



RANCANG BANGUN APLIKASI PENGAWASAN KINERJA MANDOR

BERBASIS MOBILE PADA PT MANDIRI KARYA DUA SAUDARA

LAPORAN TUGAS AKHIR

Program Studi

S1 Sistem Informasi

Oleh:

FENDY MAHATMA PUTRA

10.41010.0249

INSTITUT BISNIS
DAN INFORMATIKA

The logo for Stikom Surabaya is displayed again, this time in a larger, lighter grey. It consists of the word "stikom" in a lowercase, bold, sans-serif font with a red dot over the "i", and "SURABAYA" in a smaller, uppercase, sans-serif font below it. To the left of the main text, there is a smaller, semi-transparent version of the same logo.

FAKULTAS TEKNOLOGI DAN INFORMATIKA

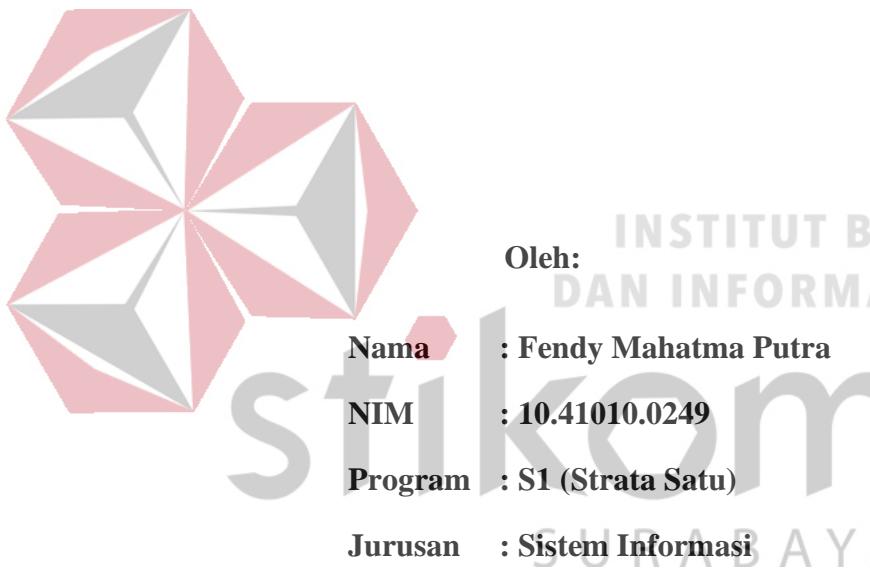
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2018

**RANCANG BANGUN APLIKASI PENGAWASAN KINERJA MANDOR
BERBASIS MOBILE PADA PT MANDIRI KARYA DUA SAUDARA**

TUGAS AKHIR

**Diajukan sebagai syarat untuk menyelesaikan
Program Sarjana**



**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

2018

**RANCANG BANGUN APLIKASI PENGAWASAN KINERJA MANDOR
BERBASIS MOBILE PADA PT MANDIRI KARYA DUA SAUDARA**

Dipersiapkan dan disusun oleh

Fendy Mahatma Putra

10.41010.0249

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

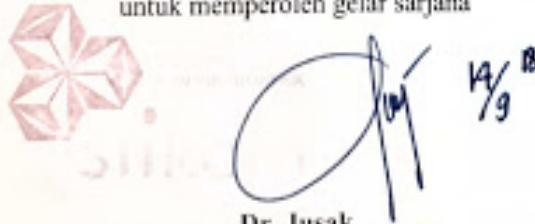
Pada Agustus 2018

Susunan Dewan Penguji



Tugas akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar sarjana



Dr. Jusak
Dekan Fakultas Teknologi dan Informatika

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Fendy Mahatma Putra
NIM : 10410100249
Program Studi : S1 Sistem Informasi
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : RANCANG BANGUN APLIKASI PENGAWASAN KINERJA MANDOR BERBASIS MOBILE PADA PT MANDIRI KARYA DUA SAUDARA

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, Agustus 2018

Yang menyatakan **METERAI TEMPEL**



Fendy Mahatma

NIM : 10.41010.0249

ABSTRAK

PT Mandiri Karya Dua Saudara (PT MKDS) adalah perusahaan yang fokus pada konstruksi properti. PT MKDS memiliki proyek di beberapa kota di Indonesia. Dengan proyek yang berbeda di lokasi yang berbeda, kontraktor tidak dapat lagi melakukan pengawasan harian.

Situasi ini memicu beberapa kondisi yang dapat menghambat proyek. Terjadinya kecurangan yang dilakukan oleh mandor tidak dapat dihindari lagi. Ketidakmampuan untuk mengumpulkan bukti pembelian bahan baku dan catatan adalah salah satu alasannya. Situasi ini, memiliki efek langsung pada alokasi anggaran proyek.

Hal ini merupakan alasan mengapa aplikasi pengawasan mandor benar-benar diperlukan. Aplikasi dibuat berbasis pada aplikasi seluler, karena sangat mudah digunakan kapan saja dan di mana saja selama pengguna terhubung ke internet. Hal ini membuat pelaporan kemajuan penggerjaan, perizinan pembelian dan pengawasan mandor dapat berjalan secara efektif.



ABSTRACT

PT Mandiri Karya Dua Saudara (PT MKDS) is a company that focused on property construction. PT MKDS has projects in several cities in Indonesia. With different projects in different locations, contractors can no longer perform daily supervision

This situation triggers some conditions that can obstruct the project. The occurrence of fraud committed by the foreman no longer can be avoided. The inability to record proof of purchase of raw materials and records is one of the reasons. This situation, has a direct effect on the budget allocation of the project.

This is the reason why foreman supervisory app are really required. Applications create based on mobile apps, because it's very easy to use in anytime and anywhere as long as the user is connected to the internet, this makes reporting progress, permission purchase and supervision of the foreman can working effectively.

Keywords: Foreman, Supervisory, Contractors, Mobile Apps



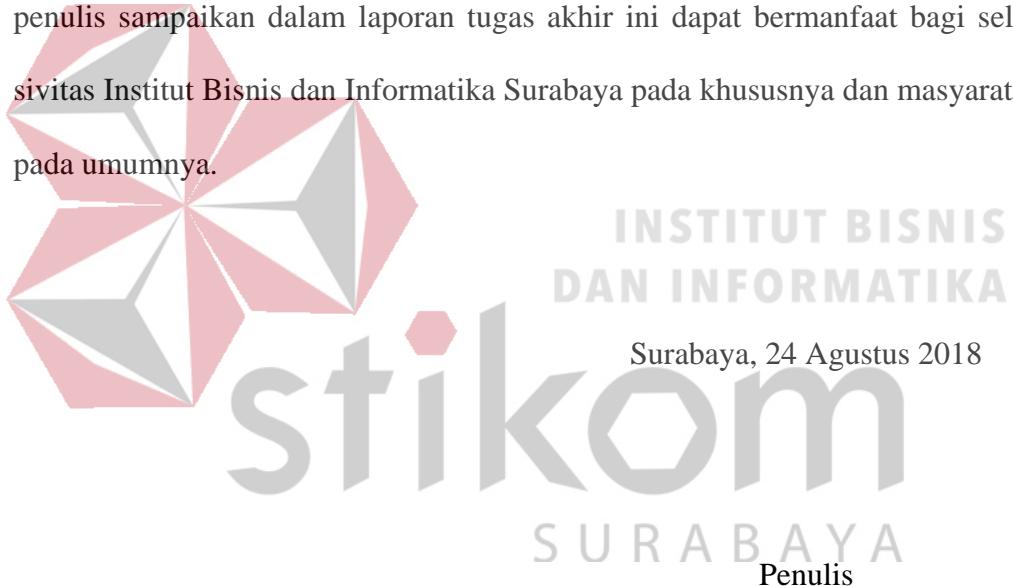
KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas segala berkat dan rahmat Nya sehingga penulis dapat menyelesaikan tugas akhir. Tugas akhir yang dibuat oleh penulis berjudul “Rancang Bangun Aplikasi Pengawasan Kinerja Mandor Berbasis Mobile Pada PT Mandiri Karya Dua Saudara” ini sebagai hasil pertanggungjawaban dari pelaksanaan tugas akhir yang telah dilaksanakan. Dalam pelaksanaan tugas akhir ini, penulis banyak mendapat bantuan dan dorongan dari berbagai pihak secara langsung maupun tidak langsung sehingga tugas akhir ini dapat selesai dan berjalan sesuai dengan yang diharapkan. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Ayah, Mama, Adik serta keluarga tercinta yang selalu mendukung serta telah memberikan izin dan doa dalam menyelesaikan tugas akhir ini.
2. Bapak Prof. Dr. Budi Jatmiko, M.Pd. selaku Rektor Institut Bisnis dan Informatika Stikom Surabaya.
3. Ibu Sri Suhandiah, S.S., M.M. selaku dosen wali sekaligus dosen pembimbing dua yang telah membimbing dan memotivasi selama penulis menempuh bangku kuliah.
4. Ibu Ayuningtyas, S.Kom., M.MT., MOS selaku dosen pembimbing satu yang selalu memberikan motivasi dan bimbingan serta dukungan dalam pelaksanaan dan pembuatan tugas akhir ini.

5. Bapak Tony Soebijono, S.E., S.H., M.Ak. selaku dosen pembahas tugas akhir yang selalu memberikan saran serta dukungannya dalam pelaksanaan tugas akhir.
6. Semua pihak yang tidak bisa penulis sebutkan satu-persatu yang telah memberikan semangat dan doanya untuk keberhasilan dalam penyelesaian dan penyusunan laporan tugas akhir ini.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan laporan tugas akhir ini, baik dari materi atau teknik penyajiannya. Semoga apa yang penulis sampaikan dalam laporan tugas akhir ini dapat bermanfaat bagi seluruh sivitas Institut Bisnis dan Informatika Surabaya pada khususnya dan masyarakat luas pada umumnya.



DAFTAR ISI

ABSTRAK	IV
ABSTRACT	VI
KATA PENGANTAR	VII
DAFTAR ISI	IX
DAFTAR TABEL	XII
DAFTAR GAMBAR	XVII
DAFTAR LAMPIRAN	XXIV
BAB I PENDAHULUAN	1
1.1 PERUMUSAN MASALAH	1
1.2 PERUMUSAN MASALAH	2
1.3 PEMBATASAN MASALAH	2
1.4 TUJUAN PENELITIAN	3
1.5 MANFAAT PENELITIAN	3
1.6 SISTEMATIKA PENULISAN	4
BAB II LANDASAN TEORI	6
2.1 MANDOR	6
2.2 KONTRAKTOR	6
2.3 KINERJA	7
2.4 RENCANA ANGGARAN BIAYA.....	7

2.5 APLIKASI	10
2.6 APLIKASI <i>MOBILE</i>	11
2.7 SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)	11
2.8 UNIFIED MODELING LANGUAGE (UML)	14
2.9 ANDROID STUDIO	15
2.10 SUBLIME TEXT 3.....	16
 BAB III ANALISIS DAN PERANCANGAN SISTEM	17
3.1 <i>PLANNING</i>	17
3.2 <i>ANALYZE</i>	18
3.3 <i>USE CASE</i>	20
3.4 <i>FLOW OF EVENT</i>	26
3.5 <i>SEQUENCE DIAGRAM</i>	66
3.6 <i>CLASS DIAGRAM</i>	80
3.7 <i>DESIGN</i>	91
3.8 <i>PHYSICAL DATA MODEL (PDM)</i>	92
3.9 STRUKTUR DAN FUNGSI TABEL	93
3.10 <i>USER INTERFACE</i>	102
3.11 <i>DESIGN TESTING</i>	118
 BAB IV IMPLEMENTASI DAN EVALUASI	156
4.1 DEPLOYMENT	156
4.2 TESTING.....	158
4.3 IMPLEMENTATION.....	247
 BAB V PENUTUP	248

5.1 KESIMPULAN	248
5.2 SARAN	248
DAFTAR PUSTAKA	250
LAMPIRAN	252



DAFTAR TABEL

Tabel 3.1 Jenis Data dan Instrumen Pengumpulan Data	17
Tabel 3.2 Keterangan <i>Use Case</i> Pengguna Kontraktor.....	21
Tabel 3.3 Keterangan <i>Use Case</i> Pengguna Mandor	24
Tabel 3.4 <i>Flow of Events</i> Mengelola Profil (Kontraktor)	26
Tabel 3.5 <i>Flow of Events</i> Mengelola Barang Gudang (Kontraktor)	28
Tabel 3.6 <i>Flow of Events</i> Mengelola Mandor (Kontraktor).....	31
Tabel 3.7 <i>Flow of Events</i> Mengelola Piutang (Kontraktor)	32
Tabel 3.8 <i>Flow of Events</i> Mengelola Proyek (Kontraktor)	34
Tabel 3.9 <i>Flow of Events</i> Mengelola Pengeluaran Terbayar (Kontraktor)	35
Tabel 3.10 <i>Flow of Events</i> Mengelola Pengeluaran Piutang (Kontraktor).....	37
Tabel 3.11 <i>Flow of Events</i> Mengelola Catatan (Kontraktor)	39
Tabel 3.12 <i>Flow of Events</i> Mengelola Perkembangan (Kontraktor).....	41
Tabel 3.13 <i>Flow of Events</i> Mengelola Permohonan Persetujuan (Kontraktor)	42
Tabel 3.14 <i>Flow of Events</i> Mengelola Persetujuan Pembelian (Kontraktor).....	44
Tabel 3.15 <i>Flow of Events</i> Melihat Detail Laporan Pengeluaran (Kontraktor)....	45
Tabel 3.16 <i>Flow of Events</i> Mengelola Profil (Mandor)	47
Tabel 3.17 <i>Flow of Events</i> Mengelola Barang Gudang (Mandor)	48
Tabel 3.18 <i>Flow of Events</i> Mengelola Mandor (Mandor).....	50
Tabel 3.19 <i>Flow of Events</i> Mengelola Piutang (Mandor)	52
Tabel 3.20 <i>Flow of Events</i> Mengelola Proyek (Mandor).....	53
Tabel 3.21 <i>Flow of Events</i> Mengelola Pengeluaran Terbayar (Mandor)	55
Tabel 3.22 <i>Flow of Events</i> Mengelola Pengeluaran Piutang (Mandor)	57

Tabel 3.23 <i>Flow of Events</i> Mengelola Catatan (Mandor)	59
Tabel 3.24 <i>Flow of Events</i> Mengelola Perkembangan (Mandor)	61
Tabel 3.25 <i>Flow of Events</i> Mengelola Permohonan Persetujuan (Mandor)	62
Tabel 3.26 <i>Flow of Events</i> Melihat Detail Laporan Pengeluaran (Mandor)	65
Tabel 3.27 <i>Users</i>	93
Tabel 3.28 Tabel <i>Works</i>	94
Tabel 3.29 Tabel <i>Works Images</i>	95
Tabel 3.30 Tabel <i>Houses</i>	95
Tabel 3.31 <i>House Expenditure</i>	96
Tabel 3.32 Tabel <i>Houses Expenditure Item</i>	98
Tabel 3.33 Tabel <i>Warehouse Item</i>	99
Tabel 3.34 Tabel <i>Expenditure Warehouse</i>	100
Tabel 3.35 Tabel <i>Expenditure Warehouse Item</i>	101
Tabel 3.36 Tabel <i>Notes</i>	101
Tabel 3.37 Tabel <i>Progress</i>	102
Tabel 3.38 <i>Test Data</i> Halaman <i>Login</i>	119
Tabel 3.39 <i>Test Case</i> Halaman <i>Login</i>	119
Tabel 3.40 <i>Test Case</i> Menu Utama	120
Tabel 3.41 <i>Test Data</i> Menu <i>Houses</i>	122
Tabel 3.42 <i>Test Case</i> Menu <i>Houses</i>	122
Tabel 3.43 <i>Test Data</i> Menu Terbayar	124
Tabel 3.44 <i>Test Case</i> Menu Terbayar	124
Tabel 3.45 <i>Test Data</i> Menu Piutang	128
Tabel 3.46 <i>Test Case</i> Menu Piutang	129

Tabel 3.47 <i>Test Data</i> Menu Note	130
Tabel 3.48 <i>Test Case</i> Menu Note	131
Tabel 3.49 <i>Test Data</i> Menu Progress	132
Tabel 3.50 <i>Test Case</i> Menu Progress	133
Tabel 3.51 <i>Test Case</i> Menu Report.....	134
Tabel 3.52 <i>Test Case</i> Menu Notifikasi.....	134
Tabel 3.53 <i>Test Data</i> Menu Mandor	135
Tabel 3.54 <i>Test Case</i> Menu Mandor	136
Tabel 3.55 <i>Test Data</i> Menu Gudang	138
Tabel 3.56 <i>Test Case</i> Menu Gudang	139
Tabel 3.57 <i>Test Data</i> Menu Piutang	141
Tabel 3.58 <i>Test Case</i> Menu Piutang	142
Tabel 3.59 <i>Test Data</i> Menu Login	143
Tabel 3.60 <i>Test Case</i> Menu Login	144
Tabel 3.61 <i>Test Case</i> Menu Utama	145
Tabel 3.62 <i>Test Case</i> Menu Notifikasi.....	146
Tabel 3.63 <i>Test Case</i> Menu Houses.....	147
Tabel 3.64 <i>Test Data</i> Menu Pengeluaran.....	147
Tabel 3.65 <i>Test Case</i> Menu Pengeluaran.....	148
Tabel 3.66 <i>Test Data</i> Menu Progress	150
Tabel 3.67 <i>Test Case</i> Menu Progress	151
Tabel 3.68 <i>Test Data</i> Menu Note	152
Tabel 3.69 <i>Test Case</i> Menu Note	153
Tabel 3.70 <i>Test Case</i> Menu Pending Request.....	154

Tabel 3.71 <i>Test Case Menu Rejected Request</i>	154
Tabel 3.72 <i>Test Case Menu Piutang</i>	155
Tabel 4.1 Daftar <i>Controller API</i>	157
Tabel 4.2 <i>Test Result Halaman Login</i>	159
Tabel 4.3 <i>Test Result Menu Utama</i>	163
Tabel 4.4 <i>Test Result Menu Houses</i>	167
Tabel 4.5 <i>Test Result Menu Terbayar</i>	172
Tabel 4.6 <i>Test Result Menu Piutang</i>	181
Tabel 4.7 <i>Test Result Menu Note</i>	185
Tabel 4.8 <i>Test Result Menu Progress</i>	188
Tabel 4.9 <i>Test Result Menu Report</i>	193
Tabel 4.10 <i>Test Result Menu Notifikasi</i>	195
Tabel 4.11 <i>Test Result Menu Mandor</i>	197
Tabel 4.12 <i>Test Result Menu Gudang</i>	204
Tabel 4.13 <i>Test Result Menu Piutang</i>	211
Tabel 4.14 <i>Test Result Menu Login</i>	214
Tabel 4.15 <i>Test Result Menu Utama</i>	219
Tabel 4.16 <i>Test Result Menu Notifikasi</i>	224
Tabel 4.17 <i>Test Result Menu Houses</i>	226
Tabel 4.18 <i>Test Result Menu Pengeluaran</i>	227
Tabel 4.19 <i>Test Result Menu Progress</i>	236
Tabel 4.20 <i>Test Result Menu Note</i>	240
Tabel 4.21 <i>Test Result Menu Pending Request</i>	244
Tabel 4.22 <i>Test Result Menu Rejected Request</i>	245

Tabel 4.23 *Test Result* Menu Piutang 246



DAFTAR GAMBAR

Gambar 2.1 Tahap dan Proses Penyusunan RAB	8
Gambar 2.2 Harga Satuan Pekerjaan	9
Gambar 2.3 Tahapan Metode <i>System Development Life Cycle</i>	14
Gambar 3.1 Kondisi Proses Pengawasan Kinerja Proyek pada PT MKDS	19
Gambar 3.2 <i>Use Case</i> Pengguna Kontraktor	21
Gambar 3.3 <i>Use Case</i> Pengguna Mandor	24
Gambar 3.4 <i>Sequence Diagram</i> Mengelola Profil (Kontraktor).....	67
Gambar 3.5 <i>Sequence Diagram</i> Mengelola Barang Gudang (Kontraktor).....	68
Gambar 3.6 <i>Sequence Diagram</i> Mengelola Mandor (Kontraktor)	69
Gambar 3.7 <i>Sequence Diagram</i> Mengelola Proyek (Kontraktor).....	70
Gambar 3.8 <i>Sequence Diagram</i> Mengelola Pengeluaran Terbayar (Kontraktor). 71	71
Gambar 3.9 <i>Sequence Diagram</i> Mengelola Pengeluaran Piutang (Kontraktor) ...	71
Gambar 3.10 <i>Sequence Diagram</i> Mengelola Catatan (Kontraktor).....	72
Gambar 3.11 <i>Sequence Diagram</i> Mengelola Perkembangan (Kontraktor)	73
Gambar 3.12 <i>Sequence Diagram</i> Mengelola Permohonan Persetujuan	73
Gambar 3.13 <i>Sequence Diagram</i> Mengelola Persetujuan (Kontraktor)	74
Gambar 3.14 <i>Sequence Diagram Detail</i> Laporan Pengeluaran (Kontraktor)	74
Gambar 3.15 <i>Sequence Diagram</i> Mengelola Profil (Mandor).....	75
Gambar 3.16 <i>Sequence Diagram</i> Melihat Piutang (Mandor)	76
Gambar 3.17 <i>Sequence Diagram</i> Melihat Permintaan Tertunda (Mandor)	76
Gambar 3.18 <i>Sequence Diagram</i> Melihat Permintaan Ditolak (Mandor)	77
Gambar 3.19 <i>Sequence Diagram</i> Mengelola Proyek (Mandor).....	77

Gambar 3.20 <i>Sequence Diagram</i> Persetujuan Pembelian (Mandor)	78
Gambar 3.21 <i>Sequence Diagram</i> Mengelola Pengeluaran (Mandor)	78
Gambar 3.22 <i>Sequence Diagram</i> Mengelola Catatan (Mandor).....	79
Gambar 3.23 <i>Sequence Diagram</i> Mengelola Perkembangan (Mandor)	79
Gambar 3.24 <i>Class Diagram</i> Mengelola Profil (Kontraktor)	80
Gambar 3.25 <i>Class Diagram</i> Mengelola Barang Gudang (Kontraktor).....	81
Gambar 3.26 <i>Class Diagram</i> Mengelola Mandor (Kontraktor).....	81
Gambar 3.27 <i>Class Diagram</i> Mengelola Piutang (Kontraktor).....	82
Gambar 3.28 <i>Class Diagram</i> Mengelola Proyek (Kontraktor).....	82
Gambar 3.29 <i>Class Diagram</i> (Kontraktor)	83
Gambar 3.30 <i>Class Diagram</i> Mengelola Pengeluaran Piutang (Kontraktor)	83
Gambar 3.31 <i>Class Diagram</i> Mengelola Catatan (Kontraktor)	84
Gambar 3.32 <i>Class Diagram</i> Mengelola Perkembangan (Kontraktor)	84
Gambar 3.33 <i>Class Diagram</i> Mengelola Permohonan Persetujuan (Kontraktor)	85
Gambar 3.34 <i>Class Diagram</i> Mengelola Persetujuan Pembelian (Kontraktor)....	85
Gambar 3.35 <i>Class Diagram</i> Melihat Detail Laporan Pengeluaran (Kontraktor)	86
Gambar 3.36 <i>Class Diagram</i> Mengelola Profil (Mandor)	87
Gambar 3.37 <i>Class Diagram</i> Melihat Piutang (Mandor)	87
Gambar 3.38 <i>Class Diagram</i> Melihat Permintaan Tertunda (Mandor)	88
Gambar 3.39 <i>Class Diagram</i> Permintaan Ditolak (Mandor)	88
Gambar 3.40 <i>Class Diagram</i> Mengelola Proyek (Mandor).....	89
Gambar 3.41 <i>Class Diagram</i> Persetujuan Pembelian (Mandor).....	89
Gambar 3.42 <i>Class Diagram</i> Mengelola Pengeluaran (Mandor)	90
Gambar 3.43 <i>Class Diagram</i> Mengelola Catatan (Mandor)	90

Gambar 3.44 <i>Class Diagram</i> Mengelola Perkembangan (Mandor)	91
Gambar 3.45 <i>Physical Data Model</i>	92
Gambar 3.46 Desain Halaman <i>Login</i> (Kontraktor).....	103
Gambar 3.47 Desain Menu Utama (Kontraktor).....	104
Gambar 3.48 Desain Menu <i>Houses</i> (Kontraktor)	105
Gambar 3.49 Desain Menu Terbayar (Kontraktor).....	105
Gambar 3.50 Desain Menu piutang (Kontraktor)	106
Gambar 3.51 Desain <i>Note</i> (Kontraktor)	107
Gambar 3.52 Desain <i>Progress</i> (Kontraktor)	107
Gambar 3.53 Desain Menu <i>Report</i> (Kontraktor)	108
Gambar 3.54 Desain Menu Notifikasi (Kontraktor)	109
Gambar 3.55 Desain Menu Mandor (Kontraktor).....	109
Gambar 3.56 Desain Gudang (Kontraktor).....	110
Gambar 3.57 Desain Menu Piutang (Kontraktor)	111
Gambar 3.58 Desain Halaman <i>Login</i> (Mandor).....	111
Gambar 3.59 Desain Menu Utama (Mandor)	112
Gambar 3.60 Desain Menu Notifikasi (Mandor)	113
Gambar 3.61 Desain <i>Houses</i> (Mandor).....	113
Gambar 3.62 Desain Menu Pengeluaran (Mandor)	114
Gambar 3.63 Desain Menu <i>Progress</i> (Mandor).....	115
Gambar 3.64 Desain Menu <i>Notes</i> (Mandor)	115
Gambar 3.65 Desain Menu <i>Pending Request</i> (Mandor)	116
Gambar 3.66 Desain Menu <i>Rejected Request</i> (Mandor).....	117
Gambar 3.67 Desain Menu Piutang (Mandor).....	118

Gambar 4.1 Proses <i>Login</i> Sebagai Kontraktor.....	160
Gambar 4.2 <i>Login</i> Dengan Data Salah.....	161
Gambar 4.3 <i>Login</i> Dengan <i>Password</i> Kosong	161
Gambar 4.4 <i>Login</i> Dengan <i>Email</i> Kosong	162
Gambar 4.5 <i>Login</i> Tanpa Masukan Data	162
Gambar 4.6 Tampilan Menu Profil	164
Gambar 4.7 Tampilan Menu <i>Houses</i>	165
Gambar 4.8 Tampilan Menu Mandor.....	165
Gambar 4.9 Tampilan Menu Gudang.....	166
Gambar 4.10 Tampilan Menu Piutang	166
Gambar 4.11 Tampilan Menu <i>Logout</i>	167
Gambar 4.12 Menambahkan Data <i>Houses</i>	170
Gambar 4.13 Menambahkan Data <i>Houses</i> Tanpa Alamat.....	170
Gambar 4.14 Menambahkan Data <i>Houses</i> Kosong	171
Gambar 4.15 Mengubah Data <i>Houses</i>	171
Gambar 4.16 Menghapus Data <i>Houses</i>	172
Gambar 4.17 Menambahkan Data <i>Cash</i> Menu Terbayar.....	177
Gambar 4.18 Menambahkan Data <i>Cash</i> Menu Terbayar Tanpa Nama Barang .	177
Gambar 4.19 Menambahkan Data Kosong <i>Cash</i> Menu Terbayar	178
Gambar 4.20 Menambahkan Data Gudang Menu Terbayar	178
Gambar 4.21 Menambah Data Gudang Menu Terbayar Tanpa Nama Barang ...	179
Gambar 4.22 Menambahkan Data Kosong Gudang Menu Terbayar.....	180
Gambar 4.23 Mengubah Data Terbayar.....	180

Gambar 4.24 Menambahkan Data Menu Piutang	183
Gambar 4.25 Menambahkan Data Menu Piutang Tanpa Nama Barang	183
Gambar 4.26 Menambahkan Data Kosong Menu Piutang.....	184
Gambar 4.27 Mengubah Data Menu Piutang.....	184
Gambar 4.28 Menambahkan Data Menu <i>Note</i>	186
Gambar 4.29 Mengubah Data Menu <i>Note</i>	187
Gambar 4.30 Menambahkan Data Kosong Menu <i>Note</i>	187
Gambar 4.31 Menghapus Data Menu <i>Note</i>	188
Gambar 4.32 Menambah Data <i>Work Start</i> Menu <i>Progress</i>	190
Gambar 4.33 Menambah Data Kosong <i>Work Start</i> Menu <i>Progress</i>	191
Gambar 4.34 Menambah Data <i>Work End</i> Menu <i>Progress</i>	191
Gambar 4.35. Menambah Data Kosong <i>Work End</i> Menu <i>Progress</i>	192
Gambar 4.36 Melihat <i>Report</i> Mingguan	193
Gambar 4.37 Melihat <i>Report</i> Harian.....	194
Gambar 4.38 Data Penerimaan Permohonan Menu Notifikasi.....	195
Gambar 4.39 Penerimaan Permohonan Menu Notifikasi	196
Gambar 4.40 Penolakan Permohonan Menu Notifikasi Data	196
Gambar 4.41 Penolakan Permohonan Menu Notifikasi.....	197
Gambar 4.42 Penambahan Data Menu Mandor	200
Gambar 4.43 Penambahan Data Menu Mandor Tanpa Gambar	201
Gambar 4.44 Penambahan Data Menu Mandor Tanpa Email	201
Gambar 4.45 Penambahan Data Kosong Menu Mandor	202
Gambar 4.46 Pengubahan Data Menu Mandor.....	203
Gambar 4.47 Penghapusan Data Menu Mandor	203

Gambar 4.48. Penambahan Data <i>Cash</i> Menu Gudang.....	207
Gambar 4.49. Penambahan Data <i>Cash</i> Menu Gudang Tanpa Nama Barang.....	208
Gambar 4.50. Penambahan Data Piutang Menu Gudang.....	208
Gambar 4.51 Penambahan Data Piutang Menu Gudang Tanpa Nama Barang... ..	209
Gambar 4.52 Pengubahan Data Menu Gudang.....	210
Gambar 4.53 Penghapusan Data Menu Gudang	211
Gambar 4.54 Pembayaran <i>Cash</i> Menu Bayar Piutang.....	212
Gambar 4.55 Pembayaran <i>Transfer</i> Menu Bayar Piutang	213
Gambar 4.56 Pembayaran <i>Transfer</i> Piutang Tanpa Nomor Rekening	214
Gambar 4.57 Proses <i>Login</i> Sebagai Mandor.....	216
Gambar 4.58 <i>Login</i> Dengan Data Salah.....	217
Gambar 4.59 <i>Login</i> Dengan Password Kosong.....	217
Gambar 4.60 <i>Login</i> Dengan Email Kosong	218
Gambar 4.61 <i>Login</i> Tanpa Masukan Data	219
Gambar 4.62 Menu <i>Profile</i> Mandor.....	221
Gambar 4.63 Menu <i>Houses</i> Mandor	221
Gambar 4.64 Menu <i>Pending Request</i> Mandor.....	222
Gambar 4.65 Menu <i>Rejected Request</i> Mandor.....	222
Gambar 4.66 Menu Piutang Mandor.....	223
Gambar 4.67 Menu <i>Logout</i> Mandor	223
Gambar 4.68 Menu Notifikasi Mandor.....	225
Gambar 4.69 Menu <i>Houses</i> Mandor	226
Gambar 4.70 Tambah Data Pengeluaran <i>Cash</i>	231
Gambar 4.71 Tambah Data Pengeluaran <i>Cash</i> Tanpa Nama Barang	231

Gambar 4.72 Tambah Data Pengeluaran <i>Cash Kosong</i>	232
Gambar 4.73 Tambah Data Pengeluaran Gudang.....	233
Gambar 4.74 Tambah Data Pengeluaran Gudang.....	233
Gambar 4.75 Tambah Data Kosong Pengeluaran Gudang	234
Gambar 4.76 Tampilan Detail Data Pengeluaran	235
Gambar 4.77 Tambah Data Pengeluaran <i>Piutang</i>	235
Gambar 4.78 Menambah Data <i>Work Start Menu Progress</i>	237
Gambar 4.79 Menambah Data Kosong <i>Work Start Menu Progress</i>	238
Gambar 4.80 Menambah Data <i>Work End Menu Progress</i>	239
Gambar 4.81 Menambah Data Kosong <i>Work End Menu Progress</i>	239
Gambar 4.82 Menambahkan Data <i>Menu Note</i>	241
Gambar 4.83 Mengubah Data <i>Menu Note</i>	242
Gambar 4.84 Menambahkan Data Kosong <i>Menu Note</i>	243
Gambar 4.85 Menghapus Data <i>Menu Note</i>	243
Gambar 4.86 Menu <i>Pending Request Mandor</i>	244
Gambar 4.87 Menu <i>Rejected Request Mandor</i>	245
Gambar 4.88 Menu <i>Piutang Mandor</i>	246

DAFTAR LAMPIRAN

Lampiran 2 Acuan Kebutuhan Data Aplikasi RAB <i>Type 40</i>	252
Lampiran 3 Acuan Kebutuhan Data Aplikasi RAB <i>Type 48</i>	255
Lampiran 4 Acuan Kebutuhan Data Aplikasi RAB <i>Type 50</i>	258



BAB I

PENDAHULUAN

1.1 Perumusan Masalah

Perusahaan PT Mandiri Karya Dua Saudara (PT MKDS) adalah perusahaan yang bergerak di bidang konstruksi perumahan. Saat ini PT MKDS sedang mengerjakan beberapa proyek di beberapa lokasi yang berbeda, yakni tiga proyek di kota Surabaya dan dua proyek di kota Manado. PT MKDS menyerahkan sepenuhnya pengerjaan dan pengawasan berjalannya proyek kepada satu kontraktor internal, sehingga tim kontraktor ini harus dapat mengetahui bagaimana perkembangan dan kegiatan seluruh proyek di setiap harinya.

PT MKDS pada setiap proyeknya memiliki mandor yang bertugas sebagai pengawas pengerjaan dan juga melakukan pembelian material yang dibutuhkan untuk pengerjaan proyek. Sampai saat ini, PT MKDS memiliki 17 mandor yang mengawasi 5 proyek yang sedang berjalan. Dengan proyek yang berada di tempat yang berbeda, kontraktor tidak mungkin lagi untuk melakukan pengawasan secara langsung pada setiap harinya, hal inilah yang menjadi pemicu terjadinya beberapa kondisi yang menghambat berjalannya proyek.

Kondisi yang menghambat pengerjaan proyek saat ini adalah sering sekali terjadi kecurangan yang dilakukan oleh mandor pada sektor pengadaan barang material karena tidak adanya pengumpulan bukti barang dan pengumpulan berkas nota yang masih tidak rapi, serta masih belum adanya sistem perizinan pembelian bahan material oleh setiap mandor sehingga pengeluaran harian tidak dapat di

kontrol. Salah satu kejadian yang terjadi pada tanggal 21 Desember 2016 lalu, adalah terjadinya kerjasama antara mandor dengan penjual material yang dapat mengubah nota pembelian sehingga total barang yang dibeli tidak sama dengan jumlah yang diterima. Kejadian itu sangat merugikan kontraktor pada sektor alokasi biaya proyek. Terjadinya hal tersebut dikarenakan tidak adanya catatan harian yang dapat digunakan sebagai bukti perkembangan harian proyek sehingga proses penggerjaan proyek tidak dapat terkontrol dengan baik.

Hal inilah yang mendasari dibuatnya aplikasi pengawasan kinerja mandor pada PT MKDS. Aplikasi akan dibuat berbasis *mobile apps* karena penggunaannya yang sangat mudah dan dapat digunakan kapan saja dan dimana saja selama terhubung dengan internet sehingga pelaporan, perizinan dan pengawasan terhadap pekerjaan mandor dapat dilakukan dengan efektif.

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah di atas, maka dapat dibuat perumusan masalah, yaitu bagaimana merancang dan membangun aplikasi pengawasan kinerja mandor berbasis *mobile apps* pada PT MKDS sehingga dapat memudahkan kontraktor untuk dapat mengawasi setiap kegiatan mandor disetiap proyek yang dikelola.

1.3 Pembatasan Masalah

Dengan pembahasan perumusan masalah di atas, maka dapat dibuat suatu pembatasan masalah sebagai berikut:

1. Aplikasi hanya meliputi pengawasan kinerja, pencatatan pengeluaran harian, dan perizinan pembelian bahan material oleh mandor kepada kontraktor.
2. Aplikasi tidak membahas tentang sistem keuangan keseluruhan pada PT MKDS.
3. Aplikasi tidak digunakan untuk mencetak dokumen apapun yang berhubungan dengan keuangan atau laporan pencatatan harian.
4. Aplikasi tidak membahas pencadangan data (*backup server*).
5. Aplikasi tidak membahas pengesahan (*validation*) nomor ponsel yang digunakan oleh mandor.
6. Aplikasi tidak membahas *Admin Control Panel* (ACP) pada aplikasi ini.
7. Aplikasi tidak membahas tentang pembayaran bahan baku kepada *supplier*.
8. Pembahasan aplikasi hanya sampai pada tahap implementasi.

1.4 Tujuan Penelitian

Dengan didasari oleh perumusan masalah di atas, maka dapat dibuat tujuan penelitian ini adalah bagaimana merancang dan membangun aplikasi pengawasan kinerja mandor berbasis *mobile apps* pada PT MKDS sehingga dapat memudahkan kontraktor untuk dapat mengawasi setiap mandor di setiap proyek yang dikelola.

1.5 Manfaat Penelitian

Dengan adanya penelitian ini, maka PT MKDS akan mendapatkan beberapa manfaat sebagai berikut:

1. Diharapkan mampu memberikan kontribusi pada pengawasan kinerja setiap mandor di setiap proyek yang dimiliki oleh PT MKDS.

2. Pembuatan aplikasi pengawasan kinerja mandor berbasis *mobile app* ini diharapkan dapat membantu mengurangi frekuensi kunjungan kontraktor ke setiap proyek untuk melakukan pengawasan langsung.
3. Dengan adanya pengawasan yang dilakukan kontraktor terhadap kinerja setiap mandor secara langsung (*real time*), dapat mengurangi terjadinya kecurangan yang dilakukan oleh mandor.

1.6 Sistematika Penulisan

Dalam penyusunan laporan tugas akhir digunakan sistematika penulisan yang dibagi kedalam beberapa bab, sebagai berikut:

BAB I

PENDAHULUAN

Bab pendahuluan ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan yang ingin dicapai, manfaat penelitian, serta sistematika dari penulisan laporan tugas akhir.

BAB II

LANDASAN TEORI

Bab landasan teori ini berisi penjelasan tentang pengertian dan penjelasan yang menjadi dasar dan menjadi pedoman dalam dibuatnya tugas akhir meliputi mandor, kontraktor, kinerja, rencana anggaran biaya, aplikasi, aplikasi *mobile*, *system development life cycle* serta penjelasan tentang *tools* yang digunakan untuk pembuatan tugas akhir meliputi *android studio* dan *sublime text 3*.

BAB III ANALISIS DAN PERANCANGAN SISTEM

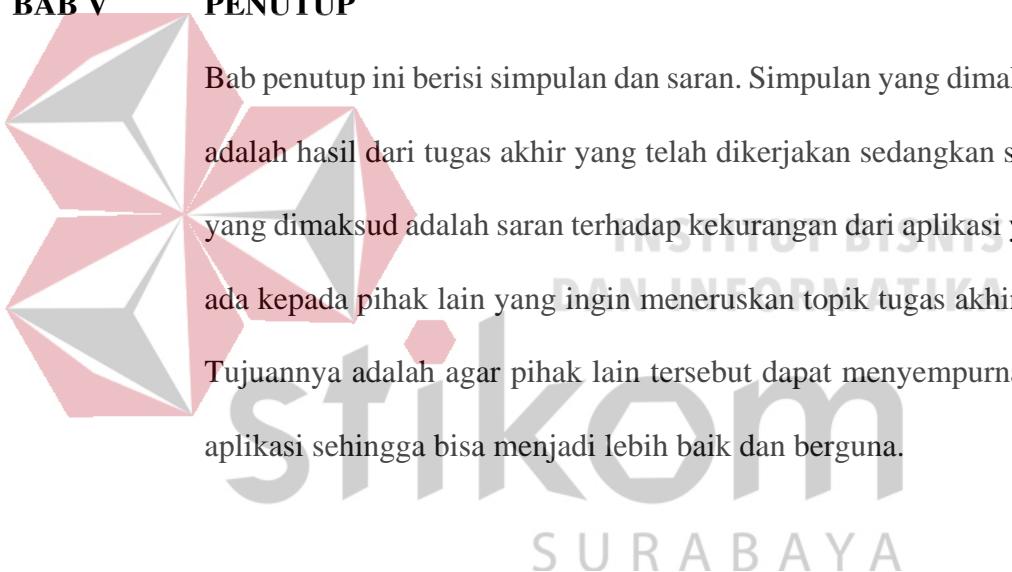
Bab analisis dan perancangan sistem ini berisi perancangan pembuatan sistem mulai dari *planning*, *analyze*, dan *design*. Sehingga dapat diketahui semua alur yang terdapat dalam sistem.

BAB IV IMPLEMENTASI DAN EVALUASI

Bab implementasi dan evaluasi ini berisi langkah pembuatan aplikasi, tampilan dari sistem yang telah dibuat dan telah melalui tahap uji coba sebelumnya.

BAB V PENUTUP

Bab penutup ini berisi simpulan dan saran. Simpulan yang dimaksud adalah hasil dari tugas akhir yang telah dikerjakan sedangkan saran yang dimaksud adalah saran terhadap kekurangan dari aplikasi yang ada kepada pihak lain yang ingin meneruskan topik tugas akhir ini. Tujuannya adalah agar pihak lain tersebut dapat menyempurnakan aplikasi sehingga bisa menjadi lebih baik dan berguna.



BAB II

LANDASAN TEORI

2.1 Mandor

Pengertian mandor pada Kamus Besar Bahasa Indonesia (2017) adalah mandor/man•dor/ n 1 orang yang mengepalai beberapa orang atau kelompok dan bertugas mengawasi pekerjaan mereka; 2 karyawan biasa yang tugasnya sama dengan tugas karyawan yang lain dan di samping itu merangkap tugas pengawasan atas rekan-rekannya.

Sumber lain mengatakan bahwa mandor merupakan salah satu bagian penting dalam pelaksanaan suatu proyek konstruksi karena mandor adalah seseorang yang bertugas langsung dalam pengawasan tukang saat bekerja di proyek konstruksi (Gunawan, 2013).

2.2 Kontraktor

Pada penelitian ini, kontraktor diartikan sebagai orang yang bertugas menjalankan sekaligus mengawasi jalannya seluruh proyek yang dimiliki oleh PT MKDS dengan tujuan agar proyek dapat berjalan sesuai dengan target dari sektor biaya yang telah ditetapkan.

Kontraktor adalah orang atau badan hukum yang menerima pekerjaan dan menyelenggarakan pelaksanaan pekerjaan sesuai dengan biaya yang telah ditetapkan berdasarkan gambar rencana, peraturan, dan syarat-syarat yang telah ditetapkan (Ervianto, 2005). Berikut merupakan tugas dan wewenang kontraktor:

1. Melaksanakan pekerjaan sesuai dengan gambar rencana, peraturan, syarat-syarat, risalah penjelasan pekerjaan, yang ditetapkan oleh pemilik proyek.
2. Membuat gambar-gambar pelaksanaan yang disahkan oleh konsultan manajemen konstruksi.
3. Membuat laporan hasil pekerjaan berupa laporan harian, mingguan, dan bulanan kepada konsultan manajemen konstruksi.
4. Menyediakan alat keselamatan kerja dan keamanan di lokasi proyek
5. Menyerahkan seluruh atau sebagian pekerjaan yang telah diselesaikan sesuai dengan ketetapan yang berlaku.



2.3 Kinerja

Menurut Sedarmayanti (2011) mengungkapkan bahwa kinerja merupakan terjemahan dari *performance* yang berarti hasil kerja seorang pekerja, sebuah proses manajemen atau suatu organisasi secara keseluruhan, dimana hasil kerja tersebut harus dapat ditunjukkan buktinya secara konkret dan dapat diukur (dibandingkan dengan standar yang telah ditentukan).

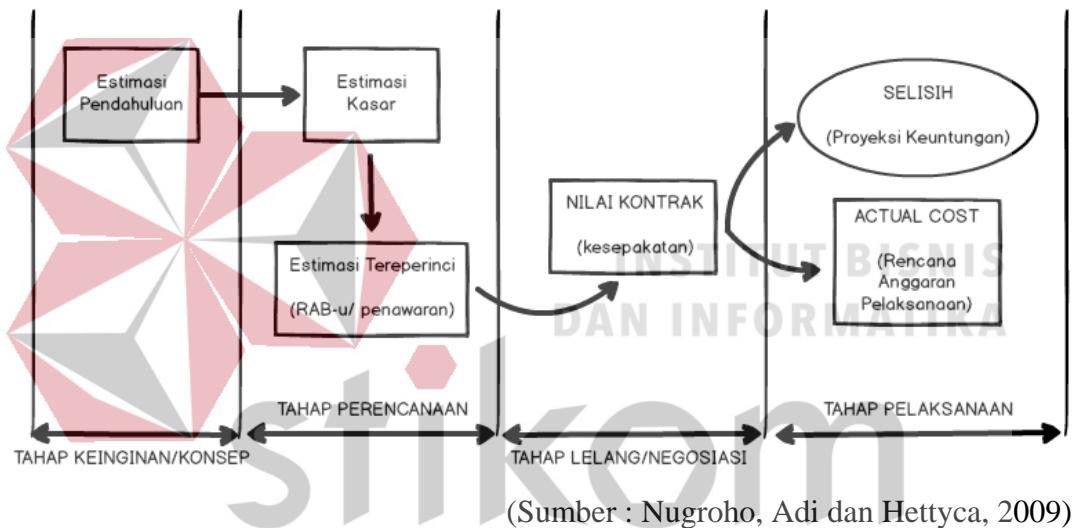
Dalam kasus pada PT MKDS ini, kinerja akan diukur sepenuhnya pada nilai efisiensi biaya yang digunakan dalam pembangunan proyek.

2.4 Rencana Anggaran Biaya

Menurut Nugroho, Adi dan Hettyca (2009) rencana anggaran biaya proyek adalah perhitungan banyaknya anggaran biaya suatu bangunan dan upah, serta biaya-biaya lain yang berhubungan dengan pelaksanaan proyek tersebut. Definisi lain mengatakan Rencana Anggaran Biaya (RAB) proyek adalah suatu proses

perhitungan *volume* pekerjaan, harga dari berbagai macam bahan dan pekerjaan yang terjadi pada suatu konstruksi.

Dari kedua definisi di atas dapat disimpulkan bahwa RAB proyek adalah perhitungan banyaknya biaya yang diperlukan dalam suatu proyek konstruksi yang terdiri dari biaya bahan, upah tenaga, serta biaya lain yang berhubungan dengan proyek tersebut berdasarkan perhitungan *volume* pekerjaan yang telah dilakukan sebelumnya.



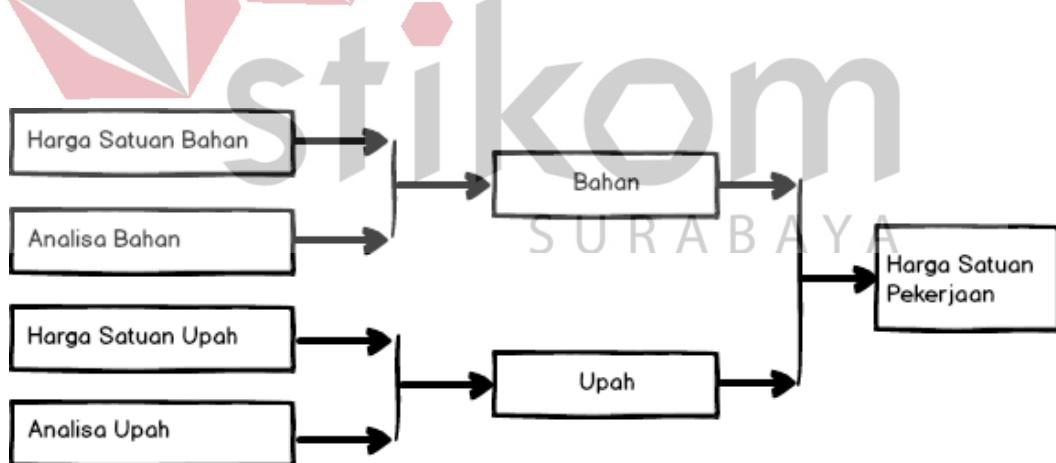
Gambar 0.1 Tahap dan Proses Penyusunan RAB

Berdasarkan Gambar 2.1, penyusunan RAB terbagi atas 2 bagian yaitu RAB terperinci dan RAB kasar. RAB kasar merupakan rencana anggaran biaya sementara dimana pekerjaan dihitung tiap ukuran luas. Pengalaman kerja sangat mempengaruhi penafsiran biaya secara kasar. Pada umumnya, hasil dari RAB kasar ini, apabila dibandingkan dengan RAB yang dihitung secara terperinci, akan terdapat selisih. Selain dari pengalaman, untuk proyek pemerintah biasanya sudah ditentukan pedoman harga satuan. RAB terperinci adalah anggaran biaya bangunan

atau proyek yang dihitung dengan terperinci dan cermat, sesuai dengan ketentuan dan syarat-syarat penyusunan anggaran biaya. Adapun untuk proyek pemerintah biasanya telah ditetapkan daftar tingkat upah, bahan dan harga alat.

Penyusunan RAB secara terperinci pada dasarnya membutuhkan 5 hal yang paling mendasar, yaitu bestek dan gambar-gambar bestek, daftar upah, daftar harga bahan-bahan (material), daftar analisis, serta daftar *volume* tiap jenis pekerjaan yang ada. Daftar tersebut dapat saling memberikan gambaran dan petunjuk-petunjuk hingga akhirnya dapat merupakan anggaran biaya. Di dalam RAB terdapat analisis harga satuan pekerjaan. Analisis harga satuan pekerjaan merupakan analisis bahan dan upah untuk membuat satu satuan pekerjaan tertentu.

Harga satuan pekerjaan terdiri atas tiga komponen, yaitu analisa harga satuan bahan/material, analisa harga satuan upah tenaga dan analisis harga satuan sewa alat yang bersifat opsiional. Untuk lebih jelasnya dapat dilihat pada Gambar 2.2.



(Sumber : Nugroho, Adi dan Hettyca, 2009)

Gambar 0.2 Harga Satuan Pekerjaan

Proses analisis harga satuan bahan/material pada dasarnya adalah menghitung banyaknya *volume* masing-masing bahan serta besarnya biaya yang dibutuhkan

untuk menyelesaikan per-satuan pekerjaan konstruksi. Analisis harga satuan bahan/material mengandung dua unsur yaitu: a) Harga satuan bahan, merupakan harga satuan bahan/material bangunan yang berlaku di pasar pada saat anggaran biaya bangunan tersebut disusun, dan b) Koefisien bahan, yaitu koefisien yang menunjukkan kebutuhan bahan/material bangunan untuk setiap satuan jenis pekerjaan.

Proses analisis harga satuan upah tenaga pada dasarnya adalah menghitung banyaknya tenaga serta biaya yang dibutuhkan, untuk menyelesaikan per-satuan pekerjaan konstruksi. Analisis harga satuan upah tenaga mengandung dua unsur yaitu: a) Harga satuan upah tenaga, merupakan upah yang diberikan kepada tenaga kerja konstruksi tiap harinya atas jasa tenaga yang dilakukan sesuai dengan keterampilannya, dan b) Koefisien tenaga, yaitu koefisien yang menunjukkan kebutuhan tenaga kerja untuk tiap-tiap posisi. Sementara itu analisis harga satuan sewa alat pada dasarnya adalah menghitung banyaknya alat yang digunakan serta besarnya biaya sewa alat, untuk menyelesaikan per-satuan pekerjaan konstruksi.

Analisis harga satuan sewa alat mengandung dua unsur, yaitu: a) Harga satuan sewa alat, merupakan harga satuan sewa alat yang berlaku di pasar pada saat anggaran biaya bangunan tersebut disusun, dan b) Koefisien alat, yaitu koefisien yang menunjukkan kebutuhan alat untuk setiap satuan jenis pekerjaan (Nugroho, Beeh, Astuningdyas, 2009).

2.5 Aplikasi

Menurut Rizal, Retnadi, dan Ikhwana (2013), aplikasi adalah penggunaan dalam suatu perangkat komputer, instruksi (*instruction*) atau pernyataan (*statement*)

yang disusun hingga sedemikian rupa komputer dapat memproses masukan (*input*) menjadi keluaran (*output*).

Dari pengertian di atas dapat disimpulkan bahwa aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna, atau aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

2.6 Aplikasi *Mobile*

Menurut Turban (2012), *mobile application* juga biasa disebut dengan *mobile apps*, yaitu istilah yang digunakan untuk mendeskripsikan aplikasi internet yang berjalan pada *smartphone* atau piranti *mobile* lainnya. Aplikasi *mobile* biasanya membantu para penggunanya untuk terkoneksi dengan layanan internet yang bisa diakses pada PC atau mempermudah mereka untuk menggunakan aplikasi internet pada piranti yang bisa dibawa.

Aplikasi *mobile* saat ini telah menjadi gaya hidup bagi seluruh manusia, karenanya dalam beberapa tahun terakhir ini, aplikasi *mobile* berkembang sangat pesat, sehingga kehadirannya saat ini membawa kemudahan bagi setiap penggunanya.

2.7 System Development Life Cycle (SDLC)

Metodologi pengembangan sistem informasi berarti suatu metode yang digunakan untuk melakukan pengembangan sistem informasi berbasis komputer. Metode yang paling umum digunakan adalah dengan siklus hidup pengembangan *System Development Life Cycle* (SDLC). SDLC merupakan metodologi klasik yang digunakan untuk mengembangkan, memelihara dan menggunakan sistem

informasi. Metode ini menggunakan pendekatan sistem yang disebut pendekatan air terjun (*waterfall approach*), yang menggunakan beberapa tahapan dalam mengembangkan sistem (Supriyanto, 2007). Adapun tahapan dalam metode SDLC adalah sebagai berikut:

1. Tahap Perencanaan Sistem (*Sistem Planning*). Tahap perencanaan adalah tahap awal pengembangan sistem yang mendefinisikan perkiraan kebutuhan-kebutuhan sumber daya seperti perangkat fisik, manusia, metode (teknik dan operasi) dan anggaran yang sifatnya masih umum (belum detail / rinci).

2. Tahap Analisis Sistem (*System Analyst*). Tahap analisis sistem adalah tahap penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbarui. Pada tahap analisa ini, akan digunakan metode UML (*Unified Model Language*) dimana akan digunakan langkah analisa mulai dari:

- a. *Use Case*

Pada diagram *use case*, akan diketahui interaksi antara *use case* dan aktor yang terkait dengan sistem. *Use case* menggambarkan fungsionalitas sistem atau persyaratan apa saja yang harus dipenuhi sistem dari pandangan pemakai, sedangkan aktor menggambarkan pengguna dari tiap *use case*.

- b. *Flow Of Event*

Pada *flow of events* akan dijelaskan detail dari tiap *use case* yang telah dibuat. *Flow of events* berfungsi menggambarkan alur logika

dari setiap *use case* yang telah digambarkan. Selain itu *flow of events* juga mendeskripsikan apa yang dilakukan oleh aktor dan sistem.

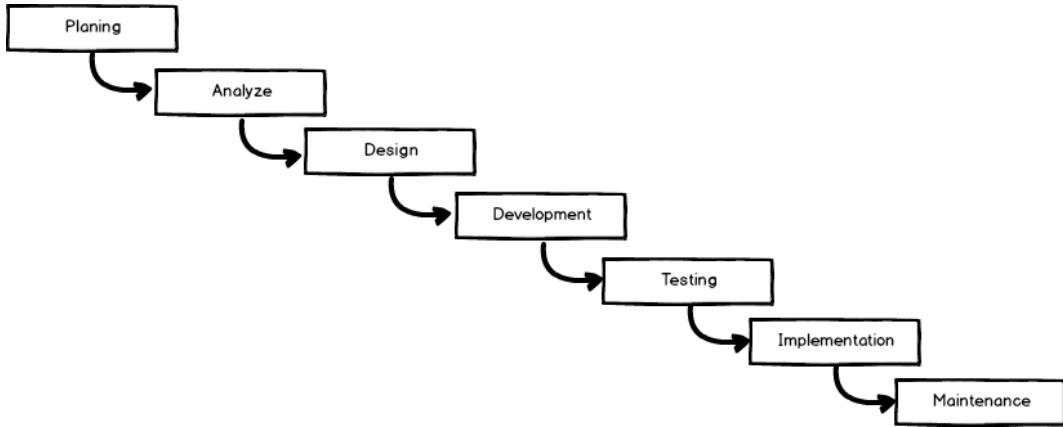
c. *Sequence Diagram*

Pada *sequence diagram* akan ditunjukkan alur fungsionalitas pada *use case* yang disusun dalam urutan waktu. Diagram sekuensial menggambarkan interaksi yang dilakukan oleh tiap objek dalam sistem. Kerjasama antar objek tersebut dilakukan dengan cara saling mengirimkan pesan yang membentuk suatu alur kerja sama.

d. *Class Diagram*

Pada *class diagram* akan ditunjukkan interaksi antar kelas pada sistem yang berhubungan langsung dengan *database*.

