengelola semua kebutuhan data yang nantinya akan ditampilkan di *website guide* dan juga *landing*. Setelah itu, penerapan pembayaran *online customer* bisa dilakukan melalui Gopay, OVO, *virtual account*, QRIS dan juga kartu. dan *customer* dapat dengan mudah melakukan pembayaran secara online seperti gambar 4.11.



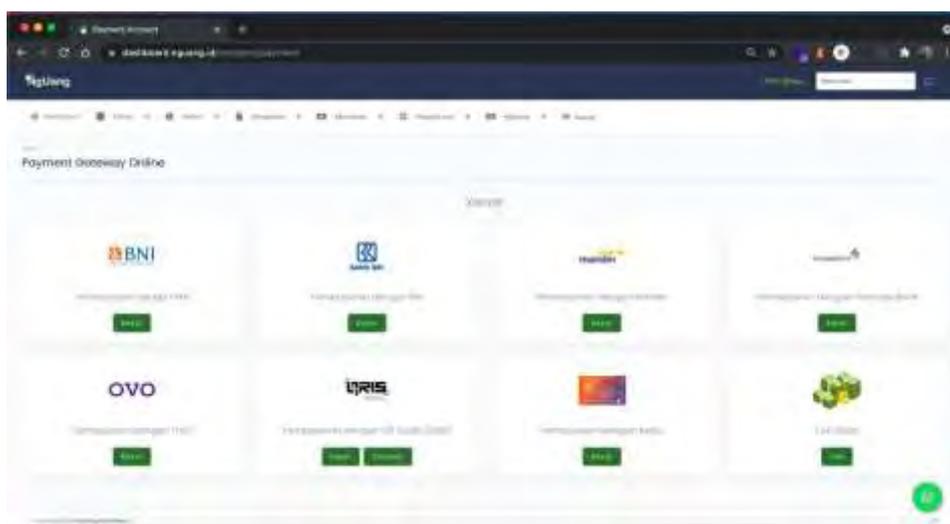
Gambar 4.9 *Website Panel*

Dengan *website panel* tim *internal* Nguang dapat mengubah semua informasi yang terdapat pada *website landing* maupun *website guide* kapanpun dengan mudah tanpa harus merubah kembali *program*, semua informasi sudah dibuat dinamis.



Gambar 4.10 Website Panel 2

Karena semuanya sudah dapat diubah secara dinamis maka tim *internal* pun sudah memangkas banyak waktu jika kedepanya terdapat perubahan informasi pada *website landing* ataupun menambah panduan panduan pada *website guide*.



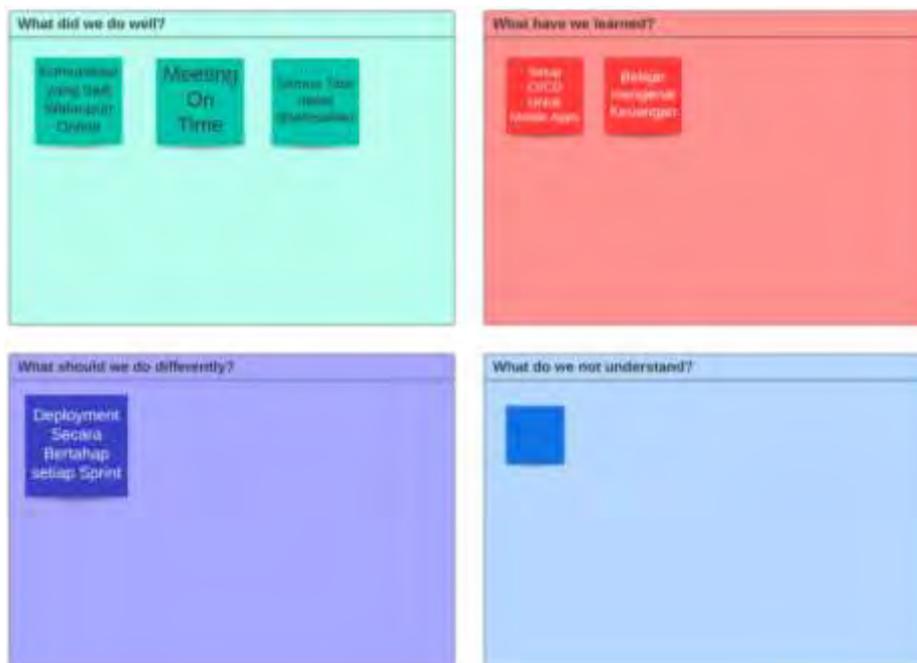
Gambar 4.11 Payment Gateway Online

Dengan adanya integrasi *payment gateway online* maka tim *internal* pun dapat menghemat waktu dengan tidak perlu lagi cek manual pembayaran *user* dan bahkan secara otomatis status *user* dapat berubah menjadi *user premium*, selain itu juga dari sisi *user* dapat meningkatkan kepercayaan dan kemudahan, pilihan yang diberikan juga banyak seperti *virtual account* dari berbagai bank atau juga bisa menggunakan kartu hingga pembayaran *e-wallet*.

3. Sprint 3

a. *Sprint Retrospective*

Pada gambar 4.12 dijelaskan mengenai hasil dari *sprint retrospective* pada *sprint 3*

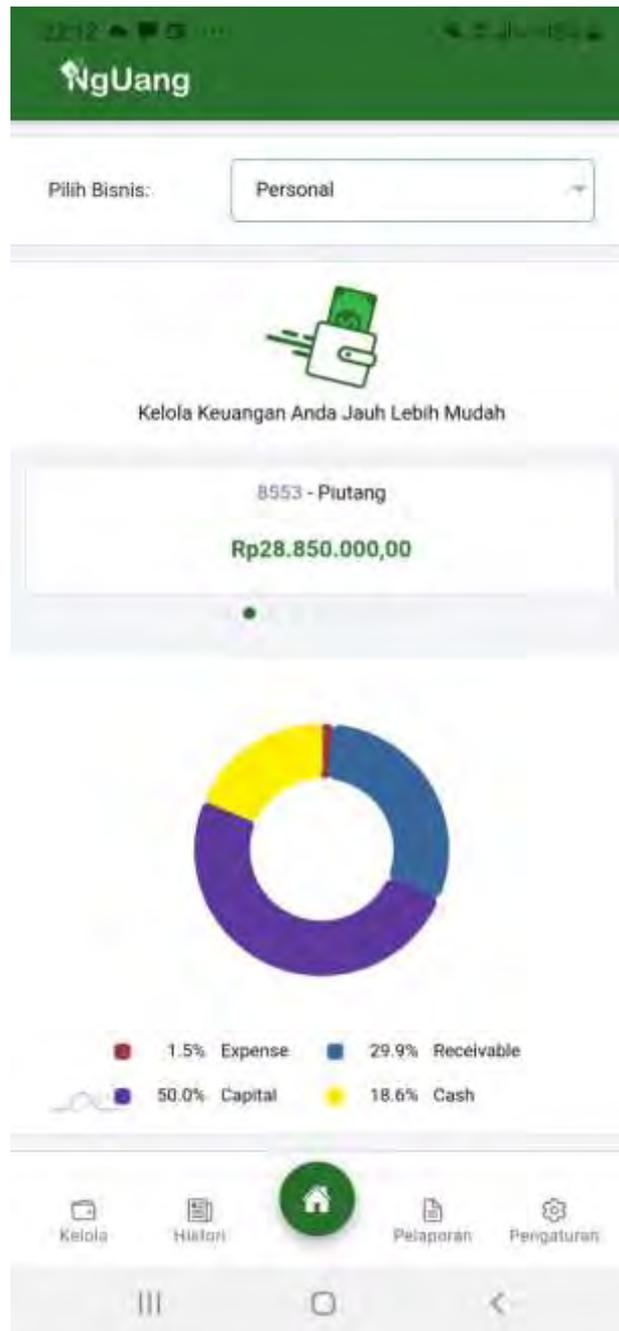


Gambar 4.12 *Sprint Retrospective Sprint 3*

Pada *sprint 3* ini yang sudah dilakukan dengan baik yaitu setiap *meeting* selalu *on time*, tidak ada yang telat dan komunikasi yang baik walaupun dilakukan secara online dan semua *task* dapat diselesaikan. Hal baru yang dipelajari mengenai *setup CI/CD* untuk *mobile apps* dan belajar mengenai keuangan akutansi. Lalu hal yang membuat berbeda yaitu *deployment* dapat dilakukan secara bertahap.

b. *Sprint Review*

Pada *sprint* 3 berfokus pada pembuatan aplikasi *mobile* untuk NgUang dan dapat diakses di <https://play.google.com/store/apps/details?id=com.nguang.apps> dengan tampilan UI seperti pada gambar 4.13 dan semua *task* dapat diselesaikan tepat waktu.



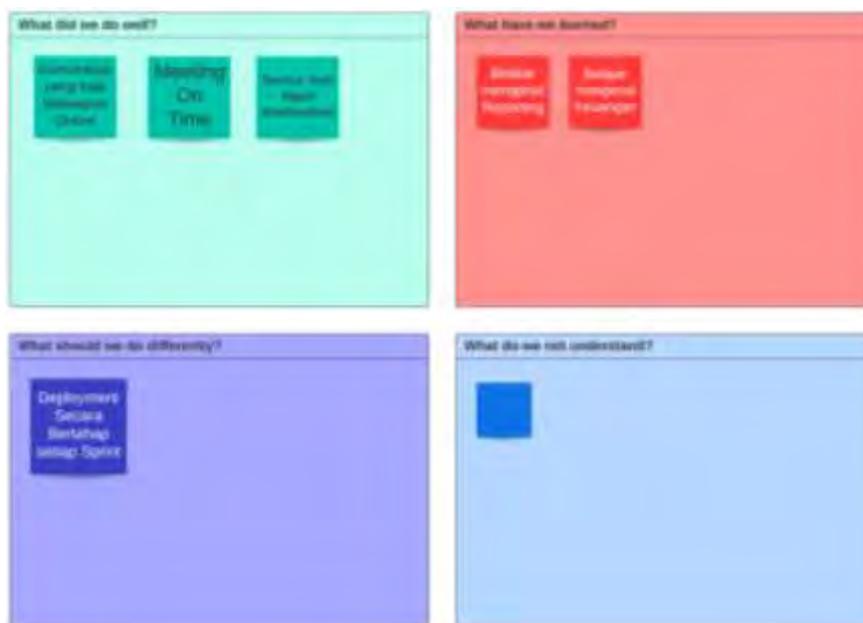
Gambar 4.13 Aplikasi *Mobile* NgUang

Dengan adanya *mobile apps* NgUang dapat meningkatkan kepuasan dari *user* dan meningkatkan jumlah pengguna NgUang itu sendiri, dimana meningkatkan *user* yang sering menggunakan hp khususnya *android*.

