



**PENGEMBANGAN STARTUP ALT.MINE.ID UNTUK PENAMBANGAN
CRYPTOCURRENCY BAGI INVESTOR DI INDONESIA**



Oleh:

AL ARIF RAHMADIKA PUTRA WICAKSONO

14.41010.0061

**PENGEMBANGAN STARTUP ALT.MINE.ID UNTUK PENAMBANGAN
CRYPTOCURRENCY BAGI INVESTOR DI INDONESIA**

TUGAS AKHIR

Diajukan sebagai syarat untuk menyelesaikan

Program Sarjana



FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA
2019

TUGAS AKHIR
PENGEMBANGAN STARTUP ALT.MINE.ID UNTUK PENAMBANGAN
CRYPTOCURRENCY BAGI INVESTOR DI INDONESIA

dipersiapkan dan disusun oleh
Al Arif Rahmadika Putra Wicaksono
NIM : 14.41010.0061

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji
Pada 28 Februari 2019

Susunan Dewan Penguji

Pembimbing

- I. **Sri Hariani Eko Wulandari, S.Kom., M.MT.**
NIDN 0726017801

myCMB 28/19

- II. **Julianto Lemantara, S.Kom., M.Eng.**
NIDN 0722108601

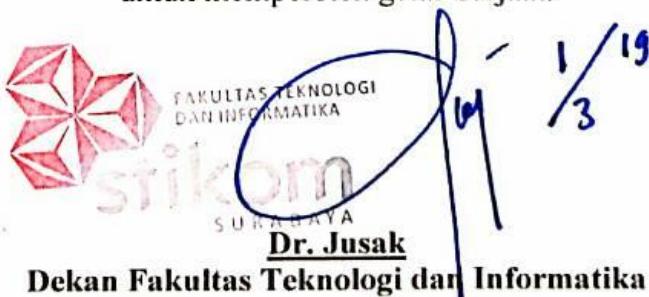
(Lemantara) 28/19

Pembahas

- I. **Tan Amelia, S.Kom., M.MT.**
NIDN 0728017602

28/19 A. Hadiwelle

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana



FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya:

Nama : Al Arif Rahmadika Putra Wicaksono
NIM : 14.41010.0061
Program Studi : S1 Sistem Informasi
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **PENGEMBANGAN STARTUP ALT.MINE.ID UNTUK PENAMBANGAN CRYPTOCURRENCY BAGI INVESTOR DI INDONESIA**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 28 Februari 2019





If I don't have to do it, I won't. If I have to do it, I'll make it quick
(Houtarou Oreki)

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

Terimakasih Kepada Allah S.W.T

Terimakasih Kepada Kedua Orang Tuaku

Terimakasih Kepada Mey Dia Astinah, S.Mat. -Bismillah 08122020 ☺

*Terimakasih Kepada Semua Orang Atas Semua Doa dan
Dukungannya*

*Kupersembahkan Kepada STIKOM Surabaya dan Orang-Orang Yang
Sering Menanyakan Kepadaku KAPAN LULUS ...*

ABSTRAK

ALT.Mine.ID adalah *startup* yang bergerak di bidang *cryptocurrency* khususnya bidang pertambangan (*mining*). *Fee* konversi mata uang dan kontrak penambangan yang bisa diputuskan kapan saja oleh perusahaan saat melakukan pembelian *hashrate* menggunakan *cloud mining* luar negeri adalah salah satu kendala yang dihadapi para penambang lokal. Selain itu penambang mandiri juga mengeluhkan besarnya modal yang harus dikeluarkan untuk memulai usaha mereka. Belum lagi kesalahan perawatan dan *overclocking hardware* yang bisa terjadi yang bisa menyebabkan kerusakan *hardware mining*.

Untuk menyelesaikan permasalahan di atas adalah dengan dibuat aplikasi *startup* berbasis *website* yang akan diberi nama ALT.Mine.ID dengan fitur manajemen coin, manajemen kontrak dan pembelian *hashrate* yang bisa menyelesaikan masing-masing masalah *miners* tersebut.

Hasil uji coba menunjukkan bahwa aplikasi ini dapat memotong biaya konversi yang harus dibayar karena sistem pembayaran menggunakan transfer bank lokal. Selain itu kontrak penambangan juga tidak akan berhenti di tengah jalan karena biaya *maintenance* dibebankan pada awal pembelian *hashrate*. Kesalahan *overclocking* dan perawatan *hadware* sepenuhnya akan menjadi tanggungjawab penyedia *startup* sehingga penambang tidak perlu lagi memikirkan bagaimana cara melakukan perawatan *hardware*.

Kata Kunci: *Bitcoin, Altcoin, Mining, Web, Scrum*

KATA PENGANTAR

Assalamu'alaikum Wr.Wb

Puji syukur Alhamdulillah kehadirat Allah SWT atas rahmat dan karunia-Nya. Shalawat salam kepada Rasulullah Muhammad SAW dan semua umatnya. Sehingga penulis bisa menyelesaikan penyusunan Laporan Tugas Akhir ini yang berjudul “Pengembangan *Startup ALT.Mine.ID Untuk Penambangan Cryptocurrency Bagi Investor Di Indonesia*”. Tugas Akhir ini disusun dalam rangka memenuhi salah satu syarat dalam menempuh Strata1 di Institut Bisnis dan Informatika Stikom Surabaya, Jawa Timur.

Menyusun laporan Tugas Akhir ini merupakan hal yang rumit bagi penulis. Namun Laporan ini dapat diselesaikan karena adanya bimbingan, petunjuk, pengarahan dan bantuan serta dorongan dari berbagai pihak yang sudah membantu penulis dengan tulus dan ikhlas dalam pembuatan Laporan Tugas Akhir ini. Jadi pada kesempatan ini, izinkan penulis mengucapkan banyak terima kasih kepada :

1. Ibu, Ayah, Saudara, dan Kekasih yang selalu memberikan semangat selama pembuatan Laporan Tugas Akhir.
2. Ibu Sri Hariani Eko Wulandari, S.Kom., M.MT. selaku dosen pembimbing I yang telah memberikan dukungan penuh berupa motivasi, wawasan, pemahaman, dan semangat selama pembuatan Laporan Tugas Akhir.
3. Bapak Julianto Lemantara, S.Kom., M.Eng. selaku dosen pembimbing II yang banyak memberikan saran, koreksi, masukan, dan semangat selama pembuatan Laporan Tugas Akhir.

4. Ibu Tan Amelia, S.Kom., M.MT. selaku dosen penguji yang telah memberikan arahan dan masukan dalam pembuatan Laporan Tugas Akhir.

5. Semua rekan-rekan mahasiswa khususnya angkatan 2014 yang telah memberikan dukungan dan masukan dalam pembuatan Laporan Tugas Akhir.

Penulis sadar bahwa penyusunan Laporan Tugas Akhir ini masih jauh dari kata sempurna. Sehingga kritik dan saran baik dari dosen maupun rekan-rekan mahasiswa/mahasiswi sangat diharapkan untuk membantu memperbaiki penulisan.

Akhir kata penulis berharap semoga Laporan Tugas Akhir ini bermanfaat bagi pembaca, khususnya bagi yang membutuhkan. Amin

Wassalamu'alaikum Wr.Wb

Surabaya, 28 Februari 2019



DAFTAR ISI

Halaman

ABSTRAK vi

KATA PENGANTAR vii

DAFTAR ISI ix

DAFTAR GAMBAR xiv

DAFTAR TABEL xix

BAB I PENDAHULUAN 1

 1.1. Latar Belakang 1

 1.2. Rumusan Masalah 5

 1.3. Batasan Masalah 5

 1.4. Tujuan 6

 1.5. Manfaat 6

 1.6. Sistematika Penulisan 6

BAB II LANDASAN TEORI 8

 2.1. *Human Computer Interaction (HCI)* Dalam Teknologi Bitcoin 8

 2.2. *Cryptocurrency* 13

 2.3. Hukum *Cryptocurrency* di Indonesia 14

 2.4. Bitcoin 16

 2.5. Ethereum 17

 2.6. *Pool Mining* 18

 2.7. *Solo Mining* 18

Halaman

| | |
|---|----|
| 2.8. <i>Directed Acyclic Graph (DAG)</i> | 19 |
| 2.9. <i>Alternative Coin</i> | 19 |
| 2.10. <i>Hashrate</i> | 20 |
| 2.11. <i>GPU Mining</i> | 20 |
| 2.12. <i>Blocks</i> | 21 |
| 2.13. <i>Blockchain</i> | 22 |
| 2.14. <i>Mining</i> | 22 |
| 2.15. <i>Website</i> | 23 |
| 2.16. <i>PHP</i> | 24 |
| 2.17. <i>MySql</i> | 24 |
| 2.18. <i>Lean Startup</i> | 25 |
| 2.19. <i>Scrum</i> | 25 |
| 2.20. <i>Daily Scrum</i> | 28 |
| 2.21. <i>Scrum Team</i> | 30 |
| BAB III ANALISIS DAN PERANCANGAN | 32 |
| 3.1. Mengidentifikasi Masalah | 32 |
| 3.2. <i>Product Backlog</i> | 50 |
| 3.3. <i>Sprint Planning</i> | 57 |
| 3.3.1. <i>Sprint Planning 1 Manajemen Coin</i> | 58 |
| 3.3.2. <i>Sprint Planning 2 Manajemen Kontrak</i> | 58 |
| 3.3.3. <i>Sprint Planning 3 Free Sign-Up Member</i> | 58 |
| 3.3.4. <i>Sprint Planning 4 Pembelian Hashrate</i> | 58 |
| 3.3.5. <i>Sprint Planning 5 Informasi Balance</i> | 58 |

Halaman

| | |
|---|----|
| 3.3.6. <i>Sprint Planning 6 Approval Withdrawal</i> | 59 |
| 3.3.7. <i>Sprint Planning 7 Laporan</i> | 59 |
| 3.4. <i>Sprint Backlog</i> | 59 |
| 3.4.1. <i>Sprint Backlog 1 Manajemen Coin</i> | 59 |
| 3.4.2. <i>Sprint Backlog 2 Manajemen Kontrak</i> | 59 |
| 3.4.3. <i>Sprint Backlog 3 Free Sign-Up Member</i> | 60 |
| 3.4.4. <i>Sprint Backlog 4 Pembelian Hashrate</i> | 60 |
| 3.4.5. <i>Sprint Backlog 5 Informasi Balance</i> | 60 |
| 3.4.6. <i>Sprint Backlog 6 Approval Withdrawal</i> | 60 |
| 3.4.7. <i>Sprint Backlog 7 Laporan</i> | 60 |
| BAB IV IMPLEMENTASI DAN EVALUASI..... | 61 |
| 4.1. <i>Sprint 1 Manajemen Coin</i> | 61 |
| 4.1.1. <i>Sprint Planning</i> | 61 |
| 4.1.2. <i>Sprint Backlog</i> | 61 |
| 4.1.3. <i>Daily Scrum</i> | 62 |
| 4.1.4. Pengerjaan <i>Sprint</i> | 63 |
| 4.1.5. <i>Sprint Review</i> | 78 |
| 4.2. <i>Sprint 2 Manajemen Kontrak</i> | 78 |
| 4.2.1. <i>Sprint Planning</i> | 78 |
| 4.2.2. <i>Sprint Backlog</i> | 79 |
| 4.2.3. <i>Daily Scrum</i> | 79 |
| 4.2.4. Pengerjaan <i>Sprint</i> | 80 |
| 4.2.5. <i>Sprint Review</i> | 87 |

Halaman

| | | |
|--------|---|-----|
| 4.3. | <i>Sprint 3 Sign-Up Member</i> | 87 |
| 4.3.1. | <i>Sprint Planning</i> | 87 |
| 4.3.2. | <i>Sprint Backlog</i> | 87 |
| 4.3.3. | <i>Daily Scrum</i> | 88 |
| 4.3.4. | Pengerjaan <i>Sprint</i> | 88 |
| 4.3.5. | <i>Sprint Review</i> | 105 |
| 4.4. | <i>Sprint 4 Pembelian Hashrate</i> | 106 |
| 4.4.1. | <i>Sprint Planning</i> | 106 |
| 4.4.2. | <i>Sprint Backlog</i> | 106 |
| 4.4.3. | <i>Daily Scrum</i> | 107 |
| 4.4.4. | Pengerjaan <i>Sprint</i> | 108 |
| 4.4.5. | <i>Sprint Review</i> | 122 |
| 4.5. | <i>Sprint 5 Informasi Balance</i> | 123 |
| 4.5.1. | <i>Sprint Planning</i> | 123 |
| 4.5.2. | <i>Sprint Backlog</i> | 123 |
| 4.5.3. | <i>Daily Scrum</i> | 124 |
| 4.5.4. | Pengerjaan <i>Sprint</i> | 124 |
| 4.5.5. | <i>Sprint Review</i> | 134 |
| 4.6. | <i>Sprint 6 Approval Withdrawal</i> | 134 |
| 4.6.1. | <i>Sprint Planning</i> | 134 |
| 4.6.2. | <i>Sprint Backlog</i> | 135 |
| 4.6.3. | <i>Daily Scrum</i> | 135 |
| 4.6.4. | Pengerjaan <i>Sprint</i> | 136 |

Halaman

| | |
|-------------------------------------|-----|
| 4.6.5. <i>Sprint Review</i> | 143 |
| 4.7. <i>Sprint 7 Laporan</i> | 143 |
| 4.7.1. <i>Sprint Planning</i> | 143 |
| 4.7.2. <i>Sprint Backlog</i> | 144 |
| 4.7.3. <i>Daily Scrum</i> | 144 |
| 4.7.4. Pengerjaan Sprint | 145 |
| 4.7.5. <i>Sprint Review</i> | 156 |
| 4.8. Pembahasan | 157 |
| BAB V KESIMPULAN DAN SARAN..... | 159 |
| 5.1. Kesimpulan..... | 159 |
| 5.2. Saran | 159 |
| DAFTAR PUSTAKA | 160 |
| BIODATA PENULIS | 162 |

DAFTAR GAMBAR

| | |
|--|---------|
| | Halaman |
| Gambar 1.1 Investasi <i>Cryprocurrency</i> 2013-2018 (coinmarketcap.com) | 2 |
| Gambar 1.2 Dominasi Bitcoin 2013-2018 (coinmarketcap.com) | 2 |
| Gambar 1.3 Harga Bitcoin Periode 2017 (databoks.katadata.co.id) | 3 |
| Gambar 2.1 Kerangka Penelitian HCI Bitcoin. (Sas & Khairuddin, 2015) | 8 |
| Gambar 2.2 Langkah Kerja Metode <i>Scrum</i> | 26 |
| Gambar 2.3 <i>Daily Scrum</i> | 28 |
| Gambar 2.4 <i>Daily Scrum (In Progress)</i> | 29 |
| Gambar 2.5 <i>Daily Scrum (Done)</i> | 29 |
| Gambar 3.1 Tahapan Penelitian Berdasarkan SCRUM | 32 |
| Gambar 3.2 Alur Proses Bisnis Saat Ini | 33 |
| Gambar 3.3 BMC Startup ALT.Mine.ID..... | 34 |
| Gambar 3.4 Desain Arsitektur..... | 49 |
| Gambar 3.5 Blok Diagram | 51 |
| Gambar 4.1 <i>Daily Scrum Sprint 1</i> | 62 |
| Gambar 4.2 <i>System Flow</i> Manajemen Coin | 63 |
| Gambar 4.3 DFD Level 1 Manajemen <i>Coin</i> | 64 |
| Gambar 4.4 CDM Manajemen <i>Coin</i> | 66 |
| Gambar 4. 5 PDM Manajemen <i>Coin</i> | 66 |
| Gambar 4.6 Desain Halaman Login Admin..... | 67 |
| Gambar 4.7 Desain Halaman Utama..... | 67 |
| Gambar 4.8 Desain Halaman Admin | 68 |

| | |
|---|----|
| Gambar 4.9 Halaman Manajemen <i>Coin</i> | 68 |
| Gambar 4.10 Halaman Manajemen Paket..... | 68 |
| Gambar 4.11 Halaman <i>Login Admin</i> | 71 |
| Gambar 4.12 Pesan Gagal <i>Login</i> | 71 |
| Gambar 4.13 Halaman Utama..... | 72 |
| Gambar 4.14 Halaman Admin | 72 |
| Gambar 4.15 Halaman Manajemen Coin..... | 73 |
| Gambar 4.16 Halaman Tambah Data <i>Coin</i> | 73 |
| Gambar 4.17 Halaman Edit Data <i>Coin</i> | 73 |
| Gambar 4.18 Notifikasi Hapus Data Coin | 74 |
| Gambar 4.19 Halaman Manajemen Paket..... | 74 |
| Gambar 4.20 Halaman Tambah Data Paket..... | 75 |
| Gambar 4.21 Halaman Edit Data Paket | 75 |
| Gambar 4. 22 Notifikasi Hapus Data Paket | 75 |
| Gambar 4.23 <i>Daily Scrum Sprint 2</i> | 79 |
| Gambar 4.24 <i>System Flow</i> Manajemen Kontrak..... | 80 |
| Gambar 4.25 DFD Level 0 Manajemen Kontrak..... | 81 |
| Gambar 4.26 CDM Manajemen Kontrak..... | 82 |
| Gambar 4.27 PDM Manajemen Kontrak | 82 |
| Gambar 4. 28 Halaman Manajemen Kontrak | 83 |
| Gambar 4.29 Halaman Manajemen Kontrak | 84 |
| Gambar 4.30 Halaman Tambah Data Kontrak..... | 85 |
| Gambar 4.31 Halaman Edit Data Kontrak | 85 |
| Gambar 4.32 Notifikasi Hapus Data Paket | 86 |

| | |
|--|-----|
| Gambar 4. 33 <i>Daily Scrum Sprint 3</i> | 88 |
| Gambar 4.34 <i>System Flow Registrasi Customer</i> | 89 |
| Gambar 4.35 DFD Level 1 <i>Free Sign-Up Member</i> | 90 |
| Gambar 4.36 CDM Free Sign-Up Member | 93 |
| Gambar 4.37 PDM <i>Free Sign-Up Member</i> | 93 |
| Gambar 4.38 Desain Halaman <i>Login Customer</i> | 94 |
| Gambar 4.39 Desain Halaman <i>Customer</i> | 95 |
| Gambar 4.40 Desain Halaman <i>Register</i> | 95 |
| Gambar 4.41 Desain Halaman <i>Account</i> | 95 |
| Gambar 4.42 Halaman <i>Request Verifikasi</i> | 96 |
| Gambar 4.43 Implementasi Halaman <i>Register</i> | 99 |
| Gambar 4. 44 Halaman Login Customer | 100 |
| Gambar 4. 45 Pesan Gagal Login | 100 |
| Gambar 4.46 Halaman <i>Customer</i> | 100 |
| Gambar 4.47 Halaman <i>Account</i> | 101 |
| Gambar 4. 48 Halaman Verifikasi | 101 |
| Gambar 4.49 Halaman <i>Edit Wallet</i> | 102 |
| Gambar 4.50 Halaman <i>Request Verifikasi</i> | 102 |
| Gambar 4.51 Halaman Verifikasi <i>Request Verifikasi</i> | 103 |
| Gambar 4. 52 <i>Daily Scrum Sprint 4</i> | 107 |
| Gambar 4.53 <i>System Flow Pembelian Hashrate</i> | 108 |
| Gambar 4.54 DFD Level 1 <i>Pembelian Hashrate</i> | 109 |
| Gambar 4.55 CDM <i>Pembelian Hashrate</i> | 111 |
| Gambar 4.56 PDM <i>Pembelian Hashrate</i> | 112 |

| | |
|---|-----|
| Gambar 4.57 Desain Halaman Pembelian | 112 |
| Gambar 4.58 Desain Halaman Pembayaran..... | 113 |
| Gambar 4.59 Halaman <i>Request</i> Pembayaran | 113 |
| Gambar 4.60 Halaman <i>Setup Mining</i> | 113 |
| Gambar 4.61 Halaman Pembelian..... | 116 |
| Gambar 4.62 Halaman Detil Pembelian..... | 117 |
| Gambar 4.63 Halaman Pesanan Saya..... | 117 |
| Gambar 4.64 Halaman Pembayaran..... | 118 |
| Gambar 4.65 Halaman <i>Setup Mining</i> | 118 |
| Gambar 4.66 Halaman <i>Edit Data Setup Mining</i> | 119 |
| Gambar 4.67 Halaman <i>Request</i> Pembayaran | 119 |
| Gambar 4.68 Halaman Verifikasi <i>Request</i> Pembayaran..... | 120 |
| Gambar 4.69 <i>Daily Scrum Sprint 4</i> | 124 |
| Gambar 4.70 System Flow Informasi <i>Balance</i> | 125 |
| Gambar 4.71 DFD Level 1 Informasi <i>Balance</i> | 126 |
| Gambar 4.72 CDM Informasi <i>Balance</i> | 128 |
| Gambar 4.73 PDM Informasi <i>Balance</i> | 128 |
| Gambar 4.74 Desain Halaman Informasi <i>Balance</i> | 129 |
| Gambar 4.75 Desain Halaman <i>History Withdraw</i> | 129 |
| Gambar 4.76 Halaman Informasi <i>Balance</i> | 131 |
| Gambar 4.77 Halaman <i>Withdraw</i> | 132 |
| Gambar 4.78 Halaman <i>Request Withdrawal</i> | 132 |
| Gambar 4.79 <i>Daily Scrum Sprint 6</i> | 135 |
| Gambar 4.80 System Flow Approval <i>Withdrawal</i> | 136 |

| | |
|--|-----|
| Gambar 4.81 DFD Level 1 <i>Approval Withdrawal</i> | 137 |
| Gambar 4.82 CDM <i>Approval Withdrawal</i> | 139 |
| Gambar 4.83 PDM <i>Approval Withdrawal</i> | 139 |
| Gambar 4.84 Halaman <i>Request Withdraw</i> | 140 |
| Gambar 4.85 Halaman <i>Request Withdraw</i> | 142 |
| Gambar 4.86 Halaman Verifikasi <i>Request Withdraw</i> | 142 |
| Gambar 4.87 Daily Scrum Sprint 7 | 145 |
| Gambar 4.88 System Flow Laporan | 146 |
| Gambar 4.89 DFD Level 1 Laporan | 146 |
| Gambar 4.90 DFD Level 0 Keseluruhan Sistem..... | 147 |
| Gambar 4.91 <i>Context Diagram</i> Keseluruhan Sistem..... | 148 |
| Gambar 4.92 Desain Halaman <i>Owner</i> | 150 |
| Gambar 4.93 Desain Halaman Laporan Pendapatan..... | 151 |
| Gambar 4.94 Desain Halaman Laporan Stok Coin | 151 |
| Gambar 4.95 Desain Halaman Laporan <i>Customer</i> | 151 |
| Gambar 4.96 Halaman <i>Owner</i> | 153 |
| Gambar 4.97 Halaman Laporan Pendapatan..... | 154 |
| Gambar 4.98 Halaman Laporan Stok Coin | 154 |
| Gambar 4.99 Halaman Laporan <i>Customer</i> | 155 |

DAFTAR TABEL

| | |
|--|---------|
| | Halaman |
| Tabel 3.1 Identifikasi Masalah..... | 36 |
| Tabel 3.2 Kebutuhan Fungsi Penambahan <i>Coin</i> Baru | 37 |
| Tabel 3.3 Kebutuhan Fungsi <i>Setting</i> Paket <i>Altcoin</i> | 38 |
| Tabel 3.4 Kebutuhan Fungsi <i>Setting</i> Kontrak <i>Altcoin</i> | 39 |
| Tabel 3.5 Kebutuhan Fungsi Verifikasi | 40 |
| Tabel 3.6 Kebutuhan Fungsi <i>Edit Wallet</i> | 41 |
| Tabel 3.7 Kebutuhan Fungsi <i>Upload</i> Foto..... | 41 |
| Tabel 3.8 Kebutuhan Fungsi Pembelian <i>Hashrate</i> | 42 |
| Tabel 3.9 Kebutuhan Fungsi Konfirmasi Pembayaran | 43 |
| Tabel 3.10 Kebutuhan Fungsi Cek Balance Altcoin..... | 44 |
| Tabel 3.11 Kebutuhan Fungsi <i>Request Withdrawal</i> | 45 |
| Tabel 3.12 Kebutuhan Fungsi Konfirmasi Pembayaran <i>Altcoin</i> | 46 |
| Tabel 3.13 Kebutuhan Fungsi Laporan..... | 47 |
| Tabel 3.14 Kebutuhan Non-Fungsional | 48 |
| Tabel 3.15 <i>Product Backlog</i> | 50 |
| Tabel 3.16 <i>Sprint Story</i> | 50 |
| Tabel 4.1 <i>Sprint Backlog</i> Manajemen <i>Coin</i> | 62 |
| Tabel 4.2 Tabel <i>Coin</i> | 65 |
| Tabel 4.3 Tabel Paket..... | 65 |
| Tabel 4.4 Desain Uji Coba <i>Form Login</i> | 69 |
| Tabel 4.5 Desain Uji Coba Manajemen <i>Coin</i> | 69 |

Halaman

| | |
|--|-----|
| Tabel 4.6 Desain Uji Coba Manajemen Paket | 70 |
| Tabel 4.7 Uji Coba Form Login..... | 76 |
| Tabel 4.8 Uji Coba Menu Manajemen <i>Coin</i> | 77 |
| Tabel 4.9 Uji Coba Menu Manajemen Paket..... | 77 |
| Tabel 4.10 Perbaikan <i>Sprint Review</i> Pertama | 78 |
| Tabel 4.11 Sprint Backlog Manajemen Kontrak..... | 79 |
| Tabel 4.12 Tabel Kontrak | 82 |
| Tabel 4.13 Desain Uji Coba Manajemen Kontrak | 84 |
| Tabel 4.14 Uji Coba Menu Manajemen Kontrak..... | 86 |
| Tabel 4.15 <i>Sprint Backlog Free Sign-Up Member</i> | 88 |
| Tabel 4.16 Tabel <i>Customer</i> | 91 |
| Tabel 4.17 Tabel <i>Wallet</i> | 92 |
| Tabel 4.18 Tabel Verif | 92 |
| Tabel 4.19 Desain Uji Coba <i>Form Login</i> | 96 |
| Tabel 4.20 Desain Uji Coba <i>Form Register</i> | 97 |
| Tabel 4.21 Desain Uji Coba Menu <i>Account</i> | 97 |
| Tabel 4.22 Desain Uji Coba Menu Request Verifikasi..... | 98 |
| Tabel 4.23 Uji Coba <i>Form Login Customer</i> | 103 |
| Tabel 4.24 Uji Coba Form <i>Registrasi Customer</i> | 104 |
| Tabel 4.25 Uji Coba Menu <i>Account</i> | 104 |
| Tabel 4.26 Uji Coba Menu <i>Request Verifikasi</i> | 105 |
| Tabel 4. 27 <i>Sprint Backlog Pembelian Hashrate</i> | 107 |
| Tabel 4.28 Tabel Pemesanan..... | 110 |

Halaman

| | |
|---|-----|
| Tabel 4.29 Tabel Pembayaran..... | 110 |
| Tabel 4.30 Desain Uji Coba Menu Pembelian..... | 114 |
| Tabel 4.31 Desain Uji Coba Menu Pesanan Saya..... | 115 |
| Tabel 4.32 Desain Uji Coba Menu <i>Request</i> Pembayaran | 115 |
| Tabel 4.33 Desain Uji Coba Menu Proses <i>Mining</i> | 116 |
| Tabel 4.34 Uji Coba Menu Pembelian | 120 |
| Tabel 4.35 Uji Coba Menu Pesanan Saya..... | 121 |
| Tabel 4.36 Uji Coba Menu <i>Request</i> Pembayaran | 121 |
| Tabel 4.37 Uji Coba Menu <i>Setup Mining</i> | 122 |
| Tabel 4.38 Perbaikan <i>Sprint Review</i> Keempat..... | 122 |
| Tabel 4.39 <i>Sprint Backlog</i> Informasi <i>Balance</i> | 123 |
| Tabel 4.40 Tabel Proses <i>Mining</i> | 127 |
| Tabel 4.41 Tabel <i>Withdraw</i> | 127 |
| Tabel 4.42 Desain Uji Coba Menu Informasi <i>Balance</i> | 130 |
| Tabel 4.43 Desain Uji Coba Menu <i>Request Withdrawal</i> | 131 |
| Tabel 4.44 Uji Coba Menu Informasi <i>Balance</i> | 133 |
| Tabel 4.45 Uji Coba Menu <i>Request Withdrawal Customer</i> | 133 |
| Tabel 4.46 Perbaikan <i>Sprint Review</i> Kelima..... | 134 |
| Tabel 4.47 <i>Sprint Backlog Approval Withdrawal</i> | 135 |
| Tabel 4.48 Tabel <i>Withdraw</i> | 138 |
| Tabel 4.49 Desain Uji Coba Menu <i>Request Withdraw</i> | 141 |
| Tabel 4.50 Uji Coba Menu <i>Request Withdrawal Admin</i> | 143 |
| Tabel 4.51 Perbaikan <i>Sprint Review</i> Keenam | 143 |

Halaman

| | |
|--|-----|
| Tabel 4.52 <i>Sprint Backlog</i> Laporan | 144 |
| Tabel 4.53 Tabel <i>Customer</i> | 149 |
| Tabel 4.54 Tabel <i>Coin</i> | 149 |
| Tabel 4.55 Tabel Pembayaran..... | 150 |
| Tabel 4.56 Desain Uji Coba Menu Laporan Pendapatan..... | 152 |
| Tabel 4.57 Desain Uji Coba Menu Laporan Stok <i>Coin</i> | 152 |
| Tabel 4.58 Desain Uji Coba Menu Laporan <i>Customer</i> | 153 |
| Tabel 4.59 Uji Coba Menu Laporan Pendapatan | 155 |
| Tabel 4.60 Uji Coba Menu Stok Coin..... | 156 |
| Tabel 4.61 Uji Coba Menu Laporan <i>Customer</i> | 156 |
| Tabel 4.62 Perbaikan <i>Sprint Review</i> Ketujuh | 156 |



BAB I

PENDAHULUAN

1.1. Latar Belakang

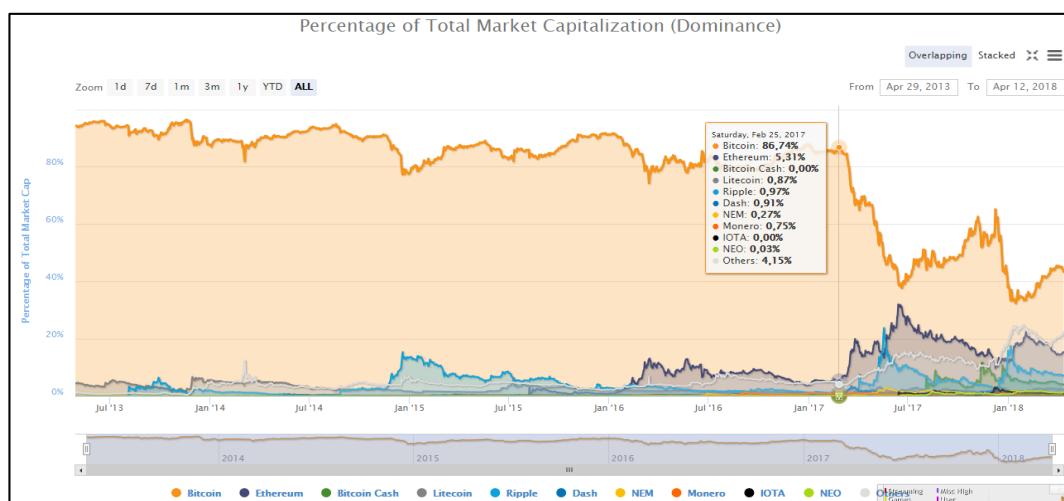
Teori ekonomi mengartikan atau mendefinisikan investasi, sebagai pengeluaran untuk membeli barang-barang modal dan peralatan-peralatan produksi dengan tujuan untuk mengganti dan terutama menambah barang-barang dan jasa di masa depan. Investasi yang lazim disebut dengan istilah penanaman modal atau pembentukan modal. Tren investasi sendiri saat ini mulai berkembang yang awalnya konvensional seperti saham, properti, dan obligasi saat ini mulai merambah ke dunia *virtual* salah satu contohnya yaitu Bitcoin. Data yang didapat dari *coinmarketcap.com* menunjukkan bahwa nilai investasi yang mengalir ke *cryptocurrency* mengalami kenaikan dari tahun 2013.



Gambar 1.1 Investasi *Cryptocurrency* 2013-2018 (coinmarketcap.com)

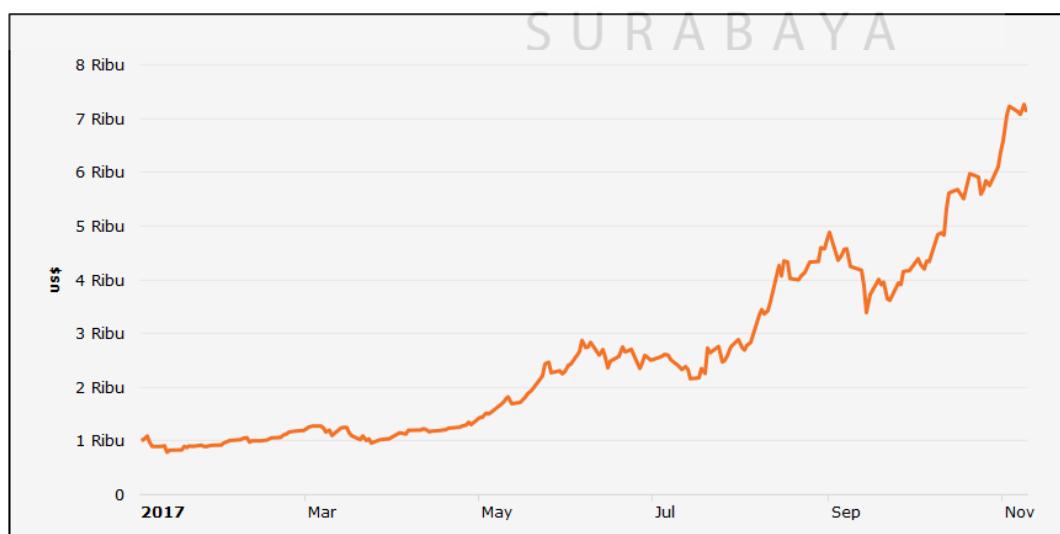
Cryptocurrency adalah alat tukar mata uang seperti USD, tetapi dirancang untuk tujuan pertukaran informasi digital melalui proses prinsip-prinsip kriptografi. Kriptografi digunakan untuk mengamankan transaksi dan untuk mengontrol penciptaan uang logam baru. *Cryptocurrency* pertama yang dibuat adalah Bitcoin pada tahun 2009. Saat ini ada ribuan *cryptocurrency* lainnya, seperti *Altcoins*. Dengan kata lain, *cryptocurrency* adalah listrik yang diubah menjadi baris kode dengan nilai moneter. Dalam bentuk sederhana, *cryptocurrency* adalah mata uang digital. Kebanyakan *cryptocurrency* dirancang untuk mengurangi produksi dari waktu ke waktu seperti Bitcoin, yang menciptakan kapitalisasi pasar. Berbeda dari mata uang fiat dimana lembaga keuangan selalu bisa menciptakan uang lebih banyak, maka inflasi bisa saja terjadi. Bitcoin tidak akan pernah lebih dari 21 juta koin yang beredar. Sistem teknis di mana semua *cryptocurrency* didasarkan sistem yang dibuat oleh Satoshi Nakamoto.

Salah satu cara investasi *cryptocurrency* yaitu dengan melakukan penambangan Alternatif coin (Altcoin) selain bitcoin. Ethereum (ETH) merupakan Altcoin paling menguntungkan saat ini yang bisa ditambang menggunakan VGA.



Gambar 1.2 Dominasi Bitcoin 2013-2018 (coinmarketcap.com)

ETH memiliki pangsa pasar terbesar kedua di dunia *cryptocurrency* dengan mendominasi pasar *crypto* sebesar 5% dan terus mengalami kenaikan setiap tahun. Selain itu ETH merupakan salah satu altcoin yang mempunyai resistant terhadap penambangan generasi 3 menggunakan ASIC. Penambangan biasanya dilakukan menggunakan aplikasi yang tersedia dalam situs ETH itu sendiri. Kumpulan data transaksi yang terkumpul menjadi satu atau biasa disebut dengan (*Block*) diciptakan setiap 12 detik. Oleh sebab itu tingkat kesulitan akan terus disesuaikan dengan kekuatan komputasi (pemecahan algoritma) di jaringan ETH. Penambangan bitcoin yang sebenarnya, adalah sebuah kegiatan menambang menggunakan perangkat tertentu. Prosesnya melalui memecahkan serangkaian persoalan matematis (*puzzle*) berdasarkan tingkat kesulitan tertentu. kegiatan ini yang disebut dengan “Menambang” atau “Bitcoin Mining”. Dari data yang didapat dari databoks.katadata.co.id sepanjang 2017 para penambang telah mendapatkan keuntungan lebih dari 600% menyesuaikan dengan terus naiknya harga bitcoin saat itu.



Gambar 1.3 Harga Bitcoin Periode 2017 (databoks.katadata.co.id)

Berdasarkan data yang didapat dari pusat penjualan komputer di Hi-Tech surabaya pada bulan Desember 2017 harga satuan VGA Rp. 4.700.000 ± sedangkan pada bulan Februari 2018 harga satuan VGA mencapai Rp. 8.800.000 ±. Hal ini cenderung menunjukkan meningkatnya peminat *miners* (Penambang) lokal yang ingin memulai usaha penambangan sendiri. Proses penambangan di *cryptocurrency* dilakukan oleh beberapa perusahaan contohnya yaitu *Genesis Mining* dan *Hashflare*. Ada beberapa kekurangan dari kedua perusahaan ini yaitu pembayaran investasi harus menggunakan mata uang *dollar* Amerika (USD) yang membuat penambang lokal harus mengeluarkan biaya lebih (*fee*) untuk melakukan konversi mata uang dari rupiah (IDR) ke USD yang pada umumnya sebesar 2% dari total pembayaran. Selain itu minimal kekuatan tambang (*hashrate*) yang ditawarkan terlalu besar yang berkisar antara 20 juta hingga 50 juta per kontrak sehingga mengakibatkan besarnya biaya investasi yang harus dikeluarkan. Sampai saat ini di Indonesia belum ada *startup* yang berbasis penambangan *cyrptocurrency*. Padahal di sisi lain investor yang ingin melakukan investasi di *cyrptocurrency* semakin meningkat. Sehingga kebanyakan investor di Indonesia kebanyakan melakukan investasi di perusahaan pertambangan luar negeri atau melakukan penambangan secara konvensional di rumah. ALT.Mine.ID merupakan *startup* dengan bentuk perusahaan perseorangan dimana pemilik usaha bertanggungjawab secara penuh terhadap jalannya perusahaan.

Startup ini dibuat sebagai lahan investasi bagi para *investor* di Indonesia yang ingin memulai investasi di dunia *cryptocurrency* dan penambang lokal yang ingin memulai penambangan altcoin namun terlalu khawatir akan biaya-biaya yang akan dikeluarkan setiap bulannya. Seperti biaya tambah daya, besarnya tagihan

listrik, perawatan *hardware*, kesalahan *overclocking hardware*, kurangnya *space*, sirkulasi udara yang kurang baik, dan lain sebagainya. Keunggulan startup ini diantaranya untuk metode pembayaran menyediakan metode pembayaran yang lebih mudah dan minimal investasi pembelian *hashrate* yang lebih kecil. Untuk mencapai tujuan itu maka *startup* ini akan membuat beberapa modul aplikasi untuk *administrator* dan *member*.

Untuk memecahkan masalah perusahaan tersebut di atas maka dalam tugas akhir ini dibuat aplikasi *startup* berbasis *website* yang akan diberi nama ALT.Mine.ID untuk memanajemen proses penambangan *cryptocurrency* yang berlangsung. Melalui aplikasi *startup* ALT.Mine.ID investor dan penambang dapat melakukan *monitoring hardware* dan *balance* pendapatan dari *mining* kapan saja dan dimana saja. Aplikasi juga mengelola pembayaran memalui transfer bank dan bisa mengelola minimal investasi yang akan dikeluarkan.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas maka masalah yang didapat yaitu bagaimana mengembangkan *startup* ALT.Mine.ID untuk manajemen penambangan *cryptocurrency* bagi investor yang tertarik di dunia *cryptocurrency* dan penambang lokal agar beralih meggunakan *startup* ALT.MINE.ID untuk memudahkan manajemen penambangan.

1.3. Batasan Masalah

Batasan masalah pada penelitian ini yaitu:

1. *Cryptocurrency* yang dibahas dalam penelitian ini terfokus pada altcoin (*coin* selain bitcoin. Terutama ETH

2. Mining yang dilakukan dalam penelitian ini adalah jenis *pool mining*.
3. VGA yang digunakan dalam penelitian ini adalah produk AMD dan NVIDIA yaitu RX570, RX580, GTX 1060, dan GTX 1070.

1.4. Tujuan

Tujuan daripada pembuatan *startup* ini adalah untuk menghasilkan *startup* ALT.Mine.ID untuk penambangan *cryptocurrency* bagi investor di indonesia yang ingin memulai investasi di dunia *cryptocurrency* dan penambang lokal untuk berinvestasi di ALT.Mine.ID dengan risiko dan modal yang lebih kecil dengan keuntungan penghasilan pasif tiap bulan.

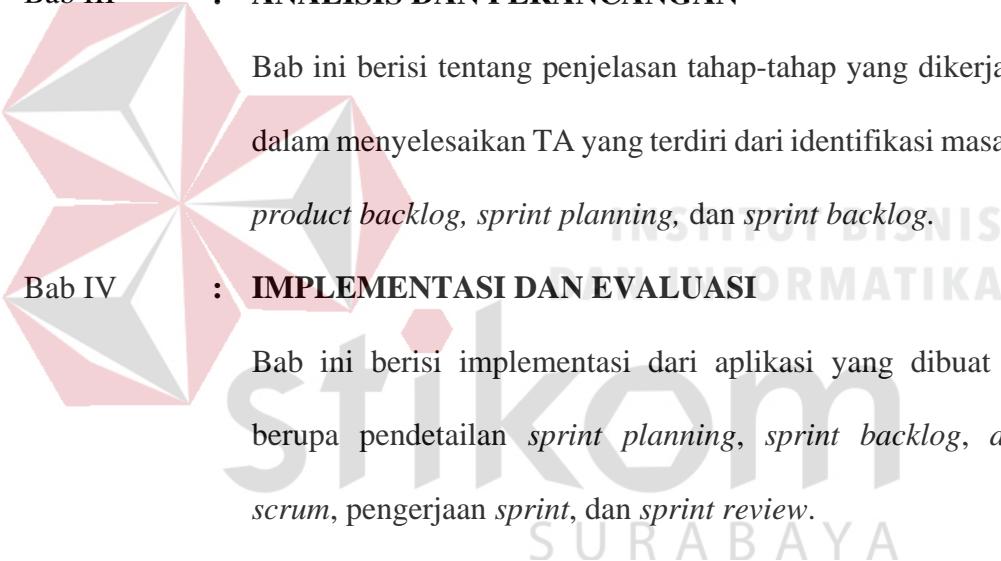
1.5. Manfaat

Manfaat dari adanya aplikasi penambangan *alternative coin* berbasis *web* ini yaitu :

- A. Memudahkan penambang dan investor lokal yang ingin memulai usaha penambangan tanpa harus memikirkan biaya tambah daya, besarnya tagihan listrik, perawatan *hardware*, kesalahan *overclocking hardware*, kurangnya *space*, sirkulasi udara yang kurang baik.
- B. Sebagai sarana investasi *cryptocurrency* di Indonesia untuk investor yang ingin mencari lahan investasi baru.

1.6. Sistematika Penulisan

Adapun sistematika penulisan Laporan Tugas Akhir ini untuk memudahkan dalam memahami persoalan dan pembahasannya yang dijabarkan sebagai berikut :

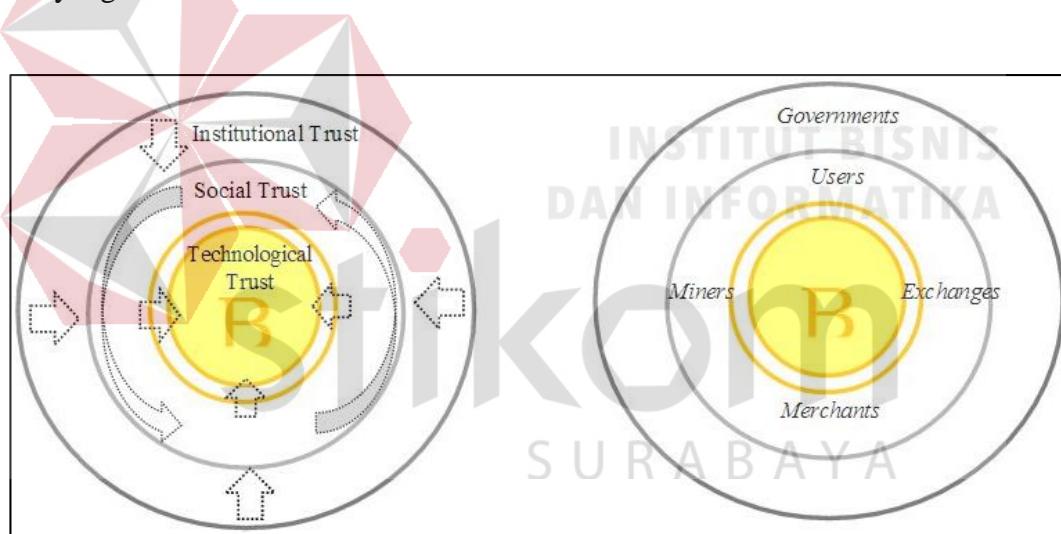
- 
- Bab I : **PENDAHULUAN**
- Bab ini dijelaskan tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan Laporan Tugas Akhir.
- Bab II : **LANDASAN TEORI**
- Bab ini berisi penjelasan teori-teori yang digunakan dalam membantu proses analisa dan desain aplikasi, seperti Bitcoin, *Mining, Website, PHP, MySQL* dan juga teori *SCRUM*.
- Bab III : **ANALISIS DAN PERANCANGAN**
- Bab ini berisi tentang penjelasan tahap-tahap yang dikerjakan dalam menyelesaikan TA yang terdiri dari identifikasi masalah, *product backlog, sprint planning, dan sprint backlog*.
- Bab IV : **IMPLEMENTASI DAN EVALUASI**
- Bab ini berisi implementasi dari aplikasi yang dibuat dan berupa pendetailan *sprint planning, sprint backlog, daily scrum, pengajaran sprint, dan sprint review*.
- Bab V : **KESIMPULAN DAN SARAN**
- Bab ini berisi tentang kesimpulan dan saran. Kesimpulan menjelaskan tentang kesimpulan dari aplikasi yang telah dibuat dan hasil evaluasi yang telah dilakukan. Saran menjelaskan tentang saran dari kekurangan aplikasi yang dibuat.