3. Tahap Perancangan/Desain Sistem (*System Design*). Tahap desain sistem adalah tahap setelah analisis sistem yang menentukan proses dan data yang diperlukan oleh sistem baru. Desain sistem dibedakan menjadi dua macam, yaitu desain sistem umum dan desain sistem terinci.
4. Tahap Penerapan/Implementasi Sistem (*System Implementation*). Tahap implementasi atau penerapan adalah tahap dimana desain sistem dibentuk menjadi suatu kode (program) yang siap untuk dioperasikan.
5. Tahap Pemeliharaan/Perawatan Sistem (*System Maintenance*). Tahap pemeliharaan/ perawatan sistem merupakan tahap yang dilakukan setelah tahap implementasi yang meliputi penggunaan sistem, audit sistem, penjagaan sistem, perbaikan sistem dan peningkatan sistem.



(Sumber : Supriyanto, 2007)

Gambar 0.3 Tahapan Metode *System Development Life Cycle*

2.8 Unified Modeling Language (UML)

Menurut Satzinger, Jackson dan Burd (2011) *Unified Modeling Language* (UML) merupakan sistem arsitektur yang bekerja dalam OOAD (*Object-Oriented Analysis/Design*) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkonstruksi, dan mendokumentasikan *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*, dapat berupa model, deskripsi, atau *software*) yang terdapat dalam sistem *software*. UML merupakan bahasa pemodelan yang paling sukses dari tiga metode OO yang telah ada sebelumnya, yaitu Booch, OMT (*Object Modeling Technique*), dan OOSE (*Object-Oriented Software Engineering*). UML merupakan kesatuan dari ketiga pemodelan tersebut dan ditambah kemampuan lebih karena mengandung metode tambahan untuk mengatasi masalah pemodelan yang tidak dapat ditangani ketiga metode tersebut. UML dikeluarkan oleh OMG (*Object Management Group, Inc*) yaitu organisasi internasional yang dibentuk pada 1989, terdiri dari perusahaan sistem informasi, *software developer*, dan para *user* sistem komputer.

Dengan adanya UML, diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. UML diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini.

Tujuan UML diantaranya adalah memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum, memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman, proses rekayasa dan menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan

2.9 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk aplikasi android. Tujuannya dibuat untuk android adalah untuk mempercepat pengembangan dan membantu pengguna untuk membuat aplikasi berkualitas tinggi pada setiap perangkat android. Menawarkan alat bantu yang dibuat khusus untuk pengembang android, meliputi pengeditan kode yang lengkap, *debugging*, pengujian, dan alat pembuatan profil. Android Studio merupakan IDE resmi untuk pengembangan aplikasi android yang disarankan oleh Google (Google LLC, 2018).

Pembuatan aplikasi pada android tidak hanya dapat dilakukan dengan android studio. Terdapat beberapa alternatif yang dapat menjadi pilihan seperti, *kotlin* dan *flutter*. Tetapi kembali pada penciptanya, google sangat menyarankan penggunaan android studio sebagai alat untuk pembuatan aplikasi pada android.

2.10 Sublime Text 3

Sublime Text 3 adalah sebuah *software* yang dikembangkan oleh Jon Skinner.

Sublime text 3 merupakan aplikasi *text editor* untuk menulis kode. Banyak sejumlah bahasa program yang ada pada aplikasi ini. Diantaranya PHP, CSS, C, C++, HTML, ASP, Java, dan sebagainya.

Tentu saja, *software* ini bisa lebih memudahkan pekerjaan pengguna saat membuat sebuah program. Sublime Text 3 juga memiliki beberapa keunggulan dibanding *text editor* lain seperti *snippet* dan *shortcut keyboard* yang sangat mudah digunakan, sehingga saat menulis kode program, tidak perlu lagi sering menggunakan kursor (Sublime HQ Pty Ltd, 2018)



BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada penelitian ini, metode yang digunakan adalah *System Development Life Cycle* (SDLC). Metode ini akan diterapkan pada penelitian pengawasan kinerja mandor ini dengan rincian tahapan sebagai berikut:

3.1 Planning

Pada tahap ini, dilakukan kegiatan pengumpulan data dari berbagai sumber yang dapat menunjang berjalannya penelitian ini, khususnya data penggerakan proyek yang dilakukan oleh PT MKDS dan *detail* langkah pengerjaannya. Berikut ini merupakan jenis data dan instrumen pengumpulan data yang digunakan dalam pembuatan aplikasi ini:

Tabel 3.1 Jenis Data dan Instrumen Pengumpulan Data

No	Variabel Penelitian	Sumber Data	Metode Pengumpulan data	Instrumen Pengumpulan data
1	Proses mendapatkan data dan laporan pengeluaran harian	Primer	Wawancara	Daftar pertanyaan dan catatan
2	Proses mendapatkan data dan laporan	Primer	Wawancara	Daftar pertanyaan dan catatan

No	Variabel Penelitian	Sumber Data	Metode Pengumpulan data	Instrumen Pengumpulan data
	pengeluaran mingguan			
3	Data awal pengerjaan proyek hingga pengerjaan dan target penyelesaian proyek	Primer	Wawancara	Daftar pertanyaan dan catatan
4	Kontraktor (pengguna aplikasi)	Primer	Wawancara	Daftar pertanyaan dan catatan
5	Mandor (pengguna aplikasi)	Primer	Wawancara	Daftar pertanyaan dan catatan

3.2 Analyze

Pada tahapan ini, akan dilakukan kegiatan untuk mengidentifikasi mengenai kebutuhan aplikasi dari proses pengawasan kinerja mandor pada PT MKDS. Berikut ini merupakan proses pengawasan kinerja mandor yang terjadi pada setiap pengerjaan proyek pada PT MKDS:



Gambar 0.1 Kondisi Proses Pengawasan Kinerja Proyek pada PT MKDS

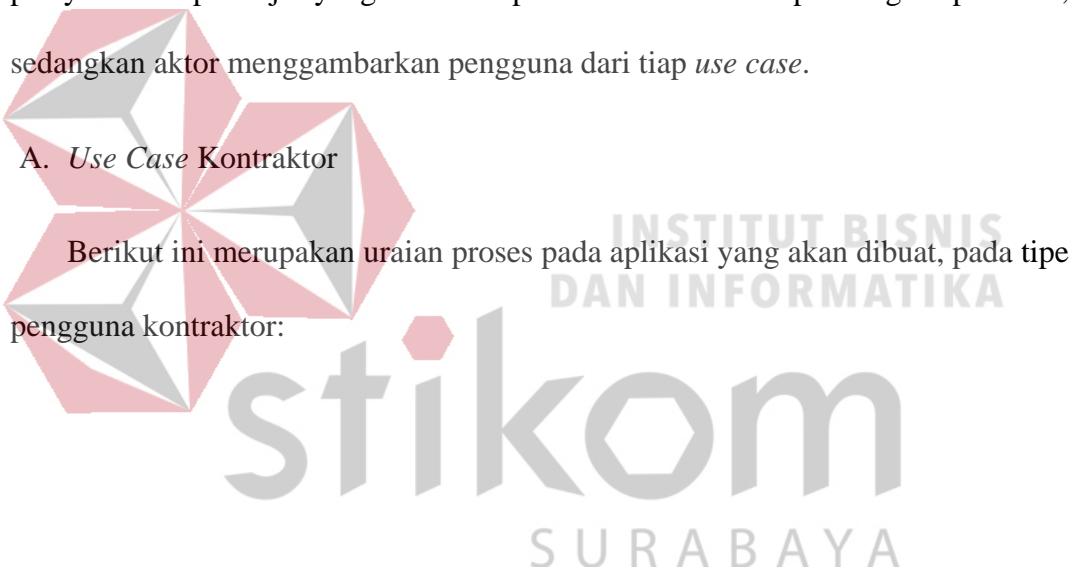
Berdasarkan pada gambar di atas, berikut ini merupakan penjelasan dari tiap langkah, proses pengawasan kinerja mandor yang terjadi saat ini:

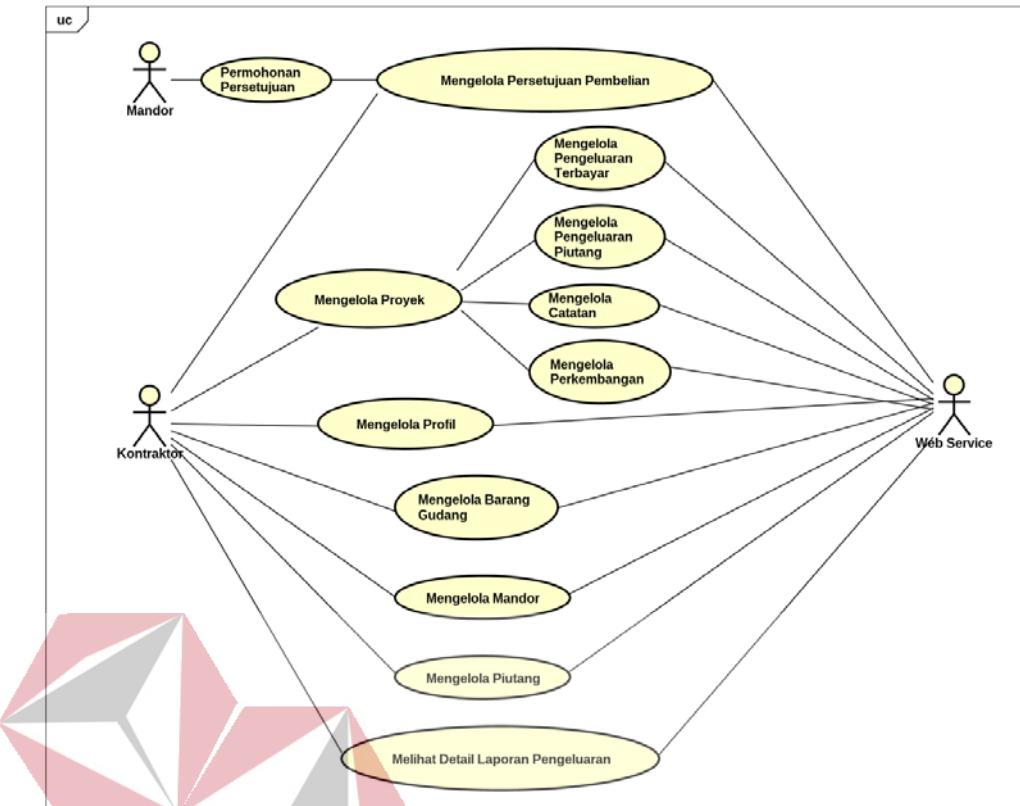
1. Kontraktor melakukan *survey* lokasi proyek
2. Kontraktor melakukan pembelian dan pencatatan stok di gudang
3. Kontraktor menentukan mandor yang akan melaksanakan proyek
4. Kontraktor menentukan anggaran yang sesuai dengan proyek
5. Mandor dan pekerja mulai melakukan pekerjaan proyek
6. Mandor melakukan pembelanjaan bahan baku pada supplier dengan persetujuan kontraktor
7. Kontraktor melakukan pengecekan dan persetujuan tentang bahan baku mana yang perlu dan dapat dibeli sendiri dan bahan baku mana yang dapat diambil dari gudang
8. Pengadaan bahan baku yang telah disetujui oleh kontraktor
9. Mandor mengumpulkan semua catatan pengeluaran pekerja
10. Kontraktor memeriksa dan mencatat semua pengeluaran kebutuhan bahan baku beserta dengan kebutuhan gaji pekerja

Dari kondisi di atas, akan dibuat aplikasi yang dapat memudahkan kontraktor dalam melakukan proses pengawasan terhadap kinerja mandor dimana pada aplikasi ini, akan terdapat dua pengguna aplikasi yakni kontraktor, dan mandor.

3.3 Use Case

Pada diagram *use case*, akan diketahui interaksi antara *use case* dan aktor yang terkait dengan sistem. *Use case* menggambarkan fungsionalitas sistem atau persyaratan apa saja yang harus dipenuhi sistem dari pandangan pemakai, sedangkan aktor menggambarkan pengguna dari tiap *use case*.





Gambar 0.2 Use Case Pengguna Kontraktor

Berikut ini merupakan penjelasan dari *use case* tipe pengguna kontraktor, pada aplikasi pengawasan kinerja mandor:

Tabel 3.2 Keterangan *Use Case* Pengguna Kontraktor

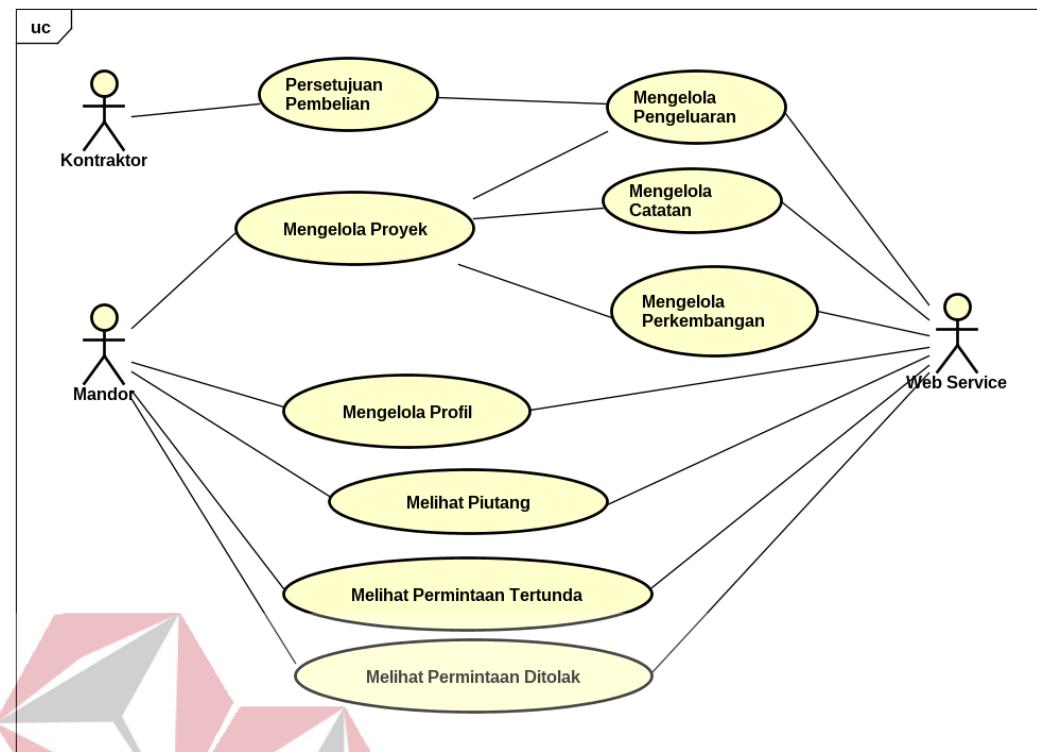
No	Use Case	Aktor Utama	Aktor Sekunder	Keterangan
1.	Mengelola Profil	Kontraktor	Tidak ada	Kontraktor dapat melihat dan mengubah data profil termasuk <i>password</i> yang digunakan untuk dapat mengakses aplikasi
2.	Mengelola Barang Gudang	Kontraktor	Tidak ada	Kontraktor dapat menambahkan, mengubah, dan

No	Use Case	Aktor Utama	Aktor Sekunder	Keterangan
				menghapus data bahan baku yang terdapat di gudang
3.	Mengelola Mandor	Kontraktor	Tidak ada	Kontraktor dapat menambahkan, menghapus dan mengubah data, termasuk <i>password</i> dari mandor yang digunakan untuk dapat mengakses aplikasi
4.	Mengelola Piutang	Kontraktor	Tidak ada	Kontraktor dapat membayarkan piutang yang telah jatuh tempo dengan pilihan pembayaran <i>cash</i> atau <i>transfer</i>
5.	Mengelola Proyek	Kontraktor	Tidak ada	Kontraktor dapat menambah, menghapus, atau mengubah data proyek
6.	Mengelola Pengeluaran Terbayar	Kontraktor	Tidak ada	Kontraktor dapat menambah, atau mengubah data pengeluaran terbayar
7.	Mengelola Pengeluaran Piutang	Kontraktor	Tidak ada	Kontraktor dapat menambah, atau mengubah data pengeluaran piutang
8.	Mengelola Catatan	Kontraktor	Tidak ada	Kontraktor dapat menambah, menghapus, atau mengubah data catatan
9.	Mengelola Perkembangan	Kontraktor	Tidak ada	Kontraktor dapat menambah data perkembangan pada awal jam kerja atau akhir jam kerja
10.	Permohonan Persetujuan	Kontraktor	Mandor	Kontraktor menerima daftar permintaan pembelian oleh

No	Use Case	Aktor Utama	Aktor Sekunder	Keterangan
				mandor yang ditugaskan terhadap satu proyek
11.	Mengelola Persetujuan Pembelian	Kontraktor	Mandor	Kontraktor dapat menerima atau menolak permintaan yang telah dilakukan oleh mandor yang ditugaskan terhadap satu proyek
12.	Melihat Detail Laporan Pengeluaran	Kontraktor	Tidak ada	Kontraktor dapat melihat detail pengeluaran per minggu, per hari, hingga per transaksi yang telah dilakukan pada proyek yang telah berjalan

B. Use Case Mandor

Diambil dari data yang telah didapat pada proses sebelumnya, dapat dibuat kebutuhan apa saja yang diperlukan oleh mandor. Hal ini terkait dengan fungsi yang akan dibuat pada aplikasi. Karena mandor sendiri akan banyak bekerja di lapangan, maka sebisa mungkin fungsi yang ada pada aplikasi tidak mengganggu atau menghambat kegiatan mandor tersebut. Berikut ini merupakan uraian proses pada aplikasi yang akan dibuat, sebagai tipe pengguna mandor:



Gambar 0.3 Use Case Pengguna Mandor

INSTITUT BISNIS
DAN INFORMATIKA

Berikut ini merupakan penjelasan dari *use case* tipe pengguna mandor, pada aplikasi pengawasan kinerja mandor.

Tabel 3.3 Keterangan Use Case Pengguna Mandor

No	Use Case	Aktor Utama	Aktor Sekunder	Keterangan
1.	Mengelola Profil	Mandor	Tidak ada	Mandor dapat melihat dan mengubah data profil termasuk <i>password</i> yang digunakan untuk dapat mengakses aplikasi
2.	Melihat Piutang	Mandor	Tidak ada	Mandor dapat melihat piutang pengeluaran proyek sehingga dapat ikut mengingatkan kontraktor

No	Use Case	Aktor Utama	Aktor Sekunder	Keterangan
				agar tidak terjadi keterlambatan pembayaran
3.	Melihat Permintaan Tertunda	Mandor	Tidak ada	Mandor dapat melihat daftar permintaan pembelian yang telah disetujui oleh kontraktor
4.	Melihat Permintaan Ditolak	Mandor	Tidak ada	Mandor dapat melihat daftar permintaan pembelian yang telah ditolak oleh kontraktor
5.	Mengelola Proyek	Mandor	Tidak ada	Mandor dapat mengelola proyek yang ditugaskan oleh kontraktor untuk dirinya
6.	Persetujuan Pembelian	Mandor	Kontraktor	Mandor dapat melihat <i>list</i> persetujuan kontraktor yang masih belum diterima oleh kontraktor
7.	Mengelola Pengeluaran	Mandor	Tidak ada	Mandor dapat menambahkan data pengeluaran untuk pembelian bahan baku
8.	Mengelola Catatan	Mandor	Tidak ada	Mandor dapat menambah, menghapus, atau mengubah data catatan
9	Mengelola Perkembangan	Mandor	Tidak ada	Mandor dapat menambah data perkembangan pada awal jam kerja atau akhir jam kerja

3.4 Flow of Event

Pada *flow of events* akan dijelaskan detail dari tiap *use case* yang telah dibuat. *Flow of events* berfungsi menggambarkan alur logika dari setiap *use case* yang telah digambarkan. Selain itu *flow of events* juga mendeskripsikan apa yang dilakukan oleh aktor dan sistem.

A. *Flow of Event* Kontraktor

Berikut merupakan *flow of events* dari tipe pengguna kontraktor pada aplikasi pengawasan kinerja mandor:



Nama Use Case	Mengelola Profil
Tujuan	Kontraktor dapat melihat dan mengubah data profil termasuk <i>password</i> yang digunakan untuk dapat mengakses aplikasi
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>
Kondisi Akhir Sukses	Data profil dapat tampil
Kondisi Akhir Gagal	Data profil yang telah diubah tidak tersimpan
Aktor Utama	Kontraktor

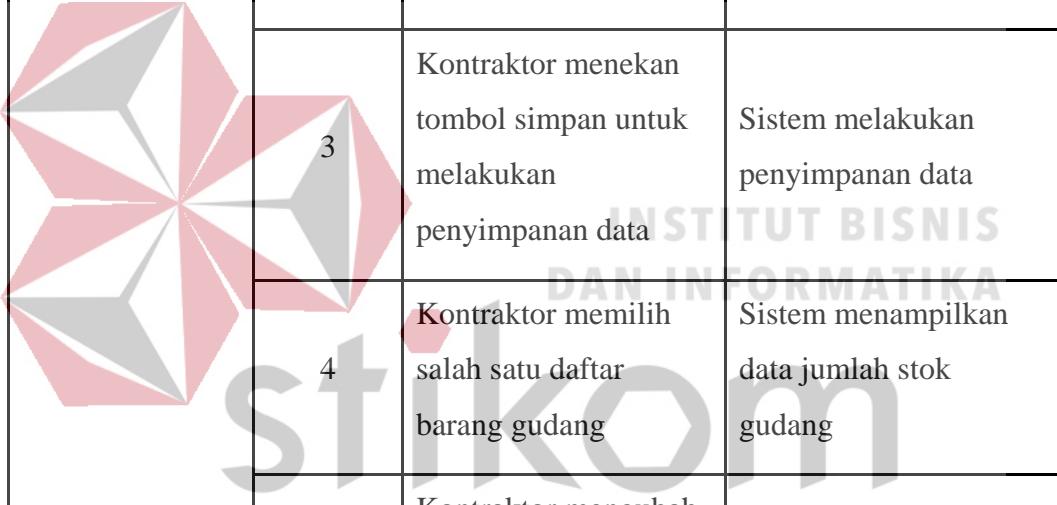
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih foto profil pada menu utama		
	Langkah	Aktor	Sistem
	1	Kontraktor memilih foto profil pada menu utama	Sistem menampilkan data profil
	2	Kontraktor mengisi data profil dan menekan tombol simpan	Sistem menampilkan informasi data telah berhasil tersimpan
	3	Kontraktor mengubah data profil dan menekan tombol simpan	Sistem menampilkan respon perubahan dan menampilkan data terbaru
	4	Kontraktor dapat menghapus data profil dengan menekan “icon trash”	Sistem menampilkan respon penghapusan data
Alur Perluasan	<p style="text-align: center;">Aksi Pencegahan</p> <p>Jika data yang disimpan tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan</p>		

2. *Flow of Events* Mengelola Barang Gudang

Flow of events mengelola barang gudang bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.5 *Flow of Events* Mengelola Barang Gudang (Kontraktor)

Nama Use Case	Mengelola Barang Gudang		
Tujuan	Kontraktor dapat menambahkan, mengubah, dan menghapus data bahan baku yang terdapat di gudang		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data barang gudang tampil, pengubahan dan penambahan data berhasil, serta data dapat dihapus		
Kondisi Akhir Gagal	Data barang gudang baru atau yang akan diubah tidak dapat tersimpan, dan data tidak dapat terhapus		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Gudang” pada menu utama		
Alur Utama	Langkah	Aktor	Sistem



	1	Kontraktor memilih menu “Gudang” pada menu utama	Sistem menampilkan daftar data proyek yang sedang dikerjakan, yang tersimpan dalam <i>database server</i>
Aktor	Sistem	Langkah	
2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan form yang harus diisi untuk menambahkan data	
3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data	
4	Kontraktor memilih salah satu daftar barang gudang	Sistem menampilkan data jumlah stok gudang	
5	Kontraktor mengubah data jumlah stok gudang dan menekan tombol simpan	Sistem melakukan penyimpanan data	
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

3. Flow of Events Mengelola Mandor

Flow of events mengelola mandor bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.



Tabel 3.6 *Flow of Events* Mengelola Mandor (Kontraktor)

Nama Use Case	Mengelola Mandor		
Tujuan	Kontraktor dapat menambahkan, menghapus dan mengubah data, termasuk <i>password</i> dari mandor yang digunakan untuk dapat mengakses aplikasi		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data mandor tampil, pengubahan dan penambahan data berhasil, serta data dapat dihapus		
Kondisi Akhir Gagal	Data mandor baru atau yang akan diubah tidak dapat tersimpan, dan data tidak dapat terhapus		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Mandor” pada menu utama		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Mandor” pada menu utama	Sistem menampilkan daftar data mandor
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data

	Langkah	Aktor	Sistem
	4	Kontraktor memilih salah satu daftar mandor	Sistem menampilkan detail data mandor
	5	Kontraktor mengubah data proyek dan menekan tombol simpan untuk menyimpan data	Sistem melakukan penyimpanan data
Alur Perluasan		Aksi Pencegahan	
Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan			

4. *Flow of Events* Mengelola Piutang

Flow of events mengelola piutang bertujuan untuk dapat mengetahui

alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.7 *Flow of Events* Mengelola Piutang (Kontraktor)

Nama Use Case	Mengelola Piutang
Tujuan	Kontraktor dapat membayarkan piutang yang telah jatuh tempo dengan pilihan pembayaran <i>cash</i> atau <i>transfer</i>
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>
Kondisi Akhir Sukses	Data piutang yang telah sampai pada tanggal bayarnya akan tampil, dan pengubahan status data berhasil

Kondisi Akhir Gagal	Perubahan status data piutang yang telah sampai pada tanggal bayarnya tidak dapat tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Piutang” pada menu utama		
	Langkah	Aktor	Sistem
Alur Utama	1	Kontraktor memilih menu “Piutang” pada menu utama	Sistem menampilkan daftar data piutang
	2	Kontraktor memilih salah satu daftar piutang	Sistem menampilkan detail data mandor
	5	Kontraktor memilih cara bayar, memasukkan data dan menekan tombol simpan	Sistem melakukan penyimpanan data
	Aksi Pencegahan		
Alur Perluasan	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

5. *Flow of Events* Mengelola Proyek

Flow of events mengelola proyek bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.8 *Flow of Events* Mengelola Proyek (Kontraktor)

Nama Use Case	Mengelola Proyek		
Tujuan	Kontraktor dapat menambah, menghapus, atau mengubah data proyek		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data proyek tampil, pengubahan dan penambahan data berhasil, serta data dapat dihapus		
Kondisi Akhir Gagal	Data proyek baru atau yang akan diubah tidak dapat tersimpan, dan data tidak dapat terhapus		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Houses” pada menu utama		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Houses” pada menu utama	Sistem menampilkan daftar data proyek
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data

	Aktor	Sistem	Langkah
	4	Kontraktor memilih salah satu daftar proyek	Sistem menampilkan menu proyek
	5	Kontraktor memilih “icon menu” pada tampilan menu proyek	Sistem menampilkan data detail proyek
	6	Kontraktor mengubah data proyek dan menekan tombol simpan	Sistem melakukan penyimpanan data
Aksi Pencegahan			
Alur Perluasan			
			Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan

6. Flow of Events Mengelola Pengeluaran Terbayar

Flow of events mengelola barang gudang bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.9 *Flow of Events* Mengelola Pengeluaran Terbayar (Kontraktor)

Nama Use Case	Mengelola Pengeluaran Terbayar
Tujuan	Kontraktor dapat menambah, atau mengubah data pengeluaran terbayar.
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>

Kondisi Akhir Sukses	Data pengeluaran terbayar tampil, pengubahan dan penambahan data berhasil		
Kondisi Akhir Gagal	Data pengeluaran terbayar baru atau yang akan diubah tidak dapat tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Terbayar” pada menu proyek		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Terbayar” pada menu proyek	Sistem menampilkan daftar data pengeluaran terbayar
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data
	4	Kontraktor memilih salah satu daftar pengeluaran terbayar yang tampil pada aplikasi	Sistem menampilkan <i>detail</i> data pengeluaran terbayar

	Aktor	Sistem	Langkah
	5	Kontraktor memilih “icon edit” pada tampilan detail pengeluaran terbayar	Sistem menampilkan data <i>form detail</i> pengeluaran terbayar
	6	Kontraktor mengubah data pengeluaran dan menekan tombol simpan	Sistem melakukan penyimpanan data
Alur Perluasan		Aksi Pencegahan	
		Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan	

7. *Flow of Events* Mengelola Pengeluaran Piutang

Flow of events mengelola pengeluaran piutang bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.10 *Flow of Events* Mengelola Pengeluaran Piutang (Kontraktor)

Nama Use Case	Mengelola Pengeluaran Piutang
Tujuan	Kontraktor dapat menambah, atau mengubah data pengeluaran piutang
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>

Kondisi Akhir Sukses	Data pengeluaran piutang tampil, pengubahan dan penambahan data berhasil		
Kondisi Akhir Gagal	Data pengeluaran piutang baru atau yang akan diubah tidak dapat tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Piutang” pada menu proyek		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Piutang” pada menu proyek	Sistem menampilkan daftar data pengeluaran piutang
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan form yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data
	4	Kontraktor memilih salah satu daftar pengeluaran piutang	Sistem menampilkan detail data pengeluaran piutang yang terdapat dalam database

	Langkah	Aktor	Sistem
	5	Kontraktor memilih “icon edit” pada tampilan detail pengeluaran piutang	Sistem menampilkan data <i>form detail</i> pengeluaran piutang
	6	Kontraktor mengubah data pengeluaran dan menekan tombol simpan	Sistem melakukan penyimpanan data
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

8. *Flow of Events* Mengelola Catatan

Flow of events mengelola catatan bertujuan untuk dapat mengetahui

alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.11 *Flow of Events* Mengelola Catatan (Kontraktor)

Nama Use Case	Mengelola Catatan
Tujuan	Kontraktor dapat menambah, menghapus, atau mengubah data catatan
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>
Kondisi Akhir Sukses	Data catatan berhasil tampil, pengubahan dan penambahan data berhasil, dan penghapusan data berhasil
Kondisi Akhir Gagal	Data catatan baru atau yang akan diubah tidak dapat tersimpan, dan data tidak dapat terhapus

Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Notes” pada menu proyek		
	Langkah	Aktor	Sistem
Alur Utama	1	Kontraktor memilih menu “Notes” pada menu proyek	Sistem menampilkan daftar data catatan
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan	Sistem melakukan penyimpanan data
	4	Kontraktor memilih salah satu daftar catatan	Sistem menampilkan <i>detail</i> data catatan
	5	Kontraktor mengubah data pengeluaran dan menekan tombol simpan	Sistem melakukan penyimpanan data
	6	Kontraktor memilih “icon trash” pada tampilan detail catatan	Sistem menghapus data catatan

	Aksi Pencegahan
Alur Perluasan	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan

9. *Flow of Events* Mengelola Perkembangan

Flow of events mengelola perkembangan bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.12 *Flow of Events* Mengelola Perkembangan (Kontraktor)

Nama Use Case	Mengelola Perkembangan
Tujuan	Kontraktor dapat menambah data perkembangan pada awal jam kerja atau akhir jam kerja.
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>
Kondisi Akhir Sukses	Data perkembangan berhasil tampil, penambahan data berhasil
Kondisi Akhir Gagal	Data catatan baru tidak dapat tersimpan
Aktor Utama	Kontraktor
Aktor Sekunder	Tidak ada

Pemicu	Kontraktor memilih menu “ <i>Progres</i> ” pada menu proyek.		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “ <i>Progres</i> ” pada menu proyek	Sistem menampilkan daftar data perkembangan
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
Alur Perluasan	3	Kontraktor menekan tombol simpan	Sistem melakukan penyimpanan data
	Aksi Pencegahan		
Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan			

10. *Flow of Events* Mengelola Permohonan Persetujuan

Flow of events mengelola permohonan persetujuan bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.13 *Flow of Events* Mengelola Permohonan Persetujuan (Kontraktor)

Nama Use Case	Mengelola Permohonan Persetujuan
Tujuan	Kontraktor menerima daftar permintaan pembelian oleh mandor yang ditugaskan terhadap satu proyek

Prasyarat	Kontraktor dalam keadaan telah login		
Kondisi Akhir Sukses	Data permohonan persetujuan berhasil tampil		
Kondisi Akhir Gagal	Data permohonan pembelian tidak dapat tampil		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Notifikasi” pada menu houses		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Notifikasi” pada menu <i>Houses</i>	Sistem menampilkan daftar data permohonan pembelian
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

11. *Flow of Events* Mengelola Persetujuan Pembelian

Flow of events mengelola persetujuan pembelian bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.14 *Flow of Events* Mengelola Persetujuan Pembelian (Kontraktor)

Nama Use Case	Mengelola Permohonan Pembelian		
Tujuan	Kontraktor dapat menerima atau menolak permintaan yang telah dilakukan oleh mandor yang ditugaskan terhadap satu proyek		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data permohonan pembelian berhasil tampil, dan pengubahan data status berhasil		
Kondisi Akhir Gagal	Perubahan data permohonan pembelian tidak dapat tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih salah satu data pada menu “Notifikasi”		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih salah satu daftar permohonan	Sistem menampilkan <i>detail</i> data permohonan
	2	Kontraktor menekan tombol setuju atau tidak setuju	Sistem melakukan penyimpanan data
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

12. *Flow of Events* Melihat Detail Laporan Pengeluaran

Flow of events melihat *detail* laporan pengeluaran bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.15 *Flow of Events* Melihat *Detail* Laporan Pengeluaran (Kontraktor)

Nama Use Case	Melihat Detil Laporan Pengeluaran		
Tujuan	Kontraktor dapat melihat detail pengeluaran per minggu, per hari, hingga per transaksi yang telah dilakukan pada proyek yang telah berjalan		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data <i>detail</i> pengeluaran dapat dilihat dari rangkuman mingguan hingga ke tiap transaksi		
Kondisi Akhir Gagal	Data <i>detail</i> pengeluaran tidak tampil atau tidak sesuai		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “ <i>Report</i> ” pada menu proyek		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “ <i>Report</i> ” pada menu proyek	Sistem menampilkan <i>detail</i> data pengeluaran mingguan

	Aktor	Sistem	Langkah
	2	Kontraktor memilih data mingguan	Sistem menampilkan <i>detail</i> data per hari pada minggu yang dipilih
	3	Kontraktor memilih data harian	Sistem menampilkan <i>detail</i> data per pengeluaran pada hari yang dipilih
	4	Kontraktor memilih data pengeluaran	Sistem menampilkan <i>detail</i> data pengeluaran yang dipilih
Aksi Pencegahan			
Alur Perluasan	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

B. *Flow of Event* Mandor

Berikut merupakan *flow of events* dari tipe pengguna kontraktor pada aplikasi pengawasan kinerja mandor:

1. *Flow of Events* Mengelola Profil

Flow of events mengelola profil bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.16 *Flow of Events* Mengelola Profil (Mandor)

Nama Use Case	Mengelola Profil		
Tujuan	Kontraktor dapat melihat dan mengubah data profil termasuk <i>password</i> yang digunakan untuk dapat mengakses aplikasi		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data profil dapat tampil		
Kondisi Akhir Gagal	Data profil yang telah diubah tidak tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih foto profil pada menu utama		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih foto profil pada menu utama	Sistem menampilkan data profil
	2	Kontraktor mengisi data profil dan menekan tombol simpan	Sistem menampilkan informasi data telah berhasil tersimpan

	Aktor	Sistem	Langkah
	3	Kontraktor mengubah data profil dan menekan tombol simpan	Sistem menampilkan respon perubahan dan menampilkan data terbaru
	4	Kontraktor dapat menghapus data profil dengan menekan “icon trash”	Sistem menampilkan respon penghapusan data
		Aksi Pencegahan	
Jika data yang disimpan tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan			

2. *Flow of Events* Mengelola Barang Gudang

Flow of events mengelola barang gudang bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.17 *Flow of Events* Mengelola Barang Gudang (Mandor)

Nama Use Case	Mengelola Barang Gudang
Tujuan	Kontraktor dapat menambahkan, mengubah, dan menghapus data bahan baku yang terdapat di gudang
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>

Kondisi Akhir Sukses	Data barang gudang tampil, pengubahan dan penambahan data berhasil, serta data dapat dihapus		
Kondisi Akhir Gagal	Data barang gudang baru atau yang akan diubah tidak dapat tersimpan, dan data tidak dapat terhapus		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Gudang” pada menu utama		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Gudang” pada menu utama	Sistem menampilkan daftar data proyek
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan form yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data
	4	Kontraktor memilih salah satu daftar barang gudang	Sistem menampilkan data jumlah stok gudang
	5	Kontraktor mengubah data jumlah stok gudang dan menekan tombol simpan	Sistem melakukan penyimpanan data

Alur Perluasan	Aksi Pencegahan
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan

3. *Flow of Events* Mengelola Mandor

Flow of events mengelola mandor bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.18 *Flow of Events* Mengelola Mandor (Mandor)

Nama Use Case	Mengelola Mandor
Tujuan	Kontraktor dapat menambahkan, menghapus dan mengubah data, termasuk <i>password</i> dari mandor yang digunakan untuk dapat mengakses aplikasi
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>
Kondisi Akhir Sukses	Data mandor tampil, pengubahan dan penambahan data berhasil, serta data dapat dihapus
Kondisi Akhir Gagal	Data mandor baru atau yang akan diubah tidak dapat tersimpan, dan data tidak dapat terhapus
Aktor Utama	Kontraktor
Aktor Sekunder	Tidak ada
Pemicu	Kontraktor memilih menu “Mandor” pada menu utama

	Langkah	Aktor	Sistem
Alur Utama	1	Kontraktor memilih menu “Mandor” pada menu utama	Sistem menampilkan daftar data mandor
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan form yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data
	4	Kontraktor memilih salah satu daftar mandor	Sistem menampilkan detail data mandor
	5	Kontraktor mengubah data proyek dan menekan tombol simpan	Sistem melakukan penyimpanan data
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

4. *Flow of Events* Mengelola Piutang

Flow of events mengelola piutang bertujuan untuk dapat mengetahui

alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.19 *Flow of Events* Mengelola Piutang (Mandor)

Nama Use Case	Mengelola Piutang		
Tujuan	Kontraktor dapat membayarkan piutang yang telah jatuh tempo dengan pilihan pembayaran <i>cash</i> atau <i>transfer</i>		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data piutang yang telah sampai pada tanggal bayarnya akan tampil, dan pengubahan status data berhasil		
Kondisi Akhir Gagal	Perubahan status data piutang yang telah sampai pada tanggal bayarnya tidak dapat tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Piutang” pada menu utama		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Piutang” pada menu utama	Sistem menampilkan daftar data piutang yang tersimpan dalam database
	2	Kontraktor memilih salah satu daftar piutang	Sistem menampilkan detail data mandor
	3	Kontraktor memilih cara bayar, memasukkan data dan menekan tombol simpan	Sistem melakukan penyimpanan data

Aksi Pencegahan	
Alur Perluasan	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan

5. *Flow of Events* Mengelola Proyek

Flow of events mengelola proyek bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.20 *Flow of Events* Mengelola Proyek (Mandor)

Nama Use Case	Mengelola Proyek		
Tujuan	Kontraktor dapat menambah, menghapus, atau mengubah data proyek		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data proyek tampil, pengubahan dan penambahan data berhasil, serta data dapat dihapus		
Kondisi Akhir Gagal	Data proyek baru atau yang akan diubah tidak dapat tersimpan, dan data tidak dapat terhapus		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “ <i>Houses</i> ” pada menu utama		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu	Sistem menampilkan daftar data proyek

	Langkah	Aktor	Sistem
		“Houses” pada menu utama	
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data
	4	Kontraktor memilih salah satu daftar proyek	Sistem menampilkan menu proyek
	5	Kontraktor memilih “icon menu” pada tampilan menu proyek yang terdapat pada aplikasi setelah data muncul	Sistem menampilkan data <i>detail</i> proyek
	6	Kontraktor mengubah data proyek dan menekan tombol simpan	Sistem melakukan penyimpanan data
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

6. *Flow of Events* Mengelola Pengeluaran Terbayar

Flow of events mengelola barang gudang bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.21 *Flow of Events* Mengelola Pengeluaran Terbayar (Mandor)

Nama Use Case	Mengelola Pengeluaran Terbayar		
Tujuan	Kontraktor dapat menambah, atau mengubah data pengeluaran terbayar.		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data pengeluaran terbayar tampil, pengubahan dan penambahan data berhasil		
Kondisi Akhir Gagal	Data pengeluaran terbayar baru atau yang akan diubah tidak dapat tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Terbayar” pada menu proyek		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Terbayar” pada menu proyek	Sistem menampilkan daftar data pengeluaran terbayar

	Langkah	Aktor	Sistem
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data
	4	Kontraktor memilih salah satu daftar pengeluaran terbayar	Sistem menampilkan <i>detail</i> data pengeluaran terbayar
	5	Kontraktor memilih “icon edit” pada tampilan <i>detail</i> pengeluaran terbayar	Sistem menampilkan data <i>form detail</i> pengeluaran terbayar
	6	Kontraktor mengubah data pengeluaran dan menekan tombol simpan	Sistem melakukan penyimpanan data
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

7. *Flow of Events* Mengelola Pengeluaran Piutang

Flow of events mengelola pengeluaran piutang bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.22 *Flow of Events* Mengelola Pengeluaran Piutang (Mandor)

Nama Use Case	Mengelola Pengeluaran Piutang		
Tujuan	Kontraktor dapat menambah, atau mengubah data pengeluaran piutang		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data pengeluaran piutang tampil, pengubahan dan penambahan data berhasil		
Kondisi Akhir Gagal	Data pengeluaran piutang baru atau yang akan diubah tidak dapat tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Piutang” pada menu proyek		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Piutang” pada menu proyek	Sistem menampilkan daftar data pengeluaran piutang

	Langkah	Aktor	Sistem
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan untuk melakukan penyimpanan data	Sistem melakukan penyimpanan data
	4	Kontraktor memilih salah satu daftar pengeluaran piutang	Sistem menampilkan <i>detail</i> data pengeluaran piutang
	5	Kontraktor memilih icon <i>edit</i> pada tampilan <i>detail</i> pengeluaran piutang	Sistem menampilkan data <i>form detail</i> pengeluaran piutang
	6	Kontraktor mengubah data pengeluaran dan menekan tombol simpan	Sistem melakukan penyimpanan data
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

8. *Flow of Events* Mengelola Catatan

Flow of events mengelola catatan bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.23 *Flow of Events* Mengelola Catatan (Mandor)

Nama Use Case	Mengelola Catatan		
Tujuan	Kontraktor dapat menambah, menghapus, atau mengubah data catatan		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data catatan berhasil tampil, pengubahan dan penambahan data berhasil, dan penghapusan data berhasil		
Kondisi Akhir Gagal	Data catatan baru atau yang akan diubah tidak dapat tersimpan, dan data tidak dapat terhapus		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Notes” pada menu proyek		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Notes” pada menu proyek	Sistem menampilkan daftar data catatan yang telah tersimpan dalam <i>database</i> .

	Langkah	Aktor	Sistem
	2	Kontraktor memilih “icon plus” untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan	Sistem melakukan penyimpanan data
	4	Kontraktor memilih salah satu daftar catatan	Sistem menampilkan <i>detail</i> data catatan
	5	Kontraktor mengubah data pengeluaran dan menekan tombol simpan	Sistem melakukan penyimpanan data
	6	Kontraktor memilih “icon trash” pada tampilan <i>detail</i> catatan	Sistem menghapus data catatan
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

9. *Flow of Events* Mengelola Perkembangan

Flow of events mengelola perkembangan bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.24 *Flow of Events* Mengelola Perkembangan (Mandor)

Nama Use Case	Mengelola Perkembangan		
Tujuan	Kontraktor dapat menambah data perkembangan pada awal jam kerja atau akhir jam kerja.		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data perkembangan berhasil tampil, penambahan data berhasil		
Kondisi Akhir Gagal	Data catatan baru tidak dapat tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “ <i>Progres</i> ” pada menu proyek		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “ <i>Progress</i> ” pada menu proyek	Sistem menampilkan daftar data perkembangan yang telah tersimpan dalam <i>database</i> pada proses sebelumnya

	Langkah	Aktor	Sistem
	2	Kontraktor memilih icon plus untuk menambahkan data	Sistem menampilkan <i>form</i> yang harus diisi untuk menambahkan data
	3	Kontraktor menekan tombol simpan	Sistem melakukan penyimpanan data
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

10. *Flow of Events* Mengelola Permohonan Persetujuan

Flow of events mengelola permohonan persetujuan bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.25 *Flow of Events* Mengelola Permohonan Persetujuan (Mandor)

Nama Use Case	Mengelola Permohonan Persetujuan
Tujuan	Kontraktor menerima daftar permintaan pembelian oleh mandor yang ditugaskan terhadap satu proyek
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>
Kondisi Akhir Sukses	Data permohonan persetujuan berhasil tampil
Kondisi Akhir Gagal	Data permohonan pembelian tidak dapat tampil

Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Notifikasi” pada menu houses		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Notifikasi” pada menu Houses	Sistem menampilkan daftar data permohonan pembelian
Alur Perluasan	Aksi Pencegahan Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

11. *Flow of Events* Mengelola Persetujuan Pembelian

Flow of events mengelola persetujuan pembelian bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.26 *Flow of Events* Mengelola Persetujuan Pembelian (Mandor)

Nama Use Case	Mengelola Permohonan Pembelian
Tujuan	Kontraktor dapat menerima atau menolak permintaan yang telah dilakukan oleh mandor yang ditugaskan terhadap satu proyek

Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data permohonan pembelian berhasil tampil, dan pengubahan data status berhasil		
Kondisi Akhir Gagal	Perubahan data permohonan pembelian tidak dapat tersimpan		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih salah satu data pada menu “Notifikasi”		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih salah satu daftar permohonan	Sistem menampilkan <i>detail</i> data permohonan
	2	Kontraktor menekan tombol setuju atau tidak setuju	Sistem melakukan penyimpanan data
Alur Perluasan	Aksi Pencegahan		
	Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan		

12. *Flow of Events* Melihat Detail Laporan Pengeluaran

Flow of events melihat detail laporan pengeluaran bertujuan untuk dapat mengetahui alur logika dari pengguna kontraktor terhadap alur penggunaan sistem.

Tabel 3.27 *Flow of Events* Melihat Detail Laporan Pengeluaran (Mandor)

Nama Use Case	Melihat Detail Laporan Pengeluaran		
Tujuan	Kontraktor dapat melihat <i>detail</i> pengeluaran per minggu, per hari, hingga per transaksi yang telah dilakukan pada proyek yang telah berjalan		
Prasyarat	Kontraktor dalam keadaan telah <i>login</i>		
Kondisi Akhir Sukses	Data <i>detail</i> pengeluaran dapat dilihat dari rangkuman mingguan hingga ke tiap transaksi		
Kondisi Akhir Gagal	Data <i>detail</i> pengeluaran tidak tampil atau tidak sesuai		
Aktor Utama	Kontraktor		
Aktor Sekunder	Tidak ada		
Pemicu	Kontraktor memilih menu “Report” pada menu proyek		
Alur Utama	Langkah	Aktor	Sistem
	1	Kontraktor memilih menu “Report” pada menu proyek	Sistem menampilkan <i>detail</i> data pengeluaran mingguan
	2	Kontraktor memilih data mingguan	Sistem menampilkan <i>detail</i> data per hari pada minggu yang dipilih

	Langkah	Aktor	Sistem
	3	Kontraktor memilih data harian	Sistem menampilkan <i>detail</i> data per pengeluaran pada hari yang dipilih
	4	Kontraktor memilih data pengeluaran	Sistem menampilkan <i>detail</i> data pengeluaran yang dipilih
Aksi Pencegahan			
Jika data yang disimpan atau diubah tidak sesuai dengan format yang ditentukan maka data tidak akan tersimpan			

3.5 Sequence Diagram

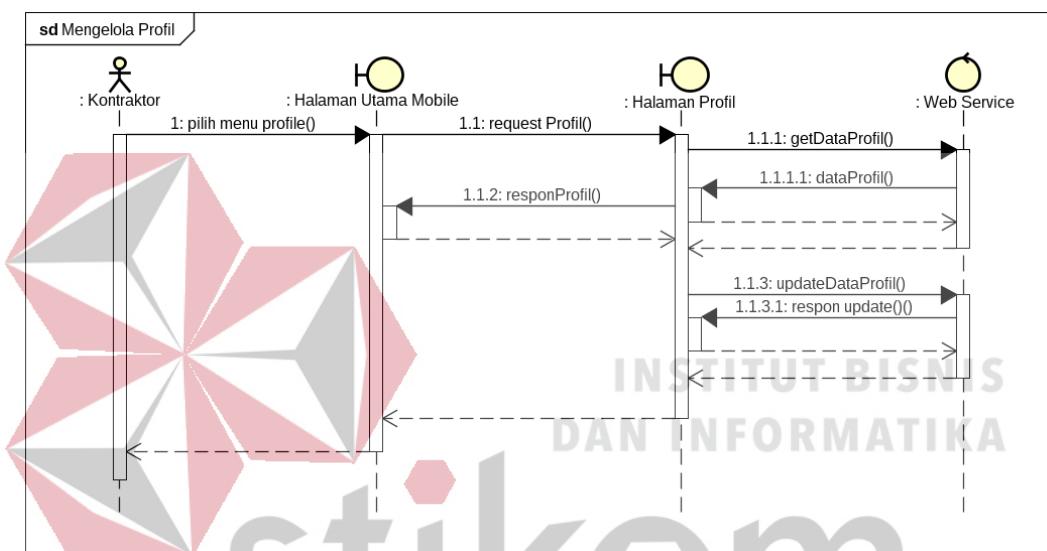
Pada *sequence diagram* akan ditunjukkan alur fungsionalitas pada *use case* yang disusun dalam urutan waktu. Diagram sekuensial menggambarkan interaksi yang dilakukan oleh tiap objek dalam sistem. Kerjasama antar objek tersebut dilakukan dengan cara saling mengirimkan pesan yang membentuk suatu alur kerja sama.

A. Sequence Diagram Kontraktor

Berikut merupakan *sequence diagrams* dari tipe pengguna kontraktor pada aplikasi pengawasan kinerja mandor:

1. Sequence Diagram Mengelola Profil

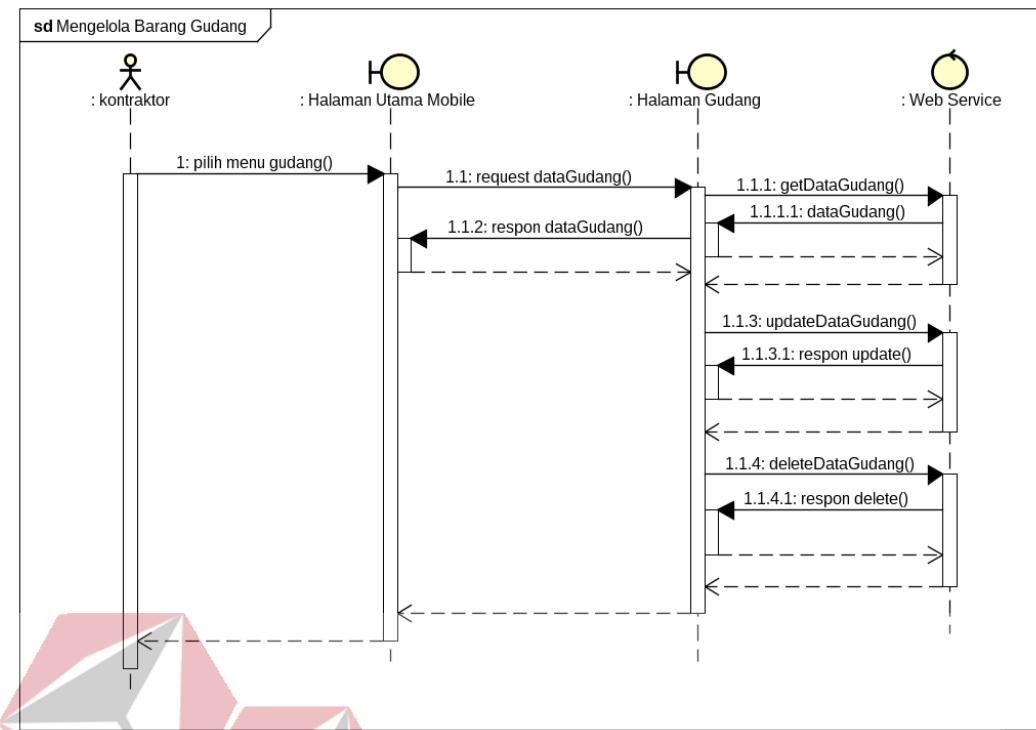
Sequence diagram mengelola profil menjelaskan tentang alur sistem saat mengelola profil pengguna.



Gambar 0.4 Sequence Diagram Mengelola Profil (Kontraktor)

2. Sequence Diagram Mengelola Barang Gudang

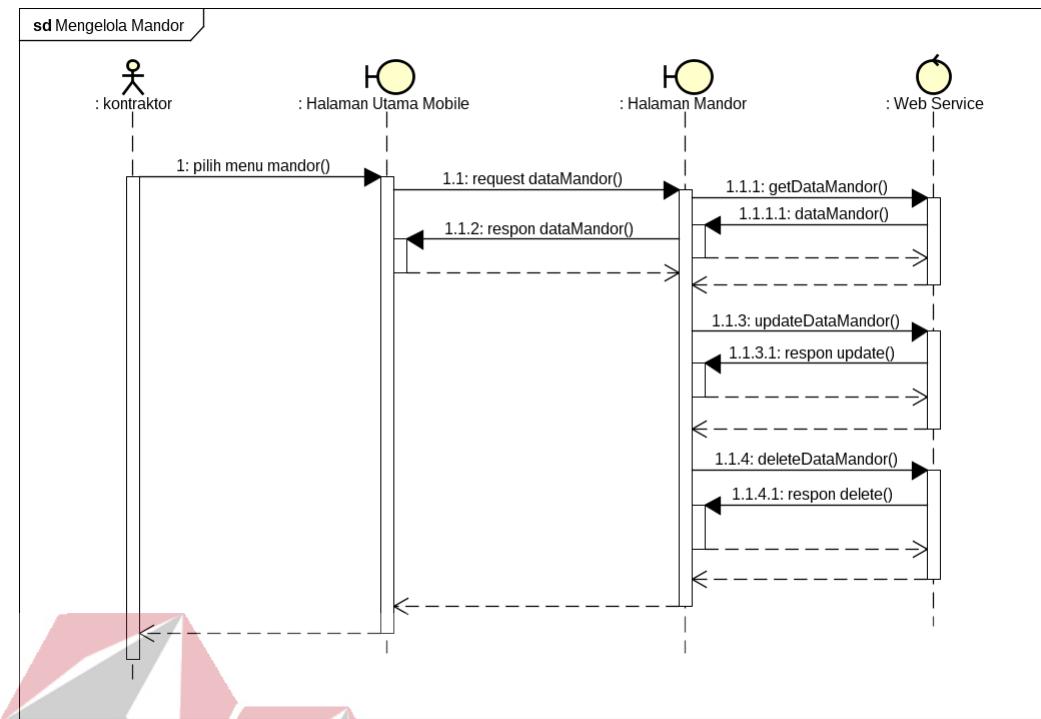
Sequence diagram mengelola barang gudang menjelaskan tentang alur sistem saat mengelola barang gudang. Barang yang terdapat di gudang dapat diambil oleh mandor. Barang yang terdapat di gudang tidak dapat diubah harganya. Jika terdapat perubahan harga, maka akan dibuat data baru. Kontraktor hanya dapat mengubah jumlah bahan baku yang terdapat pada gudang.



Gambar 0.5 Sequence Diagram Mengelola Barang Gudang (Kontraktor)

3. Sequence Diagram Mengelola Mandor

Sequence diagram mengelola mandor menjelaskan tentang alur sistem saat mengelola mandor. Kontraktor dapat menambahkan data dan mengubah data mandor yang ditugaskan, termasuk data *password* dan foto profil. Dibuat seperti ini karena terkadang mandor lupa akan *password* untuk *login*, dan kejadian pergantian mandor pada proyek juga menjadi salah satu sebab dibuatnya *system* tersebut.