4. Sprint 4

a. *Sprint Retrospective*

Pada gambar 4.14 dijelaskan mengenai hasil dari *sprint retrospective* pada *sprint 4*



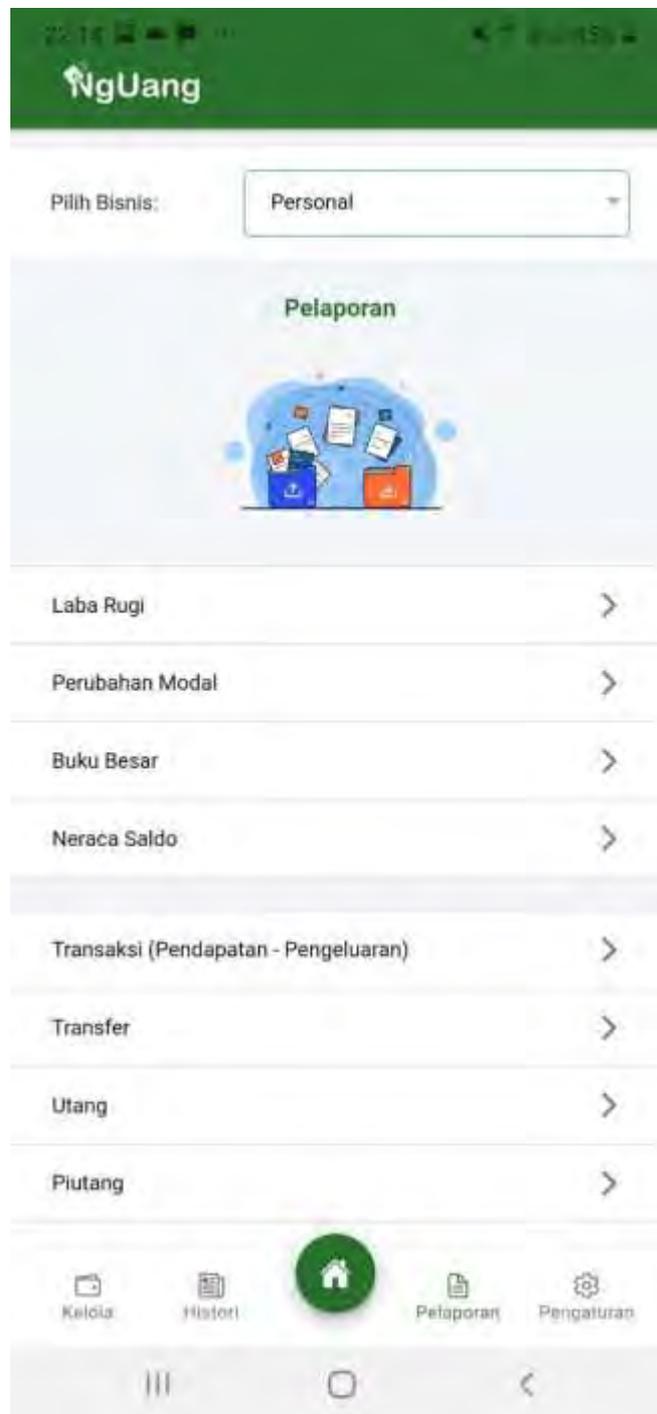
Gambar 4.14 *Sprint Retrospective Sprint 4*

Pada *sprint 4* ini yang sudah dilakukan dengan baik yaitu setiap *meeting* selalu *on time*, tidak ada yang telat dan komunikasi yang baik walaupun dilakukan secara online dan semua *task* dapat diselesaikan. Hal baru yang dipelajari mengenai laporan dan juga keuangan. Lalu hal yang membuat berbeda yaitu *deployment* dapat dilakukan secara bertahap.

b. *Sprint Review*

Pada *sprint 4* ini semua *task* yang berfokus pada pelaporan *accounting* dapat diselesaikan seperti pada gambar 4.15. Sedangkan pembuatan *website landing* dapat

diakses melalui url www.nguang.id atau seperti pada gambar 4.16. Pembuatan *website guide* dapat diakses melalui url www.guide.nguang.id atau seperti yang terlihat pada Gambar 4.17.



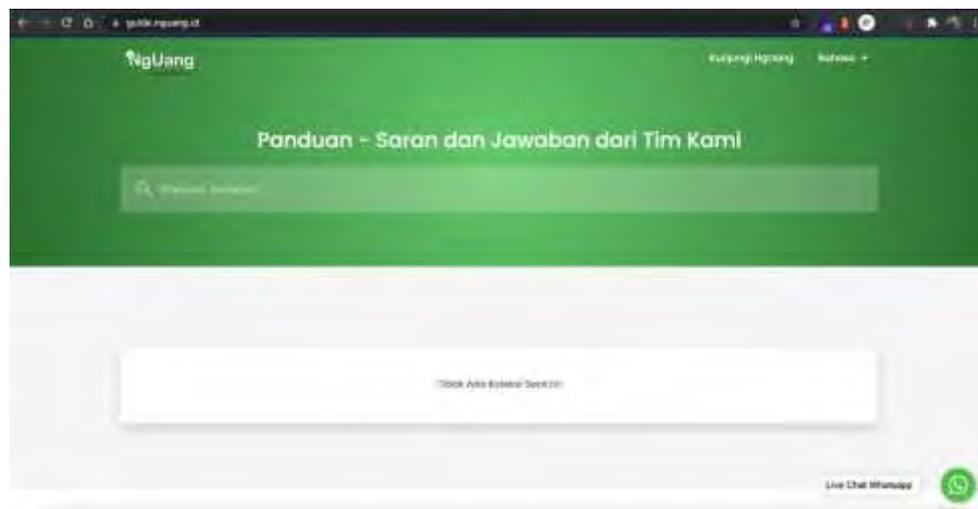
Gambar 4.15 *Reporting* Aplikasi *Mobile* NgUang

User dapat menunduh laporan keuangan dengan mudah dengan format PDF, tidak hanya laporan keuangan transaksi, tetapi hingga laporan laba rugi, perubahan modal, buku besar dan neraca saldo.



Gambar 4.16 Website Landing

Dengan adanya *website landing*, mendorong *user* untuk fokus pada satu produk hingga *user* tertarik untuk menggunakan produk NgUang.



Gambar 4.17 Website Guide

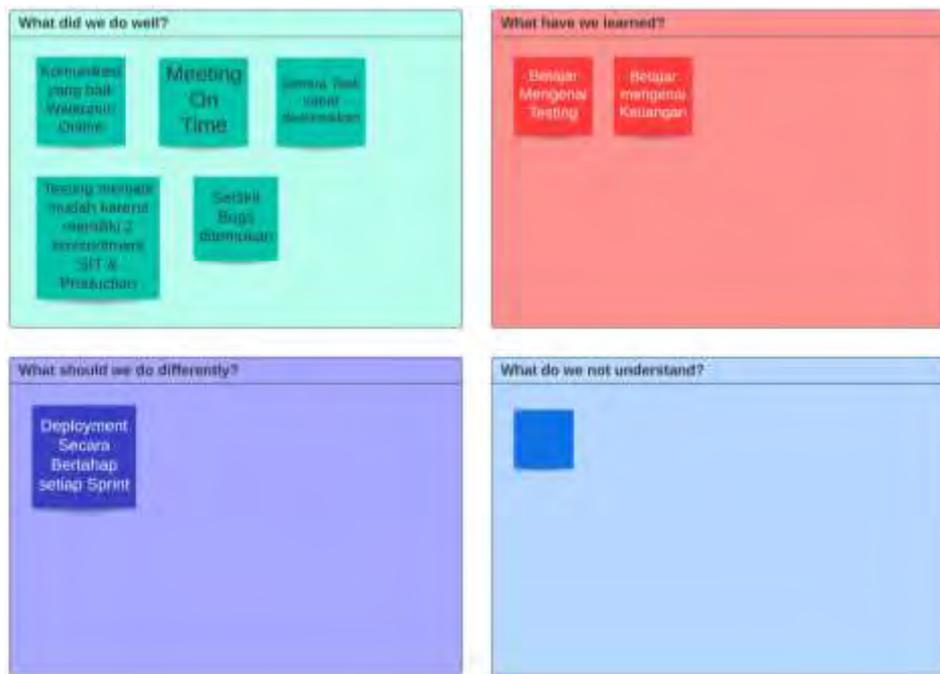
Seringkali *user* tidak tau cara penggunaan aplikasi, maka dari itu dengan adanya *website guide* bisa menjadi tempat tim *internal* NgUang untuk menyimpan panduan-panduan penggunaan

platform NgUang bisa berupa video, gambar dan penjelasan tambahan sesuai dengan kebutuhan.

5. Sprint 5

a. *Sprint Retrospective*

Pada gambar 4.18 dijelaskan mengenai hasil dari *sprint retrospective* pada *sprint 5*



Gambar 4.18 *Sprint Retrospective Sprint 5*

Pada *sprint 5* ini yang sudah dilakukan dengan baik yaitu setiap *meeting* selalu *on time*, tidak ada yang telat, *testing* menjadi lebih mudah karena memiliki *environment testing*, *bugs* sedikit, komunikasi yang baik walaupun dilakukan secara online dan semua *task* dapat diselesaikan. Hal baru yang dipelajari mengenai *testing* dan keuangan. Lalu hal yang membuat berbeda yaitu *deployment* dapat dilakukan secara bertahap.

b. *Sprint Review*

Pada *sprint 5* ini dilakukan pengujian *black box* yang dijelaskan pada Tabel 4.10.

Tabel 4.10 *Black Box*

No	Skenario Pengujian	Hasil Pengujian
1	<i>Customer</i> dapat mengakses <i>domain</i>	<i>Valid</i>
2	Tim <i>internal</i> dapat mengelola situs	<i>Valid</i>
3	Tim <i>internal</i> dapat mengelola kategori <i>storage</i>	<i>Valid</i>
4	Tim <i>internal</i> dapat mengelola <i>storage</i>	<i>Valid</i>
5	Tim <i>internal</i> dapat mengelola tim	<i>Valid</i>
6	Tim <i>internal</i> dapat mengelola <i>fact</i>	<i>Valid</i>
7	Tim <i>internal</i> dapat mengelola FAQ	<i>Valid</i>
8	Tim <i>internal</i> dapat mengelola kategori <i>event</i>	<i>Valid</i>
9	Tim <i>internal</i> dapat mengelola <i>event</i>	<i>Valid</i>
10	Tim <i>internal</i> dapat mengelola kategori berita	<i>Valid</i>
11	Tim <i>internal</i> dapat mengelola berita	<i>Valid</i>
12	Tim <i>internal</i> dapat mengelola testimonial	<i>Valid</i>
13	Tim <i>internal</i> dapat mengelola <i>client</i>	<i>Valid</i>
14	Tim <i>internal</i> dapat mengelola kategori <i>guide</i>	<i>Valid</i>
15	Tim <i>internal</i> dapat mengelola <i>guide</i>	<i>Valid</i>
16	Tim <i>internal</i> dapat mengelola bank	<i>Valid</i>
17	Tim <i>internal</i> dapat mengelola <i>price</i>	<i>Valid</i>
18	Tim <i>internal</i> dapat mengelola <i>feature</i>	<i>Valid</i>
19	Tim <i>internal</i> dapat mengelola kategori video	<i>Valid</i>
20	Tim <i>internal</i> dapat mengelola video	<i>Valid</i>
21	Tim <i>internal</i> dapat mengelola <i>profile</i>	<i>Valid</i>
22	Tim <i>internal</i> dapat mengelola kategori konten	<i>Valid</i>
23	Tim <i>internal</i> dapat mengelola konten	<i>Valid</i>
24	<i>Customer</i> dapat melakukan pembayaran via OVO, QRIS, kartu, Gopay dan VA	<i>Valid</i>