BAB II

LANDASAN TEORI

2.1. *Human Computer Interaction (HCI) Dalam Teknologi Bitcoin*

Bitcoin adalah *cryptocurrency* yang berbeda dalam cara penggunaan dengan uang tradisional. *Cryptocurrency* tidak memerlukan nama individu tapi hanya *wallet ID*, yang membuatnya lebih pribadi dan anonim. Teknologi Bitcoin saat ini kurang perlindungan sehubungan dengan transaksinya, dan struktur yang tidak didukung oleh pemerintah. Namun, memahami konsep kepercayaan adalah hal yang mendasar.



Gambar 2.1 Kerangka Penelitian HCI Bitcoin. (Sas & Khairuddin, 2015)

Berdasarkan framework di atas. Studi yang dilakukan oleh Sas & Khairuddin (2015) ini menjelaskan bahwa kerangka kerja di atas menempatkan teknologi Bitcoin di pusatnya, dan menyoroti bagaimana *stakeholder* yang berbeda terlibat dalam membentuk tiga tingkat kepercayaan yang berbeda yaitu *Institutional Trust*, *Social Trust*, dan *Technological Trust*.

1. *Institutional Trust*

Institutional Trust dalam teknologi Bitcoin. Masalah utama di sini berkaitan dengan pencucian uang dan deflasi

2. *Social Trust*

Kepercayaan sosial adalah kepercayaan yang dimiliki oleh *stakeholders* Bitcoin antara satu sama lain. Kepercayaan ini terdaftar untuk setiap jenis pertukaran yang terjadi di (dan dalam) kategori *stakeholders* yang berbeda. Misalnya transaksi pembelian barang yang melibatkan kepercayaan antara pengguna dan pedagang. Setelah selesai, transaksi ini memerlukan otorisasi penambang, sehingga pengguna dan pedagang harus memercayai para penambang untuk menyelesaikan pekerjaan mereka.

3. *Technological Trust*

Technological Trust sebagai kepercayaan masyarakat terhadap teknologi Bitcoin yang dialami sebelum, selama, dan setelah terlibat dalam transaksi online. Ini bisa termasuk kepercayaan pengguna bahwa akun Bitcoin mereka diamankan dan tidak dapat diretas, atau kepercayaan orang tua bahwa transfer tersebut diotorisasi.

Stakeholder Bitcoin, yang dikelompokkan oleh Shcherbak (2014) terdiri dari empat kategori utama yaitu pengguna, penambang, bursa dan pedagang. Pengguna adalah orang-orang yang menggunakan Bitcoin untuk membeli barang dan jasa dari pedagang Bitcoin. Pedagang adalah bisnis yang menerima Bitcoin sebagai alat tukar untuk barang dan jasa dan terhubung ke jaringan Bitcoin. Bursa adalah penyedia platform perdagangan online dimana anggota yang terdaftar dapat menukarkan Bitcoin mereka dengan mata uang tradisional dan sebaliknya.

Penambang adalah pemangku kepentingan Bitcoin yang dapat merekam transaksi (dan mengumpulkan hadiah) setelah berhasil menyelesaikan *cryptopuzzles* (Eyal & Sirer, 2014).

Berikut ini adalah hasil penerapan *framework* yang memerlukan keterlibatan HCI yang lebih kuat :

1. *User Trust* Terhadap Bitcoin

Salah satu tantangan khusus yang berkaitan dengan pengguna adalah pengetahuan mereka yang terbatas tentang bagaimana teknologi Bitcoin bekerja dan bagaimana mereka harus melindungi bitcoin mereka. Menjaga bitcoin pada komputer seseorang melibatkan risiko keamanan yang serupa dengan menyimpan sejumlah besar uang dalam dompet fisik seseorang Bitcoin Wiki (2011). Meskipun Bitcoin terdesentralisasi dan pada umumnya tidak memiliki satu titik kegagalan, namun rentan terhadap bentuk penolakan layanan (Quora Forum, 2011).

2. *Merchants Trust* Terhadap Bitcoin

Merchants Trust ditantang oleh pengetahuan terbatas mereka tentang pembeli, dan apakah pembayaran mereka akan diterima tepat waktu atau sama sekali Shcherbak (2014). Mereka juga tidak memiliki kemampuan untuk melacak pembeli yang dapat dipercaya dengan siapa mereka sebelumnya memiliki transaksi inpositif.

3. Dampak *Miners* dan *Exchanger* (Bursa) Pada *Social Trust* Bitcoin

Pertukaran sangat penting dalam mendukung kepercayaan pengguna dan pedagang, dan pada umumnya kepercayaan sosial di dalam sistem Bitcoin. Misalnya, pertukaran tidak memiliki proses audit dan tidak ada prosedur

verifikasi (Bitcoin Forum, 2010). Sama halnya, walaupun setiap transaksi harus ditandatangani dan aman secara digital setelah diverifikasi oleh penambang yang tidak dikenal, kita hanya tahu sedikit tentang mekanisme yang mengikuti kompetensi dan integritas penambang. Penelitian terbaru ini menunjukkan bahwa struktur penghargaan yang memberi insentif kepada para penambang untuk berkontribusi pada sistem dan sifatnya yang terdesentralisasi, dapat memotivasi beberapa penambang untuk menghindari protokol Bitcoin dan menambang dengan egois atas biaya penambang jujur (Eyal and Sirer 2014). Hal ini menunjukkan bahwa masalah kepercayaan juga dapat berkembang dalam kategori *stakeholders* yang sama

4. *Goverment Trust* Terhadap Bitcoin

Bitcoin adalah protokol yang dipromosikan sebagai institusi peer-to-peer pertama, menawarkan alternatif bagi bank sentral (Abramowitz 2014). Telah diperdebatkan bahwa permintaan untuk transaksi peer-to-peer dapat menjadi indikasi untuk pengembangan kepercayaan pada Bitcoin (Bitcoin.org, 2014). Dalam konteks ini berguna untuk meninjau kembali komponen utama tata kelola peer-to-peer sebagai mekanisme kepercayaan institusional di Bitcoin: arbitrase, kepercayaan, bank, asosiasi bisnis dan hukum publik. Misalnya, protokol peer-to-peer dapat menawarkan struktur melalui seperangkat aturan untuk mengendalikan teknologi Bitcoin. Protokol antariksa juga dapat digunakan oleh pemerintah untuk mengembangkan kerangka hukum terstruktur untuk teknologi Bitcoin. Dalam pengambilan keputusan peer-to-peer, arbitrase adalah salah satu cara untuk menyelesaikan perselisihan (Thornburg, 2012). Jika arbitrase peer-to-peer mampu melayani keputusan,

maka hal itu juga bisa menjadi dasar kepercayaan peer-to-peer. Akan bermanfaat bagi wali amanat untuk dapat menginvestasikan deposit bitcoin untuk menumbuhkan kepercayaan korpus mereka (Abramowitz, 2014). Namun tantangan dalam mata uang kripto adalah tidak ada mekanisme yang memungkinkan akun semacam itu memiliki aset virtual. Untuk memiliki aset, ada kebutuhan hubungan perantara antara dunia maya dan dunia nyata. Memang bank *cryptocurrency* mungkin bisa menjalin hubungan ini. Jika bank peer-to-peer dapat menerima dana bank, membuat keputusan investasi, dan menyetujui pengeluaran, maka pengambilan keputusan *peer-to-peer* dapat digunakan untuk mengoperasikan asosiasi bisnis *peer-to-peer* (Abramowitz, 2014). Hambatan yang signifikan terhadap institusi *peer-to-peer* swasta, adalah permusuhan pemerintah (Abramowitz, 2014). Meski kurang percaya, sistem peer-to-peer belum menghasilkan keputusan dengan tingkat konsensus yang tinggi. Bentuk pengambilan keputusan terbatas yang melekat pada teknologi Bitcoin dapat menjadi landasan bagi jenis mekanisme pengambilan keputusan yang lebih canggih, yang memungkinkan institusi hukum dibuat tanpa penetapan otoritas pusat.

Kerangka ini didasarkan pada perspektif teoritis kepercayaan anak dan diskriminasi antara kepercayaan teknologi, sosial dan kelembagaan, dipetakan terhadap empat pemangku kepentingan Bitcoin yang diidentifikasi: pengguna, penambang, bursa dan pedagang. Kerangka ini digunakan sebagai lensa untuk mengidentifikasi masalah kepercayaan Bitcoin, dan menunjukkan bahwa keduanya meresap, tersebunyi dan saling bergantung.

2.2. *Cryptocurrency*

Menurut Baumann & Lesoismier (2017) *Cryptocurrency* adalah mata uang digital atau virtual yang menggunakan kriptografi untuk keamanan. Ciri yang menentukan tentang *cryptocurrency*, dan bisa dibilang merupakan daya tarik yang paling menawan, adalah sifat organiknya; itu tidak dikeluarkan oleh otoritas pusat manapun, membuatnya secara teoritis kebal terhadap campur tangan atau manipulasi pemerintah.

Definisi *cryptocurrency* yang lebih rinci dapat dikelompokkan menjadi empat kategori utama.

1. *Currency Tokens* : J coin adalah salah satu contoh. Sebuah token dipatok secara resmi pada Yen Jepang yang saat ini sedang dikembangkan.
2. *Utility Tokens* : Ethereum adalah sebuah contoh dari token utilitas. Ethereum adalah platform terdesentralisasi yang menjalankan *smart contracts*.
3. *Membership Tokens* : Storj adalah contoh. Ini adalah platform *cloud storage* yang terdesentralisasi yang memungkinkan seseorang menyewa ruang *hard drive* mereka yang menganggur dan mendapatkan penghasilan dengan melakukannya. Pengguna akhir dapat menggunakan Storj untuk menyimpan file mereka dengan harga yang kompetitif dan dalam jaringan P2P yang aman dari *downtime* server, penyensoran dan *hacks*.
4. *Security Tokens* : TaaS adalah contoh dari *security tokens*. TaaS dirancang untuk berinvestasi di pasar-pasar *blockchain*.

Bitcoin dan '*altcoins*.' Konvensi penamaan ini dimulai karena, pada tahun 2011, *forks* dari bitcoin muncul di pasar. *Forks*, atau klon, masing-masing bercita-cita untuk melayani area tertentu, yang bertujuan untuk 'lebih baik' daripada bitcoin.

Sejak saat itu, tak terhitung *cryptocurrency* yang muncul, dengan cara, mengikis dominasi pasar kripto bitcoin. Sepanjang tahun 2016 dan 2017, ICO dari segala bentuk dan ukuran telah berulang kali membuat catatan penggalangan dana baru. Bitcoin dan Ethereum tanpa diragukan lagi, membuka jalan bagi inisiasi inovatif generasi baru namun keberhasilan ICO telah membuktikan bahwa pasar ekonomi sangat antusias menunggu bentuk baru pendanaan.

2.3. Hukum *Cryptocurrency* di Indonesia

Dalam kutipan artikel yang ditulis oleh Hidayat (2018) mengenai bitcoin menurut Islam. KH Cholil Nafis menukil definisi uang dari kitab Buhuts fi al-Iqtishad al-Islami.


 حال أي وعلى الا وس يط ذلك كان مهما عاما بولا ينقى لـ تبادل و س يتكل هو الـ نقد
 "بـ كون

"Uang: segala sesuatu yang menjadi media pertukaran dan diterima secara umum, apa pun bentuk dan dalam kondisi seperti apa pun."

Kemudian mantan aktivis Pergerakan Mahasiswa Islam Indonesia (PMII) itu mengemukakan kutipan fatwa Dewan Syariah Nasional yang membolehkan transaksi jual beli mata uang dengan ketentuan tidak untuk spekulasi, ada kebutuhan, apabila transaksi dilakukan pada mata uang sejenis, nilainya harus sama dan tunai (attaqabudh). Jika berlainan jenis, harus dengan kurs yang berlaku saat transaksi dan tunai.

Dengan demikian Kedudukan Bitcoin sebagai alat tukar menurut hukum Islam adalah boleh, tetapi dengan syarat harus ada serah terima (taqabudh) dan sama kuantitasnya jika jenisnya sama. Jika jenisnya berbeda, disyaratkan harus

taqabudh secara hakiki atau hukmi; ada uang, ada Bitcoin yang bisa diserahterimakan.

Sedangkan menurut Bambang Pratama (2017) Bitcoin secara objek hukum sebenarnya dapat dikatakan sebagai benda (property) yang diperjualbelikan dan dinilai (ditaksir) jumlah uangnya. Pada saat pemilik Bitcoin melakukan transaksi, maka sistem transaksinya menjadi mirip barter antara pemilik Bitcoin dengan penjual. Karena tidak semua orang atau penjual mau menerima Bitcoin sebagai alat pembayaran. Hal ini yang membedakan Bitcoin dengan mata uang biasa, yang mana jika suatu transaksi dilakukan pembayaran menggunakan uang biasa, maka penerima pembayaran tidak boleh menolak uang yang dibayarkan. Sedangkan pada transaksi menggunakan Bitcoin, penerima pembayaran memiliki hak untuk menolak dibayarkan dengan Bitcoin. Hingga saat ini mulai banyak pedagang online atau *merchant* yang menerima transaksi Bitcoin. Artinya, ruang lingkup transaksi menggunakan Bitcoin mulai terbuka secara luas. Bahkan di beberapa wilayah di dunia dapat ditemui mesin ATM Bitcoin untuk menukar Bitcoin dengan uang konvensional. Meski demikian, transaksi jual beli menggunakan Bitcoin secara normatif di Indonesia masih berada di wilayah abu-abu.

Aspek hukum lainnya, terkait penyelenggara sistem Bitcoin atau uang virtual adalah penyelenggaranya yang tidak tuntuk pada peraturan terkait pelayanan jasa keuangan. Pada pelayanan jasa keuangan, produk jasa yang ditawarkan pasti diawasi oleh lembaga negara dan tingkat keamanannya dijaga dan diawasi, sehingga tingkat risiko bagi user menjadi rendah. Pada Bitcoin tingkat risiko yang ada di dalamnya tidak bisa dikatakan rendah karena tidak diawasi. Oleh sebab itu, maka bentuk risikonya menjadi spekulasi. Selain itu terjadi juga risiko bagi user

jika penyelenggara Bitcoin yang diretas pada kasus Mt. Gox di Jepang, yang mana uang user tidak dikembalikan. Meski demikian, ada juga beberapa cerita tentang keuntungan yang didapat dengan memperjualbelikan Bitcoin. Hal penting yang perlu diperhatikan adalah risiko yang terjadi pada jual-beli Bitcoin mendadi tanggungan pribadi.

2.4. Bitcoin

Bitcoin sendiri mulai diperkenalkan kepada masyarakat luas pada tanggal 3 Januari 2009 oleh Satoshi Nakamoto. Sejak saat itu bitcoin dikembangkan dengan berbagai perbaikan dan pengembangan berdasarkan saran dari para anggota *Cyberpunk*. Pengembangan itu antaralain adalah *hardfork* dari bitcoin itu sendiri. Sejak saat itu 8 tahun berlalu kini bitcoin menjadi *cryptocurrency* dengan valuasi pasar terbesar di dunia. Salah satu *cryptocurrency* lain yang berhasil mencapai kesuksesan besar adalah ethereum.

Menurut Andreas (2017) Bitcoin adalah kumpulan konsep dan teknologi yang membentuk basis ekosistem uang digital. Unit mata uang yang disebut bitcoin digunakan untuk menyimpan dan mentransmisikan nilai di antara peserta di jaringan bitcoin. Pengguna Bitcoin berkomunikasi satu sama lain menggunakan protokol bitcoin terutama melalui internet, walaupun jaringan transportasi lainnya juga bisa digunakan. Protokol bitcoin *stack*, tersedia sebagai perangkat lunak open source, dapat dijalankan di berbagai perangkat komputasi, termasuk laptop dan smartphone, membuat teknologi mudah diakses.

Bitcoin adalah sistem *peer-to-peer* terdistribusi. Dengan demikian tidak ada server "sentral" atau titik kontrol. Bitcoin diciptakan melalui proses yang disebut "*mining*", yang melibatkan persaingan untuk menemukan solusi terhadap masalah

matematika saat memproses transaksi bitcoin. Peserta di jaringan bitcoin yaitu, siapa saja yang menggunakan perangkat.

2.5. Ethereum

Etherum dirancang oleh Vitalik Buterin pada 30 Juli 2015 yang merupakan *cryptocurrency* terbesar kedua setelah bitcoin. Token dalam sistem Ethereum disebut dengan istilah Ether (ETH) yang memiliki fitur dan cara kerja yang cukup berbeda dengan bitcoin yang dimana dalam ETH dikenal dengan istilah *smart contract* yang berguna membawa *blockchain scripting* ke level yang lebih tinggi dari bitcoin. Fungsionalitasnya yang luas membuat *cryptocurrency* ini dijuluki sebagai bitcoin versi 2.0 atau “*The Next Bitcoin*”. Ada beberapa cara untuk mendapatkan ethereum salah satunya yaitu dengan melakukan penambangan (*mining*) yang dilakukan dengan menggunakan CPU atau GPU (*Graphic Processing Unit* atau kartu grafis atau VGA).

Menurut Dannen (2017) Ethereum dibangun dengan anggapan bahwa copycats adalah kepastian, dan mungkin ada banyak hambatan, dan karenanya harus ada seperangkat protokol yang bisa mereka gunakan untuk berkomunikasi. Dengan perspektif yang berbeda secara radikal dengan pencipta Bitcoin, pencipta Ethereum secara implisit mengambil posisi bahwa *cryptocurrency*, jika ada di masa depan, tidak akan menjadi sistem terdesentralisasi. Sebagai gantinya, jaringan terdistribusi dari sistem desentralisasi, memungkinkan banyak bukti kriptografi yang berbeda, dengan berbagai tujuan dan interpretasi agar mudah dan cepat didefinisikan dan kemudian dibawa ke kehidupan.

2.6. *Pool Mining*

Menurut Szmigielski (2016) *Pool Mining* mulai muncul di tahun 2010. Ketika penambang satu bersaing penambang lainnya sebagai penambang solo, proporsi *hashing power* jaringan mungkin kecil. Akibatnya, beberapa penambang mungkin hanya bisa memecahkan satu blok mungkin sebulan sekali atau bahkan setahun sekali (tergantung pada *hashing power*) yang bisa hanyalah berapa waktu yang diharapkan untuk memecahkan blok. Namun tidak bisa memprediksi kapan tepatnya akan terjadi.

Untuk memperlancar varian ini, para penambang mulai mengatur diri mereka sendiri ke dalam *pool*. Dalam *pool mining*, sejumlah besar penambang bekerja sama untuk memecahkan satu blok. Begitu seseorang di *pool* menemukan blok, imbalannya dibagi di antara semua penambang yang tergabung dalam *pool* tersebut. Hal ini membuat pembayaran lebih sering, meski kecil, dan lebih mudah ditebak.

2.7. *Solo Mining*

Menurut Szmigielski (2016) *Solo mining* adalah proses penambangan dimana seorang penambang melakukan semua perhitungan dengan dirinya sendiri. Setiap blok yang ditemukan dikreditkan ke penambang dan semua *reward* disimpan oleh penambang. Dalam *solo mining*, penambang bersaing melawan semua penambang, *pool*, dan individu lainnya untuk memecahkan satu blok.

Solo mining dapat dianggap sebagai penambangan di *pool* di mana Anda adalah satu-satunya pekerja. Alih-alih terhubung ke kolam, Anda akan terhubung ke dompet Anda dan dompet akan bertindak sebagai kolam renang. Dompet tersebut memberi perangkat lunak pertambangan semua rincian yang diperlukan

dalam bentuk blok kandidat untuk memulai penambangan. Oleh karena itu, menjalankan dompet yang dikenal sebagai *full node* diperlukan untuk *solo mining*.

2.8. *Directed Acyclic Graph* (DAG)

Dalam matematika dan ilmu komputer, *Directed Acyclic Graph* (DAG), adalah grafik berurutan terbatas tanpa siklus terarah. Artinya, ia terdiri dari banyak simpul dan tepi, dengan masing-masing ujung diarahkan dari satu titik ke titik yang lain, sehingga tidak ada jalan untuk memulai pada titik v dan v mengikuti urutan tepi yang secara konsisten diarahkan yang akhirnya kembali ke v lagi. . Secara ekuivalen, DAG adalah grafik berarah yang memiliki urutan topologi, urutan titik - titik sehingga setiap tepi diarahkan dari yang sebelumnya ke urutan berikutnya.

Menurut Dannen (2017) DAG merupakan dataset berukuran kurang lebih 1GB yang memerlukan waktu beberapa jam dalam pembuatannya. DAG tersebut akan dibuat ulang setiap 30.000 *block* atau sekitar 100 jam (disebut dengan *epoch*). Karena pembuatan DAG hanya bergantung pada nomor block maka DAG dapat dibuat sebelumnya seperti aplikasi *mining* akan membuat 2 buah DAG agar saat jangka waktu *epoch* berakhir, proses penambangan tetap berjalan dengan lancar.

2.9. *Alternative Coin*

Menurut Afilipoiae & Shortis (2015) Ada ratusan alternatif untuk Bitcoin dan Altcoins ini tidak semuanya ditujukan untuk anonimitas. Namun, karena perhatian dari penegak hukum serta penelitian yang telah menunjukkan kelemahan pada sifat anonimisasi Bitcoin, ditambah upaya untuk mengatur penggunaannya, ada minat yang meningkat dalam menciptakan *Altcoin* yang memberikan keamanan lebih bagi pengguna yang mungkin mempekerjakan mereka. untuk tujuan terlarang.

Ketika datang untuk menciptakan *Altcoin* anonim, faktor kunci dalam pengembangan mencakup penghapusan catatan publik seperti blockchain serta penyisipan mekanisme anonim lainnya ke dalam proses mata uang. Dengan demikian, banyak *Altcoin* sudah berteknologi lebih unggul dari sistem pencampuran terpusat Bitcoin seperti Monero, Zcash, dan sebagainya.

2.10. *Hashrate*

Menurut Wiley (2016) *Hash Rate* adalah alat ukur dari *processing power* yang dimiliki oleh jaringan Bitcoin dan altcoin lainnya. Jaringan Bitcoin selalu menjalankan perhitungan matematika untuk menjalankan dan mengamankan setiap transaksi yang terjadi dan tingkat daya yang dikerahkannya dapat dilihat dari *Hash Rate*-nya. Ketika jaringan mencapai *Hash Rate* sebesar 10 Th/s, maka jaringan tersebut bisa melakukan 10 trilyun kalkulasi berbeda per detiknya untuk memproses transaksi.

2.11. GPU Mining

Menurut Szmigielski (2016) GPU *Mining* merupakan langkah kedua dalam evolusi pertambangan (pertama ada CPU *Mining*). Dalam proses menyiapkan perangkat lunak, akan ada trik kecil yang berlaku untuk perangkat keras. Juga bagaimana GPU membandingkan perangkat keras yang lebih canggih, seperti FPGAs dan ASICs. GPU *mining* sendiri cukup berguna dalam hal pertambangan altcoin.

Pertambangan altcoin bisa di paralelkan, sehingga penambang juga bisa mencoba melakukan hashing pada waktu yang sama dengan *nonce* yang berbeda. Ketika *OpenCL* mulai dirilis di tahun 2010, memungkinkan untuk menjalankan

berbagai jenis perhitungan yang lebih cepat pada sebuah kartu grafis. Karena hal ini jugalah yang pada akhirnya membuka peluang untuk bisa melakukan pertambangan altcoin melalui GPU.

2.12. *Blocks*

Menurut Dannen (2017) Blok adalah satuan waktu yang mencakup sejumlah transaksi tertentu, sama seperti detak jantung adalah periode waktu di mana sejumlah darah dipindahkan melalui tubuh binatang. Di dalam periode tersebut, data transaksi dicatat; Ketika unit waktu berlalu, blok berikutnya dimulai.

Blockchain mewakili sejarah perubahan negara dalam database jaringan EVM.

Mengutip dokumen Ethereal:

Blok pada blockchain mewakili satuan waktu; blockchain itu sendiri adalah dimensi temporal dan mewakili keseluruhan sejarah negara pada titik waktu diskrit yang ditunjuk oleh blok pada rantai.

Di Bitcoin, blok adalah 10 menit. Waktu blok yang disebut ini berasal dari konstanta yang dikodekan dengan keras menjadi skema penerbitan Bitcoin, dengan total 21 juta koin akan dilepaskan dari tahun 2009 sampai 2024, dan ganjaran berkurang setengahnya setiap empat tahun.

Sedangkan pada Ethereum, waktu blok bukanlah fungsi dari jadwal penerbitan eter. *Block time* adalah variabel yang dijaga serendah mungkin, demi konfirmasi transaksi yang cepat yaitu rata-rata sekitar 15 detik . Waktu blok yang lebih pendek dari Ethereum adalah penerima manfaat penelitian blockchain yang dilakukan setelah peluncuran Bitcoin, yang menunjukkan bahwa waktu blok yang lebih pendek tidak hanya layak secara teknis, namun diinginkan dalam banyak hal.

2.13. *Blockchain*

Menurut Andreas (2017) Struktur data *blockchain* adalah daftar blok transaksi yang tertata dengan urutan belakang. *Blockchain* dapat disimpan sebagai flat file, atau dalam database sederhana. Klien Bitcoin *Core* menyimpan metadata blockchain menggunakan basis data Google *LevelDB*. Blok dihubungkan "belakang", masing-masing mengacu pada blok sebelumnya dalam rantai. *Blockchain* sering divisualisasikan sebagai tumpukan vertikal, dengan blok berlapis di atas satu sama lain dan blok pertama berfungsi sebagai pondasi tumpukan. Visualisasi blok yang ditumpuk di atas satu sama lain menghasilkan penggunaan istilah seperti "tinggi" untuk merujuk pada jarak dari blok pertama, dan "puncak" atau "tip" untuk merujuk ke blok yang paling baru ditambahkan.

Setiap blok dalam *blockchain* diidentifikasi oleh hash, yang dihasilkan dengan menggunakan algoritma hash kriptografi SHA256 (untuk Bitcoin) pada header blok. Setiap blok juga merujuk pada blok sebelumnya, yang dikenal sebagai blok induk, melalui bidang "hash blok sebelumnya" di *header* blokir. Dengan kata lain, setiap blok berisi hash induknya di dalam headernya sendiri. Urutan hash yang menghubungkan setiap blok ke induknya menciptakan rantai yang akan kembali ke blok pertama yang pernah dibuat, yang dikenal sebagai blok *genesis*.

2.14. *Mining*

Menurut Dannen (2017) *mining* dalam dunia *cryptocurrency* adalah proses dimana jaringan Ethereum mencapai konsensus mengenai urutan transaksi dalam jangka waktu tertentu, yang pada gilirannya memungkinkan EVM untuk membuat transisi status yang benar.

Sedangkan menurut Andreas (2017) *mining* adalah mekanisme yang mendasari *clearing house* terdesentralisasi, dimana transaksi telah divalidasi dan dihapus. *Mining* adalah penemuan yang membuat bitcoin istimewa, mekanisme keamanan terdesentralisasi yang menjadi dasar bagi P2P *digital cash*.

Pertambangan mengamankan sistem bitcoin dan memungkinkan munculnya konsensus di seluruh jaringan tanpa otoritas pusat. Imbalan koin dan biaya transaksi yang baru dicetak merupakan skema insentif yang menyelaraskan tindakan penambang dengan keamanan jaringan, sekaligus menerapkan pasokan moneter.

2.15. Website

Menurut Rahmat Hidayat (2010), website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman *web* dengan *web* halaman lainnya *Hyperlink*, sedangkan teks yang dijadikan media terhubung adalah *Hypertext*.

Untuk membangun sebuah halaman website gratis harus ada unsur pendukung sebagai berikut, yaitu :

- Nama *Domain/URL (Uniform Resources Locator)*
- *Web Hosting*
- *Content Management System (CMS)*

Adapun jenis-jenis *website* berdasarkan sifatnya yaitu:

- *Website Statis*, merupakan website yang kontennya sangat jarang diubah dengan menggunakan bahasa pemrograman HTML dan belum menggunakan database sebagai penyimpanan data.
- *Website Dinamis*, merupakan website yang kontennya bisa diubah-ubah setiap saat. Berbeda dengan website statis, website dinamis menggunakan bahasa pemrograman *PHP*, *ASP*, *.NET* dan menggunakan database sebagai penyimpanan data seperti *MySQL*.

2.16. *PHP*

Menurut (Rudianto, 2011), “*PHP* adalah Bahasa *server-side-scripting* yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis. Karena *PHP* merupakan *server-side-scripting* maka sintak dan perintah-perintah *PHP* akan dieksekusi di server kemudian hasilnya akan dikirimkan ke browser dengan format *HTML*”. Menurut (Nugroho, 2006), “*PHP* atau singkatan dari *Personal Home Page* merupakan bahasa skrip yang tertanam dalam *HTML* untuk dieksekusi bersifat server side”. *PHP* termasuk dalam *open source product*, sehingga *sourcecode PHP* dapat diubah dan didistribusikan secara bebas.

2.17. *MySql*

Menurut (Arief, 2011) MySQL (*My Structure Query Language*) adalah “salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya”. *Mysql* bersifat open source dan menggunakan *SQL*

(*Structured Query Languange*). MySQL biasa dijalankan diberbagai platform misalnya windows Linux, dan lain sebagainya.

MySQL merupakan DBMS yang multithread, multi user yang bersifat gratis di bawah lisensi GNU *General Public Licence* (GPL). Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. Seperti yang telah disebutkan sebelumnya, MySQL bersifat gratis atau open source sehingga kita bisa menggunakananya secara gratis.

2.18. Lean Startup

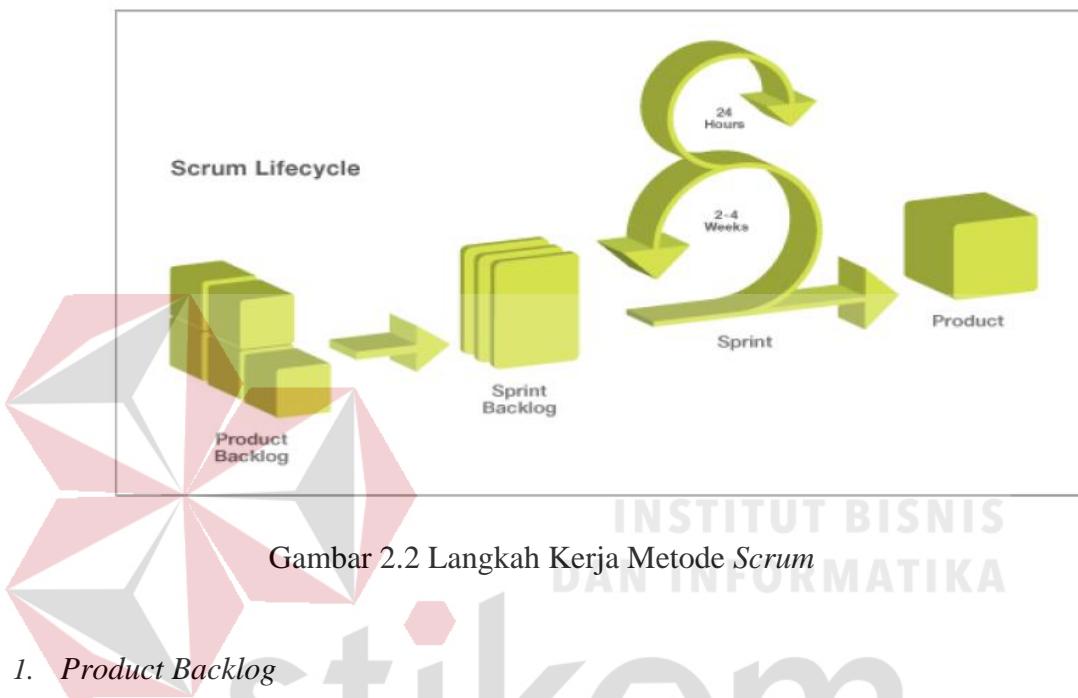
Menurut Ries (2011) Lean *Startup* mengambil nama dari revolusi lean *manufacturing Taiichi* dan *Shigeo Shingoyang* di kembangkan Toyota. Lean thinking secara radikal mengubah *system supply chain* dan sistem produksi. Dalam hubungan itu dapat digambarkan antara pengetahuan dan kreatifitas pekerja untuk menyederhanakan *batch, just in time production, inventory* dan meningkatkan *cycle time*. Lean *startup* akan mengajarkan perbedaan tentang aktivitas memberikan nilai tambah atau “sampah”, dan menunjukkan bagaimana menciptakan sebuah produk. Tentu lean startup lebih mengambil ilmu dan materi *entrepreneurship*-nya dibandingkan dengan system *manufacturing*-nya. Lean *startup* menggunakan unit kontrol *progress* yang dikenal dengan nama *validated learning*.

2.19. Scrum

Scrum adalah suatu metodologi atau kerangka kerja yang terstruktur untuk pengembangan produk yang kompleks Partogi (2015). *Scrum* terdiri dari sebuah tim yang memiliki peran dan tugas masing-masing. Setiap komponen dalam

kerangka melayani tujuan tertentu dan sangat penting untuk kesuksesan penggunaan *scrum*.

Langkah-langkah aktifitas dengan menggunakan metodologi *Scrum* adalah sebagai berikut :



1. *Product Backlog*

Bagian pertama yang perlu dilakukan adalah membuat kumpulan hal-hal yang diperlukan dan harus tersedia dari produk atau dalam hal ini adalah sistem yang akan dibangun.

2. *Sprint Backlog*

Langkah ini adalah membuat perencanaan dengan dilakukan pertemuan antara *developer* dan *user*, yang akan berkolaborasi untuk memilih *product backlog* untuk dimasukkan kedalam proses *Sprint*. Hasil pertemuan tersebut di sebut *Sprint Backlog*.

3. *Sprint*

Dalam *Scrum*, *Sprint* adalah sebuah kerangka waktu yang berdurasi maksimal

1 bulan untuk mengembangkan produk yang berpotensi untuk dirilis. Dalam *Sprint* terdapat 2 pekerjaan, yaitu :

a. *Pertemuan Harian* (Daily Standup Meeting)

Pertemuan dimana setiap 24 jam (1 hari), tim pengembang bertemu untuk membahas perkembangan produk. Hal ini kami lakukan tidak secara langsung, tetapi via dunia maya melalui *Skype* ataupun *Whatsapp*.

b. *Refleksi Sprint*

Merupakan pertemuan yang dilakukan setiap bulannya yang bertujuan untuk membahas hal dari *Sprint Backlog* yang telah berjalan dan telah berhasil dikerjakan, serta dapat memperbaiki dan meningkatkan kualitas produk *Sprint* yang berikutnya.

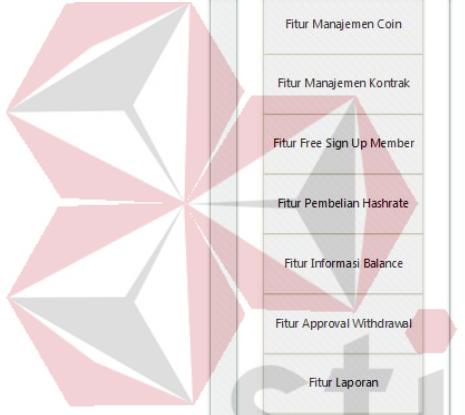
4. *Increment*

Increment merupakan hasil dari seluruh hal dalam *Product Backlog* yang telah selesai dikerjakan pada seluruh *Sprint*. Pada akhir *Sprint*, *Increment* harus sudah benar-benar selesai, yang berarti harus dalam keadaan yang *useable*. *Scrum* sendiri memiliki prinsip :

- a. Ukuran tim yang kecil melancarkan komunikasi, mengurangi biaya, dan memberdayakan satu sama lain.
- b. Proses dapat beradaptasi terhadap perubahan dan bisnis.
- c. Proses menghasilkan beberapa *software increment*.
- d. Pembangunan dan orang yang membangun dibagi dalam tim yang kecil.
- e. Dokumentasi dan pengujian terus menerus dilakukan setelah *software* dibangun.
- f. Proses *scrum* mampu menyatakan bahwa produk selesai kapanpun.

2.20. Daily Scrum

Tahap ini adalah tahap untuk mengetahui *progress* dari melakukan *coding* sesuai *sprint backlog* yang sudah ditentukan sebelumnya. Tahap ini melaporkan kemajuan dari setiap *sprint backlog* yang sudah dikerjakan. *Daily scrum* akan dilakukan setiap 24 jam sekali untuk mengetahui kemajuan setelah melakukan *coding*. Bentuk *daily scrum* dapat dilihat pada gambar dibawah.



| Pengembangan Startup ALT.Mine.ID Untuk Penambangan Cryptocurrency Bagi Investor di Indonesia | | |
|---|-------------|------|
| To Do | In Progress | Done |
| <ul style="list-style-type: none"> Fitur Manajemen Coin Fitur Manajemen Kontrak Fitur Free Sign Up Member Fitur Pembelian Hashrate Fitur Informasi Balance Fitur Approval Withdrawal Fitur Laporan | | |

INSTITUT BISNIS
DAN INFORMATIKA
stikom
SURABAYA

Gambar 2.3 *Daily Scrum*

Apabila butir penggerjaan yang ada di *to do* sedang dikerjakan, maka butir penggerjaan yang ada pada *to do* harus dipindahkan ke dalam kolom *in progress* untuk mengetahui *progress* setiap harinya.

| Pengembangan Startup ALT.Mine.ID Untuk Penambangan Cryptocurrency Bagi Investor di Indonesia | | |
|--|---|------|
| To Do | In Progress | Done |
| Fitur Free Sign Up Member Fitur Pembelian Hashrate Fitur Informasi Balance Fitur Approval Withdrawal Fitur Laporan | Fitur Manajemen Coin Fitur Manajemen Kontrak | |

Gambar 2.4 Daily Scrum (*In Progress*)

Apabila butir pengerjaan yang ada di *in progress* selesai dikerjakan, maka butir pengerjaan yang ada pada *in progress* harus dipindahkan ke dalam kolom *done* untuk mengetahui *progress* yang telah selesai.

| Pengembangan Startup ALT.Mine.ID Untuk Penambangan Cryptocurrency Bagi Investor di Indonesia | | |
|---|---------------------------|---|
| To Do | In Progress | Done |
| Fitur Pembelian Hashrate Fitur Informasi Balance Fitur Approval Withdrawal Fitur Laporan | Fitur Free Sign Up Member | Fitur Manajemen Coin Fitur Manajemen Kontrak |

Gambar 2.5 Daily Scrum (*Done*)

2.21. *Scrum Team*

Scrum Master, Product Owner, Development Team adalah manajer dalam *Scrum* yang mengelola ruang lingkupnya sendiri – sendiri. Mereka bertiga secara kesatuan disebut sebagai *Scrum Team*. *Scrum Team* bersifat swakelola dan lintas-fungsi. Tim yang swakelola memilih cara terbaik dalam mengerjakan pekerjaan mereka, bukan diperintah oleh orang lain di luar tim ini. Tim yang lintas-fungsi memiliki semua keahlian yang dibutuhkan untuk menyelesaikan pekerjaan mereka tanpa bergantung pada orang lain di luar tim ini. Bentuk tim dalam *Scrum* dirancang untuk mengoptimalkan fleksibilitas, kreativitas dan produktivitas.

Dalam *Scrum*, peran manajer didesentralisasikan oleh ketiga peran yang masing – masing memiliki akuntabilitas terhadap areanya masing – masing guna meningkatkan *sense of ownership* dalam pengembangan *software* (Partogi, 2015).

Ketiga peran manajerial ini adalah :

A. *Scrum Master*

Scrum Master adalah seorang *Servant Leader* (seorang pemimpin yang melayani dalam memimpin sebuah organisasi). Bagi seorang *Servant Leader*, segala kesuksesan di dalam perusahaan bukan milik dia, melainkan milik tim yang dia *coaching*. *Scrum Master* bertanggung jawab untuk mengedukasi orang – orang di dalam perusahaan dan meluruskan salah persepsi mengenai *Agility* dengan *Scrum*, dan akuntabel terhadap pendewasaan orang – orang di dalam perusahaan dalam memahami prinsip dan filosofi *Scrum*.

B. *Product Owner*

Product Owner adalah peran yang sangat menentukan berhasilnya pengembangan produk. *Product Owner* akuntabel terhadap produk yang ia

kelola. Ia memastikan produk yang dikembangkan oleh Tim Pengembang menghasilkan nilai tambah untuk pelanggan dan perusahaan di mana dia berada. *Product Owner* berperan dalam memaksimalkan nilai bisnis untuk perusahaan lewat *software* yang dikembangkan oleh Tim Pengembang. *Product Owner* bertanggung jawab dalam mengelola produk untuk meningkatkan nilai valuasi perusahaan di pasar.

C. *Development Team*

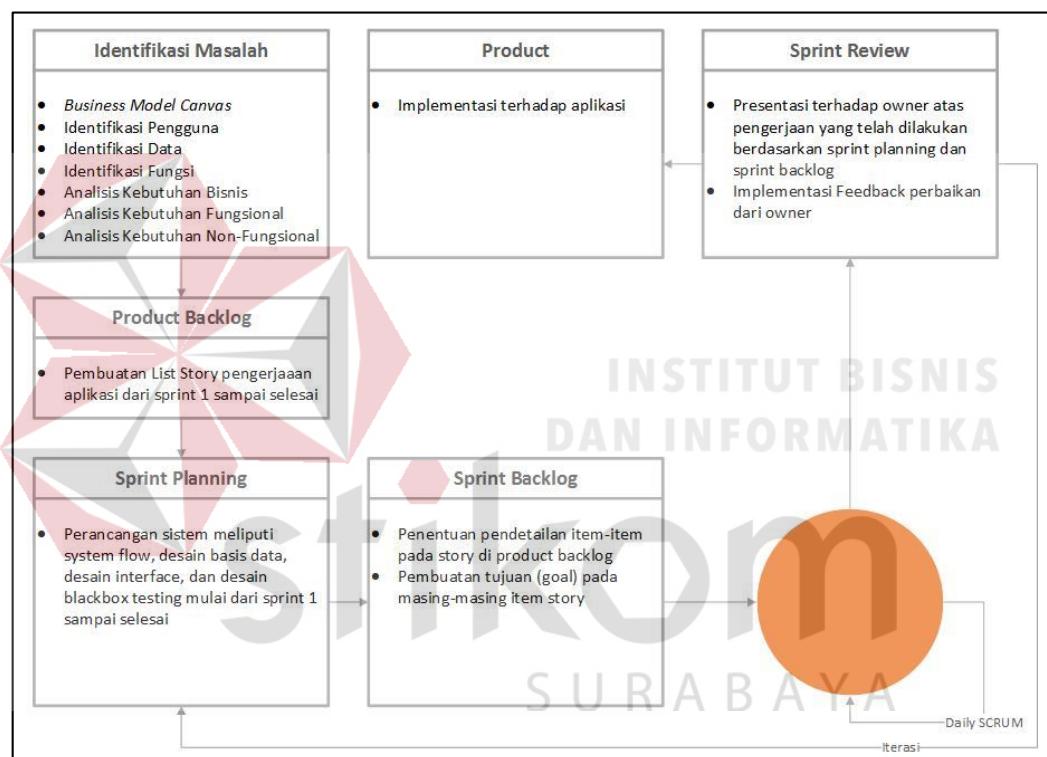
Development Team terdiri dari para ahli profesi yang bekerja untuk mengantarkan *Increment* “Selesai” yang berpotensi untuk dirilis di setiap akhir *Sprint*. *Increment* “Selesai” wajib tersedia pada saat *Sprint Review*. Hanya anggota dari *Development Team* yang membuat *Increment* ini.

D. *Development Team* dibentuk dan diberikan wewenang oleh organisasi untuk menyusun dan mengelola pekerjaan mereka sendiri. Hasil sinergi dari tim ini akan mengoptimalkan efisiensi dan efektivitas *Development Team* secara keseluruhan.

BAB III

ANALISIS DAN PERANCANGAN

Dalam melakukan analisis dan perancangan sistem adapun metode yang digunakan adalah SCRUM. Tahapan penelitian menggunakan penggeraan metode SCRUM dapat dilihat pada Gambar 3.1 di bawah ini.