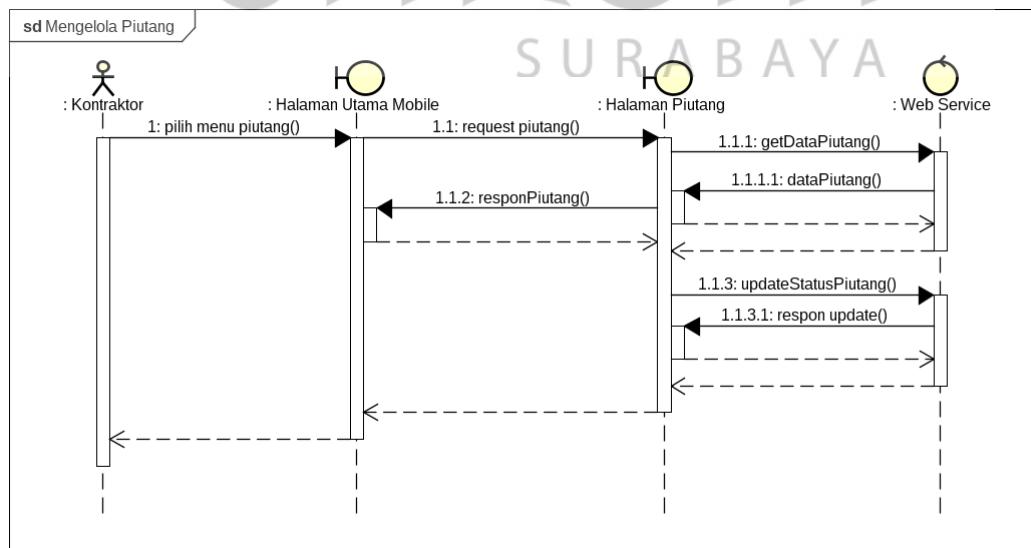


Gambar 0.6 Sequence Diagram Mengelola Mandor (Kontraktor)

4. Sequence Diagram Mengelola Piutang

INSTITUT BISNIS
AN INFORMATIKA

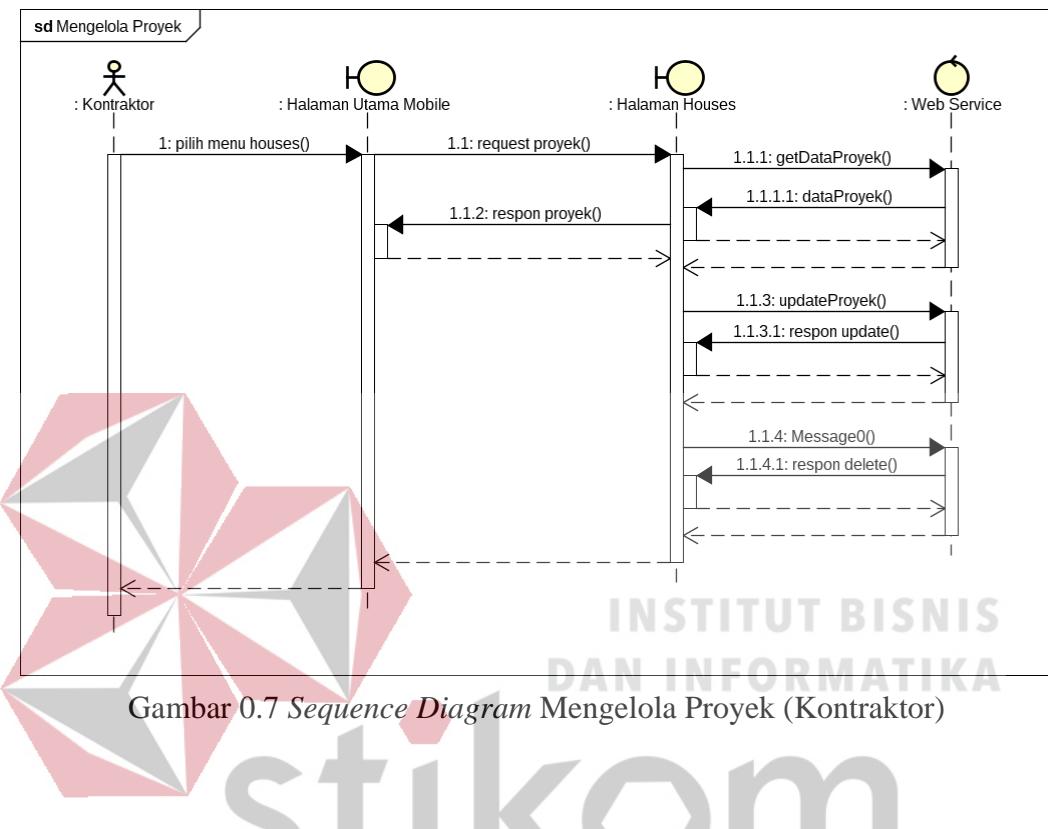
Sequence diagram mengelola piutang menjelaskan tentang alur sistem saat mengelola piutang.



Gambar 0.1 Sequence Diagram Mengelola Piutang (Kontraktor)

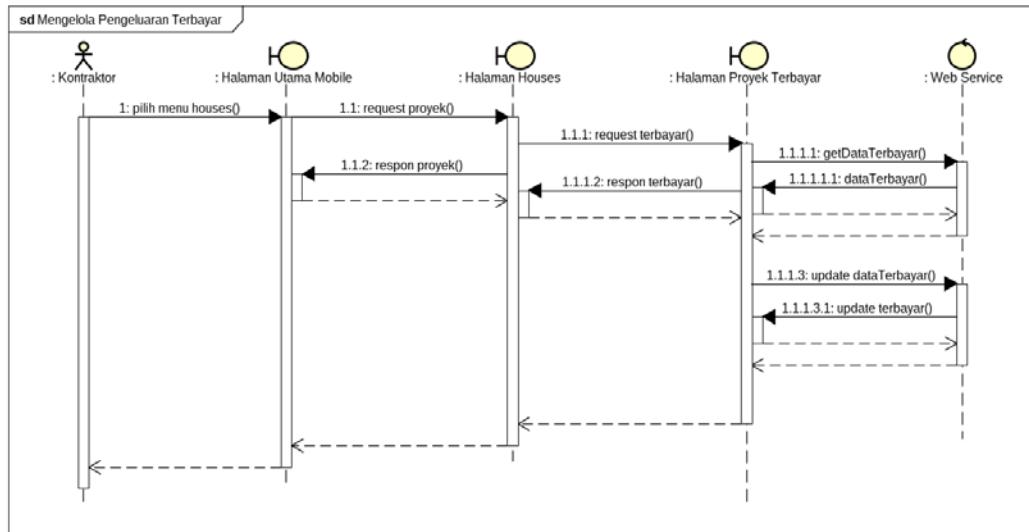
5. Sequence Diagram Mengelola Proyek

Sequence diagram mengelola proyek menjelaskan tentang alur sistem saat mengelola proyek.



6. Sequence Diagram Mengelola Pengeluaran Terbayar

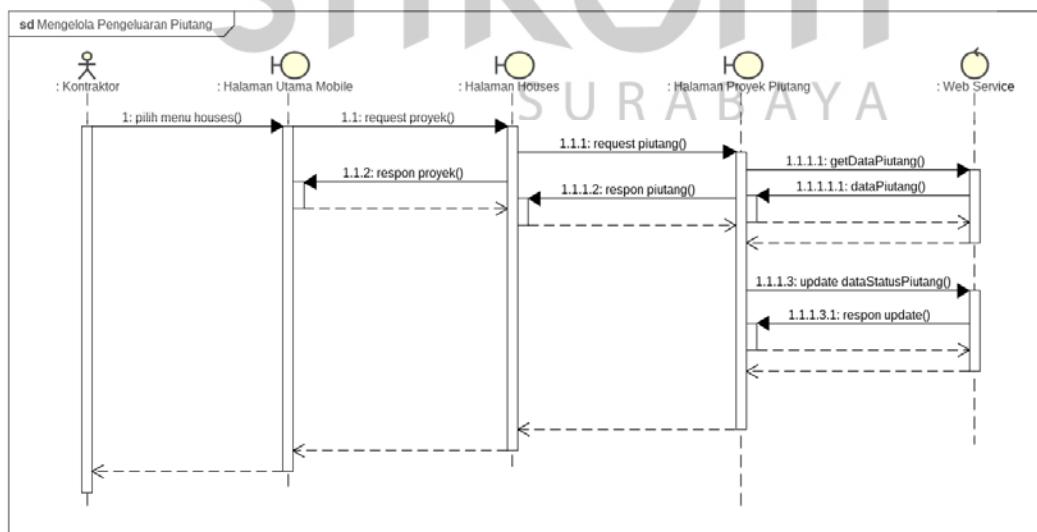
Sequence diagram mengelola pengeluaran terbayar menjelaskan tentang alur sistem saat mengelola pengeluaran terbayar. Pada menu ini akan ada dua tipe pengeluaran yaitu pengambilan barang dari gudang dan pembelian dengan pembayaran *cash*. Pada daftar pengeluarannya akan dibedakan pada tampilan aplikasi.



Gambar 0.8 Sequence Diagram Mengelola Pengeluaran Terbayar (Kontraktor)

7. Sequence Diagram Mengelola Pengeluaran Piutang

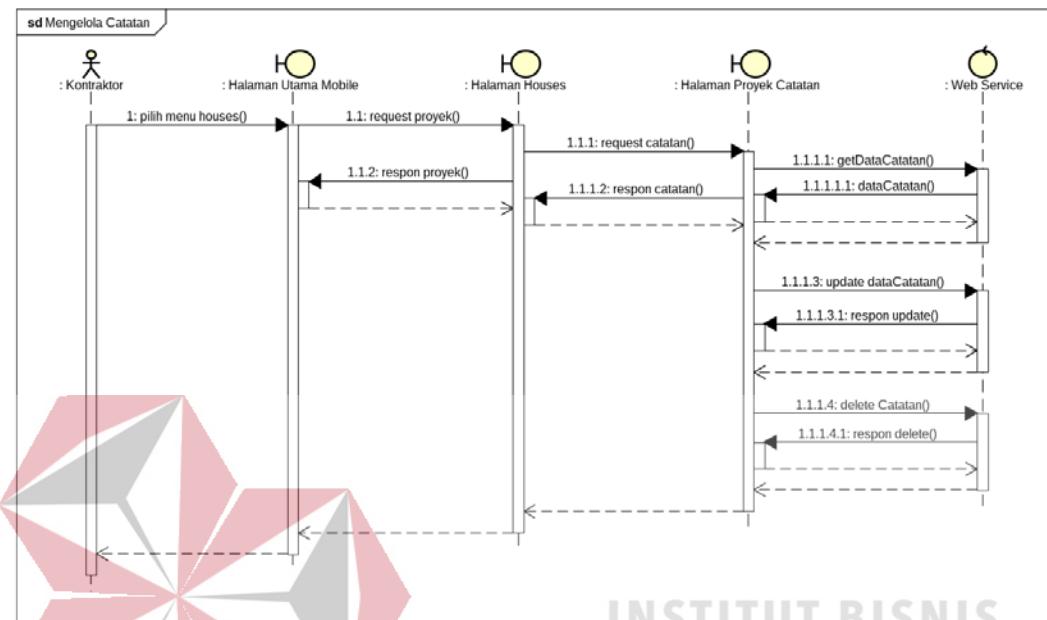
Sequence diagram mengelola pengeluaran piutang menjelaskan tentang alur sistem saat mengelola pengeluaran piutang. Pada menu ini hanya menangani untuk pembelian secara piutang. Untuk proses pencatatan pembayaran ada pada menu bayar piutang.



Gambar 0.9 Sequence Diagram Mengelola Pengeluaran Piutang (Kontraktor)

8. Sequence Diagram Mengelola Catatan

Sequence diagram mengelola catatan menjelaskan tentang alur sistem saat mengelola catatan.

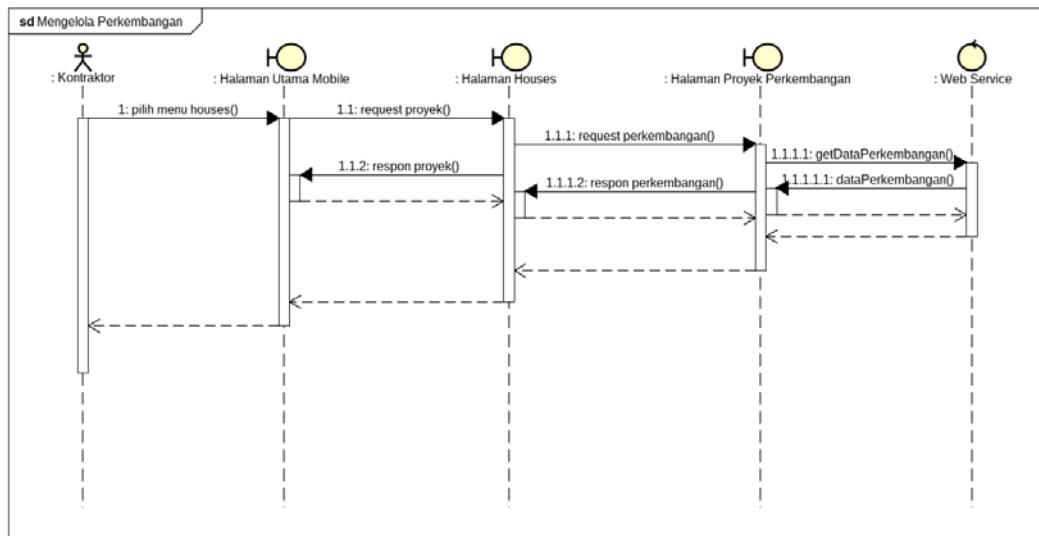


Gambar 0.10 Sequence Diagram Mengelola Catatan (Kontraktor)

9. Sequence Diagram Mengelola Perkembangan

Sequence diagram mengelola perkembangan menjelaskan tentang alur sistem saat mengelola perkembangan. Data dalam menu ini tidak dapat di ubah ataupun di hapus. Hal ini dimaksutkan agar data yang telah masuk dapat menjadi bukti kinerja mador dan kondisi asli di lapangan.

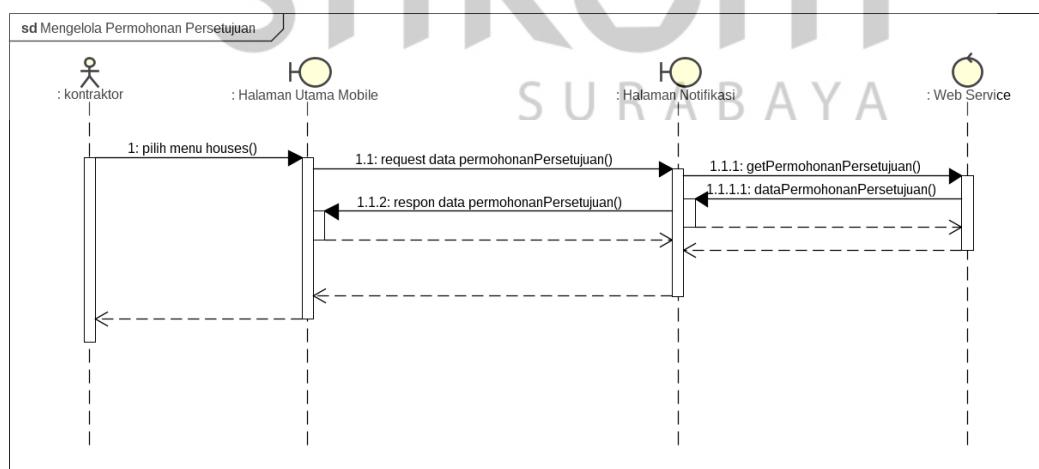
Pencatatan *progress* dapat dilakukan paling sedikit dua kali dalam sehari yaitu pada saat awal pengerjaan dan akhir pengerjaan. Meski seperti itu, pelaporan *progress* dapat dilakukan kapan saja.



Gambar 0.11 Sequence Diagram Mengelola Perkembangan (Kontraktor)

10. Sequence Diagram Mengelola Permohonan Persetujuan

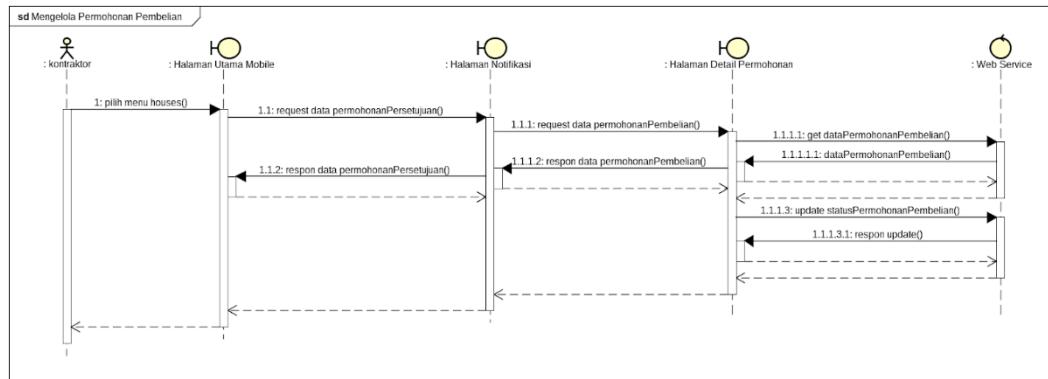
Sequence diagram mengelola permohonan persetujuan menjelaskan tentang alur sistem saat mengelola permohonan persetujuan. Permohonan dilakukan oleh mandor dan disetujui oleh kontraktor. Permohonan ini dibuat dalam menu notifikasi.



Gambar 0.12 Sequence Diagram Mengelola Permohonan Persetujuan

11. Sequence Diagram Mengelola Persetujuan Pembelian

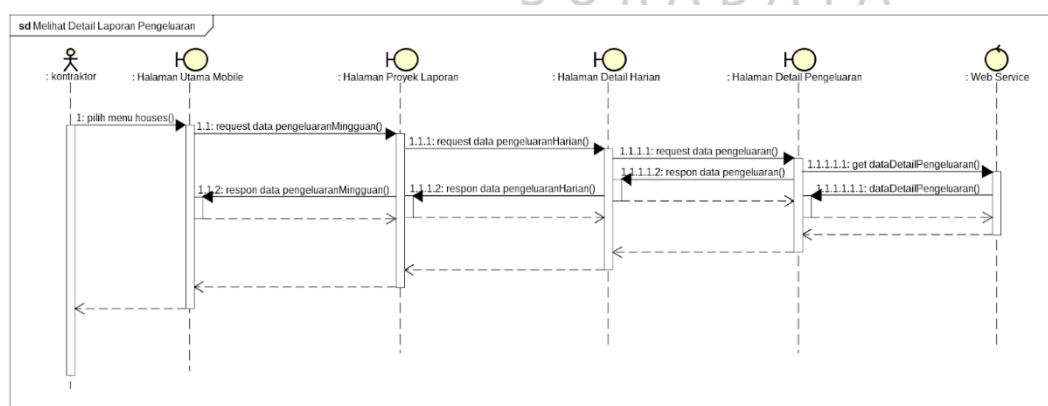
Sequence diagram mengelola persetujuan pembelian gudang menjelaskan tentang alur sistem saat mengelola persetujuan pembelian.



Gambar 0.13 *Sequence Diagram* Mengelola Persetujuan (Kontraktor)

12. Sequence Diagram Melihat Detail Laporan Pengeluaran

Sequence diagram melihat detail laporan pengeluaran gudang menjelaskan tentang alur sistem saat melihat detail laporan pengeluaran. Laporan yang dihasilkan ada laporan *detail* pengeluaran mingguan dan pengeluaran harian pada setiap proyek yang berjalan, sehingga pencatatan dapat dipantau dan koreksi oleh kontraktor.



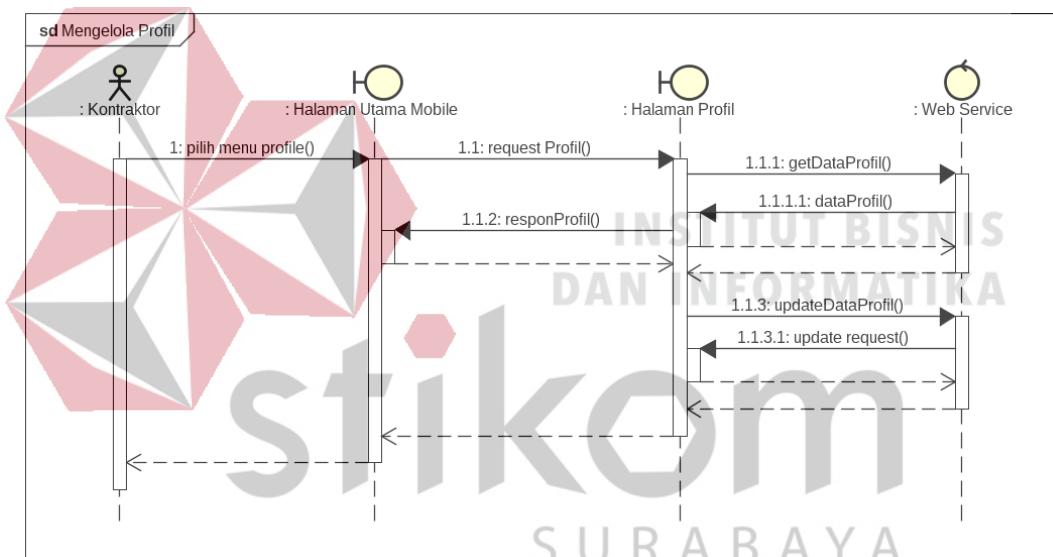
Gambar 0.14 *Sequence Diagram* Detail Laporan Pengeluaran (Kontraktor)

B. Sequence Diagram Mandor

Berikut merupakan *sequence diagrams* dari tipe pengguna mandor pada aplikasi pengawasan kinerja mandor:

1. Sequence Diagram Mengelola Profil

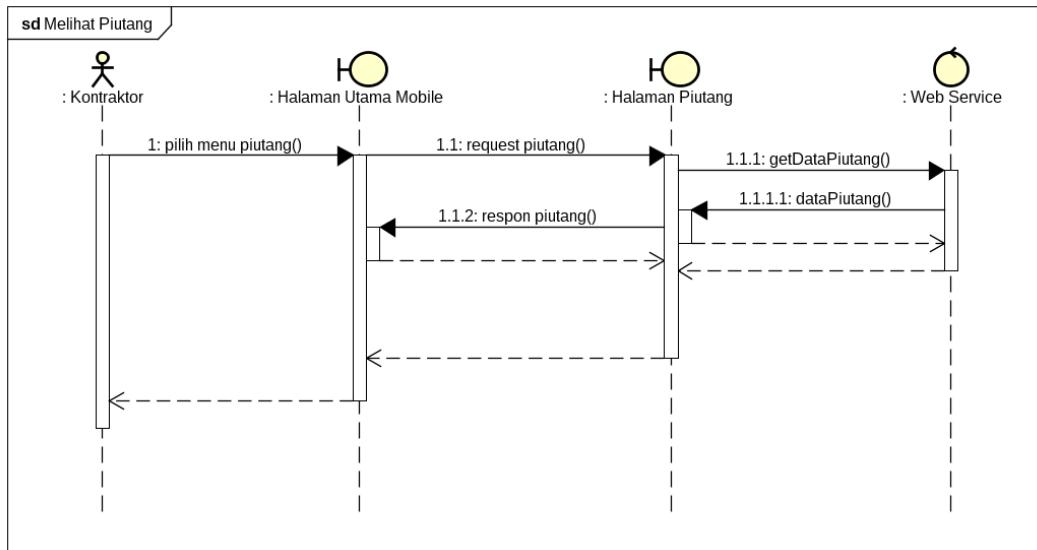
Sequence diagram mengelola profil menjelaskan tentang alur sistem saat mengelola profil. Mandor dapat mengubah data pada profil mereka masing-masing, tetapi tidak dapat melakukan penambahan ataupun penghapusan data.



Gambar 0.15 Sequence Diagram Mengelola Profil (Mandor)

2. Sequence Diagram Melihat Piutang

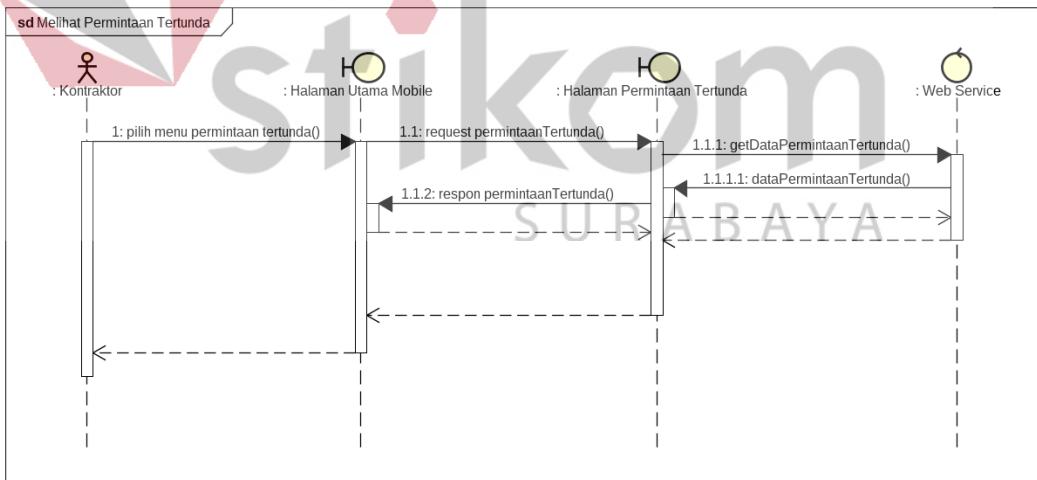
Sequence diagram melihat piutang menjelaskan tentang alur sistem saat melihat piutang. Menu ini terdapat pada tipe pengguna mandor karena diharapkan mandor dapat membantu mengingatkan tentang tagihan pembelian yang harus dilakukan pada hari itu.



Gambar 0.16 Sequence Diagram Melihat Piutang (Mandor)

3. Sequence Diagram Melihat Permintaan Tertunda

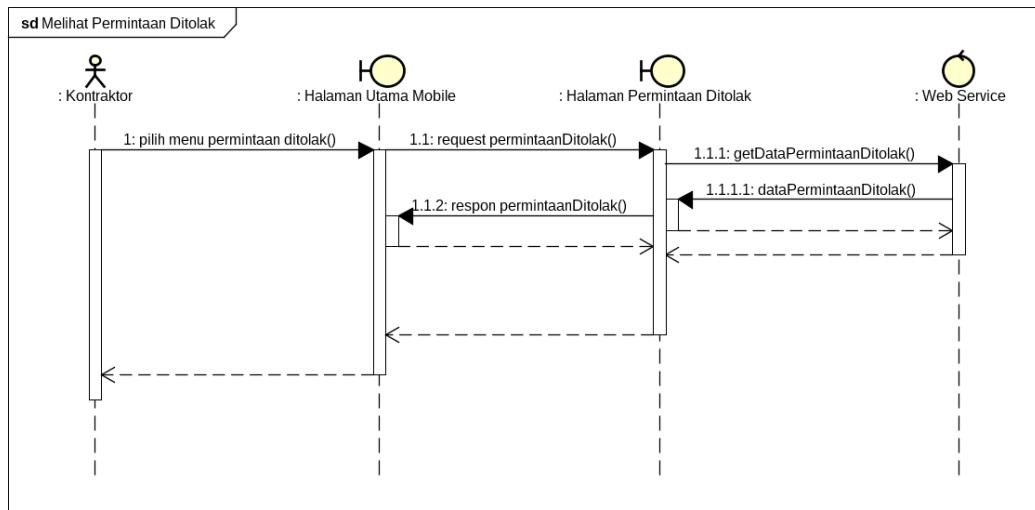
Sequence diagram melihat permintaan tertunda gudang menjelaskan tentang alur sistem saat melihat permintaan tertunda.



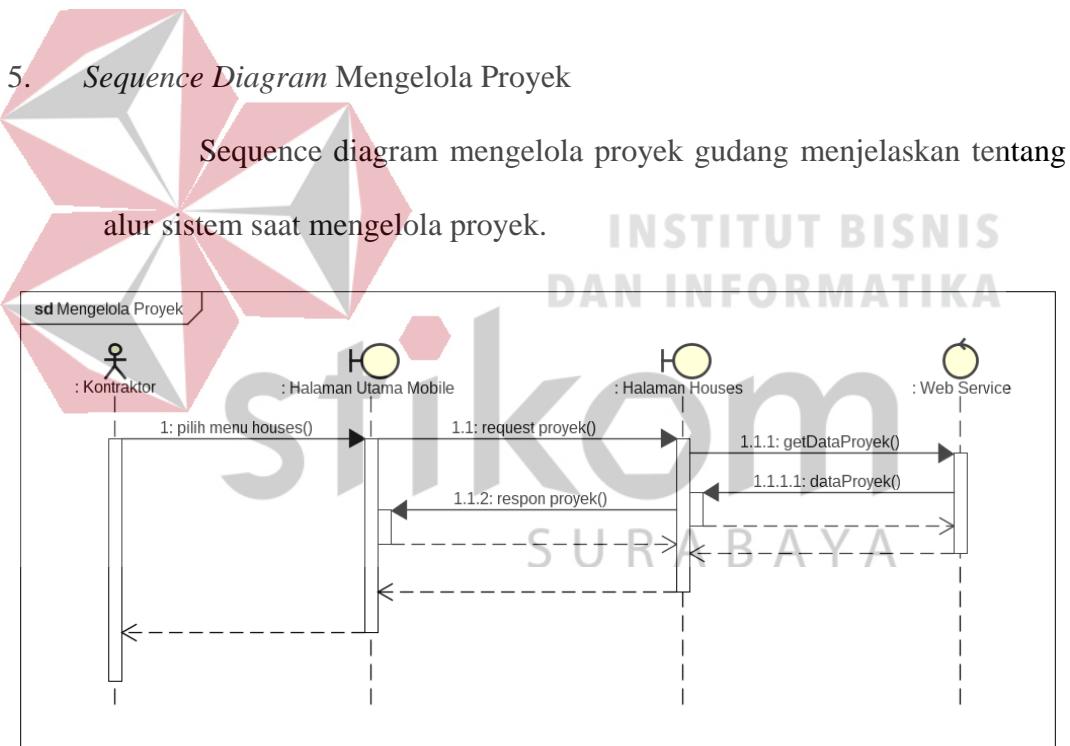
Gambar 0.17 Sequence Diagram Melihat Permintaan Tertunda (Mandor)

4. Sequence Diagram Melihat Permintaan Ditolak

Sequence diagram melihat permintaan ditolak gudang menjelaskan tentang alur sistem saat melihat permintaan ditolak.



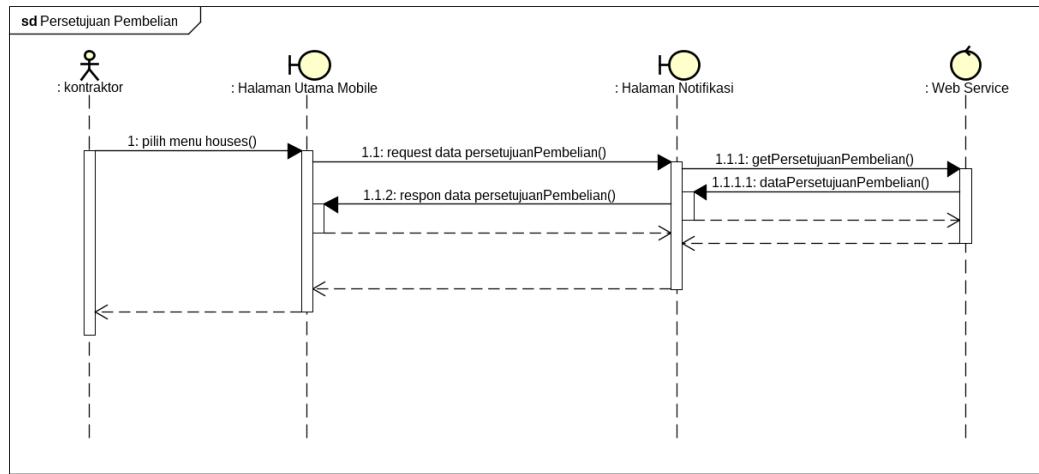
Gambar 0.18 Sequence Diagram Melihat Permintaan Ditolak (Mandor)



Gambar 0.19 Sequence Diagram Mengelola Proyek (Mandor)

6. Sequence Diagram Persetujuan Pembelian

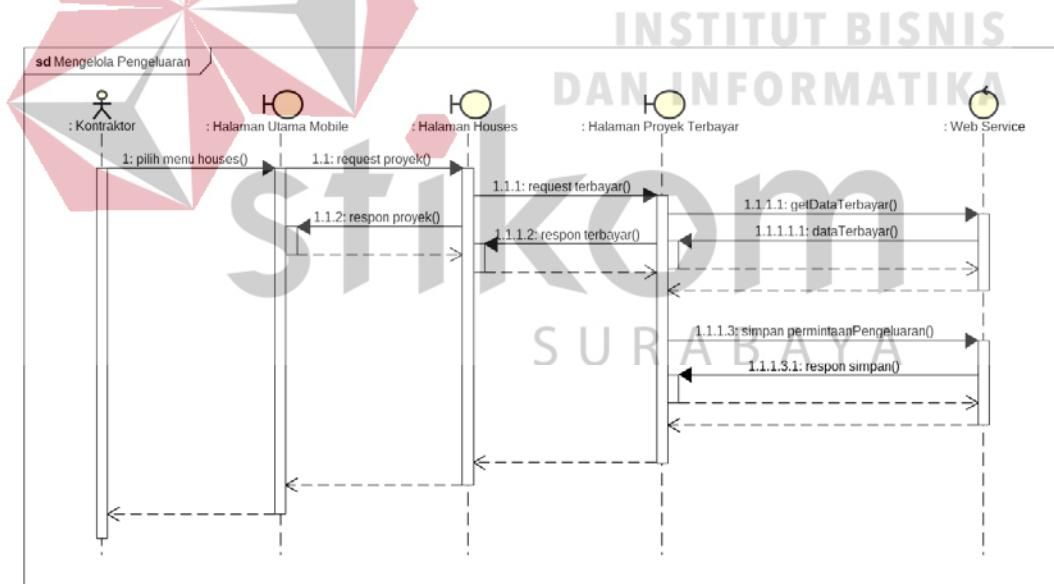
Sequence diagram persetujuan pembelian gudang menjelaskan tentang alur sistem saat persetujuan pembelian.



Gambar 0.20 *Sequence Diagram* Persetujuan Pembelian (Mandor)

7. *Sequence Diagram* Mengelola Pengeluaran

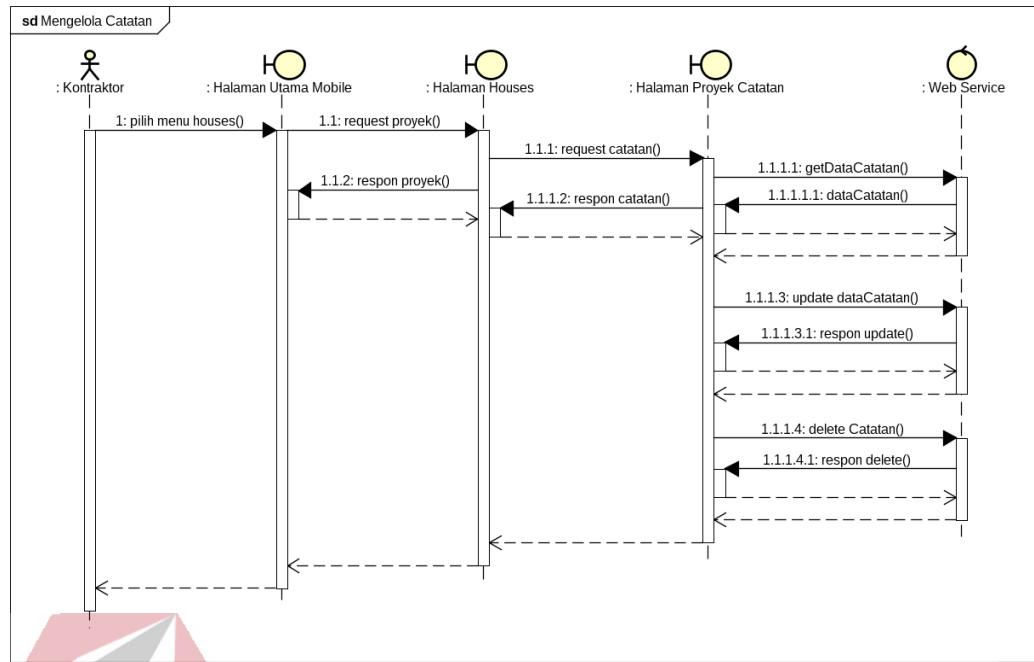
Sequence diagram mengelola pengeluaran gudang menjelaskan tentang alur sistem saat mengelola pengeluaran.



Gambar 0.21 *Sequence Diagram* Mengelola Pengeluaran (Mandor)

8. *Sequence Diagram* Mengelola Catatan

Sequence diagram mengelola catatan gudang menjelaskan tentang alur sistem saat mengelola catatan.

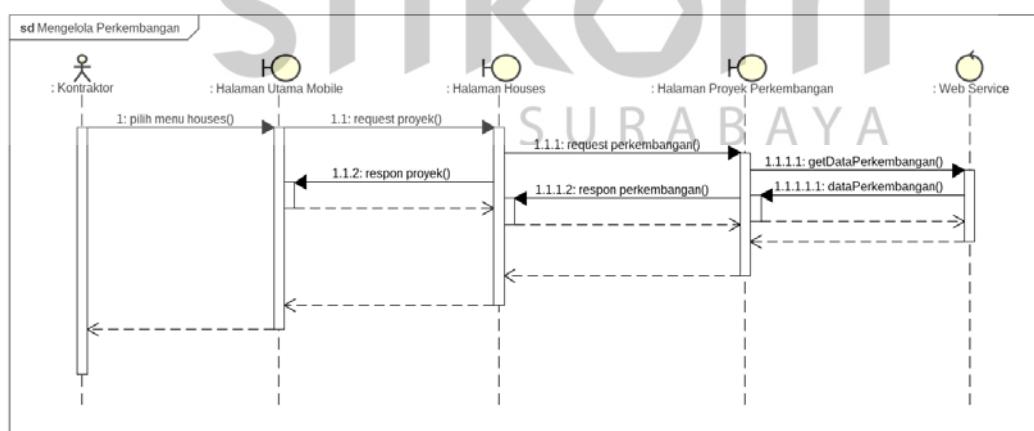


Gambar 0.22 Sequence Diagram Mengelola Catatan (Mandor)

9. Sequence Diagram Mengelola Perkembangan

Sequence diagram mengelola perkembangan gudang menjelaskan

tentang alur sistem saat mengelola perkembangan.



Gambar 0.23 Sequence Diagram Mengelola Perkembangan (Mandor)

3.6 Class Diagram

Pada *class diagram* akan ditunjukkan interaksi antara kelas pada sistem yang berhubungan langsung dengan *database*.

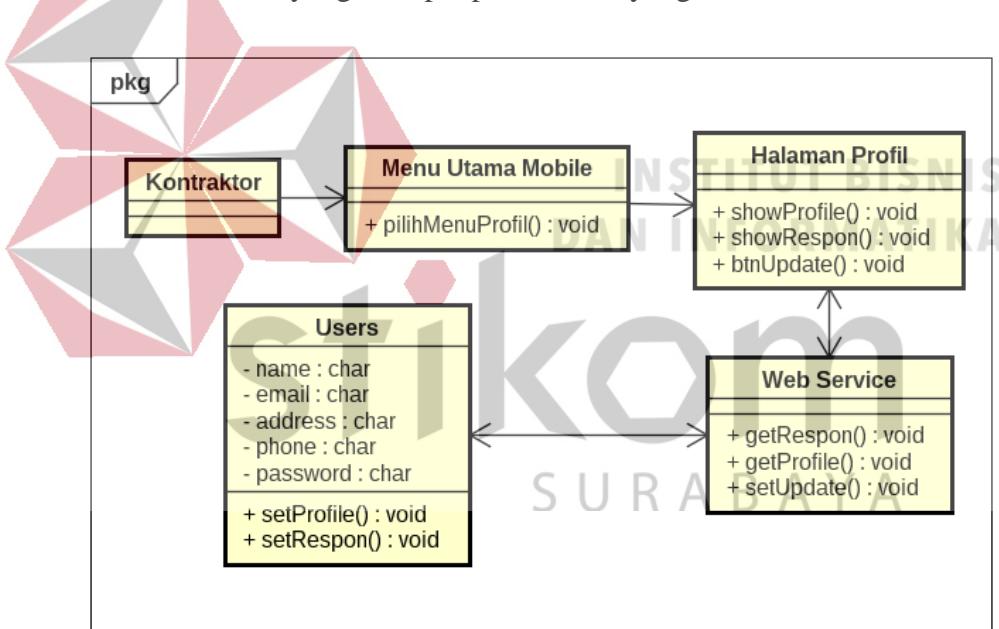
A. Class Diagram Kontraktor

Berikut merupakan *class diagram* dari tipe pengguna kontraktor pada aplikasi pengawasan kinerja mandor:

1. Class Diagram Mengelola Profil

Class diagram mengelola profil ini akan menunjukkan interaksi

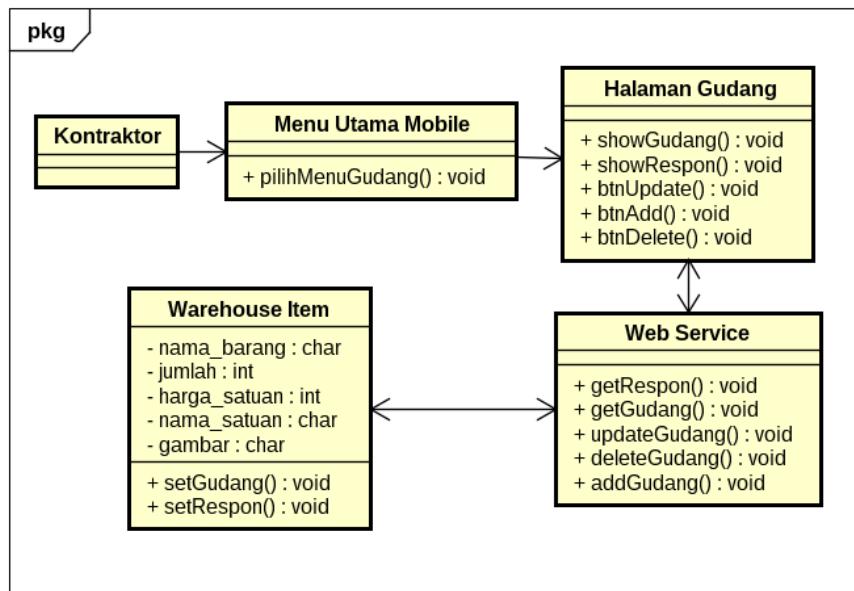
antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.24 *Class Diagram* Mengelola Profil (Kontraktor)

2. Class Diagram Mengelola Barang Gudang

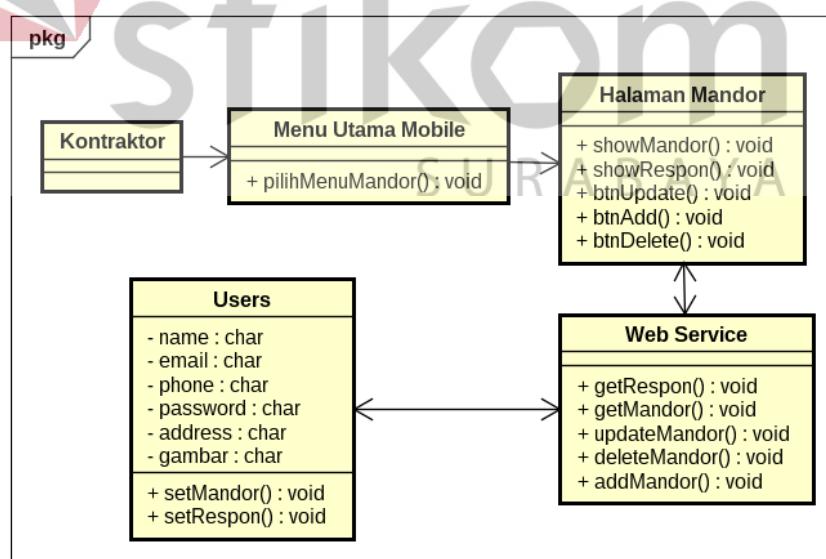
Class diagram mengelola barang gudang ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.25 *Class Diagram* Mengelola Barang Gudang (Kontraktor)

3. *Class Diagram* Mengelola Mandor

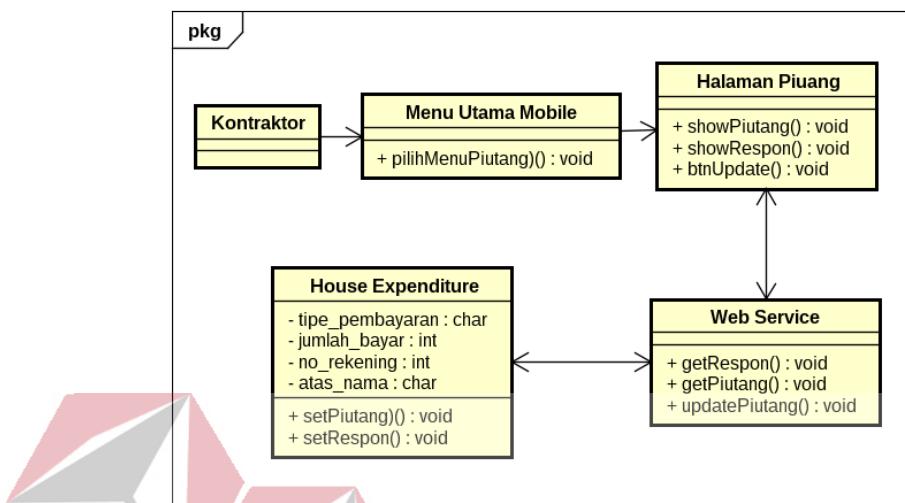
Class diagram mengelola mandor ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.26 *Class Diagram* Mengelola Mandor (Kontraktor)

4. Class Diagram Mengelola Piutang

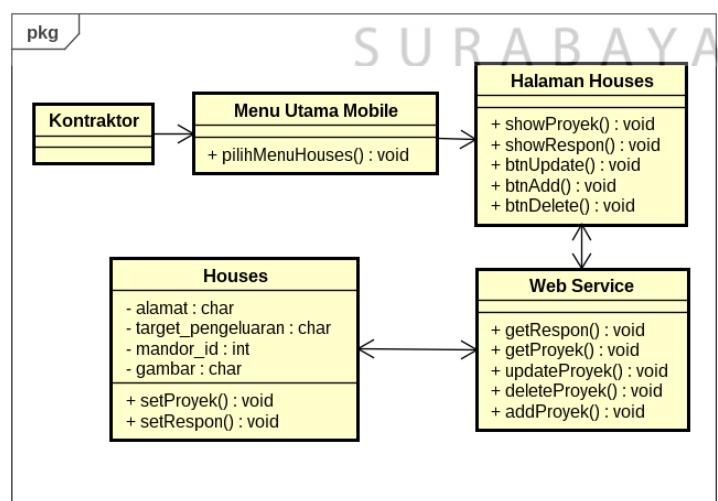
Class diagram mengelola piutang ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.27 *Class Diagram* Mengelola Piutang (Kontraktor)

5. Class Diagram Mengelola Proyek

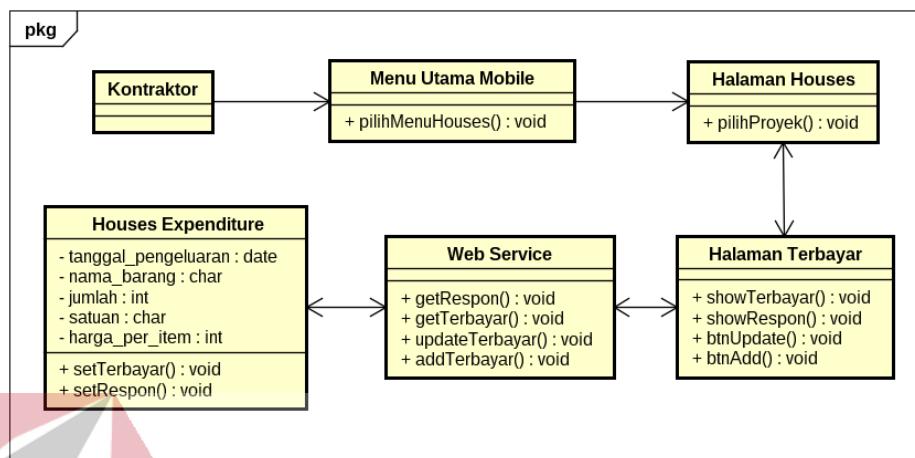
Class diagram ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.28 *Class Diagram* Mengelola Proyek (Kontraktor)

6. Class Diagram Mengelola Pengeluaran Terbayar

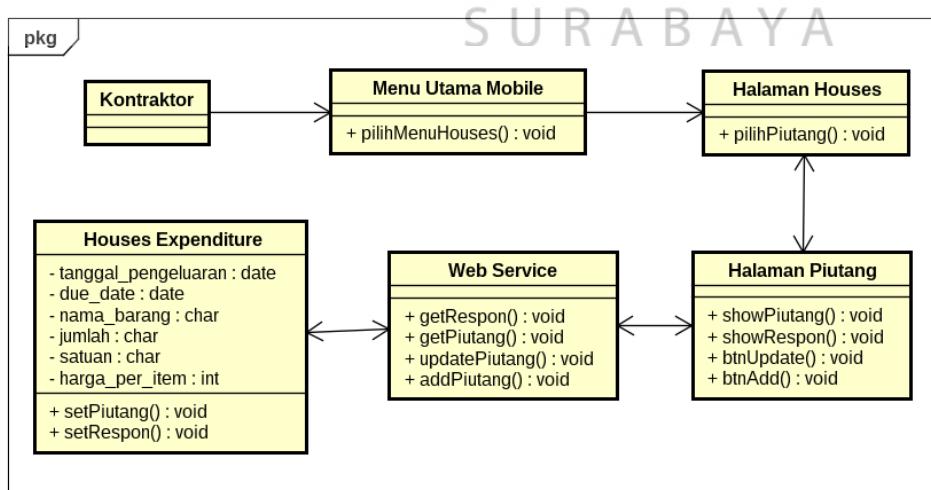
Class diagram ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.29 *Class Diagram* (Kontraktor)

7. Class Diagram Mengelola Pengeluaran Piutang

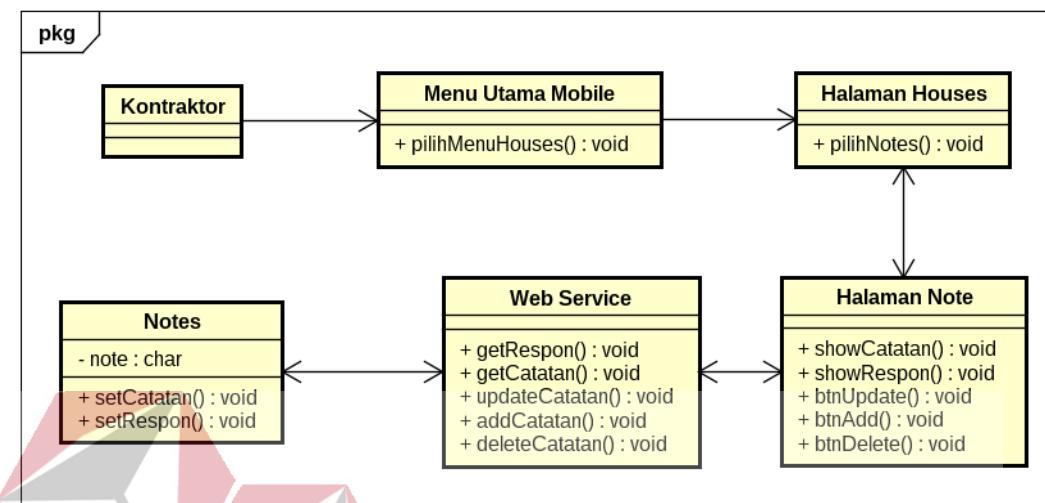
Class diagram mengelola pengeluaran piutang ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.30 *Class Diagram* Mengelola Pengeluaran Piutang (Kontraktor)

8. Class Diagram Mengelola Catatan

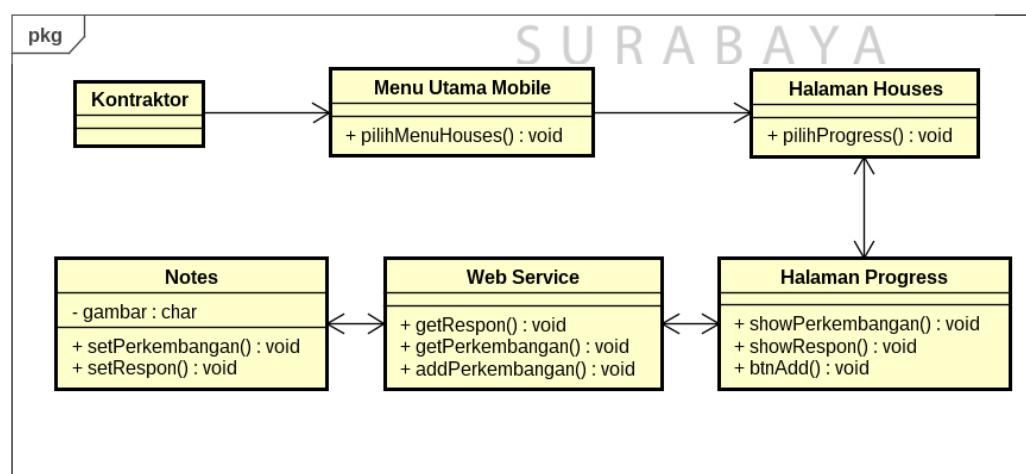
Class diagram mengelola catatan ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.31 *Class Diagram* Mengelola Catatan (Kontraktor)

9. Class Diagram Mengelola Perkembangan

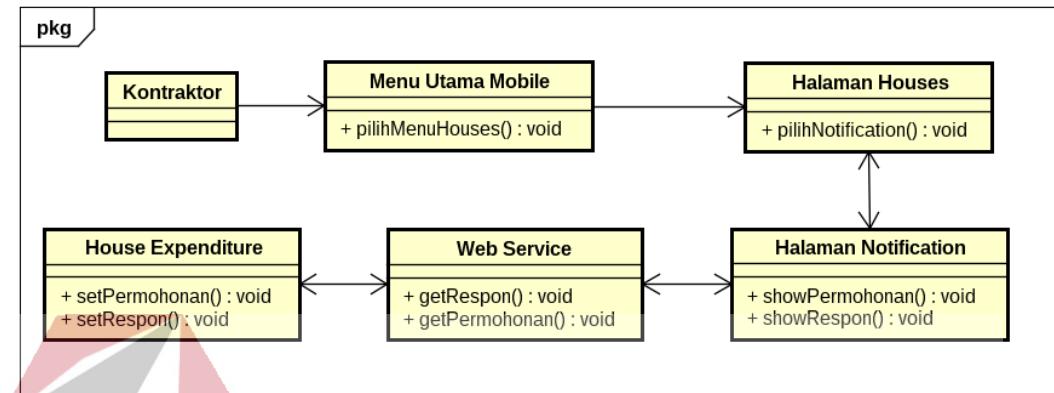
Class diagram mengelola perkembangan ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.32 *Class Diagram* Mengelola Perkembangan (Kontraktor)

10. Class Diagram Mengelola Permohonan Persetujuan

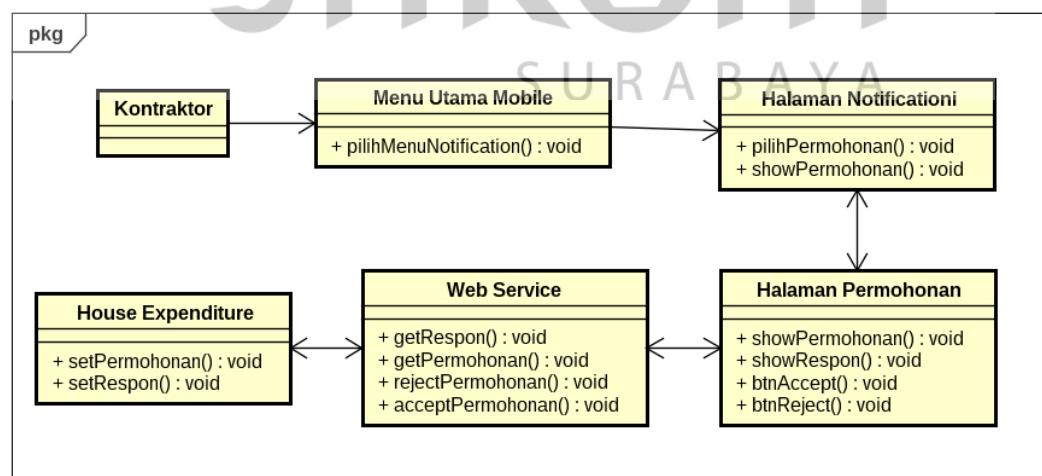
Class diagram mengelola permohonan persetujuan ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.33 *Class Diagram* Mengelola Permohonan Persetujuan (Kontraktor)

11. Class Diagram Mengelola Persetujuan Pembelian

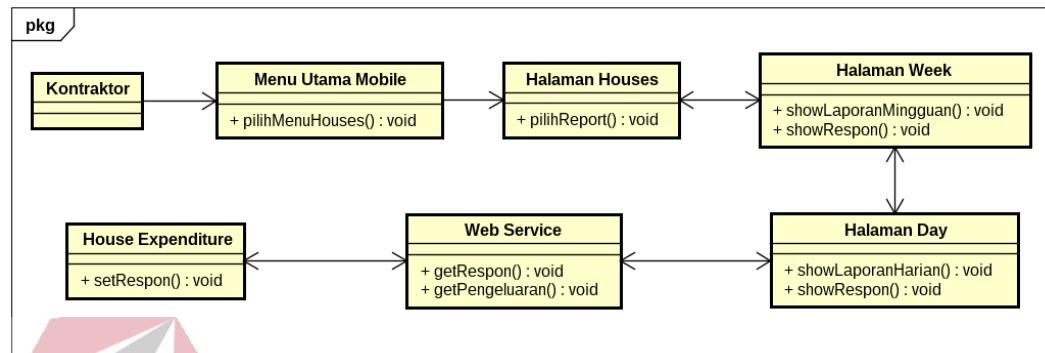
Class diagram mengelola pembelian ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.34 *Class Diagram* Mengelola Persetujuan Pembelian (Kontraktor)

12. Class Diagram Melihat Detail Laporan Pengeluaran

Class diagram melihat *detail* laporan pengeluaran ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



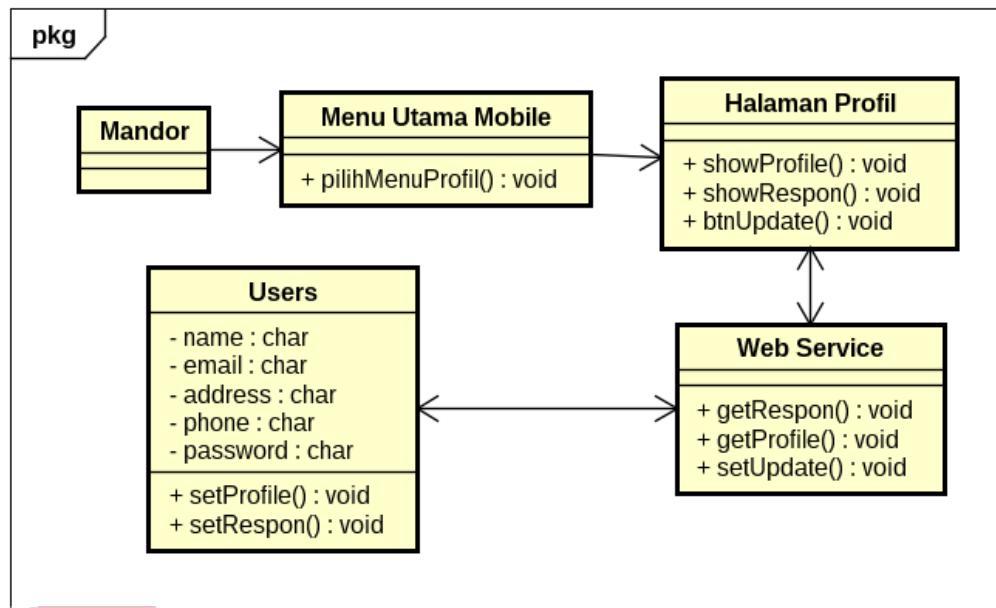
Gambar 0.35 *Class Diagram* Melihat *Detail* Laporan Pengeluaran (Kontraktor)

B. *Class Diagram* Mandor

Berikut merupakan *class diagram* dari tipe pengguna mandor pada aplikasi pengawasan kinerja mandor yang dibuat berdasarkan data kebutuhan yang telah didapatkan pada proses sebelumnya:

1. *Class Diagram* Mengelola Profil

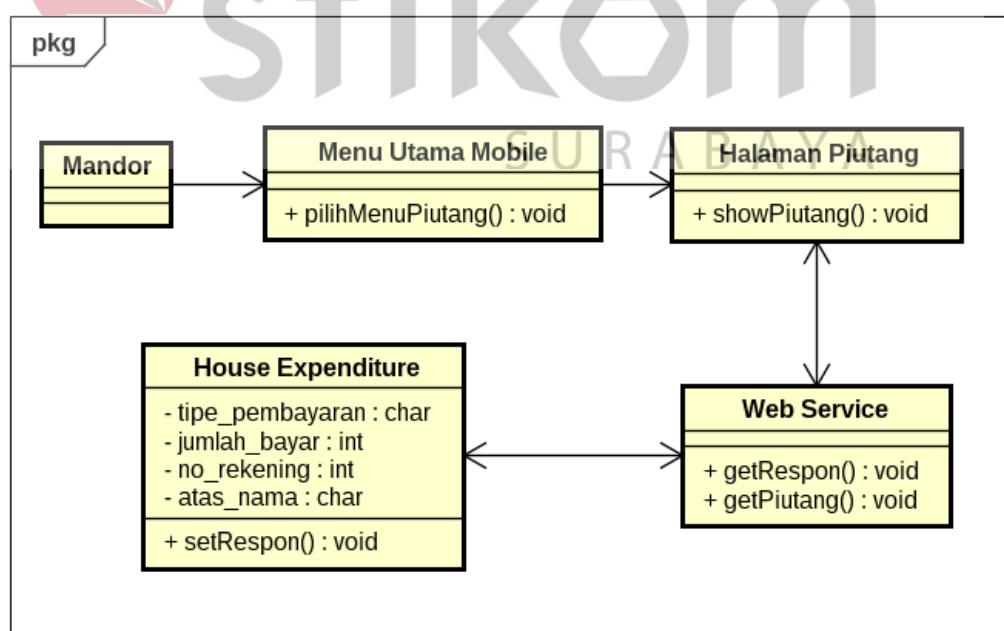
Class diagram mengelola profil ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat pada aplikasi pengawasan kinerja mandor.