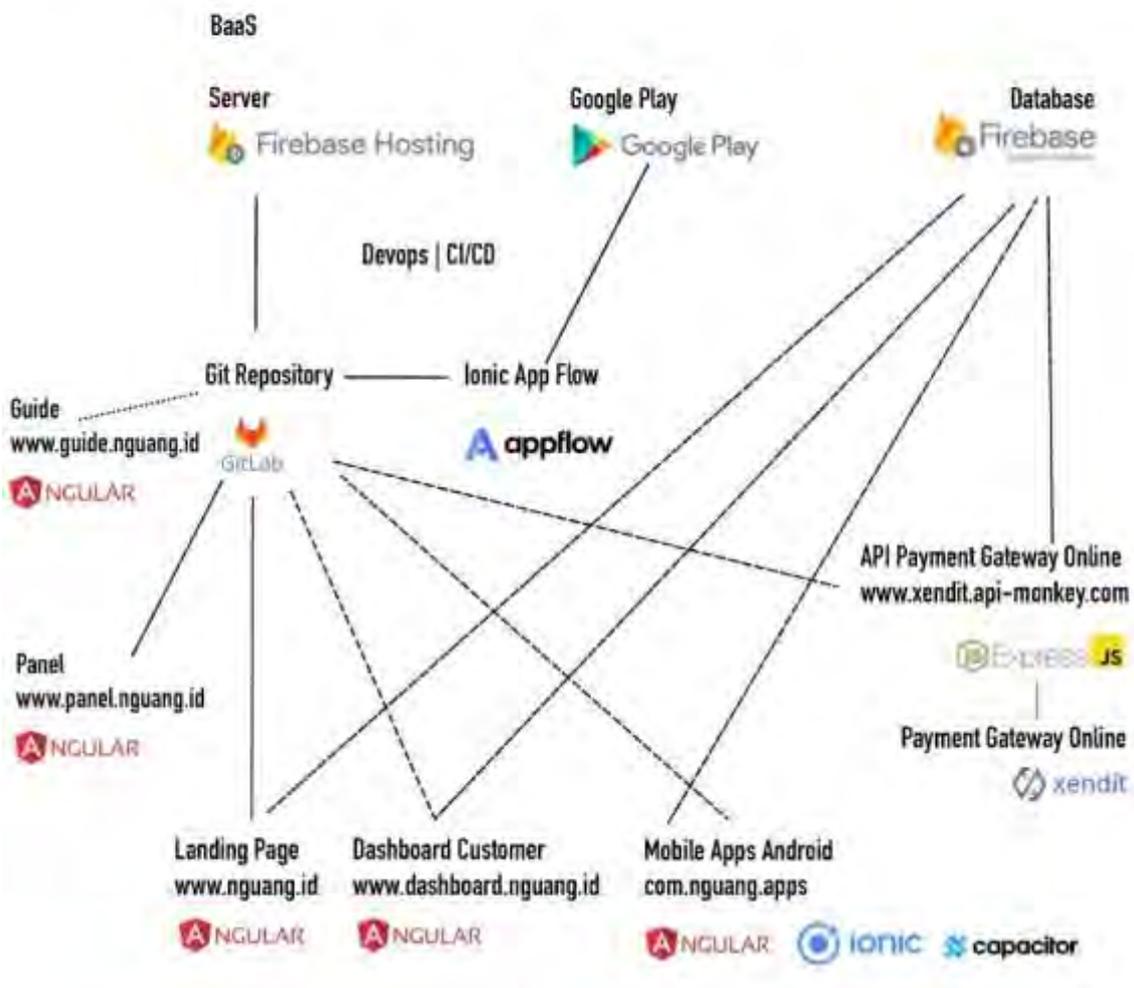
No	Skenario Pengujian	Hasil Pengujian
25	<i>Customer</i> dapat melakukan <i>login</i> , <i>register</i> dan <i>resetPassword</i>	<i>Valid</i>
26	<i>Customer</i> dapat mengelola transaksi masuk dan keluar	<i>Valid</i>
27	<i>Customer</i> dapat melakukan transfer antar akun	<i>Valid</i>
28	<i>Customer</i> dapat mengelola akun	<i>Valid</i>
29	<i>Customer</i> dapat mengelola modal, asset, hutang dan piutang	<i>Valid</i>
30	<i>Customer</i> dapat mengelola <i>profile</i>	<i>Valid</i>
31	<i>Customer</i> dapat mengelola bisnis	<i>Valid</i>
32	<i>Customer</i> dapat melihat laporan laba rugi, penyesuain modal, kas besar dan neraca saldo	<i>Valid</i>
33	<i>Customer</i> dapat melihat panduan	<i>Valid</i>
34	<i>Customer</i> dapat melihat informasi lengkap mengenai tim NgUang	<i>Valid</i>
35	<i>Customer</i> dapat melihat informasi lengkap mengenai <i>pricing</i> NgUang	<i>Valid</i>
36	<i>Customer</i> dapat melihat informasi lengkap mengenai NgUang	<i>Valid</i>
37	<i>Customer</i> dapat melihat informasi lengkap mengenai <i>event</i> NgUang	<i>Valid</i>
38	<i>Customer</i> dapat melihat informasi lengkap mengenai <i>inspiration</i> NgUang	<i>Valid</i>
39	<i>Customer</i> dapat melihat informasi lengkap mengenai FAQ NgUang	<i>Valid</i>
40	<i>Customer</i> dapat melihat informasi lengkap mengenai TOS NgUang	<i>Valid</i>
41	<i>Customer</i> dapat melihat informasi lengkap mengenai <i>Privacy Policy</i> NgUang	<i>Valid</i>

4.3 Perancangan Arsitektur Bisnis

Pada perancangan ini arsitektur bisnis peneliti mengaplikasikan *cloud computing* SaaS dan BaaS. Dimana SaaS sendiri ditujukan pada *user* NgUang tidak perlu mengeluarkan biaya yang mahal untuk pembelian domain, hosting, pembuatan aplikasi, dan biaya maintenance. Hal ini terjadi karena semuanya sudah ditangani oleh NgUang *sendiri*. Dalam hal ini, *user* hanya perlu

ke *ionic app flow* untuk *mobile apps*. Perihal kebutuhan *payment gateway online* memilih xendit dengan teknologi *express nodejs*. Xendit sendiri menjadi salah satu *payment gateway online* terbaik di Indonesia dan integrasi dengan *nodejs* sangatlah mudah. Sedangkan teknologi *frontend* menggunakan *angular*, dikarenakan untuk skala aplikasi besar sangat cocok dan mudah untuk *maintenance*. Kalau untuk teknologi *mobile* menggunakan *ionic* dengan bahasa *angular* agar memudahkan tim *develop* dengan satu bahasa *javascript*.

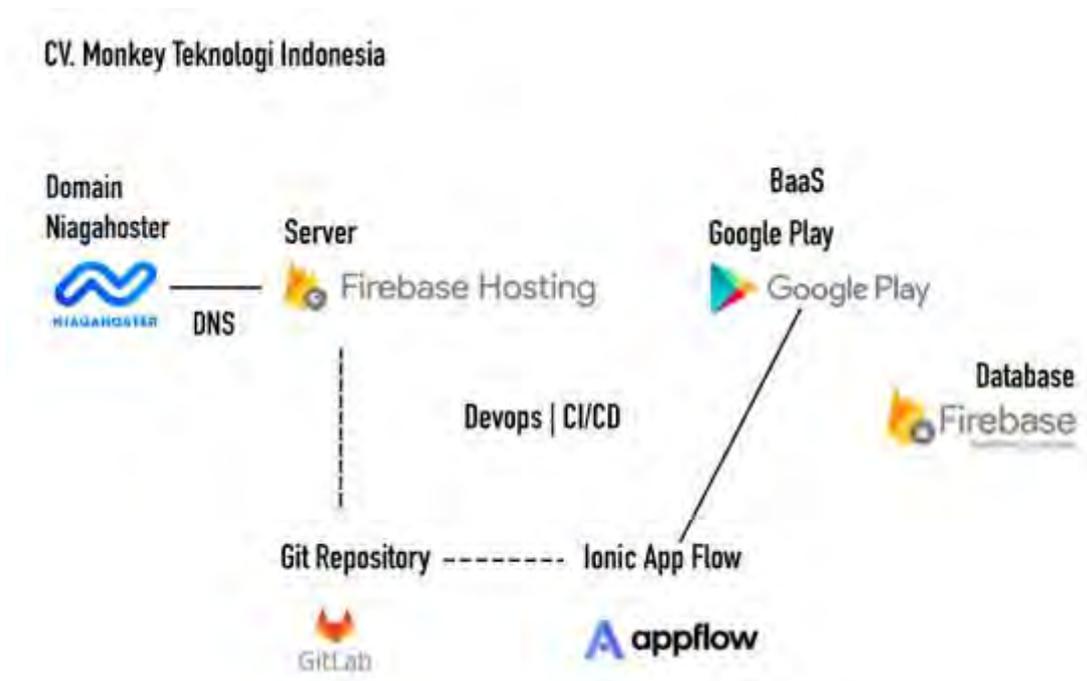
CV. Monkey Teknologi Indonesia



Gambar 4.20 Arsitektur Teknologi

4.5 Perancangan Arsitektur Server

Pada perancangan arsitektur server ini peneliti menggunakan *cloud computing* BaaS atau *backend as a service*. Dimana server berada di *cloud*, dikarenakan untuk *startup* baru model BaaS ini sangat cocok diaplikasikan, dikarenakan perusahaan tidak perlu membayar biaya yang sangat besar untuk server. Bahkan *firebase* sendiri menyediakan biaya versi *free* dengan ketentuan terbatas yang sudah sangat cukup untuk *starter startup*. Perancangan arsitektur server digambarkan pada gambar 4.21. *Firestore* sendiri memiliki integrasi yang cukup lengkap, salah satunya integrasi penerapan CI/CD dengan *git repository* yang akan dijelaskan pada Implementasi *devops*.



Gambar 4.21 Arsitektur Server

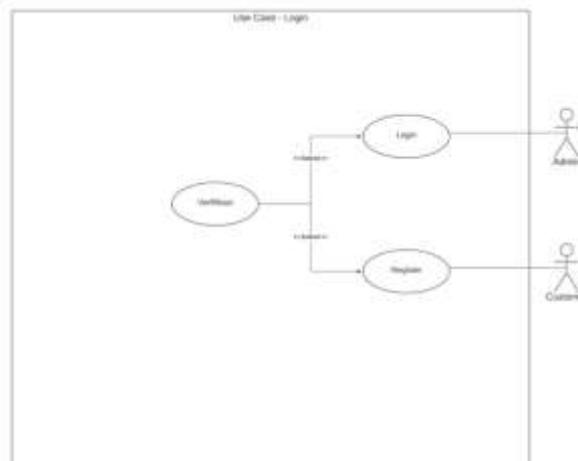
4.6 Perancangan Arsitektur Aplikasi

Arsitektur sistem berguna untuk memberikan gambaran bagaimana mendefinisikan komponen-komponen yang terlibat dalam sistem lebih detail secara struktur dengan tujuan agar struktur yang dirancang dapat menjawab kebutuhan sistem yang ada. Perancangan arsitektur sistem dimulai dari *use case diagram*, desain *database*, dan *sitemap* aplikasi.