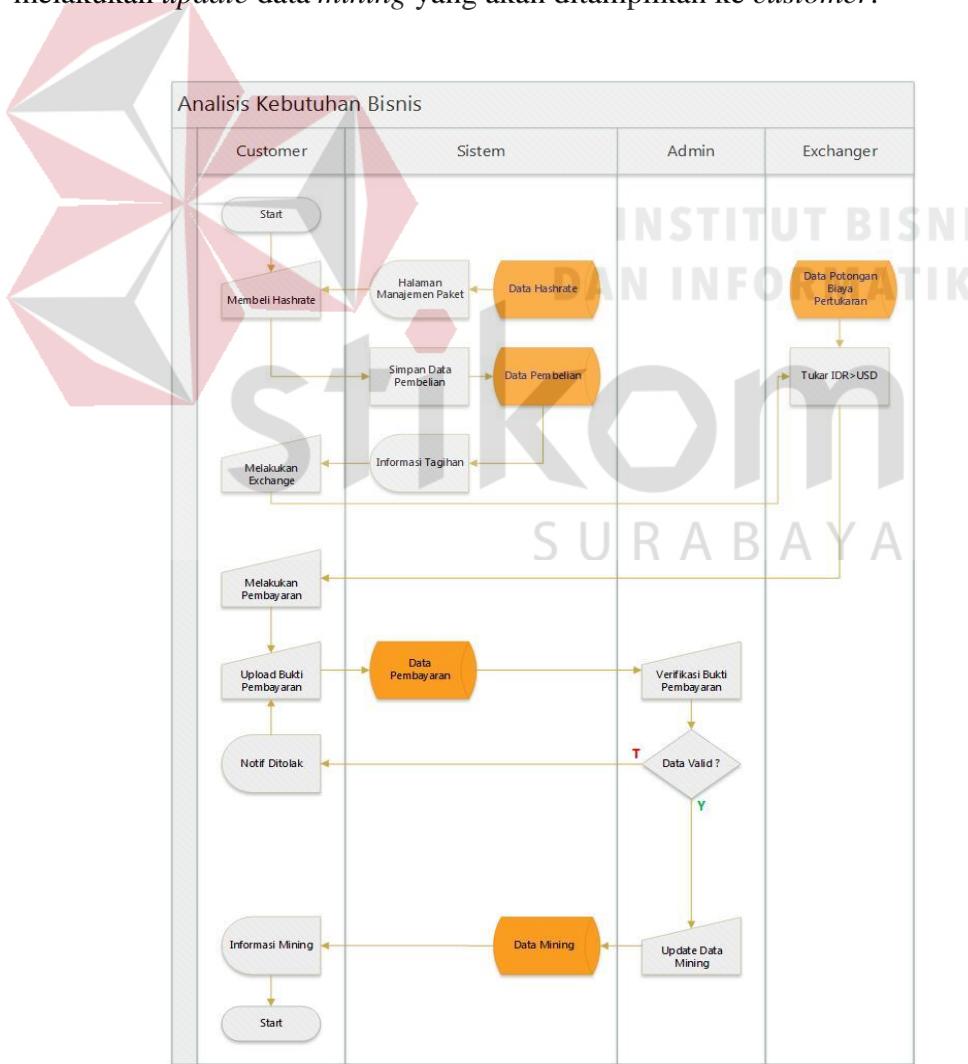
Gambar 3.1 Tahapan Penelitian Berdasarkan SCRUM

3.1. Mengidentifikasi Masalah

Saat ini *e-commerce* yang khusus menyediakan jasa investasi *cryptocurrency* berupa penyewaan alat tambang belum banyak ada di Indonesia. Hal ini membuat masyarakat yang ingin memulai investasi dalam dunia *cryptocurrency* tidak mempunyai pilihan lain selain menggunakan jasa *e-commerce*

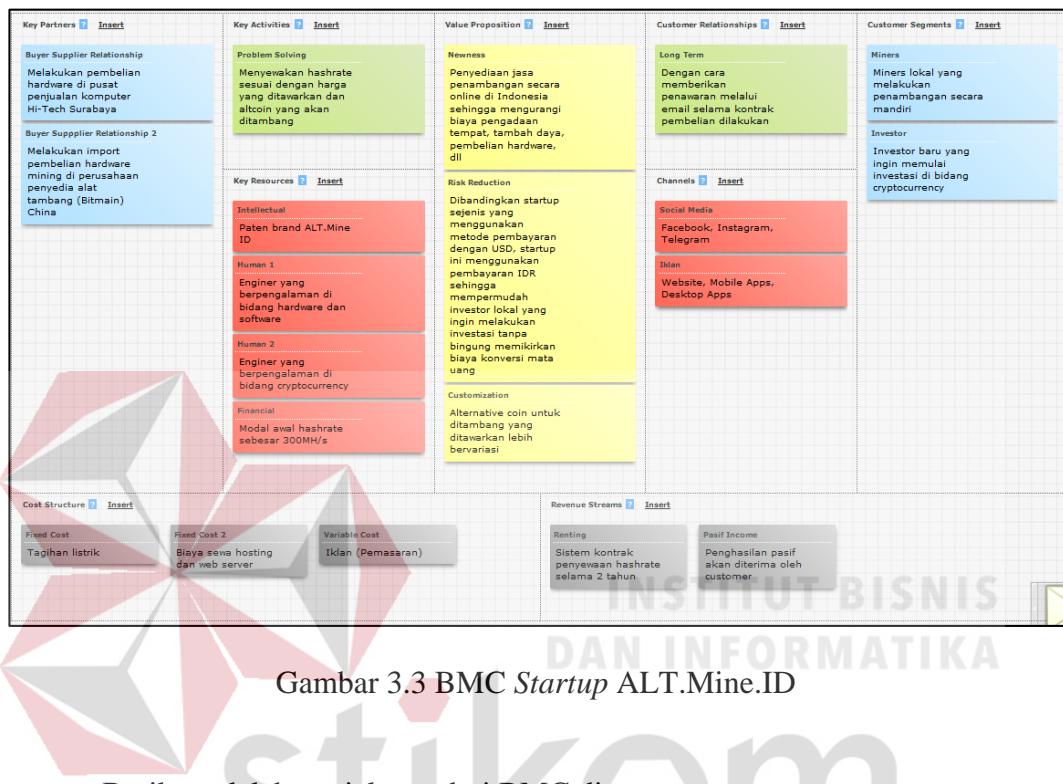
dari luar negeri. Hal ini sedikit banyak membebani para investor dalam memulai investasi karena sedikitnya pilihan yang bisa diambil untuk memulainya.

Penjelasan Gambar 3.2 di bawah adalah bagaimana proses mining saat ini dilakukan mulai dari *customer* melakukan pembelian *hashrate*, jika harga sesuai dengan keinginan *customer* maka *customer* akan melakukan pembelian *hashrate*. Selanjutnya *customer* melakukan pembayaran dengan melakukan penukaran ke *exchanger* lokal kemudian melakukan *upload* bukti bayar. *Admin* akan melakukan verifikasi apakah data *upload* pembayaran *customer* valid atau tidak kemudian melakukan *update* data *mining* yang akan ditampilkan ke *customer*.



Gambar 3.2 Alur Proses Bisnis Saat Ini

Setelah mengetahui proses bisnis yang ada saat ini kemudian dibuat *Business Model Canvas* (BMC) dari startup ALT.Mine.ID. Adapun BMC dari startup ini dapat dilihat pada gambar di bawah ini.



Gambar 3.3 BMC Startup ALT.Mine.ID

Berikut adalah penjelasan dari BMC di atas :

1. *Value Proposition*

- Newness* : Penyediaan jasa penambangan secara online di Indonesia sehingga mengurangi biaya pengadaan tempat, tambah daya, pembelian hardware.
- Risk Reduction* : Startup menggunakan metode pembayaran IDR
- Customization* : Alternative coin untuk ditambang yang ditawarkan lebih bervariasi

2. *Customer Segments*

- Miners* : Miners local yang melakukan penambangan secara mandiri

- b. *Investor* : *Investor* baru yang ingin memulai investasi di bidang *cryptocurrency*
- 3. *Customer Relationship*
 - a. *Long Term* : Memberikan penawaran melalui email selama kontrak pembelian dilakukan
- 4. *Channels*
 - a. *Social Media* : Facebook, Instagram, Telegram.
 - b. Iklan : *Website*, Mobile APP
- 5. *Key Activities*
 - a. *Problem Solving* : Menyewakan *hashrate* sesuai dengan harga yang diwarkan dan *altcoin* yang akan ditambang
- 6. *Key Resources*
 - a. *Intellectual* : Paten brand ALT.Mine.ID
 - b. *Human 1* : *Engineer* yang berpengalaman di bidang *hardware* dan *software*
 - c. *Human 2* : *Engineer* yang berpengalaman di bidang *cryptocurrency*
 - d. *Financial* : Modal awal *hashrate* sebesar 320MH/s
- 7. *Key Partners*
 - a. *Buyer Supplier Relationship* : Melakukan pembelian *hardware* di pusat penjualan computer Hi-Tech Surabaya
 - b. *Buyer Supplier Relationship* : Melakukan *import* pembelian *hardware mining* di perusahaan penyedia alat tambang (Bitmain) China.
- 8. *Cost Structure*
 - a. *Fixed Cost* : Tagihan listrik

b. *Fixed Cost 2* : Biaya sewa *hosting* dan *web server*

c. *Variable Cost* : Iklan (Pemasaran)

9. *Revenue Stream*

a. *Renting* : Sistem kontrak penyewaan *hashrate* selama 2 tahun

b. *Pasif Income* : Penghasilan pasif akan diterima oleh pelanggan

Pada umumnya, situs yang menyediakan jasa persewaan alat tambang berasal dari luar negeri. Situs-situs ini mempunyai kekurangan pada bagian transaksi pembelian dimana investor lokal yang ingin membeli *hashrate* harus melakukan *convert* mata uang dari mata uang fiat menjadi *crypto* kemudian melakukan transaksi. Proses ini tentunya memerlukan biaya, hal ini tentunya menjadi bahan perhitungan para investor kecil yang ingin memulai investasinya di dunia. Adapun permasalahan dari proses yang dijelaskan sebelumnya dapat dilihat pada Tabel 3.1 di bawah.

Tabel 3.1 Identifikasi Masalah

| No | Masalah | Dampak |
|----|---------------------------------|---|
| 1 | Investasi sendiri | Modal yang dikeluarkan lebih besar |
| 2 | <i>Hardware</i> | Kesalahan <i>overclocking</i> dan perawatan |
| 3 | <i>Cloud mining</i> luar negeri | Pembayaran lebih susah dilakukan |
| 4 | Kontrak | Kontrak penambangan pada <i>cloud mining</i> luar negeri bisa dihentikan tanpa sebab apabila <i>mining</i> sudah dianggap tidak <i>profitable</i> |

Berdasarkan tabel permasalahan diaatas selanjutnya dibuatkan analisis kebutuhan fungsional. Kebutuhan fungsionalitas adalah suatu layanan sistem yang harus disediakan dan dapat berinteraksi dengan pengguna, seperti apa saja di dalam

sistem dan apa yang dapat dilakukan oleh pengguna. Aplikasi yang dirancang harus mampu memenuhi kebutuhan fungsionalitas sebagai berikut :

1. Fungsi penambahan *coin* baru, dapat dilihat pada tabel di bawah.

Tabel 3.2 Kebutuhan Fungsi Penambahan *Coin* Baru

| Nama Fungsi | Fungsi Penambahan <i>Coin</i> baru | |
|---------------------------|---|---|
| Stakeholder | <i>Admin</i> | |
| Deskripsi | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan penambahan data <i>coin</i> baru. | |
| Kondisi Awal | 1. <i>Admin</i> sudah <i>login</i> 2. <i>Interface</i> harus sudah muncul | |
| Alur Komputerisasi | Aksi Stakeholder | Respon Sistem |
| | | Menambah Data <i>Coin</i> |
| | a. Pengguna memilih menu manajemen koin. b. Pengguna memilih button tambah data c. Pengguna menginputkan data <i>coin</i> berupa nama <i>coin</i> , kode <i>coin</i> , stok <i>coin</i> , dan icon <i>coin</i> setelah itu menekan tombol tambah. | a. Sistem akan menampilkan halaman manajemen koin. b. Sistem akan menampilkan halaman tambah data <i>coin</i> . c. Sistem akan melakukan fungsi penambahan data dari <i>form</i> tersebut ke dalam tabel <i>coin</i> dan setelah itu menampilkan pesan ‘berhasil’ jika berhasil disimpan, atau pesan ‘gagal’ jika gagal dalam proses penyimpanan. |
| | | Mengubah Data <i>Coin</i> |
| | a. Pengguna menekan tombol <i>edit</i> pada <i>coin</i> yang akan <i>diupdate</i> . b. Pengguna memasukkan data perubahan ke dalam <i>form</i> dan setelah itu menekan tombol ubah. | a. Sistem akan menampilkan data <i>coin</i> sesuai yang dipilih ke dalam <i>form</i> pengubahan data. b. Sistem akan melakukan pengubahan data pada <i>coin</i> dan menampilkan pesan ‘berhasil’ atau ‘gagal’. |
| | Menghapus Data <i>Coin</i> | |

| | | |
|----------------------|--|---|
| | a. Pengguna menekan tombol hapus pada <i>coin</i> yang akan dihapus | a. Sistem akan melakukan hapus data pada tabel <i>coin</i> dan menampilkan pesan ‘berhasil’ atau ‘gagal’. |
| Kondisi Akhir | Fungsi ini dapat melakukan pengelolaan data <i>coin</i> dan tersimpan ke dalam <i>database</i> . | |

2. Fungsi *setting* paket *altcoin*, dapat dilihat pada tabel di bawah.

Tabel 3.3 Kebutuhan Fungsi *Setting* Paket *Altcoin*

| Nama Fungsi | Fungsi <i>Setting</i> Paket Altcoin | |
|---------------------------|---|---|
| Stakeholder | <i>Admin</i> | |
| Deskripsi | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan penambahan paket pada data <i>coin</i> yang sudah terdaftar. | |
| Kondisi Awal | 1. <i>Admin</i> sudah <i>login</i> 2. <i>Interface</i> harus sudah muncul 3. Data <i>coin</i> harus sudah ada | |
| Alur Komputerisasi | Aksi Stakeholder | Respon Sistem |
| | Menambah Paket Altcoin | |
| | a. Pengguna memilih menu manajemen paket. b. Pengguna memilih button tambah data c. Pengguna menginputkan data paket berupa, kode <i>coin</i> yang telah ditambahkan, <i>hashrate coin</i> , dan harga <i>coin</i> setelah itu menekan tombol tambah. | a. Sistem akan menampilkan halaman manajemen paket. b. Sistem akan menampilkan halaman tambah data paket. c. Sistem akan melakukan fungsi penambahan data dari <i>form</i> tersebut ke dalam tabel paket dan setelah itu menampilkan pesan ‘berhasil’ jika berhasil disimpan, atau pesan ‘gagal’ jika gagal dalam proses penyimpanan. |
| | Mengubah Data Paket | |

| | | |
|----------------------------|---|--|
| | <p>a. Pengguna menekan tombol edit pada paket <i>coin</i> yang akan <i>diupdate</i>.</p> <p>b. Pengguna memasukkan data perubahan ke dalam <i>form</i> dan setelah itu menekan tombol ubah.</p> | <p>a. Sistem akan menampilkan data paket sesuai yang dipilih ke dalam <i>form</i> pengubahan data.</p> <p>b. Sistem akan melakukan pengubahan data pada paket dan menampilkan pesan ‘berhasil’ atau ‘gagal’.</p> |
| Menghapus Data Coin | | |
| | <p>a. Pengguna menekan tombol hapus pada paket <i>coin</i> yang akan dihapus</p> | <p>a. Sistem akan melakukan hapus data pada tabel paket dan menampilkan pesan ‘berhasil’ atau ‘gagal’.</p> |
| Kondisi Akhir | Fungsi ini dapat melakukan pengelolaan data paket dan tersimpan ke dalam <i>database</i> . | |

3. Fungsi *setting* kontrak *altcoin*, dapat dilihat pada tabel di bawah.

Tabel 3.4 Kebutuhan Fungsi *Setting* Kontrak *Altcoin*

| Nama Fungsi Stakeholder | Fungsi <i>Setting</i> Kontrak <i>Altcoin</i> | |
|--------------------------------|---|--|
| Deskripsi | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan penambahan kontrak persetujuan antara penyedia jasa (ALT.Mine.ID) dan pengguna jasa (<i>Customer</i>). | |
| Kondisi Awal | <p>1. Admin sudah <i>login</i></p> <p>2. Interface harus sudah muncul</p> | |
| Alur Komputerisasi | Aksi Stakeholder | Respon Sistem |
| | Edit Kontrak | |
| | <p>a. Pengguna memilih menu manajemen kontrak.</p> <p>b. Pengguna memilih <i>button</i> edit kontrak</p> <p>c. Pengguna menginputkan data setelah itu menekan tombol <i>update</i>.</p> | <p>a. Sistem akan menampilkan halaman manajemen kontrak.</p> <p>b. Sistem akan menampilkan halaman <i>edit</i> kontrak.</p> <p>c. Sistem akan melakukan fungsi <i>update</i> data dari <i>form</i> tersebut ke dalam tabel kontrak dan setelah</p> |

| | | |
|----------------------|--|--|
| | | itu menampilkan pesan ‘berhasil’ jika berhasil disimpan, atau pesan ‘gagal’ jika gagal dalam proses penyimpanan. |
| Kondisi Akhir | Fungsi ini dapat melakukan pengelolaan data kontrak dan tersimpan ke dalam <i>database</i> . | |

4. Fungsi verifikasi, dapat dilihat pada tabel di bawah.

Tabel 3.5 Kebutuhan Fungsi Verifikasi

| Nama Fungsi | Fungsi Edit Wallet | | | | | | | | |
|--|---|-------------------------|----------------------|---|--|--|--|--|--|
| Stakeholder | <i>Customer</i> | | | | | | | | |
| Deskripsi | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan verifikasi <i>customer</i> yang telah melakukan pendaftaran. | | | | | | | | |
| Kondisi Awal | <p>1. <i>Customer</i> harus sudah terdaftar</p> <p>2. Customer harus sudah bisa <i>login</i></p> <p>3. <i>Interface</i> harus sudah muncul</p> | | | | | | | | |
| Alur Komputerisasi | <table border="1"> <thead> <tr> <th>Aksi Stakeholder</th> <th>Respon Sistem</th> </tr> </thead> <tbody> <tr> <td>a. Pengguna memilih menu <i>Account</i>.</td> <td>a. Sistem akan menampilkan halaman akun.</td> </tr> <tr> <td>b. Pengguna memilih <i>button</i> verifikasi</td> <td>b. Sistem akan menampilkan halaman verifikasi.</td> </tr> <tr> <td>c. Pengguna menginputkan data file foto KTP dan KYC setelah itu menekan tombol <i>update</i>.</td> <td>c. Sistem akan melakukan fungsi update data dari <i>form</i> tersebut ke dalam tabel <i>verif</i> dan setelah itu menampilkan pesan ‘berhasil’ jika berhasil disimpan, atau pesan ‘gagal’ jika gagal dalam proses penyimpanan.</td> </tr> </tbody> </table> | Aksi Stakeholder | Respon Sistem | a. Pengguna memilih menu <i>Account</i> . | a. Sistem akan menampilkan halaman akun. | b. Pengguna memilih <i>button</i> verifikasi | b. Sistem akan menampilkan halaman verifikasi. | c. Pengguna menginputkan data file foto KTP dan KYC setelah itu menekan tombol <i>update</i> . | c. Sistem akan melakukan fungsi update data dari <i>form</i> tersebut ke dalam tabel <i>verif</i> dan setelah itu menampilkan pesan ‘berhasil’ jika berhasil disimpan, atau pesan ‘gagal’ jika gagal dalam proses penyimpanan. |
| Aksi Stakeholder | Respon Sistem | | | | | | | | |
| a. Pengguna memilih menu <i>Account</i> . | a. Sistem akan menampilkan halaman akun. | | | | | | | | |
| b. Pengguna memilih <i>button</i> verifikasi | b. Sistem akan menampilkan halaman verifikasi. | | | | | | | | |
| c. Pengguna menginputkan data file foto KTP dan KYC setelah itu menekan tombol <i>update</i> . | c. Sistem akan melakukan fungsi update data dari <i>form</i> tersebut ke dalam tabel <i>verif</i> dan setelah itu menampilkan pesan ‘berhasil’ jika berhasil disimpan, atau pesan ‘gagal’ jika gagal dalam proses penyimpanan. | | | | | | | | |
| Kondisi Akhir | Fungsi ini dapat melakukan pengelolaan data verifikasi dan tersimpan ke dalam <i>database</i> . | | | | | | | | |

5. Fungsi *edit wallet*, dapat dilihat pada tabel di bawah.

Tabel 3.6 Kebutuhan Fungsi *Edit Wallet*

| Nama Fungsi | Fungsi <i>Edit Wallet</i> | |
|---------------------------|--|----------------------|
| Stakeholder | <i>Customer</i> | |
| Deskripsi | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan perubahan data wallet <i>altcoin</i> setelah <i>customer</i> melakukan pendaftaran. | |
| Kondisi Awal | 4. <i>Customer</i> harus sudah terdaftar 5. <i>Customer</i> harus sudah <i>login</i> 6. <i>Interface</i> harus sudah muncul | |
| Alur Komputerisasi | Aksi Stakeholder | Respon Sistem |
| | <i>Edit Wallet</i> <ul style="list-style-type: none"> a. Pengguna memilih menu <i>Account</i>. b. Pengguna memilih <i>button edit wallet</i> c. Pengguna menginputkan data wallet berupa, <i>ETH wallet</i>, <i>btc wallet</i>, dan <i>wallet altcoin</i> lain yg tersedia setelah itu menekan tombol <i>update</i>. | |
| Kondisi Akhir | Fungsi ini dapat melakukan pengelolaan data <i>wallet</i> dan tersimpan ke dalam <i>database</i> . | |

6. Fungsi *upload foto*, dapat dilihat pada tabel di bawah.

Tabel 3.7 Kebutuhan Fungsi *Upload Foto*

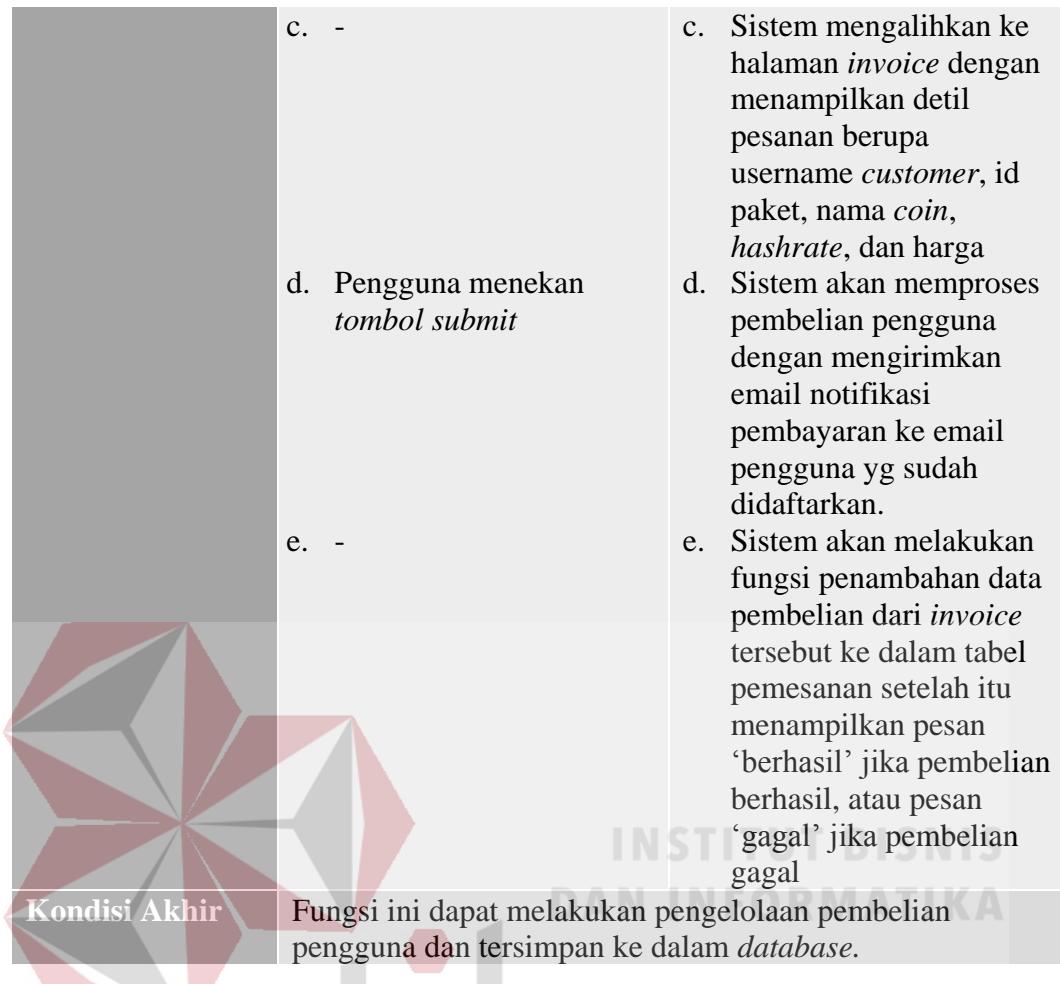
| Nama Fungsi | Fungsi <i>Upload Foto</i> |
|---------------------|---|
| Stakeholder | <i>Customer</i> |
| Deskripsi | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan perubahan foto setelah <i>customer</i> melakukan pendaftaran. |
| Kondisi Awal | 1. <i>Customer</i> harus sudah terdaftar 2. <i>Customer</i> harus sudah <i>login</i> 3. <i>Interface</i> harus sudah muncul |

| Alur Komputerisasi | Aksi Stakeholder | Respon Sistem |
|--------------------|---|--|
| | Upload Foto | |
| | <ul style="list-style-type: none"> a. Pengguna memilih menu <i>Account</i>. b. Pengguna memilih <i>button icon</i> foto c. Pengguna menginputkan data foto setelah itu menekan tombol <i>update</i>. | <ul style="list-style-type: none"> a. Sistem akan menampilkan halaman <i>akun</i>. b. Sistem akan menampilkan halaman <i>upload</i> foto. c. Sistem akan melakukan fungsi update data dari <i>form</i> tersebut ke dalam tabel <i>customer</i> dan setelah itu menampilkan pesan ‘berhasil’ jika berhasil disimpan, atau pesan ‘gagal’ jika gagal dalam proses penyimpanan. |
| Kondisi Akhir | Fungsi ini dapat melakukan update foto <i>customer</i> dan tersimpan ke dalam <i>database</i> . | |

7. Fungsi pembelian *hashrate* berdasarkan paket, dapat dilihat pada tabel di bawah.

Tabel 3.8 Kebutuhan Fungsi Pembelian *Hashrate*

| Nama Fungsi Stakeholder | Fungsi Pembelian <i>Hashrate</i> | |
|-------------------------|---|--|
| | Customer | |
| Deskripsi | | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pembelian <i>hashrate</i> berdasarkan paket. |
| Kondisi Awal | <ol style="list-style-type: none"> 1. <i>Customer</i> harus sudah terdaftar 2. <i>Customer</i> sudah <i>login</i> 3. <i>Interface</i> harus sudah muncul 4. Data paket harus sudah muncul 5. Stok <i>coin</i> tersedia | |
| | Aksi Stakeholder Melakukan Pembelian <i>Hashrate</i> | Respon Sistem |
| | <ul style="list-style-type: none"> a. Pengguna memilih menu <i>beli hashrate</i>. b. Pengguna memilih <i>hashrate</i> yang akan dibeli dengan menekan <i>button beli</i> | <ul style="list-style-type: none"> a. Sistem akan menampilkan halaman pembelian <i>hashrate</i>. b. Sistem akan memproses permintaan pengguna. |



8. Fungsi konfirmasi pembayaran dapat dilihat pada tabel di bawah.

Tabel 3.9 Kebutuhan Fungsi Konfirmasi Pembayaran

| Nama Fungsi | Fungsi Konfirmasi Pembayaran | |
|--------------------|---|---------------|
| Stakeholder | Customer | |
| Deskripsi | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan konfirmasi pembayaran. | |
| Kondisi Awal | 1. Customer harus sudah terdaftar 2. Customer sudah <i>login</i> 3. <i>Interface</i> harus sudah muncul 4. Customer harus sudah melakukan pembelian 5. Customer harus sudah melakukan pembayaran (transfer) | |
| Alur Komputerisasi | Aksi Stakeholder | Respon Sistem |
| | Melakukan Konfirmasi Pembayaran | |

| | | |
|----------------------|---|--|
| | <ul style="list-style-type: none"> a. Pengguna memilih menu pesanan saya. b. Pengguna memilih pesanan yang akan dibayar dengan menekan <i>button</i> bayar c. Pengguna melakukan <i>upload</i> bukti bayar dengan menekan tombol <i>Choose File</i> kemudian menekan tombol <i>Save</i>. | <ul style="list-style-type: none"> a. Sistem akan menampilkan menu pesanan saya. b. Sistem akan mengalihkan ke halaman <i>upload</i> bukti bayar. c. Sistem melakukan update database dengan merubah status pemesanan pelanggan dari “BelumBayar” menjadi “Pending” setelah itu menampilkan pesan ‘berhasil’ jika upload bukti berhasil, atau pesan ‘gagal’ jika gagal. |
| Kondisi Akhir | Fungsi ini dapat melakukan upload bukti bayar pengguna dan tersimpan ke dalam <i>database</i> . | |

9. Fungsi cek *balance altcoin* dapat dilihat pada tabel di bawah.

Tabel 3.10 Kebutuhan Fungsi Cek Balance Altcoin

| Nama Fungsi | Fungsi Cek <i>Balance Altcoin</i> | |
|--------------------|---|---------------|
| Stakeholder | <i>Customer</i> | |
| Deskripsi | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pengecekan saldo <i>altcoin</i> yang telah ditambang. | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. <i>Customer</i> harus sudah terdaftar 2. <i>Customer</i> sudah <i>login</i> 3. <i>Interface</i> harus sudah muncul 4. <i>Customer</i> harus sudah melakukan pembelian 5. <i>Customer</i> harus sudah melakukan pembayaran (transfer) 6. Pembayaran harus sudah diverifikasi (sukses) 7. <i>Admin</i> harus sudah melakukan <i>setup mining</i> terhadap <i>wallet</i> customer | |
| Alur Komputerisasi | Aksi Stakeholder | Respon Sistem |
| | Melakukan Cek <i>Balance Altcoin</i> | |

| | | |
|---------------|--|--|
| | a. Pengguna memilih menu informasi <i>balance</i> . | a. Sistem akan menampilkan menu informasi <i>balance</i> berupa tabel yang berisi kode <i>coin</i> , <i>hashrate</i> , saldo <i>balance</i> , dan alamat <i>wallet</i> . |
| Kondisi Akhir | Fungsi ini dapat melakukan <i>monitoring</i> saldo altcoin pengguna. | |

10. Fungsi *request withdrawal* dapat dilihat pada tabel di bawah.

Tabel 3.11 Kebutuhan Fungsi *Request Withdrawal*

| Nama Fungsi Stakeholder | Fungsi <i>Request Withdrawal</i> | | | | | | |
|--|--|------------------|---------------|-------------------------------------|--|--|--|
| Deskripsi | <p><i>Customer</i></p> <p>Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pengecekan saldo altcoin yang telah ditambang.</p> | | | | | | |
| Kondisi Awal | <ol style="list-style-type: none"> 1. <i>Customer</i> harus sudah terdaftar 2. <i>Customer</i> sudah <i>login</i> 3. <i>Interface</i> harus sudah muncul 4. <i>Admin</i> harus sudah melakukan <i>setup mining</i> terhadap <i>wallet customer</i> 5. Informasi saldo <i>balance</i> harus sudah muncul | | | | | | |
| Alur Komputerisasi | <table border="1"> <thead> <tr> <th>Aksi Stakeholder</th> <th>Respon Sistem</th> </tr> </thead> <tbody> <tr> <td>Melakukan <i>Request Withdrawal</i></td> <td></td> </tr> <tr> <td> <ol style="list-style-type: none"> a. Pengguna memilih menu informasi <i>balance</i>. b. Pengguna menekan tombol <i>withdraw</i>. c. Pengguna mengisi <i>form amount withdrawal</i> sesuai yang dikehendaki kemudian menekan tombol <i>Update</i> </td> <td> <ol style="list-style-type: none"> a. Sistem akan menampilkan menu informasi <i>balance</i> berupa tabel yang berisi kode <i>coin</i>, <i>hashrate</i>, saldo <i>balance</i>, dan alamat <i>wallet</i>. b. Sistem mengalihkan ke halaman <i>withdrawal</i> dan menampilkan <i>form balance</i> dan <i>amount withdrawal</i>. c. Sistem melakukan pengecekan apakah <i>amount withdrawal</i> kurang dari atau melebihi <i>balance</i> pengguna, jika semua </td> </tr> </tbody> </table> | Aksi Stakeholder | Respon Sistem | Melakukan <i>Request Withdrawal</i> | | <ol style="list-style-type: none"> a. Pengguna memilih menu informasi <i>balance</i>. b. Pengguna menekan tombol <i>withdraw</i>. c. Pengguna mengisi <i>form amount withdrawal</i> sesuai yang dikehendaki kemudian menekan tombol <i>Update</i> | <ol style="list-style-type: none"> a. Sistem akan menampilkan menu informasi <i>balance</i> berupa tabel yang berisi kode <i>coin</i>, <i>hashrate</i>, saldo <i>balance</i>, dan alamat <i>wallet</i>. b. Sistem mengalihkan ke halaman <i>withdrawal</i> dan menampilkan <i>form balance</i> dan <i>amount withdrawal</i>. c. Sistem melakukan pengecekan apakah <i>amount withdrawal</i> kurang dari atau melebihi <i>balance</i> pengguna, jika semua |
| Aksi Stakeholder | Respon Sistem | | | | | | |
| Melakukan <i>Request Withdrawal</i> | | | | | | | |
| <ol style="list-style-type: none"> a. Pengguna memilih menu informasi <i>balance</i>. b. Pengguna menekan tombol <i>withdraw</i>. c. Pengguna mengisi <i>form amount withdrawal</i> sesuai yang dikehendaki kemudian menekan tombol <i>Update</i> | <ol style="list-style-type: none"> a. Sistem akan menampilkan menu informasi <i>balance</i> berupa tabel yang berisi kode <i>coin</i>, <i>hashrate</i>, saldo <i>balance</i>, dan alamat <i>wallet</i>. b. Sistem mengalihkan ke halaman <i>withdrawal</i> dan menampilkan <i>form balance</i> dan <i>amount withdrawal</i>. c. Sistem melakukan pengecekan apakah <i>amount withdrawal</i> kurang dari atau melebihi <i>balance</i> pengguna, jika semua | | | | | | |

| | | |
|---------------|--|--|
| | | kondisi terpenuhi maka sistem akan menampilkan pesan ‘berhasil’ atau pesan ‘gagal’ jika tidak terpenuhi. |
| Kondisi Akhir | Fungsi ini dapat melakukan <i>request withdrawal altcoin</i> pengguna. | |

11. Fungsi konfirmasi pembayaran *altcoin* dapat dilihat pada tabel di bawah.

Tabel 3.12 Kebutuhan Fungsi Konfirmasi Pembayaran *Altcoin*

| Nama Fungsi | Fungsi Pembayaran <i>Altcoin</i> | | | | | | |
|---|--|------------------|---------------|--|--|---|---|
| <i>Stakeholder</i> | <i>Admin</i> | | | | | | |
| <i>Deskripsi</i> | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pembayaran <i>request withdrawal</i> yang telah dilakukan oleh <i>customer</i> . | | | | | | |
| <i>Kondisi Awal</i> | <ol style="list-style-type: none"> 1. <i>Customer</i> sudah melakukan <i>request withdrawal</i> 2. <i>Interface</i> harus sudah muncul 3. <i>Admin</i> harus sudah <i>login</i> | | | | | | |
| <i>Alur Komputerisasi</i> | <table border="1"> <thead> <tr> <th>Aksi Stakeholder</th> <th>Respon Sistem</th> </tr> </thead> <tbody> <tr> <td align="center" colspan="2">Melakukan Pembayaran <i>Altcoin</i></td> </tr> <tr> <td> <ol style="list-style-type: none"> a. Pengguna memilih menu <i>request withdrawal</i>. b. Pengguna menginput data TxID setelah melakukan pengiriman <i>altcoin</i> dan merubah status menjadi sukses kemudian menekan tombol <i>Update</i>. </td> <td> <ol style="list-style-type: none"> a. Sistem akan menampilkan <i>form request withdrawal</i> yang berisi username, <i>amount WD</i>, <i>balance</i>, kode <i>coin</i>, <i>wallet</i>, TxID, dan status. b. Sistem melakukan update database dengan merubah status <i>request withdraw</i> dari “Pending” menjadi “Sukses” setelah itu menampilkan pesan ‘berhasil’ jika update data berhasil atau pesan ‘gagal’ jika gagal. </td> </tr> </tbody> </table> | Aksi Stakeholder | Respon Sistem | Melakukan Pembayaran <i>Altcoin</i> | | <ol style="list-style-type: none"> a. Pengguna memilih menu <i>request withdrawal</i>. b. Pengguna menginput data TxID setelah melakukan pengiriman <i>altcoin</i> dan merubah status menjadi sukses kemudian menekan tombol <i>Update</i>. | <ol style="list-style-type: none"> a. Sistem akan menampilkan <i>form request withdrawal</i> yang berisi username, <i>amount WD</i>, <i>balance</i>, kode <i>coin</i>, <i>wallet</i>, TxID, dan status. b. Sistem melakukan update database dengan merubah status <i>request withdraw</i> dari “Pending” menjadi “Sukses” setelah itu menampilkan pesan ‘berhasil’ jika update data berhasil atau pesan ‘gagal’ jika gagal. |
| Aksi Stakeholder | Respon Sistem | | | | | | |
| Melakukan Pembayaran <i>Altcoin</i> | | | | | | | |
| <ol style="list-style-type: none"> a. Pengguna memilih menu <i>request withdrawal</i>. b. Pengguna menginput data TxID setelah melakukan pengiriman <i>altcoin</i> dan merubah status menjadi sukses kemudian menekan tombol <i>Update</i>. | <ol style="list-style-type: none"> a. Sistem akan menampilkan <i>form request withdrawal</i> yang berisi username, <i>amount WD</i>, <i>balance</i>, kode <i>coin</i>, <i>wallet</i>, TxID, dan status. b. Sistem melakukan update database dengan merubah status <i>request withdraw</i> dari “Pending” menjadi “Sukses” setelah itu menampilkan pesan ‘berhasil’ jika update data berhasil atau pesan ‘gagal’ jika gagal. | | | | | | |
| Kondisi Akhir | Fungsi ini dapat melakukan proses pembayaran <i>altcoin</i> . | | | | | | |

12. Fungsi laporan dapat dilihat pada tabel di bawah.

Tabel 3.13 Kebutuhan Fungsi Laporan

| Nama Fungsi | Fungsi Laporan | |
|---------------------------|--|--|
| Stakeholder | <i>Owner</i> | |
| Deskripsi | Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pembuatan laporan oleh <i>owner</i> . | |
| Kondisi Awal | 1. Sudah ada data pembayaran 2. Sudah ada data <i>coin</i> 3. Sudah ada data <i>customer</i> 4. <i>Interface</i> harus sudah muncul 5. <i>Owner</i> harus sudah <i>login</i> | |
| Alur Komputerisasi | Aksi Stakeholder | Respon Sistem |
| | Melakukan Pembuatan Laporan Pendapatan | |
| | a. Pengguna memilih menu laporan pembelian. b. Pengguna menginput data tanggal mulai dan tanggal akhir kemudian menekan tombol Cetak. | a. Sistem akan menampilkan form laporan pembelian yang berisi tanggal mulai dan tanggal akhir. b. Sistem mengambil data pembelian dari tanggal mulai sampai tanggal akhir kemudian menampilkan total pendapatan pada halaman pdf. |
| | Melakukan Pembuatan Laporan Stok Coin | |
| | a. Pengguna memilih menu laporan stok <i>coin</i> . b. Pengguna menekan tombol Cetak. | a. Sistem akan menampilkan form laporan stok <i>coin</i> . b. Sistem mengambil data <i>coin</i> dan menampilkan sisa stok <i>coin</i> pada halaman pdf per tanggal cetak. |
| | Melakukan Pembuatan Laporan Customer | |
| | a. Pengguna memilih menu laporan <i>customer</i> . b. Pengguna menginput data tanggal mulai dan tanggal akhir kemudian menekan tombol Cetak. | a. Sistem akan menampilkan form laporan pembelian yang berisi tanggal mulai dan tanggal akhir. b. Sistem mengambil data <i>customer</i> dari tanggal mulai sampai tanggal akhir kemudian |

| | | |
|---------------|--|---|
| | | menampilkan total <i>customer</i> terdaftar pada tanggal tersebut di halaman pdf. |
| Kondisi Akhir | Fungsi ini dapat melakukan proses pembuatan laporan. | |

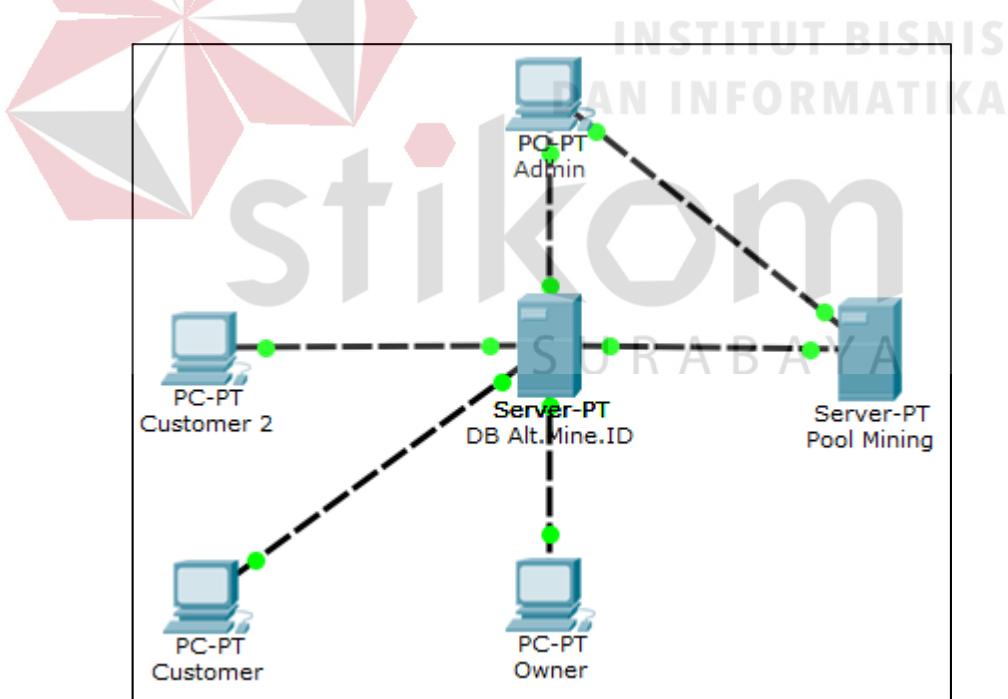
Selanjutnya dibuat analisis kebutuhan fungsional dari startup ALT.Mine.ID. Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan sistem yang mendukung berjalannya kebutuhan fungsional. Berikut adalah peranan dari kebutuhan non-fungsional yang ditunjukkan pada tabel.

Tabel 3.14 Kebutuhan Non-Fungsional

| No | Non Fungsional System | Deskripsi |
|----|-----------------------|---|
| 1 | <i>Security</i> | <p>Setiap <i>user</i> diberikan batasan hak akses dengan <i>username</i> dan <i>password</i>. <i>User</i> dalam sistem ini ada bagian <i>admin</i>, <i>customer</i>, dan <i>owner</i>.</p> <ol style="list-style-type: none"> 1. <i>Admin</i> memiliki hak akses untuk me-<i>maintenance</i> data coin, paket, kontrak, pembayaran, dan verif. 2. <i>Customer</i> memiliki hak akses untuk melakukan verifikasi akun, pembelian hashrate dan konfirmasi pembayaran pada <i>website</i>. 3. <i>Owner</i> memiliki hak akses untuk mendapatkan informasi laporan pendapatan, stok coin, dan customer |
| 2 | <i>Time Behaviour</i> | <p>Maksimal waktu pemrosesan/<i>loading</i> 5 detik.</p> <p>A. Untuk proses loading pada pembelian hashrate dan konfirmasi pembayaran membutuhkan waktu masing-masing 3 detik</p> |
| 3 | <i>Accuracy</i> | <p>Data yang tersimpan sesuai dengan data yang diinputkan.</p> <ol style="list-style-type: none"> 1. Data pembelian dan data pembayaran yang tersimpan sesuai dengan inputan customer. 2. <i>Admin</i> melakukan manajemen data coin, paket, dan kontrak konfirmasi kemudian melakukan verifikasi pembayaran sesuai dengan data pembayaran, dan verifikasi account sesuai data yang telah diupload oleh customer. |

| No | Non Fungsional System | Deskripsi |
|----|-----------------------|---|
| 4 | <i>Operability</i> | <p>3. Laporan pendapatan, stok coin, dan customer yang diterima owner berdasarkan data yang telah ada sebelumnya.</p> <p>Mudah dalam pengoperasian.</p> <ol style="list-style-type: none"> 1. <i>Customer</i> dapat dengan mudah mengoperasikan website reservasi karena tertata dan mudah dipahami. 2. <i>Admin</i> dapat dengan mudah mengelola sistem karena aplikasi mudah dipahami. 3. <i>Owner</i> dapat dengan mudah mencetak laporan dengan menginputkan periode tanggal |

Berikut ini merupakan desain arsitektur dari Alt.Mine.ID dimana sistem membutuhkan external entity dari pool mining untuk mendapatkan informasi mining.



Gambar 3.4 Desain Arsitektur

3.2. Product Backlog

Setelah mengidentifikasi masalah, maka tahap selanjutnya adalah *product backlog*. Pada tahap ini dibuat *list story* hasil dari mengidentifikasi masalah. *Story* yang dibuat dapat dilihat pada Tabel 3.15 dibawah.