Gambar 0.36 *Class Diagram* Mengelola Profil (Mandor)

2. *Class Diagram* Melihat Piutang

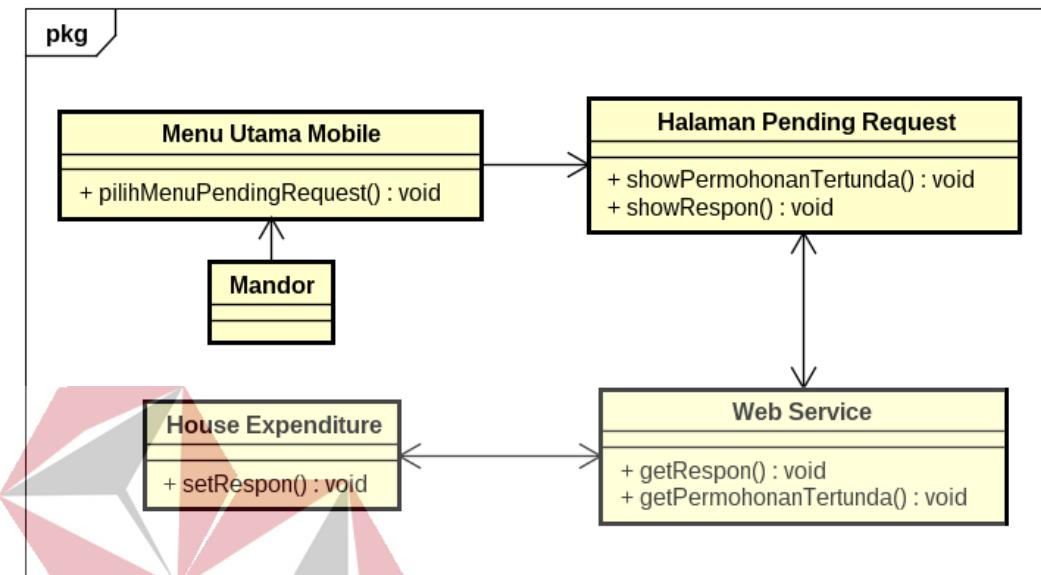
Class diagram melihat piutang ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.37 *Class Diagram* Melihat Piutang (Mandor)

3. Class Diagram Melihat Permintaan Tertunda

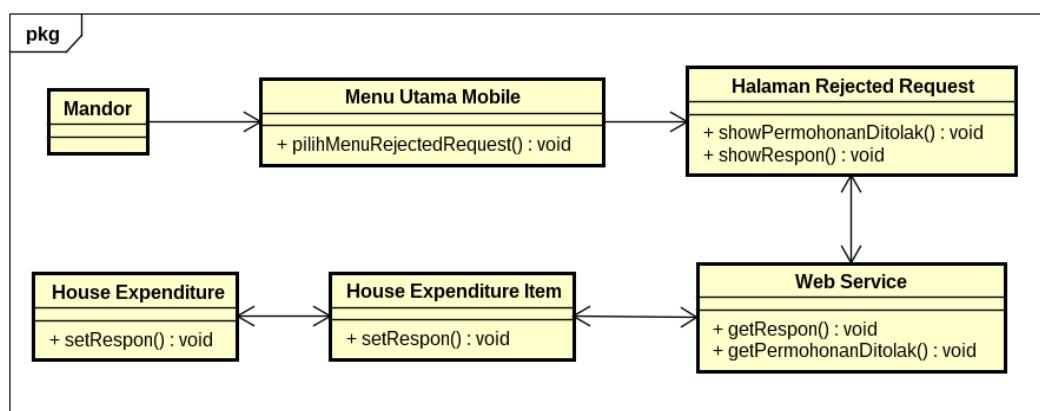
Class diagram melihat permintaan tertunda ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.38 *Class Diagram* Melihat Permintaan Tertunda (Mandor)

4. Class Diagram Melihat Permintaan Ditolak

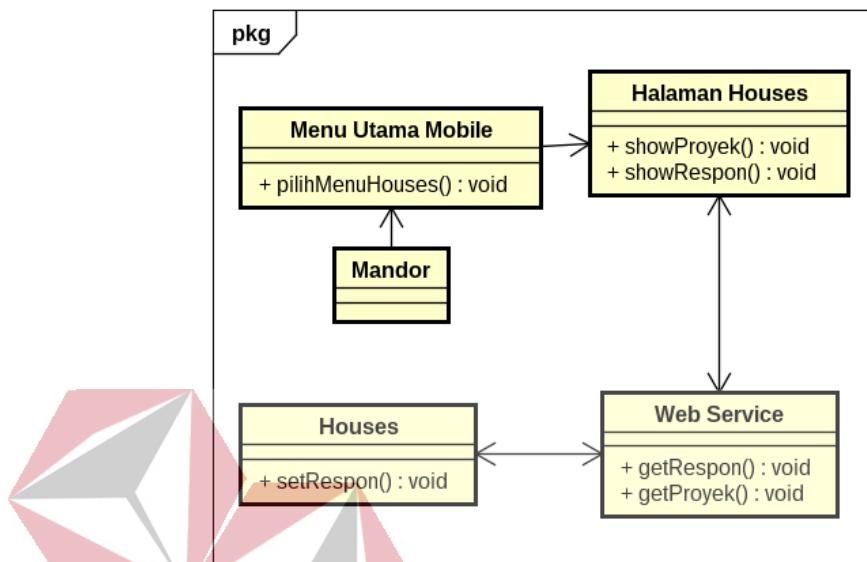
Class diagram meihat permintaan ditolak ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.39 *Class Diagram* Permintaan Ditolak (Mandor)

5. Class Diagram Mengelola Proyek

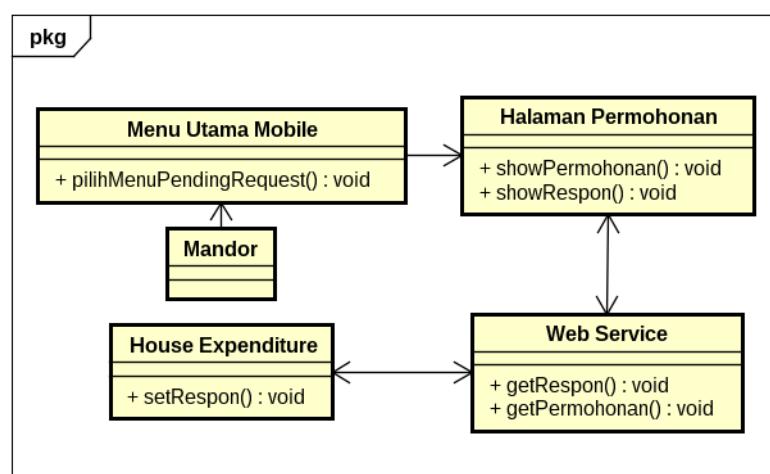
Class diagram mengelola proyek ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.40 *Class Diagram* Mengelola Proyek (Mandor)

6. Class Diagram Persetujuan Pembelian

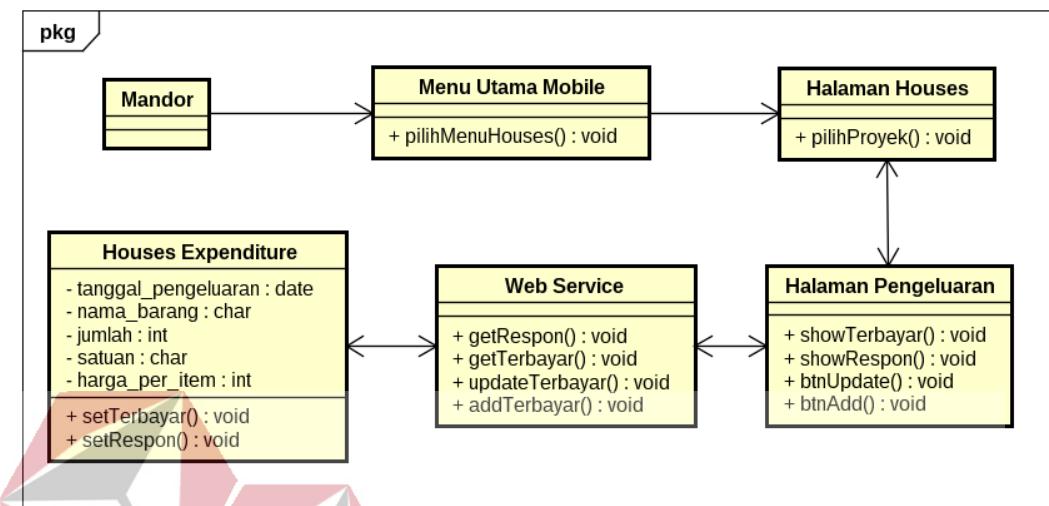
Class diagram mengelola pembelian ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.41 *Class Diagram* Persetujuan Pembelian (Mandor)

7. Class Diagram Mengelola Pengeluaran

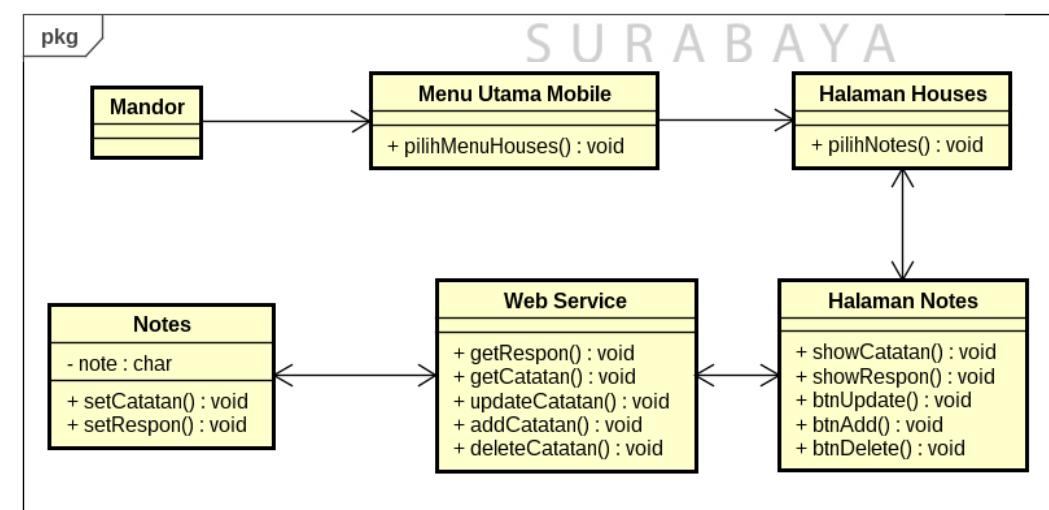
Class diagram mengelola pengeluaran ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.42 Class Diagram Mengelola Pengeluaran (Mandor)

8. Class Diagram Mengelola Catatan

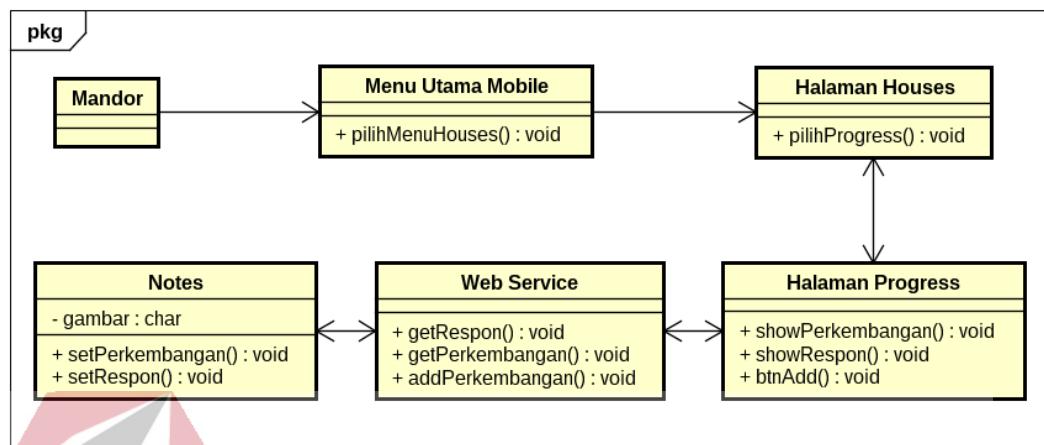
Class diagram mengelola catatan ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



Gambar 0.43 Class Diagram Mengelola Catatan (Mandor)

9. Class Diagram Mengelola Perkembangan

Class diagram mengelola perkembangan ini akan menunjukkan interaksi antar kelas yang terdapat pada sistem yang akan dibuat.



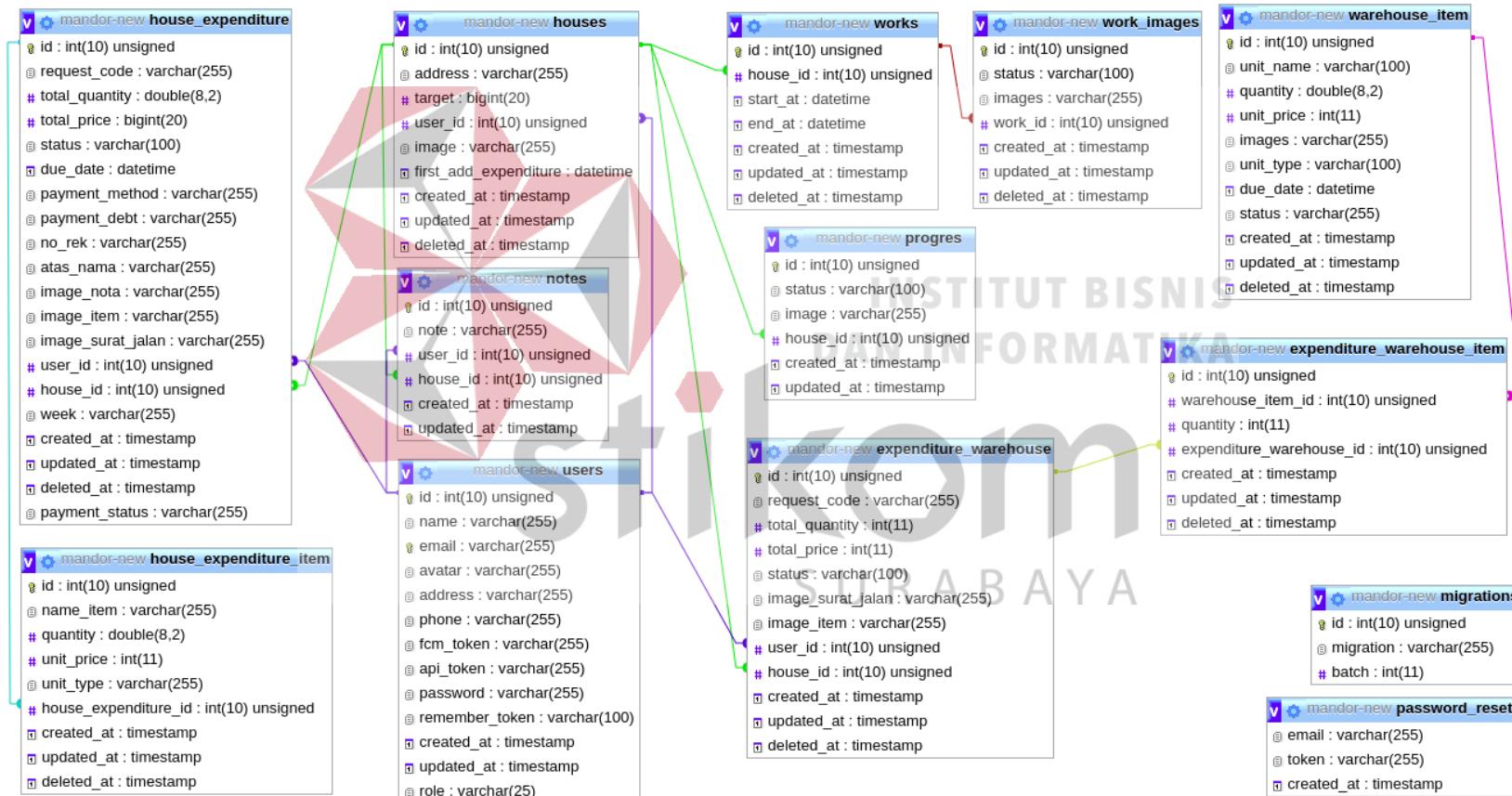
Gambar 0.44 *Class Diagram* Mengelola Perkembangan (Mandor)

3.7 Design

Pada tahap perancangan aplikasi ini akan dilakukan kegiatan untuk membuat desain aplikasi yang akan dibangun. Perancangan aplikasi ini akan menghasilkan *Physical Data Model* (PDM), Struktur dan Fungsi Tabel, dan desain *Input / Output* (*User Interface*). Database akan dibuat berdasarkan data yang telah didapatkan dari proses sebelumnya.

Kebutuhan data yang terkumpul inilah yang selanjutnya akan digunakan sebagai acuan pembuatan *database* aplikasi pengawasan kinerja mandor pada PT MKDS.

3.8 Physical Data Model (PDM)



Gambar 0.45 Physical Data Model

3.9 Struktur dan Fungsi Tabel

Berdasarkan desain *database* di atas dapat dijabarkan struktur dan fungsi dari tiap tabel adalah sebagai berikut:

1. Tabel *Users*
 - a. Nama Tabel : users
 - b. Primary Key : id
 - c. Foreign Key : email
 - d. Fungsi : Menyimpan data pengguna

Tabel 3.28 *Users*

No	Nama Field	Tipe Data	Length	Constraint
1	Id	Int	10	Primary Key
2	Name	Varchar	255	
3	Email	Varchar	255	Foreign Key
4	Avatar	Varchar	255	
5	Address	Varchar	255	
6	Phone	Varchar	255	
7	fcm_token	Varchar	255	
8	api_token	Varchar	255	
9	password	Varchar	255	
10	remember_token	Varchar	100	
11	created_at	timestamp		

No	Nama Field	Tipe Data	Length	Constraint
12	updated_at	timestamp		
13	Role	Varchar	25	

2. Tabel *Works*

- a. Nama Tabel : works
- b. Primary Key : id
- c. Foreign Key : house_id
- d. Fungsi : Menyimpan data pekerjaan

Tabel 3.29 Tabel *Works*

No	Nama Field	Tipe Data	Length	Constraint
1	Id	Int	10	Primary Key
2	house_id	Int	10	Foreign Key
3	start_id	Datetime		
4	end_id	Datetime		
5	created_at	Timestamp		
6	updated_at	Timestamp		
7	deleted_at	Timestamp		

3. Tabel *Works Images*

- a. Nama Tabel : works_images
- b. Primary Key : id

- c. *Foreign Key* : work_id
- d. Fungsi : Menyimpan data gambar pekerjaan

Tabel 3.30 Tabel *Works Images*

No	Nama Field	Tipe Data	Length	Constraint
1	id	Int	10	<i>Primary Key</i>
2	status	Varchat	100	
3	images	Varchat	255	
4	work_id	Int	10	<i>Foreign Key</i>
5	created_at	timestamp		
6	updated_at	timestamp		
7	deleted_at	timestamp		

4. Tabel *Houses*

- a. Nama Tabel : houses
- b. *Primary Key* : id
- c. *Foreign Key* : user_id
- d. Fungsi : Menyimpan data proyek

Tabel 3.31 Tabel *Houses*

No	Nama Field	Tipe Data	Length	Constraint
1	Id	Int	10	<i>Primary Key</i>

No	Nama Field	Tipe Data	Length	Constraint
2	Address	Varchar	255	
3	Target	Bigint	20	
4	user_id	Int	10	Foreign Key
5	Image	Varchar	255	
6	first_add_expenditure	Datetime		
7	created_at	Timestamp		
8	updated_at	Timestamp		
9	deleted_at	Timestamp		

5. Tabel *Houses Expenditure*

- a. Nama Tabel : houses_Expenditure
- b. Primary Key : id
- c. Foreign Key : house_id, user_id
- d. Fungsi : Menyimpan data pengeluaran proyek

Tabel 3.32 *House Expenditure*

No	Nama Field	Tipe Data	Length	Constraint
1	id	Int	10	Primary Key
2	request_code	varchar	255	
3	total_quantity	double		
4	total_price	bigint	20	

No	Nama Field	Tipe Data	Length	Constraint
5	status	varchar	255	
6	due_date	datetime		
7	payment_method	varchar	255	
8	payment_debt	varchar	255	
9	no_rek	varchar	255	
10	atas_nama	varchar	255	
11	image_nota	varchar	255	
12	image_item	varchar	255	
13	image_surat_jalan	varchar	255	
14	user_id	int	10	Foreign Key
15	house_id	int	10	Foreign Key
16	week	varchar	255	
17	created_at	timestamp		
18	updated_at	timestamp		
19	deleted_at	timestamp		
20	payment_status	varchar	255	

6. Tabel *Houses Expenditure Item*

- a. Nama Tabel : expenditure_item
- b. Primary Key : id
- c. Foreign Key : house_expenditure_id

- d. Fungsi : Menyimpan tiap item pembelian

Tabel 3.33 Tabel *Houses Expenditure Item*

No	Nama Field	Tipe Data	Length	Constraint
1	Id	Int	10	<i>Primary Key</i>
2	name_item	Varchar	255	
3	Quantity	Double		
4	unit_price	Int	11	
5	unit_type	Varchar	255	
6	house_expenditure_id	Int	10	<i>Foreign Key</i>
7	created_at	Timestamp		
8	updated_at	Timestamp		
9	deleted_at	Timestamp		

7. Tabel *Warehouse Item*

- a. Nama Tabel : warehouse_item
- b. Primary Key : id
- c. Foreign Key : -
- d. Fungsi : Menyimpan bahan baku gudang

Tabel 3.34 Tabel *Warehouse Item*

No	Nama Field	Tipe Data	Length	Constraint
1	id	Int	10	<i>Primary Key</i>
2	unit_name	varchar	100	
3	quantity	double		
4	unit_price	Int	11	
5	images	varchar	255	
6	unit_type	varchar	100	
7	due_date	datetime		
8	status	varchar	255	
9	created_at	timestamp		
10	updated_at	timestamp		
11	deleted_at	timestamp		

8. Tabel *Expenditure Warehouse*

- a. Nama Tabel : expenditure_warehouse
- b. Primary Key : id
- c. Foreign Key : user_id, house_id
- d. Fungsi : Menyimpan data pengeluaran gudang

Tabel 3.35 Tabel *Expenditure Warehouse*

No	Nama Field	Tipe Data	Length	Constraint
1	Id	int	10	<i>Primary Key</i>
2	request_code	varchar	255	
3	total_quantity	int	11	
4	total_price	int	11	
5	Status	varchar	100	
6	image_surat_jalan	varchar	255	
7	image_item	varchar	255	
8	user_id	int	10	<i>Foreign Key</i>
9	house_id	int	10	<i>Foreign Key</i>
10	created_at	timestamp		
11	updated_at	timestamp		
12	deleted_at	timestamp		

9. Tabel *Expenditure Warehouse Item*

- a. Nama Tabel : expenditure_warehouse_item
- b. *Primary Key* : id
- c. *Foreign Key* : warehouse_item_id, expenditure_warehouse_id
- d. Fungsi : Menyimpan detail item pengeluaran gudang

Tabel 3.36 Tabel *Expenditure Warehouse Item*

No	Nama Field	Tipe Data	Length	Constraint
1	Id	Int	10	<i>Primary Key</i>
2	warehouse_item_id	Int	10	<i>Foreign Key</i>
3	Quantity	Int	11	
4	expenditure_warehouse_id	Int	10	<i>Foreign Key</i>
5	created_at	Timestamp		
6	updated_at	Timestamp		
7	deleted_at	Timestamp		

10. Tabel *Notes*
- a. Nama Tabel : notes
 - b. Primary Key : id
 - c. Foreign Key : user_id, houses_id
 - d. Fungsi : Menyimpan data catatan

Tabel 3.37 Tabel *Notes*

No	Nama Field	Tipe Data	Length	Constraint
1	id	Int	10	<i>Primary Key</i>
2	note	Varchar	255	
3	user_id	Int	10	<i>Foreign Key</i>
4	houses_id	Int	10	<i>Foreign Key</i>

No	Nama Field	Tipe Data	Length	Constraint
5	created_at	timestamp		
6	updated_at	timestamp		

11. Tabel Progress

- a. Nama Tabel : progres
- b. Primary Key : id
- c. Foreign Key : houses_id
- d. Fungsi : Menyimpan data progres

Tabel 3.38 Tabel Progress

No	Nama Field	Tipe Data	Length	Constraint
1	id	Int	10	Primary Key
2	status	Varchar	100	
3	image	Varchar	255	
4	houses_id	Int	10	Foreign Key
5	created_at	timestamp		
6	updated_at	timestamp		

3.10 User Interface

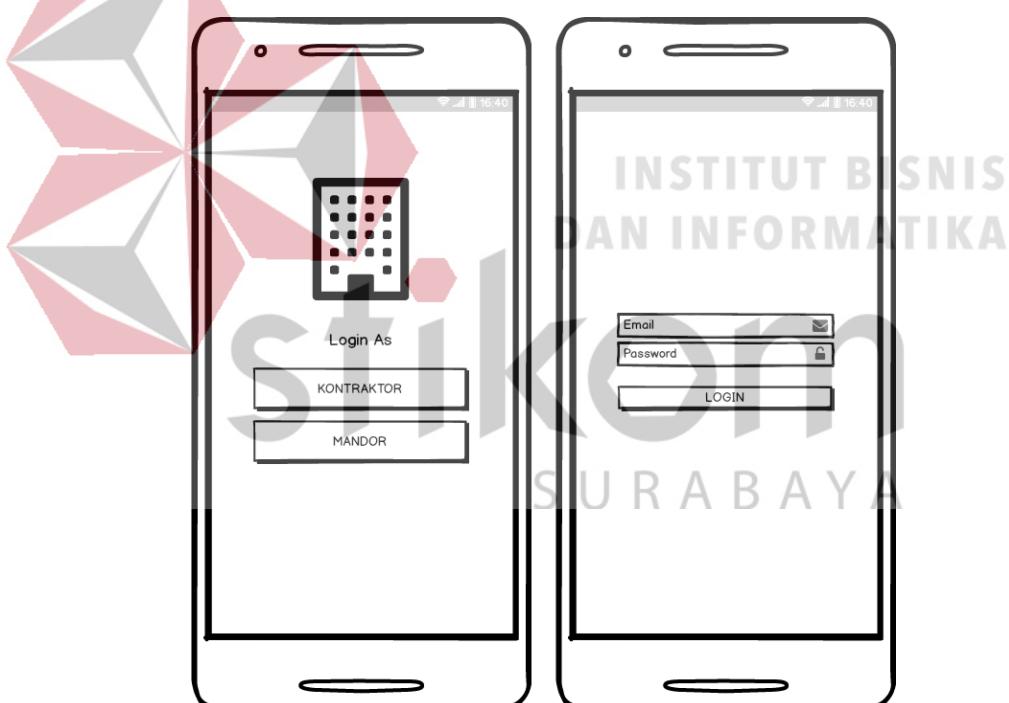
Desain *user interface* yang akan dibuat merupakan acuan desain penggerjaan yang akan dilakukan dengan mengacu pada data yang didapatkan dari proses sebelumnya.

A. User Interface Kontraktor

Berikut desain *user interface* tipe pengguna kontraktor pada aplikasi pengawasan kinerja mandor:

1. Halaman *Login*

Menu *login* ini, digunakan kontraktor untuk dapat masuk ke dalam aplikasi dengan memilih tipe pengguna kontraktor dan kemudian memasukkan data *email* dan *password*. Jika terjadi ketidaksesuaian pada *password* atau *email* yang digunakan maka sistem akan memunculkan peringatan.



Gambar 0.46 Desain Halaman *Login* (Kontraktor)

2. Menu Utama

Pada menu utama ini pengguna dapat memilih menu-menu yang tersedia dengan cara menekan tombol menu yang terdapat pada pojok kiri

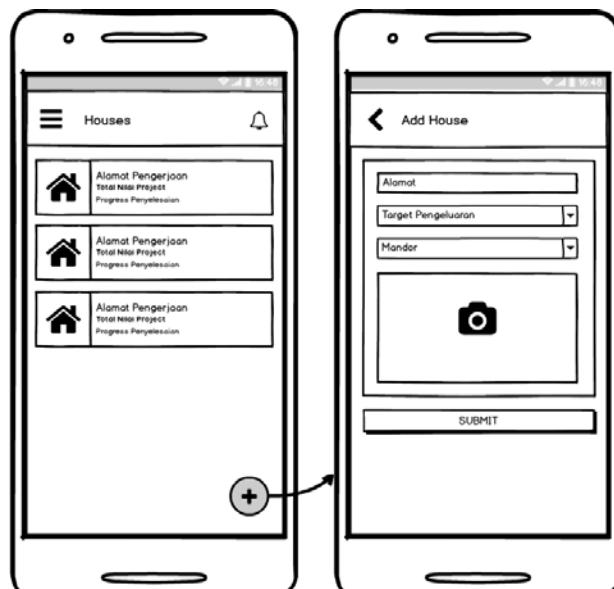
atas aplikasi. Untuk masuk ke profil, tekan gambar pada menu utama. Tipe pengguna kontraktor dapat melakukan pembaharuan pada profil miliknya.



Gambar 0.47 Desain Menu Utama (Kontraktor)

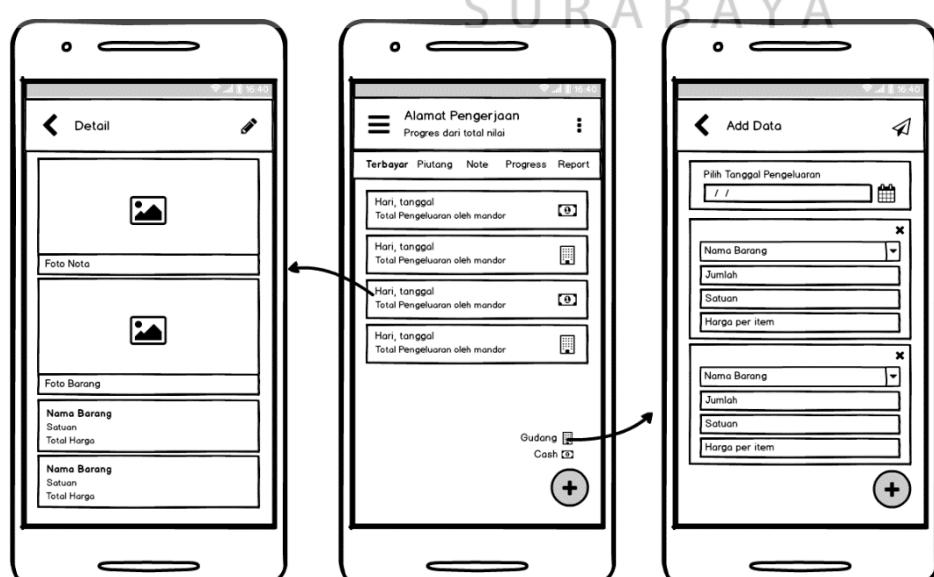
3. Menu *Houses*

Pada menu *houses* ini akan ditampilkan daftar dari proyek yang telah dibuat dan pengguna juga dapat mengelola data proyek tersebut. Untuk menambahkan data, pengguna dapat menekan *icon plus* pada layar hingga muncul tampilan “*add House*”.

Gambar 0.48 Desain Menu *Houses* (Kontraktor)

4. Menu Terbayar

Pada menu terbayar ini akan ditampilkan seluruh pengeluaran yang telah terbayarkan, kontraktor juga dapat mengelola data pengeluaran terbayar pada halaman ini. Pengguna dapat menambahkan data pengeluaran melalui dua cara yaitu mengambil barang di gudang atau dengan pembelian tunai.



Gambar 0.49 Desain Menu Terbayar (Kontraktor)

5. Menu Piutang

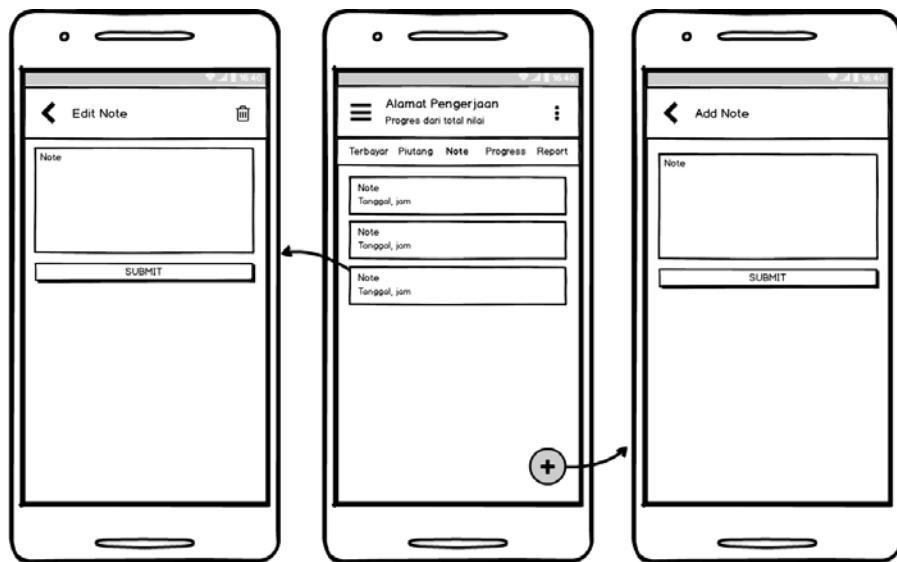
Pada menu piutang, pengguna dapat mengelola data pengeluaran piutang yang dilakukan sendiri atau dilakukan oleh mandor. Untuk menambahkan sendiri data, pengguna dengan tipe kontraktor dapat menekan *icon plus* pada layar hingga muncul tampilan “*add data*”. Untuk melakukan perubahan pada data yang telah ada, pengguna dapat memilih data yang akan diubah hingga muncul tampilan “*detail*”.



Gambar 0.50 Desain Menu piutang (Kontraktor)

6. Menu Note

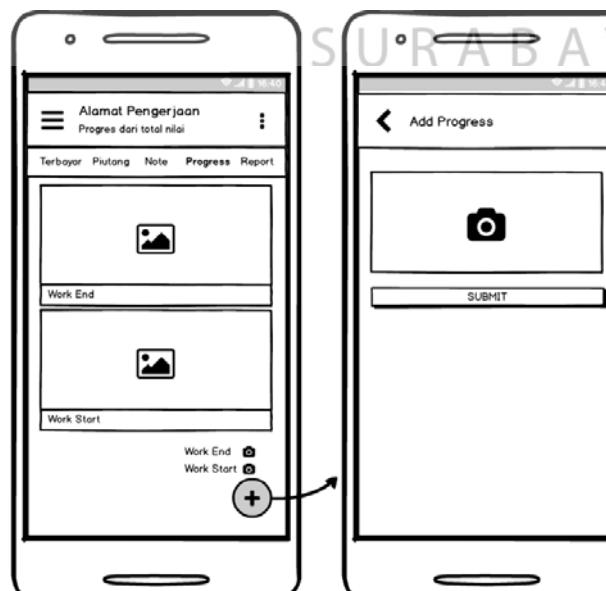
Pada menu *note* ini, pengguna dapat melakukan pencatatan yang spesifik pada proyek yang dikerjakan. Catatan yang telah dibuat dapat dibaca dan diubah oleh tipe pengguna mandor dan tipe pengguna kontraktor.



Gambar 0.51 Desain Note (Kontraktor)

7. Menu Progress

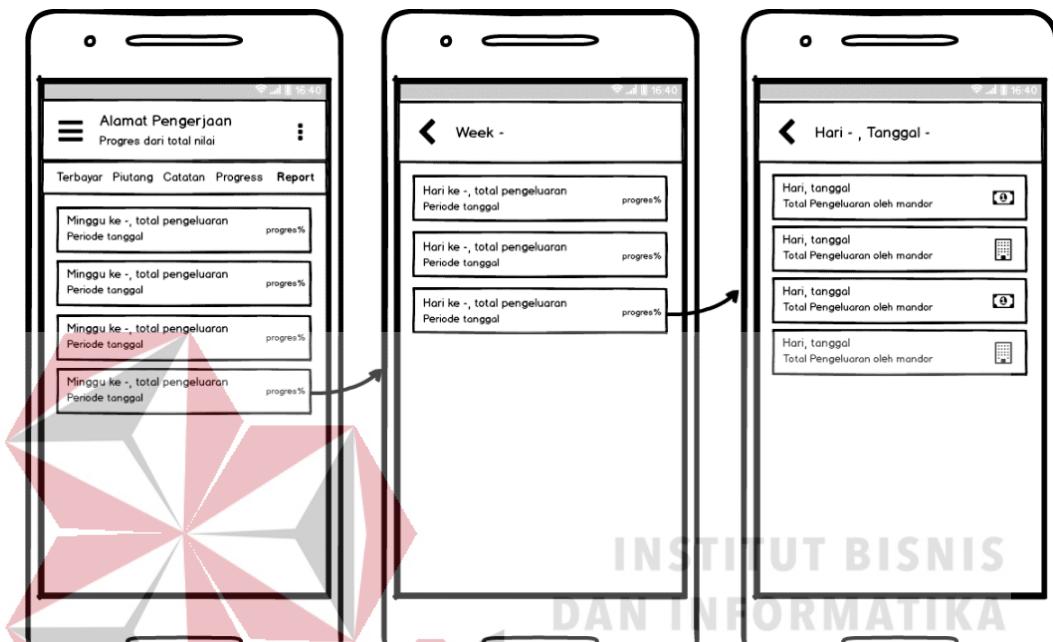
Pada menu *progress*, pengguna dapat melihat dan menambahkan kondisi proyek pada saat mulai jam kerja atau sesudah jam kerja. Untuk menambahkan pengguna dapat memilih “*icon plus*” pada tampilan hingga muncul tampilan “*add progress*”. Terdapat dua pilihan penambahan data progress yaitu saat awal pekerjaan atau akhir pekerjaan.



Gambar 0.52 Desain Progress (Kontraktor)

8. Menu Report

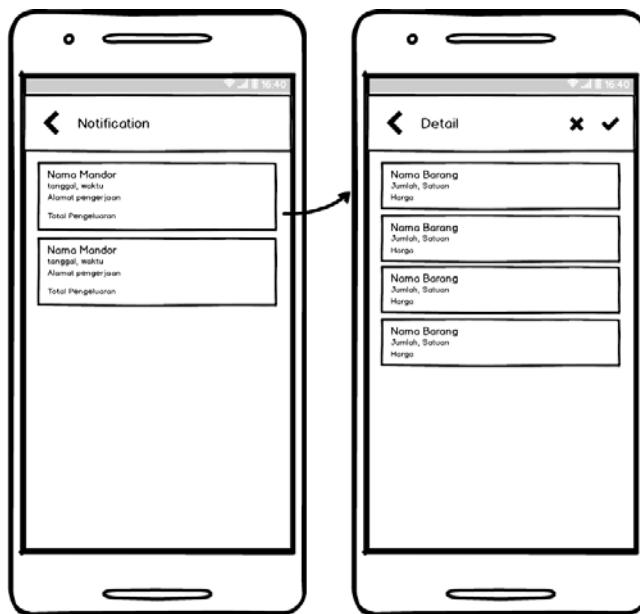
Menu *report* ini, digunakan untuk melihat catatan pengeluaran secara *detail* dari mingguan, hingga harian. Pengguna dapat menekan salah satu data untuk melihat *detil* dari tiap data yang tampil.



Gambar 0.53 Desain Menu Report (Kontraktor)

9. Menu Notifikasi

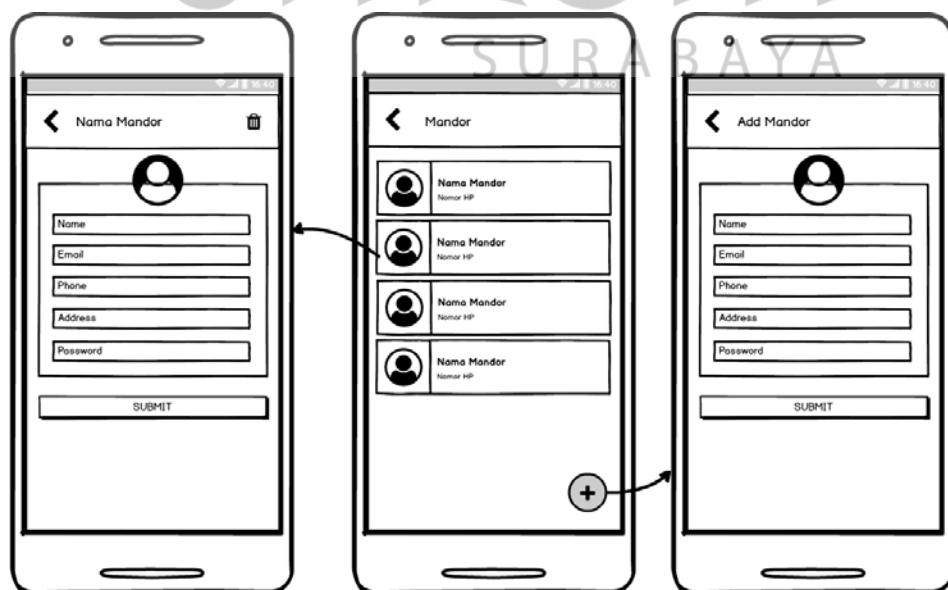
Pada menu ini, pengguna akan melihat data daftar permintaan pembelian dari mandor untuk dilakukan persetujuan pembelian bahan baku kebutuhan proyek. Untuk menyetujui pembelian tersebut pengguna dapat menekan “*icon check*” pada tampilan dan untuk menolak pengguna dapat menekan “*icon cross*” .



Gambar 0.54 Desain Menu Notifikasi (Kontraktor)

10. Menu Mandor

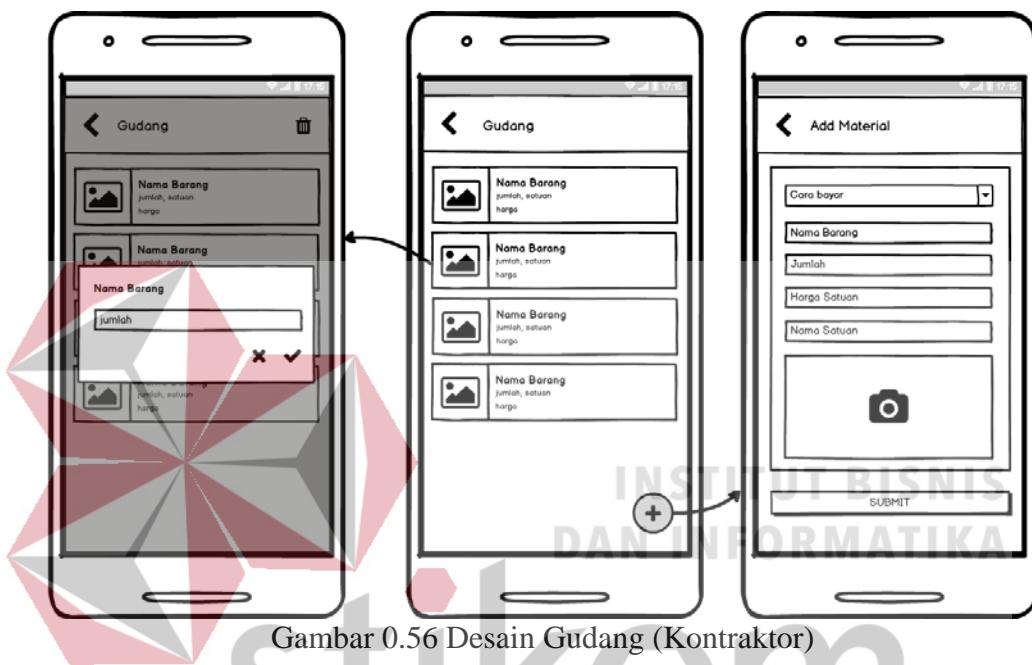
Pada menu mandor, pengguna dapat melakukan pengolahan data dan juga perubahan *password* mandor jika mandor lupa akan *password* lamanya. Untuk menambahkan data, pengguna dapat menekan “icon plus” pada tampilan hingga muncul tampilan “add mandor”.



Gambar 0.55 Desain Menu Mandor (Kontraktor)

11. Menu Gudang

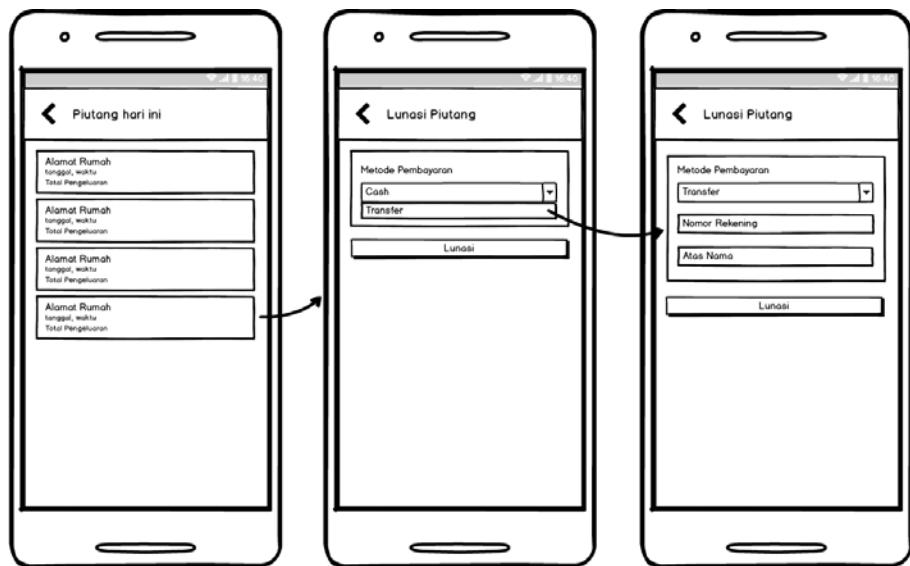
Pada menu gudang ini, pengguna dapat mengelola data bahan baku yang terdapat di gudang untuk kebutuhan proyek. Untuk menambahkan data, pengguna dapat menekan “*icon plus*” pada tampilan hingga muncul tampilan “*add material*”.



Gambar 0.56 Desain Gudang (Kontraktor)

12. Menu Bayar Piutang

Pada menu piutang ini, kontraktor dapat melakukan pencatatan pembayaran dan mendapatkan notifikasi bahwa hari ini ada pengeluaran piutang yang telah jatuh tempo. Untuk melakukan pembayaran, pengguna dapat menekan tagihan yang akan dibayarkan hingga muncul tampilan “lunasi piutang”. Terdapat dua cara pembayaran yaitu dengan *transfer* atau *cash*.



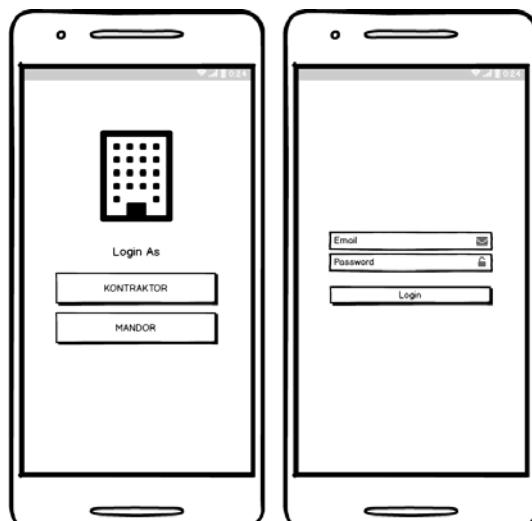
Gambar 0.57 Desain Menu Piutang (Kontraktor)

B. User Interface Mandor

Berikut desain *user interface* tipe pengguna mandor pada aplikasi pengawasan kinerja mandor:

1. Halaman *Login*

Menu *login* ini, digunakan mandor untuk dapat masuk ke dalam aplikasi. Mandor harus memilih hak akses mandor untuk dapat mengakses halaman ini.



Gambar 0.58 Desain Halaman *Login* (Mandor)

2. Menu Utama

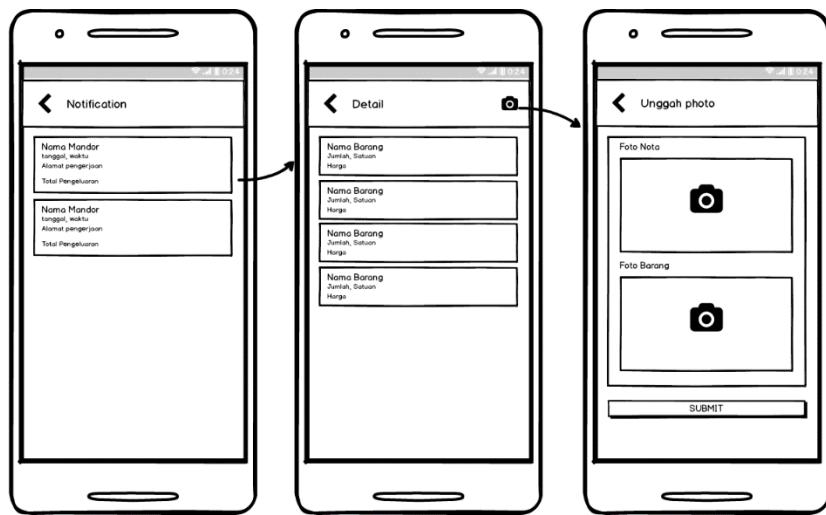
Menu utama ini, digunakan mandor untuk dapat mengakses semua fitur yang ada dalam aplikasi termasuk profil pengguna. Menu yang terdapat di sini berbeda dengan menu yang tersedia untuk kontraktor. Pengguna dapat mengakses menu *profile* dengan menekan *icon user* pada menu utama. Pengguna dengan tipe mandor dapat mengubah data mereka termasuk *password*.



Gambar 0.59 Desain Menu Utama (Mandor)

3. Menu Notifikasi

Menu notifikasi ini, digunakan mandor sebagai pemberitahuan hasil perizinan permohonan pembelian. Jika permohonan diterima, pengguna dapat melakukan pembelian dan menambahkan data nota dan data barang untuk menyimpan transaksi pembelian.



Gambar 0.60 Desain Menu Notifikasi (Mandor)

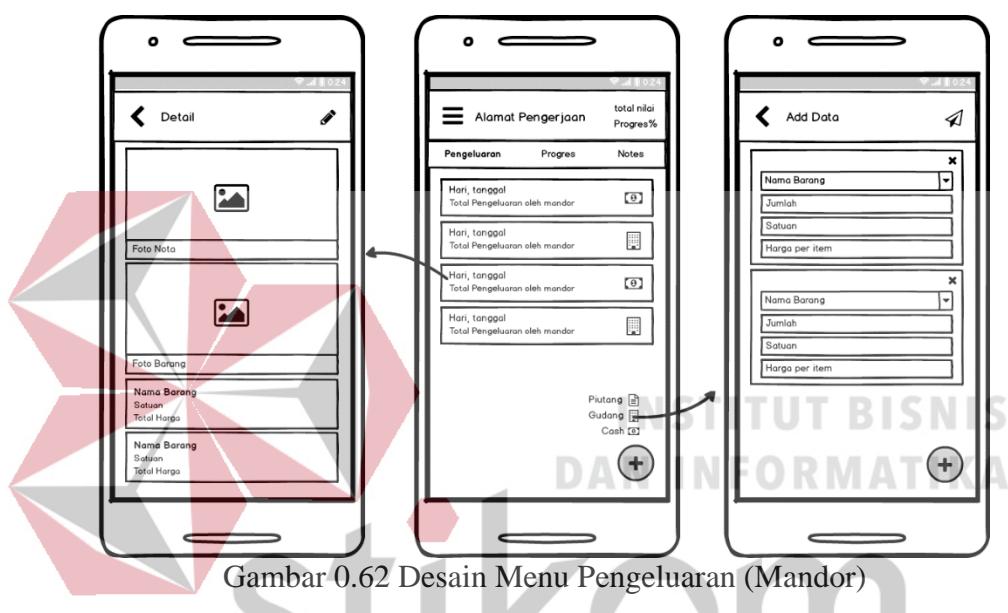
4. Menu *Houses*

Menu ini digunakan mandor untuk dapat mengelola pengeluaran dan progres proyek yang ditugaskan oleh kontraktor kepadanya. Untuk melakukan pengelolaan proyek, mandor dapat memilih data proyek mana yang akan dikelola.

Gambar 0.61 Desain *Houses* (Mandor)

5. Menu Pengeluaran

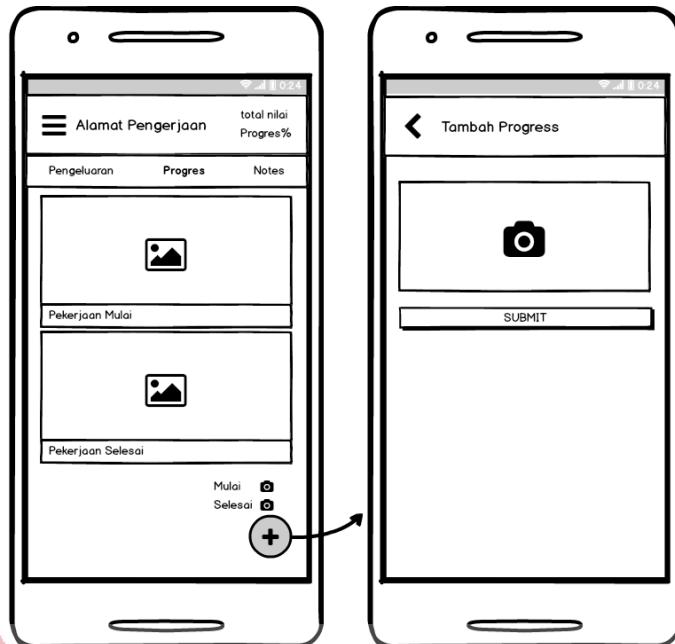
Menu ini digunakan mandor untuk dapat mengelola pengeluaran dan melihat pencatatan pengeluaran yang dilakukan hanya oleh dirinya. Untuk menambah data pengeluaran, pengguna dapat menekan *icon plus* dan memilih tipe pengeluarannya pada tampilan hingga muncul tampilan “*add data*”.