4.6.1 Use Case Diagram

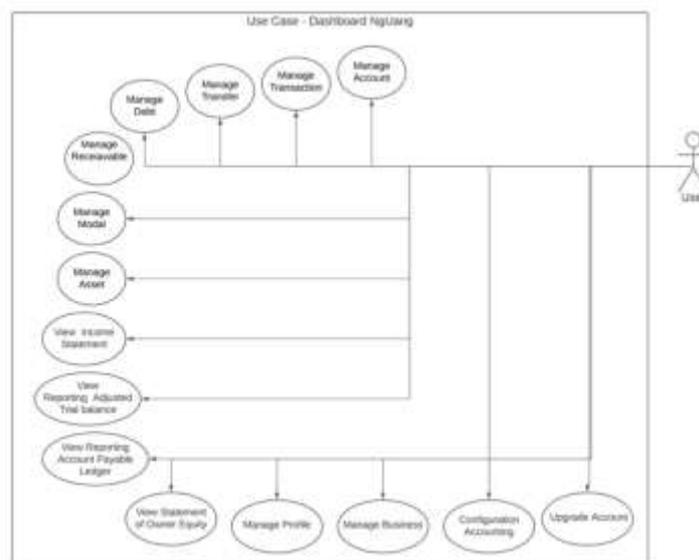
Use case diagram menggambarkan interaksi antara aktor dengan proses atau sistem yang dibuat. Dengan use case ini, dapat diketahui proses yang terjadi saat sistem berjalan, gambar use case bisa dilihat pada gambar dibawah ini:

1. Use Case Login dan Register



Gambar 4.22 Use Login dan Register

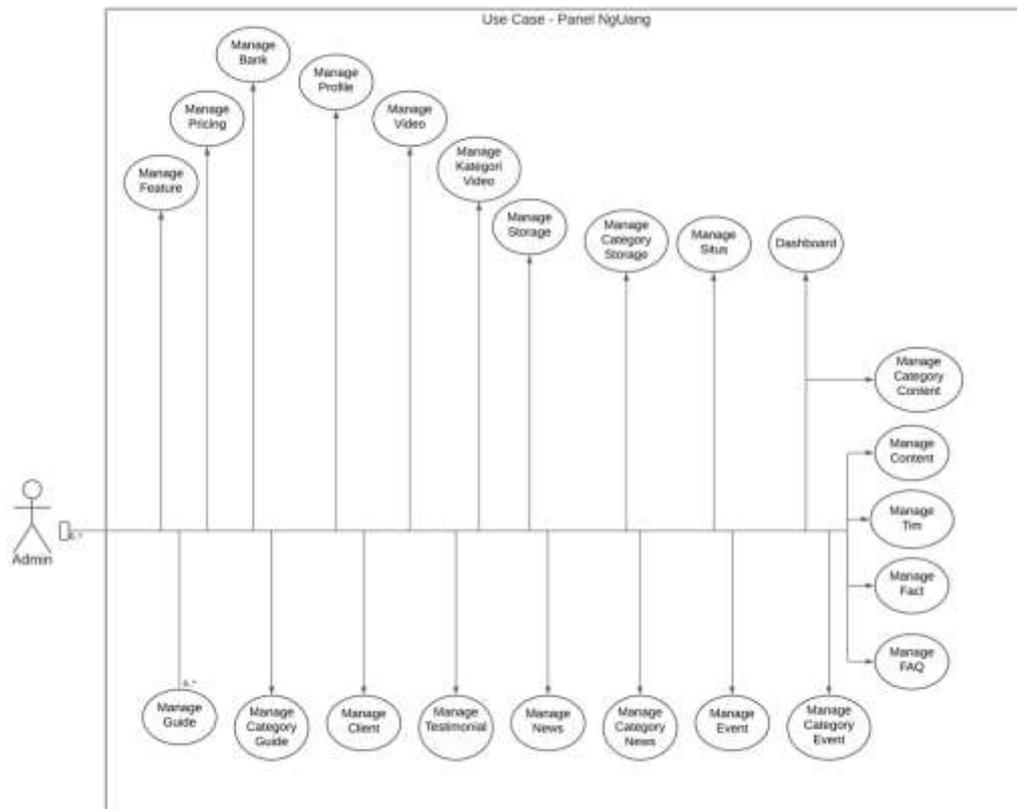
2. Use Case Diagram Mobile Dashboard NgUang



Gambar 4.23 Use Case Diagram Mobile Dashboard NgUang

Use case diagram pada gambar 4.23 sudah dijelaskan pada tabel 4.7 *user story sprint 24* mengenai perancangan *website panel*.

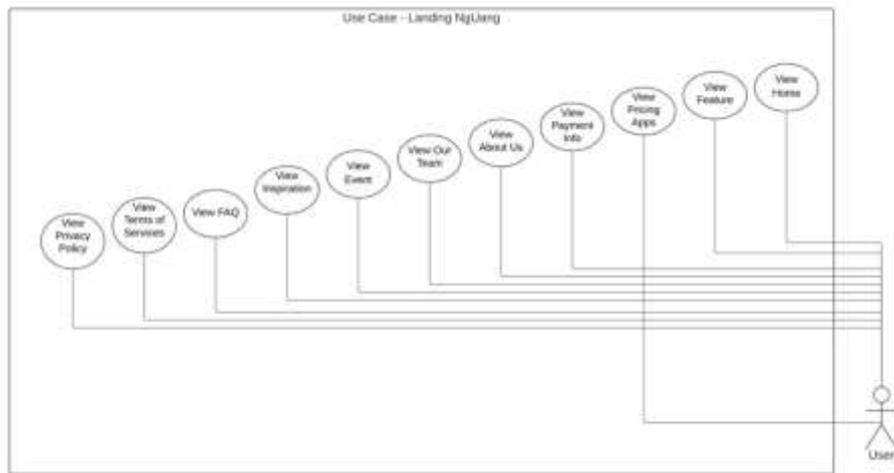
3. Use Case Diagram Panel NgUang



Gambar 4.24 Use Case Diagram Panel NgUang

Use case diagram pada gambar 4.24 sudah dijelaskan pada tabel 4.5 *user story sprint 2* mengenai perancangan *website panel*.

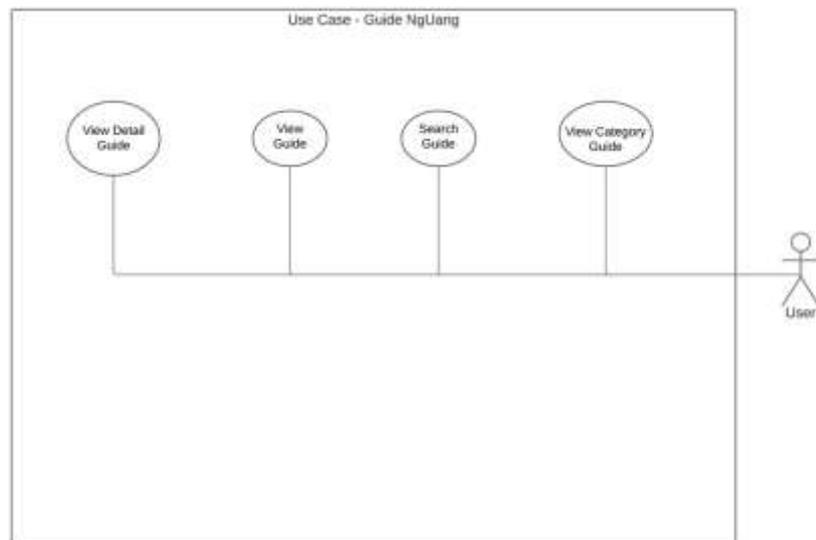
4. Use Case Diagram Landing NgUang



Gambar 4.25 Use Case Diagram Landing NgUang

Use case diagram pada gambar 4.25 sudah dijelaskan pada tabel 4.7 user story sprint 4 mengenai perancangan *website landing*.

5. Use Case Diagram Guide NgUang



Gambar 4.26 Use Case Diagram Guide NgUang

Use case diagram pada gambar 4.26 sudah dijelaskan pada tabel 4.8 user story sprint 5 mengenai perancangan *website guide*.

4.6.2 *Class Diagram*

Diagram kelas yang menunjukkan jenis objek dalam sistem semua jenis hubungan. *Diagram* kelas menunjukkan properti operasi kelas dan batasan yang terkait dengan objek. *Diagram* kelas adalah alat terbaik untuk mendesain perangkat. *Diagram* kelas memungkinkan pengembang untuk memahami struktur sistem untuk mencapai keunggulan desain sistem. Pada penelitian ini, *class diagram* yang dibuat berdasarkan *database* NoSQL yang dimana data berupa *collection* dan *document*. Pada penelitian ini dibagi menjadi 3 *collection* utama sebagai berikut:

1. *Class Diagram Apps*

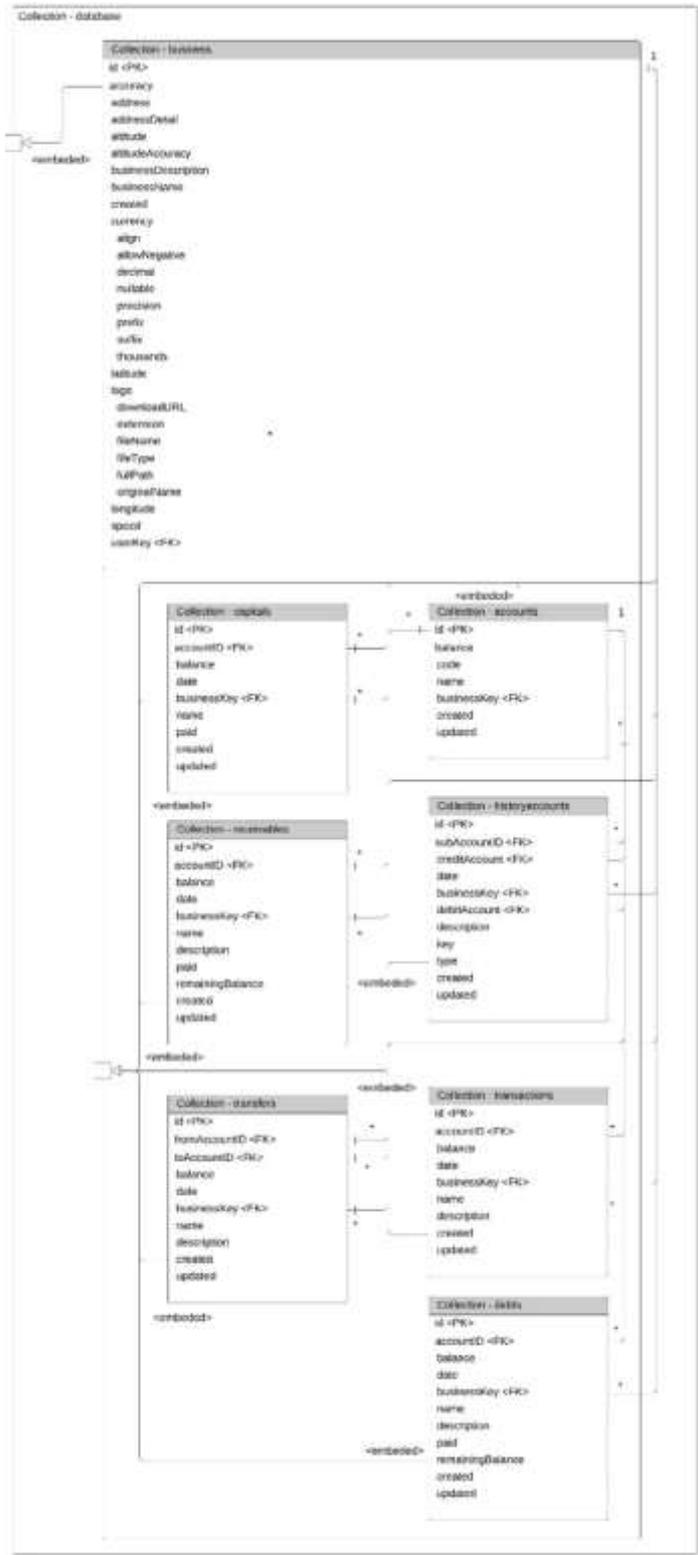
Class diagram ini digambarkan pada gambar 4.27 dan di jelaskan pada tabel 4.11.

Tabel 4.11 Class Diagram Apps

No	Collection	Sub Collection	Field
1	apps	-	id <PK>,key, projectID, projectName, created, updated
2	apps	clients	id <PK>, created, updated, category
3	apps	banks	id <PK>, created, updated, accountName, accountNumber, logo, name
4	apps	contentsCategory	id <PK>, created, updated, name
5	apps	contents	id <PK>, created, updated, description, image, key, path, title, category <FK>
6	apps	eventsCategory	id <PK>, created, updated, name
7	apps	events	id <PK>, created, updated, description, image, key, endDate, startDate, endTime, startTime, title, category <FK>
8	apps	storagesCategory	id <PK>, created, updated, name
9	apps	storages	id <PK>, created, updated, storage (downloadURL, extension, filename, filetype, fullPath, originalName, size), category <FK>
10	apps	pricings	id <PK>, created, updated, descPricing, description, pricing, title
11	apps	features	id <PK>, created, updated, key, description, image, title
12	apps	videosCategory	id <PK>, created, updated, name
13	apps	videos	id <PK>, created, updated, link, title, key, category <FK>

2. Class Diagram Database

Class diagram ini digambarkan pada gambar 4.28 dan di jelaskan pada tabel 4.10.