Tabel 3.15 *Product Backlog*

| ID | Story | Prioritas |
|----|----------------------------------|-----------|
| 1 | Fitur <i>free sign-up</i> member | 3 |
| 2 | Fitur pembelian <i>hashrate</i> | 4 |
| 3 | Fitur informasi <i>balance</i> | 5 |
| 4 | Fitur manajemen <i>coin</i> | 1 |
| 5 | Fitur <i>approval withdrawal</i> | 6 |
| 6 | Fitur laporan | 7 |
| 7 | Fitur manajemen kontrak | 2 |

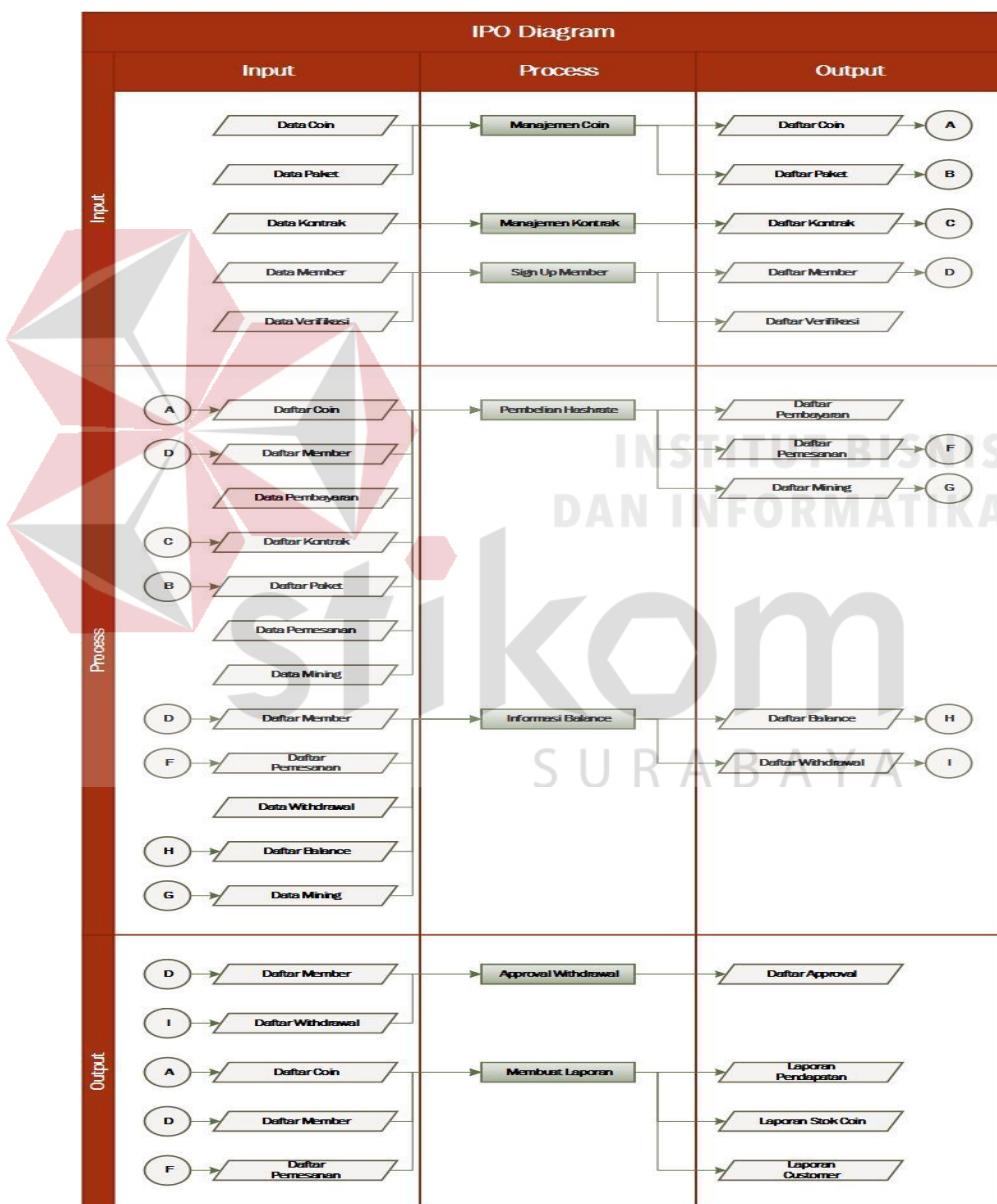
Pengerjaan product backlog diatas akan dikerjakan dalam 7 sprint. Penjelasan detail pengerjaan urut berdasarkan sprint dapat dilihat pada tabel dibawah.

Tabel 3.16 *Sprint Story*

| Sprint | Story | Perencanaan | Perencanaan | Total |
|--------|----------------------------------|-------------|-------------|-----------|
| | | Jumlah Hari | Waktu Jam | Jam Kerja |
| 1 | Fitur manajemen <i>coin</i> | 7 | 5 | 35 |
| 2 | Fitur manajemen kontrak | 7 | 5 | 35 |
| 3 | Fitur <i>free sign-up</i> member | 7 | 5 | 35 |
| 4 | Fitur pembelian <i>hashrate</i> | 7 | 5 | 35 |
| 5 | Fitur informasi <i>balance</i> | 7 | 5 | 35 |
| 6 | Fitur <i>approval withdrawal</i> | 7 | 5 | 35 |
| 7 | Fitur Laporan | 7 | 5 | 35 |

Blok diagram merupakan diagram dari sebuah sistem. dimana bagian utama atau fungsi yang digambarkan dengan blok dihubungkan dengan garis, yang

menunjukkan hubungan dari blok. Data yang dibutuhkan oleh sistem akan diolah sesuai dengan fungsi yang sudah dibuat pada sistem. Pada blok diagram *startup* ALT.Mine.ID ini didasarkan dari hasil analisis kebutuhan fungsional yang telah dilakukan, maka dapat digambarkan blok diagram untuk aplikasi sebagai berikut pada Gambar 3.3.



Gambar 3.5 Blok Diagram

Berdasarkan penjelasan blok diagram di atas, dapat dijelaskan secara mendetail yaitu:

1. Kategori *Input*

a. Data *Coin*

Di dalam inputan data *coin* terdapat kategori *coin* (Ethereum, Siacoin, Decred, Pascal, Ethereum Classic). Data *master* ini bertujuan agar memudahkan *admin* dalam manajemen pengelolaan data *coin* yang akan ditambang oleh pengguna.

b. Data Paket

Di dalam inputan data paket terdapat daftar harga paket *coin* (Ethereum, Siacoin, Decred, Pascal, Ethereum Classic). Sesuai dengan *hashrate* yang disediakan

c. Data Kontrak

Di dalam inputan data kontrak terdapat kategori kontrak terhadap *coin* (Ethereum, Siacoin, Decred, Pascal, Ethereum Classic). Data *master* ini bertujuan agar memudahkan *member* dan *admin* melakukan persetujuan.

d. Data *Member*

Di dalam inputan data *member* terdapat (e-mail, *password* baru, ulangi *password* baru). *E-mail* dan *password* yang telah didaftarkan, akan dapat digunakan untuk *login member*.

e. Data Verifikasi

Di dalam inputan data verifikasi terdapat foto diri berupa *scan ktp* dan *kyc* yang nantinya berguna untuk proses verifikasi *member*.

f. Data Pembayaran

Di dalam inputan data pembayaran terdapat informasi bukti pembayaran yang telah dilakukan oleh member untuk membeli *hashrate*.

g. Data Pemesanan

Di dalam inputan data pemesanan terdapat informasi mengenai harga coin yang akah dibeli dan *hashrate*-nya.

h. Data *Mining*

Di dalam inputan data *mining* terdapat informasi status *balance* dari member yang telah sukses melakukan transaksi pembelian *hashrate*

i. Data *Withdrawal*

Di dalam inputan data *withdrawal* terdapat kategori (*amount withdrawal*, alamat *wallet* yang dituju). Data *withdrawal* ini bertujuan untuk memudahkan *admin* dalam manajemen pembayaran *coin* sudah ditambang oleh pengguna.

2. Kategori Proses

a. Manajemen *Coin*

Untuk dapat melakukan proses ini, butuh inputan *master* data *coin*.

b. Manajemen Kontrak

Untuk dapat melakukan proses ini, butuh inputan *master* data kontrak.

c. *Sign-Up Member*

Untuk dapat melakukan proses ini, butuh inputan *master* data *member*.

d. Pembelian *Hashrate*

Untuk dapat melakukan proses ini, butuh inputan data Daftar *coin*, Daftar member, Daftar *hashrate*, dan Data pembayaran. *Member* dapat

menentukan *hashrate* yang ingin dibeli setelah memilih *coin* yang ingin ditambang.

e. Informasi *Balance*

Untuk dapat melakukan proses ini, butuh inputan data Daftar member, dan Daftar pembelian. *Member* dapat melihat pendapatan coin dengan *hashrate* yang telah dibeli. *Member* dapat melakukan penarikan dalam jumlah tertentu sesuai keinginan.

f. Approval *Withdrawal*

Untuk dapat melakukan proses ini, butuh inputan data Daftar admin, dan Daftar *withdrawal*. *Admin* akan melakukan pembayaran sejumlah *coin* kepada member apabila semua persyaratan telah terpenuhi.

g. Membuat Laporan

Untuk dapat melakukan proses ini, butuh inputan data Daftar admin, Daftar pembelian, Data pembelian, dan Bukti pembayaran. *Admin* akan melakukan rekap data terhadap pembayaran yang telah dilakukan oleh *member* untuk pembelian *hashrate*.

3. Kategori *Output*

a. Daftar *Coin*

Untuk dapat menghasilkan keluaran *master* daftar coin, terlebih dahulu *admin* membuat *master* data *coin* kemudian melakukan proses pengelolaan data tersebut. *Output* data *coin* (Ethereum, Siacoin, Decred, Pascal, Ethereum Classic).

b. Daftar Paket

Untuk dapat menghasilkan keluaran daftar paket, terlebih dahulu admin membuat master data coin dan kontrak kemudian menambah data coin agar coin yang telah ditambahkan ke dalam data master dapat memiliki daftar paket.

c. Daftar Kontrak

Untuk dapat menghasilkan keluaran *master* persetujuan kontrak, terlebih dahulu *admin* membuat *master* data kontrak kemudian melakukan proses pengelolaan data tersebut. *Output* data *coin* (Kontrak pembelian Ethereum, kontrak pembelian Siacoin, kontrak pembelian Decred, kontrak pembelian Pascal, kontrak pembelian Ethereum Classic).

d. Daftar *Member*

Untuk dapat menghasilkan keluaran daftar *member*, perlu inputan data *member* dan proses registrasi yang dilakukan oleh *member*. Untuk dapat melakukan pembelian *hashrate*.

e. Daftar Verifikasi

Untuk dapat menghasilkan keluaran daftar verifikasi diperlukan proses *upload* data berupa scan ktp dan kyc oleh *member*. Ini berfungsi untuk melakukan verifikasi akun member guna melakukan pembelian.

f. Daftar Pemesanan

Untuk dapat menghasilkan keluaran daftar *pembelian*, diperlukan proses pembelian *hashrate*. Ini berfungsi untuk memverifikasi *member* dan memudahkan *member* memilih dan membeli *hashrate* yang ingin dibeli.

g. Daftar Pembayaran

Untuk dapat menghasilkan keluaran daftar pembayaran, diperlukan proses pembelian *hashrate*. Ini berfungsi untuk memverifikasi *member* dan admin untuk memverifikasi pembelian *hashrate* oleh *member*.

h. Daftar *mining*

Untuk dapat menghasilkan keluaran daftar *mining*, diperlukan proses verifikasi pembayaran oleh admin terhadap pembelian *hashrate* yang dilakukan *customer*. Jika pembayaran valid maka daftar *mining* akan diupdate oleh admin

i. Daftar *Balance*

Untuk dapat menghasilkan keluaran daftar *balance*, diperlukan proses informasi *balance*. Ini berfungsi untuk memperlihatkan status penambangan dan *hashrate* yang didapat oleh *member* setelah melakukan pembelian.

j. Daftar *Withdrawal*

Untuk dapat menghasilkan keluaran daftar *withdrawal*, diperlukan proses *custom withdrawal*. Ini berfungsi untuk memperlihatkan status *withdrawal* yang dilakukan oleh *member*.

k. Daftar *Approval*

Untuk dapat menghasilkan keluaran daftar *withdrawal*, diperlukan proses *approval withdrawal*. Ini berfungsi untuk admin melakukan *approval* dan pembayaran sejumlah *coin* kepada *member*.

1. Data *Withdrawal*

Untuk dapat menghasilkan keluaran data withdrawal, diperlukan proses *approval withdrawal*. Ini berfungsi untuk admin melakukan rekap data pembayaran *coin* yang telah dilakukan kepada *member*.

m. Laporan Pendapatan

Untuk dapat menghasilkan keluaran laporan pendapatan, diperlukan proses pembuatan laporan. Ini berfungsi untuk admin melakukan rekap data pendapatan penjualan *hashrate*.

n. Laporan Stok *Coin*

Untuk dapat menghasilkan keluaran laporan stok *coin*, diperlukan proses pembuatan laporan. Ini berfungsi untuk admin melakukan rekap data stok *coin* yang tersisa.

o. Laporan *Customer*

Untuk dapat menghasilkan keluaran laporan *customer*, diperlukan proses pembuatan laporan. Ini berfungsi untuk admin melakukan rekap data jumlah *customer* yang terdaftar.

3.3. *Sprint Planning*

Sprint planning adalah tahap sebelum melakukan *sprint backlog* untuk menentukan *sprint goal* dan fungsi dari setiap *story* pada *product backlog* yang kemudian dimasukkan ke *sprint backlog*, selain itu ditentukan juga berapa kali *sprint* yang akan dilaksanakan untuk mengerjakan aplikasi Alt.Mine.ID.

3.3.1. Sprint Planning 1 Manajemen Coin

Sprint planning pertama memiliki yaitu aplikasi Alt.Mine.ID dapat digunakan untuk menjalankan fitur manajemen *coin* dengan fungsi penambahan *coin* baru dan setting paket *altcoin*.

3.3.2. Sprint Planning 2 Manajemen Kontrak

Sprint planning kedua pada *startup* ini memiliki *sprint goal* aplikasi Alt.Mine.ID dapat digunakan untuk menjalankan fitur manajemen kontrak dengan fungsi setting kontrak *altcoin*.

3.3.3. Sprint Planning 3 Free Sign-Up Member

Sprint planning ketiga pada *startup* ini memiliki *sprint goal* aplikasi Alt.Mine.ID dapat digunakan untuk menjalankan fitur *free sign up member* dengan fungsi verifikasi, *edit wallet*, dan *upload foto*.

3.3.4. Sprint Planning 4 Pembelian Hashrate

Sprint planning keempat pada *startup* ini memiliki *sprint goal* aplikasi Alt.Mine.ID dapat digunakan untuk menjalankan fitur pembelian hashrate dengan fungsi pembelian *hashrate* dan konfirmasi pembayaran.

3.3.5. Sprint Planning 5 Informasi Balance

Sprint planning kelima pada *startup* ini memiliki *sprint goal* aplikasi Alt.Mine.ID dapat digunakan untuk menjalankan fitur informasi *balance* dengan fungsi cek *balance altcoin* dan *request withdrawal*.

3.3.6. Sprint Planning 6 Approval Withdrawal

Sprint planning keenam pada *startup* ini memiliki *sprint goal* aplikasi Alt.Mine.ID dapat digunakan untuk menjalankan fitur *approval withdrawal* dengan fungsi konfirmasi pembayaran *altcoin*.

3.3.7. Sprint Planning 7 Laporan

Sprint planning ketujuh pada *startup* ini memiliki *sprint goal* aplikasi Alt.Mine.ID dapat digunakan untuk menjalankan fitur laporan dengan fungsi pembuatan laporan pembelian, laporan *customer*, dan laporan stok *coin*.

3.4. Sprint Backlog

Sprint backlog adalah hasil dari *sprint planning*. *Sprint backlog* terdapat fungsi dari setiap *story* yang telah didapat di *product backlog* dan *sprint goal*. Item pada *sprint backlog* ditentukan oleh *product owner*. *Product backlog* yang sudah dibuat dan telah dilakukan *sprint planning*, maka tahap selanjutnya adalah *sprint backlog*.

3.4.1. Sprint Backlog 1 Manajemen Coin

Product backlog yang sudah dibuat dan telah dilakukannya *sprint planning*, maka tahap selanjutnya adalah *sprint backlog*. Pada *sprint backlog* manajemen *coin* terdapat 2 fungsi yang dibuat yaitu penambahan *coin* baru dan *setting* paket.

3.4.2. Sprint Backlog 2 Manajemen Kontrak

Product backlog yang sudah dibuat dan telah dilakukannya *sprint planning*, maka tahap selanjutnya adalah *sprint backlog*. Pada *sprint backlog* manajemen kontrak terdapat 1 fungsi yang dibuat yaitu *setting* kontrak.

3.4.3. Sprint Backlog 3 Free Sign-Up Member

Product backlog yang sudah dibuat dan telah dilakukannya *sprint planning*, maka tahap selanjutnya adalah *sprint backlog*. Pada *sprint backlog free sign-up member* terdapat 2 fungsi yang dibuat yaitu verifikasi dan *edit wallet*.

3.4.4. Sprint Backlog 4 Pembelian Hashrate

Product backlog yang sudah dibuat dan telah dilakukannya *sprint planning*, maka tahap selanjutnya adalah *sprint backlog*. Pada *sprint backlog pembelian hashrate* terdapat 2 fungsi yang dibuat yaitu pembelian *hashrate* dan konfirmasi pembayaran.

3.4.5. Sprint Backlog 5 Informasi Balance

Product backlog yang sudah dibuat dan telah dilakukannya *sprint planning*, maka tahap selanjutnya adalah *sprint backlog*. Pada *sprint backlog informasi balance* terdapat 2 fungsi yang dibuat yaitu cek *balance* dan *request withdrawal*.

3.4.6. Sprint Backlog 6 Approval Withdrawal

Product backlog yang sudah dibuat dan telah dilakukannya *sprint planning*, maka tahap selanjutnya adalah *sprint backlog*. Pada *sprint backlog approval withdrawal* terdapat 1 fungsi yang dibuat yaitu konfirmasi pembayaran *altcoin*.

3.4.7. Sprint Backlog 7 Laporan

Product backlog yang sudah dibuat dan telah dilakukannya *sprint planning*, maka tahap selanjutnya adalah *sprint backlog*. Pada *sprint backlog laporan* terdapat 3 fungsi yang dibuat yaitu laporan pembelian, laporan *customer* dan laporan stok *coin*.

BAB IV

IMPLEMENTASI DAN EVALUASI

Pembahasan desain, implementasi, dan uji coba sistem akan di jelaskan berdasarkan *list story* pada *product backlog* yang telah dibuat pada bab 3.

4.1. Sprint 1 Manajemen Coin

Pada *sprint 1* akan dijelaskan tahapan pengerjaan fitur manajemen *coin* beserta fungsi-fungsi yang ada di dalamnya.

4.1.1. Sprint Planning

Sprint planning pertama mempunyai *goal* yaitu dapat menjalankan fitur manajemen *coin* dengan fungsi penambahan *coin* baru dan setting paket *altcoin*. Untuk menyelesaikan *goal* tersebut maka dibuat:

1. *System flow* dengan fungsi login admin dan CRUD untuk memanajemen data *coin* dan paket.
2. Desain basis data menggunakan 2 tabel yaitu tabel *coin* dan tabel paket.
3. Desain *interface* dengan menghasilkan desain halaman login, halaman utama, dan halaman admin.
4. Desain uji coba menggunakan blackbox testing dengan melakukan pengujian *form* login, menu manajemen *coin*, dan menu manajemen paket.

4.1.2. Sprint Backlog

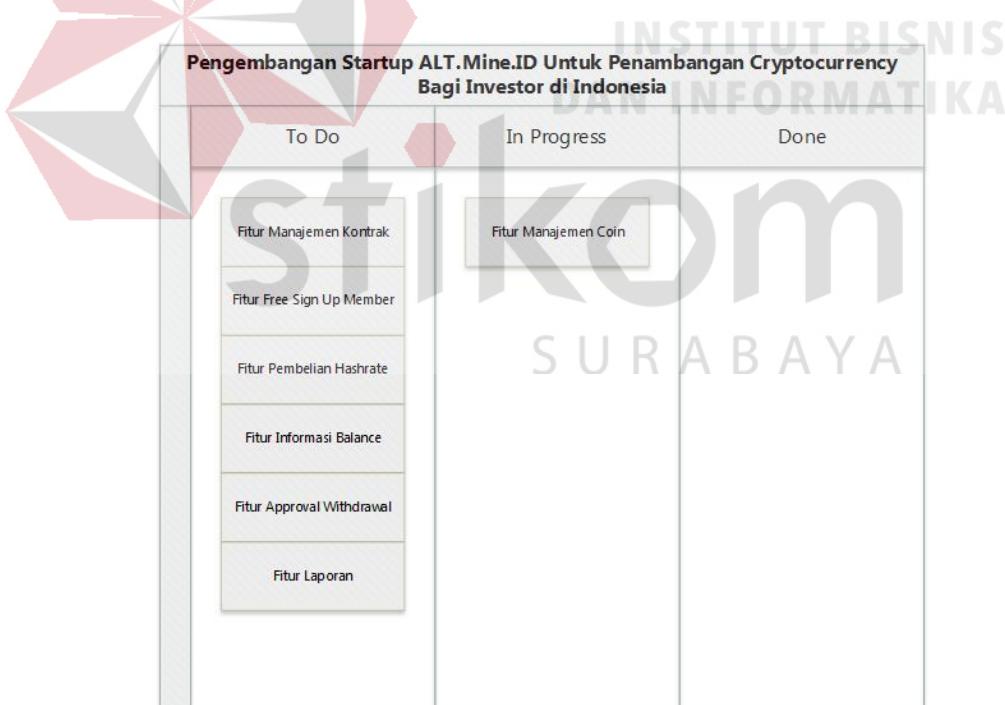
Pada *sprint backlog* manajemen *coin* terdapat 2 fungsi yang dibuat yaitu penambahan *coin* baru dan setting paket. Tabel *sprint backlog* 1 dapat dilihat pada Tabel 4.1

Tabel 4.1 *Sprint Backlog Manajemen Coin*

| No | Story | Fungsi | Goal |
|----------|----------------------|------------------------------|---|
| 1 | Fitur manajemen coin | Fungsi penambahan coin baru | Dapat menambahkan coin baru |
| | | Fungsi setting paket altcoin | Dapat melakukan <i>insert, update, delete</i> paket altcoin |

4.1.3. Daily Scrum

Pada tahap ini semua *story sprint* masuk dalam daftar yang akan dikerjakan (*to do*). Fitur manajemen coin masuk ke dalam daftar dikerjakan (*in progress*). Berikut adalah gambaran detail dari *daily scrum sprint 1*.



Gambar 4.1 *Daily Scrum Sprint 1*

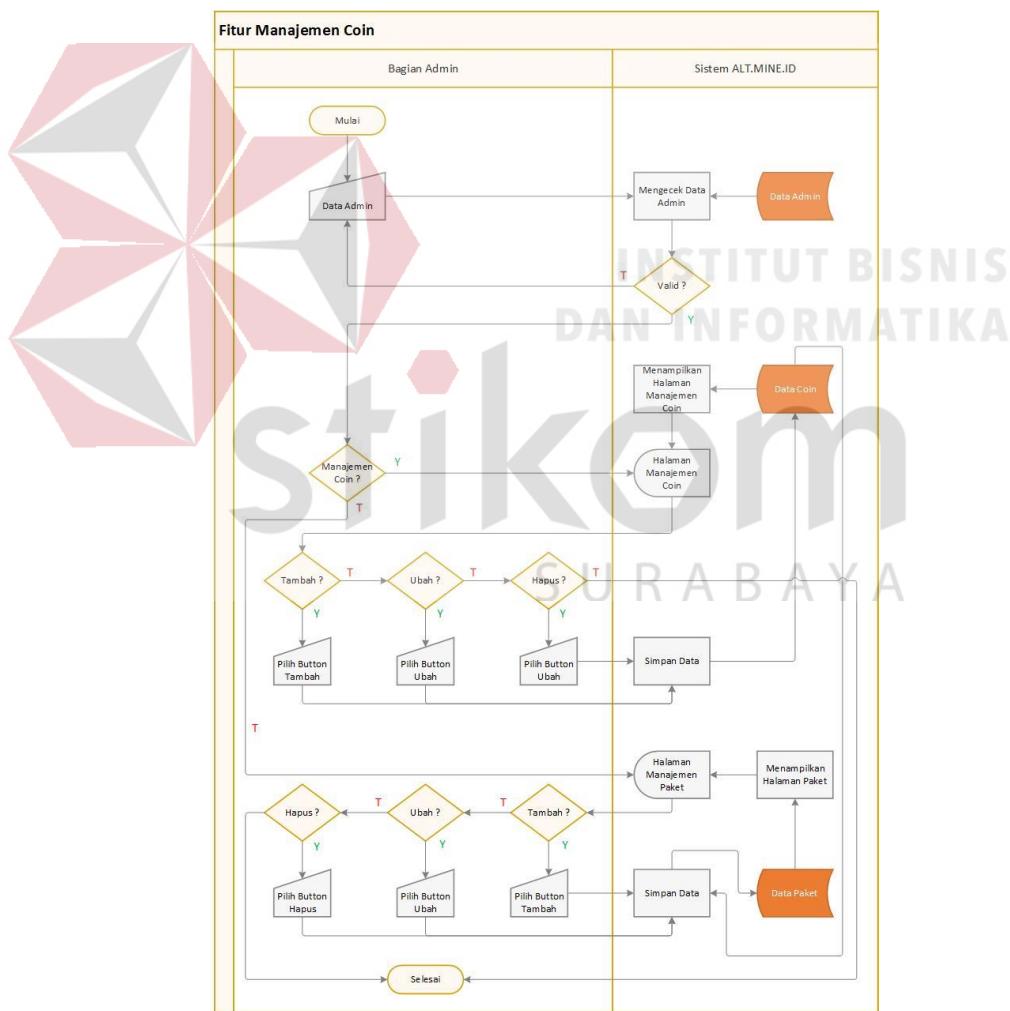
4.1.4. Pengerjaan *Sprint*

Dalam pengerjaan *sprint* ini dibuat desain sistem sampai dengan uji coba sistem yang telah jadi. Berikut adalah detail pengerjaan sistem pada *sprint* 1.

A. Desain Sistem

Setelah dilakukan perancangan selanjutnya dilakukan pembuatan desain sistem manajemen coin. Berikut adalah beberapa desain sistem yang dibuat :

1. *System Flow*

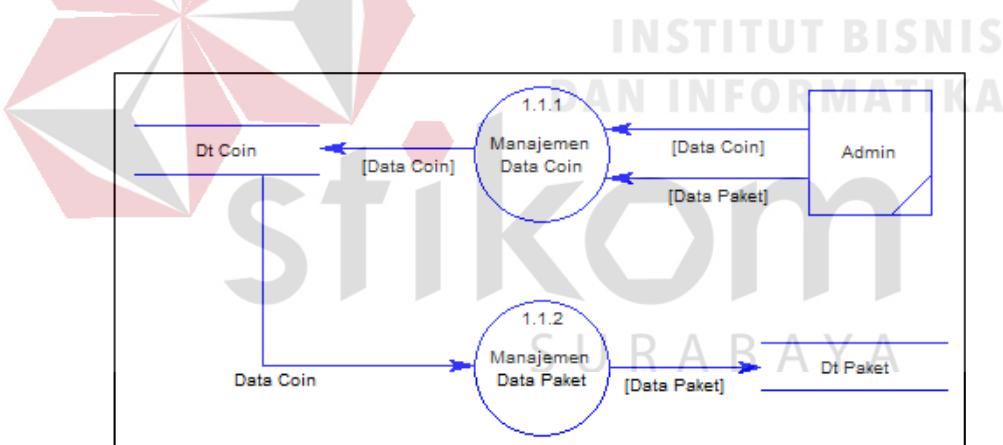


Gambar 4.2 *System Flow* Manajemen Coin

Pada Gambar 4.2 di atas, alur proses manajemen *coin* dimulai oleh *admin* dengan cara melakukan *login*. Pada saat *login* berhasil, *admin* memilih untuk masuk ke halaman manajemen *coin* atau manajemen paket. Pada halaman manajemen *coin* *admin* dapat melakukan *create*, *insert*, *update*, *delete* dan tersimpan ke dalam *database coin*. Pada halaman manajemen paket *admin* juga dapat melakukan *create*, *insert*, *update*, *delete* dan tersimpan ke dalam *database* paket.

2. Data Flow Diagram (DFD) Level 1 Manajemen Coin

DFD level 1 merupakan diagram yang menguraikan detail proses yang ada pada DFD level 0. Pada bagian ini mengalir untuk kegiatan pengolahan data *coin*, dan pengolahan data paket yang kemudian disimpan pada masing-masing tabel master. Berikut DFD level 1 manajemen *coin* dapat dilihat pada Gambar 4.3.



Gambar 4.3 DFD Level 1 Manajemen *Coin*

3. Struktur Basis Data

Pada sprint 1 ini dilakukan pembuatan tabel struktur basis data untuk fitur manajemen *coin*. Tabel pada struktur basis data yang dibentuk dapat dijelaskan sebagai berikut :

- a) Tabel Coin

Nama Tabel : Coin
Primary Key : ID_Coin
Foreign Key : -
 Fungsi : Menyimpan data *coin*

Tabel 4.2 Tabel *Coin*

| No | Field Name | Data Type | Length | Foreign Key On Field | On Table |
|----|------------|-----------|--------|-------------------------|----------|
| 1 | ID_Coin | Integer | 11 | | |
| 2 | Nama_Coin | VarChar | 30 | | |
| 3 | Kode_Coin | VarChar | 30 | | |
| 4 | Stok_Coin | Integer | 11 | | |
| 5 | Icon_Coin | VarChar | 200 | | |

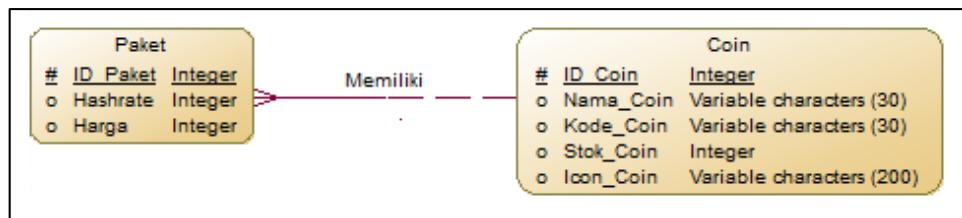


Tabel 4.3 Tabel Paket

| No | Field Name | Data Type | Length | Foreign Key On Field | On Table |
|----|------------|-----------|--------|-------------------------|----------|
| 1 | ID_Paket | Integer | 11 | | |
| 2 | Hashrate | Integer | 30 | | |
| 3 | Harga | Integer | 30 | | |

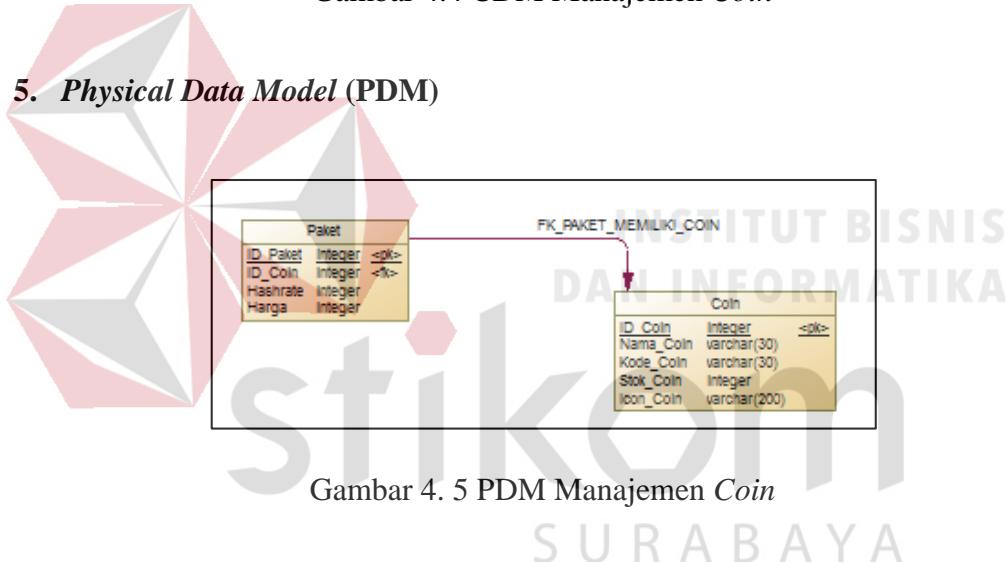
4. Conceptual Data Model (CDM)

CDM merupakan rancangan awal konsep desain *database* yang nantinya akan di *generate* kedalam bentuk *Physical Data Model*. Berikut ini merupakan desain relasi antar tabel dalam sprint manajemen *coin*.



Gambar 4.4 CDM Manajemen *Coin*

5. Physical Data Model (PDM)



Gambar 4. 5 PDM Manajemen *Coin*

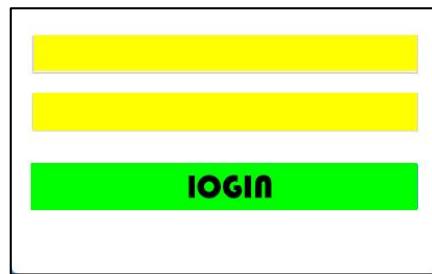
PDM merupakan hasil dari generate CDM. Secara keseluruhan PDM menggambarkan basis data yang telah jadi dan dapat di generate script kedalam database server. Berikut ini merupakan desain PDM dalam sprint manajemen coin.

6. Desain Antar Muka (*Interface*)

Pada sprint 1 ini dilakukan pembuatan desain interface halaman utama, login admin, dan halaman admin sub menu manajemen *coin*. Berikut adalah desain *interface* yang akan dibuat.:

a) Desain Halaman Login

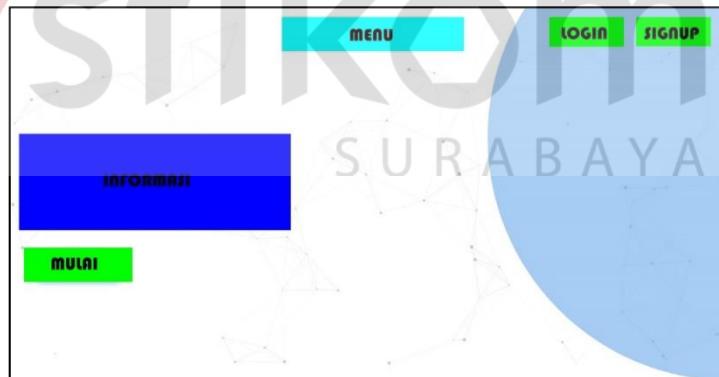
Berikut adalah desain halaman *login admin* yang digunakan sebagai acuan dalam implementasi. Halaman *login admin* dapat dilihat pada gambar 4.6 di bawah.



Gambar 4.6 Desain Halaman Login Admin

b) Desain Halaman Utama

Berikut adalah desain halaman utama yang digunakan sebagai acuan dalam implementasi halaman utama, dapat dilihat pada Gambar 4.7 di bawah.

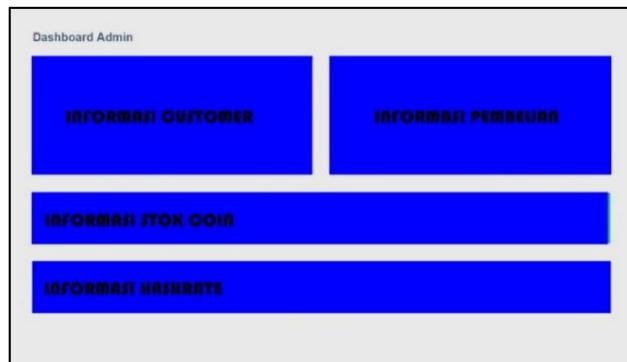


Gambar 4.7 Desain Halaman Utama

c) Desain Halaman Admin

Berikut adalah desain halaman *admin* (*sub menu* manajemen coin, manajemen paket) yang digunakan sebagai acuan dalam implementasi halaman

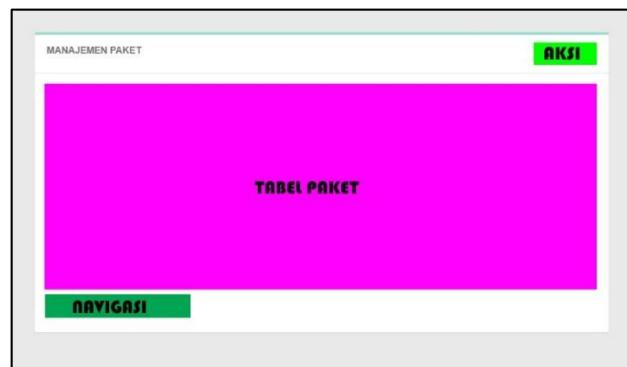
admin, dapat dilihat di bawah ini Gambar 4.8 halaman *admin*, Gambar 4.9 halaman manajemen coin, dan Gambar 4.10 halaman manajemen paket :



Gambar 4.8 Desain Halaman Admin



Gambar 4.9 Halaman Manajemen Coin



Gambar 4.10 Halaman Manajemen Paket

7. Desain Uji Coba Sistem (*BlackBox Testing*)

Pada sprint 1 ini dilakukan uji coba sistem pada halaman utama, login admin, dan halaman admin sub menu manajemen coin . Detail dari desain uji coba dapat dilihat di bawah ini :

a) Desain Uji Coba *Form Login*

Desain uji coba *login* berfungsi untuk menguji apakah fungsi *login* dapat berjalan sesuai yang diharapkan. Desain uji coba *login* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Desain Uji Coba *Form Login*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|--|---|---|
| 1 | Melakukan <i>Login</i> | Memasukkan <i>username</i> dan <i>password</i> | >Login berhasil dan masuk ke halaman utama sesuai dengan <i>role</i> masing – masing |
| 2 | Melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang salah | Memasukkan <i>username</i> dan <i>password</i> yang salah | User gagal <i>login</i> dan kembali memasukkan <i>username</i> dan <i>password</i> yang benar |

b) Desain Uji Coba Menu Manajemen *Coin*

Desain uji coba menu manajemen coin bertujuan untuk mengetahui apakah fungsi *Create, Read, Update, Delete* (CRUD) dan pencarian sesuai dengan data inputan. Desain uji coba dapat dilihat pada tabel 4.5.

Tabel 4.5 Desain Uji Coba Manajemen Coin

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|----------------------------|--|-------------------------------|
| 1 | Melakukan tambah data coin | Memasukan nama coin, kode coin, stok coin, dan icon coin | Data tersimpan ke sistem dan |

| Test Case ID | Tujuan | Input | Output Yang Diharapkan |
|--------------|---------------------------|---|---------------------------------|
| | | kemudian menyimpannya dengan mengklik tombol Simpan | menampilkan data sesuai inputan |
| 2 | Melakukan edit data coin | Mengganti data coin kemudian simpan dengan meng-klik tombol <i>update</i> | Perubahan data coin tersimpan |
| 3 | Melakukan hapus data coin | Menghapus data coin dengan meng-klik tombol Hapus | Data yang dipilih terhapus |

c) Desain Uji Coba Menu Manajemen Paket

Desain uji coba menu manajemen paket bertujuan untuk mengetahui apakah fungsi *Create, Read, Update, Delete* (CRUD) dan pencarian sesuai dengan data inputan. Desain uji coba dapat dilihat pada tabel 4.6.

Tabel 4.6 Desain Uji Coba Manajemen Paket

| Test Case ID | Tujuan | Input | Output Yang Diharapkan |
|--------------|-----------------------------|--|--|
| 1 | Melakukan tambah data paket | Memasukan kode coin, hashrate, dan harga kemudian menyimpannya dengan mengklik tombol Simpan | Data tersimpan sistem menampilkan sesuai inputan ke dan data |
| 2 | Melakukan edit data paket | Mengganti data paket kemudian simpan dengan meng-klik tombol <i>update</i> | Perubahan data paket tersimpan |
| 3 | Melakukan hapus data paket | Menghapus data paket dengan meng-klik tombol Hapus | Data yang dipilih terhapus |

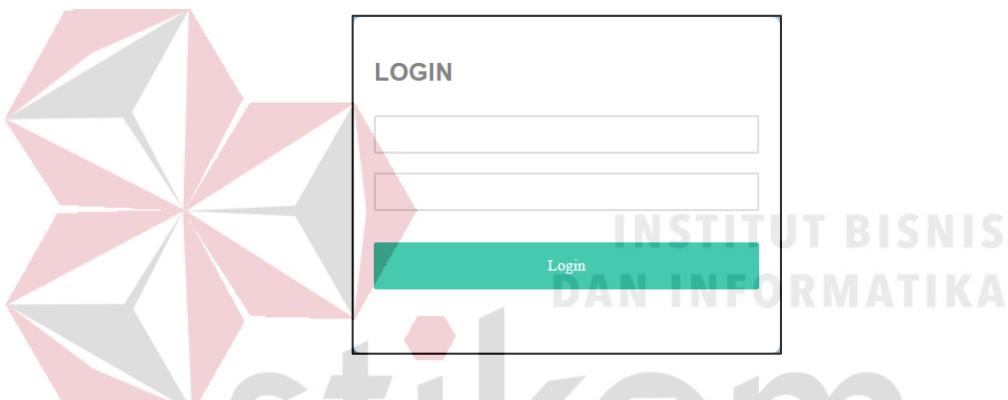
B. Implementasi Aplikasi

Setelah dilakukan pembuatan desain sistem selanjutnya dilakukan implementasi terhadap aplikasi. Berikut adalah tampilan aplikasi yang sudah dibuat.:

a) Implementasi Halaman *Login*

Halaman *login* merupakan halaman yang muncul pertama saat aplikasi dibuka. Halaman ini berfungsi untuk akses *admin*, untuk mengakses aplikasi.

Gambar 4.11 halaman *login admin*, dan Gambar 4.12 *pesan* pesan *login*.



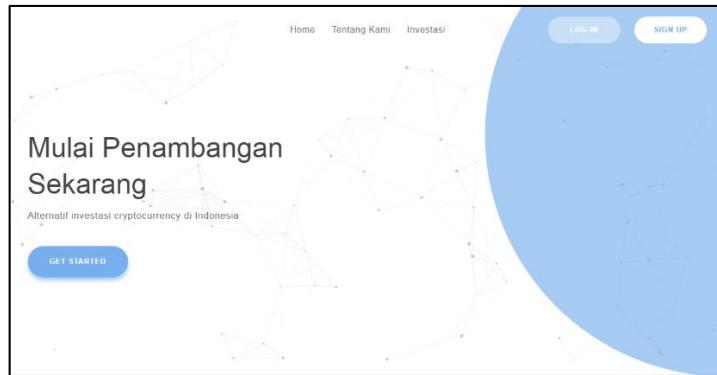
Gambar 4.11 Halaman *Login Admin*



Gambar 4.12 Pesan Gagal *Login*

b) Implementasi Halaman Utama

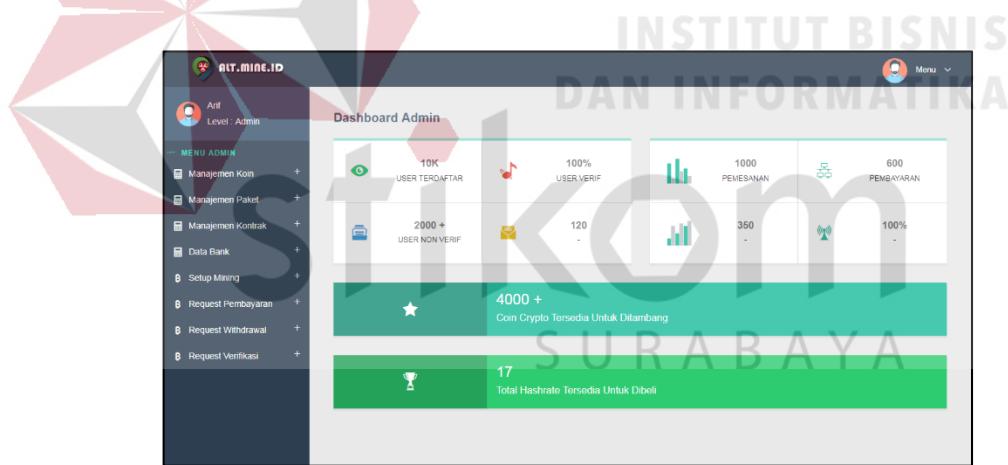
Halaman utama merupakan halaman yang muncul pertama kali saat aplikasi dibuka. Berikut adalah tampilan halaman utama dapat dilihat pada Gambar 4.13 di atas.



Gambar 4.13 Halaman Utama

c) Implementasi Halaman *Admin*

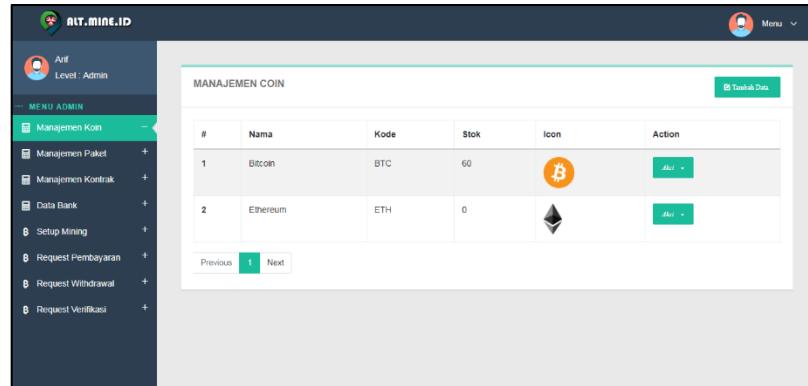
Halaman *admin* merupakan halaman yang berfungsi untuk menampilkan informasi terhadap data *master* dan data *customer*. Implementasi halaman *admin* dapat dilihat pada Gambar 4.14.