Gambar 0.62 Desain Menu Pengeluaran (Mandor)

6. Menu Progress

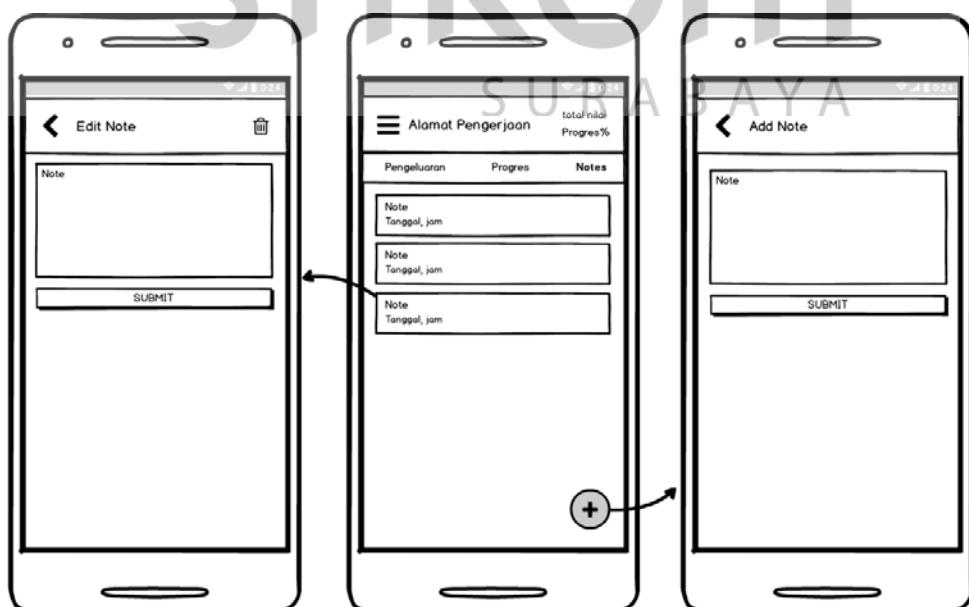
Menu ini digunakan mandor untuk mengelola *progres* proyek dimulai dari awal jam kerja hingga jam kerja berakhir. Pada menu ini mandor hanya dapat menambahkan data tanpa dapat menghapus atau mengubahnya.



Gambar 0.63 Desain Menu *Progress* (Mandor)

7. Menu Notes

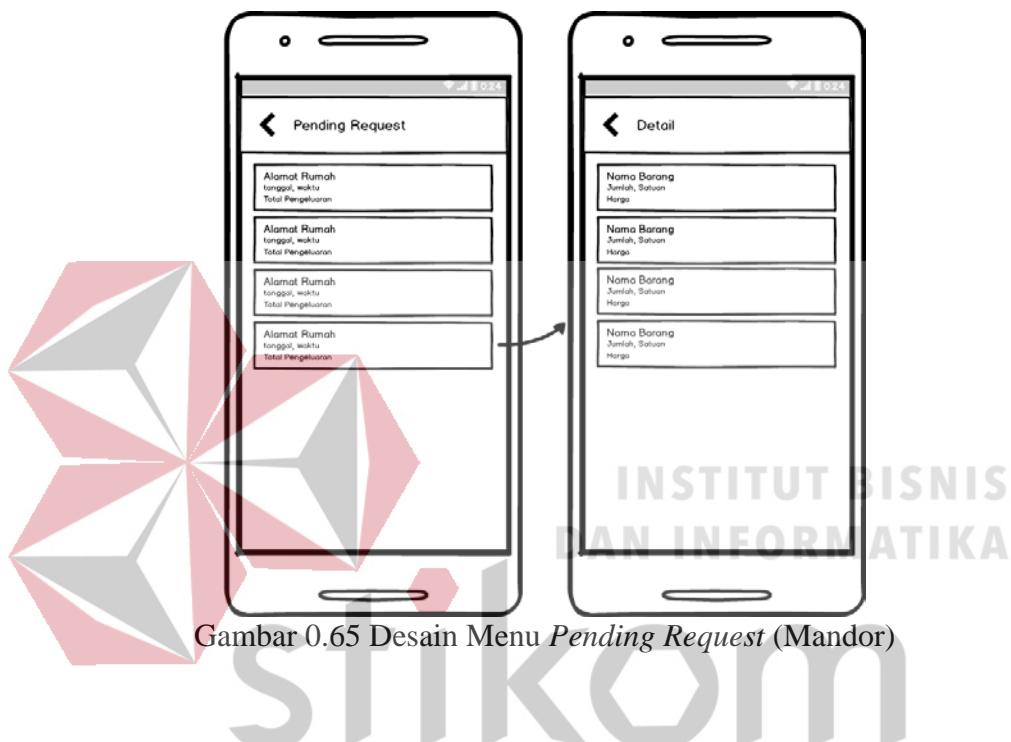
Menu ini digunakan mandor untuk mengelola catatan pada proyek yang sedang dikerjakan. Untuk menambahkan, pengguna dapat menekan *icon plus* pada tampilan hingga muncul tampilan “add note”.



Gambar 0.64 Desain Menu *Notes* (Mandor)

8. Menu *Pending Request*

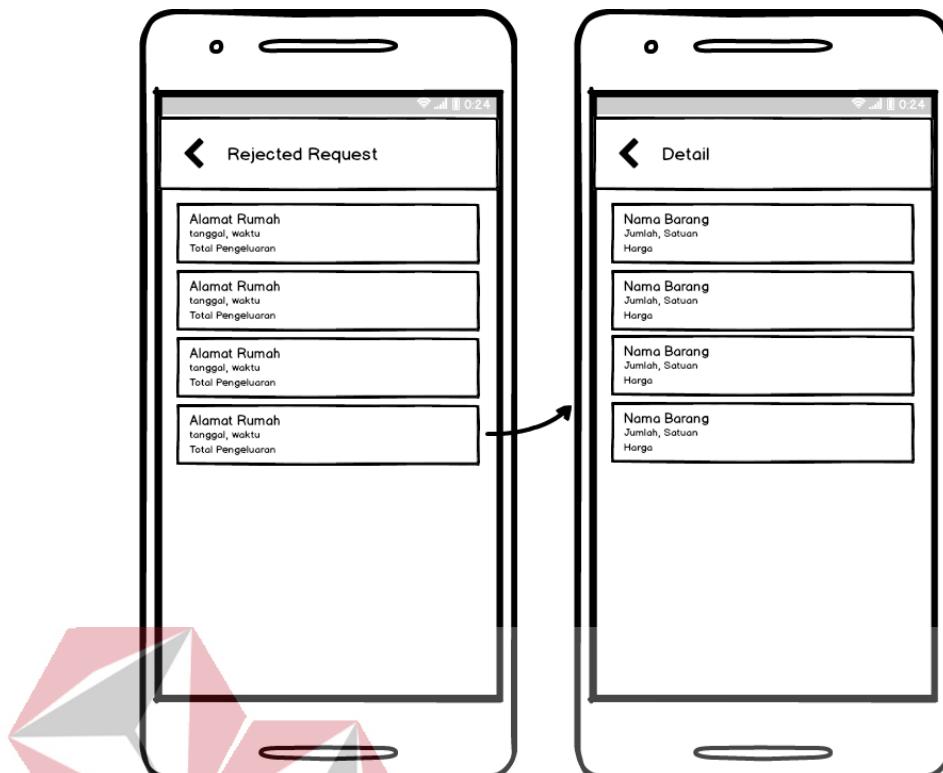
Menu ini digunakan mandor untuk mengetahui permintaan pembelian mana saja yang masih belum diproses oleh kontraktor. Selama data masih terdaftar pada menu ini, maka mandor tidak dapat melakukan pembelian.



Gambar 0.65 Desain Menu *Pending Request* (Mandor)

9. Menu *Rejected Request*

Menu ini digunakan mandor untuk mengetahui permintaan pembelian mana saja yang telah ditolak oleh kontraktor. Permintaan yang telah ditolak tidak dapat diubah ataupun dilakukan perizinan ulang.



Gambar 0.66 Desain Menu *Rejected Request* (Mandor)

10. Menu Piutang

Menu ini digunakan mandor untuk mengetahui daftar pengeluaran yang telah jatuh tempo. Dengan adanya data ini, mandor dapat membantu mengingatkan kontraktor bahwa ada tagihan yang harus diselesaikan hari itu.



3.11 Design Testing

Pada tahap *design testing* ini akan ditentukan fitur manakah yang akan divalidasi setelah sistem selesai dibuat guna memastikan fungsionalitas berjalan sesuai dengan tujuan dan kebutuhan. Menu dan fitur yang akan diuji akan dipilih berdasarkan *design interface* dan kebutuhan pengembangan sistem.

A. Design Testing Kontraktor

Berikut ini adalah *design testing* yang dibuat untuk tipe pengguna kontraktor pada aplikasi pengawasan kinerja mandor yang dibuat berdasarkan data yang diperoleh pada proses sebelumnya:

1. Halaman *Login*

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *login*. Data yang akan digunakan dalam tes pada halaman *login* ini adalah sebagai berikut:

Tabel 3.39 *Test Data* Halaman *Login*

<i>Test Data</i>	<i>Email</i>	<i>Password</i>
TD01	edo@mkds.com	123456
TD02	edo@mkds.com	qwerty
TD03	edo@mkds.com	(kosong)
TD04	(kosong)	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.40 *Test Case* Halaman *Login*

<i>Test Case</i>	<i>Tujuan</i>	<i>Test Data</i>	<i>Tahapan</i>	<i>Hasil yang Diharapkan</i>
TC01	Mengetahui respon sistem terhadap masukan data benar	TD01	<ol style="list-style-type: none"> 1. Masukkan <i>email</i> 2. Masukkan <i>password</i> 3. Tekan <i>login</i> 	Pengguna berhasil <i>login</i>

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC02	Mengetahui respon sistem terhadap masukan data salah	TD02	1. Masukkan <i>email</i> 2. Masukkan <i>password</i> 3. Tekan <i>login</i>	Pengguna gagal <i>login</i>
TC03	Mengetahui respon sistem terhadap kosongnya salah satu data	TD03	1. Masukkan <i>email</i> 2. Tekan <i>login</i>	Pengguna gagal <i>login</i>
TC04	Mengetahui respon sistem terhadap data kosong	TD04	Tekan <i>login</i>	Pengguna gagal <i>login</i>

2. Menu Utama

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada menu utama. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.41 *Test Case* Menu Utama

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC05	Mengetahui respon sistem saat menekan “icon user”	-	Tekan “icon user” pada menu utama	Halaman <i>profile</i> tampil

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC06	Mengetahui respon sistem saat menekan menu “Houses”	-	Tekan menu “Houses” pada menu utama	Halaman <i>houses</i> tampil
TC07	Mengetahui respon sistem saat menekan menu “Mandor”	-	Tekan menu “Mandor” pada menu utama	Halaman mandor tampil
TC08	Mengetahui respon sistem saat menekan menu “Gudang”	-	Tekan menu “Gudang” pada menu utama	Halaman gudang tampil
TC09	Mengetahui respon sistem saat menekan menu “Piutang”	-	Tekan menu “Piutang” pada menu utama	Halaman piutang tampil
TC10	Mengetahui respon sistem saat menekan menu “Logout”	-	Tekan menu “Logout” pada menu utama	Kembali ke halaman <i>login</i>

3. Menu *Houses*

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *houses*. Data yang digunakan saat ini merupakan data *dummy* atau data yang tidak digunakan dalam proses alsi di lapangan. Data yang akan digunakan dalam tes pada halaman *houses* ini adalah sebagai berikut:

Tabel 3.42 *Test Data Menu Houses*

Test Data	Alamat	Target Pengeluaran	Mandor	Gambar
TD05	Jl. Griya Benowo Indah O no 7	230.000.000	Pilih mandor “Fendy”	Format .jpg atau .png
TD06	(kosong)	230.000.000	Pilih mandor “Fendy”	Format .jpg atau .png
TD07	(kosong)	(kosong)	(kosong)	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.43 *Test Case Menu Houses*

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC11	Mengetahui respon sistem terhadap masukan data benar	TD05	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Masukkan Alamat 3. Masukkan Target Pengeluaran 4. Pilih Mandor 5. Masukkan Gambar 6. Tekan <i>submit</i> 	Data berhasil tersimpan

<i>Test Case</i>	Tujuan	<i>Test Data</i>	Tahapan	Hasil yang Diharapkan
TC12	Mengetahui respon sistem terhadap kosongnya salah satu data	TD06	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Masukkan Alamat Masukkan Target Pengeluaran Pilih Mandor Masukkan Gambar Tekan <i>submit</i> 	Data gagal tersimpan
TC13	Mengetahui respon sistem terhadap data kosong	TD07	Tekan <i>login</i>	Data gagal tersimpan
TC14	Mengetahui respon sistem terhadap data yang diubah	TD05	<ol style="list-style-type: none"> Pilih data proyek Pilih “icon menu” pada pojok kanan atas tampilan proyek Pilih menu “Edit” Ubah data masukan Tekan “submit” 	Data berhasil tersimpan
TC15	Mengetahui respon sistem terhadap data yang dihapus	-	<ol style="list-style-type: none"> Pilih data proyek Pilih “icon menu” pada pojok kanan atas tampilan proyek Pilih menu “Delete” 	Data berhasil terhapus

4. Menu Terbayar

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman terbayar. Data yang akan digunakan dalam tes pada halaman terbayar ini adalah sebagai berikut:

Tabel 3.44 *Test Data* Menu Terbayar

<i>Test Data</i>	Tanggal Pengeluaran	Nama Barang	Jumlah	Satuan	Harga Tiap Satuan
TD08	Tanggal pada hari dilakukan tes	Semen	5	Sak (50)	64.750
TD09	Tanggal pada hari dilakukan tes	(kosong)	5	Sak (50)	64.750
TD10	Tanggal pada hari dilakukan tes	(kosong)	(kosong)	(kosong)	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.45 *Test Case* Menu Terbayar

<i>Test Case</i>	Tujuan	<i>Test Data</i>	Tahapan	Hasil yang Diharapkan
TC16	Mengetahui respon sistem terhadap masukan data benar	TD08	1. Pilih “icon plus” pada pojok kanan bawah tampilan	Data berhasil disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
			2. Pilih menu “Cash” 3. Pilih tanggal pengeluaran 4. Masukkan nama barang 5. Masukkan jumlah 6. Masukkan satuan 7. Masukkan harga tiap satuan Tekan “icon send” pada pojok kanan atas	
TC17	Mengetahui respon sistem terhadap kosongnya salah satu data	TD09	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Cash” 3. Masukkan nama barang 4. Masukkan jumlah 5. Masukkan satuan 6. Masukkan harga tiap satuan 7. Tekan “icon send” pada pojok kanan atas	Data gagal disimpan
TC18	Mengetahui respon sistem terhadap data kosong	TD10	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Cash”	Data gagal disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
			3. Tekan “icon send” pada pojok kanan atas	
TC19	Mengetahui respon sistem terhadap masukan data benar	TD08	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Gudang” 3. Pilih tanggal pengeluaran 4. Pilih nama barang 5. Masukkan jumlah 6. Masukkan satuan 7. Masukkan harga tiap satuan 8. Tekan “icon send” pada pojok kanan atas	Data berhasil disimpan
TC20	Mengetahui respon sistem terhadap kosongnya salah satu data	TD09	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Gudang” 3. Pilih tanggal pengeluaran 4. Masukkan jumlah 5. Masukkan satuan	Data gagal disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
			<p>6. Masukkan harga tiap satuan</p> <p>7. Tekan “<i>icon send</i>” pada pojok kanan atas</p>	
TC21	Mengetahui respon sistem terhadap data kosong	TD10	<p>1. Pilih “<i>icon plus</i>” pada pojok kanan bawah tampilan</p> <p>2. Pilih menu “Gudang”</p> <p>3. Tekan “<i>icon send</i>” pada pojok kanan atas</p>	Data gagal disimpan
TC22	Mengetahui respon sistem saat melakukan pengeluaran	-	<p>1. Pilih salah satu data pada tampilan menu terbayar</p> <p>2. Pilih menu “<i>edit</i>” pada pojok kanan atas tampilan</p> <p>3. Ubah data jumlah</p> <p>4. Pilih menu “<i>save</i>” pada pojok kanan atas tampilan</p>	Data berhasil diubah

5. Menu Piutang

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman piutang. Data yang akan digunakan dalam tes pada halaman piutang ini adalah sebagai berikut:

Tabel 3.46 *Test Data* Menu Piutang

<i>Test Data</i>	Tanggal Pengeluaran	<i>Due Date</i>	Nama Barang	Jumlah	Satuan	Harga per Item
TD11	Tanggal pada hari dilakukan tes	Tanggal pada hari dilakukan tes	Semen	5	Sak (50)	64.750
TD12	Tanggal pada hari dilakukan tes	Tanggal pada hari dilakukan tes	(kosong)	5	Sak (50)	64.750
TD13	(kosong)	(kosong)	(kosong)	(kosong)	(kosong)	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.47 Test Case Menu Piutang

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC23	Mengetahui respon sistem terhadap masukan data benar	TD11	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih tanggal pengeluaran 3. Pilih <i>due date</i> 4. Masukkan nama barang 5. Masukkan jumlah 6. Masukkan satuan 7. Masukkan harga tiap satuan 8. Tekan “icon send” pada pojok kanan atas 	Data berhasil disimpan
TC24	Mengetahui respon sistem terhadap kosongnya salah satu data	TD12	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih tanggal pengeluaran 3. Pilih <i>due date</i> 4. Masukkan jumlah 5. Masukkan satuan 6. Masukkan harga tiap satuan 7. Tekan “icon send” 	Data gagal disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC25	Mengetahui respon sistem terhadap data kosong	TD13	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Tekan “icon send” pada pojok kanan atas 	Data gagal disimpan
TC26	Mengetahui respon sistem terhadap perubahan data	-	<ol style="list-style-type: none"> Pilih salah satu data pada menu piutang Pilih menu “edit” pada pojok kanan atas Ubah data Pilih menu “save” pada pojok kanan atas 	Data berhasil disimpan

6. Menu Note

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *note*. Data yang akan digunakan dalam tes pada halaman *note* ini adalah sebagai berikut:

Tabel 3.48 *Test Data Menu Note*

Test Data	Note
TD14	Pembelian terlambat kirim karena cuaca buruk tgl 15 bulan lalu
TD15	Pembelian sudah berhasil dikirim hari ini

Test Data	Note
TD16	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.49 *Test Case Menu Note*

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC27	Mengetahui respon sistem terhadap masukan data benar	TD14	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Masukkan note Tekan “submit” 	Data berhasil disimpan
TC28	Mengetahui respon sistem terhadap pengubahan data	TD15	<ol style="list-style-type: none"> Pilih salah satu data pada menu note Ubah note Tekan “submit” 	Data berhasil diubah
TC29	Mengetahui respon sistem terhadap data kosong	TD16	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Tekan “submit” 	Data gagal disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC30	Mengetahui respon sistem terhadap penghapusan data	-	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “icon trash” pada pojok kanan atas tampilan	Data berhasil terhapus

7. Menu *Progress*

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *progress*. Data yang akan digunakan dalam tes pada halaman *progress* ini adalah sebagai berikut:

Tabel 3.50 Test Data Menu *Progress*

Test Data	Gambar
TD17	Gambar sebelum pekerjaan dimulai dengan format .png atau .jpg
TD18	Gambar setelah pekerjaan selesai dengan format .png atau .jpg
TD19	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.51 Test Case Menu Progress

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC31	Mengetahui respon sistem terhadap masukan data benar	TD17	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Pilih “work start” Pilih gambar Tekan “submit” 	Data berhasil disimpan
TC32	Mengetahui respon sistem terhadap data kosong	TD19	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Pilih “work start” Tekan “submit” 	Data gagal disimpan
TC33	Mengetahui respon sistem terhadap masukan data benar	TD18	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Pilih “work end” Pilih gambar Tekan “submit” 	Data berhasil disimpan
TC34	Mengetahui respon sistem terhadap data kosong	TD19	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Pilih “work end” Tekan “submit” 	Data gagal disimpan

8. Menu *Report*

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *report*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.52 *Test Case* Menu *Report*

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC35	Mengetahui respon sistem jika data mingguan dipilih	-	Pilih salah satu tampilan pada data mingguan	Data berhasil tampil
TC34	Mengetahui respon sistem jika data harian dipilih	-	Pilih salah satu tampilan pada data harian	Data berhasil tampil

9. Menu Notifikasi

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman notifikasi. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.53 *Test Case* Menu Notifikasi

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC35	Mengetahui respon sistem jika	-	1. Pilih salah satu data permintaan	Data permintaan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
	permohonan pembelian disetujui		2. Pilih “icon check”	berhasil diterima
TC36	Mengetahui respon sistem jika permohonan pembelian ditolak	-	1. Pilih salah satu data permintaan 2. Pilih “icon cross”	Data permintaan berhasil ditolak

10. Menu Mandor

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman mandor. Data yang akan digunakan dalam tes pada halaman mandor ini adalah sebagai berikut:

Tabel 3.54 Test Data Menu Mandor

Test Data	Gambar	Name	Email	Phone	Address	Password
TD 20	Format .jpg atau .png	Ghana	ghana@gmail.com	0812346 10780	Jl. Griya Benowo	123456
TD 21	(kosong)	Ghana	<u>ghana@gmail.com</u>	0812346 10780	Jl. Griya Benowo	123456
TD 22	Format .jpg atau .png	Ghana	(kosong)	0812346 10780	Jl. Griya Benowo	123456

Test Data	Gambar	Name	Email	Phone	Address	Password
TD 23	(kosong)	(kosong)	(kosong)	(kosong)	(kosong)	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.55 *Test Case* Menu Mandor

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC37	Mengetahui respon sistem terhadap masukan data benar	TD20	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Pilih gambar Masukkan <i>name</i> Masukkan <i>email</i> Masukkan <i>phone</i> Masukkan <i>address</i> Masukkan <i>password</i> Tekan “submit” 	Data berhasil disimpan
TC38	Mengetahui respon sistem terhadap kosongnya salah satu data	TD21	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Masukkan <i>name</i> Masukkan <i>email</i> Masukkan <i>phone</i> 	Data gagal disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
			5. Masukkan <i>address</i> 6. Masukkan <i>password</i> 7. Tekan “ <i>submit</i> ”	
TC39	Mengetahui respon sistem terhadap kosongnya beberapa data	TD22	1. Pilih “ <i>icon plus</i> ” pada pojok kanan bawah tampilan 2. Pilih gambar 3. Masukkan <i>name</i> 4. Masukkan <i>phone</i> 5. Masukkan <i>address</i> 6. Masukkan <i>password</i> 7. Tekan “ <i>submit</i> ”	Data gagal disimpan
TC40	Mengetahui respon sistem terhadap data kosong	TD23	1. Pilih “ <i>icon plus</i> ” pada pojok kanan bawah tampilan 2. Tekan “ <i>submit</i> ”	Data gagal disimpan
TC41	Mengetahui respon sistem terhadap pengubahan data	TD20	1. Pilih salah satu data mandor 2. Pilih gambar 3. Masukkan <i>name</i> 4. Masukkan <i>email</i> 5. Masukkan <i>phone</i> 6. Masukkan <i>address</i> 7. Masukkan <i>password</i>	Data berhasil disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
			8. Tekan “submit”	
TC42	Mengetahui respon sistem terhadap penghapusan data	-	1. Pilih salah satu data mandor 2. Pilih “icon trash” pada pojok kanan atas tampilan	Data berhasil dihapus

11. Menu Gudang

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman gudang. Data yang akan digunakan dalam tes pada halaman gudang ini adalah sebagai berikut:

Tabel 3.56 Test Data Menu Gudang

Test Data	Tipe Bayar	Due Date	Nama Barang	Jumlah	Harga Satuan	Nama Satuan	Gambar
TD24	Cash	-	Semen	5	Sak (50)	64.750	Format .jpg atau .png
TD25	Cash	-	(kosong)	5	Sak (50)	64.750	Format .jpg atau .png

<i>Test Data</i>	Tipe Bayar	Due Date	Nama Barang	Jumlah	Harga Satuan	Nama Satuan	Gambar
TD26	Piutang	29/08	Semen	5	Sak (50)	64.750	Format .jpg atau .png
TD27	Piutang	29/08	(kosong)	5	Sak (50)	64.750	Format .jpg atau .png

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.57 *Test Case* Menu Gudang

<i>Test Case</i>	Tujuan	<i>Test Data</i>	Tahapan	Hasil yang Diharapkan
TC43	Mengetahui respon sistem terhadap masukan data benar	TD24	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “Cash” 3. Masukkan nama barang 4. Masukkan jumlah 5. Masukkan harga satuan 6. Masukkan nama satuan 7. Masukkan gambar 8. Tekan “submit” 	Data berhasil disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC44	Mengetahui respon sistem terhadap kosongnya beberapa data	TD25	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “Cash” 3. Masukkan jumlah 4. Masukkan harga satuan 5. Masukkan nama satuan 6. Masukkan gambar 7. Tekan “submit” 	Data gagal disimpan
TC45	Mengetahui respon sistem terhadap masukan data benar	TD26	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “Piutang” 3. Pilih “due date” 4. Masukkan nama barang 5. Masukkan jumlah 6. Masukkan harga satuan 7. Masukkan nama satuan 8. Masukkan gambar Tekan “submit” 	Data berhasil disimpan
TC46	Mengetahui respon sistem terhadap kosongnya beberapa data	TD27	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “Piutang” 3. Pilih “due date” 4. Masukkan jumlah 5. Masukkan harga satuan 6. Masukkan nama satuan 	Data gagal disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
			7. Masukkan gambar 8. Tekan “submit”	
TC47	Mengetahui respon sistem terhadap pengubahan data	-	1. Pilih salah satu data barang gudang 2. Ubah data masukan pada jumlah barang 3. Tekan “save”	Data berhasil diubah
TC48	Mengetahui respon sistem terhadap penghapusan data	-	1. Pilih salah satu data barang gudang 2. Ubah data masukan pada jumlah barang 3. Tekan “delete”	Data berhasil dihapus

12. Menu Bayar Piutang

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman bayar piutang. Data yang akan digunakan dalam tes pada halaman bayar piutang ini adalah sebagai berikut:

Tabel 3.58 *Test Data* Menu Piutang

Test Data	Method	No Rekening	Atas Nama
TD28	Cash	-	-
TD29	Transfer	8630026769	Intan Nurlaila
TD30	Transfer	(kosong)	Intan Nurlaila

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.59 *Test Case* Menu Piutang

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC49	Mengetahui respon sistem terhadap masukan data benar	TD28	<ol style="list-style-type: none"> 1. Pilih salah satu data piutang 2. Pilih <i>method</i> “cash” 3. Tekan tombol “lunasi” 	Data berhasil disimpan
TC50	Mengetahui respon sistem terhadap masukan data benar	TD29	<ol style="list-style-type: none"> 1. Pilih salah satu data piutang 2. Pilih <i>method</i> “transfer” 3. Tekan tombol “lunasi” 	Data berhasil disimpan
TC51	Mengetahui respon sistem terhadap kosongnya salah satu data	TD30	<ol style="list-style-type: none"> 1. Pilih salah satu data piutang 2. Pilih <i>method</i> “transfer” 3. Tekan tombol “lunasi” 	Data gagal disimpan

B. Design Testing Mandor

Berikut ini adalah *design testing* yang dibuat untuk tipe pengguna kontraktor pada aplikasi pengawasan kinerja mandor:

1. Halaman *Login*

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *login*. Data yang akan digunakan dalam tes pada halaman *login* ini adalah sebagai berikut:

Tabel 3.60 *Test Data Menu Login*

<i>Test Data</i>	<i>Email</i>	<i>Password</i>
TD31	fendy0249@gmail.com	123456
TD32	fendy0249@gmail.com	qwerty
TD33	fendy0249@gmail.com	(kosong)
TD34	(kosong)	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.61 *Test Case Menu Login*

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC52	Mengetahui respon sistem terhadap masukan data benar	TD31	1. Masukkan <i>email</i> 2. Masukkan <i>password</i> 3. Tekan <i>login</i>	Pengguna berhasil <i>login</i>
TC53	Mengetahui respon sistem terhadap masukan data salah	TD32	1. Masukkan <i>email</i> 2. Masukkan <i>password</i> 3. Tekan <i>login</i>	Pengguna gagal <i>login</i>
TC54	Mengetahui respon sistem terhadap kosongnya salah satu data	TD33	1. Masukkan <i>email</i> 2. Tekan <i>login</i>	Pengguna gagal <i>login</i>
TC55	Mengetahui respon sistem terhadap data kosong	TD34	Tekan <i>login</i>	Pengguna gagal <i>login</i>

2. Halaman Utama

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada menu utama. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.62 Test Case Menu Utama

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC56	Mengetahui respon sistem saat menekan “icon user”	-	Tekan “icon user” pada menu utama	Halaman <i>profile</i> tampil
TC57	Mengetahui respon sistem saat menekan menu “Houses”	-	Tekan menu “Houses” pada menu utama	Halaman <i>houses</i> tampil
TC58	Mengetahui respon sistem saat menekan menu “Pending Request”	-	Tekan menu “Pending Request” pada menu utama	Halaman <i>pending request</i> tampil
TC59	Mengetahui respon sistem saat menekan menu “Rejected Request”	-	Tekan menu “Rejected Request” pada menu utama	Halaman <i>rejected request</i> tampil
TC60	Mengetahui respon sistem saat menekan menu “Piutang”	-	Tekan menu “Piutang” pada menu utama	Halaman piutang tampil
TC61	Mengetahui respon sistem saat menekan menu “Logout”	-	Tekan menu “Logout” pada menu utama	Kembali ke halaman <i>login</i>

3. Menu Notifikasi

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman notifikasi. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.63 *Test Case* Menu Notifikasi

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC62	Mengetahui respon sistem jika permohonan pembelian disetujui	-	<ol style="list-style-type: none"> 1. Pilih salah satu data permintaan yang diterima 2. Pilih “icon camera” pada pojok kanan atas tampilan 3. Masukkan gambar foto nota 4. Masukkan gambar foto barang 5. Tekan ”submit” 	Data permintaan berhasil diterima

4. Menu *Houses*

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *houses*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.64 *Test Case Menu Houses*

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC63	Mengetahui respon sistem saat menampilkan data	-	Pilih salah satu data houses	Menampilkan detail data houses yang dipilih

5. Menu Pengeluaran

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman pengeluaran. Data yang akan digunakan dalam tes pada halaman pengeluaran ini adalah sebagai berikut:

Tabel 3.65 *Test Data Menu Pengeluaran*

Test Data	Tanggal Pengeluaran	Due Date	Nama Barang	Jumlah	Satuan	Harga Tiap Satuan
TD35	Tanggal pada hari dilakukan tes		Semen	5	Sak (50)	64.750
TD36	(kosong)		Semen	5	Sak (50)	64.750
TD37	(kosong)		(kosong)	(kosong)	(kosong)	(kosong)
TD38	-	Tanggal pada hari dilakukan tes	Semen	5	Sak (50)	64.750

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.66 *Test Case* Menu Pengeluaran

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC64	Mengetahui respon sistem terhadap masukan data benar	TD35	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Cash” 3. Pilih tanggal pengeluaran 4. Masukkan nama barang 5. Masukkan jumlah 6. Masukkan satuan 7. Masukkan harga tiap satuan <p>Tekan “icon send” pada pojok kanan atas</p>	Data berhasil disimpan
TC65	Mengetahui respon sistem terhadap kosongnya salah satu data	TD36	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Cash” 3. Masukkan nama barang 4. Masukkan jumlah 5. Masukkan satuan 6. Masukkan harga tiap satuan 7. Tekan “icon send” pada pojok kanan atas 	Data gagal disimpan
TC66	Mengetahui respon	TD37	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 	Data gagal disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
	sistem terhadap data kosong		2. Pilih menu “Cash” 3. Tekan “icon send” pada pojok kanan atas	
TC67	Mengetahui respon sistem terhadap masukan data benar	TD35	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Gudang” 3. Pilih tanggal pengeluaran 4. Pilih nama barang 5. Masukkan jumlah 6. Masukkan satuan 7. Masukkan harga tiap satuan 8. Tekan “icon send” pada pojok kanan atas	Data berhasil disimpan
TC68	Mengetahui respon sistem terhadap kosongnya salah satu data	TD36	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Gudang” 3. Pilih nama barang 4. Masukkan jumlah 5. Masukkan satuan 6. Masukkan harga tiap satuan 7. Tekan “icon send” pada pojok kanan atas	Data gagal disimpan
TC69	Mengetahui respon sistem terhadap data kosong	TD37	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Gudang” 3. Tekan “icon send” pada pojok kanan atas	Data gagal disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC70	Mengetahui respon sistem jika salah satu data dipilih	-	Pilih salah satu data pada tampilan menu terbayar	Muncul <i>detail</i> tampilan pengeluaran
TC71	Mengetahui respon sistem terhadap masukan data benar	TD38	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Piutang” 3. Pilih “due date” 4. Pilih nama barang 5. Masukkan jumlah 6. Masukkan satuan 7. Masukkan harga per item 8. Tekan “icon send” pada pojok kanan atas 	Data berhasil disimpan

6. Menu *Progress*

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *progress*. Data yang akan digunakan dalam tes pada halaman *progress* ini adalah sebagai berikut:

Tabel 3.67 *Test Data Menu Progress*

Test Data	Gambar
TD39	Gambar sebelum pekerjaan dimulai dengan format .png atau .jpg
TD40	Gambar setelah pekerjaan selesai dengan format .png atau .jpg

Test Data	Gambar
TD41	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.68 *Test Case Menu Progress*

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC72	Mengetahui respon sistem terhadap masukan data benar	TD39	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Pilih “work start” Pilih gambar Tekan “submit” 	Data berhasil disimpan
TC73	Mengetahui respon sistem terhadap data kosong	TD41	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Pilih “work start” Tekan “submit” 	Data gagal disimpan
TC74	Mengetahui respon sistem terhadap masukan data benar	TD40	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Pilih “work end” Pilih gambar Tekan “submit” 	Data berhasil disimpan

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC75	Mengetahui respon sistem terhadap data kosong	TD41	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “work end” 3. Tekan “submit” 	Data gagal disimpan

7. Menu Notes

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *note*. Data yang akan digunakan dalam tes pada halaman *note* ini adalah sebagai berikut:

Tabel 3.69 Test Data Menu Note

Test Data	Note
TD42	Pembelian terlambat kirim karena cuaca buruk tgl 15 bulan ini
TD43	Pembelian sudah berhasil dikirim besok hari
TD44	(kosong)

Test data di atas akan digunakan dalam proses pengujian *test case*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.70 Test Case Menu Note

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC76	Mengetahui respon sistem terhadap masukan data benar	TD42	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Masukkan note 3. Tekan “submit” 	Data berhasil disimpan
TC77	Mengetahui respon sistem terhadap pengubahan data	TD43	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Ubah note 3. Tekan “submit” 	Data berhasil diubah
TC78	Mengetahui respon sistem terhadap data kosong	TD44	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Tekan “submit” 	Data gagal disimpan
TC79	Mengetahui respon sistem terhadap penghapusan data	-	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “icon trash” pada pojok kanan atas tampilan 	Data berhasil terhapus

8. Menu *Pending Request*

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *pending request*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.71 *Test Case* Menu *Pending Request*

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC80	Mengetahui respon sistem saat menampilkan data	-	Pilih salah satu data <i>pending payment</i>	Menampilkan detail data <i>pending payment</i> yang dipilih

9. Menu *Rejected Request*

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman *rejected request*. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.72 *Test Case* Menu *Rejected Request*

Test Case	Tujuan	Test Data	Tahapan	Hasil yang Diharapkan
TC81	Mengetahui respon sistem saat menampilkan data	-	Pilih salah satu data <i>rejected request</i>	Menampilkan detail data <i>rejected request</i> yang dipilih

10. Menu Piutang

Rancangan uji coba ini digunakan untuk melakukan validasi fungsional pada halaman piutang. Dengan mengacu pada validasi yang dibutuhkan oleh sistem, maka dapat dibuat *test case* sebagai berikut:

Tabel 3.73 *Test Case* Menu Piutang

<i>Test Case</i>	Tujuan	<i>Test Data</i>	Tahapan	Hasil yang Diharapkan
TC82	Mengetahui respon sistem saat menampilkan data	-	Pilih salah satu data piutang	Menampilkan detail data piutang yang dipilih



BAB IV

IMPLEMENTASI DAN EVALUASI

Bab ini merupakan tahapan terakhir dari pembuatan aplikasi pengawasan kinerja mandor. Pada tahapan ini akan berisi proses *deployment*, dan hasil *testing* dari sistem tersebut.

4.1 Deployment

Pada tahap ini, akan dilakukan langkah-langkah yang diperlukan untuk membuat aplikasi pengawasan kinerja mandor yang diantaranya:

1. Membuat *Database*

Pada tahap ini pembuatan database akan menggunakan MySql sebagai *tools*. Database yang dibuat akan disesuaikan dengan perancangan database yang telah dibuat pada bab sebelumnya.

2. Membuat API

Pada tahap ini pembuatan API akan menggunakan bahasa pemrograman PHP dengan menggunakan *web server* Apache yang dikerjakan menggunakan *tools text editor* Sublime Text 3. Pemasangan aplikasi yang diperlukan langsung dilakukan di *server* dan pembuatan API ini telah menggunakan *framework* Laravel.

Berikut ini adalah daftar *controller* yang terdapat pada API yang akan digunakan pada aplikasi pengawasan kinerja mandor:

Tabel 4.1 Daftar *Controller API*

No	Nama <i>Controller</i>	Keterangan
1	AuthController	<i>Controller</i> ini bertugas untuk mengelola data pengguna termasuk sistem dan keamanan <i>login</i>
2	ExpenditureController	<i>Controller</i> ini bertugas untuk mengelola data pengeluaran yang terjadi pada proyek
3	EpenditureWarehouseController	<i>Controller</i> ini bertugas untuk mengelola data pengeluaran pada gudang yang digunakan untuk keperluan proyek
4	HouseController	<i>Controller</i> ini bertugas untuk mengelola data proyek yang sedang dikerjakan
5	HouseItemController	<i>Controller</i> ini bertugas untuk mengelola data detail dari pengeluaran proyek
6	MandorController	<i>Controller</i> ini bertugas untuk mengelola data mandor
7	NoteController	<i>Controller</i> ini bertugas untuk mengelola data catatan
8	ProgressController	<i>Controller</i> ini bertugas untuk mengelola data <i>progress</i>
9	ReportController	<i>Controller</i> ini bertugas untuk mengelola dan menyediakan data laporan
10	RequestController	<i>Controller</i> ini bertugas untuk mengelola dan menyediakan data permintaan pengeluaran
11	WarehouseController	<i>Controller</i> ini bertugas untuk mengelola dan menyediakan data gudang

3. Membuat Aplikasi Android

Pada tahap ini pembuatan aplikasi menggunakan bahasa pemrograman java yang digunakan pada aplikasi android *native* dengan menggunakan *tools* Android Studio. Pembuatan aplikasi ini melibatkan *database* dan API sehingga pengerjaan dapat dikerjakan setelah pembuatan *database* dan API selesai dilakukan.

4.2 Testing

Aplikasi yang telah selesai dibuat, akan diintegrasikan dan diuji, untuk meyakinkan bahwa persyaratan aplikasi telah terpenuhi dan sesuai dengan kebutuhan PT MKDS dengan menggunakan metode blackbox testing. Pada tahapan ini aplikasi akan diuji oleh kontraktor perusahaan dan mandor yang terkait dengan proyek pada PT MKDS. *Testing* akan dilakukan berdasarkan pada *design testing* yang telah dibuat sebelumnya.

A. Testing Kontraktor

Berikut ini adalah *design testing* yang dibuat untuk tipe pengguna kontraktor pada aplikasi pengawasan kinerja mandor:

1. Halaman *Login*

Halaman *login* ini digunakan untuk melakukan validasi kepada pengguna. Untuk menggunakan halaman *login*, pengguna diwajibkan untuk mengisi *username* dan *password*. Penjelasan hasil uji coba halaman *login* dapat dilihat pada tabel 4.2 berikut ini:

Tabel 4.2 *Test Result* Halaman Login

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR01	Mengetahui respon sistem terhadap masukan data benar	1. Masukkan <i>email</i> 2. Masukkan <i>password</i> 3. Tekan <i>login</i>	Pengguna berhasil <i>login</i>	Gambar 4.1	Sukses
TR02	Mengetahui respon sistem terhadap masukan data salah	1. Masukkan <i>email</i> 2. Masukkan <i>password</i> 3. Tekan <i>login</i>	Pengguna gagal <i>login</i>	Gambar 4.2	Sukses
TR03	Mengetahui respon sistem terhadap kosongnya salah satu data	1. Masukkan <i>email</i> 2. Tekan <i>login</i>	Pengguna gagal <i>login</i>	Gambar 4.3	Sukses
TR04	Mengetahui respon sistem terhadap data kosong	Tekan <i>login</i>	Pengguna gagal <i>login</i>	Gambar 4.4	Sukses

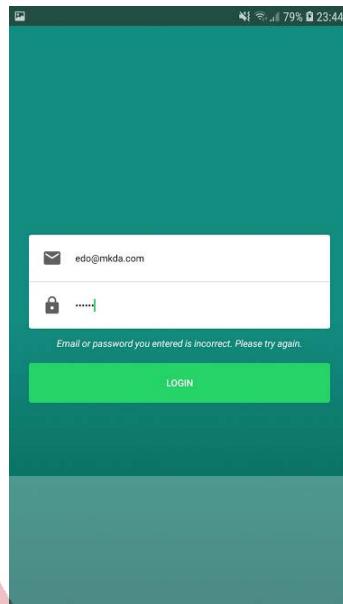
Gambar berikut ini merupakan proses *login* ke dalam aplikasi dengan memasukkan data *username* dan *password* dengan benar. Saat berhasil masuk ke dalam aplikasi, maka tampilan awal yang akan muncul adalah menu *houses*.



Gambar 0.1 Proses *Login* Sebagai Kontraktor

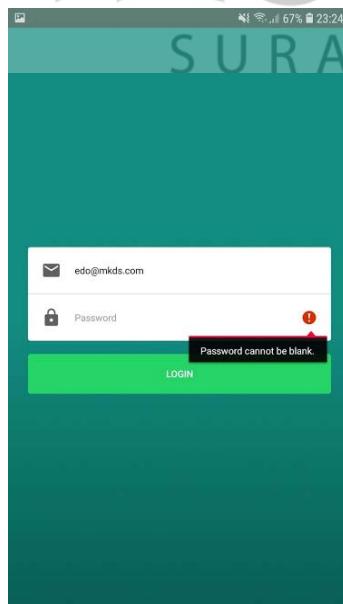
Pada gambar berikut ini, merupakan hasil dari masukan data yang tidak benar. Pada kasus ini, data yang salah adalah *password* sesuai dengan *test data* yang

dibuat sebelumnya. Jika data tidak sesuai maka akan muncul pesan “*Email or password you entered is incorrect. Please try again.*”



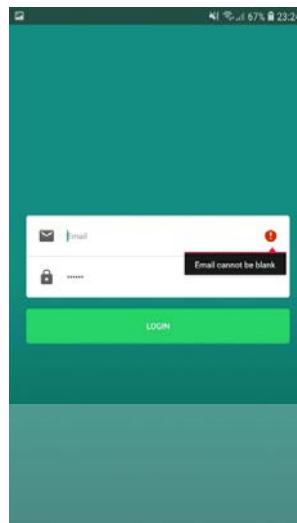
Gambar 0.2 *Login Dengan Data Salah*

Pada gambar berikut ini, merupakan hasil dari masukan data yang kosong. Pada kasus ini, data yang kosong adalah *password* sesuai dengan *test data* yang dibuat sebelumnya.



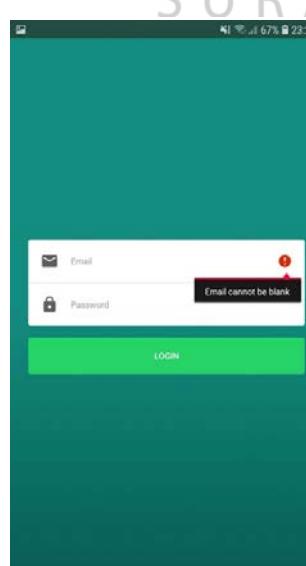
Gambar 0.3 *Login Dengan Password Kosong*

Pada gambar berikut ini, juga merupakan hasil dari masukan data yang kosong. Pada kasus ini, data yang kosong adalah *email* sesuai dengan *test data* yang dibuat sebelumnya.



Gambar 0.4 Login Dengan *Email* Kosong

Sedangkan pada gambar berikut ini, merupakan hasil dari melakukan *login* tanpa data *email* dan *password* karena sistem pada halaman ini memerlukan validasi email terlebih dahulu, maka kolom *email* yang dinyatakan harus diisi terlebih dahulu.



Gambar 0.5 *Login* Tanpa Masukan Data

2. Menu Utama

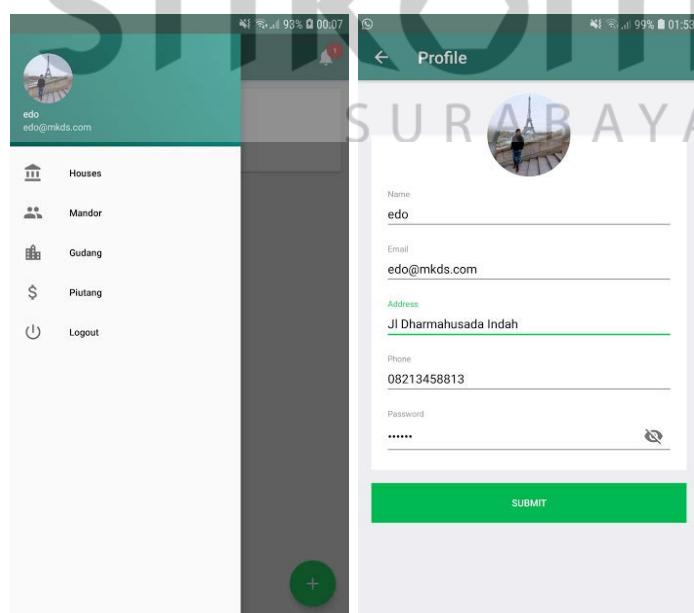
Menu utama ini digunakan untuk menunjukkan daftar menu yang tersedia pada aplikasi. Menu utama dapat ditemui pada pojok kiri atas halaman utama. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.3 berikut ini:

Tabel 4.3 *Test Result* Menu Utama

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR05	Mengetahui respon sistem saat menekan “icon user”	Tekan “icon user” pada menu utama	Halaman <i>profile</i> tampil	Gambar 4.6	Sukses
TR06	Mengetahui respon sistem saat menekan menu “Houses”	Tekan menu “Houses” pada menu utama	Halaman <i>houses</i> tampil	Gambar 4.7	Sukses
TR07	Mengetahui respon sistem saat menekan menu “Mandor”	Tekan menu “Mandor” pada menu utama	Halaman mandor tampil	Gambar 4.8	Sukses
TR08	Mengetahui respon sistem saat menekan menu “Gudang”	Tekan menu “Gudang” pada menu utama	Halaman gudang tampil	Gambar 4.9	Sukses

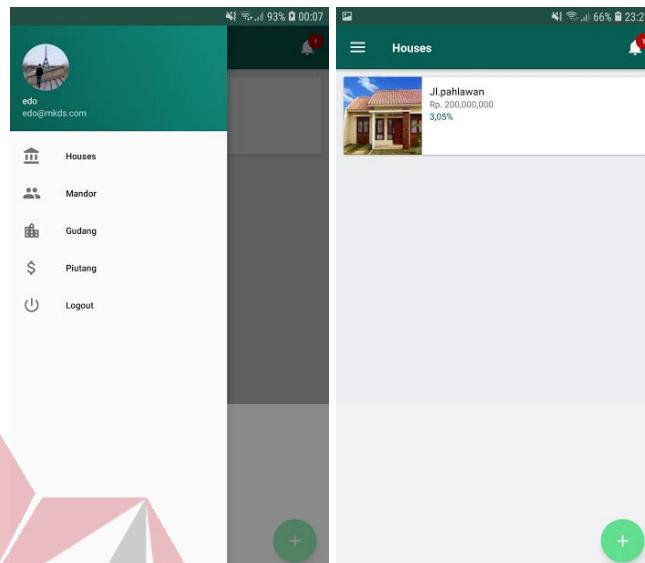
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR09	Mengetahui respon sistem saat menekan menu “Piutang”	Tekan menu “Piutang” pada menu utama	Halaman piutang tampil	Gambar 4.10	Sukses
TR10	Mengetahui respon sistem saat menekan menu “Logout”	Tekan menu “Logout” pada menu utama	Kembali ke halaman <i>login</i>	Gambar 4.11	Sukses

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan “icon profile”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



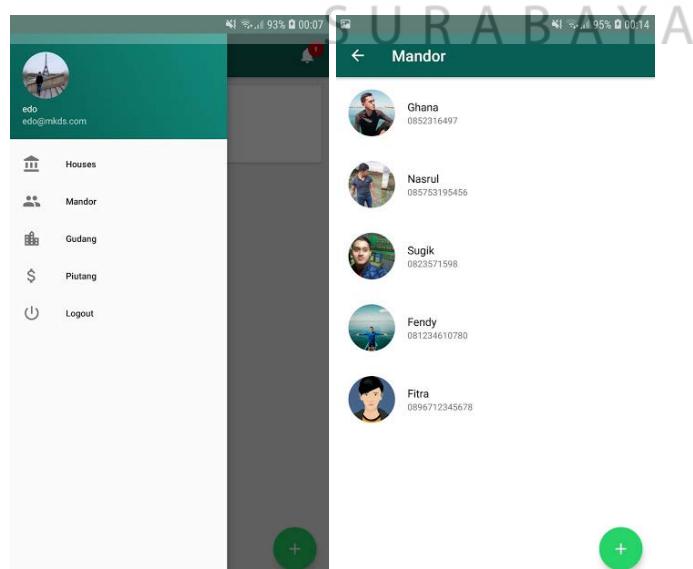
Gambar 0.6 Tampilan Menu Profil

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “houses”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



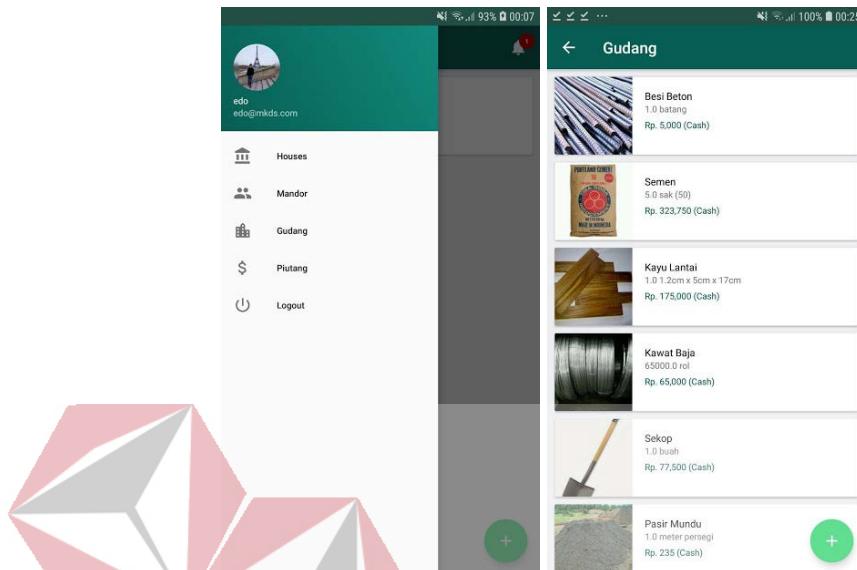
Gambar 0.7 Tampilan Menu *Houses*

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “mandor”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



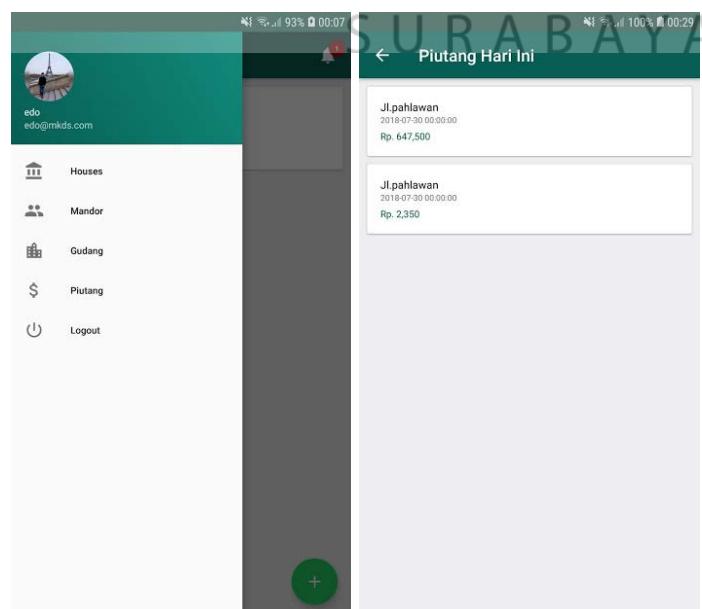
Gambar 0.8 Tampilan Menu *Mandor*

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “gudang”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



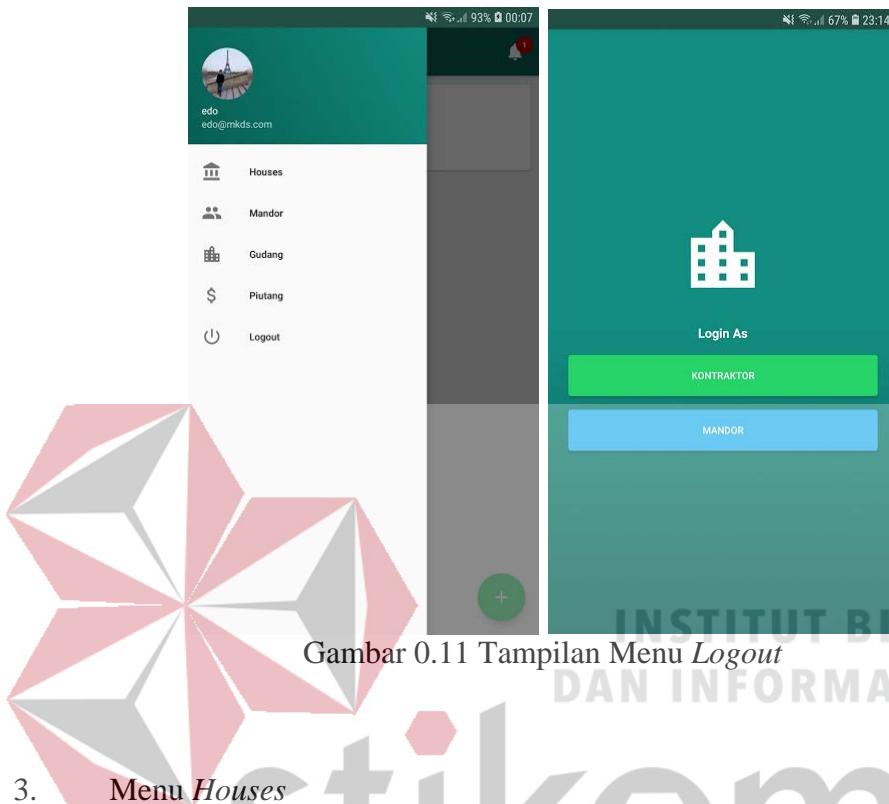
Gambar 0.9 Tampilan Menu Gudang

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “piutang”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.10 Tampilan Menu Piutang

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “*logout*”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



3. Menu *Houses*

Menu *houses* ini digunakan untuk mengolah data proyek yang sedang dikerjakan oleh kontraktor. Tidak ada batasan jumlah proyek yang dikelola, tetapi pada setiap proyeknya, kontraktor hanya dapat menugaskan satu mandor. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.4 berikut ini:

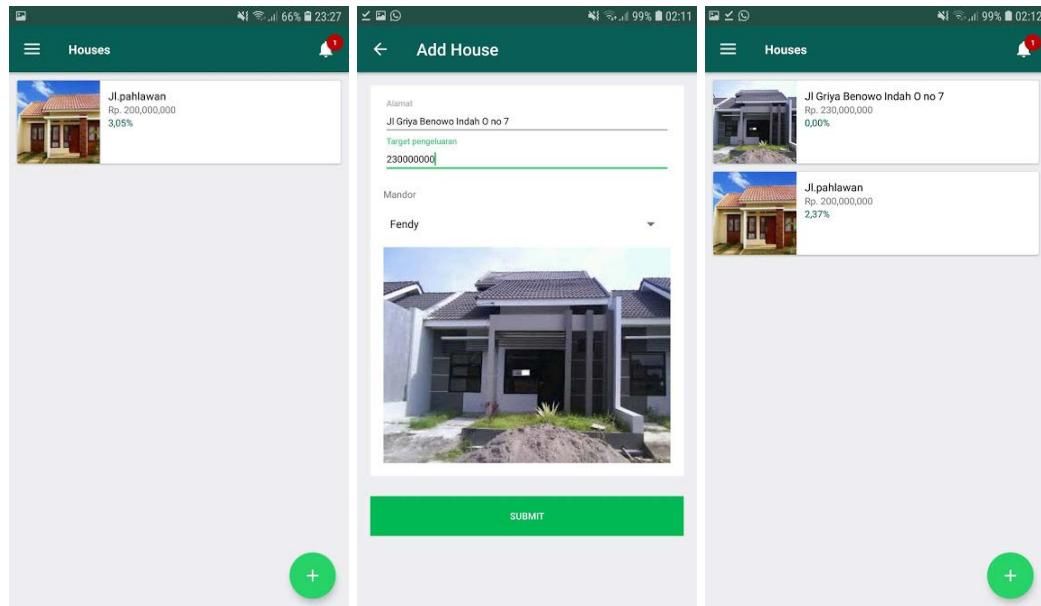
Tabel 4.4 *Test Result* Menu *Houses*

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR11	Mengetahui respon sistem	1. Pilih “icon plus” pada pojok	Data berhasil tersimpan	Gambar 4.12	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
	terhadap masukan data benar	<p>kanan bawah tampilan</p> <p>2. Masukkan Alamat</p> <p>3. Masukkan Target Pengeluaran</p> <p>4. Pilih Mandor</p> <p>5. Masukkan Gambar</p> <p>6. Tekan <i>submit</i></p>			
TR12	Mengetahui respon sistem terhadap kosongnya salah satu data	<p>1. Pilih “icon plus” pada pojok kanan bawah tampilan</p> <p>2. Masukkan Alamat</p> <p>3. Masukkan Target Pengeluaran</p> <p>4. Pilih Mandor</p> <p>5. Masukkan Gambar</p> <p>6. Tekan <i>submit</i></p>	Data gagal tersimpan	Gambar 4.13	Sukses
TR13	Mengetahui respon sistem terhadap data kosong	Tekan <i>login</i>	Data gagal tersimpan	Gambar 4.14	Sukses