Gambar 4.28 Class Diagram Database

Tabel 4.12 Class Diagram Database

No	Collection	Sub Collection	Field
1	database	-	-
2	database	buisness	<i>id <PK>, accuracy, address, addressDetail, altitude, altitudeAccuracy, businessDescription, businessName, created, currency (align, allowNegative, decimal, nullable, precision, prefix, suffix, thousands), latitude, logo (downloadURL, extension, filename, filetype, fullPath, originalName), longitude, speed, userKey <FK></i>
3	database	capitals	<i>id <PK>, created, updated, accountID<FK>, balance, date, businessKey<FK>, name, paid</i>
4	database	accounts	<i>id <PK>, created, updated, balance, code, name, businessKey<FK></i>
5	database	receivables	<i>id <PK>, accountID<FK>, balance, date, businessKey<FK>, name, description, paid, remainingBalance</i>
6	database	historyAccounts	<i>id <PK>, created, updated, subAccountID<FK>, creditAccount<FK> date, businesskey<FK>, debitAccount<FK>, description, key, type</i>
7	database	transfer	<i>id <PK>, created, updated, fromAccountID<FK>, toAccountID<FK>, balance, date, businessKey <FK>, name, description</i>
8	database	transactions	<i>id <PK>, created, updated, accouuntID<FK>, balance, date, businessKey<FK>, name, description</i>
19	database	debts	<i>id <PK>, created, updated, balance, accountID<FK>, date, name, description, paid, remainingBalance</i>

3. Class Diagram Globals

Pada *class diagram* ini digambarkan pada gambar 4.29 dan di jelaskan pada tabel 4.13.

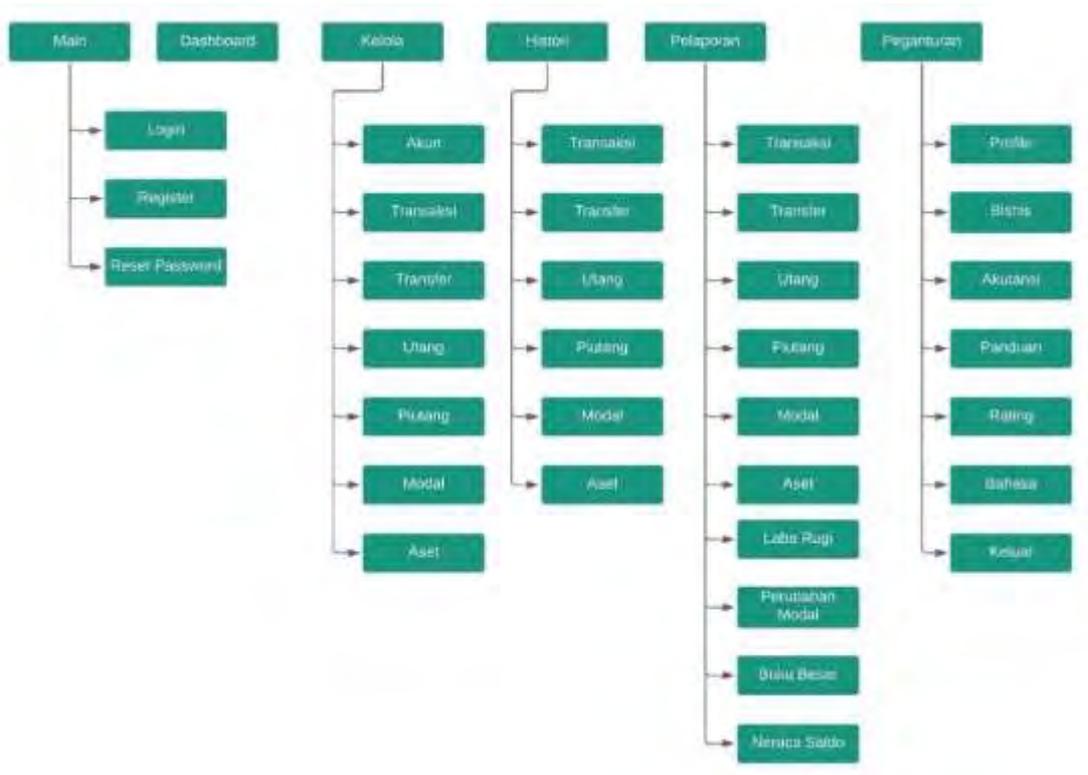
Tabel 4.13 Class Diagram Globals

No	Collection	Sub Collection	Field
1	<i>globals</i>	-	-
2	<i>globals</i>	<i>builder</i>	<i>id <PK>, created, updated, name, description</i>
3	<i>globals</i>	<i>master</i>	<i>id <PK>, created, updated, name, description</i>
4	<i>globals</i>	<i>menusCategory</i>	<i>id <PK>, created, updated, name, module, domain, order, icon, category</i>
5	<i>globals</i>	<i>menus</i>	<i>id <PK>, created, updated, module, active, domain, icon, order, name, category <FK></i>
6	<i>apps</i>	<i>storagesCategory</i>	<i>id <PK>, created, updated, name</i>
7	<i>apps</i>	<i>storages</i>	<i>id <PK>, created, updated, storage (downloadURL, extension, filename, filetype, fullPath, originalName, size), category <FK></i>
8	<i>apps</i>	<i>sites</i>	<i>id <PK>, created, updated, logo (downloadURL, extension, filename, filetype, fullPath, originalName, size), favicon (downloadURL, extension, filename, filetype, fullPath, originalName, size), accuracy, address, addressDetail, alias, altitude, altitueAccuracy, comingSoon, comingSoonDate, description, email, emailCS, emailCareer, emailInfo, Instagram, keyword, latitude, longitude, maintenance, maintenanceDate, name, notFoundButtonTitle, notFoundContent, notFoundTitle, project, publish, speed, version, vision, whatsapp, mission, youtube</i>
9	<i>globals - xendit</i>	<i>external</i>	<i>id <PK>, created, updated, amount, description, externalID, key, name, project, status, typePayment, typeXendit, userKey<FK></i>
10	<i>globals - users</i>	<i>users</i>	<i>id <PK>, created, updated, accountTypeNgUang, accuracy, activeBalanceNgUang, address, addressDetail, altitude, altitudeAccuracy, country, currency (align, alowNegativ, decimal, nullable, precision, prefix, suffix, thousands), displayName, email, emailVerified, key, lastLogin, lastLogout, latitude, longitude, password, phoneNumber, roleID, speed</i>

4.6.3 Sitemap Aplikasi

Sitemap merupakan salah satu cara sederhana untuk menginformasikan kepada pengguna tentang halaman-halaman pada suatu situs. Pada gambar 4.30 – 4.33 menampilkan bagaimana bentuk *sitemap* sistem yang akan dibuat.

1. Sitemap Dashboard Mobile



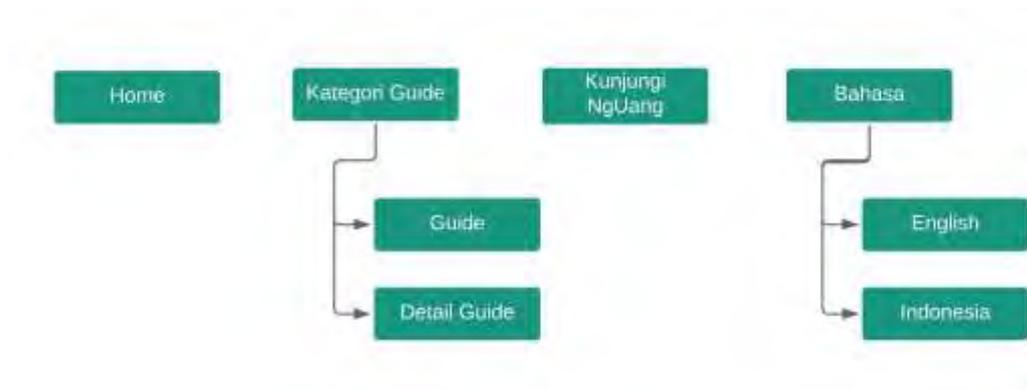
Gambar 4.30 Sitemap Dashboard Mobile

Dashboard mobile NgUang memiliki 6 menu utama yaitu:

- Menu *main*, user dapat *login*, *register* dan *reset password*.
- Menu *dashboard*, user dapat melihat ringkasan semua fitur di *dashboard*.
- Menu *kelola*, user dapat mengelola akun, transaksi, *transfer*, utang, piutang, modal dan *asset*.
- Menu *histori*, user dapat melihat histori transaksi, *transfer*, utang, piutang, modal dan *asset*.

- e. Menu laporan, *user* dapat melihat laporan transaksi, transfer, utang, piutang, modal dan *asset*, kas besar, laba rugi, perubahan modal dan neraca saldo.
- f. Menu pengaturan, *user* dapat mengelola *profile*, bisnis, melihat panduan, ganti bahasa dan keluar aplikasi.

2. Sitemap Website Guide

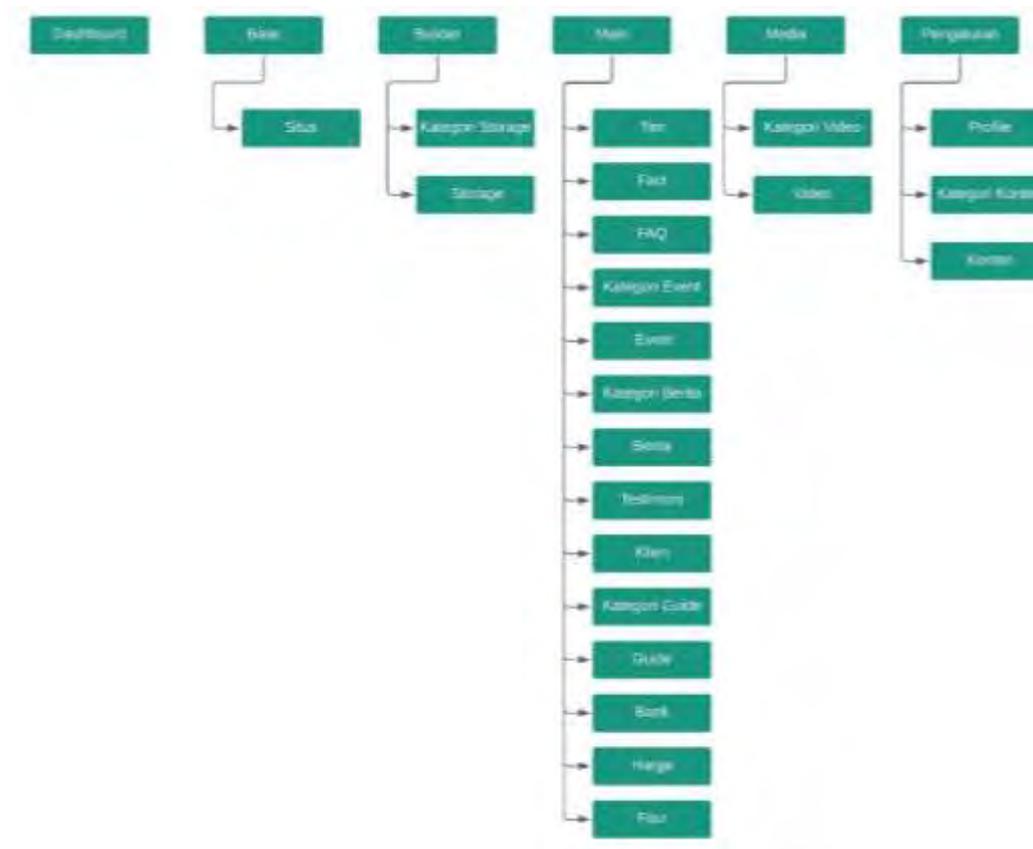


Gambar 4.31 Sitemap Website Guide

Website guide NgUang memiliki 4 menu utama yaitu:

- a. Menu *home*, *user* dapat mencari panduan yang dibutuhkan.
- b. Menu kategori *guide*, *user* dapat melihat panduan berdasarkan kategori, daftar panduan, dan detail dari panduan.
- c. Menu kunjungi NgUang untuk *user* dapat pergi ke *website landing*.
- d. Menu bahasa, *user* dapat mengganti Bahasa sesuai dengan keinginan.