Gambar 4.14 Halaman Admin

Halaman manajemen *coin* merupakan halaman yang berfungsi untuk mengelola data *coin*. Pada halaman ini dapat menambah, mengubah, serta menghapus data. Implementasi halaman manajemen *coin* dapat dilihat pada Gambar 4.15 di bawah. Untuk menambah data baru, pilih tombol tambah data

yang berada di atas, maka akan muncul tampilan seperti pada Gambar 4.16 di bawah.



| # | Nama | Kode | Stok | Icon | Action |
|---|----------|------|------|------|----------------------|
| 1 | Bitcoin | BTC | 60 | | Edit |
| 2 | Ethereum | ETH | 0 | | Edit |

Gambar 4.15 Halaman Manajemen Coin

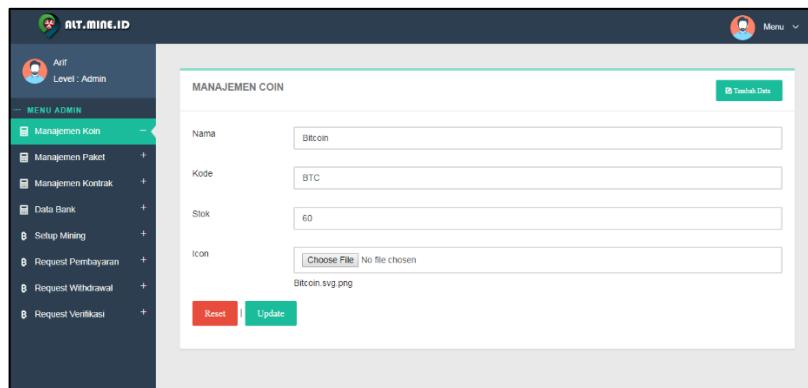


MANAJEMEN COIN

Nama:
 Kode:
 Stok:
 Icon: No file chosen

[Reset](#) [Save](#)

Gambar 4.16 Halaman Tambah Data Coin



MANAJEMEN COIN

Nama:
 Kode:
 Stok:
 Icon: Bitcoin.svg.png

[Reset](#) [Update](#)

Gambar 4.17 Halaman Edit Data Coin

Untuk mengubah data, pilih aksi *edit*, maka akan muncul tampilan seperti pada Gambar 4.17 di atas. Untuk menghapus data, pilih aksi *hapus*, maka akan muncul notifikasi seperti pada gambar 4.18.



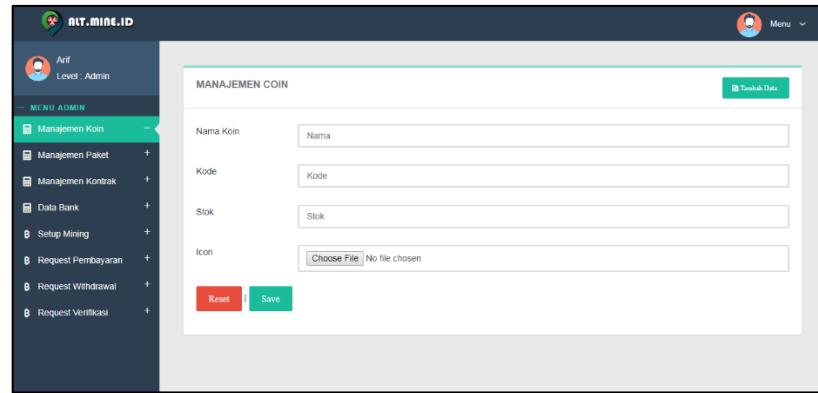
Gambar 4.18 Notifikasi Hapus Data Coin

Halaman manajemen paket merupakan halaman yang berfungsi untuk mengelola data paket. Pada halaman ini dapat menambah, mengubah, serta menghapus data. Implementasi halaman manajemen paket dapat dilihat pada Gambar 4.19.

| # | KodeCoin | Hashrate | Harga | Action |
|---|----------|----------|---------|-----------------------|
| 1 | ETH | 120 | 1000000 | <button>Edit</button> |
| 2 | BTC | 120 | 1000000 | <button>Edit</button> |
| 3 | ETH | 60 | 800000 | <button>Edit</button> |
| 4 | BTC | 60 | 800000 | <button>Edit</button> |
| 5 | BTC | 20 | 500000 | <button>Edit</button> |

Gambar 4.19 Halaman Manajemen Paket

Untuk menambah data baru, pilih tombol tambah data yang berada di atas, maka akan muncul tampilan seperti pada Gambar 4.20.

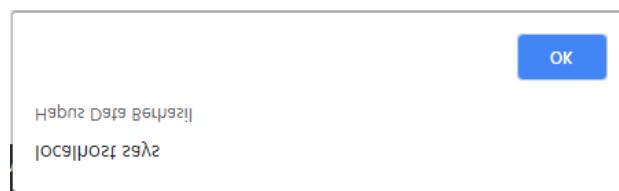


Gambar 4.20 Halaman Tambah Data Paket

Untuk mengubah data, pilih aksi *edit*, maka akan muncul tampilan seperti pada Gambar 4.21.



Gambar 4.21 Halaman Edit Data Paket



Gambar 4. 22 Notifikasi Hapus Data Paket

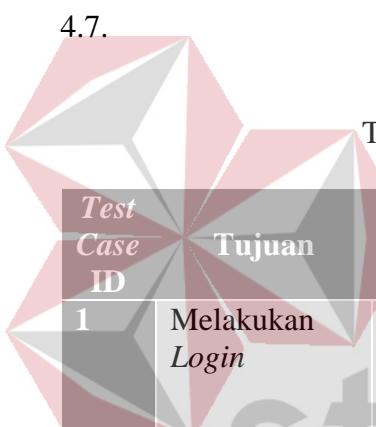
Untuk menghapus data, pilih aksi hapus, maka akan muncul notifikasi seperti pada Gambar 4.22 di atas.

C. Ujicoba dan Evaluasi Sistem

Pada sprint 1 dilakukan ujicoba dan evaluasi *form login*, menu manajemen *coin*, dan menu manajemen paket. Berikut adalah hasil ujicoba dan evaluasi sistem yang telah dibuat :

a) Uji Coba Form Login Admin

Proses uji coba *login admin* dimulai dengan memasukkan *username* dan *password* yang telah di daftarkan. Proses *login* dinyatakan berhasil apabila sistem telah masuk ke halaman utama. Hasil uji coba dapat dilihat pada Tabel 4.7.



| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|--|---|---|----------------------|
| 1 | Melakukan <i>Login</i> | Memasukkan <i>username</i> dan <i>password</i> | Login berhasil dan masuk ke halaman utama sesuai dengan <i>role</i> masing – masing | Sukses (Gambar 4.11) |
| 2 | Melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang salah | Memasukkan <i>username</i> dan <i>password</i> yang salah | User gagal <i>login</i> dan kembali memasukkan <i>username</i> dan <i>password</i> yang benar | Sukses (Gambar 4.12) |

b) Uji Coba Menu Manajemen *Coin*

Proses uji coba menu manajemen *coin* untuk menguji fungsi *create*, *read*, *update*, *delete*, dan pencarian sesuai dengan data inputan. Hasil uji coba dapat dilihat pada Tabel 4.8.

Tabel 4.8 Uji Coba Menu Manajemen *Coin*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|-----------------------------------|---|--|----------------------|
| 1 | Melakukan tambah data <i>coin</i> | Memasukan nama <i>coin</i> , kode <i>coin</i> , stok <i>coin</i> , dan <i>icon coin</i> kemudian menyimpannya dengan mengklik tombol Simpan | Data tersimpan ke sistem dan menampilkan data sesuai inputan | Sukses (Gambar 4.16) |
| 2 | Melakukan edit data <i>coin</i> | Mengganti data <i>coin</i> kemudian simpan dengan meng-klik tombol <i>update</i> | Perubahan data <i>coin</i> tersimpan | Sukses (Gambar 4.17) |
| 3 | Melakukan hapus data <i>coin</i> | Menghapus data <i>coin</i> dengan meng-klik tombol Hapus | Data yang dipilih terhapus | Sukses (Gambar 4.18) |

c) Uji Coba Menu Manajemen Paket

Proses uji coba menu manajemen *coin* untuk menguji fungsi *create*, *read*, *update*, *delete*, dan pencarian sesuai dengan data inputan. Hasil uji coba dapat dilihat pada Tabel 4.9.

Tabel 4.9 Uji Coba Menu Manajemen Paket

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|-----------------------------|--|--|----------------------|
| 1 | Melakukan tambah data paket | Memasukan kode <i>coin</i> , hashrate, dan harga kemudian menyimpannya dengan mengklik tombol Simpan | Data tersimpan ke sistem dan menampilkan data sesuai inputan | Sukses (Gambar 4.20) |
| 2 | Melakukan edit data paket | Mengganti data paket kemudian simpan dengan meng-klik tombol <i>update</i> | Perubahan data paket tersimpan | Sukses (Gambar 4.21) |
| 3 | Melakukan hapus data paket | Menghapus data paket dengan meng-klik tombol Hapus | Data yang dipilih terhapus | Sukses (Gambar 4.22) |

4.1.5. Sprint Review

Sprint Review pertama startup Alt.Mine.ID dengan fungsi admin dapat menggunakan fitur manajemen *coin*. Pada *Sprint Review* pertama ini terdapat *feedback* perbaikan dari *product owner*. Berikut adalah daftar *feedback* perbaikan yang telah diberikan oleh *project owner* dan disetujui oleh *product owner*.

Tabel 4.10 Perbaikan *Sprint Review* Pertama

| No | Story | Fungsi | Goal | Perbaikan |
|----------|-----------------------------|--|--|-----------------|
| 1 | Fitur manajemen <i>coin</i> | Fungsi penambahan <i>coin</i> baru | Dapat menambahkan <i>coin</i> baru | - <i>Header</i> |
| | | Fungsi <i>setting</i> paket <i>altcoin</i> | Dapat melakukan <i>insert, update, delete</i> paket <i>altcoin</i> | - |

4.2. Sprint 2 Manajemen Kontrak

Pada *sprint 2* akan dijelaskan tahapan pengerjaan manajemen kontrak beserta fungsi-fungsi yang ada di dalamnya.

4.2.1. Sprint Planning

Sprint planning kedua memiliki *sprint goal* yaitu dapat menjalankan fitur manajemen kontrak dengan fungsi setting kontrak *altcoin*. Untuk menyelesaikan *goal* tersebut maka dibuat:

1. *System flow* dengan fungsi login admin dan CRUD untuk memanajemen data kontrak.
2. Desain basis data menggunakan 1 tabel yaitu tabel kontrak.
3. Desain interface dengan menghasilkan desain halaman admin.

Desain uji coba menggunakan *blackbox testing* dengan melakukan pengujian menu manajemen kontrak.

4.2.2. Sprint Backlog

Pada sprint backlog manajemen kontrak terdapat 1 fungsi yang dibuat yaitu setting kontrak. Tabel *sprint backlog* 2 dapat dilihat pada Tabel 4.11 di bawah ini.

Tabel 4.11 Sprint Backlog Manajemen Kontrak

| No | Story | Fungsi | Goal |
|----|-------------------------|--|--|
| 1 | Fitur manajemen kontrak | Fungsi <i>setting</i> kontrak <i>altcoin</i> | Dapat melakukan <i>insert</i> , <i>update</i> , <i>delete</i> kontrak pembelian <i>altcoin</i> |

4.2.3. Daily Scrum



Gambar 4.23 Daily Scrum Sprint 2

Pada tahap ini fitur manajemen *coin* dan manajemen kontrak masuk ke dalam daftar dikerjakan (*in progress*) karena dalam sprint 2 ini selain mengerjakan

fitur manajemen kontrak dikerjakan juga *feedback* perbaikan yang telah didapat pada sprint 1 sebelumnya. Berikut adalah gambaran detail dari *daily scrum sprint 2*.

4.2.4. Pengerjaan Sprint

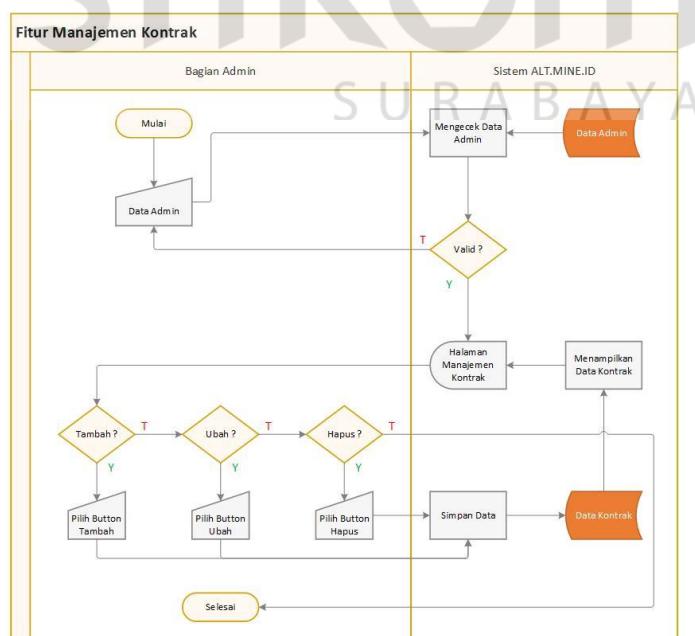
Dalam pengerjaan sprint ini dibuat desain sistem sampai dengan uji coba sistem yang telah jadi. Berikut adalah detail pengerjaan sistem pada sprint 2.

A. Desain Sistem

Setelah dilakukan perancangan selanjutnya dilakukan pembuatan desain sistem manajemen kontrak. Berikut adalah beberapa desain sistem yang dibuat:

1. System Flow

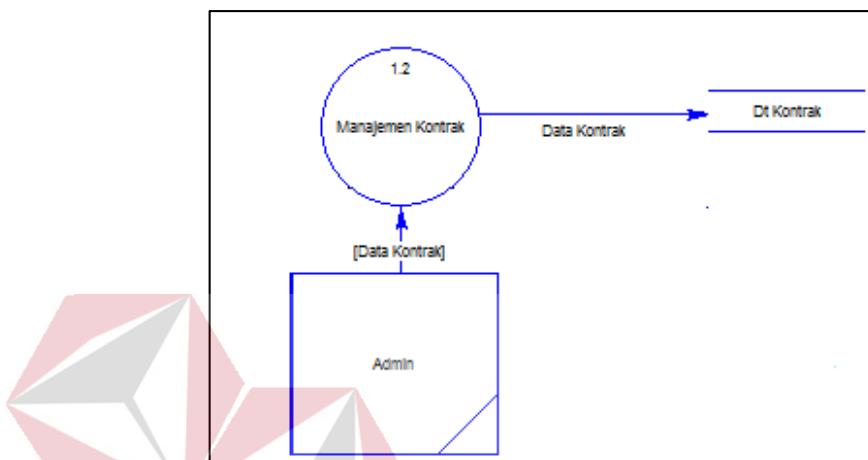
Pada Gambar 4.24 di bawah, alur proses manajemen kontrak dimulai oleh *admin* dengan cara melakukan *login*. Pada saat *login* berhasil, sistem menampilkan halaman manajemen kontrak. *Admin* dapat melakukan *create*, *insert*, *update*, *delete* dan tersimpan ke dalam *database*.



Gambar 4.24 System Flow Manajemen Kontrak

2. DFD Level 0 Manajemen Kontrak

Pada bagian ini mengalir untuk kegiatan pengolahan data kontrak, dan yang kemudian disimpan pada tabel master. Berikut DFD level 0 manajemen *kontrak* dapat dilihat pada Gambar 4.25 di bawah ini.



Gambar 4.25 DFD Level 0 Manajemen Kontrak

3. Struktur Basis Data

Pada sprint 2 ini dilakukan pembuatan tabel struktur basis data untuk fitur manajemen kontrak. Tabel pada struktur basis data yang dibentuk dapat dijelaskan sebagai berikut :

a) Tabel Kontrak

Nama Tabel : Kontrak

Primary Key : ID_Kontrak

Foreign Key : -

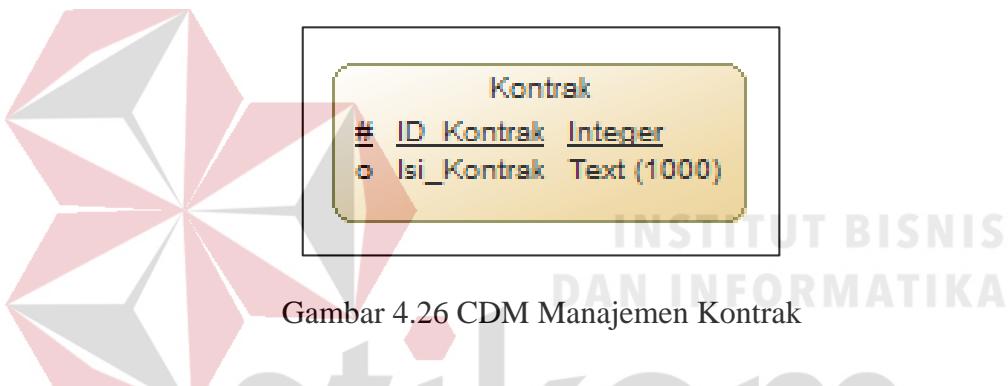
Fungsi : Menyimpan data kontrak

Tabel 4.12 Tabel Kontrak

| No | Field Name | Data Type | Length | Foreign Key | |
|----|-------------|-----------|--------|-------------|----------|
| | | | | On Field | On Table |
| 1 | ID_Kontrak | Integer | 11 | | |
| 2 | Isi_Kontrak | Text | - | | |

4. CDM

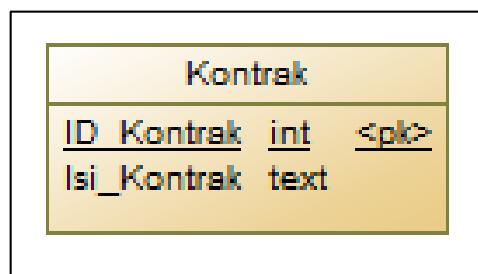
CDM merupakan rancangan awal konsep desain database yang nantinya akan di generate kedalam bentuk *Physical Data Model*. Berikut ini merupakan desain tabel dalam *sprint* manajemen kontrak.



Gambar 4.26 CDM Manajemen Kontrak

5. PDM

PDM merupakan hasil dari *generate* CDM. Secara keseluruhan PDM menggambarkan basis data yang telah jadi dan dapat di *generate script* kedalam *database server*. Berikut ini merupakan desain PDM dalam *sprint* manajemen kontrak.



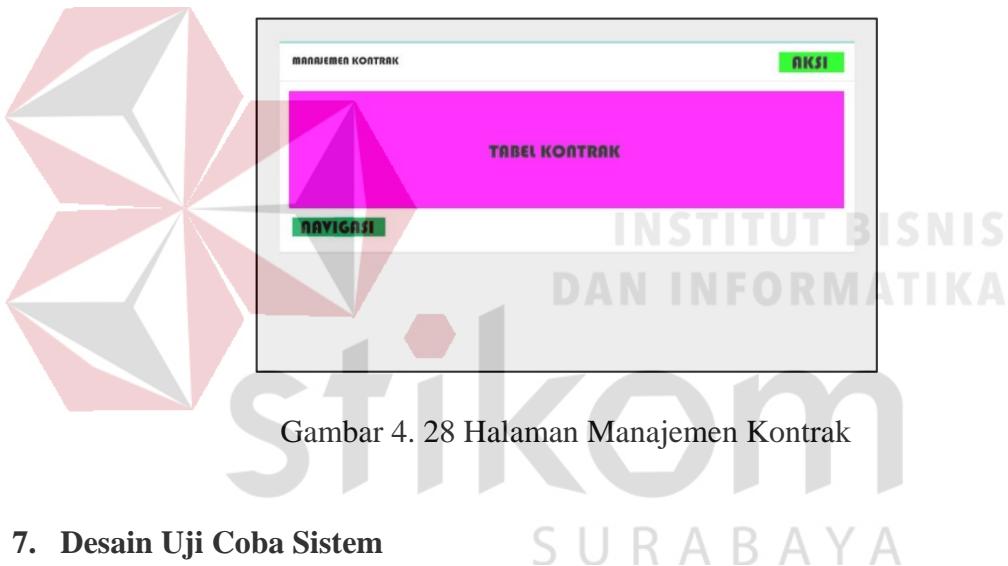
Gambar 4.27 PDM Manajemen Kontrak

6. Desain Interface

Pada sprint 2 ini dilakukan pembuatan desain interface bagian admin sub menu halaman manajemen kontrak. Berikut adalah desain *interface* yang dibuat sebagai acuan dalam implementasi aplikasi :

a) Desain Halaman Admin

Berikut adalah desain halaman *admin* (*sub menu* manajemen kontrak) yang digunakan sebagai acuan dalam implementasi halaman *admin*, dapat dilihat di bawah ini dan Gambar 4.28 halaman manajemen kontrak :



7. Desain Uji Coba Sistem

Pada sprint 2 ini dilakukan uji coba sistem pada halaman admin sub menu halaman manajemen kontrak. Detail dari desain uji coba dapat dilihat di bawah ini:

a) Manajemen Kontrak

Desain uji coba menu manajemen coin bertujuan untuk mengetahui apakah fungsi *Create, Read, Update, Delete* (CRUD) dan pencarian sesuai dengan data inputan. Desain uji coba dapat dilihat pada tabel 4.13.

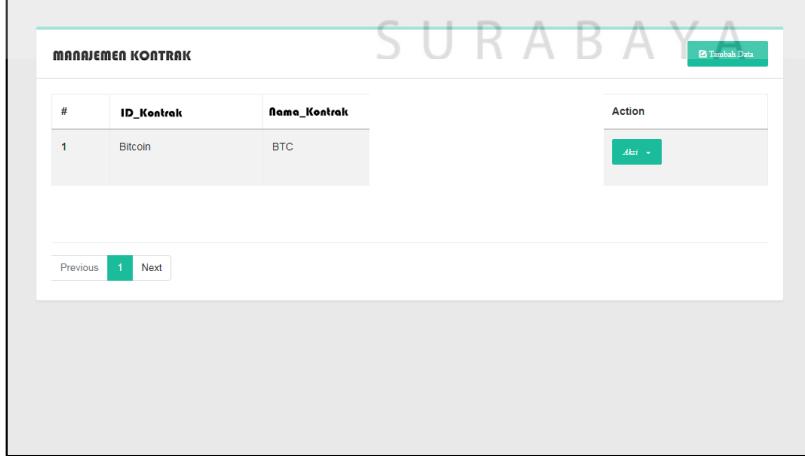
Tabel 4.13 Desain Uji Coba Manajemen Kontrak

| Test Case ID | Tujuan | Input | Output Yang Diharapkan |
|--------------|-------------------------------|--|--|
| 1 | Melakukan tambah data kontrak | Memasukan nama kontrak kemudian menyimpannya dengan mengklik tombol Simpan | Data tersimpan ke sistem dan menampilkan data sesuai inputan |
| 2 | Melakukan edit data kontrak | Mengganti data kontrak kemudian simpan dengan meng-klik tombol <i>update</i> | Perubahan data kontrak tersimpan |
| 3 | Melakukan hapus data kontrak | Menghapus data kontrak dengan meng-klik tombol Hapus | Data yang dipilih terhapus |

B. Implementasi Aplikasi

Setelah dilakukan pembuatan desain *interface* selanjutnya dilakukan implementasi terhadap aplikasi. Berikut adalah tampilan aplikasi yang sudah dibuat.:

a) Implementasi Halaman Admin



The screenshot shows a web application interface titled "MANAJEMEN KONTRAK". At the top right is a green button labeled "Tambah Data". Below the title is a table with three columns: "#", "ID_Kontrak", and "Nama_Kontrak". There is one row of data: #1, Bitcoin, BTC. To the right of the table is a "Action" column with a single button labeled "Aksi". At the bottom left of the page are navigation buttons: "Previous", a green "1", and "Next".

Gambar 4.29 Halaman Manajemen Kontrak

Halaman manajemen kontrak merupakan halaman yang berfungsi untuk mengelola data kontrak. Pada halaman ini dapat menambah, mengubah, serta menghapus data. Implementasi halaman manajemen kontrak dapat dilihat pada Gambar 4.29. Untuk menambah data baru, pilih tombol tambah data yang berada di atas, maka akan muncul tampilan seperti pada Gambar 4.30 di bawah.

The screenshot shows a web-based application interface for managing coins. On the left, there is a sidebar menu titled 'MENU ADMIN' with several options: 'Manajemen Koin', 'Manajemen Paket', 'Manajemen Kontrak', 'Data Bank', 'Setup Mining', 'Request Pembayaran', 'Request Withdrawal', and 'Request Verifikasi'. The 'Manajemen Koin' option is currently selected and highlighted in green. The main content area is titled 'MANAJEMEN COIN' and contains four input fields: 'Nama Koin' (Name), 'Kode' (Code), 'Stok' (Stock), and 'Icon' (Icon). Below these fields are two buttons: 'Reset' (red) and 'Save' (green). A small 'Tambah Data' button is located in the top right corner of the form area.

Gambar 4.30 Halaman Tambah Data Kontrak

Untuk mengubah data, pilih aksi *edit*, maka akan muncul tampilan seperti pada Gambar 4.31 di bawah.

This screenshot shows the same application interface as the previous one, but the 'Manajemen Paket' option in the sidebar is now selected and highlighted in green. The main content area is titled 'MANAJEMEN PAKET' and contains three input fields: 'Kode Coin' (Coin Code) with the value 'ETH', 'Hashrate' with the value '120', and 'Harga' (Price) with the value '10000000'. Below these fields are two buttons: 'Reset' (red) and 'Update' (green). A small 'Tambah Data' button is located in the top right corner of the form area.

Gambar 4.31 Halaman Edit Data Kontrak

Untuk menghapus data, pilih aksi *hapus*, maka akan muncul notifikasi seperti pada Gambar 4.32 di bawah.



Gambar 4.32 Notifikasi Hapus Data Paket

C. Ujicoba dan Evaluasi Sistem

Pada sprint 2 dilakukan ujicoba dan evaluasi menu manajemen kontrak.

Berikut adalah hasil ujicoba dan evaluasi sistem yang telah dibuat :

a) Uji Coba Menu Manajemen Kontrak

Proses uji coba menu manajemen *kontrak* untuk menguji fungsi *create*, *read*, *update*, *delete*, dan pencarian sesuai dengan data inputan. Hasil uji coba dapat dilihat pada Tabel 4.14.

Tabel 4.14 Uji Coba Menu Manajemen Kontrak

| Test Case ID | Tujuan | Input | Output Yang Diharapkan | Status |
|--------------|-------------------------------|--|--|----------------------|
| 1 | Melakukan tambah data kontrak | Memasukan nama kontrak kemudian menyimpannya dengan mengklik tombol Simpan | Data tersimpan ke sistem dan menampilkan data sesuai inputan | Sukses (Gambar 4.30) |
| 2 | Melakukan edit data kontrak | Mengganti data kontrak kemudian simpan dengan meng-klik tombol update | Perubahan data kontrak tersimpan | Sukses (Gambar 4.31) |
| 3 | Melakukan hapus data kontrak | Menghapus data kontrak dengan meng-klik tombol Hapus | Data yang dipilih terhapus | Sukses (Gambar 4.32) |

4.2.5. Sprint Review

Sprint Review kedua menghasilkan aplikasi startup Alt.Mine.ID yang mana admin dapat menggunakan fitur manajemen kontrak. Pada *Sprint Review* kedua ini tidak terdapat *feedback* perbaikan dari *product owner*.

4.3. Sprint 3 Sign-Up Member

Pada sprint 3 akan dijelaskan tahapan penggeraan fitur *sign-up member* beserta fungsi-fungsi yang ada di dalamnya.

4.3.1. Sprint Planning

Sprint planning ketiga memiliki *sprint goal* yaitu dapat menjalankan fitur free sign up member dengan fungsi verifikasi, *edit wallet*, dan *upload* foto. Untuk menyelesaikan goal tersebut maka dibuat :

1. System flow dengan fungsi registrasi agar customer dapat terdaftar ke dalam sistem.
2. Desain basis data menggunakan 3 tabel yaitu tabel customer, tabel *wallet*, dan tabel verif.
3. Desain interface dengan menghasilkan desain halaman login, halaman customer, dan halaman admin.
4. Desain uji coba menggunakan blackbox testing dengan melakukan pengujian *form login*, *form register*, menu *account*, dan menu *request verifikasi*.

4.3.2. Sprint Backlog

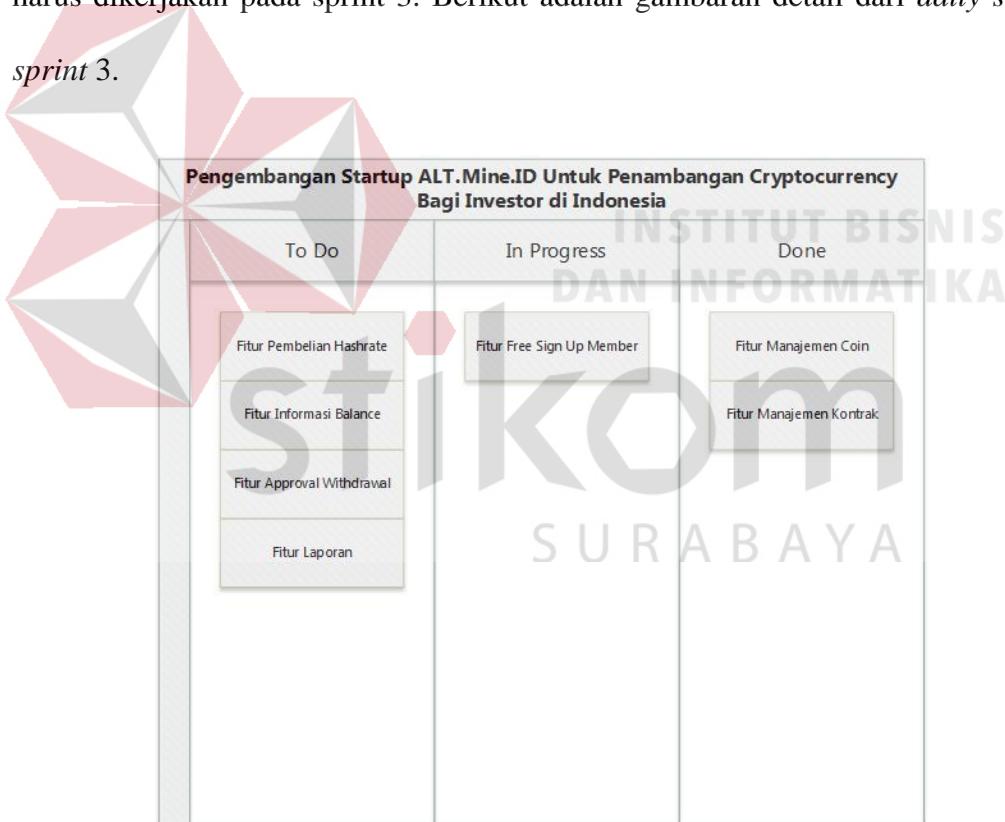
Pada *sprint backlog* *free sign-up member* terdapat 2 fungsi yang dibuat yaitu verifikasi dan *edit wallet*. Tabel *sprint backlog* 3 dapat dilihat pada Tabel 4.15 di bawah ini.

Tabel 4.15 Sprint Backlog Free Sign-Up Member

| No | Story | Fungsi | Goal |
|----|----------------------------------|---------------------------|--|
| 1 | Fitur <i>free sign-up member</i> | Fungsi verifikasi | Dapat melakukan verifikasi <i>user</i> |
| | | Fungsi <i>edit wallet</i> | Dapat melakukan <i>update</i> data <i>wallet</i> |

4.3.3. Daily Scrum

Pada tahap ini fitur *sign-up member* masuk ke dalam daftar dikerjakan (in progress) karena dalam *sprint 2* sebelumnya tidak ada *feedback* perbaikan yang harus dikerjakan pada *sprint 3*. Berikut adalah gambaran detail dari *daily scrum sprint 3*.



Gambar 4. 33 Daily Scrum Sprint 3

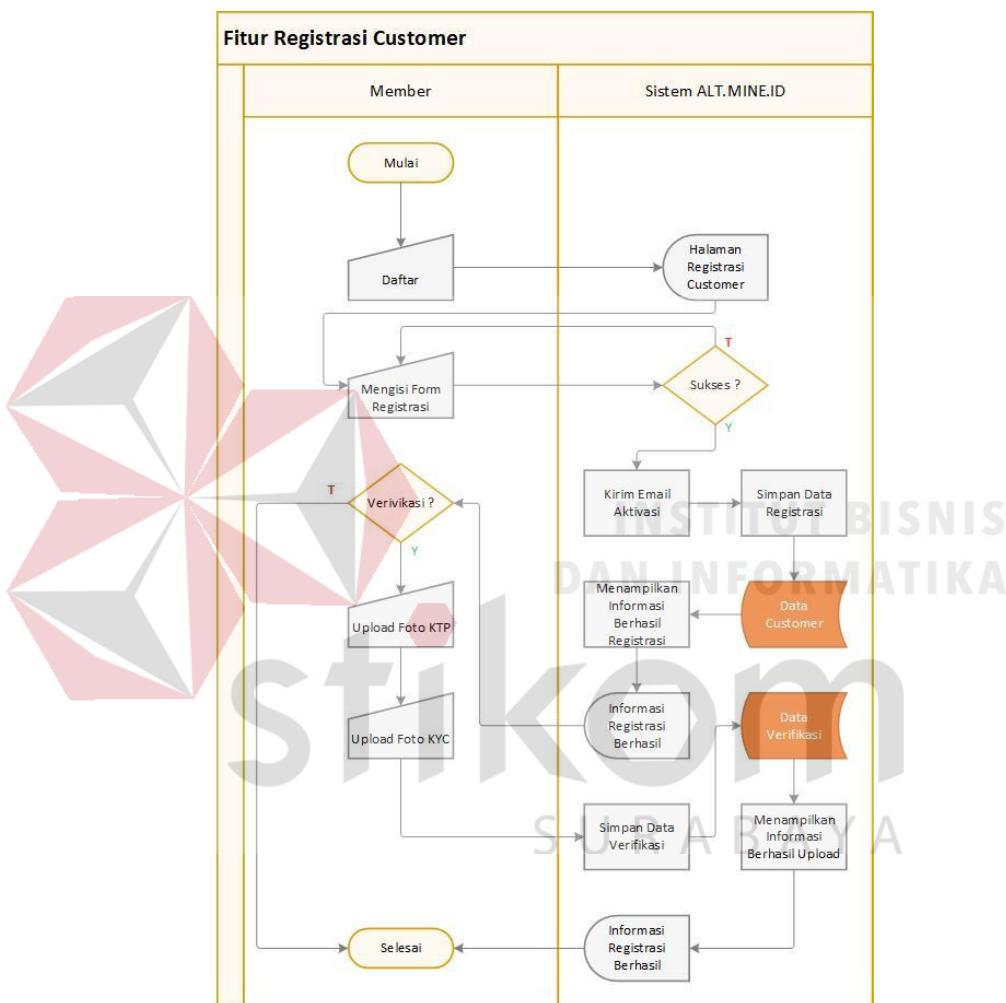
4.3.4. Pengerjaan Sprint

Dalam pengerjaan sprint ini dibuat desain sistem sampai dengan uji coba sistem yang telah jadi. Berikut adalah detail pengerjaan sistem pada sprint 3.

A. Desain Sistem

Setelah dilakukan perancangan selanjutnya dilakukan pembuatan desain sistem free sign up member. Berikut adalah beberapa desain sistem yang dibuat:

1. System Flow



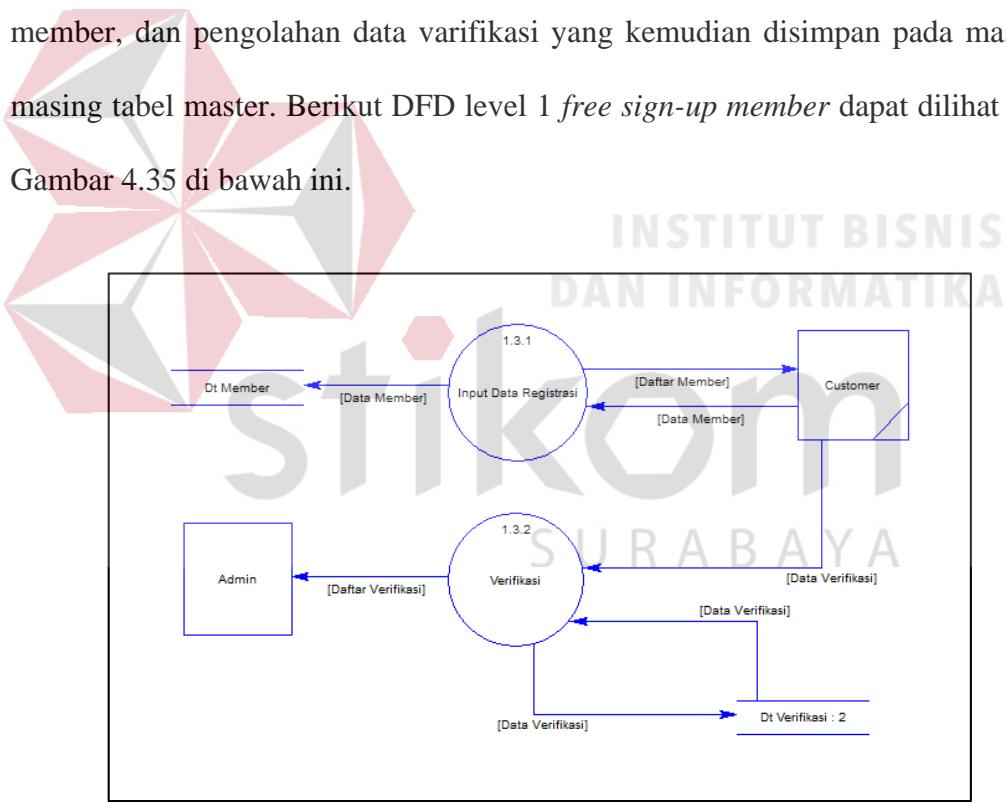
Gambar 4.34 System Flow Registrasi Customer

Pada Gambar 4.34 di bawah, pengguna dapat melakukan registrasi ke dalam sistem dengan cara menekan tombol daftar pada halaman awal. Sistem akan menampilkan halaman registrasi *member* yang megarahkan member ke form registrasi kemudian sistem akan melakukan pengecekan kesalahan input dari form

registrasi. Apabila semua kondisi telah terpenuhi sistem akan mengirim email aktivasi dan melakukan simpan data ke database. Jika *customer* melakukan verifikasi maka sistem akan mengalihkan ke halaman upload foto KTP dan KYC. Sistem akan melakukan proses simpan ke dalam database verifikasi dan menampilkan informasi bahwa simpan data berhasil.

2. DFD Level 1 *Free Sign-Up Member*

DFD level 1 merupakan diagram yang menguraikan detail proses yang ada pada DFD level 0. Pada bagian ini mengalir untuk kegiatan pengolahan data member, dan pengolahan data varifikasi yang kemudian disimpan pada masing-masing tabel master. Berikut DFD level 1 *free sign-up member* dapat dilihat pada Gambar 4.35 di bawah ini.



Gambar 4.35 DFD Level 1 *Free Sign-Up Member*

3. Struktur Basis Data

Pada sprint 3 ini dilakukan pembuatan tabel struktur basis data untuk fitur free sign up. Tabel pada struktur basis data yang dibentuk dapat dijelaskan sebagai berikut :

a) Tabel Customer

Nama Tabel : Customer

Primary Key : ID_Customer

Foreign Key : -

Fungsi : Menyimpan data *customer*

Tabel 4.16 Tabel *Customer*

| No | <i>Field Name</i> | <i>Data Type</i> | <i>Length</i> | <i>Foreign Key</i> | |
|----|-------------------|------------------|---------------|--------------------|-----------------|
| | | | | <i>On Field</i> | <i>On Table</i> |
| 1 | ID_Customer | Integer | 11 | | |
| 2 | Username | VarChar | 20 | | |
| 3 | Password | VarChar | 32 | | |
| 4 | Nama | VarChar | 30 | | |
| 5 | Email | VarChar | 30 | | |
| 6 | Alamat | Text | - | | |
| 7 | Telp | VarChar | 13 | | |
| 8 | Tgl_Daftar | Date | - | | |
| 9 | Sts_Login | VarChar | 10 | | |
| 10 | Sts_Verif | VarChar | 10 | | |

b) Tabel *Wallet*

Nama Tabel : *Wallet*

Primary Key : ID_Wallet

Foreign Key : -

Fungsi : Menyimpan alamat *wallet*

Tabel 4.17 Tabel Wallet

| No | Field Name | Data Type | Length | Foreign Key | |
|-----------|-------------------|------------------|---------------|--------------------|-----------------|
| | | | | <i>On Field</i> | <i>On Table</i> |
| 1 | ID_Wallet | Integer | 11 | | |
| 2 | ETH_Wallet | VarChar | 200 | | |
| 3 | XMR_Wallet | VarChar | 200 | | |

c) Tabel Verif

Nama Tabel : Verif

Primary Key : ID_Verif

Foreign Key : -

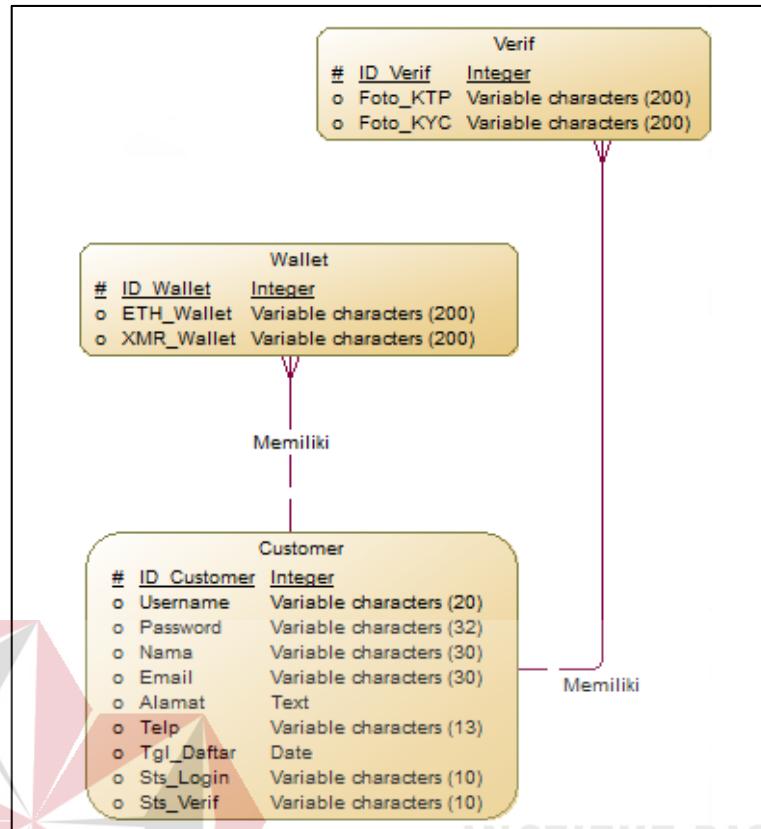
Fungsi : Menyimpan data verifikasi *customer*

Tabel 4.18 Tabel Verif

| No | Field Name | Data Type | Length | Foreign Key | |
|-----------|-------------------|------------------|---------------|--------------------|-----------------|
| | | | | <i>On Field</i> | <i>On Table</i> |
| 1 | ID_Verif | Integer | 11 | | |
| 2 | Foto_KTP | VarChar | 200 | | |
| 3 | Foto_KYC | VarChar | 200 | | |

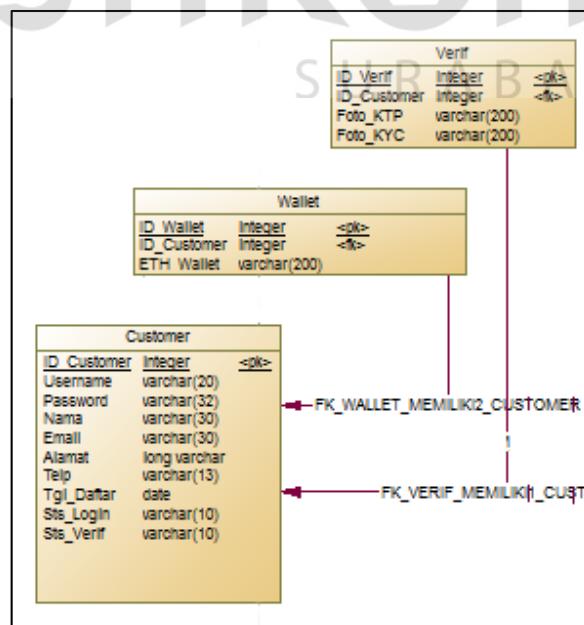
4. CDM

CDM merupakan rancangan awal konsep desain database yang nantinya akan di generate kedalam bentuk Physical Data Model. Berikut ini merupakan desain relasi antar tabel dalam sprint *free sign-up member*.



Gambar 4.36 CDM Free Sign-Up Member

5. PDM



Gambar 4.37 PDM Free Sign-Up Member

PDM merupakan hasil dari generate CDM. Secara keseluruhan PDM menggambarkan basis data yang telah jadi dan dapat di *generate script* kedalam *database server*. Berikut ini merupakan desain PDM dalam *sprint free sign-up member*.

6. Desain Interface

Pada *sprint 3* ini dilakukan pembuatan desain *interface* halaman *customer*, *login customer*, halaman *customer* sub menu *account*, dan *registrasi customer*. Berikut adalah desain *interface* yang dibuat sebagai acuan dalam implementasi aplikasi.:

a) Desain Halaman *Login*

Berikut adalah desain halaman *login customer* yang digunakan sebagai acuan dalam implementasi. Halaman *login customer* dapat dilihat pada Gambar 4.38 di bawah.