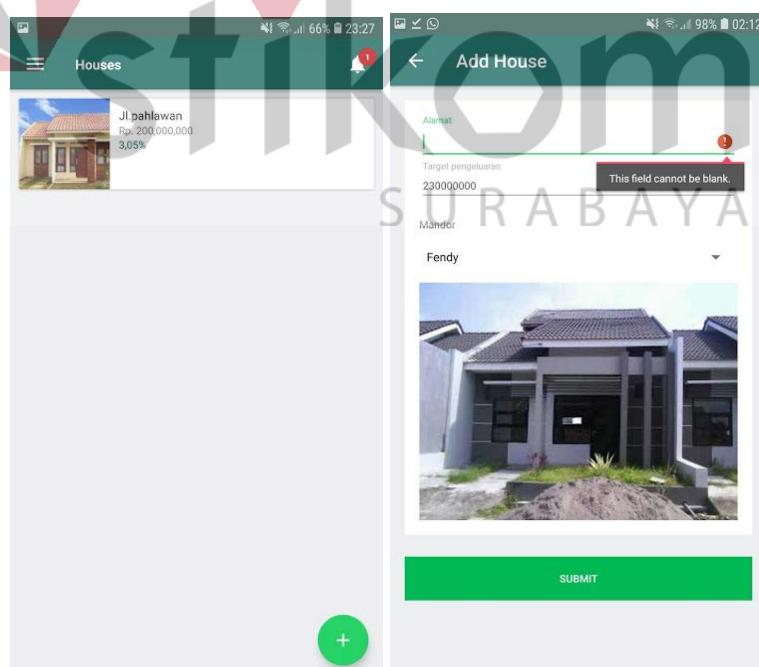
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR14	Mengetahui respon sistem terhadap data yang diubah	<ol style="list-style-type: none"> 1. Pilih data proyek 2. Pilih “icon menu” pada pojok kanan atas tampilan proyek 3. Pilih menu “Edit” 4. Ubah data masukan 5. Tekan “submit” 	Data berhasil tersimpan	Gambar 4.15	Sukses
TR15	Mengetahui respon sistem terhadap data yang dihapus	<ol style="list-style-type: none"> 1. Pilih data proyek 2. Pilih “icon menu” pada pojok kanan atas tampilan proyek 3. Pilih menu “Delete” 	Data berhasil terhapus	Gambar 4.16	Sukses

Pada gambar berikut ini, ditunjukkan proses penambahan data pada menu *houses*. Pada kasus ini akan digunakan data masukan benar. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



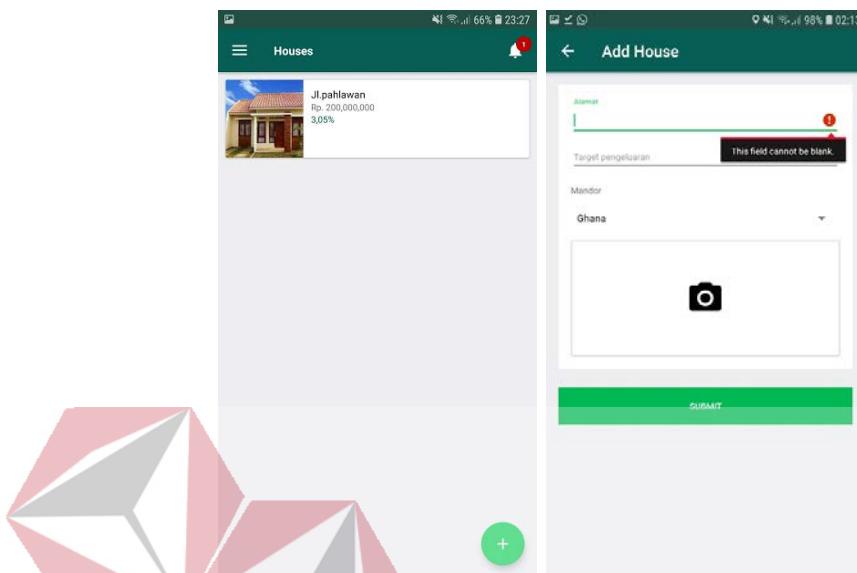
Gambar 0.12 Menambahkan Data *Houses*

Pada gambar berikut ini, ditunjukkan proses penambahan data pada menu *houses*. Pada kasus ini akan digunakan salah satu data masukan yang kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



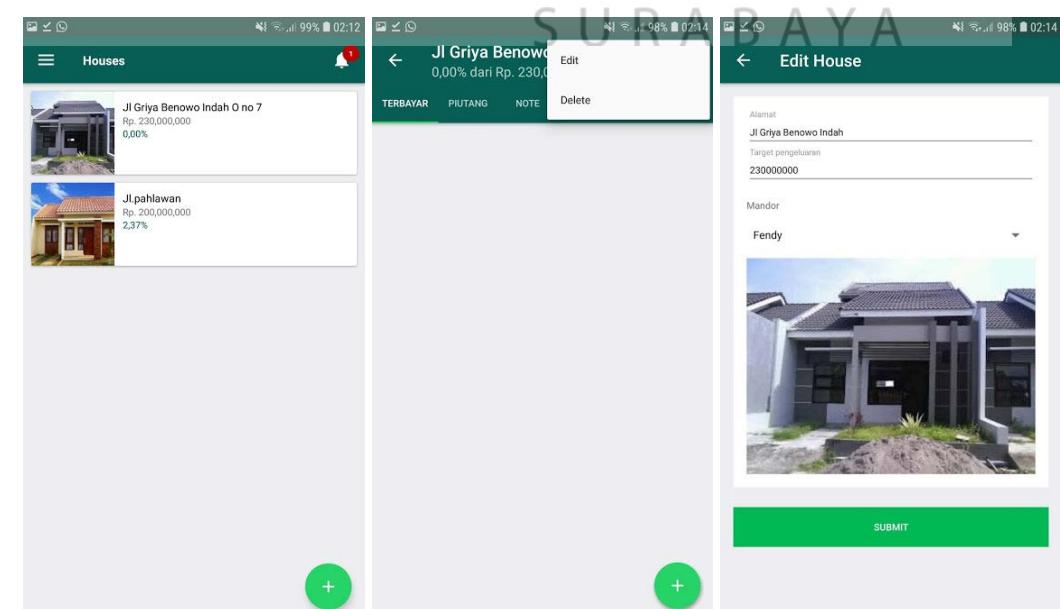
Gambar 0.13 Menambahkan Data *Houses* Tanpa Alamat

Pada gambar berikut ini, ditunjukkan proses penambahan data pada menu *houses*. Pada kasus ini akan digunakan data masukan yang kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



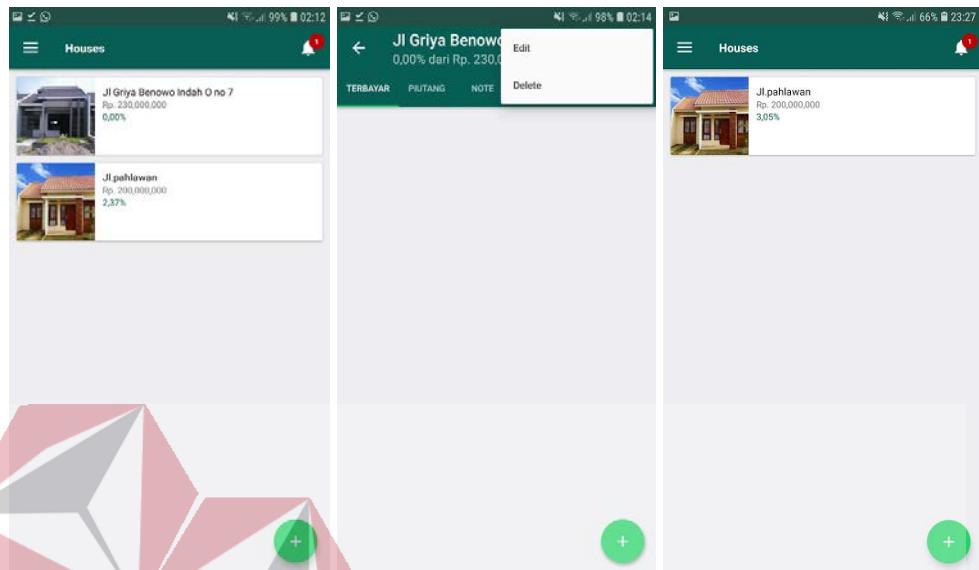
Gambar 0.14 Menambahkan Data *Houses* Kosong

Pada gambar berikut ini, ditunjukkan proses pengubahan data pada menu *houses*. Pada kasus ini akan digunakan salah satu data proyek yang telah tersedia. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.15 Mengubah Data *Houses*

Pada gambar berikut ini, ditunjukkan proses penghapusan data pada menu *houses*. Pada kasus ini akan digunakan salah satu data proyek yang telah tersedia. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.16 Menghapus Data *Houses*

4. Menu Terbayar

Menu terbayar ini digunakan untuk mengolah data pengeluaran proyek yang sedang dikerjakan oleh kontraktor dan mandor. Pada menu ini diperuntukkan untuk pengeluaran dengan metode tunai/*cash*, dan pengambilan barang gudang.

Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.5 berikut ini:

Tabel 4.5 *Test Result* Menu Terbayar

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR16	Mengetahui respon sistem terhadap	1. Pilih “icon plus” pada pojok	Data berhasil disimpan	Gambar 4.17	Sukses

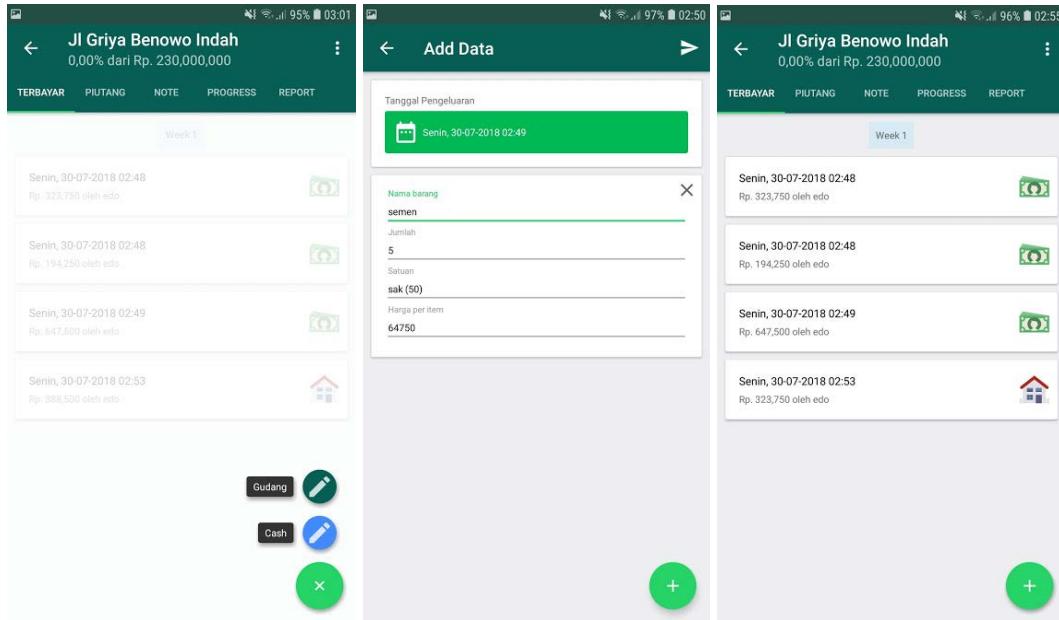
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
	masukan data benar	<p>kanan bawah tampilan</p> <p>2. Pilih menu “Cash”</p> <p>3. Pilih tanggal pengeluaran</p> <p>4. Masukkan nama barang</p> <p>5. Masukkan jumlah</p> <p>6. Masukkan satuan</p> <p>7. Masukkan harga tiap satuan</p> <p>8. Tekan “icon send” pada pojok kanan atas</p>			
TR17	Mengetahui respon sistem terhadap kosongnya salah satu data	<p>1. Pilih “icon plus” pada pojok kanan bawah tampilan</p> <p>2. Pilih menu “Cash”</p> <p>3. Masukkan nama barang</p> <p>4. Masukkan jumlah</p>	Data gagal disimpan	Gambar 4.18	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
		5. Masukkan satuan 6. Masukkan harga tiap satuan 7. Tekan “ <i>icon send</i> ” pada pojok kanan atas			
TR18	Mengetahui respon sistem terhadap data kosong	1. Pilih “ <i>icon plus</i> ” pada pojok kanan bawah tampilan 2. Pilih menu “ <i>Cash</i> ” 3. Tekan “ <i>icon send</i> ” pada pojok kanan atas	Data gagal disimpan	Gambar 4.19	Sukses
TR19	Mengetahui respon sistem terhadap masukan data benar	1. Pilih “ <i>icon plus</i> ” pada pojok kanan bawah tampilan 2. Pilih menu “ <i>Gudang</i> ” 3. Pilih tanggal pengeluaran 4. Pilih nama barang 5. Masukkan jumlah	Data berhasil disimpan	Gambar 4.20	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
		6. Masukkan satuan 7. Masukkan harga tiap satuan 8. Tekan “ <i>icon send</i> ” pada pojok kanan atas			
TR20	Mengetahui respon sistem terhadap kosongnya salah satu data	1. Pilih “ <i>icon plus</i> ” pada pojok kanan bawah tampilan 2. Pilih menu “Gudang” 3. Pilih nama barang 4. Masukkan jumlah 5. Masukkan satuan 6. Masukkan harga tiap satuan 7. Tekan “ <i>icon send</i> ” pada pojok kanan atas	Data gagal disimpan	Gambar 4.21	Sukses
TR21	Mengetahui respon sistem terhadap data kosong	1. Pilih “ <i>icon plus</i> ” pada pojok kanan bawah tampilan	Data gagal disimpan	Gambar 4.22	Sukses

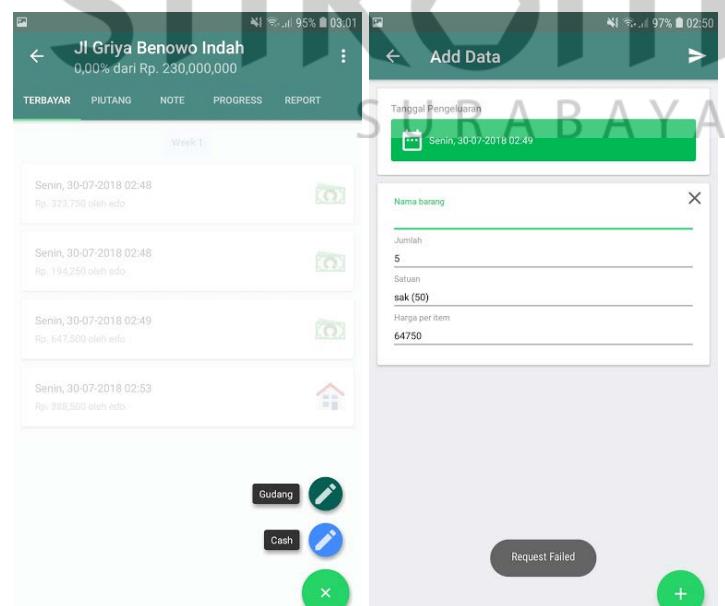
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
		2. Pilih menu “Gudang” 3. Tekan “icon send” pada pojok kanan atas			
TR22	Mengetahui respon sistem saat melakukan edit pengeluaran	1. Pilih salah satu data pada tampilan terbayar 2. Pilih menu “edit” pada pojok kanan atas tampilan 3. Ubah jumlah data 4. Pilih menu “save” pada pojok kanan atas tampilan	Data berhasil disimpan	Gambar 4.23	Sukses

Pada gambar berikut ini, ditunjukkan proses penambahan data *cash* pada menu terbayar. Pada kasus ini akan digunakan *data test* yang telah disediakan sebelumnya. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



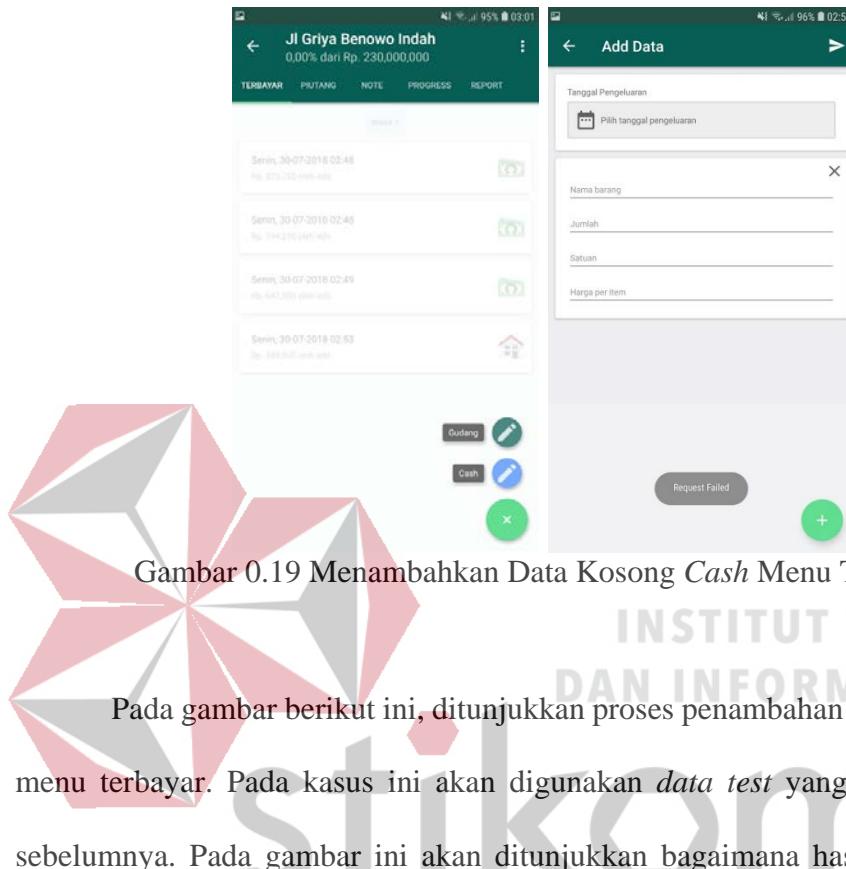
Gambar 0.17 Menambahkan Data *Cash* Menu Terbayar

Pada gambar berikut ini, ditunjukkan proses penambahan data *cash* pada menu terbayar. Pada kasus ini akan digunakan *data test* dengan mengosongkan kolom nama barang. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

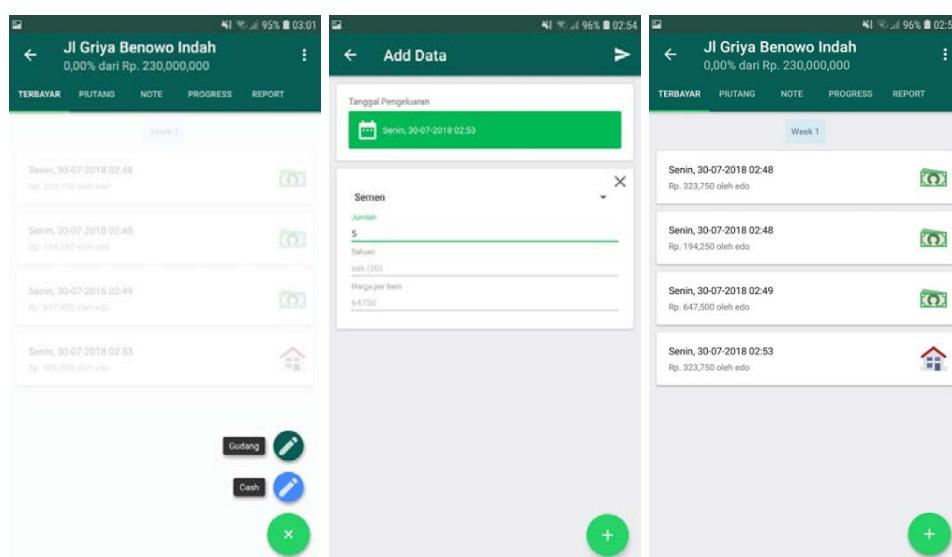


Gambar 0.18 Menambahkan Data *Cash* Menu Terbayar Tanpa Nama Barang

Pada gambar berikut ini, ditunjukkan proses penambahan data *cash* pada menu terbayar. Pada kasus ini akan digunakan data kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

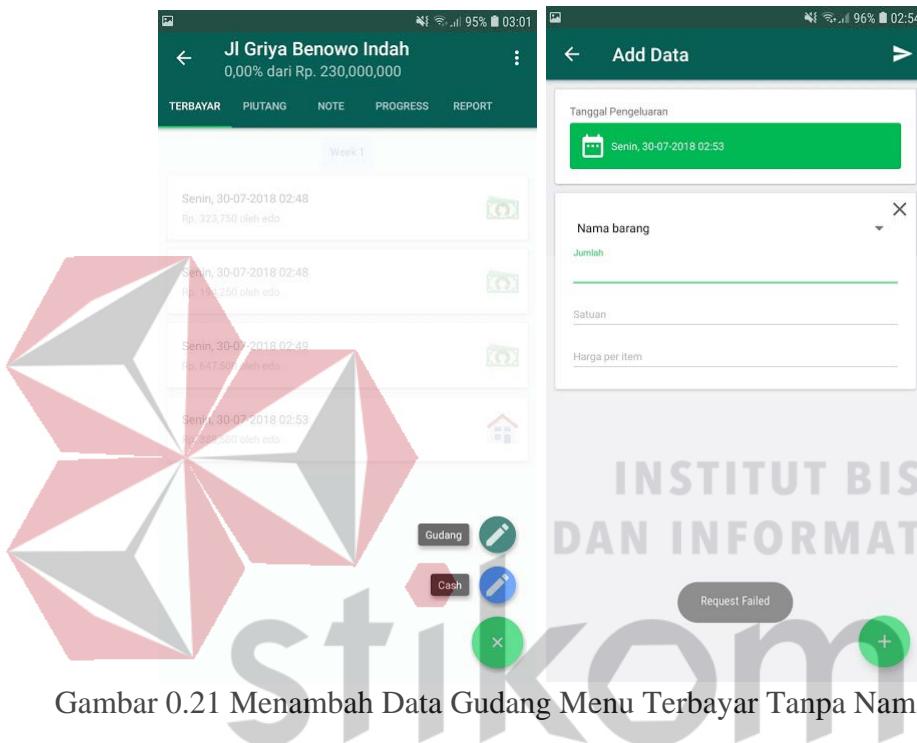


Pada gambar berikut ini, ditunjukkan proses penambahan data gudang pada menu terbayar. Pada kasus ini akan digunakan *data test* yang telah disediakan sebelumnya. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



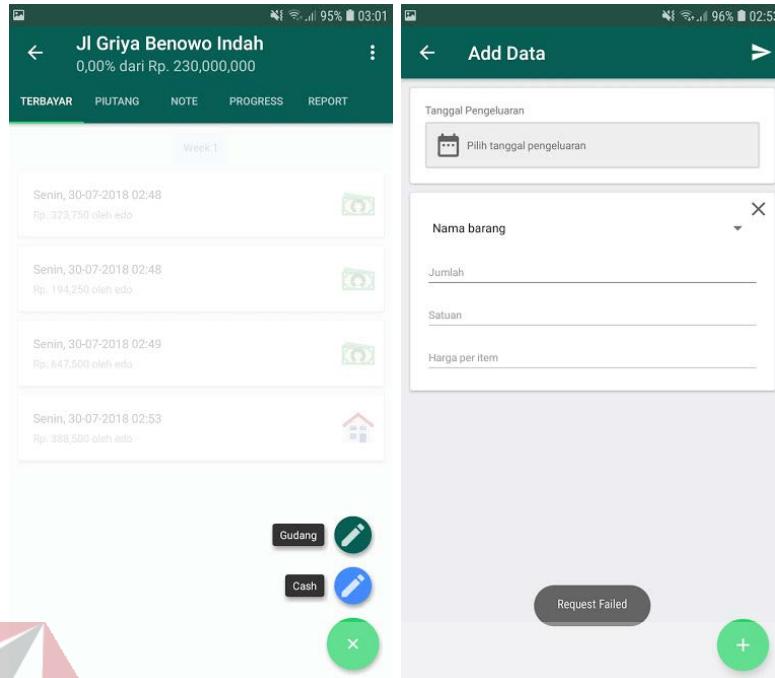
Gambar 0.20 Menambahkan Data Gudang Menu Terbayar

Pada gambar berikut ini, ditunjukkan proses penambahan data gudang pada menu terbayar. Pada kasus ini akan digunakan *data test* dengan mengosongkan kolom nama barang yang seharusnya dipilih pada tampilan ini. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



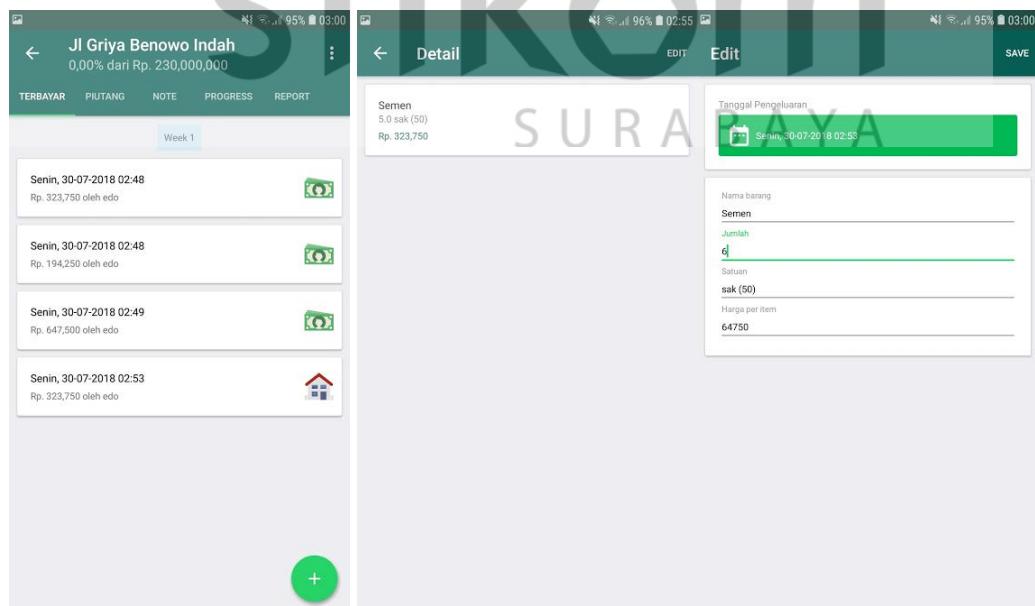
Gambar 0.21 Menambah Data Gudang Menu Terbayar Tanpa Nama Barang

Pada gambar berikut ini, ditunjukkan proses penambahan data gudang pada menu terbayar. Pada kasus ini akan digunakan data kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.22 Menambahkan Data Kosong Gudang Menu Terbayar

Pada gambar berikut ini, ditunjukkan tampilan *detil* data pada menu terbayar. Pada kasus ini akan dipilih salah satu data yang terdapat pada aplikasi. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.23 Mengubah Data Terbayar

5. Menu Piutang

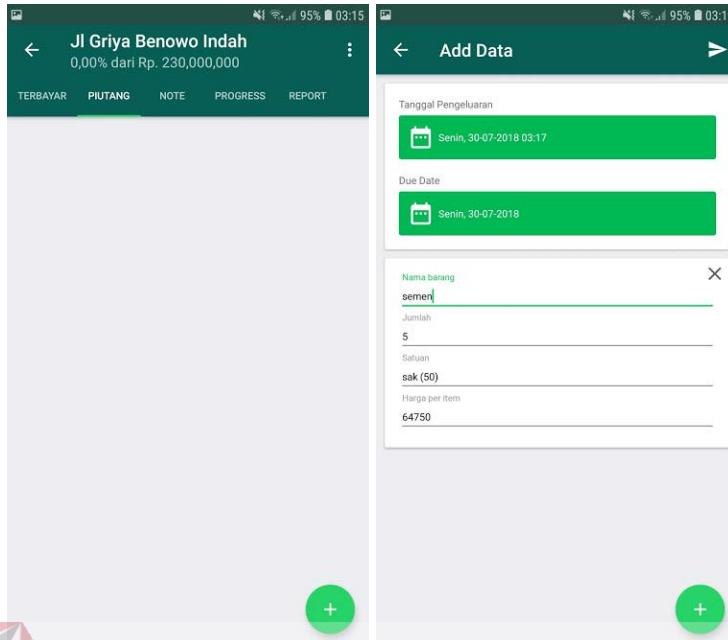
Menu Piutang ini digunakan untuk mengolah data pengeluaran proyek dengan cara piutang yang sedang dikerjakan oleh kontraktor dan mandor. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.6 berikut ini:

Tabel 4.6 *Test Result* Menu Piutang

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR23	Mengetahui respon sistem terhadap masukan data benar	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih tanggal pengeluaran 3. Pilih <i>due date</i> 4. Masukkan nama barang 5. Masukkan jumlah 6. Masukkan satuan 7. Masukkan harga tiap satuan 8. Tekan “icon send” pada pojok kanan atas 	Data berhasil disimpan	Gambar 4.24	Sukses
TR24	Mengetahui respon sistem terhadap kosongnya	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih <i>due date</i> 3. Masukkan nama barang 	Data gagal disimpan	Gambar 4.25	Sukses

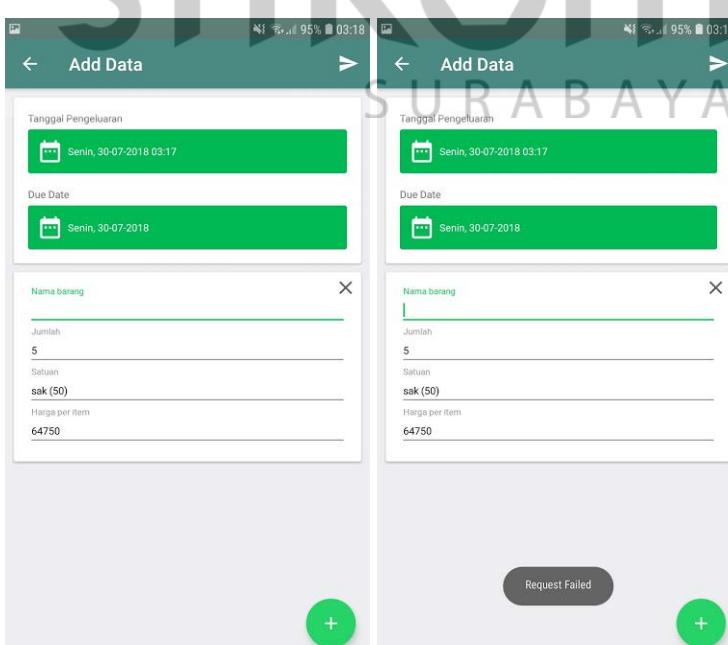
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
	salah satu data	4. Masukkan jumlah 5. Masukkan satuan 6. Masukkan harga tiap satuan 7. Tekan “icon send” pada pojok kanan atas			
TR25	Mengetahui respon sistem terhadap data kosong	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Tekan “icon send” pada pojok kanan atas	Data gagal disimpan	Gambar 4.26	Sukses
TR26	Mengetahui respon sistem terhadap perubahan data	1. Pilih salah satu data pada menu piutang 2. Pilih menu “edit” pada pojok kanan atas 3. Ubah data 4. Pilih menu “save” pada pojok kanan atas	Data berhasil disimpan	Gambar 4.27	Sukses

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu piutang. Pada kasus ini akan ditambahkan data piutang yang benar. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



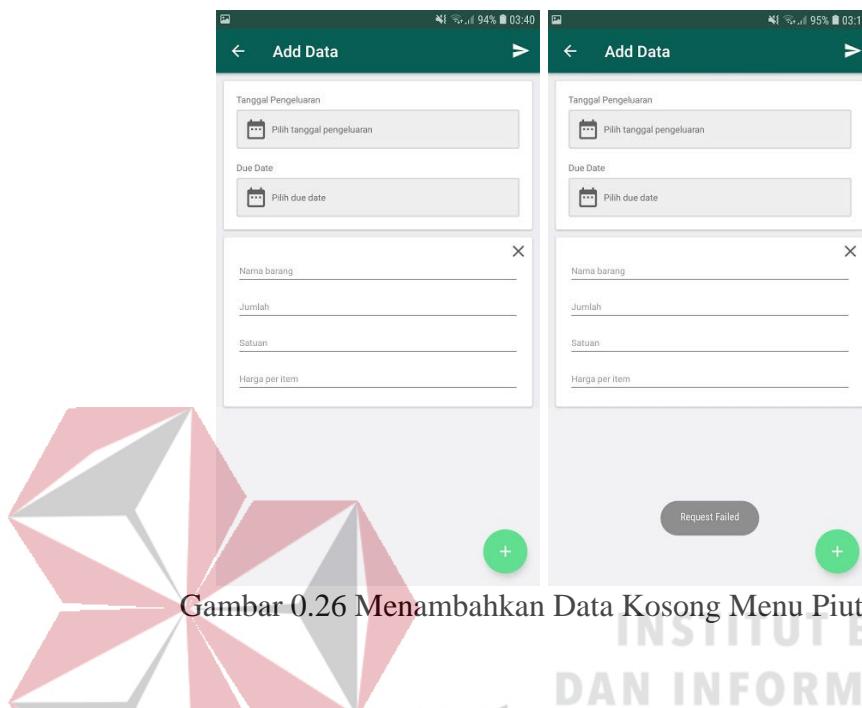
Gambar 0.24 Menambahkan Data Menu Piutang

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu piutang. Pada kasus ini akan ditambahkan data kosong pada kolom nama barang. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



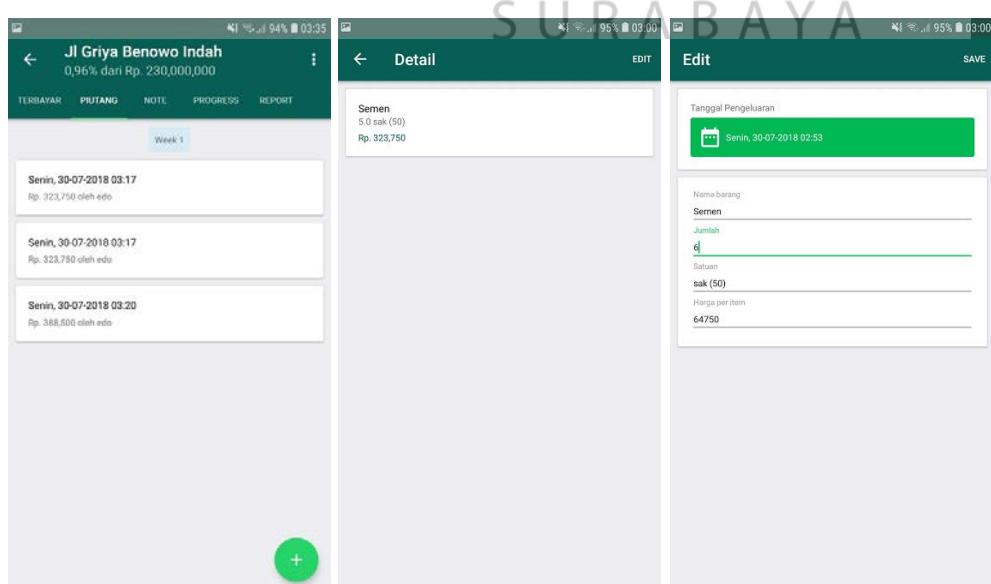
Gambar 0.25 Menambahkan Data Menu Piutang Tanpa Nama Barang

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu piutang. Pada kasus ini akan ditambahkan data kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.26 Menambahkan Data Kosong Menu Piutang

Pada gambar berikut ini, akan dijelaskan mengenai pengubahan data pada menu piutang. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.27 Mengubah Data Menu Piutang

6. Menu Note

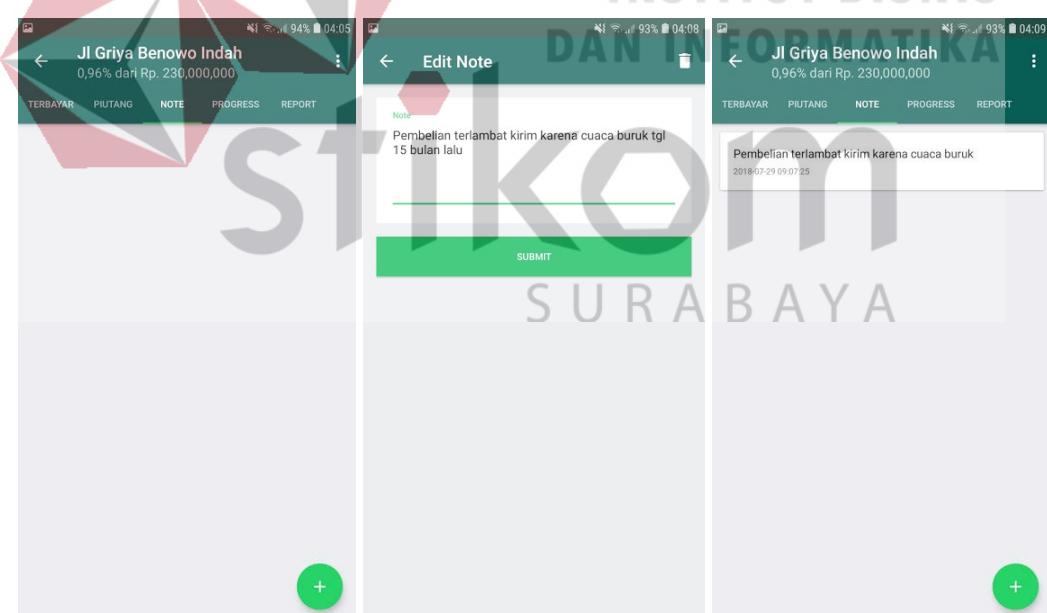
Menu *note* ini digunakan untuk mengolah data catatan proyek yang sedang dikerjakan oleh kontraktor dan mandor. Catatan ini dipergunakan sebagai pengingat hal penting yang perlu diketahui oleh kontraktor dan mandor. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.7 berikut ini:

Tabel 4.7 *Test Result* Menu Note

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR27	Mengetahui respon sistem terhadap masukan data benar	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Masukkan note 3. Tekan “submit”	Data berhasil disimpan	Gambar 4.28	Sukses
TR28	Mengetahui respon sistem terhadap pengubahan data	1. Pilih salah satu data pada menu note 2. Ubah note 3. Tekan “submit”	Data berhasil diubah	Gambar 4.29	Sukses
TR29	Mengetahui respon sistem terhadap data kosong	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Tekan “submit”	Data gagal disimpan	Gambar 4.30	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR30	Mengetahui respon sistem terhadap penghapusan data	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “icon trash” pada pojok kanan atas tampilan	Data berhasil terhapus	Gambar 4.31	Sukses

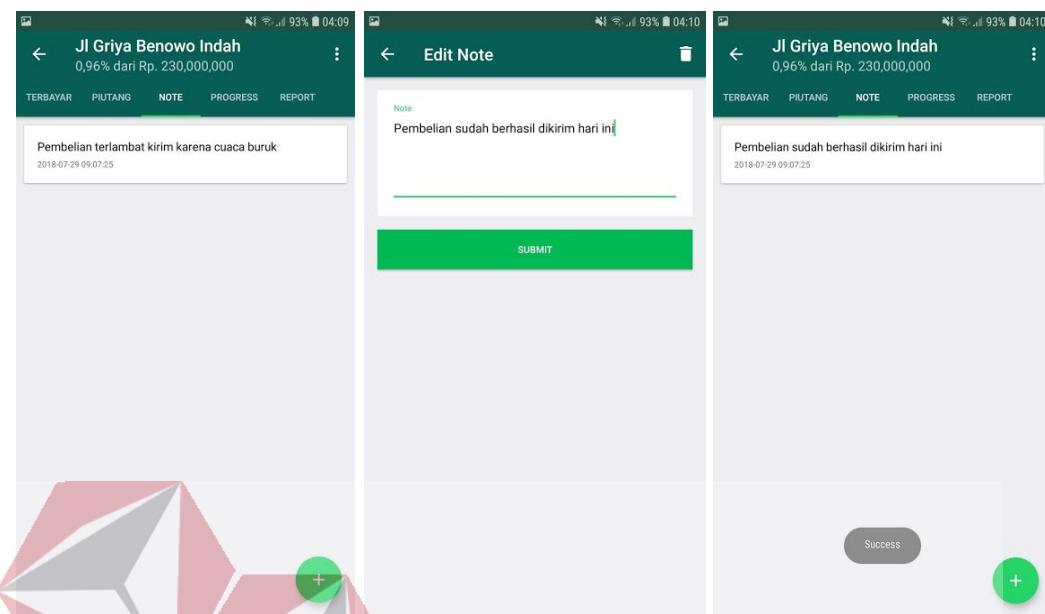
Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *note*. Pada kasus ini akan digunakan data benar yang telah dipersiapkan sebelumnya pada *test data*. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.28 Menambahkan Data Menu Note

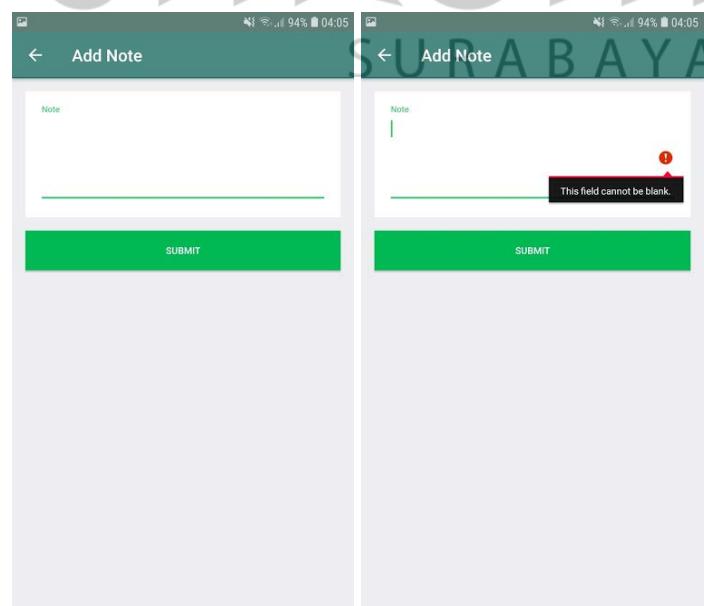
Pada gambar berikut ini, akan dijelaskan mengenai pengubahan data pada menu *note*. Pada kasus ini akan digunakan data benar yang telah dipersiapkan

sebelumnya pada *test data*. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



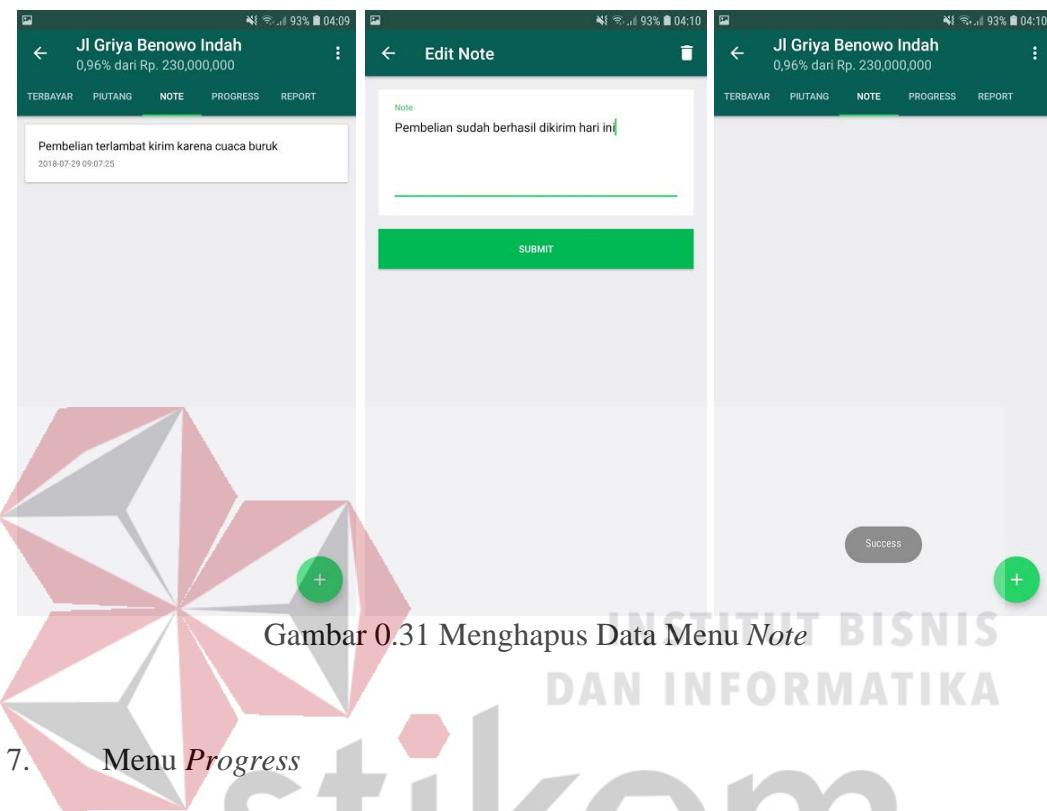
Gambar 0.29 Mengubah Data Menu Note

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *note*. Pada kasus ini akan digunakan data kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.30 Menambahkan Data Kosong Menu Note

Pada gambar berikut ini, akan dijelaskan mengenai penghapusan data pada menu *note* yang terdapat pada aplikasi. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



7. Menu Progress

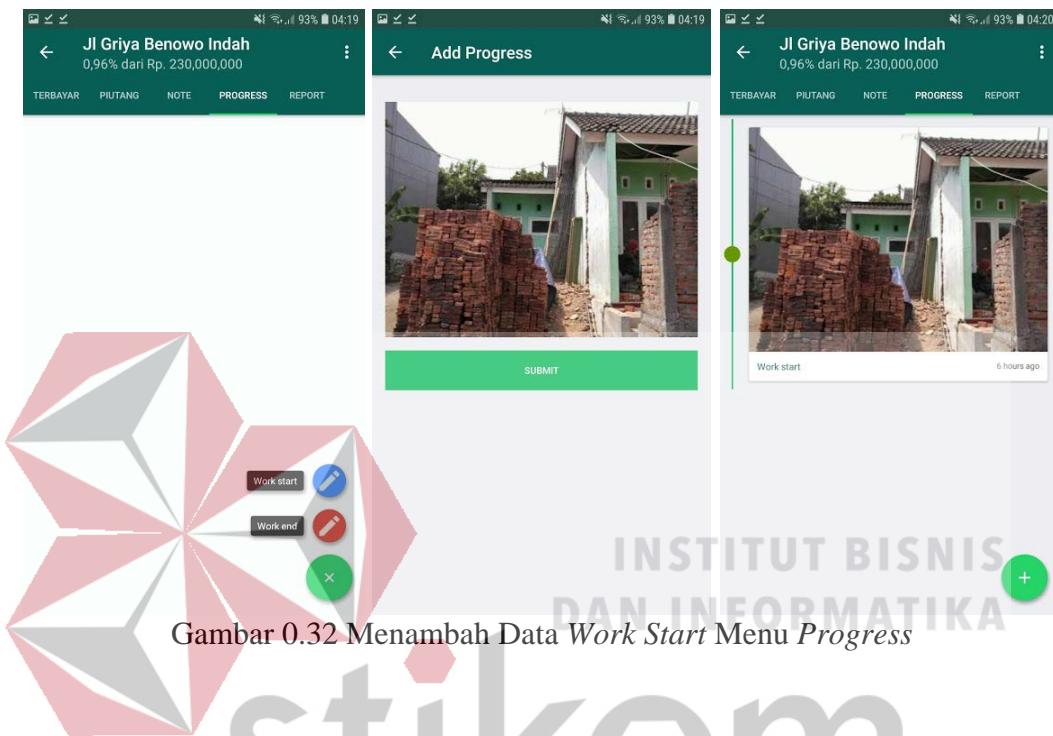
Menu *progress* ini digunakan untuk mengolah data kemajuan proyek yang sedang dikerjakan oleh kontraktor dan mandor. Menu ini dibutuhkan agar kontraktor selalu mengetahui kondisi terakhir pekerjaan proyek. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.8 berikut ini:

Tabel 4.8 *Test Result* Menu *Progress*

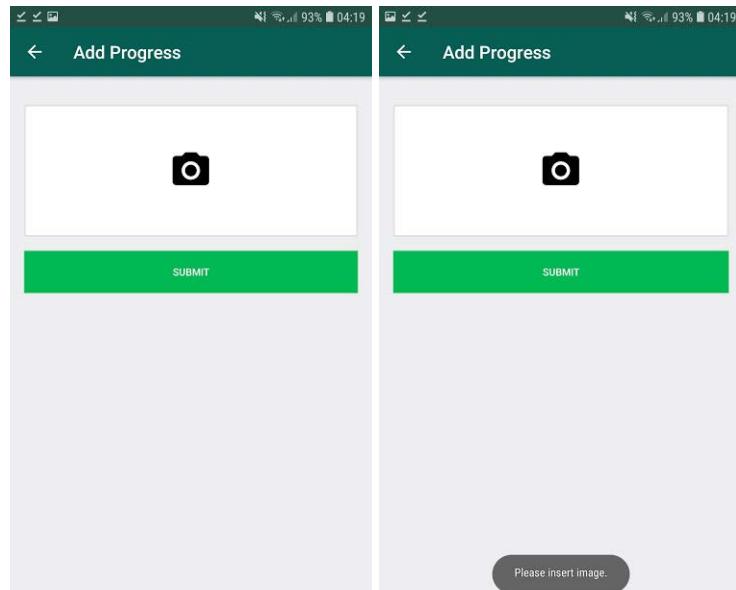
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR31	Mengetahui respon sistem	1. Pilih “icon plus” pada pojok	Data berhasil disimpan	Gambar 3.32	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
	terhadap masukan data benar	kanan bawah tampilan 2. Pilih “work start” 3. Pilih gambar 4. Tekan “submit”			
TR32	Mengetahui respon sistem terhadap data kosong	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “work start” 3. Tekan “submit”	Data gagal disimpan	Gambar 3.33	Sukses
TR33	Mengetahui respon sistem terhadap masukan data benar	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “work end” 3. Pilih gambar 4. Tekan “submit”	Data berhasil disimpan	Gambar 3.34	Sukses
TR34	Mengetahui respon sistem terhadap data kosong	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “work end” 3. Tekan “submit”	Data gagal disimpan	Gambar 3.35	Sukses

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *progress* yang terdapat pada aplikasi. Pada kasus ini akan ditambahkan gambar awal pekerjaan. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

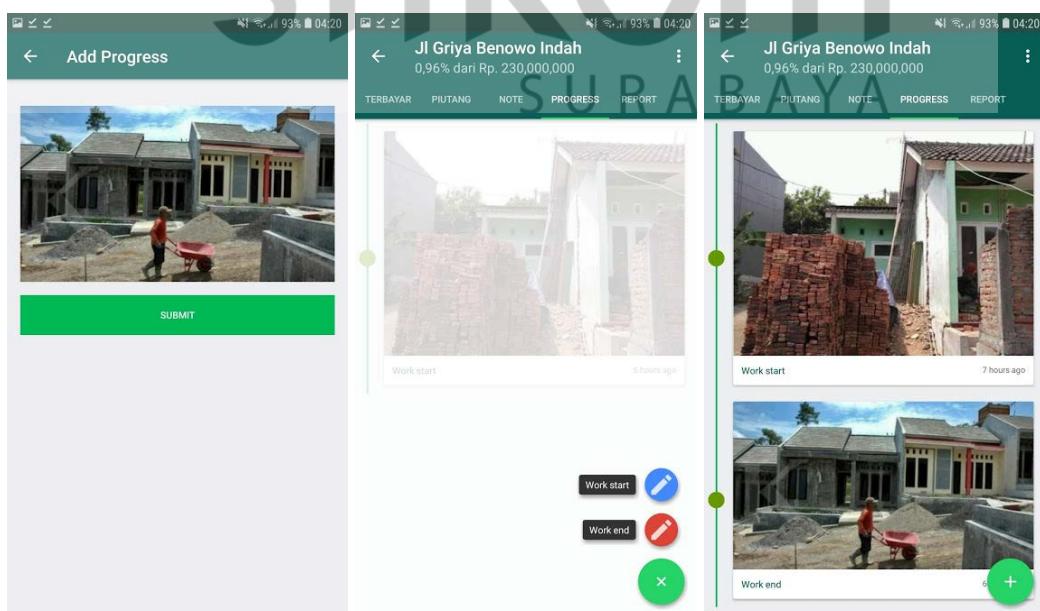


Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *progress* yang terdapat pada aplikasi. Pada kasus ini akan ditambahkan gambar kosong saat awal pekerjaan. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



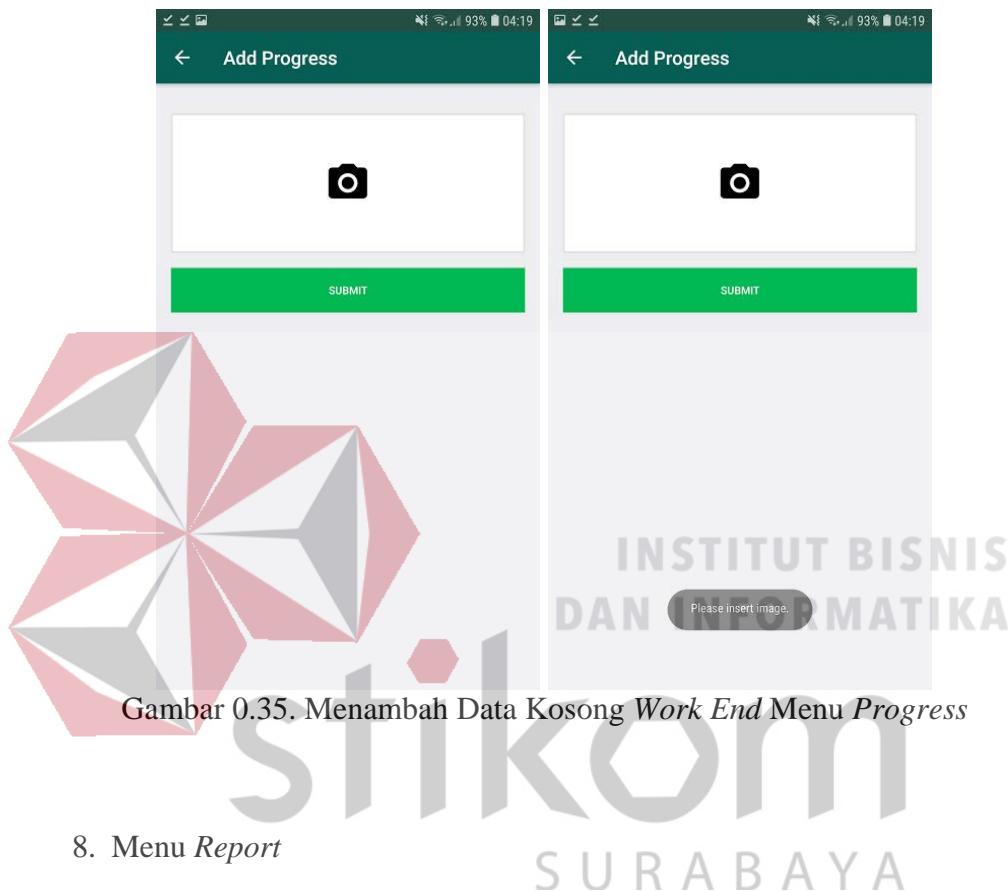
Gambar 0.33 Menambah Data Kosong Work Start Menu Progress

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *progress* yang terdapat pada aplikasi. Pada kasus ini akan ditambahkan gambar akhir pekerjaan. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.34 Menambah Data Work End Menu Progress

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *progress* yang terdapat pada aplikasi. Pada kasus ini akan ditambahkan gambar kosong saat akhir pekerjaan. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