3. Sitemap Website Panel

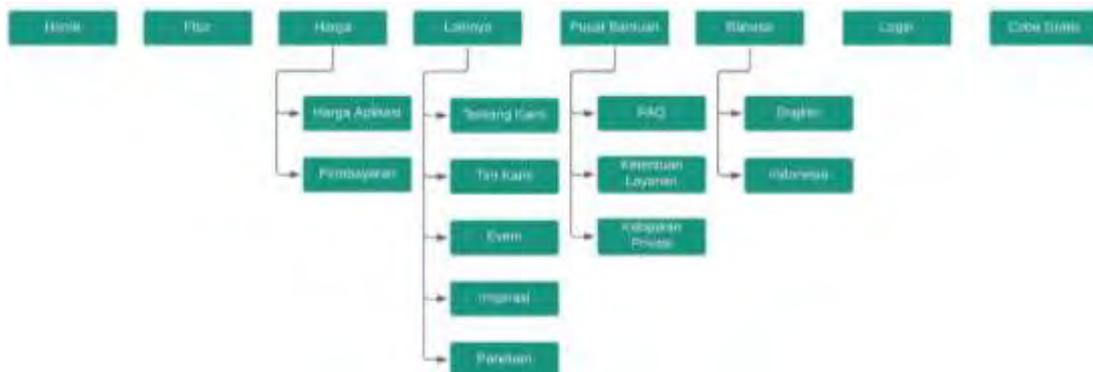


Gambar 4.32 Sitemap Website Panel

Website panel NgUang memiliki 6 menu utama yaitu:

- Menu *dashboard*, *admin* dapat melihat rangkuman *website*.
- Menu *base*, *admin* dapat mengelola informasi situs.
- Menu *builder*, *admin* dapat mengelola kategori *storage*, dan *storage*.
- Menu *main*, *admin* dapat mengelola *tim*, *fact*, FAQ, kategori *event*, *event*, kategori *berita*, *berita*, *testimoni*, *klien*, kategori *guide*, *guide*, *bank*, *harga* dan *fitur*.
- Menu *media*, *admin* dapat mengelola kategori *video* dan *video*.
- Menu *pengaturan*, *admin* dapat merubah informasi *profile*, kategori *konten* dan *konten*.

4. Sitemap Website Landing



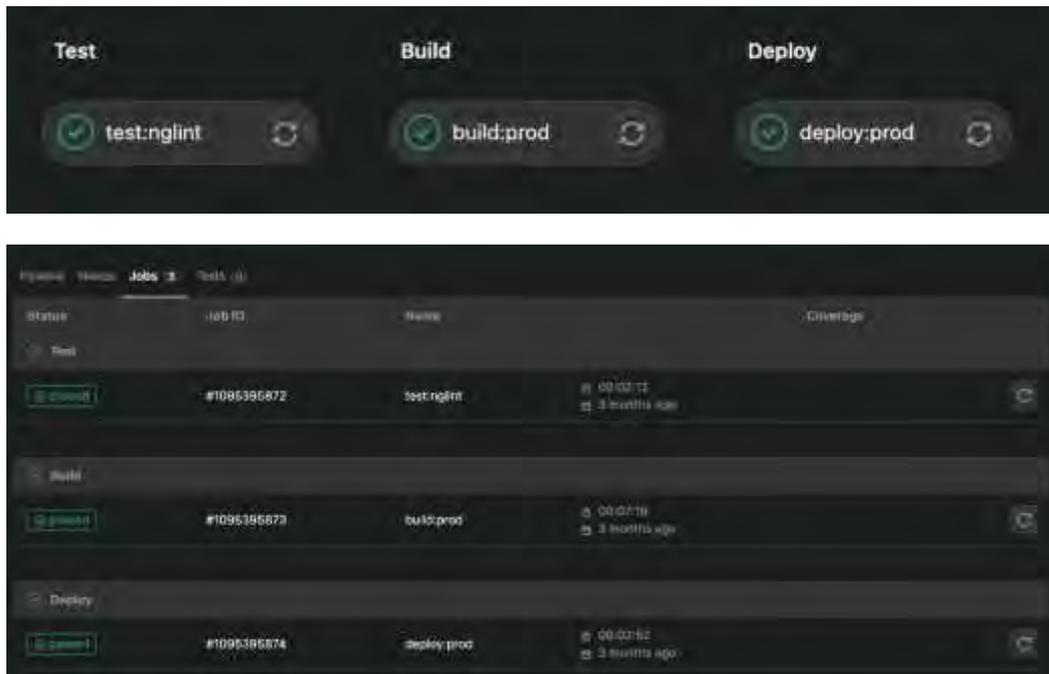
Gambar 4.33 Sitemap Website Landing

Website landing NgUang memiliki 8 menu utama yaitu:

- Menu *home*, user dapat melihat keseluruhan informasi NgUang secara singkat dan jelas.
- Menu *fitur*, user dapat melihat fitur NgUang.
- Menu *harga*, user dapat melihat harga aplikasi dan cara pembayaran NgUang.
- Menu *lainnya*, user dapat melihat informasi mengenai NgUang, tim, kegiatan, inspirasi dan panduan yang akan menuju *website guide*.
- Menu *pusat bantuan*, user dapat melihat informasi FAQ, ketentuan layanan dan kebijakan privasi.
- Menu *bahasa*, user dapat mengganti bahasa inggris atau bahasa.
- Menu *login*, user dapat masuk dan mengarah ke *website dashboard*.
- Menu *coba gratis*, user dapat *register* secara gratis dan mengarah ke *website dashboard*.

4.7 Implementasi Devops

Implementasi *devops*, peneliti membagi menjadi tiga bagian *stages*, yaitu *stage test*, *build*, dan *deploy*. *Stage test* dilakukan untuk melakukan validasi terlebih dahulu apakah *program* yang akan dibangun memiliki *bug* atau masalah, kemudian dilanjutkan dengan *stage build* seperti pada gambar 4.32 dimana *program* akan di *compile* yang nantinya akan dilanjutkan ke *stage build* dimana hasil *nuild* akan *diupload* ke *server* secara otomatis.



Gambar 4.34 Stage CI/CD

4.7.1 Continuous Integration

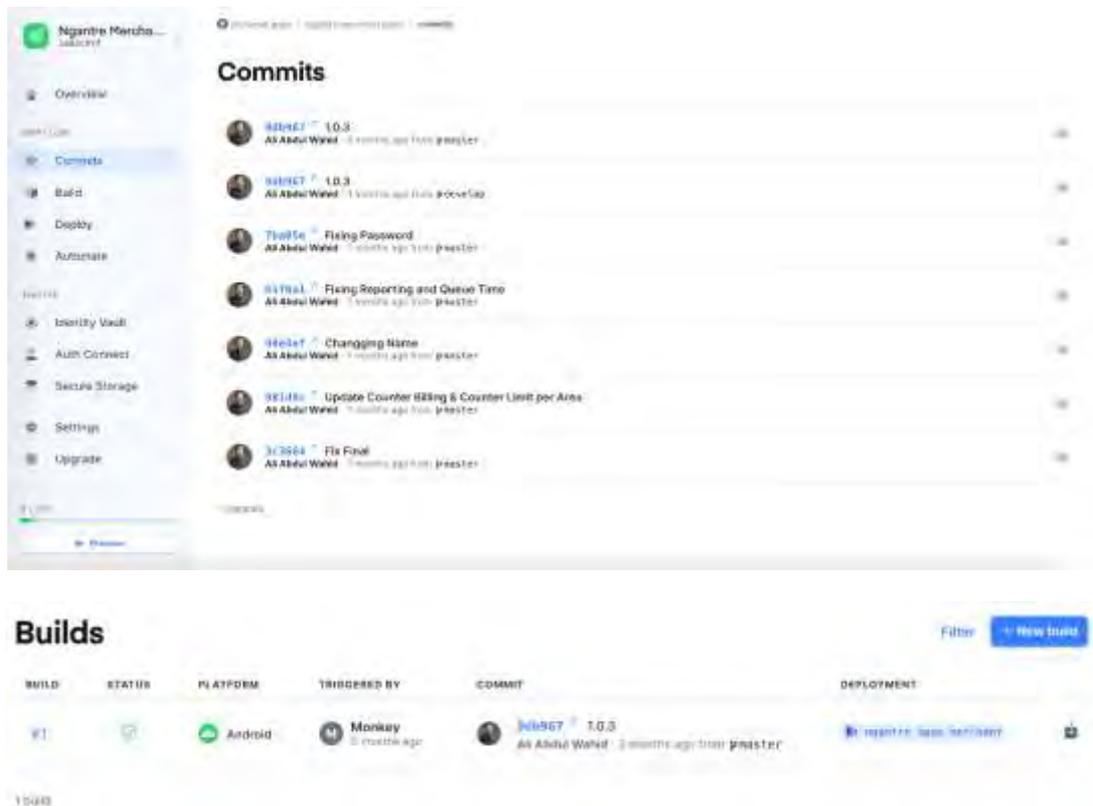
Pada tahap ini dilakukan integrasi *build*. Setiap kali ada komit dan *push* koding yang telah ditulis pengembang ke server CI gitlab, kemudian gitlab akan melakukan pengujian unit *functional*. Step di CI sendiri yaitu dimulai dari *plan* - *code* - *build* - lalu *test*. Berikut adalah konfigurasi gitlab-ci.yml untuk penerapan CI.

Setiap *commit* dan *push* yang dilakukan pada branch *master*, akan secara otomatis menjalankan *stage test* dan *build* seperti gambar 4.35 dibawah ini.



Gambar 4.35 Commit and Push

Aplikasi NgUang memiliki 2 *branch* utama yaitu *develop* yang terhubung dengan *environment testing* dan *master* yang terhubung dengan *environment production*. Begitu juga dengan *mobile apps* dengan integrasi *gitlab* dan juga *ionic app flow*, akan secara otomatis menjalankan *build* dan *test* seperti gambar 4.36 dibawah ini.



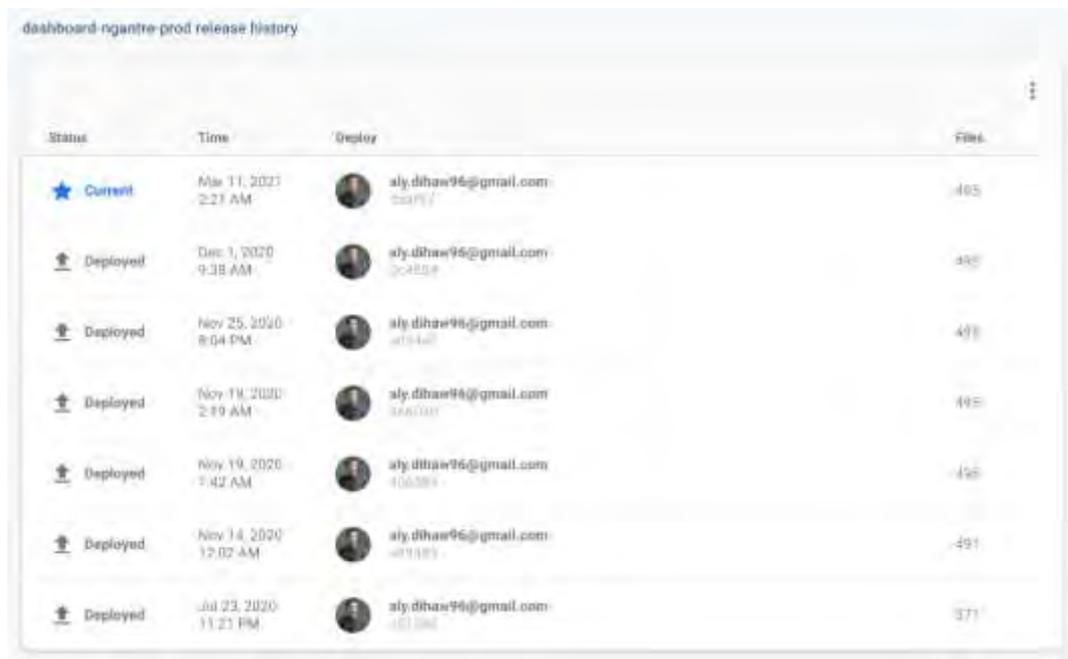
Gambar 4.36 Build Mobile Apps

4.7.2 Continuous Deployment

Pada tahap ini dilakukan integrasi *build* dimana hasil *build* yang ada akan diteruskan ke mode produksi yang akan diarahkan ke *server firebase* dan juga *ionic app flow* secara otomatis.