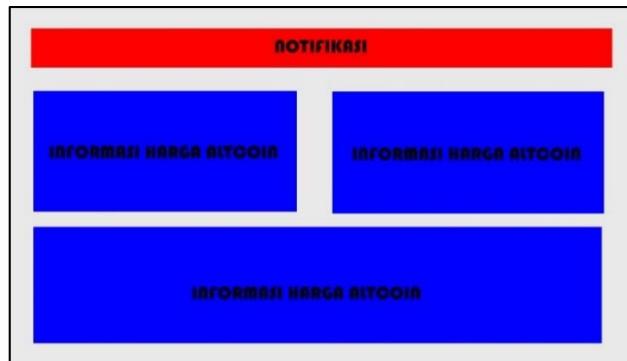


Gambar 4.38 Desain Halaman *Login Customer*

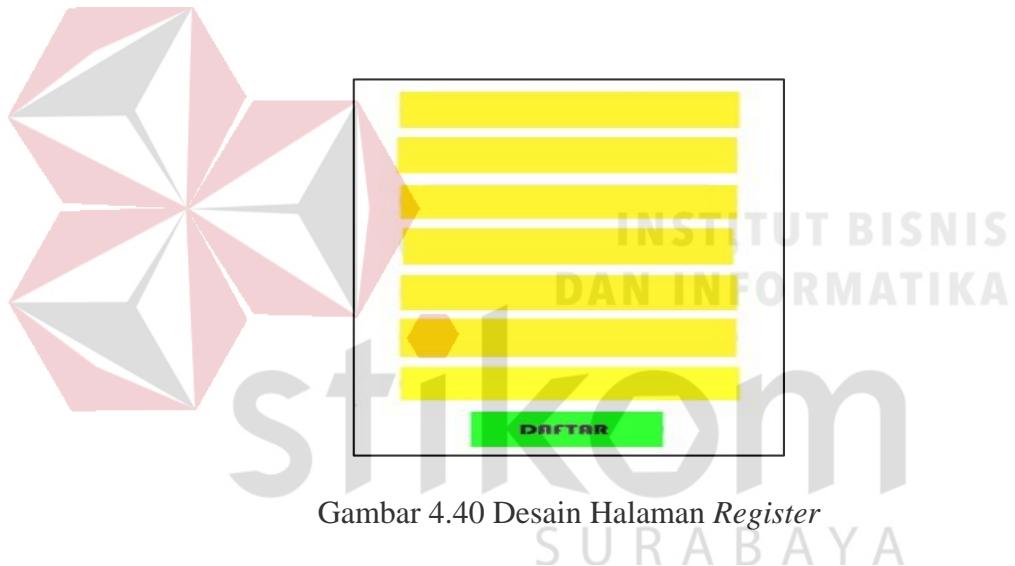
b) Desain Halaman *Customer*

Berikut adalah desain halaman *customer* (*sub menu account*, dan *registrasi customer*) yang digunakan sebagai acuan dalam implementasi halaman

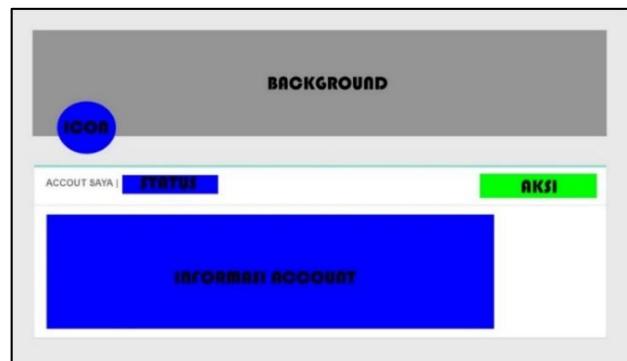
customer, dapat dilihat di bawah ini Gambar 4.39 halaman *customer*, Gambar 4.40 halaman registrasi, dan Gambar 4.41 halaman *account* :



Gambar 4.39 Desain Halaman *Customer*



Gambar 4.40 Desain Halaman *Register*



Gambar 4.41 Desain Halaman *Account*

c) Desain Halaman Admin

Berikut adalah desain halaman *admin* (sub menu *request*) yang digunakan sebagai acuan dalam implementasi halaman *admin*, dapat dilihat di bawah ini

Gambar 4.42 halaman *request* verifikasi :



Gambar 4.42 Halaman *Request Verifikasi*

7. Uji Coba Sistem

Pada sprint 3 ini dilakukan uji coba sistem pada halaman login customer, halaman customer sub menu *account*, dan registrasi *customer*. Detail dari desain uji coba dapat dilihat di bawah ini :

a) Desain Uji Coba *Form Login*

Desain uji coba *login* berfungsi untuk menguji apakah fungsi *login* dapat berjalan sesuai yang diharapkan. Desain uji coba *login* dapat dilihat pada Tabel 4.19.

Tabel 4.19 Desain Uji Coba *Form Login*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|------------------------|--|---|
| 1 | Melakukan <i>Login</i> | Memasukkan <i>username</i> dan <i>password</i> | Login berhasil dan masuk ke halaman utama sesuai dengan <i>role</i> masing – masing |

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|--|---|--|
| 2 | Melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang salah | Memasukkan <i>username</i> dan <i>password</i> yang salah | <i>User</i> gagal <i>login</i> dan kembali memasukkan <i>username</i> dan <i>password</i> yang benar |

b) Desain Uji Coba *Form Register*

Desain uji coba *register* berfungsi untuk menguji apakah fungsi *register* dapat berjalan sesuai yang diharapkan. Desain uji coba *register* dapat dilihat pada Tabel 4.20.

Tabel 4.20 Desain Uji Coba *Form Register*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|---------------------------|--|--|
| 1 | Melakukan <i>Register</i> | Memasukkan <i>username</i> , <i>password</i> , <i>email</i> , <i>nama</i> , <i>alamat</i> , <i>no_telp</i> , dan jenis kelamin | Data tersimpan ke sistem dan menampilkan data <i>customer</i> sesuai inputan |

c) Desain Uji Coba Menu *Account*

Desain uji coba menú *account* berfungsi untuk mengecek apakah dapat melakukan manajemen data *customer* dengan melakukan *edit* untuk merubah data. Desain uji coba menú *account* dapat dilihat pada Tabel 4.21.

Tabel 4.21 Desain Uji Coba Menu *Account*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|-----------------------------|--|---|
| 1 | Melihat data <i>account</i> | Memilih menu request <i>withdrawal</i> | Data <i>withdraw balance</i> akan muncul berdasarkan request <i>withdraw</i> yang telah dilakukan |

| Test Case ID | Tujuan | Input | Output Yang Diharapkan |
|--------------|------------------------------------|---|---|
| 2 | Melakukan edit data <i>account</i> | Mengganti data <i>account</i> dengan mengklik tombol <i>update</i> | Perubahan data tersimpan |
| 3 | Melakukan edit <i>icon profile</i> | Mengganti data <i>icon profile</i> dengan mengklik icon | Perubahan data <i>icon profile</i> tersimpan |
| 4 | Melakukan verifikasi akun | Memilih menu verifikasi dan <i>upload</i> foto ktp dan kyc | Data tersimpan dan menampilkan informasi status verifikasi. |
| 5 | Melakukan edit <i>wallet</i> | Mengganti data <i>icon profile</i> dengan mengklik menu <i>wallet</i> | Perubahan data <i>wallet</i> tersimpan |

d) Desain Uji Coba Menu *Request Verifikasi*

Desain uji coba menu *request verifikasi* bertujuan untuk mengetahui apakah dapat melakukan persetujuan maupun menolak data verifikasi *customer*. Desain uji coba dapat dilihat pada tabel 4.22.

Tabel 4.22 Desain Uji Coba Menu Request Verifikasi

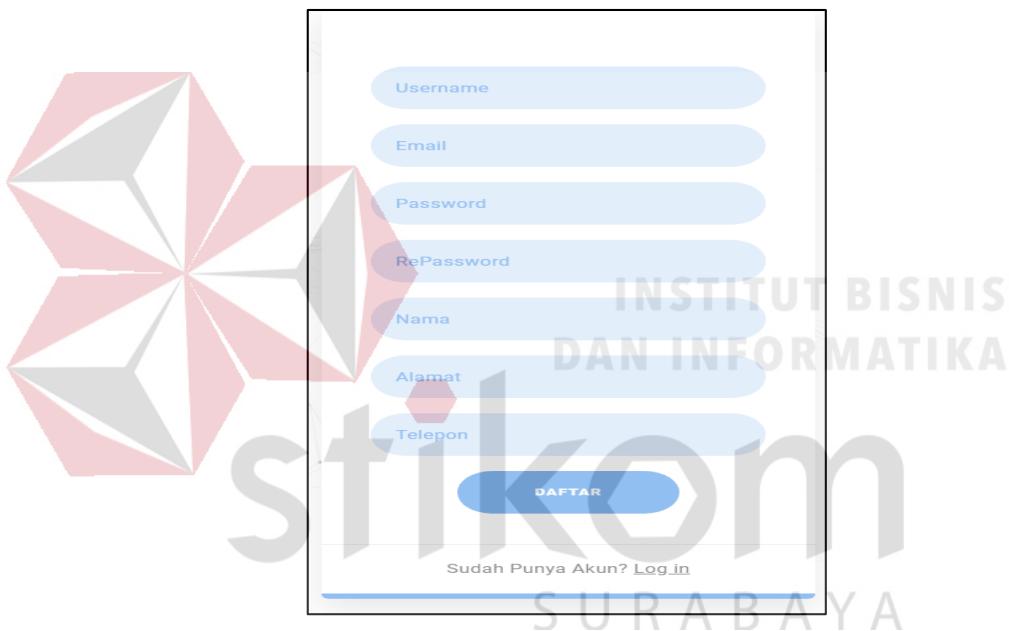
| Test Case ID | Tujuan | Input | Output Yang Diharapkan |
|--------------|--|---|---|
| 1 | Meng-approve data verifikasi <i>customer</i> | Mengklik tombol verifikasi dan merubah status menjadi <i>verified</i> | Status <i>customer</i> berhasil berubah menjadi <i>verified</i> |
| 2 | Me-reject data verifikasi <i>customer</i> | Mengklik tombol verifikasi dan merubah status menjadi <i>reject</i> | Status <i>customer</i> berhasil berubah menjadi <i>rejected</i> |

B. Implementasi Aplikasi

Setelah dilakukan pembuatan desain *interface* selanjutnya dilakukan implementasi terhadap aplikasi. Berikut adalah tampilan aplikasi yang sudah dibuat.:

a) Implementasi Halaman Registrasi

Halaman registrasi merupakan halaman yang berfungsi untuk melakukan pendaftaran. Implementasi halaman *customer* dapat dilihat pada Gambar 4.43.

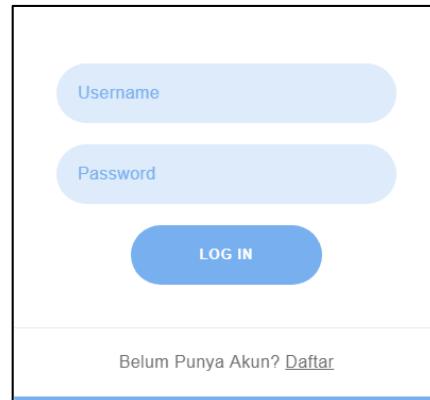


Gambar 4.43 Implementasi Halaman *Register*

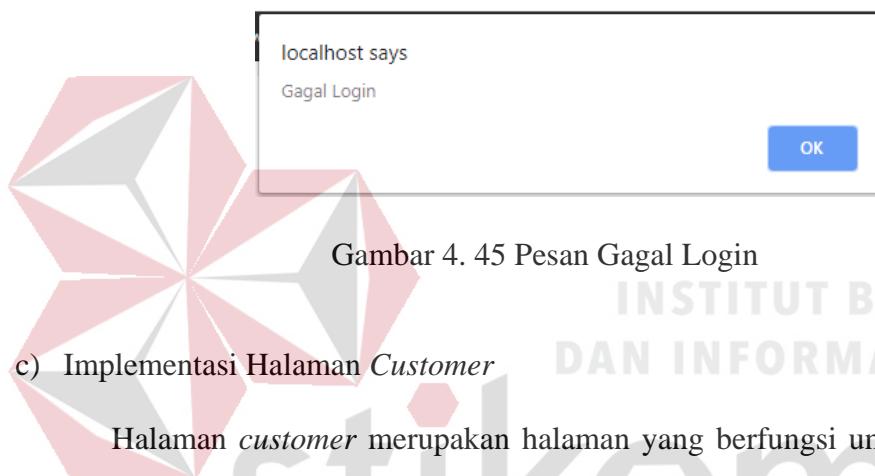
b) Implementasi Halaman *Login*

Halaman *login* merupakan halaman yang muncul pertama saat aplikasi dibuka. Halaman ini berfungsi untuk akses *customer*, untuk mengakses aplikasi.

Gambar 4.44 halaman *login customer*, dan Gambar 4.45 error pesan *login*.



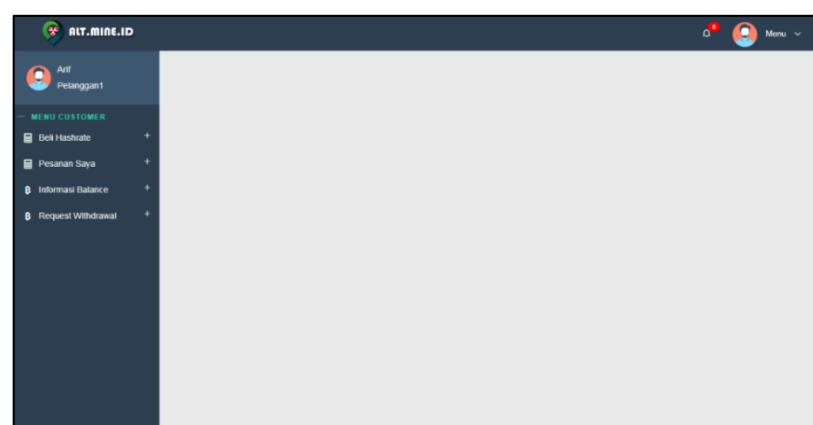
Gambar 4. 44 Halaman Login Customer



Gambar 4. 45 Pesan Gagal Login

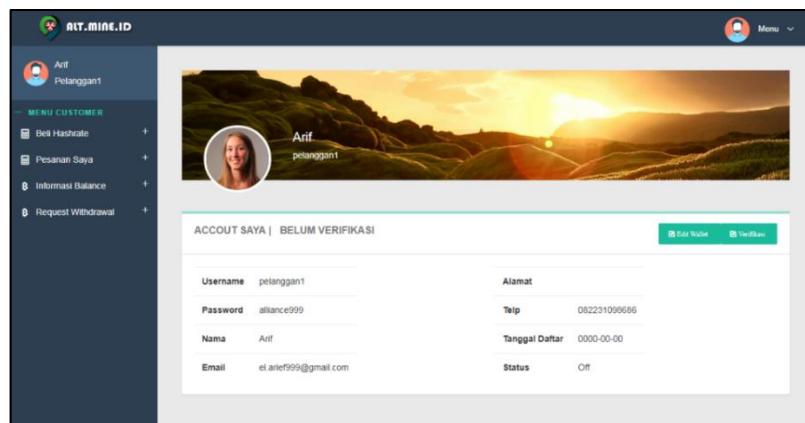
c) Implementasi Halaman *Customer*

Halaman *customer* merupakan halaman yang berfungsi untuk melakukan pembelian dan pembayaran. Implementasi halaman *customer* dapat dilihat pada Gambar 4.46.



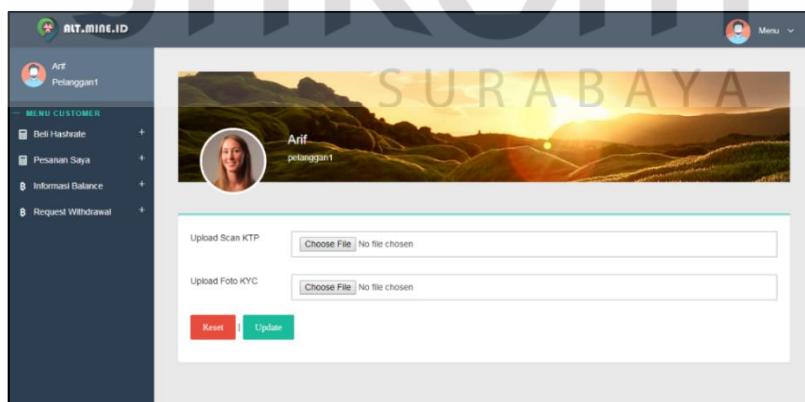
Gambar 4.46 Halaman *Customer*

Untuk melihat data account pilih *sub menu profile*, maka akan muncul tampilan seperti pada Gambar 4.47 di bawah.

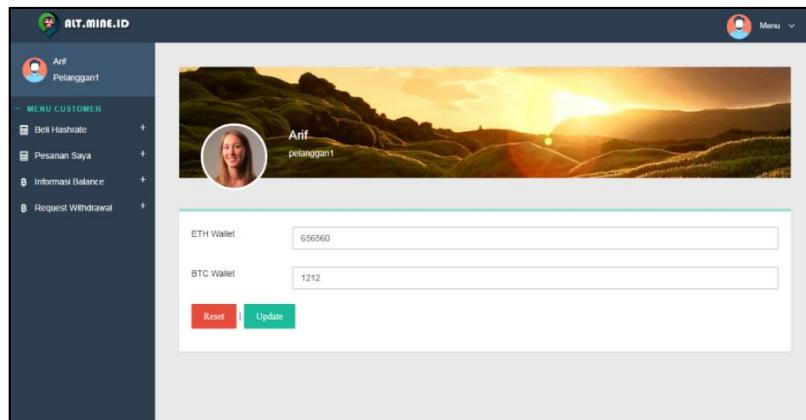
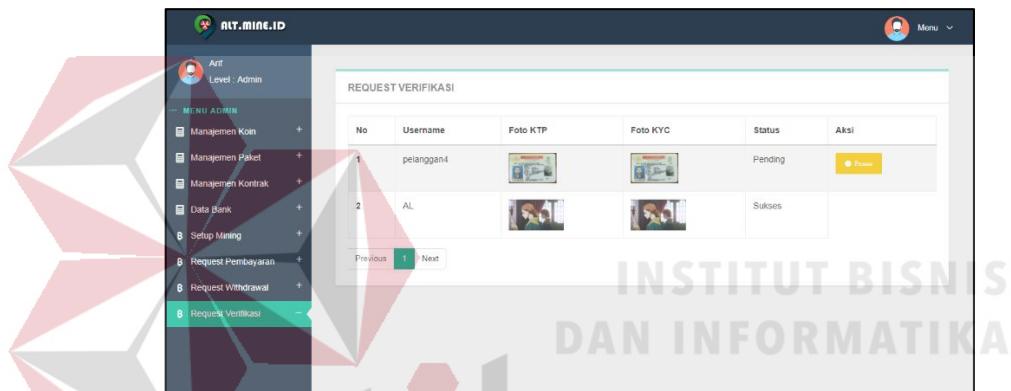


Gambar 4.47 Halaman Account

Untuk melakukan verifikasi pilih tombol verifikasi pada halaman *account*, maka akan muncul tampilan seperti pada Gambar 4.48 di bawah. Untuk melakukan *edit wallet* pilih tombol *edit wallet* pada halaman *account*, maka akan muncul tampilan seperti pada Gambar 4.49 di bawah.

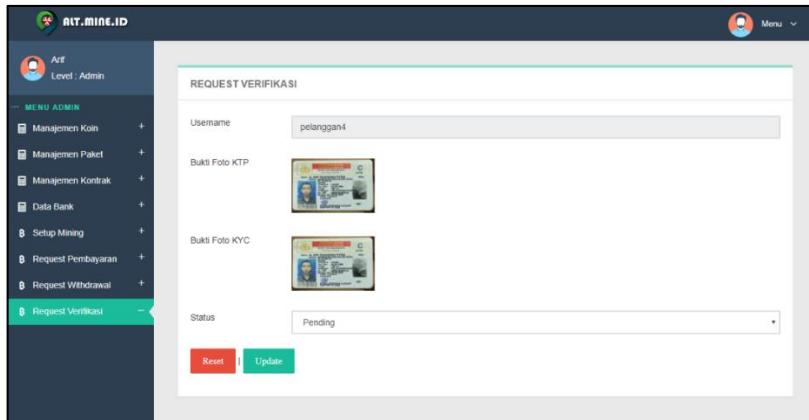


Gambar 4. 48 Halaman Verifikasi

Gambar 4.49 Halaman *Edit Wallet*Gambar 4.50 Halaman *Request Verifikasi*

d) Implementasi Halaman Admin

Halaman *request verifikasi* merupakan sub menu halaman yang berfungsi untuk mengelola data verif. Pada halaman ini dapat mengubah data verif. Implementasi halaman *request verifikasi* dapat dilihat pada Gambar 4.50 di atas. Untuk mengubah data, pilih aksi *edit*, maka akan muncul tampilan seperti pada Gambar 4.51.



Gambar 4.51 Halaman Verifikasi *Request Verifikasi*

C. Ujicoba dan Evaluasi Sistem

Pada sprint 3 dilakukan ujicoba dan evaluasi *form login customer*, *form register*, menu *account*, dan menu *request verifikasi*. Berikut adalah hasil ujicoba dan evaluasi sistem yang telah dibuat :

a) Uji Coba *Form Login Customer*

Proses uji coba *login customer* dimulai dengan memasukkan *username* dan *password* yang telah di daftarkan. Proses *login* dinyatakan berhasil apabila sistem telah masuk ke halaman utama. Hasil uji coba dapat dilihat pada Tabel 4.23.

Tabel 4.23 Uji Coba *Form Login Customer*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|--|---|---|----------------------|
| 1 | Melakukan <i>Login</i> | Memasukkan <i>username</i> dan <i>password</i> | Login berhasil dan masuk ke halaman utama sesuai dengan <i>role</i> masing – masing | Sukses (Gambar 4.44) |
| 2 | Melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang salah | Memasukkan <i>username</i> dan <i>password</i> yang salah | User gagal <i>login</i> dan kembali memasukkan <i>username</i> dan <i>password</i> yang benar | Sukses (Gambar 4.45) |

b) Uji Coba *Form Registrasi Customer*

Proses uji coba *registrasi customer* dimulai dengan memasukkan *username*, *password*, *email*, *nama*, *alamat*, *no_telp*, dan *jenis kelamin* yang telah di daftarkan. Proses registrasi dinyatakan berhasil apabila *customer* telah mendapatkan email notifikasi. Hasil uji coba dapat dilihat pada Tabel 4.24.

Tabel 4.24 Uji Coba Form *Registrasi Customer*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|---------------------------|---|--|----------------------|
| 1 | Melakukan <i>Register</i> | Memasukkan <i>username</i> , <i>password</i> , <i>email</i> , <i>nama</i> , <i>alamat</i> , <i>no_telp</i> , dan <i>jenis kelamin</i> | Data tersimpan ke sistem dan menampilkan data <i>customer</i> sesuai inputan | Sukses (Gambar 4.43) |

c) Uji Coba Menu *Account*

Proses uji coba dimulai yaitu manajemen data *customer* sesuai dengan melakukan *edit* untuk merubah dan mencari data sesuai inputan. Hasil uji coba dapat dilihat pada Tabel 4.25.

Tabel 4.25 Uji Coba Menu *Account*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|------------------------------------|--|---|----------------------|
| 1 | Melihat data <i>account</i> | Memilih menu <i>account</i> | Data <i>account</i> akan muncul berdasarkan menu yang telah dipilih | Sukses (Gambar 4.47) |
| 2 | Melakukan edit data <i>account</i> | Mengganti data <i>account</i> dengan mengklik tombol <i>update</i> | Perubahan data <i>account</i> tersimpan | Sukses (Gambar 4.47) |
| 3 | Melakukan edit <i>icon profile</i> | Mengganti data <i>icon profile</i> | Perubahan data <i>icon profile</i> tersimpan | Sukses (Gambar 4.47) |

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|------------------------------|---|---|----------------------|
| | | dengan mengklik icon | | |
| 4 | Melakukan verifikasi akun | Memilih menu verifikasi dan upload foto ktp dan kyc | Data tersimpan dan menampilkan informasi status verifikasi. | Sukses (Gambar 4.48) |
| 5 | Melakukan <i>edit wallet</i> | Mengganti data <i>wallet</i> dengan mengklik menu <i>wallet</i> | Perubahan data <i>wallet</i> tersimpan | Sukses (Gambar 4.49) |

d) Uji Coba Menu *Request Verifikasi*

Proses uji coba dimulai dengan melakukan persetujuan maupun menolak data verifikasi *customer* dan mencari sesuai dengan inputan. Hasil uji coba dapat dilihat pada Tabel 4.26.

Tabel 4.26 Uji Coba Menu *Request Verifikasi*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|--|---|---|----------------------|
| 1 | Meng-approve data verifikasi <i>customer</i> | Mengklik tombol verifikasi dan merubah status menjadi <i>verified</i> | Status <i>customer</i> berhasil berubah menjadi <i>verified</i> | Sukses (Gambar 4.51) |
| 2 | Me-reject data verifikasi <i>customer</i> | Mengklik tombol verifikasi dan merubah status menjadi <i>reject</i> | Status <i>customer</i> berhasil berubah menjadi <i>rejected</i> | Sukses (Gambar 4.51) |

4.3.5. *Sprint Review*

Sprint Review ketiga menghasilkan aplikasi startup Alt.Mine.ID yang mana admin dapat melakukan pendaftaran. Pada *Sprint Review* ketiga ini tidak terdapat *feedback* perbaikan dari *product owner*.

4.4. *Sprint 4 Pembelian Hashrate*

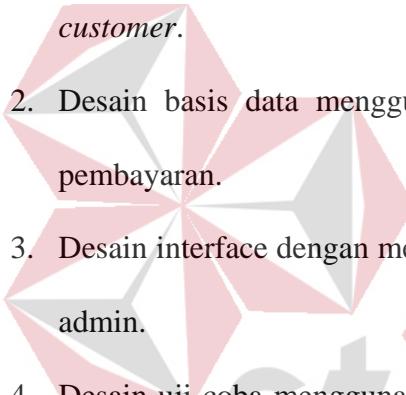
Pada sprint 4 akan dijelaskan tahapan pengerjaan fitur pembelian *hashrate* beserta fungsi-fungsi yang ada di dalamnya.

4.4.1. *Sprint Planning*

Sprint planning keempat memiliki *sprint goal* yaitu dapat menjalankan fitur pembelian hashrate dengan fungsi pembelian *hashrate* dan konfirmasi pembayaran.

Untuk menyelesaikan goal tersebut maka dibuat:

1. System flow dengan fungsi pembelian *hashrate* yang dapat dilakukan oleh

- 
- customer.*
2. Desain basis data menggunakan 2 tabel yaitu tabel pemesanan dan tabel pembayaran.
 3. Desain interface dengan menghasilkan desain halaman customer, dan halaman admin.
 4. Desain uji coba menggunakan blackbox testing dengan melakukan pengujian menu pembelian, menu pesanan, menu request pembayaran, dan menu setup *mining*.

4.4.2. *Sprint Backlog*

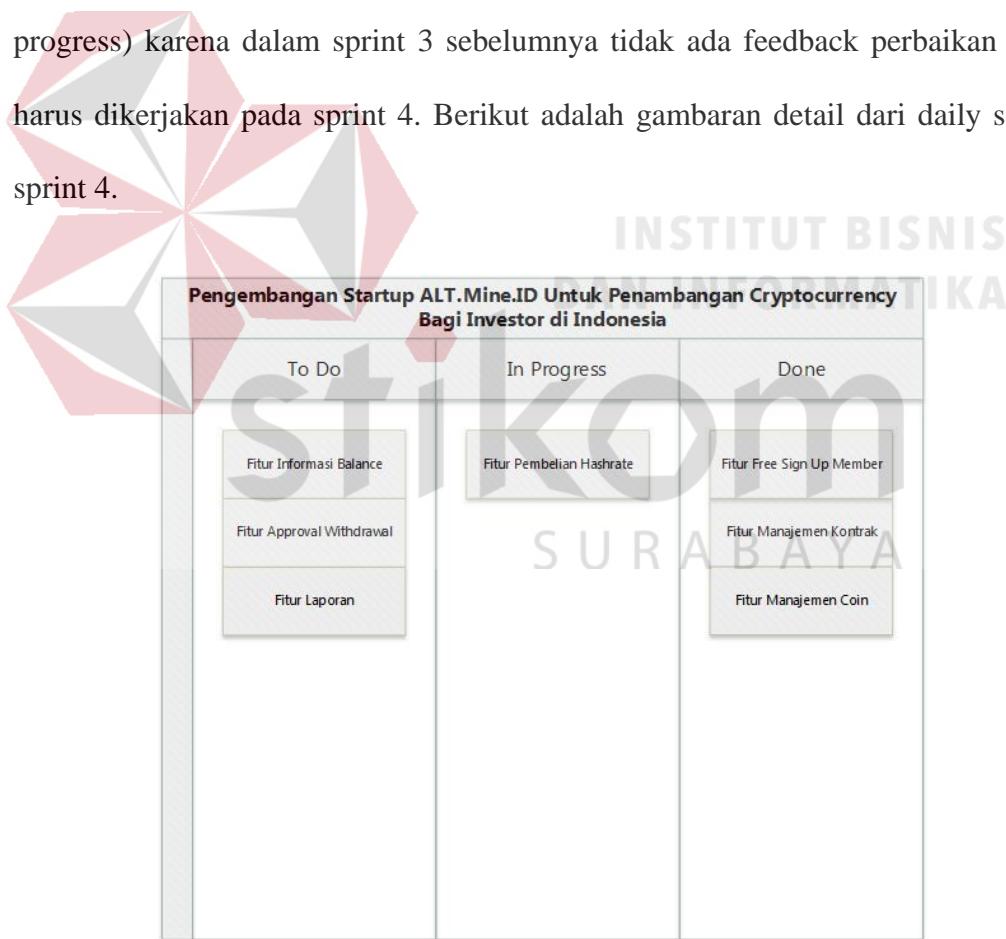
Pada sprint backlog pembelian hashrate terdapat 2 fungsi yang dibuat yaitu pembelian hashrate dan konfirmasi pembayaran. Tabel *sprint backlog* 4 dapat dilihat pada Tabel 3.20 di bawah ini.

Tabel 4. 27 *Sprint Backlog Pembelian Hashrate*

| No | Story | Fungsi | Goal |
|----|---------------------------------|--|---|
| 1 | Fitur pembelian <i>hashrate</i> | Fungsi pembelian <i>hashrate</i> berdasarkan paket | Dapat melakukan pembelian <i>hashrate</i> |
| | | Fungsi konfirmasi pembayaran | Dapat melakukan konfirmasi pembayaran |

4.4.3. Daily Scrum

Pada tahap ini pembelian hashrate masuk ke dalam daftar dikerjakan (in progress) karena dalam sprint 3 sebelumnya tidak ada feedback perbaikan yang harus dikerjakan pada sprint 4. Berikut adalah gambaran detail dari daily scrum sprint 4.

Gambar 4. 52 *Daily Scrum Sprint 4*

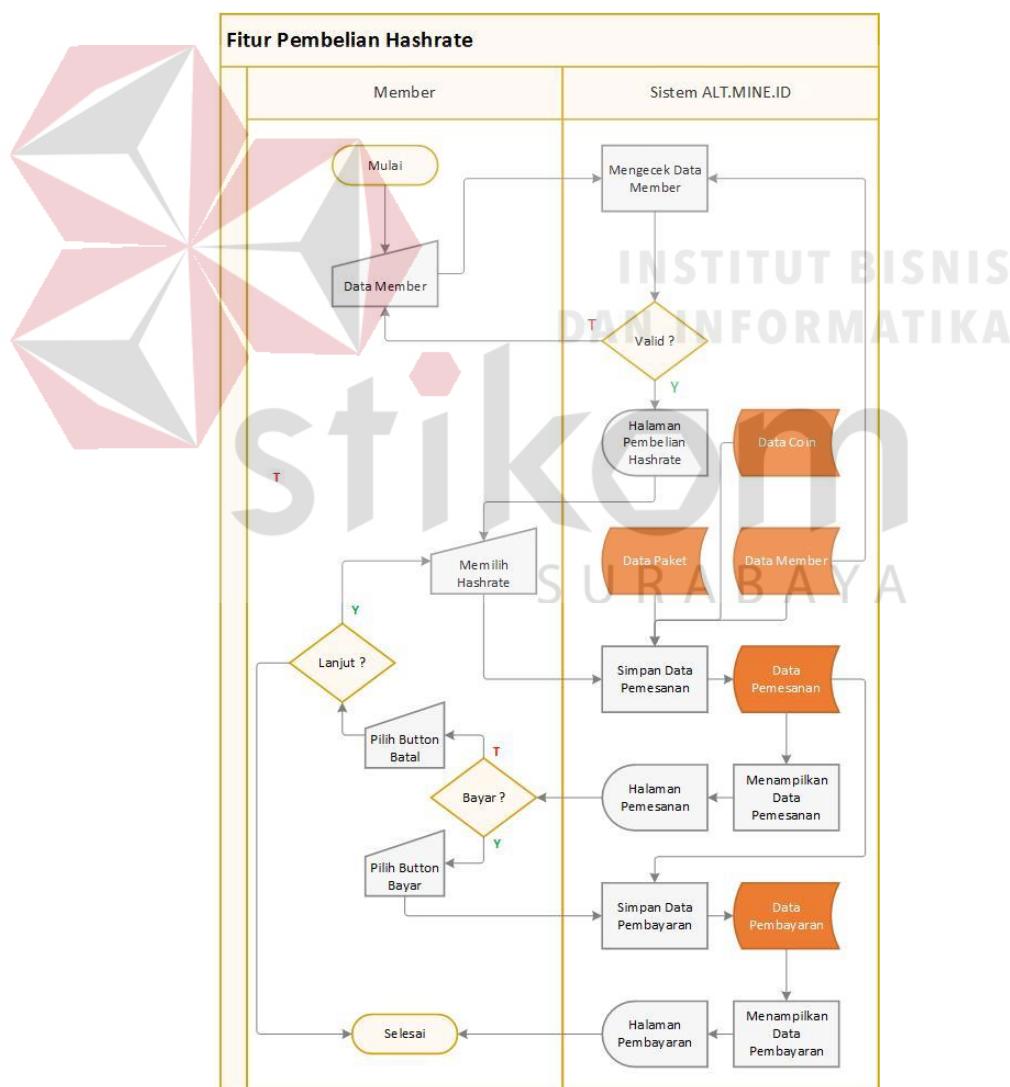
4.4.4. Pengerjaan Sprint

Dalam pengerjaan sprint ini dibuat desain sistem sampai dengan uji coba sistem yang telah jadi. Berikut adalah detail pengerjaan sistem pada sprint 4.

A. Desain Sistem

Setelah dilakukan perancangan selanjutnya dilakukan pembuatan desain sistem pembelian *hashrate*. Berikut adalah beberapa desain sistem yang dibuat:

1. System Flow

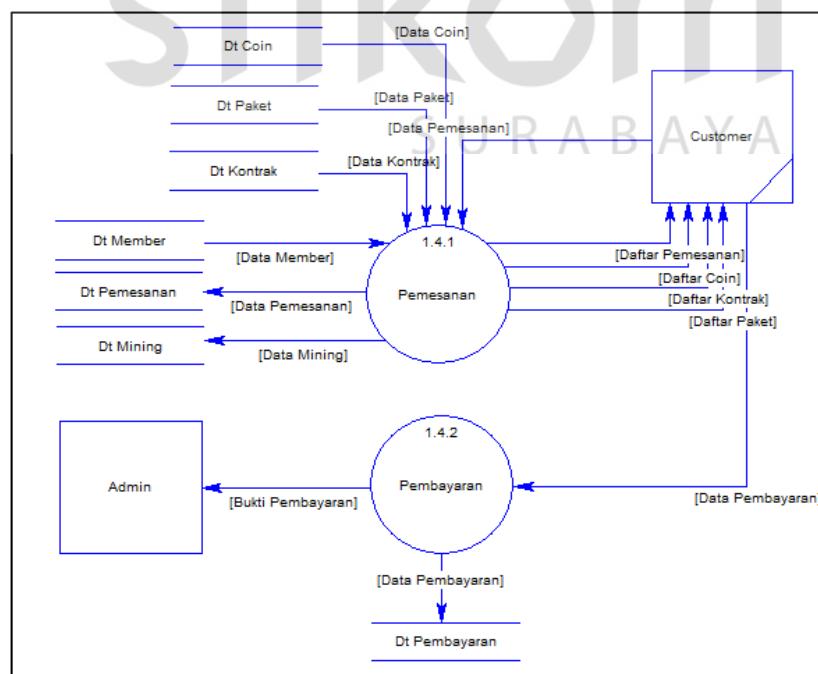


Gambar 4.53 System Flow Pembelian Hashrate

Pada Gambar 4.53 di bawah, *customer* dapat membeli *hashrate* dengan cara melakukan *login* terlebih dahulu ke sistem. Pada saat *login* berhasil, sistem menampilkan daftar *hashrate* yang berisikan harga dan *hashing power*. Data yang telah terisi diproses simpan oleh sistem ke *database* pemesanan dan menampilkan halaman pemesanan. Jika *customer* melakukan aksi bayar data pembayaran yg telah terisi diproses simpan ke *database* pembayaran dan menampilkan halaman pembayaran.

2. DFD Level 1 Pembelian Hashrate

DFD level 1 merupakan diagram yang menguraikan detail proses yang ada pada DFD level 0. Pada bagian ini mengalir untuk kegiatan pengolahan data pemesanan, pembayaran dan data mining yang kemudian disimpan pada masing-masing tabel master. Berikut DFD level 1 pembelian *hashrate* dapat dilihat pada Gambar 4.54 di bawah ini.



Gambar 4.54 DFD Level 1 Pembelian *Hashrate*

3. Struktur Basis Data

Pada sprint 4 ini dilakukan pembuatan tabel struktur basis data untuk fitur pembelian hashrate. Tabel pada struktur basis data yang dibentuk dapat dijelaskan sebagai berikut :

a) Tabel Pemesanan

Nama Tabel : Pemesanan

Primary Key : ID_Pemesanan

Foreign Key : -

Fungsi : Menyimpan data pembelian

Tabel 4.28 Tabel Pemesanan

| No | Field Name | Data Type | Length | Foreign Key | |
|----|----------------|-----------|--------|-------------|----------|
| | | | | On Field | On Table |
| 1 | ID_Pemesanan | Integer | 11 | | |
| 2 | Total_Bayar | VarChar | 30 | | |
| 3 | Jumlah_Pesanan | Integer | 11 | | |
| 4 | Sts_Pemesanan | VarChar | 20 | | |
| 5 | Tgl_Pesan | Date | - | | |

b) Tabel Pembayaran

Nama Tabel : Pembayaran

Primary Key : ID_Pembayaran

Foreign Key : -

Fungsi : Menyimpan data bukti pembayaran

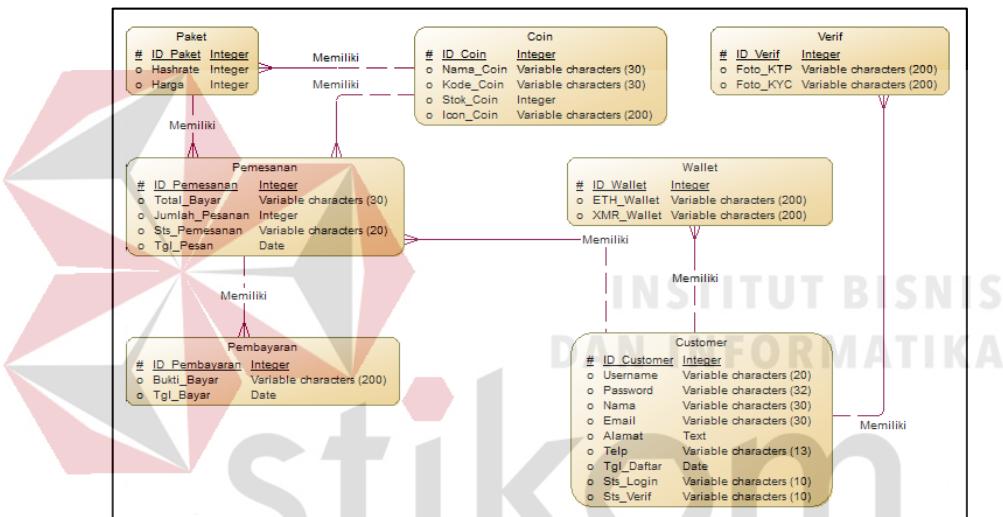
Tabel 4.29 Tabel Pembayaran

| No | Field Name | Data Type | Length | Foreign Key | |
|----|---------------|-----------|--------|-------------|----------|
| | | | | On Field | On Table |
| 1 | ID_Pembayaran | Integer | 11 | | |
| 2 | Bukti_Bayar | VarChar | 200 | | |

| No | Field Name | Data Type | Length | Foreign Key | |
|----|------------|-----------|--------|-------------|----------|
| | | | | On Field | On Table |
| 3 | Tgl_Bayar | Date | - | | |

4. CDM

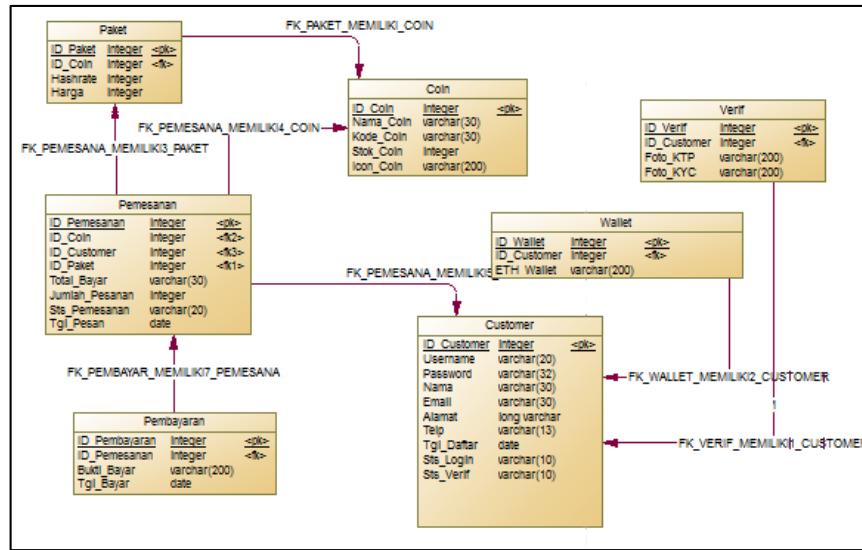
CDM merupakan rancangan awal konsep desain database yang nantinya akan di generate kedalam bentuk *Physical Data Model*. Berikut ini merupakan desain relasi antar tabel dalam sprint pembelian *hashrate*.



Gambar 4.55 CDM Pembelian *Hashrate*

5. PDM

PDM merupakan hasil dari generate CDM. Secara keseluruhan PDM menggambarkan basis data yang telah jadi dan dapat di generate script kedalam database server. Berikut ini merupakan desain PDM dalam sprint pembelian *hashrate*.

Gambar 4.56 PDM Pembelian *Hashrate*

6. Desain Interface

Pada sprint 4 ini dilakukan pembuatan desain interface halaman *customer* sub menu pembelian *hashrate* dan sub menu pembayaran. Berikut adalah desain *interface* yang dibuat sebagai acuan dalam implementasi aplikasi:

a) Desain Halaman Customer



Gambar 4.57 Desain Halaman Pembelian

Berikut adalah desain halaman *customer* (*sub menu* pembelian, dan pesanan saya (pembayaran) yang digunakan sebagai acuan dalam implementasi halaman

customer, dapat dilihat di bawah ini Gambar 4.57 halaman pembelian, dan

Gambar 4.58 halaman pembayaran :



Gambar 4.58 Desain Halaman Pembayaran

b) Desain Halaman Admin



Gambar 4.59 Halaman *Request* Pembayaran



Gambar 4.60 Halaman *Setup Mining*

Berikut adalah desain halaman *admin* (*sub menu request pembayaran, dan setup mining*) yang digunakan sebagai acuan dalam implementasi halaman *admin*, dapat dilihat di bawah, Gambar 4.59 halaman *request pembayaran*, dan Gambar 4.60 halaman *setup mining*:

7. Uji Coba Sistem

Pada *sprint 4* ini dilakukan uji coba sistem pada halaman *customer* sub menu pembelian. Detail dari desain uji coba dapat dilihat di bawah ini :

a) Desain Uji Coba Menu Pembelian

Desain uji coba menu pembelian ini berfungi untuk mengecek apakah dapat melakukan pembelian dan menampilkan detail pembelian sesuai dengan inputan. Desain uji coba menu pembelian dapat dilihat pada Tabel 4.30.

Tabel 4.30 Desain Uji Coba Menu Pembelian

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|--------------------------------|---------------------------------|--|
| 1 | Melakukan pembelian | Memilih paket yang ingin dibeli | Data paket yang dipilih akan muncul ke dalam detail pembelian |
| 2 | Melakukan konfirmasi pembelian | Menekan tombol simpan | Data tersimpan ke sistem dan menampilkan data pembelian sesuai inputan |

b) Desain Uji Coba Menu Pesanan Saya

Desain uji coba menu pesanan saya ini berfungi untuk mengecek apakah fungsi upload bukti pembayaran dapat berfungsi dan sesuai inputan. Desain uji coba menu pesanan saya dapat dilihat pada Tabel 4.31.

Tabel 4.31 Desain Uji Coba Menu Pesanan Saya

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|-------------------------------------|--|--|
| 1 | Melihat detail pemesanan | Memilih pesanan yang sudah dibeli | Data pesanan muncul berdasarkan input |
| 2 | Melakukan <i>upload</i> bukti bayar | Menekan tombol upload bukti pada pesanan yang telah dibeli | Data tersimpan ke sistem dan menampilkan data pesanan yang sudah terupdate |

c) Desain Uji Coba Menu Request Pembayaran

Desain uji coba menu *request* pembayaran bertujuan untuk mengetahui apakah dapat melakukan persetujuan maupun menolak data pembayaran *customer*. Desain uji coba dapat dilihat pada Tabel 4.32.

Tabel 4.32 Desain Uji Coba Menu *Request* Pembayaran

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|--|---|--|
| 1 | Meng- <i>approve</i> data pembayaran <i>customer</i> | Mengklik tombol verifikasi dan merubah status menjadi sukses | Status pembayaran <i>customer</i> berhasil berubah menjadi sukses |
| 2 | Me- <i>reject</i> data pembayaran <i>customer</i> | Mengklik tombol verifikasi dan merubah status menjadi ditolak | Status pembayaran <i>customer</i> berhasil berubah menjadi ditolak |

d) Desain Uji Coba Menu *Setup Mining*

Desain uji coba menu *setup mining* bertujuan untuk mengetahui apakah dapat melakukan update data proses *mining*. Desain uji coba dapat dilihat pada Tabel 4.33.