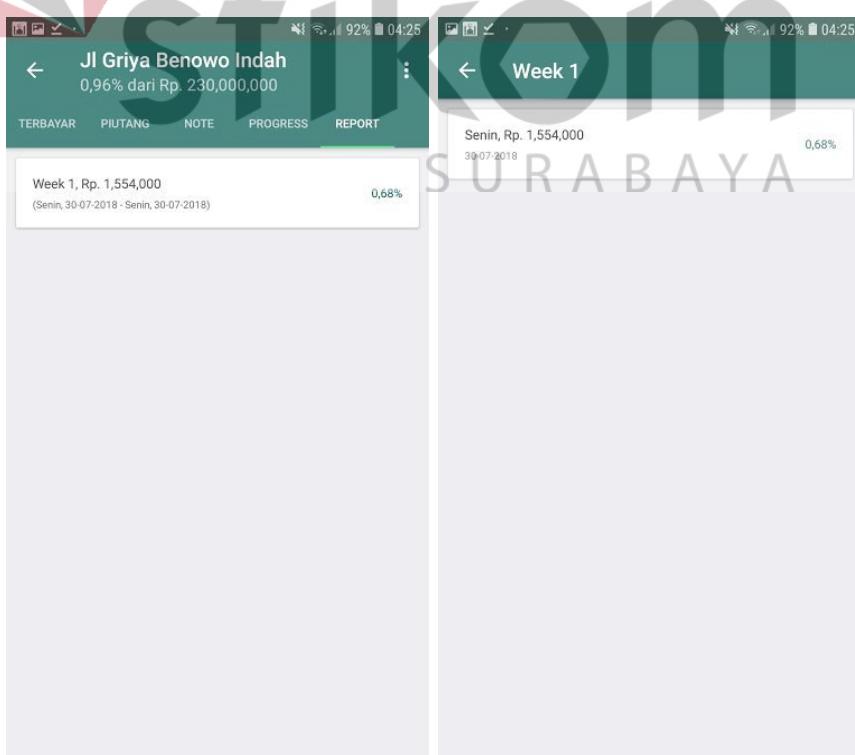
8. Menu *Report*

Menu *report* ini digunakan untuk melihat laporan pengeluaran dari awal mulai pekerjaan hingga kegiatan terakhir proyek yang sedang dikerjakan. Pada menu ini data akan disajikan pada tahap mingguan dan harian. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.9 berikut ini:

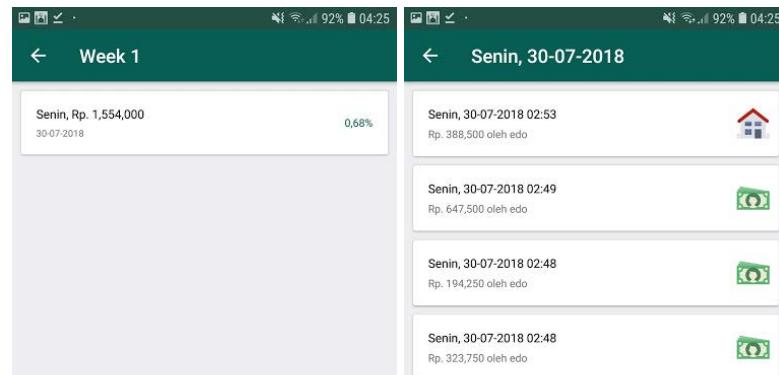
Tabel 4.9 *Test Result Menu Report*

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR35	Mengetahui respon sistem jika data mingguan dipilih	Pilih salah satu tampilan pada data mingguan	Data berhasil tampil	Gambar 4.36	Sukses
TR34	Mengetahui respon sistem jika data harian dipilih	Pilih salah satu tampilan pada data harian	Data berhasil tampil	Gambar 4.37	Sukses

Pada gambar berikut ini, akan dijelaskan mengenai laporan pengeluaran mingguan. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

Gambar 0.36 Melihat *Report* Mingguan

Pada gambar berikut ini, akan dijelaskan mengenai laporan pengeluaran harian. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.37 Melihat Report Harian

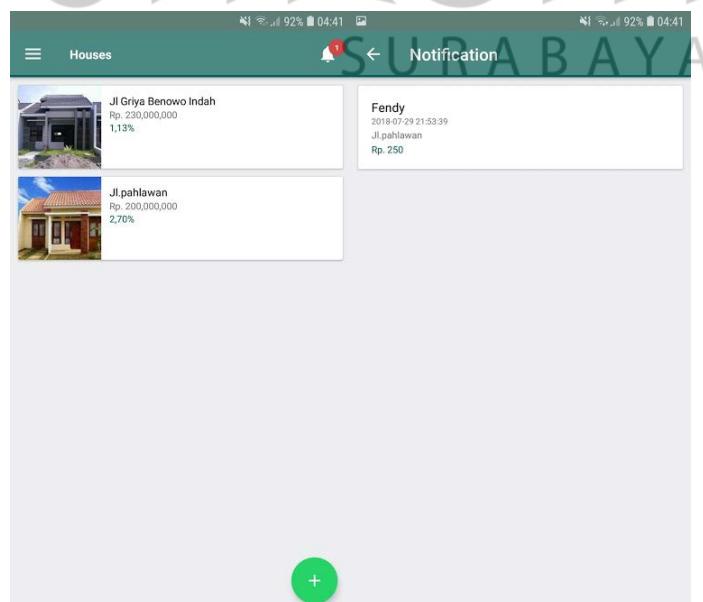
9. Menu Notifikasi

Menu notifikasi ini digunakan untuk mengetahui bahwa mendor sedang melakukan perizinan permohonan pembelian. Kontraktor akan dapat melakukan persetujuan atau penolakan terhadap permintaan tersebut. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.10 berikut ini:

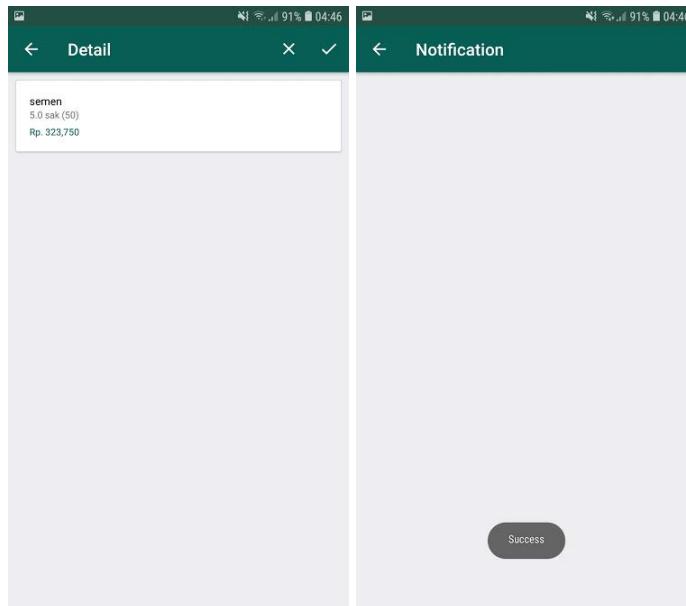
Tabel 4.10 Test Result Menu Notifikasi

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR35	Mengetahui respon sistem jika permohonan pembelian disetujui	1. Pilih salah satu data permintaan 2. Pilih “icon check”	Data permintaan berhasil diterima	Gambar 4.38 dan Gambar 4.39	Sukses
TR36	Mengetahui respon sistem jika permohonan pembelian ditolak	1. Pilih salah satu data permintaan 2. Pilih “icon cross”	Data permintaan berhasil ditolak	Gambar 4.40 dan gambar 4.41	Sukses

Pada gambar berikut ini, akan dijelaskan mengenai penerimaan permohonan pembelian. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

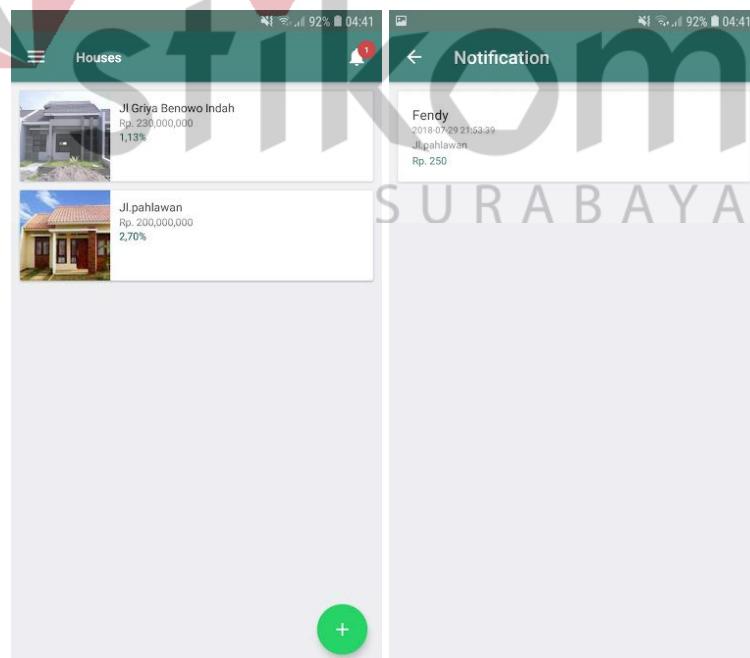


Gambar 0.38 Data Penerimaan Permohonan Menu Notifikasi

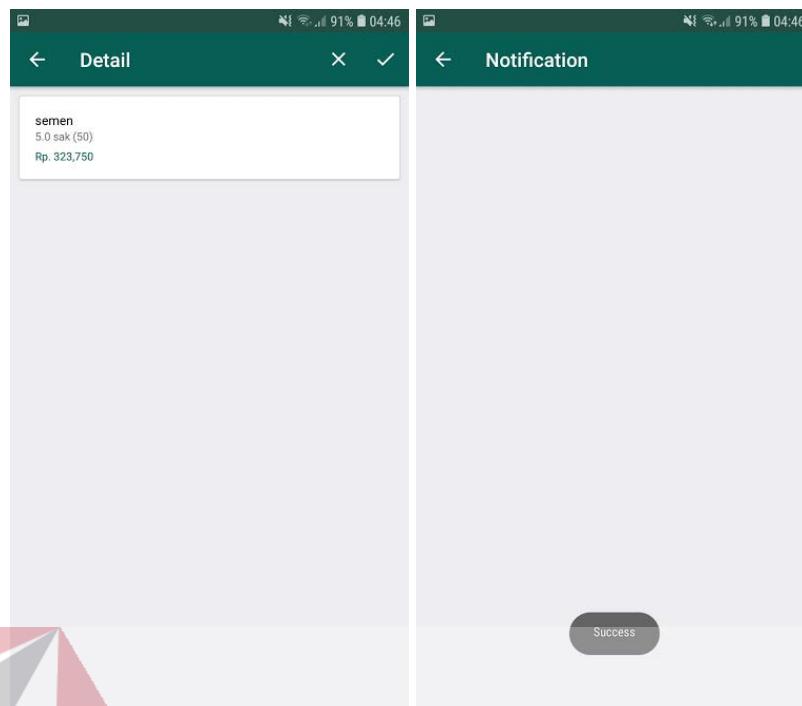


Gambar 0.39 Penerimaan Permohonan Menu Notifikasi

Pada gambar berikut ini, akan dijelaskan mengenai penolakan permohonan pembelian. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.40 Penolakan Permohonan Menu Notifikasi Data



Gambar 0.41 Penolakan Permohonan Menu Notifikasi

10. Menu Mandor

Menu mandor ini digunakan untuk mengelola data mandor yang akan mengerjakan proyek. Penambahan data mandor tidak dibatasi, tetapi pada pelaksanaan proyek, hanya akan ada satu mandor yang akan menjadi penanggung jawab proyek. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.11 berikut ini:

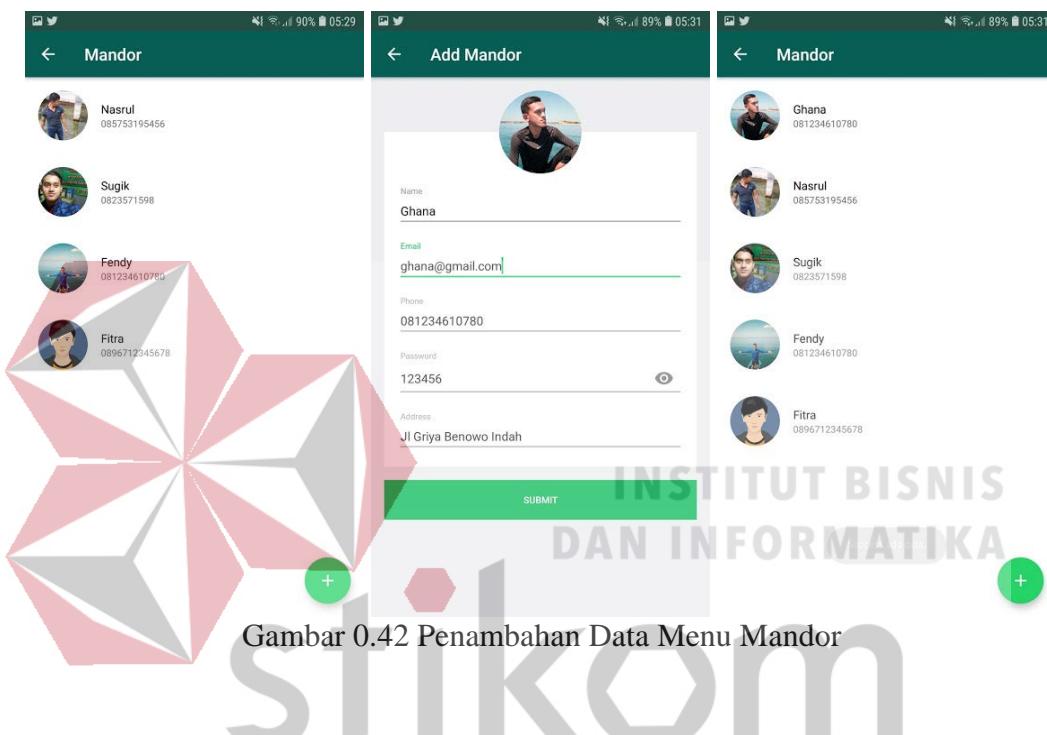
Tabel 4.11 *Test Result* Menu Mandor

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR37	Mengetahui respon system	1. Pilih “icon plus” pada pojok kanan bawah tampilan	Data berhasil disimpan	Gambar 4.42	Sukses

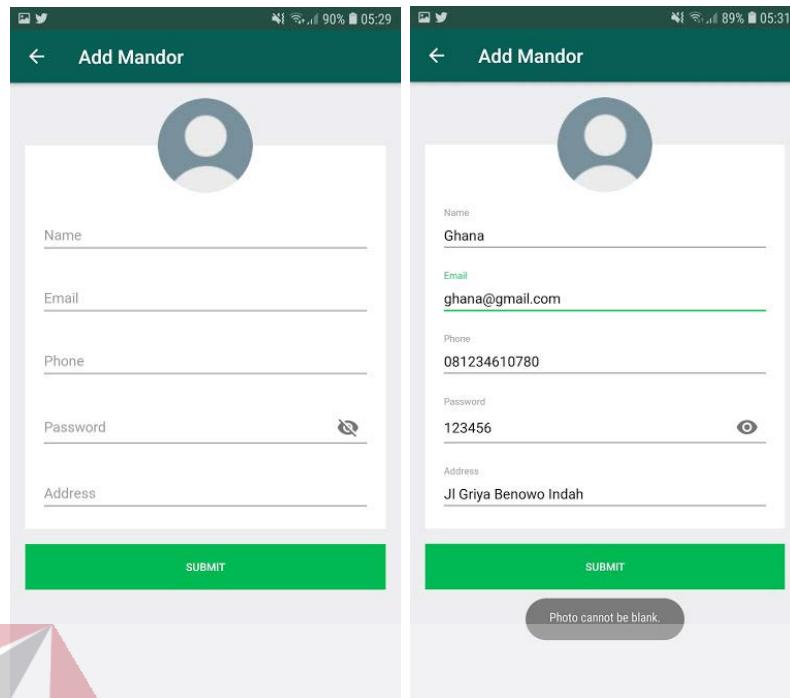
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
	terhadap masukan data benar	2. Pilih gambar 3. Masukkan <i>name</i> 4. Masukkan <i>email</i> 5. Masukkan <i>phone</i> 6. Masukkan <i>address</i> 7. Masukkan <i>password</i> 8. Tekan “submit”			
TR38	Mengetahui respon sistem terhadap kosongnya salah satu data	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Masukkan <i>name</i> 3. Masukkan <i>email</i> 4. Masukkan <i>phone</i> 5. Masukkan <i>address</i> 6. Masukkan <i>password</i> 7. Tekan “submit”	Data berhasil disimpan	Gambar 4.43	Sukses
TR39	Mengetahui respon sistem terhadap kosongnya beberapa data	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih gambar 3. Masukkan <i>name</i> 4. Masukkan <i>email</i> 5. Masukkan <i>phone</i>	Data gagal disimpan	Gambar 4.44	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
		6. Masukkan <i>address</i> 7. Masukkan <i>password</i> 8. Tekan “ <i>submit</i> ”			
TR40	Mengetahui respon sistem terhadap data kosong	1. Pilih “ <i>icon plus</i> ” pada pojok kanan bawah tampilan 2. Tekan “ <i>submit</i> ”	Data gagal disimpan	Gambar 4.45	Sukses
TR41	Mengetahui respon sistem terhadap pengubahan data	1. Pilih salah satu data mandor 2. Pilih gambar 3. Masukkan <i>name</i> 4. Masukkan <i>email</i> 5. Masukkan <i>phone</i> 6. Masukkan <i>address</i> 7. Masukkan <i>password</i> 8. Tekan “ <i>submit</i> ”	Data berhasil disimpan	Gambar 4.46	Sukses
TR42	Mengetahui respon sistem terhadap penghapusan data	1. Pilih salah satu data mandor 2. Pilih “ <i>icon trash</i> ” pada pojok kanan atas tampilan	Data berhasil dihapus	Gambar 4.47	Sukses

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu mandor. Pada kasus ini akan digunakan data benar yang telah disiapkan pada *test data* sebelumnya. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

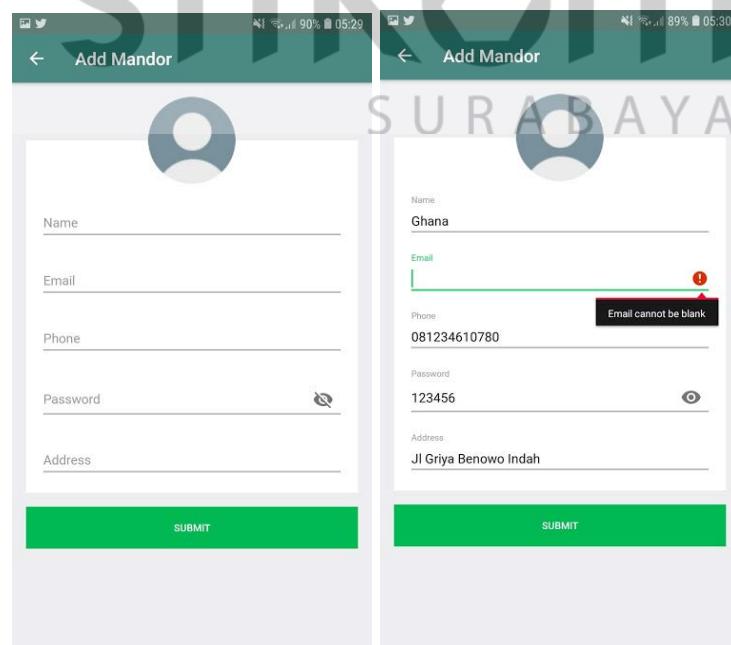


Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu mandor. Pada kasus ini akan digunakan data tanpa data gambar. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



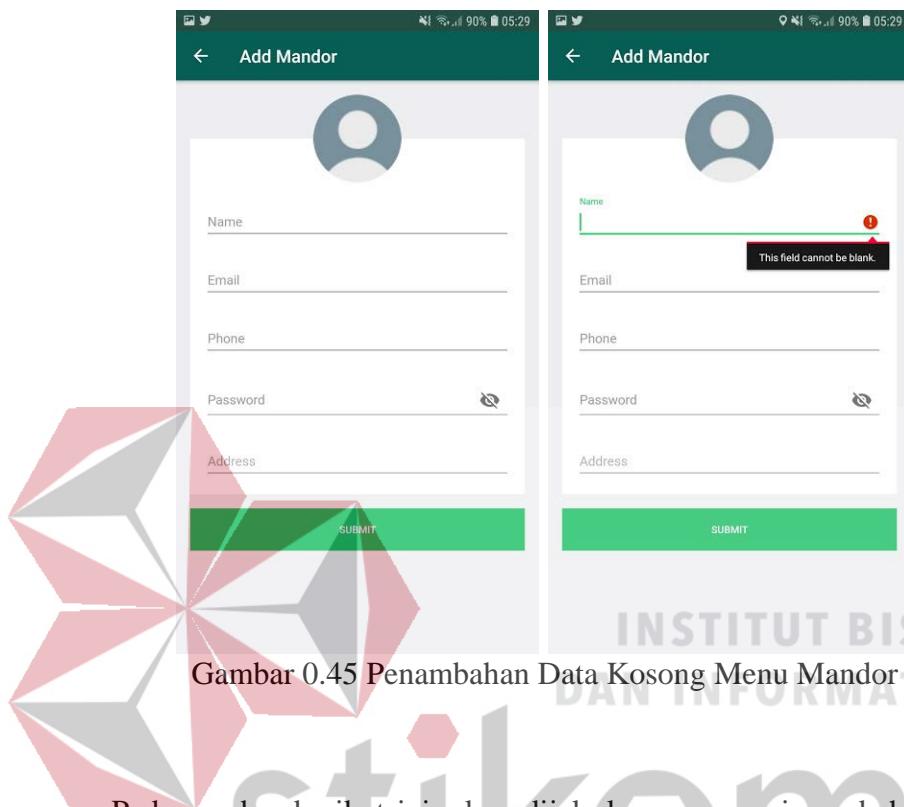
Gambar 0.43 Penambahan Data Menu Mandor Tanpa Gambar

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu mandor. Pada kasus ini akan digunakan data tanpa data email. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

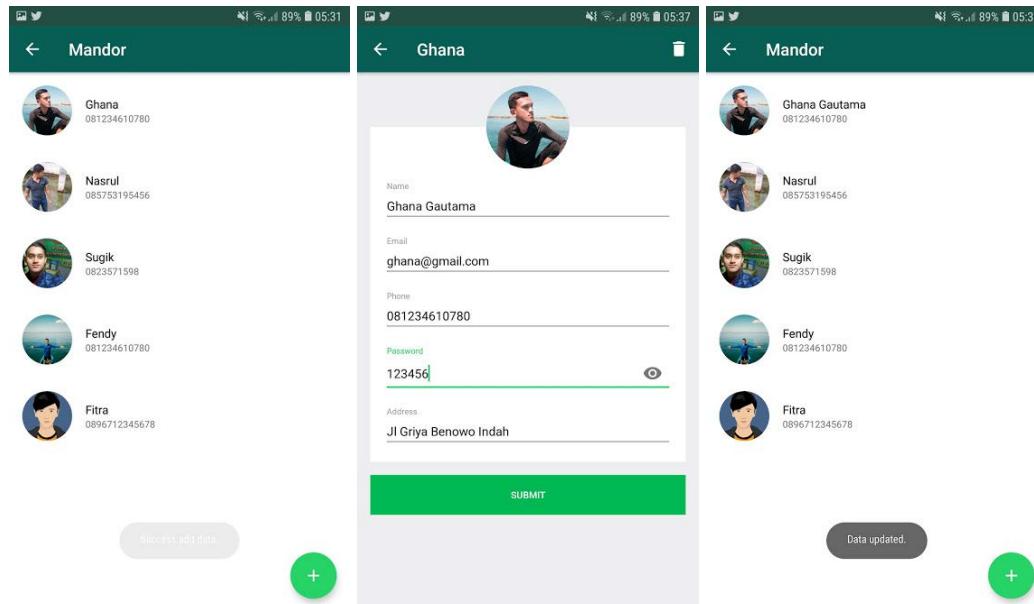


Gambar 0.44 Penambahan Data Menu Mandor Tanpa Email

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu mandor. Pada kasus ini akan digunakan data kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

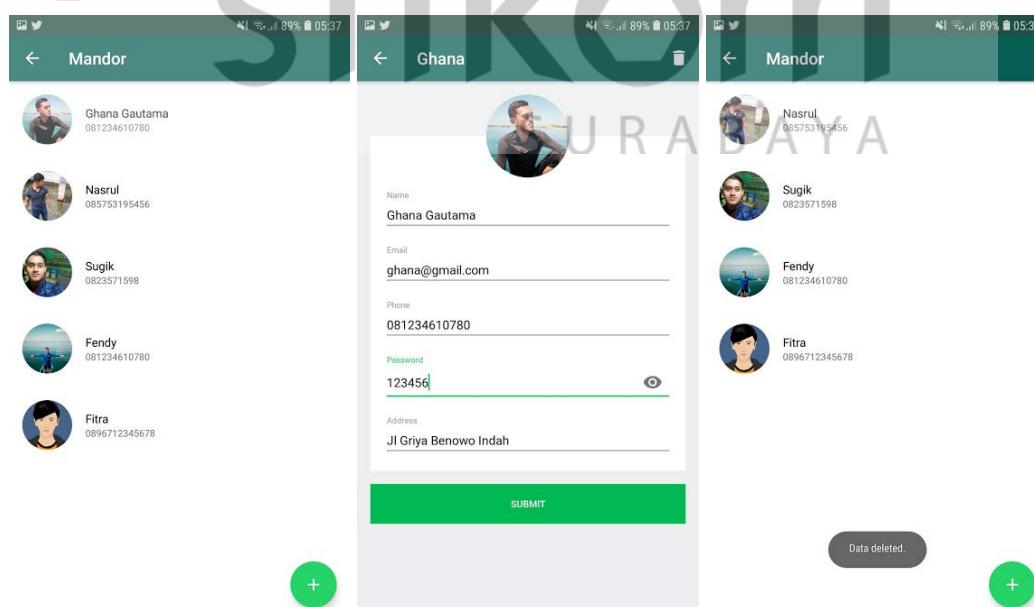


Pada gambar berikut ini, akan dijelaskan mengenai perubahan data pada menu mandor. Pada kasus ini akan digunakan salah satu data yang terdapat pada aplikasi. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.46 Pengubahan Data Menu Mandor

Pada gambar berikut ini, akan dijelaskan mengenai penghapusan data pada menu mandor. Pada kasus ini akan digunakan salah satu data yang terdapat pada aplikasi. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.47 Penghapusan Data Menu Mandor

11. Menu Gudang

Menu gudang ini digunakan untuk mengelola data gudang yang dimiliki oleh kontraktor. Perubahan harga tidak dapat dilakukan pada menu ini. Jika ada perubahan harga pada bahan baku, maka akan dianggap barang baru, karena nilai barang yang sudah berbeda. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.12 berikut ini:

Tabel 4.12 *Test Result* Menu Gudang

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR43	Mengetahui respon sistem terhadap masukan data benar	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “Cash” 3. Masukkan nama barang 4. Masukkan jumlah 5. Masukkan harga satuan 6. Masukkan nama satuan 7. Masukkan gambar 8. Tekan “submit” 	Data berhasil disimpan	Gambar 4.48	Sukses
TR44	Mengetahui respon sistem	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok 	Data gagal disimpan	Gambar 4.49	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
	terhadap kosongnya beberapa data	<p>kanan bawah tampilan</p> <p>2. Pilih “Cash”</p> <p>3. Masukkan jumlah</p> <p>4. Masukkan harga satuan</p> <p>5. Masukkan nama satuan</p> <p>6. Masukkan gambar</p> <p>7. Tekan “submit”</p>			
TR45	Mengetahui respon sistem terhadap masukan data benar	<p>1. Pilih “icon plus” pada pojok kanan bawah tampilan</p> <p>2. Pilih “Piutang”</p> <p>3. Pilih “due date”</p> <p>4. Masukkan nama barang</p> <p>5. Masukkan jumlah</p> <p>6. Masukkan harga satuan</p> <p>7. Masukkan nama satuan</p> <p>8. Masukkan gambar</p>	Data berhasil disimpan	Gambar 4.50	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
		9. Tekan “submit”			
TR46	Mengetahui respon sistem terhadap kosongnya beberapa data	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “Piutang” 3. Pilih “due date” 4. Masukkan jumlah 5. Masukkan harga satuan 6. Masukkan nama satuan 7. Masukkan gambar 8. Tekan “submit” 	Data gagal disimpan	Gambar 4.51	Sukses
TR47	Mengetahui respon sistem terhadap pengubahan data	<ol style="list-style-type: none"> 1. Pilih salah satu data barang gudang 2. Ubah data masukan pada jumlah barang 3. Tekan “save” 	Data berhasil diubah	Gambar 4.52	Sukses
TR48	Mengetahui respon sistem terhadap	<ol style="list-style-type: none"> 1. Pilih salah satu data barang gudang 	Data berhasil dihapus	Gambar 4.53	Sukses

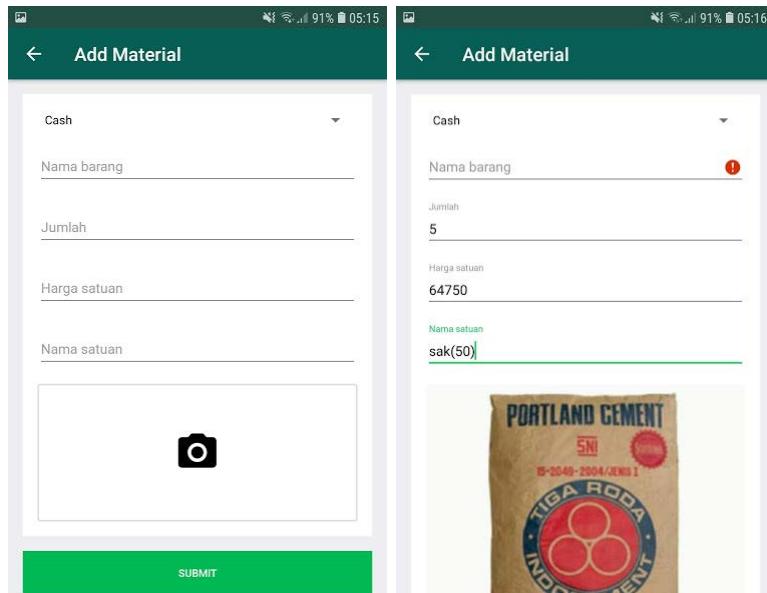
Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
	penghapusan data	2. Ubah data masukan pada jumlah barang 3. Tekan “delete”			

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu gudang dengan metode *cash*. Pada kasus ini akan digunakan data benar yang telah disiapkan sebelumnya pada *data test*. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



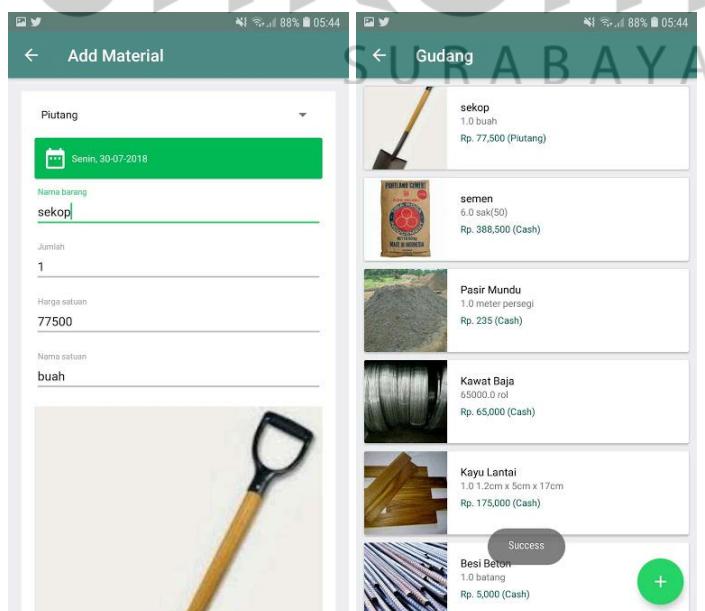
Gambar 0.48. Penambahan Data *Cash* Menu Gudang

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu gudang dengan metode *cash*. Pada kasus ini akan digunakan data yang telah disiapkan sebelumnya tanpa nama barang. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



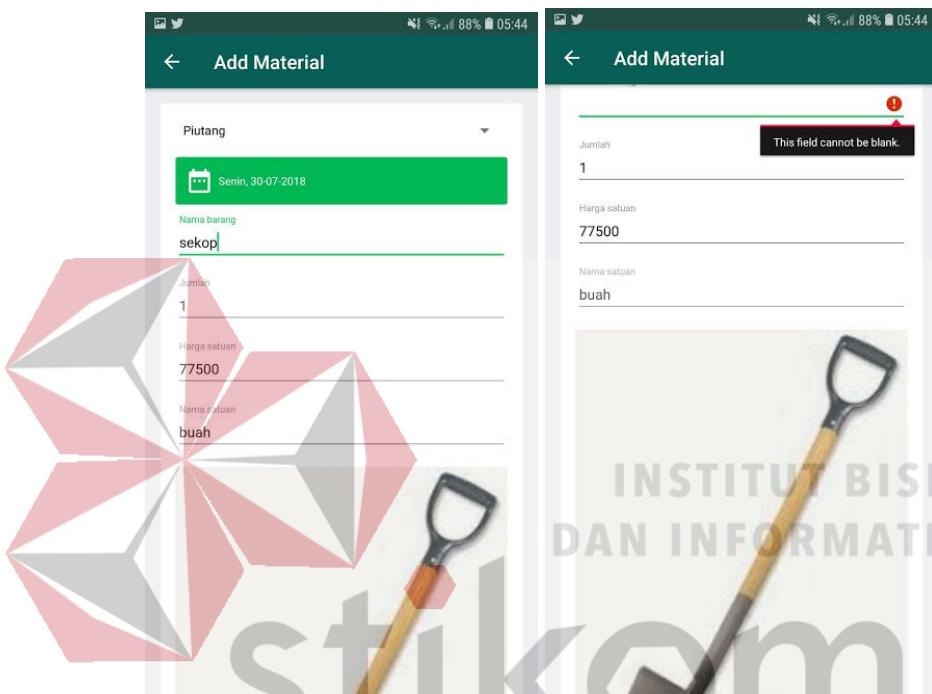
Gambar 0.49. Penambahan Data Cash Menu Gudang Tanpa Nama Barang

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu gudang dengan metode piutang. Pada kasus ini akan digunakan data benar yang telah disiapkan sebelumnya pada *data test*. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



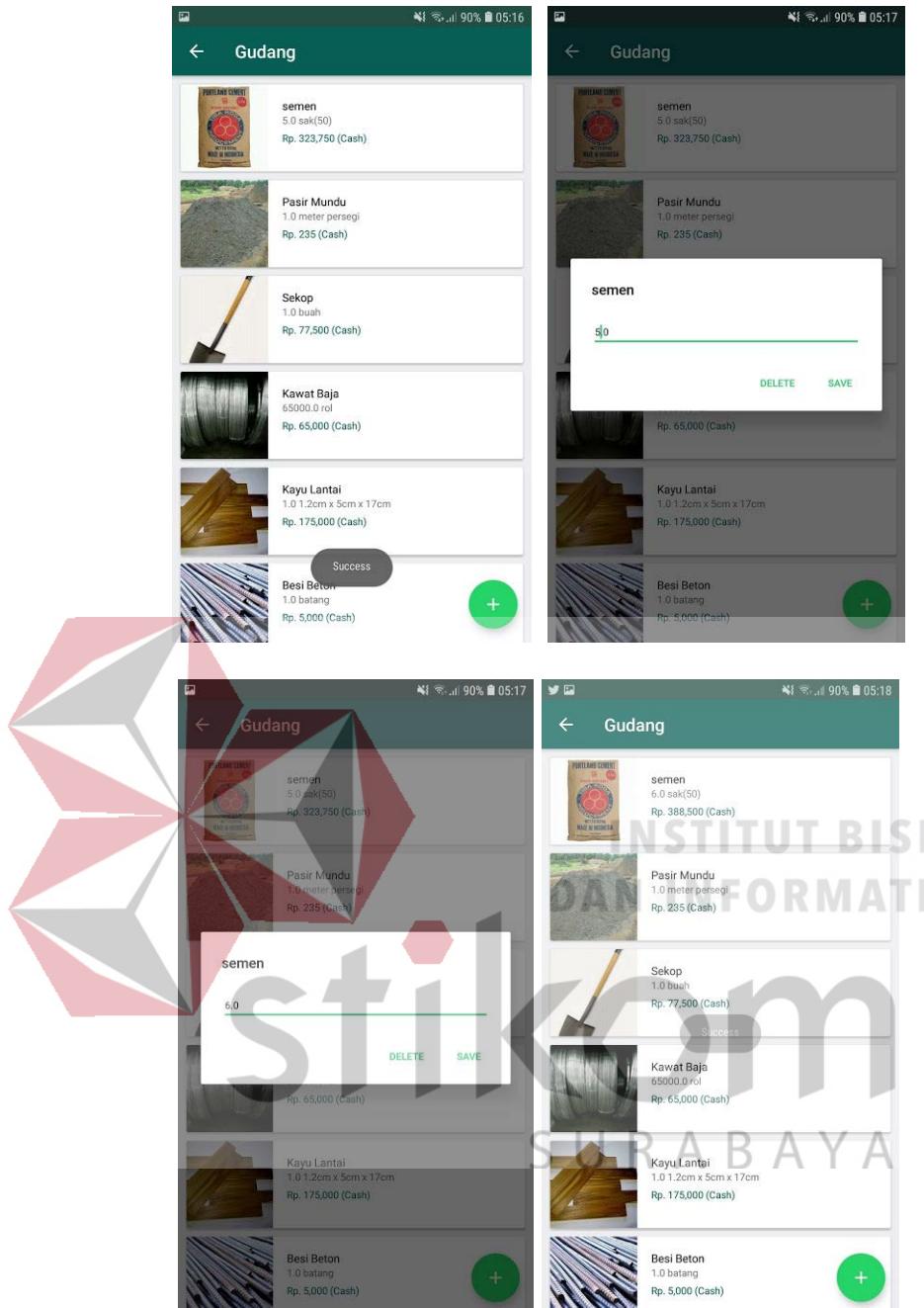
Gambar 0.50. Penambahan Data Piutang Menu Gudang

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu gudang dengan metode piutang. Pada kasus ini akan digunakan data yang telah disiapkan sebelumnya tanpa nama barang. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



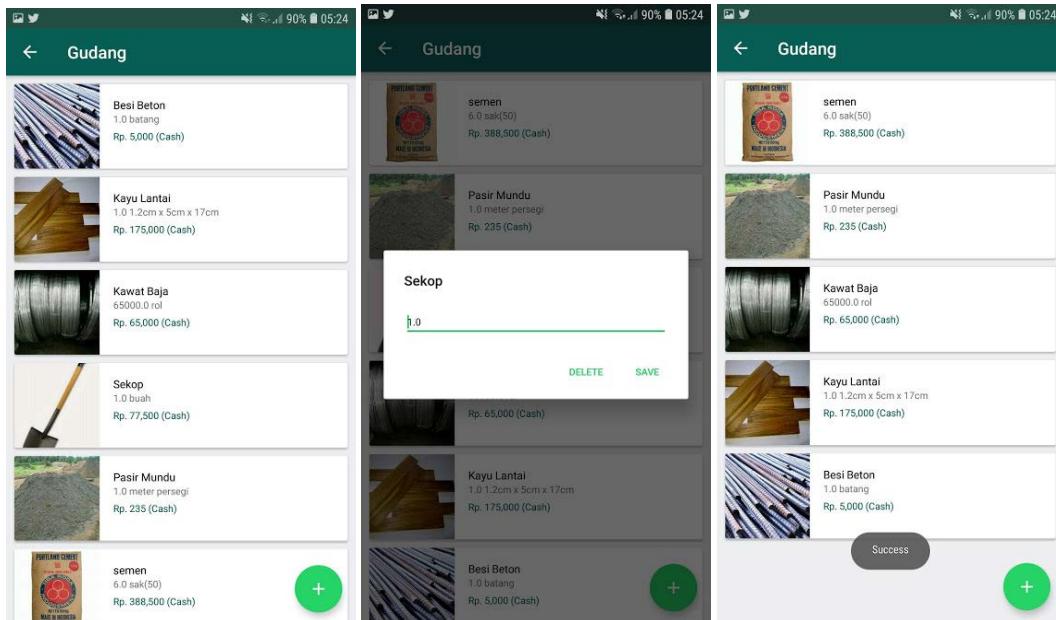
Gambar 0.51 Penambahan Data Piutang Menu Gudang Tanpa Nama Barang

Pada gambar berikut ini, akan dijelaskan mengenai pengubahan data pada menu gudang. Pada kasus ini akan digunakan salah satu data yang ada pada menu gudang. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.52 Pengubahan Data Menu Gudang

Pada gambar berikut ini, akan dijelaskan mengenai penghapusan data pada menu gudang. Pada kasus ini akan digunakan salah satu data yang ada pada menu gudang. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.53 Penghapusan Data Menu Gudang

12. Menu Bayar Piutang

Menu bayar piutang ini digunakan untuk mengelola data piutang yang belum terbayarkan. Untuk melakukan pembayaran kontraktor dapat memilih tipe pembayaran yaitu *cash* atau *transfer*. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.13 berikut ini:

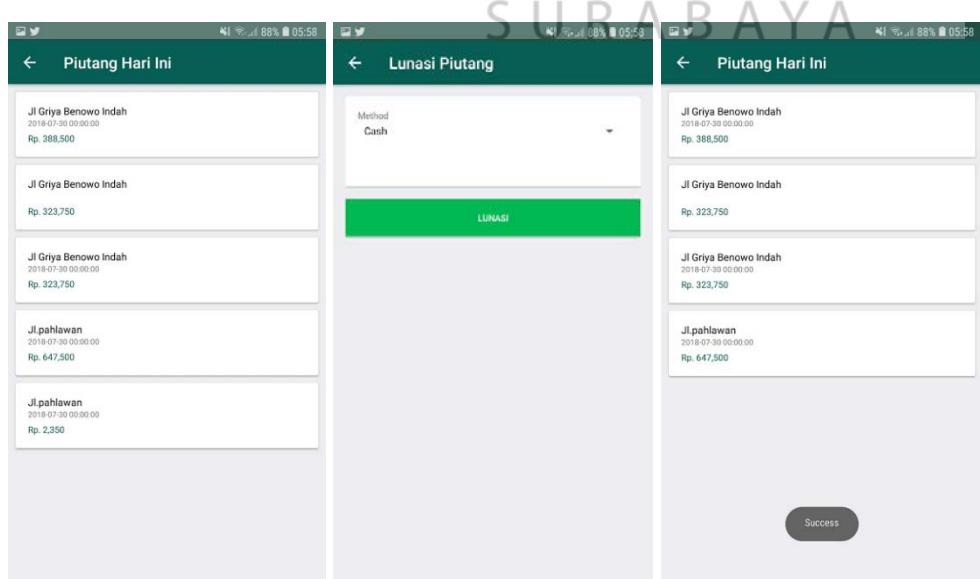
Tabel 4.13 Test Result Menu Piutang

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR49	Mengetahui respon sistem terhadap masukan data benar	<ol style="list-style-type: none"> Pilih salah satu data piutang Pilih <i>method</i> “<i>cash</i>” Tekan tombol “lunasi” 	Data berhasil disimpan	Gambar 4.54	Sukses

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR50	Mengetahui respon sistem terhadap masukan data benar	1. Pilih salah satu data piutang 2. Pilih <i>method</i> “transfer” 3. Tekan tombol “lunasi”	Data berhasil disimpan	Gambar 4.55	Sukses
TR51	Mengetahui respon sistem terhadap kosongnya salah satu data	1. Pilih salah satu data piutang 2. Pilih <i>method</i> “transfer” 3. Tekan tombol “lunasi”	Data gagal disimpan	Gambar 4.56	Sukses

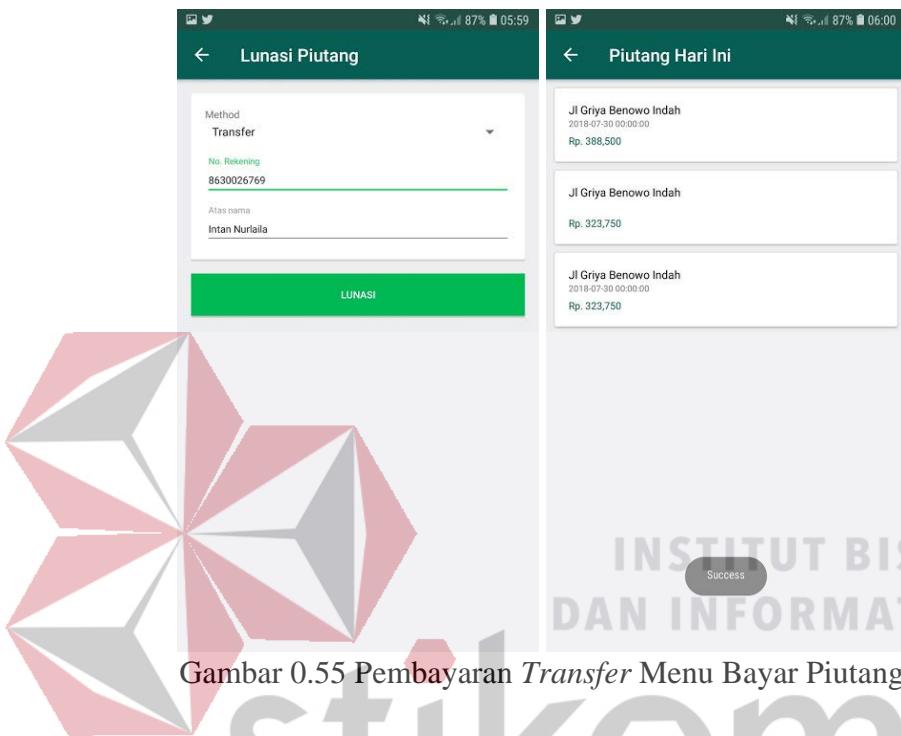
Pada gambar berikut ini, akan dijelaskan mengenai pelunasan data pada menu bayar piutang. Pada kasus ini akan dilakukan pembayaran dengan cara *cash*.

Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



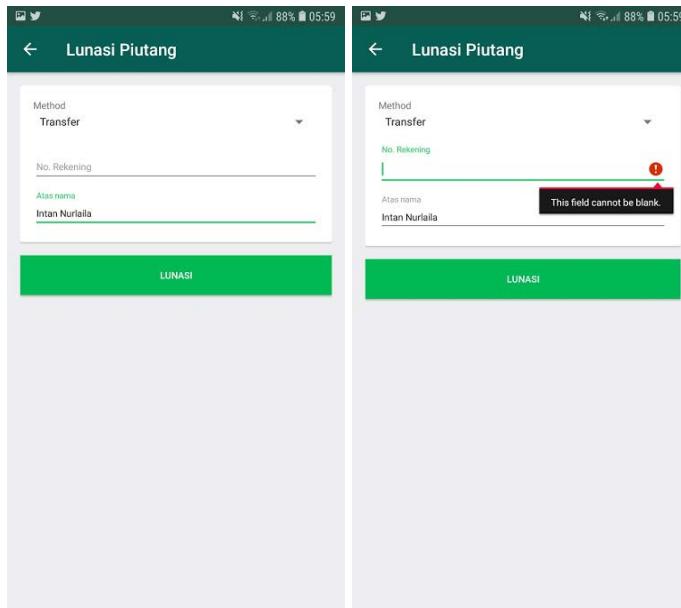
Gambar 0.54 Pembayaran *Cash* Menu Bayar Piutang

Pada gambar berikut ini, akan dijelaskan mengenai pelunasan data pada menu bayar piutang. Pada kasus ini akan dilakukan pembayaran dengan cara *transfer*. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.55 Pembayaran *Transfer* Menu Bayar Piutang

Pada gambar berikut ini, akan dijelaskan mengenai pelunasan data pada menu bayar piutang. Pada kasus ini akan dilakukan pembayaran dengan cara *transfer* tanpa nomor rekening. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.56 Pembayaran *Transfer* Piutang Tanpa Nomor Rekening

B. Testing Mandor

Berikut ini adalah *design testing* yang dibuat untuk tipe pengguna kontraktor pada aplikasi pengawasan kinerja mandor:

1. Halaman *Login*

Halaman *login* ini digunakan untuk melakukan validasi kepada pengguna.

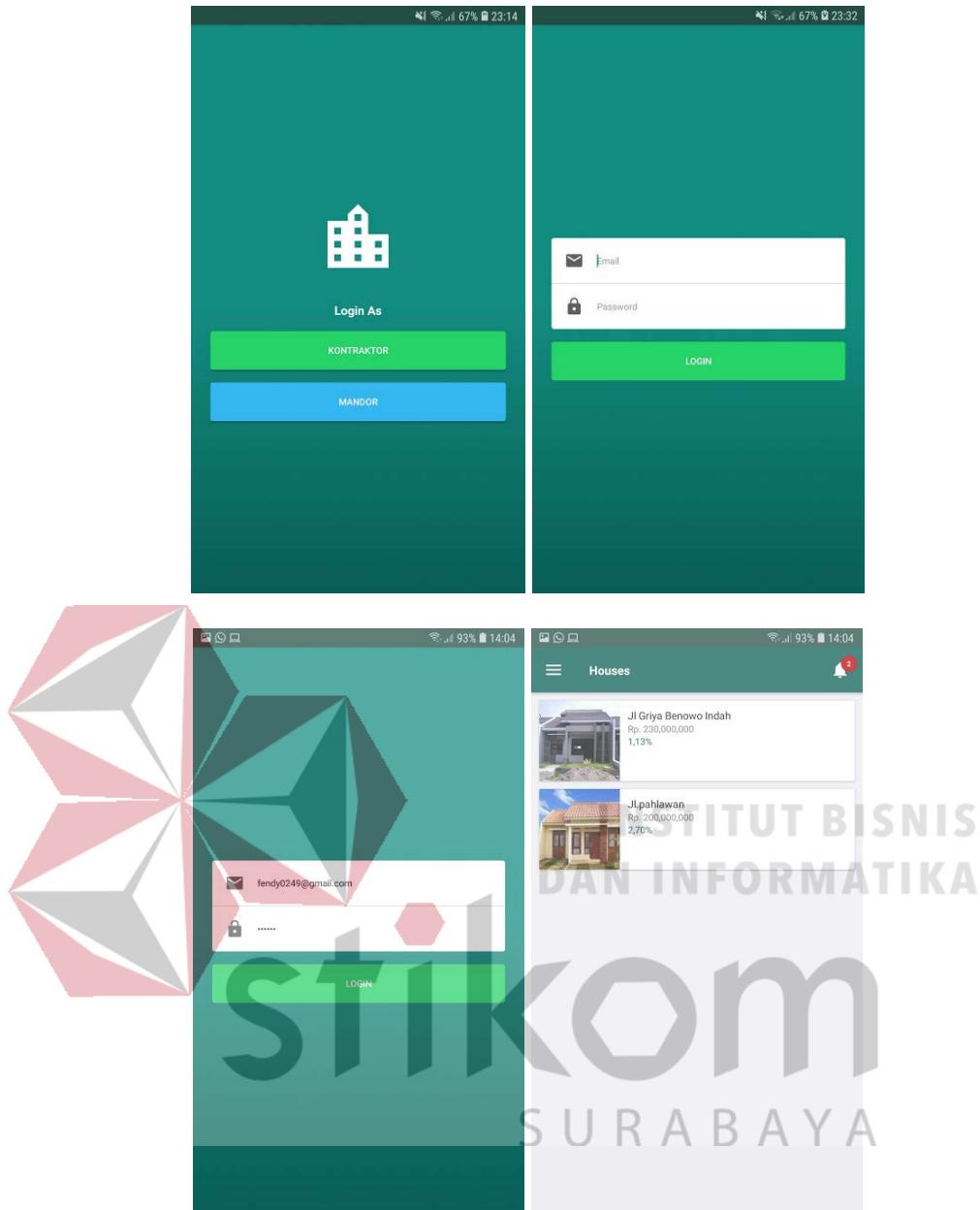
Untuk menggunakan halaman *login*, pengguna diwajibkan untuk mengisi *username* dan *password*. Penjelasan hasil uji coba halaman *login* dapat dilihat pada Tabel 4.14 berikut ini:

Tabel 4.14 *Test Result* Menu *Login*

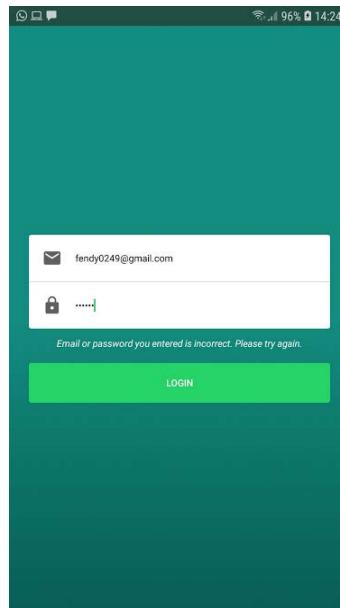
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR52	Mengetahui respon sistem	1. Masukkan <i>email</i>	Pengguna berhasil <i>login</i>	Gambar 4.57	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
	terhadap masukan data benar	2. Masukkan <i>password</i> 3. Tekan <i>login</i>			
TR53	Mengetahui respon sistem terhadap masukan data salah	1. Masukkan <i>email</i> 2. Masukkan <i>password</i> 3. Tekan <i>login</i>	Pengguna gagal <i>login</i>	Gambar 4.58	Sukses
TR54	Mengetahui respon sistem terhadap kosongnya salah satu data	1. Masukkan <i>email</i> 2. Tekan <i>login</i>	Pengguna gagal <i>login</i>	Gambar 4.59	Sukses
TR55	Mengetahui respon sistem terhadap data kosong	Tekan <i>login</i>	Pengguna gagal <i>login</i>	Gambar 4.2.61	Sukses

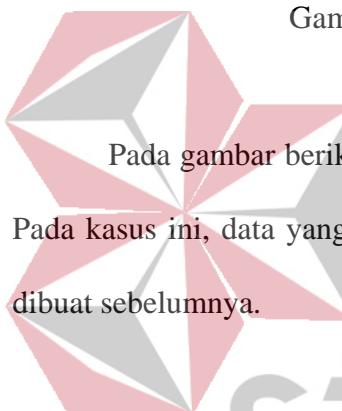
Gambar berikut ini merupakan proses *login* ke dalam aplikasi dengan memasukkan data *username* dan *password* dengan benar. Saat berhasil masuk ke dalam aplikasi, maka tampilan awal yang akan muncul adalah menu *houses*.



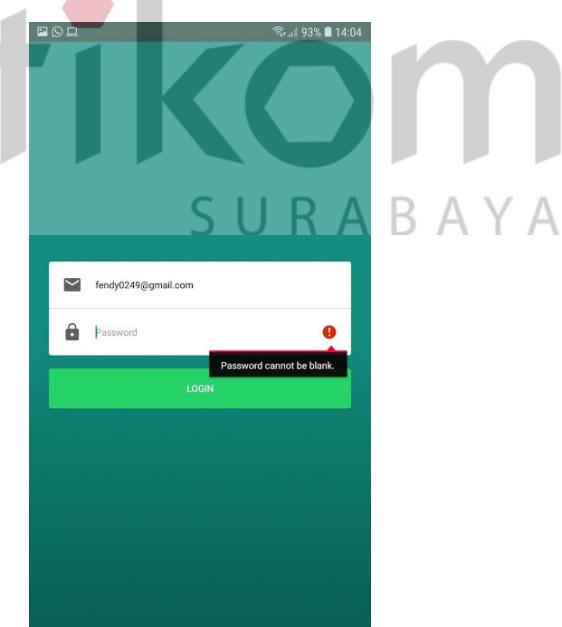
Pada gambar berikut ini, merupakan hasil dari masukan data yang tidak benar. Pada kasus ini, data yang salah adalah *password* sesuai dengan *test data* yang dibuat sebelumnya. Jika data tidak sesuai maka akan muncul pesan “*Email or password you entered is incorrect. Please try again.*”



Gambar 0.58 Login Dengan Data Salah

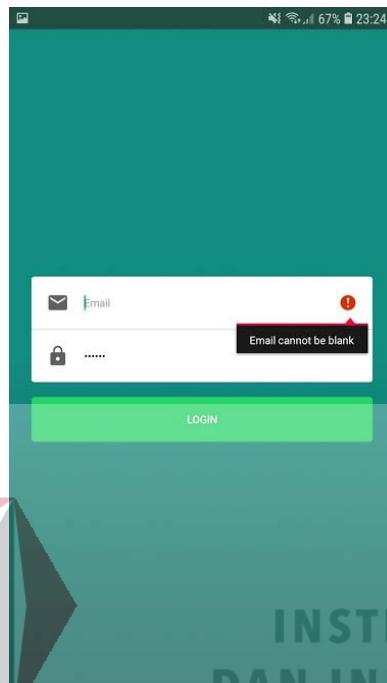


Pada gambar berikut ini, merupakan hasil dari masukan data yang kosong.
Pada kasus ini, data yang kosong adalah *password* sesuai dengan *test data* yang dibuat sebelumnya.



Gambar 0.59 Login Dengan Password Kosong

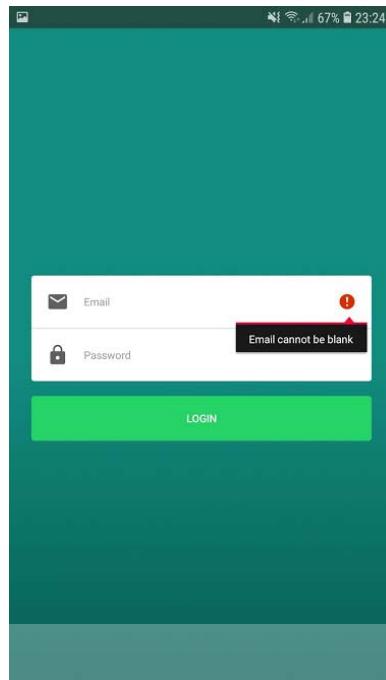
Pada gambar berikut ini, juga merupakan hasil dari masukan data yang kosong. Pada kasus ini, data yang kosong adalah *email* sesuai dengan *test data* yang dibuat sebelumnya.