Tahapan di CD sendiri yaitu dimulai dari *release - deploy* – lalu *operate*. Berikut adalah konfigurasi *gitlab-ci.yml* untuk penerapan CD.

Hasil *build* akan masuk ke *step release* dimana hasil *build* akan diarahkan kepada *server firebase hosting* dan *ionic app flow* dan dilakukan *step deployment* dan *operate*, yang artinya secara otomatis menghasilkan *file* baru pada *hosting firebase* untuk *website* seperti pada gambar 4.37 maupun *google play console* untuk *mobile apps* seperti pada 4.38.



Status	Time	Deploy	Files
★ Current	Mar 11, 2021 2:21 AM	aly.dhaw96@gmail.com 204957	465
📦 Deployed	Dec 1, 2020 9:38 AM	aly.dhaw96@gmail.com 204956	465
📦 Deployed	Nov 25, 2020 8:04 PM	aly.dhaw96@gmail.com 415460	498
📦 Deployed	Nov 18, 2020 2:19 AM	aly.dhaw96@gmail.com 204955	498
📦 Deployed	Nov 19, 2020 7:42 AM	aly.dhaw96@gmail.com 106389	498
📦 Deployed	Nov 14, 2020 12:02 AM	aly.dhaw96@gmail.com 415461	491
📦 Deployed	Jul 23, 2020 11:21 PM	aly.dhaw96@gmail.com 415462	371

Gambar 4.37 *Firebase Hosting History Deployment*



TRIGGER BY	TYPE	STATUS	TARGET	BUILD	COMMIT
Monkey 3 months ago	Binary	🟢	▶️ main@ios@releases	🟢 #1	bd89e7 1.0.3 Ali Abdul Wazid · 3 months ago from prester

Gambar 4.38 *Ionic App Flow History Deployment*

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan uraian yang telah disampaikan pada bahasan sebelumnya, berikut adalah hasil kesimpulan yang didapat:

1. Perancangan arsitektur bisnis, teknologi, aplikasi, dan *server* memudahkan *scrum* tim pada saat proses *development*, karena semua analisis dilakukan secara matang sebelum memulai membuat aplikasi dan mengurangi resiko kegagalan yang tinggi.
2. Penerapan *cloud computing BaaS* sangat cocok digunakan untuk *startup* baru yang belum memiliki *profit* dan biaya untuk membayar *server* namun ingin produknya *release* dengan cepat. Sedangkan penerapan *cloud computing SaaS* sangat cocok untuk mendatangkan banyak pelanggan, karena pelanggan tidak perlu mengeluarkan biaya yang sangat tinggi untuk membuat aplikasi dan juga biaya *maintenance*.
3. Menggunakan penerapan metode *agile* dengan *framework scrum*, membuat apa yang akan dikerjakan menjadi lebih mudah, terukur waktu kegiatannya dengan 5 *sprint* atau 5 minggu, memudahkan tim pengembang dalam pengerjaan setiap *task* yang ada sehingga dalam proses pembuatan aplikasi jauh lebih terukur, lebih cepat, dan dengan hasil yang maksimal.
4. Pengaplikasian *sprint* yang meliputi *sprint backlog*, *planning*, *review*, dan *retrospective* sangat memudahkan kerjasama antar tim, sehingga setiap harinya para pengembang tidak bingung lagi apa yang harus dikerjakan pada saat *sprint* berjalan dan kerjasama antar tim menjadi lebih terukur bahkan dapat memperbaiki apa yang kurang dari setiap *sprint* yang ada dengan penerapan *sprint retrospective* dengan hasil nyata yang dapat dilihat langsung pada *sprint review*.
5. Penerapan *devops* sangat membantu *scrum* tim karena *agile scrum* sendiri bisa sangat dinamis, dimana proses *deployment* bisa dilakukan per *sprint* ataupun kapan saja. Diberlakukannya penerapan *devops* di awal, *scrum* tim tidak perlu memikirkan *deployment*

lagi, ini akan sangat memangkas waktu *development* yang sangat besar sehingga waktu akan lebih efektif dan *deployment* dapat dilakukan secara berkala.

5.2 Saran

Berdasarkan kesimpulan yang sudah diuraikan, adanya beberapa saran yang dapat membantu pengembangan menjadi lebih baik kembali:

1. Perlunya melakukan penerapan arsitektur aplikasi pada beberapa produk, khususnya pada saat analisis *database*, untuk tau bagaimana caranya agar semua produk bisa saling terintegrasi dalam satu *database*.
2. Perlunya memasukan unsur resiko dalam perencanaan, karena perlu adanya *plan B* dimana saat implementasi tidak sesuai dengan apa yang sudah direncanakan maka dari itu harus memasukan semua resiko apa saja yang mungkin terjadi.

DAFTAR PUSTAKA

- Abidah I K, et.al., 2020. Implementasi sistem basis data *cloud computing* pada sektor pendidikan. *Jurnal sains dan teknologi*.
- Agustina, S., 2013. Perpustakaan prasekolahku, seru! Bandung: cv. Restu bumi kencana.
- A. Dyck, et.al., 2015. *Towards definitions for release engineering and devops. Proceedings of the ieee/acm 3rd international workshop on release engineering*.
- Alfatih H da Marco R, 2015., 'Analisis pengembangan dan perancangan sistem informasi akademik smart berbasis cloud computing pada Sekolah Menengah Umum Negeri (SMUN) di Daerah Istimewa Yogyakarta', *Jurnal Telematika*, Vol. 8, No. 2, hlm. 63-91.
- Aryani L., dan Assegaff, S., 2017. Perencanaan arsitektur sistem informasi pada kantor kesehatan pelabuhan jambi menggunakan togap adm. *Jurnal manajemen sistem informasi*.
- Asmara, R dan Alhamidi., 2017. Pengolahan data rehabilitasi penyalahgunaan narkoba pada klinik aqilah payakumbuh. *Jurnal sistem informasi dan manajemen informatika*.
- A.S., Rosa dan Shalahuddin, M. 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Azdy, et.al., 2012. Implementasi *scrum* pada pengembangan *software* terdistribusi. Seminar nasional informatika 201.
- Budiyanto, A., 2012. Pengantar *cloud computing*. Komunitas *cloud computing* indonesia.
- Cloudflare., 2021. *What is baas? | backend-as-a-service vs. Serverless*. Retrieved february 15, 2020, from <https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/>
- Dicoding., 2019. *Devops di google cloud (part 1): continuous integration dan continuous deployment*. Retrieved february 15, 2020, from <https://www.dicoding.com/blog/devops-di-google-cloud-cicd/>
- Dobrigkeit F, et.al., 2020. *Innodev workshop: a one day introduction to combining design thinking, lean startup and agile software development*. *IEEE Intl. Conference on software engineering education & training*.
- Erich F, et.al., 2014. *Devops literature review*. University of twente.
- Erich F, et.al., 2017. A qualitative study of devops usage in practice. *Journal of software: evolution and process*,
- Gitlab., 2021. *All gitlab features*. Retrieved february 16, 2020, from <https://about.gitlab.com/features/>
- Irfan M. et.al., 2018. *Agile development methods* dalam pengembangan sistem informasi pengajuan kredit berbasis web (studi kasus: bank bri unit kolonel sugiono). *Jurnal teknologi dan open source*.

- Jha, P dan Khan, R., 2015. A review paper on devops: beginning and more to know. *International journal of computer applications*.
- Liputan6.com. 2019. Pengertian startup adalah, sejarah singkat dan karakteristiknya. Retrieved february 15, 2020, from <https://www.liputan6.com/teknoread/3867489/pengertian-startup-adalah-sejarah-singkat-dan-karakteristiknya>
- Sandy L. A, et.al., 2017. Rancang bangun aplikasi *chat* pada *platform* android dengan media *input* berupa *canvas* dan *shareable canvas* untuk bekerja dalam satu *canvas* secara *online*. *Jurnal teknik its*.
- Lutfiani N, et.al., 2020. Inovasi manajemen proyek *i-learning* menggunakan metode agile scrumban. *Jurnal nasional informatika dan teknologi*.
- Marks, e. A dan Lozano B. 2010. *Executive's guide to cloud computing*. Canada: john wiley & soon.
- Minarni dan Susanti. 2014. Sistem informasi *inventory* obat pada rumah sakit umum daerah (rsud) padang. *Jurnal momentum*.
- Monkey., 2020. Sejarah cv. Monkey teknologi indonesia. Retrieved february 15, 2020, from <https://monkey.id/blog/gnw16ngksnqhwlvfmaso/detail>
- Partono P, et.al., 2012. Perancangan Sistem Informasi Transaksi Tabungan Bank Sampah Garut. *Jurnal STT-Garut All Right Reserved*. Vol. 09 No. 31 2012
- Pressman, r. S. 2010. *Software engineering: a practitioner's approach*. New york. Usa: mcgraw-hill.
- Robert M, et.al., 2015. Analisis pengembangan dan perancangan sistem informasi akademik *smart* berbasis *cloud computing* pada sekolah menengah umum negeri (smun) di daerah istimewa yogyakarta. *Jurnal telematika*.
- Sudarma M, et.al., 2018. Implementasi layanan *cloud computing software as a service* pada usaha mikro kecil dan menengah. *Majalah ilmiah teknologi elektro*.
- Satzinger J, et.al., 2012. *System analysis and design in a changing world*. Usa: cengage learning.
- Schwaber K dan Sutherland J., 2016. *The scrum guidetm the definitive guide to scrum: the rules of the game*. Scrum.org and scruminc.
- Sommerville I., 2011. *Software engineering (9th edition)*. Usa: pearson education.
- Subaeki B., 2014. Perancangan arsitektur sistem informasi menggunakan metode *enterprise architecture* planning (studi kasus: universitas purwakarta - purwakarta). *Jurnal informatika*.
- Tersiana, A. 2018, Metode Penelitian, Yogyakarta: Komputer dan Teknologi.
- The Open Group., 2011. The Open Group Architecture Framework (TOGAF) (9.1 ed.). U.S.
- Tohirin T, et.al., 2020., Covid-19, Implementasi *devops* pada pengembangan aplikasi *e-skrining*. *Jurnal multimedia networking informatika*.
- Utomo, a. P., 2015. Penerapan konsep *saas (software as a service)* pada aplikasi pengajian. *Simetris*.