Tabel 4.33 Desain Uji Coba Menu Proses *Mining*

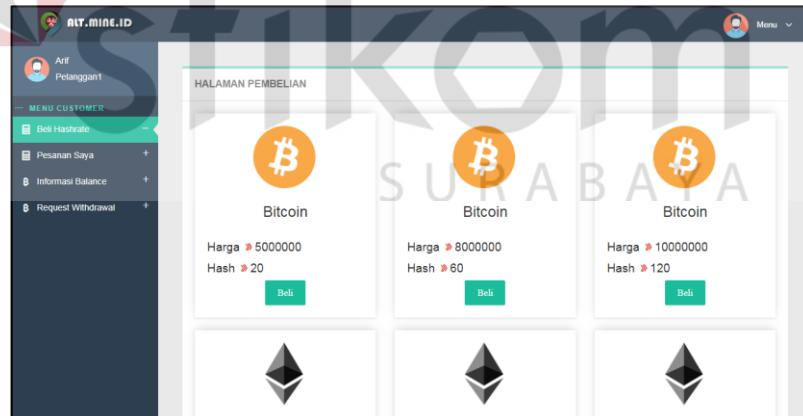
| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|--------------------------------|---|---|
| 1 | Melakukan proses <i>mining</i> | Mengklik tombol proses kemudian mengisi link <i>mining</i> dan merubah status <i>mining</i> menjadi <i>active</i> | Status mining customer berhasil berubah menjadi <i>active</i> |

B. Implementasi Aplikasi

Setelah dilakukan pembuatan desain *interface* selanjutnya dilakukan implementasi terhadap aplikasi. Berikut adalah tampilan aplikasi yang sudah dibuat.:

a) Implementasi Halaman *Customer* (Sub Menu Pembelian)

Untuk melihat pemebelian, pilih sub menu pembelian yang berada di sebelah kiri maka akan muncul tampilan seperti pada Gambar 4.61 di bawah.



Gambar 4.61 Halaman Pembelian

Setelah melakukan pembelian maka akan muncul tampilan halaman detil pembelian seperti pada Gambar 4.62 di bawah.

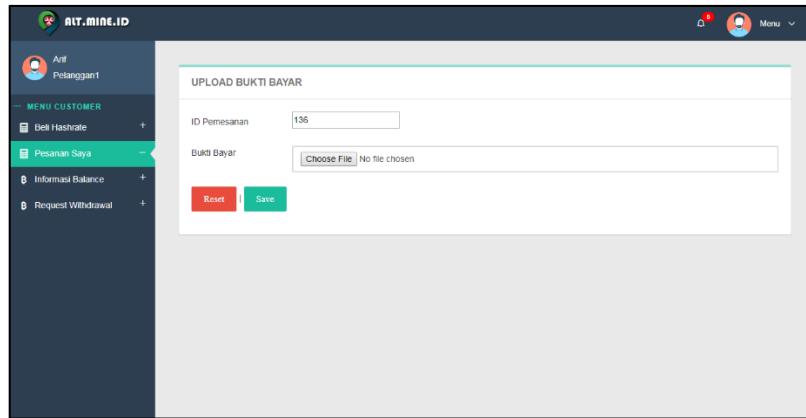
Gambar 4.62 Halaman Detil Pembelian

b) Implementasi Halaman *Customer* (Sub Menu Pembayaran)

Untuk melihat detil pesanan, pilih sub menu pesanan saya yang berada di sebelah kiri maka akan muncul tampilan seperti pada Gambar 4.63.

Gambar 4.63 Halaman Pesanan Saya

Untuk melakukan *upload* bukti pembayaran pilih tombol bayar pada halaman pesanan saya, maka akan muncul tampilan seperti pada Gambar 4.64.



Gambar 4.64 Halaman Pembayaran

c) Halaman *Setup Mining*

Halaman *setup mining* merupakan *sub menu* halaman *admin* yang berfungsi untuk mengelola data *mining*. Pada halaman ini dapat mengubah data proses *mining*. Implementasi halaman *setup mining* dapat dilihat pada Gambar 4.65.



| # | ID Psn | Username | KDCoin | Hash | Tgl Mulai | Tgl Selesai | Status Mining | Tip | Action | Link Mining |
|---|--------|------------|--------|------|------------|-------------|---------------|-------------|----------------------|---|
| 1 | 134 | pelanggan1 | BTC | 20 | 2011-01-01 | 0000-00-00 | Off | 1212 | Edit | Kimochi |
| 2 | 132 | pelanggan1 | ETH | 20 | 2018-09-24 | 0000-00-00 | Off | 656560 | Edit | Kimochi |
| 3 | 131 | pelanggan1 | BTC | 20 | 2018-09-24 | 0000-00-00 | Off | 1212 | Edit | Kimochi |
| 4 | 51 | pelanggan1 | BTC | 20 | 2018-09-02 | 2019-09-02 | Active | 03223109886 | Edit | http://dwarfpool.co |

Gambar 4.65 Halaman *Setup Mining*

Untuk mengubah data, pilih aksi *edit*, maka akan muncul tampilan seperti pada Gambar 4.66.

Gambar 4.66 Halaman *Edit Data Setup Mining*

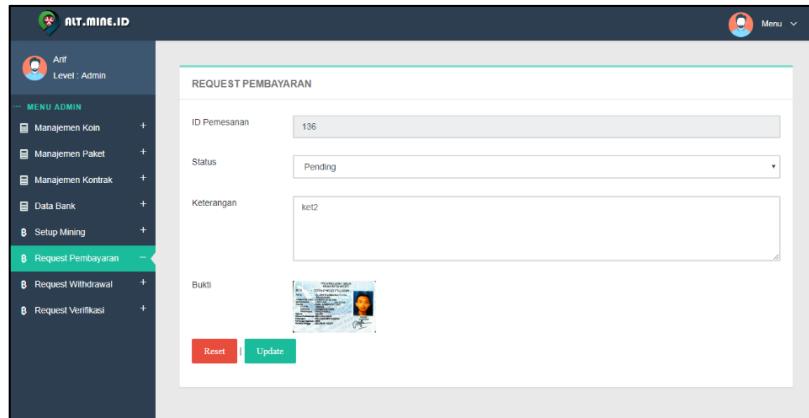
d) Halaman *Request Pembayaran*

Halaman *request pembayaran* merupakan *sub menu* halaman *admin* yang berfungsi untuk mengelola data pembayaran. Pada halaman ini dapat mengubah data pembayaran. Implementasi halaman *request pembayaran* dapat dilihat pada Gambar 4.67.

| No | Username | Kode | Nama Coin | Hashrate | Harga | Bukti Bayar | Status | Proses |
|----|------------|------|-----------|----------|----------|-------------|---------|-------------------------|
| 1 | pelanggan1 | 136 | Bitcoin | 120 | 10000000 | | Pending | <button>Proses</button> |
| 2 | pelanggan1 | 134 | Bitcoin | 20 | 5000000 | | Sukses | <button>Proses</button> |
| 3 | pelanggan1 | 132 | Ethereum | 20 | 3000000 | | Sukses | <button>Proses</button> |
| 4 | pelanggan1 | 131 | Bitcoin | 20 | 5000000 | | Sukses | <button>Proses</button> |
| 5 | pelanggan1 | 115 | Bitcoin | 20 | 5000000 | | Sukses | <button>Proses</button> |

Gambar 4.67 Halaman *Request Pembayaran*

Untuk mengubah data, pilih aksi *edit*, maka akan muncul tampilan seperti pada Gambar 4.68.



Gambar 4.68 Halaman Verifikasi *Request* Pembayaran

C. Ujicoba dan Evaluasi Sistem

Pada sprint 4 dilakukan ujicoba dan evaluasi menu pembelian, menu pesanan saya, menu *request* pembayaran, dan menu *setup mining*. Berikut adalah hasil ujicoba dan evaluasi sistem yang telah dibuat :

a) Uji Coba Menu Pembelian

Proses uji coba dimulai dengan melakukan aksi pemilihan parameter *hashrate* yg ingin dibeli berdasarkan data paket. Hasil uji coba dapat dilihat pada Tabel 4.34.

Tabel 4.34 Uji Coba Menu Pembelian

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|--------------------------------|---------------------------------|--|----------------------|
| 1 | Melakukan pembelian | Memilih paket yang ingin dibeli | Data paket yang dipilih akan muncul ke dalam detail pembelian | Sukses (Gambar 4.61) |
| 2 | Melakukan konfirmasi pembelian | Menekan tombol simpan | Data tersimpan ke sistem dan menampilkan data pembelian sesuai inputan | Sukses (Gambar 4.62) |

b) Uji Coba Menu Pesanan Saya

Proses uji coba dimulai dengan melakukan aksi pemilihan parameter pesanan yg telah dibeli. Hasil uji coba dapat dilihat pada Tabel 4.35.

Tabel 4.35 Uji Coba Menu Pesanan Saya

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|-------------------------------------|---|---|----------------------|
| 1 | Melihat detail pemesanan | Memilih pesanan yang sudah dibeli | Data pesanan muncul berdasarkan input | Sukses (Gambar 4.63) |
| 2 | Melakukan <i>upload</i> bukti bayar | Menekan tombol <i>upload</i> bukti pada pesanan yang telah dibeli | Data tersimpan ke sistem dan menampilkan data pesanan yang sudah <i>terupdate</i> | Sukses (Gambar 4.64) |

c) Uji Coba Menu *Request* Pembayaran

Proses uji coba dimulai dengan melakukan persetujuan maupun menolak data pembayaran dan mencari sesuai dengan inputan. Hasil uji coba dapat dilihat pada Tabel 4.36.

Tabel 4.36 Uji Coba Menu *Request* Pembayaran

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|--|---|--|----------------------|
| 1 | Meng- <i>approve</i> data pembayaran <i>customer</i> | Mengklik tombol verifikasi dan merubah status menjadi sukses | Status pembayaran <i>customer</i> berhasil berubah menjadi sukses | Sukses (Gambar 4.67) |
| 2 | Me-reject data pembayaran <i>customer</i> | Mengklik tombol verifikasi dan merubah status menjadi ditolak | Status pembayaran <i>customer</i> berhasil berubah menjadi ditolak | Sukses (Gambar 4.67) |

d) Uji Coba Menu *Setup Mining*

Proses uji coba *menu setup mining* untuk menguji *update* dan pencarian sesuai dengan data inputan. Hasil uji coba dapat dilihat pada Tabel 4.37.

Tabel 4.37 Uji Coba Menu *Setup Mining*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|--------------------------------|---|--|----------------------|
| 1 | Melakukan proses <i>mining</i> | Mengklik tombol proses kemudian mengisi <i>link mining</i> dan merubah status <i>mining</i> menjadi <i>active</i> | Status <i>mining customer</i> berhasil berubah menjadi <i>active</i> | Sukses (Gambar 4.66) |

4.4.5. Sprint Review

Sprint Review keempat startup Alt.Mine.ID dengan fungsi customer dapat melakukan pembelian hashrate. Pada *Sprint Review* keempat ini terdapat *feedback* perbaikan dari *product owner*. Berikut adalah daftar feedback perbaikan yang telah diberikan oleh project owner dan disetujui oleh *product owner*.

Tabel 4.38 Perbaikan *Sprint Review* Keempat

| <i>No</i> | <i>Story</i> | <i>Fungsi</i> | <i>Goal</i> | <i>Perbaikan</i> |
|-----------|---------------------------------|--|--|---|
| 1 | Fitur pembelian <i>hashrate</i> | Fungsi pembelian <i>hashrate</i> berdasarkan paket Fungsi konfirmasi pembayaran | Dapat melakukan pembelian <i>hashrate</i> Dapat melakukan konfirmasi pembayaran | - - Memperjelas detil pesanan - Perbaikan tampilan notif email - Penambahan cara pembayaran di email |

4.5. Sprint 5 Informasi Balance

Pada sprint 5 akan dijelaskan tahapan penggerjaan fitur informasi *balance* beserta fungsi-fungsi yang ada di dalamnya.

4.5.1. Sprint Planning

Sprint planning kelima memiliki *sprint goal* yaitu dapat menjalankan fitur informasi *balance* dengan fungsi cek *balance altcoin* dan *request withdrawal*. Untuk menyelesaikan goal tersebut maka dibuat :

1. System flow dengan fungsi cek status balance dari hasil mining.
2. Desain basis data menggunakan 2 tabel yaitu tabel mining dan tabel *withdraw*.
3. Desain interface dengan menghasilkan desain halaman customer.
4. Desain uji coba menggunakan blackbox testing dengan melakukan pengujian menu informasi *balance*, dan menu *request withdrawal*.

4.5.2. Sprint Backlog

Pada sprint backlog informasi balance terdapat 2 fungsi yang dibuat yaitu cek balance dan request withdrawal. Tabel *sprint backlog* 5 dapat dilihat pada Tabel 3.39 di bawah ini.

Tabel 4.39 *Sprint Backlog* Informasi *Balance*

| No | Story | Fungsi | Goal |
|----------|--------------------------------|--|--|
| 1 | Fitur informasi <i>balance</i> | Fungsi cek <i>balance coin altcoin</i> | Dapat melakukan cek <i>balance altcoin</i> |
| | | Fungsi <i>request withdrawal</i> | Dapat melakukan <i>request withdrawal</i> |

4.5.3. Daily Scrum

Pada tahap ini fitur informasi balance dan pembelian hashrate masuk ke dalam daftar dikerjakan (in progress) karena dalam sprint 5 selain mengerjakan fitur informasi balance dikerjakan juga feedback perbaikan yang telah didapat pada sprint 4 sebelumnya. Berikut adalah gambaran detail dari daily scrum sprint 5.

| Pengembangan Startup ALT.Mine.ID Untuk Penambangan Cryptocurrency Bagi Investor di Indonesia | | | |
|--|--|---|--|
| | To Do | In Progress | Done |
| | Fitur Approval Withdrawal Fitur Laporan | Fitur Informasi Balance Fitur Pembelian Hashrate | Fitur Manajemen Coin Fitur Manajemen Kontrak Fitur Free Sign Up Member |

INSTITUT BISNIS
DAN INFORMATIKA
stikom
SURABAYA

Gambar 4.69 Daily Scrum Sprint 4

4.5.4. Pengerjaan Sprint

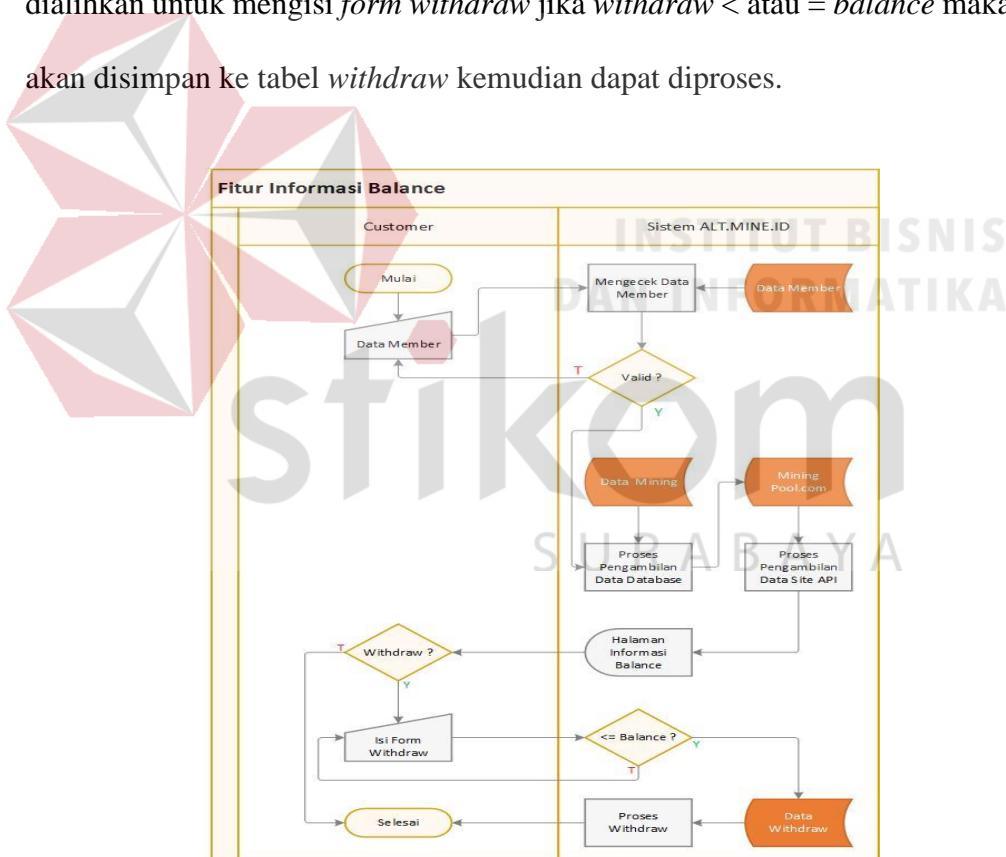
Dalam pengerjaan sprint ini dibuat desain sistem sampai dengan uji coba sistem yang telah jadi. Berikut adalah detail pengerjaan sistem pada sprint 5.

A. Desain Sistem

Setelah dilakukan perancangan selanjutnya dilakukan pembuatan desain sistem informasi *balance*. Berikut adalah beberapa desain sistem yang dibuat:

1. System Flow

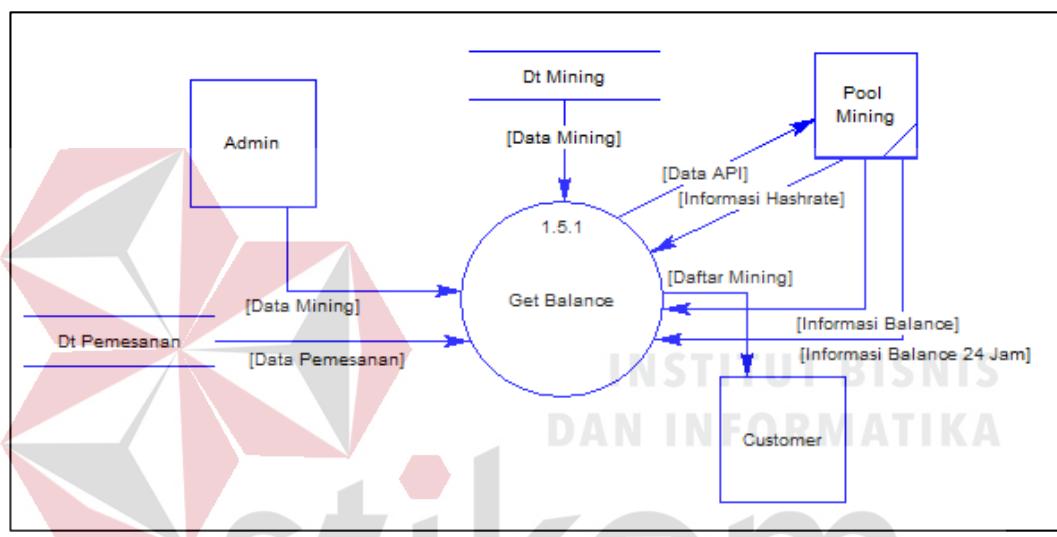
Pada Gambar 4.70 di bawah, *customer* dapat melakukan pengecekan *balance* dengan cara login dan masuk ke halaman informasi *balance*. Sistem akan melakukan proses pengambilan data dari database proses *mining* kemudian sistem melakukan *direct link* ke pool mining yang telah dituju. Dari *pool mining* yang telah dituju tersebut kemudian sistem melakukan pengambilan informasi *balance* berdasarkan alamat *wallet customer* yang kemudian ditampilkan pada halaman informasi *balance*. Jika *customer* melakukan *withdraw* maka *customer* akan dialihkan untuk mengisi *form withdraw* jika *withdraw < atau = balance* maka data akan disimpan ke tabel *withdraw* kemudian dapat diproses.



Gambar 4.70 System Flow Informasi Balance

2. DFD Level 1 Informasi *Balance*

DFD level 1 merupakan diagram yang menguraikan detail proses yang ada pada DFD level 0. Pada bagian ini mengalir untuk kegiatan pengolahan data untuk menampilkan *balance* yang kemudian disimpan pada masing-masing tabel master. Berikut DFD level 1 informasi *balance* dapat dilihat pada Gambar 4.71 di bawah ini.



Gambar 4.71 DFD Level 1 Informasi *Balance*

3. Struktur Basis Data

Pada *sprint 5* ini dilakukan pembuatan tabel struktur basis data untuk fitur informasi *balance*. Tabel pada struktur basis data yang dibentuk dapat dijelaskan sebagai berikut :

a) Tabel Proses *Mining*

Nama Tabel : Proses *Mining*

Primary Key : ID_Proses

Foreign Key : -

Fungsi : Menyimpan data *mining*

Tabel 4.40 Tabel Proses *Mining*

| No | Field Name | Data Type | Length | Foreign Key | |
|-----------|-------------------|------------------|---------------|--------------------|-----------------|
| | | | | On Field | On Table |
| 1 | ID_Proses | Integer | 11 | | |
| 2 | Tgl_Mulai | Date | - | | |
| 3 | Tgl_Selesai | Date | - | | |
| 4 | Link_Mining | VarChar | 200 | | |
| 5 | Sts_Mining | VarChar | 20 | | |

b) Tabel *Withdraw*

Nama Tabel : *Withdraw*

Primary Key : ID_Withdraw

Foreign Key : -

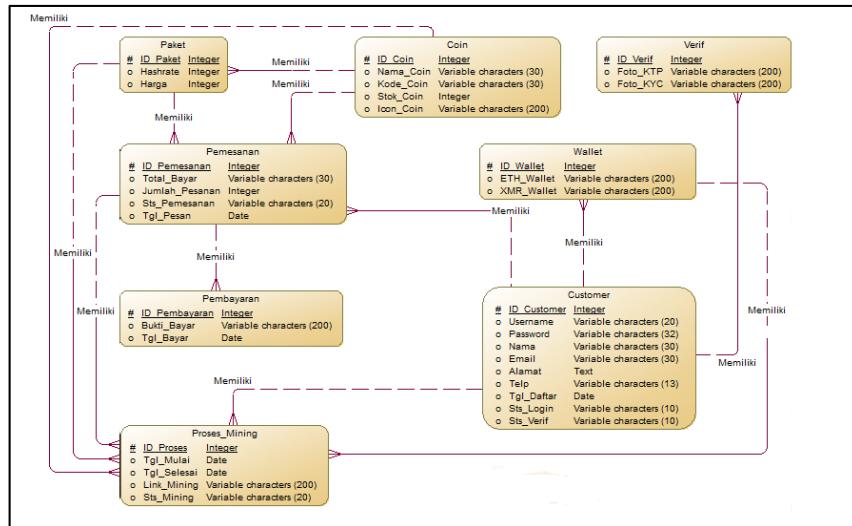
Fungsi : Menyimpan data *request withdraw*

Tabel 4.41 Tabel *Withdraw*

| No | Field Name | Data Type | Length | Foreign Key | |
|-----------|-------------------|------------------|---------------|--------------------|-----------------|
| | | | | On Field | On Table |
| 1 | ID_Withdraw | Integer | 11 | | |
| 2 | Amount_WD | VarChar | 20 | | |
| 3 | Balance | VarChar | 20 | | |
| 4 | Tgl_WD | Date | - | | |
| 5 | Sts_WD | VarChar | 20 | | |
| 6 | Tx_ID | VarChar | 200 | | |

4. CDM

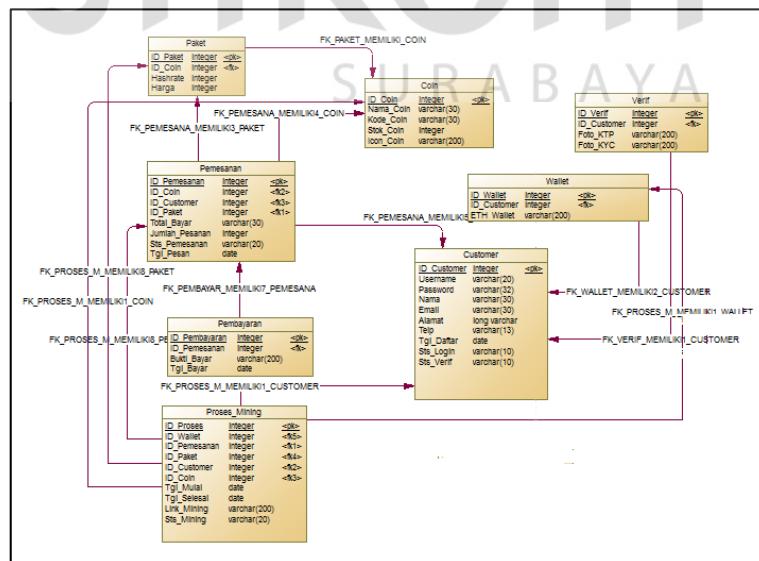
CDM merupakan rancangan awal konsep desain database yang nantinya akan di generate kedalam bentuk Physical Data Model. Desain relasi antar tabel dalam *sprint* informasi *balance* dapat dilihat pada Gambar 4.72.



Gambar 4.72 CDM Informasi Balance

5. PDM

PDM merupakan hasil dari *generate CDM*. Secara keseluruhan PDM menggambarkan basis data yang telah jadi dan dapat di generate script kedalam database server. Berikut ini merupakan desain PDM dalam *sprint* kelima informasi *balance*.



Gambar 4.73 PDM Informasi Balance

6. Desain Interface

Pada *sprint 5* ini dilakukan pembuatan desain *interface* halaman *customer* sub menu informasi *balance* dan sub menu *request withdrawal*. Berikut adalah desain *interface* yang dibuat sebagai acuan dalam implementasi aplikasi:

a) Desain Halaman *Customer*

Berikut adalah desain halaman *customer* (*sub menu* informasi *balance*, dan *request withdrawal*) yang digunakan sebagai acuan dalam implementasi halaman *customer*, dapat dilihat di bawah ini Gambar 4.74 halaman informasi *balance*, dan Gambar 4.75 halaman *request withdrawal*:



Gambar 4.74 Desain Halaman Informasi *Balance*



Gambar 4.75 Desain Halaman *History Withdraw*

7. Uji Coba Sistem

Pada *sprint 5* ini dilakukan uji coba sistem pada halaman *customer* sub menu informasi *balance* dan sub menu *request withdrawal*. Detail dari desain uji coba dapat dilihat di bawah ini :

a) Desain Uji Coba Menu Informasi *Balance*

Desain uji coba menu informasi *balance* ini berfungi untuk mengecek apakah fungsi *get data api* dapat berfungsi dan sesuai inputan dan dapat dilakukan *withdraw balance*. Desain uji coba informasi *balance* dapat dilihat pada Tabel 4.42.

Tabel 4.42 Desain Uji Coba Menu Informasi *Balance*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|----------------------------------|---|---|
| 1 | Melihat informasi <i>balance</i> | Memilih menu informasi <i>balance</i> | Data <i>balance</i> akan muncul berdasarkan pembelian yang sudah terverifikasi oleh admin |
| 2 | Melakukan <i>withdraw</i> | Menekan tombol <i>withdraw</i> pada informasi <i>balance</i> yang ditampilkan | Data tersimpan ke sistem dan menampilkan data <i>withdraw</i> yang dilakukan |

b) Desain Uji Coba Menu *Request Withdrawal*

Desain uji coba menu *request withdrawal* ini berfungi untuk mengecek apakah *withdraw* yang dilakukan sudah diproses atau belum. Desain uji coba *request withdrawal* dapat dilihat pada Tabel 4.43.

Tabel 4.43 Desain Uji Coba Menu *Request Withdrawal*

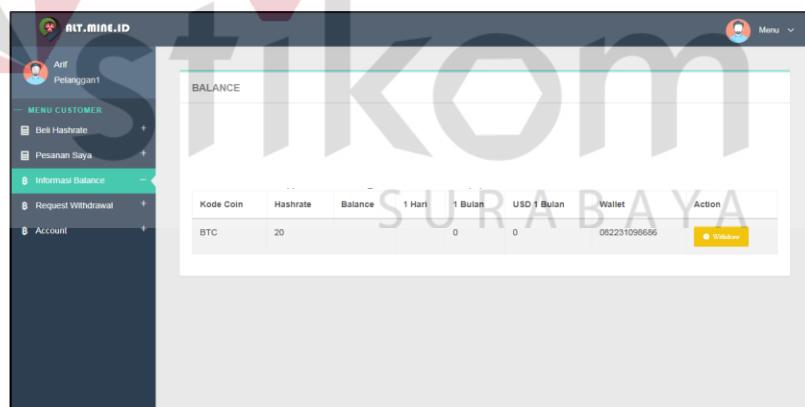
| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|--|--|---|
| 1 | Melihat histori menu <i>request withdrawal</i> | Memilih menu <i>request withdrawal</i> | Data <i>withdraw balance</i> akan muncul berdasarkan <i>request withdraw</i> yang telah dilakukan |

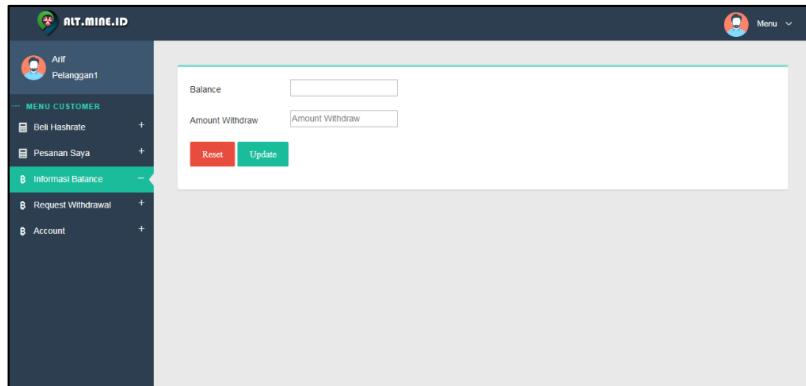
B. Implementasi Aplikasi

Setelah dilakukan pembuatan desain *interface* selanjutnya dilakukan implementasi terhadap aplikasi. Berikut adalah tampilan aplikasi yang sudah dibuat.:

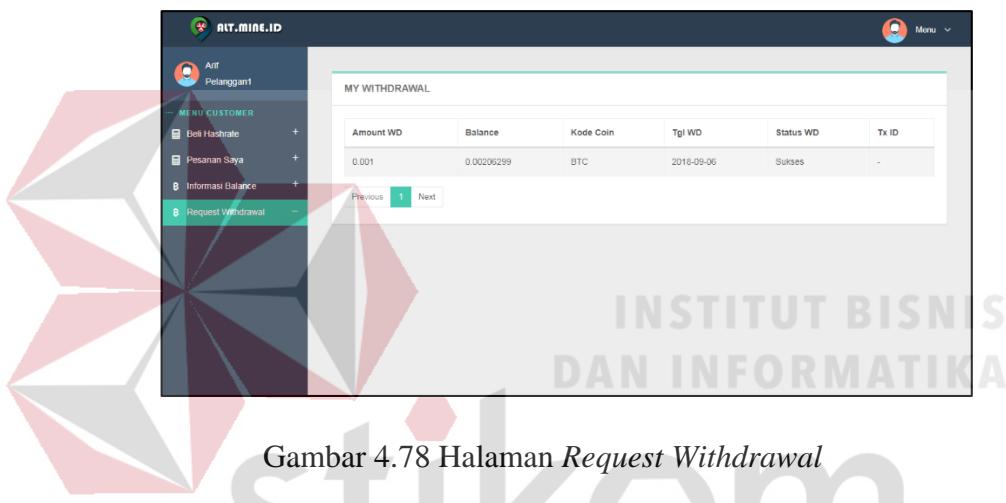
- a) Implementasi Halaman Customer (*Sub Menu* Pembelian)

Untuk melihat informasi *balance* pilih *sub menu* informasi *balance*, maka akan muncul tampilan seperti pada Gambar 4.76.

Gambar 4.76 Halaman Informasi *Balance*



Gambar 4.77 Halaman *Withdraw*



Gambar 4.78 Halaman *Request Withdrawal*

Untuk melakukan *withdraw* pilih tombol *withdraw* pada halaman informasi *balance*, maka akan muncul tampilan seperti pada Gambar 4.77 di atas. Untuk melihat histori *withdraw* pilih sub menu *request withdrawal*, maka akan muncul tampilan seperti pada Gambar 4.78 di atas.

C. Ujicoba dan Evaluasi Sistem

Pada sprint 5 dilakukan ujicoba dan evaluasi menu informasi *balance*, dan menu *request withdrawal*. Berikut adalah hasil ujicoba dan evaluasi sistem yang telah dibuat :

a) Uji Coba Menu Informasi *Balance*

Proses uji coba menu informasi *balance* ini berfungi untuk mengecek apakah fungsi *get data api* dapat berfungsi dan sesuai inputan dan dapat dilakukan *withdraw balance*. Hasil uji coba dapat dilihat pada Tabel 4.44.

Tabel 4.44 Uji Coba Menu Informasi *Balance*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|----------------------------------|---|--|----------------------|
| 1 | Melihat informasi <i>balance</i> | Memilih menu informasi <i>balance</i> | Data <i>balance</i> akan muncul berdasarkan pembelian yang sudah terverifikasi oleh <i>admin</i> | Sukses (Gambar 4.76) |
| 2 | Melakukan <i>withdraw</i> | Menekan tombol <i>withdraw</i> pada informasi <i>balance</i> yang ditampilkan | Data tersimpan ke sistem dan menampilkan data <i>withdraw</i> yang dilakukan | Sukses (Gambar 4.77) |

b) Uji Coba Menu *Request Withdrawal Customer*

Proses uji coba menu *request withdrawal* ini berfungi untuk mengecek apakah *withdraw* yang dilakukan sudah diproses atau belum. Hasil uji coba dapat dilihat pada Tabel 4.45.

Tabel 4.45 Uji Coba Menu *Request Withdrawal Customer*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|-----------------------------------|--|---|----------------------|
| 1 | Melihat histori <i>withdrawal</i> | Memilih menu request <i>withdrawal</i> | Data <i>withdraw balance</i> akan muncul berdasarkan <i>request withdraw</i> yang telah dilakukan | Sukses (Gambar 4.78) |

4.5.5. Sprint Review

Sprint Review kelima startup Alt.Mine.ID dengan customer dapat mengecek informasi balance. Pada *Sprint Review* ini terdapat *feedback* perbaikan dari *product owner*. Berikut adalah daftar feedback perbaikan yang telah diberikan oleh project owner dan disetujui oleh *product owner*.

Tabel 4.46 Perbaikan *Sprint Review* Kelima

| No | Story | Fungsi | Goal | Perbaikan |
|----------|--------------------------------|--|--|---------------------------------------|
| 1 | Fitur informasi <i>balance</i> | Fungsi cek <i>balance coin altcoin</i> | Dapat melakukan cek <i>balance altcoin</i> | - |
| | | Fungsi <i>request withdrawal</i> | Dapat melakukan <i>request withdrawal</i> | - Perbaikan informasi <i>withdraw</i> |

4.6. Sprint 6 Approval Withdrawal

Pada *sprint 6* akan dijelaskan tahapan penggerjaan fitur *approval withdrawal* beserta fungsi-fungsi yang ada di dalamnya.

4.6.1. Sprint Planning

Sprint planning keenam memiliki *sprint goal* yaitu dapat menjalankan fitur *approval withdrawal* dengan fungsi konfirmasi pembayaran *altcoin*. Untuk menyelesaikan goal tersebut maka dibuat:

1. *System flow* dengan fungsi *approve withdrawal* yang dilakukan *customer*
2. Desain basis data menggunakan tabel *withdraw*.
3. Desain *interface* dengan menghasilkan desain halaman admin.
4. Desain uji coba menggunakan *blackbox testing* dengan melakukan pengujian menu *request withdraw*.

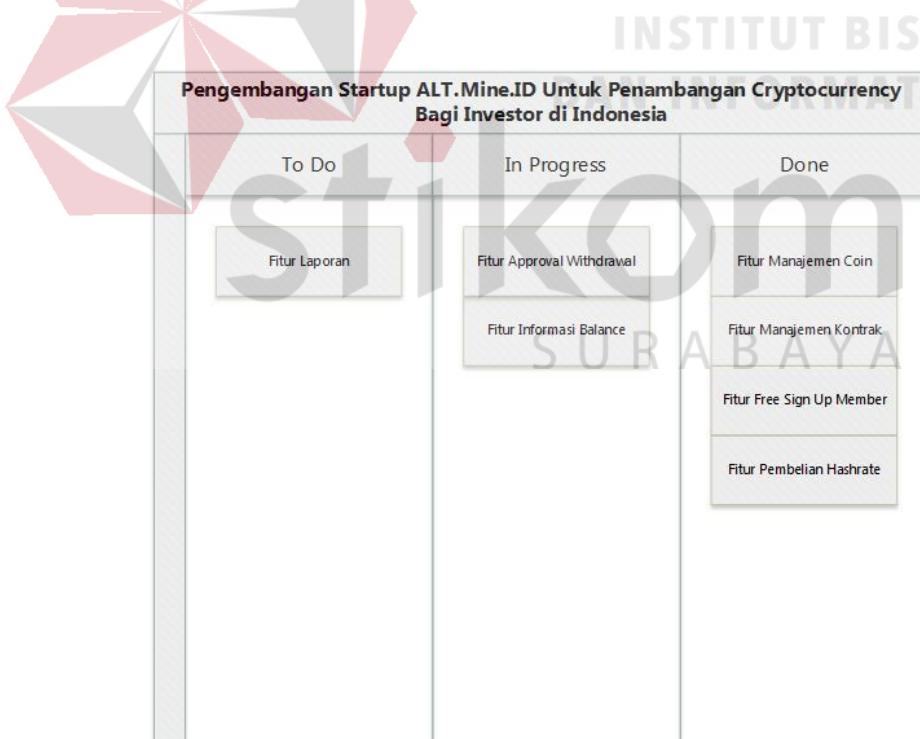
4.6.2. Sprint Backlog

Pada *sprint backlog approval withdrawal* terdapat 1 fungsi yang dibuat yaitu konfirmasi pembayaran *altcoin*. Tabel *sprint backlog* 6 dapat dilihat pada Tabel 4.47 di bawah ini.

Tabel 4.47 *Sprint Backlog Approval Withdrawal*

| No | Story | Fungsi | Goal |
|----|----------------------------------|---|--|
| 1 | Fitur <i>approval withdrawal</i> | Fungsi konfirmasi pembayaran <i>altcoin</i> | Dapat melakukan pembayaran <i>altcoin</i> dan melakukan konfirmasi |

4.6.3. Daily Scrum



Gambar 4.79 *Daily Scrum Sprint 6*

Pada tahap ini fitur approval withdrawal dan informasi *balance* masuk ke dalam daftar dikerjakan (in progress) karena dalam *sprint* 6 selain mengerjakan fitur *approval withdrawal* dikerjakan juga *feedback* perbaikan yang telah didapat pada *sprint* 5 sebelumnya. Berikut adalah gambaran detail dari *daily scrum sprint* 6.

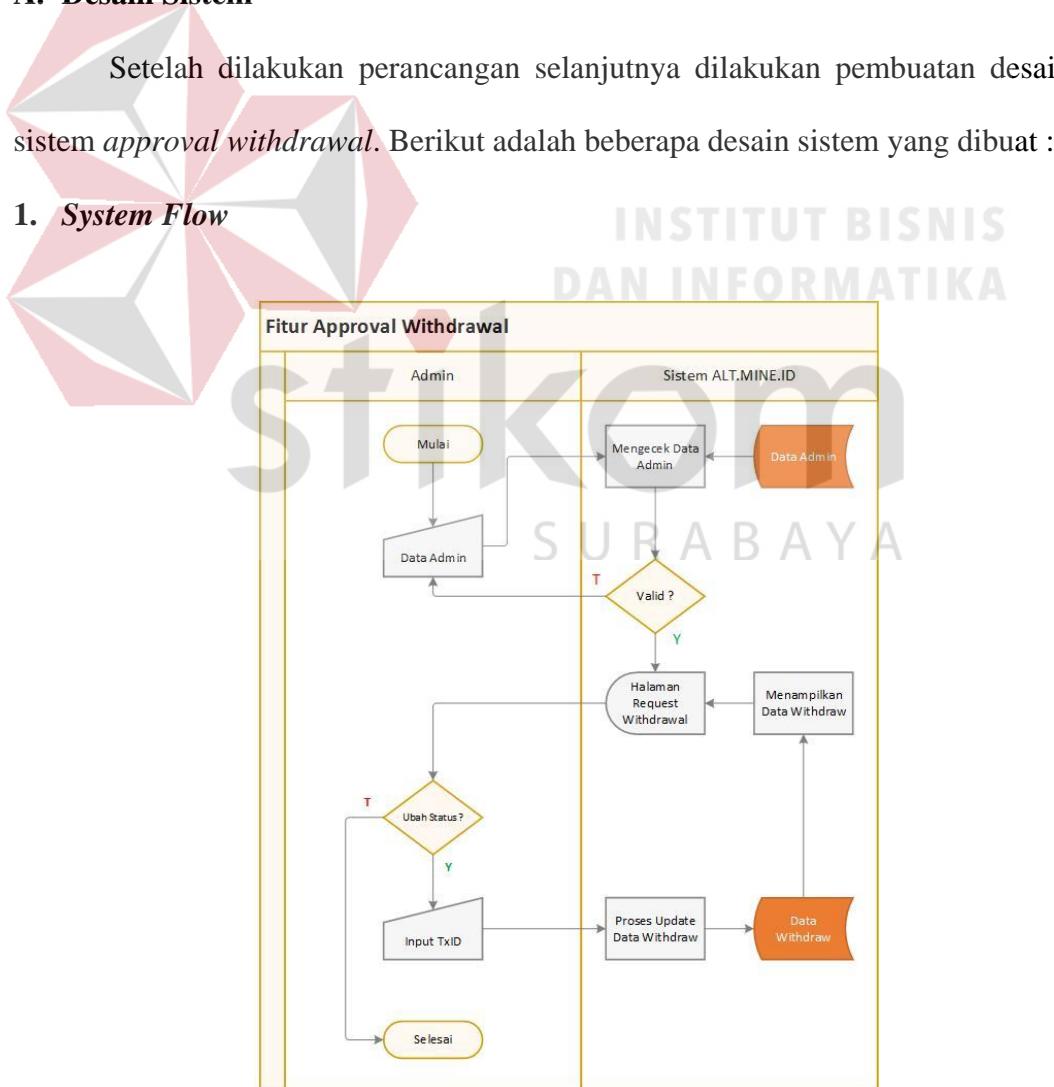
4.6.4. Pengerjaan Sprint

Dalam pengerjaan sprint ini dibuat desain sistem sampai dengan uji coba sistem yang telah jadi. Berikut adalah detail pengerjaan sistem pada sprint 6.

A. Desain Sistem

Setelah dilakukan perancangan selanjutnya dilakukan pembuatan desain sistem *approval withdrawal*. Berikut adalah beberapa desain sistem yang dibuat :

1. System Flow

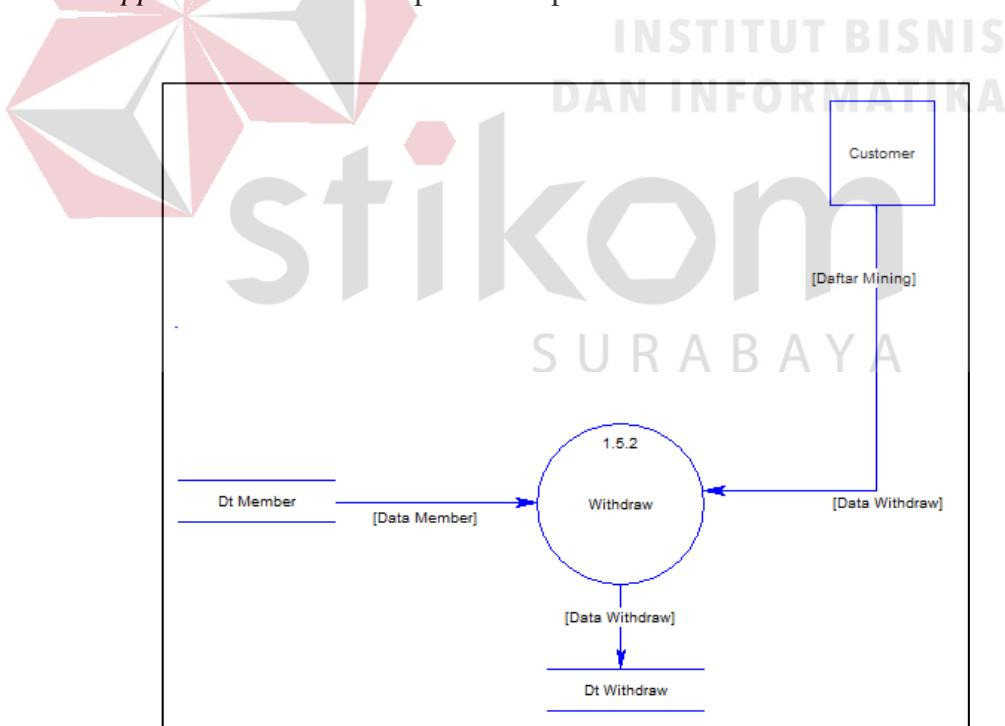


Gambar 4.80 System Flow Approval Withdrawal

Pada Gambar 4.80 di bawah, *admin* dapat melakukan *approval withdrawal* dengan cara login dan masuk ke halaman *request withdrawal*. *Admin* dapat melihat terlebih dahulu detail *amount withdraw* yang diminta oleh *customer* kemudian melakukan pengiriman *altcoin* sejumlah yang diminta *customer*. Setelah melakukan pengiriman *admin* akan mengisikan TxID. Sistem akan melakukan proses *update* yang disimpan ke *database withdraw*.

2. DFD Level 1 Approval Withdrawal

DFD level 1 merupakan diagram yang menguraikan detail proses yang ada pada DFD level 0. Pada bagian ini mengalir untuk kegiatan pengolahan data *withdraw* yang kemudian disimpan pada masing-masing tabel master. Berikut DFD level 1 *approval withdrawal* dapat dilihat pada Gambar 4.81 di bawah ini.