Gambar 0.60 Login Dengan Email Kosong



Sedangkan pada gambar berikut ini, merupakan hasil dari melakukan *login* tanpa data *email* dan *password*. Dikarenakan sistem pada halaman ini memeriksa validasi email terlebih dahulu, maka kolom email yang dinyatakan harus diisi terlebih dahulu.



Gambar 0.61 Login Tanpa Masukan Data

2. Halaman Utama

Menu utama ini digunakan untuk menunjukkan daftar menu yang tersedia pada aplikasi. Menu utama dapat ditemui pada pojok kiri atas halaman utama.

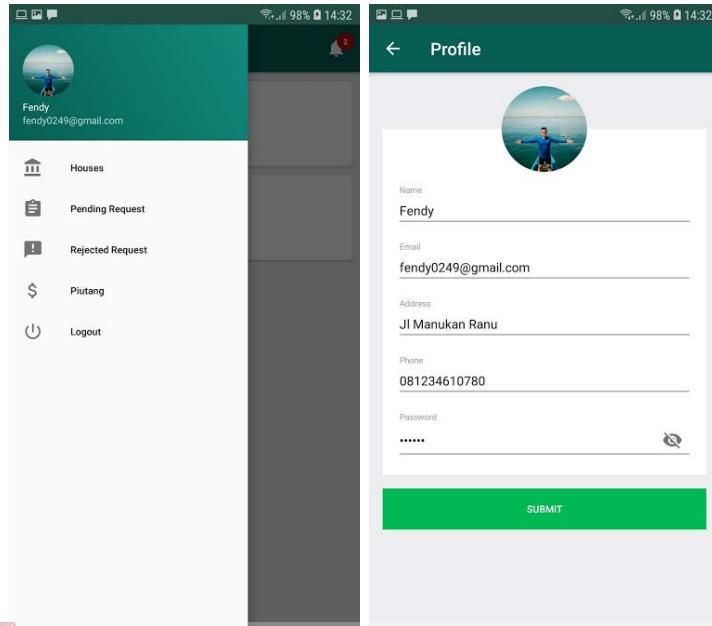
Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.15 berikut ini:

Tabel 4.15 Test Result Menu Utama

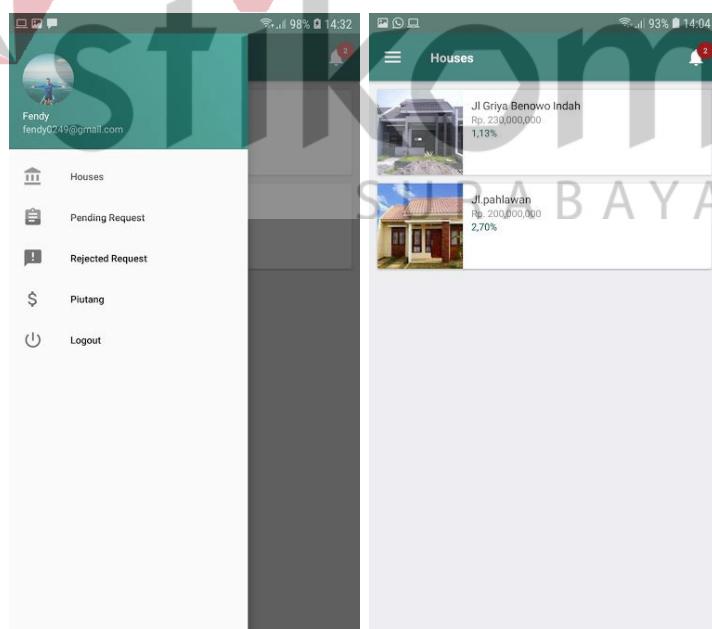
<i>Test Result</i>	<i>Tujuan</i>	<i>Tahapan</i>	<i>Hasil yang Diharapkan</i>	<i>Actual Result</i>	<i>Status</i>
TR56	Mengetahui respon sistem saat menekan “icon user”	Tekan “icon user” pada menu utama	Halaman <i>profile</i> tampil	Gambar 4.62	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR57	Mengetahui respon sistem saat menekan menu “Houses”	Tekan menu “Houses” pada menu utama	Halaman <i>houses</i> tampil	Gambar 4.63	Sukses
TR58	Mengetahui respon sistem saat menekan menu “Pending Request”	Tekan menu “Pending Request” pada menu utama	Halaman <i>pending request</i> tampil	Gambar 4.64	Sukses
TR59	Mengetahui respon sistem saat menekan menu “Rejected Request”	Tekan menu “Rejected Request” pada menu utama	Halaman <i>rejected request</i> tampil	Gambar 4.65	Sukses
TR60	Mengetahui respon sistem saat menekan menu “Piutang”	Tekan menu “Piutang” pada menu utama	Halaman piutang tampil	Gambar 4.66	Sukses
TR61	Mengetahui respon sistem saat menekan menu “Logout”	Tekan menu “Logout” pada menu utama	Kembali ke halaman <i>login</i>	Gambar 4.67	Sukses

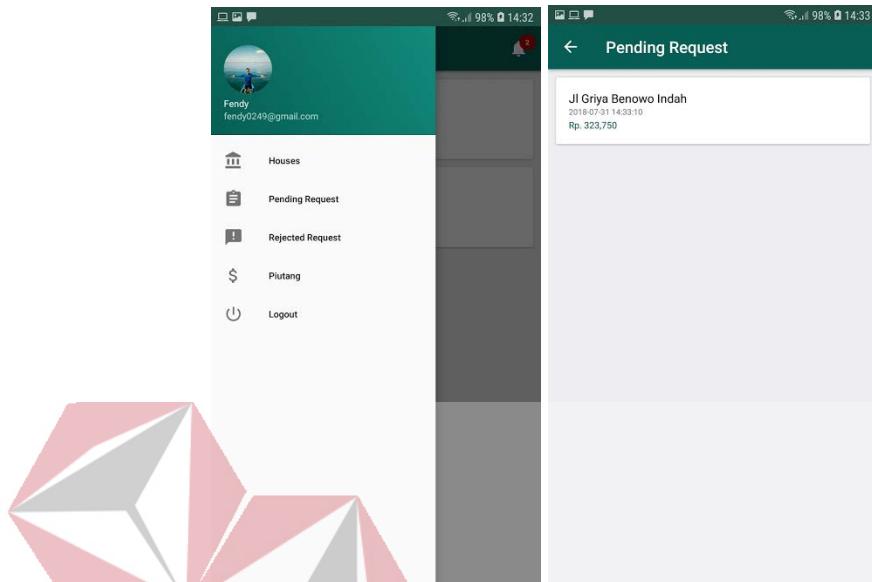
Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan “icon profile”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

Gambar 0.62 Menu *Profile* Mandor

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “houses”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

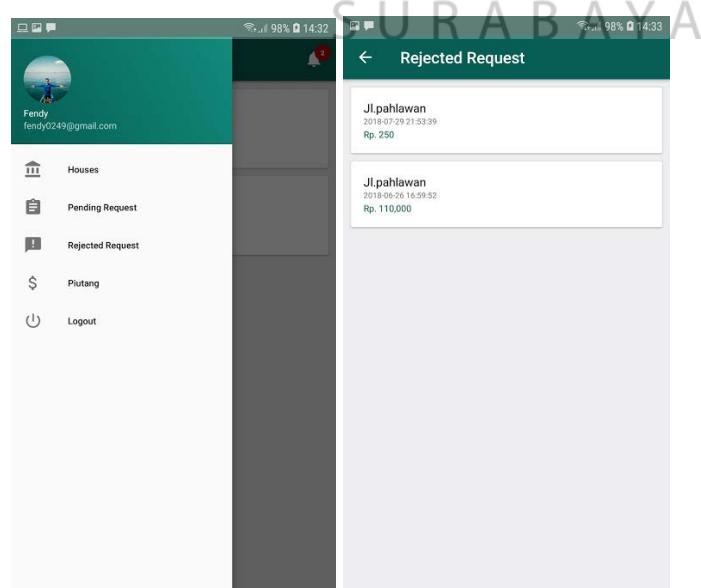
Gambar 0.63 Menu *Houses* Mandor

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “*pending request*”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



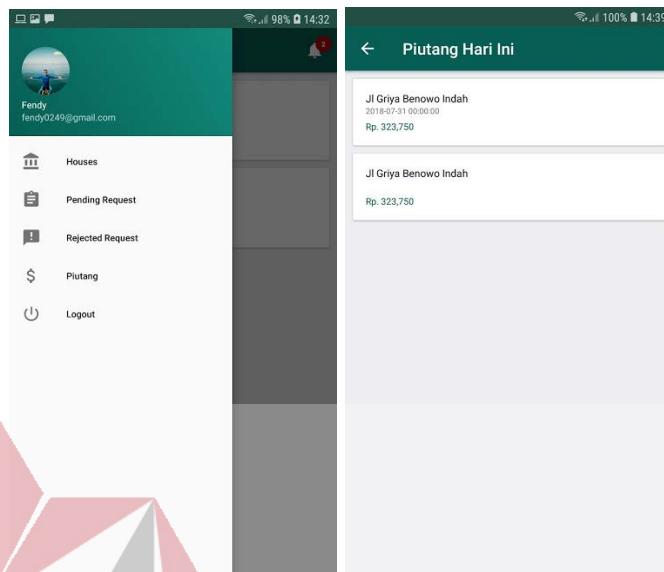
Gambar 0.64 Menu *Pending Request* Mandor

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “*rejected requeste*”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



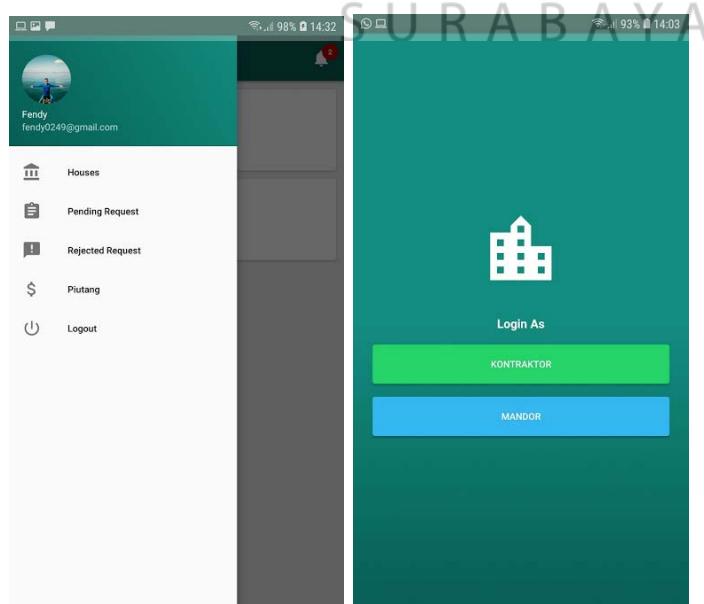
Gambar 0.65 Menu *Rejected Request* Mandor

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “piutang”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.66 Menu Piutang Mandor

Pada gambar berikut ini, merupakan hasil dari tampilan menu utama saat menekan menu “logout”. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.67 Menu Logout Mandor

3. Menu Notifikasi

Menu notifikasi ini digunakan untuk mengetahui bahwa kontraktor telah menyetujui perizinan pembelian yang telah mandor lakukan. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.16 berikut ini:

Tabel 4.16 *Test Result* Menu Notifikasi

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR2	Mengetahui respon sistem jika permohonan pembelian disetujui	<ol style="list-style-type: none"> 1. Pilih salah satu data permintaan yang diterima 2. Pilih “icon camera” pada pojok kanan atas tampilan 3. Masukkan gambar foto nota 4. Masukkan gambar foto barang 5. Tekan ”submit” 	Data permintaan berhasil diterima	Gambar 4.68	Sukses

Pada gambar berikut ini, merupakan hasil dari tampilan menu notifikasi saat permintaan pembelian disetujui. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.68 Menu Notifikasi Mandor

2. Menu *Houses*

Pada menu *houses* ini mandor dapat mengelola data proyek yang ditugaskan oleh kontraktor. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.17 berikut ini:

Tabel 4.17 *Test Result Menu Houses*

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR63	Mengetahui respon sistem saat menampilkan data	Pilih salah satu data houses	Menampilkan detail data houses yang dipilih	Gambar 4.69	Sukses

Pada gambar berikut ini, merupakan hasil dari tampilan menu houses yang hanya ditugaskan pada mandor. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan

Gambar 0.69 Menu *Houses* Mandor

3. Menu Pengeluaran

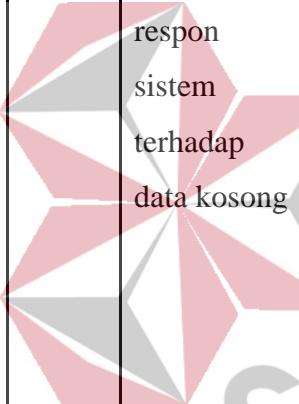
Menu pengeluaran ini digunakan untuk mengolah data pengeluaran proyek yang sedang dikerjakan oleh kontraktor dan mandor. Pada menu ini dikhususkan

untuk pengeluaran dengan metode tunai, piutang, dan pengambilan barang gudang.

Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.18 berikut ini:

Tabel 4.18 *Test Result* Menu Pengeluaran

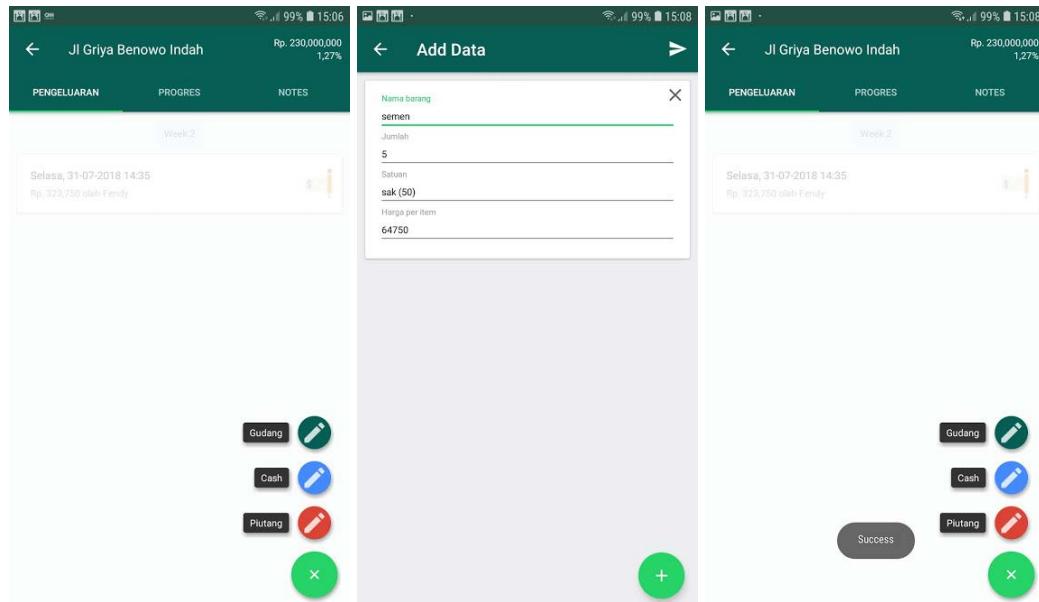
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR64	Mengetahui respon sistem terhadap masukan data benar	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Cash” 3. Pilih tanggal pengeluaran 4. Masukkan nama barang 5. Masukkan jumlah 6. Masukkan satuan 7. Masukkan harga tiap satuan 8. Tekan “icon send” pada pojok kanan atas 	Data berhasil disimpan	Gambar 4.70	Sukses
TR65	Mengetahui respon sistem terhadap kosongnya salah satu data	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Cash” 3. Masukkan nama barang 	Data gagal disimpan	Gambar 4.71	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
		4. Masukkan jumlah 5. Masukkan satuan 6. Masukkan harga tiap satuan 7. Tekan “icon send” pada pojok kanan atas			
TR66	 Mengetahui respon sistem terhadap data kosong	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Cash” 3. Tekan “icon send” pada pojok kanan atas	Data gagal disimpan	Gambar 4.72	Sukses
TR67	Mengetahui respon sistem terhadap masukan data benar	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Gudang” 3. Pilih nama barang 4. Masukkan jumlah 5. Masukkan satuan 6. Masukkan harga tiap satuan 7. Tekan “icon send” pada pojok kanan atas	Data berhasil disimpan	Gambar 4.73	Sukses

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR68	Mengetahui respon sistem terhadap kosongnya salah satu data	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah 2. Pilih menu “Gudang” 3. Pilih nama barang 4. Masukkan satuan 5. Masukkan harga tiap satuan 6. Tekan “icon send” pada pojok kanan atas 	Data gagal disimpan	Gambar 4.74	Sukses
TR69	Mengetahui respon sistem terhadap data kosong	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Gudang” 3. Tekan “icon send” pada pojok kanan atas 	Data gagal disimpan	Gambar 4.75	Sukses
TR70	Mengetahui respon sistem jika salah satu data dipilih	Pilih salah satu data pada tampilan menu terbayar	Muncul <i>detail tampilan pengeluaran</i>	Gambar 4.76	Sukses

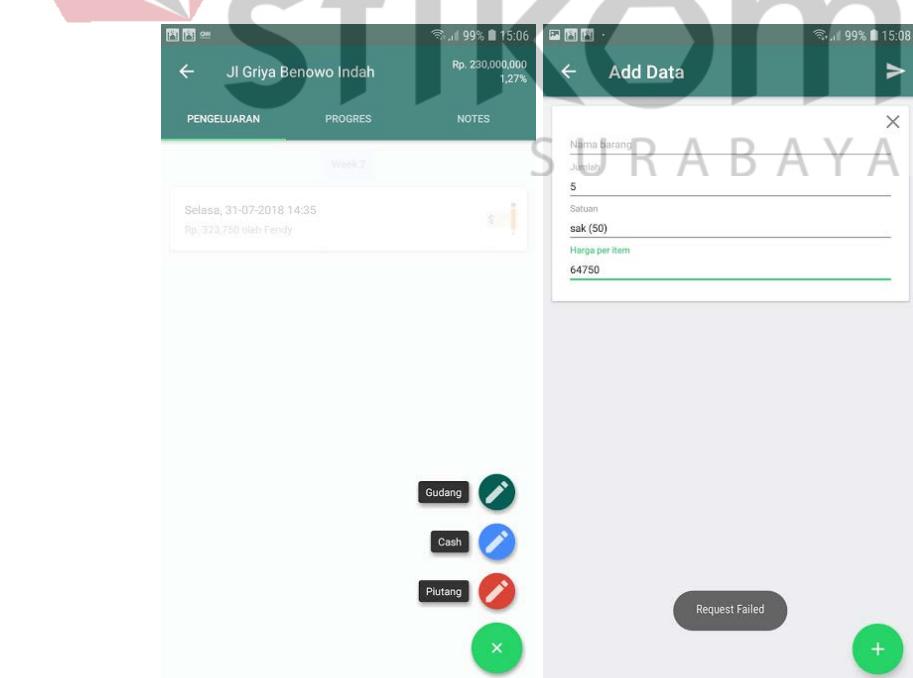
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR71	Mengetahui respon sistem terhadap masukan data benda	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih menu “Piutang” 3. Pilih “due date” 4. Pilih nama barang 5. Masukkan jumlah 6. Masukkan satuan 7. Masukkan harga per item 8. Tekan “icon send” pada pojok kanan atas 	Data berhasil disimpan	Gambar 4.77	Sukses

Pada gambar berikut ini, ditunjukkan proses penambahan data *cash* pada menu pengeluaran. Pada kasus ini akan digunakan *data test* yang telah disediakan sebelumnya. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



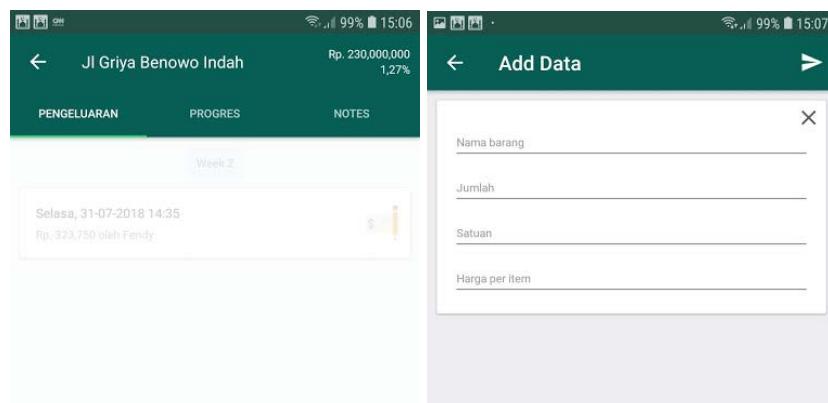
Gambar 0.70 Tambah Data Pengeluaran *Cash*

Pada gambar berikut ini, ditunjukkan proses penambahan data *cash* pada menu pengeluaran. Pada kasus ini akan digunakan *data test* dengan mengosongkan kolom nama barang. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan



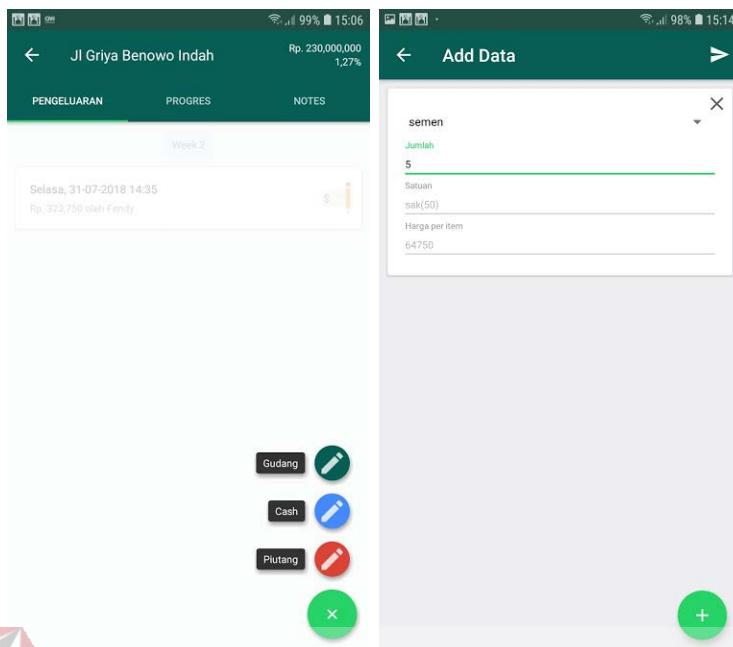
Gambar 0.71 Tambah Data Pengeluaran *Cash* Tanpa Nama Barang

Pada gambar berikut ini, ditunjukkan proses penambahan data *cash* pada menu pengeluaran. Pada kasus ini akan digunakan data kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



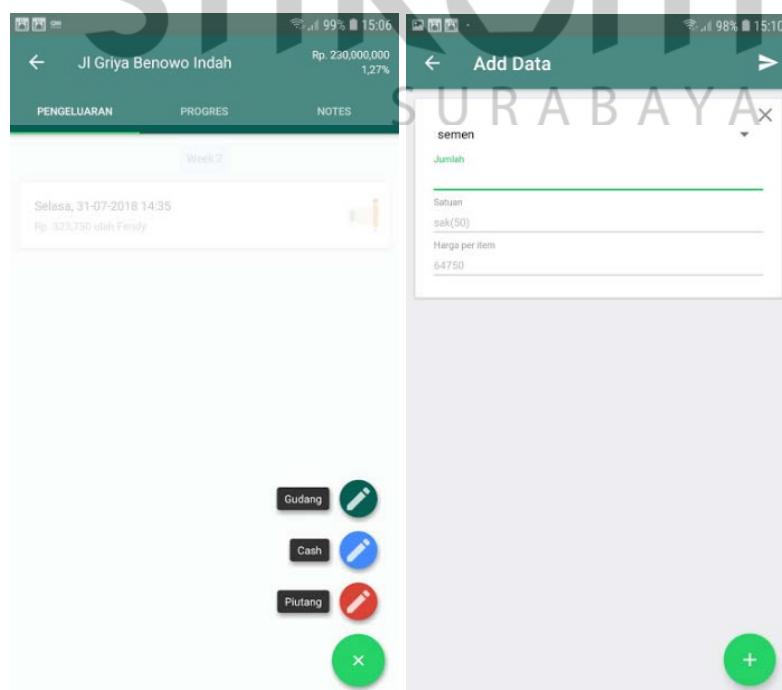
Gambar 0.72 Tambah Data Pengeluaran Cash Kosong

Pada gambar berikut ini, ditunjukkan proses penambahan data gudang pada menu pengeluaran. Pada kasus ini akan digunakan *data test* yang telah disediakan sebelumnya. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



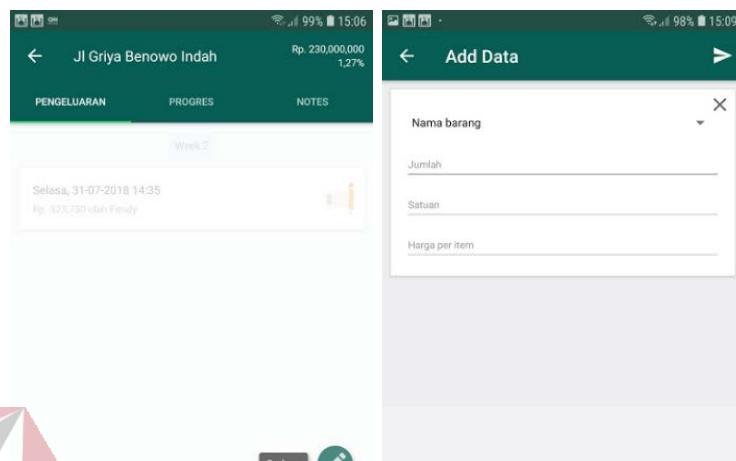
Gambar 0.73 Tambah Data Pengeluaran Gudang

Pada gambar berikut ini, ditunjukkan proses penambahan data gudang pada menu pengeluaran. Pada kasus ini akan digunakan *data test* dengan mengosongkan kolom jumlah barang yang seharusnya dipilih pda tampilan ini. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



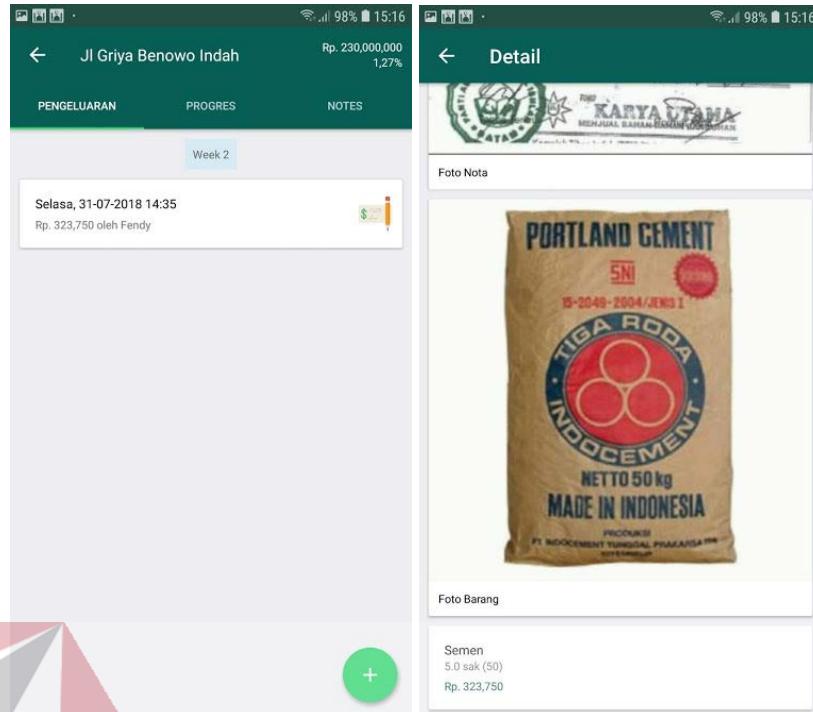
Gambar 0.74 Tambah Data Pengeluaran Gudang

Pada gambar berikut ini, ditunjukkan proses penambahan data gudang pada menu pengeluaran. Pada kasus ini akan digunakan data kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



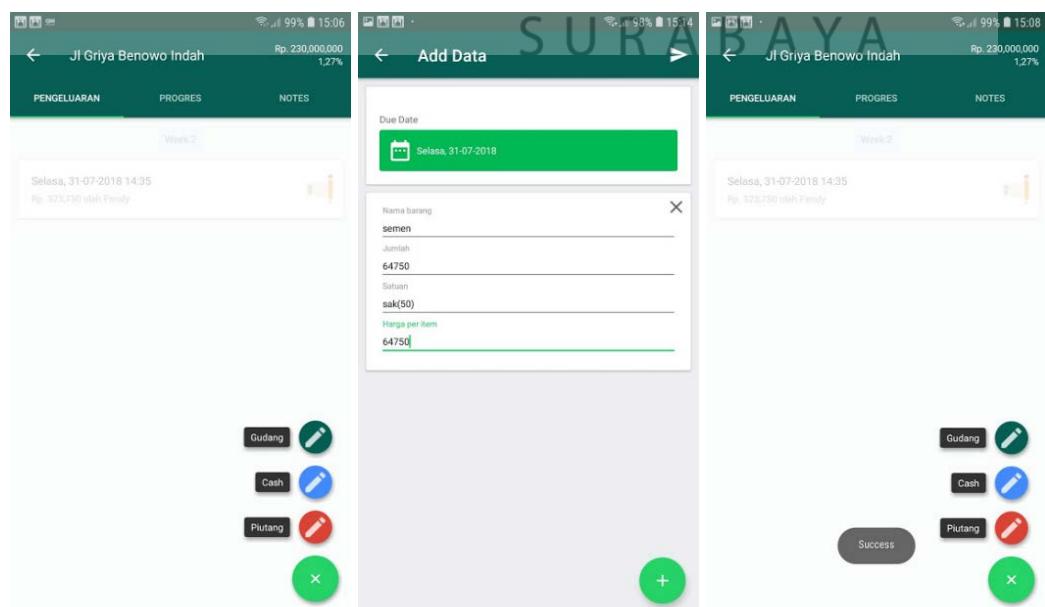
Gambar 0.75 Tambah Data Kosong Pengeluaran Gudang

Pada gambar berikut ini, ditunjukkan daftar data pengeluaran pada menu pengeluaran. Pada kasus ini akan muncul *data test* yang telah disediakan sebelumnya. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.76 Tampilan Detail Data Pengeluaran

Pada gambar berikut ini, ditunjukkan proses penambahan data piutang pada menu pengeluaran. Pada kasus ini akan digunakan *data test* yang telah disediakan sebelumnya. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.77 Tambah Data Pengeluaran *Piutang*

4. Menu *Progress*

Menu *progress* ini digunakan untuk mengolah data kemajuan proyek yang sedang dikerjakan oleh kontraktor dan mandor. Menu ini dibutuhkan agar kontraktor selalu mengetahui kondisi terakhir pekerjaan proyek. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.19 berikut ini:

Tabel 4.19 *Test Result* Menu *Progress*

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR72	Mengetahui respon sistem terhadap masukan data benar	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “work start” 3. Pilih gambar 4. Tekan “submit” 	Data berhasil disimpan	Gambar 4.79	Sukses
TR73	Mengetahui respon sistem terhadap data kosong	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “work start” 3. Tekan “submit” 	Data gagal disimpan	Gambar 4.80	Sukses
TR74	Mengetahui respon sistem terhadap masukan data benar	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “work end” 3. Pilih gambar 4. Tekan “submit” 	Data berhasil disimpan	Gambar 4.81	Sukses

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR75	Mengetahui respon sistem terhadap data kosong	1. Pilih “icon plus” pada pojok kanan bawah tampilan 2. Pilih “work end” 3. Tekan “submit”	Data gagal disimpan	Gambar 4.82	Sukses

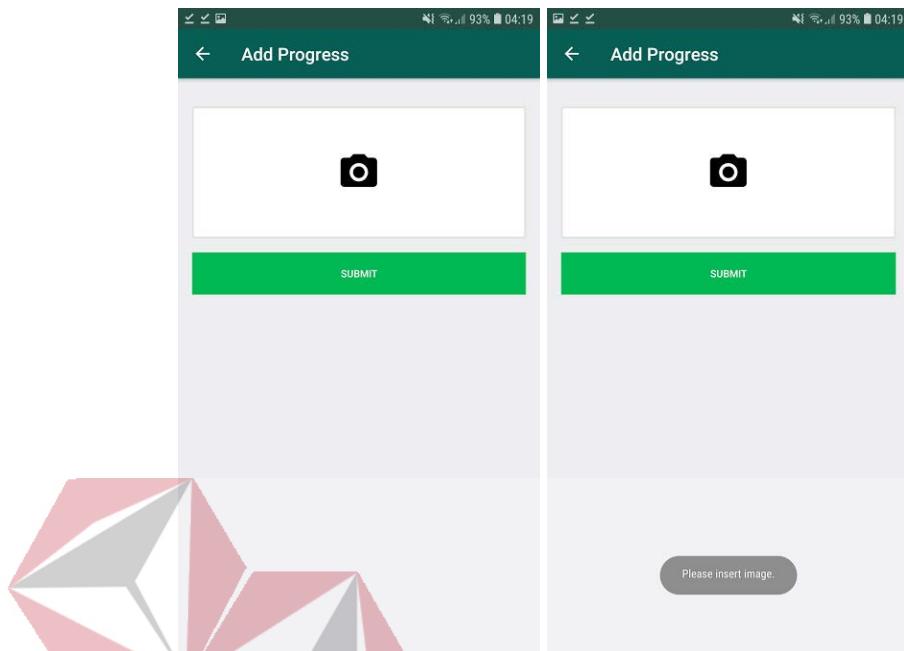
Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *progress* yang terdapat pada aplikasi. Pada kasus ini akan ditambahkan gambar awal pekerjaan. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.78 Menambah Data *Work Start* Menu *Progress*

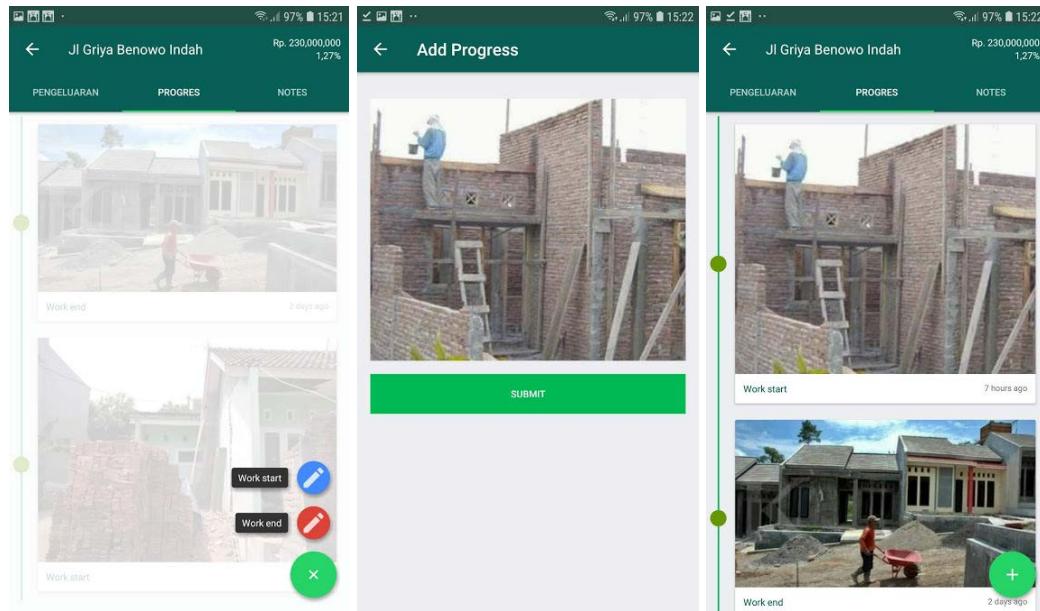
Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *progress* yang terdapat pada aplikasi. Pada kasus ini akan ditambahkan

gambar kosong saat awal pekerjaan. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



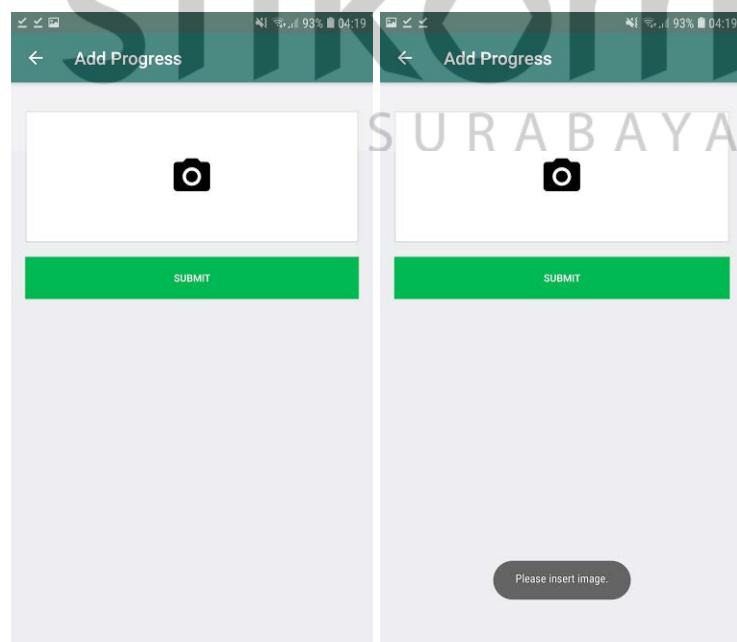
Gambar 0.79 Menambah Data Kosong Work Start Menu *Progress*

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *progress* yang terdapat pada aplikasi. Pada kasus ini akan ditambahkan gambar akhir pekerjaan. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.80 Menambah Data Work End Menu Progress

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *progress* yang terdapat pada aplikasi. Pada kasus ini akan ditambahkan gambar kosong saat akhir pekerjaan. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.81 Menambah Data Kosong Work End Menu Progress

5. Menu Note

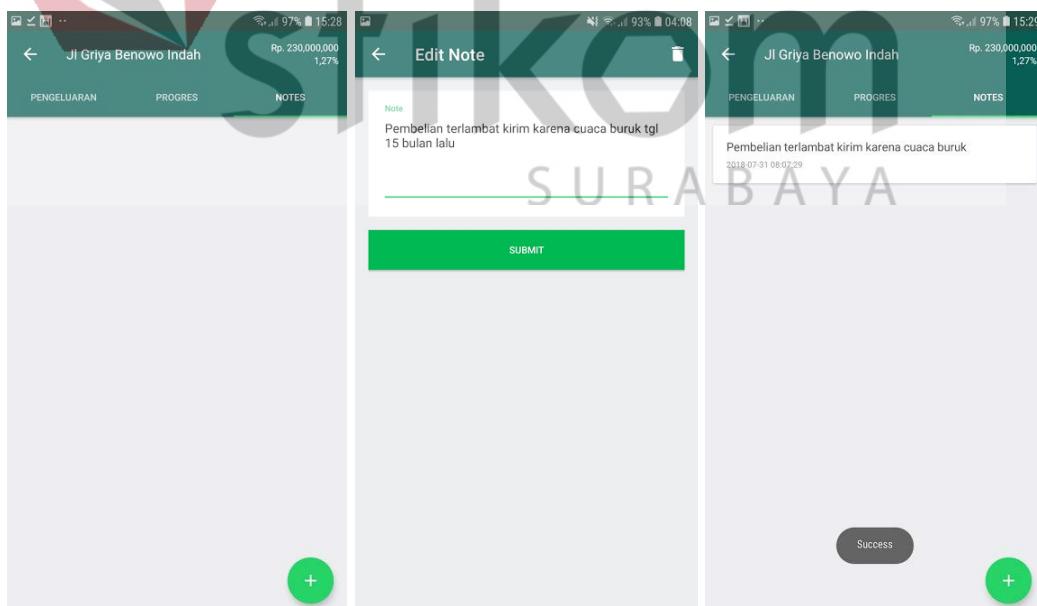
Menu *note* ini digunakan untuk mengolah data catatan proyek yang sedang dikerjakan oleh kontraktor dan mandor. Catatan ini dipergunakan sebagai pengingat hal penting yang perlu diketahui oleh kontraktor dan mandor. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.20 berikut ini:

Tabel 4.20 *Test Result* Menu Note

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR76	Mengetahui respon sistem terhadap masukan data benar	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada kanan bawah tampilan 2. Masukkan <i>note</i> 3. Tekan “submit” 	Data berhasil disimpan	Gambar 4.83	Sukses
TR77	Mengetahui respon sistem terhadap pengubahan data	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada kanan pojok bawah tampilan 2. Ubah <i>note</i> 3. Tekan “submit” 	Data berhasil diubah	Gambar 4.84	Sukses
TR78	Mengetahui respon sistem terhadap data kosong	<ol style="list-style-type: none"> 1. Pilih “icon plus” pada kanan bawah tampilan 2. Tekan “submit” 	Data gagal disimpan	Gambar 4.85	Sukses

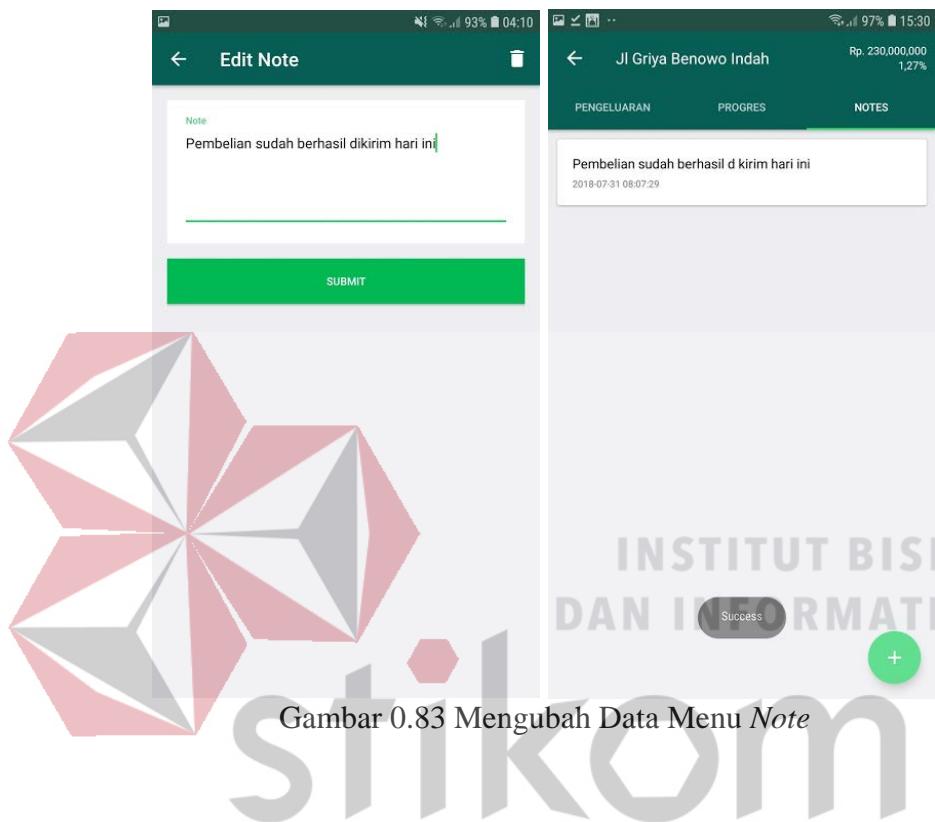
<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR79	Mengetahui respon sistem terhadap penghapusan data	<ol style="list-style-type: none"> Pilih “icon plus” pada pojok kanan bawah tampilan Pilih “icon trash” pada pojok kanan atas tampilan 	Data berhasil terhapus	Gambar 4.86	Sukses

Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *note*. Pada kasus ini akan digunakan data benar yang telah dipersiapkan sebelumnya pada *test data*. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

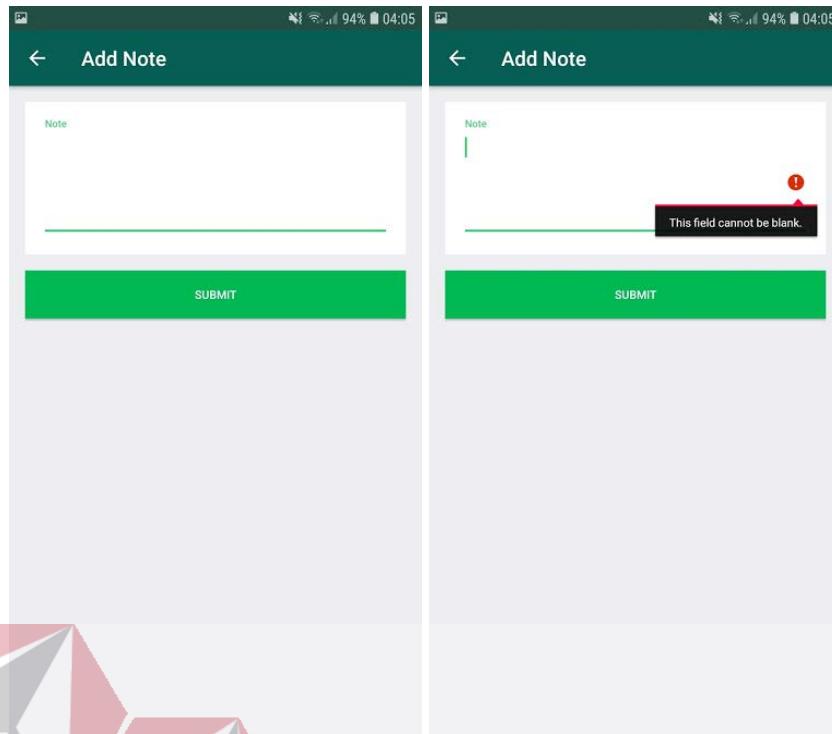


Gambar 0.82 Menambahkan Data Menu Note

Pada gambar berikut ini, akan dijelaskan mengenai pengubahan data pada menu *note*. Pada kasus ini akan digunakan data benar yang telah dipersiapkan sebelumnya pada *test data*. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.

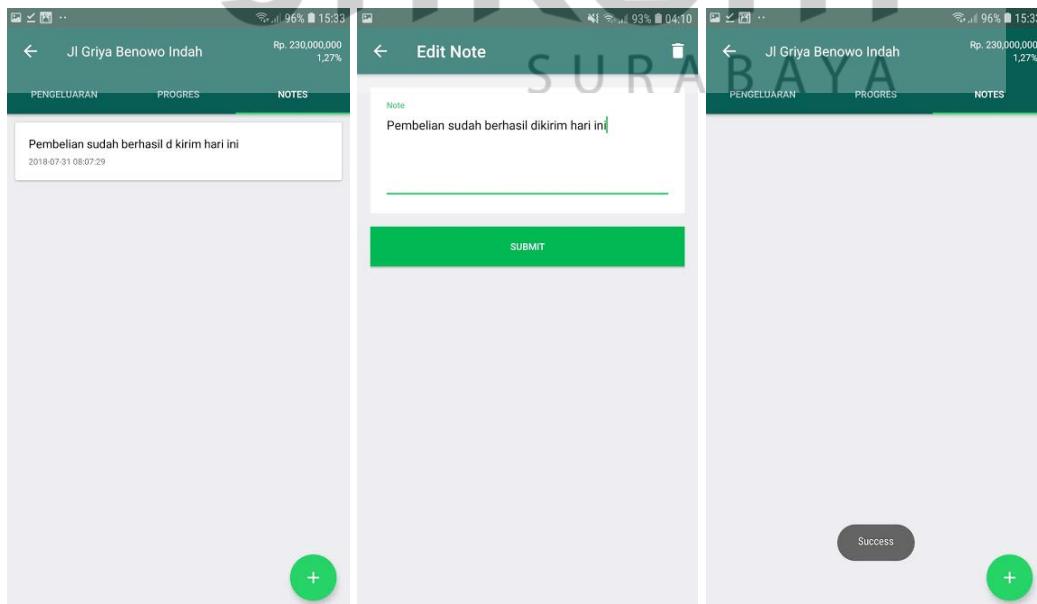


Pada gambar berikut ini, akan dijelaskan mengenai penambahan data pada menu *note*. Pada kasus ini akan digunakan data kosong. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.84 Menambahkan Data Kosong Menu Note

Pada gambar berikut ini, akan dijelaskan mengenai penghapusan data pada menu *note* yang terdapat pada aplikasi. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.85 Menghapus Data Menu Note

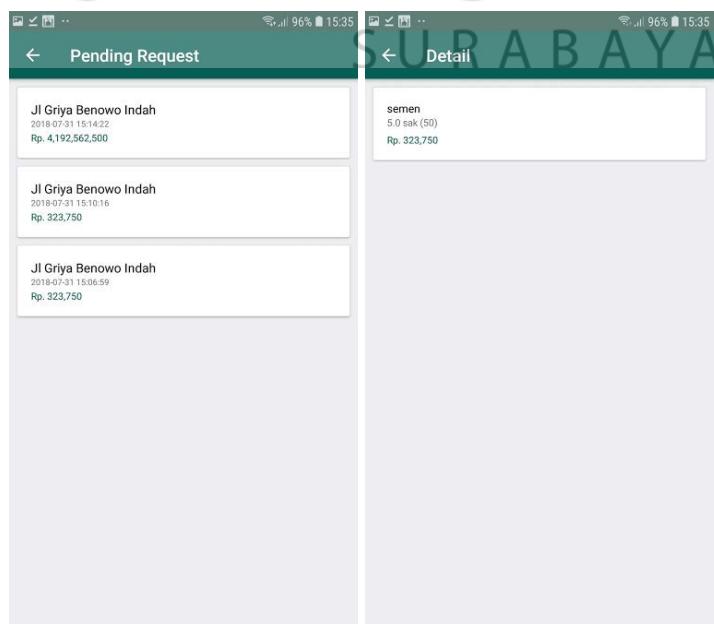
6. Menu *Pending Request*

Menu *pending request* ini berfungsi sebagai catatan daftar permintaan yang sedang dalam proses persetujuan. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.21 berikut ini:

Tabel 4.21 *Test Result* Menu *Pending Request*

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR80	Mengetahui respon sistem saat menampilkan data	Pilih salah satu data <i>pending payment</i>	Menampilkan detail data <i>pending payment</i> yang dipilih	Gambar 4.87	Sukses

Pada gambar berikut ini, akan dijelaskan mengenai daftar data pada menu *pending request* yang terdapat pada aplikasi. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.86 Menu *Pending Request* Mandor

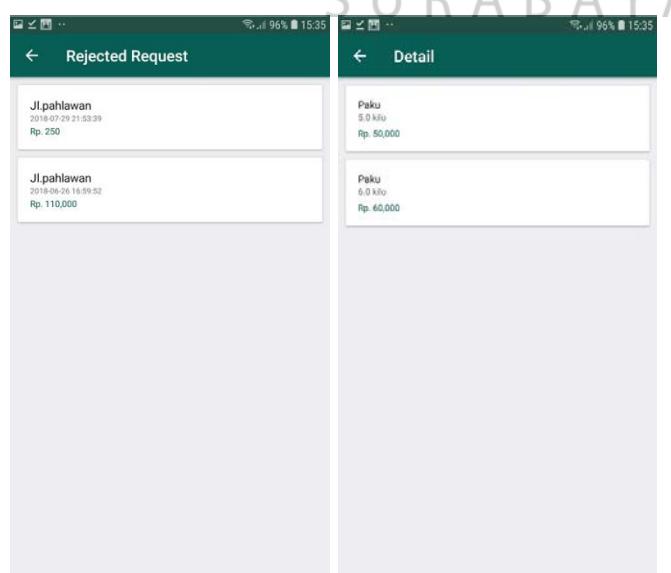
7. Menu *Rejected Request*

Menu *rejected request* ini berfungsi sebagai catatan daftar permintaan yang telah ditolak oleh kontraktor setelah proses persetujuan. Penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.22 berikut ini:

Tabel 4.22 *Test Result* Menu *Rejected Request*

Test Result	Tujuan	Tahapan	Hasil yang Diharapkan	Actual Result	Status
TR81	Mengetahui respon sistem saat menampilkan data	Pilih salah satu data <i>rejected request</i>	Menampilkan detail data <i>rejected request</i> yang dipilih	Gambar 4.88	Sukses

Pada gambar berikut ini, akan dijelaskan mengenai daftar data pada menu *rejected request* yang terdapat pada aplikasi. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.87 Menu *Rejected Request* Mandor

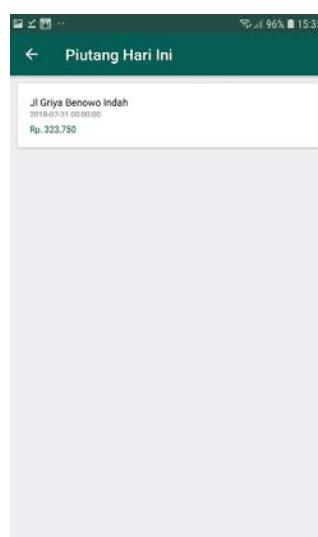
8. Menu Piutang

Menu piutang ini berfungsi sebagai catatan daftar pengeluaran piutang. Hal ini diperlukan agar mandor dapat mengingatkan kontraktor bahwa jatuh tempo pembelian piutang pada hari itu penjelasan hasil uji coba menu utama dapat dilihat pada tabel 4.23 berikut ini:

Tabel 4.23 *Test Result* Menu Piutang

<i>Test Result</i>	Tujuan	Tahapan	Hasil yang Diharapkan	<i>Actual Result</i>	Status
TR82	Mengetahui respon sistem saat menampilkan data	Pilih salah satu data piutang	Menampilkan detail data piutang yang dipilih	Gambar 4.89	Sukses

Pada gambar berikut ini, akan dijelaskan mengenai daftar data pada menu piutang yang terdapat pada aplikasi. Pada gambar ini akan ditunjukkan bagaimana hasil tampilan yang dihasilkan.



Gambar 0.88 Menu Piutang Mandor

4.3 Implementation

Pada tahapan ini, akan dilakukan penyerahan aplikasi yang diikuti dengan pemindahan *server*, kode program, dan *database* kepada PT MKDS hingga aplikasi dapat digunakan seluruh fungsinya dengan normal.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil uji coba dan evaluasi yang telah dilakukan pada pembuatan aplikasi pengawasan kinerja mandor pada PT MKDS, maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi pengawasan kinerja mandor ini dapat memberikan informasi kemajuan proyek, penyerapan anggaran proyek dan pengawasan pembelian bahan baku.
2. Aplikasi pengawasan kinerja mandor ini dapat mengurangi frekuensi kunjungan langsung kontraktor dengan fitur *progress* yang terdapat pada aplikasi sehingga pemantauan kemajuan proyek dapat dilakukan kapan saja tanpa harus melakukan kunjungan langsung.
3. Aplikasi pengawasan kinerja mandor ini dapat mengurangi peluang terjadinya kecurangan oleh mandor dengan fitur persetujuan pembelian yang terdapat pada aplikasi sehingga kontraktor dapat mengawasi seluruh pembelian yang dilakukan oleh mandor.

5.2 Saran

Sistem informasi pengawasan kinerja mandor berbasis *mobile* pada PT MKDS ini dapat dikembangkan pada beberapa fitur sebagai berikut:

1. Penerapan pemilihan dan pembayaran langsung ke *supplier* bahan baku

2. Penerapan rancangan anggaran belanja pada proyek
3. Penerapan stok gudang dengan mempertimbangkan jarak tempuh proyek
4. Laporan keuangan yang lebih terperinci
5. Tampilan *user interface* juga dapat dikembangkan saat *user experience* pengguna telah mengalami perubahan ke arah yang lebih efektif.



DAFTAR PUSTAKA

- Departemen Pendidikan Nasional. (2008). Kamus Besar Bahasa Indonesia. Balai Pustaka: Jakarta
- Ervianto, Wulfram I. (2005). Manajemen Proyek Konstruksi. Andi Offset: Yogyakarta.
- Google LLC, Android Studio. (2018). Retrieved 10 05, 2018 from <https://developer.android.com/studio/features?hl=id>
- Gunawan, Rudy (2013). Pendidikan IPS Filosofi, Kondisi, dan Aplikasi. Alfabeta: Bandung
- Kementerian Pendidikan dan Kebudayaan. Kata Dasar Mandor. Kamus Besar Bahasa Indonesia. Retrieved 10 05, 2018 from <https://kbbi.web.id/mandor>
- Nugroho, Adi., Yos, Richard., dan Hettyca, Astuningdyas. (2009) Perancangan Aplikasi Rencana Anggaran Biaya (RAB) (Studi Kasus pada Dinas Pekerjaan Umum Kota Salatiga). Jurnal Informatika : Vol 10, No 1, Mei 2009: 10-18.
- Satzinger, J. W., Jackson, R. B., & Burd, S. D., (2011). Systems Analysis and Design in a Changing World, Sixth edition.
- Sedarmayanti. (2011). Manajemen Sumber Daya Manusia. Reformasi Birokrasi dan Manajemen Pegawai Negeri Sipil, Cetakan Kelima, PT Refika Aditama: Bandung
- Sublime HQ Pty Ltd. Sublime Text 3. (2018). Retrieved 10 05, 2018 from <https://www.sublimetext.com/>
- Supriyanto, Aji. (2007). Web dengan HTML dan XML. Graha Ilmu: Yogyakarta
- Rizal, Syamsu., Retnadi, Eko., Ikhwana, Andri., (2013) Pengembangan Aplikasi Pencarian Lokasi Objek Wisata Terdekat Di Kabupaten Garut Berbasis Android. Jurnal Algoritma. Sekolah Tinggi Teknologi Garut. Vol. 10 No. 1.
- Turban, Efraim et al. (2012). Electronic Commerce: A Managerial Perspective. International Edition: New Jersey Pearson Prentice Hall.