LAMPIRAN

Lampiran 1 *Devops CI*

```
stages:
  - test
  - build
  - deploy

cache:
  policy: pull
  paths:
    - node_modules/

image: node:10

test:nglint:
  stage: test
  only:
    - develop
    - sit
    - master
  before_script:
    - npm ci
    - npm i -g @angular/cli
  script:
    - npm run lint
  variables:
    - $CI_COMMIT_MESSAGE =~ /skip-ci/

build:
  stage: build
  environment:
    name: sit
    url: https://sit.nguang.id
  only:
    - sit
  before_script:
    - npm ci
    - npm install --quiet
  script:
    - npm run build:nguang-sit
  artifacts:
    paths:
      - dist/sit/
  except:
    variables:
      - $CI_COMMIT_MESSAGE =~ /skip-ci/
```

Lampiran 2 *Devops CD*

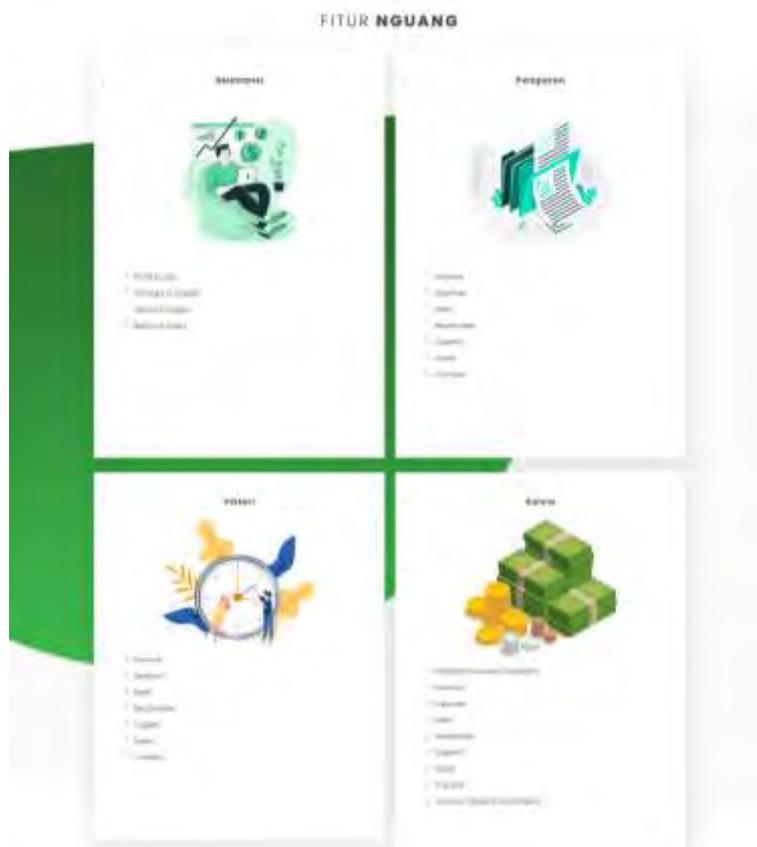
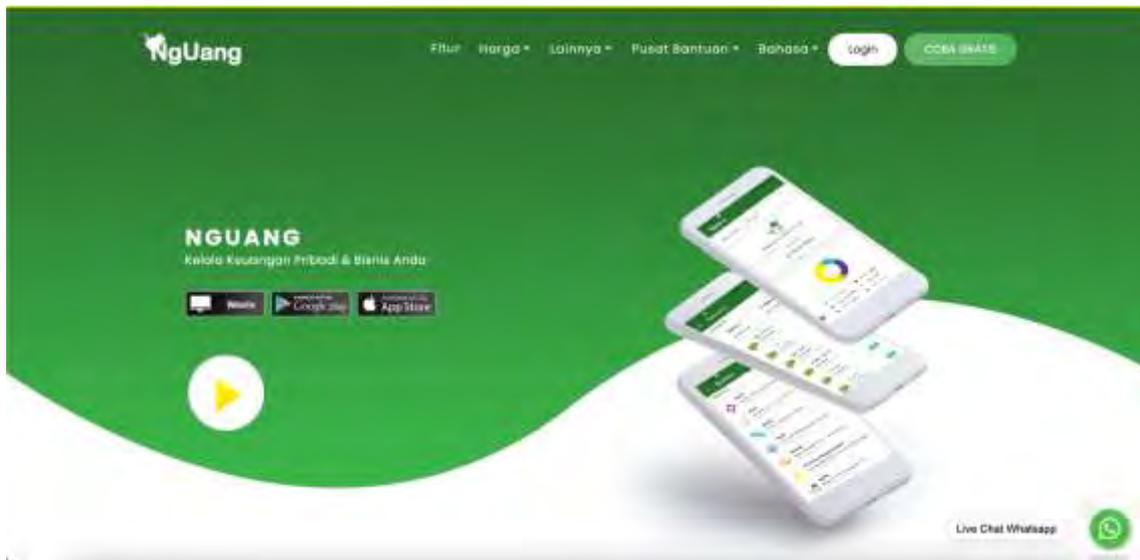
```
deploy:
  stage: deploy
  environment:
    name: prod
    url: https://nguang.id
  only:
    - master
  dependencies:
    - build:prod
  before_script:
    - npm ci
    - npm i firebase
    - npm i -g firebase-tools
  script:
    - npm run use:prod
    - npm run target:prod-nguang
    - npm run deploy-prod:firestore
```

```

- npm run deploy-prod:storage
- npm run deploy-prod-nguang:hosting
except:
variables:
- $CI_COMMIT_MESSAGE =~ /skip-ci/

```

Lampiran 3 Screenshot Website Landing Page



Berlangganan Aplikasi

GRATIS	PREMIUM
<p>RP0.00 ARUH GRATIS</p> <ul style="list-style-type: none"> Manajemen Rekening Finansial Manajemen Anggaran (Cicil, Kalk, Sisa) Transfer Rekening Account Manajemen Kas, Utang, Debit, dan Kredit History Income, Expense, Debt & Asset Accounting Income, Expense, Debt, Income, Capital Bank Forecast Manajemen Transaksi Currency Manajemen Profil 	<p>RP199,999.99 SUA SEMUA FITUR</p> <ul style="list-style-type: none"> Manajemen Rekening Finansial Manajemen Multiple Business/Company Finance Manajemen Anggaran (Cicil, Kalk, Sisa) Transfer Rekening Account Manajemen Kas, Utang, Debit, Income, Capital, Asset History Income, Expense, Debt, Income, Capital, Asset Accounting Profit & Loss, Change in Capital, Balance Ledger, Balance Sheet Bank Forecast Manajemen Transaksi Currency Manajemen Profil

PEMBAYARAN

 <p>BCA Atas Nama: Al Abdul Wahid Nama Rekening: 0211942011</p>	 <p>BNI Atas Nama: Al Abdul Wahid Nama Rekening: 1161772014</p>	 <p>BRI Atas Nama: Al Abdul Wahid Nama Rekening: 01210040110101</p>	 <p>Mandiri Atas Nama: Al Abdul Wahid Nama Rekening: 11112111111111</p>
 <p>OCBC NISP Atas Nama: Al Abdul Wahid Nama Rekening: 664810111111</p>	 <p>Digibank Atas Nama: Al Abdul Wahid Nama Rekening: 1114011111</p>	 <p>Permata Syariah Atas Nama: Al Abdul Wahid Nama Rekening: 1111111111</p>	 <p>BTPN Atas Nama: Al Abdul Wahid Nama Rekening: 1111111111</p>

TENTANG KAMI

Home / Tentang Kami



NgUang By Monkey

Kelola Keuangan Pribadi & Bisnis Anda

Visi :

Menjadi startup nomor 1 di Indonesia untuk mengelola keuangan pribadi & bisnis Anda

Misi :

Manajemen keuangan pribadi & bisnis Anda

Centennial Tower, Lantai 20,
Jakarta Selatan, Indonesia
Centennial Tower, Lantai 20, Unit D 5, 4,
Jenderal Gatot Subroto Kav. 24-25
RT/RW 002/002, Kel. Karet Semanggi, Klu.
Setiabudi, Jakarta Selatan, DC, Jakarta
12920

Contact Us:

Email Info: admin@nguang.id

Email CS: cs@nguang.id

Email Career: career@nguang.id

Nomor Telepon:

021529990043

Terhubung dengan Kami!



TIM CERDAS KAMI



ALI ABDUL WAHID

FOUNDER

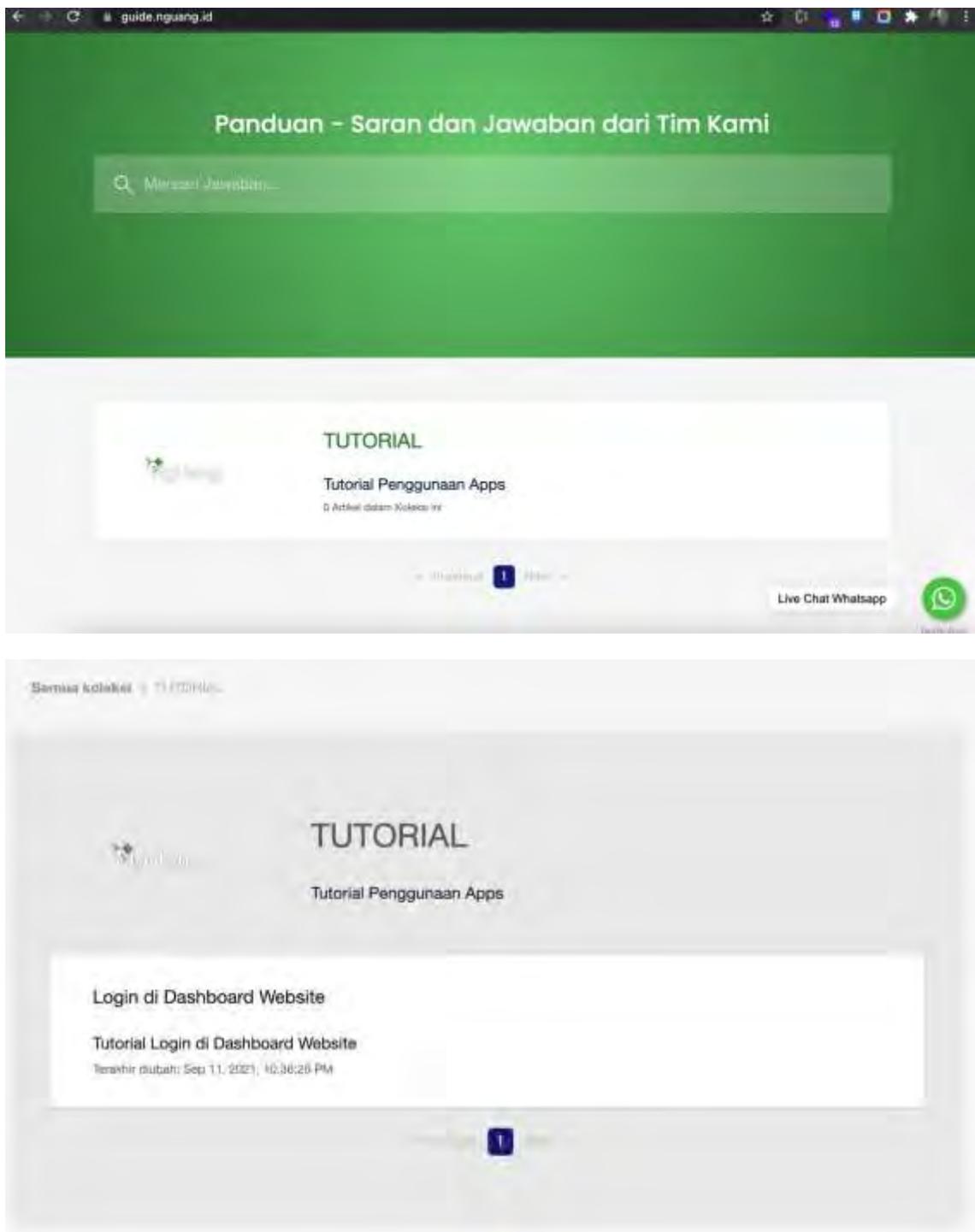


IRFAN MAYENDRA PUTRA

ACCOUNT MANAGER | PROGRAMMER



Lampiran 4 Screenshot Website Guide





Lampiran 5 Screenshot Mobile Apps