Gambar 4.81 DFD Level 1 Approval Withdrawal

3. Struktur Basis Data

Pada *sprint 6* ini dilakukan pembuatan tabel struktur basis data untuk fitur *approval withdrawal*. Tabel pada struktur basis data yang dibentuk dapat dijelaskan sebagai berikut :

a) Tabel Withdraw

Nama Tabel : *Withdraw*

Primary Key : *ID_Withdraw*

Foreign Key : -

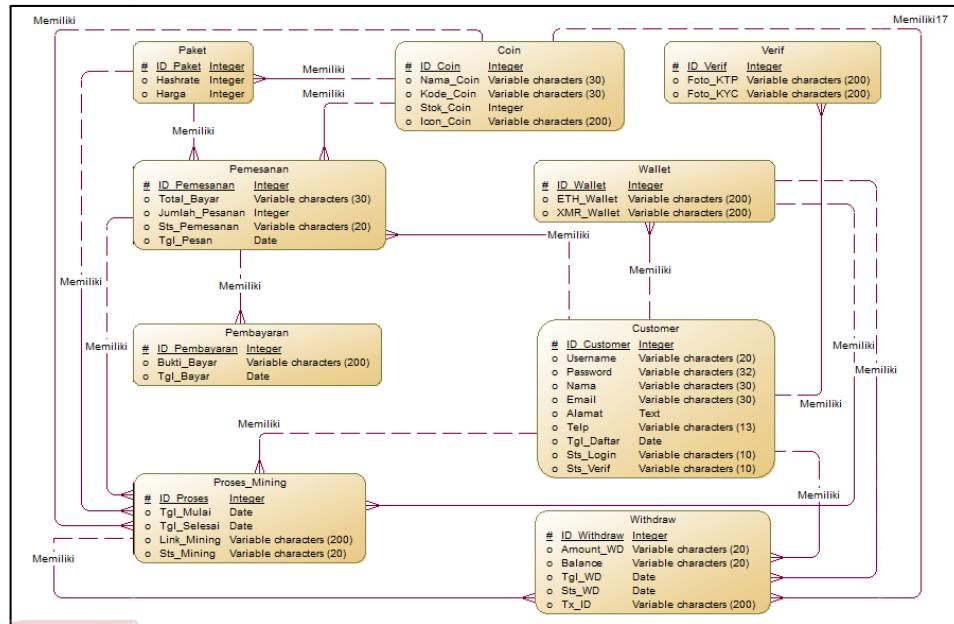
Fungsi : Menyimpan data *request withdraw*

Tabel 4.48 Tabel *Withdraw*

| No | Field Name | Data Type | Length | Foreign Key | |
|----|--------------------|-----------|--------|-------------|----------|
| | | | | On Field | On Table |
| 1 | <i>ID_Withdraw</i> | Integer | 11 | | |
| 2 | <i>Amount_WD</i> | VarChar | 20 | | |
| 3 | <i>Balance</i> | VarChar | 20 | | |
| 4 | <i>Tgl_WD</i> | Date | - | | |
| 5 | <i>Sts_WD</i> | VarChar | 20 | | |
| 6 | <i>Tx_ID</i> | VarChar | 200 | | |

4. CDM

CDM merupakan rancangan awal konsep desain database yang nantinya akan di *generate* kedalam bentuk *Physical Data Model*. Berikut ini merupakan desain relasi antar tabel dalam *sprint approval withdrawal*.



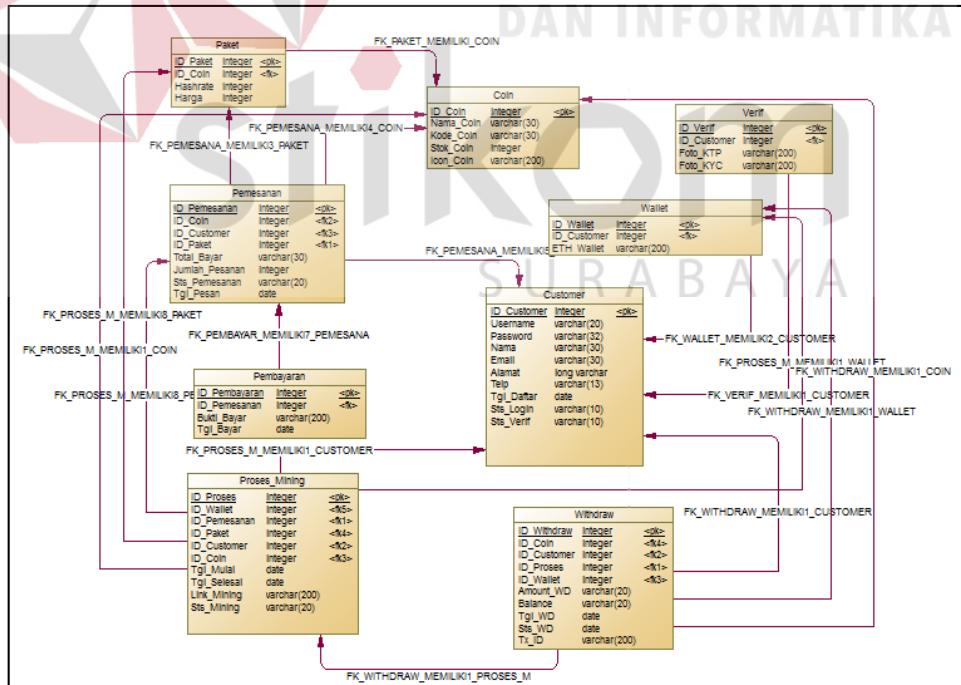
Gambar 4.82 CDM Approval Withdrawal

5. PDM

INSTITUT BISNIS

DAN INFORMATIKA

SURABAYA



Gambar 4.83 PDM Approval Withdrawal

CDM merupakan hasil dari *generate CDM*. Secara keseluruhan PDM menggambarkan basis data yang telah jadi dan dapat di *generate script* kedalam database server. Berikut ini merupakan desain PDM dalam sprint *approval withdrawal*.

6. Desain Interface

Pada *sprint 6* ini dilakukan pembuatan desain interface halaman admin sub menu request withdrawal. Berikut adalah desain *interface* yang dibuat sebagai acuan dalam implementasi aplikasi:

a) Desain Halaman Admin

Berikut adalah desain halaman *admin* (*sub menu request withdraw*) yang digunakan sebagai acuan dalam implementasi halaman *admin*, dapat dilihat di bawah ini Gambar 4.84 halaman *request withdraw* :



Gambar 4.84 Halaman *Request Withdraw*

7. Uji Coba Sistem

Pada *sprint 6* ini dilakukan uji coba sistem pada halaman admin sub menu *request withdrawal*. Detail dari desain uji coba dapat dilihat di bawah ini :

a) Desain Uji Coba Menu *Request Withdraw*

Desain uji coba menú *setup mining* bertujuan untuk mengetahui apakah dapat melakukan update data *withdraw*. Desain uji coba dapat dilihat pada Tabel 4.49.

Tabel 4.49 Desain Uji Coba Menu *Request Withdraw*

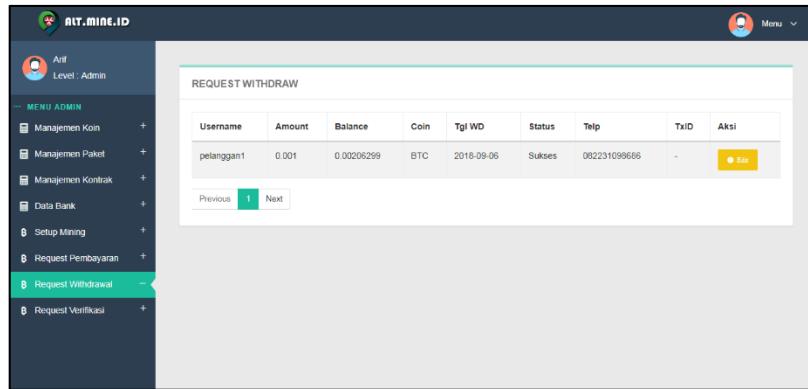
| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|------------------------------------|--|---|
| 1 | Melakukan <i>approval withdraw</i> | Mengklik tombol proses kemudian mengisi TxID dan merubah status <i>withdraw</i> menjadi sukses | Status <i>withdraw customer</i> berhasil berubah menjadi sukses |

B. Implementasi Aplikasi

Setelah dilakukan pembuatan desain *interface* selanjutnya dilakukan implementasi terhadap aplikasi. Berikut adalah tampilan aplikasi yang sudah dibuat.:

a) Implementasi Halaman *Request Withdraw*

Halaman *request withdrawal* merupakan halaman yang berfungsi untuk mengelola data *withdraw*. Pada halaman ini dapat mengubah data *withdraw*. Implementasi halaman *request pembayaran* dapat dilihat pada Gambar 4.85. di bawah. Untuk mengubah data, pilih aksi *edit*, maka akan muncul tampilan seperti pada Gambar 4.86. di bawah.

Gambar 4.85 Halaman *Request Withdraw*

REQUEST WITHDRAW

| | |
|-----------|---|
| Username | <input type="text" value="pelanggan1"/> |
| Amount WD | <input type="text" value="0.001"/> |
| Balance | <input type="text" value="0.00206299"/> |
| Kode Coin | <input type="text" value="BTC"/> |
| Wallet | <input type="text" value="082231098686"/> |
| TxID | <input type="text" value="-"/> |
| Status | <input type="text" value="Pending"/> |

Gambar 4.86 Halaman Verifikasi *Request Withdraw*

C. Ujicoba dan Evaluasi Sistem

Pada *sprint 6* dilakukan ujicoba dan evaluasi menu *request withdraw*.

Berikut adalah hasil ujicoba dan evaluasi sistem yang telah dibuat :

a) Uji Coba Menu *Request Withdrawal Admin*

Proses uji coba menu *setup mining* untuk menguji *update* dan pencarian sesuai dengan data inputan. Hasil uji coba dapat dilihat pada Tabel 4.50.

Tabel 4.50 Uji Coba Menu *Request Withdraw Admin*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|------------------------------------|--|---|----------------------|
| 1 | Melakukan <i>approval withdraw</i> | Mengklik tombol proses kemudian mengisi TxID dan merubah status <i>withdraw</i> menjadi sukses | Status <i>withdraw customer</i> berhasil berubah menjadi sukses | Sukses (Gambar 4.86) |

4.6.5. *Sprint Review*

Sprint Review keenam startup Alt.Mine.ID dengan fitur admin dapat melakukan approval withdrawal. Pada *Sprint Review* ini terdapat *feedback* perbaikan dari *product owner*. Berikut adalah daftar feedback perbaikan yang telah diberikan oleh project owner dan disetujui oleh *product owner*.

Tabel 4.51 Perbaikan *Sprint Review* Keenam

| <i>No</i> | <i>Story</i> | <i>Fungsi</i> | <i>Goal</i> | <i>Perbaikan</i> |
|-----------|----------------------------------|---|--|--------------------------------------|
| 1 | Fitur <i>approval withdrawal</i> | Fungsi konfirmasi pembayaran <i>altcoin</i> | Dapat melakukan pembayaran <i>altcoin</i> dan melakukan konfirmasi | - Memperjelas info wd <i>altcoin</i> |

4.7. *Sprint 7 Laporan*

Pada sprint 7 akan dijelaskan tahapan pelaksanaan fitur laporan beserta fungsi-fungsi yang ada di dalamnya.

4.7.1. *Sprint Planning*

Sprint planning ketujuh memiliki *sprint goal* yaitu dapat menjalankan fitur laporan dengan fungsi pembuatan laporan pembelian, laporan *customer*, dan laporan stok *coin*. Untuk menyelesaikan goal tersebut maka dibuat :

1. *System flow* dengan fungsi pembuatan laporan oleh *owner*
2. Desain basis data menggunakan 3 tabel yaitu tabel customer, tabel coin, dan tabel pembayaran.
3. Desain *interface* dengan menghasilkan desain halaman owner.
4. Desain uji coba menggunakan *blackbox testing* dengan melakukan pengujian menu laporan pendapatan, menu laporan stok *coin*, dan menu laporan *customer*.

4.7.2. Sprint Backlog

Pada *sprint backlog* laporan terdapat 3 fungsi yang dibuat yaitu laporan pembelian, laporan customer dan laporan stok coin. Tabel *sprint backlog* 7 dapat dilihat pada Tabel 4.52 di bawah ini.

Tabel 4.52 *Sprint Backlog Laporan*

| No | Story | Fungsi | Goal |
|----|---------------|---|--|
| 1 | Fitur laporan | Fungsi pembuatan laporan pembelian Fungsi pembuatan laporan <i>customer</i> Fungsi pembuatan laporan stok <i>coin</i> | Dapat melakukan cetak laporan pembelian Dapat melakukan cetak laporan <i>customer</i> Dapat melakukan cetak laporan stok <i>coin</i> |

4.7.3. Daily Scrum

Pada tahap ini fitur laporan dan *approval withdrawal* masuk ke dalam daftar dikerjakan (*in progress*) karena dalam sprint 7 selain mengerjakan fitur laporan dikerjakan juga *feedback* perbaikan yang telah didapat pada sprint 6 sebelumnya.

Berikut adalah gambaran detail dari *daily scrum sprint 7*.

| Pengembangan Startup ALT.Mine.ID Untuk Penambangan Cryptocurrency Bagi Investor di Indonesia | | |
|--|--|---|
| To Do | In Progress | Done |
| | Fitur Laporan Fitur Approval Withdrawal | Fitur Manajemen Coin Fitur Manajemen Kontrak Fitur Free Sign Up Member Fitur Pembelian Hashrate Fitur Informasi Balance |

Gambar 4.87 Daily Scrum Sprint 7

4.7.4. Pengerjaan Sprint

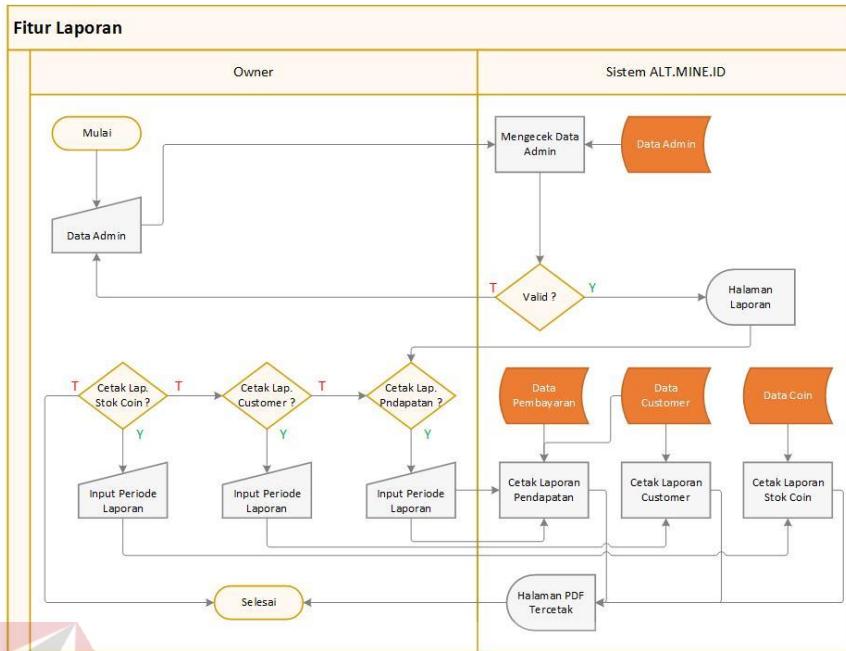
Dalam pengerjaan *sprint* ini dibuat desain sistem sampai dengan uji coba sistem yang telah jadi. Berikut adalah detail pengerjaan sistem pada *sprint* 7.

A. Desain Sistem

Setelah dilakukan perancangan selanjutnya dilakukan pembuatan desain sistem laporan. Berikut adalah beberapa desain sistem yang dibuat:

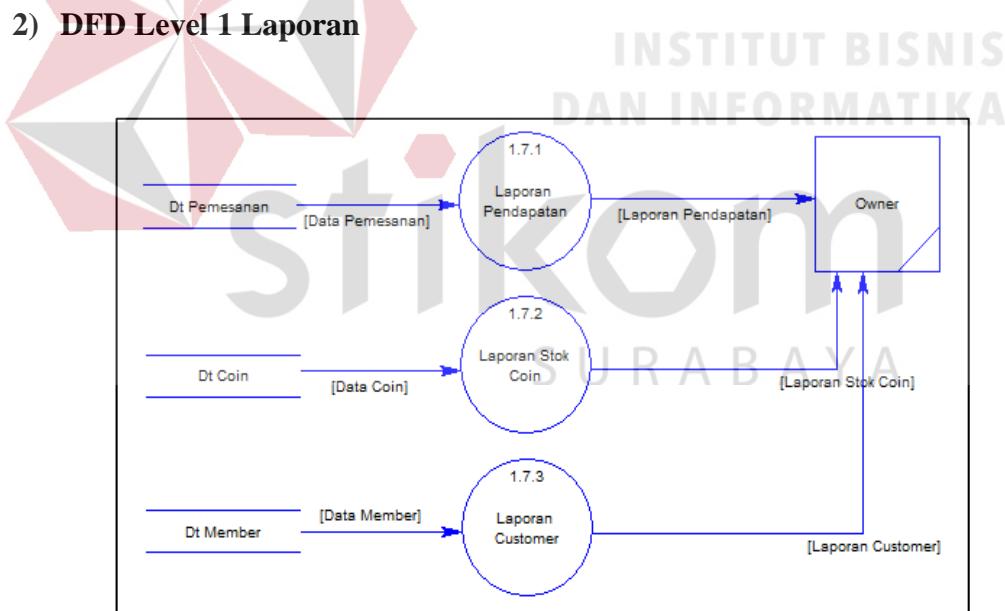
1) System Flow

Pada Gambar 4.88 di bawah, alur proses laporan dimulai oleh *owner* dengan cara melakukan *login*. Pada saat *login* berhasil, sistem menampilkan halaman laporan. *Owner* dapat melakukan cetak laporan *coin*, *customer*, atau pendapatan yang akan disimpan ke dalam file pdf.



Gambar 4.88 System Flow Laporan

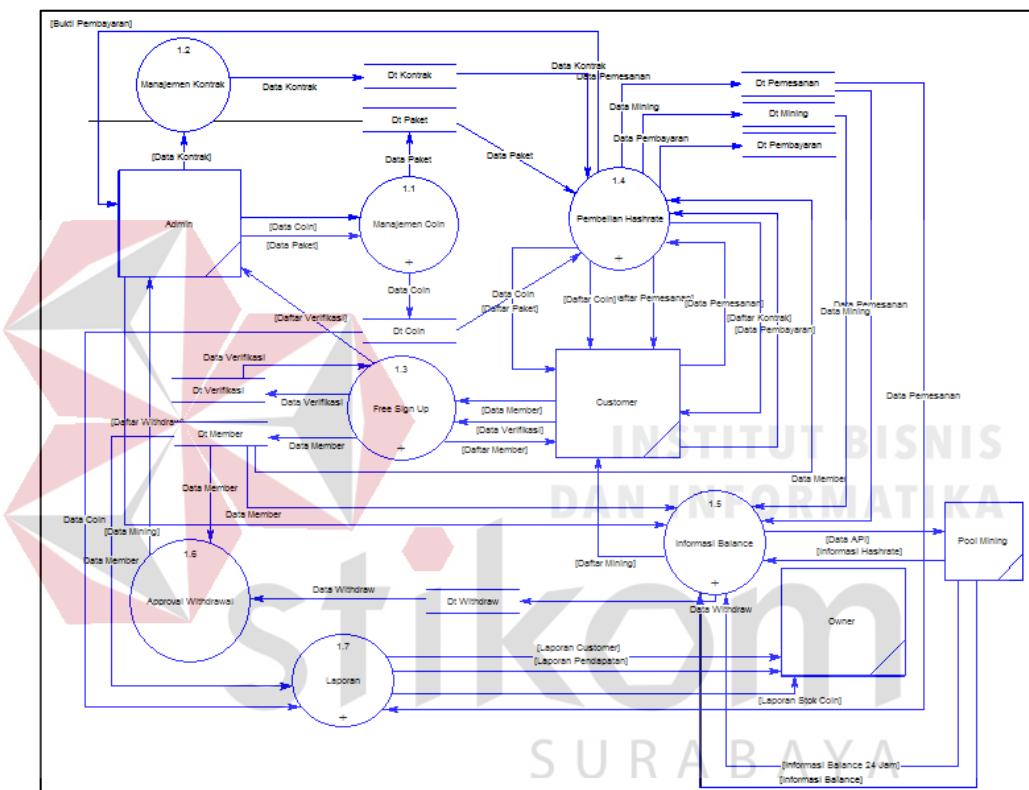
2) DFD Level 1 Laporan



Gambar 4.89 DFD Level 1 Laporan

DFD level 1 merupakan diagram yang menguraikan detail proses yang ada pada DFD level 0. Pada bagian ini mengalir untuk kegiatan pengolahan data laporan. Berikut DFD level 1 laporan dapat dilihat pada Gambar 4.89 di atas.

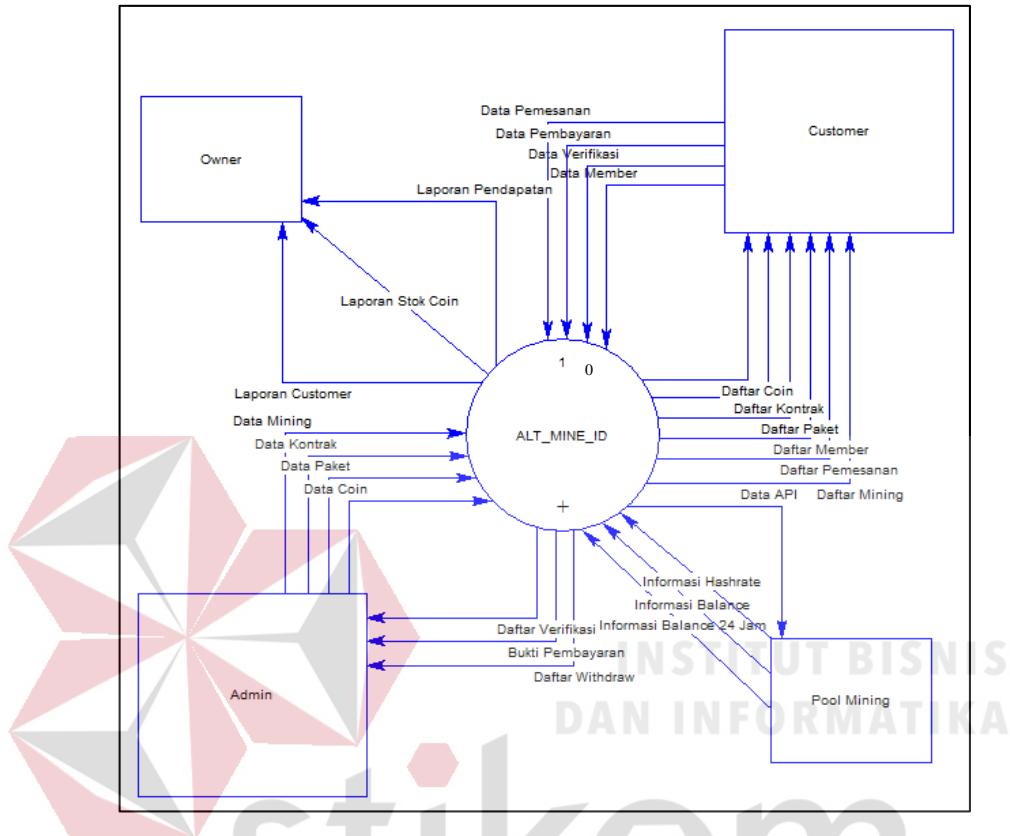
Setelah semua proses terdefinisikan dalam DFD Level 1 maka selanjutnya akan di compose menjadi DFD level 0. DFD level 0 digambarkan secara global proses-proses yang ada pada ALT.MINE.ID. DFD level 0 ini dimulai dari manajemen *coin*, manajemen kontrak, registrasi, pembelian *hashrate*, informasi *balance*, dan laporan. DFD level 0 dapat dilihat pada Gambar 4.90 di bawah ini.



Gambar 4.90 DFD Level 0 Keseluruhan Sistem

Setelah semua proses berhasil di compose menjadi DFD level 0. Maka selanjutnya melanjutkan pembuatan context diagram berdasarkan DFD level 0. *Context diagram* merupakan sebuah gambaran yang menampilkan secara menyeluruh mengenai aliran data pada Data Flow Diagram (DFD). *Context Diagram* atau diagram konteks juga merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Entitas yang

berinteraksi dalam sistem yaitu *admin*, *customer*, dan *owner*. *Context diagram* ALT.MINE.ID dapat dilihat pada Gambar 4.91 di bawah.



Gambar 4.91 *Context Diagram* Keseluruhan Sistem

3) Struktur Basis Data

Pada *sprint 7* ini dilakukan pembuatan tabel struktur basis data untuk fitur laporan. Tabel pada struktur basis data yang dibentuk dapat dijelaskan sebagai berikut :

a) Tabel Customer

Nama Tabel : *Customer*

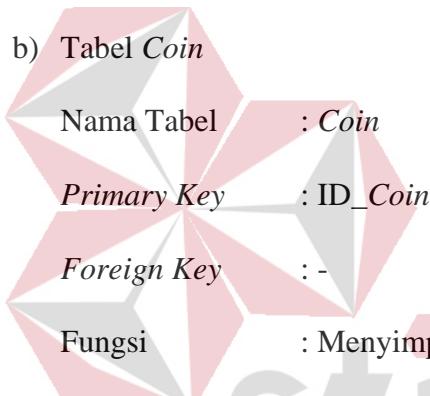
Primary Key : ID_Customer

Foreign Key : -

Fungsi : Menyimpan data *customer*

Tabel 4.53 Tabel *Customer*

| No | Field Name | Data Type | Length | Foreign Key | |
|-----------|-------------------|------------------|---------------|--------------------|-----------------|
| | | | | <i>On Field</i> | <i>On Table</i> |
| 1 | ID_Customer | Integer | 11 | | |
| 2 | Username | VarChar | 20 | | |
| 3 | Password | VarChar | 32 | | |
| 4 | Nama | VarChar | 30 | | |
| 5 | Email | VarChar | 30 | | |
| 6 | Alamat | Text | - | | |
| 7 | Telp | VarChar | 13 | | |
| 8 | Tgl_Daftar | Date | - | | |
| 9 | Sts_Login | VarChar | 10 | | |
| 10 | Sts_Verif | VarChar | 10 | | |

Tabel 4.54 Tabel *Coin*

| No | Field Name | Data Type | Length | Foreign Key | |
|-----------|-------------------|------------------|---------------|--------------------|-----------------|
| | | | | <i>On Field</i> | <i>On Table</i> |
| 1 | ID_Coin | Integer | 11 | | |
| 2 | Nama_Coin | VarChar | 30 | | |
| 3 | Kode_Coin | VarChar | 30 | | |
| 4 | Stok_Coin | Integer | 11 | | |
| 5 | Icon_Coin | VarChar | 200 | | |

c) Tabel Pembayaran

Nama Tabel : Pembayaran

Primary Key : ID_Pembayaran

Foreign Key : -

Fungsi : Menyimpan data bukti pembayaran

Tabel 4.55 Tabel Pembayaran

| No | Field Name | Data Type | Length | Foreign Key | |
|-----------|-------------------|------------------|---------------|--------------------|-----------------|
| | | | | On Field | On Table |
| 1 | ID_Pembayaran | Integer | 11 | | |
| 2 | Bukti_Bayar | VarChar | 200 | | |
| 3 | Tgl_Bayar | Date | - | | |

4) Desain Interface

Pada sprint 7 ini dilakukan pembuatan desain interface login owner, halaman owner sub menu laporan pendapatan, laporan stok coin, dan laporan customer. Berikut adalah desain *interface* yang dibuat sebagai acuan dalam implementasi aplikasi.:

a) Desain Halaman Owner

Berikut adalah desain halaman *owner* (*sub menu* laporan pendapatan, laporan stok *coin*, dan laporan *customer*) yang digunakan sebagai acuan dalam implementasi halaman *owner*, dapat dilihat di bawah, Gambar 4.92 halaman *owner*, Gambar 4.93 halaman laporan pendapatan, Gambar 4.94 halaman laporan stok *coin*, dan Gambar 4.95 halaman laporan *customer* :



Gambar 4.92 Desain Halaman *Owner*

Gambar 4. 93 Desain Halaman Laporan Pendapatan

Gambar 4.94 Desain Halaman Laporan Stok Coin

Gambar 4.95 Desain Halaman Laporan Customer

5) Uji Coba Sistem

Pada *sprint 7* ini dilakukan uji coba sistem pada halaman login owner, halaman *owner* sub menu laporan pendapatan, laporan stok *coin*, dan laporan *customer*. Detail dari desain uji coba dapat dilihat di bawah ini :

a) Desain Uji Coba Menu Laporan Pendapatan

Desain uji coba berfungsi untuk menampilkan informasi detail data pendapatan dari hasil pembelian *hashrate* oleh *customer*. Detail laporan pendapatan memudahkan *owner* untuk memantau perkembangan profit dari penjualan *hashrate*. Desain uji coba dapat dilihat pada Tabel 4.56.

Tabel 4.56 Desain Uji Coba Menu Laporan Pendapatan

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|--------------------------------|--|--|
| 1 | Menampilkan laporan pendapatan | Mengisi <i>form</i> tanggal awal dan tanggal akhir kemudian menekan tombol cetak | Menampilkan data laporan pendapatan dalam bentuk pdf |

b) Desain Uji Coba Menu Laporan Stok *Coin*

Tabel 4.57 Desain Uji Coba Menu Laporan Stok *Coin*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|--------------------------------------|--|--|
| 1 | Menampilkan laporan stok <i>coin</i> | Mengisi <i>form</i> tanggal awal dan tanggal akhir kemudian menekan tombol cetak | Menampilkan data laporan stok <i>coin</i> dalam bentuk pdf |

Desain uji coba berfungsi untuk menampilkan informasi detail data stok *coin*. Detail laporan stok *coin* memudahkan *owner* untuk memantau stok *coin* dan memudahkan *owner* dalam memutuskan *hardware* yang akan dibeli guna *restock* nantinya. Desain uji coba dapat dilihat pada Tabel 4.57 di atas.

c) Desain Uji Coba Menu Laporan *Customer*

Desain customer berfungsi untuk menampilkan informasi detail data *customer* yang terdaftar. Detail laporan *customer* memudahkan *owner* untuk memantau banyaknya *customer* yang terdaftar. Desain uji coba dapat dilihat pada Tabel 4.58.

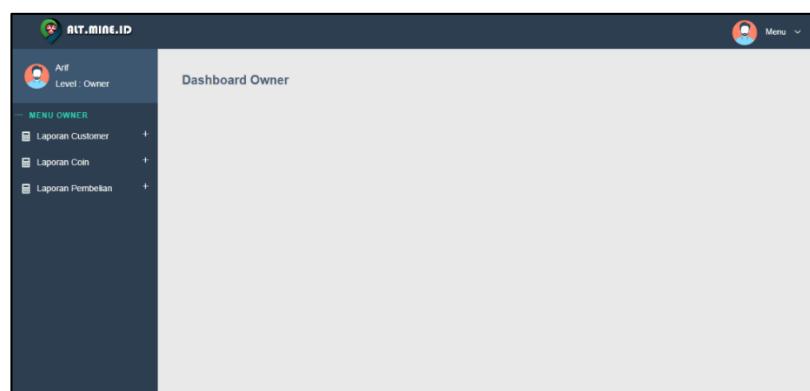
Tabel 4.58 Desain Uji Coba Menu Laporan *Customer*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> |
|---------------------|-------------------------------------|---|---|
| 1 | Menampilkan laporan <i>customer</i> | Mengisi form tanggal awal dan tanggal akhir kemudian menekan tombol cetak | Menampilkan data laporan <i>customer</i> dalam bentuk pdf |

B. Implementasi Aplikasi

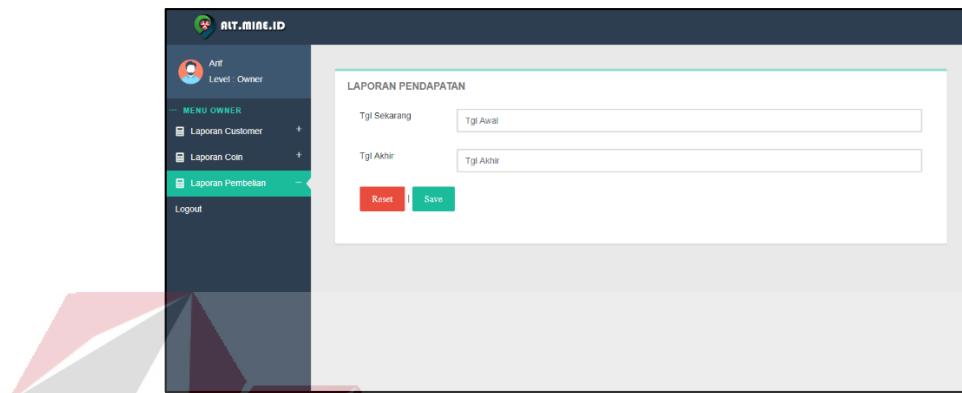
Setelah dilakukan pembuatan desain *interface* selanjutnya dilakukan implementasi terhadap aplikasi. Berikut adalah tampilan aplikasi yang sudah dibuat.:

a) Implementasi Halaman *Owner*

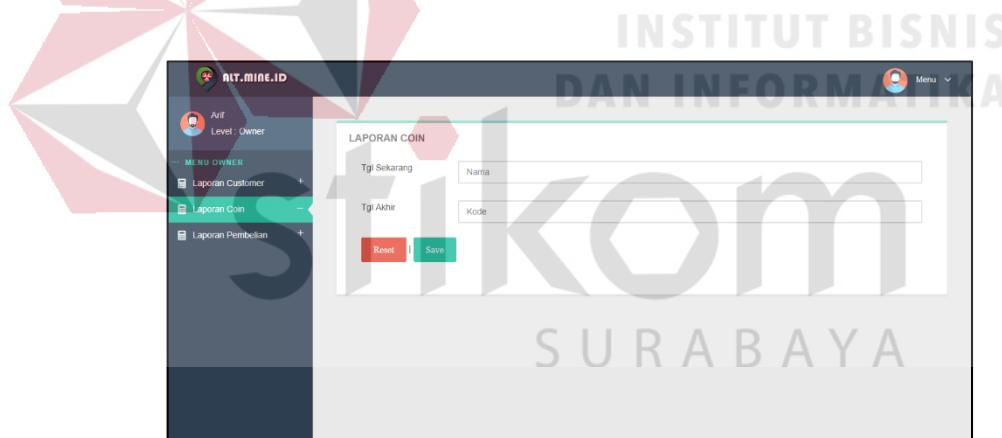


Gambar 4.96 Halaman *Owner*

Halaman *owner* merupakan halaman yang berfungsi untuk melakukan cetak laporan. Implementasi halaman *owner* dapat dilihat pada Gambar 4.96. Untuk melakukan cetak laporan pendapatan pilih tombol pada halaman laporan pendapatan, maka akan muncul tampilan seperti pada Gambar 4.97 di bawah.

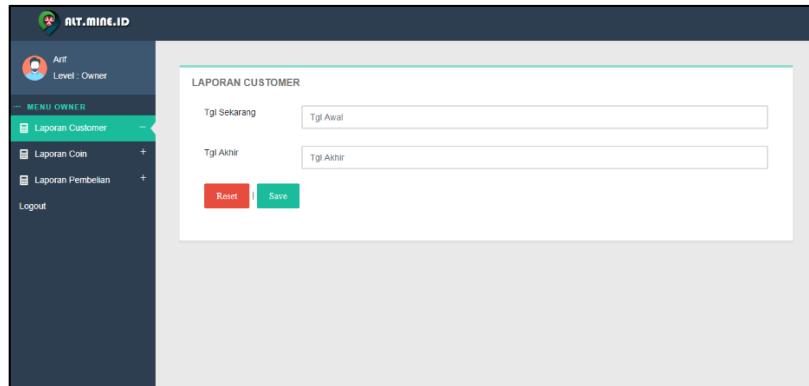


Gambar 4.97 Halaman Laporan Pendapatan



Gambar 4.98 Halaman Laporan Stok Coin

Untuk melakukan cetak laporan stok *coin* pilih tombol pada halaman laporan stok *coin*, maka akan muncul tampilan seperti pada Gambar 4.98 di atas. Untuk melakukan cetak laporan *customer* pilih tombol pada halaman laporan *customer*, maka akan muncul tampilan seperti pada Gambar 4.99 di bawah.



Gambar 4.99 Halaman Laporan *Customer*

C. Ujicoba dan Evaluasi Sistem

Pada sprint 7 dilakukan ujicoba dan evaluasi menu laporan pendapatan, menu laporan stok coin, menu laporan customer. Berikut adalah hasil ujicoba dan evaluasi sistem yang telah dibuat :

a) Uji Coba Menu Laporan Pendapatan

Proses uji coba berfungsi untuk menampilkan informasi detail pendapatan.

Hasil uji coba dapat dilihat pada Tabel 4.59.

Tabel 4.59 Uji Coba Menu Laporan Pendapatan

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|--------------------------------|---|--|----------------------|
| 1 | Menampilkan laporan pendapatan | Mengisi form tanggal awal dan tanggal akhir kemudian menekan tombol cetak | Menampilkan data laporan pendapatan dalam bentuk pdf | Sukses (Gambar 4.97) |

b) Uji Coba Menu Laporan Stok Coin

Proses uji coba berfungsi untuk menampilkan informasi detail stok *coin*.

Hasil uji coba dapat dilihat pada Tabel 4.60.

Tabel 4.60 Uji Coba Menu Stok Coin

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|--------------------------------------|---|--|----------------------|
| 1 | Menampilkan laporan stok <i>coin</i> | Mengisi form tanggal awal dan tanggal akhir kemudian menekan tombol cetak | Menampilkan data laporan stok <i>coin</i> dalam bentuk pdf | Sukses (Gambar 4.98) |

c) Uji Coba Menu Laporan *Customer*

Proses uji coba berfungsi untuk menampilkan informasi detail *customer* yang terdaftar. Hasil uji coba dapat dilihat pada Tabel 4.61.

Tabel 4.61 Uji Coba Menu Laporan *Customer*

| <i>Test Case ID</i> | <i>Tujuan</i> | <i>Input</i> | <i>Output Yang Diharapkan</i> | <i>Status</i> |
|---------------------|-------------------------------------|---|---|----------------------|
| 1 | Menampilkan laporan <i>customer</i> | Mengisi form tanggal awal dan tanggal akhir kemudian menekan tombol cetak | Menampilkan data laporan <i>customer</i> dalam bentuk pdf | Sukses (Gambar 4.99) |

4.7.5. Sprint Review

Sprint Review keenam startup Alt.Mine.ID dengan fitur *owner* dapat membuat laporan. Pada *Sprint Review* ini terdapat *feedback* perbaikan dari *product owner*. Berikut adalah daftar *feedback* perbaikan yang telah diberikan oleh *project owner* dan disetujui oleh *product owner*.

Tabel 4.62 Perbaikan *Sprint Review* Ketujuh

| <i>No</i> | <i>Story</i> | <i>Fungsi</i> | <i>Goal</i> | <i>Perbaikan</i> |
|-----------|---------------|------------------|---|----------------------------|
| 1 | Fitur laporan | Fungsi pembuatan | Dapat melakukan cetak laporan pembelian | - Memperjelas info laporan |

| No | Story | Fungsi | Goal | Perbaikan |
|----|-------|------------------------------------|---|---|
| | | laporan pembelian | | yang ditampilkan |
| | | Fungsi pembuatan laporan customer | Dapat melakukan cetak laporan <i>customer</i> | - Memperjelas info laporan yang ditampilkan |
| | | Fungsi pembuatan laporan stok coin | Dapat melakukan cetak laporan stok coin | - Memperjelas info laporan yang ditampilkan |

4.8. Pembahasan

Berikut ini adalah pembahasan dari fitur-fitur aplikasi startup ALT.Mine.ID yang telah dibuat :

- 1) Fitur manajemen *coin* sudah berhasil dibuat untuk mempermudah penambahan daftar *coin* baru dan manajemen penambangan *coin* tersebut oleh pemilik startup dan juga mempermudah *miners* yang ingin menambang tanpa harus membeli *hardware* yang membutuhkan banyak tempat dan biaya. Sehingga fitur tersebut dapat menyelesaikan masalah pengadaan *hardware* dan resiko investasi sendiri.
- 2) Fitur manajemen kontrak sudah berhasil dibuat untuk mempermudah *miners* dalam melakukan perjanjian sewa alat tambang tanpa harus khawatir penambangan akan dihentikan secara tiba-tiba karena *owner* menganggap sudah tidak *profitable* lagi. Dalam startup ini daftar harga yang yang tercantum sudah termasuk estimasi biaya listrik selama masa kontrak berlangsung, penambangan akan tetap dilakukan meskipun sudah tidak *profitable* karena semua biaya sudah dibayarkan saat pembelian *hashrate*. Sehingga fitur ini dapat menyelesaikan masalah kontrak yang ada *cloud mining* luar negeri.

- 3) Fitur pembelian *hashrate* sudah berhasil dibuat untuk mempermudah *miners* lokal dalam melakukan pembayaran terhadap *hashrate* yang akan disewa. Pada *cloud mining* luar negeri pembayaran bisa dilakukan dengan *convert* dari mata uang rupiah ke dalam mata uang digital ataupun ke mata uang lainnya yang membuat *fee* pembayaran akan membengkak. Maka dari itu fitur pembelian *hashrate* pada startup ini menyediakan alat pembayaran berupa transfer langsung dari seluruh bank di Indonesia sehingga *fee* transaksi pembelian dapat ditekan seminimal mungkin. Sehingga dapat menyelesaikan masalah pembayaran yang lebih susah dilakukan pada *cloud mining* luar negeri.



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil implementasi, uji coba dan evaluasi sistem pada aplikasi ALT.Mine.ID, maka kesimpulan yang didapat adalah sebagai berikut:

1. Aplikasi mampu menyediakan jasa mining *altcoin* (beli *hashrate*, informasi *balance*, *withdraw altcoin*) dengan fitur pembelian *hashrate* yang telah dibuat.
2. Aplikasi mampu memotong biaya konversi yang harus dibayar oleh penambang kepada *cloud mining* luar negeri.
3. Aplikasi tidak akan melakukan pemberhentian kontrak ditengah jalan akibat dari coin yang ditambang sudah tidak *profitable* karena pengaturan kontrak yang telah dilakukan oleh admin pada fitur manajemen kontrak dengan membebankan biaya *maintenance* pada awal pembelian *hashrate*.

5.2. Saran

Adapun saran yang dapat diberikan kepada peneliti berikutnya adalah:

1. Aplikasi dapat dikembangkan dalam bentuk *mobile application* berbasis *android* atau *ios*.
2. Sistem ini dapat diintegrasikan dengan metode transaksi pembayaran yang beragam.
3. Pengembangan desain *interface* yang lebih *user friendly* bagi pengguna.

DAFTAR PUSTAKA

- Abramowitz, M. (2014). Diambil kembali dari Peer-to-peer law built on Bitcoin: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2573788
- Afilipoaie, A., & Shortis, P. (2015). *The Booming Market of Alternative Cryptocurrencies*. Swansea: Swansea University Prifysgol Abertawe.
- Andreas, A. M. (2017). *Mastering Bitcoin : Programming The Open Blockchain : 2nd Edition*. O'Reilly Media.
- Barski, C., & Wilmer, C. (2015). *Bitcoin For The Befuddled*. San Fransisco: No Starch Press.
- Baumann, J., & Lesoismier, A. (2017). *The New Era Of Crypto Wealth Management : CRYPTOCURRENCIES OUTLOOK 2018 Stairway To Heaven*. Swiss Borg.
- Bitcoin Forum*. (2010). Diambil kembali dari Bitcoin: <http://bitcointalk.org/index.php?topic=241.msg8874#msg8874>
- Bitcoin Wiki*. (2011). Diambil kembali dari Anonymity: <https://en.bitcoin.it/wiki/Anonymity>
- Dannen, C. (2017). *Introducing Ethereum and Solidity : Foundations of Cryptocurrency and Blockchain Programming*. New York: Apress.
- Darmawan, O., & Wijaya, D. A. (2017). *Blockchain Dari Bitcoin Untuk Dunia*. Jasakom.
- Eyal, I., & Sirer, E. G. (2014). *Majority is not enough: Bitcoin mining is vulnerable. In Financial Cryptography and Data Security*. Berlin Heidelberg: Springer.
- Hidayat, M. W. (2018, Januari 16). *Bitcoin, Halal atau Haram?* Diambil kembali dari Liputan 6: <https://www.liputan6.com/tekno/read/3227968/bitcoin-halal-atau-haram>
- Hidayat, R. (2010). *Cara Cepat Membangun Website Gratis*. Jakarta: PT Elex Media Komputindo.
- Jogiyanto. (2003). *Sistem Teknologi Informasi*. Yogyakarta: ANDI.
- Quora Forum*. (2011). Diambil kembali dari How can Bitcoin be hacked?: <http://www.quora.com/How-can-Bitcoin-be-hacked>
- Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses* (p. 18).

- Roger S. Pressman, P. (2015). *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Rudianto, A. M. (2011). *Pemrograman Web Dinamis menggunakan PHP dan MySQL*. Yogyakarta: Andi.
- Sas, C., & Khairuddin, I. E. (2015). Exploring Trust in Bitcoin Technology : A Framework For HCI Research. *Research Gate*, 1-3.
- Shcherbak, S. (2014). How Should Bitcoin Be regulated? *European Journal of Legal Studies*, 46-91.
- Szmigielski, A. (2016). *Bitcoin Essentials*. Birmingham: PACKT Publishing.
- Thornburg, E. G. (2012). Going private: technology. Due process and internet dispute resolution. *The Federal Arbitration Act*, 1-16.
- Willey, J. (2016). *Bitcoin For Dummies*. Hoboken: John Wiley & Sons.

